ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΠΟΜΑΚΡΥΣΜΕΝΗ ΔΙΑΧΕΙΡΙΣΗ ΤΟΥ IP ΣΥΝΔΡΟΜΗΤΙΚΟΥ ΕΞΟΠΛΙΣΜΟΥ ΜΕ ΧΡΗΣΗ XML ΤΕΧΝΟΛΟΓΙΩΝ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

ΑΠΟΣΤΟΛΟΥ Ε. ΝΙΚΟΛΑΪΔΗ
Διπλωματούχου Ηλεκτρολόγου Μηχανικού & Μηχανικού Υπολογιστών του Δημοκρίτειου Πανεπιστημίου Θράκης

Αθήνα, Οκτώβριος 2006

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΠΟΜΑΚΡΥΣΜΕΝΗ ΔΙΑΧΕΙΡΙΣΗ ΤΟΥ IP ΣΥΝΔΡΟΜΗΤΙΚΟΥ ΕΞΟΠΛΙΣΜΟΥ ΜΕ ΧΡΗΣΗ XML ΤΕΧΝΟΛΟΓΙΩΝ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

ΑΠΟΣΤΟΛΟΥ Ε. ΝΙΚΟΛΑΪΔΗ
Διπλωματούχου Ηλεκτρολόγου Μηχανικού & Μηχανικού Υπολογιστών του Δημοκρίτειου Πανεπιστημίου Θράκης

Συμβουλευτική Επιτροπή : Γεώργιος Στασινόπουλος
Μιχαήλ Θεολόγου
Βασίλειος Λούμος

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την 24η Οκτωβρίου 2006.

.....	.....	.....
Γ. Στασινόπουλος	Μ. Θεολόγου	Β. Λούμος
Καθηγητής ΕΜΠ	Καθηγητής ΕΜΠ	Καθηγητής ΕΜΠ
.....	.....	.....
Ε. Συκάς	Ν. Μήτρου	Ε. Καγιάφας
Καθηγητής ΕΜΠ	Καθηγητής ΕΜΠ.	Καθηγητής ΕΜΠ

.....
Δ. Ρεΐσης
Αν. Καθηγητής Παν. Αθηνών

Αθήνα, Οκτώβριος 2006

.....

ΑΠΟΣΤΟΛΟΣ Ε. ΝΙΚΟΛΑΪΔΗΣ

Διδάκτωρ Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © **ΑΠΟΣΤΟΛΟΣ Ε. ΝΙΚΟΛΑΪΔΗΣ**, **2006**.
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Αφιερώνεται στους γονείς μου, Ευάγγελο και Ιφιγένεια

ΠΡΟΛΟΓΟΣ

Αισθάνομαι μεγάλη χαρά που κατάφερα να ολοκληρώσω την εκπόνηση της διδακτορικής μου διατριβής μετά από μακρά προσπάθεια τεσσάρων ετών. Σίγουρα ήταν ένα δύσκολο αλλά και ευχάριστο εγχείρημα ταυτόχρονα, αφού η έστω και ελάχιστη συμβολή μου σε αυτή την ιδιαίτερη επιστήμη του μηχανικού ήταν ένα όνειρο που είχα από τα πρώτα φοιτητικά μου χρόνια.

Τα δίκτυα δεδομένων και η πληροφορική έχουν ήδη γίνει ένα κομμάτι της καθημερινότητάς μας. Μάλιστα η σύγκλισή τους με τις τηλεπικοινωνίες αποτελεί μεγάλη πρόκληση για εμάς τους μηχανικούς αλλά συνάμα συνιστά και την ελπίδα του σύγχρονου ανθρώπου για μια ποιοτικότερη ζωή. Παρόλο που η σύγκλιση αυτή προχωρά με γοργούς ρυθμούς, ακόμη δεν έχει αποδώσει τους αναμενόμενους καρπούς. Μια πενταετία μετά την ευρεία εφαρμογή τεχνολογιών πρόσβασης (όπως το xDSL), που εκτινάσσουν το διαθέσιμο εύρος ζώνης οι περισσότεροι συνδρομητές εξακολουθούν να χρησιμοποιούν το διαδίκτυο κυρίως για εξερεύνηση ιστοσελίδων και ηλεκτρονικό ταχυδρομείο. Από την άλλη πλευρά οι συνδρομητές αγοράζουν μανιωδώς ηλεκτρονικές συσκευές, πολλές εκ των οποίων υποστηρίζουν δικτυακές τεχνολογίες και επομένως μπορούν να έχουν πρόσβαση στο διαδίκτυο. Αλήθεια πόσες καινοτόμες και χρήσιμες υπηρεσίες θα μπορούσαν να προσφερθούν στους συνδρομητές μέσω του διαδικτύου και των συσκευών αυτών; Πολλές, είναι η απάντηση των χειριστών του δικτύου και των παρόχων, οι οποίοι ήδη έχουν κάνει μεγάλες επενδύσεις σε αυτή την επιχειρηματική ευκαιρία. Γιατί όμως καθυστερεί τόσο η εφαρμογή και διάδοση τέτοιων υπηρεσιών; Ίσως ένας από τους βασικότερους λόγους είναι η αδυναμία για αυτόματη ρύθμιση και απομακρυσμένη διαχείριση των συσκευών και των εφαρμογών, που αποδίδουν τις υπηρεσίες στο συνδρομητή. Το γεγονός αυτό με ώθησε να ερευνήσω σε βάθος το πεδίο της διαχείρισης του συνδρομητικού εξοπλισμού, με το οποίο και ασχολείται η παρούσα εργασία.

Η ολοκλήρωση της διατριβής μου κατέστη δυνατή χάρη στη συμπαράσταση και τη βοήθεια ενός μεγάλου αριθμού ανθρώπων. Όλοι αυτοί αλλά και ο καθένας τους ξεχωριστά συνέβαλαν τα μέγιστα για την επιτυχημένη ολοκλήρωση της ερευνητικής μου δραστηριότητας. Στο σημείο αυτό, θα ήθελα να τους ευχαριστήσω προσωπικά για την ηθική και πνευματική τους υποστήριξη και να τους εκφράσω την ευγνωμοσύνη μου για τη βοήθειά τους.

Πρώτον απ' όλους θα ήθελα να ευχαριστήσω τον επιβλέποντά μου κ. Γεώργιο Στασινόπουλο, Καθηγητή του ΕΜΠ, ο οποίος μου έδειξε εμπιστοσύνη, με συμβούλεψε με την πείρα του και τις γνώσεις του και με στήριξε καθ' όλη τη διάρκεια της ερευνητικής μου προσπάθειας, από τη στιγμή που έγινα δεκτός ως Υποψήφιος Διδάκτορας μέχρι και σήμερα. Τον ευχαριστώ θερμά για τη συνεργασία που είχαμε τόσο σε επίπεδο επιστημονικό όσο και σε επίπεδο ανθρώπινων σχέσεων. Μαζί με τον κ. Γ. Στασινόπουλο, θα ήθελα να εκφράσω τις ειλικρινείς ευχαριστίες μου για τις πολύτιμες συμβουλές τους προς τα υπόλοιπα μέλη της συμβουλευτικής επιτροπής, τον κ. Μ. Θεολόγου, Καθηγητή ΕΜΠ, και τον κ. Β. Λούμο, Καθηγητή ΕΜΠ.

Ιδιαίτερη αναφορά αξίζουν οι συνάδελφοί μου στα Εργαστήρια Τηλεπικοινωνιών και Δικτύων Υπολογιστών, οι οποίοι αποδείχθηκαν για εμένα πολύτιμοι συνεργάτες και φίλοι. Μαζί τους έζησα ξεχωριστές στιγμές και οφείλω να ομολογήσω ότι με βοήθησαν σε μεγάλο βαθμό να ανταπεξέλθω στις απαιτήσεις αυτής της δοκιμασίας. Θα ήθελα λοιπόν να ευχαριστήσω θερμά τους, Γρηγόριο Δουμένη, Μάριο Δράκο, Φώτιο Ανδριτσόπουλο, Σεραφείμ Παπαστέφανο, Γιάννη Μητσό, Γεώργιο Θάνο και Κώστα Παπακωνσταντίνου, με τους οποίους συνεργάστηκα στενά όλα αυτά τα χρόνια σε πολλά ερευνητικά προγράμματα και σε πλήθος δημοσιεύσεων και εργασιών, από τις οποίες προήλθε η παρούσα διατριβή. Παράλληλα, δεν θα μπορούσα να παραλείψω το συνάδελφο, φίλο και συγκάτοικο Φώτιο Δραγανίδη, στέλεχος της εταιρίας Microsoft Hellas, ο οποίος όλο αυτόν τον καιρό με στήριξε ηθικά και πνευματικά με τις πολύτιμες συμβουλές του αλλά και με το πρόσχαρο και φιλικό περιβάλλον, που πάντα καλλιεργούσε.

Ολοκληρώνοντας αυτές τις γραμμές, θα ήθελα να εκφράσω την αμέριστη αγάπη μου στην οικογένειά μου και τα οικεία μου πρόσωπα, τα οποία υπήρξαν συνοδοιπόροι της όλης μου προσπάθειας και μου προσέφεραν απλόχερα τα απαραίτητα ψυχικά, συναισθηματικά και υλικά εφόδια για να ανταπεξέλθω στις υποχρεώσεις μου. Γι' αυτό, θα ήθελα να αφιερώσω την παρούσα διατριβή στον πατέρα μου Ευάγγελο και τη μητέρα μου Ιφιγένεια και να τους ευχαριστήσω από βάθος καρδιάς για τη συμπαράστασή τους στον δύσκολο αγώνα μου.

Απόστολος Ε. Νικολαΐδης

Αθήνα, 24 Οκτωβρίου 2006

ΠΕΡΙΛΗΨΗ

Το ευρύτερο πεδίο ενδιαφέροντος της διατριβής σχετίζεται με την εφαρμογή και αξιοποίηση των XML (eXtensible Markup Language) τεχνολογιών στην απομακρυσμένη διαχείριση των IP συνδρομητικών συσκευών αλλά και των υπηρεσιών, που προσφέρονται διαδικτυακά μέσω του εξοπλισμού αυτού από τους παρόχους. Αντικειμενικός στόχος είναι η υποστήριξη του SOAP/XML/HTTP υποδείγματος και η συμβολή στην ταχεία προώθηση των τεχνολογιών διαχείρισης που το υιοθετούν. Αρχικά προβάλλεται η ολοένα και επιτακτικότερη ανάγκη των τελευταίων ετών για μαζική διαχείριση των IP συνδρομητικών συσκευών. Κατόπιν παρουσιάζονται και αξιολογούνται οι υποψήφιες τεχνολογίες για αυτού του είδους τη διαχείριση. Μέσα από την ανάλυση διακρίνονται οι λύσεις, που βασίζονται στις τεχνολογίες ιστού και πιο συγκεκριμένα εκείνες, που χρησιμοποιούν το συνδυασμό των SOAP, XML και HTTP προτύπων της W3C. Το υπόλοιπο τμήμα της εργασίας χωρίζεται σε τρεις βασικές ενότητες, καθεμία εκ των οποίων επιχειρεί από διαφορετική σκοπιά να υποστηρίξει τον αντικειμενικό στόχο της διατριβής.

Στην πρώτη ενότητα διερευνάται η επίδραση της εφαρμογής του SOAP/XML στις επιδόσεις της διαχείρισης και κατόπιν προτείνονται δύο συμπληρωματικές προσεγγίσεις για τη βελτίωση των επιδόσεων αυτών. Η πρώτη συνίσταται στην επιλεκτική συμπίεση σε πραγματικό χρόνο των SOAP φακέλων, που μεταφέρουν τη διαχειριστική πληροφορία. Η δεύτερη προτείνει την αποφυγή της κλασσικής μεθόδου DOM για την ανάλυση των SOAP/XML μηνυμάτων και αντί αυτής υιοθετεί τους ειδικούς γραμματικούς αναλυτές. Η αξία της κάθε λύσης τεκμηριώνεται με θεωρητικούς υπολογισμούς αλλά και με πειραματικά αποτελέσματα.

Η δεύτερη ενότητα εστιάζει στην ανάπτυξη μεθοδολογιών για τη μείωση του κόστους ανάπτυξης, δοκιμών και δια βίου συντήρησης των πρακτόρων, οι οποίοι συμμορφώνονται με τα νέα SOAP/XML/HTTP πρότυπα πρωτόκολλα απομακρυσμένης διαχείρισης. Για το πρόβλημα αυτό προτείνεται μια ολοκληρωμένη μεθοδολογία, που βασίζεται στην αναπαράσταση του μοντέλου διαχείρισης με τη βοήθεια της γλώσσας XML. Μάλιστα για την αποτίμηση της προσέγγισης αυτής αναπτύσσεται μια σουίτα λογισμικού, η οποία παράγει αυτόματα και δοκιμάζει τον πηγαίο κώδικα πρακτόρων, που συμμορφώνονται με το νέο πρωτόκολλο απομακρυσμένης διαχείρισης συνδρομητικών συσκευών, CWMP.

Η τρίτη και τελευταία ενότητα προτείνει την ιδέα της γεφύρωσης των νέων προτύπων απομακρυσμένης διαχείρισης με ήδη επιτυχημένα και ευρέως υποστηριζόμενα πρότυπα τοπικής διαχείρισης. Στόχος εδώ είναι η επέκταση της διαλειτουργικότητας των νέων SOAP/XML/HTTP προτύπων απομακρυσμένης διαχείρισης. Μάλιστα για να αποτιμηθεί η εφαρμοσιμότητα της προτεινόμενης προσέγγισης σχεδιάζεται και υλοποιείται ένας μηχανισμός γεφύρωσης του νέου πρωτοκόλλου απομακρυσμένης διαχείρισης, CWMP με το ευρέως εφαρμοσμένο πρωτόκολλο τοπικής διαχείρισης, UPnP.

Τέλος παρατίθενται τα βασικά συμπεράσματα της μελέτης, επισημαίνονται οι τρόποι αξιοποίησής τους και συζητούνται ενδιαφέρουσες προεκτάσεις.

ABSTRACT

This thesis deals with the deployment of XML-based (eXtensible Markup Language) technologies for the remote management of IP-based Customer Premises Equipment (CPE). The main objective is to support the application of SOAP/XML/HTTP paradigm in the remote management field and to contribute to the fast development and wide adoption of related technologies and standards from the management market. Initially, the dissertation presents the increasing need for mass configuration and monitoring of CPEs. Then the most important candidate technologies for this management environment are analyzed and evaluated. The analysis indicates that the combination of SOAP, XML and HTTP standards presents many benefits, and thus the SOAP/XML/HTTP paradigm constitutes a promising solution for the remote management of next decade. However, the actual contribution of this work is analyzed in the next three sections, each of which attempts to support the main objective from a different perspective.

The first section investigates the effect of SOAP/XML in the performance of remote management in terms of configuration/monitoring time, bandwidth usage and computational requirements from the managed devices. Then, two complementary approaches are proposed. The first proposes the application of selective compression to the SOAP/XML envelopes, while the second approach proposes to avoid the classical and standardized DOM method for the parsing of SOAP/XML envelopes and to develop management protocol specific parsers. The value of each solution is validated by theoretical calculations and experimental results.

The second section focuses on methodologies for the cost reduction during the procedures of development, testing and life-cycle maintenance of the device agents' software. These agents are conformed to the new SOAP/XML/HTTP remote management standards. To tackle this problem, an integrated solution based on the XML representation of management information models is proposed. Also, to evaluate the proposed solution a software suite was developed. This tool generates automatically and tests the source code of management application for the CWMP (CPE WAN Management Protocol) device agents.

The last section proposes to bridge the new remote management standards with the successful and already established local management standards. The objective here is to extend the interoperability of SOAP/XML/HTTP remote management standards. To evaluate the applicability of the proposed approach, a corresponding mechanism is designed and implemented. This mechanism bridges the new CWMP protocol for remote management with the already established and widely-adopted UPnP protocol, for local management.

Finally, the conclusions coming from the thorough study of the above problems are presented, giving us useful insight to their potential exploitation, while at the same time bringing forward interesting extensions.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ 1

1.1. ΥΠΟΒΑΘΡΟ 1
1.2. ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΔΙΑΤΡΙΒΗΣ 5
1.3. ΔΟΜΗ ΤΗΣ ΔΙΑΤΡΙΒΗΣ 7
1.4. ΑΝΑΦΟΡΕΣ 8

ΚΕΦΑΛΑΙΟ 2 ΟΙ ΣΥΓΧΡΟΝΕΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΧΕΙΡΙΣΗΣ 10

2.1. ΕΙΣΑΓΩΓΗ 10
2.2. ΤΑ ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΟΥ SNMP ΠΡΟΤΥΠΟΥ 12
2.3. ΤΑ ΠΡΟΒΛΗΜΑΤΑ ΤΟΥ SNMP ΠΡΟΤΥΠΟΥ 12
2.3.1. Η αδυναμία της SNMP μοντελοποίησης 13
2.3.2. Η χρήση του αναξιόπιστου UDP πρωτοκόλλου 13
2.3.3. Η απουσία της ολοκληρωμένης διαχείρισης 14
2.3.4. Η ασφάλεια 15
2.4. ΟΙ ΥΠΟΨΗΦΙΕΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΧΕΙΡΙΣΗΣ 17
2.4.1. Η τεχνολογία OSI/TMN 17
2.4.2. Η εφαρμογή της CORBA στη διαχείριση 18
2.4.3. Οι τεχνολογίες κινητού κώδικα στη διαχείριση 19
2.4.4. Οι τεχνολογίες έξυπνων πρακτόρων στη διαχείριση 21
2.4.5. Η Java RMI και η πύλη ανοιχτών υπηρεσιών OSGi 21
2.4.6. Διαχείριση με χρήση τεχνολογιών ιστού 23
2.5. Η XML ΚΑΙ ΤΑ ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΗΣ ΣΤΗ ΔΙΑΧΕΙΡΙΣΗ 32
2.5.1. Μια σύντομη ανασκόπηση της XML 32
2.5.2. Τα πλεονεκτήματα της XML 33
2.6. ΑΝΑΚΕΦΑΛΑΙΩΣΗ - ΣΥΜΠΕΡΑΣΜΑΤΑ 34
2.7. ΒΙΒΛΙΟΓΡΑΦΙΑ - ΑΝΑΦΟΡΕΣ 37

ΚΕΦΑΛΑΙΟ 3 ΑΠΟΤΙΜΗΣΗ ΚΑΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΤΩΝ ΕΠΙΔΟΣΕΩΝ ΤΩΝ SOAP/XML/HTTP ΤΕΧΝΟΛΟΓΙΩΝ ΔΙΑΧΕΙΡΙΣΗΣ 43

3.1. ΕΙΣΑΓΩΓΗ 43
3.2. ΚΡΙΤΗΡΙΑ ΕΠΙΔΟΣΕΩΝ 44
3.2.1. Όγκος διαχειριστικής κίνησης 44
3.2.2. Χρόνος τέλεσης διαχειριστικής ενέργειας 44
3.2.3. Απαιτούμενοι πόροι επεξεργασίας και αποθήκευσης 45
3.3. ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ 46
3.3.1. Το μοντέλο διαχειριστή-πράκτορα 46
3.3.2. Η μοντελοποίηση της διαδικασίας διαχείρισης 47
3.4. ΣΧΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ 54
3.4.1. Εργασίες με χρήση SNMP – XML πυλών 55
3.4.2. Εργασίες χωρίς τη χρήση πυλών 62
3.4.3. Κοινά συμπεράσματα και αδυναμίες των σχετικών εργασιών 70

3.5. Η ΕΡΓΑΣΙΑ ΤΗΣ ΔΙΑΤΡΙΒΗΣ 71
3.5.1. Το περιβάλλον διαχείρισης 72
3.5.2. Ο όγκος της κίνησης διαχείρισης 75
3.5.3. Οι χρόνοι μετάδοσης των μηνυμάτων διαχείρισης 80
3.5.4. Η ιδέα της επιλεκτικής συμπίεσης 82
3.5.5. Η επιλογή της μεθόδου συμπίεσης 84
3.5.6. Το κέρδος της συμπίεσης στη διαχειριστική κίνηση 88
3.5.7. Το κέρδος της συμπίεσης στο χρόνο μετάδοσης 89
3.5.8. Οι συνθήκες για την εφαρμογή της συμπίεσης 91
3.5.9. Περιορίζοντας το χρόνο κωδικοποίησης 94
3.5.10. Περιορίζοντας τις απαιτήσεις σε μνήμη 97
3.5.11. Συνολική αποτίμηση των προτεινόμενων λύσεων 98
3.6. ΣΥΜΠΕΡΑΣΜΑΤΑ – ΠΑΡΑΤΗΡΗΣΕΙΣ 105
3.7. ΑΝΑΦΟΡΕΣ 107

ΚΕΦΑΛΑΙΟ 4 ΠΕΡΙΟΡΙΣΜΟΣ ΤΟΥ ΚΟΣΤΟΥΣ ΤΩΝ SOAP/XML/HTTP ΠΡΟΤΥΠΩΝ ΑΠΟΜΑΚΡΥΣΜΕΝΗΣ ΔΙΑΧΕΙΡΙΣΗΣ 111

4.1. ΕΙΣΑΓΩΓΗ 111
4.2. ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΔΙΑΧΕΙΡΙΣΗΣ ΚΑΙ ΟΙ ΕΡΓΑΣΙΕΣ ΠΡΟΤΥΠΟΠΟΙΗΣΗΣ 113
4.3. Η ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΤΩΝ ΣΥΣΚΕΥΩΝ 115
4.4. ΑΥΤΟΜΑΤΟΠΟΙΩΝΤΑΣ ΤΗΝ ΑΝΑΠΤΥΞΗ ΚΑΙ ΤΗ ΣΥΝΤΗΡΗΣΗ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ ΤΩΝ ΠΡΑΚΤΟΡΩΝ ΔΙΑΧΕΙΡΙΣΗΣ 118
4.4.1. Εργαλείο περιγραφής συσκευών 120
4.4.2. Εργαλείο δημιουργίας συσκευών 122
4.5. Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΠΑΡΑΜΕΤΡΙΚΟΥ ΠΡΑΚΤΟΡΑ 126
4.5.1. Η βάση διαχείρισης εκφρασμένη σε XML 126
4.5.2. Οι υπόλοιπες λειτουργίες 129
4.6. ΑΥΤΟΜΑΤΟΠΟΙΩΝΤΑΣ ΠΑΡΑΠΛΕΥΡΕΣ ΔΙΑΔΙΚΑΣΙΕΣ 132
4.6.1. Η αυτοματοποίηση των δοκιμών στον πράκτορα 132
4.6.2. Η συντήρηση λογισμικού του πράκτορα 135
4.6.3. Η αυτοματοποίηση της διαδικασίας ρύθμισης των συσκευών 136
4.7. ΑΝΑΚΕΦΑΛΑΙΩΣΗ ΤΩΝ ΠΡΟΤΕΙΝΟΜΕΝΩΝ ΔΙΑΔΙΚΑΣΙΩΝ ΚΑΙ ΑΠΟΤΙΜΗΣΗ ΤΗΣ ΔΙΑΔΙΚΑΣΙΑΣ ΥΛΟΠΟΙΗΣΗΣ 136
4.8. ΣΥΜΠΕΡΑΣΜΑΤΑ - ΠΑΡΑΤΗΡΗΣΕΙΣ 139
4.9. ΒΙΒΛΙΟΓΡΑΦΙΑ - ΑΝΑΦΟΡΕΣ 141

ΚΕΦΑΛΑΙΟ 5 ΕΠΕΚΤΑΣΗ ΤΗΣ ΔΙΑΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑΣ ΤΩΝ SOAP/XML/ HTTP ΠΡΟΤΥΠΩΝ ΑΠΟΜΑΚΡΥΣΜΕΝΗΣ ΔΙΑΧΕΙΡΙΣΗΣ 142

5.1. ΕΙΣΑΓΩΓΗ 142
5.2. ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΔΙΑΧΕΙΡΙΣΗΣ 144
5.3. Η UPnP ΤΕΧΝΟΛΟΓΙΑ ΤΟΠΙΚΗΣ ΔΙΑΧΕΙΡΙΣΗΣ 145
5.3.1. Διευθυνσιοδότηση 148
5.3.2. Ανίχνευση 149
5.3.3. Περιγραφή 150

5.3.4. Έλεγχος *151*
5.3.5. Ενημέρωση συμβάντων *151*
5.3.6. Παρουσίαση *152*
5.4. ΟΙ ΔΙΑΦΟΡΕΣ ΤΩΝ ΜΗΧΑΝΙΣΜΩΝ ΤΟΠΙΚΗΣ ΚΑΙ ΑΠΟΜΑΚΡΥΣΜΕΝΗΣ ΔΙΑΧΕΙΡΙΣΗΣ 153
5.5. ΟΙ ΑΠΑΙΤΗΣΕΙΣ ΤΗΣ ΓΕΦΥΡΩΣΗΣ 155
5.6. Η ΛΟΓΙΚΗ ΤΗΣ ΠΡΟΤΕΙΝΟΜΕΝΗΣ ΓΕΦΥΡΩΣΗΣ 156
5.6.1. Περιγραφή συσκευής *156*
5.6.2. Ανίχνευση συσκευής *158*
5.6.3. Έλεγχος συσκευής *158*
5.6.4. Ενημέρωση συμβάντος *159*
5.6.5. Οι κύριες διαδράσεις *160*
5.7. Η ΠΡΟΤΕΙΝΟΜΕΝΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ 161
5.8. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΕΙΡΑΜΑΤΟΣ ΚΑΙ ΣΥΝΟΛΙΚΗ ΑΠΟΤΙΜΗΣΗ 163
5.9. ΑΝΑΚΕΦΑΛΑΙΩΣΗ - ΣΥΜΠΕΡΑΣΜΑΤΑ 167
5.10. ΑΝΑΦΟΡΕΣ 169

ΚΕΦΑΛΑΙΟ 6 ΕΠΙΛΟΓΟΣ 170

6.1. ΑΝΑΣΚΟΠΗΣΗ 170
6.2. ΓΕΝΙΚΑ ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ 172

ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ ΚΑΙ ΣΧΗΜΑΤΩΝ

Σχήμα 2-1. Η θέση των δεμάτων λογισμικού (bundles) στην OSGi πλατφόρμα υπηρεσιών.....22
Σχήμα 3-1. Ιεραρχικά κατανεμημένη διαχείριση. 45
Σχήμα 3-2. Το μοντέλο διαχειριστή-πράκτορα. 47
Σχήμα 3-3. Η μοντελοποίηση της διαδικασίας διαχείρισης.....48
Σχήμα 3-4. Η δομή ενός SNMP (v1/v2c) μηνύματος.....50
Σχήμα 3-5. Η κωδικοποίηση BER του SNMP. (α) Η κεφαλίδα. (β) Η λίστα σύνδεσης των μεταβλητών. ....
Σχήμα 3-6. Ένα CWMP μήνυμα κωδικοποιημένο σε SOAP/XML.....51
Σχήμα 3-7.Η τοπολογία των δοκιμών. 56
Σχήμα 3-8. Σύγκριση των όγκων της παραγόμενης δικτυακής κίνησης για μετάφραση σε επίπεδο πρωτοκόλλου.....56
Σχήμα 3-9. Σύγκριση των όγκων της παραγόμενης δικτυακής κίνησης για μετάφραση σε επίπεδο αντικειμένου. 57
Σχήμα 3-10. Η επίδραση της συμπίεσης για μετάφραση σε επίπεδο πρωτοκόλλου (α) και σε επίπεδο αντικειμένου (β). 58
Πίνακας 3.1. Ο όγκος της παραγόμενης κίνησης (Get) στο δίκτυο. 60
Πίνακας 3.2. Ο όγκος της παραγόμενης κίνησης (Set) στο δίκτυο. 60
Πίνακας 3.3. Ο όγκος της παραγόμενης κίνησης (Trap) στο δίκτυο. 60
Πίνακας 3.4. Χρόνος απόκρισης του πράκτορα στα μηνύματα Get.....61
Πίνακας 3.5. Η χρήση πόρων στον πράκτορα. 61
Σχήμα 3-11. Σύγκριση των όγκων της παραγόμενης δικτυακής κίνησης. 64
Σχήμα 3-12. Σύγκριση των χρόνων διαχείρισης. 65
Σχήμα 3-13. Σύγκριση των όγκων πληροφορίας διαχείρισης. 66
Σχήμα 3-14. Σύγκριση των χρόνων επεξεργασίας. 67
Σχήμα 3-15. Οι χρόνοι ενορχήστρωσης συγκριτικά με τους χρόνους κωδικοποίησης. 68
Πίνακας 3.6. Η καθυστέρηση κυκλικού ταξιδιού συναρτήσει του αριθμού των αντικειμένων.....69
Πίνακας 3.7. Η χρήση της μνήμης στον πράκτορα. 69
Σχήμα 3-16. (α) Το περιβάλλον διαχείρισης και (β) η στοίβα πρωτόκολλων του CWMP.....73
Σχήμα 3-17. Μια τυπική CWMP συνεδρία. 75
Σχήμα 3-18. Απόσπασμα από μια τυπική Inform αίτηση. 76
Σχήμα 3-19. Η ανάλυση του όγκου της κίνησης για μια τυπική διαχειριστική συνεδρία.....77
Σχήμα 3-20. Ο όγκος των SOAP μηνυμάτων κατά την παρακολούθηση. 79
Σχήμα 3-21. Οι χρόνοι μετάδοσης του GetParameterValues μηνύματος αίτησης. 81
Σχήμα 3-22. Οι χρόνοι μετάδοσης του GetParameterValues μηνύματος απάντησης. 81
Πίνακας 3.8. Η ποιοτική αποτίμηση των μεθόδων συμπίεσης. 86
Σχήμα 3-23. Τα πειραματικά αποτελέσματα της εργασίας [48]. 87
Σχήμα 3-24. Τα χρονικά κέρδη (Gtx) των μεθόδων συμπίεσης. 87
Σχήμα 3-25. Η επίδραση της συμπίεσης στον όγκο των SOAP μηνυμάτων κατά την παρακολούθηση.....89
Πίνακας 3.9. Οι ταχύτητες εκτέλεσης της Lempel-Ziv (LZO1X) μεθόδου συμπίεσης. 90
Σχήμα 3-26. Η επίδραση της συμπίεσης στους χρόνους μετάδοσης του GetParameterValues μηνύματος αίτησης. 91
Σχήμα 3-27. Η επίδραση της συμπίεσης στους χρόνους μετάδοσης του GetParameterValues μηνύματος απάντησης. 91
Σχήμα 3-28. Ο λόγος συμπίεσης για τα CWMP μηνύματα διαχείρισης. 92
Σχήμα 3-29. Το ποσοστιαίο χρονικό κέρδος μετάδοσης κατά τη συμπίεση πραγματικού χρόνου. 93

Σχήμα 3-30. Η σύγκριση του κέρδος μετάδοσης στα πολύ μικρά μηνύματα για διαφορετικές μεθόδους ή υλοποιήσεις συμπίεσης. 94
Σχήμα 3-31. Οι χρόνοι κωδικοποίησης και συμπίεσης των μηνυμάτων διαχείρισης. 96
Σχήμα 3-32. Η δυναμική μνήμη που καταλαμβάνεται κατά την αποκωδικοποίηση. 98
Σχήμα 3-33. Η αρχιτεκτονική του πειραματικού συστήματος CWMP διαχείρισης..... 99
Σχήμα 3-34. Το ποσοστιαίο χρονικό κέρδος της κωδικοποίησης των ειδικών αναλυτών έναντι των κλασσικών DOM αναλυτών..... 102
Σχήμα 3-35. Το ποσοστιαίο χρονικό κέρδος της συμπίεσης πραγματικού χρόνου..... 104
Σχήμα 3-36. Ο συνολικός όγκος της κίνησης διαχείρισης. 104
Σχήμα 4-1.Το περιβάλλον διαχείρισης και η στοίβα πρωτοκόλλων του CWMP. 113
Σχήμα 4-2.Το εύρος προτυποποίησης για τη διαχείριση του συνδρομητικού εξοπλισμού. 114
Σχήμα 4-3.Το πεδίο δράσης των εργασιών προτυποποίησης για τη μοντελοποίηση της δικτυακής πύλης 115
Σχήμα 4-4.Τα βασικά στοιχεία της διαχείρισης. 115
Σχήμα 4-5.Η ιεραρχική δομή της CWMP μοντελοποίησης. 117
Πίνακας 4.1. Ένα απόσπασμα από το CWMP μοντέλο της δικτυακής πύλης..... 117
Σχήμα 4-6.Παράδειγμα ιεραρχίας σε μια δικτυακή πύλη. 118
Πίνακας 4.2.Οι βασικές μέθοδοι του CWMP πρωτοκόλλου. 119
Σχήμα 4-7.Το τμήμα της XML αναπαράστασης που αντιστοιχεί στις παραμέτρους της δικτυακής πύλης του πίνακα 4.1. 120
Σχήμα 4-8. Το κύριο τμήμα του προτεινόμενου εργαλείου (Device Descriptor) εισαγωγής και ανάγνωσης των CWMP μοντέλων των συνδρομητικών συσκευών. 122
Σχήμα 4-9.Η εφαρμογή επιλογής (Selection) των διαχειρίσιμων παραμέτρων 124
Σχήμα 4-10.Το γραφικό περιβάλλον της προτεινόμενης εφαρμογής (Device Creator) για την αυτόματη παραγωγή πηγαίου κώδικα CWMP πρακτόρων. 125
Σχήμα 4-11. Ο παραμετρικός κώδικας του CWMP πράκτορα. 125
Σχήμα 4-12. Η αρχιτεκτονική του παραμετρικού CWMP πράκτορα. 126
Σχήμα 4-13.Το απόσπασμα του κειμένου τρέχουσας πληροφορίας που αντιστοιχεί στον πίνακα 4.1..... 128
Σχήμα 4-14. Simple ACS: Μια περιορισμένη έκδοση του Device Tester. 135
Σχήμα 4-15. Simple ACS: Δοκιμή λήψης τιμών (Get). 135
Σχήμα 4-16. Οι αυτοματοποιημένες διαδικασίες και η χρήση των XML κειμένων σε καθεμία. 138
Σχήμα 5-1. Η συνύπαρξη των πρωτοκόλλων τοπικής και απομακρυσμένης διαχείρισης. 144
Σχήμα 5-2. Ο ACS διαχειρίζεται τις UPnP συσκευές μέσω της δικτυακής πύλης. 145
Σχήμα 5-3. Η στοίβα πρωτοκόλλων του UPnP μηχανισμού τοπικής διαχείρισης. 147
Σχήμα 5-4. (α) Η στοίβα πρωτοκόλλων του UPnP για κάθε λειτουργία, (β) Η στοίβα πρωτοκόλλων του CWMP είναι κοινή για όλες τις λειτουργίες. 154
Πίνακας 5.1.Οι βασικές μέθοδοι του CWMP πρωτοκόλλου. 155
Σχήμα 5-5. Τα αντικείμενα και οι παράμετροι που εκθέτει μια συσκευή-γέφυρα (CPE) στον ACS. (α)η συνολική δομή της εκτιθέμενης πληροφορίας, (β) η δυναμικά παραγόμενη ιεραρχία των UPnP παραμέτρων. 157
Σχήμα 5-6. Οι κύριες λειτουργίες της γέφυρας και οι αντίστοιχες διαδράσεις των μηχανισμών τοπικής (UPnP) και απομακρυσμένης (CWMP) διαχείρισης. 161
Σχήμα 5-7. Η αρχιτεκτονική του πειραματικού CWMP-UPnP συστήματος διαχείρισης. 162
Σχήμα 5-8. Το σενάριο του πειράματος και οι αντίστοιχες διαδράσεις μεταξύ του ACS, της πύλης-γέφυρας και των UPnP συνδρομητικών συσκευών. 165
Σχήμα 5-9. Τα πειραματικά αποτελέσματα της παραγόμενης κίνησης διαχείρισης. 167

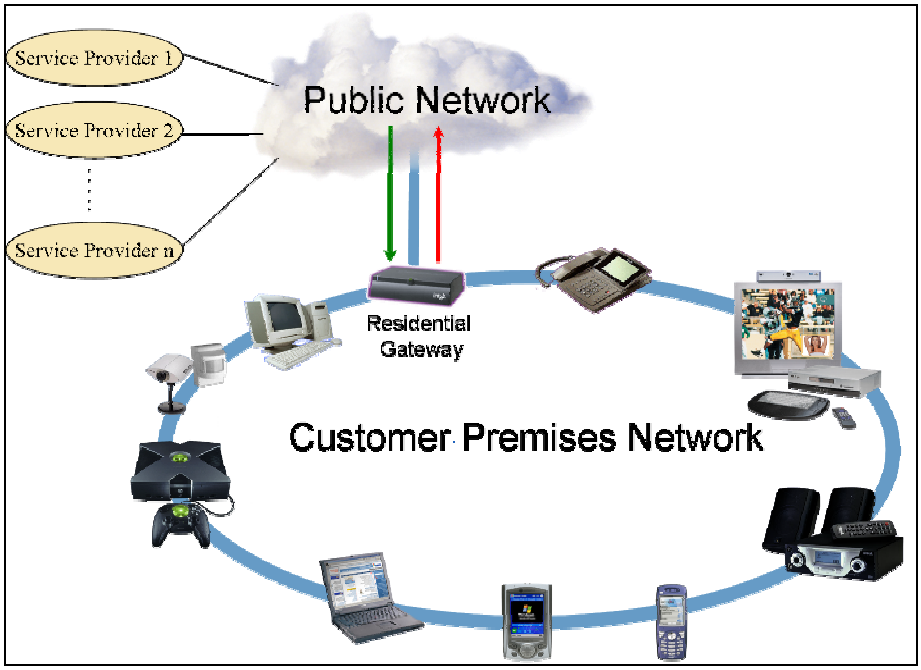
Κεφάλαιο 1
Εισαγωγή

1.1. Υπόβαθρο

Η μεγάλη εξέλιξη των τεχνολογιών πρόσβασης αλλά και η σύγχρονη τάση ενσωμάτωσης δικτυακών τεχνολογιών στα καταναλωτικά ηλεκτρονικά ανοίγουν το δρόμο για σημαντικές βελτιώσεις στην ποιότητα ζωής των πολιτών. Παράλληλα δημιουργούν νέες επιχειρηματικές ευκαιρίες για τους παρόχους, οι οποίοι μπορούν να εκμεταλλευτούν τις πλούσιες δυνατότητες των νέων συσκευών για να προσφέρουν ποικίλες υπηρεσίες μέσω του διαδικτύου. Σύντομα οι συνδρομητές θα έχουν τη δυνατότητα πρόσβασης σε μια ευρεία γκάμα πολυμεσικών υπηρεσιών, που θα παρέχονται από το τοπικό (local services) αλλά συνεχώς αυξανόμενα και από το δημόσιο δίκτυο (remote services). Υπηρεσίες όπως η εργασία από το σπίτι (home-working), η τηλε-μάθηση (e-learning), η τηλε-φροντίδα (health care services) και η ηλεκτρονική διασκέδαση (e-entertainment) είναι ορισμένες από τις σημαντικότερες ηλεκτρονικές ευκολίες, που στοχεύουν να γίνουν καθημερινή πραγματικότητα στο κοντινό μέλλον. Ένας από τους παράγοντες κλειδιά στη μαζική εκμετάλλευση των διαδικτυακών υπηρεσιών είναι το επίπεδο ευκολίας όσον αφορά στην αρχική εγκατάσταση, στη ρύθμιση αλλά και στη διαχείριση των συσκευών που τις φιλοξενούν. Η ελάχιστη ή ακόμη και μηδενική παρέμβαση του χρήστη είναι βασική απαίτηση.

Η παρούσα εργασία εστιάζει στην απομακρυσμένη διαχείριση του παραπάνω εξοπλισμού, ο οποίος από εδώ και στο εξής θα ονομάζεται «Συνδρομητικός Εξοπλισμός» (Customer Premises Equipment). Ο όρος αυτός προέρχεται από τον κόσμο των παρόχων υπηρεσιών και χρησιμοποιείται ευρέως τα τελευταία χρόνια προκειμένου να δηλώσει κάθε συσκευή μέσω της οποίας αποδίδονται μια ή περισσότερες υπηρεσίες στον τελικό χρήστη-συνδρομητή. Οι συνδρομητικές συσκευές μπορούν να διακριθούν σε δύο μεγάλες κατηγορίες: (i) τις δικτυακές συσκευές (δικτυακές πύλες, σημεία ασύρματης πρόσβασης καθώς και άλλες συσκευές τοπικής δικτύωσης) που σαν κύριο στόχο τους έχουν την παράδοση των προσφερόμενων υπηρεσιών από το διαδίκτυο στο τοπικό δίκτυο και (ii) τις συσκευές που είναι συνδεδεμένες στο τοπικό δίκτυο και «τερματίζουν» τις προσφερόμενες υπηρεσίες προκειμένου να τις αποδώσουν στον τελικό χρήστη. Η σύγχρονη τάση ενσωμάτωσης δικτυακών τεχνολογιών στα καταναλωτικά ηλεκτρονικά (Consumer Electronics) έχει μετατρέψει όλες τις συσκευές αυτές σε εν δυνάμει συνδρομητικές συσκευές, αφού μέσω της δικτυακής πύλης και του τοπικού δικτύου στο οποίο συνδέονται μπορούν να έχουν πρόσβαση στις παρεχόμενες διαδικτυακές υπηρεσίες. Το γεγονός αυτό, σε συνδυασμό με τη μεγάλη

άνθιση της βιομηχανίας των καταναλωτικών ηλεκτρονικών έχει ως συνέπεια την ολοένα και θεαματικότερη αύξηση των συσκευών, που εξοπλίζουν τα σύγχρονα νοικοκυριά και τις επιχειρήσεις. Πέραν των προσωπικών υπολογιστών, συσκευές όπως τα VoIP/Video τηλέφωνα, οι προσωπικοί ψηφιακοί βοηθοί (Personal Digital Assistances - PDAs), οι ψηφιακές μηχανές φωτογράφησης και βιντεοσκόπησης (cameras και camcorders), οι παιχνιδομηχανές (Game Consoles), οι εξυπηρετητές πολυμεσικού περιεχομένου (Media Servers), τα πολυμεσικά κινητά τηλέφωνα και οι συσκευές απόδοσης ψηφιακού συνδρομητικού video (set-top boxes) αποτελούν ορισμένες μόνο από τις βασικές κατηγορίες των συσκευών, που πρόκειται να εξοπλίσουν τα σύγχρονα και μελλοντικά συνδρομητικά δίκτυα.



Σχήμα 1-1. Η παροχή υπηρεσιών στο συνδρομητικό δίκτυο μέσω της δικτυακής πύλης.

Για την απρόσκοπτη παράδοση των διαδικτυακών υπηρεσιών στις συνδρομητικές συσκευές και την απόδοσή τους στον τελικό χρήστη απαιτείται ένα πλήθος πολύπλοκων λειτουργιών, οι οποίες επιτελούνται τόσο από τα δικτυακά στοιχεία, που μεσολαβούν μεταξύ των παρόχων και των συσκευών, που τερματίζουν τις υπηρεσίες όσο και από τις ίδιες τις τερματικές συσκευές. Η δικτυακή πύλη παίζει κυρίαρχο ρόλο καθώς είναι εκείνη που αναλαμβάνει την πρόσβαση στο διαδίκτυο και τη διανομή της στις συσκευές του τοπικού δικτύου. Ένα πλήθος λειτουργιών θα πρέπει να επιτελείται από τη δικτυακή πύλη προκειμένου να εξασφαλιστεί τόσο η συνδεσιμότητα σε όλα τα επίπεδα όσο και άλλες λειτουργίες, όπως είναι η ποιότητα υπηρεσίας (QoS). Η διευθυνσιοδότηση των τερματικών συσκευών (IP-addressing), η δρομολόγηση (Routing) και προώθηση των IP πακέτων (Forwarding) και η προστασία του τοπικού δικτύου από την επιβλαβή κίνηση (Firewalling) είναι ορισμένες από τις βασικότερες λειτουργίες μια δικτυακής πύλης. Οι λειτουργίες αυτές χρησιμοποιούν διάφορα πρωτόκολλα βασισμένα στο IP ορισμένα εκ των οποίων είναι: το DHCP (Dynamic Host Configuration Protocol) [18] για τη αυτόματη διευθυνσιοδότηση των συσκευών, το NAT (Network Address Translation) [19] για την δρομολόγηση των IP πακέτων του διαδικτύου στις συνδρομητικές

συσκευές και το IGMP (Internet Group Management Protocol) [20] για τη υποστήριξη πολυεκπομπής (IP-Multicasting). Επίσης ανάλογα με την τεχνολογία πρόσβασης η δικτυακή πύλη μπορεί να υποστηρίζει διάφορα άλλα πρωτόκολλα όπως το ATM (Asynchronous Transfer Mode) [21] για την υποστήριξη της τεχνολογίας xDSL, το Ethernet over AAL5 [22] για την ενθυλάκωση των Ethernet πακέτων σε ATM πακέτα, το PPP (Point to Point Protocol) και τα παρεμφερή του πρωτόκολλα (PPP, PAP, IPCP, PPPoE, PPPoA) για την εγκατάσταση της σύνδεσης σε δικτυακό επίπεδο.

Πολλές φορές η διανομή της πρόσβασης πρόκειται να επιτελείται με ένα τρόπο ο οποίος θα διαφοροποιεί την παροχή των δικτυακών πόρων της πύλης ανάλογα με την παρεχόμενη υπηρεσία. Με τον τρόπο αυτό θα διασφαλίζονται συγκεκριμένα επίπεδα ποιότητας υπηρεσίας (QoS) για τα οποία ο συνδρομητής θα χρεώνεται ξεχωριστά από την απλή πρόσβαση στο διαδίκτυο. Βέβαια για γίνει εφικτό κάτι τέτοιο απαιτείται και ένα πλήθος λειτουργιών του δικτύου πρόσβασης και του δικτύου κορμού (και τα δύο αυτά δίκτυα δηλώνονται με από τον όρο Public Network στο σχήμα 1-1) τα οποία είναι εκτός του πεδίου μελέτης της συγκεκριμένης διατριβής. Ωστόσο η πύλη θα πρέπει να συνεργάζεται με το δίκτυο πρόσβασης προκειμένου να μαρκάρει ανάλογα την κίνηση των συνδρομητικών συσκευών, έτσι ώστε το δίκτυο πρόσβασης αλλά και το δίκτυο κορμού να μπορούν να χειρίζονται διαφορετικά με τη σειρά τους την κίνηση των συνδρομητικών συσκευών και επομένως και την ποιότητα της παρεχόμενης υπηρεσίας.

Σε διαφορετικό βαθμό η καθεμία και ανάλογα με τις προσφερόμενες υπηρεσίες, όλες οι παραπάνω λειτουργίες απαιτούν ένα είδος ρύθμισης προκειμένου να εξασφαλίσουν τόσο την ευρυθμία τους όσο και την αρμονική συνεργασία μεταξύ τους. Για παράδειγμα, προκειμένου μια συνδρομητική συσκευή να είναι προσβάσιμη από το διαδίκτυο (π.χ. ένας οικιακός Web-Server) είναι δυνατόν (στην περίπτωση που χρησιμοποιείται το NAT πρωτόκολλο) να χρειαστεί η ρύθμιση των σχετικών πίνακες δρομολόγησης της πύλης. Πέραν των λειτουργιών της δικτυακής πύλης οι συνδρομητικές συσκευές πολλές φορές είναι δυνατόν να επιτελούν και αυτές πολύπλοκες λειτουργίες που χρειάζονται ρύθμιση. Για παράδειγμα ένα VoIP/Video τηλέφωνο μπορεί να υποστηρίζει διάφορα πρωτόκολλα σηματοδοσίας (π.χ. SIP [23], MGCP [24] και H323 [25]) αλλά και διάφορες μεθόδους κωδικοποίησης (π.χ. MPEG-2, MPEG-4, G.711 και GSM). Προκειμένου να ενεργοποιηθεί μια υπηρεσία video-φωνίας θα πρέπει οι συσκευές προηγουμένως να έχουν ενημερωθεί για το πρωτόκολλο σηματοδοσίας, που υποστηρίζει ο πάροχος αλλά και για τον τρόπο με τον οποίο πρέπει να κωδικοποιηθούν τα δεδομένα φωνής και video προκειμένου να αποδοθεί η υπηρεσία με συγκεκριμένη ποιότητα (QoS).

Πέραν της αυτόματης ρύθμισης κατά την εγκατάσταση του συνδρομητικού εξοπλισμού (auto-configuration) αλλά και κατά τη διάρκεια παράδοσης των υπηρεσιών (Dynamic Service Provisioning) είναι απαραίτητη πολλές φορές η συνεχής παρακολούθηση της κατάστασης και των επιδόσεων (Performance and Status monitoring) μιας υπηρεσίας ή της συσκευής, που την υποστηρίζει. Για παράδειγμα ένας δικτυακός πάροχος θα πρέπει να έχει πρόσβαση στις διάφορες παραμέτρους των επιδόσεων μιας δικτυακής πύλης (π.χ. ρυθμός σφαλμάτων κατά τη μετάδοση, υπερχείλιση των προσωρινών αποθηκευτών) προκειμένου να ελέγχει και να

προλαμβάνει δυσλειτουργίες της πύλης κατά την παράδοση των υπηρεσιών. Επίσης ο πάροχος θα πρέπει να είναι σε θέση να διαγνώσκει και να διορθώνει (Diagnostics) τυχόν βλάβες, που εμφανίζονται στη δικτυακή πύλη ή στις τερματικές συσκευές και διακόπτουν την παροχή των ποιοτικών υπηρεσιών στον τελικό χρήστη. Τέλος, σε ένα περιβάλλον όπου πολλοί πάροχοι διαθέτουν μια γκάμα διαφορετικών και ολοένα αναβαθμιζόμενων υπηρεσιών, θα πρέπει να παρέχονται μηχανισμοί μεταφόρτωσης και ενεργοποίησης νέου λογισμικού (software/firmware management) στις συνδρομητικές συσκευές.

Από όλα τα παραπάνω γίνεται φανερή η ανάγκη ύπαρξης ενός ενιαίου μηχανισμού απομακρυσμένης υποστήριξης και συντήρησης των συσκευών, που παραδίδουν και αποδίδουν τις διαδικτυακές υπηρεσίες στον τελικό χρήστη-συνδρομητή. Μάλιστα η υιοθέτηση ενός προτύπου απομακρυσμένης διαχείρισης από τους κατασκευαστές και τους παρόχους κρίνεται απαραίτητη για τη συγκράτηση του κόστους υποστήριξης των υπηρεσιών και την ευχάριστη εμπειρία των συνδρομητών κατά τη χρήση τους.

Κατά τη δεκαετία 1990-2000 η διαχείριση των IP δικτύων βασίστηκε σχεδόν αποκλειστικά στη χρήση ενός απλού πρωτοκόλλου, του SNMP (Simple Network Management Protocol) [1-6]. Τα κύρια πλεονεκτήματά του ήταν η απλότητα και η διαλειτουργικότητα. Ο σχεδιασμός και η ενσωμάτωση SNMP πρακτόρων στις διαχειριζόμενες συσκευές ήταν μια εύκολη διαδικασία. Το γεγονός αυτό οδήγησε στη σταδιακή αντικατάσταση των ιδιόκτητων λύσεων διαχείρισης από ανοιχτές λύσεις. Κάθε τύπος εξοπλισμού, οποιουδήποτε κατασκευαστή μπορούσε πλέον να είναι διαχειρίσιμος από μία πλατφόρμα τρίτου, όπως για παράδειγμα το HP Open View, το Cabletron Spectrum, το IBM/Tivoli Netview και το Sun Microsystems Solstice. Μάλιστα, τρία χρόνια μόλις μετά την πρώτη έκδοση των προδιαγραφών του SNMP [1] από την IETF (SNMPv1/1990) η υποστήριξή του σε όλα τα είδη των δικτυακών συσκευών ήταν μονόδρομος για τους κατασκευαστές, καθώς αποτελούσε απαίτηση των περισσοτέρων πελατών. Η κατάσταση όμως άλλαξε μετά το 1995 όπου η δεύτερη έκδοση του SNMP [2-6] παρόλο που είχε δημιουργήσει μεγάλες προσδοκίες τελικά αποδείχτηκε αποτυχία τόσο σε επίπεδο τεχνικό όσο και σε επίπεδο εμπορικής προώθησης [7-9]. Όταν το 1998 δημοσιεύτηκε η τρίτη έκδοση του SNMP [10] η αγορά δεν πίστευε πλέον στη δυνατότητά του να εξελιχθεί και να καλύψει τις τρέχουσες αλλά και μελλοντικές της ανάγκες. Η βασικότερη αιτία για αυτή την αμφισβήτηση είναι ότι το πρόβλημα για το οποίο σχεδιάστηκε το SNMP, στα τέλη της δεκαετίας του ’90 γίνεται ολοένα και πιο διαφορετικό από εκείνο που καλείται να λύσει την επόμενη δεκαετία. Σήμερα σχεδόν όλες οι συσκευές είναι δικτυωμένες, πολλές εκ των οποίων είναι ασύρματες, με σημαντικά υψηλότερες δυνατότητες επεξεργασίας και αποθήκευσης. Τα τηλεφωνικά δίκτυα συγχωνεύονται με εκείνα των δικτύων δεδομένων, ενώ η συνεχής αύξηση του εύρους ζώνης στα δίκτυα πρόσβασης δημιουργεί μια εντελώς διαφορετική εικόνα με ποικίλες υπηρεσίες να προσφέρονται από διαφορετικούς παρόχους και να καταναλώνονται από πολλούς χρήστες, μέσω ενός πλήθους διαφορετικών συσκευών.

Έτσι λοιπόν μετά το 1995 ο τρόπος διαχείρισης των δικτύων άρχισε να αναθεωρείται. Σε αυτή τη νέα προσπάθεια ήταν προφανής η τάση θεώρησης του προβλήματος από την οπτική γωνία της μηχανικής λογισμικού. Μηχανικοί λογισμικού από διαφορετικούς κόσμους (π.χ.

αντικειμενοστραφής, τεχνητή νοημοσύνη, βάσεις δεδομένων) ανέλυσαν τον τρόπο της μέχρι τότε διαχείρισης των IP δικτύων και κατέληξαν στο συμπέρασμα ότι οι λύσεις, που βασίζονται στο SNMP δεν μπορούσαν να αντέξουν τη σύγκριση με εκείνες των κατανεμημένων εφαρμογών. Η κοινότητα της μηχανικής λογισμικού είχε να προσφέρει πολλές εναλλακτικές τεχνολογίες, όπως οι τεχνολογίες ιστού, τα ενεργά δίκτυα, η CORBA, οι έξυπνοι πράκτορες κ.α. Το γεγονός αυτό σύντομα οδήγησε σε ένα μεγάλο αριθμό προτάσεων ο οποίος όμως αντί να βοηθήσει τελικά περισσότερο μπέρδεψε την αγορά. Κανείς δεν μπορούσε να εγγυηθεί για την τεχνολογία που θα επικρατούσε στο μέλλον. Όπως ήταν φυσικό οι περισσότεροι ήταν πολύ διστακτικοί στο να φύγουν από μια τόσο ομοιογενή και ασφαλή αγορά, που είχε ανοίξει το SNMP και να πάνε σε μια υπερδραστήρια αλλά και επικίνδυνη ταυτόχρονα. Οι περισσότεροι λοιπόν αποφάσισαν να αναβάλλουν την επένδυσή τους μέχρις ότου διαφανεί πιο ξεκάθαρα μια λύση ικανή να επικρατήσει στην αρένα της σύγχρονης απομακρυσμένης διαχείρισης..

Τα τελευταία πέντε χρόνια γίνεται αρκετός λόγος για μια λύση, που θα βασίζεται στις τεχνολογίες ιστού και κυρίως στο συνδυασμό του HTTP ως πρωτοκόλλου μεταφοράς της διαχειριστικής πληροφορίας και της XML σαν γλώσσα δόμησης αυτής της πληροφορίας. Μάλιστα, οι προσπάθειες προτυποποίησης της IETF (NetConf protocol) [11] του UPnP φόρουμ (UPnP protocol) [12] αλλά και πολύ πρόσφατα του DSL φόρουμ (CPE WAN Management protocol) [13] ενισχύουν τη διαπίστωση ότι ο συνδυασμός XML/HTTP μπορεί να δώσει μια κοινώς αποδεκτή λύση στο πρόβλημα της διαχείρισης των δικτύων αλλά και των υπηρεσιών που αυτά προσφέρουν. Είναι σχεδόν κοινή η διαπίστωση [14-17] για τη δραματική βελτίωση, που επιφέρει η χρήση της XML στη μοντελοποίηση της διαχειριστικής πληροφορίας αλλά και η χρήση του HTTP στη βελτίωση της προσβασιμότητας, της ασφάλειας και της αξιοπιστίας στη διαχείριση. Ωστόσο, δεν είναι λίγοι εκείνοι οι οποίοι κατά καιρούς εξέφρασαν τις αντιρρήσεις τους σχετικά με τις επιδόσεις μια τέτοιας λύσης. Πιο συγκεκριμένα, υπήρξε αρκετή ανησυχία για τις μεγάλες απαιτήσεις που ενδεχομένως να έχει η XML κωδικοποίηση της διαχειριστικής πληροφορίας τόσο σε δικτυακούς πόρους όσο και σε πόρους των πρακτόρων διαχείρισης.

1.2. Αντικείμενο της διατριβής

Το ευρύτερο πεδίο ενδιαφέροντος της διατριβής σχετίζεται με την εφαρμογή της τεχνολογίας XML (eXtensible Markup Language) στην απομακρυσμένη διαχείριση των IP συνδρομητικών συσκευών αλλά και των υπηρεσιών, που προσφέρονται διαδικτυακά, μέσω του εξοπλισμού αυτού από τους διάφορους παρόχους.

Ειδικότερα, η διατριβή επικεντρώνεται: (i) στην ανεύρεση μεθοδολογιών και τεχνικών για τη βελτίωση των επιδόσεων των XML διαχειριστικών λύσεων, (ii) στη μείωση του κόστους ανάπτυξης και συντήρησης των αντίστοιχων πρακτόρων διαχείρισης και (iii) στη βελτίωση της διαλειτουργικότητας των XML προτύπων διαχείρισης.

Ο πρώτος στόχος της διατριβής είναι να συμβάλλει στην επίλυση του προβλήματος των υψηλών χρόνων διαχείρισης, που προκύπτουν εξαιτίας του μεγάλου όγκου των XML διαχειριστικών μηνυμάτων αλλά και της XML κωδικοποίησης. Το πρόβλημα γίνεται

ιδιαίτερα έντονο σε περιβάλλοντα μαζικής διαχείρισης συνδρομητικών συσκευών μέσω δικτύων πρόσβασης με περιορισμένο εύρος ζώνης. Η χαμηλή ταχύτητα κατά τη ρύθμιση των συνδρομητικών συσκευών σε συνδυασμό με το μεγάλο πλήθος τους μπορεί να αυξήσει σημαντικά το κόστος διαχείρισης και να επιδράσει αρνητικά στην κλιμακωσιμότητα. Για το πρόβλημα αυτό προτείνεται η ιδέα της επιλεκτικής συμπίεσης των XML διαχειριστικών μηνυμάτων σε πραγματικό χρόνο και ορίζονται οι συνθήκες κάτω από τις οποίες η συμπίεση αυτή είναι κερδοφόρα και επομένως θα πρέπει να εφαρμόζεται. Επιπρόσθετα για την XML κωδικοποίηση της διαχειριστικής πληροφορίας προτείνεται η αποφυγή της κλασσικής και προτυποποιημένης μεθόδου DOM (Document Object Model). Αντί της DOM προτείνεται η προσέγγιση των ειδικών γραμματικών αναλυτών και εναλλακτικά εκείνη των SAX (Simple API for XML) γραμματικών αναλυτών.

Ο δεύτερος στόχος της διατριβής είναι να συμβάλλει στη μείωση του κόστους ανάπτυξης, δοκιμών και συντήρησης των πρακτόρων των συσκευών, που συμμορφώνονται με τα νέα XML πρότυπα διαχείρισης. Οι μεγάλες απαιτήσεις των σύγχρονων και μελλοντικών διαδικτυακών υπηρεσιών για διαχείριση, σε συνδυασμό με τη μεγάλη διαφοροποίηση των συνδρομητικών συσκευών καθιστούν αναγκαίες τις μεθοδολογίες εκείνες, που θα επιτρέψουν τη ταχεία και μαζική ανάπτυξη, δοκιμή και συντήρηση των αντίστοιχων πρακτόρων διαχείρισης. Για το πρόβλημα αυτό προτείνεται μια ολοκληρωμένη μεθοδολογία, που βασίζεται στην έκφραση του μοντέλου διαχείρισης με τη βοήθεια της γλώσσας XML. Μάλιστα για την αποτίμηση της προσέγγισης αυτής αναπτύσσεται μια σουίτα λογισμικού η οποία παράγει αυτόματα πηγαίο κώδικα για τους πράκτορες διαχείρισης, που συμμορφώνονται με το CWMP (Customer Premises Equipment WAN Management Protocol) πρωτόκολλο απομακρυσμένης διαχείρισης του DSL φόρουμ.

Ο τρίτος και τελευταίος στόχος της διατριβής είναι να συμβάλλει αφενός στη διαλειτουργικότητα των νέων XML διαχειριστικών προτύπων και αφετέρου στην επιτάχυνση της αξιοποίησης νέων υπηρεσιών, που απαιτούν τον έλεγχο και τη διαχείριση των συνδρομητικών συσκευών. Σε μια αγορά που καθημερινά αλλάζει η αργή διαδικασία της μοντελοποίησης των συνδρομητικών συσκευών από τους οργανισμούς προτυποποίησης θέτει σε κίνδυνο τη γρήγορη εξάπλωση των αντίστοιχων προτύπων και επομένως και την τελική αποδοχή τους. Παράλληλα καθυστερεί σημαντικά την ανάπτυξη και αξιοποίηση νέων υπηρεσιών για την ενεργοποίηση των οποίων απαιτείται η διαχείριση και ο έλεγχος διαφορετικών συνδρομητικών συσκευών. Για το πρόβλημα αυτό προτείνεται η ιδέα της γεφύρωσης των νέων προτύπων απομακρυσμένης διαχείρισης με ήδη επιτυχημένα και ευρέως υποστηριζόμενα πρότυπα τοπικής διαχείρισης. Μάλιστα για την αποτίμηση της εφαρμοσιμότητας μιας τέτοιας προσέγγισης σχεδιάζεται και υλοποιείται μια γέφυρα. Η γέφυρα αυτή μπορεί να φιλοξενείται σε οποιαδήποτε συνδρομητική συσκευή, που υποστηρίζει το νέο CWMP πρωτόκολλο και επιτρέπει την απομακρυσμένη διαχείριση των υπολοίπων συσκευών, που συμμορφώνονται με το ήδη επιτυχημένο πρωτόκολλο τοπικής διαχείρισης, UPnP (Universal Plug and Play).

1.3. Δομή της διατριβής

Η παρούσα διατριβή διαρθρώνεται σε έξι κεφάλαια, εκ των οποίων το πρώτο αποτελεί μία πολύ σύντομη εισαγωγή, προβάλλοντας την ανάγκη που έχει δημιουργηθεί τα τελευταία χρόνια για μαζική διαχείριση των IP συνδρομητικών συσκευών.

Στο δεύτερο κεφάλαιο, αναλύονται και αξιολογούνται οι υποψήφιες τεχνολογίες για αυτού του είδους τη διαχείριση. Μέσα από την ανάλυση διακρίνονται οι λύσεις, που χρησιμοποιούν ως βασικά τους συστατικά τις τεχνολογίες ιστού και πιο συγκεκριμένα το ευρέως διαδεδομένο πρωτόκολλο HTTP και τη γλώσσα XML. Μάλιστα σε αυτά τα δύο έρχεται να προστεθεί και το πρωτόκολλο SOAP για την κλήση των απομακρυσμένων διεργασιών διαχείρισης. Τέλος περιγράφονται συνοπτικά και αξιολογούνται οι προσπάθειες προτυποποίησης των διαφόρων οργανισμών με στόχο την καθιέρωση ενός νέου SOAP/XML/HTTP πρωτοκόλλου διαχείρισης.

Στο τρίτο κεφάλαιο περιγράφονται οι προβληματισμοί, που έχουν εκφραστεί πρόσφατα σχετικά με την καταλληλότητα του SOAP/XML υποδείγματος στη διαχείριση εξαιτίας των επιδόσεών του. Κατόπιν μοντελοποιείται το πρόβλημα της διαχείρισης, προκειμένου να αποτιμηθούν σωστά τα αποτελέσματα των πρόσφατων σχετικών εργασιών της ακαδημαϊκής κοινότητας, οι οποίες στη συνέχεια και αναλύονται. Αφού καθορίζονται τα ακριβή σημεία τα οποία δεν καλύπτουν επαρκώς οι προαναφερθείσες εργασίες, παρουσιάζεται η εργασία της διατριβής και τεκμηριώνεται θεωρητικά και πειραματικά η συνεισφορά της.

Στο τέταρτο κεφάλαιο περιγράφεται η ιδέα των παραμετρικών πρακτόρων διαχείρισης και αναλύεται η εφαρμογή της στην ανάπτυξη, δοκιμή και συντήρηση πρακτόρων, που συμμορφώνονται με το νέο πρότυπο απομακρυσμένης διαχείρισης συνδρομητικών συσκευών, CWMP. Κατόπιν παρουσιάζεται αναλυτικά η προτεινόμενη αρχιτεκτονική των παραμετρικών πρακτόρων, ενώ στο τέλος περιγράφεται το πείραμα που επιλέχθηκε για τη συνολική αποτίμηση της προτεινόμενης μεθοδολογίας.

Το πέμπτο κεφάλαιο προτείνει την επέκταση της διαλειτουργικότητας των νέων μηχανισμών απομακρυσμένης διαχείρισης μέσω της γεφύρωσης των νέων προτύπων απομακρυσμένης διαχείρισης με τα ήδη διαδεδομένα πρότυπα τοπικής διαχείρισης. Κατόπιν εξηγεί τους λόγους για τους οποίους επιλέχθηκαν τα πρωτόκολλα CWMP και UPnP, σαν τα καταλληλότερα πρότυπα για γεφύρωση. Αφού αναλύει τις ομοιότητες και διαφορές τους, προτείνει ένα ολοκληρωμένο μηχανισμό γεφύρωσης, ο οποίος πληροί τις προϋποθέσεις συμμόρφωσης με τα πρότυπα ενώ παράλληλα διασφαλίζει την κλιμακωσιμότητα της λύσης.

Τέλος το έκτο κεφάλαιο συνοψίζει το περιεχόμενο και ανακεφαλαιώνει τα καινοτόμα στοιχεία της παρούσας διατριβής. Παράλληλα για κάθε ενότητα αναδεικνύει θέματα που χρήζουν περαιτέρω διερεύνησης.

1.4. Αναφορές

[1] J. Case, M. Fedor, M. Schoffstall, and J. Davin (Eds.) "A Simple Network Management Protocol (SNMP)", IETF RFC 1157, May 1990.

[2] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser (Eds.), "Manager-to-Manager Management Information Base", IETF RFC 1451, April 1993.

[3] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser (Eds.), "Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)", IETF RFC 1902, January 1996.

[4] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser (Eds.), "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", IETF RFC 1905, January 1996.

[5] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser (Eds.), "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", IETF RFC 1906, January 1996.

[6] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser (Eds.) "Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)", IETF RFC 1907, January 1996.

[7] K. McCloghrie, "The SNMP Framework", The Simple Times, 4(1):9–10, 1996.

[8] D.T. Perkins, "SNMP Versions", The Simple Times, 5(1):13–14, 1997.

[9] W. Stallings, "SNMP, SNMPv2, SNMPv3, and RMON 1 and 2" Third edition, Addison-Wesley, Reading, MA, USA, 1999.

[10] W. Stallings. "SNMPv3: A Security Enhancement to SNMP". IEEE Communications Surveys, 1(1):2–17, 1998.

[11] IETF, "Network Configuration (Netconf)," http://www.ietf.org/html.charters/netconf-charter.html.

[12] UPnP Forum, "UPnP device architecture 1.0", December 2003, http://www.upnp.org

[13] DSL Forum, "CPE WAN management protocol", Technical Report 069, May 2004.

[14] J. Schoenwaelder, A. Pras, and J.-P. Martin-Flatin, "On the Future of Internet Management Technologies," IEEE Communications Magazine, vol. 41, no. 10, Oct. 2003.

[15] M.-J. Choi et al., "XML-Based Configuration Management for IP Network Devices," IEEE Communications Magazine, vol. 42, no. 7, July 2004.

[16] G. Pavlou, Paris Flegkas, Stelios Gouveris and Antonio Liotta, "On Management Technologies and the Potential of Web Services," IEEE Communications Magazine, vol. 42, no. 7, July 2004.

[17] JP Martin-Flatin, "Web-Based Management of IP Networks and Systems", Publisher: John Wiley and Sons, 2003.

[18] R. Droms, "Dynamic Host Configuration Protocol", IETF RFC 2131, http://www.ietf.org/rfc/ rfc2131.txt.

[19] P. Srisuresh, IP Network Address Translator (NAT) Terminology and Considerations, IETF RFC 2663, http://www.ietf.org/rfc/ rfc2663.txt.

[20] W. Fenner, "Internet Group Management Protocol, Version 2", IETF RFC 2236, http://www.ietf.org /rfc/rfc2663.txt.

[21] .M. Perez et. al., "ATM Signaling Support for IP over ATM", IETF RFC 1755, http://www.ietf.org /rfc/rfc2663.txt.

[22] D. Grossman et. al., "Multiprotocol Encapsulation over ATM Adaptation Layer 5", RFC 2684.

[23] J. Rosenberg, "SIP: Session Initiation Protocol", IETF RFC 3261, http://www.ietf.org /rfc/rfc3261.txt.

[24] M. Arango, "Media Gateway Control Protocol (MGCP) Version 1.0", RFC 2705, http://www.ietf.org /rfc/rfc2705.txt.

[25] ITU-T, "The Standard H.323", http://www.itu.int/home/index.html.

Κεφάλαιο 2
Οι σύγχρονες τεχνολογίες διαχείρισης

2.1. Εισαγωγή

Για την αυτόματη εγκατάσταση, ενεργοποίηση και υποστήριξη των διαδικτυακών υπηρεσιών αλλά και την απομακρυσμένη διάγνωση και διόρθωση ενδεχόμενων σφαλμάτων στο δίκτυο ή στις τερματικές συσκευές χρειάζεται ένα πλήθος από διαχειριστικές ενέργειες. Μάλιστα οι ενέργειες αυτές θα πρέπει να επιτελούνται σε διαφορετικά επίπεδα. Θα μπορούσε κανείς να ξεχωρίσει τις ακόλουθες βασικές κατηγορίες διαχείρισης: (i) τη διαχείριση του δικτύου (Network Management), (ii) τη διαχείριση των εφαρμογών (Application Management), (iii) τη διαχείριση των συστημάτων (Systems Management), (iv) τη διαχείριση των υπηρεσιών (Services Management) και (v) τη διαχείριση των πολιτικών (Policy Management).

Το OSI (Open Systems Interconnection) διακρίνει δύο τύπους *διαχείρισης δικτύου*: (i) τη «διαχείριση δικτυακών στοιχείων» (element management), που είναι η διαχείριση ορισμένων τμημάτων του δικτύου και (ii) τη «διαχείριση δικτύου» (network management) δηλαδή ενός ολόκληρου δικτύου σαν μια οντότητα. Η διαχείριση δικτυακών στοιχείων αφορά στις αλληλεπιδράσεις μεταξύ διαχειριστή-πράκτορα, ενώ η διαχείριση του δικτύου αφορά σε εκείνες μόνο μεταξύ διαχειριστών. Αντίθετα, για πολλά χρόνια στον κόσμο των IP δικτύων (όπου τα κομμάτια του εξοπλισμού υποστηρίζουν τη στοίβα TCP/IP) η διαχείριση δικτύου είχε να κάνει αποκλειστικά με τη διαχείριση των δικτυακών στοιχείων, όπως για παράδειγμα των IP δρομολογητών, γεφυρών και μεταγωγέων. Ωστόσο, όταν ξεκίνησε να εμφανίζεται η έννοια της ποιότητας υπηρεσίας (QoS) στα IP δίκτυα, ο κόσμος του IP άρχισε να ενδιαφέρεται και για τη διαχείριση των από άκρο-σε άκρο (end-to-end) δικτυακών χαρακτηριστικών. Έτσι η έννοια της «διαχείρισης δικτύου» στον κόσμο του IP (στον οποίο και εστιάζει η παρούσα διατριβή) περιλαμβάνει πλέον τόσο τη διαχείριση δικτυακών στοιχείων όσο και τη διαχείριση ολόκληρου του δικτύου.

Η *διαχείριση των εφαρμογών* συνίσταται στη διαχείριση όλων εκείνων των παραμέτρων λογισμικού και υλισμικού (S/W και H/W) που διασφαλίζουν τη σωστή λειτουργία μιας εφαρμογής (π.χ. VoIP) η οποία αλληλεπιδρά με τον πελάτη, μέσω της συσκευής που τη φιλοξενεί. Κύριος στόχος αυτού του είδους της διαχείρισης είναι η εγκατάσταση και ενεργοποίηση των υπηρεσιών που προσφέρονται από τον πάροχο καθώς και η διασφάλιση της ομαλής λειτουργίας των εφαρμογών αυτών. Χαρακτηριστικό παράδειγμα είναι η (απομακρυσμένη) εγκατάσταση μιας VoIP εφαρμογής στη συσκευή του πελάτη και η

περαιτέρω διαχείρισή της, η οποία μπορεί να συνίσταται στον έλεγχο της πρόσβασης, στην απεικόνιση των επιδόσεών της, στην επιλογή του κωδικοποιητή κτλ.

Η *διαχείριση των συστημάτων* στοχεύει στη διασφάλιση της ομαλής λειτουργίας της ίδιας της συσκευής που αποτελεί τη διεπαφή με το χρήστη. Πρόκειται δηλαδή για τη διαχείριση όλων εκείνων των παραμέτρων που δεν έχουν να κάνουν με την ίδια την εφαρμογή (ή τις εφαρμογές) αλλά με όλα τα υπόλοιπα συστήματα υποστήριξής της. Χαρακτηριστικά παραδείγματα διαχείρισης συστήματος είναι η απεικόνιση χρήσης της κεντρικής επεξεργαστικής μονάδας, της μνήμης, του σκληρού δίσκου, των διεργασιών, του συστήματος των αρχείων καθώς και η διάγνωση βλαβών υλικού.

Καθώς οι παρεχόμενες υπηρεσίες στον τελικό χρήστη υλοποιούνται από τις εφαρμογές, η *διαχείριση των υπηρεσιών* περικλείει μέσα της και τη διαχείριση των εφαρμογών. Επιπλέον, σε πολλές περιπτώσεις προκειμένου να λειτουργήσει και μάλιστα αποδεκτά μια εφαρμογή απαιτούνται συγκεκριμένοι δικτυακοί πόροι («δικτυακές υπηρεσίες»), οι οποίοι για να αποκτηθούν συνήθως υπογράφεται μια συμφωνία με τον πάροχο του δικτύου. Ανάλογα με το επιχειρηματικό μοντέλο οι δικτυακές αυτές υπηρεσίες μπορεί να χρεώνονται είτε άμεσα (ο πελάτης πληρώνει στον πάροχο του δικτύου για τις υπηρεσίες που προσφέρει εκείνο στην εφαρμογή του), είτε έμμεσα (ο πάροχος των εφαρμογών / περιεχομένου πληρώνει στον πάροχο του δικτύου για τις υπηρεσίες που προσφέρει ο τελευταίος στις εφαρμογές που πουλάει ο πρώτος στον τελικό πελάτη). Και στις δύο περιπτώσεις οι πελάτες συνάπτουν με τους παρόχους του δικτύου τα γνωστά «συμβόλαια επιπέδου υπηρεσίας» (Service Level Agreements – SLAs). Η έννοια λοιπόν της διαχείρισης των υπηρεσιών είναι στενά συνδεδεμένη με την έννοια του συμβολαίου (SLA) και έχει να κάνει με όλες εκείνες τις ενέργειες (π.χ. παρακολούθηση των παρεχόμενων δικτυακών πόρων και παρέμβαση στο δίκτυο ή στην εφαρμογή όταν χρειάζεται) που διασφαλίζουν την τήρηση των συμφωνηθέντων. Επομένως η διαχείριση της υπηρεσίας περιλαμβάνει τόσο τη διαχείριση των εφαρμογών όσο και τη διαχείριση εκείνων των παραμέτρων του δικτύου (π.χ. QoS παράμετροι) που διασφαλίζουν τα συμφωνηθέντα με τους πελάτες.

Τέλος, η *διαχείριση πολιτικών* έχει να κάνει με τον καθορισμό και την επιβολή υψηλού επιπέδου διαχειριστικών αποφάσεων. Μάλιστα, αλληλεπιδρά με όλους ή με ένα μέρος εκ των προαναφερθέντων τύπων διαχείρισης. Για παράδειγμα, όταν ένας νέος τύπος υπηρεσίας (π.χ. Video conferencing με συγκεκριμένα QoS χαρακτηριστικά) πρόκειται να υποστηριχτεί θα πρέπει να ενημερώνεται η διαχείριση υπηρεσίας προκειμένου να επιβάλλει τις αλλαγές τόσο σε επίπεδο δικτύου (π.χ. μήκος ουρών στους δρομολογητές) όσο και σε επίπεδο εφαρμογής (π.χ. επιλογή Video κωδικοποίησης).

Στο σημείο αυτό θα πρέπει να τονιστεί ότι η για τη συνολική διαχείριση μιας παρεχόμενης υπηρεσίας συνήθως απαιτείται ανταλλαγή πληροφορίας μεταξύ διαφορετικών διαχειριστών (π.χ. μεταξύ των διαχειριστών των πολιτικών και εκείνων των δικτυακών συσκευών). Ωστόσο η παρούσα διατριβή δεν ασχολείται με την αλληλεπίδραση μεταξύ διαχειριστών, αλλά εστιάζει σε εκείνη μεταξύ των διαχειριστών και των πρακτόρων διαχείρισης.

Στο παρόν κεφάλαιο δίνονται συνοπτικά οι λόγοι για τους οποίους το SNMP (Simple Network Management Protocol) πρωτόκολλο δεν επαρκεί για να υποστηρίξει τις σύγχρονες και μελλοντικές διαδικτυακές υπηρεσίες. Στη συνέχεια αναλύονται και αξιολογούνται τα πρότυπα και οι τεχνολογίες που έχουν προταθεί κατά καιρούς για τη διαχείριση των δικτύων και των παρεχόμενων από αυτά υπηρεσιών. Τέλος εξηγούνται αναλυτικά οι λόγοι για τους οποίους οι τεχνολογίες ιστού και συγκεκριμένα η γλώσσα XML και το HTTP πρωτόκολλο είναι ικανές να ανταποκριθούν στις σύγχρονες και μελλοντικές ανάγκες της διαχείρισης.

2.2. Τα πλεονεκτήματα του SNMP προτύπου

Πριν αναφερθούν οι αδυναμίες του SNMP προτύπου, κρίνεται σκόπιμη μια σύντομη ανασκόπηση των ισχυρών του σημείων που του χάρισαν τη σχεδόν απόλυτη κυριαρχία στη διαχείριση των δικτύων, κατά τη δεκαετία 1990-2000. Τα σημαντικότερα λοιπόν χαρακτηριστικά του είναι τα ακόλουθα: (i) η διαλειτουργικότητα, (ii) η απλότητα, (iii) η εκτεταμένη υποστήριξη από τους κατασκευαστές IP εξοπλισμού, (iv) οι μικρές απαιτήσεις των SNMP πρακτόρων για πόρους από το διαχειριζόμενο σύστημα, (v) το χαμηλό κόστος υλοποίησης των SNMP πρακτόρων και (vi) το ίδιο ελάχιστο μεσισμικό παντού. Ωστόσο η μεγάλη επιτυχία του SNMP οφείλεται κυρίως στην απλότητά του. Η OSI (Open Systems Interconnection) διαχείριση (αναλύεται σε επόμενη παράγραφο) δεν γνώρισε την επιτυχία στα IP δίκτυα εξαιτίας της πολυπλοκότητάς της. Προσπάθησε να κάνει τα πάντα σωστά και απέτυχε. Αντίθετα το SNMP εστίασε στα απολύτως απαραίτητα και ανέβαλλε κάθε τι πολύπλοκο που θα μπορούσε να καθυστερήσει τη διάδοσή του (π.χ. ασφάλεια και κατανεμημένη αρχιτεκτονική). Η στρατηγική αυτή επιλογή είχε ως αποτέλεσμα τις ταχύτατες και με μικρό κόστος υλοποιήσεις SNMP πρακτόρων από τους κατασκευαστές εξοπλισμού. Ανοιχτός πηγαίος κώδικας για τους SNMP πράκτορες ήταν ελεύθερα διαθέσιμος στο διαδίκτυο. Μάλιστα οι διανομές του Carnegie Mellon University (CMU) και του Massachusetts Institute of Technology (MIT) ήταν ίσως οι πιο δημοφιλείς στις αρχές τις δεκαετίας του 2000, όταν η αγορά του SNMP εκτινάχθηκε. Με αυτό τον τρόπο ένας κατασκευαστής μπορούσε πολύ εύκολα να πάρει τον κώδικα αυτό, να τον τροποποιήσει σύμφωνα με τις δικές του ανάγκες και να τον ενσωματώσει στον προϊόν του. Επίσης το SNMP ήταν πολύ εύκολο να γίνει κατανοητό και από τους ίδιους του πελάτες και έτσι πήρε πολύ λίγο χρόνο στους διαχειριστές των δικτύων να εξοικειωθούν με αυτό. Ωστόσο η απλότητα του SNMP, όπως θα φανεί και στη συνέχεια, αποδεικνύεται το κύριο εμπόδιο για την υποστήριξη των δικτύων και υπηρεσιών της επόμενης γενιάς.

2.3. Τα προβλήματα του SNMP προτύπου

Η εκτεταμένη χρήση του SNMP για μια ολόκληρη δεκαετία ανέδειξε ένα πλήθος προβλημάτων. Πολλά εξ αυτών είναι ελάσσονος σημασίας αλλά υπάρχουν και ορισμένα τα οποία είναι σημαντικά και έχουν να κάνουν τόσο με την αρχιτεκτονική όσο και με το ίδιο το πρωτόκολλο διαχείρισης. Η ενότητα αυτή αναλύει τις βασικότερες αδυναμίες εξαιτίας των οποίων το SNMP είναι ανεπαρκές για την υποστήριξη των μελλοντικών διαδικτυακών υπηρεσιών που θα παρέχονται στα συνδρομητικά δίκτυα. Αυτές συνοψίζονται στα ακόλουθα:

(i) στη μειωμένη δυνατότητα μοντελοποίησης των σύγχρονων συσκευών, (ii) στη χρήση του αναξιόπιστου UDP πρωτοκόλλου για τη μεταφορά της διαχειριστικής πληροφορίας, (iii) στην απουσία υποστήριξης για ολοκληρωμένη διαχείριση και (iv) στην ασφάλεια.

2.3.1. Η αδυναμία της SNMP μοντελοποίησης

Το SNMP πρότυπο ορίζει την πληροφορία διαχείρισης χρησιμοποιώντας το SMI (Structure of Management Information) μεταμοντέλο [1], του οποίου η ικανότητα μοντελοποίησης περιορίζεται σε απλούς πίνακες με μονόμετρους τύπους δεδομένων. Από την άλλη πλευρά οι σύγχρονες και πολύπλοκες συνδρομητικές συσκευές απαιτούν μια συστηματικότερη δόμηση του μεγάλου όγκου της διαχειριστικής πληροφορίας τους. Επίσης η απουσία υψηλού επιπέδου διαχειριστικών βάσεων (Management Information Bases - MIBs) σε συνδυασμό με τη δεδομενοστραφή φύση της SNMP μοντελοποίησης [2] και το περιορισμένο πλήθος διαχειριστικών πράξεων αναγκάζουν τους προγραμματιστές των διαχειριστικών εφαρμογών να γράφουν κώδικα χαμηλού επιπέδου αντί του αντικειμενοστραφούς κώδικα. Για να γίνει κατανοητό αυτό δίνεται το ακόλουθο παράδειγμα: για την εκκίνηση μιας ενέργειας σε ένα απομακρυσμένο πράκτορα, ο διαχειριστής θα πρέπει να θέσει (αφού οι κατηγορίες των πράξεων που ορίζονται είναι μόνο της μορφής set, get και notify) μια μεταβλητή κατάστασης της MIB (π.χ. doIT) σε μια συγκεκριμένη τιμή. Διαφορετικές τιμές εκκινούν και διαφορετικές πράξεις (π.χ. εάν η doIT έχει τιμή 50 σημαίνει επανεκκίνηση του μηχανήματος, ενώ εάν έχει τιμή 70 σημαίνει επανεκκίνηση μιας δικτυακής διεπαφής). Αυτός ο πολύ φτωχός σχεδιασμός συχνά οδηγεί σε "spaghetti"κώδικα εντός των εφαρμογών διαχείρισης διότι οι προγραμματιστές δεν χρησιμοποιούν βιβλιοθήκες περιτυλίγματος (wrapper libraries) αλλά αντίθετα χρησιμοποιούν τις διαφορετικές αυτές τιμές (50, 70) αυτούσιες μέσα στον κώδικα (hard-coded). Το γεγονός αυτό καθιστά δυσχεραίνει σημαντικά τη διαρκή συντήρηση των εφαρμογών διαχείρισης. Άλλωστε στη μηχανική λογισμικού είναι ευρέως γνωστό ότι οι φτωχά σχεδιασμένες προγραμματιστικές διεπαφές (Application Programming Interfaces - APIs) οδηγούν και σε χαμηλής ποιότητας σχεδιασμό των εφαρμογών που τις χρησιμοποιούν.

Μάλιστα με τον τρόπο αυτό (programming by side effect) [2] δυσχεραίνει η ανάπτυξη έξυπνων διαχειριστικών εφαρμογών και αυξάνεται σημαντικά το κόστος υλοποίησης και συντήρησής τους. Είναι αξιοσημείωτο το γεγονός ότι παρόλο που η αντικειμενοστραφής μοντελοποίηση είχε ήδη κάνει την εμφάνισή της από τις αρχές του 1990, ωστόσο δεν επιλέχτηκε από τους σχεδιαστές του SNMPv1. Το 1999 και αφού είχε προηγηθεί μια ακόμα έκδοση της SMI προτάθηκε μια πιο ολοκληρωμένη αντικειμενοστραφής έκδοση η SMIng [3]. Ωστόσο μέχρι σήμερα η έκδοση αυτή έχει περιοριστεί στην ερευνητική κοινότητα και δεν έχει γίνει ακόμη αποδεκτή από την αγορά του SNMP.

2.3.2. Η χρήση του αναξιόπιστου UDP πρωτοκόλλου

Το SNMP χρησιμοποιεί κατά κύριο λόγο το πρωτόκολλο UDP για τη μεταφορά της διαχειριστικής πληροφορίας (η TCP αντιστοίχιση του RFC 3430 δεν χρησιμοποιείται ευρέως). Το γεγονός αυτό έχει ορισμένα σημαντικά επακόλουθα. Καταρχάς το μέγιστο μέγεθος του κάθε SNMP μηνύματος τυπικά περιορίζεται στο MTU (Maximum Transfer Unit) μέγεθος του

εκάστοτε δικτύου λαμβάνοντας τιμές από 482 μέχρι και 1472 bytes [2][4]. Αυτό συμβαίνει προκειμένου να αποφευχθεί ο IP τεμαχισμός, ο οποίος αυξάνει την πιθανότητα απώλειας των SNMP πακέτων. Το γεγονός αυτό καθιστά ιδιαίτερα αναποτελεσματική τη μεταφορά μεγάλου όγκου διαχειριστικών δεδομένων και μπορεί να αυξήσει σημαντικά το φορτίο του δικτύου, εξαιτίας των πολλών και διαδοχικών ερωταπαντήσεων. Επίσης η χρήση του UDP δημιουργεί υπερκεφαλισμούς επεξεργασίας στο διαχειριστή, αφού η ίδια η εφαρμογή θα πρέπει να ελέγχει και να ξαναστέλνει τα μηνύματα αίτησης στο πράκτορα σε περίπτωση που αυτά χάνονται εξαιτίας μιας υπερχείλισης στον IP δρομολογητή.

Τέλος, η αναξιοπιστία του UDP πρωτοκόλλου είναι δυνατόν να οδηγήσει σε χαμένες ενημερώσεις συμβάντων (notifications) αφού ο πράκτορας μπορεί να στείλει ένα SNMP-Trap μήνυμα το οποίο όμως μπορεί να χαθεί κατά τη μετάδοση και έτσι να μην ληφθεί ποτέ από το διαχειριστή. Αυτό τουλάχιστον ίσχυε στην πρώτη έκδοση του SNMP (SNMPv1), ενώ στις νεότερες εκδόσεις έχει διορθωθεί με τα αξιόπιστα inform-requests μηνύματα, σύμφωνα με τα οποία ο διαχειριστής στέλνει πάντα μια επιβεβαίωση στον πράκτορα, ότι έλαβε το SNMP-Trap μήνυμα, διαφορετικά ο πράκτορας το ξαναστέλνει. Παρόλο που οι νεότερες εκδόσεις του SNMP υποστηρίζουν πλέον την αξιόπιστη ενημέρωση συμβάντων, στην πράξη η διαχείριση δεν γίνεται με βάση τα συμβάντα αλλά με τη μέθοδο των περιοδικών ερωταπαντήσεων (polling). Το γεγονός αυτό δημιουργεί υπερφόρτωση τόσο στο δίκτυο όσο και στην πλευρά του διαχειριστή, καθώς ο τελευταίος θα πρέπει να ερωτά ένα μεγάλο αριθμό συσκευών ανά τακτά χρονικά διαστήματα. Βέβαια αυτός ο τρόπος διαχείρισης δεν οφείλεται στο SNMP αλλά στην τακτική των περισσοτέρων κατασκευαστών να μην εξοπλίζουν τους SNMP πράκτορές τους με μεθόδους παρακολούθησης και αναφοράς συμβάντων. Αρκεί να αναφερθεί το γεγονός ότι το 1996 οι Cisco IP δρομολογητές μπορούσαν να στείλουν ενημερώσεις μόνο για 7 μεταβλητές [2] (IOS 10.3 – Internet Operating System – το ιδιόκτητο λειτουργικό σύστημα της Cisco), ενώ η ίδια κατάσταση παρέμεινε και το 2000 (IOS 12.0). Βέβαια η κατάσταση αυτή μπορεί να ήταν ανεκτή για τη διαχείριση των λιγοστών σχετικά στοιχείων του δικτύου πρόσβασης και του δικτύου κορμού αλλά σίγουρα θα πρέπει να αλλάξει για τη διαχείριση του μεγάλου πλήθους των οικιακών δικτυακών πυλών και του ακόμα μεγαλύτερου πλήθους των συνδρομητικών τερματικών συσκευών.

2.3.3. Η απουσία της ολοκληρωμένης διαχείρισης

Όπως φάνηκε και στην εισαγωγή αυτού του κεφαλαίου, μια πολύ σημαντική απαίτηση για την υποστήριξη των μελλοντικών διαδικτυακών υπηρεσιών είναι η ολοκληρωμένη διαχείριση (Integrated Management). Σύμφωνα με την εργασία [2] η ολοκλήρωση αυτή θα μπορούσε να πραγματοποιηθεί σε τρία στάδια. Καταρχάς θα πρέπει να γίνει η ολοκλήρωση της διαχείρισης των συστημάτων για όλες τις αγορές: δικτυακές συσκευές, Windows και Linux PCs, Unix σταθμούς εργασίας και γενικότερα για όλο το σύνολο των συνδρομητικών συσκευών που υπάρχουν ή πρόκειται να εμφανιστούν στο μέλλον. Σήμερα η διαχείριση των συστημάτων είναι σε μεγάλο βαθμό ιδιόκτητη κυρίως για τα Windows PCs. Σήμερα που η διαχείριση των δικτύων έχει ανοίξει επιτυχώς το δρόμο με το SNMP, θα έπρεπε η διαχείριση των συστημάτων να μετακινηθεί προς ανοιχτές τεχνολογίες και αρχιτεκτονικές διαχείρισης. Η

απομακρυσμένη διαχείριση των συστημάτων προβλέπεται να μειώσει δραματικά το κόστος των κέντρων υποστήριξης πελατών (Help Desks) αφού θα επιτρέπει την απομακρυσμένη διάγνωση και πολλές φορές την διόρθωση των δυσλειτουργιών στις συνδρομητικές συσκευές. Σε δεύτερη φάση η διαχείριση των δικτύων θα πρέπει να ολοκληρωθεί με εκείνη των συστημάτων. Η ολοκλήρωση αυτή όχι μόνο θα μειώσει δραματικά το κόστος λειτουργίας των κέντρων χειρισμού των δικτύων (Network Operations Center - NOC) αλλά θα επιτρέψει και την ακριβέστερη συσχέτιση συμβάντων. Όταν η διαχείριση των συστημάτων είναι αποσυνδεδεμένη από τη διαχείριση των δικτύων τότε αφήνεται στον ανθρώπινο παράγοντα η συσχέτιση των δικτυακών προβλημάτων (π.χ. μια δυσλειτουργία της δικτυακής πύλης κατά τη δρομολόγηση) με τα προβλήματα των συστημάτων (π.χ. το απρόσιτο ενός VoIP τηλεφώνου). Τέλος οι παραπάνω τύποι διαχείρισης θα πρέπει να ολοκληρωθούν με τη διαχείριση των πολιτικών και των εφαρμογών προκειμένου να υποστηριχθούν πλήρως οι υπηρεσίες των παρόχων προς τα συνδρομητικά δίκτυα. Η λογική πίσω από αυτό είναι ότι η πλήρης ολοκλήρωση των διαχειριστικών λειτουργιών θα βοηθήσει σημαντικά τους διαχειριστές στην αυτοματοποίηση της διαχείρισης, στην παροχή υπηρεσιών που μέχρι σήμερα είναι ανέφικτο να αξιοποιηθούν εμπορικά και τέλος στη μετάβαση από μια αντενεργητική διαχείριση σε μια προληπτική διαχείριση.

Το SNMP πρότυπο (SNMPv1) αρχικά σχεδιάστηκε για την διαχείριση των δικτύων και όχι για την ολοκληρωμένη διαχείριση συστημάτων, υπηρεσιών και πολιτικών. Παρόλο που δεν υπήρχαν κάποιο εσωτερικοί περιορισμοί στην SNMP αρχιτεκτονική, οι ομάδες εργασίας της IETF δεν έδειξαν κανένα ενδιαφέρον για την ολοκληρωμένη διαχείριση κατά το σχεδιασμό των μεταγενέστερων εκδόσεων του SNMP. Σαν αποτέλεσμα αυτού ήταν πολλές εταιρείες να χρησιμοποιούν διαφορετικές πλατφόρμες διαχείρισης από διαφορετικούς κατασκευαστές, παράλληλα και ανεξάρτητα. Έτσι μέχρι και σήμερα στα IP δίκτυα δεν υπάρχει κάποιος προτυποποιημένος και ανοιχτός τρόπος ολοκληρωμένης διαχείρισης.

2.3.4. Η ασφάλεια

Όταν σχεδιάζονταν η πρώτη έκδοση του SNMP (SNMPv1) οι πάροχοι αλλά και οι περισσότερες εταιρείες δεν ενδιαφέρονταν για την ασφάλεια του δικτύου και των συστημάτων, διότι το διαδίκτυο δεν ήταν αυτό που είναι σήμερα. Κατά τη δεκαετία του 2000, ιδιαίτερα μετά τις επιθέσεις σε πολύ γνωστές ιστοσελίδες και τη διάδοση των υιών μέσω e-mails, σημαντικές ανησυχίες εγέρθηκαν σχετικά με την ασφάλεια των δικτύων. Το γεγονός αυτό οδήγησε σταδιακά στη σημερινή κατάσταση, όπου τα τείχη προστασίας (firewalls) είναι κάτι πολύ κοινό, ενώ οι πελάτες γίνονται ολοένα και πιο απαιτητικοί σχετικά με την ασφάλεια. Τα προβλήματα ασφάλειας του SNMP μπορούν να διακριθούν σε δύο κατηγορίες: (i) την έλλειψη υποστήριξης απλής και διαφανούς ασφάλειας και (ii) την αδυναμία διάβασης του SNMP μέσα από τα τείχη προστασίας.

2.3.4.1. Απλή ασφάλεια

Οι πρώτες εκδόσεις του SNMP (SNMPv1 και SNMPv2c) δεν υποστηρίζουν ασφάλεια. Η αναγνώριση που βασίζεται στη σειρά χαρακτήρων κοινότητας (community string) είναι πολύ

απλοϊκή και δεν μπορεί να προσμετρηθεί σαν σχήμα ασφάλειας. Δεν υπάρχει πιστοποίηση ενώ η σειρά χαρακτήρων κοινότητας μεταδίδεται αυτούσια στο δίκτυο. Αντίθετα η τρίτη και τελευταία έκδοση του SNMP (SNMPv3) εστίασε στην ασφάλεια. Έτσι το SNMPv3 υποστηρίζει την αναγνώριση (identification), την πιστοποίηση (authentication), την κρυπτογράφηση (encryption), την αρτιότητα (integrity), τον έλεγχο πρόσβασης κ.α. Ωστόσο το είδος ασφάλειας που προτάθηκε απαιτεί σημαντικές ρυθμίσεις (κλειδιά, VACM (View-based Access Control Model) πίνακες [5] κ.α.) και σε καμία περίπτωση δεν είναι διάφανο στον τελικό χρήστη. Η εργασία [2] διακρίνει τα παρακάτω είδη ασφάλειας για τη διαχείριση: (i) την απώλεια ασφάλειας (no security), (ii) την απλή ασφάλεια (simple security) όπως είναι εκείνη του SSL, που είναι διάφανη προς το χρήστη, (iii) τη μέση ασφάλεια (medium security) που περιλαμβάνει την πολύπλοκη και έξυπνη διαχείριση των κλειδιών και είναι επαρκής για τις περισσότερες περιπτώσεις και (iv) την ισχυρή ασφάλεια (strong security) που απαιτεί επιπρόσθετο και ακριβό υλικό κρυπτογράφησης, ενώ έχει εφαρμογή μόνο σε ειδικές περιπτώσεις (π.χ. στρατός, τράπεζες).

Στην κρυπτογραφία η ασφάλεια του SNMPv3 θεωρείται ασθενής διότι υπάρχουν τρόποι για να σπάσει. Ωστόσο σύμφωνα με την παραπάνω ταξινόμηση η ασφάλεια του SNMPv3 μπορεί να τοποθετηθεί στο μεσαίο επίπεδο, διότι μπορεί να υπάρχουν γνωστοί τρόποι που επιτρέπουν την κλοπή των κλειδιών [99], αλλά από την άλλη μεριά η ασφάλεια που προσφέρεται είναι επαρκής για τους περισσότερους πελάτες. Άλλωστε στη διαχείριση ο περισσότερος κόσμος ενδιαφέρεται για το δεύτερο και τρίτο επίπεδο ασφάλειας, ενώ όπως εξηγείται και στο [2] το τέταρτο επίπεδο δεν θα έπρεπε να χρησιμοποιηθεί στη διαχείριση. Επομένως η αδυναμία του SNMP έγκειται στην έλλειψη της απλής ασφάλειας. Το γεγονός αυτό δημιουργεί ένα πραγματικό πρόβλημα διότι οι περισσότερες εταιρείες (και κυρίως οι μικρομεσαίες) θέλουν την απλή και διάφανη ασφάλεια. Επιθυμούν απλά οι κωδικοί τους να μην εκτίθενται στο διαδίκτυο (όπως στο SNMPv1 και SNMPv2c), ενώ παράλληλα η αναγνώριση και η πιστοποίηση να είναι απλές και διάφανες όπως συμβαίνει με το SSL. Ορισμένοι μάλιστα θα ήθελαν και ένα είδος κρυπτογράφησης έτσι ώστε μια συνήθης επίθεση να μην έχει πρόσβαση σε ευαίσθητα δεδομένα (π.χ. σε εκείνα για τα οποία ο πελάτης πληρώνει για μια «χρυσή» υπηρεσία). Επίσης οι περισσότεροι πελάτες ελάχιστα θα ενδιαφερθούν για το εάν η εμπιστευτικότητα των διαχειριστικών τους δεδομένων δεν είναι εγγυημένη και δεν έχουν πρόβλημα εάν μια έξυπνη επίθεση καταφέρει να κρυφακούσει τα διαχειριστικά δεδομένα του συνδρομητικού τους δικτύου που εκτίθενται στο δημόσιο δίκτυο.

2.3.4.2. Τείχη προστασίας

Οι σχεδιαστές του SNMP σε καμία έκδοσή του δεν διευθετούν το πρόβλημα με τα τείχη προστασίας (firewalls). Τα περισσότερα τείχη προστασίας δεν επιτρέπουν στην SNMP κίνηση να περάσει, με αποτέλεσμα το δίκτυο που βρίσκεται πίσω από αυτά να μην είναι προσπελάσιμο από το διαχειριστή. Το γεγονός αυτό έχει ιδιαίτερη σημασία για τις συνδρομητικές συσκευές διότι η δικτυακή πύλη αναλαμβάνει να επιτελέσει τις λειτουργίες του τείχους προστασίας με αποτέλεσμα αρχικά μόνο εκείνη να είναι προσβάσιμη στο διαχειριστή. Προκειμένου να είναι προσπελάσιμες και οι υπόλοιπες συνδρομητικές συσκευές

στο διαχειριστή απαιτείται ο τελευταίος να έχει καταρχάς πρόσβαση στην πύλη και κατά δεύτερον να τη ρυθμίσει κατάλληλα, ώστε εκείνη να επιτρέψει την SNMP κίνηση προς τα τερματικά. Ο περιορισμός αυτός δυσχεραίνει και πολυπλέκει ακόμα περισσότερο τη διαχειριστική διαδικασία.

2.4. Οι υποψήφιες τεχνολογίες διαχείρισης

Μετά το δεύτερο μισό της δεκαετίας 1990-2000 και αφού ήδη είχαν φανεί οι πρώτες σοβαρές αδυναμίες του SNMP, πολύς κόσμος προερχόμενος από την κοινότητα της μηχανικής λογισμικού άρχισε να μελετά τον τρόπο με τον οποίο γίνονταν η διαχείριση των IP δικτύων μέχρι εκείνη τη στιγμή. Άλλοι προέρχονταν από τον αντικειμενοστραφή κόσμο, άλλοι από τον κόσμο της τεχνητής νοημοσύνης, ενώ άλλοι πάλι από τον κόσμο των βάσεων δεδομένων. Οι περισσότεροι εξ αυτών δεν είχαν προηγούμενη εμπειρία στη διαχείριση των δικτύων και των συστημάτων και κοίταξαν το πρόβλημα απλά και μόνο με τα μάτια της μηχανικής λογισμικού. Έτσι λοιπόν όταν εξέτασαν τον τρόπο με τον οποίο λειτουργούσε το SNMP διαπίστωσαν ότι οι μοντέρνες κατανεμημένες εφαρμογές προσέφεραν τόσο μεγάλα πλεονεκτήματα που το SNMP δεν θα μπορούσε να αντέξει στη σύγκριση. Η κοινότητα της μηχανικής λογισμικού είχε πολλές εναλλακτικές λύσεις να προσφέρει: τους κινητούς πράκτορες, τις τεχνολογίες ιστού, τα ενεργά δίκτυα, την CORBA, την Java RMI, τους έξυπνους πράκτορες κ.α. Μερικές από τις τεχνολογίες αυτές κατάφεραν τελικά να βρουν εφαρμογή σε ορισμένες αγορές της διαχείρισης, ενώ άλλες πάλι είναι ακόμη αρκετά ανώριμες για την εμπορική τους εκμετάλλευση.

Η παρούσα ενότητα επιχειρεί να δώσει τις σημαντικότερες προσπάθειες που έγιναν μέχρι σήμερα για τη διαχείριση των δικτύων, των συστημάτων και των υπηρεσιών και εξηγεί συνοπτικά τους λόγους για τους οποίους οι αντίστοιχες τεχνολογίες δεν θα μπορούσαν να αποτελέσουν μια κοινή λύση στη διαχείριση των συνδρομητικών δικτύων, τουλάχιστον για την επόμενη δεκαετία.

2.4.1. Η τεχνολογία OSI/TMN

Η διαχείριση συστημάτων OSI-SM (Open Systems Interconnection - Systems Management) ήταν η πρώτη τεχνολογία διαχείρισης που προτυποποιήθηκε [6]. Η OSI-SM χρησιμοποιεί ένα πλήρες αντικειμενοστραφές μοντέλο πληροφορίας το οποίο υποστηρίζει την κληρονομικότητα. Οι κλάσεις των διαχειριστικών αντικειμένων προδιαγράφονται με την GDMO (Guidelines for the Definition of Managed Objects) γλώσσα. Μάλιστα η OSI-SM ορίζει το CMIP/S (Common Management Information Protocol/Services), ένα πρωτόκολλο με πλούσιες λειτουργίες (Get, Set, Action, Create, Delete και Event). Η υποστήριξη τόσο της μαζικής (μέσω του *scoping*) όσο και της επιλεκτικής (μέσω του *filtering*) ανάκτησης των δεδομένων αποτελεί ένα πολύ σημαντικό πλεονέκτημα για τη διαχείριση. Γενικότερα πρόκειται για μια έξυπνη και ισχυρή τεχνολογία, η οποία όμως εξαιτίας της πολυπλοκότητας και του μεγάλου κόστους υλοποίησής της βρήκε εφαρμογή μόνο στα τηλεπικοινωνιακά περιβάλλοντα. Ακόμη και το ήδη επιτυχημένο πρωτόκολλο SNMP αποτελεί στη ουσία μια απλοποίηση του OSI-SM [7]. Επίσης πάνω στο OSI-SM βασίστηκε και η τεχνολογία των

TMN (Telecommunications Management Network) Q/X τηλεπικοινωνιακών διεπαφών [8][9]. Ωστόσο ακόμη και σε αυτό το περιβάλλον το OSI-SM παραμερίστηκε σταδιακά για χάρη της CORBA.

2.4.2. Η εφαρμογή της CORBA στη διαχείριση

Προκειμένου να λύσει το πρόβλημα της διαλειτουργικότητας στον αντικειμενοστραφή κόσμο, η ομάδα διαχείρισης αντικειμένων (Object Management Group - OMG) προτυποποίησε την αρχιτεκτονική διαχείρισης αντικειμένων, η οποία είναι γνωστή με το όνομα CORBA (Common Object Request Broker Architecture) [10]. Μάλιστα η δεύτερη έκδοσή της [11] ήταν αυτή που σημείωσε τη μεγάλη επιτυχία στην κοινότητα της μηχανικής λογισμικού και ιδιαίτερα για τις μεγάλες επιχειρήσεις που είχαν κάνει τεράστιες επενδύσεις σε παρωχημένα συστήματα. Το γεγονός ότι η CORBA είναι αντικειμενοστραφής και επομένως τα τμήματα της SNMP διαχειριστικής βάσης (Management Information Base - MIB) μπορούσαν εύκολα να αντιστοιχιστούν σε αντικείμενα, οδήγησε στην έναρξη των εργασιών για την ενσωμάτωση της CORBA στις υπάρχουσες αρχιτεκτονικές διαχείρισης. Ο καθηγητής Παύλου ήταν από τους πρώτους που πρότειναν την CORBA σαν τη βασική τεχνολογία της TMN αρχιτεκτονικής [12][13].

Μάλιστα η JIDM (Joint Inter-Domain Management) ομάδα υποστηριζόμενη από το NMF (Network Management Forum το οποίο στη συνέχεια μετονομάστηκε σε TeleManagement Forum - TMF) φόρουμ συστάθηκε με κύριο στόχο την κατασκευή εργαλείων που θα βοηθούσαν στη διαλειτουργικότητα μεταξύ CMIP (OSI-SM), SNMP και CORBA. Η SNMP/CMIP διαλειτουργία είχε διευθετηθεί προηγουμένως από την IIMC (ISO-Internet Management Coexistence) ομάδα του NMF, η οποία προδιέγραψε τη μετάφραση μεταξύ των SNMP και των CMIP/CMIS πρωτοκόλλων, υπηρεσιών και πληροφορίας. Η JIDM ομάδα ανέλαβε τις CMIP/CORBA και SNMP/CORBA διαλειτουργίες [14] μεταφράζοντας τις προδιαγραφές αλλά και τη διάδραση μεταξύ τους. Μάλιστα αναπτύχθηκαν αλγόριθμοι για την αντιστοίχιση ανάμεσα στη GDMO/ASN.1 (Abstract Syntax Notation 1) και στην CORBA IDL (Interface Definition Language) [15]. Οι JIDM αντιστοιχίσεις επέτρεψαν στους CORBA προγραμματιστές να γράφουν OSI ή SNMP διαχειριστές και πράκτορες χωρίς καμία γνώση του GDMO, του ASN.1 ή του CMIP. Οι αντιστοιχίσεις αυτές λειτουργούν και προς την αντίθετη κατεύθυνση επιτρέποντας στους CMIS ή SNMP προγραμματιστές να προσπελαύνουν IDL πόρους, υπηρεσίες και εφαρμογές χωρίς να γνωρίζουν τίποτα για την IDL.

Η CORBA έχει γίνει ευρέως αποδεκτή στον κόσμο των τηλεπικοινωνιών με αποτέλεσμα να γίνεται ένα εκ των πραγμάτων πρότυπο (*de facto standard*), πράγμα σπάνιο για τη συγκεκριμένη βιομηχανία, η οποία παραδοσιακά βασίζεται σε επικυρωμένα πρότυπα (*de jure standards*). Μάλιστα η CORBA επιλέχθηκε αμέσως από την TINA (Telecommunication Information Networking Architecture) [16-19] για την προσθήκη ενός περιβάλλοντος κατανεμημένης επεξεργασίας στην TMN αρχιτεκτονική. Έτσι σήμερα πολλοί κατασκευαστές τηλεπικοινωνιακού εξοπλισμού στρέφονται προς στην CORBA για να διαχειριστούν τους τηλεφωνικούς μεταγωγείς και τα δίκτυά τους.

2.4.3. Οι τεχνολογίες κινητού κώδικα στη διαχείριση

Υπάρχουν αρκετές διαφορετικές τεχνολογίες κινητού κώδικα, οι οποίες ωστόσο μοιράζονται μια κοινή ιδέα: για χάρη της ευελιξίας θα μπορούσε κανείς να μεταφέρει δυναμικά τα προγράμματα στους πράκτορες προκειμένου εκείνοι να τα εκτελούν. Η μεταφορά και εκτέλεση του προγράμματος μπορεί να σκανδαλιστεί είτε από τον ίδιο τον πράκτορα είτε από ένα διαχειριστή ή ακόμα και από έναν άλλο πράκτορα. Η εργασία [20] κάνει μια λεπτομερή θεώρηση της έννοιας του κινητού κώδικα και ορίζει την έννοια της ισχυρής (strong mobility) και της ασθενούς (weak mobility) κινητικότητας. Σαν ισχυρή κινητικότητα ορίζεται η δυνατότητα που παρέχει το σύστημα κινητού κώδικα (Mobile Code System - MCS) σε μια μονάδα εκτέλεσης (π.χ. σε μια Unix διεργασία) να μετακινεί σε ένα άλλο μηχάνημα τόσο τον κώδικά όσο και την κατάσταση εκτέλεσής της (execution state). Αντίθετα στην ασθενή κινητικότητα το σύστημα κινητού κώδικα επιτρέπει μεν στη μονάδα εκτέλεσης να συνδέει δυναμικά τον κώδικα που έρχεται από κάποιο άλλο μηχάνημα αλλά δεν διασώζει αυτόματα την κατάσταση εκτέλεσης. Αφού η εργασία [20] μελετά όλα τα υπάρχοντα συστήματα κινητού κώδικα διακρίνει τρεις διαφορετικούς τύπους προτύπων:

- Την απομακρυσμένη αποτίμηση (Remote Evaluation – REV [21]), κατά την οποία όταν ο πελάτης επικαλείται μια υπηρεσία σε ένα εξυπηρετητή δεν στέλνει μόνο το όνομα της υπηρεσίας και τα ορίσματα εισόδου αλλά και τον ίδιο τον κώδικα. Με αυτό τον τρόπο ο πελάτης κατέχει τον κώδικα για την υλοποίηση της υπηρεσίας, ενώ ο εξυπηρετητής κατέχει τους πόρους.
- Τον κώδικα κατά απαίτηση (Code On Demand - COD), κατά τον οποίον όταν ένας πελάτης πρέπει να επιτελέσει μια συγκεκριμένη εργασία επικοινωνεί με τον εξυπηρετητή και μεταφορτώνει τον κώδικα από αυτόν. Στη συνέχεια τον συνδέει δυναμικά (dynamic code binding) και στο τέλος τον εκτελεί. Με αυτό τρόπο ο πελάτης κατέχει τους πόρους, ενώ ο εξυπηρετητής τον κώδικα.
- Τον κινητό πράκτορα (Mobile Agent), ο οποίος είναι μια εκτελεστική μονάδα ικανή να μετοικεί σε άλλα μηχανήματα και να συνεχίζει απρόσκοπτα την εκτέλεση. Εδώ ο πελάτης κατέχει τον κώδικα, ενώ ο εξυπηρετητής κατέχει τους πόρους και παρέχει ένα περιβάλλον για να εκτελείται ο κώδικας που στέλνεται είτε από τον πελάτη, είτε από έναν άλλο εξυπηρετητή.

Τα παραπάνω πρότυπα υλοποιούνται από ένα πλήθος τεχνολογιών. Ορισμένες από αυτές είναι απλές γλώσσες προγραμματισμού, ενώ άλλες είναι ολοκληρωμένα συστήματα που τις περισσότερες φορές περιλαμβάνουν μια εικονική μηχανή (virtual machine). Ορισμένα παραδείγματα τεχνολογιών ισχυρής κινητικότητας κώδικα είναι τα ακόλουθα: Agent Tcl, Ara, Emerald, Sumatra, Telescript και Tycoon. Παραδείγματα ασθενούς κινητικότητας κώδικα είναι το Facile, η Java (και συγκεκριμένα έναν φυλλομετρητής ιστού, ο οποίος περιλαμβάνει μια Java εικονική μηχανή για την υποστήριξη Java applets), το M0, το Mole η Obliq και η Tacoma (υπάρχουν αρκετές αναφορές για όλα αυτά στο [20]).

2.4.3.1. Η διαχείριση με εξουσιοδότηση και οι παραλλαγές της

Τα πρότυπα κινητού κώδικα χρησιμοποιήθηκαν για πρώτη φορά στη διαχείριση από τους Goldszmidt και Yemini το 1991, όταν επινόησαν την εξουσιοδότηση πράκτορα-διαχειριστή [22][23]. Η αρχιτεκτονική αυτή βελτιώθηκε και μετονομάστηκε σε διαχείριση με εξουσιοδότηση (Management by Delegation - MbD) το 1995 [23]. Το MbD είναι μια μίξη του REV προτύπου (για την αποστολή εξουσιοδοτημένων πρακτόρων σε εξυπηρετητές) και του υποδείγματος πελάτη-εξυπηρετητή (για τον απομακρυσμένο έλεγχο του χρονοπρογραμματισμού και της εκτέλεσης των εξουσιοδοτημένων πρακτόρων). Επίσης οι Burns και Quinn [24] ήταν οι πρώτοι που περιέγραψαν ένα πρωτότυπο διαχείρισης του δικτύου και των συστημάτων με τη χρήση του κινητού κώδικα. Από τότε τα συστήματα κινητού κώδικα έχουν γνωρίσει μεγάλη επιτυχία στη διαχείριση [25-29]. Οι Baldi και Picco [30] μελέτησαν τη δικτυακή κίνηση που επιφέρει ο κινητός κώδικας υλοποιώντας και τα τρία παραπάνω είδη προτύπων. Μάλιστα έκαναν και μια θεωρητική ποιοτική αποτίμηση της αποτελεσματικότητας και της εφαρμοσιμότητας του κινητού κώδικα στη διαχείριση των δικτύων. Ο οργανισμός ISO ενσωμάτωσε ιδέες του κινητού κώδικα στην OSI αρχιτεκτονική διαχείρισης προδιαγράφοντας μια νέα διαχειριστική λειτουργία με το όνομα Command Sequencer [31]. Τέλος, το 1999 η DISMAN (DIStributed MANagement) ομάδα εργασίας της IETF όρισε μια σειρά από διαχειριστικές βάσεις (MIBs) που επιτρέπουν στους διαχειριστές να κατανέμουν καθήκοντα στους SNMPv3 πράκτορες. Χαρακτηριστική είναι η Script MIB [32-33], η οποία επιτρέπει σε ένα διαχειριστή να εξουσιοδοτήσει ένα πράκτορα με την μορφή κειμένων (scripts).

2.4.3.2. Ενεργά δίκτυα

Μια περιοχή όπου ο κινητός κώδικας έχει γνωρίσει μεγάλη επιτυχία τα τελευταία χρόνια είναι εκείνη των ενεργών δικτύων (active networks). Συγκεκριμένα υπάρχει η εξελικτική και η ριζοσπαστική προσέγγιση για τα ενεργά δίκτυα. Η εξελικτική προσέγγιση ονομάζεται *προγραμματιζόμενος μεταγωγέας* (programmable switch) ή *ενεργός κόμβος* (active node) και παρέχει ένα μηχανισμό για την εισαγωγή προγραμμάτων σε δικτυακούς κόμβους, προκειμένου να επιτευχθεί ο δυναμικός προγραμματισμός τους. Τα προγράμματα αυτά θα μπορούν να επιτελούν ειδικούς υπολογισμούς στα πακέτα που περνούν από τους κόμβους και πιθανόν να αλλάζουν ακόμα και το περιεχόμενο των ίδιων των πακέτων (για παράδειγμα θα μπορούσαν ανάλογα με ορισμένες συνθήκες να συμπιέζουν και να αποσυμπιέζουν το περιεχόμενο των πακέτων στα άκρα του δικτύου). Από την άλλη πλευρά η ριζοσπαστική προσέγγιση είναι γνωστή με το όνομα *κάψουλα* (capsule) ή *έξυπνο πακέτο* (smart packet) και εισάγει την έννοια των προγραμμάτων υπό μορφή πακέτων. Τα προγράμματα αυτά ενθυλακώνονται στα πλαίσια αποστολής των πακέτων (transmission frames) και εκτελούνται σε κάθε κόμβο κατά μήκος της διαδρομής τους [34].

Η ιδέα των ενεργών δικτύων προτάθηκε για πρώτη φορά το 1995 από τους Tennenhouse και Wetherall [34], ενώ η πρώτη εφαρμογή τους στη διαχείριση έγινε ένα χρόνο αργότερα από τους Yemini και daSilva [35]. Σήμερα η έρευνα στο πεδίο αυτό είναι εξαιρετικά ενεργή [36-42] αφού τα ενεργά δίκτυα μπορούν να φέρουν μεγάλη ευελιξία και ευρωστία στη διαχείριση.

Η παρακολούθηση του δικτύου και το φιλτράρισμα των συμβάντων [42-43] είναι κατάλληλοι υποψήφιοι αφού τα προγράμματα παρακολούθησης μπορούν εύκολα να αποστέλλονται μέσα από το δίκτυο. Τα προγράμματα αυτά είναι υψηλού επιπέδου φίλτρα τα οποία εποπτεύουν και ενορχηστρώνουν τις ροές των πακέτων σε πραγματικό χρόνο. Διατηρούν μετρητές και αναφέρουν τα αποτελέσματα στους διαχειριστές τους. Ένα άλλο παράδειγμα είναι ο έλεγχος της συμφόρησης [35]. Ωστόσο ακόμα τα ενεργά δίκτυα υποφέρουν από προβλήματα που σχετίζονται με την ασφάλεια, τις επιδόσεις και τη διαλειτουργικότητα [38]. Τα προβλήματα αυτά είναι τα κυριότερα εκ των οποίων προσπαθεί να λύσει η ερευνητική κοινότητα σήμερα.

2.4.4. Οι τεχνολογίες έξυπνων πρακτόρων στη διαχείριση

Αντίθετα με τις προαναφερθείσες τεχνολογίες οι έξυπνοι πράκτορες είναι στοχοστραφείς. Για να γίνει κατανοητό αυτό αρκεί να σκεφτεί κανείς ότι στις προηγούμενες τεχνολογίες οι πράκτορες εκτελούν τις διαχειριστικές ενέργειες χωρίς όμως να γνωρίζουν τον τελικό στόχο του διαχειριστή. Οι διαχειριστές στέλνουν στους πράκτορες το «πώς» και κρατούν το «γιατί» για τον εαυτό τους. Αντίθετα στους έξυπνους πράκτορες οι διαχειριστές στέλνουν το «γιατί» και περιμένουν από τους πράκτορες να επινοήσουν το «πώς». Η τεχνολογία αυτή προέρχεται από τον κόσμο της κατανεμημένης τεχνητής νοημοσύνης (Distributed Artificial Intelligence - DAI) και πιο συγκεκριμένα από τα συστήματα πολλαπλών πρακτόρων (Multi Agent Systems – MASs), όπου η μοντελοποίηση σύνθετων συστημάτων γίνεται με μεγάλες ομάδες έξυπνων πρακτόρων.

Η μελέτη της εφαρμογής των έξυπνων πρακτόρων στη διαχείριση έχει ξεκινήσει εδώ και μια δεκαετία [63-66], ενώ κάποια πρότυπα όπως είναι οι γλώσσες KQML (Knowledge Query and Manipulation Language) [45] και ACL (Agent Communication Language) [46] είναι πολύ δημοφιλείς στην κοινωνία της διαχείρισης. Ωστόσο ένα πολύ σημαντικό πρόβλημα της τεχνολογίας των έξυπνων πρακτόρων παραμένει η μεγάλη πολυπλοκότητα και οι υψηλότατες απαιτήσεις σε πόρους όπως είναι μνήμη, ο επεξεργαστής και το διαθέσιμο εύρος ζώνης [2].

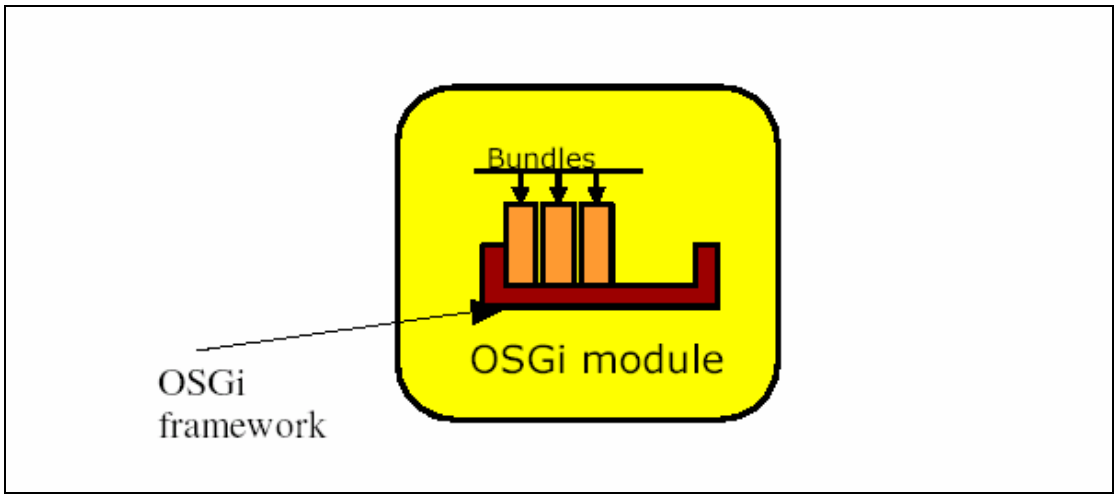
2.4.5. Η Java RMI και η πύλη ανοιχτών υπηρεσιών OSGi

Η Java RMI (Remote Method Invocation) καθιστά εφικτό τον προγραμματισμό μιας διαχειριστικής εφαρμογής σαν κατανεμημένη και αντικειμενοστραφή. Τα πάντα είναι αντικείμενα και όλα τα αντικείμενα αλληλεπιδρούν ακόμη κι αν είναι απομακρυσμένα (δηλαδή σε διαφορετικά μηχανήματα). Τα κατανεμημένα Java αντικείμενα μπορούν να αντιστοιχισθούν απευθείας στα SNMP και στα OSI διαχειριζόμενα αντικείμενα. Στην περίπτωση αυτή τα Java αντικείμενα είναι χαμηλού επιπέδου. Ωστόσο μπορούν να είναι και υψηλού επιπέδου. Για να γίνει κατανοητό αυτό δίνεται το ακόλουθο παράδειγμα. Προκειμένου να εγκατασταθεί ένα εικονικό μονοπάτι (Virtual Path - VP) κατά μήκος πολλαπλών ATM μεταγωγέων μπορεί να οριστεί μια απλή κλήση διεργασίας σε ένα απομακρυσμένο αντικείμενο που βρίσκεται στο μεταγωγέα-πηγή. Το υψηλού επιπέδου αυτό αντικείμενο θα αναλάβει τις επακόλουθες επικλήσεις διεργασιών σε όλους τους μεταγωγείς μέχρι τον προορισμό.

Μέχρι σήμερα η Sun Microsystems και η κοινωνία της Java έχουν δημοσιεύσει 4 προδιαγραφές που χρησιμοποιούν την Java RMI με σκοπό τη διαχείριση: (i) το JMAPI (Java Management Application Programming Interface) [47], (ii) το JMX (Java Management eXtensions) [48], (iii) την FMA (Federated Management Architecture) [49] αρχιτεκτονική και (iv) το OSGi (Open Services Gateway initiative) πλαίσιο εργασίας. Η JMAPI είναι ήδη απαρχαιωμένη αλλά ήταν μια επιτυχής απόδειξη της προτεινόμενης ιδέας. Η JMX εστιάζει στην πλευρά του πράκτορα, ενώ η FMA εστιάζει στην πλευρά του διαχειριστή. Τα δύο αυτά πλαίσια εργασίας αναμένεται να συνενωθούν στο μέλλον.

Η πύλη ανοιχτών υπηρεσιών OSGi είναι η σημαντικότερη προσπάθεια των τελευταίων επτά ετών που στοχεύει στην απομακρυσμένη εγκατάσταση, ενεργοποίηση και διαχείριση των εφαρμογών στις συνδρομητικές συσκευές. Η OSGi συμμαχία ξεκίνησε το 1999 με κύριο στόχο να παράσχει μια συνολική αρχιτεκτονική για τη δυναμική παροχή υπηρεσιών στη νέα εποχή του διαδικτύου. Μάλιστα αποτελείται από πολλές μεγάλες εταιρείες (Sun Microsystems, Nokia, Intel, IBM) και ερευνητικά ιδρύματα (ETRI, ITRI). Ενώ η προτεινόμενη αρχιτεκτονική (και κυρίως οι ρόλοι της κάθε οντότητας του διαδικτύου στην παροχή των υπηρεσιών) [82] είναι ακόμα υπό έντονη αμφισβήτηση από πολλούς παρόχους αλλά και ερευνητές [85]. Ωστόσο η προτεινόμενη πλατφόρμα υπηρεσιών έχει προσεγγίσει ένα υψηλό επίπεδο ωριμότητας (το 2005 εκδόθηκε η τέταρτη έκδοση των προδιαγραφών [83) και έχει προσελκύσει το ενδιαφέρον πολλών ερευνητών, προγραμματιστών, κατασκευαστών αλλά και παρόχων.

Το OSGi ορίζει μια πρότυπη πλατφόρμα, η οποία υπέρκειται μιας Java εικονικής μηχανής και επιτρέπει τη δυναμική (απ)εγκατάσταση, (απ)ενεργοποίηση και ανανέωση Java εφαρμογών που καλούνται *δέματα* (*bundles*). Κάθε *δέμα* μπορεί από μόνο του να μην υλοποιεί μια υπηρεσία αλλά να συνεργάζεται με άλλα παρόμοια, που υπέρκεινται στην ίδια πλατφόρμα, προκειμένου να υλοποιήσουν συλλογικά την τελική υπηρεσία προς το συνδρομητή.



Σχήμα 2-1. Η θέση των δεμάτων λογισμικού (bundles) στην OSGi πλατφόρμα υπηρεσιών.

Στο σημείο αυτό θα πρέπει να σημειωθεί ότι η OSGi πλατφόρμα υπηρεσιών δεν είναι τίποτα άλλο παρά ένα έξυπνο πλαίσιο λογισμικού, το οποίο επιτρέπει σε ένα πράκτορα να συνδέει

δυναμικά και να εκτελεί τον κώδικα (δηλαδή τα *bundles*) που του στέλνει ο διαχειριστής. Μέχρι εδώ θα μπορούσε κανείς να ισχυριστεί ότι πρόκειται καθαρά για μια τεχνολογία κινητού κώδικα, η οποία μάλιστα υπακούει στο υπόδειγμα του «κώδικα κατά απαίτηση» (Code On Demand), όπως αναλύθηκε παραπάνω. Ωστόσο το OSGi δεν ορίζει τον τρόπο με τον οποίο ο διαχειριστής θα επικοινωνεί με τον πράκτορα και θα τον ελέγχει. Για το σκοπό αυτό έχει προταθεί [86] η συμπληρωματική χρήση του JMX. Το JMX δίνει τη δυνατότητα στο διαχειριστή να προσπελαύνει τα απομακρυσμένα αντικείμενα ενός JMX πράκτορα με τη βοήθεια των Java απομακρυσμένων κλήσεων (Java RMIs). Αυτός ο συνδυασμός κινητού κώδικα και κατανεμημένων εφαρμογών φαίνεται να είναι πολλά υποσχόμενος, αφού αποδεικνύει ότι η κατανεμημένη διαχείριση, που βασίζεται στη Java παρέχει ένα πολύ δυνατό τρόπο για να χτίσει κανείς μια εύρωστη εφαρμογή διαχείρισης [87,88].

Ωστόσο το σημαντικό πρόβλημα είναι ότι όλοι οι διαχειριστές και οι πράκτορες θα πρέπει να υποστηρίζουν τη Java ή εναλλακτικά όλοι οι πράκτορες θα πρέπει να προσπελαύνονται μέσω μια Java διαχειριστικής πύλης. Το γεγονός αυτό αποτελεί μια σημαντική απαίτηση διότι δεν είναι εύκολο ο κατασκευαστής μιας συσκευής να πειστεί να πληρώσει την άδεια για τη χρήση της Java και της OSGi πύλης (όταν αυτή είναι εμπορική). Το σημαντικότερο ωστόσο είναι ότι ακόμη και αν αυτό συμβεί, τότε οι απαιτήσεις της Java εικονικής μηχανής σε πόρους (μνήμη, αποθηκευτικά μέσα και επεξεργαστή) είναι για πολλές συνδρομητικές συσκευές (π.χ. δικτυακές πύλες και αποκωδικοποιητές εικόνας και ήχου) σχεδόν απαγορευτικές. Βέβαια ο παράγοντας αυτός συνεχώς θα φθίνει στο μέλλον, καθώς οι τεχνολογίες ολοκλήρωσης συνεχώς εξελίσσονται και οι δυνατότητες ακόμη και των φτηνών συσκευών καθημερινά αυξάνονται. Ωστόσο από τη μέχρι τώρα εμπειρία του συγγραφέα ακόμα και μεγάλες εταιρείες (όπως η Thomson, η ST Microelectronics και η Alcatel) είναι διστακτικές στην υιοθέτηση μιας λύσης βασισμένης στη Java εξαιτίας του υψηλού κόστους που αυτή επιφέρει.

2.4.6. Διαχείριση με χρήση τεχνολογιών ιστού

Η βασιζόμενη στον ιστό διαχείριση (Web-based management) είναι μια έννοια που έχει δημιουργήσει σε πολύ κόσμο σύγχυση καθώς έχει εξελιχτεί σημαντικά τα τελευταία χρόνια. Πολλοί άνθρωποι τη θεωρούν στενά συνδεδεμένη με τη χρήση ενός φυλλομετρητή ιστοσελίδων (Web-browser) για την επιτέλεση των διαχειριστικών διεργασιών. Στην πραγματικότητα αυτή είναι η απλούστερη προσέγγιση που ονομάζεται *διαχείριση βασισμένη στο φυλλομετρητή* (*browser-based management*) [2]. Μάλιστα ο βαθμός της σύγχυσης είναι τόσο μεγάλος, ώστε ο Harler ονόμασε το βιβλίο του "Web-Based Network Management: Beyond the Browser" [50] για να τονίσει ότι στη βασιζόμενη στον ιστό διαχείριση υπάρχει κάτι περισσότερο από τον απλό φυλλομετρητή. Πολλοί επίσης το μπέρδευαν με το WBEM (Web-based Enterprise Management) εξαιτίας της υπερβολικής προβολής του από τις εταιρείες που το προωθούσαν πριν την ανάληψή του από την DMTF (Distributed Management Task Force) ομάδα της IETF. Παρακάτω δίνεται με χρονολογική σειρά μια συνοπτική ανασκόπηση της εξέλιξης της διαχείρισης με τεχνολογίες ιστού.

2.4.6.1. Μεταδιαχείριση βασισμένη στο φυλλομετρητή ιστοσελίδων

Η χρήση των τεχνολογιών ιστού (φυλλομετρητές ιστοσελίδων, HTTP, HTML, CGI) στη διαχείριση χρονολογείται σχεδόν από τη πρώτη ημέρα της εμφάνισής τους [51]. Η πρώτη στοιχειώδης μορφή διαχείρισης με τεχνολογίες ιστού ήταν μετά το 1993 και είχε ως στόχο τις παράπλευρες διαδικασίες της διαχείρισης. Για το λόγο αυτό ονομάζεται μεταδιαχείριση.

Σε πρώτη φάση οι διαχειριστές χρησιμοποίησαν τις HTML και CGI (Common Gateway Interface) τεχνολογίες σε συνδυασμό με απλές βάσεις δεδομένων προκειμένου να αποθηκεύουν, να επεξεργάζονται και να παρουσιάζουν με ένα φιλικό προς το χρήστη τρόπο τη διαχειριστική πληροφορία, που έδιναν οι εμπορικές SNMP πλατφόρμες διαχείρισης. Το γεγονός αυτό βοήθησε σημαντικά στην αυτοματοποίηση της αναφοράς προβλημάτων (online problem reporting), στη γρήγορη επίλυση των προβλημάτων αυτών από τους χειριστές του δικτύου (online management procedures, online documentation και online troubleshooting assistance) καθώς και στη συλλογή των στατιστικών στοιχείων του δικτύου (online usage reports). Στο σημείο αυτό θα πρέπει να σημειωθεί ότι η παραπάνω διαδικασία δεν αυτοματοποίησε μαζί και την επίλυση των ανιχνευθέντων προβλημάτων του δικτύου καθώς οι HTML εφαρμογές δεν είχαν πρόσβαση ούτε στις εμπορικές SNMP πλατφόρμες διαχείρισης (ώστε να χρησιμοποιήσουν κάποιες set εντολές για να επιδιορθώσουν το πρόβλημα) αλλά ούτε και στην κονσόλα του γνωστού telnet. Απλά με τη χρήση ενός μεγάλου συνόλου HTML σελίδων και CGI προγραμμάτων οι διαχειριστές μπορούσαν να εισάγουν οργανωμένα μεγάλο όγκο πληροφορίας για την ανίχνευση των βλαβών και έτσι να υλοποιήσουν ορισμένες έξυπνες λειτουργίες που τους βοηθούσαν να ανακαλύψουν εύκολα το πρόβλημα (symptom-driven navigation) [2]. Στη συνέχεια για να λύσουν το πρόβλημα αυτό έπρεπε να χρησιμοποιήσουν χειροκίνητα την κονσόλα του telnet ή εναλλακτικά την SNMP πλατφόρμα (όπως είναι γνωστό το SNMP δεν είχε σημαντικές ρυθμιστικές δυνατότητες και χρησιμοποιούνταν περισσότερο για την παρακολούθηση του δικτύου).

Βέβαια όλες οι προαναφερθείσες λειτουργίες μπορούσαν και στο παρελθόν να γίνουν χωρίς τη χρήση των τεχνολογιών ιστού αλλά με ακριβές και ιδιόκτητες λύσεις. Ωστόσο η εμφάνιση των τεχνολογιών ιστού βοήθησε του διαχειριστές να κάνουν κάτι τέτοιο μόνοι τους, απλά, με πολύ χαμηλό κόστος και σε σύντομο χρονικό διάστημα. Τέλος η διαχείριση μπορούσε πλέον να γίνει από οποιοδήποτε μηχάνημα (Windows, Linux, Mac) είχε πρόσβαση στον εξυπηρετητή ιστού του διαχειριστή.

2.4.6.2. Διαχείριση βασισμένη στο φυλλομετρητή ιστοσελίδων

Το επόμενο βήμα για τους χειριστές του δικτύου ήταν να χρησιμοποιήσουν τις τεχνολογίες ιστού όχι μόνο για επικουρικές διεργασίες αλλά και για την διαδραστική διαχείριση των πρακτόρων μέσω ενός φυλλομετρητή ιστοσελίδων. Για το λόγο αυτό η διαχείριση ονομάζεται «διαχείριση με φυλλομετρητή ιστοσελίδων» (browser-based management). Τη στιγμή εκείνη στον κόσμο των IP δικτύων υπήρχαν δύο είδη ενσωματωμένων πρακτόρων διαχείρισης : οι SNMP και οι telnet πράκτορες. Γύρω από αυτούς τους αναπτύχθηκαν οι παρακάτω βασικότερες προσεγγίσεις:

- *HTML φόρμες και CGI προγράμματα με SNMP στοίβα και telnet κείμενα.* Μετά την αυτοματοποιημένη βοήθεια που απολάμβαναν οι χειριστές του δικτύου κατά την ανίχνευση και εύρεση λύσης του προβλήματος (online problem reporting and troubleshooting assistance), που περιγράφηκε παραπάνω ήθελαν να κάνουν ένα βήμα μπροστά και να αυτοματοποιήσουν και την ίδια την επιδιόρθωση των δυσλειτουργιών. Για το λόγο αυτό στα CGI προγράμματά τους προσέθεσαν και λειτουργίες SNMP διαχειριστή. Ωστόσο το SNMP δεν τους παρείχε σημαντικές δυνατότητες ρύθμισης διότι στις περισσότερες συσκευές οι MIBs δεν υποστηρίζουν όλες τις αντίστοιχες εντολές της διεπαφής γραμμής εντολών (Command Line Interface - CLI). Έτσι ήταν αναγκασμένοι να χρησιμοποιούν την κονσόλα του telnet προκειμένου να έχουν πλήρη πρόσβαση στη συσκευή. Ωστόσο το telnet λειτουργεί με ένα διαδραστικό τρόπο και αυτό τους εμπόδιζε να φτιάξουν κείμενα ή προγράμματα αυτόματης επίλυσης των προβλημάτων του δικτύου. Η κατάσταση αυτή διατηρήθηκε μέχρι το 1994, όταν υλοποιήθηκε το εργαλείο *expect*, που παρείχε τη δυνατότητα αυτοματοποίησης των διαδραστικών προγραμμάτων [52]. Με το *expect* οι διαχειριστές μπορούσαν πλέον να κωδικοποιούν τις telnet εντολές σαν κείμενα, τα οποία στη συνέχεια συνέδεαν με τις CGI εφαρμογές ανίχνευσης και εύρεσης λύσης προβλημάτων. Μόλις ο διαχειριστής με τη βοήθεια των εφαρμογών μεταδιαχείρισης αναγνώριζε κάποιο γνωστό πρόβλημα στο δίκτυο ενεργοποιούσε την αυτόματη εκτέλεση των telnet κείμενων (scripts) και το διόρθωνε.

- *Java εφαρμογές με SNMP στοίβα.* Μετά το 1997 εμφανίστηκαν νέες εταιρείες (AdventNet, Metrix, κ.α.) που με στόχο ένα μερίδιο στην αγορά της διαχείρισης (η οποία μέχρι τότε μονοπωλούνταν από τους τέσσερις μεγάλους κατασκευαστές HP, Cabletron, IBM και Sun Microsystems) προσέφεραν Java εφαρμογές με ενσωματωμένους SNMP και telnet διαχειριστές. Οι εφαρμογές αυτές ήταν είτε αυτόνομες Java πλατφόρμες διαχείρισης, είτε ενσωματωμένες στο φυλλομετρητή (Java applets). Η βασική διαφορά τους από τις HTML εφαρμογές ήταν η φιλικότητα προς το χρήστη εξαιτίας των ανώτερων γραφικών που προσέφερε η Java έναντι της HTML. Μάλιστα μετά από δύο χρόνια άρχισαν να εμφανίζονται και δωρεάν τέτοια πακέτα λογισμικού όπως το MRTG (Multi-Router Traffic Grapher) [53] και το WebtraMon [54].

- *Διαχείριση ρυθμίσεων.* Μια στοιχειώδης μορφή διαχείρισης των ρυθμίσεων ήταν εκείνη κατά την οποία ο διαχειριστής μπορούσε εύκολα να κατασκευάσει ένα αρχείο ρύθμισης (configuration file) μέσω ενός συνδυασμού HTML σελίδων και ενός CGI προγράμματος. Συγκεκριμένα, μέσω των HTML σελίδων ο διαχειριστής είχε τη δυνατότητα να επιλέξει τις κατάλληλες ρυθμίσεις για τις συσκευές του και στη συνέχεια το CGI πρόγραμμα αναλάμβανε να κατασκευάσει το αρχείο ρύθμισης. Κατόπιν ο διαχειριστής το τοποθετούσε σε ένα συγκεκριμένο εξυπηρετητή (αρχείων) από τον οποίο η συσκευή μπορούσε να το ανακτήσει μέσω ενός πρωτοκόλλου μεταφοράς αρχείων (για παράδειγμα οι Cisco IP δρομολογητές χρησιμοποιούσαν το TFTP). Η διαδικασία αυτή γίνονταν κατά τη εκκίνηση των συσκευών, η οποία

μπορούσε να σκανδαλιστεί μέσω του telnet (και επομένως μέσω της HTML εφαρμογής).

2.4.6.3. Διαχείριση βασισμένη στο HTTP

Η ολοκλήρωση των HTTP εξυπηρετητών στις δικτυακές συσκευές και σε πολλά άλλα ενσωματωμένα συστήματα αποτέλεσε κομβικό σημείο για την απομακρυσμένη διαχείριση. Πλέον οι διαχειριστές μπορούσαν να επικοινωνήσουν με τους πράκτορες μέσω του ευρέως διαδεδομένου πρωτοκόλλου HTTP. Μάλιστα οι διαχειριζόμενες συσκευές διατήρησαν τους SNMP και Telnet πράκτορες υποστηρίζοντας παράλληλα λειτουργίες αντιστοίχισης μεταξύ των HTTP και SNMP μηνυμάτων (HTTP/SNMP πύλες) αλλά και μεταξύ των HTTP μηνυμάτων και των CLI εντολών.

Ο Bruins [55] αναφέρει ένα επιτυχημένο πείραμα που έγινε από τη Cisco το 1995, κατά το οποίο εφαρμόστηκε η προσέγγιση των CLI – URL αντιστοιχίσεων. Σύμφωνα με αυτή την προσέγγιση ο διαχειριστής μπορούσε απομακρυσμένα να εκτελέσει οποιαδήποτε CLI εντολή χωρίς τη χρήση του Telnet αλλά μέσω του HTTP πρωτοκόλλου. Μάλιστα αυτό γίνονταν με ένα πολύ απλό τρόπο κατά τον οποίον οι CLI εντολές δίνονταν μέσω του URL που αιτούνταν ο διαχειριστής (π.χ. για την εντολή "show interface Ethernet 0"σε ένα μηχάνημα με IP διεύθυνση ή domain name το *router_name*, ο HTTP πελάτης του διαχειριστή ζητούσε από τον HTTP εξυπηρετητή του πράκτορα το ακόλουθο URL: *http://router_name/exec/show/interface/ethernet0//)*. Στη συνέχεια ο HTTP εξυπηρετητής του πράκτορα μετέφραζε το URL (η CLI/URL πύλη) αυτό στην αντίστοιχη CLI εντολή την οποία και έστελνε στο CLI μεταφραστή για εκτέλεση. Το αποτέλεσμα της εντολής στέλνονταν πίσω στη CLI/URL πύλη που με τη σειρά της το μετέτρεπε σε μια HTTP σελίδα την οποία έστελνε σαν απάντηση στον HTTP πελάτη του διαχειριστή. Με αυτό τον τρόπο ο χρήστης του διαχειριστικού συστήματος μπορούσε να χρησιμοποιήσει τον φυλλομετρητή ιστοσελίδων του (ή μια Java εφαρμογή με ένα HTTP πελάτη) για την πλήρη προσπέλαση της CLI γραμμής εντολών. Το σχήμα αυτό παρουσίαζε δύο πλεονεκτήματα. Καταρχάς ήταν εξαιρετικά απλό στην υλοποίησή του εξαιτίας της άμεσης αντιστοιχίας μεταξύ του URL και των CLI εντολών. Επίσης ήταν πολύ εύκολο στη χρήση του από τους χειριστές του δικτύου, οι οποίοι ήταν αρκετά εξοικειωμένοι με το CLI και έτσι δεν χρειάζονταν να μάθουν μια ακόμα γλώσσα. Οι CLI-URL αντιστοιχίσεις ήταν το πρώτο βήμα για τη διαχείριση των ρυθμίσεων, καθώς πλέον δεν υπήρχε η ανάγκη χρήσης του (διαδραστικού) Telnet. Οι χειριστές του δικτύου μπορούσαν πλέον στις διαχειριστικές εφαρμογές τους να ορίσουν σύνολα CLI εντολών που θα εκτελούνταν αυτόματα ανάλογα (symptom-driven) με τις απαντήσεις των διαχειριζόμενων συσκευών. Βέβαια η παραπάνω προσέγγιση κληρονομεί το μεγάλο πρόβλημα της διαχείρισης με τη χρήση του CLI. Το CLI δεν έχει κοινό μοντέλο δεδομένων (διαφορετικές CLI εντολές ανάλογα με το λειτουργικό σύστημα της συσκευής αλλά και την ίδια τη συσκευή) και αυτό του αφαιρεί τη δυνατότητα να ανταποκριθεί στη μαζική ρύθμιση μέσα σε ετερογενή (multi-vendor environments) περιβάλλοντα σαν αυτό της διαχείρισης των συνδρομητικών συσκευών.

Το επόμενο σημαντικό βήμα ήταν η χρήση των ενσωματωμένων HTML σελίδων και των CGI προγραμμάτων για την επιτέλεση των διαχειριστικών πράξεων. Ορισμένες εκ των HTTP

σελίδων είναι μόνο αναγνώσιμες και περιγράφουν στατική πληροφορία, όπως για παράδειγμα τα χαρακτηριστικά της διαχειριζόμενης συσκευής. Ωστόσο οι μεγάλες δυνατότητες των HTML σελίδων φαίνονται όταν αυτές αναπτυχθούν σαν HTML φόρμες και συνδεθούν με ενσωματωμένα CGI προγράμματα. Στην περίπτωση αυτή γίνονται πραγματικά χρήσιμες, καθώς επιτρέπουν στους διαχειριστές να ρυθμίζουν τις συσκευές τους με ένα πολύ φιλικότερο τρόπο από αυτόν της διαχείρισης με το φυλλομετρητή ιστοσελίδων που περιγράφηκε παραπάνω. Η άμεση διάδραση με τον πράκτορα μέσω του HTTP εξαλείφει την ανάγκη του διαχειριστή να κατασκευάσει ένα εξωτερικό αρχείο ρύθμισης και στη συνέχεια να το φορτώσει στον πράκτορα. Τώρα ο διαχειριστής μπορεί να ρυθμίσει άμεσα τον πράκτορα. Τα ενσωματωμένα CGI προγράμματα δεν συνεργάζονται μόνο με στατικές HTML σελίδες αλλά μπορούν επίσης να παράγουν δυναμικές σελίδες κατά απαίτηση. Το γεγονός αυτό είναι ιδιαίτερα χρήσιμο για την επιτέλεση της απλής παρακολούθησης των επιδόσεων μέσω της διαρκούς ανανέωσης μιας για παράδειγμα GIF εικόνας και του φυλλομετρητή ανά τακτά χρονικά διαστήματα. Όλα τα παραπάνω τα περιέγραψε το 1996 ο Mullaney κατά την παρουσίαση ενός πρωτοτύπου της Cabletron [55]. Παρόμοια εργασία της Cisco παρουσιάστηκε το 1997 από τον Maston [56]. Σήμερα όλες σχεδόν οι δικτυακές συσκευές παρέχουν μια τέτοιου είδους διεπαφή για διαδραστική ρύθμιση από τον πελάτη. Βέβαια η διαδραστική ρύθμιση δεν είναι το ζητούμενο για τη διαχείριση δεκάδων υπηρεσιών και χιλιάδων συνδρομητικών συσκευών από τους παροχείς των υπηρεσιών και τους χειριστές του δικτύου.

2.4.6.4. Διαχείριση βασισμένη στην XML

Τα τελευταία χρόνια έχει προταθεί η χρήση της XML τεχνολογίας (XML-based network management) σαν εναλλακτική λύση στις υπάρχουσες τεχνολογίες για τη διαχείριση των δικτύων και των υπηρεσιών. Η εφαρμογή της XML και των σχετικών τεχνολογιών στη διαχείριση προσφέρει σημαντικά πλεονεκτήματα τα οποία αναλύονται λεπτομερώς στην επόμενη παράγραφο. Στην παρούσα ενότητα δίνονται οι σημαντικότερες μέχρι σήμερα προσπάθειες εφαρμογής της XML στη διαχείριση και μάλιστα τόσο σε επίπεδο ερευνητικό όσο και σε επίπεδο προτυποποίησης.

Τα τελευταία δέκα χρόνια έγιναν αρκετές εξελικτικές προσπάθειες χωρίς ωστόσο να καταλήξουν σε προτυποποίηση. Η εξελικτική προσέγγιση εστίασε κυρίως στην ολοκλήρωση της διαχειριστικής πληροφορίας, που παρέχεται από τους SNMP πράκτορες, σε XML συστήματα διαχείρισης. Η προσέγγιση αυτή προωθήθηκε από τους παρόχους, οι οποίοι επιθυμούσαν να επωφεληθούν σύντομα από τα πλεονεκτήματα που έχει η XML στην επεξεργασία της διαχειριστικής πληροφορίας. Με αυτό τον τρόπο θα έλυναν χρόνια προβλήματα όπως εκείνο της διαχείρισης των ρυθμίσεων, μέσω της XML και των XSL μετασχηματισμών [98]. Το κύριο μέλημα των προσεγγίσεων αυτών ήταν ο αποτελεσματικός μετασχηματισμός των SMI ορισμών των διαχειριστικών βάσεων (MIB) σε XML σχήματα. Αυτό συνεπάγεται την αντιστοίχιση και απεικόνιση της SNMP διαχειριστικής πληροφορίας σε XML κείμενα.

Οι βασικότερες εξελικτικές προσπάθειες ήταν οι ακόλουθες:

- Ο J.P. Martin-Flatin στη διατριβή του [2] προτείνει δύο τρόπους αντιστοίχισης: την αντιστοίχιση σε επίπεδο μοντέλου (model-level mapping) και την αντιστοίχιση σε επίπεδο μεταμοντέλου (metamodel level mapping). Σύμφωνα με τον πρώτο τρόπο το κείμενο ορισμού τύπου (DTD – Document Type Definition) είναι συγκεκριμένο για κάθε SNMP MIB, ενώ τα XML στοιχεία και ιδιότητες στο κείμενο αυτό έχουν τα ίδια ονόματα με τις SNMP μεταβλητές. Αντίθετα ο δεύτερος τρόπος αντιστοίχισης ορίζει ένα γενικό κείμενο ορισμού τύπου το οποίο είναι πανομοιότυπο για όλες τις SNMP MIBs. Το κείμενο ορισμού τύπου είναι ένας τρόπος να προδιαγράψει κανείς τα στοιχεία (elements) που επιτρέπεται να περιέχονται σε ένα συγκεκριμένο XML κείμενο.

- Ο F. Strauss [68] παρουσίασε τη βιβλιοθήκη "libsmi" η οποία προσπελαύνει την SMI (Structured of Management Information) MIB πληροφορία και μετατρέπει ολόκληρη την SMI MIB σε διάφορες γλώσσες όπως Java, Corba. C, και XML. Η βιβλιοθήκη αυτή παρέχει εργαλεία για τον έλεγχο, την ανάλυση, την εκφόρτωση, τη μετατροπή και τη σύγκριση των MIB ορισμών. Μάλιστα το εργαλείο μετατροπής "smidump" μετατρέπει τις MIB μονάδες σε XML σχήματα.

- Ο J.H.Yoon αρχικά [69] πρότεινε ένα αλγόριθμο SNMP MIB - XML μετατροπής και ανέπτυξε τον αντίστοιχο μεταφραστή. Μάλιστα για την επικύρωση του αλγορίθμου υλοποιήθηκε επιπλέον ένας SNMP MIB φυλλομετρητής βασισμένος στην XML χρησιμοποιώντας τον παραπάνω μεταφραστή. Στη συνέχεια ο J.H.Yoon [70-71] βασισμένος στον παραπάνω αλγόριθμο πρότεινε διάφορες μεθόδους μετάφρασης των διαδράσεων μεταξύ μιας SNMP/XML πύλης και ενός XML διαχειριστή. Αρχικά πρότεινε μια μεταφραστική μέθοδο βασισμένη στο DOM (Document Object Model) [89]. Σύμφωνα με αυτή τη μέθοδο ο διαχειριστής προκειμένου να ανταλλάξει πληροφορία με τον SNMP πράκτορα μπορεί να προσπελάσει άμεσα το DOM δέντρο της πύλης χρησιμοποιώντας τις DOM διεπαφές. Η τελευταία και πιο σημαντική μέθοδος που προτάθηκε ήταν βασισμένη στο SOAP (Simple Object Access Protocol) πρωτόκολλο. Χρησιμοποιώντας το SOAP η πύλη παρέχει μια ευέλικτη και προτυποποιημένη μέθοδο για διάδραση με τον XML διαχειριστή σε ένα κατανεμημένο περιβάλλον.

- Το 2003 το εργαστήριο Avaya Labs [90] ξεκίνησε την ανάπτυξη μιας διαχειριστικής διεπαφής που βασίζονταν στην XML και είχε ως στόχο τις SNMP συσκευές. Το πρωτότυπο αποτελείται από 3 μέρη:

    - Ένα εργαλείο για την αυτόματη παραγωγή ενός XML σχήματος (XML schema) βασισμένου στην SNMP SMI πληροφορία.

    - Ένα πρωτόκολλο ανταλλαγής μηνυμάτων που βασίζεται σε XML κλήσεις απομακρυσμένων διεργασιών (RPC – Remote Procedure Calls) και έχει σαν στόχο την ανάκτηση και την τροποποίηση της MIB πληροφορίας στις SNMP συσκευές. Το πρωτόκολλο αυτό καθορίζει το XML σχήμα για ένα σύνολο εντολών (GET, SET, LIST, CREATE, DELETE) και ταυτοποιεί τις MIB

μεταβλητές χρησιμοποιώντας αναγνωριστικά βασισμένα στην τεχνολογία XPath [91].

- Ένα προσαρμογέα για την ανάκτηση και τροποποίηση της διαχειριστικής πληροφορίας με τη μορφή XML δεδομένων που πηγάζουν από την SNMP MIB της συσκευής.

Διάφορες ομάδες και φόρουμ εντός των οργανισμών προτυποποίησης αλλά και πολλές πρωτοβουλίες της βιομηχανίας ασχολούνται τα τελευταία χρόνια επισταμένα με την προτυποποίηση αρχιτεκτονικών και πρωτοκόλλων διαχείρισης βασισμένα στην XML τεχνολογία. Μάλιστα εξαιτίας των σημαντικών μειονεκτημάτων του SNMP που αναλύθηκαν στην αρχή αυτού του κεφαλαίου οι προσπάθειες προτυποποίησης ήταν ριζοσπαστικές εστιάζοντας έτσι στη συνολική εφαρμογή της XML και στη ολοκληρωτική κατάργηση του SNMP ακόμη και από τους πράκτορες διαχείρισης των συσκευών.

Η NMRG (Network Management Research Group) ομάδα [73] του IRTF (Internet Research Task Force) είναι ένα φόρουμ για τους ερευνητές οι οποίοι ενδιαφέρονται να συζητήσουν και να αναπτύξουν νέες τεχνολογίες με στόχο τη βελτίωση της διαχείρισης του διαδικτύου. Στο 11^{ο} συνέδριο του NMRG συζητήθηκαν τα πλεονεκτήματα και τα μειονεκτήματα της χρήσης τεχνολογιών ιστού (που χρησιμοποιούν την XML) όπως είναι το SOAP [74], το WSDL (Web-Services Definition Language) [75] και το UDDI (Universal Discovery Description and Integration) [76]. Επίσης συζητήθηκε το θέμα της ασφάλειας στις τεχνολογίες ιστού.

Το OASIS (Organization for the Advancement of Structured Information Standards) [77] είναι μια κοινοπραξία που παράγει παγκόσμια πρότυπα για την ασφάλεια, τις τεχνολογίες ιστού, τη συμμόρφωση με την XML, τις επιχειρησιακές συναλλαγές, την ηλεκτρονική δημοσίευση κ.α. Στόχος του OASIS είναι να αναπτύξει ανοιχτά προτυποποιημένα πρωτοκόλλα προκειμένου να παράσχει ένα μηχανισμό που θα βασίζεται στις τεχνολογίες ιστού και θα επιτρέπει την παρακολούθηση και τον έλεγχο των διαχειριζόμενων στοιχείων σε ένα κατανεμημένο περιβάλλον. Μάλιστα ο μηχανισμός αυτός θα πρέπει να βασίζεται σε αποδεκτά από τη βιομηχανία μοντέλα, μεθόδους και λειτουργίες συμπεριλαμβανομένου των OMI (Open Model Interface) [78], XML, SOAP και DMTF CIM λειτουργιών. Το Μάρτιο του 2005 το OASIS δημοσίευσε το WSDM (Web-Services Distributed Management) [92] πρότυπο για διαχείριση με χρήση των υπηρεσιών ιστού (Management Using Web-Services – MUWS 1.0) . Καθώς το πρότυπο αυτό είναι αρκετά πρόσφατο μέχρι τώρα ούτε η παρούσα διατριβή αλλά ούτε και άλλες εργασίες έχουν μελετήσει την καταλληλότητά του για τη μαζική διαχείριση των συνδρομητικών συσκευών. Στο σημείο αυτό αξίζει να τονιστεί το γεγονός ότι ακόμη και αυτή η προσπάθεια προτυποποίησης στηρίζεται στη συνδυασμένη χρήση του SOAP/XML και του HTTP για την κωδικοποίηση και τη μεταφορά της διαχειριστικής πληροφορίας αντίστοιχα.

Ωστόσο οι σημαντικότερες προσπάθειες μέχρι σήμερα καταγράφονται από τις ομάδες της DMTF (Distributed Management Task Force) , της IETF (Internet Engineering Task Force), του DSL φόρουμ (Digital Subscriber Line) και του UPnP φόρουμ (Universal Plug and Play). Οι προσπάθειες αυτές μπορούν να συνοψιστούν στα ακόλουθα:

- Το WBEM (Web-based Enterprise Management) [79] είναι μια πρωτοβουλία της DMTF και περιλαμβάνει ένα σύνολο τεχνολογιών που επιτρέπουν τη διαλειτουργική διαχείριση μιας εταιρείας. Το WBEM αποτελείται από ένα κοινό μοντέλο πληροφορίας (Common Information Model - CIM) [80], ένα κείμενο ορισμού τύπου (Document Type Definition - DTD) για την αναπαράσταση του CIM μοντέλου σε XML καθώς και από τις προδιαγραφές των CIM πράξεων πάνω από το HTTP πρωτόκολλο. Το CIM παρέχει ένα εύληπτο αντικειμενοστραφές πληροφοριακό μοντέλο και τα CIM σχήματα (συλλογές από ορισμούς κλάσεων που περιγράφουν τα διαχειριζόμενα αντικείμενα) υλοποιούνται τόσο για τους διαχειριστές όσο και για τις δικτυακές συσκευές. Μάλιστα το WBEM ανανεώθηκε και συμπεριέλαβε και το πρότυπο SOAP. Όλα τα διαχειριστικά δεδομένα του WBEM κωδικοποιούνται σε XML και μεταφέρονται μέσω του HTTP πρωτοκόλλου.
- Το NetConf (Network Configuration Protocol) αποτελεί τους καρπούς της προσπάθειας της IETF για διαχείριση βασισμένη στην XML. Συγκεκριμένα το πρωτόκολλο αυτό προδιαγράφεται από τη NetConf ομάδα εργασίας [81] η οποία συστάθηκε το Μάιο του 2003. Το NetConf χρησιμοποιεί κλήσεις απομακρυσμένων διεργασιών (RPCs) για να ορίσει μια τυπική προγραμματιστική διεπαφή (API). Επίσης ορίζει διαχειριστικές πράξεις οι οποίες καλούνται σαν RPC μέθοδοι με XML κωδικοποιημένες παραμέτρους. Μάλιστα και σε αυτό το πρότυπο ο προτιμώμενος μηχανισμός για τις απομακρυσμένες κλήσεις είναι το SOAP πρωτόκολλο. Παρόλο που μέχρι σήμερα το NetConf πρωτόκολλο έχει κάνει σημαντική πρόοδο, ακόμα δεν είναι σε θέση να εφαρμοστεί σε εμπορικό επίπεδο. Θέματα όπως η κλιμακωσιμότητα εξαιτίας των μόνιμων TCP συνδέσεων μεταξύ του διαχειριστή και των πρακτόρων αλλά και η έλλειψη μοντέλων διαχείρισης έχουν καθυστερήσει σημαντικά την προώθηση και αποδοχή του εν λόγω προτύπου.
- Με κύριο στόχο τη διανομή και ενεργοποίηση των απομακρυσμένων υπηρεσιών, το DSL φόρουμ προσπαθεί να καλύψει τα παραπάνω κενά προδιαγράφοντας το πρωτόκολλο απομακρυσμένης διαχείρισης συνδρομητικών (CPE WAN Management Protocol - CWMP) [93]. Το πρωτόκολλο αυτό φαίνεται να είναι μια πολλά υποσχόμενη λύση για τη ρύθμιση και διαχείριση των συνδρομητικών συσκευών (CPEs – Customer Premises Equipment) όχι μόνο σε DSL περιβάλλοντα αλλά και σε οποιοδήποτε άλλο IP περιβάλλον, το οποίο μπορεί να υποστηρίξει την παροχή πολλαπλών υπηρεσιών στο συνδρομητή. Τυπικά ο όρος CPE περιλαμβάνει δικτυακές πύλες, προσωπικούς υπολογιστές, κάμερες, ηχοσυστήματα, Video/VoIP τηλέφωνα, συνδρομητικούς αποκωδικοποιητές εικόνας και ήχου καθώς και οποιασδήποτε άλλης δικτυωμένης συσκευής. Επιπλέον μια οικογένεια [94, 95] από συσχετιζόμενα πρότυπα για την τοπική (βασισμένο στο UPnP) και απομακρυσμένη ρύθμιση των CPEs επεκτείνει το εύρος δράσης και τη σημασία των εργασιών προτυποποίησης του DSL φόρουμ. Ομοίως με το NetConf και το WBEM, το CWMP πρωτόκολλο χρησιμοποιεί το SOAP για την κωδικοποίηση των απομακρυσμένων διεργασιών (SOAP RPCs) και το HTTP για τη μεταφορά της διαχειριστικής πληροφορίας.

Ενδεικτικό της ωριμότητας του CWMP πρωτοκόλλου είναι ότι παρόλο που αυτό ανακοινώθηκε το Μάιο του 2004 υπάρχουν δικτυακές πύλες που ήδη το υποστηρίζουν (Thomson, Alcatel, D-Link, Linksys), ενώ παράλληλα γίνεται σημαντική δουλειά για τη CWMP μοντελοποίηση και άλλων συσκευών όπως είναι τα VoIP τηλέφωνα και οι συνδρομητικοί αποκωδικοποιητές. Περισσότερες πληροφορίες για τη CWMP αρχιτεκτονική αλλά και το ίδιο το πρωτόκολλο δίνονται παρακάτω στα κεφάλαια 4 και 5.

- Βέβαια η πιο διαδομένη εμπορικά τεχνολογία αυτόματης εγκατάστασης και ρύθμισης συσκευών και υπηρεσιών είναι το πρωτόκολλο UPnP (Universal Plug and Play) [96]. Το UPnP αρχικά προωθήθηκε από τη Microsoft η οποία το 1999 οργάνωσε το UPnP φόρουμ. Σήμερα στο φόρουμ αυτό συμμετέχουν ενεργά εκατοντάδες μεγάλοι κατασκευαστές καταναλωτικών ηλεκτρονικών και συνδρομητικών συσκευών. Οι περισσότεροι εξ αυτών έχουν ενσωματώσει τη UPnP τεχνολογία στις συσκευές τους. Ομοίως με τα παραπάνω πρότυπα διαχείρισης, το UPnP χρησιμοποιεί υπάρχοντα διαδικτυακά πρότυπα (TCP, UDP, HTTP, SOAP και XML) και ορίζει μηχανισμούς για την ανίχνευση, τον έλεγχο, την περιγραφή, την ενημέρωση συμβάντων και την παρουσίαση των συσκευών. Ωστόσο το UPnP πρωτόκολλο σχεδιάστηκε με στόχο να λύσει προβλήματα διαχείρισης των τοπικών δικτύων. Τα μηνύματα εκπομπής για λόγους ανίχνευσης (SSDP) και ενημέρωσης (GENA), η χρήση του αναξιόπιστου UDP πρωτοκόλλου αλλά και η έλλειψη μηχανισμών ασφαλείας είναι ορισμένοι από τους βασικότερους λόγους που καθιστούν το UPnP ακατάλληλο να υποστηρίξει τη διαχείριση συσκευών και υπηρεσιών από το δημόσιο δίκτυο. Το κεφάλαιο 5 πέρα των άλλων κάνει και μια εκτενή ανάλυση της UPnP τεχνολογίας.

- Τέλος δύο σημαντικές πρωτοβουλίες της βιομηχανίας που χρησιμοποιούν και αυτές την XML για την κωδικοποίηση της διαχειριστικής πληροφορίας είναι το JUNOScript [67] της Juniper Networks και το Configuration Registrar (αρχειοφύλακας ρυθμίσεων) της Cisco. Το JUNOScript επιτρέπει σε εφαρμογές διαχείρισης την πρόσβαση σε δεδομένα ρύθμισης και λειτουργίας χρησιμοποιώντας ένα μηχανισμό απομακρυσμένων κλήσεων βασισμένο στην XML (XML-RPC). Το JUNOScript προδιαγράφει τα RPC μηνύματα ανάμεσα στις εφαρμογές πελάτη (διαχειριστές) και στους JUNOScript εξυπηρετητές εντός των συσκευών (πράκτορες). Οι εφαρμογές πελάτη μπορούν να ζητήσουν πληροφορία κωδικοποιώντας την αίτηση με τις JUNOScript ετικέτες τις οποίες και στέλνουν στον εξυπηρετητή μέσω του telnet ή του ssh πρωτοκόλλου. Με τη σειρά του ο JUNOScript εξυπηρετητής παραδίδει την αίτηση στις αντίστοιχες μονάδες λογισμικού εντός της συσκευής, κωδικοποιεί την απάντηση πάλι με τις JUNOScript ετικέτες και επιστρέφει το αποτέλεσμα πίσω στην εφαρμογή πελάτη. Όσον αφορά στον Configuration Registrar αυτό πρόκειται για ένα σύστημα βασισμένο στις τεχνολογίες ιστού. Στόχος είναι η αυτόματη διανομή των αρχείων ρύθμισης στις Cisco IOS δικτυακές συσκευές. Το σύστημα αυτό δουλεύει σε συνεργασία με τους CNS (Cisco Networking Services) πράκτορες ρύθμισης που βρίσκονται σε κάθε συσκευή. Ο Configuration Registrar

παραδίδει τις αρχικές ρυθμίσεις στις Cisco συσκευές μόλις αυτές συνδέονται για πρώτη φορά στο δίκτυο. Σαν πρωτόκολλο επικοινωνίας μεταξύ των πρακτόρων και του Configuration Registrar χρησιμοποιείται το HTTP, ενώ τα δεδομένα ρύθμισης κωδικοποιούνται με τη χρήση της XML. Ο πράκτορας ρύθμισης της συσκευής χρησιμοποιεί το δικό του XML γραμματικό αναλυτή (xml parser) για να μεταφράσει τα δεδομένα ρύθμισης από τα XML αρχεία που λαμβάνει.

2.5. Η XML και τα πλεονεκτήματά της στη διαχείριση

Από την προηγούμενη ενότητα γίνεται φανερή η στροφή της ακαδημαϊκής κοινότητας αλλά και των εταιρειών προς τις XML διαχειριστικές λύσεις. Τι είναι όμως πραγματικά αυτό που κάνει την XML να κυριαρχεί στο χώρο της διαχείρισης τα τελευταία πέντε χρόνια; Μήπως η XML είναι απλά μόδα που θα περάσει; Η παρούσα ενότητα επιχειρεί να απαντήσει σε τέτοιου είδους ερωτήματα.

2.5.1. Μια σύντομη ανασκόπηση της XML

Η XML (eXtensible Markup Language) εμφανίστηκε στα τέλη του 1998 και αναπτύχθηκε κάτω από την εποπτεία του W3C. Τα κύρια δομικά της στοιχεία είναι η XML 1.0 προδιαγραφή [57] που δημοσιεύτηκε το Φεβρουάριο του 1998 και η προδιαγραφή των ονοματοχώρων (Namespaces) [97] η οποία δημοσιεύτηκε τον Ιανουάριο του 1999. Αρχικά η XML επινοήθηκε σαν μια φόρμα ανταλλαγής κειμένων για τη δημοσίευση ιστού (Web publishing). Αρχικά διόρθωσε πολλές ατέλειες της HTML και προβλέπονταν να την αντικαταστήσει σταδιακά. Ωστόσο οι προγραμματιστές των φυλλομετρητών ιστού αποδείχθηκαν πολύ βραδείς στην ενσωμάτωση της XML με αποτέλεσμα η συνολική αντικατάσταση να μην διαφαίνεται στο μέλλον. Στην πραγματικότητα σήμερα η XML περισσότερο συμπληρώνει την HTML στη δημοσίευση ιστού παρά την αντικαθιστά.

Ωστόσο η πραγματική επιτυχία της XML αποδείχθηκε ότι ήταν σε άλλους τομείς και όχι στη δημοσίευση ιστού. Σύντομα φάνηκε ότι η XML ήταν πολύ χρήσιμη κατά τη δόμηση και την αναγνώριση πολύπλοκων δομών δεδομένων, οι οποίες μπορεί ποτέ να μη χρειάζονταν να εμφανιστούν ή να τυπωθούν αλλά απλά ανταλλάσσονταν ανάμεσα σε εφαρμογές που εκτελούνται σε απομακρυσμένα μηχανήματα [58]. Αυτό εξηγεί το λόγο για τον οποίο η XML έχει πετύχει στη μηχανική λογισμικού γενικότερα και όχι μόνο στη μικρή αγορά της δημοσίευσης ιστού. Η XML γίνεται γρήγορα η γλώσσα επικοινωνίας μεταξύ απομακρυσμένων εφαρμογών, που δεν μιλούν την ίδια γλώσσα, δεν είναι ανεπτυγμένες από τον ίδιο κατασκευαστή και δεν εκτελούνται απαραίτητα πάνω από το ίδιο μεσισμικό ή την ίδια εικονική μηχανή ή το ίδιο λειτουργικό σύστημα. Ο συνδυασμός της HTTP/XML γίνεται ολοένα και περισσότερο ένας πρότυπος τρόπος υλοποίησης της αρχιτεκτονικής τριών βαθμίδων (three-tier architecture) στη βιομηχανία λογισμικού.

Στην πραγματικότητα η XML είναι μια μεταγλώσσα, δηλαδή μια γλώσσα που περιγράφει άλλες γλώσσες. Τα XML κείμενα έχουν μια λογική (στοιχεία - elements) και μια φυσική (οντότητες - entities) δομή [58]. Οι οντότητες μπορούν να ονομάζονται και να αποθηκεύονται ξεχωριστά. Ένα κείμενο ορισμού τύπου (Document Type Definition) είναι στην ουσία ένας

τρόπος να προδιαγράφει κανείς τα επιτρεπτά στοιχεία σε ένα συγκεκριμένο τύπο XML κειμένου. Τα XML κείμενα συνήθως μεταφέρονται μέσω του πρωτοκόλλου HTTP. Η πληροφορία που βρίσκεται μέσα σε ένα XML κείμενο είναι αυτό-περιγραφόμενη και επιτρέπει σε διαφορετικούς αναγνώστες να την αποδώσουν διαφορετικά (π.χ. σαν ήχο, γραφικά, κείμενο) ή σε διαφορετικούς χρήστες να ορίσουν τον τρόπο απόδοσής του από τον αναγνώστη. Για περισσότερες πληροφορίες σχετικά με την XML μπορεί κανείς να ανατρέξει στον Goldfarb και Prescod [59] (εισαγωγικό υλικό) αλλά και στον Megginson [60] (για πιο προχωρημένο επίπεδο).

Παρόλο που η XML είναι μια σχετικά πρόσφατη τεχνολογία, ήδη χρησιμοποιείται σε πολλές περιοχές. Για παράδειγμα όλες οι σχεσιακές βάσεις δεδομένων ήδη υποστηρίζουν XML προγραμματιστικές διεπαφές (XML APIs). Επίσης πολλές γλώσσες μαρκαρίσματος βασισμένες στην XML έχουν ήδη προταθεί σε πολλούς τομείς της βιομηχανίας [62]. Ορισμένες εξ αυτών είναι: οι MathML (Mathematical Markup Language) για τα λογισμικά μαθητικών, η CML (Chemical Markup Language) και η BIOML (Biopolymer Markup Language) για τα λογισμικά της χημείας, η MusicML (Music Markup Language) για τα μουσικά λογισμικά και η cXML (commerce XML) για το ηλεκτρονικό εμπόριο μεταξύ επιχειρήσεων.

2.5.2. Τα πλεονεκτήματα της XML

Πώς όμως η XML κατάφερε να είναι σήμερα τόσο διαδεδομένη και μάλιστα σε τόσο μικρό χρονικό διάστημα; Τα κύρια πλεονεκτήματά της στη βιομηχανία λογισμικού μπορούν να συνοψιστούν στα ακόλουθα. Καταρχάς η XML είναι εύκολη στη μάθηση. Η εξοικείωση με την CORBA, τη J2EE ή ακόμα και με την DCOM πλατφόρμα απαιτεί συνήθως μισό χρόνο εκπαίδευσης και πρακτικής, ενώ η XML απαιτεί μόλις λίγες εβδομάδες. Ο λόγος που συμβαίνει αυτό είναι ότι η XML είναι σημαντικά λιγότερο αναλυτική και σύνθετη με αποτέλεσμα να είναι πολύ ευκολότερη στην κατανόηση της. Επίσης η XML παρουσιάζει πολύ χαμηλό κόστος ανάπτυξης, αφού πολλοί XML γραμματικοί αναλυτές είναι διαθέσιμοι δωρεάν στο διαδίκτυο [63]. Αντίθετα οι πλατφόρμες μεσισμικού είναι γενικά ακριβές και ιδιαίτερα εκείνες της CORBA και της J2EE. Οι λίγες υπάρχουσες CORBA πλατφόρμες που διατίθενται δωρεάν και παράλληλα είναι πλούσιες σε υποστηριζόμενα χαρακτηριστικά, σπάνια χρησιμοποιούνται από την αγορά για λόγους συντήρησης. Επιπλέον η XML επιτρέπει στους προγραμματιστές να χρησιμοποιήσουν την ίδια τεχνολογία παντού για οποιαδήποτε εφαρμογή. Αυτό έχει ως αποτέλεσμα τη δραματική μείωση του κόστους για τους διαχειριστές ανάπτυξης των έργων λογισμικού. Υπό αυτή την έννοια η XML είναι τόσο επιτυχημένη όσο και το HTTP, οι φυλλομετρητές ιστού και οι εφαρμογές Java (Java applets) που προηγήθηκαν αυτής. Μάλιστα τα XML κείμενα είναι αναγνώσιμα όχι μόνο από τη μηχανή αλλά και από τον ίδιο τον άνθρωπο, αντίθετα με τα SNMP BER (Basic Encoded Rules) κείμενα. Το γεγονός αυτό απλοποιεί σημαντικά την αποσφαλμάτωση.

Όσον αφορά στα πλεονεκτήματα της XML, που αφορούν αποκλειστικά στη διαχείριση το πιο σημαντικό ίσως είναι η δυνατότητά της να περιγράφει πολύπλοκες δομές και μάλιστα με ένα δεντρικό και απόλυτα ευκρινές τρόπο (αντίθετα για παράδειγμα από την BER/ASN.1 του

SNMP που είναι πολύ περιορισμένη). Το γεγονός αυτό επιτρέπει την εύκολη δόμηση και μεταφορά της διαχειριστικής πληροφορίας ανεξάρτητα από την πολυπλοκότητα της διαχειριζόμενης συσκευής ή υπηρεσίας. Επίσης ένα σύνολο τεχνολογιών όπως η DOM (Document Object Model) [89], η XPath (XML Path language) [91] και η XSL (eXtensible Stylesheet Language) [98], που είναι ανεπτυγμένες γύρω από την XML διευκολύνουν σημαντικά την πρόσβαση και την επεξεργασία των διαχειριστικών δεδομένων. Με αυτό τον τρόπο η ολοκλήρωση της διαχειριστικής πληροφορίας σε κάθε είδους εφαρμογή δεν ήταν πότε άλλοτε τόσο εύκολη. Ένα άλλο σημαντικό χαρακτηριστικό είναι ότι η XML υποστηρίζει απλές μορφές κινητού κώδικα, όπως είναι εκείνη της απομακρυσμένης αποτίμησης (remote evaluation) και ο κώδικας κατά απαίτηση (code on demand). Για παράδειγμα ένα κείμενο (script) μπορεί να μεταφερθεί από το διαχειριστή σε ένα πράκτορα ανάμεσα στις ετικέτες <script> και </script>, ενώ οι ιδιότητες (attributes) θα μπορούσαν να μεταφέρουν μεταδεδομένα όπως είναι το όνομα και την έκδοση της κείμενο-γλώσσας. Αυτό επιτρέπει στους διαχειριστές να εφαρμόσουν σταδιακά τις πολλά υποσχόμενες τεχνολογίες κινητού κώδικα καθώς οι πράκτορες στο μέλλον θα γίνονται εξυπνότεροι, θα διαθέτουν περισσότερους πόρους (αποθηκευτικούς και υπολογιστικούς) αλλά και καθώς τα θέματα ασφάλειας θα διευθετούνται. Τέλος ένα ακόμα βασικό πλεονέκτημα της XML είναι η δυνατότητα που παρέχει για ολοκληρωμένη διαχείριση. Η διατριβή του Flatin [2] εξηγεί αναλυτικά τον τρόπο με τον οποίο η XML μπορεί να γίνει η κοινή γλώσσα επικοινωνίας (*lingua franca*) μεταξύ διαχειριστών για τη διαχείριση των συστημάτων, των δικτύων, των εφαρμογών και των υπηρεσιών.

2.6. Ανακεφαλαίωση - Συμπεράσματα

Στο παρόν κεφάλαιο παρουσιάστηκαν συνοπτικά οι βασικότερες υποψήφιες τεχνολογίες για τη μαζική διαχείριση των δικτύων και υπηρεσιών της επόμενης δεκαετίας. Παρόλο που το SNMP διαδραμάτισε κυρίαρχο ρόλο την προηγούμενη δεκαετία, ωστόσο δεν φαίνεται ότι μπορεί να κάνει το ίδιο και στην επόμενη. Η αδυναμία της SMI να μοντελοποιεί πολύπλοκες δομές, η χρήση του αναξιόπιστου πρωτοκόλλου UDP, η έλλειψη απλής και διάφανης ασφάλειας αλλά και η απουσία της ολοκληρωμένης διαχείρισης ακόμη και από τις πιο πρόσφατες εκδόσεις αποτελούν ορισμένες από τις βασικότερες εγγενείς αδυναμίες της SNMP τεχνολογίας. Τα παραπάνω προβλήματα καθιστούν το SNMP ανεπαρκές για να υποστηρίξει τη διαχείριση ενός μεγάλου πλήθους πολύπλοκων υπηρεσιών, δικτύων και συνδρομητικών συσκευών και μάλιστα με ένα ολοκληρωμένο τρόπο που θα μείωνε σημαντικά το κόστος της. Ωστόσο την τελευταία δεκαετία έχουν γίνει αρκετές προσπάθειες για την εφαρμογή στη διαχείριση τεχνολογιών προερχόμενων κυρίως από τον κόσμο της μηχανικής λογισμικού. Μια από αυτές ήταν η CORBA, η οποία τα τελευταία χρόνια έγινε ευρέως αποδεκτή από την αγορά των τηλεπικοινωνιών καθώς αποτέλεσε το βασικό δομικό στοιχείο κατανεμημένης επεξεργασίας της TMN αρχιτεκτονικής. Ωστόσο το γεγονός ότι η CORBA είναι μια τεχνολογία μεσισμικού έχει ως αποτέλεσμα το σχετικά υψηλό κόστος κατά την ολοκλήρωσή της στις ενσωματωμένες συσκευές χαμηλού κόστους (π.χ. οικιακή πύλη, VoIP τηλέφωνο) αλλά και τις υψηλές απαιτήσεις σε μνήμη και επεξεργαστική ισχύ από το διαχειριζόμενο

σύστημα. Εξαιτίας των παραπάνω δεν μπορεί κανείς να περιμένει ότι όλες οι συνδρομητικές συσκευές θα υποστηρίζουν το CORBA μεσισμικό και επομένως η CORBA δεν μπορεί να αποτελέσει μια καθολική λύση μαζικής διαχείρισης. Για αυτούς τους λόγους μέχρι σήμερα η CORBA έχει περιοριστεί στην αγορά του ακριβού τηλεπικοινωνιακού εξοπλισμού.

Από την άλλη πλευρά οι τεχνολογίες κινητού κώδικα και ιδιαίτερα τα ενεργά δίκτυα αποτελούν μια πολλά υποσχόμενη λύση για τη διαχείριση. Το γεγονός ότι η ευφυΐα απομακρύνεται από το σταθμό διαχείρισης και έρχεται μέσα στο δίκτυο έχει ως αποτέλεσμα τόσο την αποσυμφόρηση του σταθμού διαχείρισης σε επίπεδο επεξεργασίας και επικοινωνίας όσο και την έγκαιρη και αποδοτική ανίχνευση των προβλημάτων. Ωστόσο θέματα που σχετίζονται με την ασφάλεια (τρόποι προστασίας από κακόβουλο κώδικα) και με τις υψηλές απαιτήσεις σε πόρους από τα διαχειριζόμενα συστήματα καθυστερούν σημαντικά τη διάδοση και αποδοχή των σχετικών λύσεων από την αγορά της διαχείρισης.

Ωστόσο τα τελευταία χρόνια με την OSGi πρωτοβουλία η κοινωνία της Java έχει κάνει μια σημαντική προσπάθεια για την απομακρυσμένη διανομή, εγκατάσταση και ενεργοποίηση των υπηρεσιών στα συνδρομητικά δίκτυα. Η OSGi πλατφόρμα απαιτεί μια Java εικονική μηχανή για την εκτέλεσή της και εξαιτίας αυτού το OSGi κληρονομεί όλα τα προβλήματα που παρουσιάζουν οι τεχνολογίες εξειδικευμένου μεσισμικού στη διαχείριση. Έτσι οι μεγάλες απαιτήσεις σε μνήμη και επεξεργαστική ισχύ αλλά και το κόστος αδειοδότησης (licensing) για τη χρήση της Java εικονικής μηχανής κάνει διστακτικούς ακόμα και τους μεγάλους κατασκευαστές συνδρομητικών συσκευών. Βέβαια η φορητότητα του κώδικα είναι πολύ σημαντική για τη γρήγορη, μαζική και με χαμηλό κόστος ανάπτυξη και διανομή των εφαρμογών που υλοποιούν τις υπηρεσίες. Για το λόγο αυτό θεωρούμε ότι όταν οι δυνατότητες των συνδρομητικών συσκευών χαμηλού κόστους αυξηθούν σημαντικά, τότε η OSGi τεχνολογία θα μπορεί να γίνει ευρέως αποδεκτή από την αγορά. Ωστόσο από την μέχρι τώρα εμπειρία του συγγραφέα (επαφές με τα ερευνητικά τμήματα μεγάλων κατασκευαστών όπως η Thomson, η Alcatel και η ST Microelectronics) κάτι τέτοιο δεν προβλέπεται να γίνει πραγματικότητα τουλάχιστον μέσα στην επόμενη πενταετία.

Από την άλλη πλευρά διαφαίνεται ότι ο συνδυασμός του προτύπου SOAP/XML για την κωδικοποίηση της διαχειριστικής πληροφορίας και του ευρέως διαδεδομένου και υποστηριζόμενου πρωτοκόλλου HTTP ίσως αποτελεί τη λύση για την ολοκληρωμένη και μαζική διαχείριση των δικτύων και υπηρεσιών της επόμενης γενιάς. Η δυνατότητα της XML να περιγράφει πολύπλοκες δομές δεδομένων, να ολοκληρώνεται πολύ εύκολα με οποιαδήποτε εφαρμογή, να είναι αυτό-επεξηγούμενη και κατανοητή στον άνθρωπο, να έχει πολύ χαμηλό κόστος ανάπτυξης της και παράλληλα να είναι ιδιαιτέρως απλή της δίνει το προβάδισμα έναντι όλων των υπολοίπων σύνθετων τεχνολογιών. Μάλιστα το SOAP πρωτόκολλο που αποτελεί κι αυτό ένα πρότυπο της W3C και χρησιμοποιεί την XML για την κωδικοποίηση των απομακρυσμένων κλήσεων αποτελεί μια προφανή και αξιόπιστη λύση για τη διαχείριση με τεχνολογίες ιστού. Επιπλέον η χρήση του πρωτοκόλλου HTTP για τη μεταφορά της διαχειριστικής πληροφορίας παρέχει σημαντικά πλεονεκτήματα έναντι άλλων τεχνολογιών. Το HTTP σήμερα υποστηρίζεται σχεδόν από όλες τις συσκευές, διασφαλίζει την εύκολη διάβαση μέσα από τα τείχη προστασίας (firewalls) και μπορεί να παρέχει απλή

και διάφανη ασφάλεια (όπως εκείνη του SSL) που είναι πολύ βασική για τη διαχείριση. Τέλος το γεγονός ότι η XML υποστηρίζει απλές μορφές κινητού κώδικα αφήνει την πόρτα ανοιχτή στο μέλλον για μια ομαλή ολοκλήρωση των πολλά υποσχόμενων (αλλά προς το παρόν ανώριμων) τεχνολογιών κινητού κώδικα με τις XML/HTTP διαχειριστικές πλατφόρμες.

2.7. Βιβλιογραφία - Αναφορές

[1] M. Rose and K. McCloghrie "Structure and Identification of Management Information for TCP/IP-based internets", IETF RFC 1065, August 1988.

[2] JP Martin-Flatin, "Web-Based Management of IP Networks and Systems", Publisher: John Wiley and Sons, 2003.

[3] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser (Eds.), "RFC 1906. Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", IETF, January 1996.

[4] M.J.Choi, "An Architectural Framework for XML-based Network Management", Ph.D thesis, Pohang University of Science and Technology, 2003.

[5] B. Wijnen, R. Presuhn, and K. McCloghrie (Eds.), "RFC 2575. View-Based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", IETF, April 1999.

[6] ITU-T Rec. X.701, "Information Technology — Open Systems Interconnection, Systems Management Overview," 1992.

[7] G. Pavlou et al., "On Management Technologies and the Potential of Web Services," IEEE Communications Magazine, vol. 42, no. 7, July 2004.

[8] ITU-T. Recommendation M.3010, "Principles for a Telecommunications management network", ITU, Geneva, Switzerland, May 1996.

[9] J.K. Shrewsbury, "An Introduction to TMN", Journal of Network and Systems Management, 3(1):13–38, 1995.

[10] J. Siegel, "CORBA Fundamentals and Programming" Publisher: Wiley, New York, NY, USA, 1996.

[11] OMG, "The Common Object Request Broker: Architecture and Specification", Revision 2.0. July 1995.

[12] G. Pavlou, "Telecommunications Management Network: a Novel Approach Towards its Architecture and Realisation Through Object-Oriented Software Platforms", Ph.D. Thesis, University College London, March 1998.

[13] G. Pavlou, "Using Distributed Object Technologies in Telecommunications Network Management", IEEE Journal on Selected Areas in Communications, 18(5):644–653, 2000.

[14] S. Mazumdar, "Inter-Domain Management between CORBA and SNMP", In Proc. 7th IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM'96), L'Aquila, Italy, October 1996.

[15] S. Mazumdar (Ed.), "Translation of SNMPv2 Specification into CORBA-IDL" Report of the XoJIDM task force, September 1996.

[16] J. Pavón, J. Tomás, Y. Bardout, and L.H. Hauw, "CORBA for Network and Service Management in the TINA Framework", IEEE Communications Magazine, 36(3):72–79, 1998.

[17] J. Pavón, "Building Telecommunications Management Applications with CORBA", IEEE Communications Surveys, 2(2):2–16, 1999.

[18] W.J. Barr, T. Boyd, and Y. Inoue, "The TINA Initiative", IEEE Communications Magazine, 31(3):70–76, 1993.

[19] H. Berndt, T. Hamada, and P. Graubmann, "TINA: Its Achievements and Its Future Directions", IEEE Communications Surveys & Tutorials, 3(1):2–16, 2000.

[20] A. Fuggetta, G.P. Picco, and G. Vigna, "Understanding Code Mobility", IEEE Transactions on Software Engineering, 24(5):342–361, 1998.

[21] J.W. Stamos and D.K. Gifford, "Remote Evaluation". ACM Transactions on Programming Languages and Systems, 12(4):537– 565, 1990.

[22] Y. Yemini, G. Goldszmidt, and S. Yemini. "Network Management by Delegation", Proc. IFIP 2nd International Symposium on Integrated Network Management (ISINM'91), Washington, DC, USA, April 1991, pp. 95–107. North-Holland, Elsevier, Amsterdam, The Netherlands, 1991.

[23] G. Goldszmidt, "Distributed Management by Delegation" Ph.D. thesis, Columbia University, New York, NY, USA, December 1995.

[24] R. Burns and M. Quinn, "The Cyber-Agent Framework", The Simple Times, 4(3):12–15, 1996.

[25] M. Baldi, S. Gai, and G.P. Picco. "Exploiting Code Mobility in Decentralized and Flexible Network Management", Proc. 1st International Workshop on Mobile Agents (MA'97), Berlin, Germany, April 1997. LNCS 1219:13–26, Springer, Berlin, Germany, 1997.

[26] A. Bieszczad and B. Pagurek, "Towards Plug-and-Play Networks with Mobile Code", Proc. International Conference for Computer Communications (ICCC'97), November 1997, Cannes, France.

[27] A. Bieszczad, T. White, and B. Pagurek, "Mobile Agents for Network Management". IEEE Communications Surveys, 1(1):2–9, 1998.

[28] M. Breugst and T. Magedanz, "Mobile Agents—Enabling Technology for Active Intelligent Network Implementation", IEEE Network, 12(3):53–60, 1998.

[29] J. Schönwälder, "Network management by delegation—From research prototypes towards standards", Computer Networks and ISDN Systems, 29(15):1843–1852, 1997.

[30] M. Baldi and G.P. Picco. "Evaluating the Tradeoffs of Mobile Code Design Paradigms in Network Management Applications", Proc. 20th International Conference on Software Engineering (ICSE'98), Kyoto, Japan, April 1998, pp. 146–155. IEEE, Piscataway, NJ, USA, 1998.

[31] ITU-T. Recommendation X.753, "Information technology—Open Systems Interconnection—Systems management: Command sequencer for systems management", ITU, Geneva, Switzerland, October 1997.

[32] D. Levi and J. Schönwälder (Eds.), "RFC 2592. Definitions of Managed Objects for the Delegation of Management Scripts", IETF, May 1999.

[33] J. Schönwälder, J. Quittek, and C. Kappler, "Building Distributed Management Applications with the IETF Script MIB", IEEE Journal on Selected Areas in Communications, 18(5):702–714, 2000.

[34] D.L. Tennenhouse and D. Wetherall, "Towards an Active Network Architecture", ACM Computer Communication Review, 26(2):5–18, 1996.

[35] T. Yemini and S. da Silva, "Towards Programmable Networks", In Proc. 7th IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM'96), L'Aquila, Italy, October 1996.

[36] M. Brunner and R. Stadler, "Service Management in Multiparty Active Networks", IEEE Communications Magazine, 38(3):144–151, 2000.

[37] R. Kawamura and R. Stadler, "Active Distributed Management for IP Networks", IEEE Communications Magazine, 38(4):114–120, 2000.

[38] S. Ortiz, "Active Networks: The Programmable Pipeline", IEEE Computer, 31(8):19–21, 1998.

[39] K. Psounis, "Active Networks: Applications, Security, Safety, and Architectures", IEEE Communications Surveys, 2(1):2–16, 1999.

[40] D. Raz and Y. Shavitt, "Active Networks for Efficient Distributed Network Management", IEEE Communications Magazine, 38(3):138–143, 2000.

[41] B. Schwartz, A.W. Jackson, W.T. Strayer, W. Zhou, R.D. Rockwell, and C. Partridge, "Smart Packets: Applying Active Networks to Network Management", ACM Transactions on Computer Systems, 18(1):67–88, 2000.

[42] J.M. Smith, K.L. Calvert, S.L. Murphy, H.K. Orman, and L.L. Peterson, "Activating Networks: A Progress Report", IEEE Computer, 32(4):32–41, 1999.

[43] T. Faber, "ACC: Using Active Networking to Enhance Feedback Congestion Control Mechanisms", IEEE Network, 12(3):61–65, 1998.

[44] D.L. Tennenhouse, J.M. Smith, W.D. Sincoskie, D.J. Wetherall, and G.J. Minden, "A Survey of Active Network Research", IEEE Communications Magazine, 35(1):80–86, 1997.

[45] T. Finin, R. Fritzson, D. McKay, and R. McEntire, "KQML as an Agent Communication Language". In N.R. Adam, B.K. Bhargava, and Y. Yesha (Eds.), Proc. 3rd International Conference on Information and Knowledge Management (CIKM'94), Gaithersburg, MD, USA, November 1994, pp. 456–463. ACM Press, New York, NY, USA, 1994.

[46] FIPA, "FIPA ACL Message Structure Specification, Revision XC00061D",. August 2000.

[47] Sunsoft, "Java Management API Architecture. Revision A", September 1996.

[48] Sun Microsystems, "Java Management Extensions White Paper. Revision 01", June 1999.

[49] Sun Microsystems, "Federated Management Architecture (FMA) Specification Version 1.0. Revision 0.4", January 2000

[50] C. Harler, "Web-Based Network Management: Beyond the Browse",. Wiley, New York, NY, USA, 1999.

[51] E. Wilde, "Wilde's WWW. Technical Foundations of the World Wide Web", Springer, Berlin, Germany, 1999.

[52] D. Libes, "Exploring Expect: A Tcl-based Toolkit for Automating Interactive Programs" Publisher: O'Reilly & Associates, Sebastopol, CA, USA, 1994.

[53] T. Oetiker and D. Rand. MRTG, "Multi-Router Traffic Grapher", Available at <http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/mrtg.html>, July 2000.

[54] J.W.K. Hong, S.S. Kwon, and J.Y. Kim, "WebTrafMon: Web-based Internet/Intranet network traffic monitoring and analysis system", Computer Communications, 22(14):1333–1342, 1999.

[55] B. Bruins, "Some Experiences with Emerging Management Technologies". The Simple Times, 4(3):6–8, 1996.

[56] M.C. Maston. "Using the World Wide Web and Java for Network Service Management", Proc. 5th IFIP/IEEE International Symposium on Integrated Network Management (IM'97), San Diego, CA, USA, May 1997, pp. 71–84. Chapman & Hall, London, UK, 1997.

[57] W3C, "Extensible Markup Language (XML) 1.0. W3C Recommendation REC-xml-19980210", Available at <http://http://www.w3.org/TR/1998/REC-xml-19980210>. February 1998.

[58] N. Bradley, "The XML Companion",Publisher: Addison-Wesley, Harlow, UK, 1998.

[59] C.F. Goldfarb and P. Prescod, "The XML Handbook", Publisher: Prentice Hall, Upper Saddle River, NJ, USA, 1998.

[60] D. Megginson, "Structuring XML Documents", Publisher: Prentice Hall, Upper Saddle River, NJ, USA, 1998.

[61] Web Developer's Virtual Library, "XML Specifications, Proposals and Vocabularies", Available at http://wdvl.com/Authoring/Languages/XML/Specifications.html, July 2000.

[62] W3C, "Extensible Markup Language (XML)", Available at http://www.w3.org/XML/, July 2000.

[63] M.A. Mountzia, "A distributed management approach based on flexible agents", Interoperable Communication Networks, 1:99–120, 1998.

[64] M. Post, C.C. Shen, and J. Wei, "The Manager/Agency Paradigm for Distributed Network Management", Proc. 1996 IEEE Network Operations and Management Symposium (NOMS'96), Kyoto, Japan, April 1996, 1:44–53. IEEE, Piscataway, NJ, USA, 1996.

[65] F. Somers, "HYBRID: Unifying Centralised and Distributed Network Management using Intelligent Agents", Proc. 1996 IEEE Network Operations and Management Symposium (NOMS'96), Kyoto, Japan, April 1996. 1:34–43. IEEE, Piscataway, NJ, USA, 1996.

[66] T. Zhang, S. Covaci, and R. Popescu-Zeletin, "Intelligent Agents in Network and Service Management", Proc. IEEE Global Telecommunications Conference (GLOBECOM'96), London, UK, November 1996. 3:1855–1861. IEEE, Piscataway, NJ, USA, 1996.

[67] P. Shafer and R. Enns, "JUNOScript: An XML-based Network Management API", http://www.ietf.org/internet-drafts/draft-shaferjs-xml-api-00.txt, Aug. 27, 2002.

[68] Frank Strauss et al, "A Library to Access SMI MIB Information," http://www.ibr.cs.tu-bs.de/projects/libsmi/.

[69] J.H. Yoon, H.T. Ju, and J.W. Hong, "Development of SNMPXML Translator and Gateway for XML-based Integrated Network Management," Int'l J. of Network Management (IJNM), vol. 13, no. 4, pp. 259-276July-Aug. 2003.

[70] Y.J. Oh, H.T. Ju, M.J. Choi, and J.W. Hong, "Interaction Translation Methods for XML/SNMP Gateway," Proc. DSOM 2002, pp. 54-65, Montreal, Canada, Oct. 2002.

[71] Y.J. Oh, H.T. Ju, and J.W. Hong, "Interaction Translation Methods for XML/SNMP Gateway Using XML Technologies," Proc. Of the Asia-Pacific Network Operations and Management Symp., pp. 11-22, Jeju, Korea, Sept. 2002,.

[72] Avaya Labs., "XML based Management Interface for SNMP Enabled Devices", http://www.research.avayalabs.com/user/ mazum/Projects/XML/.

[73] Network Management Research Group, http://www.ibr.cs.tubs.de/projects/nmrg/.

[74] W3C, "SOAP Version 1.2 Part 0: Primer," W3C Working Draft, Dec. 2001.

[75] M.J. Choi, H.T. Ju, and J.W. Hong, "Towards XML and SNMP Integrated Network Management," Proc. of the Asia-Pacific Network Operations and Management Symp, pp. 507-508. Jeju, Korea, Sept. 2002,

[76] OASIS, "Universal Description, Discovery and Integration (UDDI)", http://www.uddi.org/.

[77] Organization for the Advancement of Structured Information Standards, http://www.oasis-open.org/.

[78] WebMethods, Inc. and Hewlett-Packard Company, "Open Management Interface Specification 1.0, http://www.oasisopen.org/committees/mgmtprotocol/Docs/OMISpecification_1.0rev1_OASIS. pdf".

[79] WBEM, "WBEM Initiative", http://www.dmtf.org/wbem/.

[80] DMTF, "Common Information Model (CIM)", http://www.dmtf.org/standards/standard_cim.php.

[81] IETF, Network Configuration (Netconf), http://www.ietf.org/html.charters/netconf-charter.html.

[82] OSGi Alliance, "The OSGi Service Platform Release 3", March 2003, http://www.osgi.org

[83] OSGi Alliance, "The OSGi Service Platform Release 4 CORE Specification", October 2005, http://www.osgi.org

[84] SOAP: Don Box, et al., Simple Object Access Protocol (SOAP) 1.1, W3C Note, May 2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/ 861

[85] P. Bull and M. Harrison, "Managing Broadband Home Networks", British Telecom Technology Journal, vol.24, no.1, January 2006.

[86] E. Fleury and S. Frenot, "Bulding a JMX Management Interface inside OSGi", INRIA's Research Report, December 2003.

[87] Felix, "OSCAR An OSGi Framework Implementation", http://oscar.objectweb.org/

[88] IBM, "IBM Service Management Framework", http://www-306.ibm.com/software/wireless/smf/.

[89] W3C, "Document Object Model (DOM) Level 1 Specification," W3C Recommendation, Oct. 1998.

[90] Avaya Labs., XML based Management Interface for SNMP Enabled Devices, http://www.research.avayalabs.com/user/ mazum/Projects/XML/.

[91] W3C, "XML Path Language (XPath) Version 2.0," W3C Working Draft, Apr. 2002.

[92] OASIS, "OASIS Web Services Distributed Management (WSDM)", http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsdm.

[93] DSL Forum, "CPE WAN management protocol", Technical Report 069, May 2004.

[94] DSL Forum, "LAN-side CPE configuration", Technical Report 064, May 2004.

[95] DSL Forum, "TR-064 extensions for service differentiation", Technical Report 133, September 2005.

[96] UPnP Forum, "UPnP device architecture 1.0", December 2003, http://www.upnp.org.

[97] W3C, "Namespaces in XML. W3C Recommendation REC-xml-names-19990114, January 1999.

[98] W3C, "Extensible Stylesheet Language (XSL) Version 1.0," W3C Recommendation, Nov. 2000.

[99] J. Schönwälder, "Emerging Internet Management Standards", Tutorial given at the 6th IFIP/IEEE International Symposium on Integrated Network Management (IM'99), Boston, MA, USA, May 1999.

Κεφάλαιο 3
Αποτίμηση και βελτιστοποίηση των επιδόσεων των SOAP/XML/HTTP τεχνολογιών διαχείρισης

3.1. Εισαγωγή

Το προηγούμενο κεφάλαιο παρουσίασε αναλυτικά τους λόγους για τους οποίους, τα τελευταία χρόνια οι SOAP/XML/HTTP τεχνολογίες διαχείρισης έχουν αποκτήσει σοβαρό προβάδισμα έναντι των υπολοίπων, για τη διαχείριση των δικτύων και υπηρεσιών της επόμενης δεκαετίας. Μάλιστα σε επίπεδο προτυποποίησης πρώτη η επιτροπή IAB (Internet Architecture Board) της IETF [1] εξέφρασε τις ανησυχίες της για την καταλληλότητα του SNMP σε μια ειδική για τη διαχείριση δικτύων ημερίδα εργασίας, που οργανώθηκε το καλοκαίρι του 2002. Το βασικότερο ίσως συμπέρασμα της ημερίδας ήταν ότι θα έπρεπε σύντομα οι εργασίες προτυποποίησης να στραφούν σε εναλλακτικές τεχνολογίες διαχείρισης και μάλιστα με κύρια εστίαση σε εκείνες, που βασίζονται στην XML τεχνολογία [2,3]. Τρεις μήνες αργότερα συνεδρίασε η ομάδα NMRG (Network Management Research Group) της IRTF (Internet Research Task Force) [4], η οποία συζήτησε τα προσδοκώμενα αποτελέσματα από την ενδεχόμενη υιοθέτηση μιας ενιαίας XML/HTTP τεχνολογίας διαχείρισης. Όλοι οι συμμετέχοντες συμφώνησαν ότι η αντικατάσταση του SNMP από μια XML/HTTP τεχνολογία διαχείρισης θα αύξανε δραματικά τις δυνατότητες για μοντελοποίηση, την ασφάλεια των συναλλαγών κατά τη διαχείριση αλλά και τις δυνατότητες των παρόχων για μαζική διαχείριση των ρυθμίσεων.

Ωστόσο παρατηρήθηκαν σημαντικές διαφωνίες σχετικά με τις επιπτώσεις, που ενδεχομένως να έχει η εφαρμογή του SOAP/XML στις επιδόσεις της διαχείρισης. Μάλιστα υπήρξαν αρκετοί συμμετέχοντες, που ισχυρίστηκαν ότι οι μεγάλες απαιτήσεις του SOAP/XML σε εύρος ζώνης αλλά και σε επεξεργαστικούς πόρους στον πράκτορα μπορεί να ήταν ακόμα και απαγορευτικές για την ευρεία εφαρμογή μιας τέτοιας τεχνολογίας στη διαχείριση δικτύων και υπηρεσιών. Τελικά εκείνη η συνάντηση έληξε χωρίς να προκύψει κάποιο σχετικό συμπέρασμα, αφού δεν υπήρχαν μελέτες και πειραματικά αποτελέσματα για να αποφανθεί κανείς με βεβαιότητα. Οι συναντήσεις αυτές ενέπνευσαν ορισμένους ερευνητές να διερευνήσουν το ζήτημα της καταλληλότητας της XML/HTTP από την άποψη των επιδόσεων. Από τότε μέχρι και σήμερα έχουν δημοσιευτεί ορισμένες αξιόλογες εργασίες [5-8], που επιχειρούν να ρίξουν φως στις επιδόσεις των XML/HTTP διαχειριστικών λύσεων.

Το παρόν κεφάλαιο, αφού εξετάζει τις παραπάνω εργασίες και διαπιστώνει τις βασικές αδυναμίες και ελλείψεις τους επιχειρεί να φωτίσει ακόμα περισσότερο το θέμα των επιδόσεων αλλά και να συμβάλλει δραστικά στη βελτίωσή τους προκειμένου τελικά το SOAP/XML/HTTP υπόδειγμα να γίνει ευρέως αποδεκτό από την αγορά της διαχείρισης. Αρχικά προσδιορίζονται τα βασικά κριτήρια και οι απαιτήσεις σε επιδόσεις από μια σύγχρονη τεχνολογία διαχείρισης. Μάλιστα μοντελοποιείται το πρόβλημα της διαχείρισης με στόχο την καλύτερη κατανόηση και αποτίμηση των αποτελεσμάτων, που προέκυψαν από τις σχετικές εργασίες του παρελθόντος. Στη συνέχεια αναλύονται οι κυριότερες εξ αυτών και συγκεντρώνονται τα κοινά συμπεράσματα αλλά και οι αδυναμίες τους. Τέλος παρουσιάζεται η εργασία της παρούσας διατριβής και προτείνονται διάφορες προσεγγίσεις βελτίωσης, που τεκμηριώνονται από θεωρητικούς υπολογισμούς αλλά και από πειραματικά αποτελέσματα.

3.2. Κριτήρια επιδόσεων

Οι απαιτήσεις από την τεχνολογία διαχείρισης προκύπτουν κυρίως από τις ανάγκες των παρόχων του δικτύου αλλά και των παρόχων των εφαρμογών και του περιεχομένου οι οποίοι αλληλεπιδρούν άμεσα με τον τελικό χρήστη-συνδρομητή. Παρακάτω αναλύονται τα κύρια κριτήρια των επιδόσεων των τεχνολογιών διαχείρισης, που ακολουθούν το μοντέλο διαχειριστή-πράκτορα και δίνεται μια εικόνα των απαιτήσεων με βάση τα κριτήρια αυτά.

3.2.1. Όγκος διαχειριστικής κίνησης

Στα μελλοντικά περιβάλλοντα διανομής διαδικτυακών υπηρεσιών πολλές υπηρεσίες αναμένεται να προσφέρονται από διαφορετικούς παροχείς και να καταναλώνονται από πολλούς χρήστες, μέσω ενός πλήθους διαφορετικών συσκευών. Για να συμβεί αυτό σχεδόν κάθε δικτυωμένη συσκευή θα πρέπει να είναι διαχειρίσιμη μέσω του IP δικτύου. Μάλιστα κάθε τέτοια συσκευή αναμένεται να επικοινωνεί με το διαχειριστή της τουλάχιστον μια φορά την ημέρα για μεταφορά στατιστικών δεδομένων, ενώ η συχνότητα διαχείρισης μπορεί να γίνει και μια τάξη μεγέθους μεγαλύτερη για λόγους επαναρύθμισης των μεταβλητών κατάστασης των παρεχόμενων υπηρεσιών . Θα πρέπει λοιπόν να δοθεί ιδιαίτερη προσοχή στον όγκο της κίνησης διαχείρισης, που εμφανίζεται στο δημόσιο δίκτυο κατά την επικοινωνία διαχειριστή-πράκτορα, καθώς αυτή μπορεί να επιβαρύνει σημαντικά το δίκτυο χωρίς μάλιστα να προσφέρει κάποιο άμεσο κέρδος (πέραν της παρακολούθησης και ρύθμισης των υπηρεσιών) στον πάροχο.

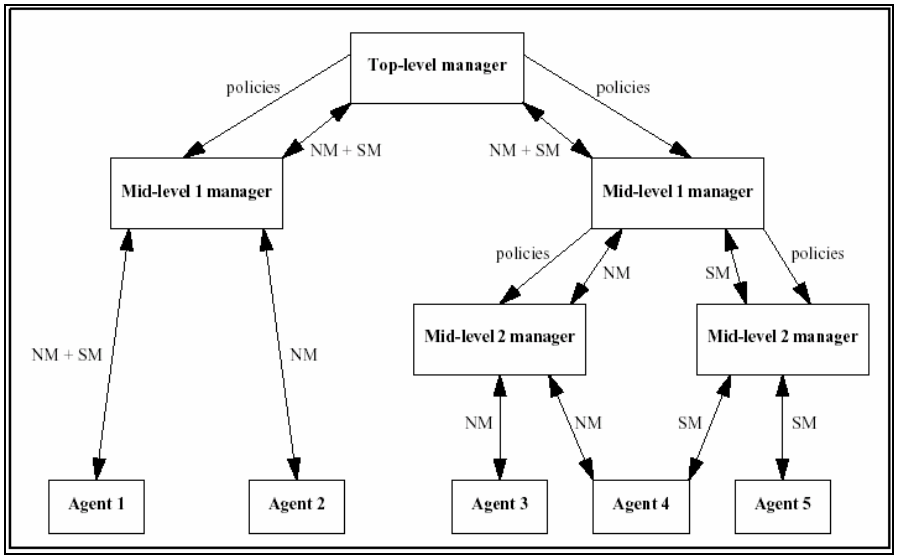
3.2.2. Χρόνος τέλεσης διαχειριστικής ενέργειας

Επιπλέον η τεχνολογία διαχείρισης θα πρέπει να είναι σε θέση να ανταποκριθεί στις μελλοντικές απαιτήσεις των διαφόρων υπηρεσιών και χρηστών για ρύθμιση σε σχεδόν πραγματικό χρόνο. Αυτό πρακτικά σημαίνει ότι όσο πιο σύντομη είναι η διαδικασία ρύθμισης της συσκευής και επομένως η ενεργοποίηση μιας υπηρεσίας, που προσφέρεται από εκείνη τόσο πιο ευχάριστη είναι η εμπειρία που έχει ο τελικός χρήστης. Το γεγονός αυτό σε συνδυασμό με την πλήρη αυτοματοποίηση της διαδικασίας ρύθμισης συμβάλλει στη θετική αξιολόγηση και αποδοχή μιας υπηρεσίας από τους τελικούς χρήστες και επομένως στην

αύξηση των κερδών των παρόχων. Επομένως πρέπει να δοθεί ιδιαίτερη έμφαση στο συνολικό χρόνο διαχείρισης καθώς αυτός εμπειρικά, δεν θα πρέπει να ξεπερνά τα 1-2 δευτερόλεπτα για την ενεργοποίηση ή επαναρύθμιση μιας υπηρεσίας και τα 5-10 δευτερόλεπτα για την πλήρη αρχικοποίηση των ρυθμίσεων (initial configuration) της συσκευής.

3.2.3. Απαιτούμενοι πόροι επεξεργασίας και αποθήκευσης

Προκείμενου να είναι κλιμακώσιμη μια διαχειριστική λύση συνήθως εφαρμόζονται οι αρχιτεκτονικές ιεραρχικής διαχείρισης [9],η αρχή των οποίων δίνεται στο σχήμα 3-1. Σύμφωνα με αυτές οι πάροχοι (top-level manager) μεταβιβάζουν τις υψηλού-επιπέδου πολιτικές διαχείρισης σε ενδιάμεσους διαχειριστές (Mid-level 1 manager, Mid-level 2 manager κλπ.) καθένας εκ των οποίων είναι επιφορτισμένος με διαφορετικά είδη διαχείρισης (διαχείριση των δικτυακών πυλών, διαχείριση Video/VoIP εφαρμογών, κ.α.). Ο τελικός διαχειριστής αναλαμβάνει να επικοινωνήσει με τον πράκτορα κάθε συσκευής και να εφαρμόσει τη χαμηλότερου επιπέδου πολιτική διαχείρισης. Οι τελικοί διαχειριστές φιλοξενούνται σε μηχανήματα με υψηλές επεξεργαστικές και αποθηκευτικές δυνατότητες και μάλιστα είναι γεωγραφικά κατανεμημένοι (π.χ. Mid-level 2 managers) ώστε ο καθένας να έχει υπό την εποπτεία του ένα πεπερασμένο σύνολο συσκευών (Agents 3, 4 και 5). Αυτό συμβαίνει προκειμένου να αποφευχθεί η υπερφόρτωση επεξεργασίας και χρήσης των δικτυακών πόρων του τελικού διαχειριστή και επομένως να μην επηρεάζονται πρακτικά οι χρόνοι διαχείρισης.



Σχήμα 3-1. Ιεραρχικά κατανεμημένη διαχείριση.

Από την άλλη πλευρά οι διαχειριζόμενες συσκευές είναι διαφόρων ειδών με επεξεργαστικές και αποθηκευτικές δυνατότητες, που μπορούν να ποικίλλουν σημαντικά. Ένας πράκτορας μπορεί να εκτελείται σε ένα σύγχρονο προσωπικό υπολογιστή με επεξεργαστική ισχύ πολλών εκατοντάδων MIPS (Million Instruction per Second – Εκατομμύρια εντολές το δευτερόλεπτο) και μνήμη ακόμη και χιλιάδων Megabytes. Ωστόσο ο ίδιος πράκτορας θα πρέπει να είναι δυνατόν να εκτελείται και σε ένα ενσωματωμένο σύστημα χαμηλού κόστους με πολύ περιορισμένες δυνατότητες όπως είναι εκείνο της οικιακής πύλης. Επομένως, μια

σύγχρονη διαχειριστική τεχνολογία οφείλει να είναι εφαρμόσιμη στα περισσότερα είδη πληροφοριακών συσκευών. Αυτό πρακτικά σημαίνει ότι το μέγεθος του εκτελέσιμου κώδικα αλλά και της μνήμης, που καταλαμβάνει η διαχειριστική εφαρμογή του πράκτορα, κατά την εκτέλεσή της θα πρέπει να διατηρείται σε τέτοια επίπεδα που να επιτρέπουν την υποστήριξή της ακόμη και από ενσωματωμένες συσκευές με περιορισμένη μνήμη RAM και ROM. Επίσης οι χρόνοι επεξεργασίας των μηνυμάτων διαχείρισης και της εκτέλεσης των αιτούμενων ενεργειών θα πρέπει να είναι ανεκτοί ακόμα και για συσκευές με μικρές επεξεργαστικές δυνατότητες.

3.3. Περιγραφή και μοντελοποίηση του προβλήματος

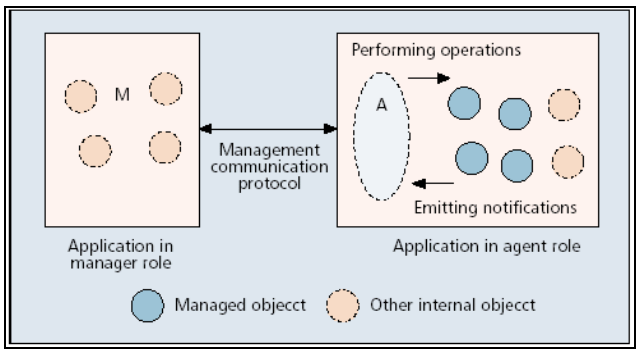
Προκειμένου να αξιολογηθούν οι επιδόσεις των XML/HTTP τεχνολογιών διαχείρισης, με βάση τα παραπάνω κριτήρια η παρούσα ενότητα επιχειρεί να μοντελοποιήσει τη συνολική διαδικασία της διαχείρισης για το μοντέλο διαχειριστή-πράκτορα. Έτσι θα είναι δυνατή η μέτρηση και αξιολόγηση των επιδόσεων για κάθε τεχνολογία που ακολουθεί το μοντέλο διαχειριστή-πράκτορα (όπως είναι το SNMP και όλες όσες βασίζονται στο XML/HTTP).

3.3.1. Το μοντέλο διαχειριστή-πράκτορα

Η ορολογία «διαχειριστής-πράκτορας» χρησιμοποιείται από το πλαίσιο εργασίας του SNMP για να υποδηλώσει τη διαχειρίζουσα και τη διαχειριζόμενη οντότητα, αντίστοιχα. Ωστόσο στις XML/HTTP τεχνολογίες, που έχουν προταθεί μέχρι σήμερα η ορολογία ποικίλει. Για παράδειγμα στο UPnP [10] η διαχειριζόμενη οντότητα ονομάζεται «Συσκευή» (Device) και η διαχειρίζουσα «Σημείο Ελέγχου» (Control Point). Επίσης στο CWMP [11] η διαχειριζόμενη οντότητα ονομάζεται «Πελάτης» (Client), ενώ η διαχειρίζουσα «Εξυπηρετητής Αυτόματης Ρύθμισης» (Automatic Configuration Server - ACS). Για να αποφευχθεί το ενδεχόμενο σύγχυσης, στην παρούσα διατριβή ακολουθείται η ορολογία του SNMP καθώς αυτή χρησιμοποιείται κατά κόρον στον κόσμο της διαχείρισης τα τελευταία δεκαπέντε χρόνια.

Οι διαχειριζόμενοι πόροι μοντελοποιούνται μέσω των «διαχειριζόμενων αντικειμένων» (managed objects). Κάθε αντικείμενο ενθυλακώνει τον υποκείμενο πόρο και προσφέρει μια ιδεατή διεπαφή πρόσβασης. Όλα τα διαχειριζόμενα στοιχεία (δίκτυα ή υπηρεσίες) εκθέτουν από μια συστάδα τέτοιων αντικειμένων (MIB – Management Information Base – Πληροφοριακή Βάση Διαχείρισης), μέσω μιας διεπαφής που παρέχεται από τον πράκτορα. Η διεπαφή τυπικά ορίζεται από τις προδιαγραφές της φύσης των αντικειμένων (π.χ. τύπος, επιτρεπτές τιμές, μήκος μεταβλητών κατάστασης) αλλά και από το πρωτόκολλο διαχείρισης. Προκειμένου οι διαχειριστικές εφαρμογές να υλοποιήσουν τις πολιτικές διαχείρισης προσπελαύνουν τα διαχειριζόμενα αντικείμενα μέσω των διεπαφών. Μια διαχειριστική εφαρμογή μπορεί να είναι είτε ένας πράκτορας, είτε ένας διαχειριστής. Ο πράκτορας είναι μια οντότητα λογισμικού, που διευθύνει τα διαχειριζόμενα αντικείμενα, ανταποκρίνεται στις διαχειριστικές αιτήσεις και ενημερώνει τον διαχειριστή για αυτογενή γεγονότα, που συνέβησαν στη διαχειριζόμενη συσκευή. Προκειμένου να προσπελάσει τα διαχειριζόμενα αντικείμενα, ο πράκτορας χρησιμοποιεί ένα μηχανισμό εξειδικευμένο για κάθε υλοποίηση. Μάλιστα, δεδομένου της συνολικής άποψης, που έχει για το σύνολο των αντικειμένων τα

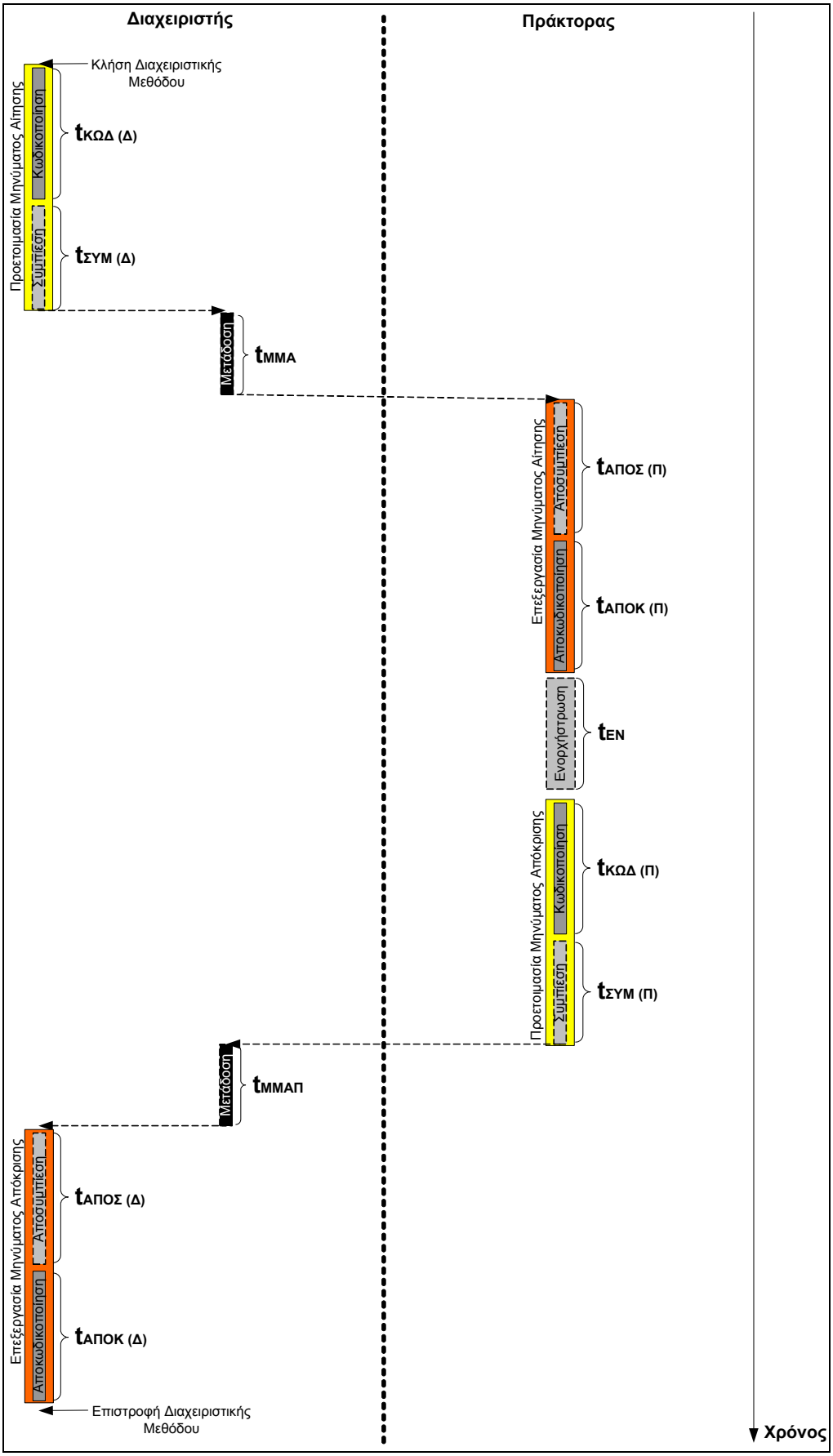
οποία διευθύνει, ο πράκτορας μπορεί να παρέχει ευφυείς τρόπους πρόσβασης στα διαχειριζόμενα αντικείμενα. Οι τρόποι αυτοί συνίστανται κυρίως στη μαζική ή επιλεκτική ανάκτηση των δεδομένων διαχείρισης. Επιπρόσθετα με βάση προκαθορισμένα κριτήρια ο πράκτορας μπορεί να αξιολογεί τα συμβάντα στην πηγή και να ενημερώνει μόνο συγκεκριμένους διαχειριστές,. Το μοντέλο διαχειριστή-πράκτορα αποτελεί ένα πλαίσιο επικοινωνίας κατά το οποίο η προτυποποίηση επηρεάζει μόνο τον τρόπο με τον οποίο μοντελοποιείται και μεταφέρεται η διαχειριστική πληροφορία, ενώ σκόπιμα, δεν προδιαγράφονται λεπτομέρειες σχετικές με την εσωτερική δομή. Η στρατηγική αυτή δίνει τη δυνατότητα για βέλτιστες υλοποιήσεις, δεδομένου ότι δεν υπάρχουν εσωτερικές προγραμματιστικές διεπαφές (APIs – Application Programming Interfaces) με τις οποίες οφείλει να συμμορφώνεται μια υλοποίηση. Αυτό το πλεονέκτημα αποτέλεσε και τη βασικότερη αιτία για την οποία μεταγενέστερες του SNMP τεχνολογίες [11], υιοθέτησαν το ίδιο μοντέλο.



Σχήμα 3-2. Το μοντέλο διαχειριστή-πράκτορα.

3.3.2. Η μοντελοποίηση της διαδικασίας διαχείρισης

Προκειμένου ο διαχειριστής να παρακολουθήσει (monitoring) ή να ρυθμίσει (configuration) την τιμή μιας ή ενός συνόλου μεταβλητών κατάστασης στη διαχειριζόμενη συσκευή «καλεί» την αντίστοιχη μέθοδο στον πράκτορα. Αυτό επιτελείται σύμφωνα με το πρωτόκολλο διαχείρισης με τη δημιουργία και αποστολή ενός μηνύματος διαχείρισης από τον διαχειριστή στον πράκτορα. Το μήνυμα αυτό ονομάζεται «αίτηση» (request) και περιλαμβάνει τόσο την καλούμενη μέθοδο (π.χ. Get/Set) όσο και τις σχετικές παραμέτρους που αντιστοιχούν στις μεταβλητές κατάστασης (π.χ. IPAddress, SubNetmask) και των οποίων η τιμή ζητείται να ληφθεί η να τεθεί. Το διαχειριστικό μήνυμα μεταδίδεται μέσω του IP δικτύου είτε με τη μορφή UDP πακέτων, όπως συμβαίνει στην περίπτωση του SNMP, είτε με τη μορφή HTTP/TCP πακέτων όπως συμβαίνει στις XML/HTTP τεχνολογίες. Στη συνέχεια η εφαρμογή του πράκτορα παραλαμβάνει και επεξεργάζεται το μήνυμα. Ανάλογα με την καλούμενη μέθοδο και τις παραμέτρους της ο πράκτορας καλεί μία ή ένα σύνολο εσωτερικών μεθόδων, που είναι στενά συνδεδεμένες με την κάθε υλοποίηση και οι οποίες αναλαμβάνουν να εκτελέσουν την αιτούμενη ενέργεια. Οι μέθοδοι αυτοί τυπικά ονομάζονται μέθοδοι ενορχήστρωσης καθώς στόχος τους είναι ο διαρκής συγχρονισμός των αντικειμένων διαχείρισης (MIB) με τους πραγματικούς πόρους του συστήματος.



Σχήμα 3-3. Η μοντελοποίηση της διαδικασίας διαχείρισης.

Αφού ολοκληρωθεί η εκτέλεση των μεθόδων ενορχήστρωσης ο πράκτορας αναλαμβάνει να αποστείλει ένα μήνυμα «απάντησης» (response) το οποίο θα περιέχει είτε τις ανακτημένες τιμές των μεταβλητών κατάστασης, είτε το αποτέλεσμα της ανάθεσης των νέων τιμών στις μεταβλητές κατάστασης.

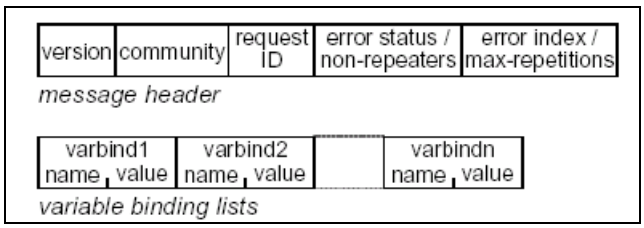
Το σχήμα 3-3 απεικονίζει λεπτομερώς τη συνολική διαδικασία. Τρεις είναι οι βασικές λειτουργίες που μπορούν να επιδράσουν καθοριστικά στις επιδόσεις της διαχειριστικής τεχνολογίας: (i) η κωδικοποίηση των μηνυμάτων διαχείρισης, (ii) η μετάδοσή τους και (iii) η εκτέλεση της υπαγορευόμενης διαχειριστικής ενέργειας (ενορχήστρωση). Μάλιστα, στη συνολική διαδικασία είναι δυνατόν να συμμετάσχει και η λειτουργία της συμπίεσης των μηνυμάτων η οποία μπορεί να χρησιμοποιηθεί για να βελτιώσει τις επιδόσεις μιας τεχνολογίας διαχείρισης χωρίς ωστόσο αυτή να είναι πάντα επικερδής. Το παρόν κεφάλαιο ασχολείται εκτεταμένα (παράγραφος 3.5) με τη διαδικασία αυτή και ορίζει τις συνθήκες υπό τις οποίες θα πρέπει να χρησιμοποιείται η συμπίεση προκειμένου να είναι πάντα επικερδής. Παρακάτω αναλύεται η επίδραση των προαναφερθέντων λειτουργιών στις επιδόσεις της διαχειριστικής τεχνολογίας.

3.3.2.1. Η κωδικοποίηση του μηνύματος διαχείρισης

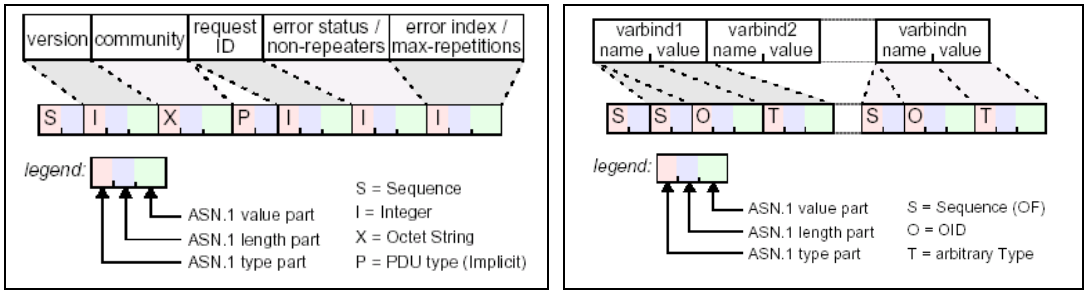
Πρόκειται για τη διαδικασία εκείνη κατά την οποία ο διαχειριστής συνθέτει το μήνυμα της αίτησης, ενώ ο πράκτορας συνθέτει το μήνυμα της απάντησης. Η κωδικοποίηση είναι άμεσα συνδεμένη με τη διαχειριστική τεχνολογία και ορίζει τον τρόπο με τον οποίο εκφράζονται οι καλούμενες μέθοδοι και οι αντίστοιχες παράμετροί τους .

Τα SNMP μηνύματα καθορίζονται με βάση τις δομές που ορίζει το πρότυπο ASN.1 (Abstract Syntax Notation One) [12] και μεταφέρονται στο δίκτυο με τη χρήση των βασικών κανόνων κωδικοποίησης (BER – Basic Encoding Rules) [13]. Ο τρόπος της κωδικοποίησης ASN.1 είναι προσανατολισμένος στη δυαδική απεικόνιση (bit-oriented) και αυτό έχει σαν αποτέλεσμα τη μεταφορά μεγάλης ποσότητας πληροφορίας με μικρά μηνύματα διαχείρισης και επομένως τη βέλτιστη χρήση του διαθέσιμου εύρους ζώνης. Η δομή ενός SNMP μηνύματος δίνεται αναλυτικά από το σχήμα 3-4 [14,15]. Κάθε μήνυμα αποτελείται από δύο μέρη: (i) την κεφαλίδα (header) και (ii) τη λίστα συνδεδεμένων μεταβλητών (variable binding list). Η κεφαλίδα απαρτίζεται από πέντε πεδία, ενώ η λίστα αποτελείται από n ζεύγη συνδεδεμένων μεταβλητών (varbinds) καθένα εκ των οποίων μεταφέρει το όνομα του αντικειμένου (OID – Object Identifier) και την τιμή του. Όπως φαίνεται στο σχήμα 3-5 κάθε κωδικοποιημένο στοιχείο αποτελείται από τρία μέρη: (i) τον ASN.1 τύπο, (ii) το μήκος και (iii) την τιμή. Στις περισσότερες περιπτώσεις η κωδικοποίηση BER καταλαμβάνει μία οκτάδα για τον τύπο και άλλη μία για το μήκος της μεταβλητής. Όσον αφορά στην πληροφορία που καταλαμβάνει μια ASN.1 τιμή, αυτή ποικίλει ανάλογα με τον τύπο και την τιμή της μεταβλητής . Το μεγαλύτερο όγκο συνήθως καταλαμβάνουν τα ονόματα των μεταβλητών, που εκφράζονται με ένα κωδικό αναγνώρισης του διαχειριστικού αντικειμένου. Ο κωδικός αυτός συνίσταται συνήθως από μεγάλες αλληλουχίες αριθμών της ακόλουθης μορφής: OID=1.3.6.1.2.1.1.0. Κάθε τέτοια τιμή χρειάζεται για να κωδικοποιηθεί τόσες οκτάδες όσο

είναι το μήκος της μείον 1 (αφού οι πρώτοι δύο κόμβοι κωδικοποιούνται σε μια απλή οκτάδα).



Σχήμα 3-4. Η δομή ενός SNMP (v1/v2c) μηνύματος.



(α) (β)

Σχήμα 3-5. Η κωδικοποίηση BER του SNMP. (α) Η κεφαλίδα. (β) Η λίστα σύνδεσης των μεταβλητών.

Αντίθετα, η XML τεχνολογία είναι προσανατολισμένη στη χρήση ολόκληρων χαρακτήρων (character-oriented) για να εκφράσουν τη διαχειριστική πληροφορία. Αυτό έχει ως άμεση συνέπεια την περιττή χρήση του εύρους ζώνης κατά τη μεταφορά της διαχειριστικής πληροφορίας. Αναλυτικότερα, οι XML/HTTP τεχνολογίες συνήθως χρησιμοποιούν το πρωτόκολλο SOAP [11] προκειμένου να κωδικοποιήσουν τις μεθόδους και τις παραμέτρους τους. Το SOAP εν συνεχεία χρησιμοποιεί τη γλώσσα XML για να δομήσει και να εκφράσει αυτή την πληροφορία. Το σχήμα 3-6 απεικονίζει μια CWMP διαχειριστική εντολή (*SetParameterValues*) η οποία απενεργοποιεί την πρόσβαση της δικτυακής πύλης στο δημόσιο δίκτυο θέτοντας μια μεταβλητή κατάστασης (*InternetGatewayDevice.WANDevice.1. WANCommonInterfaceConfig.EnabledForInternet = false*).

Παρόλο που στο σημείο αυτό θα μπορούσε να δοθεί μια θεωρητική ανάλυση για την ποσοτική σύγκριση του όγκου της διαχειριστικής πληροφορίας μεταξύ του SNMP και του CWMP, ωστόσο εσκεμμένα αποφεύγεται κάτι τέτοιο. Η αιτία είναι ότι τα πραγματικά μεγέθη μπορεί να ποικίλουν σημαντικά ανάλογα με τη μοντελοποίηση της διαχειριζόμενης συσκευής αλλά και τις ενέργειες του διαχειριστή. Ωστόσο μια ποιοτική σύγκριση των διαφορετικών τεχνολογιών ASN.1 και SOAP/XML θα είχε ίσως περισσότερο νόημα καθώς είναι πιο γενική και έτσι θα κάλυπτε όχι μόνο το CWMP αλλά και όλα τα SOAP/XML/HTTP πρωτόκολλα διαχείρισης.

Καταρχάς, κάθε SOAP φάκελος περιλαμβάνει τουλάχιστον την ακόλουθη πληροφορία:

(1) Το αναγνωριστικό πρόθεμα (namespace) του φακέλου (π.χ. "*http://schemas.xmlsoap.org/ soap/envelope/*")

(2) Το αναγνωριστικό πρόθεμα κωδικοποίησης (π.χ. "*http://schemas.xmlsoap.org/soap/ encoding* ")

(3) Το αναγνωριστικό πρόθεμα του διαχειριστικού πρωτοκόλλου (π.χ. "*urn:dslforum-org:cwmp-1-0* ")

(4) Την SOAP κεφαλίδα (SOAP header) η οποία χρησιμοποιείται από το διαχειριστικό πρωτόκολλο κυρίως για λόγους συγχρονισμού της εκτέλεσης των διαχειριστικών εντολών.

Για να δοθεί μια πρώτη εικόνα σύγκρισης του όγκου σε επίπεδο κεφαλίδας αξίζει να τονιστεί ότι η κεφαλίδα των παραπάνω SNMP (v1/v2c) μηνυμάτων δεν ξεπερνά τις 40 οκτάδες, σε αντίθεση με την πληροφορία των τεσσάρων παραπάνω SOAP πεδίων η οποία τις περισσότερες φορές υπερβαίνει τις 250 οκτάδες.

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
<soap:Header>
<cwmp: ID soap: mustUnderstand="1">1567</cwmp:ID>
</soap:Header>
<soap:Body>
<cwmp:SetPameterValues>
<ParameterList soap: arrayType="cwmp:ParameterValueStruct[1]">
<ParameterValueStruct>
<name>InternetGatewayDevice.WANDevice.1.WAN-CommonInterfaceConfig.EnabledForInternet</name>
<value xsi:type="xsd:boolean">false</value>
</ParameterValueStruct>
</ParameterList>
</cwmp:SetPameterValues>
</soap:Body>
</soap:Envelope>
```

Σχήμα 3-6. Ένα CWMP μήνυμα κωδικοποιημένο σε SOAP/XML.

Επίσης, ενώ στην ASN.1 κωδικοποίηση του SNMP ο τύπος μιας μεταβλητής καταλαμβάνει τις περισσότερες φορές 1 οκτάδα, αντίθετα σε μια SOAP/XML κωδικοποίηση ο τύπος της μεταβλητής εκφράζεται με μια σειρά χαρακτήρων (π.χ. *xsi:type= "xsd:booolean*) η οποία συνήθως καταλαμβάνει τουλάχιστον 20 οκτάδες. Επιπρόσθετα το όνομα των μεταβλητών στην κωδικοποίηση SOAP/XML εκφράζεται με τη χρήση των πραγματικών ονομάτων των αντικειμένων, που έχουν προκύψει από τη μοντελοποίηση και όχι με κωδικούς αναγνώρισης όπως στο SNMP. Αυτό μπορεί να οδηγήσει σε μεγέθη ονομάτων, που ξεπερνούν κατά πολύ τις 15-20 οκτάδες του SNMP, όπως στην περίπτωση του σχήματος 3-5, όπου το όνομα της μεταβλητής ξεπερνά τις 75 οκτάδες. Ένα επίσης χαρακτηριστικό στοιχείο της XML κωδικοποίησης είναι ότι η XML είναι μια αυτό-επεξηγούμενη γλώσσα και για αυτό το λόγο χρησιμοποιεί ετικέτες (tags) για οτιδήποτε πρέπει να δηλωθεί. Για παράδειγμα, προκειμένου η XML να δηλώσει ότι το πεδίο που ακολουθεί είναι το όνομα μια μεταβλητής χρησιμοποιεί τις ετικέτες *<name>* και *</name>*. Το ίδιο συμβαίνει και για να δηλώσει την τιμή μιας μεταβλητής (*<value></value>*), την κεφαλίδα ενός SOAP φακέλου (*<soap:Header></ soap:Header>*) και γενικά οποιοδήποτε πεδίο ενός SOAP φακέλου. Αυτή η ιδιαιτερότητα επιφέρει πρόσθετο υπερκεφαλισμό στα μηνύματα διαχείρισης. Επομένως ο όγκος των

SOAP/XML μηνυμάτων διαχείρισης αναμένεται σημαντικά υψηλότερος σε σχέση με εκείνο των SNMP μηνυμάτων. Τα πειράματα σχετικών εργασιών, που περιγράφονται στην παράγραφο 3.4 επιχειρούν να δώσουν την τάξη μεγέθους των διαφορών αυτών.

Πέραν της επίδρασης που έχει η κωδικοποίηση στον όγκο της διαχειριστικής πληροφορίας οι χρόνοι διαχείρισης μπορεί επίσης να συνδέονται με το είδος της κωδικοποίησης. Στο σχήμα 3-3 φαίνεται ότι ο διαχειριστής κωδικοποιεί τη διαχειριστική αίτηση σε χρόνο $t_{ΚΩΔ(Δ)}$ (χρόνος κωδικοποίησης στο διαχειριστή) και αποκωδικοποιεί το μήνυμα απόκρισης, που λαμβάνει από τον πράκτορα σε χρόνο $t_{ΑΠΟΚ(Δ)}$ (χρόνος αποκωδικοποίησης στο διαχειριστή). Από την άλλη πλευρά ο πράκτορας αποκωδικοποιεί τη διαχειριστική αίτηση σε χρόνο $t_{ΑΠΟΚ(Π)}$ (χρόνος αποκωδικοποίησης στον πράκτορα) και αφού εκτελεί τη διαχειριστική ενέργεια κωδικοποιεί το αποτέλεσμά της σε χρόνο $t_{ΚΩΔ(Π)}$ (χρόνος κωδικοποίησης στον πράκτορα). Οι χρόνοι αυτοί πέραν του είδους της κωδικοποίησης εξαρτώνται βέβαια και από την επεξεργαστική ισχύ του συστήματος στο οποίο εκτελείται η εφαρμογή διαχείρισης. Επομένως για να αποκτηθεί μια εικόνα όσον αφορά στη διαφορά, που επιφέρει η κωδικοποίηση της SOAP/XML σε σχέση με την κωδικοποίηση BER θα πρέπει οι δύο τεχνολογίες να δοκιμαστούν στο ίδιο σύστημα και κάτω από τις ίδιες συνθήκες. Τέτοια πειράματα δίνονται αναλυτικά στην παράγραφο 3-4.

3.3.2.2. Η μετάδοση του μηνύματος διαχείρισης

Μόλις ο διαχειριστής ετοιμάσει το μήνυμα-αίτηση, το αποστέλλει μέσω του IP δικτύου στον πράκτορα ο οποίος το λαμβάνει μετά από χρόνο $t_{ΜΜΑ}$ (χρόνος μετάδοσης μηνύματος αίτησης). Αφού ο πράκτορας εκτελέσει τη διαχειριστική ενέργεια, επιστρέφει το μήνυμα-απάντηση στο διαχειριστή ο οποίος το λαμβάνει μετά από χρόνο $t_{ΜΜΑΠ}$ (χρόνος μετάδοσης μηνύματος απόκρισης). Οι δύο αυτοί χρόνοι είναι δυνατόν να επηρεάσουν σημαντικά το συνολικό χρόνο διαχείρισης, ιδιαίτερα σε περιπτώσεις όπου το διαθέσιμο εύρος ζώνης είναι περιορισμένο. Πέραν του διαθέσιμου εύρους ζώνης οι χρόνοι αυτοί εξαρτώνται και από τον όγκο των διαχειριστικών δεδομένων και συνεπώς και από τον τρόπο μοντελοποίησης αλλά και από το είδος της κωδικοποίησης, που χρησιμοποιεί η τεχνολογία διαχείρισης. Επίσης οι χρόνοι αυτοί εξαρτώνται και από το πρωτόκολλο, που φέρει το διαχειριστικό μήνυμα. Στην περίπτωση του SNMP η χρήση του UDP είναι λιγότερο δαπανηρή σε δικτυακούς πόρους από ότι οι HTTP τεχνολογίες. Αυτό οφείλεται κυρίως στην τριμερή χειραψία του TCP πρωτοκόλλου αλλά και των σχετικά μεγάλων υπερκεφαλισμών του HTTP. Στη παράγραφο 3-5 γίνεται λεπτομερής ανάλυση του υπερκεφαλισμού των HTTP/XML τεχνολογιών διαχείρισης και δίνεται η τάξη μεγέθους κατά την οποία η διαδικασία μετάδοσης των μηνυμάτων επιδρά στο συνολικό χρόνο διαχείρισης.

3.3.2.3. Η ενορχήστρωση

Η διαδικασία αυτή αφορά στην εκτέλεση της διαχειριστικής ενέργειας από τον πράκτορα στη συσκευή που τον φιλοξενεί. Όπως ειπώθηκε και παραπάνω για τις τεχνολογίες, που ακολουθούν το μοντέλο διαχειριστή-πράκτορα η διαδικασία της ενορχήστρωσης δεν προτυποποιείται προκειμένου να επιτρέψει βέλτιστες υλοποιήσεις. Επομένως η ενορχήστρωση δεν έχει να κάνει με τις τεχνολογίες διαχείρισης, που εξετάζει η παρούσα

εργασία αλλά με τον τρόπο υλοποίησης των διαχειριστικών ενεργειών σε κάθε περίπτωση. Βέβαια, όπως θα φανεί και στη παράγραφο 3-4, σε περιπτώσεις όπου ο σχεδιασμός και η υλοποίηση της ενορχήστρωσης αγνοούν τις χρονικές επιδόσεις (t_{EN} –χρόνος ενορχήστρωσης) η λειτουργία αυτή μπορεί να αποτελέσει τον κυριότερο παράγοντα καθυστέρησης στη συνολική διαχειριστική διαδικασία.

3.3.2.4. Η συμπίεση

Η λειτουργία της συμπίεσης των μηνυμάτων διαχείρισης είναι μια προαιρετική λειτουργία η οποία ωστόσο μπορεί να έχει άμεση επίδραση στην μείωση του όγκου της διαχειριστικής πληροφορίας. Μάλιστα ανάλογα με το βαθμό της συμπίεσης και το διαθέσιμο εύρος ζώνης μπορεί να επηρεάσει σημαντικά και το συνολικό χρόνο διαχείρισης. Στο σημείο αυτό πρέπει να τονιστεί ότι η συμπίεση μπορεί να επηρεάσει θετικά αλλά και αρνητικά το χρόνο διαχείρισης. Αυτό εξαρτάται από το εάν ο συνολικός χρόνος συμπίεσης (χρόνος συμπίεσης στο διαχειριστή - $t_{ΣΥΜ(Δ)}$ + χρόνος συμπίεσης στον πράκτορα - $t_{ΣΥΜ(Π)}$) και αποσυμπίεσης των μηνυμάτων (χρόνος αποσυμπίεσης στον πράκτορα - $t_{ΑΠΟΣ(Π)}$ + χρόνος αποσυμπίεσης στο διαχειριστή - $t_{ΑΠΟΣ(Δ)}$) είναι μικρότερος ή μεγαλύτερος από το χρονικό κέρδος, που επιφέρει η συμπίεση των μηνυμάτων κατά τη μετάδοσή τους στο δίκτυο. Εάν συμβολιστούν με $Gt_{ΜΜΑ}$ και $Gt_{ΜΜΑΠ}$ τα χρονικά κέρδη, που επιφέρει η συμπίεση στους χρόνους μετάδοσης του μηνύματος αίτησης και απόκρισης αντίστοιχα, τότε το συνολικό κέρδος $Gt_{ΟΛ}$ της συμπίεσης στο χρόνο διαχείρισης μπορεί να εκφραστεί με την ακόλουθη σχέση:

$$Gt_{ΟΛ} = Gt_{ΜΜΑ} + Gt_{ΜΜΑΠ} - (t_{ΣΥΜ(Δ)} + t_{ΑΠΟΣ(Π)} + t_{ΣΥΜ(Π)} + t_{ΑΠΟΣ(Δ)}) \quad (3\text{-}1)$$

Εάν συμβολιστούν με $S_{ΜΑ}$ και $S_{ΜΑΠ}$ τα μεγέθη του μηνύματος αίτησης και απόκρισης αντίστοιχα, ενώ με $S'_{ΜΑ}$ και $S'_{ΜΑΠ}$ τα νέα μεγέθη μετά τη συμπίεση τότε τα χρονικά κέρδη μπορούν να δοθούν από την ακόλουθη σχέση:

$$Gt_{ΜΜΑ} = (S_{ΜΑ} – S'_{ΜΑ}) / LinkRate \quad (3\text{-}2) \qquad Gt_{ΜΜΑΠ} = (S_{ΜΑΠ} – S'_{ΜΑΠ}) / LinkRate \quad (3\text{-}3)$$

Οι παραπάνω σχέσεις θα μπορούσαν να εκφραστούν συναρτήσει του λόγου συμπίεσης (C_{ratio}) ο οποίος εκφράζει την αποτελεσματικότητα του κάθε αλγορίθμου συμπίεσης και ορίζεται ως ο λόγος του μεγέθους του συμπιεσμένου μηνύματος προς το μέγεθος του αρχικού μηνύματος. Επομένως για τη συμπίεση του μηνύματος αίτησης θα ισχύει: $C_{ratioΜΑ} = S'_{ΜΑ} / S_{ΜΑ}$ ενώ για τη συμπίεση του μηνύματος απόκρισης θα ισχύει: $C_{ratioΜΑΠ} = S'_{ΜΑΠ} / S_{ΜΑΠ.}$. Με βάση λοιπόν αυτούς τους ορισμούς οι σχέσεις 3-2 και 3-3 παίρνουν την ακόλουθη μορφή:

$$Gt_{ΜΜΑ} = S_{ΜΑ} (1– C_{ratioΜΑ}) / LinkRate \quad (3\text{-}4) \qquad Gt_{ΜΜΑΠ} = S_{ΜΑΠ} (1– C_{ratioΜΑΠ}) / LinkRate \quad (3\text{-}5)$$

Πρέπει να τονιστεί ότι ο λόγος συμπίεσης C_{ratio} πέραν του αλγορίθμου εξαρτάται και από τη φύση (π.χ. αρχεία με επαναλήψεις χαρακτήρων, αρχεία χαμηλής εντροπίας) αλλά και από τον όγκο των προς συμπίεση δεδομένων. Στην παράγραφο 3-5 θα γίνει λεπτομερής αξιολόγηση των επιδόσεων των διαθέσιμων αλγορίθμων συμπίεσης κατά την εφαρμογή τους στις XML/HTTP διαχειριστικές τεχνολογίες.

Επίσης, εάν οι ταχύτητες συμπίεσης στο διαχειριστή και στον πράκτορα συμβολιστούν με $U_{c(\Delta)}$ και$U_{c\,(\Pi)}$ αντίστοιχα, ενώ οι ταχύτητες αποσυμπίεσης με $U_{D\,(\Delta)}$ και$U_{D\,(\Pi)}$ τότε οι χρόνοι της σχέσης 3-1 ορίζονται ως εξής:

$$t_{\Sigma\Upsilon M(\Delta)} = S_{MA} / U_{c\,(\Delta)} \quad (3\text{-}6) \qquad t_{\Alpha\Pi O\Sigma(\Pi)} = S'_{MA} / U_{D\,(\Pi)} \quad (3\text{-}7)$$

$$t_{\Sigma\Upsilon M(\Pi)} = S_{MA\Pi} / U_{c\,(\Pi)} \quad (3\text{-}8) \qquad t_{\Alpha\Pi O\Sigma(\Delta)} = S'_{MA\Pi} / U_{D\,(\Delta)} \quad (3\text{-}9)$$

Στο σημείο αυτό πρέπει να τονιστεί ότι η διαδικασία της συμπίεσης μπορεί να λαμβάνει χώρα και τμηματικά μέσα στη διαχειριστική διαδικασία. Κάτι τέτοιο είναι δυνατό να συμβεί με τους παρακάτω δύο εναλλακτικούς τρόπους:

(1) Μόνο ο διαχειριστής συμπιέζει το μήνυμα-αίτηση, ενώ ο πράκτορας το αποσυμπιέζει. Στην περίπτωση αυτή το συνολικό κέρδος χρόνου δίνεται από την ακόλουθη σχέση:

$$Gt_{O\Lambda} = Gt_{MMA} - (t_{\Sigma\Upsilon M(\Delta)} + t_{\Alpha\Pi O\Sigma(\Pi)}) \quad (3\text{-}10)$$

Η εναλλακτική αυτή θα μπορούσε να χρησιμοποιηθεί στην περίπτωση που το άθροισμα $Gt_{MMA} - (t_{\Sigma\Upsilon M(\Delta)} + t_{\Alpha\Pi O\Sigma(\Pi)})$ είναι θετικό, ενώ το άθροισμα $Gt_{MMA\Pi} - (t_{\Sigma\Upsilon M(\Pi)} + t_{\Alpha\Pi O\Sigma(\Delta)})$ είναι μηδέν η αρνητικό και επομένως η συμπίεση στον πράκτορα είτε δεν θα προσέφερε τίποτα στο χρονικό κέρδος (κατανάλωνε άσκοπα τους επεξεργαστικούς πόρους του διαχειριστή και του πράκτορα), είτε θα μείωνε το κέρδος στο συνολικό χρόνο διαχείρισης.

(2) Μόνο ο πράκτορας συμπιέζει το μήνυμα-απάντηση ενώ ο διαχειριστής το αποσυμπιέζει. Στην περίπτωση αυτή το συνολικό κέρδος χρόνου δίνεται από την ακόλουθη σχέση:

$$Gt_{O\Lambda} = Gt_{MMA\Pi} - (t_{\Sigma\Upsilon M(\Pi)} + t_{\Alpha\Pi O\Sigma(\Delta)}) \quad (3\text{-}11)$$

Με τον ίδιο τρόπο αυτή η εναλλακτική θα μπορούσε να χρησιμοποιηθεί στην περίπτωση που το άθροισμα $Gt_{MMA\Pi} - (t_{\Sigma\Upsilon M(\Pi)} + t_{\Alpha\Pi O\Sigma(\Delta)})$ είναι θετικό ενώ το άθροισμα $Gt_{MMA} - (t_{\Sigma\Upsilon M(\Delta)} + t_{\Alpha\Pi O\Sigma(\Pi)})$ μηδέν η αρνητικό.

Συμπερασματικά, η χρησιμότητα ενός δεδομένου αλγορίθμου συμπίεσης στη διαχείριση εξαρτάται από τον ίδιο το αλγόριθμο (λόγοι και ταχύτητες συμπίεσης), από την τεχνολογία διαχείρισης (όγκος και φύση διαχειριστικών δεδομένων) αλλά και από τις συνθήκες διαχείρισης (διαθέσιμοι πόροι δικτύου, επεξεργαστικές και αποθηκευτικές δυνατότητες του διαχειριστή και του πράκτορα, διαχειριστικές ενέργειες). Στην παράγραφο 3-5 θα γίνει μια εκτεταμένη ανάλυση με στόχο τη διερεύνηση των συνθηκών κάτω από τις οποίες η εφαρμογή ενός κατάλληλου αλγορίθμου συμπίεσης (Lempel-Ziv) είναι επικερδής για τις τεχνολογίες XML/HTTP.

3.4. Σχετικές εργασίες

Η παραπάνω ανάλυση έδωσε τα βασικά κριτήρια αξιολόγησης των επιδόσεων μιας διαχειριστικής τεχνολογίας και μοντελοποίησε τη διαδικασία διαχείρισης προκειμένου να διευκολύνει τη σωστή τοποθέτηση των απαντήσεων, που δίνουν οι πρόσφατες έρευνες στο συνολικό πρόβλημα. Στη βιβλιογραφία υπάρχουν εργασίες, που διερευνούν τις επιδόσεις του

SNMP [16], συγκρίνουν τις επιδόσεις του SOAP με άλλες τεχνολογίες μεσισμικού [17], αναλύουν τις επιδόσεις της SOAP επεξεργασίας [18] και συγκρίνουν τις επιδόσεις διαφορετικών εργαλείων υλοποίησης SOAP και υπηρεσιών ιστού [19]. Ωστόσο η παρούσα ενότητα θα εστιάσει στις εργασίες, που ασχολούνται ειδικά με το θέμα των επιδόσεων της XML/HTTP τεχνολογίας κατά τη διαχείριση δικτύων και υπηρεσιών. Μάλιστα οι σχετικές εργασίες μπορούν να διακριθούν σε δύο κατηγορίες: (i) σε εκείνες που χρησιμοποιούν ένα συνδυασμό XML/HTTP διαχειριστών, SNMP πρακτόρων, και SNMP-XML πυλών και μετρούν τις επιδόσεις του SNMP και του XML σε κάθε άκρο της πύλης και (ii) σε εκείνες, που αντί για πύλες υλοποιούν ξεχωριστούς SNMP και XML/HTTP διαχειριστές και πράκτορες και μετρούν σε κάθε περίπτωση τις επιδόσεις της διαχειριστικής τεχνολογίας.

3.4.1. Εργασίες με χρήση SNMP – XML πυλών

3.4.1.1. Ricardo Neisse and Lisandro Granville

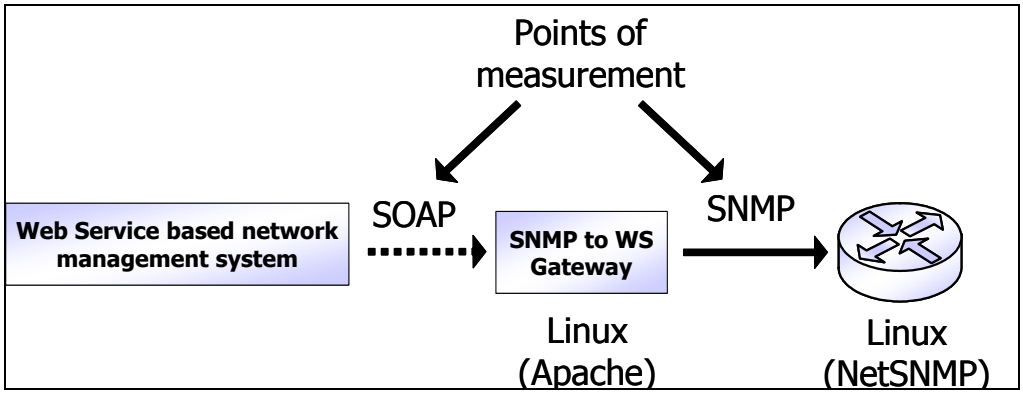
Στις αρχές του 2004 δημοσιεύτηκε η εργασία [7], που προτείνει αρχιτεκτονικές και υλοποιήσεις για ένα ολοκληρωμένο XML/HTTP σύστημα διαχείρισης του ήδη εγκατεστημένου SNMP εξοπλισμού (εξελικτική προσέγγιση). Συγκεκριμένα, αναλύονται οι λειτουργίες των XML/HTTP διαχειριστών αλλά και της προτεινόμενης XML-SNMP πύλης, που πραγματοποιεί τη μετάφραση και μετάδοση των XML μηνυμάτων στους SNMP πράκτορες. Οι συγγραφείς προτείνουν δύο εναλλακτικά σχήματα μετάφρασης των SOAP μηνυμάτων σε SNMP: (i) σε επίπεδο πρωτοκόλλου (protocol-level) και (ii) σε επίπεδο αντικειμένου (object-level). Κατά το πρώτο για κάθε SNMP μήνυμα υπάρχει και ένα αντίστοιχο SOAP μήνυμα. Για την πραγματοποίηση του πρώτου σχήματος υλοποιήθηκαν οι ακόλουθες απομακρυσμένες διεργασίες (SOAP RPCs), που υποστηρίζονται από την XML-SNMP πύλη:

- *function Get ($ip, $community, $oid)*
- *function GetNext ($ip, $community, $oid)*
- *function Set ($ip, $community, $oid, $newvalue)*

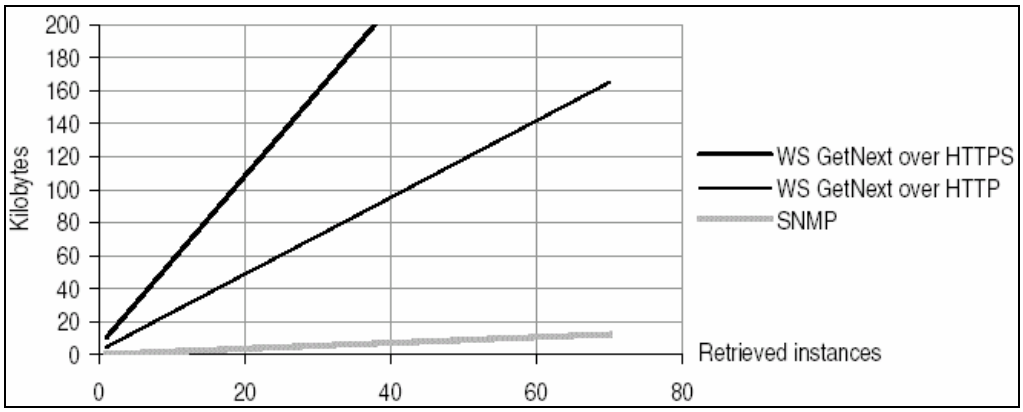
Αντίθετα, κατά το δεύτερο σχήμα ένα απλό, υψηλού επιπέδου SOAP μήνυμα αντιστοιχεί σε πολλά SNMP μηνύματα. Μια τέτοια πύλη είναι ενήμερη για τα διαχειριστικά αντικείμενα, που υποστηρίζει η κάθε SNMP συσκευή και παρουσιάζει αυτά τα αντικείμενα σαν υπηρεσίες ιστού. Για παράδειγμα, η μέθοδος *GetIfTable* ανακτά ολόκληρο τον πίνακα διεπαφών (IfTable) ο οποίος αποτελείται από πολλές μεταβλητές κατάστασης.

Για κάθε ένα από τα προαναφερθέντα μεταφραστικά σχήματα οι συγγραφείς μετρούν και συγκρίνουν τον όγκο της παραγόμενης κίνησης διαχείρισης στα δύο άκρα της πύλης, με τον τρόπο που απεικονίζει το σχήμα 3-7. Ο διαχειριστής (Web Service based network management system) καλεί απομακρυσμένα (με χρήση των SOAP RPCs) τις διεργασίες διαχείρισης στην πύλη (SNMP to WS Gateway). Η πύλη με βάση την καλούμενη μέθοδο στέλνει τις κατάλληλες SNMP εντολές στον πράκτορα (Linux Net-SNMP[20]) της διαχειριζόμενης συσκευής. Αφού εκτελεστεί η αιτούμενη ενέργεια, ο πράκτορας στέλνει

πίσω στην πύλη ένα SNMP μήνυμα απόκρισης. Με βάση αυτό το μήνυμα η πύλη απαντά στο διαχειριστή χρησιμοποιώντας πάλι το πρωτόκολλο SOAP για να κωδικοποιήσει τα δεδομένα της απάντησης. Οι συγγραφείς δεν χρησιμοποίησαν κάποια προτυποποιημένη MIB αλλά μια δοκιμαστική προκειμένου να μπορούν κάθε φορά να εισάγουν διαφορετικό αριθμό μεταβλητών σε αυτή. Έτσι μελέτησαν την ανάκτηση όλων των αντικειμένων μιας μονής στήλης ενός πίνακα με μεταβλητό αριθμό σειρών.



Σχήμα 3-7.Η τοπολογία των δοκιμών.

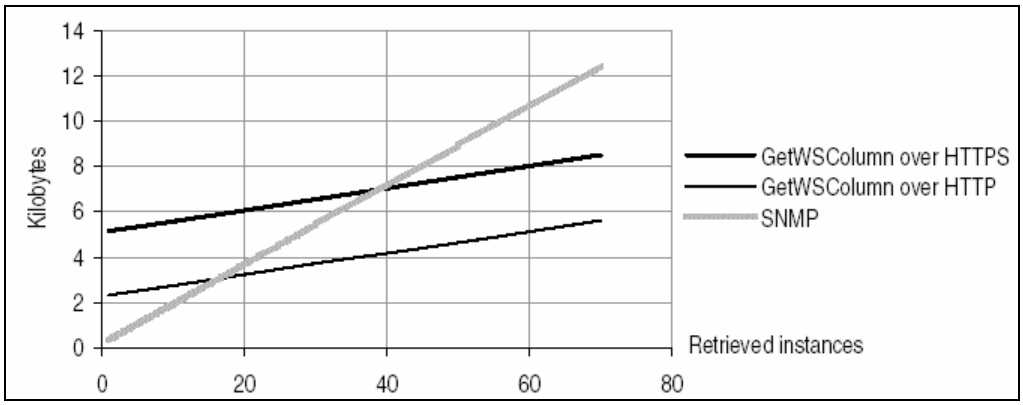


Σχήμα 3-8. Σύγκριση των όγκων της παραγόμενης δικτυακής κίνησης για μετάφραση σε επίπεδο πρωτοκόλλου.

Στην πρώτη δοκιμή εξετάζεται ο όγκος της κίνησης, που δημιουργείται κατά την ανάκτηση διαφορετικού αριθμού μεταβλητών κατάστασης, μέσω της απευθείας απεικόνισης των GetNext μηνυμάτων. Όπως φαίνεται από το σχήμα 3-8 ο όγκος της XML/HTTP κίνησης είναι τουλάχιστον μια τάξη μεγέθους μεγαλύτερος από τον αντίστοιχο του SNMP. Ένα τέτοιο αποτέλεσμα ήταν αναμενόμενο, αφού όπως έδειξε και η ανάλυση της παραγράφου 3-3, η χρήση του SOAP και του HTTP από τις XML/HTTP τεχνολογίες προκαλεί μεγάλο υπερκεφαλισμό σε σχέση με εκείνον του SNMP το οποίο χρησιμοποιεί το ASN.1/BER και το UDP για να κωδικοποιήσει και να μεταφέρει αντίστοιχα τη διαχειριστική πληροφορία. Μάλιστα, όσο αυξάνει ο αριθμός των διαχειριζόμενων αντικειμένων (και άρα ο αριθμός των GetNext μηνυμάτων) τόσο μεγαλώνει η διαφορά των υπερκεφαλισμών και επομένως οι διαφορές στην παραγόμενη δικτυακή κίνηση. Επίσης οι συγγραφείς μελετούν και την

επίδραση του ασφαλούς (HTTPS) αντί του απλού HTTP. Η κρυπτογράφηση της πληροφορίας δημιουργεί ακόμα μεγαλύτερο υπερκεφαλισμό με αποτέλεσμα ο όγκος της κίνησης να γίνεται απαγορευτικός ακόμη και για μικρό αριθμό μεταβλητών (π.χ. για την ανάκτηση μόνο 20 μεταβλητών χρειάζονται παραπάνω από 100 Kbytes).

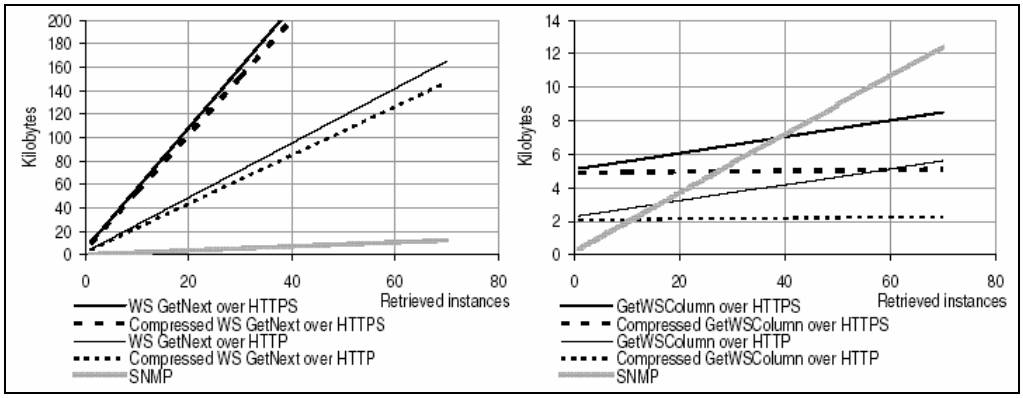
Οι ισορροπίες αλλάζουν δραματικά στην περίπτωση, που οι διαδοχικές Get λειτουργίες επιτελούνται από την πύλη και όχι από το διαχειριστή. Τώρα ο διαχειριστής στέλνει ένα απλό SOAP μήνυμα (GetWSColumn) στη πύλη η οποία με διαδοχικά GetNext συγκεντρώνει τη ζητούμενη πληροφορία από την SNMP συσκευή. Αφού ολοκληρωθεί η συλλογή, η πύλη στέλνει όλη την ανακτηθείσα πληροφορία πίσω στο διαχειριστή, μέσω ενός απλού SOAP μηνύματος. Τα αποτελέσματα απεικονίζονται στο σχήμα 3-9. Μέχρι τις πρώτες 18 μεταβλητές κατάστασης το SNMP φαίνεται πιο αποτελεσματικό από το XML/HTTP με τη χρήση του απλού HTTP, ενώ όταν χρησιμοποιείται ασφαλής σύνδεση το SNMP παραμένει πιο αποτελεσματικό για τις 40 πρώτες μεταβλητές. Ωστόσο, όσο ανεβαίνει ο αριθμός των διαχειριζόμενων αντικειμένων τόσο το SNMP εμφανίζει μεγαλύτερους υπερκεφαλισμούς από το XML/HTTP με αποτέλεσμα το SNMP να γίνεται ολοένα και πιο δαπανηρό από το XML/HTTP μετά τα προαναφερθέντα κατώφλια . Βέβαια το αποτέλεσμα αυτό δεν είναι απόλυτα ακριβές, αφού δεν μελετάται καθόλου η περίπτωση μαζικής ανάκτησης διαχειριστικών δεδομένων με ένα απλό SNMP μήνυμα (π.χ. χρησιμοποιώντας τη GetNext για σειρές ή τη GetBulk). Όπως θα αποδειχτεί και στη συνέχεια, όταν το SNMP χρησιμοποιείται σωστά παράγει πάντα χαμηλότερο όγκο κίνησης στο δίκτυο.



Σχήμα 3-9. Σύγκριση των όγκων της παραγόμενης δικτυακής κίνησης για μετάφραση σε επίπεδο αντικειμένου.

Οι συγγραφείς μελέτησαν επίσης και την επίδραση της συμπίεσης των SOAP φακέλων κατά την τέλεση των προαναφερθέντων πειραμάτων. Η συμπίεση εφαρμόζεται τόσο στα μηνύματα της αίτησης του διαχειριστή όσο και σε εκείνα της απάντησης της πύλης. Για τη συμπίεση εφαρμόστηκε ο Lempel-Ziv αλγόριθμος [21], ο οποίος είναι ιδιαίτερα αποτελεσματικός σε αρχεία με μεγάλη επαναληψιμότητα χαρακτήρων (όπως είναι SOAP/XML φάκελοι). Στην περίπτωση της αντιστοίχισης σε επίπεδο πρωτοκόλλου η συμπίεση δεν φαίνεται να επιφέρει σημαντικό κέρδος. Αυτό συμβαίνει επειδή τα SOAP μηνύματα (WS GetNext), που συμπιέζονται και στέλνονται κάθε φορά είναι πολύ μικρά (<1000 bytes) με αποτέλεσμα η επαναληψιμότητα να είναι περιορισμένη και η συμπίεση του συγκεκριμένου αλγορίθμου να

είναι αναποτελεσματική (μεγάλο C_{ratio}, βλέπε ενότητα 3.3.2.4). Αντίθετα, στην περίπτωση της αντιστοίχισης σε επίπεδο αντικειμένου η συμπίεση είναι πολύ πιο αποτελεσματική με αποτέλεσμα να μειώνει τη παραγόμενη κίνηση μέχρι και 2,5 φορές. Αυτό εξηγείται από το γεγονός ότι η πύλη επιστρέφει στο διαχειριστή ένα μήνυμα απάντησης, που συγκεντρώνει όλη τη διαχειριστική πληροφορία. Το μήνυμα αυτό είναι σχετικά μεγάλο (π.χ. 4Kbytes για 40 μεταβλητές κατάστασης) με αποτέλεσμα η συμπίεση να είναι αρκετά αποτελεσματική. Μάλιστα, όσο πιο μαζική είναι η ανάκτηση των δεδομένων τόσο μεγαλύτερο κέρδος επιτυγχάνεται στη χρήση του εύρους ζώνης. Τέλος, η εν λόγω εργασία δεν μελετά καθόλου τους χρόνους διαχείρισης, κάτι που θα είχε ιδιαίτερο νόημα ειδικά στη περίπτωση της συμπίεσης και της κρυπτογράφησης.



(α) (β)

Σχήμα 3-10. Η επίδραση της συμπίεσης για μετάφραση σε επίπεδο πρωτοκόλλου (α) και σε επίπεδο αντικειμένου (β).

3.4.1.2. Mi-Jung Choi and James Hong

Λίγους μήνες πριν την προηγούμενη εργασία είχε δημοσιευτεί μια παρόμοια μελέτη [8] η οποία υλοποιούσε επιπρόσθετα ένα ξεχωριστό XML/HTTP πράκτορα. Στόχος ήταν να αποτιμηθεί η επίδραση που μπορεί να έχει η μεσολάβηση της XML/SNMP πύλης στις διαχειριστικές επιδόσεις τις οποίες και συγκρίνει με την αμιγή SNMP και XML/HTTP διαχείριση. Οι συγγραφείς μελετούν κυρίως το πρόβλημα της παρακολούθησης του δικτύου (network monitoring με χρήση της Get/GetNext) αλλά εξετάζουν και ορισμένες περιπτώσεις ρύθμισης (SNMP Set) και ειδοποίησης (SNMP Trap).

Πιο συγκεκριμένα, μελετάται η διαχείριση μιας δικτυακής πύλης μέσω των SNMP αντικειμένων γενικής χρήσης (system group) [22] καθώς και των αντικειμένων διεπαφών (interfaces group) [23]. Στην εργασία δίνονται πειραματικά αποτελέσματα για τη διαχειριστική κίνηση που δημιουργείται κατά τις τρεις προσεγγίσεις (αμιγής SNMP διαχείριση, αμιγής XML διαχείριση και διαχείριση μέσω της XML/SNMP πύλης). Επίσης μετράται ο χρόνος απόκρισης του πράκτορα σε όλες τις παραπάνω περιπτώσεις καθώς και οι απαιτούμενοι πόροι επεξεργασίας και αποθήκευσης, των SNMP και XML/HTTP πρακτόρων.

Τα αποτελέσματα της παραγόμενης κίνησης για τις διάφορες διαχειριστικές λειτουργίες φαίνονται στους πίνακες 3.1, 3.2 και 3.3. Όπως αναμένονταν και από την θεωρητική ανάλυση της παραγράφου 3-3 προκύπτει ότι η τεχνολογία XML/HTTP επιφέρει μεγαλύτερη διαχειριστική κίνηση σε σχέση με το SNMP τουλάχιστον στην περίπτωση μικρού αριθμού μεταβλητών κατάστασης (μία ή δύο μεταβλητές). Στη συγκεκριμένη εργασία τα μηνύματα διαχείρισης κωδικοποιούνται απευθείας σε XML, χωρίς τη χρήση του SOAP πρωτοκόλλου [24] και της γλώσσας περιγραφής των υπηρεσιών ιστού WSDL (Web-Services Definition Language). Αυτό έχει ως συνέπεια πολύ μικρότερους υπερκεφαλισμούς και επομένως χαμηλότερες από τις αναμενόμενες διαφορές μεταξύ του SNMP και της XML/HTTP. Επιπρόσθετα, όσο αυξάνει ο αριθμός των διαχειριζόμενων αντικειμένων η διαφορά αυτή ψαλιδίζεται και αντιστρέφεται πολύ γρήγορα, αφού για την ανάκτηση μόλις 14 μεταβλητών (system group) το SNMP φαίνεται να παράγει διπλάσια κίνηση από το XML/HTTP. Μάλιστα όταν ο αριθμός των ληφθέντων μεταβλητών αγγίζει τις 70 (interfaces group) το SNMP παράγει τετραπλάσια κίνηση από εκείνη του XML/HTTP. Πέραν του μειωμένου υπερκεφαλισμού του XML/HTTP λόγω της μη χρήσης SOAP η διαφορά αυτή οφείλεται και στο ότι οι συγγραφείς χρησιμοποιούν πολύ αναποτελεσματικά το SNMP, αφού κάθε για κάθε μεταβλητή στέλνεται ένα νέο GetNext μήνυμα στον πράκτορα. Δεν μελετάται καθόλου η περίπτωση μαζικής ανάκτησης διαχειριστικών δεδομένων με ένα απλό SNMP μήνυμα (π.χ. χρησιμοποιώντας τη GetNext για rows ή τη GetBulk). Όπως θα αποδειχτεί και από τις εργασίες, που αναφέρονται στη συνέχεια όταν το SNMP χρησιμοποιείται σωστά παράγει πάντα χαμηλότερο όγκο κίνησης στο δίκτυο.

Οι πίνακες 3-2 και 3-3 εμφανίζουν την XML/HTTP κίνηση για τη ρύθμιση (Set) και την ειδοποίηση (Trap) αυξημένη σε σχέση με εκείνη της ανάκτησης (Get). Αυτό αποδίδεται κυρίως στο διαφορετικό τρόπο μεταφοράς της διαχειριστικής πληροφορίας. Για την ανάκτηση της πληροφορίας ο διαχειριστής στέλνει το *HTTP GET* μήνυμα στον εξυπηρετητή ιστού του πράκτορα. Αντίθετα όταν ο διαχειριστής επιθυμεί να θέσει μια τιμή στέλνει το *HTTP POST* μήνυμα στον πράκτορα. Επίσης όταν ο πράκτορας χρειαστεί να ενημερώσει το διαχειριστή για κάποιο συμβάν στέλνει ένα *HTTP POST* μήνυμα στον εξυπηρετητή ιστού του διαχειριστή. Γενικά το *HTTP POST* είναι μεγαλύτερο μήνυμα από το *HTTP GET* και για αυτό το λόγο δημιουργείται η παραπάνω διαφορά η οποία ωστόσο θα ήταν αμελητέα στην περίπτωση συμμετοχής πολλών διαχειριστικών αντικειμένων στα εν λόγω μηνύματα. Μάλιστα όσον αφορά στις απαντήσεις, αυτές υποδηλώνονται στη κεφαλή του HTTP πρωτοκόλλου (*HTTP OK*), αφού δεν χρησιμοποιείται ένας μηχανισμός σαν το SOAP για την περιγραφή των απομακρυσμένων διεργασιών διαχείρισης (Remote Procedure Calls - RPC). Για αυτό το λόγο οι απαντήσεις της XML/HTTP έχουν ακριβώς το ίδιο μέγεθος (210 bytes).

Οι συγγραφείς μελέτησαν επίσης και το χρόνο που μεσολαβεί από τη στιγμή, που καλείται η συνάρτηση παρασκευής του μηνύματος της αίτησης στον διαχειριστή μέχρι τη στιγμή, που λαμβάνεται το μήνυμα της απόκρισης από τον πράκτορα. Με βάση τη μοντελοποίηση της ενότητας 3.3.2 ο χρόνος αυτός είναι ο συνολικός χρόνος διαχείρισης μείον το χρόνο αποκωδικοποίησης ($t_{ΑΠΟΚ(Δ)}$) του μηνύματος της απόκρισης στο διαχειριστή. Αν ληφθεί υπόψη η θεώρηση της ενότητας 3.3.2.1 σχετικά με την αμελητέα συνεισφορά του διαχειριστή

($t_{ΚΩΔ(Δ)}$) + $t_{ΑΠΟΚ(Δ)}$) στο συνολικό χρόνο διαχείρισης τότε ο μετρούμενος χρόνος ($t_{ΚΩΔ(Δ)}$+ $t_{ΜΜΑ}$ +$t_{ΑΠΟΚ(Π)}$ + t_{EN} + $t_{ΚΩΔ(Π)}$ + $t_{ΜΜΑΠ}$) είναι πολύ κοντά στο συνολικό χρόνο διαχείρισης ($t_{ΚΩΔ(Δ)}$ + $t_{ΜΜΑ}$ +$t_{ΑΠΟΚ(Π)}$ + t_{EN} + $t_{ΚΩΔ(Π)}$ + $t_{ΜΜΑΠ}$ + $t_{ΑΠΟΚ(Δ)}$). Ωστόσο οι συγγραφείς δεν αναφέρουν το μέσο επικοινωνίας του διαχειριστή με τον πράκτορα αλλά ούτε και την ταχύτητα μετάδοσης των διαχειριστικών δεδομένων. Επίσης δεν αναφέρεται η πλατφόρμα υλικού, που φιλοξένησε τον διαχειριστή. Ωστόσο γίνεται γνωστή η πλατφόρμα του πράκτορα η οποία είναι ένα ενσωματωμένο σύστημα (Motorola MPC850DE) [25] με χαμηλές επεξεργαστικές και αποθηκευτικές δυνατότητες.

Πίνακας 3.1. Ο όγκος της παραγόμενης κίνησης (Get) στο δίκτυο.

Management property	Get request message (bytes)			Get response message (bytes)		
	SNMP	XML-based	XML/SNMP gateway	SNMP	XML-based	XML/SNMP gateway
sysDescr	82	238	396 (314 + 82)	145	240	401 (145 + 256)
sysContact	82	240	388 (316 +82)	103	190	309 (103 + 206)
system Group	572	241	889 (317 + 572)	722	440	1178 (722 + 456)
inOctets (2 interfaces)	169	240	485 (316 + 169)	175	252	443 (175 + 268)
outOctets (2 interfaces)	169	241	486 (317 + 169)	176	256	448 (176 + 272)
interfaces Group	3720	241	4037 (317 + 3720)	3818	1654	5488 (3818 + 1670)

Πίνακας 3.2. Ο όγκος της παραγόμενης κίνησης (Set) στο δίκτυο.

Management property	Set request message (bytes)			Set response message (bytes)		
	SNMP	XML-based	XML/SNMP gateway	SNMP	XML-based	XML/SNMP gateway
sysDescr	104	350	550 (110 +450)	104	210	330 (226 + 104)
sysContact	130	358	586 (456 +130)	130	210	356 (226 + 130)

Πίνακας 3.3. Ο όγκος της παραγόμενης κίνησης (Trap) στο δίκτυο.

Management property	Trap notification message (bytes)			Trap response message (bytes)		
	SNMP	XML-based	XML/SNMP gateway	SNMP	XML-based	XML/SNMP gateway
coldStart trap	110	450	560 (110 +450)	0	210	210
linkDown trap	127	520	647 (127 +520)	0	210	210

Από τον πίνακα 3.4 φαίνεται ότι για μια μεταβλητή (*sysDescr*) το SNMP συμπεριφέρεται καλύτερα από το XML/HTTP και αυτό είναι πολύ πιθανό να οφείλεται στην αρκετά πιο

σύνθετη κωδικοποίηση του XML έναντι του ASN.1/BER. Ωστόσο όταν ο αριθμός των μεταβλητών ανεβαίνει, αυξάνουν σημαντικά και οι χρόνοι λήψης των τιμών των μεταβλητών κατάστασης. Εξάλλου, όπως αποδεικνύεται και από μεταγενέστερες εργασίες, που αναφέρονται παρακάτω είναι πολύ πιθανό η αύξηση αυτή να οφείλεται σε διαφορετικούς παράγοντες για την κάθε τεχνολογία. Για την XML φαίνεται ότι η αύξηση δημιουργείται από το μεγάλο αριθμό των αντικειμένων που πρέπει να κωδικοποιηθούν σε XML (π.χ. 70 αντικείμενα για το interfaces Group). Αντίθετα για το SNMP η υψηλή τιμή του χρόνου εξηγείται κυρίως από το γεγονός ότι για την ανάκτηση των τιμών της MIB απαιτείται πέραν των άλλων και μια κλήση συστήματος (system call) η οποία συμβαίνει κάθε φορά, που ζητείται η τιμή ενός απλού αντικειμένου. Αν μάλιστα συνυπολογιστεί το γεγονός ότι στη συγκεκριμένη εργασία δεν γίνεται μαζική ανάκτηση των τιμών της MIB, τότε η κλήση συστήματος επιτελείται περίπου 70 φορές. Αυτό επιβαρύνει σημαντικά το χρόνο ενορχήστρωσης και επομένως το συνολικό χρόνο διαχείρισης.

Πίνακας 3.4. Χρόνος απόκρισης του πράκτορα στα μηνύματα Get.

Management property	Response time (ms)		
	SNMP	XML-based	XML/SNMP gateway
sysDescr	40	50	80
System Group	160	140	250
interfaces Group	980	800	1250

Τέλος, οι συγγραφείς δίνουν τη χρήση του επεξεργαστή και το μέγεθος της μνήμης, που καταλαμβάνει το πρωτότυπο του XML/HTTP πράκτορα σε σχέση με εκείνο του SNMP. Η πληροφορία αυτή ανακτάται από το λειτουργικό σύστημα του πράκτορα (με χρήση Linux εντολών όπως η *top* και η *cpuload*) και συγκεντρώνεται στον πίνακα 3.5. Όπως φαίνεται, η χρήση του επεξεργαστή στον XML πράκτορα είναι μόλις 3% παραπάνω από εκείνον του SNMP. Επίσης η χρήση της μνήμης κατά τη διάρκεια λειτουργίας (run-time) είναι μόλις 16% περισσότερη για τον XML/HTTP πράκτορα, ενώ το μέγεθος του εκτελέσιμου XML/HTTP κώδικα είναι 37,5% (150Kbytes) μεγαλύτερο. Αν συνυπολογιστεί το γεγονός ότι η σύγκριση γίνεται με την πιο απλή υλοποίηση SNMP πράκτορα (version 1) τότε το XML/HTTP πρωτότυπο φαίνεται να αποτελεί μια πλήρως αποδεκτή υλοποίηση. Το βασικό συμπέρασμα που προκύπτει λοιπόν είναι ότι η βελτιστοποιημένη υλοποίηση του XML/HTTP πράκτορα οδηγεί σε αποδεκτή χρήση πόρων ακόμα και όταν πρόκειται για ένα ενσωματωμένο σύστημα με πολύ μικρές επεξεργαστικές και αποθηκευτικές δυνατότητες σαν αυτό που χρησιμοποιήθηκε στην εν λόγω εργασία.

Πίνακας 3.5. Η χρήση πόρων στον πράκτορα.

Agent	CPU load	Run-time memory usage	Executable code size
SNMPv1	17%	600KB	400KB
XML-based	20%	700KB	550KB

3.4.2. Εργασίες χωρίς τη χρήση πυλών

3.4.2.1. George Pavlou et al

Το καλοκαίρι του 2004 δημοσιεύτηκε η εργασία [6], η οποία διερευνά τις επιδόσεις της XML/HTTP τεχνολογίας σε σχέση με αυτές του SNMP πρωτοκόλλου αλλά και με εκείνες της CORBA [26]. Η εργασία χρησιμοποιεί ως παράδειγμα το πρόβλημα της παρακολούθησης του δικτύου και συγκρίνει τις επιδόσεις των προαναφερθέντων τεχνολογιών όσον αφορά στον όγκο της κίνησης διαχείρισης αλλά και στο χρόνο διαχείρισης.

Πιο συγκεκριμένα, διερευνάται η διαδικασία λήψης των τιμών της TCP διαχειριστικής βάσης (MIB), έτσι όπως αυτή ορίζεται στο [27]. Μάλιστα, στο TCP διαχειριζόμενο αντικείμενο δόθηκαν προκαθορισμένες και σταθερές τιμές προκειμένου να απομονωθούν και να συγκριθούν οι υπερκεφαλισμοί που επιφέρει η κάθε τεχνολογία. Αυτό πρακτικά σημαίνει ότι ο κάθε πράκτορας (SNMP, XML/HTTP, CORBA) έχει έτοιμες τις τιμές που του ζητούνται και δεν προβαίνει σε ανάκτηση τιμών από το πραγματικό σύστημα. Επομένως στα πειράματα δεν συμμετέχει η διαδικασία της ενορχήστρωσης αφού, όπως εξηγήθηκε και στην ενότητα 3.3.2.3 η διαδικασία αυτή δεν έχει να κάνει με την τεχνολογία διαχείρισης αλλά με την υλοποίηση και την πλατφόρμα, που φιλοξενεί τον πράκτορα.

Οι εξεταζόμενες διαχειριστικές λειτουργίες είναι οι ακόλουθες:

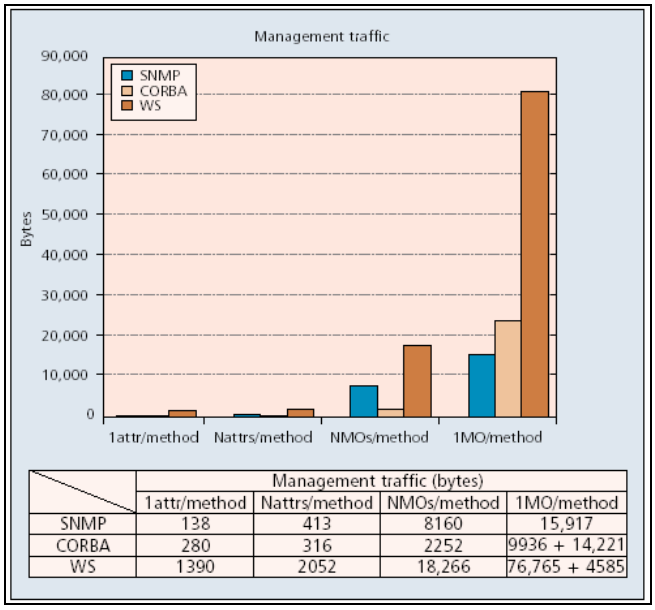
1. Η ανάκτηση μιας απλής μεταβλητής του TCP αντικειμένου με μια μέθοδο (*1attr/method*). Στην περίπτωση του SNMP η ανάκτηση αυτή γίνεται με τη χρήση της μεθόδου *Get*, ενώ μια νέα IDL (Interface Definition Language) και WSDL (Web-Services Definition Language) μέθοδος ορίζεται και υλοποιείται για την CORBA και την XML/HTTP, αντίστοιχα.
2. Η ανάκτηση 8 δυναμικών μεταβλητών του TCP αντικειμένου με μια μέθοδο (*Nattrs/method*). Οι μεταβλητές αυτές σχετίζονται με την κατάσταση της σύνδεσης (*tcpActiveOpens*, *tcpPassiveOpens*, *tcpAttemptFails*, *tcpEstabResets* και *tcpCurrEstab*) αλλά και με τους μετρητές των TCP πακέτων (*tcpInSegs*, *tcpOutSegs*, *tcpRetransSegs*). Στην περίπτωση του SNMP η ανάκτηση των παραπάνω μεταβλητών γίνεται με τη χρήση της μεθόδου *Get* ενώ μια νέα IDL και WSDL μέθοδος χρησιμοποιείται για την CORBA και XML/HTTP, αντίστοιχα.
3. Η ανάκτηση της κατάστασης 40 TCP συνδέσεων, σε καθεμιά εκ των οποίων αντιστοιχούν 5 μεταβλητές κατάστασης. Η διαδικασία αυτή γίνεται με δύο εναλλακτικούς τρόπους:
    a. Με τη χρήση μιας μεθόδου ανά TCP σύνδεση (*1 MO/method*). Η SNMP *GetNext* μέθοδος καλείται κάθε φορά που λαμβάνεται μια νέα γραμμή (με 5 μεταβλητές κατάστασης για κάθε TCP σύνδεση), ενώ χρησιμοποιούνται νέες IDL και WSDL μέθοδοι για φέρουν τα ίδια δεδομένα στην περίπτωση της CORBA και της XML/HTTP αντίστοιχα. Συνολικά η *GetNext* καλείται 41 φορές, με τις 40 πρώτες να δίνουν την κατάσταση των TCP συνδέσεων και

την τελευταία να δείχνει το τέλος του πίνακα των συνδέσεων. Επίσης 41 πράξεις πραγματοποιούνται και στις περιπτώσεις της CORBA και της XML/HTTP, αφού η πρώτη πράξη για κάθε πρωτόκολλο φέρνει τις διαλειτουργικές αναφορές (Interoperable References – IORs) και τα αναγνωριστικά ενιαίου πόρου (Uniform Resource Identifiers – URIs) των αντικειμένων της σύνδεσης.

b. Με τη χρήση μιας μεθόδου για όλες τις TCP συνδέσεις (*NMOs/method*). Εδώ πάλι δυο απλές IDL και WSDL μέθοδοι χρησιμοποιούνται για την CORBA και την XML/HTTP, ενώ για το SNMP χρησιμοποιείται η μέθοδος *GetBulk*. Μάλιστα ιδιαίτερη προσοχή δόθηκε στο μέγιστο αριθμό επαναλήψεων (max repetitions) της *GetBulk*, προκειμένου το μήνυμα της απάντησης να προσαρμόζεται σε ένα ακριβώς IP πακέτο και έτσι να αποφευχθεί η τεμαχιοποίηση (IP fragmentation). Με αυτό τον τρόπο η *GetBulk* κλήθηκε 4 διαδοχικές φορές για να φέρει τα δεδομένα και των 40 συνδέσεων.

Οι συγγραφείς χρησιμοποίησαν την Orbacus [28], τηWASP [29] και τη Net-SNMP [30] πλατφόρμα για τις CORBA, XML/HTTP και SNMP υλοποιήσεις, αντίστοιχα. Μάλιστα, απέφυγαν τη χρήση του gSOAP [31,32], που είναι μια βελτιστοποιημένη υλοποίηση για το SOAP πρωτόκολλο προκειμένου να συγκρίνουν τις γενικού σκοπού πλατφόρμες (Orbacus - WASP) μεταξύ τους. Επιπλέον οι συγγραφείς πραγματοποίησαν δύο υλοποιήσεις (σε C++ και σε Java) για κάθε περίπτωση με στόχο την αποτίμηση της επίδρασης, που έχει η προγραμματιστική γλώσσα στις επιδόσεις της κάθε τεχνολογίας. Η μόνη εξαίρεση είναι ο πράκτορας SNMP ο οποίος υλοποιήθηκε μόνο σε C. Ωστόσο ο SNMP διαχειριστής υλοποιήθηκε τόσο σε C (με τη χρήση των Net-SNMP APIs) όσο και σε Java (με την AdventNet SNMP πλατφόρμα). Πράκτορες και διαχειριστές φιλοξενούνται σε δύο ξεχωριστά αλλά πανομοιότυπα συστήματα (Pentium III Celeron στα 1GHz με 256 MB RAM) τα οποία επικοινωνούν μέσω μιας 100Mbps Ethernet σύνδεσης.

Το σχήμα 3-11 παρουσιάζει τα αποτελέσματα των μετρήσεων της κίνησης διαχείρισης, που δημιουργούν οι εξεταζόμενες τεχνολογίες για κάθε μια από τις προαναφερθείσες διαχειριστικές λειτουργίες. Όπως αναμένονταν και από τη θεωρητική ανάλυση της παραγράφου 3-3 προκύπτει ότι η τεχνολογία XML/HTTP επιφέρει μέχρι και μια τάξη μεγέθους μεγαλύτερη διαχειριστική κίνηση σε σχέση με το SNMP. Αυτό συμβαίνει στην περίπτωση ανάκτησης μόνο μιας μεταβλητής (*1attr/method*), ενώ όσο ο αριθμός της ζητούμενης διαχειριστικής πληροφορίας αυξάνει τόσο η διαφορά αυτή ψαλιδίζεται. Έτσι στην περίπτωση ανάκτησης 8 μεταβλητών ο όγκος της XML/HTTP κίνησης είναι 5 φορές μεγαλύτερος από εκείνη του SNMP, ενώ όταν ανακτώνται 200 (40*5) μεταβλητές με μία μέθοδο ο όγκος είναι μεγαλύτερος μόλις κατά 2,2 φορές. Κατά την ανάκτηση μιας μόνο μεταβλητής το SNMP υπερτερεί (χαμηλότερος όγκος παραγόμενης κίνησης) ακόμα και έναντι της CORBA η οποία γενικώς φαίνεται να είναι η πιο οικονομική τεχνολογία όσον αφορά στη χρήση του εύρους ζώνης.



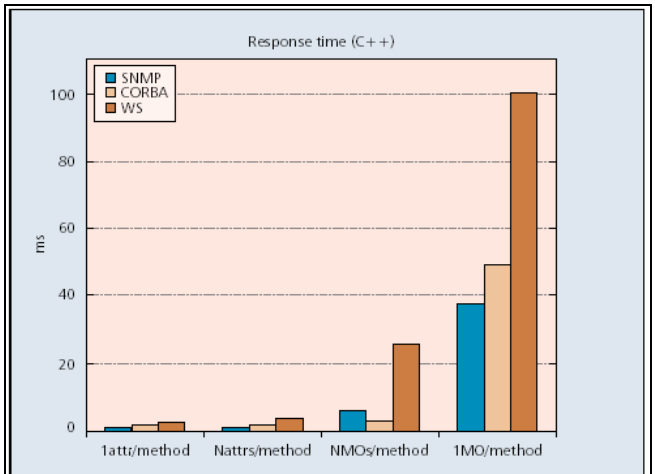
	Management traffic (bytes)			
	1attr/method	Nattrs/method	NMOs/method	1MO/method
SNMP	138	413	8160	15,917
CORBA	280	316	2252	9936 + 14,221
WS	1390	2052	18,266	76,765 + 4585

Σχήμα 3-11. Σύγκριση των όγκων της παραγόμενης δικτυακής κίνησης.

Το σχήμα 3-12 απεικονίζει το συνολικό χρόνο διαχείρισης ($t_{ΚΩΔ(Δ)}$ + $t_{ΜΜΑ}$ +$t_{ΑΠΟΚ(Π)}$ + $t_{ΚΩΔ(Π)}$ + $t_{ΜΜΑΠ}$+ $t_{ΑΠΟΚ(Δ)}$, βλέπε ενότητα 3.3.2), χωρίς όπως προαναφέρθηκε, να περιλαμβάνει το χρόνο ενορχήστρωσης (t_{EN}). Καθώς το διαθέσιμο εύρος ζώνης είναι πολύ μεγάλο (100Mbps) οι χρόνοι μετάδοσης ($t_{ΜΜΑ}$ + $t_{ΜΜΑΠ}$) είναι σχεδόν αμελητέοι (μικρότεροι του 5% του συνολικού χρόνου) ακόμα και στις λειτουργίες εκείνες (*1 MO/method* και *NMOs/method*), που ο όγκος των διαχειριστικών δεδομένων είναι σχετικά υψηλός (80Kbytes). Επομένως, οι όποιες άλλες διαφορές στο χρόνο διαχείρισης οφείλονται στη διαδικασία κωδικοποίησης και αποκωδικοποίησης στον πράκτορα και το διαχειριστή.

Όπως αναμένονταν, σε όλες τις περιπτώσεις οι C++ υλοποιήσεις είναι πολύ πιο αποτελεσματικές από τις αντίστοιχες της Java. Μάλιστα σε ορισμένες περιπτώσεις η διαφορά αγγίζει το 150%. Το δεύτερο συμπέρασμα που εξάγεται είναι ότι οι χρόνοι κωδικοποίησης και αποκωδικοποίησης των SOAP/XML μηνυμάτων αυξάνονται γραμμικά με τον αριθμό των διαχειριζόμενων αντικειμένων και μπορούν να είναι μέχρι και 4 φορές μεγαλύτεροι από εκείνους του SNMP ή ακόμα και 10 φορές μεγαλύτεροι από εκείνους της CORBA (*NMOs/method*).

Όπως αποδεικνύεται και από τις επόμενες εργασίες, οι χαμηλές αυτές επιδόσεις του SOAP πρωτοκόλλου φαίνεται να οφείλονται κυρίως στη χρήση γραμματικών αναλυτών (XML parsers) γενικής χρήσεως (DOM [33]) από τη WASP πλατφόρμα. Ένα ακόμα αβίαστο συμπέρασμα των συγγραφέων, που έδωσε και το έναυσμα για περαιτέρω διερεύνηση σε αυτή τη διατριβή είναι το ότι σε περίπτωση που εφαρμόζονταν συμπίεση στα μηνύματα διαχείρισης ο συνολικός χρόνος διαχείρισης θα αύξανε. Μετά τη μοντελοποίηση που έγινε στην ενότητα 3.3.2 και τις σχέσεις της 3.3.2.4, αποδεικνύεται ότι αυτό δεν είναι απόλυτα ακριβές και εξαρτάται από τον αλγόριθμο συμπίεσης αλλά και από τις συνθήκες διαχείρισης (διαθέσιμο εύρος ζώνης, επεξεργαστικές και αποθηκευτικές δυνατότητες του πράκτορα, διαχειριστικές ενέργειες).



		Response time (ms)			
		1attr/method	Nattrs/method	NMOs/method	1MO/method
SNMP	C++	0.8	1	6	37
	Java	1	2	8	45
CORBA	C++	1.5	1.7	2.5	45 + 4
	Java	2	4	6	90 + 7
WS	C++	2.5	3.7	25	94 + 6
	Java	5	9	40	240 + 10

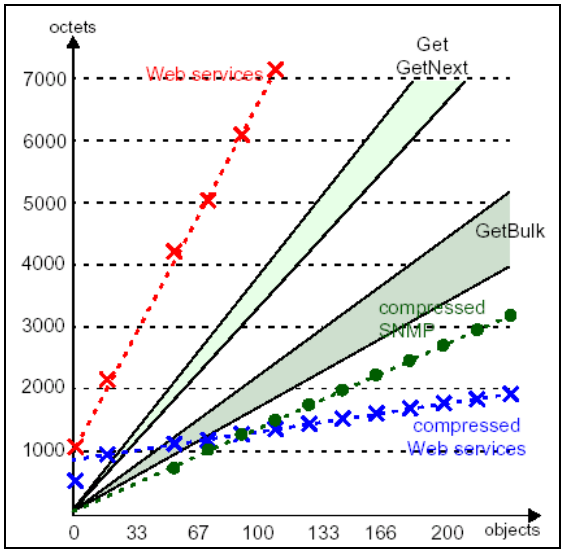
Σχήμα 3-12. Σύγκριση των χρόνων διαχείρισης.

3.4.2.2. Aiko Pras et al

Το φθινόπωρο του 2004 δημοσιεύτηκε η εργασία [5], η οποία αποτελεί ίσως την πιο ολοκληρωμένη θεώρηση για τις επιδόσεις της XML/HTTP τεχνολογίας συγκριτικά με εκείνες του SNMP. Η εργασία χρησιμοποιεί ως παράδειγμα το πρόβλημα της παρακολούθησης του δικτύου και συγκρίνει τις επιδόσεις του SNMP με εκείνες των XML/HTTP τεχνολογιών όσον αφορά στον όγκο της κίνησης διαχείρισης, στο χρόνο διαχείρισης αλλά και στις απαιτήσεις επεξεργασίας και αποθήκευσης στον πράκτορα.

Πιο συγκεκριμένα, η εργασία μελετά το πρόβλημα της ανάκτησης των δεδομένων μια δικτυακής διεπαφής (*ifTable*) [34] τόσο με τη χρήση SNMP όσο και με τη χρήση XML/HTTP πρακτόρων. Για τους SNMP πράκτορες χρησιμοποιήθηκαν έτοιμα εμπορικά συστήματα (π.χ. της 3Com , Cisco, HP, IBM κ.α) αλλά και η ανοιχτού κώδικα υλοποίηση Net-SNMP [35]. Όσον αφορά στους XML/HTTP πράκτορες εκεί έγινε χρήση των προτύπων του SOAP [36] και του WSDL [37] προκειμένου να υλοποιηθούν πρωτότυπα που ανακτούν απλά στοιχεία του πίνακα διεπαφών (ifTable elements), γραμμές (ifTable rows), στήλες (ifTable columns) ή ακόμα και ολόκληρο των πίνακα (ifTable) [38]. Αυτό συνέβη προκειμένου η αξιολόγηση των επιδόσεων να γίνει συναρτήσει του αριθμού των ανακτώμενων κάθε φορά διαχειριζόμενων αντικειμένων (managed objects).

Οι συγγραφείς πειραματίστηκαν με 23 διαφορετικούς SNMP πράκτορες και με έναν XML/HTTP πράκτορα. Αρχικά μελέτησαν τον όγκο της κίνησης διαχείρισης σε επίπεδο εφαρμογής (SNMP-SOAP/XML). Μάλιστα χρησιμοποίησαν και τις τρεις μεθόδους (*Get*, *GetNext* και *GetBulk*) προκειμένου να καλύψουν και τις βελτιστοποιήσεις του SNMP. Τα αποτελέσματα φαίνονται στο σχήμα 3-13.

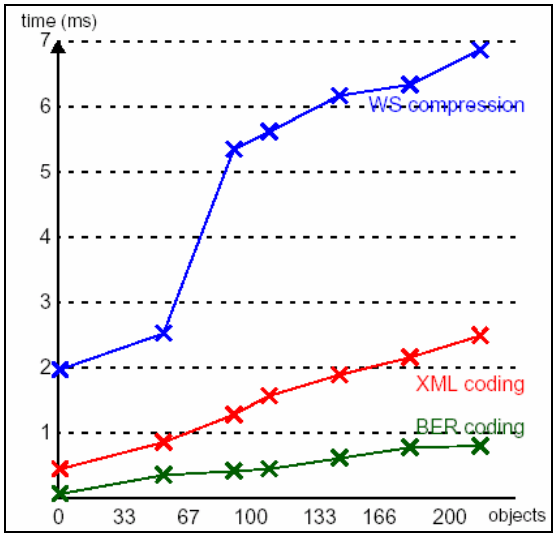


Σχήμα 3-13. Σύγκριση των όγκων πληροφορίας διαχείρισης.

Όπως αναμένονταν και από την θεωρητική ανάλυση της παραγράφου 3-3 προκύπτει ότι το SNMP είναι πολύ καλύτερο (μέχρι και 7 φορές μικρότερος όγκος δεδομένων) από το XML/HTTP στην περίπτωση ανάκτησης μικρού αριθμού αντικειμένων. Αλλά ακόμα και όταν ο αριθμός των διαχειριζόμενων αντικειμένων είναι μεγάλος το SNMP παραμένει κατά ένα παράγοντα 2 (Get) μέχρι και ένα παράγοντα 4 (GetBulk) καλύτερο από το XML/HTTP (Web-services). Στη συγκεκριμένη εργασία μελετήθηκε επίσης η επίδραση της συμπίεσης στα διαχειριστικά δεδομένα. Για το XML/HTTP χρησιμοποιήθηκε η υλοποίηση zlib [39] του Lempel-Ziv αλγορίθμου ενώ για το SNMP υιοθετήθηκε η λύση OPC (ObjectID Prefix Compression) της ομάδας EOS (IETF Evolution Of SNMP group) [40]. Τα αποτελέσματα απεικονίζονται στο ίδιο σχήμα (3-13) και φανερώνουν ότι η συμπίεση των SOAP φακέλων μπορεί να μειώσει τον όγκο τους κατά 2 (1 μεταβλητή κατάστασης) μέχρι 8 φορές (250 μεταβλητές κατάστασης). Αντίθετα, στο SNMP η μείωση του όγκου φτάνει μόλις το 75%. Μάλιστα η συμπίεση είναι γραμμική, εξαιτίας του χρησιμοποιούμενου αλγορίθμου και της φύσης των SNMP δεδομένων και επομένως ο λόγος συμπίεσης δεν βελτιώνεται με τον όγκο της διαχειριστικής πληροφορίας. Η μεγάλη διαφορά των λόγων συμπίεσης στις δύο τεχνολογίες έχει ως συνέπεια την επικράτηση (χαμηλότερος όγκος δεδομένων) της XML/HTTP, στις περιπτώσεις όπου τα διαχειριζόμενα αντικείμενα ξεπερνούν τα 70. Ωστόσο, επειδή αυτό που ενδιαφέρει είναι κυρίως η χρήση του εύρους ζώνης από τις διαχειριστικές τεχνολογίες, στην εργασία γίνεται και σύγκριση των υπερκεφαλισμών του SNMP με αυτούς του XML/HTTP. Όπως αναμένονταν τα SNMP μηνύματα απαιτούν μόνο 56 οκτάδες για να μεταδοθούν στο IP δίκτυο (με τη μορφή UDP πακέτων) ενώ τα SOAP μηνύματα απαιτούν περίπου 380 οκτάδες για την τριμερή χειραψία του TCP και άλλες 300 περίπου οκτάδες για τη μετάδοσή τους υπό μορφή HTTP/TCP πακέτων. Το γεγονός αυτό έχει ως αποτέλεσμα την ακόμα μεγαλύτερη διαφορά (μέχρι και μια τάξη μεγέθους) στην παραγόμενη δικτυακή κίνηση του SNMP και XML/HTTP.

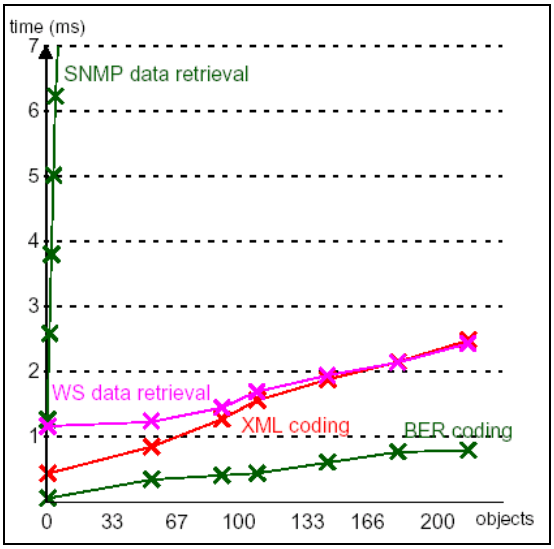
Η εν λόγω εργασία παρουσιάζει επίσης πειραματικά αποτελέσματα, που αφορούν στους χρόνους επεξεργασίας ($t_{ΑΠΟΚ(Π)}$ + $t_{ΚΩΔ(Π)}$, βλέπε ενότητα 3.3.2.1) για την αποκωδικοποίηση

των διαχειριστικών αιτήσεων και κωδικοποίηση των αντίστοιχων απαντήσεων στους SNMP και XML/HTTP πράκτορες. Επιπλέον μετρώνται οι χρόνοι συμπίεσης και αποσυμπίεσης ($t_{ΑΠΟΣ(Π)}$ + $t_{ΣΥΜ(Π)}$, βλέπε ενότητα 3.3.2.4) καθώς και οι χρόνοι ενορχήστρωσης (t_{EN}, βλέπε ενότητα 3.3.2.3), ενώ στο τέλος δίνονται αποτελέσματα για την καθυστέρηση κυκλικού ταξιδιού (round trip delay) χωρίς τη χρήση συμπίεσης. Αν ληφθεί υπόψη η θεώρηση της ενότητας 3.3.2.1 σχετικά με την αμελητέα συνεισφορά του διαχειριστή ($t_{ΚΩΔ(Δ)}$) + $t_{ΑΠΟΚ(Δ)}$) στο συνολικό χρόνο διαχείρισης, τότε θα μπορούσε να εξαχθεί το συμπέρασμα, ότι η καθυστέρηση κυκλικού ταξιδιού ($t_{ΜΜΑ}$ +$t_{ΑΠΟΚ(Π)}$ + $t_{ΚΩΔ(Π)}$ + $t_{ΜΜΑΠ}$) προσεγγίζει το συνολικό χρόνο διαχείρισης. Πρέπει να σημειωθεί ότι οι τιμές που παρουσιάζονται είναι μόνο για τις συγκεκριμένες υλοποιήσεις και για την δεδομένη πλατφόρμα (Pentium III στα 800Mhz, Debian GNU/Linux kernel v2.4.22) στην οποία έγιναν οι μετρήσεις. Ωστόσο ένας από τους στόχους της εργασίας ήταν να δοθεί μια απάντηση σχετικά με την καταλληλότητα των XML τεχνολογιών στη διαχείριση με κριτήριο το χρόνο κωδικοποίησης.



Σχήμα 3-14. Σύγκριση των χρόνων επεξεργασίας.

Το σχήμα 3-14 συγκρίνει το συνολικούς χρόνους κωδικοποίησης και αποκωδικοποίησης των SNMP (BER coding) με εκείνους των XML/HTTP πρακτόρων (XML coding) για διαφορετικούς αριθμούς απλών ανακτώμενων αντικειμένων (objects). Για ένα μόνο αντικείμενο ο συνολικός χρόνος επεξεργασίας στον XML πράκτορα είναι μέχρι και 7 φορές υψηλότερος (0,44ms) από τον αντίστοιχο σε ένα SNMP πράκτορα (0,06ms), ενώ όσο μεγαλώνει ο όγκος των ανακτώμενων αντικειμένων αυτή η διαφορά ψαλιδίζεται. Έτσι για 216 αντικείμενα ο χρόνος στον XML πράκτορα (2,5ms) είναι μόλις 3 φορές υψηλότερος από τον αντίστοιχο σε ένα SNMP πράκτορα (0,8ms). Βεβαίως, ποσοστιαία οι διαφορές είναι μεγάλες, αλλά το ζητούμενο ήταν να αποδειχθεί το κατά πόσο οι διαφορές αυτές επιδρούν στο συνολικό χρόνο διαχείρισης. Για αυτό το λόγο έγιναν μετρήσεις και στους χρόνους ενορχήστρωσης στο πρωτότυπο ανοιχτού κώδικα Net-SNMP, όπου ήταν δυνατή η τοποθέτηση συναρτήσεων μέτρησης χρόνου. Το σχήμα 3-15 απεικονίζει τα συγκριτικά αποτελέσματα.



Σχήμα 3-15. Οι χρόνοι ενορχήστρωσης συγκριτικά με τους χρόνους κωδικοποίησης.

Καταρχάς φαίνεται ότι ο χρόνος ανάκτησης (1,2ms) ενός απλού αντικειμένου του *ifTable* πίνακα είναι αρκετά μεγαλύτερος από τους αντίστοιχους χρόνους κωδικοποίησης. Η σχετικά υψηλή τιμή του χρόνου αυτού εξηγείται κυρίως από το γεγονός ότι για την ανάκτηση των τιμών του παραπάνω πίνακα απαιτείται πέραν των άλλων και μια κλήση συστήματος (system call). Το ίδιο συμβαίνει και με άλλους πίνακες. Μάλιστα ο χρόνος αυτός είναι σχεδόν ίδιος και στις δύο περιπτώσεις (WS data retrieval και SNMP data retrieval), αφού οι συγγραφείς χρησιμοποίησαν τον ίδιο κώδικα ενορχήστρωσης (του Net-SNMP) τόσο για τον SNMP όσο και για τον XML/HTTP πράκτορα. Δεν συμβαίνει όμως το ίδιο στη περίπτωση των περισσοτέρων αντικειμένων. Στο πρωτότυπο του SNMP ο χρόνος ανάκτησης των δεδομένων αυξάνει πολύ γρήγορα με τον αριθμό τους. Έτσι, τα 5 αντικείμενα απαιτούν 6ms, τα 54 απαιτούν 65 ms, ενώ για την ανάκτηση των τιμών των 270 αντικειμένων χρειάζονται 500ms. Το φαινόμενο αυτό εξηγείται από το γεγονός ότι στον SNMP πράκτορα γίνεται μια νέα κλήση συστήματος για κάθε *ifTable* αντικείμενο που ζητείται από το μήνυμα-αίτηση. Κάτι τέτοιο αποφεύγεται στην υλοποίηση του XML/HTTP πράκτορα, όπου η ίδια κλήση συστήματος επιτελείται μια μόνο φορά η οποία είναι και αρκετή για να δώσει όλα τα ζητούμενα αντικείμενα του πίνακα. Για αυτόν το λόγο στο σχήμα 3-15 φαίνεται ότι ακόμα και για μεγάλο αριθμό αντικειμένων (π.χ. 100) ο χρόνος ενορχήστρωσης παραμένει σε χαμηλά επίπεδα (1,5ms) και πολύ κοντά στο χρόνο ανάκτησης του ενός αντικειμένου (1,2ms).

Επίσης στο σχήμα 3-14 φαίνεται η περαιτέρω καθυστέρηση, που επιφέρει η εφαρμογή της συμπίεσης των SOAP φακέλων στο χρόνο διαχείρισης. Σύμφωνα με τις μετρήσεις η διαδικασία της συμπίεσης για τη συγκεκριμένη υλοποίηση (zlib) φαίνεται ότι είναι 3 με 5 φορές πιο αργή από την διαδικασία της κωδικοποίησης. Αυτό διαισθητικά οδηγεί τους συγγραφείς στο συμπέρασμα ότι μπορεί να μην είναι σκόπιμη η εφαρμογή της συμπίεσης στις περιπτώσεις που το εύρος ζώνης είναι φτηνό, ενώ ο χρόνος της CPU είναι ακριβός. Επίσης οι μετρήσεις της καθυστέρησης κυκλικού ταξιδιού (πίνακας 3.6), που έγιναν σε εμπορικά συστήματα αλλά και στα πρωτότυπα αποδεικνύουν ότι η επίδραση της

κωδικοποίησης (BER ή SOAP/XML) είναι αμελητέα σε σχέση με τη διαδικασία της ενορχήστρωσης, ιδιαίτερα όταν η τελευταία δεν γίνεται με βέλτιστο τρόπο.

Πίνακας 3.6. Η καθυστέρηση κυκλικού ταξιδιού συναρτήσει του αριθμού των αντικειμένων

Objects	**1**	**22**	**66**	**270**
WS	1.7	2.6	10.3	36.5
WS-Comp	3.3	4.3	5.6	11.8
SNMP-1	0.4	1.6	3.9	21.1
SNMP-2	0.4	1.9	5.0	
SNMP-3	0.5	1.6	4.2	
SNMP-4	0.5	1.7	4.4	
SNMP-5	0.5	1.8	4.8	
SNMP-6	0.7	2.2	5.7	
SNMP-7	0.8	1.8	2.9	
SNMP-8	0.9	1.6	3.9	
SNMP-9	0.9	6.6	18.5	
SNMP-10	1.1	1.8	3.4	58.5
SNMP-11	1.2	2.9	6.7	
SNMP-12	1.3	2.7	5.4	
SNMP-13	1.5	14.0	40.1	
SNMP-14	1.6	5.0	15.1	
SNMP-15	1.7	4.2	9.6	
SNMP-16	2.7	44.5	127.6	178.7
SNMP-17	2.7	47	140.4	251.7
SNMP-18	3.5	17.2	77.9	
SNMP-19	3.7	24.3	100.8	
SNMP-20	4.1	76.7	243.0	
SNMP-21	11.1	83.7	727.6	
SNMP-22	11.3	238.7	66	
SNMP-23	87.7	822.2	10.3	

Τέλος, οι συγγραφείς δίνουν αποτελέσματα και για τη χρήση της μνήμης του πράκτορα σε κάθε περίπτωση. Τα αποτελέσματα έχουν να κάνουν με το μέγεθος του εκτελέσιμου κώδικα (instructions), τη στατική μνήμη (static data), που διατίθεται στην αρχή του προγράμματος και παραμένει σταθερή για όλη τη διάρκεια του καθώς και με τη δυναμική μνήμη (dynamic data) που διατίθεται κάθε φορά, που λαμβάνεται ένα μήνυμα αίτησης και απελευθερώνεται αμέσως μόλις σταλθεί το μήνυμα της απάντησης.

Πίνακας 3.7. Η χρήση της μνήμης στον πράκτορα.

		Data	
	Instructions	Static	Dynamic
SNMP	1972 KB	129 KB	70-160 KB
Web services	580 KB	470 B	4 KB

Ο πίνακας 3.7 δείχνει ότι το πρωτότυπο του XML/HTTP πράκτορα απαιτεί λιγότερη στατική και δυναμική μνήμη αλλά καταλαμβάνει και πολύ μικρότερο χώρο για τον εκτελέσιμο κώδικα του σε σχέση με εκείνο του Net-SNMP (v5.0.9). Βέβαια το γεγονός αυτό δεν πρέπει να υπερεκτιμηθεί αφού οι λειτουργίες, που επιτελεί το SNMP πρωτότυπο είναι πολύ περισσότερες (υποστήριξη 3 διαφορετικών εκδόσεων του SNMP, κρυπτογράφηση, πιστοποίηση κ.α.) από εκείνες του απλού XML/HTTP πράκτορα. Ωστόσο τα αποτελέσματα τεκμηριώνουν το γεγονός ότι όταν ο XML/HTTP πράκτορας υλοποιείται με βέλτιστο τρόπο, τότε οι απαιτούμενοι πόροι είναι συγκρίσιμοι με εκείνους του SNMP και τις περισσότερες φορές αποδεκτοί. Το ίδιο συμπέρασμα εξήχθη και από την εργασία [7] (ενότητα 3.4.1.1).

3.4.3. Κοινά συμπεράσματα και αδυναμίες των σχετικών εργασιών

Αφού δόθηκαν οι βασικότερες μελέτες των διαχειριστικών επιδόσεων της XML/HTTP τεχνολογίας, στο σημείο αυτό κρίνεται σκόπιμη η αναφορά των τελικών συμπερασμάτων, που προκύπτουν από τη συνδυαστική ανάλυση των αποτελεσμάτων των παραπάνω εργασιών. Τα κοινά συμπεράσματα λοιπόν συνοψίζονται στα εξής:

1. Όταν οι τεχνολογίες XML/HTTP χρησιμοποιούν το SOAP πρότυπο για την κωδικοποίηση των διαχειριστικών μεθόδων, ο όγκος της παραγόμενης δικτυακής κίνησης βρίσκεται σε υψηλά επίπεδα. Μάλιστα, όταν στη διαχειριστική πράξη εμπλέκεται μόνο μια μεταβλητή κατάστασης, ο όγκος της XML/HTTP κίνησης είναι μέχρι και μία τάξη μεγέθους μεγαλύτερος από τον αντίστοιχο του SNMP. Αλλά ακόμα και όταν εμπλέκεται μεγάλος αριθμός μεταβλητών, η XML/HTTP συνεχίζει να είναι πιο δαπανηρή με μικρότερες βέβαια ποσοστιαίες διαφορές (2-5 φορές μεγαλύτερος όγκος κίνησης).
2. Η εφαρμογή του αλγορίθμου Lempel-Ziv για τη συμπίεση των SOAP/XML φακέλων, που μεταφέρουν τη διαχειριστική πληροφορία μπορεί να αποφέρει σημαντική μείωση στη διαχειριστική κίνηση. Μάλιστα, όσο αυξάνεται ο αριθμός των εμπλεκομένων μεταβλητών στη διαχειριστική πράξη τόσο μεγαλύτερο είναι το κέρδος. Αξίζει να σημειωθεί ότι όταν το πλήθος το μεταβλητών είναι μεγάλο (συνήθως μεγαλύτερο του 100), η συμπίεση των SOAP/XML φακέλων μπορεί να δώσει μικρότερη κίνηση από εκείνη του SNMP ακόμη και στη βέλτιστη για το SNMP περίπτωση (*GetBulk*).
3. Ωστόσο, η συμπίεση των SOAP/XML μηνυμάτων απαιτεί πρόσθετους πόρους στον πράκτορα και επιφέρει επιπλέον καθυστερήσεις στη διαχείριση.
4. Η κωδικοποίηση SOAP/XML, όταν υλοποιηθεί με χρήση κατάλληλων εργαλείων (π.χ. του gSOAP, που στοχεύει στη χαμηλή κατανάλωση πόρων και όχι του WASP που χρησιμοποιεί γενικής χρήσης γραμματικούς αναλυτές) δεν επιφέρει σημαντικές καθυστερήσεις στο χρόνο διαχείρισης.
5. Ωστόσο η διαδικασία της ενορχήστρωσης μπορεί να επιφέρει τις μεγαλύτερες καθυστερήσεις στη διαχείριση, εάν οι μέθοδοι που την υλοποιούν δεν είναι οι βέλτιστες. Κάτι τέτοιο συμβαίνει στη περίπτωση πολλών εμπορικών SNMP πρακτόρων.
6. Οι απαιτήσεις σε πόρους από ένα XML/HTTP πράκτορα, που είναι υλοποιημένος με κατάλληλα εργαλεία (π.χ. gSOAP), είναι συγκρίσιμες με εκείνες των SNMP πρακτόρων και αποδεκτές ακόμη και για ενσωματωμένα συστήματα με σχετικά περιορισμένες δυνατότητες.

Παρόλο που οι παραπάνω εργασίες φωτίζουν σημαντικά τη συμπεριφορά της XML/HTTP τεχνολογίας στη διαχείριση των δικτύων, ωστόσο δεν παύουν να παρουσιάζουν κάποιες αδυναμίες και παραλείψεις. Αυτές μπορούν να συνοψιστούν στα ακόλουθα:

1. Καμία από τις προαναφερθείσες εργασίες δεν μελετά κάποιο συγκεκριμένο πρότυπο διαχείρισης, που να βασίζεται στην XML/HTTP απλά γιατί μέχρι και τα μέσα του 2004 δεν υπήρχε κάτι τέτοιο. Η πιο σημαντική προσπάθεια που υπήρχε μέχρι τότε ήταν το πρωτόκολλο Net-Conf [41] της IETF, το οποίο εξαιτίας των πολλών προβλημάτων του [42] δεν έχει γίνει αποδεκτό μέχρι και σήμερα. Έτσι όλες οι εργασίες περιορίστηκαν σε υποθετικές διαχειριστικές μεθόδους τις οποίες συνήθως οι συγγραφείς μοντελοποιούσαν με τη γλώσσα περιγραφής των υπηρεσιών ιστού (WSDL).
2. Όλες οι εργασίες μελετούν το πρόβλημα της διαχείρισης των δικτύων μέσω ευρυζωνικών συνδέσεων υψηλών ταχυτήτων (100Mbps). Καμία δεν μελετά περιβάλλοντα διαχείρισης με χαμηλότερους ρυθμούς μετάδοσης όπως είναι αυτό της παροχής υπηρεσιών στους οικιακούς συνδρομητές. Άμεση συνέπεια αυτού είναι να μην γίνεται καθόλου λόγος για το κέρδος, που μπορεί να επιτευχθεί στο χρόνο διαχείρισης, εξαιτίας της μείωσης του όγκου των διαχειριστικών δεδομένων από τη συμπίεση. Μάλιστα στην εργασία [6] θεωρείται δεδομένο ότι η εφαρμογή της συμπίεσης θα επιφέρει περαιτέρω καθυστερήσεις στη διαχειριστική διαδικασία. Η εργασία [5] εφαρμόζει τη συμπίεση και μετρά τις καθυστερήσεις αυτές. Ωστόσο η χρήση μιας συγκεκριμένης, μη βέλτιστης υλοποίησης [39] οδηγεί τους συγγραφείς στο συμπέρασμα, ότι η συμπίεση μπορεί να είναι πολύ πιο δαπανηρή σε χρόνο από ότι και η ίδια η SOAP/XML κωδικοποίηση.
3. Οι περισσότερες εργασίες που μέτρησαν τους διαχειριστικούς χρόνους εξομοίωσαν την πλατφόρμα του πράκτορα με προσωπικούς υπολογιστές υψηλών δυνατοτήτων (π.χ. Pentium III στα 1GhZ στην εργασία [6] και Pentium III στα 800 MhZ στην εργασία [5]). Μόνο η εργασία [8] χρησιμοποίησε ένα ενσωματωμένο σύστημα αλλά και αυτή δεν μελέτησε συνολικά το πρόβλημα, αφού έκανε αποκλειστική χρήση της XML (χωρίς SOAP) για να μεταφέρει τη διαχειριστική πληροφορία. Παρόλο που οι μετρήσεις γίνονταν στη ίδια πλατφόρμα για όλες τις τεχνολογίες (XML/HTTP, SNMP και CORBA), και τα αποτελέσματά τους δόθηκαν συγκριτικά, ωστόσο (εξαιτίας της πληθώρας διαθέσιμων πόρων δικτύου και επεξεργασίας) οι τιμές των διαχειριστικών χρόνων είναι τόσο χαμηλές, που δεν απεικονίζουν τις πραγματικές διαστάσεις του προβλήματος. Έτσι, οι επιδόσεις της XML/HTTP είναι φαινομενικά αποδεκτές ακόμα και για τις υλοποιήσεις εκείνες, που οι χρόνοι διαχείρισης είναι έως και μια τάξη μεγέθους μεγαλύτεροι από εκείνους του SNMP ή της CORBA.

3.5. Η εργασία της διατριβής

Το Μάιο του 2004 το DSLF forum [43] ανακοίνωσε ένα πρωτόκολλο διαχείρισης, που βασίζεται στην τεχνολογία SOAP/XML/HTTP και στοχεύει στην απομακρυσμένη διαχείριση του συνδρομητικού εξοπλισμού. Το πρότυπο αυτό [11] ονομάζεται *πρωτόκολλο απομακρυσμένης διαχείρισης συνδρομητικού εξοπλισμού* (Customer Premises Equipment WAN Management Protocol - CWMP). Όπως προαναφέρθηκε και στο δεύτερο κεφάλαιο αυτή τη στιγμή το CWMP αποτελεί ίσως την πιο ώριμη XML/HTTP λύση για την

απομακρυσμένη διαχείριση. Ενδεικτικό της ωριμότητάς του είναι ότι παρόλο που το CWMP ανακοινώθηκε μόλις πριν από δύο χρόνια, ήδη μεγάλες εταιρίες (D-Link, Thomson, Alcatel, Cisco κ.α.) το υποστηρίζουν στο δικτυακό εξοπλισμό τους. Παράλληλα γίνεται σημαντική δουλειά για την προτυποποίηση των CWMP μοντέλων με στόχο τη διαχείριση μιας μεγάλης γκάμας συσκευών συμπεριλαμβανομένου τα VoIP τηλέφωνα, τους συνδρομητικούς αποκωδικοποιητές τις παιχνιδομηχανές κ.α. Βέβαια δεν θα πρέπει να αγνοηθεί η προσπάθεια της IETF για την προτυποποίηση του NetConf πρωτοκόλλου [41] η οποία ωστόσο δεν έχει ακόμα καρποφορήσει για τους λόγους που αναφέρθηκαν στο δεύτερο κεφάλαιο.

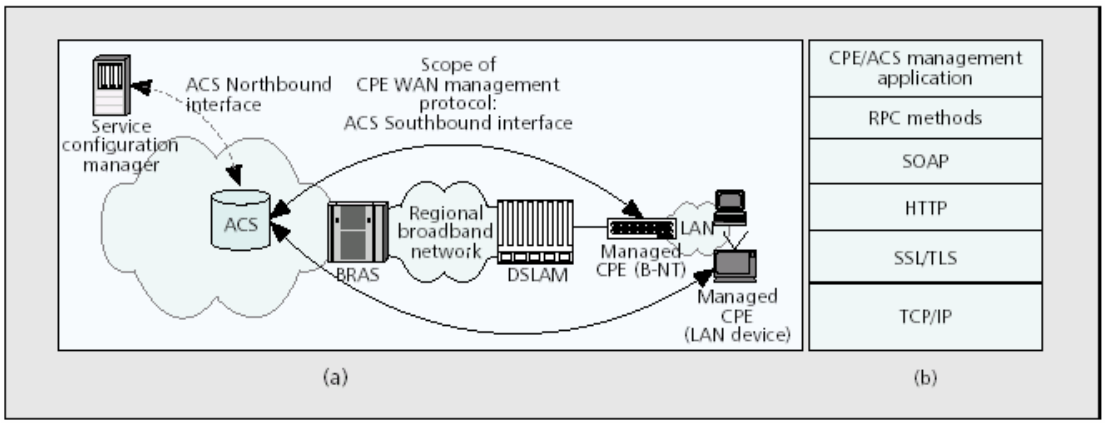
Αντίθετα με τις προαναφερθείσες εργασίες οι οποίες ορίζουν αυθαίρετα ένα XML/HTTP πρωτόκολλο χωρίς να εξετάζουν κάποιο συγκεκριμένο πρότυπο, η παρούσα εργασία διερευνά το πρότυπο CWMP. Μάλιστα η μελέτη εστιάζει στο περιβάλλον διαχείρισης του συνδρομητικού εξοπλισμού το οποίο παρουσιάζει ορισμένες ιδιαιτερότητες. Οι κυριότερες εξ αυτών είναι το περιορισμένο διαθέσιμο εύρος ζώνης του συνδρομητικού δικτύου στο δίκτυο πρόσβασης, ο μεγάλος αριθμός διαχειριζόμενων συσκευών καθώς και οι υψηλές απαιτήσεις στους χρόνους διαχείρισης.

3.5.1. Το περιβάλλον διαχείρισης

Το CWMP πρωτόκολλο χρησιμοποιείται από τους παρόχους για τη διαχείριση των υπηρεσιών, που προσφέρονται μέσω του δικτύου πρόσβασης καθώς και για τη συντήρηση του συνδρομητικού εξοπλισμού. Το CWMP υποστηρίζει μια γκάμα λειτουργιών διαχείρισης η οποία συνίσταται στην αυτόματη ρύθμιση, τη δυναμική παροχή υπηρεσίας, τη διαχείριση λογισμικού, την παρακολούθηση της κατάστασης και των επιδόσεων του συστήματος, τη διάγνωση σφαλμάτων καθώς και σε μηχανισμούς ασφαλείας. Προκειμένου το CWMP να υλοποιήσει τα παραπάνω χρησιμοποιεί αρχιτεκτονικά στοιχεία, που έχουν να κάνουν με τα ακόλουθα: (i) τις παραμέτρους για τη ρύθμιση και την παρακολούθηση συσκευών και υπηρεσιών, (ii) τη μεταφορά αρχείων για αναβάθμιση λογισμικού, (iii) τις ενημερώσεις συμβάντων και (iv) την ασύγχρονη εκκίνηση των διαχειριστικών συνεδριών από τον διαχειριστή με στόχο την υποστήριξη υπηρεσιών, που απαιτούν επαναρύθμιση σε σχεδόν πραγματικό χρόνο.

Η CWMP αρχιτεκτονική ορίζει δυο κύριες οντότητες: (i) τον *εξυπηρετητή αυτόματης ρύθμισης* (Automatic Configuration Server - ACS) ο οποίος αποτελεί τη διαχειρίζουσα οντότητα και (ii) τον CWMP πελάτη (CWMP Client) ο οποίος αποτελεί τη διαχειριζόμενη οντότητα. Για λόγους απλούστευσης αλλά και για τους λόγους που αναφέρθηκαν στην αρχή της παραγράφου 3.3.1 από εδώ και στο εξής ο εξυπηρετητής αυτόματης ρύθμισης θα καλείται *ACS διαχειριστής*, ενώ ο CWMP πελάτης θα καλείται *CWMP πράκτορας*. Καταρχάς το πρωτόκολλο ενεργοποιείται μόλις η συσκευή αποκτήσει μια IP διεύθυνση. Σύμφωνα με την αρχιτεκτονική του CWMP (σχήμα 3-16α), ένας ACS διαχειριστής φιλοξενείται είτε στο δίκτυο πρόσβασης, είτε στο δίκτυο κορμού και είναι επιφορτισμένος με την εφαρμογή των αρχικών ρυθμίσεων και την περαιτέρω διαχείριση ενός μεγάλου αριθμού συνδρομητικών συσκευών (CPE) μέσω του CWMP πρωτοκόλλου. Ο ACS διαχειριστής έχει δύο διεπαφές, που ονομάζονται *βόρειο σύνορο* (Northbound) και *νότιο σύνορο* (Southbound). Η Southbound

διεπαφή εξυπηρετεί την επικοινωνία του ACS με το συνδρομητικό εξοπλισμό και σχετίζεται με την παρούσα εργασία. Αντίθετα η Northbound διεπαφή εξυπηρετεί την ανταλλαγή της διαχειριστικής πληροφορίας μεταξύ του ACS διαχειριστή και των παρόχων διότι οι τελευταίοι είναι εκείνοι που προσφέρουν τις υπηρεσίες και εφαρμόζουν τις υψηλού επιπέδου πολιτικές διαχείρισης. Η διεπαφή αυτή δεν θα απασχολήσει την παρούσα εργασία.



Σχήμα 3-16. (α) Το περιβάλλον διαχείρισης και (β) η στοίβα πρωτόκολλων του CWMP.

Ο ACS διαχειριστής καλεί απομακρυσμένα προκαθορισμένες εντολές ρύθμισης και παρακολούθησης οι οποίες εκτελούνται από το CWMP πράκτορα της συσκευής. Για το σκοπό αυτό χρησιμοποιούνται οι SOAP κλήσεις απομακρυσμένων διεργασιών (Remote Procedure Calls -RPCs). Στην αντίθετη κατεύθυνση ο CWMP πράκτορας μπορεί να καλέσει και αυτός ορισμένες SOAP κλήσεις για αρχικοποίηση και ενημέρωση συμβάντων. Το σύνολο αυτών των αμφίδρομων αλληλεπιδράσεων συνιστά μια διαχειριστική συνεδρία.

Το σχήμα 3-16β απεικονίζει τη στοίβα πρωτοκόλλων που χρησιμοποιείται. Πάνω από ένα ασφαλή (SSL/TLS) και συνδεσμοκατευθυνόμενο μηχανισμό μεταφοράς (TCP), ο ACS επιτελεί το ρόλο ενός HTTP εξυπηρετητή, ενώ ο πράκτορας της συνδρομητικής συσκευής (CPE) επιτελεί το ρόλο ενός HTTP πελάτη. Μια τέτοια διευθέτηση θα οδηγούσε σε διαδράσεις που θα εκκινούνταν μόνο από τον CWMP πράκτορα καθώς αυτός περιλαμβάνει τον HTTP πελάτη. Αυτό θα είχε σαν συνέπεια ο πράκτορας να ρωτά περιοδικά τον ACS και να ξεκινά μια συνεδρία διαχείρισης στην περίπτωση που εκείνος έχει να εφαρμόσει κάποια νέα ρύθμιση. Κάτι τέτοιο απαιτούσε σημαντικούς δικτυακούς πόρους ιδιαίτερα στις περιπτώσεις εκείνες όπου η ρύθμιση πρέπει να τελεστεί σε σχεδόν πραγματικό χρόνο και επομένως οι ερωτήσεις προς τον ACS θα γίνονται με μεγάλη συχνότητα. Η CWMP αρχιτεκτονική προτείνει μια λύση η οποία συνίσταται στη μερικώς αμφίδρομη εκκίνηση της διαχειριστικής συνεδρίας. Τώρα ο ACS περιλαμβάνει επίσης και ένα HTTP πελάτη ο οποίος συνδέεται με ένα HTTP εξυπηρετητή, που ενσωματώνεται επιπλέον στον CWMP πράκτορα με μόνο σκοπό να «ακούσει» την πρόθεση του ACS για επικοινωνία. Σε αυτό το ρόλο ο ACS στέλνει ένα άδειο *HTTP GET* μήνυμα (*Connection Request*) στο οποίο ο πράκτορας δεν απαντά άμεσα αλλά εισέρχεται σε μια κατάσταση έναρξης της διαχειριστικής συνεδρίας. Με αυτό τον τρόπο εξαφανίζεται η ανάγκη για συνεχείς ερωτήσεις στον ACS.

Ένα στρώμα πιο πάνω, τα SOAP μηνύματα εντός των HTTP πακέτων αναλαμβάνουν τη μεταφορά της διαχειριστικής πληροφορίας. Στο περιβάλλον των κλήσεων απομακρυσμένων διεργασιών (RPCs) η πληροφορία αυτή περιλαμβάνει το όνομα της μεθόδου, που καλείται απομακρυσμένα, τις παραμέτρους εισόδου της και τις επιστρεφόμενες τιμές. Η κάθε κλήση απομακρυσμένης μεθόδου κωδικοποιείται αποκλειστικά σε XML εντός ενός SOAP φακέλου. Σε αυτό το επίπεδο ο ACS διαχειριστής λειτουργεί σαν ένας πελάτης, που εκδίδει διαχειριστικές αιτήσεις προς τον πράκτορα ο οποίος αναλαμβάνει να τις εξυπηρετήσει λειτουργώντας έτσι σαν εξυπηρετητής. Μια SOAP αίτηση από τον ACS διαχειριστή προς τον CWMP πράκτορα στέλνεται πάνω από μια HTTP απάντηση. Τυπική περίπτωση αποτελούν οι CWMP μέθοδοι που ζητούν (*GetParameterValues*) ή θέτουν (*SetParameterValues*) τις παραμέτρους της συσκευής. Η SOAP απάντηση του πράκτορα επιστρέφει στον ACS μέσω ενός επακόλουθου *HTTP POST* μηνύματος. Τυπική περίπτωση αποτελούν οι απαντήσεις των παραπάνω CWMP μεθόδων οι οποίες στην πρώτη περίπτωση (*GetParameterValues*) μεταφέρουν τα δεδομένα, που συνέλεξε ο πράκτορας της συσκευής, ενώ στη δεύτερη (*SetParameterValues*) αναφέρουν την κατάσταση της εκτέλεσης των ζητούμενων ρυθμίσεων. Τέλος όσον αφορά στην τυποποίηση των παραμέτρων, το CWMP χρησιμοποιεί ένα μικρό σύνολο SOAP τύπων.

Η διαχείριση του συνδρομητικού εξοπλισμού παρουσιάζει ορισμένες ιδιαιτερότητες σε σχέση με τη διαχείριση του δικτυακού εξοπλισμού των παρόχων. Τα βασικότερα χαρακτηριστικά της διαχείρισης των συνδρομητικών συσκευών είναι το περιορισμένο εύρος ζώνης, που υπαγορεύεται κάθε φορά από την τεχνολογία πρόσβασης (xDSL, FTTH κλπ), οι χαμηλές δυνατότητες επεξεργασίας και αποθήκευσης των συνδρομητικών συσκευών σε σχέση με εκείνες του ακριβού δικτυακού εξοπλισμού των παρόχων αλλά και ο μεγάλος αριθμός των διαχειριζόμενων συσκευών. Επομένως σε αυτό το περιβάλλον ο υψηλός όγκος της διαχειριστικής πληροφορίας μπορεί να δημιουργήσει προβλήματα κλιμακωσιμότητας αλλά και καθυστέρησης στην εφαρμογή των ρυθμίσεων και ενδεχομένως και στην ενεργοποίηση των υπηρεσιών. Τέτοιου είδους προβλήματα μπορούν να ανεβάσουν σημαντικά το κόστος της διαχείρισης (περισσότερες ACS πλατφόρμες για την αποφυγή συμφόρησης) αλλά και να δημιουργούν δυσάρεστη εμπειρία στους συνδρομητές ιδιαίτερα σε υπηρεσίες, που απαιτούν επαναρύθμιση σε σχεδόν πραγματικό χρόνο. Για παράδειγμα, όταν ο χρήστης επιλέγει μια καλύτερη ποιότητα για τη Video κλήση του δεν θα ήθελε να περιμένει παραπάνω από 1-2 δευτερόλεπτα για τη ρύθμιση των νέων QoS παραμέτρων και την έναρξη της κλήσης. Αυτού του είδους οι απαιτήσεις σε συνδυασμό με τις περιορισμένες ταχύτητες πρόσβασης δεν αφήνουν πολλά περιθώρια για άσκοπη χρήση του εύρους ζώνης από το διαχειριστικό πρωτόκολλο.

Παρακάτω γίνεται μια θεωρητική ανάλυση του όγκου της παραγόμενου κίνησης διαχείρισης από το CWMP πρωτόκολλο καθώς και της επίδρασης του όγκου αυτού στο συνολικό χρόνο διαχείρισης. Επίσης διερευνώνται οι συνθήκες κάτω από τις οποίες η συμπίεση των μηνυμάτων διαχείρισης είναι επικερδής και προτείνεται η ιδέα της επιλεκτικής συμπίεσης. Για την αποτίμηση της προτεινόμενης λύσης σχεδιάζεται και υλοποιείται ένα πρωτότυπο

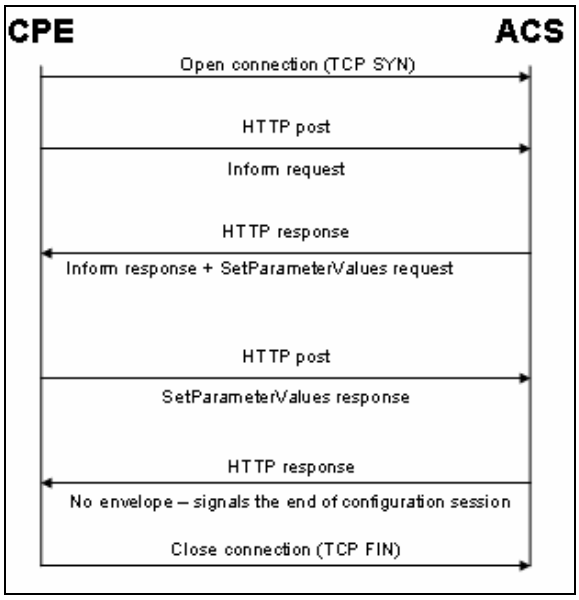
σύστημα CWMP διαχείρισης, που αποτελείται από έναν ACS διαχειριστή και ένα CWMP πράκτορα. Τέλος εκτελούνται διάφορα πειράματα και αξιολογούνται τα αποτελέσματά τους.

3.5.2. Ο όγκος της κίνησης διαχείρισης

Προκειμένου να αποκτηθεί μια ολοκληρωμένη εικόνα σχετικά με την κίνηση, που παράγεται από το CWMP πρωτόκολλο μελετώνται δύο διαφορετικές περιπτώσεις. Για κάθε μια περίπτωση κατασκευάζονται τα μηνύματα, που ορίζει το πρωτόκολλο και μετράται ο όγκος τους, συμπεριλαμβανομένου και των HTTP/TCP/IP υπερκεφαλισμών. Η πρώτη περίπτωση περιλαμβάνει μια πλήρη συνεδρία, έτσι όπως αυτή ορίζεται από το CWMP, κατά την οποία τίθεται η τιμή μιας μεταβλητής. Αντίθετα στη δεύτερη περίπτωση μελετάται το πρόβλημα της παρακολούθησης ενός μεταβλητού αριθμού παραμέτρων της συνδρομητικής συσκευής.

3.5.2.1. Περίπτωση 1: Μια πλήρης διαχειριστική συνεδρία

Το σχήμα 3-17 απεικονίζει ένα τυπικό σενάριο ρύθμισης κατά το οποίο ο ACS δίνει εντολή στον πράκτορα της συνδρομητικής συσκευής (CPE) να θέσει μια από τις παραμέτρους της συσκευής.



Σχήμα 3-17. Μια τυπική CWMP συνεδρία.

Σύμφωνα με το πρωτόκολλο, ο πράκτορας ξεκινά πάντα τη διαχειριστική συνεδρία στέλνοντας ένα *Inform* μήνυμα-αίτηση (request) στον ACS. Το SOAP αυτό μήνυμα μεταφέρεται μέσω ενός *HTTP POST* μηνύματος και περιέχει πληροφορία σχετικά με την τρέχουσα κατάσταση της συσκευής αλλά και τη διαθεσιμότητα του πράκτορα να λαμβάνει εντολές από τον ACS. Το σχήμα 3-18 απεικονίζει ένα μέρος αυτού του μηνύματος. Το κύριο μέρος του μηνύματος συνίσταται από στατική πληροφορία η οποία ωστόσο είναι υποχρεωτική για το CWMP πρωτόκολλο. Το αντίστοιχο XML αρχείο είναι αρκετά μεγάλο (τουλάχιστον 2,2 Kbytes) παρόλο που μεταφέρει μόνο 15 ζευγάρια παραμέτρων – τιμών. Όταν ο ACS λάβει το παραπάνω μήνυμα απαντά με ένα *Inform* μήνυμα-απάντηση (response).

Το τελευταίο μεταφέρεται μέσω ενός HTTP μηνύματος απάντησης, περιλαμβάνει μόνο ένα ζευγάρι παραμέτρου-τιμής και το μέγεθός του δεν ξεπερνά τα 400 bytes.

Μετά την αρχικοποίηση της συνεδρίας, ο ACS διαχειριστής στέλνει ένα *SetParameterValues* μήνυμα-αίτηση προκειμένου να θέσει μια παράμετρο της συσκευής. Το σχήμα 3-6, που δόθηκε σαν παράδειγμα στην αρχή του κεφαλαίου απεικονίζει το πλήρες μήνυμα. Παρόλο που το μήνυμα αυτό μεταφέρει μόνο ένα ζεύγος παραμέτρου-τιμής το μέγεθός του ξεπερνά τα 600 bytes. Μόλις ο πράκτορας της συσκευής λάβει αυτό το SOAP μήνυμα θέτει τη ζητούμενη παράμετρο και στη συνέχεια αποστέλλει ένα *SetParameterValues* μήνυμα–απάντηση προκειμένου να ενημερώσει τον ACS σχετικά με το αποτέλεσμα της αιτούμενης διαχειριστικής ενέργειας. Το μέγεθος του μηνύματος αυτού δεν ξεπερνά τα 400 bytes. Τέλος, σύμφωνα με το πρωτόκολλο ο ACS στέλνει ένα άδειο HTTP μήνυμα-απάντηση για να διατάξει τον πράκτορα (HTTP πελάτη) να περατώσει τη συνεδρία.

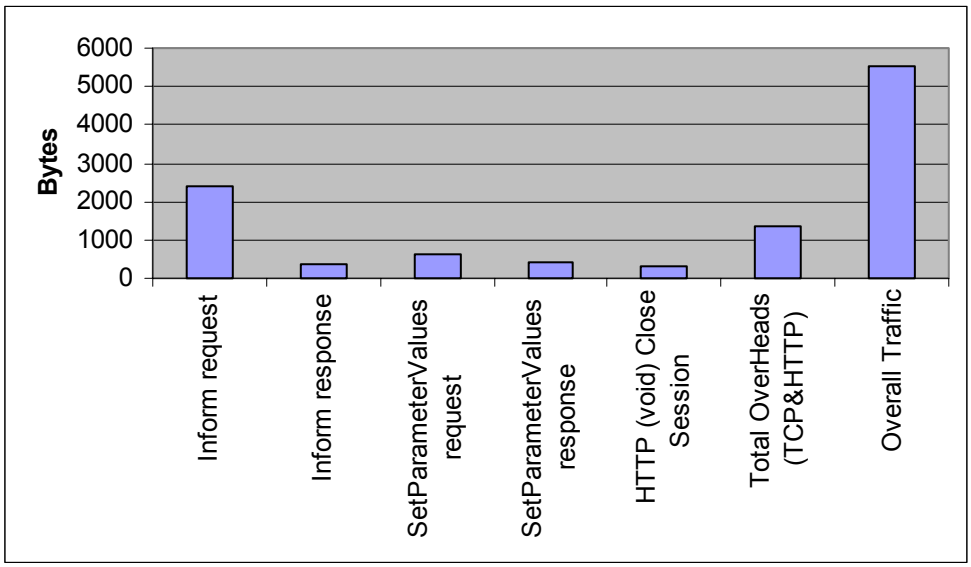
```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
<soap:Body>
<cwmp:Inform>
<DeviceId xsi:type="cwpm:DeviceIdStruct">
<Manufacturer xsi:type="xsd:string[64]">My_CPE's_Manufacturer</Manufacturer>
<OUI xsi:type="xsd:string[6]">00D09E</OUI>
<ProductClass xsi:type="xsd:string[64]">HomeGateway_with_QoS_Support</ProductClass>
<SerialNumber xsi:type="xsd:string[64]">6789-34554555355</SerialNumber>
</DeviceId>
<Event soap: arrayType="cwmp:EventStruct[1]">
<EventStruct>
<name>EventCode</name>
<value xsi:type="xsd:string">"6 CONNECTION REQUEST"</value>
</EventStruct>
</Event>
<MaxEnvelopes>0</MaxEnvelopes>


<CurrentTime>0000-00-02T03:04:05</CurrentTime>
<RetryCount>1</RetryCount>
<ParameterList soap: arrayType="cwmp:ParameterValueStruct[7]">
<ParameterValueStruct>
<name>InternetGatewayDevice.DeviceInfo.SpecVersion</name>
<value xsi:type="xsd:string">1.0</value>
</ParameterValueStruct>

 .          .

 .          .

 .          .

 <ParameterValueStruct>
 <name>InternetGatewayDevice.WANDevice.1.WANConnectionDevice.1.WANPPPConnection.1.ExternalIPAddress</name>
 <value xsi:type="xsd:string">147.102.19.22</value>
 </ParameterValueStruct>
 </ParameterList>
 </cwmp:Inform>
 </soap:Body>
 </soap:Envelope>
```

Σχήμα 3-18. Απόσπασμα από μια τυπική Inform αίτηση.

Το σχήμα 3-19 δείχνει ότι ακόμα και για την απλούστερη περίπτωση CWMP διαχείρισης (ρύθμιση μιας μόνο μεταβλητής) ο όγκος των δεδομένων, που ανταλλάσσονται μέσω του δικτύου πρόσβασης ξεπερνά τα 5,5 Kbytes. Στο σημείο αυτό πρέπει να τονιστεί ότι ο όγκος των HTTP/TCP/IP υπερκεφαλισμών, που εμφανίζεται στο σχήμα 3-19 είναι ο ελάχιστος δυνατός καθώς το παράδειγμα του σχήματος 3-17, που εξετάστηκε εκμεταλλεύεται τη δυνατότητα του CWMP πρωτοκόλλου να ενθυλακώνει περισσότερους τους ενός SOAP φακέλου σε κάθε HTTP μήνυμα. Το μήνυμα αυτό μπορεί να προέρχεται είτε από τον ACS διαχειριστή, είτε από τον πράκτορα της συσκευής. Στην εξεταζόμενη περίπτωση το μήνυμα

προέρχεται από τον ACS ο οποίος στην ίδια HTTP απάντηση ενθυλακώνει το *Inform* μήνυμα-απάντηση και το *SetParameterValues* μήνυμα-αίτηση.

Το γεγονός αυτό θα μπορούσε να δημιουργήσει προβλήματα κλιμακωσιμότητας εάν αναλογισθεί κανείς ότι οι μελλοντικές υπηρεσίες (π.χ.VoIP, TVoIP, Video conferencing) αναμένονται πολύ απαιτητικές όσον αφορά στην ανάγκη τους για επαναρύθμιση. Αυτό σημαίνει ότι ενδεχομένως στο μέλλον ένας μεγάλος αριθμός συνδρομητικών συσκευών (π.χ. δικτυακές πύλες, παιχνιδομηχανές, προσωπικοί υπολογιστές, κάμερες, ηχοσυστήματα, Video/VoIP τηλέφωνα, συνδρομητικοί αποκωδικοποιητές) να ζητά πολύ συχνά από τον ACS την επαναρύθμισή του για την ενεργοποίηση νέων δυνατοτήτων. Μάλιστα η κατάσταση γίνεται ακόμα δυσμενέστερη αν συνυπολογιστεί το γεγονός ότι η CWMP μοντελοποίηση των αντίστοιχων συνδρομητικών συσκευών ορίζει εκατοντάδες παραμέτρους για κάθε συσκευή (μόνο για τη δικτυακή πύλη ορίζονται περίπου 500 παράμετροι [44]).



Σχήμα 3-19. Η ανάλυση του όγκου της κίνησης για μια τυπική διαχειριστική συνεδρία.

Επιπλέον πέραν της ανάγκης για επαναρύθμιση, το CWMP πρωτόκολλο ορίζει ότι ένα *Inform* μήνυμα-αίτηση θα στέλνεται από τον πράκτορα στον ACS κάθε φορά που συμβαίνει μια από τις παρακάτω συνθήκες: (i) κατά την αρχική εγκατάσταση, (ii) όταν τίθεται σε λειτουργία ή επανεκκινεί η συσκευή, (iii) μια φορά ανά ένα προκαθορισμένο από τον ACS χρονικό διάστημα, (iv) οποτεδήποτε αλλάζει μια κρίσιμη παράμετρος στη συσκευή για την οποία ο πράκτορας οφείλει να ενημερώσει τον ACS (ενημέρωση συμβάντος). Κάποια από τα παραπάνω γεγονότα συμβαίνουν μια φορά την ημέρα η ακόμα και μια φορά το μήνα. Ωστόσο υπάρχουν και ορισμένα που συμβαίνουν πολύ συχνότερα όπως είναι εκείνο της εκκίνησης μιας συσκευής. Σε ένα συνδρομητικό δίκτυο αναμένεται ότι οι συσκευές με τις οποίες ο χρήστης αλληλεπιδρά άμεσα (π.χ. παιχνιδομηχανή, συνδρομητικός αποκωδικοποιητής) θα ανοίγουν και θα κλείνουν με ένα ρυθμό τουλάχιστον μια τάξη μεγέθους μεγαλύτερο από εκείνο των δικτυακών συσκευών (π.χ. πύλες και σημεία ασύρματης πρόσβασης).

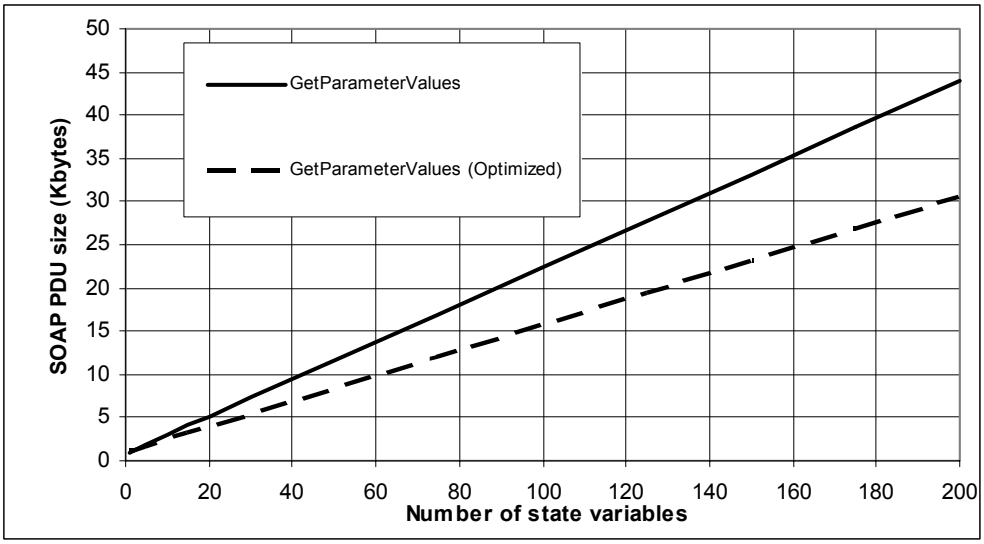
Για να δοθεί μια εικόνα για το μέγεθος του προβλήματος δίνεται το ακόλουθο παράδειγμα. Έστω ότι στο μέλλον κάθε συνδρομητικό δίκτυο έχει κατά μέσο όρο πέντε CWMP συσκευές

(π.χ. ένα συνδρομητικό αποκωδικοποιητή, ένα VoIP τηλέφωνο, ένα προσωπικό υπολογιστή, μια παιχνιδομηχανή για δικτυακά παιχνίδια και μια δικτυακή πύλη). Μάλιστα οι συσκευές αυτές πρόκειται να επικοινωνούν κατά μέσο όρο τέσσερις φορές την ημέρα με τον ACS διαχειριστή (π.χ. δύο φορές την ημέρα για κάθε εκκίνησή τους, μία φορά για περιοδική παρακολούθηση μιας παραμέτρου της συσκευής και άλλη μία φορά για επαναρύθμιση μόνο μιας παραμέτρου). Μόνο για αυτή τη στοιχειώδη διαχείριση θα απαιτούνταν περισσότερα από 110 Kbytes ανά συνδρομητικό δίκτυο, κάτι που ισοδυναμεί με τουλάχιστον 1Gigabyte ημερησίως για κάθε 10.000 συνδρομητές (συνδρομητικά δίκτυα). Μάλιστα ο πραγματικός όγκος της κίνησης διαχείρισης αναμένεται αρκετά μεγαλύτερος και σχετίζεται άμεσα με την πολυπλοκότητα των διαχειριστικών ενεργειών και συνεπώς με τον αριθμό των παραμέτρων, που θα ανακτώνται ή θα ρυθμίζονται κάθε φορά. Η περίπτωση αυτή εξετάζεται παρακάτω.

3.5.2.2. Περίπτωση 2: Το πρόβλημα της παρακολούθησης

Προκειμένου να δοθεί μια καλύτερη εικόνα για το μέγεθος της αναμενόμενης κίνησης διαχείρισης η παρακάτω ανάλυση προσπαθεί να συσχετίσει τον όγκο των αντίστοιχων μηνυμάτων με τον αριθμό των ανακτηθέντων παραμέτρων. Για το λόγο αυτό μελετάται η περίπτωση της παρακολούθησης (monitoring) των μεταβλητών κατάστασης μιας δικτυακής πύλης μέσω της *GetParameterValues* εντολής που ορίζει το πρωτόκολλο. Στα πλαίσια αυτής της εργασίας κατασκευάστηκε ένα μεγάλο πλήθος *GetParameterValues* μηνυμάτων με τρόπο τέτοιο ώστε το καθένα να περιέχει διαφορετικό αριθμό παραμέτρων της δικτυακής πύλης, έτσι όπως αυτές ορίζονται στη CWMP μοντελοποίηση [44]. Μάλιστα για κάθε παράμετρο έχουν δοθεί υποθετικές τιμές οι οποίες ωστόσο είναι σύμφωνες με τις αναμενόμενες από τη μοντελοποίηση. Στο σημείο αυτό θα πρέπει να τονιστεί ότι η διαδικασία της παρακολούθησης (*GetParameterValues*) δεν διαφέρει σημαντικά από τη διαδικασία της ρύθμισης (*SetParameterValues*) όσον αφορά στην παραγόμενη κίνηση διότι και οι δύο αντίστοιχες εντολές έχουν πανομοιότυπη δομή και φέρουν τα ονόματα των παραμέτρων και τις τιμές τους. Η μόνη ουσιαστική διαφοροποίηση είναι ότι στην περίπτωση της ρύθμισης ο κύριος όγκος της κίνησης έχει κατεύθυνση προς τον πράκτορα, ενώ στην περίπτωση της παρακολούθησης έχει κατεύθυνση προς το διαχειριστή. Αυτό πρακτικά μπορεί να επηρεάσει το χρόνο διαχείρισης, όταν πρόκειται για τεχνολογίες πρόσβασης με ασύμμετρους ρυθμούς μετάδοσης. Από εδώ και στο εξής όποια θεώρηση γίνεται για το πρόβλημα της παρακολούθησης θα ισχύει και για το πρόβλημα της ρύθμισης έχοντας βέβαια υπόψη την προαναφερθείσα διαφοροποίηση.

Το σχήμα 3-20 απεικονίζει το μέγεθος των SOAP μηνυμάτων της *GetParameterValues* εντολής για διαφορετικό αριθμό μεταβλητών κατάστασης. Για κάθε αριθμό παραμέτρων έχουν μετρηθεί τα *GetParameterValues* μηνύματα αίτησης, που περιέχουν τις παραμέτρους των οποίων ο διαχειριστής ζητά την τιμή τους, αλλά και τα *GetParameterValues* μηνύματα απάντησης, που περιλαμβάνουν τα ζεύγη παραμέτρων και τιμών.



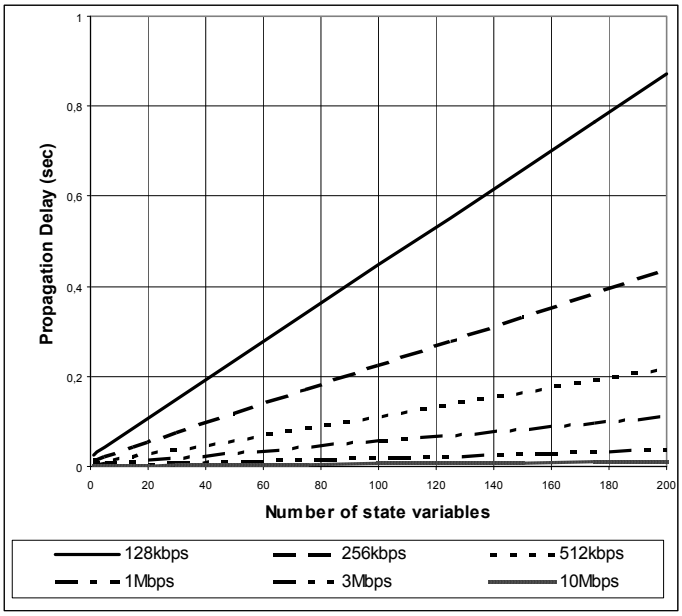
Σχήμα 3-20. Ο όγκος των SOAP μηνυμάτων κατά την παρακολούθηση.

Οι καμπύλες του σχήματος 3-20 απεικονίζουν το συνολικό μέγεθος των *GetParameterValues* ερωταπαντήσεων. Μάλιστα η πρώτη καμπύλη (*GetParameterValues*) απεικονίζει την περίπτωση εκείνη κατά την οποία το μήνυμα της αίτησης περιέχει τα ονόματα όλων των παραμέτρων των οποίων ζητείται η τιμή τους. Ωστόσο αυτή η περίπτωση δεν θα ισχύει πάντα αφού το CWMP πρωτόκολλο προβλέπει ένα μηχανισμό παρόμοιο με εκείνο της *GetBulk* του SNMP για μαζική ανάκτηση των τιμών των παραμέτρων. Η CWMP μοντελοποίηση χρησιμοποιεί αντικείμενα (objects) και παραμέτρους (parameters) για να εκφράσει τις διαχειριζόμενες λειτουργίες της συνδρομητικής συσκευής (CPE). Οι παράμετροι μεταφέρουν την πραγματική πληροφορία ρύθμισης και παρακολούθησης, ενώ τα αντικείμενα χρησιμοποιούνται για τη λογική ομαδοποίηση των παραμέτρων. Κάθε αντικείμενο αναπαριστά μια συγκεκριμένη περιοχή διαχείρισης εντός της συσκευής ή της υπηρεσίας και μπορεί να περιλαμβάνει παραμέτρους αλλά και άλλα υπό-αντικείμενα. Έτσι λοιπόν όταν ο διαχειριστής επιθυμεί να ανακτήσει ολόκληρη την πληροφορία για ένα διαχειριζόμενο αντικείμενο εντός της συσκευής (δηλαδή όλες τις παραμέτρους, που ανήκουν σε αυτό) χρησιμοποιεί την εντολή *GetParameterValues* με όρισμα μόνο το αντικείμενο εκείνο και όχι τα υπόλοιπα υπό-αντικείμενα και παραμέτρους. Με αυτό τον τρόπο μειώνεται ο όγκος των μηνυμάτων αίτησης (*GetParameterValues* Request) και επομένως ο συνολικός όγκος της παρακολούθησης. Η διακεκομμένη καμπύλη (*GetParameterValues (Optimized)*) απεικονίζει την ακραία περίπτωση κατά την οποία ο διαχειριστής ζητά την τιμή όλων των παραμέτρων με ένα και μόνο αντικείμενο σαν όρισμα (π. χ. όταν το αντικείμενο *InternetGatewayDevice.* χρησιμοποιείται στην αίτηση μπορεί να φέρει όλες τις παραμέτρους της δικτυακής πύλης). Στην περίπτωση αυτή το μέγεθος του *GetParameterValues* μηνύματος της αίτησης δεν ξεπερνά τα 500 bytes. Ωστόσο κάθε υπό -αντικείμενο περιέχει ένα μικρό αριθμό παραμέτρων με αποτέλεσμα στο *GetParameterValues* μήνυμα-αίτηση να χρειάζεται συνήθως παραπάνω από ένα υπό-αντικείμενο σαν όρισμα, προκειμένου να ανακτηθεί επιλεκτικά η πληροφορία από το σύστημα. Επομένως ο πραγματικός όγκος της διαχειριστικής πληροφορίας κατά την παρακολούθηση θα βρίσκεται πάντα μεταξύ των καμπυλών του σχήματος 3-20.

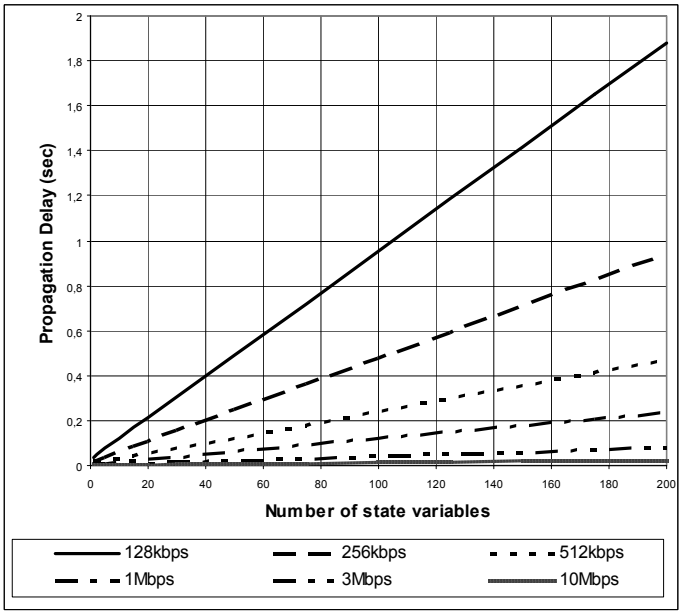
Το γράφημα αυτό αποδεικνύει ότι η χρήση του SOAP/XML για την κωδικοποίηση των διαχειριστικών εντολών έχει σαν συνέπεια μεγάλους υπερκεφαλισμούς στα μηνύματα διαχείρισης ακόμη και για μικρό αριθμό παραμέτρων. Για κάθε 20 παραμέτρους το αντίστοιχο SOAP μήνυμα αυξάνει κατά 3-5Kbytes ανάλογα με τη διασπορά των παραμέτρων στα διάφορα υπό-αντικείμενα, που ορίζει η CWMP μοντελοποίηση. Για να δοθεί μια εικόνα σχετικά με το μέγεθος του προβλήματος θα επεκταθεί το παράδειγμα, που δόθηκε στο τέλος της παραγράφου 3.5.2.1 κατά το οποίο ένα μελλοντικό συνδρομητικό δίκτυο έχει κατά μέσο όρο πέντε CWMP συσκευές. Η καθεμία εξ αυτών θα επικοινωνεί τώρα με τον ACS διαχειριστή σε ημερήσια βάση μία φορά για περιοδική παρακολούθηση 50 παραμέτρων και άλλη μία φορά για την επαναρύθμιση άλλων τόσων. Αν αναλογισθεί κανείς ότι μόνο για τη δικτυακή πύλη η CWMP μοντελοποίηση προβλέπει 500 παραμέτρους, εκ των οποίων τουλάχιστον οι 200 είναι υποχρεωτικές, τότε η παραπάνω θεώρηση φαίνεται λογική. Στην περίπτωση αυτή θα παράγονταν κίνηση τουλάχιστον 21Kbytes (2 x ≈10Kbytes (SOAP PDU) + ≈1Kbyte (HTTP/TCP/IP overheads)) ανά συσκευή και επομένως 105Kbytes ανά συνδρομητικό δίκτυο. Αυτό ισοδυναμεί με τουλάχιστον 1 Gigabyte ημερησίως για κάθε 10.000 συνδρομητές (συνδρομητικά δίκτυα). Συνυπολογίζοντας και τον όγκο που παράγεται κατά την *Περίπτωση 1* φαίνεται ότι για κάθε 10.000 συνδρομητικά νοικοκυριά θα ανταλλάσσονται μέσω του δικτύου πρόσβασης τουλάχιστον 2 Gigabytes διαχειριστικής πληροφορίας ημερησίως. Αν σκεφτεί κανείς ότι η διαχείριση είναι μια λειτουργία του παρόχου για την οποία δεν χρεώνεται άμεσα ο συνδρομητής τότε γίνεται εύκολα αντιληπτό ότι το υψηλό αυτό καταναλισκόμενο εύρος ζώνης εξαιτίας της διαχείρισης μεταφράζεται σε καθαρό κόστος στον πάροχο. Επομένως αποκτά ιδιαίτερη σημασία η όποια προσπάθεια για περιορισμό των υπερκεφαλισμών της κίνησης διαχείρισης, που πηγάζουν κατά κύριο λόγο από τη δαπανηρή κωδικοποίηση της SOAP/XML.

3.5.3. Οι χρόνοι μετάδοσης των μηνυμάτων διαχείρισης

Πέραν του άμεσου κόστους, που επιφέρει η σπάταλη χρήση του εύρους ζώνης, εξαιτίας της SOAP/XML κωδικοποίησης ο μεγάλος όγκος των SOAP μηνυμάτων μπορεί να επηρεάσει και το συνολικό χρόνο διαχείρισης. Αυτό γίνεται ιδιαίτερα έντονο στις περιπτώσεις όπου το διαθέσιμο εύρος ζώνης είναι σχετικά περιορισμένο (όπως στην περίπτωση της ADSL πρόσβασης). Προκειμένου να δοθεί μια εικόνα της επίδρασης του υψηλού όγκου των μηνυμάτων διαχείρισης στο χρόνο διαχείρισης δίνονται παρακάτω οι συνολικοί χρόνοι μετάδοσης των μηνυμάτων αυτών για διάφορες τιμές του εύρους ζώνης. Συγκεκριμένα τα σχήματα 3-21 και 3-22 δίνουν για διαφορετικό αριθμό μεταβλητών κατάστασης τους χρόνους μετάδοσης των *GetParameterValues* μηνυμάτων αίτησης ($t_{ΜΜΑ}$) και απάντησης ($t_{ΜΜΑΠ}$), αντίστοιχα. Στους χρόνους αυτούς δεν συμπεριλαμβάνονται οι HTTP/TCP/IP υπερκεφαλισμοί. Όπως εξηγήθηκε και παραπάνω ο αριθμός των παραμέτρων, που συμμετέχουν στο μήνυμα της αίτησης και επομένως και οι χρόνοι μετάδοσης του σχήματος 3-21 εξαρτώνται από τη διασπορά των διαχειριζόμενων κάθε φορά παραμέτρων στα αντικείμενά τους.



Σχήμα 3-21. Οι χρόνοι μετάδοσης του *GetParameterValues* μηνύματος αίτησης.



Σχήμα 3-22. Οι χρόνοι μετάδοσης του *GetParameterValues* μηνύματος απάντησης.

Αντίθετα οι χρόνοι του σχήματος 3-22 εξαρτώνται μόνο από τον αριθμό των παραμέτρων και το διαθέσιμο εύρος ζώνης. Επίσης πρέπει να τονιστεί ότι οι ρυθμοί μετάδοσης του σχήματος 3-21 ανταποκρίνονται στο διαθέσιμο εύρος ζώνης κατά τη μετάδοση δεδομένων από το διαχειριστή προς τον πράκτορα (download), ενώ εκείνοι του σχήματος 3-22 συνδέονται με τη μετάδοση δεδομένων από τον πράκτορα προς το διαχειριστή (upload). Έτσι ακόμα και για ασύμμετρες τεχνολογίες πρόσβασης, όπως είναι εκείνη του ADSL, μπορεί κανείς να εκτιμήσει το συνολικό χρόνο μετάδοσης συνδυάζοντας τα δύο αυτά σχήματα. Στο σημείο αυτό πρέπει να τονιστεί ότι επειδή οι συνδρομητικές συσκευές μοιράζονται μεταξύ τους τη σύνδεση στο διαδίκτυο, το διαθέσιμο εύρος ζώνης για τη διαχείρισή τους μπορεί πολλές

φορές να είναι αρκετά χαμηλότερο από εκείνο που υποστηρίζει η τεχνολογία πρόσβασης. Επομένως για τις δημοφιλέστερες τεχνολογίες πρόσβασης (π.χ. xDSL, cable TV) οι τιμές των ρυθμών μετάδοσης, που εμφανίζονται στα σχήματα 3-21 και 3-22 ανταποκρίνονται στο σήμερα αλλά και στο εγγύς μέλλον.

Αυτό που διαφαίνεται είναι ότι στις περιπτώσεις όπου το διαθέσιμο εύρος ζώνης είναι σχετικά μικρό (π.χ. 128kbps και 256kbps) οι χρόνοι μετάδοσης των μηνυμάτων διαχείρισης μπορούν από μόνοι τους να ξεπεράσουν τα αποδεκτά όρια του χρόνου αναμονής των χρηστών (δηλαδή της εμπειρικής τιμής των 1-2 δευτερολέπτων) για την ενεργοποίηση μιας υπηρεσίας ή για την τροποποίησή της. Ωστόσο ακόμα και αν οι χρήστες είναι πιο ελαστικοί με τη ρύθμιση των υπηρεσιών τους το πρόβλημα δεν σταματά εδώ. Όσο μεγαλύτερος είναι ο χρόνος διαχείρισης ανά συσκευή τόσο μειώνεται ο ρυθμός εξυπηρέτησης του ACS διαχειριστή. Όπως προβλέπεται από την CWMP αρχιτεκτονική για λόγους κλιμακωσιμότητας η εφαρμογή του ACS διαχειριστή μπορεί να φιλοξενείται σε πολλά διαφορετικά μηχανήματα προκειμένου να επιτρέπει την αποδοτική εκτέλεση της εφαρμογής για μεγάλους αριθμούς διαχειριζόμενων συσκευών. Μάλιστα προβλέπεται ένας ισοσταθμιστής φορτίου (load balancer) ο οποίος θα διαμοιράζει ισομερώς στους ACS διαχειριστές τις αιτήσεις των συσκευών για ρύθμιση. Αν υποτεθεί ότι κάθε σύστημα, που φιλοξενεί την εφαρμογή του ACS διαχειριστή μπορεί να εξυπηρετεί αποδεκτά μέχρι και n παράλληλες διαχειριστικές συνεδρίες (με n παράλληλες διαδικασίες που η καθεμία ανοίγει μια ξεχωριστή σύνδεση με την κάθε συσκευή) και ο μέσος χρόνος διαχείρισης είναι $t_{ολ}$ τότε ο μέσος ρυθμός εξυπηρέτησης του κάθε εξυπηρετητή είναι $\mu = n \times (1/ t_{ολ})$. Όσο μεγαλύτερος λοιπόν είναι ο χρόνος αυτός τόσο ελαττώνεται ο ρυθμός εξυπηρέτησης του κάθε διαχειριστή με αποτέλεσμα να χρειάζονται επιπρόσθετα ή πιο ακριβά μηχανήματα για να υποστηρίξουν ένα δεδομένο αριθμό συσκευών. Κάτι τέτοιο οδηγεί σε περαιτέρω αύξηση του κόστους διαχείρισης και επομένως είναι σκόπιμη η κάθε προσπάθεια περιορισμού του χρόνου αυτού. Τα πειραματικά αποτελέσματα των προηγούμενων εργασιών [5,6] δείχνουν ότι στις περισσότερες περιπτώσεις οι χρόνοι κωδικοποίησης και ενορχήστρωσης είναι της τάξης ορισμένων δεκάδων milliseconds. Οι τιμές λοιπόν των σχημάτων 3-21 και 3-22 οδηγούν στο συμπέρασμα ότι όταν το διαθέσιμο εύρος ζώνης είναι περιορισμένο, ο χρόνος μετάδοσης των μηνυμάτων διαχείρισης είναι τουλάχιστον μια τάξη μεγέθους μεγαλύτερος από αυτούς, που συνιστούν την υπόλοιπη διαχειριστική διαδικασία. Επομένως η προσπάθεια μείωσης του χρόνου μετάδοσης αποκτά ιδιαίτερη σημασία διότι, λόγω της δαπανηρής SOAP/XML κωδικοποίησης η μετάδοση μπορεί πολλές φορές να αποτελέσει τον καθοριστικότερο παράγοντα στο συνολικό χρόνο διαχείρισης.

3.5.4. Η ιδέα της επιλεκτικής συμπίεσης

Για όλους τους λόγους που προαναφέρθηκαν καθίσταται αναγκαία η δραστική μείωση του όγκου των μηνυμάτων διαχείρισης. Μάλιστα υπάρχουν δύο κύριες στρατηγικές για την επίτευξη του στόχου αυτού . Η πρώτη συνίσταται στην πρόταση ενός νέου σχήματος κωδικοποίησης το οποίο θα εξειδικεύεται στη διαχείριση και θα είναι πολύ λιγότερο δαπανηρό σε εύρος ζώνης από το SOAP/XML. Αντίθετα η δεύτερη στρατηγική συνίσταται

στη διατήρηση του SOAP/XML σαν τρόπου κωδικοποίησης και στη συμπίεση των SOAP μηνυμάτων προκειμένου να μειωθεί ο όγκος τους. Η πρώτη στρατηγική έχει το πλεονέκτημα ότι καταπολεμά το πρόβλημα στη ρίζα του αλλά μειονεκτεί στο γεγονός ότι η SOAP/XML κωδικοποίηση είναι ήδη πολύ διαδεδομένη όχι μόνο στην περιοχή των κατανεμημένων εφαρμογών αλλά και στην ίδια την περιοχή της διαχείρισης. Αρκεί να σκεφτεί κανείς ότι τα τελευταία επτά χρόνια το SOAP πρότυπο χρησιμοποιείται με επιτυχία τόσο από το ευρέως διαδεδομένο πρωτόκολλο τοπικής διαχείρισης UPnP [45] όσο και από τη NetConf προσπάθεια προτυποποίησης της IETF αλλά και πρόσφατα από το CWMP πρότυπο του DSL φόρουμ. Επειδή στόχος της παρούσας εργασίας ήταν να προτείνει μια λύση, που θα είχε άμεση εφαρμογή στα υπάρχοντα και αναδυόμενα πρότυπα επιλέχτηκε η διερεύνηση της δεύτερης στρατηγικής η οποία δεν επιβάλλει καμία αλλαγή στον τρόπο μοντελοποίησης των συσκευών και κωδικοποίησης της αντίστοιχης διαχειριστικής πληροφορίας παρά μόνο στον τρόπο μεταφοράς αυτής.

Βέβαια η ιδέα της συμπίεσης των μηνυμάτων, που ανταλλάσσονται μέσω δικτύων με περιορισμένο εύρος ζώνης δεν είναι καινούργια. Ήδη το HTTP 1.1 πρότυπο [47] προβλέπει τη συμπίεση του περιεχομένου των πακέτων, που μεταφέρονται από του εξυπηρετητές ιστού στους HTTP πελάτες. Με αυτό τον τρόπο αρχεία κειμένου (XHTML, Javascript, CSS κ.α) συμπιέζονται στον εξυπηρετητή και αποσυμπιέζονται αυτόματα στον HTTP φυλλομετρητή, υπό την προϋπόθεση ότι αυτός υποστηρίζει το HTTP 1.1 πρότυπο. Επιπλέον ούτε η ιδέα της επιλεκτικής συμπίεσης είναι καινούργια. Εργασίες όπως οι [48,49] προτείνουν την εφαρμογή της αυτόματα ρυθμιζόμενης συμπίεσης στις διαδραστικές και συνεργατικές κατανεμημένες εφαρμογές όπως είναι εκείνες της απομακρυσμένης συνεργασίας επιστημόνων και μηχανικών [50], της απομακρυσμένης οπτικοποίησης και των γραφικών καθώς και οι υψηλού κλίμακας εμπορικές εφαρμογές όπως είναι τα λειτουργικά πληροφοριακά συστήματα [51]. Οι εργασίες αυτές εστιάζουν κυρίως στη διερεύνηση των διαθέσιμων αλγορίθμων συμπίεσης και της καταλληλότητας του καθενός για τα δεδομένα των παραπάνω κατανεμημένων εφαρμογών. Μάλιστα για το κάθε άκρο επικοινωνίας προτείνουν την ενσωμάτωση μεσισμικού το οποίο θα παρακολουθεί συνεχώς το διαθέσιμο εύρος ζώνης και το φορτίο επεξεργασίας σε κάθε άκρο. Με βάση την πληροφορία αυτή αλλά και τη φύση των δεδομένων, που παράγει η εφαρμογή το προτεινόμενο μεσισμικό εφαρμόζει κάθε φορά ένα διαφορετικό αλγόριθμο συμπίεσης με στόχο τη βελτιστοποίηση του χρόνου μεταφοράς των δεδομένων της εφαρμογής.

Επίσης η συμπίεση των μηνυμάτων προτάθηκε και σε ορισμένες από τις προαναφερθείσες σχετικές με την XML διαχείριση εργασίες [5,6,7]. Η εργασία [6] προτείνει τη συμπίεση χωρίς να την αποτιμήσει. Μάλιστα αποφαίνεται ότι η συμπίεση των SOAP μηνυμάτων μπορεί να επιφέρει την ελάττωση του όγκου της κίνησης διαχείρισης έχοντας ωστόσο σαν τίμημα την αύξηση του διαχειριστικού χρόνου. Αντίθετα η εργασία [5] πραγματοποίησε πειράματα και αποτίμησε τη διαδικασία της συμπίεσης. Το συμπέρασμα που προέκυψε ήταν ότι ο όγκος των SOAP μηνυμάτων μπορούσε να μειωθεί μέχρι και οκτώ φορές (στην περίπτωση ανάκτησης 250 μεταβλητών). Ωστόσο παρόλο που χρησιμοποιήθηκε ο αλγόριθμος Lempel-Ziv η συμπίεση αποδείχτηκε πολύ δαπανηρή σε σχέση με τη SOAP κωδικοποίηση των μηνυμάτων

και το γεγονός αυτό οδήγησε διαισθητικά τους συγγραφείς στην εκτίμηση ότι η συμπίεση ενδεχομένως να έχει κερδοφόρα αποτελέσματα μόνο σε περιπτώσεις όπου το διαθέσιμο εύρος ζώνης είναι πολύ μικρό.

Προκείμενου λοιπόν να δοθεί μια πιο ξεκάθαρη απάντηση σχετικά με την καταλληλότητα της συμπίεσης στη διαχείριση η παρούσα εργασία διερευνά τους βέλτιστους αλγορίθμους συμπίεσης, τις επιδόσεις της συμπίεσης στο πρότυπο CWMP καθώς και τις συνθήκες εκείνες κάτω από τις οποίες η συμπίεση είναι κερδοφόρα για τη διαχείριση. Στη συνέχεια προτείνεται η λύση της επιλεκτικής συμπίεσης, που θα βασίζεται στις προαναφερθείσες συνθήκες. Μάλιστα θα αποδειχθεί πειραματικά ότι στη διαχείριση η επιλεκτική συμπίεση μπορεί να εφαρμοστεί με πολύ απλούστερο τρόπο από εκείνον, που προτείνεται στην περίπτωση των κατανεμημένων εφαρμογών [48,49].

3.5.5. Η επιλογή της μεθόδου συμπίεσης

Οι μέθοδοι συμπίεσης μειώνουν το μέγεθος των δεδομένων με την εφαρμογή διαφόρων τεχνικών συμπίεσης και αποσυμπίεσης. Η ενότητα αυτή κάνει μια σύντομη ανασκόπηση των βασικότερων αντιστρεπτών (lossless) αλγορίθμων συμπίεσης, που έχουν προταθεί στο παρελθόν. Στόχος είναι η επιλογή της βέλτιστης μεθόδου αλλά και της βέλτιστης υλοποίησής της με κριτήρια την ταχύτητα και το λόγο συμπίεσης. Στο σημείο αυτό πρέπει να τονιστεί ότι η πλήρης ανάλυση των μεθόδων συμπίεσης και των επιδόσεών τους δεν αποτελεί αντικειμενικό σκοπό της παρούσας εργασίας και για αυτό το λόγο η ανάλυση για την επιλογή του βέλτιστου αλγορίθμου στηρίζεται σε αποτελέσματα προηγούμενων εργασιών [48].

3.5.5.1. Η κωδικοποίηση Huffman και η αριθμητική κωδικοποίηση

Η κωδικοποίηση Huffman [52] είναι η πρώτη πρακτική μέθοδος συμπίεσης και χρησιμοποιείται είτε αυτόνομα, είτε σε πιο πολύπλοκες τεχνικές συμπίεσης όπως είναι εκείνη του JPEG [53,54].Η βασική της ιδέα είναι η αντιστοίχιση μικρών κωδικών σε συχνά εμφανιζόμενα σύμβολα και μεγάλων κωδικών σε σπάνια εμφανιζόμενα σύμβολα. Τα κύρια πλεονεκτήματα των κωδικών Huffman είναι η απλότητα και η ταχύτητά τους. Οι κώδικες αυτοί δουλεύουν καλά για δυαδικά δεδομένα όπου η επαναληψιμότητα των συμβολοσειρών είναι περιορισμένη. Η κωδικοποίηση Huffman υποθέτει ότι δεν υπάρχει καμία συσχέτιση μεταξύ γειτονικών χαρακτήρων. Για αυτό το λόγο δεν έχει καλές επιδόσεις σε κείμενα χαρακτήρων. Η πολυπλοκότητα της κωδικοποίησης αυτής είναι O(m+nlogn) όπου m είναι το μέγεθος του κειμένου και n είναι το μέγεθος του αλφάβητου.

Ενώ η μέθοδος της κωδικοποίησης Huffman χρησιμοποιεί μια σειρά δυαδικών ψηφίων για κάθε σύμβολο στο αρχικό αρχείο, η αριθμητική κωδικοποίηση [55,56] βελτιώνει την παραπάνω με τη χρήση κλασμάτων από δυαδικά ψηφία. Με αυτόν τον τρόπο ένα δυαδικό ψηφίο μπορεί να ανήκει σε περισσότερα του ενός σύμβολα. Συνεπώς ένας κωδικός μπορεί να αντιπροσωπεύεται από ένα ρητό αριθμό δυαδικών ψηφίων. (π.χ. 3,25 bits).

3.5.5.2. Η Lempel–Ziv κωδικοποίηση

Πρόκειται για την κλασσική μέθοδο λεξικής συμπίεσης [57, 58]. Εργαλεία συμπίεσης όπως το WinZip [59] και gZip [60] χρησιμοποιούν την εν λόγω κωδικοποίηση. Όπως προαναφέρθηκε η κωδικοποίηση Huffman και η αριθμητική κωδικοποίηση δεν λαμβάνουν υπόψη τους το περιβάλλον των συμβόλων προς κωδικοποίηση. Αντίθετα η Lempel-Ziv κωδικοποίηση δίνει μεγάλη σημασία στις προηγούμενες εμφανίσεις των συμβολοσειρών. Συγκεκριμένα η μέθοδος αυτή τοποθετεί ένα δείκτη στο μέρος κάθε προβεβλημένης συμβολοσειράς. Κάθε τέτοιος δείκτης είναι της μορφής (x,y) και παριστάνει μια συμβολοσειρά η οποία είναι η ίδια με εκείνη, που βρίσκεται x bytes πίσω και έχει μήκος y χαρακτήρες.

3.5.5.3. Ο μετασχηματισμός Burrows–Wheeler

Ο μετασχηματισμός Burrows-Wheeler [61, 62] είναι επίσης μια λεξικογραφική μέθοδος συμπίεσης. Η μέθοδος αυτή εκμεταλλεύεται τις επαναλήψεις ολόκληρων λέξεων προκειμένου να βελτιώσει τη συμπίεση. Η μέθοδος αυτή παρουσιάζει καλύτερους λόγους συμπίεσης από τη Lempel-Ziv κωδικοποίηση αλλά οι χρόνοι εκτέλεσής της μπορεί να είναι αρκετά υψηλοί. Η μέθοδος αποτελείται από πολλαπλά βήματα. Το πρώτο βήμα δημιουργεί δείκτες σε όλους τους χαρακτήρες του προς συμπίεση αρχείου. Οι δείκτες ταξινομούνται σύμφωνα με τους χαρακτήρες στους οποίους δείχνουν. Οι χαρακτήρες κάθε δείκτη στέλνονται στο επόμενο βήμα με βάση την προαναφερθείσα ταξινόμηση. Στην πραγματικότητα αυτή η σειρά των χαρακτήρων στην έξοδο έχει τους ίδιους χαρακτήρες με το αρχικό αρχείο με τη μόνη διαφορά ότι η σειρά τώρα είναι διαφορετική. Το δεύτερο βήμα εκτελεί ένα αλγόριθμο «κίνησης προς τα εμπρός». Ο αλγόριθμος αυτός κρατά και τους 256 πιθανούς χαρακτήρες σε μια λίστα. Όταν ένας χαρακτήρας στέλνεται στο επόμενο βήμα, στέλνεται η θέση του στη λίστα. Αφού ο χαρακτήρας σταλεί μετακινείται από την τρέχουσα θέση του στην αρχή της λίστας (δηλαδή στη θέση 0). Το επόμενο βήμα εφαρμόζει μια run-length κωδικοποίηση στη έξοδο του προηγούμενου βήματος. Η έξοδος της run-length κωδικοποίησης συμπιέζεται είτε με την κωδικοποίηση Huffman, είτε με την αριθμητική κωδικοποίηση. Το κύριο μειονέκτημα της Burrows-Wheeler κωδικοποίησης είναι οι υψηλοί χρόνοι εκτέλεσης εξαιτίας της ανάγκης ταξινόμησης του αρχείου. Εξάλλου η πολυπλοκότητα αυτού του αλγορίθμου συμπίεσης βρίσκεται σε υψηλά επίπεδα και δίνεται από τον τύπο O(mlogm), όπου m είναι το μέγεθος του κειμένου που συμπιέζεται.

3.5.5.4. Σύγκριση των μεθόδων συμπίεσης

Η εργασία [61,62] πραγματοποιεί εκτεταμένα πειράματα για την αποτίμηση των παραπάνω μεθόδων συμπίεσης με κριτήρια την ταχύτητα και το λόγο συμπίεσης. Αφού εξετάζονται διάφορα είδη δεδομένων (συμπεριλαμβανομένου και των XML αρχείων) και συγκρίνονται τα αντίστοιχα πειραματικά αποτελέσματα, οι συγγραφείς δίνουν ποιοτικούς χαρακτηρισμούς για τις επιδόσεις κάθε αλγορίθμου ανάλογα με τη φύση των αρχείων που συμπιέζονται. Ό πίνακας 3.8 προκύπτει από τη σταχυολόγηση των αποτελεσμάτων και δίνει τους ακόλουθους χαρακτηρισμούς: «Πολύ Καλά», «Καλά», «Μη Ικανοποιητικά».

Πίνακας 3.8. Η ποιοτική αποτίμηση των μεθόδων συμπίεσης.

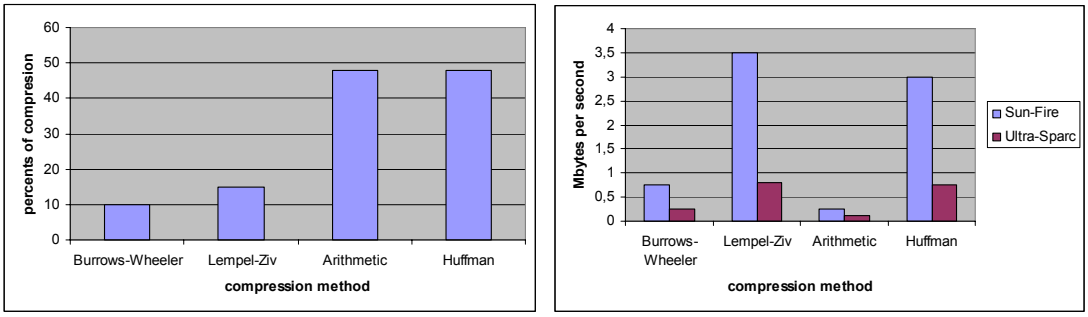
	Burrows-Wheeler	Lempel-Ziv	Arithmetic	Huffman
Λόγοι συμπίεσης αρχείων με επαναλήψεις συμβολοσειρών	Πολύ Καλά	Πολύ Καλά	Μη Ικανοποιητικά	Μη Ικανοποιητικά
Λόγοι συμπίεσης αρχείων με χαμηλή εντροπία	Πολύ Καλά	Μη Ικανοποιητικά	Πολύ Καλά	Πολύ Καλά
Ταχύτητα συμπίεσης	Μη Ικανοποιητικά	Καλά	Μη Ικανοποιητικά	Πολύ Καλά
Ταχύτητα αποσυμπίεσης	Καλά	Πολύ Καλά	Μη Ικανοποιητικά	Πολύ Καλά

Από την πρώτη γραμμή αποτελεσμάτων του πίνακα εξάγεται καθαρά το συμπέρασμα ότι οι καταλληλότεροι αλγόριθμοι για τους SOAP φακέλους, που μεταφέρουν τη διαχειριστική πληροφορία είναι ο Burrows-Wheeler και ο Lempel-Ziv, όταν το κριτήριο είναι η ικανότητα συμπίεσης. Επιπλέον οι δύο τελευταίες γραμμές υποδεικνύουν ότι ο Lempel-Ziv αλγόριθμος υπερτερεί έναντι του Burrows-Wheeler μετασχηματισμού τόσο στην ταχύτητα συμπίεσης όσο και στην ταχύτητα αποσυμπίεσης. Επομένως με βάση και μόνο τον πίνακα ποιοτικής αποτίμησης προκύπτει ότι η καταλληλότερη μέθοδος συμπίεσης είναι η Lempel-Ziv.

Ωστόσο για να επικυρωθεί η επιλογή αυτή χρησιμοποιούνται τα πειραματικά αποτελέσματα της εργασίας [48] σε συνδυασμό με τη σχέση 3-10 η οποία δίνει το κέρδος χρόνου, που αποφέρει η συμπίεση. Αν μάλιστα χρησιμοποιηθούν και οι σχέσεις 3-4, 3-6 και 3-7 τότε η σχέση 3-10 αποκτά την ακόλουθη μορφή:

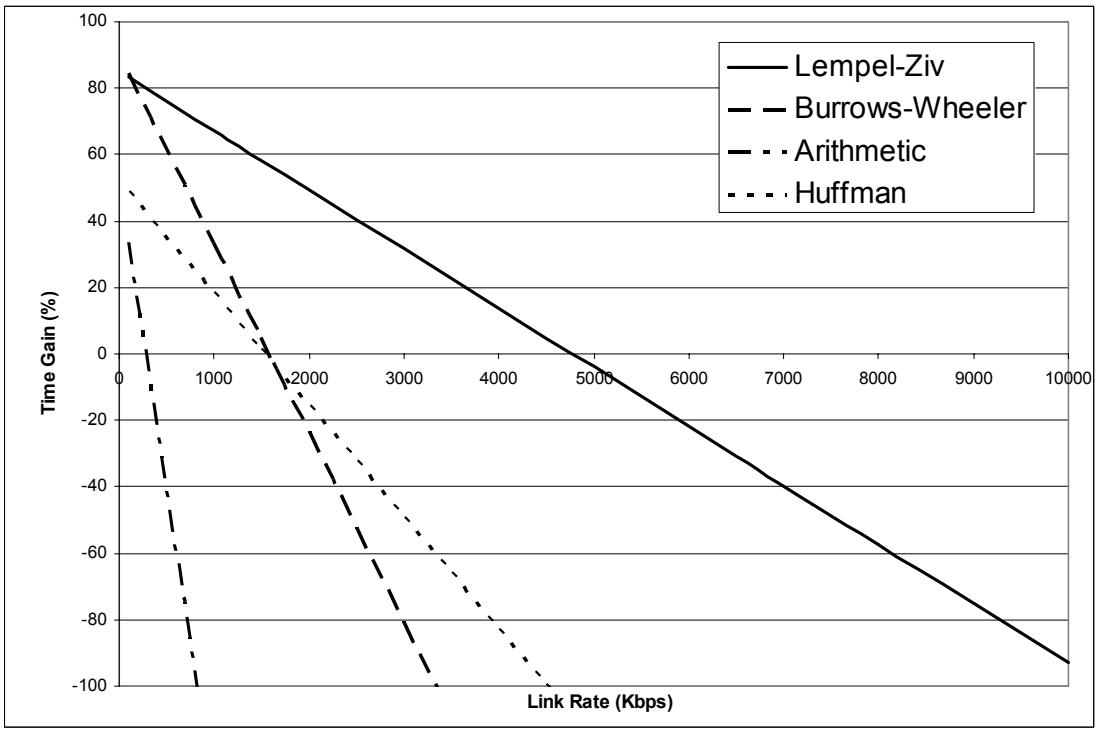
$Gt_x = S_x\,[(1 - C_{ratio}) / LinkRate - 1/U_{c(y)} - 1/U_{D(z)}]$, x: μετάδοση μηνύματος αίτησης (ΜΜΑ) ή απόκρισης (ΜΜΑΠ) και y, z: Διαχειριστής (Δ) ή Πράκτορας (Π) (3-12)

Για τη διαχείριση η βέλτιστη μέθοδος συμπίεσης είναι εκείνη που μεγιστοποιεί το χρονικό κέρδος κατά τη μετάδοση (Gt_x) των μηνυμάτων διαχείρισης. Από τη σχέση 3-12 προκύπτει ότι όσο μεγαλύτερο είναι το διαθέσιμο εύρος ζώνης σε σχέση με τις ταχύτητες συμπίεσης και αποσυμπίεσης τόσο οι ταχύτητες αυτές αποκτούν μεγαλύτερη σημασία. Αντίθετα για πολύ μικρές ταχύτητες μετάδοσης ο κυρίαρχος ρόλος ανήκει στο λόγο συμπίεσης. Αν ληφθεί υπόψη ότι οι τεχνολογίες πρόσβασης συνεχώς βελτιώνονται και επομένως η διαχείριση θα λαμβάνει χώρα σε περιβάλλοντα με ολοένα και αυξανόμενους ρυθμούς μετάδοσης, τότε η συμπίεση των διαχειριστικών δεδομένων θα πρέπει να ανταποκρίνεται σε όσο το δυνατό μεγαλύτερες ταχύτητες μετάδοσης. Συνεπώς οι ταχύτητες εκτέλεσης της μεθόδου συμπίεσης αποκτούν τον κυρίαρχο ρόλο έναντι των λόγων συμπίεσης. Προκειμένου να αποκτηθεί μια πληρέστερη εικόνα για όλα τα παραπάνω χρησιμοποιείται η σχέση 3-12 σε συνδυασμό με τα πειραματικά αποτελέσματα της εργασίας [48], που δίνουν τις ταχύτητες συμπίεσης και αποσυμπίεσης για συγκεκριμένες υλοποιήσεις [55, 63-65] και πλατφόρμες εκτέλεσης (Ultra-Sparc-II με λειτουργικό Solaris 5.8 και 128MB RAM). Επίσης δίνονται οι λόγοι συμπίεσης για μεγάλο όγκο δεδομένων με υψηλή επαναληψιμότητα συμβολοσειρών. Όλα τα παραπάνω πειραματικά αποτελέσματα απεικονίζονται στο σχήμα 3-23.



Σχήμα 3-23. Τα πειραματικά αποτελέσματα της εργασίας [48].

Το σχήμα 3-24 απεικονίζει το ποσοστιαίο (δηλαδή το $Gt_x/(S_x/LinkRate)$) χρονικό κέρδος της σχέσης 3-12 σε συνάρτηση με το διαθέσιμο εύρος ζώνης. Καταρχάς επαληθεύεται ότι η Lempel-Ziv μέθοδος είναι η καταλληλότερη, καθώς συνδυάζει τους πολύ καλούς λόγους συμπίεσης με τις υψηλές ταχύτητες εκτέλεσης. Το γεγονός αυτό έχει σαν αποτέλεσμα την κερδοφόρα συμπίεση για ένα μεγάλο εύρος τιμών του ρυθμού μετάδοσης (0-4,8Mbps).



Σχήμα 3-24. Τα χρονικά κέρδη (Gt_x) των μεθόδων συμπίεσης.

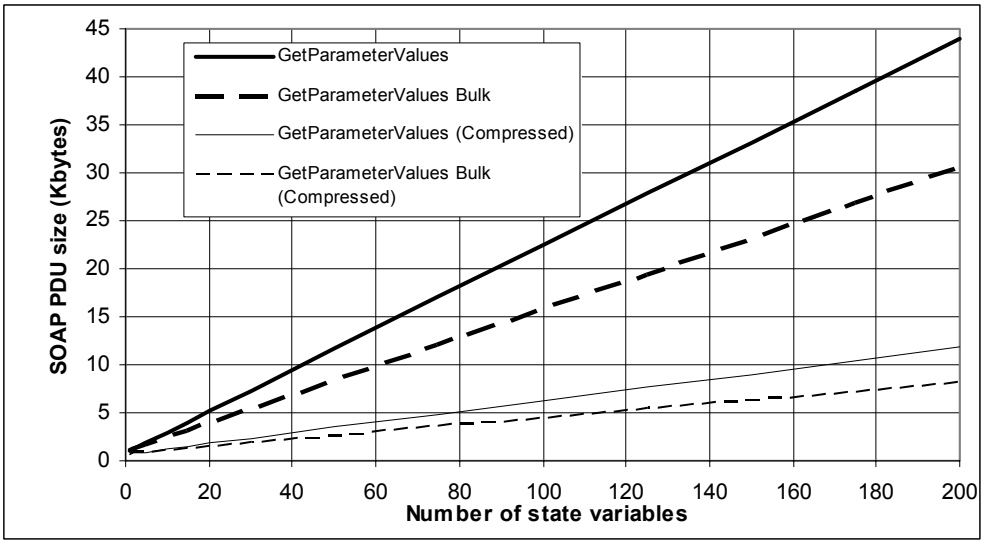
Επίσης για πολύ μικρούς ρυθμούς μετάδοσης η μέθοδος Burrows-Wheeler (που έχει τον καλύτερο λόγο συμπίεσης) βρίσκεται πολύ κοντά στη Lempel-Ziv μέθοδο, ενώ σε ορισμένες περιπτώσεις υπερτερεί αυτής (για ρυθμούς χαμηλότερους των 100Kbps). Αντίθετα όσο αυξάνουν οι ρυθμοί μετάδοσης τόσο αποκλίνουν οι δύο αυτές μέθοδοι. Μάλιστα μετά το 1,5Mbps ο αλγόριθμος Huffman (με χειρότερο λόγο συμπίεσης) υπερτερεί της μεθόδου Burrows-Wheeler επειδή υποστηρίζει πολύ μεγαλύτερες ταχύτητες εκτέλεσης. Όσον αφορά στην αριθμητική μέθοδο αυτή φαίνεται ότι είναι ακατάλληλη για συμπίεση σε πραγματικό

χρόνο αφού το αντίστοιχο κέρδος συμπίεσης εξανεμίζεται πολύ απότομα με την αύξηση του εύρους ζώνης.

Η παραπάνω ανάλυση έθεσε τα κριτήρια επιλογής της μεθόδου συμπίεσης και της αντίστοιχης υλοποίησής της. Με βάση τα κριτήρια αυτά η παρούσα εργασία πειραματίστηκε με διάφορες υλοποιήσεις του Lempel-Ziv αλγορίθμου και βρήκε ότι η βιβλιοθήκη *LZO* [66] παρουσιάζει εξαιρετικές ταχύτητες εκτέλεσης. Χαρακτηριστικό είναι το γεγονός ότι η LZO πετυχαίνει μέχρι και πενταπλάσιες ταχύτητες συμπίεσης από εκείνες της γνωστής *gzip* υλοποίησης [60]. Μάλιστα η LZO βιβλιοθήκη παρέχει πολλές υλοποιήσεις, που μεταξύ τους ανταλλάσσουν την αποτελεσματικότητα με την ταχύτητα της συμπίεσης. Ακόμη και μεταξύ αυτών επιλέχτηκε η γρηγορότερη υλοποίηση (*LZO1X-1* για τη συμπίεση και *LZO1X* για την αποσυμπίεση) προκειμένου, όπως αποδείχτηκε παραπάνω, η συμπίεση να είναι εφαρμόσιμή σε όσο το δυνατό μεγαλύτερους ρυθμούς μετάδοσης και επομένως να καλύπτει περισσότερα συνδρομητικά δίκτυα. Παρακάτω εφαρμόζεται η υλοποίηση αυτή προκειμένου να αποτιμηθούν τα πραγματικά της οφέλη στη διαχείριση.

3.5.6. Το κέρδος της συμπίεσης στη διαχειριστική κίνηση

Το επόμενο βήμα μετά την επιλογή της μεθόδου συμπίεσης ήταν η αποτίμηση της επίδρασής της στη CWMP διαχείριση. Για το λόγο αυτό στα *GetParameterValues* μηνύματα αίτησης και απάντησης (παράγραφος 3.5.2.2) εφαρμόστηκε η *LZO1X-1* συμπίεση και μετρήθηκε το νέο μέγεθος των αντίστοιχων SOAP φακέλων. Τα αποτελέσματα απεικονίζονται στο σχήμα 3-25. Οι δύο καμπύλες με το έντονο χρώμα εκφράζουν το μέγεθος των αρχικών μηνυμάτων (σχήμα 3-20) ενώ οι άλλες δύο αντιπροσωπεύουν το μέγεθος των αντίστοιχων συμπιεσμένων μηνυμάτων. Καταρχάς οι κλίσεις των καμπυλών συμπίεσης φανερώνουν ότι όσο μεγαλύτερο είναι το μέγεθος των SOAP μηνυμάτων τόσο αυξάνει η αποδοτικότητα της συμπίεσης. Ανάλογα με το μέγεθός τους τα ασυμπίεστα μηνύματα είναι από 1,5 (C_{ratio}=0,64) έως 3,7 (C_{ratio}=0,26) φορές μεγαλύτερα από τα αντίστοιχα συμπιεσμένα. Η συμπεριφορά αυτή είναι αναμενόμενη καθώς η αποδοτικότητα της Lempel-Ziv μεθόδου είναι ευθέως ανάλογη της επαναληψιμότητας των συμβολοσειρών, η οποία με τη σειρά της αυξάνει ανάλογα με το μέγεθος του μηνύματος. Στο σημείο αυτό θα πρέπει να τονιστεί ότι η υλοποίηση *LZO1X-1*, που χρησιμοποιήθηκε στην παρούσα εργασία φαίνεται να είναι λιγότερο αποδοτική από εκείνες των προηγούμενων εργασιών [48][5], όπου τα ασυμπίεστα μηνύματα φτάνουν να είναι μέχρι και 6,7 φορές (C_{ratio}=0,15) μεγαλύτερα των αντίστοιχων συμπιεσμένων. Το γεγονός αυτό είναι επίσης αναμενόμενο, καθώς η χρησιμοποιούμενη υλοποίηση εστιάζει κυρίως στον πραγματικό χρόνο με αποτέλεσμα να πληρώνει το τίμημα στο λόγο συμπίεσης. Ωστόσο οι λόγοι συμπίεσης κρίνονται ικανοποιητικοί καθώς οι μέγιστες τιμές της αποδοτικότητας επιτυγχάνονται από πολύ νωρίς στον άξονα του αριθμού των μεταβλητών κατάστασης. Ήδη για τις 20 πρώτες μεταβλητές τα ασυμπίεστα μηνύματα είναι 3 φορές μεγαλύτερα από εκείνα των συμπιεσμένων. Το γεγονός αυτό σημαίνει ότι όχι μόνο για τα μεγάλα μηνύματα αλλά και για το μεγαλύτερο πλήθος των μηνυμάτων αναμένονται σημαντικές βελτιώσεις.



Σχήμα 3-25. Η επίδραση της συμπίεσης στον όγκο των SOAP μηνυμάτων κατά την παρακολούθηση.

3.5.7. Το κέρδος της συμπίεσης στο χρόνο μετάδοσης

Στην προηγούμενη ενότητα φάνηκαν τα δραστικά αποτελέσματα της συμπίεσης στο μέγεθος των διαχειριστικών δεδομένων. Το αμέσως επόμενο ζήτημα προς διερεύνηση ήταν η επίδραση της συμπίεσης στους χρόνους μετάδοσης των SOAP μηνυμάτων αλλά και στο συνολικό χρόνο διαχείρισης. Η παρούσα παράγραφος επιχειρεί να δώσει μια εικόνα σχετικά με τα αποτελέσματα της συμπίεσης στους χρόνους μετάδοσης των παραπάνω μηνυμάτων. Για το λόγο αυτό χρειάστηκε να μετρηθούν οι χρόνοι συμπίεσης και αποσυμπίεσης των SOAP μηνυμάτων, ώστε στην αποτίμηση της χρονικής βελτίωσης να συνυπολογισθεί και ο συνολικός χρονικός υπερκεφαλισμός της συμπίεσης.

Πέραν της υλοποίησης της μεθόδου συμπίεσης, οι χρόνοι συμπίεσης και αποσυμπίεσης συνδέονται άμεσα με την πλατφόρμα εκτέλεσης. Σήμερα ήδη ένα μεγάλο μέρος του συνδρομητικού εξοπλισμού (π.χ. υπολογιστές, παιχνιδομηχανές, συνδρομητικοί αποκωδικοποιητές) είναι εξοπλισμένο με επεξεργαστές πολύ ταχύτερους από ένα Pentium I επεξεργαστή. Είναι χαρακτηριστικό το γεγονός ότι τα τελευταία 3 χρόνια μια από τις πιο διαδεδομένες παιχνιδομηχανές (Xbox) [67] για διαδικτυακά παιχνίδια υποστηρίζει ένα Pentium III (733MHz) επεξεργαστή. Μάλιστα η νέα έκδοση (PS3) [68] της ακόμα πιο διαδεδομένης μηχανής PlayStation υποστηρίζει ένα PowerPC (3,2 GHz) επεξεργαστή. Παρόμοια είναι η κατάσταση και στους συνδρομητικούς αποκωδικοποιητές (set-top-boxes) πολλοί εκ των οποίων είναι εξοπλισμένοι με υπερσύγχρονους επεξεργαστές αρκετών εκατοντάδων MIPS Όλα τα παραπάνω είναι ενδεικτικά της τάσης που επικρατεί για ενσωμάτωση ολοένα και ισχυρότερων επεξεργαστών στις συνδρομητικές συσκευές..

Στα πλαίσια της παρούσας εργασίας τα πειράματα των μετρήσεων έγιναν σε τρεις διαφορετικές πλατφόρμες. Η πρώτη πλατφόρμα (ΠΛΑΤΦ1) είναι ένα PC με επεξεργαστή Pentium I (133MHz) και μνήμη RAM 64Mbytes. Η δεύτερη πλατφόρμα (ΠΛΑΤΦ2) είναι μια κάρτα MiniITX [69] με επεξεργαστή *Via EDEN ESP Samuel* (600MHz) [74] και μνήμη

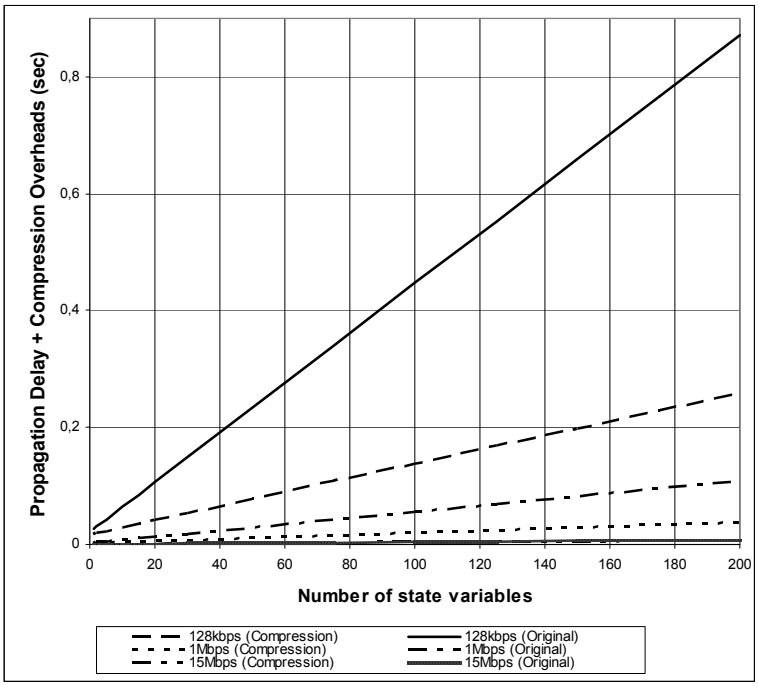
RAM 128 Mbytes. Η τρίτη πλατφόρμα (ΠΛΑΤΦ3) είναι ένα PC με επεξεργαστή Pentium III (1,6MHz) και μνήμη RAM 256 Mbytes. Όλες οι πλατφόρμες υποστηρίζονται από το λειτουργικό σύστημα Linux 2.4.22 [70]. Αφού και στις τρεις πλατφόρμες μετρήθηκε ένα πλήθος διαφορετικών μηνυμάτων και δεν παρατηρήθηκε κάποια αξιοσημείωτη διασπορά στις ταχύτητες εκτέλεσης (μεταξύ των μικρών και μεγάλων μηνυμάτων), στη συνέχεια υπολογίστηκαν οι μέσες ταχύτητες συμπίεσης και αποσυμπίεσης οι οποίες και δίνονται στον πίνακα 3.9.

Πίνακας 3.9. Οι ταχύτητες εκτέλεσης της Lempel-Ziv (LZO1X) μεθόδου συμπίεσης.

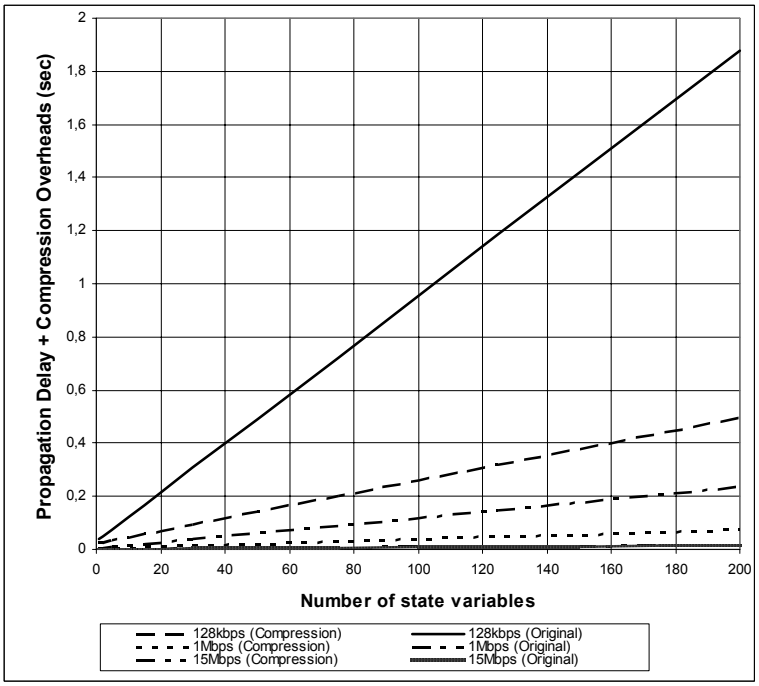
Πλατφόρμα	Ταχύτητα Συμπίεσης (Mbytes/sec)	Ταχύτητα Αποσυμπίεσης (Mbytes/sec)
ΠΛΑΤΦ1	4,5	7,2
ΠΛΑΤΦ2	10,9	18
ΠΛΑΤΦ3	15,4	23,5

Για τη διερεύνηση της επίδρασης της συμπίεσης στους χρόνους μετάδοσης των μηνυμάτων διαχείρισης επιλέχτηκε η πρώτη πλατφόρμα, καθώς είναι εκείνη με τις χαμηλότερες δυνατότητες και επομένως η ανάλυση θα κάλυπτε μια ευρύτερη γκάμα συνδρομητικών συσκευών πέραν αυτών που προαναφέρθηκαν. Μάλιστα, παρόλο που ο ACS διαχειριστής αναμένεται πολύ πιο ισχυρός και από τις τρεις πλατφόρμες του πίνακα 3.9, ωστόσο ακόμα και αυτή η πλευρά προσομοιώνεται με την πρώτη πλατφόρμα ώστε να καλύψει περιπτώσεις πολύ υψηλού φόρτου διαχείρισης στον ACS. Τα αποτελέσματα απεικονίζονται στα σχήματα 3-26 και 3-27 για το μήνυμα της αίτησης και της απάντησης, αντίστοιχα. Οι χρόνοι στον κατακόρυφο άξονα αντιπροσωπεύουν το χρόνο μετάδοσης των SOAP μηνυμάτων προσαυξημένο με τους υπερκεφαλισμούς της συμπίεσης. Έτσι σύμφωνα με τους συμβολισμούς, που δόθηκαν κατά τη μοντελοποίηση του διαχειριστικού χρόνου (σχήμα 3-3) ο κατακόρυφος άξονας του σχήματος 3-26 αντιπροσωπεύει τον εξής χρόνο: $t_{ΣΥΜ(Δ)}+t_{ΜΜΑ}+t_{ΑΠΟΣ(Π)}$. Ομοίως ο κατακόρυφος άξονας του σχήματος 3-27 αντιπροσωπεύει τον ακόλουθο χρόνο: $t_{ΣΥΜ(Π)}+t_{ΜΜΑ}+t_{ΑΠΟΣ(Δ)}$.

Καταρχάς φαίνεται ότι για μικρό εύρος ζώνης οι χρόνοι μετάδοσης των μηνυμάτων διαχείρισης μπορούν να περιοριστούν δραματικά. Για παράδειγμα, για ρυθμό μετάδοσης 128Kbps και για 20 μόνο μεταβλητές ο χρόνος μετάδοσης του μηνύματος της απάντησης μειώνεται κατά 100ms (περίπου στο μισό του αρχικού), ενώ όταν ο αριθμός των μεταβλητών αγγίξει τις 200, τότε ο χρόνος μετάδοσης μειώνεται κατά 1,3 δευτερόλεπτα. Μάλιστα οι συνολικοί υπερκεφαλισμοί της συμπίεσης δεν ξεπερνούν ποτέ τα 5ms (στην περίπτωση του *GetParameterValues* μηνύματος της απάντησης με 200 μεταβλητές) και το γεγονός αυτό έχει ως αποτέλεσμα την κερδοφορία της συμπίεσης ακόμη και για ρυθμούς μετάδοσης που ξεπερνούν τα 15Mbps.



Σχήμα 3-26. Η επίδραση της συμπίεσης στους χρόνους μετάδοσης του *GetParameterValues* μηνύματος αίτησης.

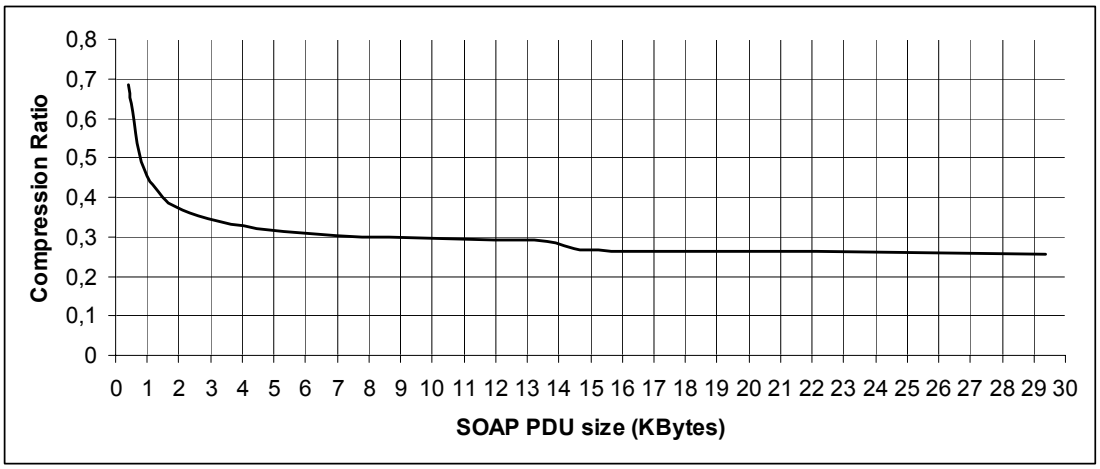


Σχήμα 3-27. Η επίδραση της συμπίεσης στους χρόνους μετάδοσης του *GetParameterValues* μηνύματος απάντησης.

3.5.8. Οι συνθήκες για την εφαρμογή της συμπίεσης

Κάτι το οποίο δεν φάνηκε στο προηγούμενο σχήμα είναι ότι στην πραγματικότητα η συμπίεση δεν αποδεικνύεται κερδοφόρα για όλα τα μηνύματα διαχείρισης. Αυτό οφείλεται στη φύση του Lempel-Ziv αλγορίθμου ο οποίος εμφανίζει μικρή αποδοτικότητα (μεγάλο

λόγο συμπίεσης) σε μικρά μηνύματα, εξαιτίας της περιορισμένης επαναληψιμότητας των συμβολοσειρών, που περιέχουν. Προκειμένου λοιπόν να καθοριστούν πλήρως οι συνθήκες κάτω από τις οποίες η συμπίεση των μηνυμάτων είναι κερδοφόρα (και επομένως θα πρέπει να εφαρμόζεται) η παρούσα εργασία πειραματίστηκε με ένα μεγάλο πλήθος από μηνύματα διαχείρισης και βρήκε τους αντίστοιχους λόγους συμπίεσης. Οι σχεδόν ίσοι λόγοι συμπίεσης σε διαφορετικά SOAP μηνύματα (*Inform, Get/SetParameterValues*) του ίδιου μεγέθους αποδεικνύουν ότι η αποδοτικότητα του αλγορίθμου σχετίζεται περισσότερο με το μέγεθος και πολύ λιγότερο με τον τύπο του μηνύματος ή τον αριθμό των παραμέτρων, που συμμετέχουν σε αυτό. Επομένως η συμπίεση μπορεί να εφαρμοστεί σε όλα τα SOAP μηνύματα με μόνο κριτήριο το μέγεθός τους. Το σχήμα 3-28 απεικονίζει το λόγο συμπίεσης για SOAP μηνύματα διαφορετικού μεγέθους.



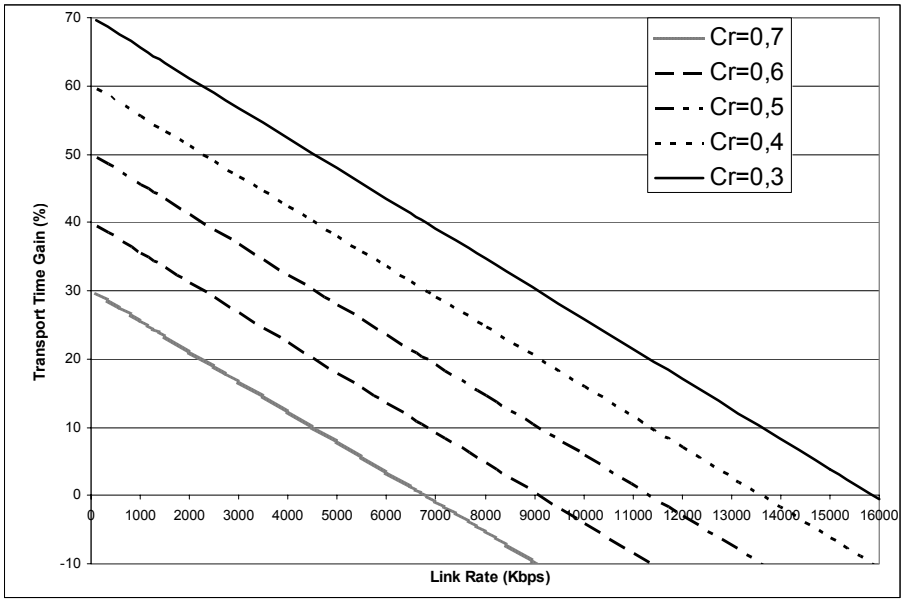
Σχήμα 3-28. Ο λόγος συμπίεσης για τα CWMP μηνύματα διαχείρισης.

Όλα τα CWMP μηνύματα ξεπερνούν τα 400bytes. Σε αυτό το μέγεθος παρουσιάζεται ο μεγαλύτερος λόγος συμπίεσης, αφού η επαναληψιμότητα είναι πολύ περιορισμένη και επομένως η απόδοση της συμπίεσης βρίσκεται σε χαμηλά επίπεδα. Ωστόσο η απόδοση ανεβαίνει πολύ γρήγορα αφού στα 1000bytes ο λόγος συμπίεσης μειώνεται στο 0,45, ενώ μετά τα 7Kbytes αγγίζει την τιμή 0,3. Η τελευταία τιμή διατηρείται για ένα μεγάλο διάστημα (7-14Kbytes) και στη συνέχεια πέφτει ακόμα περισσότερο (0,25) για ένα πάλι μεγάλο διάστημα (15-30Kbytes). Δεν κρίθηκε σκόπιμη η μέτρηση μεγαλύτερων μεγεθών αφού θεωρείται αρκετά σπάνιο ένα CWMP μήνυμα να ξεπεράσει αυτές τις τιμές (π.χ. μια GetParameterValues απάντηση με 200 μεταβλητές και τις τιμές τους φτάνει τα 29Kbytes).

Στο σημείο αυτό μπορεί να απαντηθεί το ερώτημα σχετικά με το πότε θα πρέπει να εφαρμόζεται η συμπίεση. Η απάντηση δίνεται από τη σχέση 3-12, που δίνει το κέρδος της συμπίεσης, όταν είναι γνωστό το διαθέσιμο εύρος ζώνης, ο λόγος συμπίεσης και οι ταχύτητες συμπίεσης και αποσυμπίεσης. Η σχέση 3-13 δίνει την συνθήκη εφαρμογής της συμπίεσης.

$$Gt_x = S_x[(1 - C_{ratio}) / LinkRate - 1/U_{c(y)} + 1/U_{D(z)}] > 0$$, x: μετάδοση μηνύματος αίτησης (ΜΜΑ) ή απόκρισης (ΜΜΑΠ) και y, z: Διαχειριστής (Δ) ή Πράκτορας (Π) (3-13)

Για να δοθεί μια εικόνα για το κέρδος και τις συνθήκες της συμπίεσης, στη σχέση 3-12 χρησιμοποιούνται οι ταχύτητες εκτέλεσης της πρώτης πλατφόρμας των πειραμάτων (ΠΛΑΤΦ1) και στη συνέχεια κατασκευάζονται οι καμπύλες του σχήματος 3-29, που απεικονίζουν το ποσοστιαίο κέρδος της συμπίεσης ($Gt_x/(S_x/LinkRate)$) για πέντε διαφορετικούς λόγους συμπίεσης.

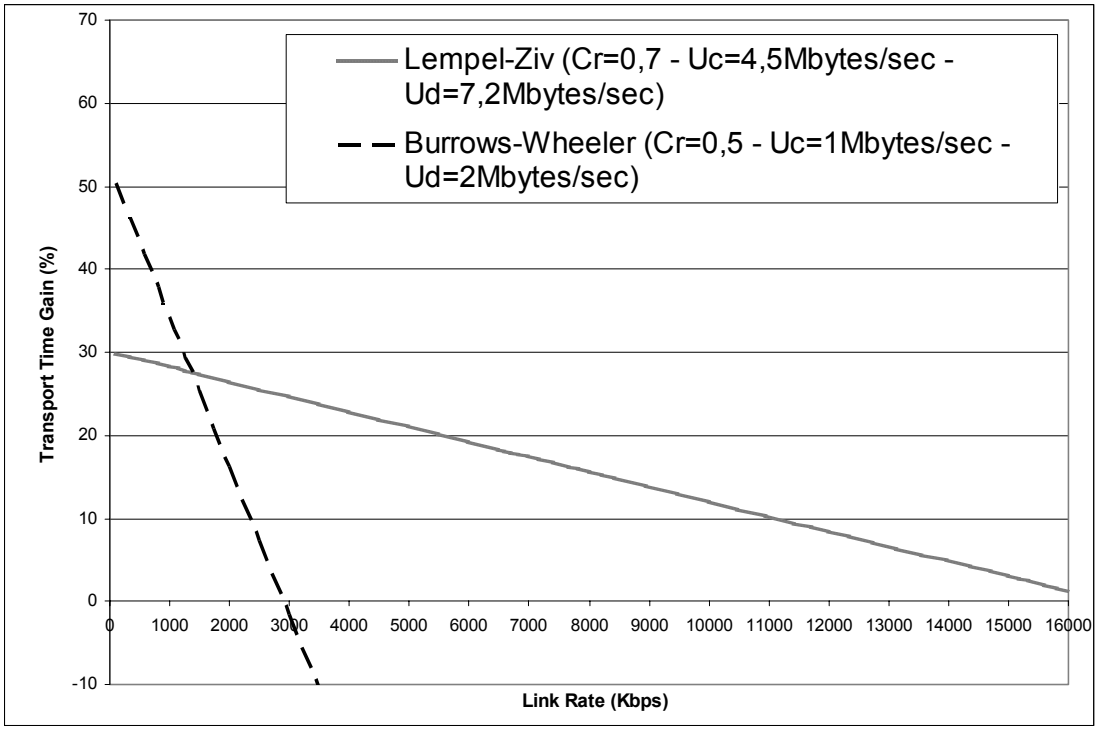


Σχήμα 3-29. Το ποσοστιαίο χρονικό κέρδος μετάδοσης κατά τη συμπίεση πραγματικού χρόνου.

Αν συνδυαστούν τα αποτελέσματα των σχημάτων 3-28 και 3-29 προκύπτει η συνθήκη εφαρμογής της συμπίεσης για την περίπτωση της πλατφόρμας ΠΛΑΤΦ1. Έτσι λοιπόν τα πολύ μικρά μηνύματα (C_r=0,7: 380-450bytes) αξίζει να συμπιέζονται όταν το διαθέσιμο εύρος ζώνης είναι μέχρι τα 6,8Mbps, διαφορετικά η συμπίεση παύει να είναι κερδοφόρα και ζημιώνει το χρόνο μετάδοσης. Αντίθετα τα μεγάλα μηνύματα (C_r=0,3: 7-14Kbytes) αξίζει να συμπιέζονται μέχρι και την τιμή των 16Mbps διαθέσιμου εύρους ζώνης. Με το ίδιο τρόπο βρίσκονται οι συνθήκες εφαρμογής της συμπίεσης και για τα υπόλοιπα μηνύματα (C_r=0,4-0,5-0,6) των οποίων οι καμπύλες βρίσκονται ανάμεσα στις δύο προηγούμενες. Είναι χαρακτηριστικό το γεγονός ότι ακόμα και για το ρυθμό μετάδοσης των 10Mbps το κέρδος χρόνου στα μεγάλα μηνύματα ξεπερνά το 25%. Μάλιστα αυτό συμβαίνει στην περίπτωση της πιο «αδύναμης» πλατφόρμας (ΠΛΑΤΦ1: Pentium I στα 133MHz), ενώ για την ισχυρότερη πλατφόρμα (ΠΛΑΤΦ3: PentiumIV στα 1,6GHz) το αντίστοιχο κέρδος ξεπερνά το 60%.

Στο σημείο αυτό πρέπει να τονιστεί ότι στην περίπτωση ύπαρξης κάποιας άλλης μεθόδου (ή και κάποιας διαφορετικής υλοποίησης της Lempel-Ziv) με μικρότερους λόγους συμπίεσης στα πολύ μικρά μηνύματα θα μπορούσε να εφαρμοστεί η λύση δύο μεθόδων συμπίεσης. Για να γίνει κατανοητή η ιδέα αυτή θα θεωρηθεί η μέθοδος Burrows-Wheeler της εργασίας [WiseMan] και θα γίνει η ακόλουθη υπόθεση: έστω ότι η υλοποίηση της μεθόδου Burrows-Wheeler παρουσιάζει λόγο συμπίεσης 0,5 (αντί 0,7 της LZOX1) στα πολύ μικρά CWMP μηνύματα (400bytes), ενώ οι ταχύτητες συμπίεσης και αποσυμπίεσης για την πλατφόρμα

ΠΛΑΤΦ1 είναι σχεδόν 4 φορές μικρότερες από εκείνες της LZOX1 (U_c=1,1Mbytes/sec και U_d=1,8Mbytes/sec). Για την περίπτωση αυτή η σύγκριση των ποσοστιαίων κερδών στα πολύ μικρά μηνύματα (400bytes) δίνεται στο σχήμα 3-30.



Σχήμα 3-30. Η σύγκριση του κέρδος μετάδοσης στα πολύ μικρά μηνύματα για διαφορετικές μεθόδους ή υλοποιήσεις συμπίεσης.

Το σχήμα 3-30 υποδηλώνει ότι για διαθέσιμο εύρος ζώνης μέχρι και 1,5Mbps η πιο κερδοφόρα μέθοδος συμπίεσης είναι η Burrows-Wheeler, ενώ μετά την τιμή αυτή η πιο κερδοφόρα γίνεται η Lemplel-Ziv (*LZOX1*). Το γεγονός αυτό οδηγεί στο συμπέρασμα ότι ο πράκτορας διαχείρισης θα μπορούσε να υποστηρίζει και τις δύο μεθόδους συμπίεσης εναλλάσσοντας την εφαρμογή τους ανάλογα με το κέρδος στο χρόνο μετάδοσης. Στην περίπτωση αυτή ο πράκτορας θα πρέπει να υπολογίσει δύο φορές τη σχέση 3-12, μια για κάθε μέθοδο συμπίεσης και να εφαρμόζει κάθε φορά εκείνη με το μεγαλύτερο κέρδος. Με παρόμοιο τρόπο θα μπορούσαν να εφαρμοστούν περισσότερες από δύο μέθοδοι συμπίεσης. Ωστόσο δεν προτείνεται κάτι τέτοιο, διότι αφενός η ύπαρξη περισσοτέρων μεθόδων θα αύξανε τις απαιτήσεις σε μνήμη του πράκτορα και αφετέρου θα προσέφερε ουσιστικά μόνο σε περιπτώσεις με εξαιρετικά χαμηλούς ρυθμούς μετάδοσης.

3.5.9. Περιορίζοντας το χρόνο κωδικοποίησης

Όπως φάνηκε και από τις εργασίες [5,6] η SOAP κωδικοποίηση μπορεί να επηρεάσει σημαντικά τον τελικό χρόνο διαχείρισης. Πιο συγκεκριμένα, η εργασία [6] χρησιμοποίησε την τεχνολογία DOM [33] και παρουσίασε αρκετά υψηλούς χρόνους κωδικοποίησης σε αντίθεση με την εργασία [5] η οποία χρησιμοποίησε ειδικούς γραμματικούς αναλυτές με αποτέλεσμα οι χρόνοι κωδικοποίησης να είναι σχεδόν αμελητέοι. Η παρούσα ενότητα

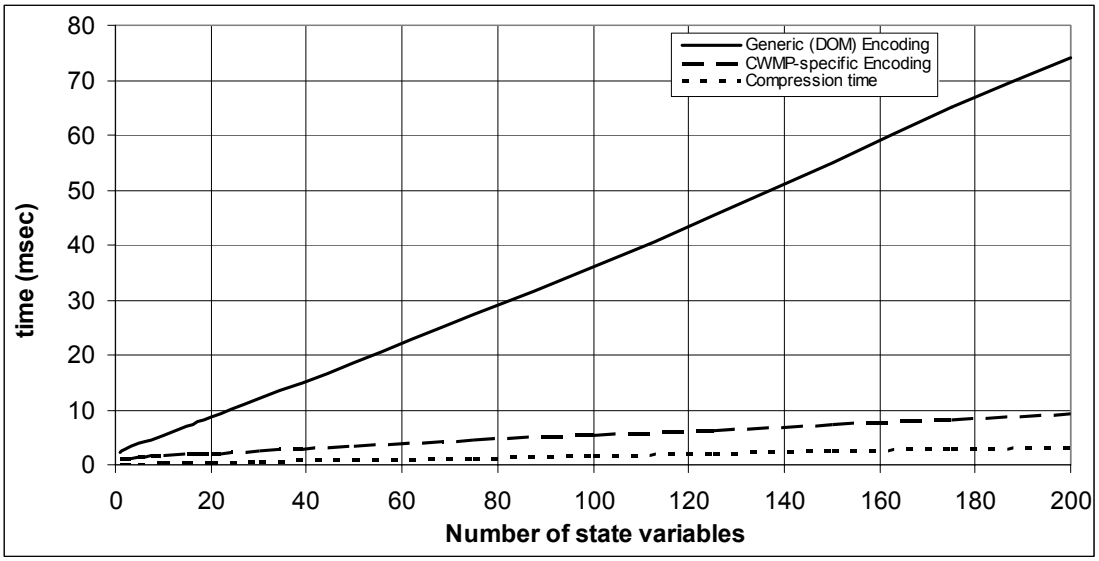
πειραματίζεται με τις δύο παραπάνω προσεγγίσεις με στόχο την ποσοτικοποίηση της χρονικής βελτίωσης, που επιφέρει στη διαχείριση η προσέγγιση των ειδικών γραμματικών αναλυτών έναντι εκείνης των κλασσικών γραμματικών αναλυτών.

Οι βασικές τεχνολογίες κλασσικών XML γραμματικών αναλυτών είναι η DOM και η SAX [71]. Οι αναλυτές που υποστηρίζουν αυτές τις τεχνολογίες καλούνται «γενικοί αναλυτές» διότι μπορούν να χρησιμοποιηθούν για την κωδικοποίηση / αποκωδικοποίηση οποιουδήποτε XML αρχείου. Η βασική διαφορά μεταξύ των δύο είναι ότι οι DOM αναλυτές διαβάζουν ολόκληρο το XML κείμενο και δημιουργούν μια δενδρική δομή στη μνήμη (DOM δέντρο), ενώ οι SAX διαβάζουν το κείμενο σειριακά και καλούν μια συνάρτηση καθορισμένη από το χρήστη κάθε φορά που συναντούν ένα XML στοιχείο. Με αυτό τον τρόπο οι SAX αναλυτές δίνουν τη δυνατότητα στον προγραμματιστή να χτίσει τη δική του δομή στη μνήμη η οποία μπορεί να περιέχει πολύ λιγότερη πληροφορία από εκείνη του DOM δέντρου. Για το λόγο αυτό οι SAX αναλυτές είναι συνήθως γρηγορότεροι, ενώ παράλληλα καταναλώνουν και λιγότερη δυναμική μνήμη.

Ωστόσο, πέραν των παραπάνω κλασσικών μεθόδων γραμματικής ανάλυσης ένα SOAP διαχειριστικό μήνυμα θα μπορούσε να αποκωδικοποιηθεί και με ειδικούς γραμματικούς αναλυτές για κάθε περίπτωση. Η προσέγγιση αυτή συνίσταται στη δημιουργία ενός διαφορετικού αναλυτή για κάθε ξεχωριστό διαχειριστικό μήνυμα. Κάθε τέτοιος αναλυτής είναι ενήμερος για τη δομή του μηνύματος που αναλύει. Το μόνο λοιπόν που χρειάζεται να κάνει είναι να ψάχνει μέσα στο μήνυμα συγκεκριμένα στοιχεία (π.χ. "*name*", "*value*") και να αποθηκεύει τις αντίστοιχες τιμές τους. Με αυτό τον τρόπο η αποκωδικοποίηση μπορεί να γίνει πολύ γρήγορα καθώς δεν χρησιμοποιούνται πολύπλοκες τεχνικές σαν και αυτές που χρησιμοποιούν οι DOM και SAX μέθοδοι. Από την άλλη πλευρά το τίμημα που πληρώνει μια τέτοια προσέγγιση είναι το κόστος υλοποίησης των πολλών διαφορετικών γραμματικών αναλυτών. Βέβαια στην περίπτωση του πρώτου έτοιμου προτύπου απομακρυσμένης διαχείρισης CWMP, ο αριθμός των SOAP μηνυμάτων είναι ιδιαίτερα περιορισμένος (*Set/GetParamaterValues, Set/GetParameterAttributes, Inform, GetRPCMethods, GetParameterNames, Reboot* και *Download*) οπότε το κόστος δεν αναμένεται υψηλό. Στο σημείο αυτό θα πρέπει να τονιστεί ότι πέραν των ειδικών αναλυτών για κάθε διαχειριστική αίτηση και απάντηση, θα πρέπει να υπάρχει και ακόμα ένας ειδικός αναλυτής. Αυτός θα υποδέχεται όλα τα μηνύματα τα οποία και θα δρομολογεί (εντός της εφαρμογής του CWMP πράκτορα) στον αντίστοιχο ειδικό αναλυτή, σύμφωνα με το πεδίο της SOAP κεφαλίδας, που υποδηλώνει τον τύπο κάθε μηνύματος (π.χ. "*CWMP: Inform*").

Τα παραπάνω σχετίζονται μόνο με την αποκωδικοποίηση των μηνυμάτων διαχείρισης. Όσον αφορά στην κωδικοποίηση, η DOM μέθοδος δημιουργεί πρώτα στη μνήμη ένα δέντρο και στη συνέχεια από αυτό κατασκευάζει αυτόματα το XML κείμενο. Αντίθετα οι SAX και οι ειδικοί αναλυτές κατασκευάζουν «χειροκίνητα» το XML αρχείο. Στην πράξη έχουν έτοιμα πρότυπα SOAP μηνυμάτων τα οποία κάθε φορά τροποποιούν για να φτιάξουν το σωστό μήνυμα. Η διαδικασία αυτή επίσης επιταχύνει την κωδικοποίηση των μηνυμάτων διαχείρισης, αφού δεν απαιτεί τη δημιουργία πολύπλοκων δομών στη μνήμη, όπως συμβαίνει στην περίπτωση του DOM.

Προκειμένου να αποκτηθεί μια εικόνα σχετικά με την επίδραση της υλοποίησης στους χρόνους κωδικοποίησης και αποκωδικοποίησης η παρούσα εργασία μέτρησε τους χρόνους αυτούς για την γρηγορότερη (ειδικοί αναλυτές) και την πιο αργή (DOM) προσέγγιση. Για τις μετρήσεις αυτές χρησιμοποιήθηκε η πλατφόρμα ΠΛΑΤΦ2. Το σχήμα 3-31 απεικονίζει τους συνολικούς χρόνους (από) κωδικοποίησης ($t_{ΑΠΟΚ(Π)}$ + $t_{ΚΩΔ(Π)}$) και (από) συμπίεσης ($t_{ΣΥΜ(Π)}$+ $t_{ΑΠΟΣ(Π)}$) της *GetParameterValues* αίτησης και απάντησης, αντίστοιχα.. Φαίνεται ότι η διαφορά μεταξύ των δύο προσεγγίσεων κωδικοποίησης είναι αρκετά μεγάλη, αφού στην περίπτωση του DOM ο συνολικός χρόνος εκτέλεσης είναι από 2 μέχρι και 8 φορές μεγαλύτερος σε σχέση με εκείνο των ειδικών αναλυτών. Μάλιστα όσο οι δυνατότητες της πλατφόρμας του διαχειριστή περιορίζονται τόσο η διαφορά αυτή αυξάνεται. Έτσι στην περίπτωση της πλατφόρμας ΠΛΑΤΦ1 οι αντίστοιχοι χρόνοι είναι σχεδόν διπλάσιοι.



Σχήμα 3-31. Οι χρόνοι κωδικοποίησης και συμπίεσης των μηνυμάτων διαχείρισης.

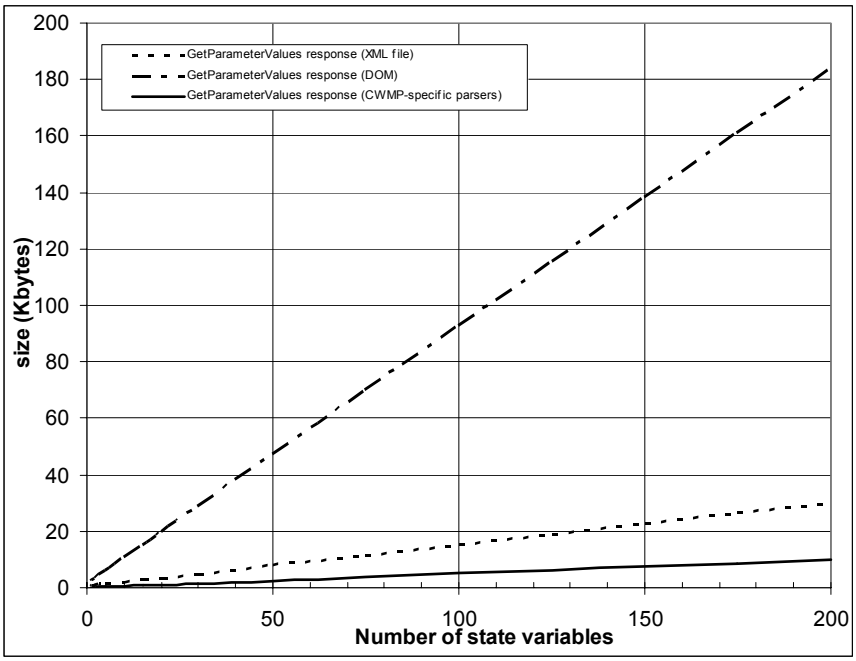
Ένα ακόμα βασικό συμπέρασμα, που προκύπτει από το σχήμα 3-31 είναι ότι ο χρόνος που απαιτείται για τη συμπίεση των μηνυμάτων είναι σχεδόν 25 φορές μικρότερος από εκείνο που απαιτείται για την DOM κωδικοποίηση και από 3 (στα πολύ μεγάλα μηνύματα) έως 13 φορές (στα πολύ μικρά μηνύματα) μικρότερος από την κωδικοποίηση των ειδικών αναλυτών. Τα αποτελέσματα αυτά έρχονται σε αντίθεση με τα αποτελέσματα της εργασίας [5], που παρουσιάστηκαν στο σχήμα 3-14 τα οποία δείχνουν ότι η συμπίεση είναι περίπου 2,5 φορές πιο χρονοβόρα από την κωδικοποίηση των ειδικών αναλυτών. Το γεγονός αυτό οφείλεται στη διαφορετική προσέγγιση, που ακολουθεί η παρούσα εργασία κατά την οποία αντί μιας αποδοτικότερης αλλά πιο αργής Lempel-Ziv υλοποίησης (όπως εκείνη της εργασίας [5]) χρησιμοποιεί την υλοποίηση συμπίεσης πραγματικού χρόνου (*LZOX1*) η οποία είναι λιγότερο αποδοτική αλλά πολύ ταχύτερη. Η προσέγγιση αυτή, όπως φάνηκε και στο σχήμα 3-29, αναιρεί την εκτίμηση της εργασίας [5] για την εφαρμογή της συμπίεσης μόνο σε εξαιρετικά χαμηλούς ρυθμούς μετάδοσης ενώ παράλληλα μειώνει δραστικά τον όγκο της κίνησης διαχείρισης.

Επίσης το γεγονός ότι τις περισσότερες φορές η κωδικοποίηση είναι τουλάχιστον κατά μια τάξη μεγέθους πιο χρονοβόρα από τη συμπίεση, σε συνδυασμό με το ότι η συμπίεση και η κωδικοποίηση είναι δυο διαδοχικές διαδικασίες οδηγεί σε μια σημαντική απλούστευση: στην περίπτωση της διαχείρισης δεν χρειάζεται πολύπλοκο μεσισμικό το οποίο να παρακολουθεί το φορτίο του επεξεργαστή κάθε στιγμή και ανάλογα με αυτό και το διαθέσιμο εύρος ζώνης να εφαρμόζει διαφορετικούς τύπους συμπίεσης (κάτι που έγινε στην εργασία [48]). Αυτό συμβαίνει διότι στην περίπτωση που ο επεξεργαστής της διαχειριζόμενης συσκευής υπερφορτωθεί, δεν θα καθυστερήσει μόνο η συμπίεση αλλά και η κωδικοποίηση (τη στιγμή που η τελευταία ακολουθεί πάντα την πρώτη) η οποία μάλιστα θα έχει πολύ μεγαλύτερη επίδραση στο χρόνο διαχείρισης καθώς είναι πολύ πιο χρονοβόρα. Έτσι λοιπόν στον τελικό χρόνο διαχείρισης η συνεισφορά της καθυστέρησης από τη συμπίεση θα είναι αμελητέα σε σχέση με εκείνη της κωδικοποίησης.

3.5.10. Περιορίζοντας τις απαιτήσεις σε μνήμη

Όπως προαναφέρθηκε, η DOM προσέγγιση παρουσιάζει ένα ακόμη μειονέκτημα το οποίο μπορεί να αποδειχτεί κρίσιμο σε συσκευές με περιορισμένες δυνατότητες μνήμης. Πρόκειται για τις μεγάλες δομές, που δημιουργούνται στη μνήμη κατά την κωδικοποίηση και αποκωδικοποίηση των SOAP μηνυμάτων διαχείρισης. Κατά τη διάρκεια των πειραμάτων πραγματοποιήθηκαν επίσης μετρήσεις οι οποίες αφορούν στη δυναμική μνήμη, που καταλαμβάνεται από τις δύο προσεγγίσεις (DOM και ειδικοί αναλυτές). Το σχήμα 3-32 απεικονίζει τη χρήση της δυναμικής μνήμης κατά την αποκωδικοποίηση της *GetParameterValues* απάντησης, που περιέχει τα ονόματα και τις τρέχουσες τιμές των μεταβλητών κατάστασης. Επειδή η δομή που κατασκευάζεται στη μνήμη κατά την κωδικοποίηση είναι η ίδια με εκείνη που κατασκευάζεται κατά την αποκωδικοποίηση (όταν πρόκειται για το ίδιο μήνυμα) τα αποτελέσματα του σχήματος 3-32 ισχύουν και για την κωδικοποίηση της *GetParameterValues* απάντησης.

Καταρχάς παρατηρείται ότι το μέγεθος της μνήμης, που απαιτεί η DOM υλοποίηση για να (απο)κωδικοποιήσει μια SOAP εντολή είναι από 4 (στα μικρά μηνύματα) έως 6 φορές (στα μεγάλα μηνύματα) μεγαλύτερο του αντίστοιχου SOAP/XML φακέλου. Για κάθε παράμετρο χρειάζεται σχεδόν 1Kbyte. Όταν πρόκειται για μικρό αριθμό παραμέτρων το μέγεθος αυτό είναι συνήθως αμελητέο, αλλά μπορεί να αποβεί κρίσιμο σε περιπτώσεις μαζικής ανάκτησης ή ρύθμισης (αφού και για τη *SetParameterValues* ισχύουν παρόμοια αποτελέσματα) παραμέτρων. Το πρόβλημα αυτό μπορεί να γίνει ιδιαίτερα έντονο σε χαμηλού κόστους συσκευές με περιορισμένη μνήμη. Από την άλλη πλευρά η χρήση της μνήμης από τον ειδικό αναλυτή είναι από 3 (στα μηνύματα με περισσότερες από 10 παραμέτρους) έως 12 φορές (στο μήνυμα με 1 παράμετρο) μικρότερη από το αντίστοιχο SOAP/XML κείμενο και μέχρι και 19 φορές μικρότερη από την αντίστοιχη του DOM αναλυτή.



Σχήμα 3-32. Η δυναμική μνήμη που καταλαμβάνεται κατά την αποκωδικοποίηση.

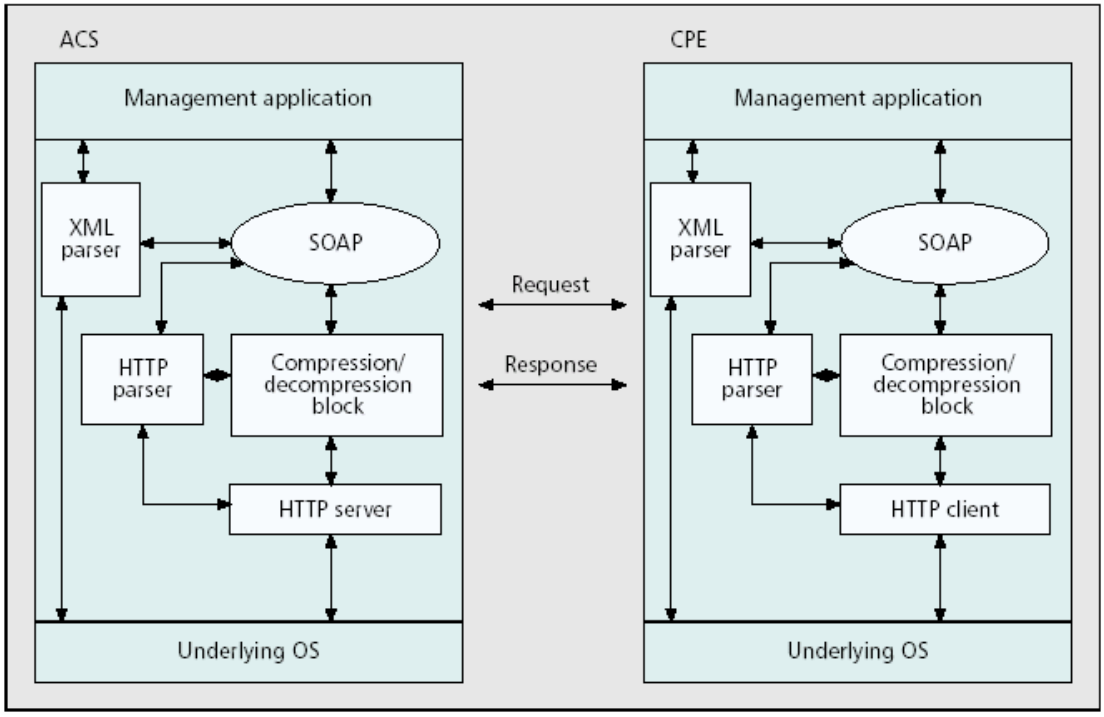
3.5.11. Συνολική αποτίμηση των προτεινόμενων λύσεων

Μέχρι εδώ παρουσιάστηκε τμηματικά η αποτίμηση των προτεινόμενων προσεγγίσεων τόσο για τη συμπίεση όσο και για την κωδικοποίηση. Συγκεκριμένα δόθηκαν τα περιθώρια μείωσης του όγκου των SOAP μηνυμάτων διαχείρισης κατά τη συμπίεση καθώς και η επακόλουθη βελτίωση των χρόνων μετάδοσης ($t_{ΜΜΑ}$ και $t_{ΜΜΑΠ}$, βλέπε σχήμα 3-3). Επίσης παρουσιάστηκε η βελτίωση, που επιφέρει στους χρόνους κωδικοποίησης ($t_{ΚΩΔ(x)}$ και $t_{ΑΠΟΚ(x)}$,βλέπε σχήμα 3-3) η προσέγγιση των ειδικών γραμματικών αναλυτών. Ωστόσο αυτό που τελικά ενδιαφέρει στη διαχείριση είναι ο συνολικός χρόνος διαχείρισης ($t_{ΚΩΔ(Δ)} + t_{ΣΥΜ(Δ)} + t_{ΜΜΑ} + t_{ΑΠΟΣ(Π)} + t_{ΑΠΟΚ(Π)} + t_{ΕΝ} + t_{ΚΩΔ(Π)} + t_{ΣΥΜ(Π)} + t_{ΜΜΑΠ} + t_{ΑΠΟΣ(Δ)} + t_{ΑΠΟΚ(Δ)}$, βλέπε σχήμα 3-3) και επομένως η επίδραση της κάθε προτεινόμενης προσέγγισης στο χρόνο αυτό. Στην παρούσα ενότητα διεξάγεται ένα ολοκληρωμένο πείραμα διαχείρισης και μετράται ο συνολικός χρόνος διαχείρισης καθώς και η συνολική κίνηση, που παράγεται κατά τη διάρκεια μιας πλήρους CWMP συνεδρίας.

3.5.11.1. Η αρχιτεκτονική του πειραματικού συστήματος διαχείρισης

Το πειραματικό σύστημα διαχείρισης αποτελείται από τον ACS διαχειριστή και τον αντίστοιχο CWMP πράκτορα. Τα βασικότερα στοιχεία του απεικονίζονται στο σχήμα 3-33. Ο πράκτορας περιλαμβάνει ένα HTTP πελάτη (HTTP client) ο οποίος εκκινεί τις διαχειριστικές συνεδρίες στέλνοντας ένα *HTTP POST* μήνυμα (*Inform* αίτηση) στον ACS διαχειριστή. Ο ACS απαντά στον πράκτορα (HTTP response) στέλνοντας τη διαχειριστική αίτηση *GetParameterValues* (πέραν της *Inform* απάντησης που προβλέπει το CWMP πρωτόκολλο). Ο HTTP πελάτης του πράκτορα λαμβάνει τα μηνύματα του ACS διαχειριστή των οποίων την

HTTP κεφαλίδα παραδίδει στον αντίστοιχο αναλυτή (HTTP parser). Στη συνέχεια μεταφέρει τον έλεγχο στη μονάδα συμπίεσης / αποσυμπίεσης (Compression / decompression block). Η τελευταία χρησιμοποιεί τον HTTP αναλυτή για να διαβάσει την κεφαλίδα του HTTP πακέτου στην οποία αναγράφεται η πληροφορία σχετικά με το εάν το περιεχόμενο του πακέτου είναι συμπιεσμένο. Στην περίπτωση που είναι συμπιεσμένο η HTTP κεφαλίδα αναγράφει και την αντίστοιχη μέθοδο με την οποία έγινε η συμπίεση. Έτσι με βάση αυτή την πληροφορία η μονάδα συμπίεσης / αποσυμπίεσης αποσυμπιέζει το περιεχόμενο του HTTP πακέτου και παραδίδει το SOAP μήνυμα στη μονάδα SOAP για περαιτέρω επεξεργασία. Η SOAP μονάδα αποκωδικοποιεί τα μηνύματα διαχείρισης τα οποία και στη συνέχεια στέλνει στη διαχειριστική εφαρμογή (Management Application) για εκτέλεση. Η εκτέλεση αφορά κυρίως στην κλήση των μεθόδων ενορχήστρωσης, οι οποίες εκτελούν στο σύστημα τις πραγματικές διαχειριστικές πράξεις, που προηγουμένως αιτήθηκε ο ACS διαχειριστής. Αφού οι μέθοδοι αυτές επιστρέψουν, η διαχειριστική εφαρμογή χρησιμοποιεί ξανά τη SOAP μονάδα προκειμένου να κατασκευάσει την απάντηση του πράκτορα στο αίτημα του διαχειριστή με βάση τα αποτελέσματα των μεθόδων ενορχήστρωσης. Πριν σταλεί το μήνυμα της απάντησης παρεμβαίνει πάλι η μονάδα συμπίεσης / αποσυμπίεσης η οποία με βάση τη σχέση 3-13 αποφασίζει για το εάν είναι κερδοφόρα η εφαρμογή της συμπίεσης. Στην περίπτωση που η απάντηση είναι θετική, η μονάδα συμπιέζει τη SOAP απάντηση την οποία και στη συνέχεια παραδίδει στον HTTP πελάτη για να τη στείλει πίσω στον ACS διαχειριστή. Μάλιστα η μονάδα συμπίεσης αναφέρει στον HTTP πελάτη ότι το μήνυμα είναι συμπιεσμένο καθώς και τη μέθοδο συμπίεσης, που χρησιμοποιήθηκε προκειμένου ο τελευταίος να εισάγει την πληροφορία αυτή στην HTTP κεφαλίδα. Περισσότερες λεπτομέρειες για την αρχιτεκτονική του CWMP πράκτορα αναφέρεται στο επόμενο κεφάλαιο.



Σχήμα 3-33. Η αρχιτεκτονική του πειραματικού συστήματος CWMP διαχείρισης

Η λειτουργία του ACS διαχειριστή είναι σε πλήρη αντιστοιχία με εκείνη του πράκτορα, που περιγράφτηκε παραπάνω με τη μόνη διαφορά ότι η εφαρμογή του διαχειριστή δεν εκτελεί κάποια μέθοδο ενορχήστρωσης, παρά μόνο εφαρμόζει τις πολιτικές διαχείρισης για την κάθε υπηρεσία, συσκευή και πελάτη.

3.5.11.2. Η περιγραφή των ολοκληρωμένων πειραμάτων

Η εφαρμογή του ACS διαχειριστή φιλοξενείται στην πλατφόρμα ΠΛΑΤΦ3 (επεξεργαστής Pentium IV στα 1,6 GHz με 1GByte μνήμη RAM), ενώ η εφαρμογή του πράκτορα στην πλατφόρμα ΠΛΑΤΦ2 (πλατφόρμα MiniITX για ενσωματωμένες εφαρμογές με τον επεξεργαστή Via EDEN ESP Samuel στα 600MHz και μνήμη 256Kbyte). Οι δύο πλατφόρμες συνδέονται πλάτη με πλάτη, μέσω μιας 100Mbps Ethernet σύνδεσης. Καθώς και οι δύο πλατφόρμες υποστηρίζονται από το λειτουργικό σύστημα Linux (Kernel 2.4.22), χρησιμοποιήθηκε το εργαλείο Traffic Controller (tc) [72] προκειμένου να περιοριστεί ο ρυθμός μετάδοσης των δεδομένων, που στέλνει η μια στην άλλη. Με αυτό τον τρόπο επιτυγχάνεται η προσομοίωση των ρυθμών μετάδοσης, που υποστηρίζονται από τις διάφορες τεχνολογίες πρόσβασης. Τα πειράματα ξεκίνησαν από τα 100Kbps, έγιναν με βήμα τα 100Kbps και έφτασαν μέχρι τα 100Mbps. Μάλιστα οι ρυθμοί μετάδοσης είναι συμμετρικοί.

Επειδή αυτό που ενδιαφέρει είναι ο συνολικός χρόνος διαχείρισης, τοποθετήθηκαν χρονοσφραγίδες (timestamps) μόνο στην πλευρά του διαχειριστή και μάλιστα στα εξής δύο σημεία: (i) αμέσως πριν την κωδικοποίηση του μηνύματος αίτησης (*GetParameterValues* request) από το διαχειριστή και (ii) αμέσως μετά την αποκωδικοποίηση του μηνύματος απάντησης (*GetParameterValues* response) από το διαχειριστή. Το σχήμα 3-3 της παραγράφου 3.3.2 βοηθά στην κατανόηση του συνολικού χρόνου διαχείρισης. Τη λειτουργία των χρονοσφραγίδων επιτελεί η συνάρτηση gettimeofday() του UNIX, η οποία αποτελεί ένα συνεχή μετρητή του χρόνου σε microseconds. Η διαφορά των δύο παραπάνω χρόνων δίνει το συνολικό χρόνο διαχείρισης ($t_{O\Lambda}$). Μάλιστα σε κάθε περίπτωση ο χρόνος αυτός υπολογίστηκε για 1000 επαναλαμβανόμενες συνεδρίες και αφού διαπιστώθηκε ότι δεν υπήρχε αξιοσημείωτη διασπορά των τιμών του, ελήφθη υπόψη η μέση τιμή του.

Στόχος ήταν να αποτιμηθούν τα εξής ακόλουθα: (i) η βελτίωση που αποφέρει η προσέγγιση των ειδικών αναλυτών έναντι της κλασσικής DOM μεθόδου κωδικοποίησης και αποκωδικοποίησης των μηνυμάτων διαχείρισης, (ii) η περαιτέρω βελτίωση που αποφέρει στο χρόνο διαχείρισης η επιλεκτική συμπίεση των SOAP μηνυμάτων διαχείρισης και (iii) η μείωση του όγκου της κίνησης διαχείρισης που αποφέρει πάλι η συμπίεση των SOAP μηνυμάτων. Για την πρώτη περίπτωση ο συνολικός χρόνος διαχείρισης μετρήθηκε μια φορά χρησιμοποιώντας στον πράκτορα και στο διαχειριστή την κλασσική DOM μέθοδο και άλλη μια χρησιμοποιώντας τη μέθοδο των ειδικών αναλυτών. Όσον αφορά στη δεύτερη περίπτωση, ο συνολικός χρόνος διαχείρισης μετρήθηκε χρησιμοποιώντας τους ειδικούς αναλυτές και εφαρμόζοντας την επιλεκτική συμπίεση. Κατά τη διάρκεια των παραπάνω πειραμάτων μετρήθηκε με το εργαλείο *Ethereal* [73] ο συνολικός όγκος της κίνησης μεταξύ του ACS διαχειριστή και του CWMP πράκτορα. Έτσι τα πειράματα της πρώτης περίπτωσης

αντιστοιχούν στο συνολικό όγκο κίνησης χωρίς συμπίεση, ενώ τα πειράματα της δεύτερης αντιστοιχούν στο μειωμένο όγκο κίνησης εξαιτίας της συμπίεσης.

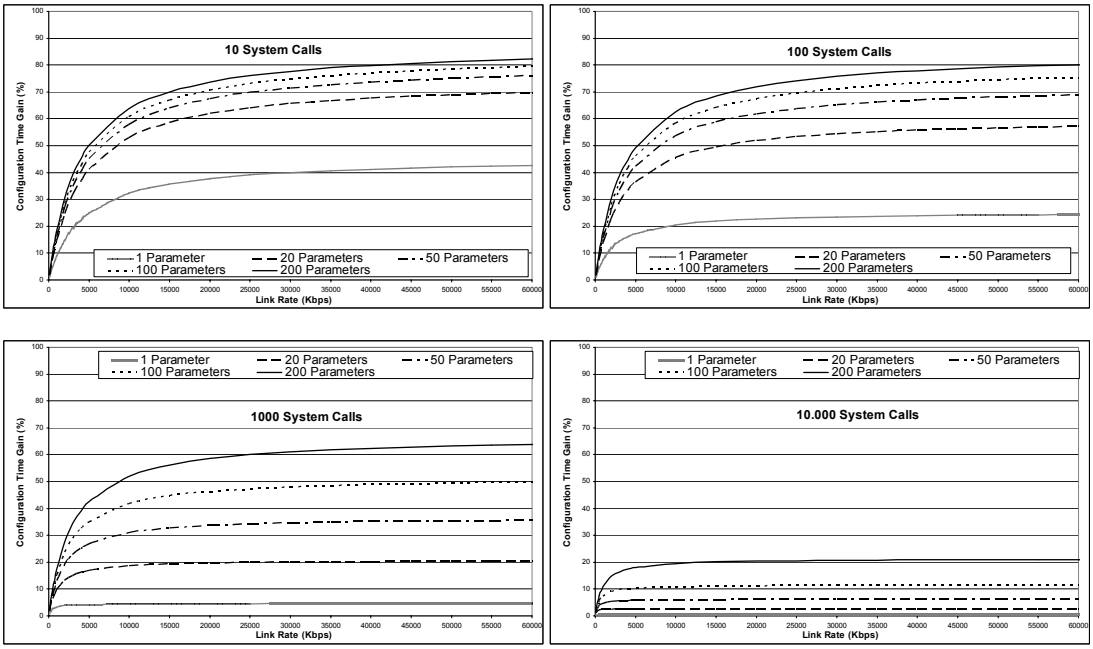
Πέραν του χρόνου κωδικοποίησης και μετάδοσης των μηνυμάτων διαχείρισης υπάρχει και ο χρόνος ενορχήστρωσης ο οποίος μπορεί να επηρεάσει σημαντικά τον τελικό χρόνο διαχείρισης. Αυτό φάνηκε ιδιαίτερα στην εργασία [5], που αναλύθηκε στην παράγραφο 3.4.2.2, η οποία απέδειξε ότι ακόμη και μετά από 15 χρόνια χρήσης του SNMP πρωτοκόλλου, σήμερα υπάρχουν πολλές SNMP συσκευές, που εξαιτίας της μη βέλτιστης υλοποίησης των μεθόδων ενορχήστρωσής τους παρουσιάζουν σημαντικά υψηλούς χρόνους διαχείρισης. Το γεγονός αυτό ελήφθη υπόψη στην παρούσα εργασία και έτσι στα πειράματα που διεξήχθησαν συμμετείχαν και οι μέθοδοι ενορχήστρωσης οι οποίες προσομοιώθηκαν από ένα μεταβλητό αριθμό κλήσεων συστήματος (system calls). Όπως απέδειξε και η εργασία [5] οι κλήσεις συστήματος είναι απαραίτητες για την υλοποίηση των πράξεων ρύθμισης και παρακολούθησης μιας συσκευής και είναι εκείνες που καθορίζουν το χρόνο ενορχήστρωσης. Για την προσομοίωση της διαδικασίας ενορχήστρωσης χρησιμοποιήθηκε μια ποικιλία από τις διαθέσιμες κλήσεις συστήματος του Linux (*open*, *close*, *execve*). Μάλιστα για την προσομοίωση της ενορχήστρωσης με 10 κλήσεις, όλες οι χρησιμοποιούμενες κλήσεις ήταν διαφορετικές, ενώ για μεγάλο αριθμό κλήσεων (100, 1000 και 10.000) γίνεται ισομερής χρήση της κάθε μιας από τις παραπάνω. Μάλιστα για την επαλήθευση των αποτελεσμάτων χρησιμοποιήθηκαν και συναρτήσεις του λειτουργικού, που επιτελούν πολλές κλήσεις συστήματος (π.χ. η *system("route")* καλεί περίπου 200 συναρτήσεις συστήματος).

3.5.11.3. Η αποτίμηση της βελτιστοποιημένης κωδικοποίησης

Το σχήμα 3-34 παρουσιάζει το ποσοστιαίο κέρδος στο συνολικό χρόνο διαχείρισης, που επιφέρει η προσέγγιση των ειδικών γραμματικών αναλυτών σε σχέση με εκείνη της κλασσικής DOM μεθόδου. Καθένα από τα 4 διαγράμματα αντιστοιχεί σε ένα διαφορετικό αριθμό κλήσεων συστήματος και επομένως σε διαφορετικούς χρόνους των μεθόδων ενορχήστρωσης. Επίσης κάθε διάγραμμα περιλαμβάνει 5 διαφορετικές καμπύλες, η καθεμία εκ των οποίων αντιστοιχεί σε διαφορετικό αριθμό παραμέτρων (1, 20, 50, 100 και 200), που εμπλέκονται στη διαχειριστική πράξη (σαν ορίσματα της μεθόδου *GetParameterValues*).

Καταρχάς φαίνεται ότι για πολύ μικρούς ρυθμούς μετάδοσης το κέρδους, που αποφέρει η βελτιστοποιημένη κωδικοποίηση είναι σχεδόν αμελητέο. Έτσι στα 100Kbps το μέγιστο κέρδος στο συνολικό χρόνο διαχείρισης είναι μόνο 2,3%. Αντίθετα όσο αυξάνει ο ρυθμός μετάδοσης, τόσο μεγαλώνει το κέρδος από τη βελτιστοποίηση της κωδικοποίησης. Μάλιστα μετά τα 40Mbps το κέρδος μπορεί να ξεπεράσει ακόμα και το 80% (δηλαδή πενταπλάσιοι χρόνοι διαχείρισης με τη μέθοδο DOM σε σχέση με αυτούς της μεθόδου των ειδικών αναλυτών). Αυτό εξηγείται από το γεγονός ότι στις μικρές τιμές του εύρους ζώνης οι χρόνοι μετάδοσης είναι πολύ υψηλότεροι από εκείνους της κωδικοποίησης με αποτέλεσμα η όποια βελτίωση στην κωδικοποίηση να διαδραματίζει μικρό ή και σχεδόν αμελητέο ρόλο στο συνολικό χρόνο διαχείρισης. Αντίθετα όσο αυξάνει το εύρος ζώνης τόσο ελαττώνονται οι χρόνοι μετάδοσης, με αποτέλεσμα στους μεγάλους ρυθμούς μετάδοσης τον κυρίαρχο ρόλο να τον διαδραματίζει η κωδικοποίηση και όχι η μετάδοση των μηνυμάτων διαχείρισης. Συνεπώς

όσο μεγαλύτερο είναι το εύρος ζώνης, τόσο μεγαλύτερο νόημα αποκτά η όποια βελτιστοποίηση στην κωδικοποίηση. Μάλιστα θα μπορούσε να ισχυριστεί κανείς ότι μετά τα 5Mbps η αντικατάσταση της DOM μεθόδου θεωρείται κρίσιμη για τις επιδόσεις του συστήματος.



Σχήμα 3-34. Το ποσοστιαίο χρονικό κέρδος της κωδικοποίησης των ειδικών αναλυτών έναντι των κλασσικών DOM αναλυτών.

Επίσης φαίνεται ότι το κέρδος της κωδικοποίησης εξαρτάται και από τον αριθμό των παραμέτρων, που εμπεριέχονται στο SOAP μήνυμα. Μάλιστα δεν είναι γραμμικό, αφού ήδη στις 20 πρώτες παραμέτρους έχει αγγίξει το 85% της μέγιστης τιμής του (που παρουσιάζεται στις 200 παραμέτρους). Το γεγονός αυτό είναι ιδιαίτερα ενθαρρυντικό, διότι το κέρδος της κωδικοποίησης θα παραμένει υψηλό για τα περισσότερα μηνύματα διαχείρισης και όχι μόνο για τα λιγοστά μεγάλα μηνύματα.

Τέλος, φαίνεται ότι όταν η διαδικασία της ενορχήστρωσης είναι χρονοβόρα, η όποια βελτιστοποίηση στην κωδικοποίηση χάνει μεγάλο μέρος της αξίας της. Για παράδειγμα, στην περίπτωση που η υλοποίηση της ενορχήστρωσης αποτελείται από 10 κλήσεις συστήματος το μέγιστο κέρδος χρόνου ξεπερνά το 80%, σε αντίθεση με την περίπτωση των 10.000 κλήσεων, όπου το μέγιστο κέρδος δεν ξεπερνά το 22%. Ο αυξημένος χρόνος ενορχήστρωσης για μια δεδομένη συσκευή (υλικό και λειτουργικό σύστημα) οφείλεται κυρίως στη μη βέλτιστη υλοποίηση των μεθόδων ενορχήστρωσης από την εφαρμογή του πράκτορα (δηλαδή στη σπάταλη χρήση των κλήσεων συστήματος) καθώς και στην πολύπλοκη φύση της αιτηθείσας διαχειριστικής ενέργειας (όταν η ενέργεια αφορά και εμπλέκει πολλές διαφορετικές λειτουργικές μονάδες του συστήματος). Η τελευταία παράμετρος δεν μπορεί να προβλεφθεί ή να περιοριστεί διότι ο διαχειριστής θα πρέπει να είναι σε θέση να εκτελεί οποιαδήποτε ενέργεια είναι απαραίτητη για την ενεργοποίηση, επαναρύθμιση, παρακολούθηση και τον έλεγχο μιας υπηρεσίας ή συσκευής. Ωστόσο η σπάταλη χρήση των κλήσεων συστήματος

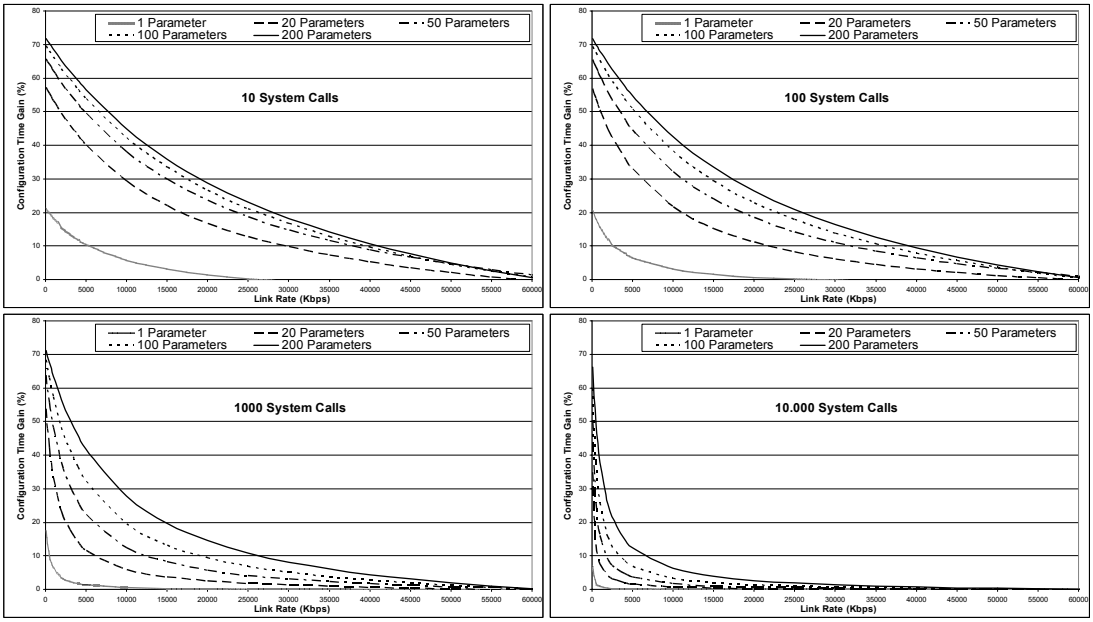
μπορεί να περιοριστεί κατά το σχεδιασμό της διαχειριστικής εφαρμογής του πράκτορα και επομένως να αποδώσει σημαντικά οφέλη στις επιδόσεις του συστήματος.

3.5.11.4. Η αποτίμηση της συμπίεσης

Το σχήμα 3-35 παρουσιάζει το επιπλέον κέρδος στο χρόνο διαχείρισης, που αποφέρει η προσέγγιση της επιλεκτικής συμπίεσης. Ομοίως με τα παραπάνω εμφανίζονται 4 διαγράμματα, καθένα εκ των οποίων αντιστοιχεί σε διαφορετικό αριθμό κλήσεων συστήματος. Επίσης κάθε διάγραμμα περιλαμβάνει 5 διαφορετικές καμπύλες, η καθεμία εκ των οποίων αντιστοιχεί σε διαφορετικό αριθμό παραμέτρων (1, 20, 50, 100 και 200), που εμπλέκονται στη διαχειριστική πράξη (σαν ορίσματα της μεθόδου *GetParameterValues*).

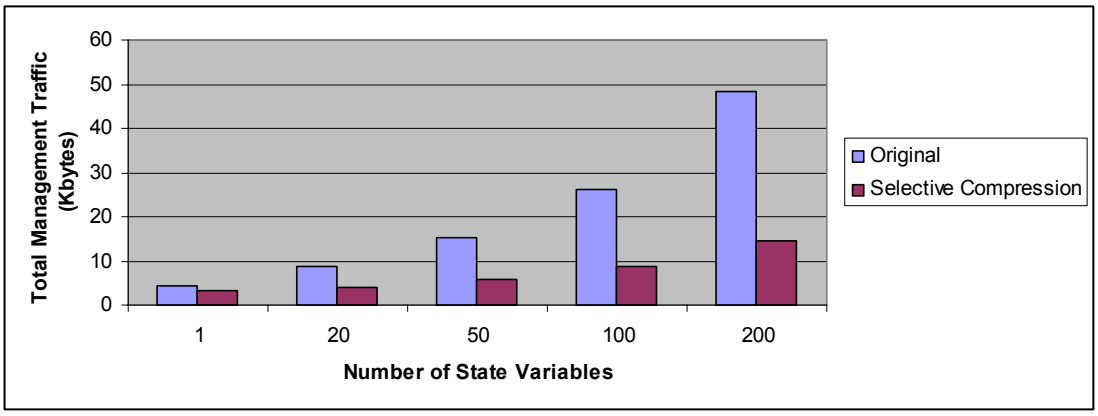
Καταρχάς φαίνεται ότι όσο χαμηλότερο είναι το διαθέσιμο εύρος ζώνης, τόσο μεγαλύτερη είναι η συνεισφορά της συμπίεσης στο χρόνο διαχείρισης. Έτσι στα 100Kbps το ποσοστιαίο κέρδος μπορεί να ξεπερνά ακόμα και το 70% (ο χρόνος διαχείρισης χωρίς συμπίεση είναι υπερτριπλάσιος του αντίστοιχου με συμπίεση). Ωστόσο όσο το διαθέσιμο εύρος ζώνης αυξάνεται τόσο το κέρδος της συμπίεσης εξανεμίζεται. Το γεγονός αυτό αποδεικνύει ότι ο όγκος των SOAP μηνυμάτων διαχείρισης και επομένως ο χρόνος μετάδοσής τους είναι τόσο υψηλός, ώστε να διαδραματίζει τον κυρίαρχο ρόλο στο συνολικό χρόνο διαχείρισης, όταν πρόκειται για σχετικά χαμηλούς ρυθμούς μετάδοσης. Βέβαια όσο αυξάνεται το εύρος ζώνης ο χρόνος μετάδοσης μειώνεται σημαντικά με αποτέλεσμα η συμπίεση να συνεισφέρει σε μικρότερο βαθμό. Ωστόσο είναι σημαντικό να τονιστεί ότι η συμπίεση είναι κερδοφόρα (και επομένως εφαρμόζεται αυτόματα από τον πράκτορα και το διαχειριστή του συστήματος) για πολύ υψηλούς ρυθμούς μετάδοσης που φτάνουν μέχρι και τα 60Mbps για το δεδομένο πείραμα. Ακόμα και στην περίπτωση των πολύ μικρών μηνυμάτων (με 1 μόνο παράμετρο) η συμπίεση είναι κερδοφόρα μέχρι τα 25Mbps. Βέβαια για τη δεδομένη μέθοδο συμπίεσης τα όρια κερδοφορίας καθορίζονται από την πλατφόρμα του πράκτορα και του διαχειριστή. Έτσι για τις πιο αδύναμες πλατφόρμες τα όρια αυτά αναμένονται χαμηλότερα, ενώ για τις ισχυρότερες αναμένονται ακόμα υψηλότερα. Μάλιστα για να επαληθευτεί αυτό τα ίδια πειράματα διεξήχθησαν και με την πιο αδύναμη πλατφόρμα ΠΛΑΤΦ1 (PentiumI στα 133MHz με 64Mbyte RAM) στο ρόλο του πράκτορα διατηρώντας παράλληλα την πλατφόρμα ΠΛΑΤΦ3 στο ρόλο του ACS διαχειριστή. Οι καμπύλες που προέκυψαν είναι πανομοιότυπες με εκείνες του σχήματος 3-35 με τη μόνη διαφορά, ότι τα νέα όρια κερδοφορίας της συμπίεσης είναι αρκετά χαμηλότερα (π.χ. 18Mbps για τις 200 παραμέτρους και 8Mbps για τη 1 παράμετρο).

Ομοίως με την κωδικοποίηση, η συμπίεση φαίνεται να χάνει μεγάλο μέρος της αξίας της όταν η διαδικασία της ενορχήστρωσης είναι χρονοβόρα. Για παράδειγμα, στα 10Mbps και στην περίπτωση, που η υλοποίηση της ενορχήστρωσης αποτελείται από 10 κλήσεις συστήματος το μέγιστο κέρδος στο χρόνο διαχείρισης ξεπερνά το 44%. Αντίθετα στην περίπτωση των 10.000 κλήσεων το μέγιστο κέρδος δεν ξεπερνά το 7%. Είναι λοιπόν πολύ σημαντικό ο χρόνος ενορχήστρωσης να διατηρείται σε χαμηλά επίπεδα προκειμένου να έχουν νόημα οι προτεινόμενες βελτιώσεις.



Σχήμα 3-35. Το ποσοστιαίο χρονικό κέρδος της συμπίεσης πραγματικού χρόνου.

Τέλος, στο σχήμα 3-36 απεικονίζεται ο συνολικός όγκος της κίνησης του δικτύου κατά τη διάρκεια κάθε διαχειριστικής συνεδρίας. Το διάγραμμα παρουσιάζει την κίνηση, που παράγουν οι 5 διαφορετικές συνεδρίες (για τη λήψη των τιμών 1, 20, 50, 100 και 200 μεταβλητών κατάστασης). Μάλιστα σε κάθε συνεδρία συμπεριλαμβάνονται και τα *Inform* μηνύματα καθώς αυτά είναι απαραίτητα για τη CWMP συνεδρία. Όπως αναμένονταν στα πολύ μικρά μηνύματα (1 παράμετρος) η συμπίεση αποφέρει ένα μικρό κέρδος (28%) ενώ ήδη στις 20 παραμέτρους το κέρδος αγγίζει το 50% (υποδιπλασιασμός του όγκου κίνησης). Τέλος, για ένα μεγάλο αριθμό παραμέτρων (200 παράμετροι) το κέρδος φτάνει στο 70% (υποτριπλασιασμός του όγκου κίνησης).



Σχήμα 3-36. Ο συνολικός όγκος της κίνησης διαχείρισης.

3.6. Συμπεράσματα – Παρατηρήσεις

Η υιοθέτηση του πρωτοκόλλου γενικής χρήσης SOAP για την κωδικοποίηση των κλήσεων απομακρυσμένων μεθόδων, σε συνδυασμό με την περιττότητα της XML στην έκφραση της πληροφορίας, έχουν σαν συνέπεια την κατακόρυφη αύξηση του όγκου της παραγόμενης κίνησης κατά τη διαχείριση. Μάλιστα το γεγονός ότι ο όγκος αυτός είναι από 2 (για μεγάλο αριθμό διαχειριζόμενων παραμέτρων) μέχρι και 10 φορές μεγαλύτερος (για μια παράμετρο) από τον αντίστοιχο του SNMP δικαιώνει, σε πρώτη φάση τις ανησυχίες εκείνων, που αρχικά αντέδρασαν στις SOAP/XML/HTTP διαχειριστικές λύσεις. Εξάλλου όταν κανείς κοιτάξει το πρόβλημα από την οπτική γωνία της διαχείρισης των συνδρομητικών συσκευών, μέσω του δημοσίου δικτύου αντιλαμβάνεται ότι αυτό γίνεται ακόμα εντονότερο. Το περιορισμένο εύρος ζώνης, που ορίζει κάθε φορά η τεχνολογία πρόσβασης σε συνδυασμό με τον υψηλό όγκο των SOAP μηνυμάτων διαχείρισης μπορεί να αυξήσει δραματικά το συνολικό χρόνο ρύθμισης και διαχείρισης συσκευών και υπηρεσιών. Επίσης η «φλύαρη» αυτή επικοινωνία του διαχειριστή με τους πράκτορες διαχείρισης των συσκευών καταναλώνει πολύτιμους πόρους από το δίκτυο πρόσβασης, χωρίς παράλληλα να αποφέρει πρόσθετα κέρδη στους παρόχους.

Η προφανής λύση στο παραπάνω πρόβλημα ήταν η εγκατάλειψη του SOAP πρωτοκόλλου και η διερεύνηση ενός νέου, ειδικά σχεδιασμένου για τη διαχείριση. Ωστόσο το SOAP μαζί με την XML αποτελούν πρότυπα της W3C και σήμερα συναντώνται σχεδόν παντού. Το γεγονός αυτό σε συνδυασμό με την ευρεία αποδοχή, που έχει γνωρίσει το SOAP τα τελευταία πέντε χρόνια από την αγορά και την προτυποποίηση της διαχείρισης (UPnP, CWMP, NetConf) καθιστά ιδιαίτερα δύσκολη και χρονοβόρα την επικράτηση ενός νέου προτύπου. Για το λόγο αυτό η παρούσα εργασία πρότεινε μια λύση η οποία δεν αλλάζει τίποτα στον τρόπο με το οποίο μοντελοποιείται, δομείται και κωδικοποιείται η πληροφορία διαχείρισης, παρά μόνο στον τρόπο που μεταφέρεται. Έτσι προτάθηκε η συμπίεση των SOAP μηνυμάτων διαχείρισης σε πραγματικό χρόνο. Τα πειραματικά αποτελέσματα αποδεικνύουν ότι όταν ο Lempel-Ziv αλγόριθμος υλοποιείται με στόχο την υψηλή ταχύτητα εκτέλεσης (*LZO* βιβλιοθήκη), η συμπίεση σε πραγματικό χρόνο των SOAP μηνυμάτων μπορεί να ελαττώσει μέχρι και κατά 3 φορές την παραγόμενη κίνηση αλλά και το συνολικό χρόνο διαχείρισης. Επιπλέον προσδιορίστηκαν οι συνθήκες κάτω από τις οποίες θα πρέπει να εφαρμόζεται η συμπίεση προκειμένου να είναι πάντα κερδοφόρα. Έτσι προτάθηκε και υλοποιήθηκε η ιδέα της *επιλεκτικής συμπίεσης*, κατά την οποία ο πράκτορας εξετάζει κάθε φορά τις παραπάνω συνθήκες και εφαρμόζει επιλεκτικά τη συμπίεση στους SOAP φακέλους. Μάλιστα το γεγονός ότι η συμπίεση και η SOAP/XML κωδικοποίηση συμβαίνουν πάντα διαδοχικά οδήγησε στην απλοποίηση της διαδικασίας λήψης της απόφασης για την εφαρμογή ή μη της συμπίεσης. Όταν η συσκευή, ο διαχειριστής και η υλοποίηση του αλγορίθμου συμπίεσης είναι δεδομένα (δηλαδή όταν είναι γνωστές οι τυπικές τιμές των ταχυτήτων συμπίεσης / αποσυμπίεσης) το μόνο που χρειάζεται να γνωρίζει η μονάδα απόφασης είναι το μέγεθος του προς συμπίεση μηνύματος και το μέσο ρυθμό μετάδοσης. Η απλοποίηση αυτή εξαλείφει την ανάγκη πολύπλοκου μεσισμικού για συνεχή παρακολούθηση των πόρων επεξεργασίας του συστήματος (για την εκτίμηση των ταχυτήτων συμπίεσης / αποσυμπίεσης) και κάνει

πραγματικά εφαρμόσιμη την επιλεκτική συμπίεση ακόμη και σε συσκευές χαμηλών δυνατοτήτων. Επίσης τα πειραματικά αποτελέσματα αποδεικνύουν ότι η υψηλή ταχύτητα εκτέλεσης του αλγορίθμου συμπίεσης επιτρέπει την εφαρμογή του σε μια ευρεία γκάμα συσκευών (ξεκινώντας με ισχύ ισοδύναμη ή και μικρότερη ενός Pentium I επεξεργαστή) και τεχνολογιών πρόσβασης (με ρυθμούς μετάδοσης, που φτάνουν μέχρι και ορισμένες δεκάδες Mbps).

Επίσης η κλασσική και πρότυπη μέθοδος DOM αποδεικνύεται ακατάλληλη για τη γραμματική ανάλυση των SOAP/XML φακέλων διαχείρισης εξαιτίας των μεγάλων υπερκεφαλισμών επεξεργασίας που εμφανίζει. Μάλιστα οι υπερκεφαλισμοί αυτοί έχουν σαν άμεση συνέπεια και την αύξηση του συνολικού χρόνου διαχείρισης η οποία γίνεται ιδιαίτερα αισθητή σε συσκευές με χαμηλή δυνατότητα επεξεργασίας. Επιπλέον οι μεγάλες απαιτήσεις της DOM μεθόδου σε δυναμική μνήμη RAM αποτελούν ένα ακόμα πρόβλημα. Το πρόβλημα αυτό μπορεί να γίνει ιδιαίτερα κρίσιμο σε περιπτώσεις μαζικής παρακολούθησης ή ρύθμισης των παραμέτρων μιας συσκευής χαμηλού κόστους, με περιορισμένα αποθέματα μνήμης. Η προσέγγιση της χρήσης γραμματικών αναλυτών, ειδικά σχεδιασμένων για το κάθε πρωτόκολλο διαχείρισης δίνει εντυπωσιακά αποτελέσματα. Αυτά συνίστανται στη βελτίωση του χρόνου διαχείρισης (μέχρι και 4 φορές μικρότερος χρόνος) αλλά και στη μείωση των απαιτήσεων του πράκτορα σε μνήμη (μέχρι και 19 φορές μικρότερη χρήση μνήμης). Βέβαια το τίμημα που πληρώνει κανείς για τη χρήση των ειδικών αναλυτών έναντι των DOM είναι το κόστος υλοποίησης για κάθε νέο πρωτόκολλο διαχείρισης και το κόστος της συντήρησής τους για κάθε νέα έκδοση του πρωτοκόλλου. Επιπρόσθετα η θεωρητική ανάλυση του τρόπου λειτουργίας των SAX αναλυτών δείχνει καλύτερες επιδόσεις από εκείνες των DOM αλλά χειρότερες από εκείνες των ειδικών αναλυτών. Από την άλλη πλευρά οι SAX αναλυτές είναι γενικής χρήσης. Ίσως λοιπόν η προσέγγιση των SAX αναλυτών να αποτελεί μια λύση που συνδυάζει τις επιδόσεις με το χαμηλό κόστος υλοποίησης. Ωστόσο αυτό είναι ένα θέμα προς περαιτέρω διερεύνηση, καθώς για να αποφανθεί κανείς με ασφάλεια χρειάζονται πειραματικά αποτελέσματα όμοια με εκείνα της παρούσας εργασίας για τις άλλες δυο προσεγγίσεις.

Μια ακόμα σημαντική παρατήρηση είναι ότι ενώ η προσέγγιση της επιλεκτικής συμπίεσης εμφανίζει καλύτερα χρονικά αποτελέσματα για μικρούς ρυθμούς μετάδοσης, η προσέγγιση των ειδικών γραμματικών αναλυτών εμφανίζει τα καλύτερα αποτελέσματα για μεγάλους ρυθμούς μετάδοσης. Θα μπορούσε λοιπόν κανείς να ισχυριστεί ότι η παρούσα εργασία πρότεινε δυο προσεγγίσεις, που αλληλοσυμπληρώνονται παρέχοντας έτσι βελτίωση για κάθε τεχνολογία πρόσβασης (v.90/.v34, ISDN, PowerLine, DBS, xDSL, FTTH κ.α.). Τέλος θα πρέπει να σημειωθεί ότι οι προγραμματιστές των πρακτόρων διαχείρισης θα πρέπει να είναι ιδιαίτερα προσεκτικοί στην ποιότητα υλοποίησης των μεθόδων ενορχήστρωσης. Μια «πρόχειρη» υλοποίηση, που δεν εστιάζει στην ελάχιστη χρήση των πόρων επεξεργασίας της συσκευής θα μπορούσε να αυξήσει σημαντικά το χρόνο διαχείρισης και να υποβαθμίσει την αξία των προτεινόμενων λύσεων.

3.7. Αναφορές

[1] The Internet Engineering Task Force, http://www.ietf.org.

[2] J. Schoenwaelder, A. Pras, and J.-P. Martin-Flatin, "On the Future of Internet Management Technologies," IEEE Communications Magazine, vol. 41, no. 10, Oct. 2003.

[3] J. Schonwalder, "Overview of the 2002 IAB Network Management Workshop," May 2003, IETF RFC 3535.

[4] The Internet Research Task Force, http://www.irtf.org.

[5] Pras, T. Drevers, R.van de Meent, D. Qurtel, "Comparing the Performance of SNMP and Web Services-Based Management", IEEE eTransactions on Network and Service Management, Fall 2004.

[6] G. Pavlou, Paris Flegkas, Stelios Gouveris and Antonio Liotta, "On Management Technologies and the Potential of Web Services," IEEE Communications Magazine, vol. 42, no. 7, July 2004.

[7] R. Neisse, R. L. Vianna, L. Z. Granville, M. J. B. Almeida, and L. M. R. Tarouco, "Implementation and Bandwidth Consumption Evaluation of SNMP to Web Services Gateways," in Proceedings of NOMS2004, 2004.

[8] M.-J.Choi, J. Hong, H-T.Ju, "XML-Based Network Management for IP Networks", ETRI Journal, vol 25, no 6, December 2003

[9] JP Martin-Flatin, "Web-Based Management of IP Networks and Systems", Publisher: John Wiley and Sons, 2003.

[10] UPnP Forum, "UPnP device architecture 1.0", December 2003, http://www.upnp.org

[11] DSL Forum, "CPE WAN management protocol", Technical Report 069, May 2004.

[12] International Organization for Standardization, "Information processing systems - Open Systems Interconnection, Specification of Abstract Syntax Notation One (ASN.1)," December 1987, International Standard 8824.

[13] International Organization for Standardization, "Information processing systems - Open Systems Interconnection, Specification of Abstract Syntax Notation One (ASN.1)," December 1987, International Standard 8825.

[14] J. Case et al., "A Simple Network Management Protocol (SNMP)," May 1990, IETF RFC 1157.

[15] R. Presuhn et al., "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)," December 2002, IETF RFC 3416.

[16] "The Web Services Description Working Group," http://www.w3. org/2002/ws/desc/.

[17] D. Davis and M. Parashar, "Latency Performance of SOAP Implementations," in Proceedings of IEEE Cluster Computing and the GRID 2002, 2002, pp. 407–412.

[18] K. Chiu, M. Govindaraju and R. Bramley, "Investigating the Limits of SOAP Performance for Scientific Computing," in Proceedings of 11th. IEEE International Symposium on High Performance Distributed Computing, 2002, pp. 246–254.

[19] A. Ng, S. Chen, and P. Greenfield, "An Evaluation of Contemporary Commercial SOAP Implementations," in Proceedings of the 5th Australasian Workshop on Software and System Architectures (AWSA 2004), 2004, pp. 64–71.

[20] NET-SNMP libraries and tools, http://net-snmp.sourceforge.net.

[21] P. Deutsch and J. L. Gailly. ZLIB Compressed Data Format Specification version 3.3, May 1996. IETF RFC 1950.

[22] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser (Eds.) "Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)", IETF RFC 1907, January 1996.

[23] K. McCloghrie and F. Kastenholz, "The Interfaces Group MIB using SMIv2", IETF RFC 2233,November 1997.

[24] W3C, "SOAP Version 1.2 Part 0: Primer," June 2003.

[25] Motorola, MPC850: PowerQUICC™ Integrated Comm. Processor, http://ewww.motorola.com /webapp/sps/site/prod_summary.jsp?code=MPC850.

[26] Object Management Group, "The Common Object Request Broker: Architecture and Specification (CORBA)," v. 2.0, 1995.

[27] K. McCloghrie, "SNMPv2 Management Information Base for the Transmission Control Protocol using SMIv2", IETF RFC 2012 , November 1996.

[28] Orbacus, CORBA platform for C++ and Java, http://www.orbacus.com.

[29] Systinet WASP, http://www.systinet.com/.

[30] NET-SNMP libraries and tools, http://net-snmp.sourceforge.net.

[31] R. v. Engelen, "gSOAP web services toolkit," http://www.cs.fsu.edu/engelen/soap.html.

[32] R. v. Engelen and K. A. Gallivany, "The gSOAP Toolkit for Web Services and Peer-To-Peer Computing Networks," in Proceedings of IEEE Cluster Computing and the GRID 2002, 2002, pp. 128–135.

[33] W3C, "Document Object Model (DOM) Level 1 Specification," W3C Recommendation, Oct. 1998.

[34] K. McCloghrie and F. KastenHolz, "The Interfaces Group MIB," June 2000, IETF RFC 2863.

[35] G. Pavlou, P. Flegkas, and S. Gouveris, "Performance Evaluation of Web Services as Management Technology," Presentation at the 15th IRTF-NMRG meeting, January 2004.

[36] "The XML Protocol Working Group," http://www.w3.org/ 2000/xp/Group/.

[37] "The Web Services Description Working Group," http://www.w3. org/2002/ws/desc/.

[38] T. Drevers, R. van de Meent, and A. Pras, "Prototyping Web Services based Network Monitoring," in Proceedings of 10th Open European Summer School School and IFIP WG 6.3 Workshop (EUNICE 2004), June 2004, pp. 135–142.

[39] "Zlib homepage," http://www.zlib.org.

[40] "The Evolution Of SNMP Working Group," http://www.ietf.org/html.charters/OLD/eos-charter.html.

[41] IETF, "Network Configuration (Netconf)," http://www.ietf.org/html.charters/netconf-charter.html.

[42] M.-J. Choi et al., "XML-Based Configuration Management for IP Network Devices," IEEE Communications Magazine, vol. 42, no. 7, July 2004.

[43] The Digital Subscriber Forum, http://www.dslforum.org

[44] DSL Forum, "Internet Gateway Device v.1.1 data model", Technical Report. 098, September 2005.

[45] UPnP Forum, "UPnP device architecture 1.0", December 2003, http://www.upnp.org.

[46] T. Berners-Lee, R. Fielding, J. Mogul, H. Frystyk, "Hypertext Transfer Protocol -HTTP/1.1," RFC 1945, IETF HTTP WG, May 1996

[47] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol -HTTP/1.1," RFC 2616, IETF HTTP WG, June 1999.

[48] .Y. Wiseman and K.Schwan., "Efficient End-to-End Data Exchange Using Configurable Compression," IEEE Proceedings. 24th International Conferenceon Distributed. Computing. Systems, 2004.

[49] E. Jeannot, et. al., "Adaptive Online Data Compression", Proceedings on IEEE High Performance Distributed Computing, Scotland, July 2002.

[50] M. Wolf, et. al, "Smart Pointers: Personalized Scientific Data Portals in Your Hand", Proceedings of Supercomputing 2002, November 2002.

[51] V. Oleson, et. al., "Operational Information Systems, An Example from the Airline Industry", WIESS 2000, November 2000.

[52] D. Huffman, "A method for the Construction of Minimum Redundancy Codes", Proceeding of the IRE, vol.40, pp. 1098-1101, 1952.

[53] G.K. Wallace, "The JPEG Still Picture Compression Standard", Communication of the ACM, vol.34, pp. 3-44, 1991.

[54] Information Technology, "Digital Compression and Coding of Continuous-Tone Still Images – Requirements and Guidelines", International Standard ISO/IEC, 10918-1, 1993.

[55] I. Witten et.al, "Arithmetic Coding for Data Compression", Communications of the ACM, vol.30, pp. 520-540, 1987.

[56] P.G. Howard and J.S. Vitter, "Arithmetic Coding for Data Compression", Proceedings of the IEEE, 82(6), pp.857-865, 1994.

[57] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression", IEEE Transactions on Information Theory, IT-23, pp.337-343,1977.

[58] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding", IEEE Transactions on Information Theory, IT-24, pp.530-536,1978.

[59] WinZip, Nico Mak Computing, Inc. Mansfield, CT, USA, 1998.

[60] gzip, Free Software Foundation, Inc, 675 Mass Ave. Cambridge, MA, USA, 1991.

[61] M. Burrows and D. Wheeler, "Block sorting Lossless Data Compression Algorithm", System research center, research report 124, Digital System research Center, Palo Alto,CA, 1994.

[62] M.R. Nelson "Data Compression with the Burrows Wheeler Transformation", Dr. Dobb's Journal, pp.46-50, 1996.

[63] A. Bookstein and S.T Klein, "Is Huffman coding dead?", Journal of Computing, vol. 50, pp. 279-296, 1993.

[64] R.P. Brent, "A linear algorithm for data compression", Australian Computer Journal, 19 (2), pp.64-68, May 1987.

[65] SGI IRIX Freeware distribution, 1600 Ampitheatre Pkwy. Mountain View, CA, USA, February 2003.

[66] M. F. X. J. Oberhumer, LZO library, http://www.oberhumer.com, 2002.

[67] Xbox homepage http://www.xbox.com/

[68] PlayStation homepage http://www.playstation.com/

[69] http://www.via.com.tw/en/products/mainboards/mini_itx/ epia_pd/index.jsp#spec.

[70] Linux homepage: http://www.linux.org

[71] Simple API for XML, http://www.saxproject.org/, The SAX project.

[72] Netherlabs BV, "Linux Advanced Routing and Traffic Control HOWTO," Oct. 29, 2003.

[73] Ethereal, a network protocol analyzer, http://www.ethereal.com.

[74] - http://www.via.com.tw/en/products/processors/specs.jsp.

Κεφάλαιο 4
Περιορισμός του Κόστους των SOAP/XML/HTTP Προτύπων Απομακρυσμένης Διαχείρισης

4.1. Εισαγωγή

Η σύγχρονη τάση ενσωμάτωσης δικτυακών τεχνολογιών στις συνδρομητικές συσκευές ανοίγει το δρόμο για σημαντικές βελτιώσεις στην εμπειρία των τελικών χρηστών. Παράλληλα δημιουργεί νέες επιχειρηματικές ευκαιρίες για τους παρόχους, οι οποίοι μπορούν να εκμεταλλευτούν τις πλούσιες δυνατότητες των νέων συσκευών για να προσφέρουν διάφορες υπηρεσίες μέσω του διαδικτύου. Οι συνδρομητές πρόκειται να έχουν τη δυνατότητα πρόσβασης σε μια ευρεία γκάμα πολυμεσικών υπηρεσιών που θα παρέχονται από το τοπικό (local services) αλλά συνεχώς αυξανόμενα και από το δημόσιο δίκτυο (remote services).

Ένα από τους παράγοντες κλειδιά στην μαζική εκμετάλλευση των δικτυακών υπηρεσιών είναι το επίπεδο ευκολίας όσον αφορά στην αρχική εγκατάσταση, στη ρύθμιση αλλά και στη διαχείριση των συσκευών που φιλοξενούν τις υπηρεσίες. Η ελάχιστη ή ακόμα και μηδενική παρέμβαση του χρήστη είναι βασική απαίτηση. Αυτή τη στιγμή το UPnP (Universal Pug and Play) πρωτόκολλο είναι η πιο διαδομένη τεχνολογία διαλειτουργικότητας και διευθετεί θέματα αυτόματης εγκατάστασης και ρύθμισης συσκευών και υπηρεσιών. Χρησιμοποιεί υπάρχοντα διαδικτυακά πρότυπα (TCP, UDP, HTTP, SOAP και XML) και ορίζει μηχανισμούς για την ανίχνευση, τον έλεγχο, την περιγραφή, την ενημέρωση συμβάντων και την παρουσίαση των συσκευών. Ωστόσο το UPnP πρωτόκολλο σχεδιάστηκε με στόχο να λύσει προβλήματα διαχείρισης των τοπικών δικτύων. Τα μηνύματα εκπομπής για την ανίχνευση συσκευών (SSDP) και την ενημέρωση συμβάντων (GENA), η χρήση του αναξιόπιστου UDP πρωτοκόλλου αλλά και η έλλειψη μηχανισμών ασφαλείας είναι ορισμένοι από τους βασικότερους λόγους, που καθιστούν το UPnP ακατάλληλο να υποστηρίξει τη ρύθμιση και διαχείριση των συσκευών μέσω του δημοσίου δικτύου.

Αντίθετα με το UPnP η πρωτοβουλία για την πύλη ανοιχτών υπηρεσιών (OSGi – Open Services Gateway initiative) φαίνεται να είναι ίσως η πιο διαδεδομένη αρχιτεκτονική για τη διανομή, εγκατάσταση και ενεργοποίηση των απομακρυσμένων υπηρεσιών. Καθορίζει μια πρότυπη πλατφόρμα, όπου διάφορα στοιχεία λογισμικού (bundles) μπορούν να προστεθούν, να επικαιροποιηθούν ή και να αφαιρεθούν κατά τη λειτουργία μιας συσκευής χωρίς να την επηρεάζουν. Το OSGi είναι ένα πλαίσιο εργασίας βασισμένο στην Java και εστιάζει

περισσότερο στις υπηρεσίες παρά στις πραγματικές συσκευές, μέσω των οποίων αποδίδονται αυτές οι υπηρεσίες. Έτσι δεν επιλύει θέματα σχετικά με την ανίχνευση, την ενημέρωση συμβάντων, τον έλεγχο και τη μοντελοποίηση των συσκευών με στόχο την απομακρυσμένη διαχείρισή τους.

Όπως φάνηκε και στο δεύτερο κεφάλαιο μέχρι σήμερα είχαν πραγματοποιηθεί αρκετές εργασίες (εξελικτικές λύσεις) με στόχο την ολοκλήρωση του SNMP πρωτοκόλλου με διάφορες XML διαχειριστικές λύσεις. Το γεγονός ότι πριν το 2004 δεν υπήρχε κάποιο ώριμο πρότυπο XML διαχείρισης, σε συνδυασμό με την αδυναμία του SNMP να καλύψει τις ανάγκες για μαζική διαχείριση των επόμενης γενιάς συνδρομητικών συσκευών και υπηρεσιών (βλέπε δεύτερο κεφάλαιο) είχε ως αποτέλεσμα την αποτυχία των εξελικτικών λύσεων σε επίπεδο προτυποποίησης. Από την άλλη πλευρά η προσπάθεια των οργανισμών προτυποποίησης για μια ενιαία XML τεχνολογία διαχείρισης (ριζοσπαστικές λύσεις) έχει αποδώσει καρπούς τα τελευταία τρία χρόνια, ιδίως με τα CWMP και NetConf πρωτόκολλα του DSL φόρουμ και της IETF, αντίστοιχα. Το γεγονός αυτό παρότρυνε το συγγραφέα να συμβάλλει στην ταχεία εξάπλωση των προτυποποιημένων XML λύσεων διαχείρισης και πιο συγκεκριμένα εκείνης του πρωτοκόλλου CWMP. Σήμερα το CWMP είναι ίσως το ωριμότερο XML πρότυπο διαχείρισης, καθώς υποστηρίζεται ήδη από ένα τμήμα της αγοράς των δικτυακών πυλών, ενώ παράλληλα γίνονται εκτεταμένες προσπάθειες για την εφαρμογή του στον υπόλοιπο συνδρομητικό εξοπλισμό. Μάλιστα η παρούσα διατριβή επιχειρεί να πετύχει την ταχεία εξάπλωση του CWMP πρωτοκόλλου συμβάλλοντας με δύο τρόπους. Ο πρώτος τρόπος είναι μέσω προσεγγίσεων, που μειώνουν το κόστος εφαρμογής των παραπάνω λύσεων, ενώ ο δεύτερος είναι με τη επέκταση του εύρους δράσης της CWMP τεχνολογίας διαχείρισης. Στο παρόν κεφάλαιο αναλύεται ο πρώτος τρόπος, ενώ ο δεύτερος παρουσιάζεται στο πέμπτο κεφάλαιο.

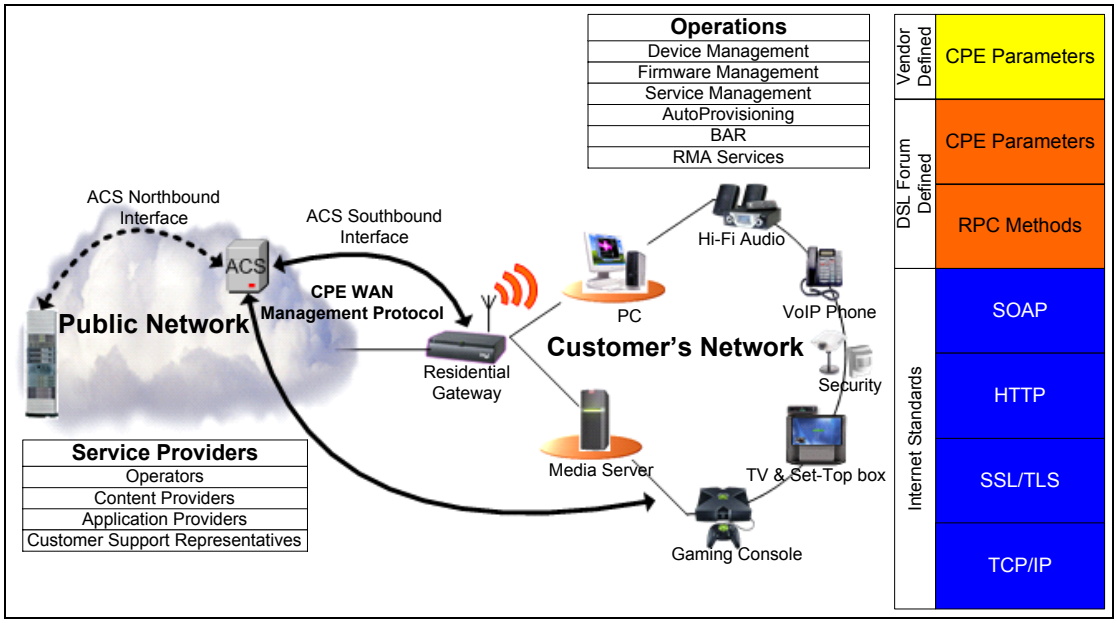
Με κύριο στόχο τη διανομή και ενεργοποίηση των απομακρυσμένων υπηρεσιών, το DSL φόρουμ προσπαθεί να καλύψει τα παραπάνω κενά προδιαγράφοντας το πρωτόκολλο απομακρυσμένης διαχείρισης συνδρομητικού εξοπλισμού (CPE WAN Management Protocol - CWMP) [1]. Το CWMP φαίνεται να είναι μια πολλά υποσχόμενη λύση για τη ρύθμιση και διαχείριση των συνδρομητικών συσκευών (CPEs – Customer Premises Equipment) όχι μόνο σε DSL περιβάλλοντα αλλά και σε οποιοδήποτε άλλο IP περιβάλλον, που μπορεί να υποστηρίξει την παροχή πολλαπλών υπηρεσιών στο συνδρομητή. Τυπικά ο όρος CPE περιλαμβάνει δικτυακές πύλες, προσωπικούς υπολογιστές, κάμερες, ηχοσυστήματα, Video/VoIP τηλέφωνα, συνδρομητικούς αποκωδικοποιητές καθώς και οποιασδήποτε άλλης δικτυωμένης συσκευής. Επιπλέον μια οικογένεια από συσχετιζόμενα πρότυπα για την τοπική (βασισμένο στο UPnP [2,3]) και απομακρυσμένη ρύθμιση των CPEs επεκτείνει το εύρος δράσης και τη σημασία των εργασιών προτυποποίησης του DSL forum.

Παρακάτω δίνεται μια συνοπτική εικόνα της CWMP αρχιτεκτονικής καθώς επίσης και το όραμα του DSL φόρουμ για την απομακρυσμένη διαχείριση κάθε είδους συνδρομητικού εξοπλισμού, μέσω της εκτεταμένης χρήσης του CWMP πρωτοκόλλου. Στη συνέχεια προτείνεται η χρήση της τεχνολογίας XML για την αναπαράσταση των προτυποποιημένων μοντέλων διαχείρισης των συνδρομητικών συσκευών. Επίσης προτείνεται μια σειρά

διαδικασιών, η οποία στηριζόμενη στην παραπάνω αναπαράσταση επιτυγχάνει την αυτόματη παραγωγή κώδικα για τους CWMP πελάτες (πράκτορες), τη δια βίου συντήρηση του κώδικα αυτού και τις δοκιμές συμμόρφωσης των συσκευών στο εν λόγω πρωτόκολλο. Η παρούσα προσέγγιση στηρίζεται στο πρότυπο και στη σημειογραφία τελείας (dot notation) που αντιστοιχεί στην ιεραρχική δομή κάθε μοντέλου συσκευής. Η διαδικασία δίνεται σε συγκεκριμένα βήματα και για κάθε ένα από αυτά προτείνεται ένα αντίστοιχο εργαλείο. Στη συνέχεια αποτιμάται η προτεινόμενη σουίτα λογισμικού χρησιμοποιώντας σαν παράδειγμα το CWMP πελάτη για μια δικτυακή πύλη.

4.2. Το περιβάλλον διαχείρισης και οι εργασίες προτυποποίησης

Το σχήμα 4-1 δίνει το περιβάλλον με το οποίο ασχολείται η παρούσα ενότητα και στο οποίο απευθύνεται το CWMP πρότυπο. Σύμφωνα με την αρχιτεκτονική του CWMP (σχήμα 4-1), ένας υπάρχει ένας εξυπηρετητής αυτόματης ρύθμισης (Auto-Configuration Server – ACS), που φιλοξενείται είτε στο δίκτυο πρόσβασης, είτε στο δίκτυο κορμού και είναι επιφορτισμένος με την εφαρμογή των αρχικών ρυθμίσεων και την περαιτέρω διαχείριση ενός μεγάλου αριθμού συνδρομητικών συσκευών, μέσω του CWMP πρωτοκόλλου. Ο ACS διαχειριστής έχει δύο διεπαφές που ονομάζονται "Northbound" (Βόρειο σύνορο) και "Southbound" (Νότιο σύνορο). Η Southbound διεπαφή εξυπηρετεί την επικοινωνία του ACS με τις πύλες και σχετίζεται με την παρούσα εργασία. Αντίθετα η Northbound διεπαφή εξυπηρετεί την ανταλλαγή της πληροφορίας διαχείρισης μεταξύ του ACS και των παρόχων, αφού οι τελευταίοι είναι εκείνοι που προσφέρουν τις υπηρεσίες και εφαρμόζουν τις υψηλού επιπέδου πολιτικές διαχείρισης. Η διεπαφή αυτή δεν θα απασχολήσει την παρούσα εργασία.

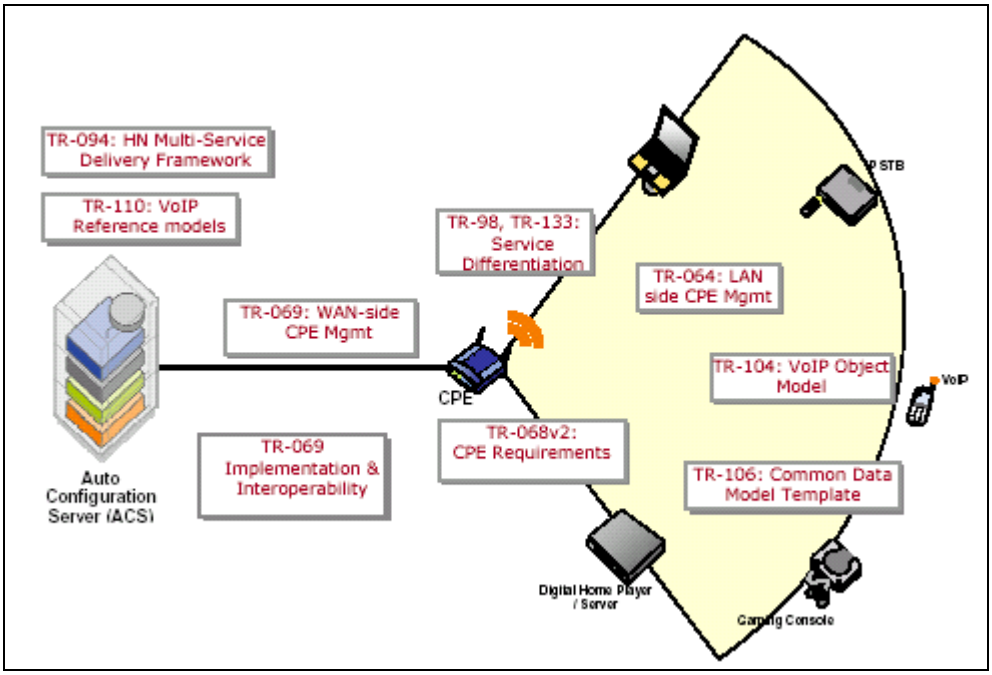


Σχήμα 4-1.Το περιβάλλον διαχείρισης και η στοίβα πρωτοκόλλων του CWMP.

Οι δυνατότητες που προσφέρει το CWMP πρωτόκολλο αναμένεται να απασχολήσουν διάφορες επιχειρηματικές οντότητες. Πέραν της διαχείρισης στα δικτυακά στοιχεία του συνδρομητή, οι πάροχοι θα μπορούν να εκτελούν λειτουργίες που σχετίζονται με τη χρέωση,

τη διαχείριση των σφαλμάτων και την απογραφή του εξοπλισμού. Επίσης τα τμήματα υποστήριξης πελατών θα μπορούν να έχουν ασφαλή πρόσβαση στις συνδρομητικές συσκευές, να ανιχνεύουν τις διάφορες δυσλειτουργίες και να τις διορθώνουν. Μάλιστα θα μπορούν να αλλάζουν τις υπάρχουσες ρυθμίσεις και να μεταφορτώνουν νέο λογισμικό στις συσκευές, προκειμένου να ενεργοποιούν σε αυτές νέες δυνατότητες. Υπηρεσίες όπως εκείνη της εξουσιοδότησης επιστρεφόμενου υλικού (RMA - Returned Material Authorization) και της επιχειρησιακής ανάλυσης και καταγραφής (BAR - Business Analytics and Reporting) θα είναι επίσης δυνατές. Όλα τα παραπάνω θα μπορούν να γίνονται είτε σε επίπεδο συσκευής, είτε μαζικά, για ένα σύνολο συσκευών.

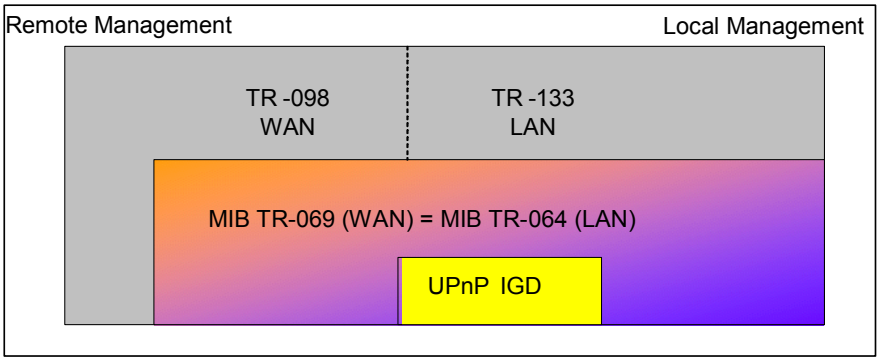
Από το Μάιο του 2004, που εκδόθηκε το CWMP πρωτόκολλο μαζί με το πρώτο μοντέλο δικτυακής πύλης [1] το DSL φόρουμ έχει κάνει σημαντική πρόοδο στη μοντελοποίηση των διαφόρων συνδρομητικών συσκευών με στόχο την απομακρυσμένη διαχείρισή τους μέσω του CWMP. Μέχρι σήμερα έχει ήδη προτυποποιηθεί ένα βελτιωμένο μοντέλο δικτυακής πύλης [4] καθώς και η υπηρεσία φωνής πάνω από το IP πρωτόκολλο (VoIP) [5]. Παράλληλα υπάρχει σημαντική πρόοδος στη μοντελοποίηση του συνδρομητικού αποκωδικοποιητή (Set-Top Box - STB) [6] καθώς και στην επέκταση της μοντελοποίησης της δικτυακής πύλης για τη διαχείριση λειτουργιών όπως είναι η πολυεκπομπή (multicast) και το τείχος προστασίας (firewall). Επίσης κάποια πρώτα βήματα έχουν γίνει για τη μοντελοποίηση των παιχνιδομηχανών και των εξυπηρετητών πολυμέσων [7]. Το σχήμα 4-2 περιγράφει την υπάρχουσα κατάσταση όσον αφορά στη μοντελοποίηση των συσκευών με στόχο την απομακρυσμένη αλλά και τοπική διαχείρισή τους.



Σχήμα 4-2.Το εύρος προτυποποίησης για τη διαχείριση του συνδρομητικού εξοπλισμού.

Όσον αφορά στη διαχείριση της δικτυακής πύλης, το DSL forum χρησιμοποιεί σαν βάση της μοντελοποίησης εκείνη του UPnP [8]. Οι UPnP μοντελοποιημένες λειτουργίες έχουν εμπλουτιστεί σημαντικά από το DSL φόρουμ και δίνονται στο TR-064 κείμενο [2] για την

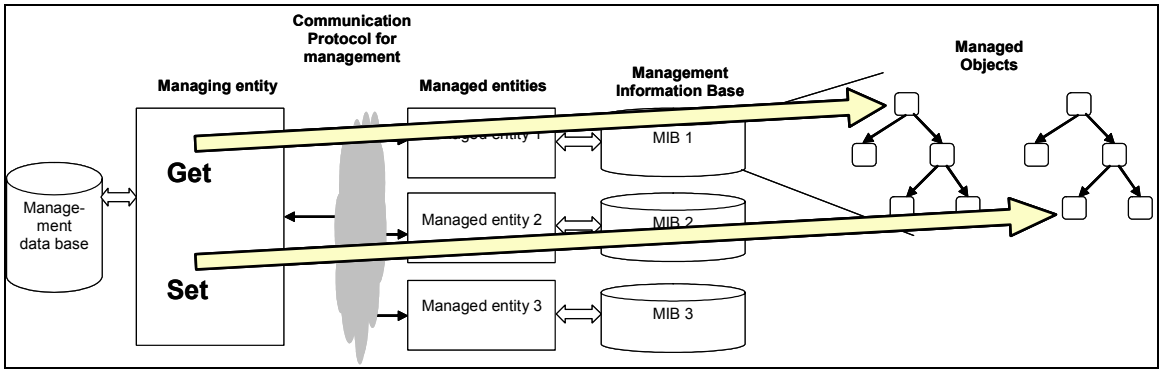
τοπική διαχείριση και στο TR-069 κείμενο για την απομακρυσμένη. Μάλιστα ενώ για την απομακρυσμένη διαχείριση το DSL φόρουμ ορίζει το CWMP πρωτόκολλο, στην τοπική διαχείριση χρησιμοποιεί τον ίδιο το μηχανισμό του UPnP [9] για την ανίχνευση και ρύθμιση της πύλης. Η μοντελοποίηση αυτή έχει εμπλουτιστεί περαιτέρω με παραμέτρους για τη διαχείριση της ποιότητα υπηρεσίας (Quality Of Service – QoS) και δίνεται τόσο για την τοπική (TR-133) [3] όσο και για την απομακρυσμένη [4] διαχείριση. Το σχήμα 4-3 δίνει τις σχέσεις των προαναφερθέντων εργασιών μοντελοποίησης.



Σχήμα 4-3.Το πεδίο δράσης των εργασιών προτυποποίησης για τη μοντελοποίηση της δικτυακής πύλης

4.3. Η μοντελοποίηση των συσκευών

Η διαχείριση των συσκευών σε συνδρομητικά περιβάλλοντα εστιάζει στην αλληλεπίδραση μεταξύ της διαχειρίζουσας (Managing Entity) και της διαχειριζόμενης (Managed Entity) οντότητας δια μέσου των συναλλαγών (Transactions) αλλά και άλλων συμπληρωματικών λειτουργιών. Οι συναλλαγές δεν είναι τίποτα άλλο παρά οι πράξεις ασφαλούς ανάγνωσης (Read) και εγγραφής (Write) των τιμών των μεταβλητών κατάστασης στις διαχειριζόμενες συσκευές, με ασφαλή τρόπο. Οι βασικότερες συμπληρωματικές λειτουργίες της διαχείρισης είναι η ανίχνευση των συσκευών (Discovery), η περιγραφή των δυνατοτήτων τους (Description) και η ενημέρωση για τα διάφορα συμβάντα (Eventing), που λαμβάνουν χώρα σε αυτές.



Σχήμα 4-4.Τα βασικά στοιχεία της διαχείρισης.

Όπως στο SNMP έτσι και στο CWMP οι διαχειριζόμενοι πόροι μοντελοποιούνται μέσω των «διαχειριζόμενων αντικειμένων» (managed objects). Κάθε αντικείμενο ενθυλακώνει τον

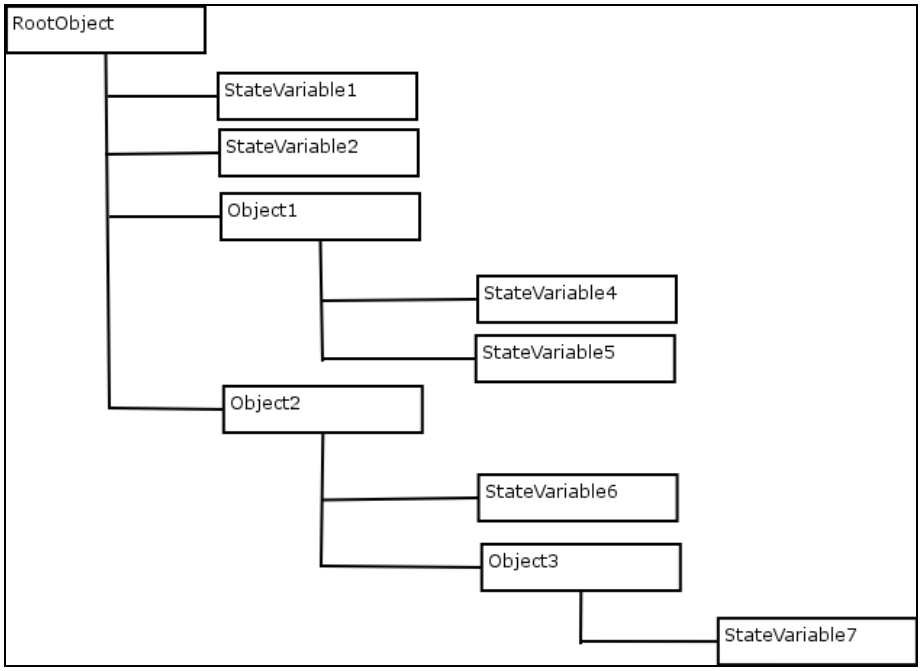
υποκείμενο πόρο και προσφέρει μια ιδεατή διεπαφή πρόσβασης. Όλα τα διαχειριζόμενα στοιχεία (συσκευές και εφαρμογές) εκθέτουν από μια συστάδα τέτοιων αντικειμένων (MIB – Management Information Base – Πληροφοριακή Βάση Διαχείρισης) μέσω μιας διεπαφής, που παρέχεται από τον πράκτορα. Η διεπαφή τυπικά ορίζεται από τις προδιαγραφές της φύσης των αντικειμένων (π.χ. τύπος, επιτρεπτές τιμές, μήκος μεταβλητών κατάστασης) αλλά και από το πρωτόκολλο διαχείρισης. Οι εφαρμογές διαχείρισης προσπελαύνουν τα διαχειριζόμενα αντικείμενα μέσω των διεπαφών αυτών προκειμένου να υλοποιήσουν τις πολιτικές διαχείρισης.

Στην περίπτωση του CWMP προτύπου χρησιμοποιούνται αντικείμενα (objects) και παράμετροι (parameters) για να μοντελοποιήσουν τη συνδρομητική συσκευή (CPE). Οι παράμετροι μεταφέρουν την πραγματική πληροφορία ρύθμισης και παρακολούθησης, ενώ τα αντικείμενα χρησιμοποιούνται για τη λογική ομαδοποίηση των παραμέτρων. Κάθε αντικείμενο αναπαριστά μια συγκεκριμένη περιοχή διαχείρισης εντός της συσκευής ή της υπηρεσίας και μπορεί να περιλαμβάνει παραμέτρους αλλά και άλλα υπό-αντικείμενα. Πέραν των ονομάτων των παραμέτρων και των αντικειμένων, το μοντέλο διαχείρισης περιλαμβάνει επίσης πληροφορία σχετική με τη φύση των παραμέτρων (τύπος, ανάγνωση / εγγραφή, περιγραφή, επιτρεπτές τιμές ή επιτρεπτό εύρος τιμών).

Το σχήμα 4-5 απεικονίζει την ιεραρχία των μοντέλων διαχείρισης. Μάλιστα η ιεραρχία αυτή εκφράζεται με τη σημειογραφία τελείας (dot notation) κάτι που φαίνεται και από τον πίνακα 4.1. Ο πίνακας αυτός προέρχεται από το μοντέλο της δικτυακής πύλης (IGD – Internet Gateway Device), που δίνεται στα κείμενα προτυποποίησης του DSL forum [1,2]. Στο τέλος του ονόματος κάθε αντικειμένου προστίθεται μια τελεία για να διαχωρίσει το αντικείμενο από τις παραμέτρους ή τα άλλα υπό-αντικείμενα που ακολουθούν.

Για παράδειγμα, στον πίνακα 4.1 η δικτυακή πύλη αναπαρίσταται με ένα αντίστοιχο αντικείμενο (*InternetGatewayDevice*.) το οποίο αποτελεί τη ρίζα του δέντρου της ιεραρχίας. Τα υπό-αντικείμενα (π.χ *DeviceInfo*) και οι παράμετροι (π.χ. *LANDeviceNumberOfEntries* και *WANDeviceNumberOfEntries, Manufacturer*), που βρίσκονται κάτω από αυτό σχηματίζουν τα κλαδιά και τα φύλλα του δέντρου αντίστοιχα.

Μάλιστα το CWMP πρότυπο περιλαμβάνει δύο τύπους αντικειμένων: τα αντικείμενα συσκευής (Device Objects) και τα αντικείμενα υπηρεσιών (Services Objects). Το αντικείμενο συσκευής περιλαμβάνει όλα τα υπό-αντικείμενα και τις παραμέτρους εκείνες που σχετίζονται με τη μοντελοποιημένη συσκευή. Το αντικείμενο υπηρεσιών περιλαμβάνει πληροφορία σχετική με τις μοντελοποιημένες υπηρεσίες που υποστηρίζονται από τη συσκευή. Για παράδειγμα, όλη η σχετική πληροφορία με μια δικτυακή πύλη (π.χ. ADSL παράμετροι, Ethernet και 802.11 παράμετροι, παράμετροι προώθησης των IP πακέτων) βρίσκεται κάτω από το αντικείμενο συσκευής. Ωστόσο μια πύλη, που ενσωματώνει και την υπηρεσία VoIP (προτυποποιημένη σε ξεχωριστό κείμενο [10]) θα εκθέτει επιπρόσθετα και τις VoIP παραμέτρους (π.χ. SIP/H.323, κωδικοποιητές φωνής) κάτω από το αντικείμενο των υπηρεσιών με τον τρόπο που απεικονίζει το σχήμα 4-6.



Σχήμα 4-5.Η ιεραρχική δομή της CWMP μοντελοποίησης.

Πίνακας 4.1. Ένα απόσπασμα από το CWMP μοντέλο της δικτυακής πύλης.

Name	Type	Write	Read	Description
Internet.GatewayDevice.	Object	-	R	The top-level object for an Internet Gateway Device.
LANDeviceNumberOfEntries	usignedInt	-	R	Number of Instances of LAN device
WANDeviceNumberOfEntries	usignedInt	-	R	Number of Instances of WAN device
Internet.GatewayDevice.DeviceInfo.	Object	-	R	This object contains general device information.
Manufacturer	String(64)	-	R	The manufacturer of the CPE (human readable string).

InternetGatewayDevice
- DeviceSummary
- **DeviceInfo**
- **ManagementServer**
- **Time**
- **UserInterface**
- **Layer3Forwarding**
- LANDeviceNumberOfEntries = 1
- **LANDevice.1**
- WANDeviceNumberOfEntries = 1
- **WANDevice.1**
- **……..**
- **Services**
- VoiceServiceNumberOfEntries = 1
- **VoiceService.{1}.**
  - **Capabilities.**
    - SignallingProtocols
    - MaxSessionCount
    - ToneGeneration
    - RTCP
    - ……...
    - **Codecs**
      - Codec
      - BitRate
      - PacketizationPeriod

Σχήμα 4-6.Παράδειγμα ιεραρχίας σε μια δικτυακή πύλη.

4.4. Αυτοματοποιώντας την ανάπτυξη και τη συντήρηση του λογισμικού των πρακτόρων διαχείρισης

Όπως έγινε φανερό από τα παραπάνω, τα δύο τελευταία χρόνια συντελείται μια διαρκής και έντονη προσπάθεια για τη μοντελοποίηση των συνδρομητικών συσκευών με απώτερο στόχο την απομακρυσμένη διαχείρισή τους από το νέο πρωτόκολλο CWMP. Μάλιστα για κάθε είδος συσκευής προτείνονται συνεχώς νέα βελτιωμένα μοντέλα (Device Versions), ενώ η κάθε συσκευή έχει τις δικές της δυνατότητες και ιδιαιτερότητες και επομένως υποστηρίζει ένα συγκεκριμένο υποσύνολο δυνατοτήτων (Device Profiles) από εκείνες που ορίζουν τα αντίστοιχα μοντέλα. Οι περισσότεροι κατασκευαστές διαθέτουν πολλούς τύπους συσκευών και για κάθε τύπο υπάρχουν συσκευές με διαφοροποιημένες δυνατότητες (διαφορετικά μοντέλα). Σε αυτές τις περιπτώσεις ο κατασκευαστής θα πρέπει για κάθε μοντέλο συσκευής να υλοποιεί και από ένα διαφορετικό πράκτορα. Με αυτό τον τρόπο η μεγάλης κλίμακας υλοποίηση και διαχείριση του λογισμικού τέτοιων πρακτόρων μπορεί να αποβεί δαπανηρή και χρειάζεται να δοθεί ιδιαίτερη προσοχή.

Αυτό που προτείνει η παρούσα ενότητα είναι η ιδέα της παραμετρικής υλοποίησης των CWMP πρακτόρων διαχείρισης. Κάτι τέτοιο θα επέτρεπε την αυτοματοποίηση της διαδικασίας ανάπτυξης λογισμικού των πρακτόρων με βάση το μοντέλο διαχείρισης, που εφαρμόζεται κάθε φορά (Device Version) και των πραγματικών δυνατοτήτων της κάθε συσκευής (Device Profile). Η αυτοματοποίηση αυτή θα μπορούσε με τη σειρά της να ελαχιστοποιήσει τόσο το κόστος υλοποίησης όσο και το κόστος συντήρησης του λογισμικού των πρακτόρων.

Πίνακας 4.2.Οι βασικές μέθοδοι του CWMP πρωτοκόλλου.

Λειτουργία	RPC Μέθοδος	Περιγραφή
Περιγραφή	GetRPCMethods	Καλείται από ένα CPE ή από τον ACS για να ζητήσει το σύνολο των υποστηριζόμενων μεθόδων από τον ACS ή το CPE αντίστοιχα..
	GetParameterNames	Καλείται από τον ACS για να ζητήσει τις διαχειρίσιμες παραμέτρους εντός του CPE.
Έλεγχος	SetParameterValues	Καλείται από τον ACS για να θέσει την τιμή μιας CPE παραμέτρου.
	GetParameterValues	Καλείται από τον ACS για να ζητήσει την τιμή μιας CPE παραμέτρου.
	Reboot	Με αυτή την εντολή ο ACS προστάζει το CPE να επανεκκινήσει.
	Download	Με αυτή την εντολή ο ACS προστάζει το CPE να μεταφορτώσει ένα συγκεκριμένο αρχείο από μια προκαθορισμένη τοποθεσία. Χρησιμοποιείται κυρίως για ανανέωση λογισμικού.
Ανίχνευση και Ενημέρωση Συμβάντος	Inform	Η μέθοδος αυτή είναι η βάση της ανίχνευσης και της ενημέρωσης συμβάντος. Στέλνεται κάθε φορά που το CPE εκκινεί μια συνεδρία με τον ACS και ενημερώνει για την τρέχουσα κατάσταση του CPE.
	SetParameterAttributes	Χρησιμοποιείται από τον ACS για να δηλώσει τις μεταβλητές κατάστασης του CPE για τις οποίες ο ACS ενδιαφέρεται να ενημερώνεται. Οι αναγγελίες του CPE προς τον ACS είναι υπό μορφή *Inform* μηνυμάτων τα οποία περιλαμβάνουν τις νέες τιμές των τροποποιημένων μεταβλητών.
	GetParameterAttributes	Με αυτή τη μέθοδο ο ACS ανακτά τα χαρακτηριστικά ενημέρωσης και πρόσβασης για κάθε μεταβλητή κατάστασης του CPE.

Όπως φαίνεται και από τη στοίβα πρωτοκόλλων του σχήματος 4-1, το CWMP αποτελείται από ένα σύνολο συγκεκριμένων μεθόδων (RPC methods), που χρησιμοποιούν σαν ορίσματα τις παραμέτρους της μοντελοποίησης (CPE Parameters) για να πραγματοποιήσουν τις διάφορες διαχειριστικές λειτουργίες. Οι βασικές αυτές μέθοδοι συγκεντρώνονται στον πίνακα 4.2. Οι μέθοδοι *SetParameterValues* και G*etParameterValues* συνιστούν τη ραχοκοκαλιά της

διαχείρισης, αφού μέσω αυτών ο ACS διαχειριστής ζητά να θέσει ή να διαβάσει τις διάφορες παραμέτρους. Επίσης οι μέθοδοι της ανίχνευσης συσκευών και ενημέρωσης συμβάντων χρησιμοποιούν σαν ορίσματα τις παραμέτρους της μοντελοποίησης προκειμένου να επιτρέπουν στον ACS την εγγραφή ενημέρωσης ανά μεταβλητή κατάστασης (*SetParameterAttributes/ GetParameterAttributes)* αλλά και την ίδια την ενημέρωση από τις συσκευές (*Inform*). Η *GetParameterNames* επιστρέφει όλες τις διαχειρίσιμες παραμέτρους της συσκευής, ενώ οι υπόλοιπες μέθοδοι (η μη σκιασμένες) είναι βοηθητικές και δεν έχουν να κάνουν με τις μοντελοποιημένες παραμέτρους. Συνεπώς στόχος της παρούσας εργασίας είναι η σταθερή υλοποίηση των μη σκιασμένων μεθόδων του πίνακα 4.2 και η παραμετρική υλοποίηση των σκιασμένων μεθόδων.

4.4.1. Εργαλείο περιγραφής συσκευών

Για να επιτευχθεί ο παραπάνω στόχος χρειάζεται καταρχάς να μετατραπεί σε ηλεκτρονική και επομένως επεξεργάσιμη μορφή, η πληροφορία των μοντέλων των συσκευών, που βρίσκεται στα σχετικά κείμενα προτυποποίησης. Όπως αναλύθηκε στο δεύτερο κεφάλαιο η XML γλώσσα είναι ιδανική για να περιγράψει και να μεταφέρει πολύπλοκες δομές δεδομένων. Για το λόγο αυτό προτείνεται η χρήση της XML για την περιγραφή των προτυποποιημένων μοντέλων των συσκευών. Έτσι λοιπόν τα αντικείμενα και οι παράμετροι κάθε συσκευής παρουσιάζονται σε ένα XML κείμενο (*CPE_DataModel.xml*) και παριστάνονται με μια δενδρική δομή. Το σχήμα 4-7 απεικονίζει το τμήμα του κείμενου, που αντιστοιχεί στο παράδειγμα του πίνακα 4.1.

```
            . . . . . . . . . .

  <object name="InternetGatewayDevice">
    <description>The top-level object for an
Internet Gateway Device</description>
    <stateVariable name="LANDeviceNumberOfEntries"
type="unsignedInt" writable="false">
      <description>Number of instances of LAN
Device</description>
    </stateVariable>
    <stateVariable name="WANDeviceNumberofEntries"
type="unsignedInt" writable="false">
      <description>Number of instances of WAN
Device</description>
    </stateVariable>
    <object name="DeviceInfo">
      <description>This object contains general
device information</description>
      <stateVariable name="Manufacturer"
type="string(64)" writable="false">
        <description>The manufacturer of the CPE
(human readable string)</description>
      </stateVariable>
    </object>
  </object>
            . . . . . . . . . .
```

Σχήμα 4-7.Το τμήμα της XML αναπαράστασης που αντιστοιχεί στις παραμέτρους της δικτυακής πύλης του πίνακα 4.1.

Για κάθε αντικείμενο δημιουργείται ένα νέο στοιχείο (*object*), το οποίο έχει σαν ιδιότητα το όνομα του αντικειμένου (π.χ. *object name= InternetGatewayDevice*). Κάθε τέτοιο στοιχείο

ενθυλακώνει μέσα του και άλλα στοιχεία όπως είναι εκείνο της περιγραφής του (*description*), των παραμέτρων (π.χ. *stateVariable name = LANDeviceNumberOfEntries*) αλλά και των υπό-αντικειμένων (π.χ. *object name =DeviceInfo*) που περιλαμβάνει. Στο σημείο αυτό πρέπει να τονιστεί ότι το εν λόγω XML κείμενο αποτελεί τη βάση πάνω στην οποία στηρίζονται όλες οι προτεινόμενες διαδικασίες αυτοματοποίησης της παρούσας εργασίας.

Για τη μεταφορά της πληροφορίας μοντελοποίησης από τα διάφορα κείμενα του DSL φόρουμ στο προαναφερθέν XML κείμενο, σχεδιάστηκε και υλοποιήθηκε το λογισμικό *Device Descriptor Tool* (Εργαλείο Περιγραφής Συσκευών). Ο *Device Descriptor* (σχήμα 4-8) είναι ένας αναγνώστης και συντάκτης ταυτόχρονα του αρχείου *CPE_DataModel.xml*. Προτρέπει το χρήστη να προσθέσει νέα αντικείμενα και παραμέτρους. Μάλιστα για κάθε αντικείμενο μπορεί να εισαχθεί πληροφορία σχετικά με την περιγραφή (Description) και την αναγκαιότητά του (Required). Στο πεδίο της περιγραφής ο χρήστης εισάγει το αντίστοιχο κείμενο, που βρίσκεται στο σχετικό πίνακα (τελευταία στήλη του πίνακα 4.1). Το πεδίο της αναγκαιότητας επιλέγεται εάν το αντικείμενο αυτό είναι απαραίτητο να υπάρχει στη συσκευή ώστε αυτή να συμμορφώνεται με το CWMP πρότυπο. Κάποια αντικείμενα είναι προαιρετικά και έτσι η συσκευή μπορεί να μην τα υποστηρίζει. Όσον αφορά στις παραμέτρους, αυτές έχουν περισσότερη πληροφορία, η οποία πέραν του ονόματος αποτελείται από τα εξής πεδία:

- Τύπος (type). Στο πεδίο αυτό εισάγεται ένας από τους επιτρεπτούς τύπους (integer, unsigned integer, string, boolean, dateTime, base64) που ορίζει το πρωτόκολλο CWMP. Ο τύπος αυτό δίνεται στη δεύτερη στήλη των πινάκων προτυποποίησης (βλέπε πίνακα 4.1).

- Μέγεθος (size). Στο πεδίο αυτό δίνεται το μέγιστο επιτρεπτό μέγεθος της κάθε μεταβλητής κατάστασης. Επίσης η πληροφορία αυτή δίνεται στη δεύτερη στήλη των πινάκων προτυποποίησης (βλέπε τελευταία σειρά του πίνακα 4.1).

- Αναγκαιότητα (Required). Μπορεί μια μεταβλητή κατάστασης να μην είναι αναγκαία για τη CWMP συμμόρφωση της συσκευής. Η αναγκαιότητα δίνεται στις στήλες τρία και τέσσερα των πινάκων και δηλώνεται με το χαρακτηριστικό γράμμα “R”. Εάν έστω σε μια από τις προαναφερθείσες στήλες υπάρχει το χαρακτηριστικό αυτό γράμμα τότε η μεταβλητή κατάστασης είναι αναγκαία για τη CWMP συμμόρφωση της συσκευής.

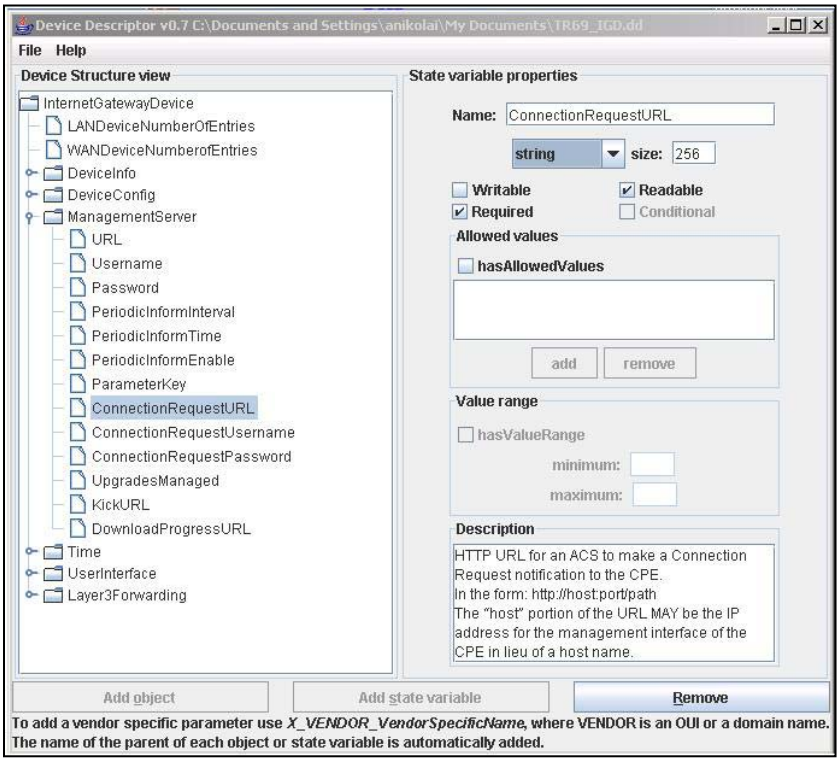
- Αναγνωσιμότητα (Read) / Εγγραψιμότητα (Write). Μια μεταβλητή κατάστασης μπορεί να είναι είτε αναγνώσιμη, είτε εγγράψιμη είτε και τα δύο μαζί. Η πληροφορία αυτή δίνεται στην τρίτη και τέταρτη στήλη των πινάκων προτυποποίησης.

- Επιτρεπτές Τιμές. Εάν μια μεταβλητή κατάστασης έχει διακριτές επιτρεπτές τιμές, αυτές θα δίνονται στη δεύτερη στήλη (μαζί με τον τύπο) των πινάκων προτυποποίησης.

- Επιτρεπτό εύρος τιμών. Ομοίως και το εύρος τιμών δίνεται στη δεύτερη στήλη των σχετικών πινάκων (π.χ. unsigned integer [1:4]).

- Περιγραφή. Ομοίως με την περίπτωση των αντικειμένων η περιγραφή των παραμέτρων δίνεται στην τελευταία στήλη (Description) των πινάκων προτυποποίησης.

Πέραν των παραπάνω ιδιοτήτων που προσδίδονται στα αντικείμενα και στις παραμέτρους υπάρχει μια ακόμα που συσχετίζει την αναγκαιότητα μιας παραμέτρου με την αναγκαιότητα ενός αντικειμένου. Η ιδιότητα αυτή ονομάζεται «Προϋπόθεση» (Conditional) και έχει να κάνει με την αναγκαιότητα μιας παραμέτρου, που ανήκει σε ένα προαιρετικό αντικείμενο. Όταν μία τέτοια παράμετρος έχει την ιδιότητα "C" (Conditional) τότε αυτή γίνεται αυτομάτως «Αναγκαία» (Required) εάν το αντικείμενο στο οποίο ανήκει επιλεγεί να υποστηρίζεται από τη συσκευή. Όπως φαίνεται και από το σχήμα 4-8, ο *Device Descriptor* υποστηρίζει με ένα φιλικό για το χρήστη τρόπο την εισαγωγή και ανάγνωση ολόκληρης της πληροφορίας, που δίνεται στα κείμενα προτυποποίησης.



Σχήμα 4-8. Το κύριο τμήμα του προτεινόμενου εργαλείου (Device Descriptor) εισαγωγής και ανάγνωσης των CWMP μοντέλων των συνδρομητικών συσκευών.

4.4.2. Εργαλείο δημιουργίας συσκευών

Επόμενος στόχος είναι η παραμετρική υλοποίηση των πρακτόρων διαχείρισης με βάση το προτυποποιημένο CWMP μοντέλο (*CPE_DataModel.xml*), του οποίου η XML αναπαράσταση κατασκευάστηκε προηγουμένως από το εργαλείο περιγραφής. Όπως προαναφέρθηκε η προτυποποίηση ορίζει ότι μια συσκευή θα πρέπει απαραιτήτως να υποστηρίζει ένα συγκεκριμένο σύνολο διαχειρίσιμων παραμέτρων, που ορίζονται από το αντίστοιχο μοντέλο. Ωστόσο κάθε μοντέλο ορίζει ένα πολύ μεγαλύτερο σύνολο παραμέτρων, οι οποίες έχουν να κάνουν με διάφορες άλλες λειτουργίες, πέραν των τυπικών, που μπορεί να

υποστηρίζει μια συσκευή. Για παράδειγμα για μια δικτυακή πύλη ορίζεται το βασικό προφίλ της (Baseline Profile), που περιλαμβάνει τις απαραίτητες παραμέτρους (π.χ. παράμετροι προώθησης πακέτων, PPP και DHCP παράμετροι,) για να θεωρείται η συσκευή αυτή μια CWMP πύλη. Ωστόσο ορίζονται ακόμη και ορισμένα άλλα σύνολα παραμέτρων (π.χ. ADSLWAN profile, WiFiLAN profile, QoS profile), τα οποία έχουν να κάνουν με προαιρετικά υποστηριζόμενες λειτουργίες, της κάθε συσκευής. Για παράδειγμα το WiFiLAN profile στο κείμενο προτυποποίησης είναι ένας πίνακας που περιλαμβάνει όλες εκείνες τις διαχειρίσιμες παραμέτρους μιας 802.11 συσκευής. Έτσι όταν η δικτυακή πύλη ενσωματώνει μια WiFi συσκευή οφείλει να υποστηρίζει όλες αυτές τις παραμέτρους. Για κάθε τύπο (π.χ. δικτυακή πύλη, συνδρομητικός αποκωδικοποιητής, εξυπηρετητής πολυμεσικού περιεχομένου) μπορούν να υπάρχουν συσκευές με προαιρετικές λειτουργίες που διαφοροποιούνται σημαντικά από κατασκευαστή σε κατασκευαστή αλλά και από μοντέλο σε μοντέλο.

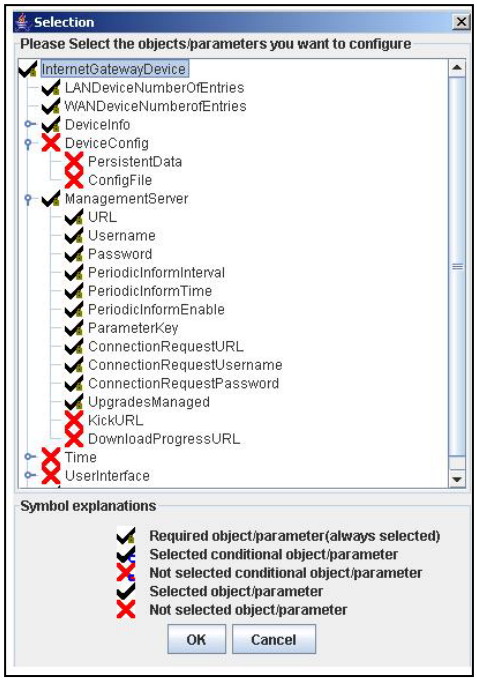
Συνεπώς η πληροφορία, που περιέχεται στο αρχείο εξόδου του Device Descriptor (*CPE_DataModel.xml*) πρέπει να δεχτεί μια περαιτέρω επεξεργασία σύμφωνα με τις δυνατότητες της κάθε συσκευής, έτσι ώστε στο τέλος να περιλαμβάνει μόνο εκείνες τις παραμέτρους, που υποστηρίζονται από τη συσκευή. Το νέο XML κείμενο που προκύπτει ονομάζεται *CPE_DeviceProfile.xml* ακριβώς για να τονίσει ότι περιέχει πληροφορία σχετική με τη συγκεκριμένη συσκευή για την οποία πρόκειται να κατασκευαστεί ο πράκτορας. Η πληροφορία αυτή δίνεται από το χρήστη – κατασκευαστή μέσω μιας νέας εφαρμογής, που σχεδιάστηκε και υλοποιήθηκε στα πλαίσια αυτής της εργασίας. Παρόλο που θα μπορούσε να είναι αυτόνομη ή να ενσωματώνεται στο εργαλείο *Device Descriptor*, ωστόσο στην παρούσα υλοποίηση η εφαρμογή *Selection* ενσωματώνεται στην εφαρμογή του εργαλείου δημιουργίας των συσκευών (*Device Creator Tool*), που αποτελεί τη βάση για την αυτόματη υλοποίηση της εφαρμογής του CWMP πράκτορα.

Το σχήμα 4-9 απεικονίζει το γραφικό περιβάλλον της εφαρμογής (*Selection*), η οποία δέχεται σαν είσοδο το αρχείο *CPE_DataModel.xml* και παρουσιάζει στο χρήστη όλες τις παραμέτρους, που ορίζει το μοντέλο. Όσον αφορά στις αναγκαίες παραμέτρους, αυτές επιλέγονται αυτόματα από την εφαρμογή, η οποία και δεν επιτρέπει στο χρήστη να τις αναιρέσει. Με αυτό τον τρόπο διασφαλίζεται η συμμόρφωση της συσκευής με το CWMP πρότυπο. Μάλιστα υποστηρίζεται και η αναγνώριση των "Conditional" παραμέτρων ώστε όταν ο χρήστης επιλέξει ένα αντικείμενο προαιρετικό, αυτόματα να επιλέγονται και οι "Conditional" παράμετροι προκειμένου να διασφαλιστεί και πάλι η συμμόρφωση της συσκευής με το πρότυπο. Όταν ολοκληρωθεί η διαδικασία της επιλογής, η εφαρμογή παράγει το αρχείο *CPE_DeviceProfile.xml,* το οποίο συνιστά την είσοδο για τον *Device Creator*.

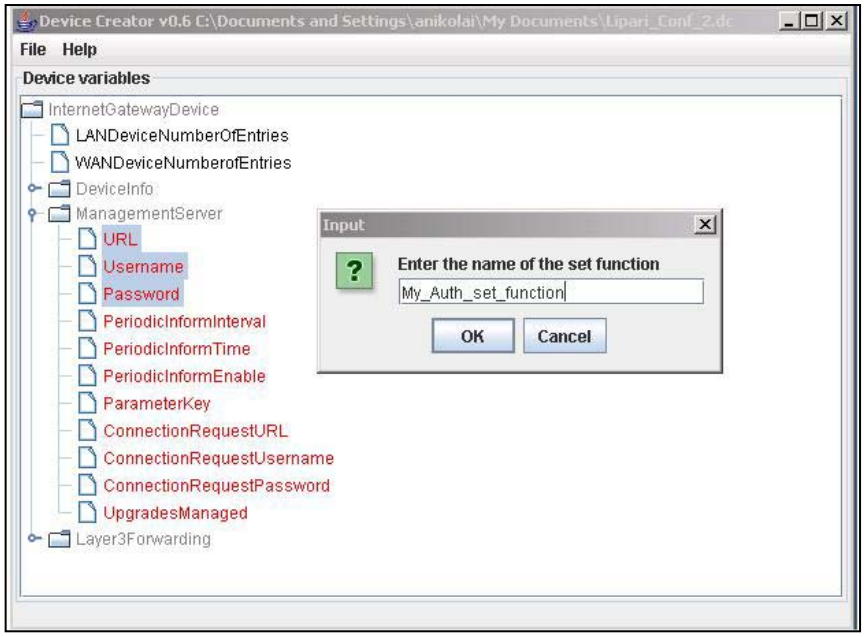
Ο *Device Creator* είναι η βασική εφαρμογή αφού παράγει αυτόματα τον πηγαίο κώδικα για τους CWMP πράκτορες. Ο κώδικας αυτός είναι σε γλώσσα C προκειμένου να είναι εύκολα μεταφέρσιμος στις διάφορες πλατφόρμες υλικού. Μάλιστα ο *Device Creator* παράγει ολοκληρωμένους πράκτορες για τα δύο πιο δημοφιλή λειτουργικά συστήματα, τα Windows και το Linux. Πριν όμως φτάσει το σημείο αυτό προηγείται μια πολύ σημαντική διαδικασία, κατά την οποία ο χρήστης ορίζει για κάθε παράμετρο ή σύνολο παραμέτρων τις συναρτήσεις

ενορχήστρωσης. Όπως ειπώθηκε και στο τρίτο κεφάλαιο, οι συναρτήσεις αυτές είναι στενά συνδεδεμένες με την πλατφόρμα του υλικού και υλοποιούν τις πραγματικές διαχειριστικές ενέργειες εντός της συσκευής. Ο *Device Creator* λοιπόν προτρέπει το χρήστη να ορίσει τα ονόματα των μεθόδων ενορχήστρωσης, που θα εκτελούνται όποτε ο ACS καλεί μια εκ των συναρτήσεων ανάθεσης (*SetParameterValues*) ή ανάγνωσης (*GetParameterValues*). Για παράδειγμα στο σχήμα 4-10, ο χρήστης ορίζει ότι η συνάρτηση *My_Auth_set_function* θα πρέπει να κληθεί από τον πράκτορα, εάν ο ACS καλέσει τη *SetParameterValues* μέθοδο με ορίσματα τουλάχιστον μια εκ των τριών παραμέτρων (URL, UserName και Password). Αυτό γίνεται διότι σε πολλές περιπτώσεις είναι δυνατό με μια μόνο κλήση συστήματος να αναγνωσθεί η να εγγραφεί η τιμή περισσοτέρων της μίας μεταβλητής κατάστασης. Προκειμένου λοιπόν να αποφευχθεί η πολλαπλή κλήση της ίδιας συνάρτησης συστήματος (μια φορά για κάθε παράμετρο) ο χρήστης μπορεί να ορίσει μια μέθοδο, η οποία θα καλεί μια και μόνο φορά τη συνάρτηση συστήματος και θα επιστρέφει ή θα θέτει την τιμή για περισσότερες παραμέτρους. Κάτι τέτοιο αποφεύγει σφάλματα, που συμβαίνουν ακόμη και στις σύγχρονες υλοποιήσεις πολλών SNMP πρακτόρων, όπως αποδείχτηκε από την εργασία της ενότητας 3.4.2.2.

Αφού ολοκληρωθεί η διαδικασία αντιστοίχισης όλων των παραμέτρων με τις συναρτήσεις υλοποίησης (Get και Set), η εφαρμογή δημιουργεί ένα νέο XML κείμενο (*CPE_DeviceImplemented.xml*), το οποίο περιλαμβάνει την πληροφορία του *CPE_DeviceProfile.xml* εμπλουτισμένη με τα ονόματα των μεθόδων ενορχήστρωσης. Η πληροφορία που βρίσκεται στο *CPE_DeviceImplemented.xml* είναι αρκετή προκειμένου η εφαρμογή του *Device Creator* να κατασκευάσει αυτόματα τον πηγαίο κώδικα του πράκτορα.



Σχήμα 4-9.Η εφαρμογή επιλογής (Selection) των διαχειρίσιμων παραμέτρων



Σχήμα 4-10.Το γραφικό περιβάλλον της προτεινόμενης εφαρμογής (Device Creator) για την αυτόματη παραγωγή πηγαίου κώδικα CWMP πρακτόρων.

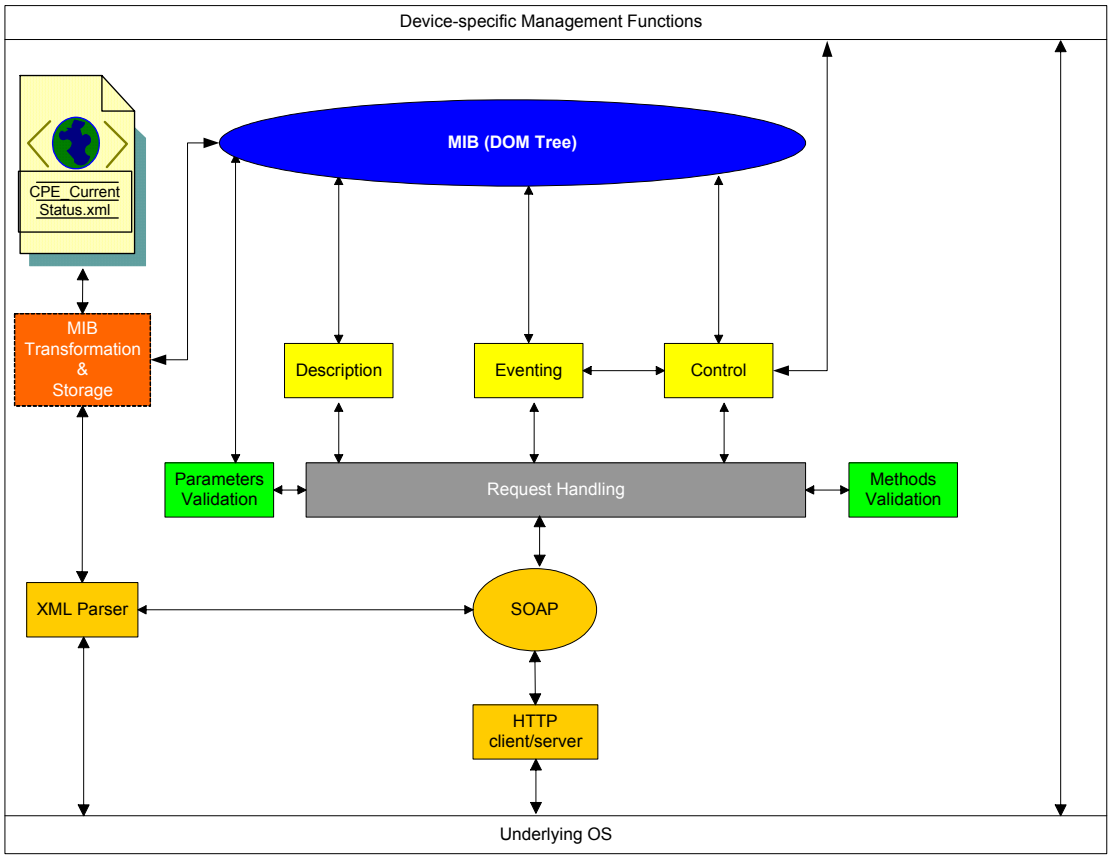
```
void My_Auth_set_function(ParameterValueList *params) {
      unsigned int i;
      //TODO: Add initialization code for this function
here...

      for(i=0;i<params->size;i++){
            if(strcmp(params->parameters[i].name,
"InternetGatewayDevice.ManagementServer.URL")==0) {
                  char *value = params->parameters[i].value;

            }

            if(strcmp(params->parameters[i].name,
"InternetGatewayDevice.ManagementServer.Username")==0) {
                  char *value = params->parameters[i].value;

            }

            if(strcmp(params->parameters[i].name,
"InternetGatewayDevice.ManagementServer.Password")==0) {
                  char *value = params->parameters[i].value;

            }

      }

}
```

Σχήμα 4-11. Ο παραμετρικός κώδικας του CWMP πράκτορα.

Στο σημείο αυτό πρέπει να τονιστεί ότι κώδικας που παράγεται από τον *Device Creator* δεν είναι ο τελικός κώδικας του πράκτορα. Στην πραγματικότητα είναι ένας σκελετός CWMP πράκτορα ο οποίος χρειάζεται τον κώδικα υλοποίησης για τις μεθόδους ενορχήστρωσης. Το σχήμα 4-11 απεικονίζει το σημείο του κώδικα στο οποίο ο προγραμματιστής προτρέπεται να εισάγει τον κώδικα υλοποίησης για τη Set συνάρτηση του σχήματος 4-10. Λεπτομέρειες για την αρχιτεκτονική του πράκτορα δίνονται στην επόμενη ενότητα.

4.5. Η αρχιτεκτονική του παραμετρικού πράκτορα

Το σχήμα 4-12 παρουσιάζει τη συνολική αρχιτεκτονική των CWMP πρακτόρων, που προκύπτουν από το εργαλείο *Device Creator*. Στην πραγματικότητα ένα μικρό μόνο σύνολο λειτουργιών (Μονάδα Μετασχηματισμού και Αποθήκευσης της βάσης διαχείρισης - MIB Transformation & Storage) χρειάζεται να παραχθεί δυναμικά, ενώ οι υπόλοιπες λειτουργίες είναι κοινές για όλους τους πράκτορες, που εξάγει το εργαλείο αυτοματοποίησης.



Σχήμα 4-12. Η αρχιτεκτονική του παραμετρικού CWMP πράκτορα.

4.5.1. Η βάση διαχείρισης εκφρασμένη σε XML

Ο πυρήνας ενός πράκτορα είναι η βάση της πληροφορίας διαχείρισης (MIB – Management Information Base). Τα βασικά στοιχεία της MIB είναι τα ονόματα των διαχειρίσιμων παραμέτρων και οι τιμές αυτών. Μάλιστα προκειμένου να υπάρχει συνέπεια μεταξύ των πραγματικών διαχειρίσιμων πόρων και της MIB, οι μέθοδοι ενορχήστρωσης κάθε φορά που θέτουν ή ανακτούν μια τιμή από το σύστημα, αναλαμβάνουν και την ανανέωση της βάσης. Ωστόσο η MIB οφείλει να διατηρεί και επιπρόσθετη πληροφορία, που αφορά κυρίως στην εγγραφή του διαχειριστή (ACS) για ενημέρωση συμβάντων. Όπως φαίνεται και στον πίνακα 4.2 ο ACS μπορεί (μέσω της μεθόδου *SetParameterAttributes)* να μαρκάρει κάθε μεταβλητή της MIB προκείμενου αυτή να παρακολουθείται από τον πράκτορα. Μόλις ο πράκτορας διαπιστώσει αλλαγή στην τιμή της ενημερώνει τον ACS. Επίσης ο ACS μπορεί για κάθε

μεταβλητή (μέσω της μεθόδου G*etParameterAttributes)* να ζητήσει την κατάσταση της εγγραφής του στην υπηρεσία ενημέρωσης του πράκτορα.

Μάλιστα η πληροφορία της MIB θα πρέπει αφενός να βρίσκεται στη μνήμη ώστε να είναι άμεσα προσβάσιμη στον πράκτορα και αφετέρου να αποθηκεύεται σε κάποιο αρχείο προκειμένου να κρατούνται οι τρέχουσες ρυθμίσεις του ACS. Κάτι τέτοιο δίνει τη δυνατότητα στη συσκευή να επαναφέρει άμεσα της προηγούμενες ρυθμίσεις, κατά την επανεκκίνησή της. Αυτό επιταχύνει τη ρύθμισή των συσκευών και περιορίζει παράλληλα το φόρτο επεξεργασίας και τη χρήση του εύρους ζώνης στον ACS διαχειριστή.

4.5.1.1. Το XML κείμενο τρέχουσας κατάστασης

Η συνολική πληροφορία (*CPE_DeviceImplemented.xml*), που έχει στη διάθεσή του ο *Device Creator* λίγο πριν δημιουργήσει τον πράκτορα, περιλαμβάνει το σύνολο των διαχειριζόμενων παραμέτρων για την κάθε υλοποίηση αλλά και την αντιστοιχία αυτών στις πραγματικές μεθόδους διαχείρισης του συστήματος. Στο σημείο αυτό προτείνεται ο εμπλουτισμός του *CPE_DeviceImplemented.xml* κειμένου με πληροφορία, που σχετίζεται με τις ανάγκες της βάσης διαχείρισης, έτσι όπως αυτές περιγράφηκαν παραπάνω. Συνολικά, για κάθε παράμετρο θα υπάρχει η ακόλουθη πληροφορία:

- Όνομα (*name*). Το όνομα της κάθε μοντελοποιημένης παραμέτρου. Η πληροφορία αυτή δίνεται για πρώτη φορά στο *CPE_Datamodel.xml* και διατηρείται μέχρι και στο *CPE_DeviceImplemented.xml* κείμενο.
- Τύπος και Μέγεθος (*type*). Ο τύπος και το μέγεθος της κάθε παραμέτρου. Η πληροφορία αυτή δίνεται για πρώτη φορά στο *CPE_Datamodel.xml* και διατηρείται μέχρι και στο *CPE_DeviceImplemented.xml* κείμενο.
- Αναγνωσιμότητα (*readable*) / Εγγραψιμότητα (*writable*). Η πληροφορία αυτή δίνεται για πρώτη φορά στο *CPE_Datamodel.xml* και διατηρείται μέχρι και στο *CPE_DeviceImplemented.xml* κείμενο.
- Επιτρεπτές Τιμές (*allowedValues*) ή Επιτρεπτό Εύρος Τιμών (*allowedValueRange*). Οι επιτρεπτές διακριτές τιμές που μπορεί να πάρει μια παράμετρος ή ένα συγκεκριμένο εύρος τιμών. Επίσης η πληροφορία αυτή δίνεται για πρώτη φορά στο *CPE_Datamodel.xml* και διατηρείται μέχρι και στο *CPE_DeviceImplemented.xml* κείμενο.
- Ονόματα Μεθόδων Ενορχήστρωσης (*getFunction - setFunction*). Η πληροφορία αυτή δίνεται από το χρήστη για πρώτη φορά στο *CPE_DeviceImplemented.xml* κείμενο και περιέχει για κάθε παράμετρο τις αντίστοιχες μεθόδους ενορχήστρωσης (Get/Set).
- Τρέχοντες δείκτες μεθόδων ενορχήστρωσης (getPointer - setPointer). Η πληροφορία αυτή σχετίζεται με τη θέση των μεθόδων ενορχήστρωσης στην προσωρινή μνήμη και δίνεται από τη μονάδα μετασχηματισμού και αποθήκευσης της MIB, κάθε φορά που εκκινεί η εφαρμογή του πράκτορα.

- Τρέχουσα Τιμή (*value*). Το πεδίο αυτό περιέχει τη βασική πληροφορία μιας MIB και φέρει την τελευταία τιμή που έχει επιστραφεί από μια μέθοδο ανάκτησης (Get) ή που έχει οριστεί από μια μέθοδο ρύθμισης (Set). Η πληροφορία αυτή ανανεώνεται συνεχώς κατά τη διάρκεια λειτουργίας της συσκευής.
- Ιδιότητα Ενημέρωσης (*Evented*). Η πληροφορία αυτή δίνεται από τον ACS στον πράκτορα κατά τη διάρκεια λειτουργίας της συσκευής. Όπως προαναφέρθηκε ο ACS μπορεί να εγγραφεί στην υπηρεσία ενημέρωσης του πράκτορα και να ενημερώνεται από αυτόν για τις αλλαγές των τιμών.

Έτσι λοιπόν προκύπτει ένα νέο XML κείμενο (*CPE_CurrentStatus.xml*) που περιέχει μικτή πληροφορία. Για τα τέσσερα πρώτα πεδία το *CPE_CurrentStatus.xml* περιλαμβάνει στατική πληροφορία που δίνεται από το προτυποποιημένο μοντέλο της συσκευής. Το πέμπτο πεδίο φέρει πάλι στατική πληροφορία την οποία αυτή τη φορά όμως δίνει ο ίδιος ο χρήστης – προγραμματιστής. Ωστόσο η πληροφορία των δύο τελευταίων πεδίων είναι δυναμική και δίνεται είτε από τον ACS (εντολές Set και εντολές εγγραφής για ενημέρωση) είτε από την ίδια τη συσκευή (εντολές Get). Στην παράγραφο 4.6.2 αναλύεται ο τρόπος με τον οποίο ο πράκτορας χρησιμοποιεί την παραπάνω πληροφορία για κάθε λειτουργία διαχείρισης.

```
            ..........

  <object name="InternetGatewayDevice">
    <stateVariable name="LANDeviceNumberOfEntries" type="unsignedInt"
writable="false" readable="true" allowedValues="" allowedValueRange=""
getFunction="My_RGW_Get" setFunction="" value="1" Evented="1">
    </stateVariable>
    <stateVariable name="WANDeviceNumberOfEntries" type="unsignedInt"
writable="false" readable="true" allowedValues="" allowedValueRange=""
getFunction="My_RGW_Get" setFunction="" value="1" Evented="0">
    </stateVariable>
    <object name="DeviceInfo">
    <stateVariable name="Manufacturer" type="string(64)" writable="false"
readable="true" allowedValues="" allowedValueRange="" getFunction="My_RGW_Get"
setFunction="My_RGW_Set" value="1" Evented="0">
    </object>
            .........
            .........
  </object>

            .........
```

Σχήμα 4-13.Το απόσπασμα του κειμένου τρέχουσας πληροφορίας που αντιστοιχεί στον πίνακα 4.1.

4.5.1.2. Η μονάδα μετασχηματισμού και αποθήκευσης της MIB

Για λόγους ταχύτητας πρόσβασης στη MIB, οι διάφορες μονάδες (Description, Control, Eventing, Parameters/Methods Validation) του πράκτορα δεν χρησιμοποιούν άμεσα το XML κείμενο. Κάθε φορά που εκκινεί, ο πράκτορας διαβάζει, με τη βοήθεια του γραμματικού αναλυτή (XML parser), το *CPE_CurrentStatus.xml* αρχείο που βρίσκεται αποθηκευμένο στο δίσκο ή τη Flash μνήμη της συσκευής. Στη συνέχεια μεταφέρει όλη την πληροφορία του αρχείου σε μια C δομή, που δημιουργεί στη μνήμη. Με αυτό τον τρόπο η MIB μεταφέρεται στη μνήμη (DOM Tree), η οποία είναι άμεσα προσπελάσιμη από τις διάφορες μονάδες του πράκτορα. Κάθε φορά που εκτελείται μια πράξη εγγραφής στη DOM δομή ο πράκτορας

μεταφέρει τα νέα δεδομένα στο *CPE_CurrentStatus.xml* προκειμένου να το κρατά συνεχώς ενήμερο. Όλες αυτές οι λειτουργίες του πράκτορα επιτελούνται από τη μονάδα μετασχηματισμού και αποθήκευσης της MIB (MIB Transformation & Storage).

Η μονάδα αυτή είναι το μοναδικό παραμετρικό τμήμα του πράκτορα. Η παραμετροποίηση έγκειται στο γεγονός ότι ο *Device Creator* κατασκευάζει κώδικα με την εξής λειτουργία: για κάθε μέθοδο ενορχήστρωσης, που ορίζεται από τον χρήστη (στο *CPE_DeviceImplemented.xml*) δημιουργείται μια C δομή, η οποία περιέχει το όνομα και το δείκτη της μεθόδου που ορίστηκε. Έτσι όταν εκκινεί ο πράκτορας δημιουργεί ένα πίνακα με τέτοιες δομές ώστε να γνωρίζει ανά πάσα στιγμή τη θέση στη μνήμη της κάθε μέθοδου ενορχήστρωσης. Κατά διάρκεια της δημιουργίας του DOM δέντρου στη μνήμη, η μονάδα μετασχηματισμού τοποθετεί την πληροφορία των δεικτών των μεθόδων στα αντίστοιχα πεδία (*setPointer* και *getPointer*). Αυτό βοηθά καθοριστικά τη μονάδα ελέγχου (Control) να αποφασίζει τις μεθόδους ενορχήστρωσης, που θα καλεί με βάση τις παραμέτρους της *Set/GetParameterValues* μεθόδου.

4.5.2. Οι υπόλοιπες λειτουργίες

Ο πράκτορας υποστηρίζει ένα HTTP πελάτη και ένα HTTP εξυπηρετητή. Η μεταφορά της πληροφορίας διαχείρισης γίνεται αποκλειστικά μέσω του HTTP πελάτη, ενώ ο HTTP εξυπηρετητής χρησιμοποιείται για την αμφίδρομη εκκίνηση των συνεδριών διαχείρισης. Σύμφωνα με το CWMP πρωτόκολλο, όταν ο ACS επιθυμεί να ξεκινήσει μια συνεδρία, ο HTTP πελάτης του συνδέεται με τον HTTP εξυπηρετητή του πράκτορα και στέλνει ένα κενό HTTP GET μήνυμα. Στο μήνυμα αυτό ο πράκτορας δεν αποκρίνεται άμεσα, αλλά κλείνει τη σύνδεση και ξεκινά μέσω του HTTP πελάτη του μια νέα συνεδρία με τον ACS. Με αυτό τον τρόπο οι διαχειριστικές συνεδρίες μπορούν να ξεκινήσουν τόσο από τον πράκτορα όσο και από τον ACS.

Ο HTTP πελάτης του πράκτορα δέχεται τις διαχειριστικές εντολές από τον ACS με τη μορφή SOAP φακέλων, οι οποίοι ενθυλακώνονται στα HTTP Response μηνύματα του τελευταίου. Στη συνέχεια ο πελάτης μεταφέρει τους φακέλους στη μονάδα SOAP για επεξεργασία. Η SOAP μονάδα χρησιμοποιεί μεθόδους του XML γραμματικού αναλυτή (parser) για να ελέγξει τη συμμόρφωση του φακέλου με το SOAP πρωτόκολλο και να αποκωδικοποιήσει την RPC μέθόδο που περιέχεται. Στη συνέχεια στέλνει τα ονόματα των μεθόδων και τα ορίσματά τους (όνομα και τιμή) στη μονάδα χειρισμού των αιτήσεων (Request Handling).

4.5.2.1. Η μονάδα χειρισμού των αιτήσεων (Request Handling)

Οι βασικές λειτουργίες της μονάδας είναι ο έλεγχος της ληφθείσας μεθόδου όσον αφορά στη συμμόρφωσή της με το CWMP πρωτόκολλο αλλά και η κλήση της αντίστοιχης μονάδας, που υλοποιεί μια από τις CWMP λειτουργίες (Description, Control, Eventing). Πιο συγκεκριμένα, η μονάδα χειρισμού συνεργάζεται με δύο επιμέρους μονάδες: τη μονάδα επικύρωσης μεθόδων (Methods Validation) και τη μονάδα επικύρωσης παραμέτρων (Parameters Validation) . Η πρώτη έχει αποθηκευμένες όλες τις υποστηριζόμενες μεθόδους του CWMP πράκτορα (αυτές του πίνακα 4.2 και ενδεχομένως και ορισμένες προαιρετικές που ορίζει το

CWMP) και ελέγχει εάν το όνομα της μεθόδου, που κλήθηκε από τον ACS, ανήκει σε μια από αυτές. Στην περίπτωση που η καλούμενη μέθοδος δεν υποστηρίζεται, η μονάδα χειρισμού της αίτησης απαντά αρνητικά στον ACS, σύμφωνα με το πρωτόκολλο. Στην αντίθετη περίπτωση η μονάδα χειρισμού χρησιμοποιεί τη μονάδα επικύρωσης παραμέτρων για να ελέγξει εάν όλοι οι παράμετροι της κληθείσας μεθόδου υποστηρίζονται από τη συσκευή. Η μονάδα επικύρωσης ερευνά την ύπαρξη των παραμέτρων αυτών στο DOM δέντρο της MIB. Στην περίπτωση που κάποια από τις παραμέτρους δεν υποστηρίζεται, η μονάδα χειρισμού στέλνει ένα μήνυμα σφάλματος ενημερώνοντας τον ACS για τις άγνωστες παραμέτρους (*Invalid Arguments*). Εάν όλες οι παράμετροι είναι γνωστές, η μονάδα χειρισμού (έχοντας την πληροφορία του πίνακα 4.2) περνάει τον έλεγχο στην αντίστοιχη CWMP μονάδα.

4.5.2.2. Η μονάδα περιγραφής (Description)

Η μονάδα αυτή εξυπηρετεί δύο CWMP μεθόδους: την *GetRPCMethods* και την *GetParameterNames.* Για τη πρώτη μέθοδο η μονάδα έχει αποθηκευμένες όλες τις υποστηριζόμενες μεθόδους του CWMP πράκτορα (αυτές του πίνακα 4.2 και ενδεχομένως και ορισμένες προαιρετικές που ορίζει το CWMP) και απλά τις επιστρέφει. Όσον αφορά στη δεύτερη μέθοδο η μονάδα λαμβάνει από το DOM δέντρο της MIB όλες τις υποστηριζόμενες παραμέτρους τις οποίες και επιστρέφει.

4.5.2.3. Η μονάδα ελέγχου (Control)

Η μονάδα ελέγχου εξυπηρετεί τέσσερις CWMP μεθόδους: τη *Reboot*, τη *Download*, τη *SetParameterValues* και τη *GetParameterValues*. Οι δύο πρώτες δεν χρησιμοποιούν σαν ορίσματα κάποια παράμετρο της MIB οπότε η μονάδα ελέγχου γνωρίζει εκ των προτέρων και καλεί τις μεθόδους του συστήματος που υλοποιούν την επανεκκίνηση και τη μεταφόρτωση αντίστοιχα. Αντίθετα οι δύο τελευταίες παίρνουν πάντα σαν όρισμα ένα αυθαίρετο σύνολο διαχειρίσιμων παραμέτρων και έτσι η μονάδα ελέγχου συμβουλεύεται το DOM δέντρο της MIB προκειμένου να πληροφορηθεί τις μεθόδους ενορχήστρωσης, που αντιστοιχούν σε κάθε παράμετρο. Μάλιστα έχει δοθεί ιδιαίτερη προσοχή ώστε, στην περίπτωση που η *Set/GetParameterValues* περιέχει μια ή περισσότερες παραμέτρους, που αντιστοιχούν στην ίδια μέθοδο ενορχήστρωσης τότε η μέθοδος αυτή καλείται μια και μόνο φορά από τη μονάδα ελέγχου. Ο λόγος που γίνεται αυτός είναι διότι σε πολλές περιπτώσεις είναι δυνατό με μια μόνο κλήση συστήματος να αναγνωσθεί η να εγγραφεί η τιμή περισσοτέρων της μίας μεταβλητής κατάστασης. Προκειμένου λοιπόν να αποφευχθεί η πολλαπλή κλήση της ίδιας συνάρτησης συστήματος ο χρήστης του *Device Creator* έχει προκαθορίσει μια μέθοδο, η οποία θα καλεί μια και μόνο φορά τη συνάρτηση συστήματος και θα επιστρέφει ή θα θέτει την τιμή για περισσότερες παραμέτρους.

4.5.2.4. Η μονάδα ενημέρωσης συμβάντων (Eventing)

Η μονάδα ενημέρωσης υλοποιεί την υπηρεσία ενημέρωσης συμβάντων του πράκτορα. Καταρχάς εξυπηρετεί την εγγραφή ενημέρωσης του ACS ανά μεταβλητή κατάστασης και εν

συνεχεία αναλαμβάνει τη διαρκή παρακολούθηση των σχετικών παραμέτρων του συστήματος. Στην περίπτωση που κάποια παράμετρος έχει μεταβληθεί η μονάδα ξεκινά μια νέα συνεδρία με τον ACS προκειμένου να τον ενημερώσει για τη μεταβολή. Μάλιστα η μονάδα υποστηρίζει και τα δύο είδη ενημέρωσης (Παθητική και Ενεργητική – Passive & Active Notification), που προβλέπονται από το CWMP πρωτόκολλο. Στην ενεργητική ενημέρωση όταν παρατηρείται αλλαγή στην τιμή της παραμέτρου ο πράκτορας αμέσως ξεκινά τη συνεδρία ενημέρωσης (*Inform* μέθοδος). Αντίθετα κατά την παθητική ενημέρωση ο πράκτορας δεν ενημερώνει τον ACS κάθε φορά που αλλάζει η παράμετρος αλλά την επόμενη φορά που θα ξεκινήσει (π.χ. εξαιτίας του ACS) μια συνεδρία μεταξύ του ACS και του πράκτορα. Το πρώτο είδος έχει το πλεονέκτημα της ενημέρωσης σε σχεδόν πραγματικό χρόνο αλλά καταναλώνει τόσο δικτυακούς όσο και επεξεργαστικούς πόρους του ACS. Για το λόγο αυτό είναι σκόπιμο να εφαρμόζεται μόνο σε πραγματικά κρίσιμες μεταβλητές και να αποφεύγεται σε μεταβλητές, που η τιμή τους μεταβάλλεται με μεγάλη συχνότητα . Αντίθετα το δεύτερο είδος ενημέρωσης δεν δημιουργεί υπερκεφαλισμούς αλλά ταυτόχρονα στερείται του πλεονεκτήματος της έγκαιρης ενημέρωσης.

Η μονάδα ενημέρωσης υλοποιεί τις μεθόδους *SetParameterAttributes* και *GetParameterAttributes*. Όπως προαναφέρθηκε, με τη *SetParameterAttributes* ο ACS μπορεί να εγγράφεται στη υπηρεσία ενημέρωσης, ενώ με τη *GetParameterAttributes* μπορεί να πληροφορείται για τις ιδιότητες της ενημέρωσης (παθητική, ενεργητική ή απενεργοποίηση ενημέρωσης) ανά μεταβλητή. Κατά την εγγραφή του ACS η μονάδα ενημέρωσης επικοινωνεί πρώτα με τη μονάδα ελέγχου και ζητά από αυτή να διαβάσει την τιμή ή τις τιμές των μεταβλητών για τις οποίες επιθυμεί να ενημερώνεται ο ACS. Έτσι η μονάδα ελέγχου λειτουργεί ακριβώς όπως θα λειτουργούσε εάν δέχονταν μια G*etParameterValues* μέθοδο με ορίσματα τα ορίσματα της *SetParameterAttributes*. Με αυτό τον τρόπο η μονάδα αφενός αρχικοποιείται (λαμβάνει τις αρχικές τιμές των παραμέτρων) και αφετέρου διασφαλίζει ότι δύναται να διαβάσει και επομένως να παρακολουθήσει τις τιμές των ζητούμενων παραμέτρων του συστήματος. Ανάλογα με το αποτέλεσμα αυτής της διαδικασίας ο πράκτορας πληροφορεί τον ACS (με τη *SetParameterAttributes* response) για την επιτυχία ή την αποτυχία της εγγραφής του ανά μεταβλητή. Για τις παραμέτρους εκείνες που η παραπάνω διαδικασία ήταν επιτυχής, η μονάδα ενημέρωσης επικοινωνεί με τη MIB (DOM Tree) και θέτει τις ιδιότητες ενημέρωσης που όρισε ο ACS κατά την εγγραφή. Όταν ο ACS ζητά τις ιδιότητες ενημέρωσης η μονάδα επικοινωνεί πάλι με τη MIB, διαβάζει την ιδιότητα *Evented* (σχήμα 4-13) για τις παραμέτρους που περιέχει η *GetParameterAttributes* και επιστρέφει στον ACS διαχειριστή τις τιμές αυτής της ιδιότητας.

Αφού περατωθεί η διαδικασία της εγγραφής η μονάδα ενημέρωσης αρχίζει την παρακολούθηση των σχετικών παραμέτρων. Η διαδικασία αυτή επιτυγχάνεται πάλι μέσω της μονάδας ελέγχου με την οποία η μονάδα ενημέρωσης επικοινωνεί ανά τακτά χρονικά διαστήματα για να ζητήσει τις τιμές των υπό επιτήρηση παραμέτρων. Μάλιστα οι παράμετροι που είναι ικανές να παράγουν ενημέρωση επιτηρούνται με συχνότητα ανάλογη της ιδιότητάς τους (παθητικές ή ενεργητικές). Έτσι οι τιμές των ενεργητικών παραμέτρων αναμένεται να ανακτώνται με συχνότητα μιας ή και περισσοτέρων τάξεων μεγέθους μεγαλύτερη από εκείνη

των παθητικών. Οι τιμές αυτές συγκρίνονται κάθε φορά με τις τελευταίες γνωστές τιμές που είναι αποθηκευμένες στη MIB (DOM Tree). Στη συνέχεια η μονάδα ενημέρωσης ενημερώνει τη MIB με τις νέες τιμές, συλλέγει τις αλλαγμένες μεταβλητές και ανάλογα με την ιδιότητά τους τις ομαδοποιεί πάλι σε δύο κατηγορίες: στις ενεργητικές και στις. παθητικές Οι πρώτες χρήζουν άμεσης εξυπηρέτησης και για αυτό ο πράκτορας εκκινεί μια νέα συνεδρία κατά την οποία δίνει όλες τις νέες τιμές μέσω της *Inform* μεθόδου. Αντίθετα η ομάδα των παθητικών παραμέτρων αποθηκεύεται σε μια εσωτερική δομή της μονάδας ενημέρωσης και αναφέρεται στον ACS διαχειριστή στην αμέσως επόμενη συνεδρία.

4.6. Αυτοματοποιώντας παράπλευρες διαδικασίες

Όμως η χρησιμότητα της XML τεχνολογίας δεν σταματά εδώ. Διαδικασίες όπως η δοκιμή και η συντήρηση του λογισμικού των πρακτόρων αλλά και η ίδια η ρύθμιση των συσκευών από το διαχειριστή αποκτούν σημαντικά πλεονεκτήματα όταν επιτελούνται με τη βοήθεια της XML και σχετικών τεχνολογιών. Παρακάτω προτείνεται μια σειρά μεθόδων, που έχουν σαν στόχο την απλοποίηση και επιτάχυνση των παραπάνω διαδικασιών με τη βοήθεια διαφόρων XML κειμένων, τα οποία πηγάζουν και αυτά από το προτυποποιημένο μοντέλο της συσκευής.

4.6.1. Η αυτοματοποίηση των δοκιμών στον πράκτορα

Το αμέσως επόμενο βήμα μετά την υλοποίηση ενός πράκτορα είναι η δοκιμή του. Η διαδικασία αυτή μπορεί να γίνει αρκετά σύνθετη και χρονοβόρα, ιδιαίτερα όταν πρόκειται για πρωτόκολλα διαχείρισης προσανατολισμένα στο μοντέλο της συσκευής. Μάλιστα η κατάσταση δυσχεραίνει ακόμα περισσότερο όταν πρόκειται για πολύπλοκες συσκευές με μεγάλο αριθμό μεταβλητών κατάστασης και για πρωτόκολλα διαχείρισης με ποικίλες λειτουργίες. Μια τέτοια κατάσταση συναντάται στη μοντελοποίηση του DSL forum [1-5, 10] για διαχείριση μέσω του πρωτοκόλλου CWMP. Για να δοθεί μια εικόνα του προβλήματος αρκεί να αναφερθεί ότι το προτυποποιημένο μοντέλο της δικτυακής πύλης [4] ορίζει πάνω από 120 αναγκαίες μεταβλητές κατάστασης και 400 προαιρετικές, εκ των οποίων τουλάχιστον οι μισές χρησιμοποιούνται στις περισσότερες εμπορικές υλοποιήσεις (π.χ. επιπλέον 70 διαχειρίσιμες παράμετροι χρειάζονται στην περίπτωση που η δικτυακή πύλη υποστηρίζει ένα ADSL διαμορφωτή / αποδιαμορφωτή). Για όλες αυτές τις παραμέτρους ο πράκτορας οφείλει να υποστηρίζει τουλάχιστον τις μεθόδους του πίνακα 4.2.

Παρακάτω δίνονται σε διαδοχικά στάδια οι κύριες δοκιμές, που θα πρέπει να επιτελούνται και προτείνεται μια σειρά από διαδικασίες αυτοματοποίησης. Οι τελευταίες θα μπορούν να ενσωματωθούν σε μια κοινή εφαρμογή (*Device Tester*) προκειμένου να επιτευχθεί η ολοκλήρωση των αυτοματοποιημένων δοκιμών. Η εφαρμογή του *Device Tester* προσομοιώνει έναν ACS διαχειριστή, ο οποίος καλεί τις διάφορες CWMP μεθόδους και ελέγχει τις αποκρίσεις του πράκτορα.

4.6.1.1. Στάδιο 1: δοκιμή συμμόρφωσης με το πρωτόκολλο διαχείρισης

Καταρχάς ο πράκτορας θα πρέπει να μπορεί να εκκινεί τη συνεδρία διαχείρισης αλλά και να δέχεται τις αιτήσεις σύνδεσης (Connection Request) του ACS ώστε να υποστηρίζεται η

αμφίδρομη σύνδεση. Επίσης θα πρέπει να παράγει τους SOAP φακέλους έτσι όπως αυτοί προβλέπονται από το CWMP πρωτόκολλο [1]. Μάλιστα ο πράκτορας θα πρέπει να αποκρίνεται θετικά μόνο στις RPC μεθόδους του πίνακα 4.2 και μόνο στην περίπτωση που αυτές οι μέθοδοι περιέχουν σωστά ορίσματα. Αυτό σημαίνει ότι θα πρέπει να περιέχονται μόνο παράμετροι που υποστηρίζει η συσκευή και μάλιστα με τιμές σύμφωνες με τους τύπους που ορίζει το μοντέλο. Η πληροφορία αυτή βρίσκεται στο *CPE_DeviceProfile.xml* και στο *CPE_DeviceImplemented.xml* που χρησιμοποιήθηκαν παραπάνω. Ωστόσο χρησιμοποιείται το δεύτερο κείμενο διότι αυτό είναι απαραίτητο για τα επόμενα στάδια. Επίσης θα πρέπει η εφαρμογή του *Device Tester* να παρέχει στο χρήστη ένα περιβάλλον, στο οποίο εκείνος θα μπορεί σχεδιάζει και να αποθηκεύει σε ένα νέο XML κείμενο (*TestProtocol.xml*) τα σενάρια των δοκιμών πρωτοκόλλου.

4.6.1.2. Στάδιο 2: δοκιμή περιγραφικής λειτουργίας

Στο στάδιο αυτό ερωτάται ο πράκτορας για τις μεθόδους που υποστηρίζει (*GetRPCmethods*) και στη συνέχεια διασταυρώνεται η απόκρισή του με τη λίστα των μεθόδων, που θα έπρεπε να υποστηρίζει η κάθε υλοποίηση (πίνακας 4.2). Επίσης ο πράκτορας θα πρέπει να ελέγχεται εάν μπορεί να πληροφορεί σωστά το διαχειριστή για όλες τις παραμέτρους (*GetParameterNames*), που υποστηρίζονται από τη δεδομένη υλοποίηση. Για να γίνει κάτι τέτοιο θα πρέπει ο *Device Tester* να διαθέτει την πληροφορία του κειμένου *CPE_DeviceImplemented.xml* από το οποίο και δημιουργήθηκε αυτόματα ο πράκτορας. Η ίδια πληροφορία περιέχεται επίσης και στο *CPE_DeviceProfile.xml.* Ωστόσο αυτό το κείμενο δεν χρησιμοποιείται διότι δεν περιέχει την πληροφορία των μεθόδων ενορχήστρωσης, που είναι χρήσιμη για το επόμενο στάδιο. Μέχρι εδώ λοιπόν η εφαρμογή των δοκιμών χρειάζεται σαν είσοδο μόνο ένα XML κείμενο, το οποίο είναι ήδη διαθέσιμο από το στάδιο της υλοποίησης.

4.6.1.3. Στάδιο 3: δοκιμή λειτουργίας ελέγχου

Πρόκειται για το κύριο στάδιο των δοκιμών αφού εδώ ελέγχεται η δυνατότητα του πράκτορα να ρυθμίσει και να παρακολουθήσει τη συσκευή. Μάλιστα αυτό θα πρέπει γίνεται ατομικά για όλες τις παραμέτρους. Πιο συγκεκριμένα, για κάθε παράμετρο ο *Device Tester* θέτει την τιμή της (*SetParameterValues*) και στη συνέχεια ελέγχει (G*etParameterValues)* εάν ο πράκτορας έχει όντως πραγματοποιήσει τη συγκεκριμένη ανάθεση. Η διαδικασία είναι σειριακή και για κάθε αποτυχία καταγράφεται και παρουσιάζεται στο χρήστη-προγραμματιστή η μέθοδος ενορχήστρωσης, οι αντίστοιχες παράμετροι, που παρουσίασαν το πρόβλημα καθώς και η πιθανή αιτία αυτού (μέσω των CWMP κωδικών σφάλματος που επιστρέφει ο πράκτορας). Σε αυτή τη διαδικασία η μόνη πληροφορία που χρειάζεται και πάλι ο *Device Tester* είναι το κείμενο *CPE_DeviceImplemented.xml*, που χρησιμοποιήθηκε και στο προηγούμενο στάδιο.

4.6.1.4. Στάδιο 4: ολοκληρωμένη δοκιμή διαχείρισης υπηρεσίας

Πέραν της δυνατότητας του πράκτορα να θέτει και να λαμβάνει τις τιμές των παραμέτρων ατομικά, θα πρέπει να ελέγχεται και κατά πόσο η συσκευή αποκρίνεται στις συνολικές ρυθμίσεις που απαιτούν οι διάφορες υπηρεσίες. Έτσι λοιπόν ο *Device Tester* προτείνεται να παρέχει ένα περιβάλλον στο οποίο ο χρήστης θα κατασκευάζει τις ρυθμίσεις που χρειάζονται ανά υπηρεσία. Η πληροφορία αυτή θα εκφράζεται με ένα ακόμα XML κείμενο (*TestScenario.xml*) το οποίο θα περιέχει την αλληλουχία των μεθόδων διαχείρισης που θα καλούνται καθώς και τις αντίστοιχες τιμές των ορισμάτων τους. Με αυτό τον τρόπο ελέγχεται η διαχειρισιμότητα της συσκευής για ένα σύνολο βασικών υπηρεσιών, που πρόκειται να υποστηριχτούν από εκείνη. Μάλιστα αυτού του είδους οι δοκιμές απαιτούν επιπρόσθετα ένα περιβάλλον, το οποίο θα προσομοιώνει τις πραγματικές συνθήκες παροχής των υπηρεσιών. Για παράδειγμα, σε μία συσκευή που υποστηρίζει την υπηρεσία φωνής (VoIP) θα πρέπει (κατόπιν της εφαρμογής των σχετικών ρυθμίσεων από τον *Device Tester*) να ελέγχεται εάν αυτή μπορεί να επικοινωνήσει κανονικά με το υπόλοιπο δίκτυο (π.χ. σωστή ρύθμιση δικτυακών παραμέτρων και σηματοδοσίας) και μάλιστα αν παρέχει το προσδοκώμενο επίπεδο υπηρεσίας (π.χ. χρήση προβλεπόμενου κωδικοποιητή φωνής και σωστό μαρκάρισμα των παραγόμενων IP πακέτων για υποστήριξη QoS).

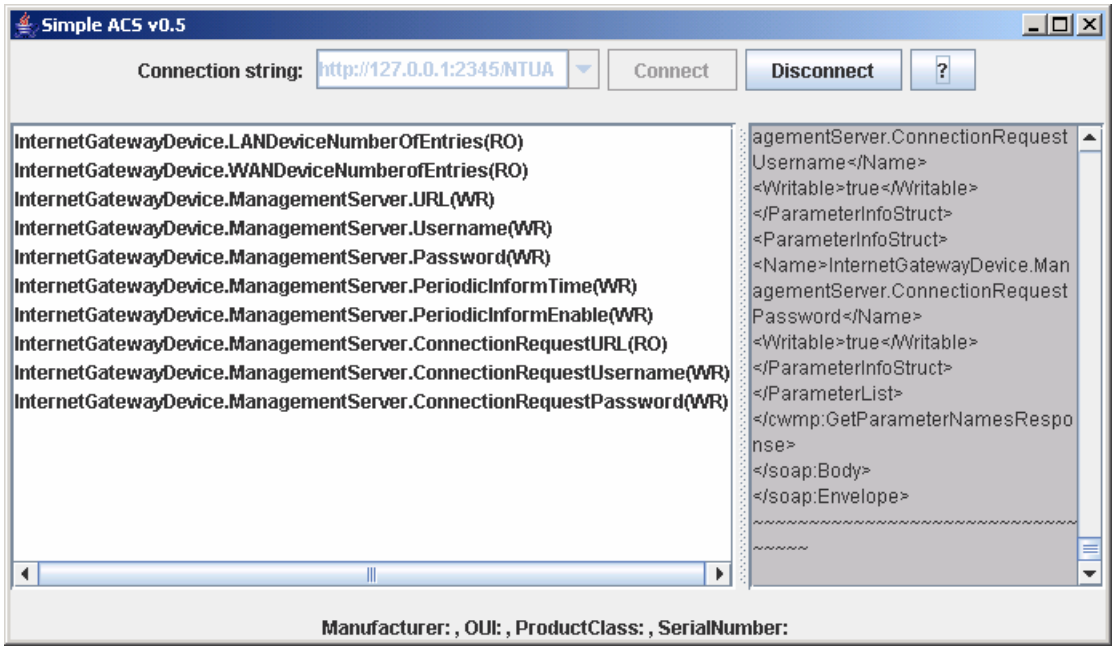
4.6.1.5. Στάδιο 5: δοκιμή λειτουργίας ενημέρωσης συμβάντων

Ο πράκτορας θα πρέπει επίσης να είναι ικανός να παρακολουθεί όλες τις παραμέτρους και να ενημερώνει το διαχειριστή σχετικά με τις μεταβολές τους. Έτσι λοιπόν ο *Device Tester* θα πρέπει καταρχάς να εγγράφεται στην υπηρεσία ενημέρωσης του πράκτορα για κάθε παράμετρο που αυτός υποστηρίζει. Στο σημείο αυτό θα πρέπει να τονιστεί ότι το συγκεκριμένο πρωτόκολλο απαγορεύει την ενημέρωση συμβάντων, που προκαλούνται από τον ACS διαχειριστή και για αυτό το λόγο ο *Device Tester* δεν μπορεί να προκαλέσει συστηματικό σκανδαλισμό με διαδοχικές μεθόδους ανάθεσης για κάθε μεταβλητή. Επομένως θα πρέπει στη συσκευή να εφαρμόζεται ένας «χειρονακτικός» σκανδαλισμός συμβάντων ανάλογα με τη λειτουργία της κάθε ομάδας παραμέτρων (είτε αυτή σχετίζεται με το υλικό, είτε με το δίκτυο, είτε με την εφαρμογή). Για παράδειγμα θα μπορούσαν να στέλνονται IP πακέτα στην Ethernet μονάδα μιας δικτυακής πύλης, τα οποία θα αλλάζουν τις παραμέτρους *BytesReceived* και *PacketsReceived*. Κατά τη διάρκεια των σκανδαλισμών η εφαρμογή θα καταγράφει τις αποκρίσεις του πράκτορα και έτσι θα ελέγχει την ικανότητα ενημέρωσης για κάθε παράμετρο. Μάλιστα ο *Device Tester* θα πρέπει να παρέχει ένα περιβάλλον στο οποίο ο χρήστης θα μπορεί να αντιστοιχεί σε κάθε ομάδα παραμέτρων το είδος του σκανδαλισμού, που πρέπει να πραγματοποιείται κάθε φορά για να παράγονται τα μηνύματα ενημέρωσης. Με αυτό τον τρόπο ο χρήστης - προγραμματιστής μπορεί να καθοδηγείται από τον *Device Tester* για την εφαρμογή των χειρονακτικών σκανδαλισμών.

4.6.1.6. Η εφαρμογή των δοκιμών

Στα πλαίσια της παρούσας διατριβής σχεδιάστηκε και αναπτύχθηκε μια εφαρμογή που υλοποιεί ένα μέρος των προτεινόμενων λειτουργιών δοκιμής. Η εφαρμογή ονομάζεται *Simple*

ACS και υποστηρίζει τις βασικές μεθόδους του CWMP πρωτοκόλλου. Επίσης υποστηρίζει την αμφίδρομη εκκίνηση της επικοινωνίας με τους CWMP πράκτορες. Όταν η εφαρμογή συνδέεται με τον πράκτορα καλεί αυτόματα τη μέθοδο *GetParameterNames*, που επιστρέφει το σύνολο των διαχειρίσιμων παραμέτρων. Στη συνέχεια η εφαρμογή εμφανίζει στα αριστερά της οθόνης (σχήμα 4-14) μια λίστα με όλες αυτές τις παραμέτρους. Έτσι ο χρήστης μπορεί να επιλέξει την παράμετρο, που επιθυμεί να θέσει ή να διαβάσει (Set/Get values). Οθόνες σαν εκείνες των σχημάτων 4-15 δίνουν ένα γραφικό περιβάλλον μέσω του οποίου ο χρήστης μπορεί να ελέγξει τη λειτουργία του ελέγχου, που αποτελεί τον πυρήνα της διαχείρισης.



Σχήμα 4-14. Simple ACS: Μια περιορισμένη έκδοση του Device Tester.

Πέραν της δοκιμής του ελέγχου, η εφαρμογή παρέχει ένα περιβάλλον καταγραφής και εμφάνισης (δεξιά της οθόνης του σχήματος 4-15) όλων των SOAP φακέλων, που ανταλλάσσονται μεταξύ του ACS και του πράκτορα. Έτσι ο χρήστης μπορεί να έχει πλήρη εποπτεία των καλούμενων μεθόδων και των ορισμάτων τους. Το γεγονός ότι οι SOAP φάκελοι είναι κωδικοποιημένοι στην αυτό-επεξηγούμενη γλώσσα XML δίνει σημαντικό πλεονέκτημα στο χρήστη για γρήγορη αποσφαλμάτωση (debugging).

Σχήμα 4-15. Simple ACS: Δοκιμή λήψης τιμών (Get).

4.6.2. Η συντήρηση λογισμικού του πράκτορα

Όλες οι διαδικασίες που περιγράφτηκαν μέχρι τώρα μπορούν να εφαρμοστούν όχι μόνο κατά την ανάπτυξη του προϊόντος αλλά και κατά τη συντήρηση του λογισμικού του. Όταν ο

κατασκευαστής αναβαθμίζει το λογισμικό της συσκευής με νέες λειτουργίες, που χρήζουν διαχείριση θα πρέπει να αναβαθμίσει και τον πράκτορα προκειμένου να υποστηρίξει τη ρύθμιση και την παρακολούθηση των νέων παραμέτρων της. Προτείνεται η εφαρμογή του *Device Creator* να παρέχει τη δυνατότητα στο χρήστη να προσθέτει νέες παραμέτρους από το αντίστοιχο μοντέλο (*CPE_DataModel.xml*) και έτσι να δημιουργείται ένα νέο XML κείμενο (*CPE_DeviceImplemented.xml*) το οποίο θα χρησιμοποιείται για την παραγωγή του σκελετού του νέου κώδικα. Μάλιστα η εφαρμογή του *Device Creator* θα δέχεται επιπλέον και τον παλαιό κώδικα του πράκτορα ώστε απλά να προσθέτει σε αυτόν το νέο αυτόματα παραγόμενο κώδικα. Μάλιστα η υποστήριξη μεθόδου για μεταφόρτωση αρχείων (μέθοδος *Download*, πίνακας 4.2) από το πρωτόκολλο διαχείρισης διευκολύνει σημαντικά και μειώνει το κόστος της αναβάθμισης του λογισμικού όχι μόνο του πράκτορα αλλά της συσκευής συνολικά.

4.6.3. Η αυτοματοποίηση της διαδικασίας ρύθμισης των συσκευών

Για λόγους πληρότητας πρέπει να σημειωθεί και η χρησιμότητα της XML τεχνολογίας στην πλευρά του διαχειριστή. Κάτι τέτοιο έχει ήδη αναγνωριστεί από πολλές εργασίες στο παρελθόν [11-14]. Έτσι λοιπόν η γλώσσα XML μπορεί να χρησιμοποιηθεί με στόχο τη δόμηση και την αποθήκευση των ρυθμίσεων των συσκευών. Μάλιστα η πληροφορία της ρύθμισης του συνολικού περιβάλλοντος θα μπορεί να βρίσκεται σε ένα και μόνο XML κείμενο (EnvConf.xml) το οποίο θα αποτελεί τη βάση στην οποία θα εφαρμόζεται μια σειρά XML μετασχηματισμών (XSLT κείμενα) [15,16] προκειμένου να διαφοροποιούνται οι ρυθμίσεις για κάθε συσκευή και υπηρεσία. Μια εφαρμογή (*Environment Configurator*) είναι δυνατόν παρέχει ένα περιβάλλον στο οποίο ο διαχειριστής μπορεί να εισάγει εύκολα τις συνολικές ρυθμίσεις του περιβάλλοντος για κάθε υπηρεσία και συσκευή.

4.7. Ανακεφαλαίωση των προτεινόμενων διαδικασιών και αποτίμηση της διαδικασίας υλοποίησης

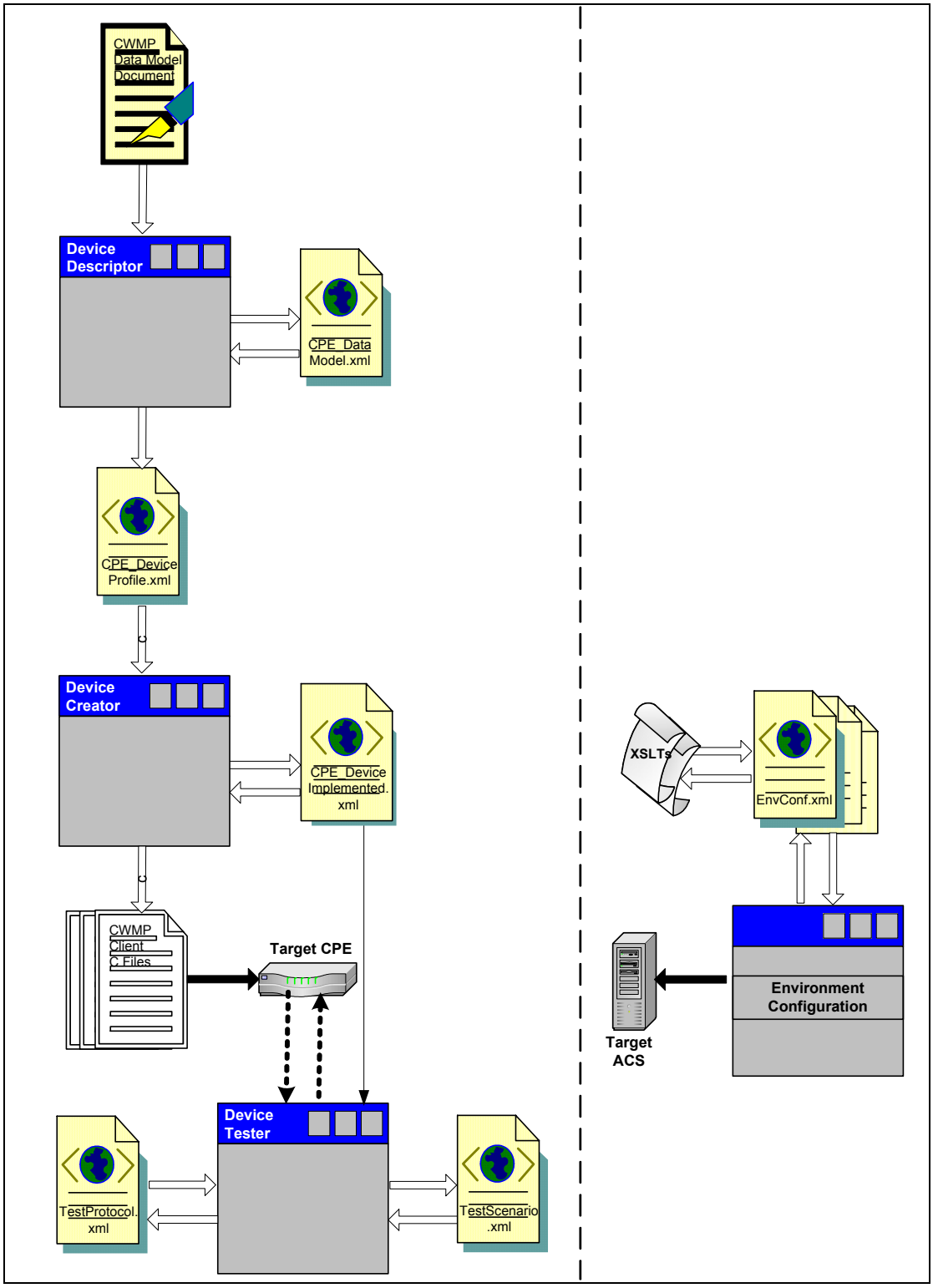
Στο αριστερό τμήμα του σχήματος 4-16 απεικονίζεται η συνολική διαδικασία της αυτοματοποιημένης ανάπτυξης και δοκιμής των CWMP πρακτόρων διαχείρισης. Η διαδικασία δίνεται σε διαδοχικά βήματα, ξεκινώντας από πάνω προς τα κάτω. Το πρώτο βήμα περιλαμβάνει τη μεταφορά και μετατροπή του μοντέλου της διαχειριζόμενης συσκευής από το αντίστοιχο κείμενο προτυποποίησης (*CWMP Data Model Document)* σε ένα XML κείμενο (*CPE_DataModel.xml*). Η διαδικασία αυτή επιτελείται μέσω του εργαλείου περιγραφής των συσκευών (*Device Descriptor)* με τον τρόπο που αναλύθηκε παραπάνω. Στη συνέχεια με βάση τις πραγματικές δυνατότητες της συσκευής, ο χρήστης – κατασκευαστής επιλέγει (μέσω της εφαρμογής *Selection* του σχήματος 4-9) τις διαχειρίσιμες παραμέτρους, που θα υποστηρίζει τελικά ο πράκτορας. Έτσι προκύπτει ένα νέο XML κείμενο (*CPE_DeviceProfile.xml*) το οποίο αποτελεί την είσοδο για το εργαλείο δημιουργίας της συσκευής (*Device Creator*).

Κατόπιν ο *Device Creator* εμφανίζει στο χρήστη όλες τις διαχειρίσιμες παραμέτρους της συσκευής (δηλαδή την πληροφορία του *CPE_DeviceProfile.xml*) και τον καλεί να

ομαδοποιήσει και να αντιστοιχίσει τις παραμέτρους αυτές σε αυθαίρετα οριζόμενες (από το χρήστη) μεθόδους ενορχήστρωσης. Με αυτό τον τρόπο γίνεται γνωστή στον προς ανάπτυξη πράκτορα η μέθοδος ενορχήστρωσης, που πρέπει να εκτελεστεί κάθε φορά που εκείνος καλείται να ανταποκριθεί σε μια απομακρυσμένη μέθοδο λήψης (*GetParameterValues*) ή ανάθεσης τιμής (*SetParameterValues*) του ACS διαχειριστή. Τελικά προκύπτει ένα νέο XML κείμενο (*CPE_DeviceImplemented.xml*), το οποίο φέρει την πληροφορία του *CPE_DeviceProfile.xml,* εμπλουτισμένη με τις αντίστοιχες μεθόδους ενορχήστρωσης για κάθε μεταβλητή.

Με τον τρόπο που περιγράφηκε στις προηγούμενες ενότητες, ο *Device Creator* χρησιμοποιεί το τελευταίο XML κείμενο και παράγει αυτόματα το σκελετό του πηγαίου κώδικα (*CWMP Client C Files*) για τον αντίστοιχο CWMP πράκτορα. Ο κώδικας αυτός εμπλουτίζεται από τον κατασκευαστή με τον κώδικα των μεθόδων ενορχήστρωσης και έτσι προκύπτει ο τελικός κώδικας του πράκτορα. Στη συνέχεια ο κώδικας αυτός «χτίζεται» με τα αντίστοιχα εργαλεία της κάθε συσκευής (Compiler και Linker) από τα οποία και προκύπτει το τελικό εκτελέσιμο αρχείο της εφαρμογής του πράκτορα. Όπως αναφέρθηκε και στην παράγραφο 4.6.2, η παραμετρική υλοποίηση σε συνδυασμό με τις δυνατότητες απομακρυσμένης μεταφόρτωσης λογισμικού, που παρέχει το πρωτόκολλο, διευκολύνει σημαντικά τη δια βίου συντήρηση των CWMP πρακτόρων.

Αμέσως μετά την υλοποίηση ακολουθεί το στάδιο των δοκιμών. Το εργαλείο δοκιμών (*Device Tester*) παρέχει ένα περιβάλλον στο οποίο ο χρήστης δίνει τις δοκιμές συμμόρφωσης με το πρωτόκολλο διαχείρισης (*TestProtocol.xml*) αλλά και τις ολοκληρωμένες δοκιμές διαχείρισης των υπηρεσιών (*TestScenario.xml*). Ο *Device Tester* παίζει το ρόλο του ACS, «καθοδηγείται» από τα δύο παραπάνω XML κείμενα, επικοινωνεί με τον πράκτορα της συσκευής και εκτελεί σειριακά όλες τις απαραίτητες δοκιμές. Μάλιστα για τους λόγους που αναφέρθηκαν στην παράγραφο 4.7.1 το εργαλείο των δοκιμών συμβουλεύεται το κείμενο υλοποίησης του πράκτορα της συσκευής (*CPE_DeviceImplemented.xml)*. Επίσης στο δεξιό τμήμα του σχήματος 4-16 απεικονίζεται η αυτοματοποιημένη διαδικασία ρύθμισης των συσκευών, που παρουσιάστηκε στην παράγραφο 4.7.3. Η διαδικασία αυτή είναι παρόμοια με τις ολοκληρωμένες δοκιμές διαχείρισης των υπηρεσιών του *Device Tester*, με τη μόνη διαφορά ότι εδώ το αντίστοιχο εργαλείο (*Environment Configuration Tool*) πρόκειται να εξυπηρετεί τις ανάγκες του διαχειριστή του δικτύου και των υπηρεσιών και όχι του κατασκευαστή της συσκευής. Αυτό έχει ως συνέπεια το εν λόγω εργαλείο να είναι προσανατολισμένο στη διαφοροποίηση (μέσω των XML μετασχηματισμών) και διαχείριση των ρυθμίσεων ανά συσκευή και ανά υπηρεσία.



Σχήμα 4-16. Οι αυτοματοποιημένες διαδικασίες και η χρήση των XML κειμένων σε καθεμία.

Στα πλαίσια της παρούσας εργασίας αναπτύχθηκαν πλήρως τα εργαλεία περιγραφής (*Device Descriptor*) και αυτόματης υλοποίησης (*Device Creator)* των παραμετρικών CWMP πρακτόρων, ενώ πραγματοποιήθηκε και μια στοιχειώδης υλοποίηση του εργαλείου δοκιμών (*Simple ACS*). Καθώς η παρούσα διατριβή εστιάζει κυρίως στους XML πράκτορες δεν

κρίθηκε σκόπιμη η υλοποίηση του εργαλείου *Environment Configuration*. Εξάλλου υπάρχουν αρκετές εργασίες στο παρελθόν που πραγματεύτηκαν τη χρήση της XML τεχνολογίας στην πλευρά του διαχειριστή [11-14].

Για την αποτίμηση των εργαλείων περιγραφής και αυτόματης υλοποίησης χρησιμοποιήθηκε η περίπτωση του πράκτορα μιας δικτυακής πύλης. Έγινε χρήση του προτυποποιημένου μοντέλου [4] και επιλέχθηκαν μόνο οι παράμετροι εκείνοι που αντιστοιχούν σε μια πύλη με (i) ένα ADSL διαμορφωτή / αποδιαμορφωτή (70 παράμετροι), (ii) μια Ethernet διεπαφή για το τοπικό δίκτυο (10 παράμετροι) και (iii) μια διεπαφή WiFi πάλι για το τοπικό δίκτυο (37 παράμετροι). Πέραν των παραπάνω χρειάστηκαν ακόμα 120 παράμετροι για το βασικό προφίλ της πύλης προκειμένου αυτή να συμμορφώνεται με το CWMP μοντέλο. Μάλιστα οι παραπάνω παράμετροι ομαδοποιήθηκαν σε 40 μεθόδους ενορχήστρωσης.

Η συνολική διάρκεια της διαδικασίας περιγραφής της συσκευής και υλοποίησής του πράκτορα (χωρίς τις μεθόδους ενορχήστρωσης) ήταν λιγότερο από οκτώ ώρες. Μάλιστα η διαδικασία της περιγραφής ήταν η πιο χρονοβόρα (περισσότερο από το 70% του συνολικού χρόνου), αφού έπρεπε να μεταφερθούν οι ιδιότητες 500 περίπου παραμέτρων από το αντίστοιχο κείμενο προτυποποίησης στην εφαρμογή του *Device Descriptor*. Ωστόσο η πληροφορία αυτή (*CPE_DataModel.xml*) μπορεί πλέον να επαναχρησιμοποιηθεί για όλες τις δικτυακές πύλες. Επίσης υλοποιήθηκαν και ορισμένες μέθοδοι ενορχήστρωσης. Η διαδικασία αυτή αποδείχτηκε η πιο χρονοβόρα από όλες αφού απαίτησε περίπου το 50% του συνολικού χρόνου υλοποίησης του πράκτορα (στην περίπτωση που δεν χρησιμοποιείται η εφαρμογή αυτοματοποίησης). Αυτό το τμήμα της υλοποίησης δεν μπορεί να αυτοματοποιηθεί διότι συναρτάται άμεσα με την πλατφόρμα του υλικού της συσκευής αλλά και με το λειτουργικό σύστημα που την υποστηρίζει.

Τέλος, θα πρέπει να τονιστεί ότι οι παραπάνω συσχετισμοί των χρόνων δεν θα πρέπει να θεωρηθούν σαν απόλυτα μεγέθη διότι η υλοποίηση των μεθόδων ενορχήστρωσης εξαρτάται άμεσα από τη γνώση της πλατφόρμας. Στην περίπτωση που ο χρήστης των προτεινόμενων εργαλείων ήταν ένας προγραμματιστής της κατασκευάστριας εταιρίας της συσκευής, οι χρόνοι υλοποίησης των μεθόδων ενορχήστρωσης θα μπορούσαν να είναι σημαντικά μειωμένοι, εξαιτίας της κεκτημένης γνώσης και εμπειρίας. Η αυτοματοποίηση λοιπόν της κατασκευής του σκελετού ενός CWMP πράκτορα όταν συνδυαστεί με την καλή γνώση της πλατφόρμας, μπορεί να δώσει το καλύτερο αποτέλεσμα και να απαλλάξει τον κατασκευαστή από τη διαδικασία εκμάθησης και υλοποίησης του πρωτοκόλλου διαχείρισης.

4.8. Συμπεράσματα - Παρατηρήσεις

Στην εποχή της ευρυζωνικότητας και του υψηλού βαθμού ολοκλήρωσης των κυκλωμάτων οι κατασκευαστές έρχονται αντιμέτωποι με νέα πρότυπα διαχείρισης και πολύπλοκες καταναλωτικές συσκευές. Μάλιστα η συμμόρφωση με τα εν λόγω πρότυπα γίνεται ολοένα και πιο σημαντική, καθώς στο μέλλον η παροχή των υπηρεσιών πρόκειται να επιτελείται από πολλούς διαφορετικούς παρόχους σε διαφορετικούς χρήστες και μάλιστα μέσω ενός πλήθους διαφορετικών συσκευών. Η πολυπλοκότητα των νέων συσκευών και υπηρεσιών δημιουργεί

ένα αναπόφευκτα μεγάλο αριθμό παραμέτρων, που χρειάζονται διαχείριση. Τα νέα πρότυπα αλλά και ο αυξημένος όγκος των καθηκόντων διαχείρισης επιφέρουν δυσκολίες στην υλοποίηση, στον έλεγχο και στη συντήρηση των πρακτόρων διαχείρισης. Η ιδέα της δημιουργίας ομάδων εργασίας με εξειδικευμένους μηχανικούς στα νέα πρότυπα και στις εφαρμογές, που υλοποιούν τις υπηρεσίες αναμένεται να ανεβάσει σημαντικά το κόστος διαχείρισης των συνδρομητικών συσκευών. Στην παρούσα εργασία προτάθηκε μια διαφορετική προσέγγιση, σύμφωνα με την οποία η εξειδικευμένη γνώση ενσωματώνεται σε έτοιμα εργαλεία, που απαλλάσσουν τους μηχανικούς σχεδίασης και εφαρμογών από τις πολλές λεπτομέρειες των σύγχρονων πρωτοκόλλων διαχείρισης. Μάλιστα δόθηκε ιδιαίτερη προσοχή στη μέγιστη δυνατή αυτοματοποίηση της συνολικής παραγωγικής διαδικασίας των πρακτόρων προκειμένου ένας πολύ μικρός αριθμός μηχανικών να είναι αρκετός για την υλοποίηση, τον έλεγχο και τη συντήρησή τους.

Κίνητρο της παρούσας εργασίας ήταν να αποδείξει ότι τα πλεονεκτήματα της XML τεχνολογίας δεν περιορίζονται μόνο στη δόμηση, μεταφορά και αποθήκευση της πληροφορίας διαχείρισης, κατά τη διάδραση πράκτορα-διαχειριστή. Η έκφραση των προτυποποιημένων μοντέλων διαχείρισης των συσκευών σε XML μορφή μπορεί να οδηγήσει σε πλήρως αυτοματοποιημένες λύσεις υλοποίησης, δοκιμών και συντήρησης του λογισμικού των ίδιων των πρακτόρων, που εξοπλίζουν τις συσκευές. Μάλιστα κατά την αυτοματοποίηση εμπλέκεται ένα πλήθος XML κειμένων, καθένα εκ των οποίων πηγάζει από το αρχικό κείμενο του μοντέλου της συσκευής και εξυπηρετεί μια ξεχωριστή διαδικασία. Παρόλο που η παρούσα εργασία πρότεινε ένα ολοκληρωμένο σύνολο από διαδικασίες αυτοματοποίησης, θα μπορούσε κανείς να σκεφτεί και να προτείνει ένα πλήθος εναλλακτικών μεθόδων με στόχο την αύξηση του βαθμού αυτοματοποίησης και τελικά την περαιτέρω μείωση του συνολικού κόστους εφαρμογής των νέων προτύπων διαχείρισης. Ωστόσο η συμβολή της παρούσας εργασίας έγκειται κυρίως στον τρόπο με τον οποίο μπορεί κανείς να εκφράσει και να εκμεταλλευτεί το XML μοντέλο διαχείρισης της συσκευής προκειμένου να αυτοματοποιήσει ένα πλήθος διαδικασιών, που σχετίζονται με τη διαχείριση.

4.9. Βιβλιογραφία - Αναφορές

[1] DSL Forum, "CPE WAN management protocol", Technical Report. 069, May 2004.

[2] DSL Forum, "LAN-side CPE configuration", Technical Report 064, May 2004.

[3] DSL Forum, "TR-064 extensions for service differentiation", Technical Report 133, September 2005.

[4] DSL Forum, "Internet Gateway Device v.1.1 data model", Technical Report. 098, September 2005.

[5] DSL Forum, "Provisioning parameters for VoIP CPE", Technical Report. 104, September 2005.

[6] DSL Forum, "Data model for a TR-069 enabled STB", Working Text. 135, May 2005.

[7] DSL Forum, "Data model for a Network Attached Strorage (NAS)", Working Text. 140, May 2005.

[8] UPnP Forum, Internet Gateway Device (IGD) V 1.0, http://www.upnp.org/standardizeddcps/igd.asp.

[9] UPnP Forum, "UPnP device architecture 1.0", December 2003, http://www.upnp.org.

[10] DSL Forum, "Data model template for TR-069 enabled devices", Technical Report 106, September 2005.

[11] JP Martin-Flatin, "Web-Based Management of IP Networks and Systems", Publisher: John Wiley and Sons, 2003.

[12] J-H. Yoon, H-T.Ju and J.W.Hong, "Development of SNMP-XML Translator and Gateway for XML-based Integrated Network Management", International Journal of Network Management, vol. 13, 2003, pp. 259-276.

[13] M.-J.Choi, J. Hong, H-T.Ju, "XML-Based Network Management for IP Networks", ETRI Journal, vol 25, no 6, December 2003.

[14] J. Schoenwaelder, A. Pras, and J.-P. Martin-Flatin, "On the Future of Internet Management Technologies," IEEE Communications Magazine, vol. 41, no. 10, Oct. 2003.

[15] W3C, "Extensible Stylesheet Language (XSL) Version 1.0," W3C Recommendation, Nov. 2000.

[16] W3C, "XSL Transformations Version 1.0," W3C Recommendation, Nov. 1999.

Κεφάλαιο 5
Επέκταση της Διαλειτουργικότητας των SOAP/XML/ HTTP Προτύπων Απομακρυσμένης Διαχείρισης

5.1. Εισαγωγή

Όπως έγινε φανερό και από το δεύτερο κεφάλαιο, η γλώσσα XML μπορεί να περιγράφει πολύπλοκες δομές και να ενσωματώνεται με μεγάλη ευκολία στις εφαρμογές διαχείρισης. Μάλιστα, καθημερινά γίνεται ολοένα και πιο οικεία στους προγραμματιστές, ενώ η διαρκής ανάπτυξη νέων εργαλείων και σχετικών τεχνολογιών οδηγεί σε λύσεις χαμηλού κόστους. Όλα αυτά παρέχουν στην XML το προβάδισμα για την επικράτησή της στο χώρο της διαχείρισης και της αυτόματης ρύθμισης. Τα τελευταία δέκα χρόνια έγιναν αρκετές εξελικτικές προσπάθειες χωρίς ωστόσο να καταλήξουν σε προτυποποίηση. Η εξελικτική προσέγγιση εστίασε κυρίως στην ολοκλήρωση της πληροφορίας διαχείρισης, που παρέχεται από τους SNMP πράκτορες σε XML συστήματα διαχείρισης. Η προσέγγιση αυτή προωθήθηκε από τους παρόχους, οι οποίοι επιθυμούσαν να επωφεληθούν σύντομα από τα πλεονεκτήματα που έχει η XML στην επεξεργασία της πληροφορίας διαχείρισης. Με αυτό τον τρόπο θα έλυναν χρόνια προβλήματα όπως εκείνο της διαχείρισης των ρυθμίσεων, μέσω της XML και των XSL μετασχηματισμών. Το κύριο μέλημα των προσεγγίσεων αυτών ήταν ο αποτελεσματικός μετασχηματισμός των SMI ορισμών των βάσεων διαχείρισης (MIB) σε XML σχήματα. Αυτό συνεπάγεται την απεικόνιση της SNMP πληροφορίας διαχείρισης σε XML κείμενα. Καθώς το SNMP είναι ένα απλό σχετικά πρωτόκολλο, οι προτεινόμενες SNMP-XML πύλες χρειάζονταν να λύσουν θέματα που σχετίζονταν μόνο με τις λειτουργίες του έλεγχου (Control – SNMP Get/Set) και της αναγγελίας των συμβάντων (Eventing – SNMP Trap).

Τα τελευταία πέντε χρόνια και αφού μέχρι τότε η εξελικτική προσέγγιση δεν είχε οδηγήσει σε κάποιο πρότυπο, οι προσπάθειες στράφηκαν προς τις ριζοσπαστικές λύσεις, δηλαδή σε εκείνες που προέβλεπαν την πλήρη αντικατάσταση του SNMP από τεχνολογίες βασισμένες στην XML. Αυτό ισοδυναμούσε με την αντικατάσταση των SNMP πρακτόρων από αμιγείς XML/HTTP πράκτορες. Σήμερα οι βασικότερες προσπάθειες προς αυτήν την κατεύθυνση είναι το NetConf (Network Configuration Protocol) [1] πρωτόκολλο της IETF και το CWMP (CPE WAN Management Protocol) [2] πρωτόκολλο του DSL forum. Όσον αφορά στο πρώτο, αρκετά σημαντικά θέματα εκκρεμούν [3]. Από την άλλη πλευρά οι μηχανισμοί που προβλέπει το CWMP φαίνεται να είναι αρκετά ωριμότεροι για να υποστηρίξουν τη

διαχείριση των συνδρομητικών δικτύων αλλά και των συσκευών, που αποδίδουν τις υπηρεσίες στους τελικούς χρήστες. Η κλιμακώσιμη αρχιτεκτονική [2], οι εκτεταμένες προσπάθειες για την προτυποποίηση των μοντέλων διαχείρισης μια μεγάλης γκάμας συνδρομητικού εξοπλισμού (δικτυακές πύλες, συνδρομητικοί αποκωδικοποιητές, παιχνιδομηχανές, εξυπηρετητές πολυμεσικού περιεχομένου κ.α.) καθώς επίσης και η ταχεία υιοθέτηση του CWMP από μεγάλους κατασκευαστές της Ευρώπης και της Αμερικής, έχουν δημιουργήσει πολλές προοπτικές για την ευρεία αποδοχή του.

Το Μάιο του 2004 δημοσιεύτηκε το CWMP πρωτόκολλο διαχείρισης μαζί με το πρώτο μοντέλο διαχείρισης της δικτυακής πύλης [2]. Από τότε έχει γίνει σημαντική πρόοδος σχετικά με τη μοντελοποίηση του υπόλοιπου συνδρομητικού εξοπλισμού, χωρίς ωστόσο ακόμα να έχει ολοκληρωθεί η προτυποποίηση άλλων συσκευών πέραν της δικτυακής πύλης. Έτσι σήμερα οι μοναδικές συσκευές στην αγορά, που υποστηρίζουν τη CWMP τεχνολογία είναι οι δικτυακές πύλες. Το γεγονός αυτό καθυστερεί την ανάπτυξη και αξιοποίηση πολλών υπηρεσιών, που απαιτούν τη ρύθμιση και την παρακολούθηση όχι μόνο της δικτυακής πύλης αλλά και άλλων συσκευών που συνδέονται σε αυτή. Ορισμένες από αυτές είναι τα σημεία ασύρματης πρόσβασης, οι συσκευές αυτοματισμού κτιριακών εγκαταστάσεων οι εξυπηρετητές πολυμεσικού περιεχομένου, οι συσκευές απόδοσης πολυμέσων κ.α. Επιπλέον μεγάλο τμήμα του εξοπλισμού που υπάρχει αυτή τη στιγμή στα δίκτυα των συνδρομητών υποστηρίζει το UPnP πρωτόκολλο, το οποίο είναι βασισμένο και αυτό στην XML τεχνολογία, αλλά το εύρος δράσης του περιορίζεται στην αυτόματη ανίχνευση και ρύθμιση των συσκευών από το τοπικό δίκτυο. Τα τελευταία έξι χρόνια το UPnP πρωτόκολλο έχει γνωρίσει μεγάλη επιτυχία και έχει γίνει ευρέως αποδεκτό από κατασκευαστές πολλών διαφορετικών τύπων συσκευών (π.χ. εκτυπωτές, σαρωτές, σημεία ασύρματης πρόσβασης, πολυμεσικούς εξυπηρετητές, συσκευές απόδοσης πολυμέσων, συστήματα αυτοματισμού κτιριακών εγκαταστάσεων κ.α.).

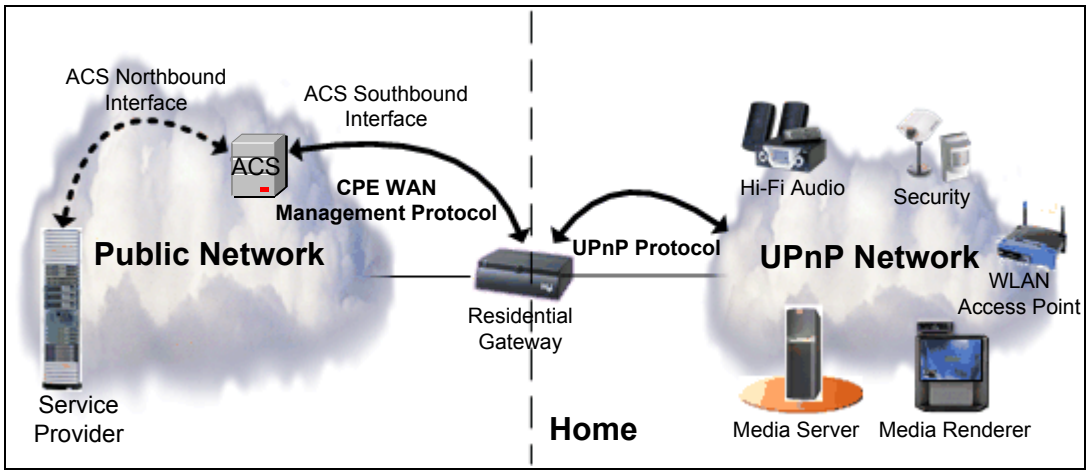
Επομένως η συνδυασμένη χρήση του CWMP και του UPnP πρωτοκόλλου θα μπορούσε να επεκτείνει το πεδίο διαχείρισης των παρόχων στις UPnP συσκευές. Κάτι τέτοιο θα επέτρεπε την ανάπτυξη πολλών υπηρεσιών, πριν ακόμα ολοκληρωθεί η αργή προτυποποίηση της μοντελοποίησης του συνδρομητικού εξοπλισμού με βάση το πρωτόκολλο απομακρυσμένης διαχείρισης CWMP. Επιπρόσθετα, μια τέτοια προσέγγιση θα μπορούσε να εκμηδενίσει την ανάγκη για κάποιες UPnP συσκευές με περιορισμένους πόρους (π.χ. σημεία ασύρματης πρόσβασης) να υποστηρίζουν ένα επιπρόσθετο πρωτόκολλο για απομακρυσμένη διαχείριση. Κάτι τέτοιο εξοικονομεί πόρους και συμβάλλει στη διατήρηση του χαμηλού κόστους αυτού του εξοπλισμού.

Η παρούσα εργασία διερευνά τη φύση των δύο πρωτοκόλλων και προσδιορίζει τις απαραίτητες λειτουργίες γεφύρωσης, λαμβάνοντας υπόψη τις απαιτήσεις για διαφάνεια και κλιμακωσιμότητα που πηγάζουν κυρίως από την πλευρά των παρόχων. Πειραματικά αποτελέσματα υποστηρίζουν την εφαρμοσιμότητα και τα πλεονεκτήματα της προτεινόμενης προσέγγισης, υπογραμμίζουν τη σημασία του φιλτραρίσματος των μηνυμάτων και φωτίζουν τις πολύπλοκες λειτουργίες διάδρασης αυτού του είδους των μηχανισμών διαχείρισης. Αρχικά δίνεται το περιβάλλον διαχείρισης καθώς και οι διαφορές των προς γεφύρωση

πρωτοκόλλων. Στη συνέχεια αναλύονται οι απαιτήσεις που πηγάζουν από τις παραπάνω διαφορές και προτείνονται οι λειτουργίες γεφύρωσης. Παράλληλα, για κάθε προτεινόμενη λειτουργία τεκμηριώνεται η συμμόρφωσή της στις παραπάνω απαιτήσεις. Τέλος, περιγράφεται η υλοποίηση ενός πρωτοτύπου και η διεξαγωγή πειραμάτων με στόχο τη συνολική αποτίμηση της προτεινόμενης λύσης.

5.2. Το περιβάλλον διαχείρισης

Ένα δίκτυο συνδρομητή αποτελείται τυπικά από διάφορες συσκευές συνδεδεμένες μεταξύ τους μέσω μιας ή περισσοτέρων τεχνολογιών τοπικής δικτύωσης (π.χ. Ethernet, 802.11, HomePlug και HomePNA). Η δικτυακή πύλη είναι συνδεδεμένη τόσο στο δίκτυο πρόσβασης (π.χ. xDSL) όσο και στο τοπικό δίκτυο και αναλαμβάνει την παροχή διαδικτυακής πρόσβασης στις τοπικές συσκευές. Το υποσύνολο των UPnP συνδρομητικών συσκευών ονομάζεται UPnP δίκτυο και απεικονίζεται στα δεξιά του σχήματος 5-1.



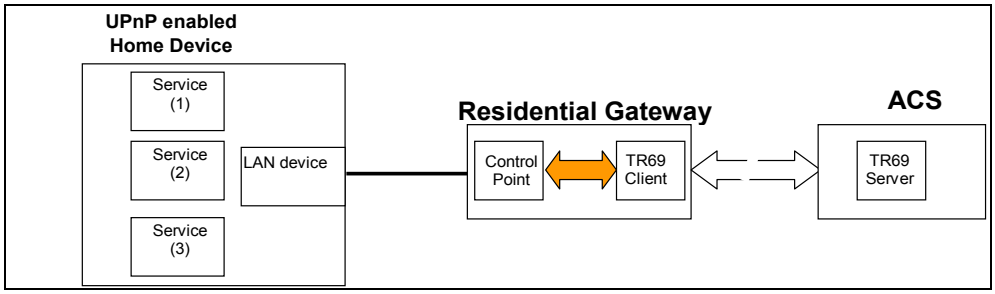
Σχήμα 5-1. Η συνύπαρξη των πρωτοκόλλων τοπικής και απομακρυσμένης διαχείρισης.

Τυπικά οι UPnP συνδρομητικές συσκευές περιλαμβάνουν ένα πράκτορα, οποίος στην UPnP αρχιτεκτονική ονομάζεται συσκευή (UPnP Device). Στόχος είναι η ρύθμιση και ο έλεγχος των παραπάνω συσκευών είτε από ένα προσωπικό υπολογιστή, είτε από μια UPnP συσκευή τηλεχειρισμού. Επομένως μια δικτυακή πύλη θα μπορούσε να επικοινωνήσει με τις συσκευές αυτές και να τις διαχειριστεί υπό την προϋπόθεση, ότι η πύλη θα ενσωματώνει ένα UPnP σημείο ελέγχου (UPnP Control Point), δηλαδή τη διαχειρίζουσα οντότητα του UPnP.

Από την άλλη πλευρά, μια δικτυακή πύλη που ενσωματώνει ένα CWMP πράκτορα (πελάτη στην αρχιτεκτονική του CWMP) είναι πλήρως διαχειρίσιμη από τον πάροχο μέσω του εξυπηρετητή αυτόματης ρύθμισης (ACS – Automatic Configuration Server), ο οποίος παίζει το ρόλο της διαχειρίζουσας οντότητας στο CWMP. Σύμφωνα με την αρχιτεκτονική του CWMP (αριστερά του σχήματος 5-1), ένας ACS διαχειριστής φιλοξενείται είτε στο δίκτυο πρόσβασης, είτε στο δίκτυο κορμού και είναι επιφορτισμένος με την εφαρμογή των αρχικών ρυθμίσεων και την περαιτέρω διαχείριση ενός μεγάλου αριθμού δικτυακών πυλών μέσω του CWMP πρωτοκόλλου. Ο ACS έχει δύο διεπαφές που ονομάζονται “Northbound” (Βόρειο σύνορο) και “Southbound” (Νότιο σύνορο). Η Southbound διεπαφή εξυπηρετεί την

επικοινωνία του ACS με τις πύλες και σχετίζεται με την παρούσα εργασία. Αντίθετα η Northbound διεπαφή εξυπηρετεί την ανταλλαγή της πληροφορίας διαχείρισης μεταξύ του ACS και των παρόχων, αφού οι τελευταίοι είναι εκείνοι που προσφέρουν τις υπηρεσίες και εφαρμόζουν τις υψηλού επιπέδου πολιτικές διαχείρισης. Η διεπαφή αυτή δεν θα απασχολήσει την παρούσα εργασία.

Επομένως πέραν των δικτυακών πυλών, οι πάροχοι θα μπορούσαν εμμέσως να διαχειριστούν επίσης και τις UPnP συσκευές. Αυτό γίνεται ως ακολούθως : (i) ο ACS στέλνει την εντολή διαχείρισης στον CWMP πελάτη, που βρίσκεται εντός της πύλης, (ii) η γέφυρα που βρίσκεται ανάμεσα στο CWMP πελάτη και στο UPnP σημείο ελέγχου μεταφέρει αυτή την πληροφορία εσωτερικά στο UPnP σημείο ελέγχου της πύλης και (iii) το UPnP σημείο ελέγχου αναλαμβάνει τη ρύθμιση και τον έλεγχο του UPnP δικτύου. Ο ρόλος της κάθε ομάδας λειτουργιών απεικονίζεται στο σχήμα 5-2.



Σχήμα 5-2. Ο ACS διαχειρίζεται τις UPnP συσκευές μέσω της δικτυακής πύλης.

Στο σημείο αυτό πρέπει να τονιστεί ότι δεν είναι απαραίτητο η γέφυρα να φιλοξενείται στη δικτυακή πύλη. Η ίδια αρχή ισχύει και για κάθε άλλη συσκευή που υποστηρίζει το πρωτόκολλο CWMP και είναι προσβάσιμη από τον ACS. Αυτό πρακτικά σημαίνει ότι όταν στο μέλλον προκύψουν και άλλες CWMP συσκευές, τότε η γέφυρα θα μπορεί εναλλακτικά να φιλοξενείται σε μία από αυτές αντί στη δικτυακή πύλη, χωρίς καμία αλλαγή στις λειτουργίες που αναφέρονται παρακάτω. Παρόλα αυτά η δικτυακή πύλη παρέχει καλύτερη συνδεσιμότητα αφού είναι άμεσα προσπελάσιμη από τον ACS, χωρίς να μεσολαβεί κάποιος μηχανισμός μετάφρασης των δικτυακών διευθύνσεων (NAT – Network Address Translation) ή κάποιο τείχος προστασίας (Firewall). Ακριβώς για να τονιστεί αυτή η γενικότητα, από εδώ και στο εξής η συσκευή που θα φιλοξενεί την εν λόγω γέφυρα θα προσδιορίζεται με το γενικότερο όρο «συνδρομητικός εξοπλισμός» (CPE) αντί του όρου «δικτυακή πύλη».

5.3. Η UPnP τεχνολογία τοπικής διαχείρισης

Η τεχνολογία τοπικής διαχείρισης UPnP ορίζει μία αρχιτεκτονική για διεισδυτική και ομότιμη δικτύωση μιας μεγάλης γκάμας συνδρομητικού εξοπλισμού. Έχει σχεδιαστεί για να συνδράμει αποφασιστικά στην εφαρμογή εύχρηστης και σύμφωνης με τα πρότυπα διασύνδεσης είτε σε ad-hoc και μη διαχειριζόμενα δίκτυα όπου αυτά εμφανίζονται, όπως στο οικιακό περιβάλλον, σε μικρές επιχειρήσεις και σε δημόσιους χώρους ή σε τοπικά δίκτυα που συνδέονται στο διαδίκτυο. Η UPnP τεχνολογία προσφέρει μία κατανεμημένη, ανοιχτή αρχιτεκτονική δικτύωσης που είναι σε θέση να συγκεράσει τις δυνατότητες των TCP/IP

πρωτοκόλλων και των τεχνολογιών του διαδικτύου για να ενδυναμώσει τη διάφανη δικτύωση, τον έλεγχο και τη μεταφορά δεδομένων ανάμεσα σε δικτυωμένες συσκευές.

Η UPnP αρχιτεκτονική δεν είναι μόνο μία απλή επέκταση του μοντέλου ενός plug-and-play περιφερειακού. Έχει σχεδιαστεί για να υποστηρίζει ρύθμιση από μηδενική βάση, διαφανή δικτύωση και αυτόματη ανίχνευση για ένα μεγάλο εύρος κατηγοριών συσκευών από ένα ευρύτατο πεδίο κατασκευαστών. Αυτό σημαίνει ότι μία συσκευή μπορεί δυναμικά να προσχωρήσει σε ένα δίκτυο, να πάρει μία IP διεύθυνση, να ανακοινώσει τις δυνατότητές της και να μάθει για την ύπαρξη και τις δυνατότητες άλλων συσκευών. Μάλιστα με τη χρήση της UPnP τεχνολογίας, μία συσκευή μπορεί να αποχωρήσει από ένα δίκτυο ομαλά και αυτόματα χωρίς να αφήσει κάποια ανεπιθύμητη κατάσταση πίσω.

Οι τεχνολογίες που χρησιμοποιούνται στα πλαίσια της UPnP αρχιτεκτονικής περιλαμβάνουν διαδικτυακά πρωτόκολλα όπως είναι το TCP/IP, το UDP, το HTTP και η XML. Η χρησιμοποίηση των παραπάνω ευρέως χρησιμοποιούμενων και προτυποποιημένων τεχνολογιών δίνει στο UPnP τη δυνατότητα εφαρμογής της σε πολλά διαφορετικά φυσικά μέσα (αφού το UPnP είναι ανεξάρτητο του φυσικού μέσου) και παράλληλα ενισχύει τη διαλειτουργικότητα μεταξύ διαφορετικών συσκευών (διαφορετικός τύπος ή κατασκευαστής). Επίσης επιτυγχάνει τη συνέργια με το διαδίκτυο αλλά και πολλά εσωτερικά δίκτυα, τόσο οικιακά όσο και επιχειρησιακά. Άλλωστε ο σχεδιασμός της UPnP τεχνολογίας έχει σαν κύριο στόχο του τα παραπάνω περιβάλλοντα. Μάλιστα οι UPnP συσκευές (πράκτορες διαχείρισης) μπορούν να υλοποιηθούν με τη χρήση οποιασδήποτε γλώσσας προγραμματισμού και για οποιοδήποτε λειτουργικό σύστημα. Η UPnP αρχιτεκτονική δεν καθορίζει ούτε περιορίζει τη σχεδίαση της προγραμματιστικής διεπαφής (API - Application Programming Interface) για την εφαρμογή διαχείρισης.

Η UPnP αρχιτεκτονική ορίζει δύο βασικές κατηγορίες συσκευών. Η πρώτη αποτελείται από τις διαχειριζόμενες συσκευές (Devices), ενώ η δεύτερη από τις διαχειρίζουσες συσκευές (Control Points – σημεία ελέγχου). Η διαχειριζόμενη συσκευή ενσωματώνει ένα HTTP εξυπηρετητή, ο οποίος ανταποκρίνεται στις αιτήσεις των HTTP πελατών που βρίσκονται στα σημεία ελέγχου. Οι συσκευές και των δύο κατηγοριών μπορούν να υλοποιηθούν σε διάφορες πλατφόρμες, συμπεριλαμβανομένων των προσωπικών υπολογιστών και των ενσωματωμένων συστημάτων. Στο ίδιο μηχάνημα μπορούν να λειτουργούν ταυτόχρονα τόσο η UPnP συσκευή όσο και το UPnP σημείο ελέγχου επιτρέποντας έτσι στις συνδρομητικές συσκευές να είναι ταυτόχρονα διαχειριστές αλλά και διαχειριζόμενοι.

Το UPnP πρωτόκολλο περιλαμβάνει τις ακόλουθες έξι βασικές λειτουργίες:

- Τη διευθυνσιοδότηση (device addressing), κατά την οποία μία συσκευή πρέπει να λάβει μία IP διεύθυνση για να εισέλθει στο τοπικό IP δίκτυο.
- Την ανίχνευση (device/service discovery), κατά την οποία τα σημεία ελέγχου αναζητούν τις διαχειρίσιμες UPnP συσκευές αλλά και τις υπηρεσίες διαχείρισης και ελέγχου που αυτές προσφέρουν. Αντίστοιχα, οι UPnP συσκευές ανακοινώνουν την παρουσία τους κάθε φορά που εισέρχονται στο τοπικό δίκτυο.

- Την περιγραφή (device/service description), κατά την οποία ένα σημείο ελέγχου προσπαθεί να ανακτήσει από μία UPnP συσκευή την πλήρη περιγραφή της. Η περιγραφή συνίσταται στις λειτουργικές μονάδες της συσκευής και στις υπηρεσίες ελέγχου και διαχείρισης που προσφέρει η κάθε μονάδα.
- Τον έλεγχο (control), κατά τον οποίο ένα σημείο ελέγχου μπορεί να ελέγξει μία ή περισσότερες UPnP συσκευές.
- Την ενημέρωση συμβάντος (eventing), κατά την οποία τα σημεία ελέγχου ενημερώνονται για την αλλαγή κατάστασης των υπηρεσιών ελέγχου και διαχείρισης για τις οποίες ενδιαφέρονται.
- Την παρουσίαση (presentation), κατά την οποία η UPnP συσκευή προσφέρει στο χρήστη μια διεπαφή σε μορφή HTML κειμένων με όλες τις ελέγξιμες υπηρεσίες της.

Για όλες τις παραπάνω λειτουργίες η πρότυπη αρχιτεκτονική μίας UPnP συσκευής χρησιμοποιεί την ακόλουθη γενική στοίβα πρωτοκόλλων:

<table>
<tr><td colspan="5">UPnP Vendor</td></tr>
<tr><td colspan="5">UPnP Forum</td></tr>
<tr><td colspan="5">UPnP Device Architecture</td></tr>
<tr><td>GENA</td><td>SSDP</td><td>SSDP</td><td>SOAP</td><td>GENA</td></tr>
<tr><td colspan="2">HTTPMU</td><td>HTTPU</td><td>HTTP</td><td>HTTP</td></tr>
<tr><td colspan="3">UDP</td><td colspan="2">TCP</td></tr>
<tr><td colspan="5">IP</td></tr>
</table>

Σχήμα 5-3. Η στοίβα πρωτοκόλλων του UPnP μηχανισμού τοπικής διαχείρισης.

Στο υψηλότερο επίπεδο, τα μηνύματα περιέχουν μόνο την πληροφορία των κατασκευαστών για τις UPnP συσκευές τους. Χαμηλότερα στην στοίβα πρωτοκόλλων, τα μηνύματα αυτά συμπληρώνονται από πληροφορία που προέρχεται από το UPnP Forum και την προτυποποιημένη UPnP αρχιτεκτονική. Ανάλογα με τη UPnP λειτουργία που εκτελείται, τα μηνύματα εισάγονται σε πρωτόκολλα, που σχετίζονται με τη UPnP αρχιτεκτονική, όπως είναι το SSDP (Simple Service Discovery Protocol) [4] και το General Event Notification Architecture (GENA) [4]. Τα μηνύματα, στη συνέχεια, μεταφέρονται είτε από τα διαφοροποιημένα πρωτόκολλα του http για μοναδικό (HTTPU – HTTP unicast) [5] και πολλαπλούς παραλήπτες (HTTPMU – HTTP multicast), είτε μέσω του κανονικού HTTP πρωτοκόλλου. Τελικά, όλα τα μηνύματα ανταλλάσσονται κατά τη διάρκεια των UPnP λειτουργιών πάνω από το IP πρωτόκολλο. Στη συνέχεια αναλύονται οι έξι διαφορετικές λειτουργίες της UPnP αρχιτεκτονικής.

5.3.1. Διευθυνσιοδότηση

Η βάση της UPnP δικτύωσης είναι η IP διευθυνσιοδότηση (device addressing). Μέσω της διευθυνσιοδότησης, οι συσκευές παίρνουν μία διεύθυνση δικτύου η οποία τους επιτρέπει την ανίχνευση, κατά την οποία τα σημεία ελέγχου ανιχνεύουν «ενδιαφέρουσες» συσκευές, την περιγραφή, κατά την οποία τα σημεία ελέγχου μαθαίνουν για τις δυνατότητες μίας συσκευής, τον έλεγχο, κατά τον οποίο τα σημεία ελέγχου στέλνουν εντολές στις συσκευές, την καταγραφή συμβάντων, κατά την οποία τα σημεία ελέγχου καταλαβαίνουν τις αλλαγές κατάστασης σε συσκευές και, τέλος, την παρουσίαση, κατά την οποία τα σημεία ελέγχου εμφανίζουν μία διεπαφή χρήστη για τις συσκευές. Μία UPnP συσκευή μπορεί να υποστηρίζει το δικτυακό πρωτόκολλο IPv4, το IPv6, ή ακόμα και τα δύο.

Κάθε UPnP συσκευή πρέπει να έχει ένα DHCP (Dynamic Host Configuration Protocol) πελάτη και να ψάχνει για ένα DHCP εξυπηρετητή όταν αυτή συνδέεται για πρώτη φορά στο δίκτυο. Στην περίπτωση που υλοποιεί η ίδια έναν DHCP εξυπηρετητή, θα μπορεί να δεσμεύει μία διεύθυνση από την τράπεζα διευθύνσεων που κατέχει. Αν είναι διαθέσιμος ένας DHCP εξυπηρετητής, τότε η συσκευή θα πρέπει να χρησιμοποιήσει την διεύθυνση που της αποδόθηκε. Αν δεν βρεθεί εξυπηρετητής, η συσκευή θα πρέπει να χρησιμοποιήσει αυτόματη IP διευθυνσιοδότηση (Auto-IP) για να λάβει μία διεύθυνση.

Το Auto-IP πρωτόκολλο καθορίζει τον τρόπο με τον οποίο η συσκευή καταλήγει στο ότι δεν υπάρχει διαθέσιμος DHCP εξυπηρετητής και διαλέγει μία IP διεύθυνση από μία ομάδα διευθύνσεων τοπικής διασύνδεσης. Αυτή η μέθοδος της απόδοσης διευθύνσεων καθιστά τη συσκευή ικανή να μετακινείται εύκολα ανάμεσα σε διαχειριζόμενα και μη δίκτυα.

Τα βήματα της διαδικασίας διευθυνσιοδότησης μίας συσκευής είναι τα ακόλουθα:

- Η συσκευή καθορίζει αν πρέπει ή όχι να χρησιμοποιήσει το Auto-IP πρωτόκολλο για να πάρει διεύθυνση. Ενεργοποιεί τον DHCP πελάτη για να ανιχνεύσει διαθέσιμους εξυπηρετητές. Αν λάβει έγκαιρα ένα DHCPOFFER μήνυμα, ολοκληρώνει τη διαδικασία δυναμικής ανάθεσης διεύθυνσης. Αν δεν βρεθεί κανένας εξυπηρετητής, τότε η συσκευή πρέπει να ενεργοποιήσει το Auto-IP πρωτόκολλο αυτοδιάθεσης IP διεύθυνσης.
- Η συσκευή χρησιμοποιεί έναν οποιοδήποτε αλγόριθμο για να διαλέξει μία διεύθυνση στο πεδίο 169.254/16.
- Η συσκευή ελέγχει αν η διεύθυνση που επέλεξε στο προηγούμενο βήμα χρησιμοποιείται μοναδικά. Για το σκοπό αυτό χρησιμοποιεί ένα ARP (Address Resolution Protocol) μήνυμα διερεύνησης. Σε περίπτωση που η διεύθυνση χρησιμοποιείται, η συσκευή πρέπει να προσπαθήσει με μία άλλη διεύθυνση.
- Η συσκευή θα πρέπει ακόμα και μετά την επιτυχή αυτοδιάθεση διεύθυνσης, να ελέγχει περιοδικά τη διαθεσιμότητα ενός DHCP εξυπηρετητή. Η συχνότητα ελέγχου εξαρτάται από την υλοποίηση, αλλά ένας έλεγχος κάθε 5 λεπτά θα ήταν ιδανικός για την εξασφάλιση ισορροπίας μεταξύ του απαιτούμενου εύρους ζώνης δικτύου και της διατήρησης της διασύνδεσης.

- Η συσκευή θα πρέπει να δίνει την δυνατότητα σε χρήστες να έχουν πρόσβαση σε αυτήν χρησιμοποιώντας ένα φιλικό όνομα αντί της δύσχρηστης IP διεύθυνσης. Για το λόγο αυτό, μία UPnP συσκευή θα πρέπει να παρέχει ένα όνομα στο DHCP εξυπηρετητή και να το καταχωρεί σε έναν DNS εξυπηρετητή, ελέγχοντας αν το όνομα αυτό είναι μοναδικό.

5.3.2. Ανίχνευση

Η ανίχνευση (device/service discovery) συσκευών και υπηρεσιών αποτελεί το επόμενο βήμα του UPnP πρωτοκόλλου. Στην ορολογία του UPnP σαν υπηρεσίες ορίζονται οι δυνατότητες διαχείρισης και ελέγχου που παρέχει η συσκευή στο σημείο ελέγχου. Όταν μία συσκευή έχει προστεθεί στο δίκτυο, το UPnP πρωτόκολλο ανίχνευσης επιτρέπει στην συσκευή να διαφημίσει (advertisement) τις υπηρεσίες της στα σημεία ελέγχου του δικτύου. Αντίστοιχα, όταν ένα σημείο ελέγχου εισέρχεται στο δίκτυο, το ίδιο πρωτόκολλο επιτρέπει σε αυτό να ψάξει για ενδιαφέρουσες συσκευές στο δίκτυο. Η πληροφορία που ανταλλάσσεται και στις δύο περιπτώσεις περικλείεται σε ένα μικρό μήνυμα ανίχνευσης, το οποίο περιέχει λίγες αλλά σημαντικές πληροφορίες για τη συσκευή και για τις υπηρεσίες που αυτή μπορεί να προσφέρει, π.χ. τον τύπο της, έναν καθολικά μοναδικό ταυτοποιητή και ένα δείκτη σε πιο λεπτομερή πληροφορία για τη συσκευή και τις υπηρεσίες της.

Όταν μία συσκευή προστίθεται στο δίκτυο, στέλνει έναν αριθμό από μηνύματα πολλαπλών παραληπτών διαφημίζοντας την ίδια την συσκευή, τις ενσωματωμένες μονάδες της, καθώς και τις προσφερόμενες υπηρεσίες. Κάθε ενδιαφερόμενο σημείο ελέγχου μπορεί να λάβει αυτά τα μηνύματα «ακούγοντας» στην πρότυπη διεύθυνση πολλαπλών παραληπτών και να ενημερωθεί για τις προσφερόμενες δυνατότητες. Αντίστοιχα, όταν ένα σημείο ελέγχου εισέρχεται στο δίκτυο, αποστέλλει ένα μήνυμα ανίχνευσης πολλαπλών παραληπτών, ψάχνοντας για ενδιαφέρουσες συσκευές ή υπηρεσίες. Όλες οι συσκευές στο δίκτυο πρέπει να «ακούνε» στην πρότυπη διεύθυνση πολλαπλών παραληπτών για αυτά τα μηνύματα και να ανταποκρίνονται, εφόσον κάποιες από τις ενσωματωμένες μονάδες ή υπηρεσίες τους ταιριάζουν στα κριτήρια του μηνύματος ανίχνευσης.

Επομένως ένα σημείο ελέγχου μπορεί να πληροφορηθεί για μία συσκευή που παρουσιάζει ενδιαφέρον είτε επειδή η τελευταία έστειλε μηνύματα αυτοδιαφήμισης (διαφημίζοντας αυτήν και τις υπηρεσίες της), είτε διότι η συσκευή ανταποκρίθηκε θετικά σε ένα μήνυμα ανίχνευσης του πρώτου. Και στις δύο περιπτώσεις, αν ένα σημείο ελέγχου «ενδιαφέρεται» να μάθει περισσότερα για μία συσκευή και τις υπηρεσίες της, μπορεί, χρησιμοποιώντας την πληροφορία του μηνύματος ανίχνευσης, να στείλει ένα μήνυμα εξέτασης περιγραφής (επόμενο στάδιο).

Όταν μία συσκευή αποσύρεται από το δίκτυο, θα πρέπει να στέλνει πάλι ένα μήνυμα πολλαπλών παραληπτών στο οποίο θα περιέχει την ανάκληση των προηγουμένων ανακοινώσεων, δηλώνοντας έτσι ότι οι ενσωματωμένες μονάδες της και οι υπηρεσίες της δεν θα είναι πλέον διαθέσιμες. Όταν η IP διεύθυνση μίας συσκευής αλλάζει, θα πρέπει να ανακαλέσει προηγούμενες ανακοινώσεις και να διαφημίσει μονάδες και υπηρεσίες χρησιμοποιώντας την καινούρια IP διεύθυνση.

Για συσκευές και σημεία ελέγχου με πολλαπλές δικτυακές διεπαφές, οι UPnP διαφημίσεις και ανιχνεύσεις θα πρέπει να αποστέλλονται σε όλες τις εσωτερικές διεπαφές που είναι ενεργοποιημένες για UPnP δικτύωση. Όλα τα παραπάνω, υλοποιούνται στα πλαίσια του UPnP από το αντίστοιχο πρωτόκολλο ανίχνευσης, γνωστό ως SSDP (Simple Service Discovery Protocol). Η διαφήμιση και ανίχνευση συσκευών και των υπηρεσιών τους παίζει σημαντικότατο ρόλο στη διαλειτουργικότητα μεταξύ συσκευών και σημείων ελέγχου.

5.3.3. Περιγραφή

Μετά την ανίχνευση μιας ενδιαφέρουσας συσκευής από το σημείο ελέγχου, μόνο λίγα στοιχεία είναι γνωστά για τη συσκευή και τις ρυθμιστικές υπηρεσίες που αυτή μπορεί να προσφέρει. Αυτά δεν είναι παρά τα στοιχεία, που περιέχονται στο μήνυμα ανίχνευσης, δηλαδή ο τύπος της UPnP συσκευής ή υπηρεσίας, ο καθολικά μοναδικός ταυτοποιητής της συσκευής και η ενιαία θέση πόρου (Unique Resource Location - URL) - που δείχνει στα XML κείμενα περιγραφής της UPnP συσκευής. Για να είναι σε θέση όμως, το σημείο ελέγχου να μάθει περισσότερα για τη συσκευή και τις δυνατότητες της αλλά και να αλληλεπιδράσει μαζί της, θα πρέπει να ανακτήσει την περιγραφή της συσκευής και των υπηρεσίων της από το μοναδικό URL που παρέχεται από την συσκευή στο μήνυμα ανίχνευσης.

Η περιγραφή της UPnP συσκευής μοιράζεται σε δύο λογικά μέρη: την περιγραφή της συσκευής, που εκτείνεται στην περιγραφή των φυσικών και λογικών περιεχομένων της και την περιγραφή των υπηρεσιών της, που ουσιαστικά αναδεικνύει τις δυνατότητες της συσκευής. Η περιγραφή μίας UPnP συσκευής περιλαμβάνει πληροφορίες σχετικές με τον κατασκευαστή της, όπως είναι το όνομα και τον αριθμό του μοντέλου, το σειριακό αριθμό, το όνομα του κατασκευαστή, ένα URL που να δείχνει στον δικτυακό τόπο του κατασκευαστή κ.α. Η περιγραφή κάθε UPnP υπηρεσίας περιέχει μία λίστα από ενέργειες, στην κλήση των οποίων είναι σε θέση να ανταποκριθεί επιτυχώς η υπηρεσία. Επίσης περιλαμβάνει και μία λίστα από μεταβλητές που μοντελοποιούν την κατάσταση της υπηρεσίας κατά τη διάρκεια της εκτέλεσης. Για κάθε υπηρεσία που περιλαμβάνεται στην συσκευή, η περιγραφή της συσκευής παραθέτει τον τύπο της υπηρεσίας, το όνομά της, ένα URL που να δείχνει στην περιγραφή της υπηρεσίας, ένα άλλο URL για έλεγχο, καθώς και ένα τρίτο URL για ενημέρωση συμβάντος. Η περιγραφή της συσκευής περιέχει επίσης μία περιγραφή όλων των ενσωματωμένων μονάδων της.

Πρέπει να σημειωθεί ότι μία μοναδική φυσική συσκευή μπορεί να περιέχει περισσότερες από μία λογικές συσκευές. Αυτές οι πολλαπλές λογικές συσκευές μπορούν να μοντελοποιηθούν είτε σαν μία μοναδική βασική συσκευή με πολλές ενσωματωμένες συσκευές και υπηρεσίες ή σαν πολλαπλές βασικές συσκευές, πιθανότατα χωρίς ενσωματωμένες συσκευές. Στην πρώτη περίπτωση, θα υπάρχει μία περιγραφή UPnP συσκευής για τη βασική συσκευή, η οποία θα περιέχει και την περιγραφή των ενσωματωμένων συσκευών. Στη δεύτερη περίπτωση, θα υπάρχουν πολλαπλές περιγραφές UPnP συσκευών, μία για κάθε βασική συσκευή.

Η περιγραφή της κάθε UPnP συσκευής υλοποιείται από τον κατασκευαστή της, δίνεται σε μορφή XML κειμένων και βασίζεται σε μία πρότυπη φόρμα UPnP συσκευής, η οποία καθορίζεται από το UPnP Forum. Η φόρμα εξάγεται από την αντίστοιχη γλώσσα της UPnP

φόρμας, που με τη σειρά της προήλθε από τις πρότυπες δομές της XML σύνταξης. Αντίστοιχα, η περιγραφή μιας UPnP υπηρεσίας είναι σε μορφή XML κειμένου, γίνεται από έναν UPnP πωλητή και βασίζεται σε μία πρότυπη φόρμα UPnP υπηρεσίας. που δίνεται και αυτή από το UPnP Forum. Οι πωλητές των UPnP συσκευών μπορούν να διαφοροποιήσουν τα προϊόντα τους επεκτείνοντας τις υπηρεσίες διαχείρισης και συμπεριλαμβάνοντας επιπρόσθετες UPnP υπηρεσίες ή ενσωματώνοντας επιπρόσθετες λειτουργικές μονάδες. Ουσιαστικά, τα κείμενα περιγραφής της συσκευής και των υπηρεσίων της τεκμηριώνουν την υλοποίησή της. Η ανάκτηση της περιγραφής μίας UPnP συσκευής είναι απλή: το σημείο ελέγχου χρησιμοποιώντας το URL που περιέχεται στο μήνυμα ανίχνευσης στέλνει μία HTTP αίτηση (HTTP GET) προς τη συσκευή. Κατόπιν η συσκευή επιστρέφει το αντίστοιχο κείμενο περιγραφής. Η ανάκτηση της περιγραφής μίας UPnP υπηρεσίας είναι μία αντίστοιχη διαδικασία, η οποία ωστόσο χρησιμοποιεί ξεχωριστά URLs για τα κείμενα περιγραφής της κάθε υπηρεσίας.

5.3.4. Έλεγχος

Με δεδομένη τη γνώση για την ύπαρξη συσκευών στο τοπικό δίκτυο αλλά και για τις υπηρεσίες που αυτές δύνανται προσφέρουν, ένα σημείο ελέγχου μπορεί να ζητήσει (έλεγχος συσκευής) από τις υπηρεσίες αυτές να ρυθμίσουν κάποιες παραμέτρους της συσκευής ή να θέσουν σε λειτουργία κάποια εφαρμογή. Όλες οι παραπάνω ενέργειες υλοποιούνται με SOAP [6] κλήσεις απομακρυσμένων διεργασιών (RPC – Remote Procedure Calls). Με αυτό τον τρόπο το σημείο ελέγχου στέλνει την επιθυμητή ενέργεια στην υπηρεσία της συσκευής και, όταν η ενέργεια αυτή ολοκληρωθεί, η υπηρεσία επιστρέφει τα αποτελέσματα ή τα λάθη, αν υπάρχουν. Τα αποτελέσματα της καλούμενης ενέργειας της υπηρεσίας μπορούν να μοντελοποιηθούν με βάση τις αλλαγές στις μεταβλητές που περιγράφουν την κατάσταση της υπηρεσίας κατά την εκτέλεση της. Όταν αυτές οι μεταβλητές κατάστασης αλλάζουν, τα σημεία ελέγχου μπορούν να ενημερώνονται για το συμβάν.

Τα μηνύματα ελέγχου καθώς και οι απαντήσεις κωδικοποιούνται με βάση το UTF-8 [7] και εκφράζονται σε XML μορφή με χρήση του SOAP (Simple Object Access Protocol) πρωτοκόλλου. Στην περίπτωση που πρέπει να αποσταλεί μεγάλος αριθμός δεδομένων σε συνάρτηση με κάποια ενέργεια, προτείνεται από τη UPnP αρχιτεκτονική να μην αποστέλλονται ούτε σαν μέρος ενός SOAP ορίσματος αλλά ούτε και σαν MIME προσάρτηση σε ένα SOAP μήνυμα. Αντίθετα, προτείνεται να χρησιμοποιείται μία αποστολή εκτός ζώνης (out of band).

5.3.5. Ενημέρωση συμβάντων

Εφόσον ένα σημείο ελέγχου έχει ανιχνεύσει μία UPnP συσκευή και έχει λάβει την περιγραφή αυτής και των υπηρεσίων της, έχει όλα τα απαραίτητα στοιχεία για την ενημέρωση συμβάντων (eventing). Όπως προαναφέρθηκε, η περιγραφή κάθε UPnP υπηρεσίας περιέχει μία λίστα από ενέργειες, στην κλήση των οποίων είναι σε θέση να ανταποκριθεί επιτυχώς η υπηρεσία. Επίσης περιλαμβάνει και μία λίστα από μεταβλητές που μοντελοποιούν την κατάσταση της υπηρεσίας κατά τη διάρκεια της εκτέλεσης. Όταν αλλάξει η τιμή μίας ή

περισσότερων εκ των μεταβλητών κατάστασης, η υπηρεσία εκδίδει ενημερωτικά μηνύματα για τις αλλαγές αυτές και κάθε σημείο ελέγχου που ενδιαφέρεται μπορεί να απαιτήσει να δέχεται αυτή την πληροφορία. Στην περίπτωση αυτή το σημείο ελέγχου ορίζεται ως *συνδρομητής*, ενώ σαν *εκδότης* αναφέρεται η υπηρεσία της UPnP συσκευής που προσφέρει την πληροφορία. Για να γίνει συνδρομητής της ενημέρωσης συμβάντος, το σημείο ελέγχου αποστέλλει ένα μήνυμα συνδρομής. Αν γίνει αποδεκτή η συνδρομή, ο εκδότης απαντάει με την διάρκεια της συνδρομής. Για να παραμείνει ενεργή η συνδρομή, ο συνδρομητής θα πρέπει να ανανεώνει την συνδρομή πριν αυτή λήξει. Στην περίπτωση που ο συνδρομητής δεν χρειάζεται ενημέρωση συμβάντων από την υπηρεσία, θα πρέπει να ακυρώσει τη συνδρομή.

Ο εκδότης παρατηρεί τις αλλαγές στις μεταβλητές κατάστασης και τις μεταδίδει με μηνύματα συμβάντος. Τα μηνύματα αυτά περιέχουν τα ονόματα μίας ή περισσοτέρων μεταβλητών κατάστασης και τις τρέχουσες τιμές αυτών, χρησιμοποιώντας και εδώ XML σύνταξη. Ένα ειδικό αρχικό μήνυμα συμβάντος αποστέλλεται μόλις ο συνδρομητής εγγραφεί. Αυτό το μήνυμα περιέχει τα ονόματα και τις τιμές για όλες τις μεταβλητές που παρακολουθούνται και επιτρέπει στο συνδρομητή να αρχικοποιήσει το μοντέλο για την κατάσταση της υπηρεσίας. Τα μηνύματα ενημέρωσης αποστέλλονται σε όλους τους συνδρομητές. Οι συνδρομητές λαμβάνουν ενημέρωση για συμβάντα σε όλες τις ελεγχόμενες μεταβλητές της υπηρεσίας και τα μηνύματα ενημέρωσης αποστέλλονται ανεξάρτητα από την αιτία αλλαγής της μεταβλητής κατάστασης. Όλα τα παραπάνω υλοποιούνται με τη χρήση της γενικής αρχιτεκτονικής ενημέρωσης συμβάντος GENA (Generic Event Notification Architecture) [4].

5.3.6. Παρουσίαση

Ενώ τα παραπάνω στάδια οφείλει να τα υλοποιεί κάθε UPnP συσκευή, το στάδιο της παρουσίασης είναι προαιρετικό και εξυπηρετεί τη διαχείριση και τον έλεγχο από τον ίδιο το χρήστη, μέσω μιας HTML διεπαφής που παρέχεται από τον εξυπηρετητή ιστού της UPnP συσκευής. Ο εν λόγω τρόπος διαχείρισης περιγράφηκε αναλυτικά στο δεύτερο κεφάλαιο και είναι γνωστός με τον όρο «διαχείριση βασισμένη στο HTTP». Εφόσον το σημείο ελέγχου έχει ανιχνεύσει μία UPnP συσκευή και έχει λάβει την περιγραφή αυτής και των υπηρεσίων της, έχει την δυνατότητα να ξεκινήσει την παρουσίαση της συσκευής. Εφόσον η συσκευή το υποστηρίζει, παρέχει στο κείμενο περιγραφής της ένα URL το οποίο δείχνει στην HTML σελίδα παρουσίασης της συσκευής. Η σελίδα αυτή αλλά και το σύνολο των υπολοίπων HTML σελίδων και CGI προγραμμάτων που συνδέονται με αυτή φιλοξενούνται και εκτελούνται στη UPnP συσκευή. Το σημείο ελέγχου χρησιμοποιεί το παραπάνω URL και μέσω του HTTP πρωτοκόλλου την ανακτά από τη συσκευή και τη φορτώνει σε ένα φυλλομετρητή ιστού. Ανάλογα με την παρεχόμενη HTML διεπαφή, η συσκευή μπορεί να επιτρέψει στο χρήστη τον έλεγχό της αλλά και την παρακολούθηση της κατάστασή της. Ο βαθμός ικανοποίησης του χρήστη, όσον αφορά στον έλεγχο και στην παρακολούθηση της συσκευής, εξαρτάται από τις συγκεκριμένες δυνατότητες της σελίδας παρουσίασης και της συσκευής.

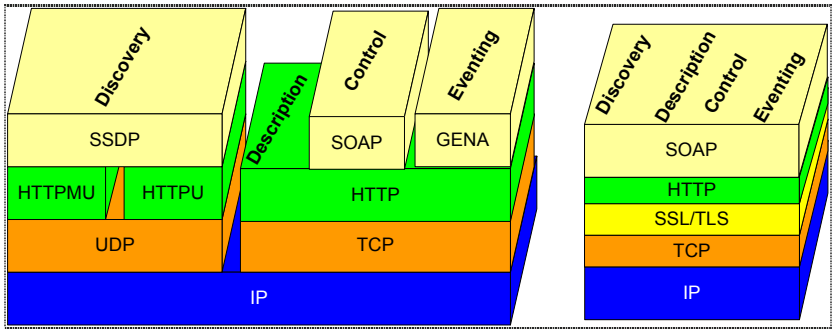
5.4. Οι διαφορές των μηχανισμών τοπικής και απομακρυσμένης διαχείρισης

Στο σημείο αυτό είναι σκόπιμο να διερευνηθούν οι διαφορές και οι ομοιότητες μεταξύ των UPnP και CWMP μηχανισμών, προκειμένου να σχηματιστεί μια καθαρή εικόνα σχετικά με τις απαιτήσεις της γεφύρωσης. Καταρχάς και τα δύο πρωτόκολλα ενεργοποιούνται μόλις ολοκληρωθούν τα στάδια αυτόματης ρύθμισης σε επίπεδο 2 και 3 (Layer 2,3 auto-configuration). Παρόλο που κανένα από τα πρωτόκολλα δεν απαιτεί συγκεκριμένους μηχανισμούς για την παραπάνω ρύθμιση, ωστόσο και τα δύο χρησιμοποιούν τυπικά το DHCP (Dynamic Host Configuration Protocol) πρωτόκολλο για την αυτόματη ρύθμιση της δικτυακής συνδεσιμότητας (Layer 3) των συσκευών τους. Επίσης τόσο το UPnP όσο και το CWMP υποστηρίζουν επιπρόσθετες λειτουργίες από εκείνες του SNMP. Έτσι, οι διαχειρίζουσες οντότητες και των δύο πρωτοκόλλων μπορούν να ανιχνεύουν τις διαχειρίσιμες συσκευές (Ανίχνευση - Discovery), να μαθαίνουν για τις δυνατότητές τους (Περιγραφή - Description), να καλούν εντολές διαχείρισης (Έλεγχος - Control) και να ενημερώνονται (Ενημέρωση συμβάντος - Eventing) για τις αλλαγές κρίσιμων μεταβλητών κατάστασης των συσκευών.

Υπάρχουν αρκετές ομοιότητες αλλά και αρκετές διαφορές, που προκύπτουν κυρίως από τα διαφορετικά περιβάλλοντα, στα οποία απευθύνεται η κάθε μια τεχνολογία. Το UPnP είναι σχεδιασμένο να λειτουργεί σε τοπικά δίκτυα. Εκεί θέματα κίνησης διαχείρισης και φόρτου επεξεργασίας στο σταθμό διαχείρισης (UPnP σημείο ελέγχου) είναι δευτερεύουσας σημασίας, εξαιτίας του σχετικά μεγάλου εύρους ζώνης και του περιορισμένου αριθμού των διαχειριζόμενων συσκευών. Όπως απεικονίζεται στο σχήμα 5-3, το UPnP χρησιμοποιεί διαφορετικούς μηχανισμούς για την ανίχνευση (SSDP – Simple Service Discovery Protocol [4]), την περιγραφή (με μηνύματα HTTP Get για την απόκτηση των αρχείων περιγραφής), τον έλεγχο (SOAP RPCs) και την ενημέρωση συμβάντος (GENA – Generic Event Notification Architecture). Μάλιστα κατά τη διαδικασία της ανίχνευσης των συσκευών εκπέμπεται ένας μεγάλος αριθμός μηνυμάτων εξαιτίας της φύσης του SSDP αλλά και της αναξιοπιστίας του UDP πρωτοκόλλου. Επίσης όσον αφορά στη διαδικασία της ενημέρωση συμβάντος, το UPnP δεν υποστηρίζει την εγγραφή ανά μεταβλητή (per variable subscription) αλλά μόνο την εγγραφή ανά υπηρεσία (per service subscription). Αυτό σημαίνει ότι εάν ένα UPnP σημείου ελέγχου επιθυμεί να ενημερωθεί σχετικά με μια και μόνο μεταβλητή, τότε η αντίστοιχη διαχειριζόμενη συσκευή θα στέλνει μηνύματα ενημέρωσης συμβάντος όχι μόνο όταν η συγκεκριμένη τιμή αλλάζει, αλλά κάθε φορά που μια άλλη μεταβλητή η οποία ανήκει στην ίδια υπηρεσία αλλάζει. Κάτι τέτοιο δημιουργεί αφενός άχρηστη κίνηση στο δίκτυο και αφετέρου άσκοπο φορτίο στο διαχειριστή, ο οποίος αναπόφευκτα επεξεργάζεται μηνύματα που δεν τον αφορούν.

Από την άλλη πλευρά το CWMP είναι σχεδιασμένο να λειτουργεί στα δημόσια δίκτυα, όπου ένας ACS είναι επιφορτισμένος με τη διαχείριση χιλιάδων συνδρομητικών συσκευών, μέσω διαφορετικών τεχνολογιών πρόσβασης και επομένως και διαφορετικών ρυθμών μετάδοσης. Προκειμένου να αποφύγει ενδεχόμενα προβλήματα κλιμακωσιμότητας, το CWMP είναι πολύ

απλούστερο από το UPnP και χρησιμοποιεί ένα μικρό σύνολο από SOAP RPCs (Remote Procedure Calls - κλήσεις απομακρυσμένων διεργασιών) για να υποστηρίξει όλες τις σχετικές λειτουργίες. Το σχήμα 5-3 απεικονίζει αυτή τη λιτότητα όσον αφορά στη χρησιμοποιούμενη στοίβα πρωτοκόλλων. Επιπρόσθετα ο πίνακας 5.1 συλλέγει τις βασικές RPC μεθόδους του CWMP και τις κατηγοριοποιεί σύμφωνα με τις λειτουργίες που επιτελεί η κάθε μια. Αντίθετα με το UPnP που χρησιμοποιεί (broadcast) μηνύματα εκπομπής (SSDP/HTTPMU) για να ανιχνεύσει τις συσκευές και τις υπηρεσίες, το CWMP απαιτεί μόνο κάθε CPE να γνωρίζει την ενιαία θέση πόρου (URL - Uniform Resource Location) που αντιστοιχεί στον ACS διαχειριστή. Το URL αυτό μπορεί να παρέχεται είτε δυναμικά, κατά τη διάρκεια της DHCP διαδικασίας, είτε με τοπική ρύθμιση ή μπορεί ακόμα να είναι και προεγκατεστημένο στο CPE. Η πληροφορία ελέγχου κωδικοποιείται σε SOAP/XML και στα δύο πρωτόκολλα. Η πληροφορία αυτή ενθυλακώνεται σε HTTP πακέτα και μεταφέρεται πάνω από τη στοίβα TCP/IP. Το CWMP υποστηρίζει επίσης ασφαλείς συνδέσεις (SSL/TLS). Μια διαφορά που αξίζει να σημειωθεί είναι ότι η διαχειριζόμενη οντότητα του UPnP περιλαμβάνει ένα HTTP εξυπηρετητή, ενώ η αντίστοιχη του UPnP περιλαμβάνει ένα πελάτη για την ανταλλαγή των SOAP μηνυμάτων και ένα εξυπηρετητή για την εγκατάσταση αμφίδρομων συνδέσεων με τον ACS.



(α) (β)

Σχήμα 5-4. (α) Η στοίβα πρωτοκόλλων του UPnP για κάθε λειτουργία, (β) Η στοίβα πρωτοκόλλων του CWMP είναι κοινή για όλες τις λειτουργίες.

Το CWMP χρησιμοποιεί μόνο δύο τυποποιημένες RPC μεθόδους (*GetParameterValues* /*SetParameterValues*) για τον έλεγχο, εν αντιθέσει με το UPnP, το οποίο για κάθε υπηρεσία προτυποποιεί διαφορετικά σύνολα πράξεων ελέγχου. Για παράδειγμα, ο ACS καλεί τη μέθοδο *SetParameterValues* (με όρισμα το *SubnetMask*) για να θέσει τη μάσκα υποδικτύου (subnet mask) σε μία πύλη. Αντίθετα όταν ένα UPnP σημείο ελέγχου θέτει την ίδια παράμετρο, χρησιμοποιεί την ειδική μέθοδο *SetSubnetMask*. Φαίνεται λοιπόν ότι το CWMP είναι περισσότερο βασισμένο στο μοντέλο διαχείρισης των συσκευών παρά στις ενέργειες διαχείρισης. Αυτή η δεδομενοστραφής (data-oriented) μοντελοποίηση έχει κληρονομηθεί από την ιδέα της βάσης διαχείρισης (MIB) του SNMP. Στη συνέχεια αναλύονται οι απαιτήσεις της προτεινόμενης γέφυρας που πηγάζουν από τις προαναφερθείσες διαφορές μεταξύ των πρωτοκόλλων απομακρυσμένης και τοπικής διαχείρισης.

Πίνακας 5.1.Οι βασικές μέθοδοι του CWMP πρωτοκόλλου.

Λειτουργία	RPC Μέθοδος	Περιγραφή
Περιγραφή	*GetRPCMethods*	Καλείται από ένα CPE ή από τον ACS για να ζητήσει το σύνολο των υποστηριζόμενων μεθόδων από τον ACS ή το CPE αντίστοιχα..
	GetParameterNames	Καλείται από τον ACS για να ζητήσει τις διαχειρίσιμες παραμέτρους εντός του CPE.
Έλεγχος	*SetParameterValues*	Καλείται από τον ACS για να θέσει την τιμή μιας CPE παραμέτρου.
	GetParameterValues	Καλείται από τον ACS για να ζητήσει την τιμή μιας CPE παραμέτρου.
	Reboot	Με αυτή την εντολή ο ACS προστάζει το CPE να επανεκκινήσει.
	Download	Με αυτή την εντολή ο ACS προστάζει το CPE να μεταφορτώσει ένα συγκεκριμένο αρχείο από μια προκαθορισμένη τοποθεσία. Χρησιμοποιείται κυρίως για ανανέωση λογισμικού.
Ανίχνευση και Ενημέρωση συμβάντος	*Inform*	Η μέθοδος αυτή είναι η βάση της ανίχνευσης και της ενημέρωσης συμβάντος. Στέλνεται κάθε φορά που το CPE εκκινεί μια συνεδρία με τον ACS και ενημερώνει για την τρέχουσα κατάσταση του CPE.
	SetParameterAttributes	Χρησιμοποιείται από τον ACS για να δηλώσει τις μεταβλητές κατάστασης του CPE για τις οποίες ο ACS ενδιαφέρεται να ενημερώνεται. Οι αναγγελίες του CPE προς τον ACS είναι υπό μορφή *Inform* μηνυμάτων τα οποία περιλαμβάνουν τις νέες τιμές των τροποποιημένων μεταβλητών.
	GetParameterAttributes	Με αυτή τη μέθοδο ο ACS ανακτά τα χαρακτηριστικά ενημέρωσης και πρόσβασης για κάθε μεταβλητή κατάστασης του CPE.

5.5. Οι απαιτήσεις της γεφύρωσης

Οι κύριες προϋποθέσεις που πρέπει να πληρεί μια τέτοια γέφυρα είναι η διαφάνεια και η κλιμακωσιμότητα. Η διαφάνεια συνίσταται στη δυνατότητα της γέφυρας να επιτρέπει στον ACS να διαχειρίζεται απομακρυσμένα τις UPnP συσκευές και μάλιστα σαν αυτός να ήταν ένα UPnP σημείο ελέγχου μέσα στο δίκτυο του συνδρομητή. Η απαίτηση αυτή μεταφράζεται σε μια πλήρη αντιστοίχιση των UPnP λειτουργιών με εκείνες του CWMP. Πιο συγκεκριμένα, η γέφυρα θα πρέπει να επιτρέπει στον ACS τα ακόλουθα:

- Να ενημερώνεται για την κατάσταση του UPnP δικτύου (Ανίχνευση).
- Να μπορεί να πληροφορείται για όλες τις δυνατότητες των ανιχνευθέντων UPnP συσκευών (Περιγραφή).
- Να καλεί κάθε UPnP πράξη ελέγχου και να λαμβάνει τις αντίστοιχες (SOAP) απαντήσεις χωρίς καμία τροποποίηση από τη γέφυρα (Έλεγχος).
- Να επιτελεί εγγραφές ενημέρωσης συμβάντος και να ενημερώνεται για τις αλλαγές των τιμών των αντιστοίχων μεταβλητών κατάστασης (Eventing).

Μάλιστα για κάθε μια από τις παραπάνω δυνατότητες θα πρέπει να διασφαλίζεται ο χαμηλός όγκος της κίνησης διαχείρισης στο δημόσιο δίκτυο και παράλληλα θα πρέπει να προστατεύεται ο ACS από την επεξεργασία άχρηστων μηνυμάτων διαχείρισης.

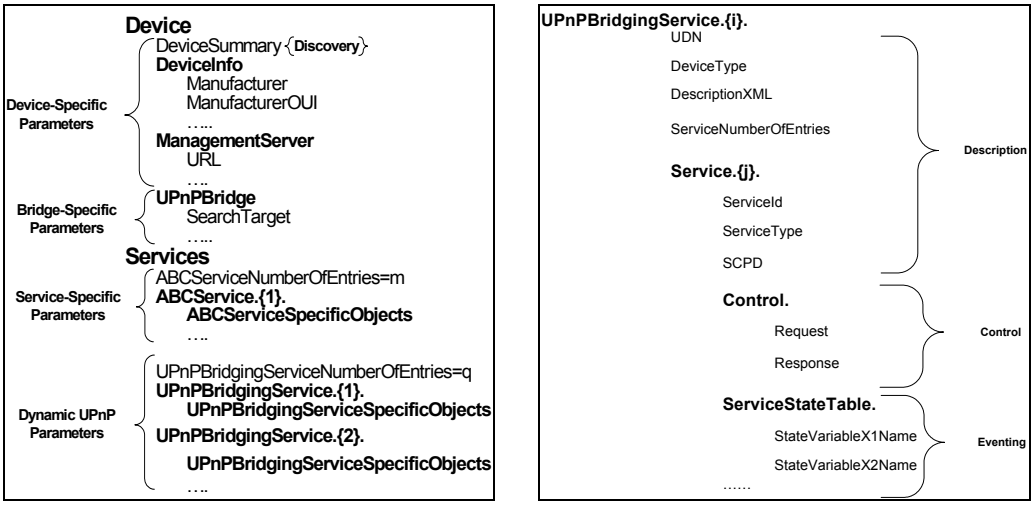
5.6. Η λογική της προτεινόμενης γεφύρωσης

Παρακάτω προσδιορίζονται αναλυτικά οι λειτουργίες της γεφύρωσης και εξηγείται ο τρόπος με τον οποίο η κάθε μια εκπληρώνει τις παραπάνω απαιτήσεις.

5.6.1. Περιγραφή συσκευής

Παρόλο που η διαδικασία της ανίχνευσης προηγείται από εκείνη της περιγραφής, η ανάλυση θα ξεκινήσει από τη δεύτερη, καθώς αυτή είναι η λειτουργία πάνω στην οποία στηρίζεται η βασική ιδέα της προτεινόμενης λύσης. Η κύρια ιδέα της εν λόγω προσέγγισης είναι ότι ο CWMP πελάτης (πράκτορας) πέραν των καθορισμένων παραμέτρων, που σχετίζονται με το μοντέλο διαχείρισης της συσκευής (CPE) εκθέτει στον ACS και μία άλλη ομάδα παραμέτρων, οι οποίες σχετίζονται με τις ανιχνευθείσες UPnP συσκευές. Μέσω αυτών ο ACS μπορεί να επικοινωνήσει με τις UPnP συσκευές. Το σχήμα 5-4 (α) απεικονίζει τη θέση της πληροφορίας αυτής (UPnP Dynamic Parameters) εντός της συνολικής CWMP περιγραφικής δομής. Η απεικονιζόμενη δομή είναι σύμφωνη με τους κανόνες μοντελοποίησης του DSL forum. Το κείμενο αυτό [8] είναι η πηγή κατασκευής όλων των υπαρχόντων βάσεων διαχείρισης (MIBs) καθώς επίσης και των μεθόδων εκείνων για τον καθορισμό νέων MIBs που αντιστοιχούν σε άλλα είδη συσκευών. Σε αυτού του είδους τις MIBs καθορίζονται δύο αντικείμενα: τα αντικείμενα συσκευής (Device Objects) και τα αντικείμενα υπηρεσιών (Services Objects). Το αντικείμενο συσκευής (Device) περιλαμβάνει όλα τα υπό-αντικείμενα και τις παραμέτρους εκείνες που σχετίζονται με τη μοντελοποιημένη συσκευή. Το αντικείμενο υπηρεσιών περιλαμβάνει πληροφορία σχετική με τις μοντελοποιημένες υπηρεσίες που υποστηρίζονται από τη συσκευή. Για παράδειγμα, όλη η σχετική πληροφορία με τη δικτυακή πύλη (π.χ. ADSL παράμετροι, Ethernet και 802.11 παράμετροι, παράμετροι προώθησης των IP πακέτων) βρίσκεται κάτω από το αντικείμενο συσκευής. Ωστόσο μια πύλη που ενσωματώνει και την υπηρεσία VoIP (προτυποποιημένη σε ξεχωριστό κείμενο [9]) θα εκθέτει τις VoIP παραμέτρους (π.χ. SIP/H.323, κωδικοποιητές φωνής) κάτω από το αντικείμενο των υπηρεσιών. Έτσι το αντικείμενο *ABCService.{1}.* αντιπροσωπεύει προτυποποιημένες υπηρεσίες σαν και αυτή του VoIP.

Καταρχάς ο ACS θα πρέπει να μπορεί να ρυθμίσει κάποιες στατικές παραμέτρους, που σχετίζονται με την ίδια τη γέφυρα. Για αυτό το λόγο ένα αντικείμενο (*UPnPBridge*) θα πρέπει να βρίσκεται κάτω από το αντικείμενο της συσκευής. Επίσης η δυναμικά παραγόμενη UPnP πληροφορία θα πρέπει να βρίσκεται κάτω από το αντικείμενο υπηρεσιών καθώς αυτή σχετίζεται με την υπηρεσία γεφύρωσης που παρέχεται από τη συσκευή (CPE). Για κάθε UPnP συσκευή δημιουργείται δυναμικά ένα νέο αντικείμενο (UPnPBridgingService.{i}.) το οποίο περιλαμβάνει όλα τα υπό-αντικείμενα και τις παραμέτρους εκείνες που σχετίζονται με τη συσκευή και τις υπηρεσίες της. Η πληροφορία αυτή συνιστά τη διεπαφή επικοινωνίας του ACS με τις UPnP συσκευές και αναλύεται στο σχήμα 5-4 (β).



(α) (β)

Σχήμα 5-5. Τα αντικείμενα και οι παράμετροι που εκθέτει μια συσκευή-γέφυρα (CPE) στον ACS. (α)η συνολική δομή της εκτιθέμενης πληροφορίας, (β) η δυναμικά παραγόμενη ιεραρχία των UPnP παραμέτρων.

Κάτω από το *UPnPBridgingService.{i}.* αντικείμενο υπάρχει ένα σύνολο παραμέτρων (*UDN, DeviceType* κ.α.) που σχετίζονται με την ανιχνευθείσα UPnP συσκευή. Υπάρχει επίσης ένα σύνολο υπό-αντικειμένων (*Service.{j}.*), το οποίο ανταποκρίνεται στις υπηρεσίες, που είναι διαθέσιμες από τη UPnP συσκευή. Οι *DescriptionXML* και SCPD (Service Control Protocol Description – Περιγραφή Πρωτοκόλλου Ελέγχου Υπηρεσίας) παράμετροι μεταφέρουν ολόκληρα τα XML κείμενα περιγραφής της συσκευής και των υπηρεσιών αντίστοιχα. Για κάθε UPnP συσκευή υπάρχει πάντα ένα μοναδικό κείμενο που την περιγράφει αλλά μπορούν να υπάρχουν πολλά SCPD κείμενα, ένα για κάθε υπηρεσία. Το CWMP χρησιμοποιεί μεγάλες σειρές χαρακτήρων (strings) για να μεταφέρει ακόμα και ολόκληρα αρχεία ρύθμισης. Επομένως δύναται να υποστηρίξει τη μεταφορά των αρχείων περιγραφής μέσω απλών παραμέτρων τύπου string αφού το μέγεθος των αρχείων αυτών δεν ξεπερνά κάποιες δεκάδες Kbytes. Εναλλακτικά η προαιρετική μέθοδος *upload* του CWMP θα μπορούσε να χρησιμοποιηθεί για να μεταφέρει μεγαλύτερα αρχεία μέσω του HTTP πρωτοκόλλου. Τα κείμενα περιγραφής λαμβάνονται από τη γέφυρα αμέσως μόλις ολοκληρωθεί το στάδιο της UPnP ανίχνευσης. Η γέφυρα δίνει εντολή στο UPnP σημείο ελέγχου του CPE να μεταφορτώσει από τις ανιχνευθείσες συσκευές όλα τα αρχεία περιγραφής. Στη συνέχεια παρασκευάζει σε πραγματικό χρόνο την ιεραρχία του σχήματος 5-4 και συμπληρώνει τις παραμέτρους του δέντρου με βάση τη ληφθείσα UPnP πληροφορία. Στη συνέχεια ο ACS μπορεί να ανακτήσει τις τιμές των παραμέτρων και έτσι να πληροφορηθεί πλήρως για τις διαχειρίσιμες UPnP συσκευές. Με αυτό τον τρόπο επιτυγχάνεται η διαφάνεια σε επίπεδο περιγραφής. Μάλιστα όπως θα δειχθεί στη συνέχεια η προτεινόμενη ιεραρχία επιτρέπει στον ACS να προχωρήσει στη διαδικασία ελέγχου και ενημέρωσης συμβάντος χωρίς να είναι απαραίτητη η μεταφόρτωση ολόκληρων των κειμένων περιγραφής. Αυτό μπορεί να μειώσει σημαντικά τον όγκο της κίνησης που παράγεται στο δημόσιο δίκτυο κατά τη διαδικασία της περιγραφής.

5.6.2. Ανίχνευση συσκευής

Κάθε φορά που η κατάσταση του UPnP δικτύου αλλάζει, το UPnP σημείο ελέγχου του CPE ενημερώνει τη γέφυρα. Κατόπιν η γέφυρα δίνει εντολή στον CWMP πελάτη του CPE να ξεκινήσει μια συνεδρία με τον ACS προκειμένου να του δηλώσει αυτή την αλλαγή. Η *Inform* μέθοδος χρησιμοποιείται γενικά από τον CWMP πελάτη για να ενημερώνει ασύγχρονα τον ACS. Έτσι η παρούσα προσέγγιση χρησιμοποιεί αυτή τη μέθοδο με όρισμα την *DeviceSummary* παράμετρο ώστε να ενημερώνεται ο ACS για τη νέα κατάσταση στο UPnP δίκτυο του συνδρομητή. Η επιλογή της *DeviceSummary* παραμέτρου για τη μεταφορά της UPnP πληροφορίας έγινε στηριζόμενη στο γεγονός ότι η εν λόγω παράμετρος είναι ήδη τυποποιημένη και χρησιμοποιείται από τον ACS προκειμένου να ενημερώνεται για τη φύση των συσκευών (CPE) καθώς και για να ελέγχει τη συμβατότητά του με εκείνες. Έτσι πέραν της συνοπτικής περιγραφής του ίδιου του CPE, τώρα η *DeviceSummary* παράμετρος περιλαμβάνει ακόμα και την πληροφορία που αποκτήθηκε από το σημείο ελέγχου κατά τη διάρκεια της UPnP ανίχνευσης (τύπος συσκευής / υπηρεσίας, αναγνωριστικοί κωδικοί κ.α.). Για να διασφαλιστεί η διαφάνεια κατά τη λειτουργία της ανίχνευσης, η γέφυρα εισάγει στην *DeviceSummary* παράμετρο ολόκληρη την πληροφορία της UPnP ανίχνευσης και όχι μόνο ένα μέρος αυτής (Βλέπε Παράρτημα 2).

Για να αποφευχθεί η άσκοπη χρήση του εύρου ζώνης και οι υπερκεφαλισμοί επεξεργασίας στον ACS διαχειριστή προτείνεται να γίνει χρήση *παθητικής ενημέρωσης* (*Passive Notification*) που προβλέπεται από το CWMP πρωτόκολλο. Με αυτό τρόπο το CPE μπορεί να μην ενημερώνει τον ACS κάθε φορά που αλλάζει η κατάσταση του UPnP δικτύου, αλλά την επόμενη φορά που θα ξεκινήσει μια συνεδρία μεταξύ του ACS και του CPE. Διαφορετικά, εάν ο ACS επιθυμεί να είναι ενήμερος για το UPnP δίκτυο σε πραγματικό χρόνο, τότε η *DeviceSummary* παράμετρος σημειώνεται από τον ACS σαν παράμετρος «ενεργητικής ενημέρωσης» (Active Notification). Στην περίπτωση αυτή για κάθε αλλαγή του UPnP δικτύου το CPE εκκινεί μια νέα συνεδρία με τον ACS κατά τη διάρκεια της οποίας τον ενημερώνει για την αλλαγή. Μάλιστα για να μειωθεί περαιτέρω ο όγκος της δικτυακής κίνησης στο δημόσιο δίκτυο και το φορτίο επεξεργασίας στον ACS, προτείνεται η δυνατότητα να μπορεί ο ACS να επιλέγει τον τύπο των UPnP συσκευών για τις οποίες επιθυμεί να ενημερώνεται. Ο ACS μπορεί να δηλώσει κάτι τέτοιο θέτοντας την *SearchTarget* παράμετρο, που βρίσκεται κάτω από το *UPnPBridge* αντικείμενο (σχήμα 5-4). Βασισμένη στη τιμή αυτής της παραμέτρου η γέφυρα φιλτράρει τις UPnP συσκευές που παρουσιάζονται στον ACS. Για παράδειγμα, εάν ο ACS ενδιαφέρεται να ανιχνεύει και να διαχειρίζεται μόνο συσκευές τύπου εξυπηρετητή πολυμέσων, τότε η γέφυρα εκθέτει μόνο αυτού του είδους τις συσκευές, ενώ το υπόλοιπο UPnP δίκτυο δεν εμφανίζεται καθόλου στον ACS.

5.6.3. Έλεγχος συσκευής

Η λειτουργία αυτή είναι η κύρια δραστηριότητα της διαχείρισης. Όπως φαίνεται και στον πίνακα 5.1 το CWMP πρωτόκολλο προδιαγράφει μόνο τις *GetParameterValues* και *SetParameterValues* μεθόδους για την παρακολούθηση και τη ρύθμιση των συσκευών. Προκειμένου η προτεινόμενη λύση να είναι συμβατή με το CWMP πρότυπο, η UPnP

πληροφορία ελέγχου θα πρέπει να μεταφέρεται μόνο μέσω αυτών των δύο μεθόδων. Για να επιτευχθεί η διαφάνεια σε επίπεδο ελέγχου προτείνεται ο ACS να παρασκευάζει ο ίδιος και να στέλνει τις UPnP SOAP αιτήσεις στη γέφυρα. Κατόπιν, η γέφυρα δίνει εντολή στο UPnP σημείο ελέγχου να προωθήσει τις αιτήσεις στις σωστές UPnP συσκευές. Στην αντίθετη κατεύθυνση το UPnP σημείο ελέγχου λαμβάνει και μεταφέρει στη γέφυρα τις SOAP απαντήσεις των UPnP συσκευών. Κατόπιν, η γέφυρα κάνει διαθέσιμες στον ACS τις SOAP απαντήσεις μέσω του CWMP πελάτη. Ο ACS στη συνέχεια μπορεί να συλλέξει τις απαντήσεις είτε για να επαληθεύσει ότι οι εντολές διαχείρισης εκτελέστηκαν σωστά εντός των UPnP συσκευών, είτε για να επεξεργαστεί πιθανούς κωδικούς σφάλματος, σε περίπτωση αποτυχίας.

Σύμφωνα με τη UPnP αρχιτεκτονική [4] μια συσκευή παρέχει υπηρεσίες, οι οποίες μπορούν να αξιοποιηθούν από τα UPnP σημεία ελέγχου μέσω μιας προγραμματιστικής διεπαφής (δηλαδή ένα σύνολο UPnP μεθόδων). Για κάθε υπηρεσία η διεπαφή αυτή δίνεται στα αντίστοιχα κείμενα περιγραφής (Service Control Protocol Description - SCPDs). Το σχήμα 5-4 δείχνει ότι η γέφυρα παρουσιάζει στον ACS όλη τη διαθέσιμη πληροφορία για κάθε υπηρεσία (το SCPD κείμενο, το ServiceType κ.α.) μαζί με ένα αντικείμενο που ονομάζεται *Control*. Το αντικείμενο αυτό περιλαμβάνει μόνο δύο παραμέτρους τύπου σειράς χαρακτήρων (string): την *Request* και την *Response* παράμετρο. Η πρώτη φέρει το UPnP SOAP φάκελο της αίτησης, τον οποίο ο ACS στέλνει στη γέφυρα μέσω της *SetParameterValues* μεθόδου. Η δεύτερη φέρει το UPnP SOAP φάκελο της απάντησης, τον οποίο ο ACS ανακτά από τη γέφυρα μέσω της *GetParameterValues* μεθόδου. Με αυτό τον τρόπο η γέφυρα αποθηκεύει και προωθεί τους UPnP SOAP φακέλους από και προς τον ACS. Το μόνο που χρειάζεται η γέφυρα είναι να δίνει τις κατάλληλες εντολές στο UPnP σημείο ελέγχου, ώστε εκείνο να μπορεί να προωθήσει τις αιτήσεις στις σωστές UPnP συσκευές. Η προτεινόμενη ιεραρχία του σχήματος 5-4 εξυπηρετεί το μοναδικό προσδιορισμό της διαχειριζόμενης συσκευής (*UPnPBridgingService.{i}.*) και της διαχειριζόμενης υπηρεσίας εντός της συσκευής (*UPnPBridgingService.{i}.Service.{j}*). Το τελευταίο μονοπάτι είναι η μόνη πληροφορία που χρειάζεται να γνωρίζει ο ACS πριν θέσει (set *Request*) ή πριν συλλέξει (get *Response*) του UPnP SOAP φακέλους. Η πληροφορία αυτή (δηλαδή ποιο μονοπάτι αντιστοιχεί σε ποια συσκευή / υπηρεσία) είναι ήδη διαθέσιμη στον ACS από το στάδιο της ανίχνευσης.

5.6.4. Ενημέρωση συμβάντος

Με τον ίδιο ακριβώς τρόπο η γέφυρα εκθέτει στον ACS τις UPnP παραμέτρους κάθε υπηρεσίας που μπορούν να παράγουν μηνύματα ενημέρωσης. Οι παράμετροι αυτοί είναι ομαδοποιημένες ανά υπηρεσία (Service.{j}) και εμφανίζονται κάτω από το *ServiceStateTable* αντικείμενο. Στην πράξη αναπαριστούν τις UPnP μεταβλητές κατάστασης, που παράγουν ενημέρωση συμβάντος και αναφέρονται στο κείμενο περιγραφής της κάθε υπηρεσίας. Έτσι η γέφυρα αντί του ACS επεξεργάζεται τα κείμενα περιγραφής και παρουσιάζει σε αυτόν μόνο εκείνες τις μεταβλητές που μπορούν να παράγουν ενημέρωση κατά την τροποποίησή τους. Με αυτό τον τρόπο ο ACS απαλλάσσεται από την επεξεργασία των κειμένων περιγραφής.

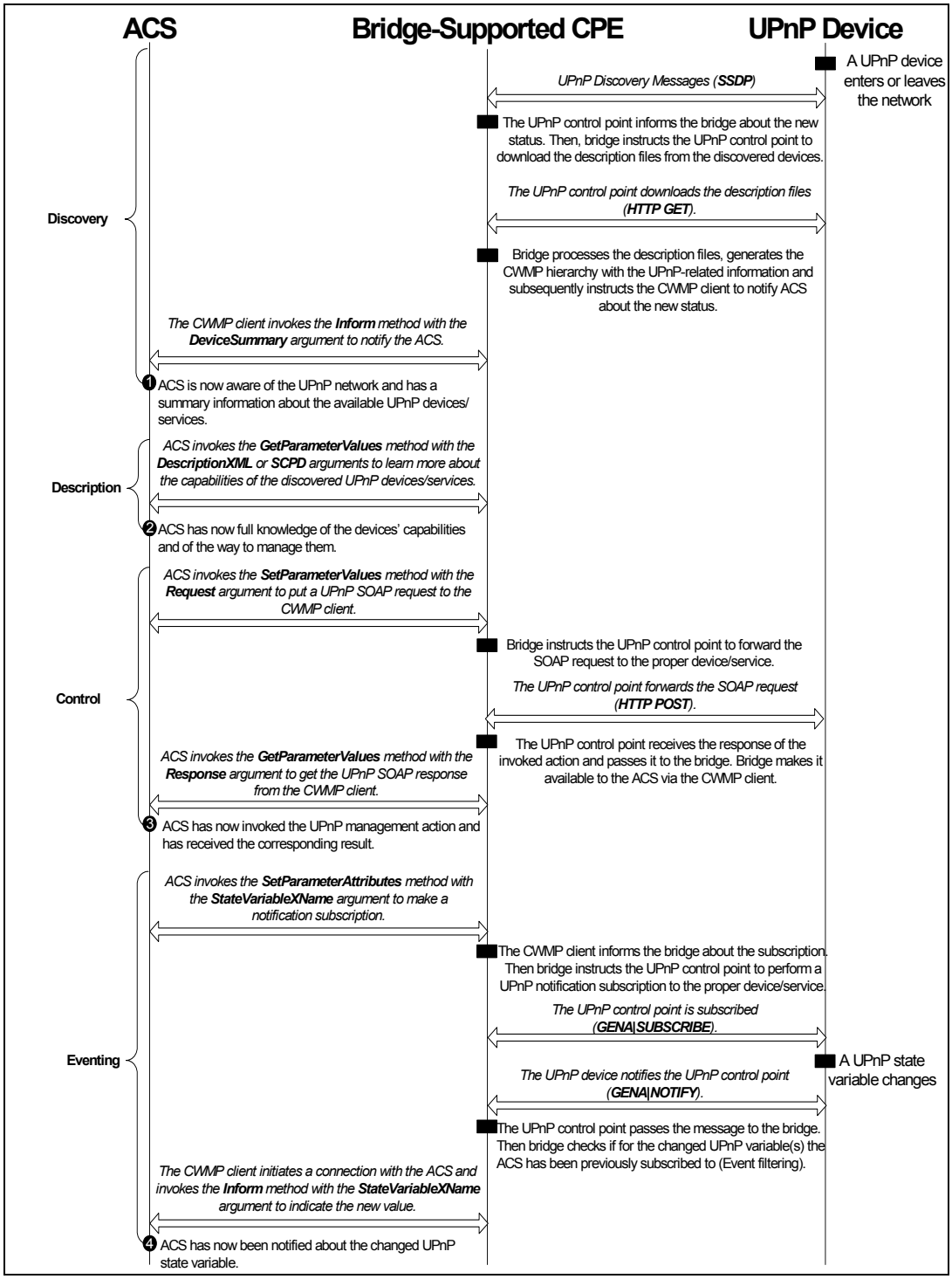
Γενικά το CWMP χρησιμοποιεί τη μέθοδο *SetParameterAttributes* για να δηλώσει τις παραμέτρους για τις οποίες επιθυμεί να ενημερώνεται από τη συσκευή-γέφυρα (CPE). Όπως συμβαίνει και με τις υπόλοιπες CWMP παραμέτρους, ο ACS μπορεί να καλέσει αυτή τη μέθοδο για να μαρκάρει τις μεταβλητές εκείνες που βρίσκονται κάτω από το αντικείμενο *ServiceStateTable* και με αυτό τον τρόπο να υποδείξει τις UPnP μεταβλητές για τις οποίες επιθυμεί να ενημερώνεται.

Όταν ο ACS μαρκάρει για ενημέρωση μια συγκεκριμένη UPnP παράμετρο, η γέφυρα δίνει εντολή στο UPnP σημείο ελέγχου να πραγματοποιήσει μια εγγραφή ενημέρωσης στην αντίστοιχη υπηρεσία. Μόλις ολοκληρωθεί η εγγραφή και αλλάξει η τιμή μίας UPnP μεταβλητής, η UPnP συσκευή στέλνει μηνύματα ενημέρωσης (GENA) προς το UPnP σημείο ελέγχου. Τα μηνύματα αυτά μεταφέρονται στη γέφυρα για περαιτέρω επεξεργασία. Η γέφυρα στη συνέχεια στέλνει ένα CWMP μήνυμα ενημέρωσης στον ACS μόνο στην περίπτωση που η σχετική λαμβανόμενη UPnP ενημέρωση περιλαμβάνει μια μεταβλητή για την οποία ο ACS είχε εκδηλώσει το ενδιαφέρον του προηγουμένως (δηλαδή είχε κάνει μια CWMP εγγραφή ενημέρωσης). Με αυτό τον τρόπο η γέφυρα διευθετεί το πρόβλημα της «εγγραφής ανά υπηρεσία» του UPnP. Το φιλτράρισμα αυτό μπορεί να μειώσει δραστικά την κίνηση ενημέρωσης συμβάντων στο δημόσιο δίκτυο και να προστατεύσει τον ACS από την επεξεργασία πολλών άχρηστων μηνυμάτων.

Ωστόσο, σε κάποιες UPnP υπηρεσίες το μοντέλο συμβάντων ορίζει μια και μόνη μεταβλητή κατάστασης η οποία εξυπηρετεί την ενημέρωση για περισσότερα τους ενός συμβάντα. Για παράδειγμα, η υπηρεσία *RenderingControl* που παρέχεται από μια UPnP συσκευή απόδοσης πολυμέσων (UPnP Media Renderer) χρησιμοποιεί τη μεταβλητή κατάστασης *LastChange*. Η μεταβλητή αυτή αλλάζει οποτεδήποτε πατιέται ένα διαφορετικό κουμπί (π.χ. “STOP”, “PLAY”, “FF” και “RW”). Στην περίπτωση που ο ACS επιθυμεί να ενημερώνεται για ένα και μόνο συμβάν (π.χ. “STOP”) εγγράφεται για την *LastChange* στην υπηρεσία ενημέρωσης του CPE. Τότε όμως λαμβάνει ενημερώσεις κάθε φορά που αλλάζει η μεταβλητή αυτή και όχι μόνο όταν αλλάζει εξαιτίας του συμβάντος που ενδιαφέρει (δηλ. του “STOP”). Κάτι τέτοιο ακυρώνει την προτεινόμενη λύση και μπορεί να δημιουργήσει σημαντικούς υπερκεφαλισμούς στην παραγόμενη δικτυακή κίνηση και στην επεξεργασία του ACS. Προκειμένου λοιπόν να διευθετηθεί ακόμα και αυτή η περίπτωση (η οποία δεν είναι καθόλου σπάνια) προτείνεται ο ACS, πέραν της εγγραφής να δηλώνει παράλληλα (μέσω της *SetParameterValues* μεθόδου) ένα σχέδιο φιλτραρίσματος (filter pattern) για την κάθε μεταβλητή. Έτσι, στο παρόν παράδειγμα η γέφυρα ενημερώνει τον ACS μόνο στην περίπτωση που η τιμή της *LastChange* μεταβλητής (εντός του ληφθέντος UPnP ενημερωτικού μηνύματος) ταιριάζει με το σχέδιο που έχει δηλωθεί προηγουμένως από τον ACS (δηλ. περιλαμβάνει τη λέξη “STOP”).

5.6.5. Οι κύριες διαδράσεις

Το σχήμα 5-5 απεικονίζει τις κύριες προτεινόμενες λειτουργίες της γέφυρας καθώς και τη διάδραση μεταξύ του ACS, της γέφυρας - συσκευής και του UPnP συνδρομητικού εξοπλισμού.

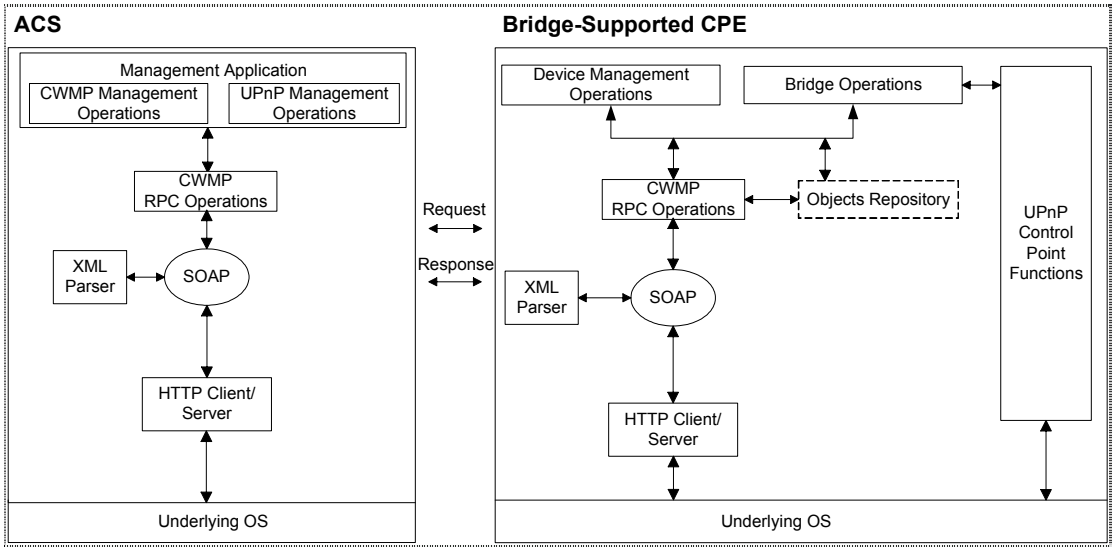


Σχήμα 5-6. Οι κύριες λειτουργίες της γέφυρας και οι αντίστοιχες διαδράσεις των μηχανισμών τοπικής (UPnP) και απομακρυσμένης (CWMP) διαχείρισης.

5.7. Η προτεινόμενη αρχιτεκτονική

Προκειμένου να αποτιμηθεί συνολικά η παραπάνω προτεινόμενη προσέγγιση σχεδιάστηκε και αναπτύχθηκε ένα ολοκληρωμένο σύστημα απομακρυσμένης διαχείρισης του UPnP συνδρομητικού εξοπλισμού. Το σύστημα αποτελείται από τον ACS διαχειριστή και από τη

συσκευή-γέφυρα η οποία περιλαμβάνει τον CWMP πελάτη, την προτεινόμενη γέφυρα και ένα UPnP σημείο ελέγχου.



Σχήμα 5-7. Η αρχιτεκτονική του πειραματικού CWMP-UPnP συστήματος διαχείρισης.

Ο HTTP πελάτης του CPE εκκινεί τη συνεδρία διαχείρισης με τον HTTP εξυπηρετητή του ACS. Στη συνέχεια ο ACS στέλνει στο CPE τις εντολές διαχείρισης με τη μορφή SOAP RPC μηνυμάτων. Ο πελάτης μεταφέρει τα μηνύματα αυτά στη *SOAP* μονάδα, η οποία χρησιμοποιεί τον XML γραμματικό αναλυτή για να αποκωδικοποιήσει τις αιτήσεις. Στη συνέχεια τα αποκωδικοποιημένα μηνύματα εισέρχονται στη μονάδα επεξεργασίας απομακρυσμένων *CWMP* μεθόδων (CWMP RPC operations), η οποία ελέγχει τη συμμόρφωση της καλούμενης μεθόδου και των παραμέτρων της με το CWMP πρωτόκολλο. Αφού ολοκληρωθεί επιτυχώς η παραπάνω διαδικασία, ο έλεγχος περνά είτε στη μονάδα διαχείρισης της συσκευής (*Device Management Operations*), είτε στη μονάδα της γέφυρας (Bridge Operations). Αυτό εξαρτάται από τα ορίσματα της CWMP μεθόδου διαχείρισης του ACS. Με αυτό τον τρόπο η υλοποίηση του CWMP πελάτη υποστηρίζει ταυτόχρονα τη διαχείριση της ίδιας της συσκευής (CPE) που φιλοξενεί το CWMP πελάτη αλλά και των UPnP συσκευών. Η καρδιά του συστήματος είναι η μονάδα αποθήκευσης αντικειμένων (Objects Repository), η οποία περιλαμβάνει ολόκληρη τη βάση διαχείρισης (MIB) του CPE, της γέφυρας και των διαχειριζόμενων UPnP συσκευών. Σε ένα τυπικό CWMP πελάτη οι λειτουργίες ενορχήστρωσης ανανεώνουν μόνο τις τιμές των παραμέτρων. Με την προσθήκη των λειτουργιών της γέφυρας ανανεώνεται επιπλέον και η δομή της πληροφορίας διαχείρισης. Έτσι λοιπόν η γέφυρα προσθέτει και αφαιρεί UPnP αντικείμενα και παραμέτρους (σχήμα 5-4) εντός της MIB. Μάλιστα εισάγει σε αυτά τιμές σχετικές με την αποκτηθείσα UPnP πληροφορία. Τέτοιου είδους αλλαγές στη MIB θα συμβαίνουν κάθε φορά που μια UPnP συσκευή εισάγεται ή εξάγεται από το δίκτυο του συνδρομητή.

Η αρχιτεκτονική του ACS είναι παρόμοια με εκείνη του CPE. Πέραν των λειτουργιών διαχείρισης των αμιγώς CWMP συσκευών, ο ACS ενσωματώνει και μια ακόμη μονάδα για την επεξεργασία της UPnP πληροφορίας. Από τη στιγμή που οι UPnP συσκευές είναι διάφανες στον ACS η UPnP εφαρμογή διαχείρισης είναι πολύ κοντά με εκείνη ενός

κανονικού UPnP σημείου ελέγχου. Τόσο ο ACS όσο και το CPE υποστηρίζουν μονάδες HTTP εξυπηρετητή / πελάτη για την αμφίδρομη εγκατάσταση των συνεδριών διαχείρισης. Για την υλοποίηση των UPnP λειτουργιών χρησιμοποιήθηκε η Java έκδοση του πακέτου ανάπτυξης Cyberlink [9]. Οι CWMP λειτουργίες αλλά και οι λειτουργίες της γέφυρας αναπτύχθηκαν από την αρχή, ενώ οι XML πράξεις υλοποιήθηκαν με τη χρήση των Java προγραμματιστικών διεπαφών Xerces [10].

5.8. Περιγραφή του πειράματος και συνολική αποτίμηση

Προκειμένου να αποτιμηθούν οι δυνατότητες της προτεινόμενης γέφυρας όσον αφορά στη διαφάνεια και στο φιλτράρισμα της κίνησης διαχείρισης διερευνάται ένα πλήρες σενάριο. Το σενάριο παρουσιάζεται στο σχήμα 5-8 και αποτελείται από πέντε κύρια βήματα (STEP1-5). Σύμφωνα με αυτό, ο συνδρομητής, πέραν της δικτυακής πύλης (η οποία ενσωματώνει τη λειτουργία της προτεινόμενης γέφυρας) κατέχει μια UPnP συσκευή αναπαραγωγής πολυμεσικού περιεχομένου (*UPnP Media Renderer*) και έναν UPnP εξυπηρετητή πολυμεσικού περιεχομένου (*UPnP Media Server*). Μάλιστα ο συνδρομητής έχει προηγουμένως εγγραφεί σε μια υπηρεσία, που του προσφέρει διαρκώς ανανεωμένο περιεχόμενο (π.χ. νέα μουσικά video clips ή ειδήσεις) κάθε φορά που εκείνος ανοίγει τις συσκευές του.

Για να επιτευχθεί η τοπολογία του σχήματος 5-8 η δικτυακή πύλη (Residential Gateway / UPnP-CWMP bridge) προσομοιώνεται με ένα προσωπικό υπολογιστή Pentium III (1GHz – 512MB RAM) ο οποίος είναι εξοπλισμένος με δύο Ethernet κάρτες δικτύου. Το σύστημα αυτό εκτελεί την εφαρμογή της πύλης – γέφυρας και είναι συνδεδεμένο απευθείας με δύο ακόμα όμοια μηχανήματα (σε κάθε κάρτα δικτύου του συνδέεται ένα ακόμα PC). Το πρώτο εκ των δύο PCs προσομοιώνει τον ACS διαχειριστή, που βρίσκεται στο δημόσιο δίκτυο δεδομένων. Το δεύτερο PC προσομοιώνει τις UPnP συσκευές του συνδρομητή εκτελώντας τις εφαρμογές του *UPnP Media Server* και του *UPnP Media Renderer* της Intel [11].

Μόλις ο συνδρομητής εκκινήσει τις UPnP συσκευές του, η γέφυρα ενημερώνει τον ACS σταθμό διαχείρισης του παρόχου για τη νέα κατάσταση στο τοπικό δίκτυο (Discovery). Κατά τη διαδικασία αυτή (STEP1) η πύλη-γέφυρα αρχικά «αντιλαμβάνεται» την παρουσία των UPnP συσκευών, μέσω των SSDP μηνυμάτων αυτοδιαφήμισης και στη συνέχεια εκκινεί μια νέα συνεδρία διαχείρισης με τον ACS προκειμένου να τον ενημερώσει για τις συσκευές, που εμφανίστηκαν στο τοπικό δίκτυο.

Κατόπιν ο ACS (ή και ο ίδιος ο πάροχος) μεταφορτώνει το νέο περιεχόμενο σε ένα μοιρασμένο κατάλογο του *UPnP Media Server* (STEP2). Με αυτό τον τρόπο ο συνδρομητής έχει τη δυνατότητα να το αναπαραγάγει όσες φορές επιθυμεί, χωρίς να χρειάζεται κάθε φορά να το μεταφορτώνει από το δημόσιο δίκτυο και να ξοδεύει έτσι άσκοπα το διαθέσιμο εύρος ζώνης του.

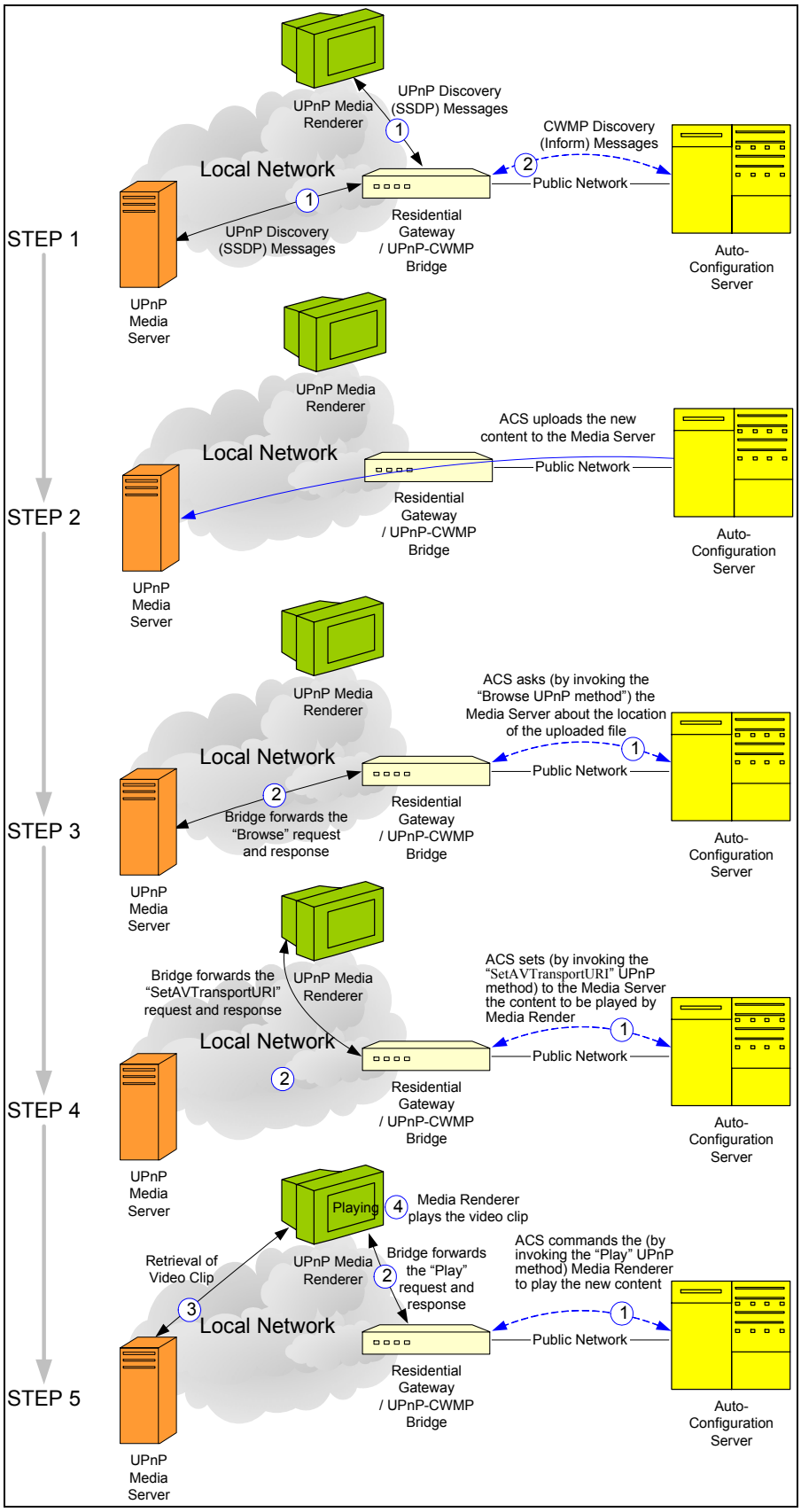
Στη συνέχεια ο ACS «εκκινεί» απομακρυσμένα τη Video ροή από τον εξυπηρετητή (*UPnP Media Server*) στη συσκευή αναπαραγωγής (*UPnP Media Renderer*), προκειμένου ο συνδρομητής να παρακολουθήσει αυτόματα το νέο περιεχόμενο. Η διαδικασία αυτή γίνεται

σύμφωνα με την UPnP AV αρχιτεκτονική [12] η οποία ορίζει τρία υποχρεωτικά επιμέρους στάδια. Καθένα από αυτά περιλαμβάνει την απομακρυσμένη κλήση μιας διαφορετικής UPnP μεθόδου είτε στον εξυπηρετητή, είτε στη συσκευή αναπαραγωγής (Έλεγχος).

Αρχικά (STEP3) ο ACS διαχειριστής αποκτά, μέσω της γέφυρας, από τον UPnP Media Server το αναγνωριστικό ενιαίου πόρου (Uniform Resource Identifier) του αρχείου, που μεταφορτώθηκε στον εξυπηρετητή κατά τη διάρκεια του προηγούμενου σταδίου (STEP2). Αυτό πραγματοποιείται μέσω μιας UPnP υπηρεσίας (Content Directory), που παρέχει ο εξυπηρετητής για τη διαχείριση του περιεχομένου του. Πιο συγκεκριμένα, από την παραπάνω υπηρεσία χρησιμοποιήθηκε η UPnP μέθοδος *Browse*, κατά την οποία ένα UPnP σημείο ελέγχου (ο ACS στη συγκεκριμένη περίπτωση) μπορεί να ψάχνει στα περιεχόμενα του εξυπηρετητή για συγκεκριμένα αρχεία. Η μέθοδος αυτή επιστρέφει το URI αναγνωριστικό ενιαίου πόρου το οποίο μπορεί να χρησιμοποιηθεί πλέον ως αναφορά για το συγκεκριμένο αρχείο. Στη συνέχεια (STEP4) ο ACS διαχειριστής γνωστοποιεί, μέσω της γέφυρας, στον *UPnP Media Renderer* το αρχείο, που επιθυμεί να παρουσιαστεί στο συνδρομητή. Αυτό πραγματοποιείται μέσω της UPnP υπηρεσίας *AVTransport*, η οποία παρέχεται από τον *UPnP Media Renderer* για το έλεγχο της Video ροής. Μάλιστα η υπηρεσία αυτή περιέχει διάφορες UPnP μεθόδους, που η καθεμιά εκτελεί και από μια ξεχωριστή ενέργεια (*Stop, Play, Pause, Seek, Next, Previous, Record, SetAVTransportURI* κ.α.). Από την παραπάνω υπηρεσία ο ACS καλεί την απομακρυσμένη μέθοδο *SetAVTransportURI* με όρισμα το URI που έλαβε κατά το προηγούμενο στάδιο (STEP3). Αφού τελειώσει η διαδικασία αυτή ο *UPnP Media Renderer* είναι πλέον έτοιμος για την αναπαραγωγή του αρχείου που του ζητήθηκε. Στη συνέχεια (STEP5) ο ACS καλεί, μέσω της γέφυρας, στον *UPnP Media Renderer* τη μέθοδο *Play* της υπηρεσίας *AVTransport*. Η μέθοδος αυτή επιβάλλει στον *UPnP Media Renderer* να λάβει το αρχείο από τον *UPnP Media Server* (με HTTP GET) και το αναπαραγάγει.

Κατά τη διάρκεια της αναπαραγωγής ο συνδρομητής μπορεί να πατήσει διάφορα κουμπιά του *UPnP Media Renderer* (π.χ. "Pause", "Play"), που παράγουν ανεπιθύμητα μηνύματα ενημέρωσης συμβάντος (Eventing), αλλά ωστόσο είναι υποχρεωτικά για τη *AVTransport* υπηρεσία. Το μόνο ενδιαφέρον συμβάν για τον ACS είναι εκείνο που ενημερώνει για το τέλος του video clip. Η ενημέρωση για αυτό το συμβάν δίνει τη δυνατότητα στον πάροχο να μεταφορτώσει και να ξεκινήσει αυτόματα την αναπαραγωγή του επόμενου διαθέσιμου video clip. Για λόγους απλούστευσης το σχήμα 5-8 δεν περιλαμβάνει τη διαδικασία της ενημέρωσης συμβάντων.

Οι δύο UPnP συσκευές παρέμειναν σε λειτουργία για έξι συνεχόμενες ώρες προκειμένου να καταγραφούν οι ανά-διαφημίσεις (SSDP μηνύματα), που προβλέπει το UPnP πρωτόκολλο για λόγους ανίχνευσης.



Σχήμα 5-8. Το σενάριο του πειράματος και οι αντίστοιχες διαδράσεις μεταξύ του ACS, της πύλης-γέφυρας και των UPnP συνδρομητικών συσκευών.

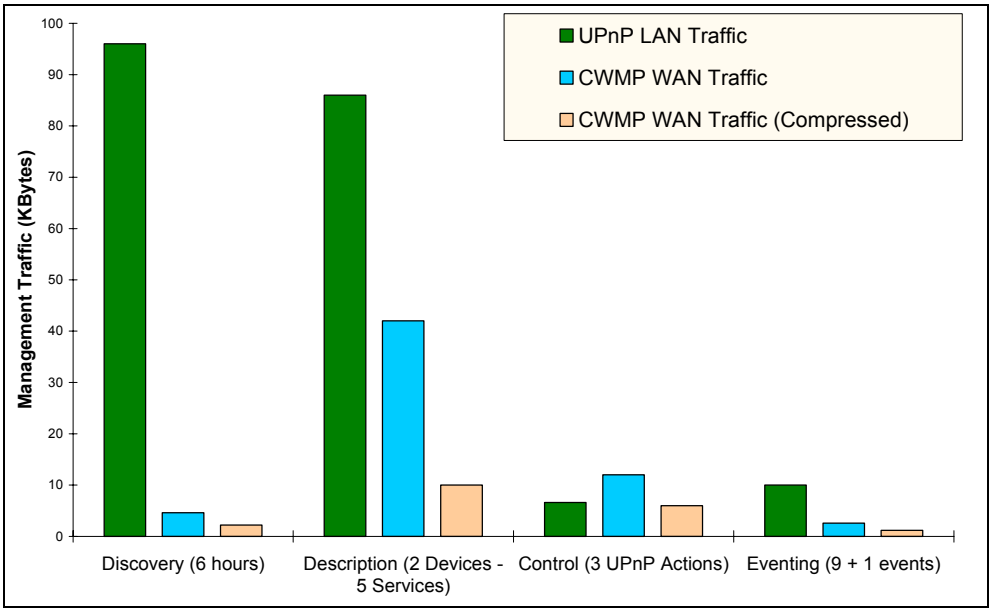
Στο σχήμα 5-9 απεικονίζονται συγκεντρωμένα τα αποτελέσματα του πειράματος. Η λειτουργία της UPnP ανίχνευσης φαίνεται να είναι αρκετά δαπανηρή όσον αφορά στη χρήση του εύρους ζώνης και στην επεξεργασία των μηνυμάτων. Οι UPnP συσκευές στο συγκεκριμένο πείραμα στέλνουν μηνύματα ανά-διαφημίσεων κάθε τριάντα λεπτά. Η τιμή αυτή είναι μια τυπική τιμή για τις περισσότερες εμπορικές υλοποιήσεις. Έτσι σε μία ημέρα για ένα δίκτυο συνδρομητή με δύο μόνο UPnP συσκευές ο όγκος της παραγόμενης κίνησης για την ανίχνευση ξεπερνά τα 380 Kbytes και τα 1080 SSDP μηνύματα εκπομπής. Όλα αυτή η κίνηση απορροφάται από τη γέφυρα η οποία για αυτό το σενάριο στέλνει στον ACS διαχειριστή μόνο δύο CWMP (*Inform*) μηνύματα. Ένα για κάθε συσκευή, που εισέρχεται στο τοπικό δίκτυο. Η αντίστοιχη κίνηση είναι λιγότερη από 5Kbytes.

Κατά τη διάρκεια της διαδικασίας περιγραφής (η οποία για λόγους απλούστευσης δεν απεικονίζεται στο σχήμα 5-8) η γέφυρα μεταφορτώνει από τις UPnP συσκευές όλα τα διαθέσιμα αρχεία περιγραφής. Συνολικά μεταφορτώνει δύο αρχεία περιγραφής συσκευής (ένα για τον *UPnP Media Renderer* και ένα για τον *UPnP Media Server*) και πέντε αρχεία περιγραφής υπηρεσίας (ένα για κάθε υπηρεσία των UPnP συσκευών). Ωστόσο για την πραγματοποίηση του παραπάνω σεναρίου ο ACS διαχειριστής χρειάζεται και μεταφορτώνει τα αρχεία περιγραφής μόνο των δύο υπηρεσιών που αναφέρθηκαν παραπάνω (*Content Directory* και *AVTransport*). Γενικά η διαδικασία αυτή αναμένεται να συμβαίνει σπάνια διότι ο ACS μπορεί να αποθηκεύει και ένα επαναχρησιμοποιεί τα αρχεία περιγραφής των συνδρομητικών συσκευών ή ακόμα και τις ίδιες τις UPnP SOAP εντολές διαχείρισης.

Όπως αναμένονταν, η διαδικασία του CWMP ελέγχου των UPnP συσκευών χρησιμοποιεί μεγαλύτερο εύρος ζώνης από εκείνο που χρησιμοποιεί το UPnP σημείο ελέγχου της πύλης-γέφυρας μέσα στο τοπικό δίκτυο. Η ενθυλάκωση των UPnP SOAP μηνυμάτων μέσα στα CWMP μηνύματα καθώς και η ανάγκη του ACS να ζητά πίσω τη UPnP SOAP απάντηση προκαλούν τον υπερκεφαλισμό αυτό. Όπως προαναφέρθηκε, κατά το στάδιο του ελέγχου (STEP3-5) κλήθηκαν τρεις UPnP μέθοδοι (υπό τη μορφή SOAP RPCs) και ο συνολικός υπερκεφαλισμός στο διακινούμενο όγκο δεδομένων άγγιξε το 80% (σχήμα 5-9, Control). Ωστόσο ο ACS είναι δυνατόν να μη χρειάζεται να συλλέξει τη UPnP SOAP απάντηση. Κάτι τέτοιο μπορεί να συμβεί σε περιπτώσεις, όπου η κληθείσα μέθοδος δεν επιστρέφει την τιμή κάποια μεταβλητής αλλά απλά αναφέρει την κατάσταση της εκτέλεσης. Τέτοιες UPnP SOAP απαντήσεις επεξεργάζονται από τη γέφυρα, η οποία τώρα χρησιμοποιεί ορισμένες σημαίες (flags) της *SetParameterValues* απάντησης (δηλαδή του πρώτου μηνύματος ελέγχου στο σχήμα 5-6) προκειμένου να ενημερώσει για την κατάσταση της εκτέλεσης. Μόνο σε περιπτώσεις αποτυχίας ο ACS διαχειριστής ζητά να λάβει πίσω τη UPnP-SOAP απάντηση για να την επεξεργαστεί (π.χ. για να διαβάσει τους UPnP κωδικούς σφάλματος). Στο εν λόγω πείραμα υλοποιήθηκε και αυτή η προσέγγιση και αποτιμήθηκε. Κατά την εφαρμογή λοιπόν αυτής την προσέγγισης ο συνολικός υπερκεφαλισμός, για το ίδιο σενάριο μειώθηκε στο 40%. Βεβαίως ένας ακόμα πολύ αποτελεσματικός τρόπος για να ξεπεραστούν οι παραπάνω υπερκεφαλισμοί είναι η εφαρμογή της συμπίεσης στους CWMP SOAP φακέλους, όπως προτάθηκε στο τρίτο κεφάλαιο της παρούσας διατριβής. Μάλιστα η συμπίεση μπορεί να εφαρμοστεί και για τις υπόλοιπες διαδικασίες. Όπως φαίνεται από το σχήμα 5-9 (CWMP

WAN Traffic - Compressed), όσο μεγαλύτερα είναι τα CWMP μηνύματα τόσο πιο αποτελεσματική είναι και η συμπίεση.

Κατά τη διάρκεια της αναπαραγωγής του video clip παρήχθησαν εσκεμμένα εννέα συμβάντα με το πάτημα των διαφόρων κουμπιών του *UPnP Media Renderer*. Για κάθε πάτημα, στέλνεται από τη UPnP συσκευή ένα νέο μήνυμα ενημέρωσης συμβάντος. Όταν η αναπαραγωγή φτάσει στο τέλος της, ο *UPnP Media Renderer* στέλνει αυτόματα ένα μήνυμα ενημέρωσης συμβάντος. Αυτό είναι το μοναδικό μήνυμα που σκανδαλίζει τη γέφυρα για να στείλει μια ενημέρωση συμβάντος στον ACS διαχειριστή. Η υπόλοιπη κίνηση της UPnP ενημέρωσης απορροφάται από τη γέφυρα.



Σχήμα 5-9. Τα πειραματικά αποτελέσματα της παραγόμενης κίνησης διαχείρισης.

5.9. Ανακεφαλαίωση - Συμπεράσματα

Η παρούσα ενότητα, που αποτελεί και την τελευταία συμβολή της διατριβής στο χώρο της XML διαχείρισης, εστίασε για πρώτη φορά στη γεφύρωση μεταξύ XML λύσεων διαχείρισης. Κίνητρο της εργασίας αυτής ήταν να αποδείξει ότι τα νέα SOAP/XML/HTTP πρότυπα απομακρυσμένης διαχείρισης, μπορούν να προωθηθούν και να επεκταθούν ραγδαία εάν ενσωματώσουν, ήδη εγκατεστημένους και επιτυχημένους μηχανισμούς τοπικής διαχείρισης.

Όπως φάνηκε και στο δεύτερο κεφάλαιο μέχρι σήμερα είχαν πραγματοποιηθεί αρκετές εργασίες (εξελικτικές λύσεις) με στόχο την ολοκλήρωση του SNMP πρωτοκόλλου με διάφορες XML λύσεις διαχείρισης. Το γεγονός ότι πριν το 2004 δεν υπήρχε κάποιο ώριμο πρότυπο XML διαχείρισης, σε συνδυασμό με την αδυναμία του SNMP να καλύψει τις ανάγκες για μαζική διαχείριση των επόμενης γενιάς συνδρομητικών συσκευών και υπηρεσιών (βλέπε δεύτερο κεφάλαιο) είχε ως αποτέλεσμα την αποτυχία των εξελικτικών λύσεων σε επίπεδο προτυποποίησης. Από την άλλη πλευρά η προσπάθεια των οργανισμών προτυποποίησης για μια ενιαία XML τεχνολογία διαχείρισης (ριζοσπαστικές λύσεις) έχει αποδώσει καρπούς τα τελευταία τρία χρόνια, ιδίως με τα CWMP και NetConf πρωτόκολλα

του DSL φόρουμ και της IETF, αντίστοιχα. Το γεγονός αυτό παρότρυνε το συγγραφέα να συμβάλλει στη γρήγορη εξάπλωση των προτυποποιημένων XML λύσεων διαχείρισης και πιο συγκεκριμένα εκείνης του πρωτοκόλλου CWMP. Σήμερα το CWMP είναι ίσως το ωριμότερο XML πρότυπο διαχείρισης, καθώς υποστηρίζεται ήδη από ένα τμήμα της αγοράς των δικτυακών πυλών, ενώ παράλληλα γίνονται εκτεταμένες προσπάθειες για την εφαρμογή του στον υπόλοιπο συνδρομητικό εξοπλισμό.

Συνδυάζοντας όλα τα παραπάνω με το γεγονός ότι το UPnP είναι ήδη μια ώριμη και ευρέως εφαρμοσμένη XML τεχνολογία τοπικής διαχείρισης, η παρούσα εργασία πρότεινε τη στρατηγική της ολοκλήρωσης των μηχανισμών τοπικής και απομακρυσμένης διαχείρισης του UPnP και του CWMP, αντίστοιχα. Μια τέτοια ενέργεια οδηγεί στην άμεση επέκταση του εύρους της δράσης διαχείρισης των παρόχων στα ήδη υπάρχοντα (UPnP) συνδρομητικά δίκτυα. Η παρεχόμενη αυτή ευκολία αναμένεται να προσελκύσει το ενδιαφέρον των παρόχων, που επιθυμούν την ταχύτερη εμπορική εκμετάλλευση πολλών υπηρεσιών, οι οποίες ωστόσο εξαρτώνται από τη δυνατότητα αυτόματης ρύθμισης και διαχείρισης μιας ομάδας συνδρομητικών συσκευών. Πιο συγκεκριμένα, το ισχυρό κίνητρο της απομακρυσμένης διαχείρισης του συνδρομητικού εξοπλισμού, μέσω μιας CWMP οικιακής πύλης έρχεται ίσως την καταλληλότερη χρονική στιγμή, που οι περισσότεροι πάροχοι και κατασκευαστές εξετάζουν με μεγάλη προσοχή την ενσωμάτωση μιας CWMP λύσης στα διαχειριστικά τους συστήματα. Το γεγονός αυτό μπορεί να συμβάλει καθοριστικά στην ευρεία αποδοχή του CWMP προτύπου από την αγορά της διαχείρισης. Μάλιστα η προτεινόμενη λύση έχει ήδη παρουσιαστεί στα μέλη της πρωτοβουλίας για την προτυποποίηση της οικιακής πύλης HGI (Home Gateway Initiative) [13] και έχει προσελκύσει το έντονο ενδιαφέρον πολλών εκ των μεγάλων δικτυακών χειριστών (France Telecom, British Telecom, Telefonica, Deutsche Telecom κ.α.).

Όσον αφορά στο τεχνικό κομμάτι της παρούσας εργασίας, αποδείχτηκε ότι η ιδέα ενός ενιαίου μηχανισμού, που βασίζει αποκλειστικά όλες τις λειτουργίες του (ανίχνευση, περιγραφή, έλεγχος και ενημέρωση συμβάντων) στις απομακρυσμένες κλήσεις διεργασιών (SOAP RPCs) διευκολύνει σημαντικά τέτοιου είδους ολοκληρώσεις. Ακόμα και για πολύπλοκα πρωτόκολλα, όπως είναι εκείνο του UPnP η πλήρης αντιστοίχιση των λειτουργιών διαχείρισης (Ανίχνευση, Περιγραφή, Έλεγχος και Ενημέρωση συμβάντος) φαίνεται να είναι μια σχετικά εύκολη διαδικασία, υπό την προϋπόθεση ότι η λύση της απομακρυσμένης διαχείρισης ακολουθεί αυτή την «όλα με απομακρυσμένες κλήσεις διεργασιών» προσέγγιση. Η παρούσα ανάλυση των λειτουργιών διαχείρισης, υποστηριζόμενη από πειράματα αποδεικνύει ότι ορισμένες λειτουργίες (π.χ. Ανίχνευση και Ενημέρωση συμβάντος για το UPnP) χρήζουν ιδιαίτερης προσοχής όταν αντιστοιχίζονται σε μηχανισμούς απομακρυσμένης διαχείρισης. Ο μεγάλος αριθμός των πακέτων, που ανταλλάσσονται στο δίκτυο (έτσι όπως επιβάλλουν οι μηχανισμοί τοπικής διαχείρισης) και επομένως ο όγκος της παραγόμενης κίνησης μπορεί να αποδειχτεί απαγορευτικός για μια μήνυμα- προς- μήνυμα αντιστοιχία. Έτσι λοιπόν τα θέματα φιλτραρίσματος στη γέφυρα που μεσολαβεί αποκτούν μείζονα σημασία για την πλήρη αξιοποίηση ενός καλοσχεδιασμένου προτύπου απομακρυσμένης διαχείρισης.

5.10. Αναφορές

[1] IETF, "Network Configuration (Netconf)," http://www.ietf.org/html.charters/netconf-charter.html.f

[2] DSL Forum, "CPE WAN management protocol", Technical Report. 069, May 2004.

[3] M.-J. Choi et al., "XML-Based Configuration Management for IP Network Devices," IEEE Communications Magazine, vol. 42, no. 7, July 2004.

[4] UPnP Forum, "UPnP device architecture 1.0", December 2003, http://www.upnp.org.

[5] IETF, HTTP Extension Framework, http://www.ietf.org/rfc2774.txt.

[6] W3C, "SOAP Version 1.2 Part 0: Primer," June 2003.

[7] UTF-8, a transformation format of ISO 10646 (character encoding), http://www.ietf.org/rfc2279.txt

[8] DSL Forum, "Data model template for TR-069 enabled devices", Technical Report 106, September 2005.

[9] Satoshi Konno, "Cyberlink for Java Development Package", http://www.cybergarage.org/.

[10] Apache Software Foundations, http://www.apache.org/.

[11] Intel, "Intel Tools for UPnP technologies", http://www.intel.com.

[12] UPnP Forum, "UPnP AV Architecture", http://www.upnp.org/specs/av/UPnP-av-AVArchitecture-v1-20020622.pdf.

[13] HGI, The Home Gateway Initiative, http://www.homegateway.org/.

Κεφάλαιο 6
Επίλογος

6.1. Ανασκόπηση

Η εξέλιξη των τεχνολογιών πρόσβασης σε συνδυασμό με τη μεγάλη ανάπτυξη και ευρεία αποδοχή των καταναλωτικών ηλεκτρονικών από την αγορά αλλά και την ενσωμάτωση σε αυτά δικτυακών τεχνολογιών, ανοίγουν νέους ορίζοντες για την παροχή καινοτόμων υπηρεσιών μέσω του διαδικτύου. Ένας από τους κρίσιμους παράγοντες για την επιτυχή διάδοση και εμπορική εκμετάλλευση των υπηρεσιών αυτών είναι το επίπεδο ευκολίας όσον αφορά στην αρχική εγκατάσταση, ρύθμιση και δια βίου συντήρηση των υπηρεσιών και των συσκευών, που τις φιλοξενούν. Η ελάχιστη ή ακόμη και μηδενική παρέμβαση του χρήστη αποτελεί βασική απαίτηση, δημιουργώντας έτσι την επιτακτική ανάγκη για απομακρυσμένη διαχείριση. Στο εγγύς μέλλον πολλοί και διαφορετικοί πάροχοι θα προσφέρουν στους συνδρομητές τους σύνθετες υπηρεσίες μέσω μιας ευρείας γκάμας πολύπλοκων συνδρομητικών συσκευών. Το νέο αυτό περιβάλλον δημιουργεί την ανάγκη για φτηνή, ασφαλή και ολοκληρωμένη διαχείριση δικτύων, συστημάτων και εφαρμογών. Το ήδη επιτυχημένο SNMP πρωτόκολλο απομακρυσμένης διαχείρισης δεν είναι αρκετό για να ανταποκριθεί με επιτυχία στις σύγχρονες αυτές απαιτήσεις. Κατά καιρούς έχουν προταθεί διάφορες εναλλακτικές τεχνολογίες και λύσεις οι οποίες μέχρι και σήμερα για διαφόρους λόγους δεν έχουν υιοθετηθεί από τους χειριστές των δικτύων και τους παρόχους των υπηρεσιών.

Ωστόσο την τελευταία πενταετία παρατηρείται, τόσο σε εμπορικό επίπεδο όσο και σε επίπεδο προτυποποίησης μια στροφή προς τις τεχνολογίες ιστού και πιο συγκεκριμένα προς το υπόδειγμα SOAP/XML/HTTP. Πράγματι η πανταχού παρουσία του HTTP και η ασφάλειά του σε συνδυασμό με τις μεγάλες δυνατότητες της XML να περιγράφει πολύπλοκες δομές και παράλληλα να ενσωματώνεται με χαμηλό κόστος σε οποιαδήποτε εφαρμογή, υπόσχονται πολλά στη διαχείριση. Μάλιστα όταν σε αυτά έρχεται να προστεθεί ένα εξίσου με τα προηγούμενα διαδεδομένο πρότυπο για την πραγματοποίηση των κλήσεων των απομακρυσμένων διεργασιών διαχείρισης στους πράκτορες των συσκευών, τότε η συνολική λύση γίνεται πραγματικά ελκυστική για την αγορά της διαχείρισης. Ωστόσο όπως και με κάθε άλλη τεχνολογία στο παρελθόν, έτσι και εδώ υπήρξαν αρκετοί, που εξέφρασαν έντονες ανησυχίες για την καταλληλότητά του SOAP/XML/HTTP υποδείγματος στη διαχείριση.

Αντικειμενικός στόχος της παρούσας διατριβής είναι η υποστήριξη του SOAP/XML/HTTP υποδείγματος και η συμβολή στην ταχεία προώθηση των τεχνολογιών διαχείρισης που το υιοθετούν. Ειδικότερα, η μελέτη της διατριβής επικεντρώθηκε στα εξής:

- Αρχικά στη διερεύνηση της επίδρασης, που έχει η εφαρμογή του SOAP/XML στις επιδόσεις της διαχείρισης και κατόπιν στην εύρεση μεθόδων και τεχνικών για τη βελτίωση των επιδόσεων αυτών. Αφού διαπιστώθηκε ότι ο όγκος των SOAP/XML μηνυμάτων είναι αρκετά υψηλός και μπορεί να δημιουργήσει προβλήματα που σχετίζονται με το χρόνο διαχείρισης και την κλιμακωσιμότητα των σχετικών λύσεων, προτάθηκαν δύο συμπληρωματικές προσεγγίσεις για την αποφυγή τέτοιου είδους προβλημάτων. Η πρώτη συνίσταται στην *επιλεκτική συμπίεση σε πραγματικό χρόνο* των SOAP φακέλων, που μεταφέρουν την πληροφορία διαχείρισης. Η δεύτερη προτείνει την *αποφυγή της κλασσικής μεθόδου DOM* για την ανάλυση των SOAP/XML μηνυμάτων και υιοθετεί τους *ειδικούς γραμματικούς αναλυτές*.

- Στην ανάπτυξη μεθοδολογιών για τη μείωση του κόστους ανάπτυξης, δοκιμών και συντήρησης των πρακτόρων, οι οποίοι συμμορφώνονται με τα νέα SOAP/XML/ HTTP πρότυπα πρωτόκολλα απομακρυσμένης διαχείρισης. Για το πρόβλημα αυτό προτάθηκε μια ολοκληρωμένη μεθοδολογία, που βασίζεται στην *έκφραση του μοντέλου διαχείρισης με τη βοήθεια της γλώσσας XML*. Μάλιστα για την αποτίμηση της προσέγγισης αυτής αναπτύχθηκε μια σουίτα λογισμικού, η οποία παράγει αυτόματα, πηγαίο κώδικα για τους πράκτορες, που συμμορφώνονται με το νέο πρωτόκολλο απομακρυσμένης διαχείρισης συνδρομητικών συσκευών, CWMP.

- Στην πρόταση και ανάπτυξη λύσης για την επέκταση της διαλειτουργικότητας των νέων SOAP/XML/HTTP προτύπων απομακρυσμένης διαχείρισης . Για το πρόβλημα αυτό προτάθηκε η ιδέα της *γεφύρωσης των νέων προτύπων απομακρυσμένης διαχείρισης* με ήδη επιτυχημένα και ευρέως υποστηριζόμενα *πρότυπα τοπικής διαχείρισης*. Μάλιστα για την αποτίμηση της εφαρμοσιμότητας της προσέγγισης αυτής σχεδιάστηκε και υλοποιήθηκε μια τέτοια γέφυρα. Η γέφυρα αυτή ενσωματώνεται σε οποιαδήποτε συνδρομητική συσκευή και επιτρέπει, μέσω του νέου πρωτοκόλλου CWMP την απομακρυσμένη διαχείριση των συσκευών, που υποστηρίζουν το UPnP πρωτόκολλο τοπικής διαχείρισης.

Όλα αυτά παρουσιάστηκαν σε ένα ολοκληρωμένο πλαίσιο, το οποίο ξεκινάει από ένα θεωρητικό υπόβαθρο με αναφορά στις βασικές τεχνολογίες διαχείρισης και στα πλεονεκτήματα της XML διαχείρισης και καταλήγει στα ειδικότερα προβλήματα μελέτης της διατριβής. Έτσι το πρώτο κεφάλαιο, το οποίο αποτελεί την εισαγωγή της διατριβής, προέβαλλε την ανάγκη, που έχει δημιουργηθεί τα τελευταία χρόνια για μαζική διαχείριση των IP συνδρομητικών συσκευών. Στο δεύτερο κεφάλαιο, παρουσιάστηκαν και αξιολογήθηκαν οι υποψήφιες τεχνολογίες για αυτού του είδους τη διαχείριση. Μέσα από την ανάλυση διακρίθηκαν οι λύσεις, που βασίζονται στις τεχνολογίες ιστού και πιο συγκεκριμένα εκείνες, που χρησιμοποιούν το συνδυασμό των SOAP, XML και HTTP προτύπων της W3C. Τα κεφάλαια 3, 4 και 5 αποτελούν τις τρεις βασικές ενότητες της διατριβής, καθεμία εκ των

οποίων αντιστοιχεί σε ένα από τα αντικείμενα μελέτης, που αναλύθηκαν προηγουμένως (Επιδόσεις, Κόστος, και Διαλειτουργικότητα).

6.2. Γενικά συμπεράσματα και προτάσεις για μελλοντική έρευνα

Η υιοθέτηση του πρωτοκόλλου γενικής χρήσης SOAP σε συνδυασμό με την περιττότητα της XML στην έκφραση της πληροφορίας, έχουν σαν πρωτογενή συνέπεια την κατακόρυφη αύξηση του όγκου της παραγόμενης κίνησης. Μάλιστα όταν πρόκειται για διαχείριση συνδρομητικών συσκευών μέσω του δημοσίου δικτύου τότε δημιουργείται μια δευτερογενής συνέπεια, που σχετίζεται με τον υψηλό χρόνο διαχείρισης. Οι δυο αυτές συνέπειες μπορούν να επιφέρουν σημαντικά προβλήματα κλιμακωσιμότητας και κόστους, τα οποία είναι ανάλογα με τις απαιτήσεις διαχείρισης και αντιστρόφως ανάλογα με το εύρος ζώνης, που ορίζει κάθε φορά η τεχνολογία πρόσβασης. Για να προληφθεί και να αντιμετωπιστεί το πρόβλημα των επιδόσεων δεν συνίσταται η εγκατάλειψη του SOAP πρωτοκόλλου (ενός ευρέως διαδεδομένου προτύπου της W3C) και η διερεύνηση ενός νέου, ειδικά σχεδιασμένου για τη διαχείριση. Αντίθετα προτείνεται μια πιο ήπια λύση, η οποία εφαρμόζεται άμεσα στα υπάρχοντα πρότυπα και μπορεί να μειώσει δραστικά τόσο τον όγκο της παραγόμενης κίνησης όσο και το χρόνο διαχείρισης. Η πρόταση αυτή συνίσταται σε δύο συμπληρωματικές προσεγγίσεις.

Η πρώτη έχει να κάνει με την εφαρμογή σε πραγματικό χρόνο των αντιστρεπτών αλγορίθμων λεξικής συμπίεσης στα SOAP/XML μηνύματα διαχείρισης. Τα πειραματικά αποτελέσματα αποδεικνύουν ότι οι αλγόριθμοι αυτοί μπορούν να μειώσουν δραστικά τον όγκο της κίνησης διαχείρισης αλλά και το χρόνο διαχείρισης. Αρκεί κατά την υλοποίησή τους να δίνεται έμφαση στην υψηλή ταχύτητα εκτέλεσης. Ο γνωστός αλγόριθμος Lempel-Ziv και η LZO υλοποίησή του αποτελούν ένα τέτοιο παράδειγμα. Ωστόσο ο Lemplel-Ziv δεν παύει να είναι μια γενική μέθοδος λεξικής συμπίεσης. Θα είχε ιδιαίτερο νόημα λοιπόν η περαιτέρω διερεύνηση αλγορίθμων εξειδικευμένων στην XML συμπίεση ή ακόμα καλύτερα στη συμπίεση SOAP/XML φακέλων. Εξάλλου η εργασία του τρίτου κεφαλαίου έχει πλέον ανοίξει το δρόμο για την εύκολη αξιολόγηση της καταλληλότητας όλων των αλγορίθμων συμπίεσης στην XML διαχείριση. Ένα ακόμη θέμα με το οποίο δεν καταπιάστηκε η παρούσα διατριβή και το οποίο χρήζει διερεύνησης είναι οι επιπτώσεις, που ενδεχομένως να έχει στις επιδόσεις και κυρίως στο χρόνο διαχείρισης η ενεργοποίηση της ασφάλειας του HTTP πρωτοκόλλου (HTTPS). Η κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων διαχείρισης συνιστά μια επιπλέον διαδικασία, η οποία αναμένεται να είναι ιδιαίτερα χρονοβόρα, κυρίως για συσκευές με χαμηλές δυνατότητες επεξεργασίας.

Η δεύτερη προσέγγιση για τη βελτίωση των επιδόσεων συνίσταται στην αποφυγή της κλασσικής και πρότυπης μεθόδου DOM για τη γραμματική ανάλυση των SOAP/XML φακέλων. Οι μεγάλοι υπερκεφαλισμοί επεξεργασίας της DOM ανάλυσης αλλά και οι απαιτήσεις της σε μνήμη την καθιστούν ακατάλληλη για μια υψηλών απαιτήσεων διαχείριση. Αντίθετα η χρήση γραμματικών αναλυτών, ειδικά σχεδιασμένων για το κάθε πρωτόκολλο διαχείρισης δίνει εντυπωσιακά αποτελέσματα, που συνίστανται στη μείωση του χρόνου διαχείρισης και στην αποδοτικότερη χρήση της μνήμης. Βέβαια το τίμημα που πληρώνει

κανείς για τη χρήση των ειδικών αναλυτών έναντι των DOM είναι το κόστος υλοποίησης για κάθε νέο πρωτόκολλο διαχείρισης και το κόστος της συντήρησής τους για κάθε νέα έκδοση του πρωτοκόλλου. Επιπρόσθετα η θεωρητική ανάλυση του τρόπου λειτουργίας των SAX αναλυτών δείχνει καλύτερες επιδόσεις από εκείνες των DOM αλλά χειρότερες από εκείνες των ειδικών αναλυτών. Από την άλλη πλευρά οι SAX αναλυτές είναι γενικής χρήσης. Ίσως λοιπόν η προσέγγιση των SAX αναλυτών να αποτελεί μια λύση, που συνδυάζει τις επιδόσεις με το χαμηλό κόστος υλοποίησης. Ωστόσο αυτό είναι ένα θέμα προς περαιτέρω διερεύνηση, καθώς για να αποφανθεί κανείς με ασφάλεια χρειάζονται πειραματικά αποτελέσματα. Εξάλλου η παρούσα εργασία έχει ήδη δώσει τη μεθοδολογία υλοποίησης και αξιολόγησης σχετικών πειραμάτων.

Η δεύτερη ενότητα αυτής της διατριβής προέβαλλε την αναμενόμενη πολυπλοκότητα στην υλοποίηση, στον έλεγχο και στη συντήρηση των πρακτόρων διαχείρισης εξαιτίας της συνθετότητας των σύγχρονων συσκευών, υπηρεσιών και προτύπων διαχείρισης,. Η ιδέα της δημιουργίας ομάδων εργασίας με εξειδικευμένους μηχανικούς στα νέα πρότυπα και στις εφαρμογές, που υλοποιούν τις υπηρεσίες αναμένεται να ανεβάσει σημαντικά το κόστος διαχείρισης των συνδρομητικών συσκευών. Στην παρούσα εργασία προτάθηκε μια διαφορετική προσέγγιση. Σύμφωνα με αυτή, η εξειδικευμένη γνώση μπορεί να ενσωματώνεται σε έτοιμα εργαλεία, που απαλλάσσουν τους μηχανικούς σχεδίασης και εφαρμογών από τις πολλές λεπτομέρειες των σύγχρονων πρωτοκόλλων διαχείρισης. Μάλιστα αποδείχτηκε ότι η έκφραση των προτυποποιημένων μοντέλων διαχείρισης των συσκευών σε XML μορφή μπορεί να οδηγήσει σε πλήρως αυτοματοποιημένες λύσεις υλοποίησης, δοκιμών και δια βίου συντήρησης του λογισμικού των πρακτόρων διαχείρισης. Επιπρόσθετα, εάν οι παραπάνω λύσεις εστιάσουν στη μέγιστη δυνατή αυτοματοποίηση της συνολικής διαδικασίας, τότε ένας πολύ μικρός αριθμός μηχανικών μπορεί να είναι αρκετός. Η εν λόγω ενότητα πρότεινε ένα ολοκληρωμένο σύνολο από διαδικασίες αυτοματοποίησης. Βεβαίως θα μπορούσε κανείς να επινοήσει ένα πλήθος εναλλακτικών τεχνικών και υλοποιήσεων, με στόχο την αύξηση του βαθμού αυτοματοποίησης και τελικά την περαιτέρω μείωση του συνολικού κόστους εφαρμογής των νέων προτύπων διαχείρισης. Ωστόσο η συμβολή της παρούσας εργασίας έγκειται κυρίως στον τρόπο με τον οποίο μπορεί κανείς να εκφράσει και να εκμεταλλευτεί το XML μοντέλο διαχείρισης της συσκευής προκειμένου να αυτοματοποιήσει αρκετές διαδικασίες, που σχετίζονται με τη διαχείριση.

Η τρίτη και τελευταία ενότητα πρότεινε τη στρατηγική της ολοκλήρωσης των νέων XMLπροτύπων απομακρυσμένης διαχείρισης με τα ήδη δημοφιλή και επιτυχημένα πρότυπα τοπικής διαχείρισης. Μάλιστα σαν παράδειγμα προς μελέτη χρησιμοποιήθηκε το νέο και πολλά υποσχόμενο πρότυπο απομακρυσμένης διαχείρισης, CWMP και το ευρέως εφαρμοσμένο πρότυπο τοπικής διαχείρισης, UPnP. Η σχετική ευκολία κατά τη γεφύρωση των παραπάνω πρωτοκόλλων απέδειξε ότι η διαλειτουργικότητα των νέων XML προτύπων απομακρυσμένης διαχείρισης με τις τεχνολογίες τοπικής διαχείρισης είναι μια εφαρμόσιμη και ιδιαίτερα ελκυστική λύση για τους παρόχους. Το ισχυρό κίνητρο της απομακρυσμένης διαχείρισης του συνδρομητικού εξοπλισμού, μέσω μιας γέφυρας που ενσωματώνεται στη δικτυακή πύλη, παρέχει στην αντίστοιχη τεχνολογία σημαντικό πλεονέκτημα για την

επικράτησή της στην αρένα της απομακρυσμένης διαχείρισης. Επίσης μπορεί να επιταχύνει σημαντικά την ανάπτυξη και παροχή υπηρεσιών, που απαιτούν τη συνεχή ρύθμιση και παρακολούθηση ενός πλήθους ηλεκτρονικών συσκευών στις οποίες μέχρι και σήμερα οι πάροχοι δεν έχουν καμία πρόσβαση.

Ωστόσο για να αποδειχτεί η πραγματική αξία της εν λόγω προσέγγισης χρειάζεται περαιτέρω διερεύνηση, η οποία μάλιστα θα πρέπει να κινηθεί σε δύο άξονες: (i) στην αναγνώριση υπηρεσιών και εμπλεκόμενων συσκευών (σενάρια χρήσης), που θα χρειαστούν απομακρυσμένη διαχείριση στο μέλλον και (ii) στην επίλυση προβλημάτων τα οποία σχετίζονται με την ασφαλή διαχείριση της εξουσιοδότησης, που πρόκειται να αποκτούν οι πάροχοι στις συνδρομητικές συσκευές αλλά και στην οριοθέτησή αυτής. Το τελευταίο ζήτημα θεωρείται ιδιαίτερα κρίσιμο για την επιτυχία όλων των παραπάνω, καθώς η τεχνολογία θα πρέπει να παρέχει τα κατάλληλα εχέγγυα στους συνδρομητές προκειμένου να τους πείσει να δεχτούν, σε ένα βαθμό τον απομακρυσμένο έλεγχο των συσκευών τους.