MobLab: A Mobility Emulation Platform

Azarias Reda Brian Noble
University of Michigan
{azarias, bnoble}@umich.edu

ABSTRACT

Mobility powered systems provide the core routing mechanism in many ad-hoc and delay tolerant networks. Evaluating such systems under real life scenarios is often not practical because they involve multiple moving participants over a wide area. As a result, the principal way these systems are evaluated is though discrete event simulators that are often specific to the system at hand. While these systems provide a valid way to test ideas, and get some numbers, they often require systems to be built towards the simulation platform rather than deployment. As a result, researchers often stop at simulation. MobLab provides a mobility emulation platform for evaluating systems, while still working towards deployment. By providing an easy way to iteratively debug and evaluate production ready code, MobLab enables developers to deploy their mobility powered systems as soon as they are done evaluating them, rather than going through a different phase of implementing a simulation result. Our system can utilize mobility models or recoded traces to drive node mobility in the system, and provides a simple way for aggregating and presenting emulation results. MobLab is built on top of EmuLab, a network emulation platform, and has been successfully used to evaluate two mobility based systems.

1. INTRODUCTION

Experimental testbeds play a significant role in building systems, and this can't be any truer in the case of mobility powered systems. These systems include delay tolerant networks, meant to operate in challenged and extreme environments where node mobility is a significant part of the system, or ad-hoc networks, which might be formed among vehicular and mobile nodes. Evaluating these systems involves multiple nodes moving in an environment, while possibly connected with a network link, and occasionally running into each other. Under the circumstances, it is often impossible to recreate a real life scenario without significant monetary expenditure, let alone iterate on a design over a long period of time. As a result, experimental testbeds form an essential component of the development cycle.

In the past, simulation has been the de facto approach in evaluating mobility powered systems [4, 5, 6]. These systems provide a reasonable interface for trying out different routing protocols, and evaluating resource utilization. While this is an acceptable way to evaluate different ideas, it also has the undesired property that systems are written towards a simulation platform rather than deployment. This has been the case in many research projects where the end result has been a simulation result, rather than a system deployed in the wild. This is understandable from the researchers' point of view as it takes significant effort to design a system, and essentially implement it twice, once for simulation and once for deployment.

The other end of this spectrum is building real life testbeds that can be used to implement ideas. These often come after significant capital expenditures, and are often limited to a campus or a group of researchers, rather than the community in general. While results obtained from such deployments are excellent, they are hardly repeatable by other researchers who do not have access to the deployment. This is a difficult problem in mobility powered systems, which are often targeted towards challenged network environments, and might involve several groups of researchers in different countries, spanning from design through several cycles of evaluation.

This poster presents MobLab, our mobility emulation platform that aims to provide a sensible middle ground in evaluating mobility powered system. In a nutshell, MobLab enables developers to evaluate production ready code using significantly less resources than a real life testbed. Emulation accomplishes this by reproducing the needed behaviors of the emulated environment, without requiring changes to the system to be evaluated. While researchers can still simulate their ideas, MobLab provides an efficient way to evaluate mobility powered systems by writing deployment ready code that can be directly taken to production once debugging and evaluation is completed successfully.

2. RELATED WORK

Simulation has been the primary way to evaluate mobility powered systems. While network simulators like NS-2 and OMNeT++ provide good generic support for packet-based communications, ad-hoc and delay tolerant network builders often rely on discrete event simulators to evaluate their system [4, 6]. These systems generally take some mobility model, and provide an abstraction giving developers a simple way to evaluate routing and resource utilization of systems. Our system also provides similar benefits, but enables developers to build their systems towards deployment rather than a specific simulator. This in turn reduces the effort needed to build production ready systems.

The ONE simulator for DTN protocol evaluation [5] provides a very useful set of tools and reference implementations, incorporating several lessons from prior simulation platforms for mobility based systems. Besides supporting a number of mobility models, the simulator also comes with a few well known routing protocols which make comparisons easier. As discussed

above, while mobility simulation provides a good way to test out ideas, emulation provides a better way because it allows for evaluation and deployment preparation simultaneously.

On the other hand, DieselNet [1], the bus based adhoc network, provides a real life testbed with more than 30 participating buses with scheduled routes. Whenever possible, this is arguably the best way to evaluate mobility systems, but unfortunately, not many researchers have access to such a testbed. Our work aims to provide a close functionality, but at fraction of the cost. In addition, when such a platform is available, a system evaluated using our system should be able to take advantage of it without any modification.

3. DESIGN

MobLab is built on top of EmuLab, the network emulation platform from the University of Utah [7]. EmuLab provides a framework for specifying an arbitrary network topology, allowing the user to have a predictable and repeatable environment, including PC nodes with full root access. EmuLab nodes can be connected directly to each other, or through a switch, allowing traffic shaping in all cases. MobLab uses this capacity to build a mobility layer that can emulate how nodes move about the system, providing a straightforward way to evaluate mobility powered systems. At its core, a MobLab session consists of several general purpose computers running deployment ready code, each emulating a node in the mobility powered system. These nodes could be stationary or mobile. MobLab emulates the mobile nodes, and their movement relative to other nodes in the system, using bundles that are sent in the system according to corresponding mobility models.

MobLab can support an arbitrary mobility model, or trace of real life mobility records. A number of mobility models have been suggested for capturing how nodes move in the environment, ranging from random walks and random waypoints to those mimicking working day mobility [2]. A developer can also feed MobLab with any custom routine that implements a new mobility model. In addition, the emulator can use traces of real life mobility, such as those found in the CRAWDAD [3] collection, which has several traces from different locations. The mobility model guides how nodes interact with each other. In most cases, a portion of the nodes will be mobile, while others are stationary.

MobLab uses a tracker that is well connected to all nodes to coordinate mobility in the system. Mobility for each participating node can be independently provided, allowing developers to evaluate a mix of different models in their systems. Initial configuration at the tracker involves describing the per-node specifications with regards to location, mobility, network and buffer capacity. MobLab provides automating tools to generate these specifications based on a pattern provided. These tools come handy when dealing with a large number of nodes in the system.

Once nodes are properly configured, MobLab handles booting and initializing them with the provided piece of code, as well as performing any arbitrary execution supplied by the developer. The tracker's high bandwidth link to each node in the system is used to emulate how mobile nodes move relative to other participating nodes. It's often the case that the nodes have varying network capacity with respect to each other, as specified in their initialization. However, the MobLab tracker maintains a separate high bandwidth side channel to each node used for mobility emulation through bundle exchange.

In order to take advantage of the emulation platform, application developers only need include an interface in their applications to respond to communication from the tracker. The supported interface allows the node to accept bundles from the tracker, and later respond with emulation results. The individual nodes can collect any pertinent data during the evaluation process, which is later aggregated at the MobLab tracker for further processing by the user.

MobLab provides tools for analyzing emulation results collected from participating nodes. These tools can be used to generate different characteristics of the experiment such as delivery rates, number of mobility and network hops, buffer utilization at nodes and classification of results to buckets. As expected, developers can also provide custom analysis routines that can be run on the aggregated data. Finally, MobLab can generate source files for plotting graphs using gnuplot[1]. By providing a number of example scripts, MobLab makes it easy to customize these graphs as needed.

In summary, MobLab provides an easy way for emulating mobility powered systems. This enables developers to write systems geared towards deployment, while still allowing for iterative evaluation and feedback. This is especially useful when dealing with mobility powered systems where real life evaluation testbeds are not practical in most cases. MobLab was built out of necessity, and it has been successfully used to evaluate two mobility based systems—a hybrid overlay network for bulk data in challenged environments, as well as a probabilistic routing ad-hoc network.

4. REFERENCES

[1] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In *Proc. IEEE INFOCOM*, April 2006.

[2] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *WCMC 2002*.

[3] T. Henderson, D. Kotz, I. Abyzov, and J. Yeo. CRAWDAD trace set movement. http://crawdad.cs.dartmouth.edu.

[4] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *SIGCOMM '04*. ACM, 2004.

[5] A. Keränen, J. Ott, and T. Kärkkäinen. The one simulator for dtn protocol evaluation. In *Simutools 2009*.

[6] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *Lecture Notes in Computer Science*, 3126:239–254, January 2004.

[7] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. An integrated experimental environment for distributed systems and networks. In *OSDI'02*, Boston, MA, 2002.

[1] `http://www.gnuplot.info/`