IEEE 802.3 CSMA/CD Protocol

AVACS S2
Phase 2

July 28, 2011

1 Description of the Model

In this case study, we build a model implementing the IEEE 802.3 Carrier Sense Multiple Access with Collision Dectection (CSMA/CD) Protocol which is one of the most important parts of the Ethernet. A station using the CSMA/CD protocol aborts transmissions as soon as it detects them. If it senses the channel idle, transmission of data begins immediately. Unlike in CSMA, the station senses the channel during transmission to be able to detect collisions as soon as possible. In case a collision is detected, i.e. the data read is different to the data transmitted, the current transmission is aborted and retransmission is attempted after a certain amount of time.

To determine when a station should try to retransmit, the exponential backoff algorithm is used. First, a "Jamming signal" is sent over the channel to prevent other stations to transmit data. Thereafter, a number out of $\{0, m\}$ is randomly chosen, where m could be any positive value. If retransmitting fails again, the set is extended to $\{0, m, 2 \cdot m, 3 \cdot m\}$. In general, after the n-th failed transmission, resending the frame is attempted after $k \cdot m$ time units, where k is a random integer between 0 and $2^n - 1$.

Our model of the CSMA/CD protocol consists of two stations sharing one channel with an initial collision. It is based on the probabilistic timed automata given in the corresponding *PRISM* case study[1] [1]. The model's parameters are PD, the jamming signal propagation delay, TD, the packet transmission delay, and BCMAX, the maximum value of the backoff counter. We are interested in the following properties:

1. **Probabilistic reachability properties:**
   (a) Eventually both stations send their packet correctly with probability 1.
   $P_{\geq 1} : P(\Diamond \text{ did(end1) \&\& did(end2)}) \geq 1.0$
2. **Expected-time reachability properties:**
   (a) The maximum expected time until both stations correctly deliver their packets.
   $E_{max} : T_{max}(\text{did(end1) \&\& did(end2)})$

[1] `http://www.prismmodelchecker.org/casestudies/csma.php`

Property	Result
$P_{\geq 1}$	true
E_{max}	$1770 \mu s$
E_{min}	$1735 \mu s$
D_{max}	0.872
D_{min}	0.729

MDP	States
Standard	22991
Deadlines	12165193

Property	Time (s)	Memory (kB)
MC	5	-
$P_{\geq 1}$	3	n/a
MC	6	-
E_{max}	3	1441
E_{min}	9	1441
MC	267	-
D_{max}	760	305376
D_{min}	230	305346

Table 1: Statistics for the *mcpta* generated model

(b) The minimum expected time until both stations correctly deliver their packets.
E_{min} : T_{min}(did(end1) && did(end2))

3. **Probabilistic time-bounded reachability properties:**

   (a) The maximum probability of both stations correctly delivering their packets by the deadline D
   D_{max} : $P_{max}(\Diamond$ did(end1) && did(end2) && time $\leq D)$

   (b) The minimum probability of both stations correctly delivering their packets by the deadline D
   D_{min} : $P_{min}(\Diamond$ did(end1) && did(end2) && time $\leq D$

To construct our model, we used *mcpta* [2]. We want to compare this generated model to the hand-written one of the *PRISM* case study.

2 Results

We applied *PRISM 3.2* to our model using the "sparse" engine, which performed best. All properties were checked for $PD = 26\mu s$, $TD = 808\mu s$ and $BCMAX = 1$. As a deadline we chose $D = 1800\mu s$. Results are given in Table 1 and were obtained on an Intel Core Duo T9300 (2.5 GHz) system running Windows Vista x64. The left table depticts the model-checking results for all considered properties. Notably, probabilities for properties D_{max} and D_{min} are higher than the ones obtained from the hand-written model. In fact, our determined lower bound matches the determined upper bound of the other model.

The middle table shows the number of states of the underlying Markov Decision Process for so called standard properties ($P_{\geq 1}, E_{max}, E_{min}$) and deadline properties (D_{max}, D_{min}). Notably, the state space explodes for deadline properties due to higher maximum constants for some clocks and the deadline of $1800\mu s$. As shown in [2] we were able to reduce the size of the state-space significantly for this test case. Nevertheless, the manually generated model's state-space only contains approximately half of the states needed in the automatically generated one.

The right table depicts model checking times and consumed memory for each property. "MC" refers to the model construction time. In case of $P_{\geq 1}$, model checking times of the generated model are at least 10 times faster as for the hand-written one. Considering properties D_{max} and D_{min}, our model outperforms the other in construction time, but more time is needed to check the properties due to the exploration of a larger state-space.

References

[1] M. Duflot, L. Fribourg, T. Hérault, R. Lassaigne, F. Magniette, S. Messika, S. Peyronnet, and C. Picaronny. Probabilistic Model Checking of the CSMA/CD protocol using PRISM and APMC. In *Proc. 4th Workshop on Automated Verification of Critical Systems (AVoCS'04)*, volume 128(6) of *Electronic Notes in Theoretical Computer Science*, pages 195–214. Elsevier Science, 2004.

[2] Arnd Hartmanns and Holger Hermanns. A Modest Approach to Checking Probabilistic Timed Automata. In *QEST*. IEEE Computer Society, September 2009.