Exploiting non-deterministic analysis in the integration of transient solution techniques for Markov Regenerative Processes

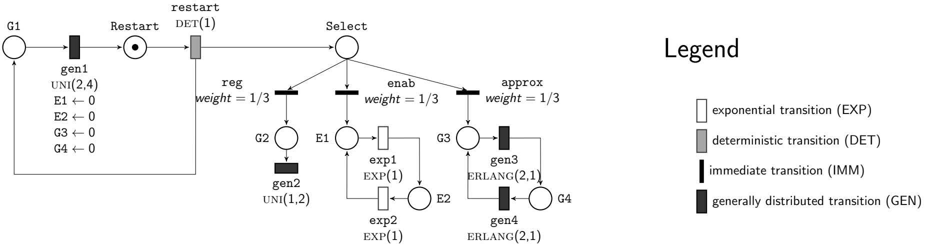
Marco Biagi[1], Laura Carnevali[1], Marco Paolieri[2],
Tommaso Papini[1] and Enrico Vicario[1]

[1] University of Florence, Italy
[2] University of Southern California, USA

QEST
Berlin, September 5, 2017

- This is about:
  - transient analysis of Markov Regenerative Processes
  - application of a distinct technique to each regenerative epoch
  - a novel iterative technique for approximated analysis

Stochastic Time Petri Net (STPN)



- a variant of Stochastic Petri Nets(SPN) with generally distributed durations (GEN), possibly with bounded support
- STPN can be seen as a Time Petri Net (TPN) decorated with stochastic parameters

Stochastic Time Petri Net: marking process

- **Marking process**: $\{M(t), t \geq 0\}$,
  - $M(t)$ is the marking at time t
  - specifies the logical state of an STPN at each time instant

- **Markov Regenerative Process (MRP)**: always, with probability 1, the process eventually reaches a regeneration
- **Regeneration**: a state that provides sufficient information to characterize the future evolution of the system (not affected by its previous history)

- In a STPN, a regeneration is a marking where all enabled GEN transitions are newly enabled

Markov Regenerative Processes: transient solution

- $\pi_{ij}(t) := P\{M(t) = j \mid M(0) = i\}$
- **Transient evolution** completely characterized by:
  - initial states probabilities
  - local kernel $L_{ij}(t) := P\{M(t) = j, T_1 > t \mid M(0) = i\}$
  - global kernel $G_{ik}(t) := P\{M(T_1) = k, T_1 \leq t \mid M(0) = i\}$
- **Markov renewal equations**:

$$\pi_{ij}(t) = L_{ij}(t) + \sum_{k \in \Theta} \int_0^t \frac{dG_{ik}(x)}{dx} \pi_{kj}(t - x)\, dx$$

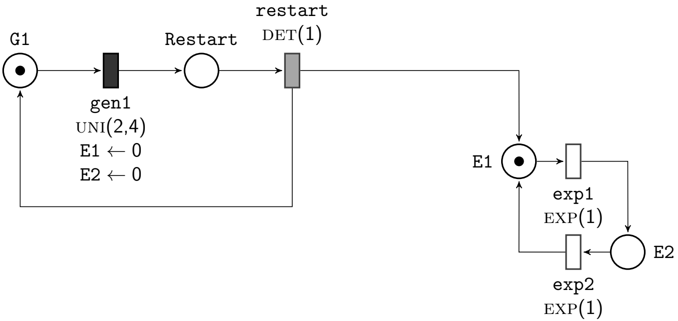
Contributions

- **integration of different techniques**:
  exploiting state class graph of the underlying TPN
  - identify regeneration points
  - identify states visited between regenerations (**regenerative epoch**)
  - identify which solution technique apply to each regenerative epoch

- **iterative approximate technique**:
  - adaptive approximation of kernel entries
    based on partial exploration of the state space
  - heuristics to reduce the error on transient probabilities

1/3: Analysis under enabling restriction [1]

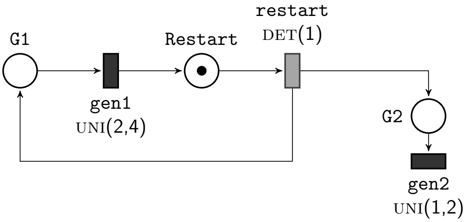
- **enabling restriction**: at most a GEN timer enabled in each state
- kernel rows computed analyzing the CTMC
  subordinated to the activity interval of the active GEN

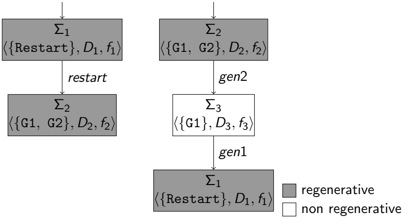


[1] German et al. "Transient analysis of Markov regenerative stochastic Petri nets: A comparison of approaches." Petri Nets and Performance Models, 1995

2/3: Analysis with stochastic state classes [2]

- **bounded regeneration restriction**: a regeneration is always reached within a bounded number of discrete events
- **stochastic state class**: $S = \langle m, D, f \rangle$
  - m: is the marking
  - D: support of enabled transitions remaining ttf
  - f: joint pdf of enabled transitions remaining ttf
- **decomposed state space**: kernel rows computed
  enumerating *stochastic transient trees*
  of reached classes between two subsequent regenerations

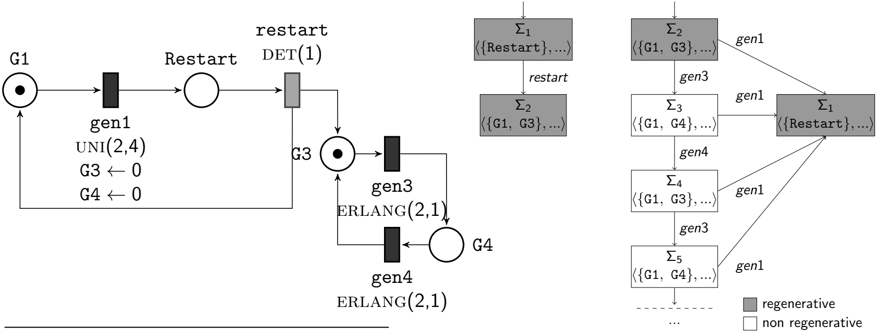




[2] Horváth et al., "Transient analysis of non-Markovian models using stochastic state classes". Performance Evaluation 2012

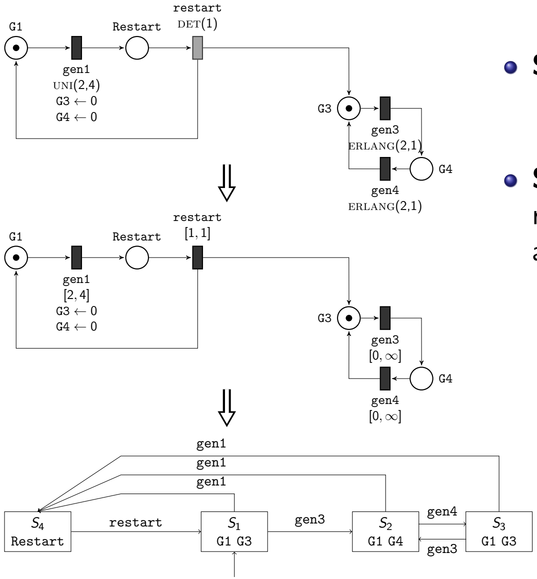
3/3: Approximated analysis with stochastic state classes [3]

- **not restricted**: epochs that break both the restrictions
- kernel rows defectively approximated truncating stochastic transient trees



[3] Horváth et al., "Transient analysis of non-Markovian models using stochastic state classes". Performance Evaluation 2012

Integration of different techniques: non-deterministic analysis



- **State class**: $S = \langle m, D \rangle$
  - m is the marking
  - D support of the remaining ttf
- **State Class Graph (SCG)**: represents reachability relation among state classes

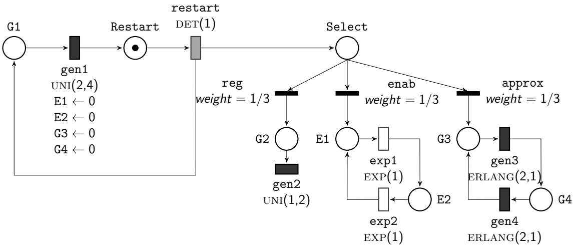
Integration of different techniques: regeneration classification

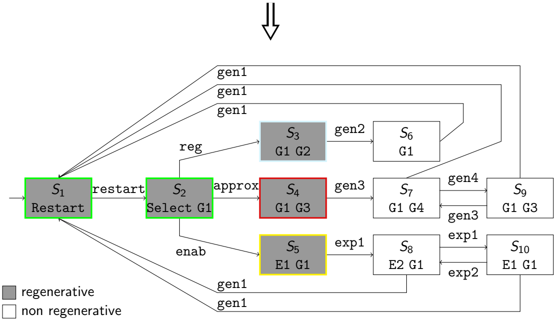
- regenerations identified in the SCG from newly enabled transition
- **regenerative epochs**: TPN state space can be decomposed in a set of SCGs, each rooted in a regeneration and containing all non-regenerative successors reached before any regeneration

Regenerative epoch classification:

- *enabling restriction*: at most one GEN transition is enabled
- *bounded regeneration restriction*: no cycles present

Classification example





5 regenerative epochs
(5 kernel rows) rooted in:

- S_1: both restriction
- S_2: both restriction
- S_3: bounded restriction
- S_4: no restriction
- S_5: enabling restriction

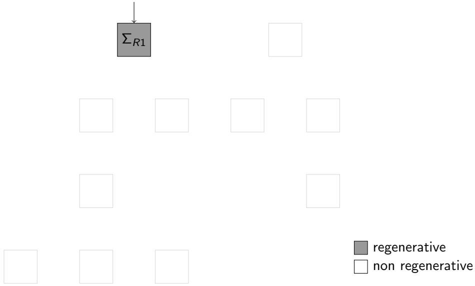
Iterative approximate technique: basic concept

- approximation required for epochs that do not satisfy any restriction
- approximation derived truncating transient stochastic classes tree, obtaining a defective measure of transient probabilities $\tilde{\pi}_{ij}(t)$
- a good approximation technique should:
  - roughly truncate epochs visited rarely
  - analyze in details epochs visited more often

Iterative approximate technique: algorithm

Iterative algorithm:

1. expand at most ν_{start} nodes for each *non restricted epoch*
2. identify the truncated node Φ with the largest estimated probability to be reached (based on steady state analysis of an embedded DTMC)
3. expand at most ν_{iter} nodes of successors of Φ
4. if at least ν_{max} nodes were expanded stop expansion, otherwise return to 2



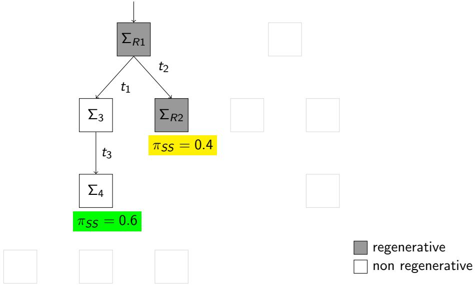
Example parameters:

- $\nu_{start} = 3$
- $\nu_{iter} = 3$
- $\nu_{max} = 9$

Iterative approximate technique: algorithm

Iterative algorithm:

1. expand at most ν_{start} nodes for each *non restricted epoch*
2. identify the truncated node Φ with the largest estimated probability to be reached (based on steady state analysis of an embedded DTMC)
3. expand at most ν_{iter} nodes of successors of Φ
4. if at least ν_{max} nodes were expanded stop expansion, otherwise return to 2



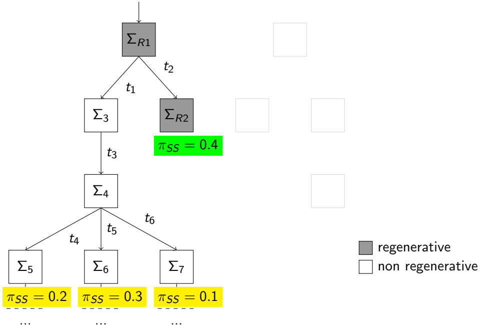
Example parameters:

- $\nu_{start} = 3$
- $\nu_{iter} = 3$
- $\nu_{max} = 9$

Iterative approximate technique: algorithm

Iterative algorithm:

1. expand at most ν_{start} nodes for each *non restricted epoch*
2. identify the truncated node Φ with the largest estimated probability to be reached (based on steady state analysis of an embedded DTMC)
3. expand at most ν_{iter} nodes of successors of Φ
4. if at least ν_{max} nodes were expanded stop expansion, otherwise return to 2



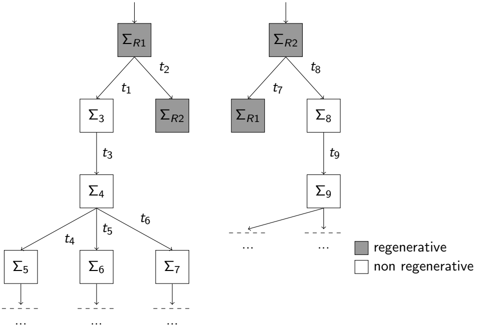
Example parameters:

- $\nu_{start} = 3$
- $\nu_{iter} = 3$
- $\nu_{max} = 9$

Iterative approximate technique: algorithm

Iterative algorithm:

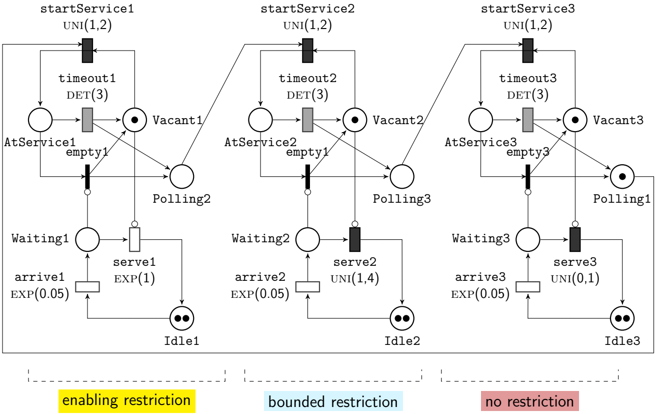
1. expand at most ν_{start} nodes for each *non restricted epoch*
2. identify the truncated node Φ with the largest estimated probability to be reached (based on steady state analysis of an embedded DTMC)
3. expand at most ν_{iter} nodes of successors of Φ
4. if at least ν_{max} nodes were expanded stop expansion, otherwise return to 2



Example parameters:

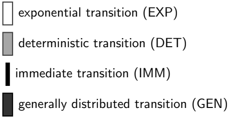
- $\nu_{start} = 3$
- $\nu_{iter} = 3$
- $\nu_{max} = 9$

Case study: a polling system [4]



- 3 stations
- exhaustive
- DET timeout

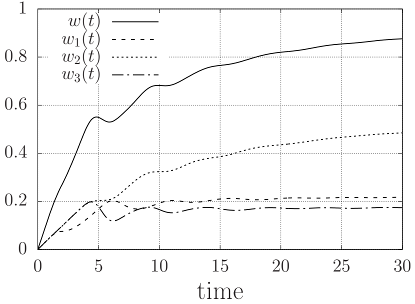
Legend



[4] Trivedi et al. "Stochastic Petri net models of polling systems." IEEE Journal on Selected areas in Communications(1990)

Case study: a polling system

- experiments implemented using Oris tool API[5] [6]
- 135 regenerations (99 bounded, 18 enabling, 16 no restriction)
- $t_{limit} = 30s$ and $t_{step} = 0.1s$

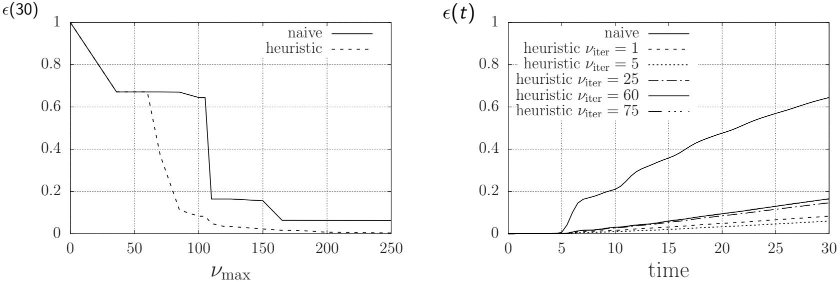


- $w(t)$ = average number of messages waiting to be served at time t in the overall system
- $w_n(t)$ = average number of messages waiting to be served at time t in station n

[5] Carnevali et al. "A framework for simulation and symbolic state space analysis of non-Markovian models." Computer Safety, Reliability, and Security (2011)

[6] http://www.oris-tool.org/

- **naive**: same number of node expanded for each non restricted regenerative epoch
- **heuristic**: regenerative epochs visited more often are analyzed more in detail (iterative approximate technique)



(a) **(b)**

- approximation error at time t: $\epsilon(t) = 1 - \sum_{i,j \in markings} \tilde{\pi}_{ij}(t) > 0$

Conclusions

- non-deterministic analysis exploited to drive the integration of different techniques in the evaluation of the kernels of an MRP
- novel iterative approximated technique for kernel evaluation
- accurate results achieved by the heuristic-based approximate analysis while maintaining a moderate computational cost

Future directions:

- open to integration with other analytical/simulative technique
- open to integration of other heuristics for iterative approximated analysis e.g.: considering the mean sojourn time in epochs