

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΕΡΓΑΣΤΗΡΙΟ ΣΧΕΔΙΑΣΜΟΥ ΟΛΟΚΛΗΡΩΜΕΝΩΝ
ΚΥΚΛΩΜΑΤΩΝ ΜΕΓΑΛΗΣ ΚΑΙΜΑΚΑΣ (VLSI DESIGN)**

**Υλοποίηση σε Υλικό του Πρωτοκόλλου
Mobile IPv6 - Home Agent**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΜΕΣΣΗΝΗΣ ΘΕΟΔΩΡΟΣ του ΓΕΩΡΓΙΟΥ

**ΦΟΙΤΗΤΗ ΤΟΥ ΤΜΗΜΑΤΟΣ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΕΠΙΒΛΕΠΩΝ: ΚΑΘΗΓΗΤΗΣ
ΚΟΥΦΟΠΑΥΛΟΥ ΟΔΥΣΣΕΑΣ**

**ΑΡΙΘΜΟΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ: /2005
ΙΟΥΛΙΟΣ 2005**

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ	4
-----------------	----------

ΚΕΦΑΛΑΙΟ 1 - MOBILE IPV6: ΕΙΣΑΓΩΓΗ, ΟΡΟΛΟΓΙΑ

1.1. Γενικά για το πρωτόκολλο	5
1.2. Τι είναι κινητικότητα;	5
1.3. Οι απαιτήσεις μας από το Mobile IPv6	7
1.4. Η ορολογία του Mobile IPv6	8
1.5. Βασικές λειτουργίες του Mobile IPv6 στην πλευρά του Home Agent	9

ΚΕΦΑΛΑΙΟ 2 – ΤΟ ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΟ ΣΥΣΤΗΜΑ MOBILE IPV6

2.1. Βελτιώσεις του IPv6 σε σχέση με το IPv4	10
2.2. IPv6	11
2.2.1. Απεικόνιση IPv6 διευθύνσεων	11
2.2.2. Είδη IPv6 Διευθύνσεων	12
2.3. Είδη πακέτων στο Mobile IPv6	15
2.3.1. Binding Update	15
2.3.2. Binding Acknowledgement	17
2.3.3. Binding Request	19
2.3.4. Router Advertisement	19
2.3.5. Router Solicitation	21
2.4. Διαδικασία απόκτησης Care-of-Address	22
2.4.1. Stateful Address Autoconfiguration	22
2.4.2. Stateless Address Autoconfiguration	22
2.5. Ingress Filtering	23
2.6. Tunneling	23

ΚΕΦΑΛΑΙΟ 3 – Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

3.1. Εισαγωγή	26
3.2. Η Αρχιτεκτονική σε Επίπεδα Ιεραρχίας	27
3.3. Η Εκτέλεση των έργων του Συστήματος σε βήματα	30
3.3.1. Εισαγωγή πακέτων από το Data Link Layer	30
3.3.2. Binding Update without Acknowledgement	31
3.3.3. Binding Update with Acknowledgement	33
3.3.4. Binding Request	35

3.3.5. Router Advertisement	36
3.3.6. Packet Encapsulation	38
3.3.7. IPv6 Packet	40
3.3.8. IPv4 Packet	41
3.4. Η Ανάλυση των Δομικών Μονάδων του Συστήματος	42
3.4.1. Memory Management	42
3.4.2. Packet Recognition	48
3.4.3. Control Unit	53
3.4.4 Content Addressed Memory	58
3.4.5. Binding Watchdog	58
3.4.6. Packet Builder	59
3.4.7. Main Memory	61
3.4.8. IPv6 Unit	61
3.4.9 IPv4 Unit	62

ΚΕΦΑΛΑΙΟ 4 – Η ΣΥΝΘΕΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

4.1. Εισαγωγή	64
4.2. Τα Σχηματικά και τα Κυκλώματα	64
4.2.1. Control Memory	64
4.2.2. Packet Recognition	65
4.2.3. Control Unit	66
4.2.4. Content Addressed Memory	67
4.2.5. Binding Watchdog	67
4.2.6. Packet Builder	68
4.2.7. RecCamBuildWatch	69
4.2.8. Control	70
4.2.9. System	71
4.3. Οι Χρονισμοί και η Απόδοση του Συστήματος σε FPGA	73
4.3.1. Control Memory	73
4.3.2. Control Unit	74
4.3.3. Control	75
4.3.4. Packet Builder	76
4.3.5. Content Addressed Memory	78
4.3.6. Binding Watchdog	80
4.3.7. Packet Recognition	81
4.3.8. IPv4 Unit	82
4.3.9. IPv6 Unit	83
4.3.10 RecCamBuildWatch	84
4.3.11 System	92

ΚΕΦΑΛΑΙΟ 5 – ΕΞΟΜΟΙΩΣΕΙΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

5.1. Binding Acknowledgement	104
5.2. Binding Request	106
5.3. Router Advertisement	108
5.4. Packet Encapsulation	110
5.5. Packet Builder	113
5.6. Content Addressed Memory	115
5.6.1. Cam 1	115
5.6.2. Cam 2	116

ΚΕΦΑΛΑΙΟ 6 – ΣΥΜΠΕΡΑΣΜΑΤΑ ΒΕΛΤΙΩΣΕΙΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

6.1. Βελτιώσεις της παρούσας Υλοποίησης	119
6.2. Βελτιώσεις του Πρωτοκόλλου Mobile IPv6	119
6.2.1. Γρήγορα Handovers στο Mobile IPv6	120
6.2.1.1 Αναμονή Μετακίνησης και Ενεργοποίηση Handover	120
6.2.1.2 Ενημέρωση Access Router	121
6.2.2. Το κόστος Αναμονής Μετακίνησης	122

ΒΙΒΛΙΟΓΡΑΦΙΑ

Βιβλία και Άρθρα	123
Αναφορές	123
Πηγές από το Διαδίκτυο	123

Εισαγωγή

Στην παρούσα διπλωματική εργασία, γίνεται μελέτη του τηλεπικοινωνιακού πρωτοκόλλου Mobile IPv6. Το συγκεκριμένο πρωτόκολλο πρόκειται για επέκταση του IPv6 με σκοπό να καταστεί δυνατή η κίνηση κόμβων ενώ βρίσκονται συνδεδεμένοι στο διαδίκτυο, χωρίς να απαιτείται ο τερματισμός και η επανεκκίνηση των διαδικασιών επικοινωνίας που διεξάγονται. Το πρωτόκολλο επιδιώκει να καταστήσει την κινητικότητα (mobility) διαφανής υπηρεσία στα υψηλότερα επίπεδα.

Η βιβλιογραφία που υπάρχει για το συγκεκριμένο θέμα είναι μόνο στην Αγγλική γλώσσα, έγινε προσπάθεια όσο το δυνατόν καλύτερης μετάφρασης, ωστόσο σε πολλά σημεία διατηρήθηκε η Αγγλική ορολογία για να αποφύγουμε τον κίνδυνο λανθασμένης μετάφρασης.

Στο 1^ο κεφάλαιο αναφέρονται τα αρχικά του πρωτοκόλλου και η ορολογία την οποία χρησιμοποιούμε στην συνέχεια.

Στο 2^ο κεφάλαιο γίνεται εκτεταμένη αναφορά στις λεπτομέρειες του πρωτοκόλλου. Αναφέρονται λεπτομέρειες για τις διευθύνσεις IPv6, για τα μηνύματα τα οποία ανταλλάσσονται στα πλαίσια του πρωτοκόλλου και άλλες έννοιες οι οποίες είναι απαραίτητες για να κατανοήσουμε το Mobile IPv6.

Στο 3^ο κεφάλαιο παρουσιάζεται η αρχιτεκτονική του συστήματος το οποίο υλοποιήθηκε με σχηματική ανάλυση των έργων σε βήματα και της λειτουργίας κάθε μονάδας ξεχωριστά.

Στο 4^ο κεφάλαιο παρουσιάζεται η σύνθεση του συστήματος. Κάθε υπομονάδα όπως την συνέθεσε το πρόγραμμα Leonardo Spectrum (Level 3) και οι χρονισμοί που επιτεύχθηκαν με χρήση συγκεκριμένων FPGA.

Στο κεφάλαιο 5^ο παρουσιάζονται χαρακτηριστικές εξομοιώσεις του συστήματος.

Στο κεφάλαιο 6^ο παρουσιάζονται βελτιώσεις του συστήματος για βελτίωση της απόδοσης και βελτιώσεις του πρωτοκόλλου για πιο γρήγορα handover.

Στο παράρτημα υπάρχει ο πηγαίος κώδικας ο οποίος υλοποιεί το σύστημα και στο τέλος της εργασίας υπάρχει η βιβλιογραφία και μια μικρή περίληψη.

Κεφάλαιο 1^ο

Mobile IPv6: Εισαγωγή - Ορολογία

1.1 Γενικά για το πρωτόκολλο

Το Mobile IPv6 είναι ένα πρωτόκολλο του επίπεδου δικτύου το οποίο κάνει δυνατή την κινητικότητα σε δίκτυα βασισμένα σε IPv6. Η τεχνολογία της κινητικότητας έχει αναπτυχθεί πάρα πολύ τα τελευταία χρόνια και κάποιοι παράγοντες για αυτή την ανάπτυξη είναι η εξάπλωση των 3G δικτύων τα οποία υποστηρίζουν υπηρεσίες βασισμένες στην διαμεταγωγή πακέτων, η χρήση των δικτύων υψηλών ταχυτήτων UMTS, cdma2000 και GPRS/ EDGE, οι αυξανόμενοι αριθμοί πωλήσεων notebook υπολογιστών, οι οποίοι ξεπέρασαν τον αριθμό των desktop στο πρώτο μισό του 2003 και η διαθεσιμότητα του 802.11 πρωτοκόλλου που πλέον συναντάτε σε πλήθος εφαρμογών από μικρά τοπικά δίκτυα μέχρι δίκτυα δημοσίων χώρων όπως ξενοδοχεία, αεροδρόμια.

Το πρωτόκολλο IPv6 έχει αρχίσει να αναπτύσσεται σε κάποιες περιοχές του κόσμου και η μελλοντική έλλειψη στις διευθύνσεις που μπορεί να παρέχει το IPv4 θα επιταχύνει ακόμα περισσότερο την διάδοσή του στις αναπτυσσόμενες και στις αναπτυγμένες χώρες. Η κινητικότητα για τις IP συσκευές δεν αποτελούσε παλιότερα κάποιο θέμα που απασχολούσε τον κόσμο της τεχνολογίας, όμως στις μέρες μας με την ενσωμάτωση IP stacks στα PDA, κινητά τηλέφωνα και πολλά είδη notebook η κινητικότητα έχει προαχθεί σε ένα κρίσιμο ζήτημα.

Το Mobile IP αποτελεί ένα εσωτερικό μέρος της στοίβας του IPv6 και εκτιμάτε ότι θα υπάρχει υλοποιημένο σε κάθε IPv6 συσκευή στο μέλλον. Με αυτή την αλλαγή θα περάσουμε από το νομαδικό μοντέλο του χρήστη σε ένα μοντέλο που δεν είναι τόσο εμφανές στην φύση.

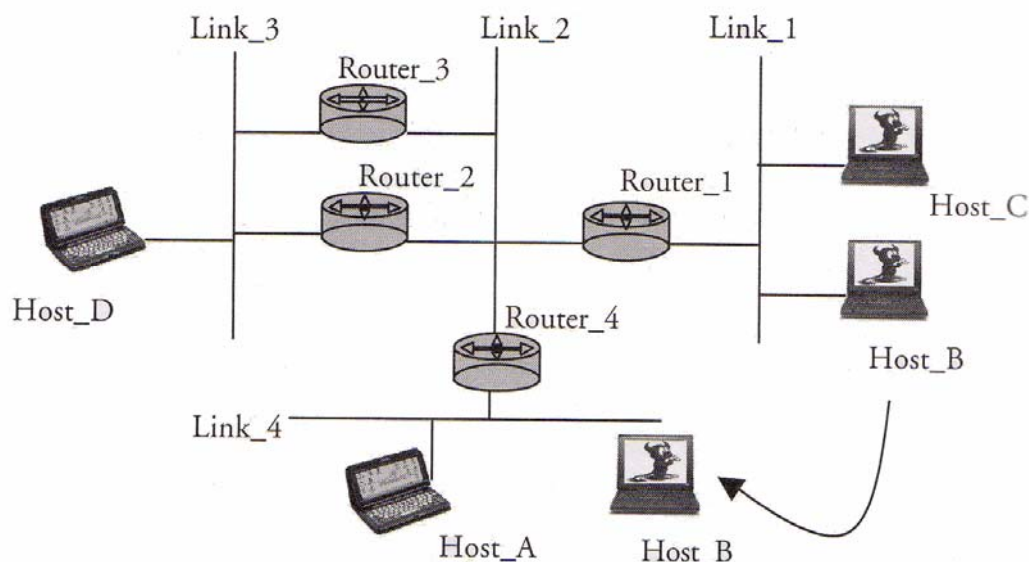
Το πρωτόκολλο Mobile IPv6 έχει μια μακρά ιστορία στην IETF. Στα μέσα του 1990 άρχισε να αναπτύσσεται και εγκρίθηκε για έκδοση ως RFC μόλις τον Ιούλιο του 2003. Πολύ δουλειά έχει γίνει κατά την διάρκεια αυτών των χρόνων ως προς τα specifications του πρωτοκόλλου. Επίσης το πρωτόκολλο εξελίχθηκε σημαντικά από το 2001 όπου ανακαλύφθηκαν κάποιες ελλείψεις ως αναφορά την ασφάλεια για την βελτιστοποίηση διαδρομής.

1.2 Τι είναι η κινητικότητα;

Η κινητικότητα είναι η αλλαγή της IP διεύθυνσης, μιας κινητής συσκευής λόγω της αλλαγής του σημείου σύνδεσης της συσκευής με την υπόλοιπη τοπολογία του διαδικτύου. Αυτή η αλλαγή μπορεί να προέλθει από φυσική κίνηση, όπως όταν κάποιος μεταφέρει έναν υπολογιστή από ένα δωμάτιο σε ένα άλλο ή όταν κάποιος βρίσκεται μέσα σε κάποιο κινούμενο όχημα το οποίο εξυπηρετείται από διαφορετικές ζεύξεις ανάλογα με την θέση που βρίσκεται (π.χ μέσα σε ένα κινούμενο αεροπλάνο).

Η κινητικότητα μπορεί να συμβεί εξαιτίας μιας αλλαγής στην τοπολογία, η οποία θα προκαλέσει την αλλαγή της διεύθυνσης ενός node χωρίς την φυσική μεταφορά.

Παρακάτω παρουσιάζεται ένα σενάριο στο οποίο μπορούμε να αντιληφθούμε καλύτερα τις παραπάνω έννοιες.



Ας υποθέσουμε ότι ο ενώ ο Host_B επικοινωνεί με το Host_D αποφασίζει να μεταφερθεί από το link_1 στο link_4. Από αυτή την μετακίνηση προκύπτουν 3 διαφορετικά προβλήματα:

- Ο Host_D δεν είναι ενήμερος για αυτή την αλλαγή της θέσης του Host_B με τον οποίο επικοινωνεί με αποτέλεσμα να συνεχίσει να στέλνει πακέτα το Host_B σαν να ήταν στο link_1. Αυτά τα πακέτα προφανώς δεν θα φθάσουν στον Host_B και θα απορριφθούν.
- Η διεύθυνση του Host_B δεν είναι πλέον έγκυρη γιατί δεν συμφωνεί με το prefix της διεύθυνσης που υπάρχει στο link_4 αλλά βασίζεται σε ένα prefix το οποίο αντιστοιχεί σε nodes που είναι συνδεδεμένοι στο link_1. Συνεπώς για να μπορέσει να επικοινωνήσει θα πρέπει να αποκτήσει μια νέα διεύθυνση και μέχρι να την αποκτήσει δεν θα μπορεί να παραλάβει κανένα πακέτο από αυτά τα οποία στέλνονται στην παλιά του διεύθυνση.
- Από την στιγμή που αποκτήσει ο mobile node (Host_B) την καινούργια διεύθυνση με βάση το prefix του link_4 θα πρέπει να αλλάξει το socket το οποίο περιέχει την πληροφορία για την επικοινωνία με τον χρήστη Host_D όμως αυτό δεν είναι δυνατό γιατί και οι δυο χρήστες θα πρέπει να διατηρήσουν την ίδια πληροφορία για κάθε σύνδεση. Αυτό θα οδηγήσει σε τερματισμό της επικοινωνία από την στιγμή που οι δυο peers δεν θα μπορούν να επικοινωνήσουν.

Τα παραπάνω προβλήματα προκύψαν επειδή έγινε αλλαγή στην IP διεύθυνση ενός host ενώ είχε ενεργές συνδέσεις με άλλους hosts. Αν ο τρόπος με τον οποίο γίνεται η δρομολόγηση των πακέτων στα δίκτυα ή τα sockets ήταν ορισμένα με κάποιο διαφορετικό τρόπο που δεν περιελάμβανε την IP διεύθυνση τα παραπάνω προβλήματα δεν θα υπήρχαν. Αυτό το πρόβλημα λύνει το Mobile IPv6 με μια προσθήκη: **επιτρέπει στους κινητούς χρήστες να είναι προσβάσιμοι προσφέροντας τους μια σταθερή διεύθυνση IP.**

1.3 Ποιές είναι οι απαιτήσεις μας από το Mobile IPv6;

- **Η αδιάλειπτη λειτουργία των συνδέσεων:** είναι σημαντικό να διασφαλίσουμε ότι οι συνδέσεις δεν θα σταματήσουν όταν ένα κινητός χρήστης αλλάξει την διεύθυνσή του – όπως δεν σταματούν οι συνδέσεις στα κινητά τηλέφωνα όταν ο χρήστης αλλάζει τόπο.
- **Η διαθεσιμότητα:** δεν υπάρχει χρησιμότητα στο να μπορεί ένας χρήστης να είναι κινητός χωρίς παράλληλα να είναι διαθέσιμος. Με βάση την λειτουργία του DNS αυτό μπορεί να συμβεί αν διατηρήσει σταθερή IP διεύθυνση.
- **Η συμβατότητα με όλες τις εφαρμογές:** αυτό συμβαίνει καθώς το Mobile IP είναι διαφανές στα παραπάνω στρώματα από το επίπεδο δικτύου(μεταφοράς, εφαρμογών).
- **Η ανεξαρτησία των χαμηλότερων επιπέδων:** μπορεί να χρησιμοποιηθεί από όλα τα πρωτόκολλα επιπέδου data link.
- **Σηματοδότηση από άκρο-σε-άκρο:** Όλοι οι κόμβοι στο διαδίκτυο μπορούν να διαιρεθούν σε δυο κατηγορίες, στους hosts και τους routers. Οι routers, δρομολογούν τα πακέτα με βάση την διεύθυνση προορισμού. Οι hosts είναι υπεύθυνοι για να διατηρήσουν τις συνδέσεις, τον έλεγχο ροής μεταξύ τους, κτλ. Ο διαχωρισμός των ρόλων, ενσωματώνοντας την ευφυΐα στους hosts και αφήνοντας το δίκτυο να κάνει την διαμεταγωγή των πακέτων έχει συμβάλει στην σημερινή εξάπλωση του διαδικτύου. Το Mobile IP ως πρωτόκολλο συμβαδίζει με αυτό το μοντέλο καθώς η εφαρμογή του δεν προϋποθέτει αλλαγή στην δρομολόγηση, ή αλλαγή κάποιων κόμβων αλλά αντίθετα την αλλαγή κάποιων λειτουργιών στα τερματικά σημεία της επικοινωνίας. Αυτό το χαρακτηριστικό δίνει στο πρωτόκολλο επεκτασιμότητα καθώς αν υπήρχαν συγκεκριμένα μηχανήματα διασκορπισμένα στο διαδίκτυο υπεύθυνα για την εφαρμογή του Mobile IP αυτά τα σημεία πολύ πιθανόν να αποτελούσαν bottlenecks σε περίπτωση αύξησης του αριθμού των χρηστών.

1.4 Mobile IPv6 Ορολογία

Mobile Node: είναι ένας κόμβος ο οποίος αλλάζει την θέση του μέσα στην τοπολογία του διαδικτύου. Η κινητικότητα του κόμβου μπορεί να είναι αποτέλεσμα φυσικής κίνησης ή αποτέλεσμα αλλαγής της τοπολογίας του δικτύου. Παραδείγματα φυσικής κίνησης είναι μια συσκευή η οποία αλλάζει links καθώς κινείται μέσα σε ένα όχημα (αεροπλάνο, αυτοκίνητο, τρένο). Ένα παράδειγμα αλλαγής τοπολογίας του δικτύου συμβαίνει όταν κάποιος δρομολογητής παθαίνει κάποια βλάβη, με αποτέλεσμα ο κόμβος να βρίσκεται συνδεδεμένος σε άλλο δρομολογητή. Ίσως πιο κατάλληλος όρος για τον κινητό χρήστη του πρωτοκόλλου θα ήταν το mobile host, ωστόσο ο όρος mobile node έχει επικρατήσει στην βιβλιογραφία.

Correspondent Node: είναι κάθε κόμβος ο οποίος επικοινωνεί με τον κινητό κόμβο. Οι όροι mobile και correspondent αναφέρονται σε συγκεκριμένες λειτουργίες του πρωτοκόλλου καθώς κάποιος host μπορεί να αντιμετωπίζεται ως correspondent node από τον mobile node αλλά θα μπορούσε και ο correspondent node να κινείται το οποίο θα τον έκανε mobile node και ενώ θα φαίνεται ως correspondent node από τον κινητό peer.

Home Address: μια σταθερή διεύθυνση η οποία ανήκει στο κινητό χρήστη και χρησιμοποιείται από τους correspondent nodes ώστε να επικοινωνούν με τον κινητό χρήστη. Όπως όλες οι IPv6 διευθύνσεις, η home address βασίζεται σε ένα 64bit prefix το οποίο έχει ανατεθεί στο home link του κινητού χρήστη.

Home Link: μια σύνδεση στην οποία έχει ανατεθεί το prefix της home address.

Home Agent: ένας δρομολογητής που βρίσκεται στο home link και λειτουργεί εκ μέρους του mobile node όταν τελευταίος βρίσκεται μακριά από το home link του. Η βασική αποστολή του home agent είναι να ανακατευθύνει τα πακέτα που φτάνουν με προορισμό την Home Address στην care-of-address στην οποία βρίσκεται ο κινητός χρήστης.

Foreign Link: οποιοδήποτε link από το οποίο περνάει ο κινητός χρήστης.

Care-of-address: είναι μια διεύθυνση που ανατίθεται στον κινητό χρήστη όταν αυτός επισκέπτεται κάποιο foreign link. Αυτή η διεύθυνση βασίζεται στο prefix το οποίο αντιστοιχεί σε εκείνο το link.

Binding: η σύνδεση του κινητού χρήστη (home address) με την care-of-address για μια συγκεκριμένη περίοδο χρόνου. Αυτό επιτρέπει στον home agent να προωθεί τα πακέτα προς την συγκεκριμένη τοποθεσία στην οποία βρίσκεται ο κινητός χρήστης κάθε φορά. Το binding ανανεώνεται κάθε φορά που αλλάζει ο κινητός χρήστης care-of-address ή μετά από συγκεκριμένα χρονικά διαστήματα ώστε να είναι πάντα έγκυρο.

1.5 Βασικές Λειτουργίες του Mobile IPv6 από την πλευρά του Home Agent

Οι βασικές λειτουργίες της οποίες θα πρέπει να εκπληρώνει ο home agent ώστε να υποστηρίζει το πρωτόκολλο mobile IPv6 είναι οι παρακάτω:

A) Να παρέχει Binding με το κινητό χρήστη. Αυτό σημαίνει ότι για κάθε κινητό χρήστη θα πρέπει να έχει καταχωρημένες τις παρακάτω πληροφορίες:

- Αν βρίσκεται στο Home Link ή σε κάποιο Foreign Link.
- Την Care-of-address κάθε κινητού χρήστη για κάθε δεδομένη στιγμή.
- Την παράμετρο Lifetime, η οποία καθορίζει τον χρόνο εγκυρότητας των στοιχείων που αποτελούν την Binding πληροφορία.

B) Να κατασκευάζει και να στέλνει Binding Request πακέτο στο κινητό χρήστη κάθε φορά που το Lifetime που αναφέρεται στο Binding με τον Home Agent λήγει.

Γ) Να ενημερώνει τις binding πληροφορίες του μετά από κάθε Binding Update πακέτο το οποίο έρχεται από τον κινητό χρήστη με σκοπό να ενημερώσει για την καινούργια θέση του. Σε περίπτωση που ο κινητός χρήστης στο Binding Update πακέτο ζητήσει να του αποσταλεί πίσω Binding Update Acknowledgement πακέτο ο Home Agent κατασκευάζει αυτό το πακέτο και το αποστέλλει στο Data Link Layer.

Δ) Να ενθυλακώνει τα πακέτα τα οποία προορίζονται για τον κινητό χρήστη ο οποίος βρίσκεται συνδεδεμένος σε κάποιο Foreign Link και να τα αποστέλλει στην διεύθυνση Care-of-address ώστε να μην διακόπτεται η in-going επικοινωνία με τον κινητό χρήστη αν και εκείνος μπορεί να αλλάζει σημείο πρόσβασης στο παγκόσμιο ιστό.

E) Να κατασκευάζει και να αποστέλλει ICMPv6 Router Advertisement κάθε φορά που δέχεται ICMPv6 Router Solicitation πακέτο για την εκπλήρωση της διαδικασίας Router Discovery με την οποία ο κινητός χρήστης καθορίζει την θέση του κάθε φορά που αλλάζει σημείο πρόσβασης.

Δ) Να παρέχει την διασύνδεση με τις μονάδες IPv6 και IPv4 οι οποίες αναλαμβάνουν την διαχείριση του πακέτου από την στιγμή που αναγνωρίζεται ότι ανήκει στην δικαιοδοσία τους.

Κεφάλαιο 2^ο

Mobile IPv6: Εισαγωγή - Ορολογία

Εισαγωγή

Στο παρόν κεφάλαιο θα αναλύσουμε την λειτουργία του πρωτοκόλλου. Θα γίνει αναφορά στα πλεονεκτήματα τα οποία έχει το IPv6 έναντι του IPv4. Η υλοποίηση του Mobile IP στα πλαίσια της παρούσας διπλωματικής εργασίας έγινε με βάση το IPv6. Θα πρέπει να τονίσουμε πως το πρωτόκολλο Mobile IP παρόλου που έχει αναπτυχθεί πολύ τα τελευταία χρόνια, περιλαμβάνει αρκετά θέματα εν εξελίξει. Η παρούσα εργασία βασίστηκε κυρίως στη δεύτερη έκδοση του αρχείου που αναφέρεται στο Mobile IPv6 [draft-ietf-mobileip-ipv6-02.txt] αλλά και σε μεταγενέστερες αναφορές που έχουν γίνει σχετικά με το θέμα.

2.1 Βελτιώσεις του IPv6 σε σχέση με το IPv4 που αφορούν την σχεδίαση του Mobile IPv6

Οι σημαντικότερες διαφορές του IPv6 σε σχέση με τον σχεδιασμό του Mobile IPv6 είναι:

1. Το IPv6 παρέχει μεγαλύτερο εύρος διεύθυνσεως καθορίζοντας την IP διεύθυνση από τα 32 bit στα 128 bit. Με την αλλαγή αυτή, ο κινητός χρήστης μπορεί πολύ πιο εύκολα να λαμβάνει μια διεύθυνση Care-of-address (απλοποιείται η διαδικασία Address Autoconfiguration) στο Foreign Link στο οποίο βρίσκεται κάθε φορά με αποτέλεσμα να εξαλείφεται η ανάγκη ύπαρξης Foreign Agent ο οποίος ήταν βασικός στην περίπτωση του Mobile IPv4.
2. Επιτρέπει την ενσωμάτωση σύγχρονων τεχνικών στην ασφάλεια (IPsec). Το IPsec δίνει την δυνατότητα στους κόμβους να προστατεύουν τις επικοινωνίες τους κάνοντας χρήση κρυπτογράφησης και αυθεντικοποίησης.
3. Η ύπαρξη της επικεφαλίδας Routing Header, επιτρέπει την δρομολόγηση πακέτων απευθείας στον κινητό χρήστη.
4. Η επικεφαλίδα Destination Option Header με την οποία υπάρχει δυνατότητα καθορισμού επιλογών στα πακέτα που αποστέλλονται χωρίς σημαντική μείωση στην ταχύτητα.

2.2.1 Απεικόνιση των IPv6 διευθύνσεων

Λόγω του μεγέθους των IPv6 διευθύνσεων, απεικονίζονται με δεκαεξαδικό τρόπο. Οι διευθύνσεις διαιρούνται σε 8 μέρη κάθε ένα από το οποίο περιέχει 2 οκτάδες bits (words), τα οποία διαχωρίζονται με το σύμβολο (:). Ακολουθούν κάποια παραδείγματα διευθύνσεων.

3ffe:0200:0008:72AB:1434:0:0:1 ή

3ffe:200:8:72AB:1434:0:0:1

Παρακάτω παρατηρούμε ότι όταν πολλές λέξεις (words) αποτελούνται από πολλά μηδενικά μπορούμε να συμπίεσουμε αυτές τις διευθύνσεις χρησιμοποιώντας το σύμβολο (::)

3ffe:0:0:5:104A:2A61:0:0 μπορεί να γραφεί

3ffe::5:104A:2A61:0:0 ή

3ffe:0:0:5:104A:2A61::

Ωστόσο η ίδια διεύθυνση δεν μπορεί να αναπαρασταθεί με τον παρακάτω τρόπο γιατί με αυτόν τον τρόπο δεν μπορούμε να συμπεράνουμε πόσες από τις λέξεις είναι 0 σε κάθε ένα από τα πεδία που έχει γίνει συμπίεση.

3ffe::5:104A:2A61::

Οι IPv6 διευθύνσεις γράφονται με τρόπο ώστε να περιγράφεται η διεύθυνση και το network prefix του link

Address / Prefix Length in bits

3ffe:0200:0008:72AB:1434:0:0:1 / 64 ή

3ffe:0200:0008:72AB:1434::1

2.2.2 Είδη IPv6 διευθύνσεων:

Τα είδη των IP διευθύνσεων όπως αυτά περιγράφονται στο RFC 1884 και RFC 1924 διακρίνονται στις παρακάτω κατηγορίες:

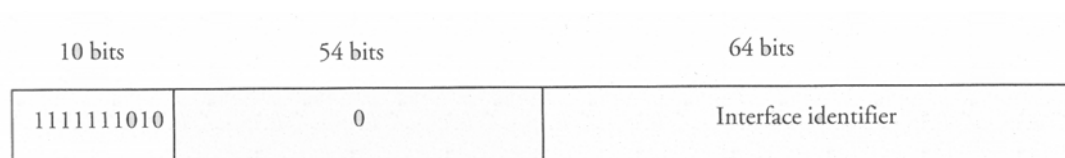
A) Unicast

Οι διευθύνσεις αυτές ανατίθενται σε κάποιο μοναδικό interface. Τυπικά σε κάθε συσκευή στο internet έχει ανατεθεί μια τέτοια διεύθυνση. Οι Unicast διευθύνσεις έχουν 4 διαφορετικά είδη:

- Loopback address
- Link-local
- Site-local
- Global

Η **loopback** διεύθυνση είναι ισχύουσα μόνο για κάθε κόμβο ξεχωριστά και σχηματίζεται θέτοντας τα 127 πιο σημαντικά bits στην τιμή 0 και το λιγότερο σημαντικό στην τιμή 1. Όπως αποκαλύπτει και το όνομά της επιτρέπει την επικοινωνία των εφαρμογών που τρέχουν στον ίδιο κόμβο. Αυτή η μέθοδος υπερτερεί έναντι οποιασδήποτε άλλης μεθόδου επικοινωνίας με χρήση κάποιας εσωτερικής συνάρτησης μεταξύ των εφαρμογών, γιατί δημιουργεί ανεξαρτησία του λογισμικού που “τρέχει” σε κάποιο κόμβο σε σχέση με το υπόλοιπο λογισμικό που μπορεί να “τρέχει” στον ίδιο κόμβο.

Η **link-local** διεύθυνση μπορεί να χρησιμοποιηθεί μόνο για επικοινωνία κόμβων που βρίσκονται στο ίδιο τοπικό δίκτυο επιπέδου 2. Οι δρομολογητές δεν μπορούν να δρομολογήσουν πακέτα πέρα από τα όρια του τοπικού δικτύου. Το πεδίο αυτής της διεύθυνσης φαίνονται στο παρακάτω σχήμα.



Link-local address format.

Οι **site-local** διευθύνσεις είναι μοναδικές και χρήσιμες μόνο μέσα στο ίδιο IPv6 domain. Με τον όρο IPv6 site εννοούμε μια περιοχή της οποίας τα όρια καθορίζονται από τον διαχειριστή του δικτύου και μπορεί να είναι από ένα μικρό οικιακό δίκτυο μέχρι το δίκτυο κάποιας πολυεθνικής εταιρίας. Τα πεδία αυτής της διεύθυνσης αναφέρονται στο σχήμα που ακολουθεί.

10 bits	38 bits	16 bits	64 bits
1111111011	0	Subnet ID	Interface identifier

Site-local address format.

Οι **global** διευθύνσεις, είναι μοναδικές σε όλη την έκταση του internet. Κάθε IPv6 κόμβος είναι πιθανόν να έχει μια ή περισσότερες τέτοιες διευθύνσεις. Οι global διευθύνσεις δημιουργούνται με την συνένωση ενός παγκοσμίως μοναδικού προθέματος το οποίο έχει αποδοθεί στο τοπικό δίκτυο επιπέδου 2 (link) και ενός μοναδικού αριθμού που προσδιορίζει το interface στο link. Τα πεδία μιας global διεύθυνσης φαίνονται στο παρακάτω σχήμα.

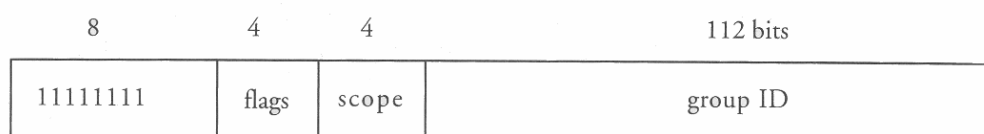
N bits	M bits	128 - N - M bits
Global routing prefix	Subnet ID	g u Interface identifier

Global address format.

Το άθροισμα $N+M$ είναι πάντα 64, εκτός από τις διευθύνσεις που ξεκινούν με “000” στα πιο σημαντικά bit. Το u-bit και το g-bit είναι τα λιγότερο σημαντικά ψηφία στο περισσότερο σημαντικό byte του interface identifier. Το u-bit υποδεικνύει αν το interface identifier είναι παγκοσμίως μοναδικό. Ο λόγος που υπάρχει το u-bit είναι ότι μπορεί να είναι χρήσιμο για άτομα να παίρνουν κάποιο παγκοσμίως μοναδικό αναγνωριστή διεπαφής στο διαδίκτυο αν και ακόμα δεν υπάρχει κάποια εφαρμογή που να κάνει χρήση. Το g-bit κάνει γνωστό αν ο αναγνωριστής είναι ενός ή ενός group από διεπαφές.

B) Multicast

Στο IPv6 δεν υπάρχουν Broadcast διευθύνσεις και ο χειρισμός των broadcast γίνεται ως ειδική περίπτωση των multicast. Οι εφαρμογές στις οποίες χρησιμοποιείται η multicast δυνατότητα μπορεί να είναι από μια διάλεξη με video broadcast στην οποία οι ενδιαφερόμενοι δηλώνουν το ενδιαφέρον να λάβουν τα συγκεκριμένα πακέτα (για ένα συγκεκριμένο χρονικό διάστημα) μέχρι την ενημέρωση ενός DNS (χρονικά μόνιμη σύνδεση). Κάθε κόμβος που συμμετέχει στο multicast θα λάβει ένα αντίγραφο κάθε πακέτου που θα στείλει ο εξυπηρετητής. Το πρωτόκολλο Multicast Listener Discovery (MLD) έχει αναπτυχθεί για να επιτρέπει σε κόμβους να εισέρχονται σε IPv6 multicast groups. Το πεδίο των σημαιών καθορίζει αν η σύνδεση θα είναι για μια χρονική περίοδο ή μόνιμη. Το πεδίο the scope καθορίζει την φυσική σύνδεση των κόμβων που συμμετέχουν στην ομάδα.



The multicast address format.

Γ) Anycast

Ένα τέτοιο πακέτο παραδίδεται σε ένα μόνο κόμβο από αυτούς που αναγνωρίζονται ως παραλήπτες του συγκεκριμένου πακέτου. Τυπικά στον κόμβο που βρίσκεται πλησιέστερα στον αποστολέα του πακέτου. Όταν το πακέτο φτάσει σε κάποιο δρομολογητή ο οποίος συνδέεται με περισσότερους από ένα παραλήπτες τότε ο δρομολογητής με κάποια μέθοδο (αλγόριθμος round robin, άλλος) επιλέγει κάποιο από αυτούς τους κόμβους. Anycast πακέτα μπορούν να χρησιμοποιηθούν στην περίπτωση που υπάρχει redundancy και ένα πακέτο μπορεί να σταλθεί παραληφθεί από περισσότερους από έναν κόμβους.



Format for anycast addresses.

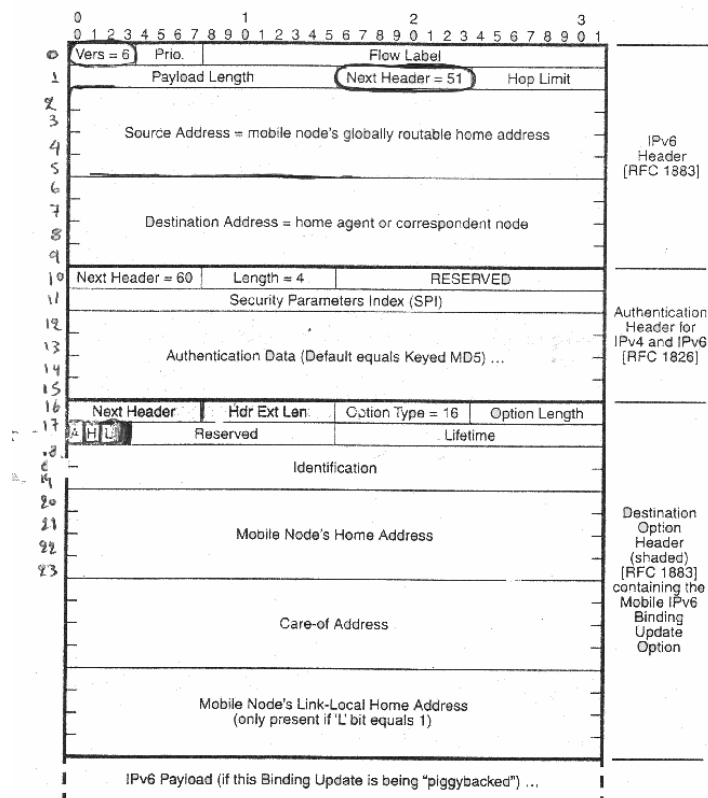
2.3. Τα είδη των πακέτων τα οποία ανταλλάσσονται στο πρωτόκολλο Mobile IPv6

2.3.1 Binding Update

Από την στιγμή που ο κινητός χρήστης αποκτήσει μια Care-of-Address, θα πρέπει να ενημερώσει τον Home Agent του και πιθανώς επιλεγμένους χρήστες με τους οποίους θα ήθελε να έχει άμεση επικοινωνία χωρίς την εμπλοκή του Home Agent. Ο Home Agent θα πρέπει να γνωρίζει πάντα την παρούσα διεύθυνση του κινητού χρήστη που εξυπηρετεί για να κάνει forward τα πακέτα που φθάνουν στο domain του στην έγκυρη διεύθυνση του χρήστη.

Η δυνατότητα ενημέρωσης και κάποιων επιλεγμένων συνδρομητών είναι μια δυνατότητα που παρέχεται από το πρωτόκολλο. Είναι προαιρετική, και αφορά κυρίως χρήστες οι οποίοι έχουν μεγάλες απαιτήσεις για άμεση και έντονη επικοινωνία με τον κινητό χρήστη, ώστε να υπάρξει και αποφόρτιση του έργου του Home Agent και εξάλειψη της πιθανότητας δημιουργίας bottlenecks στο δίκτυο από την κυκλοφορία επικοινωνίας από πολύ δραστήριους χρήστες.

Η διαδικασία της ενημέρωσης περιλαμβάνει την αποστολή του μηνύματος Binding Update κάθε φορά που βρεθεί σε κάποια καινούργια θέση. Στο μήνυμα binding update υπάρχει ένα πεδίο που καθορίζει αν ο κινητός χρήστης επιθυμεί ή όχι να του αποσταλεί πίσω μήνυμα Binding Acknowledgement από τον Home Agent αμέσως μετά από την ενημέρωση των στοιχείων που διατηρεί ο Router.



Το Binding Update πακέτο μπορεί να είναι αυτούσιο πακέτο είτε να συμπεριλαμβάνεται σε κάποιο άλλο πακέτο IPv6 (piggybacked). Στα Binding Update επιβάλλεται να συμπεριλαμβάνεται ο IPv6 Authentication Header για την εξασφάλιση της γνησιότητας των πακέτων που ανταλλάσσονται. Για τον παραπάνω λόγο θα πρέπει να υποστηρίζονται τεχνικές πιστοποίησης και κρυπτογράφησης από κάθε υλοποίηση Mobile IPv6.

Τα πεδία τα οποία περιλαμβάνει ένα πακέτο Binding Update είναι τα ακόλουθα:

Next Header: Υποδεικνύει τον τύπο Header που ακολουθεί τον παρών Header. Αν δεν ακολουθεί άλλος Header τότε η τιμή του πεδίου είναι 59.

Header Extension Length: Καθορίζουν τον αριθμό των ομάδων 4 Byte του μήκους του συγκεκριμένου Header, χωρίς να συνυπολογίζονται τα υποχρεωτικά Byte του Header.

Option Type, Option Length: Καθορίζουν το είδος και το μέγεθος της επιλογής στην οποία αναφέρονται. Η τιμή 16 στο πεδίο option type αντιστοιχεί σε ένα Binding Update πακέτο.

A bit: Καθορίζει αν ο κινητός χρήστης επιθυμεί να του αποσταλεί πίσω από τον Home Agent μήνυμα Binding Acknowledgement επιβεβαίωσης ότι η διαδικασία ολοκληρώθηκε με επιτυχία.

H bit: Ο κινητός χρήστης θέτοντας αυτό το bit ενημερώνει τον παραλήπτη του πακέτου να λειτουργήσει ως Home Agent.

L bit: Τίθεται από τον κινητό χρήστη (L = 1) για να ζητήσει από τον Home Agent να ανακατευθύνει προς αυτόν όχι μόνο τα πακέτα με διεύθυνση προορισμού (Destination Address) την Globally Routable Home Address του κινητού χρήστη, αλλά και εκείνα με διεύθυνση προορισμού την Link-local Home Address του κινητού χρήστη στο Home Link του.

Lifetime: Καθορίζει την διάρκεια ζωής των πακέτων.

Identification: Επιτρέπει στον παραλήπτη να καθορίσει σε πιο δεδομενογράφημα ανήκει το νεοφερμένο τεμάχιο. Όλα τα τεμάχια (fragments) του ίδιου δεδομενογραφήματος περιέχουν την ίδια τιμή identification.

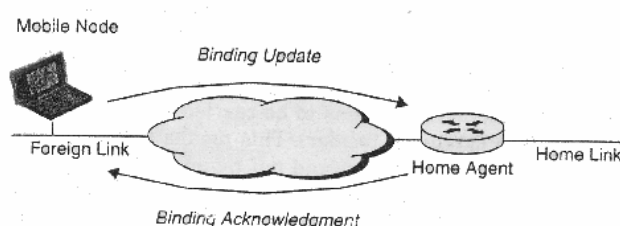
Mobile Node's Home Address: Αυτοπεριγραφικό πεδίο.

Care - of -Address: Αυτοπεριγραφικό πεδίο.

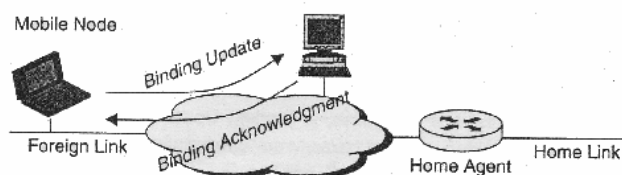
2.3.2 Binding Acknowledgement

Το πακέτο αποστέλλεται από τον παραλήπτη ενός binding update πακέτου ο οποίος μπορεί να είναι είτε ο Home Agent ή κάποιος Correspondent Node αν το πεδίο A bit του binding update έχει την τιμή 1. Το πακέτο μπορεί να σταλεί ως ξεχωριστό πακέτο ή piggybacked σε κάποιο άλλο IPv6 πακέτο. Περιλαμβάνουν τις ίδιες απαιτήσεις πιστοποίησης και κρυπτογράφησης για την ολοκλήρωση μιας ασφαλούς επικοινωνίας. Στο σχήμα που ακολουθεί παρακολουθούμε την διαδικασία Binding Acknowledgement από το Home Agent στην 1^η περίπτωση και από ένα Correspondent Node στην 2^η περίπτωση.

1.



2.



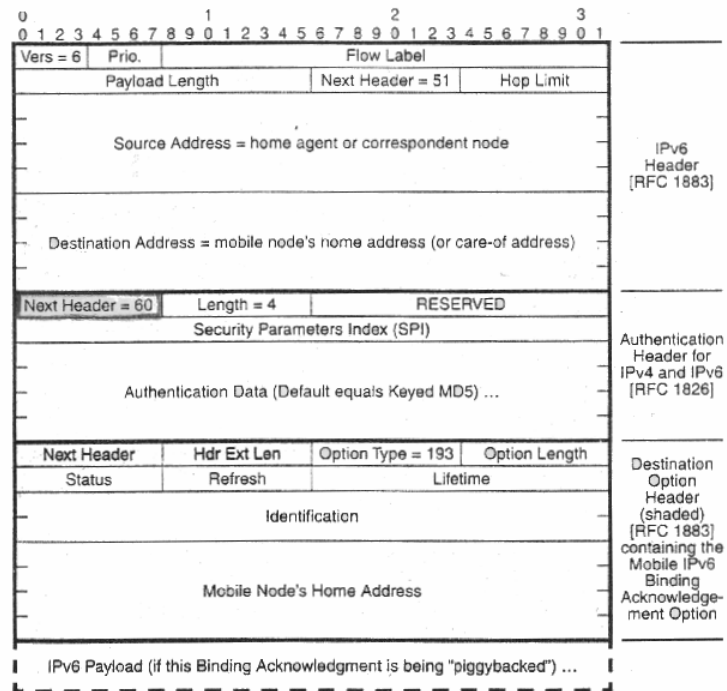
Τα πεδία του πακέτο:

Option Type: Το πεδίο έχει τιμή 193, η χρήση παραμένει ο προσδιορισμός του τύπου του πακέτου μέσω του κωδικού.

Status: Αποτελεί τον κωδικό που ενημερώνει τον κινητό χρήστη αν έγινε δεκτή η αίτησή του ή για ποιούς λόγους απορρίφθηκε. Παραθέτουμε και ένα πίνακα με τις τιμές του πεδίου status μαζί με την ερμηνεία που έχει κάθε κωδικός.

Refresh: Αναφέρει για πόσο χρόνο ο αποστολέας του Binding Acknowledgement προτίθεται να κρατήσει αποθηκευμένη στην μνήμη την τρέχουσα Care-of-address του κινητού χρήστη. Με αυτό τον τρόπο καθορίζεται πόσο συχνά θα πρέπει ο κινητός χρήστης να αποστέλλει ένα μήνυμα Binding Update στον Home Agent ή στον Correspondent Node.

Lifetime, Identification, Mobile Node's Home Address: Αποτελούν αντίγραφα των αντίστοιχων πεδίων του Binding Update.



Παρακάτω υπάρχει ένας πίνακας με τους πιθανούς κωδικούς τους οποίους μπορεί να έχει το πεδίο status word μαζί με την ερμηνεία τους.

Status Word									
Status					Ερμηνεία				
0	0	0	0	0	0	0	0	0	Επιτυχής διεκπεραίωση
128	1	0	0	0	0	0	0	0	Μη αναγνωρίσιμο λάθος
130	1	0	0	0	0	0	1	0	Το μήνυμα απορρίφθηκε από τον Administrator
131	1	0	0	0	0	0	1	1	Δεν είναι επαρκείς οι υπάρχουσες πληροφορίες
132	1	0	0	0	0	1	0	0	Δεν υποστηρίζεται λειτουργία HA
133	1	0	0	0	0	1	0	1	Δεν υπάρχει υπολογιστής συνδεδεμένος σε αυτό το υποδίκτυο
135	1	0	0	0	0	1	1	1	Απάντηση για δυναμική αναζήτηση
136	1	0	0	0	1	0	0	0	Λανθασμένο πεδίο Interface Identifier
137	1	0	0	0	1	0	0	1	Δεν υπάρχει HA για το συγκεκριμένο MN

Binding Acknowledgement Status Word

2.3.3 Binding Request

Ο Home Agent ή κάθε Correspondent Node διατηρούν κάποιους πίνακες με τα στοιχεία δρομολόγησης για κάθε mobile node το οποίο εξυπηρετούν ή με τον οποίο επικοινωνούν. Κάθε στοιχείο σε αυτούς τους πίνακες συνδέεται με τον χρόνο lifetime για τον οποίο το δεδομένο θεωρείται ισχύων. Κάποιο διάστημα πριν λήξουν οι χρόνοι εμπιστοσύνης δεδομένων για κάθε node ο Home Agent έχει την υποχρέωση να στείλει στον κινητό χρήστη μήνυμα Binding Request με το οποίο του ζητά να του αποστείλει μήνυμα Binding Update για να ενημερώσει τα δεδομένα του και τα χρονικά διαστήματα εμπιστοσύνης(lifetime). Τα Binding Request μηνύματα δεν περιέχουν Authentication Header καθώς δεν περιέχουν κάποια εμπιστευτική πληροφορία αλλά αποτελούν απλά μια ειδοποίηση,

Τα πεδία του Binding Request μηνύματος:

Option Type: Η τιμή του πεδίου είναι 194

Option Length: Η τιμή του πεδίου είναι 0

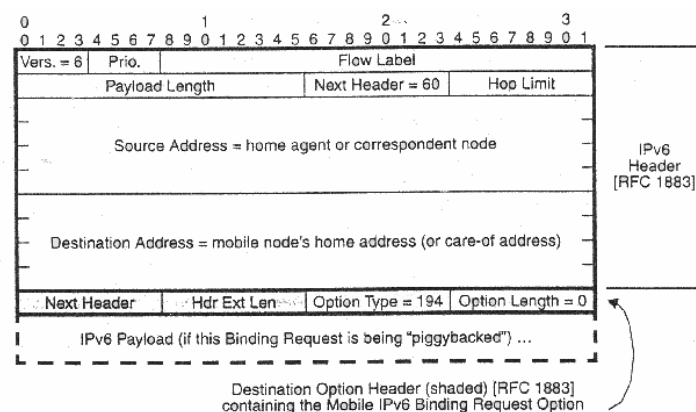
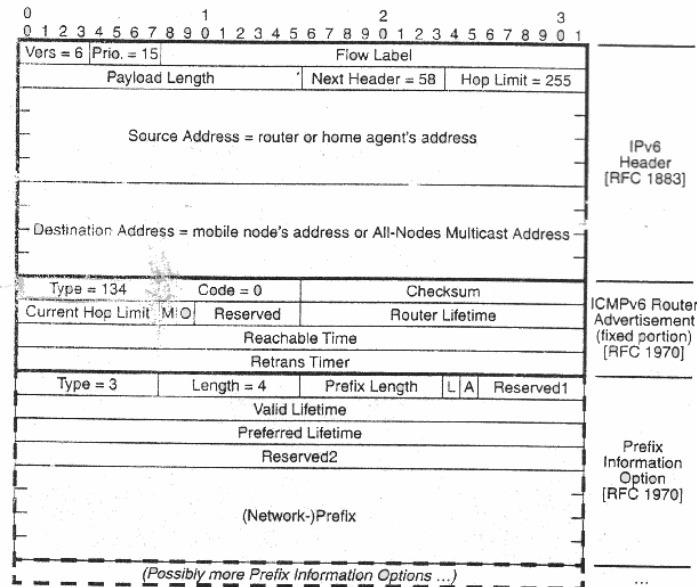


Figure 12-7 Binding Request

2.3.4 Router Advertisement

Τα μηνύματα Router Advertisement μπορούν να είναι αποτέλεσμα λήψης κάποιου Router Solicitation μηνύματος ή να στέλνονται περιοδικά. Όταν είναι αποτέλεσμα κάποιου μηνύματος Solicitation τότε αποστέλλονται στην διεύθυνση από την οποία στάλθηκε το Router Solicitation μήνυμα. Στην περίπτωση που πρόκειται για περιοδικό μήνυμα στέλνεται στην all-nodes-multicast address.

Τα Router Advertisement μηνύματα επιτρέπουν στους κόμβους πάνω στο link να ανακαλύψουν την IP διεύθυνση του default router και την διεύθυνση link layer. Επιπρόσθετα περιέχουν χρήσιμες πληροφορίες για το link όπως το link prefix και το MTU. Αυτή η πληροφορία μεταφέρεται σαν ένα ICMPv6 option.



Τα πεδία του πακέτου είναι:

Type Code: Η τιμή του πεδίου είναι 134.

Router Lifetime: Η τιμή αυτού του πεδίου (16 bit) υποδεικνύει το χρονικό διάστημα για το οποίο, ο αποστολέας του μηνύματος θα είναι διαθέσιμος για να εξυπηρετήσει τον κινητό χρήστη ως Default Router. Σε περίπτωση που ο αποστολέας θέσει τιμή 0 σε αυτό το πεδίο σημαίνει ότι δεν μπορεί να εξυπηρετήσει τον κινητό χρήστη. Σε αυτή την περίπτωση αντιλαμβάνεται ο κινητός χρήστης την θέση στην οποία βρίσκεται.

M bit: Καθορίζει αν η διαδικασία απόκτηση care-of-address για τον κινητό χρήστη θα γίνει με Stateful (M = 1) είτε θα γίνει με stateless address Autoconfiguration (M=0).

O bit: Υποδεικνύει αν έχει τεθεί ότι οι κινητοί χρήστες θα πρέπει να χρησιμοποιήσουν μηχανισμούς Stateful για να προσδιορίσουν πληροφορίες διαφορετικές από διευθύνσεις όπως DNS διευθύνσεις.

Reachable Time: Υποδεικνύει τον χρόνο σε milliseconds για τον οποίο μπορεί να θεωρήσει ένας κινητός χρήστης ότι ένας γείτονας είναι διαθέσιμος. Στην υλοποίηση του κινητού χρήστη ο μετρητής αυτός μπορεί να μηδενιστεί από την στιγμή που λάβει κάποια επιβεβαίωση λήψης πακέτου από κάποιο κόμβο στο link. Για παράδειγμα αν αυτό το πεδίο έχει τεθεί στην τιμή 10.000 τότε ο κινητός χρήστης μπορεί να υποθέσει ότι οι γείτονές του θα είναι διαθέσιμοι για 10 sec από την στιγμή που επιβεβαιώσει την επικοινωνία με κάποιο γείτονα.

Retransmission Timer: Υποδεικνύει την συχνότητα (σε milliseconds) της διαδικασίας ανακάλυψης των διευθύνσεων των γειτονικών κόμβων στο link.

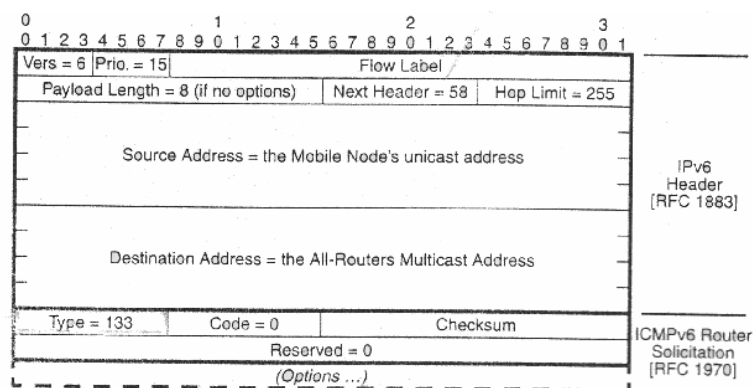
L bit: Το bit υποδεικνύει αν το network prefix μπορεί να χρησιμοποιηθεί για on-link προορισμούς όταν έχει τεθεί. Αυτό θα μπορούσε να επιτρέψει την απευθείας επικοινωνία του κινητού κόμβου με τους γείτονές του στο link χωρίς την συμμετοχή του δρομολογητή. Όταν η τιμή του bit είναι 0 ο κινητός χρήστης θα πρέπει πάντα να στέλνει τα πακέτα στον default router όταν η διεύθυνση αποστολής προκύπτει από το prefix του δικτύου.

Valid Lifetime: Το πεδίο χρησιμοποιείται για να υποδείξει στους κόμβους για πόσο χρόνο το network prefix μπορεί να θεωρηθεί ως on-link. Με άλλα λόγια τον χρόνο για τον οποίο η πληροφορία που φέρει το L bit μπορεί να θεωρηθεί έγκυρη.

Network Prefix: Ο κινητός χρήστης ελέγχει αν κάποιο από τα Network Prefixes ταιριάζει με το network prefix του Home Link του, σε αυτή την περίπτωση συμπεραίνει ότι έχει επιστρέψει στο Home Link του. Σε αντίθετη περίπτωση αν τα network prefixes δεν ταιριάζουν με το Home Link network prefix συμπεραίνει ότι είναι συνδεδεμένος σε κάποιο Foreign Link. Σε αυτή την περίπτωση συγκρίνει το πεδίο με το αντίστοιχο πεδίο στο τελευταίο Router Advertisement το οποίο έλαβε για να διαπιστώσει αν έχει μετακινηθεί από ένα Foreign Link σε κάποιο άλλο.

2.3.5 Router Solicitation

Ο κινητός χρήστης αποστέλλει μήνυμα Router Solicitation όταν θέλει άμεσα να ανιχνεύσει την θέση του και δεν επιθυμεί να περιμένει το επόμενο Router Solicitation μήνυμα. Το μήνυμα στέλνεται στη all-router multicast address. Αυτή η διεύθυνση θα πρέπει να υπάρχει hardcoded σε κάθε υλοποίηση δρομολογητή. Η λήψη ενός τέτοιου μηνύματος θα οδηγήσει σε αποστολή Router Advertisement μηνύματος από όλους τους δρομολογητές που βρίσκονται στο Link. Οι κινητοί χρήστες μπορούν να στέλνουν Router Solicitation μηνύματα συχνά για να ανακαλύπτουν δρομολογητές ή μετά την αλλαγή κάποιου link. Για την αποφυγή των συγκρούσεων μεταξύ των πακέτων όταν πολλοί κινητοί χρήστες προσκολλούνται ταυτόχρονα στο ίδιο link το πρωτόκολλο Neighbor Discovery προτείνει στους κινητούς χρήστες να καθυστερούν την αποστολή κατά μια τυχαία τιμή η οποία μπορεί να είναι από το 0 μέχρι το 1ο δευτερόλεπτο. Η δομή του μηνύματος είναι απλή και δεν χρειάζεται κάποια πιστοποίηση.



2.4. Η διαδικασία απόκτησης Care-of-Address

Η διαδικασία λήψης Care-of-Address γίνεται μέσω της διαδικασίας του Address Autoconfiguration. Μπορεί να χρησιμοποιηθεί είτε η μέθοδος Stateful Address Autoconfiguration είτε η μέθοδος Stateless Address Autoconfiguration. Την μέθοδο την καθορίζει ο Router με την τιμή του M bit στο πακέτο Router Advertisement. Εάν το M bit έχει τιμή 1 τότε η επιλεγμένη μέθοδος είναι η Stateful address Autoconfiguration ενώ αν η τιμή είναι 0 η επιλεγμένη μέθοδος είναι η stateless address Autoconfiguration. Οι δυο μέθοδοι περιγράφονται παρακάτω.

2.4.1. Stateful Address Autoconfiguration

Στην περίπτωση αυτή ο κινητός χρήστης ζητά από τον εξυπηρετητή να του παραχωρήσει μια διεύθυνση. Ο δρομολογητής θα πρέπει να κρατάει κάποιο αρχείο με τις διευθύνσεις που έχουν ανατεθεί σε κάθε κόμβο. Αυτό μπορεί να γίνει μέσω της διαδικασίας Stateful Address Autoconfiguration του πρωτοκόλλου Dynamic Host Configuration Protocol (DHCPv6 – RFC 2131) είτε με το PPP's IPv6 Configuration Protocol (RFC 2033). Το DHCP έχει επεκταθεί για να παρέχει και network prefixes σε δρομολογητές.

2.4.2 Stateless Address Autoconfiguration

Αυτή είναι η μέθοδος με την οποία ο κινητός χρήστης αποκτά “αυτόματα” μια διεύθυνση. Η μέθοδος αυτή έχει εισαχθεί για πρώτη φορά στο IPv6 (RFC 1971) και δεν υπάρχει αντίστοιχη διαδικασία στο IPv4. Η λειτουργία της μπορεί να περιγραφεί ως εξής

- Ο κινητός χρήστης αρχικά σχηματίζει μια σκυτάλη διεπαφής, έναν αναγνωριστή ο οποίος εξαρτάται από το link καθώς τυπικά είναι η link-layer address του interface. Για παράδειγμα στο πρωτόκολλο Ethernet, η σκυτάλη θα είναι η 48 bit Ethernet διεύθυνση.
- Ο κινητός χρήστης εξετάζει τα Prefix Information Option τα οποία περιέχονται μέσα στο Router Advertisement για να αναγνωρίσει τα ισχύοντα network prefixes στο link το οποίο βρίσκεται.
- Ο κινητός χρήστης σχηματίζει μια care-of-address συνενώνοντας κάποιο από τα ισχύοντα network prefixes με την σκυτάλη της διεπαφής του.

Αυτή η διαδικασία ονομάζεται stateless Autoconfiguration καθώς μια διεύθυνση δημιουργείται χωρίς να χρειάζεται να φυλαχτεί αυτή η διεύθυνση σε κάποιο κόμβο ή δρομολογητή εκτός από τον κινητό κόμβο στο οποίο ανατίθεται σε κάποια από τις διεπαφές δικτύου του. Η όλη διαδικασία είναι μια κατανεμημένη διαδικασία υπολογισμού διευθύνσεων IPv6. Αυτό προφυλάσσει από την περίπτωση δυο κόμβοι να χρησιμοποιούν την ίδια διεύθυνση και την ανάγκη να υπάρχει κάποιο ένα αρχείο το οποίο να διατηρεί όλες τις διευθύνσεις οι οποίες υπάρχουν σε κάποιο link.

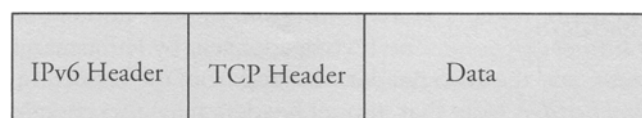
2.5. Ingress Filtering

Η δομή του διαδικτύου δεν επιτρέπει την δρομολόγηση πακέτων τα οποία δεν έχουν network prefix, ίδιο με αυτό του link στο οποίο βρίσκονται. Αυτό σημαίνει ότι αν ένα κινητός χρήστης έχει μια ορθή διεύθυνση IPv6 η οποία όμως δεν συμφωνεί με το network prefix του link στο οποίο βρίσκεται δεν θα μπορέσει ποτέ να λάβει πακέτα, γι' αυτό τον λόγο η διεύθυνση care-of-address την οποία σχηματίζει ο κινητός σε κάθε foreign link στο οποίο συνδέεται, πρέπει πάντα να συμφωνεί με την διεύθυνση του δικτύου και γι' αυτό τα πεδία τα οποία περιέχονται στο μήνυμα Router Advertisement είναι πάρα πολύ σημαντικά.

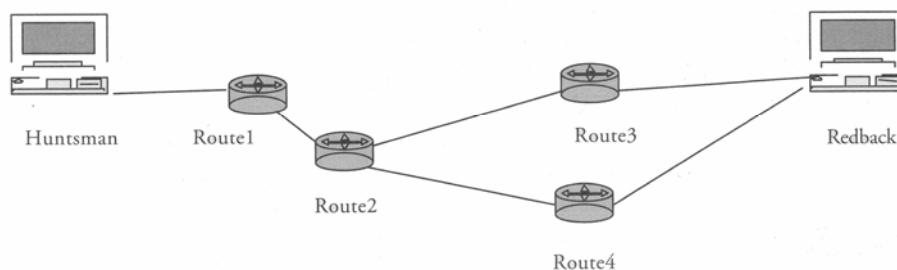
Ένας επιπρόσθετος λόγος για την αναγκαιότητα του on-link prefix είναι η ύπαρξη του Ingress Filtering. Είναι η διαδικασία η οποία χρησιμοποιείται από τους δρομολογητές ώστε να διασφαλίσει ότι **μόνο** κόμβοι με ορθές τοπολογικές διευθύνσεις μπορούν να στέλνουν μηνύματα εκτός του link. Είναι σαφές ότι με αυτό τον τρόπο αποκόπτεται η επικοινωνία του κόμβου και προς τα έξω εκτός από την προς τα μέσα επικοινωνία η οποία απαγορεύεται από τον τρόπο που γίνεται η δρομολόγηση στο διαδίκτυο ακόμα και σε routers οι οποίοι δεν υποστηρίζουν το ingress filtering. Αυτή η μέθοδος έχει αναπτυχθεί για προστασία από επιθέσεις IP spoofing. Παρόλου που η ύπαρξη του ingress filtering δεν μπορεί να αποτρέψει κάποιο κακόβουλο χρήστη να κλέψει κάποια έγκυρη τοπολογικά διεύθυνση για να κάνει μια επίθεση μπορεί να εγγυηθεί ότι η τοποθεσία από την οποία θα ξεκινήσει κάποια επίθεση μπορεί να ανιχνευτεί.

2.6. Tunneling

Επειδή η βασική διαδικασία την οποία πρέπει να υλοποιεί ένας Home Agent ώστε να μπορεί ο κινητός χρήστης να λαμβάνει τα πακέτα του κανονικά σαν να βρίσκεται στο Home Link χωρίς διακοπές στην επικοινωνία, είναι το Packet Encapsulation, θεωρήσαμε ότι θα πρέπει να γίνει ιδιαίτερη αναφορά σε αυτή την διαδικασία. Ο όρος αναφέρεται στην ενθυλάκωση ενός IP πακέτου μέσα σε κάποιο άλλο IP header το οποίο έχει ως αποτέλεσμα να δημιουργούνται πακέτα τα οποία το καθένα περιέχει ένα ή περισσότερα πακέτα (ανάλογα με το πόσες φορές έχει γίνει ενθυλάκωση του αρχικού πακέτου). Ο κόμβος από τον οποίο ξεκινά ένα tunnel λέγεται tunnel entry point ενώ ο κόμβος στο οποίο καταλήγει tunnel exit point. Η σύνδεση μεταξύ του σημείου που ξεκινά και του σημείου στο οποίο καταλήγει ένα tunnel είναι μια εικονική σημείο-προς-σημείο σύνδεση. Η μορφή του πακέτου πριν και μετά από την διαδικασία της ενθυλάκωσης φαίνεται στα παρακάτω σχήματα.



Η διαδικασία του tunneling είναι εξαιρετικά χρήσιμη για την ανάπτυξη Virtual Private Networks(VPNs) τα οποία αναπτύσσονται πολύ γοργά στις μέρες μας σε εταιρικά δίκτυα για εξασφάλιση διασύνδεσης μεταξύ διαφορετικών sites με ασφαλή τρόπο. Ας εξετάσουμε την περίπτωση κατά την οποία τα δίκτυα πίσω από τους δρομολογητές Router1, Router3, Router4, ανήκουν στην ίδια εταιρία και τυχαίνει επειδή έχουν τοπογραφική απόσταση (π.χ βρίσκονται σε διαφορετική πόλη) να υπάρχουν ενδιάμεσα πολλούς δρομολογητές στο διαδίκτυο. Αυτό που θα ήθελε να διασφαλίσει μια εταιρία είναι η κυκλοφορία μεταξύ του 1 και του 3 ή 4 να είναι ασφαλής παρόλου που διέρχεται από μη ασφαλής κόμβους. Ένας τρόπος για να ολοκληρωθεί αυτό το έργο είναι με την χρήση του IPsec(π.χ ESP) μεταξύ των 2 δρομολογητών οι οποίοι βρίσκονται στα όρια του δικτύου δηλαδή 1 και 3 ή 1 και 4. Ο μόνος τρόπος να προστατεύσεις αυτά τα πακέτα χωρίς να αλλάξεις το περιεχόμενό τους είναι να τα ενθυλακώσεις σε ασφαλή tunnel. Σε αυτή την περίπτωση ο δρομολογητής 1 θα ενθυλακώσει το αρχικό πακέτο με κάποιο άλλο IPv6 Header χρησιμοποιώντας την διεύθυνσή του ως την διεύθυνση αποστολέα και την διεύθυνση του δρομολογητή 4 ως διεύθυνση παραλήπτη.

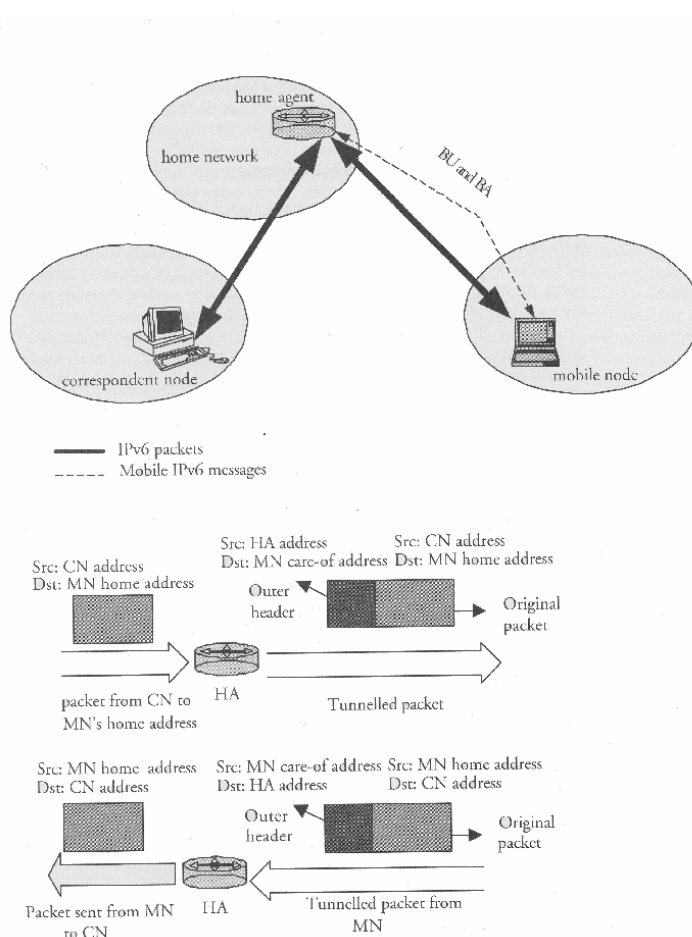


Σε αυτό το σενάριο ο δρομολογητής 1 είναι το tunnel entry point και ο δρομολογητής 4 το tunnel exit point. Ο αρχικός header ονομάζεται original header ενώ αυτός που προκύπτει από την ενθυλάκωση ονομάζεται tunnel header. Πρέπει να σημειωθεί ότι οι tunnel header μπορούν να περιέχουν και κάποιους extension headers. Στο παραπάνω σχήμα φαίνεται το περιεχόμενο ενός πακέτου όταν χρησιμοποιείται η μέθοδος ESP.

Ο παραλήπτης του ενθυλακωμένου πακέτου θα πρέπει να το εξετάσει σε σχέση με την ασφάλεια την οποία του έχει προσδοθεί και αν έχει παραδοθεί ασφαλώς να το απο-θυλακώσει και να το παραδώσει στο τοπικό δίκτυο το οποίο βρίσκεται πίσω του στην αρχική του μορφή.

Στο σχήμα που ακολουθεί φαίνεται το πως η διαδικασία της ενθυλάκωσης ενσωματώνεται στο πρωτόκολλο Mobile IPv6. Τα πακέτα τα οποία έχουν ως προορισμό τον mobile node ο οποίος έχει απομακρυνθεί από το home link αλλά έχει φροντίσει να διατηρήσει το binding με τον home agent ενθυλακώνονται με διεύθυνση προορισμού την care-of-address του κινητού χρήστη που αντιστοιχεί στο foreign link. Μόλις το πακέτο φτάσει στον mobile node αναγνωρίζεται ότι πρόκειται για ενθυλακωμένο, απο-θυλακώνεται και προκύπτει το αρχικό πακέτο το οποίο είχε στείλει ο correspondent κόμβος. Ο mobile node μπορεί να επικοινωνήσει με κάθε correspondent κόμβο με δυο τρόπους:

- Να αποστείλει το πακέτο ενθυλακωμένο στον Home Agent και στην συνέχεια ο δρομολογητής να το αποστείλει στον correspondent node, αποφεύγοντας το πιθανό ingress filtering καθώς τα πακέτα θα έχουν ορθές τοπολογικά διευθύνσεις IPv6. Την περίπτωση αυτή παρακολουθούμε στο σχήμα που ακολουθεί.
- Να δημιουργήσει απευθείας binding με τον correspondent node αποφεύγοντας την συμφόρηση κυκλοφορίας στον home agent αλλά αντιμετωπίζοντας τον κίνδυνο λόγω ingress filtering να απορριφθούν σε κάποιο δρομολογητή τα πακέτα του.



Κεφάλαιο 3^ο

Η Αρχιτεκτονική του Συστήματος

3.1 Εισαγωγή

Στο κεφάλαιο αυτό, παρουσιάζεται η αρχιτεκτονική δομή του συστήματος που υλοποιήθηκε. Περιγράφονται οι μονάδες που χρησιμοποιήθηκαν καθώς και ο λειτουργικός τους ρόλος. Για την ανάπτυξη του συστήματος επιλέχθηκε ο σχεδιασμός με έμφαση σε **structural** σχεδιασμό ψηφιακών συστημάτων για να επιτευχθεί:

- Λειτουργικός διαχωρισμός των έργων τα οποία πρέπει να πραγματοποιηθούν για την υλοποίηση των λειτουργιών του πρωτοκόλλου.
- Ευκολότερη περαιτέρω ανάπτυξη του συστήματος με προσθήκη νέων μονάδων (IPsec, διαχείρισης κλειδιών).
- Ευκολότερη ενσωμάτωση στο παρόν κύκλωμα υλοποιήσεων των IPv6, IPv4 ώστε να αποτελεί μια ολοκληρωμένη υλοποίηση σε επίπεδο IP. Στην παρούσα υλοποίηση έχουν προβλεφθεί οι διεπαφές προς IPv6 και IPv4 όπως θα περιγραφεί στην συνέχεια.
- Ευκολότερη λογική κατανόηση του συστήματος και των λειτουργιών του για τους αναγνώστες της παρούσας διπλωματικής εργασίας.

Στο παρόν κεφάλαιο αναλύονται ενδελεχώς οι λειτουργίες κάθε μονάδας με χρήση διαγραμμάτων ροής των διαφόρων μηχανών καταστάσεων οι οποίες υλοποιούνται σε κάθε δομική μονάδα αλλά και διαγραμμάτων διεργασιών και επικοινωνίας των δομικών μονάδων του συστήματος ώστε να δοθεί με όσο το δυνατόν παραστατικότερο τρόπο η λειτουργία του συστήματος.

3.2 Η Αρχιτεκτονική σε Επίπεδα Ιεραρχίας

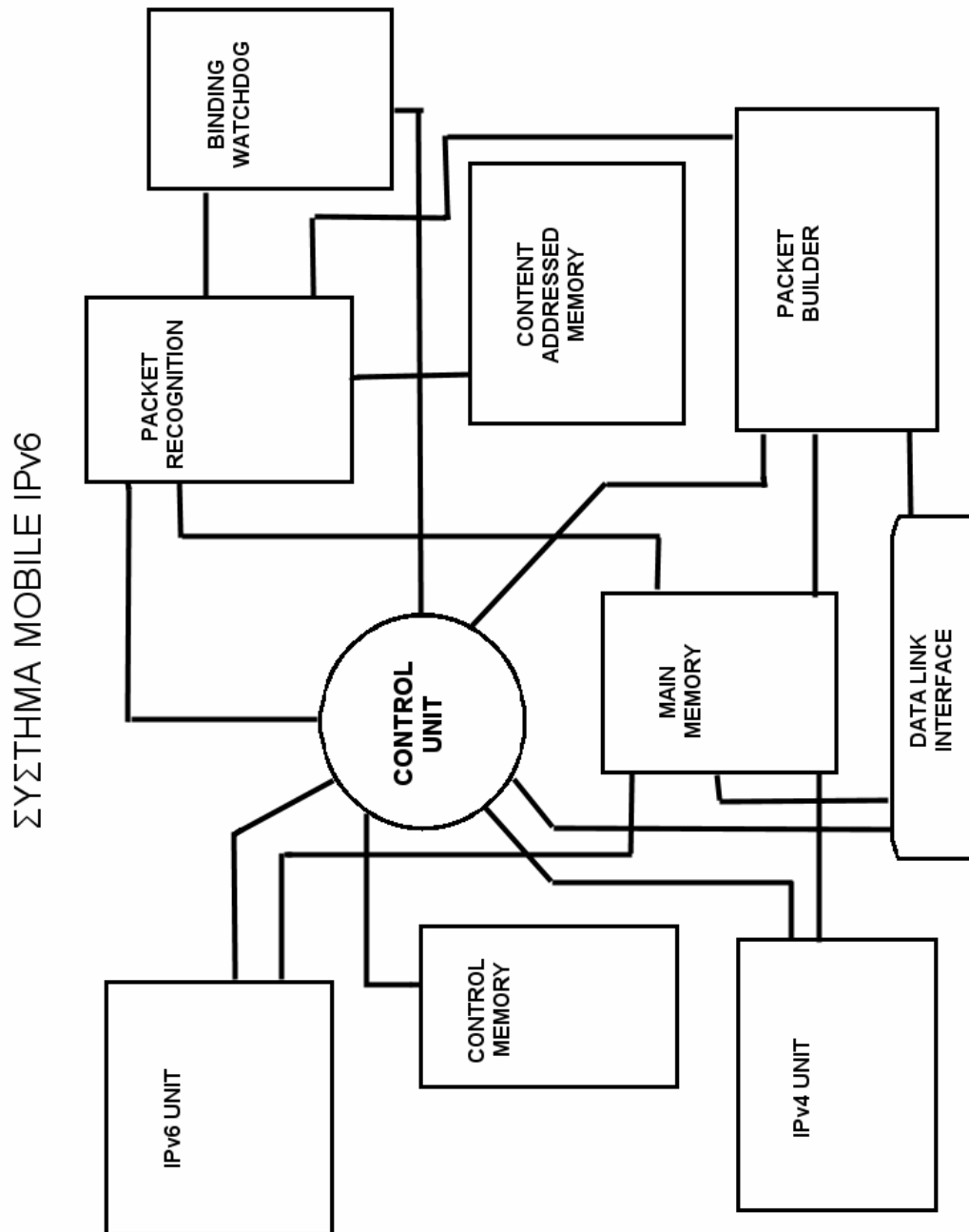
Το σύστημα έχει υλοποιηθεί σε structural δομή σε 3 επίπεδα αρχίζοντας από πάνω προς τα κάτω(Οι υλοποιήσεις των IPv6, IPv4 δεν έχουν προσμετρηθεί στην αρίθμηση των επιπέδων του συστήματος).

1^ο Επίπεδο: Είναι το επίπεδο του συστήματος το οποίο έχει τις διεπαφές του προς το Data Link Layer και μόνο.

2^ο Επίπεδο: Περιγράφετε στην σελίδα που ακολουθεί και αποτελείται από τις εξής μονάδες:

- Control Unit
- Control Memory
- Main Memory
- Packet Recognition
- Content Addressed Memory
- Binding Update Watchdog
- Packet Builder
- IPv4 Unit
- IPv6 Unit

2^ο Επίπεδο Structural



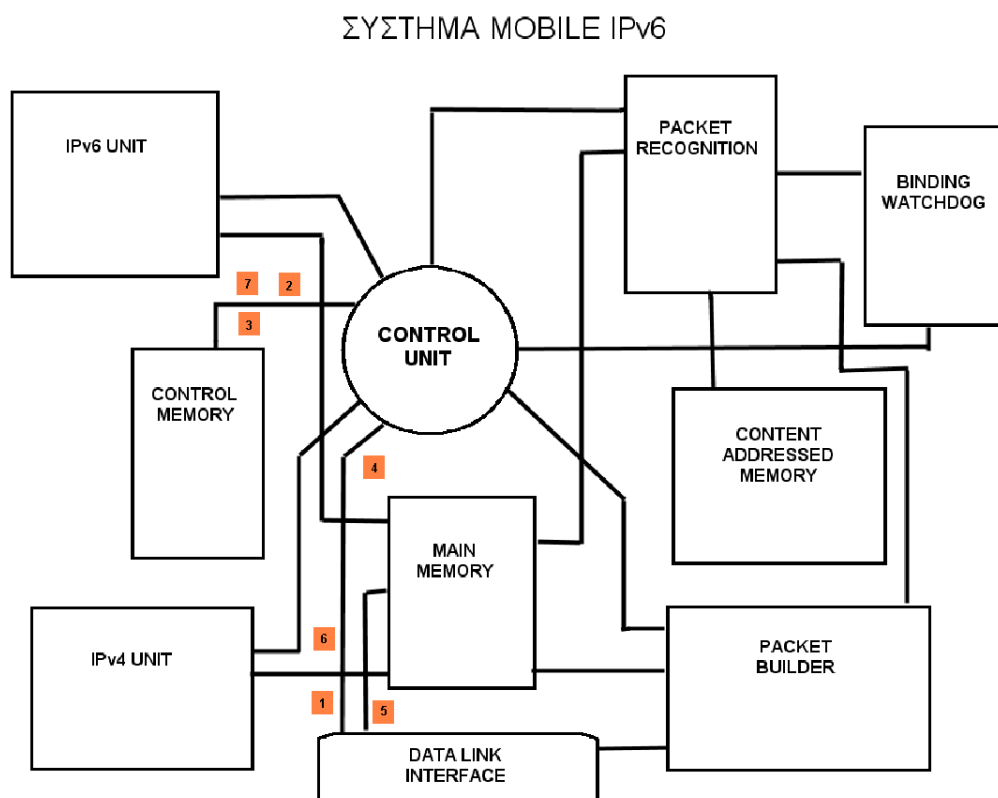
3ο Επίπεδο: Κάποιες μονάδες από αυτές που παρουσιάζονται στο προηγούμενο 2^ο επίπεδο περιέχουν υπομονάδες. Συγκεκριμένα

- Η μονάδα Packet Builder αποτελείται από 6 υπομονάδες
- Η μονάδα Content Addressed Memory αποτελείται από 2 μονάδες
- Οι μονάδες IPv4, IPv6

Τα χαμηλότερα επίπεδα από τις structural υλοποιήσεις αποτελούνται από behavioural λογικές. Η ανάλυση των λειτουργιών του συστήματος θα γίνει στο 2^ο επίπεδο και όπου περιέχονται και υπομονάδες θα περνάμε και στο 3^ο επίπεδο για να αναλυθεί η λειτουργία κάθε υπομονάδας ξεχωριστά.

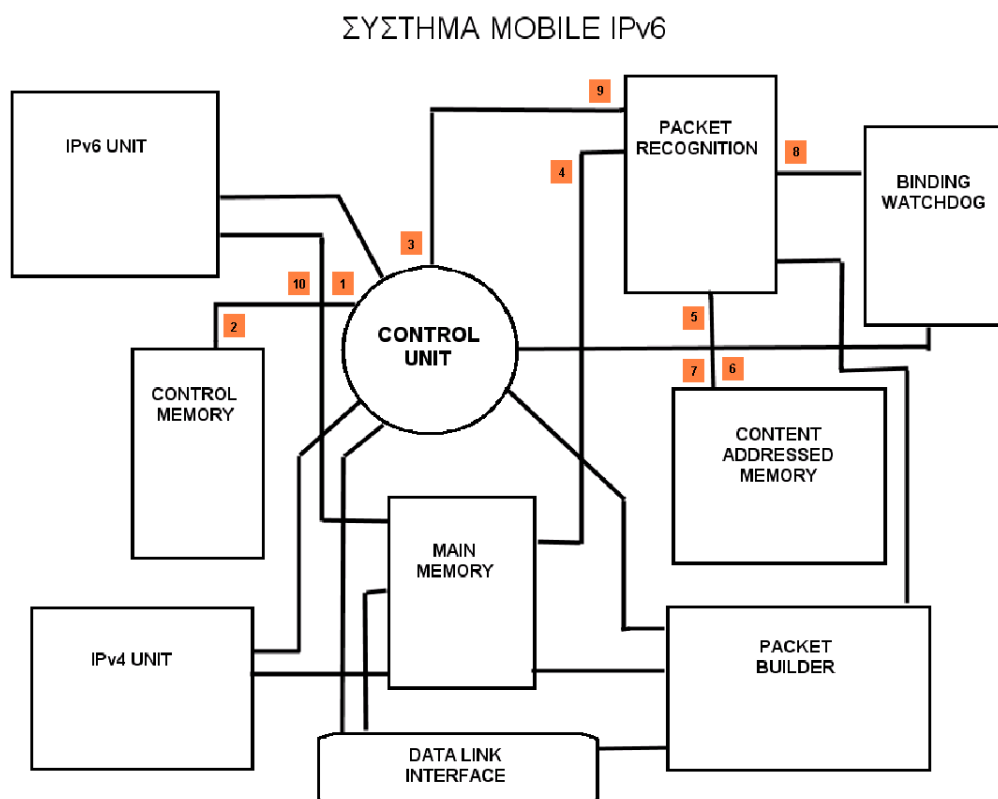
3.3 Η εκτέλεση των έργων του Συστήματος σε βήματα

3.3.1 Εισαγωγή πακέτων από το Data Link Layer



- 1) Το Data Link ειδοποιεί την Control Unit ότι έχει πακέτο να στείλει στο IP Layer συγκεκριμένου μεγέθους σε blocks 2048 bytes.
- 2) Η Control Unit κάνει ερώτηση στο Memory Management Module (Control Memory) αν υπάρχει διαθέσιμος χώρος προς αποθήκευση στην Main Memory.
- 3) Η Control Memory ελέγχει την κατάσταση μνήμης και διαπιστώνει ότι υπάρχει διαθέσιμος χώρος αποθήκευσης για το πακέτο, ο οποίος αρχίζει από κάποιο συγκεκριμένο block στην Main Memory και ενημερώνει σχετικά την Control Unit.
- 4) Η Control Unit σε περίπτωση που δεχθεί θετική απάντηση από την Control Memory ενημερώνει το Data Link Layer ότι μπορεί να γράψει το πακέτο σε κάποιο συγκεκριμένο block της μνήμης.
- 5) Το Data Link Layer κάνει εγγραφή του πακέτου στην Βασική Μνήμη, αρχίζοντας να γράφει από την αρχή του block το οποίο ήταν διαθέσιμο.
- 6) Όταν το Data Link τελειώσει το γράψιμο του πακέτου στην Βασική Μνήμη του συστήματος ενημερώνει την Control Unit ότι τελείωσε η εγγραφή.
- 7) Η Control Unit ενημερώνει την μονάδα Control Memory ότι η εγγραφή του τελευταίου πακέτου τελείωσε ώστε να ενημερώσει την κατάσταση της μνήμης. Επίσης κάνει εγγραφή στο log system της έγινε εγγραφή πακέτου από το Data Link Layer.

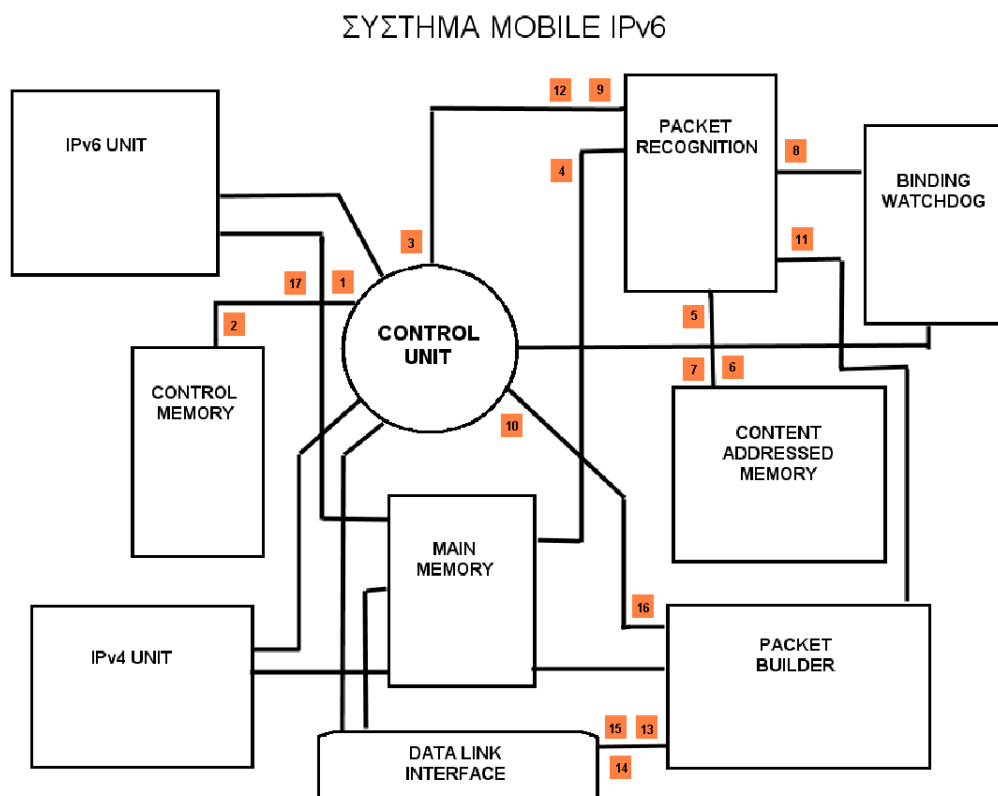
3.3.2 Binding Update Without Binding Acknowledgement



- 1) Η Control Unit υλοποιεί εσωτερικά κάποιο αλγόριθμο για τον προσδιορισμό της επόμενης εργασίας. Αν δεν έχει έρθει κάποιο interrupt από το Binding Watchdog ή δεν είναι σε αναμονή εγγραφή πακέτου από το Data Link Layer τότε εκκινεί την διαδικασία ανάγνωσης του επόμενου πακέτου. Κάνει λοιπόν αίτηση στην μονάδα Control Memory ποια είναι η διεύθυνση block του επόμενου πακέτου.
- 2) Η Control Memory απαντά στην Control Unit ότι τα δεδομένα προς επεξεργασία αρχίζουν από κάποιο συγκεκριμένο block της μνήμης, χωρίς να γνωρίζει το μέγεθος του πακέτου καθώς λειτουργεί σε υψηλότερο επίπεδο σε σχέση με την μονάδα Packet Recognition η οποία επεξεργαζόμενη το πακέτο στην Βασική Μνήμη αναγνωρίζει το μέγεθος του πακέτου.
- 3) Η μονάδα Control Unit στέλνει κωδικό εντολής εκκίνησης της διαδικασίας αναγνώρισης του επόμενου πακέτου, όπως επίσης και της διεύθυνση του block από την οποία αρχίζει το συγκεκριμένο πακέτο.
- 4) Η Packet Recognition μονάδα κάνει ανάγνωση της Βασικής Μνήμης με σκοπό να αναγνωρίσει το είδος του πακέτου το οποίο βρίσκεται στην συγκεκριμένη θέση.
- 5) Η Packet Recognition κάνει αίτηση αναζήτησης στην CAM μονάδα, για να διαπιστώσει αν η source address του πακέτου είναι μια διεύθυνση mobile node home address δηλαδή αν αντιστοιχεί στην Globally Routable Address του κινητού χρήστη. Το συγκεκριμένο σύστημα Home Agent υποστηρίζει την κινητικότητα σε 32 κινητούς χρήστες.

- 6) Η μονάδα CAM απαντά ότι η source address ανήκει σε υποστηριζόμενο κινητό χρήστη, όπως επίσης και την θέση στο table Mobile Node Home Address στην οποία υπάρχει η καταχώρηση, οπότε η Packet Recognition διαπιστώνει μαζί με τους υπόλοιπους ελέγχους σε διάφορα πεδία του μηνύματος ότι πρόκειται για μήνυμα Binding Update without Binding Acknowledgement και εκκινεί την διαδικασία Binding Update.
- 7) Η Packet Recognition κάνει εγγραφή στον δεύτερο πεδίο της CAM το οποίο περιλαμβάνει την Care of Address του κινητού χρήστη, την διεύθυνση Care of Address την οποία διαβάζει από το πακέτο. Η μονάδα σύμφωνα με την αρχιτεκτονική μιας CAM εγγράφει το δεδομένο στην αντίστοιχη θέση στην οποία βρίσκεται στον άλλο πίνακα η Home Address, ώστε να υπάρχει άμεση αντιστοιχία και εύκολη μεταγλώττιση.
- 8) Η Packet Recognition κάνει εγγραφή στην μονάδα Binding Watchdog του Lifetime πεδίου το οποίο διαβάζει από το Binding Update μήνυμα στην θέση που αντιστοιχεί στον συγκεκριμένο κινητό χρήστη.
- 9) Η μονάδα Packet Recognition ενημερώνει μέσω της διεπαφής feedback code την Control Unit ότι η διαδικασία Binding Update Without Acknowledgement ολοκληρώθηκε με επιτυχία όπως επίσης την ενημερώνει και για το μέγεθος του πακέτου σε blocks μνήμης.
- 10) Η Control Unit ενημερώνει την μονάδα Control Memory ότι τελείωσε η επεξεργασία του πακέτου συγκεκριμένου μεγέθους blocks ώστε να ενημερώσει την κατάσταση μνήμης, ουσιαστικά θεωρώντας ότι μετά την επεξεργασία τα συγκεκριμένα blocks του πακέτου είναι διαθέσιμα για εγγραφή νέου πακέτου. Επίσης η Control Unit κάνει εγγραφή στο Log System της ότι πραγματοποιήθηκε επεξεργασία Binding Update without Acknowledgement με την εγγραφή συγκεκριμένου κωδικού στην κυκλική Log Memory (χωράει 256 events) στην θέση που υποδεικνύεται από το process_id **mod(256)** καθώς κάθε καινούργια διαδικασία που εκκινείτε έχει ως χαρακτηριστικό αναγνώρισης ένα process_id.

3.3.3 Binding Update With Binding Acknowledgement

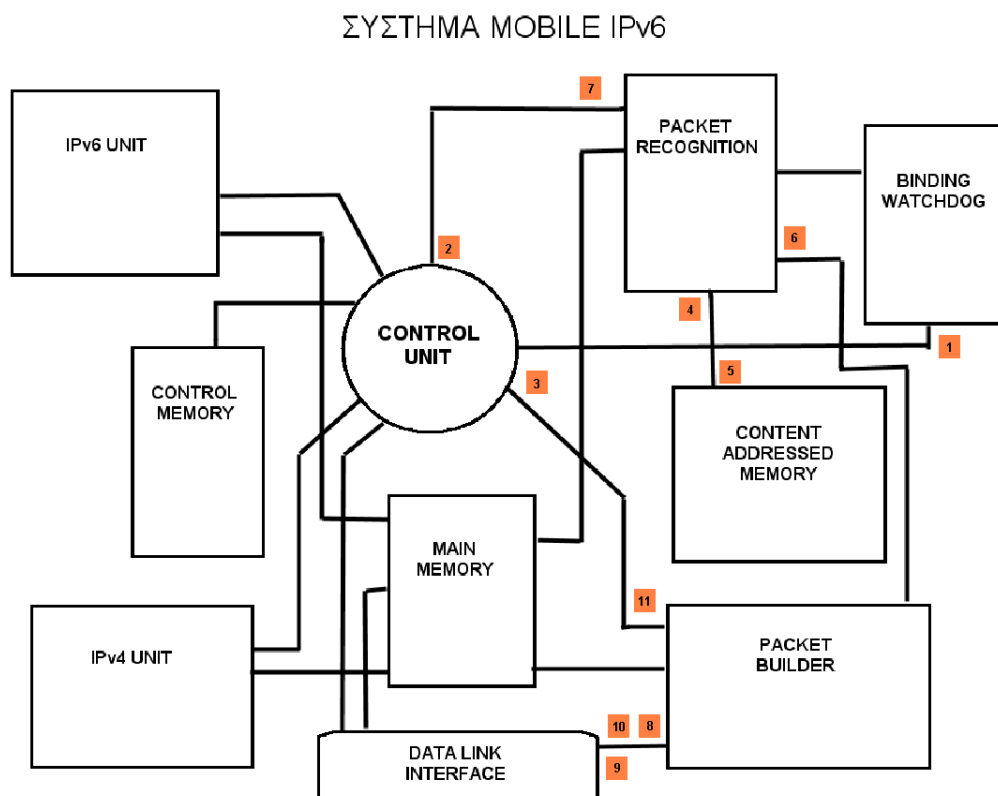


- 1) Η Control Unit υλοποιεί εσωτερικά κάποιο αλγόριθμο για τον προσδιορισμό της επόμενης εργασίας. Αν δεν έχει έρθει κάποιο interrupt από το Binding Watchdog ή δεν είναι σε αναμονή εγγραφή πακέτου από το Data Link Layer τότε εκκινεί την διαδικασία ανάγνωσης του επόμενου πακέτου. Κάνει λοιπόν αίτηση στην μονάδα Control Memory ποια είναι η διεύθυνση block του επόμενου πακέτου.
- 2) Η Control Memory απαντά στην Control Unit ότι τα δεδομένα προς επεξεργασία αρχίζουν από κάποιο συγκεκριμένο block της μνήμης, χωρίς να γνωρίζει το μέγεθος του πακέτου καθώς λειτουργεί σε υψηλότερο επίπεδο σε σχέση με την μονάδα Packet Recognition η οποία επεξεργαζόμενη το πακέτο στην Βασική Μνήμη αναγνωρίζει το μέγεθος του πακέτου.
- 3) Η μονάδα Control Unit στέλνει κωδικό εντολής εκκίνησης της διαδικασίας αναγνώρισης του επόμενου πακέτου, όπως επίσης και της διεύθυνση του block από την οποία αρχίζει το συγκεκριμένο πακέτο.
- 4) Η Packet Recognition μονάδα κάνει ανάγνωση της Βασικής Μνήμης με σκοπό να αναγνωρίσει το είδος του πακέτου το οποίο βρίσκεται στην συγκεκριμένη θέση.
- 5) Η Packet Recognition κάνει αίτηση αναζήτησης στην CAM μονάδα, για να διαπιστώσει αν η source address του πακέτου είναι μια διεύθυνση mobile node home address δηλαδή αν αντιστοιχεί στην Globally Routable Address του κινητού χρήστη. Το συγκεκριμένο σύστημα Home Agent υποστηρίζει την κινητικότητα σε 32 κινητούς χρήστες.
- 6) Η μονάδα CAM απαντά ότι η source address ανήκει σε υποστηριζόμενο κινητό χρήστη, όπως επίσης και την θέση στο table Mobile Node Home

Address στην οποία υπάρχει η καταχώρηση, οπότε η Packet Recognition διαπιστώνει μαζί με τους υπόλοιπους ελέγχους σε διάφορα πεδία του μηνύματος ότι πρόκειται για μήνυμα Binding Update with Binding Acknowledgement και εκκινεί την διαδικασία Binding Update.

- 7) Η Packet Recognition κάνει εγγραφή στον δεύτερο πεδίο της CAM το οποίο περιλαμβάνει την Care of Address του κινητού χρήστη, την διεύθυνση Care of Address την οποία διαβάζει από το πακέτο. Η μονάδα σύμφωνα με την αρχιτεκτονική μιας CAM εγγράφει το δεδομένο στην αντίστοιχη θέση στην οποία βρίσκεται στον άλλο πίνακα η Home Address, ώστε να υπάρχει άμεση αντιστοιχία και εύκολη μεταγλώττιση.
- 8) Η Packet Recognition κάνει εγγραφή στην μονάδα Binding Watchdog του Lifetime πεδίου το οποίο διαβάζει από το Binding Update μήνυμα στην θέση που αντιστοιχεί στον συγκεκριμένο κινητό χρήστη.
- 9) Η μονάδα Packet Recognition ενημερώνει μέσω της διεπαφής feedback code την Control Unit ότι η διαδικασία πρόκειται για Binding Update with Binding Acknowledgement ώστε η Control Unit να δώσει τα κατάλληλα σήματα επιλογής κυκλώματος στην μονάδα Packet Builder η οποία εσωτερικά υλοποιεί 4 μονάδες κατασκευής διαφορετικών πακέτων οι οποίες έχουν κοινά interfaces προς Packet Recognition και Data Link Layer και η επιλογή γίνεται με decoder και mux αντίστοιχα.
- 10) Η μονάδα Control Unit δίνει Control σήμα “00” στην μονάδα Packet Builder Module για να επιλεγεί το κύκλωμα Binding Acknowledgement Building Module.
- 11) Η μονάδα Packet Recognition στέλνει στην Packet Builder Module τα δεδομένα των πεδίων του πακέτου τα οποία δεν είναι σταθερά σύμφωνα με την σύμβαση που έχει αναπτυχθεί, δηλαδή ότι στο 1^ο clk θα δοθεί κάποιο δεδομένο, στο 2^ο κάποιο άλλο ώστε να συμπληρωθούν όλα τα μεταβλητά πεδία του πακέτου.
- 12) Όταν η Packet Recognition ολοκληρώσει την κατασκευή του πακέτου ενημερώνει ότι το χτίσιμο πακέτου τελείωσε, όπως επίσης ενημερώνει και για το μέγεθος του πακέτου που επεξεργάστηκε σε memory blocks.
- 13) Η μονάδα Packet Builder από την στιγμή που ολοκληρωθεί η κατασκευή του πακέτου ενημερώνει μέσω του interface που έχει με το Data Link Layer ότι έχει διαθέσιμο πακέτο προς μεταφορά συγκεκριμένου μεγέθους.
- 14) Το Data Link Layer απαντά θετικά στην μεταφορά του πακέτου, δηλαδή ότι έχει διαθέσιμη μνήμη για αποθήκευση του πακέτου.
- 15) Η μονάδα Packet Builder από την στιγμή που πάρει θετική απάντηση αντιγράφει το πακέτο της 32 bits ανά clk στην διεπαφή προς Data Link Layer.
- 16) Όταν τελειώσει η μεταφορά του πακέτου στο Data Link Layer η μονάδα Packet Builder ενημερώνει ότι είναι διαθέσιμη για νέα επεξεργασία.
- 17) Η Control Unit ενημερώνει την μονάδα Control Memory ότι τελείωσε η επεξεργασία του πακέτου συγκεκριμένου μεγέθους blocks ώστε να ενημερώσει την κατάσταση μνήμης, ουσιαστικά θεωρώντας ότι μετά την επεξεργασία τα συγκεκριμένα blocks του πακέτου είναι διαθέσιμα για εγγραφή νέου πακέτου. Επίσης η Control Unit κάνει εγγραφή στο Log System της ότι πραγματοποιήθηκε επεξεργασία Binding Update with Acknowledgement με την εγγραφή συγκεκριμένου κωδικού στην κυκλική Log Memory (χωράει 256 events) στην θέση που υποδεικνύεται από το process_id **mod(256)** καθώς κάθε καινούργια διαδικασία που εκκινείτε έχει ως χαρακτηριστικό αναγνώρισης ένα process_id.

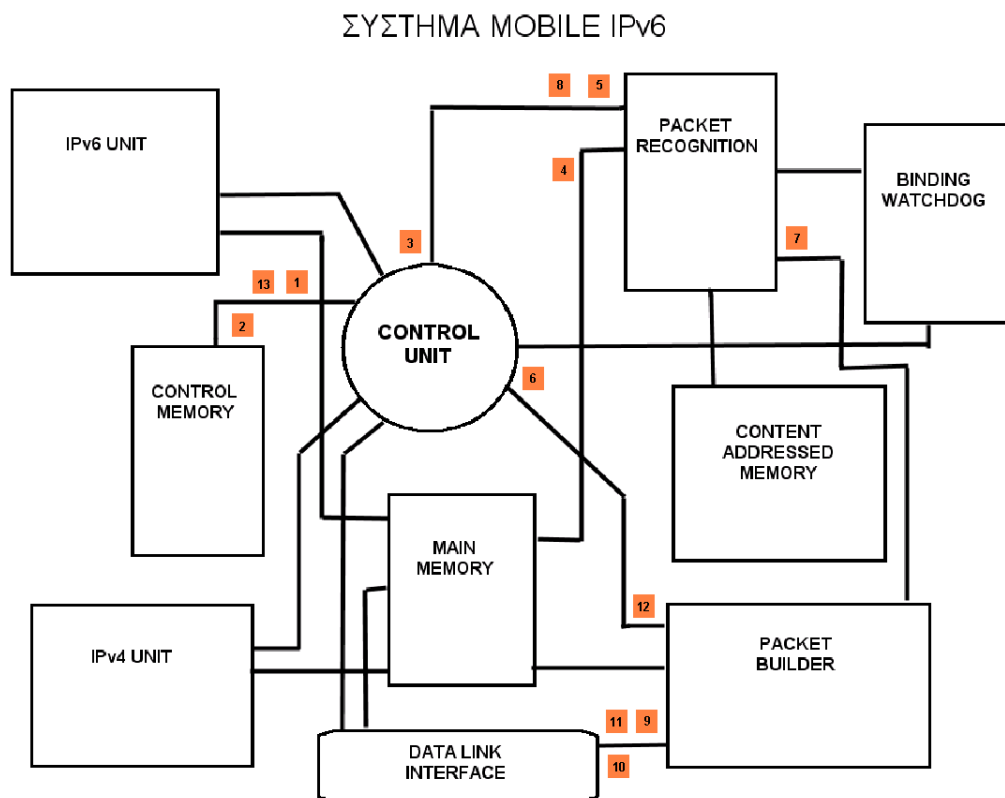
3.3.4 Binding Request



- 1) Η μονάδα Binding Lifetime Watchdog όταν κάποια τιμή Binding Lifetime για κάποιο κόμβο πέσει κάτω από κάποιο προκαθορισμένο όριο, μετά από την μείωση που γίνεται ύστερα από κάθε sec για όλα τα Lifetime πεδία, δίνει interrupt στην Control Unit με τον αριθμό του κινητού χρήστη ο οποίο χρειάζεται ανανέωση binding, ώστε η Control Unit να εκκινήσει διαδικασία Binding Request για τον συγκεκριμένο κινητό χρήστη.
- 2) Η Control Unit στέλνει στην μονάδα Packet Recognition κωδικό εκκίνησης διαδικασίας κατασκευής πακέτου Binding Request για κάποιο συγκεκριμένο κινητό χρήστη.
- 3) Η Control Unit στέλνει Control σήμα “01” στην Packet Builder μονάδα, ώστε να επιλεγεί το κύκλωμα Binding Request Building Module.
- 4) Η Packet Recognition ζητά από την CAM να του στείλει την διεύθυνση Care-of-Address του κινητού χρήστη στον οποίο πρέπει να στείλουν Binding Request.
- 5) Η CAM μνήμη απαντά με την Care-of-Address η οποία αντιστοιχεί για τον συγκεκριμένο κινητό χρήστη η οποία έχει εγγραφεί στην CAM κατά την διάρκεια του τελευταίου Binding Update.
- 6) Η Packet Recognition μονάδα στέλνει τα κατάλληλα δεδομένα στην Packet Builder μονάδα για να κατασκευαστεί το πακέτο Binding Request.
- 7) Όταν η μονάδα Packet Recognition τελειώσει την κατασκευή του Binding Request πακέτου ενημερώνει μέσω της διεπαφής feedback code την Control Unit.
- 8) Η Packet Builder μονάδα ενημερώνει το Data Link Layer ότι έχει διαθέσιμο πακέτο προς αποστολή συγκεκριμένου μεγέθους.

- 9) Το Data Link Layer ενημερώνει ότι είναι έτοιμο για να δεχθεί το πακέτο.
- 10) Η μονάδα Packet Builder στέλνει το πακέτο ανά 4 bytes ανά clk στο Data Link Layer.
- 11) Όταν τελειώσει η μεταφορά η μονάδα Packet Builder ενημερώνει ότι η Binding Request Building μονάδα είναι έτοιμη για επόμενη επεξεργασία και η Control Unit με βάσει τις προτεραιότητες στις διαδικασίες θα διαλέξει την επόμενη διαδικασία προς εκτέλεση.

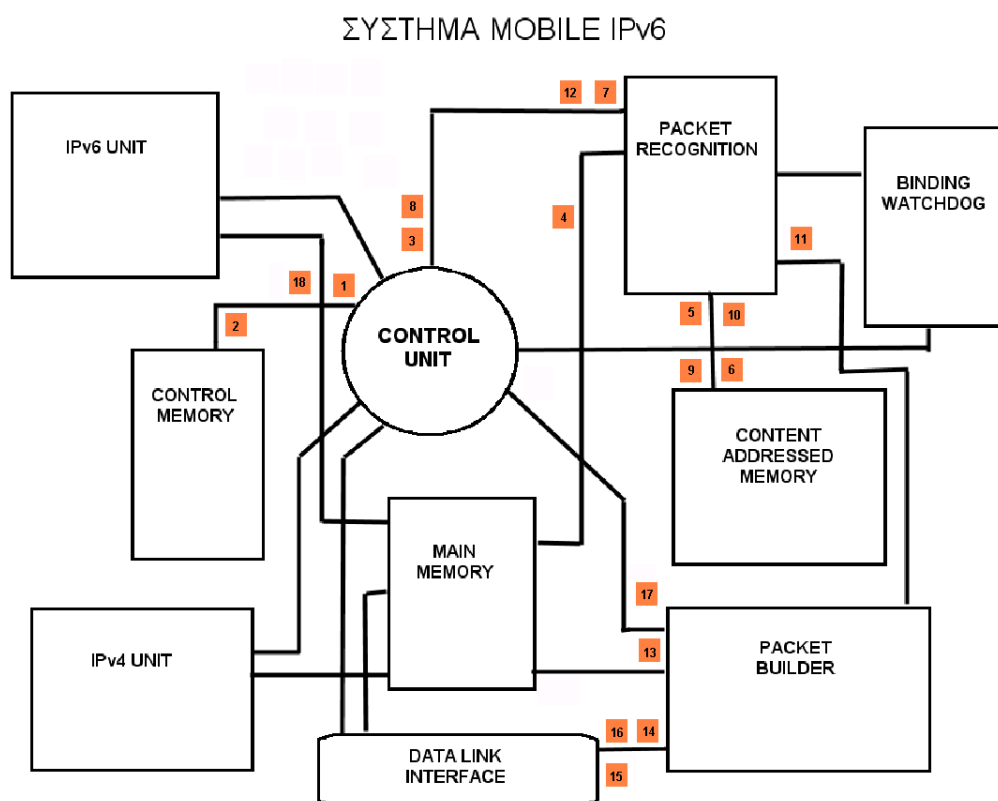
3.3.5 Router Advertisement



- 1) Η Control Unit υλοποιεί εσωτερικά κάποιο αλγόριθμο για τον προσδιορισμό της επόμενης εργασίας. Αν δεν έχει έρθει κάποιο interrupt από το Binding Watchdog ή δεν είναι σε αναμονή εγγραφή πακέτου από το Data Link Layer τότε εκκινεί την διαδικασία ανάγνωσης του επόμενου πακέτου. Κάνει λοιπόν αίτηση στην μονάδα Control Memory ποια είναι η διεύθυνση block του επόμενου πακέτου.
- 2) Η Control Memory απαντά στην Control Unit ότι τα δεδομένα προς επεξεργασία αρχίζουν από κάποιο συγκεκριμένο block της μνήμης, χωρίς να γνωρίζει το μέγεθος του πακέτου καθώς λειτουργεί σε υψηλότερο επίπεδο σε σχέση με την μονάδα Packet Recognition η οποία επεξεργαζόμενη το πακέτο στην Βασική Μνήμη αναγνωρίζει το μέγεθος του πακέτου.
- 3) Η μονάδα Control Unit στέλνει κωδικό εντολής εκκίνησης της διαδικασίας αναγνώρισης του επόμενου πακέτου, όπως επίσης και της διεύθυνση του block από την οποία αρχίζει το συγκεκριμένο πακέτο.

- 4) Η Packet Recognition μονάδα κάνει ανάγνωση της Βασικής Μνήμης με σκοπό να αναγνωρίσει το είδος του πακέτου το οποίο βρίσκεται στην συγκεκριμένη θέση. Μετά από έλεγχο των διάφορων πεδίων του πακέτου αναγνωρίζει ότι το πακέτο πρόκειται για Router Solicitation μήνυμα.
- 5) Ενημερώνει την Control Unit μέσω της διεπαφής feedback code ότι αναγνώρισε πακέτο Router Solicitation, συνεπώς πρέπει να κατασκευαστεί Router Advertisement πακέτο και να σταλθεί στην all-nodes multicast address ή στην διεύθυνση από την οποία προήλθε το Router Solicitation μήνυμα δηλαδή από την διεύθυνση του κινητού χρήστη.
- 6) Η Control Unit στέλνει στην μονάδα Packet Builder κωδικό “10” ώστε να επιλεγεί το κύκλωμα Router Advertisement Building Module.
- 7) Η μονάδα Packet Recognition δίνει τα κατάλληλα πεδία στην μονάδα Packet Builder ώστε να σχηματιστεί το πακέτο, συγκεκριμένα την mobile node address την οποία γνωρίζουμε από το μήνυμα Router Solicitation.
- 8) Η μονάδα Packet Recognition ενημερώνει την Control Unit ότι τελείωσε την διαδικασία της κατασκευής πακέτου Router Advertisement.
- 9) Η μονάδα Packet Builder από την στιγμή που ολοκληρωθεί η κατασκευή του πακέτου ενημερώνει μέσω του interface που έχει με το Data Link Layer ότι έχει διαθέσιμο πακέτο προς μεταφορά συγκεκριμένου μεγέθους.
- 10) Το Data Link Layer απαντά θετικά στην μεταφορά του πακέτου, δηλαδή ότι έχει διαθέσιμη μνήμη για αποθήκευση του πακέτου.
- 11) Η μονάδα Packet Builder από την στιγμή που πάρει θετική απάντηση αντιγράφει το πακέτο της 32 bits ανα clk στην διεπαφή προς Data Link Layer.
- 12) Όταν τελειώσει η μεταφορά του πακέτου στο Data Link Layer η μονάδα Packet Builder ενημερώνει ότι είναι διαθέσιμη για νέα επεξεργασία.
- 13) Η Control Unit ενημερώνει την μονάδα Control Memory ότι τελείωσε η επεξεργασία του πακέτου συγκεκριμένου μεγέθους blocks ώστε να ενημερώσει την κατάσταση μνήμης, ουσιαστικά θεωρώντας ότι μετά την επεξεργασία τα συγκεκριμένα blocks του πακέτου είναι διαθέσιμα για εγγραφή νέου πακέτου. Επίσης η Control Unit κάνει εγγραφή στο Log System της ότι πραγματοποιήθηκε επεξεργασία Router Advertisement με την εγγραφή συγκεκριμένου κωδικού στην κυκλική Log Memory (χωράει 256 events) στην θέση που υποδεικνύεται από το process_id **mod**(256) καθώς κάθε καινούργια διαδικασία που εκκινείται έχει ως χαρακτηριστικό αναγνώρισης ένα process_id.

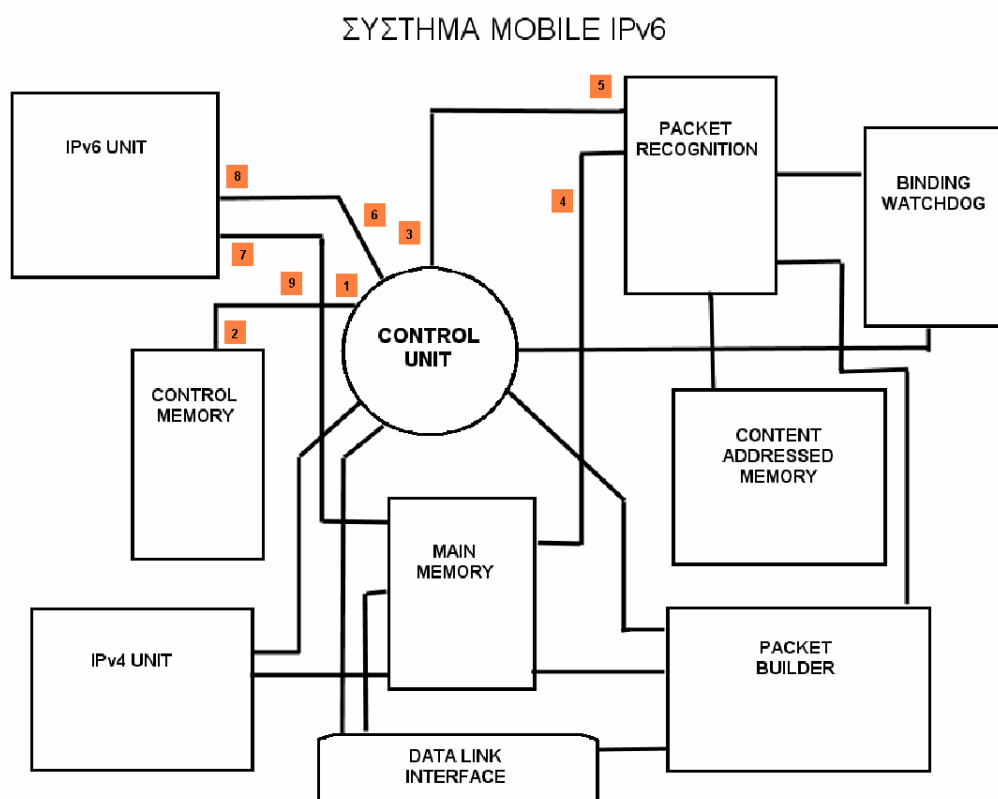
3.3.6 Packet Encapsulation



- 1) Η Control Unit υλοποιεί εσωτερικά κάποιο αλγόριθμο για τον προσδιορισμό της επόμενης εργασίας. Αν δεν έχει έρθει κάποιο interrupt από το Binding Watchdog ή δεν είναι σε αναμονή εγγραφή πακέτου από το Data Link Layer τότε εκκινεί την διαδικασία ανάγνωσης του επόμενου πακέτου. Κάνει λοιπόν αίτηση στην μονάδα Control Memory ποια είναι η διεύθυνση block του επόμενου πακέτου.
- 2) Η Control Memory απαντά στην Control Unit ότι τα δεδομένα προς επεξεργασία αρχίζουν από κάποιο συγκεκριμένο block της μνήμης, χωρίς να γνωρίζει το μέγεθος του πακέτου καθώς λειτουργεί σε υψηλότερο επίπεδο σε σχέση με την μονάδα Packet Recognition η οποία επεξεργαζόμενη το πακέτο στην Βασική Μνήμη αναγνωρίζει το μέγεθος του πακέτου.
- 3) Η μονάδα Control Unit στέλνει κωδικό εντολής εκκίνησης της διαδικασίας αναγνώρισης του επόμενου πακέτου, όπως επίσης και της διεύθυνση του block από την οποία αρχίζει το συγκεκριμένο πακέτο.
- 4) Η Packet Recognition μονάδα κάνει ανάγνωση της Βασικής Μνήμης με σκοπό να αναγνωρίσει το είδος του πακέτου το οποίο βρίσκεται στην συγκεκριμένη θέση.
- 5) Η Packet Recognition μονάδα κάνει ερώτηση στην CAM μνήμη αν το Destination Address αποτελεί κινητό χρήστη που υποστηρίζει το σύστημα.
- 6) Η CAM απαντά θετικά και ενημερώνει για το ποιος κινητός χρήστης είναι.
- 7) Η μονάδα Packet Recognition ζητά από την Control Unit την κατάσταση στην οποία βρίσκεται ο συγκεκριμένος χρήστης δηλαδή αν βρίσκεται στο home link ή σε κάποιο foreign link.

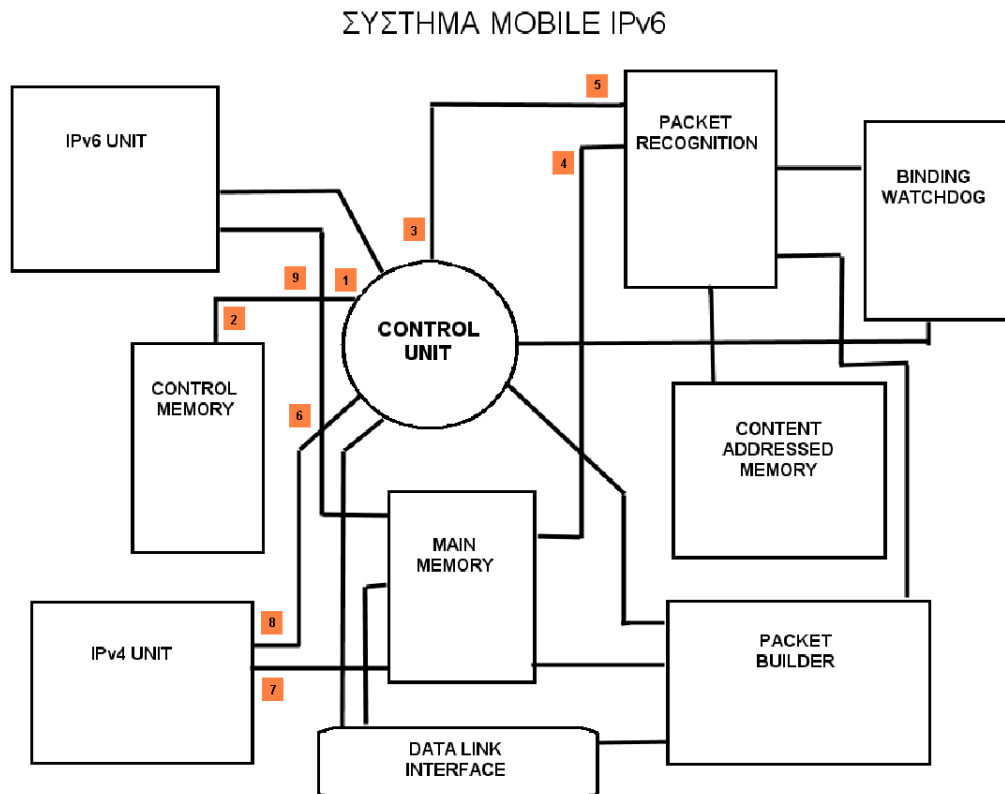
- 8) Η Control Unit απαντά στην Packet Recognition ότι ο κινητός χρήστης βρίσκεται σε κάποιο foreign link.
- 9) Η Packet Recognition ζητά από την CAM την Care-of-address του συγκεκριμένου κινητού χρήστη.
- 10) Η CAM στέλνει στο Packet Recognition την Care-of-Address του κινητού χρήστη.
- 11) Η Packet Recognition στέλνει τα κατάλληλα πεδία στην Packet Builder Μονάδα για να φτιαχτεί ο Encapsulation Header όπως επίσης την θέση και το μέγεθος του πακέτου προς encapsulation.
- 12) Η Packet Recognition μονάδα ενημερώνει την Control Unit για το μέγεθος του πακέτου σε blocks και ότι ολοκλήρωσε το έργο της.
- 13) Η Packet Encapsulation διαθέτει δική της διεπαφή με την Κύρια Μνήμη του συστήματος από την οποία σύμφωνα με τις υποδείξεις της Packet Recognition αντιγράφει το πακέτο στην εσωτερική της μνήμη.
- 14) Όταν τελειώσει η κατασκευή του πακέτου ενημερώνει το Data Link Layer ότι έχει πακέτο προς αποστολή συγκεκριμένου μεγέθους.
- 15) Το Data Link Layer απαντά θετικά στο request για μεταφορά.
- 16) Η μονάδα Packet Builder αντιγράφει 4 byte / clk στην διεπαφή της προς Data Link Layer μέχρι να στείλει όλο το πακέτο.
- 17) Η Packet Builder μονάδα ενημερώνει την Control Unit ότι τελείωσε την αποστολή και ότι είναι έτοιμη για την επόμενη επεξεργασία.
- 18) Η Control Unit ενημερώνει την Control Memory ότι τελείωσε η επεξεργασία του συγκεκριμένου πακέτου, συγκεκριμένου μεγέθους, ώστε να ενημερώσει την κατάσταση μνήμης. Επίσης η μονάδα γράφει στο LOG System της ότι ολοκληρώθηκε με επιτυχία Packet Encapsulation Process.

3.3.7 IPv6 Packet



- 1) Η Control Unit ζητά από την Control Memory την διεύθυνση αρχής του επόμενου πακέτου.
- 2) Η Control Memory απαντά στην Control Unit την διεύθυνση του block από το οποίο ξεκινά το επόμενο πακέτο προς επεξεργασία.
- 3) Η Control Unit στέλνει κωδικό αναγνώρισης του επόμενου πακέτου, μαζί με την διεύθυνση στην μονάδα Packet Recognition.
- 4) Η Packet Recognition κάνει ανάγνωση της μνήμης και ανακαλύπτει ότι το πακέτο αναφέρεται στις λειτουργίες του πρωτοκόλλου IPv6.
- 5) Η Packet Recognition ενημερώνει την Control Unit μέσω του feedback code ότι το πακέτο προορίζεται για την IPv6 μονάδα όπως επίσης και για το μέγεθος του πακέτου.
- 6) Η Control Unit ενημερώνει την μονάδα IPv6 ότι το πακέτο το οποίο αρχίζει από συγκεκριμένο block μνήμης και έχει συγκεκριμένο μέγεθος προορίζεται για αυτήν και ότι πρέπει να το αντιγράψει στην εσωτερική μνήμη της για περαιτέρω επεξεργασία.
- 7) Η μονάδα IPv6 κάνει ανάγνωση της κύριας μνήμης του συστήματος και αντιγράφει το πακέτο στην εσωτερική της μνήμη.
- 8) Όταν η μονάδα τελειώσει την μεταφορά ενημερώνει την Control Unit.
- 9) Η Control Unit ενημερώνει την Control Memory ότι το συγκεκριμένο πακέτο, συγκεκριμένου μεγέθους επεξεργάστηκε και μπορούμε τον χώρο να τον χρησιμοποιήσουμε για νέα εγγραφή πακέτων. Η Control Unit καταγράφει ότι επεξεργάστηκε πακέτο IPv6 στο LOG system της.

3.3.8 IPv4 Packet



- 1) Η Control Unit ζητά από την Control Memory την διεύθυνση αρχής του επόμενου πακέτου.
- 2) Η Control Memory απαντά στην Control Unit την διεύθυνση του block από το οποίο ξεκινά το επόμενο πακέτο προς επεξεργασία.
- 3) Η Control Unit στέλνει κωδικό αναγνώρισης του επόμενου πακέτου, μαζί με την διεύθυνση του στην μονάδα Packet Recognition.
- 4) Η Packet Recognition κάνει ανάγνωση της μνήμης και ανακαλύπτει ότι το πακέτο αναφέρεται στις λειτουργίες του πρωτοκόλλου IPv4.
- 5) Η Packet Recognition ενημερώνει την Control Unit μέσω του feedback code ότι το πακέτο προορίζεται για την IPv4 μονάδα όπως επίσης και για το μέγεθος του πακέτου.
- 6) Η Control Unit ενημερώνει την μονάδα IPv4 ότι το πακέτο το οποίο αρχίζει από συγκεκριμένο block μνήμης και έχει συγκεκριμένο μέγεθος προορίζεται για αυτήν και ότι πρέπει να το αντιγράψει στην εσωτερική μνήμη της για περαιτέρω επεξεργασία.
- 7) Η μονάδα IPv4 κάνει ανάγνωση της κύριας μνήμης του συστήματος και αντιγράφει το πακέτο στην εσωτερική της μνήμη.
- 8) Όταν η μονάδα τελειώσει την μεταφορά ενημερώνει την Control Unit.
- 9) Η Control Unit ενημερώνει την Control Memory ότι το συγκεκριμένο πακέτο, συγκεκριμένου μεγέθους επεξεργάστηκε και μπορούμε τον χώρο να τον χρησιμοποιήσουμε για νέα εγγραφή πακέτων. Η Control Unit καταγράφει ότι επεξεργάστηκε πακέτο IPv4 στο LOG system της.

3.4 Ανάλυση της λειτουργίας κάθε Μονάδας

3.4.1 Μονάδα Ελέγχου Μνήμης

Οι διεπαφές της μονάδας:

- Control Unit

Τα έργα της μονάδας:

- Διατήρηση της κατάστασης μνήμης με ενημερώσεις που δέχεται από την μονάδα ελέγχου του συστήματος μετά από διαδικασίες επεξεργασίας πακέτων που ολοκληρώνονται.
- Ενημέρωση της μονάδας ελέγχου για την ύπαρξη ή όχι χώρου για αποθήκευση πακέτων τα οποία φτάνουν από Data Link Layer.
- Ενημέρωση της μονάδας ελέγχου για την διεύθυνση της βασικής μνήμης στην οποία υπάρχει η αρχή του επόμενου πακέτου προς επεξεργασία.

Σχεδιασμός του Συστήματος Μνήμης

Η ύπαρξη MTU=1500 bytes σε πολλά links του διαδικτύου και αποτελέσματα στατιστικών μελετών σχετικά με το μέσο μέγεθος των πακέτων που ανταλλάσσονται στο διαδίκτυο μας οδήγησε στην δόμηση του συστήματος διαχείρισης μνήμης με βάση την κοντινότερη προς τα πάνω τιμή δύναμης του 2 δηλαδή 2048 bytes. Αυτή η ρύθμιση επιτρέπει την εισαγωγή πακέτων οποιουδήποτε μεγέθους το οποίο όμως αναλύεται σε **blocks των 2048 bytes**. Αν ένα πακέτο είναι μεγαλύτερο από αυτό το όριο καταλαμβάνει και επόμενα blocks από αυτό που υποδεικνύεται ως αρχή του πακέτου ενώ αν το μέγεθος του είναι μικρότερο κάποια από τα bytes που περιέχονται στο block θεωρούνται don't care καθώς το πακέτο επεξεργάζεται μέχρι το μέγεθός του το οποίο καθορίζεται από τον IP header και το πεδίο Payload. Η συγκεκριμένη επιλογή είναι ζωτικής σημασίας για το σύστημα καθώς κάνει πολύ πιο εύκολη την διαχείριση των πακέτων που επεξεργάζονται και την επικοινωνία με το Data Link Layer αλλά και την εσωτερική επικοινωνία μεταξύ των μονάδων του συστήματος. Επίσης κρίνεται βέλτιστη καθώς αποτελεί την πιο οικονομική επιλογή σε σχέση με το μέγεθος μνήμης μετά την διατήρηση πίνακα διευθύνσεων στις οποίες αρχίζει κάθε πακέτο σε επίπεδο Cell.

Η βασική μνήμη του συστήματος είναι **2 MB** τα οποία υπάρχουν με την μορφή 1024 blocks μνήμης που κάθε ένα έχει μέγεθος **2048 bytes**. Κάθε Cell μνήμης αποτελείται από 32 bit. Οι διευθύνσεις που ανταλλάσσονται μεταξύ των μονάδων του συστήματος αλλά και της Control Unit με το Data Link Layer είναι 10bit και καθορίζουν πάντα το block της αρχής του επόμενου πακέτου για ανάγνωση η το block από το οποίο πρέπει να αρχίσει να γράφει το Data Link Layer το νέο πακέτο.

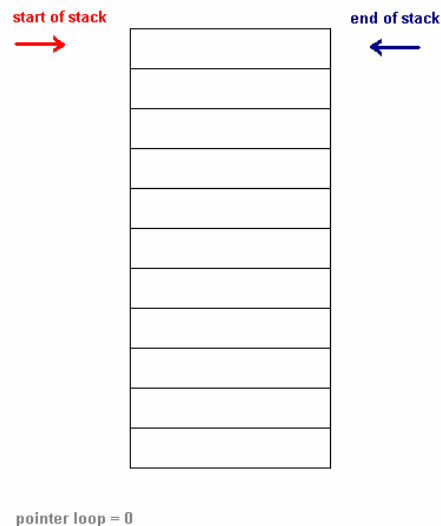
Η λειτουργία της μονάδας με περισσότερες λεπτομέρειες

Η μονάδα διαχείρισης της μνήμης βασίζεται στην ύπαρξη δύο δεικτών του **start_of_stack** και **end_of_stack**. Τα δεδομένα καθώς αποθηκεύονται στην μνήμη αυξάνεται ο δείκτης **end_of_stack** μέχρι να γεμίσει όλη η μνήμη. Όταν ένα πακέτο έχει περάσει από την επεξεργασία του συστήματος τότε ενημερώνεται η μονάδα και ο δείκτης **start_of_stack** αυξάνεται ανάλογα με το μέγεθος του πακέτου, αυτή η διαδικασία αντιστοιχεί με σβήσιμο των δεδομένων από την μνήμη καθώς τα ισχύοντα δεδομένα είναι εκείνα τα οποία βρίσκονται μεταξύ του δείκτη **end_of_stack** που προπορεύεται και του δείκτη **start_of_stack** που πάντα υπολείπεται και δείχνει πάντοτε την πρώτη θέση των ισχυόντων δεδομένων. Τα ισχύοντα δεδομένα βρίσκονται μεταξύ του δείκτη **start_of_stack** και **end_of_stack** όταν η μεταβλητή **pointer_loop** έχει τιμή 0 και μεταξύ **start_of_stack** και το τέλος της μνήμης μαζί με την περιοχή αρχής μνήμης μέχρι το **end_of_stack** στην περίπτωση που η μεταβλητή **pointer_loop** έχει τιμή 1. Τα παραπάνω εξηγούνται παραστατικά στα παρακάτω σχεδιαγράμματα. Δεν υπάρχει λόγος να διατηρεί η μονάδα κάποιο πίνακα με τα μεγέθη κάθε πακέτου που υπάρχει στην μνήμη καθώς ενημερώνεται για το μέγεθος των πακέτων που δίνει προς επεξεργασία από την μονάδα αναγνώρισης των πακέτων κάθε φορά που τελειώνει η επεξεργασία ενός πακέτου ώστε να ενημερώσει τους δυο **pointer** αλλά ενημερώνεται και κατά την αίτηση γραψίματος πακέτου από το **Data Link** για το μέγεθος του πακέτου το οποίο πρόκειται να γραφτεί στην κεντρική μνήμη ώστε να ενημερώσει τον **pointer start_of_stack**. **Το παραπάνω αποτελεί μια ιδέα που μοιάζει κατά πολύ την γενικότερη λειτουργία των δικτύων στα οποία υπάρχουν επίπεδα και στα οποία οι διεργασίες που γίνονται στο χαμηλότερο επίπεδο όπως αυτές που γίνονται από το **Data Link Layer** και **Packet Recognition** είναι διαφανής και παρέχονται ως υπηρεσία στο επίπεδο διαχείρισης μνήμης.** Τα παραπάνω θα γίνουν περισσότερα κατανοητά κατά την ανάλυση του συστήματος. Παρακάτω αναλύεται παραστατικά η ανανέωση των δεικτών **start_of_stack** και **end_of_stack** στους οποίους βασίζεται η λειτουργία της μονάδας σε μια υπόθεση εργασίας για το σύστημα διαχείρισης μνήμης.

Φάση 1^η

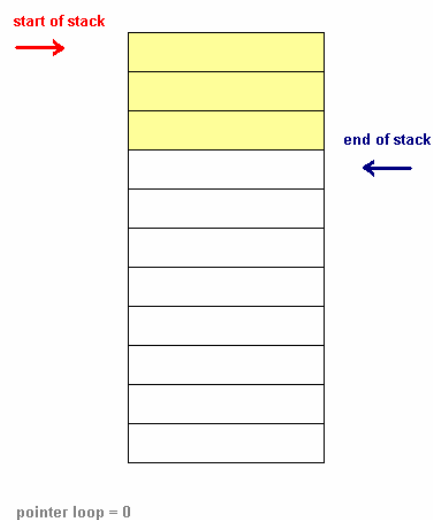
Μετά από reset στην αρχικοποίηση του συστήματος και οι δύο pointers δείχνουν στην θέση 0 ενώ η μεταβλητή pointer loop έχει τιμή 0. Το σύστημα περιέχει 0 blocks δεδομένων καθώς το μέγεθος των δεδομένων υπολογίζεται:

- end of stack – start of stack, για pointer loop = 0
- memory length – (start of stack – end of stack) , για pointer loop = 1



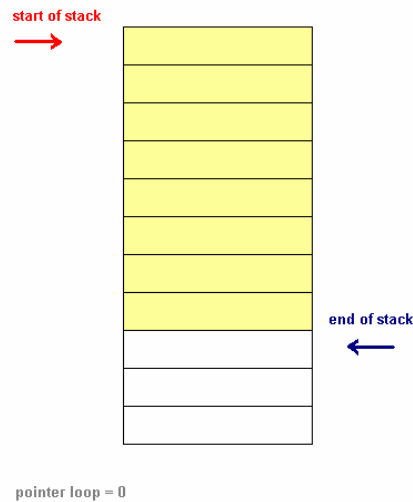
Φάση 2^η

Μετά την εγγραφή ενός πακέτου το οποίο καταλαμβάνει χώρο $2 < \text{μέγεθος} \leq 3$ block όπου κάθε block μνήμης όπως παρουσιάζεται αντιστοιχεί σε 2048 byte δηλαδή 64 memory cells από 32bit το κάθε ένα. Παρατηρούμε ότι ο pointer end of stack ενημερώνεται για την είσοδο των πακέτων στην μνήμη και ότι δείχνει πάντα την πρώτη διαθέσιμη θέση για εγγραφή.



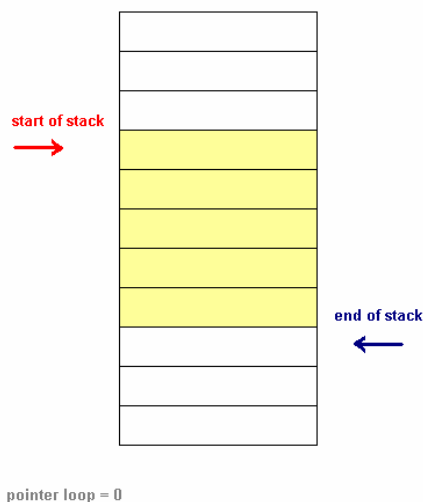
Φάση 3^η

Μετά την εγγραφή ενός πακέτου το οποίο καταλαμβάνει χώρο $4 < \text{μέγεθος} \leq 5$ block. Ο δείκτης end of stack αυξάνεται κατά 5 blocks ώστε να δείχνει το πρώτο διαθέσιμο block για εγγραφή δεδομένων. Η περιοχή από start of stack μέχρι end of stack η οποία είναι και χρωματισμένη είναι η περιοχή της μνήμης που περιέχει δεδομένα.



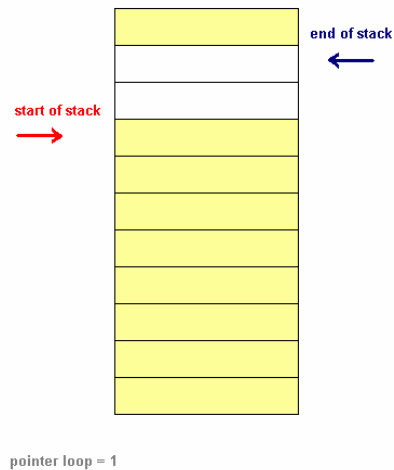
Φάση 4^η

Αφού έχει τελειώσει η επεξεργασία του πακέτου $2 < \text{μέγεθος} \leq 3$ block που γράφτηκε κατά την 2^η Φάση ενημερώνεται ο δείκτης start of stack και δείχνει την αρχή των valid δεδομένων.



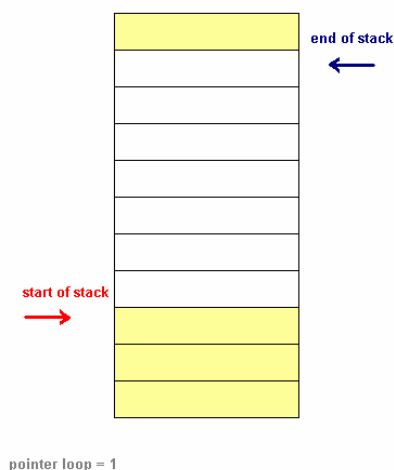
Φάση 5^η

Μετά την εγγραφή ενός πακέτου το οποίο καταλαμβάνει χώρο $3 < \text{μέγεθος} \leq 4$ block, ο δείκτης end of stack αρχίζει να μετράει από την αρχή και ενημερώνεται η μεταβλητή pointer loop σε 1 ώστε να υποδείξει ότι το μέγεθος της ελεύθερης μνήμης θα πρέπει να υπολογίζεται πλέον με διαφορετικό τρόπο από την στιγμή που ο δείκτης end of stack έχει πλέον μικρότερη τιμή από τον start of stack.



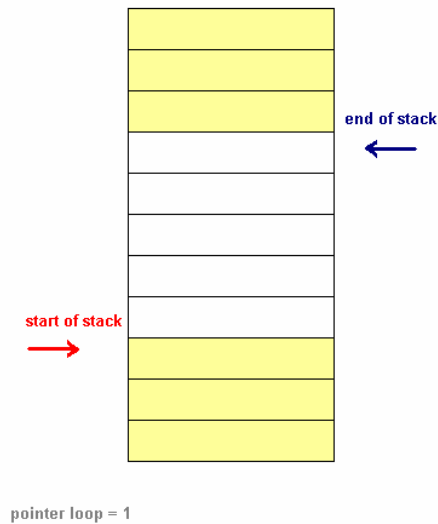
Φάση 6^η

Αφού έχει τελειώσει η επεξεργασία του πακέτου $4 < \text{μέγεθος} \leq 5$ block το οποίο γράφτηκε στην Φάση 3^η ενημερώνεται ο δείκτης start of stack και δείχνει την αρχή των valid δεδομένων.



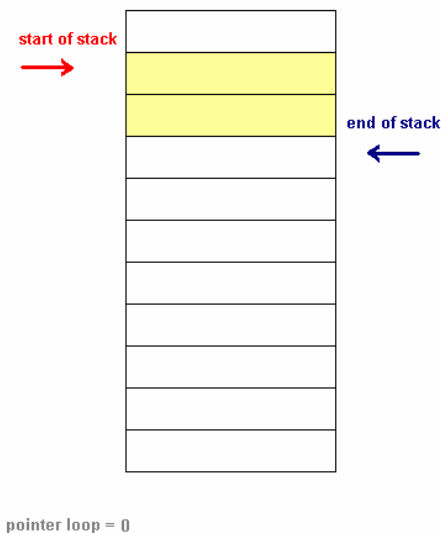
Φάση 7^η

Μετά την εγγραφή ενός πακέτου το οποίο καταλαμβάνει χώρο $1 < \text{μέγεθος} \leq 2$ block, ενημερώνεται ο δείκτης end of stack όπως δείχνει και η παρακάτω εικόνα.



Φάση 8^η

Αφού έχει τελειώσει η επεξεργασία ενός πακέτου $3 < \text{μέγεθος} \leq 4$ block το οποίο γράφτηκε κατά την Φάση 5^η ενημερώνεται ο δείκτης start of stack και δείχνει την αρχή των valid δεδομένων.



3.4.2 Μονάδα Αναγνώρισης Πακέτου

Οι διεπαφές της μονάδας:

- Control Unit
- Binding Watchdog Module
- Packet Builder Module
- Content Addressed Memory
- Main Memory

Περισσότερες λεπτομέρειες σχετικά με τις διεπαφές και τις αλληλεπιδράσεις την μονάδας με τις υπόλοιπες μονάδες του συστήματος.

Διεπαφές και Αλληλεπίδραση με Control Unit

Υπάρχει μια διεπαφή η οποία αποτελείται από μια συστάδα buses τα οποία αναλύονται παρακάτω:

- Bus Process Code (4 bit): IN από την Control Memory το οποίο δίνει εντολές στην μονάδα για το ποια είναι η λειτουργία η οποία θα πρέπει να εκτελέσει.
- Bus Feedback Code (5 bit): OUT προς την μονάδα Control Memory με την οποία την ενημερώνει σχετικά με τα αποτελέσματα αναγνώρισης των πακέτων, δηλαδή ενημερώνει π.χ αν το πακέτο που της ζητήθηκε να αναγνωρίσει είναι ένα πακέτο Binding Update και αν ο συγκεκριμένος κινητός χρήστης επιθυμεί να του αποσταλεί πίσω μήνυμα Binding Acknowledgement.
- Mobile Node Status (1 bit): IN μέσω του οποίου η Control Unit η οποία διατηρεί πίνακα με την κατάσταση κάθε mobile node δηλαδή αν βρίσκεται στο Home Link ή σε κάποιο Foreign Link.
- Bus Update Node (5 bit): IN μέσω του οποίου ζητά η Control Unit να γίνει δημιουργία και αποστολή πακέτου Binding Request επειδή δέχθηκε interrupt από την μονάδα Binding Lifetime ότι ο χρόνος εγκυρότητας του συγκεκριμένου κινητού χρήστη τελειώνει.
- Bus Start of data (10 bit): IN με το οποίο η Control Unit ενημερώνει την μονάδα αναγνώρισης πακέτου σε ποιο block της βασικής μνήμης αρχίζει το επόμενο πακέτο το οποίο της ζητά να επεξεργαστεί.
- Bus Packet Length (10 bit): OUT με το οποίο η μονάδα αναγνώρισης ενημερώνει την Control Unit η οποία με την σειρά της θα ενημερώσει την μονάδα Control Memory για να ενημερώσει τους δείκτες της μετά από κάθε επεξεργασία πακέτου που έχει τελειώσει.
- Bus Mobile Node (5 bit): OUT με το οποίο η μονάδα αναγνώρισης πακέτου ενημερώνει την Control Unit για την παρούσα κατάσταση του node του οποίου επεξεργάστηκε binding update πακέτο.

Διεπαφές και Αλληλεπίδραση με την CAM

Οι διεπαφές που υπάρχουν είναι ικανές να κάνουν ερωτήσεις για το αν υπάρχει μια συγκεκριμένη διεύθυνση στο πίνακα των globally routable addresses των κινητών χρηστών που εξυπηρετούνται ώστε να μπορέσει να διαβαστεί η care of address η οποία αντιστοιχεί στη συγκεκριμένη θέση μνήμης και να σταλθούν στην μονάδα packet encapsulation τα κατάλληλα δεδομένα για να γίνει η ενθυλάκωση του πακέτου το οποίο διαβάζει η μονάδα packet encapsulation απευθείας από την μονάδα μνήμης.

Επίσης η μονάδα είναι σε θέση να ενημερώνει τον πίνακα care of address ο οποίος περιέχεται στην μονάδα CAM μετά από μηνύματα Binding Update, ώστε το σύστημα Mobile IP να είναι πάντα ενήμερο για την θέση στην οποία βρίσκεται ο κινητός χρήστης κάθε φορά.

Αλληλεπίδραση με την Μονάδα Packet Builder

Η μονάδα περιέχει τέσσερις μονάδες οι οποίες κατασκευάζουν πακέτα και η μονάδα packet recognition είναι υπεύθυνη να δώσει τα κατάλληλα σήματα counter reset τα οποία είναι απαραίτητα ώστε τα δεδομένα που ακολουθούν να μπουν στα κατάλληλα πεδία του πακέτου. Η μονάδα packet recognition είναι σε θέση να γνωρίζει ανά πάσα στιγμή το είδος του δεδομένου που θα χρειαστεί η μονάδα packet builder ανάλογα με το είδος του πακέτου που πρέπει να κατασκευαστεί, καθώς κάθε είδος πακέτου έχει διαφορετικές απαιτήσεις ως προς τα δεδομένα τα οποία θα πρέπει να του στείλει η μονάδα.

Διεπαφές και αλληλεπίδραση με την μονάδα Binding Lifetime Watchdog

Η μονάδα μετά από κάθε μήνυμα binding update το οποίο δέχεται ενημερώνει το care of address table στην CAM μνήμη και την μονάδα Watchdog ως προς το Lifetime του binding.

Διεπαφή με την Κύρια Μνήμη του συστήματος

Η μονάδα όπως είναι φυσικό παρέχει διεπαφή ανάγνωσης της Κύριας Μνήμης του συστήματος καθώς θα πρέπει να είναι σε θέση να διαβάζει τα πακέτα από την κύρια μνήμη απευθείας και να παίρνει αποφάσεις ανάλογα με το είδος του πακέτου το οποίο επεξεργάζεται. Η μονάδα έχει εσωτερικά υλοποιημένες συναρτήσεις οι οποίες έχουν διευθυνσιοδότηση σε σχέση με την αρχή του πακέτου την οποία μας την παρέχει η μονάδα control memory όταν της ζητείται να δώσει την διεύθυνση του block στην οποία αρχίζει το επόμενο πακέτο προς επεξεργασία. Επίσης ανάλογα με τον τύπο του πακέτου γίνεται η αναγνώριση των πεδίων και η λήψη αποφάσεων σε σχέση με την μετέπειτα επεξεργασία.

Τα έργα της μονάδας:

- Η μονάδα έχει ως έργο να αναγνωρίζει το είδος του πακέτου που της ανατίθεται να αναγνωρίσει μέσω της διεύθυνσης και το process code που υποδεικνύει αναγνώριση πακέτου από την βασική μνήμη του συστήματος.
- Να στέλνει στην Control Unit το είδος του πακέτου που της ζητήθηκε να αναγνωρίσει, το μέγεθός του και το στάδιο στο οποίο βρίσκεται στην επεξεργασία του πακέτου.
- Να κάνει ερωτήσει στην CAM για να αναγνωρίσει αν το πακέτο ανήκει ή προορίζεται για κάποιο mobile node.
- Την ενημέρωση της μονάδας CAM μετά από κάθε επιτυχημένο Binding Update για την νέα care of address του mobile node και την διατήρηση του πίνακα ο οποίος βρίσκεται στην Control Unit και έχει την κατάσταση κάθε mobile node δηλαδή αν βρίσκεται στο Home Link ή σε κάποιο Foreign Link.
- Να παρέχει στην Μονάδα κατασκευής των πακέτων τα κατάλληλα σήματα (counter reset) και τα κατάλληλα δεδομένα για να σχηματιστούν τα πακέτα στις τέσσερις μονάδες κατασκευής πακέτων:

- 1) Router Advertisement Build Module
- 2) Binding Request Build Module
- 3) Binding Acknowledgement Build Module
- 4) Packet Encapsulation Build Module

- Να ενημερώσει τους πίνακες οι οποίοι υπάρχουν στην μονάδα Binding Lifetime Watchdog με τις τιμές τις οποίες διαβάζει από τα μηνύματα Binding Update. Εσωτερικά στην μονάδα Binding Lifetime Watchdog γίνεται μείωση των τιμών lifetime κάθε ένα sec και όταν η τιμή πέσει κάτω από ένα καθορισμένο όριο στέλνεται interrupt στην Control Unit ότι κάποιο κόμβος χρειάζεται να του αποσταλεί Binding Update Request.

Το έργο της συγκεκριμένης μονάδας είναι αρκετά εκτεταμένο καθώς αποτελεί νευραλγικό δομικό στοιχείο του συστήματός μας. Από το διάγραμμα ροής που ακολουθεί μπορούμε να διαπιστώσουμε πως έχουν υλοποιηθεί σχεδόν όλα τα έργα στο Mobile IP και τα Feedback Codes τα οποία αποστέλλει στην μονάδα Control Unit ώστε η τελευταία να έχει πάντα ενημέρωση για το ποια διεργασία εκτελεί το σύστημα, να στέλνει τα κατάλληλα σήματα στην μονάδα Packet Builder και να ενημερώνει το LOG SYSTEM το οποίο διαθέτει για να παρακολουθεί τις διαδικασίες του συστήματος που λειτουργεί όπως στα συστήματα UNIX σαν το ιστορικό του συστήματος. Η συγκεκριμένη μονάδα λόγω του εκτεταμένου χαρακτήρα της και τις πολλές αρμοδιότητες που διαθέτει έχει υλοποιηθεί με 2000 γραμμές κώδικα.

Στο σχήμα που ακολουθεί φαίνεται η μηχανή κατάστασης την οποία υλοποιεί η μονάδα. Πρέπει να τονισθεί ότι η μηχανή κατάστασης της μονάδας επικοινωνεί με την μηχανή κατάστασης της Control Unit. Η επικοινωνία μεταξύ των δυο καταστάσεων γίνεται για να αναθέσει η Control Unit έργα στην μονάδα Packet Recognition αλλά και για να δεχθεί feedback για το είδος των πακέτων τα οποία αναγνωρίστηκαν και για την πορεία της επεξεργασίας των πακέτων. Είναι σημαντικό να αναφέρουμε τους κωδικούς που ανταλλάσσονται κατά την διάρκεια της επικοινωνίας των δυο μονάδων και της σημασίας που έχουν. Στην αρχή της κάθε δήλωσης αναφέρετε ο κωδικός ο οποίος ανταλλάσσεται και στο τέλος η κατάσταση κατά την οποία γίνεται η ανταλλαγή του κωδικού.

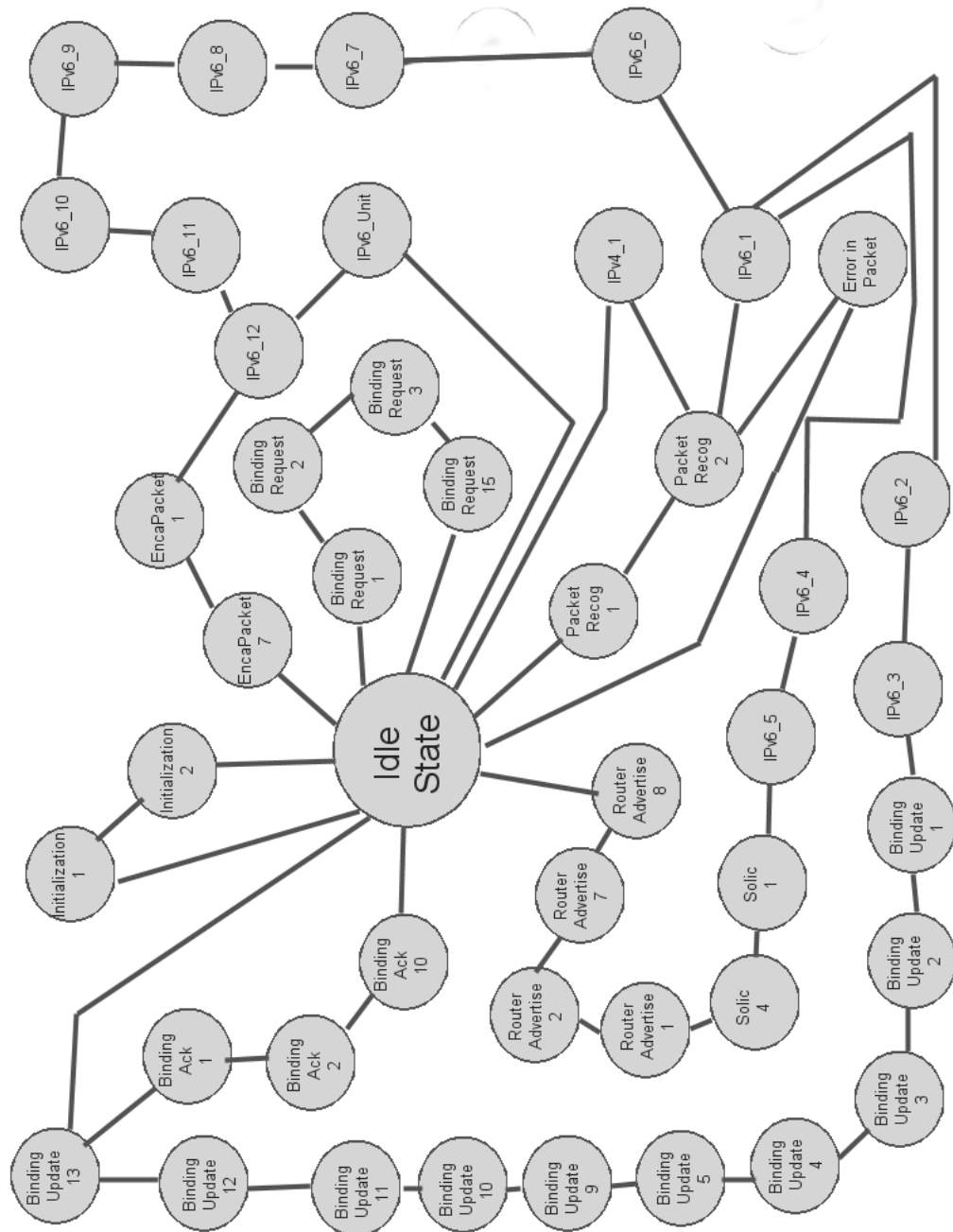
Control Unit σε Packet Recognition Μονάδα (**Process Code** – 4 bit)

- “**1010**” Εκίνησε Διαδικασία αναγνώρισης νέου πακέτου το οποίο ξεκινά από την διεύθυνση που υποδεικνύει η κατάλληλη διεπαφή (**Idle State**)
- “**0100**” Εκκίνηση διαδικασία δημιουργίας Binding Request πακέτου (**Idle State**).

Packet Recognition σε Control Unit Μονάδα (**Feedback Code** – 5 bit)

- “**00011**” Λάθος στην έκδοση του πακέτου (**Error in Packet Version**).
- “**01000**” Το πακέτο είναι IPv4 και πρέπει να το διαχειριστεί η IPv4_Unit (**IPv4_1**).
- “**01001**” Το πακέτο είναι IPv6 και δεν περιέχει κάποια πληροφορία η οποία να φορά το Mobile IPv6 οπότε πρέπει να διαχειριστεί από την IPv6_Unit (**IPv6_12**).
- “**00101**” Η διαδικασία Binding Update χωρίς Acknowledgement τελείωσε επιτυχώς. (**Binding Update 13**).
- “**00110**” Το πακέτο αναγνωρίστηκε ως Binding Update με Acknowledgement και η μονάδα ελέγχου πρέπει να δώσει τα κατάλληλα σήματα ελέγχου στην μονάδα κατασκευής πακέτων για την δημιουργία του πακέτου Acknowledgement (**Binding Update 13**).
- “**00111**” Η διαδικασία Binding Update με Acknowledgement τελείωσε επιτυχώς. (**Binding Acknowledgement 10**)
- “**01100**” Ο κινητός χρήστης βρίσκεται σε κάποιο foreign link, ώστε να ενημερωθεί ο εσωτερικός πίνακας καταστάσεως ο οποίος βρίσκεται στην Control Unit (**Binding Update 13**).
- “**01101**” Ο κινητός χρήστης επέστρεψε στο Home Link του (**Binding Update 13**).
- “**00010**” Η διαδικασία δημιουργίας Binding Request τελείωσε επιτυχώς. (**Binding Request 15**).
- “**01010**” Η διαδικασία ενθυλάκωσης του πακέτου που προοριζόταν για τον κινητό χρήστη ο οποίος βρίσκεται σε κάποιο foreign link τελείωσε επιτυχώς (**Packet Encapsulation 7**).
- “**01110**” Η μονάδα διέγνωσε ότι πρέπει να κατασκευαστεί και να αποσταλεί πακέτο Router Advertisement και η Control Unit πρέπει να συμβάλλει σε αυτό δίνοντας τα κατάλληλα σήματα στην μονάδα κατασκευής πακέτων (**Router Solicitation 4**).
- “**01111**” Η διαδικασία αποστολής μηνύματος Router Advertisement τελείωσε επιτυχώς (**Router Advertisement 8**).

Στο παρακάτω σχήμα δείχνεται συνοπτικά η μηχανή κατάστασης την οποία υλοποιεί η μονάδα αναγνώρισης των πακέτων.



3.4.3 Control Unit

Όπως μπορούμε να φανταστούμε η μονάδα αποτελεί την πλέον νευραλγική μονάδα η οποία γνωρίζει ανά πάσα στιγμή την κατάσταση της επεξεργασίας, το είδος της επεξεργασίας η οποία πραγματοποιείται, επικοινωνεί με το κατώτερο επίπεδο Data Link και συντονίζει την διαδικασία της εισαγωγής των πακέτων στην Κύρια Μνήμη, δίνει τα κατάλληλα σήματα τα οποία είναι απαραίτητα για να πραγματοποιηθούν τα διάφορα έργα, κρατάει σε εσωτερική μνήμη την κατάσταση κάθε κινητού χρήστη δηλαδή το αν βρίσκεται στο Home Link ή σε κάποιο Foreign Link, κρατάει το ιστορικό των διεργασιών τις οποίες έχει φέρει σε πέρας το σύστημα καθώς αυτές είναι δυναμικές και εξαρτώνται από το περιεχόμενο των πακέτων τα οποία φτάνουν από το Data Link Layer.

Οι διεπαφές της μονάδας:

Η μονάδα όπως είναι φυσικό επικοινωνεί με όλες τις υπομονάδες του συστήματος εκτός από την Main Memory και Content Addressed Memory.

- Packet Recognition
- Binding Watchdog Module
- Packet Builder Module
- Data Link Layer
- Control Memory
- IPv6 Module
- IPv4 Module

Περισσότερες λεπτομέρειες σχετικά με τις διεπαφές και τις αλληλεπιδράσεις την μονάδας με τις υπόλοιπες μονάδες του συστήματος.

Packet Recognition

Όπως περιγράψαμε και στο προηγούμενο κομμάτι σχετικά με την μονάδα αναγνώρισης πακέτων, η σχέση των δύο μονάδων είναι πολύ στενή καθώς ανταλλάσσουν μηνύματα η μια για να καθορίσει τα έργα τα οποία πρέπει να πραγματοποιηθούν και η άλλη για να ενημερώσει για την ροή της επεξεργασίας και των έργων των οποίων πραγματοποιούνται. Στο προηγούμενο κεφάλαιο έχει γίνει εκτενής αναφορά στους κωδικούς οι οποίοι ανταλλάσσονται και στην σημασία την οποία αυτοί έχουν.

Binding Watchdog

Η μονάδα όταν διαπιστώνει ότι το lifetime κάποιου κινητού χρήστη το οποίο μειώνεται κάθε 1 δευτερόλεπτο πέσει κάτω από συγκεκριμένη τιμή alarm ειδοποιεί την Control Unit με ένα σήμα διακοπής Interrupt ώστε η τελευταία να εκκινήσει διαδικασία δημιουργίας πακέτου Binding Request. Πρέπει να σημειώσουμε ότι η ενημέρωση των Lifetimes στη μονάδα γίνεται από την μονάδα Packet Recognition γι αυτό άλλωστε υπάρχει και διεπαφή μεταξύ τους.

Packet Builder Module

Η σχέση με την μονάδα κατασκευής πακέτων είναι πολύ σημαντική. Επειδή κατά τον σχεδιασμό υπήρχε το αίτημα όσο το δυνατόν λιγότερων interfaces – buses μεταξύ των μονάδων ώστε να αποτελεί ο σχεδιασμός ένα ρεαλιστικό κατασκευάσμα εσωτερικά στην μονάδα υπάρχει μια μονάδα πολυπλέκτη και μια μονάδα αποπλέκτη ώστε να υπάρχει 1 interface με την μονάδα Packet Recognition και 1 με το Data Link Layer για όλες τις μονάδες κατασκευής πακέτων. Όπως είναι φυσικό οι μονάδες mux και decoder χρειάζονται κάποιο σήμα επιλογής της εισόδου και της εξόδου το οποίο το παρέχει η μονάδα Control Unit.

Control Memory

Η μονάδα Control Unit είναι η μοναδική η οποία μπορεί να κάνει ερωτήσεις στην μονάδα Control Memory και η μοναδική η οποία μπορεί να την ενημερώσει. Αυτό έγινε για να υπάρχει συνοχή στην μνήμη καθώς η μονάδα Control Memory αν δεχόταν ερωτήματα και ενημερώσεις από όλες τις μονάδες οι οποίες έχουν πρόσβαση στην Κύρια Μνήμη θα χρειαζόταν να αναπτυχθεί ειδικό πρωτόκολλο διατήρησης της συνοχής της μνήμης, με μια λογική κατανεμημένου συστήματος. Οι μονάδες που έχουν πρόσβαση στην 5-port μνήμη είναι οι παρακάτω:

Εγγραφή

- Data Link Layer

Ανάγνωση

- Packet Recognition
- Packet Builder (Encapsulation Sub-Module)
- IPv4 Unit
- IPv6 Unit

IPv4

Η μονάδα Control Unit είναι εκείνη η οποία ενημερώνει την μονάδα IPv4 (μετά από ενημέρωση μέσω του feedback code της Packet Recognition μονάδας) για την θέση στην κεντρική μνήμη και το μέγεθος του πακέτου το οποίο έχει αναγνωριστεί ώστε η μονάδα IPv4 να το αντιγράψει στην εσωτερική της μνήμη για περαιτέρω επεξεργασία σύμφωνα με τις λειτουργίες του πρωτοκόλλου το οποίο έχει υλοποιηθεί εσωτερικά στην μονάδα IPv4_Unit.

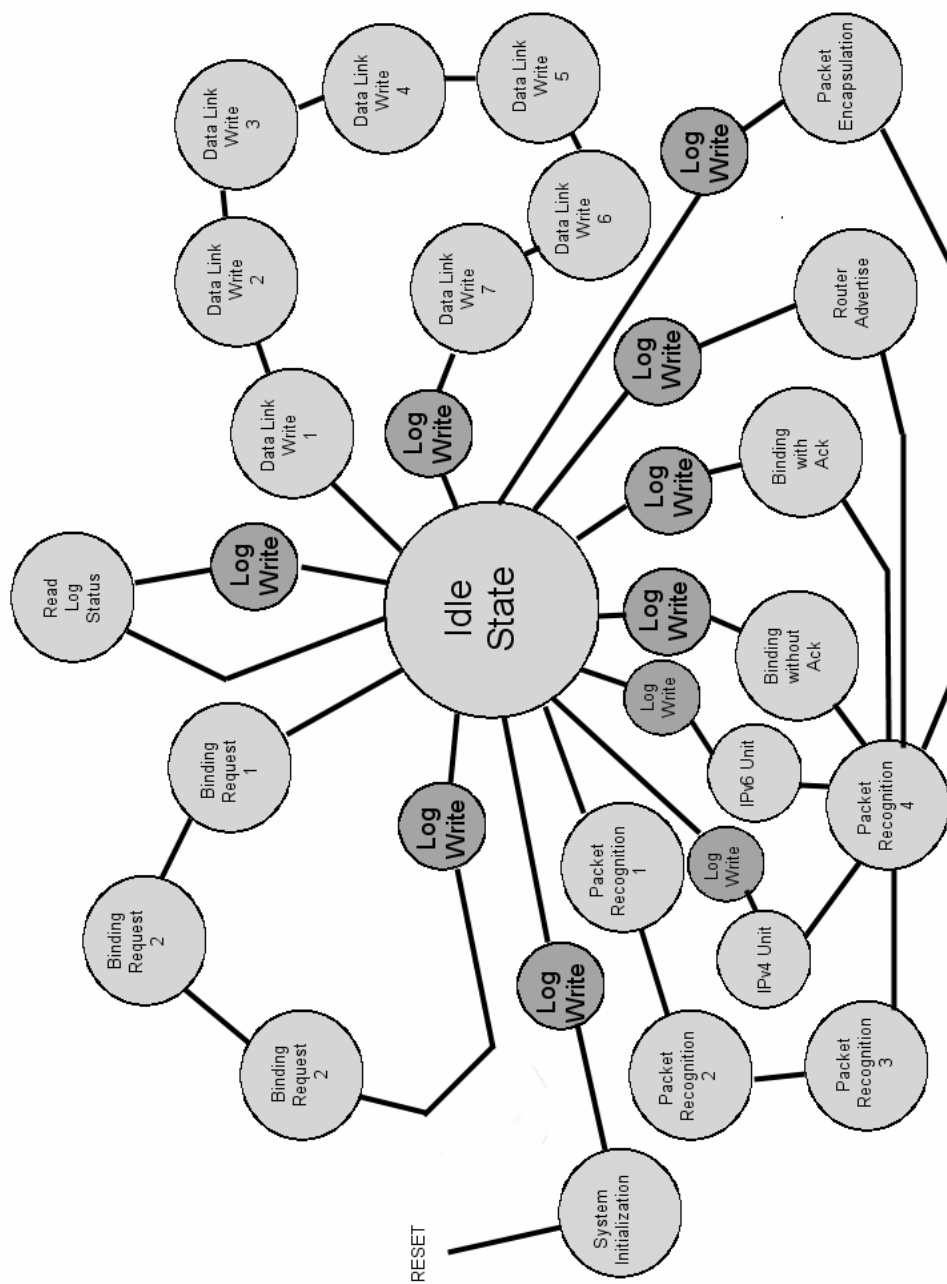
IPv6

Η μονάδα Control Unit είναι εκείνη η οποία ενημερώνει την μονάδα IPv6 (μετά από ενημέρωση μέσω του feedback code της Packet Recognition μονάδας) για την θέση στην κεντρική μνήμη και το μέγεθος του πακέτου το οποίο έχει αναγνωριστεί ώστε η μονάδα IPv6 να το αντιγράψει στην εσωτερική της μνήμη για περαιτέρω επεξεργασία σύμφωνα με τις λειτουργίες του πρωτοκόλλου το οποίο έχει υλοποιηθεί εσωτερικά στην μονάδα IPv6_Unit. Η Control Memory λειτουργεί με τον παραπάνω τρόπο όταν το πακέτο δεν αποτελεί κάποιο μήνυμα το οποίο έχει σχέση με το πρωτόκολλο Mobile IPv6 και αποτελεί πακέτο το οποίο θα πρέπει να το δρομολογήσει το κλασσικό πρωτόκολλο IPv6.

Data Link Layer

Όταν το χαμηλότερο επίπεδο Data Link Layer επιθυμεί να στείλει κάποιο πακέτο στο επίπεδο IP τότε κάνει ερώτηση αν μπορεί να εγγράψει στην Κύρια Μνήμη ένα πακέτο συγκεκριμένου μεγέθους. Όταν η Control Unit δεχθεί αυτή την αίτηση κάνει ερώτηση στην μονάδα Control Memory η οποία είναι υπεύθυνη της διαχείρισης μνήμης και στέλνει ανάλογα με την απάντηση που δέχεται πίσω στο Data Link. Η απάντηση της μονάδας Control Memory μπορεί να είναι είτε αρνητική που σημαίνει ότι η Κύρια Μνήμη δεν χωράει το καινούργιο πακέτο, συνεπώς η μεταφορά από το Data Link Layer θα πρέπει να περιμένει είτε θα απαντήσει θετικά και θα δώσει και την διεύθυνση του block στο οποίο θα πρέπει να αποθηκευτεί το νέο πακέτο.

Ακολουθεί η μηχανή κατάστασης την οποία υλοποιεί η κεντρική μονάδα. Δεν παρουσιάζονται όλες οι λειτουργίες της μέσω αυτής της μηχανής καθώς υπάρχουν και sub components όπως η εσωτερική μνήμη της κατάστασης του κινητού χρήστη. Η δομή της μηχανής κατάστασης δίνει την αίσθηση ότι η Control Unit αποτελεί το υψηλότερο Layer αν μπορούσαν αυτά να υπάρχουν στην υλοποίηση του πρωτοκόλλου Mobile IP καθώς είναι εκείνη η οποία θα θέσει σε προτεραιότητα τις διάφορες διαδικασίες οι οποίες υπάρχουν προς εκτέλεση ανάλογα με την σημαντικότητα της εργασίας. Η μονάδα δεν “γνωρίζει” με ποιο τρόπο έρχονται σε πέρας οι διάφορες διεργασίες οι οποίες υλοποιούνται ούτε τις πολύ μικρές λεπτομέρειες για το πως κατασκευάζονται τα πακέτα, αυτές είναι διεργασίες που πρέπει να γνωρίζουν μονάδες που βρίσκονται χαμηλότερα ιεραρχικά όπως είναι η μονάδα Packet Recognition. Η μονάδα ενημερώνεται όπως βλέπουμε πάντα για την κατάσταση στην οποία βρίσκεται το σύστημα και με την χρήση του συστήματος καταγραφής των διεργασιών κρατάει το ιστορικό του συστήματος από την αρχικοποίηση του συστήματος μέχρι κάποια στιγμή.



3.4.4 Content Addressed Memory

Η μονάδα μνήμης είναι βασική σε κάθε τηλεπικοινωνιακό σύστημα το οποίο περιλαμβάνει δρομολόγηση και γενικά σε όλες τις εφαρμογές στις οποίες υπάρχει ανάγκη για μεταγλώττιση ενός στοιχείο σε ένα άλλο. Κάθε δομή αυτού του είδους μνήμης περιλαμβάνει 2 μνήμες Ram. Στην πρώτη Ram γράφεις το στοιχείο προς μεταγλώττιση και στην δεύτερη στην αντίστοιχη θέση την μεταγλώττιση του στοιχείου.

Οι διεπαφές της μονάδας:

- Packet Recognition

Τα έργα της μονάδας

- Διατηρεί Mobile Node Home Address στον πίνακα A και Care of Address στον πίνακα B.
- Απαντά στις ερωτήσεις την μονάδας Packet Recognition για το αν υπάρχει στο πίνακα A, δηλαδή στον πίνακα με τις διευθύνσεις κινητών χρηστών που εξυπηρετούνται κάποια διεύθυνση και αν υπάρχει σε ποια θέση βρίσκεται αυτή.
- Ενημερώνει τον πίνακα B με την καινούργια Care of Address του κινητού χρήστη αν πρόκειται για διαδικασία Binding Update ή δίνει την παρούσα Care of Address του κινητού χρήστη σε περίπτωση που πρόκειται για διαδικασία Packet Encapsulation.

3.4.5 Binding Watchdog

Τα Bindings των κινητών χρηστών με τον Home Agent έχουν κάποια διάρκεια που μετά το πέρας της σταματά να υφίσταται το binding καθώς θεωρείτε από το πρωτόκολλο ότι δεν μπορεί να εμπιστευτεί ο Home Agent πληροφορίες παρωχημένες να ενθυλακώσει και να αποστείλει το πακέτο σε κάποια διεύθυνση Foreign Link η οποία δεν είναι σίγουρος ότι ισχύει. Γι αυτό το λόγο υπάρχει η συγκεκριμένη μονάδα η οποία αποτελείτε από δυο υπομονάδες. Η μια δέχεται το Lifetime για κάποιο node για το οποίο γίνεται εγγραφή και μειώνει την τιμή του Lifetime για κάθε κόμβο κάθε 1 sec. Η άλλη υπομονάδα δίνει ένα παλμό με περίοδο 1 sec κάτι σαν κρύσταλλος με αυτή την συχνότητα.

3.4.6 Μονάδα Κατασκευής Πακέτων

Το πρωτόκολλο Mobile IPv6 πρέπει να έχει την δυνατότητα να κατασκευάζει 4 ειδών πακέτα. Αυτά είναι τα:

- Router Advertisement
- Binding Request
- Binding Acknowledgement
- Packet Encapsulation

Κάθε ένα από τα παραπάνω είδη πακέτων περιλαμβάνει κάποια σταθερά πεδία τα οποία είναι ίδια σε κάθε πακέτο που κατασκευάζεται και κάποια τα οποία θα πρέπει να τα παρέχει η μονάδα Packet Recognition η οποία έχει άμεση σχέση με την Κύρια Μνήμη και κάνει αναγνώριση των δεδομένων τα οποία περιέχονται σε κάποιο πακέτο. Στον σχεδιασμό δόθηκε βάση στην μείωση των διεπαφών μεταξύ των μονάδων και γι αυτό το λόγο όλες οι μονάδες κατασκευάστηκαν με παρόμοια λογική για να μπορούν να έχουν όλες κοινές διεπαφές με το υπόλοιπο σύστημα και για να μπορέσουμε να ομαδοποιήσουμε αυτές τις διεπαφές σε μια με χρήση κυκλωμάτων mux και decoder και κάποιο σημάτων οι οποίες κάνουν έλεγχο των παραπάνω κυκλωμάτων. Η διαδικασία για την κατασκευή των πακέτων για κάθε μονάδα περιγράφεται παρακάτω:

- 1) Η μονάδα ελέγχου δίνει κάποιο σήμα Control ώστε μέσω του decoder στην είσοδο του συστήματος τα σωστά σήματα να πάνε από την μοναδική είσοδο στην μονάδα στην είσοδο της σωστής μονάδας (counter_reset, data).
- 2) Η μονάδα Packet Recognition μηδενίζει τους counters ώστε να ακολουθήσει η διαδικασία τροφοδότησης της μονάδας με τα δεδομένα με τα οποία χτίζεται κάθε πακέτο. Ανάλογα με το είδος του πακέτου υπάρχει μια σύμβαση για το ποια δεδομένα πρέπει να σταλούν σε κάθε παλμό ρολογιού.
- 3) Όταν τελειώσει η κατασκευή κάθε πακέτου η αντίστοιχη μονάδα κατασκευής μέσω του κυκλώματος mux στέλνει στην μοναδική έξοδο του συστήματος προς το Data Link Layer ότι υπάρχει διαθέσιμο πακέτο για μεταφορά μαζί με το μέγεθος του πακέτου.
- 4) Όταν το Data Link Layer ανταποκριθεί σε αυτή την handshaking επικοινωνία τότε η εκάστοτε μονάδα κατασκευής πακέτου αναλαμβάνει να μεταβιβάσει το πακέτο από την εσωτερική μνήμης της στο Data Link Layer δίνοντας 32 bit σε κάθε παλμό ρολογιού.
- 5) Όταν τελειώσει η διαδικασία μεταφοράς του πακέτου στο Data Link Layer η μονάδα ενημερώνει την Control Unit ότι είναι έτοιμη για επεξεργασία νέου πακέτου.
- 6) Η Control Unit όταν δεχθεί σήμα ότι ολοκληρώθηκε η μεταφορά του πακέτου στο Data Link Layer ενημερώνει την μονάδα Control Memory ότι το συγκεκριμένο πακέτο, το οποίο έχει συγκεκριμένο μέγεθος επεξεργάστηκε ώστε να ενημερώσει την κατάσταση μνήμης, να ελευθερωθεί η μνήμη στην οποία αντιστοιχούσε το πακέτο το οποίο επεξεργάστηκε και να υπάρχει διαθέσιμος χώρος για εγγραφή νέων πακέτων στην Βασική Μνήμη του συστήματος.

Υπάρχει μια ιδιαιτερότητα σχετικά με την μονάδα Encapsulation καθώς έχει ξεχωριστές διεπαφές με το σύστημα (Packet Recognition, Main Memory). Η συγκεκριμένη μονάδα παίρνει τις πληροφορίες που χρειάζεται για την κατασκευή του encapsulation header από την μονάδα Packet Recognition, μαζί με την διεύθυνση block στην οποία βρίσκεται το πακέτο προς ενθυλάκωση και το μέγεθός του και από εκεί και πέρα κάνει ανεξάρτητα ανάγνωση από την Βασική Μνήμη και δημιουργεί το ενθυλακωμένο πακέτο. Αυτή η μονάδα έχει υλοποιημένη εσωτερικά μια πολύ μεγαλύτερη μνήμη RAM καθώς το μέγεθος του πακέτου προς ενθυλάκωση μπορεί να διαφέρει σημαντικά. Όταν η μονάδα τελειώσει την διαδικασία της ενθυλάκωσης στέλνει σήμα στο Data Link Layer μέσω της ξεχωριστής της διεπαφής ότι έχει πακέτο προς μεταφορά και όταν το τελευταίο ανταποκριθεί αναλαμβάνει να διαβάσει την εσωτερική της μνήμη και να μεταφέρει το πακέτο στο χαμηλότερο επίπεδο.

Οι διεπαφές της μονάδας:

- Packet Recognition
- Control Unit
- Main Memory
- Data Link Interface (Basic)
- Data Link Interface (Packet Encapsulation)

Τα σήματα τα οποία προέρχονται από την Control Unit τα οποία ρυθμίζουν τα κυκλώματα mux και decoder είναι τα εξής:

- “00” Binding Acknowledgement Building Module.
- “01” Binding Request Building Module.
- “10” Router Advertisement Building Module.

Με τα παραπάνω γίνεται η επιλογή των κυκλωμάτων τα οποία συμμετέχουν ενώ τα υπόλοιπα μένουν ανενεργά όταν δεν έχουν επιλεγθεί.

3.4.7 Βασική Μνήμη του συστήματος

Η Βασική μνήμη του συστήματος είναι μια RAM η οποία έχει την ιδιαιτερότητα να έχει 2 ειδών interfaces. Παρέχει 1 διεπαφή για εγγραφή δεδομένων τα οποία έρχονται από το Data Link Layer και 4 διεπαφές για ανάγνωση των πακέτων από μονάδες που επεξεργάζονται και διαβάζουν τα πακέτα.

Οι διεπαφές της μονάδας:

- Data Link Layer (Write)
- Packet Recognition (Read)
- Packet Builder (Read) – Encapsulation Building Module
- IPv4 Unit (Read)
- IPv6 Unit (Read)

Το μέγεθος της μνήμης

Η Βασική Μνήμη έχει μέγεθος 2MB και αποτελείται από 65536 cells των 32 bit. Λογικά όπως έχουμε διατυπώσει και στην ανάλυση της μονάδας Control Memory υπάρχει μια δομή που αποτελείται από 1024 blocks των 2048 bytes.

3.4.8 IPv6 Unit

Όταν το σύστημα αναγνωρίσει ότι δεν υπάρχει σχέση του πακέτου που αναγνωρίζεται με το πρωτόκολλο Mobile IPv6 τότε η Control Unit ειδοποιεί την μονάδα ότι το πακέτο που αρχίζει από συγκεκριμένη θέση στην μνήμη και έχει συγκεκριμένο μέγεθος πρέπει να το διαχειριστεί η μονάδα IPv6. Όταν η μονάδα λάβει την παραπάνω ειδοποίηση μεταφέρει το πακέτο κάνοντας ανάγνωση της κεντρικής μνήμης στην εσωτερική της μνήμη για περαιτέρω επεξεργασία. Στην συγκεκριμένη υλοποίηση καθώς δεν ήταν ο στόχος μας να αναπτύξουμε το IPv6 η συγκεκριμένη μονάδα αποτελεί μια ψευδομονάδα, δηλαδή δεν έχει εσωτερικές λειτουργίες εκτός από το να ειδοποιεί την Control Unit ότι μετέφερε το πακέτο ένα κύκλο ρολογιού μετά την στιγμή που δέχεται την εντολή, ώστε και η μονάδα Control Unit να ειδοποιήσει την μονάδα Control Memory η οποία διατηρεί την κατάσταση της μνήμης.

3.4.9 IPv4 Unit

Όταν το σύστημα αναγνωρίσει ότι το πακέτο το οποίο αναγνωρίζει είναι ένα IPv4 πακέτο, τότε η Control Unit ενημερώνει την μονάδα IPv4 Unit να μεταφέρει το συγκεκριμένο πακέτο στην εσωτερική της μνήμη για περαιτέρω επεξεργασία. Σε αυτή την μονάδα όπως συνέβη και με την IPv6 Unit δεν υλοποιήθηκαν όλες οι λειτουργίες του πρωτοκόλλου IPv4 καθώς κάτι τέτοιο θα ήταν πέρα από τις απαιτήσεις αυτής της διπλωματικής εργασίας. Το σύστημα το οποίο σχεδιάστηκε παρέχει κατάλληλη διεπαφή ώστε να μπορεί να ενσωματωθεί σε αυτή την μονάδα η υλοποίηση του συγκεκριμένου πρωτοκόλλου και να επιτευχθεί με αυτό τον τρόπο η ομαλή συνύπαρξη διαφορετικών υλοποιήσεων οι οποίες θα αποτελούν το IP Layer.

Κεφάλαιο 4^ο

Η Σύνθεση του Συστήματος

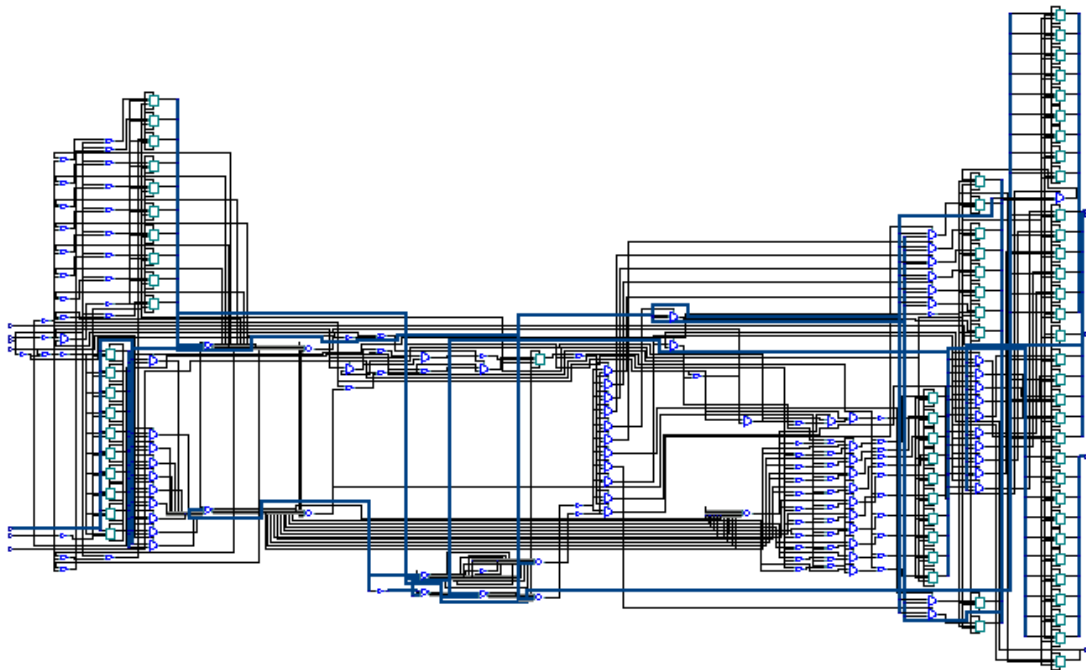
4.1 Εισαγωγή

Σε αυτό το κεφάλαιο παρουσιάζουμε τα κυκλώματα όπως τα Συνέθεσε το πρόγραμμα Leonardo Spectrum Level 3. Στο δεύτερο μέρος του κεφαλαίου παρουσιάζονται τα αποτελέσματα των συνθέσεων και των υλοποιήσεων με συγκεκριμένα FPGA.

4.2 Τα σχηματικά και τα κυκλώματα

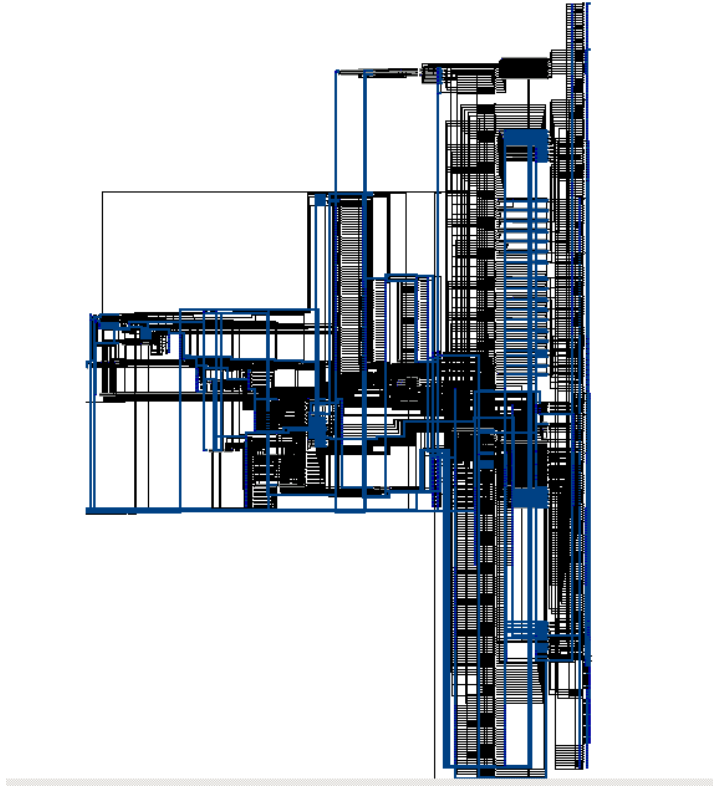
4.2.1 Control Memory

Η μονάδα Control Memory όπως την συνέθεσε το πρόγραμμα Leonardo Spectrum.



4.2.2 Packet Recognition

Η μονάδα Packet Recognition όπως την συνέθεσε το Leonardo Spectrum.



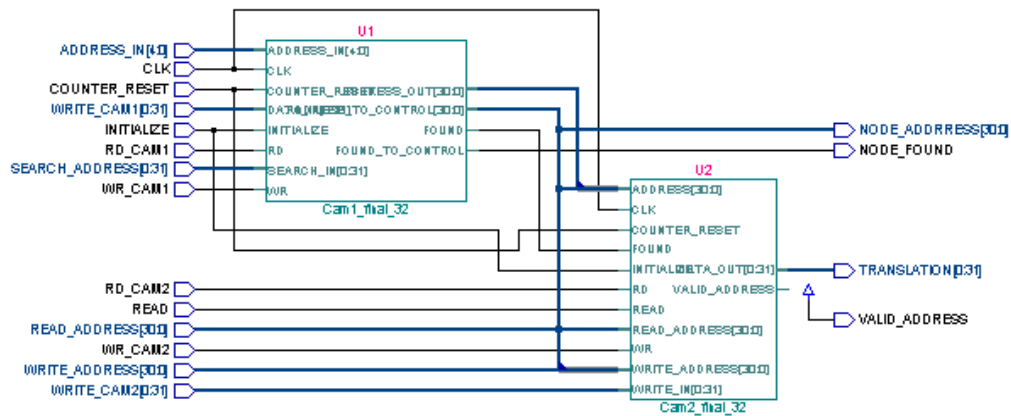
4.2.3 Control Unit

Ακολουθεί το ψηφιακό σύστημα της Control Unit όπως το συνέθεσε το πρόγραμμα Leonardo Spectrum.



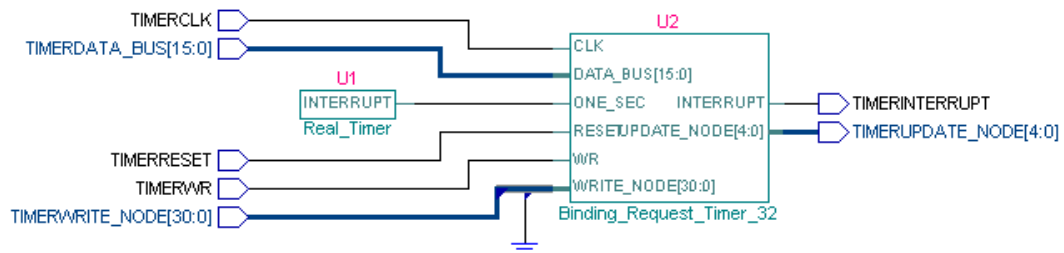
4.2.4 Content Addressed Memory

Το σχηματικό της μονάδας όπως παρουσιάζεται στο Leonardo φαίνεται παρακάτω:



4.2.5 Binding Watchdog

Το κύκλωμα όπως φαίνεται από την σύνθεση του Leonardo Spectrum είναι το παρακάτω:

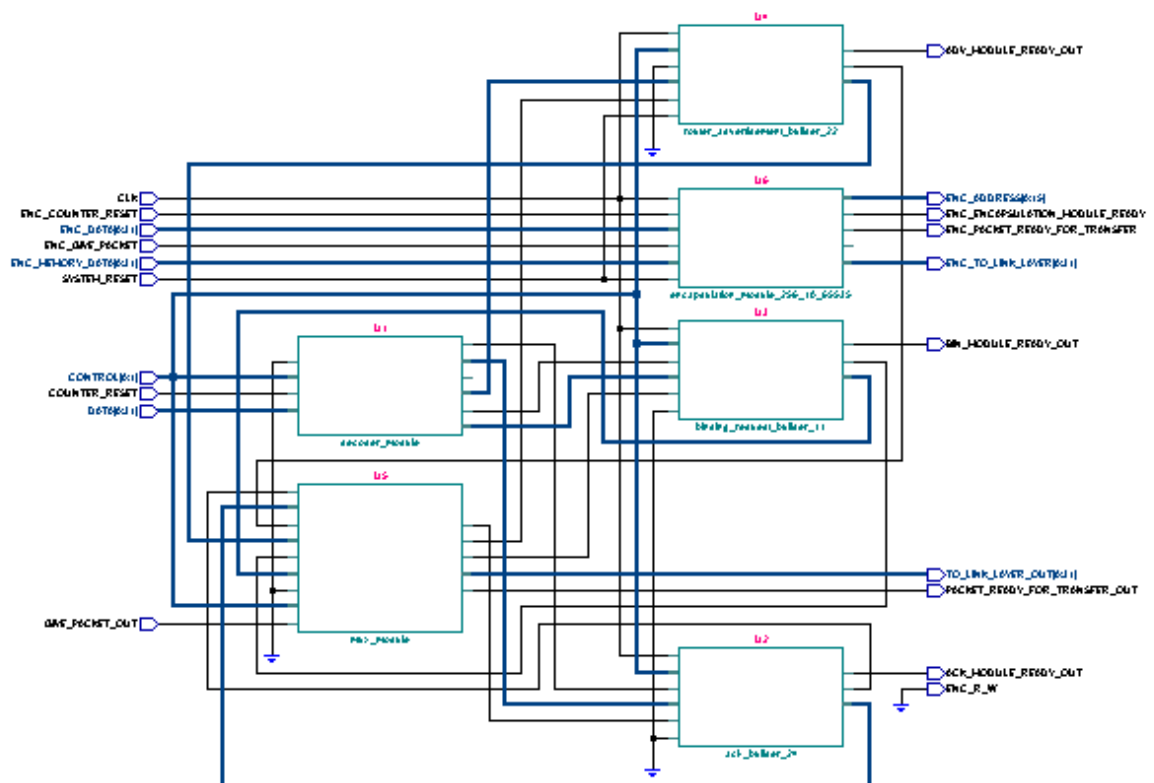


Το κύκλωμα το οποίο συνέθεσε το πρόγραμμα Leonardo Spectrum φαίνεται παρακάτω:



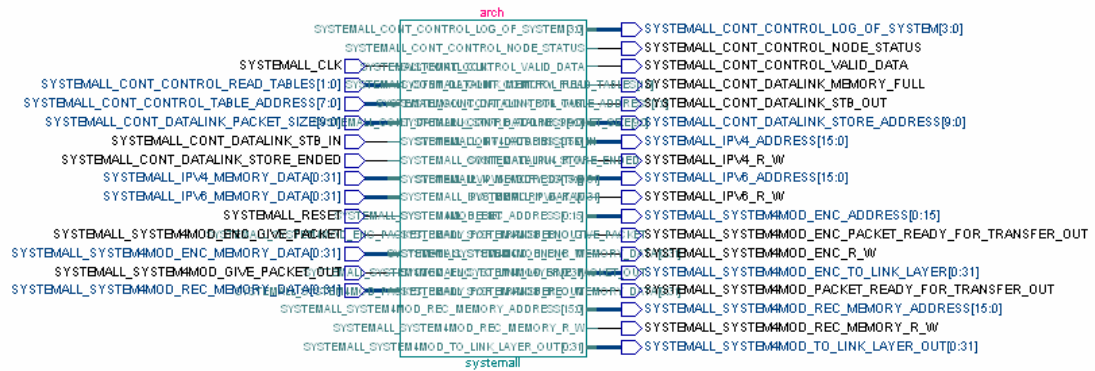
4.2.6 Packet Builder

Το σχηματικό της μονάδας όπως προκύπτει στο πρόγραμμα σύνθεσης Leonardo φαίνεται παρακάτω:

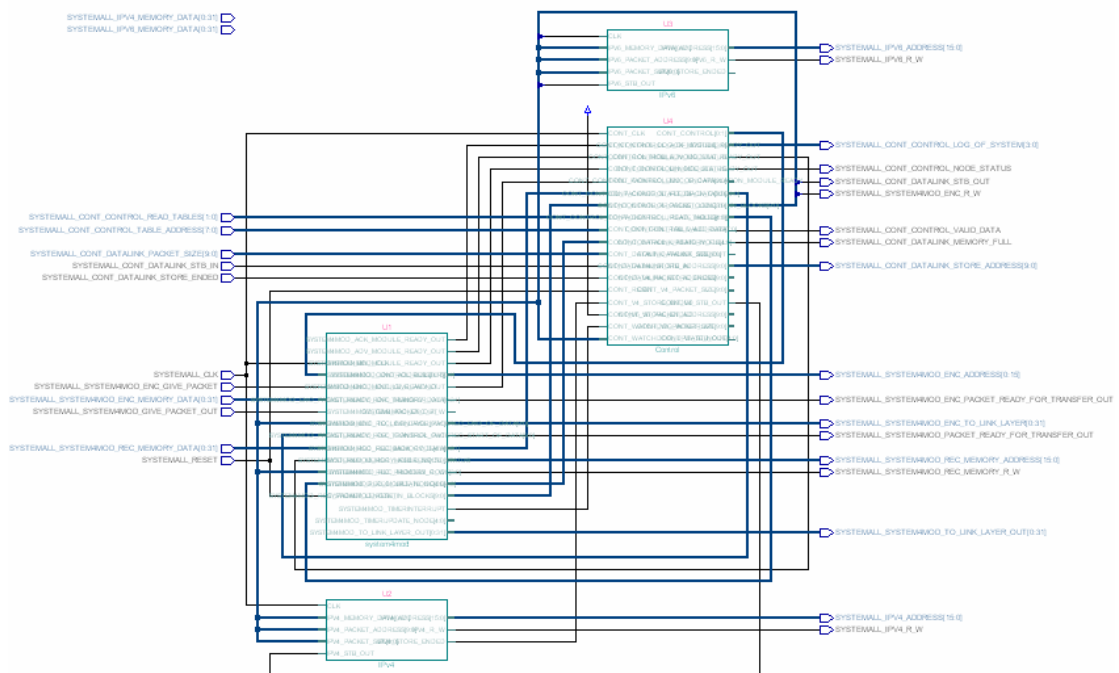


4.2.9 System

Το παρακάτω σχηματικό όπως το συνέθεσε το πρόγραμμα Leonardo Spectrum ανήκει στο **συνολικό σύστημα**:



Αν κατέβουμε ένα επίπεδο παίρνουμε το επόμενο σχηματικό:



Οι 4 μονάδες που παρατηρούμε είναι οι:

- IPv6
- IPv4
- RecCamBuldeWatch-System4mod (Packet Recognition, CAM, Packet Builder, Binding Watchdog
- Control (Control Unit, Control Memory)

4.3 Αποτελέσματα Συνθέσεων

4.3.1 Control Memory

```
*****
Cell: control_memory_final    View: design    Library: work
*****
```

```
Total accumulated area :
Number of BUFGP :                1
Number of Dffs or Latches :      63
Number of Function Generators :  134
Number of GND :                   1
Number of IBUF :                  15
Number of MUX CARRYs :           74
Number of MUXF5 :                  1
Number of OBUF :                  32
Number of VCC :                    1
Number of XORCY :                 50
Number of gates :                135

Number of ports :                  48
Number of nets :                  388
Number of instances :             372
Number of references to this view : 0
```

```
*****
Device Utilization for 2V250fg256
*****
Resource                Used    Avail    Utilization
-----
IOs                      48      172      27.91%
Function Generators     134     3072      4.36%
CLB Slices               67     1536      4.36%
Dffs or Latches         63     3588      1.76%
-----
```

Using wire table: xcv2-250-5_avg

Clock Frequency Report

```
Clock                : Frequency
-----
CLK                   : 65.8 MHz
```

4.3.2 Control Unit

```
*****
Cell: ControlUnit      View: design      Library: work
*****

Total accumulated area :
  Number of BUFGP :                1
  Number of Dffs or Latches :      1192
  Number of Function Generators :   1517
  Number of GND :                   1
  Number of IBUF :                   81
  Number of MUX CARRYs :            12
  Number of MUXF5 :                 362
  Number of MUXF6 :                 132
  Number of OBUF :                   75
  Number of OBUFT :                 21
  Number of VCC :                    1
  Number of XORCY :                  8
  Number of gates :                 1479

  Number of ports :                 184
  Number of nets :                  3485
  Number of instances :             3403
  Number of references to this view :    0
```

```
*****
Device Utilization for 2V1500bg575
*****
Resource                Used      Avail      Utilization
-----
IOs                      184       392       46.94%
Function Generators     1517     15360       9.88%
CLB Slices               759      7680       9.88%
Dffs or Latches         1192     16536       7.21%
-----
```

Using wire table: xcv2-1500-5_avg

Clock Frequency Report

```
Clock                : Frequency
-----
CLK                  : 60.5 MHz
```

4.3.3 Control (Control Unit, Control Memory)

Cell: Control View: arch Library: work

Total accumulated area :

Number of BUFGP :	1
Number of Dffs or Latches :	1229
Number of Function Generators :	1555
Number of GND :	1
Number of IBUF :	50
Number of MUX CARRYs :	86
Number of MUXF5 :	328
Number of MUXF6 :	132
Number of OBUF :	60
Number of OBUFT :	32
Number of VCC :	1
Number of XORCY :	58
Number of gates :	1518

Number of ports :	148
Number of nets :	3584
Number of instances :	3533
Number of references to this view :	0

Device Utilization for 2V250fg256

Resource	Used	Avail	Utilization
IOs	148	172	86.05%
Function Generators	1555	3072	50.62%
CLB Slices	778	1536	50.65%
Dffs or Latches	1229	3588	34.25%

Using wire table: xcv2-250-5_avg

Clock Frequency Report

Clock	: Frequency
CONT_CLK	: 61.7 MHz

4.3.4 Packet Builder (Encapsulation Building Module, Router Advertisement Building Module, Binding Request Building Module, Binding Acknowledgement Building Module, Mux, Decoder).

```
*****
Cell: packet_builder_32      View: arch      Library: work
*****

Total accumulated area :
Number of BUFG :                4
Number of BUFGP :               1
Number of Dffs or Latches :    9286
Number of Function Generators : 12361
Number of GND :                2
Number of IBUF :               103
Number of MULT_AND :           31
Number of MUX CARRYs :         422
Number of MUXF5 :              4542
Number of MUXF6 :              2072
Number of OBUF :               87
Number of VCC :                2
Number of XORCY :              262
Number of gates :              12066

Number of ports :               191
Number of nets :               2740
Number of instances :          2587
Number of references to this view : 0

*****
Cell: encapsulation_module_256_10_65535      View: module_unfold_1777
Library: work
*****

Total accumulated area :
Number of Dffs or Latches :    8480
Number of Function Generators : 11014
Number of GND :                1
Number of MULT_AND :           31
Number of MUX CARRYs :         332
Number of MUXF5 :              4489
Number of MUXF6 :              2072
Number of VCC :                1
Number of XORCY :              169
Number of gates :              10725

Number of ports :               119
Number of nets :               26657
Number of instances :          26589
Number of references to this view : 1
```

```

*****
Device Utilization for 2V250fg256
*****
Resource                Used      Avail    Utilization
-----
IOs                      191      172      111.05%
Function Generators     12361    3072     402.38%
CLB Slices              6181    1536     402.41%
Dffs or Latches        9286    3588     258.81%
-----

```

This design does not fit in the device specified!
Trying an alternate device ...

```

*****
Device Utilization for 2V1500bg575
*****
Resource                Used      Avail    Utilization
-----
IOs                      191      392      48.72%
Function Generators     12361    15360     80.48%
CLB Slices              6181    7680     80.48%
Dffs or Latches        9286    16536     56.16%
-----

```

Clock Frequency Report

```

Clock                  : Frequency
-----
CLK                    : 37.6 MHz

```

4.3.5 CAM (CAM1, CAM2)

Cell: Cam128bit_32 View: Cam_stuct Library: work

Total accumulated area :

Number of BUFG :	6
Number of BUFGP :	1
Number of Dffs or Latches :	8737
Number of Function Generators :	18768
Number of GND :	2
Number of IBUF :	118
Number of MUX CARRYs :	2048
Number of MUXF5 :	2119
Number of OBUF :	65
Number of VCC :	3
Number of gates :	18100

Number of ports :	236
Number of nets :	349
Number of instances :	188
Number of references to this view :	0

Cell: Cam1_final_32 View: mem_unfold_978 Library: work

Total accumulated area :

Number of BUFG :	6
Number of Dffs or Latches :	4376
Number of Function Generators :	6527
Number of GND :	1
Number of MUX CARRYs :	2048
Number of MUXF5 :	1
Number of VCC :	1
Number of gates :	6292

Number of ports :	138
Number of nets :	13034
Number of instances :	12960
Number of references to this view :	1

Cell: Cam2_final_32 View: mem_unfold_979 Library: work

Total accumulated area :

Number of Dffs or Latches :	4361
Number of Function Generators :	12241
Number of MUXF5 :	2118
Number of VCC :	1
Number of gates :	11808

Number of ports :	165
Number of nets :	18775
Number of instances :	18721
Number of references to this view :	1

Device Utilization for 2V1500bg575

Resource	Used	Avail	Utilization
IOs	236	392	60.20%
Function Generators	18768	15360	122.19%
CLB Slices	9384	7680	122.19%
Dffs or Latches	8737	16536	52.84%

This design does not fit in the device specified!
Trying an alternate device ...

Device Utilization for 2V2000bg575

Resource	Used	Avail	Utilization
IOs	236	408	57.84%
Function Generators	18768	21504	87.28%
CLB Slices	9384	10752	87.28%
Dffs or Latches	8737	22728	38.44%

Using wire table: xcv2-1500-5_avg

Clock Frequency Report

Clock	: Frequency
CLK	: 47.5 MHz
RD_CAM1	: 49.7 MHz
WR_CAM1	: Infinity

4.3.6 Binding Watchdog

Cell: Watchdog_32 View: arch Library: work

Total accumulated area :

Black Box AND2 :	530
Black Box DFFERS :	519
Black Box FALSE :	1
Number of Function Generators :	512
Black Box INV :	651
Black Box MUX :	69
Number of MUX CARRYs :	512
Black Box OR2 :	1770
Black Box TRUE :	1
Number of XORCY :	512

Number of ports :	56
Number of nets :	5069
Number of instances :	5077
Number of references to this view :	0

Device Utilization for 2V250fg256

Resource	Used	Avail	Utilization
IOs	56	172	32.56%
Function Generators	512	3072	16.67%
CLB Slices	256	1536	16.67%
Dffs or Latches	376	3588	10.40%

Clock Frequency Report

Clock	: Frequency
-------	-------------

CLK	: 60.5 MHz
-----	------------

4.3.7 Packet Recognition Module

```
*****
Cell: RecogPhasel_32_65536_32    View: arch    Library: work
*****
```

```
Total accumulated area :
Number of BUFGP :          1
Number of Dffs or Latches :    682
Number of Function Generators : 1091
Number of GND :              1
Number of IBUF :             118
Number of MUX CARRYs :        173
Number of MUXF5 :             63
Number of OBUF :              268
Number of OBUFT :             53
Number of VCC :               1
Number of XORCY :             110
Number of gates :            1092

Number of ports :            440
Number of nets :             2680
Number of instances :        2561
Number of references to this view :    0
```

```
Device Utilization for 2V250fg256
*****
Resource                Used      Avail    Utilization
-----
IOs                      440       172     255.81%
Function Generators      1091      3072     35.51%
CLB Slices               546      1536     35.55%
Dffs or Latches          682      3588     19.01%
```

```
-----
This design does not fit in the device specified!
Trying an alternate device ...
```

```
*****
Device Utilization for 2V2000bf957
*****
Resource                Used      Avail    Utilization
-----
IOs                      440       624     70.51%
Function Generators      1091     21504     5.07%
CLB Slices               546     10752     5.08%
Dffs or Latches          682     23376     2.92%
```

Clock Frequency Report

```
Clock                : Frequency
-----
```

```
CLK                  : 54.0 MHz
```

4.3.8 IPv4 Unit

Η εμφανής διαφορά στον χρονισμό της μονάδας οφείλεται στο γεγονός ότι η μονάδα αποτελεί ψευδομονάδα, δηλαδή το έργο της είναι να τραβήξει το αντίστοιχο πακέτο οποίο έχει αναγνωριστεί ως IPv4 στην εσωτερική μνήμη, χωρίς να υλοποιεί τις λειτουργίες του IPv4 καθώς αυτό θα ήταν έξω από τις απαιτήσεις της συγκεκριμένης διπλωματικής εργασίας.

Cell: IPv4 View: module Library: work

Total accumulated area :

Number of BUFGP :	1
Number of Dffs or Latches :	1
Number of GND :	1
Number of IBUF :	1
Number of OBUF :	1
Number of OBUFT :	17
Number of VCC :	1
Number of gates :	0

Number of ports :	72
Number of nets :	25
Number of instances :	23
Number of references to this view :	0

Device Utilization for 2V250fg256

Resource	Used	Avail	Utilization
IOs	72	172	41.86%
Function Generators	0	3072	0.00%
CLB Slices	1	1536	0.07%
Dffs or Latches	1	3588	0.03%

Using wire table: xcv2-250-5_avg

Clock Frequency Report

Clock	: Frequency
-------	-------------

CLK	: 383.0 MHz
-----	-------------

4.3.9 IPv6 Unit

Η εμφανής διαφορά στον χρονισμό της μονάδας οφείλεται στο γεγονός ότι η μονάδα αποτελεί ψευδομονάδα, δηλαδή το έργο της είναι να τραβήξει το αντίστοιχο πακέτο οποίο έχει αναγνωριστεί ως IPv6 στην εσωτερική μνήμη, χωρίς να υλοποιεί τις λειτουργίες του IPv6 καθώς αυτό θα ήταν έξω από τις απαιτήσεις της συγκεκριμένης διπλωματικής εργασίας.

Cell: IPv6 View: module Library: work

Total accumulated area :

Number of GND :	1
Number of OBUF :	1
Number of OBUFT :	17
Number of VCC :	1
Number of gates :	0

Number of ports :	72
Number of nets :	20
Number of instances :	20
Number of references to this view :	0

Device Utilization for 2V250fg256

Resource	Used	Avail	Utilization
IOs	72	172	41.86%
Function Generators	0	3072	0.00%
CLB Slices	0	1536	0.00%
Dffs or Latches	0	3588	0.00%

Using wire table: xcv2-250-5_avg

Clock Frequency Report

Clock	: Frequency
-------	-------------

CLK	: 385.0 MHz
-----	-------------

4.3.10 RecCamBuildWatch (Packet Recognition, CAM, Packet Builder, Binding Watchdog)

Cell: system4mod View: arch Library: work

Total accumulated area :

Black Box AND2 :	1243
Black Box DFFERS :	1664
Black Box DFFRS :	7
Number of Dffs or Latches :	12934
Black Box FALSE :	1
Number of Function Generators :	23646
Number of GND :	1
Black Box INV :	1058
Black Box LATRS :	274
Number of MULT_AND :	34
Black Box MUX :	800
Number of MUX CARRYs :	1157
Number of MUXF5 :	6607
Number of MUXF6 :	2072
Black Box OR2 :	1944
Black Box TRI :	16
Black Box TRUE :	1
Number of VCC :	2
Number of XORCY :	899
Number of gates :	22533
Black Box xmplr_lut_10 :	13
Black Box xmplr_lut_11 :	28
Black Box xmplr_lut_12 :	21
Black Box xmplr_lut_13 :	1
Black Box xmplr_lut_14 :	1
Black Box xmplr_lut_15 :	32
Black Box xmplr_lut_3 :	2
Black Box xmplr_lut_4 :	13
Black Box xmplr_lut_6 :	1
Black Box xmplr_lut_7 :	1
Black Box xmplr_lut_8 :	33
Black Box xmplr_lut_9 :	36

Number of ports :	230
Number of nets :	9783
Number of instances :	9668
Number of references to this view :	0

Cell	Library	References		Total Area	
AND2	PRIMITIVES	1243 x	1	1243	AND2
Cam2_final_32	work	1 x	11808	11808	gates
			1	1	VCC
			12063	12063	Function
Generators			2118	2118	MUXF5
			4361	4361	Dffs or
Latches					
DFFERS	PRIMITIVES	1664 x	1	1664	DFFERS
DFFRS	PRIMITIVES	7 x	1	7	DFFRS
FALSE	PRIMITIVES	1 x	1	1	FALSE
FDRE	xcv2	93 x	1	93	Dffs or
Latches					
INV	PRIMITIVES	1058 x	1	1058	INV
LATRS	PRIMITIVES	274 x	1	274	LATRS
LUT1_L	xcv2	197 x	1	197	Function
Generators					
LUT2_L	xcv2	22 x	1	22	Function
Generators					
LUT4_L	xcv2	606 x	1	606	Function
Generators					
MULT_AND	xcv2	3 x	1	3	MULT_AND
MUX	PRIMITIVES	800 x	1	800	MUX
MUXCY	xcv2	50 x	1	50	MUX CARRYs
MUXCY_L	xcv2	775 x	1	775	MUX CARRYs
OR2	PRIMITIVES	1944 x	1	1944	OR2
TRI	PRIMITIVES	16 x	1	16	TRI
TRUE	PRIMITIVES	1 x	1	1	TRUE
XORCY	xcv2	730 x	1	730	XORCY
encapsulation_module_256_10_65535	work			1 x	10725
10725 gates					
			1	1	GND
			1	1	VCC
			2072	2072	MUXF6
			4489	4489	MUXF5
			8480	8480	Dffs or
Latches					
			169	169	XORCY
			332	332	MUX CARRYs
Generators			10758	10758	Function
			31	31	MULT_AND
xmplr_lut_10	OPERATORS	13 x	1	13	xmplr_lut_10
xmplr_lut_11	OPERATORS	28 x	1	28	xmplr_lut_11
xmplr_lut_12	OPERATORS	21 x	1	21	xmplr_lut_12
xmplr_lut_13	OPERATORS	1 x	1	1	xmplr_lut_13
xmplr_lut_14	OPERATORS	1 x	1	1	xmplr_lut_14
xmplr_lut_15	OPERATORS	32 x	1	32	xmplr_lut_15
xmplr_lut_3	OPERATORS	2 x	1	2	xmplr_lut_3
xmplr_lut_4	OPERATORS	13 x	1	13	xmplr_lut_4
xmplr_lut_6	OPERATORS	1 x	1	1	xmplr_lut_6
xmplr_lut_7	OPERATORS	1 x	1	1	xmplr_lut_7
xmplr_lut_8	OPERATORS	33 x	1	33	xmplr_lut_8
xmplr_lut_9	OPERATORS	36 x	1	36	xmplr_lut_9

Cell: Cam2_final_32 View: mem_unfold_979 Library: work

Total accumulated area :
 Number of Dffs or Latches : 4361
 Number of Function Generators : 12063
 Number of MUXF5 : 2118
 Number of VCC : 1
 Number of gates : 11808

 Number of ports : 165
 Number of nets : 18597
 Number of instances : 18543
 Number of references to this view : 2

	Cell	Library	References	Total Area
Latches	FDE_1	xcv2	4361 x	1 4361 Dffs or
Generators	LUT1	xcv2	6 x	1 6 Function
Generators	LUT2	xcv2	129 x	1 129 Function
Generators	LUT3	xcv2	1582 x	1 1582 Function
Generators	LUT4	xcv2	10346 x	1 10346 Function
	MUXF5	xcv2	2118 x	1 2118 MUXF5
	VCC	xcv2	1 x	1 1 VCC

Cell: xmplr_lut_8 View: DUMMY Library: OPERATORS

Total accumulated area :
 Number of xmplr_lut_8 : 1

 Number of ports : 9
 Number of nets : 0
 Number of instances : 0
 Number of references to this view : 98

```

*****

Cell: xmplr_lut_11      View: DUMMY      Library: OPERATORS

*****

Total accumulated area :
Number of xmplr_lut_11 :                1

Number of ports :                12
Number of nets :                  0
Number of instances :             0
Number of references to this view :    72

*****

Cell: xmplr_lut_12      View: DUMMY      Library: OPERATORS

*****

Total accumulated area :
Number of xmplr_lut_12 :                1

Number of ports :                13
Number of nets :                  0
Number of instances :             0
Number of references to this view :    58

*****

Cell: xmplr_lut_9       View: DUMMY      Library: OPERATORS

*****

Total accumulated area :
Number of xmplr_lut_9 :                1

Number of ports :                10
Number of nets :                  0
Number of instances :             0
Number of references to this view :   104

*****

Cell: xmplr_lut_10      View: DUMMY      Library: OPERATORS

*****

Total accumulated area :
Number of xmplr_lut_10 :                1

Number of ports :                11
Number of nets :                  0
Number of instances :             0
Number of references to this view :    26

```

Cell: xmplr_lut_14 View: DUMMY Library: OPERATORS

Total accumulated area :
Number of xmplr_lut_14 : 1

Number of ports : 15
Number of nets : 0
Number of instances : 0
Number of references to this view : 2

Cell: xmplr_lut_4 View: DUMMY Library: OPERATORS

Total accumulated area :
Number of xmplr_lut_4 : 1

Number of ports : 5
Number of nets : 0
Number of instances : 0
Number of references to this view : 26

Cell: xmplr_lut_7 View: DUMMY Library: OPERATORS

Total accumulated area :
Number of xmplr_lut_7 : 1

Number of ports : 8
Number of nets : 0
Number of instances : 0
Number of references to this view : 2

Cell: xmplr_lut_15 View: DUMMY Library: OPERATORS

Total accumulated area :
Number of xmplr_lut_15 : 1

Number of ports : 16
Number of nets : 0
Number of instances : 0
Number of references to this view : 64

Cell: xmplr_lut_3 View: DUMMY Library: OPERATORS

Total accumulated area :
Number of xmplr_lut_3 : 1

Number of ports : 4
Number of nets : 0
Number of instances : 0
Number of references to this view : 4

Cell: xmplr_lut_13 View: DUMMY Library: OPERATORS

Total accumulated area :
Number of xmplr_lut_13 : 1

Number of ports : 14
Number of nets : 0
Number of instances : 0
Number of references to this view : 2

Cell: xmplr_lut_6 View: DUMMY Library: OPERATORS

Total accumulated area :
Number of xmplr_lut_6 : 1

Number of ports : 7
Number of nets : 0
Number of instances : 0
Number of references to this view : 2

Cell: encapsulation_module_256_10_65535 View: module_unfold_1777
Library: work

Total accumulated area :

Number of Dffs or Latches :	8480
Number of Function Generators :	10758
Number of GND :	1
Number of MULT_AND :	31
Number of MUX CARRYs :	332
Number of MUXF5 :	4489
Number of MUXF6 :	2072
Number of VCC :	1
Number of XORCY :	169
Number of gates :	10725

Number of ports :	119
Number of nets :	26401
Number of instances :	26333
Number of references to this view :	2

	Cell	Library	References	Total Area
Latches	FDE	xcv2	8449 x	1 8449 Dffs or
Latches	FDRE	xcv2	31 x	1 31 Dffs or
	GND	xcv2	1 x	1 1 GND
Generators	LUT1	xcv2	19 x	1 19 Function
Generators	LUT1_L	xcv2	115 x	1 115 Function
Generators	LUT2	xcv2	59 x	1 59 Function
Generators	LUT2_L	xcv2	90 x	1 90 Function
Generators	LUT3	xcv2	8494 x	1 8494 Function
Generators	LUT3_L	xcv2	10 x	1 10 Function
Generators	LUT4	xcv2	1850 x	1 1850 Function
Generators	LUT4_L	xcv2	121 x	1 121 Function
	MULT_AND	xcv2	31 x	1 31 MULT_AND
	MUXCY	xcv2	7 x	1 7 MUX CARRYs
	MUXCY_L	xcv2	325 x	1 325 MUX CARRYs
	MUXF5	xcv2	4489 x	1 4489 MUXF5
	MUXF6	xcv2	2072 x	1 2072 MUXF6
	VCC	xcv2	1 x	1 1 VCC
	XORCY	xcv2	169 x	1 169 XORCY

```

*****
Device Utilization for 2V8000bf957
*****
Resource                Used      Avail    Utilization
-----
IOs                      230      684      33.63%
Function Generators      23646    93184     25.38%
CLB Slices               11823    46592     25.38%
Dffs or Latches         12934    95236     13.58%
-----

```

Clock Frequency Report

```

Clock                  : Frequency
-----
CLK                    : 45.6 MHz

```

4.3.11 System

Cell: systemall View: arch Library: work

Total accumulated area :

Black Box AND2 :	3842
Black Box DFFERS :	2897
Black Box DFFRS :	14
Number of Dffs or Latches :	12934
Black Box FALSE :	2
Number of Function Generators :	23756
Number of GND :	1
Black Box INV :	1460
Black Box LATRS :	274
Number of MULT_AND :	34
Black Box MUX :	1905
Number of MUX CARRYs :	1267
Number of MUXF5 :	6607
Number of MUXF6 :	2072
Black Box OR2 :	4893
Black Box TRI :	52
Black Box TRUE :	2
Number of VCC :	2
Number of XORCY :	980
Number of gates :	22533
Black Box xmplr_lut_10 :	14
Black Box xmplr_lut_11 :	29
Black Box xmplr_lut_12 :	21
Black Box xmplr_lut_13 :	2
Black Box xmplr_lut_14 :	1
Black Box xmplr_lut_15 :	32
Black Box xmplr_lut_16 :	1
Black Box xmplr_lut_2 :	1
Black Box xmplr_lut_3 :	15
Black Box xmplr_lut_4 :	13
Black Box xmplr_lut_6 :	32
Black Box xmplr_lut_7 :	2
Black Box xmplr_lut_8 :	33
Black Box xmplr_lut_9 :	38

Number of ports :	306
Number of nets :	8751
Number of instances :	8539
Number of references to this view :	0

Cell	Library	References		Total Area	
AND2	PRIMITIVES	2474	x 1	2474	AND2
DFFERS	PRIMITIVES	1233	x 1	1233	DFFERS
DFFRS	PRIMITIVES	7	x 1	7	DFFRS
FALSE	PRIMITIVES	1	x 1	1	FALSE
INV	PRIMITIVES	402	x 1	402	INV
LUT1_L	xcv2	44	x 1	44	Function
Generators					
LUT2_L	xcv2	61	x 1	61	Function
Generators					
LUT4_L	xcv2	5	x 1	5	Function
Generators					
MUX	PRIMITIVES	1105	x 1	1105	MUX
MUXCY	xcv2	9	x 1	9	MUX CARRYs
MUXCY_L	xcv2	101	x 1	101	MUX CARRYs
OR2	PRIMITIVES	2926	x 1	2926	OR2
TRI	PRIMITIVES	36	x 1	36	TRI
TRUE	PRIMITIVES	1	x 1	1	TRUE
XORCY	xcv2	81	x 1	81	XORCY
system4mod	work	1	x 2072	2072	MUXF6
			1	1	GND
			899	899	XORCY
			1157	1157	MUX CARRYs
			34	34	MULT_AND
			7	7	DFFRS
			1	1	xmplr_lut_6
			1	1	xmplr_lut_13
			2	2	xmplr_lut_3
			32	32	xmplr_lut_15
			1	1	xmplr_lut_7
			13	13	xmplr_lut_4
			1	1	xmplr_lut_14
			13	13	xmplr_lut_10
			36	36	xmplr_lut_9
			21	21	xmplr_lut_12
			28	28	xmplr_lut_11
			33	33	xmplr_lut_8
			1967	1967	OR2
			1368	1368	AND2
			274	274	LATRS
			1664	1664	DFFERS
			800	800	MUX
			1058	1058	INV
			12934	12934	Dffs or
Latches					
			6607	6607	MUXF5
Generators			23646	23646	Function
			2	2	VCC
			22533	22533	gates
			16	16	TRI
			1	1	TRUE
			1	1	FALSE

xmplr_lut_10	OPERATORS	1 x	1	1 xmplr_lut_10
xmplr_lut_11	OPERATORS	1 x	1	1 xmplr_lut_11
xmplr_lut_13	OPERATORS	1 x	1	1 xmplr_lut_13
xmplr_lut_16	OPERATORS	1 x	1	1 xmplr_lut_16
xmplr_lut_2	OPERATORS	1 x	1	1 xmplr_lut_2
xmplr_lut_3	OPERATORS	13 x	1	13 xmplr_lut_3
xmplr_lut_6	OPERATORS	31 x	1	31 xmplr_lut_6
xmplr_lut_7	OPERATORS	1 x	1	1 xmplr_lut_7
xmplr_lut_9	OPERATORS	2 x	1	2 xmplr_lut_9

Cell: system4mod View: arch_unfold_1184 Library: work

Total accumulated area :

Number of AND2 :	1368
Number of DFFERS :	1664
Number of DFFRS :	7
Number of Dffs or Latches :	12934
Number of FALSE :	1
Number of Function Generators :	23646
Number of GND :	1
Number of INV :	1058
Number of LATRS :	274
Number of MULT_AND :	34
Number of MUX :	800
Number of MUX CARRYs :	1157
Number of MUXF5 :	6607
Number of MUXF6 :	2072
Number of OR2 :	1967
Number of TRI :	16
Number of TRUE :	1
Number of VCC :	2
Number of XORCY :	899
Number of gates :	22533
Number of xmplr_lut_10 :	13
Number of xmplr_lut_11 :	28
Number of xmplr_lut_12 :	21
Number of xmplr_lut_13 :	1
Number of xmplr_lut_14 :	1
Number of xmplr_lut_15 :	32
Number of xmplr_lut_3 :	2
Number of xmplr_lut_4 :	13
Number of xmplr_lut_6 :	1
Number of xmplr_lut_7 :	1
Number of xmplr_lut_8 :	33
Number of xmplr_lut_9 :	36

Number of ports :	230
Number of nets :	9930
Number of instances :	9816
Number of references to this view :	1

Cell	Library	References	Total Area
AND2	PRIMITIVES	1368 x 1	1368 AND2
Cam2_final_32	work	1 x 11808	11808 gates
		1	1 VCC
		12063	12063 Function
Generators		2118	2118 MUXF5
		4361	4361 Dffs or
Latches			
DFFERS	PRIMITIVES	1664 x 1	1664 DFFERS
DFFRS	PRIMITIVES	7 x 1	7 DFFRS
FALSE	PRIMITIVES	1 x 1	1 FALSE
FDRE	xcv2	93 x	1 93 Dffs or
Latches			
INV	PRIMITIVES	1058 x 1	1058 INV
LATRS	PRIMITIVES	274 x 1	274 LATRS
LUT1_L	xcv2	197 x	1 197 Function
Generators			
LUT2_L	xcv2	22 x	1 22 Function
Generators			
LUT4_L	xcv2	606 x	1 606 Function
Generators			
MULT_AND	xcv2	3 x	1 3 MULT_AND
MUX	PRIMITIVES	800 x 1	800 MUX
MUXCY	xcv2	50 x 1	50 MUX CARRYs
MUXCY_L	xcv2	775 x 1	775 MUX CARRYs
OR2	PRIMITIVES	1967 x 1	1967 OR2
TRI	PRIMITIVES	16 x 1	16 TRI
TRUE	PRIMITIVES	1 x 1	1 TRUE
XORCY	xcv2	730 x 1	730 XORCY
encapsulation_module_256_10_65535	work	1 x	10725
10725 gates			
		1	1 GND
		1	1 VCC
		2072	2072 MUXF6
		4489	4489 MUXF5
		8480	8480 Dffs or
Latches			
		169	169 XORCY
		332	332 MUX CARRYs
Generators		10758	10758 Function
		31	31 MULT_AND
xmplr_lut_10	OPERATORS	13 x 1	13 xmplr_lut_10
xmplr_lut_11	OPERATORS	28 x 1	28 xmplr_lut_11
xmplr_lut_12	OPERATORS	21 x 1	21 xmplr_lut_12
xmplr_lut_13	OPERATORS	1 x 1	1 xmplr_lut_13
xmplr_lut_14	OPERATORS	1 x 1	1 xmplr_lut_14
xmplr_lut_15	OPERATORS	32 x 1	32 xmplr_lut_15
xmplr_lut_3	OPERATORS	2 x 1	2 xmplr_lut_3
xmplr_lut_4	OPERATORS	13 x 1	13 xmplr_lut_4
xmplr_lut_6	OPERATORS	1 x 1	1 xmplr_lut_6
xmplr_lut_7	OPERATORS	1 x 1	1 xmplr_lut_7
xmplr_lut_8	OPERATORS	33 x 1	33 xmplr_lut_8
xmplr_lut_9	OPERATORS	36 x 1	36 xmplr_lut_9

Cell: Cam2_final_32 View: mem_unfold_979 Library: work

Total accumulated area :

Number of Dffs or Latches : 4361
 Number of Function Generators : 12063
 Number of MUXF5 : 2118
 Number of VCC : 1
 Number of gates : 11808

Number of ports : 165
 Number of nets : 18597
 Number of instances : 18543
 Number of references to this view : 2

	Cell	Library	References	Total Area
Latches	FDE_1	xcv2	4361 x	1 4361 Dffs or
Generators	LUT1	xcv2	6 x	1 6 Function
Generators	LUT2	xcv2	129 x	1 129 Function
Generators	LUT3	xcv2	1582 x	1 1582 Function
Generators	LUT4	xcv2	10346 x	1 10346 Function
	MUXF5	xcv2	2118 x	1 2118 MUXF5
	VCC	xcv2	1 x	1 1 VCC

Cell: xmplr_lut_8 View: DUMMY Library: OPERATORS

Total accumulated area :

Number of xmplr_lut_8 : 1
 Number of ports : 9
 Number of nets : 0
 Number of instances : 0
 Number of references to this view : 98

Cell: xmplr_lut_11 View: DUMMY Library: OPERATORS

Total accumulated area :

Number of xmplr_lut_11 : 1

Number of ports : 12

Number of nets : 0

Number of instances : 0

Number of references to this view : 74

Cell: xmplr_lut_12 View: DUMMY Library: OPERATORS

Total accumulated area :

Number of xmplr_lut_12 : 1

Number of ports : 13

Number of nets : 0

Number of instances : 0

Number of references to this view : 58

Cell: xmplr_lut_9 View: DUMMY Library: OPERATORS

Total accumulated area :

Number of xmplr_lut_9 : 1

Number of ports : 10

Number of nets : 0

Number of instances : 0

Number of references to this view : 108

Cell: xmplr_lut_10 View: DUMMY Library: OPERATORS

Total accumulated area :

Number of xmplr_lut_10 : 1

Number of ports : 11

Number of nets : 0

Number of instances : 0

Number of references to this view : 28

Cell: xmplr_lut_14 View: DUMMY Library: OPERATORS

Total accumulated area :
Number of xmplr_lut_14 : 1

Number of ports : 15
Number of nets : 0
Number of instances : 0
Number of references to this view : 2

Cell: xmplr_lut_4 View: DUMMY Library: OPERATORS

Total accumulated area :
Number of xmplr_lut_4 : 1

Number of ports : 5
Number of nets : 0
Number of instances : 0
Number of references to this view : 26

Cell: xmplr_lut_7 View: DUMMY Library: OPERATORS

Total accumulated area :
Number of xmplr_lut_7 : 1

Number of ports : 8
Number of nets : 0
Number of instances : 0
Number of references to this view : 4

Cell: xmplr_lut_15 View: DUMMY Library: OPERATORS

Total accumulated area :
Number of xmplr_lut_15 : 1

Number of ports : 16
Number of nets : 0
Number of instances : 0
Number of references to this view : 64

Cell: xmplr_lut_3 View: DUMMY Library: OPERATORS

Total accumulated area :
Number of xmplr_lut_3 : 1

Number of ports : 4
Number of nets : 0
Number of instances : 0
Number of references to this view : 30

Cell: xmplr_lut_13 View: DUMMY Library: OPERATORS

Total accumulated area :
Number of xmplr_lut_13 : 1

Number of ports : 14
Number of nets : 0
Number of instances : 0
Number of references to this view : 4

Cell: xmplr_lut_6 View: DUMMY Library: OPERATORS

Total accumulated area :
Number of xmplr_lut_6 : 1

Number of ports : 7
Number of nets : 0
Number of instances : 0
Number of references to this view : 64

Cell: encapsulation_module_256_10_65535 View: module_unfold_1777
Library: work

Total accumulated area :

Number of Dffs or Latches : 8480
Number of Function Generators : 10758
Number of GND : 1
Number of MULT_AND : 31
Number of MUX CARRYs : 332
Number of MUXF5 : 4489
Number of MUXF6 : 2072
Number of VCC : 1
Number of XORCY : 169
Number of gates : 10725

Number of ports : 119
Number of nets : 26401
Number of instances : 26333
Number of references to this view : 2

	Cell	Library	References	Total Area
Latches	FDE	xcv2	8449 x	1 8449 Dffs or
Latches	FDRE	xcv2	31 x	1 31 Dffs or
Generators	GND	xcv2	1 x	1 1 GND
Generators	LUT1	xcv2	19 x	1 19 Function
Generators	LUT1_L	xcv2	115 x	1 115 Function
Generators	LUT2	xcv2	59 x	1 59 Function
Generators	LUT2_L	xcv2	90 x	1 90 Function
Generators	LUT3	xcv2	8494 x	1 8494 Function
Generators	LUT3_L	xcv2	10 x	1 10 Function
Generators	LUT4	xcv2	1850 x	1 1850 Function
Generators	LUT4_L	xcv2	121 x	1 121 Function
	MULT_AND	xcv2	31 x	1 31 MULT_AND
	MUXCY	xcv2	7 x	1 7 MUX CARRYs
	MUXCY_L	xcv2	325 x	1 325 MUX CARRYs
	MUXF5	xcv2	4489 x	1 4489 MUXF5
	MUXF6	xcv2	2072 x	1 2072 MUXF6
	VCC	xcv2	1 x	1 1 VCC
	XORCY	xcv2	169 x	1 169 XORCY

Cell: xmplr_lut_16 View: DUMMY Library: OPERATORS

Total accumulated area :
Number of xmplr_lut_16 : 1

Number of ports : 17
Number of nets : 0
Number of instances : 0
Number of references to this view : 2

Cell: xmplr_lut_2 View: DUMMY Library: OPERATORS

Total accumulated area :
Number of xmplr_lut_2 : 1

Number of ports : 3
Number of nets : 0
Number of instances : 0
Number of references to this view : 2

Device Utilization for 2V250fg256

Resource	Used	Avail	Utilization
IOs	306	172	177.91%
Function Generators	23756	3072	773.31%
CLB Slices 1	1878	1536	773.31%
Dffs or Latches	12934	3588	360.48%

This design does not fit in the device specified!
Trying an alternate device ...

```

*****
Device Utilization for 2V3000bf957
*****
Resource                Used      Avail    Utilization
-----
IOs                      306      684     44.74%
Function Generators      23756    28672    82.85%
CLB Slices               11878    14336    82.85%
Dffs or Latches         12934    30724    42.10%
-----

```

Clock Frequency Report

```

Clock                  : Frequency
-----
CLK                    : 42.6 MHz

```

Κεφάλαιο 5^ο

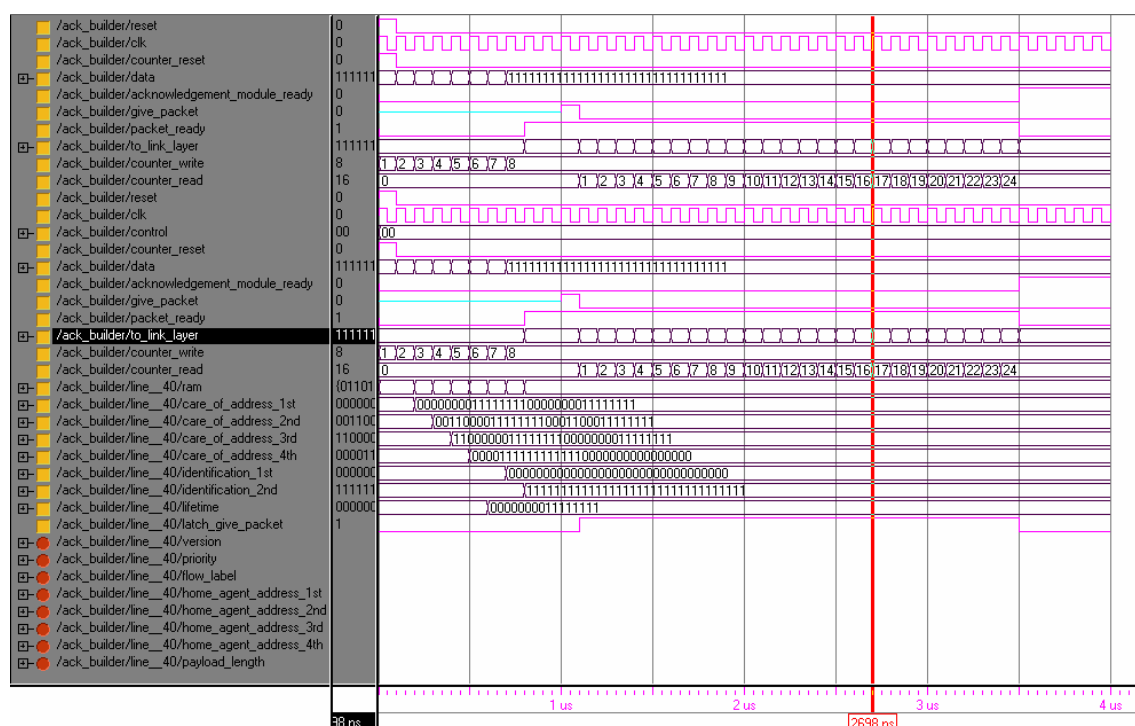
Οι εξομοιώσεις του Συστήματος

5.1 Binding Acknowledgement

Στα σχήματα τα οποία ακολουθούνε παρουσιάζουμε την διαδικασία κατασκευής των πακέτων Binding Acknowledgement. Όπως έχουμε αναφέρει, τα συγκεκριμένα πακέτα κατασκευάζονται ύστερα από την διαδικασία Binding Update αν το A flag στο μήνυμα Binding Update το οποίο το αποστέλλει ο κινητός χρήστης στον Home Agent του είναι άσος.

Στο σχήμα που ακολουθεί παρακολουθούμε την διαδικασία κατά την οποία τα κενά πεδία στην μνήμη της μονάδας κατασκευής καλύπτονται με τα δεδομένα τα οποία έρχονται από την μονάδα Packet Recognition.

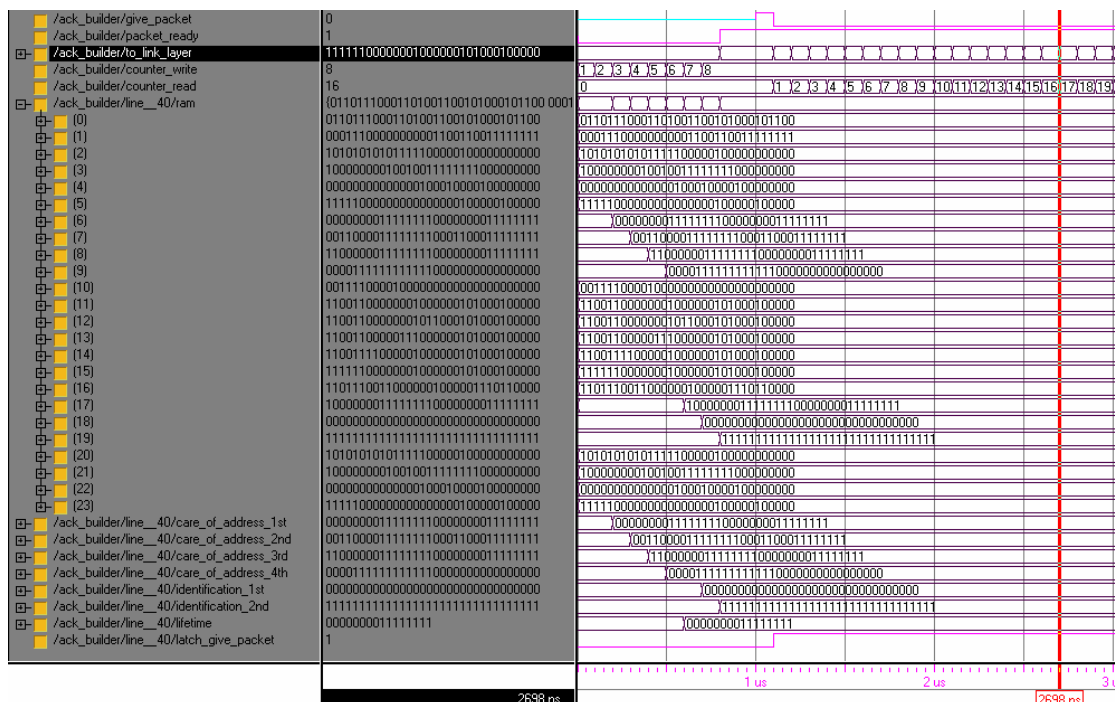
- **Counter_reset**: απαραίτητο για να μηδενιστούν οι μετρητές **counter_write** και **counter_read**. Ο μετρητής counter_write είναι ο μετρητής που χρησιμοποιείται για να τοποθετηθούν τα δεδομένα στα κατάλληλα πεδία του πακέτου, χρησιμοποιώντας μια σύμβαση την οποία γνωρίζει και η μονάδα Packet Recognition η οποία τροφοδοτεί τα δεδομένα. Ο counter_read για να αναγνώσει το πακέτο και να μεταφερθεί στο Data Link Layer. Ο counter_read αυξάνεται ανάλογα με το μέγεθος του πακέτου π.χ στην προκειμένη περίπτωση έχει μέγιστη τιμή 24 όσα και τα 32 bit block του Binding Acknowledgement.
- **Data**: το bus των δεδομένων που έρχονται από την Packet Recognition.



- **Control**: Η τιμή "00" όπως έχουμε περιγράψει.
- **Acknowledgement_module_ready**: ειδοποιεί την control unit ότι η μονάδα είναι έτοιμη για νέα διεργασία.
- **Packet_ready**: Ενημερώνει το Data Link Layer ότι υπάρχει πακέτο διαθέσιμο το οποίο ζητά μεταφορά μεγέθους όσο η τιμή του data bus **to_link_layer** την ίδια χρονική στιγμή.

- **Give_packet:** Το Data Link Layer ειδοποιεί ότι μπορεί να δεχθεί το πακέτο.
- **To_link_layer:** Το data bus interface προς το Data Link Layer.

Στην εξομοίωση που ακολουθεί παρακολουθούμε την διαδικασία κατά την οποία γεμίζουν τα διάφορα πεδία με δεδομένα, από τις 0 – 23 θέσεις μνήμης. Στη συγκεκριμένη εξομοίωση παρατηρούμε ότι κάποια σταθερά πεδία όπως η διεύθυνση Home Agent τίθενται από την αρχή μετά το πρώτο reset και κατά την διάρκεια της κατασκευής καθώς ο μετρητής WRITE αυξάνονται και τα υπόλοιπα με τα δεδομένα τα οποία έρχονται από την Packet Recognition Module.

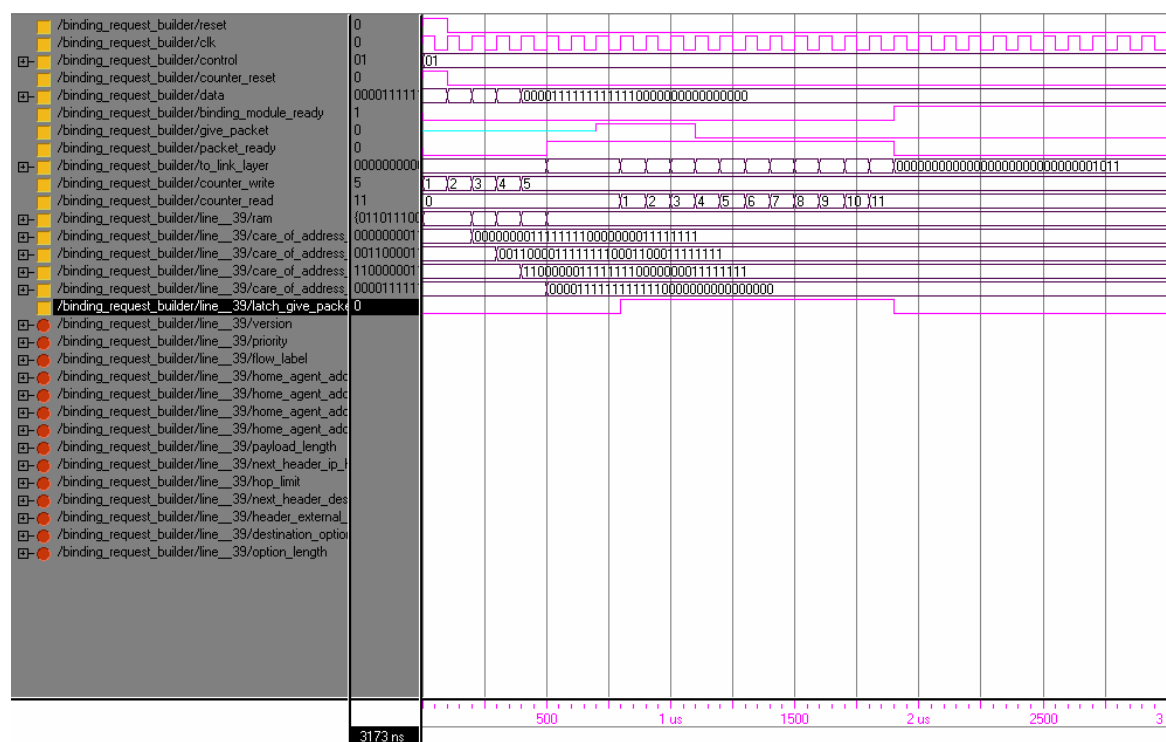


Καθώς αυξάνεται ο μετρητής READ από 1 έως 24 που είναι και το μήκος του πακέτου (συνολικό μέγεθος 24 x 32 bit) και ένα 32 bit block από το πακέτο που έχει σχηματιστεί μεταφέρεται στο data link layer μέσω του bus to_link_layer. Όταν τελειώσει η μεταφορά τότε η μονάδα ειδοποιεί την control unit με το σήμα acknowledgement_module_ready.

5.2 Binding Request

Η μονάδα κατασκευάζει τα πακέτα binding requests τα οποία είναι τα πακέτα τα οποία ειδοποιούν κάποιο κινητό χρήστη ότι η περίοδος για την οποία το binding θεωρείται έγκυρο λήγει. Ουσιαστικά ζητά από τον κινητό χρήστη να αποστείλει πακέτο binding update με την παρούσα care of address του.

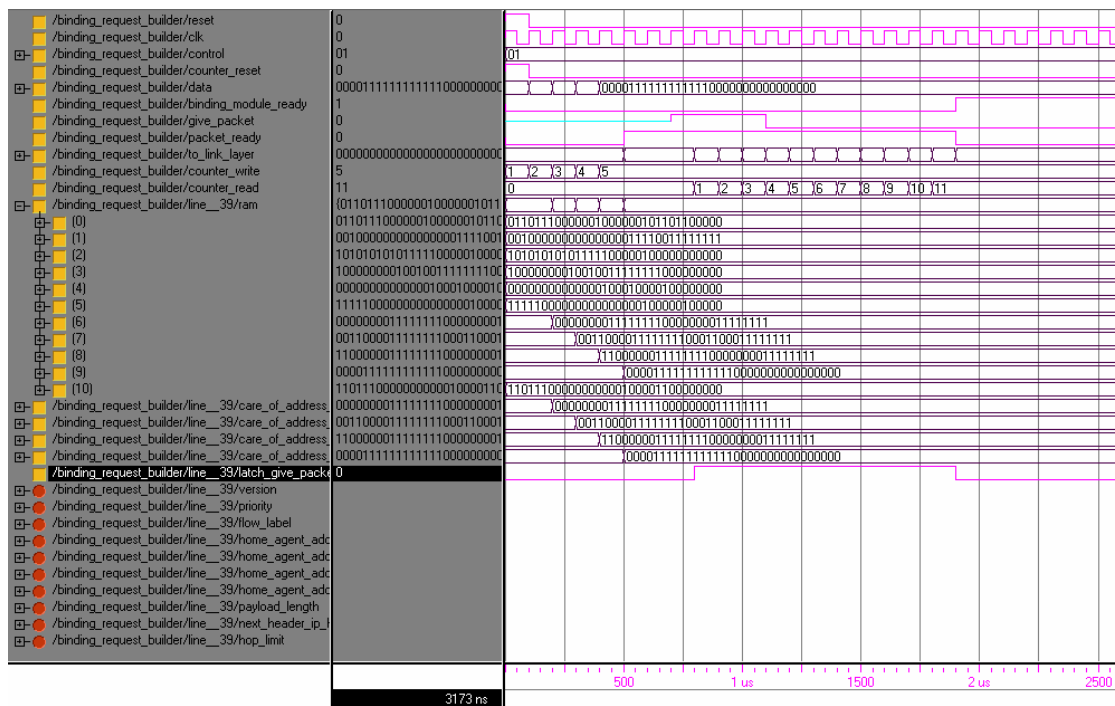
- **Counter_reset**: απαραίτητο για να μηδενιστούν οι μετρητές **counter_write** και **counter_read**. Ο μετρητής counter_write είναι ο μετρητής που χρησιμοποιείται για να τοποθετηθούν τα δεδομένα στα κατάλληλα πεδία του πακέτου, χρησιμοποιώντας μια σύμβαση την οποία γνωρίζει και η μονάδα Packet Recognition η οποία τροφοδοτεί τα δεδομένα. Ο counter_read για να αναγνωστεί το πακέτο και να μεταφερθεί στο Data Link Layer. Ο counter_read αυξάνεται ανάλογα με το μέγεθος του πακέτου π.χ στην προκειμένη περίπτωση έχει μέγιστη τιμή 11 όσα και τα 32 bit block του Binding Request μηνύματος.
- **Data**: το bus των δεδομένων που έρχονται από την Packet Recognition.



- **Control**: Η τιμή “01” όπως έχουμε περιγράψει.
- **Binding_module_ready**: ειδοποιεί την control unit ότι η μονάδα είναι έτοιμη για νέα διεργασία.

- **Packet_ready:** Ενημερώνει το Data Link Layer ότι υπάρχει πακέτο διαθέσιμο το οποίο ζητά μεταφορά μεγέθους όσο η τιμή του data bus **to_link_layer** δηλαδή 12 block 32 bit την ίδια χρονική στιγμή.
- **Give_packet:** Το Data Link Layer ειδοποιεί ότι μπορεί να δεχθεί το πακέτο.
- **To_link_layer:** Το data bus interface προς το Data Link Layer.

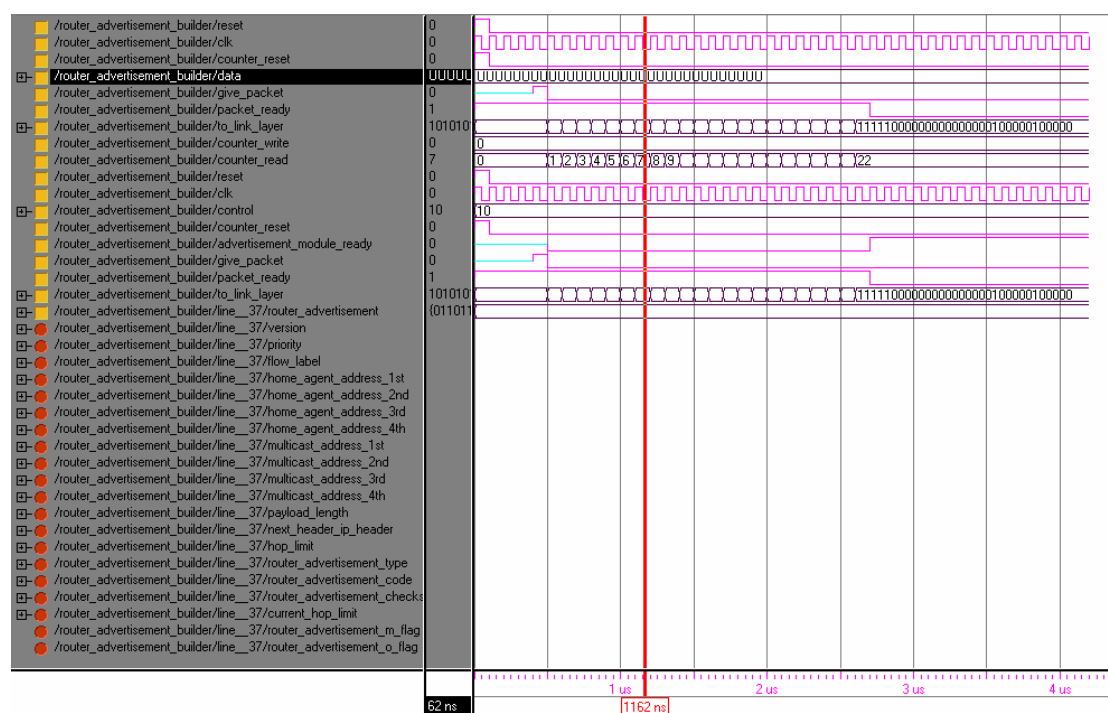
Στην εξομοίωση που ακολουθεί παρακολουθούμε την διαδικασία κατά την οποία γεμίζουν τα διάφορα πεδία με δεδομένα, από τις 0 – 10 θέσεις μνήμης. Στη συγκεκριμένη εξομοίωση παρατηρούμε ότι κάποια σταθερά πεδία όπως η διεύθυνση Home Agent τίθενται από την αρχή μετά το πρώτο reset και κατά την διάρκεια της κατασκευής καθώς ο μετρητής WRITE αυξάνονται και τα υπόλοιπα με τα δεδομένα τα οποία έρχονται από την Packet Recognition Module.



Καθώς αυξάνεται ο μετρητής READ από 1 έως 11 που είναι και το μήκος του πακέτου (συνολικό μέγεθος 11 x 32 bit) και ένα 32 bit block από το πακέτο που έχει σχηματιστεί μεταφέρεται στο data link layer μέσω του bus **to_link_layer**. Όταν τελειώσει η μεταφορά τότε η μονάδα ειδοποιεί την control unit με το σήμα **binding_module_ready**.

5.3 Router Advertisement

Το σύστημα IPv6 στην πλευρά του Home Agent θα πρέπει να έχει την δυνατότητα να στέλνει πακέτα Router Advertisement στην Home Link Multicast Address ώστε να γνωρίζει ο κινητός χρήστης την care of address του και να μπορεί να την συγκρίνει με προηγούμενες τιμές της, ώστε να μπορεί να αντιλαμβάνεται την κίνησή του και την θέση του κάθε φορά.



Η λειτουργία της μονάδας είναι παρόμοια με τις υπόλοιπες μονάδες που περιγράφηκαν οι οποίες έχουν ως έργο να κατασκευάζουν ένα συγκεκριμένο είδος πακέτου.

- **Counter_reset**: απαραίτητο για να μηδενιστούν οι μετρητές **counter_write** και **counter_read**. Ο μετρητής **counter_write** είναι ο μετρητής που χρησιμοποιείται για να τοποθετηθούν τα δεδομένα στα κατάλληλα πεδία του πακέτου, χρησιμοποιώντας μια σύμβαση την οποία γνωρίζει και η μονάδα Packet Recognition η οποία τροφοδοτεί τα δεδομένα. Ο **counter_read** για να αναγνωστεί το πακέτο και να μεταφερθεί στο Data Link Layer. Ο **counter_read** αυξάνεται ανάλογα με το μέγεθος του πακέτου π.χ στην προκειμένη περίπτωση έχει μέγιστη τιμή 22 όσα και τα 32 bit block του Router Advertisement μηνύματος.
- **Data**: το bus των δεδομένων που έρχονται από την Packet Recognition.

- **Control:** Η τιμή “10” επιτρέπει την επικοινωνία της μονάδας με τις μονάδες Packet Recognition, Control Unit.
- **Advertisement_module_ready:** ειδοποιεί την control unit ότι η μονάδα είναι έτοιμη για νέα διεργασία.
- **Packet_ready:** Ενημερώνει το Data Link Layer ότι υπάρχει πακέτο διαθέσιμο το οποίο ζητά μεταφορά μεγέθους όσο η τιμή του data bus **to_link_layer** την ίδια χρονική στιγμή.
- **Give_packet:** Το Data Link Layer ειδοποιεί ότι μπορεί να δεχθεί το πακέτο.
- **To_link_layer:** Το data bus interface προς το Data Link Layer.

Το συγκεκριμένο είδος πακέτου παρουσιάζει μια ιδιομορφία η οποία είναι ότι δεν έχει μεταβλητού περιεχομένου πεδία στο πακέτο, δηλαδή για έναν συγκεκριμένο Router τα πεδία είναι απόλυτα καθορισμένα και το πακέτο βρίσκεται έτοιμο στην εσωτερική μνήμη της μονάδας μετά την αρχικοποίηση της. Δίνει σήμα στο επίπεδο Data Link Layer όταν το σύστημα δεχθεί κάποιο πακέτο Router Solicitation το οποίο είναι ένα μήνυμα το οποίο ζητά από τον δρομολογητή να αποστείλει τα στοιχεία του άμεσα καθώς ο κινητός χρήστης δεν επιθυμεί να περιμένει το καθορισμένο διάστημα μετά από το οποίο ο δρομολογητής στέλνει το πακέτο αυτοβούλως.

5.4 Encapsulation

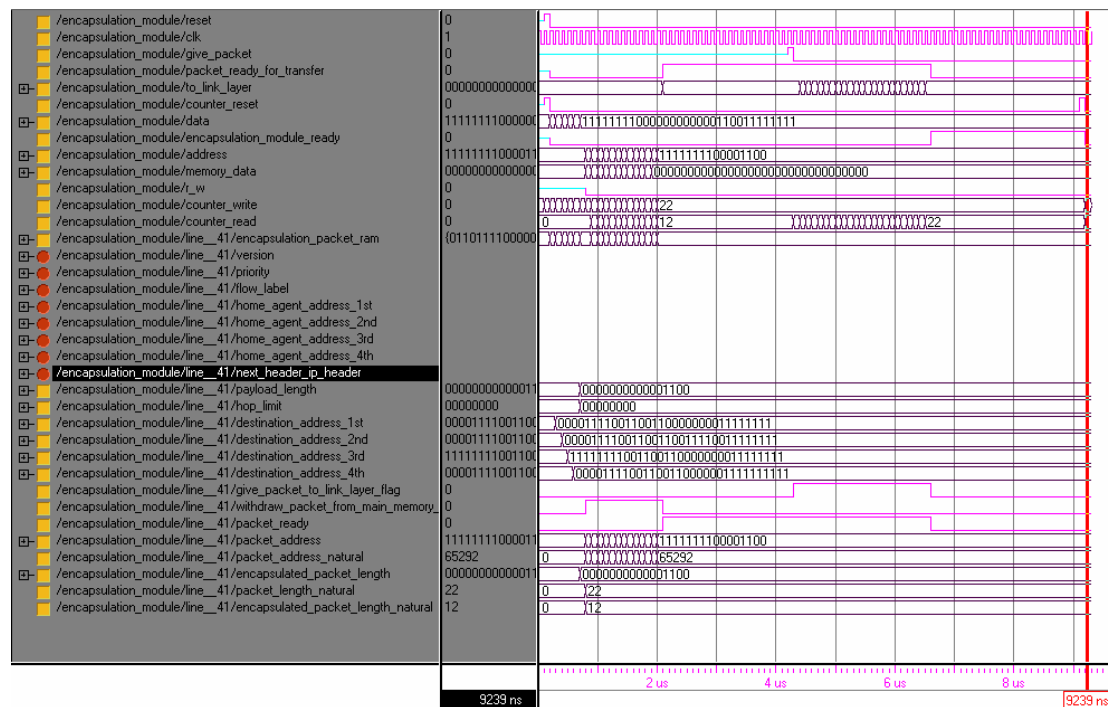
Όπως έχουμε περιγράψει στην υλοποίηση του Mobile IPv6 μια από τις πιο βασικές διαδικασίες τις οποίες πρέπει να εκτελεί το σύστημα είναι να μπορεί να ενθυλακώσει τα πακέτα τα οποία έρχονται με προορισμό τον κινητό χρήστη και να τα αποστέλλει στην έγκυρη διεύθυνση care of address στην οποία βρίσκεται ο κινητός χρήστης κάθε φορά.

Από άποψη υλοποίησης η μονάδα είναι η πιο σύνθετη από της μονάδες κατασκευής πακέτων καθώς παρουσιάζει την ιδιομορφία να έχει δικό της ξεχωριστό interface με την μονάδα Packet Recognition, με την Κύρια Μνήμη και με το Data Link Layer. Η διεπαφή με την Κύρια Μνήμη χρησιμοποιείται για να τραβήξει η μονάδα από την Μνήμη το πακέτο το οποίο είναι προς ενθυλάκωση χωρίς τα δεδομένα να περνάνε πρώτα από την μονάδα Packet Recognition όπως συμβαίνει με τις υπόλοιπες μονάδες κατασκευής. Αυτή η σχεδιαστική επιλογή οδηγεί σε χαμηλότερη κατανάλωση του συστήματος ενθυλάκωσης. Η διεπαφή με την μονάδα Packet Recognition έχει διατηρηθεί ωστόσο χρησιμοποιείται για να ενημέρωση της μονάδας ενθυλάκωσης για τα παρακάτω στοιχεία:

- Care of Address του ενθυλακωμένου πακέτου
- Block μνήμης από το οποίο αρχίζει το πακέτο
- Μέγεθος του πακέτου

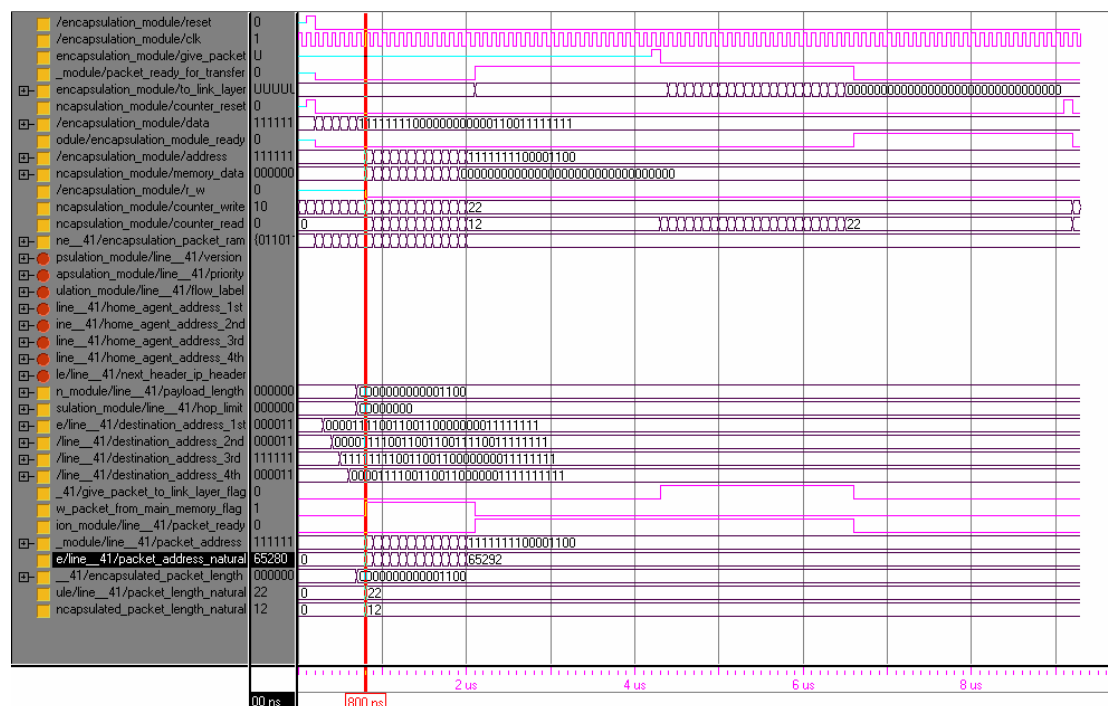
Όπως γνωρίζουμε το μέγεθος του IPv6 Header είναι 40 bytes. Το μέγεθος του πακέτου προς ενθυλάκωση είναι μεταβλητό. Στο παράδειγμα που ακολουθεί βλέπουμε την περίπτωση που το πακέτο προς ενθυλάκωση είναι 48 bytes οπότε συνολικά το πακέτο αποτελείται από 22 blocks των 32 bit. Όταν γίνεται εγγραφή δεδομένων από την μονάδα Packet Recognition για τον σχηματισμό του Header και για την εγγραφή του πακέτου προς ενθυλάκωση αυξάνεται ο μετρητής counter_write, ο οποίος στην προκειμένη περίπτωση φτάνει την τιμή 22 όσο και το μέγεθος του πακέτου που κατασκευάστηκε. Ο μετρητής Counter_Read κατά την διαδικασία fetch των δεδομένων από την Κύρια Μνήμη αυξάνεται από 1-12 όσα δηλαδή κατά blocks τα οποία μεταφέρθηκαν από την Κύρια Μνήμη. Συνεπώς ο πρώτος counter είναι εκείνος που αντιστοιχεί στην θέση στην εσωτερική μνήμη της μονάδας και ο δεύτερος στον αριθμό του block το οποίο μεταφέρεται από την Κύρια Μνήμη. Κατά την μεταφορά του πακέτου στο data link layer ο counter_read δείχνει το block μνήμης το οποίο μεταφέρεται.

Στο σχήμα που ακολουθεί παρατηρούμε την αύξηση της διεπαφής εξόδου memory address σε κάθε κύκλο ρολογιού και την διεπαφή r_w η οποία βρίσκεται σταθερά στην τιμή 0 καθώς πρόκειται για διαδικασία memory read. Όταν ολοκληρώνεται η κατασκευή του μηνύματος, η τιμή της διεπαφής εξόδου packet_ready_for_transfer γίνεται 1 ώστε να ειδοποιήσει το data link layer ότι υπάρχει διαθέσιμο πακέτο προς μεταφορά. Όταν το data link layer ανταποκριθεί στην μονάδα με σήμα give_packet τότε η μονάδα αρχίζει να κάνει ανάγνωση της εσωτερικής μνήμης που έχει και να βγάζει στην διεπαφή εξόδου to_link_layer τα δεδομένα από την εσωτερική της μνήμης καθώς αυξάνεται και η μεταβλητή counter_read. Όταν όλο το πακέτο μεταφερθεί στο data link layer τότε η μονάδα ενημερώνει την μονάδα Control Unit με το σήμα encapsulation_module_ready.

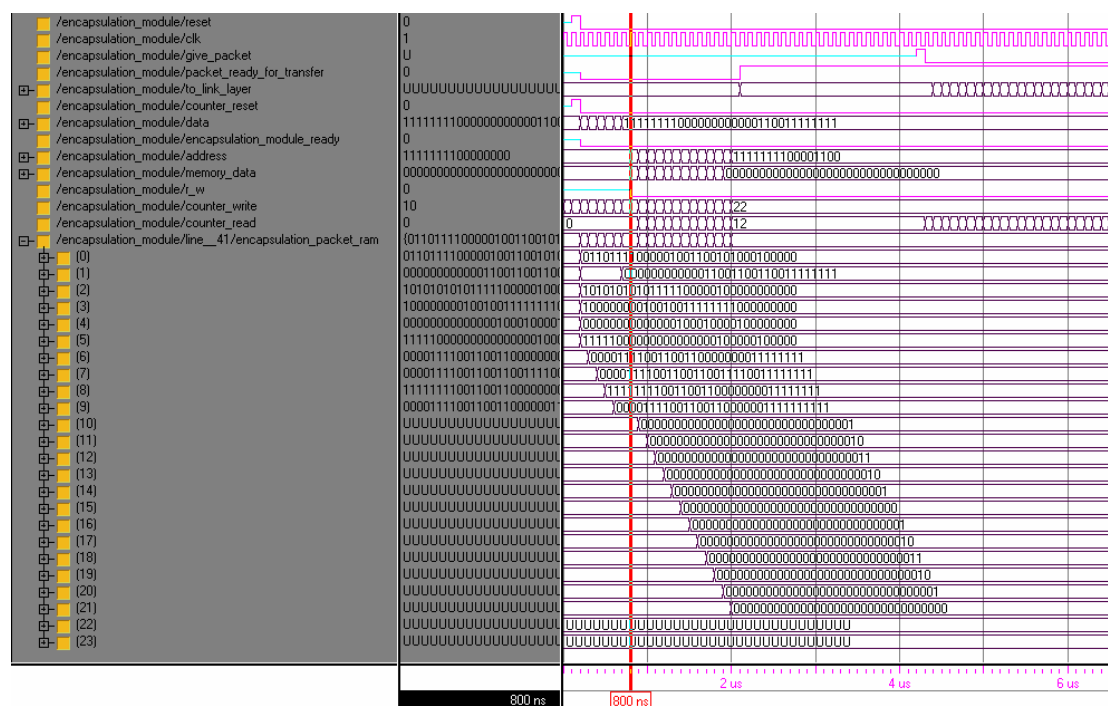


Στο παραπάνω σχήμα παρατηρούμε και την μεταβλητή withdraw_packet_from_main_memory_flag η οποία όταν έχει τιμή '1' δείχνει ότι γίνεται η μεταφορά του πακέτου από την βασική μνήμη στη μονάδα. Αμέσως αφού γίνει η μεταφορά το σήμα packet_ready γίνεται '1' καθώς τα δεδομένα του πακέτου το οποίο ενθυλακώνετε είναι τα τελευταία τα οποία μπαίνουν στην μνήμη της μονάδας για τον σχηματισμό του πακέτο. Επίσης υπάρχει η διεπαφή εξόδου packet_address_natural η οποία αναφέρεται στην διεύθυνση μνήμης από την οποία γίνεται μεταφορά block 32 bit.

Παρακάτω παρακολουθούμε την φόρτωση των μεταβλητών destination_address κατά την διάρκεια επικοινωνίας της μονάδας με την Packet Recognition Module, στην πρώτη φάση της κατασκευής του πακέτου.

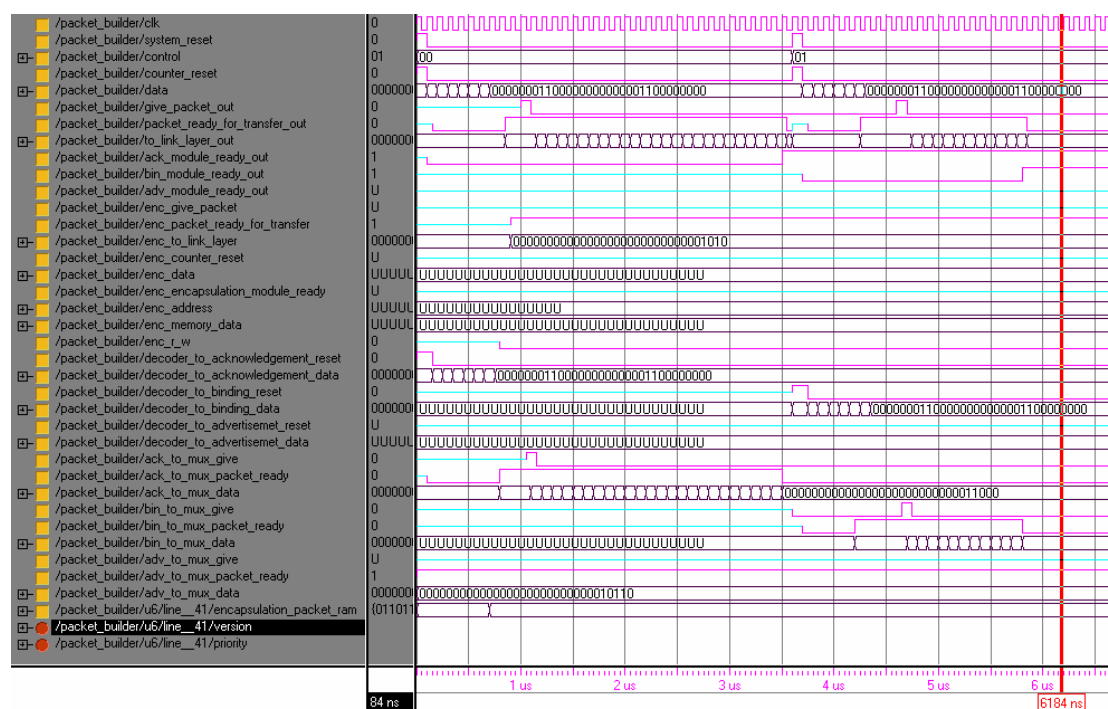


Στην εξομοίωση που ακολουθεί παρακολουθούμε την διαδικασία της φόρτωσης των δεδομένων από την Βασική Μνήμη του συστήματος στην εσωτερική μνήμη της μονάδας. Συγκεκριμένα βλέπουμε το πως γεμίζουν με δεδομένα οι θέσεις από 10 έως 22.

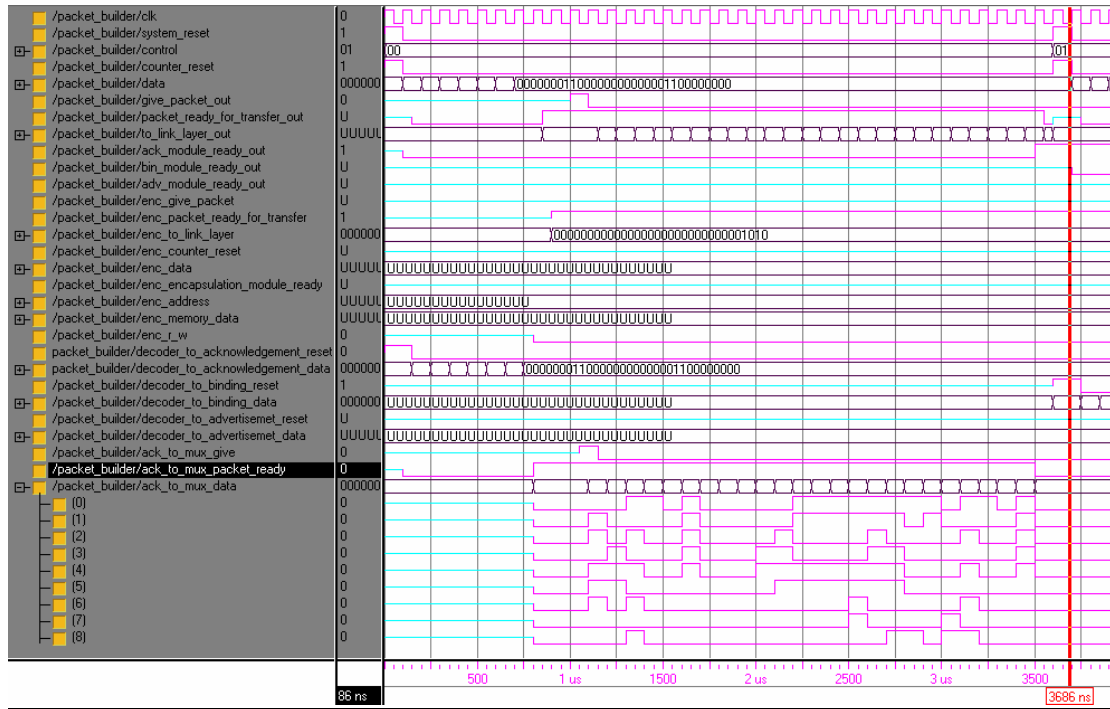


5.5 Packet Builder

Στην εξομοίωση που ακολουθεί παρατηρούμε την λειτουργία την μονάδας κατασκευής των πακέτων η οποία αποτελείται από 6 υπομονάδες. Συγκεκριμένα παρακολουθούμε την κατασκευή πρώτα ενός πακέτου binding acknowledgement και στην συνέχεια την κατασκευή ενός πακέτου binding request και πως τα σήματα εξόδου πολυπλέκονται στο data bus **to_link_layer**.



Στην συνέχεια δίνεται μια εξομοίωση που περιγράφει πιο παραστατικά την πολύπλεξη των δεδομένων στην έξοδο προς το κατώτερο επίπεδο. Επίσης μπορούμε να διακρίνουμε την εργασία που εκτελεί η μονάδα decoder η οποία αναλαμβάνει να παραδώσει τα σήματα counter_reset και data στις κατάλληλες μονάδες ανάλογα με το σήμα control το οποίο δίνει η μονάδα Control Unit η οποία είναι πάντα ενήμερη για το είδος της επεξεργασίας το οποίο πραγματοποιεί

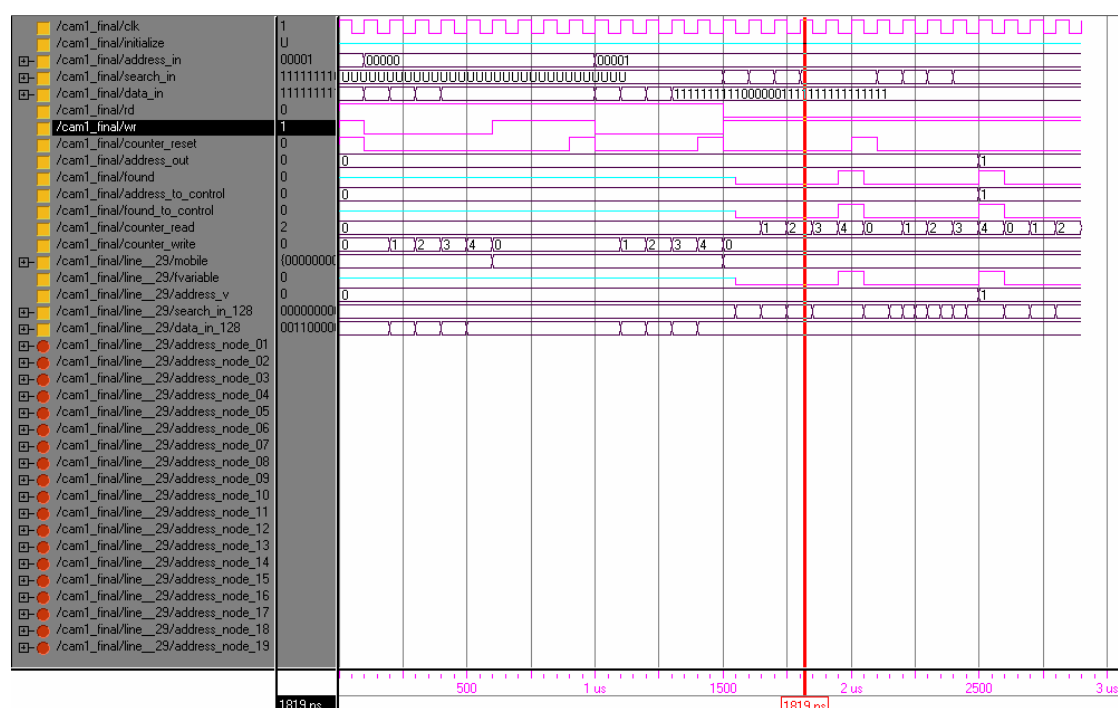


5.6 Content Addressed Memory

Όπως έχουμε αναφέρει η μονάδα χρειάζεται για να κρατάει αρχείο με τις home link address addresses, ουσιαστικά να γνωρίζει ποίους κινητούς χρήστες εξυπηρετεί αλλά και για να κάνει μεταγλώττιση των mobile home link addresses σε care of addresses ώστε να γίνεται η διαδικασία της ενθυλάκωσης.

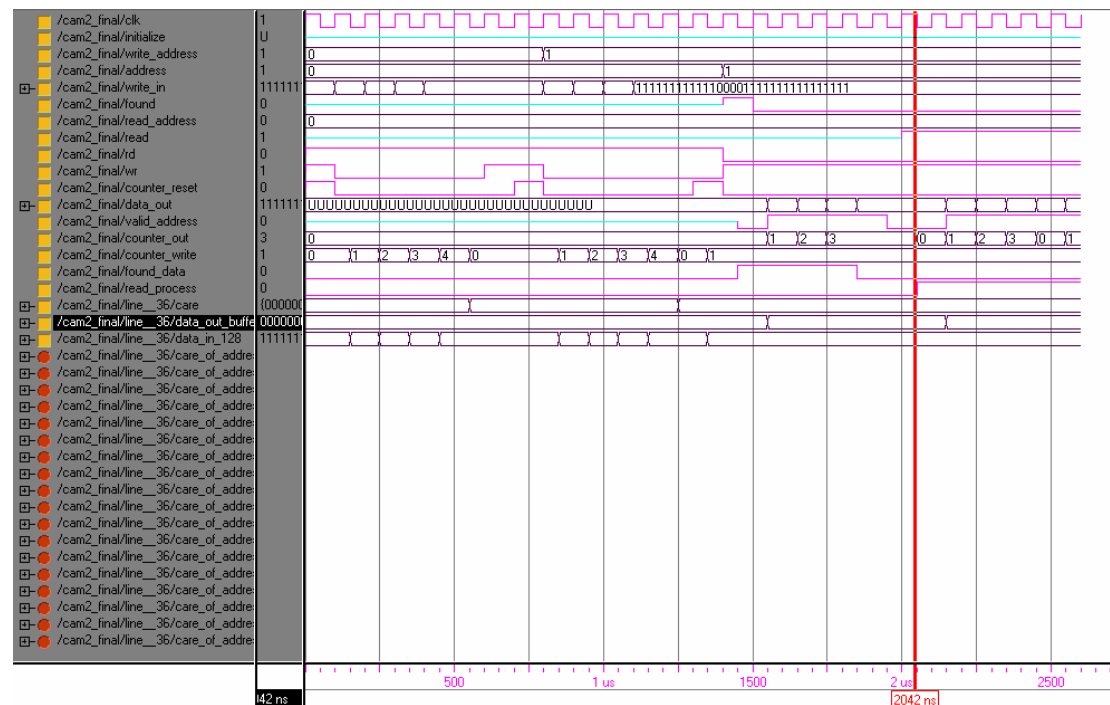
Έχουμε αναφέρει ότι για να σχηματίσουμε μια CAM θα πρέπει να συνυπάρχουν στην μονάδα 2 υπομονάδες εξομοιώσεις των οποίων υπάρχουν παρακάτω.

5.6.1 CAM1



Παρατηρούμε ότι πριν από κάθε εγγραφή ή ανάγνωση θα πρέπει να θέσουμε τουλάχιστον για ένα ρολόι το σήμα counter reset ώστε τα δεδομένα τα οποία ανταλλάσσονται να μπου η να δοθούν στην κατάλληλη στιγμή ώστε να πραγματοποιούνται οι διαδικασίες ασφαλώς. Όπως παρατηρούμε οι μετρητές παίρνουν τιμές από 0 έως 4 καθώς στο 0 γίνεται μηδενισμός τους και από 1 έως 4 γίνεται μεταφορά των δεδομένων 1st 32 bit, 2nd 32 bit, 3rd bit, 4th bit. Το είδος της διαδικασίας που πραγματοποιείται μπορούμε να το διακρίνουμε από τα σήματα rd, wr. Αν το σήματα είναι rd = '0', wr = '1' τότε έχουμε διαδικασία στην οποία διαβάζουμε ένα δεδομένο από την μνήμη, δηλαδή ψάχνουμε αν υπάρχει ο κινητός χρήστης με την διεύθυνση mobile node home link στον πίνακα δηλαδή αν εξυπηρετείται η όχι. Η διεργασία εύρεσης αν εξυπηρετείτε μια διεύθυνση από το σύστημα διαρκεί ένα κύκλο ρολογιού από τη στιγμή που μπει η διεύθυνση σε 4 κύκλους ρολογιού ανά 32 bit σε κάθε κύκλο.

Η συγκεκριμένη μονάδα κρατάει πίνακα με την care of address κάθε κινητού κόμβου και η αντιστοίχιση είναι άμεση δηλαδή στην 3 θέση της μνήμης θα υπάρχει η care of address του κινητού χρήστη του οποίου η home link address βρίσκεται στην 3 θέση του πίνακα της πρώτης μονάδας.



Παρατηρούμε στην παραπάνω εξομοίωση αναγνώσεις και εγγραφές στην μνήμη του συστήματος. Συγκεκριμένα αν έχουμε:

- Binding Update >> γίνεται εγγραφή της νέας care of address.
- Packet Encapsulation >> γίνεται ανάγνωση της care of address.

Παρακάτω παρουσιάζεται η μονάδα κατά την διάρκεια ταυτόχρονης εγγραφής στους πίνακες της πρώτης και δεύτερης υπομονάδας. Γίνεται εγγραφή στις θέσεις 0, 15 και 31 σε 5 στάδια η κάθε μια όπως περιγράφηκε στο προηγούμενο κεφάλαιο. Στο δεύτερο μέρος της εξομίωσης γίνεται διαδικασία ανίχνευσης δεδομένου, όπως παρατηρούμε η διεύθυνση που εισάγουμε αναγνωρίζεται ότι υπάρχει με το σήμα μεταξύ των δυο μονάδων found_int στην θέση 31 και γίνεται ανάγνωση της δεύτερης μνήμης στη θέση 31 και ανά clk βγαίνει η διεύθυνση care of address στην έξοδο της μονάδας.



Κεφάλαιο 6^ο

Συμπεράσματα – Βελτιώσεις του Συστήματος

Στο παρόν κεφάλαιο, καταγράφονται τα τελικά συμπεράσματα, όσον αφορά την παρούσα υλοποίηση αλλά και προτάσεις που έχουν γίνει για περαιτέρω βελτιώσεις του πρωτοκόλλου σε κάποια ανοιχτά θέματα. Αξίζει να σημειωθεί, ότι παράλληλα με την ανάπτυξη του Mobile IPv6 μεγάλες εταιρίες όπως η Sony και η Motorola διεξάγουν έρευνα για την ανάπτυξη δικών τους πρωτοκόλλων τα οποία θα χρησιμοποιηθούν για την υποστήριξη της κινητικότητας των χρηστών και ότι αρκετές τέτοιες υλοποιήσεις μπορεί να βρει κανείς στο διαδίκτυο.

6.1 Βελτιώσεις της παρούσας υλοποίησης

Το σύστημα Mobile IPv6 σχεδιάστηκε και υλοποιήθηκε με την γλώσσα περιγραφής υλικού VHDL. Η υλοποίηση του πρωτοκόλλου θα μπορούσε να πραγματοποιηθεί και σε λογισμικό. Υπάρχουν σαφή πλεονεκτήματα και μειονεκτήματα στην υλοποίηση ενός συστήματος σε hardware αντί σε software. Το βασικότερο πλεονέκτημα το οποίο υπάρχει είναι η καλύτερη απόδοση του συστήματος ειδικά αν χρησιμοποιηθούν τεχνικές βελτιστοποίησης του συστήματος όπως *pipelining* και *parallel unit* όπως επίσης υπάρχει και το μειονέκτημα της σταθερότητας και δυσκολίας τροποποιήσεων του συστήματος σε σχέση με τις υλοποιήσεις σε λογισμικό.

Στα πλαίσια αυτής της διπλωματικής εργασίας καταβλήθηκε μεγαλύτερη προσπάθεια στην κατάστρωση της αρχιτεκτονικής του συστήματος, της κατανόησης και υλοποίησης του τηλεπικοινωνιακού πρωτοκόλλου παρά στην βελτιστοποίηση του συστήματος με προηγμένες τεχνικές βελτιστοποίησης καθώς το πρωτόκολλο βρίσκεται ακόμα σε ανάπτυξη. Θα μπορούσε η βελτιστοποίηση του συστήματος να αποτελέσει περαιτέρω εξέλιξη του συστήματος. Μια μέθοδο με την οποία θα μπορούσε να βελτιστοποιηθεί το σύστημα είναι η εύρεση των πιο γρήγορων module που στην προκειμένη περίπτωση είναι οι μονάδες Control Unit και Control Memory και η κατακερμάτιση των υπολοίπων μονάδων σε υπομονάδες με χρήση καταχωρητών ώστε η συχνότητα του συστήματος να βελτιωθεί και να κινηθεί γύρω από την τιμή των 60 MHz. Για περαιτέρω βελτίωση θα μπορούσαμε να προχωρήσουμε σε περαιτέρω κατακερματισμό ακόμα και των μονάδων Control και αύξηση της συχνότητας λειτουργίας του συστήματος.

Η συγκεκριμένη υλοποίηση μπορεί να αποτελέσει πλατφόρμα ανάπτυξης ολόκληρου του IP επιπέδου καθώς προσφέρει τις κατάλληλες διεπαφές με τις υλοποιήσεις του IPv6 και IPv4 και ολόκληρο το σύστημα μπορεί να λειτουργεί αρμονικά κάτω από την καθοδήγηση της Control Unit. Επίσης για την ολοκλήρωση του IP επιπέδου θα πρέπει να υλοποιηθούν και κάποιες επιπλέον μονάδες οι οποίες θα έχουν να φέρουν σε πέρας το έργο της ασφάλειας, την διαχείριση των κλειδιών και την αυθεντικοποίηση.

6.2 Βελτιώσεις του πρωτοκόλλου Mobile IPv6

Θα πρέπει να τονίσουμε ότι υπάρχει ανάγκη μείωσης των πακέτων τα οποία χάνονται λόγω των Mobile IPv6 Handovers. Οι καθυστερήσεις που ευθύνονται κυρίως για αυτά τα χαμένα πακέτα είναι:

- Η καθυστέρηση επικοινωνίας κινητού χρήστη με Home Agent.
- Η καθυστέρηση που σχετίζεται με την ανίχνευση κίνησης του κινητού χρήστη.
- Επιπλέον καθυστερήσεις στα handovers από το Data Link Layer.

Παρακάτω παρουσιάζονται μερικές τεχνικές οι οποίες έχουν ως σκοπό να λύσουν κάποια από τα παραπάνω προβλήματα.

6.2.1 Γρήγορα Handovers στο πρωτόκολλο Mobile IPv6

Θα πρέπει οι κινητοί χρήστες να μπορούν να ξέρουν το καινούργιο δρομολογητή και το network-prefix του πριν αποσυνδεθούν από το δρομολογητή που βρίσκονται εκείνη την στιγμή. Μια ράδιο σύνδεση η οποία επιτρέπει αυτή την λειτουργία επειδή ο κινητός χρήστης είναι σε θέση να λάβει μηνύματα Router Advertisement και από τους 2 δρομολογητές ταυτόχρονα (Αυτό μπορεί να συμβεί αν υπάρχει αρκετά μεγάλη αλληλοκαλυπτώμενη περιοχή στις κυψελίδες επικοινωνίας – overlapped cells). Παρόλου που πολλές μεθόδους για την διαδικασία handover έχουν αναπτυχθεί σε επίπεδο data link layer μπορούν να χρησιμοποιηθούν και αρκετές μέθοδοι στο επίπεδο IP.

6.2.1.1 Αναμονή Μετακίνησης και Handover Ενεργοποίηση

Ας υποθέσουμε ότι ένας κινητός χρήστης είναι συνδεδεμένος με ένα connectionless link δευτέρου επιπέδου, όπως το πρωτόκολλο IEEE802.11b. Όταν ο κινητός χρήστης διαπιστώσει ότι η ενέργεια του σήματος του Access Point στο οποίο είναι συνδεδεμένος αρχίζει να εξασθενεί θα πρέπει να αρχίσει την αναζήτηση νέου AP. Αυτό ο κινητός χρήστης το κάνει ψάχνοντας όλες τις συχνότητες με βάση το πρωτόκολλο IEEE802.11b και συγκρίνει την ενέργεια του σήματος και την ποιότητα επικοινωνίας σε σχέση με αυτή που έχει συνδεδεμένος στο δεδομένο AP. Αν ο κινητός χρήστης ανακαλύψει κάποιο AP το οποίο του παρέχει καλύτερο σήμα τότε πρέπει να αλλάξει σε αυτό. Αν και η υλοποίηση data link layer του κινητού χρήστη δεν γνωρίζει αν το καινούργιο AP είναι συνδεδεμένο στον ίδιο ή σε άλλο Access Router καθώς γνωρίζει μόνο τα ονόματα των AP και τις MAC διευθύνσεις. Η υλοποίηση IP του κινητού χρήστη μπορεί να ζητήσει από τον AR (Access Router) να παρέχει το network prefix και την διεύθυνση του AR στο οποίο συνδέεται το AP. Η ιδέα που υλοποιείται είναι η ύπαρξη ενός πίνακα δεδομένων σε κάθε Access Router ο οποίος να περιέχει τις MAC των AP κάθε γειτονικού AR. Αυτό το σενάριο είναι ρεαλιστικό στην περίπτωση που ο κινητός χρήστης κινείται σε μια περιοχή ενός διαχειριστή δικτύου. Ο πίνακας που ακολουθεί είναι ένας πίνακας από αυτούς που περιγράψαμε.

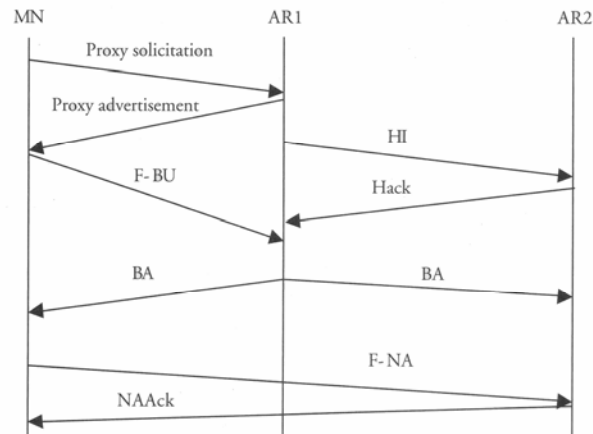
Router	AP MAC Address
Router_1	00601DF02CB2
	00901DF02CB2
	00901DF02AB2
Router_2	00601DA02CB2
	00601DA02DB2

Αν ο κινητός χρήστης ο οποίος είναι συνδεδεμένος στον δρομολογητή 1 και ζητήσει πληροφορίες για το AP με διεύθυνση 00601DF02CB2 ο δρομολογητής τον ενημερώνει ότι βρίσκεται συνδεδεμένος στον δρομολογητή 1 επίσης. Όμως αν ζητήσει περαιτέρω πληροφορίες για το AP με MAC address 00601DA02CB2 τότε ο δρομολογητής 1 θα ειδοποιήσει τον κινητό χρήστη ότι είναι συνδεδεμένος στον δρομολογητή 2. Με αυτό τον τρόπο γνωρίζει ο κινητός χρήστης ότι ένα IPv6 Handover θα συμβεί μέσα στο επόμενο άμεσο διάστημα. Το μήνυμα το οποίο περιέχει την MAC address ενός ή περισσότερων AP ονομάζεται **router solicitation for proxy** ενώ η απάντηση από τον δρομολογητή ονομάζεται **proxy router advertisement**. Πρέπει να παρατηρήσουμε ότι στις connectionless συνδέσεις όπως αυτές στο πρωτόκολλο IEEE802.11b ο κινητός χρήστης πρέπει να προκαλέσει το μήνυμα router advertisement γιατί είναι ο μόνος που γνωρίζει ότι χάνει την καλή επαφή του με κάποιο AP και αποκτά με κάποιο άλλο. Αντίθετα στις connection-oriented συνδέσεις το κάθε AP γνωρίζει ότι ο κινητός χρήστης θα αλλάξει AP.

6.2.1.2 Η ενημέρωση του Access Router

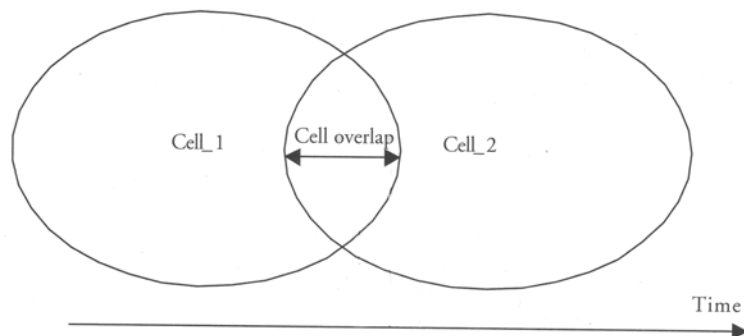
Από την στιγμή που έχουμε προσμονή μετακίνησης ο κινητός χρήστης θα γνωρίζει το network prefix και θα μπορεί να σχηματίσει μια care-of-address ενώ βρίσκεται συνδεδεμένος ακόμα στον δρομολογητή 1. Ο κινητός χρήστης θα ήθελε επίσης να γνωρίζει αν η συγκεκριμένη διεύθυνση είναι έγκυρη στο link, ωστόσο αυτό δεν μπορεί να συμβεί μέχρι να μετακινηθεί στο νέο Access Router.

Έχουμε περιγράψει ότι στα έργα του Mobile IPv6 συγκαταλέγεται και η ενθυλάκωση πακέτων στην care of address του κινητού χρήστη. Ο παρών AR μπορεί να ενσωματώνει λειτουργίες home agent. Ο κινητός χρήστης μπορεί να στείλει μήνυμα binding update στο παρών της access server για να στέλνει τα πακέτα ενθυλακωμένα στην καινούργια care-of-address. Αυτό σημαίνει ότι ο Access Router που εξυπηρετεί τον κινητό χρήστη θα λειτουργήσει σαν προσωρινός home agent για την παρούσα διεύθυνση του κινητού χρήστη. Ωστόσο για να αποφύγουμε την ύπαρξη σύγκρουσης διευθύνσεων στο καινούργιο link ο δρομολογητής ελέγχει την εγκυρότητα της διεύθυνσης care of address αποστέλλοντας μήνυμα Handover Initiate (HI) στο δρομολογητή 2. Αυτό το μήνυμα εξυπηρετεί δυο λόγους, πρώτον ζητά να ελέγξει ο δρομολογητής 2 αν είναι ενήμερος κάποιας αλληλοκάλυψης στην διεύθυνση care –of – address την οποία σχημάτισε αυθαίρετα το κινητός χρήστης και δεύτερον δημιουργεί ένα tunnel μεταξύ των τωρινό AR και τον καινούργιο. Από την στιγμή που εγκαταστηθεί το tunnel ο καινούργιος AR απαντάει με ένα Handover Acknowledgement (Hack) μήνυμα στον τωρινό AR. Το μήνυμα F-BU το οποίο παρουσιάζεται παρακάτω χρησιμοποιείται για να ενημερώσει τον AR1 ότι η κίνηση που προορίζεται για τον κινητό χρήστη πρέπει να προωθείτε στο AR2.



6.2.2 Το κόστος της αναμονής Μετακίνησης

Η μέθοδος του γρήγορου handover στηρίζεται στην αναμονή της μετακίνησης που περιγράψαμε. Αν η αναμονή είναι επιτυχημένη τότε ο χρόνος αναγνώρισης κίνησης μπορεί να ελαχιστοποιηθεί. Επίσης ο χρόνος ενημέρωσης του Home Agent και των Correspondent μπορεί να εξαφανιστεί από το χρόνο handover, καθώς η κυκλοφορία επαναδρομολογείται στην καινούργια θέση του κινητού χρήστη καθώς ενημερώνονται οι παραπάνω κόμβοι. Ωστόσο η αναμονή όπως και κάθε επιλογή έχει και τα αρνητικά της στοιχεία τα οποία ο μηχανικός πρέπει να συνυπολογίσει για να δημιουργήσει την βέλτιστη λύση.



Αν το cell overlap είναι πολύ μικρό, ο κινητός χρήστης δεν έχει πολύ χρόνο να επιβεβαιώσει ότι βρίσκεται στο cell 2 και να ενημερώσει το δίκτυο. Ο κινητός χρήστης μπορεί να χάσει την σύνδεση του πολύ γρήγορα (ειδικά όταν κινείται με μεγάλες ταχύτητες). Επιπλέον το δίκτυο δεν έχει αρκετό χρόνο να προετοιμάσει το handover. Αντίθετα αν το cell overlap είναι πολύ μεγάλο υπάρχει πολύ μεγάλη παρεμβολή μεταξύ των δυο cell το οποίο προκαλεί την μείωση της περιοχής κάλυψης και μικρότερη χωρητικότητα. Για τους παραπάνω λόγους οι σχεδιαστές του συστήματος θα πρέπει να κάνουν το trade-off και να διαλέξουν μια επικάλυψη που να είναι αρκετά μεγάλη ώστε να παρέχει αρκετό χρόνο στον κινητό χρήστη να αντιλαμβάνεται την κινητικότητά του, αλλά όχι πολύ μεγάλη γιατί θα μειωνόταν η χωρητικότητα της κυψέλης.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Βιβλία και Άρθρα

- James D. Solomon, Mobile IP – The Internet Unplugged
- Hesham Soliman, Mobile IPv6 – Mobility in a Wireless Internet
- Rachid Ait Yaiz and Osman Oyturk, Mobility in IPv6
- Charlie Perkins, Mobile IPv6 and Cellyler Telephony
- David B. Johnson, Mobile IPv6 Status
- David B. Johnson, IPv6 Changes in Mobile IPv6
- A. Garyfalos, K. Almeroth, J. Finney, A Comparison of Networks and Application Layer for Mobile IPv6 Networks
- Gloria Yuquerres, Marcos Rogerio Salvador and Ron Sperkels, Mobile IP: Security and Applications
- Cisco Certified Network Associate – Professional Resources
- LIN-IETF(Sony), A Solution to Multi-Homing and Mobility in IPv6
- EUROSCOM(Motorola), Support of Mobile Networks
- INRIA(Motorola), Mobile Networks Supporting IPv6

References

- RFC 1256 – ICMP Router Discovery Message
- RFC 1321 – The MD5 Message – Digest Algorithm
- RFC 1700 – Assigned Numbers
- RFC 1883 – IPv6 Specifications
- RFC 1924 – A Compact Representation of IPv6 Addresses
- RFC 1970 – Neighbour Discovery for IPv6
- RFC 1971 – IPv6 Stateless Address Autoconfiguration
- RFC 2003 – IP Encapsulation within IP
- RFC 2004 – Minimal Encapsulation within IP
- RFC 2460 – IPv6 Specification
- RFC 2461 – Neighbour Discovery for IPv6
- RFC 2463 – ICMP for IPv6 Specification
- RFC 2473 – Generic Packet Tunneling in IPv6 Specifications
- RFC 2526 – Reserved IPv6 Subnet Anycast Addresses
- RFC 3306 – Unicast – Prefix – based Multicast Addresses
- RFC 2131 – Dynamic Host Configuration Protocol

Internet Resources

- <http://www.ipv6.org>
- <http://www.ipv6-net.org>

ΤΙΤΛΟΣ: Υλοποίηση σε Υλικό του Πρωτοκόλλου Mobile IPv6 - Home Agent.

Φοιτητής : Μεσσήνης Θεόδωρος

Επιβλέπων : Κουφοπαύλου Οδυσσέας
Καθηγητής

Περίληψη

Αντικείμενο της παρούσας διπλωματικής εργασίας, είναι η σχεδίαση αρχιτεκτονικής και η υλοποίηση σε υλικό με την γλώσσα περιγραφής υλικού VHDL του τηλεπικοινωνιακού πρωτοκόλλου Mobile IPv6 για τον Home Agent. Ο Home Agent είναι ένας δρομολογητής ο οποίος βρίσκεται στο Home Link του κινητού χρήστη και αναλαμβάνει να διατηρεί το binding με τον κινητό χρήστη ώστε να μπορεί να είναι δυνατή η επιθυμητή κινητικότητα. Το πρωτόκολλο Mobile IPv6 επιτρέπει την μετάβαση του κινητού χρήστη από ένα Foreign link σε κάποιο άλλο Foreign link χωρίς την διακοπή της επικοινωνίας.

Ένα μεγάλο μέρος του έργου που είχε να πραγματοποιηθεί ήταν η σχεδίαση κατάλληλης αρχιτεκτονικής η οποία θα μπορούσε να πραγματοποιήσει βέλτιστα τις λειτουργίες του πρωτοκόλλου. Δόθηκε μεγάλη έμφαση στον σχεδιασμό structural αρχιτεκτονικής η οποία επιτρέπει την μετέπειτα εύκολη κατανόηση και βελτίωση του Συστήματος. Επίσης το Σύστημα που υλοποιήθηκε ουσιαστικά αποτελεί μια πλατφόρμα ανάπτυξης ολόκληρου του IP Layer καθώς παρέχει τις κατάλληλες διεπαφές για το IPv6 και IPv4.

Για την υλοποίηση του Συστήματος σε υλικό χρησιμοποιήθηκαν η γλώσσα περιγραφής υλικού VHDL και μια σειρά από εργαλεία τα οποία υποστηρίζουν την εξομοίωση και σύνθεση του παραγόμενου κώδικα. Στα ειδικά κεφάλαια της εργασίας αναφέρονται τα αποτελέσματα των εξομοιώσεων και της σύνθεσης του Συστήματος.

Αρχικά αναφέρονται τα θεωρητικά του πρωτοκόλλου. Στην συνέχεια δίνουμε μια σαφή περιγραφή της αρχιτεκτονικής και των λειτουργιών του Συστήματος. Ακολουθεί η Σύνθεση του Συστήματος και οι Εξομοιώσεις οι οποίες επιβεβαιώνουν την ορθότητα της λειτουργίας του. Τέλος αναφέρονται οι βελτιώσεις που μπορούν να πραγματοποιηθούν στο Σύστημα αλλά και στο πρωτόκολλο. Στο Παράρτημα υπάρχει ο πηγαίος κώδικας.