# DIVINEMAN: A System for Easing Management of Broadband Wireless Access Networks

Giacomo Bernardi, Mahesh K. Marina
School of Informatics, The University of Edinburgh, UK

## I. Overview

Technical advances and the commoditzation of wireless equipment have boosted the business volume of commercial Wireless Internet Service Providers (WISPs) around the world. It is more and more common for WISPs to deploy networks exclusively based on wireless technologies both for the backbone links and for the "last mile" access; wireless access can now be considered a valid low-cost alternative to traditional copper or fibre-based solutions.

While wireless links have major advantages over their wired counterparts in terms of cost and time-to-market, they pose major challenges for network planning and management. Along with the traditional configuration tasks, the network administrator has to cope with new problems such as frequency planning, routing over time-varying channels and the difficulties of in-band management of remote devices. Such difficulties are exacerbated in *community-driven deployments* with greater likelihood of device heterogeneity and lack of trained on-site personnel, stressing the need for a simplified management scheme.

We propose DIstributed and VIsual NEtwork MANagement (DIVINEMAN) system to ease the task of managing WISP network infrastructures, with a particular focus on rural scenarios where the availability of IT infrastructure and services is poor. Next section describes the network model under consideration. Section III reviews most common network management approaches and protocols with the intent of analyzing the reasons behind their wide adoption and identifying their limitations. We then outline our proposed approach in Section IV.

## II. Network Model

A typical Broadband Wireless Access (BWA) network consists of two types of devices: Base Transmitting Station (BTS) and Customer Premise Equipment (CPE). BTSs are devices located on transmission towers with sectorial or directional antennas and several radio interfaces; they are connected to one another via independent point-to-point links, commonly using directional antennas. On the other hand, CPEs are usually equipped with one or two wireless interfaces and are located at the subscriber premises, typically on a roof; each of them connects to a local BTS in a point-to-multipoint configuration, in which a common wireless channel is shared between several CPEs (subscribers).

## III. State of the Art

Broadly speaking, network management indicates the set of activities that have to be carried out in order to ensure uninterrupted network operation and ensure stable performance. Several models have been proposed to classify management activities. One is the FCAPS scheme [1], which categorizes actions into fault, configuration, accounting, performance and security management. Another is the Telecommunications Management Model (TMN) approach suggested by the ITU [2], which vertically partitions the management infrastructure into five layers of different responsibilities.

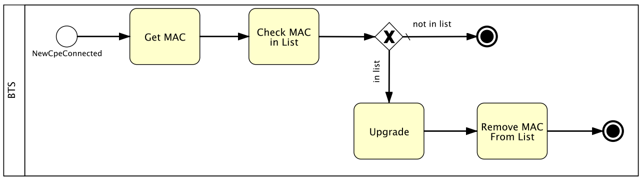
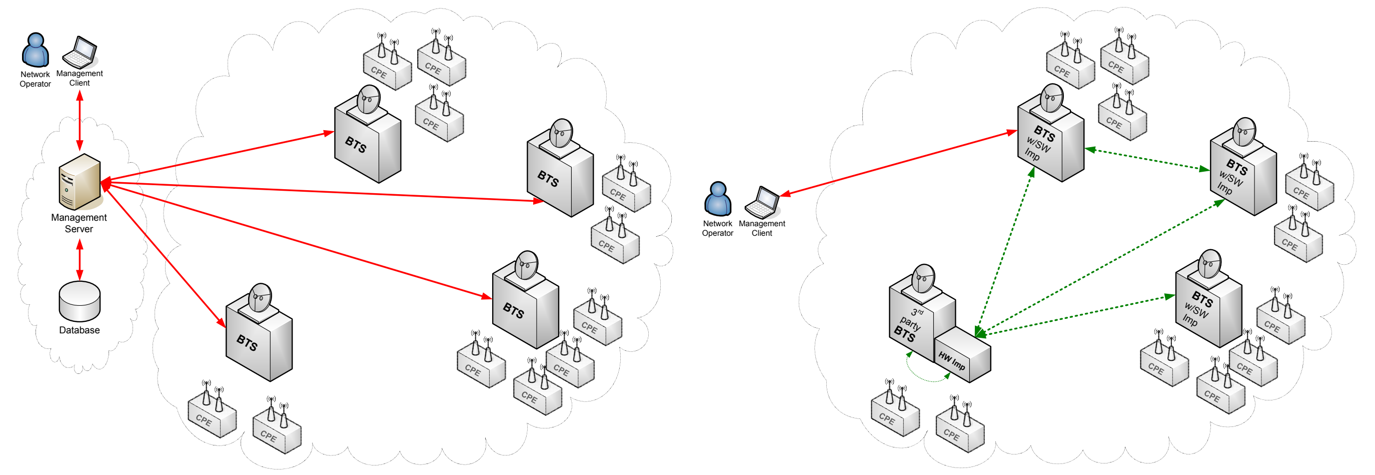


Fig. 2: NPM workflow specification for firmware upgrade of network devices.

Despite the efforts from Internet standards organizations to introduce a common management framework, there is a plethora of proprietary solutions introduced by hardware vendors for the administration of their own products. The only exception is the Simple Network Management Protocol (SNMP) [3], a key component of the Internet-Standard Management Framework, which has itself become a synonym for network management. Within three decades it has become the primary method for polling remote network devices and controlling their configurations. Its success is unanimously attributed to the minimalist design which made it easy to implement, and its lightweight nature making device vendors adopt it even when processing power and memory are limited. However, managing devices from different vendors using SNMP is non-trivial as the protocol defines only the syntax of data but not its semantics, which is up to the product developer. Also, a common practice for device manufacturers is to implement management access over SNMP outside the standard "MIB tree", instead providing most of the relevant management data using a custom addressing scheme. Although proprietary software tools can make use of this private information, many manufactures provide scarce documentation about the available object identifiers (OIDs) in the private MIB tree and no details are given about how the information is actually obtained. This semantic heterogeneity is thus an obstacle for unified network management and development of generic management software.

Over the years it became clear that SNMP was becoming something different from its original aim, as operators were primarily using it for monitoring devices, adopting instead proprietary Command Line Interfaces for configuring them. However, the use of an interactive CLI compromise the ability of using script to automate tasks, because the output may be unpredictable or difficult to parse. NETCONF [4] is an attempt to provide an extensible XML-based management scheme on top of different transport protocols. The standard also defines asynchronous event notifications, a step forward from SNMP TRAPs. Moreover, traditional network management systems take a centralized approach by having the administrator monitor, control and configure devices in the network from one location with a server storing network information on a database and offering access via command line or some form of graphical interface. The server sends requests and receives responses from devices via a network management protocol (typically SNMP). Such a centralized management architecture is illustrated in Fig. 1(a). While a management system that follows this kind of centralized paradigm is easy to implement, it presents several problems which will be apparent from the discussion in the next Section.



(a) Traditional Centralized Management

(b) DIVINEMAN Distributed Management

Fig. 1: Network management paradigms.

## IV. Our Approach

We propose a novel network management system, termed DIVINEMAN, for broadband wireless access networks. DIVINEMAN addresses the limitations of existing systems via two salient features: (1) distributed management paradigm exploiting ever-expanding capabilities of embedded network devices; (2) a visual network process modeling approach to ease the specification of management tasks. The rest of this section elaborates on each of these features.

### A. Distributed Management

Rather than relying on a centralized management point, DIVINEMAN allows the administrator to connect to any network device to control the whole network. As shown in Fig. 1(b), there is no need for always-on servers. Instead a small agent installed on each device, referred to as IMP, allows the devices to communicate among themselves and to receive incoming management-related queries from the administrator. The IMP can be deployed either as a software process running within the device operating system, or as an add-on hardware device physically connected to the managed device in the case of proprietary devices that do not permit modifying kernel/firmware. Advantages of doing away with a centralized server are clear: there is no single point of failure anywhere in the network; no existing data center infrastructure is required; and the network management data gathered at various devices can leverage local storage and benefit from in-network processing, thus reducing the overhead incurred for network management purposes. Moreover, the notion of IMP agent enables management of heterogeneous networks.

### B. Network Process Modeling

DIVINEMAN system additionally eases the process of specifying management tasks for the network administrator via the Network Process Modeling (NPM) subsystem, a novel visual approach that allows him/her to focus on the overall management *goals* rather than on programming the individual *tasks*. Specifically, NPM provides a graphical environment in which the management task/activity to be performed on the network can be intuitively expressed by assembling elementary tasks and communication primitives. Our approach is inspired by Business Process Modeling Notation (BPMN) [5], the defacto standard for specifying business processes. With NPM, each activity takes the form of a “workflow” to be executed on one or more network devices automatically when a certain event happens (e.g., a new node connects) or when a certain condition is met (e.g., timer expires). For example, NPM workflow specification for firmware upgrade of network devices is shown in Fig. 2. Note that NPM also facilitates nodes to exchange messages containing notifications and metrics, and also aggregate received information or trigger specific procedures when a message is received. Elementary tasks constitute the building blocks for NPM workflows; they are implemented internally as packages that can be dynamically deployed, installed and updated on network devices regardless of their CPU architecture and hardware features. The complexity of communicating with different types and brands of devices is embedded inside each block, so that the administrator can ignore the specifics of the apparatus being controlled and instead concentrate on the management activity. When specifying a workflow in NPM, elementary tasks are presented as a library of blocks each implementing a common functionality (e.g., pinging a network device); this library can be expanded by incorporating additional blocks.

### C. Current Status

Our first step towards the development of the DIVINEMAN system has been to draft a formal definition of the underlying notation used in NPM and the subsequent implementation of a basic software interpreter that can parse and execute workflows specified using NPM. The outcomes of the research are continuously tested on our Tegola testbed [6], located in the northwest of Scotland providing broadband wireless access to some of the most remote communities in the UK; in future, we aim to involve a commercial WISP. We also plan to make our implementation public to build a user community. We are also investigating the use of model-checking systems to verify the correctness of NPM workflows, especially those corresponding to network management protocol tasks. Moving forward, we intend to explore automated network management, focusing on developing mechanisms for automated configuration of new devices; identifying solutions for automatic identification and resolution of failures; and building an efficient algorithm to manage global routing across the BWA network. Our findings on these issues will be incorporated into the next version of the DIVINEMAN system.

## References

[1] ITU/CCITT Recommendation M.3400. *TMN Management Functions.* October 1992.

[2] ITU-T Recommendation M.3010. *Principle for a Telecommunication Management Network.* May 1996.

[3] D. Harrington, R. Presuhn, B. Wijnen. *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks.* RFC 3411. December 2002.

[4] R. Enns. *NETCONF Configuration Protocol.* RFC 4741. December 2006.

[5] *Business Process Modelling Notation (BPMN)*, website: http://www.bpmn.org.

[6] *The Tegola Project*, website: http://www.tegola.org.uk.