Zastosowanie sieci HTCPN i QPN do modelowania generatorów strumieni ruchu wejściowego

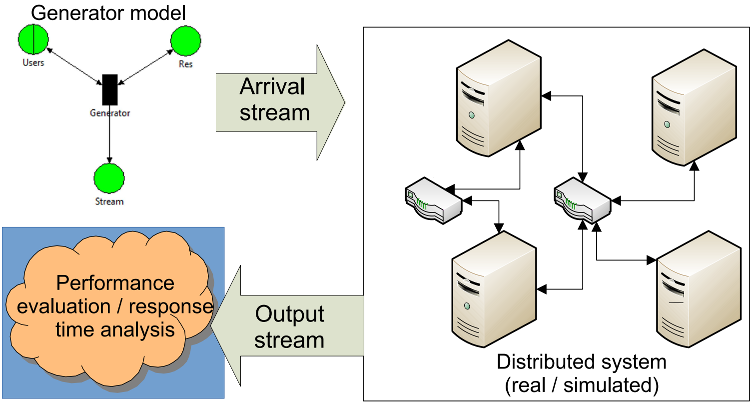
Tomasz RAK, Dariusz RZOŃCA

Department of Computer and Control Engineering
Rzeszow University of Technology

RZESZOW UNIVERSITY
OF TECHNOLOGY

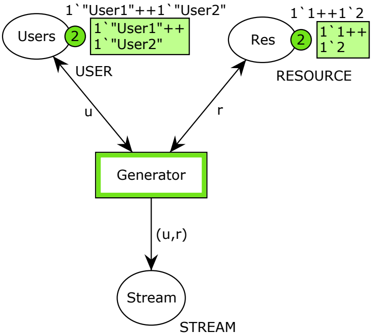
TOC

System Overview



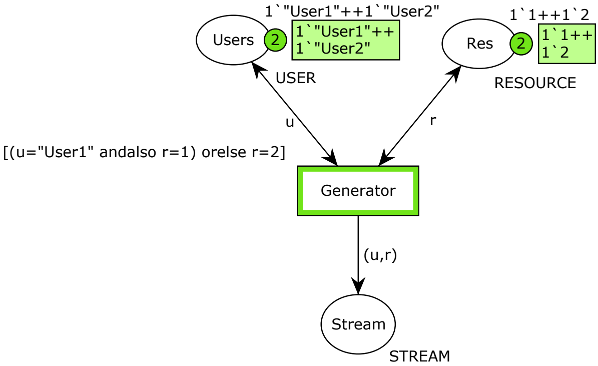
- Arrival stream generator is modelled in HTCPN / QPN

Simple Generator



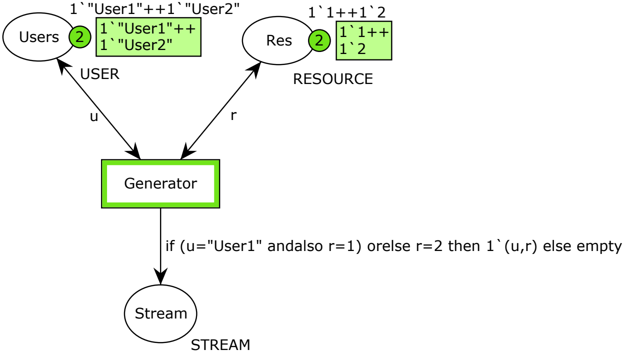
- The stream of generated tokens reflects the mapping between users and resources
- Two users, two resources, no constraints

Generator with Constraints in the Guard



- Inscription in the guard restrict the mapping
- User1 may use Res1 or Res2, User2 only Res2

Generator with Constraints in the Arc

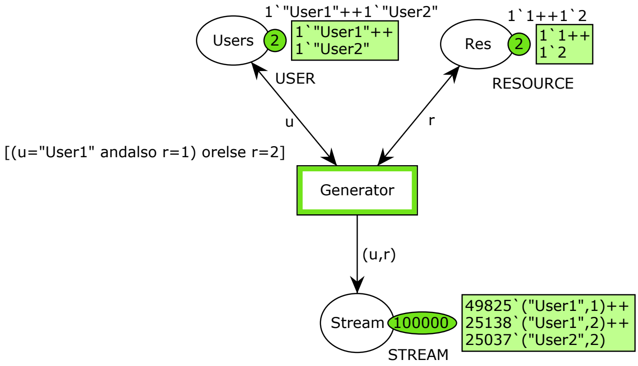


- Inscription in the arc restrict the mapping
- User1 may use Res1 or Res2, User2 only Res2

Comparison of the Models

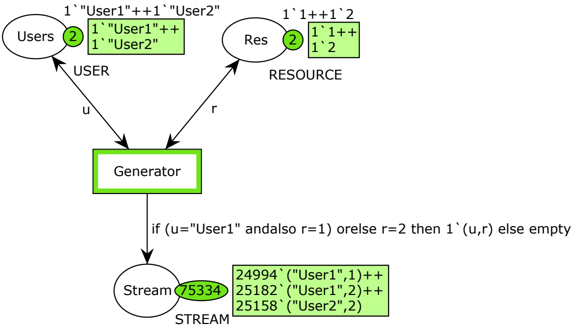
- Both models are functionally identical, in both of them User1 may use Res1 or Res2, User2 only Res2
- The models impose similar constraints, but in different way. Will they behave identically during simulation?
- What kind of distribution in the generated stream (user/resource pairs) is expected in each case?

Simulation Results – Generator 1 (Guard)



- Transition has been fired 100k times
- One of the generated pairs occurs more frequently than others

Simulation Results – Generator 2 (Arc)



- Transition has been fired 100k times, but only ∼75k tokens are produced
- Distribution of each pair is equal

Comparison of the Models

- HTCPN formalism allows for definition which tokens will be produced, but not for precision set of their distribution, it depends on the implementation (simulator)
- Fairness in the CPN Tools:
  "... we also need to ensure that the search algorithm is, in some sense, fair so that every enabled binding has a non-zero probability of being used ..."
  Authors of the CPN Tools
- Simulator by design does not guarantee that every enabled binding has equal probability of being selected, but ensure, that every enabled binding can possibly be used at all

Motivation for Checking Another Formalism

- In HTCPN networks complex models of the generator (numerous users, many resources, complicated constraints), precise shaping of the distribution of the arrival stream is difficult to achieve and may be ambiguous (it depends on the simulator)
- It is advisable to check another formalism
- Will the QPN networks allow for flexible and precise generation of the input stream with the expected distribution?

Workload Generators

Different approaches[a]

[a] **Rak, T.; Rzońca, D.** : Recommendations for Using QPN Formalism for Preparation of Incoming Request Stream Generator in Modeled System, (**2021**) doi: 10.3390/app112311532
Curiel, M.; Pont, A.:Workload Generators for Web-Based Systems: Characteristics, Current Status, and Challenges, (**2020**) 10.1109/COMST.2018.2798641
Kolbusz, J.; Paszczynski, S.; Wilamowski, B.: Network traffic model for industrial environment, (**2019**) doi:10.1109/INDIN.2005.1560411
Lukichev, M.: Formation of equivalent simulation model of an real-time video stream generator used in packet-oriented communication networks, taking into account the structure of the H.264 compression algorithm, (**2019**)
Rzońca, D., Rząsa, W., Samolej, S.: Consequences of the Form of Restrictions in Coloured Petri Net Models for Behaviour of Arrival Stream Generator Used in Performance Evaluation, (**2018**)
Tang, W.; Fu, Y.; Cherkasova, L.; Vahdat, A.: Modeling and generating realistic streaming media server workloads, (**2007**) doi:10.1016/j.comnet.2006.05.003

Users and Resources

The model allows us to freely join the $u1$ or $u2$ user tokens with the $r1$ or $r2$ resource tokens. Thus, any distribution in *Stream* can be obtained in this way.

The logic lies in the firing of transitions with weights and handling of the token colors. The following cases are considered:

- In a simple SG, any user can use any resource in the stream generation process.
- In other cases, any stream can be generated based on a predetermined weight value.

We get the populations with the expected distribution of the output stream (approximately):

- Simple model $(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$;
- Varied models, e.g., $(\frac{1}{10}, \frac{1}{5}, \frac{3}{10}, \frac{2}{5})$; $(\frac{1}{7}, \frac{2}{7}, \frac{1}{7}, \frac{3}{7})$; $(\frac{1}{4}, \frac{1}{4}, 0, \frac{1}{2})$; $(\frac{1}{5}, \frac{1}{5}, 0, \frac{3}{5})$; $(0, \frac{1}{4}, 0, \frac{3}{4})$; $(0, 0, \frac{1}{4}, \frac{3}{4})$; $(0, 0, 0, 1)$.

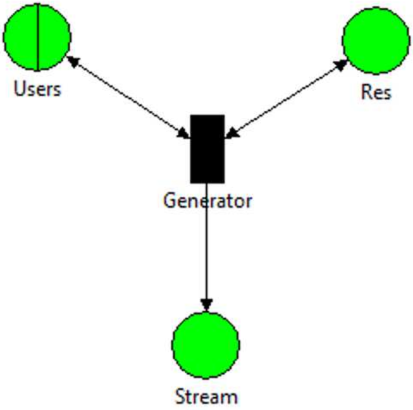
Mathematical Model of SG in QPN

$$QPN_{SG} = (PL, TR, CO, IN, MA, QU, WE) \quad (1)$$

where:

- Set of places $PL = \{Res, Stream\}$.
- Set of transitions $TR = \{Generator\}$.
- Color function $CO(p_i)$ for colors $c = \{u1, u2, r1, r2, u1r1, u1r2, u2r1, u2r2\}$. where:
    - $u1$ and $u2$-user-classes;
    - $r1$ and $r2$-resources;
    - $u1r1$, $u1r2$, $u2r1$, $u2r2$-stream.
- Incidence functions specify the interconnections between places and transitions $IN(p, t)$.
- Initial marking specify how many tokens are contained in each place $MA(p) = \{Users(u1 = 1, u2 = 1), Res(r1 = 1, r2 = 1), Stream(u1r1 = u1r2 = u2r1 = u2r2 = 0)\}$.
- $QU = (QU_1, QU_2, (-/M/\infty/IS_{Users}, null, null))$, where:
    - $QU_1 = \{Users\}$;
    - $QU_2 = \varnothing$;
- $WE = (WE_1, WE_2, W)$, where:
    - $WE_1 = \varnothing$;
    - $WE_2 = TR$, where the *Generator* transition is immediate;
    - $W = (w(mode_1), w(mode_2), w(mode_3), w(mode_4))$, for simple model $W = (1, 1, 1, 1)$ and for exemplary varied models $W = (1, 2, 3, 4)$, $W = (1, 2, 1, 3)$, $W = (1, 1, 0, 2)$, $W = (1, 1, 0, 3)$, $W = (0, 1, 0, 3)$, $W = (0, 0, 1, 3)$, $W = (0, 0, 0, 1)$, where:
        - Color function $\forall c \in CO(t_j) : w_j(c) \in \mathbb{R}^+$;
        - Colors $c = \{mode_1, mode_2, mode_3, mode_4\}$;
        - *Generator* transition modes $mode_1, \ldots, mode_4$.

SG Model



Firing Modes of *Generator* Transition

Transition	Mode	Action
Generator	$mode_1$	input$\{u1\}$ $\longrightarrow$ output$\{u1r1\}$ input$\{r1\}$ $\longrightarrow$ output$\{u1r1\}$
Generator	$mode_2$	input$\{u1\}$ $\longrightarrow$ output$\{u1r2\}$ input$\{r2\}$ $\longrightarrow$ output$\{u1r2\}$
Generator	$mode_3$	input$\{u2\}$ $\longrightarrow$ output$\{u2r1\}$ input$\{r1\}$ $\longrightarrow$ output$\{u2r1\}$
Generator	$mode_4$	input$\{u2\}$ $\longrightarrow$ output$\{u2r2\}$ input$\{r2\}$ $\longrightarrow$ output$\{u2r2\}$

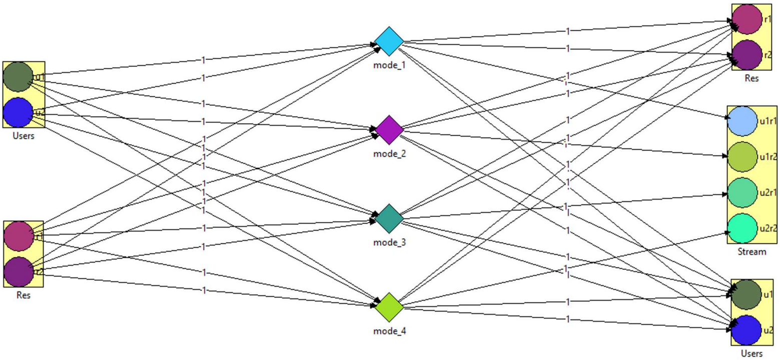
Places Used in the Model/Token Colors

Place	Token	Queue	Description
Users	$\{u1, u2\}$	$-/M/\infty/IS$ [a]	Queueing place used to model concurrent clients.
Res	$\{r1, r2\}$	null	Ordinary place used to model resource.
Stream	$\{u1r1, u1r2, u2r1, u2r2\}$	null	Ordinary place used to model the generated stream.

[a] Different requests interarrival time/exponential distribution of the request service time/the maximum number of requests that can arrive in a queue/the Infinite Server scheduling strategy.

Token Color	Place	Description
$u1$ $u2$	*Users*	First user Second user
$r1$ $r2$	*Res*	First resource Second resource
$u1r1$ $u1r2$ $u2r1$ $u2r2$	*Stream*	First user and first resource First user second resource Second user and first resource Second user and second resource

Incidence Function



Modes

Name	Real Color	Firing Weight
mode_1		1
mode_2		2
mode_3		1
mode_4		3

Token Colors (Simple Model)

Modes	Firing Weight w	Token Color	Mean Token Population for *Stream*
$mode_1$	1.0	$u1r1$	25,172.11
$mode_2$	1.0	$u1r2$	24,858.411
$mode_3$	1.0	$u2r1$	24,733.689
$mode_4$	1.0	$u2r2$	25,095.054

Modes	Firing Weight w	Token Color	Mean Token Population for *Stream*
$mode_1$	1.0	$u1r1$	9,978.882
$mode_2$	2.0	$u1r2$	19,827.981
$mode_3$	3.0	$u2r1$	29,978.301
$mode_4$	4.0	$u2r2$	39,679.93
$mode_1$	1.0	$u1r1$	14,195.611
$mode_2$	2.0	$u1r2$	28,625.108
$mode_3$	1.0	$u2r1$	14,308.058
$mode_4$	3.0	$u2r2$	42,688.806
$mode_1$	1.0	$u1r1$	24,931.845
$mode_2$	1.0	$u1r2$	25,170.6
$mode_3$	0.0	$u2r1$	0.0
$mode_4$	2.0	$u2r2$	50,286.744
$mode_1$	1.0	$u1r1$	20,154.555
$mode_2$	1.0	$u1r2$	19,958.521
$mode_3$	0.0	$u2r1$	0.0
$mode_4$	3.0	$u2r2$	60,068.627
$mode_1$	0.0	$u1r1$	0.0
$mode_2$	1.0	$u1r2$	24,933.095
$mode_3$	0.0	$u2r1$	0.0
$mode_4$	3.0	$u2r2$	75,169.879
$mode_1$	0.0	$u1r1$	0.0
$mode_2$	0.0	$u1r2$	0.0
$mode_3$	1.0	$u2r1$	24,894.468
$mode_4$	3.0	$u2r2$	75,242.374
$mode_1$	0.0	$u1r1$	0.0
$mode_2$	0.0	$u1r2$	0.0
$mode_3$	0.0	$u2r1$	0.0
$mode_4$	1.0	$u2r2$	99,950.075

Concluding Remarks

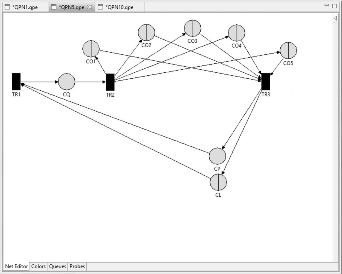
- The QPME tool generates a report showing the predicted SG population for the individual model configuration.
- It can be observed that the quantities of the generated tokens in the *Stream* place for each mode i are consistent with the appropriate firing weights w.

$$\frac{tokens_{mode_i}}{\sum_{j=1}^{n} tokens_{mode_j}} \approx \frac{w(mode_i)}{\sum_{j=1}^{n} w(mode_j)} \tag{2}$$

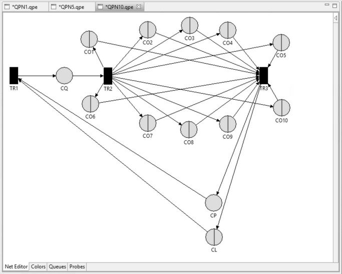
- Various distributions of the tokens in the stream can easily be modeled by appropriate changes in the firing weights, according to the designer's needs.

Rak, T.; Rzońca, D. Recommendations for Using QPN Formalism for Preparation of Incoming Request Stream Generator in Modeled System. Appl. Sci. 2021, 11, 11532. https://doi.org/10.3390/app112311532

Future Work



QPN5



QPN10