Multi-service, Multi-protocol Management for Residential Gateways

Yvan Royon, Pierre Parrend, Stéphane Frénot, Serafeim Papastefanos, Humberto Abdelnur, Dirk van de Poel

Multi-service, Multi-protocol Management for Residential Gateways

Yvan Royon,
Pierre Parrend,
Stéphane Frénot
INRIA Ares / CITI, INSA-Lyon
F-69621, France
{first.last}@insa-lyon.fr

Serafeim Papastefanos
NTUA
Patission street 42
10682 Athens, Greece
serafeim@telecom.ntua.gr

Humberto Abdelnur,
INRIA Madynes / LORIA
F-54506, France
Humberto.Abdelnur@loria.fr

Dirk Van de Poel
Thomson
Prins Boudewijnlaan 47
B-2650 Edegem, Belgium
dirk.vandepoel@thomson.net

Abstract

When providing services to home users, management is a key activity. In-home devices, and especially the Residential Gateway, can use multiple management technologies for multiple management activities: read/write parameters, but also deploy, update, start and stop software components.

This paper defines management *realms* around the Residential Gateway, where different actors perform different management activities, using different technologies. We propose techniques that integrate these technologies (TR-069, UPnP, NetConf and JMX). We also address transient issues related to security.

Introduction

Service provisioning to the home is evolving, both on business and technical concerns. The growth is two-dimensional: first, new services emerge both inside and outside the home. Second, these services can be provided by new business players. In this context, the Residential Gateway (RG) holds a crucial role. It interconnects the LAN and the WAN worlds; it is the single equipment that links multiple service providers to home services.

In the context of MUSE project we studied the RG in a multi-provider context. This specific context brings new challenges we addressed and we focused on two levels of management for the RG: access management and service management.

The remainder of this paper presents the management challenges in a multi-provider environment; then we present results in two areas: the access management to the RG and an architecture to enable high service management.

Management challenges in a multi-provider environment

In the multi-play model [1], three management *realms* can be identified, defining who manages what around Residential Gateways. They identify the actors, the management activities, and the technologies involved. These realms are shown in figure 1.

- In the Access realm, the gateway operator manages access-related parameters on the Residential Gateway.
- In the Service realm, various service providers deploy, manage and operate services on the Residential Gateway or on in-home devices.
- In the User realm, home users set preferences on their in-home devices or on the Residential Gateway.

In this multi-management for multi-provider context we focused on three challenges. How can we cope with management protocol diversity, which service model can we use, and how can we handle security issues.

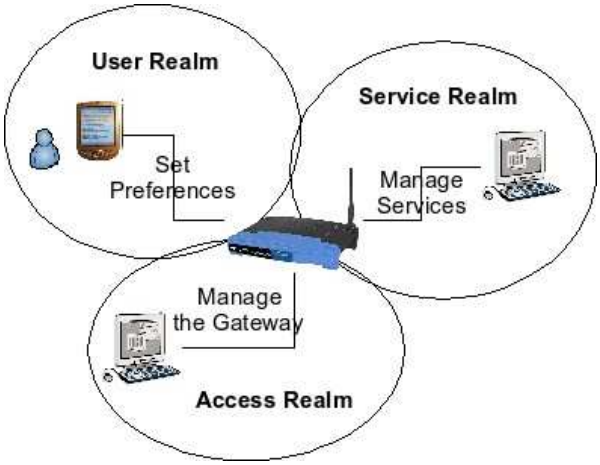

Illustration 1: Management Realms

Managing entities in a multi-* world leads to a proliferation of management protocols. This diversity in management technologies brings integration issues. For instance, local management protocols such as UPnP are secluded within the home network; if a manager on the WAN side needs to access a UPnP device, he needs to translate directives between UPnP and the manager's protocol of choice.

RG should host various services managed by various service providers. Those services are piece of applications that are triggered on a pay-per use schema. We studied a global architecture based on OSGi to manage those services. In this OSGi service provisionning environment we focused on the related security issues.

The next section details the studies we made about RG access management an service management architecture.

RG Management proposal

Access Management

The remote management of RG devices is based on the TR-069 document [2] published by the DSL forum. This specification presents CWMP (CPE WAN Management Protocol), which is used for the remote management of the RG and other home devices by a device called Auto Configuration Server (ACS), and the version 1.0 of the MIB - Management Information Base for the RG. The MIB is a collection of parameters that can be remotely managed by the ACS. This MIB is further elaborated in the TR-098 [3] document of the DSL Forum, which presents the version 1.1 of the RG MIB. Also, in the TR-106 [4] document the DSL Forum presents a template

for the MIMs of home devices. All home devices that support TR-069 should also be TR-106 compliant. A last issue concerns the mono-provider design of the TR-069 protocol: each device can be managed by only one ACS.

- Integrating TR-069 with UPnP

Many in-home equipments do not support TR-069 but instead use UPnP technology [5]. Because the proximity of UPnP is within the home network, these devices cannot be managed remotely with the current infrastructure. Example applications include the UPnP/AV protocol, which enables media servers (video and audio) to provide streams to media renderers (TV screens or computers). Many devices declare their behaviors through UPnP messages. We designed a UPnP Proxy for TR-069; it is a component that acts as an intermediate between the TR-069 and UPnP protocols and enables the ACS to discover, control and receive events from the UPnP devices even if the ACS is not in the local network of these devices. It is described in the MUSE public deliverable B3.4 [6]. The Proxy component is located in the home network and has a connection with the ACS.

An implementation of the Proxy as a software component in the Java programming language is available within the MUSE project. The requirement of this component is that the hosting device has a Java Runtime Environment installed. The general architecture of the Proxy can be seen in the following figure:

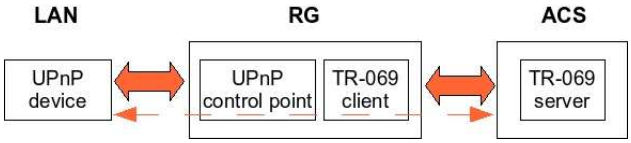

Illustration 2: Configuration of UpnP-enabled CPE via CWMP/UPnP bridge

- Integrating TR-069 with NetConf

TR-069 is not the only management protocol currently available. Another candidate is the NetConf [7] proposal from IETF. We propose a extension to NetConf that enables TR-069 RG management from a NetConf agent. This agent acts as a TR-069 proxy for the remote RG as shown in figure 3.

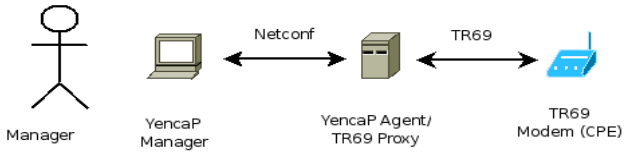

Illustration 3: Interaction with a TR69 based CPE through Netconf

- TR-069 extensions to multi-provider management

Finally some extensions to the TR-069 protocol have been made to cope with multi-provider environments. The first improvement deals with auto-provisioning the RG without a predetermined ACS. The ACS is discovered during the first boot of the device, and depending on subscription it is redirected to a specific ACS.

The second extension deals with PING diagnostics. The managing ACS can trigger a PING procedure handled by the RG. Since the RG can be managed by multiple service provider, the connection can be tangled and the PING procedure is useful to obtain some status.

The last extension to the TR-069 protocol enables TR-069 LAN devices. The use case is based on TR-111 [8] which considers two use cases: a direct connection from ACS to in-home equipments and the use of a NAT translation connection.

All these improvements over the TR-069 RG model enable a better access management of multi-provider environments. In the next section we assume that the connection is functional and we propose an architecture to manage high-level services in a multi-service provider context.

Service Management

OSGi is an emerging standard to manage high-level Java services for in-home environments. We designed an end-to-end architecture to handle a multi-provider and multi-services OSGi. The architecture provides lightweight isolation, JMX-based management and security extensions, and is demonstrated with a mock-up developed for the MUSE project.

OSGi-based Residential Gateways

OSGi [9] is the best contender to become the standard execution environment for Residential Gateways. Forums such as HGI base their requirements on the OSGi specifications. The basic idea is that software, packaged as deployment units called *bundles*, can be dynamically deployed on Residential Gateways running an OSGi framework. Bundles can be downloaded, installed, started, stopped and removed without rebooting the RG. This allows to update the firewall, device drivers, or any software part with close to zero downtime. We have extended OSGi in three domains: multi-provider, remote management through JMX [10] and secure deployment of bundles.

A Multi-provider OSGi Gateway

Each service provider runs a “virtual gateway” that runs the services he provides. All virtual gateways are handled by a “core gateway” that is handled by some access manager. The core gateway hosts common services and provides an access control to virtual gateways. More details on virtual and core gateways can be found in [11]. This virtualisation enables a co-location of services within the same gateway. Each service set is associated to a specific provider. In order to enable remote management of services, we provide a JMX management framework to OSGi.

Managing Virtual and Core OSGi gateways

MOSGi (for Managed OSGi) has been elaborated during MUSE phase 1, and is currently available in the Apache Felix [12] project. MOSGi runs a JMX management agent within the RG. This agent enables the remote management of Java applications. The MOSGi project also contains a dynamic management console whose presentation automatically adapts to the managed entity.

This means that the management console is dynamically built at run-time depending on the deployed services.

In the multi-provider environment each service provider can upload new services (as OSGi bundles). Security becomes an issue when potential competitors access and configure the same equipment. This lead to some security issues that need to be handled by the OSGi framework.

A Secure OSGi Gateway

The first problem that comes to mind is that download and installation must be verified. We need to identify the sender (the service provider who wants to deploy the OSGi bundle). We also need to check the integrity of the bundle, to ensure it has not been corrupted. This is achieved in three phases.

First, when the Residential Gateway boots, it loads a keystore, which contains public cryptographic keys for service providers. Each system bundle is then verified using this keystore: core system, bundles for remote management, etc. At this point, the OSGi platform is in a usable and verified state.

The second phase is when a service provider creates a bundle. The contents of the bundle is signed, using a `jarsigner` application. The bundle is then put in a repository.

The third phase is bundle deployment. When a Residential Gateway downloads a bundle from a repository, it validates the integrity of its contents against its hash checksum. If the verification fails, the bundle is rejected. We must alter the OSGi specifications so that life cycle management integrates validation, as is shown in figure 4.

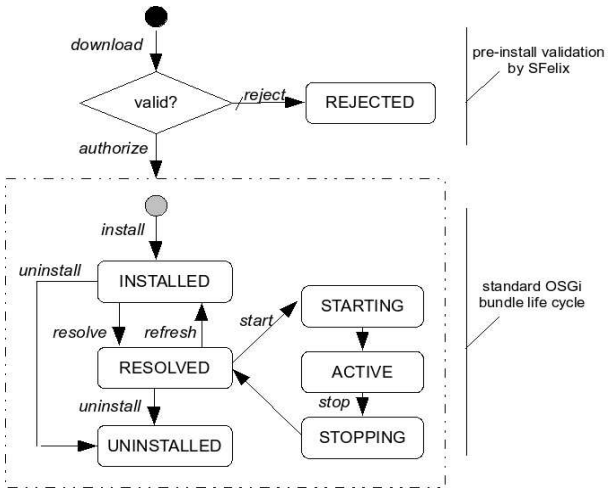

Illustration 4: Bundle life cycle with REJECTED state

Secure OSGi deployment is covered in details in [13].

Access management and service management activities have been integrated in a demonstration mock-up which shows asecure, end-to-end management of multi-service multi-provider environment.

An end-to-end multi-* management mock-up

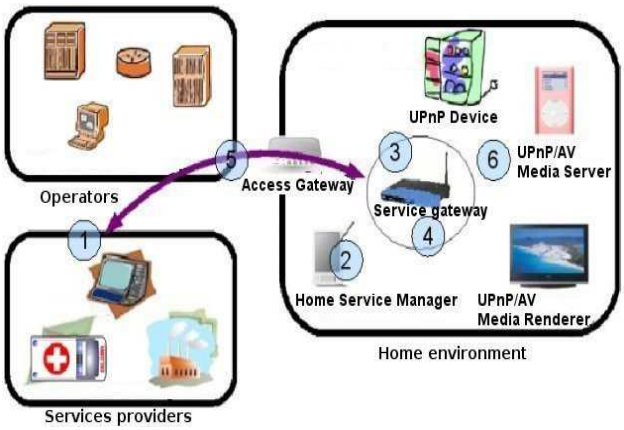

Illustration 5: end-to-end mock-up

Figure 5 shows the general management environment trial. It shows the various elements we provide in the mock-up. The Access Gateway (5) hosts the TR-069 enabled equipment, the Service Gateway (3) hosts the OSGi extended framework. The service provider environment (1) holds the remote management interfaces. The iPaq (2) holds a local management interface (user realm) and (6) represents various UPnP devices.

The provided use-case relies on two service providers. A fridge service provider proposes a fridge monitoring application, while the UPnP/AV proposes a video provisioning service. We show that both service providers host their services on isolated OSGi virtual gateways. They access in-home equipments (fridge and UPnP media server and renderer devices) through JMX management. Every device is a UPnP device that declares its presence to the RG. Each device has a local user interface that is dynamically uploaded on the iPaq. These local interfaces enable the local user to manage his equipments.

All service-related elements (1, 3, 2, 6) run under Gentoo Linux with the Felix OSGi implementation. We also show heterogeneous systems running the same middleware stack (OSGi/Felix/JMX management): it runs on top of i386/Gentoo Linux (management station), ARM/Gentoo Linux (NLSU2/Linksys), ARM/Debian Linux (iPaq), Via Nehemiah (1000Mhz)/ home-made Linux (service gateway).

Conclusion

The multi-service, multi-provider tendency leads to new kind of problems. During the MUSE project we focused on the impact this paradigm has on the Residential Gateway. We study these impacts at two levels: access management and service management. Access management is a short time issue that already impacts IP connection providers. The service management is a more long-term view of what the user may want in a near future. The permanent connection the xDSL world brings to our homes opens new possibilities for services, and the RG must be able to support it. Management-related proposals presented here do work on hardware that can be found today. However, if we want to

run several high-level OSGi services, RG with at least 64 MB of memory are recommended. Alternatively, using gaming consoles or high-end TV set-top boxes instead could be investigated.

Acknowledgments

This research is performed within the MUSE project that is partially funded by the European Commission as part of the European 6th Framework programme.

References

1. Alex De Smedt et al., "The multi-play service enabled Residential Gateway", *Broadband Europe* 2006, Geneva.
2. DSL Forum, "TR-069: WPE WAN Management Protocol," http://www.dslforum.org/techwork/tr/TR-069.pdf
3. DSL Forum, "TR-098: DSLHome Internet Gateway Device Version 1.1 Data Model for TR-069," http://www.dslforum.org/techwork/tr/TR-098.pdf
4. DSL Forum, "TR-106: DSLHome Data Model Template for TR-069-Enabled Devices," http://www.dslforum.org/techwork/tr/TR-106.pdf
5. UPnP Forum, standards and specifications, http://www.upnp.org/standardizeddcps/
6. MUSE Consortium, "Public deliverable DB3.4", http://www.ist-muse.eu/Abstracts/abstract_DB3.4.htm
7. IETF NETCONF WG, "NETCONF Configuration Protocol", http://www.ietf.org/rfc/rfc4741.txt
8. DSL Forum, "TR-111: Applying TR-069 to Remote Management of Home Networking Devices," http://www.dslforum.org/techwork/tr/TR-111.pdf
9. OSGi Alliance, "OSGi Service Platform Release 4", http://www.osgi.org/
7. Java Community Process, "Java Management Extensions", http://jcp.org/en/jsr/detail?id=003
8. Yvan Royon, Stéphane Frénot and Frédéric Le Mouël, "Virtualization of Service Gateways in Multi-provider Environments", *CBSE* 2006, Västerås.
9. Apache Software Foundation, "Felix OSGi implementation", http://felix.apache.org/
10. Pierre Parrend and Stéphane Frénot, "Supporting the Secure Deployment of OSGi Bundles", WoWMoM ADAMUS workshop, 2007, Helsinki.