Dioid of Formal Power Series for the Determination of Performance Parameters of Communication Networks

Mábia Daniel-Cavalcante and Rafael Santos-Mendes

Abstract—Discrete Event Dynamic Systems (DEDS) are systems whose state transitions are triggered by events that occur at discrete instants. The communication networks are examples of this kind of systems. The mathematical constraints of some DEDS can be described more adequately using the dioid algebra. Previous works show that the problem of determining performance bounds for communication networks is simplified if modeled using this algebra. The compilation of existing rules and results on this field is called Network Calculus (NC). The goal of this article is to propose a dioid of formal power series for the treatment of NC problems. To illustrate the adequacy of the proposed dioid, we analyze a FIFO multiplexer. The results obtained for this particular system represent an extension of previous results.

Index Terms—DEDS Modeling, Diod Algebra, Diod Systems, Network Calculus, Quality of Service, Service Curves.

I. INTRODUCTION

NETWORKS that support Integrated Services (IntServ) [1] or Differentiated Services (DiffServ) [2] can carry traffic from a wide variety of applications such as video-on-demand (VoD), videoconference and voice over IP (VoIP) [3]. Each application demands different Quality of Service (QoS) requirements, such as maximum end-to-end delay and jitter and minimum transmission rate. This scenario of traffic imposes the determination of QoS parameter bounds and traffic shaping mechanisms, in order to avoid the monopolization of the network resources by one traffic flow. As a consequence, the degradation of the QoS seen by all other traffic flows is avoided.

In this context, the Network Calculus (NC) appeared as a set of rules and results that can be used to compute tight bounds for QoS parameters of communication networks [4]. NC has its roots in [3], [5]–[7]. The theory is mainly presented in [8]–[10] and a short introduction is found in [11]–[12]. NC is based on the idea that a detailed analysis of traffic flows is not required in order to specify performance of communication networks.

Communication networks are examples of Discrete Event Dynamic Systems (DEDS), i.e., systems whose state transitions are triggered by events that occur at discrete instants. For example, the number of packets inside a buffer is a state variable that changes at the occurrence of events like “arrival” and “departure” of packets. DEDS are not appropriately described by the mathematical tools centered on differential equations. For such systems, there are alternative techniques, as presented in [13]–[14]. Among these techniques, the one based on the dioid algebra (or Max-plus algebra) allows the linear description of DEDS where synchronization phenomena predominate.

Chang [15] showed that the determination of performance bounds of communication networks is simplified if their mathematical constraints are represented using the dioid algebra. Anne Bouillard and Eric Thierry [16] addressed the issue of how to efficiently implement the NC operations. The goal of this article is to show that the mathematical constraints of communication networks can be adequately modeled using dioids of formal power series. We are motivated by the fact that the manipulation of these series has already been implemented, e.g., by Gaubert [17] and Hardouin and colleagues [18].

As a practical result, we analyze a FIFO (First In, First Out) multiplexer. The results obtained for this particular system represent an extension of previous results.

The article is divided as follows. Section II presents some important definitions and results on the dioid algebra. Section III introduces the Network Calculus. Section IV presents the proposed dioid. Section V defines a set of constraints that are useful to represent performance bounds in communication networks and shows that the proposed dioid is able to represent these constraints. Finally, Section VI illustrates the adequacy of the proposed dioid, by the analysis of a FIFO multiplexer.

II. ALGEBRAIC BACKGROUND

In this section, we present definitions and results of the dioid algebra that are necessary to understand the further analysis.

Definition 1 (Dioid and complete dioid [13]): A dioid $\mathcal{D}$ is a set endowed with two closed operations called “addition” ($\oplus$) and “multiplication” ($\otimes$). The usual notation is $(\mathcal{D}, \oplus, \otimes)$. The operations of a dioid must satisfy the following axioms: associativity of addition and multiplication, commutativity of addition, distributivity of multiplication over finite sums and idempotency (i.e., $\forall a \in \mathcal{D}$, $a \oplus a = a$). Besides, there must exist neutral elements for both operations, where $1$ is the unity element and $\epsilon$ is the null element, that must be absorbing (i.e., $\forall a \in \mathcal{D}$, $\epsilon \otimes a = a \otimes \epsilon = \epsilon$). If the dioid is closed for infinite sums and distributivity applies for infinite sums, the dioid is called complete. Table I illustrates the axioms of a dioid.

---

M. D. Cavalcante and R. S. Mendes are with the Department of Computer Engineering and Industrial Automation (DCA), School of Electrical and Computer Engineering (FEEC), State University of Campinas (UNICAMP), Campinas-SP, Brazil (DCA-FEEC-UNICAMP). C. P. 6101. Campinas-SP, Brazil. CEP 13083-970. {madaca,rafael}@dca.fee.unicamp.br
TABLE I
AXIOMS OF A DIOID
\[ \forall a, b, c \in D: \]
\[
\begin{align*}
a \oplus b &= b \oplus a \\
(a \oplus b) \oplus c &= a \oplus (b \oplus c) \\
(a \otimes b) \otimes c &= a \otimes (b \oplus c) \\
\varepsilon \otimes a &= a \otimes \varepsilon = \varepsilon
\end{align*}
\]
\[
\varepsilon \otimes a = a \otimes \varepsilon = a
\]

Dioids can be classified depending on their characteristics. For example, a dioid is said “commutative” if its multiplication is commutative, i.e., \( \forall a, b \in D: a \oplus b = b \oplus a \).

Example 1 (Min-plus dioid): Consider \( R = \mathbb{R}^+ \cup \{+\infty\} \). Given \( a, b \in R \). Define the following operations:
\[
a \oplus b = \min \{a, b\},
a \otimes b = a + b.
\]
For example: \( 2 \oplus 3.5 = 2; 2 \otimes 3.5 = 5.5; \varepsilon = +\infty \); and \( e = 0 \). It is simple to check that \( R = (R, \min, +) \) is a complete commutative dioid.

Definition 2 (Subdioid [13]): A subset \( D' \) of a dioid is called a subdioid of \( D = (\mathcal{D}, \oplus, \otimes) \) if \( \varepsilon, e \in D' \) and \( D' \) is closed for \( \oplus \) and \( \otimes \).

Remark 1 (Order in a complete dioid [13]): Let \( a, b \in D \) be two elements of a complete dioid \( D \). Define \( a \wedge b \in D \) as the greatest element of \( D \) that is smaller than or equal to \( a \) and \( b \). A partial order in \( D \) can be defined as follows:
\[
a \succ b \iff a = a \oplus b \land b = a \wedge b.
\]
In particular, define \( \top \) as the sum of all elements in \( D \). From the above equivalences, \( \top \) is the maximum element in \( D \).

Definition 3 (Residuation function and residual [13]): Let \( f : D \to D \) be an isotonous mapping from a complete dioid \( D \) into a complete dioid \( D' \). If, \( \forall b \in D' \), the set of solutions of \( f(x) \leq b \) is non-empty and has a maximum element, denoted \( f^\#(b) \) (unique), then \( f \) is called a residuated function and \( f^\#(b) \) is called the residual of \( f \) at \( b \).

Define functions \( l_a(x) = a \otimes x \) and \( r_a(x) = x \otimes a \). It can be proved that, for any dioid, \( l_a(x) \) and \( r_a(x) \) are residuated. The residual functions are denoted, respectively, \( l_a^\#(y) = a \land y \) and \( r_a^\#(y) = y \lor a \). Table II presents some useful properties of the residuation operation.

Theorem 1: (Theorem 4.75 in [13]) Given \( a, b \in D \) (complete), the minimum solution of equation \( x = a \oplus b \) is given by \( x = a^* \ominus b \), where \( a^* = \bigoplus_{i \in \mathbb{N}} a_i^* \), \( a^0 = e \), and \( a^i = a \oplus \ldots \oplus a \) (i times) (Kleene closure). Moreover, any solution of \( x = a \oplus b \) satisfies \( x = a^* \ominus x \).

Consider the following equivalence relation (i.e., reflexive, symmetric and transitive).

Definition 4 (mod z-equivalence): Given \( a, b, z \in D \), where \( D \) is a commutative dioid, \( a \) and \( b \) are mod \( z \)-equivalent (notation \( a \equiv b \mod z \)), if \( az^+ = bz^+ \).

It is possible to properly define \( \oplus \) and \( \otimes \) operations between the equivalence classes defined from this relation. The result is a new dioid, called quotient dioid, denoted \( D/z \), whose elements are classes of \( D \). Usually, a class is represented by its greatest element.

Definition 5 (Formal power series in \( \delta [13] \)): A formal power series with coefficients in a dioid \( D \) is a mapping \( f \) from \( \mathbb{N} \) (or \( Z \)) into \( D \): \( \forall t \in \mathbb{N} \) (or \( Z \)), \( f(t) \) represents the coefficient of \( \delta^t \), i.e.,
\[
f(t) = \bigoplus_{r \in \mathbb{N} \lor \mathbb{Z}} [f(r) \delta^r].
\]  
(1)

Remark 2: Not every term in \( (1) \) must be explicitly written. The missing terms have coefficient \( \varepsilon \).

Definition 6 (Support and degree of a series [13]): The support \( \text{supp}(f) \) of a series \( f \) is defined as \( \text{supp}(f) = \{ t \in \mathbb{N} : f(t) \neq \varepsilon \} \). The degree of \( f \) is the maximum element of \( \text{supp}(f) \), i.e., \( \deg(f) = \max \{ \text{supp}(f) \} \).

Definition 7 (Monomial and polynomial [13]): A monomial is a series with a support reduced to a singleton; a polynomial is a series with a finite support.

The set of formal series is endowed with the following operations:
\[
f \oplus g : \ (f \oplus g)(t) = f(t) \oplus g(t),
\]
\[
f \otimes g : \ (f \otimes g)(t) = \bigoplus_{s \in \mathbb{Z}} [f(s) \otimes g(t - s)].
\]
(2)

(3)
The set of series endowed with these two operations is a dioid denoted \( D[\delta] \), whose zero element \( \varepsilon \) is defined by \( f(k) = \varepsilon, \forall k \), and whose unity element \( e \) is defined by \( f(0) = e \) and \( f(k) = e \), otherwise [13].

Given \( f, g \in D[\delta] \), the following order is defined:
\[
f \succ g \iff f(t) \succ g(t), \forall t \in \mathbb{N}.
\]

Remark 3: If lower bounds can be defined in \( D \), in particular when \( D \) is complete, these lower bounds extend to \( D[\delta] \) “coefficientwise”.

Given \( f, h \in D[\delta] \), from Definition 3, we have [13]:
\[
g = f \otimes h = \bigoplus_{t \in \mathbb{Z}} \bigwedge_{s \in \mathbb{Z}} [f(s) \otimes h(t + s)] \delta^t.
\]
(4)

Example 2: Consider \( f = f(m) \delta^m \) (i.e., a monomial). From (4): \( f \otimes h = \bigoplus_{t \in \mathbb{Z}} [f(m) \otimes h(t)] \delta^{t-m} \).

III. NETWORK CALCULUS BACKGROUND

Network Calculus is a set of rules and results regarding performance parameters of communication networks. NC is based on the idea that a detailed analysis of traffic flows

\[
\mathcal{D}/z,
\]
where \( \mathcal{D} \) is a set of rules and results regarding performance parameters of communication networks. NC is based on the idea that a detailed analysis of traffic flows
is not required in order to specify bounds for a network performance in terms of service requirements. As usual in NC theory, the following conditions must be satisfied:

- The input flow must be limited;
- Some service guarantee must be provided.

The above conditions define a minimum system to the NC: a filter that limits the input traffic; and a packet network that offers minimum service guarantees. Fig. 1 illustrates this system. These conditions are directly related to the regulation and service curves concepts presented in Definitions 8 and 9.

As Cruz [19], we consider a discrete time model, where a discrete function \( x(t) : \mathbb{Z} \rightarrow \mathbb{R} \) (Example 1) denotes the number of packets that arrive at (or leave) a given network element during the interval \([0, t]\). Thus, \( x(t) \) must be a non-decreasing function of \( t \). Besides, systems are considered empty at \( t = 0 \). Therefore, \( \forall t < 0: x(t) = 0 \). Define \( \mathcal{F} = \{ x : x(t_1) \geq x(t_2) \} \), if \( t_1 \geq t_2 \) and \( x(t) = 0, \forall t < 0 \). From the previous discussion, \( x \in \mathcal{F} \).

**Definition 8 (Regulation curve [5]):** Consider functions \( y, \sigma \in \mathcal{F} \). Function \( \sigma \) is a regulation curve for flow \( y \), if \( y \) is such that, \( \forall s, t \in \mathbb{Z} \) with \( s \leq t \):

\[
y(t) - y(s) \leq \sigma(t - s).
\]

(5)

In general, the regulation curves are referred to in the NC literature as “arrival curves” or “departure curves”, depending on if \( y \) in (5) represents the input flow or the output flow of a network. In this work, we use the term “regulation” curve, to emphasize that both (arrival and departure curves) have the same meaning of burst limitation.

The most common regulation curve in the literature is \( \sigma(t) = b + t \cdot \forall t > 0 \), and \( \sigma(t) = 0 \), otherwise. A filter that satisfies this \( \sigma \) is called a leaky bucket (or token bucket).

**Definition 9 (Service curve [9], [20]):** Given functions \( x, y, \beta \in \mathcal{F} \), function \( \beta \) is a service curve between flows \( x \) and \( y \) if, \( \forall t \in \mathbb{Z} \), \( \exists s \in \mathbb{Z} \) such that \( s \leq t \) and the following inequality is satisfied:

\[
y(t) \geq x(s) + \beta(t - s).
\]

(6)

Define \( x^+ = \max \{0, x\} \). The most common service curve in the literature is called “rate latency” curve. It is defined as follows, \( \forall t \in \mathbb{Z} \): \( \beta(t) = R \cdot (t - T)^+ \).

Equations (5) and (6) can also be written by the following way, \( \forall t \in \mathbb{Z} \):

\[
y(t) \leq \min_{0 \leq s \leq t} \left\{ y(s) + \sigma(t - s) \right\},
\]

(7)

\[
y(t) \geq \min_{0 \leq s \leq t} \left\{ x(s) + \beta(t - s) \right\}.
\]

(8)

Given a system with input flow \( x \) and output flow \( y \), define:
1) the backlog of the system at instant \( t \) is the vertical deviation between \( x(t) \) and \( y(t) \), i.e., \( x(t) - y(t) \); 2) the virtual delay is the horizontal deviation between \( x(t) \) and \( y(t) \), i.e.,

\[
d(t) = \inf \{ d \geq 0 : x(t) \leq y(t + d) \}.
\]

(9)

From the concepts of regulation and service curves, it is possible to obtain bounds for the maximum delay and backlog between \( x \) and \( y \), respectively. \( D \) and \( W \), and a regulation curve \( \sigma_y \) for the output traffic. From [9], we have:

\[
D = \inf \{ d \geq 0 : \sigma^y(t) \leq \beta(t + d), \forall t \},
\]

\[
W = \sup_{s \geq 0} \{ \sigma^y(s) - \beta(s) \},
\]

\[
\sigma_y(t) = \sup_{s \geq 0} \{ \sigma^y(t + s) - \beta(s) \}. \tag{10}
\]

### IV. PROPOSED DIOID

The results previously mentioned show that the \( \inf \) and \( \sup \) operations appear regularly in the Network Calculus. This suggests the utilization of the dioid theory to represent NC constraints. In this section, we propose a dioid denoted \( [\mathcal{R}] [\delta] \) for the treatment of NC problems.

Consider dioid \( [\mathcal{R}] [\delta] \) presented in Example 1, where \( \mathcal{R} = [\mathbb{R}]^+ \cup \{ -\infty \} \). The residuations \( a \preceq y, y \not\preceq a \in \mathcal{R} \) are given by \( a \preceq y = y \not\preceq a = [y - a]^+ \).

To prove this, consider inequality \( x \otimes a \preceq y \). It implies \( x + a \geq y \) or equivalently \( x \geq y - a \). But \( x \) is an element of \( \mathcal{R} \) and then \( x \geq 0 \). From both inequalities \( x \geq [y - a]^+ \). Thus, \( y \not\preceq a = [y - a]^+ \). Finally, as \( \otimes \) is commutative in \( [\mathcal{R}] [\delta] \), \( y \not\preceq a = a \preceq y \). Given that both residuations lead to the same result, define \( y \otimes a = y \not\preceq a = a \preceq y \).

The elements manipulated in NC are functions of the type \( x : \mathbb{Z} \rightarrow \mathcal{R} \), where \( x(t) \) represents the quantity of data that arrives at or departs from a network element in the time interval \([0, t]\). Any mapping \( x \) from \( \mathbb{Z} \) into \( \mathcal{R} \) can be associated to an element of dioid \( \mathcal{R}[\delta] \) by the following way:

\[
x = \bigoplus_{t \in \mathbb{Z}} x(t) \delta^t.
\]

In dioid \( \mathcal{R}[\delta] \), from (2)-(3) and the definition of \( \mathcal{R} \), we have:

\[
f \otimes g : \quad (f \otimes g)(t) = \min \{ f(t), g(t) \}, \tag{11}
\]

\[
f \otimes g : \quad (f \otimes g)(t) = \inf_{s \in \mathbb{Z}} \{ f(s) + g(t - s) \}. \tag{12}
\]

**Remark 4:** From Remark 1 and (11), symbols \( \preceq \) and \( \preceq \) that represent the partial order in \( \mathcal{R}[\delta] \) correspond, respectively, to \( \leq \) (\( \forall t \)) and \( \geq \) (\( \forall t \)), in the usual algebra (notice the inversion). Table III presents this result.

**Proposition 2:** In \( \mathcal{R}[\delta] \), we have:

\[
y \not\preceq a = a \preceq y : \quad (y \otimes a)(t) = \sup_{s \in \mathbb{Z}} \{ y(t + s) - a(s) \}^+.
\]

**Proof:** This proposition is a direct consequence of the definition of \( \mathcal{R} \) and (4).

| Table III
<table>
<thead>
<tr>
<th>Equivalent constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \forall t \geq 0: y(t) \leq x(t) )</td>
</tr>
<tr>
<td>( y = y \otimes x )</td>
</tr>
</tbody>
</table>


Example 3: Consider $a, b \in \mathcal{R}[\delta]$ defined as follows: $a = 5\delta^2 \oplus 7\delta^7$ and $b = 2\delta$. The following equalities are satisfied:

$$a \oplus b = 2\delta \oplus 5\delta^2 \odot 7\delta^7,$$

$$a \odot b = (5\delta^2)\odot^2 \oplus (7\delta^7)\odot^2 + 7\delta^3 \oplus 9\delta^8 = b \odot a,$$

$$b \ltimes a = (2\delta \ltimes 5\delta^2) \oplus (2\delta \ltimes 7\delta^7) = 3\delta \oplus 5\delta^6 = a \not\ltimes b.$$

From dioid $\mathcal{R}[\delta]$, it is possible to define dioid $\mathcal{R}[\delta]/\delta^{-1}$. In this new dioid, we have: $\forall a \in \mathcal{R}[\delta]/\delta^{-1}, a = a(\delta^{-1})^\ast$. The elements of $\mathcal{R}[\delta]/\delta^{-1}$ are classes of power series, whose representative elements (i.e., the greatest element of each class) have non-decreasing coefficients in $t$. In $\mathcal{R}[\delta]/\delta^{-1}$, we have:

$$\varepsilon = \bigoplus_{t \leq 0} (+\infty)\delta^t,$$

$$e = \bigoplus_{t > 0} 0\delta^t \bigoplus_{t \geq 0} (+\infty)\delta^t = (\delta^{-1})^\ast.$$  \hspace{1cm} (13)  

Now, consider the subset $\mathcal{C}$ of $\mathcal{R}[\delta]/\delta^{-1}$ whose elements satisfy the condition $x(t) = x(0), \forall t < 0$.

**Proposition 3:** $\mathcal{C}$ is a subdioid of $\mathcal{R}[\delta]/\delta^{-1}$.

**Proof:** From Definition 2, $\mathcal{C}$ is a subdioid of $\mathcal{R}[\delta]/\delta^{-1}$ if it contains $\varepsilon$ and $e$ and it is closed to $\oplus$ and $\otimes$. From (13) and (14), it is clear that $\varepsilon, e \in \mathcal{C}$. Consider $f, g \in \mathcal{C}$, i.e., $f$ and $g$ have non-decreasing coefficients in $t$ and $\forall t < 0$:

$$f(t) = f(0) \text{ and } g(t) = g(0).$$

For all $t < 0$:

$$(f \oplus g)(t) = \min\{f(t), g(t)\},$$

$$(f \odot g)(t) = \min\{f(0), g(0)\} = (f \odot g)(0),$$

$$(f \otimes g)(t) = \min\\{\inf_{s < 0} \{f(s) + g(t - s)\}; \inf_{s \geq 0} \{f(s) + g(t - s)\}\} = \min\{f(0) + \inf_{s < 0} \{g(t - s)\}; \inf_{s \geq 0} \{f(s)\} + g(0)\} = f(0) + g(0) = (f \otimes g)(0).$$

Thus, the $\oplus$ and $\otimes$ operations defined in (11) and (12) are closed in $\mathcal{C}$. $\blacksquare$

Consider the multiplication in $\mathcal{C}$. We have, $\forall t \geq 0$:

$$(f \odot g)(t) = \inf_{s \geq 0} \{f(s) + g(t - s)\},$$

$$= \min\{\inf_{s < 0} \{f(s) + g(t - s)\}; \inf_{s \geq 0} \{f(s) + g(t - s)\}\} = \min\{f(0) + \inf_{s < 0} \{g(t - s)\}; \inf_{s \geq 0} \{f(s)\} + g(0)\}.$$

But $f(t)$ and $g(t)$ are non-decreasing in $t$. Then,

$$\inf_{s \geq 0} \{g(t - s)\} = g(t + 1) \geq g(t),$$

$$\inf_{s > t} \{f(s)\} = f(t + 1) \geq f(t).$$

Therefore, from (15), $\forall t \geq 0$:

$$(f \odot g)(t) = \min_{0 \leq s \leq t} \{f(s) + g(t - s)\}. \hspace{1cm} (16)$$

By a similar reasoning, from Proposition 2, the residuations in $\mathcal{C}$ are given by:

$$y \not\ltimes a = a \not\ltimes y : (y \circ a)(t) = \sup_{s \geq 0} \{y(t + s) - a(s)\}^+ \hspace{1cm} (17)$$

Notice that the multiplication in (16) is exactly the same operation that appears in (7)-(8). Therefore, these expressions can be represented by the multiplication of two power series in $\mathcal{C}$. The residuations in (17) are not exactly the same operation that appears in (10). Actually, the expressions in (10) can lead to results that are not in $R = R^+ \cup \{+\infty\}^1$. This inconvenient is satisfactorily overcome by the residuations in $\mathcal{C}$.

In the following, we present elements and mappings in $\mathcal{C}$ that are useful to represent some common constraints of communication networks.

**Definition 10 (Vertical displacement function):** For a given $B \in R$, define the vertical displacement function, $\lambda_B \in \mathcal{C}$, as follows: $\lambda_B = B = B \otimes (\delta^{-1})^\ast$.

Function $\lambda_B$ satisfies the following equality:

$$x \otimes \lambda_B = \bigoplus_{t \in N} x'(t)\delta^t,$$

where $x'(t) = x(t) + B, \forall t$.

**Example 4:** The leaky bucket regulation curve, i.e., $\delta(t) = b + \tau \cdot t$ if $t > 0$ and $\delta(t) = 0$ otherwise, corresponds to the following element in $\mathcal{C}$:

$$\sigma = b \otimes (r\delta)^\ast.$$  \hspace{1cm} (18)

**Definition 11 (Delay Function):** For a given $D \in R$, define the delay function $\delta_D \in \mathcal{C}$ as follows: $\delta_D = \delta^D$.

Function $\delta_D$ satisfies the following equalities:

$$x \otimes \delta_D = \bigoplus_{t \in Z} x(t)\delta^t + D, \hspace{1cm} (19)$$

$$x \otimes \delta_D = \bigoplus_{t \in Z} x(t)\delta_D - D. \hspace{1cm} (20)$$

**Example 5:** From (18), the rate latency curve $\beta = R \cdot (t - T)^+$ can be represented as $\delta_T \otimes (R\delta)^\ast$.

Finally, mappings $F$ and $G$ below are useful to represent systems with multiplexing.

**Definition 12 (F and G mappings):** Given $a \in \mathcal{C}$ and $b \in \mathcal{C}$, mappings $F$ and $G$ are:

$$F\{a, b\} : (F\{a, b\})(t) = a(t) + b(t), \hspace{1cm} (19)$$

$$G\{a, b\} : (G\{a, b\})(t) = [a(t) - b(t)]^+. \hspace{1cm} (20)$$

**V. NETWORK MODELING**

NC concerns the determination of QoS parameters such as end-to-end delay, maximum backlog and maximum throughput. Mathematical constraints can be associated to these parameters, expressing performance specifications as well as physical limitations of systems. Further, we describe five types of constraints [21]-[24]: flux constraints, coupling constraints, regulation constraints, service constraints and...
backlog constraints. We do not claim that this set of constraints is complete in the sense that it can solve all network problems. However, it covers a large set of fundamental problems related to QoS parameters and for this reason we call it basic set of constraints. In the following, we show that dioid $\mathcal{D}$ is able to represent all the basic constraints.

The flux constraint guarantees that the quantity of data that leaves a system, $y(t)$, is never greater than the one that enters in it, $x(t)$. Mathematically, $y(t) \leq x(t)$, $\forall t$. From the definition of $\mathcal{D}$ and Table III, this inequality can be written in $\mathcal{D}$ by the following way: $y = y \oplus x$.

The coupling constraint allows the modeling of systems whose output traffic is the coupling (multiplexing) of its input traffics. Let $x_k$, $k = 1, \ldots, M$, be the input flows and $y$ be the output flow of this system. Mathematically, $y(t) = x_1(t) + x_2(t) + \ldots + x_M(t)$, or in $\mathcal{D}$, from (19):

$$y = F \left\{ x_1, \ldots, x_M \right\} = F_{k=1,\ldots,M} \{ x_k \}. \quad (21)$$

The regulation constraint shapes the output flow by limiting it. Mathematically, (7) must be satisfied, i.e., $y(t) \leq \min_{0 \leq s \leq t} \{ y(s) + \sigma(t-s) \}$. In $\mathcal{D}$, from (16) and Table III, this inequality can be written as

$$y = y \oplus y \otimes \sigma. \quad (22)$$

The service constraint is related to the minimum quantity of data served by a network element. Mathematically, (8) must be satisfied. Or in $\mathcal{D}$, from (16) and Table III:

$$x \otimes \beta = x \otimes \beta \oplus y. \quad (23)$$

**Example 6 (Delay constraints):** From (9), a system has maximum virtual delay limited by $D$ when, $\forall t \in \mathbb{Z}$: $x(t) \leq y(t + D)$. Equivalently, $\forall s \in \mathbb{Z}$: $x(s - D) \leq y(s)$. Now, from (18), $x(s - D) = x \otimes \delta_D$. Thus, from Table III, a system has maximum virtual delay $D$ if the following equation is satisfied in $\mathcal{D}$: $x \otimes \delta_D = x \otimes \delta_D \oplus y$. Comparing this equation with (23), we can see that delay constraints can be represented by a service curve $\beta = \delta_D$.

Finally, the backlog constraint represents limitations on buffer capacity. We say that a given queue has maximum backlog capacity limited by $B \geq 0$, if the following inequality is satisfied, $\forall t$: $x(t) - y(t) \leq B$. From Definition 10, this inequality can be rewritten in $\mathcal{D}$ by the following way:

$$x = x \oplus y \oplus \lambda_B. \quad (24)$$

For practical purposes, we can define pictorial elements based on the mathematical constraints described before (the flux constraint is always present for causal systems): the $M$-coupler, the $\sigma$-regulator, the $\beta$-network and the $B$-queue. These definitions allow the modular description of more complex network systems, as will be exemplified in Section VI. Table IV presents the equations that characterize each basic element. We consider that the basic elements have infinite buffer capacity, so that the results obtained by this methodology represent a limit when losses are avoided.

**Remark 5:** The arrows on the graphical representations of the basic elements indicate the direction of the flux constraints and not necessarily the physical path of packets, as presented in [21].

<table>
<thead>
<tr>
<th>Element</th>
<th>Representation</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M$-coupler</td>
<td>$x_1 \oplus y$</td>
<td>$y = F_{k=1,\ldots,M} { x_k }$</td>
</tr>
<tr>
<td>$\sigma$-regulator</td>
<td>$x \oplus y$</td>
<td>$y = y \oplus x \quad y = y \oplus y \sim \sigma$</td>
</tr>
<tr>
<td>$\beta$-network</td>
<td>$x \oplus \beta$</td>
<td>$y = x \oplus \beta \oplus y$</td>
</tr>
<tr>
<td>$B$-queue</td>
<td>$x \oplus y \lambda_B$</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2. FIFO multiplexer with $M$ input flows. (a), physical description. In (b), model using basic elements.

**VI. EXAMPLE**

In this section, we illustrate the adequacy of dioid $\mathcal{D}$ and the basic elements previously defined for the treatment of NC problems by the analysis of a FIFO multiplexer.

Consider the FIFO multiplexer with $M$ input flows shown in Fig. 2. Each input flow $x_k$, $k = 1, \ldots, M$, satisfies a regulation curve $\sigma_k$, $k = 1, \ldots, M$. Moreover, the system offers a service curve $\beta$ to the aggregate of flows. We want to determine a service curve $\beta_i$ that this system can offer to the input flow $x_i$.

Fig. 2(b) illustrates a model for this system using the basic elements introduced in the previous section. Define $x_i$ as the flow of interest and $x_{(i)}$ as the aggregate of the other flows excluding $x_i$, i.e., from (21): $x_{(i)} = F_{k \neq i} \{ x_k \}$. If flow $x_k$ satisfies a regulation curve $\sigma_k$, from (22), $x_k = x_k \oplus x_k \sigma_k$. Then, from Theorem 1: $x_i = x_i \sigma_{\beta_i}^*$. Consequently:

$$x_{(i)} = F_{k \neq i} \{ x_k \} = F_{k \neq i} \{ x_k \sigma_k^* \}. \quad (24)$$

It is easy to check that, $\forall a, b \in \mathcal{D}$, $F_{k \neq i} \{ a_k b_k \} \supseteq F_{k \neq i} \{ a_k \} \otimes F_{k \neq i} \{ b_k \}$. Therefore, from (24), $x_{(i)} \supseteq F_{k \neq i} \{ x_k \} \otimes F_{k \neq i} \{ \sigma_k^* \} = x_{(i)} \otimes F_{k \neq i} \{ \sigma_k^* \}$. Define $\sigma_{(i)} = F_{k \neq i} \{ \sigma_k^* \}$. Thus, $x_{(i)} = x_{(i)} \otimes x_{(i)} \sigma_{(i)}$ and $\sigma_{(i)}$ is a regulation curve for the aggregate flow $x_{(i)}$.

From the previous discussion, considering $x_i$, the flow of interest, the multiplexer with $M$ input flows can be viewed as a multiplexer with two input flows: $x_i$ and $x_{(i)}$. In [25], Cruz presented the service curve of a traffic in a multiplexer with two inputs. This result is presented next.

**Theorem 4 (Theorem 4 in [25]):** Consider a FIFO multiplexer with two input flows and a service curve $\beta$. Define $x_k$ and $y_k$, respectively, the input and output flows of traffic $k$,
\[ \beta = (R\delta)^* \]
\[ \sigma (i)\delta \Theta = [(r_2\delta)^* \oplus b(r_1\delta)^*] \delta \Theta \]
\[ \beta_0 \]

Finally, we remark that many results presented in [21]–[24] can be implemented with the formalism herein presented, and can be used as a service curve.

Fig. 3. Curves \( \beta, \sigma (i)\delta \Theta \), \( \beta_0 \) and \( \beta_i \) for a FIFO multiplexer with \( M \) input traffics and traffic of interest \( i \). For these curves, \( R = 100, b = 600, r_1 = 10, r_2 = 200 \) and \( \Theta = 10 \).

\[ k = 1, 2. \] If \( x_2 \) satisfies a regulation curve \( \sigma_2 \) and, for a given \( \Theta \in \mathbb{N} \), if \( \beta_0^-(t) = [\beta(t) - \sigma_2^{-}(t - \Theta)]^+ \) is a non-decreasing function of \( t \), then the multiplexer offers a service curve \( \beta_0^+(t) \) to flow 1.

From (18) and (20), Theorem 4 can be extended to a FIFO multiplexer with \( M \) inputs. In this case, the \( M \)-multiplexer offers to flow \( i \) the following service curve in \( \mathcal{C} \):

\[ \beta_i = G \{\beta, \sigma (i)\delta \Theta\} . \]

(25)

Unlike Theorem 4, (25) defines a service curve \( \beta_i \) for any \( \Theta \in \mathbb{N} \). Thus, (25) is an extension of Theorem 4. Fig. 3 presents an example where \( \beta_0^+(t) \) is not defined for a given \( \Theta \) (notice the decreasing portion), but \( \beta_i \in \mathcal{C} \) is well defined and can be used as a service curve.

VII. CONCLUSION

In this article, we defined a dioid of formal power series for dealing with NC problems. To test the adequacy of the proposed dioid, we defined a basic set of mathematical constraints that can represent a large set of fundamental problems concerning the determination of performance bounds of communication networks. The proposed dioid is able to represent each of these constraints. As an example, we analyzed a FIFO multiplexer. The results obtained for this particular system represent an extension of previous ones. Finally, we remark that many results presented in [21]–[24] can be implemented with the formalism herein presented, taking advantage of existing tools for the manipulation of power series dioids.

REFERENCES


