

Βελτίωση του οδηγού για ασύρματες κάρτες Atheros που εμπεριέχεται στον πυρήνα του Linux (ath5k) Final Report

Αρχικά επέλεξα να αρχίσω με το calibration του AGC (Automatic Gain Control) στα chips 2ης γενιάς της Atheros (RF5111/RF5112 + AR5212) όπως ανέφερα και στο χρονοδιάγραμμα καθότι πρόκειται για μια σχετικά απλή διαδικασία, είχαμε ήδη κάποιον κώδικα και έπρεπε να τον ξαναγράψω σωστά και να κάνω review την όλη διαδικασία. Σε μέσες άκρες η κάρτα χρειάζεται έναν τρόπο να μετράει τις μεταβολές στο κέρδος (gain) του αναλογικού μέρους της, για να το πετύχουμε αυτό δυναμικά στέλνουμε περιοδικά κάποια πακέτα με μικρότερη ισχύ απ' την ισχύ που εκπέμπει κανονικά η κάρτα, και το AGC χρησιμοποιώντας έναν Peak to Average Power Detector (PAPD) μετράει τη μεταβολή του κέρδους (που συνήθως οφείλεται σε θερμικές μεταβολές κλπ) και μας δίνει ένα feedback το οποίο και χρησιμοποιούμε όταν προγραμματίζουμε την κάρτα (όταν κάνουμε reset). Τα patches που αφορούν το συγκεκριμένο κώδικα είναι τα παρακάτω...

<http://git.kernel.org/?p=linux/kernel/git/linville/wireless-testing.git;a=commit;h=a985bab13f54b41b99e9ad0d3e89ee04cff40038>

<http://git.kernel.org/?p=linux/kernel/git/linville/wireless-testing.git;a=commit;h=5b25a2088eef98ca3c8526f4b71c501833636f68>

<http://git.kernel.org/?p=linux/kernel/git/linville/wireless-testing.git;a=commit;h=212d498752d76ef378d2db1a0803993f358315a3>

Όπως ανέφερα και στο χρονοδιάγραμμα και μπορείτε να δείτε και στα παραπάνω patches επέλεξα στο πλαίσιο των γενικότερων αλλαγών στον κώδικα που αφορά το PHY/RF κομμάτι της κάρτας να κάνω εκκαθάριση και ουσιαστικά επανεγγραφή ενός μεγάλου μέρους του κώδικα. Ο λόγος είναι ότι ο κώδικας που προϋπήρχε ήταν ασαφής και χωρίς τεκμηρίωση, ενώ πολλά κοινά χαρακτηριστικά των διαφόρων chip μπορούσαν να γίνουν handled με τον ίδιο κώδικα, χρησιμοποιώντας μερικές έξυπνες πρακτικές. Επειδή η εκκαθάριση του κώδικα δεν θεωρείται “functional change” είναι σε 1-2 patches γιατί δεν περνάει ιδιαίτερο review (έτσι είναι το policy στον kernel για να μη γεμίζουμε τα logs).

Κατόπιν συνέχισα κάνοντας update τον κώδικα που αφορούσε το reset της κάρτας, ξεκίνησα ανανεώνοντας τα assays με τις αρχικές ρυθμίσεις (αρχικές τιμές των βασικών καταχωρητών της κάρτας -οι υπόλοιποι μηδενίζονται κατά το reset) όπως φαίνεται σε αυτό το patch...

<http://git.kernel.org/?p=linux/kernel/git/linville/wireless-testing.git;a=commit;h=8be5e97a252d1a77d901ce340553992c30661d60>

...και συνέχισα κάνοντας και πάλι πρώτα εκκαθάριση (έσπασα την συνάρτηση hw_reset σε περισσότερες, έκανα μια γενικότερη ανακατάταξη κλπ) και προσθέτοντας διάφορα τμήματα που έλειπαν στη διαδικασία με ποιο σημαντικό ίσως την ρύθμιση των ρολογιών της κάρτας για το power saving (εφόσον υπάρχει εξωτερικός κρύσταλλος χαμηλής συχνότητας/κατανάλωσης, η κάρτα χρησιμοποιεί αυτόν για ρολόι αναφοράς κι έτσι μπορεί να πέφτει σε power save mode και να επανέρχεται χωρίς να δημιουργείται πρόβλημα με τους διάφορους timers)...

<http://git.kernel.org/?p=linux/kernel/git/linville/wireless-testing.git;a=commit;h=a742001f1908a72bd01adc48ba4fd900af8f2978>

Κάνοντας debug τον παραπάνω κώδικα βρήκα και 2 bugs τα οποία έκαναν αρκετά ασταθή τον driver και τα οποία διόρθωσα. Τα patches βρίσκονται εδώ ->

<http://git.kernel.org/?p=linux/kernel/git/linville/wireless-testing.git;a=commit;h=0e814c9a23f18bb959ce7622d8ee69f1e73d3921>

<http://git.kernel.org/?p=linux/kernel/git/linville/wireless-testing.git;a=commit;h=732c666ef1b910c1429c7a805ebc5a2b1dc10e0a>

Αφού ολοκλήρωσα τα παραπάνω συνέχισα με την υλοποίηση του calibration του baseband έτσι ώστε να αλλάζει σωστά η ισχύ εκπομπής (txpower). Με απλά λόγια η κάρτα συνολικά (μαζί με τον προενισχυτή και όλη την υλοποίηση), έχει διαφορετική απόκριση σε κάθε κανάλι και δεν αλλάζει γραμμικά. Οπότε προκειμένου να αλλάζει γραμμικά η ισχύς (να υπάρχει μια σχέση $\chi = \psi$ μεταξύ της ισχύος που θέτουμε και της πραγματικής εκπεμπόμενης ισχύος) και να είναι σωστή κατά μέτρο, πρέπει να προγραμματίσουμε έναν πίνακα στο hw με κάποια offsets τα οποία χοντρικά ρυθμίζουν το έλειμα/πλεόνασμα ισχύος ώστε η κάρτα να μπορεί να κάνει κάθε φορά προσαρμογή. Ο μηχανισμός που υλοποίησα παραπάνω είναι συμπληρωματικός αυτού του μηχανισμού. Αυτός ο μηχανισμός είναι ιδιαίτερα πολύπλοκος και υλοποιήθηκε για όλα τα chips απ' την αρχή. Μια αναλυτικότερη περιγραφή μπορείτε να δείτε στο mail που έστειλα στην mailing list και συνόδευε το patch series...

<https://lists.ath5k.org/pipermail/ath5k-devel/2009-March/002062.html>

Αρχικά χρειάστηκε να γράψω τον κώδικα που διαβάζει τα δεδομένα για τη δημιουργία των διαφόρων tables (για τα διάφορα κανάλια που υποστηρίζει η κάρτα) απ' την EEPROM και να τα μετατρέψω σε ένα format κοινό για όλα τα chips (κάθε γενιά chip έχει διαφορετικά δεδομένα για το συγκεκριμένο calibration, πχ. οι νεότερες γενιές υποστηρίζουν και αρνητικές τιμές ισχύος, ενώ αλλάζει το resolution μεταξύ 0.5dB και 0.25dB)...

<http://git.kernel.org/?p=linux/kernel/git/linville/wireless-testing.git;a=commit;h=4438ab10db512815bb9ffcd2dda3732bfd095384>

...κατόπιν έγραψα απ' την αρχή όλο τον κώδικα για τη δημιουργία των tables καθώς και τον αρχικό κώδικα για regulatory control (χρήση των Conformance Test Limits που βρίσκονται στην EEPROM και ουσιαστικά πρόκειται για κάποια επιπλέον power limits που επιβάλουμε έτσι ώστε η κάρτα να εκπέμπει στα νόμιμα όρια ανάλογα με τη χώρα που βρίσκεται)...

<http://git.kernel.org/?p=linux/kernel/git/linville/wireless-testing.git;a=commit;h=e5903d87f0f82bbcb32d146ffb63c3566fc2f058>

...ο κώδικας αυτός μου πήρε περισσότερο καιρό απ' ότι υπολόγιζα καθότι δεν είχα ουσιαστικά documentation απ' την Atheros όσον αφορά τη διαδικασία ρύθμισης του txpower (πώς αφού προγραμματίσουμε τον πίνακα θέτουμε την ισχύ εκπομπής στην κάρτα), ενώ το documentation και ο κώδικας που είχα για την δημιουργία του πίνακα ήταν πολύ misleading και πολύ κακογραμμένα. Κοινός μου πήρε αρκετό καιρό να καταλάβω τι γίνεται για κάθε chip ξεχωριστά. Αυτό όμως είχε ως αποτέλεσμα ο κώδικας να βγει αρκετά καλός και με πολύ καλή τεκμηρίωση (στα πλαίσια του δυνατού), ενώ είναι ταυτόχρονα πολύ μικρότερος και απλούστερος από τον κώδικα της Atheros και όταν τον έκανα debug φάνηκε ότι είναι και ακριβής (δουλεύει δηλαδή).

Κι εδώ βρήκα ένα bug κατά το debugging το οποίο και διόρθωσα...

<http://git.kernel.org/?p=linux/kernel/git/linville/wireless-testing.git;a=commit;h=8629fbf8a13a514c6c4d47bafa2f8b29ce0ca04a>

Πριν επιτρέψουμε στον driver να θέσει κανονικά το txpower έπρεπε να υλοποιήσουμε το regulatory control (δηλαδή τους περιορισμούς ανάλογα με τα νόμιμα όρια της εκάστοτε χώρας για την οποία έχει προγραμματιστεί η κάρτα ή θέτει ο χρήστης ή ο driver -πχ. μέσω του 802.11d-). Στο επίπεδο του driver αυτό ουσιαστικά είχε ήδη γίνει με τον κώδικα που έγραψα για τα CTLs, το μόνο που έλειπε ήταν τα διάφορα mappings για κάθε χώρα και ο κώδικας που θα έκανε κατόπιν register τον driver στο υπόλοιπο CRDA framework του kernel (το framework που έχουμε για να κάνει handle το regulatory control για όλους τους drivers). Επειδή ο σχετικός κώδικας είναι κοινός μεταξύ του ath5k και του ath9k (του driver για τις .11n κάρτες της Atheros) αποφασίσαμε με το team του ath9k να κάνουμε merge και να χρησιμοποιούμε και οι δύο ένα νέο module που το ονομάσαμε ath. Επίσης οι δύο drivers μεταφέρθηκαν σε ένα καινούριο directory, το drivers/net/wireless/ath και αποφασίσαμε εκτός απ' τον κώδικα για το regulatory control να μοιραστούμε στο μέλλον και άλλα κομμάτια κώδικα, όπως πχ. τον αλγόριθμο για το ANI που αναφέρω στην πρόταση (το Adaptive Noise Immunity). Για τον αλγόριθμο αυτό έγραψα τον κώδικα (σε αυτό το patch ο Bob τον έχει κάνει rebase έτσι ώστε να γίνεται diffed με το current tree, οπότε φαίνεται ολόκληρος -> <https://lists.ath5k.org/pipermail/ath5k-devel/2009-March/002125.html>) αλλά τελικά δεν ενσωματώθηκε για να γίνει και το τελικό debugging και να φτιαχτεί το initialization (ο αλγόριθμος είναι εντάξει) μέχρι να αποφασίσουμε αν θα χρησιμοποιήσουμε τη δικιά μου έκδοση ή της Atheros. Εν το μεταξύ το debugging του κώδικα για το txpower καθώς και το debugging για τον κώδικα για την διαχείριση των κεραιών που έγραψα στη συνέχεια απεδείχθη αρκετά χρονοβόρο και πολύπλοκο. Αφού το regulatory control υλοποιήθηκε πλήρως κατά το merging απ' τον Bob κυρίως και από εμένα στο κομμάτι του phy. Έστειλα το τελικό patch που επιτρέπει τον ορισμό απ' τον χρήστη/driver της ισχύος και τον έλεγξα χρησιμοποιώντας όχι μόνο spectrum analyzer (όπου δεν έχουμε per-packet μέτρηση αλλά χρησιμεύει μόνο για να μετρήσουμε κατ' απόλυτη τιμή τα dBm εκπομπής και να δούμε την spectral mask) αλλά και με έναν network analyzer (το airlive, είναι κάτι σαν kismet + wireshark σε ένα λογισμικό). Το patch δεν έχει μπει ακόμα στο git tree αλλά το σχετικό mail για το commit είναι εδώ ->

<https://lists.ath5k.org/pipermail/ath5k-devel/2009-April/002258.html>

ένας απ' τους λόγους που ήθελα per-packet μετρήσεις ήταν και για να κάνω debug το TPC γενικότερα. Τελικά ο κώδικας που έχω βάλει για per-packet power control είναι εντάξει αλλά αφενός ο developer του HAL της Atheros (το binary blob που βρίσκεται στον MadWiFi) μας είπε ότι το per-packet power control δεν παίζει σωστά σε όλες τις κάρτες (γι' αυτό και είναι disabled by default) και αφετέρου στο protocol stack δεν έχουμε τέτοια πληροφορία (δεν θέτουμε πχ. ξεχωριστό txpower για τα acks ή για διαφορετικές tx queues). Έτσι αποφάσισα να τον αφήσω disabled by default μέχρι να γίνει debug από περισσότερο κόσμο.

Για την διαχείριση των κεραιών δεν είχα κάποιο κώδικα στη διάθεσή μου, ούτε είχε υλοποιηθεί κάτι αντίστοιχο στους binary drivers της Atheros για να κάνω reverse engineer/decompile. Είχα ευτυχώς docs και δεν ήταν ιδιαίτερα δύσκολο να υλοποιήσω αυτά που περιγράφονταν. Ωστόσο έχασα πολύ χρόνο για να υλοποιήσω το diversity σωστά (την δυναμική εναλλαγή μεταξύ των διαφόρων κεραιών που κάνει αυτόματα το hw για να διαλέξει την καλύτερη), καθώς για το συγκεκριμένο θέμα δεν είχα documentation και το μόνο που είχα ουσιαστικά ήταν μερικά comments στον κώδικα της Atheros. Τελικά βρήκα ότι έπρεπε να γράφω τους καταχωρητές με μια συγκεκριμένη σειρά για να μπορώ να κάνω σωστά enable/disable το diversity και έκανα εκτενές debug του κώδικα φτιάχνοντας ένα debug

mode στο οποίο εξανάγκαζα το hw να χρησιμοποιεί τη μία κεραία μόνο για rx και την άλλη μόνο για tx. Ο κώδικας δεν έχει μπει ακόμα στο git repository αλλά το σχετικό mail για το commit είναι εδώ ->

<https://lists.ath5k.org/pipermail/ath5k-devel/2009-April/002264.html>

Κατά το debugging τόσο του txpower όσο και του antenna control χρειάστηκε να χρησιμοποιήσω την κάρτα σε master mode (να την κάνω δηλαδή Access Point). Ενώ κάτι τέτοιο έχει θεωρητικά υλοποιηθεί στον driver υπήρχαν κάποια bugs στον κώδικα που αφορά το hw τα οποία βρήκα και έφτιαξα...

<https://lists.ath5k.org/pipermail/ath5k-devel/2009-April/002262.html>

και πρότεινα μετά κα από αυτό να ενεργοποιήσουμε και επίσημα το AP support στον driver καθότι είναι κάτι που ζητούν πολλοί χρήστες.

<https://lists.ath5k.org/pipermail/ath5k-devel/2009-April/002263.html>

Τέλος για την καλύτερη μικρορύθμιση (calibration) του PHY chip και συγκεκριμένα των συνιστωσών I και Q του QAM constellation υλοποίησα τον μηχανισμό για το φίλτρο spur θορύβου. Ο spur θόρυβος είναι κάτι τέτοιο -> <http://www.comlab.hut.fi/opetus/333/2004slides/topic23.pdf> (βλ. σελ. 10) και συνήθως δημιουργείται από διάφορες συσκευές που έχουν παλμούς και χαρακτηριστικά εκπομπής spread spectrum (πχ. κάποια rtc). Υπάρχουν τεχνικές λοιπόν για τη διόρθωση αυτού του φαινομένου πειράζοντας κάποιες παραμέτρους στο viterbi decoder και παίζοντας με τη διαφορά φάσης και το amplitude (http://zone.ni.com/cms/images/devzone/tut/dhall_hwspec_im7.JPG) των συνιστωσών I/Q καθώς και το pilot tracking (κάτι σαν training που κάνει η κάρτα όταν της έρχεται ένα frame, κάθε frame πριν αρχίσουν τα data έχει το preamble που είναι κάτι γνωστό, η κάρτα που ξέρει πως πρέπει να είναι το preamble διορθώνει τα H/Q έτσι ώστε όταν φτάσει στα data να είναι σίγουρη ότι διαβάζει σωστά).

Στον driver αρχικά έπρεπε να διαβάσουμε απ' την EEPROM τα κανάλια τα οποία επηρεάζονται απ' τον spur θόρυβο (εκτός απ' τα κανάλια που ξέρουμε ότι επηρεάζονται λόγω καναλιού απ' το ref clock της κάρτας) και κατόπιν έπρεπε να κάνουμε κάποιους υπολογισμούς, να γράψουμε τις παραμέτρους για το spur φίλτρο στο hw και να το κάνουμε enable. Τα σχετικά patches επίσης δεν έχουν μπει ακόμα στο git repository αλλά τα σχετικά mails για το commit είναι εδώ ->

<https://lists.ath5k.org/pipermail/ath5k-devel/2009-April/002259.html>

<https://lists.ath5k.org/pipermail/ath5k-devel/2009-April/002265.html>

Αυτό που απομένει και αναφέρω στην πρότασή είναι η εκκαθάριση του κώδικα της EEPROM, αυτό έχει γίνει τμηματικά στα παραπάνω patches που πειράζουν τον κώδικα της EEPROM αλλά ολοκληρώνεται με το παρακάτω patch

<https://lists.ath5k.org/pipermail/ath5k-devel/2009-April/002261.html>

Για όποια άλλη διευκρίνηση είμαι στη διάθεσή σας ;-)

Νίκος Κοσσυφίδης
Ρούμελης 38 Χαλάνδρι/ Αθήνα
κιν: +306973695206
ΑΦΜ: 137223950