

Abstract

Programmable networks are a novel approach to network architecture and they rely on the idea that a fraction of computations should be transferred in the network, from end - points. There are two main approaches to programmable networks : programmable switches and capsules. Both of them support faster service innovation service, flexibility and faster evolution of network industry.

Περίληψη

Τα προγραμματιζόμενα δίκτυα είναι μια καινοτόμα προσέγγιση στην αρχιτεκτονική των δικτύων και βασίζονται στην ιδέα ότι ένα μέρος των υπολογισμών που εκτελούνται στα άκρα του δικτύου πρέπει να μεταφερθούν μέσα στο δίκτυο. Υπάρχουν δύο βασικές προσεγγίσεις των προγραμματιζόμενων δικτύων : τα προγραμματιζόμενα switches και οι κάψουλες. Και οι δυο υποστηρίζουν ευελιξία, ταχύτερη παροχή νέων υπηρεσιών και πρόοδο της βιομηχανίας δικτύων.

Εισαγωγή

Στα παραδοσιακά τηλεπικοινωνιακά δίκτυα όλη η ευφυΐα του δικτύου εναπόκειται στον χειριστή (operator) του δικτύου. Σύμφωνα με την κλασική φιλοσοφία του Internet, το δίκτυο έχει πολύ μικρή ευφυΐα και η επικοινωνία των άκρων ελέγχεται από εφαρμογές που τρέχουν σε αυτά. Ωστόσο, καθώς τα τερματικά γίνονται εξυπνότερα και τα συστήματα σηματοδοσίας πιο ευαίσθητα (responsive) στις ανάγκες των πελατών και επίσης καθώς γίνεται προσπάθεια παροχής υπηρεσιών διακεκριμένης ποιότητας στο Internet (QoS), οι διαφορές τερματικών - συστημάτων σηματοδοσίας τείνουν να εκλείψουν.

Το επίπεδο ποιότητας παροχής υπηρεσιών θα αποκτά ολοένα και περισσότερη σημαντικότητα καθώς στο μέλλον αναμένεται η μεταφορά ήχου, video και δεδομένων σε μεγάλες ποσότητες και από περισσότερους χρήστες.

Η πρόσφατη πρόοδος στα καταμεμημένα συστήματα και στο μεταφέρσιμο (transportable) λογισμικό σε συνδυασμό με την αυξανόμενη απαίτηση για καλύτερο έλεγχο των πόρων του δικτύου οδηγούν σε μια νέα θεώρηση της αρχιτεκτονικής των δικτύων.

Η νέα προοπτική είναι αρχιτεκτονικές για δυναμικά προγραμματιζόμενα δίκτυα που υποστηρίζουν γρήγορη δημιουργία υπηρεσιών και διαχείριση πόρων μέσω ενός συνδυασμού από εφαρμογές που είναι ενήμερες για το δίκτυο και δίκτυα που είναι ενήμερα για τις εφαρμογές.

Μια ενδιαφέρουσα αρχική προσέγγιση είναι η σύγκριση των αρχιτεκτονικών των ενδιάμεσων κόμβων (δρομολογητών, switches κ.τ.λ.) και των άκρων (PC, servers, workstations).

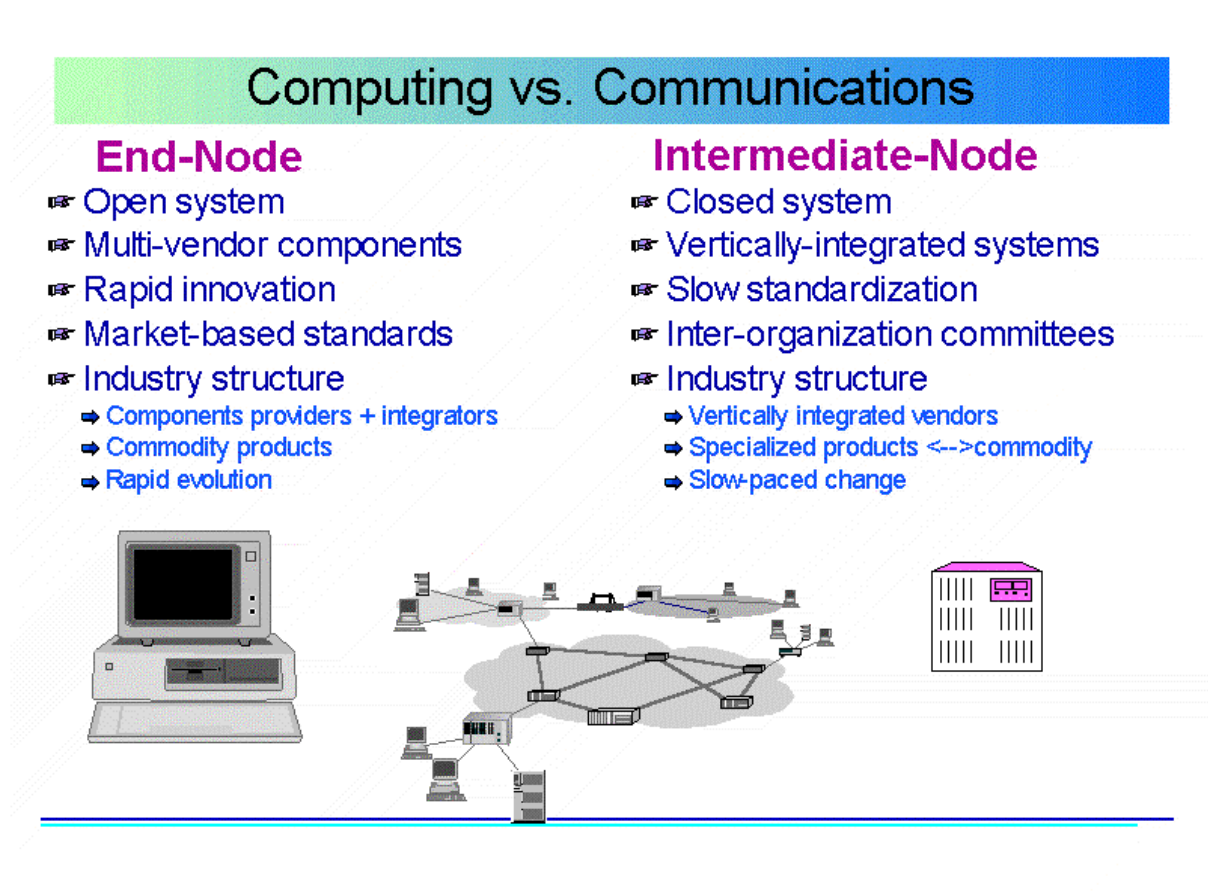
Τα τελικά άκρα (end nodes) είναι «ανοιχτά κουτιά» που δημιούργησαν μια τεράστια αγορά για λογισμικό και hardware. Έτσι όταν οι κατασκευαστές προωθούν στην αγορά π.χ. λογισμικό, αναμένουν ότι ένας πολύ μεγάλος αριθμός end nodes θα μπορεί να το

εκτελέσει. Αποτέλεσμα αυτού του γεγονότος είναι πολλές επενδύσεις και μεγάλη τεχνολογική εξέλιξη.

Σε αντίθεση με τα παραπάνω, η βιομηχανία των ενδιάμεσων κόμβων παραδίδει στην αγορά «κλειστά» συστήματα. Αυτή η τακτική θυμίζει περίπου την αγορά των υπολογιστών πριν 15 χρόνια όταν ένας μικρός αριθμός κατασκευαστών προμήθευε την αγορά με «κλειστά» συστήματα mainframes, κατάσταση που όπως είναι γνωστό άλλαξε δραματικά.

Κάτι τέτοιο φαίνεται ότι θα πρέπει να γίνει και στην αγορά των ενδιάμεσων κόμβων, ανοιχτές αρχιτεκτονικές δηλαδή με συστήματα που μπορούν να φέρουν εξαρτήματα και λογισμικό από πολλούς και διάφορους προμηθευτές έτσι ώστε να υπάρξει παρόμοια εξέλιξη όπως αυτή των PC.

Η σύγκριση μεταξύ των δύο μορφών της παραγωγής και διάθεσης των προαναφερθέντων προϊόντων φαίνεται συνοπτικά και παρακάτω : [4]



Προβλήματα των σημερινών δικτύων

Έχει παρατηρηθεί στα τρέχοντα δίκτυα, μείωση της απόδοσής τους λόγω περιττών λειτουργιών σε διάφορα επίπεδα των πρωτοκόλλων.

Επίσης, είναι αποδεδειγμένη η δυσκολία ενσωμάτωσης νέων υπηρεσιών στα υπάρχοντα αρχιτεκτονικά μοντέλα.

Στα σημερινά δίκτυα πολλές από τις υπηρεσίες παρέχονται από τους lead users (firewalls, web proxies, transport gateways, κ.τ.λ). Θα ήταν ευχής έργον να αντικαταστήσουμε τις πολύπλοκες και «ad hoc» διεργασίες των lead users με μια γενικότερη δυνατότητα των χρηστών να προγραμματίζουν τα δίκτυά τους.

Ενας σημαντικός λόγος που θα πρέπει να προσαρμόσουμε τους ενδιάμεσους κόμβους στη λογική των PC είναι ότι θα πρέπει να ενθαρρυνθεί η εισαγωγή νέων τεχνολογιών στην αγορά. Ας πάρουμε σαν παράδειγμα τα πολυμέσα. Ενώ τα PC μπορούν να εκμεταλλευτούν τις δυνατότητές τους, οι υπάρχουσες τεχνολογίες δικτύων ίσως πρέπει να περιμένουν 5-10 χρόνια για να πετύχουν κάτι αντίστοιχο [4].

Στη βιομηχανία δικτύων χρειάζεται τυποποίηση για την επικοινωνία διατάξεων προερχόμενων από διάφορους προμηθευτές, που επικοινωνούν μεταξύ τους (συνήθως) μέσω καλωδίων. Η τυποποίηση αυτή απαιτεί πολλές και περίπλοκες διαδικασίες από επιτροπές που μπορεί να διαρκέσουν και μια δεκαετία για την εισαγωγή μιας νέας τεχνολογίας. Αντίθετα στη βιομηχανία των PC οι αλλαγές είναι ραγδαίες (π.χ. στην ταχύτητα των επεξεργαστών).

Η καθυστέρηση αυτή της εισαγωγής μιας νέας τεχνολογίας δημιουργεί αβεβαιότητα στους κατασκευαστές οι οποίοι διστάζουν να επενδύσουν σε μια νέα τεχνολογία.

Ένας άλλος σημαντικός παράγοντας για τα προϊόντα δικτύων είναι η οικονομία τους. Αν συγκρίνουμε το μερίδιο του λογισμικού με αυτό του hardware θα δούμε ότι για έναν ενδιάμεσο κόμβο το μερίδιο του λογισμικού είναι μεγαλύτερο. Οι κατασκευαστές των διατάξεων δικτύου επενδύουν μεγάλα ποσά για την ανάπτυξη λογισμικού που όμως μπορούν να το πουλήσουν μόνο στους πελάτες που προμηθεύτηκαν το αντίστοιχο hardware, δηλαδή σε ένα μικρό ποσοστό της αγοράς. Σίγουρα η κατάσταση θα ήταν καλύτερη για τους πελάτες και τους κατασκευαστές αν οι τελευταίοι μπορούσαν να κατασκευάσουν λογισμικό που απευθύνεται σε όλη την αγορά, έτσι ώστε να περιμένουν γρήγορη απόσβεση της επένδυσής τους.

Είναι επίσης ενδιαφέρον να σημειώσουμε ότι είναι δύσκολο να ξεχωρίσουμε την αρχιτεκτονική των ενδιάμεσων και των τελικών κόμβων. Και οι δυο έχουν διαύλους πάνω στους οποίους προσαρμόζονται κάρτες οι οποίες διεκπεραιώνουν συγκεκριμένες εργασίες.

Στην περίπτωση των τελικών κόμβων τυποποιούνται οι διεπαφές των διαύλων οπότε ο καθένας μπορεί να κατασκευάσει κάρτες που να προσαρμόζονται στους διαύλους. Στην περίπτωση των ενδιάμεσων κόμβων οι δίαυλοι είναι «πατέντα» άρα οι κάρτες ακολουθούν κάθε φορά μια συγκεκριμένη τυποποίηση ανά κατασκευαστή.

Οι ενεργοί κόμβοι μπορούν να εκτελέσουν πολλά διαφορετικά προγράμματα (π.χ. μπορούν να εκτελέσουν διάφορους υπολογισμούς σε κάθε ένα από τα πακέτα που διέρχονται μέσω αυτών). Έτσι αυξάνουν το επίπεδο της «αυθαιρεσίας» επιτρέποντας τις εφαρμογές να παράγουν προσαρμοσμένα μηνύματα ανάλογα με τους σκοπούς τους.

Ορισμοί προγραμματιζόμενων δικτύων

- Ένα προγραμματιζόμενο δίκτυο είναι ένα δίκτυο στο οποίο οι χρήστες μπορούν να προγραμματίσουν switches και άλλες διατάξεις (devices) του δικτύου ώστε να τις

προσαρμόσουν στις ανάγκες τους. Ωστόσο το δίκτυο θα πρέπει να συνεχίσει να είναι ομοιογενές και ανθεκτικό (resistant) σε τεχνολογικά και ανθρώπινα λάθη (safe) καθώς και ασφαλές (secure) [1].

- Τα ενεργά δίκτυα είναι μια νέα προσέγγιση της αρχιτεκτονικής των δικτύων, στα οποία τα switches διεκπεραιώνουν προσαρμοσμένους (customized) υπολογισμούς στα μηνύματα (messages) που διέρχονται μέσω αυτών [6].
- Ένα ενεργό δίκτυο είναι το δίκτυο του οποίου οι κόμβοι προγραμματίζονται δυναμικά [4].

Προτάσεις

Έτσι όπως το λογισμικό εκτελείται στους τελικούς κόμβους, έτσι θα γίνεται και στους ενδιάμεσους. Αυτό το λογισμικό θα μπορούσε να κατασκευαστεί και να εκτελεστεί από οποιονδήποτε. Έτσι ο κάθε προμηθευτής θα μπορούσε να εισάγει νέα πρωτόκολλα που υποστηρίζουν καινοτόμες λειτουργίες. Αυτό το λογισμικό, πακεταρισμένο σε κινητούς «πράκτορες» (mobile agents) θα μπορούσε να ενεργοποιείται από τους παροχές του δικτύου αλλά και τους χρήστες.

Για παράδειγμα, ο χρήστης ενός ενεργού δικτύου θα μπορούσε να στέλνει ένα ίχνος (trace) προγράμματος σε κάθε δρομολογητή και να καθορίζει πότε θα εκτελεστεί το πρόγραμμα. Το παρακάτω σχεδιάγραμμα δείχνει πως οι δρομολογητές ενός IP δικτύου μπορούν να αυξηθούν για να εκτελούν τέτοιες προσαρμοσμένες διαδικασίες. Αυτοί οι ενεργοί δρομολογητές μπορούν να συνεργάζονται με τους ήδη υπάρχοντες δρομολογητές [6].

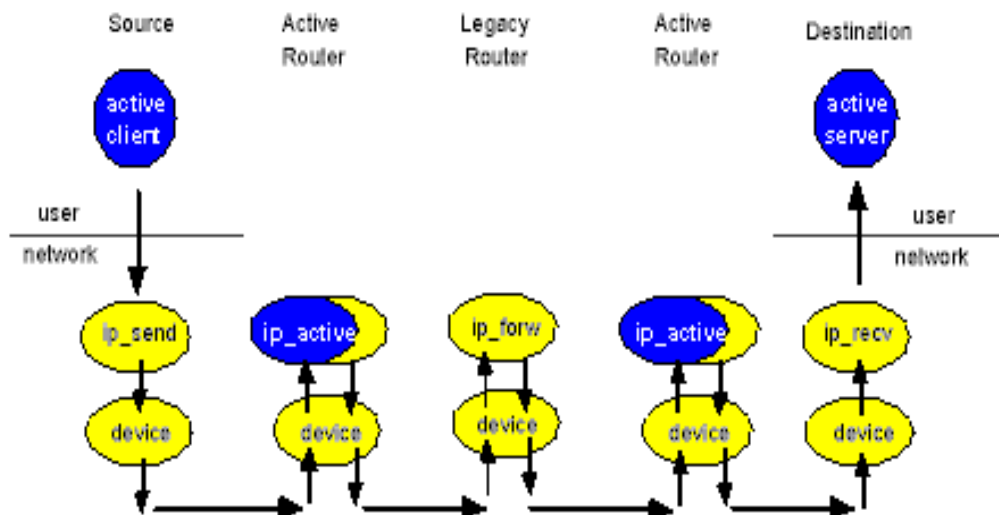


Figure 1. Application-specific processing within the nodes of an Active Network

Μια αρχιτεκτονική πρόταση των ενεργών κόμβων φαίνεται διαγραμματικά στο επόμενο σχήμα και θα αναλυθεί παρακάτω :

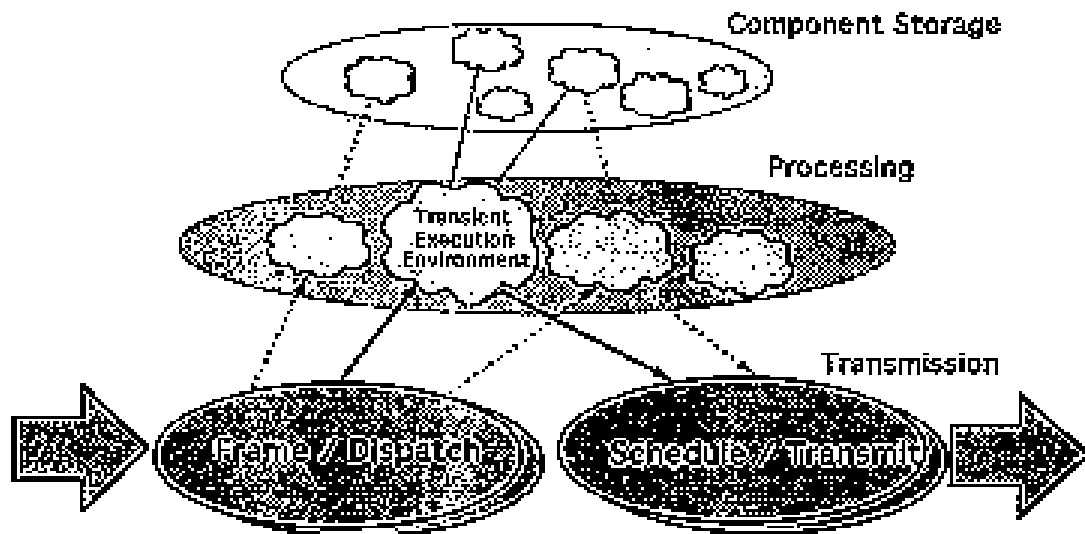


Figure 1. Active Node Organization

Αυτά τα δίκτυα είναι ενεργά από την άποψη ότι οι κόμβοι τους εκτελούν υπολογισμούς και τροποποιούν τα περιεχόμενα των πακέτων. Επιπρόσθετα, οι διαδικασίες μπορούν να προσαρμοστούν ανάλογα με τον χρήστη ή την εφαρμογή. Αντίθετα, στα κλασικά δίκτυα όπως στο Internet, ο ρόλος των υπολογισμών είναι εξαιρετικά περιορισμένος. Ένα άλλο μείζον θέμα είναι το τι λειτουργίες θα επιτελεί το λογισμικό και κατά ποιον τρόπο.

Παρακάτω παρουσιάζονται ορισμένες εναλλακτικές προτάσεις για τη λειτουργία των προγραμματιζόμενων δικτύων :

Ο διαχωρισμός ολοκληρωμένης και διακριτής πρότασης αφορά στο αν τα προγράμματα και τα δεδομένα μεταφέρονται ξεχωριστά.

Προγραμματιζόμενα switches - μια διακριτή προσέγγιση

διατηρεί την υπάρχουσα φόρμα (format) πακέτου / κελιού και παρέχει έναν διακριτό μηχανισμό που υποστηρίζει το «κατέβασμα» (downloading) των προγραμμάτων.

Η επεξεργασία των μηνυμάτων μπορεί να διαχωριστεί αρχιτεκτονικά από τη διοχέτευση προγραμμάτων στους κόμβους, με ξεχωριστούς μηχανισμούς για κάθε διεργασία. Αυτό διατηρεί τον υπάρχοντα διαχωρισμό μεταξύ της μεταφοράς δεδομένων εντός του εύρους ζώνης και των καναλιών διαχείρισης εκτός μάντας. Οι χρήστες θα διοχετεύουν πρώτα τις ρουτίνες τους στους απαιτούμενους δρομολογητές. Έπειτα θα στέλνουν τα πακέτα τους μέσω προγραμματιζόμενων κόμβων όπως περίπου γίνεται σήμερα. Όταν ένα πακέτο φτάνει σε έναν κόμβο, εξετάζεται το header του και εκτελείται το κατάλληλο πρόγραμμα.

Οι ξεχωριστοί μηχανισμοί για το φόρτωμα και την εκτέλεση μπορούν να είναι πολύτιμοι όταν το φόρτωμα του προγράμματος πρέπει να ελέγχεται πολύ προσεκτικά. Επιτρέποντας στους χειριστές να φορτώνουν κώδικα δυναμικά στους δρομολογητές τους θα ήταν χρήσιμο για σκοπούς εκτασιμότητας ακόμα κι αν τα προγράμματα δεν εκτελούν υπολογισμούς προσαρμοσμένους στο χρήστη ή στις εφαρμογές.

Η παραπάνω προσέγγιση φαίνεται διαγραμματικά στο παρακάτω σχήμα :

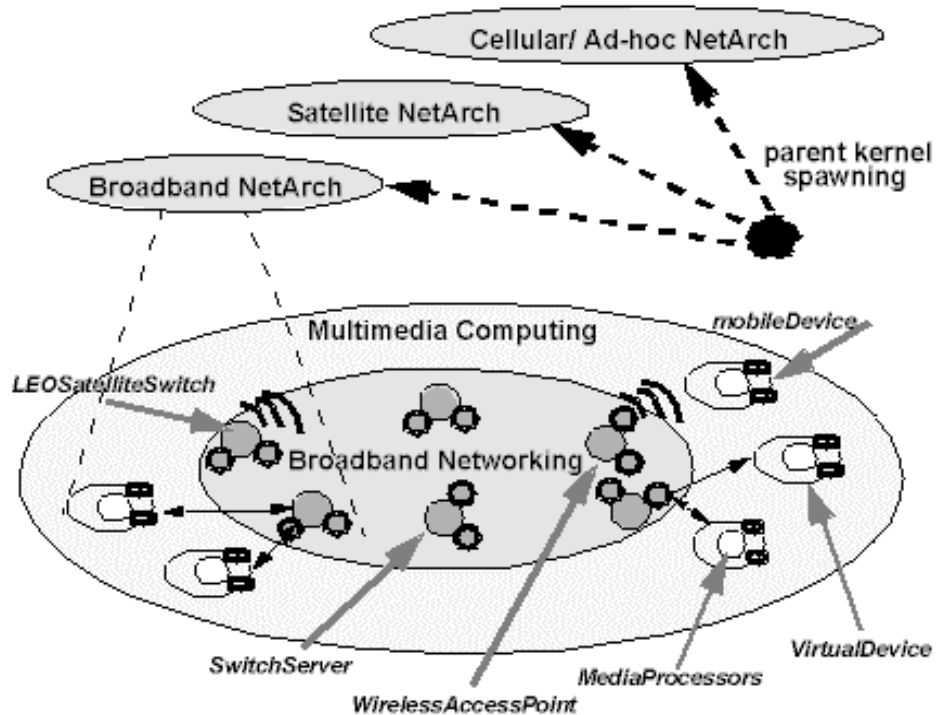


Figure 1. Programmable Networks

Κάψουλες - μια ολοκληρωμένη προσέγγιση

Η προσέγγιση της κάψουλας (capsule) αντικαθιστά τα υπάρχοντα παθητικά πακέτα με μικρά προγράμματα. Κάθε μήνυμα είναι ένα πρόγραμμα. Κάθε μήνυμα ή κάψουλα περιέχει ένα μέρος του προγράμματος ή τουλάχιστον μια εντολή και μπορεί να περιέχει ενσωματωμένα δεδομένα.

Όταν μια κάψουλα φτάνει σε έναν ενεργό κόμβο τα περιεχόμενά της αξιολογούνται με τον ίδιο τρόπο που ένας PostScript εκτυπωτής μεταφράζει τα περιεχόμενα κάθε αρχείου που λαμβάνει.

Τα bits που φτάνουν επεξεργάζονται από ένα μηχανισμό που προσδιορίζει τα όρια της κάψουλας. Τα περιεχόμενα της κάψουλας εκτελούνται σε ένα προσωρινό περιβάλλον όπου μπορούν να αξιολογηθούν με ασφάλεια.

Καθώς η κάψουλα κινείται στο δίκτυο μπορεί να εκτελεί κάποιες λειτουργίες στους κόμβους και να αφήνει κατάλοιπα (residues) για τις επόμενες κάψουλες.

Κωδικοποίηση Προγραμμάτων

Τα προγράμματα του δικτύου θα πρέπει να φορτώνονται από διάφορες πλατφόρμες και να μεταφέρονται πάνω από διάφορα δικτυακά υποστρώματα. Έτσι, θα πρέπει να υπάρξει μια τυποποίηση για την κωδικοποίηση των προγραμμάτων δικτύου, τα αρχικά built in σε κάθε κόμβο και την περιγραφή και κατανομή των πόρων των κόμβων.

Θα πρέπει να δοθεί ιδιαίτερη προσοχή στους παράγοντες που θα πρέπει να πληρούνται κατά τη δημιουργία ενός κοινού μοντέλου :

mobility : η ικανότητα μεταφοράς και εκτέλεσης προγραμμάτων πάνω σε ένα ευρύ φάσμα πλατφόρμων

Ασφάλεια : η ικανότητα να περιοριστούν οι πόροι στους οποίους μπορούν τα προγράμματα να έχουν πρόσβαση

Αποδοτικότητα : να επιτευχθούν τα παραπάνω χωρίς να έχουμε επιπτώσεις στην απόδοση

Για να υλοποιηθούν τα προγραμματιζόμενα δίκτυα χρειάζεται τεχνολογία mobile agent για να εκτελείται το λογισμικό σε απομακρυσμένους κόμβους. Έχουν ήδη προταθεί διάφορες τεχνολογίες όπως φαίνεται και στον παρακάτω πίνακα 1 με αξιολόγηση των τριών παραγόντων που προαναφέρθηκαν . Εναλλακτικά θα μπορούσε να χρησιμοποιηθεί τεχνολογία ανεξάρτητη γλώσσας όπως οι εξουσιοδοτημένοι πράκτορες (agents) που αναπτύχθηκε στο εργαστήριο DCC για την εκτέλεση προγραμμάτων διαχείρισης σε απομακρυσμένους κόμβους. Αυτός ο τρόπος επιτρέπει να χρησιμοποιούμε προγράμματα στους απομακρυσμένους κόμβους, γραμμένα σε οποιαδήποτε γλώσσα.

Project	M	S	E	Περιγραφή
Safe Tel (source)	X	X		γλώσσα τύπου script που παρέχει ασφάλεια μέσω μετάφρασης ενός πηγαίου κώδικα και περιορισμό του περιθωρίου ονοματολογίας του. Εξαρτάται από το πόσο σωστός είναι ο μεταφραστής και από περιορισμούς για το αν τα προγράμματα θα βρεθούν σκόπιμα ή από λάθος, πέρα από το προβλεπόμενο περιβάλλον εκτέλεσής τους
Java (intermediate)	X	X	x	η Java χρησιμοποιεί ένα ενδιάμεσο set εντολών για να καταφέρει τη μεταφερσιμότητα (mobility). Συνήθως η ασφαλής εκτέλεση ενδιάμεσου κώδικα εναπόκειται στην προσεκτική μετάφρασή του. Ένα από τα πλεονεκτήματα της Java είναι ότι ο σχεδιασμός της ελαχιστοποιεί το φόρτο του μεταφραστή.
Omniware (object code)	x	X	X	ο φορητός κώδικας Omniware εξαρτάται σε απομόνωση των σφαλμάτων βασισμένη σε λογισμικό. Καθορίζει μια ομάδα κανόνων που οι εντολές πρέπει να τηρούν . Μέσω αυτών των κανόνων σχηματίζεται μια περιοχή «sandbox» που περικλείει το πρόγραμμα.

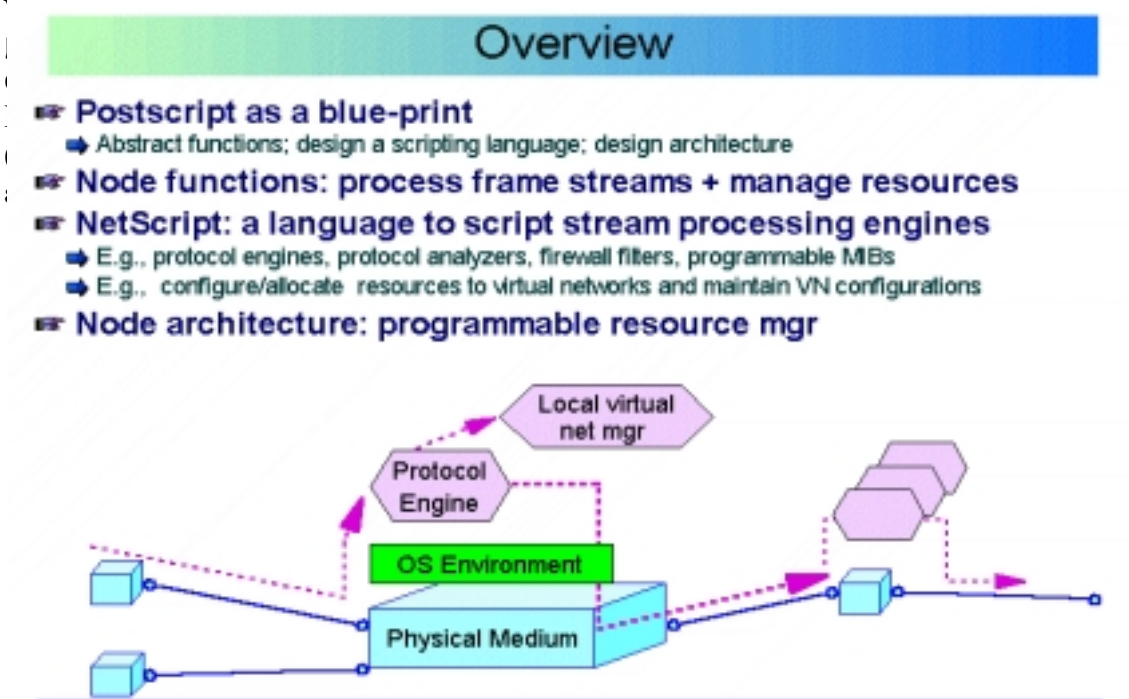
Proof-Carrying Code (object code)		X	X	χρησιμοποιείται μια νέα προσέγγιση για να επιτευχθεί ασφάλεια : προσαρμόζει τυπική πιστοποίηση των ιδιοτήτων του δυαδικού προγράμματος και ο αποδέκτης ελέγχει αν ισχύει. Αυτή η προσέγγιση είναι προς το παρόν πρακτική μόνο για μικρά προγράμματα.
-----------------------------------	--	---	---	--

Πίνακας 1. Τεχνολογίες κωδικοποίησης (M = mobility, S = safety, E = efficiency)

Μια εναλλακτική προσέγγιση είναι αυτή του Netscript project [4] και η οποία είναι παρόμοια με τη λογική των προγραμμάτων Postscript των εκτυπωτών. Η Postscript εστιάζεται στο να διαχωρίζει τις αρχικές (primitive) λειτουργίες και να τις εγκλωβίζει σε μια υψηλού επιπέδου script γλώσσα και στο να δημιουργεί ένα περιβάλλον εικονικής εκτυπωτικής μηχανής για να επεξεργαστεί αυτά τα scripts. Χάρη σ' αυτόν το διαχωρισμό, οι κατασκευαστές μπορούν να παράγουν ομοιογενή προϊόντα και να εκμεταλλευτούν νέες τεχνολογίες και δυνατότητες. Ουσιαστικά η Postscript τυποποίησε το περιβάλλον εκτέλεσης και τη γλώσσα διαχωρίζοντάς τα.

Ομοίως, η Netscript ακολουθεί τη δημιουργία ενός κοινού set διαχωρισμού των πρωταρχικών λειτουργιών των κόμβων του δικτύου, εγκλωβίζοντάς τες σε μια script γλώσσα και αναπτύσσοντας ένα περιβάλλον (virtual machine) όπου θα επεξεργάζονται τα scripts.

Ουσιαστικά, οι κόμβοι του δικτύου είναι μηχανές (engines) που επεξεργάζονται ροή πακέτων και διαχειρίζονται τους πόρους του δικτύου. Τα scripts της Netscript μπορούν να χρησιμοποιηθούν για να φτιάξουν μηχανές επεξεργασίας πακέτων και να τις διαμορφώσουν για να υποστηρίζουν τη ροή των πακέτων μεταξύ αυτών των μηχανών.. Τα scripts μπορούν να δημιουργήσουν πρωτόκολλα δρομολόγησης, firewalls, η εικονικά δίκτυα (virtual networks). Ένας προγραμματιζόμενος κόμβος ένα περιβάλλον στο οποίο μπορούν να εκτελεστούν προγράμματα σε Netscript. Τα πακέτα επεξεργάζονται από αυτά τα προγράμματα και εκτελούνται σε άλλες μηχανές έως ότου



Αρχιτεκτονική δικτύων ANTS

Παρακάτω θα παρουσιαστεί μια ολοκληρωμένη πρόταση ενεργού δικτύου, του ANTS [7] έτσι ώστε να γίνει περισσότερη κατανοητή η δομή ενός τέτοιου δικτύου.

Όπως τα περισσότερα δίκτυα, το ANTS αποτελείται από διασυνδεδεμένους μεταξύ τους κόμβους που εκτελούν μια κοινή ρουτίνα.. Οι κόμβοι μπορούν να είναι συνδεδεμένοι στο τοπικό ή ευρύτερο (wide) επίπεδο και σημείο με σημείο ή με διαμοιραζόμενα κανάλια.

Αντίθετα με το IP, οι υπηρεσίες ενός δικτύου ANTS είναι ευέλικτες. Επιπρόσθετα από τις υπηρεσίες του IP το ANTS επιτρέπει την εισαγωγή νέων πρωτοκόλλων στο δίκτυο. Οι εφαρμογές το επιτυγχάνουν αυτό, καθορίζοντας τις ρουτίνες που θα εκτελεστούν στέλνοντάς τες στους ενεργούς κόμβους οι οποίοι με τη σειρά τους διαβιβάζουν (forward) τα μηνύματά τους. Σαν αποτέλεσμα, οι εφαρμογές προωθούν ένα μέρος των διεργασιών που επιτελούν στο δίκτυο, ή με τον συνήθη τρόπο (caching) ή με νέες μεθόδους που έχουν εφαρμογή μόνο στα ενεργά δίκτυα.

Στον σχεδιασμό του δικτύου ANTS έγινε προσπάθεια να επιτευχθούν τρεις στόχοι για τα καινοτόμα πρωτόκολλα δικτύων. Το δίκτυο θα πρέπει να υποστηρίζει :

- ✓ Ταυτόχρονη χρήση μιας ποικιλίας από διαφορετικού επιπέδου πρωτόκολλα
- ✓ Την κατασκευή και χρησιμοποίηση νέων πρωτοκόλλων με αμοιβαίες συμφωνίες μεταξύ των ενδιαφερόμενων αντί μιας περισσότερο συγκεντρωτικής ρύθμισης των πρωτοκόλλων
- ✓ Τη δυναμική εκτέλεση των νέων πρωτοκόλλων αφού είναι αδικαιολόγητο να θεωρούμε τμήματα του δικτύου off line για να τα ανασχηματίσουμε, ειδικά καθώς τα δίκτυα μεγαλώνουν

Για να επιτευχθούν οι παραπάνω στόχοι, χρησιμοποιήθηκαν τρεις νέες ιδέες

- ◇ Τα πακέτα των παραδοσιακών δικτύων αντικαταστάθηκαν από κάψουλες που η επεξεργασία τους γίνεται στους ενεργούς κόμβους. Οι τύποι των κάψουλων που διαμοιράζονται πληροφορία μέσα στο δίκτυο είναι ομαδοποιημένες σε πρωτόκολλα. Ένα πρωτόκολλο παρέχει μια υπηρεσία και είναι η μονάδα προσαρμογής και προστασίας του δικτύου
- ◇ Επιλεγμένοι δρομολογητές του Internet και γειτονικοί τελικών άκρων αντικαθιστούνται από ενεργούς κόμβους που τρέχουν τις κάψουλες ενός πρωτοκόλλου. Αντίθετα με τους συνηθισμένους δρομολογητές, οι ενεργοί κόμβοι παρέχουν ένα Application Programming Interface για την επεξεργασία των ρουτινών των κάψουλων και εκτελούν αυτές τις ρουτίνες με ασφάλεια χρησιμοποιώντας τεχνικές γλωσσών και του λειτουργικού συστήματος.

◇ Ένας μηχανισμός διαμερισμού του κώδικα εξασφαλίζει ότι οι ρουτίνες των κάψουλων θα μεταφερθούν αυτόματα και δυναμικά στους ενεργούς κόμβους όπου χρειάζονται.

Το μοντέλο που υποστηρίζουν τα παραπάνω στοιχεία για τον προγραμματισμό νέων δικτυακών υπηρεσιών έχει τα παρακάτω χαρακτηριστικά :

- * Η προωθούμενη ρουτίνα μιας κάψουλας δομείται από τον αποστολέα και δεν μπορεί να αλλάξει καθώς κινείται μέσα στο δίκτυο. Επίσης, κάψουλες που ανήκουν σε ένα πρωτόκολλο δε μπορούν να δημιουργήσουν ή να προσπελάσουν κάψουλες άλλου πρωτοκόλλου. Δεδομένων των προαναφερθέντων, ένας χρήστης Δε μπορεί να ελέγχει τις κάψουλες κάποιου άλλου χρήστη.
- * Μερικοί ενεργοί κόμβοι μπορούν να επιλέξουν να μην εκτελέσουν συγκεκριμένες ρουτίνες που προωθήθηκαν σε αυτούς, ανάλογα με τους διαθέσιμους πόρους του κόμβου και τις πολιτικές ασφάλειας που ακολουθούνται. Όταν συμβαίνει κάτι τέτοιο τότε ο κόμβος προωθεί την κάψουλα όπως στο IP. Επιπρόσθετα οι ίδιες οι κάψουλες μπορούν να επιλέξουν τους κόμβους που θα εκτελεστούν οι ρουτίνες τους ανάλογα με το που βρίσκονται οι κόμβοι και ανάλογα με τις δυνατότητές τους.
- * Από τη στιγμή που οι κάψουλες μπορούν να οριστούν από διάφορους χρήστες που ίσως υποπέσουν σε λάθη, υπάρχει ένας μηχανισμός Time - to - live. Αν δηλαδή δεν εκτελεστούν σε ένα σύντομο και προκαθορισμένο χρονικό διάστημα τότε προωθούνται.

Συστατικά της αρχιτεκτονικής ANTS

Πρωτόκολλα και κάψουλες

Η βάση για να έχουμε προσαρμοσμένες (customized) διεργασίες στο δίκτυο, είναι τα πρωτόκολλα που συντίθενται από μια ομάδα κάψουλων συσχετισμένων μεταξύ τους. Το format των κάψουλων καθώς μεταφέρονται σε ένα γενικό κανάλι διασυνδεδεμένων επιπέδων, είναι το κάτωθι :



Figure 1: Capsule Format

Όταν τρέχουν σε ένα δίκτυο IP με ενεργούς και κλασσικούς δρομολογητές, αυτή η διάταξη θα πρέπει να είναι συμβατή μ' αυτήν του IP. Οι κλασσικοί δρομολογητές IP σ' αυτήν την περίπτωση εμφανίζονται σαν ενεργοί που επέλεξαν να μην εκτελέσουν παραπάνω υπηρεσίες από τις κλασσικές.

Η πιο σημαντική λειτουργία του format της κάψουλας είναι να περιέχει έναν δείκτη για το πρωτόκολλο και για τη ρουτίνα που προωθείται μέσα στο πρωτόκολλο. Ο δείκτης

αυτός προέρχεται από τον κώδικα του πρωτοκόλλου ο οποίος είναι μέρος του (του πρωτοκόλλου) και είναι κρίσιμος για δυο λόγους :

- Μειώνει σημαντικά τον κίνδυνο «εξαπάτησης» (spoofing) του πρωτοκόλλου. Αυτό συμβαίνει διότι ένα αποτύπωμα βασισμένο σε ένα ασφαλές τμήμα είναι μια μονοσήμαντη συνάρτηση από τον κώδικα στον δείκτη. Κάθε ενεργός κόμβος μπορεί να κρίνει από μόνος του χωρίς τη βοήθεια εξωτερικού παράγοντα αν τα προγράμματα που δέχεται ταιριάζουν με έναν δοσμένο δείκτη .
- Επιτρέπει τους τύπους των πρωτοκόλλων και των κάψουλων αν κατανεμηθούν γρήγορα και με αποκεντρωμένο τρόπο αφού ο δείκτης τους εξαρτάται μόνο από το αποτύπωμα στον κώδικα του πρωτοκόλλου.

Εξαιτίας αυτών των ιδιοτήτων, η εν λόγω αρχιτεκτονική χρησιμοποιεί τα πρωτόκολλα σα μονάδες ασφάλειας του δικτύου, προλαμβάνοντας την περίπτωση το ένα πρωτόκολλο να συγκρουστεί με ένα άλλο μέσα στον ενεργό κόμβο. Η προστασία του δικτύου ανά πρωτόκολλο παρέχει ένα πιο αποτελεσματικό μοντέλο από αυτό της προστασίας ανά χρήστη, αφού Δε χρειάζεται πιστοποίηση για κάθε τροποποίηση των δεδομένων.

Κάποιες ρουτίνες είναι γνωστές για το ότι θα είναι διαθέσιμες σε κάθε κόμβο. Αυτές θα περιέχουν κυρίως ρουτίνες με συνήθη καθήκοντα όπως π.χ. μεταφορά μη αξιόπιστων δεδομένων . Άλλες ρουτίνες θα είναι ειδικές για εφαρμογές και δε θα μένουν σε κάθε κόμβο αλλά θα πρέπει να μεταφέρονται σ'έναν κόμβο από τον κώδικα για το σχέδιο κατανομής, πριν οι πρώτες κάψουλες αυτού του τύπου επεξεργαστούν.

Τα κατάλοιπα του format της κάψουλας αποτελούνται από μια συνήθη κεφαλίδα που περιέχει πεδία που υπάρχουν σε όλες τις κάψουλες, μια κεφαλίδα που εξαρτάται από τον τύπο της κάψουλας και που μπορεί να αλλάξει καθώς η κάψουλα κινείται στο δίκτυο και τις χρήσιμες πληροφορίες. Τα πιο σημαντικά στοιχεία της πρώτης κεφαλίδας είναι οι διευθύνσεις αποστολέα και παραλήπτη και πληροφορίες για τα όρια των πόρων που πρέπει να επιβληθούν από τους κόμβους.

Ενεργοί κόμβοι

Μια δυσκολία στο σχεδιασμό ενός προγραμματιζόμενου δικτύου είναι το να επιτρέψουμε στους κόμβους να εκτελούν προγράμματα προσαρμοσμένα στο χρήστη χωρίς να επακολουθούν ανεπιθύμητες αλληλεπιδράσεις.

Η προσέγγιση του ANTS είναι να εκτελούνται τα πρωτόκολλα σ' ένα περιορισμένο περιβάλλον που ουσιαστικά περιορίζει την πρόσβασή τους στις διαθέσιμες πηγές.

Οι ενεργοί κόμβοι παίζουν αυτόν τον ρόλο. Εξάγουν ένα API για χρήση από ρουτίνες ορισμένες από την εφαρμογή (application defined) , που συνδυάζει τους παραπάνω κανόνες χρησιμοποιώντας τις αρχές μιας γλώσσας προγραμματισμού γενικών εφαρμογών και όχι ένα πιο αυστηρό μοντέλο. Επίσης οι ενεργοί κόμβοι διαμοιράζουν τους πόρους στα πρωτόκολλα και καθορίζουν περιορισμούς για το πως αυτοί οι πόροι θα χρησιμοποιηθούν καθώς εκτελούνται τα πρωτόκολλα.

API που προωθεί κάψουλες (Capsule forwarding API)

Οι σχεδιαστές της αρχιτεκτονικής ANTS διαλέγουν ένα αρχικό API μ' ένα προηγθέν (predecessor) σύστημα. Απ' το προηγθέν αυτό σύστημα (μέθοδο) προτείνεται ότι ένα σχετικά μικρό set λειτουργιών είναι επαρκές για να εκφράσει έναν ικανοποιητικό αριθμό διαφορετικών και χρήσιμων ρουτινών προώθησης. Προς το παρόν υποστηρίζονται οι παρακάτω αρχές για τους κόμβους με μερικές βέβαιες μελλοντικές προσθήκες (πιστοποίηση, ταυτοποίηση, συμπύκνωση κ.τ.λ.) :

- προσβασιμότητα του περιβάλλοντος του κόμβου, ώστε να αντλούνται πληροφορίες για την τοποθεσία του κόμβου, την κατάσταση των συνδέσμων (links), πίνακες δρομολόγησης κ.τ.λ.
- διαχείριση των κάψουλων με πρόσβαση και στην κεφαλίδα και στα «ωφέλιμα» δεδομένα
- χειρισμοί ελέγχου για να επιτραπεί στις κάψουλες να δημιουργήσουν άλλες κάψουλες και να προωθήσουν, να αναστείλουν ή να αχρηστεύσουν τους εαυτούς τους οι ίδιες οι κάψουλες.
- αποθήκευση, για διαχείριση αντικειμένων ορισμένων από τις εφαρμογές που κρατούνται για μικρό χρονικό διάστημα.

Το API του κόμβου είναι σημαντικό γιατί καθορίζει το είδος των ρουτινών που μπορούν να συνθέσουν οι εφαρμογές. Για παράδειγμα, χωρίς τη δυνατότητα αποθήκευσης και πρόσβασης στην κατάσταση του κόμβου, τα αυτόνομα προγράμματα των κάψουλων δε θα μπορούν να επικοινωνήσουν μεταξύ τους. Επίσης, η αποδοτικότητα της συμπύκνωσης και της εκτέλεσης επηρεάζονται από τις παραπάνω αρχές.

Εκτέλεση κάψουλων

Ο ενεργός κόμβος πρέπει να παρέχει ένα περιβάλλον που εκτελεί τις ρουτίνες που χρησιμοποιούν το API, παρέχοντας ασφάλεια και επιτυχή διαχείριση των διαθέσιμων πόρων. Από τη στιγμή που η διαδικασία που ακολουθούν οι κάψουλες μοιάζουν με ένα καταναμημένο προγραμματιστικό σύστημα στο οποίο συμμετέχουν πολλοί δικαιούχοι χρήστες με μικρά καθήκοντα, η πιστοποίηση και άλλα παραδοσιακά σχήματα ασφάλειας φαίνεται να είναι μη δόκιμα στην περίπτωση των ενεργών δικτύων για τα οποία θα προτιμηθούν τεχνολογίες φορητού κώδικα όπως έχει ήδη προαναφερθεί (Java κ.τ.λ.). Αντιστρόφως, η περιστασιακή χρήση κανόνων που χειρίζονται τους διαμερισμένους πόρους πρέπει να πιστοποιούνται (π.χ. ενημέρωση των πινάκων δρομολόγησης).

Επιπρόσθετα, ο κόμβος περιορίζει τους πόρους που μπορούν να χρησιμοποιήσουν τα προγράμματα της κάψουλας και διαχειρίζεται όποια λάθη προκύψουν (όπως προαναφέρθηκε η κάψουλα εμπεριέχει έναν περιορισμό στη χρήση των πόρων που λειτουργεί σαν TTL). Αυτό το όριο μεταφέρεται με την κάψουλα και μειώνεται από τους κόμβους καθώς αυξάνεται η κατανάλωση των πόρων. Μόνο οι κόμβοι μπορούν να αλλάξουν αυτό το πεδίο ή να αχρηστεύσουν κάψουλες που το όριό τους γίνεται 0. Προσοχή επίσης πρέπει να ληφθεί στο να μεταφέρονται πόροι όταν μια κάψουλα

δημιουργεί μια άλλη , έτσι ώστε να καθορίζονται τα συνολικά όρια του δικτύου. Είναι σημαντικό να χρεώνονται οι πόροι όταν καταναλώνονται. Ο χρόνος επεξεργασίας και το εύρος των links κατανέμονται σε χρόνο και σε μέρη κάψουλων, αντιστοίχως. Η μνήμη του κόμβου κατανέμεται σε cached αντικείμενα.

Διανομή κώδικα

Το τρίτο στοιχείο της αρχιτεκτονικής ANTS είναι ένα σύστημα διανομής κώδικα. Δοσμένης μιας προγραμματιζόμενης δομής, είναι απαραίτητος ένας μηχανισμός που θα διαδίδει τους ορισμούς των προγραμμάτων, όπου αυτοί είναι απαραίτητοι. Ένα καλό σχήμα θα πρέπει να είναι αποδοτικό, να προσαρμόζεται σε αλλαγές όσο αφορά στη συνδεσιμότητα (connectivity) των κόμβων και να περιορίζει τις δραστηριότητές του ώστε το δίκτυο να παραμένει συμπαγές.

Πολλοί διαφορετικοί μηχανισμοί είναι εφικτοί. Σε μια πρόταση, τα προγράμματα μπορούν να μεταφέρονται μέσα στις κάψουλες. Αυτή η εκδοχή είναι κατάλληλη όταν οι κάψουλες χρειάζεται να μεταφέρουν πολύ μικρά προγράμματα και όταν το εύρος ζώνης δεν είναι πρώτη προτεραιότητα. Μια άλλη πρόταση είναι να φορτώνονται τα προγράμματα σε κάθε κόμβο που μπορεί να τα χρειαστούν χρησιμοποιώντας ένα κανάλι εκτός εύρους ζώνης ή κανάλι διαχείρισης (management channel), πριν να χρησιμοποιηθεί ένα νέο πρωτόκολλο. Αυτή όμως η πρόταση δεν ενδείκνυται για ένα αποκεντρωμένο σύστημα.

Η πρόταση της αρχιτεκτονικής ANTS είναι να ζευχθεί η μεταφορά του κώδικα με τη μεταφορά των δεδομένων, εντός εύρους ζώνης. Τα πλεονεκτήματα αυτής της προσέγγισης είναι ότι περιορίζει τη διανομή του κώδικα μόνο εκεί όπου αυτός είναι απαραίτητος ενώ ταυτόχρονα ο κώδικας προσαρμόζεται στον κόμβο και σε ενδεχόμενα λάθη της σύνδεσης. Επίσης βελτιώνει το start up του δικτύου και διευκολύνει τα πρωτόκολλα μικρής διάρκειας επικαλύπτοντας τη διανομή του κώδικα με την εκτέλεσή του.

Σύμφωνα με την παραπάνω πρόταση, σχεδιάστηκε ένα διάγραμμα προσαρμοσμένο σε ροές π.χ. σειρά από κάψουλες που ακολουθούν το ίδιο μονοπάτι και χρειάζονται την ίδια επεξεργασία. Στα άκρα, οι εφαρμογές μπορούν να αρχίσουν να χρησιμοποιούν ένα νέο πρωτόκολλο οποιαδήποτε στιγμή, εγγράφοντας τους ορισμούς του κώδικα στον τοπικό τους κόμβο.. Τότε οι κάψουλες του νέου τύπου μπορούν να διοχετευθούν στο δίκτυο και να ληφθούν από αυτό. Στους ενδιάμεσους κόμβους, η αποθήκευση (caching) του κώδικα του πρωτοκόλλου γίνεται για να επιτευχθεί υψηλότερη απόδοση, αποφεύγοντας να φορτωθεί ο κώδικας μονομιάς με τη ροή μιας σειράς κάψουλων. Αρχικά, για να γεμίσουμε τις cache μνήμες , χρησιμοποιείται ένα πρωτόκολλο μικρού μεγέθους για να μεταφέρει των κώδικα του πρωτοκόλλου, σταδιακά από τον ένα κόμβο στον άλλο . Αυτός ο τρόπος διανομής κώδικα, φαίνεται στο παρακάτω σχήμα :

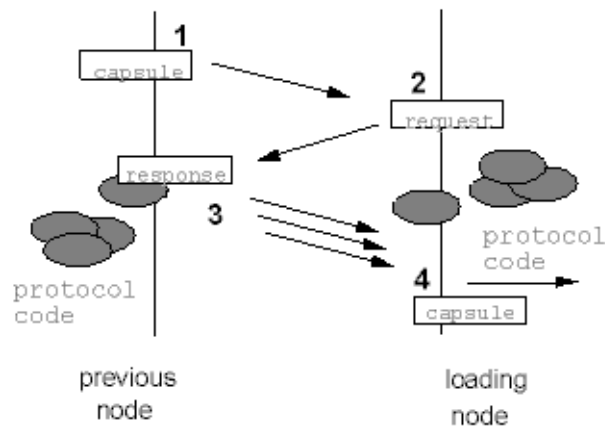


Figure 2: Demand Loading of Protocol Code

1. Όταν μια κάψουλα φτάνει σε έναν κόμβο, ελέγχεται ο αποθηκευμένος κώδικας του πρωτοκόλλου. Αν ο απαιτούμενος κώδικας δεν υπάρχει ολόκληρος στον κόμβο, στέλνεται μια αίτηση για το υπόλοιπο τμήμα του κώδικα. Η αίτηση ακολουθεί τη διαδρομή της κάψουλας και μπορεί να φτάσει στον τελευταίο ενεργό κόμβο αυτής της διαδρομής. Η εκτέλεση της κάψουλας αναβάλλεται για ένα πεπερασμένο χρονικό διάστημα.
2. Όταν ένας κόμβος λάβει την αίτηση και μπορεί να την απαντήσει, τότε απαντά αμέσως. Στέλνει δηλαδή τα απαιτούμενα τμήματα του κώδικα που λείπουν από τον αιτούμενο κόμβο.
3. Όταν ένας αιτούμενος κόμβος λαμβάνει απάντηση, ενσωματώνει τον κώδικα στην cache του. Αν τελικά έχει όλο τον απαιτούμενο κώδικα, ενεργοποιεί τις κάψουλες που είχαν αναβάλλει τη δραστηριότητά τους. Εάν δεν έρθει απάντηση σε κάποιο ορισμένο χρονικό διάστημα, τότε οι κάψουλες αχρηστεύονται.

Το τελευταίο είναι και ένα από τα μειονεκτήματα της πρότασης, αφού σημειώνεται απώλεια κάψουλων, πράγμα πολύ πιθανό με τη σημερινή συμφόρηση των δικτύων. Ωστόσο το παραπάνω σχήμα έχει το πλεονέκτημα (δεδομένου ενός άνω ορίου μεγέθους πρωτοκόλλων) ότι το ποσό της επεξεργασίας και του εύρους που θα επιβαρύνει το δίκτυο για το φόρτωμα του κώδικα, είναι περιορισμένα.

Νέες υπηρεσίες

Τα ενεργά δίκτυα θα κάνουν δυνατή την παροχή νέων υπηρεσιών που θα βασίζονται σε :

- συγχώνευση της πληροφορίας βασισμένης στο δίκτυο
- ενημερότητα του χρήστη για την προστασία του δικτύου
- διαχείριση του ενεργού δικτύου

Η συγχώνευση και διανομή της πληροφορίας

Ήδη βρισκόμαστε στην multi user και multi sites εποχή και αυτό το γεγονός επιβεβαιώνει ότι χρειαζόμαστε υπηρεσίες που θα υποστηρίζουν τη συμπύκνωση και διανομή της πληροφορίας. Ωστόσο τα υπάρχοντα συστήματα βασίζονται σε υπηρεσίες που παρέχουν πολύ περιορισμένες λειτουργίες (π.χ. αντιγραφή IP πακέτων) χωρίς υποστήριξη διανομής συγκεκριμένων εφαρμογών. Στο παρακάτω σχήμα φαίνεται πως multi site εφαρμογές θα αυξήσουν το επίπεδο υπολογισμών και αποθήκευσης μέσα στο δίκτυο.

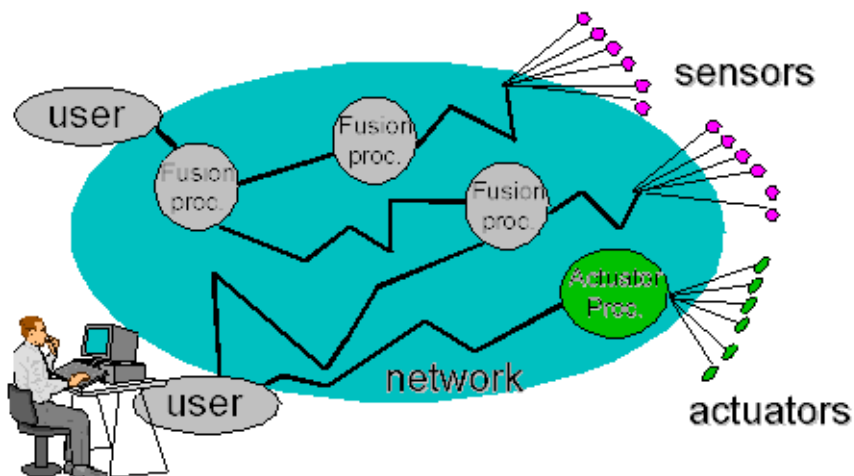


Figure 2. Exploiting the network-based merging and distribution of information.
(Diagram courtesy of Prof. Henry Fuchs, UNC)

Σε αυτό το παράδειγμα, μια εφαρμογή όπως π.χ. προσομοίωση κάθε χρήστης μπορεί να «δει» σύνθετες εικόνες που αποτελούνται από συνενωμένες πληροφορίες οι οποίες ανακτώνται από ένα μεγάλο αριθμό αισθητήρων. Επίσης, κάθε αισθητήρας μπορεί να είναι «ορατός» από πολλούς χρήστες που έχουν διαφορετικές απαιτήσεις ως προς τη μορφή της πληροφορίας στην οποία έχουν πρόσβαση. Ενοποιώντας τα δεδομένα μέσα στο δίκτυο μειώνουμε τις απαιτήσεις σε εύρος ζώνης για τους χρήστες. Ομοίως, οι ειδικές απαιτήσεις του χρήστη σε υπηρεσίες multi cast μέσα στο δίκτυο μειώνουν το φόρτο των αισθητήρων και του κορμού του δικτύου.

Ενημερότητα του χρήστη για την προστασία του δικτύου

Προστασία της πληροφορίας νοείται ότι η κατάλληλη πληροφορία φτάνει στους σωστούς ανθρώπους στο σωστό χρόνο και στο σωστό μέρος. Αν και έχουν προταθεί πολλοί μηχανισμοί ασφαλείας και πιστοποίησης, τα ενεργά δίκτυα παραδέχονται έναν ολοκληρωμένο μηχανισμό που διέπει όλους τους πόρους του δικτύου και τις πληροφορίες που διέρχονται μέσω αυτών.

Αυτή η προσέγγιση καταργεί την ανάγκη για πολλούς μηχανισμούς πιστοποίησης / ασφαλείας που τρέχουν ανεξάρτητα σε κάθε επίπεδο του πρωτοκόλλου.

Διαχείριση ενεργού δικτύου

Πολλά από τα καθήκοντα της διαχείρισης δικτύου περιλαμβάνουν τη συλλογή και σύγκριση δεδομένων. Για να παρέχουμε τα ποιο σημαντικά δεδομένα που αφορούν στη διαχείριση, θα πρέπει να φιλτράρουμε τα λιγότερα σημαντικά γεγονότα. Οι τεχνολογίες των ενεργών δικτύων θα μπορούσαν να χρησιμοποιηθούν για ένα τέτοιο φιλτράρισμα και παρακολούθηση των δεδομένων. Ομοίως, τα ενεργά δίκτυα διαθέτουν ευελιξία η οποία είναι απαραίτητη για τη βελτίωση ανίχνευσης λαθών .

Ασφάλεια και ασφάλιση προγραμματιζόμενων δικτύων

Η ασφάλεια και η ασφάλιση του δικτύου είναι δύο χαρακτηριστικά αξιοπιστίας του. Ένα ασφαλές (safe) σύστημα παρέχει προστασία από τα λάθη των χρηστών ενώ ένα ασφαλισμένο σύστημα , από μη εξουσιοδοτημένους χρήστες. Υπάρχει μεγάλου βαθμού επικάλυψη μεταξύ των μηχανισμών που εξασφαλίζουν τις παραπάνω δύο ιδιότητες.

Σε υπάρχοντα δίκτυα όπως το Internet, οι περισσότερες από τις διαδικασίες που αφορούν στην ασφάλεια και στην ασφάλιση του δικτύου, εκτελούνται στα άκρα, όπως π.χ. η ανάλυση ονόματος . Υπηρεσίες όμως μέσα στο δίκτυο, όπως η δρομολόγηση, εισήγαγαν στο δίκτυο τρωτότητα π.χ. αν ένας δρομολογητής σηματοδοτεί low-cost διαδρομές, είτε σκόπιμα είτε από λάθος, τότε θα κατευθυνθεί προς αυτόν μεγάλος όγκος κίνησης με αποτέλεσμα να μη μπορεί να ανταποκριθεί ο εν λόγω δρομολογητής.

Η ελεγχόμενη πρόσβαση στους πόρους είναι η βάση της ασφαλείας του δικτύου και η ασφάλεια των κόμβων. Καθώς τα υπάρχοντα συστήματα έχουν σχεδιασθεί αρκετά παλιότερα με πιθανά ελαττωματικούς κόμβους, η επιδίωξή μας είναι να έχουμε τουλάχιστον μια πλειοψηφία από ασφαλή κόμβους. Ωστόσο η ασφάλεια ολόκληρου του δικτύου είναι ένα πιο πολύπλοκο πρόβλημα από την ασφάλεια ξεχωριστά των κόμβων, αφού ένα δίκτυο είναι ένα ολοκληρωμένο σύστημα από καταναμημένους πόρους.

Ας δούμε το παρακάτω παράδειγμα : η υπηρεσία multicast είναι βασική υπηρεσία ενός δικτύου κι έτσι ένα δίκτυο θα πρέπει να υποστηρίζει μια τέτοια εφαρμογή.. Το απλούστερο πρόγραμμα που υποστηρίζει multicast υπηρεσία είναι κάποιο που στέλνει ένα πακέτο από μια θύρα σε δύο άλλες, δομώντας ένα απλό δυαδικό δέντρο. Σ έναν κόμβο, αυτό το πρόγραμμα θα αναλύοταν και θα αξιολογούνταν. Ας υποθέσουμε ότι ένα αντίγραφο του προγράμματος είναι συνδεδεμένο με κάθε input θύρα σε κάθε κόμβο στο δίκτυο, μια έγκυρη διαμόρφωση (configuration) που χρησιμοποιεί ένα έγκυρο πρόγραμμα. Όταν τα πακέτα εισέρχονται στο δίκτυο, δύο αντίγραφα θα προκύψουν από το δίκτυο και ούτω καθ' εξής και έτσι το το δίκτυο θα καταρρεύσει. Το παραπάνω ήταν ένα παράδειγμα ασφαλούς κόμβου αλλά όχι ασφαλούς δικτύου. Η ιδέα που προκύπτει από το προηγούμενο παράδειγμα είναι ότι τα ασφαλή σε δίκτυο προγράμματα είναι υποσύνολο των ασφαλών σε κόμβο προγραμμάτων τα οποία με τη σειρά τους είναι

υποσύνολα όλων των πιθανών προγραμμάτων. Σε κάθε περίπτωση το πιο σημαντικό θέμα είναι να ανακαλύψουμε τεχνικές που κρατούν το σύστημα στα επιθυμητά υποσύνολα.

Οι πιο εξελιγμένες και περισσότερο υποσχόμενες προσεγγίσεις στο παραπάνω πρόβλημα είναι αυτές που αντιμετωπίζουν την εσωτερική δομή ενός προγραμματιζόμενου δικτύου σαν ένα κατανεμημένο σύστημα. Μ' αυτόν τον τρόπο οι τοπικές ενέργειες μπορούν να συνδυαστούν σε μια καθολική συμπεριφορά. Αυτή είναι και η βασική αρχή που χρησιμοποιείται από το TCP/IP για τον έλεγχο της συμφόρησης, όπου τοπικές αντιδράσεις σε συμφόρηση, προστατεύουν το δίκτυο.

Εξετάζοντας τα μοντέρνα δίκτυα, φαίνεται μάλλον απίθανο ότι περιορισμοί του συστήματος βασισμένοι σε δυναμικούς ελέγχους, μπορούν να επιβιώσουν [5]. Φαίνεται περισσότερο πιθανό να επιβιώσουν δομές που θα επιβάλλουν πολιτικές δύο φάσεων, συμπαγή στατική ανάλυση πριν την είσοδο των προγραμμάτων στο δίκτυο συνδυασμένη με έλεγχο ότι αυτή η ανάλυση έχει γίνει. Αυτή η πιθανότητα φαίνεται ότι θα επικρατήσει γιατί οι υπολογισμοί στα άκρα θα είναι φτηνοί και η επαλήθευση ότι ο έλεγχος έχει γίνει είναι φθηνότερος από το να ελέγχει το δίκτυο τον εαυτό του.

Ωστόσο ήδη έχουν αναφερθεί στον πίνακα 1 εναλλακτικοί τρόποι ασφάλισης του δικτύου. Τα πλεονεκτήματα τους είναι ότι μπορούν να διαβαστούν από το χρήστη, είναι απλοί και συντίθενται εύκολα π.χ. Safe Tel, ενώ μειονέκτημα είναι η επιβάρυνση του μεγέθους του προγράμματος.

Όσον αφορά στον εξαρτώμενο από την πλατφόρμα κώδικα, αντί των «βαριών» παραδοσιακών μηχανισμών έχουν αντιπροταθεί τεχνολογίες compiler που θα παράγουν ασφαλή κώδικα.

Συμπεράσματα

Τα προγραμματιζόμενα δίκτυα προσφέρουν νέους τρόπους για να χτίσουμε τα μελλοντικά δίκτυα να επεκτείνουμε και να βελτιώσουμε τις εφαρμογές τους, τη διαχείριση και την ασφάλειά τους. Επίσης θα ωθήσουν το βιομηχανικό κλάδο που ασχολείται με το λογισμικό και hardware των δικτύων σε νέες επιχειρηματικές δράσεις και επενδύσεις. Ήδη πολλές ερευνητικές ομάδες ασχολούνται με τα ενεργά δίκτυα και είναι αναμενόμενο ότι τελικά θα επικρατήσουν έναντι των παραδοσιακών δικτύων.

Αναφορές

[1] Ian Wakeman, Morris Sloman, Doug Shepherd, David Duce, Deborah Miller, Rob Cole Report to the EPSRC on the first UK programmable networks and telecommunications workshop

<http://www.cogs.susx.ac.uk/projects/safetynet/prognet/epsrc.html>

[2] David Tennenhouse, David Wetherall Towards an active network architecture

<http://www.atm.cs.ndsu.nodak.edu/~zzhang/atmseminar/ActiveNet.html>

[3] Aurel Lazar Building open programmable multimedia networks
<http://www.comet.columbia.edu>

[4] Y.Yemini, Sushil Da Silva Towards Active Programmable Networks
<http://www.cs.columbia.edu/~dasilva/content/netscript/pubs/dsom96-slides/dsom96-notes.html>

[5] Scott Alexander, William Arbaugh, Angelos Keromytis, Jonathan Smith
Safety and security of programmable network infrastructures
<http://www.cis.upenn.edu/~switchware/>

[6] David Tennenhouse, Jonathan Smith, David Sincoskie, David Wetherall
A survey of active network research
<http://tns-www.lcs.mit.edu/publications/ieeecomms97.html>

[7] David Wetherall, Ulana Legedza, John Guttag
Introducing new Internet services : why and how
<http://www.sds.lcs.mit.edu/publications/network98.html>

[8] Scott Alexander, Michael Hicks, Pankaj Kakkar, Angelos Keromytis, Marianne Shaw, Jonathan Moore, Carl Gunter, Trevor Jim, Scott Nettles, Jonathan Smith
The switch ware active network implementation
<http://www.cis.upenn.edu/~switchware/>

[9] Aurel Lazar, Andrew Campbell , Spawning Networking Architectures
(white papers)