Improving the Dependability of Mobile Ad-hoc Networks through Formal Reasoning

Martín López-Nores*, David Pereira-Paz†, José J. Pazos-Arias*, Jorge García-Duque*
and Esther Casquero-Villacorta*
*Department of Telematics Engineering, University of Vigo
ETSE Telecomunicación, Campus Lagoas-Marcosende s/n, 36310 Vigo (Spain)
Email: {mlnores,jose,jgd,esther}@det.uvigo.es
†DMR Consulting
Avda. Diagonal, 605 4ª planta, 08028 Barcelona (Spain)
Email: david.pereira.paz@dmr-consulting.com

***Abstract*— This paper describes ongoing work aimed at improving the dependability of mobile ad-hoc networks in terms of service provision. Building upon the idea of *knowledge dissemination and exploitation*, we motivate the use of formalisms intended to reason about uncertainty and inconsistencies, and then outline an integrated solution that borrows mechanisms from the field of specifying distributed real-time systems.**

I. Introduction

The past few years have witnessed a shift from traditional desktop machines reliant on fixed wired networks to ubiquitous, wireless-communication-enabled mobile devices. Mobile ad-hoc networks (MANETs) represent a new environment with a number of increasingly relevant real-world applications, from sensor networks to peer-to-peer wireless computing.

MANETs undergo frequent changes in network topology. Small devices arise opportunistically and communicate with no reliance on any form of fixed infrastructure, and their physical mobility results in unpredictable connectivity. In turn, this volatility leads to limited dependability at the level of service provision, as the applications may suffer disconnections at any time. To face this problem, it was argued in [1] that it is necessary to augment the predictability of the networks through *knowledge dissemination and exploitation.* The idea is to have the hosts expose and gather information that allows them to guess how the network is set up at a given moment and how it will be in the near future. Thus, it would be possible for the hosts to detect whether the *service requirements* [1] are likely to be satisfied, and react conveniently in case not (moving to specific locations, accomplishing service migrations, etc.).

The commented approach faces two fundamental problems:

1) **Uncertainty:** it is unrealistic to assume that a host may have complete knowledge about the MANET, because every host gathers information in a progressive way, from an initial situation in which it knows nothing about others. Moreover, it frequently happens that a host cannot expose complete information about itself —for example, it may not be able to predict its *motion profile* (in brief, its future moves).
2) **Inconsistencies:** it must be reckoned that the information handled by a host may not be correct, either because it is stale or due to the presence of malicious hosts.

[1]By *service requirements*, we mean indications that certain services should be available at specific times and places.

The comments made in [1], [2] evidence that there is still much research to do regarding what can be done when there are limitations in the knowledge available about a MANET. The authors in [1] suggested that it could be a good starting point to assume *perfect knowledge*, and then examine the implications of gradually eliminating that assumption; that approach can bring light from a theoretical point of view, but the initial assumption makes it inapplicable in practice. On their part, the authors in [2] proved the advantages of having the hosts gather and exploit knowledge about the network, but they left the consideration of "*the partial observability of the domain*" as a crucial feature to consider in the future to attain much better results. None of the two approaches has yet considered the management of contradictions.

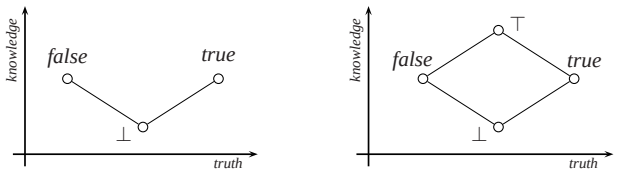
Our goal is to endow the hosts with the ability to reason over uncertain and inconsistent knowledge. In the following section, we discuss what kind of formalisms are needed to enable that reasoning. After that, we present the sketch of the solution we plan to build, describing its facilities to improve MANETs' dependability. The paper finishes with indications on how we are implementing our solution by borrowing mechanisms from the field of specifying distributed real-time systems.

II. Enabling Formalisms

As a cornerstone for our work, we cannot build a solution to reason about MANETs over the classical Boolean logic, because this logic cannot reflect uncertainty (everything is either *true* or *false*) and it is trivialized in the presence of inconsistencies (the principle of *ex falso quod libet*: anything follows from a contradiction [3]). It is necessary to lean on an alternative, more expressive and reliable logic.

Kleene's three-valued logic [4] was the first one capable of modeling uncertainty. To this aim, a new logical value (commonly denoted by $\perp$, read "*bottom*") was introduced to represent the missing knowledge. As shown in Fig. 1(a), this value lies halfway between the *truth levels* of *false* and *true* (certainly, *unknown* is neither falser than *false*, nor truer than

true); it is also placed in a *knowledge level* below *false* and *true*, thus capturing the point that learning new information can turn the *unknown* facts into known ones.



(a) Kleene's (b) Belnap's

Fig. 1. The Basic Multi-valued Logics

Applied to a modeling formalism, the three values of Kleene's logic allow differentiating what is known to be *true* (meaning "*allowed*", "*possible*", "*reachable*" or "*available*"), what is known to be *false* (meaning the opposite), and what is simply *unknown*. However, this does not serve to model the contradictions that arise when a fact is reported to be both *true* and *false*. For this purpose, Belnap's logic [5] introduced a fourth truth value (denoted by $\top$, read "*top*") to explicitly indicate the facts about which there is contradictory knowledge (see Figure 1(b)).

Several authors have generalized the ideas of Kleene and Belnap. In [6], new logical values were introduced between $\bot$ and $\{false, true\}$ to identify cases when partial knowledge is enough to obtain certain conclusions (therefore removing the need to complete the information available); likewise, [7] handled new values between $\{false, true\}$ and $\top$ to capture levels of agreement when several sources provide contradictory information. Those features are welcome in the modeling of MANETs, provided that we reach a balance between expressiveness and complexity (obviously, the more logical values, the more complex the reasoning over them).

III. Outlining a Solution

Our work aims at using formal modeling techniques to construct a layer like the one depicted in Fig. 2, which is to be placed between the applications and networking levels of every host in a MANET.

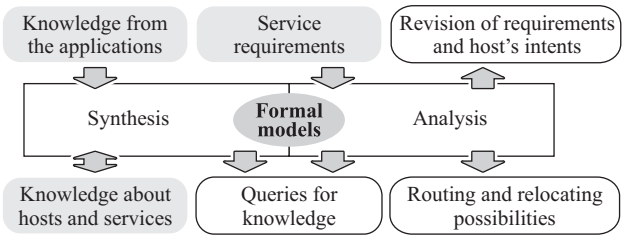


Fig. 2. A Layer to Reason about Service Provision

This layer receives information about the host that lodges it from the applications level, indicating whatever is known about the host's intended motion profile and the services it plans to provide. From the networking level, it receives analogous information about other hosts. From either source, it can also receive information about the impossibility to take certain moves (e.g. due to the presence of walls), the fact that a given service can only be provided by certain hosts, the possibility to migrate or clone certain services, etc.

Given a suitable language for knowledge exchange, the "*Synthesis*" module can use the commented information to generate formal models representing the intents of the known hosts and, by extension, the present and future states of the network they make up. Over those models, the "*Analysis*" module can check whether it is possible to satisfy the service requirements issued by applications or human users.

We plan to build the "*Analysis*" module over *model-checking* [7], [8] techniques, because they are fully systematic (even with multi-valued logics) and not at all limited to finding YES/NO responses. Quite the opposite, model-checking makes it possible to identify the reasons for a negative outcome and the particular situations that yield a positive one. Combining this characteristic with a suitable modeling formalism (according to the comments of the preceding section), we expect our solution to enable the following features (the outputs in Fig. 2):

- If it is found that a service requirement can be fulfilled, the traces of the model-checking algorithm should serve to derive routing possibilities in the form of *direct*, *multihop* or *disconnected* routes [9]. If the requirement cannot be fulfilled, it should be possible to automatically derive planning decisions to relocate (migrate, clone, etc.) certain services. In this case, the traces of the model-checking algorithm would help deriving the communications needed with the hosts involved in the relocations.
- If there is not sufficient knowledge to conclude whether it is possible to fulfill a service requirement, the modeling of uncertainty allows identifying accurately what further information would be necessary to conclude. Thus, a host can inquire others just about the precise knowledge it needs. Similarly, when there are contradictions that impede concluding about the possibility to fulfill a service requirement, it is possible to ask other hosts to support one of the conflicting stances.
- Finally, the *Analysis* module should be able to revise the service requirements, to specify any details left open (related to spatial or temporal conditions), or to recommend changes to the intended plans in case these impeded fulfilling the requirement. The suggestions would be interpreted, for example, as "*instead of the path you indicated, follow this alternative one*" or "*stay a little longer in that specific location*".

These features can be combined into a practical solution in terms of computational cost, because the explicit support to deal with partial knowledge allows each host to tune the amount of information it handles according to its computing and memory capabilities.

IV. Work in Progress

To furnish the features commented in the preceding section, we are resorting to solutions from the incremental development of real-time systems, specifically from the SCTL/MUS-T methodology [10]. The motivation for this approach stems

[12]	Incremental synthesis of formal models from temporal logic statements. Model-checking of desirable system properties expressed in temporal logic.
	Synthesis of formal models along with the progressive acquisition of knowledge about the MANET. Analysis of service requirements.
[13]	Exploiting the modeling of uncertainty to guide requirements elicitation tasks. Revisions of a system model when there are problems with the satisfaction of a desirable property.
	Guiding the search for knowledge when it is not possible to conclude about the possibility to fulfill a service requirement. Revisions of the formal model of the MANET to solve the problems with a service requirement.
[14]	Mechanisms to address the contradictions between the viewpoints of different developers.
	Mechanisms to resolve the inconsistencies between multiple sources reporting contradictory information about the MANET.
[15]	Combined use of temporal logic and scenarios to specify requirements. Revision of the requirements and scenarios provided by the developers from modifications of a system model.
	Entry of service requirements as temporal logic statements or scenarios. Derivation of alternatives to complete/modify the service requirements from the revisions of the formal model of the MANET.
[16]	Decomposition of formal models to isolate certain features of functionality.
	Modularization of the knowledge bases to avoid problems of state space explosion in the formal models.
[17]	Automatic revision of several system components when their conjoint behavior implies problems with a desirable property.
	Derivation of routing/relocation possibilities and revisions of the hosts' intents.
[18]	An agile and adaptive analysis-revision cycle, that suggests evolutions to solve problems in a system's specification learning from the acceptance/rejection of evolutions considered in the past.
	A low-cost scheme to browse the different possibilities to revise the planning of the MANET, without ever insisting on unwanted solutions.

TABLE I

FEATURES OF THE SCTL/MUS-T METHODOLOGY AND THEIR APPLICATION TO THE PLANNING OF MANETS

from the fact that, in close resemblance with the planning of service provision in MANETs, incremental development requires support to model time dependence, to analyze partial versions of a system (i.e. systems at intermediate stages of development), and to solve the contradictions raised by conflicting viewpoints of different developers.

SCTL/MUS-T leans on a six-valued logic (the first generalization of Kleene's) to model uncertainty, and adds three other values to handle inconsistencies (the minimal solution that provides the advantages of generalizing Belnap's ideas) —the details can be found in [11], [12]. Besides, it employs a sort of temporal logic as the language to express the functional requirements of a real-time system. That language serves to exchange knowledge between hosts in a MANET and to enunciate service requirements, since it can readily express delays, usage times, dependencies between the requested services and spatial locations, etc. Finally, a scenario-like formalism is also available, which may be more accessible for human users to enunciate service requirements than temporal logic.

Table I enumerates some of the features implemented in SCTL/MUS-T and their expected use in the field of MANETs, evidencing the great synergies between the specification of distributed real-time systems and the planning of service provision in MANETs. All the features of SCTL/MUS-T were devised to help developers reach a correct and complete specification of a system in the face of uncertainty and inconsistencies. Substituting '*developers*' for '*users and applications*', we do believe that the same features can be successfully applied to improve the dependability (or, at least, the predictability) of mobile ad-hoc networks. The implantation of this approach is currently on the go, and we are ready to initiate research on the best usage policies for our layer from the applications' point of view (whether to use a *push* or *pull* model for knowledge dissemination, what to do when a service requirement ends up unfulfilled contrary to the expectations, etc.).

ACKNOWLEDGMENT

The authors from the University of Vigo received funding for this work from the Xunta de Galicia Basic Research Project PGIDIT04PXIB32201PR.

REFERENCES

[1] R. Sen, G. Hackmann, G.-C. Roman, and C. Gill, "Opportunistic exploitation of knowledge to increase predictability of agent interactions in MANETs," in *Proc. of the 4th International Workshop on Software Engineering for Large-scale Multi-agent Systems*, May 2005.

[2] Y.-H. Chang, T. Ho, and L. Pack Kaelbling, "Mobilized ad-hoc networks: A reinforcement learning approach," Artificial Intelligence Lab., Massachusetts Institute of Technology, Tech. Rep. AIM-2003-025, 2003.

[3] N. C. A. da Costa, "α-models and the systems T and T^*," *Notre Dame Journal of Formal Logic*, vol. 15, no. 3, pp. 443–454, 1974.

[4] S. C. Kleene, *Introduction to Metamathematics*, ser. Bibliotheca Mathematica. North-Holland, 1952, vol. 1.

[5] N. D. Belnap, "A useful four-valued logic," in *Modern uses of multiple-valued logic*. Reidel, 1977, pp. 7–37.

[6] M. Fitting, "Kleene's logic, generalized," *Journal of Logic and Computation*, vol. 1, no. 6, pp. 797–810, 1991.

[7] S. Easterbrook and M. Chechik, "A framework for multi-valued reasoning over inconsistent viewpoints," in *Proc. of the 23rd International Conference on Software Engineering*, May 2001.

[8] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model checking*. The MIT Press, 2000.

[9] T. Rappaport, *Wireless communications: Principles and practice*. Prentice Hall, 2002.

[10] A. Fernández-Vilas, J. J. Pazos-Arias, A. Gil-Solla, R. P. Díaz-Redondo, J. García-Duque, and B. Barragáns-Martínez, "Incremental specification with SCTL/MUS-T: A case study," *Journal of Systems and Software*, vol. 70, no. 2, pp. 189–208, 2004.

[11] B. Barragáns-Martínez, J. J. Pazos-Arias, and A. Fernández-Vilas, "On measuring levels of inconsistency in multi-perspective requirements specifications," in *Proc. of the 1st International Conference on the Principles of Software Engineering*, Nov. 2004.

[12] J. J. Pazos-Arias and J. García-Duque, "SCTL-MUS: A formal methodology for software development of distributed systems. a case study," *Formal Aspects of Computing*, vol. 13, pp. 50–91, 2001.

[13] J. García-Duque, J. J. Pazos-Arias, and B. Barragáns-Martínez, "An analysis-revision cycle to evolve requirements specifications by using the SCTL-MUS methodology," in *Proc. of the 10th IEEE International Conference on Requirements Engineering*, Sept. 2002.

[14] B. Barragáns-Martínez, J. García-Duque, J. J. Pazos-Arias, A. Fernández-Vilas, and R. P. Díaz-Redondo, "Requirements specification evolution in a multi-perspective environment," in *Proc. of the 26th Intl. Computer Software and Applications Conference*, Aug. 2002.

[15] J. J. Pazos-Arias, J. García-Duque, M. López-Nores, and B. Barragáns-Martínez, "Eliciting requirements and scenarios using the SCTL-MUS methodology. The shuttle system case study," *ACM Software Engineering Notes*, vol. 30, no. 4, 2005.

[16] J. García-Duque, M. López-Nores, J. J. Pazos-Arias, A. Fernández-Vilas, R. P. Díaz-Redondo, A. Gil-Solla, M. Ramos-Cabrer, and Y. Blanco-Fernández, "Guidelines for the incremental identification of aspects in requirements specifications," *Requirements Engineering*, 2006, in press.

[17] M. López-Nores, J. J. Pazos-Arias, J. García-Duque, B. Barragáns-Martínez, R. P. Díaz-Redondo, A. Fernández-Vilas, A. Gil-Solla, and M. Ramos-Cabrer, "Tracing integration analysis in component-based formal specifications," in *Proc. of the 7th IFIP International Conf. on Formal Methods for Open Object-based Distributed Systems*, June 2005.

[18] M. López-Nores, J. J. Pazos-Arias, J. García-Duque, and B. Barragáns-Martínez, "An agile approach to support incremental development of requirements specifications," in *Proc. of the IEEE Australian Software Engineering Conference*, Apr. 2006.