

UNIVERSITY OF MACEDONIA
MASTER IN INFORMATION SYSTEMS

Active Networks Architecture

by

Anthimos Zachariadis

Course: Networking Technologies
Professor : A.A. Economides

Thessaloniki
January 2002

ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΠΜΣ ΣΤΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

Η αρχιτεκτονική των ενεργών δικτύων

του φοιτητή

Άνθιμου Ζαχαριάδη

Μάθημα: Τεχνολογίες Τηλεπικοινωνιών και δικτύων
Καθηγητής: Α.Α. Οικονομίδης

Θεσσαλονίκη
Ιανουάριος 2002

ΠΕΡΙΕΧΟΜΕΝΑ

1. Abstract – Περίληψη
2. Εισαγωγή
3. Η αρχιτεκτονική των ενεργών δικτύων
4. Ο ενεργός κόμβος σε επίπεδο υλικού (Hardware)
5. Το λειτουργικό σύστημα του κόμβου (NodeOS)
6. Τα περιβάλλοντα εκτέλεσης (Execution Environments, EEs)
 - 6.1 Active Network Transport System (ANTS)
 - 6.2 Programming Language for Active Networks (PLAN)
 - 6.3 Distributed Code Caching for Active Networks (DAN)
7. Ενεργές εφαρμογές (Active Applications)
8. Active Network Encapsulation Protocol (ANEP)
9. Active Network Backbone, ABONE
10. Συμπεράσματα - Επίλογος

1. Abstract

Active Networking architecture was created in order to address the problem of slow network evolution compared with the evolution of applications. These networks are "active" in two ways: routers can perform computations on user data flowing through them, while users can program the network by supplying their own code to perform computations. The code is being carried in the packets and is executed in a programmable router. The architecture is composed of three layers: the node operating system layer, the execution environment layer and the active application layer. The first layer is responsible for resource management, the second is a virtual machine which interprets the code of packets and the third is responsible for programming the execution environment for the supply of service. ABONE is the first active network implemented only for experimental purposes by US army.

1.Περίληψη

Η αρχιτεκτονική των ενεργών δικτύων δημιουργήθηκε για να αντιμετωπίσει το πρόβλημα της αργής ανάπτυξης των δικτύων σε σχέση με την ανάπτυξη των εφαρμογών. Τα δίκτυα αυτά είναι ενεργά με δύο τρόπους: οι δρομολογητές μπορούν να εκτελέσουν υπολογισμούς στα δεδομένα των χρηστών καθώς περνούν από μέσα τους, ενώ οι χρήστες μπορούν να προγραμματίσουν το δίκτυο παρέχοντας τον δικό τους κώδικα για να κάνει τους υπολογισμούς αυτούς. Ο κώδικας μεταφέρεται μέσα στα πακέτα και εκτελείται μέσα στον προγραμματιζόμενο δρομολογητή. Η αρχιτεκτονική αποτελείται από τρία επίπεδα: το επίπεδο του λειτουργικού συστήματος του κόμβου, το επίπεδο του περιβάλλοντος εκτέλεσης και το επίπεδο ενεργούς εφαρμογής. Το πρώτο επίπεδο είναι υπεύθυνο για την διαχείριση των πόρων, το δεύτερο αποτελεί μια εικονική μηχανή όπου μεταφράζεται ο κώδικας των πακέτων, και το τρίτο είναι υπεύθυνο για τον προγραμματισμό του περιβάλλοντος εκτέλεσης για την παροχή της υπηρεσίας. Το ABONE είναι το πρώτο ενεργό δίκτυο που αναπτύχθηκε για ερευνητικούς σκοπούς μόνο από τον Αμερικάνο στρατό.

2. Εισαγωγή

Η ιδέα των «Ενεργών δικτύων» (Active Networks) γεννήθηκε κατά την διάρκεια συζητήσεων για την πορεία των δικτύων, από την ερευνητική κοινότητα του DARPA¹, κατά τα έτη 1994 και 1995. Κατά τις συζητήσεις αυτές εντοπίστηκαν διάφορα προβλήματα σχετικά με την φιλοσοφία των δικτύων όπως: η εισαγωγή νέων τεχνολογιών και προδιαγραφών στην υπάρχουσα δομή του δικτύου, η χαμηλή απόδοση λόγω πλεονάζουσων λειτουργιών από τα διάφορα πρωτόκολλα των επιπέδων και κυρίως η δυσκολία εισαγωγής νέων υπηρεσιών στο υπάρχων αρχιτεκτονικό μοντέλο. Ο κύριος σκοπός του οργανισμού ήταν η εξέλιξη των αμυντικών δικτύων μέσα από την άμεση ανάπτυξη νέων στρατηγικών και την προσαρμογή της υποδομής στις απαιτήσεις της στιγμής. Έχοντας ως βάση τους παραπάνω προβληματισμούς ο DARPA ξεκίνησε το ερευνητικό πρόγραμμα των ενεργών δικτύων που ως στόχο είχε «την παραγωγή μιας νέας δικτυακής πλατφόρμας, ευέλικτης και επεκτάσιμης σε ελάχιστο χρόνο έτσι ώστε να διευκολύνει την γρήγορη εξέλιξη και ανάπτυξη των δικτυακών τεχνολογιών καθώς επίσης και να παρέχει τις αυξανόμενα πολύπλοκες υπηρεσίες που απαιτούνται από τις αμυντικές εφαρμογές» [1].

Την ίδια εποχή που ο DARPA ξεκινούσε το ερευνητικό πρόγραμμα των «Ενεργών Δικτύων», από πολλούς ερευνητές είχαν εκφραστεί προβληματισμοί για την πορεία ανάπτυξης του Internet. Συγκεκριμένα αναγνωρίστηκαν προβλήματα ευελιξίας, αφού η ενσωμάτωση νέων χαρακτηριστικών χρειαζόταν πολλές φορές έως και δέκα χρόνια, κυρίως λόγω της μακράς διαδικασίας προτυποποίησης (standardization) και της χρονοβόρας εγκατάστασης στα φυσικά στοιχεία του δικτύου (routers, switches). Κατά δεύτερο λόγο, υπηρεσίες όπως η πολλαπλή αποστολή (multicasting) και η τηλεδιάσκεψη, δεν αναπτύχθηκαν στο βαθμό που αναμενόταν αφού το διαδίκτυο αντιμετώπιζε όλες τις εφαρμογές με τον ίδιο τρόπο, χωρίς να αναγνωρίζει την σημαντικότητά τους. Οι λόγοι που οι απαιτητικές σε εύρος ζώνης εφαρμογές δεν μπόρεσαν να αναπτυχθούν σε ικανοποιητικό βαθμό ήταν: οι μεγάλοι και απρόβλεπτοι χρόνοι αναμονής, τα μεγάλα ποσοστά απόρριψης, οι μη βέλτιστη επιλογή διαδρομής κ.α..

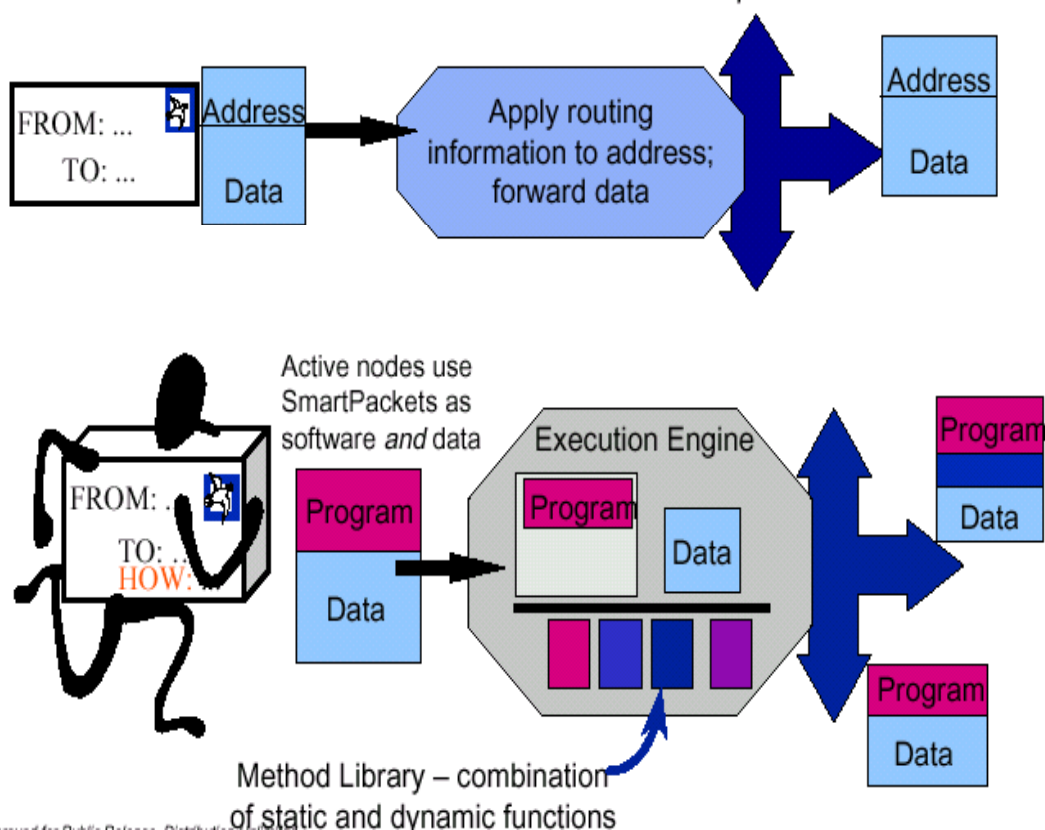
Συμπερασματικά λοιπόν, θα μπορούσε να ειπωθεί ότι η ιδέα των ενεργών δικτύων γεννήθηκε από την αδυναμία των δικτυακών υποδομών να ακολουθήσουν τον ρυθμό ανάπτυξης των εφαρμογών και κατ' επέκταση των απαιτήσεων των χρηστών.

Τα ενεργά δίκτυα αποτελούν ένα βήμα πέρα από την παραδοσιακή δικτυακή τεχνολογία όπου οι δρομολογητές έχουν τον ρόλο της παθητικής προώθησης πακέτων και της ανανέωσης των πινάκων δρομολόγησης. Σύμφωνα με τον ορισμό του MIT [4] «τα ενεργά δίκτυα επιτρέπουν σε μεμονωμένους χρήστες ή σε ομάδες χρηστών να εισάγουν προσαρμοσμένα προγράμματα μέσα στους κόμβους του δικτύου. Η «ενεργή» αρχιτεκτονική επιτρέπει την μαζική αύξηση της πολυπλοκότητας και της προσαρμοστικότητας των υπολογισμών που διεξάγονται μέσα στο δίκτυο».

¹ Ο DARPA (Defense Advanced Research Projects Agency) είναι ο κύριος ερευνητικός οργανισμός του Αμερικάνικου υπουργείου Άμυνας και διεξάγει εφαρμοσμένη έρευνα και ανάπτυξη για την βελτίωση των στρατιωτικών αποστολών.

Ο χαρακτηρισμός των δικτύων ως ενεργά δόθηκε κυρίως για δύο λόγους: οι δρομολογητές και οι μεταγωγείς μέσα στο δίκτυο μπορούν να δράσουν, δηλαδή να διεξάγουν υπολογισμούς στα δεδομένα των χρηστών καθώς περνούν, ενώ και οι χρήστες μπορούν να προγραμματίσουν το δίκτυο παρέχοντας τα δικά τους προγράμματα για διεξάγουν υπολογισμούς.

Η επανάσταση που έφερε η ιδέα των ενεργών δικτύων ήταν κυρίως στον τρόπο δρομολόγησης των πακέτων. Στα δίκτυα όπου χρησιμοποιείται το IP πρωτόκολλο, οι δρομολογητές εξετάζουν το πεδίο διεύθυνσης προορισμού της IP κεφαλίδας, σε σχέση με τους εσωτερικούς πίνακες δρομολόγησης, έτσι ώστε να καθορίσουν σε ποια γειτονική τοποθεσία θα προωθήσουν το IP πακέτο. Έτσι ο έλεγχος του χρήστη στο δίκτυο περιορίζεται στο εύρος των τιμών που μπορούν να τοποθετηθούν στην IP κεφαλίδα. Άλλωστε η βασική επεξεργασία (μέχρι και το επίπεδο εφαρμογής) στο δίκτυο γίνεται στους «έξυπνους» κεντρικούς υπολογιστές (hosts) που βρίσκονται στα άκρα της δομής. Οι δρομολογητές που διασυνδέουν τους υπολογιστές αυτούς έχουν ελάχιστες δυνατότητες επεξεργασίας (το πολύ μέχρι το επίπεδο Δικτύου). Αντίθετα οι δρομολογητές ενός ενεργού δικτύου μπορούν να εκτελούν υπολογισμούς κάτω από τον έλεγχο των τελικών χρηστών μέχρι και το επίπεδο εφαρμογής, πράγμα που σημαίνει ότι μπορούν διεξάγουν υπολογισμούς στα δεδομένα των χρηστών. Με αυτό τον τρόπο, οι πόροι του υποστρώματος, οι μηχανισμοί και οι πολιτικές του δρομολογητή είναι γνωστές και προγραμματιζόμενες από τους τελικούς χρήστες, με αποτέλεσμα οι υπηρεσίες να αναπτύσσονται και να παρέχονται άμεσα από το δίκτυο (λόγω της εύκολης προσαρμογής του) χωρίς την καθυστέρηση χρονοβόρων διαδικασιών προτυποποίησης.

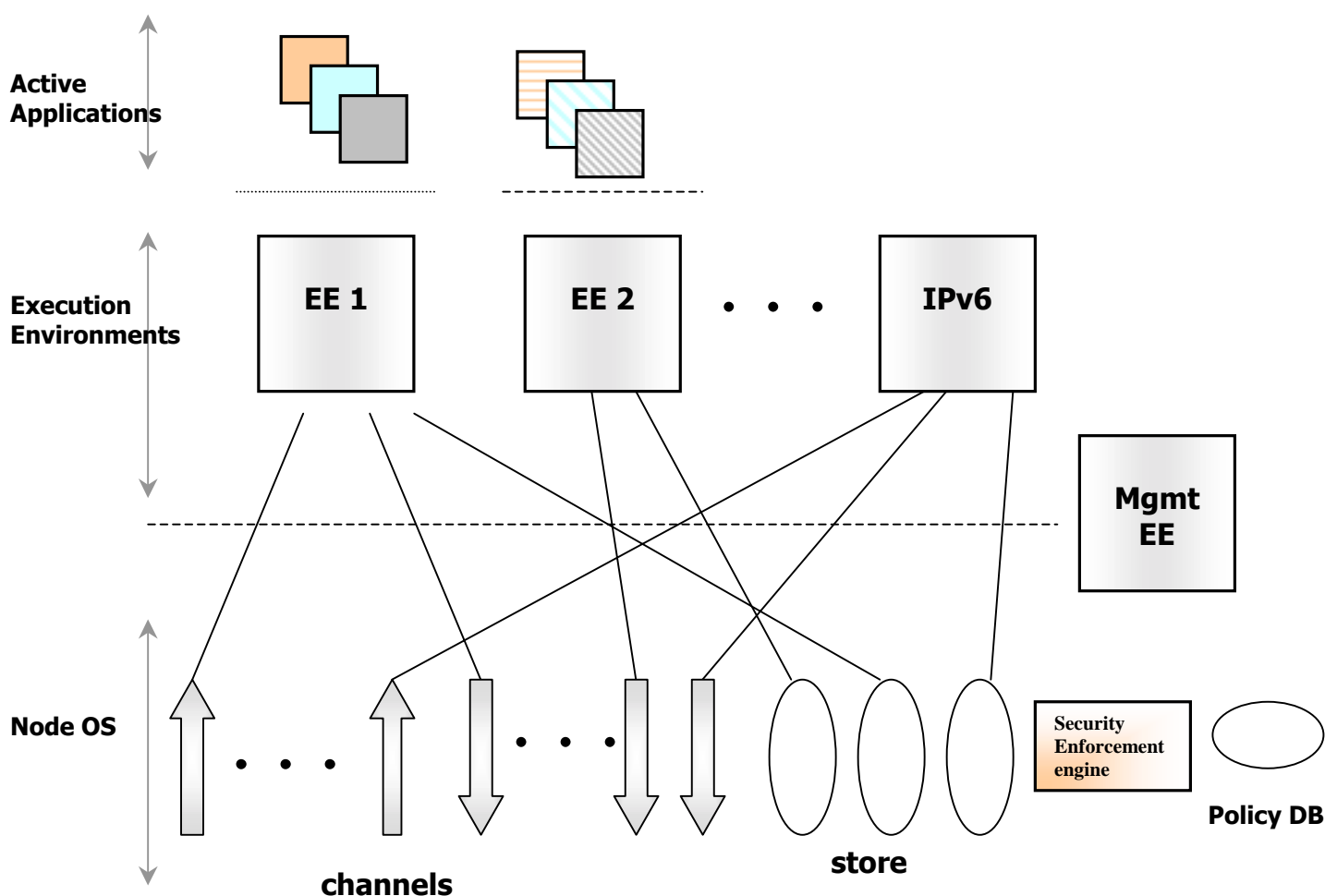


Σχήμα 1: Δρομολόγηση IP πακέτων και δρομολόγηση Ενεργών πακέτων. Πηγή Darpa

3. Η αρχιτεκτονική των ενεργών δικτύων

Το μοντέλο που περιγράφεται παρακάτω είναι αρκετά γενικό, χωρίς να εμβαθύνει στα χαρακτηριστικά του κάθε επιπέδου, και περιγράφεται στο κείμενο «Architectural Framework for Active Networks» που συντάχθηκε από τον K.L.Calvert στα πλαίσια της ομάδας εργασίας για τα Ενεργά δίκτυα [8].

Σύμφωνα με αυτό, ένα ενεργό δίκτυο αποτελείται από ένα σύνολο κόμβων (nodes), που δεν είναι αναγκαστικά ενεργοί, ενώ είναι συνδεδεμένοι μεταξύ τους με διάφορες δικτυακές τεχνολογίες. Κάθε ενεργός κόμβος (active node) τρέχει ένα λειτουργικό σύστημα κόμβου (Node Operating System, NodeOS) και ένα ή πολλά Περιβάλλοντα Εκτέλεσης (Execution Environments). Το NodeOS είναι υπεύθυνο για την εκχώρηση και τον χρονικό προγραμματισμό των πόρων του κόμβου (εύρος ζώνης διασύνδεσης, κύκλοι CPU, αποθηκευτικός χώρος). Κάθε περιβάλλον εκτέλεσης (EE) υλοποιεί μια εικονική μηχανή που μεταφράζει τα ενεργά πακέτα που φτάνουν στον κόμβο. Διαφορετικά EEs ορίζουν διαφορετικές εικονικές μηχανές. Οι χρήστες λαμβάνουν υπηρεσίες από το ενεργό δίκτυο μέσω των Ενεργών Εφαρμογών (Active Applications, AAs). Ο ρόλος των AAs είναι να προγραμματίσουν την εικονική μηχανή που παρέχεται από το EE, έτσι ώστε να διαθέσουν την υπηρεσία από άκρο σε άκρο (end to end service).



Σχήμα 2: Το γενικό αρχιτεκτονικό μοντέλο των ενεργών δικτύων

Η αρχιτεκτονική των ενεργών δικτύων βασίζεται στην αρχιτεκτονική των κόμβων, η οποία ασχολείται με τον τρόπο που επεξεργάζονται τα πακέτα αλλά και τον τρόπο με τον οποίο διαχειρίζονται οι τοπικοί πόροι. Αντίθετα με την αρχιτεκτονική των δικτύων που είναι άμεσα εξαρτημένη από το προγραμματιστικό περιβάλλον, η αρχιτεκτονική των κόμβων που παρουσιάζεται παρακάτω είναι σχεδιασμένη έτσι ώστε να επιτρέπει την συνύπαρξη πολλών προγραμματιστικών περιβαλλόντων δικτύου (Network API) και αρχιτεκτονικές δικτύων.

Όπως φαίνεται και από το σχήμα 2 κάθε κόμβος ενός ενεργού κόμβου αποτελείται από το λειτουργικό σύστημα του κόμβου, το περιβάλλον εκτέλεσης και τις ενεργές εφαρμογές. Το NodeOS διαχειρίζεται τους πόρους του ενεργού δικτύου όπως είναι μετάδοση, υπολογισμοί, αποθήκευση και μεσολαβεί κατά τη ζήτηση τους. Με τον τρόπο αυτό το NodeOS απομονώνει τα περιβάλλοντα εκτέλεσης (EEs) από τις λεπτομέρειες διαχείρισης των πόρων καθώς επίσης και από την συμπεριφορά άλλων EEs. Κατ' επέκταση τα EEs κρύβουν από το NodeOS τις λεπτομέρειες διάδρασης με τον τελικό χρήστη. Όταν ένα EEs ζητάει μια υπηρεσία από το NodeOS, η αίτηση συνοδεύεται από ένα πιστοποιητικό για την αρχή από την οποία γίνεται η αίτηση. Η αρχή αυτή μπορεί να είναι το ίδιο το EEs, η κάποιο άλλο μέρος για λογαριασμό του οποίου το EE εκτελεί μια ενεργή εφαρμογή (AA). Το NodeOS παρουσιάζει τις πληροφορίες αυτές σε μια μηχανή εφαρμογής (enforcement engine) η οποία επιβεβαιώνει την αυθεντικότητα των πληροφοριών και ελέγχει ότι η βάση δεδομένων ασφαλείας του κόμβου (security policy DB) εγκρίνει την παροχή της υπηρεσίας. Κάθε κόμβος έχει ξεχωριστό περιβάλλον εκτέλεσης διαχείρισης, μέσω του οποίου ελέγχονται συγκεκριμένα θέματα πολιτικής και διαμόρφωσης του κόμβου (Διατήρηση τη βάσης δεδομένων ασφαλείας, εγκατάσταση ή τροποποίηση νέων EEs κτλ.)

Ένα περιβάλλον εκτέλεσης (EE) ορίζει μια εικονική μηχανή (virtual machine) και ένα προγραμματιστικό περιβάλλον (programming interface) το οποίο μπορεί να ελεγχθεί μέσω των κατάλληλων κωδικοποιημένων οδηγιών. Το NodeOS παρέχει ένα συγκεκριμένο σύνολο δυνατοτήτων που μπορεί να χρησιμοποιήσει το EE για να υλοποιήσει την εικονική μηχανή. Η ανάπτυξη ενός EE δεν μπορεί να θεωρηθεί τριτομμένη εργασία, και αναμένεται ότι σε οποιαδήποτε στιγμή τα EEs θα είναι λίγα στον αριθμό. Γενικά αναμένεται η ανάπτυξη και η εγκατάσταση των EEs να γίνεται από τους διαχειριστές του κόμβου.

Μια Ενεργή εφαρμογή (AA) είναι ένα πρόγραμμα που εκτελείται από την εικονική μηχανή του EE και υλοποιεί μια υπηρεσία από άκρο σε άκρο, δηλαδή προσαρμοσμένες υπηρεσίες για εφαρμογές τελικών χρηστών χρησιμοποιώντας το προγραμματιστικό περιβάλλον που παρέχεται από το EE. Οι λεπτομέρειες σχετικά με τον τρόπο που ο κώδικας, που συνθέτει την AA, φορτώνεται στον κόμβο του δικτύου καθορίζονται από το EE. Ο κώδικας μπορεί να μεταφερθεί μαζί με το πακέτο, ή αυτοτελώς. Η ανάπτυξη των ενεργών εφαρμογών προβλέπεται ότι θα γίνεται ειδικούς. Το πιθανότερο είναι οι εφαρμογές που τρέχουν στα συστήματα των τελικών χρηστών να βάζουν σε λειτουργία Ενεργές εφαρμογές και έτσι να έχουν πρόσβαση στις υπηρεσίες των ενεργών δικτύων.

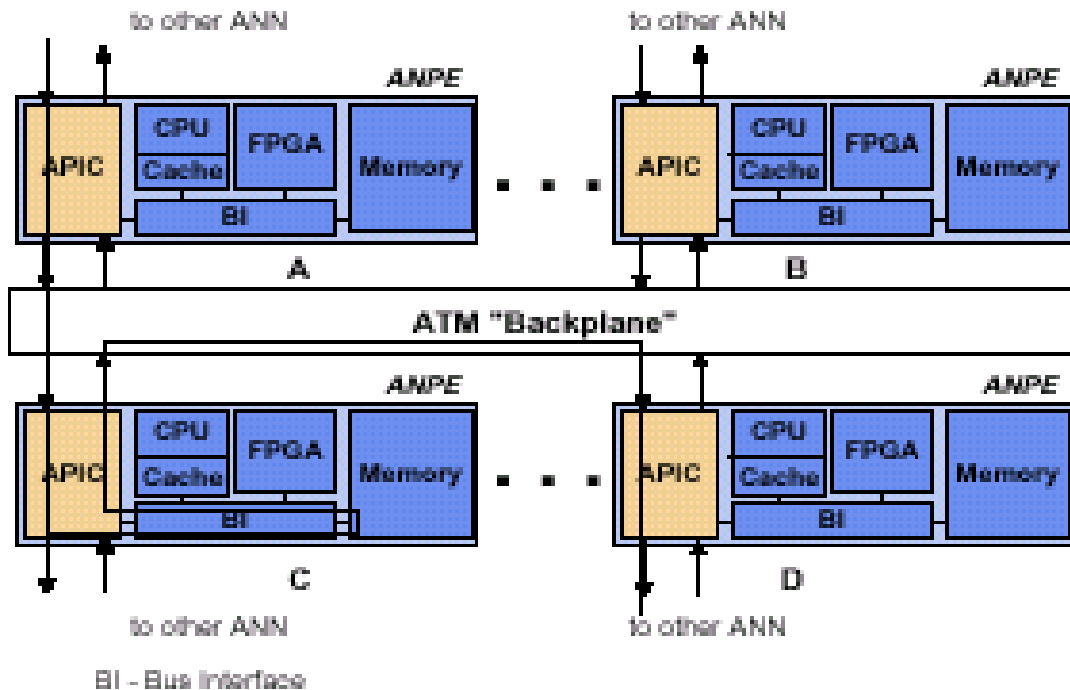
4. Ο ενεργός κόμβος σε επίπεδο υλικού (Hardware)

Για την υλοποίηση ενός δρομολογητή ενεργού δικτύου (που αποτελεί την βάση του κόμβου) η ομάδα εργασίας του Washington University θεωρεί ότι απαιτούνται τα παρακάτω:

- Ένας μεγάλος αριθμός επεξεργαστικών στοιχείων συγκρινόμενα με τις θύρες του δρομολογητή. Σε κάθε θύρα του δρομολογητή χρησιμοποιείται ένας επεξεργαστής γενικής χρήσης (CPU) και ένα FPGA (Field Programmable Gate Array). Ο συνδυασμός της CPU και του FPGA ονομάζεται επεξεργαστική μηχανή (Processing engine). Η CPU φροντίζει για την πλειονότητα των ενεργών λειτουργιών σε ένα πακέτο, ενώ το FPGA υλοποιεί λειτουργίες οι οποίες είναι ιδιαίτερα κρίσιμες για την απόδοση του υλικού. Τόσο η CPU όσο και το FPGA μπορούν να προγραμματιστούν "on the fly".
- Ισχυρή σύζευξη μεταξύ την επεξεργαστικής μηχανής και του δικτύου, καθώς και μεταξύ της επεξεργαστικής μηχανής και του switch backplane. Με δεδομένο ότι οι κύριοι περιοριστικοί παράγοντες είναι η επεξεργαστική ισχύ και το εύρος της μνήμης, πρέπει να καταστεί βέβαιο ότι αυτοί οι πολύτιμοι πόροι χρησιμοποιούνται με τον αποτελεσματικό τρόπο. Για να γίνει αυτό εφικτό τα μη ενεργά πακέτα δρομολογούνται απευθείας μέσω του switch backplane χωρίς την μεσολάβηση της CPU. Επίσης με την σύζευξη της CPU με το σύνδεσμο τα πακέτα φτάνουν στον κόμβο με ελάχιστο overhead.
- Κλιμακωτή επεξεργαστική δύναμη για να καλύψει τις ανάγκες ενεργής επεξεργασίας των πακέτων. Οι υπολογισμοί στις ροές των ενεργών πακέτων πρέπει να κατανεμηθούν εξ' ίσου στις επεξεργαστικές μηχανές του κόμβου.

Η αρχιτεκτονική του υλικού που χρησιμοποιείται βασίζεται στην αρχιτεκτονική IP δρομολόγησης υψηλής απόδοσης, η οποία έχει προσαρμοστεί για να ικανοποιεί τους σκοπούς των ενεργών δικτύων. Ο κόμβος αποτελείται από ένα σύνολο Επεξεργαστικών Στοιχείων Ενεργού Δικτύου (Active Network Processing Element, ANPE) συνδεδεμένα σε ένα σκελετό μεταγωγής ATM. Ο υλοποιημένος, βαθμιδωτός σκελετός μεταγωγής υποστηρίζει 8 θύρες με ρυθμό δεδομένων έως και 2.4Gbps σε κάθε θύρα. Κάθε ANPE αποτελείται από έναν επεξεργαστή γενικής χρήσης, ένα μεγάλο FPGA (100.000 θύρες) και από μνήμες. Τα ANPE συνδέονται στον σκελετό μεταγωγής μέσω των Ελεγκτών διασύνδεσης θύρας ATM (ATM Port Interconnect Controller, APIC).

Η κλιμάκωση επιτυγχάνεται μέσω της δυνατότητας τροποποίησης οποιουδήποτε αριθμού ANPEs, της βαθμιδωσης του σκελετού μεταγωγής και ενός αλγορίθμου κατανομής του φορτίου με τον οποίο οι ροές των ενεργών φορτίων κατανέμονται δυναμικά σε λιγότερο φορτωμένα ANPEs. Στο παρακάτω σχήμα φαίνεται ένα παράδειγμα ροής ενεργών πακέτων (σχήμα 3) όπου μια ροή ενεργών πακέτων εισέρχεται στον κόμβο μέσω του ANPE A, και εξέρχεται από το ANPE D. Λόγω φόρτου στο ANPE A ο αλγόριθμος κατανομής



Σχήμα 3: Υλικό Ενεργού Κόμβου (Active Network Node)

5. Το λειτουργικό σύστημα του κόμβου (NodeOS)

Η ανάγκη για την ύπαρξη ενός κοινού λειτουργικού συστήματος σαφώς διαχωρισμένου από τα περιβάλλοντα εκτέλεσης (EEs) προέρχεται από την ανάγκη υποστήριξης πολλών προγραμματιστικών γλωσσών από ένα κόμβο καθώς επίσης και για την εισαγωγή μιας προγραμματιστικής γλώσσας σε πολλούς κόμβους. Ο διαχωρισμός του λειτουργικού συστήματος του κόμβου από τα περιβάλλοντα εκτέλεσης θέτει τον παρακάτω προβληματισμό: που ακριβώς θα σχεδιαστούν τα όρια μεταξύ του NodeOS και του EE. Γενικά, το NodeOS είναι υπεύθυνο για την πολύπλεξη των πόρων του κόμβου μεταξύ των διαφόρων ροών πακέτων, ενώ ο ρόλος των EEs είναι να παρέχουν ένα επαρκές προγραμματιστικό περιβάλλον για τις ενεργές εφαρμογές (AAs). Ειδικότερα το NodeOS έχει τους παρακάτω σχεδιαστικούς στόχους:

- Ο κύριος στόχος του λειτουργικού είναι να υποστηρίζει την προώθηση των πακέτων. Όλος ο σχεδιασμός του λειτουργικού γίνεται γύρω από την ιδέα των ροών πακέτων δικτύου. Η επεξεργασία των πακέτων, οι υπολογισμοί για την χρήση των πόρων και ο έλεγχος εισόδου γίνονται ανά ροή (per – flow basis). Επίσης το λειτουργικό δεν μπορεί να προδιαγράψει μόνο έναν ορισμό μιας ροής.
- Οι διάφορες υλοποιήσεις του NodeOS δεν θα εξάγουν το ίδιο ακριβώς σύνολο χαρακτηριστικών, αλλά θα περιλαμβάνουν κάποιες ειδικές δυνατότητες τις οποίες θα εκμεταλλεύονται τα EEs και κατ' επέκταση οι

AAs. Επίσης θα πρέπει να προωθεί πακέτα διαφόρων ειδών (όπως μη ενεργά IP πακέτα) σε πολύ μεγάλες ταχύτητες.

- Κάθε φορά που το NodeOS θα απαιτείται κάποιο μηχανισμό που δεν είναι μοναδικός στα ενεργά δίκτυα, θα τον δανείζεται από καθιερωμένα λειτουργικά συστήματα.

Οι βασικές αφαιρέσεις που κάνει το λειτουργικό σύστημα είναι πέντε και σχετίζονται με τους διάφορους πόρους του συστήματος:

<i>Abstraction</i>	<i>Resource encapsulation</i>
Thread pools	Computation
Memory pools	Memory
Channels	Communication
Files	Persistent storage
Domains	Control and scheduling of resources

Πίνακας 1: Οι βασικές αφαιρέσεις του NodeOS.

Κάθε domain σχετίζεται με μια memory pool που περιλαμβάνει σελίδες φυσικής μνήμης. Κάθε domain περιέχει thread pools από τα οποία χρησιμοποιούνται αντικείμενα thread για να χειριστούν τα πακέτα που εισέρχονται από τα input channels, βασιζόμενα στο κλειδί αποπολύπλεξης και να τα στείλουν στα output channels. Τέλος ένα cut-through channel επιτρέπει στο NodeOS να συνδέει απευθείας το input channel και το output channel.

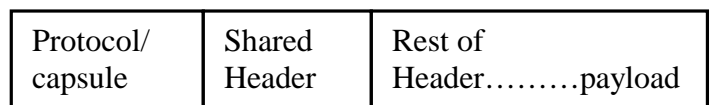
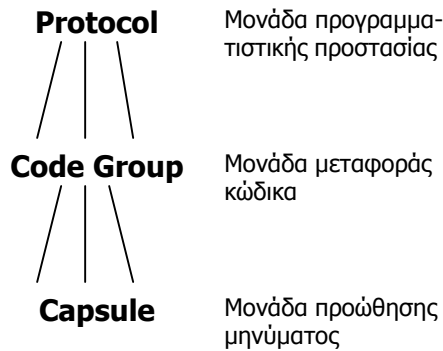
6. Τα περιβάλλοντα εκτέλεσης (Execution Environments, EEs)

Ο ρόλος των περιβαλλόντων εκτέλεσης επικεντρώνεται κυρίως στην δημιουργία ενός προγραμματικού περιβάλλοντος για την συγγραφή κώδικα Ενεργών εφαρμογών. Μέχρι σήμερα έχουν αναπτυχθεί αρκετά EEs τα σημαντικότερα των οποίων είναι το ANTS, το PLAN και το DAN τα οποία αναπτύσσονται παρακάτω.

6.1 Active Network Transport System (ANTS)

Το ANTS είναι ένα μοντέλο που αναπτύχθηκε από το MIT, το οποίο είναι ίσως το πιο αντιπροσωπευτικό EE για τα ενεργά δίκτυα. Ο ενεργός δρομολογητής εμπλουτίζεται με ένα JAVA μεταφραστή, ο οποίος μπορεί να εκτελεί και να εγκαθιστά κώδικα 'on the fly' από οπουδήποτε στο δίκτυο. Όλα τα πακέτα αντικαθιστούνται από θύλακες (capsules), οι οποίοι ομαδοποιούνται

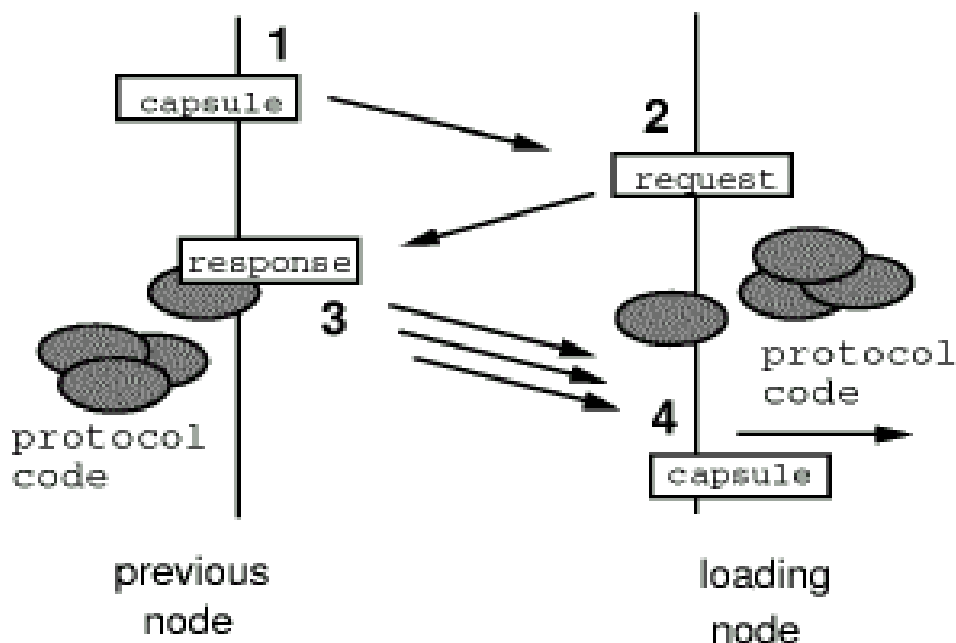
σε ομάδες και πρωτόκολλα. Εκτός από την κεφαλίδα και το payload, κάθε θύλακας φέρει ένα πεδίο (protocol identifier), το οποίο είναι το συνοπτικό μήνυμα του κώδικα του πρωτοκόλλου. Έτσι εξασφαλίζεται η εκχώρηση των πρωτοκόλλων με αποκεντρωμένο τρόπο και η ελαχιστοποίηση της πιθανότητας παρεμβολής άλλων πρωτοκόλλων.



Σχήμα 4: Ιεραρχία σύνθεσης θύλακα

Σχήμα 5: Η μορφή του θύλακα

Όταν ένας θύλακας φτάνει σε ένα κόμβο, ο κώδικας που φέρει εκτελείται άμεσα σε ένα προστατευόμενο περιβάλλον. Για τον έλεγχο των συνολικών πόρων που χρησιμοποιούνται από τον θύλακα χρησιμοποιείται ένα γενικοποιημένο TTL (time to live) σχήμα. Ο κώδικας του θύλακα καλεί το API (Application programming interface) που είναι διαθέσιμο από τον ενεργό κόμβο. Το API περιλαμβάνει μεταξύ άλλων πρόσβαση στο περιβάλλον του κόμβου και χειρισμό του πακέτου. Αν ο κώδικας αναφέρεται σε ένα πρωτόκολλο που δεν υπάρχει στον κόμβο, τότε αυτό φορτώνεται δυναμικά, χρησιμοποιώντας ένα ευέλικτο πρωτόκολλο κατανομής.



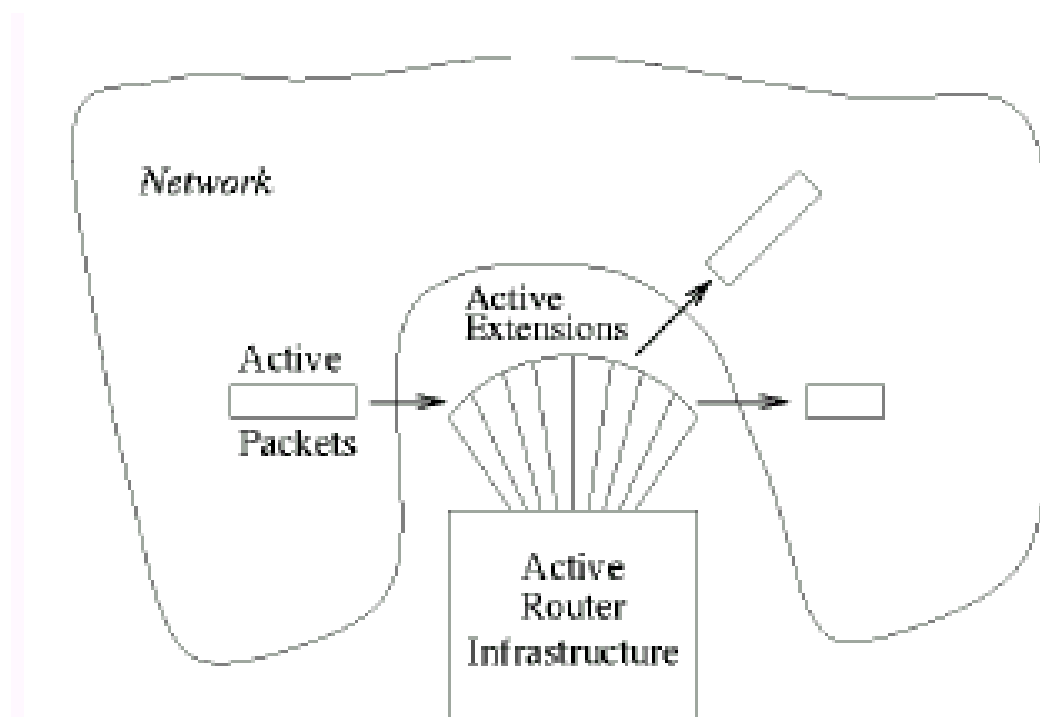
Σχήμα 6: Κατ' αίτηση φόρτωση ομάδων κώδικα (group code) Πηγή [15]

Η φόρτωση κατ' αίτηση έχει το μειονέκτημα ότι αυξάνει το χρόνο αναμονής (latency). Παρόλα αυτά, δεν εφαρμόστηκαν οι λύσεις της μεταφοράς του όλου του κώδικα από τον θύλακα ή της εκ' των προτέρων φόρτωσης του κώδικα στον κόμβο, γιατί το δίκτυο δεν θα ήταν εύκολα προσαρμόσιμο σε πιθανές αποτυχίες, ούτε κατάλληλο για πρωτόκολλα μικρής διάρκειας ζωής, ο σκοπός των οποίων δεν είναι από την αρχή γνωστός.

6.2 Programming Language for Active Networks (PLAN)

Το περιβάλλον εκτέλεσης PLAN αναπτύχθηκε από το Pennsylvania University και αποτελεί μέρος της ευρύτερης αρχιτεκτονικής με τίτλο "The SwitchWare Active Network Architecture" [16] (πολλές φορές αναφέρεται και ως αρχιτεκτονική προγραμματιζόμενου μεταγωγέα, Programmable Switch). Η δομή της αρχιτεκτονικής αυτής αποτελείται από τρία επίπεδα:

- Τα ενεργά πακέτα (active packets), που αντικαθιστούν τα παραδοσιακά πακέτα και περιέχει κινητό κώδικα
- Τις ενεργές επεκτάσεις (active extensions), οι οποίες παρέχουν υπηρεσίες στα στοιχεία του δικτύου, και οι οποίες μπορούν να φορτωθούν δυναμικά
- Τους ενεργούς ασφαλείς δρομολογητές (secure active routers), οι οποίοι σχηματίζουν μια βάση υψηλής ακεραιότητας, στην οποία βασίζεται η ασφάλεια των άλλων επιπέδων.



Σχήμα 7: Η λειτουργία του μοντέλου Programmable Switch. Πηγή [16]

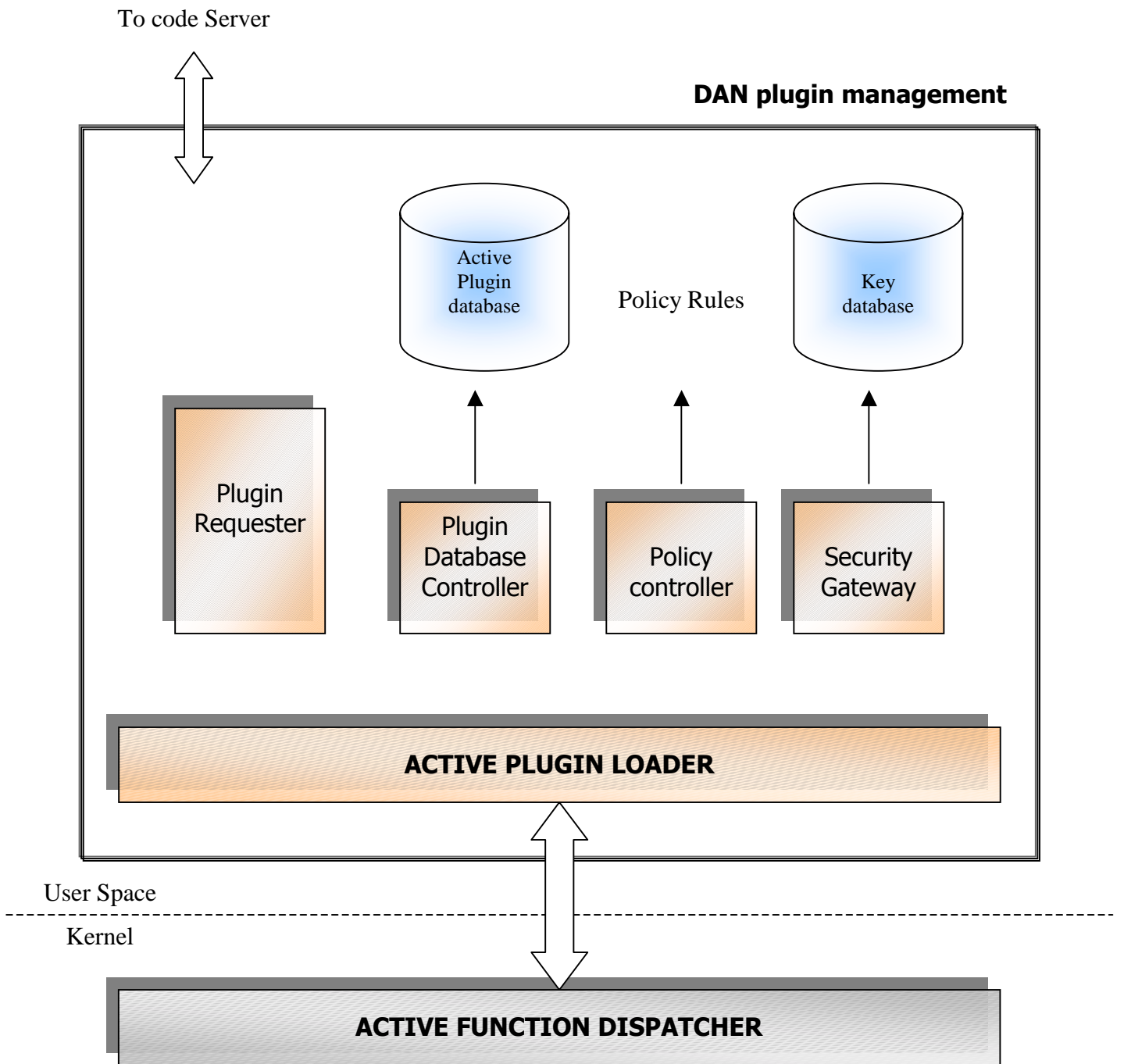
Στο σχήμα 7 φαίνεται συνοπτικά η λειτουργία του μοντέλου. Η εισαγωγή νέων λειτουργιών στον ενεργό κόμβο ελέγχεται από τις τοπική αρχή. Οι επεκτάσεις (που λέγονται και switchlets) είναι κατά βάση βιβλιοθήκες. Οι χρήστες του δικτύου έχουν την δυνατότητα να αξιολογήσουν τα PLAN προγράμματα στο περιβάλλον που τους παρέχει ο κόμβος. Η γλώσσα έχει τα παρακάτω χαρακτηριστικά:

- Εγγυημένος τερματισμός
- Έλλειψη συναρτήσεων μεγάλων τάξεων
- Μη αναδρομικότητα, έλλειψη δομών επανάληψης
- Έλλειψη λειτουργιών αναφοράς προώθησης
- Δυναμική ανάκληση βιβλιοθηκών που εγκαθίστανται από τα switchlets

Η δομή του ενεργού δρομολογητή, εξασφαλίζει το ασφαλές περιβάλλον για την εκτέλεση των switchlets και των PLAN προγραμμάτων. Βασικές υπηρεσίες όπως η δρομολόγηση, η απόδοση διευθύνσεων, η προώθηση ακόμα και η μετάφραση των PLAN προγραμμάτων υλοποιούνται ως switchlets. Για τον κώδικα των ενεργών πακέτων χρησιμοποιείται η γλώσσα Caml.

6.3 Distributed Code Caching for Active Networks (DAN)

Το περιβάλλον εκτέλεσης DAN αναπτύχθηκε από το Washington University και περιγράφεται αναλυτικά στα πρακτικά της INFOCOM '98, με τίτλο "DAN: Distributed Code Caching for Active Networks". Το DAN αποτελεί μια ενδιάμεση λύση ανάμεσα στα δύο προηγούμενα EEs καθώς συνδυάζει την φιλοσοφία του προγραμματιζόμενου μεταγωγέα και των πακέτων-θυλάκων. Ο συνδυασμός αυτός είναι αναγκαίος κατά πολλούς, για την αντιμετώπιση προβλημάτων απόδοσης και ασφάλειας σε δίκτυα με ταχύτητες gigabit. Πιο συγκεκριμένα, ο κώδικας του θύλακα αντικαθίσταται από μια αναφορά σε ένα ενεργό plugin το οποίο είναι αποθηκευμένο στον code server. Κάθε φορά που η αναφορά είναι άγνωστη στον δρομολογητή τότε αυτή «κατεβαίνει» από τον code server. Ο κώδικας εκτελείται ταχύτατα, λόγω της δυναμικής του σύνδεσης με τον κόμβο. Τα προβλήματα ασφάλειας αντιμετωπίζονται με την χρήση γνωστών κρυπτογραφικών τεχνικών και για το λόγο αυτό δεν απαιτούνται αργές εικονικές μηχανές θέτοντας βέβαια έτσι κάποιους περιορισμούς όσον αφορά την πηγή του κώδικα του κόμβου και την αρχή διαχείρισης του κόμβου. Το περιβάλλον εκτέλεσης DAN φαίνεται στο σχήμα 8.



Σχήμα 8: Το περιβάλλον εκτέλεσης DAN. Πηγή [12]

7. Ενεργές εφαρμογές (Active Applications)

Οι τεχνολογίες κινητού κώδικα, όπως είναι η JAVA, γνώρισαν σημαντική εξάπλωση λόγω των εφαρμογών που σχετίζονται με το δίκτυο, τους agent και τα applets, που χρησιμοποιούνται σε μεταφρασμένη μορφή στις διάφορες πλατφόρμες. Παρ' όλα αυτά, οι applets και οι agents έχουν σχεδιαστεί για την ανάπτυξη λειτουργιών στα άκρα του δικτύου, ενώ συγκεκριμένα προβλήματα, όπως ο έλεγχος συμφόρησης, χρειάζονται meta-πληροφορία η οποία είναι διαθέσιμη μόνο μέσα στο δίκτυο. Οι εφαρμογές που τρέχουν προσαρμοσμένες λειτουργίες μέσα στο δίκτυο μπορούν να χειριστούν εξαρτήσεις μεταξύ των μονάδων δεδομένων και να διαχωρίσουν τις σημαντικές μονάδες. Η αποστολή μιας MPEG ροής μέσω ενός δικτύου παθητικών πακέτων, δεν δίνει κανένα έλεγχο σε σχέση με την απόρριψη των πακέτων σε μια πιθανή συμφόρηση. Αντίθετα μια μέθοδος transcoding, θα απέρριπτε τα λιγότερο σημαντικά πλαίσια (frames), η ακόμα και πακέτα που δεν χρησιμοποιούνται πλέον αφού κάποιο πακέτο έχει χαθεί νωρίτερα.

Υπάρχουν εφαρμογές για τις οποίες η απόδοση τους αξιολογείται σε ειδικές μονάδες αντί για τις συνηθισμένες μεθόδους μετρήσεις των δικτύων όπως το εύρος ζώνης ή η καθυστέρηση. Τέτοιες εφαρμογές είναι οι δημοπρασίες μέσω Internet καθώς και η παρακολούθηση του χρηματιστηρίου.

Στον παρακάτω πίνακα ορισμένες εφαρμογές οι οποίες θα βασιστούν στο μέλλον στην αρχιτεκτονική των ενεργών δικτύων καθώς και ο τρόπος υλοποίησης τους στην σημερινή δομή του Internet:

Εφαρμογή	Μη ενεργή λύση
Firewall	Εγκατάσταση φίλτρων ελέγχου στους δρομολογητές που βρίσκονται στα άκρα
Web proxy	Squid, Harvest, χειρονακτική ανάπτυξη
Scalable auction Server	Χρειάζεται ειδικό κώδικα για την εγκατάσταση στους δρομολογητές γύρω από τον server
Routing around congestion	Υπηρεσίες διάκρισης, κρατήσεων
Multicast, Reliable Multicast	MBONE
Transcoding	Media gateways
Congestion control	Πολιτικές από άκρο σε άκρο (TCP)
Protocol boosters	Επιταχυντές που εγκαθίστανται χειρονακτικά
WAN cooperative caches	Squid, Harvest
Custom stock tickers	Από άκρο σε άκρο, χρησιμοποιώντας agents σε κάθε άκρο
Overlay networks	Xbone

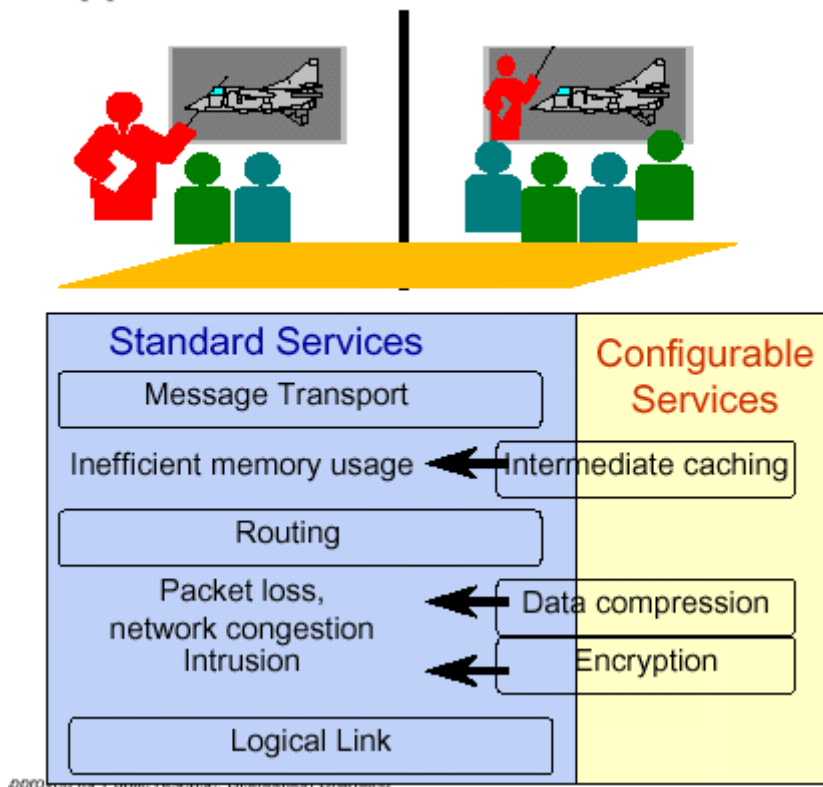
Εφαρμογή	Μη ενεργή λύση
Quality of service	RSVP
Mobile routing	Mobile IP, snoop TCP
mSMTP, mHTTP, CTCP	-
Ad hoc multicast	Unicast links
Multicast, Reliable Multicast	MBONE
Transcoding	Media gateways
Congestion control	Πολιτικές από άκρο σε άκρο (TCP)
Conjunctive/ disconjunctive data streams	Επεξεργασία μέσα στους servers, ή στους ακραίους hosts.

Πίνακας 2: Ένεργές εφαρμογές και μη ενεργές λύσεις.

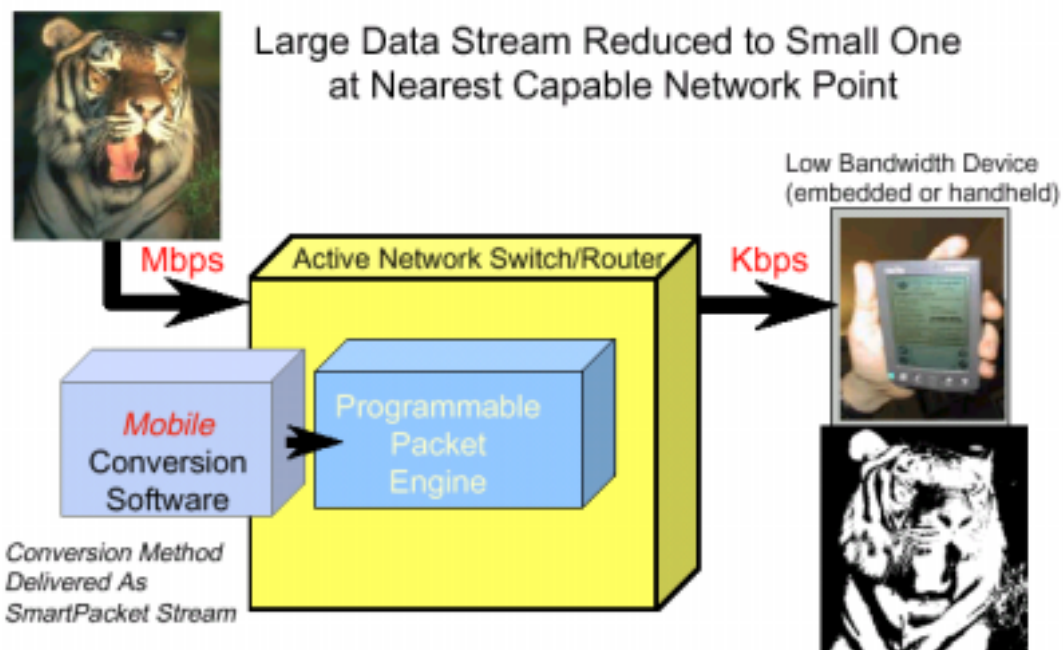
Από των παραπάνω πίνακα γίνεται φανερή η ανάγκη για προσαρμογή των κόμβων στο εσωτερικό του δικτύου έτσι ώστε να μπορούν φιλοξενήσουν νέες εφαρμογές που υλοποιούνται άμεσα , τροποποιώντας το λογισμικό του δρομολογητή, όπως είναι οι Firewalls, ενώ όταν η πρόσβαση στον κόμβο δεν είναι δυνατή η δρομολόγηση να γίνεται στο επίπεδο Εφαρμογής, όπως είναι οι Web-proxies, το transcoding, η το overlay network. Μερικά από τα παραπάνω προβλήματα δεν μπορούν να λυθούν από άκρο σε άκρο, με πιο χαρακτηριστικά τα παραδείγματα της πολυεκπομπής (multicasting), της κινητικότητας (mobility), και της εγγύησης της παρεχόμενης ποιότητας των υπηρεσιών (QoS).

Κατά τα επόμενα χρόνια το Internet αναμένεται να αντιμετωπίσει σημαντική αύξηση στον όγκο της διακινούμενης πληροφορίας, καθώς επίσης και την κυριαρχία των εφαρμογών πολυμέσων. Η φιλοσοφία των ενεργών δικτύων, κυρίως λόγω των επεξεργαστικών δυνατοτήτων των δρομολογητών, θα βοηθήσει σημαντικά στην διακίνηση ροών μακράς διάρκειας, που χρησιμοποιούνται από τις δημοφιλέστερες εφαρμογές σήμερα όπως είναι η τηλεδιάσκεψη. Η στροφή των ενδιάμεσων κόμβων κοντά στα προβλήματα που παρουσιάζονται κατά τη διάρκεια μιας τηλεδιάσκεψης έχει ως αποτέλεσμα οι χρήστες να αντιλαμβάνονται ποιοτική βελτίωση της υπηρεσίας (καθαρός ήχος, ομαλό video).

Applications Use Network Services



Σχήμα 9: Η βελτίωση των υπηρεσιών τηλεδιάσκεψης στα ενεργά δίκτυα



Σχήμα 10: Ανταλλαγή δεδομένων σε δίκτυα με διαφορετικό εύρος ζώνης

8. Active Network Encapsulation Protocol (ANEP)

Το πρωτόκολλο ANEP παρέχει τη δυνατότητα στους χρήστες να ελέγχουν την δυνατότητα δρομολόγησης των πακέτων τους σε ένα συγκεκριμένο περιβάλλον εκτέλεσης. Η επικεφαλίδα του ANEP περιέχει το πεδίο "Type Identifier" σύμφωνα με το οποίο γίνεται η ανάθεση του περιβάλλοντος εκτέλεσης (Active Networks Assigned Numbers Authority , ANANA). Αν το συγκεκριμένο ΕΕ είναι φορτωμένο στον κόμβο, τότε τα πακέτα που αναφέρονται μέσω της ANEP κεφαλίδας σε αυτό θα δρομολογηθούν στα κανάλια που είναι συνδεδεμένα στο υποδεικνυόμενο ΕΕ. Η δημιουργία των καναλιών γίνεται κατά την εκκίνηση του ΕΕ. Δεν είναι υποχρεωτική η ύπαρξη ANEP κεφαλίδας, για την επεξεργασία του πακέτου από το ΕΕ. Ακόμη και σε περιπτώσεις μη ενεργών πακέτων η προώθηση γίνεται με τη δημιουργία ψευδο-ΕΕ (π.χ. IPv4 προώθηση).

Εκτός από τα προαναφερθέντα το πρωτόκολλο παρέχει οδηγίες για των χειρισμό των λαθών. Όταν ένα πακέτο αποτύχει να φτάσει στο επιθυμητό ΕΕ, τότε το ANEP επιτρέπει στο χρήστη να καθοδηγήσει το λειτουργικό σύστημα του κόμβου για τον χειρισμό του λάθους. Τα μηνύματα που μπορούν να σταλούν είναι της μορφής, ρίξε το πακέτο, προσπάθησε να το προωθήσεις, στείλε ένα μήνυμα λάθους, και δηλώνονται στο πεδίο Flags.

Τέλος η κεφαλίδα του ANEP μπορεί να χρησιμοποιηθεί ως πιστοποιητικό για το πακέτο έτσι ώστε να κινείται ανάμεσα στους κόμβους του δικτύου χωρίς να χρειάζεται κάθε φορά να γίνεται σε κάθε κόμβο πιστοποίηση.

Η μορφή της ANEP κεφαλίδας φαίνεται στο παρακάτω σχήμα:

Version	Flags	Type ID
ANEP Header Length		ANEP Header Length
Options		
Payload		

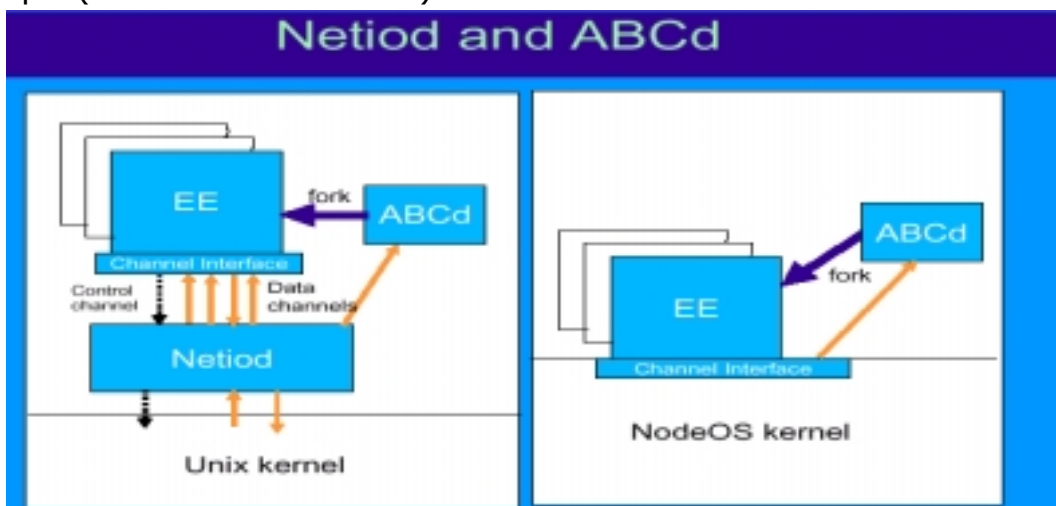
Σχήμα 11: Η μορφή του ενεργού πακέτου στο πρωτόκολλο ANEP

9. Active Network Backbone, ABONE

Το Active Network Backbone (ABONE) είναι ένα πείραμα που χρηματοδοτείται από τον DARPA και σκοπό έχει την διατήρηση ενός μέσου δοκιμής (testbed) έτσι ώστε να υποστηριχθεί η έρευνα γύρω από τα ενεργά δίκτυα. Το ABONE σχηματίζει μια εικονική δικτυακή υποδομή όπου πολλά κομμάτια έρευνας όπως είναι τα περιβάλλοντα εκτέλεσης, οι ενεργές εφαρμογές και τα λειτουργικά συστήματα των κόμβων, μπορούν να ελεγχθούν και να υλοποιηθούν σε πειραματικό στάδιο.

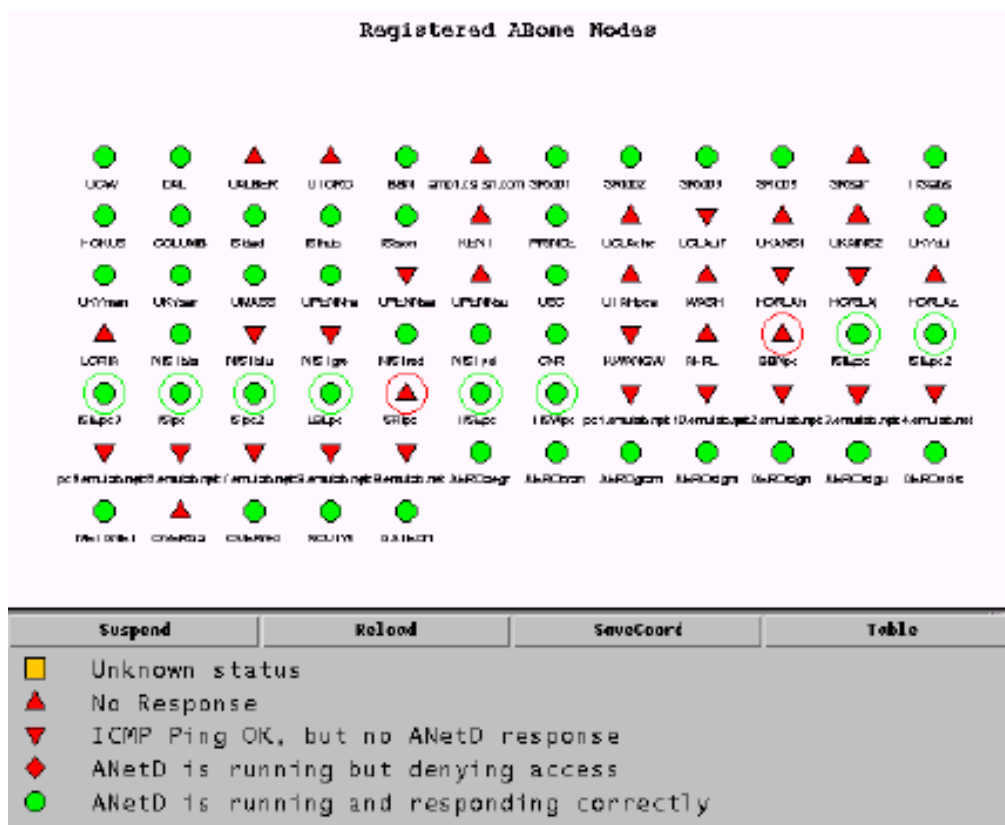
Η υποδομή του ABONE βασίζεται σε υπάρχοντες κόμβους και συνδέσμους. Οι κόμβοι αυτοί είναι κυρίως οι κόμβοι του project CRAIN που εκπονείται από τον DARPA καθώς και κόμβοι που παρέχονται από τους ερευνητές ενώ τελευταία χρησιμοποιείται και η δομή του Internet. Μεταξύ των διάφορων πλατφόρμων των κόμβων περιλαμβάνονται το Linux, το Solaris, το FreeBSD κ.α. Οι κόμβοι διακρίνονται σε πυρήνες (core nodes) και τελικούς (edge nodes). Οι κόμβοι – πυρήνες αποτελούν δημόσιο πόρο για τους ερευνητές που αναπτύσσουν EE και AA γιατί είναι πάντα διαθέσιμοι ενώ φέρουν και μόνιμα EEs με καθορισμένες τοπολογίες έτσι ώστε να είναι ευκολότερη η ανάπτυξη των ενεργών εφαρμογών. Οι τελικοί κόμβοι ανήκουν στους ερευνητές, και δεν χρειάζεται να πληρούν το κριτήριο της επ' αόριστον διαθεσιμότητας. Η είσοδος τους καθώς και η έξοδος τους από την τοπολογία των EEs είναι δυναμική, ενώ η ανάπτυξη των EEs και AAs γίνεται καταρχήν σε ένα τοπικό αντίγραφο.

Το πρωταρχικό λογισμικό διαχείρισης του ABONE ήταν το ANETD (Active Network daemon), ενώ αργότερα εμφανίστηκαν το λογισμικό NETIOD (Active Network I/O daemon) καθώς και το ABCd (Abone Control daemon). Τα παραπάνω λογισμικά ελέγχουν την εκτέλεση των EEs με κρυπτογράφηση δημοσίου κλειδιού, καθώς και τι είδους κώδικας μπορεί να εκτελεστεί κάθε φορά (Trusted code server list).



Σχήμα 12: Τα λογισμικά διαχείρισης Netiod και ABCd

Την ευθύνη επίβλεψης της δομής την φέρει το κέντρο συντονισμού του ABONE (Abone Coordination Center, ABOCC). Ο φορέας αυτός καταγράφει και ελέγχει τους κόμβους, τους ερευνητές ανάπτυξης EEs, διατηρεί τα κύρια ACL και TCL αρχεία, αποδίδει κωδικούς, και παρακολουθεί την κατάσταση του ABONE σε πραγματικό χρόνο.



Σχήμα 13: Η οθόνη δυναμικής παρακολούθησης του ABONE

Το ABONE αποτελεί την πρώτη ουσιαστικά υλοποίηση της αρχιτεκτονικής των ενεργών δικτύων. Σήμερα απαριθμεί 77 κόμβους 57 είναι συνεχώς διαθέσιμοι και μπορούν να προσαρμοστούν. Το κύριο προγραμματιστικό περιβάλλον που χρησιμοποιείται είναι το ANTS ενώ βασικοί προβληματισμοί για το μέλλον είναι θέματα ασφάλειας και λειτουργικότητας της δομής.

10. Συμπεράσματα - Επίλογος

Η αρχιτεκτονική των ενεργών δικτύων συνδυάζει θέματα έρευνας σχετικά με τα λειτουργικά συστήματα, τις γλώσσες προγραμματισμού και τις τεχνολογίες δικτύων. Εντάσσεται δε στην γενικότερη φιλοσοφία για εντατικότερα προγράμματα, όπως τα ενεργά λειτουργικά συστήματα, όπου χρήστες έχουν την δυνατότητα να προσαρμόσουν τις λειτουργίες και τις υπηρεσίες. Τα ενεργά δίκτυα τεχνολογικά στηρίζονται στην πρόοδο του κινητού κώδικα αλλά και στο γεγονός ότι η ταχύτητα των δικτύων δεν αυξάνεται με τον ρυθμό που αυξάνεται η επεξεργαστική ισχύς.

Η εισαγωγή νέων λειτουργιών στο δίκτυο έχει το βασικό μειονέκτημα ότι τις περισσότερες φορές πρέπει να αλλάξει το λογισμικό δρομολόγησης το οποίο συνήθως γίνεται χειρονακτικά και έχει αυξημένο κόστος. Ο κύριος στόχος των ενεργών δικτύων είναι να καταστήσει τους δρομολογητές προγραμματιζόμενους έτσι ώστε να είναι δυνατή η αποκεντροποιημένη κατασκευή και χρήση νέων πρωτοκόλλων. Η δυνατότητα ελέγχου των κόμβων καθώς και η χρήση των πόρων τους από τους χρήστες θέτει σοβαρά προβλήματα λειτουργικότητας, ασφάλειας και επεκτασιμότητας. Επίσης είναι σαφές ότι η ανάπτυξη των υπηρεσιών στα ενεργά δίκτυα θα συνεχίσει να γίνεται από ειδικούς και όχι από τους τελικούς χρήστες, μειώνοντας έτσι τις προσδοκίες για ταχύτερη ανάπτυξη των υπηρεσιών. Είναι όμως βέβαιο ότι ο χρόνος ανάπτυξης και εφαρμογής της υπηρεσίας στο δίκτυο θα μειωθεί σημαντικά.

Συμπερασματικά, θα μπορούσε να ειπωθεί ότι τα ενεργά δίκτυα αποτελούν μια ελπιδοφόρα αρχιτεκτονική που θα δώσει μεγάλη ώθηση σε εφαρμογές που μέχρι σήμερα ήταν δύσκολο να αναπτυχθούν λόγω των περιορισμένων δυνατοτήτων των δικτυακών υποδομών. Η υλοποίηση της δομής έστω και σε ερευνητικό επίπεδο δείχνει ότι δεν αποτελεί ουτοπία. Το μέλλον όμως θα δείξει κατά πόσο η ιδέα αυτή δεν θα έχει την άσχημη κατάληξη που είχαν τόσες άλλες λαμπρές ιδέες.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Darpa Information Technology office.
www.darpa.mil/ito/research/anets/ Ο δικτυακός τόπος του οργανισμού Darpa, ο οποίος έχει το συντονισμό της έρευνας για τα ενεργά δίκτυα. Περιέχει πολλά άρθρα και εργασίες καθώς και πολυάριθμες συνδέσεις σε άλλους τόπους.
- [2] Omid Technologies Inc. www.omid.ca/ActiveNets/
Δικτυακός τόπος канаδέζικης εταιρίας πληροφορικής όπου γίνεται συνοπτική παρουσίαση των ενεργών δικτύων και περιγράφεται ένα παράδειγμα εφαρμογής τους.
- [3] D.L. Tennenhouse, S.J. Garland, L. Shrira, M.F. Kaashoek, *From Internet to Active Networks*. 1996, MIT.
- [4] The Active Networks Project at MIT
<http://www.sds.lcs.mit.edu/activeware/> Η δικτυακή σελίδα του MIT για το ερευνητικό πρόγραμμα των ενεργών δικτύων. Περιλαμβάνει εργασίες, διατριβές καθώς και διάφορα links σε οργανισμούς που διεξάγουν έρευνα σχετική με τα ενεργά δίκτυα.
- [5] The Composable Active Network Elements (CANEs).
<http://www.cc.gatech.edu/projects/canes/> Έρευνα σχετικά με την αρχιτεκτονική και την εφαρμογή των ενεργών δικτύων από το Ινστιτούτο Τεχνολογίας της Georgia.
- [6] A. Tan, A Galis, *Active Network Node Developments*, 2000, University College London.
- [7] T. Ramteke, *Networks*, 2001, New Jersey.
- [8] K.L.Calvert, *Architectural Framework for Active Networks*, 1999, University of Kentucky.
- [9] The switchware project, <http://www.cis.upenn.edu/~switchware/>
Ιστοσελίδα του πανεπιστημίου της Πενσυλβάνια όπου υπάρχει πληθώρα πληροφοριών σχετικά με την αρχιτεκτονική Switchware και το προγραμματιστικό περιβάλλον PLAN.
- [10] Tiger NodeOS Project Overview
<http://anet.it.rit.edu/TigerOSOverview/>
Ερευνητικό πρόγραμμα σχετικά με την δομή ενός NodeOS από το RIT's Information Technology Lab.
- [11] ANN – A scalable, high performance Active Network Node,
<http://www.arl.wustl.edu/arl/projects/ann/ann.html> Η ιστοσελίδα ερευνητικού προγράμματος σχετικά με την ανάπτυξη ενός λειτουργικού ενεργού κόμβου από το Washington University.

- [12] D.Decasper, G. Parulkar, S. Choi, J. Hart, T. Wolf, B. Plattner, *A scalable high performance Active Network Node*, 1998, Washington University.
- [13] Janos A Java-oriented Active Network Operating System <http://www.cs.utah.edu/flux/talks/janos-pim-990416/> Υλοποίηση ενός λειτουργικού συστήματος για ενεργούς κόμβους βασισμένο στην JAVA.
- [14] Patrick Tullmann, Mike Hibler, Jay Lepreau, *Janos: A Java-oriented OS for Active Network Nodes*, 2001, University of Utah
- [15] D. Wetherall, J. Guttag, D. Tennenhouse, *ANTS: A toolkit for building and dynamically deploying network protocols*. 1998, IEEE OPENARCH'98, San Francisco.
- [16] D. Scott, W. Arbaugh, M. Hicks, et al, *The SwitchWare Active Network Architecture, 1998*, University of Pennsylvania.
- [17] D. Decasper, B. Plattner, *DAN: Distributed Code Caching for Active Networks*, 1998, Proceedings of INFOCOM '98.
- [18] D. Tennenhouse, J. Smith, W. Sincoskie, D. Wetherall, G. Minden, *A survey of Active Network Research*, 1996.
- [19] AN Node OS Working Group, *NodeOS Interface Specification*, 2001
- [20] Active Networks, <http://users.forthnet.gr/ath/glynos/active.html> Ιστοσελίδα με links σε ενδιαφέροντα papers σχετικά με την έρευνα που διεξάγεται γύρω από τα ενεργά δίκτυα.
- [21] Active Network Backbone (Abone) <http://www.isi.edu/abone/> Η δικτυακή σελίδα του Information Sciences Institute σχετικά με την πορεία του πειράματος ABONE.
- [22] The active networking homepage <http://www.itc.ukans.edu/~kulkarn/Active-Networks.html> Δικτυακός τόπος με άφθονο υλικό γύρω από τα ενεργά δίκτυα και κυρίως σχετικά με την αρχιτεκτονική Magician.
- [23] S. Berson, B. Braden, L. Ricciulli, *Introduction to the ABONE*, 2000.
- [24] S. Dawson, *ABCd: The Abone Control Daemon*, 2001.
- [25] S. Bhattacharjee, K. Calvert, E. Zegura, *Reasoning about active network protocols*, Atlanta.
- [26] S. Merugu, S. Bhattacharjee, E. Zegura, K. Calvert, *BOWMAN: a node OS for Active Networks*, 2000, Infocom, Tel Aviv.