Analysis of Group Communication Protocols to Assess Quality of Service Properties

A. Coccoli[1], S. Schemmer[2], F. Di Giandomenico[3], M. Mock[2], A. Bondavalli[4]

1 *Dip. di Ing. dell'Informazione, Univ. of Pisa, via Diotisalvi 2, 56126 Pisa, Italy - a.coccoli@guest.cnuce.cnr.it*
2 *GMD-AiS, D-53754 St. Augustin, Germany - {schemmer, mock}@gmd.de*
3 *IEI-CNR, Area della Ricerca di Pisa, Via Alfieri 1, 56127 Pisa, Italy - digiandomenico@iei.pi.cnr.it*
4 *Dip. di Sistemi e Informatica, Univ. of Firenze, Via Lombroso 6/17, Firenze, Italy - a.bondavalli@dsi.unifi.it*

Abstract

This paper focuses on QoS analysis carried out through analytical modelling and experimental evaluation. QoS is defined as a set of qualitative and quantitative characteristics of a (sub)system, which are necessary for obtaining the required functionality of an application. Its analysis is a necessary step for the early verification and validation of an appropriate design, and for taking design decisions about the most rewarding choice, in relation with the user requirements. In this paper, we concentrate on a family of group communication protocols in a wireless environment, which is used as the reference system to which the QoS analysis is applied. Specific indicators have been defined and evaluated, which capture the main characteristics of the protocols and of the environment, focusing our attention on performance and dependability attributes. The model-based analysis is devoted to give an estimate of the coverage of the protocol assumptions and performance. We integrate the analytical modelling with fine-grained experimental measurements to determine realistic parameters of message delays and messages losses, and we compare the measured protocol performance with the analytical results. The main purpose of our analysis is to provide a fast, cost effective, and formally sound way to further analyse and understand the protocol behaviour and its environment.

1. Introduction

The development of distributed applications has already reached a good success both based on the Internet and the Web and on mobile networks that allow a very high connectivity. However, there is a clear need to define new computational and structural models to allow users, programmers and designers of such applications to reason at the proper abstraction level. Another very important problem consists in devising designs providing the proper quality of service (QoS) as required by applications [6], [7]. QoS can be defined as a set of qualitative and quantitative characteristics of a distributed system which are necessary for obtaining the required functionality of an application. Therefore the term QoS encompasses many aspects such as reliability, availability, fault tolerance and also properties such as the atomicity or reliability of broadcast/multicast services.

It is clear that the usefulness and practical utilisation of such (sub)system designs depend on the possibility to provide a QoS analysis of their offered features, in terms of proper defined dependability and performability related measures. When building a system, this is a necessary step for the early verification and validation of an appropriate design, and for taking design decisions about the most rewarding choice, in relation with the user requirements. Our approach for contributing towards these objectives is through analytical modelling and experimental evaluation. In this paper, we concentrate on a family of group communication protocols in wireless environment, which is used as the reference system to which the QoS analysis is applied.

To achieve real-time reliable group communication in wireless local area networks is a hard task. The mobility of the system components has a direct effect on the definition of a co-operative group and the hostility of the environment produces a great loss of messages. In this context, a protocol able to tolerate erroneous and uncooperative behaviour of the system components is required. In order to plan future co-operating actions, the protocol must dynamically detect which of the mobile autonomous systems currently form a co-operative group, and it must provide the group with a consistent view of its state. The protocol defined in [3] provides these properties by extending the IEEE 802.11 standard for wireless local area networks [1]. An interesting feature of this protocol is the possibility to handle a trade-off between different aspects of the QoS offered, such as performances, delay time, reliability and formal properties of the broadcast. Actually, according to the different assumptions that can be made on the environment and on the protocol

characteristics a family of protocols can be devised, with different quality of service offered.

In this paper, we focus on behavioural analysis based on quantitative estimates. We conduct an analysis of the protocol (family) and of the environment, focusing our attention on typical performance indicators and on the coverage of the assumptions the correctness of the protocol is based on. Our analysis of the protocols is carried on through both analytical and experimental approaches. The usefulness of exploiting both approaches is twofold: i) through experimental usage of the protocol in an representative context, field data are collected to provide numerical values to input parameters of the analytical model; ii) devising basic behavioural indicators through both approaches is a means to check the correctness of the analytical model, thus raising the confidence on the accuracy of the (more complex) final figures derivable from the analytical model. The main purpose is to provide a fast, cost effective, and formally sound way to further analyse and understand the protocol behaviour and its environment.

The rest of the paper is organised as follows. Section 2 is devoted to the description of the considered communication protocols, together with the definition of relevant metrics representative of the QoS in the selected environment. In Section 3, the models defined to evaluate the QoS indicators are presented. Section 4 contains the description of the experiments conducted to derive estimates of the model parameters, which are used in the next Section 5 to provide numerical evaluation of the analytical models. Some comparisons between QoS indicators determined by analytical models and experimental evaluation are also performed. Finally, concluding remarks and future work are outlined in Section 6.

2. A family of group communication protocols for wireless local area networks

2.1. Design of the protocols

A basic means for supporting the co-operation of autonomous mobile systems is their ability to communicate via wireless links. To achieve a real-time reliable group communication [2] in wireless local area networks is a hard task. The mobility of the system components has a direct effect on the definition of a co-operative group and the hostility of the environment produces a great loss of messages. The protocol presented in [3] provides reliable and efficient group communication services, based on extending the IEEE 802.11 standard for wireless local area networks.

The IEEE 802.11 [1] standard has the great advantage of providing the basic means for the implementation of a real-time communication protocol via the "Contention Free Period" (CFP) with centralised medium arbitration that regulates the accesses of all the system units. During the CFP, a central station that is denoted as the "Access Point" (AP) co-ordinates the access to the medium for a group of stations. The access point grants exclusive access to the medium by transmitting a polling message to some station in the group. Although there are no contentions during the CFP, the problem of message losses still has to be tackled. In fact the number of message losses is considerably higher than for wired local area networks, because the wireless medium is unshielded exposed to external interference. Broadcast messages are just unreliable datagrams sent on a best effort basis, neither order nor atomic delivery of broadcast messages is considered.

The protocols in [3] have been developed based on the following fault assumptions:

1) Messages delivered during the CFP are delivered correctly within a fixed time-bound (t_m).
2) Messages may be lost (omission faults). Furthermore, the losses may be asymmetric; i.e., some stations may receive a broadcast message and some may not. We assume that the number of consecutive message losses is bounded by the so-called *omission degree OD*.
3) Stations may suffer crash failures or leave the reach of the access point.
4) The access point is stable; i.e., it is not subject to any kind of error.

A first developed version aims at providing a reliable group communication protocol satisfying the properties of

i) *validity*, i.e., a message broadcast by a correct station is eventually delivered by every correct station;
ii) *agreement*, i.e., a message delivered by a station is eventually delivered by any other correct station;
iii) *integrity*, i.e., for any message m, every correct station delivers m at most once and only if m has been broadcast.

Using the AP as central co-ordinator, the communication of the group has been structured into rounds. During each round, the AP polls each station of the group exactly once. After being polled, a station returns a *broadcast request message* to the access point, which assigns a sequence number to that message and broadcasts it to the stations group. The broadcast request message is also used to acknowledge each of the preceding broadcasts by piggy-backing a bit field on the header of the request message. Each bit is used to acknowledge one of the preceding broadcasts. By this, one round after sending a *broadcast message*, the access point is able to decide whether each group member has received the message or not. In the latter case, the access point will retransmit the affected message. By the assumptions made above, a message is successfully transmitted after at most $OD+1$ rounds. If the AP does not receive the request message within a certain period of time after polling the station, it considers the request message (or polling message) to be lost, and transmits the last broadcast message of the not responding station if it has not yet been

acknowledged by all stations. If the AP does not receive the request message from a station for more than *OD* consecutive times, it considers that station to have left the group and broadcasts a message indicating the change in the group membership.

In order to enable the user to improve the timing guarantees, a variant of the protocol has been developed, that allows the user to specify the maximum number of retransmissions of the messages. This user-defined bound on message retransmissions (the so-called *resiliency degree*, *res(c*)) may be varying for different message classes *c*. Obviously, it is not useful to choose *res(c)* greater than *OD*. Choosing *res(c)* smaller than *OD*, however, allows trading reliability of message transmission for shorter transmission delays. If a message *m* is acknowledged by all stations after at most *res(c)*+1 rounds, the AP issues the decision to deliver *m* to the applications, through the broadcast of a *decision message*, which is retransmitted *OD*+1 consecutive times (to guarantee reception by all the correct stations under assumption 2) above). If, however, the AP does not receive the acknowledgement of at least one station after *res(c)*+1 rounds, a decision not to deliver *m* is issued, again through the broadcast of a *decision message*. To make the implementation efficient, the access point piggy backs its decisions on the broadcast messages it sends, by properly extending their headers. In this version of the protocol, the shorter delivery time for a message, obtained by reducing the maximum number of retransmissions for a broadcast message to *res(c)* times, is paid in terms of violation of the validity property (point i) above): now, a message requested to be broadcast by a correct station may be not received by all the other stations and therefore not delivered. However, the agreement and integrity properties are retained, which is enough for significant application scenarios.

Because of the different characteristics shown by the two versions, they cannot be compared one against the other in an absolute way; the choice of which one is better suited to be employed in a system depends on the requirements of the specific application at hand.

2.2. Definition of appropriate QoS indicators

The protocols described above have been defined to provide reliable and efficient co-operation of autonomous mobile systems via wireless links. In order to prove such basic characteristics of reliability and efficiency, we concentrate in the following on an analysis to estimate two groups of figures of interest, namely dependability related measures and performance related ones. Specifically, the dependability-related figures are directed to give an estimate of the coverage of the protocol assumption on the maximum number of consecutive message losses *OD*. For this purpose, a measure $P_{R>OD}$ is defined and evaluated for the first version of the communication protocol, indicating the probability that a *broadcast message* (i.e., a message requested to be broadcast by a station) is not received by at least one of the receiving stations after *OD*+1 transmissions, in the time interval T_{CFP} (representing the duration of a CFP, which is the timing window during which the protocol operates). A corresponding measure $P_{D>OD}$ is defined and evaluated for the second version of the protocol, indicating the probability that a *decision message* (i.e., a message broadcast by the AP to commit or abort the delivery of a *broadcast message*) misses to be received by at least one station, again evaluated in a certain interval of time, T_{CFP}, representing the duration of a CFP. So, both $P_{R>OD}$ and $P_{D>OD}$ represent an estimation of the probability, for the protocols, to fail in an undetected way, a very undesirable event with possibly catastrophic consequences on the system and its users (we say, the protocol experiences a *catastrophic failure*). Knowing such measures is very important at design phase, in order to take appropriate actions to avoid or limit such undesirable failure event. For example, should $P_{R>OD}$ result too high, possible actions to cope with it could be: a) use an higher value for *OD*, properly trading an higher probability of delivering a message to all the stations within *OD* retransmissions with the consequent diminishing of messages which can be broadcast within a CFP, or b) slightly change the protocol, by disconnecting the stations which do not acknowledged the receipt of the message (they are known to the AP), so as to maintain a consistent view of the received messages by all the active stations.

The performance analysis is intended to determine the technical limitations imposed by the communication system and the way the protocol behaves according to them. Representative figures to evaluate will be: i) the average number R_m of retransmissions for a single message; ii) the throughput, determined in number of delivered messages per second; and iii) for the second version of the protocol only, the probability P_{UM} that the AP does not receive acknowledgements on a message by all the stations in *res(c)* retransmissions, and therefore broadcasts to the active stations the decision not to deliver that message to the applications. The first two metrics are typical performance indicators, with R_m also useful to properly tune the protocol parameter *res(c)*. P_{UM} gives an indication of the extent of the violation of the validity property (point i) in section 2.1); to get desired values of P_{UM} implies proper tuning of the protocol parameter *res(c)*.

3. Analytical Modelling

The behaviour of the two versions of the communication protocol has been modelled by Stochastic Activity Networks (SAN) [4]. Three SANs have been derived: the first is used to compute the figures related to the assumptions coverage for the two versions of the

protocol ($P_{R>OD}$ and $P_{D>OD}$); the other two are used to measure the performance related indicators, again for the two versions (i.e., R_m and P_{UM}).

3.1. Assumptions and notation

The analysis has been conducted under the following assumptions:

1) The time-bound for sending a message over the network is fixed and denoted with t_m. It represents a bound for both a) the time to exchange a message between two agents in the network (namely, the AP and any other mobile station), and b) the time to broadcast a message from the AP to all the other stations;
2) the couple of actions "AP polls station$_i$- station$_i$ sends a request to broadcast" (indicated as *poll-request*) is managed as a single event. In fact, it is of minor importance in our study whether the access point does not receive an answer because of a lost polling- or a lost broadcast-request message. The probability of failure for the couple *poll-request* is constant and equal for each station; it is indicated as q_{pr};
3) failures considered are only those affecting the messages, which may fail to be received by the mobile stations and/or by the AP (omission failure). Mobile stations are therefore reliable. However, a station may migrate from the group; this event is taken into account by the protocol, through counting the number of *poll-request* messages towards the station which leaves the group. Although the probability of this event is not of interest per-se in the derived models, it contributes to the probability of failure of a message. The AP is assumed to be stable and reachable by all the stations belonging to the group;
4) the probability of failure of a broadcast message (that is, the message fails to be received) is constant for each station and indicated with q_{bc}. Message failures are stochastically independent (we are aware that this assumption is an approximation of the reality; our next step is actually the definition of analytical models where this assumption is released);
5) the value of *res (c)* is the same for all the messages;
6) the models for the evaluation of the dependability-related figures assume that the group membership remains the same during the whole T_{CFP} interval, that is, no station misses *OD*+1 consecutive *poll-requests*, which is the condition for the AP to consider that station as migrated from the group.

Table 1 summarises the internal parameters of the protocols and those of the models, together with the default values used in the subsequent numerical analysis (unless otherwise specified), as determined through experimental measurements, described in the next Section 4. For the sake of clarity, also the figures of merits under evaluation are reported in Table 1.

Table 1. Notations and definitions

Notation	Description	Value
q_{bc}	probability of failure of a broadcast message	$1.6*10^{-4}$
q_{pr}	probability of failure of the couple *poll-request*	$6.041*10^{-4}$
N	mobile stations in the group	4
t_m	time-bound for a message transmission (μsec)	7646.68
OD	omission degree	4
res(c)	resiliency degree (second version of the protocol)	2
T_{CFP}	duration of a Contention Free Period (in millisec)	variable
$P_{D>OD}$	probability that a *decision message* is lost by a station after *OD*+1 retransmissions	
$P_{R>OD}$	probability that a *broadcast message* is lost by a station after *OD*+1 retransmissions	
R_m	number of retransmissions for a broadcast message (average)	
T_{hr}	throughput (number of delivered messages per second)	
P_{UM}	probability that a broadcast message is not delivered	

3.2. Models for the Dependability-related measures

3.2.1. SAN used to evaluate the probability $P_{D>OD}$ for any station to miss a decision message after *OD*+1 consecutive retransmissions, in the time interval T_{CFP}. From the protocol behaviour, once a *decision message* (regarding the delivery/ not delivery of a *broadcast message*) is taken by the AP, it is retransmitted *OD*+1 consecutive times. Consecutive retransmissions of such a message are performed by the AP cyclically at the end of the *poll-request* couple engaging station$_i$, i=1, ..,N, by properly setting the header of the broadcast message requested by station$_i$. Thus, the time interval between two consecutive retransmissions is $3t_m$.

The SAN used for the evaluation of the probability $P_{D>OD}$ is based on the replication of the model representing the reception of a broadcast by a generic station *i* (sub-model *station$_i$*), illustrated in Figure 1. In fact, thanks to the assumptions 4) above, the model of the whole system, for this measure, is simply given by a straightforward composition of the N (equal) models of the single stations, having in common the place *Fail.*

The activity *rec_bc* represents the reception of a message by the generic station$_i$. It is enabled when a token is in the place *active* and no token is present in the place *Fail*. *rec_bc* has an exponential distribution (rate $3t_m$) and has two possible outcomes (*case1* and *case2*) corresponding to a successful and un-successful reception

of the message (with probability (1-q_{bc}) and q_{bc} respectively). If the message is not received by the station, the number of tokens in *counter* is incremented (through the output gate *rec_fail*). If the message is received instead, the marking of *counter* is set to zero (through the output gate *rec_ok*), since we are interested only in consecutive failures in reception. Therefore, the marking of the place *counter* gives the number of consecutive broadcasts lost by station$_i$. As long as the marking of *counter* does not exceed OD, a token is put again in *active*, i.e. the station is ready to receive a new broadcast. Otherwise, the failure event whose probability we are looking for occurred, and a token is put in the place *Fail*.

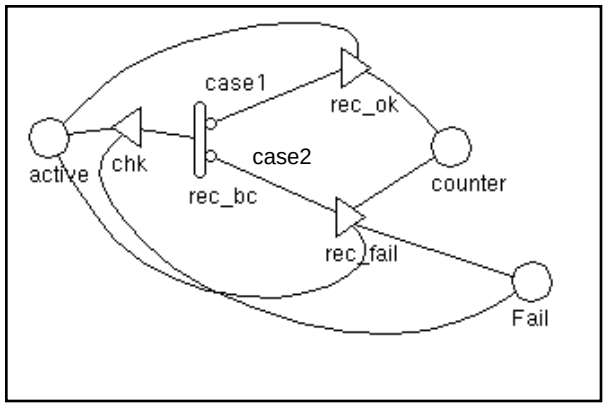

Figure 1. Sub-model *station$_i$* used to evaluate PD>OD

Through the *replicate* operation, a (parametric) number of SANs as described in Figure1 are combined to form the complete SAN of our system, the place *Fail* being in common among all the sub-models. The evaluation of $P_{D>OD}$ is obtained through a transient analysis at time T_{CFP}, by the use of a rate reward variable associated with the presence of a token in the place *Fail*.

3.2.2. SAN used to evaluate the probability for any station to miss a broadcast message after $OD+1$ retransmissions, in the time interval T_{CFP} ($P_{R>OD}$). As $P_{D>OD}$, $P_{R>OD}$ is still a probability of loosing $OD+1$ consecutive retransmissions of the same message, but this time we are interested in *broadcast messages* instead of *decision messages*. From the protocol behaviour, retransmissions of the same broadcast message occur once per round, where a round is composed of the sequence of actions "poll of station$_i$ by the AP - request to broadcast by station$_i$ - broadcast of the message requested by station$_i$", i=1, .., N. The time for completing a round is therefore $3t_m$*N, which becomes the time two consecutive retransmissions of the same broadcast message occur in.

Under the assumption of independence and of a constant probability of message failure at each broadcast, the SAN just described for measuring $P_{D>OD}$ can also be used for the evaluation of the probability $P_{R>OD}$. In fact, the only difference is in the time at which two successive broadcasts of a *broadcast message* occur wrt a *decision message;* this timing difference can be easily accounted for in the model by properly setting the rate of the activity *rec_bc* to $3t_m$*N.

3.3. Models for Performance-related measures

When a station is polled by the AP, it replies with a request for a message to be broadcast and with an acknowledgement (positive or negative) on the receipt of the N previous messages. Therefore, for the AP to consider a broadcast message msg_X to be received by station$_i$, it has to happen that: i) station$_i$ receives msg_X, and, ii) the pair of messages *poll-request* is correctly exchanged between AP and station$_i$.

For the generic message msg_X to be correctly broadcast, events at point i) and ii) have to occur for every station$_i$, i=1,..,N. The models for the performance related measures will be built based on the events at points i) and ii). Again, because of the assumption on independent failures, it is possible to define the performance models of the whole system as a composition of the N (equal) models, each one referred to a station.

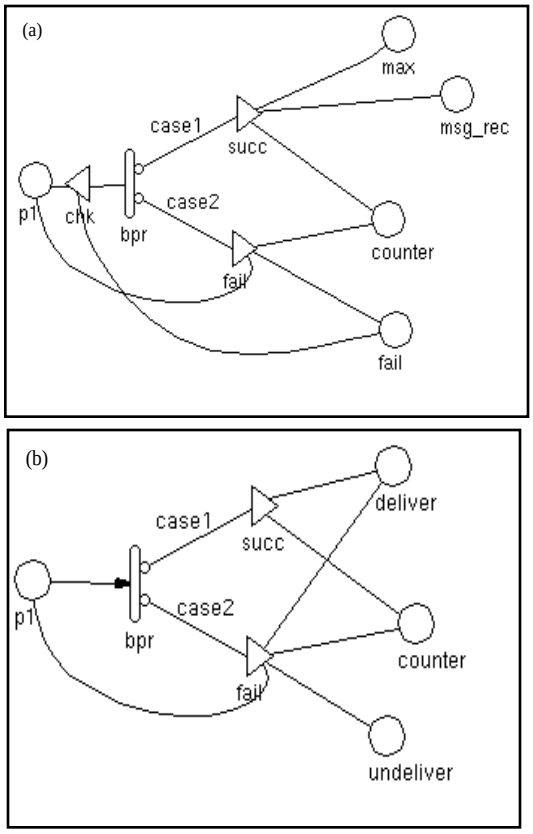

Figure 2. Sub-model *station$_i$* used to evaluate R_m (2.a) and P_{UM}(2.b)

3.3.1. SAN used to evaluate the average number R_m of retransmissions for a single message. As already introduced, to determine R_m, we focus on a generic $station_i$, and observe its behaviour with respect to the broadcast of a message msg_X. To know if msg_X has to be retransmitted or not (and, in the case, how many times), we have to consider the outcome of the composed event *broadcast-poll-request* between $station_i$ and the AP. The sub-model for $station_i$ is shown in Figure 2.a; the model for the evaluation of R_m is then given by N replications of this sub-model in which the places *fail*, *msg_rec* and *max* are in common.

The place *p1* represents the initial state of the message, and the timed activity *bpr* (with exponential distribution and rate = 1) the occurrence of the composed event *broadcast-poll-request*. Depending on the success or failure of such composed event, two outcomes are possible (*case1* and *case2* respectively). In presence of a failure, which occurs with probability $(q_{bc} + (1-q_{bc})q_{pr})$, the message has to be retransmitted: the tokens in the place *counter* are incremented so to count the number of retransmissions, and a new token is put in *p1*. If the number of retransmissions is greater than *OD*+1, a token is put in *fail* indicating a catastrophic failure of the protocol. If no failures occur, the tokens in *msg_rec* are incremented by 1 to count the number of stations for which the composed event is successful, and the marking of *max* is replaced by that of *counter*, if this last is higher. When all the sub-models terminate their execution, the marking of *max* is the actual number of retransmissions needed by the message.

3.3.2. SAN used to evaluate the probability P_{UM} that a broadcast message is not delivered to the applications (second version of the protocol). The model for the evaluation of P_{UM} is quite similar to the model for the evaluation of R_m. In this case, we have to trace when a decision to not deliver the message is taken, which is determined by the composed event *broadcast-poll-request* failing for more than *res(c)* times. In Figure 2.b, the sub-model for $station_i$ is shown. In this model, *p1* and *bpr* have the same meaning as in the model in Figure 2.a. As long as the number of failures of the event *broadcast-poll-request* (marking of *counter*) is less than *res(c)*, then a token is put in *p1* (and the message can be retransmitted). When *res(c)* retransmissions are exceeded, a token is put in the place *undeliver*. If no errors occur, then a token is added to *deliver*. The model for the evaluation of P_{UM} is then obtained by N replications of the sub-model for $station_i$, in which the places *undeliver* and *deliver* are in common. P_{UM} is determined as the probability of having one token in *undeliver*.

A consideration on the relationship between the values of *OD* and *res(c)* is noteworthy, in relation with the evaluation of P_{UM}. As stated in section 2, stations may decide to leave the group and stop sending messages to the AP. Thus when $station_i$ migrates, the AP misses to receive its acknowledgement on every message broadcast from the time $station_i$ migrated till the recognition of its departure (which needs *OD*+1 rounds to occur). If *OD* is set to a value much higher than *res(c)*, a significant number of broadcast messages will not be delivered to the applications (easily computed as (N-1)(*OD*+1)/*res(c)* messages). The volunteer migration phenomenon can significantly affect the usability of this version of the protocol because of the resulting unacceptable level of violation of the validity property. In such a case, a refined protocol could be defined, where each station has to notify the AP its intention to leave the group, retransmitting this message a certain number of times to cope with message failures (*OD*+1 times would guarantee the reception of this notification by the AP, under the assumption of the maximum omission degree stated in section 2.1).

4. Experimental analysis to determine model parameters

This section presents experimental measurements performed to determine the model parameters q_{bc}, q_{pr}, and t_m. The experiments were conducted using an implementation of the second version of the protocol based on a system of Windows NT 4.0 Workstations and Laptops connected by an IEEE 802.11 Standard compliant wireless network. All measurements have been carried out in an office environment under good physical conditions.

4.1. Determining q_{bc}

In order to determine the parameter q_{bc} (the probability that a certain station does not receive a broadcast message issued by the access point) we let each station count the number of broadcast messages it lost while running the protocol. This was easily achieved by using the global sequence numbers, which are assigned to the broadcast messages by the AP. A gap between the sequence numbers of two consecutively received broadcast messages indicates that some broadcast messages have been lost. For example, if station *s1* consecutively receives broadcast messages with global sequence numbers *50* and *53*, it can conclude that it has lost two broadcast messages. We then counted the number of messages lost by the stations while the access point transmitted a total number of 13355215 broadcast messages (see Table 2).

As expected, because of the different physical locations of the stations with respect to the AP, stations experience slightly different message loss rates; we used the mean value in the subsequent model evaluation.

Table 2. Number and rate of lost broadcast messages

	Station		
	s1	s2	s3
Lost broadcast messages	1280	2783	2549
Loss rate	$9.58 * 10^{-5}$	$2.08 * 10^{-4}$	$1.9 * 10^{-4}$

4.2. Determining q_{pr}

A further parameter of the model regarding the reliability of the medium is the probability q_{pr} of failure of the couple *poll-request*. Whereas the parameter q_{bc} describes the loss rate of single broadcast messages, the parameter q_{pr} describes the loss rate of couples of point-to-point messages. Since the access point transmits the polling message and receives the reply, if no message loss occurs, the number of losses of such couples can be locally observed by the access point without using sequence numbers.

To determine a realistic value for q_{pr}, we let the access point count the number of polling messages for which it did not receive a broadcast-request message from the polled station. During our experiments the access point transmitted a total number of 22845125 polling messages of which 13798 were not successfully replied by the polled station. This yields a loss rate of 13798/22845125 = $6.04*10^{-4}$. Again, in the models resolution, we assume that this rate is equal for all the stations.

4.3. Determining t_m

In case of our implementation of the protocol, the delay t_m that a message may experience between the instant it is sent and the instant it is received corresponds to delays of broadcast messages sent on UDP-sockets. Since the events of sending and receiving a message take place at different stations, we would normally require a global clock to measure the delay of a message. In order to avoid the need for a global clock we applied the following experimental setup: a process *ping* on station *s1* sends a message to process *pong* on station *s2.* Process *ping* includes a timestamp into the message immediately before transmitting it. Upon reception of the message, process *pong* simply returns it to process *ping*, which takes a second timestamp immediately after reception of the returned message. Since the process *pong* does nothing but calling the receive and send primitives, we suppose that the execution time of process *pong* that is not part of the transmission delay can be neglected. Thus, the difference between the two timestamps yields the round-trip-time of the message from *ping* to *pong* and back to *ping.* Assuming that delays between *ping* and *pong* in either direction are equally distributed, the delay of a message can be computed as half its round-trip-time. Applying the described experimental setup, we performed several delay measurements, for message sizes varying between 100 and 1000 Bytes, as depicted in Table 3.

Most of the measured delays are within a small range around the mean value, as the standard deviations indicate. Nevertheless, there are some comparatively huge values in each sample, which result in the big values for the maximum and the range. In the model resolutions, t_m is assigned the value corresponding to the average delay for message size of 1000 Bytes.

Table 3. Parameters of the delay distribution as a function of the packet size

	Packet size (Bytes)		
	100	500	1000
Sample length	2000	2000	2000
Min delay (μsec)	1631.5	3163	4689
Max delay(μsec)	5003	7811.5	14505
Mean delay(μsec)	2843	4995.24	7646.68
Standard deviation	96.08	398.26	395.91
Range	3371.5	4648.5	9816

5. Numerical evaluation of the defined QoS indicators

A numerical evaluation of the SAN models presented in section 3 has been carried out, by using the tool UltraSAN [5]. The basic parameter settings, partly derived from the previously described experimental evaluation, are summarised in Table 1. To fully exploit the combined usage of analytical and experimental evaluations, the numerical results of the performance figures obtained by resolving the SANs are also compared with corresponding measures determined experimentally.

Tables 4 to 7 concern dependability-related indicators. Table 4 shows the values of the probability $P_{D>OD}$ of missing *OD*+1 consecutive decision messages as a function of different values of the omission degree *OD*, for varying values of the time observation interval T (in a range which would be indicative of a Contention Free Period duration).

Table 4. $P_{D>OD}$ as a function of *OD*, for varying values of the observation time interval T.

	$T = T_{CFP}$ (sec)			
OD	60	180	600	2400
1	2.69E-04	8.08E-04	5.37E-04	1.07E-03
2	4.31E-08	1.29E-07	8.62E-07	1.72E-06
3	6.89E-12	2.07E-11	1.38E-10	2.76E-10
4	1.10E-15	3.31E-15	2.21E-14	4.41E-14
5	1.76E-19	5.29E-19	3.53E-18	7.06E-18

From the above table, it can be appreciated the dependency of $P_{D>OD}$ from the setting of both *OD* and T. De-

pending on the application requirements about an acceptable probability of catastrophic failure, the designer can then choose the appropriate value for *OD* for which the value of $P_{D>OD}$ is kept at the desired value. $P_{D>OD}$ is also influenced by the probability of message failure q_{bc}, which has been kept fixed to $1.6*10^{-4}$ in the previous analysis.

Table 5 reports the results of the evaluation of $P_{D>OD}$ varying q_{bc} and the time T. The impact of the last parameter, the number of mobile stations N, is less relevant under the hypothesis of independence of message failures; e.g., we determined variations in $P_{D>OD}$ from $5.52*10^{-15}$ to $2.76*10^{-14}$ when N=2 and N=10 respectively, and T=600 sec.

Table 5. $P_{D>OD}$ as a function of q_{bc}, for varying values of the observation time interval T.

q_{bc}	T (sec)			
	60	180	600	1200
1.6E-04	1.10E-15	0.33E-14	2.21E-14	2.21E-14
5 E-04	3.28E-13	9.86E-13	3.29E-12	6.57E-12
1 E-03	1.05E-11	3.15E-11	1.05E-10	2.10E-10
5 E-03	3.27E-08	9.81E-08	3.27E-07	6.54E-07
1 E-02	1.04E-06	3.12E-06	1.04E-05	2.08E-05

Now we report on the evaluation of $P_{R>OD}$, following the same approach as for $P_{D>OD}$.

Table 6. $P_{R>OD}$ as a function of *OD*, for varying values of the observation time interval T.

OD	$T = T_{CFP}$ (sec)			
	60	180	600	2400
1	6.73E-05	2.02E-04	6.73E-04	2.69E-03
2	1.07E-08	3.23E-08	1.08E-07	4.31E-07
3	1.72E-12	5.17E-12	1.72E-11	6.90E-11
4	2.74E-16	8.26E-16	2.76E-15	1.10E-14
5	4.38E-20	1.32E-19	4.41E-19	1.77E-18

Table 7. $P_{R>OD}$ as a function of q_{bc}, for varying values of the observation time interval T.

q_{bc}	T (sec)			
	60	180	600	2400
1.6E-04	2.74E-16	8.26E-16	2.76E-15	1.10E-14
5 E-04	8.17E-14	2.46E-13	8.21E-13	3.29E-12
1 E-03	2.61E-12	7.87E-12	2.63E-11	1.05E-10
5 E-03	8.13E-09	2.45E-08	8.18E-08	3.27E-07
1 E-02	2.59E-07	7.80E-07	2.60E-06	1.04E-05

It can be noted that values in Tables 6 and 7 are a bit lower than the corresponding ones in Tables 4 and 5; this is due to the different time interval between retransmissions of two consecutive decision messages and broadcast messages, as already discussed in section 3.2.2.

Next we discuss performance-related indicators. As previously mentioned, in order to understand the adequacy of the analytical models, and especially of the failure independence assumption on some of the evaluated measures, we have determined experimentally both the rate of messages lost by applications (to be compared with P_{UM}) and the throughput.

For the setting chosen, the results obtained from the evaluation of the average number of message retransmissions (including the initial broadcast), R_m, have shown to be very close to 1, slightly differing at varying values of the most impacting model parameters *OD* and q_{bc}. To give a numerical example, the higher value obtained in the range of q_{bc} ($[1.6*10^{-4} \div 10^{-2}]$) and *OD* ([1÷4]) analysed, R_m=1.172 when $q_{bc}=10^{-2}$ and *OD*=4.

Table 8. Probability of not delivering a message, derived both analytically and experimentally, as a function of *res(c)*, for varying values of q_{bc} (P_{UM})

	Probability of not delivering a message			
	Analytical derivation (P_{UM})			Experimental derivation
res(c)	q_{bc}=1.6E-04	q_{bc}=1.0E-03	q_{bc}=1.0E-02	
0	2.98E-03	1.85E-02	1.73E-01	5.808E-3
1	2.22E-06	8.67E-05	8.52E-03	6.714E-4
2	1.66E-09	4.04E-07	3.95E-04	9.572E-6
3	1.24E-12	1.88E-09	1.83E-06	0

Table 8 reports the values of the probability of not delivering a message to the applications, evaluated both analytically (solving the model for P_{UM}) and experimentally. In the experiments, each application included a local sequence number into its messages, so that the receiving application was able to detect lost messages. Since the omission degree assumption was not violated during these experiments, the messages not delivered are exactly those that are either never received by the access point or not acknowledged by all the stations.

Comparing the values in the column "Experimental derivation" with those in the column "$q_{bc}=1.6*10^{-4}$" (which is the average of the values of q_{bc} experimentally evaluated for each station, as reported in Table 2), one can notice a significant under-estimation of the percentage of messages not delivered to the application. To some extent this can be attributed to the fact that, in the experimental setting an end to end measurement is performed, whereas the model does not consider the application level. On the

other hand, the under-estimation is a clear symptom of the inadequacy of the independence assumption made, which we are going to relax in future work, as already anticipated. However, in the basic case where just one broadcast of a message is performed (*res(c)*=0) the experimental and analytical measures match.

The last indicator we consider is the throughput T_{hr}. The analytical evaluation of T_{hr} has been based on the average number of message retransmissions R_m and the average message delay t_m. To perform such an estimate, we have taken into account the time spent in performing the *poll-request* couple, which has to be subtracted to obtain the time effectively spent for broadcasting messages. Because the poll message is reasonably much shorter than the request or the broadcast message, it has been assigned a shorter t_m (taken as the average delay for messages of size 100 Bytes in Table 3, i.e., $t_m^{\wedge}$=2843 μsec). Therefore, the formula for the throughput can be expressed as $T_{hr}= 1/((t_m^{\wedge} + 2t_m)*R_m)$. In order to determine the throughput experimentally, we counted the number of broadcast messages delivered by the second version of the protocol during a certain period of time, in a number of experiments conducted for varying *res(c)*. T_{hr} was then computed by dividing the number of delivered messages by the elapsed time.

Tables 9 and 10 show the values of T_{hr} as determined by the analytical and experimental evaluation, respectively. In Table 9, the values of the throughput have been computed for both versions of the protocol and it can be observed that the difference is very small (for completeness, the corresponding value of R_m, is also given).

Table 9. Throughput determined analytically, as a function of q_{bc}.

		T_{hr} [msg/s]	
q_{bc}	R_m	version 1	version2
0.00016	1.0029787	55.39056368	55.39056359
0.001	1.0184957	54.54667659	54.54665455
0.005	1.0184957	50.88415131	50.88160697
0.01	1.1725012	47.38208844	47.36337252

Table 10. Throughput determined experimentally, as a function of the resiliency

res(c)	T_{hr} [msg/s]
0	57.00
1	57.71
2	57.87
3	57.97
15	57.77

Interestingly, the throughput as depicted in Table 10 is not significantly affected by the resiliency. Actually message losses are a small fraction of the total number of messages transmitted and the number of retransmissions is low, this explains why the impact of the resiliency is minor. This result would suggest that using the (more complex) protocol version with both *res(c)* and *OD* is not worthwhile if only throughput considerations are taken into account; however, if real-time requirements have to be accounted for, restricting the retransmissions to *res(c)* allows meeting the timing guarantees. The values in Table 10 for *res(c)*=2 are very similar to those in Table 9 (q_{bc} set to $1.6*10^{-4}$); the difference is mainly due to the fixed (average) value of t_m used in the analytical evaluation.

6. Concluding remarks and future work

In this paper, we have applied a QoS analysis to a family of group communication protocols resorting to both analytical modelling and experimental evaluations. Appropriate QoS metrics have been identified, which have been classified in dependability related measures and performance related ones. Specifically, the dependability-related figures aim at giving an estimate of the coverage of the assumptions on which the protocols rely, while the performance figures can be used as indicators of the technical limitations imposed by the communication system and the way the protocol behaves according to them. We used both experimental and analytical methods, and exploited their integration. The measurement results obtained by experimental settings in a realistic environment have been used to determine reasonable parameters for the analytical model. Furthermore a validation of the adequacy and usability of the analytical model has been made by comparing experimental results with those obtained by model resolution. This validation activity led us to determine that the analytical models defined so far are not yet fully adequate for the estimation of the QoS provided to the applications. This is largely due to the assumption made of independence regarding the message failure. As we already knew, this assumption is clearly inadequate. Nevertheless, since it allows simplicity in model definition and resolution, we wanted to determine and verify which was the impact of such an assumption and whether the results obtained could have been of some use. For those measures and setting where independence has not a significant impact, instead, the analytical models provide results matching with those obtained through the experimental activity.

The work performed so far is still uncompleted; however, it has allowed to gain insights on the model behaviour, useful for appropriate choices and refinements. In this respect a few interesting considerations have been developed and discussed in the paper. We are going to extend this work with the objective, on one side, to determine experimentally the forms and the extent of

correlation among the events of interest; and on the other to refine the analytical models by relaxing independence and account for correlation.

Acknowledgement

We would like to thank Professor Edgar Nett for his helpful advises and support.

This work has been performed in the framework of a formalised co-operation between GMD (Germany) and CNR (Italy), more precisely, in the context of the DECOR project. Also, this work has been partially supported by the project CNR ASWA.

References

[1] IEEE 802.11, "IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications." 1997.

[2] V. Hadzilacos and S. Toueg, "Fault-tolerant Broadcasts and Related Problems," in *Distributed Systems*, S. J. Mullender (Eds.), Reading, Addison-Wesley, 1993, pp. 97-145.

[3] M. Mock, E. Nett and S. Schemmer, "Efficient Reliable Real-Time Group Communication for Wireless Local Area Networks," *3rd European Dependable Computing Conference*, Prague, Czech Republic, 1999, pp. 380-397.

[4] W. H. Sanders and J. F. Meyer, "A Unified Approach for Specifying Measures of Performance, Dependability and Performability," in *Dependable Computing for Critical Applications, Vol. 4: of Dependable Computing and Fault-Tolerant Systems*, (Eds.), Springer-Verlag, 1991, pp. 215-237.

[5] W. H. Sanders, W. D. Obal, M. A. Qureshi and F. K. Widjanarko, "The UltraSAN Modelling Environment," *Performance Evaluation Journal, special issue on Performance Modelling Tools*, vol. 24, pp. 89-115, 1995.

[6] B. Teitelbaum, J. Sikora and T. Hanss, "Quality of Service for Internet2," *First Internet2 Joint Applications/Engineering Workshop: Enabling Advanced Applications Through QoS*, Santa Clara, CA, 1998, pp. 5-16.

[7] A. Vogel, B. Kerherve and G. Von Bochmann, "Distributed Multimedia and QOS: A Survey," *IEEE Multimedia*, vol. 2, pp. 10-19.