Model Checking Dependability Attributes of Wireless Group Communication

Mieke Massink
C.N.R.-ISTI, Via Moruzzi 1,
I-56124 Pisa, Italy,
M.Massink@isti.cnr.it

Joost-Pieter Katoen
Dept. of Comp. Science, Univ. of Twente,
P.O. Box 217, 7500 AE Enschede, The Netherlands
katoen@cs.utwente.nl

Diego Latella
C.N.R.-ISTI, Via Moruzzi 1,
I-56124 Pisa, Italy,
D.Latella@isti.cnr.it

Abstract

Models used for the analysis of dependability and performance attributes of communication protocols often abstract considerably from the details of the actual protocol. These models often consist of concurrent sub-models and this may make it hard to judge whether their behaviour is faithfully reflecting the protocol. In this paper, we show how model checking of continuous-time Markov chains, generated from high-level specifications, facilitates the analysis of both correctness and dependability attributes. We illustrate this by revisiting a dependability analysis [8] of a variant of the central access protocol of the IEEE 802.11 standard for wireless local area networks. This variant has been developed to support real-time group communication between autonomous mobile stations. Correctness and dependability properties are formally characterised using Continuous Stochastic Logic and are automatically verified by the ETMCC model checker. The models used are specified as Stochastic Activity Nets.

1. Introduction

Continuous-time Markov chains (CTMCs) are widely used to analyse important system dependability and performance issues. Usually such CTMCs are generated automatically from higher-level specifications such as e.g. Stochastic Activity Nets (SAN) [21] for which mature tool-support is available. Such tools support both the development of high-level specifications and the calculation of relevant measures, e.g. steady-state and transient state probabilities.

The high-level specifications used for the analysis of dependability and performance aspects of communication protocols often abstract considerably from the details of the actual protocol. In many cases, these models, in their turn, consist of concurrently composed sub-models. This may complicate the judgement whether their behaviour is faithfully reflecting the protocol. The extension of dependability and performance analysis tools with model-checking capabilities and a temporal logic allows for the verification of behavioural aspects as well as for the convenient, concise and unambiguous specification and automated verification of dependability and performance measures.

In this paper, we illustrate these advantages in practice by revisiting part of a dependability analysis of a variant of the *centralised* medium access protocol of the IEEE 802.11 standard for wireless local area networks [4, 8, 9]. This variant has been developed to provide reliable real-time group communication within teams of autonomous mobile robotic systems over wireless (radio) networks [22, 19]. In the IEEE 802.11 standard, the problem of message loss is addressed by defining two alternating periods of medium access control; a centralised one suitable for the exchange of time-critical messages, and a distributed one, suitable for less or non-time critical messages. The *distributed* medium access control mechanism for non time-critical communication over wireless networks has been studied in [16] applying probabilistic model checking techniques.

Group communication between autonomous mobile stations via wireless local area networks presents particular problems due to the *locomotion* of the mobile stations and the *unshieldedness* of wireless communication. It is therefore susceptible to a high degree of message losses in a bursty fashion.

The variant of the protocol that we consider in this paper proposes to reduce the number of retransmissions required to guarantee reliable communication in order to improve the real-time performance of the protocol. The reduction of reliability due to fewer retransmissions is compen-

sated for by a mechanism of active acknowledgments and the distribution of decision information that is included in the header of broadcasted messages. For many applications on real-time mobile stations the reduced reliability does not cause a serious problem as long as all mobile stations in the network agree in time not to deliver a message to their application when there is some station that did not receive the user message, viz. the property of agreement is satisfied.

In this paper, we analyse the models developed in [4, 8, 9] to determine the probability that a station misses a decision message and the probability that a user message is never delivered. First we check the correctness of the analytic model by generating the CTMC using the UltraSAN tool [21] and verify correctness properties of the concurrent model by the prototype stochastic model checker ETMCC (Erlangen Twente Markov Chain Checker) [13]. This model checker allows for the verification of both qualitative and quantitative (stochastic time) properties expressed as formulas of the (stochastic) branching time logic CSL (Continuous Stochastic Logic) [1, 2]. UltraSAN is a software package for model-based evaluation of systems represented as SAN's. It provides analytic solvers as well-as discrete-event simulators but has no model-checking facilities.

The contribution of this paper is threefold. We use a model checking approach on a case study on which numerical and experimental results are available in the literature. We show that the model checking capability to verify both qualitative and quantitative properties of concurrent models can greatly enhance the effectiveness of existing dependability and performance analysis tools to increase the confidence in the accuracy and faithfulness of the models on which the analyses are based. In fact, its automatic verification reveals serious problems of the existing model and gives rise to the development of a *new* model that more faithfully reflects the synchronous broadcast aspects of the protocol. We show this by comparing the verification results for qualitative and quantitative properties for both models. Finally, the direct link between the high-level specification in SAN and the derived CTMCs on which the analysis by both UltraSAN and ETMCC are based gives an opportunity to compare the results and to obtain feedback on the performance of the tools. An extended and more detailed version of the present paper can be found in [17].

2. Model Checking Dependability

In the model checking approach to dependability analysis a *model* of the system under consideration is required together with a desired *property* or *dependability measure*. Model checking provides a systematic check whether the given model satisfies the property. Effective, optimised model checking algorithms have been developed to dramatically reduce the state space that needs to be searched, and to keep its representation compact as well [7]. Typically, models are finite-state automata, where transitions model the evolution of the system while moving from one state to another. These automata are usually generated from a high-level description language. In the case of stochastic modelling, such models are typically CTMCs and languages such as stochastic Petri nets, stochastic process algebras or SANs are used to generate them. In the model checking approach, the properties are usually expressed in some form of temporal logic. In this paper the Continuous Stochastic Logic [1, 2] is used, which is a stochastic variant of the well-known Computational Tree Logic (CTL) (see e.g. [7]). CTL allows for stating properties over *states*, and over *paths*. CSL extends CTL by two probabilistic operators that refer to the steady-state and transient behaviour of the system being studied. Whereas the steady-state operator refers to the probability of residing in a particular set of *states* (specified by a state-formula) in the long run, the transient operator allows us to refer to the probability of the occurrence of particular *paths* in the CTMC. In order to express the time-span of a certain path, the path-operators until $\mathcal{U}$ and next X are extended with a parameter that specifies a time-interval. Let I be an interval on the real line, p a probability value and $\bowtie$ a comparison operator, i.e., $\bowtie \in \{<, \leqslant, \geqslant, >\}$. The syntax of CSL is:

State-formulas

$$\Phi ::= a \mid \neg\Phi \mid \Phi \vee \Phi \mid \mathcal{S}_{\bowtie p}(\Phi) \mid \mathcal{P}_{\bowtie p}(\varphi)$$

$\mathcal{S}_{\bowtie p}(\Phi)$: prob. that Φ holds in steady state $\bowtie p$
$\mathcal{P}_{\bowtie p}(\varphi)$: prob. that a path fulfils $\varphi \bowtie p$

Path-formulas

$$\varphi ::= X^I \Phi \mid \Phi \mathcal{U}^I \Phi$$

$X^I \Phi$: next state is reached at time $t \in I$ and fulfils Φ
$\Phi \mathcal{U}^I \Psi$: Φ holds along path until Ψ holds at $t \in I$

The meaning of atomic propositions (a), negation ($\neg$) and disjunction ($\vee$) is standard; note that using these operators, other boolean operators such as conjunction ($\wedge$), implication ($\Rightarrow$), true (true) and false (false), and so forth, can be defined. The state-formula $\mathcal{S}_{\bowtie p}(\Phi)$ asserts that the steady-state probability for the set of Φ-states meets the bound $\bowtie p$. The operator $\mathcal{P}_{\bowtie p}(.)$ replaces the usual CTL path quantifiers $\exists$ and $\forall$. In CTL, the state-formula $\exists\varphi$ is valid in state s if there *exists* some path starting in s and satisfying φ and $\forall\varphi$ is valid if *all* paths satisfy φ. In CSL, $\exists\varphi$ can be written as $\mathcal{P}_{>0}(\varphi)$ and $\forall\varphi$ as $\mathcal{P}_{\geqslant 1}(\varphi)$. This allows for the expression of qualitative as well as stochastic properties in CSL. We shall frequently use this aspect. $\mathcal{P}_{\bowtie p}(\varphi)$ asserts that the probability measure of the set of paths satisfying φ meets the bound $\bowtie p$. In CTL, a path satisfies an until-formula $\Phi\mathcal{U}\Psi$ if there is a state on the path where Ψ holds, and at every preceding state on the path, if any, Φ holds. In CSL,

temporal operators like $\Diamond$, $\Box$ and their real-time variants $\Diamond^I$ or $\Box^I$ can be derived, e.g., $\mathcal{P}_{\bowtie p}(\Diamond^I\,\Phi) = \mathcal{P}_{\bowtie p}(\mathsf{true}\;\mathcal{U}^I\,\Phi)$ and $\mathcal{P}_{\geqslant p}(\Box^I\,\Phi) = \mathcal{P}_{<1-p}(\Diamond^I\,\neg\Phi)$. The untimed next- and until-operators are obtained by $X\Phi = X^I\Phi$ and $\Phi_1\,\mathcal{U}\,\Phi_2 = \Phi_1\,\mathcal{U}^I\,\Phi_2$ for $I = [0,\infty)$.

CSL allows for the expression of four different types of performance and dependability measures, viz. steady-state measures, transient-state measures, path-based measures, and nested measures. In this paper we shall use several transient-state measures and nested measures.

The ETMCC model checker [13] is a prototype tool that supports the verification of CSL-properties over CTMCs. The model checker takes as input a model file with a textual representation of a CTMC, a label file associating each state to the atomic propositions that hold in that state and a given accuracy. ETMCC is based on sparse matrix representations of CTMCs. Alternative model checkers for CSL include PRISM [15], Prover [23] and the APNN (Abstract Petri Net Notation) toolbox [5].

3. Group Communication Protocols for Wireless Local Area Networks

Real-time group communication protocols for wireless local area networks are very important for applications where autonomous mobile stations are expected to cooperate and synchronise their behaviour in order to accomplish a common goal.

A real-time group communication protocol needs to (a) guarantee real-time communication, i.e., it needs to guarantee an upper bound on the delay of message communication, (b) provide reliable communication, (c) be able to handle failure of mobile stations in a group and keep the stations informed about the status of each station, and finally, (d) guarantee that all stations receive the same messages in the same order.

The main problem that a real-time group communication protocol for wireless networks needs to overcome is the high degree of message losses. This high degree of losses is caused by the unshieldedness of the wireless medium, and partially also by the velocity of the mobile stations. A further characteristic of these losses is their occurrence in bursts, which means that often series of consecutive messages are lost.

In the IEEE 802.11 standard [14], the problem of message losses and the real-time communication requirement have been addressed by the introduction of two alternating periods of medium access control. In the Contention Period (CP), distributed medium arbitration takes place and collisions may occur. The arbitration scheme used during CP is standard carrier sense multiple access with collision avoidance (CSMA/CA). This period is useful for the exchange of non time-critical or less time-critical messages. The Contention Free Period (CFP) has a *centralised medium arbitration* and the group members get exclusive access right to communicate over the shared medium. The CFP is specifically designed for real-time communication. The two periods, CP and CFP, are activated in an alternating way under the control of a central Access Point (AP). This is a special fixed node in the network with a central position with respect to the mobile stations with which it communicates to obtain an optimal reachability. Both periods can have variable length.

During the CFP the AP coordinates the medium access for all stations that are reachable over the wireless network At the beginning of the CFP all stations remain silent, except for the AP that transmits a polling message to some station in the group. When a station receives a polling message it may broadcast a message over the network. The polling strategy is decided by the AP which is also in charge of assigning a sequence number to the broadcasted message in order to make total ordering of messages possible.

The real-time group communication protocol that we analyse in this paper is a variant of the protocol used for the CFP in the IEEE 802.11 standard and has been developed by Mock et al. [19].

3.1. Basic operation of the real-time group communication protocol

The protocol is based on a *dynamic redundancy scheme*. In this scheme a message is only retransmitted upon the detection of its loss. Such a scheme needs the explicit recognition of communication failures and an acknowledgment strategy that reports the status of a transmission.

The protocol is based on the following fault assumptions about the wireless network [8]: (a) if a message is delivered (during the CFP), it is delivered correctly and within a fixed time bound (t_m), (b) messages may be lost, possibly in an asymmetric way, i.e. some stations may receive a broadcast message while others do not. In any case, the number of consecutive losses of the *same* message is assumed to be bounded by the so-called *omission degree* OD, (c) the AP is reliable, i.e. it is not subject to any kind of error and finally (d) stations may suffer from crash failures or leave the reach of the AP.

The group communication protocol is structured into *rounds* and it is assumed that there is a maximum number n_{max} of stations in a group and that $N \leqslant n_{max}$ is the current size of the group. A round is composed of a series of slots, one for each station in the group, where each slot consists of a triple of message transmissions; a *polling* message from the AP to the station, a *broadcast request* message from the station to the AP and a *broadcast* message by which the AP sends the user message of the sending station to all stations. Each round is identified by a unique round

number, starting from 0 and incremented by 1 at the beginning of each new round. The AP polls each station of the group exactly once in each round, and polls the stations always in the same order, sending them the round number in the polling message. After being polled, station s (denoted as originator in the following) sends a broadcast request message to the AP. This message is composed of an acknowledgment field, a local sequence number and a *user* message m. The acknowledgment scheme implies that exactly one round after broadcasting a user message of a certain station, the AP is able to decide whether each group member has received the user message or not. For details we refer to [22] and [17].

3.2. Further enhancement of the protocol

The improvement of the protocol proposed in [22] aims at a further decrease in the latency of real-time messages by reducing the maximum number of message retransmissions from OD to a user-specified number lower than OD. This number is the so-called *resilience degree*, res. With this reduction of the number of retries full reliability of the protocol can no longer be guaranteed under the assumptions about the network (as long as $res < OD$). This means that it may happen that a user message that is broadcasted is not received by all stations in the group within res retransmissions. This is not a serious problem for many applications as long as all stations agree in time not to deliver that message to their respective application processes.

Thus, the stations are allowed to *deliver* a user message to the application only if the message is received by *all* stations. This is decided by the AP, based on a positive acknowledgment for that user message from every station. The decision of the AP is to be communicated in a reliable and timely way. This is achieved by means of the transmission of the *decision* for each user message through a field in the header of each broadcast message composed of $OD + 1$ bit-*pairs*. Every decision is retransmitted $OD + 1$ times, so there is no need for an acknowledgment of the reception of the decision by the station (given the network assumptions).

The functional correctness of the group communication protocol has been treated extensively in [22] where also a specification of the protocol is given in SDL (Specification and Design Language [18]). An analysis of the real-time performance of the protocol is provided in [19]. None of these works have used automatic verification tools for the verification of the properties.

3.3. Dependability measures

A number of dependability and performance measures for the protocol are addressed in [8], where a numerical analysis has been carried out by means of the UltraSAN tool [21]. We revisit two of the dependability measures in this paper, but follow a model-checking approach for their analysis. We start from the high-level SAN specifications in [8] and formulate the measures as CSL formulas. The two dependability measures that we address are:

$\mathbf{P_{D>OD}}$: The probability that a *decision message* (i.e. a message issued by the AP to commit or abort the delivery of a broadcast message) is not received by at least one station in the group, within T_{CFP} (duration of the CFP phase). This measure represents an estimate of the probability for the protocol to fail in an undetected and undesirable way with possible catastrophic consequences on the system and its users. Therefore, this probability should be sufficiently low.

$\mathbf{P_{UM}}$: The probability that the AP does not receive acknowledgments for a user message by all the stations within res retransmissions within T_{CFP}. In this case, the AP broadcasts to all stations in the group the decision not to deliver that message to their applications. In other words, P_{UM} is the probability that some station in the group has not acknowledged a user message sent by the AP after res retransmissions. This property gives an indication to which extent the validity property is violated.

4. A Dependability Model

In [8], a model is developed that covers relevant aspects of the protocol and its environment that are necessary to analyse the dependability measures of interest. A single model is used to analyse several dependability measures by varying the values of its parameters.

4.1. Fading model

In modelling the environment, the interference between different versions of the transmitted signal and the Doppler shift caused by the relative motion of receiving and sending stations, has been taken into account. Both effects cause the so-called *signal fading phenomenon*. The probability of message loss resulting from fading signals has been approximated by the first-order discrete time Markov chain (DTMC) [24] depicted in Fig. 1. The DTMC has two states, S and F, standing for (previous) success and failure of a communication respectively. If the previous communication has been successful, with probability p the next communication will also be successful. With probability $1 - p$, the next communication will be a failure. If the previous communication has failed, then with probability q the next communication fails, and with probability $1 - q$ it is successful. This behaviour can be presented as a transition probability matrix in a standard way. The probability of success or failure of a number of consecutive message losses (success) can be obtained by matrix multiplication. The param-

eters p and q have been derived considering the communication between the AP and the stations as Rayleigh fading channels and using experimental data available to calculate the approximate values [8]. In particular, p and q are functions of the steady state probability that a communication fails (PE) and the normalized Doppler frequency. For details we refer to [8, 24].

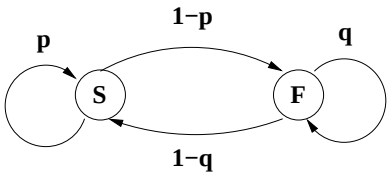

Figure 1. DTMC modelling channel fading

4.2. Station model

The fading model has been integrated into the model of a station defined using the SAN formalism [21] which is shown in Fig. 2. SANs are a high-level modelling formalism for the specification of dependability and performability models. SAN models consist of four primitive objects: places, activities, input gates and output gates. *Places* represent the state of the system and are marked by tokens, like in Petri nets. *Activities* represent transitions or actions of the system. *Input gates* are used to control the enabling of activities, and *output gates* are used to change the state of the model on completion of an activity.

Let us briefly explain the SAN model. The place $POLL$ models that the station is polled by the AP. Initially it has one token. The input gate chk enables the timed activity $nprb$ only if there is a token in place $POLL$ and no token at place $FAIL$. If this condition is satisfied, chk removes the token from place $POLL$. The exact interpretation that is given to the failing of a station depends on the dependability measure that is analysed. For the analysis of $P_{D>OD}$ a station fails if it missed more than OD consecutive decision messages within T_{CFP}. In the case of P_{UM} it fails if it missed more than res consecutive user messages within T_{CFP}. The timed activity $nprb$ (probabilistic broadcast) models the performance aspects of the wireless network and forms the central part of the model.

Model for the analysis of $P_{D>OD}$. When the model is used to analyse property $P_{D>OD}$, the time distribution function is chosen to be exponential with a rate being the reciprocal of the duration of one slot, i.e. the sum of the transition time of one polling message, a broadcast request message and a broadcast message. Let TP be the *mean time* required for the polling message to be transferred from the AP to a station, and TM the same for a broadcast message. Then the exponential distribution rate of a slot is $1/(2 * TM + TP)$.

The timed activity $nprb$ has two cases, represented as two small circles attached to the hollow oval in Fig. 2. The probability distribution of the two cases is defined by the case distribution and may also depend on the marking of the network at the moment of completion of the activity. In this model, the distribution depends on the marking of place $SUCCESS$. A token in place $SUCCESS$ means that the previous triple of polling, broadcast request and broadcast messages, has been a success. We obtain the fading characteristics as the outcome of the product of three matrices $M'.M.M$. Here M' represents the matrix defining the fading characteristics of the short polling message, with its characteristic probabilities p' and q', and M defines the fading characteristics of a broadcast message with its respective probabilities. Let P and Q be the resulting probabilities of this matrix multiplication, i.e. P is the probability in that resulting matrix of the self-loop from state S to itself, $1 - P$ is the probability in that matrix of the transition from S to F, etc. The probabilities associated with the two cases in the timed activity $nprb$ are then derived from the DTMC in Fig. 1 where p and q are now P and Q. So case 1, denoting a broadcast failure, connected to output gate $FAIL\_BC$ becomes $1 - P$ and case 2, denoting a successful broadcast, becomes P. If there is no token on place $SUCCESS$, the probabilities for the two cases are Q and $1 - Q$, respectively.

The output gate $FAIL\_BC$ removes any token from place $SUCCESS$, increments the number of tokens on place $COUNTER$ by one, and if the number of tokens on $COUNTER$ exceeds the omission degree OD, it puts a token on place $FAIL$. Otherwise, it puts a token on place $POLL$, modelling that the station is ready for the next communication (triple). The $COUNTER$ represents the number of consecutive failed communications for a given station. The output gate $SUCC\_BC$ changes the state of the model after a successful broadcast has taken place. It puts a token on place $SUCCESS$, resets $COUNTER$ to zero (i.e. removes all its tokens) and puts a token on place $POLL$. Initially, there is one token on place $SUCCESS$ and on place $POLL$, and all other places are empty.

UltraSAN provides a mechanism for replicating a single station. This allows for the easy generation of a model with N stations that may share places. In this case the stations share place $FAIL$.

Model for the analysis of P_{UM}. The model used for analysing property P_{UM} is the same as that for $P_{D>OD}$ except for the values of P and Q and the rate of the exponential distribution of the timed activity. In fact, for P_{UM} we