Petri Net Modeling of Irrigation Canal Networks

Giorgio Corriga, Alessandro Giua, Giampaolo Usai

DIEE: Dip. di Ingegneria Elettrica ed Elettronica — Università di Cagliari

P.zza d'Armi – 09123 CAGLIARI, Italy

Email: giua@diee.unica.it — Fax: +39 (70) 675-5900

Abstract

The paper describes a formal procedure to construct a Petri net model starting from the differential equations that describe the behavior of a canal network. In a first approach, the equations are discretized and the corresponding model takes the form of a deterministic timed Petri net. In a second approach, timed continuous Petri nets are used. The dynamic behavior of the system can be studied as sequences of reachable markings of the net and can be computed with standard Petri net execution techniques.

1 Introduction

Canal networks equipped with automatic gates which maintain constant upstream levels are commonly used for open-channel irrigation [4, 5]. In this paper we show how to model such systems with a discrete event formalism based on Petri nets.

Petri nets [10, 13] are a well known discrete event model of great expressive power and rich of analytical tools. Simulation software for Petri nets is also readily available. Recently, Petri nets have also been used to describe systems with continuous dynamics such as hybrid [9], batch [2], and qualitative systems [8].

We are currently investigating the modeling of canal networks with different Petri net models, ranging from timed nets to continuous nets. The topology of the canal network can be readily modeled by paths in the net. The time lags associated to the propagation can be explicitly modeled associating time delays to the transitions. The flow of tokens will correspond to the flow of water. External control inputs are modeled using transitions whose firing rate may change with time.

At the present stage, we have only considered modeling issues. The development of such a model may be useful for several reasons.

- Standard Petri net software tools may be used to perform simulations.
- It may be possible to adapt some of the Petri nets analysis techniques to the study of properties of interest for these systems.
- Many control problems that have been studied in the Petri net framework, bear a close resemblance to problems of managing irrigation canal networks. As an example, the problem discussed in [3] of

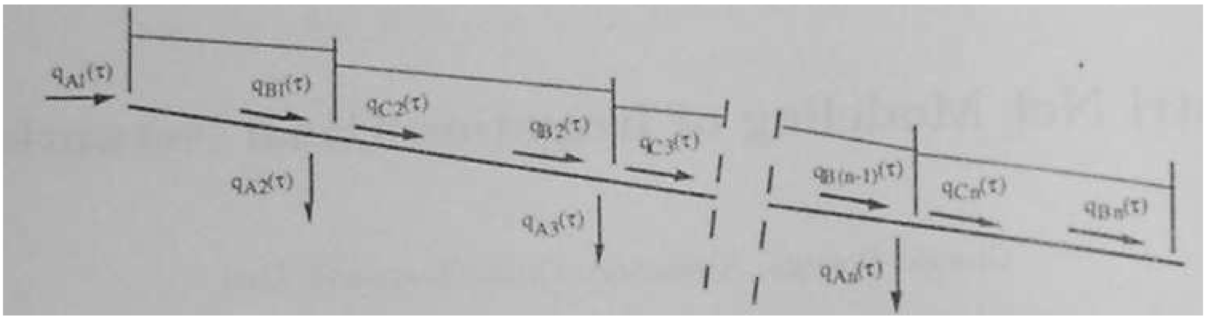

Figure 1: A canal network with n cascaded reaches.

deriving appropriate control laws to coordinate the users' withdrawal of water with relative upstream immission is a resource allocation (scheduling) problem. A large body of results exists in the Petri net literature on this kind of problems.

The paper is structured as follows. In Section 2 we recall the mathematical model of a canal network with self-levelling gates. In Section 3 we present background material on Petri nets. In Section 4 we present a model based on discrete timed Petri nets. In Section 5 we present a model based on timed continuous Petri nets.

2 Model for canal networks with self-levelling gates

Self-levelling control gates are often used as regulating organs in irrigation networks: they allow to keep a preset level upstream for wide ranges of rates of flow, thus enabling to deliver the required water amounts to the utilizers, the delivering sections being usually just upstream from the gates. A water distribution scheme with self-levelling control gates is essentially a system of the supply type, since the water delivery to the utilizers depends mainly on the water feeding policies at the upstream end of the network: it is indeed impossible to act over the control gates throughout the canals. A deeper knowledge of the network dynamic behaviour (rate of flow variations and delays at each gate due to water feeding changes) can help in achieving optimal operating policies, by rationalizing the water deliveries and thus avoiding water wastes, which always occur to some extent in such distribution systems.

The system we analyze, as shown in Figure 1, is made up of n cascaded canal reaches, separated by self-levelling gates with delivery points just upstream from the gates; water is fed only to the upstream end of the first reach.

As system inputs we take the variables which we can directly vary, i.e. the inflow rate $q_{A1}(\tau)$ at the upper end of the first reach and the rates of flow $q_{A1}(\tau)$, $(i = 2, \ldots n)$ delivered to the utilizers, whose amount can be regulated by proper operation of the outlet gates independently of the actual state of the system. In fact self-levelling gates just downstream the delivering points ensure a steady water level at each point, whatever be the rate of flow in the reach, within the allowed operating ranges of the gates. System outputs are the rates of flow $q_{Bi}(\tau)$, $(i = 1, 2, \ldots n)$ at the lower ends of each reach which, according to the lumped elements system model assumed, characterize the hydraulic behaviour of each reach.

Since the time variation of inputs can be assumed to be slow enough wave fronts are not taken into account and furthermore the evolution of the system between two steady state situations is supposed to be a succession of steady flow configurations, characterized by the rate of flow through each reach, by the related steady surface profile and by the water level set by the control gates at the lower end of each reach, which is supposed to stand still during the system evolution, if the dynamics of the gates are neglected.

For every steady flow situation, to each rate of flow through a reach corresponds a water volume stored in the same reach, which can be reckoned by putting into account the canal geometry, the steady surface profile, the hydraulic characteristics of the canal and the water level, set at the lower end of the reach by the control gate, as well.

Thus it is possible to determine for each reach a function

$$q_{Bi} = \varphi(\rho_i) \qquad (i = 1, \ldots, n) \tag{1}$$

which expresses the rate of flow q_{Bi} as a function of the stored volume ρ_i [4, 5].

Each reach is characterized also by a time delay D_i, which is the time required by a perturbation due to a rate of flow variation at the upper end of the reach to arrive at the lower end. Assuming subcritical flow, such a delay may be expressed as: $D_i = \dfrac{L_i}{V_{im} + C_{im}}$, where L_i is the reach length, and V_{im} and C_{im} are, respectively, the mean water velocity and the celerity of the infinitesimal perturbation in the middle cross section of the reach corresponding to the initial rate of flow. C_{im} is given by the formula $C_{im} = \left(g\dfrac{S_{im}}{T_{im}}\right)^{0.5}$ where S_{im} and T_{im} are the cross section area and the water surface width in the middle of the reach.

We can also write the continuity equation at the i-th node as:

$$\begin{cases} q_{C1}(\tau) = & q_{A1}(\tau) \\ q_{Ci}(\tau) = & q_{B(i-1)}(\tau) - q_{Ai}(\tau) \qquad (i = 2, \ldots, n) \end{cases} \tag{2}$$

Thanks to the assumed hypothesis about the hydraulic transients, the increment of the reach water volume $\dot{\rho}_i \, d\tau$ equals the elementary volume variation between two steady flow configurations infinitely close to each other, corresponding to the rates of flow $q_{Bi}(\tau + d\tau)$ and $q_{Bi}(\tau)$ for which equation (1) holds.

Therefore the system behaviour can be described by the following equations [4, 5]:

$$\begin{cases} \dot{\rho}_1 & = & q_{C1}(\tau - D_1) - q_{B1}(\tau) \\ \dot{\rho}_2 & = & q_{C2}(\tau - D_2) - q_{B2}(\tau) \\ \cdots & & \cdots \\ \dot{\rho}_n & = & q_{Cn}(\tau - D_n) - q_{Bn}(\tau) \end{cases} \tag{3}$$

In order to integrate system (3) we need to know, besides the initial water volumes stored in each reach, the input and output rates of flow for time intervals before the initial time of the same length as the time delays.

3 Petri nets

Many Petri net models have been defined in literature (see [10, 13]). Here we will only recall the basic elements that will be used in the next session to model canal networks.

3.1 Place/transition nets

A *place/transition net* [10, 11, 13] is a structure $N = (P, T, Pre, Post)$, where P is a set of *places* represented by circles; T is a set of *transitions* represented by bars; $Pre : P \times T \rightarrow I\!N$ is the *pre-incidence function* that specifies the arcs directed from places to transitions; $Post : P \times T \rightarrow I\!N$ is the *post-incidence function* that specifies the arcs directed from transitions to places.

A *marking* is a vector $M : P \rightarrow I\!N$ that assigns to each place of a P/T net a non-negative integer number of tokens, represented by black dots. $I\!N^{|P|}$ will denote the set of all possible markings that may be defined on the net. A *place/transition system* or *net system* $\langle N, M_0 \rangle$ is a net N with an initial marking M_0.

A transition $t \in T$ is *enabled* at a marking M iff $M \geq Pre(\cdot, t)$. If t is enabled at M, then t may fire yielding a new marking M' with $M' = M + Post(\cdot, t) - Pre(\cdot, t)$. We will write $M\ [t\rangle\ M'$ to denote that t may fire at M yielding M'. A *firing sequence* from M_0 is a (possibly empty) sequence of transitions $\sigma = t_1 \ldots t_k$ such that $M_0\ [t_1\rangle\ M_1\ [t_2\rangle\ M_2 \cdots\ [t_k\rangle\ M_k$. A marking M is *reachable* in $\langle N, M_0 \rangle$ iff there exists a firing sequence σ such that $M_0\ [\sigma\rangle\ M$. The set of markings reachable on a net N from a marking M is called *reachability set* of N and M and is denoted as $R(N, M)$.

The evolution of a Petri net starting from the initial marking can be studied using the *reachability graph*, where all reachable markings and the transition firings that lead from one marking to another are represented.

Example 1 Let us consider the Petri net in Figure 2.(a). The set of places is $P = \{p_1, p_2, p_3\}$, the set of transitions is $T = \{t_1, t_2, t_3\}$, the pre-incidence and post-incidence functions can be expressed as matrices

$$Pre = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \qquad Post = \begin{bmatrix} 0 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Note that the double arc from t_1 to p_2 has been represented, as usual, with a single arc of weight 2.

Representing the markings of this net as vectors $M = [M(p_1)\ M(p_2)\ M(p_3)]^T$, the initial marking, shown in the figure, is $M_0 = [2\ 0\ 0]^T$.

To construct the reachability graph we put a node labeled M_0 in the graph. Starting from M_0, both t_1 and t_2 are enabled. If t_1 fires we reach the marking $M' = [1\ 2\ 0]^T$. If t_2 fires we reach the marking $M'' = [1\ 1\ 1]^T$. Thus we add a node labeled M', a node labeled M'', an arc labeled t_1 from M_0 to M', and an arc labeled t_2 from M_0 to M''. We continue this construction to obtain the graph shown in Figure 2.(b).

The general algorithm to compute a reachability graph is given in [11]. Note that a reachability graph may be infinite. In this case it is still possible to construct a finite graph named *coverability graph* [11].

A final remark regards the notion of *conflict*. In the net in Figure 2.(a), when place p_1 is marked transitions t_1 and t_2 are enabled. The reachability graph shows all possible evolutions of the net, and it can be seen that from marking, say, $[2\ 0\ 0]^T$ two different markings may be reached depending on which transition fires. We may avoid this nondeterminism of the net by assuming that one transition has priority over the firing of the other one [1].

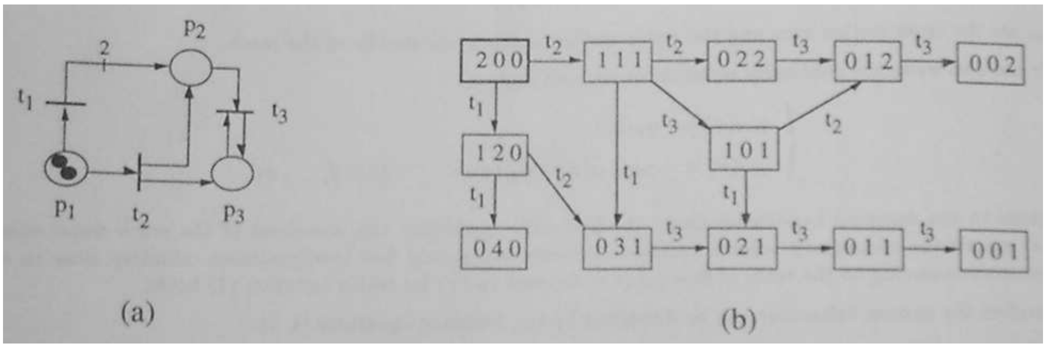

Figure 2: A Petri net (a), and its reachability graph (b).

3.2 Timed Petri nets

A simple extension of place/transition nets considers a function $d : T \rightarrow I\!R$ that associates to each transition $t_i \in T$ a *firing delay* d_i. Such a model is know as *deterministic timed Petri net* (DTPN). Note that timed transitions will be represented as black boxes as in Figure 3.

In a DTPN, each time a transition t_i becomes enabled a timer is set, and when the timer reaches the value d_i the transition is fired. We assume that the timer is reset to zero when t_i fires or when it is disabled (this policy is called *enabling memory* in [1]).

Note that the firing delay may be constant, may be a function of time τ (time dependent delay) or a function of the marking of some place of the net (marking dependent delay). A time dependent firing delay may be used to represent an external input that influences the net evolution.

In a DTPN a transition represents an operation that needs some time to be performed. It is possible to associate different semantics to this notion.

- *Infinite server:* Each transition represents an operation that can be performed by any of an infinite number of operating units that work in parallel. Thus in the net in Figure 3.(a), transition t_1 will fire 3 times at time d because the operating units may process all tokens at the same time.
- *Single server:* Each transition represents an operation that is performed by a single operating unit. Thus in the net in Figure 3.(b), transition t_1 will fire at times d, $2d$, and $3d$ because the operating unit processes one token at a time.
- *Multiple server:* Each transition represents an operation that is performed by a finite number k of operating units that can work in parallel. Thus in the net in Figure 3.(c), assuming $k = 2$, transition t_1 will fire twice at time d and once at time $2d$ because the operating units may processes up to 2 tokens at the same time.

Note that under the infinite server semantic, a token in p_1 will only remain in p_1 for a time d, while under the other semantics it may remain longer if all servers are busy processing other customers (i.e., tokens). We will explicitly represent the number of servers associated to a transition t by introducing a place p self-looped with t and containing k tokens as in Figure 3.(b) and Figure 3.(c). There will be no self-looped place for a transition with an infinite number of servers.

Note that a single server transition with delay d can fire with maximal firing rate $f = \frac{1}{d}$. The firing rate can be less than f if the transition is starved, i.e., there are not enough tokens to keep it always enabled.

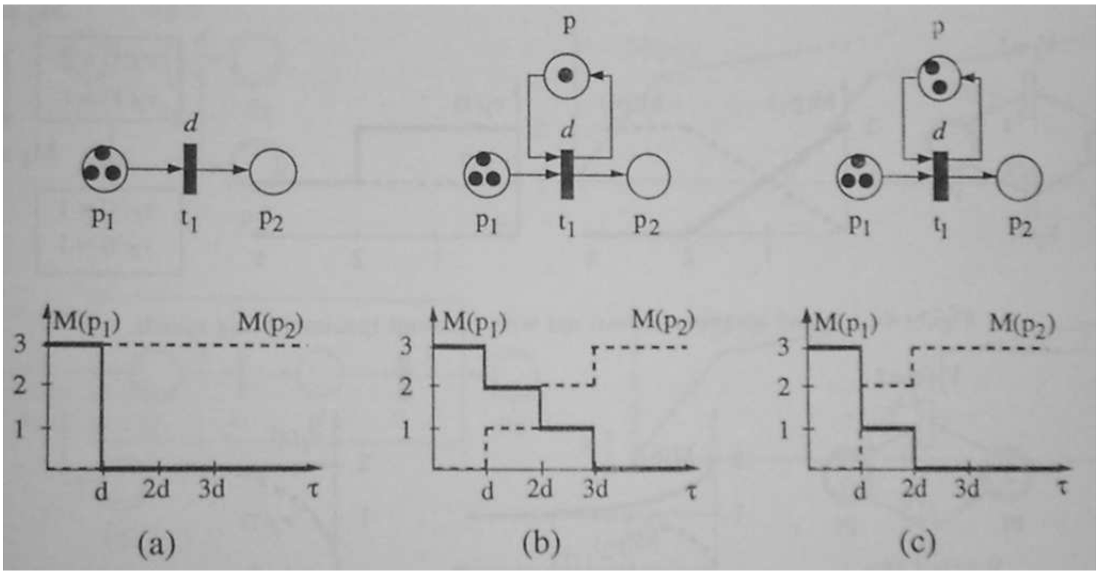

Figure 3: A deterministic timed Petri net. (a) Infinite server semantics. (b) Single-server semantics. (c) Two-server semantics.

The evolution of a timed net can still be studied using the reachability graph.

3.3 Continuous and hybrid Petri nets

The reachability set of a net with a great number of tokens may be very large. To avoid this problem, David and Alla [6, 7] have introduced *timed continuous Petri nets* (TCPN). In a TCPN the marking of a place (represented by a double circle as in Figure 4) and the *Pre* and *Post* functions are nonnegative real numbers. A *maximal firing speed* V_i will be associated to each continuous transition t_i (represented as a white box as in Figure 4), while its *instantaneous firing speed* will be denoted $v_i(\tau)$. During a interval $d\tau$ the continuous firing of transition t_i will remove $Pre(p, t_i)v_i(\tau)\ d\tau$ units of token (that we will call *marks*) from each input places p and add $Post(p, t_i)v_i(\tau)\ d\tau$ marks to each input places p.

The instantaneous firing speed $v_i(\tau)$ will be equal to V_i if all input places of t_i are marked, but may be less otherwise.

Example 2 Consider the net in Figure 4. Initially place p_1 is marked and $v_1(\tau) = V_1 = 2$. Thus, during a interval $d\tau$ transition t_1 will remove 2 $d\tau$ marks from p_1 and add 2 $d\tau$ marks to p_2. During the same time interval transition t_2 may fire at his maximal firing speed, i.e., $v_2(\tau) = V_2 = 1$, removing 1 $d\tau$ marks from p_2 and adding 1 $d\tau$ marks to p_1. Thus the marking evolution is initially ruled by the equations:

$$\frac{dM(p_1)}{d\tau} = v_2(\tau) - v_1(\tau) = -1$$
$$\frac{dM(p_2)}{d\tau} = v_1(\tau) - v_2(\tau) = 1$$

At time $\tau = 2$, place p_1 will be empty. Now, transition t_1 may not fire at its maximal speed, but may at most remove the marks added to place p_1 by the firing of t_2 at speed $v_2(\tau) = 1$; thus $v_1(\tau) = 1$

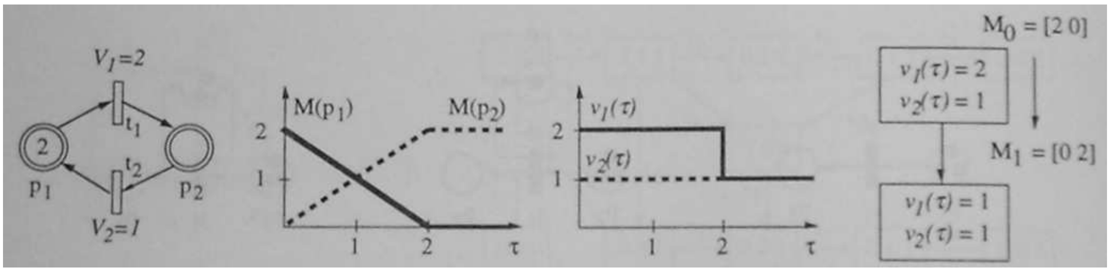

Figure 4: A timed continuous Petri net with constant maximal firing speeds.

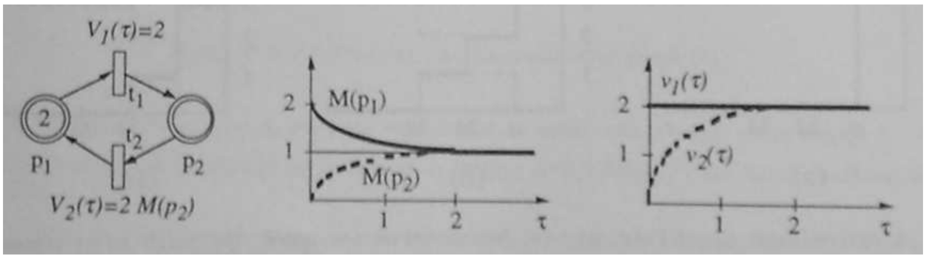

Figure 5: A timed continuous Petri net with variable maximal firing speeds.

henceforth. The new evolution is given by the equations:

$$\frac{dM(p_1)}{d\tau} = v_2(\tau) - v_1(\tau) = 0$$
$$\frac{dM(p_2)}{d\tau} = v_1(\tau) - v_2(\tau) = 0$$

The evolution of a timed net is given by a graph in which the different phases (each one characterized by a particular value of the firing speeds) are shown, as in the RHS of Figure 4.

It is also be possible to consider nets with variable maximal firing speeds. In this case V_i may be a function of time τ (time dependent maximal speeds) or a function of the marking of some place of the net (marking dependent maximal speeds).

Example 3 Consider the net in Figure 5. Initially place p_1 is marked and $v_1(\tau) = V_1 = 2$. Thus, in a interval $d\tau$ transition t_1 will remove 2 $d\tau$ marks from p_1 and add 2 $d\tau$ marks to p_2. During the same interval, transition t_2 may fire at his maximal firing speed, i.e., $v_2(\tau) = 2M(p_2)$, removing $2M(p_2)$ $d\tau$ marks from p_2 and adding $2M(p_2)$ $d\tau$ marks to p_1. Thus the marking evolution is ruled by the equations:

$$\frac{dM(p_1)}{d\tau} = v_2(\tau) - v_1(\tau) = 2M(p_2) - 2$$
$$\frac{dM(p_2)}{d\tau} = v_1(\tau) - v_2(\tau) = 2 - 2M(p_2)$$

As $M(p_2)$ goes asymptotically to 1 the net reaches a steady state, as shown in the plots in Figure 5.

Finally, a *hybrid net* [9], is net in which both discrete and continuous places and transitions are present.

We conclude this section defining a new kind of transition that will be used in the modeling of canal networks: the *continuous transition with delay arc* shown in Figure 6.(a). The arc marked with Δ — that goes from a continuous transition to a continuous place — introduces a delay d. The firing of t_1 in Figure 6.(a) will change the marking of place p'_2 as shown in the plot. The marking of place p_2 will have the same evolution with a delay d. It is interesting to note that although the notion of *arc with delay* is new, it can be considered as the limit for $k \to 0$ of the hybrid net shown in Figure 6.(b). Here,

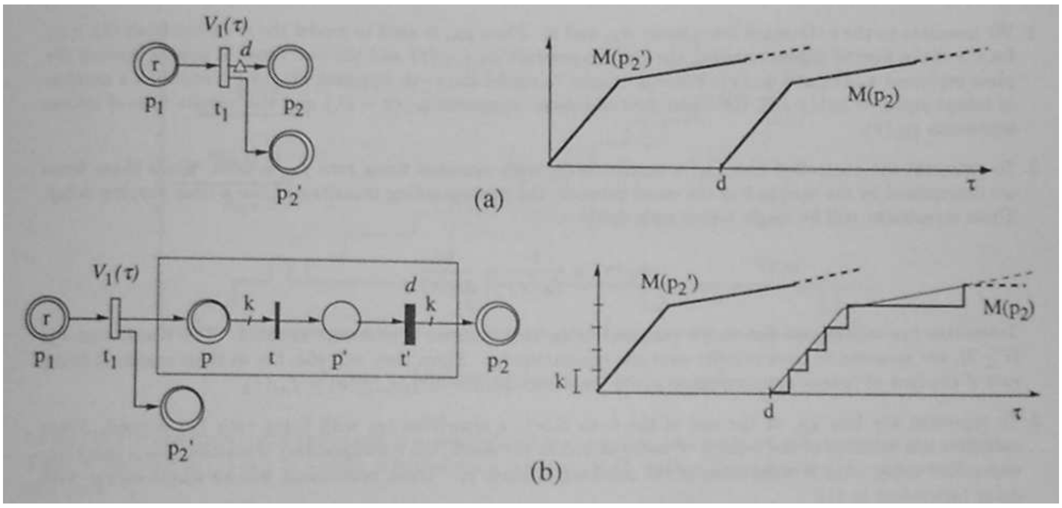

Figure 6: (a) A continuous transition with delay arc. (b) Equivalent structure.

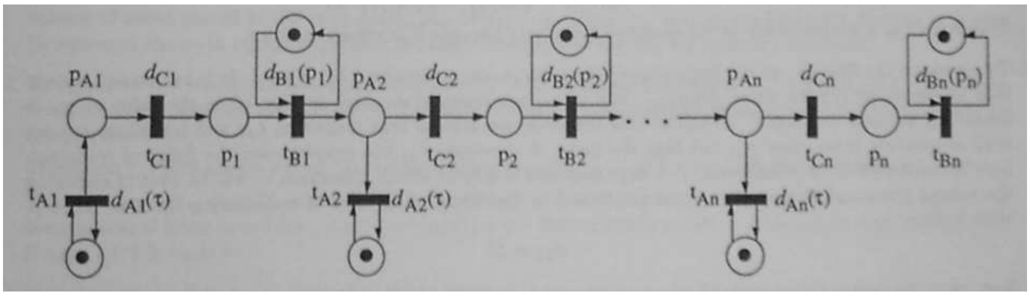

Figure 7: Discrete Petri net model of a canal with n reaches.

we assume that the marks produced by the firing of t_1 are put in places p. As soon as the marking of place p reaches the value k the immediate transition t (it is a transition with zero delay) fires removing the k marks from p and adding a token to the discrete place p'. After a delay of d, transition t' will fire adding k marks to place p_2, as shown in the plot in Figure 6.(b).

4 Discrete Petri net model

In this section we show how a deterministic timed Petri net model that implements equations (1-3) may be constructed. The model will be discretized, using a suitable unit, say, u [m^3]. Thus, a marking $M(p) = n$ in a place p represents a volume $\rho = nu$ [m^3], while a firing rate $f = r$ [s^{-1}] represents a flow $q = ru$ [m^{3}s^{-1}]. We write $\mathtt{int}(x)$ to denote the largest integer smaller or equal to x.

The DTPN model for the system shown in Figure 1, has the form shown in Figure 7.

1. We associate to the $i-$th reach two places: p_{Ai} and p_i. Place p_{Ai} is used to model the $i-$th equation (2), e.g., for $i > 2$ the flow of tokens entering the place represents $q_{B(1-i)}(\tau)$ and the two flows of tokens leaving the place represent $q_{Ai}(\tau)$ and $q_{Ci}(\tau)$. Place p_i is used to model the $i-$th equation (3), i.e., it contains a number of tokens equal to $\mathtt{int}(\rho_i/u)$, the input flow of tokens represents $q_{Ci}(\tau - D_i)$ and the output flow of tokens represents $q_{Bi}(\tau)$.
2. To represent the controlled flow q_{Ai}, a transition t_{Ai} with maximal firing rate f_{Ai} is used. Since

these flows are determined by the operator of the canal network, the corresponding transitions have a time-varying delay. These transitions will be single server with delay:

$$d_{Ai}(\tau) = \frac{1}{f_{Ai}(\tau)} = \frac{u}{q_{Ai}(\tau)}$$

Transition t_{A1} will always fire at his maximal firing rate because it is always enabled. The transitions t_{Ai} $(i \geq 2)$, are assumed to have priority over the transitions t_{Ci}. Thus, they will also fire at their maximal firing rate if the flow of tokens entering place p_{A2} is large enough, i.e., if $f_{B(i-1)}(\tau) \geq f_{Ai}(\tau)$.

3. To represent the flow q_{Bi} at the end of the $i-$th reach, a transition t_{Bi} with firing rate f_{Bi} is used. Since each flow is a function of the volume of water stored in the reach, the corresponding transition has a marking-dependent delay, that is a function of the marking of place p_i. These transitions will be single server with delay (according to (1):

$$d_{Bi}(p_i) = \frac{1}{f_{Bi}(p_i)} = \frac{u}{q_{Bi}(\rho_i)} = \frac{u}{\varphi(M(p_i) \cdot u)}$$

Transition t_{Bi} will always fire at his maximal firing rate because it is always enabled.

4. To represent the flow q_{Ci} at the beginning of the $i-$th reach, a transition t_{Ci} is used. It is important to note that in equations (2) the value of $q_{B(i-1)}$ and of q_{Ai} are assumed as given inputs, while the value of q_{Ci} is computed as their difference. To model this behavior, we assume that transition t_{Ai} and transition t_{Ci} are both outputting from place p_{Ai} but that the firing of transition t_{Ai} has priority over the firing of transition t_{Ci}. Thus, the effect of transition t_{Ci} — note that this is infinite server transition — will be that of removing the tokens produced by $t_{b(i-1)}$ and not consumed by the firing of t_{Ai} and of transferring them to place p_i after a delay

$$d_{Ci} = D_i$$

Note that the delay of transition t_{Ci} is not used to impose a firing rate — in fact, the firing rate of an infinite server transition depends only on the tokens available — but to model the delay D_i associated to each reach.

As an example, in Figure 8 we have represented the marking evolution of a place p_{Ai}. We have also represented with a thin continuous line the contribution to the marking of place p_{Ai} given by the firing of transitions $t_{B(i-1)}$ and t_{Ai}, while we have represented with a thin dotted line the contribution to the marking of place p_i given by the firing of transition t_{Ci}. Note that starting from a zero marking, the marking evolution converges to a steady state.

The initial marking of the net represents the initial conditions of the canal network. We assume that the initial marking of the places p_i is equal to $\texttt{int}(\rho_i(0)/u)$, while we assume that the places p_{Ai} are initially empty. As shown in Figure 8, after a transient the simulation result will converge to a steady state.

We finally observe that the model may be further refined. As an example, we may want to model the overflow condition in each reach introducing an immediate transition that will fire removing tokens form a place p_i as soon as the value of $M(p_i)$ goes over a given threshold. We may also explicitly model the maximal flows of each reach by assuming that the transitions t_{Ci} are not infinite server but k-server transitions.

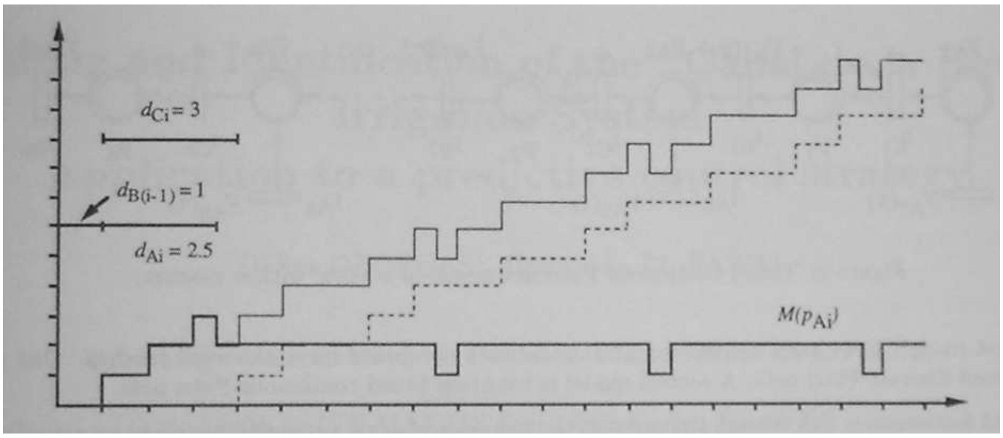

Figure 8: Marking evolution at the $i-$th node.

5 Continuous Petri net model

A timed continuous Petri net model may also be constructed for the canal network shown in Figure 1, and takes the form of the net in Figure 9.

In the TCPN the marking of the places is a nonnegative real number, and we need not discretize the variables. Thus, a marking $M(p) = \rho$ in a place p represents a volume ρ [m^3], while a firing speed $v = q$ represents a flow of q [m^3s^{-1}].

1. We associate to the $i-$th reach two places: p_{Ai} and p_i. Place p_i contains a number of tokens equal to the volume of water stored in the $i-$th reach, i.e., $M(p_i) = \rho_i$. Place p_{Ai} will always be empty because it is used to represent the $i-$th equation (2) and the flows entering and leaving the node are balanced.

2. To represent the controlled flow q_{Ai}, a transition t_{Ai} with maximal firing speed $V_{Ai} = q_{Ai}$ is used. Since these flows are determined by the operator of the canal network, the corresponding transitions have a time-varying firing speed.

   Transition t_{A1} will always fire at his maximal firing speed (i.e., $v_{A1}(\tau) = V_{A1}(\tau)$) since it has no input places.

   The transitions t_{Ai} $(i \geq 2)$, are assumed to have priority over the transitions t_{Ci}. Thus, they will also fire at their maximal firing speed (i.e., $v_{Ai}(\tau) = V_{Ai}(\tau)$) if the flow of marks entering place p_{A2} is large enough, i.e., if $v_{B(i-1)}(\tau) \geq V_{Ai}(\tau)$.

3. To represent the flow q_{Bi} at the end of the $i-$th reach, a transition t_{Bi} with maximal firing speed V_{Bi} is used. Since each flow is a function of the volume of water stored in the reach, the corresponding transition has a marking-dependent speed, that is a function of the marking $M(p_i) = \rho_i$ of place p_i, i.e.,

$$V_{Bi}(p_i) = q_{Bi}(M(p_i))$$

   Note that the instantaneous speed of this transition is $v_{Bi} = V_{Bi}$, because place p_i is always marked.

4. To represent the flow q_{Ci} at the beginning of the $i-$th reach, a transition t_{Ci} is used. It is important to note that the firing of transition t_{Ai} has priority over the firing of transition t_{Ci}. Thus, the effect of transition t_{Ci} will be that of removing the marks not consumed by the firing of t_{Ai} and transfer them to place p_i after a delay

$$d_i = D_i$$

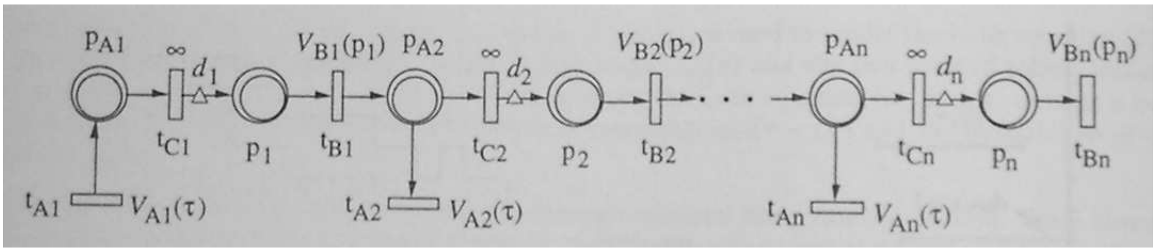

Figure 9: Timed continuous Petri net model of a canal with n reaches.

Note that we have assigned to all t_{Ci} a maximal firing speed $V_{Ci} = \infty$. However, the instantaneous firing speed will always be finite and equal to $v_{Ci}(\tau) = v_{B(i-1)}(\tau) - v_{Ai}(\tau)$.

The initial marking of the net represents the initial conditions of the canal network. Thus we assume that the initial marking of the places p_i is equal to $\rho_i(0)$, while we assume that the places p_{Ai} are initially empty.

6 Conclusions

In this paper we have used Petri nets to model canal networks with self-levelling gates.

Two different models have been derived for a canal network composed on n cascaded reaches. One model is based on timed discrete Petri nets. A second model is based on timed continuous Petri nets.

The Petri net formalism is rich enough to represent the behavior of canal networks described by equations (1-3). The approach can be easily extended to networks with more complex topologies, e.g., with forks and joins.

In the future, we plan to extend this approach to canal networks ruled by more general equations, such as those described in [5], in which the water level at the end of the $i-$th reach may vary according to a prescribed control law. We also plan to address simulation and water management issues.

References

[1] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, F. Franceschinis, *Modelling with Generalized Stochastic Petri Nets*, John Wileys and Sons, 1995.

[2] D. Andreu, J.C. Pascal, H. Pingaud, and R. Valette, "Batch process modelling using Petri nets," *IEEE Int. Conf. on Systems, Man and Cybernetics* (San Antonio, Texas), pp. 314-319, October, 1994.

[3] G. Corriga, G. Fadda, S. Sanna, G. Usai, "Optimum management of irrigation canal networks with self-levelling gates," *Proc. Int. ASME Conf. on Modelling and Simulation* (Athens, Greece), Vol. 4.1, pp. 121-134, June, 1984.

[4] G. Corriga, F. Patta, S. Sanna, G. Usai, "Mathematical model for networks of canals with self-levelling gates," *Proc. 17th Cong. of the Int. Association for Hydraulic Research* (Baden-Baden, Germany), Vol. 2, pp. 335-342, August, 1977.

[5] G. Corriga, F. Patta, S. Sanna, G. Usai, "A mathematical model for open-channel networks," *Applied Mathematical Modeling*, Vol. 3, pp. 51-54, February, 1977.

[6] R. David, H. Alla, "Autonomous and timed continuous Petri nets," *Advances in Petri Nets 1993*, G. Rozenberg (Ed.), LNCS Vol. 674, pp. 71–90, Springer-Verlag, 1992.

[7] R. David, H. Alla, *Petri Nets and Grafcet*, Prentice-Hall, 1992.

[8] A. Fanni, A. Giua, D.-Y. Lee, "Petri nets in the qualitative modelling of systems," *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics* (San Antonio, Texas), pp. 2316–2321, October, 1994.

[9] J. Le Bail, H. Alla, R. David, "Hybrid Petri nets," *Proc. 1st European Control Conference* (Grenoble, France), pp. 1472-1477, July 1991.

[10] T. Murata, "Petri nets: properties, analysis and applications," *Proc. IEEE*, Vol. 77(4), pp. 541–580, 1989.

[11] J.L. Peterson, *Petri net theory and the modeling of systems*, Prentice-Hall, 1981.

[12] P.J. Ramadge, W.M. Wonham, "The control of discrete event systems," *Proc. IEEE*, Vol. 77(1), pp. 81–98, 1989.

[13] W. Reisig, *Petri nets: an introduction*, Springer-Verlag, 1985.