A Mean Field Based Methodology for Modeling Mobility in Ad Hoc Networks

Marco Beccuti[1], Massimiliano De Pierro[1], András Horváth[1], Ádám Horváth[2], Károly Farkas[2]
[1]Dipartimento di Informatica - Università di Torino - beccuti@di.unito.it, depierro@di.unito.it, horvath@di.unito.it
[2]Institute of Informatics and Economics - University of West Hungary - horvath@inf.nyme.hu, farkas@inf.nyme.hu

***Abstract*—In this paper we propose a methodology for the modeling and analysis of ad hoc networks composed by a large number of nodes moving among geographical regions. This methodology uses compositional construction of stochastic Petri nets (SPN) for building the model which allows for specifying the model and the required performance indices at a high level of abstraction. As our aim is to consider real scenarios with several geographical regions and non-trivial user behavior in each region, the size of the state space of the model can easily grow too large to analyze with exact analytical approaches or even with simulation. For this reason, we propose to carry out the analysis by constructing the mean field approximation of the behavior of the SPN. The approximation is provided by a set of ordinary differential equations (ODE) that can be derived automatically from the SPN and can be solved numerically with low computational effort even for large models.**

The methodology is illustrated on a case study, modeling application spreading in a mobile environment. It will be shown that the approximate results obtained by the mean field approach capture well the behavior of the system.

***Index Terms*—ad hoc networks, mean field analysis, Petri nets;**

I. Introduction

A mobile ad hoc network (MANET) is a wireless mobile network in which a collection of mobile nodes with wireless network interfaces may form a temporary network, without the aid of any established infrastructure or centralized administration. It provides a low cost and flexible network solution in those cases when the infrastructure is not available or not feasible.

In recent years, several approaches have been proposed to study MANET systems both by means of simulation and by means of analytical models. For instance, in [1] the authors study through simulation the persistence of communication paths under various mobility patterns and network sizes; in [2] simulation is used to analyze some routing algorithms for MANET. Examples of works based on analytical models are: [3] gives a random walk mobility model to characterize node movement and study link availability, in [4] the authors show that a Poisson Boolean model is suitable for sensor networks with nodes switching between a sleeping and an active phase; and in [5] application spreading is studied through a closed queuing network.

However, in case of modeling a large number of nodes and their mobility, analytical approaches soon become unfeasible because of the size of the state space. Moreover, even simulation based analysis can become hard to apply because a high number of simulation runs have to be generated and it is not straightforward to guarantee that the whole state space is explored.

In this paper we propose a methodology to overcome the state space explosion problem. We use stochastic Petri nets (SPN) [6] to build, in a compositional manner, a model of a generic MANET system with geographical regions. We show that it is possible to derive automatically the set of ordinary differential equations (ODEs) that correspond to the mean field approximation of the model. These ODEs can be solved then by numerical integration to obtain the required performance indexes.

The paper is organized as follows. Section II gives a brief introduction of SPN. In Section III we present the proposed methodology to model and analyze MANET. A case study illustrates the approach in Section IV. Conclusions are drawn in Section V.

II. Stochastic Petri Net formalism

In this section we provide a brief introduction to SPNs while a detailed introduction with applications can be found in [6].

SPNs are bipartite directed graphs with two types of nodes: places and transitions. An example of a SPN is the one depicted in Fig. 2. The places, graphically represented as circles, correspond to the state variables of the system, while the transitions, graphically represented as boxes, correspond to the events that can induce a state change. Examples of places for the SPN in Fig. 2 are *Act_A* and *Act_B*; while examples of transitions are *Stop_A* and *Stop_B*. The arcs connecting places to transitions and vice versa express the relation between states and event occurrence.

Places can contain tokens drawn as black dots within places. The state of a SPN, called marking, is defined by the number of tokens in each place. In the paper we use the notation M to indicate a marking in general. We will denote by $M(p)$ the number of tokens in place p in marking M. Now we recall the basic definitions that are necessary for the rest of the paper.

Definition 2.1: An SPN system is a tuple

$$(P, T, I, O, \lambda, M_0)$$

where:

- $P = \{p_i\}$ is the set of *places* of cardinality k;
- $T = \{t_i\}$ is the set of *transitions* of cardinality m;
- $I, O : T \times P \to \mathbb{N}$ are the *input* and *output* functions that define the arcs of the net and their multiplicities;

- $\lambda : T \rightarrow \mathbb{R}$ is the function that assigns to each transition its firing intensity;
- M_0 is the *initial marking* of the net.

A transition is "enabled" if each of its input places contains the "necessary" amount of tokens where "necessary" is defined by the input function I. Formally, transition t is enabled in marking M if for all places p of the net we have $M(p) \geq I(t,p)$. For example, transition Start_A in Fig. 2 is enabled if the current marking contains at least one token in place Pass_A and at least one token in place Act_seeds. An enabled transition can fire and the firing removes tokens from the input places of the transition and puts tokens into the output places of the transition. The new marking M' after the firing of transition t is formally given $M'(p) = M(p) + O(t,p) - I(t,p)$, $\forall p \in P$.

The firing of a transition occurs after a random delay. The random delay associated with a transition has exponential distribution whose parameter depends on the firing intensity of the transition and on the actual marking. In this paper we assume that the more tokens enable a transition the faster the transition fires. This concept, called infinite server policy, is captured formally by the definition of the enabling degree. (Other server policies also exist and can be found in [6].)

Definition 2.2: The enabling degree of transition t in marking M, denoted by $ed(t,M)$, is d iff $\forall p \in P, M(p) \geq dI(t,p)$ and $\exists p \in P : M(p) < (d+1)I(t,p)$.

For instance, if there are three tokens in place Pass_A and four tokens in place Act_seeds then the enabling degree of transition Start_A is three. The random delay associated with transition t in marking M is exponentially distributed with parameter $\lambda(t)ed(t,M)$. When a marking is entered, a random delay is chosen for all enabled transitions by sampling the associated delay distribution. The transition with the lowest delay fires and the system changes marking.

We define the effect of the firing of transition t with an integer vector $L(t)$, place indexed, defined as: $L : T \rightarrow \mathbb{N}^k$ and the ith entry of $L(t)$ is $L(t)_i = O(t,p_i) - I(t,p_i)$ for $1 \leq i \leq k$ and $\forall t \in T$. For sake of avoiding cumbersome notation we assume that $\nexists t,t' \in T : t \neq t', L(t) = L(t')$ and $\nexists t : L(t) = 0$.

III. Methodology

In this section we propose a methodology to model and analyze MANET. First, we describe how the model is built and then provide an approximate analysis technique based on the mean field approach.

A. Modeling phase

Each single MANET, corresponding to a single geographical region, is modeled by an SPN. Then the composition of the SPNs modeling the regions is formally defined by the couple $\{N, V\}$ where:

- N is the set of SPNs to be composed and
- V is the function that describes the movement of the tokens among the regions; more specifically, given two places, p_1 and p_2, belonging to two different regions, $V(p_1,p_2)$ is the intensity with which tokens move from p_1 to p_2.

The result of the composition is a SPN containing the nets present in N and additional transitions modeling the movement among the regions. For every ordered pair of transitions, $\{p_1,p_2\}$, for which $V(p_1,p_2) > 0$ a transition with firing intensity $V(p_1,p_2)$ is added whose only input (output) place is p_1 (p_2).

In practice, the above described composition can be performed easily thanks to the composition operators defined and implemented for SPN [7].

B. Analysis phase

The most standard approach to analyze the obtained SPN model consists in constructing the continuous time Markov chain (CTMC) corresponding to the underlying stochastic behavior of the SPN [6] and performing its transient and/or steady-state analysis by analytical approaches [8] or simulation. However, this approach can be unfeasible due to the size of the state space which can grow too large especially if we consider several MANETs composed by a large number of mobile components.

In this paper we propose to evaluate the model by fluid approximation. The following definition and theorem [9], which we present in a form that is directly related to the applied definition of SPN, provide a formal relation between the CTMC and its fluid approximation.

Definition 3.1: A parametric family of Markov chains, $X_v(t)$ with $v \in \mathbb{N}$, with state spaces $E_v \subset \mathbb{Z}^k$, is called density dependent iff there exists a continuous function $f(x,l), x \in \mathbb{R}^k, l \in \{L(t_1), ..., L(t_m)\}$, such that the non-diagonal entries of the infinitesimal generator corresponding to $X_v(t)$ can be written in the form

$$q_{k,k+l} = vf\left(\frac{k}{v}, l\right),\ l \in \{L(t_1), ..., L(t_m)\} \qquad (1)$$

and the initial state of the chain is $vx_0, x_0 \in \mathbb{Z}^k$, with probability one.

Let $X(t)$ denote the solution of the ODEs

$$\frac{dX(t)}{dt} = \sum_{l \in \{L(t_1),...,L(t_m)\}} f(X(t), l) \qquad (2)$$

with initial condition $X(0) = x_0$.

Theorem 3.1: Under mild conditions on the function f (for details see [9]), the following relation holds between the function $X(t)$ and a trajectory of the CTMC $X_v(t)$:

$$\forall \delta > 0 : \lim_{v \to \infty} P\left\{\sup_{s \leq t} \left|\frac{1}{v}X_v(s) - X(s)\right| > \delta\right\} = 0. \qquad (3)$$

The interpretation of the above theorem is the following. Consider a CTMC modeling the interaction of k quantities whose state space hence is $\mathbb{Z}^k$. Imagine to observe a sequence of CTMCs with increasing initial state (i.e., the sequence of initial states is $x_0, 2x_0, 3x_0, ...$). If the increase of the initial states gives rise to a sequence of infinitesimal generators corresponding to the form in (1) then, as v is increased, the behavior of the CTMC converges to the solution of the

ODEs given in (2). The convergence is in the sense that the probability of finding any small difference between a trajectory of the CTMC and the solution of the ODEs in a finite time horizon $(0, t)$ is zero.

It has been shown in [10] that modeling interacting populations in process algebra gives rise to a family of density dependent CTMCs and, consequently, for large population sizes the corresponding ODEs provide a good approximation of the behavior of the system. With the same reasoning used in [10], it is straightforward to show that the CTMCs constructed from the Petri nets defined in Sec. IV as well are density dependent. It follows that the behavior of Petri nets with high number of tokens can be approximated by ODEs. Moreover, as we will illustrate it numerically in Sec. IV, even with a lower number of tokens the average behavior of the CTMC is approximated reasonably well by the ODEs.

In the rest of this section we describe the ODEs which provide the fluid approximation of a SPN=$\{P, T, I, O, \lambda, M_0\}$. The state of the system is a vector of real numbers and transition $t_i, 1 \le i \le m$, is moving "fluid" tokens in state $x, x \in \mathbb{R}^k$, with speed

$$s(t_i, x) = \lambda(t_i) \min_{j: I(t_i, p_j) \neq 0} \{x_j / I(t_i, p_j)\}, \quad (4)$$

i.e., the speed depends on the rate of the transition, $\lambda(t_i)$, and on the quantity of tokens present in its input places. Namely, as infinite server policy is applied, the minimum amount of fluid found in the input places (considering the multiplicity of the input place as well) limits the speed of the transition. The amount of tokens in the ith place is changing then according to the ODEs

$$\frac{dx_i(t)}{dt} = \sum_{j=1}^{m} s(t_j, x)(O(t_j, p_i) - I(t_j, p_i)), 1 \le i \le k, \quad (5)$$

i.e., if place p_i is an input (output) place of transition t_j then transition t_j is removing (adding) tokens from (to) place p_i according to the multiplicities given by the function I (O).

Given the initial state, the ODEs in (5) can be solved by numerical integration for which various reliable tools exist. There is one equation per place and for this reason the computational complexity grows linearly with the number of regions while in case of analyzing the underlying CTMC the growth would be exponential.

IV. Case Study

In this section we present a case study to illustrate our methodology. The case study models application spreading in a mobile environment. In the following, first, we describe the communication model over which application spreading takes place. Second, we provide the SPN we use to model application spreading in a single geographical region. Then, we build the complete model by replicating the SPN for every region and by defining the mobility among the regions. Finally, we investigate the transient behavior of the model and show that the approximate results obtained by the mean field approach are reliable estimates of the average behavior.

A. Communication Model

The communication model we apply is a simplified version of the one provided in [5]. In this model, since the users who are not interested in the application do not influence the spreading process, we consider only those users that are eventually ready to purchase or at least to use the application under study.

The users form ad hoc networks from time to time and communicate directly between each other. The application is a multi-user application and it has two versions, a trial and a full version. If there are users possessing the full version of the application (*seeds*[1]) then the others can download the trial version of the application directly from the seeds. Later, they can also purchase the full version of the application.

Users possessing the trial version (*leeches*[1]) suffer restrictions in the usage of the application. These restrictions, which are motivating the users to purchase the application, are caused by the fact that only a limited number of leeches can connect to one seed simultaneously. This limit will be referred to as the *leech limit*. In this sense, seeds can be considered as servers which serve only a limited number of users (leeches).

Furthermore, we distinguish two types of users based on their behavior. $Type_A$ users are interested in using the application and are willing to purchase it if they like it. $Type_B$ users instead will never purchase the application but still they influence the spreading process because they decrease the probability that a $Type_A$ user finds an available seed to connect to.

Fig 1 shows a possible scenario of the application usage. We assume that the leech limit is one and hence only one leech (the light laptop) can connect to the only available seed (the PDA) while the others (the dark laptops) have to wait. As soon as one of them purchases the full version of the application, they too can use it.

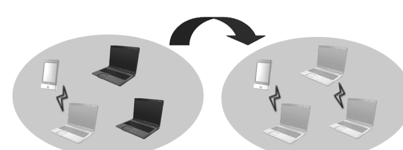

Fig. 1. Change of application usage when a user purchases the application

B. The Single Region Model

A stochastic Petri net is appropriate to model the stochastic process arising from the communication model described above. The SPN of a single region is shown in Fig. 2. A user is represented by a single token in a given place depending on the user's type ($Type_A$ or $Type_B$) and on the version of the application possessed by the user (seed or leech). Initially, each user is a leech and only $Type_A$ users can change their state to seed by purchasing the application. Each user is either in active or in passive state, depending on whether he is currently using the application or not. The passive $Type_A$ users are represented in place Pass_A, while

[1]After the terminology of Bit-Torrent.

the active $Type_A$ users in place Act_A. Similarly, the passive and active $Type_B$ users are represented in place Pass_B and place Act_B, respectively, while the passive and active seeds in place Pass_seeds and place Act_seeds, respectively. We count the number of purchases of the application in place Purchases.

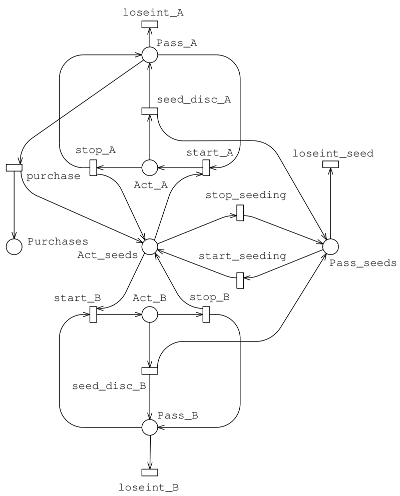

Fig. 2. The Petri net model of a single region

Leeches can change their state to active only if at least one free seed is available (for simplicity, the leech limit is one in this case study), while each user can change his state to passive in the active states. If an active seed changes his state to passive when a leech is connected to him, the state of that leech changes to passive. A $Type_A$ user can change his state to seed by purchasing the application. Since application purchase is not bounded to the ad hoc network, it can be done in passive state.

The users can loose their interest in using the application when they are in passive state. The uninterested users leave the system and will not be interested again. Therefore, there will eventually be a state when every user will have left the system and will there be not enabled transitions. If so, the life cycle of the given application has terminated. The transitions are listed and described in Table I.

C. The Composition of Single Regions

To model four regions we use four copies of the SPN depicted in Fig. 2 and model mobility among them by additional transitions as described in Section III-A. Fig. 3 shows the topology built of the four regions.

$R1$, $R2$, $R3$ and $R4$ denote the single regions, while mv_12, mv_21, mv_13, mv_31, mv_24, mv_42, mv_34 and mv_43 are the intensities of the mobility among them. Every arrow in Fig. 3 corresponds to transitions with the same firing intensity moving tokens from one region to another. We assume that only passive nodes move and, consequently, only those token can leave a region that are in places Pass_A, Pass_B or Pass_seeds. The state of the moving node in the destination region is the same as it was in the region where it comes from. Accordingly, there is, for example, a transition moving tokens from place Pass_B of $R2$ to place Pass_B of $R4$ whose intensity is mv_24.

TABLE I
DESCRIPTION OF THE TRANSITIONS OF THE SPN MODELING A REGION

Transitions	Description
start_A / start_B	A $Type_A$ / $Type_B$ user starts to run the application. It can fire only if at least one seed is available in the network.
start_seeding	A seed starts to run the application.
stop_A / stop_B	A $Type_A$ / $Type_B$ user stops running the application. The seed to which he was connected will be available for other leeches to connect to.
seed_disc_A / seed_disc_B	A seed stops running the application when one $Type_A$ / $Type_B$ user was connected to him. This user changes his state to passive, as well.
stop_seeding	A seed stops running the application when no leech was connected to him.
loseInt_A / loseInt_B / loseInt_seed	A $Type_A$ user / $Type_B$ user / seed loses the interest in using the application.
purchase	A $Type_A$ user purchases the application. He becomes an active seed.

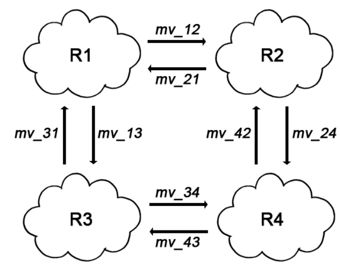

Fig. 3. The model of the mobility among the regions

D. Transient analysis of the model

In this section we illustrate the kind of results that one can obtain by applying the mean field approach to the complete model. In particular, we show that even with a low number of tokens the results obtained from the ODEs are good approximations of the average behavior of the model. As the number of tokens is increased the variability of the behavior of the model is decreasing and hence the ODEs capture well not only the average but the overall behavior of the system.

The number of states of the model is huge even with a low number of tokens. For this reason we used simulation to analyze the model and compared the results obtained from the ODEs against the results obtained from simulation. The simulation runs were generated using GreatSPN [11] and we used 20000 runs to compute the mean. The firing intensities of the transitions are collected in Table II.

We evaluated three scenarios that differ only in the initial marking. In scenario one, the initial marking is 1 token in place Pass_A and 1 token in place Pass_B in every region (i.e., there are eight users in the system, initially two in each region). We put 10 tokens in the same places in scenario 2 and 50 in scenario 3. We depicted the average number of purchases in Fig. 4 and, in order to illustrate the mobility

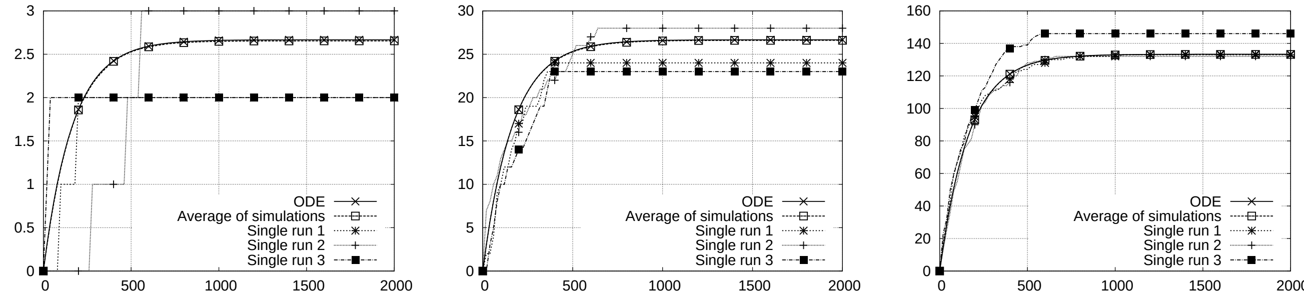

Fig. 4. Total number of purchases as function of elapsed time for scenario 1,2 and 3

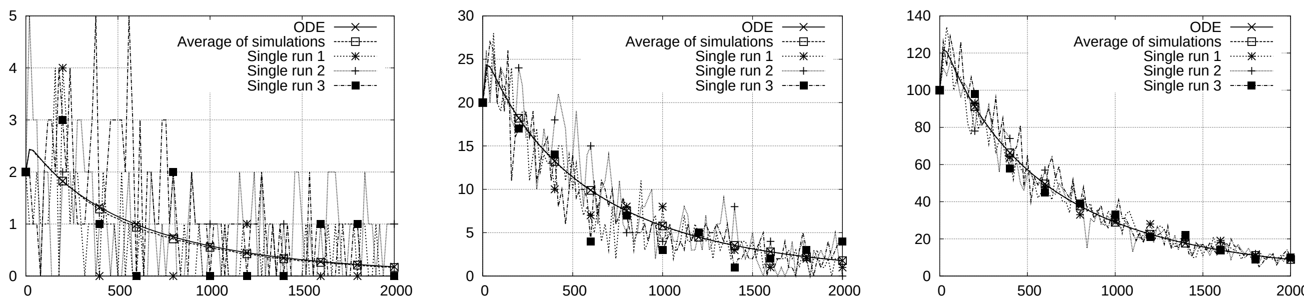

Fig. 5. Number of users in region R2 as function of elapsed time for scenario 1,2 and 3

TABLE II
THE FIRING INTENSITIES OF THE TRANSITIONS

Transitions	Value [1/hour]
start_A / start_B / start_seeding	0.04
stop_A / stop_B / stop_seeding	0.9
seed_disc_A / seed_disc_B	0.9
loseInt_A / loseInt_B	0.002
loseInt_seed	0.001
purchase	0.004
mv_12	0.07
mv_21, mv_42	0.06
mv_13	0.03
mv_31, mv_34	0.05
mv_24, mv_43	0.04

present in the model, the average number of users in region R2 in Fig. 5. In both figures we depicted three individual simulation runs as well. The results obtained by the ODEs are precise approximations of the average behavior in all cases (only in scenario 1, and for what concerns the number of users in region R2, one can observe some differences between the result obtained by the ODEs and the mean obtained by simulation). Moreover, as the number of users is increased, it is less and less probable that an individual simulation run is far from the behavior predicted by the ODEs. For all cases, the analysis is much faster by the ODEs (order of seconds) than by simulation (order of hours).

V. Conclusions

In this paper, we have proposed a methodology for the analysis of ad hoc networks with mobility. This methodology is based on the compositional construction of the model and its approximate analysis by the mean field approach. We have illustrated the applicability of the model on a case study. Future work must include the extension of the approach for Petri nets with more general characteristics like inhibitor arcs and general server policies.

VI. Acknowledgements

This work has been partially supported by the Hungarian Scientific Research Fund (OTKA, PD 72984).

References

[1] A. Kamat and R. Prakash, "Effects of link stability and directionality of motion on routing algorithms in manet," in *Proceedings of 9th Int. Conference on Computer Communications and Networks*, pp. 380–385, 2004.

[2] M. Feeley, N. Hutchinson, and S. Ray, "Realistic Mobility for Mobile Ad Hoc Network Simulation," in *Ad-Hoc, Mobile, and Wireless Networks* (I. Nikolaidis, M. Barbeau, and E. Kranakis, eds.), vol. 3158 of *Lecture Notes in Computer Science*, pp. 630–630, Springer Berlin / Heidelberg, 2004.

[3] A. B. Mcdonald and T. Znati, "A path availability model for wireless ad-hoc networks," in *Proc. IEEE WCNC*, pp. 35–40, 1999.

[4] O. Dousse, P. Mannersalo, and P. Thiran, "Latency of wireless sensor networks with uncoordinated power saving mechanisms," in *in Proceedings of Mobihoc, 2004*, pp. 109–120, 2004.

[5] Á. Horváth and K. Farkas, "Modeling application spreading using mobile ad hoc networks," in 3^{rd} *Joint IFIP Wireless Mobile Networking Conference*, (Budapest, Hungary), 2010.

[6] G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, *Modelling with Generalized Stochastic Petri Nets*. John Wiley & Sons, 1995.

[7] S. Bernardi, S. Donatelli, and A. Horváth, "Implementing compositionality for stochastic petri nets," *International Journal on Software Tools for Technology Transfer*, vol. 3, no. 4, 2001.

[8] W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1995.

[9] T. G. Kurtz, "Solutions of ordinary differential equations as limits of pure jump Markov processes," *Journal of Applied Probability*, vol. 1, no. 7, pp. 49–58, 1970.

[10] M. Tribastone, *Scalable Analysis of Stochastic Process Algebra Models*. PhD thesis, School of Informatics, University of Edinburgh, 2010.

[11] S. Baarir, M. Beccuti, D. Cerotti, M. De Pierro, S. Donatelli, and G. Franceschinis, "The greatspn tool: recent enhancements," *SIGMETRICS Perform. Eval. Rev.*, vol. 36, no. 4, pp. 4–9, 2009.