A Stochastic Process Algebra Model for the Analysis of the Alternating Bit Protocol *

Marco Bernardo Lorenzo Donatiello Roberto Gorrieri

Università di Bologna, Dipartimento di Scienze dell'Informazione
Mura Anteo Zamboni 7, 40127 Bologna, Italy
E-mail: {bernardo, donat, gorrieri}@cs.unibo.it

Abstract

We illustrate an application of the integrated approach for modeling and analyzing concurrent systems proposed in [4]. The case study concerns the alternating bit protocol. The protocol is specified by means of a set of terms of the stochastic process algebra EMPA, and then studied from the functional and the performance point of view after constructing the semantic models associated with the terms.

Keywords: process algebras, Petri nets, performance modeling.

1 Introduction

In [4] an approach for modeling and analyzing concurrent systems has been proposed. The purpose of this approach is to tackle the problem of integrating the performance modeling and analysis of a concurrent system into the design process of the system itself, in order to avoid both negative consequences for design costs and late delivery arising from the fact that efficiency tests are usually performed after functionality tests.

The approach is based on stochastic process algebras and stochastic Petri nets, since they are formal description techniques well suited to describe the functionality and the performance of concurrent systems.

Stochastic Petri nets (see, e.g., [1]) constitute one of the most mature fields where functional and performance aspects of concurrent systems are considered since the beginning of the design process. The advantage of using a stochastic Petri net is that these two aspects can be analyzed on two different "projected models" (a classical Petri net and a stochastic process) obtained from the same "integrated model" (the stochastic Petri net), so we are guaranteed that the two projected models are consistent. However two problems have to be addressed: (*i*) lack of compositionality, i.e. the capability of constructing nets by composing smaller ones, and (*ii*) inability to perform an integrated analysis, i.e. an analysis carried out directly on the integrated model, which can be much more efficient as there is no need of building projected models. Both problems can be overcome by resorting to stochastic process algebras.

Stochastic process algebras (see, e.g., [7, 9, 4]) are algebraic languages whose key feature is compositionality. Syntactical compositionality is related to system modeling:

*This research has been partially funded by MURST, CNR and COST 247.

stochastic process algebras provide the system designer with a small set of powerful operators whereby it is possible to construct process terms from simpler ones. Semantic compositionality is related to system analysis: stochastic process algebras enable the system designer to study separately the various system components. Functional and performance analyses can be carried out on two consistent projected semantic models (a transition system labeled only on the type of the actions, and a stochastic process), as well as directly on the integrated semantic model (a transition system labeled on both the type and the duration of the actions), provided that a notion of integrated equivalence is developed that relates process terms representing concurrent systems having the same functional and performance characteristics.

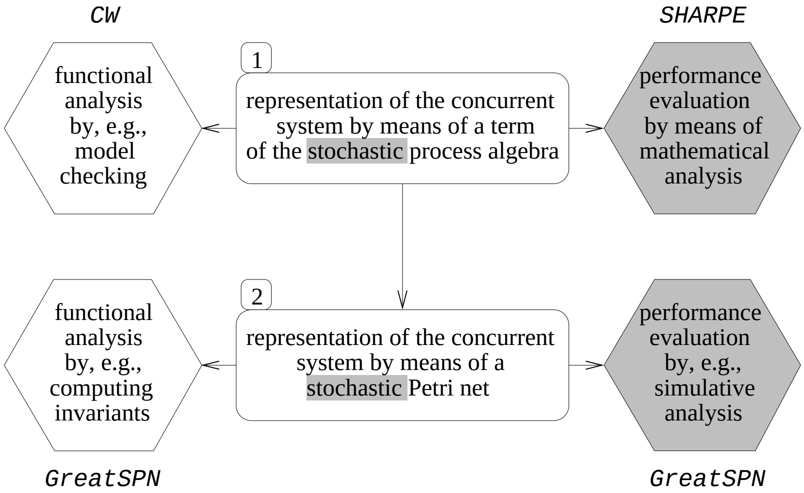

Figure 1: Integrated approach

The approach based on stochastic process algebras and stochastic Petri nets implements three orthogonal integrations. The first integration relates the abstract and the concrete views of concurrent systems. The abstract view is provided by process terms: they give an algebraic representation of system components and their interactions, whose semantic model is obtained by interleaving actions of concurrent components. The concrete view is provided by Petri nets: they give a machine-like representation of systems with the explicit description of concurrency. This integration results in the two phases depicted in Fig. 1. The second integration relates functional and performance aspects of concurrent systems. This integration is depicted in Fig. 1 by means of the contrast between the nonshaded part and the shaded part. Finally, the third integration consists of exploiting several tools tailored for specific purposes. Let us explain in more details the two phases in the light of the three orthogonal integrations.

1. The first phase requires the system designer to specify the concurrent system as a term of the stochastic process algebra. Because of syntactical compositionality, the system designer is allowed to develop the algebraic representation of the system in a modular way. From the algebraic representation, an integrated interleaving semantic model is automatically derived in the form of a transition system labeled on both the type and the duration of the actions. The integrated interleaving semantic model can be analyzed as a whole by a notion of integrated equivalence, or is projected on a functional semantic model and a performance semantic model that can be analyzed by means of tools like CW [6] and SHARPE [13], respectively.

2. The second phase consists of automatically obtaining from the algebraic representation of the system an equivalent net representation in the form of a stochastic Petri net. The net representation is useful whenever a less abstract representation is required highlighting dependencies, conflicts and synchronizations among system activities, and helpful to detect some properties (e.g., partial deadlock) that can be easily checked only in a distributed setting. Additionally, the net representation is usually more compact than the integrated interleaving semantic model, since concurrency is kept explicit instead of being simulated by alternative computations obtained by interleaving actions of concurrent components. The functional and performance analyses of the net can be assisted by tools like GreatSPN [5].

Since the two phases above are complementary, the choice between them is made depending on the adequacy of the related representation with respect to the analysis of the concurrent system under consideration, and the availability of the corresponding tools. In any case, the system designer has to start with an algebraic representation of the system in order to exploit syntactical compositionality and avoid graphical complexity of nets.

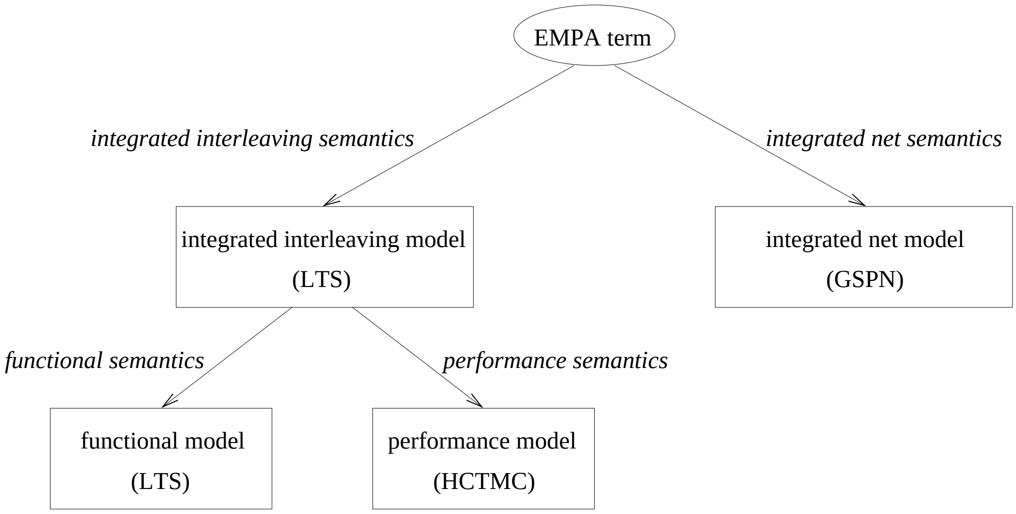

Figure 2: EMPA semantics

In order to implement the integrated approach, we have chosen the class of Generalized Stochastic Petri Nets (GSPNs) [1] due to their success, and we have developed a stochastic process algebra called Extended Markovian Process Algebra (EMPA) which is endowed with expressive features typical of GSPNs. In order to support the various phases and analyses of the integrated approach, EMPA has been equipped with a collection of semantics as depicted in Fig. 2, as well as a notion of integrated equivalence. Each term has an integrated interleaving semantics represented by a labeled transition system (LTS) whose labels consist of both the type and the duration of the actions, and an integrated net semantics represented by a GSPN. From the integrated interleaving semantic model, two projected semantic models can be obtained: a functional model given by a LTS labeled only on the type of the actions, and a performance model given by a homogeneous continuous-time Markov chain (HCTMC).

In this paper we show an application of the integrated approach to the alternating bit protocol. The protocol is specified by means of a set of EMPA terms, and then studied from the functional and the performance point of view after constructing the semantic models associated with the terms. Due to lack of space, the theory underlying EMPA is not reported here: the interested reader is referred to [4].

2 The alternating bit protocol

The alternating bit protocol [3] is a data-link level communication protocol that establishes a means whereby two stations, one acting as a sender and the other acting as a receiver, connected by a full-duplex communication channel that may lose messages, can cope with message loss. The name of the protocol stems from the fact that each message is augmented with an additional bit: since consecutive messages that are not lost are tagged with additional bits that are pairwise complementary, it is easy to distinguish between an original message and its possible duplicates. Initially, if the sender obtains a message from the upper level, it augments the message with an additional bit set to 0, sends the tagged message to the receiver, and starts a timer: if an acknowledgement tagged with 0 is received before the timeout expires, then the subsequent message obtained from the upper level will be sent with an additional bit set to 1, otherwise the current tagged message is sent again. On the other side, the receiver waits for a message tagged with 0: if it receives such a tagged message for the first time, then it passes the message to the upper level, sends an acknowledgement tagged with 0 to the sender, and waits for a message tagged with 1, whereas if it receives a duplicate tagged message (due to message loss, acknowledgement loss, or propagation taking an arbitrarily long time), then it sends an acknowledgement tagged with the same additional bit to the sender.

How to model the alternating bit protocol with EMPA? Since it is helpful to take advantage from syntactical compositionality, we figure out to deal with three entities: the sender *Snd*, the channel *Ch*, and the receiver *Rcv*. The interaction between *Snd* and *Ch* is described by action types tm_i, $i \in \{0,1\}$, standing for "transmit message tagged with i", and da_i, $i \in \{0,1\}$, standing for "deliver acknowledgement tagged with i". The interaction between *Rcv* and *Ch* is described by action types dm_i, $i \in \{0,1\}$, standing for "deliver message tagged with i", and ta_i, $i \in \{0,1\}$, standing for "transmit acknowledgement tagged with i". Recalling that in EMPA "$\|_S$" is the parallel composition operator forcing the synchronization on actions whose type is in S, the scenario can be modeled as follows:

$$ABP \triangleq Snd_0 \,\|_{\{tm_0,tm_1,da_0,da_1\}}\, Ch \,\|_{\{dm_0,dm_1,ta_0,ta_1\}}\, Rcv_0$$

Now we can focus our attention on the individual entities thanks to syntactical compositionality. *Ch* is composed of two independent half-duplex lines Ln_m and Ln_a. The local activities of *Ch* are described by action types pm_i, $i \in \{0,1\}$, standing for "propagate message tagged with i", and pa_i, $i \in \{0,1\}$, standing for "propagate acknowledgement tagged with i". Additionally, there are other two activities local to *Ch* that are described by the internal action type τ and represent the fact that a message or an acknowledgement gets lost or not. Concerning the performance part of actions in which *Ch* is involved, we recall that EMPA allows us to express only exponentially distributed durations as well as zero durations. Supposing that the length of a message and the length of an acknowledgement are exponentially distributed, it turns out that transmission, propagation, and delivery times are exponentially distributed. However, the three phases given by transmission, propagation, and delivery are temporally overlapped. As a consequence, in order to determine correctly the time taken by a message/acknowledgement to reach *Rcv*/*Snd*, we model actions related to transmission and delivery as immediate and we associate the actual timing with actions related to propagation: we assume that the message propagation time is exponentially distributed with rate δ, the acknowledgement propagation time is exponentially distributed with rate γ, and the loss probability is $p \in \mathbb{R}_{]0,1[}$.

Before the description of *Ch*, we recall that in EMPA each action is denoted by a pair "$<a, \tilde{\lambda}>$" where a is the type and $\tilde{\lambda}$ is the rate: $\tilde{\lambda} \in \mathbb{R}_+$ if the duration of the action is exponentially distributed with parameter $\tilde{\lambda}$ (exponentially timed action), $\tilde{\lambda} = \infty_{l,w}$ if the duration of the action is zero and the action has priority level l and weight w (immediate action), $\tilde{\lambda} = *$ if the duration of the action is unspecified (passive action). Furthermore, we recall that in EMPA "." is the prefix operator expressing the sequential composition of an action and a term, and "+" is the alternative composition operator expressing a choice between two terms. *Ch* can be modeled as follows:

- $Ch \stackrel{\Delta}{=} Ln_m \,\|_{\emptyset}\, Ln_a$:
  - $Ln_m \stackrel{\Delta}{=} {<}tm_0, *{>}.{<}pm_0, \delta{>}.({<}\tau, \infty_{1,1-p}{>}.{<}dm_0, \infty_{1,1}{>}.Ln_m + {<}\tau, \infty_{1,p}{>}.Ln_m) + {<}tm_1, *{>}.{<}pm_1, \delta{>}.({<}\tau, \infty_{1,1-p}{>}.{<}dm_1, \infty_{1,1}{>}.Ln_m + {<}\tau, \infty_{1,p}{>}.Ln_m)$;
  - $Ln_a \stackrel{\Delta}{=} {<}ta_0, *{>}.{<}pa_0, \gamma{>}.({<}\tau, \infty_{1,1-p}{>}.{<}da_0, \infty_{1,1}{>}.Ln_a + {<}\tau, \infty_{1,p}{>}.Ln_a) + {<}ta_1, *{>}.{<}pa_1, \gamma{>}.({<}\tau, \infty_{1,1-p}{>}.{<}da_1, \infty_{1,1}{>}.Ln_a + {<}\tau, \infty_{1,p}{>}.Ln_a)$.

Observe that the probabilistic choice between the reception and the loss of a message or an acknowledgement has been easily represented by means of the weights associated with the two immediate actions ${<}\tau, \infty_{1,1-p}{>}$ and ${<}\tau, \infty_{1,p}{>}$.

The local activities of *Snd* are described by action types *gm* standing for "generate message", and *to* standing for "timeout". Assuming that the message generation time is exponentially distributed with rate λ, and that the timeout period is exponentially distributed with rate θ (though this is not realistic), *Snd* can be modeled as follows:

- $Snd_0 \stackrel{\Delta}{=} {<}gm, \lambda{>}.{<}tm_0, \infty_{1,1}{>}.Snd'_0$,
  $Snd'_0 \stackrel{\Delta}{=} {<}da_0, *{>}.Snd_1 + {<}da_1, *{>}.Snd'_0 + {<}to, \theta{>}.Snd''_0$,
  $Snd''_0 \stackrel{\Delta}{=} {<}tm_0, \infty_{1,1}{>}.Snd'_0 + {<}da_0, *{>}.Snd_1 + {<}da_1, *{>}.Snd''_0$,
  $Snd_1 \stackrel{\Delta}{=} {<}gm, \lambda{>}.{<}tm_1, \infty_{1,1}{>}.Snd'_1$,
  $Snd'_1 \stackrel{\Delta}{=} {<}da_1, *{>}.Snd_0 + {<}da_0, *{>}.Snd'_1 + {<}to, \theta{>}.Snd''_1$,
  $Snd''_1 \stackrel{\Delta}{=} {<}tm_1, \infty_{1,1}{>}.Snd'_1 + {<}da_1, *{>}.Snd_0 + {<}da_0, *{>}.Snd''_1$.

Since Snd''_0 and Snd''_1 model the situation after a timeout expiration, they should comprise the retransmission action only in order to be consistent with the definition of the protocol. Snd''_0 and Snd''_1 are also allowed to receive acknowledgements in order to avoid the deadlock that may occur whenever, after a sequence of premature timeouts (i.e. timeouts expired although nothing gets lost), the sender is waiting to be able to retransmit the message, the receiver is waiting to be able to retransmit the corresponding acknowledgement, the message line is waiting to be able to deliver a previous copy of the message, and the acknowledgement line is waiting to be able to deliver a previous copy of the acknowledgement.

The only local activity of *Rcv* is described by action type *cm* standing for "consume message". Assuming that message consumption is an immediate action in that irrelevant from the performance viewpoint, *Rcv* can be modeled as follows:

- $Rcv_0 \stackrel{\Delta}{=} {<}dm_0, *{>}.{<}cm, \infty_{1,1}{>}.{<}ta_0, \infty_{1,1}{>}.Rcv_1 + {<}dm_1, *{>}.{<}ta_1, \infty_{1,1}{>}.Rcv_0$,
  $Rcv_1 \stackrel{\Delta}{=} {<}dm_1, *{>}.{<}cm, \infty_{1,1}{>}.{<}ta_1, \infty_{1,1}{>}.Rcv_0 + {<}dm_0, *{>}.{<}ta_0, \infty_{1,1}{>}.Rcv_1$.

We have chosen the alternating bit protocol as a case study to illustrate the integrated approach because such a protocol has become a standard example in the literature (see, e.g., [10, 6, 11, 8, 12, 2]), so it can be used to compare the EMPA model with other models. For example, it turns out that in [10, 6] performance aspects are completely neglected because a classical process algebra is used, while in [11] a stochastic Petri net model is adopted but the unrealistic assumption that the timeout expires only if a loss actually occurs is made. In [8] a stochastic process algebraic description is given, where the deterministic duration of the timeout period has been approximated by means of a sequence of exponentially distributed delays. However, this description does not

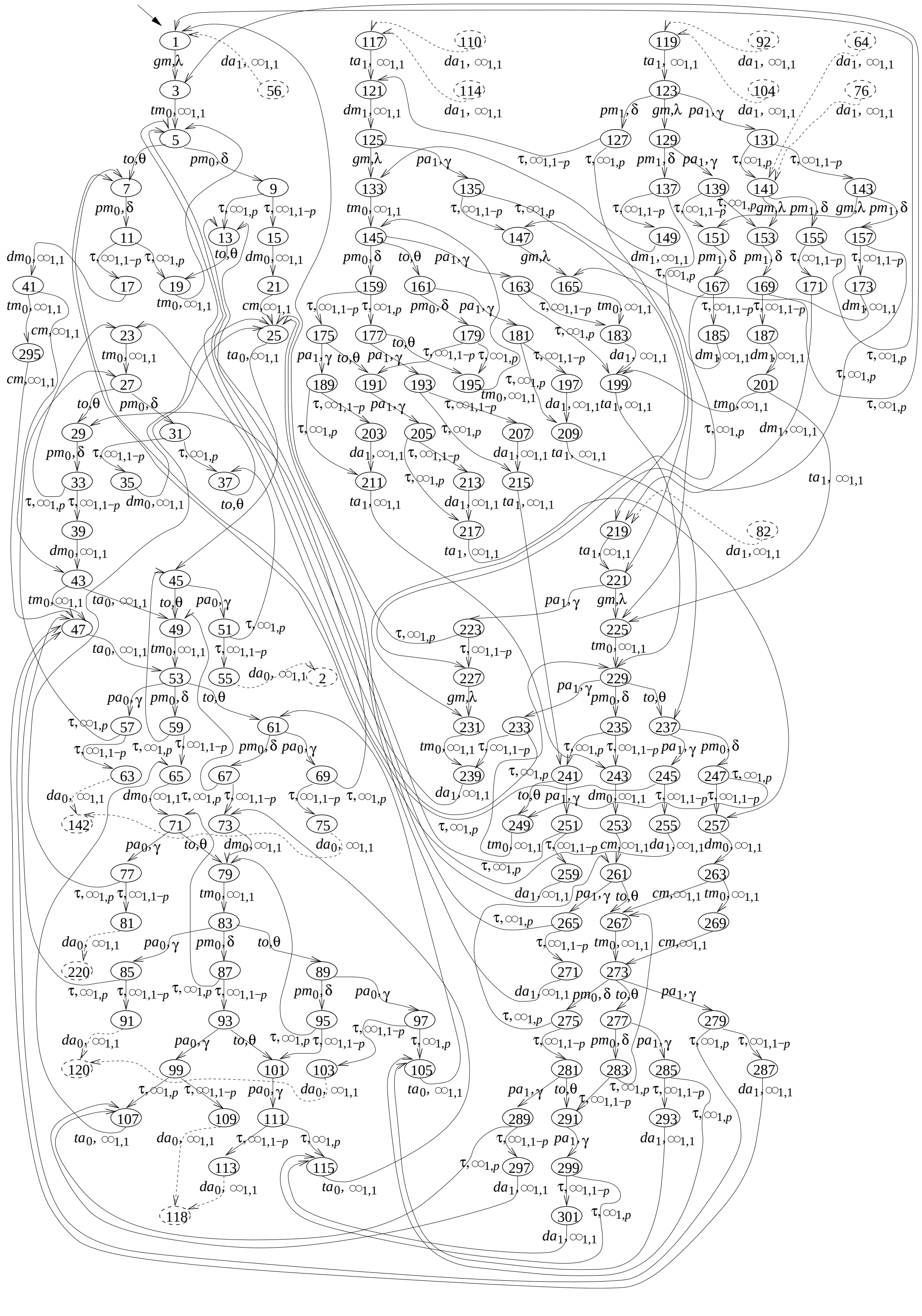

Figure 3: Integrated interleaving semantics of *ABP*

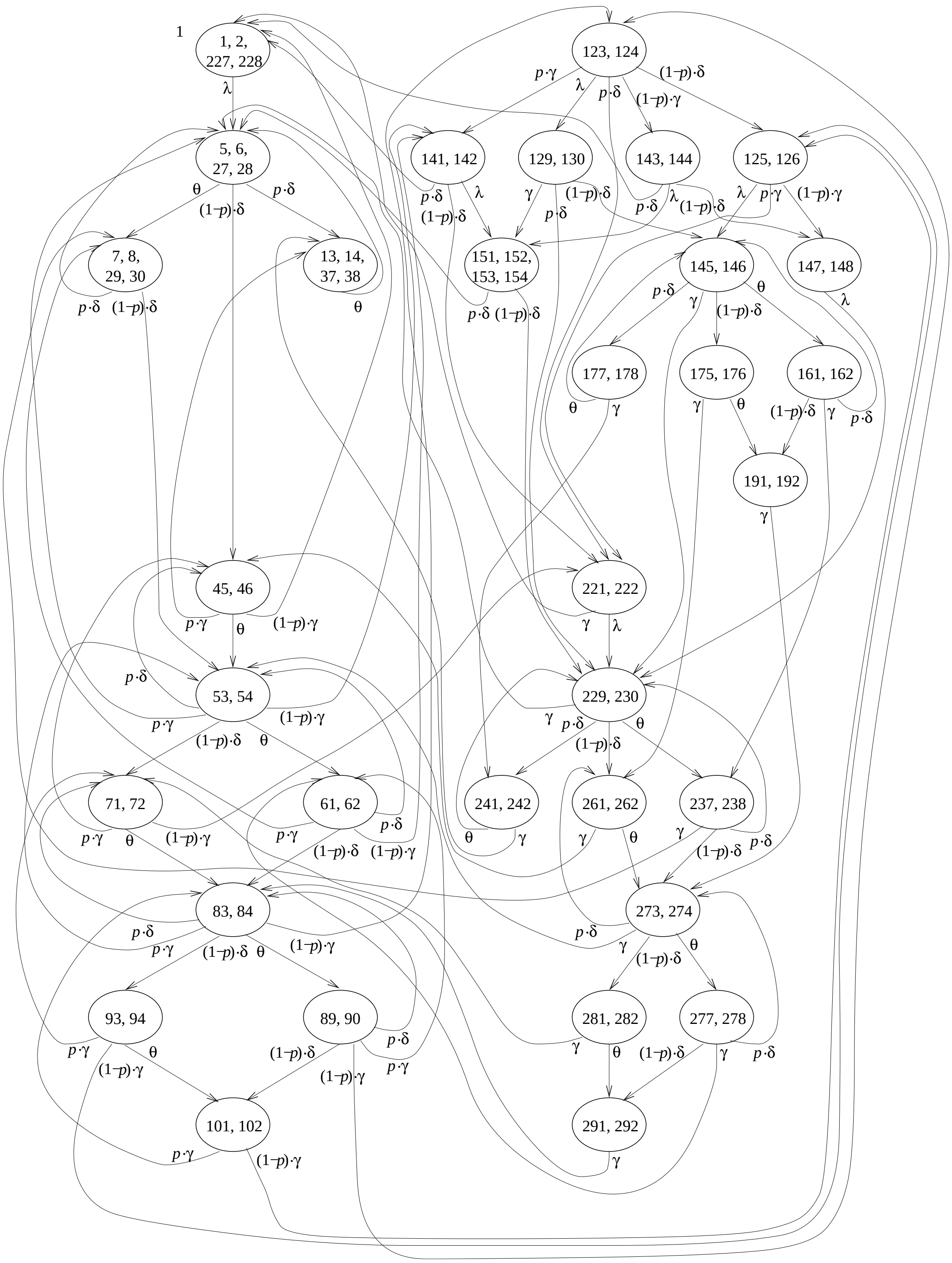

Figure 4: Performance semantics of *ABP*

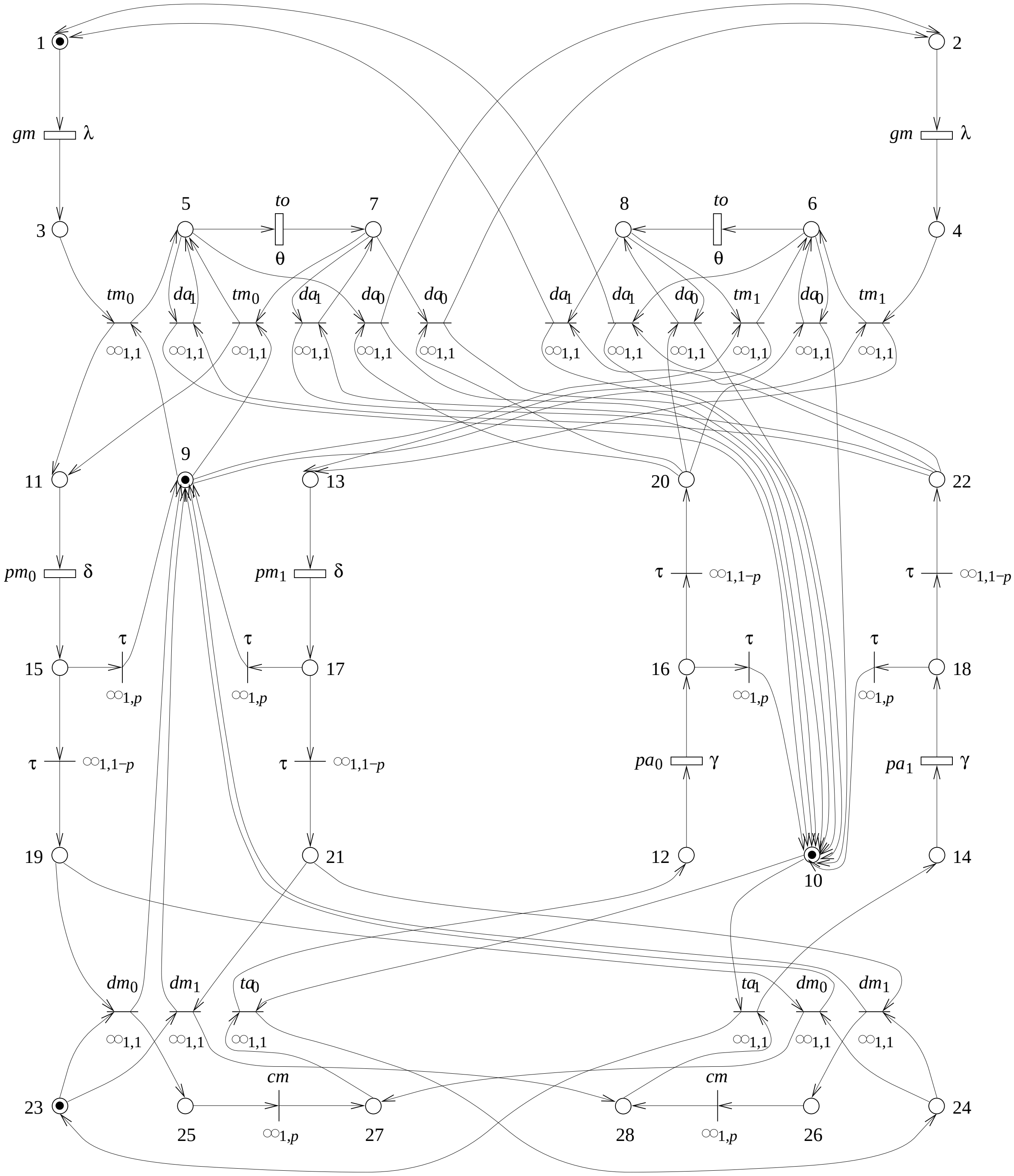

Figure 5: Integrated net semantics of *ABP*

accurately take into account the division into three temporally overlapped phases (like in [12, 2]), and represents the probabilistic choice between the reception and the loss of a message/acknowledgement in a tricky way by giving a context-dependent meaning to the rate of the actions.

The integrated interleaving semantics $\mathcal{I}[\![ABP]\!]$ is presented in Fig. 3 (for the shorthands of state names, the reader is referred to [4]). This LTS has 302 states: due to the symmetry of the protocol, only half states have been drawn (dashed transitions depict the link with the remaining states). Whenever neither losses nor premature timeouts occur, the states visited by the protocol are 1, 3, 5, 9, 15, 21, 25, 45, 51, 55 and the corresponding symmetric ones, i.e. 2, 4, 6, 10, 16, 22, 26, 46, 52, 56. Following the proposed approach, we can use the functional semantics $\mathcal{F}[\![ABP]\!]$, obtained from $\mathcal{I}[\![ABP]\!]$ by dropping action rates, to detect some functional properties. For example, we see that each state has at least one incoming transition and one outgoing transition: this means that the protocol is deadlock free. The LTS $\mathcal{F}[\![ABP]\!]$ could be fully analyzed by using tools like CW [6].

The performance semantics $\mathcal{M}[\![ABP]\!]$ is presented in Fig. 4. This HCTMC has 33 states, and it has been obtained from the LTS $\mathcal{I}[\![ABP]\!]$ by discarding action types, removing the 226 states having outgoing immediate transitions, and aggregating the remaining 76 states. Since $\mathcal{M}[\![ABP]\!]$ is finite and strongly connected, it represents a HCTMC for which the steady-state probability distribution function $\boldsymbol{\pi}$ exists. Following the proposed approach, we can exploit such a HCTMC for assessing some performance indices. For example, the throughput of the protocol is given by λ times the sum of the steady-state probabilities of the states having an outgoing transition labeled with λ. The HCTMC $\mathcal{M}[\![ABP]\!]$ could be fully analyzed by using tools like SHARPE [13].

The integrated net semantics $\mathcal{N}[\![ABP]\!]$ is presented in Fig. 5 (for the shorthands of place names, the reader is referred to [4]). This GSPN comprises 28 places and 36 transitions. ABP and $\mathcal{N}[\![ABP]\!]$ model exactly the same protocol in two different ways: the algebraic description is compositional and more readable, the net description is more concrete and highlights dependencies, conflicts and synchronizations among activities. Also, $\mathcal{N}[\![ABP]\!]$ is more compact than $\mathcal{I}[\![ABP]\!]$ because the concurrency is kept explicit instead of being simulated by means of interleaving. Following the proposed approach, we can exploit such a GSPN for studying functional and performance properties. This can be assisted by tools like GreatSPN [5].

3 Conclusion

In this paper we have shown an application of the exponential version of the integrated approach proposed in [4] to the alternating bit protocol. Such a case study should have stressed, besides the advantages gained by integrating different paradigms, the expressiveness of EMPA. This has been one of the major goal during the development of our stochastic process algebra. In particular, in EMPA action durations are mainly expressed by means of exponentially distributed random variables (like in the stochastic process algebras MTIPP [7] and PEPA [9]), but it is also possible to express prioritized weighted immediate actions (like in GSPNs [1]) and passive actions. Immediate actions permit to model activities that are irrelevant from the performance viewpoint, and allow to express both prioritized and probabilistic choices. Finally, passive actions have unspecified durations thus allowing for pure nondeterminism.

Our current research is directed along two different lines. From the applicative point of view, we are developing a software tool that implements the integrated approach in the exponential case. From the theoretical point of view, we are addressing the problem of scaling the integrated approach to general distributions. It is however important to point out that the limitation to exponentially distributed durations is (*i*) convenient because exponential timing allows for a Markovian analysis without resorting to time-costly simulations, and (*ii*) not so restrictive because many frequently occurring distributions are (or can be approximated by) phase-type distributions, and these are expressible in EMPA by means of the interplay of exponentially timed and immediate actions.

References

[1] M. Ajmone Marsan, G. Balbo, G. Conte, *"A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems"*, ACM Trans. on Computer Systems 2:143-172, 1984

[2] M. Ajmone Marsan, A. Bianco, L. Ciminiera, R. Sisto, A. Valenzano, *"A LOTOS Extension for the Performance Analysis of Distributed Systems"*, IEEE/ACM Trans. on Networking 2(2):151-164, 1994

[3] K. A. Bartlett, R. A. Scantlebury, P. T. Wilkinson, *"A Note on Reliable Full-Duplex Transmission over Half-Duplex Links"*, CACM 12(5):260-261, 1969

[4] M. Bernardo, L. Donatiello, R. Gorrieri, *"Integrating Performance and Functional Analysis of Concurrent Systems with EMPA"*, Technical Report UBLCS-95-14, University of Bologna (Italy), 1995, available from `ftp.cs.unibo.it:/pub/TR/UBLCS`

[5] G. Chiola, *"GreatSPN 1.5 Software Architecture"*, Proc. of the 5th Int. Conf. on Modeling Techniques and Tools for Computer Performance Evaluation, Elsevier, pp. 121-136, 1991

[6] R. Cleaveland, J. Parrow, B. Steffen, *"The Concurrency Workbench: a Semantics-Based Tool for the Verification of Concurrent Systems"*, ACM Trans. on Programming Languages and Systems 15(1):36-72, 1993

[7] N. Götz, U. Herzog, M. Rettelbach, *"Multiprocessor and Distributed System Design: the Integration of Functional Specification and Performance Analysis Using Stochastic Process Algebras"*, Proc. of PERFORMANCE '93, LNCS 729:121-146, 1993

[8] H. Hermanns, V. Mertsiotakis, M. Rettelbach, *"Performance Analysis of Distributed Systems Using TIPP - A Case Study"*, Proc. of the 10th UK Performance Engineering Workshop for Computer and Telecommunication Systems, 1994

[9] J. Hillston, *"A Compositional Approach to Performance Modelling"*, Ph.D. Thesis, University of Edinburgh (UK), 1994

[10] R. Milner, *"Communication and Concurrency"*, Prentice Hall, 1989

[11] M. K. Molloy, *"Performance Analysis using Stochastic Petri Nets"*, IEEE Trans. on Computers 31(9):913-917, 1982

[12] N. Nounou, Y. Yemini, *"Algebraic Specification-Based Performance Analysis of Communication Protocols"*, Protocol Specification, Testing and Verification V:541-560, 1985

[13] R.A. Sahner, K.S. Trivedi, *"Reliability Modelling Using SHARPE"*, IEEE Trans. on Reliability 13:186-193, 1987