Fault Prediction in an Ad-hoc Network

Everaldo Leme, Danilo Nogarotto, Paulo S. Martins, R. L. O. Moraes
University of Campinas - UNICAMP
School of Technology
Limeira-SP, Brasil

Abstract—A limitation that impacts the dependability of communication systems is its delivery rate, which can be affected by communication faults. Predictive models play an important role in furnishing the necessary knowledge to develop mechanisms to improve system dependability. This work is an evolution from a previous one with the goal of improving and validating a one-inflated beta regression model. The model is used to predict the behavior of an ad-hoc network in the presence of packet loss faults. In previous work, a set of experiments was performed using a 5-node ad-hoc network under an epidemic protocol and a first model was proposed. In this work, complementary experiments were performed based on an extended 5 to 8-node ad-hoc network to analyze the model when different explanatory variables are used. A second model was also introduced to allow the comparison analysis. The results confirmed the influence of the percentage of faults as a determinant variable to explain the delivery rate of the network and they also indicated the suitability of the beta regression model.

Index Terms—Epidemic Protocol, Ad-Hoc Networks, Fault Injection, Fault Prediction.

I. Introduction

Mobile Ad-hoc Wireless Networks (MANETs) are frequently used to emulate a cloud of mobile devices such as laptops, phones, tablets and household appliances, as shown by Niroshinie et al. [1]. By using this private network the users share resources and collaborate on common activities among them. The importance of D2D (Device-To-Device) environments is more evident when the users are exposed to environmental and mass-casualty disasters where com-munication is seldom available or where the activities they need to perform exceed the computational and communication capacity available in a single device.

Particularly in environments exposed to severe conditions, where the proper operation of the network (its dependability) is mostly needed, the network infrastructure is prone to failures and can be exposed to the failure of stations (server and clients), network partitioning, packet loss and delay, as well as arbitrary behavior of system components.

Cristian [2] presented traditional fault models for distributed systems and mainly included crash faults (loss of services or internal state), omission (no replies for some inputs), timing (early or delayed responses), incorrect responses (for some entries) as well as arbitrary (unpredictable behavior) faults. Schneider [3] extended this model by adding fail-stop and by dividing omission into send and receive faults.

To understand network behavior under the presence of faults as well as to validate its dependability, fault injection techniques are broadly used. To apply fault injection it is necessary to know the workload profile and characteristic faults (i.e., faults that commonly occur in the System Under Test - SUT) [4]. Fault injection techniques accelerate the activation of faults that are deliberately inserted and facilitate the understanding of the SUT's behavior under their presence [4].

In this work, the Fault Injection Relocatable Module for Advanced Manipulation and Evaluation of Network Transport - FIRMAMENT - is used to emulate communication faults. FIRMAMENT was implemented as a core module of the Linux operation system [5] and it was previously applied by Weber et al. [6].

The protocol used in the experiments was the Network-friendly Epidemic Multicast - NEEM protocol, which aims at ensuring no network congestion during overload periods [7]. Normally, a fault injection tool is necessary to support the dependability validation through fault injection techniques. Clearly, an epidemic protocol is quite useful in this environment. Techniques based on Gossip ensure epidemic dissemination since they offer scalability and robustness. An interruption may slow down the delivery of data, but it can not prevent it from reaching those nodes that remain active and connected to the network [8]. Although the epidemic protocol is efficient, communication faults may compromise the delivery of messages in the ad-hoc network environment and the prediction of their occurrence is relevant to the proactive management of network services.

Prediction can be used in several areas and it is supported by both mathematical and statistical models. Regression models are widely used to identify explanatory variables that are related to a response variable, as well as to make their prediction. To model ratios, i.e. when the response variable lies in a continuous interval (0,1), one may apply the beta regression model [9] or the inflated beta regression model in zero or one [10] when the endpoints of the interval (0,1) are present in the data.

In a previous work, we proposed one-inflated beta regression to model the occurrence of communication faults in a 5-node experimental ad-hoc network [11]. In this work, we 1) corroborate the suitability of the beta model using different explanatory variables; 2) validate the model by using additional and complementary experiments that were performed based on an extended 5 to 8-node ad-hoc network, and 3) introduce a linear regression model to allow the comparison of results and the analysis of the suitability of both models.

In addition to confirming the suitability of the one-inflated

beta regression model for a larger network (8-node ad-hoc network) in the laboratory, the results in this paper show the behavior of redundant messages under the epidemic protocol and reinforce the importance of the percentage of faults to determine the delivery rate of the network. The network nodes become unavailable when faults were injected in a percentage of around 60%.

This paper is organized around the following sections: In Section II we cover the background and related work, Section III addresses the prediction model and approach employed; The results and discussions are presented in Section IV, and the conclusions and future work are shown in Section V.

II. Background and Related Work

Dependability may be regarded as the capacity of a system for delivering services that can be trusted [4]. It is measured as its ability of avoiding service faults that are more frequent and severe than acceptable. Faults may decrease the dependability of a software product, since they hinder or prevent one or more of its attributes, which are: availability (readiness for the correct service); reliability (continuity of correct service); safety (absence of catastrophic consequences on the user and on the environment); integrity (absence of inadequate alter-ations in the system), and maintainability (ability to undergo modifications and repairs).

According to the IEEE standard [12], an ad-hoc network is composed only by stations within mutual communication range through the wireless medium. An ad-hoc network is typically created spontaneously. The main characteristic of an ad-hoc network is its limited temporal and spatial extent. These limitations allow the simple creation and dissolution of the network by non-technical users. Ad-hoc networks are self-configuring and offer a fast, easy and inexpensive multi-hop wireless communication. Dependability in ad-hoc networks should be focused on availability and reliability issues [13].

An approach to multicast communication is Gossip, as shown by Pao et al. [14]. Each receiver in a multicast group periodically exchanges state information with some other members of the group. Lost messages can be retrieved by other nodes in the network. This type of protocol distributes the load among the nodes of the group and it is resistant to failures. The Gossip-based approach provides reliable message delivery and the overhead can be alleviated by adjusting protocol parameters.

Communication through epidemic protocols, due to redundancy, ensures high probability that all nodes involved in the communication receive all messages. They are potentially suitable for a mobile environment where it is common to loose the connection due to the network range and signal quality. On the other hand, as the messages are typically resent several times, protocols based on epidemic communication may curtail to a greater or lesser extent the network performance [15], by making unnecessary use of network bandwidth, which is clearly a scarce resource in a mobile and cloud environment. Thus, scalability and resilience is achieved at the expenses of increased message traffic, which further affects the perfor-mance.

These gossip-based protocols typically include several configurable parameters, e.g. 1) Fanout, which is the maximum number of nodes to which a message must be sent; 2) Hops, which refers to the maximum allowable number of times a given message is retransmitted, usually initialized with zero. This value is increased each time a message is retransmitted and the nodes retransmit the message if its current hop value is smaller than the Maximum Hops, and 3) Gossip strategy (e.g., eager push approach or pull approach), which indicates the way to propagate a message in a gossip multicast communication. Strategies such as Push gossip, Pull gossip, Lazy Push gossip, Hybrid gossip, Structured gossip and Dynamic gossip are shown by Leitão [16].

Oliveira et al. [17] describe distributed fault scenarios in several parts of the network with a high level of abstraction. They present an approach for the execution of tests aimed at assessing dependability in distributed systems, considering: (i) developing a representation for describing distributed fault scenarios; (ii) development of a distributed fault injector, focusing on communication faults with emphasis on network partitioning; (iii) development of a global coordinator for synchronization, monitoring and collection of environment data. The fault injector was designed as a core module of the Linux operating system. It intercepts and suppresses messages in order to emulate a link crash.

III. Prediction Modeling Approach

The experiments carried out in this work are based on a IEEE 802.11 WLAN operating in ad-hoc mode and composed from 5 to 8 nodes arranged on a fully-connected topology (where all nodes are neighbors). These nodes are placed in the laboratory, i.e. there is a relatively short-distance among them. In addition to the network, the complete experimental environment contains three main elements: the fault injector, the workload and the epidemic communication protocol.

The NetEpidemic application was developed from the work of Wilges [19], which is a distributed presence-service work-load where each node informs the other nodes in the network the details about their current state (i.e. activities) through the NEEM epidemic protocol [7]. Each node sends a message containing its own identifier every 30 seconds.

The application was run for a total of 15 minutes. The nodes that receive this periodic identifier message update their own list of active nodes. Each node has a parameter which defines the time after which, if the identifier message is not received, allows its sender node to be deemed inactive (in this case, 8 seconds).

All events for each node are logged into a file and exported to a database for further analysis. To perform the epidemic communication, some parameters must be set for the proto-col. The four parameters that direct and mostly impact the epidemic communication are show in in Table I.

The Fanout parameter identifies the number of neighbors that a node must replicate an incoming message. The Hop

TABLE I
NEEM Protocol Parameters

Parameter	Value
Fanout	2
Hop	6
Neighbor	4
Gossip strategy	Discard messages from Buffer

parameter identifies the number of times ("jitter") that a message will be replicated in the environment. Each node receives the message body, the message identifier and the number of hops of the specific message. Then, the number of hops will be incremented by 1 and sent to other nodes. The message is no longer replicated when the number of hop reaches this threshold. The Neighbor parameter indicates the maximum number of neighbors that a node can have while it is active on the network. In this case the disposing message strategy was used from the buffer of each node after a certain time, decreasing the overload of receiving and sending messages.

FIRMAMENT, developed by Drebes [5], was used to inject artificial faults. FIRMAMENT is a fault injection tool with the goal of enabling test engineering to specify communication faults in complex scenarios. In this work the faultload is represented by omission faults of received messages.

The scenarios created for fault injection allow us to set a percentage of message loss in a node. Faults were injected considering message losses from 50 to 95% with a 5% increment on each network node. In addition to that, we also applied a golden run (with no faults). Fig. 1 shows the scenarios used in the experiments.

The metric chosen for the performance evaluation of the epidemic communication was the delivery rate, i.e. the number of messages that were sent and successfully delivered to each network node in relation to the total number of messages sent.

In an earlier work by Leme et al. [11], the authors addressed the prediction of communication faults using 5 nodes in a fully-connected ad-hoc network topology. That work revealed - through the predictive model of beta regression inflated in one - that it is possible to predict the communication faults in ad-hoc networks using the explanatory variables percentage of injected faults and percentage of nodes under faults, having as dependent variable the percentage of possible incoming messages.

In this work, the number of network nodes was increased up to 8 nodes, considering experiments in the range of 50-62% of network nodes under fault, i.e., we applied the following combinations of total number of nodes vs. number of nodes under faults: (5,3)(6,3)(7,4)(8,5). For example, for 6 nodes in network we considered 3 nodes under faults. The fault injection was performed exclusively by the FIRMAMENT tool following a Bernoulli process on the nodes under fault.

The classical Beta regression inflated-in-one model applied in our work is defined by Eq. (1):

$$f_1(y_t;\ _t;\ _t;\ ) = \begin{cases} (1)f(y_t;\ _t;\ ) & \text{if } y_t \ 2\ (0;\ 1) \\ _t & \text{if } y_t = 1 \end{cases} \quad (1)$$

where $f(y_t;\ _t;\ ) =$

$$\frac{(\ )}{(\ _t)\ ((1\ \ _t))}\ y_t^{\ 1}\ \ (1\ \ y\ )^{(1)1}_t \qquad _t \quad (2)$$

is the probability density function of the beta distribution, with mean $_t$. Furthermore, in order to include in the model the regression structure by means of the explanatory variables, we used the logit linking function given by expression (3) and (4).

$$X_{i=0}^{p}\ x_{ti\ i} = \log\left(\frac{_t}{1\ \ _t}\right) \quad (3)$$

$$X_{i=0}^{p}\ x_{ti\ i} = \log\left(\frac{_t}{1\ \ _t}\right) \quad (4)$$

The one-inflated beta regression model was employed to predict the delivery rate of messages y since the variables under the beta distribution are always within the interval (0,1), as it is the case of the dependent variable y. Specifically, the one-inflated model was selected since there are cases where all messages are received (i.e. y = 100% = 1). Thus, data is inflated in one, i.e. the dependent variable y has its limit within the interval (0,1]. By adopting the logit function, we reached an important interpretation, i.e. the odds ratio (OR). This is why the link function is one of the mostly adopted functions.

In order to compare the one inflated beta regression model, we also fit the normal regression model as described in [20]. In this case, we assume that the response variable has a normal distribution (and not a beta distribution such as represented by Equation (1)).

IV. Results and Discussion

Having collected and analyzed the data from the experiments we observed that, beyond the rate of 60% of fault injection, the delivery rate started a slow decline for all scenarios, even when we increased the number of nodes in the network[1]. This behavior is very similar to the one observed in the 5-node network in previous work [11]. Fig. 2 and Table II show the metric delivery rate - i.e. percentage of received messages in all the scenarios evaluated. It presents the percentage of injected faults versus the number of nodes in the network when 50 to 62 % of the nodes are under faults.

An interesting behavior was observed in the experiments using 7 and 8-nodes, which had not occurred previously. Redundant messages were registered in the application, i.e. the same ID (identification) message was received more than once. This actually occurs due to the epidemic protocol, as the NEEM protocol is based on Multicast-Gossip. In this protocol, neighbors send the same message more than once, and the fanout and hop parameters affect the spread of messages

[1]for the fully-connected topology

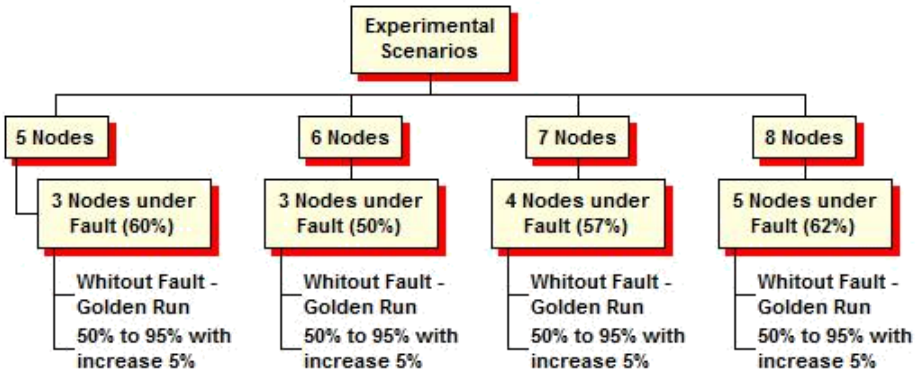

Fig. 1. Experimental scenarios.

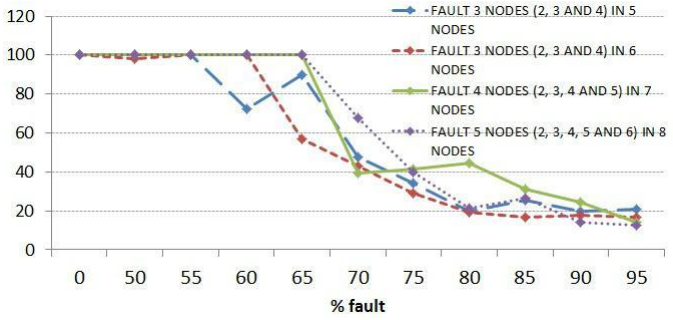

Fig. 2. **%** delivered (received) messages in distinct experimental scenarios.

TABLE II
% Received Messages in experimental scenarios

% Fault	5 Nodes	6 Nodes	7 Nodes	8 Nodes
0	100.00	100.00	100.00	100.00
50	100.00	98.33	100.00	100.00
55	100.00	97.22	100.00	100.00
60	72.41	100.00	97.14	100.00
65	89.65	56.66	94.76	96.25
70	47.58	42.77	39.52	67.50
75	34.48	28.88	41.42	40.00
80	20.00	19.44	44.76	21.25
85	25.51	16.66	31.42	26.66
90	20.00	17.77	24.28	14.16
95	21.37	16.66	14.28	12.50

through various paths, generating redundancy and a possible overload in the network. Even if the NEEM protocol aims at ensuring good buffer management through its internal mechanisms, a residual redundancy in the message delivery remains. This redundancy was not evaluated in this work, i.e. the redundant messages received were disregarded in the predictive model without sacrificing the quality of results.

Based on these data, we applied the one-inflated beta regression model. The percentage of injected faults and the total number of nodes in the network were considered the explanatory variables.

The estimated intercept is the value of the logit function when the value of the explanatory variables (i.e. percentage of injected faults and the total number of nodes in the network) is zero. We have two intercepts: α_0 for the logit function within (0,1) and γ_0 for the logit function within the inflated part.

TABLE III
Fitting Model 1

coefficients	Estimate	Std deviation	p-value
α_0 Intercept	6.3448	0.8284	<0.001
α_1 % of injected faults	-8.8246	0.9766	<0.001
α_2 Total nodes in network	0.0047	0.0908	0.9590

coefficients	Estimate	Std deviation	p-value
γ_0 Intercept	12.1565	6.1442	0.0553
γ_1 % of injected faults	-33.5943	12.7076	0.0120
γ_2 Total nodes in network	1.3577	0.7645	0.0840

For each parameter of the regression model, a p-value is calculated which corresponds to the test that checks whether or not the parameter has an effect on the dependent variable. A small p-value shows that the variable is relevant and it does explain a change in the dependent variable. On the contrary, a large p-value indicates that the variable has not an effect on the dependent variable.

We have fit two models. The first adjusted model is shown in Table III. The parameters (Average beta distribution), and (the probability to receive all messages) were modeled by the explanatory variables.

Para uma variável ser significativa a 5%, ela deve ter um p-valor menor que 5%.

For this first model, the total number of nodes in the network with coefficients α_2 and γ_2 is not significant, because the p-value is larger than 5% (i.e. this variable does not explain the percentage of received messages). For a variable to have a significance level of 5%, it must have a p-value below 5%.

A second model was then adjusted disregarding the explanatory variable of total number of nodes in the network (Table IV). The adjusted model only has the percentage of injected faults as the explanatory variable (coefficients α_1 and γ_1). They are significant to explain the percentage of received message (the mean) and the probability of receiving all

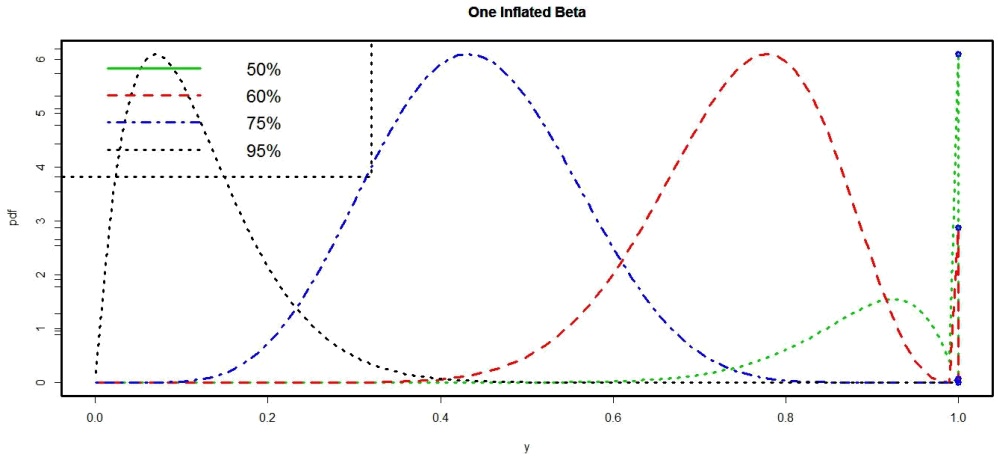

Fig. 3. Probability fit of received messages under 50, 60, 75 and 95**%** of injected faults.

TABLE IV
Fitting Model 2

coefficients	Estimate	Std deviation	p-value
β_0 Intercept	6.3653	0.7279	<0.001
β_1 % of injected faults	-8.8125	0.9478	<0.001

coefficients	Estimate	Std deviation	p-value
β_0 Intercept	15.8220	5.4450	0.0060
β_1 % of injected faults	-25.3490	8.6190	0.0055

messages ( ), respectively. This model was chosen to interpret the prediction results.

By analyzing the coefficient, it can be observed that an increase in the injected fault rate of 5% decreases the chances of receiving a message by approximately 35.6%. The graph in Fig. 3 shows the fault injection simulation considering 50, 60, 75 and 95% of injected faults.

Table V shows the delivery rate metric obtained experimentally as well as predicted by the model. For example, for 50% of injected faults (i.e. first table entry), the statistic model indicated a range of 98-100% of percentage delivery rate, whereas our experiments showed a corresponding values of 100, 98.33, 100, 100 % for 5, 6, 7 and 8 nodes respectively. Therefore, it is possible to confirm that the model represents the values that were found in the experiments.

TABLE V
Comparison analysis: Statistic Model vs Laboratory Results

% Fault	Statistic Model	5 Nodes	6 Nodes	7 Nodes	8 Nodes
50	98.00- 100.00	100.00	98.33	100.00	100.00
60	60.00 - 90.00	72.41	100.00	97.14	100.00
75	25.00 - 60.00	34.48	28.88	41.42	40.00
95	00.00 - 25.00	21.37	16.66	14.28	12.50

The analysis of the coefficient indicates that an increase of 5% in the injected fault rate decreases the chances of receiving all messages by 71.8%.

Therefore, regardless the number of network nodes, if the range of 50-62% of the nodes are under fault, the percentage of received messages will be influenced by the percentage of injected faults, but not by the number of nodes under fault, confirming the proportionality in fully-connected ad-hoc networks operating on epidemic communication. This is true if one considers the epidemic parameters and the equipment used to ensure this proportionality.

TABLE VI
FITTING THE NORMAL MODEL

coefficients	Estimate	Std deviation	p-value
β_0 Intercept	1.3231	0.0979	<0.001
β_1 % of injected faults	-1.1091	0.1390	<0.001

In a second step, we fit the normal linear regression model. In this case, we have not fitted the inflated part, i.e. we have not fitted the probability of receiving all messages, which is one disadvantage of this model. As with the one inflated beta model, the best fit considered only the percentage of injected faults variable (Table VI). The number of nodes was not significant in this model as well. Thus, in Table VI we have only the fitting of this model, similar to the model adjusted by the one inflated beta regression model as shown in Table IV.

The AIC (Akaike Information Criterion) and the BIC (Bayesian Information Criterion) [20] were used to compare the one inflated beta regression model and the normal linear regression model (Table VI). These results are shown in Table VII. Smaller AIC and BIC values are better when comparing two or more models.

Note that the results for the beta regression model were better in comparison to the normal model. Both AIC and BIC values were smaller. Therefore, the one inflated beta regression model is an interesting choice for fitting this type of data. This model is more adequate since, unlike the normal model,

TABLE VII
AIC E BIC CRITERION FOR MODELS

	Normal model	Beta Inflated model
AIC	-0.5292	-21.5098
BIC	4.8234	-12.5888

it takes into consideration that the response variable is within the interval (0,1].

V. CONCLUSION

This paper presented an experimental analysis in a wireless ad-hoc network operating in a fully-connected-topology mode. An application (NetEpidemic) executing under an epidemic protocol (NEEM) was developed to generate the workload. Communication faults, i.e. omission faults of received packets, were injected through a fault injection tool organized around several scenarios. In essence, we may summarize and conclude as follows:

Protocol efficiency: the epidemic protocols are efficient, due to their relative high performance in spreading a message between network nodes, even in the presence of faults. Our experiments were able to quantify this efficiency by showing that the decrease in the delivery of messages starts from a rate of 60% of message loss - independently of the number of nodes in the network. This percentage value may be regarded as a guideline for designers aiming at ad-hoc networks that are more resilient to faults. The experiments have not identified buffer overflow and redundant messages, considered not relevant to this application, as the NEEM protocol manages these requirements.

Predictive model and validation: the adjusted predictive model was able to show the same behavior as the one exhibited by the real environment, and it was also capable of predicting the percentage of received messages in this environment for a larger number of nodes in the network.

The impact of the number of nodes: An important conclusion of this work was that the response variable for total or partially delivered messages is explained by the percentage of injected faults and not by the number of network nodes, for the environment under consideration. Therefore, whereas the number of nodes (i.e. network size) may continue to be a parameter under consideration, we argue that the analysis presented lessens its relevance or priority in subsequent research. To some extent, the re-sults presented in this paper allowed a generalization (by statistical means) of the results presented in our previous work, in regards to the variable “network size”, without the need of performing a costly scalability analysis on a real network platform.

One beta inflated regression model: When comparing a one beta inflated regression with a normal regression model (most commonly used), we realize the advantages of the former. Both AIC and BIC information criteria were better for the beta model. Furthermore, for the response variable “percentage of received messages” con-sidered, the beta distribution was the most adequate in this case.

In future work, we intend to build a discrete event simulation model that is validated by the experimental model we introduced (and vice-versa), and then use the simulation model to analyze in more detail the impact of epidemic communication variables such as fanout, hop and gossip strategy (among others) on the message delivery rate, when the system is subject to diverse fault-injection scenarios.

REFERENCES

[1] F. Niroshinie, W. Seng, R. Wenny, Mobile Cloud Computing: A Survey. ELSEVIER. Future Generation Computer Systems, Vol. 29, pp. 84–106, 2013.

[2] F. Cristian, Understanding Fault-Tolerant Distributed Systems. Com-munications of the ACM, Vol. 34(2), pp. 56–78, 1991.

[3] F. B. Schneider, What Good are Models and What Models are Good? Mullender, S., editor, Distributed Systems, Addison-Wesley, Workingham, 2nd edition, pp. 17–26, 1993.

[4] J. Arlat, A. Costes, Y. Crouzet, J. C. Laprie and D. Powel Fault Injection and Dependability Evaluation of Fault Tolerant Systems. IEEE Transaction on Software Engineering, 16, 166-174, 1993.

[5] R. Drebes, Firmament: A Communication Fault Injection Module for Linux (in portuguese) MSc in Computer Science, UFRGS, 2005.

[6] T. Weber, R. Drebes, G. Jacqyes-Silva, J. Trindade, A Kernel-based Communication Fault Injector for Dependability Testing of Distributed Systems. Proc. of the First Haifa Intl. Conf. on Hardware and Software Verification and Testing. Springer-Verlag, pp. 177-190, 2006.

[7] J. Pereira, L. Rodrigues, M.J. Monteiro, R. Oliveira, Neem: Network-Friendly Epidemic Multicast., Reliable Distributed Systems. Proceed-ings. 22nd Intl. Symp. on, 2003

[8] P.T. Eugster R. Guerraoui, A. M. Kermarrec, M. Massoulié, Epidemic Information Dissemination in Distributed Systems., IEEE Computer Society, 2004.

[9] S. L P. Ferrari and F. Cribari-Neto Beta regression for modelling rates and proportions. Journal of Applied Statistics 31, 799-815, 2004.

[10] R. Ospina and S. L. P. Ferrari A general class of zero-or-one inflated beta regression models. Computational Statistics and Data Analysis 56, 1609-1623, 2012

[11] E. Leme, D. Nogarotto, R. Moraes, P. S. Martins, Analyzing the behavior of communication faults in ad-hoc networks. Computing and Comm. (IEMCON), 2015 Intl. Conf. and Workshop on, Vancouver, BC, 2015.

[12] IEEE 802.11, 1999 Ed. (ISO/IEC 8802-11: 1999) A general class of zero-or-one inflated beta regression models. IEEE Standards for Information Technology, Wireless LAN Medium Access Control (MAC) And Physical Layer (PHY) Specifications, 1999.

[13] C. Basile, M. O. Killijian, D. Powell, A Survey of Dependability Issues in Mobile Wireless Networks. IEEE Technical Report, LAAS CNRS, Toulouse, France, 2003.

[14] D. Pao, K. S. Lau, Tree-Based versus Gossip-Based Reliable Multicast in Wireless Ad Hoc Networks. IEEE Communications Society subject matter experts for publication in the IEEE CCNC 2006 proceedings, 2006.

[15] N. Carvalho, J. Pereira, R. Oliveira, L. Rodrigues, Emergent Structure in Unstructured Epidemic Multicast. IEEE/IFIP Int. Conference on Dependable Systems and Networks, (DSN'07), pp. 481-490, 2007.

[16] J. A. Leitão, Gossip-Based Broadcast Protocols. MSc Dissertation, Universidade de Lisboa, Lisboa, Portugal, 2007.

[17] G. M. Oliveira, S. L. Cechin, T. Weber, Distributed Communication Fault Injection with the support of Control and Coordination of Exper-iments (in Portuguese). Workshop on Testing and Fault Tolerance - WTF, João Pessoa. pp. 101-114, 2009.

[18] D. Andrés, J. Friginal, J. C. Ruiz, P. Gil, Attack Injection to Support the Evaluation of Ad Hoc Networks. 29th IEEE Int. Symp. on Reliable Distributed Systems, 2010.

[19] P. Wilges, A Distributed Presence Service over Epidemic Multicast., Vol.2 pp 50-59. Journal of Applied Computing Research, 2012.

[20] S. Weisberg, Applied Linear Regression. John Wiley & Sons, Inc., 3rd edition, pp. 310, 2005.