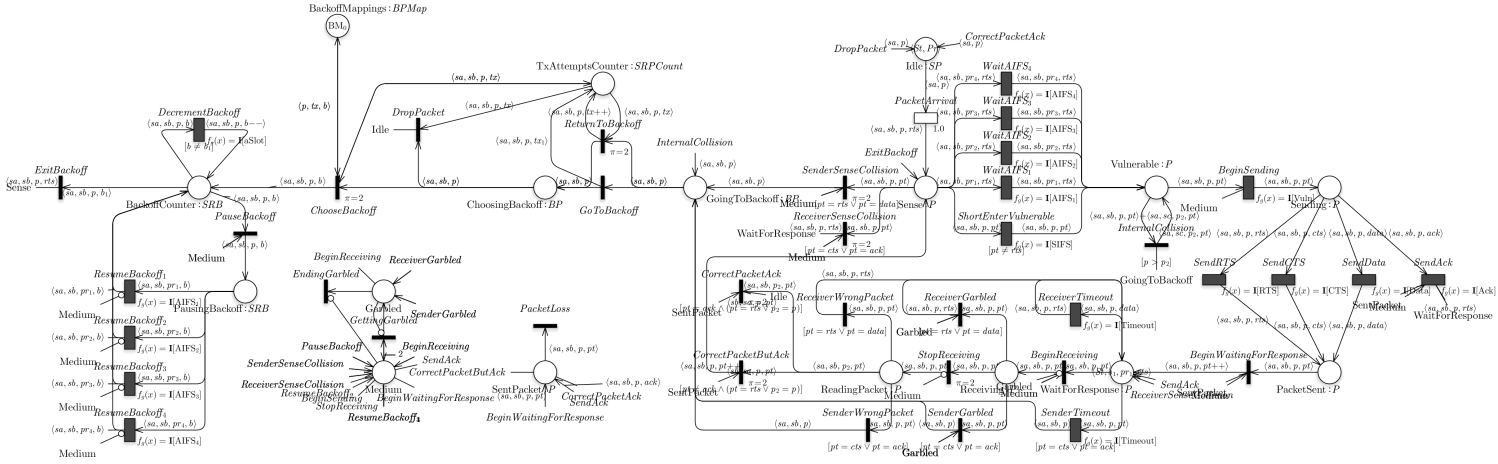
Paolo Ballarini and Benoît Barbot
paolo.ballarini@centralesupelec.fr

This form is a summary description of the model entitled "VehicularWifi" proposed for the Model Checking Contest @ Petri Nets. Models can be given in several instances parameterized by scaling parameters. Colored nets can be accompanied by one or many equivalent, unfolded P/T nets. Models are given together with property files (possibly, one per model instance) giving a set of properties to be checked on the model.

Description

Carrier sense multiple access with collision avoidance (CSMA/CA) is the basic scheme upon which access to the shared medium is regulated in many wireless networks. With CSMA/CA a station willing to start a transmission has first to find the channel free for a given duration otherwise it will go into *backoff*, i.e. , refraining for transmitting for a randomly chosen delay. Performance analysis of a wireless network employing CSMA/CA regulation is not an easy task: except for simple network configurations, analytical solutions of key performance indicators (KPI) cannot be obtained; hence one has to resort to formal modeling tools. This model presents different kinds of CSMA/CA-based wireless networks, namely: the IEEE 802.11 Wireless Local Area Networks (WLANs) and the 802.11p Vehicular Ad Hoc Networks (VANETs), which extend 802.11 with priorities over packets.

In this model, only the qualitative behavior is described. The original model specifies the temporal and stochastic behaviors of the system and can be found in the reference.



Graphical representation

References

Ballarini, P., Barbot, B., and Vasselin, N. (2019). Performance modelling of access control mechanisms for local and vehicular wireless networks. CoRR, abs/1901.04285. http://arxiv.org/abs/1901.04285

Scaling parameter

This model is not parameterized.

Size of the model

number of places:	21
number of transitions:	41
number of arcs:	136

Paolo Ballarini and Benoît Barbot
paolo.ballarini@centralesupelec.fr

Structural properties

ordinary — *all arcs have multiplicity one* ..... ✗
simple free choice — *all transitions sharing a common input place have no other input place* ..... ✗
extended free choice — *all transitions sharing a common input place have the same input places* ..... ?
state machine — *every transition has exactly one input place and exactly one output place* ..... ✗
marked graph — *every place has exactly one input transition and exactly one output transition* ..... ✗
connected — *there is an undirected path between every two nodes (places or transitions)* ..... ?
strongly connected — *there is a directed path between every two nodes (places or transitions)* ..... ?
source place(s) — *one or more places have no input transitions* ..... ?
sink place(s) — *one or more places have no output transitions* ..... ?
source transition(s) — *one or more transitions have no input places* ..... ?
sink transitions(s) — *one or more transitions have no output places* ..... ?
loop-free — *no transition has an input place that is also an output place* ..... ?
conservative — *for each transition, the number of input arcs equals the number of output arcs* ..... ?
subconservative — *for each transition, the number of input arcs equals or exceeds the number of output arcs* ..... ?
nested units — *places are structured into hierarchically nested sequential units*[(a)] ..... ✗

Behavioural properties

safe — *in every reachable marking, there is no more than one token on a place* ..... ?
dead place(s) — *one or more places have no token in any reachable marking* ..... ?
dead transition(s) — *one or more transitions cannot fire from any reachable marking* ..... ?
deadlock — *there exists a reachable marking from which no transition can be fired* ..... ?
reversible — *from every reachable marking, there is a transition path going back to the initial marking* ..... ?
live — *for every transition t, from every reachable marking, one can reach a marking in which t can fire* ..... ?

Size of the marking graph

number of reachable markings:	?
number of transition firings:	?
max. number of tokens per place:	?
max. number of tokens per marking:	?

Other properties

Any packet created by the *PacketArrival* transition will eventually be the binding of either the *DropPacket* transition or the *CorrectPacketAck* transition.

[(a)] the definition of Nested-Unit Petri Nets (NUPN) is available from http://mcc.lip6.fr/nupn.php