NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCES
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

PROGRAM OF POSTGRADUATE STUDIES

PhD THESIS

Learning Enhanced Situation Perception for Self-Managed Networks

Panagiotis P. Spapis

ATHENS

February 2015

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Αντίληψη Κατάστασης Ενισχυμένη με Γνώση για Αυτοδιαχειριζόμενα Δίκτυα

Παναγιώτης Π. Σπαπής

ΑΘΗΝΑ

Φεβρουάριος 2015

PhD THESIS

Learning Enhanced Situation Perception for Self-Managed Networks

Panagiotis P. Spapis

SUPERVISOR: Athanasia (Nancy) Alonistioti, NKUA Assistant Professor

THREE-MEMBER ADVISORY COMMITTEE:
Athanasia (Nancy) Alonistioti, NKUA Assistant Professor
Lazaros Merakos, NKUA Professor
Efstathios Hadjiefthymiades, NKUA Associate Professor

SEVEN-MEMBER EXAMINATION COMMITTEE

Athanasia (Nancy) Alonistioti,
NKUA Assistant Professor

Lazaros Merakos,
NKUA Professor

Efstathios Hadjiefthymiades,
NKUA Associate Professor

Ioannis Stavrakakis,
NKUA Professor

Dimitrios Syvridis,
NKUA Professor

Dimitrios Varoutas,
NKUA Assistant Professor

Maria Virvou,
UNIPI Professor

Examination Date 25/2/2015

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Αντίληψη Κατάστασης Ενισχυμένη με Γνώση για Αυτοδιαχειριζόμενα Δίκτυα

Παναγιώτης Π. Σπαπής

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Αθανασία (Νάνση) Αλωνιστιώτη, Επίκουρη Καθηγήτρια ΕΚΠΑ

ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ:

Αθανασία (Νάνση) Αλωνιστιώτη, Επίκουρη Καθηγήτρια ΕΚΠΑ
Λαζαρος Μεράκος, Καθηγητής ΕΚΠΑ
Ευστάθιος Χατζηευθυμιάδης, Αναπληρωτής Καθηγητής ΕΚΠΑ

ΕΠΤΑΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Αθανασία (Νάνση) Αλωνιστιώτη,
Επίκουρη Καθηγήτρια ΕΚΠΑ

Λάζαρος Μεράκος,
Καθηγητής ΕΚΠΑ

Ευστάθιος Χατζηευθυμιάδης,
Αναπληρωτής Καθηγητής ΕΚΠΑ

Ιωάννης Σταυρακάκης,
Καθηγητής ΕΚΠΑ

Δημήτριος Συβρίδης,
Καθηγητής ΕΚΠΑ

Δημήτριος Βαρουτάς,
Επίκουρος Καθηγητής ΕΚΠΑ

Μαρία Βίρβου,
Καθηγήτρια ΠΑΠΕΙ

Ημερομηνία εξέτασης 25/2/2015

ABSTRACT

The networks in the future it is envisaged that they will be able to operate in an autonomous manner. In other words, it is consider that the networks will monitor their environment, analyze the environment stimuli, plan their operation and execute their plan. Moreover, the networks should be efficient and adaptable solutions so as to cover the diverse network requirements, ranging from typical human traffic, to ultra reliable or massive machine type traffic.

Towards this direction, this thesis aims at providing a scheme for situation aware networking, based on a hierarchical architecture, which enables the network elements to operate in a self managed way. We propose the introduction of two levels of hierarchy in the network management and control, the Network Element Controllers (NEC), and the Network Domain Controllers (NDC). The first ones have local network view and may proceed in handling of local problems, whereas the later have broader network view and may identify optimization opportunities or problems that are related to larger network compartments. Both NECs and NDCs are able to characterize their environment and identify their operational status, a functionality defined as situation perception. The development of the previously mentioned situation perception mechanisms, based on fuzzy reasoners, is the second major contribution of this thesis. Fuzzy logic with its environment modeling mimics human logic, with the inference system based on policies. This scheme is suitable for identifying optimization goals and faults in the network. The developed Situation schemes target QoS degradation events' identification, Load events' identification, and Cooperative power control.

The third major contribution of this dissertation is the proposal of two adaptation schemes for the enhancement of the situation perception mechanisms. The enhancement is related to the adaptation of the environment modeling of the fuzzy reasoners. This functionality enables the network elements to operate in new, unknown environments. The presentation of a generic reference problem, which is linked to the learning schemes, enables the application of these solutions in other problems, with similar formulations. Finally, the learning schemes are compared so as to provide directions on how these schemes may be used in other similar problems on the one hand, and what are the drawbacks and benefits of each scheme on the other.

Concluding, it should be mentioned that this dissertation aims at giving a holistic approach in the problem of the Situation Perception problem. Initially, we define the architectural framework, which should be followed. The fuzzy reasoners target the

actual situation perception functionality, which is further enhanced with adaptation mechanisms for enabling the network elements to operate in totally new environments.

SUBJECT AREA: Communication Networks

KEYWORDS: autonomic networking, wireless networks, situation awareness, situation perception, fuzzy logic, supervised learning, unsupervised learning.

ΠΕΡΙΛΗΨΗ

Τα μελλοντικά δίκτυα θα έχουν τη δυνατότητα να λειτουργούν με αυτόνομο τρόπο. Αυτό σημαίνει οτι θα μπορούν να λειτουργούν χωρίς να απαιτούν ανθρώπινη παρέμβαση για τη λειτουργία τους. Τα συστήματα αυτά έχουν μπορούν να διαχειρίζονται ένα σύνολο απο λειτουργίες, που περιλαμβάνουν να την παρατήρηση του περιβάλλοντος, την αντίληψη της κατάστασης στην οποία βρίσκονται, την αντίληψη αν η κατάσταση αυτή είναι προβληματική, τη λήψη αποφάσεων για την αντιμετώπιση των προβληματικών καταστάσεων και στη συνέχεια την αλληλεπίδραση με το περιβάλλον τους βάσει των προηγούμενων αποφάσεων τους. Επιπλέον, τα δίκτυα θα πρέπει να μπορούν να μεταβάλλουν τον τρόπο λειτουργίας τους, ώστε να καλύπτουν τις συγχρονες ανάγκες για κίνηση που προέρχεται απο υπηρεσίες φωνής και δεδομένων, υπηρεσίες υψηλής αξιοπιστίας, υπηρεσίες παρατήρησης περιβάλλοντος απο αισθητήρες κοκ.

Προσπαθώντας να αντιμετωπίσει το ανωτέρω πρόβλημα, η παρούσα διατριβή προτείνει ένα μοντέλο αυτόνομης λειτουργίας δικτύων, το οποίο βασίζεται σε ιεραρχική δομή, που δινει τη δυνατότητα στα δικτυακά στοιχεία να λειτουργούν με αυτοδιαχειριζόμενο τρόπο. Στα πλαίσια αυτά προτείνεται η εισαγωγή δύο επιπέδων ιεραρχίας στη διαχείριση των δικτυων, αυτό του ελεγκτήρα δικτυακού στοιχείου, και αυτού του ελεγκτήρα δικτυακού τομέα. Ο πρώτος αναλαμβάνει να διαχειριστει λειτουργίες ελέγχου, αντιμετωπίζοντας τοπικά προβλήματα, βάσει της τοπικής εικόνας που έχει. Από την άλλη πλευρά, ο ελεγκτήρας δικτυακού τομέα αναλαμβάνει να διαχειριστεί λειτουργίες ελέγχου στα πλαίσια μια ευρύτερης δικτυακής γειτονιάς, που περιλαμβάνει περισσότερα από ένα δικτυακά στοιχεία, εκμεταλλευόμενος την ευρύτερη εικόνα που έχει. Και οι δύο ελεγκτήρες μπορούν να αναλύουν δεδομένα από το περιβάλλον τους και να καταλήγουν στην αναγνώριση προβληματικών καταστάσεων, μία λειτουργία που ονομάζεται αντίληψη κατάστασης. Στη συγκεκριμένη διατριβή προτείνεται η χρήση ασαφούς λογικής για την αντιμετώπιση του προβλήματος της αντίληψης κατάστασης. Η ασαφής λογική είναι ένα ιδανικό εργαλείο για διαχείριση πολυκριτηριακών προβλημάτων, με αντικρουόμενες εισόδους, που ενδεχομένως σχετίζονται με απώλεια δεδομένων. Η συγκεκριμένη πρόταση εφαρμόστηκε σε τρία δικτυακά προβλήματα, αυτό της αντίληψης κατάστασης φόρτου σε WiFi σημεία πρόσβασης (WiFi Access Points), αυτό της αντίληψης κατάστασης χαμηλής ποιότητας παρεχόμενης υπηρεσίας σε τερματικές συσκευές χρήστη για VoIP υπηρεσίας και αυτό

της αντίληψης κατάστασης περιβάλλοντος για συνεργατική ρύθμιση ισχύος σε WiFi σημεία πρόσβασης.

Στη συνέχεια, για να αποφευχθεί η αναγκη συνεχούς παραμετροποίησης των δικτυακών στοιχείων, δεδομένου ότι, όπως αναφέρθηκε νωρίτερα, το δικτυακό περιβάλλον ενδέχεται να μεταβάλλεται, προτείνεται η εισαγωγή μηχανισμών εκμάθησης που θα μεταβάλλουν τη λειτουργία των ελεγκήρων ασαφούς λογικής. Αναπτύχθηκαν δύο μηχανισμοί εκμάθησης που βασίζονται σε αλλαγή της μοντελοποίησης του περιβάλλοντος του μηχανισμού λήψης αποφάσεων. Ο πρώτος μηχανισμός εκμάθησης βασίζεται σε ένα σχήμα εποπτευμένης μάθησης, ενώ ο δεύτερος σε ένα σχήμα μη εποπτευμένης μάθησης. Οι δύο μηχανισμοί στηρίζονται σε μία βασική μοντελοποίηση του προβλήματος εκμάθησης (που επίσης προτείνεται στα πλαίσια αυτής της διατριβής) και αποσκοπεί στη γενίκευση των συγκεκριμένων μεθόδων εκμάθησης.

Κλέινοντας αξίζει να αναφερθεί οτι στα πλάισια της παρούσας διατριβής έγινε μία απόπειρα να αντιμετοπιστεί το πρόβλημα της αντίληψης κατάστασης συνολίκα, προτείνοντας μία λύση που περιλαμβάνει αφενός την βασική αρχιτεκτονική του προτεινόμενου δικτύου, καθώς επίσης και τους μηχανισμούς που επιτρέπουν την αντίληψη κατάστασης, και την εξέλιξη τους βάσει του περιβάλλοντος στο οποίο λειτουργούν.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Δίκτυα Επικοινωνιών
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: αυτόνομα δίκτυα, ασύρματα δίκτυα, επίγνωση κατάστασης, αντίληψη κατάστασης, ασαφής λογική, εποπτευμένης μάθηση, μη εποπτευμένης μάθηση

I would like to dedicate this dissertation to my parents Pavlos and Ligeri, to my brother Vassilis, to Giouli and to Katerina.

“Opportunities can’t wait” -Thucydides (460-394 B.C.)

“Life is a series of pulls back and forth... A tension of opposites, like a pull on a rubber band. Most of us live somewhere in the middle. ” - Mitch Albom, Tuesdays with Morrie

Στους γονείς μου Παύλο και Λυγερή, στον αδερφό μου Βασίλη, στη Γιούλη και στην Κατερινα.

“Οι καιροί ου μενετοί” – Θουκυδίδης (460-394 π.Χ.)

“Η ζωή είναι ένα σύνολο από βήματα μπροστά και πίσω... μία έλξη αντιθέτων, μία διελκυνστίνδα. Οι περισσότεροι από εμάς ζούμε κάπου στη μέση...” – Μιτς Αλμπομ, Το μεγαλύτερο μάθημα της ζωής

ACKNOWLEDGEMENTS

At the outset, I would like to express my sincere gratitude to my thesis advisory committee: Assistant Professor Nancy Alonistioti, Professor Lazaros Merakos and Associate Professor Efstathios Hadjiefthymiades.

I would like to thank and express my appreciation to my advisor, Assistant Professor Nancy Alonistioti for guiding and supporting me. Our academic interactions have always been very beneficial and thought provoking. I would also like to thank you for the experiences and the opportunities that you have provided me as well as the excellent collaboration that we had. Your support, ideas, and feedback have been absolutely invaluable for this thesis and my research activities.

I would also like to thank Professor Lazaros Merakos for his assistance during my post-graduate studies. The discussions regarding the research topics of my thesis helped me structuring my ideas. You have set an example of excellence as a researcher, and instructor.

Moreover, I would like to express my appreciation to colleague and friend Assistant Professor Alexandros Kaloxylos, for his help during the elaboration of this thesis. Our discussions have always been fruitful and ideas triggering. Also, the way you have been working, thinking, and living will be an example for me in the future.

I would like to sincerely thank my friends and (ex or present) colleagues Dr. Apostolos Kousaridas, Dr. Panagis Magdalinos, Makis Stamatelatos, George Katsikas, Konstantinos Chatzikokolakis, Dr. Eleni Patouni, Roi Arapoglou, Sokratis Barmpounakis, Panagiotis Theodoropoulos, Dr. Zacharias Boufidis, Dr. Andreas Merentitis, Stefanos Falangitis, George Beinas, Petros Makris, Costas Polychronopoulos, Damianos Kypriadis, Aristotelis Glentis, Dimitris Makris, Christos Tsilopoulos, Thanos Makris, Christos Papazafeiropoulos, and Evangelos Rekkas. Furthermore, I would like to thank the younger members of the SCAN laboratory Lambros Katsikas, Angelos Groumas, Vasilis Sarris, Konstantina Dimtsa, Panagiotis Panagiotopoulos, Nick Syntichakis, Nick Bobetsis, and, Dimitris Soukaras. You have all directly or indirectly contributed in the shaping of this thesis and my personal research profile. I will never forget the endless and constructive debates we had for technological, philosophical, societal and historical issues, as well as the great times we had during breaks, recreational, and team activities.

I want to thank all members of SCAN laboratory whose friendly company, during my thesis work, was a source of great pleasure. I also thank the partners form the various European projects that I have been involved during my thesis work (E3, Self-NET, CONSERN, UniverSelf, SACRA, METIS 2020) and especially Dr. Stylianos Georgoulas, and Dr. Andrej Mihailovic.

Last but not least, I want to thank my parents Pavlos and Lygeri, my brother Vasilis, and his wife Giouli for their love, help and support all these years. I would also like to thank my friend Apostolos Ioannidis for his support from the beginning of this period of my life. Finally, I would also like to express my sincere thanks and appreciation to Katerina for her love, assistance and support in this journey. I would also like to refer especially in the last months of my Ph.D., which helped me understand how important is to have people supporting you. These months made me better person and helped us come closer.

ΣΥΝΟΠΤΙΚΗ ΠΑΡΟΥΣΙΑΣΗ ΔΙΔΑΚΤΟΡΙΚΗΣ ΔΙΑΤΡΙΒΗΣ

Η παρούσα διατριβή πραγματεύεται την θεματική περιοχή των αυτόνομων συστημάτων και πιο συγκεκριμένα στους μηχανισμούς αντίληψης κατάστασης αυτοδιαχειριζόμενων δικτύων. Πιο συγκεκριμένα, παρουσιάζει ένα μηχανισμό αντίληψης κατάστασης βασισμένο σε συστήματα ασαφούς λογικής. Ο εν λόγω μηχανισμός, αναλύεται σε συγκεκριμένα δικτυακά περιβάλλοντα, για διαφορετικά προβλήματα και αναλύονται τα μειονεκτήματά του. Ο μηχανισμός αποτελεί ενα σχήμα με το οποίο μπορεί ενα δικτυακό στοιχείο να μοντελοποιεί το περιβάλλον του και να παίρνει αποφάσεις για την κατάσταση στην οποία βρίσκεται. Ωστόσο, σε ενδεχόμενη αλλαγή των συνθηκών του περιβάλλοντος, το προαναφερθέν σχήμα αδυνατεί να λειτουργήσει το ίδιο καλά, επειδή η παραμετροποίηση του από ειδικούς στόχευε στην αρχικό περιβάλλον. Για τον παραπάνω λόγο, είναι απαραίτητη η ύπαρξη μηχανισμών εκμάθησης ώστε να εξελίσσεται το σχήμα αντίληψης τρέχουσας κατάστασης βασισμένο στην εξέλιξη των περιβάλλουσων συνθηκών και τελικά να απαλλάξουν το σύστημα απο τη χειροκίνητη παραμετροποίηση. Στα πλαίσια της διατριβής, τέτοιες λύσεις αναπτύχθηκαν και παρουσιάστηκαν για το υλοποιηθέν σχήμα αντίληψης κατάστασης.

Ένα αυτοδιαχειριζόμενο σύστημα ενσωματώνει όλες εκείνες τις λειτουργίες που του επιτρέπουν να παρατηρεί το περιβάλλον του, να αντιλαμβάνεται την κατάσταση στην οποία βρίσκεται, να αντιλαμβάνεται αν η κατάσταση αυτή ειναι προβληματική, να παίρνει αποφάσεις για την αντιμετώπιση των προβληματικών καταστάσεων και στη συνέχεια να αλληλεπιδρά με το περιβάλλον του βάσει των προηγούμενων αποφάσεων του. Η παραπάνω διαδικασία αποτελεί τη βάση λειτουργίας των αυτόνομων συστημάτων, όπως αυτή εχει περιγραφεί εκτενώς στη βιβλιογραφία. Επιπλέον των λειτουργιών που έχουν περιγραφεί στη βιβλιογραφία, εχουν αναλυθεί και ανάλογες αρχιτεκτονικές δικτύων για την υλοποίηση των αυτοδιαχειριζόμενων δικτύων.

Κύριο κομμάτι της λειτουργίας τέτοιων συστημάτων, ειναι η δυνατότητα του συστήματος (και των αντίστοιχων δικτυακών στοιχείων) να παρατηρούν το περιβάλλον και να προχωρούν σε ανάλυση της κατάστασης, για να αναγνωρίσουν προβληματικές καταστάσεις ή ενδεχόμενες ευκαιρίες βελτιστοποίησης της λειτουργίας του συστήματος. Η παραπάνω διαδικασία αναφέρεται ώς λειτουργικότητα επίγνωσης κατάστασης. Η συγκεκριμένη λειτουργία έχει προταθεί από διάφορους ερευνητές στη βιβλιογραφία. Στα πλαίσια της τρέχουσας διατριβής, αναλύθηκαν αντιπροσωπευτικές προτάσεις και προτείνεται μία νέα αρχιτεκτονική λύση, η οποία είναι καταλληλότερη για δικτυακές εφαρμογές, δεδομένων των ιδιαίτερων χαρακτηριστικών της πρότασης (δηλ. ανάλυσης/διάσπασης ενός προβλήματος σε υποπροβλήματα, ιεραρχική και συνεργατική δομή του συστήματος).

Η επίγνωση κατάστασης, μπορεί να αναλυθεί περεταίρω σε τρεις βασικές υπό-λειτουργίες, την αντίληψη κατάστασης, την ανάλυση κατάστασης και την πρόβλεψη. Σύμφωνα με την πρώτη υπό-λειτουργία, ένα δικτυακό στοιχείο μπορεί να αναλύσει τις μετρήσεις που λαμβάνει από το περιβάλλον και να καταλήξει σε ένα χαρακτηρισμό της τρέχουσας κατάστασης. Η δεύτερη λειτουργία εστιάζει στην ανάλυση των πιθανών δράσεων (ή λύσεων) για τη δεδομένη κατάσταση, ενώ η τρίτη αναφέρεται στην πραγματοποίηση προβλέψεων για τις καταστάσεις στις οποίες ενδέχεται να βρεθεί το σύστημα ή το δικτυακό στοιχείο στο μέλλον. Στη βιβλιογραφία και στις προσφερόμενες λύσεις από τους κατασκευαστές, έχει δοθεί βάση στην πραγματοποίηση προβλέψεων, ωστόσο η αντίληψη κατάστασης δεν έχει αναλυθεί επαρκώς. Πιο συγκεκριμένα, μέχρι στιγμής το εν λόγω πρόβλημα αντιμετωπίζεται με τη χρήση αυστηρών (στατικών ή δυναμικών) ορίων, λύση που δεν ειναι αποδοτική, επειδή η μετάβαση από μία κατάσταση σε μία άλλη δεν μπορεί να περιγραφεί με αυστηρά όρια. Για παράδειγμα, ένα σημείο πρόσβασης WiFi (Access Point) δεν μπορεί να θεωρείται με χαμηλό φόρτο με τέσσερις χρήστες και με υψηλό φόρτο οταν συνδεθεί ένας πέμπτος χρήστης σε αυτό, αντίστοιχα για τη διεκπεραίωση, κ.α.). Για το λόγο αυτό, στη συγκεκριμένη διατριβή προτείνεται η χρήση ασαφούς λογικής για την αντιμετώπιση του προβλήματος της αντίληψης κατάστασης. Η ασαφής λογική είναι ένα ιδανικό εργαλείο για διαχείριση πολυκριτηριακών προβλημάτων, με αντικρουόμενες εισόδους, που ενδεχομένως σχετίζονται με απώλεια δεδομένων. Η συγκεκριμένη πρόταση εφαρμόστηκε σε τρία δικτυακά προβλήματα, αυτό της αντίληψης κατάστασης φόρτου σε WiFi σημεία πρόσβασης (WiFi Access Points), αυτό της αντίληψης κατάστασης χαμηλής ποιότητας παρεχόμενης υπηρεσίας σε τερματικές συσκευές χρήστη για VoIP υπηρεσίας και αυτό της αντίληψης κατάστασης περιβάλλοντος για συνεργατική ρύθμιση ισχύος σε WiFi σημεία πρόσβασης. Και στις τρεις περιπτώσεις το προτεινόμενο σχήμα αντίληψης κατάστασης δοκιμάστηκε και αξιολογήθηκε τόσο με προσομοιώσεις όσο και με πειραματικές διατάξεις, και η λειτουργία του κρίθηκε ικανοποιητικότατη. Η συγκεκριμένη πρόταση για να συνεχίσει να λειτουργει εξίσου ικανοποιητικά σε περίπτωση που αλλάξουν οι συνθήκες του περιβάλλοντος, απαιτεί τη χειροκίνητη παραμετροποίηση του από ειδικούς εκ νέου.

Δεδομένης της ανάγκης για αποφυγή της συνεχούς χειροκίνητης παραμετροποίησης του συστήματος, αναπτύχθηκαν δύο μηχανισμοί εκμάθησης για τον μηχανισμό αντίληψης κατάστασης. Οι συγκεκριμένοι μηχανισμοί βασίζονται σε αλλαγή της μοντελοποίησης του περιβάλλοντος του μηχανισμού λήψης αποφάσεων. Πιο συγκεκριμένα σε έναν ελεγκτή ασαφούς λογικής τροποποιούνται τα ασαφή σύνολα που χρησιμοποιούνται για την μοντελοποίηση των εισόδων. Ο πρώτος μηχανισμός εκμάθησης βασίζεται σε ένα σχήμα εποπτευμένης μάθησης, ενώ ο δεύτερος σε ένα σχήμα μη εποπτευμένης μάθησης. Ο πρώτος χρησιμοποιεί μηχανισμούς από το ερευνητικό πεδίο της εξόρυξης γνώσης (k-Means, και kNN) για να δημιουργηθούν συστάδες που στη συνέχεια διασυνδέονται με τις εισόδους ώστε παραμετροποιηθεί εκ νέου η αντίληψη κατάστασης. Ο δεύτερος μηχανισμός εκμάθησης βασίζεται στη στατιστική ανάλυση του δείγματος για να παραμετροποιήσει την αντίληψη κατάστασης· προτείνονται δύο λύσεις σχετικές με την στατιστική ανάλυση του δείγματος. Η πρώτη βασίζεται στην γενικευμένη ανάλυση της κατανομής του δείγματος βρίσκοντας την γκαουσιανή κατανομή, ενώ η δεύτερη βασίζεται στην «ακριβή» κατανομή του δείγματος. Στη συνέχεια αυτές οι κατανομές χρησιμοποιούνται για τον χαρακτηρισμό ορίων των ασαφών συνόλων των ελεγκτών ασαφούς λογικής. Οι παραπάνω μηχανισμοί δοκιμάστηκαν και αξιολογήθηκαν τόσο για την αποτελεσματικότητα τους όσο και για την απόδοσή τους με προσομοιώσεις και πειραματικές διατάξεις στα σενάρια χρήσης που αναφέρθηκαν στην προηγούμενη παράγραφο (δηλ. αντίληψη κατάστασης φόρτου σε WiFi σημεία πρόσβασης (WiFi Access Points), αντίληψη κατάστασης χαμηλής ποιότητας παρεχόμενης υπηρεσίας σε τερματικές συσκευές χρήστη για VoIP υπηρεσίες και η αντίληψη κατάστασης περιβάλλοντος για συνεργατική ρύθμιση ισχύος σε WiFi σημεία πρόσβασης), και η λειτουργία του κρίθηκε ικανοποιητικότατη. Πιο συγκεκριμένα με την πρώτη μέθοδο (στις περιπτώσεις αντίληψης κατάστασης φόρτου σε WiFi Access Points, και αντίληψης κατάστασης χαμηλής ποιότητας παρεχόμενης υπηρεσίας σε τερματικές συσκευές χρήστη για VoIP υπηρεσίες), η βελτίωση στην απόδοση κυμάνθηκε από 10-20% ενώ με τη δεύτερη μέθοδο η ικανότητα του συστήματος να αντιλαμβάνεται τη σωστή κατάσταση αυξήθηκε μέχρι και 30%, με αντίστοιχο αντίκτυπο στο απαιτούμενο υπολογιστικό κόστος. Αντίστοιχα στην περίπτωση συνεργατικής ρύθμισης ισχύος σε WiFi APs παρατηρούμε σημαντική μείωση στην καταναλισκόμενη ενέργεια (έως και 30%) λόγω της ισχύος εκπομπής, βελτιώνοντας σημαντικά τα επίπεδα θορύβου.

Η συγκεκριμένη διατριβή περιλαμβάνει επτά κεφάλαια. Το πρώτο κεφάλαιο αποτελεί την εισαγωγή στο αντικείμενο της διατριβής. Σε αυτό το κεφάλαιο αποτυπώνονται οι τεχνολογικές εξελίξεις και οι προκλήσεις των μελλοντικών δικτύων επικοινωνιών. Οι προαναφερθείσες προκλήσεις αφορούν στη δυνατότητα των δικτυακών στοιχείων να λειτουργήσουν σε πολύ πυκνά δικτυακά περιβάλλοντα· η πυκνότητα σχετίζεται με τον αριθμό των δικτυακών συσκευών και την ενδεχόμενη διαλειτουργικότητα και συνεργασία τους. Στη συνέχεια, αναφέρονται τα κίνητρα για την εισαγωγή αυτόνομων δυνατοτήτων στις επικοινωνίες καθώς κινούμαστε προς την κατεύθυνση των αυτοδιαχειριζόμενων δικτύων. Επίσης συνοψίζονται τα βασικά προβληματα που πραγματεύθηκε η συγκεκριμένη διατριβή, καθώς επίσης και η συνεισφορά σε επίπεδο δικτυακής αρχιτεκτονικής, μηχανισμών, αλγορίθμων και αποτελεσμάτων. Στο τέλος του κεφαλαίου περιγράφεται η δομή της διατριβής.

Το δεύτερο κεφάλαιο ασχολείται με τις βασικές αρχές των αυτόνομων επικοινωνιών. Αρχικά δίνονται οι βασικοί ορισμοί των αυτόνομων επικοινωνιών και οι βασικές αρχές τους, όπως αυτές έχουν προταθεί και αναλυθεί στη βιβλιογραφία. Στη συνέχεια αναλύονται οι δραστηριότητες των οργανισμών προτυποποίησης (δηλ. ITU, ETSI, 3GPP, IETF, και IEEE) για τη δημιουργία αυτόνομων δικτύων. Ακολουθεί μία επισκόπηση της σύγχρονης βιβλιογραφίας όσον αφορά στα αυτόνομα συστήματα επικοινωνιών· πιο συγκεκριμένα συνοψίζονται οι δραστηριότητες της ερευνητικής κοινότητας και παρουσιάζονται τα κύρια χαρακτηριστικά της εκάστοτε ερευνητικής πρότασης. Το κεφάλαιο καταλήγει παρουσιάζοντας τις βασικές απαιτήσεις των αυτοδιαχειριζόμενων δικτύων.

Το τρίτο κεφάλαιο πραγματεύεται μία βασική έννοια των αυτόνομων και αυτοδιαχειριζόμενων δικτύων, αυτή της επίγνωσης κατάστασης. Αρχικά παρουσιάζονται οι έννοιες «δεδομένα», «πληροφορία», και «γνώση» και αναλύεται η συσχέτιση τους με την ικανότητα των δικτύων να λειτουργούν με αυτόνομο τρόπο. Στη συνέχεια παρουσιάζονται οι βασικές λειτουργικότητες των αυτόνομων δικτύων που είναι η παρακολούθηση του περιβάλλοντος, η λήψη αποφάσεων, και η εκτέλεσή τους (Monitor, Decide, Execute). Η ανωτέρω ανάλυση όμως, δεν καταφέρνει να μοντελοποιήσει ικανοποιητικά την επίγνωση κατάστασης, βασει της οποίας τα δικτυακά στοιχεία μπορούν να λειτουργήσουν αυτόνομα. Για το λόγο αυτό αναλύονται οι βασικές προσεγγίσεις επίγνωσης κατάστασης, όπως αυτές έχουν προταθεί στη βιβλιογραφία· έπειτα, προτείνεται μία νέα προσέγγιση όσον αφορά στην αρχιτεκτονική ενός μηχανισμού επίγνωσης κατάστασης, εστιασμένη στα συστήματα αυτόνομων επικοινωνιών που συνδυάζει τα πλεονεκτήματα των μηχανισμών της βιβλιογραφίας. Σύμφωνα με αυτή την προσέγγιση ένας μηχανισμός επίγνωσης κατάστασης για αυτόνομα συστήματα αναλύεται περεταίρω σε τρία στάδια, την αντίληψη κατάστασης, την κατανόηση της τρεχουσας κατάστασης και τις προβλέψεις. Στο πρώτο στάδιο ο λήπτης αποφάσεων καταλήγει σε ποια κατάσταση βρίσκεται, στο δεύτερο βρίσκει τις τρέχουσες εναλλακτικές λύσεις, ενώ στο τρίτο κάνει προβλέψεις για τις μελλοντικές καταστάσεις που ενδέχεται να βρεθεί. Ακολούθως, αναλύονται οι μηχανισμοί επίγνωσης κατάστασης που είναι διαθέσιμοι στη βιβλιογραφία και γίνεται μία συγκεντρωτική αποτίμηση τους. Η συγκεντρωτική αποτίμηση καταλήγει στο γεγονός ότι το πρώτο στάδιο, αυτό της αντίληψης κατάστασης, δεν έχει αναλυθεί επαρκώς, και αντιμετωπίζεται χρησιμοποιώντας αυστηρά όρια, στατικά ή δυναμικά ορισμένα. Το κεφάλαιο καταλήγει σχολιάζοντας βασικές προτάσεις κατασκευαστών, που επίσης προσεγγίζουν το προβλημα χρησιμοποιώντας στατικά ή δυναμικά όρια και καταλήγει στο γεγονός ότι το συγκεκριμένο προβλημα δεν εχει αναλυθεί επαρκώς και πιο εξελιγμένοι τρόποι αντιμετώπισης του εν λόγω προβλήματος χρειάζονται για την ικανοποιητική διαχείρισή του.

Το τέταρτο κεφάλαιο αναλύει τις βασικές αλγοριθμικές τεχνικές που χρησιμοποιούνται στη διατριβή. Πιο συγκεκριμένα το κεφάλαιο παρουσιάζει τα κύρια χαρακτηριστικά της ασαφούς λογικής (Fuzzy Logic) και των μηχανισμών εξόριξης γνώσης (Data Mining) και εκμάθησης (Learning). Όσον αφορά στην ασαφή λογική, αρχικά αναλύονται οι βασικές αρχές των ασαφών συνόλων (fuzzy sets), συμφωνα με τα οποία μία τιμή δεν ανήκει εξ ολοκλήρου σε μία κατάσταση, αλλά σε παραπάνω από μία, με συγκεκριμένο βαθμό για κάθε μια εξ αυτών. Ενισχύωντας τα ασαφή σύνολα με κανόνες μπορούμε να δημιουργήσουμε ελεγκτές ασαφούς λογικής, οι οποίοι είναι ιδανικοί για να αντιμετωπίζουν πολυκριτιριακά προβλήματα, με αντικρουόμενες εισόδους, που ενδεχομένως σχετίζονται με απώλεια δεδομένων. Το δεύτερο κομμάτι του κεφαλαίου αναλύει τις βασικές οικογένειες μηχανισμών εξόρυξης γνώσης, και πιο συγκεκριμένα αυτούς τις μη-εποπτευμένης μάθησης, της εποπτευμένης μάθησης και της ενισχυμένης μάθησης. Στη συνέχεια αναλύονται τα ιδιαίτερα χαρακτηριστικά της κάθε οικογένειας και συγκεκριμένοι αντιπροσωπευτικοί μηχανισμοί (k-Means, Hierarchical Clustering, kNN, και άλλοι), οι οποίοι στη πορεία χρησιμοποιήθηκαν για τον σχεδιασμό των μηχανισμών εκμάθησης που προτείνονται στα πλαίσια της εν λόγω διατριβής. Το κεφάλαιο καταλήγει στα ιδιαιτερα χαρακτηριστικά που οδήγησαν στην χρήση των εν λόγω μηχανισμών.

Το πέμπτο κεφάλαιο εστιάζει στη χρήση συστημάτων ασαφούς λογικής σε δικτυακά ερευνητικά θέματα για την αντιμετώπιση του προβλήματος της αντίληψης (δικτυακής) κατάστασης. Πιο συγκεκριμένα, αρχικά αναλύει την χρήση ελεγκτών ασαφούς λογικής για δικτυακά προβλήματα και πιο συγκεκριμένα αυτό της αντίληψης κατάστασης. Στη συνέχεια, για τρία συγκεκριμένα σενάρια χρήσης αναλύεται λεξιτενώς ο τρόπος με τον οποίο τα προβληματα έχουν μοντελοποιηθεί και αντιμετωπιστεί, καθώς επίσης και ο τρόπος με τον οποίο το προτεινόμενο σύστημα αντίληψης κατάστασης εχει υλοποιηθεί και αξιολογηθεί. Τα προβλήματα που προσεγγίστηκαν είναι η αντίληψη κατάστασης φόρτου σε WiFi σημεία πρόσβασης (WiFi Access Points), η αντίληψη κατάστασης χαμηλής ποιότητας παρεχόμενης υπηρεσίας σε τερματικές συσκευές χρήστη για VoIP υπηρεσίες και η αντίληψη κατάστασης περιβάλλοντος για συνεργατική ρύθμιση ισχύος σε WiFi σημεία πρόσβασης. Και στις τρεις περιπτωσεις προτείνεται η αρχιτεκτονική του συστήματος, η πλήρης μοντελοποίηση του συστήματος, και τέλος περιγράφεται ο τρόπος με τον οποίο το σύστημα έχει αξιολογηθεί· και στις τρεις περιπτώσεις τα αποτελέσματα κρίνονται ικανοποιητικότατα, δεδομένου οτι το σύστημα μπορεί να αντιληφθεί το περιβάλλον του καλύτερα και πιο ρεαλιστικά.

Στο έκτο κεφάλαιο παρουσιάζονται οι δύο μηχανισμοί εκμάθησης για τη βελτιστοποίηση του συστήματος αντίληψης κατάστασης ασαφούς λογικής, που προτάθηκε στο προηγούμενο κεφάλαιο. Οι μηχανισμοί, εκ των οποίων ο πρώτος βασίζεται σε εποπτευμένη μάθηση και ο δεύτερος σε μη-εποπτευμένη αποτελούν υβριδικές προσεγγίσεις μηχανισμών εξόρυξης γνώσης και στατιστικής ανάλυσης του δείγματος. Συγκεκριμένα, ο πρώτος αποτελείται από το πρώτο στάδιο, στο οποίο οι αποφάσεις οσον αφορά στην αντίληψη καταστασης αξιολογούνται σε σωστές και λάθος (labeled data), και στη συνέχεια αυτα τα δεδομένα χρησιμοποιούνται ως δεδομένα εκμάθησης (με τον αλγόριθμο kNN) για το υπόλοιπο δείγμα. Τέλος, τα δεδομένα αναλύονται χρησιμοποιώντας έναν αλγόριθμο συσταδοποίησης (k-Means), ώστε να είναι δυνατή η άμεση διασύνδεση με τις εισόδους (και πιο συγκεκριμένα με τις συναρτήσεις συμμετοχής – membership functions) των ελεγκτών ασαφούς λογικής. Ο δεύτερος μηχανισμός εκμάθησης, βασίζεται στη στατιστική αναλυση του δείγματος χωρίς προηγουμένως να εχει πραγματοποιηθεί αξιολογηση των ειλημμένων αποφάσεων (μη-εποπτευόμνη μάθηση) και έχει δύο παραλλαγές. Η πρώτη παραλλαγή βασίζεται στην εύρεση της Γκαουσιανής κατανομής του δείγματος ενώ η δεύτερη βασίζεται στην εύρεση της ακριβούς κατανομής του δειγματος για κάθε μεταβλητή εισόδου. Στη συνέχεια, οι παραπάνω συναρτήσεις εισάγονται στις εισόδους των ελεγκτών ασαφούς λογικής. Οι παραπάνω μηχανισμοί εφαρμόστηκαν στα τρία σενάρια χρήσης που περιγράφηκαν στο πέμπτο κεφάλαιο (αντίληψη κατάστασης φόρτου σε WiFi σημεία πρόσβασης (WiFi Access Points), αντίληψη κατάστασης χαμηλής ποιότητας παρεχόμενης υπηρεσίας σε τερματικές συσκευές χρήστη για VoIP υπηρεσίες και η αντίληψη κατάστασης περιβάλλοντος για συνεργατική ρύθμιση ισχύος σε WiFi σημεία πρόσβασης) και αξιολογήθηκαν βάσει της αποτελεσματικότητας τους να μοντελοποιούν ορθότερα το περιβάλλον τους. Σε συγκεκριμένες περιπτώσεις αξιολογήθηκε επιπλέον ο υπολογιστικός φόρτος που εισάγουν στο σύστημα, καθώς και το απαιτούμενο δείγμα για την επίτευξη ικανοποιητικών αποτελεσμάτων. Το κεφάλαιο κλείνει συγκρίνοντας τους προτεινόμενους αλγορίθμους και αναλύοντας τα ιδιατερα χαρακτηριστικά τους.

Τέλος, στο κεφάλαιο 7 συνοψίζεται η ερευνητική συνεισφορά της διατριβής. Επιπλέον αποτυπώνονται τα προβλήματα που εντοπίστηκαν και οι προτεινόμενες λύσεις. Η τρέχουσα διατριβή καταλήγει παρουσιάζοντας τις σημαντικότερες προεκτάσεις των ερευνητικών θεμάτων που μελετήθηκαν κατά την εκπόνηση της διατριβής και ενδιαφέρουσες μελλοντικές επεκτάσεις.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Future Networks Landscape

Moving towards future networks, the predictions indicate that mobile and wireless data traffic will increase considerably. Mobile data traffic will increase globally 13-fold between 2012 and 2017 whereas global IP traffic has increased more than fourfold in the past five years [1]. Additionally, a huge increase will take place regarding the number of the connected devices (100 billions). On the other hand, future networks will be more dynamic and complex compared to the contemporary ones, including new technologies, new services, and new demands from the users, new business cases [2]. Thus, the network elements will have to be agile and dynamic so as to manage the users' networking needs and the network operational environment. In such a complex environment, where billions of diverse devices will (simultaneously) ask for resources from the network, networks' self-management aroused as a potential solution.

For covering the identified needs, it is considered that future networks will have to combine all the available technologies (e.g., 3G, 4G, 5G, WLAN, etc.) and also will have to incorporate intelligence for providing efficiently the coverage, capacity and quality of service (QoS) requirements. For covering the requirements, new air interfaces, and signaling schemes will be required. Furthermore, the densification (i.e., the increase of the number of BSs, APs, etc.) of the network is assumed to be a way forward. This densification will require on the one hand the autonomous operation of the network elements, since the network administrators will not be able to monitor and configure manually all the network elements, and on the other hand advanced mechanisms for coordinating the operation of the network elements as well as the increased complexity.

1.2 Motivation and Challenges

The networks in the future, it is envisaged that they will be able to operate in an autonomous manner. This implies that the network elements will have the ability to monitor their environment, analyze the environment stimuli, plan their operation and execute their plan. This model has been proposed by IBM in the early 2000's and has been a huge innovation regarding the way that future networks shall be designed. Towards this direction, several schemes and mechanisms have been proposed by the research community to tackle the problems that arose.

Taking into consideration the explosion of the number of network devices and the exponential increase of the traffic volumes we notice that the design criteria for future systems [3] include aspects that haven't been considered in the past. Some of them are the following:

- Management cost,
- Complexity,
- Efficient utilisation of network resources,
- Energy saving.

Additionally, we should consider that the future network conditions are hard to predict. This implies that the networks should be able configure their operation according to the new network stimuli, so as to handle unpredicted problems, optimization opportunities, and network conditions. For facilitating a network element to operate in the future demanding networks, a network element shall have the ability to:

- Monitor its environment,
- Identify its state,
- Make decisions,
- Make projections,
- Execute,
- Learn based on the previous actions.

The previous functionalities suggest some of the key aspects of a self-aware system, which is a system able to observe the environment and deduce its state, within a volume of time and space, and make decisions and projections based on its current status, and the available alternative actions. The literature analysis (see Section 3) has highlighted an attempt of the schemes to mimic the human reasoning. However, due to their static definition of the environment, the schemes fail to meet requirement for human behavior approximation. Additionally, up to now both academia and industry attempt to handle the problem of environment modeling using rules and policies, combined with thresholds. The thresholds are either defined by experts or by user surveys. Thus, concluding, we observe that:

- A major gap of sophisticated solutions in the available proposals both in the literature and the industry solutions exists (i.e., mainly threshold based).

- The available solutions fail to mimic human situation perception and awareness, thus making the building of systems hard.
- Situation perception schemes that are not based on fixed or predefined views of the network operator are required; these solutions shall avoid using the subjective users' decisions.

The purpose of this thesis is to meet the previous requirements for situation perception by applying special schemes based on fuzzy logic and applying adaptation techniques on these schemes. The following subsection positions this dissertation with the previously described open research topics.

1.3 General Framework and Dissertation Contribution

The contributions of this dissertation move towards three directions, namely an architectural scheme for situation aware networking, a fuzzy logic situation perception scheme, and the corresponding learning mechanisms that enable this scheme to adapt to its environment.

Regarding the first contribution of this thesis, we have proposed a hierarchical architecture, which enables the network elements to operate in a self managed way. More specifically, we propose the introduction of two levels of hierarchy in the network management and control the Network Element Controllers (NEC) and the Network Domain Controllers (NDC). The first ones have local network view and may proceed in handling of local problems, whereas the later have bigger network view and may identify optimization opportunities or problems that are related to larger network compartments. These elements have to be able to characterize their environment and identify their operational status. Thus we have proposed a functional decomposition of the Situation Awareness functionality in such way that the network elements will be able to reason for their condition. The basic principles of the Situation Awareness functionality are related to the problem decomposition, to the network elements' cooperation, and to the targeting of networking problems. Additionally, the Situation Awareness model is built on the already available knowledge, on the one hand, assuming though the ability of the network elements to produce their local knowledge according to their decisions and the environment stimuli. In the proposed model, the Situation Awareness is being decomposed in three steps, the situation perception, the analysis of the environment (i.e., identification of the alternatives), and the projections. Finally, in terms of this thesis we attempted to provide an analysis of the state of the art, so as to identify the gaps in the proposed functional architecture. The analysis highlighted that the first step that of the situation perception has not been analyzed satisfactory (both by industry or academia).

The second contribution of this thesis is related to the afore-described problem. Thus, we propose the use of the fuzzy logic algorithmic tool for the situation perception of the network element. Fuzzy logic with its environment modeling mimics human logic, with the inference system based on policies. Additionally, it is a multi-variable mechanism, able to handle contradictive inputs and uncertainty cases. This scheme is suitable for identifying optimization goals and faults in the network. Fuzzy logic relies on the modeling of the environment (input and output models) and policy based inference engine. In terms of this dissertation, we have implemented the architectural framework that we have developed using the fuzzy logic situation perception in three cases. The proposed scheme exploits the sophisticated modeling of the environment and the policy based induced knowledge (i.e., by network experts) and thus manages to meet the requirement from the industry for following the human logic.

Finally, the third major contribution of this dissertation is the proposal of two adaptation schemes for the enhancement of the situation perception mechanisms. Initially a reference problem is being presented. Such description enables the application of the learning mechanisms to every problem that can be modeled according to the reference problem. Two learning schemes are being proposed, a supervised learning scheme and an unsupervised one (with two versions). The enhancement is related to the adaptation of the environment modeling of the fuzzy reasoners. This functionality enables the network elements to operate in new, unknown environments; the only requirement concerns the ability of the network elements to operate “well” with their generic configurations in the unknown contexts. The proposed learning scheme is inline to the architectural concept, also provided in terms of this thesis. The learning functionalities are being mapped to the hierarchical model so as to distribute the processing cost and relieve the less powerful NEC. Finally, an additional contribution of this dissertation is the comparative analysis of the learning schemes. The analysis attempts to provide some directions on how these schemes may be used in other similar problems on the one hand, and what are the drawbacks and benefits of each scheme.

1.4 Dissertation Structure

The dissertation is structured into eight chapters. Following this chapter, the structure is briefly presented below:

Section 2 introduces the main concepts of the dissertation. Specifically notions of autonomicity, autonomous networking, and self-managed networking are being introduced. Additionally, the activities of the standardization bodies towards autonomous networking are being presented. Afterwards, a brief description of the research activities of both academia and industry, as it is captured by their literature works or the project activities, is being presented. The chapter concludes by presenting the key requirements of the Self Managed networks.

Section 3 deals with a main concept of the autonomous networks that of Situation Awareness. Initially, the concepts of data, information and knowledge are being presented, as well as their relation for enabling the autonomous operation of the network. Afterwards, the functional architecture of Situation Aware networks, as it is captured by the research community, is being presented, as well as why such schemes fail to meet the requirements of the Self Managed networks. Thus, an innovative proposal, combining features of the literature and incorporating new ones, is being presented. Additionally, a literature survey of the mechanisms enabling situation aware networking is being presented, as well as their key characteristics. The comparative analysis of the previous schemes presents the gap in the literature, regarding environment modeling and the corresponding reasoning (i.e., Situation Perception) in situation awareness, which is mainly captured by rules and policies, and thresholds, thus failing to meet the human logic.

In **Section 4** the basic algorithm schemes used in this dissertation are being presented. More specifically, the key aspects of Fuzzy Logic are being presented, as well as the most important characteristics of data mining and learning schemes. Regarding the former, initially the fundamentals of fuzzy sets and its combination with rules are being described in details. Afterwards, the basic categories (i.e., supervised, unsupervised, reinforcement learning) of learning schemes are being described. Then, the most representative schemes of each category are being analysed so as on the one hand to provide the background for the following sections, and on the other to justify why the selected approach has been followed.

Section 5 presents the developed situation perception schemes, based on fuzzy logic. Initially, the concept of use of fuzzy logic in such problems is being presented. Afterwards, the use of fuzzy logic in three problems is being described. More specifically, the fuzzy reasoners are used in three problems:

- QoS degradation events' identification,
- Load events' identification,
- Cooperative power control.

In all three cases the situation perception schemes are being thoroughly discussed (i.e., description of the functional architecture, the fuzzy reasoners configuration, the evaluation/experimental setups). Additionally, the gains from the incorporation of the fuzzy logic situation perception schemes are being analysed and discussed.

In **Section 6** the drawbacks of the fuzzy logic based situation perception schemes are being presented, regarding their ability to operate in diverse environments. Thus, towards this direction, in terms of this dissertation two learning schemes are being proposed, a supervised one and an unsupervised one (with two versions). Both schemes are being evaluated in the case studies that have been analyzed in Section 5, so as to identify the corresponding benefits from the introduction of the adaptation mechanisms. Finally, a comparison between the learning schemes is drawn in the conclusion of this section, trying to highlight the key aspects of each proposed solution.

In **Section 7** the conclusions and the scientific contribution of the dissertation are summarized, by describing the challenges that have been addressed and the solutions that have been proposed. Additionally, a set of useful outcomes is being presented, regarding the tackling of similar problems in the future. More specifically, it is attempted to provide a description of the characteristics of the problems that could be solved using the same methodology. Finally, open issues and suggestions for future work are provided.

2. Autonomic Networking

In this section the basic principles of autonomic networking are being analysed. Initially the notions of autonomicity, and autonomous networking are being introduced combined with a state of the art analysis (in terms of standardization, and research activities). Then the section concludes with the analysis of the notion of self managed networking combined with the key requirements of the aforementioned networks.

2.1 Definitions

The definitions of the main concepts that are used in this chapter are summarized below:

- Network Management: captures all the operations used by the network to improve its performance. Furthermore, network management defines in an explicit manner policy rules for security, handling special customers, defining services, accounting, etc. Also, it includes monitoring functionalities regarding the traffic and the status of network equipment. The philosophy of network management is that it shall operate on a slow time scale and provide network elements with the information they need to react on faster time scales as the context dictates. At this point we should highlight the differentiation of the network management compared to signaling, since signaling mechanisms react to external causes on a very fast time scales and serve as the nervous response system of the network. On the other hand network management operations are triggered by the network administrator or control software detecting that some reallocation or expansion of resources is needed to serve the active contracts at the desired quality level [1].
- Autonomic Network Management: are the network management systems that are capable of self-governing and reducing the duties of the human operators who are not able to deal with increasingly complex situations. The systems should exhibit some level of intelligence so that their capability can improve over time, assuming more and more tasks that are initially allocated to skilled administrators [5].
- Self-managed network: is a network that has the capability to monitor, configure, adapt itself, based on the perceived network stimuli. The

operation of the network is dictated by a set of high level policies induced in the network by the network administrator [6].

2.2 Autonomic Networking Principles

Moving towards 5G networks, the estimation is that mobile and wireless data traffic will increase considerably. In [1] it is predicted that mobile data traffic will increase globally 13-fold between 2012 and 2017 and at the same time global IP traffic has increased more than fourfold in the past five years. Furthermore, the number of connected devices will reach the 100 billions [7]; each device will have its specific characteristics regarding its capabilities (CPU, memory, battery, etc.) as well as the traffic that it will induce to the network. Furthermore, contemporary networks are dynamic and complex, including new technologies, new services, and new demands from the users, new business cases [2]. The networks now days have to manage changes in both in the users' needs, and the network operational environment. Additionally, network management is expensive because network devices understand only low-level settings, and the diagnostics/monitoring is primitive. In such a complex environment, where billions of devices will (simultaneously) ask for resources from the network, networks' self-management aroused as a potential solution.

Self-management notion has been initially introduced by IBM in [8], where they first attempted to make an analogy to the human body, its nervous system, and investigate how the self-management capabilities are defined. Following this approach, they concluded in four basic functionalities for the Self-Managed systems, namely:

- Self-configuration is the ability to dynamically adapt to the environment changes (e.g., deployment of new components or the removal of existing ones, dramatic changes in the system characteristics etc.), by configuring the network's components using policies provided by the IT professional.
- Self-healing is the system's ability to discover, diagnose and react to disruptions. Self-healing enabled elements detect system malfunctions and initiate (policy-based) corrective actions without disrupting the IT environment. Such actions may be related to altering the state of one or more components or change of the configuration of a component etc. This

provides resiliency to the whole system, since typical operations are less likely to fail.

- Self-optimization is the system's ability to monitor and tune its resources automatically. The aim is to tune the elements to meet end-user or business needs. Potential tuning actions are related to resource allocation, control function adjustments etc. Self-optimization helps providing a high standard of service for both the system's end users and a business's customers.
- Self-protection is the ability of the system to anticipate, detect, and identify potential threats; afterwards, the network may proceed in protection against threats. Self-protecting components detect hostile behaviors (e.g., unauthorized access, denial of service attacks, etc.) and take corrective actions to make themselves less vulnerable, following the business goals and policies.

Following the previous analysis, IBM has built the Autonomic manager, which shall be able to Monitor (M), Analyze (A), Plan (P), and Execute (E), using a predefined or built Knowledge (K) (MAPE-K model - Figure 2-1). This model is being used more and more to communicate the architectural aspects of autonomic systems. Likewise it is a clear way to identify and classify much of the work that is being carried out in the field [9].

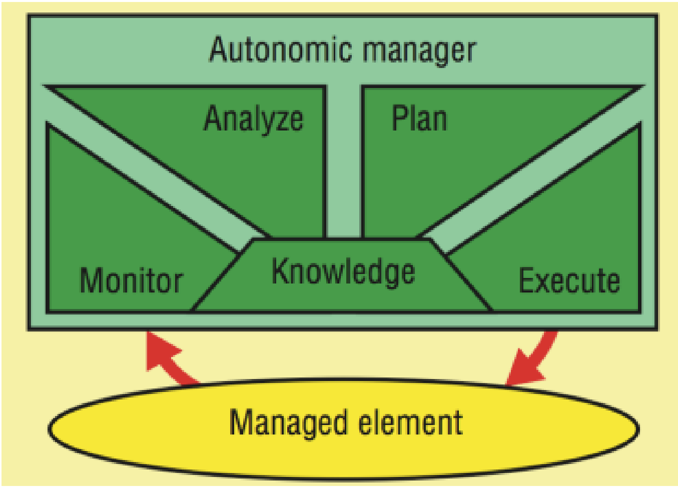


Figure 2-1: MAPE-K model as it is proposed by IBM [9]

The IBM analysis and proposal focused mainly in self-managed systems and autonomic computing. However, the same principles were quickly adopted in communication networks for various control and management plane tasks.

Consequently, self-managed and autonomous networks are able to monitor their behavior and environment, deduce about their status, and conclude in decisions; afterwards the decisions shall be executed. Several works have been based on the idea of autonomous networking and management since its initial proposal. The basic notions and ideas of autonomic networking have been summarized in the numerous surveys [10], [6], [11], [12], [13], [14], [15], [16]. In [10], Dobson et al. give the definition of autonomic networking, based on which an autonomic system collects information from a variety of sources (including traditional sources); such information is being analyzed to model the evolving situation. Thereinafter, the system makes decisions and acts (i.e., records actions, informs administrators, etc.) (Figure 2-2). Furthermore, Dobson highlights the differences between his autonomic communications and networking definition and the one for autonomous systems (IBM's definition):

- Autonomic communication is oriented towards distributed systems and services and the management is both at the infrastructure and the user,
- Autonomic computing focuses on application software and management of computing resources.

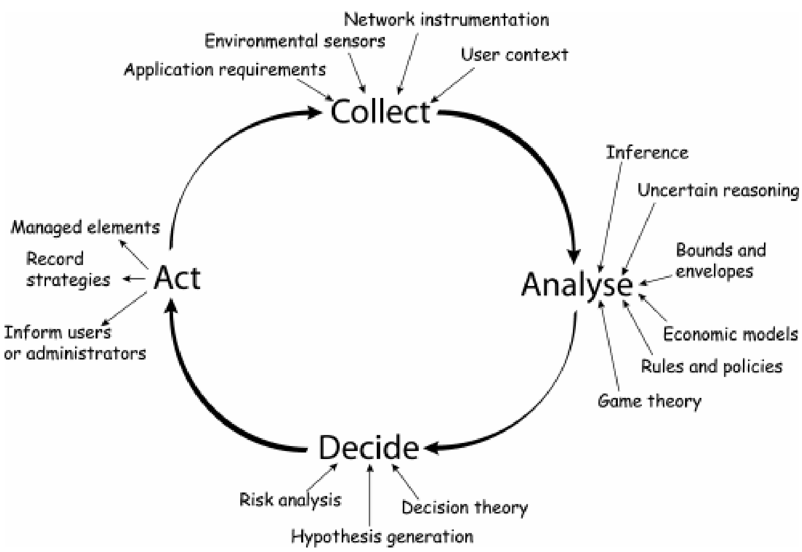


Figure 2-2: Dobson's model for autonomic communications and networking [10]

In [6] Kramer and Magee give another definition to the self-managed networks, closer to that by IBM, including also (apart from the Self-CHOP functionalities) the self-adaptation, the self-monitoring, and the self-tuning. According to the definition such systems need to configure and adapt their operation so as to satisfy their goals, properties, constraints, or to report that they cannot. Furthermore, the elements/systems shall reconfigure their operation in order to satisfy the changed specification of the environment, and report an exception.

In two more recent works, in [11], and [12] the authors focus on the principles of autonomous systems and include the recent advances in the autonomous networking area. In the former they try to introduce six key criteria for evaluating autonomic network management systems: a) the activity (i.e., ability to act in a re/pro-active manner), b) the adaptability (i.e., ability to learn and evolve), c) intelligence, d) awareness, e) memory strength, and f) autonomicity. These criteria materialize the key aspects of an autonomous network. In [13], [14], [15], and [16], they clearly introduce the notion of cognition in autonomous systems – meaning the ability of a network system (element as well) to learn and evolve.

2.3 Autonomic Networking Standardization Activities

Over the past decades, since the identification of the need for automation in network management, several schemes have been proposed in the literature. This motivated the International Telecommunications Union (ITU-T) to try to harmonize the research initiatives, by building the Telecommunications Management Network (TMN) architecture. TMN architecture is a reference model for a hierarchical telecommunications management approach trying to partition the functional areas of management into layers. TMN is defined in the M.3000 series ([17], [18], [19], [20], [21], [22], [23], [24]). In M.3010, the basic principles for a TMN are being described and the basic structure of a network management scheme according to its responsibilities is provided. TMN solutions target to [25]:

- Reduce time to market,
- Reduce cost,
- Support increased demands for higher quality,
- Incorporate legacy systems,
- Incorporate future-proof solutions,
- Conform to industry standards.

TMN architecture is based on a logical layers' model, which captures from corporate or enterprise goals to a network resource management and network elements' operation. Starting from the bottom level, such hierarchy incorporates network elements layer (NEL), element-management layer (EML), network-management layer (NML), service-management layer (SML), and business-management layer (BML). The management operations of lower layers are being linked to the corresponding operations of the higher layers. Table 2-1 presents the responsibility of each layer mentioned afore [26].

Table 2-1: TMN architecture based on logical layers

Layer	Responsibilities
Business-Management Layer (BML)	High level planning, budgeting, goal setting, business level agreements, etc.
Service-Management Layer (SML)	Managing aspects directly observed by the users of the telecommunication network. Builds upon management information that is provided by the NML without considering the internal structure of the network. Quality of Service management (delay, loss, etc.), accounting, addition and removal of users, etc. are considered responsibilities of the SML.
Network-Management Layer (NML)	Managing the functions related to the interaction between multiple pieces of equipment. At network management level the internal structure of the network elements is not visible; this implies that buffer space within routers, the temperature of switches etc. cannot be directly managed at this level.
Element-Management Layer (EML)	Handling the Operations Systems Functions (OSF); such layer included vendor specific management

	functions and hides these functions from the above layers.
Network Element (Layer) Functions	Providing TMN manageable information, which in other words could be described as interfacing the proprietary manageable information and the TMN infrastructure.

Following the ITU-T activities, both industry and academia have intensified their efforts towards networks' self-management. The previous developments of the researchers has required for further alignment, thus European Telecommunications Standards Institute (ETSI), in order to harmonize these activities, has formed a well-focused Special Working Group, seeking the establishment of a common understanding on what an autonomic behavior is and how an autonomic/self-managing network should be engineered. The Special Working Group is an Industry Specification Group (ISG) called ”Autonomic network engineering for the self-managing Future Internet" (AFI) [27], [28]. ETSI AFI defined the key management functions that shall be considered defined by the FACPS management framework (Fault, Configuration, Accounting, Performance and Security) [29] as well as the fundamental network functions such as routing, forwarding, monitoring, etc. To this end, ETSI – AFI has built an architectural Reference Model of a Generic Autonomic Network Architecture, the so called GANA framework, which is a framework for autonomic elements functions' definition, together with their relationships, and the corresponding self-management capabilities. According to [30], the mechanisms of the Self-Managed systems are not defined and are left for further study, though at least they shall incorporate the following properties:

- Automation,
- Awareness.,
- Adaptiveness,
- Stability,
- Scalability,

- Robustness,
- Security,
- Switchable,
- Federation.

The self-managed systems shall be able to detect, reconfigure and reregister its managed resources or managed devices (e.g., routers, UEs, etc.) and enable session continuity with no disruption. Autonomics should manage and control the mobility of an ambient system, in order to provide session continuity; local mobility decision should take into account the preferences, the capabilities, the objectives of the different players involved in session in order to identify a common decision able to provide session continuity [30]. In GANA, four levels of abstractions are being considered, for Managed Entities (MEs), Decision Making Elements (DMEs), and Control Loops [31]:

- Level 1: Protocol level solutions by which self-management is associated with the network protocol itself.
- Level 2: Abstracted Network Functions (e.g., routing, forwarding, mobility management) that abstract some protocols and mechanisms associated with a particular network function(s).
- Level 3: Node/device's overall functionality and behavior. In other words, a node or system as a whole is also considered as level of self-management functionality.
- Level 4: Network's overall functionality and behavior.

Regarding 3GPP networks, the network's self-management and autonomous operation is related to the introduction of the self configuring and self optimising mechanisms, also known as Self-Organizing Network (SON) functions [32]. A self-organizing system has a certain structure and functionality. The structure part captures the manner that the entities of the system interact (i.e., communicate) between each other, whereas the functionality part captures the purpose of the system. A system is self-organized if it is organized without any external or central dedicated control entity. In other words, the individual entities interact directly with each other in a distributed peer-to-peer fashion. Interaction between the entities is usually localized [33]. Extending the previous definition, a Self-

Organizing Network also incorporates the notion of network governance (including the planning, set up, and maintenance); thus the self-organizing network is able to set itself up and then manage the resources to enable the optimum performance to be achieved at all times [34]. Based on the problem that is being tackled, SON solutions may be divided into three categories: Self-Configuration, Self-Optimization and Self-Healing. The SON architecture may be a centralized, distributed or a hybrid solution [35]. SON is introduced by 3GPP in LTE releases 8, 9, and 10 thus making the SON functionalities fully 3GPP compliant. The standards provide network intelligence, automation and network management features in order to automate the configuration and optimization of wireless networks to adapt to varying radio channel conditions, thereby lowering costs, improving network performance and flexibility. Additional enhancements allow inter-radio access technology (I-RAT) operation, enhanced inter-cell interference coordination (e-ICIC), coverage and capacity optimization (CCO), energy efficiency and minimization of operational expenses through minimization of drive tests [36].

Also motivated by the IBM proposals, IETF has tried to give a harmonized approach in the autonomics area. As shown in Figure 2-3, traditional functionalities such as monitoring, reporting, and elements configurations assume the network administrator intervention, whereas autonomic networks require the network administrator's intervention for inducing the policies and targets to the network.

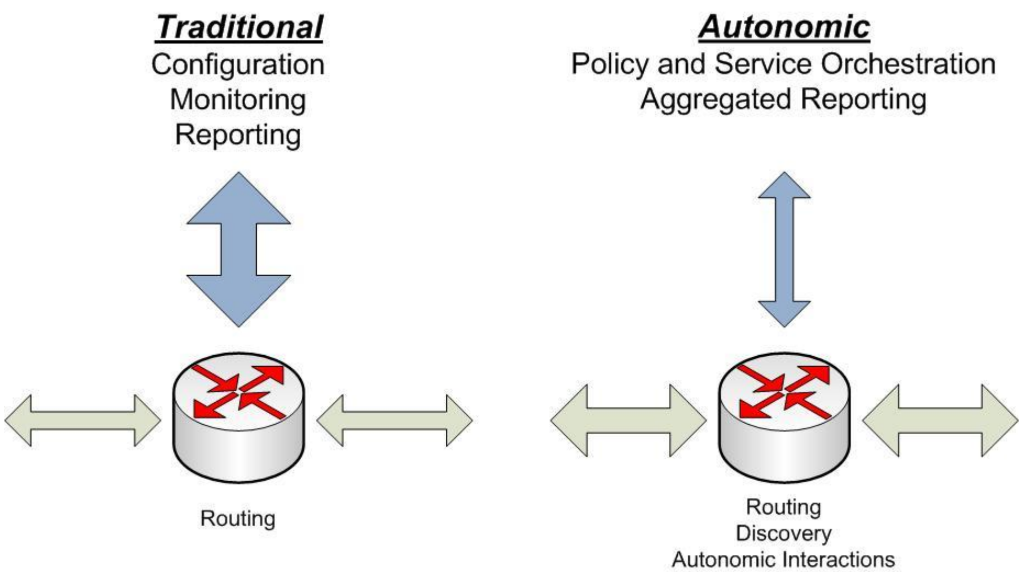


Figure 2-3: IETF autonomic model

Thus, IETF has recently introduced a hierarchy, which includes the following levels of autonomicity [37]:

- Autonomic: Self-managing is a self-managed network (Self-CHOP). It assumes though allowing high-level guidance by a central entity, through intent.
- Intent: Is an abstract, high-level policy used to operate the network autonomically; it does not contain configuration or information for a specific node.
- Autonomic Domain: Is a collection of autonomic nodes that instantiate the same intent.
- Autonomic Function: Is a function which requires no configuration, and may derive all required information either through self-knowledge, discovery, or through intent.
- Autonomic Node: Is a node (e.g., router, switch, etc.), which employs (exclusively) autonomic functions. It may operate on any layer of the networking stack.
- Autonomic Network: A network containing (exclusively fully) autonomic nodes.

Figure 2-4 visualizes the reference model of an autonomic node. In such scheme, what shall be standardized are the intents, the autonomic service agents (i.e., autonomic functions concerning Network Knowledge), and the interfaces of the feedback loops, as well as the message exchanges [37]. This will facilitate a harmonized approach of introducing new services, without enforcing to the vendors specific approaches.

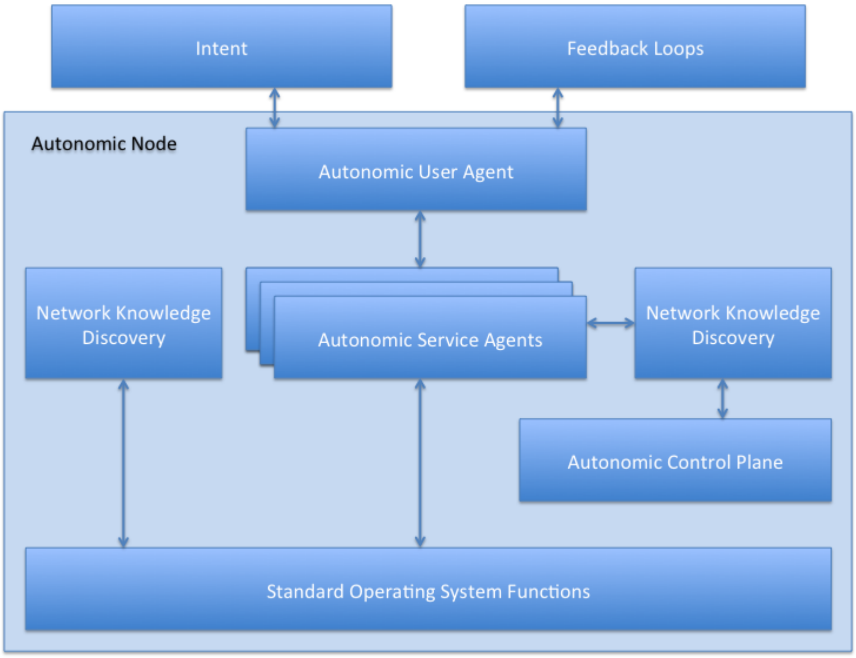


Figure 2-4: IETF levels of autonomicity

Finally, similar harmonization approaches have been followed by IEEE, which has sponsored the Dynamic Spectrum Access Networks (DySPAN – formerly known as Standards Coordinating Committee 41 (SCC41), and IEEE P1900 Standards Committee) for providing a coordinated way to handle radio and spectrum management. Several working groups (1900.1–1900.7) have been formed, each one tackling different future network aspects. 1900.4 Working Group focused on the building blocks comprising network resource managers, device resource managers and the information to be exchanged between the building blocks, for enabling coordinated network-device distributed decision making which will aid in the optimization of radio resource usage, including spectrum access control, in heterogeneous wireless access networks [38]. P1900.4 suggests a logical channel for the communication between the network and the terminals using the Reconfiguration Management Entities (RMEs); such entities on the network side are called Network Reconfiguration Manager (NRM), and on the terminal side is the Terminal Reconfiguration Manager (TRM) [39]. P1900.4 activities, in 2011 published the 1900.4a-2011 standard, which provides the architectural building blocks enabling network-device distributed decision making for optimized radio resource usage in heterogeneous networks [40]. P1900.5 defines a vendor-independent set of policy-based control architectures and corresponding policy language requirements for managing the functionality and behavior of dynamic spectrum access networks. Up to now the definition of the policy language and the methodology for the spectrum consumption are ongoing; future research will link these two activities with the architecture, by defining the detailed interfaces between policy architecture components [41].

2.4 Autonomic Networking Research Initiatives

The initial interest of individual researchers has been followed by coordinated attempts. Thus, numerous international consortia have been formed and attempted to draw their lines towards self-managed networks. Table 2-2 provides a non exhaustive list of the European Union funded projects that have attempted to provide a formulated description of an autonomic and self-managed network. The list is complemented by a brief summary of the key characteristics of each project.

Table 2-2: Brief description of the European Research projects for autonomic and self-managed network

Architectural Approach	Description
4D [42]	It is a clean slate architecture that introduces four planes, namely, decision, dissemination, discovery, and data, focusing on core network. It focuses on IP networks.
4WARD [43], [44]	It is a clean slate architecture that introduces In-Network Management (INM) functions that are located close to the management services, in most of the cases co-located on the same nodes. It focuses on various network environments (wireless, wired).
ACCORDANCE [45]	It is an evolution of 3GPP architectural solutions, where the focus is on various aspects of wireless communications (CoMP, wireless/wireline convergence, aging, etc.). The project focuses on centralized SON.
AMBIENT Networks [46]	Introduced the Network Composition which enables allows for networks to dynamically and automatically interconnect for the purpose of gaining access to

	and/or controlling the resources (and services) of the networks participating in the composition.
ANA [47]	It is a novel approach that can incorporate clean-slate or legacy solutions. The intension of the project to address the self-* features (self-configuration, self-optimization, self-monitoring, self-management, self-repair, and self-protection) of autonomic networking. ANA introduces an autonomic network meta-architecture that enables flexible, dynamic, and fully autonomous formation of network nodes as well as whole networks according to the working, economical and social needs of the users. It focuses on various network environments (wireless, wired).
ARTIST4G [48]	They follow the 3GPP architecture, but complement it with a new interface for conveying user plane inputs to the network for Coordinated Multipoint (CoMP - transmission and reception). The focus of this project is on the RAN part of the network.
AutoI [49]	It is a clean slate approach that introduces an architectural model consisting of a number of distributed management systems running within the network. For the operation of such distributed network five abstractions and distributed systems are defined, namely, the Virtualization, the Management, the Knowledge, the Service Enablers, and the Orchestration Planes. It focuses on various network environments (wireless, wired).
BuNGee [50]	They focus on heterogeneous architectural solutions (joint design of access &backhaul networks) by applying their solution in 3GPP (LTE-A) and IEEE (802.16) networks. Within the project they propose to design a data and control plane protocol suite that facilitates autonomous operation by means of a

	complete self-organising networking paradigm.
CASCADAS [51]	It is a clean slate approach that introduces Autonomic Communication Element concept, which is an abstracted component model. The Autonomic Communication Element is used for situated and autonomic communication entities, at all levels of granularity. It focuses on various network environments (wireless, wired).
CONMan [52]	It is a clean slate approach that introduces protocol module abstractions (concepts, properties, capabilities) for enabling manageability of future protocols in a “complexity-obvious way”. Additionally, it allows dynamic protocol stack composition on the fly. It focuses on various network environments (wireless, wired).
CONSERN [53], [54]	It is a revolutionary approach trying to introduce the newly concept of self-growing and re-purposing. Architecturally they propose the introduction of a cognitive engine in the nodes that enables the autonomic operation of the node and its cooperation with the policy and coordination controllers. The proposed scheme is a generic one, targeting sensors and access network elements (WiFi APs, eNBs, etc.).
E3 [55]	It is an evolutionary approach that introduces Cognitive Management of heterogeneous networks wireless access part, exploiting local and global pilot channels, the so-called Cognitive Pilot Channel (CPC). It focuses on heterogeneous wireless access networks environments.
EARTH [56]	It is an evolutionary approach of 3GPP networks that focuses on energy gains. They propose the separation of signaling and data, and move towards

	four directions, namely, context aware resource management independence of the architecture from the wireless technology, sophisticated signaling, and Network planning.
FAME [57], [58]	FAME project focuses on forming federations and their management for creating end-to-end communication services. It provides the holistic view, including the properties, and principles that need to be fulfilled by any system or architecture for managing and maintaining such federations.
FOCALE [59]	FOCALE proposes an incremental approach that can incorporate both clean-slate and legacy solutions. FOCALE introduces the architecture for network entities to self-govern their behavior within the constraints of business goals that the network as a whole seeks to achieve. It focuses on various network environments (wireless, wired).
METIS 2020 [60]	Moving towards 5G, METIS 2020 proposes significant evolution of 4G solutions. The innovative part is related to the 5G objectives that may be related to different objectives (e.g., huge number of devices, high data rate, ultra reliable communications, etc.); thus they propose to have a flexible architecture, enabling the network to meet its objectives according to the current situation.
NESTOR [61]	It is an evolutionary approach that introduces an architecture for the automation of configuration by using policy scripts that access and manipulate respective network elements via a resource directory server (RDS). It combines several techniques from object modeling, constraint systems, active databases, and distributed systems. It focuses on IP network infrastructures.

SACRA [62], [63]	It is an evolutionary approach focusing on 3GPP networks. The project ended up in an architecture compliant to that of LTE-A, incorporating self-management functionalities in the 3GPP network entities. In the project they have split the functionalities in RAN functionalities (i.e., cognition, learning, knowledge), Core network (i.e., governance), and user equipment functionalities (sensing, cognition, etc.). The project mainly focused on access network functionalities.
SelfNET [64]	It is an evolutionary approach that introduces Cognitive network element (Network Element Cognitive Managers - NECMs) and Cognitive domain managers (Network Domain Cognitive Managers - NDCMs) for future Internet elements self- management in a semi-distributed and cooperative manner. If focuses on various network environments (wireless, wired).
SerWorks [65]	The proposed clean slate approach consists of three frameworks: the Service Framework in the upper layer of the architecture, the Interaction Framework in the middle layer, and the Networking Framework in the lowest layer. It initially focused on Wireless Sensor Network (WSN) solutions but it is extended for more generic service infrastructure in wireless and also wired domain.
SOCRATES [66]	It is a 3GPP evolutionary approach, introducing Self-* functionalities for Self- Organizing networks (SON). The project addressed RAN issues.
UNIVERSELF [67]	The key outcome is the Unified Management Framework for providing a functional specification, the interfaces, and the supporting core mechanisms. The project has concluded in 3-level UMF nodes

	(i.e., controlling a node, a network domain, an overall network).
WINNER I/II/plus [68], [69], [70], [71]	It is a 3GPP evolutionary approach focusing on flexible protocol architecture. The main point of interest is the radio access side; M2M, CoMP, D2D have been considered.

All these architectures in general converge for the need of some key functionalities. More specifically, context-awareness, knowledge plane, policy-based decision making, and the network operator governance/coordination are main (and common) components of the proposed architectures [72], [73].

In the following section, we present a reference architecture scheme, which is considered the basis for autonomic network management. The scheme that will be analyzed incorporates all the key functionalities of afore-described architectures.

2.5 Requirements for Self-Managed Networks

The previous analysis enables the derivation of the key characteristics of a self managed network, as it is envisaged by both the inventors of the autonomic systems and networking idea, as well as the researchers that followed their principles. Even though it is not clearly highlighted, research initiatives tend to converge to a set of requirements and capabilities [2]. According to Jonhsson et al., self-managed networks shall be:

- Re-applied to all parts of the system. Such requirement captures the need for continuous control and supervision of all the system parts from a network management perspective. This will enable the network to offer specific QoS to the users, given the fact that the network will be aware of the offered QoS and will know whether the signed agreements (Service Level Agreements - SLAs) are covered. Furthermore, the control loops shall always be correlated for giving to the network administrator a full view of the network.
- Go across the system boundaries. The (Self-Managed) system's (or network's) shall go across the system's boundaries for handling inter-domain operations. Up to now, the self management has been considered

objective for one single domain and network; however, contemporary networks assume applying self-management capabilities also for interconnected domains. This implies a need for interconnecting and composing the resources of the all the involved domains and networks. Such interconnecting must be able to support end-to-end service and resource abstraction for all the involved networks.

- Enforced and monitored for operating under constraints. This requirement correlates the constraints under which a self-managed network shall operate with the enforced actions and monitoring operations (and requirements). This also implies that the constraints posed by the network administrator shall pose meaningful constraints. Additionally, the self-managed network shall be compliant to the Design by Contract (DbC) ([74]) concept, which enables the posing of pre- and post-conditions.
- Self-mutatable. Such aspects are related to the ability of a self-managed system to tackle new challenges that may arise in its lifecycle. The new challenges could be linked to new types of applications, new technologies being deployed such as new hardware for network nodes or new radio/wireless technologies, etc. Potential lack of this characteristic will make the self-managed system inefficient and unable to meet its objectives. Concluding, the self-mutable functionality re-designs the capabilities of the self-managed system/network in run-time operation (at least this is how it is observed by an external reference point).

3. Situation Perception

Section 3 deals with situation perception concept. Starting from the situation awareness definition and its functional decomposition, this chapter proposes a novel hierarchical architecture for situation aware networks. Then, the gaps of the literature analysis and the industry state of the art are being identified, so as to set the motivation for the forthcoming sections of the dissertation. These gaps highlight the need for novel situation perception mechanisms that will be able to meet the requirements from the key players in the area.

3.1 Definitions

The definitions in this section concern the information fusion among the several functionalities when we are moving towards situation awareness. The definitions are related to the Data, Information, Knowledge and Wisdom model, as it is described in the literature [75].

- Data: is the product of observations in a structured way. It should be note that raw data are in general useless and is no benefit until they are processed into a usable form to become information [76].
- Information: Is the formulated data in such way so as to answer to questions [76].
- Knowledge: is the transformation of information into instructions and appropriate structures so as to make control of a system possible [75]. In other words, knowledge is the appropriate collection of information, such that its intent is to be useful. It should be highlighted that it is a deterministic process [77], [78].
- Wisdom: means an ability to see the long-term consequences of any act and evaluate them relative to the ideal of total control (even that it is not used in this chapter – it is included for completeness) [75], [78].
- Data-Information-Knowledge-Wisdom hierarchy (DIKW) (Figure 3-1): is the hierarchy of the above notions; various different kinds of data, information, knowledge, and wisdom exist. The DIKW hierarchical and pyramidal model captures mainly organizational aspects of the inputs representation [75] [76], [78].

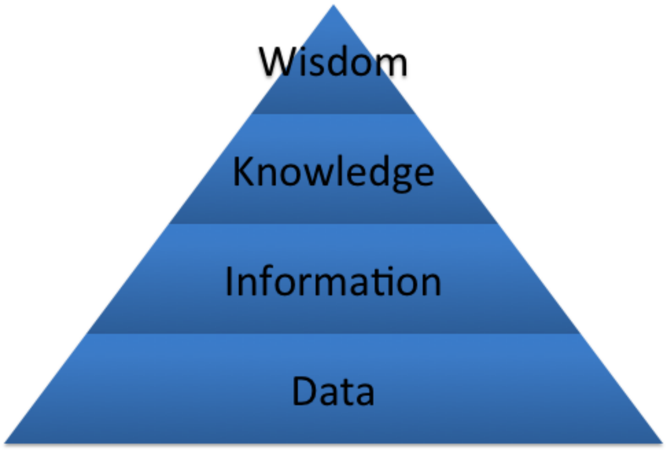


Figure 3-1: The Data-Information-Knowledge-Wisdom hierarchy as a pyramid [75]

3.2 Towards Situation Awareness

Section 2 has introduced and defined the key requirements of a self-managed system. When decomposing the previous requirements to technical requirements the next step is to define the main functionalities of a self-managed system. A self-manageable system must have [77]:

- An internal representation of its experiences as it perceives entities, events and situations in the world;
- An internal model that captures its knowledge, and;
- A proper mechanism for computing values and priorities that enables it "to decide what it wishes to do/perform".

The previous functionalities enable high autonomy of network elements in order to allow distributed management, fast decisions, and continuous local optimization. The actual functional decomposition results to the following phases/processes of a generic (autonomic) cycle for describing the autonomous network elements (Figure 3-2):

- Monitoring process involves gathering of information about the environment and the internal state of an autonomous element.
- Decision-making process includes the reconfiguration and adaptation decisions by exploiting an already available knowledge base; also captures knowledge building by exploiting the environment stimuli.
- Execution process involves (self-) reconfiguration, software-component replacement or re-organization and optimization actions.

As captured by the Figure 3-2 the Monitoring process is directly linked to the Execution phase. On the other hand, the Execution and the Monitoring are linked (directly or indirectly) so as to deduce whether the previous decision was effective or not. Such “loop” enables the update of the knowledge model used for the Decision making process.

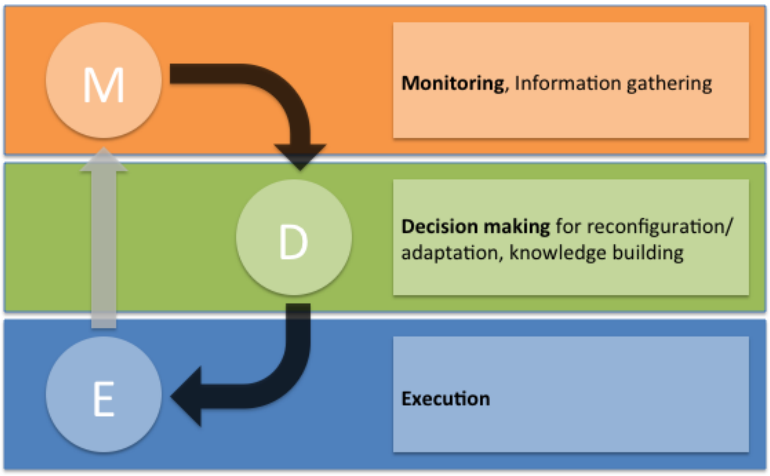


Figure 3-2: The Monitor Decide Execute (MDE) cycle [79]

The autonomous network elements may be (Figure 3-3):

- network elements such as router, base station, mobile device, etc.,
- network managers, or,
- software elements that lie at the service layer.

Autonomous network elements have a process for monitoring and perceiving internal and environmental conditions, and then planning, deciding and adapting (self-reconfiguring) on these conditions. Such an element is able to learn from these adaptations (reconfigurations) and use them for future decision making, while taking into account end-to-end goals.

In general such network management requires a distributed/decentralized management approach over a hierarchical distribution of management and (re)configuration decisions to:

a) (autonomic) network elements,
b) to network domain types, and
c) up to the service provider realm.

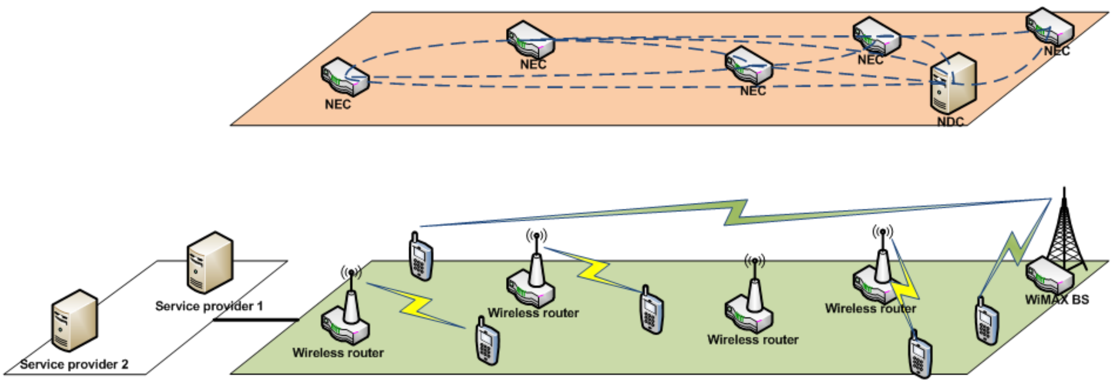


Figure 3-3: Autonomous network elements in a Heterogeneous Network

The MDE cycle may have various levels of realization according to the type of the device and the hierarchical level that it is placed. Each reasoning entity (i.e., entity that realizes the MDE cycle) has the ability to expand and consider the results of other neighboring entities in a collaborative manner, thus leading to an incremental development of local (element level) and global knowledge (network wide). Figure 3-4 captures the main elements of the MDE cycle, attempting to break down the monitoring operation and depict the information/data exchange among the phases.

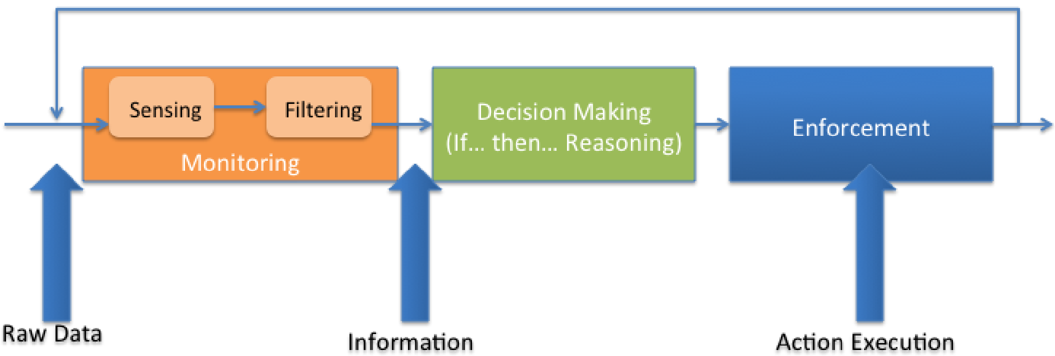


Figure 3-4: Elaborated MDE cycle

Moving towards a self-managed network the “simple” functions need to be further enhanced by the “Situation Awareness” functionality. According to Endslay in [80], a situation aware network (i.e., a network whose the network elements understand their environment and make projections for their near future) shall be able to identify its state, proceed in decisions and perform the corresponding actions, by exploiting the induced and built knowledge, the induced long term goals, the available interfaces, and system’s capacities (Figure 3-5). Situation awareness functionality is “the perception of elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future”. In other words, the relevant model introduces time factors, “space” (i.e., environment) assessments, and interpretation & prediction aspects, in the near future; the definition is more generic (initially related to dynamic systems – air traffic control, power plants, etc.). The key levels of Situation Awareness are [81]:

- Perception of Elements in Current Situation (Level 1) deals with perception of the status, attributes/characteristics, and dynamics of all related elements in the surrounding environment. It is the most essential level of Situation Awareness, since it translates the monitoring, and performs simple recognition, directing towards awareness of situational elements (such as events, and environmental factors) and their current states (i.e., conditions, modes, etc.).
- Comprehension of Current Situation (Level 2) involves a kind of “synthesis” of disjointed Level 1 situation awareness inputs. It applies sophisticated methods, such as pattern recognition, interpretation, and evaluation. The purpose is to identify the impact of the current state with regards to its impact to predefined goals and objectives.
- Projection of Future Status (Level 3) captures the ability to predict the future environment conditions, status, as well as future actions. It is realized through knowledge of the status and dynamics of the elements and comprehension of the situation (i.e., the previous Levels 1 and 2 SA), and then extrapolating this information “forward in time” to conclude the manner it will have effect(s) on future states of the operational environment.

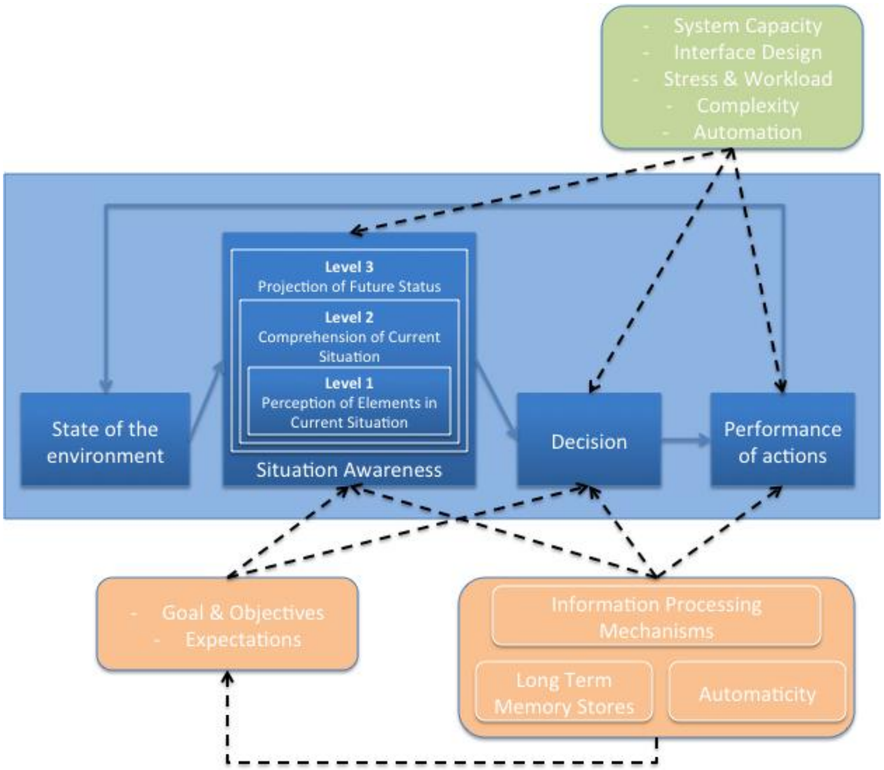


Figure 3-5: Endslay's model for Self-Awareness [80]

As mentioned afore, Endslay provided the generic Situation Awareness model. In [82] Smirnov et al. give a telecommunications oriented definition for situation awareness; the definition focuses on autonomic networking. According to this (contextual) approach, Situation Awareness is a prerequisite for making appropriate decisions in networks. The Situation Awareness scheme is based on knowledge (described as ontologies, models, etc.), which is constantly enhanced by exploiting the evolution of the environment stimuli. Smirnov identifies three "new" (more targeted) levels of Situation Awareness and describes them as follows:

- Level 1 (Perception): The first level deals with perceiving critical factors in the "environment" of concern.
- Level 2 (Inference): The second level aims in the appropriate understanding of "what those factors mean" for the specific decision maker's goals.
- Level 3 (Prediction): The third level aims to "predict" what will happen in the near future.

The above can be considered as a "start" for interpretation of situation awareness in autonomic networking, clearly depicting a "three stage" process.

In [83] Springer et al. propose a different approach for achieving Situation Awareness; they decompose complex situations into sub-situations, which can be handled independently with respect to sensing and reasoning. Each sub-situation represents a certain aspect of the overall situation and has to be combined with other sub-situations at a certain level of a hierarchical reasoning process. Below that point, a sub-situation can be handled separately. The afore-described approach enables a hierarchical situation analysis and awareness. To reflect all necessary steps for deriving the overall situation from sensed in- formation, three layers are considered (Figure 3-6):

- Sensing layer, which aims at combining information from devices measuring/sensing different types of inputs; this layer exploits raw data, linked to location (topological) information. This implies the division of the "world" to "areas of interest" enabling the situation awareness mechanism to use data for specific sub-situations. At these areas of interest different types of sensor devices may be placed and logically grouped.
- Feature extraction layer, which focuses on the combination of inputs from several sensing layers either from the same location or different ones. The outcome of this procedure is logic facts, which are forwarded then to the reasoning steps.
- Reasoning layer, which hierarchical situation deduction and handles potentially complex or contradictory inputs from the previous layers.

The whole process of situation decomposition and the identification of the components of the different layers of the conceptual architecture are determined based on an iterative process. Starting with a small set of sub-situations, the developer can test the system and extend it stepwise to create a more-and-more complex system; the understanding of a complex situation grows during testing and practical trials.

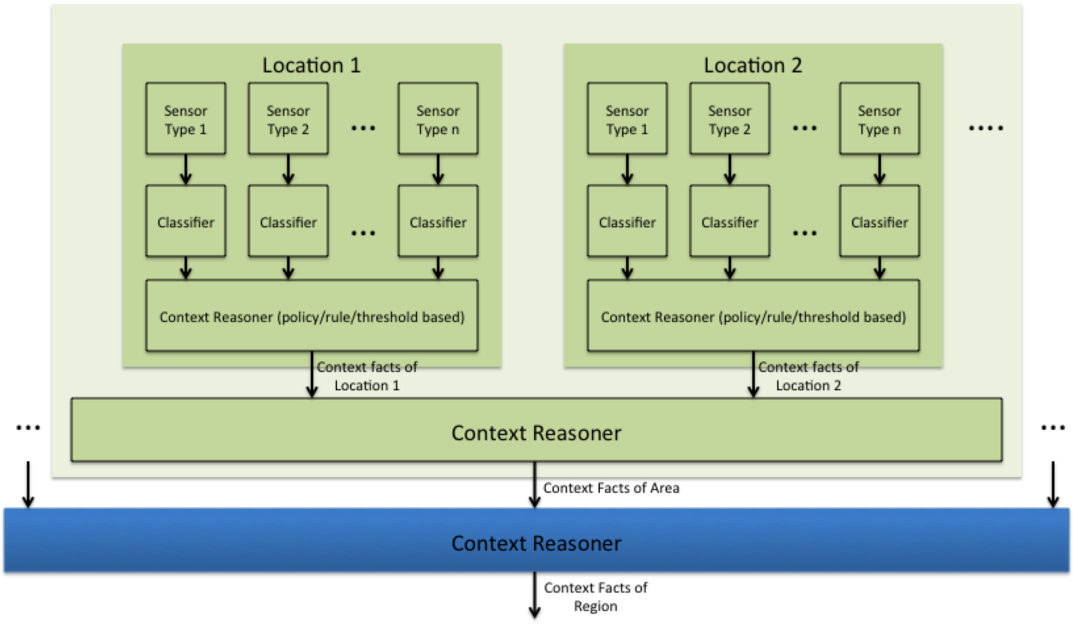


Figure 3-6: Abstraction process for situation detection based on sensor data [83].

3.3 Situation Awareness Model

The previously described Situation Awareness schemes have both benefits and drawbacks. For example, the Endslay's approach is broad and decomposes situation awareness in three levels. However, the presented scheme is too generic and rough and does not fit to specific problems as is; this implies that it requires extended modifications for being incorporated in telecommunication environments. On the other hand Smirnov's scheme, is a slightly modified approach of that of Endslay so as to be suitable for networking problems, though it does not manage to capture the local view and the problem decomposition aspects. Such aspects are being captured by the third situation awareness approach, which clearly defines the notions of problem and sub-problem, area, location, sensing element type and tries to combine these for context reasoning.

Thus a combination of the previous models is required; Figure 3-7 presents the considered Situation Awareness scheme. The three levels of the Situation Awareness models of Endslay and Smirnov, are also used, instantiated of course for contemporary autonomous networks.

- Monitoring: This includes separate activities like sensing and correlating/filtering; it refers to data corresponding to the information process of interpretation.
- Situation Awareness:

  - Level 1: Is the proper perception of the operational status of the system or the network element where information is primarily interpreted (i.e., to an elementary knowledge interpretation). For instance, information like "85% of a link is utilized" may lead to the concluding characterization that "the load level of a WiFi Access Point is congested above the tolerable threshold" where any such characterization is predetermined by system designers.
  - Level 2: This implicates an exhaustive evaluation of the surrounding environment, including identifying alternatives, potential solutions etc.
  - Level 3: This includes "projections", i.e., prediction on what would happen in the future and/or what conditions need to be met, to satisfactorily proceed to the decision-making.
- Decision-making: Is the decision making scheme that considers the available options, and picks the most suitable one for the network element or system. A decision making scheme may range from a simple reasoned using "if… then… " rules to sophisticated reasoning schemes as game theoretic, genetic algorithms, etc. schemes.
- Knowledge Base:
  - Interpretation Library of operational states: This refers to thresholds and input levels that enable the Level 1 of situation awareness.
  - Situation Trigger: This refers to thresholds or states that lead to Level 2 of SA.
  - Procedure for Assessment of the Environment: This captures the history of states and their evaluation.
  - Deduction Completion: It is an abstraction to indicate finalisation of all "checks" for the environment's assessment. In case of predictions (i.e., needed Level 3 SA) this additionally refers to what needs to further happen, to deduce a situation and proceed to the decision-making stage.
  - Intelligence for decision-making: Captures the parameters of the decision-making schemes (i.e., the parameters are related to the reasoning scheme).
- Update of operational States: The reasoning scheme may lead to new states or new relations among the already defined states. Thus the

knowledge base needs to be updated.

- Self-Awareness: Captures all monitored inputs and deductions, as well as the continuous activity with the corresponding time and location information of a network element. The history information is also part of the situation awareness [79], [84], [85], [86].

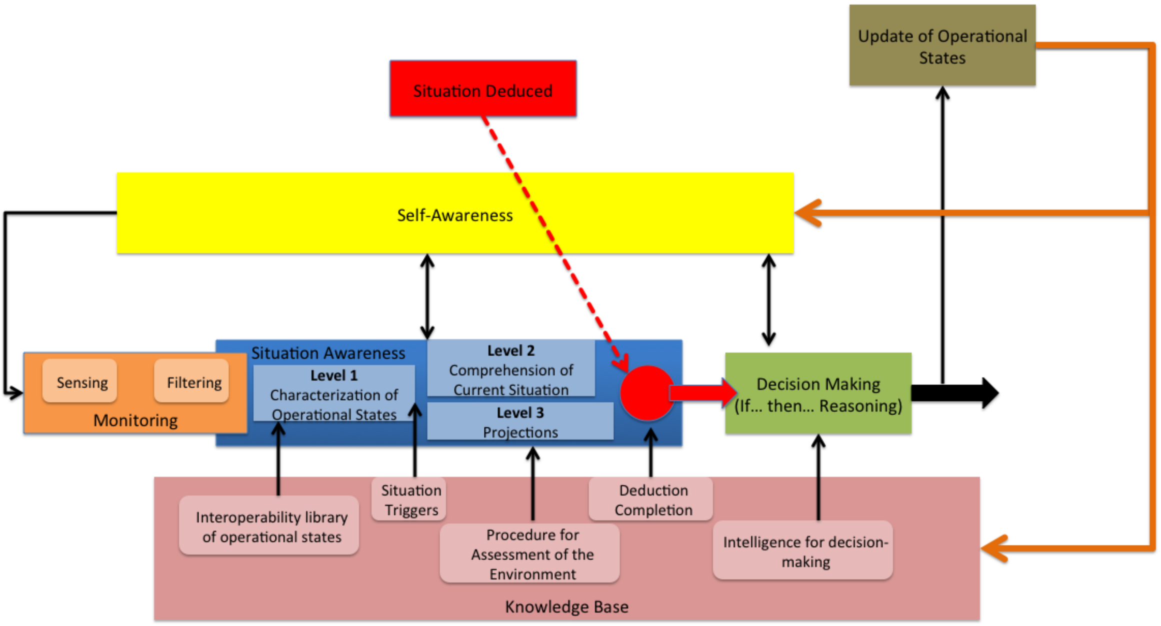


Figure 3-7: Proposed Situation Awarness Model

For achieving enhanced Situation and Self Awareness, the overall scheme shall be developed following a both distributed and hierarchical paradigm (Figure 3-8) [87]. The Network Elements shall have an instantiation of the MDE cycle for monitoring deciding and executing configuration actions; from this point in this document we define the Network Element Controller (NEC) as the instantiation of the MDE cycle and the more sophisticated Situation Awareness functionality in the Network Elements. Furthermore, the Network Elements in case more complex problems arise, they shall cooperate and for more focused and enhanced problem handling. However, the Network Elements may not be able to handle several problems, either locally or cooperatively; thus the intervention of a network element that has a greater network view may be required. This network element is the Network Domain Controller (NDC) identifies optimization opportunities and solves problems that require a greater view of network status, the cooperation between neighboring domains, or even the resolution of conflicts in metrics/parameters. The NDC does not disturb the distributed and hierarchical nature of the proposed architecture, as it is not a centralized physical entity.

Various NDCs may exist in a specific network domain, hosted even at the available network devices that already implement NECs. The criteria for the selection of the number of NDC may be (though not limited to these criteria):

- The network technology (or technologies),
- The network element capabilities,
- The System requirements,
- Spatial and geographical features.

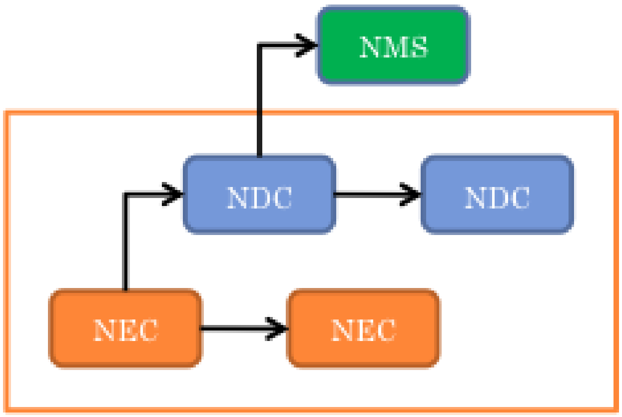


Figure 3-8: Hierarchical Model for Situation and Self - Awareness

The decomposition of network management into responsibility areas provides flexibility to the network for handling problems, faults and optimization opportunities either locally or globally. Of course, such decomposition is coupled with the introduction of autonomic functionalities at all layers and enables decisions and configurations at shorter time-scales, in reflex reaction manner. Hints and requests/recommendations are exchanged among the layers, in order to indicate a new situation or an action for execution. The automated and dynamic incorporation of various layers requirements (e.g., SLAs) into the management aspects provides also novel features to network management capabilities. Moreover, the resolution of conflicting requests will be an issue of situation awareness and elements' domain policy prioritisation.

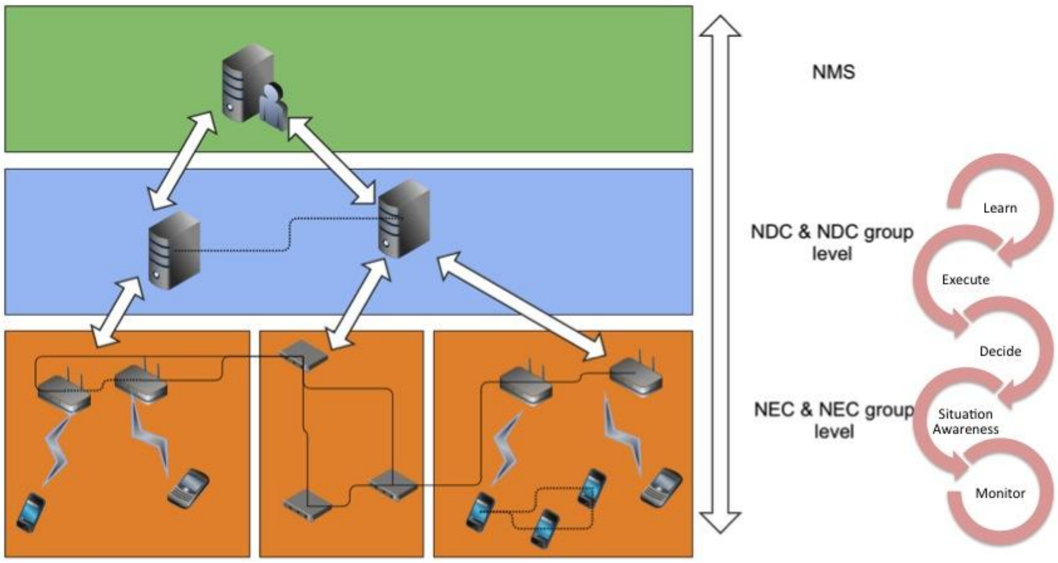


Figure 3-9: Hierarchical network view for situation awareness

3.4 State of the Art analysis on Situation Awareness

There is limited work done in relation to generic situation awareness schemes. Several solutions available in the literature are problem-dependent focusing on social networking (some of them are the following [88], [89], [90], [91]). However, such solutions are too focused and are dealing with the (user) behavior identification.

Regarding "traditional" situation awareness, as Endslay, Smirnov, etc. define it, only a few mechanisms are proposed in the literature. In [92], Hossein Parvar et al. propose a policy and threshold-based scheme for "cooperative" situation awareness. According to this scheme each problem is decomposed to several sub-problems and when "linking" all these aspects the decision maker is able to proceed in identification of the current state; the authors call this process "multi-resolutional representation". This implies that the decision takes place in one point that has greater network view. In [93] the authors also propose a problem decomposition approach; their proposal is policy and threshold-based similar to the previous one. The main difference, compared to the first one, is that they further formulate the problem using ontologies and they end up in the primitive relations among the observations. According to them, this enables the decision maker to have principle (global) knowledge, independent from the (network) context. It should also be highlighted that the decision maker requires global view so as to be able to proceed in proper situation awareness decision-making.

In [94] the authors propose a scheme that uses a mixture of Ontologies and Bayesian networks, which combine First-Order Logic (ontologies) with Bayesian Networks for representing and reasoning about uncertainty in complex, knowledge-rich domains [95]. This scheme does not address levels 1 and 2 of situation awareness (i.e., perception and comprehension) but captures level 3 (i.e., projections) by predicting the likelihood of an event to happen. The authors give generic examples (e.g., accidents in streets, etc.) for presentation simplicity but they mention that this approach is applicable in networking problems as well. In [96] an ontology-based situation awareness scheme for making predictions is proposed; the solution requires global view for achieving good predictions. The scheme considers temporal and spatial relations among the entries of the ontology and builds adaptive knowledge for several domains. The solution is rather generic, though it is validated in social networking problems. In [97] they focus on the problem decomposition, which enables the decision maker to proceed in projections based on the current inputs. The scheme is a generic, rule-based (i.e., case based reasoning – CBR [98]) approach which, by using the current measurement of the overall system's view, it matches the inputs to a set of predefined ones states; this enables the decision maker to have, with a certain probability, a view for the foreseen state(s). Finally, in [99], the authors propose a model-based solution, which is based on fuzzy knowledge schemes; the scheme is traditional rule and threshold-based solution. The fuzziness is related to things that the decision maker may forget on the one hand, and the reflex decisions that he may make on the other. The decision making scheme identifies the current situation of the decision maker (level 1 of situation awareness) and does not perform projections.

In [100] the authors propose a scheme for Situation Awareness using a special version of Neural Networks, the Self-Organizing Map (SOM). The scheme provides the tools for visualization and decision-making. Additionally, the reasoning scheme is adaptive based on the training set it will receive during the training period. The SOMs are tested/evaluated in security situation awareness problems in Mobile Ad – Hoc Networks (MANETs) and perform well in identifying suspicious behaviors. It should be highlighted that the networking problem (i.e., MANETs) used for validation underlines the ability of the scheme to operate in decentralized environments.

In [101] a cooperative agent based scheme is proposed. The scheme is called Cooperative Agent-based QoS Framework (CAQF) and incorporates several functionalities (among them rooting, scheduling, resource management, etc.). The scheme applies information fusion for proper event identification (situation awareness); this is performed via data mining (i.e., clustering), which captures anomalies in the measurements.

In [102], and [103] the authors proceed in Situation Awareness using the D'BRAIN (Dynamic Bayesian Reasoning & Advanced Intelligent Network). The schemes are based on problem decomposition to sub-problems; linked to probabilities for moving from one state to another. The objective of this situation awareness approach is to make projections for future conditions. Global view is assumed, which in conjunction to the fact that the information fusion schemes are not being described, implies a centralized decision making scheme. The second scheme is an enhanced version of the D'BRAIN for enhanced situation awareness by reducing the number of the complete Bayesian Tree; additionally, the enhanced D'BRAIN is adaptive and builds knowledge according to the environment stimuli.

Finally, a completely different approach is presented in [104]. The presented Situation Awareness scheme assumes distributed monitoring and reasoning in a hierarchical manner. The proposal is based on problem decomposition where the different layers of reasoning deduce on different problems; the authors call this approach as “individual situation awareness of team members”. According to this scheme of critical importance are:

- The confidence for each measurement,
- The time it took place,
- For how long an inference is valid, given that a situation is identified.

For identifying the similarity of the measurements/inputs to previously known states, Fast Furrier Transform (FFT) is applied; this formulation is generic and may be applied for several problem types.

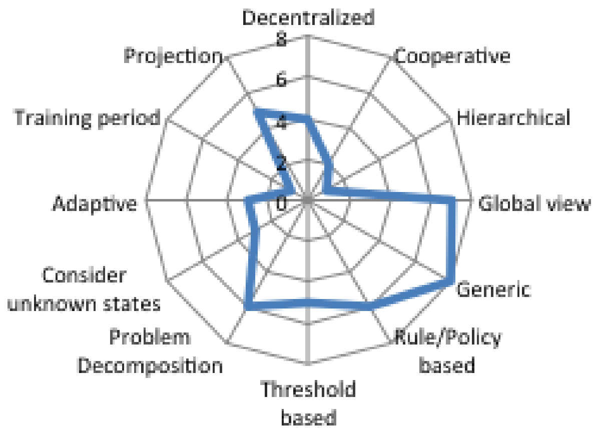


Figure 3-10: Schematic representation of the state of the art analysis

Figure 3-10 and Table 3-1 summarize the findings of the previous analysis which leads to a set of very useful outcomes. In the literature, the two most popular schemes for situation awareness are the rule and threshold-based ones and the probabilistic ones. The former are mainly versions of model and ontology schemes whereas the latter are Bayesian networks and trees related ones. Both schemes however have the key deficiency that they require a good description of the environment's states which may lead to too many dimensions in the decision maker (also known as the curse of dimensionality), which hampers the decisions. Furthermore, the Bayesian networks require the linking of the states with probabilities. Regarding the ontologies, it is a rather computational and time demanding scheme that for large ontology models is even harder to build and exploit. Concerning the other aspects of the afore analysis, in general the schemes require a global view, which in most of the cases implies a centralized scheme for reasoning, even though it is not clearly stated. Only a few schemes are decentralized and only two assume cooperation between the monitoring and reasoning points. Finally, only one scheme assumes a hierarchical architecture among the reasoning entities.

Regarding the problem formulation, usually the problem is decomposed in sub-problems so as to mimic the human reasoning. However, most of the schemes do not consider unknown states/situations; more specifically, all but one have a predefined set of situations and two map potentially unidentified inputs to previously defined ones (via similarity or probabilistic metrics). Considering the adaptability of each scheme, only a few have learning/knowledge building capabilities, with one of them requiring training period (neural network). Finally, half of the investigated schemes are related to situation perception (state/problem identification - situation awareness level 1) whereas the rest are related to projections (situation awareness level 3).

Table 3-1: Comparative table of the literature proposals

Approach	Centralized	Decentralized	Cooperative	Hierarchical	Global view	Generic/Specific	Rule/Policy based	Threshold based	Problem Decomposition	Consider unknown states	Additional scheme	Adaptive	Training period	Projection
[92]		X	X			Generic	X	X	X					
[93]					X	Generic	X	X	X		Ontology			
[95]					X	Problem oriented	X	X		X	Bayesian			X
[96]					X	Generic	X	X			Ontology	X		X
[97]					X	Generic	X		X		Case based			X
[99]					X	Generic	X	X			Model based			
[100]		X				Problem oriented					SOM	X	X	
[101]		X	X			Generic				X	DM			

[102]					X	Generic			X		Bayesian			X
[103]					X	Generic			X		Bayesian	X		X
[104]		X		X		Problem oriented			X	X	FFT			

3.5 Situation Percpetion

The previous sub-sections have provided a detailed analysis of the Situation Awareness Models available in the literature, as well as the enhanced model proposed in terms of this thesis. Furthermore, a state of the art analysis has been presented regarding situation awareness schemes literature proposals; this analysis has highlighted where the researchers have up to now focused their efforts.

An observation of prime importance is that research has moved towards the Situation Awareness in two ways, namely, the development of overall situation awareness (i.e., applying all three steps of the SA module – situation perception, comprehension, projections), or only of projection schemes. Regarding the first part of Situation Awareness, the Situation Perception, there are several attempts to define it in the literature; some of them may be found below:

- *"Perception is our sensory experience of the world around us and involves both the recognition of environmental stimuli and action in response to these stimuli."* [105]
- *Perception is the primary basis of human intelligence and experience* [106]. *Forming high-level abstractions from machine and user inputs is a key enabler for developing situation-aware applications that can intelligently respond to changes in the real world.* [107]

In our model we define as situation perception as:

> *"[…] all correlations that take place in order to analyze data received by monitoring points and thus identify problems and select appropriate configuration actions."* [108]

In the literature solutions, these correlations are performed using rules and policies, combined with thresholds.

Up to now, industry has handled this problem following a similar approach. Thresholds are being defined either statically or dynamically, but the crisp nature of the thresholds remains. Cisco has proposed static and dynamic threshold definitions for networking solutions regarding load threshold values (dynamic scheme), call drop rate (for QoS provision) (static values definition), security events identification (static values definition), etc. [109], [110]. Similarly, Ericsson in its "Traffic and Market report" of 2012 ([111]) has defined static thresholds for load events, for QoS levels identification to the users, etc. For these classifications, Ericsson has used users' surveys; such approach has the drawback that the measurements are not objective (the subjective views of the users are incorporated). Additionally, in recent research solutions, Ericsson researchers have used static thresholds for modeling the high load levels of HSDPA eNodeBs [112].

The previous analysis underlines that the Situation Perception in telecommunications is based on crisp values, which does not follow the abovementioned definitions that are close to human like knowledge and perception. Thus, concluding, we observe that:

- A major gap of sophisticated solutions in the available proposals both in the literature and the industry solutions exists (i.e., mainly threshold based).
- The available solutions fail to mimic human situation perception and awareness, thus making the building of systems hard.
- Situation perception schemes that are not based on fixed or predefined views of the network operator are required; these solutions shall avoid using the subjective users' decisions.

The purpose of this thesis is to meet the previous requirements for situation perception. More specifically, Section 4 provides the background knowledge for the proposed situation perception schemes; Section 5 presents the Situation Perception mechanisms based on fuzzy sets and fuzzy logic so as to avoid crisp values, and Section 6 presents the adaptive Situation Perception algorithms for capturing the evolution of the networking environment.

4. Background

Section 4 provides the background of this dissertation. More specifically, the Fuzzy Logic decision making scheme is being presented as well as its principles related to fuzzy sets. Additionally, the basic categories (i.e., supervised, unsupervised, reinforcement learning) of learning schemes are being described briefly. Then, the most representative schemes of each category are being analysed so as on the one hand to provide the background for the following sections, and on the other to justify why the selected approach has been followed.

4.1 Fuzzy Logic

Fuzzy logic decision-making is scheme based on fuzzy sets, which have been introduced by Lotfi Zadeh in [113] and are the basis for the paradigm shift from crisp logic to fuzzy logic. Fuzzy Logic is a multivalued logic that allows intermediate values to be defined between conventional evaluations like true/false, yes/no, high/low, etc. making it an ideal tool for situation perception schemes. Notions like fast, rather fast/slow, problematic, urgent, etc. can be formulated mathematically and processed by computers, in order to apply a more human-like way of thinking in the programming of computers [114]. Compared to other reasoning methods tries to link fuzzy values and enable the decision maker to proceed in decisions.

The key benefits of fuzzy logic are linked to its simplicity and flexibility as well as its ability to handle imprecise and incomplete data. This section summarizes the key aspects of fuzzy sets and fuzzy logic. Furthermore, simple examples are provided for depicting its applicability.

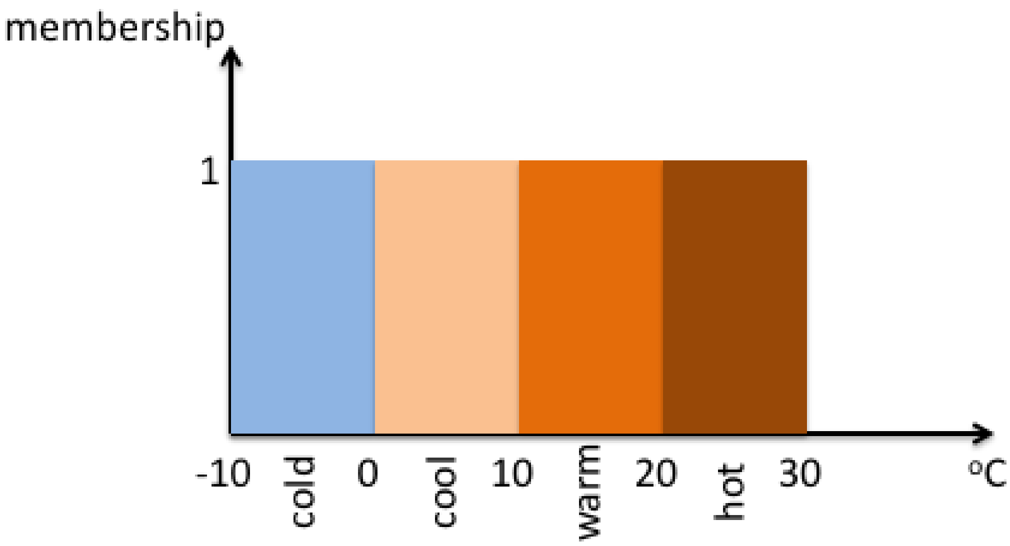
4.1.1 Fuzzy Sets

Based on its initial proposal in the literature (by Lotfi Zadeh), fuzzy sets are relying on a paradigm shift from crisp logic and crisp sets to fuzzy sets [115]. A simple example could be the splitting of days to weekdays and weekend days. The days could be split in weekdays (i.e., Monday, Tuesday, Wednesday, Thursday, and Friday) and weekend days (i.e., Saturday, and Sunday). However, this splitting is not fully correct, because Friday is “when weekend starts”, so Friday is mainly weekday, though it should be considered as partly a weekend day as well.

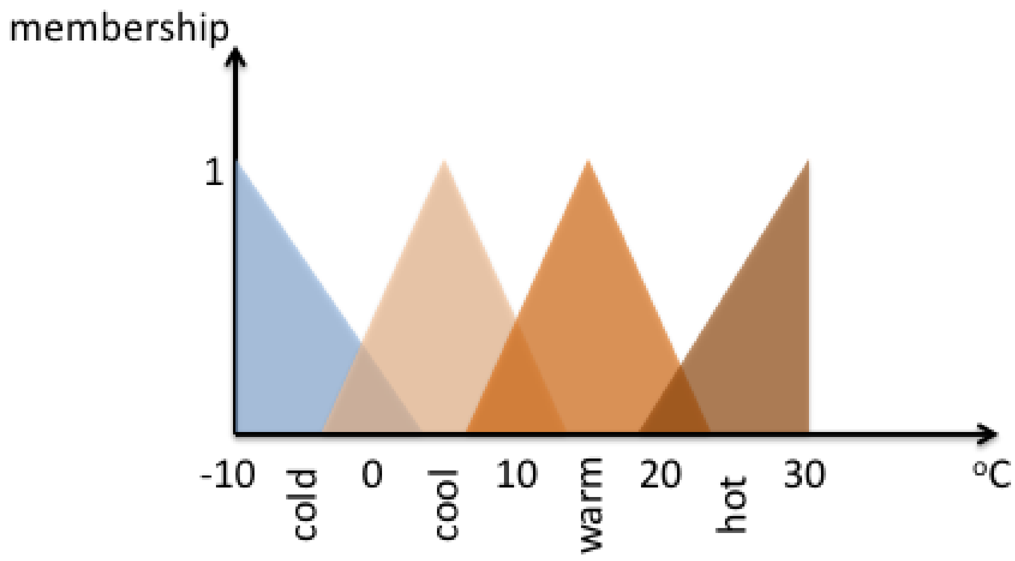
Another example could be the splitting of the real variable temperature in cold and warm ones. A simple scheme with crisp values could be the one depicted in Figure 4-1 (a) where the temperatures are characterized as cold, cool, warm, and hot ones. Though, as it may be observed:

- The sets are mutually exclusive which means that a value (i.e., temperature) may not belong to more than one set
- Moving from one set to another is arbitrarily set, and depends to a increase (or decrease) of the input value for a degree (or a fraction of a degree).

It is obvious that this approach is not accurate; overlaps among the sets should exist for describing the phenomenon sufficiently (Figure 4-1 (b)). This modeling enables the description of cases that you have a specific temperature you may belong in more than one states. This implies that a temperature is somehow cold and cool, or cool and warm, or warm and hot. This approach is closer to the human logic [116].



(a)



(b)

Figure 4-1: (a) Crisp classification of the temperatures (b) Classification of the temperatures using Fuzzy Sets

The previous examples (days of the week and temperature classification) are closer to the human logic and are not fully mathematically formulated. This is presented in the forthcoming example [117]. Consider the real interval [1, 10] as universe of discourse; also consider that the "x is true between 3 and 5". This mathematically is represented as:

$$\mu_{3-5}: [0, 10] \rightarrow [0, 1] \quad (4.1)$$

Where:

$$\mu_{3-5}(x) = \begin{cases} 1, & x \in (3, 5) \\ 0, & otherwise \end{cases} \quad (4.2)$$

This definition characterizes the linking of a real interval to crisp values.

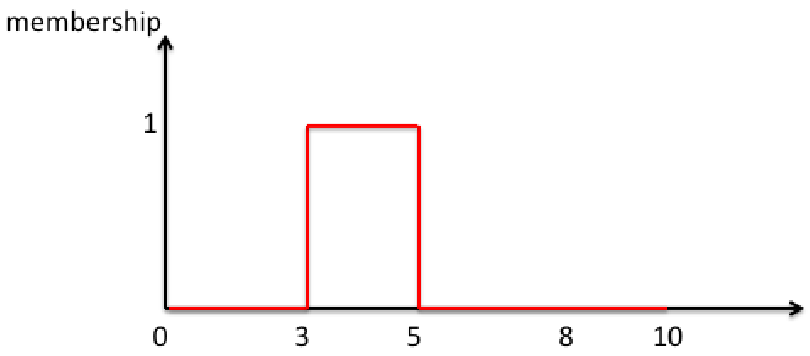
If we consider a new statement that "x is true near to 4" the formulation is different. More specifically, assuming an ε, a very small positive number, then (4-ε) is "near to 4". At this point, considering several ε with increasing values, we shall also consider degrees of "nearness to 4", until reaching a value where the x value is that far from 4 so as "not being true at all"; let us assume that the values for "not being true at all" are the vales below 3 and above 5. In this experiment we observe that a symmetric behaviour occurs. The mathematical formulation of the previously described approach is the following one:

$$\mu_{near\ to\ 4}: [0, 10] \rightarrow [0, 1] \quad (4.3)$$

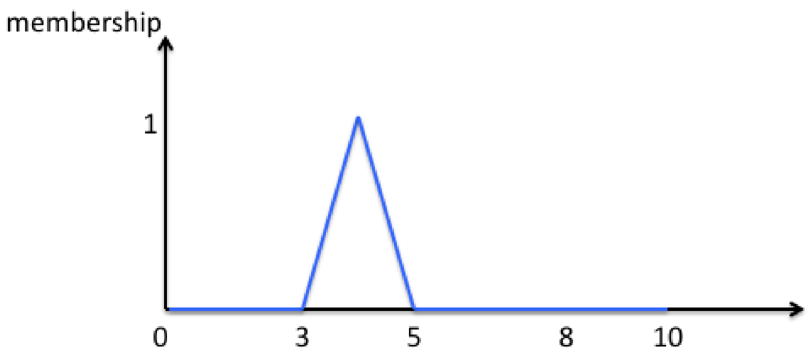
Where:

$$\mu_{near\ to\ 4}(x) = \begin{cases} 1 - |4 - x|, & x \epsilon (3, 5) \\ 0, & otherwise \end{cases} \quad (4.4)$$

Figure 4-2 presents the previously described analysis; near to 4 is a fuzzy set. It should be highlighted that every element belonging to a fuzzy set has a degree of membership, which is expressed with a numerical value in [0, 1] (i.e., 0 means “not true at all”, whereas 1 means “definitely true”). For this reason the function μ is called “membership function” of the corresponding fuzzy set.



(a)



(b)

Figure 4-2: (a) Traditional set with bi-valued logic, (b) “near to 4” fuzzy set defintion.

The variables of the fuzzy systems are called linguistic variables. Each linguistic variable has:

- A name, which shall be consistent and representative to the real variable that it represents,
- A definition domain, which captures the environment and the universe that the linguistic variable applies to. The definition domain incorporates a set of linguistic terms that represent the values of a definition domain,

- A set of values, for describing the intensity of a state, and,
- The corresponding interpretation for each set of values.

A simple example for explaining the previous notions could be ground humidity levels. The linguistic variable is the "ground humidity" and is defined in the environment of 0% humidity to 100% humidity. The environment is characterized by five (considered) linguistic terms labeled as {low, below average, average, above average, high}. These terms are chosen arbitrarily for characterizing adequately the environment; if the environment modeler would assume that more, or less linguistic terms are required he could add or remove terms. It worth mentioning that for a specific value of the environment (e.g., 35% humidity), the fuzzy state is a linear combination of the two linguistic values (i.e., x of below average and y above average).

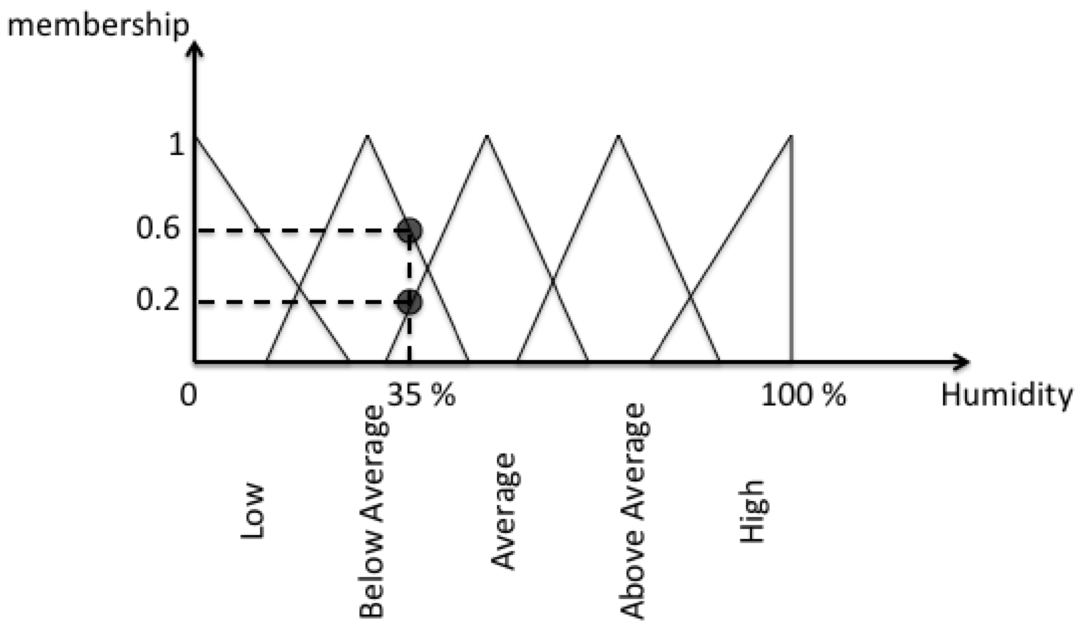
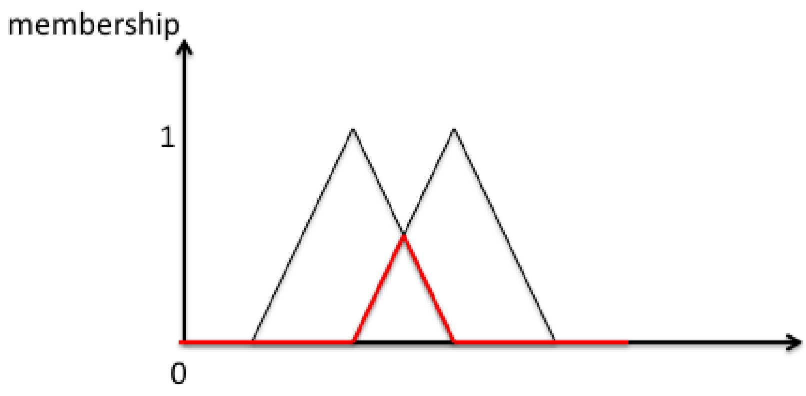


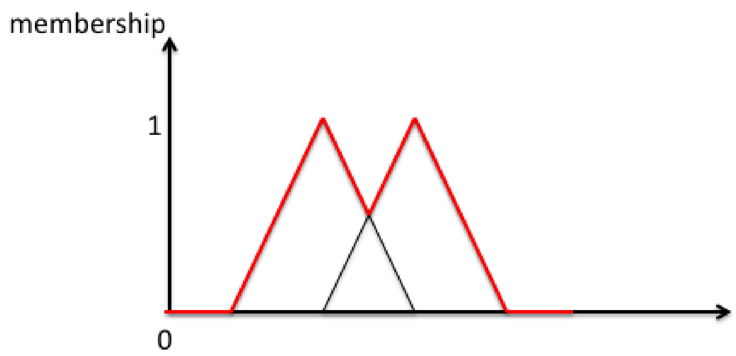
Figure 4-3: Ground humidity classification using fuzzy sets.

In [113], apart from the fuzzy sets definition, Zadeh has defined the operations among the sets. The most used operations are:

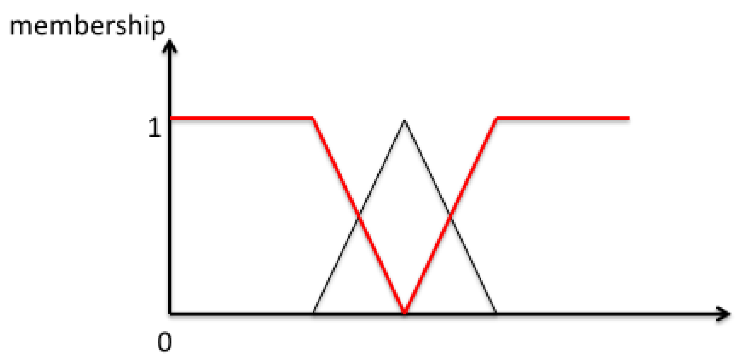
- the minimum operator for the intersection (Figure 4-4 (a)),
- the maximum operator for the union of two fuzzy sets (Figure 4-4 (b)),
- the complement of a fuzzy set A with μ_A, which is defined as 1 - μ_A (Figure 4-4 (c)).



(a)



(b)



(c)

Figure 4-4: (a) minimum operation application in two fuzzy sets, (b) maximum operation application in two fuzzy sets, (c) completement operation application in a fuzzy set.

4.1.2 Fuzzy Inference Systems

The Fuzzy Sets theory enables the environment modeling and the simplified representation of the system. This will facilitate the decision maker to proceed in the corresponding identification of the most proper actions. By combining the fuzzy sets with formalisms (such as rules, policies, etc.) we build fuzzy models. A system incorporating the fuzzy sets for the definition of the environment and the corresponding rules for decision-making suggests a fuzzy inference system [118]. Fuzzy Inference Systems (FIS) have been used extensively in the past for automatic control, data classification, decision analysis, and telecommunication systems. In terms of this thesis the focus in on telecommunication systems where fuzzy logic has been used for decision making [119], [120], [121], event identification, and event classification [122], [123], environment modeling [124], etc. (the list of the application fields, nor the literature work is exhaustive).

The simplest fuzzy models are using rules with “IF… THEN…” structure:

IF <condition 1> and <condition 2> and … and <condition n>

THEN <conclusion>

Where,

- condition i is a statement of type “x_i is $L_{i,j}$”,
- x_i is the i-th value of the a linguistic variable,
- $L_{i,j}$ is a fuzzy set, which captures the j-th linguistic term of the i-th linguistic variable,
- conclusion is also a fuzzy set which characterizes the output behavior (with a set of linguistic terms).

A Fuzzy Inference System consists of three parts, namely the fuzzyfier, the inference engine, and the defuzzifier (Figure 4-5).

- The fuzzifier transforms the crisp values to fuzzy ones. In other words, the fuzzifier takes the inputs for every linguistic variable and determines the degree to which they belong to each of the appropriate fuzzy sets via the corresponding (input) membership functions. The input is a numerical value limited to the universe of discourse of the input variable (it could be a real value, integer, natural, etc.) and the output is a fuzzy degree of membership (always the interval between 0 and 1).
- The inference system undertakes three roles:
    - To apply the fuzzy operators: once the inputs have been fuzzified the fuzzy operators apply to the rules. In case the antecedent of a given rule has more than one part, the fuzzy operator is applied to obtain one number that represents the result (single number) of the antecedent for that rule for applying it to the output function.

- To apply the implication method: when the fuzzy operators are applied, the inputs shall be linked to the output(s). This means that the result of the application of a fuzzy operator will be mapped to the output fuzzy set. This implication takes place for all the available rules.
- To aggregate all inputs: in the aggregation phase all the single outputs of every rule are being aggregated to a single fuzzy set. Several aggregation schemes have been proposed and applied, namely the maximum, the probabilistic or, the sum, etc. with the latter being the most common one.

- The defuzzifier undertakes the defuzzification process, which is the aggregation of the outcomes of all the rules and the production of a single number; the single number is a crisp value. The use of fuzziness is beneficial for matching the human logic and for the rule evaluation but it is not facilitates the reasoning, thus the transformation from fuzzy notions to crisp values is required in the final stages of the decision-making. Several defuzzification methods have been proposed:
  - The centroid, which returns the center of gravity of all the aggregated inputs using the:

$$u_{COG} = \frac{\int u_i \mu_F(u_i) du}{\int \mu_F(u_i) du} \quad (4.5)$$

  - The middle of maximum, which returns the average of the maximum value of the output sets,
  - The largest of maximum, which returns the largest value of the maximum of the output sets,
  - The smallest of maximum, which returns the smallest value of the maximum of the output sets.

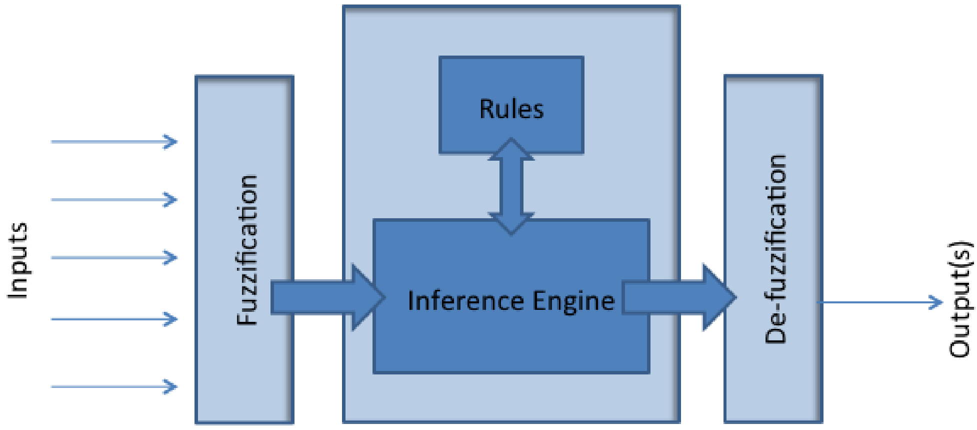


Figure 4-5: Schematic representation of the parts of a Fuzzy Inference System

Assume a two input (input 1 and input 2) and one output Fuzzy Inference System, where the inputs are modeled with three triangular symmetric membership functions (low, medium, high) and the output is modeled with also three Gaussian membership functions (low, medium, high). The inputs and the outputs are linked by a set of three rules as shown in Table 4-1 and the applied aggregation scheme is the summation of the output of every single rule, whereas the defuzzification method is the centroid one. Figure 4-6 presents the previously mentioned steps for making a decision with a Fuzzy Inference System.

Table 4-1: Rules of the Fuzzy Inference Engine

Input 1	Input 2	Output
low	high	low
medium	low	medium
high	high	high
medium	high	medium

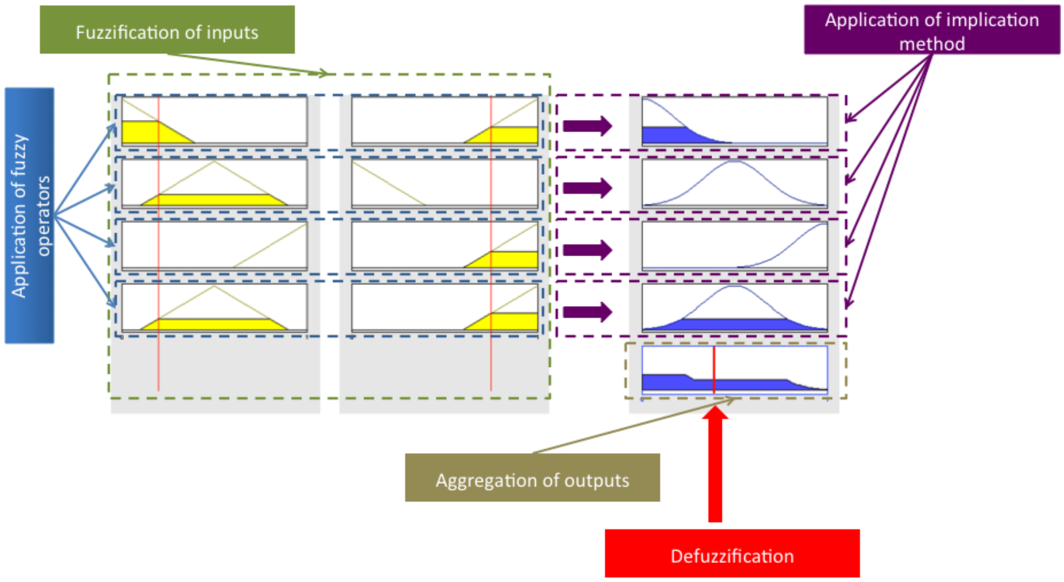


Figure 4-6: Sequential representation of the three steps process of a Fuzzy Inference System

The previous analysis concerns the most commonly used Fuzzy Inference Systems, the Mamdani ones. Ebrahim Mamdani has proposed this approach for introducing control operations in Fuzzy Sets in [125]. Mamdani has used as background the Zadeh's definitions of fuzzy sets for approaching human logic in complex problems. In 1985 Takagi – Sugeno ([126], [127]) proposed a slightly different approach that will not been analyzed in terms of this section, because it is not used in this thesis. The main differences are [128]:

- The outputs are singleton spikes instead of fuzzy sets,
- The implication method is multiplication,
- The aggregation method is addition of all singletons.

The Mamdani's proposal is more intuitive and close to human logic whereas the Sugeno fits better in control problems, is more computational efficient, and has better mathematical formulation [128].

4.1.3 Analysis of Fuzzy Logic

The previous sections present the key aspects of fuzzy logic, as well as its basic types and simple examples. This paragraph intends to describe the benefits and the drawbacks of fuzzy logic as a reasoning scheme.

In general fuzzy logic is a conceptual approach; the corresponding modeling resembles human logic and is easy to understand. Additionally, it provides a mathematical formulation easy to understand and model, which also resembles the human logic. With this simplified mathematical formulation, fuzzy logic may be used to model nonlinear functions of arbitrary complexity for linking the inputs with the outputs of a system.

The fuzzy set universe modeling makes the fuzzy inference systems flexible. In other words, the FIS may be easily modified/adapted according to the environment observations. Also the system is tolerant to imprecise inputs, which are likely to appear in control applications (i.e., in general control applications incorporate cheap and unreliable parts).

The inference engine is based on a set of rules that link the environment model with decisions and deductions. Suppose that the universe is built adequately, experts may complement it with the corresponding rules. This implies that the Fuzzy Inference Systems do not require training periods and may be incorporated in the systems easily. Also, Fuzzy Logic may be blended with other techniques as well. Examples exist in the literature for combining fuzzy logic with neural networks ([129]), learning schemes ([130], [131]), ontologies ([132], [133]), etc. Finally, Fuzzy logic in general is a simple scheme with small computing requirements [134]. Additionally, there are several techniques in the literature that enable the building of even more efficient fuzzy logic schemes by reducing the number of the rules etc. [135].

On the other hand, it should be highlighted that fuzzy logic is linked to appropriate system/environment modeling. This implies that the system complexity is removed from the system inference engine (simple “if… then…” rules that may easily be built by experts) but is transferred to the system modeling (need for “correct” fuzzy sets modeling). A second drawback is related to the number of rules in case of many inputs (linguistic variables) and states (linguistic values) per input; this will cause a huge increase in the number of the required rules for covering the problem space (this is similar to the curse of dimensionality).

Concluding, fuzzy logic is an ideal tool when dealing with imprecise or contradictive data, which may be modeled adequately with fuzzy sets, and combined with human logic. This implies that this approach give a good view of the significance of a combination of a set of inputs by sacrificing the precision.

4.2 Knowledge building schemes

As also discussed in Section 3 there are huge data sets and plenty of information available, though both they are not exploitable. John Naisbitt described the problem very well with his quote “we are drowning in information but starved for knowledge”. Thus knowledge discovery schemes are required for deriving useful outcomes from all the available sources of inputs. Several schemes have been proposed in the literature for knowledge building (learning) that could be classified in three categories, listed below:

- Unsupervised learning: Unsupervised learning schemes attempt to identify patterns and hidden structure of a set of (unlabeled) data. The learner has no bias (in terms of rewards, or error functions) against the data. The validation of the learning procedure is performed in later phases by (human) experts.
- Supervised learning: Supervised learning schemes attempt to identify patterns and structures using a training set. The training set is a dataset with labeled data, from where the learner may extract the patterns and the trends of the dataset that the decision maker will receive afterwards. The validity of the pattern in the training set is assumed a-priori.
- Reinforcement learning: Reinforcement learning is a learning scheme where the learner attempts to identify patterns in a set of unlabeled data using policies and reward functions. This family of mechanisms in general focuses on online learning and does try to build links between inputs and outputs.

Somewhere in the middle of supervised and unsupervised learning schemes lie the semi-supervised learning schemes, which identify patterns in unlabeled datasets using information provided by human experts. It is usually applied when the user has a limited set of labeled data, which are augmented with unlabeled records [136].

The main difference between the unsupervised schemes and the supervised and reinforcement ones is that in the first category the raw data drive the learner, whereas in the latter two categories the learner has a view of what to expect (supervised learning) or what he wants to do in the future (reinforcement learning).

4.2.1 Knowledge Discovery Cycle

From methodological point of view all the identified learning schemes share some basic principles; these principles are summarized in the knowledge discovery cycle. The knowledge cycle (is a slightly modified version of [137]), depicted in Figure 4-7 may be summarized in the following five steps:

- Feasibility study: suggests the first step of the analysis, which captures the problem identification (i.e., “what is the problem that we want to solve?”), and the problem description. In other words, the special characteristics of a dataset shall be identified. Additionally, the experts for handling the afore-described dataset shall be identified as well.
- Data Sources Identification: upon identifying the need for a learning process, as well the goals of the process, the proper types of inputs shall be considered. Additionally, the dataset to be used shall be considered and extracted.
- Data Cleaning and Preparation: schemes for cleaning and validating the inputs are required. This incorporates the cleaning of the inputs from outliers, and the building of a credible dataset that is suitable to the input dataset that will be fed to the learner. Finally, the significance of each input shall be extracted in this step.
- Data Enhancement: the data may be in such form that it is not exploitable from the learner (i.e., the inputs do not help the learner to make deductions or to learn, the data are too many, etc.). In parallel, new variables are inserted, calculated as combinations of other variables available in the data. These variables typically highlight relations among data and can be derived either heuristically based on statistics or deterministically using domain experts’ knowledge.
- Training, and Testing: is related to the operation of the algorithm and its training period (in supervised learning schemes). In unsupervised learning schemes the training period also exists but it is a part of the algorithm operation and is described as online learning.
- Evaluation: is related to the evaluation of the algorithm regarding its effectiveness and efficiency against a considered dataset. Regarding this step there is no holistic approach due to several reasons related to the

special characteristics of each algorithm (supervised learning, availability of labeled dataset, availability of experts, online/offline operation, etc.).

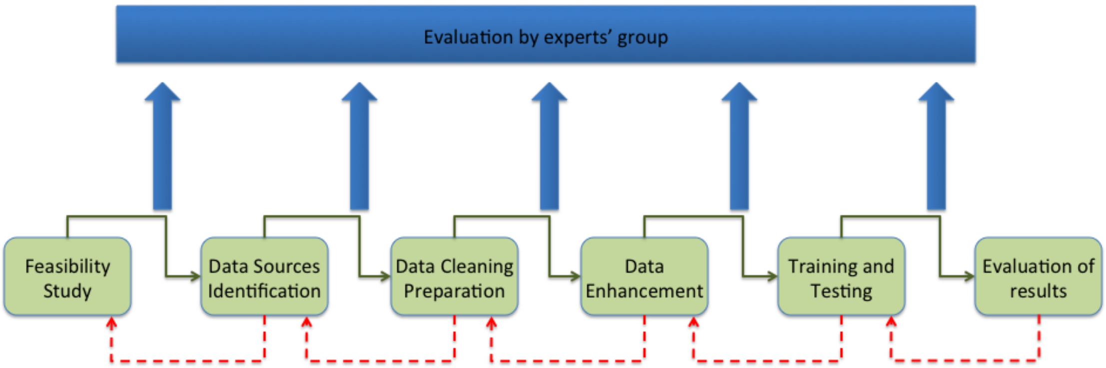


Figure 4-7: The Knowledge Discovery Cycle

The effectiveness of a learning solution is measured regarding the long-term gains, where the actual results are compared to the predicted results. If gains are achieved then the model is considered successful and is maintained otherwise it is recalibrated. Over time, the due to the environment evolution, the model may require modifications/enhancements. The key assumptions of the described methodology are:

- that the past is a good predictor of the future, and,
- that adequate data are available in order to build a model.

Both of these assumptions are strong, though numerous examples have proven that such problems may be overcome provided the sufficiently large dataset and the support of experts. Additionally, the effectiveness of a learning algorithm is associated to the need for data.

Similar methodological descriptions have been provided in the literature. In [138] the authors propose the Knowledge Discovery Life Cycle (Figure 4-8), which incorporates the following steps: (a) the planning, (b) hypothesis generation and testing, (c) knowledge discovery, (d) knowledge relevancy determination, (e) knowledge evolution, (f) evaluation against experts. The main difference of the two schemes (i.e., [136] with our modifications, and [138]) lies on the fact that they latter one incorporate in the knowledge discovery step all the data cleaning, and enhancement, as part of the knowledge extraction/data mining process. Still the Knowledge Discovery Life Cycle is an iterative process critiqued by experts at every single stage.

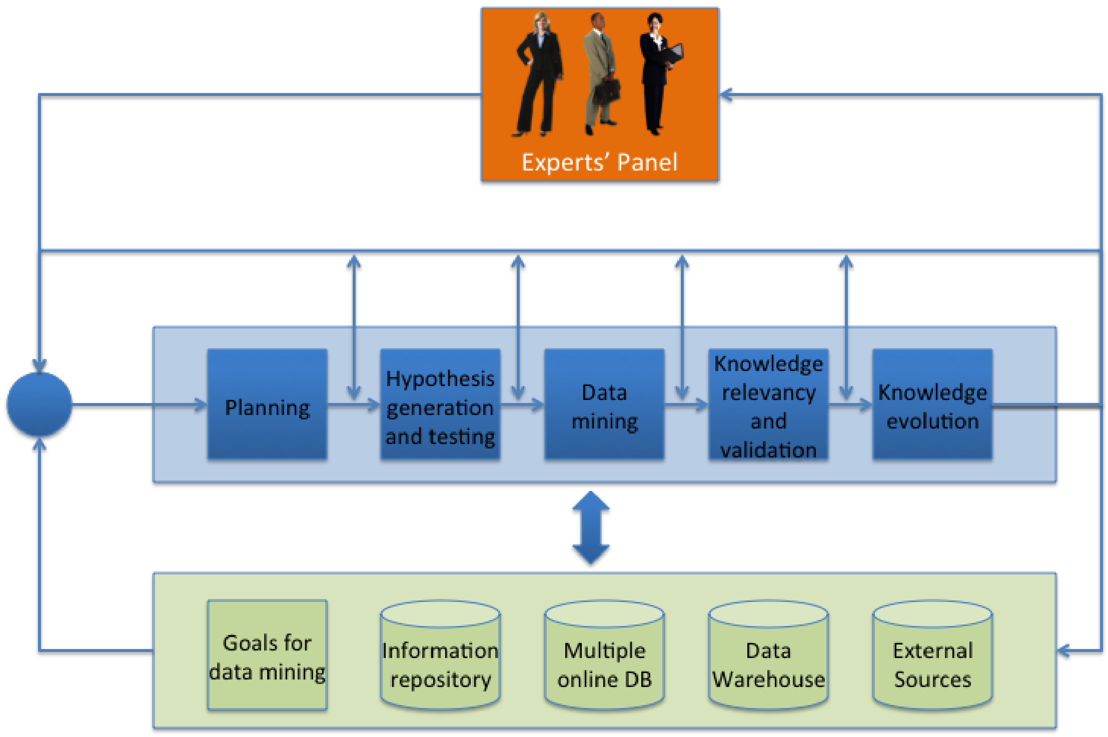


Figure 4-8: The Knowledge Discovery Cycle as proposed in [138]

4.2.2 Data pre-processing

Upon identifying the data to be used for the knowledge extraction, these data need to be cleaned and enhanced. Cleaning is the identification of bad or not representative inputs that are related to problematic measurement procedure, or lost measurements. The data enhancement concerns potential data transformations that will facilitate the knowledge building in terms of effectiveness and efficiency. More specifically:

- Data cleaning concerns the handling of incomplete, noisy, and inconsistent real world data. The purpose of data cleaning is to clean the data by filling in missing values, smooth noisy data and identify or remove outliers for avoiding confusing the learning schemes. Several schemes have been proposed in the literature ([139], [140], [141], [142]); the most important are shortlisted below:
    - Schemes for incomplete data
        - Ignore data: A naïve approach that is only applied when a considerable number of attributes or inputs is missing. It should be noted that this approach is in general ineffective and is applied in specific cases.

        - Fill in values manually: Although effective, this approach is time consuming and cannot be applied to large dataset with many missing values.
        - Fill in values automatically using predetermined inputs: A standard constant is used in order to replace all missing values of a field. Although it seems a valid compromise between the first and the second approaches, it is a rather poor choice since it is likely that the learning scheme will be confused by the repeated appearance of the same, "dummy" value. Variations of this scheme are:
            - Fill in using the mean value: Fill missing values over an attribute with the mean values of attribute. This method is widely used in practice but it offers poor results. Better results may be provided if instead of the global mean the attribute mean for all samples belonging to the same class as the missing tuple is employed (or using specific time windows).
            - Predict the missing value: Employ a prediction method (e.g. decision trees) in order to predict the expected value.
    - Schemes for noise reduction (noise is a random error or variance in a measured variable) [143]:
        - Binning: is applied to sorted values for smoothing variations and outliers. The idea is to replace a number of consecutive values with their mean.
        - Outlier detection: is an overall group of algorithms varying from statistical methods, to data mining ones. The schemes have been thoroughly described in [144].
        - Human Experts: human experts may be used for identifying inconsistencies in a dataset; in large datasets such approach is hard and may be applied only in specific contexts (e.g., intuitively wrong values for antennas gains, or received power values, etc.).

      - Regression: This is a typical and formal way of removing noise from data; data are replaced by a fitting curve which best describes their distribution. It is important here however to avoid overfitting, i.e., a curve which precisely captures data, since it will also contain the outliers we want to remove.
- Data transformation concerns transformations on the available dataset for facilitating the application of learning algorithms. Typically, this task involves bounding the range of variables, normalizing input values in order to avoid overflows or over-influence of a variable over the other variables. Out of the available schemes for data transformations we refer to the two most common:
   - Normalization, which is particularly useful for distance measurements such as nearest-neighbor classification and clustering [145]. There are many methods for data normalization; however the following three are the most widely used in practice.
      - Max-Min normalization: Given a variable X and it's minimum as well as maximum value X_{min}, X_{max} respectively, all values of X can be mapped to a new interval [a,b] using the following transformation

$$X' = (b-a)\frac{(X - X_{min})}{(X_{max} - X_{min})} + a \qquad (4.6)$$

      - Zero Mean Normalization: Given a variable X, its mean value X_m and standard deviation *std(X)*, all values of X can be normalized using the following transformation

$$X' = \frac{(X - X_{min})}{std(X)} \qquad (4.7)$$

      - Decimal Scaling Normalization: Given a variable X and its maximum value Xmax, all values of X are normalized using the following transformation:

$$X' = \frac{X}{10ceil(log_{10}X_{max})} \qquad (4.8)$$

   - Kernels are transformations that enable the learner to identify in a more clear way the similarities or the differences among the observations. The formal definition of a kernel function is the following [146]:

- A function $k: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is called a kernel on $\mathbb{R}^d$ if there is some function $\varphi: \mathbb{R}^d \to \mathbb{F}$ into some space $\mathbb{F}$ with scalar product $\langle .,. \rangle_{\mathbb{F}}$ such that:

$$k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathbb{F}} \ \ for\ all\ x, x' \in \mathbb{R}^d$$

Therefore, the computational cost of a kernel approach is at least square of the number of training data points and the memory requirement makes them intractable [147].

4.2.3 Unsupervised Learning

As mentioned in the introduction section, unsupervised learning tries to extract patterns from a large of unlabeled inputs. This implies that the learner does not have clues about the dataset that it will use for deriving knowledge. In general, in unsupervised learning, the learner attempts to find similarities among a set of observations and group them, a procedure usually referred as clustering. Within a cluster, objects tend to be similar and dissimilar with objects of other clusters. Following the taxonomy proposed in [148] the unsupervised learning schemes could be categorized using the following orthogonal aspects:

- Hierarchical vs. Partitional Methods:
    - Hierarchical clustering algorithms induce on the data a clustering structure parameterized by a similarity parameter.
    - Partitional methods essentially produce one partition of the data into clusters.
- Agglomerative vs. Divisive Methods:
    - Agglomerative methods start by assigning each sample to its own cluster, and proceed by merging clusters.
    - Divisive methods start by assigning all the samples to a unique cluster, and proceed by splitting clusters.
- Monothetic vs. Polythetic Methods:
    - Monothetic methods learn clusters using one feature at a time.
    - Polythetic methods use collections of features.
- Hard vs. Fuzzy:

  - As also described in fuzzy sets in subsection 4.1, In hard clustering each sample belong to one and only one cluster
  - In fuzzy clustering, samples have different degrees of membership to different clusters.
- Using Deterministic vs. Probabilistic Clusters.
  - If clusters are deterministic, a point either belongs to a cluster or does not belong to it.
  - If clusters are probabilistic, a point belongs to a certain cluster with a certain probability. Compared to fuzzy clusters, in the probabilistic ones, an observation belongs to a cluster with a certain probability, whereas in the former case to a certain degree.
- Using Deterministic vs. Stochastic Algorithms considering the algorithm used for the clustering.
- Incremental vs. Non-Incremental:
  - Incremental are the methods where the dataset may be augmented during the learning process.
  - Non-incremental are the methods where the full dataset is required for the initiation of the learning process.

In the following subsections we present two key algorithms that are the basis of the learning schemes developed in terms of this thesis and described in Section 6. The algorithms are the k-Means, which is a partitional, agglomerative, hard algorithm, and the Hierarchical clustering, which is a monothetic, non-incremental algorithm. It should be noted that, since observations are represented as vectors, similarity (or dissimilarity) could be measured by means of distance (e.g. Euclidean, Minkowski, Mahalanobis) or cosine similarity (e.g. the angle formed by the vectors defined by two objects).

k-Means

k-Means is a well-known data-mining clustering technique. The core idea of data clustering is to partition a set of N, d-dimensional, observations into such groups that intra-group observations exhibit minimum distances from each other (Figure 4-9), while inter-group distances are maximized. It should be noted that in its basic formulation k-Means is NP-Hard even for k = 2 ([149]), and there exist several heuristic solutions that derive local optima approximations to the objective function (for a literature survey one can refer to [150]).

k-Means [136] is based on the following objective function:

$$J = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \left(\sum_{k, x_k \in G_i} \|x_k - c_i\| \right) \quad (4.9)$$

Where,

- *c*: the number of clusters,
- G_i: the i^{th} group,
- x_k: the k^{th} vector in group J_i and represents the Euclidean distance between x_k and the cluster center c_i.

The partitioned groups are defined by using a membership matrix described by the variable U. Each element U_{ij} of this matrix equals to 1 if the specific j^{th} data point x_j belongs to cluster i, and 0 otherwise. The element U_{ij} is analyzed as follows:

$$U_{ij} = \begin{cases} 1, & if \|x_k - c_i\|^2 \leq \left\|x_j - c_k\right\|^2, for\ each\ k \neq i \\ 0, & otherwise \end{cases} \quad (4.10)$$

This means that x_j belongs to group *i*, if c_i is the closest of all centers.

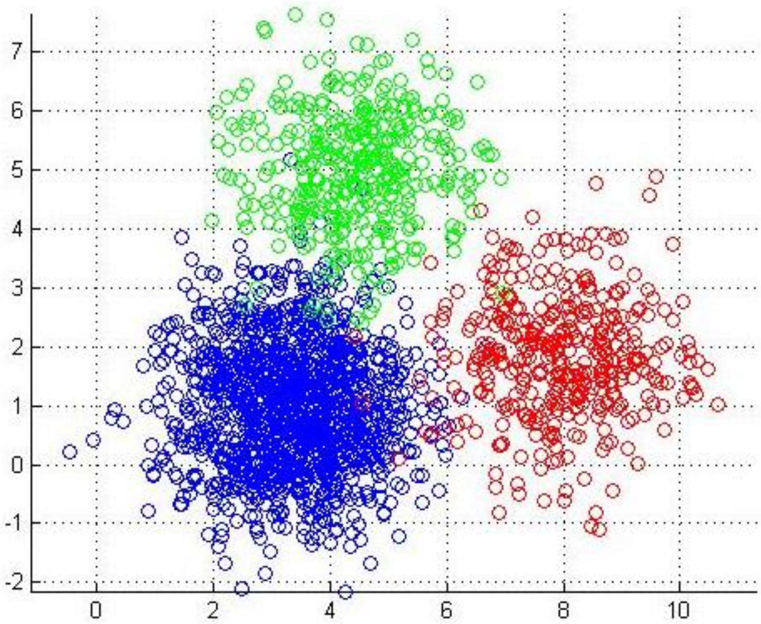


Figure 4-9: Visualization of the k-Means clustering for the three clusters

Hierarchical Clustering

Hierarchical clustering groups objects into a tree of clusters. Hierarchical clustering may be either built in top-down (divisive) approach or as a bottom-up (agglomerative) [136].

Hierarchical divisive clustering has as input n objects and groups them in the same cluster. In each iteration a cluster is split based on the criteria set (e.g., split the cluster from which will emerge two clusters with the least similarity between them). Gradually, smaller clusters are formed and the algorithm finishes either when all objects belong to single-object clusters or when certain termination condition is satisfied (e.g., number of clusters, minimum distance threshold between clusters). At high levels of hierarchy the split decision is difficult due to the comparison of all the objects. As a result this approach has rarely been applied.

Hierarchical agglomerative clustering is the complete opposite process. To begin with, each object forms a single-object cluster. The algorithm continues by merging two clusters with the highest similarity at each step (e.g., choose two clusters with the smallest distance between them). This process is repeated until all objects belong to the same group or until certain criteria are met (e.g., number of cluster, maximum distance threshold between clusters). Unlike divisive method, agglomerative clustering is greatly applied and lots of variations have been developed. Their diversification resides in their definition of between-cluster similarity.

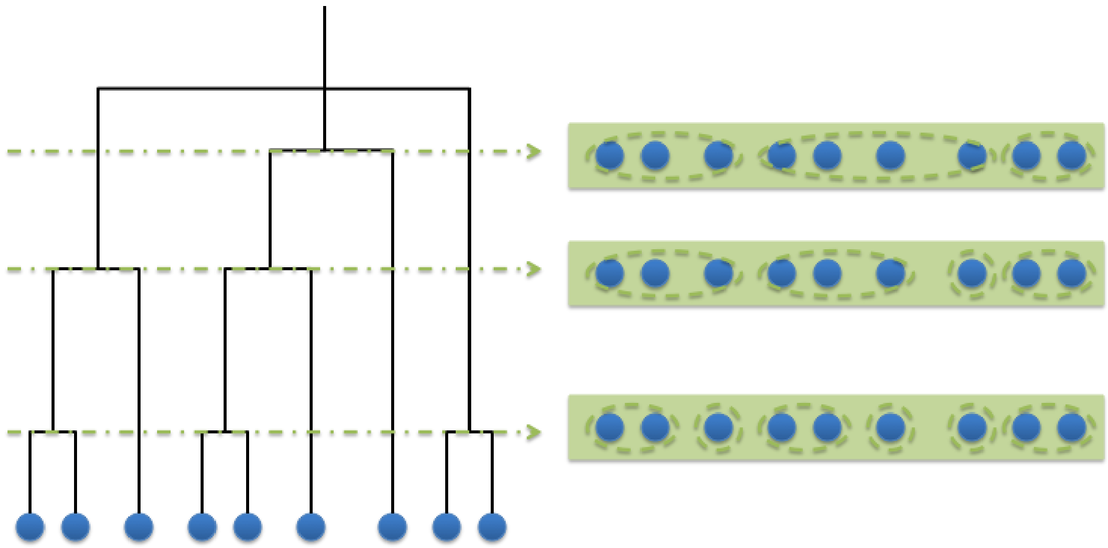


Figure 4-10: Visualization of the agglomerative HAC, for nine observations; at the right part the clusters at different phases of clustering (i.e., different termination conditions)

4.2.4 Supervised Learning

Supervised learning is the application of a learning algorithm to a training dataset for developing a model. Then, the model is being evaluated with a test dataset, which evaluates the accuracy and the efficiency of the dataset. Both training data and test data are labeled inputs that are being used for identification of the patterns of that appear in the environment [145].

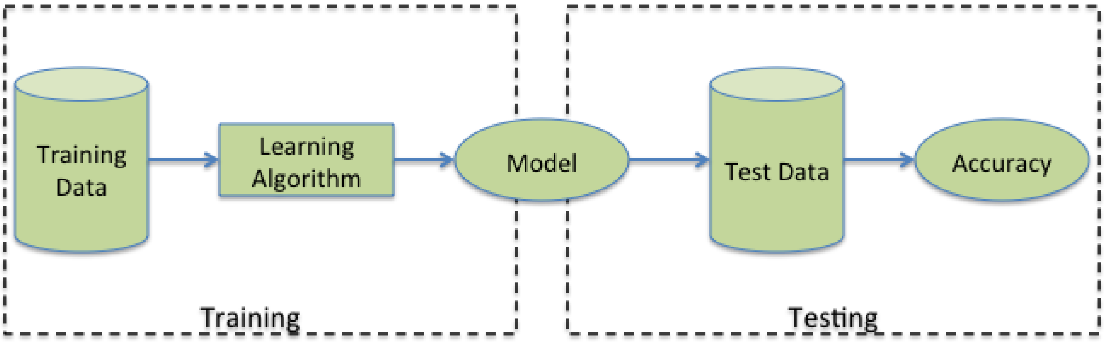


Figure 4-11: Supervised learning model [145]

According to [136] and [145], Supervised Learning algorithms may be categorized into the following categories (the predictive techniques are proposed in the [136]):

- Decision tree learning is one of the most widely used techniques for classification, because its classification accuracy is competitive with other methods, and it is very efficient. It may be considered as flowcharts where each non-leaf node corresponds to an evaluation of an attribute while every branch starting from that node represents different possibilities for categorization. It should be added that all current tree-building algorithms are heuristic algorithms.

- Bayesian Classification: Bayesian classifiers are classifiers that are based on statistics and in particular Bayes' theorem. Their main advantages are that they are easy to implement, very efficient, and they manage to achieve good results obtained in many applications. Their main disadvantage lies on the class conditional independence advantage, which leads to loss of accuracy when the assumption is seriously violated (those highly correlated data sets).

- Support Vector Machines: Support Vector Machines (SVMs) are linear classifiers that find a hyperplane to separate two class of data, positive and negative with the use of kernel functions for nonlinear separation. SVMs have a rigorous theoretical foundation, and are very accurate in classification compared to other methods in applications, especially for high dimensional data, (it is perhaps the best classifier for text classification).

- Learning from Neighborhood or Lazy Learners: This family of classification techniques gets its name from the way it operates on the training set. Instead of building a model, as their name also highlights, neighborhood learners exploit the information from the neighborhood for categorizing a new observation, by exploiting the training data set.
- Predictive Techniques: Predictive techniques are algorithms that given a set of time-labelled observations attempt to predict the value of a variable in the future.

In the following sections the decision trees and the k Nearest Neighbors (kNN) techniques are being analyzed. The first one because it is a traditional supervised learning technique that will help capture the key aspects of supervised learning, and the latter because it will be used in the for the hybrid learning scheme proposed in Section 6.

Decision trees

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features [151].

A model consists of a dendrogram and a set of rules. For building a decision tree, labeled observations (vectors) are used as inputs; these vectors shall be categorized in such way so as to return Yes/No results. Decision Trees typically implement Boolean classification functions (True/False, 0/1, Cold/Cool/Warm/Hot) on every level of the hierarchy; every level transition corresponds to the evaluation of a single variable [136]. Given a set of observations, the goal is to construct a decision tree so as to facilitate future decisions. A simple example could be the approval or the rejection of loans' requests, depending on the profile of the person that makes the request. Table 4-2 presents the training set, used for identifying the potential links among the observations regarding the ability of the people to pay their loan, in conjunction with their employment status, their residence status, and their credit rating. Figure 4-12 is the developed tree for the Table 4-2 dataset. When asking a simple question, for example, "will a young person with no job, no house, but with good credit rating get his loan request approved?", we observe that the request will not be approved (red arrows in Figure 4-12) [145].

Table 4-2: Load Data table

ID	Age	Has Job	Own House	Credit Rating	Class
1	young	false	false	fair	No
2	young	false	false	good	No
3	young	true	false	good	Yes
4	young	true	true	fair	Yes
5	young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

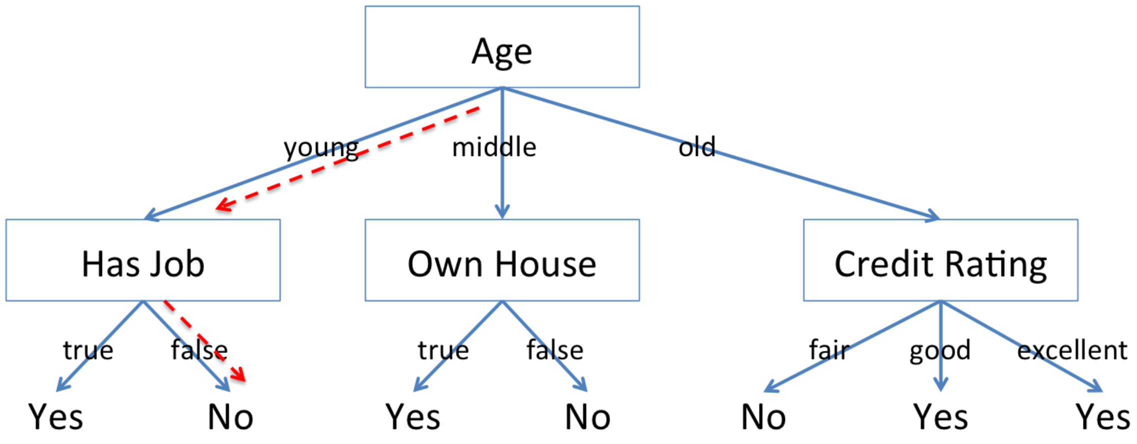


Figure 4-12: Decision tree based on the load data. The red dashed line represents the answer to the query "will a young person with no job, no house, but with good credit rating get his loan request approved?" [145]

Decision trees have many advantages, which are shortlisted below:

- They may handle large number of variables,
- They may support diverse input types (e.g., nominal, numeric etc.),
- They are well studied in computer science,
- They may be enhanced for enhanced efficiency (e.g., greedy versions of the decision trees)
- They are humanly understandable.

Their main disadvantages lie on the way missing values may be handled, overfitting, and how attributes with different weights are being handled [145].

k Nearest Neighbor

k Nearest Neighbor (kNN) classifier is a "learning from your neighborhood" technique (the family also is called lazy learners). Compared to other learning methods, kNN does not build model from the training data; instead it evaluates the dataset every time a new vector/input arrives (i.e., it is considered supervised learning because it requires a training dataset). The scheme, for a test instance d, calculates its distance from the neighboring instances and classifies d, to that class which is more likely to belong, based on its neighbors (the algorithm is presented in Table 4-3) [145]. As also stated above, it should be highlighted the fact that no training is required, and that the classification time is linear in training set size for each new observation. The number k is usually chosen empirically via a validation set or cross-validation by trying a range of k values. Figure 4-13 presents a simple example with k = 3, and how a new observation (the gray one) would be classified into red or blue class.

Table 4-3: The kNN algorithm

Algorithm kNN *(D, d, k)*	
1	Compute the distance between *d* and every example in *D*
2	Choose the *k* examples in *D* that are nearest to *d,* denote the set by $P(\subseteq D)$
3	Assign *d* the class that is the most frequent class in *P* (or the majority class)

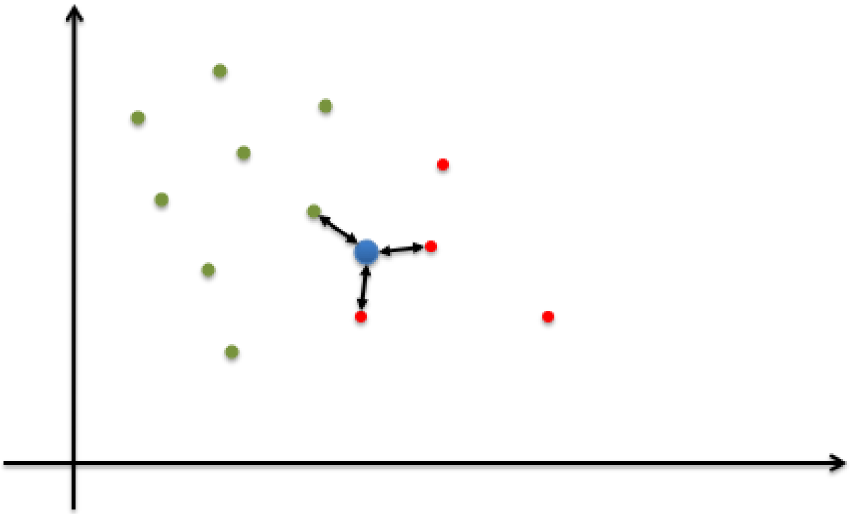


Figure 4-13: Graphical representation for of the kNN algorithm for k = 3.

As a conclusion, it should be mentioned that kNN can deal with complex and arbitrary decision boundaries, and despite its simplicity, the classification accuracy of kNN can be quite strong. On the other hand, kNN is slow at the classification time and it does not produce an understandable model; also the distance function to be used is crucial, and is application dependent.

4.2.5 Reinforcement Learning

Reinforcement learning is a sub-area of machine learning focusing on how an agent should take actions in an environment, so as to maximize its long-term reward. In this scope, the learner is not told which actions to take, but instead must discover which actions yield the most reward by trying them. Reinforcement learning algorithms attempt to find a policy that maps states of the world to the actions the agent ought to take in those states. Therefore, it is different from supervised learning, in which knowledge is acquired via examples provided by an external supervisor [152]. There are three fundamental methods for solving reinforcement learning problems, listed below:

- Dynamic programming, which is very well-developed mathematically but requires a complete and accurate model of the environment that unfortunately does not exist in many application scenarios.
- Monte Carlo methods, which do not require a model and are very simple conceptually but are not suited for step-by-step incremental computation.
- Temporal difference (TD) schemes, which do not require an exhaustively structured model of the environment and are naturally implemented in an online, fully incremental fashion. TD methods learn their estimates in part on the basis of other estimates

TD, which is a combination of the other two methods is the most attractive, because it is simpler and may work both in online and offline manner [153]. In the following subsections, initially the mathematical formulation for the environment representation is described for setting the basis for the analysis of a well studied and representative technique, the Q-learning, of the third family (i.e. TD), which is presented afterwards, for highlighting the key aspects of reinforcement learning algorithms.

Environment modeling

In general in reinforcement learning, the environment is modeled as a set of state and action pairs; the actions link the states. Formally, the environment model of reinforcement learning scheme consists of [154]:

- a discrete set of environment states, S,
- a discrete set of agent actions, A, and,
- a set of scalar reinforcement signals; typically (0; 1), or the real numbers.

Problems with delayed reinforcement (feedback) are well modeled as Markov decision processes (MDPs) [154]. An MDP is represented by:

- a set of states S,
- a set of actions A,
- a reward function $\mathbb{R}: \mathrm{S} \times \mathrm{A} \rightarrow \mathbb{R}$, and
- a state transition function $\mathrm{T}: S \times \mathrm{A} \rightarrow \Pi(S)$, where a member of $\Pi(S)$ is a probability distribution over the set *S* (i.e., it maps states to probabilities).

We write $\mathrm{T}(s, a, s')$ for the probability of making a transition from state s to state s' using action a.

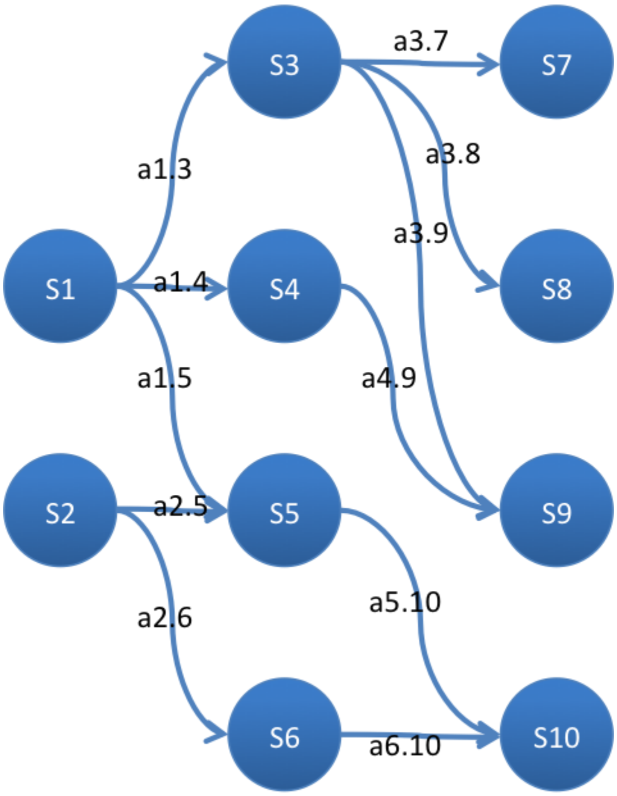


Figure 4-14: Environment modeling (state-action pairs) for 10 states and 11 actions linking the states.

The state transition function probabilistically specifies the next state of the environment as a function of its current state and the agent's action. The reward function specifies expected instantaneous reward as a function of the current state and action. The model is Markov if the state transitions are independent of any previous environment states or agent actions.

The large number of states and the corresponding actions leads to a huge number of state-actions pairs; such huge number makes several problems hard to be handled (curse of dimensionality). Thus combinations of reinforcement learning and fuzzy logic have been proposed in the literature for reducing the number of the states.

Q-Learning

In general, reinforcement learning is a form of learning in which an agent interacts with its environment. The agent takes an action and this action changes the environment in some manner, and this change is communicated to the agent through a scalar reinforcement signal. The environment is typically formulated as a finite-state Markov decision process (MDP). Therefore, when using reinforcement learning, the environment needs to be envisioned as a set of discrete states.

In Q-Learning the environment is envisioned as a set of States $s \in S$ and a set of Actions $a \in \mathrm{A}$. At each given time step, t, the agent performs an action and moves between states. Each state provides the agent a reward $r \in \mathrm{R}$, with the aim of the agent being to maximize the total received reward. The algorithm therefore has a function, which calculates the Quality of a state-action combination: $\mathrm{Q}: \mathrm{S} \times \mathrm{A} \rightarrow \mathbb{R}$. Q-values are updated iteratively ([155]):

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + a_t(s_t, a_t) \times [R(s_{t+1}) + \gamma \max_a Q(s_{t+1}, a_t) - Q(s_t, a_t)] \quad (4.11)$$

The architecture of a reinforcement learning agent is provided in Figure 4-15. The state and the reward denote the two signals that the agent received from the environment stimuli. On the right, the action denotes the only signal the environment receives from the agent. For each step, the agent receives state and reward signals and then produces an action signal that changes the environment. The time shifting is captured by the dashed line in the bottom of the figure [156].

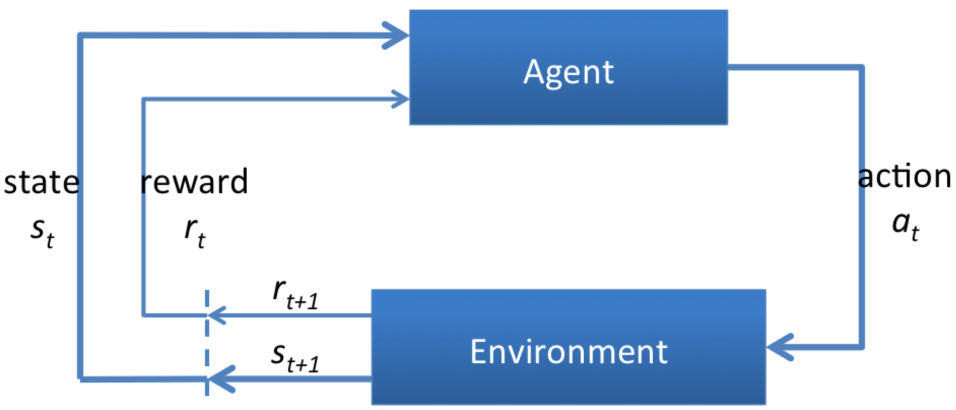


Figure 4-15: Schematic representation of the Q-Learning [156], [157]

5. Fuzzy Logic based Situation Perception

The previous sections have set the foundations for describing the Autonomous Systems (Section 2), as well as a key aspect of these systems, the Situation Awareness procedure (Section 3). Additionally, several approaches, both architectural and algorithmic ones, have been analyzed thoroughly so as to highlight the gaps that exist in such systems. A key gap that was identified by this analysis is the Situation Perception functionality, which is the ability of a system (or an entity) to effectively observe its environment and identify its current status. Thus, in Section 4 a set of schemes has been presented that are suitable for enhancing situation perception. The rest of this section initially presents briefly, how fuzzy logic could be used for Situation Perception mechanisms; then three different case studies are being presented exploiting fuzzy logic for proper situation perception and environment modeling.

5.1 Fuzzy Reasoners' Situation Perception

The term "Situation Perception" is used to describe all correlations that take place in order to analyze data received by monitoring points and thus identify problems and select appropriate configuration actions. This task is considered as a complex one due to:

- Its multi-variable nature since multiple optimization goals or faults may arise,
- often contradictive inputs may occur,
- missing data that may arise.

These aspects may be handled using Fuzzy Sets and rule-based schemes. Additionally, Situation Perception is related to identifying the decision makers' status, which is at least adequately handled by fuzzy systems. Therefore, Fuzzy Logic algorithmic tool (i.e., fuzzy sets enhanced with rules and policies) is ideal for dealing with situation perception problems. The Fuzzy Logic based situation perception, takes into account a set of metrics/parameters, and after their joint correlation analysis, maps them to a degree that depicts how the network elements perceives its environment (e.g., load status). This perception is mapped to a value for each state that ranges between 0 and 1, and describes the degree of each state e.g., load, interference, noise, mobility status, uncertainty status, etc. This approach provides a human-like perception of the environment, which is easy to model and to be handled by the system administrators.

As thoroughly described in Section 4, The Fuzzy Logic Controller (FLC) consists of three parts, namely the fuzzifier, the inference system and the defuzzifier (Figure 5-1). The fuzzifier undertakes the transformation (fuzzification), of the input values (crisp values) to the degree that these values belong to a specific state (e.g., low, high). Then, the inference system correlates the inputs and the outputs using simple “IF...THEN...” rules. Table 5-1 presents the rules that combine the membership functions (i.e., degrees belonging to each state) of N inputs with the corresponding membership functions of each of the outputs. This combination may be exhaustive (i.e., including all inputs’ combinations) or incomplete (i.e., with missing combinations). Each rule results to a certain degree for belonging to a specific state for every output. Thereinafter, the output degrees for all the rules of the inference phase are being aggregated. The actual output of the decision making process, comes from the defuzzification procedure, which captures the degree of the state of the decision maker (e.g., the network element is x% loaded; the radio link is y% interference, the user experiences z% QoS etc.). The degree is obtained using several defuzzification methods; the most popular is the centroid calculation (Section 4.1.2), which returns the center of gravity of the degrees of the aggregated outputs.

Table 5-1: Correlation table between inputs and outputs in fuzzy logic

Input 1	**Input 2**	**…**	**Input N**	**Output 1**	**…**	**Output M**
I1_MF1	I2_MF1	…	IN_MF1	O1_MF1	…	OM_MF1
I1_MF2	I2_MF1	…	IN_MF1	O1_MF1	…	OM_MF1
…	…	…	…	…	…	…
I1_MFN	I2_MFN	…	IN_MFN-1	O1_MFN-1	…	OM_MF1
I1_MFN	I2_MFN	…	IN_MFN	O1_MFN	…	OM_MF1

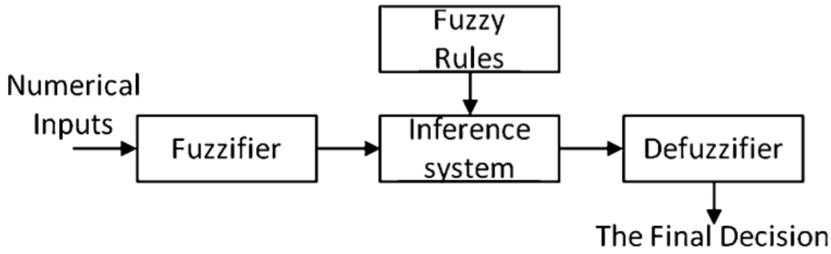


Figure 5-1: High level view of fuzzy reasoner for situation perception

5.2 Fuzzy Logic based Situation Perception case studies

The previously described proposal, for using fuzzy logic for situation perception problems, is one of the main proposals of this thesis. In section 5.2 specific case studies are being analyzed. In the following subsections we provide the analysis of fuzzy logic based Situation Awareness approaches developed for:

- QoS Degradation Events' Identification,
- Load events' identification,
- Environment modeling for Cooperative Power Control.

5.2.1 QoS Degradation Events' Identification

The increasing complexity of new networks, combined with the versatility of newly introduced services, poses the need to incorporate new capabilities and intelligence into the network elements in order to meet the management needs of such services and network contexts. The capability of the network to identify its status enables it to react promptly and autonomously once an event or error has been identified. The use of service information in this process enables the network elements to identify even more composite problems or to act more targeted in order to solve complex errors. Towards this direction, fuzzy logic-based QoS degradation events' identification mechanism has been developed. The QoS degradation events' identification scheme targets identification of QoS degradation events in IP networks for the VoIP service, though the same approach could be applied for other services, with limited modifications. In the following sections we present the details of the algorithm as well as the evaluation approach used for the benchmarking of our solution.

QoS Degradation Events' Identification high-level description

The key idea in the QoS degradation events' identification is to differentiate the different services and apply fuzzy reasoned for each service. This will enable to handle the services (e.g., voice services, video streaming, keep alive messages, etc.) with different characteristics in a different manner for capturing their special characteristics. Thus, we propose the introduction of a situation perception mechanism, which will allow the management system, using inputs from network elements (network and service data) to proceed in identification of QoS degradation events in IP networks. More specifically, we propose the introduction of a scheme that will exploit service information and by combining it with network measurements will identify whether a given service flow experiences a problematic situation or not.

According to the previous description, each network element shall employ the following set of functionalities (Figure 5-2):

- The "Application" function represents each separate service that may be implemented in a mobile device. Examples are the Voice over IP service, video streaming applications, etc.
- The "Communication" module represents the communication layers 1, 2, 3, and 4, and may have different implementations depending on the service (e.g., other applications require TCP connection whereas other require UDP one, or keep alive messages may be sent only via WiFi interface, etc.).
- The "Monitoring" functionality is responsible for gathering local information and feeding the situation perception function. It is lined with the communication module for gathering network related information. It also maintains a link with the active applications for having knowledge on the inputs that it shall forward to the situation perception.
- The "Situation perception" function is employed using fuzzy logic reasoners. The fuzzy logic reasoners have different inputs and configuration depending on the application and are linked to the memory module for storing the outcome of each decision.

The "Learning" part incorporates the learning mechanism for enhancing the network element's situation perception that will be analyzed in Section 6. After each learning action, the situation perception is being updated according to the outcome of the adaptation process.

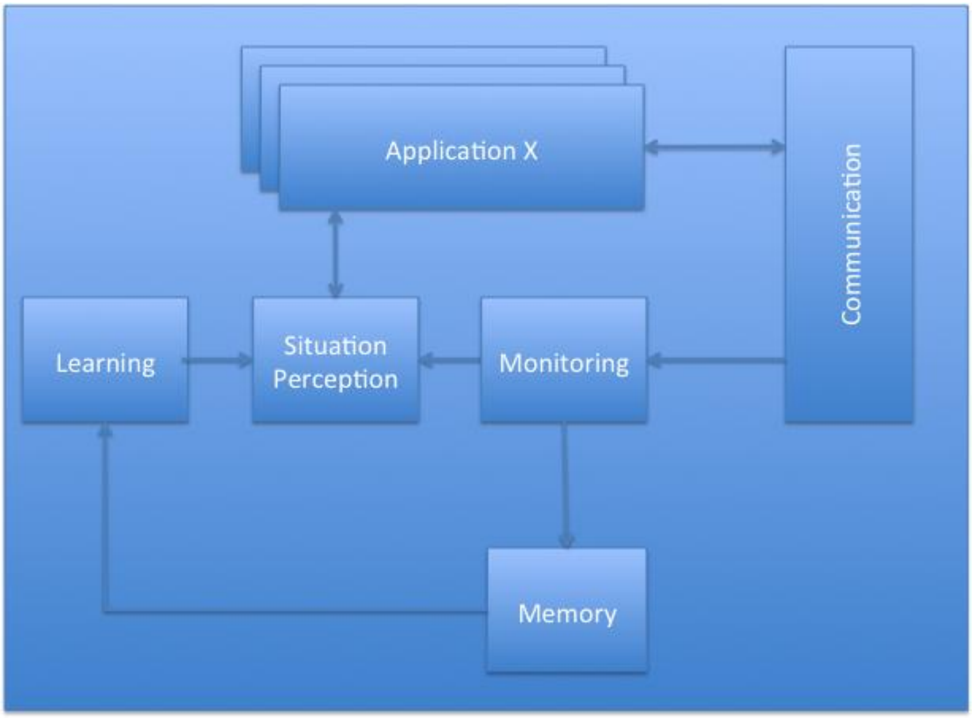


Figure 5-2: Functional description of QoS Degradation Events' Identification

QoS Degradation Events' Identification description and performance analysis

For the case study under consideration and for the VoIP service, delay, jitter and packet loss are identified as playing significant role in QoS degradation, thus the situation awareness engine is based on the aforementioned inputs for every active session; for other services (e.g., highly reliable communications, IPTV etc.) other monitoring inputs could be used. The "Delay", the "Jitter" and the "Packet Loss" per flow comprise the input vector, whereas the output is the "QoS degradation" [158], [159], [160]. The membership functions indicate the values of each parameter, the range of each value and the magnitude of their participation. The shapes chosen for the representation of the degree of certainty are trapezoidal in the case of "Packet Loss", mainly for simplicity reasons and to so as to exploit the certainty areas for such inputs (Figure 5-3) and triangular for the jitter and the delay for highlighting the symmetry and the absence of total certainty areas (Figure 5-4, Figure 5-5). For the QoS level the Gaussian membership functions are being used. The idea behind such adoption is mainly based on the smooth (i.e., the QoS should be related to the inputs in a smooth manner without non-linear alterations - Figure 5-6) and non-zero (the decision maker needs to conclude to a decision based on all inputs' range) nature at all points; for simplicity reason symmetric membership functions are being used. The rules for the linking of the inputs with the output are presented in Table 5-2.

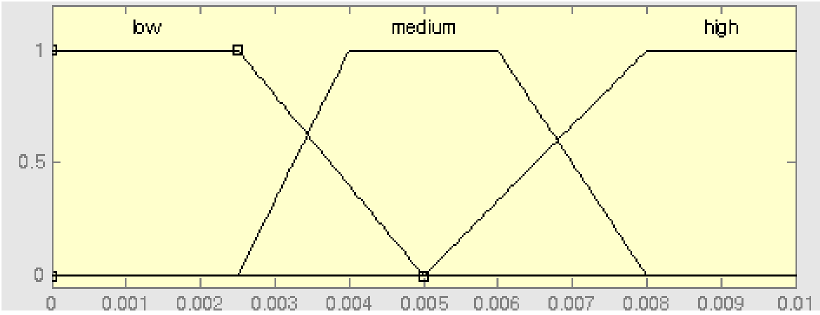


Figure 5-3: Input membership functions for the Packet Loss

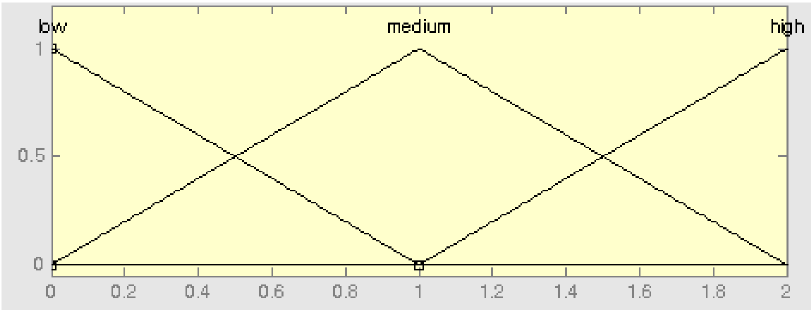


Figure 5-4: Input membership functions for the Jitter (jitter is in seconds)

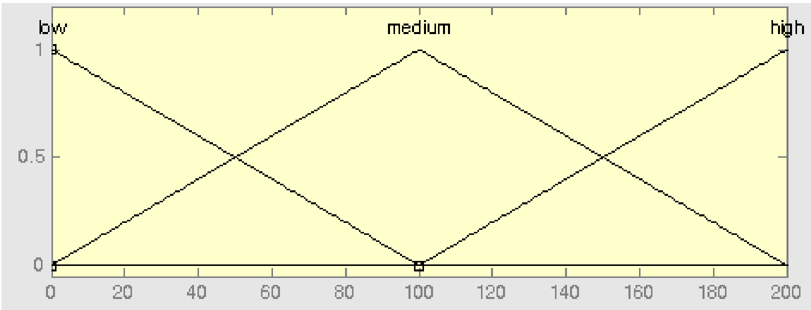


Figure 5-5: Input membership functions for the Delay (delay is in ms)

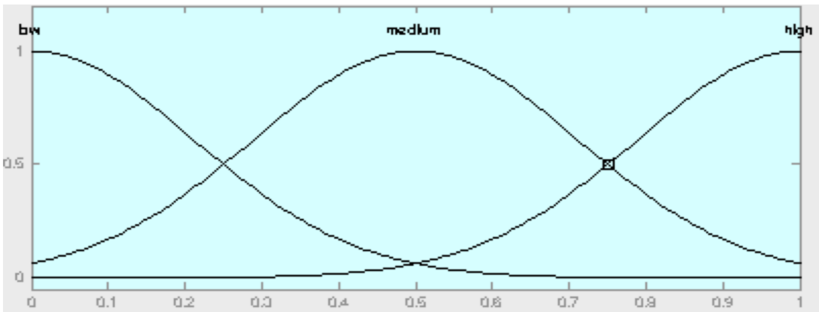


Figure 5-6: Output membership functions for the QoS Degradation

Table 5-2: Correlation table between inputs and outputs for the QoS degradation identification

Rule Number	Delay	Jitter	PL	QoS
1	Low	Low	Low	High
2	Low	Low	Medium	High
3	Low	Low	High	Medium
4	Low	Medium	Low	High
5	Low	Medium	Medium	Medium
6	Low	Medium	High	Medium
7	Low	High	Low	Medium
8	Low	High	Medium	Medium
9	Low	High	High	Low
10	Medium	Low	Low	High
11	Medium	Low	Medium	Medium
12	Medium	Low	High	Medium
13	Medium	Medium	Low	Medium
14	Medium	Medium	Medium	Medium
15	Medium	Medium	High	Low
16	Medium	High	Low	Medium
17	Medium	High	Medium	Low
18	Medium	High	High	Low
19	High	Low	Low	Medium
20	High	Low	Medium	Medium
21	High	Low	High	Low
22	High	Medium	Low	Medium
23	High	Medium	Medium	Low
24	High	Medium	High	Low
25	High	High	Low	Medium
26	High	High	Medium	Low
27	High	High	High	Low

The evaluation of the fuzzy logic based situation awareness scheme for QoS degradation events is based on 50000 tuples that have been generated randomly, using typical values available in the literature [158], [159], [160]. For the analysis of the dataset, and given the huge number of the considered values, the dataset is being automatically evaluated against a set of predefined fuzzy logic rules, which are strict for the identified service. The extracted labels from now on will be called ground truth and are extracted using rules that perfectly fit to the considered dataset and are only used for the evaluation of the generic definition. The used membership functions are captured from the Table 5-3. Using the afore described initial configuration, the success rate is 64%, which represents the number that the situation awareness scheme concluded in the same decision compared to the ground truth.

Table 5-3: Inputs Membership Functions for the evaluation of the dataset

	Delay	**Jitter**	**PL**
Low	0 – 10	0 – 0.4	0 – 0.005
Medium	8 – 80	0.33 – 1	0 0.004 – 0.01
High	30 – 200	0.63 – 2	0.005 – 0.01

5.2.2 Load Events Identification

In the autonomic network vision, each network device (e.g., router, access point, etc.) is potentially considered as an autonomic element, which is capable of monitoring its network-related state and modifying it based on conditions that administrators have specified. The autonomous element, with embedded cognition, includes processes for monitoring and perceiving network node's internal state and environmental conditions, and then planning, deciding and finally adapting according to these conditions. Furthermore, such an element is able to learn from these adaptations (re-configurations) and assess their effectiveness for improving decision-making mechanisms.

To this end, the autonomous elements shall be able to make the basic reasoning decisions for identifying their status, for example identifying whether they are interfered or overloaded. Such decisions are related to access (e.g., WiFi APs, BSs, etc.) and core network elements (e.g., routers, servers, etc.). Regarding the access network elements, and more specifically the WiFi APs, a key decision regarding their status is the identification of load events.

Up to now such decisions have been based on vendors' configurations using static thresholds. The configuration is generic, based on strict thresholds, and aims at capturing all the potential environments (ultra dense environments and environments with very demanding users in terms of bandwidth). However, the use of strict thresholds fails at meeting the requirements for generic configuration because the load identification is a multi-criteria problem with potentially contradictive inputs. For example, when in a WiFi AP there are numerous UEs associated it is considered loaded, even if the users are not consuming the available BW. Similarly, if a UE consumes all the available bandwidth of an AP, it is loaded. Simple static thresholds do not enable the distinction of the previously mentioned cases. Thus, the introduction of fuzzy logic based situation perception is proposed. This section presents the fuzzy logic based situation perception mechanism and corresponding evaluation of the proposal. The situation perception is incorporated in a self-managed network.

Fuzzy Logic Based Load Event's Identification high-level description

As mentioned in the previous paragraph, the Load events' identification mechanism is introduced in a self-managed network. The network architecture follows the principles presented in Section 3. The basic idea of the network can be summarized by Figure 3-3, which captures a hierarchical distribution of cognitive cycles, breaking down the respective functional entities and mechanisms for solving network management problems and other self-management operations (e.g., learning, monitoring) to:

- Network elements (e.g., access points – Network Element Controller level),
- Network compartments (opportunistic/short-term federations of network elements – Group of Network Element Controllers level),
- Network domains (structured/long-term federations of network elements – Network Domain Controller level), and,
- Network management system that controls the underlying entities and provides the human (e.g., administrator) interface.

Figure 5-7 provides a high level description of the functionalities of the NEC and the NDC. The functional decomposition also describes the learning/adaptation functionalities of the proposed scheme that are being presented in Section 6.

The scheme is highly decentralized, in the sense that a part of the algorithm is executed at the network elements level (NEC) and another part at the controller (NDC) level. Each NEC periodically monitors its operational environment and evaluates the identified information in order to deduce (using a fuzzy logic inference engine) faults or optimization opportunities (e.g., high load, high interference) and –if necessary– an appropriate configuration action to be executed (e.g., channel reallocation, assisted handover of associated terminals). Each NEC evaluates the correctness of each deduction (followed by a configuration action or not) building the so-called ground truth. Ground truth characterizes the actual conditions (i.e., load) and is identified by assessing the network node status after the re-configuration [134].

However, in many cases, according to the network conditions, the range for the characterization of a situation should be adapted according to the effectiveness of the selected configuration action and the specific features of the network environment. Nevertheless, this decision necessitates a holistic view of the network and consequently implies the dissemination of the aggregated set of local observations (NEC) to a higher level – in terms of architecture – entity (NDC). The NDC collects individual NECs observations and by applying data mining techniques specifies the new bounds for the fuzzy logic inference engine.

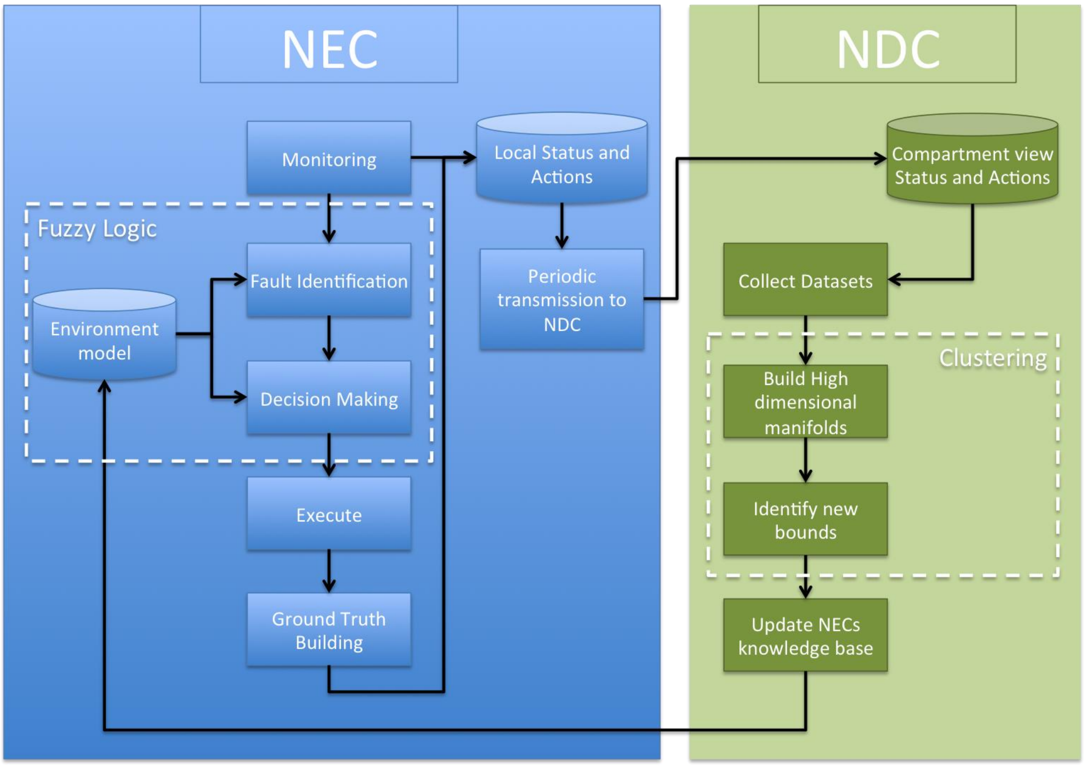


Figure 5-7: Functional decomposition of the Load events identification scheme

<u>Fuzzy Logic Based Load Event's Identification – Fuzzy reasoners' configuration and performance analysis</u>

In the considered case, we focus on the identification of Load events using fuzzy logic. The problem of load identification is under the coverage and capacity optimization umbrella. The analysis could be extended for any inference process that an NEC should execute. In this use case, each AP monitors its operational environment (Packet Error Rate, Channel Utilization, and Number of Associated Terminals) and attempts to identify potential (high) load situations. If such a problem occurs (high load) then they collaborate in order to select the most appropriate configuration action, which in this case is the optimal reallocation of the associated terminals among the available homogeneous or heterogeneous access points in the corresponding network area; the UEs reallocation has been extensively studied in the literature and is out of the scope of this analysis [134] [161].

The rules used for the decision-making procedure follow the format of equation below and are being presented in Table 5-4.

IF PER IS low AND CU IS low AND AT IS High THEN Load IS low.

Table 5-4: Correlation table between inputs and outputs for the Load events identification

Rule Number	PER	CU	AT	Load
1	Low	Low	Low	Low
2	Low	Low	Medium	Medium
3	Low	Low	High	Medium
4	Low	Medium	Low	Low
5	Low	Medium	Medium	Medium
6	Low	Medium	High	High
7	Low	High	Low	Medium
8	Low	High	Medium	Medium
9	Low	High	High	High
10	Medium	Low	Low	Low
11	Medium	Low	Medium	Low
12	Medium	Low	High	Medium
13	Medium	Medium	Low	Low

14	Medium	Medium	Medium	Medium
15	Medium	Medium	High	High
16	Medium	High	Low	Medium
17	Medium	High	Medium	Medium
18	Medium	High	High	High
19	High	Low	Low	Low
20	High	Low	Medium	Medium
21	High	Low	High	High
22	High	Medium	Low	Medium
23	High	Medium	Medium	Medium
24	High	Medium	High	High
25	High	High	Low	High
26	High	High	Medium	High
27	High	High	High	High

The shape of the membership functions (MF) is related to their special characteristics. More specifically, for the AT the MFs are trapezoidal. The key characteristic of this MF is its simplicity and is mainly used to describe inputs that have a homogeneity degree and linear behavior. Similarly, for the strict nature of the PER and its relation to the QoS we have decided to use the trapezoidal MFs, which describe in a satisfactory manner the considered error ranges for ideal ("low"), acceptable ("medium") and non acceptable ("high") [162]. Finally, the triangular MFs have been selected for the "Channel Utilization" parameter due to the linear affect of this input to a WiFi AP [163]. In order to test the effectiveness of the proposed solution we have used three initial configurations of the fuzzy logic decision making controller so as to capture more generic and more targeted configurations of the network equipment (Table 5-4 – FL_i, where i captures the initial configuration, ranging from 1 -very generic- to 3 -more targeted). Figure 5-8, Figure 5-9, and Figure 5-10 present the first (more generic) initial configuration of the fuzzy logic controller. Similarly, we have configured the fuzzy logic controller in the other two configurations.

Table 5-5: Membership functions bounds for the Load events identification

	PER	Channel Utilization	Update Interval
Low	$FL_{1,2,3}$:[0…0.01]	$FL_{1,2}$: [0…0.2] FL_3: [0…0.5]	$FL_{1,2,3}$: [0…10]
Medium	$FL_{1,2,3}$: [10^{-4}…0.05]	$FL_{1,2,3}$: [0.1…0.9]	$FL_{1,3}$: [9…16] FL_2: [5…20]
High	$FL_{1,2,3}$: [$7*10^{-3}$…1]	$FL_{1,2}$: [0.8…1] FL_3:[0.5…1]	$FL_{1,2,3}$: [15…25]

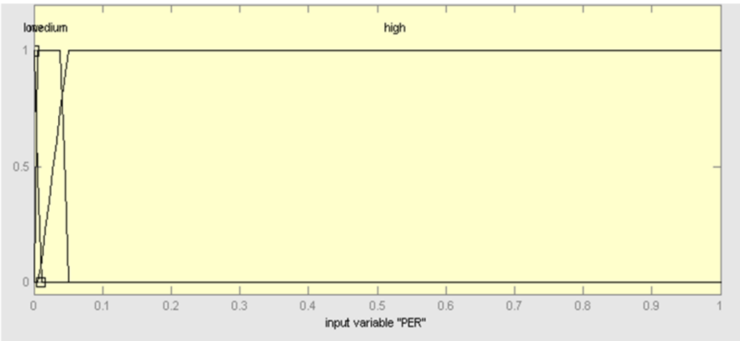


Figure 5-8: PER initial membership functions shape for the first configuration FL_1

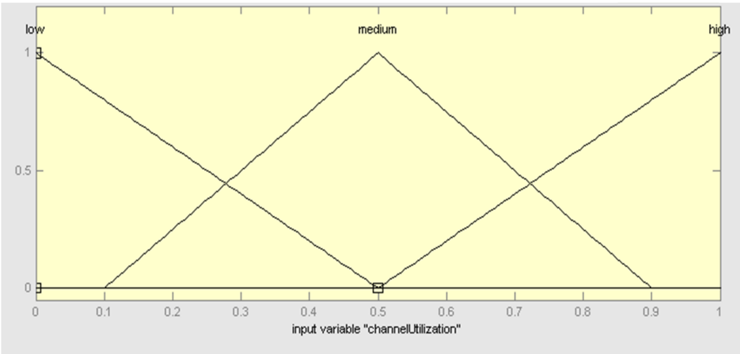


Figure 5-9: Channel Utilization initial membership functions shape for the first configuration FL_1

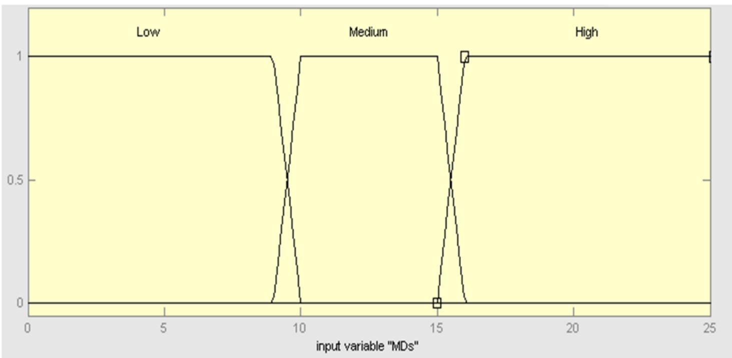


Figure 5-10: Number of Associated Terminals initial membership functions shape for the first configuration FL_1

The fuzzy logic based situation perception scheme has been evaluated using an extended experimentation analysis for measuring the performance of the situation perception mechanisms. The aim of the experimental assessment process is to validate the effectiveness and efficiency of the fuzzy logic situation perception on real life problems and highlight its scalability. The obtained results prove the suitability and viability of our fuzzy logic based situation perception for network management problems in the context of future Internet networks.

The experimental analysis was carried out for the problem of load identification, which is under the coverage and capacity optimization umbrella. If such a problem appears (high load) then they collaborate in order to select the most appropriate configuration action, which in this case is the optimal reallocation of the associated terminals among the available homogeneous or heterogeneous access points in the corresponding network area. The load in injected by the video stream that the terminals associated at the WiFi APs consume.

The experimentation is based on a deployment of six IEEE 802.11 Soekris access points (AP) and an IEEE 802.16 base station (BS) [164], each embedding an NEC implemented in Java [165]. Soekris devices are low-power, low-cost, advanced communication computers that act as re-programmable 802.11 access routers [166]. Moreover, several single-RAT (i.e., WiFi) and multi-RAT (i.e., WiFi, WiMAX) terminals are located in the corresponding area, consuming a video service delivered by VLC-based service provider [167].

This testbed was employed in order to extract the dataset used in the experimental assessment. We collected 50.000 tuples; each tuple was described by three variables indicating the status of an access point at time tx, namely Packet Error Rate (PER) ranging in [0…1], Channel Utilization (CU) ranging in [0…1] and Number of Associated Terminals (AT) in [0…25]. 10% of the dataset has been sampled from the deployment of the testbed and has been manually labeled (Y_i labeling), while the rest has been artificially generated according to the distribution derived from the initial set. The resulting dataset consists of 6667 tuples marked as Load (True according to the algorithmic notation), 9956 marked as No Load (False) while the remaining 33377 correspond to the Medium Load case (Neutral).

The basis of the analysis is a pre-evaluation of the dataset, with a very strict set of rules, directly capturing the environment where the APs are placed. The evaluated dataset is called ground truth and is used only for evaluation purposes. Evaluated against this dataset, all the three fuzzy logic controllers' configurations performed well, considering ofcourse the fact that they have been generally configured. Table 5-6 presents the success rate (i.e., the correctly classified tuples) using the three different configurations. We observe that the more generic a configuration is, the lower success rate he achieves, which is understandable due to the fact that the configurations matches several environments.

Table 5-6: Classification success rate results for the three configurations of the fuzzy reasoners

	FL1	FL2	FL3
Classification Success Rate	65.64%	71.86%	75.40%

5.2.3 Fuzzy Logic based Cooperative Power Control

A popular technique, both in research and industry, for covering the requirement for increased coverage and capacity is the networks' densification, by the introduction of small cells [168]. However, the densification as a technique is related to interference problems and also energy waste. Thus, power control mechanisms applied in such networks, aim at optimizing the networks' capacity and coverage and at the same time at achieving interference mitigation, reducing power consumption and extending battery lifetime. The purpose is to have improved QoS for the users as well as having the optimum overall network's utility and reduced cost from the network operator's perspective. At the same time, the need for signaling reduction in the ultra dense networks makes imperative the use of a cooperative and distributed paradigm. This will also enable avoidance of selfish behaviors that lead to suboptimum solutions.

This section aims at presenting a Situation Perception mechanism for WiFi APs operating in an Ultra Dense Environment, where uncertainties may occur. The idea is to extend algorithms for cooperative power control coming from sensor networks' application field [169], [170], apply the solution in WiFi APs, and address the situation perception problem due to uncertainties that may occur in the network. In the following subsections initially the baseline reference algorithm for cooperative power control is briefly described. Then the extended Cooperative Power Control scheme with the Fuzzy Logic situation perception mechanism is described in details coupled with the functional decomposition of the considered scheme. Finally, the experimental results are presented also in details, for proving the merits of the introduction of such a mechanism.

Cooperative Power Control - Baseline Algorithm

The proposed Fuzzy Logic enhanced CPC algorithm is based on [169], [170]; both approaches propose a scheme for distributed interference compensation in Cognitive Radio that operates in license exempt spectrum bands, using transmission power adjustment methodologies. The initial solution concerns ad-hoc networks and is based on an information exchange scheme for the identification of the appropriate transmission power levels. Each independent node of the topology sets its power by considering individual information, as well as information related to the neighboring nodes. More specifically, a node sets its power level by considering its Signal to Interference plus Noise Ratio (SINR) and the interference caused to its neighbors. The main idea of this approach is to prevent users to operate in the maximum transmission power levels.

The authors assume a set of node pairs L that operate in the same frequency. The SINR for the ith pair is given below:

$$\gamma\left(p_i^k\right) = \frac{p_j^k \cdot h_{ii}}{n_0 + \sum_{j \neq i} p_j^k \cdot h_{ij}} \quad (5.1)$$

Where

- p_i^k: transmission power for user i on channel k
- h_{ii}: link gain between ith receiver and ith transmitter
- n_0: noise level (equals to 10^{-2})
- p_j^k: transmission power for all other users on channel k, assuming that $j \in \{1,2,\ldots,L\}$ and $j \neq i$
- h_{ij}: link gain between i^{th} receiver and j^{th} transmitter (Figure 5-11)

It is also assumed that the channel is flat-faded without shadowing effects. Since the channel is static, the only identified attenuation is the path loss h (channel attenuation or channel gain). Given that indoor urban environments are considered, the channel gain is $h_{ji} = d_{ji}^{-3}$, where d is the distance between the j^{th} transmitter and the i^{th} receiver.

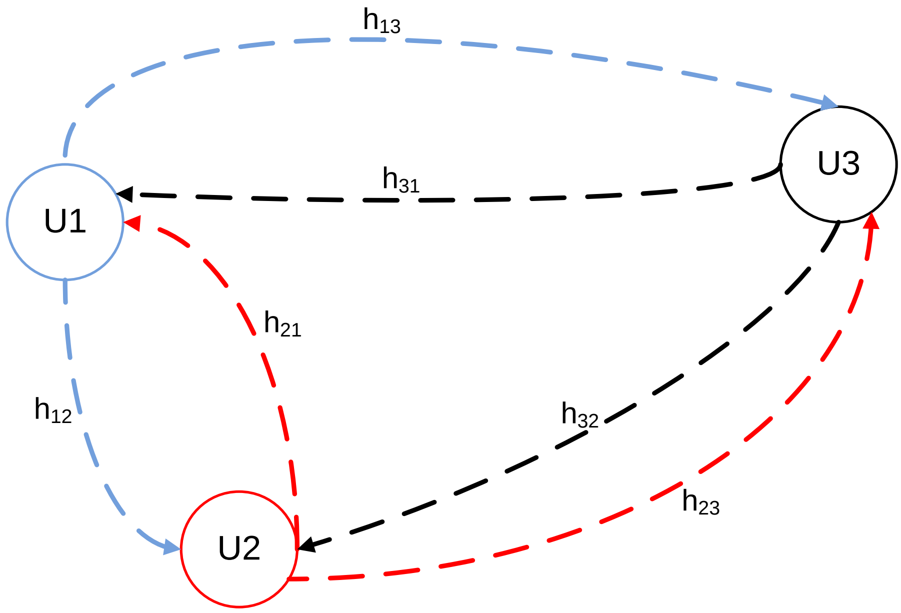


Figure 5-11: Representation of link gain among UEs in the network

The decision for the transmission power levels takes into account the negative impact (i.e., interference) of a node to its neighboring nodes. This is formalized using Equation 5.2, which captures the notion of interference price; such price reflects the interference a user causes to other users within its transmission range and is given by:

$$\pi_i^k = \frac{\partial u_i(\gamma_i(p_i^k))}{\partial(\sum_{j\neq i} p_i^k \cdot h_{ji}} \quad (5.2)$$

Where

- $u_i(\gamma_\iota(p_{ki})) = \theta_i \log(\gamma_\iota(p_{ki}))$: logarithmic utility function,
- θ_i: user dependent parameter.

Both of the algorithms presented in [169] and [170] are based on a tradeoff between the capacity of a node and the interference caused to the corresponding neighborhood. This balance is being captured by the following objective function:

$$u_i\left(\gamma_i(p_i^k)\right) - a \cdot p_i^k \cdot \sum_{j\neq i} \pi_i^k \cdot h_{ji} \quad (5.3)$$

The first part indicates a relation to the Shannon capacity for the corresponding user, while the second part captures the negative impact in terms of interference prices that a user causes to its neighborhood. The *α* factor is introduced so as to capture uncertainties in the network; these uncertainties reflect the precision of the received and compiled information of each network element regarding the interference price, which should have been available by the node's neighbors.

This is related to the fact that once a network element adjusts its transmission power, it informs its neighbors in an ad-hoc manner. This implies that even though a network element has collected information from all of its neighbors in order to adjust its transmission, the gathered data could be obsolete and, as a consequence, they will not capture neighborhood's current state. The obsolescence of the interference prices is related to the update interval (i.e., the periodic update) of each network element. In [169], α is set in a static manner as 25%. In [170], a fuzzy reasoner is introduced in order to identify, in a more dynamic way, uncertainties in the network based on the network's status; the inputs (number of users, mobility, update interval) of the fuzzy reasoner capture the volatile nature of the ad-hoc network, whereas the output of the fuzzy reasoner is the Interference Weight. The α factor is defined as 1/β Interference Weight + 1 (β has the maximum value of the Interference Weight).

The algorithm consists of three steps, namely, the initialization, the power update and the interference price update. The former is related to the assignment of initial valid transmission power and interference price values. The second part concerns the transmission power update based on the interference prices each node receives from its neighbors. Finally, the interference price update captures the communication of its interference prices to the neighborhood, by every network node. The second and the third steps are asynchronously repeated until the algorithm reaches a steady state (i.e., a state where every network element has the same transmission power for two consecutive time iterations).

Fuzzy Logic Enhanced Cooperative Power Control for WiFi APs

The considered solution is related to the incorporation of the previous schemes in an Ultra Dense Environment. In the case study under investigation, we assume the presence of several WiFi APs located in the considered area. These APs communicate via wireless links in order to exchange their interference values. Based on these values each network element adjusts its transmission power (Figure 5-12).

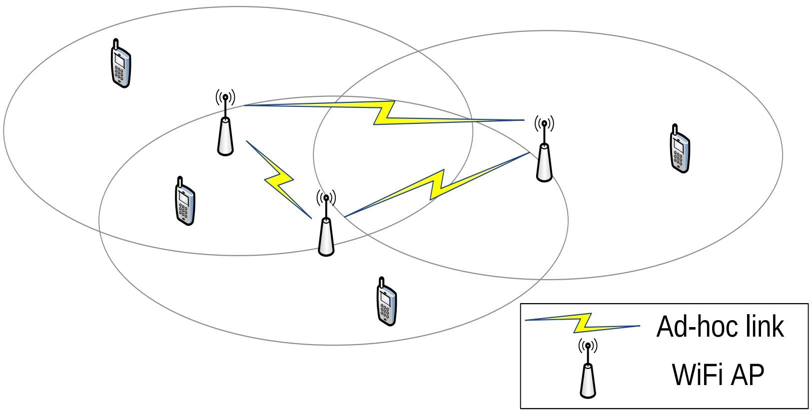


Figure 5-12: Envisaged network topology

Given the assumption that the APs communicate asynchronously and each one might have its locally-set update period, it is possible that the APs are unaware of the current network's status (from the messages exchange). Such problem becomes even more acute if we consider that the network elements might lose some messages during the messages exchange procedure due to the nature of the applied information fusion scheme and the sensitivity of the wireless medium. This implies that the use of the fuzzy reasoner is imperative for capturing the uncertainties [171].

The WiFi application area poses the need for modification of the inputs and the inference engine of the fuzzy logic controller. Thus, the number of the WiFi APs in the vicinity, the number of users in the vicinity (associated to WiFi APs) and the update interval are used as inputs of the fuzzy reasoner. In case of completely new application areas, new/modified fuzzy reasoners could be incorporated so as to be more suitable to the use case under discussion. The way a network element perceives its environment is based on the input and output membership functions. As in [170], the inputs' membership functions are set to have triangular shape, mainly in order to capture the strict nature of the inputs.

Table 5-7 provides the rules of the inference engine of the fuzzy reasoner. The most crucial input for the decision making process is the update interval. This input depicts the frequency of the information updates about the interference price of a network element to its neighbors thus capturing how recent is the view of a network element, based on the inputs from its neighbors. These inputs will be used for the calculation of the TxPower. Figures 5-13 to 5-15 present the interference weight (i.e., outcome of the fuzzy reasoner) as a function of:

- the APs' and the users' number, having as parameter the time interval (set 0.5 sec - Medium) – Figure 5-13,
- the update interval and the number of APs, having as parameter the users' number (set 25 – Low to Medium) – Figure 5-14,
- the update interval and the users' number, having as parameter the number of APs (set 15 – Medium to High) – Figure 5-15.

From the figures we may observe that interference weight (capturing the uncertainties) is related to either larger number of users and APs in the networks (which may cause uncertainties) because they introduce dynamics in the network, or slower update interval (which implies that the network elements will not have the latest network view).

Table 5-7: Correlation table between inputs and outputs for the fuzzy reasoners in the Cooperative Power Control

Rule Number	Num of WiFi APs	Num of Users	Update Interval	Interference price
1	Low	Low	Low	Low
2	Low	Low	Medium	Low
3	Low	Low	High	Medium
4	Low	Medium	Low	Low
5	Low	Medium	Medium	Medium
6	Low	Medium	High	Medium
7	Low	High	Low	Medium
8	Low	High	Medium	Medium
9	Low	High	High	High
10	Medium	Low	Low	Low
11	Medium	Low	Medium	Medium
12	Medium	Low	High	High
13	Medium	Medium	Low	Medium
14	Medium	Medium	Medium	Medium
15	Medium	Medium	High	High
16	Medium	High	Low	Medium
17	Medium	High	Medium	Medium
18	Medium	High	High	High
19	High	Low	Low	Medium
20	High	Low	Medium	Medium
21	High	Low	High	High
22	High	Medium	Low	Medium

23	High	Medium	Medium	Medium
24	High	Medium	High	High
25	High	High	Low	Medium
26	High	High	Medium	High
27	High	High	High	High

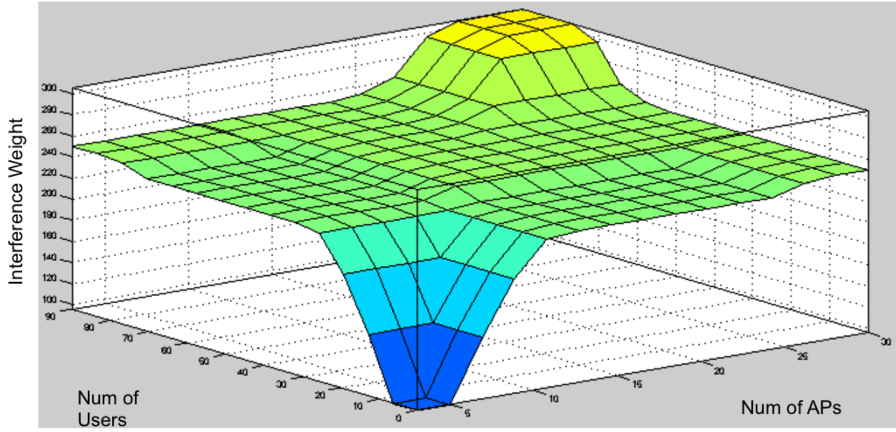


Figure 5-13: The interference weight as a function of the APs' and the users' number, having as parameter the time interval (set 0.5 sec - Medium)

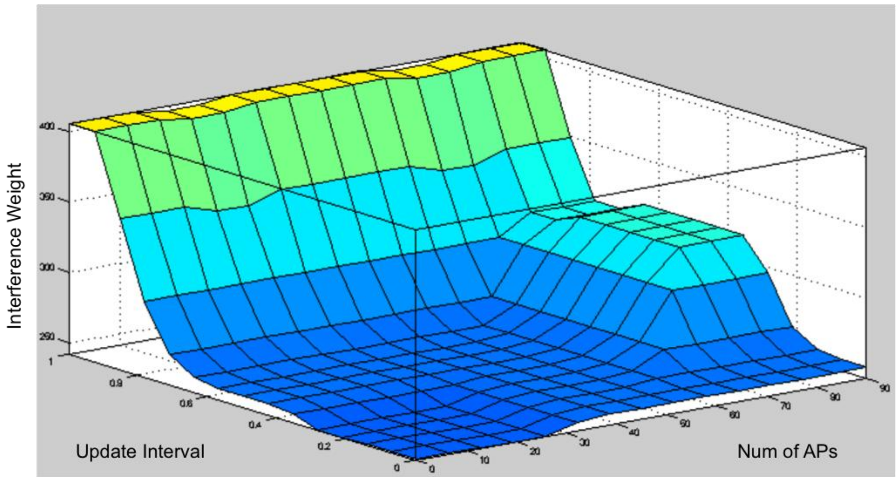


Figure 5-14: the interference weight as a function of the update interval and the number of APs, having as parameter the users' number (set 25 – Low to Medium)

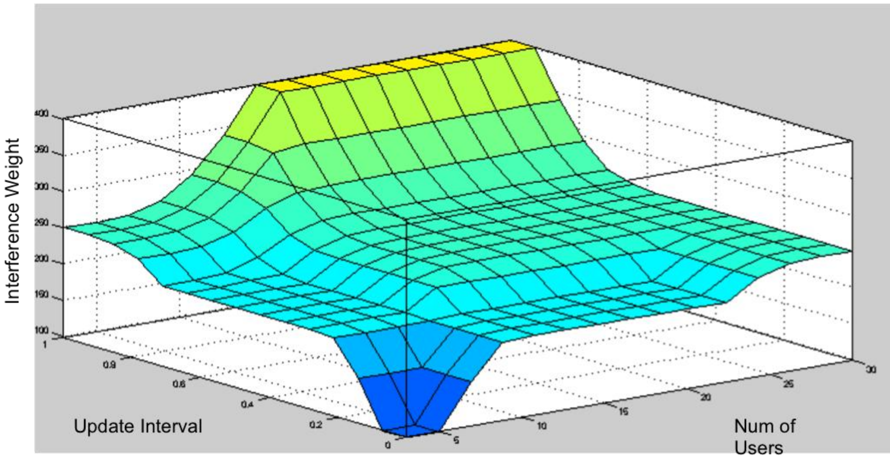


Figure 5-15: the interference weight as a function the update interval and the users' number, having as parameter the number of APs (set 15 – Medium to High)

The CPC consists of two separate iterative procedures, the power update and the interference price update. In the former, consider a network element i, which updates its transmission power using a time interval $t_{ai} \in T_{ai}$, where T_{ai} is a set of positive time instances in which the AP i will update its transmission power level and $t_{a1} \neq t_{a2} \neq \cdots \neq t_{ai}$. Similarly, each WiFi AP i has an interference price update interval $t_{bi} \in T_{bi}$, where T_{bi}, where it updates its interference price and announces the updated interference price π_i^k to the rest of the WiFi APs belonging in the scheme. Figure 5-16 provides the messages exchange and the operations' sequence on a scheme with two WiFi APs; this could be generalized for more APs as well.

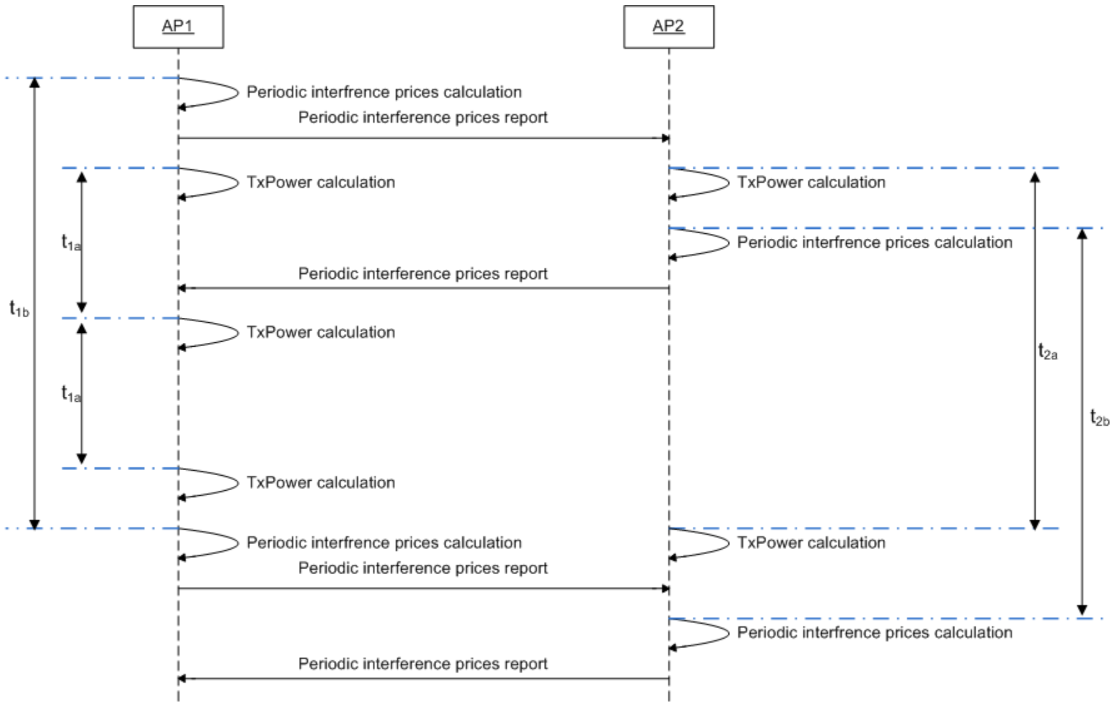


Figure 5-16: Messages exchange on a scheme with two WiFi APs

In order to deploy the CPC in the considered environment, network elements should be enhanced with a set of software modules namely “Power Control”, the “Learning” (being analyzed in the following chapter), the “Memory”, the “CPC communication”, the “Control Engine” and the “Monitoring”. Figure 5-17 presents the functional decomposition of the CPC.

Each software module provides a set of functionalities in order to enable the instantiation of the CPC in WiFi APs; more specifically:

- The “Power Control” incorporates the functionalities for the calculation of the metrics (interference prices) and the objective function that each network element has to maximize. Furthermore, this part of the mechanism implements the fuzzy logic reasoner for the calculation of the interference weight and the α factor,

- The “Learning” part incorporates the learning mechanism for enhancing the network element’s situation perception that will be analyzed in Section 5.

- The “Memory” contains all the information required for the CPC; this information may be local and related to the AP under consideration (ex. TxPower, SINR, local IPs and MACs etc.), or related to neighboring network APs (physical topology information – distances from neighbors, network information – neighbors' IPs and MACs, algorithm information – neighbors' interference prices and TxPowers).

- The "CPC communication" software module consists of two parts, the client and the server. As mentioned afore, the basis of the CPC scheme is related to the asynchronous information exchange among the network elements. This implies that each network element operates as a server, where the neighboring WiFi APs are being associated and also as a client in order to associate to the neighboring APs.
- The "Control Engine" is responsible for the enforcement of the re-configuration action, which in the considered case is the TxPower adjustment.
- The "Monitoring" software module is responsible for the two types of monitoring tasks, the local and the neighborhood/cluster. The former is related to monitoring of local metrics and measurements (e.g., identification of local TxPower, associated users, sensed APs etc.) whereas the latter is related to cluster information (e.g. MACs and IPs of neighboring APs, physical topology graph, etc.).

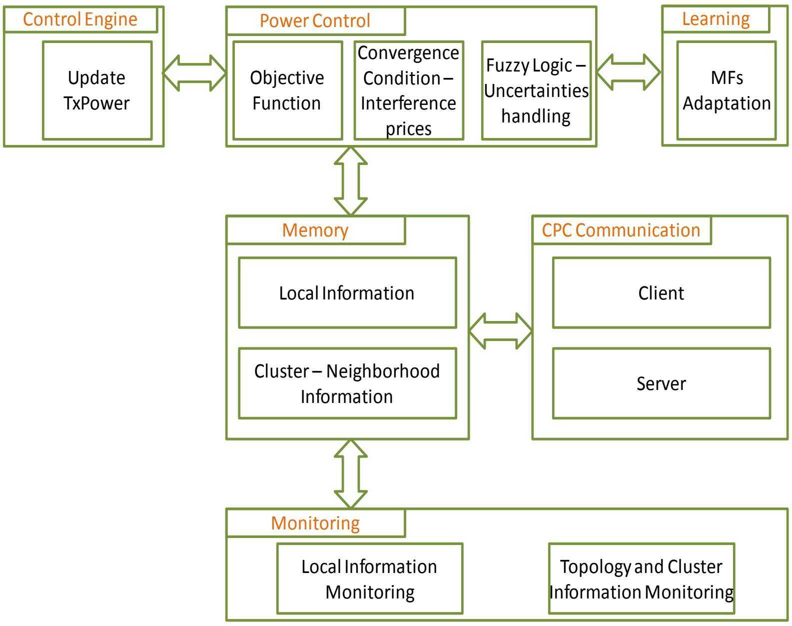


Figure 5-17: Functional architecture of the CPC

Deployment and Environment Description

In order to experiment with the fuzzy logic enhanced cooperative power control a series of real life experimentations has been performed. The experimentation is based on a proof of concept which instantiates the architecture described above.

Environment Description

For the experimentation a set of Soekris devices has been used; such devices are low-power, low-cost, Linux-based communication computers (500MHz AMD Geode LX, 512MByte DDR-SDRAM) that act as re-programmable WiFi APs by using IEEE 802.11b/g radio access technology [166]. In all Soekris devices two wireless interfaces are installed, one is the actual AP interface and the other one is used for monitoring; the former is the AR5413 mini-PCI [172] Card whereas the latter is the WUSB54GC USB card [173]. The APs deploy their own network and route the information to the internet through NAT. APs are connected through the backbone network and communicate with a standalone machine which aggregates information and provided triggers for the initiation of algorithms. The CPC implementation is based on Java programming language using several external libraries. The most important of them are the jFuzzyLogic [174] for the "Power Control" module and Apache MINA [175] for the "CPC communication" module. For the "Monitoring" module the Linux kernel utilities are exploited.

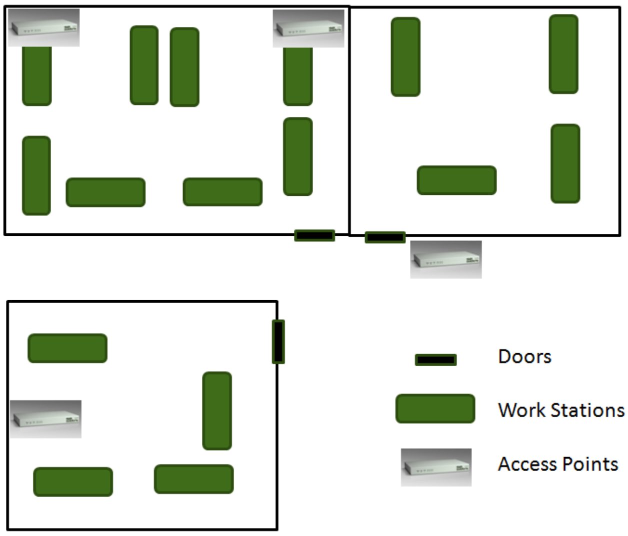


Figure 5-18: Physical topology of the experimentation environment

Four Soekris devices have been placed in a typical small office environment consisting of three rooms, with 15 employees (Figure 5-18). The employees used these APs for two consecutive days for 10 hours each day (from 10:00 CET until 20:00 CET on July 9th 2012, where our algorithms are not installed and the measurements are used for extracting the control data, and on July 10th where the fuzzy logic enhanced Cooperative Power Control algorithm operated for the transmission power control) in order to access the internet and perform all normal, working-day, activities. Overall traffic throughout the day ranged from 1 to 10 Mbps while APs were configured to operate at 5.5 Mbps throughput. The network layout is depicted in Figure 5-19.

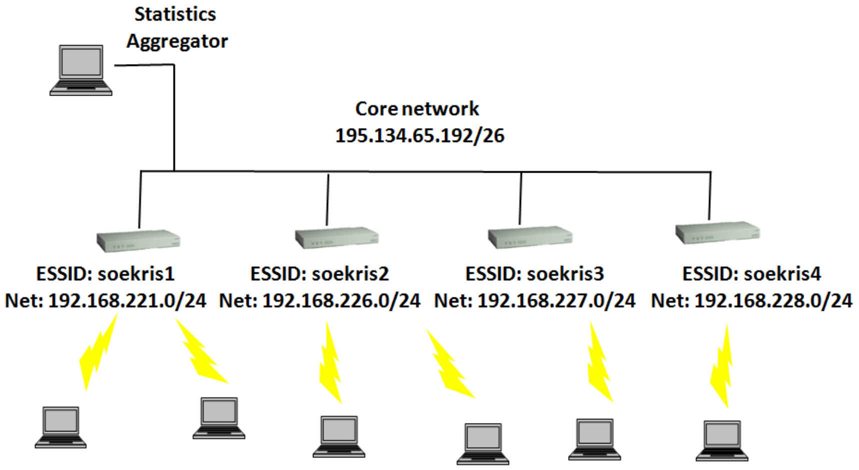


Figure 5-19: Network topology of the experimentation environment.

Of the two days of the experiment, the one has been used for the control data generation and the second has been sued for the evaluation of the fuzzy logic enhanced CPC was embedded in the Soekris devices. In both days it has been attempted to procedure almost identical experimental conditions. The bandwidth requirements were reproduced – however user's mobility could not be identically reproduced. It should be noted that the two days of the experiment could be characterized as follows:

a) The first day, July 9th 2012, the control data were built, using the Soekris devices operating with their maximum transmission power.

b) The second day, July 10th 2012, the Soekris devices operated having embedded the Fuzzy Logic enhanced Cooperative Power Control

Assumptions

As mentioned afore, the CPC scheme is based on the assumption that it will operate on an urban area. Thus, the generic assumptions of the algorithm should be also adapted accordingly.

The WiFi APs are placed in an indoor environment and communicate via specific communication interfaces. This implies that the distance among the network elements needs to be defined. In the proposed approach, the methodology of [176] and [177] is being followed.

The propagation obeys to certain models, from which the log-distance model is one of the most simple; the following equation describes the behavior of such model:

$$\log d = \frac{1}{10 \cdot n}(P_{TX} - P_{RX} + G_{TX} + G_{RX} - X_a + 20\log\lambda - 20\log(4\pi)) \quad (5.4)$$

Where

- $d(m)$: the estimated distance between the transmitter and the receiver,
- $P_{TX}(dBm)$: the transmitted power level,
- $P_{RX}(dBm)$ is the power level measured by the receiver,
- $G_{TX}(dBi)$: the antenna gain of the transmitter,
- $G_{RX}(dBm)$: the antenna gain of the receiver,
- n: measure of the influence of obstacles like partitions and ranges from 2-5 (2 for free space, 4-5 in case obstacles are considered),
- $X\alpha$: normal random variable with standard deviation of α. This variable captures the variance of the fading phenomena in an indoor environment,
- λ: the wavelength of the signal (for WiFi can be considered 0.12m).

In the proposed experimentation, and for a typical office environment, n has been set to 5 and $X\alpha$ to 20. Regarding the transmission power, which is the actual parameter of our implementation, it is related to the equipment's capabilities. Specifically, $TxPower$ is limited by the WiFi card's capabilities; 10dBm is the lowest price whereas 27dBm is the highest.

<u>Experimentation Analysis</u>

Figures 5-20 to 5-28 capture the experimentation results for the first day (July 10th 2012). The experiment has started 10:00 CET and has finished 20:00 CET. The four Soekris APs have been placed in our testbed and we have been measuring for this period the transmission power of their WiFi cards; the transmission power ranges from 10 to 27 dBm. Figures 5-20 to 5-23 present the transmission power for the 10 hours of the experiment. In order to evaluate the operation of the network for several topologies, initially we have all four Soekris operating, whereas as the experiment proceeds we turn them off one by one and we leave only one operational. For each of the Soekris devices (and considering that the 10dBm is the basis of the $TxPower$ for each AP) we see the actual gain compared to setting the transmission power to the maximum $\mathrm{TxPower}$ (i.e., 27 dBm). The energy gain at each of APs 1 (Figure 5-20), 2 (Figure 5-21), 3 (Figure 5-22), and 4 (Figure 5-23) is 12.51%, 10.75%, 33.33% and 21.23% respectively. Also, it is obvious that the more the APs, the more energy gains we have, due to the collaborative nature of the algorithm. Also, what should be noticed is the fact that the APs change very often their $TxPower$ levels. This is related to the highly volatile office environment, with moving users and the many interference sources (i.e., moving users, cell phones, Bluetooth devices, etc.), in relation to the fact that the APs identify the network topology considering indoor path loss models. Such models, if we assume static environments, without moving users operate with accuracy, however in the case under discussion, the network elements need to calculate the topology on a constant basis, in every CPC loop.

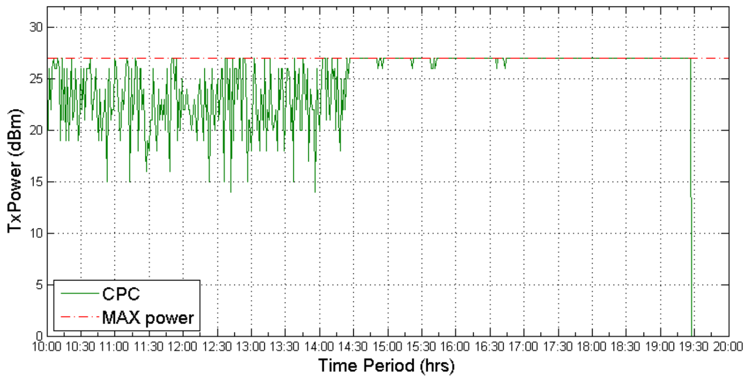


Figure 5-20: Transmission power adjustments in the four Soekris APs using the Cooperative Power Control scheme in Soekris AP1

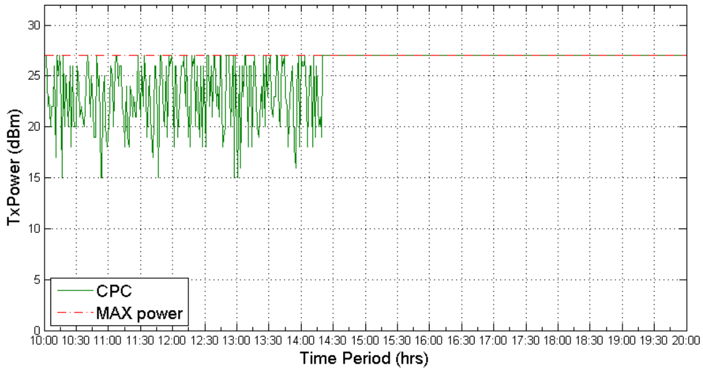


Figure 5-21: Transmission power adjustments in the four Soekris APs using the Cooperative Power Control scheme in Soekris AP2

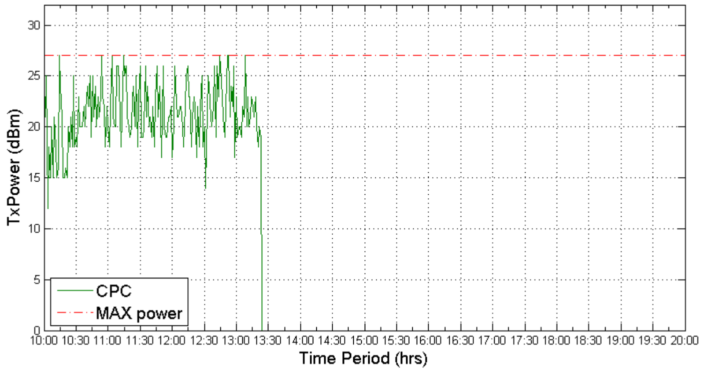


Figure 5-22: Transmission power adjustments in the four Soekris APs using the Cooperative Power Control scheme in Soekris AP3

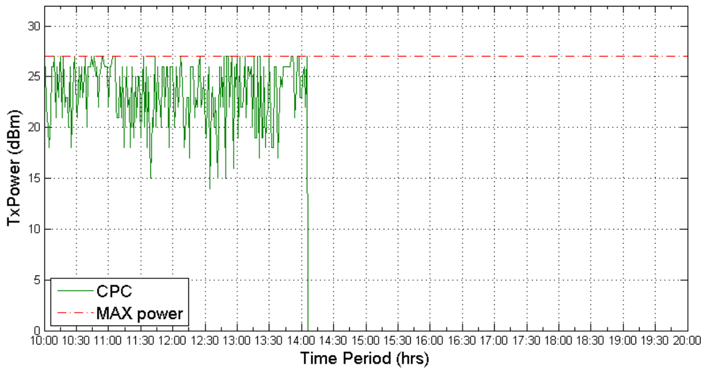


Figure 5-23: Transmission power adjustments in the four Soekris APs using the Cooperative Power Control scheme in Soekris AP4

Figures 5-24 to 5-27 provides the 6th degree polynomial function of the SINR measurements during the experimentation. At any case, the SINR is better compared to the case where maximum $TxPower$ has been set to the APs. For the AP 3 and 4 the experiment stops at the time that these APs are being turned off (13.20 and 14:20 respectively) and we see that when all four Soekris operate, the SINR to all of them is low. When we start turning off AP we observe that the SINR to all the operating ones starts increasing; this is related to the fact that the interference that is caused reduces as well. Finally, only one, AP 2, remains operational and we have a huge increase in the SINR, which has started when we turned off AP3 and AP4; however we should take under consideration that the overall capacity reduces.

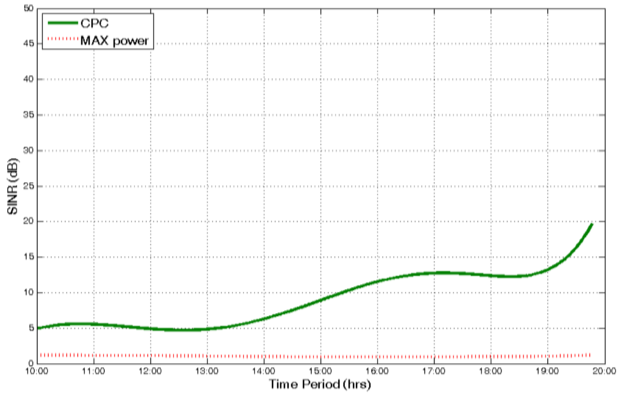


Figure 5-24: SINR evolution during the experimentation period for Soekris AP1

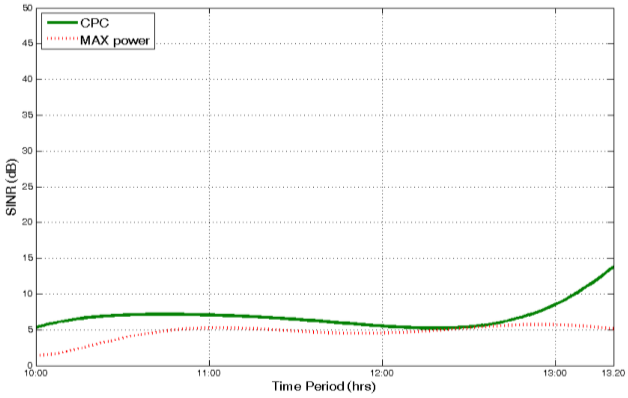


Figure 5-25: SINR evolution during the experimentation period for Soekris AP2

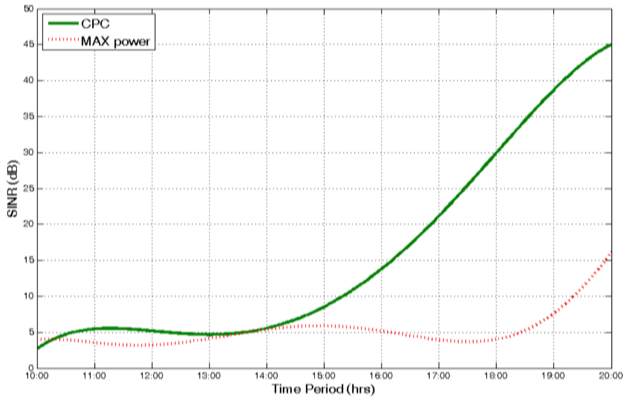


Figure 5-26: SINR evolution during the experimentation period for Soekris AP3

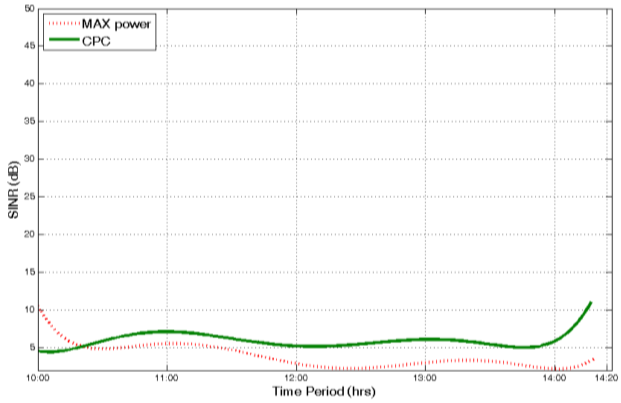


Figure 5-27: SINR evolution during the experimentation period for Soekris AP4

Figure 5-28 presents the number of iterations every time the CPC is being triggered. We consider that the CPC is being triggered periodically, every 5 minutes. The Soekris APs exchange messages asynchronously; everyone using its own intervals. We observe that the scheme converges in small number of iterations most of the times (mean value of iterations 3.876).

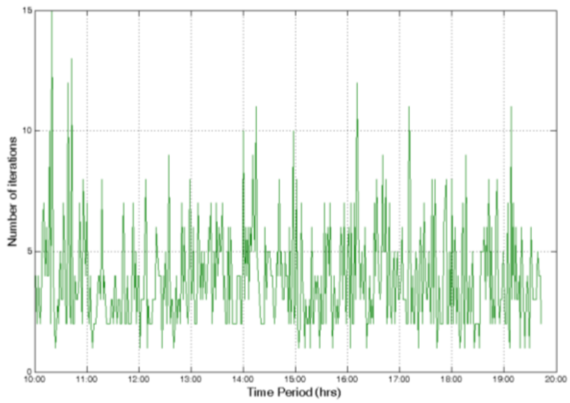


Figure 5-28: Number of iterations every time the CPC is being triggered

6. Learning Enhanced Situation Perception

The Situation Perception schemes aim to identify problematic situations or optimization opportunities and in general perform well in the environments where they are built to operate. On the other hand, if the network conditions change/evolve, or the network elements get re-located in a totally new environment, they do not manage perform their situation perception task in a satisfactory manner, without being manually configured by the network administrators. This adaptability (described as self-mutation in Section 2.4) is a key requirement of the self-managed networks for enabling them to autonomously evolve their reasoning schemes for meeting the new context.

Fuzzy logic enhanced situation perception faces the same problems regarding their adaptability as the rest of the situation perception schemes in the literature. Thus, new schemes for enhancing the way the network elements model their environment have been developed and are being presented in this thesis. More specifically, two learning approach schemes are being proposed, one based on supervised learning and one on unsupervised learning. For the unsupervised learning scheme several variations of the solution have been proposed, depending on effect they have to the fuzzy logic reasoners.

The rest of this section is structured as follows. Initially, the reference (learning) problem is being described. Afterwards the learning schemes are being described in Section 6.2. Finally, in Section 6.3 the learning schemes are being applied in the situation perception schemes that have been developed and described in Section 5.

6.1 Reference Problem Description

In order to identify the direction to move towards, we have defined the key points of the problem to be addressed; such description will enable the formalism of the solution on the one hand and the generalization of the proposed solution to every problem that could be formulated the same way on the other hand.

We consider that the decision maker has several states (related to network measurements); let $s \in S$ be the network state and S the network state space.

Suppose that each decision making mechanism i is defined by a set of parameters Θ_i and $U_i: S \to R$ its utility.

We consider a set of decisions that we are definite about its validity; let it be named ground truth *G*. At each given time step, t, if $|G(s_t) - U(s_t)| \leq \varepsilon, \forall \varepsilon > 0$ then $U(s_t)$ is considered a "correct decision".

Also let N_{ij} be the number of the occasions where the decision of the decision making mechanism is $U_i(x_i)$ and the ground truth is $G(x_j)$. Table 4-1 summarizes all the cases for the given states and the identified decisions. Intuitively, what such table is capturing is that the diagonal contains the numbers of the correct decisions whereas the rest of the positions of the table host the wrong ones. We define as adaptation mechanism *A*, a mechanism that calibrates the parameters Θ_i so as to maximize the "correct" decisions and "minimize" the wrong ones:

$$max\left\{\sum_{i=1}^{K}\sum_{j=1}^{K} N_{ij,i=j}\right\} \quad \text{or} \quad min\left\{\sum_{i=1}^{K}\sum_{j=1}^{K} N_{ij,i\neq j}\right\} \qquad (6.1)$$

It is worth noting that the ground truth in this work is used only for the evaluation of the overall decision-making and adaptation mechanisms. In other words, what is meant is that the scheme does not assume knowledge of the ground truth during the learning and adaptation process. Such an assumption would have been unrealistic and would contradict the autonomic nature of the scheme, which is needed for any scheme to be operational during runtime in any real network environment.

Table 6-1 Decisions versus ground truth

	$U_i(s_1)$	…	$U_i(s_k)$
$G(s_1)$.	N_{11}	…	N_{1k}
…	…	…	…
$G(s_k)$.	N_{k1}	…	N_{kk}

6.2 Learning Enhanced Fuzzy Logic

As thoroughly described in section 4.2, the learning schemes may be categorized according to the availability of labels in the dataset. The labels capture the state of the input vectors. More specifically, supervised learning schemes attempt to identify patterns and structures using a dataset with labeled data, whereas unsupervised learning identify patterns and hidden structure of a set of unlabeled data. In terms of this thesis, both approaches have been followed and evaluated. In the following sections, both schemes are being described.

6.2.1 Supervised Learning Algorithm

The high level description of the supervised learning algorithm is a decentralized one, with parts of the algorithm being implemented in the NEC level and others being implemented in NDC level. Figure 6-1 provides the high level description of the learning scheme. Initially, each network element monitors its environment for identifying problematic situations using the Situation Perception fuzzy reasoners. In the case that problematic situations are being identified the NEC proceeds in problem solving decisions and the corresponding execution of such decisions. Afterwards, the learning procedure takes place, which consists of three distinct phases, namely:

- the labeling phase,
- the classification step, and,
- the fuzzy logic enhancement procedure.

Throughout the presentation we assume that each time a network device (i.e. NEC) monitors its operational environment it extracts a d-dimensional vector which can be classified as True, indicating that a particular problem has appeared, False – no problem- or Medium/Neutral, implying that although there is currently no problem there is a chance that a problematic situation may appear in the future. Additionally, given a problem, the device triggers a remedy action, which is guaranteed to solve the problem; in other words it will enable the device to transit from a True state to either a False or a Neutral state. Obviously, upon start-up, the device has no pre-installed knowledge base, apart from the set of fuzzy logic rules and the set of configuration actions.

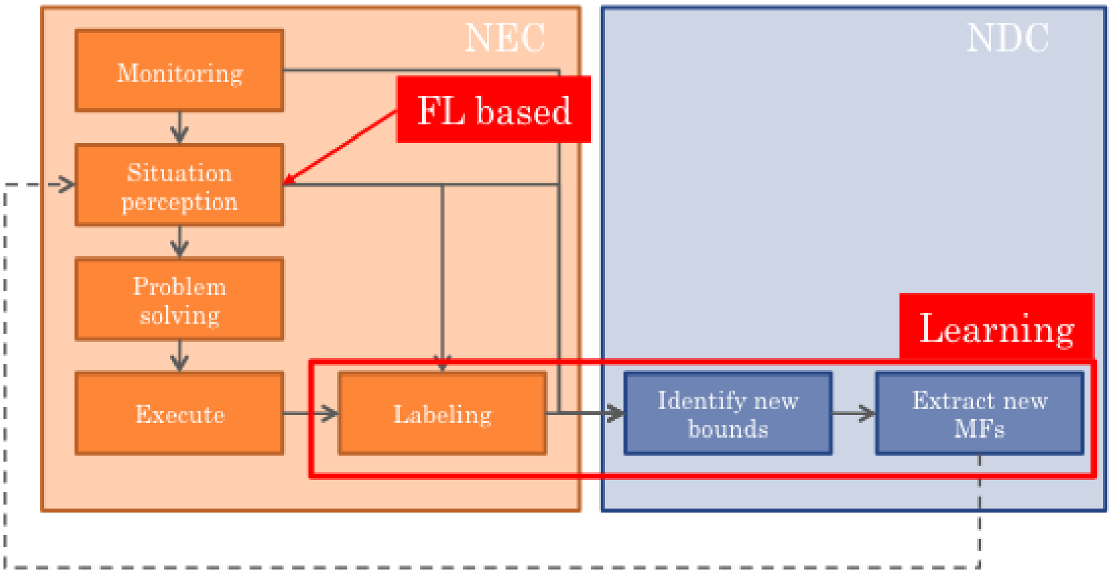


Figure 6-1: High level view of the supervised learning scheme

The labeling algorithm appears in Table 6-2. The device monitors its operational environment and extracts a d- dimensional tuple (step 4.2), which is evaluated against the set of pre-installed fuzzy logic rules (step 4.3). At this step, the inference process of the NEC for the situation perception (fault or optimization opportunity identification) takes place. If the outcome denotes a problematic situation (step 4.4, label X_i) then the appropriate solution is applied (step 4.4.1). Each cognitive manager per network has a set of solutions or configuration actions that could be enforced. Each problem is associated with one or more solutions. According to the global status of the network device, and possible interactions among different solutions the most appropriate is selected. Given the fact that the applied solution will always solve the problem we compare the $(i+1)^{th}$ tuple with the i^{th} (step 4.4.3). In case their distance is less than a predefined bound ε we assume that we performed a classification error (false positive, i.e., we classified a Neutral tuple as true) and attribute the correct label Y_i. Y_i corresponds to the actual conditions (ground truth), while label X_i to the fuzzy logic perception of the environment. In any other case, we cannot decide about the label and leave it as it is. As soon as a significant amount of vectors (i.e., N) is aggregated we halt the procedure and proceed with the application of the k-NN classifier.

Table 6-2: Labeling algorithm on the network element level

Input:	**Approximation Parameter ε**
Output:	**Set of labeled observations S, Set of unlabeled observations T**

1.	S $\leftarrow$ O
2.	T $\leftarrow$ O
3.	i=0
4.	while true
4.1	i++
4.2	Retrieve vector $Z^{\rightarrow}_{i}$
4.3	$X_i \leftarrow$ fuzzy logic ($Z^{\rightarrow}_{i}$)
4.4	if (X_i = True)
4.4.1	Select and Apply appropriate Solution
4.4.2	Wait for S_{i+1}
4.4.3	if($\lVert S_{i+1}-S_i \rVert < \varepsilon$) $\Rightarrow$ Y_i= Neutral
4.4.4	else Y_i = True
4.4.5	S = S U {$Z^{\rightarrow}_{i}$, X_i, Y_i}
4.5	else if (X_i = Neutral / False)
4.5.1	Y_i = ?
4.5.2	T = T U {$Z^{\rightarrow}_{i}$, X_i, Y_i}
5	return S, T

The k-NN classifier enables the identification of all missing Y_i labels. The set of labeled instances (S) is used as the training set, while all unlabeled records (T) are used as testing set. Recall from the last step that although we can accurately predict the labels of all observations appearing in S, we only have tuples from Neutral and True. In order to overcome this, we artificially generate a small number of tuples, which are in advance labeled as False (i.e. all tuples are located in the beginning of the coordinates systems axes). It should be pointed out that this step appears only the first time that the algorithm is executed. In subsequent executions, the training set is populated with previously labeled records. The algorithm appears in Table 6-3. The successful execution of this step essentially generates a set of correctly labeled observations, which can be used in order to quantify the quality of the procedure.

Table 6-3: k-NN classification for the extraction of the missing labels

Input:	**Set of labeled observations S, Set of unlabeled observations T**
Output:	**Final set of labeled observations F**
1.	F $\leftarrow$ O
2.	kNN.training set $\leftarrow$ {S}
3.	kNN.test set $\leftarrow$ {T}
4.	F $\leftarrow$ kNN(Training, Test)
5	return F

At this point it should be noted that the kNN classification could be replaced with alternative classification methods, such as the Support Vector Machines. However, such approach would increase the complexity of the overall scheme, due to the fact that, in general, the tuples cannot be captured by simple SVM classification. This implies that we should proceed in complex transformations and increase the dimensionality of the input tuples. Even though the previous scheme can also successfully classify input data, the limited processing power of the NECs - where classification takes place- prohibits the use of such approach.

Periodically the stored tuples are validated in order to provide an indirect assessment measure with respect to the quality of the fuzzy logic rules. The evaluation is done according to the following formula:

$$A = \sum_{\iota=1}^{N} \frac{|X_\iota - Y_\iota|}{N} \quad (6.2)$$

A essentially quantifies the percentage of cases that we made an erroneous decision (i.e., the ground truth label Y_i is different than the fuzzy logic label X_i) and is compared with a predefined tolerance bound A_p. If the number of mistakes is not tolerable ($A > A_p$) then the network element sends all data to the domain controller (NDC).

The domain controller receives data from all network elements for which condition ($A>A_p$) holds true. All measurements lay in a d-dimensional space and by exploiting the ground truth labels (Y_i) we can categorize them in three distinct classes $C_i \in \{True, Neutral, False\}$ which correspond to three high dimensional manifolds (D_T, D_N, D_F). For ease of presentation, in the context of this work, we will assume that data points form hyperspheres, however the work can be extended to address a multitude of high dimensional manifolds.

Each sphere is centered at $CE_i = \sum_{j=1}^{|C_i|} \frac{S_j}{|C_i|}, S_j \in C_i$ and has radius $R_i = \max_{j=1,|C_i|} \left\| CE_i - S_j \right\|$. We assume that the cluster of tuples labeled as True is centered at $CE_t = (x_1, x_2, \dots, x_d)$ and has radius R_T, while tuples corresponding to Neutral are centered at $CE_N = (y_1, y_2, \dots, y_d)$ with radius R_N. Similarly, False points are centered at $CE_F = (Z_1, Z_2, \dots, Z_d)$ with radius R_F. Without loss of generality we consider only points CE_T and CE_N. These two points define a line ε, which is described by the following set of equations:

$$p_i = x_i + u \cdot (y_i - x_i), i = 1 \dots d \quad (6.3)$$

This line intersects with spheres D_T and D_N in four points, which can be retrieved by substituting the pi values into the following hypersphere equations:

$$D_T \rightarrow \sum_{i=1}^{d}(p_i - x_i)^2 = R_T^2 \quad (6.4)$$

$$D_N \rightarrow \sum_{i=1}^{d}(p_i - y_i)^2 = R_N^2 \quad (6.5)$$

Consequently, a simple way of identifying the bounds for the fuzzy logic rules would be to extract the intersection points which

- belong to different hyperspheres and
- exhibit minimum distance from each other.

Simply stated, the two intersection points are provided by:

$$\{B_1, B_2\} = min\{\|P_{1T} - P_{1N}\|, \|P_{1T} - P_{2N}\|, \|P_{2T} - P_{1N}\|, \|P_{2T} - P_{2N}\|\} \qquad (6.6)$$

By applying a similar procedure we can also extract points B_3 and B_4 from spheres D_N and D_F. Notice at this point that each B_i corresponds to a tuple $\{b_1, b_2, \ldots, b_d\}$; thus each B can be directly set as a new bound for the fuzzy logic rules. More precisely:

- Point B_1 would correspond to the bound for the True-Neutral situation, in other words, B_1 is a point labeled as True that exhibits maximum distance from CE_T and is closest to CE_N than any other point P labeled as True.
- Point B_2 would correspond to the upper bound for the Neutral-True situation. Similarly B_2 is labeled as Neutral and exhibits maximum distance from CE_N and is closest to CE_T than any other point labeled as Neutral. Since D_T and D_N are adjacent, B_1=B_2.
- Point B_3 would correspond to the lower bound for the Neutral-False situation
- Point B_4 corresponds to the bound for the False-Neutral situation (obviously B_3=B_4)

A graphical representation of this procedure appears in Figure 6-2 where the application of the proposed scheme is demonstrated the application of our algorithm in R^3.

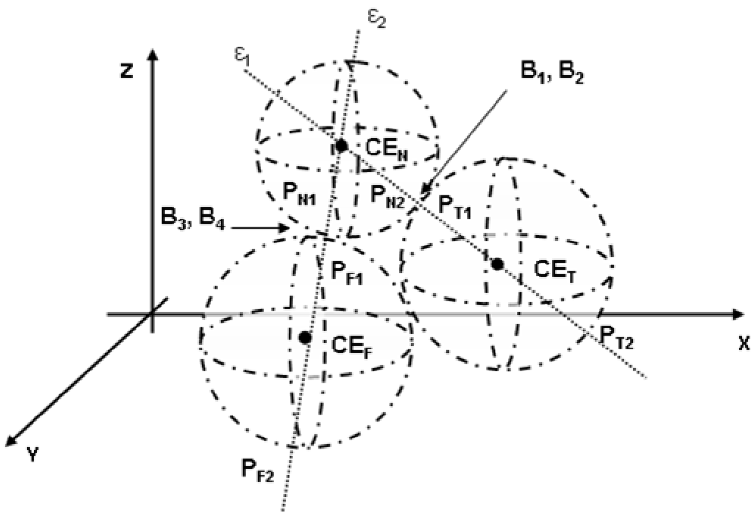


Figure 6-2: Graphical representation of data records according to ground truth labels

It should be stressed out that hyperspheres comprise an abstraction of the actual manifold formed by the data points so the actual manifold is enclosed in the hyperspheres. However, this abstraction fits our purposes for two reasons. Firstly, its computation is simple and fast thus it can be implemented on any kind of device without imposing any memory or CPU overhead. Secondly, the circumference of the hypersphere will contain at least one point of the class under process, thus indirectly signifying the range of values of that class.

On the other hand, the adjacency of the spheres may yield poor discrimination quality, in the sense that a lot of *True* and *Neutral* cases as well as *Neutral* and *False* cases may have been placed together. The latter, is due to the fact that the hyperspheres enclose a larger area than the actual manifold. In order to overcome this we employ a hierarchical divisive clustering (HDC) approach based on *k*-means. An indirect gain however, is that the application of HDC will take place on the D_T and D_F spheres and not on the larger, D_N sphere. Essentially, the fitting of data into three spheres comprises a fast implementation of the first step of HDC in O(N) time which is significantly smaller than the O(eN) requirement of k-means.

Afterwards, k-Means is applied on the two spheres, which correspond to *False* and *True* and direct the algorithm to split it into two clusters, *False* or *True* and *Neutral*. The result will be two adjacent spheres maintaining elements belonging to both classes. The new sphere corresponding to *Neutral* is merged with the initial *Neutral* class. The division continues on the resulting *True* and *False* clusters until we start experiencing loss in the *Recall* (Recall = Retrieved Relevant

Records / Total Relevant Records) or high *Precision* (Precision = Retrieved Relevant Records / Total Retrieved Records) in conjunction with high *Recall* (high F-measure value, where F-measure = 2*Recall*Precision / Recall + Precision). The geometric interpretation of our approach is depicted in Figure 6-3. The algorithm simply augments the sphere corresponding to Neutral cases and shrinks the other two by extracting falsely classified points. When the procedure is halted then we have three overlapping hyperspheres. The intersecting points of the line defined by the spheres' centers with the spheres correspond to the desired solution.

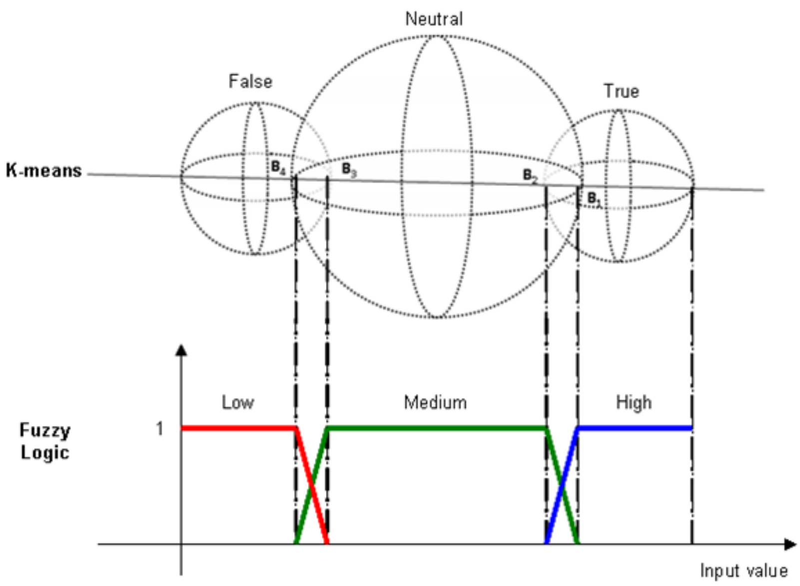


Figure 6-3: Geometric interpretation of the approach

6.2.2 Unsupervised Learning Algorithm

The unsupervised version of the learning enhanced fuzzy logic situation perception scheme is based on the modification/adaptation of the previously described solution, following the same generic principles. The key difference of the unsupervised scheme lies at the skip of the complicated algorithm for labeling, following the assumption that the network administrator will have, in general, configured the network elements to operate adequately; thus enabling the system to converge. Compared to the previously described scheme, which is based on a two layer approach where initially the ground truth is being built by using classification approaches (k-nearest neighbors - kNN), and then, by using this ground truth, the available measurements are used for the knowledge extraction. Additionally, the previously described scheme exploits the measurements for the extraction of the new membership functions in a rather simple manner, using a well-known clustering method, the k-Means. This however leads to loss of information in the overlap areas of the clusters, as shown in the Figure 6-4, because the density of the measurements is not being consider, but only the radius of the hyperspheres is being exploited.

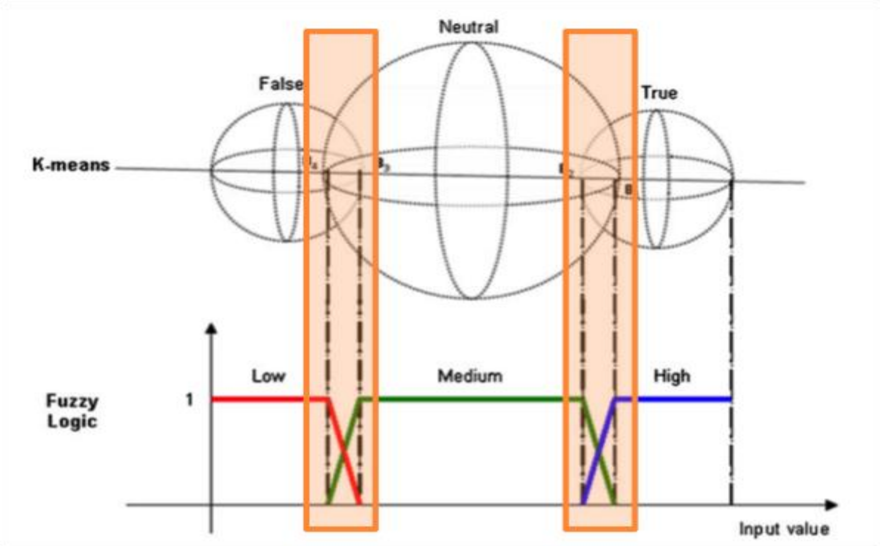


Figure 6-4: Geometric representation of the supervised learning approach and highlight of the overlap areas

In contrast, in the unsupervised learning scheme, the diversity of the input measurements via statistical analysis of the monitored instances is being considered. Furthermore, the labeling part used in the supervised learning scheme is being omitted. The unsupervised learning approach assumes an operational phase where data are being gathered on-the-fly and are then used for the adaptation of the input membership functions of the situation perception fuzzy reasoner (Figure 6-5).

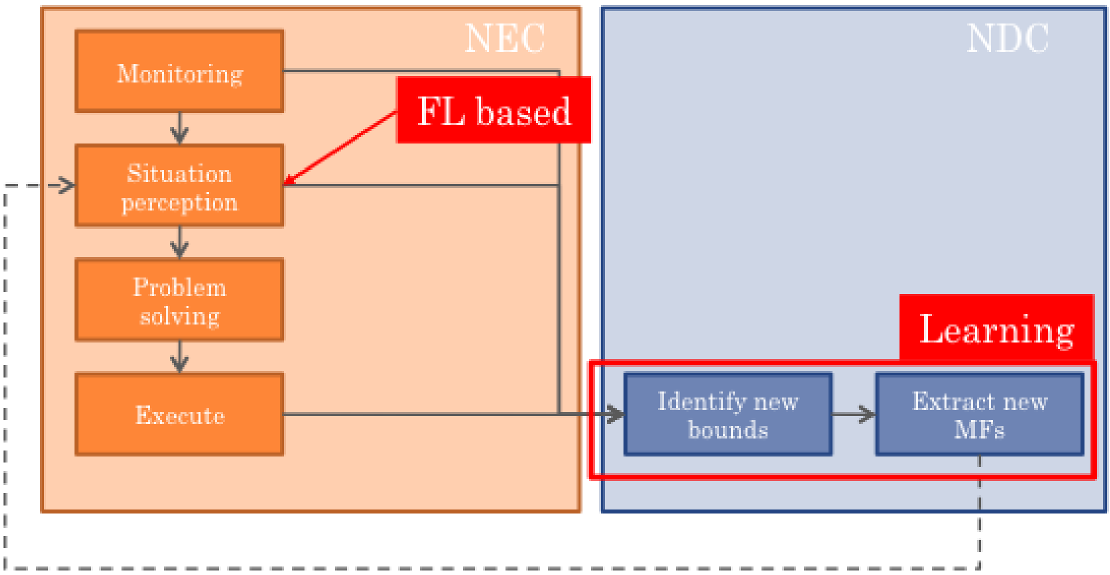


Figure 6-5: High level view of the unsupervised learning scheme

The proposed scheme is based on a set of assumptions (the brackets contain the link to the generic problem formulation described in (Section 6.1)):

- Each NEC monitors the network inputs for identifying the network current state (i.e., network states $s \in S$) and proceeds in self-diagnosis (i.e., $U_i: S \rightarrow R$ and Θ_i are the input membership functions).
- Each monitored tuple is being evaluated by the fuzzy reasoner and classified as low, medium and high. The following cases could arise:
    - Low: the situation perception mechanism identifies a problematic situation. In the specific case where the identifier mistakenly considers a situation as low whereas it is not, the problem solving mechanism that undertakes to solve the identified problem intervenes without being needed (true negative),
    - Medium: the situation perception mechanism concludes to a medium (network) state, which is not problematic, but it could lead to either low or high QoS state without major alterations in the inputs,
    - High: the situation perception mechanism identifies a normal situation (i.e., high QoS). In the problematic case, where the fuzzy reasoner mechanism identifies a high QoS situation instead of a low one, the problem solving mechanism does not intervene (false positive).

Once we have gathered enough (classified) measurements we have three sets of tuples, labeled as Low, Medium and High; misclassified tuples of the diagnosis mechanism from the true negatives and false possitives are also included in the three sets. The classification is based solely on the current understanding of the decision maker on what constitutes Low, Medium and High respectively. The approaches that we have followed regarding the statistical processing of the measurements are:

- the use of the Gaussian distribution and
- the non-parametric one (i.e., which uses the Kernel Density Estimator (Gaussian Kernel is used) of the measurements histogram) ([178], [179], [180], [181]).

The former approach is simpler whereas the latter provides a better “fitting” to the available data. For the Gaussian distribution approach, we obtain the mean value and the variance of each of the three states of each input i.e., low, medium, high for Delay, Jitter and Packet Loss respectively). This enables the extraction of a Gaussian distribution as shown in Figure 6-6. The mapping of the Gaussian distribution to membership functions is straightforward and suggests the adaptation mechanism A that calibrates the parameters Θ_ι (membership functions) so as to maximize the “correct” decisions and “minimize” wrong ones.

For the non-parametric approach we extract the density of the dataset in every point of the domain of definition (Figure 6-7) and use the Kernel Density Estimator of the measurements histogram for the building of the non-parametric curves. Then we normalize and map these curves into membership functions. As in the Gaussian case, the identification of the new membership functions is the adaptation mechanism A that calibrates the parameters Θ_ι (membership functions) so as to maximize the “correct” decisions and “minimize” wrong ones.

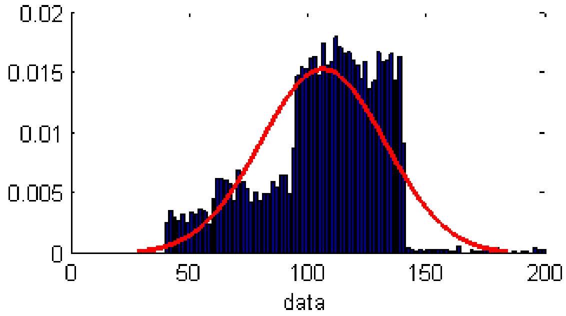


Figure 6-6: Mapping of a cluster to Membership functions using Gaussian statistical analysis

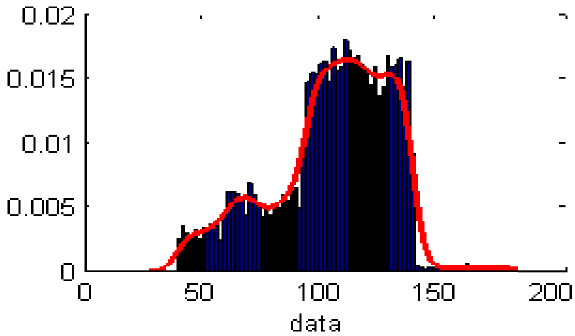


Figure 6-7: Mapping of a cluster to Membership functions Non-parametric statistical analysis

6.3 Fuzzy Logic based learning enhanced situation perception case studies

In Section 5 a description of the situation perception scheme based on fuzzy logic, and its application in several use cases has been provided. For the same use cases, we have applied and evaluated the learning enhanced situation perception for measuring the merits of the learning enhanced schemes. The following section presents the application of the learning schemes and mainly the outcomes of the evaluation.

6.3.1 QoS Degradation Events' Identification

In order to quantify the benefits from the introduction of the proposed learning schemes we have conducted a series of MATLAB simulations for the evaluation of the learning enhanced situation perception. Both algorithms have been applied for evaluating them and for identifying the benefits from their introduction.

The dataset that has been developed and used for the evaluation of the situation perception scheme (Section 5.2.1) has been used also for the evaluation of the learning algorithms. Having the initial generic configuration that was described in details in Section 5.2.1 (Table 5-3), the algorithm performs relatively well and achieves a success rate of 64%. Given the fact that such self-diagnosis scheme is built to operate adequately in all environments we consider this success rate as acceptable.

By applying the supervised learning algorithm (Section 6.2.1) the situation perception algorithm has a success rate of 70.01% (amelioration of 9.4%). The output membership functions are trapezoidal ones, and membership functions have the ranges shown in Table 6-4. By incorporating the unsupervised learning mechanism with the Gaussian adaptation approach and following the methodology presented in Section 6.2.2 we modify the input membership functions as shown in Figures 6-8 – 6-10. As it is obvious, the input states are now being captured by new membership functions, which are being described by Gaussian distributions, with higher overlap areas. The success rate of the adapted scheme reaches 84.07% compared to the ground truth (an amelioration of 31.36%). The required time for the processing of the dataset and the extraction of the new membership functions is 13.07 seconds in an average consumer laptop (i.e., Quad core, 1.6 GHz, 4GB RAM).

Table 6-4: Input membership functions of the self-diagnosis fuzzy reasoners after the clustering adaptation procedure

	Low	Medium	High
Delay (in ms)	0 –20	5 –80	30 –200
Jitter (in ms)	0 – 40	0.35 – 1	0.55 –2
Packet Loss (%)	0 – 0.005	0.004 – 0.0057	0.0055 –0.01

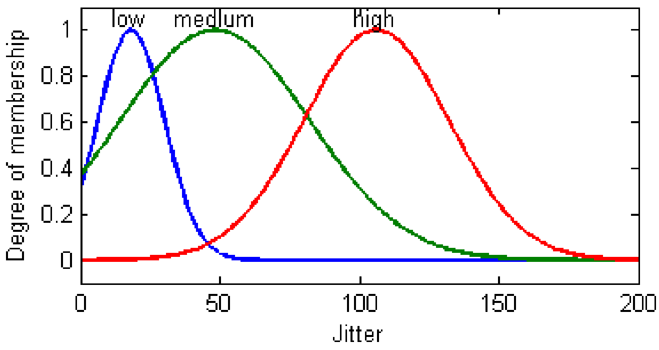


Figure 6-8: Membership functions for the Jitter input after the Gaussian adaptation procedure.

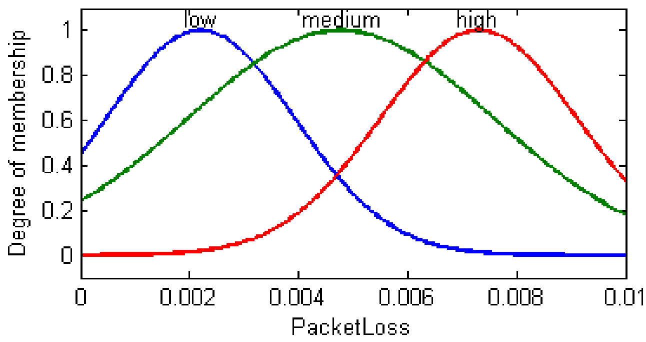


Figure 6-9: Membership functions for the Packet Loss input after the Gaussian adaptation procedure.

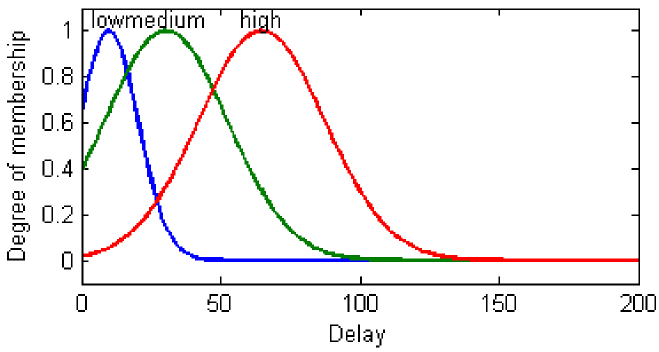


Figure 6-10: Membership functions for the Delay input after the Gaussian adaptation procedure.

Table 6-5: Mean values (μ) and standard variations (σ) of the input membership functions of the Gaussian adaptation scheme

		Low	Medium	High
Delay (in ms)	μ	9.8	30.32	64.67
	σ	10.48	22.2	22.81
Jitter (in ms)	μ	18	48	106

	σ	12	34	26.11
Packet Loss (%)	μ	0.0022	0.0047	0.0073
	σ	0.0017	0.0028	0.0018

For the same dataset, we also apply the unsupervised learning non-parametric approach. Given the fact that for the adaptation of the membership functions we must have a finite number of points (MATLAB fuzzy logic toolbox limitation [128]) we choose 16 points of the extracted distribution curves and we define the membership functions. Apparently the new membership functions are closer to the actual distribution of the dataset (Figures 6-11 - 6-13), and reach a success rate of 84.16% compared to the ground truth (an amelioration of 31.51%). In this case the required time for the adaptation procedure (processing and extraction of the new membership functions) is 22.38 seconds. The results of the analysis for the three learning schemes are being summarized in Figure 6-14.

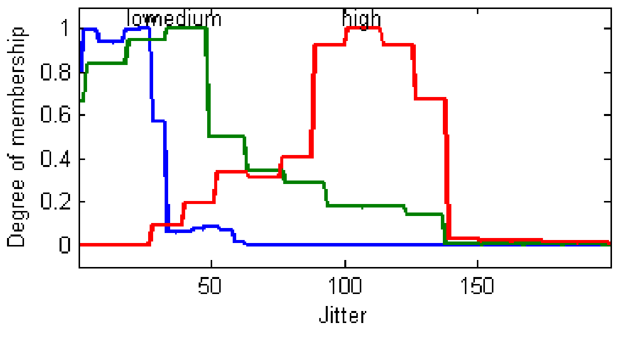


Figure 6-11: Membership functions for the Jitter input after the non-parametric adaptation procedure.

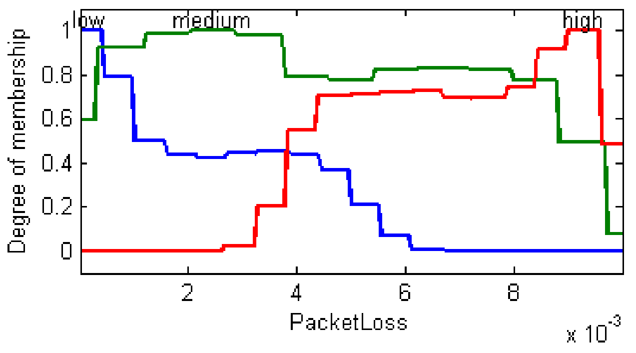


Figure 6-12: Membership functions for the Packet Loss input after the non-parametric adaptation procedure.

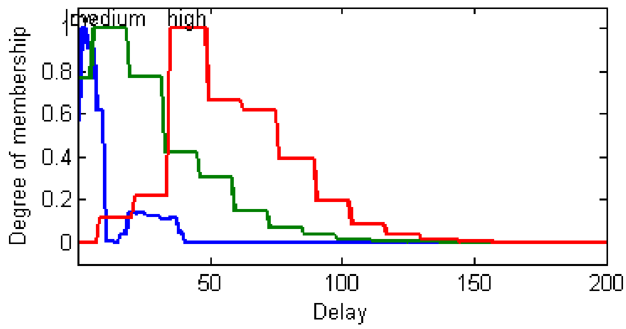


Figure 6-13: Membership functions for the Packet Loss input after the non-parametric adaptation procedure.

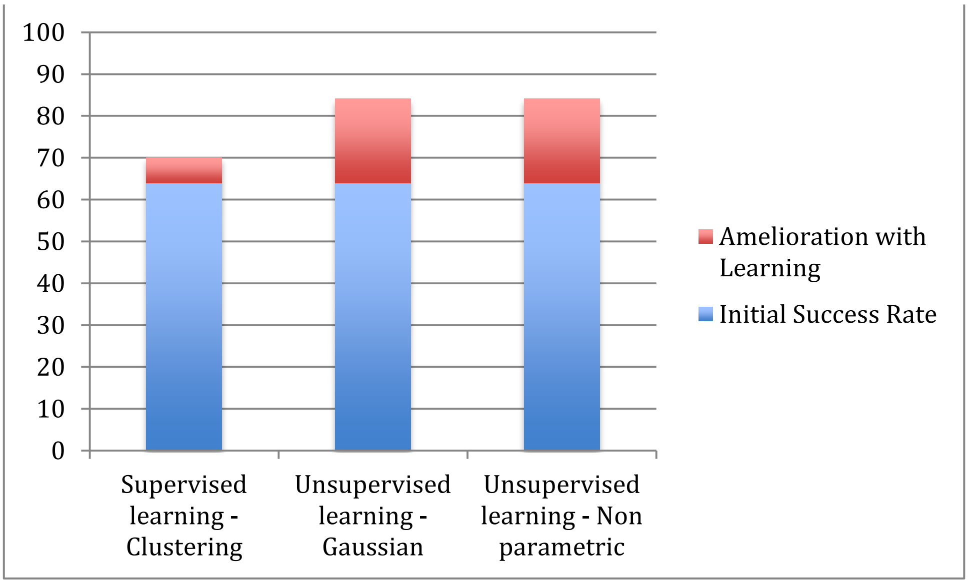


Figure 6-14: Comparative analysis of the amelioration for the three learning schemes

6.3.2 Load Events Identification

The supervised learning scheme has been also applied for the identification of load events. The experimentation analysis is the same as the one described in Section 5.2.2 for the fuzzy logic situation perception scheme. In the first experiment the ability of kNN to support the derivation of the ground truth label for the first step of the algorithm is validated (Table 6-3). In most of the cases in the literature, the k value is chosen heuristically. For identifying the optimum value of neighbors, we have performed several experiments with values of k 1, 5, and 10. The results have been assessed through a 10-fold cross validation procedure, while all experiments verified our initial intuition regarding the applicability of k-NN in the context of our problem exhibiting, a classification rate larger than 98%. The latter lead us to the additional conclusion that any value between 1 and 10 will provide results of adequate quality. The overall results are presented in Table 6-6.

Table 6-6: kNN classification ability in the context of load identification use case

Number of kNNs	**1**	**5**	**10**
Correctly Classified Instances	98.7%	98.6%	98.5%

For the experimentation with the learning algorithm, the three configurations presented in section 5.2.2 have been used. The three configurations (ranging from 1 to 3 – Table 5-3) are more generic (i.e., the 1st configuration) to more targeted to the environment (3rd configuration), thus they result to different success rates (Table 5-6).

An interesting outcome after observing the original dataset is that the results will be heavily influenced by the values of the AT parameter, while on the other hand PER seems to provide little information in the clustering process. The latter is due to the fact that these variables are in different scale. Indeed, AT Є N takes values from the range [0…25] while PER Є R and specifically in [0…1] with the majority of its values concentrated in [0…0.015]. In order to overcome this issue, the values of PER and AT are normalized using the formulas (8) and (9), respectively.

$$PER_i = \frac{PER_i}{\max\limits_{j=1..N} PER_j} \quad (6.7)$$

$$AT_i = \frac{AT_i}{\max\limits_{j=1..N} AT_j} \quad (6.8)$$

Figure 6-15, Figure 6-16, and Figure 6-17 present the results after executing the algorithm on the dataset. The algorithm uses the decision obtained from the fuzzy logic controller and divides the input tuples into three classes according to the identified state of the network element (i.e., Load, Medium Load, No Load). Then the tuples identified as Load are used as input to the clustering algorithm that iteratively divides the set into two clusters, Load and Medium Load, until the halting condition is satisfied; the same procedure is being held for the tuples identified as No Load.

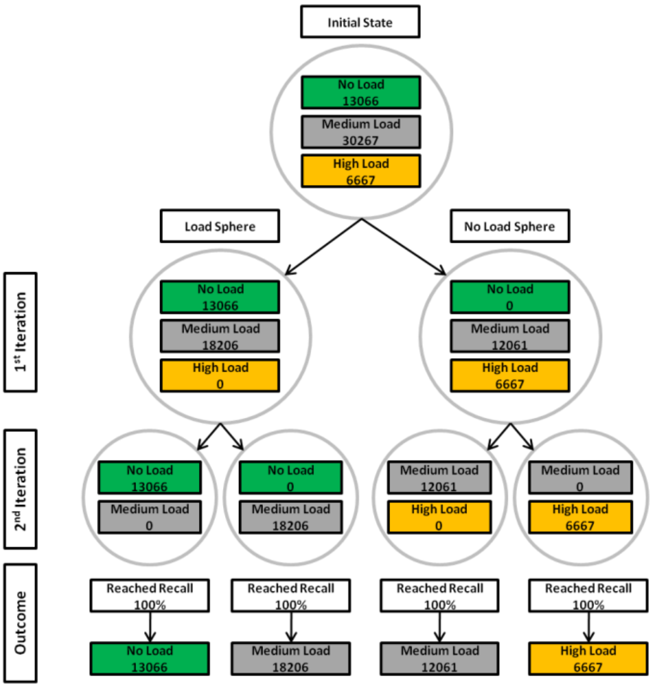


Figure 6-15: Dendogram corresponds to FL1 derived after applying our algorithm on the original dataset

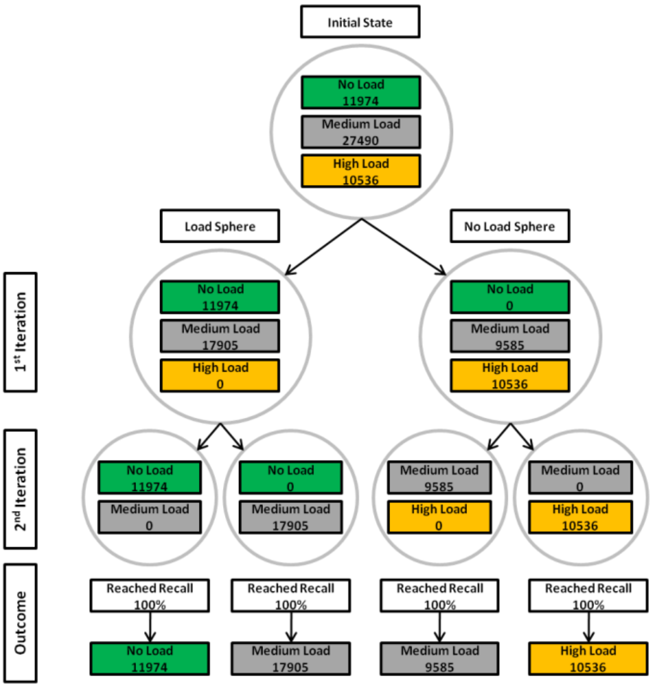


Figure 6-16: Dendogram corresponds to FL2 derived after applying our algorithm on the original dataset.

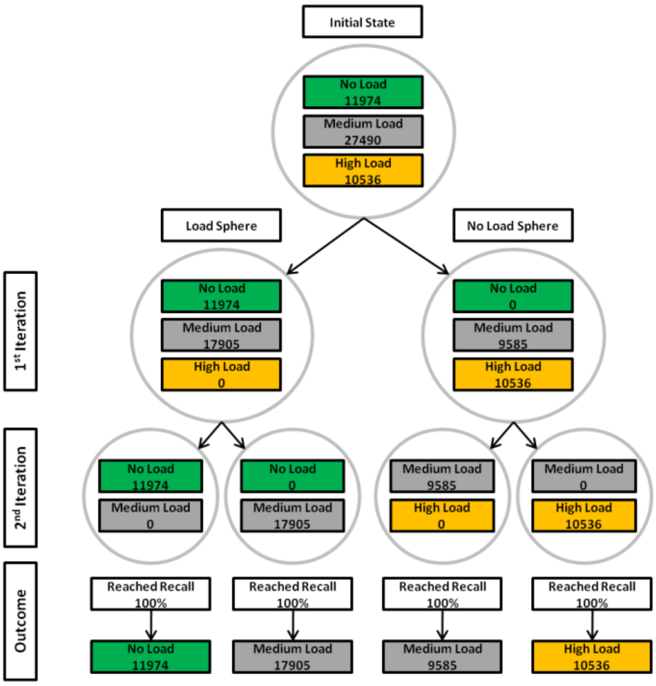


Figure 6-17: Dendogram corresponds to FL3 derived after applying our algorithm on the original dataset

When the halting condition is validated we acquire a good approximation of the actual sets, while any potential overlapping corresponds to the intersection of the membership functions. The bounds obtained from these experiments appear in Table 6-7. By employing the derived points as the new bounds for the membership functions we apply the algorithm of Section 6.2.1 and obtain the classification results appearing in Figure 6-18.

Table 6-7: The bounds extracted from k-Means after clustering on the normalized dataset

		PER	CU	AT
Low	FL_1	[0 ... 3.2*10-3]	[0 ... 0.375]	[0... 5.23]
	FL_2	[0 … 2.83*10-3]	[0 … 0.337]	[0 … 4.26]
	FL_3	[0… 3.03*10-3]	[0 … 3.03*10-3]	[0… 3.03*10-3]
Medium	FL_1	[2.87*10-3 ... 1.17*10-2]	[0.357 ... 0.756]	[4.29 ... 16.18]
	FL_2	[2.43*10-3 … 1.13*10-2]	[0.312 … 0.741]	[3.03 … 15.74]
	FL_3	[2.69*10-3 … 1.10*10-2]	[0.33… 0.719]	[4.29 … 16.18]
High	FL_1	[1.16*10-2 ... 1]	[0.747... 1]	[15.89 … 25]
	FL_2	[1.12*10-2 … 1]	[0.728 … 1]	[15.3 … 25]
	FL_3	[1.08*10-2 … 1]	[0.698 … 1]	[15.89…25]

Based on the experiments we conclude that the algorithm performs significantly well and tends to provide rules, which converge to decisions closer to the ground truth, independently of the initial configuration of the network element's decision making engine. For the three initial configurations of the fuzzy logic controller the achieved amelioration is of 14 - 17% (Table 6-8). The presented amelioration focuses on the situation awareness of a cognitive manager. The characterization of events, which is a situation awareness phase, is the pilot for the successful optimization or fix of the network system. If the cognitive manager (i.e., NEC) cannot assess effectively the local status, then the performance of the network in many cases will not be improved by applying a reconfiguration action. Thus, the correct labeling of events is an important task for autonomic network management systems, where the learning process has merit.

Table 6-8: Classification results after the enhancement of the fuzzy logic rules

	FL_1	FL_2	FL_3
Initial	65.64%	71.86%	75.40%
Learning	76.73%	84.09%	86.06%
Amelioration	16.8%	17.01%	14.13%

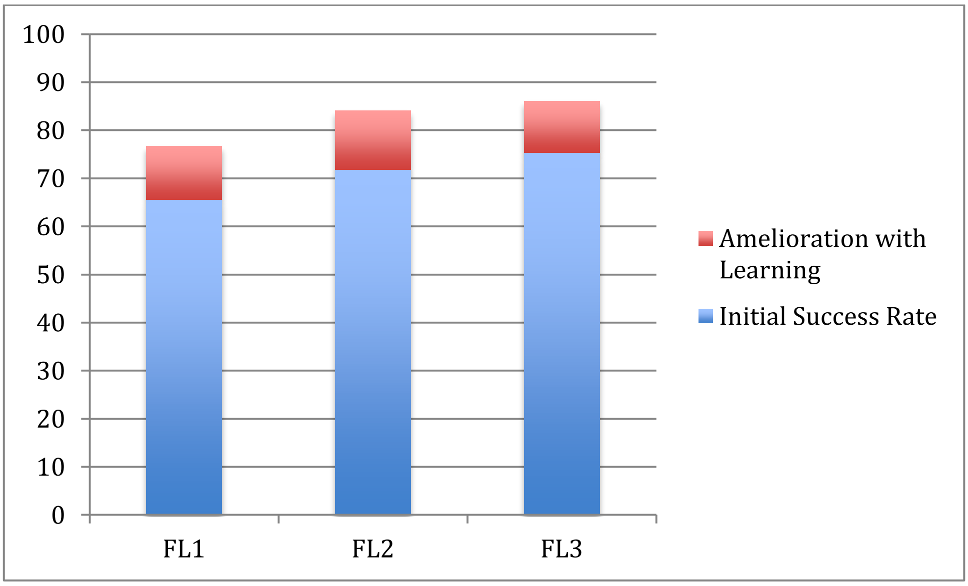
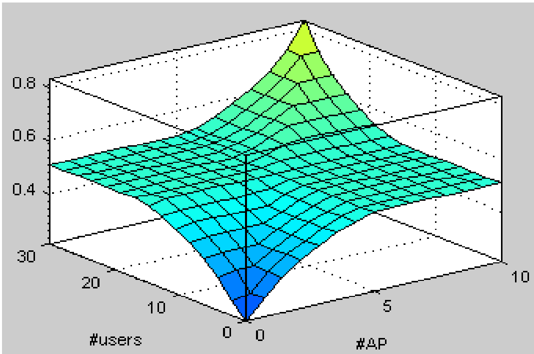


Figure 6-18: Comparative analysis of the amelioration for the three initial configurations

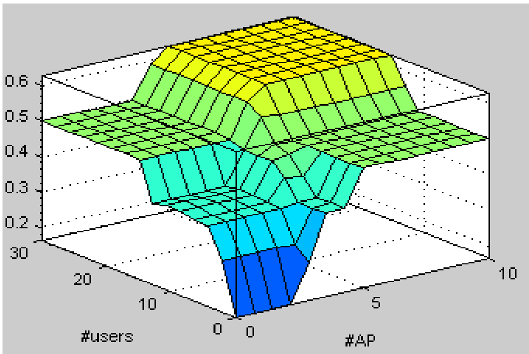
6.3.3 Fuzzy Logic-based Cooperative Power Control

The supervised learning scheme has been applied for the Fuzzy Logic-enhanced Cooperative Power Control. The performed analysis is based on two series of experiments, one based on simulations, and one based on the experimental platform that was presented in Section 5.2.3.

The scheme without learning (Section 5.2.3) is used as the baseline for the comparisons. For the realization of the simulations we have artificially created a dataset consisting of 1000 pseudo-random tuples. The dataset reflects network topologies with a relatively small number of APs, as well as the collocated users. Figure 6-19 provides the Interference weight (i.e., outcome of the fuzzy reasoner) as a function of the APs' and the users' number, having as parameter the time interval before (Figure 6-19 (a)) and after (Figure 6-19 (b)) the learning procedure. It is apparent that the weight of the interference part of equation CPC (Equation 5.3) is significantly affected, based on the feedback from the learning procedure; this implies that the transmission power extraction procedure is affected as well.



(a)



(b)

Figure 6-19: Interference weight before (a) and after (b) the learning procedure

For the whole dataset we capture the values of the α factor; then we perform a fitting procedure in order to identify the polynomial functions that capture in the most suitable way the outputs. Figure 6-20 provides the 8th polynomial degree functions of the α factor before and after the learning procedure. After the learning procedure, the fuzzy reasoner has become more sensitive to the environment; this is being captured by the variation of the new a values (0.0458) instead of the old ones (0.0091).

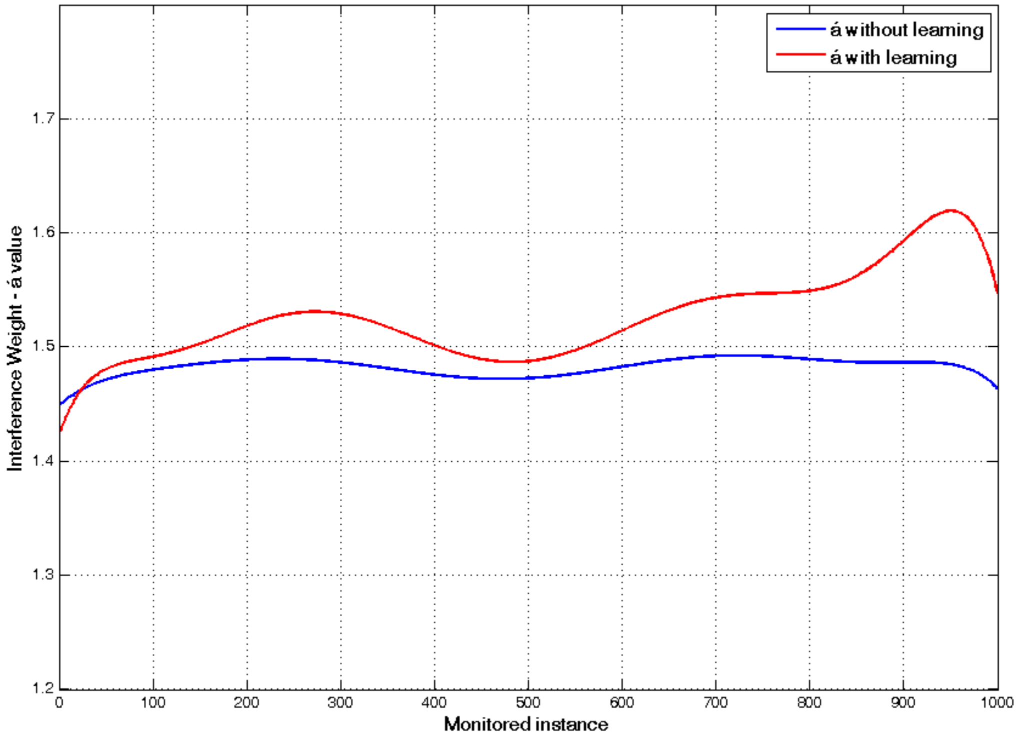


Figure 6-20: Interference weight a values before and after the learning procedure

For a given instance of the dataset, we identify the transmission power before and after the learning procedure. We randomly create a set of experiments (10 random topologies) for the identified instance, and evaluate the algorithm performance. As depicted in Figure 6-21, certain deviations to the final power values can be noticed when learning procedure is applied. In specific topologies (i.e., 2nd, 3rd and 8th) significant energy gains are achieved. In the rest of the topologies the learning framework achieves less significant gains but in no occasion energy waste occurs.

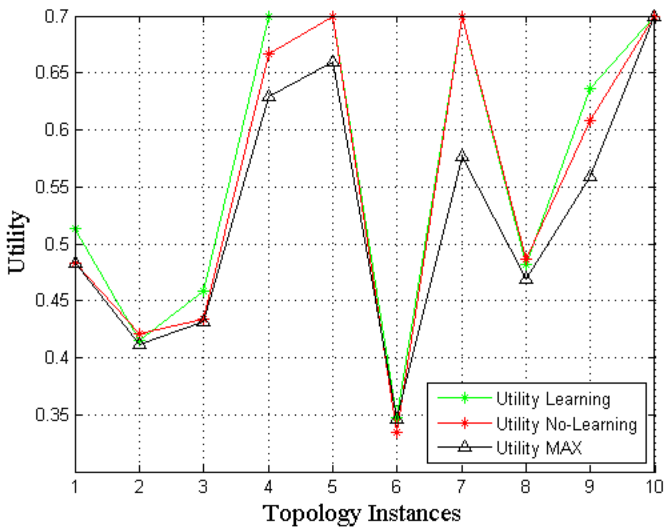


Figure 6-21: Overall utility before and after the learning procedure

In Figure 6-22 the overall utility of the network for ten same experiments is presented. The utility with the incorporation of the learning framework is significantly ameliorated compared to the one with the transmission power set to the maximum valid level. Moreover, after the deployment of the learning algorithm, the network elements achieve better results in the overall utility, in comparison to the ones with the cooperative power control without learning capabilities.

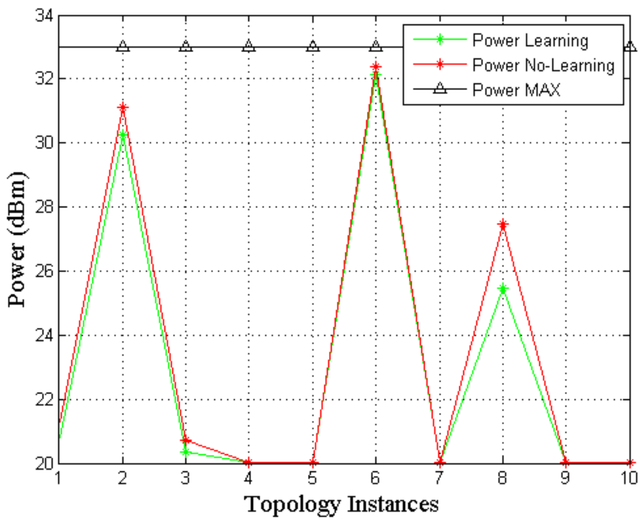


Figure 6-22: Transmission Power before and after the learning procedure

A similar analysis has been performed using the experimentation platform presented in Section 5.2.3. Again, the initial configuration of the network elements is a generic one and captures a great variety of environments. However, for both the physical and network topology (Figure 5-18 and Figure 5-19) which has been used for experimentation, the set configuration is not the most suitable one. Thus, the supervised learning scheme has been incorporated. During the first experimentation day of the CPC in the Soekris devices, the inputs of the fuzzy reasoner are being gathered and characterized as neutral, beneficiaries and non beneficiaries. Then, the tuples are being clustered and the overlapping areas are being mapped to the uncertainty bounds in the input membership functions. Figures 6-23 – 6-26 provides the transmission power throughout the second experimentation day for all Soekris devices, with the adapted input membership functions (supervised learning-based CPC scheme).

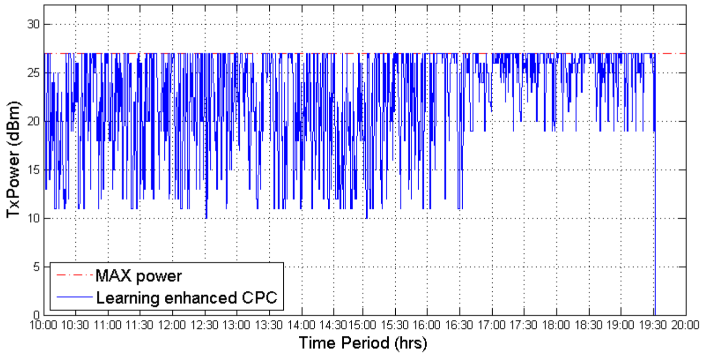


Figure 6-23: Transmission power adjustments using the Learning enhanced Cooperative Power Control scheme in Soekris AP1

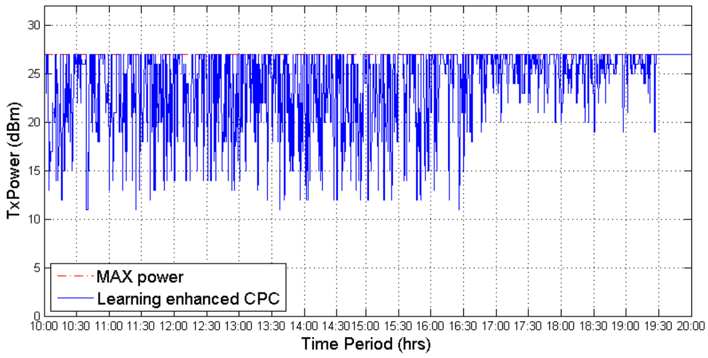


Figure 6-24: Transmission power adjustments using the Learning enhanced Cooperative Power Control scheme in Soekris AP2

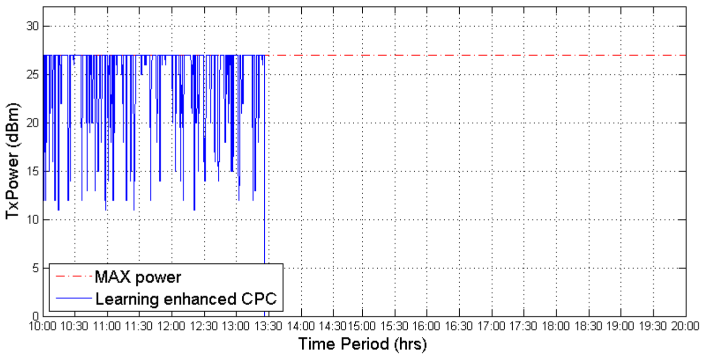


Figure 6-25: Transmission power adjustments using the Learning enhanced Cooperative Power Control scheme in Soekris AP3

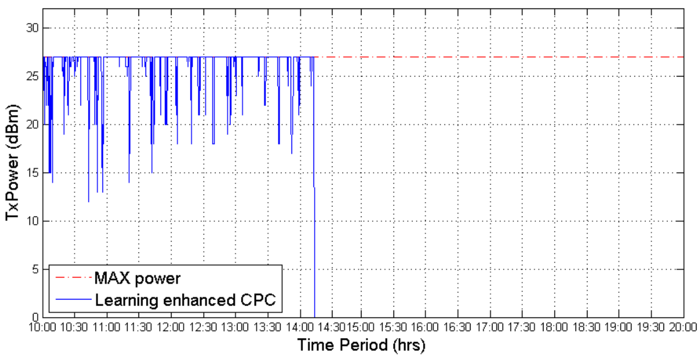


Figure 6-26: Transmission power adjustments using the Learning enhanced Cooperative Power Control scheme in Soekris AP1

As it is obvious from the flunctuations, the CPC scheme is more sensitive to the environment, compared to the second day of experimentations (CPC without learning). Given the fact that they operate in the same environment, the APs proceed even more often in transmission power adjustments. Similarly to the case without the learning enhancements the APs are being turned off so as to capture the way it operates. When only two APs remain operational, as the experimentation proceeds, we observe that they proceed in transmission power adjustments, according to the environment stimuli, contrary to the first day, where the transmission power adjustment mainly occurred when all the APs were operational.

Furthermore, we observe significant energy gains, in relation to the case without learning capabilities. More specifically, AP 1 has a 24.73% less power consumption compared to the maximum transmission power, whereas AP 2 consumes 18.01% less power, AP 3 14.69% and AP 4 5.65%. The reason that AP1 and 2 have more energy gains is that they remain operational almost throughout the experiment. Regarding the SINR, it remains in the same levels as in the case of the core CPC algorithm (Figures 6-27 – 6-30), due to the fact that the objective function to be optimized is the same. The APs proceed in power adjustments in lower transmission power levels resulting in less interference as well; however the SINR remains at the same levels, due to the decrease in both metrics (i.e., TxPower and interference).

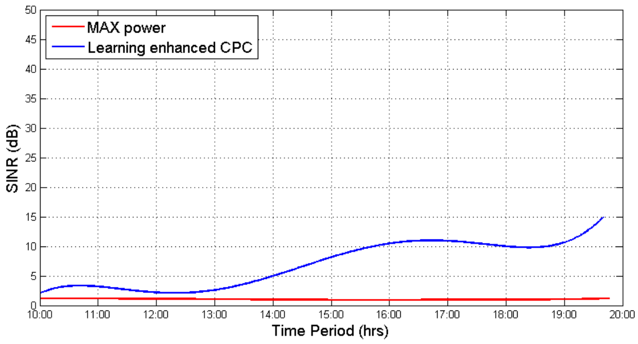


Figure 6-27: SINR evolution during the experimentation period for Soekris AP1

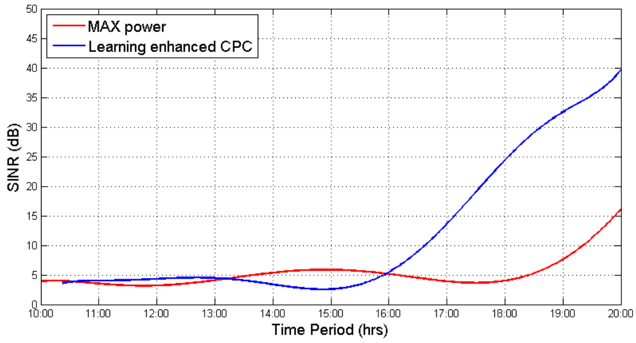


Figure 6-28: SINR evolution during the experimentation period for Soekris AP2

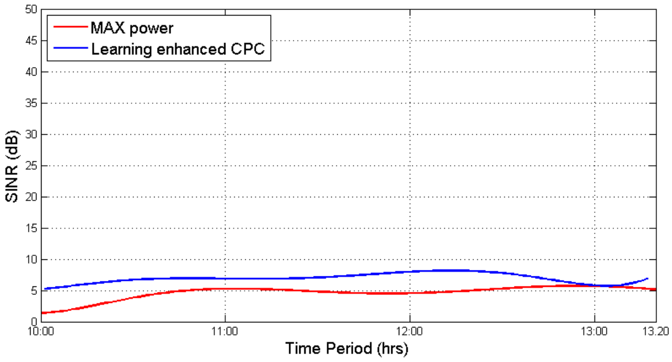


Figure 6-29: SINR evolution during the experimentation period for Soekris AP3

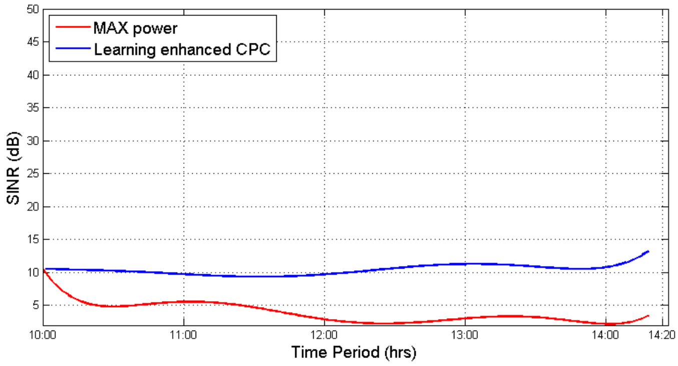


Figure 6-30: SINR evolution during the experimentation period for Soekris AP4

Additionally, Figure 6-31 presents the number of iterations every time the CPC is being triggered after the learning procedure. Similarly to the core CPC we observe that the scheme converges in small number of iterations most of the times; furthermore we observe a slight decrease in the overall mean value of iterations (3.47), which also highlights that the system has become more suitable to its environment. Finally, considering that the algorithm is being triggered periodically, every 5 minutes for 10 hours, we observe that the adaptation algorithm enhances the situation perception scheme and the overall CPC performance using relatively small amount of measurements (4 AP * 120 measurements/AP = 480 measurements).

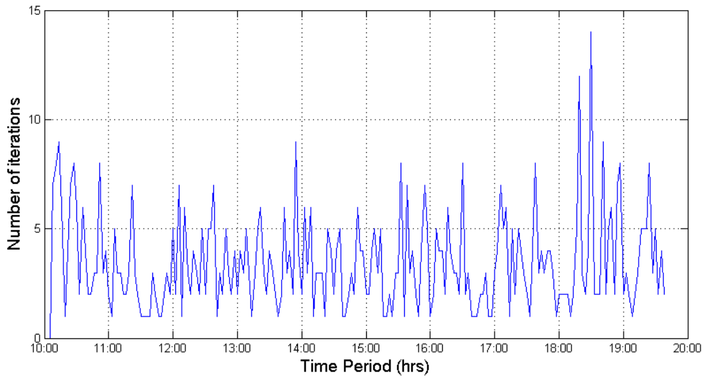


Figure 6-31: Number of iterations every time the Cooperative Power Control is being triggered

6.4 Analysis of Learning schemes

Section 6 presented two mechanisms for updating Situation Perception schemes based on fuzzy logic. The updating is based on the evolution of the environment modeling of the Situation Perception functions, which are based on fuzzy logic controllers. The schemes have been applied in three different fields and have been evaluated for presenting the applicability of the approaches on the one hand and measuring the benefits by the introduction of these schemes.

Subsection 6.4 provides a summary of the outcomes of the performed analysis for highlighting the benefits and the drawbacks of each approach, as well as the key differences among the adaptation mechanisms. More specifically we observe that:

1) All the learning schemes perform well. More specifically, the amelioration in the success rate in the event identification, using the enhanced schemes, ranges from 10-30% depending on the initial configuration, the number of measurements, and the problems that have to be tackled.

2) The schemes are becoming more sensitive to their environment. This is related to the fact that the Situation Perception functions are more suitable to the environment that they have to operate, because they model the way they perceive it using the analysis of the measurements.

3) The cost for the learning in all three approaches is relatively small, because the adaptation procedure is offline, thus minimizing the

processing requirements. The signalling cost on the other hand shouldn't be neglected. This is why, it is attempted to be communicated via the wired backhaul links (i.e., in the load events' identification, and the cooperative power control). In the QoS degradation events' identification such aspect has not been considered, though such information could be communicated in cases where the communication overhead is limited [182].

4) The supervised learning solution requires the classification phase which is demanding. On the other hand, the unsupervised learning solutions operate satisfactory without the classification procedure. However, they are based on the assumption that the situation perception scheme with the generic configuration operates at least satisfactory (events' identification success rate above 50%) at every environment where it is placed (this requirement is not very hard to meet, assuming that all network elements perform relatively well with the generic static configurations the present day).

5) The two versions of the unsupervised learning scheme (i.e., Gaussian and non-parametric one) perform well and succeed almost the same results (~84% success rate). However, the non-parametric requires approximately 40% more processing time compared to the Gaussian. At this point it should be noted that the required time is indicative of the processing cost of the proposed solution and should not be considered as an absolute value taking into account that the validation is performed using MATLAB.

6) The experimental analysis showed that the supervised learning scheme is highly related to the initial success rate. This on the one hand implies that the algorithm will converge and will achieve high success rates, but on the other hand means that it may need big dataset, if the initial configuration is not suitable for the environment (e.g., the initial configuration has success rate ~55-60%).

7) The situation awareness scheme for the CPC needs relative small amount of measurements for achieving power gains. This is related to the fact that in the CPC the environment modeling directly affects the TxPower setting, and not a decision about the state. Thus even small changes in the situation perception lead to changes in the TxPower setting and consequently to energy savings.

The previous list summarizes the benefits from the introduction of adaptation schemes in the situation perception mechanisms based on fuzzy logic. However, it should be mentioned that the previous analysis is related to the specific problem of situation awareness with fuzzy logic controllers, with special characteristics that enable the application of learning algorithms, such as the offline operation of the adaptation scheme, the gathering of adequately enough data, and the satisfactory initial configuration of the fuzzy reasoners.

7. Conclusions

The objective of this thesis is to analyze the concepts of the Situation Awareness and Situation Perception and present solutions for these research areas. Situation Awareness is the ability of the network elements to model their environment, assess it and interpret it so as to predict the near future. The situation awareness may be decomposed into three steps, namely, the situation perception, the comprehension of current situations, and the projections. As situation perception we define the proper perception of the operational status of the system or the network element and is the primarily interpretation of the available information (i.e., to an elementary knowledge interpretation). In the context of this thesis, the focus has been placed on the analysis of the previous notions, and on the development of an architectural solution that enables network elements to perceive their environment correctly and efficiently. Additionally, new schemes for efficient and effective situation perception based on fuzzy logic have been proposed. These schemes have been enhanced by adaptation-learning mechanisms, so as to be able to adapt their contextual models, based on the environment stimuli.

The idea of self-awareness is based on the principle that the network elements will have the ability to operate in an autonomous manner, which will enable them to operate without human intervention – in a self managed manner. In Section 2 the principles of autonomous networking have been analyzed thoroughly, as well as the research activities towards this direction, and we have concluded to the key requirements of a Self-Managed network. Towards the direction of Self-Managed networking, the network elements shall be able to monitor their environment and reason about their status and condition. This functionality is the self-awareness, which has been analyzed thoroughly in terms of this thesis. More specifically, in Section 3, we have identified the key aspects of Self-awareness and we have proposed the functional architecture of a Self-Aware (and Self Managed) network. This architecture is a two layer hierarchical approach, where the functions are split between network elements of different hierarchy levels (lower – network element controllers (NEC) and higher ones – network domain controllers (NDC)) depending on their capabilities, their restrictions, and their network view. Afterwards, an extensive state of the art analysis is being presented, highlighting that the situation perception problem has not been satisfactory discussed in the literature up to now, and mainly has been considered as a policy and threshold based problem. This, in conjunction to the fact that several manufacturers have shown interest to mechanisms that are able mimic human behavior leads to the outcome that new mechanisms for situation awareness are required. Such schemes shall:

- Be easy to introduce, implement, and configure,
- Be easy to implement,
- Mimic human behavior.

In order to meet the aforementioned requirements of new schemes for situation perception, this disertation presents, an innovative fuzzy logic scheme, to be generalized and applied in several problems. The rational for the use of fuzzy logic could be summarized in its key characteristics:

- It is an multi-variable mechanism,
- It is a scheme for handling multiple optimization goals or faults may arise,
- It may handle contradictive inputs, or uncertainty cases.

Thus, for three use cases (that of the QoS degradation events' identification, Load events' identification, and that of Cooperative power control) we have developed the situation perception mechanisms based on fuzzy logic reasoners. The analysis incorporates the description of the functional architecture of the NEC and the NDC for implementing the solutions, which is in accordance to the proposed generic functional architecture for self-managed networks. Finally, we have developed the proposed schemes and experimented for capturing their applicability on the one hand, and their efficiency and effectiveness on the other. All the proposed schemes performed well and with generic configurations managed to succeed satisfactory event identification (success) rates. With generic configurations, in the use cases of QoS events identification and Load events' identifications, the fuzzy reasoners managed to make correct decisions in 64-66% of the cases. If the fuzzy reasoners were having more targeted configurations, better success rates could be achieved. Regarding the power control use case, the developed scheme managed to have significant energy gains (10 - 30%) compared to static setting of the transmission power (to maximum). Additionally, the use of fuzzy logic enhanced cooperative power control, has ameliorated significantly the SINR levels in developed setup.

The previously described schemes, based on fuzzy logic, even though that they perform well in the environments where they are configured to operate, they do not manage to adapt and operate well in totally unknown and new environments (self-mutable requirement of the self-managed networks). This implies that the network elements shall be manually configured by the network administrators, according to the environment changes. For enabling the network elements to self-configure for operating in new environments, two learning schemes have been developed, a supervised learning one and an unsupervised learning one. Both schemes have been mapped to the reference problem of situation perception and learning, thus providing a generic methodology for the application of such schemes in similar problems in the future. The learning schemes have been applied in the use cases that have been used to validate the fuzzy logic based situation perception. In all three cases we observed that the learning schemes perform very well. More specifically, the amelioration in the success rate in the event identification (QoS events' degradation and Load events' identification), using the enhanced schemes, ranges from 10-30% depending on the initial configuration, the number of measurements, and the problems that have to be tackled. This is being achieved with relatively small cost, because the learning process, which is based on the analysis of a rather big dataset, is an offline one, thus minimizing the processing requirements. Regarding the Cooperative Power Control use case, the situation awareness scheme needs relative small amount of measurements for achieving power gains because the modeling directly affects the TxPower setting, and not a decision about the state. However, the signalling cost on the other hand shouldn't be neglected, thus the decision of the learning points should be carefully decided.

Concluding, a set of outcomes of the performed analysis should be drawn. The following list summarizes these outcomes:

- The environment modeling (such as that of the situation perception) that the network elements incorporate shall evolve according to the environment stimuli.

- The hierarchical approach enables the effiecnt and scalable handling of several problems, that may require larger network view, or more data to be analyzed.
- The decision regarding the placement of the higher hierarchy network elements (Network Domain Controllers - NDC) in the network shall be studied carefully for reducing the significant signalling overhead.
- The exact analysis of the dataset may be not required. Rough analyses of the dataset lead to comparable gains to very detailed analyses, with significantly lower computational cost.
- The information gathering in the NDC from the NEC shall concern similar environments, thus mechanisms for characterising the environment are required.
- The information that will be used for learning shall be valid. Thus either resilient mechanisms for monitoring are required, or outlier detection schemes shall be introduced.
- Supervised and unsupervised learning methods may perform equivalently well, in the situation perception problems, with the requirement of having satisfactory initial event identification levels. This implies that the initial configuration shall be generic. This is the cost compared to other methods, such as the reinforcement learning, which does not require initial configuration, but requires a very accurate mathematical formulation of the network/environment model.

Potential next steps are mainly related to the key characteristics described above. More specifically:

- The fuzzy logic situation perception schemes are based on predefined rules. These rules are developed by experts. However, due to the environment changes, the rules may not be valid in the future. Thus adaptation schemes for the fuzzy logic rules shall be developed. Such schemes may be based on reinforcement learning [130].
- Methods for compression of the information for signaling reduction. This may be achieved with preprocessing of the information in the NECs.

- Schemes for identifying the proper placement of the NDCs in the network. This analysis shall be related with the functionalities that shall be centralized, as well as the degree of the centralization of such functionalities.
- Methods for identifying the area where the NECs have the same environment shall be identified. This implies either spatial neighborhood, or logical one.

Acronyms

2G	2nd Generation
3G	3rd Generation
3GPP	3rd Generation Partnership Project
4G	4th Generation
5G	5th Generation
AFI	Autonomic network engineering for the self-managing Future Internet
AP	Access Point
AT	Associated Terminal
BS	Base Station
BML	Business Management Layer
CAQF	Cooperative Agent-based QoS Framework
CBR	Case Based Reasoning
CCO	Coverage and Capacity Optimization
CET	Central European Time
CoMP	Coordinated Multi Point
CPC	Cooperative Power Control
CPU	Central Processing Unit
CU	Channel Utilization
D2D	Device to Device
DbC	Design by Contract
D-BRAIN	Dynamic Bayesian Reasoning & Advanced Intelligent Network
DIKW	Data-Information-Knowledge-Wisdom
DME	Decision Making Element
DT	Decision Tree
e-ICIC	enhanced Inter-Cell Interference Coordination
EML	Element Management Layer
eNB	evolved NodeB
ETSI	European Telecommunications Standards Institute
FACPS	Fault, Configuration, Accounting, Performance and Security
FFT	Fast Fourier Transform
FL	Fuzzy Logic

FLC	Fuzzy Logic Controller
GANA	Generic Autonomic Network Architecture
HAC	Hierarchical Agglomerative Clustering
HDC	Hierarchical Divisive Clustering
HSDPA	High Speed Downlink Packet Access
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
I-RAT	inter-radio access technology
ITU	International Telecommunications Union
kNN	K Nearest Neighbor
LTE	Long Term Evolution
LTE-A	Long Term Evolution Advanced
MAPE-K Model	Monitor Analyze Plan Execute Knowledge Model
MDE	Monitor Decide Execute
MDP	Markov decision processes
ME	Managed Entity
NDC	Network Domain Controller
NDCM	Network Domain Cognitive Manager
NEC	Network Element Controller
NECM	Network Element Cognitive Manager
NEL	Network Elements Layer
NML	Network Management Layer
NMS	Network Management System
NRM	Network Reconfiguration Manager
PER	Packet Error Rate
PL	Packet Loss
QoS	Quality of Service
RAN	Radio Access Network
RDS	Resource Directory Server
RME	Reconfiguration Management Entity
SA	Situation Awareness
Self-CHOP	Self Configuration Healing Optimizing Protecting
SCC41	Standards Coordinating Committee 41

SINR	Signal to Interference plus Noise Ratio
SLA	Service Level Agreement
SML	Service Management Layer
SOM	Self Organizing Map
SON	Self-Organizing Network
SVM	Support Vector Machine
TCP	Transmission Control Protocol
TD	Temporal Difference
TMN	Telecommunications Management Network
TRM	Terminal Reconfiguration Manager
TxPower	Transmitted Power
UE	User Equipment
UDP	User Datagram Protocol
UMF	Unified Management Framework
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WSN	Wireless Sensors Network

References

[1] Cisco 2013, “Cisco Visual Networking Index: Forecast and Methodology, 2012–2017”, May 2013.

[2] M. Johnsson, B. Jennings, D. Botvich, "Inherently self-managed networks: Requirements, properties and an initial model," IFIP/IEEE International Symposium on Integrated Network Management (IM) 2011, Dublin 2011.

[3] WWRF 2011, “Visions and research directions for the Wireless World”, October 2011.

[4] C. Courcoubetis and R. Weber, eds., “Pricing Communication Networks: Economics, Technology and Modelling”, Wiley, 2003.

[5] T. Chen, H. Zhang, Z. Zhao, X. Chen, “Towards green wireless access networks”, in Proc. CHINACOM, Beijin, China, Aug. 2010, pp.1-6.

[6] J. Kramer and J. Magee, “Self-Managed Systems: an Architectural Challenge”, Future of Software Engineering 2007, IEEE-CS Press, 2007.

[7] David Soldani, “On the Advanced 5G Network Infrastructure for the Future Internet”, IEEE, ICC 2013, Budapest 2013.

[8] J. O. Kephart and D. M. Chess, “The vision of autonomic computing. Computer”, vol. 36, Issue: 1, pp. 41–52, 2003.

[9] M. Huesbcher, J. McCann, “A survey of autonomic computing—degrees, models, and applications”, Journal ACM Computing Surveys (CSUR), vol. 40, Issue 3, 2008

[10] S. Dobson, S. Denazis, A. Fernández, et al., "A survey of autonomic communications", ACM Transactions on Autonomous and Adaptive Systems, vol 1, no. 2, pp. 223-259, 2006.

[11] N. Agoulmine, “Autonomic Network Management Principles: From Concepts to Applications”, Elsevier/Academic Press: New York, 2011.

[12] N. Samaan, A. Karmouch, “Towards autonomic network management: an analysis of current and future research directions”, IEEE Communications Surveys and Tutorials, vol. 11, issue 3, pp. 22–36., 2009.

[13] K. Tsagkaris, G. Nguengang, A. Galani, I. Yahia, M. Ghader, A. Kaloxylos, M. Bouet, S. Georgoulas, A. Bantouna, N. Alonistioti, P. Demestichas, “A survey of autonomic networking architectures: towards a Unified Management Framework”, International Journal of Network Management, vol. 23, issue 6, pp. 402–423, November/December 2013.

[14] C. Fortuna, M. Mohorcic, “Trends in the development of communication networks: cognitive networks” Computer Networks, vol. 53, issue 9, pp. 1354–1376, 2009.

[15] S. Szott, M. Wodczak, R. Chaparadza, T. Ben Meriem, K. Tsagkaris, A. Kousaridas, B. Radier, A. Mihailovic, M. Natkaniec, K. Loziak, K. Kosek-Szott, M. Wagrowski, "Standardization of an autonomicity-enabled mesh architecture framework, from ETSI-AFI group perspective: Work in progress (Part 1 of 2) ", IEEE Globecom Workshops (GC Wkshps), 2012, pp.847,851, 3-7 Dec. 2012.

[16] M. Hasegawa, Y. Kon, K. Ishizu, H. Murakami, H. Harada, "Autonomous optimization techniques for cognitive dynamic wireless systems," IEEE Wireless Communications and Networking Conference Workshops (WCNCW), 2012, pp.420-424, April 2012.

[17] ITU-T, “M.3000: Tutorial Introduction to TMN”, 2000.

[18] ITU-T, “M.3010 Principles for a TMN”, 2005.

[19] ITU-T, “M.3020 TMN Interface Specification Methodology”, 2011.

[20] ITU-T, “M.3050 Enhanced Telecommunications Operations Map (eTOM)”, 2004.
[21] ITU-T, “M.3060 Principles for the Management of the Next Generation Networks”, 2006.
[22] ITU-T, “M.3100 Generic Network Information Model for TMN”, 2009.
[23] ITU-T, “M.3200 TMN Management Services Overview”, 1997.
[24] ITU-T, “M.3300 TMN Management Capabilities at the F Interface”, 1998.
[25] Vertel, The international engineering consortium, “Telecommunications Management Network (TMN)”
[26] Aiko Pras, Bert-Jan van Beijnum, Ron Sprenkels, “Introduction to TMN”, CTIT Technical Report 99-09, University of Twente, April 1999.
[27] ETSI GS AFI 001 V1.1.1, “Autonomic network engineering for the self-managing Future Internet (AFI); Scenarios, Use Cases and Requirements for Autonomic/Self-Managing Future Internet", 06/2011.
[28] R. Chaparadza, et al. “ETSI Industry Specification Group on Autonomic Network Engineering for the Self-managing Future Internet (ETSI ISG AFI)”, Web Information Systems Engineering - WISE 2009, Springer Lecture Notes in Computer Science, vol. 5802, pp. 61-62, 2009.
[29] ITU-T, “M.3400 : TMN management functions”, 2000.
[30] ETSI GS AFI 002 V1.1.1, “Autonomic network engineering for the self-managing Future Internet (AFI); Generic Autonomic Network Architecture (An Architectural Reference Model for Autonomic Networking, Cognitive Networking and Self-Management)", 04/2013.
[31] R. Chaparadza, “Creating a viable Evolution Path towards Self-managing Future Internet via a standardisable Reference Model for autonomic network engineering”, Workshop on Self-Organisation for Beyond 3G Wireless Networks, ICT Mobile Summit 2009, Santander, Spain, 2009.
[32] Li Zhaojun, "An autonomic hierarchical mobility management framework for 3GPP heterogeneous networks," Future Network and Mobile Summit, 2010, pp.1-8, 16-18 June, 2010.
[33] C. Prehofer, C. Bettstetter, “Self-organization in communication networks: principles and design paradigms”, IEEE Communication Magazine, Vol. 43, Issue: 7, pp. 78 – 85, 2005.
[34] Self-Organizing Networks, SON, tutorial, http://www.radio-electronics.com/info/cellulartelecomms/self-organising-networks-son/basics-tutorial.php
[35] 3GPP, Self-Organizing Networks, Keywords and Acronyms, “Self-Organizing Networks”, http://www.3gpp.org/technologies/keywords-acronyms/105-son
[36] 4G Americas, “Self-Optimizing Networks in 3GPP Release 11: The Benefits of SON in LTE”, October 2013.
[37] Michael Behringer, Max Pritikin, Steinthor Bjarnason, Alex Clemm, “A Framework for Autonomic Networking”, 88th IETF, 9 Nov 2013.
[38] S. Buljore, H. Harada, P. Houze, K. Tsagkaris, V. Ivanov, K. Nolte, T. Farnham, O. Holland, M. Stamatelatos, "IEEE P1900.4 Standard: Reconfiguration of multi-radio systems," IEEE Region 8 International Conference on Computational Technologies in Electrical and Electronics Engineering, 2008. SIBIRCON 2008, pp.413-417, 21-25 July 2008.
[39] O. Holland et al., “Development of a Radio Enabler for Recon- figuration Management within the IEEE P1900.4 Working Group,” IEEE DySPAN 2007, Dublin, Ireland, April 2007.
[40] 1900.4a-2011 - IEEE Standard for Architectural Building Blocks Enabling Network-Device Distributed Decision Making for Optimized Radio Resource Usage in Heterogeneous Wireless Access Networks

Amendment 1: Architecture and Interfaces for Dynamic Spectrum Access Networks in White Space Frequency Bands

[41] IEEE Standards Association, IEEE 1900.5 Working Group (WG) on Policy Language and Architectures for Managing Cognitive Radio for Dynamic Spectrum Access Applications, http://grouper.ieee.org/groups/dyspan/5/

[42] A. Greenberg, G. Hjalmtysson, D.A. Maltz, et al., "A clean slate 4D approach to network control and management," ACM SIGCOMM Computer Communication Review, vol. 35, no. 5, pp. 41-54, 2005.

[43] A. M. Hadjiantonis and G. Pavlou, "Policy-based self-management of hybrid ad hoc networks for dynamic channel configuration," in Proc. 11th IEEE/IFIP Network Operations and Management Symposium (NOMS). Salvador - Bahia - Brazil, pp. 433-440, April 2008

[44] M. Johnsson, J. Huusko, T. Frantti, F. U. Andersen, T. M. T. Nguyen, and M. Ponce de Leon, "Towards a new architectural framework – the Nth stratum concept," in Proc. 4th International ICST Mobile Multimedia Communications Conference, ICST, May 2008.

[45] FP7 ACCORDANCE, WP5, D5.5, "Centralized wireless MAC: Definition of architectures making use of the centralized processing. Assessment of performance and complexity", April 2012.

[46] C. Kappler, P. Poyhonen, M. Johnsson, and S. Schmid, "Dynamic Network Composition for Beyond 3G networks – A 3GPP Viewpoint," IEEE Network, vol. 21, no. 1, pp. 47–52, 2007.

[47] FP6 ANA, WP1, D1.4, "ANA Blueprint – First Version", February 2007.

[48] FP7 ARTIST4G, WP4, D4.3, "Final conclusions on RAN architecture", June, 2012.

[49] A. Galis, S. Denazis, A. Bassi, et al., "Management Architecture and Systems for Future Internet Networks", in Towards the Future Internet – A European Research Perspective, G. Tselentis, J. Domingue, A. Galis, et al., Eds, Netherlands: IOS Press, pp. 95-104, 2009.

[50] FP7 BuNGee, WP1, D1.3, "Final BuNGee Architecture", January 2012.

[51] A. Manzalini, F. Zambonelli, "Towards Autonomic and Situation-Aware Communication Services: the CASCADAS Vision". Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06), 2006, pp. 383-388.

[52] H. Ballani and P. Francis, "CONMan: a step towards network manageability," ACM SIGCOMM Computer Communication Review, vol. 37, no. 4, pp. 205-216, 2007.

[53] FP7 CONSERN, WP3, D3.3, "Energy-Efficient Cooperative Decision and Control Schemes", August 2011.

[54] FP7 CONSERN, WP4, D4.4, "Documentation of final self-growing architecture, functions, interfaces and procedures", September 2012.

[55] K.Nolte, A.Kaloxylos, K.Tsagkaris, et al, "The E3 architecture: Enabling future cellular networks with cognitive and self-x capabilities", International Journal Of Network Management, vol. 21, no. 5, pp. 360–383, 2011.

[56] FP7 EARTH, WP3, D3.3, "Final Report on Green Network Technologies", July 2012.

[57] B. Jennings, R. Brennan, W. Donnelly, S. N. Foley, D. Lewis, D. O'Sullivan, J. Strassner, and S. van der Meer, "Challenges for federated, autonomic network management in the Future Internet," in Proc. 1st IFIP/IEEE International Workshop on Management of the Future Internet, pp. 87–92, 2009.

[58] B. Jennings, K. C. Feeney, R. Brennan, S. Balasubramaniam, D. Botvich, and S. van der Meer, Autonomic Network Management Principles: From Concepts to Applications. Academic Press, ch.

Federating Autonomic Network Management Systems for Flexible Control of End-to-end Communications Services, pp. 101–118, 2010.
[59] B. Jennings, S. Van der Meer, S. Balasubramaniam, et al., “Towards Autonomic Management of Communications Networks”, IEEE Commun. Mag. vol. 45, no. 10, pp. 112–121, 2007.
[60] FP7 METIS 2020, WP6, IR6.1, “Methodology for System Architecture Investigations”, May 2013.
[61] Y. Yemini, A. Konstantinou, D. Florissi, “NESTOR: an architecture for network self-management and organization”, IEEE Journal on Selected Areas in Communications, vol.18, no.5, pp.758-766, 2000.
[62] FP7 SACRA, WP1, D1.3, “SACRA System Recommendation Definition”, November 2012.
[63] FP7 SACRA, WP3, D3.4, “Cooperative and Cognitive RRM components and architecture for the SACRA scenarios”, November 2012.
[64] A. Kousaridas, G. Nguengang, J. Boite, et al., “An experimental path towards Self-Management for Future Internet Environments”, in Towards the Future Internet - Emerging Trends from European Research, G. Tselentis, A. Galis, A. Gavras, et al., Eds, Netherlands: IOS Press, pp. 95 – 104, 2010.
[65] Bionets, http://www.bionets.eu
[66] Socrates FP7 Project, http://www.fp7-socrates.org
[67] A. Galani, K. Tsagkaris, P. Demestichas, G. Nguengang, I.G. BenYahia, M. Stamatelatos, E. Kosmatos, A. Kaloxylos, L. Ciavaglia, "Core functional and network empower mechanisms of an operator-driven, framework for unifying autonomic network and service management," 2012 IEEE 17th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), pp.191-195, 17-19 Sept. 2012.
[68] FP6 WINNER I, WP7, D7.6, “WINNER System Concept Description”, October 2005.
[69] FP6 WINNER II, WP6, D6.13, “WINNER II System Concept Description”, January 2008.
[70] FP6 WINNER II, WP4, D4.8, “Integration of cooperation on WINNER II System Concept”, November 2007.
[71] FP7 WINNER+, WP2, D2.2, “Enabling Techniques for LTE-A and beyond”, July 2007.
[72] Z. Movahedi, M. Ayari, R. Langar, et al., “A Survey of Autonomic Network Architectures and Evaluation Criteria”, IEEE Trans. Commun. Surveys & Tutorials, vol. 14, no. 2, pp. 464-490, May 2012
[73] P. Makris, D. N. Skoutas, C. Skianis, “A Survey on Context-Aware Mobile and Wireless Networking: On Networking and Computing Environments' Integration”. IEEE Communications Surveys and Tutorials, vol. 15, issue 1, pp. 362-386, 2013.
[74] B. Meyer, “Applying ‘Design by Contract’ ”, Computer, vol.25, issue10, pp. 40–51, 1992.
[75] R. L. Ackoff, “From Data to Wisdom”, Journal of Applies Systems Analysis, vol. 16, pp. 3-9, 1989.
[76] Paul Gray, “Knowledge Management Overview”, University of California, 2000.
[77] A. Mihailovic, I.P. Chochliouros, E. Georgiadou, A.S. Spiliopoulou, E. Sfakianakis, M. Belesioti, G. Nguengang, J. Borgel, N. Alonistioti, "Situation awareness mechanisms for cognitive networks," International Conference on Ultra Modern Telecommunications & Workshops, 2009. ICUMT '09, pp.1-6, 12-14 Oct. 2009
[78] A. M. Serban, J. Luan, “Overview of Knowledge Management”, New Directions for institutional Research, Wiley Periodicals, no. 113, Spring 2002.

[79] A. Kousaridas, C. Polychronopoulos, N. Alonistioti, A. Marikar, J. Mödeker, A. Mihailovic, G. Agapiou, I. Chochliouros, G. Heliotis, “Future Internet elements: cognition and self-management design issues”, In Proc. ICST/ACM 2nd International Conference on Autonomic Computing and Communication Systems, 2008.

[80] M.R. Endsley, “Toward a theory of situation awareness in dynamic systems”, Human Factors 37(1), 1995, pp.32-64.

[81] H. Parvar, M. N. Fesharaki, B. Moshiri, “Shared Situation Awareness Architecture (S2A2) for Network Centric Disaster Management (NCDM)”, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 2, July 2012

[82] M. Smirnov, T. Zseby, and R. Roth, “Situation aware composition of low- level traffic handling functions”, Proceedings of the 1st IEEE International Workshop on Modelling Autonomic Communications Environments (MACE2006), Dublin, Ireland, October 25-26, 2006.

[83] T. Springer, P. Wustmann, I. Braun, W. Dargie, M. Berger, “A Comprehensive Approach for Situation-Awareness based on Sensing and Reasoning about Context”, Ubiquitous Intelligence and Computing Lecture Notes in Computer Science, Vol. 5061, pp 143-157, 2008.

[84] T. Raptis, C. Polychronopoulos, A. Kousaridas, P. Spapis, V. Gazis, N. Alonistioti, I. Chochliouros, "Technological Enablers for Self-Manageable Future Internet Elements", In Proc. COMPUTATIONWORLD, 2009, pp.499-504.

[85] I.P. Chochliouros, E. Sfakianakis, A. Kousaridas, J. Modeker, D. Wagner, A.S. Spiliopoulou, G. Agapiou, et al. “Design and Development of Essential Use-Cases for Self-Management in Future Internet Wireless Networks”, In Proc. MOBILIGHT, 2010.

[86] A. Mihailovic, G. Nguengang, A. Kousaridas, M. Israel, V. Conan, et al., "An approach for designing cognitive self-managed Future Internet," In Proc. Future Network and Mobile Summit, Jun 2010, pp.1-9.

[87] A. Kousaridas, G. Nguengang, J. Boite, et al., “An experimental path towards Self-Management for Future Internet Environments”, in Towards the Future Internet - Emerging Trends from European Research, G. Tselentis, A. Galis, A. Gavras, et al., Eds, Netherlands: IOS Press, 2010, pp. 95 – 104.

[88] Paik Incheon, Ryohei Komiya, Chen Wuhui, Lee Kyeongmu, "Active situation awareness framework for social network services," 2012 4th International Conference on Awareness Science and Technology (iCAST), pp.54-57, 21-24 Aug. 2012.

[89] Paik Incheon, Ryohei Komiya, "Active Situation Awareness on Web APIs for Information on Social Network Services," 2012 IEEE First International Conference on Services Economics (SE), pp.68-69, 24-29 June 2012.

[90] I. Paik, T. Tanaka, H. Ohashi, Chen Wuhui, "Big Data Infrastructure for Active Situation Awareness on Social Network Services," 2013 IEEE International Congress on Big Data (BigData Congress), , pp.411-412, June 27 2013-July 2 2013.

[91] D. Ragbir, P. Mohan, "Using Social Networking and Virtual Reality to Harness Creativity in Students in E-learning Situations," 2012 IEEE 12th International Conference on Advanced Learning Technologies (ICALT), pp.734-735, 4-6 July 2012.

[92] H. Parvar, M.N. Fesharaki, B. Moshiri, "Shared Situation Awareness System Architecture for Network Centric Environment Decision Making," 2010 Second International Conference on Computer and Network Technology (ICCNT), pp.372-376, 23-25 April 2010.

[93] N. Baumgartner, W. Retschitzegger, "Towards a Situation Awareness Framework Based on Primitive Relations," Information, Decision and Control, 2007, (IDC '07), pp.291-295, 12-14 Feb. 2007.
[94] Cheol Young Park, Kathryn Blackmond Laskey, Paulo C. G. Costa, Shou Matsumoto, “Multi-Entity Bayesian Networks Learning For Hybrid Variables In Situation Awareness”, 16th International Conference on Information Fusion, Istanbul, Turkey, July 9-12, 2013
[95] K. B. Laskey, MEBN: A Language for First-Order Bayesian Knowledge Bases. Artificial Intelligence, vol. 172, pp. 2-3, 2008.
[96] R. Komiva, Paik Incheon, Ryu Keunho, "Customizable active situation awareness framework based on meta-process in ontology," 2013 International Joint Conference on Awareness Science and Technology and Ubi-Media Computing (iCAST-UMEDIA), pp.114-120, 2-4 Nov. 2013.
[97] N. Nwiabu, I. Allison, P. Holt, P. Lowit, B. Oyeneyin, "Case-based situation awareness," 2012 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), pp.22-29, 6-8 March 2012.
[98] S. Liao, “Case-based decision support system: Architecture for simulating military command and control”, European Journal of operational Research, Vol. 123, pp. 558–567, 2000.
[99] B. Peacock, K.H. Chai, "A fuzzy model of knowledge awareness," 2012 IEEE International Conference on Management of Innovation and Technology (ICMIT), pp.220-225, 11-13 June 2012.
[100] A.P. Kärkkäinen, "Improving situation awareness in cognitive networks using the Self-Organizing Map," 2011 IEEE First International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), pp.40-47, 22-24 Feb. 2011.
[101] Feng Ding, Shuai Zhang, Heng Wang, Huang Songhua, Jun Tao, "Cooperative Agent-Based QoS Framework for Heterogeneous Networks," International Conference on Information Technology, Computer Engineering and Management Sciences (ICM), 2011, Vol.2, pp.214-217, 24-25 Sept. 2011.
[102] G. W. Ng, K. H. Ng, K. H. Tan, C. H. K. Goh, “The Ultimate Challenge of Commander’s Decision Aids: The Cognition Based Dynamic Reasoning Machine”, in Proceedings of 25th Army Science Conference, Orlando, Florida, Nov 2006.
[103] Pan Xinghao, Teow Loo-Ni, Tan Kheng Hwee, J.H.B Ang, Gee-Wah Ng, "A cognitive system for adaptive decision making," 15th International Conference on Information Fusion (FUSION), 2012, pp.1323-1329, 9-12 July 2012.
[104] J. Preden, L. Motus, M. Meriste, A. Riid, "Situation awareness for networked systems," IEEE First International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2011, pp.123-130, 22-24 Feb. 2011.
[105] Kendra Cherry, “Perception and the Perceptual Process”, http://psychology.about.com/od/sensationandperception/ss/perceptproc.htm
[106] P. Barnaghi, F. Ganz, C. Henson, A. Sheth, "Computing perception from sensor data," 2012 IEEE Sensors, pp.1-4, 28-31 Oct. 2012
[107] A. Sheth, "Computing for Human Experience: Semantics-Empowered Sensors, Services, and Social Computing on the Ubiquitous Web," IEEE Internet Computing, vol. 14, no. 1, pp. 88-91, 2010.
[108] Panagis Magdalinos, Apostolos Kousaridas, Panagiotis Spapis, Giorgos Katsikas, Nancy Alonistioti, “Enhancing a Fuzzy Logic Inference Engine through Machine Learning for a Self-

Managed Network", ACM Springer Mobile Networks and Applications (MONET) MONAMI, vol. 16, Issue 4, pp. 475-489, August 2011

[109] CISCO, "CSS Content Load-Balancing Configuration Guide", November 2006.

[110] CISCO, "Solution Overview, Airport Security: IP-Based Solutions for Increased Situational Awareness and Operational Effectiveness", 2011.

[111] ERICSSON, "Traffic and Market report – on the pulse of the networked society", June 2012.

[112] S. Nádas, S. Rácz, Z. Nagy, S. Molnar, "Providing Congestion Control in the Iub Transport Network for HSDPA," IEEE Global Telecommunications Conference, 2007. GLOBECOM '07, pp.5293-5297, 26-30 Nov. 2007

[113] L. A. Zadeh, "Fuzzy sets", Information and Control, no.8, pp.338–353, 1965.

[114] L. A. Zadeh, "Making computers think like people", IEEE. Spectrum, vol. 8, pp. 26-32, 1984.

[115] L. A. Zadeh, "Outline of A New Approach to the Analysis of Complex Systems and Decision Processes", IEEE Transactions on Systems, Man, and Cybernetics, vol. 3, 1973.

[116] Fuzzy Logic and its rules, "Fuzzy Logic Introduction", http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol2/jp6/article2.html

[117] Claudio Moraga, "Introduction to Fuzzy Logic", FACTA UNIVERSITATIS (NIS), Elec. Energ., vol. 18, No. 2, pp. 319-328, August 2005.

[118] M. Hellmann, "Fuzzy Logic Introduction", doi=10.1.1.85.9757

[119] O. E. Falowo, H. A. Chan, "Fuzzy Logic Based Call Admission Control for Next Generation Wireless Networks" 3rd International Symposium on Wireless Communication Systems, 2006. ISWCS '06, pp. 574-578, 6-8 Sept. 2006.

[120] O. E. Falowo, H. A. Chan, "Multiple-call handover decisions using fuzzy MCGDM in heterogeneous wireless networks", Military Communications Conference 2011 - MILCOM 2011, pp.1909-1914, 7-10 Nov. 2011.

[121] O. E. Falowo, H. A. Chan, "Stochastic model of a user-centric joint call admission control scheme for heterogeneous wireless networks," 5th International Conference on Broadband Communications, Networks and Systems, 2008. BROADNETS 2008, pp.241-248, 8-11 Sept. 2008.

[122] K. Kapitanovaa, Sang H. Sona, Kyoung-Don Kangb, "Using fuzzy logic for robust event detection in wireless sensor networks", Ad Hoc Networks, vol. 10, issue 4, June 2012.

[123] V. Chandola, A. Banerjee, V. Kumar, "Anomaly Detection: A Survey", technical report, University of Minnesota.

[124] A. Merentitis, D. Triantafyllopoulou, "Resource Allocation with MAC Layer Node Cooperation in Cognitive Radio Networks," International Journal of Digital Multimedia Broadcasting, vol. 2010, Article ID 458636, 11 pages, 2010.

[125] E. H. Mamdani, S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," International Journal of Man-Machine Studies, Vol. 7, No. 1, pp. 1-13, 1975.

[126] M. Sugeno, "Industrial applications of fuzzy control", Elsevier Science Pub. Co., 1985.

[127] T. Takagi, M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control" IEEE transactions on systems, man, and cybernetics, vol. 15, no. 1, pp. 116{132, 1985.

[128] J.S. Roger Jang Ned Gulley, "MATLAB Fuzzy Logic Toolbox User's Guide", 2014.

[129] A. Kawamura, N. Watanabe, H. Okada, K. Asakawa, "A prototype of neuro-fuzzy cooperation system," IEEE International Conference on Fuzzy Systems, 1992, pp.1275-1282, 8-12 Mar 1992.

[130] Stylianos Georgoulas, Klaus Moessner, Alexis Mansour, Menelaos Pissarides, Panagiotis Spapis, “A Fuzzy Reinforcement Learning Approach for Pre-Congestion Notification Based Admission Control”, 6th International Conference on Autonomous Infrastructure, Management and Security (AIMS 2012), June 04-08, 2012.

[131] R. Arapoglou, K. Kolomvatsos, S. Hadjiefthymiades, "Buyer agent decision process based on automatic fuzzy rules generation methods," 2010 IEEE International Conference on Fuzzy Systems (FUZZ), pp.1-8, 18-23 July 2010.

[132] Zhang Fu, M. Z. Ma, Yan Li, "A fuzzy ontology approach for representing Fuzzy Petri Nets," 2012 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp.1-8, 10-15 June 2012

[133] H. Ghorbel, A. Bahri, R. Bouaziz, "Fuzzy ontologies building method: Fuzzy OntoMethodology," Fuzzy Information Processing Society (NAFIPS), 2010 Annual Meeting of the North American, pp.1-8, 12-14 July 2010

[134] Panagis Magdalinos, Dimitris Makris, Panagiotis Spapis, Christos Papazafeiropoulos, Apostolos Kousaridas, Makis Stamatelatos and Nancy Alonistioti, “Coverage and Capacity Optimization of Self-Managed Future Internet Wireless Networks”, ServiceWave 2010, Ghent and “Towards a Service-Based Internet Lecture Notes in Computer Science, 2010”, Volume 6481, 2010.

[135] A. Merentitis, A. Kaloxylos, M. Stamatelatos, N. Alonistioti, "Optimal periodic radio sensing and low energy reasoning for cognitive devices," 15th IEEE Mediterranean Electrotechnical Conference 2010, MELECON 2010, pp.470-475, 26-28 April 2010.

[136] Jiawei Han, Micheline Kamber, Jian Pei, “Data Mining: Concepts and Techniques”, 3rd Edition, ISBN-9780123814791, 2011.

[137] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, “From data mining to knowledge discovery in databases”, AI magazine, vol. 17, issue 3, pp. 37–54, 1996.

[138] Won Lee Seek, L. Kerschberg, "A methodology and life cycle model for data mining and knowledge discovery in precision agriculture," 1998 IEEE International Conference on Systems, Man, and Cybernetics, vol.3, pp. 2882-2887, Oct 1998.

[139] R. J. A. Little, D. B. Rubin, “Statistical analysis with missing data”, New York, Wiley, 1987.

[140] P. D. Allison, “Missing Data Thousand Oaks”, CA: Sage Publications, 2001.

[141] A. N. Barladi, C. K. Enders, “An introduction to modern missing data analyses”, Journal of School Psychology, vol. 48, pp. 5-37, 2010.

[142] D. C. Howell, “The analysis of missing data”, Handbook of Social Science Methodology, London, 2007

[143] Gregory Piatetsky-Shapiro, “From Data Mining to Knowledge Discovery: An Introduction”, http://www.kdnuggets.com/data_mining_course/x4-data-mining-to-knowledge-discovery.ppt

[144] Yang Zhang, N. Meranthia, P. Havinga, “Outlier Detection Techniques for Wireless Sensor Networks”, IEEE Communication Surveys & Tutorials, Second Quarter 2010.

[145] Columbia lectures, “Chapter 3: Supervised Learning”, http://www.cs.uic.edu/~liub/teach/cs583-fall-05/CS583-supervised-learning.ppt

[146] Columbia lectures, “KERNELS”, http://stat.columbia.edu/~porbanz/teaching/W4240/slides_lecture6.pdf

[147] Do Thanh-Nghi, F. Poulet, "Kernel-based Algorithms and Visualization for Interval Data Mining," Sixth IEEE International Conference on Data Mining Workshops, 2006. ICDM Workshops 2006, pp.295-299, Dec. 2006.

[148] Columbia lectures, “Chapter 4: Unsupervised Learning”, http://www.cs.uic.edu/~liub/teach/cs583-fall-05/CS583-unsupervised-learning.ppt

[149] P. Drineas, M. A. Frieze, R. Kannan, S. Vempala, V. Viany, “Clustering large graphs via the singular value decomposition”, Machine Learning, vol. 56, issue 1-3, pp. 9–33. 2004.

[150] R. Ostrovsky, Y. Rabani, L. J. Schulman, C. Swamy. “The effectiveness of lloyd-type methods for the k-means problem”. In FOCS ’06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS’06). IEEE Computer Society, Washington, DC, USA, 165–176, 2006.

[151] S. B. Kotsiantis, “Supervised Machine Learning: A Review of Classification Techniques”, Informatica 31, 2007.

[152] Panagiotis Spapis, Rouzbeh Razavi, Stylianos Georgoulas, Zwi Altman, Richard Chombes, Aimilia Bantouna, “On the role of learning in autonomic network management: the UniverSelf project approach”, Future Network & Mobile Summit, Berlin, July, 2012

[153] R. Combes, Z. Altman, E. Altman, "Self-organizing relays in LTE networks: Queuing analysis and algorithms," 7th International Conference on Network and Service Management (CNSM), 2011, pp.1-8, 24-28 Oct. 2011.

[154] L. P. Kaelbling, M. L. Littman and A. W. Moore, "Reinforcement Learning: A Survey", Volume 4, pages 237-285, 1996.

[155] Richard S. Sutton, “Reinforcement Learning”, http://129.237.66.221/Sutton-99-MITECS.pdf, 1999.

[156] Stanford edu, “Chapter 9 Temporal-Difference Learning”, http://www.stanford.edu/group/pdplab/pdphandbook/handbookch10.html

[157] R. S. Sutton, A. G. Barto, “Reinforcement Learning: An Introduction”, MIT Press, 1998.

[158] Quality of Service for Voice over IP, CISCO, 6/30/2001

[159] Latency and QoS for Voice over IP, SANS Institute 2004

[160] http://www.slac.stanford.edu/comp/net/wan-mon/tutorial.html

[161] M. Bouet, G. Nguengang, V. Conan, A. Kousaridas, P. Spapis, N. Alonistioti, "Embedding cognition in wireless network management: an experimental perspective", IEEE Communications Magazine, vol.50, no.12, pp.150,160, December 2012

[162] Tee A., Cleveland J.R, Chang J.W., Implication of End-user QoS requirements on PHY & MAC, IEEE 802.20 Working Group on Mobile Broadband Wireless Access, http://www.ieee802.org/20/Contribs/C802.20-03-106.ppt

[163] Andrews N., Kondareddy Y., Agrawal P. (2010) Channel management in collocated WiFi-WiMAX networks, System Theory, 42nd Southeastern Symposium on, pp. 133-137, doi: 10.1109/SSST.2010.5442848

[164] AN-100U/UX Single Sector Wireless Access Base Station User Manual, RedMAX, Redline Communications, 2008

[165] Java Programming Language, http://www.oracle.com/technetwork/java/index.html

[166] Soekris Engineering net5501, http://www.soekris.com/net5501.htm

[167] VLC: open-source multimedia framework, player and server, http://www.videolan.org/vlc

[168] Alcatel Lucent - Application Note “The Impact of Small Cells on MME Signaling-Methods to reduce and optimize MME core signaling caused by small cells”, October, 2013 Alcatel-Lucent, NP2013092761EN

[169] J. Huang, R. Berry, and M. Honig,“Spectrum sharing with distributed interference compensation”, First IEEE International Symposium New Frontiers in Dynamic Spectrum Access Networks (DySPAN), 2005.

[170] A. Merentitis, and D. Triantafyllopoulou, “Transmission Power Regulation in Cooperative Cognitive Radio Systems Under Uncertainties”, IEEE International Symposium on Wireless Pervasive Computing (ISWPC), 2010.

[171] Panagiotis Spapis, George Katsikas, Makis Stamatelatos, Konstantinos Chatzikokolakis, Roi Arapoglou, Nancy Alonistioti, "Learning Enhanced Environment Perception for Cooperative Power Control", International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, UBICOMM 2011, Lisbon, Portugal, November, 2011.

[172] http://homesupport.cisco.com/en-eu/support/adapters/WUSB54GC , Dec. 2012.

[173] http://www.pcengines.ch/pdf/wlm54ag23.pdf , Dec. 2012.

[174] http://jfuzzylogic.sourceforge.net/html/index.html , Dec. 2012.

[175] http://mina.apache.org/ , Dec. 2012.

[176] A. Bose, and F. H. Chuan, "A practical path loss model for indoor WiFi positioning enhancement," 6th International Conference on Information, Communications & Signal Processing, 2007, pp.1-5, 2007.

[177] J. S. Seybold, “Introduction to RF propagation”, Wiley, ISBN-13 978-0-471-65596-1.

[178] Inaltekin, H., "Gaussian Approximation for the Wireless Multi-access Interference Distribution," IEEE Transactions on Signal Processing, vol.PP, no.99, pp.1, 0

[179] Yong Sheng, Tan, K., Guanling Chen, Kotz, D., Campbell, A., "Detecting 802.11 MAC Layer Spoofing Using Received Signal Strength," IEEE 27th Conference on Computer Communications INFOCOM 2008. pp.1768-1776, 13-18 April 2008

[180] Oliver Johnson, "Information Theory And The Central Limit Theorem" , Imperial College, 2004, ISBN 1-86094-473-6

[181] Na Luo, Feng Qian, "Evolutionary algorithm using kernel density estimation model in continuous domain," Asian Control Conference, 2009. ASCC 2009. 7th, pp.1526-1531, 27-29 Aug. 2009

[182] O. N. C. Yilmaz, C. Wijting, P. Lundén, J. Hämälainen, "Optimized Mobile Connectivity for Bandwidth-Hungry, Delay-Tolerant Cloud Services toward 5G," International Symposium on Wireless Communication Systems (ISWCS), 9th International, Barcelona, Spain, August 2014