Class Based Queuing Model (CBQ-Model)

1 Description of the CBQ-Model

Before we start with the description of the model we will provide some basic information about the notation of Petri nets within the APNN-Toolbox. Places are presented as circles and transitions as boxes. Two types of transitions are supported by the APNN-Toolbox: timed and immediate. Timed transitions are represented by green outlined boxes and immediate transitions are represented by filled black boxes. Furthermore the APNN-Toolbox supports hierarchical modelling. Submodels are represented as double circles in the higher model and can be used like places. This allows the connection with transitions, which makes the transitions available in the submodel, where they are used to connect the submodel with the higher model.

The CBQ-Model is divided into five parts. The first part interconnects the other four parts and we will refer to it as the central part. The second part models a phase-type arrival process. Each of the remaining three parts models the handling of incoming packets of one packet class.

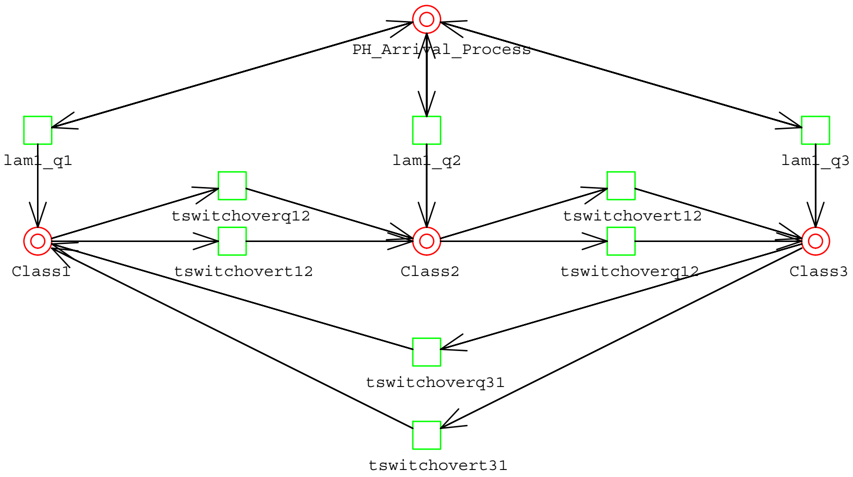

Figure 1: Class based queuing model

The central part controls the flow of tokens between the four submodels and can be seen in Figure 1. By doing this it fulfils two tasks. The first one is to transfer the tokens between the phase-type arrival process and the three other submodels. This is achieved through the transitions lam1_q1 to lam1_q3. The second task is to determine which of the queue models should be served next. This is done via a round robin solution. Note that the decision if the queue may be served is made in the respective submodel. In this circle two tokens move from one queue model to the next. For example if the queue of packet class 1 which is named "Class1" has finished the token moves through tswitchovert12 and tschwitchoverq12 to the queue of packet class 2 which is named "Class2". Note that the token that moves through the transitions starting with "tswitchovert" is used to determine if a queue should be served and the token that moves through the transitions starting with “tswitchoverq” is used to activate the actual service and guarantee that only one queue is served at a time. The transitions used in the main model are timed transitions with a negative exponential distribution of 10.000. Theoretically immediate transitions would fit the model better but due to technical issues timed transitions have to be used.

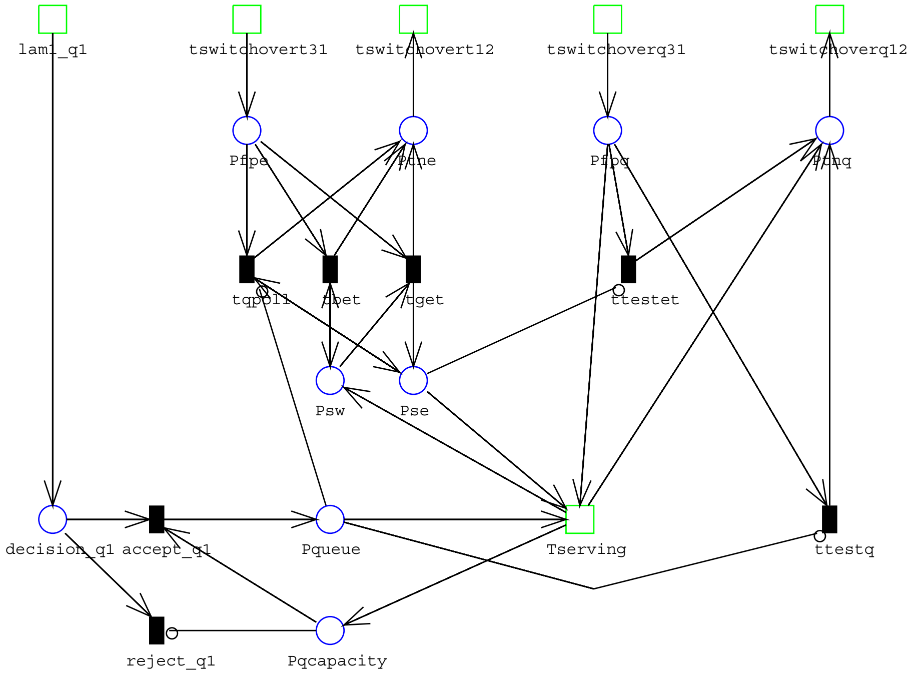

Figure 2: Submodel of queue 1

The submodels for all queues are basically identical so we will only describe the model of “Class1” which is displayed in Figure 2. Note that the numbers in the names are different for each queue model and refer to the number of the respective packet class. The transitions on the top of Figure 2 are the global transitions which you have already seen in Figure 1. As we are dealing with a queuing model we will start with the description of the queue which is represented via the place “Pqueue”. Tokens can only move to this queue if transition “accept_q1” is enabled. This is only the case if there are tokens in the places “decision_q1” and “Pqcapacity”. “decision_q1” is followed by two immediate transitions, where assumed that there is a token in “decision_q1” exactly one is enabled. This situation models the following aspect. Either the queue has not reached its maximum capacity which means there is at least one token in “Pqcapacity” and the arriving packet is moved into the queue or the queue is full and the incoming packet is rejected. The sum of tokens in “Pqueue” and “Pqcapacity” is always 10. So the queue cannot be filled with more than 10 tokens which correspond to the size of the buffer.

Tokens can move from “Pqueue” back to “Pqcapacity” through transition “Tserving” which models the situation that a packet which is in the queue is processed. As you can see there are several conditions that have to be met before “Tserving” is enabled. We can divide these into two parts. The first part starts with “tswitchovert23” and ends with “tswitchovert31”. The other part starts with “tswichoverq23” and ends with “tswitchoverq31” models. As we have mentioned before these parts deal with the decision if a queue may be served and the actual serving.

We will now deal with the first part. A token enters the submodel through “tswitchover23” and is positioned at place “Ptpe”. There it has three possibilities. The first one is to move through transition “tqpd” which is enabled if the queue is empty and the model may be served when a token arrives in the queue. This allows the token to move directly to place “Ptne” and leave the submodel. This transition is only enabled if the other two transitions are disabled. These transitions are either both enabled or disabled but before we can discuss these transitions we have to deal with the places “Psw” and “Pse”. A token in place “Psw” means that the queue is currently in waiting state. A token in place “Pse” means that the queue is enabled to be served. The sum of tokens in places “Psw” and “Pse” is always 1. Now it easy to understand that the activation of transition “tbet” means that the queue stays in waiting state and that the activation of transition “tget” enables the queue to be served. Both transitions can only be enabled if the queue is currently in waiting state. As “tbet” and “tget” are immediate transitions we have to define weights. This means that the sum of weights of “tbet” and “tget” for each queue model equals 1. Furthermore the sum of probabilities of “tget” for all queue models equals 1. The value of “tget” corresponds to the weight the queue. The weights are considered variable.

Now we will describe the last part of the queue model. If a token enters the submodel through transition “tswitchoverq23” it is moved to place “Ptpq”. Then exactly one of the transitions is enabled. Transition “ttestet” is enabled if the queue is not enabled to be served which means there is no token in place “Pse”. Transition “ttestq” is enabled if place “Pqueue” is empty which means that there are no tokens that may be served. As these transitions are immediate they will be served first. If they are not enabled the timed transition “Tserving” is activated which means that one packet of the queue is served. We already described the phase-type arrival process in Section 4 so will not deal with in greater detail in this section.

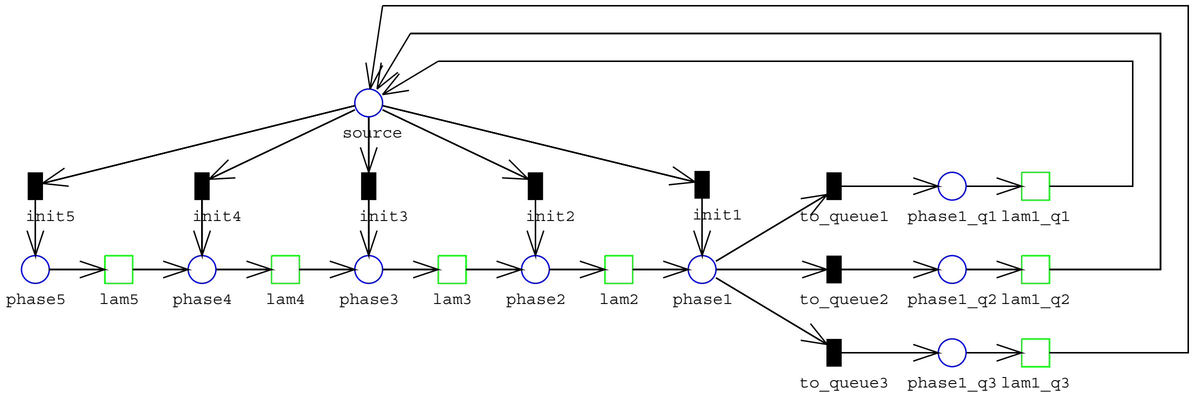

Figure 3: Phase-type arrival process

2 Tables of places and transitions

2.1 Places

Queue models:

Name	Initial tokens
Pfpe	Class 1: 1, otherwise: 0
Ptne	0
Pfpq	Class 1: 1, otherwise: 0
Pfpq	0
Pqueue	0
Pqcapacity	10
Psw	1
Pse	0

Phase-type arrival process:

Name	Initial tokens
Source	1
phase1, ..., phase5	0
phase1_q1, ..., phase1_q3	0

2.2 Immediate transitions

Queue models:

Name	Weight
Tbet	Variable, but equals 1 – weight of tget
Tget	Weight w_i for each queue
Tqpoll	Weight is irrelevant because no competing transition is enabled
Ttested	Weight is irrelevant because no competing transition is enabled

Ttestq	Weight is irrelevant because no competing transition is enabled
accept_q1, ..., accept_q3	Weight is irrelevant because no competing transition is enabled
reject_q1, ..., reject_q3	Weight is irrelevant because no competing transition is enabled

Phase-type arrival process:

Name	Weight
init1	0.009981306
init2	0.2282218954
init3	0.1766767211
init4	0.363430587
init5	0.22168949
to_queue1, …, to_queue3	1

2.3 Timed transitions

Central model:

Name	Fire rate
lam1_q1, ..., lam1_q3	1 104.058155
tswitchoverq12, tswitchoverq23, tswitchoverq31	10 000
tswitchovert12, tswitchovert23, tswitchovert31	10 000

Queue models:

Name	Fire rate
Tserving	100

Phase-type arrival process

Name	Fire rate
lam2	1 104.058155
lam3	172.5881243
lam4	172.5881243
lam5	31.4008596