



Πανεπιστήμιο Πατρών
Πολυτεχνική Σχολή
Τμήμα Μηχανικών Η/Υ και Πληροφορικής

Διπλωματική Εργασία

«Σχεδιασμός και Ανάλυση Κατανεμημένων Αλγορίθμων για τον Έλεγχο και Συντονισμό Ασύρματων, Κινητών Δικτύων Αισθητήρων »

Οικονόμου Γεώργιος
ΑΜ: 3278

Επιβλέπων : Καθηγητής Παύλος Σπυράκης

Συνεπιβλέπων : Δρ. Ιωάννης Χατζηγιαννάκης

*Στον
Παναθηναϊκό*

Ευχαριστίες

Πριν παρουσιάσω το περιεχόμενο, την υλοποίηση και τελικά τα συμπεράσματα αυτής της διπλωματικής, αισθάνομαι υποχρεωμένος να αναφερθώ σε όσους με βοήθησαν στην εκπόνηση της.

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή του Τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής κ.Παύλο Σπυράκη, ο οποίος μου έδωσε την δυνατότητα να εντρυφήσω στον γνωστικό πεδίο, που πραγματεύεται η παρούσα διπλωματική και με τίμησε με την επιλογή του.

Ακόμα να δηλώσω ότι περιποιεί τιμή η συνεργασία μου με τον διδάσκοντα Δρ. Ιωάννη Χατζηγιαννάκη , ο οποίος με την με την βαθεία εμπειρία του και τις γνώσεις του, με καθοδήγησε κάθολη την διάρκεια της προσπάθειας μου. Η συχνή επαφή και η ουσιαστική επικοινωνία οδήγησαν στην υπερπήδηση όλων των εμποδίων και στην πραγματικά ευχάριστη αναζήτηση της γνώσης.

Ακόμα, νιώθω βαθεία την ανάγκη να υπογραμμίσω και να εξάρω την κομβικής σημασίας συμβολή του επιστήθιου φίλου και ικανότατου μεταπτυχιακού Ορέστη Ακριβόπουλου. Διότι στάθηκε αειφόρο πνεύμα και πολύτιμος αρωγός, τόσο κατά τη διάρκεια της υλοποίησης όσο και της συγγραφής της διπλωματικής εργασίας.

Τέλος θέλω να εκφράσω την ευγνωμοσύνη μου στο μεταπτυχιακό Χρήστο Κονίνη για το υλικό άλλα κα τις πολύτιμες συμβουλές που απλόχερα μου παρείχε.

Περίληψη

Σε αυτή τη διπλωματική ασχολούμαι με την ανάλυση του Directed Diffusion πρωτοκόλλου. Αρχικά αναλύεται και αξιολογείται θεωρητικά. Στη συνέχεια υλοποιείται το πρωτόκολλο αυτό στον εξομοιωτή Shawn. Προηγουμένως έχει γίνει μια ανάλυση των ιδιοτήτων του εξομοιωτή και των δυνατοτήτων του. Τέλος γίνεται μια πειραματική αξιολόγηση του πρωτοκόλλου. Το πρωτόκολλο έχει υλοποιηθεί σε Cplusplus. Το πρωτόκολλο έχει ελεγχθεί για την απόδοση του και τη σωστή του λειτουργία.

Περιεχόμενα

Ευχαριστίες	3
Περίληψη	4
1. Εισαγωγή	9
1.1. Κίνητρο και σημασία του θέματος	8
1.2. Στόχοι της διπλωματικής εργασίας	8
1.3. Συνεισφορά της διπλωματικής εργασίας.....	9
1.4. Δομή της διπλωματικής εργασίας	9
2. Το όραμα της περιβάλλουσας νοημοσύνης	10
2.1. Εισαγωγή	10
2.2. Παραδείγματα Εφαρμογών	12
2.3. Τύποι Εφαρμογών	14
2.4. Προκλήσεις για τα Ασύρματα Δίκτυα Αισθητήρων	15
2.4.1. Απαιτούμενα Χαρακτηριστικά.....	16
2.4.2. Απαιτούμενοι Μηχανισμοί.....	17
3. Directed Diffusion (κατευθυνόμενη διάχυση).....	20
3.1. Εισαγωγή.....	20
3.2. Χαρακτηριστικά στοιχεία του Directed Diffusion	21
3.2.1. Naming	22
3.2.2. Ενδιαφέροντα	23
3.2.3. Gradients	26
3.2.4. Διάδοση δεδομένων.....	27
3.2.5. Τεχνικές ενίσχυσης στην καθιέρωση και στην αποκοπή μονοπατιών	28
3.2.6. Ανακεφαλαίωση	31
3.3. Αναλυτική Αξιολόγηση.....	33
3.3.1. Flooding (Πλημμύρα).....	34
3.3.2. Omniscient Multicast	35
3.3.3. Directed Diffusion	36

3.3.4.	Σύγκριση.....	37
3.4.	Εξομοίωση.....	38
3.4.1.	Στόχοι, Μετρικές και Μεθοδολογία.....	38
3.4.2.	Συγκριτική Αξιολόγηση.....	40
3.4.3.	Αντίκτυπος των Δυναμικών Φαινομένων.....	42
3.4.4.	Αντίκτυπος της Επεξεργασίας Δεδομένων και της Αρνητικής Ενίσχυσης.....	44
3.4.5.	Ανάλυση Ευαισθησίας.....	44
3.5.	Παρόμοιες Εργασίες.....	45
3.6.	Συμπέρασμα.....	46
4.	Περιγραφή και παρουσίαση του Shawn και της υλοποίησης.....	47
4.1.	Εισαγωγή στον Shawn.....	47
4.2.	Αρχιτεκτονική του Shawn.....	50
4.2.1.	Τα μοντέλα (Models).....	51
4.2.2.	Διαδοχέας (Sequencer).....	54
4.2.3.	Περιβάλλον Υλοποίησης.....	56
4.3.	Οδηγίες για την εγκατάσταση του Shawn σε Unix συστήματα.....	56
4.4.	Βασικά βήματα για την πραγματοποίηση εξομοίωσης στον Shawn.....	64
4.4.1.	Παράθεση διαδοχικών βημάτων σε μια τυπική εξομοίωση.....	64
4.4.2.	Χρήση αρχείου διαμόρφωσης (Configuration File).....	68
4.4.3.	Χρησιμοποίηση πολλαπλών configuration αρχείων.....	69
5.	Κώδικας του Directed Diffusion στο Shawn.....	70
6.	Πειραματική Αξιολόγηση του Directed Diffusion στο Shawn.....	80
7.	Τελικά συμπεράσματα.....	88
8.	Βιβλιογραφία.....	89

Περιεχόμενα σχημάτων και πινάκων

Σχήμα 2.1:	Αισθητήρες	10
Σχήμα 2.2:	Εφαρμογές των αισθητήρων	12
Σχήμα 3.2:	Απλό σχήμα του DD	22
Σχήμα 3.2.3:	Παρουσιάζονται διαφορετικές πτυχές του Diffusion	26
Σχήμα 3.2.5:	Αρνητική ενίσχυση	30
Σχήμα 3.3:	Παραδείγματα τετατραγωνικής grid τοπολογίας	34
Σχήμα 3.3.4:	Αντίκτυπος διάφορων παραμέτρων του DD και omniscient multicast	38
Σχήμα 3.4.1:	Σύγκριση του DD με flooding και omniscient multicast	40
Σχήμα 3.4.2:	Αντίκτυπος των αποτυχημένων κόμβων στο DD	42
Σχήμα 3.4.4:	Αντίκτυπος διαφόρων συντελεστών του DD	43
Σχήμα 4.1:	Κύκλος Ανάπτυξης του Εργαλείου Shawn	49
Σχήμα 4.2:	Στοιχεία αρχιτεκτονικής του Shawn	50
Σχήμα 4.2.1.α:	Υλοποιήσεις του communication model	51
Σχήμα 4.2.1.β:	Υλοποιήσεις του edge model	53
Σχήμα 4.2.1.γ:	Τα βασικά transmission models	54
Σχήμα 4.2.2:	Event scheduler	55
Σχήματα 4.3	κεφαλαίου	57-63
Σχήματα 4.4	κεφαλαίου	64-67
Σχήμα 5.1:	Δενδρική αναπαράσταση της αντιμετώπισης του interests	72
Σχήμα 6:	Σύγκριση εξομοιωτών	80
Σχήματα 6	κεφαλαίου	81-87
Πίνακας 3.2.2:	Σχεδιαστικές επιλογές για τις διάφορες φάσεις του Diffusion	21
Πίνακας 6:	Συγκριτικό πρωτοκόλλων DD, TORA, AODV	87

1. Εισαγωγή

1.1. Κίνητρο και σημασία του θέματος

Στις μέρες μας η χρήση αισθητήρων έχει λάβει αν μη τι άλλο αξιοπρόσεκτες διαστάσεις. Πλειάδα εφαρμογών βασίζονται στη χρήση τους και οι πιθανοί τρόποι χρησιμοποίησης τους ποικίλουν σημαντικά. Το ενδιαφέρον έγκυται στο ότι, το πεδίο αυτό της επιστήμης των υπολογιστών βρίσκει άμεση εφαρμογή στη καθημερινή μας ζωή (π.χ. αισθητήρες σε κινητά, ευφυή κτίρια). Επίσης πολλές εφαρμογές ξεφεύγουν από τα τετρημένα και ακουμπούν πάνω σε ευαίσθητα και απαιτητικά θέματα (π.χ ιατρικά, στρατιωτικές εφαρμογές). Αρκετές από τις εφαρμογές που χρησιμοποιούν αισθητήρες, παρουσιάζονται εκτενέστερα και παρακάτω. Όμως και από αυτά τα λίγα, εύκολα γίνεται αντιληπτή η σημασία αυτού του ερευνητικού πεδίου. Όλες αυτές οι εφαρμογές και ακόμα περισσότερες που δεν φανταζόμαστε, έχουν στη καρδιά τους έναν ή συνδυασμό αλγορίθμων. Ο αλγόριθμος με τη σειρά του θα ελέγξει και θα καθοδηγήσει τα ασύρματα δίκτυα αισθητήρων. Οπότε η ορθότητα του αλγορίθμου, που υπάρχει ή σχεδιάζεται για της ανάγκες της εφαρμογής θα αποτελέσει τη στέρεα βάση για το κτίσιμο. Βέβαια μόνο η βάση δεν φτάνει, για αυτό προχωρούμε στη διαδικασία της ανάλυσης του αλγορίθμου. Αυτό γίνεται για να δούμε σε βάθος τις δυνατότητες του και να εκμεταλευτούμε τα δυνατά του σημεία στο έπακρο. Η βαθειά κατανόηση οδηγεί όχι μόνο σε ορθούς αλλά και σε αποτελεσματικούς αλγορίθμους. Αφού μιλάμε για πραγματικού χρόνου εφαρμογές αντιλαμβανόμαστε τη βαρύτητα του χαρακτηριστικού της αποτελεσματικότητας.

Επίκουρος στις παραπάνω προσπάθειες στέκεται μία ειδική κατηγορία προγραμμάτων λογισμικού. Αυτοί είναι οι εξομοιωτές. Οι εξομοιωτές θέτουν τα πρώτα όρια στη μαθηματική προσέγγιση, όρια που ίσως περιοριστούν ακόμα περισσότερο από το υλικό (hardware του σένσορα). Δίνουν έτσι μια απεικόνιση πολύ κόντα στη πραγματικότητα. Ταυτόχρονα μας παρέχουν την δυνατότητα πραγματοποίησης πολλών πειραμάτων και δοκιμών αποσφαλμάτωσης. Όταν πλέον ο αλγόριθμος έχει αναπτυχθεί σε ένα εξομοιωτή και έχει αξιολογηθεί, δίνεται να χρησιμοποιηθεί για την υποστήριξη συνθετότερων σεναρίων.

Λίγο έως πολύ οι παραπάνω παράγραφοι σκιαγράφησαν τη σημασία του θέματος αλλά του εργαλείου (του εξομοιωτή). Από τη σημασία προκύπτει πλέον καθαρά και το κίνητρο. Και μόνο ο χαρακτήρας ή το επεδικώμενο από μια εφαρμογή, μπορούν να συγκινήσουν και να αποτελέσουν κίνητρο. Τέλος η ανάπτυξη λογισμικού για τέτοια εργαλεία παρουσιάζει ακρετές προκλήσεις.

1.2. Στόχοι της διπλωματικής εργασίας

Η παρούσα διπλωματική εργασία έχει ως στόχο την ανάλυση του Directed Diffusion πρωτοκόλλου/αλγορίθμου (με την ευρεία έννοια). Φιλοδοξεί να περιγράψει σε βάθος το πρωτόκολλο και να αναλύσει ξεχωριστά, όλα τα στοιχεία που το διαμορφώνουν. Ασχολείται εκτενώς πέρα από το βασικό κορμό του πρωτοκόλλου και με περιπτώσεις βελτιστοποίησης του. Ακόμα το συγκρίνει, για να αποδείξει ότι υπερτερεί έναντι άλλων κλασσικών πρωτοκόλλων σε ακρετούς τομείς (π.χ εξοικονόμηση ενέργειας). Γενικά θέλει να αναδείξει όλες τις περιπτώσεις, στις οποίες το Directed Diffusion δείχνει τις δυνατότητες του. Επίσης στόχος της διπλωματικής αποτελεί και η παρουσίαση

του εξομοιωτή Shawn. Όχι μόνο σε θεωρητικό επίπεδο αλλά και σε πρακτικό. Κορυφαίος στόχος είναι η ανάπτυξη του Directed Diffusion στον Shawn και η διενέργεια πειραματικής αξιολόγησης του. Τέλος σημαντική θεωρείται και η εξοικείωση με τεχνικές της γλώσσας Cplusplus αλλά και του τρόπου, που αναπτύσσονται εφαρμογές στο περιβάλλον ενός εξομοιωτή.

1.3. Συνεισφορά της διπλωματικής εργασίας

Στα πλαίσια της διπλωματικής παρουσιάζεται συνοπτικά το εύρος των εφαρμογών των αισθητήρων. Ακόμα πετυγχάνονται σε μεγάλο βαθμό οι στόχοι που τέθηκαν στον προηγούμενο κεφάλαιο, συνεισφέροντας τα αποτελέσματα. Με απλά λόγια παρουσιάζει σφαιρικά το Directed Diffusion. Δίνει κατανοητές οδηγίες χρήσης του εργαλείου Shawn, ταυτόχρονα παρουσιάζει την υλοποίηση του πρωτοκόλλου για ασύρματα δίκτυα αισθητήρων. Έτσι αποτελεί ένα καλό μπούσουλα για την εκμάθηση του εργαλείου και την συγγραφή σύνθετων πρωτοκόλλων. Συγκεκριμένα στο Directed Diffusion δίνεται μια στέρεη βάση για την περετέρο εκλέπτυνση του αλγορίθμου. Επίσης εξετάζει και αναλύει αρκετές πτυχές της διαδικασίας ανάπτυξης εφαρμογών. Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκαν εργαλεία, που διευκόλυναν την διαδικασία αλλά και εξασφάλισαν την ορθότητα.

1.4. Δομή της διπλωματικής εργασίας

Σε αυτήν την ενότητα θα παρουσιαστεί η δομή της διπλωματικής εργασίας. Αρχικά συναντούμε την εισαγωγή της διπλωματικής, η οποία περιλαμβάνει τη σημασία του θέματος, το κίνητρο ενασχόλησης, τους στόχους, την συνεισφορά και τέλος τη δομή της. Μετά περνάμε στο κυρίως σώμα της διπλωματικής. Πιο συγκεκριμένα στο **κεφάλαιο 2** κάνουμε μια γνωριμία με τα δίκτυα αισθητήρων. Βλέπουμε το σύνολο των εφαρμογών τους, αλλά και τις προκλήσεις που καλούνται να αντιμετωπίσουν. Στο **κεφάλαιο 3** παρουσιάζεται το Directed Diffusion το σύνολο των χαρακτηριστικών του, αξιολογείται σε βάθος και τέλος παρουσιάζονται μερικά σενάρια εξομοίωσης. Στο **κεφάλαιο 4** περιγράφεται η αρχιτεκτονική του Shawn, τα διάφορα μοντέλα στα οποία βασίζεται, οδηγίες εγκατάστασης και ένα απλό παράδειγμα εξομοίωσης. Στο **κεφάλαιο 5** εισάγουμε το Directed Diffusion στον Shawn και αναλύουμε τον κώδικα του. Στο **κεφάλαιο 6** κάνουμε πειραματική αξιολόγηση του Directed Diffusion σε πολλαπλές τοπολογίες. Τέλος γίνεται μια μικρή αποτίμηση και παρουσιάζεται η βιβλιογραφία.

2. Το όραμα της περιβάλλουσας νοημοσύνης

2.1. Εισαγωγή



Σχήμα 2.1: Αισθητήρες

Η πιο συνηθισμένη μορφή επεξεργασίας της πληροφορίας λαμβάνει χώρα σε γενικού σκοπού υπολογιστικές μηχανές, οι οποίες περιλαμβάνουν από κεντρικούς υπολογιστές μέχρι σύγχρονα laptop ή palmtop. Σε πλειάδα εφαρμογών, όπως οι εφαρμογές γραφείου, οι εν λόγω υπολογιστικές μηχανές επεξεργάζονται πληροφορίες που είναι προσανατολισμένες στον ανθρώπινο χρήστη του εκάστοτε συστήματος. Όμως δεν υπάρχουν καθόλου πληροφορίες που αφορούν το φυσικό περιβάλλον. Παρόλα αυτά υπάρχουν εφαρμογές οι οποίες τοποθετούν στο κέντρο του ενδιαφέροντος τους, το φυσικό περιβάλλον και προσπαθούν να έχουν υπό έλεγχο φυσικές διαδικασίες. Εδώ ο υπολογισμός είναι ενιαίος με τον έλεγχο. Συστήματα αυτού του είδους συνήθως δεν βασίζονται σε αλληλεπιδράσεις με τους ανθρώπους, αντιθέτως απαιτείται να δουλεύουν χωρίς αυτούς. Με την ανάπτυξη της τεχνολογίας όλο και περισσότερες συσκευές (όπως ένα πλυντήριο) αλλά και αναλώσιμα αγαθά θα ενσωματώνουν τέτοια συστήματα. Αργά ή γρήγορα ποικίλες συσκευές θα μαζεύουν και θα επεξεργάζονται πληροφορίες από διαφορετικές πηγές. Έτσι ελέγχουν τις φυσικές διαδικασίες αλλά και την αλληλεπίδραση τους με τον άνθρωπο. Με λίγα λόγια τα μοντέλα αλληλεπίδρασης ανθρώπου-ανθρώπου, ανθρώπου-μηχανής και μηχανής-μηχανής θα δώσουν τη θέση τους στο μοντέλο ανθρώπου-φυσικού περιβάλλοντος.

Για να αντιληφθεί κανείς το παραπάνω όραμα πρέπει να προσθέσει στον υπολογισμό και στον έλεγχο μια ακόμα διάσταση. Αυτή που δίνεται από την επικοινωνία. Όλες αυτές οι πηγές πληροφορίας χρειάζεται να μεταφέρουν την πληροφορία εκεί που είναι ανάγκη και ταυτόχρονα πρέπει να υπάρχει συνεργασία. Έτσι ώστε να αποτυπώνεται η ακριβής εικόνα του πραγματικού κόσμου.

Σε ορισμένες εφαρμογές όλα μπορούν να υλοποιηθούν από την υπάρχουσα ενσύρματη τεχνολογία. Για άλλες όμως η ανάγκη καλωδίωσης όλων αυτών των οντοτήτων συναντά πλήθος δυσκολιών. Αυτές εντοπίζονται στον χρηματικό κόστος, στη συντήρηση των καλωδίων, στην παρεμπόδιση της κίνησης των οντοτήτων, αλλά και στο να βρεθούν κοντά στον τόπο του φαινομένου που παρατηρούν. Για όλες αυτές τις περιπτώσεις χρησιμοποιούμε συσκευές ασύρματης επικοινωνίας. Απόρροια όλων αυτών, ήταν η εμφάνιση των αποκαλούμενων Ασυρμάτων Δικτύων Αισθητήρων, τα οποία αποτελούνται από ατομικούς κόμβους. Οι κόμβοι αυτοί είναι ικανοί να αλληλεπιδρούν με το περιβάλλον αισθανόμενοι ή ελέγχοντας τις φυσικές παραμέτρους. Οι κόμβοι συνεργάζονται για να φέρουν σε πέρας εγγραφές που πολύ πιθανόν να μην μπορούσαν από μόνοι τους. Η δυναμική των Ασυρμάτων Δικτύων Αισθητήρων έγκειται στο πλήθος και την ποικιλία των εφαρμογών πραγματικού χρόνου στις οποίες μπορούν να συμμετάσχουν. Η ευελιξία τους οδηγεί σε ενδιαφέρουσες ερευνητικές αλλά και πρακτικές προκλήσεις. Επίσης δεν υπάρχουν συγκεκριμένες προδιαγραφές που να ορίζουν ένα Ασύρματο Δίκτυο Αισθητήρων καθώς και μοναδικές τεχνικές σχεδιασμού τους. Παραδείγματος χάριν σε ορισμένες εφαρμογές Ασυρμάτων Δικτύων Αισθητήρων δεν υπάρχει καμία δυνατότητα ενσύρματης παροχής ενέργειας και αναγκαστικά λειτουργούν μόνο με μπαταρίες. Σε αυτές τις εφαρμογές προτιμώνται λύσεις που δεν είναι ενεργοβόρες. Αλλού πάλι το θέμα της τροφοδοσίας είναι δευτερεύον και λαμβάνονται υπόψη άλλες μετρικές. Τέτοιες μπορεί να είναι η ακρίβεια των μεταφερόμενων δεδομένων, το μέγεθος και το κόστος των κόμβων. Στενά συνδεδεμένη με το μέγεθος είναι και η μετρική της χωρητικότητας της μπαταρίας. Το κόστος συχνά συμβαδίζει με την ποιότητα του κόμβου και κατ' επέκταση με τη ακρίβεια. Επιπλέον καλούμαστε να λάβουμε μερικές φορές σύνθετες σχεδιαστικές αποφάσεις που αφορούν τον αριθμό, τη χαμηλή τιμή και την ακρίβεια των κόμβων, όταν διαλέγουμε ένα καταναμημένο σύστημα πολλών κόμβων. Σε αντιδιαστολή με ένα κεντροποιημένο με λιγότερους, αλλά σαφώς πιο ακριβούς και ποιοτικότερους κόμβους. Υπό συνθήκες χαμηλότερης ποιότητας και υπεράριθμοι αισθητήρες, που βρίσκονται πλησίον του φαινομένου μπορούν να οδηγήσουν σε μια αρχιτεκτονική απλούστερη και με αξιοσημείωτη εξοικονόμηση ενέργειας για το συνολικό σύστημα. Για τη καλύτερη κατανόηση της σημασίας των αισθητήρων θα αναφερθούν ορισμένοι τομείς στους οποίους βρίσκουν ή αναμένεται να βρουν εφαρμογή.

2.2. Παραδείγματα Εφαρμογών



Σχήμα 2.2: Εφαρμογές των αισθητήρων

Πέρα από την ανάγκη για την κατασκευή φτηνών, απλών στο προγραμματισμό, στη δικτύωση και μεγάλης διάρκειας αισθητήρων, πρωταρχική είναι η σημασία της αίσθησης αυτής καθέ αυτής και των ελεγκτικών ικανοτήτων του κόμβου-αισθητήρα. Για αρκετές φυσικές διεργασίες υφίσταται η αντίστοιχη τεχνολογία αισθητήρων. Μερικές δημοφιλείς είναι η θερμοκρασία, η υγρασία, η οπτική, η υπέρυθη ακτινοβολία (κάμερες), η ακουστική, η δόνηση (σεισμοί), η πίεση, οι χημικοί αισθητήρες (για αέρια διαφόρων ειδών), ο μαγνητισμός κ.α. Εκτός από τα τετριμμένα υπάρχουν και πιο εκλεπτισμένες δυνατότες για την τεχνολογία των αισθητήρων, δηλαδή τα παιχνίδια σε ένα νηπιαγωγείο μπορούν να φέρουν αισθητήρες αφής-κίνησης ή με δυνατότητα εξακρίβωσης της θέσης-ταχύτητας τους. Στο επίπεδο των κόμβων ζητούμενο αποτελούν οι δυνατότητες αίσθησης-ελέγχου σε συνδιασμό με αυτές υπολογισμού-επικοινωνίας. Ακόμα πολλές διαφορετικού είδους εφαρμογές δύναται να κατασκευαστούν από διαφορετικού τύπου κόμβους. Ακολουθούν ορισμένες ενδεικτικές εφαρμογές:

- **Εφαρμογές Αντιμετώπισης Καταστροφών:** Ένα τυπικό σενάριο είναι η προστασία από καταστρεπτικές πυρκαγιές. Οι αισθητήρες διαθέτουν θερμομέτρα και έχουν γνώση της θέσης τους στο χώρο. Οι αισθητήρες αυτοί σε μια δασική φωτιά ρίπτονται από ένα αεροπλάνο και σχηματίζουν ένα θερμικό χάρτη της περιοχής, ο οποίος θα είναι προσπελάσιμος από εξωτερικές συσκευές, όπως Personal Digital Assistants (PDAs) τοποθετημένα σε πυροσβεστικά αεροπλάνα. Αντίστοιχες καταστάσεις συναντούμε στον έλεγχο ατυχημάτων σε χημικά εργοστάσια. Αρκετές από τις παραπάνω εφαρμογές παρουσιάζουν ομοιότητες με τις στρατιωτικές εφαρμογές. Εδώ οι αισθητήρες ασχολούνται με τον εντοπισμό εχθρικών δυνάμεων αντί για το καταστροφικό σενάριο. Σε όλες αυτές τις εφαρμογές οι αισθητήρες είναι φτηνοί, διότι θεωρούνται αναλώσιμοι

και χρειάζονται αρκετοί από αυτούς. Τέλος η διάρκεια ζωής της μπαταρίας τους δεν είναι ιδιαίτερα μακρά.

- **Παρακολούθηση Περιβάλλοντος και Χαρτογράφηση Βιοποικιλότητας:** Παρακολούθηση των διαφόρων ρυπαντικών ουσιών σε χωματερές, του υποθαλάσσιου εδάφους καθώς και του αριθμού και του είδους της χλωρίδας και της πανίδας σε μια περιοχή. Τα πλεονεκτήματα των ασυρμάτων δικτύων αισθητήρων εδώ εντοπίζονται στην μεγάλης διάρκειας, ανεπιτήρητης και ασύρματης λειτουργίας των αισθητήρων. Αυτοί τοποθετούνται κοντά στην περιοχή του φαινομένου. Ακόμα το μικρό τους μέγεθος δεν ενοχλεί τα ζώα και τα φυτά. Σε αυτές τις εφαρμογές συνήθως απαιτείται μεγάλος αριθμός αισθητήρων με αξιοσημείωτη διάρκεια ζωής.
- **Ευφυή Κτίρια:** Τα κτίρια σπαταλούν σημαντικές ποσότητες ενέργειας λόγω αστοχιών στους τομείς της υγρασίας, του εξαερισμού και της χρήσης κλιματιστικών. Ένας πραγματικού χρόνου και υψηλής ακρίβειας έλεγχος της θερμοκρασίας, υγρασίας, ροής αέρα αλλά και λοιπών φυσικών παραμέτρων σε ένα κτίριο, ενδεχομένως οδηγεί σε βελτίωση της παρεχόμενης άνεσης προς τους διαβιούντες. Αλλά και στην μείωση της καταναλισκόμενης ενέργειας. Ενδεικτικά συστήματα που ασχολούνται με αυτόν τον τομέα είναι τα BACnet, LonWorks και KNX αν και δεν είναι πλήρως ασύρματα καθώς διαθέτουν αρκετές ενσύρματες μονάδες. Οι μηχανικές πιέσεις που ασκούνται σε ένα κτίριο κατά τη διάρκεια αλλά και μετά το πέρας ενός σεισμού μετρούντε από αισθητήρες. Η γνώση αυτή ίσως οδηγήσει στη διάγνωση κατάρρευσης μέρους ή ολόκληρου του κτιρίου. Πέρα από κτίρια αντίστοιχα συστήματα έχουμε και σε γέφυρες. Παραμένοντας στον τομέα το σεισμών, αισθητήρες είναι δυνατόν να χρησιμοποιηθούν στο εντοπισμό ανθρώπων, που έχουν παγιδευτεί σε χαλάσματα και να στείλουν την πληροφορία στις ομάδες διάσωσης. Ανάλογα με τη εκάστοτε εφαρμογή οι αισθητήρες που συζητώνται εδώ, δίνονται να εκσυγχρονίσουν ένα υπάρχον κτίριο ή να τοποθετηθούν κατά την ανέγερση του. Αν η παροχή ενέργειας δεν είναι σταθερή και δεδομένη τότε οι απαιτήσεις που αφορούν στη διάρκεια ζωής είναι μεγάλες – ανέρχονται σε χρόνια. Όμως ο αναγκαίος αριθμός αισθητήρων είναι μικρός και επομένως και το συνολικό κόστος τους, λαμβάνοντας υπόψην το κόστος κατασκευής ολόκληρου του κτιρίου.
- **Διαχείριση Εγκαταστάσεων:** Αναφέρομαστε σε εγκαταστάσεις κατά πολύ μεγαλύτερες από ένα κτίριο. Απλά παραδείγματα αποτελούν η είσοδος σε χώρους χωρίς τη χρήση κλειδιών, με τη χρήση σημάτων τα οποία φέρουν τα άτομα. Η εφαρμογή αυτή μπορεί να επεκταθεί και στον εντοπισμό εισβολέων έτσι ώστε να υπάρξει έγκαιρη προειδοποίηση του προσωπικού ασφαλείας. Οι εφαρμογές αυτές συναντώνται ευρέως και σε στρατιωτικούς σκοπούς και στον έλεγχο χημικών εργοστασίων για τον εντοπισμό διαρροών. Για όλα τα παραπάνω απαιτείται μεγάλος αριθμός αισθητήρων, που οφείλουν να συνεργάζονται μεταξύ τους αλλά και να λειτουργούν για μεγάλα χρονικά διαστήματα κάνοντας χρήση των μπαταριών τους.
- **Παρακολούθηση Μηχανημάτων και Προληπτική Συντήρηση:** Μια ιδέα είναι να τοποθετήσουμε κόμβους σε δύσκολα προσεγγίσιμες περιοχές των μηχανημάτων, όπου ο εντοπισμός τριγμών δηλώνει την ανάγκη συντήρησης. Τα βασικά πλεονεκτήματα των ασύρματων δικτύων αισθητήρων είναι η έλλειψη καλωδίων και η αποφυγή του προβλήματος της συντήρησης. Ενσύρματη παροχή ενέργειας μπορεί να είναι διαθέσιμη ανάλογα με το σενάριο. Αμα δεν υπάρχει τότε οι μπαταρίες πρέπει να έχουν ικανοποιητική διάρκεια, αφού η αλλαγή τους δεν είναι πρακτική και κοστίζει. Ακόμα το μέγεθος των κόμβων και το κόστος τους, δεν είναι αποφασιστικής σημασίας.

- **Αγροτικές Καλλιέργειες:** Εδώ η χρήση αισθητήρων οδηγεί σε ορθότερη άδρευση και λίπανση του χώρου καλλιέργειας. Ένας σχετικά μικρός αριθμός αισθητήρων της τάξεως του ενός ανά 100 μ επί 100 μ λογίζεται λειτουργικός. Επίσης ενδεχομένως υπάρξει δυνατότητα εντοπισμού και ελέγχου των ζιζανίων. Δεν πρέπει να ξεχνούμε ότι και η κτηνοτροφία δέχεται της ευργετικές επιδράσεις της τοποθέτησης αισθητήρων. Για παράδειγμα μετρούμε το επίπεδο υγείας των εκτρεφόμενων ζώων μέσω της θερμοκρασίας του σώματος τους και του αριθμού των βημάτων τους.
 - **Ιατρική και Πρόληψη:** Εδώ οι αισθητήρες εμφανίζονται να έχουν διττό ρόλο μίας και πέρα από την ευεργετική τους δράση εγείρουν αμφιλεγόμενα ηθικά ζητήματα. Τα ενδεχόμενα σενάρια διατρέχουν μια ευρεία γκάμα από μετεγχειρητικές καταστάσεις όπου οι αισθητήρες ενσωματώνονται στον ασθενή, μέχρι παρακολούθηση εξ αποστάσεως από το γιατρό συνήθως ηλικιωμένων και την αυτόματη χωρήγηση φαρμάκων. Αλλά και στα νοσοκομεία θα υπάρξουν αμφίδρομα συστήματα εντοπισμού ιατρού-ασθενούς.
 - **Διοικητική Μέριμνα (Logistics):** Πολλές φορές χρειάζεται να εξοπλίσουμε αγαθά με απλούς αισθητήρες. Αυτοί θα μας επιτρέπουν να εντοπίζουμε τη θέση τους κατά τη διάρκεια της μετακίνησης τους και όταν τοποθετούνται σε αποθήκες ή στα ράφια μαγαζιών. Οι εφαρμογές αυτές δεν χρησιμοποιούν πάντα ενεργητική επικοινωνία, αλλά παθητική παρουσίαση των ψηφιακών δεδομένων, παράδειγμα μια βαλίτσα που μετακινείται σε ένα αεροδρόμιο και περνά από συγκεκριμένες περιοχές ελέγχου. Αυτού του είδους η επικοινωνία είναι σαφώς απλούστερη και φτηνότερη από αυτή που συναντήσαμε σε προηγούμενες εφαρμογές. Αυτή η ταυτοποίηση μέσω ραδιοσυχνοτήτων όμως δεν έχει απεριόριστες δυνατότητες, όταν φτάνουμε σε πιο αναπτυγμένες εφαρμογές. Ένα ακόμα μείον αποτελεί η μη δυνατότητα καταγραφής πληροφοριών του ιστορικού του αγαθού στο οποίο έχουμε προσκολήσει τον αισθητήρα, εδώ πλέον πρέπει ο αισθητήρας να έχει ενεργώ συμμετοχή.
- 1) **Τηλεμετρία:** Οι αισθητήρες τοποθετούνται στους δρόμους ή πέριξ αυτών και δέχονται πληροφορίες υψηλής ανάλυσης. Τα συστήματα αισθητήρων αλληλεπιδρούν με τα αυτοκίνητα μεταφέροντας προειδοποιήσεις για επικίνδυνες καταστάσεις στο δρόμο αλλά και κυκλοφοριακά προβλήματα.

2.3. Τύποι Εφαρμογών

Αρκετές από αυτές τις εφαρμογές μοιράζονται ορισμένα κοινά χαρακτηριστικά. Σε πολλές υπάρχει σαφής διαχωρισμός των πηγών των δεδομένων από τους κόμβους που αισθάνονται τα δεδομένα και των κόμβων που καταλήγουν (καταβόθρες δεδομένων). Οι καταβόθρες άλλοτε αποτελούν μέρος του δικτύου και άλλες φορές βρίσκονται εκτός αυτού. Ακόμα υπάρχουν περισσότερες πηγές από καταβόθρες. Οι διαδικασίες αλληλεπίδρασης μεταξύ πηγών και καταβοθρών εμφανίζουν μια τυποποίηση

- **Ανίχνευση Γεγονότων:** Οι αισθητήρες πρέπει να ενημερώνουν τις καταβόθρες μόλις εντοπίζουν την ύπαρξη ενός συγκεκριμένου συμβάντος. Τα απλούστερα συμβάντα εντοπίζονται τοπικά από ένα απλό αισθητήρα. Αντίθετα πιο σύνθετα γεγονότα απαιτούν την συνεργασία αρκετών κόμβων για να αποφασιστεί αν το συμβάν έχει λάβει χώρα. Σε περιπτώσει που λαμβάνουν χώρα διάφορα συμβάντα τότε θέλουμε και ταξινόμηση γεγονότων.

- **Περιοδικές Μετρήσεις:** Οι αισθητήρες μπορούν να προγραμματιστούν για να αναφέρουν περιοδικά τις μετρήσιμες τιμές. Συχνά αυτές οι αναφορές πυροδοτούνται από κάποιο γεγονός. Η περίοδος των αναφορών εξαρτάτε σαφώς από το είδος της εφαρμογής.
- **Συναρτησιακή Προσέγγιση:** Μια φυσική μεταβλητή όπως η θερμοκρασία λαμβάνει διάφορες τιμές, αυτή η διαδικασία μπορεί να αναπαρασταθεί από μια συνάρτηση. Ενώ ασύρματο δίκτυο αισθητήρων είναι ικανό να προσομοιώσει αυτή την άγνωστη συνάρτηση, με τη χρήση δειγματοληψίας από τους κόμβους-αισθητήρες. Η χαρτογράφηση που προκύπτει, μπορεί να διατεθεί στις καταβόθρες. Το πότε και το πως θα ανανεώνεται ο χάρτης εξαρτάτε από τις ανάγκες της εφαρμογής, όπως αυτές προσδιορίζονται από την ακρίβεια της αναπαράστασης και την κατανάλωση ενέργειας.
- **Εντοπισμός:** Η πηγή ενός γεγονότος μπορεί να είναι κινητή (εισβολέας σε σενάρια παρακολούθησης). Το δίκτυο αισθητήρων αναφέρει τις καινούριες πληροφορίες από την πηγή καθώς αυτές κινούνται, κατευθύνοντας τις στην καταβόθρα. Αρκετές φορές η πληροφορία αυτή περιέχει την ταχύτητα και την κατεύθυνση. Η περιγραφόμενη διαδικασία συνήθως προϋποθέτει πρότερη συνεργασία μεταξύ των αισθητήρων πριν την αναφορά στην καταβόθρα. Αυτή η συνεργασία γίνεται τόσο στο επίπεδο του χρόνου όσο και στο χώρο με δυναμικά χαρακτηριστικά καθώς περνάει ο χρόνος.

Τα παραδείγματα καταδεικνύουν ότι υπάρχει ποικιλομορφία στις επιλογές τοποθέτησης. Τα όρια των επιλογών αυτών εκτείνονται από καλοσχεδιασμένες και σταθερές τοποθετήσεις, μέχρι τυχαίες όπως αυτές που προκύπτουν μετά τη ρίψη από αεροπλάνο. Ακόμα οι κόμβοι ίσως έχουν τη δυνατότητα μετακίνησης κατά τη διάρκεια της τοποθέτησης τους, έτσι ώστε να λάβουν καλύτερες θέσεις. Ακόμα μπορεί να χρειαστεί μετακίνηση, επειδή έχουν προσκολληθεί πάνω σε κάποιο αντικείμενο μελέτης (π.χ. κάποιο ζώο).

Οι εφαρμογές επηρεάζουν και τις επιλογές στο θέμα της συντήρησης. Για παράδειγμα μερικά σενάρια προϋποθέτουν μακρά και απρόσκοπτη λειτουργία χωρίς τη δυνατότητα συντήρησης. Σε άλλες περιπτώσεις η δυνατότητα συντήρησης δεν έχει καμία σημασία διότι είναι αναλώσιμοι, οι αισθητήρες. Άμεση είναι η εξάρτηση της συντήρησης με επιλογές στον τομέα της παροχής ενέργειας. Εφαρμογές με τη δυνατότητα παροχής ενέργειας από καλώδιο δεν εμφανίζουν προβληματισμούς σε αυτόν τον τομέα. Όμως για τις υπόλοιπες περιπτώσεις η διάρκεια ζωής εξαρτάται από την αποστολή και την παροχή ενέργειας. Εδώ συναντούμε από τετριμμένες έως άκρως απαιτητικές καταστάσεις που απαιτούν ρηζικέλευθες λύσεις. Προφανώς το κόστος κατασκευής και το μέγεθος του κόμβου διαδραματίζουν το ρόλο τους στη σχεδίαση της παροχής ενέργειας.

2.4. Προκλήσεις για τα Ασύρματα Δίκτυα Αισθητήρων

Η εμφάνιση συγκεκριμένων χαρακτηριστικών γνωρισμάτων μας οδηγεί στη βαθύτερη κατανόηση των ασυρμάτων δικτύων αισθητήρων. Ιδιαίτερα, όπως αυτά διαφαίνονται μέσω των χαρακτηριστικών και των απαιτούμενων μηχανισμών.

2.4.1. Απαιτούμενα Χαρακτηριστικά

- **Είδος υπηρεσιών:** Το είδος υπηρεσιών που παρέχεται από ένα συμβατικό δίκτυο επικοινωνίας είναι εμφανής - μετακινεί τα bits από μια θέση προς μια άλλη. Για ένα WSN, η μεταφορά των bits δεν αποτελεί τον πραγματικό σκοπό, γιατί ένα WSN αναμένεται να διακινεί bits από τα οποία προκύπτει ξεκάθαρη πληροφορία. Τα WSN πρέπει να προβαίνουν σε ενέργειες προς έναν δεδομένο στόχο: «Οι άνθρωποι θέλουν απαντήσεις, όχι αριθμούς» (Steven Glaser, Μπέρκλεϊ). Επιπλέον, πεδία που αφορούν αλληλεπιδράσεις σε συγκεκριμένες γεωγραφικές περιοχές ή σε χρονικά διαστήματα αποκτούν διαφορετική σημασία. Ως εκ τούτου, από τη χρησιμοποίηση τέτοιων δικτύων προκύπτουν νέα παραδείγματα, καινούριες διεπαφές και τέλος νέοι τρόποι προσέγγισης των υπηρεσιών.
- **Η ποιότητα παρεχόμενων υπηρεσιών:** Στενά συνδεδεμένη με το είδος της υπηρεσίας ενός δικτύου είναι και η ποιότητα της. Οι παραδοσιακές απαιτήσεις ποιότητας εξυπηρέτησης συνήθως προέρχονται από τις εφαρμογές πολυμέσων, όπως η φραγμένη καθυστέρηση ή το ελάχιστο εύρος ζώνης. Όμως είναι άσχετες σε εφαρμογές που εξαρχής χαρακτηρίζονται ως ανεκτικές στη λανθάνουσα κατάσταση (latency) ή που το εύρος ζώνης των μεταδιδόμενων δεδομένων τους είναι πολύ μικρό. Σε μερικές περιπτώσεις, μόνο η περιστασιακή παράδοση ενός πακέτου μπορεί να είναι ικανοποιητική, σε άλλες πάλι υπάρχουν πολύ υψηλές απαιτήσεις αξιοπιστίας. Ο ρυθμός παράδοσης πακέτων είναι μια ανεπαρκής μετρική. Αντίθετα μείζονα είναι το ποσό και η ποιότητα των πληροφοριών, που μπορούν να οδηγηθούν στις καταβόθρες των περιοχών ενδιαφέροντος. Ποιοτικές έννοιες οι οποίες έχουν μεγάλη σημασία, είναι η αξιόπιστη ανίχνευση των γεγονότων και η ποιότητα προσέγγισης τους (π.χ. χάρτης θερμοκρασίας).
- **Η ανοχή σφαλμάτων δεδομένου:** Οι κόμβοι μπορούν να ξεμείνουν από ενέργεια ή να καταστραφούν. Ακόμα δεδομένου ότι η ασύρματη επικοινωνία μεταξύ δύο κόμβων μπορεί να διακοπεί μόνιμα έχει σημασία το WSN να είναι σε θέση να ανταπεξέλθει σε τέτοιες αστοχίες. Για να αντιμετωπιστούν τα σφάλματα των κόμβων, η πλεονάζουσα χρήση τους κρίνεται απαραίτητη. Δηλαδή χρησιμοποιούμε περισσότερους κόμβους από αυτούς που είναι αυστηρά απαραίτητοι, δεδομένου ότι όλοι λειτουργούν κανονικά.
- **Η διάρκεια ζωής:** Σε πολλά σενάρια, κόμβοι θα πρέπει να λειτουργήσουν με μια περιορισμένη παροχή ενέργειας (χρήση μπαταρίας). Η αντικατάσταση αυτών των πηγών ενέργειας δεν είναι συνήθως εφικτή. Ταυτόχρονα ένα WSN πρέπει να λειτουργήσει τουλάχιστον για το δεδομένο χρόνο αποστολής του ή όσο το δυνατόν περισσότερο. Ως εκ τούτου, η διάρκεια ζωής ενός WSN αποτελεί μια πολύ σημαντική μετρική. Προφανώς, έχουμε απαίτηση για την εξέβρεση ενεργειακά συμφέροντος τρόπου λειτουργίας του WSN. Σαν εναλλακτική λύση προβάλλει η διαθεσιμότητα μιας περιορισμένης πηγής ισχύος (όπως τα ηλιακά κύτταρα) σε έναν κόμβο αισθητήρων. Χαρακτηριστικά, αυτές οι πηγές δεν είναι αρκετά ισχυρές ώστε να εξασφαλίσουν συνεχή λειτουργία αλλά μπορούν να παρέχουν επαναφόρτιση των μπαταριών. Υπό τέτοιους όρους, η διάρκεια ζωής του δικτύου πρέπει ιδανικά να είναι άπειρη. Η διάρκεια ζωής ενός δικτύου έχει άμεση συσχέτιση με την ποιότητα εξυπηρέτησης: μια ενδεχόμενη ενεργοβόρα επιλογή μπορεί να αυξήσει την ποιότητα, αλλά μειώνει τη διάρκεια ζωής. Ο ακριβής καθορισμός της διάρκειας ζωής εξαρτάται από την εφαρμογή. Μια απλή επιλογή για τη διάρκεια ζωής των δικτύων είναι να χρησιμοποιηθεί ο χρόνος έως ότου αποτύχει ο πρώτος κόμβος (ή ξεμείνει από ενέργεια). Άλλες επιλογές

περιλαμβάνουν το χρόνο έως ότου αποσυνδέεται το δίκτυο σε δύο ή περισσότερα κομμάτια, ο χρόνος έως ότου 50% (ή κάποια άλλη σταθερή αναλογία) των κόμβων έχει αποτύχει και τέλος ο χρόνος που για πρώτη φορά ένα σημείο στην περιοχή ενδιαφέροντος δεν καλύπτεται από τουλάχιστον έναν κόμβο αισθητήρα.

- **Η Κλιμακωση:** Ένα WSN ενδεχομένως περιλαμβάνει έναν μεγάλο αριθμό κόμβων, οι αρχιτεκτονικές και τα πρωτόκολλα πρέπει να είναι σε θέση να κλιμακώνουν σε αυτούς τους αριθμούς.
- **Ευρύ φάσμα πυκνότητας:** Σε ένα WSN ο αριθμός κόμβων ανά περιοχή ή αλλιώς η πυκνότητα του δικτύου μπορεί να ποικίλει αρκετά. Διαφορετικές εφαρμογές έχουν πολύ διαφορετικές πυκνότητες κόμβων. Ακόμη και μέσα σε μια δεδομένη εφαρμογή, η πυκνότητα μπορεί να ποικίλει με την πάροδο του χρόνου και να αραιώσει επειδή οι κόμβοι αποτυγχάνουν ή μετακινούνται. Η πυκνότητα δεν παρουσιάζει ομοιογένεια σε ολόκληρο δίκτυο (λόγω αστοχίας στην τοποθέτηση) οπότε το δίκτυο πρέπει να μπορεί να προσαρμόζεται σε τέτοιες περιστάσεις.
- **Ικανότητα Προγραμματισμού:** Θεωρείτε απαραίτητη ώστε οι κόμβοι να έχουν τη δυνατότητα να επεξεργαστούν τις πληροφορίες, αλλά και να έχουν την ευελιξία της αντίδρασης σε τυχόν αλλαγές στις διεργασίες τους. Οι κόμβοι πρέπει να είναι προγραμματίσιμοι και ο προγραμματισμός οφείλει να μεταβάλλεται κατά τη λειτουργία του δικτύου, όταν σημειώνονται αλλαγές. Με λίγα λόγια, εάν η επεξεργασία των πληροφοριών δεν παρουσιάζει δυναμικά στοιχεία τότε είναι ανεπαρκής.
- **Η συντηρησιμότητα:** Τόσο το περιβάλλον ενός WSN όσο και στο ίδιο το WSN αλλάζουν (μειωμένες μπαταρίες, αποτυχία στους κόμβους, νέες διεργασίες). Σε αυτές τις περιπτώσεις το σύστημα πρέπει να προσαρμοστεί. Είναι ανάγκη να ελέγχει την κατάσταση και τη θέση του αλλάζοντας τις λειτουργικές παραμέτρους ή προβαίνοντας σε διαφορετικές επιλογές (π.χ. παροχή χαμηλότερης ποιότητας όταν η ενέργεια γίνεται λιγότερη). Εν ολίγοις, το δίκτυο πρέπει να αυτοδιατηρηθεί. Ακόμα θα μπορούσε να είναι σε θέση να αλληλεπιδράσει με εξωτερικούς μηχανισμούς συντήρησης. Έτσι εξασφαλίζει επέκταση στη λειτουργία του ικανοποιώντας ταυτόχρονα τις ποιοτικές απαιτήσεις.

2.4.2. Απαιτούμενοι Μηχανισμοί

Για να αντιμετωπίσουμε αυτές τις απαιτήσεις, χρειάζεται η ανεύρεση καινοτόμων μηχανισμών για τα δίκτυα επικοινωνίας, νέες αρχιτεκτονικές και καινούρια πρωτόκολλα. Ιδιαίτερη πρόκληση αποτελεί η ανεύρεση μηχανισμών αρκετά προσανατολισμένων στην ιδιοσυγκρασία μιας δεδομένης εφαρμογής. Για να υποστηριχθεί η συγκεκριμένη ποιότητα παρεχόμενων υπηρεσιών, η διάρκεια ζωής και οι απαιτήσεις στο τομέα της συντήρησης. Αφ' ετέρου, αυτοί οι μηχανισμοί πρέπει να παρέχουν δυνατότητες γενίκευσης σε ένα ευρύτερο φάσμα εφαρμογών. Μερικοί από τους μηχανισμούς που συναντούμε στα WSNs είναι:

- **Η ασύρματη επικοινωνία Multihop:** Ενώ η ασύρματη επικοινωνία αποτελεί σημαντικότερη τεχνική, η άμεση επικοινωνία αποστολέα - δέκτη εμφανίζει περιορισμούς. Η επικοινωνία σε μεγάλες αποστάσεις καθίσταται δυνατή μόνο με τη χρήση, απαγορευτικά υψηλής δύναμης μετάδοσης. Όμως η χρήση ενδιάμεσων κόμβων

μπορεί να μειώσει τη συνολική απαιτούμενη δύναμη. Ως εκ τούτου, σε διάφορες κατηγορίες WSNs, η επικοινωνία τύπου multihop αποτελεί απαραίτητο συστατικό.

- **Η Αποτελεσματική Ενεργειακή Λειτουργία:** Για να υπάρξει ικανοποιητική διάρκεια ζωής, η αποτελεσματική ενεργειακή λειτουργία θεωρείται βασική τεχνική. Οι επιλογές που εξετάζονται περιλαμβάνουν τη αποτελεσματική μεταφορά bits μεταξύ δύο κόμβων (που μετριοούνται σε J/bit) ή τον ενεργειακά αποτελεσματικό προσδιορισμό των ζητούμενων πληροφοριών. Επίσης αποτελεί ζήτημα η ανομοιογενής ενεργειακή κατανάλωση.
- **Η αυτόδιαμόρφωση:** Ένα WSN οφείλει να διαμορφώνει αυτόνομα τις λειτουργικές παραμέτρους του, χωρίς να χρειάζεται εξωτερική διαμόρφωση. Για παράδειγμα, οι κόμβοι πρέπει να είναι σε θέση να καθορίσουν τις γεωγραφικές συντεταγμένες τους χρησιμοποιώντας μόνο άλλους κόμβους του δικτύου. Επίσης κρίνεται σημαντικό για το δίκτυο η ανοχή στους αποτυχόντες κόμβους (π.χ. λόγω μειωμένης μπαταρίας) καθώς και η ενσωμάτωση νέων (λόγω της αυξητικής επέκτασης του δικτύου μετά από την αποτυχία).
- **Συνεργασία και ενδοδικτυακή επεξεργασία:** Σε μερικές εφαρμογές, ένας αισθητήρας δεν είναι ικανός να αποφασίσει εάν ένα γεγονός έχει συμβεί, αντίθετα διάφοροι αισθητήρες πρέπει να συνεργαστούν ώστε να το ανιχνεύσουν. Τότε, μόνο το σύνολο των ενοποιημένων δεδομένων, των πολλών αισθητήρων παρέχει αρκετές πληροφορίες. Οι πληροφορίες επεξεργάζονται από το ίδιο το δίκτυο με διάφορες μορφές. Για να επιτευχθεί αυτή η συνεργασία, διαβιβάζονται όλα τα στοιχεία σε ένα εξωτερικό δίκτυο που τις επεξεργάζεται «στις παρυφές» του αρχικού δικτύου. Ένα παράδειγμα είναι ο καθορισμός της υψηλότερης ή μέσης θερμοκρασίας μέσα σε μια περιοχή και η προώθηση της τιμής της σε μια καταβόθρα. Για να λυθούν τέτοιες καταστάσεις αποτελεσματικά, αναγνώσεις από μεμονωμένους αισθητήρες μπορούν να επεξεργάζονται καθώς διαδίδονται στο δίκτυο. Πράγμα που οδηγεί στη μείωση του πλήθους των δεδομένων τα οποία διαβιβάζονται και έτσι βελτιώνεται η ενεργειακή αποδοτικότητα. Το πώς θα επιτευχθεί αυτή η επεξεργασία αποτελεί ανοιχτό πεδίο έρευνας.
- **Δεδομένο-κεντρικότητα:** Τα παραδοσιακά δίκτυα επικοινωνίας είναι προσανατολισμένα στη μεταφορά δεδομένων μεταξύ δύο συγκεκριμένων συσκευών, που κάθε μια διαθέτει (τουλάχιστον) μια διεύθυνση δικτύου. Η λειτουργία τέτοιων δικτύων είναι διεύθυνσιο-κεντρική. Σε ένα WSN, οι κόμβοι είναι τοποθετημένοι σε πλεόνασμα για να προστατευτεί από τις αποτυχίες των κόμβων ή για να αντισταθμιστεί η χαμηλή ποιότητα των αισθανομένων συμβάντων ενός ατομικού κόμβου. Η ταυτότητα του συγκεκριμένου κόμβου που παρέχει τα στοιχεία δεν έχει σχέση. Αυτό που είναι σημαντικό είναι οι απαντήσεις και οι τιμές και όχι ο κόμβος που τις παρέχει. Μεταπηδούμε από ένα διεύθυνσιο-κεντρικό παράδειγμα σε ένα δεδομένο-κεντρικό στο σχεδιασμό των πρωτοκόλλων αρχιτεκτονικής και επικοινωνίας. Ένα παράδειγμα για μια τέτοια δεδομένο-κεντρική αλληλεπίδραση θα ήταν να ζητηθεί η μέση θερμοκρασία σε μια δεδομένη περιοχή, σε αντιδιαστολή με την ανάγνωση θερμοκρασιών από μεμονωμένους κόμβους. Ένα τέτοιο δεδομένο-κεντρικό παράδειγμα μπορεί επίσης να χρησιμοποιηθεί σε περιπτώσεις καθορισμένων κατωφλιών για τις ειδοποιήσεις ή τα γεγονότα («σήμανε συναγερμό εάν η θερμοκρασία υπερβαίνει ένα κατώτατο όριο»). Από αυτή την άποψη, η δεδομένο-κεντρική προσέγγιση εμφανίζει σχετικότητα με τις ερωτήσεις (queries) όπως αυτές είναι γνωστές από τις βάσεις δεδομένων.

- **Η Γειτονικότητα:** Περισσότερο μια σχεδιαστική οδηγία παρά ένας κατάλληλος μηχανισμός, υποδηλώνει ότι η αρχή της γειτονικότητας θα πρέπει να τηρηθεί εκτενώς ώστε να εξασφαλιστεί η δυνατότητα κλιμάκωσης. Οι κόμβοι με περιορισμένους πόρους όπως η μνήμη, προσπαθούν κατά την επεξεργασία του πρωτοκόλλου, να περιοριστούν μόνο στις πληροφορίες για τους άμεσους γείτονές τους. Αυτό ενδεχομένως επιτρέπει την κλιμάκωση του δικτύου για μεγάλους αριθμούς κόμβων χωρίς να στηριχθούμε σε ισχυρή επεξεργασία σε κάθε ατομικό κόμβο. Το πώς συνδυάζουμε την αρχή γειτονικότητας με αποδοτικά πρωτόκολλα παραμένει ανοικτό.
- **Εκμετάλλευση Ανταλλαγμάτων:** Κατά τη διάρκεια του σχεδιασμού του πρωτοκόλλου αλλά και του χρόνου εκτέλεσης χρειάζεται να επιτύχουμε στόχους, οι οποίοι πολλές φορές είναι αλληλο-αντικρουόμενοι και τότε ξεκινά το παιχνίδι των ανταλλαγμάτων. Επί παραδείγματι οι υψηλότερες ενεργειακές δαπάνες επιτρέπουν την υψηλότερη ακρίβεια αποτελέσματος αλλά και τη χαμηλότερη διάρκεια ζωής. Γενικά τα πρωτόκολλα θα πρέπει να χειριστούν πολύ διαφορετικές καταστάσεις, ενδεχομένως σε διάφορα μέρη ενός ενιαίου δικτύου.

Η εκμετάλλευση αυτών των μηχανισμών έτσι ώστε είναι εύχρηστοι και αρκετά γενικοί, για έναν προγραμματιστή εφαρμογής αποτελούν μια σημαντική πρόκληση.

3. Directed Diffusion (κατευθυνόμενη διάχυση)

3.1. Εισαγωγή

Στο προσεχές μέλλον η ανάπτυξη της τεχνολογίας στον τομέα της επεξεργασίας, των μνημών και των ραδιοσυχνοτήτων θα επιτρέπουν σε μικρούς και φτηνούς κόμβους αξιοσημείωτες δυνατότητες. Τόσο στο τομέα της αίσθησης, όσο και σε αυτούς της επικοινωνίας και του υπολογισμού. Πέρα από τη αισθητήρια ατομική δυνατότητα των συσκευών αυτών, παρέχεται και η δυνατότητα συνεργασίας μεταξύ των συσκευών αυτών. Αυτό οδηγεί στην επίτευξη αποστολών με συνθετότερες απαιτήσεις στον τομέα της αίσθησης. Τέτοιες τεχνολογίες δύναται να αλλάξουν ριζικά, τις διαδικασίες περισυλλογής και επεξεργασίας σε αρκετές περιπτώσεις. Μεγάλης κλίμακας, ταχύτατα μεταβαλλόμενα και εύρωστα δίκτυα αισθητήρων μπορούν να αναπτυχθούν σε αφιλόξενα φυσικά περιβάλλοντα (π.χ. τοξικές αστικές περιοχές). Ακόμα θα επιτρέψουν χαμηλό κόστος συντήρησης σε όχι τόσο επικίνδυνες περιοχές, που όμως είναι λιγότερο προσιτές. Τέτοια περιβάλλοντα θεωρούνται οι μεγάλες βιομηχανικές εγκαταστάσεις και το εσωτερικό των αεροσκαφών. Για να αναδειχθεί η σημασία της έρευνας σε αυτούς τους τομείς, ας θεωρήσουμε το παρακάτω απλουστευμένο πρότυπο λειτουργίας για ένα δίκτυο αισθητήρων. Ένας ή περισσότεροι άνθρωποι χειριστές θέτουν, σε οποιοδήποτε κόμβο στο δίκτυο, ερωτήματα της μορφής:

«Πόσοι πεζοί παρατηρούνται στη γεωγραφική περιοχή X;» ή «Σε ποια κατεύθυνση κινείται το αυτοκίνητο στην περιοχή Y;»

Αυτές οι ερωτήσεις καταλήγουν στους αισθητήρες, που βρίσκονται μέσα στη περιοχή ενδιαφέροντος και επιφορτίζουν τους κόμβους με τη συλλογή πληροφοριών. Μόλις ανιχνεύσουν οι μεμονωμένοι κόμβοι τους πεζούς ή τις μετακινήσεις των οχημάτων, ενδεχομένως συνεργαστούν με γειτονικούς κόμβους. Η συνεργασία γίνεται στα πλαίσια της προσπάθειας εξεύρεσης της θέσης των πεζών ή της κατεύθυνσης μετακίνησης των οχημάτων. Ένας από αυτούς τους κόμβους στη συνέχεια δημοσιοποιήσει το αποτέλεσμα πίσω στον ανθρώπινο χειριστή. Παρακινούμενοι από την ευρωστία, τη δυνατότητα κλιμάκωσης και τις απαιτήσεις σε ενεργειακή αποδοτικότητα, εξετάζουμε ένα νέο πρότυπο διάδοσης δεδομένων για τέτοια δίκτυα αισθητήρων. Αυτό το πρότυπο, που καλείτε κατευθυνόμενη διάχυση (Directed Diffusion) χαρακτηρίζεται ως δεδομένο-κεντρικό. Τα δεδομένα που παράγονται από τους κόμβους των αισθητήρων αποτελούνται από ζευγάρια ιδιοτήτων - τιμών. Ένας κόμβος αιτάται δεδομένα με την αποστολή ενδιαφέροντος για τα προσδιοριζόμενα δεδομένα. Τα δεδομένα που ταιριάζουν με το ενδιαφέρον κατευθύνονται προς τον αιτούντα κόμβο. Οι ενδιάμεσοι κόμβοι μπορούν να αποθηκεύσουν, να μετασχηματίσουν δεδομένα καθώς και να κατευθύνουν τα ενδιαφέροντα βασισμένοι στα παλαιότερα αποθηκευμένα δεδομένα.

Χρησιμοποιώντας αυτό το πρότυπο επικοινωνίας, το προαναφερθέν παράδειγμα μπορεί να έχει ως εξής. Η ερώτηση (εκδήλωση ενδιαφέροντος για κάτι) του ανθρώπινου χειριστή θα μετασχηματιστεί σε ενδιαφέρον το οποίο διαχέεται (π.χ., ραδιομετάδοση, γεωγραφική καθοδήγηση) προς τους κόμβους στις περιοχές X ή Y. Όταν ένας κόμβος στην ζητούμενη περιοχή λαμβάνει ένα ενδιαφέρον, ενεργοποιεί τους αισθητήρες του, που αρχίζουν να μαζεύουν πληροφορίες για τους πεζούς. Όταν οι αισθητήρες αντιληφθούν την παρουσία των πεζών, τότε οι πληροφορίες επιστρέφουν ακολουθώντας αντίστροφη πορεία από εκείνη της διάδοσης του ενδιαφέροντος. Οι ενδιάμεσοι κόμβοι

επεξεργάζοντε τα δεδομένα. Κλασσικό παράδειγμα επεξεργασίας αποτελεί ο ακριβέστερος προσδιορισμός της θέσης του πεζού, μέσο του συνδυασμού των εκθέσεων-πληροφοριών από τους διάφορους αισθητήρες. Εξέχουσα σημασία χαρακτηριστικό γνώρισμα του Directed Diffusion αποτελεί η τοπικότητα των αλληλεπιδράσεων, στους τομείς της διάδοσης και της επεξεργασίας, τόσο των ενδιαφερόντων όσο και των δεδομένων. Με λίγα λόγια έχουμε ανταλλαγές μηνυμάτων μεταξύ γειτόνων ή κόμβων που παρουσιάζουν κάποια εγγύτητα.

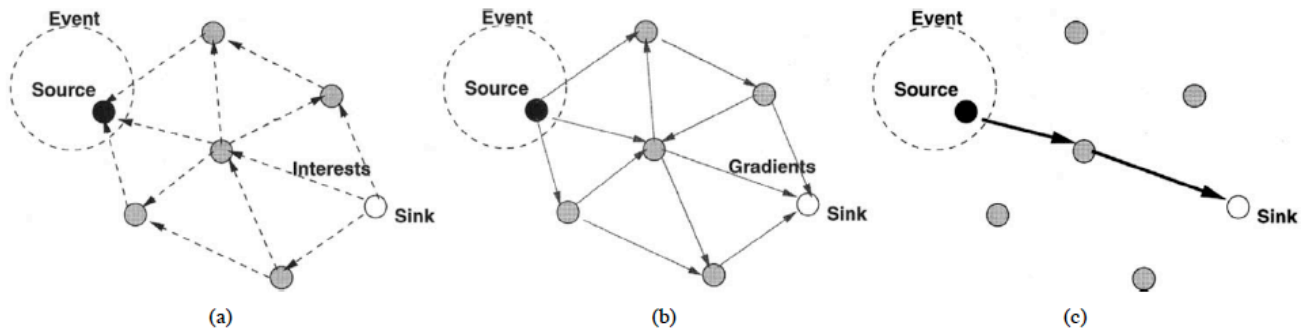
Η κατευθυνόμενη διάχυση παρουσιάζει σημαντικές διαφορές από την επικοινωνία τύπου IP . Στην τελευταία οι κόμβοι προσδιορίζονται από τα σημεία κατάληξης τους. Η ενδοκομβική επικοινωνία βασίζεται σε μία από άκρη σε άκρη υπηρεσία μεταφοράς, που παρέχει το δίκτυο . Σε αυτό το σημείο θα περιγράψει η κατευθυνόμενη διάχυση. Ακόμα θα επεξηγηθεί ένα παράδειγμα αυτού του προτύπου, που αφορά στη διάδοση και στην επεξεργασία των ερωτήσεων που τίθενται στους αισθητήρες. Καθίσταται εμφανές ότι με τη υιοθέτηση της κατευθυνόμενης διάχυσης, παρέχονται δυνατότητες: πολλαπλών μονοπατιών κατά την διαδικασία της παράδοσης, εμπειρικής προσαρμογής σε ένα μικρό υποσύνολο μονοπατιών στο δίκτυο και επιτυγχάνεται σημαντική εξοικονόμηση ενέργειας, όταν οι ενδιάμεσοι κόμβοι προβαίνουν σε επεξεργασία των απαντήσεων. Ακόμα θα παρουσιαστεί στη συνέχεια η εφαρμογή της κατευθυνόμενης διάχυσης σε διάφορες μικρές πλατφόρμες αισθητήρων.

Επίσης περιγράφεται το πρότυπο της κατευθυνόμενης διάχυσης και εξηγούνται τα κύρια χαρακτηριστικά του. Ακολουθεί περιγραφή κάποιων λεπτομερειών ενός ιδιαίτερου παραδείγματος, εντοπισμού οχημάτων από δίκτυο αισθητήρων. Διευκρινίζεται το τι επιτυγχάνουν οι τοπικοί κανόνες όσο αφορά την επιθυμητή συμπεριφορά κατά τη διάδοση του ενδιαφέροντος και των δεδομένων. Με αυτό τον τρόπο, αναδεικνύονται οι διαφορές του προτύπου της κατευθυνόμενης διάχυσης σε σχέση με την παραδοσιακή δικτύωση. Από ποιοτικής πλευράς υποστηρίζεται ότι αυτό το πρότυπο οδηγεί σε οφέλη στην κλιμάκωση, στην ευρωστία και την ενεργειακή αποδοτικότητα. Τα οφέλη αυτά στη συνέχεια ποσοτικοποιούνται μέσω λεπτομεριακής εξομοίωσης σε επίπεδο πακέτου.

3.2. Χαρακτηριστικά στοιχεία του Directed Diffusion

Η κατευθυνόμενη διάχυση αποτελείται από διάφορα στοιχεία: μηνύματα εκδήλωσης ενδιαφέροντος, μηνύματα δεδομένων, gradients (είναι η πίσω σύνδεση) και διαδικασίες ενίσχυσης. Ένα μήνυμα ενδιαφέροντος είναι μια ερώτηση ή μια ανακριτική διαδικασία που διευκρινίζει τι θέλει ο χρήστης. Κάθε μήνυμα ενδιαφέροντος περιέχει την περιγραφή μιας αισθητήριας διαδικασίας, η οποία υποστηρίζεται από το δίκτυο αισθητήρων και οδηγεί στην απόκτηση των δεδομένων. Τυπικά, ως δεδομένα στα δίκτυα αισθητήρων χαρακτηρίζονται οι συλλεχθείσες ή επεξεργασμένες πληροφορίες που προκύπτουν από ένα φυσικό φαινόμενο. Τέτοια δεδομένα μπορούν να αποτελέσουν τα γεγονότα (events) τα οποία στην ουσία είναι μια σύντομη περιγραφή του αισθανόμενου φαινομένου. Στην κατευθυνόμενη διάχυση, τα δεδομένα χαρακτηρίζονται από ζευγάρια ιδιότητας-τιμής. Αισθητήρια αποστολή (task) αποτελεί και η διάδοση σε όλο το δίκτυο των ενδιαφερόντων με τη μορφή των ονοματοποιημένων δεδομένων. Αυτή η διάδοση θέτει gradients μέσα στο δίκτυο με σκοπό να «κατευθύνουν» τα γεγονότα (δηλ. Δεδομένα που ταιριάζουν με τα ενδιαφέροντα). Συγκεκριμένα, ένα gradient δημιουργείτε προς την κατεύθυνση του κόμβου από όπου και έλαβε το μήνυμα ενδιαφέροντος. Για την ακρίβεια από τον γειτονικό κόμβο από τον οποίο το έλαβε. Τα συμβάντα ταξιδεύουν προς τους κόμβους που δημιούργησαν τα ενδιαφέροντα και κατά μήκος των πολλαπλών

gradient μονοπατιών. Το δίκτυο αισθητήρων θα ενισχύσει ένα ακριβώς ή μικρό αριθμό από αυτά τα μονοπάτια.



Σχήμα 3.2: Απλό σχήμα του DD. (a) Διάδοση interest. (b) Αρχικοποίηση gradient. (c) Μεταφορά δεδομένων σε ενισχυμένα μονοπάτια.

Στη συνέχεια περιγράφονται στοιχεία του Directed Diffusion όπως προκύπτουν από αναφορές σε συγκεκριμένο είδος δικτύων αισθητήρων – δηλαδή σε αυτά που ασχολούνται με τον προσδιορισμό της θέσης. Είναι χαρακτηριστικό ότι αρκετές σχεδιαστικές επιλογές παρουσιάζουν φοβερές ομοιότητες με αυτό το παράδειγμα. Η αξιολόγηση στρέφεται μόνο σε ένα υποσύνολο από όλες τις πιθανές εφαρμογές του Directed Diffusion. Οι διαφορετικές επιλογές εφαρμογής του Directed Diffusion οδηγούν σε διαφορετικές συμπεριφορές. Ακόμα εξετάζοντας την παραλλαγή του Directed Diffusion για rate-based εφαρμογές, διαπιστώνουμε ότι το πρωτόκολλο έχει διαφορετική συμπεριφορά σε εφαρμογές που πυροδοτούνται από την εμφάνιση κάποιου συμβάντος (event-trigger).

3.2.1. Naming

Στο Directed Diffusion οι περιγραφές των tasks ονομάζονται από έναν κατάλογο ζευγαριών ιδιοτήτων-τιμών που περιγράφουν το task. Ένα task εντοπισμού οχημάτων μπορεί να περιγραφεί ως εξής :

τύπος (type)	=	όχημα τροχοφόρο	//ανιχνεύει τη θέση των οχημάτων
διάστημα (interval)	=	20 ms	//στέλνει τα γεγονότα κάθε 20 ms
διάρκεια(duration)	=	10 s	//για τα επόμενα 10 s
rect	=	[-100 100 200 400]	//από τους αισθητήρες εντός του ορθογώνιου

Για ευκολία επιλέγεται η αναπαράσταση της υποπεριοχής να είναι ένα ορθογώνιο που ορίζεται σε κάποιο σύστημα συντεταγμένων. Συνήθως βασίζεται σε GPS συντεταγμένες. Η περιγραφή του task αναδεικνύει το ενδιαφέρον για το ταίριασμα των δεδομένων με τις τιμές. Για αυτόν τον λόγο, μια τέτοια περιγραφή του task καλείται ενδιαφέρον (interest). Τα δεδομένα που στέλνονται ως απάντηση στα ενδιαφέροντα ονομάζονται και αυτά χρησιμοποιώντας ένα παρόμοιο ονοματολογικό σχήμα. Συνεπώς, ένας αισθητήρας που ανιχνεύει ένα τροχοφόρο όχημα στέλνει τα ακόλουθα στοιχεία:

τύπος (type) = τροχοφόρο όχημα	//τύπος οχήματος που εντοπίστηκε
διάστημα (interval) = φορτηγό	//στιγμιότυπο αυτού του τύπου
θέση (location) = [125 220]	//θέση κόμβων
ένταση (intensity) = 0.6	//μέτρο εύρους σημάτων
ένταση (intensity) = 0.85	//μέτρο εκτίμησης της αντιστοιχίας
timestamp = 01:20:40	//χρόνος παραγωγής του γεγονότος

Λαμβάνοντας υπόψη ένα σύνολο από tasks υποστηριζόμενα από ένα δίκτυο αισθητήρων και επιλέγοντας ένα ονομάτολογικό σχήμα, έχουμε προβεί στο πρώτο βήμα του σχεδιασμού του Directed Diffusion για το δίκτυο. Για το δίκτυο αισθητήρων έχει επιλέγει ένα απλό ονοματολογικό σχήμα, βασισμένο στο ζευγάρι ιδιότητας-τιμής για τα ενδιαφέροντα και τα δεδομένα. Κάθε ιδιότητα (attribute) έχει ένα αντιστοιχιζόμενο εύρος τιμών. Παραδείγματος χάριν, το εύρος της ιδιότητας type αποτελείται από ένα σύνολο τιμών που αντιπροσωπεύουν κινούμενα αντικείμενα (οχήματα, ζώα, άνθρωποι). Η τιμή μιας ιδιότητας μπορεί να είναι οποιοδήποτε υποσύνολο του εύρους της. Στο παράδειγμά μας, η τιμή της ιδιότητας type για το ενδιαφέρον, είναι αυτή που αντιστοιχίζεται στα τροχοφόρα οχήματα.

Υπάρχουν και άλλες επιλογές για το εύρος τιμών των ιδιοτήτων (π.χ. ιεραρχικές) και άλλα σχήματα ονομασίας (όπως τα σκόπια ονόματα). Ως ένα ορισμένο βαθμό, η επιλογή του σχήματος ονομασίας μπορεί να έχει επιπτώσεις στα tasks και στην αποδοτικότητα του αλγορίθμου του Directed Diffusion. Εδώ ο στόχος είναι να κερδηθεί μια αρχική κατανόηση του Directed Diffusion. Για αυτόν τον λόγο, η αναζήτηση πιθανών σχημάτων ονομασίας δεν θα επεκταθεί.

3.2.2. Ενδιαφέροντα

Ένα ενδιαφέρον εισάγεται συνήθως στο δίκτυο από κάποιον (ενδεχομένως αυθαίρετο) κόμβο του. Χρησιμοποιούμε όρο καταβόθρα χαρακτηρίσουμε αυτό τον κόμβο.

Διάδοση Ενδιαφερόντων (interests): Δεδομένης της επιλογής του σχήματος ονομασίας, περιγράφεται τώρα πώς τα ενδιαφέροντα διαχέονται μέσω του δικτύου αισθητήρων. Υποθέτουμε ότι ένα task, με διάρκεια (duration) 10 min και χρονικού διαστήματος (interval) 10 ms, λαμβάνει χώρα σε έναν συγκεκριμένο κόμβο του δικτύου. Το interval καθορίζει ένα ρυθμό από δεδομένα, που αφορούν γεγονότα (events). Κατά συνέπεια, ο καθορισμένος ρυθμός δεδομένων είναι 100 events ανά δευτερόλεπτο. Ο κόμβος καταβόθρα καταγράφει το task. Το task εκκαθαρίζεται (παύει να υπάρχει) από τον κόμβο μετά το χρόνο που υποδεικνύεται από το duration.

Για κάθε ενεργό task η καταβόθρα εκπέμπει περιοδικά μηνύματα ενδιαφέροντος σε κάθε έναν από τους γείτονές της (αποδοτικότερες μέθοδοι αποστολής ενδιαφερόντων θα συζητηθούν αργότερα). Αυτό το αρχικό ενδιαφέρον περιέχει μία ορισμένη περιοχή (rect), την ιδιότητα της διάρκειας (duration) αλλά και ένα πολύ μεγαλύτερο χρονικό διάστημα (interval). Το αρχικό ενδιαφέρον μπορεί να θεωρηθεί ως διερευνητικό (exploratory) καθώς προσπαθεί να καθορίσει, εάν πράγματι υπάρχουν κόμβοι αισθητήρων που ανιχνεύουν το τροχοφόρο όχημα. Για να επιτευχθεί αυτό, το αρχικό διερευνητικό ενδιαφέρον ορίζει ένα χαμηλό ρυθμό αποστολής δεδομένων (παράδειγμά ένα event ανά δευτερόλεπτο). Αργότερα θα περιγράψει, πώς ο επιθυμητός ρυθμός δεδομένων επιτυγχάνεται με τη χρήση ενισχυτικών διαδικασιών. Μετά τα παραπάνω, το αρχικό ενδιαφέρον λαμβάνει την ακόλουθη μορφή:

```
τύπος (type) = τροχοφόρο όχημα
διάστημα (interval) = 1 s
περιοχή (rect) = [- 100 200 200 400]
timestamp = 01: 20: 40 //hh:mm:ss
Λήξη (expiresAt) = 01: 30: 40
```

Προτού περιγράψει το πώς τα ενδιαφέροντα υποβάλλονται σε επεξεργασία, υπογραμμίζεται ότι το ενδιαφέρον είναι soft state και περιοδικά θα ανανεώνεται από την καταβόθρα. Για γίνει αυτό, η καταβόθρα απλά στέλνει εκ νέου το ίδιο ενδιαφέρον με ένα μονοτονικά αυξανόμενο timestamp. Αυτό είναι απαραίτητο επειδή τα ενδιαφέροντα δεν διαβιβάζονται με αξιόπιστο τρόπο σε όλο το δίκτυο. Ο ρυθμός ανανέωσης αποτελεί μια παράμετρο του πρωτοκόλλου, η οποία ισορροπεί ανάμεσα στις ανάγκες για αυξημένη ευρωστία και στα χαμένα ενδιαφέροντα.

Κάθε κόμβος διατηρεί μια μνήμη (cache) ενδιαφερόντων. Κάθε καταχώρηση στην μνήμη αντιστοιχεί σε ξεχωριστό ενδιαφέρον. Δύο ενδιαφέροντα διακρίνονται εάν η παράμετρος type διαφέρει ή όταν το rect δεν είναι ίδιο. Οι οντότητες των ενδιαφερόντων μέσα στην cache δεν περιέχουν πληροφορίες για την καταβόθρα, παρά μόνο το αμέσως προηγούμενο hop. Από τον ορισμό των διακριτών ενδιαφερόντων μας δίνεται δυνατότητα για παραπέρα επεξεργασία. Δυο ενδιαφέροντα I_1 και I_2 με ίδια types και εντελώς επικαλυπτόμενα rect, μπορούν σε μερικές περιπτώσεις να αναπαρασταθούν ως ένα απλό ενδιαφέρον.

Η καταχώρηση στη μνήμη των ενδιαφερόντων εμφανίζει πολλά πεδία. Το πεδίο timestamp αναφέρεται στο τελευταίο παραληφθέν ενδιαφέρον, που ταιριάζει. Εδώ ας σημειωθεί ότι χρειάζεται χρονικός συγχρονισμός μεταξύ των κόμβων του δικτύου. Αυτό γίνεται είτε μέσω GPS είτε μέσω post-facto μηνυμάτων. Η καταχώρηση του ενδιαφέροντος περιέχει αρκετά πεδία από gradients, όσα και οι γείτονες. Κάθε gradient περιέχει ένα πεδίο ρυθμού δεδομένων προς ένα συγκεκριμένο γείτονα, το οποίο προκύπτει από το χρονικό διάστημα (interval) του ενδιαφέροντος. Επίσης περιέχει ένα πεδίο διάρκειας (duration) όπως αυτό προκύπτει από τις ιδιότητες timestamp και expiresAt των δεδομένων. Αυτή η διάρκεια πρέπει να είναι μακρύτερη από την καθυστέρηση του δικτύου.

Όταν ένας κόμβος λαμβάνει ένα ενδιαφέρον τσεκάρει εάν αυτό βρίσκεται στη μνήμη του. Αν δεν εντοπίσει κάποια καταχώρηση που ταιριάζει τότε δημιουργεί μια νέα. Οι παράμετροι από την καταχώρηση του ενδιαφέροντος δημιουργούνται από τον λαμβανόμενο ενδιαφέρον. Αυτή η καταχώρηση έχει ένα gradient προς το γείτονα από τον οποίο έλαβε το ενδιαφέρον καθώς και ένα συγκεκριμένο ρυθμό δεδομένων από εισερχόμενα γεγονότα (events). Στο παράδειγμα που εξετάζουμε εδώ ο γείτονας της καταβόθρας, θα δημιουργήσει μία καταχώρηση ενδιαφέροντος με gradient ρυθμού ενός γεγονότος ανά δευτερόλεπτο προς την καταβόθρα. Για αυτό είναι αδήρητος ανάγκη να είμαστε σε θέση να ξεχωρίσουμε τους γείτονες. Οποιαδήποτε τοπική μοναδική ταυτότητα μπορεί να χρησιμοποιηθεί για τον παραπάνω σκοπό. Τέτοιες ταυτότητες είναι οι 802.11 MAC διευθύνσεις, οι Bluetooth cluster διευθύνσεις ή άλλες τοπικές μοναδικές ταυτότητες. Αν υπάρχει μια καταχώρηση ενδιαφέροντος αλλά όχι ένα gradient προς τον αποστολέα του ενδιαφέροντος, ο κόμβος προσθέτει το gradient. Επιπλέον ανανεώνει τα πεδία timestamp και της χρονικής διάρκειας (duration) της καταχώρησης. Αν τώρα έχουμε τόσο καταχώρηση όσο και gradient ο κόμβος απλά προβαίνει σε ανανέωση των πεδίων timestamp και duration.

Αργότερα θα περιγραφεί το πώς χρησιμοποιούνται τα gradients. Όταν το gradient λήγει απομακρύνεται από την καταχώρηση του ενδιαφέροντος. Πρέπει να σημειωθεί ότι δεν λήγουν ταυτόχρονα όλα τα gradients. Δυο διαφορετικές καταβόθρες στέλνουν διαφορετικά ενδιαφέροντα με διαφορετικούς χρόνους λήξης. Τότε μερικοί κόμβοι θα έχουν μία καταχώρηση ενδιαφέροντος με

διαφορετικό χρόνο λήξης για το gradient. Μόλις όλα τα gradients για μια καταχώρηση ενδιαφέροντος εκπνεύσουν, τότε η καταχώρηση του ενδιαφέροντος διαγράφεται από την μνήμη.

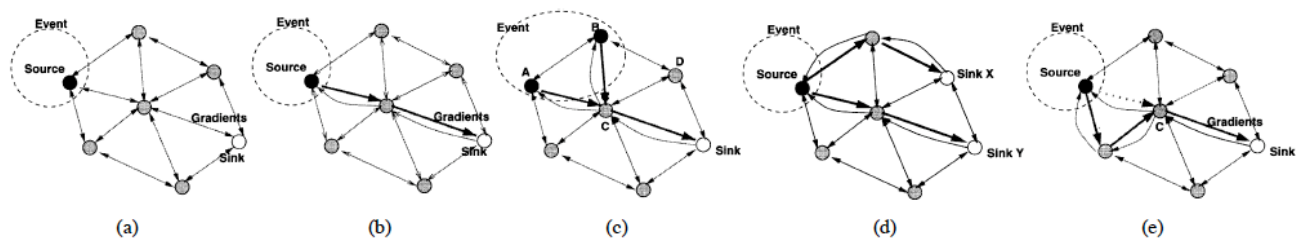
Αφού έχει ληφθεί ένα ενδιαφέρον ο κόμβος μπορεί να ξανά στείλει το ενδιαφέρον σε μερικούς από τους γείτονες του. Σε αυτούς το ενδιαφέρον θα εμφανίζεται ως έχει προέλευση από τον αποστολέα κόμβο. Αν και μπορεί να προέρχεται από μια απομακρυσμένη καταβόθρα. Το προαναφερθέν αποτελεί παράδειγμα τοπικής αλληλεπίδρασης. Έτσι λέμε πως έχουμε διάχυση του ενδιαφέροντος στο δίκτυο. Ένας κόμβος μπορεί να κόψει ένα λαμβανόμενο ενδιαφέρον εάν πρόσφατα έχει στείλει ένα παρόμοιο.

Γενικά υπάρχει πλειάδα πιθανών επιλογών για τους γείτονες. Η απλούστερη εναλλακτική είναι η επαναμετάδοση του ενδιαφέροντος σε όλους τους γείτονες. Με λίγα λόγια μιλάμε για αλγόριθμους πλημμύρας (flooding) στο δίκτυο. Όταν δεν υπάρχει πληροφορία για το ποιό κόμβοι αναμένεται να ικανοποιήσουν το ενδιαφέρον τότε η πλημμύρα καθίσταται μονόδρομος. Σε άλλες εκδοχές είναι δυνατή η γεωγραφική δρομολόγηση, η οποία οδηγεί στο περιορισμό του δικτύου σε συγκεκριμένους κόμβους. Κατά συνέπεια και σε εξοικονόμηση ενέργειας. Τέλος σε ένα ακίνητο δίκτυο αισθητήρων ο κόμβος μπορεί να χρησιμοποιεί αποθηκευμένα δεδομένα κατευθείαν ως ενδιαφέροντα. Για παράδειγμα ένας κόμβος μπορεί να λάβει νεότερα δεδομένα από ένα γείτονα A, για ένα προηγούμενο ενδιαφέρον και εν συνεχεία να κατευθύνει αυτό το ενδιαφέρον στο A. Τα δεδομένα εστάλησαν από κάποιο αισθητήρα μέσα στην προσδιοριζόμενη περιοχή. Δηλαδή αποφεύγεται η ραδιοφωνική αναμετάδοση σε όλους τους γείτονες.

Στοιχεία του Diffusion	Σχεδιαστικές επιλογές
Διάδοση interest	<ul style="list-style-type: none"> • Flooding • Κατευθυνόμενο flooding βασισμένο σε συντεταγμένες • Κατευθυνόμενη διάδοση βασισμένη σε αποθηκευμένα δεδομένα
Διάδοση δεδομένων	<ul style="list-style-type: none"> • Ενίσχυση ενός και μόνο μονοπατιού για τη μεταφορά • Πολλαπλή μεταφορά με επιλογή ποιότητας μεταξύ των μονοπατιών • Πολλαπλή μεταφορά με πιθανοτική προώθηση
Επεξεργασία αποθήκευση δεδομένων	<ul style="list-style-type: none"> • Για εύρωστη μεταφορά data λόγω αποτυχιών στους κόμβους • Για υπό συνθήκες sensing και μείωση άρχιστων δεδομένων • Για κατευθυνόμενα interests
Ενίσχυση	<ul style="list-style-type: none"> • Κανόνες απόφασης για τη χρήση ενίσχυσης • Κανόνες για το αριθμό των γειτόνων που πρέπει να ενισχυθούν • Μηχανισμοί και κανόνες αρνητικής ενίσχυσης

Πίνακας 3.2.2: Σχεδιαστικές επιλογές για τις διάφορες φάσεις του Directed Diffusion

3.2.3. Gradients



Σχήμα 3.2.3: Παρουσιάζονται διαφορετικές πτυχές του Diffusion. (a) Παγίωση των gradients. (b) Ενίσχυση. (c) Πολλαπλές πηγές. (d) Επισκευή.

Σημειωτέων κάθε ζευγάρι των γειτονικών κόμβων διαθέτει gradient αναμεταξύ τους. Πράγμα που αποτελεί σημαντική απόρροια των τοπικών αλληλεπιδράσεων. Όταν ένας κόμβος λαμβάνει ένα ενδιαφέρον από το γείτονά του, δεν διαθέτει κανένα τρόπο αποσαφήνισης για το αν αποτελεί απάντηση σε ενδιαφέρον που εστάλει νωρίτερα. Ακόμα μπορεί να είναι όμοιο με κάποιο ενδιαφέρον που προέρχεται από άλλη καταβόθρα. Τέτοια αμφίδρομα gradient μπορούν να οδηγήσουν έναν κόμβο στο να λάβει ένα αντίγραφο χαμηλού ρυθμού δεδομένων από κάθε γείτονα του. Εντούτοις, όπως παρουσιάζεται αργότερα, αυτή η τεχνική μπορεί να επιτρέψει τη γρήγορη αποκατάσταση των αποτυχημένων μονοπατιών, την ενίσχυση των εμπειρικά καλύτερων μονοπατιών καθώς και στην αποφυγή βρόχων.

Στο δίκτυο αισθητήρων ένα gradient χαρακτηρίζεται από το ρυθμό δεδομένων και την κατεύθυνση στην οποία στέλνει τα γεγονότα. Γενικότερα ένα gradient ορίζει μια τιμή και μια κατεύθυνση. Το πρότυπο του Directed Diffusion δίνει στο σχεδιαστή την ελευθερία να συνδέει σημασιολογία (semantics) που ποικίλει, με τις τιμές (values) του gradient. Στα δίκτυα αισθητήρων τα gradient έχουν δύο τιμές που καθορίζουν το ρυθμό αναφοράς των συμβάντων. Σε άλλα δίκτυα αισθητήρων οι τιμές του gradient μπορούν να χρησιμοποιηθούν πιθανολογικά για να προωθήσουν δεδομένα κατά μήκος διαφορετικών μονοπατιών. Με αυτόν τον τρόπο επιτυγχάνουμε κάποια εξισορρόπηση στο φόρτο που προκύπτει από τα δεδομένα.

Περιληπτικά η διάδοση των ενδιαφερόντων ετοιμάζει το δίκτυο (ή μέρη αυτού) για να διευκολυνθεί η μεταφορά των δεδομένων προς την καταβόθρα. Οι κανόνες διάδοσης ενδιαφερόντων εμφανίζουν τοπικότητα και παρουσιάζουν ομοιότητα με τη διάδοση σε ορισμένα multicast (πολλαπλών διανομών) πρωτόκολλα δρομολόγησης στο Διαδίκτυο. Εδώ μια σημαντική διαφορά είναι ότι μπορούμε να έχουμε unicast πίνακες δρομολόγησης προς τις πηγές, πράγμα μη εφικτό στη διάδοση ενδιαφερόντων.

Εως τώρα έχουμε περιγράψει τους κανόνες διάδοσης ενδιαφέροντος για ένα συγκεκριμένο task. Γενικότερα ένα δίκτυο αισθητήρων μπορεί να υποστηρίξει πολλούς και διαφορετικούς τύπους από tasks. Οι κανόνες διάδοσης ενδιαφέροντος είναι διαφορετικοί για τα διαφορετικά tasks. Παραδείγματος χάριν, ένα είδος task όπως το «μετρήστε τον αριθμό το διακριτών τροχοφόρων οχημάτων στο ορθογώνιο P, που εντοπίζονται κατά τη διάρκεια των επόμενων δευτερολέπτων» δεν μπορεί να ανιχνεύσει το ρυθμό δεδομένων από τα γεγονότα. Σε αντίθεση με παράδειγμά που έχει παρουσιαστεί μέχρι τώρα. Όμως μερικά στοιχεία της διάδοσης ενδιαφέροντος είναι παρόμοια και στα δύο: η μορφή των καταχωρήσεων στη cache, των κανόνων αναδιανομής των ενδιαφερόντων, κλπ.

3.2.4. Διάδοση δεδομένων

Ένας κόμβος αισθητήρων που βρίσκετε μέσα σε συγκεκριμένη περιοχή, επεξεργάζεται τα ενδιαφέροντα όπως περιγράφηκε προηγουμένως. Επιπλέον ο κόμβος θέτει τον τοπικό του αισθητήρα σε κατάσταση συλλογής δειγμάτων (για εξοικονόμηση ενέργειας, οι αισθητήρες είναι κλειστοί αρχικά). Εδώ δεν συζητούνται λεπτομέρειες αλγορίθμων αναγνώρισης. Εν συντομία αυτοί αλγόριθμοι απλά αντιστοιχίζουν δείγματα κυματομορφών με τη βοήθεια μιας βιβλιοθήκης αποθηκευμένων κυματομορφών. Όλα αυτά βασίζονται στο γεγονός ότι ένα τροχοφόρο όχημα έχει ένα διαφορετικό ακουστικό ή σεισμικό αποτύπωμα από έναν άνθρωπο. Οι δειγματοληπτημένες κυματομορφές μπορούν να ταιριάζουν με τις αποθηκευμένες κυματομορφές σε κάποιο βαθμό. Οι αλγόριθμοι συνήθως περιέχουν βαθμό εμπιστοσύνης στο κατά πόσο έχουμε αντιστοιχία. Επιπλέον η ένταση των δειγματοληπτημένων κυματομορφών μπορεί να αναδείξει κατά προσέγγιση την απόσταση της προέλευσης των σημάτων, όχι όμως και την κατεύθυνση.

Ένας κόμβος αισθητήρων που ανιχνεύει έναν στόχο ανατρέχει στην cache ενδιαφερόντων για να εντοπίσει πιθανά ταιριάσματα. Σε αυτήν την περίπτωση, μια καταχώρηση που ταιριάζει θα έχει rect που εμπεριέχει τη θέση του αισθητήρα. Ακόμα η ιδιότητα type της καταχώρησης θα ταιριάζει με το type του ανιχνεύσιμου στόχου. Στη συνέχεια υπολογίζει τον υψηλότερο ρυθμό λήψης των ζητούμενων γεγονότων, ανάμεσα σε όλους τους εξερχόμενους gradient του. Οι κόμβοι παράγουν δείγματα γεγονότων με αυτό το υψηλότερο ρυθμό δεδομένων. Στο παράδειγμά μας, αυτός ο ρυθμός δεδομένων είναι αρχικά ένα γεγονός ανά δευτερόλεπτο (έως ότου εφαρμοστούν οι τεχνικές ενίσχυσης που θα δούμε παρακάτω). Η πηγή στέλνει κάθε δευτερόλεπτο, σε γείτονες με τους οποίους έχει gradient, μια περιγραφή γεγονότος της μορφής:

```
τύπος (type) = τροχοφόρο όχημα //τύπος οχήματος που εντοπίζεται
περίπτωση (instance) = φορτηγό //περίπτωση από τον τύπο
θέση (location) = [125 220] //θέση κόμβων
ένταση (intensity) = 0.6 //μετρική εύρους σήματος
εμπιστοσύνη (confidence) = 0.85 //εμπιστοσύνη για την αντιστοιχία
timestamp = 01:20:40 // χρόνος παραγωγής τοπικού γεγονότος
```

Αυτό το μήνυμα δεδομένων κατευθύνεται με unicast τρόπο προς τους σχετικούς γείτονες. Ο ακριβής μηχανισμός προκύπτει από τη χρήση μίας συνάρτησης του MAC επιπέδου πράγμα που έχει εξαιρετική σημασία στο τομέα της απόδοσης.

Ένας κόμβος που λαμβάνει ένα μήνυμα δεδομένων από τους γείτονές του προσπαθεί να βρει ταιριάσματα με καταχωρίσεις ενδιαφερόντων στην μνήμη του. Ο κανόνας για το ταίριασμα είναι όπως έχει περιγραφεί σε προηγούμενη παράγραφο. Εάν δεν βρεθεί καμία αντιστοιχία, τα μηνύματα των δεδομένων χάνονται. Όταν όμως υπάρχει αντιστοιχία ο κόμβος ελέγχει την cache των δεδομένων, η οποία συνδέεται με το ταίριασμα των καταχωρημένων ενδιαφερόντων. Αυτή η cache περιέχει δεδομένα που έχουν ληφθεί πρόσφατα. Οι πιθανές χρήσεις της ανάγονται σε αρκετές, όπως η αποφυγή βρόχων. Ένα ληφθέν μήνυμα δεδομένων αγνοείται, όταν ταιριάζει με την καταχώρηση στη cache των δεδομένων. Διαφορετικά το ληφθέν μήνυμα προστίθεται στην cache των δεδομένων και το μήνυμα των δεδομένων προωθείται στους γείτονες του κόμβου.

Με προσπέλαση της μνήμης δεδομένων ο κόμβος δύναται να καθορίσει το ρυθμό των λαμβανομένων δεδομένων. Ένας κόμβος για να στείλει εκ νέου ένα ληφθέν μήνυμα δεδομένων, οφείλει να εξετάσει τη λίστα των gradient με τις καταχωρήσεις ενδιαφερόντων. Εάν όλα τα gradient

έχουν ρυθμό δεδομένων μεγαλύτερο ή ίσο με το ρυθμό των εισερχόμενων γεγονότων, ο κόμβος μπορεί κάλλιστα να στείλει τα μηνύματα δεδομένων στο σωστό γείτονα. Αν πάλι ορισμένοι gradient έχουν χαμηλότερο ρυθμό από άλλους, ο κόμβος θα πρέπει περιοριστεί (downconvert) στους κατάλληλους gradient. Επί παραδείγματι μπορεί να έχουμε λήψη 100 γεγονότων ανά δευτερόλεπτο, αλλά ένας gradient να βρίσκεται στα 50 γεγονότα ανά δευτερόλεπτο. Σε αυτήν την περίπτωση ο κόμβος διαβιβάζει εναλλακτικά κάθε γεγονός προς τον αντίστοιχο γείτονα. Επίσης μπορεί να παρεμβάλει (interpolate) δύο διαδοχικά γεγονότα. Ο τρόπος της παρεμβολής ορίζεται από την εφαρμογή (να επιλέξει το δείγμα που εμφανίζει την υψηλότερη αντιστοιχία εμπιστοσύνης). Η πρόληψη στη δημιουργία βρόχων και ο περιορισμός (downconversion) αναδεικνύουν τη δύναμη της ενσωμάτωσης των σχημάτων (semantics) των εφαρμογών σε όλους τους κόμβους. Αν και αυτό το σχήμα δεν αρμόζει σε παραδοσιακά δίκτυα, είναι αποτελεσματικό σε δίκτυα αισθητήρων που εξαρτώνται από την εφαρμογή. Εμφανίζει σημαντική βελτίωση στην απόδοση αυτών των δικτύων.

3.2.5. Τεχνικές ενίσχυσης στην καθιέρωση και στην αποκοπή μονοπατιών

Στο σχήμα που έχει περιγραφεί οι καταβόθρες επανειλημμένα διαχέουν ενδιαφέροντα. Αυτά τα ενδιαφέροντα προέρχονται από χαμηλού ρυθμού ανακοινώσεις γεγονότων. Τα γεγονότα καλούνται διερευνητικά αφού προορίζονται για την αρχική οργάνωση (set up) και επισκευή των μονοπατιών. Οι gradients που διαμορφώνονται από διερευνητικά γεγονότα, καλούνται διερευνητικοί gradients. Μόλις μια πηγή ανιχνεύσει έναν στόχο που ταιριάζει, στέλνει τα διερευνητικά γεγονότα προς την καταβόθρα. Πράγμα που ενδεχομένως γίνεται κατά μήκος πολλαπλών μονοπατιών. Αφού οι καταβόθρες αρχίσουν να λαμβάνουν τα διερευνητικά γεγονότα, προβαίνουν στην ενίσχυση του συγκεκριμένου γείτονα. Αυτό γίνεται προκειμένου να «αντλήσουν» τα πραγματικά δεδομένα (δηλ. γεγονότα με υψηλότερο ρυθμό δεδομένων που επιτρέπει ποιοτικότερο εντοπισμό στόχων). Έτσι ονομάζουμε τα gradient που δημιουργούνται για τη λήψη των υψηλής ποιότητας εντοπιζόμενων γεγονότων, gradients δεδομένων.

1. Καθιέρωση Μονοπατιών με τη χρήση τεχνικών ενίσχυσης

Γενικά το χαρακτηριστικό γνώρισμα που διαφοροποιεί το Directed Diffusion έγγειται στο ότι τα δεδομένα ακολουθούν τοπικούς κανόνες. Ένα παράδειγμα ενός τέτοιου κανόνα είναι η ενίσχυση οποιουδήποτε γείτονα από το οποίο ο κόμβος έλαβε ένα γεγονός για πρώτη φορά. Για να ενισχύσει αυτόν τον γείτονα, η καταβόθρα στέλνει εκ νέου το αρχικό μήνυμα ενδιαφέροντος αλλά με μικρότερο χρονικό περιθώριο (υψηλότερος ρυθμός δεδομένων). Συνεπώς έχουμε την παρακάτω μορφή:

```
τύπος (type) = τροχοφόρο όχημα
διάστημα (interval) = 10 ms
rect = [-100 200 200 400]
timestamp = 01: 22: 35
expiresAt = 01: 30: 40
```

Όταν ο γειτονικός κόμβος λάβει αυτό το ενδιαφέρον παρατηρεί, ότι έχει ήδη gradient προς αυτόν τον γείτονα. Επιπλέον, παρατηρεί ότι το ενδιαφέρον του αποστολέα περιέχει ένα υψηλότερο ρυθμό

δεδομένων από πριν. Εάν αυτός ο νέος ρυθμός δεδομένων είναι υψηλότερος από οποιουδήποτε gradient (διαισθητικά, εάν η «εκροή» από αυτόν τον κόμβο έχει αυξηθεί) ο κόμβος οφείλει να ενισχύσει τουλάχιστον έναν γείτονα. Ο κόμβος χρησιμοποιεί την cache των δεδομένων του για αυτόν τον σκοπό. Οι ίδιες τοπικές επιλογές κανόνων ισχύουν και εδώ. Παραδείγματος χάριν, ο κόμβος επιλέγει εκείνο τον γείτονα από τον οποίο έλαβε αρχικά το πιο πρόσφατο γεγονός. Αυτό το γεγονός πρέπει να ταιριάζει με το ενδιαφέρον. Εναλλακτικά, επιλέγονται όλοι οι γείτονες που αναφέρουν νέα γεγονότα. Αυτά τα γεγονότα έχουν παραληφθεί πρόσφατα. Πράγμα που υπονοεί ότι ενισχύουμε εκείνο τον γείτονα που στέλνει διερευνητικά γεγονότα. Προφανώς, δεν πρέπει να ενισχυθούν οι γείτονες που δημιουργούν πρόβλημα στην κυκλοφορία (traffic) καθώς αποστέλλουν με υψηλό ρυθμό δεδομένων. Μέσω αυτής της ακολουθίας τοπικών αλληλεπιδράσεων καθιερώνεται ένα μονοπάτι από την πηγή στην καταβόθρα και μεταδίδονται τα δεδομένα.

Ο τοπικός κανόνας που περιγράψαμε παραπάνω, στη συνέχεια επιλέγει ένα εμπειρικό μονοπάτι χαμηλής καθυστέρησης. Επίσης είναι ευαίσθητος στις αλλαγές της ποιότητας των μονοπατιών. Όποτε ένα μονοπάτι παραδίδει ένα γεγονός γρηγορότερα από άλλα, η καταβόθρα το χρησιμοποιεί για να λάβει υψηλής ποιότητας δεδομένα. Επειδή όλα αυτά πυροδοτούνται με τη λήψη ενός νέου γεγονότος, ενδεχομένως οδηγηθούμε σε σπατάλη πόρων. Πιο εκλεπτυσμένοι τοπικοί κανόνες είναι δυνατόν να υπάρξουν. Τέτοιοι είναι η επιλογή εκείνου του γείτονα από τον οποίο τα περισσότερα γεγονότα έχουν παραληφθεί. Μια ακόμα επιλογή θα ήταν ο γείτονας που στέλνει με συνέπεια τα γεγονότα πριν από τους άλλους. Αυτές οι επιλογές ισορροπούν ανάμεσα στην ικανότητα άμεσης αντίδρασης και τη σταθερότητα.

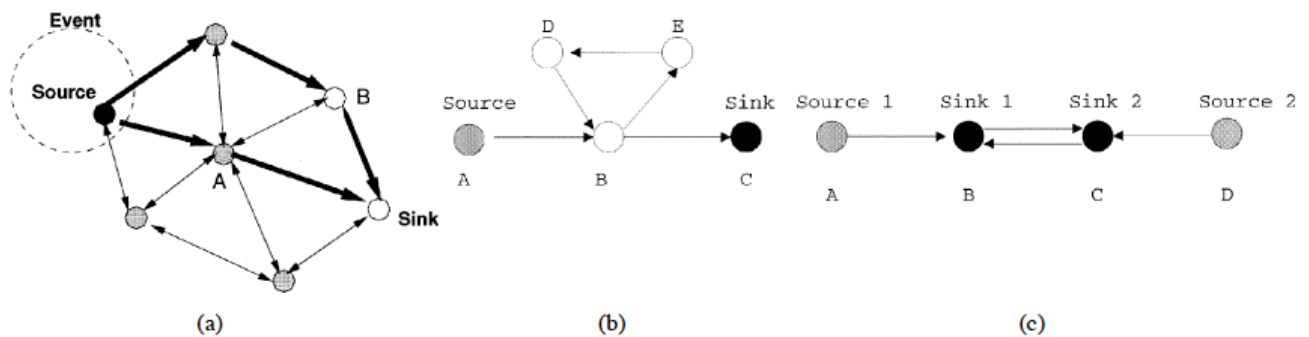
2. Καθιέρωση Μονοπατιών για πολλαπλές πηγές και καταβόθρες

Περιγράφοντας τις τεχνικές ενίσχυσης μέχρι τώρα, σιωπηρά έχει γίνει η παραδοχή ύπαρξης μόνο μιας πηγής. Στην πραγματικότητα οι κανόνες που έχουν περιγράψει βρίσκουν εφαρμογή και για πολλαπλές πηγές. Για να αντιληφθούμε το παραπάνω θεωρούμε Σχ.3.2.2 (c). Υποθέτουμε αρχικά ότι όλοι οι gradients είναι διερευνητικοί. Σύμφωνα με αυτήν την τοπολογία δεδομένα και από τις δύο πηγές φθάνουν στην καταβόθρα μέσω των γειτόνων τους C και D. Εάν ένας από τους γείτονες, π.χ. ο C έχει συνεχώς τη χαμηλότερη καθυστέρηση, οι κανόνες θα ενισχύσουν μόνο την πορεία μέσω του C (αυτό απεικονίζεται στη εικόνα). Εάν η καταβόθρα λάβει τα γεγονότα του B νωρίτερα μέσω του D, αλλά του A νωρίτερα μέσω του C, θα προσπαθήσει να τραβήξει τα υψηλής ποιότητας ρεύματα δεδομένων (streams) και από τους δύο γείτονες. Σε αυτήν την περίπτωση η καταβόθρα συγκεντρώνει τα δεδομένα και των δύο πηγών. Το κόστος όμως είναι εμφανές στο ενεργειακό τομέα. Τέτοιο προβλήματα αποφεύγονται με την εισαγωγή κάποιας επιμέρους πολυπλοκότητας.

Ομοίως και όταν δύο καταβόθρες εκφράζουν ίδια ενδιαφέροντα, η διάδοση ενδιαφερόντων, η καθιέρωση των gradient και οι κανόνες ενίσχυσης δουλεύουν απρόσκοπτα. Χωρίς να απωλέσουμε τη γενικότητα, υποθέτουμε ότι η καταβόθρα Y στο σχήμα 3.2.2(d) έχει ενισχύσει ήδη ένα υψηλής ποιότητας μονοπάτι προς την πηγή. Σημειώνουμε, ότι οι άλλοι κόμβοι συνεχίζουν να λαμβάνουν διερευνητικά γεγονότα. Όταν ο ανθρώπινος χειριστής αναθέτει στη καταβόθρα X ένα ίδιο ενδιαφέρον, η X μπορεί να χρησιμοποιήσει τους κανόνες ενίσχυσης. Έτσι επιτυγχάνεται η πορεία που παρουσιάζεται. Για να καθορίσει την εμπειρικά καλύτερη πορεία το X δεν χρειάζεται να περιμένει για δεδομένα, αφού μπορεί να χρησιμοποιήσει την cache των δεδομένων του.

3. Τοπική επίσκεψη των αποτυχημένων μονοπατιών

Μέχρι τώρα έχουν περιγράψει καταστάσεις στις οποίες η ενίσχυση προκαλείται από την καταβόθρα. Στο Directed Diffusion οι ενδιαμέσοι κόμβοι ενός προηγούμενως ενισχυμένου μονοπατιού, εφαρμόζουν τους κανόνες ενίσχυσης. Η χρησιμότητα έγγειται στο ότι επιτρέπεται η τοπική επισκευή των αποτυχημένων μονοπατιών. Οι αιτίες της αποτυχίας ή της υποβάθμισης περιλαμβάνουν την ενεργειακή μείωση κόμβων και τους περιβαλλοντικούς παράγοντες που επηρεάζουν την επικοινωνία (π.χ., εμπόδια). Εξετάζοντας το Σχ.3.2.2(e) βλέπουμε ότι η ποιότητα της σύνδεσης μεταξύ της πηγής και του κόμβου C φθίνει και τα γεγονότα συχνά αλλοιώνονται. Όταν το C ανιχνεύσει την υποβάθμιση μπορεί να εφαρμόσει τους κανόνες ενίσχυσης για να δημιουργηθεί το μονοπάτι που παρουσιάζεται στο σχήμα. Το παραπάνω γίνεται με την ανακοίνωση του γεγονός της μείωσης του ρυθμού από τον γείτονά του (η πηγή) ή όταν αντιλαμβάνεται ότι οι άλλοι γείτονες εκπέμπουν καινούριες τοπικές εκτιμήσεις. Τελικά, το C ενισχύει αρνητικά το άμεσο σύνδεσμο προς την πηγή (που δεν παρουσιάζεται στο σχήμα). Η περιγραφή μέχρι τώρα έχει βασιστεί στο γεγονός, ότι μια απλή εφαρμογή των κανόνων ενίσχυσης θα αναγκάσει όλους τους κόμβους να αποφύγουν τα επικίνδυνα link. Τελικά θα οδηγηθούμε στην ανακάλυψη μιας εμπειρικά καλής πορείας, με κόστος όμως στην σπατάλη πόρων. Ένας τρόπος να αποφευχθεί αυτό είναι το C να περιλάβει εκτιμήσεις θέσης από τα γεγονότα που λαμβάνουν χώρα.



Σχήμα 3.2.5: Αρνητική ενίσχυση. (a) Πολλαπλά μονοπάτια. (b) Αποφυγή loop. (c) loop που δεν αποφεύγονται

4. Αποκοπή μονοπατιών που χρησιμοποιούν αρνητική ενίσχυση

Το αλγόριθμος που έχει περιγραφεί, είναι δυνατόν να καταλήξει στην ενίσχυση παραπάνω του ενός μονοπατιού. Παραδείγματος χάριν, στο σχημ.3.2.5(a), εάν η καταβόθρα ενισχύει το γείτονα A και ταυτόχρονα λαμβάνει ένα νέο γεγονός από το γείτονα B, θα ενισχύσει το μονοπάτι μέσω του B. Αν το μονοπάτι μέσω του B είναι συνεχώς καλύτερο (δηλ. το B στέλνει τα γεγονότα προτού το A), θα χρειαστούμε έναν μηχανισμό για να ενισχύσουμε αρνητικά το μονοπάτι μέσω του A.

Ο μηχανισμός για την αρνητική ενίσχυση είναι soft state. Π.χ. να εκπνεύσουν όλοι οι gradient δεδομένων στο δίκτυο εκτός αν ενισχυθούν ρητά. Με αυτήν την προσέγγιση, η καταβόθρα θα ενισχύει περιοδικά το γείτονα B εν αντιθέση με το γείτονα A. Όλοι οι gradient κατά μήκος του μονοπατιού του A θα υποβιβαστούν τελικά σε διερευνητικά gradient. Μία ακόμα προσέγγιση αποτελεί και η απευθείας υποβάθμιση του μονοπατιού μέσω του A, με την αποστολή ενός μηνύματος αρνητικής ενίσχυσης. Σε αυτήν την ρυθμο-βασισμένη διάχυση, η αρνητική ενίσχυση χαρακτηρίζεται από το ενδιαφέρον με το χαμηλότερο ρυθμό δεδομένων. Όταν το A λάβει αυτό το ενδιαφέρον, υποβιβάζει τον gradient του, που βρίσκεται προς την καταβόθρα. Εάν όλοι οι gradient είναι τώρα διερευνητικοί, το A ενισχύει αρνητικά εκείνους τους γείτονες που του έχουν στείλει τα δεδομένα. Ο εν λόγω τοπικός κανόνας λειτουργεί ακόμα κι αν τα μονοπάτια μέσω του A και του B είναι μερικώς

κοινά. Οι κοινές συνδέσεις δεν θα ενισχυθούν αρνητικά εκτός αν και τα δύο μονοπάτια λάβουν τέτοια ενίσχυση. Αυτή η ακολουθία τοπικών αλληλεπιδράσεων εξασφαλίζει ότι το μονοπάτι μέσω του A υποβιβάζεται γρήγορα, αλλά με το κόστος της αυξανόμενης χρησιμοποίησης των πόρων.

Για να ολοκληρωθεί η περιγραφή της αρνητικής ενίσχυσης είναι ανάγκη να διευκρινιστεί ποιο τοπικό κανόνα χρησιμοποιεί ένας κόμβος. Από αυτό θα αποφασιστεί εάν θα ενισχύσει αρνητικά έναν γείτονα ή όχι. Σημειωτέον αυτός ο κανόνας είναι ορθογώνιος στην επιλογή του μηχανισμού της αρνητικής ενίσχυσης. Μια εύλογη επιλογή για έναν τέτοιο κανόνα αποτελεί η αρνητική ενίσχυση του γείτονα από το οποίο κανένα νέο γεγονός δεν έχει παραληφθεί. Για παράδειγμα άλλοι γείτονες έχουν στείλει με συνέπεια τα γεγονότα πριν από αυτόν. Όλα αυτά βέβαια μέσα σε ένα παράθυρο από N γεγονότα ή T χρόνο.

Ο τοπικός κανόνας που αξιολογούμε σε υστερότερη παράγραφο βασίζεται σε ένα χρονικό παράθυρο 2 s στην προσομοίωση. Ένας τέτοιος κανόνας θεωρείται αρκετά συντηρητικό και ενέργεια ανεπαρκής. Ακόμα κι αν ένα γεγονός από τα δέκα παραληφθεί πρώτα από το γείτονα A, η καταβόθρα δεν θα ενισχύσει αρνητικά εκείνο τον γείτονα. Άλλες παραλλαγές περιλαμβάνουν αρνητική ενίσχυση του γείτονα από τον οποίο λαμβάνουμε τα λιγότερα νέα γεγονότα.

5. Αφαίρεση βρόχων με χρήση αρνητικής ενίσχυση

Πέρα από την προσπάθεια ελέγχου της υψηλής καθυστέρησης ή των αποτυχημένων μονοπατιών, η αρνητική ενίσχυση χρησιμοποιείται και για την αφαίρεση βρόχων επειδή τα κυκλικά μονοπάτια δεν παραδίδουν ποτέ τα γεγονότα γρηγορότερα [Σχ.3.2.5(b)]. Αν και το κυκλικό (looping) μήνυμα θα εξαφανιστεί όταν θα γίνει χρήση της cache μηνυμάτων, γενικά επωφελούμαστε από την περικοπή κυκλικών μονοπατιών. Κυρίως στον τομέα της αποταμίευσης των πόρων. Παρόλλα αυτά η αφαίρεση των βρόχων δεν είναι πάντα ορθή επιλογή. Συγκεκριμένα για μερικούς κοινούς, υψηλού ρυθμού gradient-χάρτες με πολλαπλές πηγές και καταβόθρες. Παράδειγμα [Σχ.3.2.5(c)] αν και οι δύο πηγές στέλνουν διακριτά γεγονότα, οι gradient B-C και C-B δεν θα περικοπούν επειδή καθένας τους από θεωρείται απαραίτητος για την παράδοση των γεγονότων. Καθένας για το συγκεκριμένο ζεύγος πηγής – καταβόθρας. Αν και τέτοια gradient παραδίδουν μερικά επαναλαμβανόμενα γεγονότα, εν τούτοις παραδίδουν συστηματικά και νέα. Με ένα συντηρητικό κανόνα αρνητικής ενίσχυσης, εκείνα τα gradient δεν θα ενισχυθούν αρνητικά.

Ακόμη και χωρίς βρόχους είναι λογικό να κρατηθεί ο αρνητικός κανόνας ενίσχυσης συντηρητικός. Με αυτόν τον τρόπο οι χρήσιμες διαδρομές δεν θα περικοπούν. Παράδειγμα [Σχ.3.2.2(c)] και οι δύο πηγές με συνέπεια στέλνουν διακριτά γεγονότα αλλά ενίοτε στέλνουν και τα ίδια γεγονότα. Αν και προερχόμενα από διαφορετικές πηγές, τα ίδια γεγονότα θεωρούνται αντίγραφα για το directed diffusion. Το μονοπάτι μίας εκ των πηγών θα περικοπεί αν ο αρνητικός κανόνας ενίσχυσης είναι πάρα πολύ επιθετικός σε περιπτώσεις ύπαρξης αντιγράφων. Αντίθετα, με τη χρήση του συντηρητικού κανόνα καμία πηγή δεν ενισχύεται αρνητικά.

3.2.6. Ανακεφαλαίωση

Κατά την γνωριμία μας με διάφορα στοιχεία του Directed Diffusion, σιωπηρά περιγράφηκε μια συγκεκριμένη χρήση (usage). Δεν είναι άλλη από χρήση των gradient ως τρόπου άντλησης των δεδομένων. Το ίδιο το παράδειγμα του Directed Diffusion δεν περιορίζει το σχεδιαστή σε αυτήν την συγκεκριμένη επιλογή. Άλλες επιλογές είναι επίσης δυνατές. Όπως αυτή στην οποία οι κόμβοι

μπορούν να διαδώσουν τα δεδομένα ακόμα και ελλείψει ενδιαφερόντων. Τα gradient δημιουργούνται καθώς λαμβάνουν χώρα τα παραπάνω. Αυτό είναι χρήσιμο για την αυθόρμητη διάδοση ένας σημαντικού γεγονότος σε κάποιο τμήμα του πεδίου των αισθητήρων. Ένας κόμβος αισθητήρων μπορεί να το χρησιμοποιήσει για να προειδοποιήσει άλλους κόμβους αισθητήρων για την επικείμενη δραστηριότητα. Επιπλέον, περισσότερες σχεδιαστικές επιλογές για κάθε στοιχείο της διάχυσης είναι δυνατές (πίνακας 3.2.2).

Η περιγραφή έως τώρα επισημαίνει τα κύρια χαρακτηριστικά της διάχυσης και πώς διαφέρει από την παραδοσιακή δικτύωση. Κατ' αρχάς, η διάχυση είναι δεδομενο-κεντρική. Όλη η επικοινωνία σε ένα βασισμένο στη διάχυση δίκτυο αισθητήρων χρησιμοποιεί τα ενδιαφέροντα. Αυτά με τη σειρά τους διευκρινίζουν ονοματοποιημένα στοιχεία. Δεύτερον, όλη η επικοινωνία στη διάχυση είναι από γείτονα σε γείτονα. Αντίθετα από την end-to-end επικοινωνία στα παραδοσιακά δίκτυα δεδομένων. Με άλλα λόγια, κάθε κόμβος είναι ένα «τέλος» / προορισμός σε ένα δίκτυο αισθητήρων. Αφού δεν υπάρχει κανένας «δρομολογητής» στο δίκτυο αισθητήρων. Κάθε κόμβος αισθητήρων μπορεί να ερμηνεύει τα μηνύματα των δεδομένων και των ενδιαφερόντων. Αυτή η σχεδιαστική επιλογή δικαιολογείται από τους ιδιόμορφους στόχους των δικτύων αισθητήρων. Τα δίκτυα αισθητήρων δεν είναι γενικής χρήσης δίκτυα επικοινωνίας. Τρίτον, οι κόμβοι αισθητήρων δεν πρέπει να έχουν παγκόσμια μοναδικά προσδιοριστικά/ταυτότητες ή μοναδικές διευθύνσεις. Οι κόμβοι χρειάζεται να διακρίνονται μεταξύ των γειτόνων τους. Τέλος, σε ένα IP-βασισμένο δίκτυο αισθητήρων η συλλογή και η επεξεργασία δεδομένων από τους κόμβους μπορούν να εκτελεστούν από εξειδικευμένους κεντρικούς υπολογιστές (server). Αυτοί βρίσκονται σε απόσταση και δεν επιβαρύνονται με τη διαδικασία της αίσθησης των φαινομένων. Στο δίκτυο αισθητήρων, επειδή κάθε κόμβος μπορεί να αποθηκεύσει και να επεξεργαστεί τα μηνύματα της διαδικασίας, είναι γενικά επιθυμητό να μπορεί να εκτελεστεί συντονισμένη αίσθηση κοντά στα παρακολουθόντα φαινόμενα. Η διάχυση συσχετίζεται σαφώς με τους παραδοσιακούς αλγόριθμους δρομολόγησης δικτύων. Υπό κάποια έννοια, είναι μια διαδραστική τεχνική δρομολόγησης, δεδομένου ότι «οι διαδρομές» δημιουργούνται κατόπιν παραγγελίας. Όμως, διαφέρει από άλλες ad hoc διαδραστικές τεχνικές δρομολόγησης σε πολλά σημεία. Κατ' αρχάς, καμία προσπάθεια δεν γίνεται για να βρεθεί μονοπάτι μεταξύ πηγής και καταβόθρας που να μην περιέχει βρόχο. Όλα αυτά πριν αρχίσει η μετάδοση δεδομένων. Αντ' αυτού, η περιορισμένη ή κατευθυνόμενη πλημμύρα (flooding) χρησιμοποιείται για την οργάνωση πολλαπλών μονοπατιών. Τα δεδομένα των μηνυμάτων στέλνονται αρχικά κατά μήκος αυτών των μονοπατιών. Δεύτερον, σύντομα η τεχνική ενίσχυσης προσπαθούν να μειώσουν την πολλαπλότητα των διαδρομών σε μικρότερο αριθμό. Η μείωση βασίζεται στην εμπειρικά παρατηρηθείσα απόδοση των μονοπατιών. Τέλος, μια cache μηνυμάτων χρησιμοποιείται για την αποφυγή δημιουργίας βρόχων. Τα ενδιαφέροντα και οι gradients δημιουργούν μηχανισμούς οργάνωσης που δεν εγγυώνται μονοπάτια δίχως βρόχους μεταξύ πηγής και καταβόθρας.

Γιατί αυτή η ιδιαίτερη επιλογή σχεδίασμου;

Συνειδητά επιλέχτηκε να ερευνηθούν αλγόριθμοι οργάνωσης μονοπατιών, οι οποίοι καθιερώνουν τα μονοπάτια των δικτύων χρησιμοποιώντας αυστηρά τοπική επικοινωνία (γείτονα σε γείτονα). Πίσω από αυτήν την επιλογή κρύβεται η παρατήρηση ότι φυσικά συστήματα (π.χ., αποικίες μυρμηγκιών) ενισχύουν τα μονοπάτια μετάδοσης χρησιμοποιώντας τέτοιας κλίμακας επικοινωνίες. Η επικοινωνία αυτή κρίνεται καλή και είναι εξαιρετικά ανθεκτική. Εντούτοις η αυστηρά τοπική επικοινωνία υπονοεί ότι η οργάνωση διαδρομών δεν χρησιμοποιεί γενικές (global) μετρικές για τις τοπολογίες. Η τοπική επικοινωνία δηλώνει ότι ένας κόμβος ξέρει, ότι τα στοιχεία που έλαβε από έναν γείτονα του προήλθαν από εκείνο το γείτονα και μόνο. Αυτό μπορεί να είναι ενέργεια αποδοτικό ιδιαίτερα στα

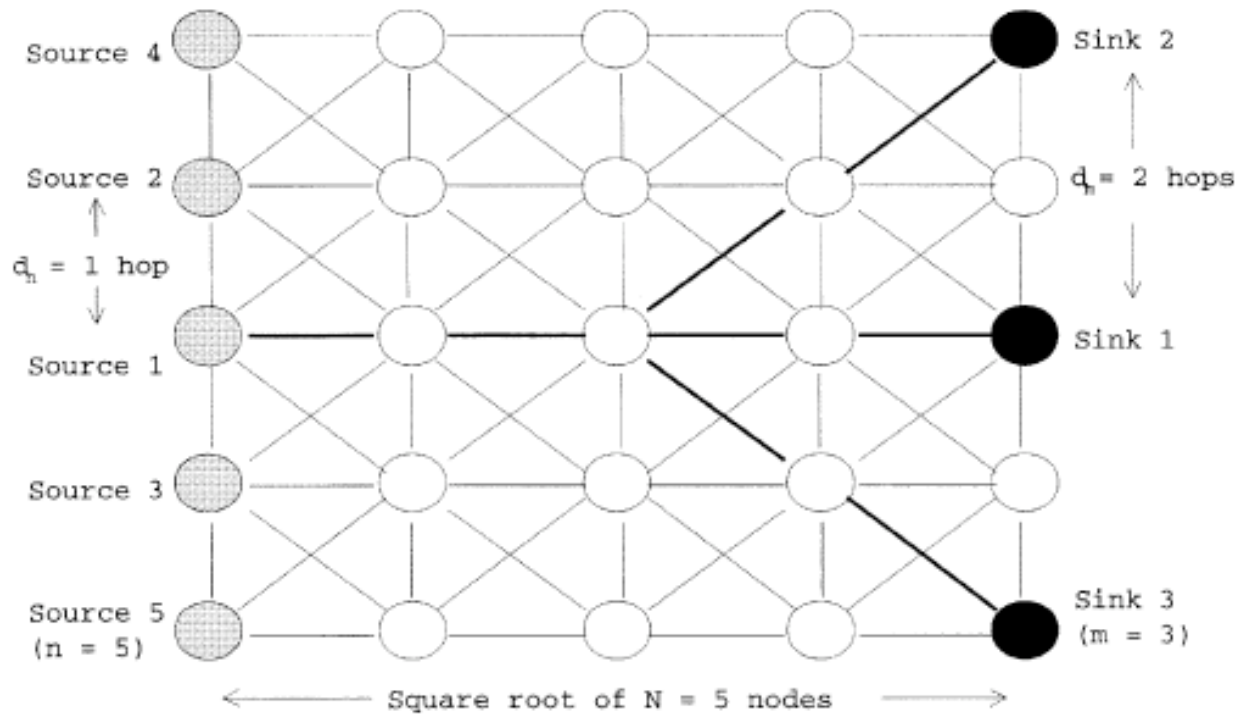
δυναμικά δίκτυα. Εδώ δεν χρειάζεται να διαδίδονται οι αλλαγές στην τοπολογία μέσω του δικτύου. Φυσικά, τα προκύπτον μονοπάτια επικοινωνίας είναι ημι-βέλτιστα. Στον αντίποδα η ενεργειακή ανεπάρκεια λόγω των ημι-βέλτιστων μονοπατιών αντιμετωπίζεται από προσεκτικά σχεδιασμένες τεχνικές ενδοδικτυακάκης επεξεργασίας. Καταλήγοντας θεωρούμε ότι αυτή η προσέγγιση ανταλλάσσει κάποια ενεργειακή αποδοτικότητα, κερδίζοντας όμως ευρωστία και δυνατότητα κλιμάκωσης.

Τέλος, ενδεχομένως να θεωρεί κανείς ότι το συγκεκριμένο στιγμιότυπο που επιλέγει (εντοπισμός θέσης) έχει περιορισμένες δυνατότητες εφαρμογής. Όμως αναδεικνύει πολλά από τα ουσιαστικά χαρακτηριστικά γνωρίσματα μιας μεγάλης κατηγορίας δικτύων αισθητήρων, που ασχολούνται με τη απομακρυσμένη επιτήρηση. Υπογραμμίζεται ότι χρειάζονται πειράματα και αξιολόγηση των διάφορων μηχανισμών. Παρά το ότι έχει συζητηθεί το δίκτυο εντοπισμού με αρκετές λεπτομέρειες. Είναι απαραίτητα και τα δύο ώστε να γίνουν πλήρως αντιληπτές η ευρωστία, η κλιμάκωση, και οι επιπτώσεις στην απόδοση της διάχυσης καθώς και μερικών μηχανισμών της πιο συγκεκριμένα. Στη συνέχεια θα αναδειχθούν αυτές οι δύο κατευθύνσεις.

3.3. Αναλυτική Αξιολόγηση

Σε αυτό το τμήμα παρουσιάζεται μια αναλυτική αξιολόγηση του κόστους της παράδοσης των δεδομένων για την κατευθυνόμενη διάχυση. Επίσης παρουσιάζονται και δύο εξιδανικευμένα σχήματα: omniscient multicast και η πλημμύρα (flooding). Αυτή η ανάλυση εξυπηρετεί στον ορθολογικό έλεγχο της κατευθυνόμενης διάχυσης, πέρα από τη διαίσθητική αντίληψη της. Επίσης δίνεται έμφαση σε μερικές από τις μεταξύ τους διαφορές.

Η ανάλυση αυτών των τριών σχημάτων γίνεται σε μια απλή εξιδανικευμένη ρύθμιση (setting). Υποθέτουμε ένα τετραγωνικό πλέγμα που αποτελείται από N κόμβους. Σε αυτό το πλέγμα, το εύρος μετάδοσης των κόμβων είναι έτσι, ώστε κάθε κόμβος να μπορεί να επικοινωνήσει με ακριβώς οκτώ γειτονικούς κόμβους στο πλέγμα. Το Σχ.3.3 παρουσιάζει τις συνδέσεις μεταξύ των ζευγαριών των κόμβων, που μπορούν να επικοινωνήσουν μεταξύ τους. Όλες οι n πηγές τοποθετούνται κατά μήκος των κόμβων της αριστερής άκρης του πλέγματος, ενώ όλες οι καταβόθρες κατά μήκος της δεξιάς άκρης. Η πρώτη πηγή είναι στο κέντρο του αριστερού συνόρου. Η i -οστή πηγή είναι $d_{ni/2}$ hops πάνω (εάν το i είναι ζυγός) ή κάτω (αν το i είναι περιττός) από την πρώτη πηγή. Αυτό το σχέδιο τοποθέτησης χρησιμοποιείται και στις καταβόθρες. Η διαφορά έγγυται στο ότι η απόσταση ανάμεσα σε δύο γειτονικές καταβόθρες είναι d_m hops και όχι d_n .



Σχήμα 3.3: Παράδειγμα τεταγωνικής grid τοπολογίας

3.3.1. Flooding (Πλημμύρα)

Στο σχήμα/τεχνική της πλημμύρας, οι πηγές διαχέουν όλα τα γεγονότα σε κάθε κόμβο του δικτύου. Η τεχνική της πλημμύρας αποτελεί ορόσημο για την κατευθυνόμενη διάχυση. Εάν η τελευταία δεν αποδίδει καλύτερα από την πλημμύρα, τότε παύει να θεωρείται βιώσιμη για τα δίκτυα αισθητήρων.

Σε αυτήν την αναλυτική αξιολόγηση, ως μετρική απόδοσης λογίζεται το συνολικό κόστος της μετάδοσης και της λήψης ενός γεγονότος από κάθε πηγή προς όλες τις καταβόθρες. Το κόστος καθορίζεται από μια μονάδα μετάδοσης μηνυμάτων και μια μονάδα υποδοχής μηνυμάτων. Αυτές οι υποθέσεις είναι σαφώς εξιδανικευμένες σε δύο σημεία. Τα κόστη μετάδοσης και λήψης δεν είναι παρόμοια και ενδεχομένως υπάρξουν και άλλες μετρικές για τα ενδιαφέροντα.

Από αυτή τη μετρική, το κόστος της πλημμύρας που ορίζεται ως $C_f(N, n, m, d_n, d_m)$ ή απλούστερα C_f δίνεται από τη σχέση:

$$\begin{aligned}
 C_f &= nN + 2n \left(2(\sqrt{N} - 1)\sqrt{N} + 2(\sqrt{N} - 1)^2 \right) \\
 &= nN + 4n(\sqrt{N} - 1)(2\sqrt{N} - 1). \tag{1}
 \end{aligned}$$

Σχέση 1

Το κόστος μετάδοσης για την πλημμύρα n γεγονότων (ένα γεγονός από κάθε πηγή) είναι nN . Αυτό επειδή κάθε κόμβος στέλνει μόνο μια MAC μετάδοση ανά γεγονός. Ακόμα κάθε κόμβος μπορεί να

λάβει το ίδιο γεγονός από όλους τους γείτονες. Κατά συνέπεια, το κόστος λήψης για εκείνα τα γεγονότα καθορίζεται ως $2n$ φορές του αριθμού των συνδέσεων στο δίκτυο $(4n\sqrt{N} - 1)(2\sqrt{N} - 1)$. Το κόστος παράδοσης δεδομένων στη πλημμύρα είναι $O(nN)$, πράγμα ασυμπτωτικά υψηλότερο από το κόστος των άλλων σχημάτων/τεχνικών.

3.3.2. Omniscient Multicast

Εδώ κάθε πηγή εκπέμπει τα γεγονότα κατά μήκος της συντομότερης διαδρομής ενός multicast δέντρου προς τις καταβόθρες. Στην ανάλυση καθώς επίσης και στις προσομοιώσεις που ακολουθούν δεν προσμετράται το κόστος της κατασκευής των δέντρων. Το Omniscient Multicast αναδεικνύει την καλύτερη δυνατή απόδοση, που είναι επιτεύξιμη σε ένα IP-βασισμένο δίκτυο αισθητήρων. Βέβαια χωρίς να λαμβάνει υπό όψη τις επιβαρύνσεις (overhead). Χρησιμοποιείται αυτό το σχήμα για να αποκτηθεί κάποια διαίσθηση, του πώς η επιλογή της ενδοδικτυακής επεξεργασίας επηρεάζει την απόδοση.

Για το Omniscient Multicast το κόστος παράδοσης δεδομένων καθορίζεται από δύο φορές τον αριθμό των συνδέσεων των δέντρων με τα συντομότερα μονοπάτια. Ακόμη και σε αυτήν την απλή τοπολογία πλέγματος, υπάρχουν διάφορα συντομότερα μονοπάτια για κάθε ζευγάρι πηγής - καταβόθρας. Επιλέγουμε την μικρότερη διαδρομή χρησιμοποιώντας τον ακόλουθο απλό αιτιοκρατικό κανόνα.

Από μία καταβόθρα προς μια πηγή, μια διαγώνια σύνδεση είναι πάντα ο επόμενος σύνδεσμος (hop) εφ' όσον οδηγεί σε συντομότερα μονοπάτια. Διαφορετικά επιλέγεται μια οριζόντια σύνδεση. Αυτός ο κανόνας επιλογής μονοπατιού επαναλαμβάνεται έως ότου φτάσουμε στην πηγή. Κατά συνέπεια καμία συντομότερη διαδρομή δεν περιλαμβάνει κάθετες συνδέσεις.

Παραδείγματος χάριν, εάν συμβολίσουμε το συντομότερο μονοπάτι του δέντρου που η ρίζα του βρίσκεται στη πηγή j ως T_j , τότε ο αριθμός των συνδέσεων στο T_1 έχει δύο στοιχεία:

Τον αριθμό των οριζόντιων συνδέσεων $(\sqrt{N} - 1)$ και των διαγώνιων συνδέσεων $(d_{lm/2l}(l_{m/2l} + 1) - d_{lm/2l}((m - 1) \bmod 2))$. Άλλες επιλογές θα οδηγούσαν σε διαφορετικό κόστος, δεδομένου ότι ο αριθμός των κοινών συνδέσεων στο δέντρο είναι πλέον διαφορετικός.

Το κόστος του omniscient multicast C_0 λογίζεται ως το άθροισμα του κόστους n δέντρων. Το καθένα έχει τη ρίζα του σε κάθε πηγή. Εάν με το $C(T_j)$ υποδηλώσουμε το κόστος της μετάδοσης ενός γεγονότος από την πηγή j , τότε προκύπτει ότι μπορούμε να εκφράσουμε αυτό το κόστος συναρτήση του T_1 . Δηλαδή θα είναι $C(T_j) = C(T_1) + C(T_j - T_1) - C(T_1 - T_j)$. Με $C(T_j - T_k)$ δηλώνουμε το κόστος εκπομπής και λήψης κατά μήκος του δέντρου που διαμορφώνεται με την αφαίρεση από το T_j , εκείνων των συνδέσεων που είναι κοινές και με το T_k . Επιπλέον ως $C(T_j)$ μπορεί να εκφραστεί ως άθροισμα δυο κοστών: του κόστους της μετάδοσης και της λήψης κατά μήκος των οριζόντιων συνδέσεων $H(T_j)$ και του ανάλογου κόστους $D(T_j)$, κατά μήκος των διαγώνιων συνδέσεων.

Μπορούμε να γράψουμε C_0 ως

$$C_o = \sum_{j=1}^n \left\{ D(T_1) + H(T_j) + D(T_j - T_1) - D(T_1 - T_j) \right\} \quad (2)$$

where

$$H(T_j) = 2 \left\{ \sqrt{N} - 1 - \left(\left\lfloor \frac{j}{2} \right\rfloor d_n - \min \left(\left\lfloor \left\lfloor \frac{j}{2} \right\rfloor \frac{d_n}{d_m} \right\rfloor, \left\lfloor \frac{m - (j \bmod 2)}{2} \right\rfloor \right) d_m \right) \right\} \quad (3)$$

$$D(T_j - T_1) = 2 \left\{ \left\lceil \frac{m + (j \bmod 2)}{2} \right\rceil \left\lfloor \frac{j}{2} \right\rfloor d_n + \sum_{l=1}^{\min(\lfloor \lfloor j/2 \rfloor d_n / d_m \rfloor, \lfloor m - (j \bmod 2) / 2 \rfloor)} \left(d_n \left\lfloor \frac{j}{2} \right\rfloor - l d_m \right) \right\} \quad (4)$$

$$D(T_1 - T_j) = 2 \left\{ \sum_{l=1}^{\lfloor m - (j \bmod 2) / 2 \rfloor} \min \left(d_n \left\lfloor \frac{j}{2} \right\rfloor, l d_m \right) \right\}. \quad (5)$$

Σχέσεις 2, 3, 4, 5

Ασυπτωτικά το κόστος μετάδοσης των δεδομένων στο omniscient multicast C_0 είναι $O(n\sqrt{N})$ για $m \ll \sqrt{N}$

3.3.3. Directed Diffusion

Η ανάλυση της διάχυσης προχωρά σύμφωνα με τις ίδιες άρχες του omniscient multicast. Για να απλοποιηθεί η ανάλυση, υποθέτουμε ότι το δέντρο που τοπικοί αλγόριθμοι της διάχυσης κατασκευάζουν, είναι η «ένωση» (union) του δέντρου του συντομότερου μονοπατιού με ρίζα σε κάθε πηγή. Αυτή η υπόθεση ισχύει σε μεγάλο βαθμό όταν το δίκτυο λειτουργεί σε επίπεδα χαμηλού φόρτου δεδομένων. Επιπλέον, από τα πλήστα διαθέσιμα συντομότερα μονοπάτια, η διάχυση επιλέγει ένα σύμφωνα με τον ακόλουθο κανόνα: Από μία καταβόθρα σε μια πηγή, μια διαγώνια σύνδεση είναι πάντα ο επόμενος σύνδεσμος (hop) εφ' όσον είναι κατά μήκος της συντομότερης διαδρομής προς την πηγή. Διαφορετικά, μια οριζόντια σύνδεση επιλέγεται. Αυτός ο κανόνας είναι ο ίδιος που χρησιμοποιήσαμε και στο προηγούμενο πρωτόκολλο.

Παρά τη χρησιμοποίηση της ίδιας τεχνικής για την επιλογή του μονοπατιού, το κόστος της διάχυσης C_d διαφέρει από αυτο του omniscient multicast. Πρωτίστως λόγω της σε επίπεδο εφαρμογής επεξεργασίας δεδομένων. Συγκεκριμένα, εάν όλες οι πηγές στέλνουν ίδιες εκτιμήσεις για τις θέσεις των στόχων, με τη διάχυση μπορούμε να εξαλείψουμε τις κόπιες. Το κόστος για την παράδοση των δεδομένων της διάχυσης είναι δύο φορές ο αριθμός των συνδέσεων, των ενώσεων (unions) όλων των συντομότερων μονοπατιών των δέντρων με ρίζες στην πηγή. Ως εκ τούτου

$$\begin{aligned}
 C_d &= C(UT_{1 \rightarrow n}) \\
 &= C(T_1) \\
 &\quad + \sum_{j=2}^n \left\{ H(T_j - UT_{1 \rightarrow (j-1)}) + D(T_j - UT_{1 \rightarrow (j-1)}) \right\}
 \end{aligned} \tag{6}$$

where

$$H(T_j - UT_{1 \rightarrow (j-1)}) = H(T_j) \tag{7}$$

$$\begin{aligned}
 D(T_j - UT_{1 \rightarrow (j-1)}) &= 2 \left\{ \left\lceil \frac{m + (j \bmod 2)}{2} \right\rceil d_n \right. \\
 &\quad \left. + \sum_{l=1}^{\min(\lfloor j/2 \rfloor d_n / d_m, \lfloor m - (j \bmod 2) / 2 \rfloor)} \min \left(d_n, d_n \left\lfloor \frac{j}{2} \right\rfloor - l d_m \right) \right\}.
 \end{aligned} \tag{8}$$

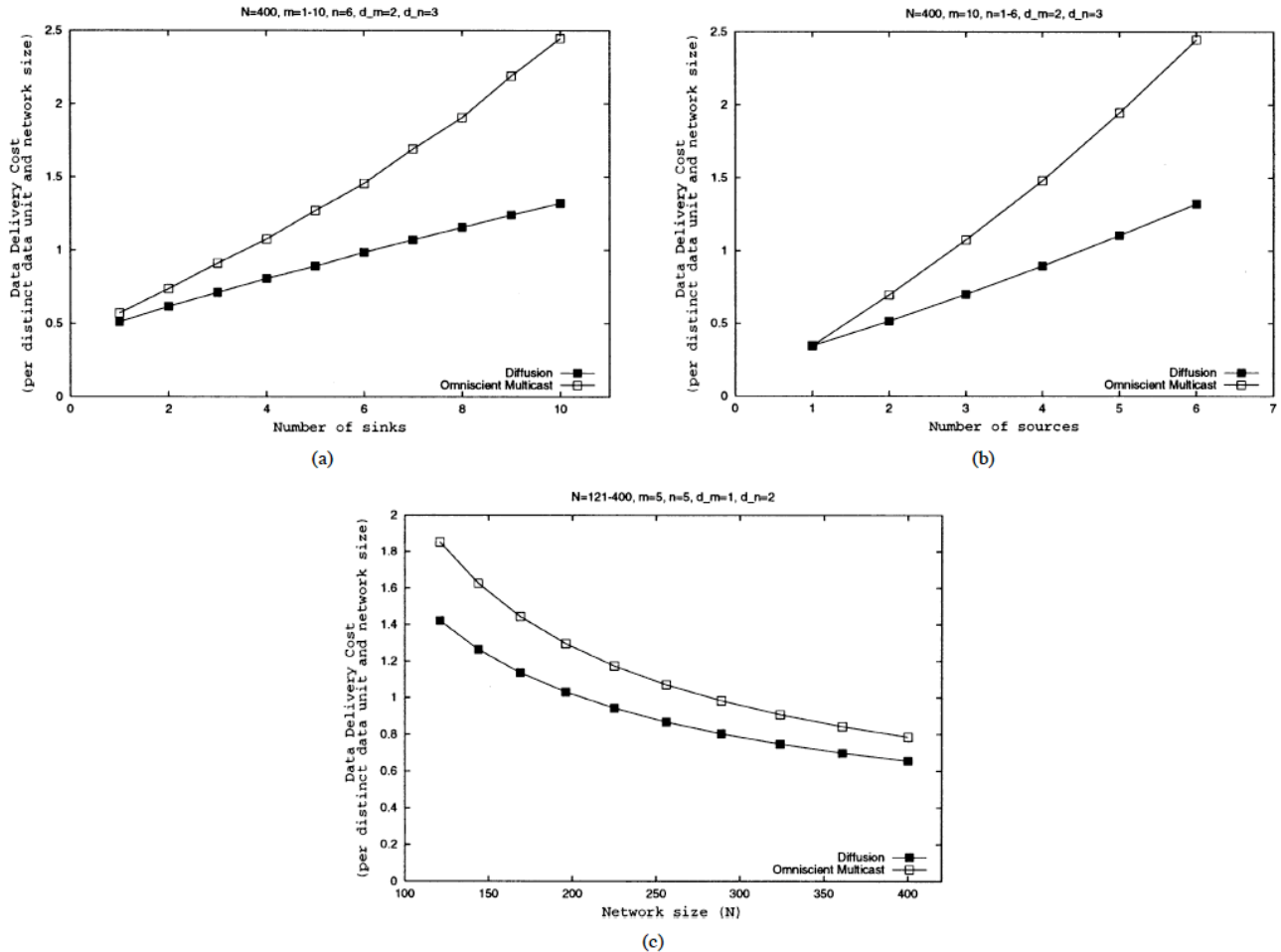
Σχέσεις 6, 7, 8

Παρομοίως με το C_0 , Το C_d είναι $O(n\sqrt{N})$ για $m \ll \sqrt{N}$

3.3.4. Σύγκριση

Το κόστος παράδοσης των δεδομένων της πλημμύρας C_f είναι ακρετές τάξεις μεγέθους μεγαλύτερο από αυτό του omniscient multicast C_0 . Όμως το C_0 παραμένει υψηλότερο από το κόστος διάχυσης C_d . Αυτά επειδή $D(T_1) - D(T_1 - T_j) \geq 0$ κ $D(T_j - T_1) \geq D(T_j - UT_{1 \rightarrow (j-1)})$. Για την επικύρωση του συλλογισμού, το κόστος παράδοσης δεδομένων (που τυποποιείται από το μέγεθος δικτύων) για το directed diffusion και το omniscient multicast συμβολίζεται χρησιμοποιώντας διάφορες παραμέτρους (N, m, n). Καθώς ο αριθμός των πηγών και των καταβοθρών αυξάνεται [Σχ.3.3.4(a) και (b)], η μείωση του κόστους λόγω της ενδοδικτυακής επεξεργασίας (π.χ. διαχείριση στις κόπιες) της διάχυσης γίνεται εμφανέστατη. Λαμβάνοντας υπόψη ότι C_d αυξάνεται με χαμηλότερο ρυθμό από το C_0 .

Ιδιαίτερο καθίσταται το ενδιαφέρον της αποτύπωσης του κόστους σε αντιδιαστολή με το μέγεθος του δικτύου [Σχ.3.3.4 (c)]. Δεδομένου ότι η διάχυση μπορεί να εμποδίσει αντίγραφο σε επίπεδο εφαρμογής, θα αναμέναμε το C_0 όχι παραπάνω από nC_d . Ο κύριος λόγος που αυτό δεν συμβαίνει είναι ότι στη ανάλυσή υπολογίζεται κάπως συντηρητικά το κόστος διάχυσης. Στην πράξη η διάχυση θα είχε προβεί σε αρνητική ενίσχυση αρκετών συνδέσεων, που περιλαμβάνει η ανάλυση.



Σχήμα 3.3.4: Αντίκτυπος διάφορων παραμέτρων του DD και omniscient multicast. (a) Αντίκτυπος του αριθμού των sinks. (b) Αντίκτυπος τους αριθμού των πηγών. (c) Αντίκτυπος του μεγεθους του δικτύου

3.4. Εξομοίωση

3.4.1. Στόχοι, Μετρικές και Μεθοδολογία

Ο εντοπισμός οχημάτων στην κατευθυνόμενη διάχυση έλαβε χώρα στον προσομοιωτή NS-2. Οι στόχοι που οδήγησαν στην πραγματοποίηση της μελέτης αξιολόγησης ήταν τετραπλοί: 1) έλεγχος και συμπλήρωση της αναλυτικής αξιολόγησή 2) διερεύνηση αντίκτυπου δυναμικών φαινομένων (dynamics), όπως η αποτυχία του κόμβου κατά τη διάχυση 3) διερεύνηση της επιρροής του MAC

στην απόδοση της διάχυσης 4) μελέτης της απόδοσης του directed diffusion ανάλογα με την επιλογή των παραμέτρων.

Επιλέγονται τρεις μετρικές για να αναλυθεί η απόδοση της κατευθυνόμενης διάχυσης και για να συγκριθεί με τα άλλα πρωτόκολλα: ο μέσος όρος κατανάλωσης της ενέργειας, η μέση καθυστέρηση και ο ρυθμός παράδοσης ενός διακριτού γεγονότος.

Ο μέσος όρος κατανάλωσης ενέργειας μέτρα το ρυθμό της συνολικής κατανάλωσης ενέργειας ανά κόμβο στο δίκτυο. Σε σχέση με τον αριθμό διακριτών γεγονότων στις καταβόθρες. Αυτή η μετρική υπολογίζει τη μέση εργασία που γίνεται από έναν κόμβο για την παράδοση χρήσιμων πληροφοριών στις καταβόθρες. Η μετρική δείχνει επίσης τη συνολική διάρκεια ζωής των κόμβων αισθητήρων. Η μέση καθυστέρηση μετρά τη μέση μονομερής (one-way) καθυστέρηση μεταξύ της εκπομπής ενός γεγονότος και της λήψης του από μια καταβόθρα. Αυτή η μετρική καθορίζει την προσωρινή ακρίβεια των εκτιμήσεων για τις τοποθεσίες (locations) που διακινούνται από το δίκτυο αισθητήρων.

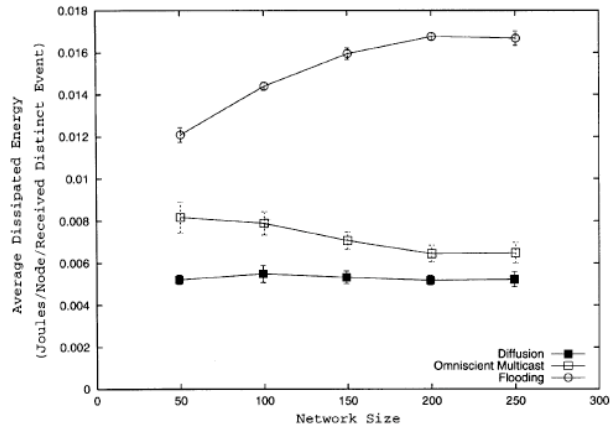
Ο ρυθμός παράδοσης διακριτών γεγονότων περιγράφει την αναλογία του αριθμού ευδιάκριτων γεγονότων, που λαμβάνονται σε σχέση με τον αριθμό που στέλνεται αρχικά. Παρόμοιες μετρικές χρησιμοποιήθηκαν και σε προηγούμενες εργασίες για τα ad hoc σχήματα δρομολόγησης.

Προκειμένου να μελετηθεί η απόδοση της διάχυσης συναρτήση του μεγέθους του δικτύου θα υπάρξουν ποικίλα σενάρια για τους κόμβους. Τόσο σε επίπεδο πεδίου όσο και μεγέθους. Σε κάθε ένα από τα πειράματά μελετώνται πέντε διαφορετικοί πεδία αισθητήρων, με εύρος 50 έως 250 (το βήμα είναι 50 κόμβοι). Το πεδίο των 50 κόμβων κατανέμει με τυχαίο τρόπο τους κόμβους σε ένα 160 επί 160 m τετράγωνο. Κάθε κόμβος έχει ραδιο-εύρος 40 μ. Τα άλλα μεγέθη παράγονται με την κλιμάκωση του τετραγώνου και την σταθερή διατήρηση του ραδιο-εύρους. Όλα αυτά προκειμένου να κρατηθεί σταθερή η μέση πυκνότητα των κόμβων-αισθητήρων. Το παραπάνω γίνεται επειδή η μακροσκοπική συνδεσιμότητα (connectivity) ενός πεδίου αισθητήρων βρίσκεται σε συνάρτηση με τη μέση πυκνότητα. Εάν κρατηθεί σταθερή η περιοχή του πεδίου των αισθητήρων αλλά αυξηθεί το μέγεθος του δικτύου θα παρατηρηθούν επιπτώσεις στην απόδοση. Αυτές δεν οφείλονται μόνο στο μεγαλύτερο αριθμό κόμβων αλλά και στην αυξημένη συνδεσιμότητα. Ακόμα θα μελετηθεί και το αντίκτυπο του μεγέθους των δικτύων και σε μερικούς από τους μηχανισμούς που λαμβάνουν χώρα.

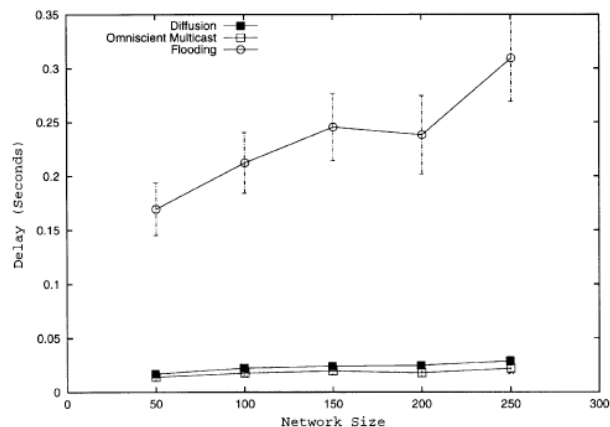
Ο εξομοιωτής NS-2 διαθέτει ένα 1.6 Mb/s 802.11 MAC επίπεδο. Οι εξομοιώσεις χρησιμοποιούν ένα τροποποιημένο 802.11 MAC επίπεδο. Για να προσεγγισθούν περισσότερο ρεαλιστικά οι ραδιοσυχνότητες των δικτύων αισθητήρων, αλλάχθηκε το ραδιοενεργειακό μοντέλο του NS-2. Έτσι στην νεκρή (idle) περίοδο η κατανάλωση ενέργειας φτάνει περίπου τα 35 mW ή σχεδόν το 10% της ενέργειας που αφορά τη λήψη (395 mW) και περίπου το 5% της καταναλισκόμενης ενέργειας μετάδοσης (660 mW). Αυτό το MAC επίπεδο δεν είναι απολύτως ικανοποιητικό, δεδομένου ότι η ενεργειακή αποδοτικότητα οδηγεί στην επιλογή του TDMA-style MAC για τα δίκτυα αισθητήρων. Πολύ περισσότερο από τα πρωτόκολλα που βασίζονται στις συνδέσεις. Εν συντομία, αυτοί οι λόγοι έχουν να κάνουν με την ενέργεια, που καταναλώνεται κατά τις νεκρές περιόδους. Με ένα TDMA-style MAC δίνετε να τεθούν οι ραδιοεκπομπές σε αναμονή σε αυτές τις περιόδους. Σε αντίθεση το 802.11 καταναλώνει τα ίδια ποσά ενέργειας όταν είναι idle όπως και όταν λαμβάνει μεταδόσεις. Αργότερα αναλύεται ο αντίκτυπος στο ενεργειακό μοντέλο του MAC, τόσο όταν οι κόμβοι αφογκράζονται τις μεταδόσεις όσο και όταν τις λαμβάνουν.

Τέλος, οι περισσότερες από τις προσομοιώσεις χρησιμοποιούν έναν σταθερό φόρτο εργασίας που προέρχεται από πέντε πηγές και πέντε καταβόθρες. Όλες οι πηγές επιλέγονται τυχαία από τους κόμβους σε ένα 70 επί 70 μ τετράγωνο από την κάτω αριστερή γωνία του πεδίου αισθητήρων. Οι καταβόθρες είναι ομοιόμορφα διεσπαρμένες κατά μήκος του πεδίου των αισθητήρων. Κάθε πηγή παράγει δύο γεγονότα ανά δευτερόλεπτο. Ο ρυθμός για τα διερευνητικά γεγονότα επιλέχτηκε να είναι, ένα γεγονός κάθε 50 δευτερόλεπτα. Τα γεγονότα διαμορφώθηκαν ως πακέτα των 64

ψηφιολέξεων και τα ενδιαφέροντα 36 ψηφιολέξεων. Τα ενδιαφέροντα παρήχθησαν περιοδικά κάθε 5 s και η διάρκεια ενδιαφέροντος ήταν 15 s. Το παράθυρο για την αρνητική ενίσχυση επιλέγει στα 2 s. Αυτές οι επιλογές παραμέτρων διαμορφώθηκαν από το ιδιαίτερο δίκτυο αισθητήρων που βρίσκεται υπό εξέταση (μικρές περιγραφές γεγονότων, πηγές μέσα σε μια γεωγραφική περιοχή). Ρόλο στη διαμόρφωση έπαιξε και η επιθυμία για απλοποίηση της κατανόησης των αποτελεσμάτων. Τα σημεία των δεδομένων σε κάθε γραφική παράσταση αντιπροσωπεύουν δέκα σενάρια με 95% διαστήματα (intervals) εμπιστοσύνης.



(a)



(b)

Σχήμα 3.4.1: Σύγκριση του DD με flooding και omniscient multicast. (a) Μέση καταναλισκόμενη ενέργεια. (b) Μέση καθυστέρηση.

3.4.2. Συγκριτική Αξιολόγηση

Το πρώτο πείραμα συγκρίνει τη διάχυση με το omniscient multicast και το flooding ως προς τη διάδοση των δεδομένων στα δίκτυα. Το Σχ.3.4.1(a) παρουσιάζει μέση κατανάλωση ενέργειας ανά πακέτο συναρτήσει του μεγέθους των δικτύων. Το omniscient multicast καταναλώνει λίγο λιγότερο από το ήμισυ της ενέργειας του flooding ανά πακέτο (ανά κόμβο). Επιτυγχάνει αυτήν την ενεργειακή αποδοτικότητα, από την μεταφορά των γεγονότων κατά της μήκος ενιαίας διαδρομής από κάθε πηγή σε κάθε καταβόθρα. Το Directed Diffusion έχει εμφανώς καλύτερη ενεργειακή αποδοτικότητα από

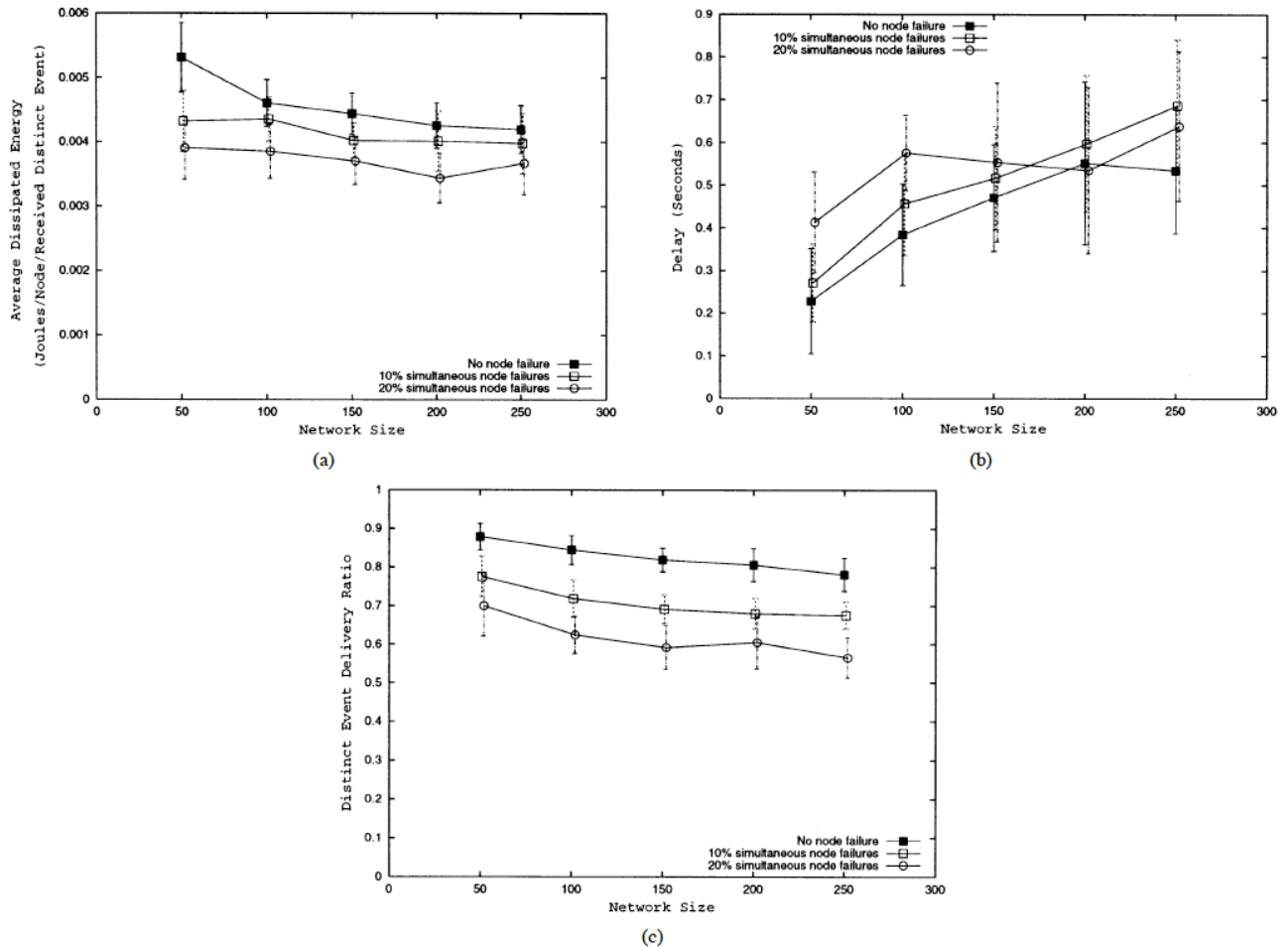
το omniscient multicast. Για μερικά πεδία αισθητήρων, η καταναλισκόμενη ενέργειά είναι μόνο 60% από αυτή του omniscient multicast. Όπως με το omniscient multicast, το Directed Diffusion επιτυγχάνει χαμηλή ενεργειακή κατανάλωση. Αυτό επιτυγχάνεται με τη μείωση του αριθμού των μονοπατιών. Επιπλέον, η διάχυση επωφελείται σημαντικά από την ενδοδικτυακή επεξεργασία.

Στα πειράματα οι πηγές μεταφέρουν ίδιες εκτιμήσεις θέσεων και οι ενδιάμεσοι κόμβοι απαλοφούν τις διπλές εκτιμήσεις για αυτές τις θέσεις. Πράγμα που αντιστοιχεί στην περίπτωση, όπου υπάρχει ένα όχημα στην υπό μελέτη περιοχή.

Τότε δεδομένου ότι υπάρχουν πέντε πηγές, γιατί η διάχυση (με την αρνητική ενίσχυση) στον τομέα της ενέργεια δεν είναι σχεδόν πέντε φορές αποδοτικότερη από το omniscient multicast; Πρώτον, και τα δύο πρωτόκολλα χαρακτηρίζονται από παρόμοιες ενεργειακές απαιτήσεις στον τομέα της ακρόασης (listening) των μεταδόσεων. Δεύτερον, οι επιλογές ενίσχυσης και αρνητικής ενίσχυσης στο Directed Diffusion οδηγούν συχνά τα υψηλής ποιότητας δεδομένα κατά μήκος πολλαπλών μονοπατιών. Με αυτόν τον τρόπο χρησιμοποιείται περισσότερη ενέργεια. Συγκεκριμένα, ο κανόνας ενίσχυσης είναι πολύ επιθετικός. Αφού ενισχύει οποιοδήποτε γείτονα έχει προηγουμένως στείλει ένα πρωτοεμφανιζόμενο γεγονός. Αντίθετα, ο κανόνας αρνητικής ενίσχυσης είναι πολύ συντηρητικός. Αφού ενισχύει αρνητικά τους γείτονες, που συνεχώς στέλνουν διπλο-εμφανιζόμενα (δηλ. Έχουν ξανά ειδοθεί) γεγονότα.

Το Σχ.3.4.1 (b) παριστάνει τη μέση καθυστέρηση συναρτήση του μεγέθους του δικτύου. Το Directed Diffusion έχει καθυστέρηση συγκρίσιμη με το omniscient multicast. Αυτό είναι ενθαρρυντικό. Σε ένα δίκτυο αισθητήρων χωρίς συμφόρηση ή οποιοδήποτε παρεμπόδιση, η συντομότερη διαδρομή οδηγεί και στη χαμηλότερη καθυστέρηση. Κατά συνέπεια, οι κανόνες ενίσχυσης φαίνεται να εντοπίζουν τις διαδρομές με τη χαμηλότερη καθυστέρηση. Όμως η καθυστέρηση σε σύγκριση με αυτή του flooding είναι σχεδόν μια τάξη μεγέθους μεγαλύτερη, από αυτή των άλλων σχημάτων. Αυτό είναι απότοκο της λειτουργίας του επιπέδου MAC: για τη αποφυγή συγκρούσεων λόγω μετάδοσης, επιβάλλεται η τυχαία επιλογή της καθυστέρησης σε όλες τις μεταδόσεις του MAC. Το flooding χρησιμοποιεί για της εκπομπές (broadcasts) αποκλειστικά το MAC. Το Directed Diffusion χρησιμοποιεί broadcasts μόνο για να μεταδώσει τα αρχικά ενδιαφέροντα. Εάν στην συχνοτική εκπομπή των αισθητήρων υιοθετηθεί στο Mac-στρώμα το TDMA, τότε αναμείνουμε στο flooding καθυστέρηση συγκρίσιμη με τα άλλα.

Εν περιλήψει, το Directed Diffusion εμφανίζει καλύτερη ενεργειακή διανομή από το omniscient multicast και διαθέτει καλύτερες latency (λανθάνουσα κατάσταση) ιδιότητες. Τέλος, και τα τρία πρωτόκολλα είχαν ένα ρυθμό παράδοσης γεγονότων κοντά στο ένα. Αυτό διότι σε αυτό το πείραμα αγνοήθηκε η δυναμική (dynamics) των δικτύων και η συμφόρηση.



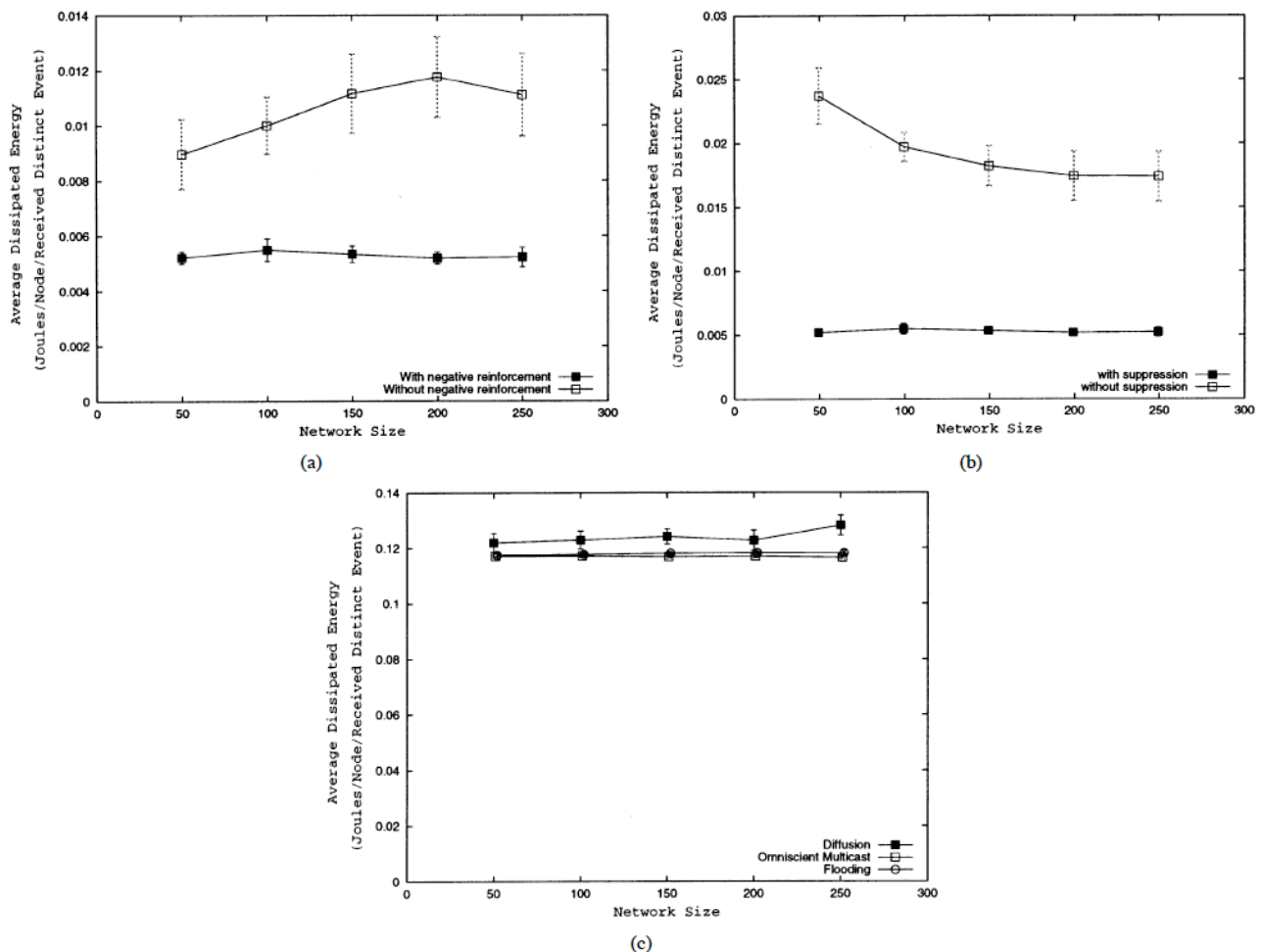
Σχήμα 3.4.2: Αντίκτυπος των αποτυχημένων κόμβων στο DD. (a) Μέση καταναλισκόμενη ενέργεια. (b) Μέση καθυστέρηση. (c) Ρυθμός παράδοσης δεδομένων.

3.4.3. Αντίκτυπος των Δυναμικών Φαινομένων

Για να μελετηθεί το αντίκτυπο των δυναμικών φαινομένων στην κατευθυνόμενη διάχυση, οι αποτυχίες των κόμβων είχαν ως εξής. Σε κάθε πεδίο αισθητήρων, απενεργοποιήθηκε επανειλημμένα ένα σταθερό μέρος (10% ή 20%) των κόμβων για 30 sec. Αυτοί οι κόμβοι επιλέχτηκαν ομοιόμορφα από το πεδίο των αισθητήρων, με τον πρόσθετο περιορισμό ότι ένας παρόμοιος αριθμός κόμβων από τις πηγές στις καταβόθρες (μέσο των συντομότερων μονοπατιών του δέντρου) επίσης απενεργοποιήθηκε κατά την ίδια διάρκεια. Η πρόθεση ήταν η δημιουργία αποτυχημένων κόμβων στις διαδρομές που η διάχυσης πιθανόν θα χρησιμοποιήσει. Επιπλέον, αντίθετα από το προηγούμενο πείραμα, κάθε πηγή στέλνει τις διαφορετικές εκτιμήσεις θέσης (που αντιστοιχούν στις καταστάσεις κατά τις οποίες κάθε πηγή εντοπίζει διαφορετικά οχήματα). Αυτό έγινε επειδή ο αντίκτυπος της δυναμικής είναι λιγότερο εμφανής όταν κόβονται οι εκτιμήσεις των τοποθεσιών (location) που είναι ίδιες και σε άλλες πηγές. Θα μπορούσαμε επίσης να μελετηθεί τον αντίκτυπο της δυναμικής σε άλλα πρωτόκολλα, αλλά το omniscient multicast είναι εξιδανικευμένο. Δεν εμπεριέχεται ο παράγοντας υπολογισμού του κόστους των διαδρομών, άρα δεν είναι σαφές ότι μια τέτοια σύγκριση έχει νόημα.

Αυτό το πείραμα επιβάλλει αρκετά δυσμενείς όρους σε ένα πρωτόκολλο διάδοσης δεδομένων. Ανά πάσα στιγμή ένα 10% ή 20% των κόμβων του δικτύου δεν χρησιμοποιείται. Επιπλέον, δεν επιτρέπουμε κανένα χρονικό διάστημα (settling time) για την αντιμετώπιση των αποτυχιών των κόμβων. Ακόμα κι έτσι, η διάχυση είναι σε θέση να διατηρήσει λογικό ρυθμό παράδοσης γεγονότων [Σχ.3.4.2(c)]. Ενώ ταυτόχρονα απαιτείται παραπάνω, μόλις 20% μέση χρονική καθυστέρηση [Σχ.3.4.2(b)]. Επιπλέον, η μέση καταναλισκόμενη ενέργεια παρουσιάζει βελτίωση, παρά τις αποτυχίες των κόμβων. Αυτό θεωρείται λίγο αντιφατικό, δεδομένου ότι θα αναμενόταν η κατευθυνόμενη διάχυση να χρησιμοποιεί ενέργεια για την ανεύρεση εναλλακτικών διαδρομών. Όπως προκύπτει, αν και οι κανόνες αρνητικής ενίσχυσης είναι συντηρητικοί, αρκετά ενισχυμένα μονοπάτια (υψηλής ποιότητας) διατηρούνται κανονικά σε λειτουργία. Πράγμα που γλιτώνει τη διάχυση από αρκετή δουλειά. Η χαμηλότερη ενεργειακή κατανάλωση προκύπτει από την αποτυχία μερικών υψηλής ποιότητας μονοπατιών.

Από αυτά τα αποτελέσματα αντιλαμβανόμαστε ότι οι μηχανισμοί στη διάχυση είναι σχετικά σταθεροί, απέναντι σε δυναμικές καταστάσεις. Επίσης συμπεραίνουμε ότι η διάχυση σε δυναμικές καταστάσεις, δεν υφίσταται εντυπωσιακά υψηλότερες καθυστερήσεις, ενεργειακές καταναλώσεις ή καθυστερήσεις στην παράδοση πακέτων.



Σχήμα 3.4.4: Αντίκτυπος διαφόρων συντελεστών του DD. (a) Αρνητική ενίσχυση. (b) εξάλειψη αντιγράφων. (c) υψηλό ανενεργό radio power.

3.4.4. Αντίκτυπος της Επεξεργασίας Δεδομένων και της Αρνητικής Ενίσχυσης

Για να εξηγηθεί η ενεργειακή αποδοτικότητα του Directed Diffusion, θα περιγράψουν δύο χωριστά πειράματα. Και στα δύο πειράματα, δεν εμφανίζονται αποτυχίες κόμβων. Υπολογίζεται η ενεργειακή αποδοτικότητα της διάχυσης με και χωρίς επεξεργασία. Εφαρμόζεται μια απλή στρατηγική επεξεργασίας κατά την οποία ένας κόμβος «εξοντώνει» τα ίδια δεδομένα, που στέλνονται από τις διαφορετικές πηγές. Όπως παρουσιάζεται στο Σχ.3.4.3(b), η διάχυση εξοικονομεί σχεδόν πέντε φορές περισσότερη ενέργεια, για μικρά δίκτυα αισθητήρων, όταν μπορεί να εξαλείψει τα αντίγραφα. Για μεγαλύτερα πεδία αισθητήρων, η αναλογία πέφτει στο 3. Ο συντηρητικός κανόνας αρνητικής ενίσχυσης έχει συμβολή στη διαφορά της απόδοσης της διάχυσης, χωρίς εξαίρεση των αντιγράφων. Βρίσκεται σε συνάρτηση με το μέγεθος του δικτύου.

Με τον ίδιο αριθμό πηγών και καταβοθρών, τα μεγαλύτερα δίκτυα εμφανίζουν μακρύτερα εναλλακτικά μονοπάτια. Αυτές οι εναλλακτικές διαδρομές περικλύονται από την αρνητική ενίσχυση, επειδή συνεχώς παραδίδουν γεγονότα μεγάλη καθυστέρηση (latency). Κατά συνέπεια το μεγαλύτερο δίκτυο καταναλώνει λιγότερη ενέργεια.

Ο δεύτερος μηχανισμός του οποίου τα οφέλη αξιολογούνται είναι η αρνητική ενίσχυση. Αυτός ο μηχανισμός κόβει τα μονοπάτια με το μεγαλύτερο latency και μπορεί να συμβάλει σημαντικά στην εξοικονόμηση ενέργειας. Σε αυτό το πείραμα, απενεργοποιείται επιλεκτικά η αρνητική ενίσχυση και συγκρίνεται η απόδοση της κατευθυνόμενης διάχυσης με ή χωρίς την ενίσχυση. Διαισθητικά θα αναμέναμε η αρνητική ενίσχυση να συμβάλει σημαντικά στην εξοικονόμησης ενέργειας. Πράγματι, όπως βλέπουμε στο Σχ.3.4.3(a), η διάχυση χωρίς αρνητική ενίσχυση, χρησιμοποιεί σχεδόν δύο φορές περισσότερη ενέργεια από όταν δεν υιοθετείται η αρνητική ενίσχυση. Αυτό δείχνει ότι ακόμη και οι συντηρητικοί κανόνες αρνητικής ενίσχυσης, κόβουν μονοπάτια που παραδίδουν συνεχώς με υψηλότερο latency.

Ελλείψη αρνητικής ενίσχυσης ή εξάλειψης των αντιγράφων, η καθυστέρηση της διάχυσης αυξάνεται κατά τρεις έως οκτώ τάξεις. Στη διάχυση η κυκλοφορία(traffic) των δεδομένων γίνεται με τη χρήση του MAC unicast. Καθώς περισσότερα μονοπάτια χρησιμοποιούνται (ελλείψη της αρνητικής ενίσχυσης), ή περισσότερα αντίγραφα δεδομένων στέλνονται (χωρίς εξάλειψη), οι διαμάχες εντός του καναλιού στο Mac επιπέδου αυξάνονται. Κατά συνέπεια και οι καθυστερήσεις.

3.4.5. Ανάλυση Ευαισθησίας

Τέλος αξιολογούνται η ευαισθησία των συγκρίσεων για τη συγκεκριμένη επιλογή του ενεργειακού μοντέλου. Η ευαισθησία της διάχυσης σε άλλους παράγοντες (αριθμός καταβοθρών, μέγεθος της περιοχής της πηγής) συζητείται με περισσότερες λεπτομέρειες σε άλλες εργασίες. Στις εδώ συγκρίσεις έγινε προσπάθεια για ρεαλιστική προσέγγιση στο radio των αισθητήρων. Οι συγκρίσεις πλέον έγιναν με την καταναλισκόμενη ενέργεια του AT&T Wavelan: 1.6-W για μετάδοση, 1.2-W για λήψη, και 1.15-W σε «νάρκη» (idle). Σε αυτήν την περίπτωση, όπως παρουσιάζεται στο Σχ.3.4.3(c) η διάκριση μεταξύ των πρωτοκόλλων εξαφανίζεται. Υπό αυτές τις συνθήκες καλύτερα να προβούμε σε flooding των γεγονότων. Αυτό είναι επειδή κατά την idle περίοδο η ενέργειας επηρεάζει καθοριστικά την απόδοση όλων των πρωτοκόλλων. Αυτός είναι και ο λόγος που οι αισθητήρες προσπαθούν πολύ σκληρά να ελαχιστοποιήσουν τις μεταδόσεις κατά την περίοδο ακρόασης (listening).

3.5. Παρόμοιες Εργασίες

Τα καταναμημένα δίκτυα αισθητήρων συγκεντρώνουν την προσοχή τα τελευταία χρόνια. Η εργασία αυτή έχει επηρεαστεί από ποικίλες άλλες ερευνητικές προσπάθειες, οι οποίες θα περιγράψουν τώρα. Τα καταναμημένα δίκτυα αισθητήρων αποτελούν μια συγκεκριμένη περίπτωση του ubiquitous computing (πανταχού παρόντος υπολογισμού) όπως διατυπώθηκε από τον Weiser. Οι πρωταρχικές προσπάθειες του ubiquitous computing, δεν εστίασαν στα ζητήματα κλιμάκωσης και συντονισμού των κόμβων. Εστίαζαν περισσότερο στα ζητήματα σχεδιασμού μικρών, ασυρμάτων συσκευών. Μεραγενέστερες προσπάθειες, όπως το WINS και το Piconet εξετάζουν θέματα δικτύωσης και επικοινωνίας για μικρές ασύρματες συσκευές. Το WINS σημείωσε σημαντική πρόοδο στον προσδιορισμό εφικτών ραδιο σχεδίων (radio designs) για χαμηλής ισχύος περιβαλλοντική αίσθηση. Ακόμα εστίασε σε χαμηλού επιπέδου συγχρονισμό που είναι απαραίτητος για την αυτοδιόρθωση των δικτύων. Το πρόγραμμα Piconet στρέφεται περισσότερο στην ανακάλυψη πληροφοριών σε σπίτια και γραφεία. Η εργασία αυτή βασίζεται περισσότερο σε κεντροποιημένες υποδομές.

Επιπλέον άλλες προσπάθειες, όπως το SPIN και το LEACH έχουν επισημάνει μερικά από τα πλεονεκτήματα της διάχυση στα πλαίσια των δικτύων αισθητήρων. Ιδιαίτερα, το SPIN ανέδειξε πώς ενσωματώνοντας τη σημασιολογία (semantics) της εφαρμογής μέσα στο flooding μπορεί να βελτιώσει την ενεργειακή αποδοτικότητα. Το LEACH και η διάχυση ερευνούν αρκετά από αυτά τα θέματα, αλλά στο πλαίσιο εκλεπτυσμένων καταναμημένων αλγορίθμων αίσθησης. Συγκεκριμένα το LEACH επιτυγχάνει ενεργειακή εξοικονόμηση από την επεξεργασία δεδομένων στο επίπεδο της εφαρμογής. Η επεξεργασία λαμβάνει χώρα στα cluster heads, ενώ στη διάχυση (directed diffusion) οπουδήποτε στο δίκτυο.

Μερικές εμπνεύσεις για την κατευθυνόμενη διάχυση προέρχονται από το χώρο της βιολογίας. Τέτοια είναι τα πρότυπα αντίδραση-διάχυσης για τη μορφογένεση και τα πρότυπα συμπεριφοράς μυρμηγκιών-αποικιών.

Η κατευθυνόμενη διάχυση δανείζεται αρκετά από τη βιβλιογραφία της ad hoc unicast δρομολόγησης. Προσεγγίζει περισσότερο την κατηγορία διάφορων πρωτοκόλλων δρομολόγησης, που βασίζονται στην αντίδραση στα γεγονότα που προκύπτουν (reactive). Από αυτά ξεχωρίζουν αυτά που προσπαθούν να εντοπίσουν και να επισκευάσουν κόμβους που αποτυγχάνουν και δεν δίνουν έμφαση στα βέλτιστα μονοπάτια.

Η κατευθυνόμενη διάχυση επηρεάζεται από το σχεδιασμός των multicast πρωτοκόλλων δρομολόγησης. Πιο συγκεκριμένα η διάδοση των ενισχύσεων και οι αρνητικές ενισχύσεις είναι παρόμοιες με τις ενώσεις (joins) και τα κοψήματα στην κατασκευή διαμοιραζόμενων δέντρων (shared trees). Η αρχική οργάνωση της διάδοσης των gradients και των ενδιαφερόντων θυμίζει την οργάνωση της οδήγησης των δεδομένων στα μονοπάτια των δέντρων. Η διαφορά είναι ότι ενώ τα πρωτόκολλα Διαδικτύου βασίζονται στην underling unicast δρομολόγηση για την οργάνωση των δέντρων, στη διάχυση δεν έχουμε τέτοιες δυνατότητες. Η διάχυση στο αντίποδα μπορεί να κάνει ενδοδικτυακή επεξεργασία δεδομένων (caching και aggregation).

Τέλος, η διάδοση ενδιαφέροντος, η διάδοση δεδομένων και η αποθήκευση (caching) στην κατευθυνόμενη διάχυση εμφανίζουν ομοιότητες με μερικές από τις ιδέες που χρησιμοποιούνται στην προσαρμοστική αποθήκευση του Ιστού (Web caching). Σε αυτά τα σχήματα, οι μνήμες (caches) αυτοοργανώνονται σε μια ιεραρχία συνεργαστικών μνημών (caches). Μέσα από αυτές τα αιτήματα (requests) για τις σελίδες διαχέονται αποτελεσματικά.

3.6. Συμπέρασμα

Περιγράφηκε το Directed Diffusion παράδειγμα, για το σχεδιασμό καταναμημένων αλγορίθμων αισθητήρων. Από όλα αυτά εξάγουμε αρκετά συμπεράσματα. Πρώτον το Directed Diffusion οδηγεί σε εξοικονόμηση ενέργειας. Ακόμα και αν έχουν επιλεγεί μη βέλτιστα μονοπάτια, η απόδοση του ξεπερνά ιδεατά παραδοσιακά πρωτόκολλα όπως το omniscient multicast. Δεύτερον οι μηχανισμοί της διάχυσης παραμένουν ανεπηρέαστοι, κάτω από την επίδραση των δυναμικών αλλαγών του δικτύου. Τρίτον για να αναδείξει το Directed Diffusion της δυνατότητες του χρειάζεται προσοχή στο MAC επίπεδο.

4. Περιγραφή και παρουσίαση του Shawn και της υλοποίησης

4.1. Εισαγωγή στον Shawn

Το λογισμικό των ασυρμάτων δικτύων αισθητήρων πρέπει να εξετάζεται πλήρως πριν από την ανάπτυξη του σε πραγματικές συσκευές, καθώς οι κόμβοι ενός τέτοιου δικτύου δεν προσφέρουν κατάλληλα interface για αποσφαλμάτωση. Επιπλέον, ο επιτυχημένος σχεδιασμός αλγορίθμων και πρωτοκόλλων για τα ασύρματα δίκτυα αισθητήρων απαιτεί κατανόηση εις βάθος αυτών των πολύπλοκων κατανεμημένων δικτύων. Για την επίτευξη των παραπάνω στόχων συνήθως χρησιμοποιούνται τρεις βασικές προσεγγίσεις:

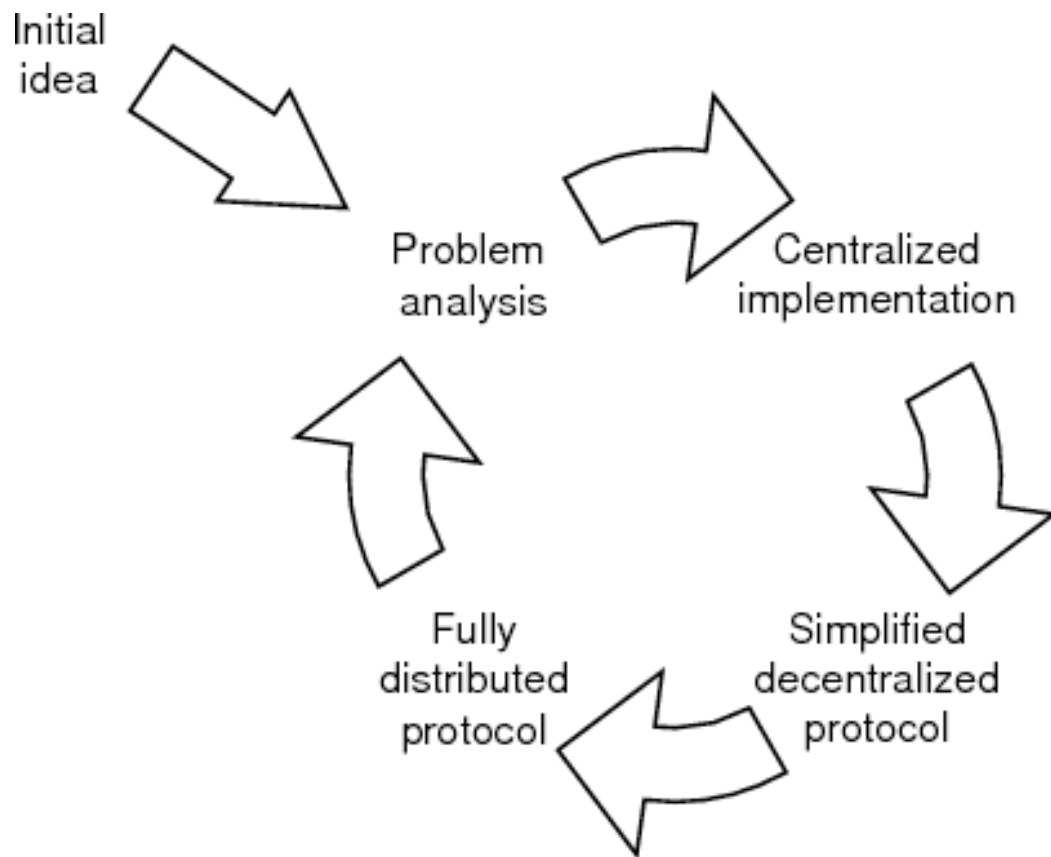
- Αναλυτικές Μέθοδοι: Οι αναλυτικές μέθοδοι τυπικά δεν είναι κατάλληλες για την υποστήριξη εφαρμογών σε ασύρματα δίκτυα αισθητήρων. Παρά την εκφραστικότητα και την γενικότητα τους είναι αρκετά δύσκολο να κατανοήσουν τις λεπτομέρειες τέτοιων πολύπλοκων κατανεμημένων εφαρμογών με επίσημο τρόπο.
- Πειράματα σε Πραγματικές Συσκευές: Τα πειράματα σε πραγματικές συσκευές είναι μια δελεαστική επιλογή και αποτελούν πειστικά μέσα καθώς μπορούν να επιδείξουν (εφόσον η τεχνολογία είναι διαθέσιμη) ότι μια εφαρμογή πραγματοποιεί μια συγκεκριμένη εργασία. Παρόλα αυτά λόγω απρόβλεπτων περιβαλλοντικών συνθηκών είναι δύσκολο να αναπαραχθούν τα αποτελέσματα ή να απομονωθούν οι πηγές λάθους. Επιπλέον, η ανάπτυξη δικτύων στον *πραγματικό κόσμο* είναι επίπονη και περιλαμβάνει εργασίες διαχείρισης που δεν σχετίζονται άμεσα με το πρόβλημα. Για το λόγο αυτό τα πειράματα περιορίζονται σε μερικές δεκάδες συσκευές, ενώ μελλοντικά σενάρια προυποθέτουν δίκτυα χιλιάδων έως και εκατομμυρίων κόμβων.
- Εξομοιώσεις με Χρήση Υπολογιστών: Οι εξομοιώσεις σε υπολογιστές αποτελούν ένα μέσο που υπόσχεται την αντιμετώπιση της δημιουργίας αλγορίθμων και πρωτοκόλλων σε δίκτυα αισθητήρων. Κατά τις εξομοιώσεις με χρήση υπολογιστών είναι διαθέσιμο ένα σύνολο από εργαλεία προσομοίωσης, τα οποία αναπαράγουν συνθήκες πραγματικού κόσμου (όπως ιδιότητες της διάδοσης των συχνοτήτων, περιβαλλοντικές συνθήκες) μέσα σε ένα περιβάλλον εξομοίωσης. Το γεγονός αυτό μετριάζει τις απαραίτητες προσπάθειες για την ανάπτυξη ενός δικτύου στον πραγματικό κόσμο και βοηθάει στην αύξηση του μεγέθους του. Τα εργαλεία αυτά παρέχουν πολύ μεγάλο επίπεδο λεπτομέρειας όμως δεν λαμβάνουν υπόψιν έναν σημαντικό συντελεστή: το πλήθος των παραγόντων που επηρεάζουν τη συμπεριφορά ολόκληρου του δικτύου καθιστά σχεδόν αδύνατη την απομόνωση μιας συγκεκριμένης παραμέτρου.

Οι πρωταρχικοί σχεδιαστικοί στόχοι του εργαλείου Shawn αναφέρονται παρακάτω:

- Εξομοίωση των Αποτελεσμάτων που Προκαλούνται από ένα Φαινόμενο: Τα περισσότερα εργαλεία εξομοίωσης προσομοιώνουν πλήρως το MAC επίπεδο συμπεριλαμβανομένων των ιδιοτήτων της διάδοσης συχνοτήτων όπως εξασθένιση, σύγκρουση και διάδοση μέσω

πολλαπλών μονοπατιών. Μία βασική σχεδιαστική ιδέα του Shawn είναι η εξομοίωση των αποτελεσμάτων ενός φαινομένου και όχι μόνο το φαινόμενο. Έτσι το Shawn εξομοιώνει μόνο τα αποτελέσματα ενός MAC επιπέδου (απώλεια πακέτων, καθυστέρηση, φθορά) στην εκάστοτε εφαρμογή. Αυτό έχει διάφορες επιπτώσεις στις εξομοιώσεις που γίνονται με το εργαλείο Shawn. Αφενός, οι εξομοιώσεις είναι προβλέψιμες και υπάρχει κέρδος απόδοσης καθώς ένα τέτοιο μοντέλο μπορεί να υλοποιηθεί αποδοτικά, αφετέρου, το εργαλείο αυτό δεν μπορεί να παρέχει λεπτομέρεια ίδιου επιπέδου π.χ. με το εργαλείο NS-2, σχετικά με το φυσικό επίπεδο ή με φαινόμενα επιπέδου πακέτου. Παρόλα αυτά, εάν το μοντέλο επιλεγθεί σωστά τα αποτελέσματα της εξομοίωσης είναι τα ίδια. Θεωρήστε, για παράδειγμα, δύο διαφορετικές υλοποιήσεις ενός MAC επιπέδου: μια αφηρημένη που αποφέρει αυξημένη απώλεια πακέτων σε υψηλή τοπική κίνηση και μία που υπολογίζει την παρεμβολή μοναδικών πακέτων χρησιμοποιώντας μοντέλα διάδοσης συχνοτήτων (radio). Οι δύο υλοποιήσεις θα παράγουν τα ίδια αποτελέσματα στο επίπεδο εφαρμογής. Στο σημείο αυτό πρέπει να ανφερθεί ότι η ερμηνεία των αποτελεσμάτων πρέπει να λαμβάνει υπόψιν τις ιδιότητες των μοντέλων. Για παράδειγμα, η χρήση ενός απλοποιημένου μοντέλου επικοινωνίας για την παροχή των αποτελεσμάτων ενός αλγορίθμου εντοπισμού, δεν επηρεάζει την ποιότητα της λύσης. Όμως ο χρόνος εκτέλεσης του αλγορίθμου δεν είναι αντιπροσωπευτικός για πραγματικές εφαρμογές καθώς δεν υπάρχει καθυστέρηση ή απώλεια μηνυμάτων.

- Επεκτασιμότητα και Υποστήριξη Μεγάλων Δικτύων: Ένας κεντρικός στόχος του Shawn είναι να υποστηρίξει σημαντικά μεγαλύτερες τάξεις μεγέθους δικτύων από τους υπάρχοντες εξομοιωτές. Το εργαλείο Shawn μπορεί να εξομοιώσει τεράστια δίκτυα καθώς απλοποιεί τη δομή πολλών χαμηλού-επιπέδου παραμέτρων. Για την επίτευξη χαμηλών χρόνων εξομοίωσης μπορούν να επιλεγούν οι κατάλληλες επιλογές διαμόρφωσης. Αυτό επιτρέπει στους προγραμματιστές να βελτιστοποιούν την απόδοση του Shawn για κάθε διαφορετικό σενάριο. Για παράδειγμα, ένα σενάριο που δεν προϋποθέτει κινητικότητα κόμβων μπορεί να εξεταστεί διαφορετικά από ένα άλλο στο οποίο οι κόμβοι κινούνται.
- Ελεύθερη Επιλογή του Μοντέλου Υλοποίησης: Το εργαλείο Shawn υποστηρίζει ένα πολυ-επίπεδο κύκλο ανάπτυξης στον οποίο οι προγραμματιστές μπορούν ελεύθερα να επιλέξουν το μοντέλο υλοποίησης όπως φαίνεται και στο παρακάτω σχήμα.



Σχήμα 4.1: Κύκλος Ανάπτυξης του Εργαλείου Shawn

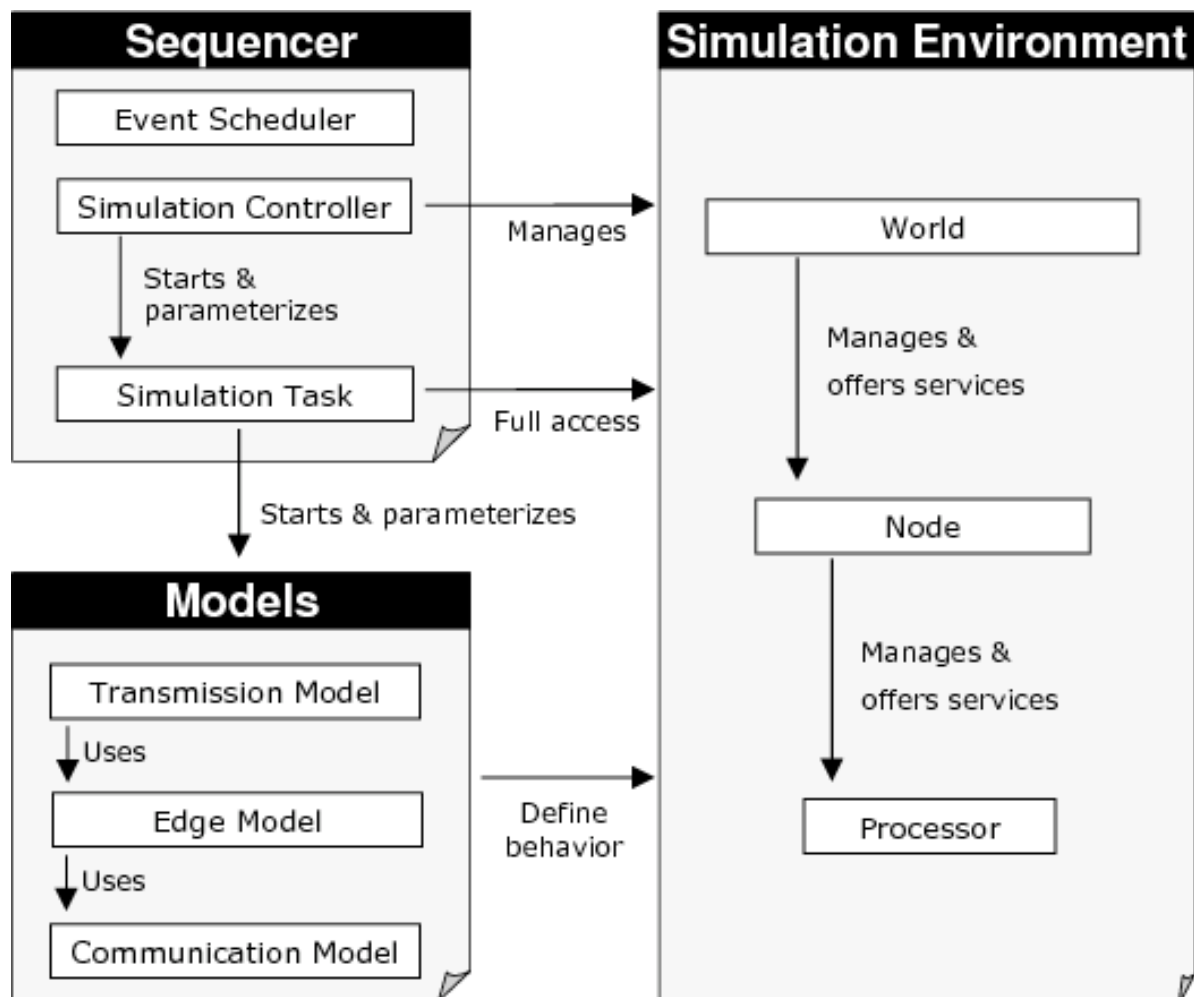
Η λογική πίσω από αυτήν την προσέγγιση είναι ότι δεδομένης μιας ιδέας για έναν αλγόριθμο, δεν σχεδιάζεται άμεσα ένα καταναμημένο πρωτόκολλο αλλά γίνεται πρώτα μια δομική ανάλυση του προβλήματος. Για την καλύτερη κατανόηση του προβλήματος προτείνεται η δομική ανάλυση διαφορετικών δικτύων. Το επόμενο βήμα είναι η υλοποίηση μιας κεντρικοποιημένης εκδοχής του αλγορίθμου για την επίτευξη μιας πρωτότυπης εκδοχής. Ένας κεντρικοποιημένος αλγόριθμος έχει πλήρη πρόσβαση σε όλους τους κόμβους και επίπεδα όψη του δικτύου. Από την διαδικασία αυτή δημιουργείται μια πρώτη εκτίμηση για την απόδοση του αλγορίθμου καθώς και τρόποι βελτίωσης του. Αφού μια κεντρικοποιημένη εκδοχή του αλγορίθμου έχει υλοποιηθεί εξετάζεται η δυνατότητα επίτευξης μιας καταναμημένης εκδοχής. Για το σκοπό αυτό χρησιμοποιείται ένα απλοποιημένο μοντέλο επικοινωνίας μεταξύ των κόμβων το οποίο επιτρέπει μια αποτελεσματική και γρήγορη υλοποίηση με σημαντικά αποτελέσματα. Τέλος, η απλοποιημένη εκδοχή μεταμορφώνεται σε ένα πρωτόκολλο το οποίο ανταλλάσσει μηνύματα στο δίκτυο. Με το πρωτόκολλο και τις κατάλληλες δομές έτοιμες, η απόδοση της καταναμημένης υλοποίησης μπορεί να αξιολογηθεί ως προς διάφορα θέματα όπως αριθμό μηνυμάτων, κατανάλωση ενέργειας και ελαστικότητα στην απώλεια και φθορά μηνυμάτων. Ο παραπάνω κύκλος ανάπτυξης που περιγράφηκε βοηθάει έναν προγραμματιστή να φτάσει από την αρχική ιδέα σε ένα πλήρως καταναμημένο πρωτόκολλο.

4.2. Αρχιτεκτονική του Shawn

Η αρχιτεκτονική του εξομοιωτή Shawn αποτελείται από τρία βασικά μέρη τα οποία φαίνονται παρακάτω:

- Τα μοντέλα (Models)
- Διαδοχέας (Sequencer)
- Περιβάλλον Εξομοίωσης (Simulation Environment)

Το εργαλείο Shawn επηρεάζεται από ένα ή περισσότερα μοντέλα (Models), τα οποία αποτελούν κλειδί για την ευελιξία και επεκτασιμότητα του. Ο διαδοχέας Sequencer είναι η κεντρική συνεργατική μονάδα που διαμορφώνει και ελέγχει την εξομοίωση και εκτελεί διαδοχικά τις εργασίες. Το περιβάλλον εξομοίωσης είναι αυτό που φιλοξενεί τον εικονικό κόσμο στον οποίο βρίσκονται οι κόμβοι του δικτύου. Παρακάτω, καθένα από τα στοιχεία της αρχιτεκτονικής εξετάζεται ξεχωριστά.



Σχήμα 4.2: Στοιχεία αρχιτεκτονικής του Shawn

4.2.1. Τα μοντέλα (Models)

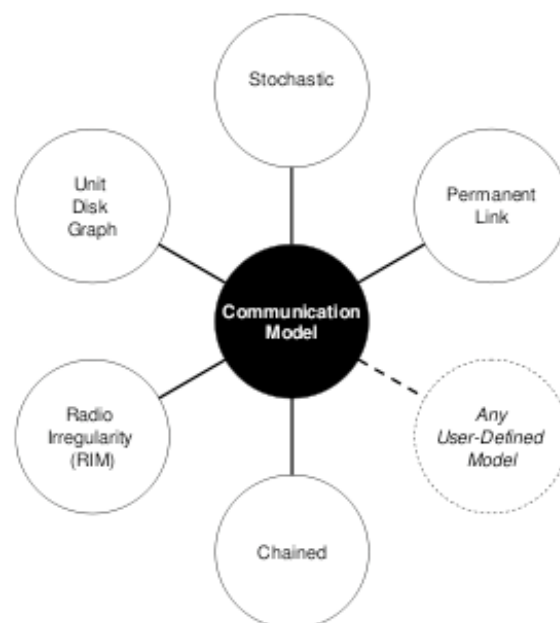
Τα μοντέλα είναι οι διεπαφές (interfaces) που χρησιμοποιεί το Shawn για να ελέγχει την εξομοίωση χωρίς να γνωρίζει την δομή μιας υλοποίησης. Το Shawn διατηρεί μια αποθήκη από υλοποιήσεις μοντέλων που μπορούν να χρησιμοποιηθούν για τον ορισμό μιας εξομοίωσης επιλέγοντας την επιθυμητή συμπεριφορά. Η υλοποίηση ενός μοντέλου μπορεί να είναι απλοποιημένη και γρήγορη ή να παρέχει κοντινή προσέγγιση της πραγματικότητας. Αυτό επιτρέπει στον χρήστη να διαλέξει την πιο κατάλληλη υλοποίηση μοντέλου για μια συγκεκριμένη εργασία. Όπως φαίνεται και στο παραπάνω σχήμα τα βασικά μοντέλα που υποστηρίζει το Shawn είναι τα παρακάτω:

- Communication Model
- Edge Model
- Transmission Model

Τα μοντέλα αυτά παρουσιάζονται εκτενώς στη συνέχεια.

1. Communication Model

Όταν ένας κόμβος του εικονικού δικτύου μεταδίδει ένα μήνυμα, οι πιθανοί παραλήπτες πρέπει να αναγνωρίζονται από τον εξομοιωτή. Το μοντέλο επικοινωνίας ορίζει εάν δύο κόμβοι μπορούν να επικοινωνήσουν. Διάφορα πρότυπα επικοινωνίας μπορούν να υλοποιηθούν. Κάποιες υλοποιήσεις του communication model φαινονται παρακάτω



Σχήμα 4.2.1.α: Υλοποιήσεις του communication model

Το μοντέλο Unit Disk Graph βασίζεται στην παρατήρηση ότι η ισχύς ενός σήματος εξασθενίζει αντιστρόφως ανάλογα με το τετράγωνο της απόστασης από τον αποστολέα. Δεδομένης μιας ισχύς σήματος που απαιτείται για λήψη εκπομπής, δύο κόμβοι μπορούν να επικοινωνήσουν αν η ευκλείδεια απόσταση τους είναι μικρότερη από r_{max} . Το μοντέλο αυτό χρησιμοποιείται συχνά στην έρευνα γύρω από τα ασύρματα δίκτυα αισθητήρων λόγω της απλότητας του.

Το μοντέλο ραδίου Quasi-Unit Disk Graph ορίζει δύο αποστάσεις r_1 και r_2 για τις οποίες ισχύει $r_1 < r_2$. Στην περίπτωση όπου $0 < d < r_1$ ή $d > r_2$ η συμπεριφορά του είναι ίδια με αυτήν του Unit Disk Graph. Στην περίπτωση όπου $r_1 \leq d \leq r_2$, η πιθανότητα λήψης ενός πακέτου μειώνεται γραμμικά από το 1 στο 0. Ισχύει δηλαδή το γεγονός ότι η πιθανότητα επιτυχημένης λήψης μειώνεται καθώς η απόσταση αυξάνεται.

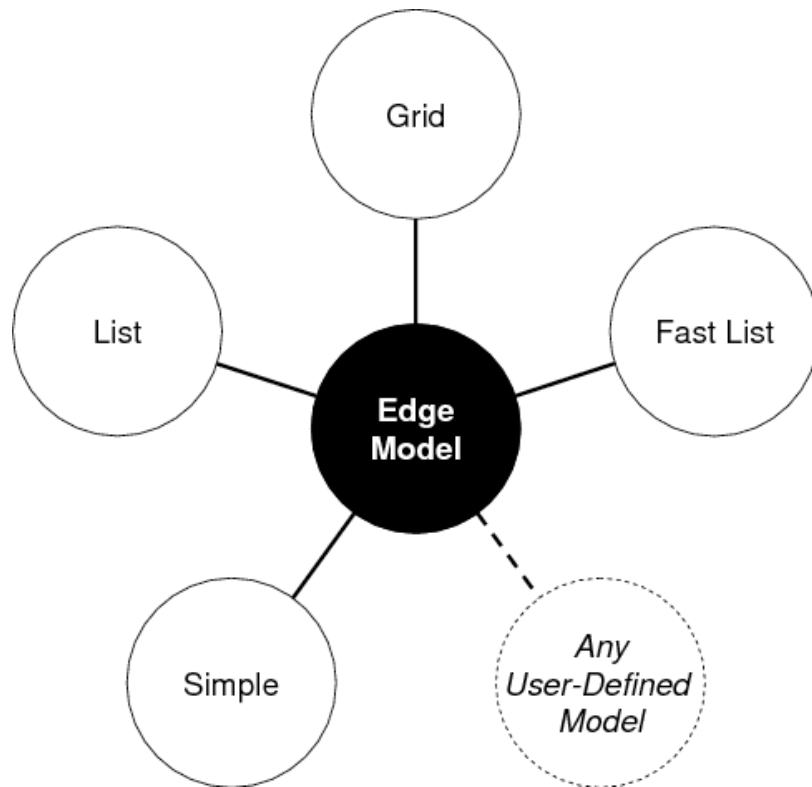
Βασιζόμενο σε πραγματικά πειράματα, το μοντέλο Radio Irregularity Model, προτείνει ένα υπο γωνία εύρος εκπομπής ανάμεσα σε ένα ελάχιστο και ένα μέγιστο εύρος επικοινωνίας (r_{min} , r_{max}). Ένας παράγοντας καθορίζει τη μέγιστη αλλαγή στο εύρος εκπομπής και έτσι ελέγχεται η αταξία. Σε αντίθεση με τα προηγούμενα μοντέλα επικοινωνίας το Radio Irregularity Model υποστηρίζει και συνδέσμους μονής κατεύθυνσης.

Εκτός από τα παραπάνω μοντέλα υπάρχουν και τα μοντέλα Permanent Link και Chained Communication Model. Το πρώτο επιτρέπει τον προσδιορισμό στατικών συνδέσμων όπως για παράδειγμα κάποιες ενσύρματες συνδέσεις που υποστηρίζουν οι κόμβοι πύλης (gateway nodes). Το Chained Communication Model υποστηρίζει τον συνδυασμό πολλών μοντέλων επικοινωνίας σε ένα καινούριο. Για παράδειγμα μπορεί όλοι οι κόμβοι του δικτύου να χρησιμοποιούν το Unit Disk Graph μοντέλο επικοινωνίας ενώ οι κόμβοι πύλης να υποστηρίζουν κάποιες ενσύρματες συνδέσεις που να μοντελοποιούνται με το μοντέλο Permanent Link.

2. Edge Model

Το μοντέλο Edge Model προσφέρει μια γραφική αναπαράσταση του δικτύου. Οι εξομοιωμένοι κόμβοι αποτελούν τις κορυφές του γραφήματος και μία ακμή ανάμεσα σε δύο κόμβους προστίθεται όποτε το Communication Model επιστρέφει true. Για τη συναρμολόγηση του γραφήματος το Edge Model συνέχεια θέτει ερωτήσεις (query) στο Communication Model, προσπελώνοντας με τον τρόπο αυτό τους άμεσους γείτονες ενός κόμβου, τους γείτονες των γειτόνων και ούτω καθ'εξής. Έτσι το Shawn παρατηρώντας τους γείτονες ενός κόμβου αναγνωρίζει τους πιθανούς παραλήπτες ενός μηνύματος. Απλοί, κεντρικοποιημένοι αλγόριθμοι που χρειάζονται τις πληροφορίες από το γράφημα επικοινωνίας μπορούν να υλοποιηθούν εύκολα και αποτελεσματικά καθώς δεν χρειάζεται να γίνει ανταλλαγή μηνυμάτων για την εύρεση των γειτόνων των κόμβων.

Ανάλογα τις απαιτήσεις και τις ιδιότητες της εκάστοτε εφαρμογής χρειάζονται διαφορετικά μοντέλα αποθήκευσης των γραφημάτων. Για παράδειγμα, σενάρια που υποθέτουν κινητικότητα των κόμβων απαιτούν διαφορετικό μοντέλο αποθήκευσης του γραφήματος του δικτύου από εκείνα που θεωρούν τους κόμβους του δικτύου στατικούς. Επιπλέον, εξομοιώσεις μικρών δικτύων επιτρέπουν την αποθήκευση ολόκληρης της γειτονιάς ενός κόμβου στη μνήμη του κάτι που είναι πολύ δύσκολο σε τεράστια δίκτυα λόγω μεγάλων απαιτήσεων μνήμης. Για τους παραπάνω λόγους, παρέχει τις παρακάτω υλοποιήσεις του Edge Model.

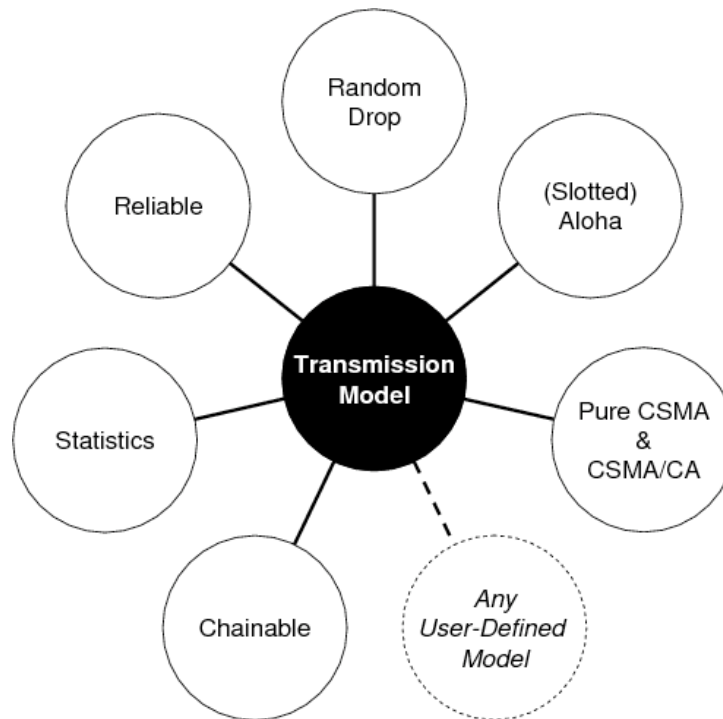


Σχήμα 4.2.1.β: Υλοποιήσεις του edge model

3. *Transmission Model*

Για κάθε μήνυμα που μεταδίδεται από έναν κόμβο, η συμπεριφορά του καναλιού μετάδοσης μπορεί να είναι τελείως διαφορετική. Παραδείγματος χάριν, η δικτυακή κίνηση από άλλους κόμβους μπορεί εμποδίζει το ασύρματο κανάλι επικοινωνίας ή ακόμα και να υποβαθμίζει την ποιότητα του. Για την εξομοίωση αυτών των παροδικών χαρακτηριστικών το εργαλείο Shawn χρησιμοποιεί το Transmission Model το οποίο καθορίζει τις ιδιότητες μιας μετάδοσης ενός μηνύματος. Το μοντέλο αυτό μπορεί τυχαία να καθυστερήσει, να τροποποιήσει ή να απορίψει ένα μήνυμα.

Η επιλογή μιας υλοποίησης του Transmission Model εξαρτάται από τον σκοπό της εξομοίωσης. Αν μας ενδιαφέρει μόνο η ποιότητα του αλγορίθμου και όχι ο χρόνος εκτέλεσης, τότε ένα απλό μοντέλο μετάδοσης χωρίς καθυστέρηση, τροποποίηση και απώλεια μηνυμάτων είναι αρκετό. Μοντέλα που είναι πιο εξηζητημένα, λαμβάνουν υπόψιν την σύγκρουση μηνυμάτων, τον χρόνο μετάδοσης και τα λάθη μετάδοσης για την μέτρηση της απόδοσης του πρωτοκόλλου. Παρακάτω φαίνονται τα βασικά Transmission Models που υποστηρίζει το Shawn.



Σχήμα 4.2.1.γ: Τα βασικά transmission models

4.2.2. Διαδοχέας (Sequencer)

Ο Sequencer αποτελεί το κέντρο ελέγχου της εξομοίωσης: προετοιμάζει τον κόσμο μέσα στον οποίο οι κόμβοι θα αναπτυχθούν, αρχικοποιεί και παραμετροποιεί τις υλοποιήσεις των μοντέλων που έχουν επιλεγεί από την διαμόρφωση εισόδου. Αποτελείται από:

- Simulation Tasks
- Simulation Controller
- Event Scheduler

1. Simulation Tasks

Τα Simulation Tasks αποτελούν κομμάτια κώδικα που καλούνται από την διαμόρφωση της εξομοίωσης που παρέχεται από τον χρήστη. Δεν είναι άμεσα συνδεδεμένα με την εφαρμογή αλλά έχουν πρόσβαση στο περιβάλλον εξομοίωσης και είναι ικανά να εκτελέσουν ένα σύνολο από εργασίες (tasks), όπως π.χ. διαχείριση της εξομοίωσης, εκτέλεση κεντρικοποιημένων αλγορίθμων και συλλογή δεδομένων από τους κόμβους.

Το εργαλείο Shawn χρησιμοποιεί τα tasks για να αποκαλύψει τα εσωτερικά του χαρακτηριστικά στον χρήστη. Περιλαμβάνει ένα σύνολο εργασιών που δημιουργούν κόσμους εξομοίωσης, κόμβους μέσα σε αυτούς, πρωτόκολλα δρομολόγησης, τυχαίες μεταβλητές κτλ. Ακόμα και η εξομοίωση

ενεργοποιείται χρησιμοποιώντας μια εργασία στην οποία ο χρήστης μπορεί να ορίσει το χρόνο εξομοίωσης.

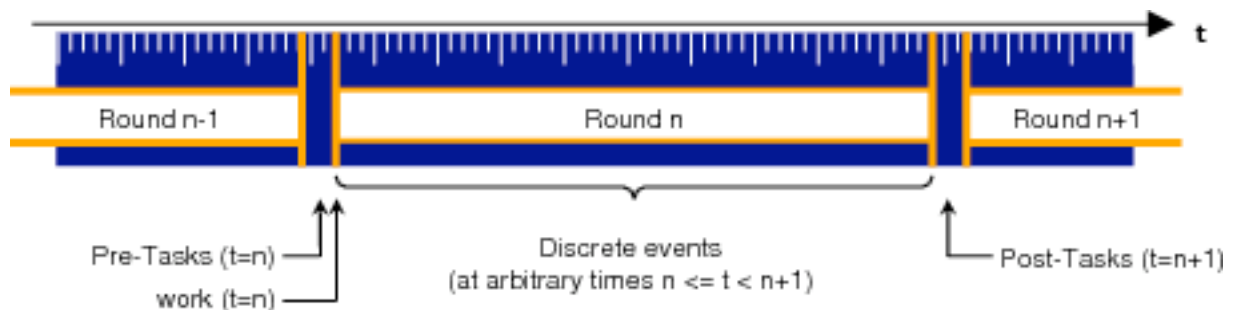
2. Simulation Controller

Ο Simulation Controller αποτελεί την κεντρική αποθήκη για όλες τις υλοποιήσεις των μοντέλων και εκτελεί την εξομοίωση μετατρέποντας τη διαμόρφωση εισόδου (configuration) σε παραμετροποιημένες επικλήσεις των Simulation Tasks. Με τον τρόπο αυτό ο Simulation Controller αποτελεί το μέσο ανάμεσα στον πυρήνα της εξομοίωσης και τον χρήστη. Όπως και τα υπόλοιπα στοιχεία του Shawn, ο Simulation Controller μπορεί να προσαρμοστεί από έναν προγραμματιστή ώστε αυτός να αποκτήσει τον έλεγχο της εξομοίωσης. Η βασική υλοποίηση διαβάζει τις εντολές διαμόρφωσης από ένα αρχείο ή από το ρεύμα εισόδου.

3. Event Scheduler

Το εργαλείο Shawn χρησιμοποιεί τον Event Scheduler για να μοντελοποιήσει τον χρόνο. Τα αντικείμενα που χρειάζονται την έννοια του χρόνου καταχωρούνται στον Event Scheduler και ειδοποιούνται σε τυχαία στιγμή. Η εξομοίωση πάντα προχωράει προς το επόμενο γεγονός και αντίστοιχα ειδοποιεί τον κατάλληλο χειριστή του γεγονότος αυτού. Η διαδικασία συνεχίζεται μέχρι όλοι οι κόμβοι του δικτύου να γίνουν ανενεργοί ή μέχρι να τελειώσει ο χρόνος εξομοίωσης.

Η χρήση του Event Scheduler έχει κάποια πλεονεκτήματα σε σχέση με εκείνες τις προσεγγίσεις που χρησιμοποιούν προκαθορισμένα διαστήματα χρόνου (όπως το clock-tick κάθε 1 ms): Πρώτον, οι χειριστές καλούνται μόνο κατά την ακριβή ώρα που έχουν ζητήσει αποφεύγοντας έτσι κλήσεις σε κόμβους που είναι αδρανείς ή σε αναμονή. Δεύτερον, ένα γεγονός μπορεί να συμβεί με τη μέγιστη ακρίβεια που προσφέρεται από τους αριθμούς κινητής υποδιαστολής. Στην παρακάτω εικόνα φαίνεται πως ο Event Scheduler επιτρέπει στις εξομοιώσεις να χρησιμοποιήσουν αυτόν το χρονομηχανισμό.



Σχήμα 4.2.2: Event scheduler

Το Shawn οργανώνει λογικά τις εξομοιώσεις σε γύρους ($i=0,1,2,\dots$). Ένας χρήστης μπορεί να καταχωρήσει Simulation Tasks ως pre-step ή ως post-step εργασίες οι οποίες εκτελούνται αμέσως πριν ή αμέσως μετά αντίστοιχα, από τον γύρο της εξομοίωσης. Αυτό είναι χρήσιμο για την παραγωγή πληροφορίας από την εξομοίωση χωρίς να αναμειχθεί κώδικας της εξομοίωσης με κώδικα που αφορά την απόδοση του αλγορίθμου. Στην αρχή κάθε γύρου, καλείται η work() μέθοδος ενός κόμβου. Οι εφαρμογές μπορούν να επιλέξουν να χρησιμοποιήσουν αυτή τη μέθοδο για μία πιο άνετη

υλοποίηση εφόσον δεν απαιτείται ακριβής συγχρονισμός. Εκτός από αυτά τα σταθερά διαστήματα, οι κόμβοι μπορούν να καταχωρήσουν ένα γεγονός σε οποιοδήποτε σημείο στο χρόνο. Με τον τρόπο αυτό οι εφαρμογές έχουν την ευελιξία να ενσωματώσουν χρονικά πλαίσια στην εξομοίωση χωρίς να υποβαθμίσουν την απόδοσή τους.

4.2.3. Περιβάλλον Υλοποίησης

Το περιβάλλον εξομοίωσης (Simulation Environment) φιλοξενεί τον εικονικό κόσμο μέσα στον οποίο τα αντικείμενα της εξομοίωσης αναπτύσσονται. Οι κόμβοι ανήκουν σε ένα στιγμιότυπο του εικονικού κόσμου και περιέχουν τους λεγόμενους επεξεργαστές (processors).

Οι προγραμματιστές που χρησιμοποιούν το εργαλείο Shawn υλοποιούν την εφαρμογή που επιθυμούν σαν στιγμιότυπα των επεξεργαστών. Κάθε κόμβος μπορεί να περιέχει πολλαπλούς επεξεργαστές έτσι ώστε πολλές εφαρμογές να μπορούν να συνδυαστούν σε μια και μόνο εκτέλεση εξομοίωσης. Για παράδειγμα, ένας επεξεργαστής μπορεί να εκτελεί ένα συγκεκριμένο πρωτόκολλο ενώ παράλληλα ένας άλλος μαζεύει στατιστικά δεδομένα.

4.3. Οδηγίες για την εγκατάσταση του Shawn σε Unix συστήματα

Εγκατάσταση του CMake

1. Εάν δεν είναι ήδη εγκατεστημένο το κατεβάζουμε από <http://www.cmake.org/HTML/Download.html> . Για τη σωστή λειτουργία πρέπει να κατεβάσουμε την 2.4 Cmake έκδοση ή νεότερη. Εναλλακτικά ανάλογα με τη διανομή, επιλέγουμε το πακέτο από τον αντίστοιχο διαχειριστή πακέτων. Για παράδειγμα στο Debian και στις βασισμένες σε αυτό διανομές καλούμε στην κονσόλα το : `sudo apt-get install cmake`. Επίσης μπορούμε να εγκαταστήσουμε το `ccmake`, ένα πιο γραφικό περιβάλλον διεπαφής για το `cmake`. Αυτό σε ορισμένες διανομές Linux δεν αποτελεί μέρος του πακέτου CMake.
2. Εάν το build γίνεται με το «χέρυ», δηλαδή μέσω κονσόλας, αφού αποσυμπέσουμε το αρχείο με το περιεχόμενο εργαλείο, ακολουθούμε τις οδηγίες του αρχείου `Readme.txt` . Τα βήματα της εγκατάστασης είναι τα εξής:
 - `./bootstrap`
 - `make`
 - `make install`

Όλα αυτά εντός του αποσυμπιεσμένου φακέλου `cmake`.

Εγκατάσταση του Java Runtime περιβάλλοντος

Για να χρησιμοποιηθεί ο JShawn πρέπει να εγκατασταθεί το Java Runtime Environment και η ελάχιστη έκδοση που απαιτείται είναι η 1.6. Μπορεί κανείς να την κατεβάσει από το : <http://www.java.com/en/>

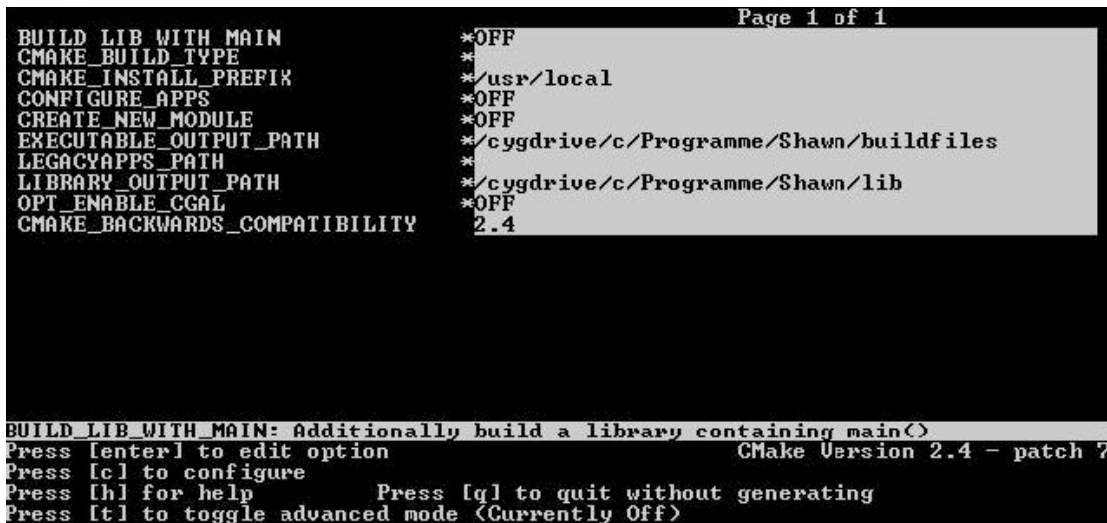
Κατέβασμα του Shawn

Αρχικά πρέπει να βεβαιωθούμε ότι το Subversion είναι εγκαταστημένο. Αν όχι το κατεβάζουμε από το http://subversion.tigris.org/project_packages.html . Αλλιώς χρησιμοποιούμε τον διαχειριστή πακέτων της διανομής (sudo apt-get install subversion για Debian).

1. Ανοίγουμε την κονσόλα και πηγαίνουμε στη περιοχή, που έχει δημιουργηθεί ο νέος φάκελος του Shawn
2. Τσεκάρουμε τον Shawn από την κονσόλα κάνοντας χρήση του εργαλείου svn
svn co <https://shawn.svn.sourceforge.net/svnroot/shawn>

Παράγουμε το makefile

1. Μέσο του τερματικού πάμε στο φάκελο shawn/buildfiles
2. Εκεί καλούμε cmake ../src
3. Πατάμε c για να αρχικοποιήσουμε τη διαμόρφωση του εξομοιωτή και αναμένουμε μέχρι να λάβουμε την εικόνα :



```
Page 1 of 1
BUILD_LIB_WITH_MAIN      *OFF
CMAKE_BUILD_TYPE        *
CMAKE_INSTALL_PREFIX    */usr/local
CONFIGURE_APPS           *OFF
CREATE_NEW_MODULE        *OFF
EXECUTABLE_OUTPUT_PATH  */cygdrive/c/Programme/Shawn/buildfiles
LEGACYAPPS_PATH         *
LIBRARY_OUTPUT_PATH     */cygdrive/c/Programme/Shawn/lib
OPT_ENABLE_CGAL          *OFF
CMAKE_BACKWARDS_COMPATIBILITY 2.4

BUILD_LIB_WITH_MAIN: Additionally build a library containing main()
Press [enter] to edit option                               CMake Version 2.4 - patch 7
Press [c] to configure
Press [h] for help                                       Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)
```

4. Πηγαίνουμε στη γραμμή CONFIGURE_APPS και την θέτουμε σε ενεργή κατάσταση, δηλαδή ON.
5. Πατούμε ξανά c και καταλήγουμε στην παρακάτω εικόνα

```

Page 1 of 2
MODULE_APPS_AUTOCAST *OFF
MODULE_APPS_DISTEST *OFF
MODULE_APPS_EXAMPLES *OFF
MODULE_APPS_ISENSE *OFF
MODULE_APPS_LOCALIZATION *OFF
MODULE_APPS_MOTIONEVENTTASK *OFF
MODULE_APPS_READING *OFF
MODULE_APPS_ROUTING *OFF
MODULE_APPS_SPYGLASS *OFF
MODULE_APPS_TCPIP *OFF
MODULE_APPS_TOPOLOGY *OFF
MODULE_APPS_TRACICLIENT *OFF
BUILD_LIB_WITH_MAIN OFF
CMAKE_BACKWARDS_COMPATIBILITY 2.4
CMAKE_BUILD_TYPE
CMAKE_INSTALL_PREFIX /usr/local
CONFIGURE_APPS ON
CREATE_NEW_MODULE OFF

MODULE_APPS_AUTOCAST: Activate module AUTOCAST
Press [enter] to edit option CMake Version 2.4 - patch 7
Press [c] to configure
Press [h] for help Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)

```

6. Ενεργοποιούμε όλες τις απαιτούμενες εφαρμογές ξεκινώντας με την MODULE_APPS_ . Αυτό γίνεται μετακινώντας τον κέρσορα στη σωστή γραμμή και πατώντας Enter. Για αρχή είναι αρκετή η ενεργοποίηση των module apps *_EXAMPLES, \\ *_LOCALIZATION, *_READING, and *_TOPOLOGY.
Αν δέν αναπτύσσεται εφαρμογή στο iSense-API δεν πρέπει να ενεργοποιηθεί το *_ISENSE, επειδή θα οδηγηθούμε σε σφάλματα.
7. Πατούμε πάλι c για να ανανεώσουμε την υπάρχουσα διαμόρφωση του εξομοιωτή.
8. Τέλος πατούμε g για να παράγουμε το makefile, που χρησιμοποιείται για το compile του Shawn.

Compile Shawn

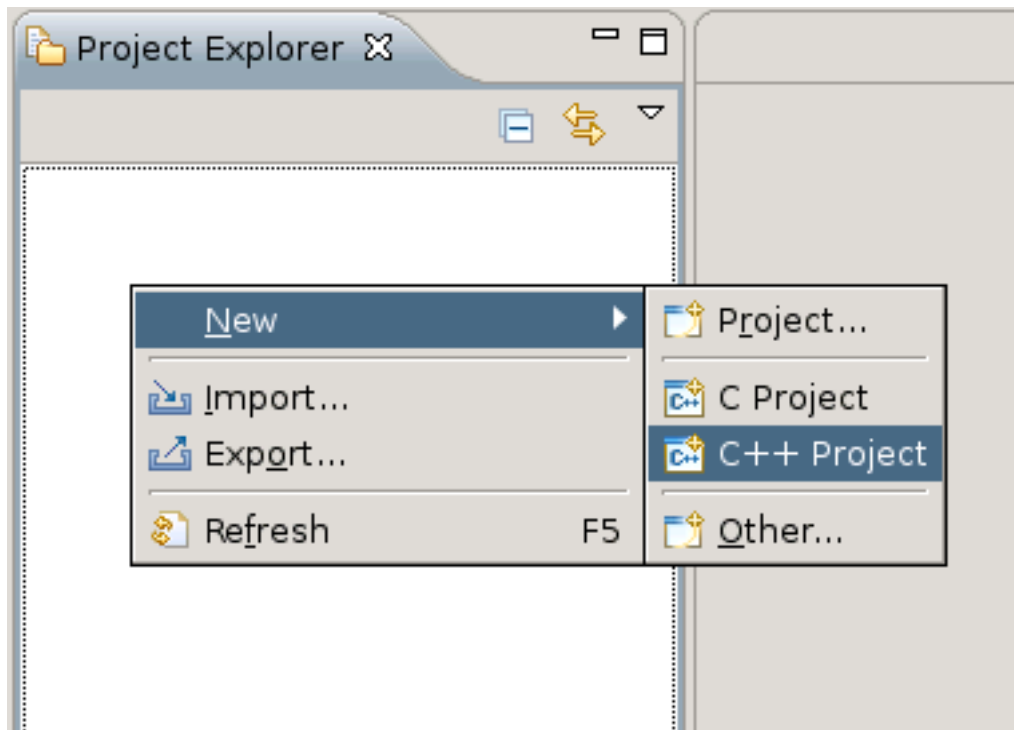
Όντας στο shawn/buildfiles καλούμε στο τερματικό την εντολή make, για να αρχίσει η διαδικασία του compile. Ανάλογα με τις δυνατότητες της cpu θα χρειαστούν μερικά λεπτά. Με τη λήξη της διαδικασίας θα βρίσκεται εντός του φακέλου buildfiles ένα εκτελέσιμο shawn.

Η τελευταία επιτυχημένη έκδοση του GCC που έχει τεσταριστεί, είναι η 4.3 (Debian 4.3.0-3) 4.3.1 20080401 (prerelease).

Εισαγωγή του Shawn στο Eclipse

Για την ευκολότερη διαχείριση των αρχείων και την απρόσκοπτη συγγραφή ατομικών εφαρμογών, μπορούμε να χρησιμοποιήσουμε το Eclipse ή οποιοδήποτε άλλο IDE. Για να γίνει αυτό πρέπει να εισάγουμε τον Shawn στον Eclipse. Για αυτό :

1. Κατεβάζουμε το Eclipse από <http://www.eclipse.org/downloads/> . Έχουμε την δυνατότητα να επιλέξουμε το Eclipse IDE για C/C++ developers απευθείας ή να κατεβάσουμε επιπλέον τον CDT πακέτο.
2. Δημιουργούμε ένα νέο C++ project και θέτουμε το μονοπάτι στο οποίο βρίσκεται ο Shawn. Για αυτό πατούμε δεξί click στο project explorer και επιλέγουμε New -> C++ Project. Οι εικόνες δίνουν πιο σαφή αίσθηση.



Απενεργοποιούμε το κουτί Use default location και επιλέγουμε το φάκελο που περιέχει τον Shawn. Επίσης θέτουμε ένα όνομα για το project. Ακόμα επιλέγουμε Makefiles project και Linux GCC ως toolchain (δηλαδή σύνολο από εργαλεία/προγράμματα απαραίτητα για την περάτωση μιας εφαρμογής).

C++ Project

⚠ Directory with specified name already exists.



Project name:

Use default location

Location:

Project types:

- ▶ Executable
- ▶ Shared Library
- ▶ Static Library
- ▶ **Makefile project**

Toolchain:

- Other Toolchain --
- Linux GCC**

Show project types and toolchains only if they are supported on the platform

Select Configurations

Select platforms and configurations you wish to deploy on



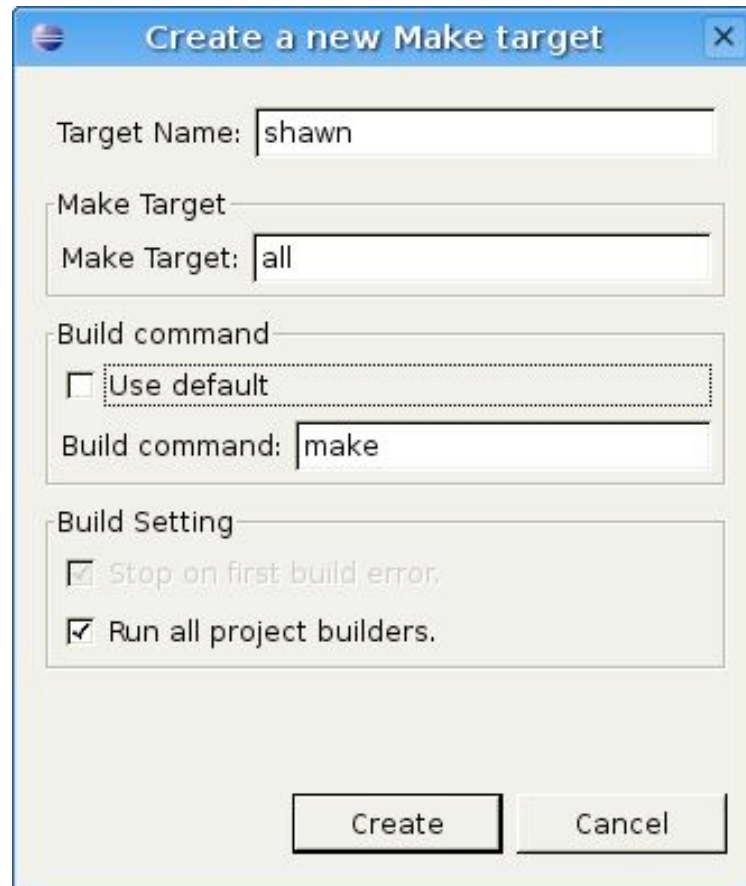
Project type: Makefile project
Tool-chains: Linux GCC
Configurations:

Linux GCC

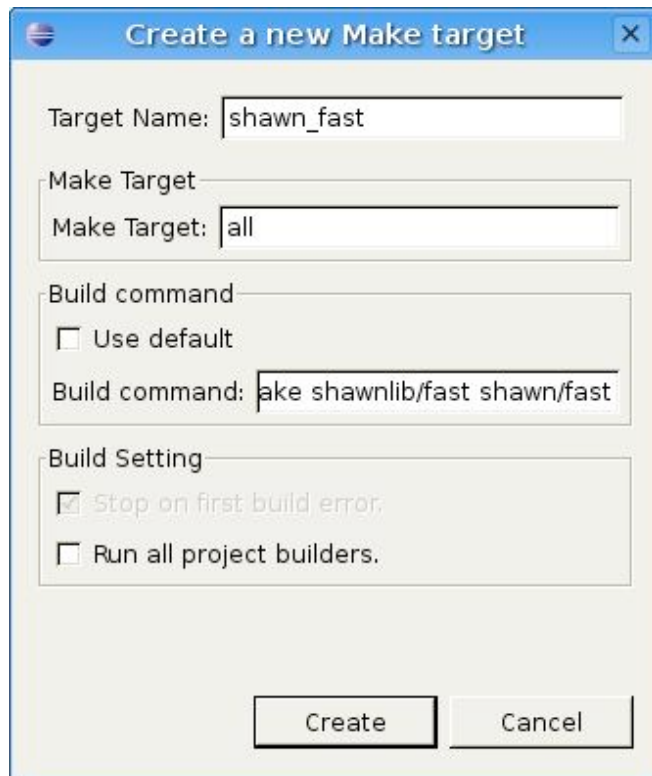
Use "Advanced settings" button to edit project's properties.
Additional configurations can be added after project creation.
Use "Manage configurations" buttons either on toolbar or on property pages.

3. Δημιουργούμε ένα καινούριο make target για αυτό το project εντός του Shawn/buildfiles. Για αυτό το σκοπό ανοίγουμε το Make Target – View, πάμε στο Shawn -> buildfiles. Μετά δεξί click εκεί και επιλέγουμε Add Make Target. Ένα νέο παράθυρο εμφανίζεται, που συμπληρώνεται ως εξής:

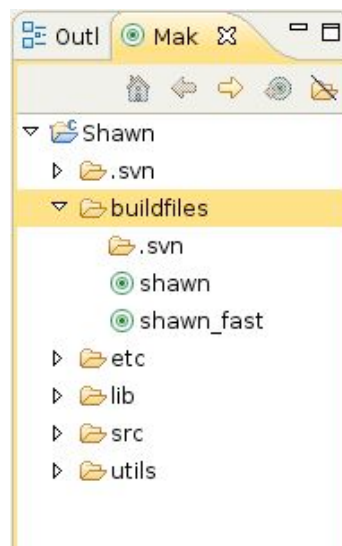
Target Name: shawn
Make Target: all
Build command: make



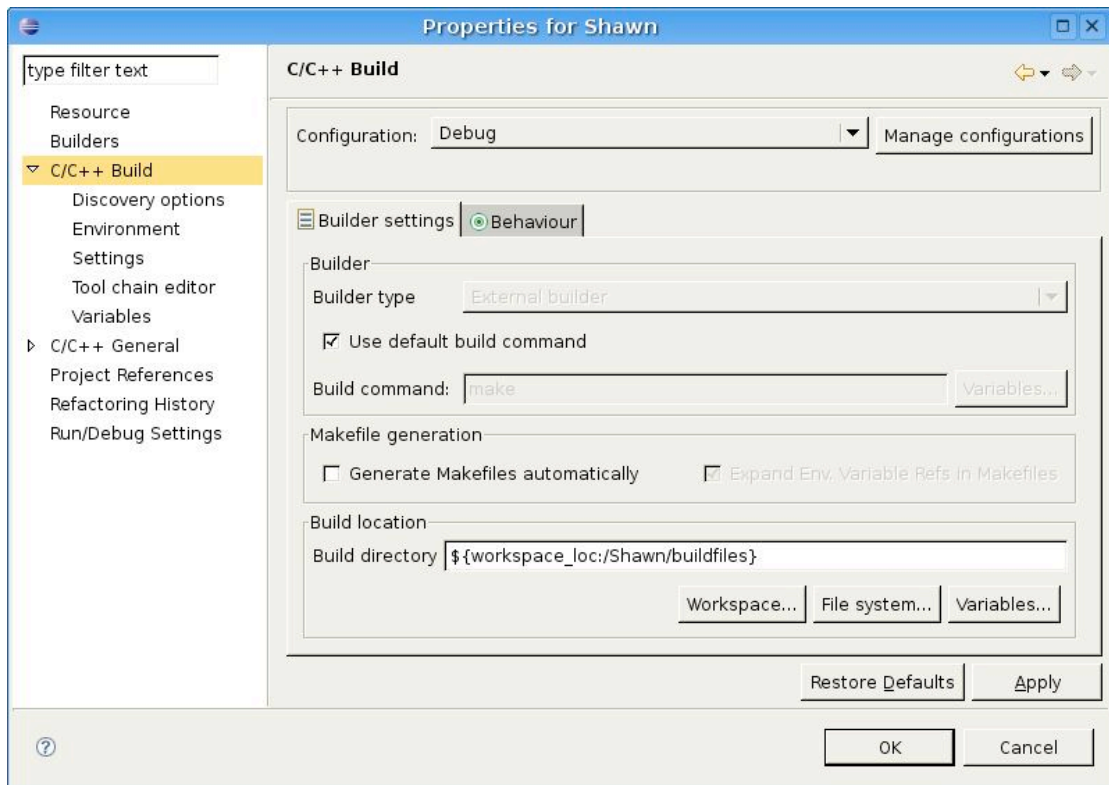
Τελειώνουμε κλικάροντας το κουμπί Create. Με αυτό το σταθερό make target μια πολύπλοκη διαδικασία ξεκινά. Οι εξαρτήσεις επανα-υπολογίζονται και ελέγχεται το Cmake σύστημα για τυχόν αλλαγές. Ακόμα και η προσθήκη ενός σχολίου σε ένα Cmake configuration αρχείο, οδηγεί σε compile ολόκληρου του κώδικα. Για να γλιτώσουμε τη χρονοβόρα διαδικασία δημιουργούμε ένα «fast target». Αυτό κάνει compile στοχευμένα. Οι αλλαγές εντοπίζονται στο πηγαίο αρχείο και αναδημιουργείται το binary.



Στη συνέχεια το Make Targets-View θα είναι :



4. Θέτουμε στο project, properties->C/C++ Build: Build directory: workspace_loc:/Shawn/buildfile οπότε η εικόνα θα έχει ως εξής:

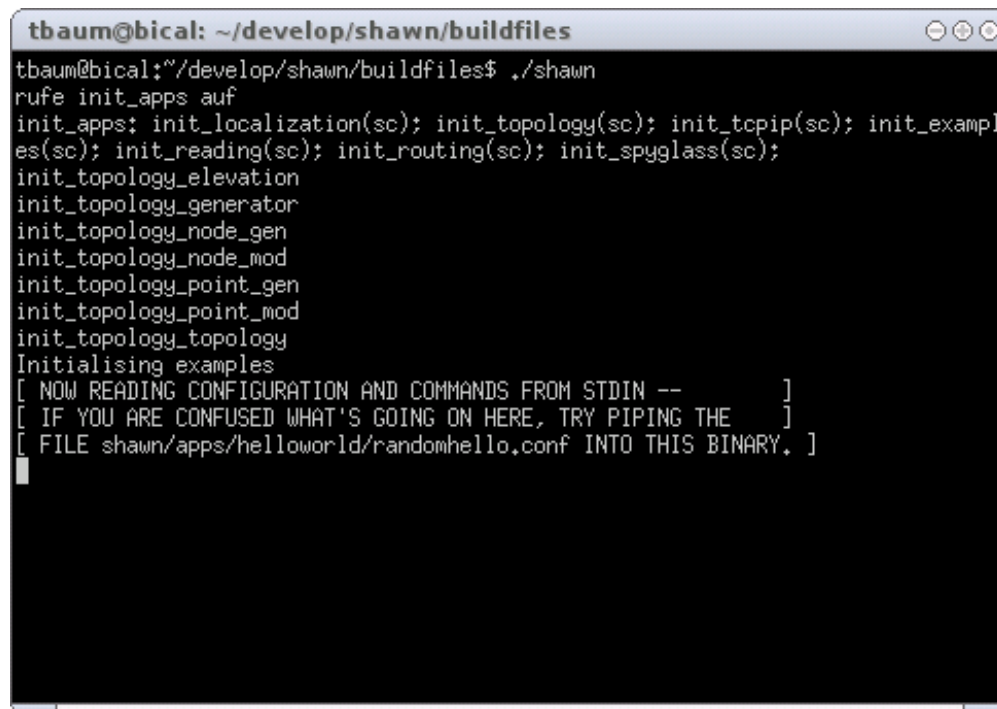


5. Κάνουμε διπλό click σε ένα από τα make target που έχουμε δημιουργήσει, για να γίνει compile ο Shawn.

4.4. Βασικά βήματα για την πραγματοποίηση εξομοίωσης στον Shawn

4.4.1. Παράθεση διαδοχικών βημάτων σε μια τυπική εξομοίωση

Αφού γίνει compile με επιτυχία ο Shawn θα προκύψει ένα εκτελέσιμο αρχείο εντός του φακέλου buildfiles (~/*shawn/buildfiles/*, στις εικόνες που ακολουθούν ο φάκελος *shawn* βρίσκεται εντός του φακέλου *develop*, πράγμα μη απαραίτητο). Για τους χρήστες των Windows θα είναι το *shawn.exe* και για τα Unix συστήματα θα είναι απλά *shawn*. Χρησιμοποιούμε οποιοδήποτε shell για να εκτελέσουμε το αρχείο (με την εντολή *./shawn*) οπότε και αντικρίζουμε την παρακάτω οθόνη. Εννοείται ότι βρισκόμαστε πάντα εντός του φακέλου *buildfiles*.



```
tbaum@bical: ~/develop/shawn/buildfiles
tbaum@bical:~/develop/shawn/buildfiles$ ./shawn
rufe init_apps auf
init_apps; init_localization(sc); init_topology(sc); init_tcpip(sc); init_examples(sc); init_reading(sc); init_routing(sc); init_spyglass(sc);
init_topology_elevation
init_topology_generator
init_topology_node_gen
init_topology_node_mod
init_topology_point_gen
init_topology_point_mod
init_topology_topology
Initialising examples
[ NOW READING CONFIGURATION AND COMMANDS FROM STDIN -- ]
[ IF YOU ARE CONFUSED WHAT'S GOING ON HERE, TRY PIPING THE ]
[ FILE shawn/apps/helloworld/randomhello.conf INTO THIS BINARY. ]
```

Η παραπάνω εικόνα ενδεχομένως να διαφέρει, ανάλογα με τις εφαρμογές (applications) που έχουμε ενεργοποιήσει προηγουμένως.

Είναι ανάγκη να ενεργοποιηθεί η εφαρμογή Examples (Παραδείγματα) κατά τη διάρκεια της διαμόρφωσης του *shawn* με τη χρήση του *cmake*. Με απλούστερα λόγια να επιλεγεί σε κατάσταση ON το *MODULE_APPS_EXAMPLES*. Η γραμμή *init_examples(sc)* εμφανίζεται μετά το *init_apps* επιβεβαιώνοντας ότι αυτή η εφαρμογή είναι ενεργοποιημένη.

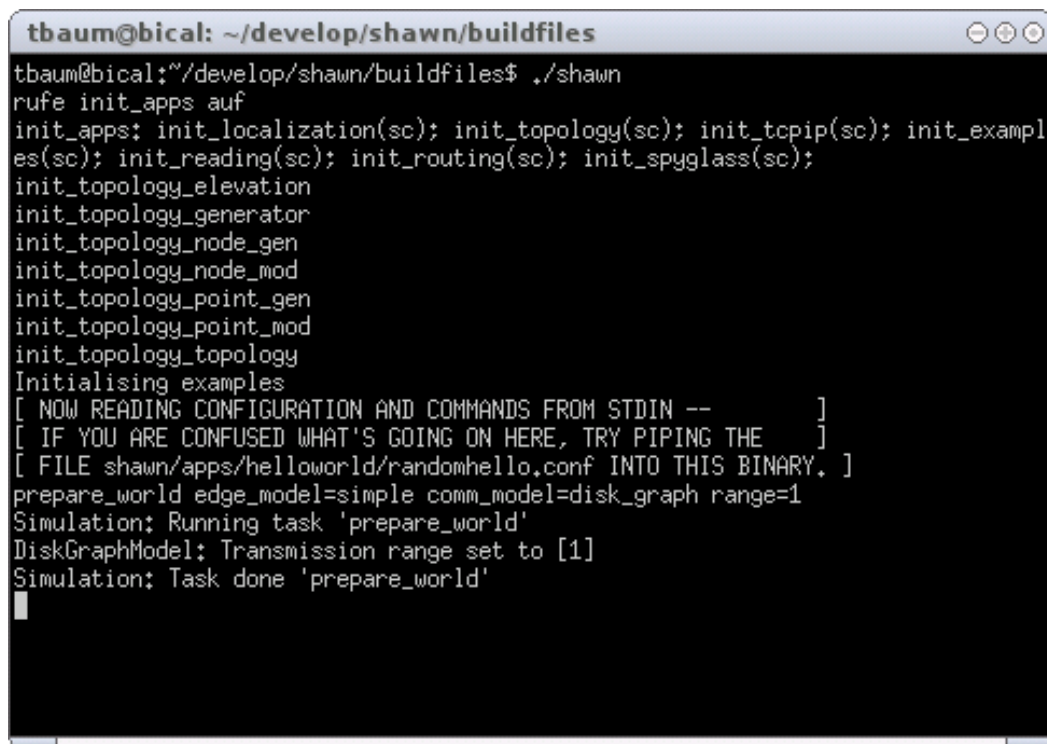
Αρχικά θα μελετηθεί το HelloWorld-processor παράδειγμα. Όπως υπονοεί και το όνομα είναι πολύ απλό. Μετά την εναρκτήρια διαδικασία (boot) κάθε κόμβος στέλνει ένα μήνυμα συγκεκριμένου είδους. Όλοι κόμβοι που το λαμβάνουν, εκτυπώνουν την ετικέτα αναγνώρισης (label) τόσο τη δικιά τους, όσο και αυτή του κόμβου αποστολέα. Εάν ένας κόμβος δεν δεχτεί κάποιο μήνυμα για πέντε γύρους, τότε θα εκτυπώσει τον αριθμό τον γνωστών γειτόνων (δλδ τους κόμβους από τους οποίους

έχει δεχτεί ένα μήνυμα) και τις ετικέτες αναγνώρισης (labels) αυτών των γειτόνων. Ακολούθως θα αυτο-απενεργοποιηθεί.

Σε πρώτο χρόνο πρέπει να ρυθμιστούν οι βασικές συνθήκες του περιβάλλοντος εξομοίωσης. Αυτό θα επιτευχθεί με την εισαγωγή της ακόλουθης γραμμής στην κονσόλα.

```
prepare_world edge_model=simple comm_model=disk_graph range=1
```

Η εντολή αυτή δημιουργεί στο Shawn ένα καινούριο «κόσμο» και τον παραμετροποιεί. Έτσι ρυθμίζονται τα Edge Model, Communication Model και η ακτίνα επικοινωνίας. Η οθόνη έχει ως εξής :



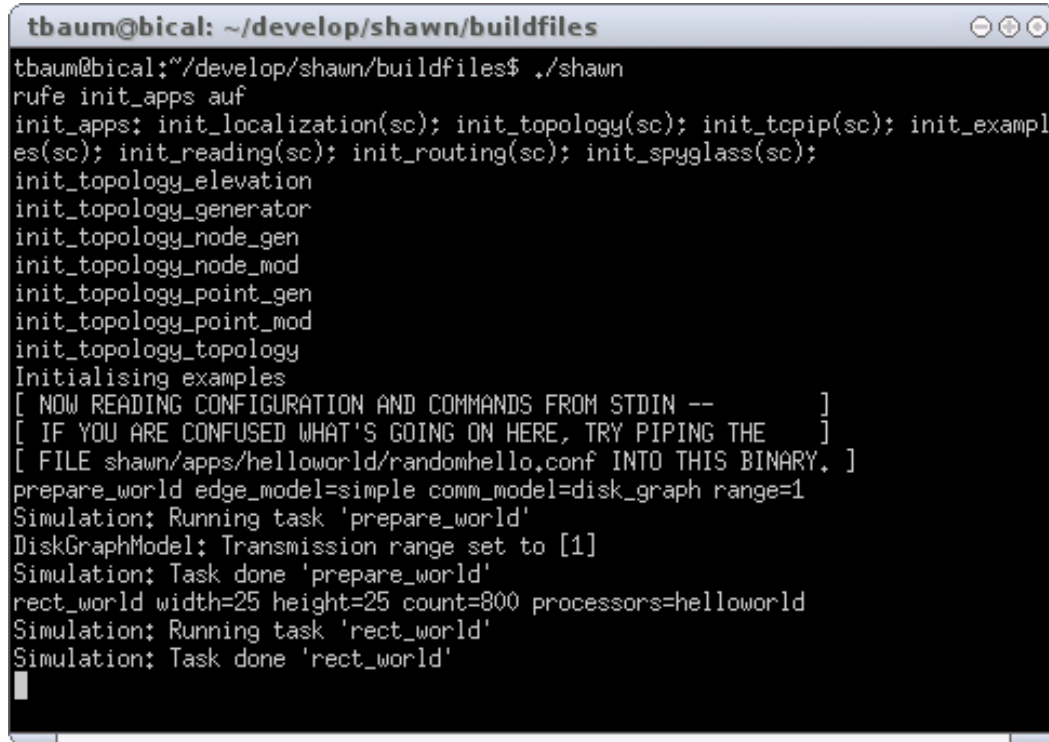
```
tbaum@bical: ~/develop/shawn/buildfiles
tbaum@bical:~/develop/shawn/buildfiles$ ./shawn
rufe init_apps auf
init_apps: init_localization(sc); init_topology(sc); init_tcpip(sc); init_examples(sc); init_reading(sc); init_routing(sc); init_spyglass(sc);
init_topology_elevation
init_topology_generator
init_topology_node_gen
init_topology_node_mod
init_topology_point_gen
init_topology_point_mod
init_topology_topology
Initialising examples
[ NOW READING CONFIGURATION AND COMMANDS FROM STDIN --          ]
[ IF YOU ARE CONFUSED WHAT'S GOING ON HERE, TRY PIPING THE     ]
[ FILE shawn/apps/helloworld/randomhello.conf INTO THIS BINARY. ]
prepare_world edge_model=simple comm_model=disk_graph range=1
Simulation: Running task 'prepare_world'
DiskGraphModel: Transmission range set to [1]
Simulation: Task done 'prepare_world'
```

Κατά την εκτέλεση ενός task ακολουθείται μία επαναλαμβανόμενη διαδικασία. Πρώτον εκτυπώνεται το : *Simulation: Running task 'name_of_task'*. Δεύτερον το παράγωγο (output) του task και τρίτον το εκτυπώνεται η γραμμή *Simulation: Task done 'name_of_task'*.

Αφού δημιουργηθεί με επιτυχία το περιβάλλον εξομοίωσης πρέπει να του εμφυσήσουμε «ζωή». Για αυτό το λόγο προσθέτουμε επεξεργαστές τύπου *HelloWorld*. Σημειωτέων ότι δεν δημιουργείται νέο περιβάλλον εξομοίωσης, απλά προστίθενται κόμβοι στην παρεχόμενη περιοχή του περιβάλλοντος εξομοίωσης. Η εντολή είναι :

```
rect_world width=25 height=25 count=800 processors=helloworld
```

Από αυτή την εντολή έχουμε την προσθήκη 800 επεξεργαστών τύπου *HelloWorld* σε ένα ορθογώνιο μέρος του κόσμου (με μέγεθος 25 επί 25). Όπως φαίνεται και στην εικόνα.



```
tbaum@bical: ~/develop/shawn/buildfiles
tbaum@bical:~/develop/shawn/buildfiles$ ./shawn
rufe init_apps auf
init_apps: init_localization(sc); init_topology(sc); init_tcpip(sc); init_exampl
es(sc); init_reading(sc); init_routing(sc); init_spyglass(sc);
init_topology_elevation
init_topology_generator
init_topology_node_gen
init_topology_node_mod
init_topology_point_gen
init_topology_point_mod
init_topology_topology
Initialising examples
[ NOW READING CONFIGURATION AND COMMANDS FROM STDIN --      ]
[ IF YOU ARE CONFUSED WHAT'S GOING ON HERE, TRY PIPING THE  ]
[ FILE shawn/apps/helloworld/randomhello.conf INTO THIS BINARY. ]
prepare_world edge_model=simple comm_model=disk_graph range=1
Simulation: Running task 'prepare_world'
DiskGraphModel: Transmission range set to [1]
Simulation: Task done 'prepare_world'
rect_world width=25 height=25 count=800 processors=helloworld
Simulation: Running task 'rect_world'
Simulation: Task done 'rect_world'
```

Τέλος η έναρξη της εξομοίωσης γίνεται με τη παρακάτω εντολή :

```
simulation max_iterations=10
```

Στο συγκεκριμένο παράδειγμα η παράμετρος *max_iterations* δεν είναι απαραίτητη, δεδομένου του ότι οι επεξεργαστές απενεργοποιούν τους εαυτούς τους με τη πάροδο μερικών γύρων (εδώ οι γύροι είναι 6). Γενικά όμως αυτή η παράμετρος συνίσταται να ορίζεται για την αποφυγή αέναων γύρων εκτέλεσης. Κατά την εξομοίωση η οθόνη μοιάζει ως εξής :

```
tbaum@bical: ~/develop/shawn/buildfiles
ID 'v341-rqBzTB-hF' GOT HELLO FROM 'v728-rqBzTB-wL'
ID 'v662-rqBzTB-sK' GOT HELLO FROM 'v728-rqBzTB-wL'
ID 'v395-rqBzTB-ZG' GOT HELLO FROM 'v729-rqBzTB-xL'
ID 'v456-rqBzTB-YH' GOT HELLO FROM 'v729-rqBzTB-xL'
ID 'v530-rqBzTB-kI' GOT HELLO FROM 'v729-rqBzTB-xL'
ID 'v637-rqBzTB-TK' GOT HELLO FROM 'v729-rqBzTB-xL'
ID 'v236-rqBzTB-OD' GOT HELLO FROM 'v730-rqBzTB-yL'
ID 'v592-rqBzTB-kJ' GOT HELLO FROM 'v730-rqBzTB-yL'
ID 'v93-rqBzTB-hB' GOT HELLO FROM 'v731-rqBzTB-zL'
ID 'v188-rqBzTB-ED' GOT HELLO FROM 'v731-rqBzTB-zL'
ID 'v578-rqBzTB-WJ' GOT HELLO FROM 'v731-rqBzTB-zL'
ID 'v647-rqBzTB-dK' GOT HELLO FROM 'v731-rqBzTB-zL'
ID 'v262-rqBzTB-QE' GOT HELLO FROM 'v732-rqBzTB-OL'
ID 'v359-rqBzTB-zF' GOT HELLO FROM 'v732-rqBzTB-OL'
ID 'v562-rqBzTB-GJ' GOT HELLO FROM 'v732-rqBzTB-OL'
ID 'v759-rqBzTB-RM' GOT HELLO FROM 'v732-rqBzTB-OL'
ID 'v584-rqBzTB-cJ' GOT HELLO FROM 'v733-rqBzTB-1L'
ID 'v782-rqBzTB-oM' GOT HELLO FROM 'v733-rqBzTB-1L'
ID 'v799-rqBzTB-5M' GOT HELLO FROM 'v733-rqBzTB-1L'
ID 'v127-rqBzTB-FC' GOT HELLO FROM 'v734-rqBzTB-2L'
ID 'v370-rqBzTB-AG' GOT HELLO FROM 'v734-rqBzTB-2L'
ID 'v74-rqBzTB-OB' GOT HELLO FROM 'v735-rqBzTB-3L'
ID 'v121-rqBzTB-9B' GOT HELLO FROM 'v735-rqBzTB-3L'
ID 'v603-rqBzTB-vJ' GOT HELLO FROM 'v736-rqBzTB-4L'
```

```
tbaum@bical: ~/develop/shawn/buildfiles
v788-rqBzTB-uM'
ID 789 DONE: 7 neighbours: 'v47-rqBzTB-x', 'v84-rqBzTB-YB', 'v163-rqBzTB-pC', 'v
352-rqBzTB-sF', 'v465-rqBzTB-hH', 'v663-rqBzTB-tK', 'v789-rqBzTB-vM'
ID 790 DONE: 3 neighbours: 'v558-rqBzTB-CJ', 'v631-rqBzTB-NK', 'v790-rqBzTB-wM'
ID 791 DONE: 5 neighbours: 'v37-rqBzTB-n', 'v101-rqBzTB-pB', 'v278-rqBzTB-gE', '
v474-rqBzTB-qH', 'v791-rqBzTB-xM'
ID 792 DONE: 3 neighbours: 'v221-rqBzTB-ID', 'v786-rqBzTB-sM', 'v792-rqBzTB-yM'
ID 793 DONE: 5 neighbours: 'v475-rqBzTB-rH', 'v531-rqBzTB-1I', 'v553-rqBzTB-7I',
'v627-rqBzTB-JK', 'v793-rqBzTB-zM'
ID 794 DONE: 1 neighbours: 'v794-rqBzTB-OM'
ID 795 DONE: 6 neighbours: 'v49-rqBzTB-z', 'v140-rqBzTB-SC', 'v266-rqBzTB-UE', '
v348-rqBzTB-oF', 'v636-rqBzTB-SK', 'v795-rqBzTB-1M'
ID 796 DONE: 7 neighbours: 'v52-rqBzTB-2', 'v60-rqBzTB-AB', 'v114-rqBzTB-2B', 'v
232-rqBzTB-wD', 'v246-rqBzTB-AE', 'v523-rqBzTB-dI', 'v796-rqBzTB-2M'
ID 797 DONE: 3 neighbours: 'v299-rqBzTB-1E', 'v428-rqBzTB-6G', 'v797-rqBzTB-3M'
ID 798 DONE: 6 neighbours: 'v158-rqBzTB-kC', 'v245-rqBzTB-9D', 'v293-rqBzTB-vE',
'v306-rqBzTB-8E', 'v375-rqBzTB-FC', 'v798-rqBzTB-4M'
ID 799 DONE: 6 neighbours: 'v141-rqBzTB-TC', 'v584-rqBzTB-cJ', 'v677-rqBzTB-7K',
'v733-rqBzTB-1L', 'v782-rqBzTB-oM', 'v799-rqBzTB-5M'
----- DONE ITERATION 6
[ 0 active, 0 sleeping, 800 inactive ]

Simulation: Task done 'simulation'
```

Εδώ οι κόμβοι τυπώνουν αυτούς από τους οποίους έχουν δεχτεί μήνυμα. Παραδείγματος χάριν στην πρώτη γραμμή του παραπάνω σχήματος ο κόμβος 341 δέχεται μήνυμα από τον κόμβο 728. Στη συνέχεια όλοι κόμβοι τυπώνουν το μέγεθος της γειτονίας τους. Αυτό ακολουθείται από τις ετικέτες αναγνώρισης (labels) των γειτόνων. Ο κόμβος 799 έχει 6 γείτονες με πρώτο τον τον κόμβο 141 όπως δείχνει το επόμενο σχήμα. Η εξομοίωση τελειώνει στον 6 γύρο σε αντίθεση με τους 10 γύρους που έχουμε οριστεί ως μέγιστο αριθμό επαναλήψεων. Αυτό οφείλεται στο ότι οι κόμβοι αυτο-απενεργοποιούνται. Αμέσως μετά τη γραμμή που δείχνει ότι η εξομοίωση έλαβε τέλος στον έκτο γύρο τυπώνονται κάποιες πληροφορίες. Αυτές είναι ο αριθμός των κόμβων που είναι ενεργοί, σε «ύπνωση» (sleeping) και απενεργοί. Αν το άθροισμα των ενεργών και των εν υπνώσει είναι μηδεν τότε η εξομοίωση έχει τελειώσει.

4.4.2. Χρήση αρχείου διαμόρφωσης (Configuration File)

Εύκολα γίνεται αντιληπτό, ότι το να γράφει κανείς αυτές τις γραμμές κάθε φορά που ξεκινά μία εξομοίωση είναι μονότονο και κουραστικό. Προς αποφυγή του παραπάνω έχουμε την δυνατότητα να δημιουργήσουμε ένα αρχείο, στο οποίο θα εισάγουμε όλες της γραμμές που θα θέλουμε να περάσουμε στον Shawn. Στη συνέχεια θα καλούμε το αρχείο αυτό.

Οπότε δημιουργούμε ένα αρχείο με το όνομα my_conf (εντός φακέλου buildfiles) το οποίο περιέχει τα παρακάτω :

```
# This is a comment that is not processed. Comments *must* be
# written in a separate line, and are *not* allowed to be
# attached to a line that contains a command for Shawn.
prepare_world    edge_model=simple    comm_model=disk_graph    range=1
rect_world      width=25    height=25    count=800    processors=helloworld
simulation max_iterations=10
```

Γενικά υπάρχουν λίγοι κανόνες που διέπουν αυτά τα αρχεία. Κάθε γραμμή που ξεκινά με δίεση (#) αποτελεί γραμμή σχολίων και δεν περνά στον parser. Ένα σχόλιο δεν επιτρέπεται να προσκολλάται σε μια έγκυρη εντολή του Shawn. Ακόμα μια παράμετρος μπορεί να γίνει κοινή (global) άμα γράψουμε key = value. Εν τωιαύτη περιπτώσει το key μπορεί να προσπελαστεί από όλα τα ακολουθούμενα tasks. Τέλος μπορούμε να καλέσουμε ένα task απλά γράφοντας το όνομα του. Για παράδειγμα το simulation καλεί το task simulation στο οποίο αντιστοιχεί. Ένας τέτοιο task μπορεί επίσης να ακολουθείται από μια παράμετρο. Ενδεχομένως έχουμε ανάθεση στη παράμετρο max_iterations αφού καλεσθεί το simulation. Σε αυτή την περίπτωση η ανάθεση έχει αυστηρά τοπική ισχύ και αφορά μόνο το αντίστοιχο task. Τα επόμενα task δεν έχουν πρόσβαση σε αυτήν την παράμετρο (για να είναι global, η ανάθεση πρέπει να γραφτεί σε ξεχωριστή γραμμή όπως περιγράφεται προηγουμένως).

Για να εισαχθεί στον Shawn το αρχείο χρειάζεται η παράμετρος `-f`. Οπότε θα γράψουμε στην κονσόλα :

```
./shawn -f my_conf
```

4.4.3. Χρησιμοποίηση πολλαπλών configuration αρχείων

Αν τα configuration αρχεία αποκτούν εκτεταμένη πολυπλοκότητα δίνονται να χωριστούν σε περισσότερα αρχεία. Έτσι έχουμε ένα κύριο (master) αρχείο, το οποίο και περνάμε στον Shawn και περισσότερα δευτερεύοντα (slaves), που περιέχονται στο κύριο. Για να γίνει η συμπερίληψη αυτή των δευτερευόντων αρχείων, πρέπει να εμπεριέχει τη ακόλουθη γραμμή :

```
# some commands for Shawn ...  
include_file slave_file_to_include  
# more commands for Shawn ...
```

Το `include_file` βρίσκεται ακριβώς στην αρχή της κατάλληλης γραμμής του configuration αρχείου. Πολλαπλές συμπερίληψεις δεν είναι εφικτές. Δηλαδή ένα αρχείο το οποίο έχει γίνει συμπερίληψη δεν μπορεί να περιλαμβάνει κάποιο άλλο.

5. Κώδικας του Directed Diffusion στο Shawn

5.1 Σχολιασμός Κώδικα

Στο αρχείο `dd_processor.h` περιέχεται η δομή `interest`. Με τον όρο `interest` στο Directed Diffusion αναφερόμαστε στην ζητούμενη πληροφορία από την καταβόθρα (sink). Αυτή η δομή είναι abstract και περιλαμβάνει όλες τις ιδιότητες που αποτελούν ένα `interest`. Αυτό σημαίνει ότι μπορούμε να την επαυξήσουμε ώστε να προσαρμόζεται στα διάφορα σενάρια.

Listing 5.1: Δομή `interest`

```
struct interest {
    friend bool operator <(const interest &id1, const interest &id2) {
        if ((id1.type == id2.type) && (id1.value == id2.value)
            && (id1.source_id == id2.source_id)) {

            return false;
        }

        return true;
    }
    string type;
    string value;
    int source_id;
    int expires_at;
    int timestamp;
    int interval;
    int hops;
    int previous_node;
    int next_node;
    bool alive;
};
```

Η σημασία των μεταβλητών έχει αναλυθεί στο προηγούμενο κεφάλαιο όπου και παρουσιάστηκε θεωρητικά το πρωτόκολλο. Εκτός από τη μεταβλητή `alive`, που περιγράφει το μονοπάτι από το οποίο φτάνει πρώτα ένα `interest`. Τα υπόλοιπα μονοπάτια θα τίθενται ως `back up`.

Επίσης στο αρχείο ορίζεται το `map interests`. Αυτό περιλαμβάνει ένα `string` που περιέχει το `type` και το `value` του `interest`, το οποίο είναι `unique key`. Κάθε `unique key` έχει σαν `value` ένα άλλο `map`. Αυτό έχει ως `unique key` το `identifier` του sink που με τη σειρά του έχει σαν `value` μια λίστα από `interests`.

```
map<string, map<int, list<struct interest> > > interests ;
```

Στον επόμενο πίνακα γίνεται μια προσπάθεια απεικόνισης του `map interests_`

Node Type & Value	Sink id	Interests
"node_name&v5-000000-G"	0	C ₀ -ALIVE
	0	C ₀ -backup.1

C0-backup.2
C0-backup.3
C1-ALIVE
C1-backup.1
C1-backup.2
C1-backup.3

Στο αρχείο dd_processor.cpp συναντάμε τη συνάρτηση

Listing 5.2: Η συνάρτηση bool process_message(const shawn::ConstMessageHandle& mh)

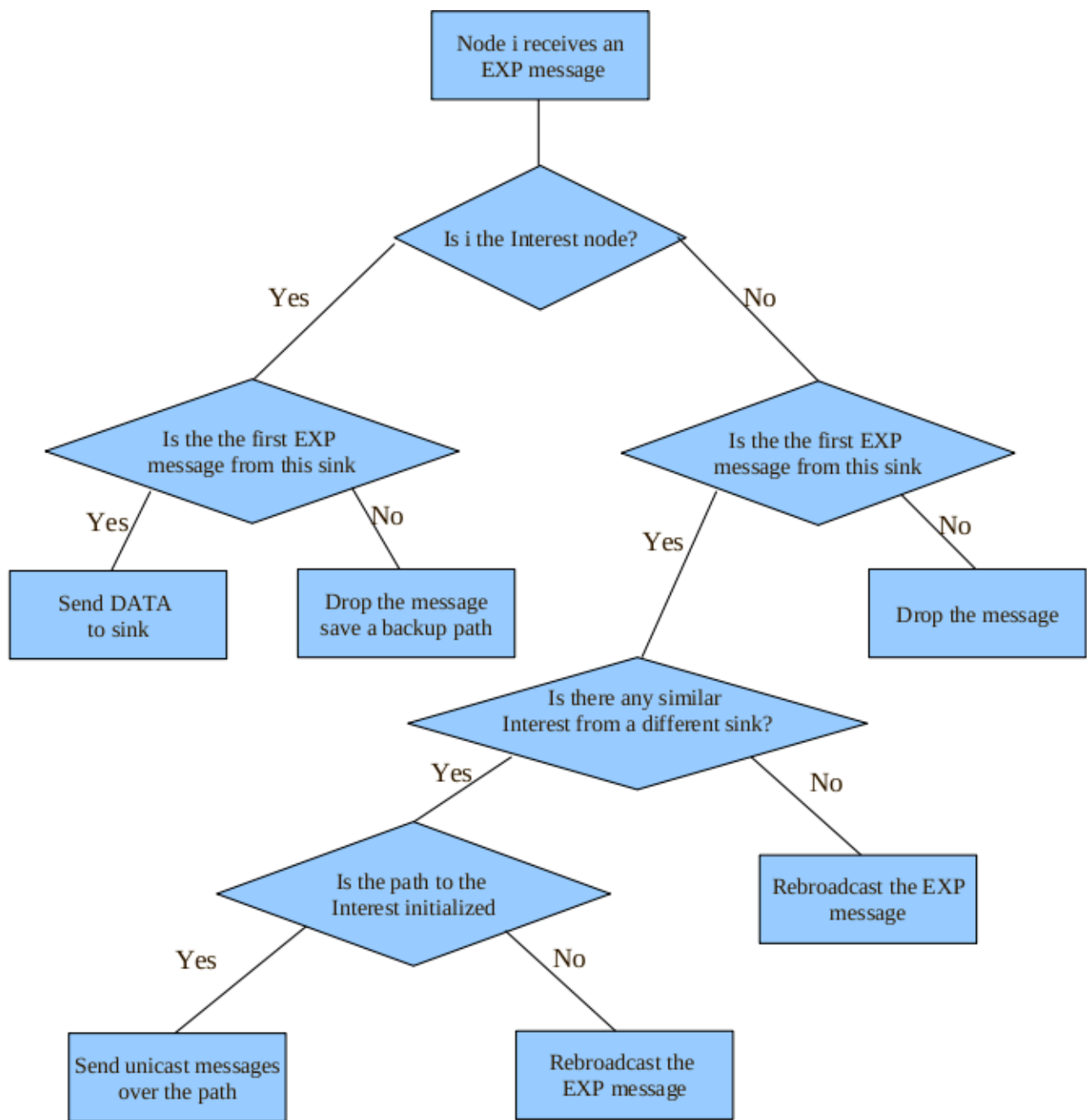
```
bool DdProcessor::process_message(const shawn::ConstMessageHandle& mh) throw () {
    const DdMessage* ddmsg = dynamic_cast<const DdMessage*> (mh.get());

    if (ddmsg != NULL && owner() != ddmsg->source()) {
        int msg_type = ddmsg->msg_type;
        switch (msg_type) {

            case EXP: {
                proc_exp(ddmsg);
                break;
            }
            case RES:
                proc_res(ddmsg);
                break;
            default:
                cout << owner().id() << ": Received UNRECOGNIZED message type "
                     << msg_type << endl;
                break;
        }
    }
    return true;
}
```

Είναι φανερό ότι έχουμε δύο τύπους μηνυμάτων. Το πρώτο το EXP, αφορά τα αρχικά μηνύματα που στέλνονται προς αναζήτηση της πληροφορίας. Το δεύτερο, το RES είναι τα response μηνύματα που στέλνονται προς την καταβόθρα, μόλις κάνει match το interest. Τέλος οποιοδήποτε άλλο μήνυμα αγνοείται.

Αντίστοιχα καλούνται και οι συναρτήσεις proc_exp(ddmsg) και proc_res(ddmsg). Όπως βλέπουμε από το σώμα της proc_exp(ddmsg) ένα interest αντιμετωπίζεται ως εξής:



Σχήμα 5.1: Δενδρική αναπαράσταση της αντιμετώπισης του interests - συνάρτηση proc_exp().

Ο κώδικας του παραπάνω κώδικα είναι ο εξής :

Listing 5.3: Η συνάρτηση void proc_exp(const DdMessage*)

```

void DdProcessor::proc_exp(const DdMessage* ddmsg) {
    //Destination Found
    if (ddmsg->getValue() == owner().label()) {
        cout << owner().id() << " : Exp msg received to dest with sink "
             << ddmsg->getSource_id() << " from " << ddmsg->source().id()
             << endl;
        if (!interest_exists(ddmsg)) {
  
```



```

//Send response message

//Add full interst struct in interests map.
interest C;

C.type = ddmsg->getType();
C.value = ddmsg->getValue();
C.source_id = ddmsg->getSource_id();
C.expires_at = ddmsg->getExpires_at();
C.timestamp = ddmsg->getTimestamp();
C.interval = ddmsg->getInterval();
C.hops = ddmsg->getHops();
C.next_node = ddmsg->source().id();
C.previous_node = owner().id();
C.alive = true;
interests_[ddmsg->getType()+"&" + ddmsg->getValue()]
            [ddmsg->getSource_id()].push_back(C);

//Sending response Messages to sink.
DdMessage* dd = new DdMessage();
dd->msg_type = RES;
dd->setType(ddmsg->getType());
dd->setValue(ddmsg->getValue());
dd->setExpires_at(ddmsg->getExpires_at());
dd->setTimestamp(ddmsg->getTimestamp());
dd->setSource_id(ddmsg->getSource_id());
dd->setNext_node(C.next_node);
for (int i = 0; i < C.interval; i++) {
    send(dd);
}
} else {
//Add backup paths
interest C;
C.type = ddmsg->getType();
C.value = ddmsg->getValue();
C.source_id = ddmsg->getSource_id();
C.expires_at = ddmsg->getExpires_at();
C.timestamp = ddmsg->getTimestamp();
C.interval = ddmsg->getInterval();
C.hops = ddmsg->getHops();
C.next_node = ddmsg->source().id();
C.previous_node = ddmsg->source().id();
C.alive = false;
interests_[ddmsg->getType() + "&" + ddmsg->getValue()]
            [ddmsg->getSource_id()].push_back(C);
cout << "\tAdd backup path" << endl;
}
} else {
cout << owner().id() << " : " << " EXP msg received from "
    << ddmsg->source().id() << " sink " << ddmsg->getSource_id()
    << endl;

//check if this is a new interest
if (!interest_exists(ddmsg)) {

    if (similar_interest_exists(ddmsg)) {
        //There is an existing path
        if (ddmsg->getNext_node() == -1 || ddmsg->getNext_node()
            == owner().id()) {

            //Find the similar path and send UNICAST messages.
            map<string, map<int, list<struct interest>>> >>::iterator
                vIter =
            interests_.find(map<string, map<int, list<
                struct interest> >>::key_type(
                ddmsg->getType() + "&" + ddmsg->getValue()));
            if (vIter != interests_.end()) {
                map<int, list<struct interest> >
                    ::iterator alter;
                for (aIter = vIter->second.begin();

```

```

aIter!= vIter->second.end(); aIter++) {
    if (aIter->first != ddmsg->getSource_id()) {
        list<struct interest>::
            iterator intIter;
        for (intIter = interests_[ddmsg->getType()
            +"&"+ddmsg->getValue()]
            [aIter->first].begin(); intIter
            != interests_[ddmsg->getType() + "&"
            + ddmsg->getValue()][aIter->first].end();
            intIter++) {
            if (intIter->alive == true &&
                intIter->previous_node != -1) {
                cout << "Send unicast msg to :" <<
                    intIter->previous_node<< endl;

                interest C;
                C.type = ddmsg->getType();
                C.value = ddmsg->getValue();
                C.source_id = ddmsg->getSource_id();
                C.expires_at = ddmsg->getExpires_at();
                C.timestamp = ddmsg->getTimestamp();
                C.interval = ddmsg->getInterval();
                C.hops = ddmsg->getHops();
                C.next_node = ddmsg->source().id();
                C.previous_node
                    = intIter->previous_node;
                C.alive = true;

                interests_[ddmsg->getType() + "&" +
                    ddmsg->getValue()][ddmsg
                    ->getSource_id()].push_back(C);

                //Rebroadcasting the interest request.
                DdMessage* dd = new DdMessage();
                dd->msg_type = EXP;
                dd->setType(ddmsg->getType());
                dd->setValue(ddmsg->getValue());
                dd->setExpires_at(
                    ddmsg->getExpires_at());
                dd->setTimestamp(ddmsg->getTimestamp());
                dd->setHops(ddmsg->getHops() + 1);
                dd->setInterval(ddmsg->getInterval());
                dd->setSource_id(ddmsg
                    ->getSource_id());
                dd->setPrevious_node(ddmsg
                    ->source().id());
                dd->setNext_node(
                    intIter->previous_node);
                send(dd);
                break;
            }
        }
    }
}

}

}

} else {
    //DROP the message.
    cout << "\tEXP msg DROPPED" << endl;
}

} else {
    //Save the Interest and re-broadcast to find the path.
    interest C;
    C.type = ddmsg->getType();
    C.value = ddmsg->getValue();
    C.source_id = ddmsg->getSource_id();
    C.expires_at = ddmsg->getExpires_at();
    C.timestamp = ddmsg->getTimestamp();
    C.interval = ddmsg->getInterval();

```

```

C.hops = ddmsg->getHops();
C.next_node = ddmsg->source().id();
C.previous_node = -1;
C.alive = true;
interests_[ddmsg->getType()
            + "&" + ddmsg->getValue()][ddmsg->getSource_id()].push_back(C);

//Rebroadcasting the interest request.
DdMessage* dd = new DdMessage();
dd->msg_type = EXP;
dd->setType(ddmsg->getType());
dd->setValue(ddmsg->getValue());
dd->setExpires_at(ddmsg->getExpires_at());
dd->setTimestamp(ddmsg->getTimestamp());
dd->setHops(ddmsg->getHops() + 1);
dd->setInterval(ddmsg->getInterval());
dd->setSource_id(ddmsg->getSource_id());
dd->setPrevious_node(ddmsg->source().id());
send(dd);
}
} else {
//Drop the msg
cout << "\tEXP msg DROPPED" << endl;
}
}
}

```

Στη συνέχεια έχουμε την συνάρτηση `proc_res()` η οποία χειρίζεται τα data messages. Η συνάρτηση αυτή ελέγχει αν τα δεδομένα έχουν φτάσει στην καταβόθρα ή αν βρίσκονται σε κάποιο ενδιάμεσο κόμβο. Στην πρώτη περίπτωση απλά τυπώνει ότι το μήνυμα έφτασε στο sink και ανανεώνει την τιμή του `previous_node` στην δομή του συγκεκριμένου Interest. Στη δεύτερη περίπτωση όπου τα δεδομένα βρίσκονται σε κάποιο ενδιάμεσο κόμβο απλά προωθούνται στο `nextNode` του συγκεκριμένου Interest. Εδώ να σημειωθεί ότι καθώς γίνεται Broadcast το EXP μήνυμα για ένα Interest όλοι οι κόμβοι μόλις λάβουν ένα EXP μήνυμα αποθήκευουν το συγκεκριμένο Interest και θέτουν επίσης την μεταβλητή `next_node` ίση με το `node_id` του κομβου από τον οποίο έλαβαν το συγκεκριμένο EXP. Η τιμή του `previous_node` αρχικοποιείται και είναι ίση με -1. Έτσι όλοι οι κομβοι γνωρίζουν ποιο είναι το `next_node` στο οποίο πρέπει να στείλουν έτσι ώστε να επικοινωνήσουν με το sink του κάθε Interest. Μόλις λάβουν ένα RES μήνυμα για κάποιο συγκεκριμένο Interest ανανεώνουν την τιμή του `previous_node` από -1 και την θέτουν ίση με το `id` του κόμβου από τον οποίο έλαβαν το RES μήνυμα. Πλέον εκτός από το πως θα φτασουν στο sink γνωρίζουν και που πρέπει να στείλουν αν θέλουν να επικοινωνήσουν με κάποιο συγκεκριμένο Interest. Η κάθε καταχώρηση θα είναι της μορφής

Node Type & Value	Sink id	C ₁
		C ₂
		...

Listing 5.4: Η συνάρτηση `void proc_res(const DdMessage*)`

```

void DdProcessor::proc_res(const DdMessage* ddmsg) {
    if (ddmsg->getNext_node() == owner().id() && ddmsg->getSource_id()
        == owner().id()) {
        cout << owner().id() << " Message Delivered to source" << endl;
        //Update interest struct
        update_interest(ddmsg);
    }
}

```

```

} else if (ddmsg->getNext_node() == owner().id()) {
    cout << owner().id() << " : " << " Received RES msg with sink id "
        << ddmsg->getSource_id() << " from " << ddmsg->source().id()
        << endl;

    //Update interest struct
    update_interest(ddmsg);
    list<struct interest> tmp = interests_[ddmsg->getType() + "&"
        + ddmsg->getValue()][ddmsg->getSource_id()];
    list<struct interest>::iterator aIter;

    for (aIter = tmp.begin(); aIter != tmp.end(); ++aIter) {
        if (aIter->alive == true) {
            //Rebroadcasting the interest request.
            DdMessage* dd = new DdMessage();
            dd->msg_type = RES;
            dd->setType(ddmsg->getType());
            dd->setValue(ddmsg->getValue());
            dd->setExpires_at(ddmsg->getExpires_at());
            dd->setTimestamp(ddmsg->getTimestamp());
            dd->setHops(ddmsg->getHops() + 1);
            dd->setInterval(ddmsg->getInterval());
            dd->setSource_id(ddmsg->getSource_id());
            dd->setNext_node(aIter->next_node);
            send(dd);
            break;
        }
    }
}
}
}

```

Η συνάρτηση `similar_interest_exists()` αναζητά ίδια interests αλλά για διαφορετικά sink. Δηλαδή στον παραπάνω πίνακα διαφέρει η δεύτερη γραμμή (Sink id). Αυτό σημαίνει ότι κάποιος άλλος sink έχει αναζητήσει το συγκεκριμένο Interest και αν η τιμή του `previous_node` δεν είναι ίση με -1 σημαίνει ότι υπάρχει μονοπάτι για να φτάσει το καινούριο EXP μήνυμα στο Interest. Επομένως αντί να κάνει rebroadcast ο κόμβος το EXP μήνυμα το στέλνει μέσω του μονοπατιού των `previous_nodes` κατευθείαν στο Interest. Έτσι αποφεύγεται το broadcast.

Listing 5.5: Η συνάρτηση `bool similar_interest_exists(const DdMessage*)`

```

bool DdProcessor::similar_interest_exists(const DdMessage* ddmsg) {
    map<string, map<int, list<struct interest> > >::iterator vIter =
        interests_.find(
            map<string, map<int, list<struct interest> > >
            >::key_type(
                ddmsg->getType() + "&" + ddmsg->
                >getValue()));
    if (vIter != interests_.end()) {
        map<int, list<struct interest> >::iterator aIter;
        for (aIter = vIter->second.begin(); aIter != vIter->second.end(); aIter++)
        {
            if (aIter->first != ddmsg->getSource_id()) {
                list<struct interest>::iterator intIter;
                for (intIter = interests_[ddmsg->getType() + "&"
                    + ddmsg->getValue()][aIter->first].begin();
                    intIter
                        != interests_[ddmsg->getType() + "&"
                            + ddmsg->getValue()][aIter->
                            >first].end(); intIter++) {
                    if (intIter->alive == true && intIter->previous_node
                        != -1) {
                        return true;
                    }
                }
            }
        }
    }
}

```

```

    }
}
return false;
}

```

Κάθε Interest έχει ένα expiration_time. Η συνάρτηση send_pending_inter_req() ελέγχει το expiration time του κάθε Interest και εφόσον δεν έχει λήξει στέλνει στο sink τα δεδομένα που ζήτησε. Το interval του κάθε Interest δηλώνει πόσα μηνύματα ανά γύρο θα αποστέλονται από το Interest προς το Sink. Επομένως η συγκεκριμένη συνάρτηση δεν στέλνει απλά δεδομένα από το Interest προς το Sink αλλά το κάνει και σύμφωνα με το Interval του κάθε Interest.

Listing 5.6: Η συνάρτηση void send_pending_inter_req()

```

Void DdProcessor::send_pending_inter_req() {
    map<string, map<int, list<struct interest> > >::iterator vIter;
    map<int, list<struct interest> >::iterator interIter;
    list<struct interest>::iterator interests;
    for (vIter = interests_.begin(); vIter != interests_.end(); vIter++) {
        for (interIter = vIter->second.begin(); interIter
            != vIter->second.end(); interIter++) {
            for (interests = interIter->second.begin(); interests
                != interIter->second.end(); interests++) {
                if (interests->alive == true && interests->expires_at > 0
                    && interests->value == owner().label()) {
                    interests->expires_at--;
                    cout << owner().id() << " : " << "Send res msg" <<
endl;
                    break;
                }
            }
        }
    }
}
}
}
}

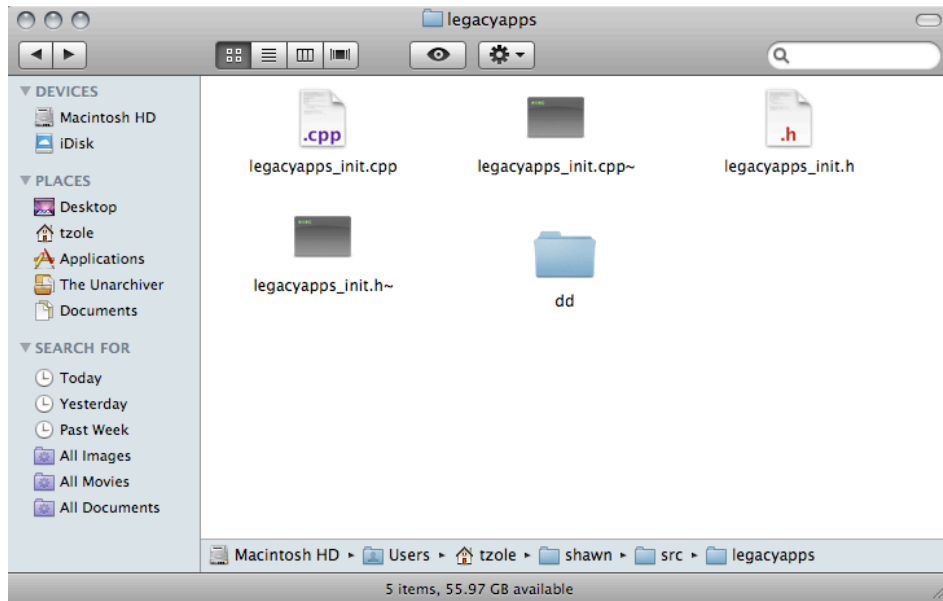
```

Στη work() τώρα έχουμε τη δομή interest C στην οποία περνάμε όλες της μεταβλητές της και στην συνέχεια στέλνουμε στο δίκτυο. Ας σημειωθεί ότι για το C έχω :
Source_id = sink, next_node=sink, previous_node =-1, alive =true

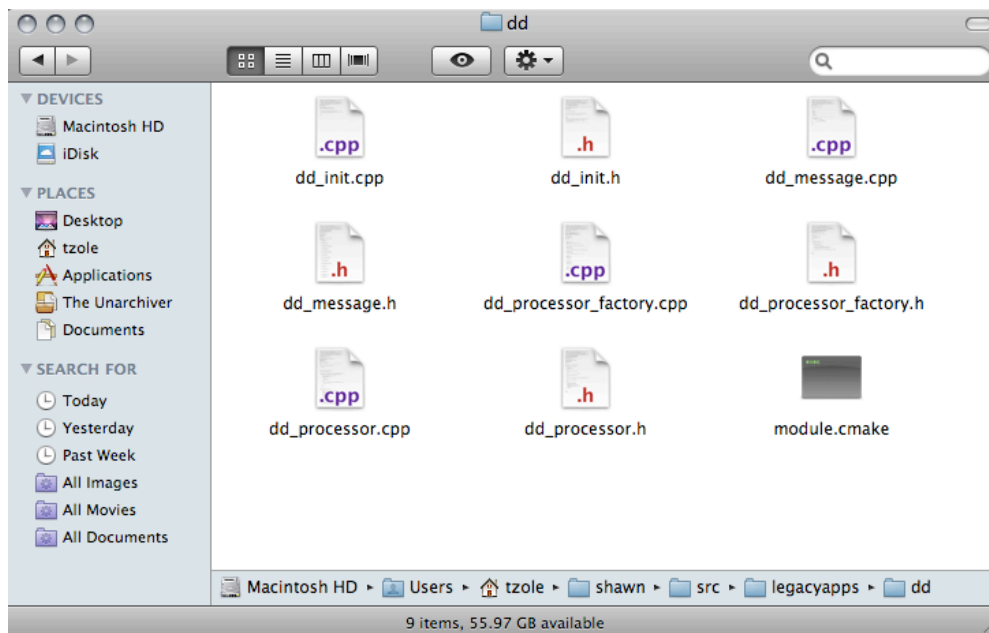
5.2 Εισαγωγή του Directed Diffusion στο Shawn

Για να συμπεριλάβει κανείς το πρωτόκολλο του Directed Diffusion στο Shawn πρέπει να ακολουθήσει την ακόλουθη διαδικασία. Αυτή σημειώνει ορισμένες παραλλαγές, αφού δεν περιέχεται η υλοποίηση του Directed Diffusion στον βασικό κορμό του Shawn.

Εντός του φακέλου shawn δημιουργείται ένας φάκελος legacyapps. Αυτός ο φάκελος περιέχει όλες τις υλοποιήσεις που μπορεί να αναπτύξει ο καθένας. Οι υλοποιήσεις θα κάνουν χρήση πολλών υπαρχόντων βασικών συναρτήσεων του Shawn ή άλλων χαρακτηριστικών του. Εν προκύψον το μονοπάτι είναι ~/shawn/src/legacyapps. Όπως φαίνεται και στην εικόνα ο φάκελος legacyapps έχει εκτός από το πρωτόκολλο και δύο αρχεία. Τα legacyapps_init.cpp και legacyapps_int.h. Αυτά προήλθαν από τα προυπάρχοντα αρχεία apps_init.cpp και apps_int.h μόνο που αντικαταστάθηκε το apps με το legacyapps και στο εσωτερικό των αρχείων.



Εντός του legacyapps εισάγεται ο φάκελος dd. Ο φάκελος περιέχει τα αρχεία της φωτογραφίας.



Στη συνέχεια καλούμε στο shawn/buildfiles cmake ../src. Στη γραμμή LEGACYAPPS_PATH πατάμε Enter και εισάγουμε ολόκληρο το μονοπάτι του νεο δημιουργηθέντος φακέλου legacyapps. Εδώ θα είναι /Users/tzole/shawn/src/legacyapps. Μετά πατούμε Enter. Αμέσως μετά πατούμε c και ύστερα q. Εδώ πρέπει να αναφερθεί ότι CONFIGURE_LEGACYAPPS είναι ON.

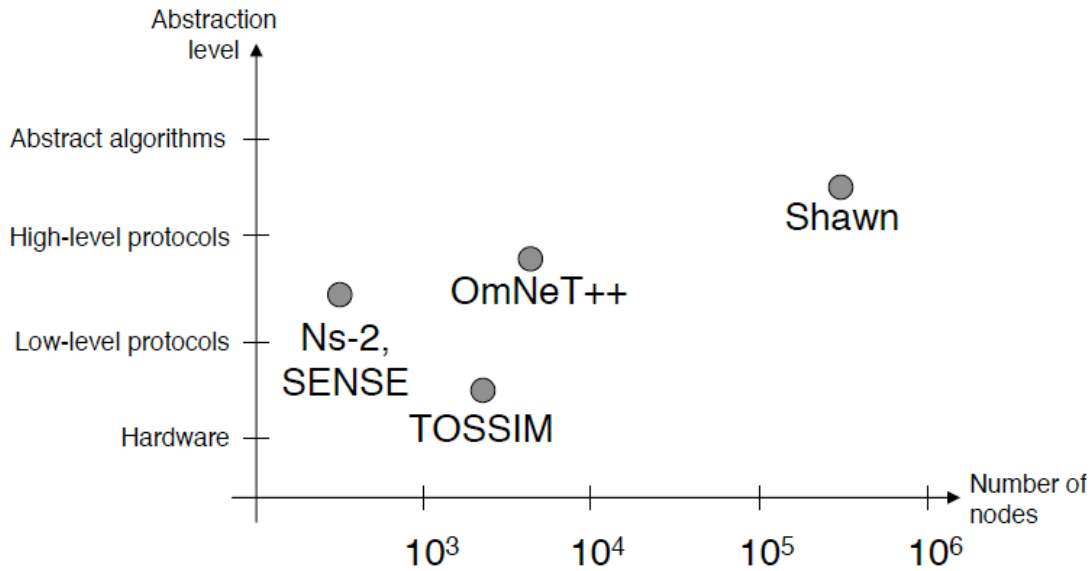
Για τη δημιουργία του module τοποθετούμε το αρχείο dd.conf που περιέχει τα εξής:

```
prepare_world edge_model=simple comm_model=disk_graph range=3  
load_world file=/Users/tzole/shawn/buildfiles/dd.xml processors=dd snapshot=11  
simulation max_iterations=14
```

Όπου dd.xml είναι το αρχείο με την τοπολογία που επίσης βρίσκεται στον buildfiles φακελο. Στη συνέχεια θα πρέπει μαζί με τα προυπάρχουσα MODULE να εμφανιστεί και το MODULE_LEGACYAPPS_DD το οποίο θέτουμε και ως ON, πατούμε c και q. Υστερα make.

6. Πειραματική Αξιολόγηση του Directed Diffusion στο Shawn

Σε αυτό το κεφάλαιο θα γίνει μια συγκριτική παρουσίαση του Directed Diffusion, η οποία θα επικεντρωθεί στον τομέα της αναταλλαγής μηνυμάτων. Ο τομέας αυτός θα μελετηθεί πάνω σε 6 διαφορετικές τοπολογίες. Αυτές οι τοπολογίες θα περιέχουν 10,100,1000,2000,5000,10000 κόμβους αντίστοιχα. Οι τοπολογίες κλιμακώνουν στον πλήθος των κόμβων και μάλιστα φτάνουν το 10^4 . Αυτό γίνεται για την ανάδειξη των εκτεταμένων δυνατοτήτων του εξομοιωτή Shawn σε αυτόν τον τομέα, σε σχέση με τους εξομοιωτές Ns-2, TOSSIM, OmNET++.



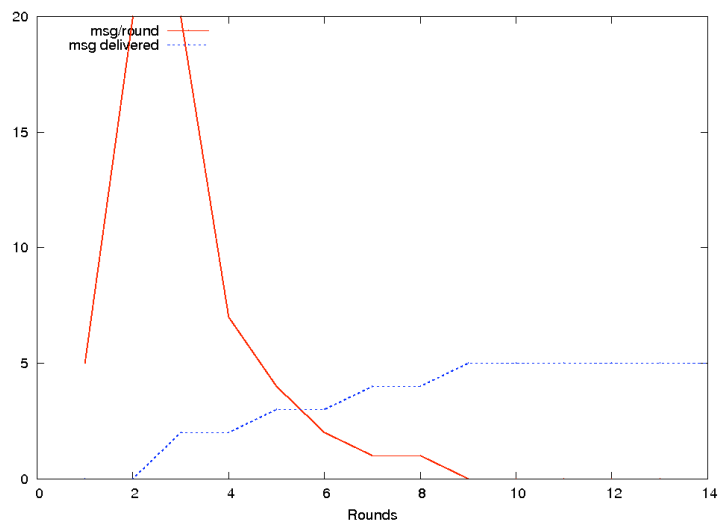
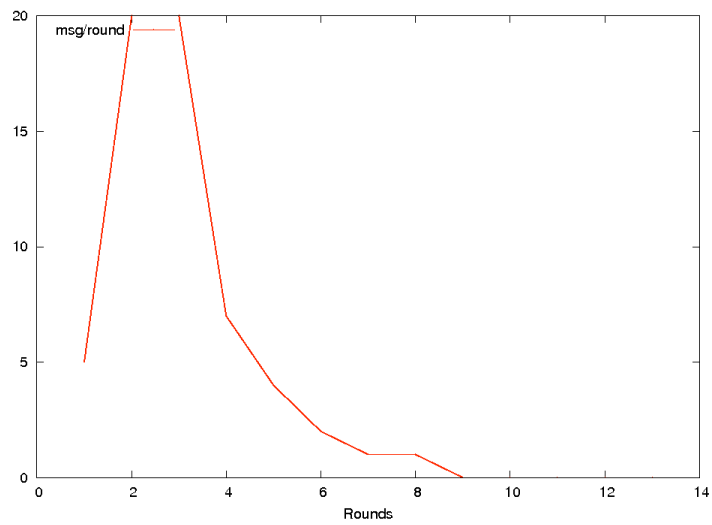
Σχήμα 6: σύγκριση εξομοιωτών

Πρὶν ξεκινήσει το evaluation μετρήθηκαν οι ελάχιστες αποστάσεις ανάμεσα σε κάθε ζεύγος κόμβων. Έτσι κατά τη δοκιμή του Directed Diffusion να είναι δυνατός ο υπολογισμός, του πόσο κοντά στο ελάχιστο μονοπάτι βρίσκεται το μονοπάτι που ανακαλύπτει ο Directed Diffusion. Ακόμα όλοι κόμβοι επικοινωνούν με όλους και μετρήσεις που έγιναν ελέγχουν μόνο τα μονοπάτια που υπάρχουν. Άλλωστε δεν μας ενδιαφέρει η συμπεριφορά του αλγορίθμου σε συνθήκες αναζήτησης ενός interests που δεν υπάρχει.

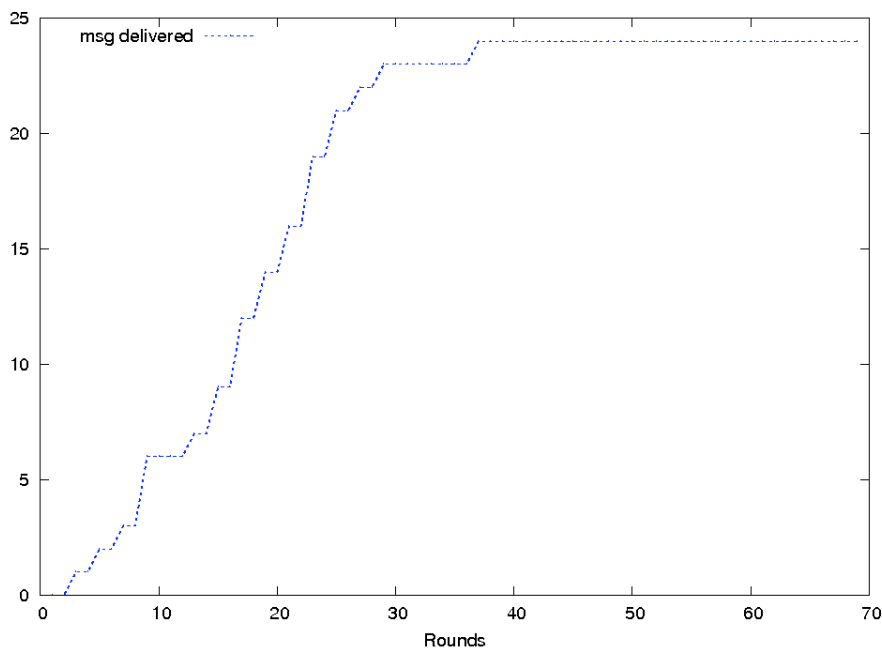
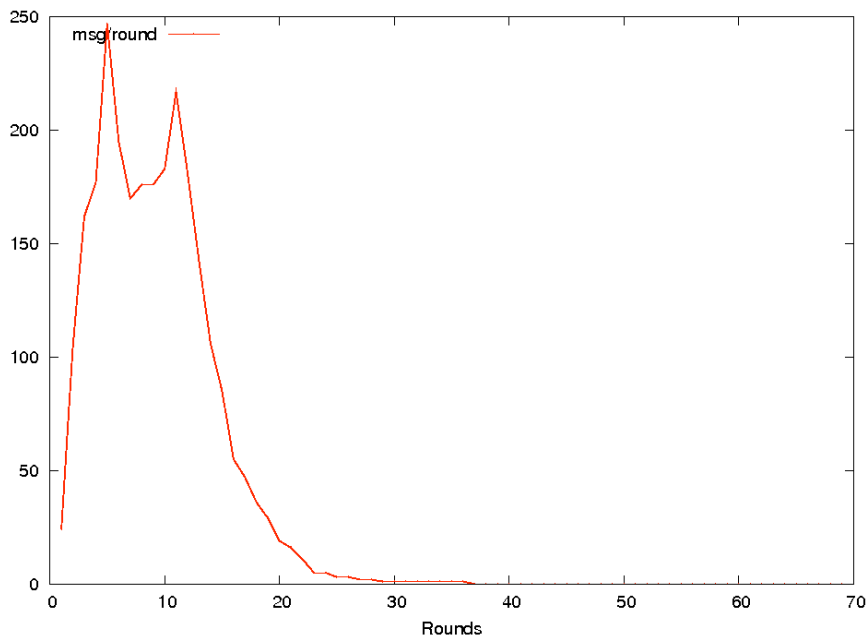
Πρέπει να σημειωθεί παρότι δεν φαίνεται στις κάτωθι γραφικές, ότι το success rate του αλγορίθμου είναι ίσο με ένα (100% επιτυχία). Αυτό αφορά όλες τις τοπολογίες στις οποίες δοκιμάστηκε ο αλγόριθμος. Επίσης πρέπει να αναφερθεί πως εκτός από τη συγκεκριμένο success rate η συγκεκριμένη υλοποίηση βρίσκει τα ελάχιστα μονοπάτια ανάμεσα στα sink και στα interest.

Στη πρώτη γραφική που παρουσιάζεται έχουμε τοπολογία 10 κόμβων και 5 καταβοθρών (sinks). Όπως φαίνεται και στη γραφική περί τον τρίτο γύρο έχουμε τη μέγιστη ανταλλαγή μηνυμάτων. Οι μεγάλες κλίσεις της γραφικής τόσο μέχρι το peak αλλά και μετά δικαιολογούνται, επειδή αρχικά στο πρωτόκολλο γίνεται flooding. Έτσι διαδίδονται ταχύτατα τα μηνύματα και σετάρεται το δίκτυο. Επίσης επειδή προβλέπονται τεχνικές αντιμετώπισης των αναμεταδόσεων, αποφεύγοντα φαινόμενα κατακλισημού του δικτύου από τα ίδια μηνύματα. Πράγμα που οδηγεί στην περεταίρω ελάττωση των μηνυμάτων. Στον γύρο 9 βλέπουμε από τη γραφική ότι πλέον δεν έχουμε ανταλλαγή μηνυμάτων. Η δεύτερη γραφική πέρα από το σύνολο των μηνυμάτων που παρουσιάζει η πρώτη, μας αποτυπώνει και

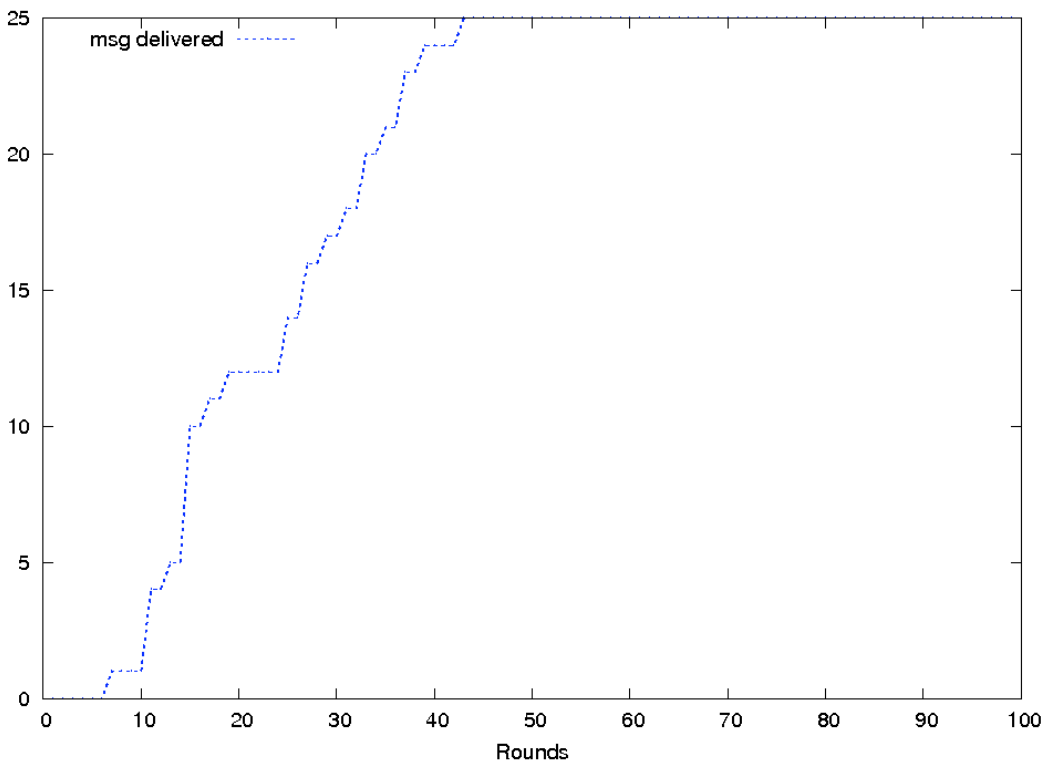
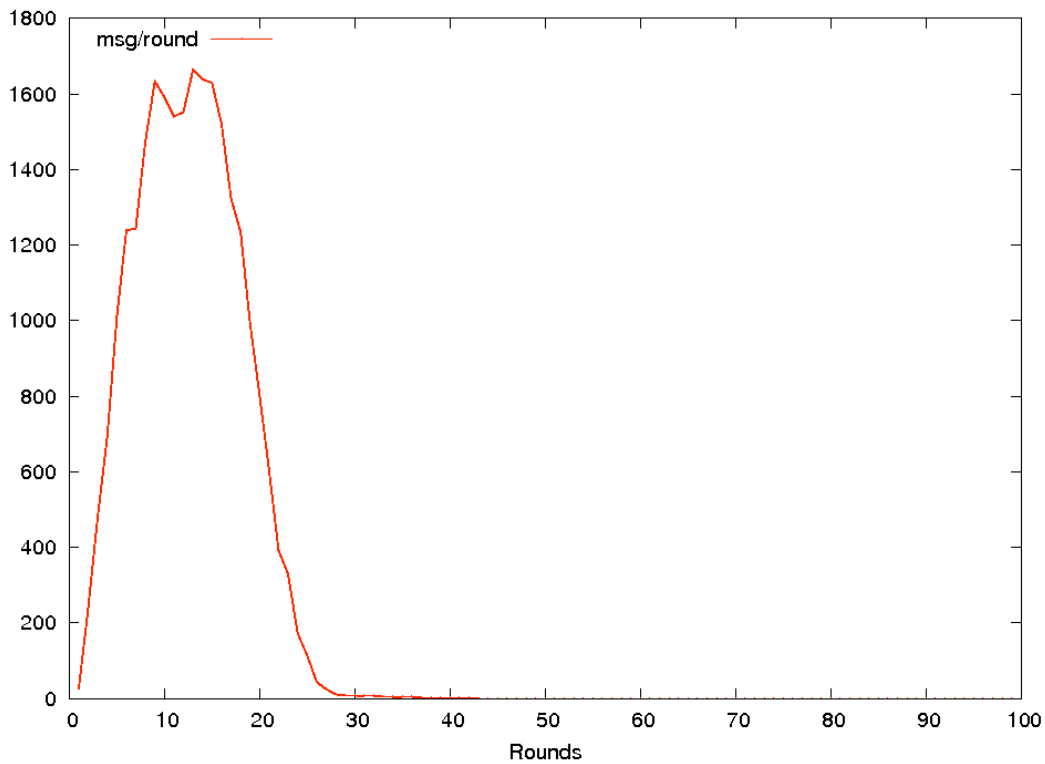
τα interests. Σε κάθε «σκαλοπάτι» της μπλε γραφικής έχουμε τον εντοπισμό του interest. Όταν πλέον στο γύρο 9 η μπλε γραφική έχει σταθεροποιηθεί, σημαίνει ότι έχουν βρεθεί όλα τα interests. Η διάμετρος του δικτύου στα δυο πρώτα σχήματα είναι 4.



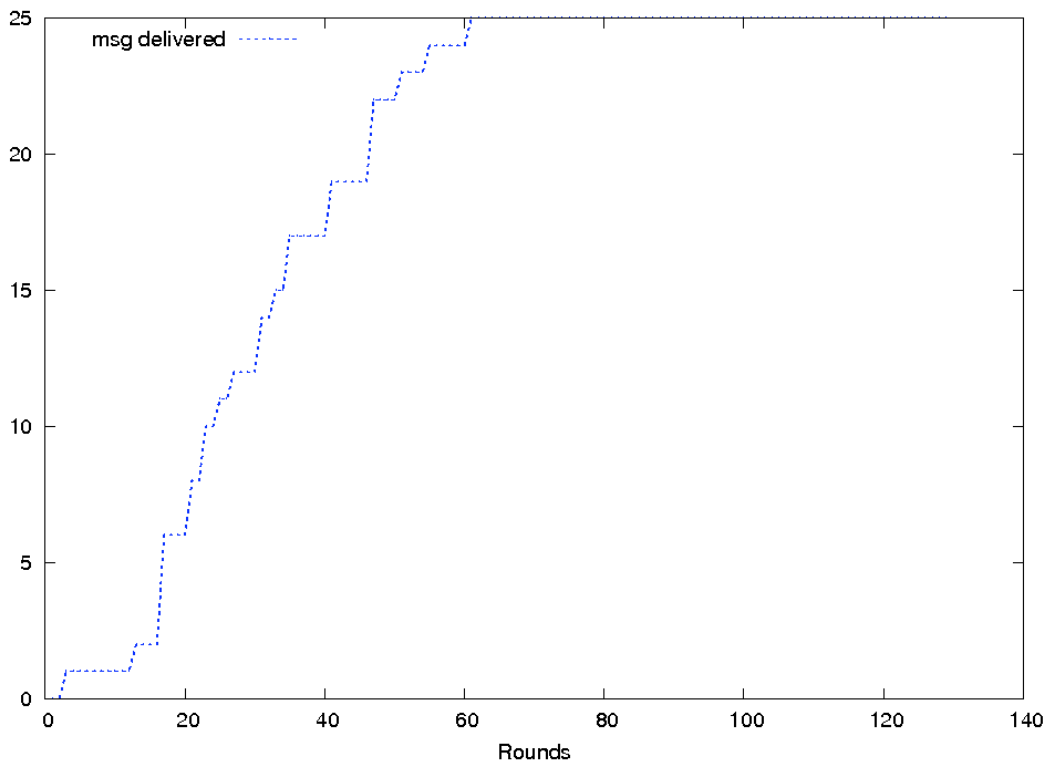
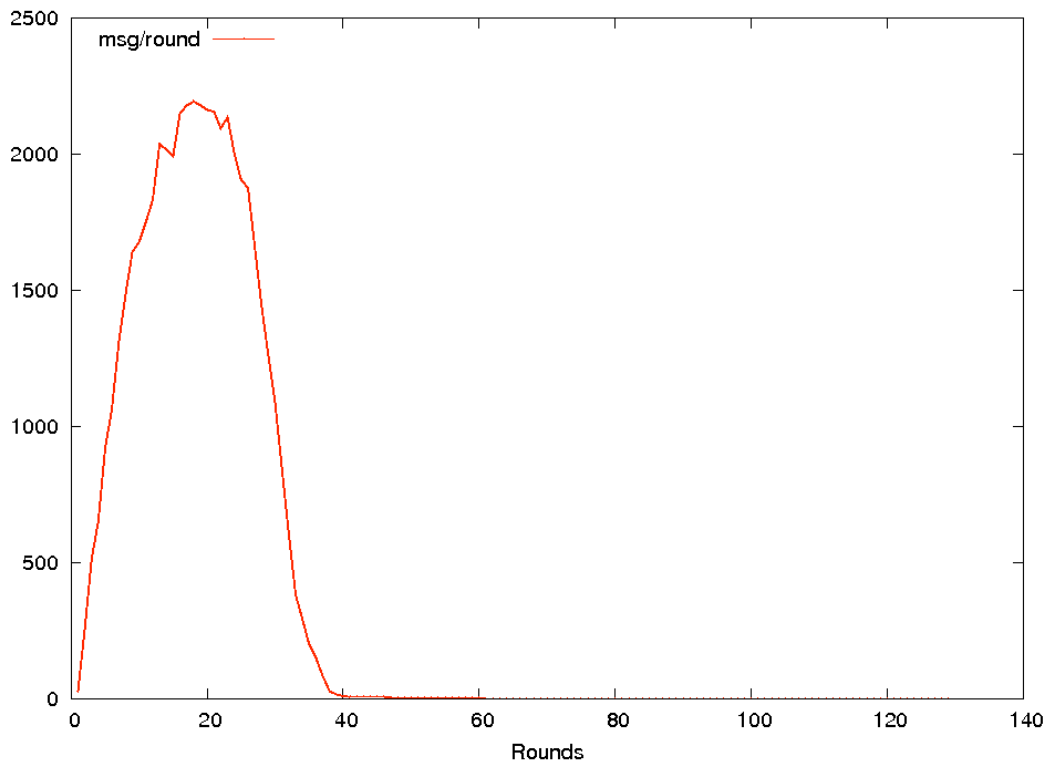
Εδώ έχουμε τις γραφικές για τους 100 κόμβους και τα 24 sinks, στο σύνολο των μηνυμάτων αλλά και των interest που εντοπίζονται. Στον 7^ο γύρο έχουμε τα μέγιστα μηνύματα και στον 37^ο τελειώνουν, βλέπουμε άλλωστε πως σε αυτόν ακριβώς των γύρο σταθεροποιείται και η γραφική των interest. Η διάμετρος του δικτύου είναι 22.



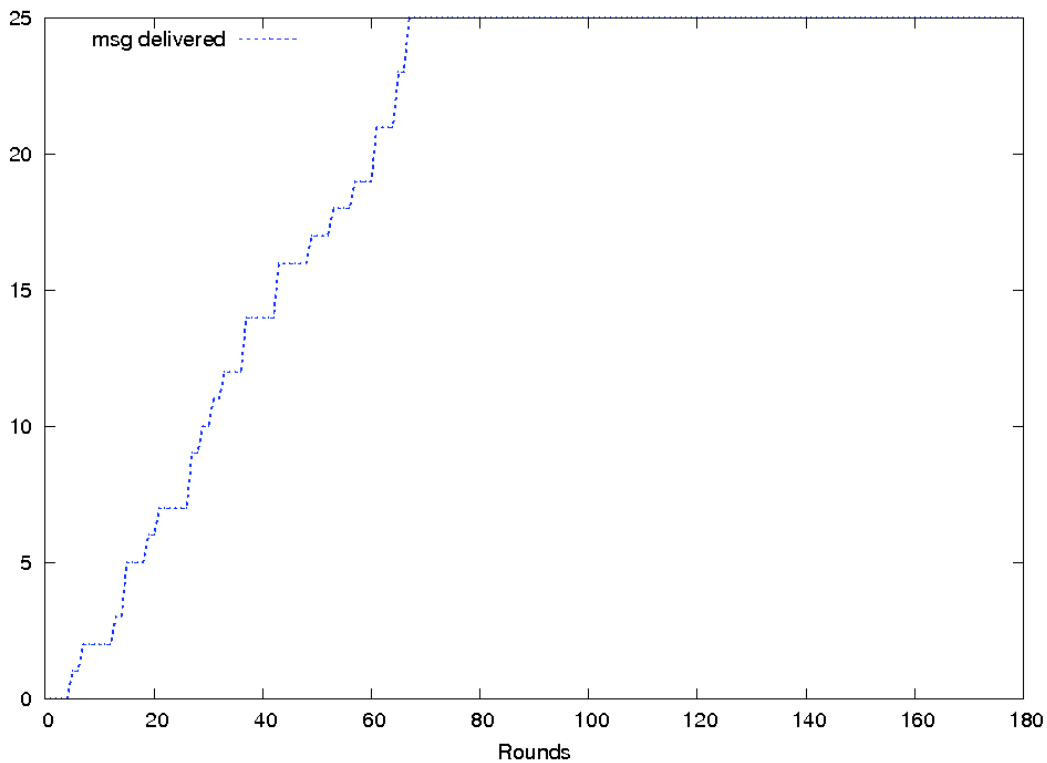
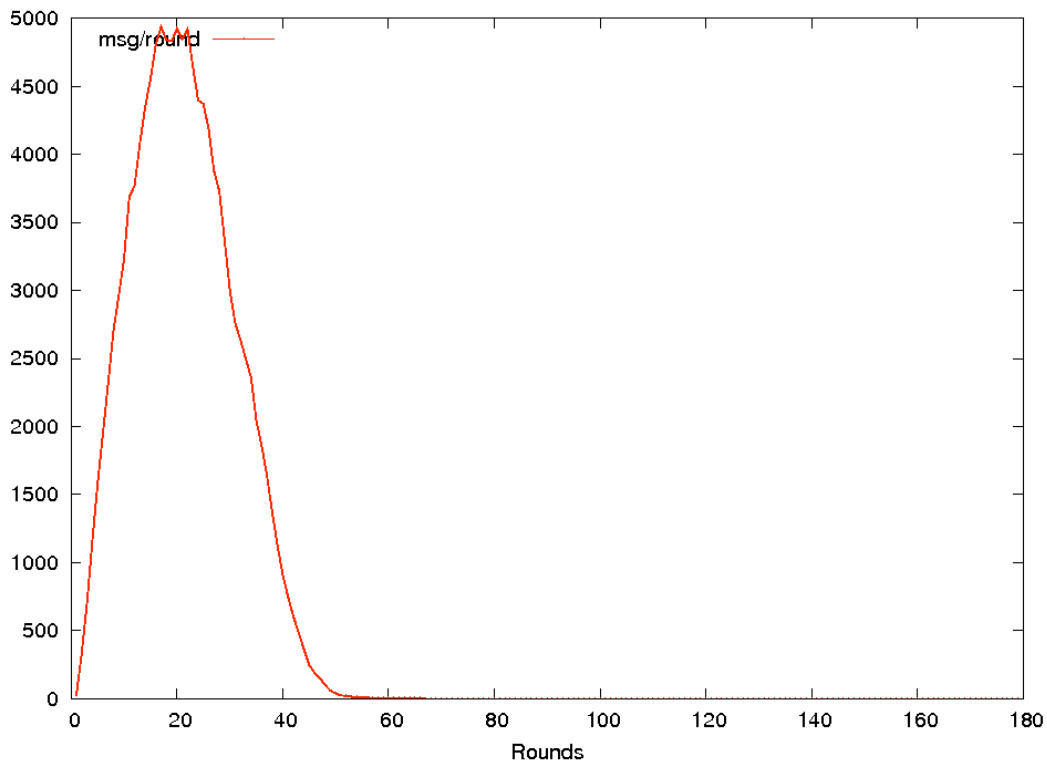
Εδώ έχουμε τις γραφικές για τους 1000 κόμβους και τα 25 sinks, στο σύνολο των μηνυμάτων αλλά και των interest που εντοπίζονται. Από τον 8^ο έως τον 15^ο γύρο έχουμε τα μέγιστα μηνύματα και λίγο μετά τον 40^ο τελειώνουν, βλέπουμε άλλωστε πως σε αυτόν ακριβώς των γύρο σταθεροποιείται και η γραφική των interest. Η διάμετρος του δικτύου είναι 30.



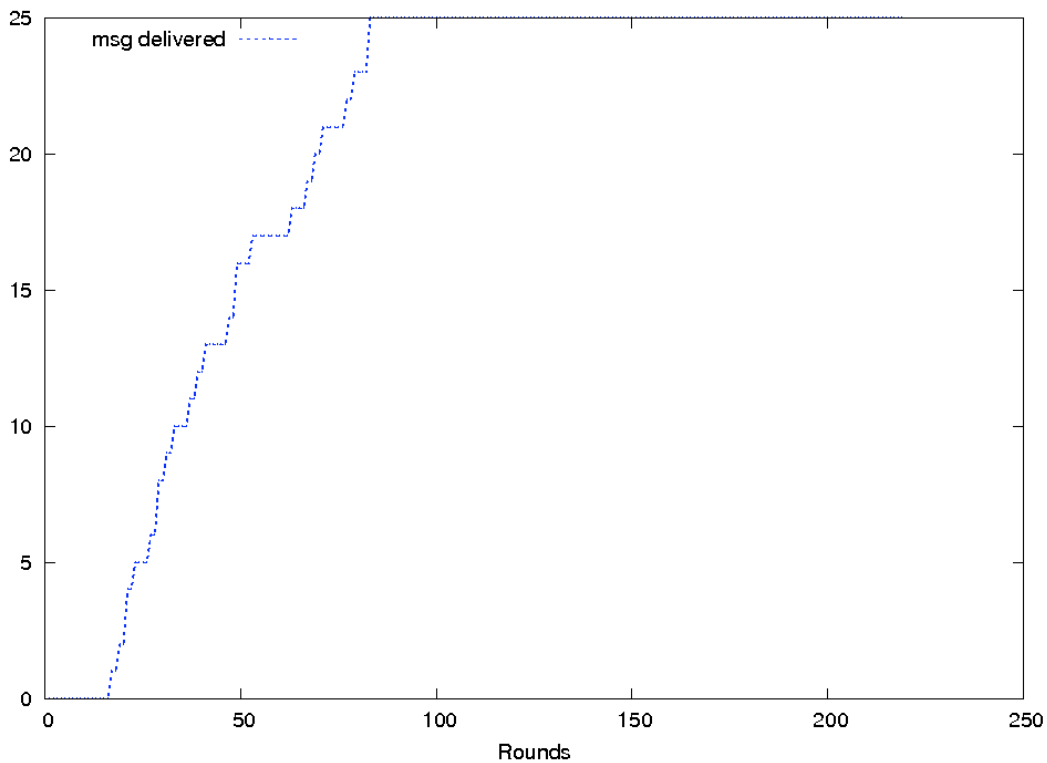
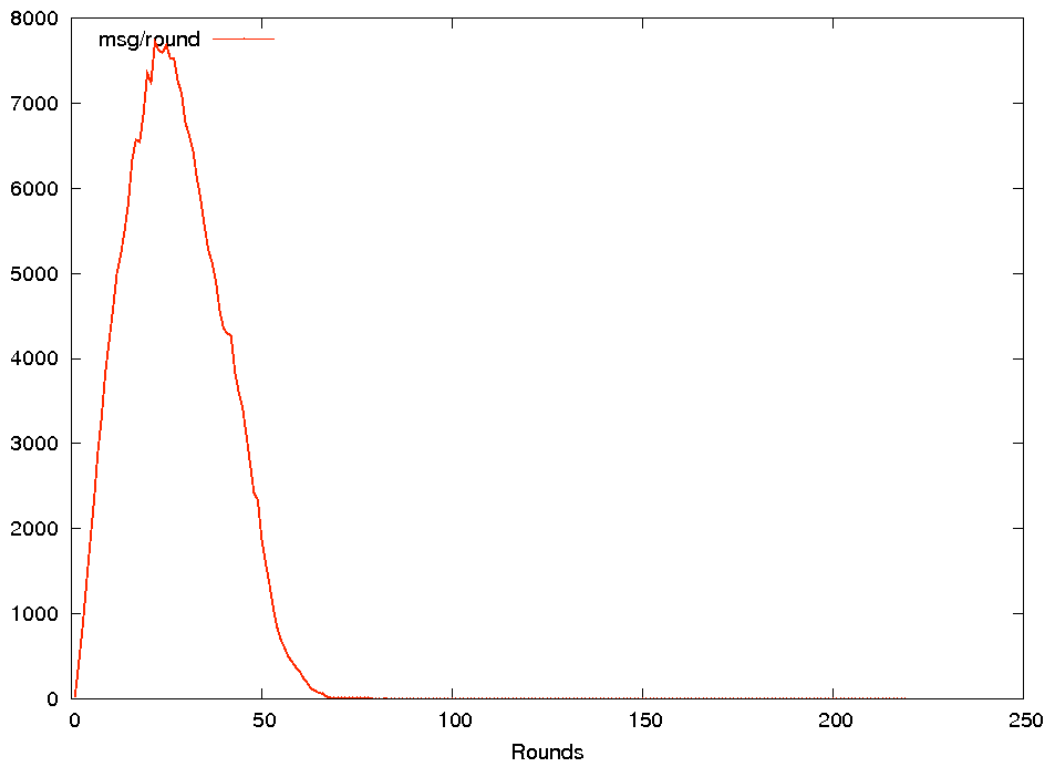
Εδώ έχουμε τις γραφικές για τους 2000 κόμβους και τα 25 sinks, στο σύνολο των μηνυμάτων αλλά και των interest που εντοπίζονται. Στον 18^ο γύρο έχουμε τα μέγιστα μηνύματα και στον 60^ο τελειώνουν, βλέπουμε άλλωστε πως σε αυτόν ακριβώς των γύρο σταθεροποιείται και η γραφική των interest. Η διάμετρος του δικτύου είναι 43.



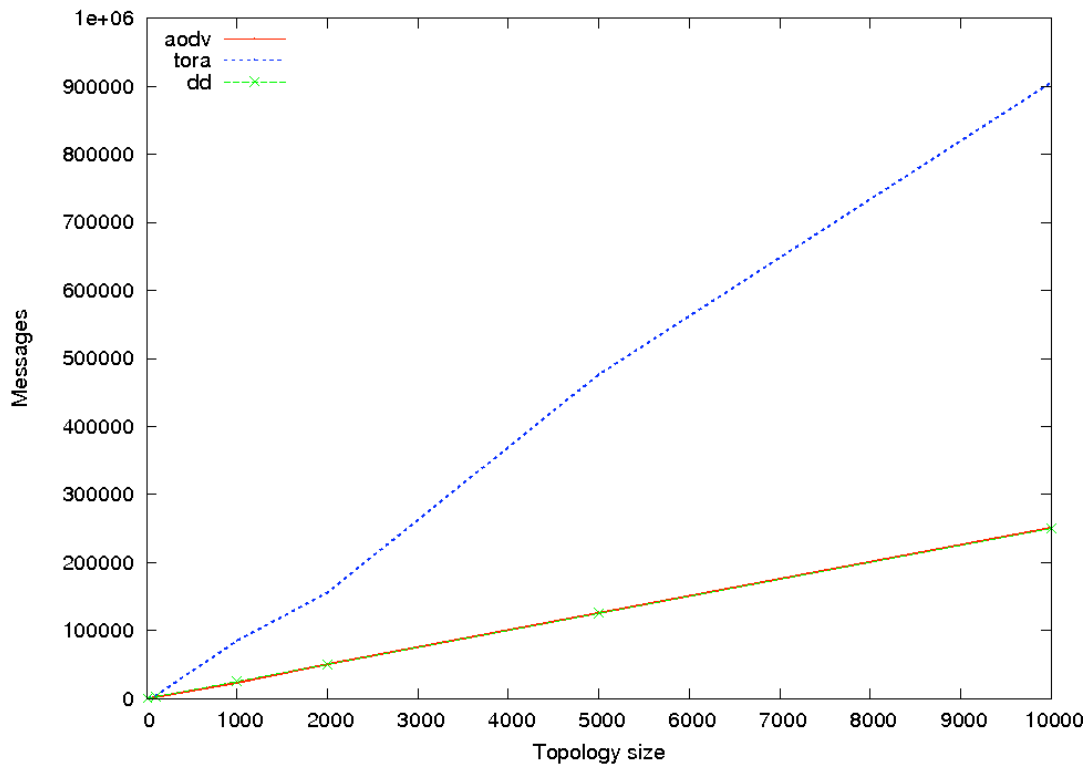
Εδώ έχουμε τις γραφικές για τους 5000 κόμβους και τα 25 sinks, στο σύνολο των μηνυμάτων αλλά και των interest που εντοπίζονται. Από το 20^ο έως 25^ο γύρο έχουμε τα μέγιστα μηνύματα και στον 65^ο τελειώνουν, βλέπουμε άλλωστε πως σε αυτόν ακριβώς των γύρο σταθεροποιείται και η γραφική των interest. Η διάμετρος του δικτύου είναι 58.



Εδώ έχουμε τις γραφικές για τους 10000 κόμβους και τα 25 sinks, στο σύνολο των μηνυμάτων αλλά και των interest που εντοπίζονται. Στον 27^ο γύρο έχουμε τα μέγιστα μηνύματα και στον 75^ο τελειώνουν, βλέπουμε άλλωστε πως σε αυτόν ακριβώς των γύρο σταθεροποιείται και η γραφική των interest. Η διάμετρος του δικτύου είναι 72.



Τέλος ακολουθεί μια συγκριτική γραφική μηνυμάτων/μεγέθους τοπολογιών για τρία πρωτόκολλα. Τα πρωτόκολλα αυτά είναι το TORA, το AODV και το Directed Diffusion. Αν και δεν διακρίνεται εμφανέστατα στη γραφική το DD ανταλλάσει κατ ελάχιστον λιγότερα μηνύματα και από το aodv.



Μέγεθος Τοπολογίας	Directed Diffusion	AODV	TORA
10	60	63	75
100	2586	2355	4277
1000	25280	23209	85062
2000	50344	50729	155935
5000	125455	125897	476416
10000	250433	251107	906224

Πίνακας 6: Συγκριτικό πρωτοκόλλων DD, TORA, AODV

7. Τελικά συμπεράσματα

Το Directed Diffusion είναι ένα πρωτόκολλο, το οποίο έχει σημαντικές επιδόσεις στο τομέα της εξοικονόμησης ενέργειας. Αυτό προκύπτει τόσο από τη θεωρητική ανάλυση, αλλά και από τις γραφικές της ενότητας 3.4. Άλλωστε στην ενότητα της πειραματικής αξιολόγησης (στον Shawn) το Directed Diffusion ανταλλάσει λιγότερα μηνύματα από τον aodv και κατά πολύ λιγότερα από τον tova. Τα λιγότερα μηνύματα οδηγούν και μικρότερες σπατάλες στην ενέργεια. Ακόμα η τοπικότητα που εμφανίζει το Directed Diffusion θεωρείται και είναι θετικό χαρακτηριστικό σε διαδραστικά δίκτυα. Επιπλέον το Directed Diffusion επιδέχεται αρκετές βελτιστοποιήσεις, οι οποίες καθιστούν τον αλγόριθμο εξαιρετικά σημαντικό και βρίσκει πολλαπλές εφαρμογές. Παρόλα αυτά το MAC επίπεδο επηρεάζει καταλυτικά τις δυνατότητες του Directed Diffusion.

Γενικά αν και περιγράφονται όλες οι πτυχές του πρωτοκόλλου, σε αρκετά σημεία δεν έχουμε εξειδίκευση. Αυτό στον τομέα της υλοποίησης μας δίνει αρκετές δυνατότητες. Όμως να προσεχτεί ότι οι δυνατότητες αυτές περιορίζονται στη συνέχεια από τον κώδικα που τελικά τρέχει στις συσκευές (π.χ. isense). Με απλά λόγια δεν υποστηρίζονται όλες οι τεχνικές στον κώδικα. Ο ίδιος ο Shawn καλό είναι να προσεγγιστεί αρχικά σε επίπεδο απλών παραδειγμάτων. Έτσι αρχικά θα πρέπει ο χρήστης του εξομοιωτή να μάθει το πως δημιουργεί το σύνολο των αναγκαίων αρχείων, τις τοπολογίες αλλά και το πως αλληλεπιδρά με το interface του εξομοιωτή. Στη συνέχεια θα μπορεί ο ενδιαφερόμενος να αναπτύξει την εφαρμογή του. Για αυτό χρησιμοποιείται κατά κύριο λόγο το αρχείο που proccessor. Η συγγραφή πρωτοκόλλων σε εξομοιωτές απαιτεί προσεκτική απασφαλμάτωση και γνώση τεχνικών ανάπτυξης κώδικα.

Βιβλιογραφία

- [1] *Holger Karl & Andreas Willing*, Protocols and Architectures for Wireless Sensor Networks
- [2] *Deliverable D4.1: Software Implementation and Integration Guidelines*
- [3] *Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann and Fabio Silva*, Directed Diffusion for Wireless Sensor Networking
- [4] *Shawn web site*, <http://shawn.sourceforge.net/>
- [5] *Κατανεμημένα Συστήματα web site*,
<http://www.ceid.upatras.gr/courses/katanemhmena/wiki/index.php/%CE%9A%CE%B1%CF%84%CE%B1%CE%BD%CE%B5%CE%BC%CE%B7%CE%BC%CE%AD%CE%BD%CE%B1%CE%A3%CF%85%CF%83%CF%84%CE%AE%CE%BC%CE%B1%CF%84%CE%B1>
- [6] *Deepak Ganesan*, TinyDiffusion Application Programmer's Interface (API) 0.1
- [7] *Fabio Silva, John Heidemann and Ramesh Govindan*, Network Routing Application Programmer's Interface (API) and Walk Through 9.0.1
- [8] *Bjarne Stroustrup*, The C++ Programming Language
- [10] <http://el.wikipedia.org>