Model-based analysis of a protocol for reliable communication in railway worksites

L. Montecchi[1], P. Lollini[1], B. Malinowsky[2], J. Grønbæk[2], A. Bondavalli[1]
[1]University of Florence
Viale Morgagni, 65
I-50134, Firenze, Italy
{lmontecchi,lollini,bondavalli}@unifi.it
[2]Forschungszentrum Telekommunikation Wien
Donau-City-Straße 1
A-1220 Wien, Austria
{malinowsky,gronbaek}@ftw.at

ABSTRACT
In this paper we perform a model-based analysis of the Timed Reliable Communication (TRC) protocol, which is being used within the EU funded ALARP project for railway worksite communication. TRC is a group communication protocol based on IEEE 802.11 networks, targeting safety-critical applications with limited bandwidth requirements. The paper contains an in-depth analysis of the performance and reliability characteristics of the protocol using a Stochastic Activity Networks model. The results are first compared with available experimental measurements for the sake of model validation. The validated model is then used for a thorough analysis of a set of key metrics under different environment and network conditions. The obtained results allow: i) to assess that the protocol allows to satisfy the ALARP targeted performance and reliability requirements, and ii) to evaluate the existing tradeoffs and help in choosing parameter values for the final implementation.

Categories and Subject Descriptors
B.8.2 [**Performance and Reliability**]: Performance Analysis and Design Aids; C.4 [**Performance of Systems**]: Modeling Techniques; Reliability, availability and serviceability.

General Terms
Performance, Reliability, Human Factors, Verification.

Keywords
broadcast, wireless, 802.11, ALARP, reliable timed communication

1. INTRODUCTION
Wireless communication technologies are rapidly improving, and replacing wired communication links in several application scenarios. Moreover, wireless devices are now easily embedded in everyday appliances, like telephones, cars, televisions, printers, and cameras. Technologies based on the IEEE 802.11 standard [14] allow indoor, office and home connectivity at low costs, 802.15 [24] supports wireless broadband access (e.g., WiMax), while 802.16 [25] allows communication within a limited range and it is commonly used for personal devices (e.g., Bluetooth).

The spread of such technologies allows for applications that were not possible in the past. However, ensuring reliable and timely message transmission in wireless networks is still a challenge. Wireless communication links reliability depends on the actual distances between nodes, which varies in mobile settings, on the presence of obstacles, and on possible interference from other transmitting devices. Executing critical distributed services and achieving the required reliability, safety and timeliness of communication on wireless devices requires a careful design and analysis of communications.

In this paper, we analyze a message dissemination protocol that is being used in building an Automated Track Warning System (ATWS) for the safety of railway workers, based on wirelessly connected personal devices. The analysis is performed using a Stochastic Activity Networks model, which is evaluated using a numerical solver. The results are first compared with available experimental measurements for the sake of model validation. The validated model is then used for a thorough analysis of a set of key metrics in different scenarios.

This paper is organized as follows. Section 2 describes the context in which analyses are drawn and their motivations, while the analyzed protocol is described in Section 3. Related work is reviewed in Section 4; the modeling approach and the target measures of interest are described in Section 5, while the evaluation and results are discussed in Section 6. Finally, conclusions are drawn in Section 7.

2. THE ALARP SYSTEM
ALARP ("A railway automatic track warning system based on distributed personal mobile terminals" [1]) is a research project funded within the Seventh Framework Programme (FP7). Its objective is to design and develop an innovative Automatic Track Warning System (ATWS), to improve the safety of trackside workers.

2.1 System Description
The ALARP ATWS is able to inform the trackside workers about dangerous events within the worksite, e.g., approaching trains on the track, maintenance events on power lines and/or safety equipment that may put at risk workers' safety (e.g. being hit by a train or by an electric shock), emergencies on tracks and tunnels nearby the workers (e.g. fires in a tunnel, toxic smoke, etc.). Additionally, it keeps track of the status and position of the workers, in order to identify and localize those at risk, suggest escape routes, or provide information in case of rescue.

The ALARP architecture is COTS-based and involves the following main components: i) the track-side train presence alert device (TPAD), able to sense an approaching train on the interested track, ii) a set of wearable wireless Mobile Terminals (MTs), to inform the workers about possible approaching trains and other events that could put at risk their safety, and iii) infrastructure for wireless communication [2].

The MT has to communicate and interact through wireless connections with other MTs and the TPADs. Most importantly, the MTs need to be capable to receive information about approaching trains from the TPADs, and notify it to the workers. Railway regulations require a Controller Of Site Safety (COSS) to supervise the workers team and take care of safety concerns. The ALARP system supports the COSS by providing useful information about the location of workers and the state of their MTs.

The ALARP communication architecture joins two technologies in a two-hop setup. Worksite communication relies on off-the-shelf IEEE 802.11 [14] technology, having limited range but some properties that can be exploited for timely communication. A coordinator (i.e., an Access Point) provides access to the medium and serves as gateway for Mobile Terminals. TPADs communicate with the worksite via longer range wireless technology, delivering messages to the coordinator at the worksite for dissemination. In this paper we focus on the protocol that is used for communication at the worksite.

2.2 Relevant Requirements

The communication protocol adopted in ALARP shall allow the communication layer to reliably distribute messages. It should support broadcast (distribution to all nodes), multicast (one-to-many communication), and unicast (message exchange between two dedicated nodes) communication primitives. The communication layer defines three criticality levels: high for system safety-critical messages (LEVEL HIGH), medium for messages affecting system availability (LEVEL MEDIUM), and low for messages with no requirements (LEVEL LOW, best effort) [12].

According to ALARP requirements [2], the alert of an approaching train shall reach the workers within the time interval $[T_0, T_1]$ before the train approaches the worksite, i.e. the worker should be alerted at time t, $T_1 > t > T_0$, where T_0 is the minimum allowed time to reach a safe position and T_1 is a programmable value that depends on national regulations.

Typically, the TPAD would be placed at distance such that the train will take about one minute to reach the worksite, and the minimum time T_0 for workers to reach the safety position would be around 45 seconds. Therefore, an informal derived requirement on worksite communication states that any broadcast should take a maximum of 10-12 seconds in the worst case.

3. THE TIMED RELIABLE COMMUNICATION PROTOCOL

In order to satisfy the ALARP requirements, [6] defined the ALARP Timed Reliable Communication (TRC) protocol, by properly adapting the existing Real-time Group Communication Protocol (RGCP) [8,9]. The latter was originally defined to support real-time group decisions between autonomous robots communicating through wireless networks.

3.1 The Reliable Group Communication Protocol (RGCP)

RGCP relies on the IEEE 802.11 Point Coordination Function (PCF) [14] and the existence of a coordinator (Access Point) to realize a polling scheme of nodes, and allocating fixed time slots using a round-based approach. During the "Contention Free Period" (CFP), a central coordinator controls the access to the medium for a group of stations. Every station remains silent until it is polled by the access point; then, the AP grants exclusive access to the medium by transmitting the polling message.

The RGCP protocol is based on the following assumptions [8,9]:

- Messages delivered during the CFP are delivered correctly within a fixed time-bound (t_m).
- Messages may be lost (omission faults).
- Message losses are asymmetric; i.e., some stations may receive a broadcast message and some may not.
- The number of consecutive message losses is bounded by the omission degree OD.
- Stations may crash or leave the reach of the AP.
- The AP is not subject to any kind of error.

RGCP satisfies the properties of validity, agreement, and integrity [8,9]. The communication is structured into rounds, and each round into N slots, with N being the amount of nodes. During each slot, three messages are exchanged: "poll", "request", and "broadcast". In each slot, the AP sends a poll to a station, which returns a "request" message to the AP, containing the message to be broadcast. The AP assigns a sequence number to that message and broadcasts it to the group of stations. The "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. Therefore, 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. Each message is retransmitted at most $OD+1$ times to ensure, according to assumptions, that it is correctly delivered to all nodes. If the AP does not receive the request message from a station for $OD+1$ consecutive times, it considers that station to have left the group and broadcasts a message indicating the change in group membership [8,9].

A variant of the protocol, called RGCP with resiliency degree, allows the user to specify the maximum number of retransmissions of the messages. This bound on message retransmissions (called resiliency degree $res(c)$) may vary for different message classes c. Choosing $res(c)$ smaller than OD 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). 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 [8,9].

3.2 Modifications for the ALARP system

The RGCP with resiliency degree allows trading the probability of broadcast delivery for resource utilization, which is a useful feature for the ALARP system: less critical messages could be transmitted using a lower resiliency degree, thus reducing network utilization and leaving room for possibly arriving safety-critical messages.

However, RGCP with resiliency degree ensures many more properties than are really necessary for the ALARP system. More in detail, it guarantees ordering and agreement through a final decision message. The ALARP system does not require neither ordering nor agreement; therefore these two properties can be relaxed. Practically, this simply consists in removing the last part of the protocol, in which the decision message is sent [6]. Additional modifications have been made with respect to the original protocol, in order to adapt to the ALARP requirements and architecture. In particular, three different message classes have been defined, mapping the three ALARP message classes: *res(LH)*, *res(LM)*, and *res(LL)*.

Protocol implementation relies on off-the-shelf equipment, fully compatible with the 802.11 standard. Although developed for the ALARP system, the protocol is therefore applicable to a wider range of COTS-based critical applications and infrastructures.

4. RELATED WORK

Communications are a fundamental part of modern systems and several works have been devoted to the analysis of their performance and dependability properties. Concerning wireless communication, a large collection of works focuses on physical properties of the wireless medium. The authors of [15] describe the physics of signal propagation in different environments, considering both indoor and outdoor scenarios. Other works focus on defining good approximations of the physical channel through a channel model; a nice survey on this topic can be found in [18]. Another set of works focuses on the analysis of a specific radio technology or protocol: in [20] and [21] the transmission of multicast packets in wireless LANs is analyzed, while in [16] capacity estimation for downlink WCDMA networks is performed. Conversely, [17] focus on the uplink direction, also considering inter-to-intra cell interference. The authors of [19] evaluate and improve the performance of the MAC protocol in 802.11 networks, in a dedicated short-range communication environment. Sensor networks are taken into account in [22], where a secure routing protocol is designed and analyzed. In [5] stochastic models and measurements are combined to analyze the performance of a consensus algorithm.

The original RGCP protocol has already been analyzed in a number of different ways. In [11] the protocol has been analyzed combining a Stochastic Activity Networks model and measurements; the model has then be refined in [3] to relax the assumption of independent losses, taking into account for loss correlation among consecutive packet transmissions. A modified version of the model in [11], for the analysis of the same protocol, has been proposed also in [23], although some major flaws impair the model, which actually does not represent the same system.

In this paper we take an orthogonal approach to the analysis in [11] and [3], where the main focus was broadcast throughput under continuous operation. In this paper we focus on transmission delay and delivery probability of a single application-level broadcast message, which are the main concerns in the ALARP system. Moreover, in this paper we also consider degraded protocol execution, as opposed to a simple failure/success model as in previous evaluations of the RGCP protocol. Finally, wireless technology has considerably improved over the past 10 years, and therefore we base our analysis on more recent experimental measurements, performed within the ALARP project.

A parallel evaluation work of the ALARP time reliable communication has been performed also in [6], where an experimental worksite test setup has been established, and results then used to parameterize a stochastic simulation model. The experimental setup used in such work is aligned to the ALARP system communication requirements (low bandwidth, but strict bounds on time behavior) and investigates cross-traffic and interference scenarios using unlicensed bands. In particular, a scenario with contending nodes and saturated channel conditions representing a highly challenging IEEE 802.11 communication environment is considered, as encountered in dense urban worksite setups. After successful validation against the experimental results, the simulation model has been used to generalize the experimental results to common railway worksite settings.

The results in [6] provide an input to support our analytical approach, in which we perform an extended analysis of TRC performance and selected quantitative properties. Further details on these aspects are provided in Section 6.2. The main advantage of adopting an analytical solution approach relies in the accuracy of the computed solution, and in the avoidance of the rare event problem, in which a large number of simulation runs need to be executed to be able observe events that rarely occur. This is especially relevant in safety analysis as performed in this paper, where the objective is to evaluate the occurrence of hazardous events, which need to be extremely rare for the system to fulfill its requirements.

5. MODELING APPROACH

The main objective that drives this analysis is to assess the effectiveness of the TRC protocol in guaranteeing timely and reliable delivery of the safety-critical messages exchanged by the components of the ALARP system. In our approach, we focus on the transmission of a single application-level message using the TRC protocol, and aim to evaluate its quantitative properties.

In the following we will only consider broadcast messages, however the analysis can be easily adapted to multicast and unicast messages as well. The main reason is twofold: i) the most critical message in the ALARP system is a broadcast message (the "RISK_EVENT" message informing the workers of approaching trains), and ii) broadcast messages constitute the worst case, both in terms of successful delivery probability, and in delivery delay.

5.1 Measures of interest

Following from the definition of the protocol, the transmission of a message terminates when any of the following occurs: i) the poll-request fails for *OD+1* consecutive rounds, ii) the same message is broadcast for *res(c)+1* consecutive rounds, or iii) the coordinator receives the acknowledgment from all the recipient nodes. When any of these events occur, the execution of the protocol for that message terminates, and the next application-level message is processed.

Upon termination, the result of message transmission may be categorized in four classes. To precisely identify them, we first define the following three predicates, which are evaluated at the end of message transmission:

- PR_FMAX is true if the poll-request has failed *OD+1* consecutive times for that message.

- BC_FMAX is true if the message broadcast has failed *res(c)+1* consecutive times, i.e., if the message has not been delivered to all the nodes within *res(c)+1* rounds.
- ACK_OK is true if the coordinator has received the acknowledgements from all the recipient nodes.

Based on such predicates, we now define the four possible results of message transmission, which identify four different degrees of success in disseminating the message to all nodes:

- "Complete success"
  $csucc = \neg PR\_FMAX \wedge \neg BC\_FMAX \wedge ACK\_OK$
- "Partial success"
  $psucc = \neg PR\_FMAX \wedge \neg BC\_FMAX \wedge \neg ACK\_OK$
- "Dissemination failure"
  $disfail = \neg PR\_FMAX \wedge BC\_FMAX$
- "Poll-Request failure"
  $prfail = PR\_FMAX$

It is easy to verify that the four cases above are disjoint, and that they represent the universe with respect to the possible outcomes of a message transmission using TRC. Our analysis aims to evaluate the relative probability that sending a message using the TRC protocol results in each of the four cases. Distinguishing between them will help us to better understand how the system parameters affect the execution of the protocol, and also to assess the improvements that TRC protocol introduced for the ALARP scenario over the reference RGCP protocol.

It should be noted that, from a strict safety point of view, both the failures defined above represent a catastrophic failure (i.e., workers are not notified of an approaching train). However, for the purpose of the analysis, it is of interest to distinguish between them. Firstly, the two different failures depict two completely different scenarios. If a "Poll-Request failure" occurs, it means that the message has not even been able to reach the coordinator, and therefore nobody (including the COSS) will be alerted of the approaching train. Conversely, if the "Dissemination failure" occurs, a part of the MTs may still have received the message, and thus part of the workers may be aware of the approaching train. Although a "Dissemination failure" would still be a violation of ALARP safety requisites, means to mitigate its consequences would still be applicable, especially considering the "physical" interaction and communication between workers, and the supervising of the COSS on the worksite. Secondly the "Partial Success" outcome affects both resource utilization and false alarms, since the coordinator may wrongly believe that the broadcast has not been received by all the MTs.

Another measure which is of interest is the time required to execute the protocol. Using our model-based analysis we are able to evaluate the average time required to execute the protocol and, most importantly, its probability distribution, which allows to accurately assess the ability of the protocol to adhere to the time bounds imposed by the ALARP system. To summarize, using our model, the following measures of interest can be evaluated:

- P_{csucc}, P_{psucc}, P_{prfail}, $P_{disfail}$: Relative probability that each of the four cases occur in the transmission of a message using the TRC protocol ("Complete Success", "Partial Success", "Poll-Request Failure" and "Dissemination Failure", respectively).
- $T_{deliver}$: Time required the deliver the message to all the recipients, measured from the instant at which the coordinator polls the sender for the first time.
- $N_{deliver}$: Number of nodes that have received the message when the protocol ends.
- T_{end}: Duration of a single execution of the protocol. It measures the interval between the instant at which the coordinator polls the node for the first time, and the instant at which it polls it again asking for the next application-level message.

5.2 Modeling assumptions

Here we summarize the assumptions that have been made in the construction of the model, in addition to those introduced by the definition of the protocol (see Section 3).

- All the nodes connected to the coordinator are communicating using the TRC protocol only. This assumption completely holds in the ALARP scenario, since all the messages at the worksite (broadcast, multicast, and unicast) are transmitted using the TRC protocol, with different values of the *res(c)* parameter based on the reliability class of the message.
- Any application-level message is transmitted within a single 802.11 packet. This assumption holds within the ALARP system, because of the limited amount of data to be transmitted.
- A single unicast packet may be lost with probability p_m. This probability holds both for the "poll" and "request" messages. The fault model assumed in the definition of the protocol considers only "omission" faults, i.e., messages may not be delivered. This assumption characterizes the fault model from a quantitative point of view.
- A broadcast (or multicast) packet may not be received by a given station with probability p_b.
- Homogeneous links. All the links are homogeneous, i.e., they are characterized by the same properties.
- Independent losses. The message loss probability of a message sent by a station *i* is independent from losses of messages sent by other stations, as well as losses of previous messages sent by the same station.

The above is a reasonable set of assumptions that are often used in the modeling of wireless communication. Moreover, these assumptions are in part motivated also by the objective of: i) obtaining a model that can be solved analytically, and ii) using existing experimental measurements as input parameters.

5.3 Stochastic Activity Networks model

The model that will be used for our analyses has been created using the Stochastic Activity Networks (SAN) formalism [7,10], using the Möbius modeling framework [4]. Stochastic Activity Networks are an extension of Stochastic Petri Nets (SPN), having a powerful set of primitives that allow specifying very complex stochastic processes. The firing time of activities ("transitions" in classic Petri nets terminology) can follow different kinds of probability distributions, although the use of non-exponential activities (with some particular exceptions) prevents applying analytical solution techniques.

As described in Section 3, the TRC protocol is a slot-based protocol, in which the time advances in deterministic steps. Since

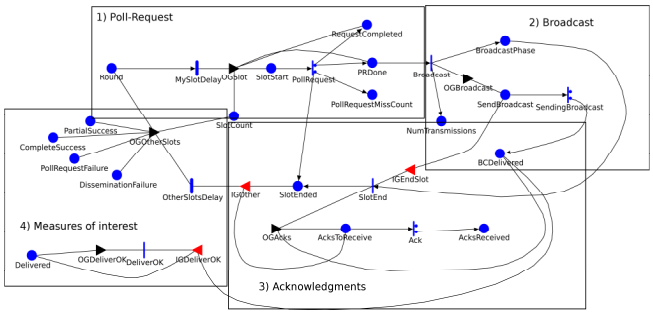

Figure 1. The complete SAN model for the TRC protocol, and the four parts in which it is organized.

the duration of slots is fixed, it suffices to count the number of elapsed slots to evaluate time-related quantities. For this reason, we are only interested in the number of firing of activities, and not in the actual elapsed time. Moreover, no concurrency behavior is present in the model, and therefore time distribution of activities has no impact on the computed measures of interest. Since we target an analytical solution, we choose to have only exponentially distributed (and instantaneous) activities; moreover, the rate of all exponential activities is set to one. Actually, the model could be implemented as a Discrete-Time Markov Chains (DTMC) as well; however a different model would be needed for each combination of *OD*, *res(c)*, and total number of nodes, thus making it difficult to analyze different scenarios.

The overall SAN model is shown in Fig. 1, and it basically consists of four parts: i) poll-request, ii) broadcast, iii) for acknowledgments, and iv) support for measures of interest. Each of these parts will be discussed in details in the following subsections.

5.3.1 *Poll-Request*

The first step in sending a message using the TRC protocol is performing the poll-request message exchange with the coordinator. The corresponding part of the model is highlighted in Fig. 2.

Initially there is a token in place "Round", which means that we are at the beginning of a round of the protocol. For simplicity we assume that the slot reserved to the sender node is the first slot in the round. The elapsing of such slot is represented by the activity "MySlotDelay"; when it fires, the code in the output gate "OGSlot" is executed. First of all, a token is added in place "SlotCount", which counts the number of slots that have elapsed in the execution of the protocol. Then, it is checked if the poll-request has already been completed, (i.e., if the coordinator has received the message to be disseminated to the other nodes), by checking if there is a token in place "RequestCompleted".

If the poll-request part has not been completed yet, a token is added in place "SlotStart", causing the instantaneous activity "PollRequest" to become enabled and fire. The activity has two

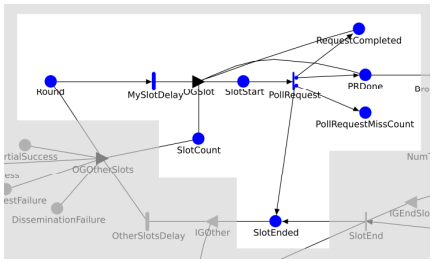

Figure 2. SAN model for the "poll-request" part.

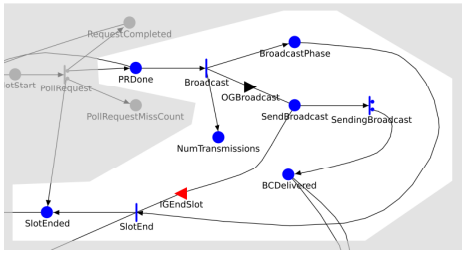

Figure 3. SAN model for the "broadcast" part.

cases, representing the success and failure of the message exchange, respectively. The successful case occurs when both the "poll" and "request" messages are delivered correctly; the associated probability value is therefore $(1–p_m)(1–p_m)$. The failure case occurs with the complementary probability value of $1–(1–p_m)(1–p_m)$. If the poll-request succeeds, a token is added to "RequestCompleted" and "PRDone" places. Otherwise, a token is added to place "PollRequestMissCount", which counts the number of consecutive poll-request failures, and to place "SlotEnded", to model the end of the slot reserved for the node.

5.3.2 *Broadcast*

The part of the model that represents the actual broadcast is depicted in Fig. 3. When place "PRDone" contains a token, the "Broadcast" activity is enabled and fires, removing the token from "PRDone", and adding a token in places "BroadcastPhase" and "TransmissionsCount", which counts the number of broadcast transmissions that have been performed for the current message. Place "BCDelivered" holds the number of nodes that have successfully received the broadcast message. When the activity "Broadcast" fires, the code in the gate "OGBroadcast" checks if the broadcast has already been delivered to all nodes, by checking the number of token in "BCDelivered".

If the broadcast has not been delivered yet, the gate adds in place "SendBroadcast" a number of token equal to the number of nodes that still have to receive the broadcast. This number correspond exactly to $(N–1)$ minus the number of tokens in place "BCDelivered", where N is the total number of nodes in the scenario. When a token is in place "SendBroacast" the activity "SendingBroadcast" is enabled and fires, removing the token from "SendBroadcast". The activity "SendingBroadcast" has two cases, corresponding to the successful reception of the broadcast by the node (occurring with probability p_b), and to its loss. If the broadcast succeeds, a token is added in place "BCDelivered".

Finally, when place "BroadcastPhase" contains a token and "SendBroadcast" is empty, the instantaneous activity "SlotEnd" fires, removing the token from place "BroadacastPhase" and adding a token in place "SlotEnded", which models the end of the slot reserved to the sender of the message.

5.3.3 *Acknowledgments*

In the subsequent timeslots (each one reserved for the other nodes in the connected set), the coordinator will poll each of the other nodes, and possibly receive the acknowledgments piggybacked to "request" messages. Then, in the following round, within the slot reserved to the sender node, the coordinator will check if it has received all the $N–1$ acknowledgments. If it is not the case, and the maximum number of retransmissions has not been reached, it will retransmit the broadcast.

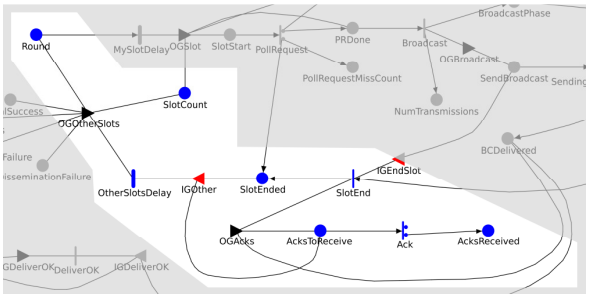

Figure 4. SAN model for the "acknowledgments" part.

The part of the model that represents this aspect of the protocol is shown in Fig. 4. It is important to note that, for our measures of interest, we are not interested in the order in which the coordinator receives the acknowledgments, neither in which acknowledgments it has already received. Instead, we are interested in the number of nodes that have acknowledged the message; this number corresponds to the number of tokens in place "AcksReceived", which is initially empty.

When the activity "SlotEnd" fires, the output gate "OGAcks" is also executed. The effect of this gate is to add in place "AcksToReceive" a number of tokens equal to the number of nodes that i) have received the broadcast, and ii) from which the coordinator has not yet received the acknowledgment, which is exactly the number of tokens in place "BCDelivered" minus those in place "AcksReceived". The activity "Ack" models the transmission of a single acknowledgment message, and fires once for each token in place "AsksToReceive". The coordinator receives the acknowledgment from a given node only if the poll-request succeeds with that node. Therefore, the probability that the coordinator receives the acknowledgment from a given node (within one round) is the probability that the poll-request succeeds, which equals to $(1–p_m)(1–p_m)$ as discussed above. The probability that the acknowledgment is not received is given by the complementary probability $1–(1–p_m)(1–p_m)$. If success occurs, a token is added in place "AcksReceived".

When the number of tokens in "AcksToReceive" reaches zero, the "Ack" activity becomes disabled, while the activity "OtherSlotsDelay" becomes enabled and fires, representing the end of current round. The output gate "OGOtherSlots" adds $N–1$ tokens in place "SlotCount", to record that additional $N–1$ slots have elapsed (one for each of the other nodes). The gate also checks the stopping conditions for the protocol. If they are not satisfied, a token is added again in place "Round", and the next round is started; otherwise the transmission of the current message ends.

5.3.4 Support for measures of interest

To support the evaluation of the measures of interest defined in Section 5.1 some additional elements have been added to the SAN model (Fig. 5). The activity "DeliverOK" fires as soon as the broadcast is successfully delivered to all nodes (i.e., the number of tokens in place "BCDelivered" reaches $N–1$). The output gate "OGDeliverOK" adds to place "Delivered" the current number of tokens that are hold in place "SlotCount", thus also recording the number of slots that have elapsed. For each of the four outcomes defined in section 5.1 an additional place is added to the model (places "TotalSuccess", "PartialSuccess", "DisseminationFailure" and "PollRequestFailure"), and they are used to record the outcome of protocol execution. While checking the stopping conditions for the protocol, the output gate "OGOtherSlots" also checks the conditions for the different outcomes, and sets the number of token in the corresponding place to the number of tokens in "SlotCount".

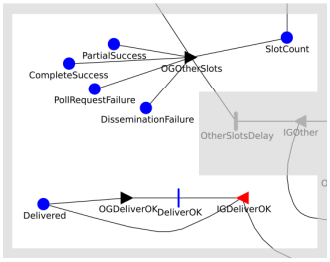

Figure 5. SAN model for the "measures of interest" part.

The relative probability that message transmission results in one of the defined outcomes (P_{csucc}, P_{psucc}, P_{prfail}, $P_{disfail}$) is evaluated by evaluating the probability that the number of tokens in the related place is greater than zero at the end of the protocol. Similarly, the probability distribution of the time required to deliver the broadcast to all the nodes ($T_{deliver}$) is obtained by evaluating the probability distribution of the number of tokens in place "Delivered". The duration of the broadcast (T_{end}) can be evaluated by evaluating the number of tokens in place "SlotCount", while the number of nodes that have received the message ($N_{deliver}$) is evaluated by counting the number of tokens in place "BCDelivered".

The evaluation is performed by transient analysis, using a sufficiently large instant of time to ensure that the termination of the protocol is reached. Due to the nature of the protocol, termination is reached in at most $N(OD+res(c))+1$ slots; therefore, we can derive the maximum number k of consecutive exponential activities that should fire to reach termination. The selected instant of time t should then allow all the k activities to fire, i.e., the sum of the k exponentially distributed random variables should be smaller than t with a probability that approaches one. All the exponential variables in the model are independent and identically distributed; therefore, their sum is described by an Erlang distribution with shape k. It is then easy to verify if the selected value of t is really sufficiently large; in our analyses we have used $t=10000$, which corresponds to a probability lower than 10^{-50} that, in the worst case, some activity has still not fired.

6. EVALUATION AND RESULTS

6.1 Model parameters and settings

The model described in Section 5.3 is based on a set of parameters, which are listed in Table 1. Some of these parameters are part of the environment, and the system has no control on them, e.g., the message loss probabilities pm and p_b, or the number of nodes N. Instead, other parameters are implementation-dependent and should be chosen with care. Although the number of parameters is limited, each of them has a great impact on the measures of interest, as it is shown in the following sections.

Table 1 also shows the default values that will be used in the following evaluations, which have been carefully selected, based on system requirements, typical working conditions, and existing experimental measurements. The number of nodes in the scenario (N) has been chosen based on information from railway expertise within the ALARP project, stating that the "typical maximum" number of workers involved in a worksite is 20 [2]. Most worksites will probably feature a smaller amount of workers,

however for safety reasons it is preferable to analyze worst-case scenarios. Concerning the duration of the timeslot we based on measurements and simulations described in [6], where a timeslot duration of 25 ms is suggested as a lower bound to have sufficient time to deliver the "poll", "request" and "broadcast" messages in the same slot.

As discussed in Section 2.2, it has been agreed within ALARP that the maximum time to send the RISK_EVENT message should be around 10 seconds. After such amount of time all the nodes should have received the information, or know that they have missed it. The nodes that do not have received the message will be aware of that by receiving a message with a greater sequence number, or because they will be disconnected by the access point. The worst-case execution time is $N(OD+res(LH))+1$ protocol slots. By assuming $OD=res(LH)=10$, in a scenario comprising 20 nodes, and using a timeslot of 25 ms, this quantity is still within the acceptable bound (exactly 10.025 seconds).

The default value for the loss probability of unicast messages, p_m, has been set according to experimental results on a prototypal implementation of the protocol, described in [6]. In particular, we are using results from the "cross-traffic" scenario, since interference is likely to occur, for example, in case the ALARP system is going to operate in an urban worksite (e.g., near a railway station in a large city).

Finally, concerning the probability that a single station will not receive a broadcast packet (p_b), measurements have been performed in [11] on the original RGCP protocol. Reusing such measurements is problematic mainly because: i) wireless technology has greatly improved in the last ten years and ii) such results refer to a controlled office environment and good physical conditions, which are by far not the typical ALARP conditions. Using the same approach as in [11] we plan to perform up-to-date measurements of p_b, which are however not available at this time. Obviously some correlation exists between parameters p_m and p_b, since both are influenced by network conditions, e.g., interference or obstacles that may exist between the sender node and the receiver. As a conservative setting, in the following evaluations we set p_b to the same value as p_m, i.e., a broadcast packet is simply considered as N individual unicast packets. This is a conservative approximation, since in reality the content of a broadcast message is transmitted using a single datagram.

6.2 Model validation with experimental data

In this section we compare the results obtained with the SAN model with results collected from the experimental runs conducted in [6]. In establishing an effective analytic framework, such experimental studies are an important aspect, since they allow to perform lower-layer measurements in order to assess the validity of the adopted assumptions. The scenario that is used for model validation is a simple scenario comprising two nodes and a dedicated access point, communicating with a prototype implementation of the TRC protocol. To keep the scenario as similar as possible to the experimental setup we have used the same parameters as in [6]: two nodes ($N=2$), an omission degree of 15, and a timeslot duration of 50 ms. The unicast message loss probability, p_m, is set as the experimentally measured packet loss rate.

The experimental test setup described in [6] allows to measure the following quantities: i) packet loss ratio, with packet denoting a single IP packet; ii) poll to request transmission time, describing the duration between the TRC coordinator sending a poll message and the coordinator receiving the corresponding request; iii) request to broadcast reception delay, characterizing the delay between the coordinator receiving a new broadcast request message for dissemination, and the same broadcast message being successfully received by all nodes; iv) request to broadcast completion delay, characterizing the delay between the coordinator receiving a new message for dissemination, and the instant in which it receives the last acknowledgement for that message.

Table 1. SAN Model Parameters and Description

Name	Description	Default
N	Number of nodes in the scenario (coordinator excluded)	20
t_{slot}	Duration of a timeslot	25 ms
OD	Omission degreee	10
res(c)	Resiliency degree of the message	10
p_m	Loss probabilitty of a single unicast packet	0.177
p_b	Probability that a station does not receive a broadcast (or multicast) packet	p_m

For data collection, two protocol execution scenarios are specified, one cross-traffic scenario and one non-cross-traffic scenario. The main coordinator is an off-the-shelf IEEE 802.11 Access Point (11Mbit/s bandwidth), running the TRC protocol. The scenario includes two additional nodes and another Access Point, contending the same channel. Channel contention is generated by constant transmission and saturated transmission buffers using a frame length of 1506 bytes. The remainder of this section discusses the experimental results with regard to their influence on TRC protocol configuration, and model parameterization.

6.2.1 Packet-loss ratio

Packet-loss ratio is estimated by counting the total number of executed protocol rounds and the number of successful poll-request packet sequences, thus providing an estimator p' of packet loss probability. This estimator is used to set the SAN model parameter p_m, thus allowing for a realistic setting regarding the probability of packet loss for unicast packets.

6.2.2 Poll to request transmission time

For studying the poll-request packet distribution within a single slot, the transmission times are captured using the coordinator system clock. The first timestamp is captured immediately before sending a poll packet; the second time-stamp is captured on receiving the request packet. Knowing this transmission characteristic is essential to appropriately configure the node slot parameter, t_{slot}. With a t_{slot} value set very high, bandwidth is allocated without being actually required for transmission, i.e., most poll-request-broadcast sequences finish in a fraction of the slot. This unnecessarily increases the overall protocol time bounds. Vice-versa, setting t_{slot} too low leads to an increase of necessary packet retransmissions, but having the benefit of shorter round times.

6.2.3 Request to broadcast reception delay, and request to broadcast completion delay

The completion delay characterizes the TRC protocol performance, having influence on selecting the appropriate OD and $res(c)$. From a safety perspective in the ALARP system, the re-

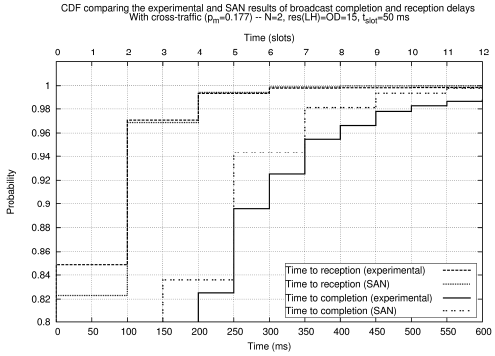

Figure 6. CDFs for validation with experimental data (experimental results taken from [6], for the case with cross-traffic.

ception delay is crucial for disseminating information to workers. From a throughput performance perspective, the completion delay is more relevant, since it determines when the sender node will be able to issue a new broadcast request.

While the first two experimental metrics can be used to set SAN model parameters, the other two metrics can be used to validate the stochastic SAN model against the experimental measurements. Fig. 6 provides a graphical comparison of the experimental CDF results and analytic SAN results, in the "cross-traffic" scenario. Both the metrics concerning broadcast completion and broadcast reception have been taken into account. It should be noted that there is a minor difference in the way the metrics are obtained; in the experimental setup the delay is measured from when the coordinator receives the broadcast, while in the SAN model it is measured from when the node has the message ready to be sent (i.e., the beginning of the timeslot). Moreover, experimental measurements have a higher precision with respect to the SAN model, which evaluates the selected measures with a resolution of protocol rounds.

Results in Fig. 6 show however a good correlation between SAN results and experimental results, on both metrics. The higher resolution of measurements is however observable on the time to broadcast completion measure. A slight shift towards a higher required reception and completion time is observable in the experiments, compared to the SAN. This stems from regular, short send interruptions in the communication equipment for outgoing packets, therefore occasionally "missing" the slot, and has been observed and described in [6] as well. This behavior is particular to the COTS under test, and of course neglected in the model.

Similar results are obtained also in the case without cross-traffic, thus confirming that the stochastic model developed in this paper provides a good analytical match for the considered scenario. Link-layer network properties depend on several factors, including distance between nodes, weather conditions, interference sources. The validation performed in this paper demonstrates the ability of our model to match real-world data. More thorough measurement campaign could allow to accurately parameterize the model under different network conditions.

6.3 Evaluations

In this section we perform further evaluations of the TRC protocol using the SAN model that has been described in Section 5.3. The model is solved using the numerical solver provided by the Möbius framework [4].

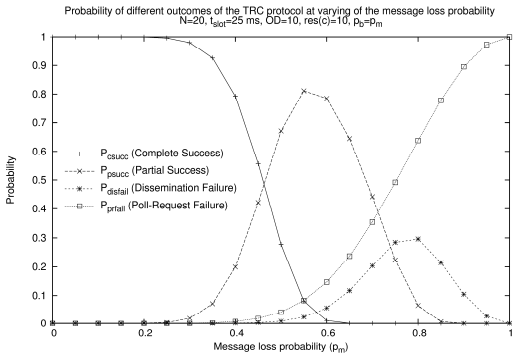

Figure 7. Probability of occurrence of the possible outcomes of the protocol, at varying of message loss probability.

6.3.1 Varying message loss probability

In Fig. 7 the relative probabilities of the four possible outcomes of message transmission are shown, at varying of the message loss probability p_m (and consequently p_b), with default parameters as listed in Table 1. Results shows that the probability of "Complete Success" drops rapidly to zero as pm increases above 0.4, while the probability of "Partial Success" starts to increase: due to the increase in message loss probability the coordinator is not able to receive the acknowledgment from all the nodes within the allowed retransmissions. For values of pm above 0.7, the probability of the "Partial Success" outcome starts to decrease as well. Consequently, the probability of protocol failure starts to increase when p_m raises above 0.4, and continues to increase until it reaches 1 when pm is equal to 1.

Results in Fig. 7 highlight two interesting points. First, the changes introduced by TRC protocol with respect to "RGCP with resiliency degree" (see Section 3.1) actually help in increasing the probability of success of the protocol in the ALARP environment. More in detail, since in "RGCP with resiliency degree" the receiving of all the acknowledgments is a mandatory condition for the delivery of the message to the applications, the "Partial Success" case would not have been possible. In TRC the "Partial Success" outcome provides a significant contribution to the success probability, especially when the message loss probability increases above 50%. The second point is that, as the message loss probability increases, the failure of the protocol is almost always caused by a failure of the poll-request message exchange. Failure to deliver the broadcast ("Dissemination Failure") occurs as well, but its occurrence probability does never exceed 0.3. Even if using a conservative setting, the most critical part of the protocol is the poll-request exchange. The probability of "Dis-

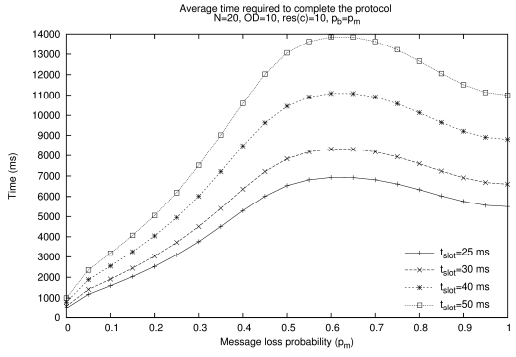

Figure 8. Average time required to complete the protocol, at varying of message loss probability, and timeslot duration.

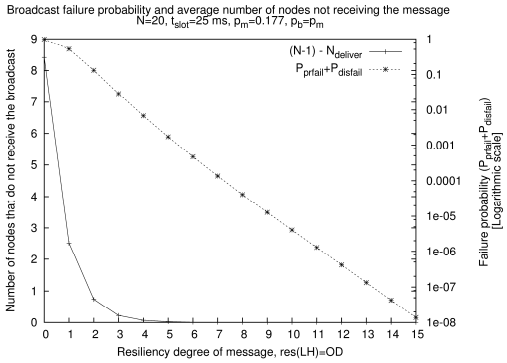

Figure 9. Failure probability, and number of nodes that do not receive the message, at varying of the resiliency degree.

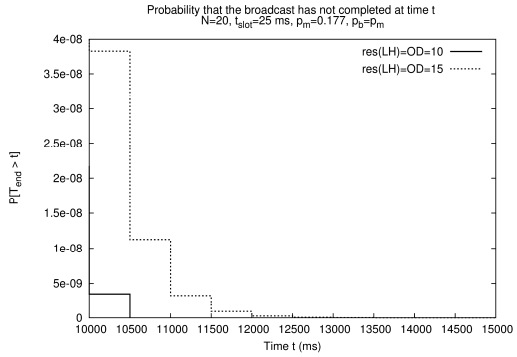

Figure 10. Probability that the broadcast has not completed at time t, at varying of the resiliency degree.

semination Failure" reaches its maximum when pm is around 0.8, then it starts to decrease because the increasing failures of the poll-request exchange do not even allow the protocol to reach the "broadcast" stage.

Fig. 8 shows how the mean time required to complete the protocol varies at varying of the message loss probability p_m, for different timeslot durations. Starting from about $p_m=0.5$, the mean time required to complete the broadcast decreases with the decrease of the message loss probability. The lower bound is reached when the message loss probability is zero, i.e., all the messages are correctly delivered. Under these conditions, the protocol always terminates successfully after the first round. With the default settings, comprising 20 nodes and timeslot duration of 25 ms, it takes exactly 500 ms to complete one round. Conversely, when the message loss probability is 1, the protocol always terminates after having failed the poll-request *OD+1* consecutive times, which yields exactly 220 slots, corresponding to 5500 ms. Results show that, on the average, the time required to complete the broadcast with default settings is below 7 seconds, which means that on the average there is enough room even to process additional emergency messages.

It is interesting to note that the greatest amount of time is needed not when pm approaches 1, but when it is around the value of 0.6. This behavior is due to the fact that when the message loss probability approaches 1 it is more likely that the poll-request message exchange will fail *OD+1* times, thus shortening the total time required for protocol execution, since the actual broadcast is not performed at all.

6.3.2 Varying the resiliency degree

In the following evaluations we hold the parameter p_m to the value obtained from the experimental setup in [6], (the default value in Table 1), and evaluate the protocol quantitative properties at varying of the *res(c)* parameter. The plot in Fig. 9 shows the impact of the resiliency degree of the message on two related measures of interest. The failure probability of the protocol (P_{prfail} + $P_{disfail}$) is plotted with respect to the y-axis on the right, in logarithmic scale. The other line depicts the number of nodes that do not have received the message when the protocol terminates (given by the quantity N-1-$N_{deliver}$) and refers to the left y-axis. As expected, both measures decrease as the resiliency degree parameter increases.

When using a resiliency degree of zero, meaning that no retransmissions are allowed, on the average more than 8 out of 19 nodes will not receive the broadcast. By increasing *res(c)* this quantity drops very rapidly to zero: by just increasing *res(c)* to 2 only one node, on the average, is not able to receive the message. The failure probability drops very rapidly as well: an increase of two points in resiliency degree yields a decrease of failure probability of about one order of magnitude. For example, in the default scenario (*OD=10*) the failure probability is between 10^{-5} and 10^{-6}; by increasing *res(c)* to 12 the failure probability drops below 10^{-6}.

Results in Fig. 9 show that the failure probability of the TRC protocol can be reduced, by simply increasing the omission degree parameter, and consequently the resiliency degree of the transmitted message. This increase in success probability, however, comes at a price, which corresponds to an increase in protocol duration. Fig. 10 shows the probability that the protocol has not completed at a given instant of time, which is evaluated as *1–F(t)*, where *F(t)* is the cumulative distribution function of the time at which the protocol terminates. To emphasize the connection with ALARP requirements, only the portion after 10 seconds is shown. The figure clearly shows that increasing the resiliency degree from 10 to 15 increases such probability of almost an order of magnitude.

These kinds of tradeoffs are of greatest importance in the analysis of the ALARP system and results show that their careful analysis is mandatory, given the limited amount of time in which communication should take place. The model described in this paper has been proven useful for this kind of analysis, and will provide a valuable resource in the final assessment of the ALARP system.

7. CONCLUSIONS AND FUTURE WORK

In this paper we have performed a model-based analysis of the Timed Reliable Communication (TRC) protocol, a broadcast protocol based on the IEEE 802.11 wireless technology that is being used within the EU funded ALARP project for railway worksite communication. The model results have been compared with experimental measurements collected within the project in a basic scenario, and good correspondence has been observed. A set of key metrics have then been defined and evaluated, assessing that the protocol allows to satisfy the ALARP targeted performance and reliability requirements. Finally, tradeoffs in choosing protocol parameters have been observed and discussed. Future work consists in performing accurate measurements in real worksite setups, to the extent to which it is possible, and

analyzing the protocol behavior with parameters taken from such real application scenario.

8. ACKNOWLEDGMENTS

This work has been partially supported by the European Project FP7-IST-234088 ALARP [1] and by the Italian Ministry for Education, University, and Research (MIUR) in the framework of the Project of National Research Interest (PRIN) "DOTS-LCCI": Dependable Off-The-Shelf based middleware systems for Large-scale Complex Critical Infrastructures.

9. REFERENCES

[1] ALARP – A railway automatic track warning system based on distributed personal mobile terminals – FP7-IST-2010-234088 http://www.alarp.eu/.

[2] ALARP D1.2, "Requirements Specifications", v0.5, Project FP7-IST-2010-234088 ALARP, June 2010.

[3] A. Bondavalli, A. Coccoli, and F. D. Giandomenico. "QoS Analysis of Group Communication Protocols in Wireless Environment". In Ezhilchelvan, P., Romanovsky A. (eds), Concurrency in Dependable Computing. Kluwer Academic Publishers. pp. 169-188. 2002.

[4] G. Clark, T. Courtney, D. Daly, D. Deavours, S. Derisavi, J. M. Doyle, W. H. Sanders, and P. Webster. "The Möbius Modeling Tool", Proc. of the 9th International Workshop on Petri Nets and Performance Models, Aachen, Germany, September 11-14, pp. 241-250, 2001.

[5] A. Coccoli, P. Urban, A. Bondavalli, and A. Schiper, "Performance analysis of a consensus algorithm combining Stochastic Activity Networks and Measurements", Proc. International Conference on Dependable Systems and Networks (DSN'02). pp. 551-560, 2002

[6] B. Malinowsky, J. Grønbæk, H.P. Schwefel, A. Ceccarelli, A. Bondavalli, E. Nett, "Realization of Timed Broadcast via Off-the-Shelf WLAN DCF Technology for Safety-critical Systems", Proc. 9th European Dependable Computing Conference (EDCC'12), Sibiu, Romania, May 8-11, 2012.

[7] J. F. Meyer, A. Movaghar, and W. H. Sanders, "Stochastic activity networks: structure, behaviour and applications", Proc. International Workshop on Timed Petri Nets. IEEE, New York, pp. 106–115, 1985.

[8] M. Mock, E. Nett, and S. Schemmer, "Efficient reliable real-time group communication for wireless local area networks", Proc. 3rd European Dependable Computing Conference (EDCC'99), pp. 380-400, 1999.

[9] E. Nett and S. Schemmer, "Reliable real-time communication in cooperative mobile applications", IEEE Transactions on Computers, vol. 52, no. 2, pp. 166–180, 2003.

[10] W. H. Sanders, and J. F. Meyer, "Stochastic activity networks: formal definitions and concepts", Lectures on formal methods and performance analysis, Springer-Verlag, 315-343, 2002.

[11] A. Coccoli, S. Schemmer, F. Di Giandomenico, M. Mock, and A. Bondavalli, "Analysis of Group Communication Protocols to Assess Quality of Service Properties" 5th IEEE High Assurance System Engineering Symposium (HASE00), 247-256, 2000.

[12] ALARP D2.2, "Preliminary wireless communication solution", v0.7, July 2011.

[13] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," Selected Areas in Communications, IEEE Journal on, vol.18, no.3, pp. 535–547, 2000.

[14] "IEEE Standard for Local and Metropolitan Area Networks, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications" IEEE Std 802.11-2007, June 12, 2007.

[15] J.B. Andersen, T.S. Rappaport, and S. Yoshida, "Propagation measurements and models for wireless communications channels" IEEE Communications Magazine, vol.33, no.1, pp.42-49, Jan 1995.

[16] K. Hiltunen and R. de Bernardi, "WCDMA downlink capacity estimation", IEEE 51st Vehicular Technology Conference Proceedings (VTC 2000), pp. 992-996, Tokyo, 2000.

[17] R. Owen, P. Jones, S. Dehgan, and D. Lister, "Uplink WCDMA capacity and range as a function of inter-to-intra cell interference: theory and practice" IEEE 51st Vehicular Technology Conference Proceedings (VTC 2000), pp. 298-302, Tokyo, 2000.

[18] M. Zorzi and R.R. Rao, "On channel modeling for delay analysis of packet communications over wireless links", 36th Annual Allerton Conference on Communications, Control and Computing, Allerton House, Monticello, IL, Sept. 23-25, 1998.

[19] M.I. Hassan, H.L. Vu, and T. Sakurai, "Performance Analysis of the IEEE 802.11 MAC Protocol for DSRC Safety Applications" IEEE Transactions on Vehicular Technology, vol.60, no.8, pp. 3882-3896, Oct. 2011.

[20] C. Tang and P.K. McKinley, "Modeling Multicast Packet Losses in Wireless LANs", In Proc. of the 6th ACM international workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM'03), San Diego, CA, USA, September 2003.

[21] J.C.P. Wang, M. Abolhasan, D.R. Franklin, F. Safaei, "Characterising the Behaviour of IEEE 802.11 Broadcast Transmissions in Ad Hoc Wireless LANs", IEEE International Conference on Communications, (ICC '09). pp. 1-5, June 14-18, 2009.

[22] H. Cheng, C. Rong, and G. Yang, "Design and Analysis of a Secure Routing Protocol Algorithm for Wireless Sensor Networks" IEEE International Conference on Advanced Information Networking and Applications (AINA'11), pp. 475-480, March 22-25, 2011.

[23] M. Massink, D. Latella, and J.-P. Katoen, "Model checking dependability attributes of wireless group communication" International Conference on Dependable Systems and Networks (DSN'04) pp. 711-720, June 28th – July 1st, 2004.

[24] "IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Broadband Wireless Access Systems", IEEE Std 802.16m-2011, May 5, 2011.

[25] "IEEE Standard for Local and Metropolitan Area Networks, Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs)", IEEE Std 802.15.1-2005, 2005