# Outsourced Management of Home and SOHO Windows Desktops

Tiago Cruz, Paulo Simões,
João Rodrigues, Edmundo Monteiro
DEI-CISUC – University of Coimbra
Coimbra, Portugal

Fernando Bastos
Alexandre Laranjeira
PT Inovação
Aveiro, Portugal

**Abstract— In this paper we propose a solution that allows traditional desktop management technologies designed for the corporate LAN environment – namely the widely available Windows Management Instrumentation (WMI) – to operate in domestic, SOHO and small organization environments.**

**This solution is based on the integration of WMI with the Broadband Forum's CPE Wan Management Protocol (CWMP), an industry standard for remote management of CPE in commodity broadband access environments. The proposed approach is fully compliant with the CWMP standards, making it easy to integrate with existing CWMP platforms.**

**This integration extends the applicability of desktop management technologies to a wide range of Windows-based devices located in home networks, small offices and small organizations that currently lack proper management solutions. Consequently, it also opens the way for a number of novel application scenarios, which will also be discussed in this paper.**

**Keywords— Desktop Management, CWMP, WMI**

## I. INTRODUCTION

Large organizations always had to deal with the problem of managing hundreds or thousands of desktop PCs, each one with a Total Cost of Ownership (TCO) which tends to largely exceed its initial acquisition cost, when including maintenance and indirect costs. As a result, considerable amounts of money and effort have been invested in enterprise desktop management solutions, motivating industry standards such as the well-known Web-Based Enterprise Management initiative (WBEM [1]), promoted by the Desktop Management Task Force (DMTF [2]). The WBEM framework was developed for distributed enterprise management scenarios and has spawned several implementations, like the well known Microsoft's Windows Management Instrumentation (WMI) [3].

A different reality faces domestic and SOHO (Small Office, Home Office) users, which are required to directly manage their own PCs, despite frequently lacking the required technical expertise to do so. As opposed to corporate users, those users do not have proper tools or technologies at their disposal – in fact, the vast majority of desktop management standards focuses on enterprise LAN management paradigms, excluding domestic and SOHO environments or even small organizations served by commodity broadband Internet services, due to design and practical limitations.

Industry-led initiatives such as HGI [4] and the Broadband Forum [5] did produce a number of recommendations and technical standards for remote management of devices on broadband environments [6-8], such as the well-known CPE (Customer Premises Equipment) Wan Management Protocol suite (CWMP [9]), a *de facto* standard that currently covers a considerably large array of managed devices located inside the customer premises, such as home gateways, network equipment, set-top-boxes, VoIP devices, web terminals and all sorts of storage and media devices. However, the desktop is still not covered by CWMP – in fact, one could almost say that the PC is the last device standing out.

In this paper we propose precisely to bridge this gap between desktop management technologies – namely WMI, due to its wide installed base of Windows PCs – and the CWMP framework. For this purpose, we present an extension to the CWMP protocol that allows broadband operators to remotely access and manage Windows-based PCs, servers and appliances by using the WMI management API. This extension is fully compliant with the CWMP standard easily integrating with existing CWMP management infrastructures.

The potential applications leveraged by this integration are also discussed in this paper. Integration of WMI with the CWMP management framework for commodity Internet access obviously requires a number of adjustments in the traditional desktop management paradigm – which was focused on the corporate LAN. Several novel application scenarios are presented, introducing the concepts of outsourced and cooperative desktop management.

The paper is organized as follows. Section 2 and Section 3 present WMI and CWMP, respectively. The proposed CWMP-WMI integration framework is described in Section 4. The new application scenarios leveraged by these integration mechanisms are discussed in Section 5. Section 6 addresses validation and Section 7 discusses related work. Section 8 concludes the paper.

## II. WMI

WMI is an implementation of DTMF's WBEM standard for Microsoft Windows environments. WMI can be used to automate administrative tasks on remote computers – where it is also used by the operating system for internal management purposes. WMI uses the DMTF Common Information Model (CIM) to represent devices, applications and other managed components [10], federating information from several sources.

By defining a model of the operational aspects of a

Windows-based environment, WMI provides an extensible remote management interface for accessing information about aspects such as installed software/updates, drivers, devices or user profiles, among others.

WMI has two types of information providers: event providers (which generate notifications) and data providers (that deal with management data). In WMI terminology, data is structured in classes that contain properties. WMI Query Language (WQL) [11] – with a SQL-style syntax – can be used to generate and receive event notifications or to retrieve instances of class data and class definitions.

WMI is clearly oriented towards LAN-based environments, an option that reflects on its design. Until Windows OS versions 7 and Server 2003 R2 the WMI API was exposed (for direct access) through Distributed COM (DCOM) [12], a Windows-specific RPC mechanism not suitable for use outside the LAN environment. Later versions also support Windows Remote Management (WinRM) [13], a firewall-friendlier implementation of the WS-Management protocol based on SOAP. Still, both DCOM and WinRM have trouble operating on NAT environments, requiring the configuration of port mappings for inbound connections on the NAT gateway (such as, for instance, a home gateway), a solution which only works for one managed device at a time.

While its design limitations make it unsuitable for use in broadband environments, the resources provided by WMI are valuable in the context of operator-assisted remote management scenarios. To address these limitations, we propose to integrate WMI with the Broadband Forum's CWMP protocol, which will be presented in the next section.

## III. CWMP

CWMP is a remote management protocol that enables operators to remotely manage CPE devices on the customer premises, allowing for secure auto-configuration, service provisioning, diagnostics, software/firmware management and monitoring of such devices. It provides an API of Remote Procedure Calls (RPCs) based on SOAP web-services [14] with support for Secure Socket Layer (SSL) [15] or Transport Layer Security (TLS) [16]. Sessions are always initiated by the CPE, either directly or by request of the management server, designated as the Auto-Configuration Server (ACS).

CWMP is supported by a set of data models defined by related standards such as TR-098 [17] or TR-106 [18]. The standard data model for a CWMP-capable device follows a common set of requirements for which the structure, is hierarchically organized like a directory tree and has a single root object. In the case of a home gateway, for instance, the root object must be *InternetGatewayDevice* (as per TR-098).

As its name implies, CWMP is not a protocol targeted towards managing devices and services within LAN environments, like WMI. Instead, CWMP focuses on providing remote device management services for the ACS run by the operator. Since the two application fields are orthogonal to each other, this is not a problem by itself. Nevertheless, this isolation between WAN and LAN management mechanisms means that operators are often unable to take advantage of management services that are present at the LAN level (as is the case of WMI). As a result, the dynamic environment of the domestic LAN becomes invisible to them, representing the loss of valuable management information.

CWMP specifications have somewhat anticipated this problem by including data model extensions that allow the implementation of proxy management mechanisms. TR-106 provides support for home gateways to behave as management proxies for devices and services inside the domestic LAN, instantiating each one as a "Service" of the Home Gateway [18], which can be managed by the ACS in the same way as a service embedded on the gateway itself.

### A. CWMP Agent Extensibility (X-CWMP)

We have previously developed a dynamic agent extensibility framework [20] that, whilst keeping compliance with CWMP, decouples protocol services (concentrated in the so called "Master Agent") from the specialized component and device-specific management services to be provided via CWMP (distributed across "Subagents"). This framework is a CWMP agent development/extension toolkit which also eases its integration with other technologies such as WMI.

Master and subagents communicate using the specifically conceived X-CWMP [19] protocol, based on XML messages transported over TCP/IP, with optional use of SSL. Each Subagent is responsible for its own TR-106 extensions mapped and registered on the data model of the Master CWMP agent, which acts as a management proxy. Master Agents receive, convert and forward requests, while subagents deal with the processing of each request. Also, Master Agents and Subagents may be located in different devices.

X-CWMP subagents constitute a generic mechanism that allows the CWMP stack to mediate and abstract: managed services of the host device; managed services of other CPE devices; and management services provided by other protocols and associated data models (such as WMI), therefore operating as "protocol proxies".

In this Section we present two alternative approaches for CWMP-WMI integration. The first approach is completely generic and can be implemented using any CWMP agent development toolkit. The second approach takes direct advantage of the X-CWMP agent extensibility mechanisms discussed in the previous Section.

### B. CWMP-WMI Integration Architecture

As mentioned, the proposed CWMP-WMI extension framework allows two integration scenarios (Figure 1):

- The integration is performed at the home gateway. In this situation the specialized component responsible for the integration is a full part of the CWMP agent of the home gateway (as a "local CWMP/WMI subagent"). This option is natively supported by all Windows devices and can be implemented using a generic CWMP stack, as the WMI service is remotely accessed using DCOM or WinRM.

- The CWMP-WMI integration component is located in the managed Windows device, acting as an X-CWMP subagent. Access to WMI services is performed locally using DCOM or WinRM, therefore avoiding a number of restrictions typically found in WMI services (e.g. access control mechanisms), while also providing a performance advantage by dispensing the inefficient use of DCOM over the network (see Section VI). However, it does require X-CWMP and the installation of software in the Windows PC.

In either case the CWMP/WMI integration component is responsible for declaring which WMI attributes are mapped to the CWMP data model, for each Windows device instance.

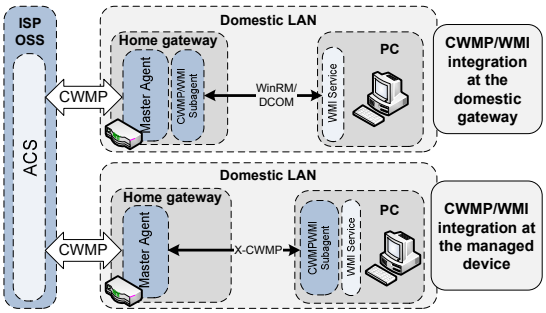


**Figure 1: CWMP/WMI integration alternatives.**

### C. *Integration of WMI on the CWMP Data model*

When the CWMP/WMI subagent registers on the master agent it becomes associated with the CWMP Data Model objects and parameters for which it is responsible. After a valid registration the subagent is ready to receive requests.

Information and properties of LAN devices managed through the CWMP/WMI agents are embedded on the home gateway CWMP data model through the use of dynamic TR-106 data model extensions. This way, the subagent maps WMI CIM namespace data into the CWMP data model.

Specifically, TR-106 allows an *InternetGatewayDevice* (home gateway) to act as management proxy for devices inside the subscriber LAN. Each proxied WMI device is modeled through a "*X_<OUI>_WMIService*" vendor-specific object instance – for which a default OUI (Organizational Unique Identifier) of 00000 was defined, for test purposes – which contains the correspondent data model. In this case, the ACS only communicates with the CWMP capable device, which incorporates the data models for the devices for which it is acting as a management proxy (Figure 2).

```
InternetGatewayDevice
      (..)
      Services
      X_000000_WMIServiceNumberOfEntries = 1
      X_000000_WMIService.1
          DeviceInfo
          (..)
      WMIServiceWQL
          Query
          Result
```

**Figure 2: CWMP TR-106 Data Model for Proxied WMI Agent**

Integration of the WMI data model into CWMP is performed using one of two methods:

- **Static CWMP-WMI mapping.** Specific CWMP parameters inside each *X_<OUI>_WMIService* instance are statically mapped into WMI class attributes. In this case the ACS user only manipulates conventional CWMP parameters (see Figure 3). Each mapping can be configured as write-protected or read-only, at the subagent level. Also, WMI classes (attribute containers) can be mapped as CWMP objects, for better structuring of mapped data.

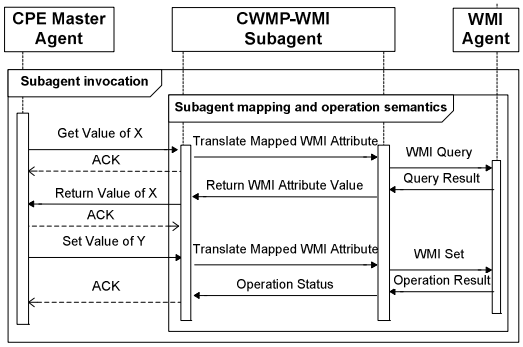


**Figure 3: Read and Write on Statically Mapped Attributes**

- **Dynamic queries.** Embedded WQL queries are supported by asynchronous calls, using the *X_<OUI>_WMIService.{i}.WMIServiceWQL* object and directly embedding WQL queries on its *Query* parameter whose result is stored on the *Result* parameter. The ACS still manipulates CWMP parameters, but its user needs to know WQL syntax and semantics to perform operations. The *Result* parameter is defined on the CWMP data model with the Forced Active Notification attribute enabled, allowing asynchronous operations using change notification events (see Figure 4).

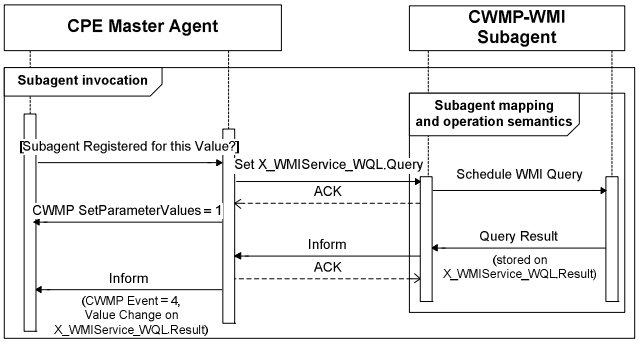


**Figure 4: Dynamic WQL Query Operations**

Static CWMP-WMI mappings are configured at the subagent level, with 3 different types of relationship semantics (depending on the nature of the mapped WMI attributes). If the mapped WMI attribute is static (e.g. OS version), it is queried and stored on an internal data structure, which is only updated at startup. If the value of the WMI attribute changes over time, its mapping can be configured as such:

- as a **direct-mapping attribute** which is queried/set by the subagent each time the corresponding CWMP parameter is requested or set. This is the default behavior.
- or as a **deferred-update attribute** stored on an internal data structure updated by a refresher thread. This behavior is independently configurable for Get and Set operations on each mapped WMI attribute. A parameter corresponding to a mapped attribute with deferred write properties will always be defined with CWMP Forced Active Notification

attributes, allowing asynchronous write operations using parameter change notification events (Figure 5).

The refresher thread used for deferred updates is meant to be dynamically scheduled, in order to make use of spare processing power on the host (Home Gateway or Windows Device), minimizing the overhead of the Subagent. Since operations on deferred-update attributes do not need to wait for the result of a WMI operation, they will take less time to execute, at the expense of reduced resolution on attribute data updates. This thread may also be integrated with WMI Events, enabling native event-driven update solutions.

CWMP notification change parameter attributes are supported on the CWMP-WMI subagent data model instances, either being preconfigured (using Forced Active Notification) or defined at runtime, using a CWMP *SetParameterAttributes* operation. Thus, it is possible to adjust static attribute mapping behavior in order to balance the tradeoff between fine-grained temporal resolution and subagent load penalty on the managed PC, while complying with CWMP operation semantics.

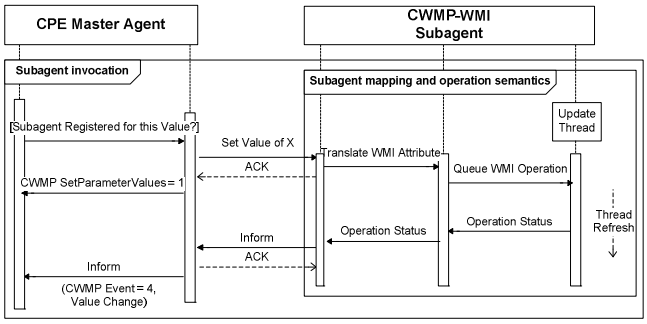


**Figure 5: Write Operations on Deferred-update Attributes**

## IV. Application Scenarios

As already mentioned, the proposed CWMP/WMI integration framework obviously requires a change in the traditional desktop management paradigm.

While in the corporate LAN the desktops and the management platform usually belong to the same entity (the corporation), here the internet service provider – that controls the ACS – plays an important role, either as the direct manager of the customers desktops or as a mediator for third party management services. These outsourced or cooperative models call for a trust relationship between involved parties. They also bring legal and ethical concerns, albeit not much different from privacy and security issues brought by popular cloud-based services such as Google Apps and Dropbox. In fact, typical users take more risks installing third-party software than they would by trusting the management of their desktops to providers they already know. In addition, those providers are already managing other devices *inside* the customer LAN, such as home gateways and set-top-boxes. As long as users can choose which devices are remotely managed and which management information is retrieved, this seems an acceptable compromise, when compared with other situations.

Next, we discuss two possible application scenarios that demonstrate how the CWMP-WMI extension can be used to provide a new class of added-value services related to remote management of desktop computers.

### A. Operator-assisted Management of Windows Devices

Internet Service Providers may sell to domestic or SOHO customers PCs bundled with software and remote management services (antivirus, software updates, remote recovery, etc.).

Using the CWMP-WMI Subagent, operators can remotely diagnose and solve common issues on Windows networks, such as domain/workgroup or network stack misconfiguration. Also, the Subagent can be used to enable remote registry scans to detect a wide range of issues, including Trojans and other security vulnerabilities. Information about the update level of the OS, its services, running processes and even access event logs can be fed to operator-level Intrusion Detection and Prevention Systems (IDS/IPS) to support active security policies and mechanisms [20].

### B. Desktop Management as a Service

While the CWMP management framework was designed with the needs of Internet service providers in mind, it is possible to extend its scope for other uses. Typically, broadband access provider Operations Support Systems (OSS) already integrate the CWMP ACS within their infrastructure, as a component that interfaces with asset management, billing or provisioning systems. In this line of though, an Internet provider may provide an interface for third-party service providers specialized in desktop management (Figure 6). Such a service would be attractive for small corporations, enabling them to outsource (or cloud-source) the remote management of their Windows desktops and servers.

The proposed usage scenario makes use of a middleware layer, which interfaces with the ACS (and possibly with other OSS services) and exposes an API designed to provide managed access to WMI agents on devices. Third-parties can use this API to offer a remote desktop management service with asset, inventory policy or software lifecycle management capabilities. User privacy is protected by granular non-repudiation and access control mechanisms.

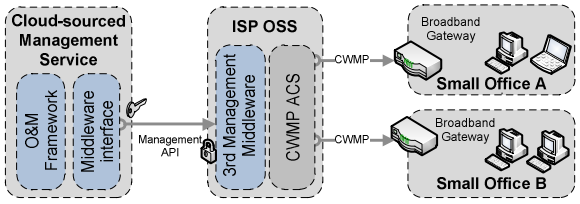


**Figure 6: Cloud-sourced Desktop Management**

## V. Validation

A proof-of-concept subagent was implemented in Java, using *J-Interop* [21] to access the WMI API through DCOM. WinRM could better fit our purposes, as it is an interoperability-focused protocol, but support is only available on more recent versions of the Windows OS family, whereas DCOM is supported since Windows 2000. As such, DCOM was used to maximize the span of supported OS versions.

### A. CWMP-WMI Validation Testbed

The implemented agent was evaluated in a testbed

emulating the ACS, the customer premises LAN and the service provider infrastructure. In order to mimic the conditions and restrictions imposed by the broadband access link, a transparent *Dummynet* bridge [22] interconnects the customer with the ISP segments. There also was a Linux-based access router with two network interfaces, supporting NAT - it has an embedded CWMP agent (the "Master Agent" for the CPE environment). A Windows PC acts as a WMI managed PC, with the CWMP-WMI Subagent installed.

### B. CWMP-WMI Validation Tests

Testing methodology has focused on latency protocol overhead. For this purpose, the test plan entails both static-mapping (with direct-mapped attributes) and dynamic queries.

The following static-mapping attributes were defined:

- **X_<OUI>_WMIService.{i}.WMIServiceDM.Uptime:** maps to the *SystemUpTime* WMI attribute of the *Win32_PerfFormattedData_PerfOS_System* class.
- **X_<OUI>_WMIService.{i}.WMIServiceDM.FreeMem:** maps to the *AvailableKBytes* WMI *attribute* of the *Win32_PerfFormattedData_PerfOS_Memory* class.
- **X_<OUI>_WMIService.{i}.WMIServiceDM.TotalMem:** maps to the *TotalPhysicalMemory* WMI attribute of the *Win32_ComputerSystem* class.

For dynamic queries, a specific "bulk" WQL query (*SELECT * FROM meta_class*) was chosen in order to assess CWMP-WMI performance for a larger payload (roughly 1MB).

All repetitive CWMP queries (averaged for 10 tests) were performed using both single/isolated (one operation performed on a single CWMP session) and pipelined operations (several operations performed on a single CWMP session).

A two-stage test plan was followed. First, reference results were obtained on a scenario where all equipments were connected using 100Mbit Ethernet (*Dummynet* bridge disabled), with the Linux router configured for packet forwarding only, instead of NAT. Latency data was obtained both for end-to-end WMI-native queries and for proxied CWMP-WMI access. Next, tests were performed to gather latency data for CWMP-WMI Subagent usage on broadband scenarios, with the *Dummynet* bridge being configured to enforce the network conditions of typical commercial offers (see Table I - configurations are discussed in [23]).

TABLE I. BROADBAND TEST REFERENCE SCENARIOS

| | Nominal bandwidth (bps) (Down/Up) | | Effective bandwidth (bps) (Down/Up) | | RTT Latency | Pkt. Loss |
|---|---|---|---|---|---|---|
| ADSL | 4M | 512K | 3.34M | 427.5K | 20ms | 0.1% |
| | 24M | 1M | 20.04M | 835K | 20ms | 0.1% |
| GPON | 20M | 2M | 18.6M | 1.86M | 5ms | 0% |
| | 100M | 10M | 93M | 9.3M | 5ms | 0% |
| LAN | 100M | 100M | 100M | 100M | <1ms | 0% |

Native WMI/DCOM tests were performed using a simple Java application which directly queried the WMI API using DCOM. The Subagent code was modified to record local WMI query execution times in order to calculate CWMP latency overhead (obtained by subtracting the local WMI query latency from the full CWMP operation latency).

### C. CWMP-WMI Validation Test Results (100Mbit LAN)

Figure 7 presents results for queries performed on the reference scenario (100Mbps LAN). Static, single-session operations perform significantly worse than native or pipelined session queries, because single-session queries are more affected by CWMP session establishment latency. This is partly due to the fact that an ACS cannot directly initiate a connection – instead, CWMP defines a request mechanism that instructs the CPE to initiate a session as soon as possible.

With pipelined operations the results are greatly improved, to the point where they surpass native WMI queries over the network. The reason for this has to do with two factors:

- Local DCOM/WMI queries, used by the X-CWMP Subagent, are faster than queries performed over the LAN.
- The CWMP RPC protocol is more efficient than DCOM in terms of operation latency (less network exchanges).

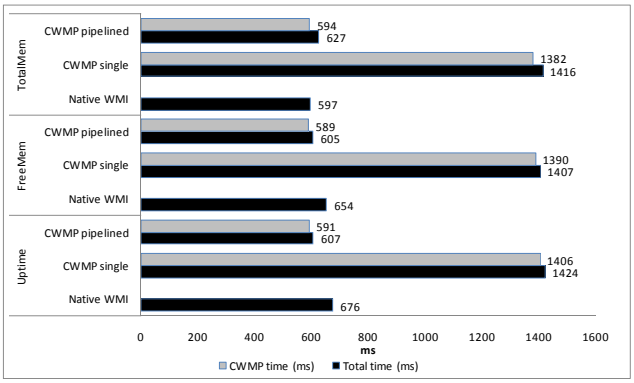


**Figure 7: WMI/DCOM vs. CWMP Latency for Static Attribute Queries (100Mbps LAN; values averaged for 10 experiments)**

For dynamic "bulk" queries, results show a similar situation. WMI queries are significantly faster when performed locally on the managed PC (Figure 8) by the X-CWMP Subagent, rather than over the LAN. This makes the CWMP latency overhead comparable with native WMI.

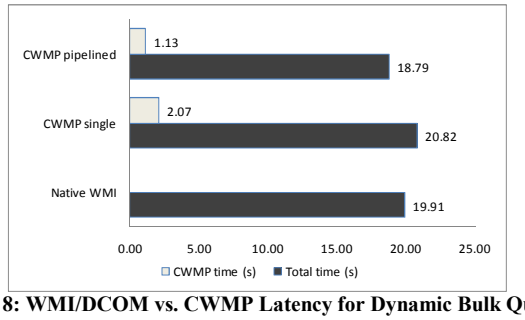


**Figure 8: WMI/DCOM vs. CWMP Latency for Dynamic Bulk Queries (100Mbps LAN; values averaged for 10 experiments)**

### D. CWMP-WMI Validation Test Results (Broadband)

For each of the defined CWMP-mapped WMI attributes (*Uptime*, *FreeMem*, *TotalMem*) a set of 10 queries was performed both for single and pipelined CWMP sessions, on different emulated broadband access network conditions.

Average results for total operation time and CWMP latency (Figure 9) show significant difference between single and pipelined operations. The average individual operation penalty is 1637ms for single-session and 641ms for pipelined

sessions, a difference explained by session setup overhead, which accounts for most of the difference between pipelined and single-session results. The influence of the access network technology is clearly higher on single-session CWMP operations – specifically in terms of latency, which penalizes session setup because of the higher number of protocol exchanges that are performed. Yet, for such small payloads (between 80 and 95 bytes), available upstream bandwidth has little or no influence.

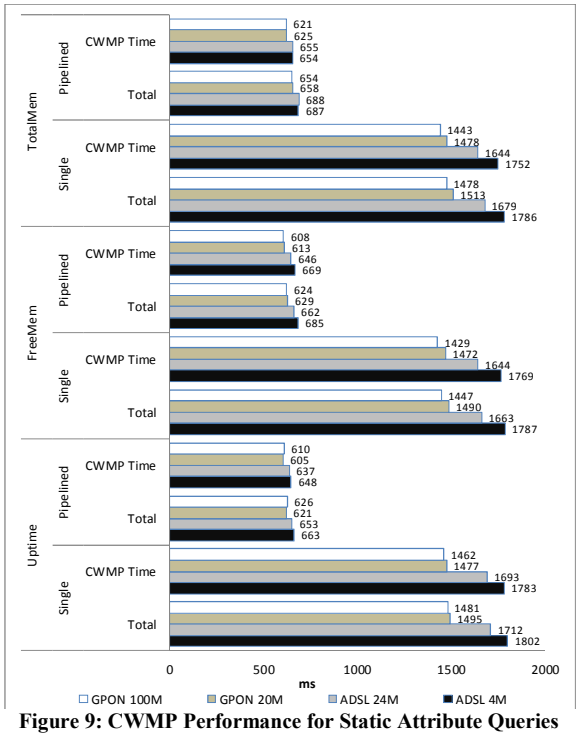


**Figure 9: CWMP Performance for Static Attribute Queries (values averaged for 10 experiments)**

For the dynamic "bulk" queries, the results show significant differences (Figure 10), due to payload encoding and marshalling. In this situation, available upstream bandwidth negatively impacts data transfer times. In comparison with the 100Mbit LAN reference results for native WMI/DCOM (19.9s – Figure 8), CWMP on 100Mbit GPON has an average operation latency of 21.6s, with a CWMP penalty of 2.9s for single-session operations and 19.4s with a 1.9s penalty for pipelined sessions.

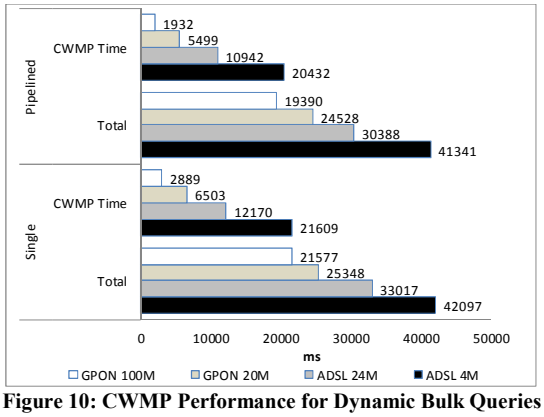


**Figure 10: CWMP Performance for Dynamic Bulk Queries (values averaged for 10 experiments)**

Overall, these results demonstrate the viability of integrating WMI operations over CWMP. Additionally, they could be further improved by adopting some of the encoding and traffic management techniques proposed by [24].

## VI. Related Work

The notion of protocol proxying based on CWMP-compliant gateways is not novel. The idea is latent in a number of specifications produced by the Broadband Forum, especially in the already mentioned TR-106, which provides proxying management support on a CPE data model. Also, recent Broadband Forum presentations [25] reveal that proxying extensions are being developed for CWMP, documented in PD-174 [26] (a "Proposed Draft", only available to members of the Broadband Forum). Predating the Broadband Forum efforts, [27] and [28] also proposed CWMP-UPnP protocol integration, using dedicated bridging mechanisms. More recently, Minokoshi et al., [29] proposed a plug-in based generic approach for LAN protocol proxying for UPnP, SNMP and LLDP using CWMP. To the best of our knowledge, our proposal is the first attempt to integrate CWMP with desktop management technologies such as WMI.

While being among the first known contributions for such a discussion [19], our CWMP extensibility framework distinguishes itself from other proposals by using a distributed approach. Instead of embedding the protocol proxying components exclusively on the home gateway, using dedicated bridging mechanisms or more generic plug-in capabilities, our framework make use of Subagents that can reside either on the home gateway or on the managed device. These subagents extend the Master agent data model at runtime, allowing it to integrate new capabilities with greater flexibility.

## VII. Conclusion

CWMP is steadily becoming the key standard for remote operator management of devices located in the local network of domestic and SOHO broadband users. Nevertheless, effective deployment of CWMP-based applications is rolling with some signs of success but still slower than desirable.

In this paper we introduced the concept of CWMP-WMI integration, proposing two alternative architectural approaches (bridging at the gateway or at the desktop level). Integration between both data models has also been addressed, including support for WMI attribute mapping and embedded dynamic queries, without disrupting CWMP operation semantics.

We also demonstrated the feasibility of integrating WMI operations within CWMP, enabling the creation of operator-assisted management services for Windows-based devices while overcoming the limitations of WMI operation over firewalls, NAT environments and broadband access networks.


## Acknowledgements

This research work was partially funded by Fundação para a Ciência e Tecnologia (FCT grant SFRH/BD/29118/2006) and by PT Inovação, in the context of the S3P Project.

REFERENCES


[1] Desktop Management Task Force, "Web-Based Enterprise Management", http://www.dmtf.org/standards/wbem

[2] Desktop Management Task Force (DMTF), www.dmtf.org

[3] Microsoft Corporation, "*Windows Management Instrumentation Remote Protocol Specification v10.1*", March 2010.

[4] Home Gateway Initiative (HGI), www.homegatewayinitiative.org

[5] Broadband Forum, http://www.broadband-forum.org

[6] Broadband Forum, "Functional Requirements for Broadband Residential Gateway Devices (TR-124) issue 1.0", 2006

[7] HGI, "Home Gateway Technical Requirements Residential Profile Version 1.0", April 2008.

[8] HGI, "Home Gateway Technical Requirements: Release 1", July 2006.

[9] Broadband Forum, "TR-069 - CPE WAN Management Protocol specification v1.1, Amendment 2", December 2007.

[10] DMTF, "Common Information Data Model (CIM) Infrastructure Specification v2.6.0", March 2010.

[11] Microsoft Corporation, "Querying with WQL", MSDN Library, 2008.

[12] Microsoft Corporation, " [MS-DCOM]: Distributed Component Object Model (DCOM) Remote Protocol Specification", MSDN Library, 2008

[13] Microsoft Corporation, "Windows Remote Management", 2009.

[14] W3C Consortium, "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)", www.w3.org, April 2007.

[15] A. Freier, P. Karlton, P. Kocher, "The SSL Protocol Version 3.0", Internet Draft, November 1996

[16] T. Dierks, C. Allen, "The TLS Protocol, Version 1.0", RFC 2246, January 1999

[17] Broadband Forum, "Internet Gateway Device Data Model for TR-069, TR-098 Amendment 1", December 2006.

[18] Broadband Forum, "Data Model Template for TR-069 Enabled Device, TR-106 Amendment 1", December 2006.

[19] T. Cruz et al., "CWMP Extensions for Enhanced Management of Domestic Network Services", Proc. of LCN'2010 (The 35th IEEE Conf. on Local Computer Networks), Denver, USA, September 2010.

[20] T. Cruz et al., "How to Cooperatively Improve Broadband Security", Proc. of the 9th ECIW (European Conference on Information Warfare and Security), Lisbon, Portugal, July 2009

[21] Dimentrix Technologies, "j-Interop: Pure Java – DCOM Bridge", http://www.j-interop.org

[22] M. Carbone, L. Rizzo, "Dummynet revisited", SIGCOMM CCR, Vol. 40, No. 2, November 2009.

[23] T. Cruz et al., "Integration of PXE-based Desktop Solutions into Broadband Access Networks", Proc. of CNSM'2010 (The 6th IEEE/IFIP Conf. on Network and Services Management),Niagara Falls, Canada, October 2010.

[24] A. Nikolaidis et al., "Management Traffic in Emerging Remote Configuration Mechanisms for Residential Gateways and Home Devices," IEEE Commun. Mag.,vol. 43, no. 5, May 2005.

[25] Broadband Forum, "About BroadbandHome", http://www.broadband-forum.org/downloads/BBHome_Remote_Mgmt.pdf

[26] Broadband Forum, "PD-174: Remote Management of Non TR-069 Devices", http://www.broadband-forum.org/technical/technicalwip.php

[27] A. Delphinanto et al., "Remote discovery and management of end-user devices in heterogeneous private networks", Proc. of the 6th Annual IEEE Consumer Communications and Networking Conference (CCNC 2009), Las Vegas, USA, January 2009.

[28] A.Nikolaidis et al. "Local and Remote Management Integration for Flexible Service Provisioning to the Home", IEEE Communications Magazine, pp. 130-138, October 2007.

[29] R. Minokoshi et al, "A Study on CWMP (TR-069) Proxy with Home Network Protocols", Proc. of IEICE Technical Committee on Information and Communication Management (ICM) 2010, Hokaido, Japan, July 2010.