Simulating MANET-Hosted Service-Based Systems

Petr Novotny and Alexander L. Wolf
Department of Computing, Imperial College London
\{p.novotny09,a.wolf\}@imperial.ac.uk

Bong Jun Ko
IBM T.J. Watson Research Center
bongjun_ko@us.ibm.com

Abstract—The emerging use of mobile ad hoc networks combined with current trends in the use of service-based systems pose new challenges to accurate simulation of these systems. Current network simulators lack the ability to replicate the complex message exchange behaviour of services, and service simulators do not provide accurate simulation of mobile network properties. In this paper we provide an overview of a framework for capturing both the service behavioural model and a precise network simulation engine.

I. INTRODUCTION

Service-based systems are becoming integral elements of tactical wireless, mobile networks. Structured as a complex of interdependent, interrelated services, they can provide end users with a rich set of information combined from multiple sources. This model of information delivery is in sharp contrast to the flow-based, point-to-point model that has been traditionally supported in tactical networks. Analysing the behaviour of such complex applications is a challenging task. The challenge is compounded when services run on highly dynamic networks such as mobile, ad hoc networks (MANETs), in which the fluidity of the underlying network greatly impacts performance and availability.

Service-based systems consist of some number of services that interact with one another in order to complete client requests. Each service provides a set of methods for use by other services or clients. A method may use any number of other methods provided by other services to carry out its functionality. Thus, services are interconnected with each other. Clients initiate the flow of service requests by sending messages that request method executions, and then wait for some response.

A simulator that can closely replicate the behaviours of service-based systems running on MANETs can be a valuable analysis tool. In particular, it can provide a means to predict performance when designing, deploying, or managing the system. Furthermore, it can model network traffic workloads that are characteristic of MANETs. Existing simulation tools fall short of providing such capabilities. Packet-level network simulators, such as NS-3\(^1\) and QualNet\(^2\) provide detailed implementations of mobile, wireless networks, but lack the ability to replicate complex behavioural aspects of service-based systems. These aspects are addressed in high-level service simulators [2], which unfortunately do not provide a means to simulate a complex network layer.

In this paper we introduce a new simulation tool for service-based systems hosted on MANETs. With the system and behavioural models of services built on top of a packet-based simulator, our approach allows the replication of various critical aspects, such as the cascading flows of messages in complex conversations, comprehensive client-driven workload profiles, and the propagation of faults through services. Furthermore, the simulator provides generic and easily extended models that can be used to capture modern service-based platforms, such as SOA, operating in MANET or hybrid networks.

II. SIMULATOR DESIGN

A. Architecture

The simulator engine is built on top of the discrete event network simulator NS-3, extended with additional higher-level abstraction layers for simulating service entities and their interactions. NS-3 provides a comprehensive network simulation with detailed implementation of low-level network protocols. However, NS-3 provides only a simple mechanism for simulating the flow of packets from point to point. At the highest abstraction level, NS-3 provides sockets and packets as a basic network data transfer mechanism.

Our simulation engine encapsulates the socket layer into a messaging layer that provides the abstraction of messages exchanged between services and between clients and services. The messaging layer is then encapsulated into a service layer that provides abstractions for entities (services and clients) and their interconnection models. Finally, the simulator provides methods for engineers to configure the simulation scenarios and their parameters, and to run the simulation. In what follows, we describe these models and the current implementation.

B. Simulation models

The simulator consists of several abstraction models: entities, interconnections, workloads, faults, and messages.

1. **Entity models**: Entity models provide the building blocks of the service-based system simulation.

   - **Clients** represent applications used by end users. Each client behaves as an autonomous entity that contacts a set of services at times (random or deterministic) configured by the engineer.
   - **Services** represent autonomous self-contained functional units. Each service has a pre-configured set of methods
that are available to be used by clients and other services. Each method contains an abstract definition of its computation consisting of delays to simulate processing time, and a set of steps that send requests to other services.

(2) Interconnection model: The interconnection model defines the methods in other services with which each entity in the service-based system interacts (i.e., sends service requests and receives responses). Two types of interconnections are defined: client-to-service, and service-to-service. Currently, a randomised interconnection model with predefined connection probabilities are provided.

(3) Message model: There are three types of messages exchanged between entities: requests, responses, and exceptions. Request messages are used to invoke methods in other services, while response messages are sent by services back to the requesting entity upon the completion of the requested method. Exception messages are used to propagate fault symptoms caused by network or service faults. The flow of messages exchanged between services during the processing of a client request is called a conversation. In the simulator, all messages contain information about the conversation to which they belong. The conversation information is designed to replicate the behaviour of WS-* standards such as WS-Addressing.

(4) Fault model: Services running on MANETs are exposed to potentially frequent faults in the network due to network instability and in the services themselves due to resource constraints (and bugs). While network failures are immediately captured by the network simulation layer, we must include a service fault model that defines the failure behaviour of the services, which are currently based on probabilistic or deterministic variables.

(5) Workload model: In service-based systems the workload is initiated by clients sending requests to services. The workload model defines the rates of such requests. In our current implementation, clients repeatedly, and at pre-configured random times, select at random one method to request out of the set of available service methods, and then waits for a response.

C. Simulation scenarios

The simulation scenarios are created by a configuration generator that creates scenario configurations based on a set of parameters of system characteristics. In addition to the network parameters configured through NS-3’s configuration (e.g., number of nodes, mobility, wireless link characteristics, etc.), the services, clients, and their interactions and behaviours, including how and where the services are hosted, are configured for the above models. During the simulation run, the simulator records events, such as service message exchanges and fault symptoms, into trace files for analysis. Below, we provide an example of how the simulation is used.

III. EXAMPLE

We have used the simulator to exercise both a dependence analysis tool [1] and a fault-localisation tool. For dependence analysis, the dependencies between services are detected by examining message traces. For fault localisation, dependence information is combined with data from fault traces to uncover the root causes of client failures.

In the future, we intend to use the simulator in exploring the fault localisation problem in mobile networks [3] and also for multi-level fault localisation, in which the availability of the high-level messages and fault-trace data in combination with low-level packet traces is particularly important.

ACKNOWLEDGEMENTS

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES