Εφαρμογές του Επεξεργαστή Δίκτυων IXP 1200 σε Συστήματα Ανιχνεύσεως Είσβολων για Δίκτυα

Ιωάννης Χαριτάκης
— Μεταπτυχιακή Εργασία —

Ακαδημαϊκός Σύμβουλος: Ευάγγελος Μαρκάτος

Τμήμα Επιστήμης Τεχνολογιών
Πανεπιστήμιο Κρήτης

Σεπτέμβριος 2003
Περάθεψη

Οι Επεξεργαστές Δικτύων και τα Συστήματα Ανίχνευσης Εισβολέων για τα δίκτυα αποτελούν δύο ραγδαία αναπτυσσόμενες τομείς, τόσο στην έρευνα όσο και στην βιομηχανία. Σε αυτή την εργασία διερευνούμε την χρήση του επεξεργαστή δικτύων IXP 1200 σε δύο πτυχές της Ανίχνευσης Εισβολέων: την Ανάλυση Επικεφαλής Παρέτον και την Κατανομή Ανίχνευση Εισβολέων. Σκοπός αυτής της ανάλυσης είναι να εντοπίσουμε σε αρχικά τον ισχυρότερο παράδειγμα σε αποτελέσματα επικεφαλίδους τα αποτελέσματα είναι μία καλή ένδειξη απόπειρας εισβολής. Παράλληλα με την αρχιτεκτονική λογισμικού υλοποίησαμε το εργαλείο S2I το οποίο παράγει ως είσοδο περιγραφές ασυνήθεια παρέτον γραμμένες σε μία γλώσσα υψηλού επιπέδου και παράγει χαμηλό επιπέδου αποδοτικό χώδικα για τον προγραμματισμό του IXP 1200. Συνολικά, η αρχιτεκτονική λογισμικού και το εργαλείο S2I αποτελούν ένα πρωτό τρόπο για την ενσωμάτωση του επεξεργαστή IXP 1200 σε Συστήματα Ανίχνευσης Εισβολέων. Όσο αφορά την Κατανομή Ανίχνευσης Εισβολέων, παρουσιάζουμε την αρχιτεκτονική ενός Ενεργού Διαμορφώστη για τον έξυπνο διαμορφωτή της δικτύωσης χίνησης σε πολλαπλά Συστήματα Ανίχνευσης Εισβολέων που δουλεύουν παράλληλα. Ο Ενεργός Διαμορφώστης χρησιμοποιεί τις τεχνικές του φύλλαρισμάτος σε χαμηλό επίπεδο και των ενταμυθών τοποθέτησης προσεκίνησης να βελτιώσει την απόδοση αδιάκριτου, από το Συστήματα Ανίχνευσης Εισβολέων. Αρχικά, η ιδέα του Ενεργού Διαμορφώστη μελετήθηκε προσεκτικά με την χρήση προσανατολίσματος. Οι μετρήσεις που πήραμε έδειξαν όταν συνολική απόδοση των παράλληλων Συστήματα Ανίχνευσης Εισβολέων μπορεί να βελτιώσει μέχρι και 20%. Εφόσον τα πλεονεκτήματα του Ενεργού Διαμορφώστη έγιναν φαινόμενα, προχωρήσαμε στην υλοποίησή του με τον IXP 1200. Η υλοποίησή έγινε πάνω σε πραγματικό υλικό και τα πλεονεκτήματα του Ενεργού Διαμορφώστη επιβεβαιώθηκαν με πραγματικές μετρήσεις.
Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον ακαδημαϊκό σύμβουλό και επόπτη κύριο Ευάγγελο Μαρκάτο. Η όρεξή του για δουλειά και η αγαπή του για τα συστήματα υπολογιστών είναι πέρα για πέρα μεταδοτική. Καθοριστική ήταν η επίσης και η συνεισφορά του Κώστα Αναγνωστάκη που έστω και από πολύ μικρά βοήθησε εξαφαντικά στις δύο πράτες μου δημιουργείσες και όχι μόνο. Θα ήθελα επίσης να ευχαριστήσω τον κύριο Διονύση Πνευματικό για την πολύτιμη συνεισφορά του γενικότερα και ειδικότερα για την βοήθεια που μου έδωσε στο τρίτο κεφάλαιο αυτής της εργασίας. Επίσης θα ήθελα να ευχαριστήσω τον κύριο Μανώλη Κατεβαίνη, που αν και τελικά δεν συνεργαστήκαμε περισσότερο, οι ακαδημαϊκές του συμβουλές είχαν μεγάλη σημασία στη συνέχεια του μεταπτυχιακού μου. Θα ήθελα τέλος να ευχαριστήσω τον κύριο Αγγελο Μπίλα και τον κύριο Βασίλειο Σύρη που δέχτηκαν να παραστούν στην τελική και πιο χρήσιμη ράση αυτής της εργασίας.

Θα ήθελα επίσης, μέσα από αυτές τις γραμμές, να ευχαριστήσω τον πολύ καλό φίλο Γιώργο Σαπουντζή, τόσο για την φιλία του όσο και για τις πολύτιμες ακαδημαϊκές του συμβουλές. Τον Νίκο Γαλανόπουλο για τις απελευθέρωσε ώρες ευχάριστες συζητήσεις και σκακιού. Τον Βασίλη Κοντογιάννη, ομοσποδή μεταπτυχιακού, με τον οποίο ξερέψαμε που και ποιο για τρέξιμο. Τον Μανώλη Τρικονάνθη για την εκπληκτική παρατηρητικότητα του που έβαζε ευχάριστες παρενθέσεις σε κάθε είδους συζήτηση! Τον Γιώργο Παπαζωγόλου με τον οποίο χάσαμε πολλές ώρες τον αριθμό του 'Σημαντήριδες' στο μοναδικό μέρος που σερβίροντα, που γκουέρνισα... (Λέγει πριν πάει φωτιά η δουλειά στο μεταπτυχιακό)

Επιστρέφοντας στο περβάλλον του ΠΤΕ, θα ήθελα να ευχαριστήσω τα παιδιά του εργαστηρίου, τον Δημήτρη (συγκάτοικο εργασίας), τον Σπύρο και τον Μιχάλη με τους οποίους είχαμε επανειλημμένα άτυπες συναντήσεις για διάφορα θέματα επιστημονικού ενδιαφέροντος (και όχι μόνο :-)). Θα ήθελα να πω ένα ευχαριστώ στους system administrators και ιδιαίτερα στον Βαγγέλη Καραγιάνη, ο οποίος από τότε που το θυμάμαι, βρίσκεται συνέχεια στο πόδι.

Επίσης, θεωρώ πάρα πολύ σημαντική την βοήθεια που είχα από πολλούς

---

1 Ποσό εμπνευσμένο από τον πολύ καλό φίλο Γιώργο Σημαντήρη
ανθρώπους που πιθανότατα ποτέ δεν θα γνωρίσω και οι οποίοι χωρίς να με ξέρουν καθόλου αφιέρωσαν μέρος από τον χρόνο τους για να με βοηθήσουν στην εργασία μου με τον IXP. Ιδιαίτερα θα ήθελα να πάω ένα ευχαριστώ στους Charles M. "Chip" Coldwell, Erik Johnson και Kenneth M. Mackenzie.

Σε όλη την διάρκεια του μεταπτυχιακού αλλά και πολύ πριν από αυτό, είχα την υποστήριξη της Βαγγελιώς μου. Βοήθησε με χίλιους δύο τρόπους που δεν χωρούν σε λόγια. Αυτή η παράγραφος είναι ένα πολύ μορφό ευχαριστώ για όσα μου έχει δώσει.

Τέλος, ορείλα πολλά στους καταπληκτικούς γονείς μου, τον Χάρη και την Ειρήνη, που μου πρόσφεραν τα πάντα και στους οποίους αφιερώνω αυτή την εργασία.
Περιεχόμενα

1 Εισαγωγή

2 Βασικές Πληροφορίες 3
   2.1 Εισαγωγή στους Επεξεργαστές Δοτών ...................... 3
   2.2 Ο Επεξεργαστής Δοτών IXP 1200 .......................... 4
       2.2.1 Αρχιτεκτονική Υλικού του IXP 1200 .................. 4
       2.2.2 Οι Μηχανές ............................................. 7
       2.2.3 Περι Αποστολής και Λήψης Πακέτων ..................... 9
       2.2.4 Πλατφόρμες Υλικού για Ανάπτυξη Κώδικα ............. 12
   2.3 Συστήματα Ανίχνευσης Εισβολέων ......................... 13
   2.4 Snort ......................................................... 14
       2.4.1 Η Αρχιτεκτονική του Snort ....................... 16

3 Ανάλυση Επικεφαλίδων με τον IXP 1200 19
   3.1 Εισαγωγή ................................................. 19
   3.2 Σχετικές Εργασίες ....................................... 20
   3.3 Το Εργαλείο S2I ............................................ 21
       3.3.1 Στατικό Τμήμα ......................................... 21
       3.3.2 Δυναμικό Μέρος ..................................... 23
   3.4 Αξιολόγηση .................................................. 26
       3.4.1 Αξιολόγηση Στατικού Μέρους ......................... 26
       3.4.2 Αξιολόγηση Δυναμικού Μέρους ....................... 26
       3.4.3 Αξιολόγηση Χώρου .................................. 26
       3.4.4 Αξιολόγηση Χρόνου Εκτέλεσης ..................... 28
3.4.5 Ένα ελαφρύ Snort για συστήματα i386 ........................ 29
3.5 Ανακεφαλαίωση και Μελλοντική Εργασία .......................... 30

4 Ο Ενεργός Διαμοιραστής για ΣΑΕ ................................. 31
4.1 Εισαγωγή ................................. 31
4.2 Σχετικές Εργασίες .......................... 32
4.3 Σχεδίαση ................................. 34
4.3.1 Φίλτρα σε Χαμηλό Επίπεδο .......................... 34
4.3.2 Διαμοιρασμός Φόρτου .......................... 35
4.3.3 Εντομοειδές Τοπικότητας .......................... 35
4.4 Περιέβαστα ................................. 37
4.4.1 Αξιολόγηση Φίλτρα σε Χαμηλό Επίπεδο .......................... 38
4.4.2 Απόδοση της Συνάρτησης Hash .......................... 39
4.4.3 Επίδραση των ET στην Απόδοση του ΣΑΕ .......................... 40
4.4.4 Αξιολόγηση Φίλτρα σε Χαμηλό Επίπεδο και ET .......................... 45
4.5 Περίληψη ................................. 46

5 Υλοποίηση του Διαμοιραστή στον IXP 1200 .......................... 47
5.1 Υλοποίηση ................................. 48
5.1.1 Περιγραφή του Συστήματος .......................... 48
5.1.2 Περιγραφή του Διαμοιραστή .......................... 48
5.1.3 Διαδικασία Λήψης .......................... 52
5.1.4 Διαδικασία Αποστολής .......................... 56
5.1.5 Συγχρονισμός Ινών Λήψης και Ελεγκτή Αποστολής .......................... 58
5.1.6 Συγχρονισμός των Ινών Λήψης .......................... 59
5.2 Περιέβαστα ................................. 60
5.2.1 Απόδοση Συστήματος Αποστολής .......................... 61
5.2.2 Απόδοση Συστήματος Λήψης .......................... 61
5.2.3 Απόδοση του Διαμοιραστή .......................... 61
5.2.4 Αξιολόγηση των ET .......................... 62
5.3 Περίληψη ................................. 64

6 Συμπεράσματα .......................... 65
Κατάλογος Σχημάτων

2.1 Απόδοση (Performance) και Ευελιξία (Flexibility) για διαφορετικές σχεδιαστικές λύσεις ................. 4
2.2 Αρχιτεκτονική Τλικού του IXP 1200 ................. 5
2.3 Η Αρχιτεκτονική Εσωτερικού Διαύλου του IXP 1200 .... 7
2.4 Διαδικασία λήψης με την λειτουργία Αρχής Πόρτας ............. 10
2.5 Η λειτουργία Γρήγορης Πόρτας ..................... 11
2.6 Η πλακέτα ENP-2506 από την Radisys ................. 13
2.7 Τοποθέτηση ενός Συστήματος Ανάγνωσης Εισβολών .... 14
2.8 Δομή μίας υπογραφής του Snort ................. 15
2.9 Παράδειγμα αρχείου υπογραφών για το Snort ............. 15
2.10 Περιγραφή της μηχανής του Snort ................. 17

3.1 Μέθοδος ίδιας: τα παράτατα διαμοιράζονται στις ίνες της μηχανής. Όλες οι ίνες εκτελούν τον ίδιο κώδικα ................. 22
3.2 Μέθοδος μηχανής: παράτατα κατανεμούνται στις μηχανές. Η μια ίνα κανείς τους ελέγχει, ενώ μια άλλη διαχειρίζεται την ουρά εργασίας ................. 23
3.3 Παράδειγμα Δένδρου ..................... 24
3.4 Κώδικας με βάση την Εκόνα 3.3 .................. 25
3.5 Ορείχαλκος χώρος σε κέλυφος μηχανής (232 MHz). Για την μετρητή χρησιμοποιήσαμε έναν σύνδεσμο των 100 Mbit/s και παράτατα ελάχιστο μεγέθους ................. 27
3.6 Γραμμωτός χώρος που παράγεται από το εργαλείο S2I χωρίς την χρήση της δενδροποιίας δομής .................. 28

4.1 Η αρχιτεκτονική του ενεργού διαμορφωτή ................. 34
4.2 Ομαδοποίηση παρατατών με τους ενταμειτές τοποκόψεις .... 36
4.3 Η επίδραση του Φύλτραρίσματος σε Χαμηλό Επίπεδο στην απόδοση του ανιχνευτή.

4.4 Συνολικός χρόνος χρήσης για όλους τους ανιχνευτές σε σχέση με το πλήθος των ανιχνευτών.

4.5 Χρόνος χρήσης του πιο βαρύνταξιμένου ανιχνευτή σε σχέση με το πλήθος των ανιχνευτών για το ίδιο πείραμα της εικόνας 4.4.

4.6 Μέσο μέγεθος ριπής πακέτων σε σχέση με το πλήθος των ανιχνευτών για το πείραμα των Εικόνων 4.4 και 4.5.

4.7 Ιστόγραμμα του μεγέθους ριπής με χρήση ET και χωρίς ET για την περίπτωση τετσάρων ανιχνευτών της Εικόνας 4.6.

4.8 Βελτίωση της απόδοσης για διαφορετικό πλήθος ET.

4.9 Βελτίωση της απόδοσης για διαφορετικό μέγεθος ET.

4.10 Βελτίωση για διαφορετικές πολιτικές χρήσης των ET.

5.1 Περίληψη του συστήματος.

5.2 Περίληψη της εργασίας του διαμοιραστή.

5.3 Κατανομή εργασίας στα στοιχεία του ΙΧΡ.

5.4 Τα βήματα των ισόων λήψης πακέτου.

5.5 Σειρά εκτέλεσης των Ισόων Λήψης.

5.6 Δομή του κουτιών μηνυμάτων (message box).

5.7 Συγχρονισμός των Ισόων Λήψης.

5.8 Δεδομένα Πακέτου.

5.9 Δεδομένα Ενταμητών Τοπικότητας.

5.10 Δεδομένα υπάρχοντων ενός Ενταμητή Τοπικότητας.

5.11 Προσθήκη ενός ET στην υφασμα αποστολής.

5.12 Προσθήκη στην υφασμα αποστολής χωρίς ET.
Κατάλογος Πινάκων

3.1 Εξοικονόμηση χώρου με την δενδρική δομή . . . . . . . . . . . . . . 27
3.2 Κύκλοι (232 MHz) για τον έλεγχο των πεδίων . . . . . . . . . . . . 29
4.1 Μέθοδοι για την χρήση των Ενταμευτών Τοποκόσμησης . . . . . . . 37
4.2 Απόδοση της απλοποιημένης CRC16 hashing συνάρτησης . . . 40
5.1 Μετρήσεις του IXP-Διαμορφωτή . . . . . . . . . . . . . . . . . . . . . 63
Στον Χάρη και την Ειρήνη
Κεφάλαιο 1

Εισαγωγή

Αυτή η εργασία θέτει χοντά δύο από τους πλέον αναπτυσσόμενους τομείς των δοκιμών, τους Επεξεργαστές Δοκιμών (Network Processors) και τα Συστήματα Ανίχνευσης Εισβολέων (Network Intrusion Detection Systems). Η χρήση Επεξεργαστών Δοκιμών σε Συστήματα Ανίχνευσης Εισβολέων είναι κάτι το κοινό, τουλάχιστον μετά την στιγμή που αυτή η εργασία γράφτηκε.

Από τη μια μεριά, οι Επεξεργαστές Δοκιμών επιταχύνουν εφαρμογές που βασίζονται στην επεξεργασία πακέτων. Αυτό γίνεται με το να παρέχουν επιπλέον μονάδες υλικού που μπορούν να χρησιμοποιηθούν παράλληλα. Η εξέλιξη τους είναι ραγδαία τα τελευταία χρόνια. Μάλιστα, πολλές γνωστές εταιρείες από τον χώρο της βιομηχανίας υπολογιστών έχουν επεκτείνει τα προϊόντά τους ώστε να χαλύπτουν αυτή τη νέα γενιά επεξεργαστών (π.χ. Intel, IBM, Cisco και άλλες). Επίσης, η ακαδημαϊκή κοινότητα έχει δείξει διαφορές στόχους στην χρήση τους σε εφαρμογές δρομολόγησης πακέτων όσο και για την αρχιτεκτονική τους αυτή χαλάματα. Είναι λοιπόν σαφές πως υπάρχει ένα έντονο ενδιαφέρον για την ανάπτυξη και την χρήση των Επεξεργαστών Δοκιμών. Συγκεκριμένα, σε αυτή την εργασία διαλέξαμε τον Επεξεργαστή Δοκιμών IXP 1200 από την Intel (για λόγους που αναφέρονται αναλυτικά στο σχετικό κεφάλαιο).

Από την άλλη μεριά, τα Συστήματα Ανίχνευσης Εισβολέων προσφέρουν περισσότερη ασφάλεια σε οργανισμούς με αυξημένη δικτυωση παρουσία. Δεδομένου ότι, στο διαδραστικό παρουσιάζονται όλο και περισσότεροι χίλιοι (τοις εκατομμύριοι ταχυδρομείο, άκολουθα του διαδικτύου, και) ο ρόλος τους είναι ακόμα πιο σημαντικός. Όπως και με τους Επεξεργαστές Δοκιμών, υπάρχει διαφορές έρευνα και ανάπτυξη και δεδομένων Συστήματα Ανίχνευσης Εισβολέων, τόσο σε ακαδημαϊκό όσο και σε βιομηχανικό επίπεδο. Για την εργασία αυτή επιλέξαμε το Σύστημα Ανίχνευσης Εισβολέων Snort.

Η εργασία αυτή μπορεί να διαμορφεί σε δύο μέρη. Το πρώτο μέρος χαλύπτεται από το Κεφάλαιο 3 ενώ το άλλο χαλύπτεται από τα δύο τελευταία κεφάλαια.
Στο Κεφάλαιο 2 παρουσιάζουμε τα βασικά από τις περιοχές των Επεξεργαστών Δακτύλων και των Συστήματων Ανίχνευσης Εισβολών για δίκτυα. Στη συνέχεια, στο Κεφάλαιο 3, παρουσιάζουμε μία πρωτοποριακή αρχιτεκτονική λογισμικού που καθιστά δυνατή τη χρήση του IXP ως μια μονάδα ανάλυσης επικεφαλίδων πακέτων. Επιπλέον παρουσιάζουμε το εργαλείο S2I που αναπτύχθηκε προκειμένου να προγραμματιστεί ο IXP εύκολα και αποδοτικά ώστε να ανιχνεύει πακέτα που παθανώς είναι επικίνδυνα.

Στο κεφάλαιο 5 δείχνουμε την υλοποίηση του Ενεργού Διαμορφωτή πάνω στον IXP. Περιγράφουμε με αρκετή λεπτομέρεια την δομή του καθώς και τα περιβάλλοντα που κάνονται. Οι μετρήσεις που πήραμε συμπεράνθησαν με τα αποτελέσματα των προσομοιώσεων του προηγούμενου κεφαλαίου.

Τέλος, κλείνουμε την παρούσα εργασία με το Κεφάλαιο 6 στο οποίο συνοψίζουμε τα συμπεράσματα όλης της εργασίας.
Κεφάλαιο 2

Βασικές Πληροφορίες

Σε αυτό το κεφάλαιο εισάγουμε τις βασικές έννοιες και πληροφορίες γύρω από τις οποίες κυνείται η παρούσα εργασία. Το Τμήμα 2.1 αυτού του κεφαλαίου αποτελεί μία γενική εισαγωγή στους Επεξεργαστές Δικτύων. Ο συγκεκριμένος επεξεργαστής δικτύων ΙΧΡ 1200 που χρησιμοποιήσαμε, παρουσιάζεται αναλυτικά στο Τμήμα 2.2. Στο Τμήμα 2.3 εισάγουμε βασικές πληροφορίες σχετικές με τα Συστήματα Ανίχνευσης Εισβολέων για δίκτυα. Στο τελευταίο τμήμα αυτού του κεφαλαίου παρουσιάζουμε πιο αναλυτικά το Snort, ένα σύστημα για την ανίχνευση εισβολέων, που αποτελεί ασχήμη της τεχνολογίας.

2.1 Εισαγωγή στους Επεξεργαστές Δικτύων

Οι Επεξεργαστές Δικτύων (Network Processors - ΕΔ) είναι ειδικά σχεδιασμένα ώστε να εκτελούν αποτελεσματικά δικτυακές εφαρμογές. Δεδομένης της ραγδαίας εξέλιξης των δικτύων (1 Gb - 10 Gbit δίκτυα) και των εφαρμογών που χρειάζονται επεξεργασία ανα πασίγνωστα, οι ΕΔ γίνονται μια αρκετά δημοφιλής λύση. Αυτή την στιγμή δεν είναι λίγες οι εταιρίες που ασχολούνται με ΕΔ. Ενδεικτικά μερικές εταιρίες που διαθέτουν επεξεργαστές δικτύων είναι οι Intel[18], IBM[16], Agere[1], AMCC[2], EZ Chip[3] και Teja[33].

Οι ΕΔ γενικά συνδέουν το χάσμα μεταξύ των Ολοκληρωμένων Κυκλωμάτων Ειδικής Εφαρμογής - OKEE (Application Specific Integrated Circuits (ASICs)) και των Γενικών Σχετικών Επεξεργαστών (ΓΣΕ). Τα OKEE είναι παράδοσιοι εξαιρετικά αποδοτικοί και γρήγοροι, ομως κατασκευάζονται αρκετά δύσκολα και με πολύ υψηλό κόστος. Επίσης είναι δύσκολο να προσαρμοστούν σε καινούριες προδιαγραφές μετα και το θα λογικής λειτουργίας. Από την άλλη μεριά, οι ΓΣΕ είναι ιδανικά ευέλικτοι και μπορούν να υλοποιήσουν οποιαδήποτε λειτουργικήτητα. Υποτελούν ομως κατά πολύ σε απόδοση. Η Eucon 2.1 απεικονίζει την σχετική θέση των τριών εννοιών ενώλου (flexibility) και την απόδοση (performance). Οι ΕΔ προσφέρουν την καλύτερη ισορροπία μεταξύ
ΣΧΗΜΑ 2.1: Απόδοση (Performance) και Ευελιξία (Flexibility) για διαφορετικές σχεδιασμένες λύσεις

απόδοσης και ευελιξίας.

Η ευελιξία των ΕΔ στηρίζεται στο ότι μπορούν να προγραμματίσουν όπως και οι ΓΣΕ. Επομένως, το λογισμικό που εκτελούν μπορεί ευελιξιά σαν και γραμμική ή να εκτελεί τη διαδικασία ως τον διαδικαστή ξεκινήσεις και αναπτύξεις τεχνολογίες.

Η απόδοση τους ορίζεται στο ότι εκμεταλλεύονται το ότι τα πακέτα μπορούν να επεξεργασθούν παράλληλα. Η παράλληλη επεξεργασία πακέτων υπο-στηρίζεται από επιπλέον κοιτάδες επεξεργασίας που υπάρχουν υλοποιημένες στο σύστημα. Για παράδειγμα, ο IXP 1200 ΕΔ [17] διαθέτει εξί απλές μηχανές που μπορούν να εκτελούν παράλληλα κατά τόμημα κάθε ελλιμών σε διαφορετικά πακέτα. Επιπλέον στο ίδιο κοιμάται κυκλικό βρισκόνταν και άλλες κοιτάδες (μηκές μήνες, κυκλικά μήνες, καθεμιές, και οικοδομή μετατροπή, και) με αποτέλεσμα ο IXP 1200 να μπορεί να επεξεργάζεται πακέτα με την μέγιστη ταχύτητα ενός συνδέσμου.

2.2 Ο Επεξεργαστής Δικτύων IXP 1200

Σε αυτή την εργασία χρησιμοποιήσαμε τον IXP 1200 για διάφορους λόγους:

- Ο IXP 1200 είναι μία δημοφιλής πλατφόρμα για έρευνα πάνω στα δίκτυα.
- Υπάρχουν δυνατά και ώριμα εργαλεία για την ανάπτυξη λογισμικού.
- Η αρχιτεκτονική του IXP 1200 παραμένει η ίδια και στα καινούρια μοντέλα ΕΔ της Intel.

2.2.1 Αρχιτεκτονική Όλης του IXP 1200

Ο IXP 1200 αποτελείται από τα παρακάτω στοιχεία [19] (Εικόνα 2.2):
ΣΧΗΜΑ 2.2: Αρχιτεκτονική Ύλικού του IXP 1200

- Έναν Γενικό Σχηματικό Επεξεργαστή, τον StrongArm.
- Ένα ασήμαντο επεξεργαστή μικρο-μηχανές (μηχανές ή uEngines).
- Μία μονάδα μνήμης SDRAM ώστε να υπάρχει επικοινωνία με εξωτερική μνήμη SDRAM.
- Μία μονάδα μνήμης SRAM ώστε να υπάρχει επικοινωνία με εξωτερική μνήμη SRAM.
- Μία μονάδα PCI ώστε να υπάρχει επικοινωνία με δίκυκλο PCI.
- Μία μονάδα για Hashing.
- Το Scratchpad το οποίο είναι τέσσερα kilobytes εσωτερικής μνήμης SRAM.
- Μία μονάδα διαύλου IX ώστε να υπάρχει επικοινωνία με εξωτερικές συσκευές (π.χ. Κάρτες Δεδομένων).
- Μία μονάδα FBI η οποία ρυθμοποιεί την μονάδα διαύλου IX, το scratchpad και την μονάδα για hashing.
- Έναν δίκυκλο διεπικοινωνιών για την επικοινωνία εσωτερικά του πυρήνα.
Ο StrongArm είναι ενας γενικού σκοπού επεξεργαστής που στηρίζεται στην αρχιτεκτονική του επεξεργαστή ARM. Μπορεί να λειτουργεί στα 232 MHz και έχει 16 KByte χρωμίας μήνυσης δεδομένων και 8 KBytes χρωμίας μήνυσης εντολών. O StrongArm χρησιμοποιείται συχρόνως για την αρχικοποίηση του συστήματος, για να χειριστείται ιδιαίτερα πακέτα, για να επανακαθορίζει με άλλους επεξεργαστές, κ.α.

Οι μηχανές χρησιμοποιούνται για την συνηθισμένη επεξεργασία πακέτων. Κάθε μηχανή λειτουργεί στην ίδια συχνότητα με τον StrongArm και διαθέτει έναν μεγάλο αριθμό καταχωρητών. Τις περιγράφουμε περισσότερο λίγο παρακάτω.

Η μονάδα SDRAM χρησιμοποιείται για την επανακαθορίση μηνήματος. Η επανακαθορίση γίνεται πάνω από ένα δίκυκλο των 64 bit που λειτουργεί στην συχνότητα των 116 MHz. Το απότελεσμα είναι ρυθμός επανακαθορίσης προς την μία κατεύθυνση ισός με 7.4 Gbit/s. Η μονάδα SRAM χρησιμοποιεί δίκυκλο πλάτους 32 bit και ίσας συχνότητας, οπότε και ο ρυθμός επανακαθορίσης προς την μία κατεύθυνση είναι ο μησός. Σε σχέση όμως με την SDRAM, η SRAM έχει πολύ μικρότερη καθυστέρηση στην ανταπόκριση. Τέλος υπάρχει μέσα στο chip και μία πολύ μικρή μηνή, το scratchpad, στην οποία η πρόσβαση γίνεται μεσω ενός διακόπτη των 232 MHz.

Η μονάδα διακόπτων IX έχει πλάτος 64 bits και λειτουργεί σε συχνότητα 104 MHz. Μπορεί να χρησιμοποιηθεί με δύο τρόπους, είτε όλα τα bits να έχουν την ίδια κατεύθυνση, είτε τα μικά να πηγάζουν προς τα έξω και τα υπόλοιπα προς τα μέσα. Στην πρώτη περίπτωση, αν έχουμε κίνηση μονο προς την μία κατεύθυνση τότε η απόδοσή του είναι 6.6 Gbit/s. O δίκυκλος αυτός χρησιμοποιείται για την επανακαθορίση με εξωτερικές συσκευές όπως χάρτες δικτύων ή άλλους IXP.

Ο εσωτερικός δίαυλος που ενώνει όλα τα παραπάνω ψάχνεται στην Εικόνα 2.3. Τα βασικά του χαρακτηριστικά είναι:

- Διαφορετικοί (υπο-) δίαυλοι για την μηνήμη μεγάλης καθυστέρησης και για την μηνήμη μικρής καθυστέρησης.
- Διαφορετικοί (υπο-) δίαυλοι για γράφημα και διάβασμα.
- Όλοι οι δίαυλοι μοιράζονται μεταξύ των μηχανών και του StrongArm.

Η πρόσβαση στους κοινόχρηστους διακόπτους ελέγχεται από τον Διατητή Εντολών του διακόπτη (Command Bus Arbiter). Σε κάθε μηχανή υπάρχει μία μικρή σειρά FIFO (command FIFO) η οποία κρατάει δύο εντολές που έχει εκτελεστεί μια μηχανή. O Διατητής Εντολών του διακόπτη επιλέγει ποια εντολή θα εκτελεστεί από τις εξί FIFO με βάση:

- Τον τύπο της εντολής. Διαφορετικές εντολές έχουν διαφορετική προτεραιότητα. Πη όλες εντολές που προσπελάσουν την μονάδα FBI έχουν χαμηλότερη προτεραιότητα από τις εντολές που προσπελάσουν την SRAM.
Σχήμα 2.3: Η Αρχιτεκτονική Εσωτερικού Διαύλου του IXP 1200

- Εναν ολόγραμμο round robin μεταξύ των μηχανών.
- Back pressure σήματα που έρχονται από τις μονάδες.

Κάθε μονάδα (πχ SRAM, SDRAM, και) κρατάει μία ουρά από εντολές που πρέπει να εξυπηρετηθούν. Όταν οι ουρές γεμίζουν τότε η μονάδα ενεργοποιεί τα σήματα backpressure και δεν δέχεται πλέον καινούριες εντολές. Εάν, χρειάζεται προσοχή το πρόγραμμα στα σημεία που έχει τέτοιες εντολές ώστε να μην καλλιεργηθεί κάποια μηχανή. Ενα κάλυμμα το προσωπικό όταν ο κώδικας που εκτελεί η μηχανή χρειάζεται να εκτελέσει μία τέτοια εντολή και η τοπική FIFO μνήμη είναι γεμάτη. Τοτε η μηχανή θα πρέπει να περιμένει ωστόσο ο Διαμεταφορικός Εντολών του Διαύλου να πάρει μία από τις εντολές, ελευθερώνοντας μία θέση.

2.2.2 Οι μηχανές

Κάθε μηχανή είναι ένας απλοποιημένος επεξεργαστικός μηχανής αποκλειστικά σε εντολές (Reduced Instruction Set CPU - RISC). Η εκτέλεση των εντολών γίνεται σε πέντε στάδια (five stage pipeline). Η αρχιτεκτονική εντολών υποστηρίζει αποκλειστικό γερμανικό σειρά εντολών παιχνίδι (πχ υπάρχει shifter, μονάδα εύρεσης ενεργοποιημένου bit, κατε.) Ο πολλαπλασιασμός δεν υποστηρίζεται απευθείας από το υλικό και η αρμιθμητική κινητής υποδομής δεν υποστηρίζεται καθόλου.

Από το υλικό υπάρχει υποστήριξη για τέσσερις ίνες (threads) εκτέλεσης. Ο κώδικας κάθε ίνας είναι ιδιωτικό και βρίσκεται στην μνήμη κώδικα της μηχανής, ομως για κάθε ίνα εκτέλεσης υπάρχουν διαφορετικοί καταχωρητές.
κατάστασης (τη program counter, καταχωρητής συμβάντων, και). Η αλλαγή
ekλέψης κώδικα από την μία στην άλλη γίνεται εξολοκλήρου από το υλικό
cαι ελέγχεται από τον ΔΙΕ (Διαβάσαση-Γραφήμα) Εκτέλεσης. O

ΔΙΕ υπάρχει σε κάθε μ.θ. για και εκτελεί έναν από το σύνολο round robin
metαξύ των ενεργών κώδικων της μ.θ. Αρχεία ένας χώρος της μ.θ. για να
γίνει η αλλαγή πλασίου εκτέλεσης και έτσι μία ήλεκτέλεση να κατακληθεί
gια να ξεκινήσει μία άλλη.

Για να υποστηρίζεται πιο αποδοτικά η επικοινωνία και η συγχρονισμός με-
tαξί των κώδικων εκτέλεσης που βρίσκονται σε όλες τις μ.θ. το σετ εντο-
λών διαθέτει αρκετές εισόδημες εντολές. Παραδείγματα είναι οι εντολές ατομικού
eλέγχου και ενεργοποίησης (atomic test and set) και οι εντολές κλειδώματος -
ξεκλειδώματος συγκεκριμένων περιοχών μνήμης. Επιπλέον, υπάρχουν εντολές
που βοηθούν στην συντήρηση υφάνσεων που βρίσκονται στην μνήμη SRAM.

Σε κάθε μ.θ. βρίσκονται συνολικά 128 καταχωρητές των 32-bit γενι-
κής χρήσης (διαβάσαση-γραφήμα) και 128 καταχωρητές των 32-bit για μεταφο-
ρές δεδομένων από και προς τις διάφορες μνήμες. Κάθε ίνα εκτέλεση σε μια
μ.θ. παίρνει το ένα τέταρτο των καταχωρητών χρησιμοποιώντας σχετική
διεύθυνση δότησης. Χρησιμοποιώντας απώλεια διευθυνσιοδότηση μαίνται
εκτέλεσης μπορεί να χρησιμοποιηθεί σε οποιονδήποτε καταχωρητής (ετσι μπορούν να
δημιουργηθούν κοινές χρήσης καταχωρητές για την ενδοεπικοινωνία μέσα σε
μία μ.θ.).

Η προσπέλαση στους κοινοχρήστους πόρους (μονάδες μνήμης, hashing,
και) γίνεται μέσω αναφορών. Τέτοιες αναφορές για παράδειγμα είναι το 'γράφε
δεδομένα', 'διαβάζε-δεδομένα', 'υπολογίσεις hash' και άλλα. Η εκτέλεση μιας
αναφοράς από μία ίνα μπορεί προαιρετικά να την θέσει από κατάσταση εκτέλε-
σης σε κατάσταση αναμονής μέχρι ότου η αναφορά να ολοκληρωθεί. Κατά την
dιάρκεια που η συγκεκριμένη ίνα εκτέλεσης βρίσκεται σε αναμονή κάποια άλλη
ινα εκτέλεσης μπορεί να ενεργή στην μ.θ. Εννοιολογικά, η ίνα εκτέλε-
σης μπορεί να παραμένει ενεργή και να ειδοποιηθεί, μέσω ενός σηματού που
ελέγχει περιοδικά, ότι η συγκεκριμένη αναφορά ολοκληρώθηκε.

Οι μ.θ. μπορούν να διαβάζουν και να γράφουν δεδομένα απευθείας
από και προς τις μνήμες SRAM, SDRAM και Scratchpad. Δεν μπορούν όμως
να ανταλάβουν δεδομένα μεταξύ τους. Αντί αυτού, τα μνήμητα τους πρέ-
πει να αντιλαμβάνονται μέσω κάποιας κοινοχρήσης μνήμης. Αυτό είναι αρκετά
dιαπιστεύει, για παράδειγμα το διάβασμα 4 bytes από την SRAM παίρνει 22 κύ-
ώλους.

Η μεταφορά πακέτων από τους ενταμειτές που βρίσκονται στην πόρτα ει-
σόδου της κάρτας δικτύου στην SDRAM ή στην SRAM ή στο IXP ο
γίνεται εξολοκλήρου με ευθύνη του κώδικα που τρέχει στις μ.θ. Δηλα-
δή, οι μ.θ. θα πρέπει να ελέγχουν περιοδικά την στάθμη του ενταμειτή
eισόδου και να ξεκινούν την μεταφορά δεδομένων όταν το κρίνουν σκόπιμο. Η
αποστολή και η λήψη πακέτων περιγράφεται σύντομα λίγο πιο κάτω.
Κάθε μηχανή έχει 2 KB μνήμης εντολών. Όλες οι ινές εκτέλεσης που
βρίσκονται στην ίδια μηχανή μοιράζονται την ίδια μνήμη εντολών και άρα την
ιδία κώδικα. Ομως μεταξύ των μηχανών ο κώδικας που εκτελείται μπορεί να
είναι τελείως διαφορετικός. Η μνήμη εντολών κάθε μηχανής είναι προσπελά-
σημα από τον StrongArm μέσω κατάληξης διενεργειακής μνήμης. Ετσι
η φόρτωση προγραμμάτων γίνεται με το γράφιμο τους σε κατάληξες θέσεις
μνήμης, χρησιμοποιώντας τον StrongArm.

Είναι ξεκάθαρο ότι ο προσανατολισμός της αρχιτεκτονικής των μηχανών
είναι τέτοιος ώστε να αποκτήσει τις διάφορες καθυστερήσεις λόγω προσπέλα-
σης μνήμης. Επίσης η επεξεργαστική τους δύναμη έγινε στο ότι μπορούν
να διαχειρίζονται αποτελεσματικά bits και σκεπαίκ να χει το στο να υπ-
ωλιγίζουν πολύπλοκες παρατάσεις.

### 2.2.3 Περι Αποστολής και Λήψης Πακέτων

Η λήψη και αποστολή πακέτων με τον IXP είναι αρχετά πολύπλοκη και
εισάγει ορισμένους περιορισμούς στην μορφή του κώδικα που τρέχει στις μη-
χανές. Για το λόγο αυτό περιγράφουμε με συντομία την διαδοχικά λήψης
και αποστολής πακέτων. Κατά την διάρκεια αποστολής και λήψης τα πακέτα
χωρίζονται σε μονάδες των 64 bytes οι οποίες λέγονται μΠακέτα.

#### Λήψη Πακέτων

Η λήψη μΠακέτων στον IXP μπορεί να γίνει με δύο διαφορετικές μεθόδους
που λέγονται λειτουργία Αργής Πόρτας και λειτουργία Γρήγορης Πόρτας.
Η λειτουργία Αργής Πόρτας χρησιμοποιείται σε συνδέσμους των 100 Mbit/s, ενώ
η λειτουργία Γρήγορης Πόρτας σε συνδέσμους του 1 Gbit/s. Η τελευταία
αισθάνεται από την πόρτα εισόδου της χάρτας δεσιών να έχει κάποιες ιδιαίτερες
δυνατότητες.

Η Εικόνα 2.4 συνοψίζει τα απαραίτητα βήματα για την λήψη ενός μΠακέ-
του χρησιμοποιώντας λειτουργία Αργής Πόρτας. Η Μηχανή Κατάστασης Λή-
ψης (Receive State Machine), παρακολουθεί την ένδειξη έτοιμη λήψη (receive
ready) της πόρτας εισόδου. Η ένδειξη αυτή ενεργοποιείται όταν υπάρχουν αρ-
χετά δεδομένα στον ενταμευτή της πόρτας εισόδου.

Όταν η ένδειξη ενεργοποιείται, η ίνα εκτέλεσης ξεκινά μία αίτηση λήψης.
Η αίτηση λήψης είναι μία αναφορά που στέλνεται στην μονάδα FBI και πισω
συχνές στην Μηχανή Κατάστασης Λήψης. Η αίτηση λήψης περιγράφει από
ποιά πόρτα εισόδου θα πρέπει να μεταφερθούν δεδομένα και σε ποιο μέρος του
ενταμευτή λήψης θα πρέπει να αποθηκευτούν. Ο ενταμευτής λήψης (όπως και ο
ενταμευτής αποστολής) βρίσκονται μέσα στην μονάδα FBI. Η Μηχα-
νή Κατάστασης Λήψης εξυπηρετεί τέτοιες αίτησεις που έρχονται από όλες τις
 ΒΙΒΛΙΟΘΗΚΗ
ΣΧΗΜΑ 2.4: Διαδικασία λήψης με την λειτουργία Άργης Πόρτας

ίναι εκτέλεσες. Για κάθε αίτηση, μεταφέρει τα δεδομένα και όταν η μεταφορά ολοκληρωθεί ενημερώνει την αντιστοιχή ίνα εκτέλεση. Η ενημέρωση γίνεται βάζοντας πληροφορίες σχετικά με την μεταφορά των δεδομένων σε μία συγκεκριμένη ουρά και κατόπιν στέλνοντας ένα σήμα στην ίνα εκτέλεση.

Όταν η ίνα εκτέλεση λάβει την εισόδο πολής, θα πρέπει να διαβάσει τις πληροφορίες σχετικά με την μεταφορά των δεδομένων. Μεταξύ άλλων αυτές οι πληροφορίες αναφέρουν από πού πόρτα ήρθαν τα δεδομένα, που έχουν τοποθετηθεί προσωρινά και πόσα bytes ήρθαν.

Όταν αυτές οι πληροφορίες διαβαστούν και επεξεργαστούν, η ίνα εκτέλεση θα στέλει στην μονάδα FBI μία καινούρια αίτηση (αναφορά), για την μεταφορά των δεδομένων επί προς την μνήμη SDRAM επί τοπικά στους καταχωρητές της ίνας. Ο ενταμευτής λήψης (οπως και ο ενταμευτής αποστολής) είναι προσωρινό χώρο αποθήκευσης.

Είναι αξιοσημείωτο ότι υπάρχει μεγάλη χαλκυσμομένη οποία από την ιστιμή που η πόρτα θα ενεργοποιηθεί την έννοια ετοιμή λήψης μέχρι την ιστιμή που η έννοια αυτή θα διαβάσει από την ίνα εκτέλεση και η αντίστοιχη αίτηση λήψης θα φτάσει την Μηχανή Κατάστασης Λήψης. Με άλλα λόγια υπάρχει σημαντική χαλκυσμομένη από την ιστιμή που αρχετά δεδομένα έχουν φτάσει στην πόρτα εισόδου, μέχρι αυτά τα δεδομένα να μεταφερθούν εσωτερικά στον IXP ή στην SDRAM. Όταν λειτουργούμε στο 1 Gbit/s, αυτή η χαλκυσμομένη είναι αρχετή για να υπερπροπήλατη ο ενταμευτής της πόρτας εισόδου. Για το λόγο
αυτό χρησιμοποιείται η λειτουργία Γρήγορης Πόρτας.

Για την αντιμετώπιση της μεγάλης καθυστέρησης που υπάρχει στην λειτουργία Αργής Πόρτας, η λειτουργία Γρήγορης Πόρτας εισάγει την έννοια της υποθετικής αίτησης λήψης πακέτου. Μία ινα εκτέλεσης χώσκει, εκδίδει μία υποθετική αίτηση λήψης πακέτου χωρίς πρώτα να έχει ελέγξει την ένδειξη 'έτοιμη λήψη' της αντίστοιχης πόρτας εισόδου.

Όταν η Μηχανή Κατάστασης Λήψης επεξεργάζεται τέτοια αίτηση, ελέγχει αν η πόρτα έχει δεδομένα ή δεν έχει. Αν εχει, τότε συνεχίζει την επεξεργασία της αίτησης κανονικά, όπως θα έκανε και στην λειτουργία Αργής Πόρτας. Αν όμως δεν υπάρχουν δεδομένα, τότε αφαίρεται την υποθέτηση αίτηση. Η αφαίρεση της υποθέτησης αίτησης γίνεται θέτοντας κατάλληλα τις πληροφορίες που αφορούν την αίτηση και στέλνοντας κανονικά σήμα. Η ινα εκτέλεσης που εστελεί την αίτηση θα διαβάζει τις από τις πληροφορίες σχετικά με την αίτηση ότι αυτή αφαίρεθηκε.

Η Μηχανή Κατάστασης Λήψης, ελέγχει αν υπάρχουν δεδομένα στον ενταμειωτή της πόρτας εισόδου μέσα ενός ιδιαίτερου σήματος που πρέπει να εξάγει η πόρτα εισόδου, το σήμα 'γρήγορης ένδειξης έτοιμης λήψης'.

Σε κάθε λειτουργία, Αργή είτε Γρήγορη, κάθε μΠακέτο μαρκάρεται ως Αργή Πακέτου (Start Of Packet - SOP), Τέλος Πακέτου (End Of Packet - EOP) ή τίποτα από τα δύο. Επιπλέον κάθε μΠακέτο παίρνει έναν αύξοντα αριθμό μεταξύ 0 και 15. Κάθε ΑΠ-μΠακέτο παίρνει επίσης έναν αύξοντα αριθμό μεταξύ
0 και 15. Αυτές οι πληροφορίες δίνονται στην ίνα εκτέλεσης αμέσως μετά την εξυπηρέτησή της αίτησης λήψης.

Αποστολή Πακέτων

Αντίθετα με την λήψη πακέτων, η αποστολή είναι ιδια για κάθε τύπο πόρτας. Κατά την αποστολή, τα μήνυμα είναι λιγότερα και πιο απλά.

Τα δεδομένα αποστέλλονται από την ενταμευτή αποστολής που βρίσκεται μέσα στην μονάδα FBI. Ο ενταμευτής αποστολής έχει 16 θέσεις για 16 μΠακέτα. Τα μΠακέτα γράφονται πρώτα σε αυτές τις θέσεις και κατόπιν αντιγράφονται στον ενταμευτή της πόρτας εξόδου από οποιου και αποστέλλονται.

Οι ίνες εκτέλεσες μπορούν να προσπελάσουν κάθε θέση του ενταμευτή αποστολής τυχαία. Ομως η αντιγραφή από αυτών στον ενταμευτή της πόρτας εξόδου είναι απόλυτα σειρασια. Υπάρχει ένας δείκτης, ο δείκτης αποστολής, που δείχνει πόντα στην επόμενη θέση που θα αντιγραφεί στον ενταμευτή της πόρτας εξόδου.

Αυτή η τελευταία μεταφορά ξεκινάει μελώς η συγκεκριμένη θέση μαρκαριστεί έγχυση. Τότε, αφού το αντίστοιχο μΠακέτο αντιγραφεί στον ενταμευτή της πόρτας εξόδου, ο δείκτης αποστολής θα προχωρήσει στην επόμενη θέση. Αν η θέση μαρκαριστεί ως λανθασμένη, αντί για έγχυση, τότε ο δείκτης θα προχωρήσει στην επόμενη θέση χωρίς να γίνει αντιγραφή μΠακέτου.

Πριν μαρκαριστεί μία θέση ως έγχυση, θα πρέπει να εξασφαλιστεί ότι υπάρχει αρκετός διαλέγοντας χώρος στον ενταμευτή της πόρτας εξόδου. Αυτό επιτυγχάνεται παρακαλούμενως περιοδικά την τιμή της ένδειξης ετοιμή για αποστολή της αντίστοιχης πόρτας.

2.2.4 Πλατφόρμες Υλικού για Ανάπτυξη Κώδικα

Υπάρχουν διάφορες πλατφόρμες υλικού για την ανάπτυξη κώδικα για τον IXP. Οι κυρίως κτητορικοί κατασκευαστές είναι η Intel και η Radisys. Οι πλαστές που κατασκευάζονταν αποτελούντα από:

- Το τσιπάκι του IXP 1200,
- Mnήμες (SDRAM, SRAM),
- Ελεγκτές Δικτύου (Network Interface Controllers),
- Πόρτες Δικτύου (Network Physical Interfaces (ports)),
- Διεπαφή για τον δίαυλο PCI (PCI interface).
ΣΧΗΜΑ 2.6: Η πλακέτα ENP-2506 από την Radisys

Οι ακριβείς προδιαγραφές για κάθε πλακέτα εξαρτώνται από τον κατασκευαστή και το μοντέλο. Για παράδειγμα, η πλακέτα της Radisys, ENP-2506 (Εικόνα 2.2.4), περιλαμβάνει 256 MB SDRAM, 8 MB SRAM, 8 MB μνήμη Flash, δύο οπτικές διεπαφές για δίκτυο gigabit και μία 64 bit διεπαφή για διάλυμα PCI. Το τσιπακι του IXP είναι χρονισμένο στα 232 MHz. Αντίστοιχα η πλακέτα της Intel έχει 32 MB SDRAM, 2 MB SRAM, 2 MB μνήμη Flash, δύο οπτικές διεπαφές για δίκτυο gigabit και οκτώ διεπαφές για δίκτυο Fast Ethernet. Το τσιπακι του IXP είναι χρονισμένο στα 200 MHz.

Συνήθως, οι πλακέτες αυτές έχουν κάποιο εμπορικό λειτουργικό σύστημα όπως το VxWorks. Το Linux ομοίως έχει αρχίσει και διαδίδεται και στον IXP 1200 και αυτό διευκολύνει ιδιαίτερα την ανάπτυξη κώδικα για τον StrongArm.

2.3 Συστήματα Ανίχνευσης Εισβολέων

Για έναν οργανισμό, τα Συστήματα Ανίχνευσης Εισβολέων (ΣΑΕ) προσφέρουν αυξημένη ασφάλεια με το να παρακολουθούν συνεχώς την κίνηση που εισέρχεται στο δίκτυο του οργανισμού. Συνήθως τοποθετούνται στις άκρες του δικτύου όπου φαίνεται στην Εικόνα 2.7.

Τα ΣΑΕ εξετάζουν την κίνηση και για κάθε πακέτο απορριστικού ονοματολογεί μέρος μιας τιμής επιθέσεων / εισβολέων. Ο έλεγχος που κάνουν αφορά και τις επικεφαλής του πακέτου άλλα και το περιγράμμα και γίνεται με βάση περιγραφές επιθέσεων (κανόνες ή υπογραφές επιθέσεων). Για την ανίχνευση επιθέσεων σε διάφορα δικτυακά επίπεδα, τα ΣΑΕ συχνά κρατούν κατάσταση ως προς κάποιες τρέχουσες συνθήκες.

Η λίστα με τους οργανισμούς και τις εταιρείες που εισδοχεύονται στον τομέα αυτόν είναι αρκετά εκτενής ([28]). Για παράδειγμα αναφέρουμε μερικές εταιρείες
ΣΧΗΜΑ 2.7: Τοποθέτηση ενός Συστήματος Ανάλυσης Εισβολών


2.4 Snort

Η επιλογή του Snort έγινε κυρίως για δύο λόγους. Πρώτων, αναπτύσσεται συνεχώς από πολλούς προγραμματιστές και αποτελεί στήρις της στημή αρχή της τεχνολογίας στον συγκεκριμένο τομέα. Δεύτερον, σε αντίθεση με άλλα αντίστοιχα συστήματα, είναι ανοικτό κώδικα. Αυτό μας επέτρεψε να το εξετάσουμε και να πειραματιστούμε περισσότερο.

Το Snort παρέχει ως είσοδο περιγραφές επικυμάνων πακέτων (υπογραφές). Αυτή την στιγμή υπάρχουν περίπου 1500 τέτοιες υπογραφές. Το Snort παρακολούθησε την κίνηση και προσπαθεί να παίξει πακέτα που ταραώνουν σε τουλάχιστον μία από αυτές τις υπογραφές. Αν βρεθεί κάποιο ταίριασμα τότε η αντίστοιχη αντίδραση εκτελείται. Η αντίδραση σε μία περιγραφή ορίζεται και αυτή στην υπογραφή της. Δεδομένου του ότι η απόδοση είναι πολύ σημαντική, το Snort οργανώνει τις υπογραφές επωτερικά με τέτοιο τρόπο ώστε να μπορεί γρήγορα, για κάθε πακέτο να βρίσκει αν ταραώνει σε κάποια υπογραφή.

Κάθε υπογραφή είναι και μία περιγραφή ενός πιθανότατα επικυμάνου πακέτου. Οι υπογραφές του Snort βασίζονται στο tcpdump [32] και είναι αρκετά απλές. Όπως φαίνεται και στην Εικόνα 2.8 αποτελούνται από τρία μέρη, την αντίδραση (action), την επεκτελομένη (header) και τις επιλογές (options). Η Εικόνα 2.9 δείχνει ένα σετ από υπογραφές που χρησιμοποιείται ως μέρος της
### 2.4. SNORT

```plaintext
alert icmp $ENET any -> $HNET any (content:"|495353504e475251|"; itype:8; depth:32;)
```  
**ACTION:** possible values include `alert`, `log`, `dynamic`

```plaintext
icmp $ENET any -> $HNET any
```

**HEADER:** defines values of protocol, source and destination IPs and ports

```plaintext
(content:"|495353504e475251|"; itype:8; depth:32;)
```

**OPTIONS:** defines values of other protocol fields, including payload searches

### ΣΧΗΜΑ 2.8: Δομή μικροσκοπής του Snort

# All rights reserved.
#—
# ICMP RULES
#——
# Description:
# These rules are potentially bad ICMP traffic. They include most of the
# ICMP scanning tools and other "BAD" ICMP traffic (Such as redirect host)
#—
# Other ICMP rules are included in icmp-info.rules
alert icmp $ENET any -> $HNET any (itype:5;icode:1; )
alert icmp $ENET any -> $HNET any (itype:5;icode:0; )
alert icmp $ENET any -> $HNET any (itype: 4; ide: 0; )
alert icmp $ENET any -> $HNET any (itype: 8; icmp_id: 0; icmp_seq: 0; dsizel4; )
alert icmp $ENET any -> $HNET any (content:"—495353504e475251—"; itype:8; depth:32;)
alert icmp $ENET any -> $HNET any (itype: 8; icode: 0; depth: 32; )
alert icmp $ENET any -> $HNET any (content: "ABCDFGHJKLMNOPQRSTUVWXYZWABCDGFHGI"; itype: 8; icode: 0; depth: 32;)
alert icmp $ENET any -> $HNET any (dsizel20; itype: 8; icmp_id: 0; icmp_seq: 0; content: "—0000000000—"; )
alert icmp $ENET any -> $HNET any (dsizel20; itype: 8; )
alert icmp $ENET any -> $HNET any (id: 666; dsize: 0; itype: 8; icmp_id: 666 ; icmp_seq: 0; )
alert icmp $ENET any -> $HNET any (content:"—0000000000000000—"; itype: 8; dsize:8; )
alert icmp $ENET any -> $HNET any (ipopts: rr; itype: 0; )
alert icmp $ENET any -> $HNET any (content:"—00 00 00 00 45 45 45 45 45 45 45—"; itype: 8; icode: 0; )
alert icmp $ENET any -> $HNET any (content:"—3839 3a3b 3c3d 3e3f—"; depth: 100; itype: 8; )
alert icmp $ENET any -> $HNET any (content:"—544a50696e6750726f206279204a696d—"; itype:8;depth:32; )
```

### ΣΧΗΜΑ 2.9: Παράδειγμα αρχείου υπογραφών για το Snort
εισόδου για το Snort.

Η επεξεργαλίδα αποτελείται από το πρωτοκόλλο, την διεύθυνση τηγής IP, τον αριθμό της αρχικής πόρτας καθώς και την διεύθυνση προορισμού IP και τον αριθμό τέλους πόρτας. Η Εικόνα 2.8 δίχως την δομή παραγράφων. Η λέξη χλειδί αποχωρισμοί εκεί που δεν μας ενδιαφέρει τίμη έχει το πεδίο.

Το μέρος των επιπλογών, μεταξύ των παρενθέσεων, ορίζει όλα τα υπόλοιπα μέρη ενός πακέτου. Πάνω επιπλογών τριοδιαλεκτικό, περιεχόμενα πακέτου, και. Η αντίδραση ορίζει τι θα πρέπει να κάνει το Snort σε περίπτωση που θεωρεί ένα τέτοιο πακέτο. Ενα alert (συναγερμός) σημαίνει ότι θα πρέπει στη συγκεκριμένη διεύθυνση ενός το Snort είναι αρχετά ασφαλή.

Ελλάδας συνηθισμένα τα αρχεία εισόδου ενός SAE βασισμένο στο Snort να περιέχουν γύρω στις 1500 υπογραφές. Κάθε επιπλογόμενο πακέτο θα πρέπει να ελέγχεται με κάθε όλες αυτές τις υπογραφές. Ελλάδας επίσης αξιοσημείωτο κάθε οι περιποιήσης απο αυτές ελέγχους όχι μόνο τις επεξεργαλίδες αλλά και τα περιεχόμενα του πακέτου. Επίσης, η λειτουργία ενός SAE είναι αρχετά ασφαλή.

Στις επόμενες παραγράφους περιγράφουμε πώς οι υπογραφές οργανώνται ώστε να πετυχητούμε το κατά το κατά τον το Snort με χρημάτιση εκείνη που περιγράφεται στην Εικόνα 2.10.

Κατά το ξεκίνημα, όλα το σετ υπογραφών μοιράζεται στα διαδικτύους κανόνων. Κάθε ομάδα κανόνων περιέχει όλες τις υπογραφές για ένα συγκεκριμένο τύπο πακέτων. Ο τόπος του κάθε πακέτου καθορίζεται από το πρωτόκολλο του και τις διευθύνσεις IP. Για παράδειγμα οι κανόνες που αναφέρονται σε κωπηλαρίχα του υποδομή εισόδου (World Wide Web - WWW) περιλαμβάνονται σε δύο διαδικτύους κανόνων: στην ομάδα 'TCP, any source port, destination port 80' και στην ομάδα 'TCP, source port 80, any destination port'. Σελίδα κάθε ομάδα κανόνων περιέχει όλους τους κανόνες που χρειάζεται πάντα πακέτο να ελέγχουν για ένα πακέτο. Κατά το ένα πακέτο θα ελέγχει από ασφαλής μία ομάδα κανόνων.

Για κάθε ομάδα κανόνων, το Snort αρχικοποιεί ξεκινωμένα μία μηχανή ανα-
ξήτησης πολλαπλών μοτίβων χειμένων (multipattern search engine). Οι μη-
χανές αναξήτησης πολλαπλών μοτίβων χειμένων μπορούν αποδοτικά να χρησιμοποιηθούν ώστε να περιέχει κάποιο χειμένα από ένα σετ χειμένων. Το Snort υποστηρίζει διάφορους υποδομές αλγόριθμούς για αναξήτηση πολλαπλών μοτίβων χειμένων [4, 35].

Κάθε πακέτο που το Snort επεξεργάζεται αρχικά κατανοημοποιείται (class-
ification). Δηλαδή, αποφασίζεται σε ποια ομάδα κανόνων ανήκει.
incoming packet

all packet rules (100 %)

classification

all web related rules (40 %)
(protocol : TCP
destination port : 80)

multipattern search

web rules with content contained in current packet (8 %)

header fields

web rules with header fields matched in current packet (4 %)

signature priorities

matched rule

Action

ΣΧΗΜΑ 2.10: Περιγραφή της μηχανής του Snort
Κατόπιν, το Snort χρησιμοποιεί την μηχανή αναζήτησης πολλαπλών μο-
τίβων για να βρεί όλα τα κομμάτια κειμένου που εμπεριέχονται στο πακέτο.
Δεδομένου ότι οι επιθέσεις δεν είναι συχνές, συνήθως το Snort δεν θα βρίσκει
τίποτα. Αν βρει κάποια κομμάτια κειμένου ότι υπάρχουν στο πακέτο τοτε θα
προκαλήσει να ελέγξει τις υπογραφές στις οποίες αντιστοιχούν τα κομμάτια
κειμένου. Αν κάποια υπογραφή ταιριάζει πλήρως, τότε η αντίστοιχη αντίδραση
ενεργοποιείται.
Κεφάλαιο 3

Ανάλυση Επικεφαλίδων με τον IXP 1200

Στο κεφάλαιο αυτό παρουσιάζουμε μια αρχιτεκτονική λογισμικού για τον IXP 1200 η οποία μπορεί με αποδοτικό τρόπο να ανιχνεύει τις επικεφαλίδες πακέτων ώστε να εντοπίζει ανωμαλίες. Η εργασία αποτελείται από δύο μέρη. Το πρώτο είναι μια απλή και αποδοτική υποδομή για την διαχείριση των μονάδων του IXP και την ομαλή λήψη πακέτων. Το δεύτερο είναι το εργαλείο S2I (Snort to IXP), το οποίο παράγει χαμηλού επιπέδου κώδικα ξεκινώντας από υψηλό επιπέδου υπογραφές. Αυτή η αρχιτεκτονική λογισμικού επιτρέπει την χρήση του IXP σε Συστήματα Ανίχνευσης Εισβολέων ενώ ταυτόχρονα τα πειράματα μας έδειξαν ότι ο κώδικας που παράγει είναι εξίσου αποδοτικός με κώδικα γραμμένο με το χέρι.

3.1 Εισαγωγή

Το γεγονός ότι τα ΣΛΕ χρειάζεται να προσαρμόζονται συχνά σε χαμηλή σύνθετα υπογραφές, καθώς και το ότι απαιτούν μεγάλη υπολογιστική ισχύ ώστε να είναι αποδοτικά μας οδήγησε στο να μελετήσουμε την χρήση Επεξεργαστών Διατάξεων (ΕΔ) για να τα υλοποιήσουμε. Οι εργασιακές αυτές διαφέρουν αρκετά από υπηρεσίες όπως η δρομολόγηση πακέτων, οι οποίες έχουν ήδη μελετηθεί με ΕΔ.

Σε αυτό το κεφάλαιο παρουσιάζουμε το εργαλείο S2I που υποδέχεται στην χρησιμοποίηση του επεξεργαστή IXP 1200 σε ΣΛΕ που στρέφονται στο Snort. Η είσοδος του εργαλείου S2I είναι ένα απλό αρχείο με υπογραφές για γνωστές επιθέσεις.

Το S2I θα μετατρέψει αυτό το σε σε αποδοτικό κώδικα για τις μηχανές του IXP. Ο παραγόμενος κώδικας συμπληρώνεται από μία γενική υποδομή για
3.2 Σχετικές Εργασίες

Οι έρευνες σχετικά με εργαλεία και μεθοδολογίες για ΕΔ έχει επικεντρωθεί κυρίως γύρω από εφαρμογές σχετικά με δρομολόγηση πακέτων. Επιτάχυναν δε περισσότερο στην τμηματοποίηση ενός προγράμματος (modularity) και στην επαναχρησιμοποίηση χώρων (re-usability).

Στο [31], ο Spalink και άλλοι χρησιμοποιούν τον IXP 1200 για να φτιάχνουν ενα τμηματοποιητή. Προτείνουν μία αρχιτεκτονική που χωρίζεται σε δύο μέρη. Το ενα είναι μία μόνιμη υποδομή και το άλλο είναι ένα δυναμικά παραγόμενο μέρος. Στο [6], οι συγγραφείς επίσης περιγράφουν την αρχιτεκτονική ενός δρομολογητή ο οποίος είναι αρχετά τμηματοποιημένος.

Ο δυναμικά παραγόμενος χώρος για φιλτράρισμα πακέτων είναι ανταλλακτικό της εργασίας των Engler και άλλων στο [11]. Εκεί παρουσιάζεται ένα εργαλείο που μετατρέπει κανόνες φιλτράρισματος γραμμένος σε μία υψηλού επιπέδου γλώσσα σε χώδεις μηχανής. Κάθε φίλτρο ενσωματώνεται σε ένα δεντροφάκτο. Ο τρόπος που γίνεται η ενσωμάτωση εκμεταλλεύεται το ότι είναι γνωστές οι τιμές που τα φίλτρα εξετάζουν, τεχνική που χρησιμοποιήθηκε και εμπειρία.

Τέλος, o Cho και άλλοι υλοποίησαν στο [8] αυτόματη μετατροπή της αναλυτικής συγκεκριμένων χοματικών χειμένου σε γλώσσα VHDL για τον προγραμματισμό FPGA.
3.3 Το Εργαλείο S2I

Το εργαλείο Snort-to-IXP 1200 (S2I) παράγει micro-C κώδικα για τις μηχανές ξεκινώντας από ένα αρχείο με υπογραφές του Snort. Ο παραγωγικός κώδικας αποτελείται από ένα στατικό και ένα δυναμικό μέρος. Το στατικό μέρος είναι ένας σκελετός ανεξάρτητος από το σετ υπογραφών και περιέχει την υποδομή για την ροή των πακέτων σωστικά στον IXP. Το δυναμικό μέρος παράγεται με βάση ένα σύνολο υπογραφών και είναι αυτού που υλοποιεί τον πραγματικό ελέγχο επαφολίδων και ενεργοποιεί τις αντίστοιχες αντιδράσεις.

Τα πλεονεκτήματα σε χωρο και χρόνο με την χρήση του S2I βασίζονται στην ακόλουθη παρατήρηση. Η διερμηνεία των υπογραφών θα ήταν πιο ακριβής και σε χώρο και σε χρόνο μιας και ο διερμηνέας θα επέτρεπε να ζητεί διερμηνεία όλες τις υπογραφές. Επίσης δεδομένου ότι κάθε υπογραφή περιέχει πολλά πεδία, η συγκεκριμένη μέθοδος θα χρειαζόταν περισσότερη μνήμη. Αντίθετα, με την μετάφραση των υπογραφών μπορούμε να χρησιμοποιήσουμε κοινές τεχνικές ώστε και τα δεδομένα να συμπεστούν σε ισχυρό χώρο (τη χρήση των υπογραφών που έχουν χορηγήσει και επιπλέον ο χρόνος ελέγχων να μειωθεί (χρησιμοποίοντας δενδρόες δομή).

Μάλιστα, μια από τις βασικές βελτιστοποιήσεις είναι η κατασκευή ενός δενδρού που περιέχει όλες τις υπογραφές, και αυτοί οι δακτυλιογραφημένοι σείτες χρησιμοποιούνται για χρήση μαθηματικών θεωρημάτων απλίκων χρήσης. Αν και αυτός ο χρόνος αυτός είναι μικρός και ενιαίος, που προηγματιστά περιήγηση βασισμένη στο σύνολο των υπογραφών, η εφαρμογή μπορεί να είναι εφαρμοστική επίπονη. Αντιθέτα, το εργαλείο S2I την απλοποιεί και προσφέρει εξίσου αποδοτικό κώδικα.

3.3.1 Στατικό Τμήμα

Το στατικό μέρος του παραγωγικού κώδικα αποτελείται από μια ελάχιστη υποδομή για τον διαχειρισμό πακέτων και εννοιολογικό για την κατανομή της επεξεργασίας πακέτων στις μονάδες του ΕΔ. Η τρέχουσα υλοποίηση αφορά δύο σειρές των 100 Mbit/s.

Για τους σκοπούς της συγκεκριμένης σχεδίασης, κάθε πακέτο επεξεργάζεται αποκλειστικά από μία μηχανή. Ο διαμορφωτής του φόρτου θα μπορούσε θεωρητικά να είναι πιο ακριβής. Δηλαδή, μέρος της επεξεργασίας του πακέτου να γίνεται σε μία μηχανή και ένα άλλο μέρος σε μία άλλη, δεδομένου ότι διαφορετικά πακέτα χρειάζονται διαφορετικά χρόνο επεξεργασίας. Εφόσον όμως η επικοινωνία μεταξύ των μηχανών είναι αρκετά ακριβής, ένας τέτοιος αλγόριθμος απορρίπτεται.
Σχήμα 3.1: Μέθοδος ίνας: τα πακέτα διαμορφώνονται στις ίνες της μηχανής. Όλες οι ίνες εκπέμπουν τον ίδιο κώδικα.

Ετσι, κάθε πακέτο ανατίθεται εξ ολοκλήρου σε μία μηχανή και δώσεις έμπας στο να ελαχιστοποιηθούν οι κανόνες τώρα που χρειάζεται μια μηχανή προκειμένου να επεξεργαστεί ένα πακέτο.

Για την κατανόηση της εργασίας μεταξύ των ίνων εκτέλεσης κάθε μηχανής, μελετήσαμε δύο εναλλακτικές λύσεις. Είτε κάθε ίνα εκτέλεσης αναλαμβάνει εξ ολοκλήρου ένα πακέτο, είτε οι ίνες εκτέλεσης μιας μηχανής συνεργάζονται για την επεξεργασία ενός πακέτου. Η πρώτη μέθοδος ονομάζεται 'μέθοδος ίνας' και η δεύτερη 'μέθοδος μηχανής'. Και οι δύο παρουσιάζονται παρακάτω.

Κατα την μέθοδος ίνας, (Εικόνα 3.1) κάθε ίνα μια μηχανής αναλαμβάνει ένα πακέτο. Η μέθοδος αυτή έχει το πλέονέκτημα της απλότητας και του ότι και οι τέσσερις ίνες μοιράζονται τον ίδιο κώδικα. Τα 2 KB της μηχανής εντολών είναι ενοποιημένα και χρησιμοποιούνται από όλες τις ίνες.

Ενα μειονέκτημα της μεθόδου είναι ότι οι καταχωρητές της μηχανής θα πρέπει να μοιραστούν στα τέσσερα. Κάθε ίνα ομας θα πρέπει να ρίξει τοπικά τις επικεφαλίδες του πακέτου της προκειμένου να τις ελέγξει. Ετσι 14 καταχωρητές (για τις επικεφαλίδες του Ethernet, του IP και του TCP) θα είναι καταχωρημένοι για κάθε ίνα. Συνολικά 56 καταχωρητές, ή το 30% των καταχωρητών που μπορούν να διαβάζονταν θα είναι καταχωρημένοι από επικεφαλίδες. Επιπλέον, ερώτων μόνο μία ίνα είναι κάθε φορά ενεργοποιημένη, οι περισσότεροι από αυτούς του καταχωρητής θα είναι μη διαδέχομαι χωρίς λόγο.

Ο συγχρονισμός μεταξύ των ίνων προκειμένου να πάρουν δουλειά γίνεται σε δύο βήματα. Αρχικά, από κάθε μηχανή επιτρέπουμε μία ίνα να είναι ενεργή. Κατόπιν, οι έξι ίνες από τις διαφορετικές μηχανές ανταγωνίζονται για να κλειδώσουν την πόρτα εισόδου. Αυτή που θα τα κατακρεί, παραλαμβάνει το πακέτο, ξεκινεί ο κώδικα και ξεκινάει τον ελέγχο των επικεφαλίδων.

Η εναλλακτική μέθοδος, είναι η μέθοδος της μηχανής όπου ολοκληρώνει η μηχανή αναλαμβάνει τον ελέγχο ενός πακέτου (Εικόνα 3.2). Σε αυτή την περίπτωση οι ίνες έχουν συγκεκριμένες δουλειές όπως την μεταφορά δεδομένων και την εκτέλεση των ελέγχων.
Σχήμα 3.2: Μέθοδος μηχανής: πακέτα κατανεμομένα στις μηχανές. Μια να χάνει τους ελέγχους, ενώ μια άλλη διαχειρίζεται την ουρά εργασίας.

Σε αντίθεση με την μέθοδο τών, η μέθοδος μηχανής έχει μόνο ένα ενεργό πακέτο κάθε σημείο, αποδημευμένο σε καταχωρητές. Ετσι μόνο 14 καταχωρητές κρατούν τις υπο κείται εποχιακές χρήσεις. Το να υπάρχουν περισσότεροι καταχωρητές ελεύθερη αυξάνει την παθονομία το θα δυναμικό μέρος να χρησιμοποιήσει μόνο καταχωρητές για την λειτουργία του. Δηλαδή να μην χρειαστεί κάποια μεταβλητή να αποδημευτεί στην μνήμη. Αν κάτι τέτοιο συμβεί επειδή όλες οι καταχωρητές χρησιμοποιούνται, τότε η απόδοση θα πέσει δραματικά μιας και η προσπέλαση μικρής είναι κατα πολύ πιο αργή από την προσπέλαση ενός καταχωρητή και θα συμβαίνει σε κάθε πακέτο.

Δεδομένου ότι τώρα υπάρχει μόνο ένα πακέτο, αντι για τέσσερα, ενεργό κάθε φορά, χρειαζόμαστε μια ουρά εργασίας στην οποία θα βάζουμε τα υπόλοιπα πακέτα που πρέπει να ελέγχουμε στο μέλλον. Ετσι, η επεξεργασία πακέτων γίνεται τελείως σειρακές, αντί με εναλλαγές όπως στην περίπτωση της μεθόδου τών.

Σε αυτή την μέθοδο χρησιμοποιούμε μία πιο απλή μηχανισμό συγχρονισμού. Κάθε μηχανή περιμένει να πάρει ένα πακέτο και μόλις τα δεδομένα του αρχίσουν να εργούνται άνευ σήμα στην επόμενη για να ξεκινήσει. Ετσι δεν χρειάζονται δομές δεδομένων για το κλείδωμα/ξεκλείδωμα της πόρτας.

Και οι δύο μέθοδοι είχαν παρόμοια απόδοση. Επειδή όμως η μέθοδος της μηχανής χρειάζεται λιγότερους πόρους την διαλέξαμε για το υπόλοιπο της εργασίας.

3.3.2 Δυναμικό Μέρος

Σε αυτό το τμήμα περιγράφουμε το δυναμικό τμήμα, δηλαδή των μεροκώδικα που παράγει το εργαλείο S2I. Ο κώδικας είναι δυναμικός με την έννοια ότι παράγεται αυτόματα από ένα σετ υπογραφών κάθε φορά που ο χρήστης το απαιτεί. Δεν σημαίνει όμως ότι μπορεί να αλλάξει κατά την διάρκεια που το σύστημα λειτουργεί. Αν και κάτι τέτοιο θα ήταν επαρκή, θα στούντιζε σε απόδοση προεξεγενούς να υποστηριχτεί. Από την άλλη, ισως με την χρήση
Σχήμα 3.3: Παράδειγμα Δέντρου

επιλέον υλικό να μπορεί να υποστηρίζει συνεχείς λεπτομέρειες ξεκινά και ταυτόχρονη ενημέρωση των υπογραφών χωρίς μείωση της απόδοσης του συστήματος. Μια τέτοια έρευνα ομως είναι πέρα από τους σκοπούς αυτών της εργασίας.

Το δυναμικό τμήμα βασίζεται σε ενα συγχρεμένο αρχείο εισόδου του Snort. Προς στήριξη, δεν υποστηρίζονται όλες οι δυνατότητες του Snort και πιο συγχρονίζεται δεν υποστηρίζεται ψάζουμε κεμένου. Η λειτουργία του εργαλείου γίνεται σε δύο φάσεις. Άρχικα κατασκευάζεται το δέντρο αναπαράστασης των υπογραφών. Κατόπιν παράγεται ο αντίστοιχος κώδικας micro-C.

Κατασκευή Δέντρου: Αρχικά οι υπογραφές διαβάζονται από το αρχείο εισόδου και μετατρέπονται σε μια απλή εισωτερική δομή. Κατόπιν το εργαλείο S2I ξεκινάει να ενώνει υπογραφές σε μια δενδρική δομή. Κάθε επίπεδο της δενδρικής δομής αντιστοιχεί σε ένα τύπο πεδίου επικεφαλίδας, την αρχική πόρτα. Αυτο φαίνεται στην Εικόνα 3.3 με ένα απλό δέντρο υπογραφών.

Το εργαλείο αρχικα αποτελεί το δέντρο με την πρώτη υπογραφή. Κατόπιν για κάθε επόμενη υπογραφή, ενσωματώνει όσα πεδία μπορεί γραμμικά και γραμμικές παραστάσεις μια προ-διεγγεμένη σειρά επίσκεψης των πεδίων. Νέα χώρισμα προστίθενται όταν σε κάποιο επίπεδο πρέπει να γίνει έλεγχος για μία καιωμαία τιμή. Κάθε κατασκευάζεται το δέντρο, κάθε επίπεδο διατάσσεται έτσι ώστε να βρίσκονται πρώτα οι πιο συγχρονισμένες τιμές. Ετσι οι πιο αυστηροί κανόνες θα ελέγχουν κατά-
τερα από τους πιο γενικούς.

Παραγωγή κώδικα: Σε αυτή την φάση παράγεται ο κώδικας που μαζί με το στατικό μέρος τα μεταφραστικά, βελτιστοποιηθούν και θα μορφώθηκαν στον IXP. Η Εικόνα 3.4 δείχνει ένα μικρό μέρος του κώδικα που παράγει το εργαλείο
3.3. ΤΟ ΕΡΓΑΛΕΙΟ S2I

имп (ETHPROTOL==0x0800 & PROTOCOL==0x6) {
    if (PORT2==0x50) {
        if (IP2==0xa000001) {
            if (IP1==0xa000000) {
                if (DSIZE>0x200) {
/*Action for "tcp 10.0.0.0 any -> 10.0.0.1..." */
            }}
        ctx_swap();
        if (IP2 == 0xa000002) {
            if (IP1==0xa000001 || IP1==0xa000002) {
                if (ACK>0x200) {
/*Action for "tcp [10.0.0.1 10.0.0.2] any..."*/
            }}
        } //<<<<PORT2
    if (PORT2==0x14) {
        if (IP2==0xa00012c) {
            if (DSIZE>0x200) {
/*Action for "tcp any any -> 10.0.0.300 20..."*/
            }}
        ctx_swap();
    }\\\\\n\\\\\\nΣΧΗΜΑ 3.4: Κώδικας με βάση την Εικόνα 3.3

S2I.

Μία σημαντική λεπτομέρεια είναι η χρήση σταθερών στα διάφορα σημεία που γίνεται έλεγχοι. Για παράδειγμα ο έλεγχος αν η τέλικη πόρτα είναι 80 γίνεται με τον κώδικα: 'if (PORT2 == 0x50)' και όχι με κώδικα παρόμοιο με 'if (PORT2 == ports[i])'. Με τον τρόπο αυτό μετάφρασκε τις προσπολύμενες μνήμες που βγάλουν την απόδοση δραματικά. Αυτή η βελτιστοποίηση υπάρχει και στο [11].

Θα πρέπει να σημειώσουμε ότι αν και η εργασία εστιάζει στον IXP ΕΔ, ο κώδικας που παράγεται σε αυτό το στάδιο μπορεί με ελάχιστες αλλαγές να χρησιμοποιηθεί σε άλλον επιχειρηματική για τον οποίο υπάρχει μεταφραστής. Ωστόσο για παράδειγμα για εναν γενικού χοπού επιχειρηματική. Παρασκεύω μάλιστα αναφέροντας τις πρώτες εμπειρίες από μία τέτοια απόδειξη.

Συγκεκριμένα για τον IXP το εργαλείο θα εισάγει στον κώδικα σε συγκεκριμένα σημεία, εντολές για την αλλαγή ίαρχιας εκτέλεσης. Αυτό γίνεται γιατί αν η ίαρχια ελέγχεται τις επικεφαλής πάση δουλειά, τότε για πολύ άρρητα δεν θα αφήσει χωρία άλλη ίαρχι να τρέξει. Είτε η ίαρχι που μεταφέρει ιδιοκτήτης δεν θα τρέξει με κίνδυνο να υπερχειλήσουν οι εντομευτές της πόρτας εισόδου.
3.4 Αξιολόγηση

Σε αυτό το τμήμα αξιολογούμε πρώτα το στατικό μέρος και κατόπιν το
dυναμικό μέρος της αρχιτεκτονικής λογισμικού που προτείνουμε.

3.4.1 Αξιολόγηση Στατικού Μέρους

Η αξιολόγηση του στατικού μέρους έγινε με την χρήση της έννοιας του όρου
χώρου (headroom) [30, 6]. Δηλαδή του μέγιστου πλήθους των κύκλων
eπεξεργασίας που μπορούν να αφηρωθούν καθαρά για έλεγχο επικεφαλίδων,
χωρίς να χάσουμε κανένα πακέτο και έχουν ακολουθήσει συνεχή ρόλη πακέτων ελάχιστου
μεγέθους.

Ετσι, βάλαμε κίνηση μόνο ελάχιστου μεγέθους πακέτων να ερχεται από μία
πορτά των 100 Mbit/s. Με την χρήση της μεθόδου μΜηχανής, κάθε πακέτο
erχόταν τοπολά σε μία από τις μΜηχανές. Η επεξεργασία στο πακέτο ξεκινούσε
με το να αναγνωρίζουμε τις τιμές κάσπων πεδίων. Κατόπιν ενός βρόγχους έληγχυ
αυτές τιμές για πολλές φορές. Ο βρόγχος έπερνε σταθερά πλήθος κύκλων
σε κάθε του εκτέλεσης. Μετρήσαμε το πλήθος των βρόγχων που μπορούσαμε να
εκτελέσουμε χωρίς να χάσουμε πακέτο. Μάλιστα, αλλάζει τον αριθμό των
διακλάσεων μΜηχανών έτσι ώστε να δώμε πως ο οριζόντιος χώρος κλιμακώνει.
Τα αποτελέσματα φαίνονται στην Εικόνα 3.5 η οποία μας δείχνει για
dιαφορετικό πλήθος μΜηχανών, ποιος είναι ο μέγιστος αριθμός κύκλων του
οριζόντιου χώρου. Όταν χρησιμοποιούμε και τις έξι μΜηχανές, έχουμε περίπου
4920 κύκλους διακλάσεως για κάθε πακέτο των 64 bytes.

Το αποτέλεσμα είναι συμβατό με παρόμοια εκτίμηση που έγινε στο [30] και
στην οποία οι συγγραφείς είχαν παρόμοιο περιβάλλον πειράματος (μικρά πακέτα, 100 Mbit/s συνάθεσμος).

3.4.2 Αξιολόγηση Δυναμικού Μέρους

Η απόδοση του δυναμικού μέρους εξαρτάται κατά κύριο λόγο από την δομή
dένδρου. Σε αυτό το τμήμα αξιολογούμε την χρησιμότητα της δενδρικής δομής
tόσο σε χώρο όσο και σε απόδοση.

3.4.3 Αξιολόγηση Χώρου

Η εξευκονόμηση χώρου είναι ιδιαίτερα σημαντική μιας και όλο το σύνολο
tων κανόνων θα πρέπει να χωρέσει στην μνήμη εντολών η οποία είναι πολύ
μικρή. Σε αυτό το τμήμα μελετούμε πόσο κερδίζουμε σε χώρο με την χρήση
tης δενδρικής δομής.

Για να έχουμε ένα μέτρο σύγχρονης, αλλάζουμε το εργαλείο S2Iωστε να μπο-
ΣΧΗΜΑ 3.5: Οφέλιμος χώρος σε κύκλους μεθόπων (232 MHz). Για την μέτρηση χρησιμοποιήσαμε έναν συνόλο των 100 Mbit/s και παρέχαμε ελάχιστο μεγέθους.

ΠΙΝΑΚΑΣ 3.1: Εξουσιοδότηση χώρου με την δενδρική δομή

<table>
<thead>
<tr>
<th>Ονομα</th>
<th>Πλήθος</th>
<th>Υπογραφών</th>
<th>εντολές Γραμμοκ. Χ.</th>
<th>εντολές Χ. Δέντρου</th>
<th>Μέτωση</th>
</tr>
</thead>
<tbody>
<tr>
<td>icmp-info</td>
<td>79</td>
<td>δεν χώρεσε</td>
<td>479</td>
<td>&gt;69.00%</td>
<td></td>
</tr>
<tr>
<td>backdoor</td>
<td>44</td>
<td>1531</td>
<td>886</td>
<td>42.13%</td>
<td></td>
</tr>
<tr>
<td>web-misc</td>
<td>18</td>
<td>401</td>
<td>277</td>
<td>30.92%</td>
<td></td>
</tr>
<tr>
<td>virus</td>
<td>6</td>
<td>173</td>
<td>149</td>
<td>13.87%</td>
<td></td>
</tr>
<tr>
<td>web-cgi</td>
<td>4</td>
<td>145</td>
<td>120</td>
<td>17.24%</td>
<td></td>
</tr>
</tbody>
</table>

ρεί να παράγει κώδικα χωρίς την χρήση δενδρικής δομής. Ετσι κάθε υπογραφή μεταφράζεται στο δικό της κομμάτι κώδικα που περιλαμβάνει όλους τους ελέγχους. Ένα παράδειγμα τέτοιου κώδικα χαρίζεται στην Εικόνα 3.6. Ο κώδικας που βασίζεται στο δέντρο παρουσιάζεται νωρίτερα στην Εικόνα 3.4.

Χρησιμοποιώντας διάφορα αρχεία με υπογραφές για είσοδο, μετρήσαμε το πλήθος εντολών που συνιστούν τους ελέγχους. Ο Πίνακας 3.1 συγκεντρώνει τα αποτελέσματα που πήραμε.

Το εργαλείο S2I συμπεριέχει όλα τα αρχεία με βαθμό συμπίεσης μεταξύ 17.24% και 69%. Μάλιστα, ο βαθμός συμπίεσης αυξάνεται όσο το αρχείο εισόδου μεγαλώνει κάτι που είναι ιδιαίτερα θεμελιώδες. Για την ακραία περίπτωση του αρχείου icmp-info, το εργαλείο S2I καταφέρνει να χωρέψει το αρχείο στην μνήμη της μηχανής, ενώ στην απλή περίπτωση δεν χωρούσε.
\\texttt{(.....................................................)}
\begin{verbatim}
if (ETHPROTOCOL==0x0800 && PROTOCOL==0x6) {
  if (PORT2==0x50) {
    if (IP2==0xa000001) {
      if (IP1==0xa000000) {
        if (DSIZE>0x200) {
          /* Action for "tcp 10.0.0.0 any -> 10.0...."*/
        }}
    }
  }

  //alert tcp [10.0.0.1 10.0.0.2] any -> 10.0.0.2 80 (ack: >512;)
  if (ETHPROTOCOL==0x0800 && PROTOCOL==0x6) {
    if (PORT2==0x50) {
      if (IP2==0xa000002) {
        if (IP1==0xa000001 || IP1==0xa000002) {
          if (ACK>0x200) {
            /* Action for "tcp [10.0.0.1 10.0.0.2] ..."*/
          }
        }
      }
    }

  }

}
(.....................................................)
\end{verbatim}

\\textbf{ΣΧΗΜΑ 3.6:} Γραμμικοί χώδεις που παράγεται από το εργαλείο S21 χωρίς την χρήση της δενδρικής δομής.

\\textbf{3.4.4 Αξιολόγηση Χρόνου Εκτέλεσης}

Εκτός από την εξουσιονόμηση χώρου, η χρήση της δενδρικής δομής θα πρέπει επίσης να βελτιώνει την ταχύτητα ελέγχου των υπογραφών. Αυτό είναι αναμενόμενο, μιας και το να διαπρέπει καινείς το δέντρο είναι πολύ πιο αποδοτικό από το να ελέγχει μία μία τις υπογραφές. Για να δούμε το μέγεθος της βελτίωσης κάναμε τα παρακάτω πειράματα στον προσωπικότητα του IXP 1200.

\\textbf{Με χρήση τεχνικών υπογραφών και τεχνιτής κίνησης.} Χρησιμοποιήσαμε πέντε υπογραφές και τις μεταφράσαμε και με την χρήση δενδρικού και χωρίς κυτή. Κατάπιε κατασκευάσαμε κίνηση με πακέτα που ταιριάζει σειριακά σε κάθε μία από τις υπογραφές. Η πέμπτη υπογραφή ταιριάζει με κάθε πακέτο.

Μετρήσαμε τον αριθμό χώδεων που διαπερνώνται σε έλεγχο πέδιων για τις δύο περιπτώσεις. Στον Πίνακα 3.2 παρουσιάζουμε αυτα που βρήκαμε. Για κάθε περίπτωση (πχ το πακέτο ταιριάζει με την υπογραφή 1, το πακέτο ταιριάζει με την υπογραφή 2, ...) γράφουμε τον αριθμό των χώδεων που χρειάστηκαν για τους ελέγχους.

Κατα μέσο όρο έχουμε μία μείωση των χώδεων κατά 21.2%. Είναι ενδιαφέρον το ότι όταν έχουμε λιγότερα ταιριάσματα (οπως στην περίπτωση της υπογραφής τέσσερα μόνον της), έχουμε περισσότερο κέρδος. Αυτό οφέλεται στο ότι ο Γραμμικός χώδεις θα πρέπει να περάσει αναγκαστικά από όλες τις υπογραφές ενώ ο δενδρικος θα ητάνει πιο γρήγορα στην τελική υπογραφή μέσω των κλαδιών του δέντρου.
3.4. ΑΞΙΟΛΟΓΗΣΗ

ΠΙΝΑΚΑΣ 3.2: Κύκλοι (232 MHz) για τον έλεγχο των πεδίων

<table>
<thead>
<tr>
<th>Περίπτωση</th>
<th>Γραμμικός χώδικας</th>
<th>χώδικας Δένδρου</th>
<th>Μέτρηση</th>
</tr>
</thead>
<tbody>
<tr>
<td>υπογραφή0+υπογραφή4</td>
<td>75</td>
<td>60</td>
<td>20.00%</td>
</tr>
<tr>
<td>υπογραφή1+υπογραφή4</td>
<td>74</td>
<td>62</td>
<td>16.22%</td>
</tr>
<tr>
<td>υπογραφή2+υπογραφή4</td>
<td>74</td>
<td>59</td>
<td>20.27%</td>
</tr>
<tr>
<td>υπογραφή3+υπογραφή4</td>
<td>74</td>
<td>61</td>
<td>17.57%</td>
</tr>
<tr>
<td>υπογραφή4</td>
<td>47</td>
<td>29</td>
<td>38.30%</td>
</tr>
<tr>
<td>Μέσος Ορος</td>
<td>68.8</td>
<td>54.2</td>
<td>21.22%</td>
</tr>
</tbody>
</table>

Με χρήση τεχνητών υπογραφών και πραγματικής κίνησης. Το πελάτημα αυτό το κάνωμε περισσότερο για να σχηματιστέμενα περισσότερο για την ορόσταση των προγραμμάτων αποτελεσμάτων. Χρησιμοποιήσαμε επίσης ένα μικρό σε αυτά την περίπτωση υπογραφές που μετρούσαν πετάει με βάση το προτόκολλο το υπολογιστή της, του υπολογιστή της εφαρμογής και το μέγεθος του πεδίου. Αντίθετα ομοίως από πριν, χρησιμοποιήσαμε αληθινή κίνηση. Η κίνηση προερχόταν από ένα εξειδικευτήρα του FORTH [14], κατά την διάρκεια μιας εργασίας μέρας. Και πάλι μετρήσαμε ένα 20% σε μέση μείωση του χρόνου που απαιτείται για τον έλεγχο πεδίων.

Χρήση πραγματικών υπογραφών και πραγματικής κίνησης. Τέλος χρησιμοποιήσαμε την προγραμμένη κίνηση με το σύνολο υπογραφών του Snort 'backdoor'. Τα αποτελέσματα έδειξαν ότι χωρίς την κίνηση δεν ήταν δομής, ο έλεγχος πεδίων των 44 υπογραφών έπεφτε περίπου 280 κύκλους. Όταν χωρίς συμπεριέλθε την κίνηση δένδρου, τότε οι κύκλοι πέρτον στους 180, μία μείωση της τάξης των 35%.

Συνοψίζοντας, η κίνηση δένδρου αποδεικνύεται αποδοτική πάντα σε χώρο, όπως και σε χρόνο. Όσο αφορά τον χώρο, παρατηρήσαμε μία εξοικονόμηση χώρου τουλάχιστον 17.3% στην μήνυμα εντολών. Όσο αφορά τον χρόνο, με κατάστα ακόμα synthesis είδαμε μείωση περίπου 20% στον χρόνο που χρειάζεται για την εφαρμογή των ελέγχων των υπογραφών.

3.4.5 Ενα ελαφρό Snort για συστήματα και86

Όπως προαναφέραμε, ο χώδικας που παράγει το εργαλείο S2I, μπορεί να χρησιμοποιηθεί και σε άλλους επεξεργαστές. Κανείς λογιστικός κύκλοι περάματα με έναν Intel Pentium επεξεργαστή. Συγχρίσαμε τον χρόνο χρήση (user time) της εκτέλεσης του πραγματικού Snort, και της εκτέλεσης του πραγματικού Snort, και της εκτέλεσης του πραγματικού Snort, και της εκτέλεσης του πραγματικού Snort, και της εκτέλεσης του πραγματικού Snort.
3.5 Ανακεφαλαίωση και Μελλοντική Εργασία

Η κωδικοποίηση με το χέρι δεκάδων υπογραφών στο micro-C ή assembly είναι αρκετά δύσκολη δουλειά και γίνεται ασόμα δυσκολότερη αν δεν θέλουμε λάθη. Σε αυτό το κεφάλαιο παρουσιάζουμε μία αρχιτεκτονική λογισμικού που αυτοματοποιεί την παραγωγή κώδικα από περιγραφές υπογραφών γραμμένες σε υψηλή γλώσσα. Με την χρήση του εργαλείου S2I ο IXP 1200 μπορεί να χρησιμοποιηθεί σε περιβάλλοντα ανύψωσης επιθέσεων, χωρίς να είναι απαραίτητη η λεπτομερής γνώση της αρχιτεκτονικής του και της γλώσσας micro-C. Επί το εργαλείο S2I ανύξατε την ευελιξία και την ταχύτητα ανάπτυξης κώδικα χωρίς να επιβαρύνετε την απόδοση και χωρίς σπατάλη στους πόρους που χρησιμοποιεί.

Υπάρχουν πολλές κατευθύνσεις που μπορούν να ακολουθηθούν για την βελτίωση της αρχιτεκτονικής. Για παράδειγμα, η δενδρική δομή μπορεί να συμπιεστεί ακόμα περισσότερο με το να επιλέγουμε πιο πεδίο θα αντιστοιχεί σε κάθε επίπεδο, για κάθε υποδενδρό (πχ αναδρομικά). Επίσης η σειρά με την οποία ελέγχονται τα πεδία θα μπορούσε να βασιστεί σε μελέτη των συναρμολογημένων πακέτων που περιέχονται στην κίνηση ώστε να μειωθεί ο χρόνος ελέγχων. Επίσης, θα μπορούσαμε να δούμε τι μπορούμε να πετύχουμε με πόρτες του 1 Gbit/s. Τέλος, ενδεικτεί ότι παρουσιάζει να εφαρμόζουμε παρόμοιες τεχνικές για ψάξιμο στο περιεχόμενο του πακέτου.

Κεφάλαιο 4

Ο Ενεργός Διαμοιραστής για ΣΑΕ

Ενας τρόπος για να ανιχνεύουμε εισβολές σε δίκτυα πολύ υψηλών ταχυτήτων είναι να χρησιμοποιήσουμε πολλαπλά Συστήματα Ανίχνευσης Εισβολέων μαζί με έναν διαμοιραστή (Splitter) που θα μοιράζει το φορτίο ομοιόμορφα. Σε αυτό το κεφάλαιο εξετάζουμε την αρχιτεκτονική ενός τέτοιου διαμοιραστή, ο οποίος χρησιμοποιεί δύο μεθόδους για να βελτιώσει την απόδοση του συστήματος. Η πρώτη μέθοδος, είναι το φιλτράρισμα σε χαμηλό επίπεδο κατά το οποίο ένα μέρος των πακέτων επεξεργάζεται ο διαμοιραστής και όχι τα Συστήματα Ανίχνευσης Εισβολέων. Η δεύτερη μέθοδος είναι η χρήση ενταμπυτικών τοπικότητας, κατά την οποία, ο διαμοιραστής αναδιατάσσει τα πακέτα έτσι ώστε να βελτιώσει την τοποκύτητα αναφορών σε κάθε έναν από τους ανιχνευτές (sensors). Βασίζοντας στους πειραμάτους που κάναμε, το φιλτράρισμα σε χαμηλό επίπεδο μείωνε το πλήθος των πακέτων που πρέπει να επεξεργαστούμε κατά 32% και έτσι αυξάνει την απόδοση των ανιχνευτών κατά 8%. Από την άλλη, η χρήση των ενταμπυτικών τοπικότητας αυξάνει την απόδοση των ανιχνευτών κατά 10%. Αν συνδιαστούν μαζί, οι δύο μέθοδοι πετυχάνουν συνολική αύξηση της απόδοσης κατά περίπου 20%.

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο παρουσιάζουμε δύο μηχανισμούς που μπορούν να υλοποιηθούν από έναν διαμοιραστή φόρτου, ώστε να βελτιώσει την απόδοση του συνολικού συστήματος ανίχνευσης εισβολέων. Ο πρώτος μηχανισμός βασίζεται στην παρατήρηση ότι ένα σημαντικό μέρος των πακέτων χρειάζεται μόνο επεξεργασία επεκτάσεων. Δεδομένου ότι η επεξεργασία επεκτάσεων είναι σχετικά εύκολη και μπορεί να υλοποιηθεί από το υλικό, μπορούμε να την εν-
σωματώσουμε στην δουλεία που κάνει ο διαμορφωτής. Το πλεονέκτημα του φιλτραρήματος σε χαμηλό επίπεδο (early filtering) είναι ότι ένα σημαντικό ποσοστό της χίνησης δεν χρειάζεται να μεταδοθεί καθόλου στους ανιχνευτές και έτσι ο φόρτος τους μπορεί να μειωθεί.

Ο δεύτερος μηχανισμός βασίζεται στην παρατήρηση ότι διαμορφωτικό τύπου πακέτα θα ενεργοποιηθούν διαμορφωτικά σύνολα κανόνων των SAE και έτσι θα επιβράγουν το σύστημα μνήμης του ανιχνευτή (θα μειώσουν την τοποκίνηση των αναφορών μνήμης). Έτσι, η μέθοδος των ερευνητών τοποκίνησης φυσικά ώστε πακέτα του ίδιου τύπου να ομαδοποιούνται από τον διαμορφωτή, πριν αποσταλλούν στον ανιχνευτή για έλεγχο. Η μέθοδος αυτή βελτιώνει την απόδοση των ανιχνευτών χωρις να ωλιάζει την σημασιολογία της χίνησης και επίσης χωρις να απαιτεί αλλαγές στους ανιχνευτές.

Για τα πειράματα που κάναμε χρησιμοποιήσαμε το snort ως SAE και πακέτα από προγραμματισμένη σχίνηση προκειμένου να μελετήσουμε με όσο το δυνατό πιο ακριβώς το πρόβλημα προσνέμει μεθόδους.

Το υπόλοιπο του κεφαλαίου αποτελείται από τα παρακάτω τμήματα. Στο Τμήμα 4.2 παρουσιάζουμε κάποιες γενικές πληροφορίες για συστήματα διαμορφωτικού φόρτου. Στο Τμήμα 4.3 παρουσιάζουμε την αρχιτεκτονική του Ενεργού Διαμορφωτή ο οποίος υλοποιεί τις δύο μεθόδους που προανέφεραμε. Τέλος, στο Τμήμα 4.4 παρουσιάζουμε αναλυτικά τα πειραματικά αποτελέσματα που πήραμε και κλείνουμε το κεφάλαιο με το Τμήμα 4.5 στο οποίο ανακεφαλαίονουμε.

### 4.2 Σχετικές Εργασίες

**Η εισαγωγή φόρτου**. Η εισαγωγή φόρτου είναι μία μέθοδος που έχει μελετηθεί και εφαρμοστεί εκτενώς στην κατασκευή χλιμακώμενων υπηρεσιών για το διαδίκτυο (scalable internet services) όπως είναι οι εξυπηρετήτες του παγκόσμιου ιστού (Web Servers) [15]. Στο [15] η εισαγωγή φόρτου βασίζεται σε πληροφορίες που στέλνει ο χάδες εξυπηρετητής. Με τον τρόπο αυτό ο διαμορφωτικός φόρτου είναι σχεδόν τέλειος, δηλαδή όλοι οι εξυπηρετήτες έχουν σχεδόν τον ίδιο φόρτο. Από την άλλη άποψη, ο διαμορφωτικός φόρτου το πρέπει να υπαρχεί σε πολλές εξυπηρετήτες έχει αναλυθεί μία συγκεκριμένη ροή από πακέτα (flow). Αυτή η μνήμη που πρέπει να διατηρεί ο διαμορφωτής χάνει την υλοποίηση του δισκοπλαστήρα ιδιαίτερα σε πολύ υψηλές ταχύτητες. Αλλά και οι διευκολύνεις λεπτομέρειες ενός διαμορφωτή φόρτου έχουν σημασία στην μελέτη των μεθόδων που προτείνουμε, σε αυτό το κεφάλαιο θα υποθέσουμε ότι μικρές ανισομεμβέρες φόρτου είναι ανεπτύκτες και έτσι για τον διαμορφωτή του φόρτου θα χρησιμοποιήσουμε μία πιο απλή μέθοδο που βασίζεται σε συναρτήσεις hashing.

Η χρήση συμπεριφερίων hashing για διαμορφωτικό φόρτου ώστε χάδες πακέτα που ανήκουν στην ίδια ροή να πηγαίνουν πάντα στον ίδιο εξυπηρετητή, είναι
κατά το συνηθισμένο [26, 10, 7]. Στην περίπτωση των κρυφών μνημών για τον ιστό (Web caches) με την χρήση συναρτήσεων hashing μπορούμε αποδοτικά να βρούμε τον τελικό εξυπηρέτητη [26]. Μάλιστα η συνάρτηση μπορεί συμ-
περιλάβει πλαγιές αλλαγές στο σύνολο των διαδικασιών εξυπηρέτησης, δηλαδή
αφιερώσεις (κάτω από εξυπηρέτητης που χάλασε) ή προσθέσεις (καινούριοι εξυ-
πηρέτητης). Στο [10], οι συγγραφείς χρησιμοποιούν μια υβριδική μέθοδο που συναρ-
τήσεων hashing και διατήρησης μνήμης. Χρησιμοποιούν hashing προεξε-
μένου να κατατάξουν μία ροή σε ένα σύνολο ροών. Κατόπιν, για κάθε σύνολο
κρατούν μνήμη προεξεμένου να διηρύνεται προς τα πού πρέπει να προωθηθεί μια
συγκεκριμένη ροή. Μια εκτενής μελέτη των συναρτήσεων hashing παρουσιά-
ζεται στο [7]. Άλλες εργασίες πάνω σε διαμορφασμό φόρτου είναι οι [27] και
[21] στις οποίες οι συγγραφείς διερευνούν και υλοποιούν διαμορφασμό φόρτου
με Επεξεργαστές Δικτών.

Διαμορφασμός φόρτου σε ΣΑΕ. Αν και υπάρχουν εμπορικά προϊόντα για
tον διαμορφασμό φόρτου σε ΣΑΕ [34], οι διαδέχονται πληροφορίες σχετικά με
tις πολιτικές που υλοποιούν και την εσωτερική τους αρχιτεκτονική είναι ελαχι-
στες. Από την άλλη, στον ερευνητικό τομέα υπάρχει η εργασία [22] η οποία
μελετά μια γενικότερη υποδομή για υψηλής ταχύτητας ανάγνωση εισβολέων.
Οι συγγραφείς προτείνουν μια αρχιτεκτονική δύο επιπέδων. Το πρώτο επίπεδο
αναλαμβάνει τον εξ ίσου διαμορφασμό του φόρτου ανάμεσα στα διαδέχομαι ΣΑΕ.
Το δεύτερο επίπεδο επιλέγει ένα σετ από ΣΑΕ για την τελική επεξεργασία του
πολέμου. Η απόφαση που πρέπει να αποφασίσει ένα πολέμο βασίζεται σε κανόνες
που περιγράφουν το πλάσιο των επιθέσεων στο οποίο πλανούν να ανέχει ένα
πολέμο. Ωστόσο, η εργασία επεκτείνεται στο να διατηρήσει την σημασιολογία
tης ανάγνωσης επιθέσεων σε ένα γενικότερο μοντέλο ανάγνωσης επιθέσεων.
Βασίζεται δηλαδή στο ότι υπάρχουν διαφορετικές μέθοδοι ανάγνωσης όπως
στατιστικές, ανάγνωσης ακαμαλείων και ανάγνωσης περιεχομένου. Αντίθετα,
η εργασία που παρουσιάζουμε εμπειρεστήρια στην απόδοση ενός παράλληλου ΣΑΕ
και πιο συγκεκριμένα, ενός ΣΑΕ που βασίζεται σε ανάγνωση
περιεχομένου.

Φιλτράρισμα σε χαμηλό επίπεδο και διαμορφασμό φόρτου. Η ιδέα του
φιλτραρισμού σε έναν διάμορφο φόρτο συζητείται επίσης στο [15]. Εκεί,
ο διαμορφασμός φόρτου επελεγεί διαμορφώθηκε έτσι ώστε να μπλοκάρει την κίνηση
που πηγαίνει σε αδημοσίευτες πόρτες. Ο σκοπός ήμες που το κάνει είναι για
προστασία και όχι για βελτίωση απόδοσης.

Τεχνικές βελτίωσης τοποκότητας. Οι τεχνικές βελτίωσης της τοποκότητας
έχουν, επίσης, μελετηθεί. Για παράδειγμα, στο [23] οι συγγραφείς εφαρμόζουν
μια τεχνική που βελτιώνει την τοποκότητα των απήχεων σε χρωμές μνήμες για
τον ιστό (Web Caches). Η εφαρμογή της τεχνικής παρουσιάζει πολύ σημαντική
βελτίωση στην απόδοση του συστήματος διαχείρισης αρχείων. Παράλληλα, η
παρούσα εργασία είναι η πρώτη που μελετά την επίπεδη τοποκότητας μέσω
κατάλληλου διαμορφασμού φόρτου και μάλιστα στο πλάσιο των ΣΑΕ.
4.3 Σχεδίαση

Τέσσερεις είναι οι βασικοί στοιχείοι στον σχεδιασμό ενός διαμορφωτή χίνης για το SAE. Πρώτον το πακέτο που ανήκουν στο ίδιο πλαίσιο επίθεσης θα πρέπει να τα επεξεργαστούν οίδος ανιχνευτές. Σε κάθε άλλη περίπτωση θα υπάρχει χίνης να μην ανιχνευθούν κάποιες επίθεσεις. Για επίθεσεις που στηρίζονται στο περιεχόμενο των πακετών, χρειάζομαι όλα τα πακέτα μίας ροής να κατευθυνθούν στον ίδιο ανιχνευτή. Δεύτερον, η χίνης θα πρέπει να κατανεμηθεί έτσι ώστε η συνολική απόδοση του SAE να μειωθεί. Αν υποθέσουμε N ανιχνευτές, ένας καθός τρόπος για να το επιτύχουμε είναι να στέλνουμε περίπου 1/N της συνολικής χίνης σε κάθε ανιχνευτή. Κάτι τέτοιο το πετυχαίνουμε με το να διαμοιράζουμε της ροές σε αυτό που θα συνηθίσουμε παρακάτω είναι πως το φίλτρασμα σε χαμηλό επίπεδο και οι ενταμειτές τοποκάττας μπορούν να βελτιώσουν περιστότερα την απόδοση. Τρίτων, ο αλγόριθμος διαμορφωμού του φορτίου θα πρέπει να είναι πολύ αποδοτικός ώστε να λειτουργεί σε δίκτυο υψηλής ταχύτητας. Τέταρτων, το σύστημα (ιδιαίτερα) δεν θα απαιτεί άλλο γράμμα στους ανιχνευτές.

Το σύστημα αποτελείται από τη μονάδα φίλτρασματος σε χαμηλό επίπεδο, τη μονάδα διαμορφωμού του φόρτου και ένα σύνολο απο ενταμειτές τοποκάττας για κάθε ανιχνευτή. Σε αυτό το τμήμα θα παρουσιάζουμε χάθε μονάδα με περισσότερες λεπτομέρειες.

4.3.1 Φιλτράσμα σε Χαμηλό Επίπεδο

Το φίλτρασμα σε χαμηλό επίπεδο, σημαίνει υπερπαντρικά την υλοποίηση λειτουργικότητας του ανιχνευτή στον διαμορφωτή. Δεδομένου ότι κάποιο πακέτο δεν είναι είναι πολύ ριζώδες από τον έλεγχο περιεχομένων παρατηρεί την υλοποίηση απαντήσεις πάνω στον διαμορφωτή. Με αυτό τον τρόπο μειώνομε τον ύποπτο κάθε ανιχνευτή και χρησιμοποιούμε το συστήμα απο πολλά στον διαμορφωτή. Με αυτό τον τρόπο μειώνουμε τον ύποπτο κάθε ανιχνευτή και χρησιμοποιούμε το συστήμα απο πολλά στον διαμορφωτή.

Για να υλοποιήσουμε το φίλτρασμα σε χαμηλό επίπεδο πήραμε από τα σετ...
4.3. ΣΧΕΔΙΑΣΗ

υπογραφάς των ΣΑΕ όλες τις υπογραφές που δεν απαντούσαν ψάχιμο στα περιεχόμενα των πακέτων. Για το σοβατές οι υπογραφές αποτελούν ένα μικρό υποσύνολο: 165 υπογραφές στις 1700 του κανονικού συνόλου υπογραφών. Περιμένουμε ότι η επεξεργασία των 165 υπογραφών που αφορούν μόνο ανάλυση επικεφαλίδων θα γίνεται αρκετά εύκολα από το υλικό.

Όπως η λειτουργία του διαμοιρασμού έχει εισάγει εξής: Κάθε πακέτο που έρχεται περνάει πρώτα από το σύνολο των 165 υπογραφών. Αν και μία υπογραφή δεν ταμιάζει και το πακέτο δεν έχει περιεχόμενο τότε δεν προσαρμόζεται για περαιτέρω ανάλυση. Αν το πακέτο περιεχόμενο τότε πρέπει να το ελέγξουμε από το προαπαιτούμενο στους ανιχνευτές. Αν το πακέτο δεν έχει περιεχόμενο, άλλα ταμιάζει με κάποια από τις 165 υπογραφές τότε πάλι το υποσύνολο στους ανιχνευτές για να προκαλέσει την κατάληψη αντίδραση.

4.3.2 Διαμοιρασμός Φόρτου

Ενας απλός και αποδοτικός τρόπος για τον διαμοιρασμό του φόρτου είναι η χρήση μιας συνάρτησης hash πάνω στις επικεφαλίδες του πακέτου. Μια τέτοια συνάρτηση, όπως παραδείγματα είναι το CRC16 μπορεί να μοιράσει ομοιόμορφα το φορτίο έτσι ώστε κάθε ανιχνευτής να πάρει περίπου τον ίδιο όγκο δοσείτας. Αν μάλιστα προσέχουμε τι θα δώσουμε ως είσοδο την συνάρτηση μας, τότε μπορούμε να έχουμε διαμοιρασμό φόρτου που να διατηρεί την ανάληψη ροών. Δηλαδή όλα τα πακέτα της ίδιας ροής να πηγάζουν πάντα στον ίδιο ανιχνευτή. Για να το πετύχουμε αυτό αρχίζουμε με το δίκτυο (Source and destination IP and port). Αν μάλιστα έχουμε κίνηση με καλή συμπεριφορά (π.χ. TCP-friendly), τότε αυτός τρόπος θα ισοκρατούνει το φορτίο ως ομοιόμορφα σε αυξομοιώσεις της κίνησης δικτύου. Φυσικά βέβαια ο τρόπος αυτός δεν αντιμετωπίζει το πρόβλημα μιας επίθεσης που επεκείνεται όταν παραγωρώνει το σύστημα προστασίας να χρυσίει μία επίθεση. Όμως μια τέτοια μελέτη δεν έχει να κάνει με τους μηχανισμούς που προτείνουμε.

Στα πλαίσια της δικής μας σχεδίασης χρησιμοποιήσαμε μία κατάλληλη συνάρτηση που βασίζεται στο CRC-16, μιας και αποδεικνύει καλά [7], δεν χρειάζεται μνήμη και υπάρχουν υλοποιήσεις της στο υλικό.

4.3.3 Ενταμειωτές Τοπικότητας

Οι ενταμειωτές τοπικότητας είναι μια μέθοδος για την μορφοποίηση της κίνησης πακέτων ώστε τέλος να βελτιώνει η απόδοση του ανιχνευτή που θα τα επεξεργαστεί. Η ιδέα βασίζεται στο γεγονός ότι υπάρχουν συγκεκριμένα σύνολα υπογραφών για συγκεκριμένον τύπο πακέτου. Όπως, όταν ένας ανιχνευτής θα πρέπει να ελέγξει ένα πακέτο ενός τύπου θα χρειαστεί το αντίστοιχο CRC16,
ΣΧΗΜΑ 4.2: Ομαδοποίηση πακέτων με τους ενταμευτές τοπικότητας

σετ εργασίας (working set, σύνολο από δεδομένα και εντολές). Η εναλλαγή μεταξύ διαφορετικών σετ εργασίας δημιουργεί πολλαπλές αστοχίες στην κρυφή μνήμη (cache misses). Αναπόφευκτα η απόδοση πέφτει.

Για να αυξηθεί η τοπικότητα των αναφορών, ο διαμορφωτικός αναλύει τα εισερχόμενα πακέτα και τα βάζει σε ξεχωριστούς ενταμευτές. Προστατεύει, σε κάθε ενταμευτή να υπάρχουν πακέτα από ένα είδος μόνο. Όταν ένας ενταμευτής χορηγεί, όλα τα πακέτα που περιέχουν αποστέλλονται ως μία συνεχίσματα στον τελικό ανηχυτή για ανάλυση. Ετσι, αυξάνονται τα πακέτα του ίδιου τύπου που θα ράφησαν στον ανηχυτή το ένα πίσω από το άλλο. Όποτε η τοπικότητα των αναφορών θα βελτιωθεί και έτσι και η απόδοση θα αυξηθεί.

Μειώνεται και η πολύτλοκης των πακέτων με βάση τα σύνολα υπογραφών των ΣΔΕ είναι αρχετά πολύτλοκης. Ετσι, βασιστικά στις παρακάτω ευριστικές μεθόδους για να υπολογίζουμε τον τελικό ενταμευτή για ένα πακέτο:

ΑΤ Χρήση συνάρτησης hash και στην αρχική και στην τελική πόρτα ενός πακέτου. Με αυτό τον τρόπο διαμορφώνουμε την κίνηση εξ ίσους σε όλους τους ενταμευτές. Επεξεργαστική κίνηση στέκεται σε περισσότερα καθήκοντα, και δεδομένου ότι τα πακέτα με την ίδια αρχική και τελική πόρτα θα πηγαίνουν πάντα στον ίδιο ενταμευτή, περιμένουμε περισσότερα πακέτα ίδιου τύπου να βρεθούν το ένα πίσω από το άλλο.

Τ Χρήση συνάρτησης hast μόνο στην τελική πόρτα. Δεδομένου ότι παίρνουμε μόνο την τελική πόρτα υπόψη, περιμένουμε ότι θα αυξήθει ακόμα περισσότερο την πιθανότητα να έχουμε πακέτα ίδιου τύπου το ένα πίσω από το άλλο.

π* Δέσμευση ενός αριθμού από ενταμευτές για ένα συγκεκριμένο πρωτόκολλο. Με αυτό τον τρόπο, οι ενταμευτές τοπικότητας χωρίζονται σε ομάδες ανάλογως με το πρωτόκολλο. Παράδειγμα, 16 από τους 16 ενταμευτές χρησιμοποιούνται μόνο για πακέτα του πρωτοκόλλου TCP και οι υπόλοιποι για όλα τα άλλα. Για κάθε ομάδα ενταμευτών χρησιμοποιούμε είτε την μέθοδο ΑΤ ή την μέθοδο Τ και συνεπάγεται και το πλήρες όνομα της μεθόδου.
4.4. ΠΕΙΡΑΜΑΤΑ

<table>
<thead>
<tr>
<th>Μέθοδος</th>
<th>Περιγραφή</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT</td>
<td>hash(Αρχική + Τελική πόρτα)</td>
</tr>
<tr>
<td>T</td>
<td>hash(Τελική πόρτα)</td>
</tr>
<tr>
<td>π_AT</td>
<td>διαφάνεια ET κατά πρωτόκολλο</td>
</tr>
<tr>
<td></td>
<td>Χρήση μεθόδου AT</td>
</tr>
<tr>
<td>π_T</td>
<td>διαφάνεια ET κατά πρωτόκολλο</td>
</tr>
<tr>
<td></td>
<td>Χρήση μεθόδου T</td>
</tr>
<tr>
<td>Σ_T</td>
<td>αφύρωση ET σε Συγκεκριμένη χίνηση + μέθοδος T</td>
</tr>
<tr>
<td>π_Σ_T</td>
<td>διαφάνεια ET κατά πρωτόκολλο</td>
</tr>
<tr>
<td></td>
<td>Χρήση μεθόδου Σ_T</td>
</tr>
</tbody>
</table>

ΠΙΝΑΚΑΣ 4.1: Μέθοδοι για την χρήση των Ενταμευτών Τοπικότητας

Σ_T Δέσμευση ενός πλήθους ενταμευτών για συγκεκριμένες ροές χίνησης και χρήση των υπολοιπών με βάση την μέθοδο T. Για παράδειγμα, ένας ενταμευτής θα δέχεται μόνο χίνηση web ένας άλλος μόνο χίνηση nntp κτλ. Τα υπόλοιπα πακέτα πηγαίνουν στους υπόλοιπους ενταμευτές χρησιμοποιώντας την μέθοδο T.

π_Σ_T Δέσμευση μερικών ενταμευτών για συγκεκριμένο πρωτόκολλο και κατά πιν για συγκεκριμένο τύπο χίνησης. Για όλα τα υπόλοιπα πακέτα χρησιμοποιήσει την μέθοδο T. Με τον τρόπο αυτό οι ενταμευτές χωρίζονται αρχικά σε ομάδες με βάση το πρωτόκολλο. Κατόπιν, μέσα σε κάθε ομάδα χωρίζονται σε κάποιους που δέχονται μόνο συγκεκριμένο είδος χίνησης και σε κάποιους γενικούς. Οι γενικές χρήσεις ενταμευτές χρησιμοποιούν την μέθοδο T.

Τα αποτελέσματα έχουν οι παραπάνω μέθοδοι στην απόδοση του συστήματος αναλύεται στο Τμήμα 4.4.

4.4 Πειράματα

Τα πειράματα που παρουσιάζουμε παρακάτω μελετάνε την επίδραση του υλικοφόρου σε χαμηλό επίπεδο και των ενταμευτών τοπικότητας στην απόδοση του SAE.

Για τα περισσότερα πειράματα χρησιμοποιούμε έναν υπολογιστή με επεξεργαστή Pentium III στα 1.13 GHz με 8 KB L1 cache, 512 KB L2 cache και με 512 MB κεντρικός μνήμης. Το λειτουργικό σύστημα είναι Linux (kernel version 2.4.17, RedHat 7.2). Το SAE που χρησιμοποιούμε βασίζεται στην έκδοση 2.0-beta20 του snort την οποία μεταφράσαμε με τον gcc, version 2.96 (optimization flags 02).
Τα περισσότερα πειράματα διεξάγονται με το διάβασμα των πακέτων από το δίσκο, εκτός των πειραμάτων που αφορούν το φυλτράσιμο σε χαμηλό επίπεδο. Εκτός είδους αυτών παρακολουθούμε έργασα από το δίσκο ως τη μετρήσιμη της επίδρασης του φυλτράσιμου σε χαμηλό επίπεδο σε ένα σύστημα δικτύου του ΣΑΕ. Για το σκοπό αυτό χρησιμοποιούμε ένα απλό δίσκο με δύο υπολογιστές A και B και έναν ανιχνευτή. Ο υπολογιστής A διαβάζεται το αρχείο με τα πακέτα από το δίσκο και τα στέλνει στον B με το πρόγραμμα tcpreplay. Αυτό γίνεται πάνω από ένα 100 Mbit/s Ethernet switch το οποίο έχει ρυθμιστεί κατάκλημα ώστε να στέλνει την κίνηση και προς τον ανιχνευτή. Επιπλέον, ο ασφαλής χρονισμός των πακέτων δεν έχει ιδιαίτερη σημασία, αλλά στέλνουμε τα πακέτα με του μέγιστο ρυθμό, οπότε και η χρήση του δικτύου έργασε το 90%.

Το σύνολο πακέτων που χρησιμοποιούμε έρχεται από το αρχείο του NLANR και έχει κωδικό planar.MRA.1031627450. Δημιουργήθηκε τον Σεπτέμβριο του 2002 παρακολουθώντας έναν σύνδεσμο OC12e των 622 Mbit/s. Επιπλέον, από τα πακέτα είχε αφαιρεθεί το περιεχόμενο τους, προσθέτοντας ομοιόμορφα κατανεμημένα τεχνικά χαίνη σαν να δημιουργήσουμε πια ρεαλιστικά πακέτα. Η τεχνική αυτή έχει χρησιμοποιηθεί και στο παρελθόν προκειμένου να μετρηθεί η απόδοση ΣΑΕ και δίνει αρκετά ασφαλή αποτελέσματα [3].

### 4.4.1 Αξιολόγηση Φυλτράσιμου σε Χαμηλό Επίπεδο

Αναλύοντας το αρχείο πακέτων, παρατηρούμε ότι πάνω από το 40% των πακέτων δεν έχουν περιεχόμενο. Στην πλευρή τους είναι πακέτα TCP acknowledgments και τα περισσότερα δεν ταράζουν σε χαμηλό από τις 165 υπογραφές. Τα πακέτα αυτά μπορούν να καταλήξει από τον διαμορφωτή κατά την διάρκεια του φυλτράσιμου σε χαμηλό επίπεδο. Μόνο ένα πολύ μικρό ποσοστό αυτών των πακέτων (λιγότερο του 1% του συνόλου των πακέτων) ταράζει με κάποια από τις 165 υπογραφές και έτσι πρέπει να το προωθήσουμε στους ανιχνευτές.

Σε κάθε ανιχνευτή, η μέθοδος του φυλτράσιμου σε χαμηλό επίπεδο θα μειώσει τον όρο ανιχνευτής όλα τον όρο χρήσης του δικτύου. Δηλαδή την διαχείριση διαστατών (interrupt processing) και την μεταφορά πακέτων από την κάρτα δικτύου στον χώρο χρήσης για επεξεργασία.

Το μέτρημα της εργασίας επεξεργασίας για μια παράδειγμα με το χρόνο της μεταφοράς των πακέτων από τον διαμορφωτή στον χώρο χρήσης για επεξεργασία.

Το μετρημένο τετελεσμένο όριο επεξεργασίας, υπολογίζοντας το χρόνο της μεταφοράς των πακέτων από τον διαμορφωτή στον χώρο χρήσης για επεξεργασία, έχει το ποσοστό της μεταφοράς των πακέτων στον χώρο χρήσης για επεξεργασία, οπότε η απόδοση του φυλτράσιμου σε χαμηλό επίπεδο περιποιείται και δίνει αρκετά ασφαλή αποτελέσματα [3].

Για να μετρήσουμε την επίδραση του φυλτράσιμου σε χαμηλό επίπεδο στην απόδοση του ανιχνευτή μετρήσαμε τον χρόνο συστήματος (system time)
και τον χρόνο χρήστη (user time) για να τρέξουμε το snort με δύο ξεχωριστά σύνολα πακέτων. Το πρώτο είναι το αρχικό σύνολο πακέτων, ενώ το δεύτερο είναι το αρχικό χωρίς τα πακέτα που θα πετάζει το φιλτράρισμα σε χαμηλό επίπεδο. Τα αποτελέσματα φαίνονται στην Εικόνα 4.3. Παρατηρούμε ότι ο χρόνος χρήστη μειώθηκε κατά 6.6% (από 45.67 δευτ. σε 42.66 δευτ.) ενώ ο χρόνος συστήματος μειώθηκε κατά 16.8% (από 10.1 δευτ. σε 8.7 δευτ.). Αν πάρουμε και τον χρόνο χρήστη και τον χρόνο συστήματος μαζί, η συνολική βελτίωση είναι περίπου 8%.

4.4.2 Απόδοση της Συνάρτησης Hash

Η απόδοση της συνάρτησης hashing CRC16 έχει ήδη μελετηθεί στο [7]. Επειδή όμως έχουμε μία απλότητα δυσκαλοποιών, είναι απαραίτητο να επιβεβαιώσουμε ότι η διεπαντήμενη συνάρτηση είναι εξίσου δίκαιη, δηλαδή μοιράζεται τον φόρτο εξ ίσου σε όλους τους ανιχνευτές. Για το σκοπό αυτό μετρήσαμε τον μέγιστο αριθμό πακέτων που πήρε κάθε ανιχνευτής για τις περιπτώσεις των δύο, τεσσάρων και οκτώ ανιχνευτών. Κατόπιν, το συγκρίνουμε με το θεωρητικά δίκαιο ποσοστό (δηλαδή την διάρκεια του συνόλου των πακέτων με το πλήθος των ανιχνευτών). Χρησιμοποιήσαμε ένα μεγάλο αρχείο πακέτων διάρκειας περίπου 80 δευτερόλεπτων ενός συνδέσμου 622 Mbit/ε. Δεδομένου ότι το hashing προσωπικούν έναν ψεύδοστοχαστικό αλγόριθμο, χρειάζονται αρκετά πακέτα ώστε η μορφασά να είναι δίκαιη. Για μικρό αριθμό πακέτων έχουμε ανιχνευτή-φόρτο. Αυτό όμως δεν είναι πολύ σοβαρό μειονέκτημα μιας και το σύστημα μπορεί να απορρητήσει τέτοιες ανισοκατανομές αν υπάρχει επαρκής μνήμη.

Ο πίνακας 2, δείχνει την μέγιστη διαφορά από το θεωρητικά δίκαιο, για δια-
ΠΙΝΑΚΑΣ 4.2: Απόδοση της απλοποιημένης CRC16 hashing συνάρτησης.

<table>
<thead>
<tr>
<th>Ανιγνωστές</th>
<th>διαφορά επί τους εκατό του πιο βαρυφορμωμένου από το δίκαιο μερίδιο του</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.25%</td>
</tr>
<tr>
<td>4</td>
<td>5.70%</td>
</tr>
<tr>
<td>8</td>
<td>13.55%</td>
</tr>
</tbody>
</table>

η πιο βαρυφορμωμένος ανιγνωστής παίρνει 13.55% περισσότερα ποσά από το θεωρητικά δίκαιο μερίδιο του. Αυτό είναι αποδεκτό για μία ακήλη υλοποίηση της συνάρτησης hashing. Αναμένουμε ότι με περισσότερη κίνηση, αυτή η διαφορά μεταξύ δίκαιου και πραγματικού μεριδίου θα μειωθεί.

Αυτά τα αποτελέσματα είναι σημαντικά στην κατανόηση του πως αλληλεπιδρά η πολιτική διαμορφωμοί γόρτο με τους ενταμευτές τοποκόητας που παρουσιάζονται στο επόμενο τμήμα.

4.4.3 Επίδραση των ET στην Απόδοση του SAE

Προκειμένου να δούμε το μέγεθος της βελτίωσης που προσφέρουν οι ενταμευτές τοποκόητας μετρήσαμε το συνολικό γόρτο του SAE μέσω του χρόνου χρήσης (user time) για κάθε ανιγνωστή. Επίσης, δεδομένου ότι ο διαμορφωμός γόρτος δεν είναι τέλειος μετρήσαμε τον χρόνο χρήσης του πιο βαρυφορμωμένου ανιγνωστή.

Μελετάμε πως επηρέαζε στην απόδοση του SAE το πλήθος των ET, το μέγεθος τους, το ποσό τους κατανάλωμε σε ροές καθώς επίσης και το πλήθος των ανιγνωστών.

Πλήθος Ανιγνωστών

Η Εικόνα 4.4 δείχνει τον συνολικό χρόνο χρήσης για διαφορετικό πλήθος ανιγνωστών και η Εικόνα 4.5 δείχνει τον χρόνο χρήσης του πιο βαρυφορμωμένου ανιγνωστή. Για τις περιπτώσεις αυτές χρησιμοποιήσαμε 16 ET των 256 KB και την μέθοδο Σ.Β. Όπως βλέπουμε, ο συνολικός χρόνος χρήσης βελτιώνεται τουλάχιστον 11.4% (για 8 ανιγνωστές) και γενικά 13% (για 40 ανιγνωστές).

Μειώνει ενδιαφέρουσα παρατήρηση με βάση την Εικόνα 4.4 είναι ότι καθώς το πλήθος των ανιγνωστών αυξάνεται, η επίδραση των ενταμευτών τοποκόητας μειώνεται ελαφρά. Μάλιστα, επιφέρει μια βελτίωση στην απόδοση και χωρίς την χρήση ενταμευτών. Η ατία του πραγματικού είναι στο ότι η απόδοση των
4.4. ΠΕΙΡΑΜΑΤΑ

Σχήma 4.4: Συνολικός χρόνος χρήσης για όλους τους ανιχνευτές σε σχέση με το πλήθος των ανιχνευτών.

Σχήma 4.5: Χρόνος χρήσης του πιο βαρυμετρημένου ανιχνευτή σε σχέση με το πλήθος των ανιχνευτών για το ίδιο περιόδημα της εικόνας 4.4.
ΣΧΗΜΑ 4.6: Μέσο μέγεθος ριπής πακέτων σε σχέση με το πλήθος των ανιχνευτών για το πείραμα των Εικόνων 4.4 και 4.5.

πακέτων σε περισσότερους ανιχνευτές αυξάνει την πιθανότητα πακέτα του ιδίου τύπου να βρεθούν το ένα πίσω από το άλλο. Πάρολα, η θετική επίδραση των ενταμένων τοποκότητας είναι φανερή.

Για βεβαιωθούμε για τις υποθέσεις μας, μετρήσαμε το μέσο μέγεθος ριπής πακέτων για τα παραπάνω πειράματα. Δηλαδή μετρήσαμε πόσα πακέτα ιδίου τύπου αποστέλλονται σε έναν ανιχνευτή το ένα πίσω από το άλλο κατά μέσο όρο. Τα αποτελέσματα που βρήκαμε παρουσιάζονται στην Εικόνα 4.6.

Παρατηρούμε ότι το μέσο μέγεθος ριπής σχεδόν διπλασιάζεται όταν χρησιμοποιούμε ΕΤ. Στην περίπτωση που δεν χρησιμοποιούμε ΕΤ παρατηρούμε ότι το μοίρασμα σε πολλαπλούς ανιχνευτές, επίσης ευνοεί το μέσο μέγεθος ριπής πακέτων. Το ιστόγραμμα του μέσου μέγεθους ριπής για την περίπτωση των τεσσάρων ανιχνευτών φαίνεται στην Εικόνα 4.7. Βλέπουμε ότι περίπου τα μισά πακέτα ομαδοποιούνται σε κόπτοι ριπή ενώ ένα 5% ανήκει σε ριπές μεγαλύτερες από 17 πακέτα.

Διαστάσεις ΕΤ

Σε αυτό το σύνολο πειραμάτων διερευνούμε πώς το πλήθος και το μέγεθος των ΕΤ επηρεάζει την απόδοση. Κρατάμε σταθερό το πλήθος των ανιχνευτών σε τέσσερεις όπως επίσης και την μέθοδο κατανομής πακέτων στο ΣΒ. Σε κάθε πείραμα μετράμε το ποσοστό βελτίωσης, δηλαδή πόσο τις εκατο μειώνουμε τον χρόνο χρήστη σε σχέση με την περίπτωση που δεν χρησιμοποιούμε ΕΤ.

Η Εικόνα 4.8 δείχνει τα αποτελέσματα για διαφορετικά πλήθος στο ΕΤ ανά ανιχνευτή όταν το μέγεθος τους είναι 256 KB. Παρατηρούμε μία βελτίωση του συνολικού χρόνου χρήστη που κυμαίνεται από 6.8% (για 4 ΕΤ) μέχρι 12.9% (για 32 και 64 ΕΤ). Περισσότεροι από 32 ΕΤ δεν βελτιώνουν σημαντικά το συνολικό
4.4. ΠΕΙΡΑΜΑΤΑ

Σχήμα 4.7: Ιστόγραμμα του μεγέθους ρυθμός με χρήση ET και χωρίς ET για την περίπτωση τεσσάρων αντικειμένων της Εικόνας 4.6.

Σχήμα 4.8: Βελτίωση της απόδοσης για διαφορετικό πλήθος ET.
% Performance Improvement by LB size

<table>
<thead>
<tr>
<th>Locality Buffer Size (KB)</th>
<th>User time of most loaded sensor</th>
<th>Aggregate user time</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>256</td>
<td></td>
<td></td>
</tr>
<tr>
<td>512</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

% Improvement

0 2 4 6 8 10 12 14 16 18

ΣΧΗΜΑ 4.9: Βελτίωση της απόδοσης για διαφορετικό μέγεθος ET.

χρόνο χρήστη αν και υπάρχει μια μικρή βελτίωση στο χρόνο χρήστη του πιο βαρυφροτομομένου ανιχνευτή. Οπότε η χρήση 32 ET είναι μια αρχετά λογική επιλογή.

Η Εικόνα 4.9 δείχνει την βελτίωση της απόδοσης στον συνωλικό χρόνο χρήστη για διαφορετικό μέγεθος από ET. Η μείωση χρησιμοποιείται από 9.3% σε 13.31% για τις περιπτώσεις των 64 KB και 512 KB αντίστοιχα. Η επιλογή των 256 KB φαίνεται λογική μιας και για 512 KB η βελτίωση είναι ορίσιμη.

Πολιτωτικής Χρήσης ET

Σε αυτό το μέρος εξετάζουμε πως οι διάφορες ευριστικές μέθοδοι για την επιλογή ET επηρεάζουν την απόδοση. Για το σετ περιμέτρων αυτού του τρήματος χρησιμοποιήσαμε τέσσερες ανιχνευτές, 16 ET ανα ανιχνευτή και 256 KB ανα ET. Όπως και πριν, μετράμε το ποσοστό βελτίωσης του χρόνου χρήστη, δηλαδή ποσοτικά μείωσεται σε σχέση με το να μην χρησιμοποιούμε ET.

Η Εικόνα 4.10 δείχνει την βελτίωση της απόδοσης ως προς τον συνωλικό χρόνο χρήστη άλλα και ως προς τον χρόνο χρήστη του πιο βαρυφροτομομένου ανιχνευτή. Όπως παρατηρούμε, οι μέθοδοι που εμπλέκουν μονο την τελική πόρτα, \( (T \text{ and } \Sigma_T) \) έχουν καλύτερη απόδοση από ότι οι μέθοδοι που εμπλέκουν και τις δύο πόρτες. Για τις τελευταίες η βελτίωση στην απόδοση είναι περίπου 4.19% εκ τω για τις πρώτες η βελτίωση είναι περίπου 7%. Επίσης παρατηρούμε ότι η βελτίωση είναι μεγαλύτερη όταν χάσει ΕΤ παίρνουν ένα μόνο είδος χίνησης. Αυτό φαίνεται από τις μεταφέρε των \( \Sigma_T \) αντί \( \Sigma_T \) στις οποίες η βελτίωση είναι 12.19% και 12.21% αντίστοιχα. Αυτό είναι αναμενόμενο μιας και ένα μεγάλο μέρος του αρχείου πακέτων είναι χίνηση Web στάντ άν υπάρχει ET αφενερωμένη σε τέτοια περίπτωση θα προκύψουν μεγαλύτερες ριπές πακέτων.
4.4. ΠΕΙΡΑΜΑΤΑ

% Performance Improvement
by LB Allocation Method

- User time of most loaded sensor
- Aggregate user time

% Improvement

16
14
12
10
8
6
4
2
0

SD D p_SD p_D p_T_D T_D

Locality Buffer Allocation Method

ΣΧΗΜΑ 4.10: Βελτίωση για διαφορετικές πολιτικές χρήσης των ET.

ίδιου τύπου.

Το να μιμηθούμε τους ET ανα πρωτόκολλο φαίνεται ότι δεν έχει ιδιαίτερη επιδράση. Αυτό φαίνεται από τις μπάρες \( \Sigma_T \) και \( p_\Sigma T \) καθώς επίσης από τις μπάρες \( T \) και \( p_T \). Ο λόγος είναι ότι στο αρχείο πακέτων που έχουμε τα TCP πακέτα είναι η πλευρονομία. Οπότε δεν έχει μεγάλη σημασία να διαχωρίζουμε τα πακέτα ανα πρωτόκολλο.

4.4.4 Αξιολόγηση Φιλτραρίσματος Χαμηλού Επιπέδου και ET

Για να δούμε την επίδραση των δύο τεχνικών στην απόδοση του ΣΑΕ κάναμε το παραπάνω πείραμα. Εφαρμόσαμε την τεχνική του φιλτραρίσματος σε χαμηλό επίπεδο και έτσι ακριβέσαμε από το αρχείο πακέτων όλα τα πακέτα που απέρριψε αυτή η μέθοδος. Κατοπιν για τα υπόλοιπα πακέτα εκτελέσαμε το πείραμε με τους τεσσάρες ανιχνευτές, με 16 ET ανα ανιχνευτή και με 256 KB ανα ET. Χρησιμοποιήσαμε την μέθοδο Σ\( \Sigma \)T για τους ET. Ο συνολικός χρόνος χρήσης που μετρήσαμε ήταν 37.88 δευτ., ο οποίος είναι 8.9% φορές καλύτερος από την περίπτωση όταν χρησιμοποιούμε φιλτράρισμα σε χαμηλό επίπεδο (41.61 δευτ.). Αν δεν εφαρμόσουμε χαμία τεχνική, ο χρόνος χρήσης είναι 47.27 δευτ., οπότε η βελτίωση που πετυχάμε είναι 19.8%. Για τον πιο βαρύ-φορτωμένο ανιχνευτή, η βελτίωση ως προς την περίπτωση που χρησιμοποιούμε μόνο ET είναι 5% (από 11.52 δευτ. σε 10.93 δευτ.). Χωρίς χαμία μέθοδος η βελτίωση ανέρχεται σε 14.4%.
4.5 Περίληψη

Σε αυτό το κεφάλαιο παρουσιάζουμε την αρχιτεκτονική ενός ενεργού διαμοιραστή χίνησης για ανίχνευση επιθέσεων. Ο ενεργός διαμοιραστής, εκτός του ότι διαμοιράζει την χίνηση, χρησιμοποιεί απλές μεθόδους επεξεργασίας της χίνησης ώστε να βελτιώνει την απόδοση των συστημάτων ανίχνευσης εισβολέων.

Παρουσιάζουμε και αναλύουμε δύο τέτοιες μεθόδους, το οικτράρισμα σε χαμηλό επίπεδο και τους ενταμευτές τοπικότητες. Η πρώτη μέθοδος αποτελείται από την επεξεργασία κάποιων πακέτων απευθείας στον διαμοιραστή. Ετσι, οπως είδαμε ο φόρτος των ανιχνευτών μειώνεται και πετυχάμε πολύ μικρή μέμβρα του χρόνου χρήσης κατά περίπου 8%. Οι υπογραφές που πρέπει να ελεγχθούν στον διαμοιραστή απαιτούν μόνο ανάλυση επικεφαλής και αποτελούν ένα μικρό υποσύνολο όλων των υπογραφών. Η δεύτερη μέθοδος, οι ενταμευτές τοπικότητας, συμβάλλουν στα πακέτα παρόμοια με τον τρόπο που τα ΣΑΕ ομαδοποιούν τις υπογραφές. Ετσι ιδίου τύπου πακέτα στέλνονται σε ρίτες προς τους ανιχνευτές. Η μέθοδος αυτή αυξάνει την τοπικότητα αναφορών στους ανιχνευτές και έτσι βελτιώνεται η απόδοση. Με 32 ET των 256 KB πετύχαμε ένα 10% μείωση στον συνολικό φόρτο του ΣΑΕ. Χρησιμοποιώντας και τις δύο μεθόδους η βελτίωση του χρόνου χρήσης φτάνει στο 19.8% για το συνολικό σύστημα, ενώ ο πιο βαρυφορτωμένος ανιχνευτής βελτιώνεται κατά 14.4 και αποτελούν ένα μικρό υποσύνολο όλων των υπογραφών. Η δεύτερη μέθοδος, οι ενταμευτές τοπικότητας, συμβάλλουν στα πακέτα παρόμοια με τον τρόπο που τα ΣΑΕ ομαδοποιούν τις υπογραφές. Ετσι ιδίου τύπου πακέτα στέλνονται σε ρίτες προς τους ανιχνευτές. Η μέθοδος αυτή αυξάνει την τοπικότητα αναφορών στους ανιχνευτές και έτσι βελτιώνεται η απόδοση. Με 32 ET των 256 KB πετύχαμε ένα 10% μείωση στον συνολικό φόρτο του ΣΑΕ. Χρησιμοποιώντας και τις δύο μεθόδους η βελτίωση του χρόνου χρήσης φτάνει στο 19.8% για το συνολικό σύστημα, ενώ ο πιο βαρυφορτωμένος ανιχνευτής βελτιώνεται κατά 14.4%.

Η δουλειά αυτή έγινε δεκτή στο 11ο IEEE/ACM Διεθνές Συμπόσιο τάνω στο Modelling, Ανάλυση και Προσομοίωση Συστημάτων Προλογιστών και Τηλεπικοινωνιών (MASCOTS 2003) στο Ορλάντο της Φλόριντα των Ηνωμένων Πολιτειών Αμερικής.
Κεφάλαιο 5

Υλοποίηση του Διαμοιραστή στον IXP 1200

Σε αυτό το κεφάλαιο παρουσιάζουμε την υλοποίηση ενός Ενεργού Διαμοιραστή για δύο 1 Gbit/s με την χρήση του ΕΔ IXP 1200. Αυτό το κεφάλαιο είναι η φυσική συνέχεια της δουλειάς του κεφαλαίου 4 κατά την οποία μελετήσαμε το κέρδος σε απόδοση χρησιμοποιήσαμε έναν Ενεργό Διαμοιραστή. Στο κεφάλαιο λοιπόν αυτό παρουσιάζουμε την σχεδίαση ενός τέτοιου διαμοιραστή με την βοήθεια του IXP 1200. Κατόπιν, επαληθεύουμε τις μετρήσεις που έχουμε πάρει με προσομοιώσεις χρησιμοποιώντας μετρήσεις σε πραγματικό περιβάλλον. Παρά το γεγονός ότι ο τρέχον Ενεργός Διαμοιραστής υλοποιεί μόνο την μέθοδο των ET, η βελτίωση της απόδοσης είναι σημαντική. Επίσης αρκετοί πόροι από τον IXP 1200 υπάρχουν διαθέσιμοι για μελλοντική χρήση όπως για παράδειγμα για την υλοποίηση του φιλτραρίζομενο σε χαμηλό επίπεδο.

Η δομή του τρέχοντος κεφαλαίου είναι η ακόλουθη. Ξεκινάμε με μία περιγραφή της υλοποίησης του Ενεργού Διαμοιραστή στον IXP 1200. Αυτή χωρίζεται σε τρία μέρη. Στο πρώτο μέρος (παράγραφος 5.1.3), παρουσιάζουμε το τμήμα λήψης πακέτων. Στο δεύτερο μέρος (παράγραφος 5.1.4) παρουσιάζουμε την αποστολή πακέτων. Στο τρίτο μέρος περιγράφουμε τον συγχρονισμό μεταξύ αποστολής και λήψης πακέτων. Στο επόμενο τμήμα του κεφαλαίου παρουσιάζουμε τα πειράματα και τα αποτελέσματα που πήραμε από τον πραγματικό Ενεργό Διαμοιραστή. Τέλος στο τελευταίο τμήμα συνοψίζουμε και συζητάμε μελλοντικές εργασίες στο ίδιο πλαίσιο.
5.1 Υλοποίηση

5.1.1 Περιλήψη του Συστήματος

Η Εικόνα 5.1 μας δείχνει περιληπτικά την αρχιτεκτονική του κατανεμημένου συστήματος ανιχνευσής εισβολέων που βασίζεται στον IXP-Διαμοιραστή και σε δύο snort-ανιχνευτές. Τα μέρη που αποτελούν το σύστημα είναι:

- Μια πηγή κίνησης που παράγει την κίνηση που πράκεται να παρακολουθήσουμε.
- Έναν διαμεταγωγέα πακέτων (switch) που δρομολογεί τα πακέτα μεταξύ των οντοτήτων του συστήματος.
- Την πλατφόρμα ENP-2506 στην οποία τρέχει ο IXP Διαμοιραστής.
- Δύο ανιχνευτές που βασίζονται στο snort.

Η πηγή κίνησης δημιουργεί ethernet πακέτα είτε δυναμικά, είτε διαβάζοντας από ένα αρχείο πακέτων. Τα πακέτα μέσω του switch θα δρομολογηθούν στην πλατφόρμα ENP-2506. Εκεί βρίσκεται το IXP ο οποίος υλοποιεί τον διαμοιραστή. Ο διαμοιραστής θα χωρίζει την κίνηση στα δύο, μία ροή για κάθε ανιχνευτή. Οι δύο ροές κίνησης θα σταλούν πίσω στο switch το οποίο και θα τις δρομολογήσει κατάλληλα. Κάθε ανιχνευτής τρέχει το snort.

5.1.2 Περιλήψη Διαμοιραστή

Η Εικόνα 5.2 συνοψίζει τις εργασίες που πρέπει να εκτελέσει το λογισμικό που υλοποιεί τον διαμοιραστή πάροικο στον IXP. Για κάθε εισερχόμενο πακέτο
Ο διαμορφωτής θα εκτελέσει μία hashing συνάρτηση και θα το στείλει σε ένα ανιγνωστή. Όλα τα πακέτα της ίδιας ροής στέλνονται πάντα στον ίδιο ανιγνωστή. Είναι αξίο σημείωση ότι η συνάρτηση hashing υλοποιείται απευθείας από το υλικό του IXP.

Μετά που θα απορριπτεί σε ποιον ανιγνωστή θα προωθηθεί ένα πακέτο, το πακέτο καταχωρείται σε ένα ενταμιέυτη τοποκότητας. Οι ET χωρίζονται ως εξής: ένα μορφό μέρος τους είναι αφιερωμένο σε συγκεκριμένη κίνηση. Η κίνηση web, κίνηση peer to peer, κ.α. Οι υπόλοιποι ET δέχονται όλων των ειδών τα πακέτα. Για κάθε πακέτο ένα hashing στην τελική πόρτα θα δώσει τον αριθμό του ET στον οποίο πρέπει να αποθηκευτεί. Η τρέχουσα υλοποίηση χρησιμοποιεί 5 ET αφιερωμένους σε συγκεκριμένη κίνηση και 16 ET γενικής χρήσης με βάση την εμπειρία του προηγούμενου κεφαλαίου.

Όταν ένα ΛΒ βεομες ψηλά, άλλα τα πακέτα αρχικά ενεγκενέδιν στην τρανσμιτ γχενε (ζιμοτ γχενε) ανά τρανσμιτέρες το ικανό σενσόρ ακ α βικ βιρατ (βακτ ακ βακτ). Ας δεσερβρίδων δεταλ δια προς χρηστήρ, της υπο ορ Δοκαλιτή Βυφτίος αμη αν ενεργοποιήσει σε η εκατέρο Σενσόρ. Με της ως, της περιστασιας ας εκατέρο Σενσόρ ας αλλά ας της σεφαλέ περιστασιας ας μπροσέδι.

Όταν ένας ET γεμίσει τότε όλα τα πακέτα του ενσωματώνονται στην ουρά αποστολής (xmit queue). Με τον τρόπο αυτό αυξάνεται η τοποκότητα αναφορών στον τελικό ανιγνωστή. Ετσι η απόδοση του ανιγνωστή βελτιώνεται, όπως αναλυτικά περιγράφαμε στο προηγούμενο κεφάλαιο.
ΣΧΗΜΑ 5.3: Κατανομή εργασίας στα στοιχεία του IXP
5.1. ΥΛΟΠΟΙΗΣΗ

Στην Εικόνα 5.3 βλέπουμε ποιά οριζόντια εργασία αναλαμβάνει κάθε ένα από τα προγράμματα ζώμενα στοιχεία του ΙΧΡ. Επίσης, στην ίδια εικόνα φαίνεται ο τρόπος που διαχειρίζομαι την μνήμη. Πιο αναλυτικά:

Ινές Λήψης. Δύο μηχανές μοιράζονται την διαδικασία λήψης πακέτων. Δηλαδή, οποιεσδήποτε λήψεις πακέτων αναλαμβάνουν να παραλάβουν ένα πακέτο, να βρουν σε ποιον ανήκει ανάγκη πρέπει να το στείλουν και να το βάλουν στον κατάλληλο ET. Αν ένας ET γεμίσει οι ινές λήψεις αναλαμβάνουν να τον βάλουν στην ουρά αποστολής (xmit queue).

Ινές Αποστολής. Μία μηχανή αναλαμβάνει την αποστολή πακέτων χρησιμοποιώντας τέσσερεις ινές. Μία ινή αναλαμβάνει την διαχείριση της ουράς αποστολής, ενώ για κάθε πακέτο οι τρεις υπώλοιπες ινές αναλαμβάνουν την αποστολή του.

Αδειασμα των ET. Μία μηχανή περιοδικά ελέγχει τους ET και τους αδειάζει αν έχουν μείνει για πολύ άρα με τα ιδία πακέτα. Ετσι αποφεύγουμε το φαινόμενο να μείνουν κάποια πακέτα στον ET κατά το κλείσιμο (shutdown) του διαμορφαστή.

Αναφορά Κατάστασης. Μία μηχανή μαζεύει περιοδικά στατιστικά σχετικά με την εκτέλεση του διαμορφαστή. Για παράδειγμα, πόσα πακέτα έχουν περάσει, πόσα ήταν SOP, πόσα ήταν EOP, x.o.x. Αυτά τα στατιστικά στοιχεία τα βάζει σε ένα πακέτο ethernet και το τοποθετεί στην ουρά αποστολής. Αυτό το πακέτο προορίζεται για έναν υπολογιστή ο οποίος θα παρατηρεί την εργασία του διαμορφαστή.

StrongArm. Ο StrongArm τρέχει το Linux και είναι υπεύθυνος για το ξεκίνημα του διαμορφαστή και το σταμάτημα του. Επίσης χρησιμοποιήθηκε εκτενώς για την αποστολή του προγράμματος.

Παρατηρούμε ότι μία μηχανή δεν χάνει τίποτα. Επίσης η διουλεύση των μηχανών 2 και 3 μπορεί να συμπεριλάβει σε μία μηχανή. Ετσι, αλλάζουμε το λογισμικό μπορεί εύκολα να χωρέσει σε τέσσερεις μηχανές, αφήνοντας αρκετό χώρο για μελλοντικές επεκτάσεις (π.χ. υλικά σε χρηματικό επίπεδο). Σχετικά με την κατανομή δεδομένων:

SDRAM 32 ΜΒ από την SDRAM χρησιμοποιούνται για την αποθήκευση των πακέτων που έρχονται από το δίκτυο. Αν και η πλατφόρμα ENP-2506 έχει 256 ΜΒ μνήμης, η σχεδίαση έγινε έτσι ώστε η διαμορφαστής να μπορεί να τρέξει σε οποιαδήποτε πλατφόρμα διαθέτει τουλάχιστον 32 MB μνήμης.

SRAM 2 MB από την SRAM χρησιμοποιούνται για:

- Μία λίστα από περιγραφές πακέτων (packet descriptors).
5.1.3 Διαδικασία Λήψης

Η Εικόνα 5.4 περιγράφει τον αλγόριθμο που εκτελούν οι ίνες λήψης πακέτου.

Αρχικοποίηση (Initialization)

Κατά την αρχικοποίηση προ-υπολογίζουμε τιμές που θα χρειαστούμε συχνά κατά την διάρκεια εκτέλεσης του κώδικα λήψης πακέτων. Τέτοιες πληροφορίες είναι οι αιτήσεις για λήψη πακέτων (receive requests), οι πληροφορίες για την ενημέρωση της επόμενης ίνας, πληροφορίες σχετικά με την επόμενη κενή περιγραφή πακέτου (packet descriptor), κτλ. Αυτή η τεχνική ουσιαστικά σταθμίζει δύο ανταγωνιστικές έννοιες: την δέσμευση καταχωρητών και την ταχύτητα εκτέλεσης του κώδικα. Δηλαδή, να μην από την μια πλευρά θέλουμε σοστά το δυνατόν περισσότερες τιμές έτοιμες σε κάποιο καταχωρητή ωστε να μην χάνουμε χρόνο, από την άλλη όμως οι καταχωρητές είναι περιορισμένου αριθμού και αρα δεν θέλουμε να σπαταλήσουμε αρχετούς.

Οι αρχικοποιήσεις τελεύωνουν σε ένα σημείο αναγκάζεις (barrier). Όλες οι ίνες λήψης περιμένουν να συναντηθούν στο σημείο αναγκάζεις πριν ξεκινήσουν την εκτέλεση του κώδικα λήψης πακέτων. Αυτό είναι αναγκαίο, γιατί ο τρόπος που προγραμματίζεται κάθε μηχανή είναι ότι φορτώνεται ο κώδικας της και αρχίζει αμέσως η εκτέλεσή του. Εάν, αν για παράδειγμα η ίνα ένα της μηχανής ένα φτάσει στο σημείο που στέκει σήμα στην ίνα 4 της μηχανής 2 πριν η μηχανή 2 προγραμματιστέτει, το σήμα θα χάθει και θα η μηχανή 2 θα περιμένει επ αόρατο. Είναι σημαντικό να παρατηρήσουμε ότι αυτή η συμπεριφορά μπορεί να εμφανιστεί μόνο κατά την εκτέλεση κώδικα στο πραγματικό υλικό και δεν είναι αναγκαστική στο προσωπικό.

Αίτηση για Δουλειά (Request Work)

Η Εικόνα 5.5 μας δείχνει την σειρά με την οποία οι ίνες λήψης ζητούν δουλειά. Οι ίνες ζητούν δουλειά εναλλάξ, μία από την μία μηχανή και μία από την άλλη προεκμένου να υπάρχει εξ αρχής καλή ισορροπία φόρτου.

Μέτα που μία ζήτηση θα πάρει σήμα από την προηγούμενη ότι μπορεί να ξεκινήσει, θα αρχίσει να κοστίζει την σήμανση ‘έτοιμη λήψη’ της πόρτας εισόδου. Όταν αυτή ενεργοποιηθεί, τότε υπάρχουν αρχετά δεδομένα ώστε να μεταφέρθουν εσωτερικά στον IXP. Όπως η ίνα λήψης θα ξεκινήσει μία αίτηση
Σχήμα 5.4: Τα βήματα των ινών λήψης πακέτου

Σχήμα 5.5: Σειρά εκτέλεσης των Ινών Λήψης
希腊文：

54 ΚΕΦΑΛΑΙΟ 5. ΤΑΞΙΩΝΗΣΗ ΤΟΥ ΔΙΑΜΟΡΦΑΣΤΗ ΣΤΟΝ ΙΧΡ 1200

ΣΧΕΜΑ 5.6: Δομή του κουτιού μνημημάτων (message box)

για την μεταφορά του μέγιστου αριθμού μπακέτων από τον ενταμεντή της πόρτας εισόδου εσωτερικά στην μηνή λήψης του FBI. Ο μέγιστος αριθμός μπακέτων είναι δύο και ο λόγος που προσπαθούμε να φέρουμε πάντα δύο μπακέτα ανα αίτηση λήψης είναι για να ελαχιστοποιήσουμε τις εναλλαγές στην κατεύθυνση των δεδομένων του IQ διαύλου.

Πρέπει να παρατηρήσουμε ότι υπάρχει μία σημαντική χαλάστρηση από την στιγμή που θα ενεργοποιηθεί η σήμανση 'έτοιμη λήψη' μέχρι την στιγμή που πραγματικά, τα αντίστοιχα μπακέτα θα αρχίσουν να μεταφέρονται. Για τον λόγο αυτό μία αίτηση λήψης μπορεί:

- Είναι να συμφωνείτε επειδή δεν υπάρχουν δεδομένα.
- Είναι να μεταφέρετε ένα μπακέτο αντί για δύο.
- Είναι κανονικά να μεταφερθούν δύο μπακέτα.

Στην περίπτωση που ένα μπακέτο έχει την ένδειξη SOP, δηλαδή είναι η αρχή ενός καινούριου πακέτου, η αντίστοιχη ίνα θα χρησιμοποιήσει την ελεύθερη περιγραφή πακέτου που έχει πρόγευση για να δεί που στην SDRAM θα αποθηκεύτει το καινούριο πακέτο. Κατόπιν θα προωθήσει αυτή την πληροφορία στην επόμενη ίνα.

Η μεταφορά μνημημάτων από την μία ίνα στην επόμενη γίνεται μέσω συγκεκριμένων θέσεων μηνής στην SDRAM, τα κουτιά μνημημάτων (message boxes). Η δομή ενός κουτιού μνημημάτων χαρακτηρίζεται στην Εικόνα 5.6. Κάθε ίνα λήψης αρχικά περιμένει το κουτί της να φυλακτεί από την προηγούμενη, πριν συνεχιστεί στον κώδικα της. Κατόπιν, διαβάζει τα περιεχόμενα του κουτιού της, τα αυξάνει και με βάση το μπακέτο της ενημερώνει το κουτί της επόμενης ίνας. Με τον τρόπο αυτό οι ίνες συνεργάζονται και αποδημάτικα τα πακέτα σε καινούριες θέσεις στην μηνή.

Επιπλέον τα πακέτα σειροποιούνται και έτσι δεν χάνεται η σειρά τους. Ο χρονισμός των ινών λήψης χαρακτηρίζεται στο σχήμα της Εικόνας 5.7.

Μετά την μεταφορά ενός μπακέτου στην SDRAM, κάποιοι μετρητές ενημερώνονται και αν αυτό ήταν το τελευταίο μπακέτο και ένας πακέτου τότε τα δεδομένα πακέτου (Εικόνα 5.8) επισής ενημερώνονται. Όπως χαρακτηρίζεται στα δεδομένα πακέτου καταχωρείται και ο ET στον οποίο βρίσκεται το τρέχον πακέτο. Η
5.1. ΥΛΟΠΟΙΗΣΗ

ΣΧΗΜΑ 5.7: Συγχρονισμός των Ινών Λήψης

Packet meta data

ΣΧΗΜΑ 5.8: Δεδομένα Πακέτου
ΣΧΗΜΑ 5.9: Δεδομένα Ενταμευτών Τοπικότητας

ΣΧΗΜΑ 5.10: Δεδομένα ουράς ενός Ενταμευτή Τοπικότητας

απόφαση σε ποιόν ΕΤ θα πάει ένα πακέτο παίρνεται απο την ίνα που θα επεξεργαστεί το SOP μΠακέτο.

Όπως, η ίνα που θα επεξεργαστεί το EOP μΠακέτο θα ξεκινήσει με το ποιόν ΕΤ να βάλει το συγκεκριμένο πακέτο. Ωστόσο τα αντίστοιχα δεδομένα θα ενημερωθούν (Σχόλια των 5.9 και 5.10). Αν ο συγκεκριμένο ΕΤ έχει γεμίσει τότε όλα τα πακέτα του θα ενσωματωθούν στην ουρά αποστολής. Το τελευταίο βήμα απαιτεί την άλλη μόνο του δείκτη ουράς (tail pointer) της ουράς αποστολής.

5.1.4 Διαδικασία Αποστολής

Περιθώριες της Διαδικασίας Αποστολής

Τρεις ίνες συνεργάζονται για την αποστολή ενός πακέτου. Καθε ίνα αναλαμβάνει την αποστολή δύο ή ένας μΠακέτων από το τέρμα του πακέτου. Οι τρεις ίνες εκτελούν τον ίδιο κώδικα που εφαρμόζεται παρακάτω:

1. Καταγράφωση των επόμενων δύο (ή ένα) μΠακέτων.
2. Μεταφορά τους από την SDRAM στον Ιντερμετάτη Αποστολής.
3. Καθοδομάς των πληροφοριών για την αποστολή των μΠακέτων.
4. Αναμονή της προηγούμενης ίνας.
5. Αναμονή συσκευής (πάρτας).
6. Ξεκίνημα αποστολής των μΠακέτων.
7. Αν το τελευταίο μΠακέτο ήταν EOP μΠακέτο, τότε επέστρεψε την περιγραφή πακέτου (packet descriptor) στην στοά, με τις ελεύθερες περιγραφές πακέτων.
5.1. ΥΛΟΠΟΙΗΣΗ

Κοινή δομή δεδομένων για την αποστολή

Ο συγχρονισμός των τριών ιων για την αποστολή ενός πακέτου γίνεται μέσω μίας κοινής δομής που περιγράφει το τρέχον πακέτο και την κατάσταση αποστολής του. Η δομή αυτή έχει τα παρακάτω πεδία:

sdramAdd: Διεύθυνση στην SDRAM στην οποία ξεκινάνε τα δεδομένα του πακέτου.

dsc: Η περιγραφή του πακέτου (descriptor) από την οποία προσέπτεται το πεδίο sdramAdd.

firstMpktNo: Αριθμητική του SOP μΠακέτου. Όλα τα μΠακέτα αριθμούνται με έναν αύξοντα αριθμό (ο οποίος επιτρέπεται να υπεργειληθεί).

lastMpktNo: Αριθμός του τελευταίου μΠακέτου του τρέχοντος πακέτου.

pktMetaData: Πληροφορίες σχετικά με το πακέτο. Πρόκειται για ένα αντιγραφή της δομής της Εκόνας 5.8.

lastMpktValidBytes: Πλήθος έγχυμων bytes στο EOP μΠακέτο.

Οι ίνες που αναλαμβάνουν την αποστολή ενός πακέτου, κλειδώνουν, διαβάζουν, ενημερώνουν και μετα ξεκλειδώνουν αυτή την δομή. Αυτές οι λειτουργίες γίνονται πάρα πολύ γρήγορα μιας και άλλες οι ίνες τρέχουν στην ίδια μηχανή. Η λειτουργία ομοιότατης που είναι σχετικά αργή είναι η αφάνεια ενός πακέτου από την ουρά αποστολής προκειμένου να το στείλουμε. Και αυτό γιατί οι σχετικές δομές βρίσκονται αναγκαστικά και οινόρχηστη μνήμη. Για το λόγο αυτό, υπάρχει μια ίνα που έχει αναλάβει αποσκευαστική αυτή την δουλεία (Ελεγκτής Αποστολής) και παρουσιάζεται στην επόμενη παράγραφο.

Η Ινα Ελεγκτής Αποστολής

Ο Ελεγκτής Αποστολής εκτελείται στην ίδια μηχανή με τις ίνες αποστολής και έτσι οι μεταξύ τους επικοινωνίας απλοποιείται. Ο ρόλος του είναι να αποκρύψει την καθυστέρηση που έχει η διαδικασία αφαίρεσης ενός πακέτου από την ουρά αποστολής. Αυτό επιτυγχάνεται με το να αποκρύπτεται ο χρόνος αυτός με την αποστολή ενός πακέτου την οποία εκτελούν παράλληλα οι υπάλληλοι ίνες. Ωστόσο, η λειτουργία του είναι η εξής:

1. Πάρε το επόμενο πακέτο για αποστολή.

(α') Διάβασε τις πληροφορίες για την ουρά αποστολής.
(β') Πάρε το πακέτο που βρίσκεται στο τέλος της ουράς.
ΣΧΗΜΑ 5.11: Προσθήκη ενός ET στην ουρά αποστολής.

2. Προστεθείσε την δομή packet_xfer_data.

3. Περίμενε μέχρι το τρέχων πακέτο να αποσταλεί πλήρως.

4. Ενεργοποιήσε την προετοιμασμένη δομή packet_xfer_data.

5. Επανέλαβε.

5.1.5 Συγχρονισμός μεταξύ Ιστότοπων Λήψης και Ελεγκτή Αποστολής

Ο συγχρονισμός των ιστότοπων λήψης και του Ελεγκτή Αποστολής γίνεται μέσω της κοινόχρησης ουράς αποστολής. Οι ίνες λήψης προσθέτουν πακέτα στην ουρά αποστολής και ο Ελεγκτής Αποστολής αφαιρεί απο την αρχή της. Για να υπάρξει συνοχή (consistency) ο Ελεγκτής Αποστολής δεν θα περάζει την δομή οσο βλέπει στις περιέχει ένα μόνο πακέτο.
5.1.6 Συγχρονισμός των Ινών Λήψης

Δεδομένου ότι όλες οι Ινές λήψης πρέπει να προσπελάσουν την ουρά αποστολής, είναι απαραίτητο ένα σχήμα συγχρονισμού. Μάλιστα, επειδή οι Ινές λήψης βρίσκονται σε δύο μηχανές, ο μηχανισμός συγχρονισμού πρέπει να βασίζεται σε κάποιον κοινόχρηστο πόρο. Στην συγκεκριμένη υλοποίηση χρησιμοποιούμε τον SRAM. Ενα θέμα που δημιουργείται είναι η συγνώμη των κλειδιώματων/εξελεκτρολήματων των κοινόχρηστων δομών. Αν γίνονται συγχέως προσπελάσεις τότε η απόδοση θα μειωθεί δραστικά. Αυτό οφείλεται στο ότι για άμεση ώρα μία άλλη προσπαθεί να πάρει το κλείδι μιας κοινόχρηστης δομής καταναλώνει και άλλου κοινόχρηστους πόρους (της τους διαύλους). Έτσι εμποδίζει την ομαλή λειτουργία του συστήματος μέχρι να πάρει τελικά το κλείδι. Οπότε είναι σημαντικό το σχήμα συγχρονισμού να μειώνει δραστικά το ποσοστό κλειδιώματων/εξελεκτρολήματων. Η τρέχουσα υλοποίηση διαφοροποιεί δύο περιπτώσεις ανάλογα με το αν έχουμε ET ή αν δεν έχουμε.

Συγχρονισμός όταν υπάρχουν ET

Όταν υπάρχουν ET τότε το απλό κλειδίωμα και εξελεκτρόλημα αρχεί και δεν δημιουργεί προβλήματα. Ο λόγος είναι ότι λόγω των ET τα κλειδιώματα/εξελεκτρολήματα δεν γίνονται συγχέως επιλεγόμενα πρόστιμα αρκετά παρεκτά, γεμίζει ένας ET και μετά όλα μαζί με ένα κλειδίωμα/εξελεκτρόλημα μπαίνουν στην ουρά αποστολής. Οπότε οι ET ρίγουν δραστικά το πλήθος των κλειδιώματων/εξελεκτρολήματων έτσι ώστε αυτά να μην δημιουργούν πρόβλημα. Μάλιστα χρησιμοποιούνται την τεχνική του αστόδοξου κλειδιώματος (optimistic lock) μεκόνιστα αοιδά περισσότερο το overhead του μηχανισμού.

Το πλήθος και το μέγεθος των ET επηρεάζει τη συγνώμη των κλειδιώματων και εξελεκτρολήματων. Προφανώς όσο πιο μεγάλα είναι οι μέγεθος των ET και όσο πιο κακή επιδόση μαζί να είναι η συγνώμη τουτώ. Πειραματικά βρέθηκε ότι 20 ET με μέγεθος 200 παρέτων μπορούν και δουλεύουν ομαλά.

Συγχρονισμός όταν δεν υπάρχουν ET

Όταν δεν υπάρχουν ET τα πράγματα είναι πολύ πιο δύσκολα μας και κάθε εισέρχομενο πακέτο πρέπει να απευθείας να μπει στην ουρά αποστολήςς. Οπότε θεωρητικά έχουμε ένα κλειδίωμα και εξελεκτρολήματα της ουράς αποστολήςς αναπαραγώγοντας. Κάτι τέτοιου όμως είναι πάρα πολύ δύσκολο να δουλέψει σε υψηλή χαράτζη. Οπότε διαλέγονται ένα διαφορετικό τρόπο συγχρονισμού, πολύ πιο αποδοτικό.

Ο μηχανισμός συγχρονισμού σε αυτή την περίπτωση απορρέει πλήρως τα κλειδιώματα/εξελεκτρολήματα βασιζόμενος στην παρακάτω παρατήρησης:

- Κάθε πακέτο έχει πάρει μία αρίθμηση από 0 έως 15 από την Μηχανή Κατάστασης Λήψης.
ΣΧΗΜΑ 5.12: Προσθήκη στην ουρά αποστολής χωρίς ΕΤ.

- Έχουμε οκτώ ίνες λήψης που λειτουργούν με αυτοτηρή σειρά.

Οπότε, σε κάθε χρονική στιγμή, μέσα στον IXP μπορούν να υπάρχουν το πολύ οκτώ πακέτα. Αυτά τα πακέτα θα αρίθμούνται συνεχόμενα από το 0 έως το 15. Αρα, αν υπάρχουν οκτώ διαφορετικές ουρές αποστολής, κάθε πακέτο θα πάει αναγκαστικά σε διαφορετική ουρά, αφετέρου να θεωρήσει το υπόλοιπο της διαίρεσης του αριθμού του πακέτου με το οκτά. Αυτό γίνεται στην Εικόνα 5.12. Οπότε κάθε ίνα βάζει το πακέτο της απευθείας στην αντίστοιχη ουρά χωρίς να χρειάζεται να την κλειδώσει από τις άλλες ίνες λήψης.

Ο Ελεγκτής Αποστολής όμως χρειάζεται να διαφοροποιηθεί μιας και τάρα έχουμε οκτώ ουρές αποστολής αντί για μία. Η αλληλή όμως είναι πολύ απλή, και αυτό που χρειάζεται ο Ελεγκτής Αποστολής είναι κάθε φορά να παίρνει το επόμενο πακέτο προς αποστολή από την επόμενη ουρά.

5.2 Πειράματα

Σε αυτή την παράγραφο παρουσιάζουμε τα αποτελέσματα της αξιολόγησης του IXP-Διαμορφώσης. Εξετάζουμε αρχικά ξεκινώντας την απόδοση του τμήματος λήψης και του τμήματος αποστολής και κατόπιν αλλάξει θετικού του συστήματος. Στην συνέχεια, συγκρίνουμε την επίδραση των ΕΤ σε ενα ΣΑΕ με δύο ανενεργές και τον ενεργό διαμορφώση. Εν συντομία, ο ενεργός διαμορφώσης μπορεί να διαμορφώσει οποιαδήποτε κίνηση εργάζεται πάνω από έναν σύνδεσμο του 1 Gbit/s. Επιπλέον, τα αποτελέσματα του προηγούμενου κεφα-
5.2. ΠΕΙΡΑΜΑΤΑ

λαίου επαληθεύονται και υπάρχει ένα κέρδος περίπου 10% από την χρήση των ET.

5.2.1 Απόδοση Συστήματος Αποστολής

Σε αυτό το πείραμα βάλαμε τον IXP στο πρότζεκτ το ξεκινά και ξεκίνησε μετά το γρήγορα μπορούν να στείλουν πακέτα ας εννέα αποστολής. Ο τρόπος που το υλοποιήσαμε αυτό είναι με το να βάλουμε στην ωρά αποστολής δύο πακέτα που όμως δεν αφαιρούντα ποτέ πραγματικά.

Για μεγάλα πακέτα το σύστημα αποστολής μειώνει εύκολα τον Gbit σώνδεση πετυχαίνοντας περίπου 980 Mbit/s. Για κατά λίγο μεγαλύτερη (δηλαδή 630 Mbit/s) αλλά λίγο μικρότερη, 500 Mbit/s. Ο λόγος οφείλεται στο ότι κάθε ίνα αποστολής αναλαμβάνει ένα πακέτο μικρό και έτσι το μεγάλο overhead είναι η αραία της πακέτου από την ωρά αποστολής που την κάνει μόνο μία ίνα, ο Ελεγκτής Αποστολής. Εκεί οφείλεται και η πτώση στην απόδοση.

Παραλαβτά, ο κώδικας αποστολής μπορεί να χρησιμοποιηθεί για πειραματικές δοκιμές άλλων συστημάτων, μιας και εύκολα μπορεί να παράγει κίνηση πάρα πολύ υψηλής ταχύτητας.

5.2.2 Απόδοση Συστήματος Λήψης

Το σύστημα λήψης δεν είχε απολύτως κανένα πρόβλημα στην απόδοση. Μπορεί εύκολα να παραλαμβάνει πακέτα οποιουδήποτε μεγέθους σε μέγιστη ταχύτητα. Το σύστημα αυτό μπορεί να χρησιμοποιηθεί ως βάση για την καταχείριση άλλων συστημάτων παρακολούθησης δικτύων για συνδέσμους Gbit.

5.2.3 Απόδοση του Διαμοιραστή

Χρησιμοποιήσαμε δύο ανηχεντές, αρχικά για να πειραματιστούμε πάνω στο πόσο γρήγορα μπορεί να δουλέψει ο διαμοιραστής. Δεδομένου ότι ένας κοινός υπολογιστής δεν μπορεί να στείλει πακέτα οποιουδήποτε μεγέθους με την μέγιστη ταχύτητα πάνω από έναν σύνδεσμο ενός Gbit, βάλαμε έναν άλλο IXP να κάνει αυτή τη δουλειά.

Γενικά, ο διαμοιραστής δεν παρουσίασε καμία απώλεια σε πακέτα, όμως ενδιαφέρον παρουσιάζουν τα παρακάτω γεγονότα που εντοπίσαμε:

Παρενέργειες του Διαύλου διπλής κατεύθυνσης

Δεδομένου ότι ο δίαυλος IX είναι διπλής κατεύθυνσης, χρειάζεται μία μικρή καθυστέρηση ωστόσο να αλλάξει η ροή των δεδομένων μεταξύ αποστολής και
Μηνύματα Flow Control από αργούς ανιχνευτές

Κάτι άλλο που παρατηρήσαμε είναι το εξής: Στέλναμε στον διαμορφασμένο χώρο των 600 Mbit/s και τότε η μορφή του χώρου ήταν 300 Mbit/s. Αν ένας αργός ανιχνευτής εμφανίζει σημαντικά χωρίς να στέλνει μηνύματα παώσης (pause flow control message).

Αν η πόρτα από τη δεξί από τον ΙXP στέλνει παώση δεν έχει ρυθμιστεί κατάλληλα, τότε θα παύει υπόψη της μηνύματα παώσης και όντως θα σταματά και μήπως και τα δύο ροές για τους δύο ανιχνευτες. Ανάλογα ο ένας αργός ανιχνευτής επιφέρει πάλι το συστήμα. Είναι φυσικό πολύ, να δημιουργηθούν χαμένα παώση πάνω στον διαμορφασμένο χώρο από τα δύο φορές παρά παρά παίρνει.

Η λύση στο πρόβλημα αυτό είναι να ρυθμιστεί η πόρτα εξόδου του ΙXP ώστε να ανασχεδιάσει αυτά τα παώση παώσης.

5.2.4 Αξιολόγηση των ET

Περιβάλλον Πειράματος

Χρησιμοποιήσαμε την τοπολογία δικτύου που εμφανίζεται στην Εικόνα 5.1 στην σελίδα 48. Η κίνηση εργάτων από τον AMD Server οποίος διάβαζε το δικτύο αρχείων πάνω στην κρατούσε πάνω στο προηγούμενο κεφάλαιο. Ο AMD Server εκέβαλε δύο επεξεργαστές στα 2 GHz με 512 KB cache και 512 MB χώρος μνήμης. Είναι εξοπλισμένος με μία Gbit κάρτα δικτύου του 64 bit και SCSI συλληφό δίσκο. Το λειτουργικό του σύστημα είναι Redhat Linux 9 με τον τυρίνα 2.4.20.

Κάθε ανιχνευτής είναι ένας Pentium 4 στα 2 GHz με 512 KB χρωφής μνήμης και 1 GB χώρας μνήμης. Χρησιμοποιούσε το Snort 2 final, σε λειτουργικό σύστημα Linux Redhat 8 με τυρίνα 2.4.20.

Ο ΙXP-Διαμορφασμένος έχει 21 ενταμειντές τοποκάτησης, πέντε γκειρομένους σε συγκεκριμένη κίνηση και 16 γενικού σκοπού. Κάθε ET μπορεί να χωρέσει
Πίνακας 5.1: Μετρήσεις του ΙΧΡ-Διαμορφωτή

<table>
<thead>
<tr>
<th></th>
<th>Αν.1</th>
<th></th>
<th>Αν.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>μέθοδος</td>
<td>πωκ.</td>
<td>alerts</td>
<td>χρήστη</td>
</tr>
<tr>
<td>χωρίς LBs</td>
<td>1287k</td>
<td>9303</td>
<td>10.6</td>
</tr>
<tr>
<td>με LBs</td>
<td>1286k</td>
<td>9281</td>
<td>9.4</td>
</tr>
<tr>
<td>διαφ.</td>
<td>0.02%</td>
<td>0.23%</td>
<td>11.26%</td>
</tr>
</tbody>
</table>

περίπου 300 πακέτα των 512 Β καθένα.

Το χρονικό ρετλιάζοντας τη μαξιμπ καθεστώτας του ΑΜΔ σε περίπτωση 330 Mbit/s βραδύτερα την τάση της 8ης της ΙΕΠ Σελίτερ, όταν ο καθένας οδηγός της 8ης της ΙΕΠ Σελίτερ, όταν ο καθένας οδηγός της 8ης της ΙΕΠ Σελίτερ, όταν ο καθένας οδηγός της 8ης της ΙΕΠ Σελίτερ, όταν ο καθένας οδηγός της 8ης της ΙΕΠ Σελίτερ, όταν ο καθένας οδηγός της 8ης της ΙΕΠ Σελίτερ, όταν ο καθένας οδηγός της 8ης της ΙΕΠ Σελίτερ, όταν ο καθένας οδηγός της 8ης της ΙΕΠ Σελίτερ, όταν ο καθένας οδηγός της 8ης της ΙΕΠ Σελίτερ, όταν ο καθένας οδηγός της 8ης της ΙΕΠ Σελίτερ, όταν ο καθένας οδηγός της 8ης της ΙΕΠ Σελίτερ, όταν ο καθέ

Δυστυχώς, με τον ΑΜΔ Server δεν μπορούμε να διαμορφώσουμε κάνονα παρατάσσοντα από 330 Mbit/s. Ο λόγος είναι ότι η κίνηση έπειτα να διαβάζεται από τον δίσκο. Παράλληλα, τα περίπτωση δείχνουν κατά πόσο μπορούν να οφειλθούν οι εντάσεις τοποκότητας.

Αποτελέσματα

Στον Πίνακα 5.1 βλέπουμε τις μετρήσεις που πήραμε για τους δύο ανιχνευτές με την χρήση ET και χωρίς αυτούς.

Παρατηρούμε ότι και στις δύο περιπτώσεις το πλάθος των πακέτων που επεξεργάζονταν και οι δύο ανιχνευτές είναι το ίδιο, οπότε επίσης είναι ιδίο και το πλάθος των alerts. Μεταξύ των ανιχνευτών υπάρχει μία αποκλίση κατά 6.5% από το δίσκο μερίδα του σαβαγματο οφειλόμενη στην συνάρτηση hashing που χρησιμοποιήσαμε.

Παράλληλα, η επίδραση των ET είναι φανερή. Παρατηρούμε σημαντική μείωση στον χρόνο χρήσης και στον χρόνο συστήματος και για τους δύο ανιχνευτές. Για τους ανιχνευτές 1, ο χρόνος χρήσης βελτιώνεται κατά 11.26% ενώ ο χρόνος συστήματος κατά 29.1%. Ο δεύτερος ανιχνευτής οφείλεται κατά 14.71% στο χρόνο χρήσης και κατά 12.83% στο χρόνο συστήματος. Συνολικά ο χρόνος χρήσης πέφτει από 22.99 δ.εκ. σε 20.04 δ.εκ. δηλαδή κατά 12.8%. Τα αποτελέσματα ήταν αναμενόμενα βάσει του προγράμματος κεφάλαιου και εδώ απλά τα επιβεβαιώσαμε.
5.3 Περίληψη

Σε αυτό το κεφάλαιο παρουσιάζουμε την υλοποίηση ενός ενεργού διαμοιραστή που υποστηρίζει ενταμιευτές τοπικότητας με τον επεξεργαστή δικτύου IXP-1200. Η υλοποίηση του ενεργού διαμοιραστή βασίζεται στις μεθόδους που διευρενώσαμε στο Κεφάλαιο 4. Τα πειράματα έδειξαν ότι ο IXP μπορεί να υλοποιήσει ένα τέτοιο διαμοιραστή για συνδέσμους του 1 Gbit. Επιπλέον, επαληθεύσαμε τα αποτελέσματα του κεφαλαίου 4 ότι οι ET μπορούν να προσφέρουν πάνω από 10% βελτίωση στην απόδοση του ΣΑΕ.
Κεφάλαιο 6

Συμπεράσματα

Συνοψίζοντας, σε αυτή την εργασία ασχολήθηκαμε με δύο θέματα που αφορούν την τεχνολογία και την ασφάλεια των δικτύων. Την ανάλυση επικεφαλίδων πακέτων και τον διαμαρσαμό της διαστυσαθής κίνησης σε πολλαπλούς ανυγευτές εισβολέοι.

Στο πρώτο θέμα μελετήσαμε πως μπορούμε να χρησιμοποιήσουμε τον Επεξεργαστή Δικτύων IXP 1200 στην ανάλυση επικεφαλίδων πακέτων. Αναπτύξαμε μάλιστα ένα κατάλληλο περιβάλλον ως το οποίο ο προγραμματισμός του IXP 1200 να μπορεί να γίνει με πολύ απλό τρόπο. Δηλαδή με την χρήση περιγραφών σε υψηλού επιπέδου γλώσσα (που υπάρχουν έτοιμες στο διαδίκτυο) και με το εργαλείο S21 το οποίο θα επισημαίνεται σε χαμηλό επίπεδο κώδικα. Οπότε εδώ, η δύναμη του IXP 1200 είναι αρκετή για να ελέγχει ένα μικρό σύνολο περιγραφών σε ταχύτερες συνδέσμους Fast Ethernet. Μια πιθανή μελλοντική εξέλιξη αυτής της εργασίας είναι η μελέτη συνδέσμων Gbit.

Είδαμε επίσης, ότι το εργαλείο S21 μπορεί με λίγες αλλαγές να παράγει κώδικα και για άλλες αρχιτεκτονικές. Οπότε σε αυτή την εργασία μπορούν να βασιστούν μελέτες για ανίχνευση επικεφαλίδων πακέτων για πιο γρήγορα δίκτυα που να βασίζονται σε άλλους IXP, ή ασόμα και τελείως διαφορετικούς επεξεργαστές.

Το δεύτερο θέμα της παρούσας εργασίας αφορά τον διαμαρσαμό της διαστυσαθής κίνησης σε πολλαπλούς ανυγευτές εισβολέοι. Η βασική συνεισφορά αυτής της εργασίας σώζει τον τομέα είναι διπλή. Καταρχήν δείχνει ότι τα συστήματα ανίχνευσης εισβολέων μπορούν σημαντικά να επιδερμίζονται από αυξημένη τοποκάτηνα αναφοράς. Αυτό το δείχνει ότι την τεχνική των εντομούντων τοποκάτηνας, η οποία αυξάνει την τοποκάτηνα των αναφορών για κάθε ανίχνευτη εισβολέων και όπως είδαμε αυξάνει και την απόδοσή του. Κατόπιν είδαμε ότι η τεχνική αυτή μαζί με την τεχνική του φίλτραρίσματος σε χαμηλό επίπεδο μπορούν να βελτιώσουν μέχρι και 20% την συνολική απόδοση ενός συστήματος ανίχνευσης εισβολέων όταν υλοποιηθούν απευθείας από τον διαμαρσαμό.
Δείξαμε μάλιστα ότι ένας παρόμοιος διαμοιρασμός είναι εφικτό να υλοποιηθεί με τον IXP 1200.

Και απο τα δύο θέματα που μας απασχόλησαν αποκτήσαμε βαθύτατη γνώσης της αρχιτεκτονικής των Επεξεργαστών Δικτύων IXP. Είδαμε ότι ο προγραμματισμός τους δεν είναι απλή υπόθεση. Αποτελούν όμως πολύ ισχυρές και εύπλαστες μονάδες επεξεργασίας που με κατάλληλη χρήση μπορούν να υλοποιήσουν με φθηνό τρόπο αρκετά απαιτητικά συστήματα.
Bibliography


