

Simulation platform for distributed package management network

Internship Proposal

Fabien Dagnat Gwendal Simon

Computer Science Department, Telecom Bretagne, France

{fabien.dagnat,gwendal.simon}@telecom-bretagne.eu

Master Recherche 2010-2011

Keywords component based architecture, crowdsourced software, package management system, peer-to-peer network, Internet of Things

Context

Evolution of the Internet has endowed software engineering with collective knowledge and intelligence during the last couple of years. Today, the community of *Free and Open Source Software (FOSS)* contains typically several millions of software producers (from amateurs to professionals). The increasing popularity of *application stores* (e.g. more than 225,000 applications in the Apple AppStore for a total of 5 billions downloads since its inception two years ago ¹) confirms a major trend in the software industry. While crowdsourced engineering reinforces the proliferation of innovative software, it also propels us to revise current approaches for software distribution and deployment. Additionally, in the prospective *Internet of Things*, billions of devices running different operating systems require to be administered. Deploying and updating software packages in such a heterogeneous and dynamic context appears to be a challenging task.

Package Management System Modern software often consists of a large number of small packages, e.g. more than 25,000 in *Debian*. These packages have inter-dependent relationships that may easily be broken during the deployment life-cycle. On a computing device, the installed packages and their inter-dependencies are linked in a *dependency graph*, which should remain consistent while a package is installed, upgraded or removed[1]. This is guaranteed, in most current operating systems, by a package management system. Package management systems are in relationships with *software repositories* whose main responsibility is the storage of packages. The software distributor ensures the global consistency of the repository and certifies every user-created package before it can be integrated into the repository. Such centralized mechanism exhibits several limitations.

¹http://www.appleinsider.com/articles/10/06/07/apple_says_app_store_has_made_developers_over_1_billion.html

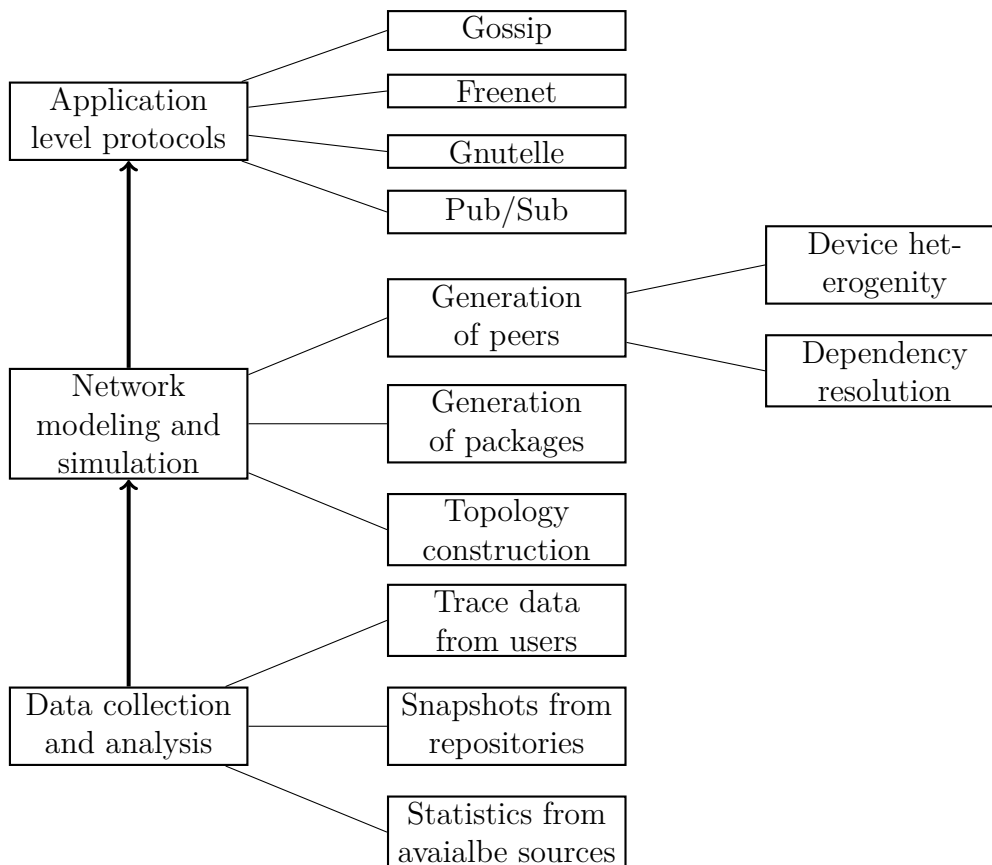


Figure 1: Architecture of the simulation program

Limitations of Current Architecture First, as the number of machines to serve and the number of software to update grow dramatically, the scalability of current systems has recently appeared to be a major issue. Although some works have addressed the management of large repositories [3, 4], the results are not expected to scale with the growth of the Internet. Second, the centralized architecture is vulnerable to failures and attacks. A peer-to-peer system would offer more reliability. At last, the centralized certification mechanism degrades the availability of user generated packages. Important overhead exists between the moment when a package is created (modified) and the moment it is released, see complains about the Apple AppStore ².

A Repository-less Package Management Network To fill the gaps between the current approach and the exigence imposed by the future software commercial ecosystem, we envision a repository-less package management system. Ideally, our system should not rely on any central administrator. Every participant is called a *peer*, which acts as both a client and a server. The workload of package storage and delivery is distributed across all the peers in the network. The deployment of software is expected to be guaranteed with better performance by managing more efficiently the power of network edges. In our previous work, we identified the major principles and challenges of constructing such a system. A repository-less system model was formally defined and several candidate approaches were proposed for

²<http://www.paulgraham.com/apple.html>

implementing our application. In a next step, we expect to detail the application level protocols and carry out realistic simulations to test our proposal. A blueprint of the simulation program is shown in Figure 1. At the application level, several techniques for efficient package delivery and advertisement should be evaluated, *e.g.* gossiping-based data dissemination, Publish/Subscribe based algorithms, classic P2P protocols like Freenet and Gnutelle *etc.* These algorithms should be executed on top of a virtual network that we model using general simulation tools such as *PeerSim* or *NS2*. To model peers and packages, two solutions are possible:

- **use trace data collected from real machines:** the sample peers should not be uniformly administered, and should be voluntary to provide their installation status periodically. It currently appears to be a difficulty for us to acquire data in a large scale network.
- **generate synthetic data that simulates a real network:** study the major properties and users' behavioral convergence in package distribution, and use these characteristics to model the real network. The information we are interested in includes: the distribution of peers' sizes, the distribution of packages among peers, the distribution of interdependency requirements, the updating frequency and the historical information of packages *etc.* Such information may be obtained by mining large software repositories or by extracting statistical results from existing works [5]. So far, we have acquired a large quantity of package metadata from the Debian snapshot website ³, and have conducted several preliminary statistics. More sophisticated tools for data collection and analysis still need to be elaborated.

Objective of the internship

The goal of the internship is to contribute to the dPAN (distributed Package Management Network) project, which has been initiated in Mars 2009, in the Department of Computer Science of Telecom Bretagne. The intern student will focus on the modeling and simulation of the package management network. The main tasks are:

- Identify the characteristics that are crucial for modeling the network.
- Collect and analyze realistic data, provide input to the simulation program
- Explore approaches to simulate the high heterogeneity in the Internet of things
- Model the network using appropriate simulation tools

Through the internship, the student may gain knowledges in multiple disciplines including graph theory, component based software engineering and distributed system. Skills in database and Object Oriented Programming are also appreciated.

References

- [1] M. Belguidoum and F. Dagnat. *Dependency management in software component deployment*. Electron. Notes Theor. Comput. Sci., 182:17–32, 2007.

³<http://snapshot.debian.org/>

- [2] *Ubuntu blueprint for using torrent's to download packages.* (Online) <https://blueprints.launchpad.net/ubuntu/+spec/apt-torrent>
- [3] R. Di Cosmo. Report on Formal Management of Software Dependencies. Deliverable WP2-D2.2, EDOS Project, April 2006.
- [4] F. Déchelle and F. Mancinelli. EDOS-Tools Tutorial: EDOS Tools for Linux Distributions Dependencies Management and Quality Assurance. In *OSS*, pages 363–364, 2007.
- [5] P. Shah, J. Morgan, J. Shettino, J.F. Pâris and C. Venkatraman A P2P-Based Architecture for Secure Software Delivery Using Volunteer Assistance. In *Eighth Int. Conf. on Peer-to-Peer Computing (P2P)*, 2008.