Bounded Retransmission Protocol

AVACS S2
Phase 2

July 28, 2011

1 Description of the Model

The model considered in this case study represents an implementation of the Bounded Retransmission Protocol (BRP) [4]. BRP is a data link protocol for transmitting partitioned files over unreliable channels. Each part, in the following referred to as chunk, is bounded in the number of permitted retransmissions. The underlying concept is similiar to the Alternating Bit Protocol. Each chunk contains a sequence bit. The sender transmits a chunk continually as long as it receives an acknowledgement for this chunk. Thereafter, the sequence bit is flipped and the next chunk is transmitted. Instead of trying retransmitting arbitrary often, BRP bounds the number of retransmissions for each chunk. The protocol has been modelled and verified with a multitude of different formalisms and methods. In particular, timed and functional properties have been checked using a timed automata model [2], while a probabilistic model based on Markov decision processes has been studied using both the RAPTURE verification tool [1] and *PRISM*.

Our obtained Modest models [3] represent probabilistic timed automata, thus combining former timed automata and probabilistic models. The combination of both approaches is reasonable if we consider the loss of a chunk in a certain channel. If a timed automata model is used, the unlikely case that every chunk is lost is also considered. Since real life channels on a physical level, e.g. copper or coaxial-cables, fail or introduce noise with a certain probability this should be taken into account. Furthermore, this allows us to not only check the properties from the previously mentioned case studies, but also probabilistic time-bounded as well as expected-time reachability properties.

For the purpose of this case study, we built two different models both of which implement the same behaviour. The first model is a flat one, i.e. it consists of many nested statements like invariant-checks. As mentioned before, the model is functionally correct, but is rather difficult to understand, debug and extend. For this reason, we also provide a modular model easier to understand and extend.

For both models, we checked the following properties:

1. **Timed (invariant) properties:**

(a) T_1 : there is at most one message in transit for each channel
invariant !threw(channel$_{overflow}$)

(b) T_2 : there is at most one message in transit in total
invariant !(inTransitK || inTransitL)

(c) T_{A_1} : no premature timeouts
invariant !threw(premature$_{timeout}$)

(d) T_{A_2} : sender starts new file only after receiver reacted to failure
invariant !did(get$_k$) || !did(s$_{restart}$) && did(r$_{timeout}$)

2. **Probabilistic reachability properties:**

(a) P_A : the maximum probability that eventually the sender reports a certain unsuccessful transmission but the receiver got the complete file
$P_{max}(\diamond\ did(s_{nok})\ ||\ did(r_{ok}))$

(b) P_B : the maximum probability that the sender reports a certain successful transmission but the receiver did not get the complete file
$P_{max}(\diamond\ did(s_{ok})\ \&\&\ !did(r_{ok}))$

(c) P_1 : the maximum probability that eventually the sender does not report a successful transmission
$P_{max}(\diamond\ did(s_{ok})\ ||\ did(s_{dk}))$

(d) P_2 : the maximum probability that eventually the sender reports an uncertainty on the success of the transmission
$P_{max}(\diamond\ did(s_{dk}))$

(e) P_3 : the maximum probability that the sender reports an unsuccessful transmission after more than 8 chunks have been sent successfully
$P_{max}(\diamond\ did(s_{nok})\ \&\&\ i > 8)$

(f) P_4 : the maximum probability that eventually the receiver does not receive any chunk and the sender tried to send a chunk
$P_{max}(\diamond\ (did(s_{ok})\ ||\ did(s_{nok})\ ||\ did(s_{dk}))\ \&\&\ !did(get_k))$

3. **Expected-time reachability properties:**

(a) E_{max} : the maximum expected time until the transfer of the first file is finished (successfully or unsuccessfully)
$T_{max}(first\_file\_done)$

(b) E_{min} : the minimum expected time until the transfer of the first file is finished (successfully or unsuccessfully)
$T_{min}(first\_file\_done)$

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

(a) D_{max} : the maximum probability that the sender reports a successful transmission within 64 time units
$P_{max}(\diamond\ did(s_{ok})\ \&\&\ time \leq 64)$

Prop.	(16,2,1)	(16,2,4)	(64,5,1)	(64,5,4)
T_1	true	true	true	true
T_2	true	true	true	true
T_{A_1}	true	true	true	true
T_{A_2}	true	true	true	true
P_A	0.000	0.000	0.000	0.000
P_B	0.000	0.000	0.000	0.000
P_1	$4.233 \cdot 10^{-4}$	$4.233 \cdot 10^{-4}$	$4.482 \cdot 10^{-8}$	$4.482 \cdot 10^{-8}$
P_2	$2.645 \cdot 10^{-5}$	$2.645 \cdot 10^{-5}$	$7.003 \cdot 10^{-10}$	$7.003 \cdot 10^{-10}$
P_3	$1.852 \cdot 10^{-4}$	$1.852 \cdot 10^{-4}$	$3.852 \cdot 10^{-8}$	$3.852 \cdot 10^{-8}$
P_4	$8.000 \cdot 10^{-6}$	$8.000 \cdot 10^{-6}$	$6.400 \cdot 10^{-11}$	$6.400 \cdot 10^{-11}$
E_{max}	33.473	132.413	133.897	529.691
E_{min}	1.480	4.442	5.897	17.692
D_{max}	1.000	1.000	1.000	0.999
D_{min}	1.000	0.000	0.000	0.000

Table 1: Model-checking results of the considered properties

(b) D_{min} : the minimum probability that the sender reports a successful transmission within 64 time units
$P_{min}(\diamond\ did(s_{ok})\ \&\&\ time \leq 64)$

The timed invariant properties have been taken from [2] and hold for both the timed automata and the Modest models. The probabilistic reachability properties have already been studied in the *PRISM*/RAPTURE models and we can reproduce the exact results on both models.

2 Results

We checked all properties for four different combinations of (N, MAX, TD) on both models where N is the number of frames (chunks) per file, MAX is the maximum number of retransmissions per frame and TD is the transmission delay. We used *PRISM* 3.3 with the "sparse" engine, which performed best. Table 1 lists the model-checking results for the properties we checked. We note that these results match those obtained from previous models where applicable. We obtained the same results from both the flat and the modular model.

In Table 2, we give state-space size for the underlying MDP of both models, as reported by *PRISM*. Again, we tested them with the same parameters as in the model-checking part. Since time-bounded reachability properties introduce a certain overhead, we checked them in a separate run, denoted *deadlines*. As expected, the modular model is slightly less efficient, i.e. the MDP contains more states.

Table 3 lists the model-checking time and memory usage as reported by *PRISM*. All runs were performed on an Intel Core Duo T9300 (2.5 GHz) system. *MC* refers to the

flat	(16,2,1)	(16,2,4)	(64,5,1)	(64,5,4)
Standard	15866	89900	199466	1219940
Deadlines	791961	3175527	8981283	19821997

modular	(16,2,1)	(16,2,4)	(64,5,1)	(64,5,4)
Standard	20701	101908	250627	1347700
Deadlines	1372931	3719147	12035563	24255791

Table 2: State-space size of the underlying MDP of both models

model construction time.

References

[1] Pedro R. D'Argenio, Bertrand Jeannet, Henrik Ejersbo Jensen, and Kim Guldstrand Larsen. Reachability Analysis of Probabilistic Systems by Successive Refinements. In Luca de Alfaro and Stephen Gilmore, editors, *PAPM-PROBMIV*, volume 2165 of *Lecture Notes in Computer Science*, pages 39–56. Springer, 2001. ISBN: 3-540-42556-X.

[2] Pedro R. D'Argenio, Joost-Pieter Katoen, Theo C. Ruys, and Jan Tretmans. The Bounded Retransmission Protocol Must Be on Time! In Ed Brinksma, editor, *TACAS*, volume 1217 of *Lecture Notes in Computer Science*, pages 416–431. Springer, 1997. ISBN: 3-540-62790-1.

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

[4] Leen Helmink, M. P. A. Sellink, and Frits W. Vaandrager. Proof-Checking a Data Link Protocol. In Henk Barendregt and Tobias Nipkow, editors, *TYPES*, volume 806 of *Lecture Notes in Computer Science*, pages 127–165. Springer, 1993. ISBN: 3-540-58085-9.

flat	(16,2,1)		(16,2,4)		(64,5,1)		(64,5,4)	
MC	1s	-	3s	-	6s	-	22s	-
T_1	0s	n/a	0s	n/a	0s	n/a	0s	n/a
T_2	0s	n/a	0s	n/a	0s	n/a	0s	n/a
T_{A_1}	0s	n/a	0s	n/a	0s	n/a	0s	n/a
T_{A_2}	0s	n/a	0s	n/a	0s	n/a	0s	n/a
MC	1s	-	3s	-	6s	-	22s	-
P_A	0s	n/a	0s	n/a	0s	n/a	0s	n/a
P_B	0s	n/a	0s	n/a	0s	n/a	0s	n/a
P_1	7s	503 kB	25s	2764 kB	91s	6348 kB	391s	37885 kB
P_2	2s	503 kB	6s	2764 kB	16s	6348 kB	58s	37885 kB
P_3	4s	500 kB	13s	2764 kB	64s	6348 kB	247s	37885 kB
P_4	1s	446 kB	2s	2457 kB	4s	5632 kB	22s	33894 kB
MC	1s	-	2s	-	5s	-	20s	-
E_{max}	0s	722 kB	2s	4198 kB	3s	9216 kB	31s	57856 kB
E_{min}	0s	722 kB	2s	4198 kB	3s	9216 kB	23s	57856 kB
MC	27s	-	89s	-	219s	-	477s	-
D_{max}	10s	21913 kB	37s	91648 kB	189s	258662 kB	483s	619827 kB
D_{min}	6s	21606 kB	18s	80281 kB	55s	227840 kB	60s	488345 kB

Table 3: Comparison of model-checking times and memory consumption for different parameters