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

INFOCOM’06 Poster Session

Martín López-nores*, David Pereira-Paz†, José J. Pazos-Arias*, Jorge García-Duque* and Esther Casquero-Villacorta*

* Department of Telematics Engineering, University of Vigo
{mlnores,jose,jgd,esther}@det.uvigo.es

† DMR Consulting
david.pereira.paz@dmr-consulting.com
Introduction

- The context
- The problem
- Proactive services planning
- Obstacles

Enabling formalisms

Outlining a solution

Work in progress

End
The context

- **Mobile ad hoc networks** (MANETs).
  - Highly dynamic computing environments, collectively supported by the hosts they comprise.
  - The hosts collaborate to support complex tasks, each one making use of the *services* provided by the others.
The problem

- The dynamism of a MANET makes it hard to offer guarantees of **service provision**.
  - The applications may suffer disconnections at any time.

- To ensure a certain level of **predictability** (and thus **make the services more dependable**), the hosts must expose some information about themselves, including:
  - **Services** offered for other hosts to use.
  - **Motion profiles**: characterizations of the intended spatial trajectories against time.
  - **Elasticity properties**: whether the services can be migrated from one host to another, cloned, leased, etc.
Proactive services planning

- **Knowledge** allows the hosts to guess how the network is set up at a given moment, and how it will be in the near future.
  - The hosts can find out whether it is possible to satisfy the **service requirements**.
  - In case not, the hosts can take proactive actions to ensure satisfiability.
    - Moving to specific locations,
    - accomplishing service relocations,
    - etc.

* Indications that certain services should be available at specific times and places (can be issued by applications or by human users).
**Example:**
- There is a software agent $A$ running in host $h_1$.
- $A$ needs to use some data stored in host $h_2$, which is out of communication range.

- Knowing the motion profile of $h_3$, $h_1$ finds a **disconnected route** to $h_2$ through $h_3$.
- So, $h_3$ can be used as a relay to take the agent $A$ from $h_1$ to $h_2$; alternatively, $A$ can access the data on $h_2$ while the communication ranges of $h_2$ and $h_3$ overlap.
Obstacles

- **Uncertainty**: it is unrealistic to assume that a host may have complete knowledge about the MANET.
  - Every host gathers information in a progressive way, from an initial situation when it knows nothing about others.
  - It frequently happens that a host cannot expose complete information about itself (for example, it may not be able to predict its motion profile).
  - Complete knowledge could require managing too much information for the limited computing/memory capabilities of a mobile device.

- **Inconsistencies**: not only partial, the knowledge gathered about the MANET may also be incorrect.
  - The dynamism of the network can make the gathered information stale.
  - There may be malicious hosts publishing erroneous information.

- **Our goal**:
  - To build a framework for knowledge dissemination and exploitation...
  - ... endowing the hosts with the ability to reason safely over uncertain and inconsistent knowledge.
Enabling formalisms

- Limitations of Boolean formalisms
- Handling uncertainty
- Handling inconsistencies
- More granularity?

Outlining a solution

Work in progress

End
Limitations of Boolean formalisms

- **We cannot** build a solution to reason about MANETs over the classical Boolean logic.
  - It cannot reflect uncertainty (everything is either true or false), making it possible to mistake for false what is indeed unknown.
  - It is trivialized in the presence of inconsistencies (the principle of *ex falso quod libet*: anything follows from a contradiction).

- It is necessary to lean on an alternative, more expressive and reliable logic.
Handling uncertainty

Kleene’s three-valued logic was the first one capable of modeling uncertainty, introducing a new logical value (⊥) to represent the missing knowledge.

- ⊥ lies halfway between the truth levels of false and true (certainly, unknown is neither falser than false, nor truer than true).
- ⊥ has a knowledge level lower than false and true, meaning that learning new information can turn the unknown facts into known ones.

Applied to a formal modeling of MANETs, Kleene’s logic can differentiate what is known to be true ("allowed", "possible", "reachable" or "available"), what is known to be false (meaning the opposite), and what is simply unknown.
Handling inconsistencies

- Kleene’s logic does not serve to model the **contradictions** that arise when a fact is reported to be both *true* and *false*.

- Belnap’s logic introduced a **fourth truth value** ($\top$) to indicate the facts about which there is contradictory knowledge.
More granularity?

- New logical values can be introduced between $\bot$ and \{false, true\} to identify cases when partial knowledge is enough to obtain certain conclusions.
  - This removes the need to manage complete knowledge bases.

- New values between \{false, true\} and $\top$ can capture levels of agreement when several sources provide contradictory information.
  - Useful to resolve contradictions.

- We must reach a balance between expressiveness and complexity.
  - The more logical values, the more complex the reasoning over them.
Outlining a solution

- Between the network and application levels
- Harnessing model-checking
- Negotiating service requirements

Work in progress
Between the network and application levels

- We are building a layer to reason about service provision in MANETs using **formal modeling techniques**.

  ![Diagram showing the process between the network and application levels]

  - **Inputs:**
    - **From the application level:** knowledge about the lodging host (intended motion profile, services it plans to provide, etc.)
    - **From the network level:** analogous knowledge about other hosts.
    - **From either source:** knowledge about the impossibility to take certain moves, the fact that a given service can only be provided by certain hosts, the availability of certain services to be migrated or cloned, etc.
Harnessing model-checking

- The **Synthesis** module uses the knowledge to generate formal models that capture what is known about the present and future states of the MANET.

- Over those models, the **Analysis** module checks the satisfiability of the service requirements using **model-checking techniques**.
  - Model-checking is **fully systematic**, even with multi-valued logics.
  - Moreover, it is **not limited to finding YES/NO responses**.
    - If the model-checker finds that a service requirement can be fulfilled, the traces it followed over the formal models provide **routing possibilities**, in form of *direct*, *multihop* or *disconnected* routes.
    - If the requirement cannot be fulfilled, the traces serve to automatically derive **relocation possibilities** involving service migrations, cloning, etc.
    - If the knowledge available does not serve to conclude about the satisfiability of the requirement, it is easy to automatically derive **queries for knowledge**.
The **Analysis** module can also suggest **revisions of 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 requirements.

The suggestions can be accepted, rejected or ignored.

This mechanism is the basis to implement **policies** by which the hosts can **negotiate changes** in the MANET to reach the configuration that best satisfies their service requirements.
Work in progress
- We are implementing the approach by borrowing solutions from the incremental development of real-time systems.

- Specifically, from the SCTL/MUS-T methodology.
  - A six-valued logic (the first generalization of Kleene's) to model uncertainty, and three additional values to handle inconsistencies.
  - The minimal solution to achieve the advantages of generalizing Kleene's and Belnap's ideas.
  - A sort of temporal logic as the language to express the functional requirements of a real-time system.
  - Suitable to exchange knowledge between hosts in a MANET and to enunciate service requirements.
  - A scenario-like formalism also available, more accessible for human users.

- Many features of SCTL/MUS-T are readily applicable to reasoning about service provision in MANETs.
Expected results

- A flexible solution to reason **soundly** about service provision in MANETs, which is not possible with Boolean formalisms.
  - Managing uncertainty and inconsistencies.

- The basis for **different policies** to negotiate service requirements and proactively define the best network configuration for their interests.

- A **practical solution** in terms of computational cost.
  - 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.