ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ

ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Λειτουργία αναπαράστασης (Virtualization) στοίβας πρωτοκόλλων μελλοντικού διαδικτύου

Διπλωματική εργασία

Σταράς Γεώργιος

Ξάνθη, 2020

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Επιβλέπων Καθηγητής: Κος Τσαουσίδης Βασίλειος

ΕΥΧΑΡΙΣΤΙΕΣ

Επιθυμώ να εκφράσω τις ευχαριστίες μου σε όλους εκείνους που συνέβαλλαν άμεσα ή έμμεσα στην ολοκλήρωση της διπλωματικής μου εργασίας και κατά συνέπεια των προπτυχιακών σπουδών μου.

Θα ήθελα να ευχαριστήσω ιδιαίτερα τον κύριο Τσαουσίδη Βασίλειο ο οποίος υπήρξε ο επιβλέπων της παρούσας εργασίας και τον βοηθό του κύριο Σάρρο Αλέξανδρο. Η υποστήριξη και διαθεσιμότητά τους καθ' όλη τη διάρκεια εκπόνησης της εργασίας, όπως και οι ουσιαστικές παρατηρήσεις και προτάσεις τους, βοήθησαν σημαντικά στην ολοκλήρωση της.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Περίληψη

Η τρέχουσα αρχιτεκτονική του διαδικτύου, που βασίζεται στις Internet Protocol (IP) διευθύνσεις, επικεντρώνεται στη δημιουργία συνομιλίας μεταξύ δύο τερματικών. Αυτό καθίσταται εμφανές από το σύστημα ονοματολογίας, στο οποίο οι διευθύνσεις μέσω του Domain Name System (DNS) υποδεικνύουν έναν κεντρικό υπολογιστή για την επίτευξη επικοινωνίας, η οποία έχει ως τελικό σκοπό την εκτέλεση μίας ενέργειας ή τη λήψη δεδομένων.

Η εν λόγω αρχιτεκτονική ωστόσο δεν ανταποκρίνεται επαρκώς στις ανάγκες της σημερινής εποχής. Χρησιμοποιείται ευρέως για τη διανομή περιεχομένου τόσο σε σταθερούς υπολογιστές όσο και σε κινητές συσκευές, παρόλο που έχει σχεδιαστεί για επικοινωνία τελικών σημείων. Αυτό σε συνδυασμό με αδυναμίες που παρουσιάζονται στην ασφάλεια, στη διαχείριση διευθύνσεων και σε άλλους τομείς, οδήγησαν στην ανάγκη για έρευνα και ανάπτυξη εναλλακτικών αρχιτεκτονικών.

Τέτοιες υποσχόμενες αρχιτεκτονικές αποτελούν τα Named Data Network (NDN), Delay Tolerant Networking (DTN) και Umobile, τα οποία βασίζονται στα επιτεύγματα του σημερινού διαδικτύου, αντικατοπτρίζοντας την κατανόηση των δυνατοτήτων και των περιορισμών της αρχιτεκτονικής IP.

Στην παρούσα πτυχιακή, γίνεται ανάλυση των προαναφερθέντων αρχιτεκτονικών σε βάθος και παρουσιάζονται βήμα προς βήμα όλες οι δοκιμές που γίνανε, χρησιμοποιώντας τες σε τεχνικό επίπεδο.

Εισαγωγή

Καθώς η τεχνολογία εξελίσσεται, νέες συσκευές και εφαρμογές έρχονται στο προσκήνιο ως διαφορετικά συστατικά ενός γιγάντιου παζλ, τα οποία πρέπει να υποστηρίζονται από την αρχιτεκτονική του διαδικτύου. Το διαδίκτυο έχει αντιμετωπίσει και με το παραπάνω όλες τις προκλήσεις και ανάγκες που έχουν προκύψει.

Παρόλα αυτά, περιορισμοί που οφείλονται στην αρχιτεκτονική του σημερινού διαδικτύου καθώς και στην όλο και αυξανόμενη χρήση κινητών συσκευών και της τεχνολογίας Internet of Things (IoT), έχουν οδηγήσει στην ανάγκη εξεύρεσης νέων τεχνολογιών του διαδικτύου. Προκειμένου να αντιμετωπιστούν αυτοί οι περιορισμοί, εκτενής χρόνος και έρευνα έχουν επενδυθεί.

Αποτέλεσμα της έρευνας είναι η ανάπτυξη διάφορων αρχιτεκτονικών προσεγγίσεων. Τρεις ενδιαφέρουσες αρχιτεκτονικές, οι οποίες αποτελούν αντικείμενο της παρούσας εργασίας, είναι οι εξής:

- Named Data Networking (NDN)
- Delay Tolerant Networking (DTN)
- Umobile

Όσον αφορά την αρχιτεκτονική NDN, στόχος της είναι η δημιουργία ενός δικτύου που θα είναι προσανατολισμένο στην πληροφορία. Συγκεκριμένα, σε περίπτωση αιτήματος του χρήστη για κάποια πληροφορία, δεν θα χρησιμοποιείται η διεύθυνση στην οποία ενδεχομένως βρίσκεται αυτή, αλλά θα γίνεται αίτημα στο ίδιο το περιεχόμενο χρησιμοποιώντας μία περιγραφή.

Το Delay Tolerant Networking (DTN) είναι μια προσέγγιση στην αρχιτεκτονική δικτύου υπολογιστών, που επιδιώκει να αντιμετωπίσει τα τεχνικά ζητήματα σε ετερογενή δίκτυα, τα οποία ενδέχεται να μην έχουν συνεχή συνδεσιμότητα στο δίκτυο.

Τέλος, το umobile αποτελεί μία mobile-centric και service-oriented αρχιτεκτονική, κύριος στόχος της οποίας είναι η αποτελεσματική παροχή περιεχομένου και υπηρεσιών στους τελικούς χρήστες ακόμα και σε περιοχές που χαρακτηρίζονται ως δύσκολα προσβάσιμες.

Σκοπός της παρούσας πτυχιακής εργασίας είναι η αναπαράσταση ενός δικτύου χρησιμοποιώντας τις παραπάνω αρχιτεκτονικές, στις οποίες εικονικοί υπολογιστές που έχουν υλοποιηθεί με την τεχνολογία των Docker containers μπορούν να συνδεθούν και να επικοινωνήσουν μεταξύ τους. Συγκεκριμένα μία τοπολογία από Docker containers δημιουργήθηκε, τα οποία τρέχουν Ubuntu OS και στα οποία έχουν εγκατασταθεί βιβλιοθήκες και εργαλεία απαραίτητα για την επίτευξη της επικοινωνίας, χρησιμοποιώντας τα προαναφερθέντα.

Η παρούσα εργασία χωρίζεται σε θεωρητική και προγραμματιστική, με το θεωρητικό μέρος να αναλύεται στις ενότητες 1 έως 3 και το προγραμματιστικό κομμάτι στην ενότητα 4. Για το θεωρητικό κομμάτι, μελετήθηκαν επιστημονικά άρθρα ξένων και Ελλήνων επιστημόνων καθώς και επιστημονικές και τεχνικές ιστοσελίδες.

Με βάση τα παραπάνω, η δομή της παρούσας εργασίας διαμορφώνεται ως εξής:

Στην πρώτη ενότητα γίνεται ανάλυση της εικονικοποίησης (virtualization) και περιγράφεται η μετάβαση από την δημιουργία εικονικών μηχανών (που τρέχουν ολόκληρο το λειτουργικό σύστημα) στη δημιουργία containers (που χρησιμοποιούν μόνο τις απαραίτητες για αυτά διεργασίες). Επιπλέον παρουσιάζεται η υλοποίηση των containers με την χρήση της τεχνολογίας Docker.

Στόχος της δεύτερης ενότητας είναι η παρουσίαση της τρέχουσας υποδομής του διαδικτύου καθώς και των πρωτοκόλλων επικοινωνίας και συστημάτων που το απαρτίζουν. Πιο συγκεκριμένα, αναφέρονται και αναλύονται πρωτόκολλα όπως το TCP/IP, DHCP και HTTP καθώς και το σύστημα DNS.

Η τρίτη ενότητα αφορά την αρχιτεκτονική NDN. Αναλύει τη δομή και τον τρόπο λειτουργίας της, ενώ παράλληλα κάνει αναφορά στα οφέλη που επέρχονται από τη χρησιμοποίηση της καθώς και στις διαφορές της με την τρέχουσα αρχιτεκτονική του διαδικτύου.

Η τέταρτη και τελευταία ενότητα περιγράφει το προγραμματιστικό σκέλος της πτυχιακής εργασίας. Πιο συγκεκριμένα, αναλύεται η υποδομή, τα εργαλεία και οι εντολές που χρησιμοποιήθηκαν για την επίτευξη επικοινωνίας υπολογιστών μέσω της NDN αρχιτεκτονικής.

Ενότητα 1: Περιγραφή Τεχνολογιών Εικονικοποίησης

1.1 Εικονικοποίηση (Virtualization)

Μία εικονική μηχανή (virtual machine - VM) είναι ένα ολόκληρο λειτουργικό σύστημα που τρέχει μέσα σε ένα λειτουργικό σύστημα κεντρικού υπολογιστή. Οι εικονικές μηχανές γνωστές ως και «επισκέπτες» (guests), μοιράζονται και χρησιμοποιούν τους πόρους του κεντρικού υπολογιστή (host), που τις φιλοξενεί και είναι απομονωμένες η μία από την άλλη.

Ο όρος «εικονικοποίηση» αναφέρεται στην δυνατότητα ταυτόχρονης εκτέλεσης πολλών τέτοιων εικονικών μηχανών σε μία φυσική μηχανή. Ουσιαστικά, η εικονική μηχανή αποτελεί λογισμικό προσομοίωσης ενός φυσικού συστήματος.

Η δημιουργία, διαχείριση και ασφαλή εκτέλεση των εικονικών μηχανών εκτελείται από ένα λογισμικό που ονομάζεται «ελεγκτής εικονικών μηχανών», γνωστό ως και «Hypervisor» (Virtual Machine Monitor). Ο Hypervisor δημιουργεί την ψευδαίσθηση στις εικονικές μηχανές ότι τρέχουν στο bare metal, ενώ παράλληλα ελέγχει την πρόσβαση στους κάτωθι πόρους του συστήματος:

- Μετάφραση διευθύνσεων
- Είσοδο / Έξοδο (I/O)
- Διακοπές
- Μεταφορά σε προνομιούχο κατάσταση

Ένα από τα μεγάλα πλεονεκτήματα του virtualization είναι η δυνατότητα δημιουργίας στιγμιότυπων. Ως στιγμιότυπο νοείται η κατάσταση μιας εικονικής μηχανής σε ένα ακριβές χρονικό σημείο. Οι εικονικές μηχανές μπορούν να μεταφερθούν από έναν υπολογιστή σε έναν άλλον και να συνεχίσουν την κατάσταση λειτουργίας τους χωρίς ιδιαίτερες ρυθμίσεις. Αυτό οφείλεται στο γεγονός, ότι η εικονική μηχανή στη πραγματικότητά είναι αρχεία τα οποία ονομάζονται εικόνες (images).

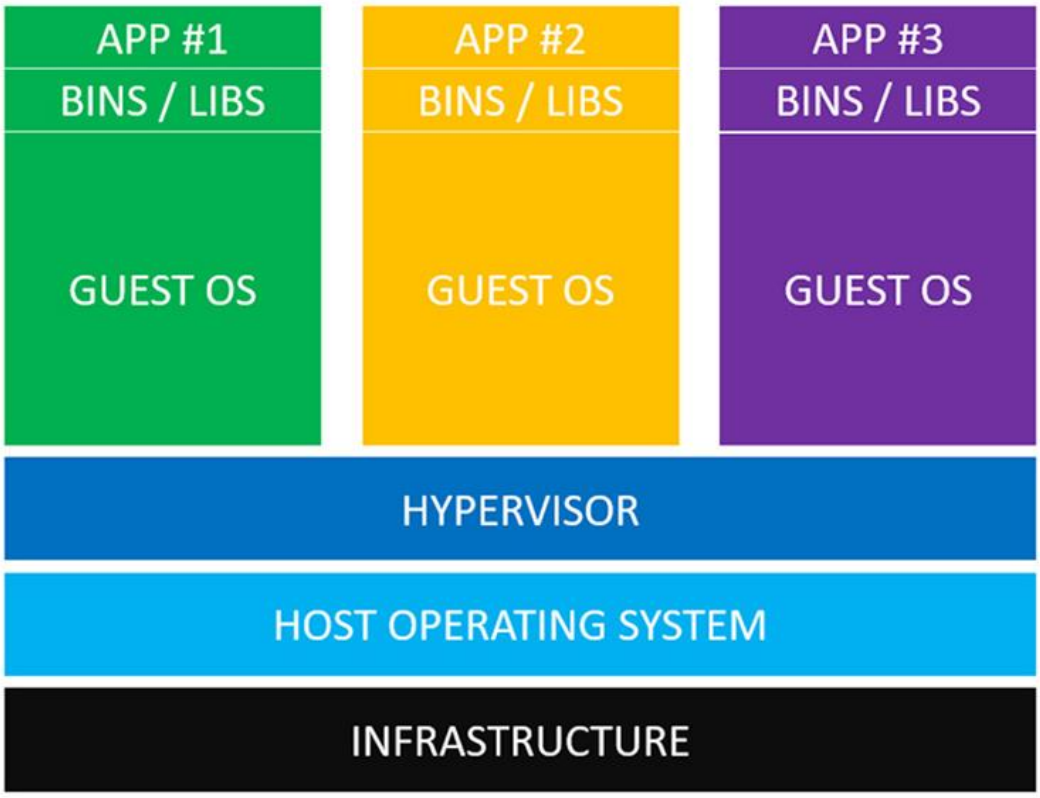
Το virtualization παρέχει πολλαπλά οφέλη:

- **μείωση κόστους και ενέργειας**

  Στα data centers [1], υπάρχουν όλο και λιγότεροι φυσικοί υπολογιστές (servers) οι οποίοι φιλοξενούν πλήθος εικονικών μηχανών. Αυτό έχει ως αποτέλεσμα, να υπάρχει μικρότερη υποδομή άρα λιγότερες ανάγκες για ενέργειες λειτουργίας και ψύξης.
- **απομόνωση και προστασία μεταξύ των εικονικών μηχανών που τρέχουν στον ίδιο Host**

  Οι εικονικές μηχανές δεν γνωρίζουν η μία την ύπαρξη της άλλης και τρέχουν σε απομονωμένα περιβάλλοντα, ενώ διαμοιράζονται τους πόρους του κεντρικού υπολογιστή
- **εύκολη, γρήγορη ανάπτυξη και δοκιμή εφαρμογών σε διάφορα λειτουργικά συστήματα**

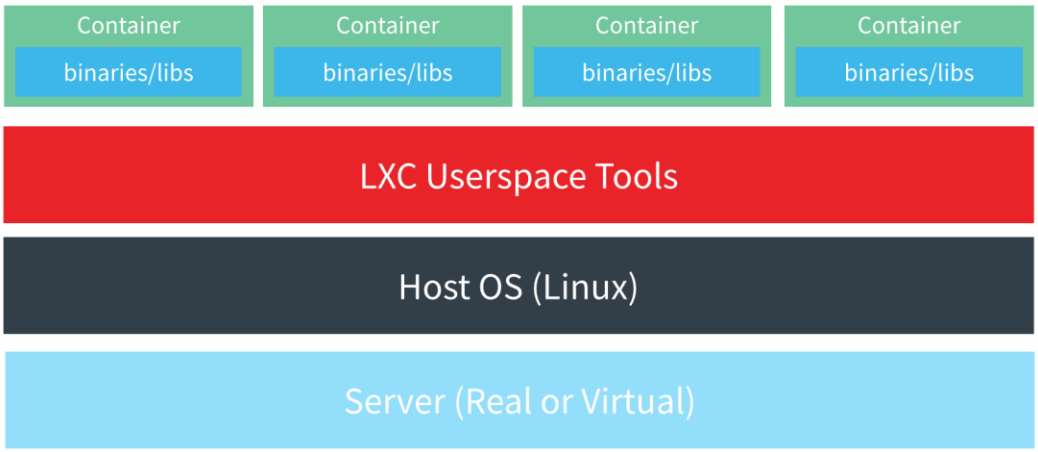
  Ο προγραμματιστής που χρησιμοποιεί λειτουργικό «Macintosh» [2] με τη χρήση μιας εικονικής μηχανής που τρέχει το λειτουργικό «Windows» [3], μπορεί να δοκιμάσει την ιστοσελίδα του στον browser της Microsoft [4] για θέματα συμβατότητας
- **Η κλιμακωμένη χρήση των εικονικών μηχανών αποτελεί συστατικό στοιχείο του cloud computing** [5]



1.2 Linux containers

Η όλο και αυξανόμενη χρήση των εικονικών μηχανών αποτέλεσε το έναυσμα για να μελετηθεί και ερευνηθεί ο τομέας του virtualization εκτενέστερα. Αυτό είχε ως αποτέλεσμα την δημιουργία της «lightweight process virtualization» (ελαφριά εικονικοποίηση διεργασιών). Συγκεκριμένα στο εν λόγω virtualization, τα περιττά και επαναλαμβανόμενα στοιχεία του εικονικού συστήματος έχουν αφαιρεθεί και έχει δημιουργηθεί μια ελαχιστοποιημένη έκδοσή του. Αυτό είχε ως αποτέλεσμα τη δημιουργία των Linux Containers (LXC).

Ένας Linux Container είναι ένα περιβάλλον virtualization (στο επίπεδο του λειτουργικού συστήματος) στο οποίο τρέχουν πολλαπλά απομονωμένα Linux [6] συστήματα, χρησιμοποιώντας ένα μοναδικό Linux kernel [7]. Δεν δημιουργεί μια ολόκληρη εικονική μηχανή, έχει ελάχιστες απαιτήσεις πόρων και εκκινεί πολύ γρήγορα.

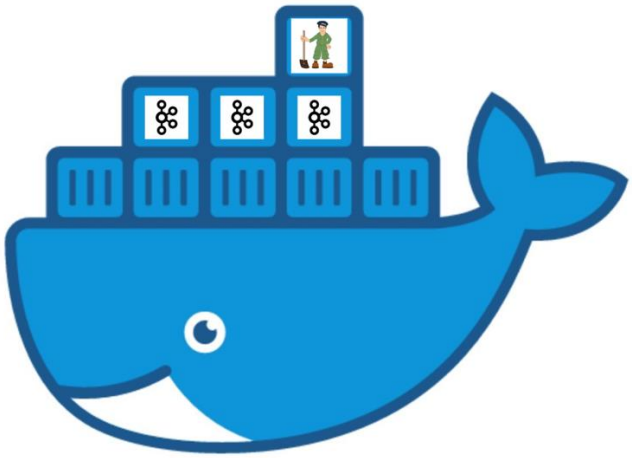


1.3 Docker

Το Docker είναι open source [8] λογισμικό, το οποίο υλοποιεί virtualization σε επίπεδο λειτουργικού συστήματος. Προσφέρει αυτοματοποιημένες διαδικασίες για την ανάπτυξη εφαρμογών σε απομονωμένες περιοχές χρήστη (User Spaces), που ονομάζονται Software Containers. Το συγκεκριμένο λογισμικό χρησιμοποιεί τεχνολογίες του πυρήνα του Linux όπως τα cgroups [9] και τα kernel namespaces [10], προκειμένου να επιτρέπει σε ανεξάρτητα software

containers να εκτελούνται στο ίδιο λειτουργικό σύστημα. Έτσι αποφεύγεται η χρήση επιπλέον υπολογιστικών πόρων που θα απαιτούσε μια εικονική μηχανή.

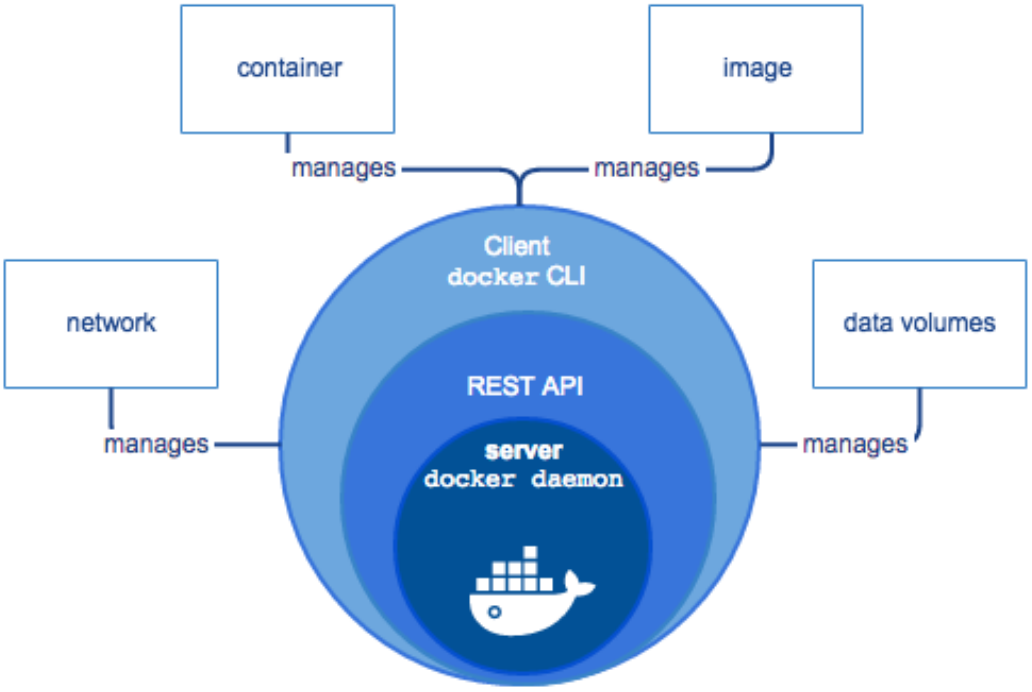
Το Docker δημιουργήθηκε από την εταιρία Docker Inc. [11], η οποία ιδρύθηκε από τους Solomon Hykes [12] και Sebastien Pahl [13]. Κυκλοφόρησε για πρώτη φορά το 2013 και μπορεί να τρέξει σε Windows, Linux και Macintosh.



1.3.1 Docker Engine

Είναι μια τεχνολογία για χτίσιμο και «containerizing» εφαρμογών. Ενεργεί ως μία client-server [14] εφαρμογή με τα παρακάτω δομικά στοιχεία:

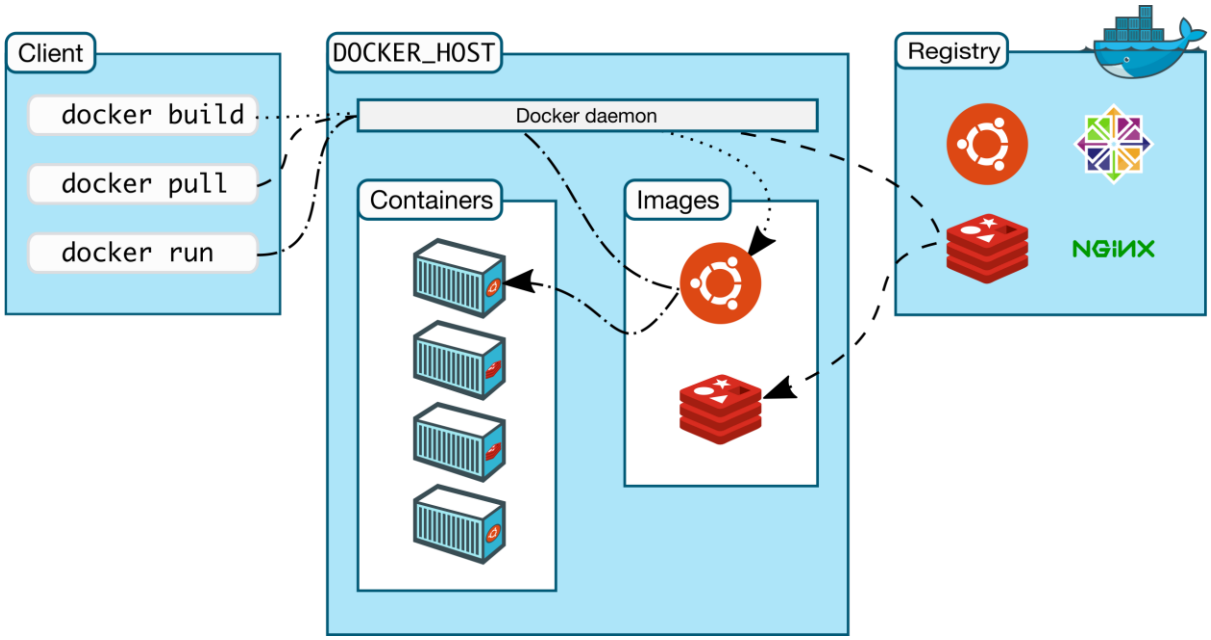
- Ένα server με μια μακροχρόνια διαδικασία daemon [15] (daemon process)
- Ένα REST [16] API που καθορίζει τις διεπαφές που μπορούν να χρησιμοποιήσουν τα προγράμματα για να μιλήσουν και να δώσουν εντολή στον daemon
- Μια διεπαφή γραμμής εντολών (CLI) [17]. Το CLI χρησιμοποιεί το REST API για να ελέγξει ή να αλληλοεπιδράσει με το daemon χρησιμοποιώντας scripts ή απευθείας με εντολές. Ο daemon δημιουργεί και διαχειρίζεται Docker αντικείμενα, όπως εικόνες και containers.



1.3.2 Αρχιτεκτονική

Το Docker εκμεταλλεύεται τη δυνατότητα του Linux να επιτυγχάνει την απομόνωση των πόρων σε ένα εικονικό περιβάλλον στο οποίο μπορεί να τρέξει μια εφαρμογή. Δεν περιορίζεται μόνο στη δημιουργία ενός απομονωμένου συστήματος αρχείων αλλά συμπεριλαμβάνει και τους άλλους υπολογιστικούς πόρους όπως αυτού του δικτύου, του επεξεργαστή και της μνήμης.

Η αρχιτεκτονική του Docker συνοψίζεται στην παρακάτω εικόνα:



Docker Client

Αποτελεί τον κύριο τρόπο αλληλεπίδρασης των χρηστών με το Docker. Όταν χρησιμοποιούνται εντολές (π.χ. docker run), ο client στέλνει τις εντολές στον daemon, ο οποίος και τις εκτελεί. Η εντολή docker χρησιμοποιεί το API που παρέχεται από το Docker. Ο client μπορεί να επικοινωνεί με παραπάνω από ένα daemon.

Docker Daemon

Ο daemon χειρίζεται αιτήματα που αφορούν το Docker API και Docker αντικείμενα όπως εικόνες, containers και δίκτυα. Μπορεί επίσης να επικοινωνεί με άλλους daemon με σκοπό να διαχειριστεί άλλες Docker υπηρεσίες.

Docker Image

Αποτελεί ένα υπόδειγμα με οδηγίες για τη δημιουργία Docker containers. Συχνά μία εικόνα βασίζεται σε άλλη εικόνα με κάποιες επιπλέον αλλαγές και ρυθμίσεις. Για παράδειγμα, μπορεί

να δημιουργηθεί μία εικόνα βασισμένη στην εικόνα της έκδοσης του Linux Ubuntu [18] , το οποίο εγκαθιστά επίσης Java [19] και την επιθυμητή βάση δεδομένων [20]. Ο κάθε χρήστης μπορεί να δημιουργήσει τις δικές του εικόνες με τη χρήση ενός DockerFile ή μπορεί να χρησιμοποιήσει εικόνες τρίτων από κάποιο Docker registry.

Containers

Το container αποτελεί το τρέχον στιγμιότυπο μιας εικόνας και είναι απομονωμένο από άλλα containers και τον host. Ένα container ορίζεται από την εικόνα του καθώς και από τις ρυθμίσεις που παρέχονται κατά τη δημιουργία του. Επιπλέον τα container μοιράζονται τον πυρήνα με το host λειτουργικό σύστημα.

Ένας νέος container μπορεί να δημιουργηθεί από μία εικόνα με την εντολή «docker run -options image_name -command».

Docker Registry

Ένα Docker registry περιέχει εικόνες Docker. Το Docker hub [21] είναι ένα ανοιχτό registry προσβάσιμο από όλους. Χρησιμοποιώντας τις εντολές «docker pull» ή «docker run», οι απαιτούμενες εικόνες αντλούνται από το registry. Με την εντολή «docker push», η εικόνα που έχει δημιουργηθεί, ανεβαίνει στο συγκεκριμένο registry.

Docker Service

Οι υπηρεσίες επιτρέπουν το scale των containers σε πολλαπλούς daemons, οι οποίοι λειτουργούν ως ένα σμήνος (swarm) με πολλαπλούς managers και workers. Κάθε μέλος του σμήνους είναι ένας daemon και όλοι οι daemons επικοινωνούν χρησιμοποιώντας το Docker API. Η υπηρεσία επιτρέπει να οριστεί η επιθυμητή κατάσταση, όπως ο αριθμός των αντιγράφων της υπηρεσίας, τα οποία πρέπει να είναι διαθέσιμα ανά πάσα στιγμή. Επίσης γίνεται εξισορρόπηση φόρτου εργασίας ανάμεσα σε όλους τους workers γνωστό ως και «load balancing». Στον client, η Docker υπηρεσία εμφανίζεται ως μία εφαρμογή.

Docker compose

Είναι ένα εργαλείο με το οποίο μπορούμε να ορίσουμε και να τρέξουμε εφαρμογές που χρησιμοποιούν πολλαπλά containers. Χρησιμοποιώντας ένα αρχείο YAML [22], μπορούμε να ρυθμίσουμε τις υπηρεσίες της εφαρμογής, και μέσω μια εντολής, όλες τις υπηρεσίες που είχαν ρυθμιστεί, δημιουργούνται και τρέχουν.

Ένα παράδειγμα τέτοιου αρχείου YAML είναι το εξής:

```
version: '3'
services:
  web:
    build: .
    ports:
    - "5000:5000"
    volumes:
    - .:/code
    - logvolume01:/var/log
    links:
    - redis
  redis:
    image: redis
volumes:
  logvolume01: {}
```

1.3.3 Χώροι ονομάτων (namespaces)

Το Docker χρησιμοποιεί μια τεχνολογία γνωστή ως «χώροι ονομάτων» για τη παροχή ενός απομονωμένου περιβάλλοντος που ονομάζεται container. Όταν το container τρέχει, το Docker δημιουργεί ένα σύνολο από namespaces για αυτό και παρέχουν ένα στρώμα απομόνωσης. Κάθε πτυχή ενός container τρέχει σε ξεχωριστό χώρο ονομάτων και η πρόσβασή του, περιορίζεται σε αυτό το χώρο ονομάτων.

Το Docker engine χρησιμοποιεί Linux name spaces, όπως τα ακόλουθα:

- pid: Απομόνωση διεργασιών (PID: Process ID)
- net: Διαχείριση διεπαφών δικτύου (NET: Networking)
- ipc: Διαχείριση πρόσβασης σε IPC πόρους (IPC: InterProcess Communication)
- mnt: Διαχείριση σημείων συστήματος αρχείου (MNT: Mount)
- uts: Απομόνωση πυρήνα και αναγνωριστικά έκδοσης (UTS: Unix Timesharing System)

1.3.4 Ομάδες ελέγχου (control groups)

Το Docker engine στο Linux βασίζεται επίσης σε μια άλλη τεχνολογία που ονομάζεται «ομάδες ελέγχου» (control groups). Ο έλεγχος ομάδων επιτρέπει στο Docker engine να διαμοιράζει τους διαθέσιμους πόρους υλικού σε containers και να επιβάλει όρια και περιορισμούς.

Για παράδειγμα μπορεί να επιβληθεί όριο στη διαθέσιμη μνήμη ενός συγκεκριμένου container.

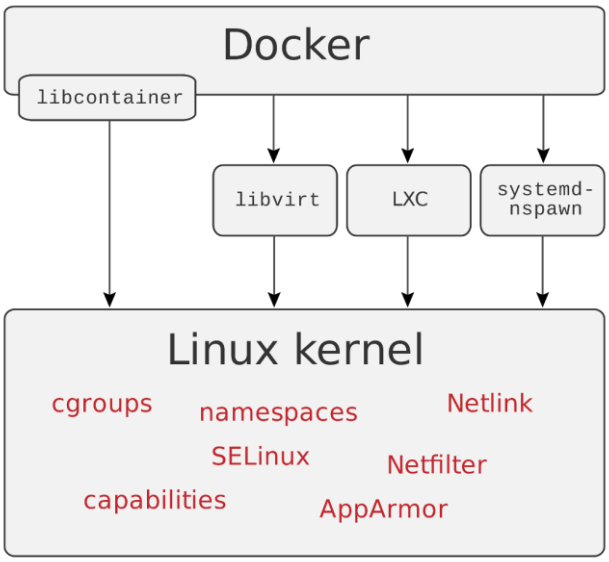
1.3.5 Union συστήματα αρχείων

Union συστήματα αρχείων, γνωστά και ως UnionFS, είναι συστήματα αρχείων που λειτουργούν με τη δημιουργία στρώσεων, καθιστώντας τα πολύ ελαφριά και γρήγορα. Το Docker engine χρησιμοποιεί το UnionFS για να παρέχει τα δομικά στοιχεία για τα containers.

Διάφορες εκδοχές του UnionFS είναι το AUFS [23], btrfs [24], vfs [25] και DeviceMapper [26].

1.3.6 Μορφή Container

Το Docker engine συνδυάζει τους χώρους ονομάτων, τις ομάδες ελέγχου και τα Union συστήματα αρχείων σε μια συσκευασία που λέγεται «μορφή container».



1.3.7 Συστήματα διαχείρισης containers

Για τη διαχείριση των Docker containers υπάρχουν διάφορα συστήματα διαχείρισης. Τα πιο σημαντικά είναι τα εξής:

- Borg
- Omega
- Kubernetes

Borg

Αποτελεί το πρώτο ενοποιημένο σύστημα διαχείρισης containers και δημιουργήθηκε από την Google [27]. Είναι ένα ευρύ οικοσύστημα από εργαλεία και υπηρεσίες. Χαρακτηριστικό είναι ότι όλο και περισσότερες εφαρμογές δημιουργούνταν για να τρέξουν πάνω σε αυτό. Αποτελεί μέχρι και σήμερα, το σύστημα διαχείρισης που χρησιμοποιεί η Google εσωτερικά λόγω των πολλαπλών λειτουργιών που διαθέτει καθώς και της ανθεκτικότητας και στιβαρότητας του συστήματος.

Omega

Απόγονος του Borg που φτιάχτηκε λόγω της επιθυμίας για βελτίωση της μηχανικής του προαναφερθέντος οικοσυστήματος. Πολλές από τις καινοτομίες του Omega έχουν υιοθετηθεί από το Borg.

Kubernetes

Δημιουργήθηκε λόγω του ενδιαφέροντος που άρχισαν να δείχνουν όλο και περισσότεροι εξωτερικοί προγραμματιστές σε Linux containers καθώς και στη Google που ήθελε να επενδύσει σε υποδομές στο cloud. Αποτελεί open source εργαλείο, σε αντίθεση με τα προηγούμενα δύο, και δημιουργήθηκε με σκοπό να παρέχει ευκολία στην ανάπτυξη και στη διαχείριση πολύπλοκων κατανεμημένων συστημάτων.

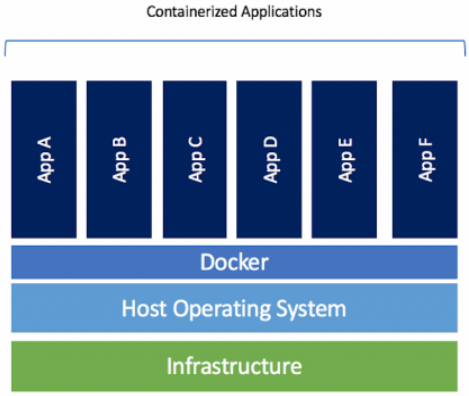
1.3.8 Πλεονεκτήματα του Docker

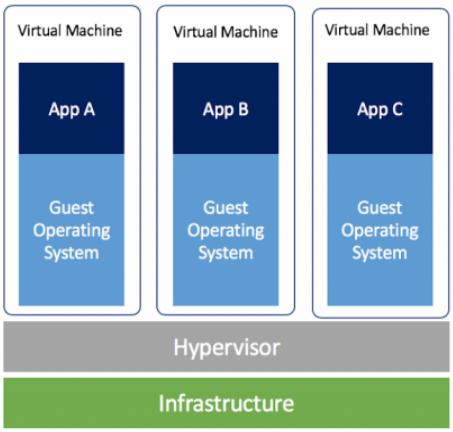
Το Docker είναι πολύ δημοφιλές και χρησιμοποιείται όλο και περισσότερο σε σχέση με τις εικονικές μηχανές λόγω των εξής πλεονεκτημάτων:

- **Ευκολία στη χρήση**: Είναι γενικά απλό και εύκολο στην εκμάθηση και στη μετέπειτα χρήση
- **Μεγαλύτερη απόδοση**: Τα containers δεν χρησιμοποιούν ένα πλήρες λειτουργικό σύστημα, οπότε οι απαιτήσεις σε πόρους είναι λιγότερες σε σχέση με τις εικονικές μηχανές. Δεν υπάρχει κάποιος hypervisor, και μπορούν να αξιοποιήσουν αποτελεσματικότερα τους διαθέσιμους πόρους.
- **Scaling**: Το scaling των containers είναι απίστευτα γρήγορο, κάτι που τα καθιστά ιδανικά για εφαρμογές που έχουν υλοποιηθεί για cloud.
- **Φορητότητα**: Τα containers είναι φορητά και μπορούν να τρέξουν σχεδόν παντού χωρίς να χρειαστεί να γίνουν ιδιαίτερες αλλαγές στα containers κατά τη μεταφορά
- **Η δημιουργία νέων containers είναι πολύ γρήγορη**. Έτσι μειώνεται ο χρόνος που απαιτείται για την εγκατάσταση της υποδομής στην οποία θα αναπτυχθούν και δοκιμαστούν οι διάφορες εφαρμογές.

1.4 Docker Containers vs Virtual Machines

Μία από τις σημαντικές διαφορές είναι ότι το Docker δεν χρησιμοποιεί κάποιον hypervisor, κάτι που μειώνει σημαντικά το overhead. Στο Docker επίσης τα containers μοιράζονται το λειτουργικό σύστημα καθώς και άλλους πόρους, εν αντίθεση με τις εικονικές μηχανές που κάθε μία έχει ένα πλήρες αντίγραφο του λειτουργικού συστήματος και των βιβλιοθηκών που χρησιμοποιούν. Αυτό έχει ως αποτέλεσμα τα containers να έχουν σημαντικά πιο μικρό μέγεθος σε σχέση με τις εικονικές μηχανές και να ξεκινάνε πολύ πιο γρήγορα.





	Start Time	Stop Time
Docker Containers	<50ms	<50ms
VMs	30-45 seconds	5-10 seconds

Ενότητα 2: Τρέχουσα Υποδομή Διαδικτύου

2.1 Πρωτόκολλα Επικοινωνίας

Το πρωτόκολλο αποτελεί ένα σύνολο από συμβάσεις που καθορίζουν το πως ανταλλάσσουν μεταξύ τους δεδομένα οι υπολογιστές του δικτύου. Είναι αυτό που καθορίζει το πως διακινούνται τα δεδομένα και το πως γίνεται ο έλεγχος και ο χειρισμός των λαθών.

2.1.1 TCP/IP

Το διαδίκτυο είναι ένα απέραντο δίκτυο αποτελούμενο από πολλά υπό-δίκτυα. Για αυτό το λόγο απαιτείται πλήθος συμβάσεων που καθορίζουν τον τρόπο ανταλλαγής δεδομένων μεταξύ υπολογιστών διαφορετικού τύπου και διαφορετικού δικτύου.

Οι συμβάσεις αυτές παρέχονται από το TCP/IP. Όλοι οι υπολογιστές, που είναι συνδεδεμένοι τρέχουν το πρωτόκολλο TCP/IP και έτσι μιλούν την ίδια γλώσσα ώστε να μπορούν να καταλάβουν ο ένας τον άλλον παρά τις διαφορές τους. Τα δύο αυτά πρωτόκολλα (που θα παρουσιαστούν λεπτομερώς σε παρακάτω ενότητες) χρησιμοποιούνται σε συνδυασμό για την επίτευξη της ανταλλαγής δεδομένων.

Το διαδίκτυο χρησιμοποιεί μεταγωγή πακέτων για την μεταφορά των δεδομένων. Τα δεδομένα σπάνε σε κομμάτια γνωστά και ως πακέτα και στην κεφαλίδα του κάθε πακέτου εισάγονται οι διευθύνσεις του αποστολέα και του παραλήπτη. Κάθε υπολογιστής που συνδέεται στο διαδίκτυο αποκτά μια IP διεύθυνση.

Το IP πρωτόκολλο είναι υπεύθυνο για την μετάβαση του πακέτου μέσω των διάφορων υπό-δικτύων ως ότου φτάσει στον παραλήπτη. Δρομολογεί το κάθε πακέτο και προσπαθεί να το παραδώσει χωρίς όμως να παρέχει εγγυήσεις για το αν φτάσει το πακέτο ή αν αλλοιωθεί ή αν δεν φτάσει με τη σωστή σειρά. Αυτές τις εγγυήσεις τις παρέχει το TCP πρωτόκολλο που τρέχει πάνω από το IP και είναι υπεύθυνο για τη διάσπαση των δεδομένων σε πακέτα και την επανασύνδεσή τους στον παραλήπτη.

Το TCP πρωτόκολλο πρέπει να γνωρίζει για κάθε ένα πακέτο σε ποια σύνδεση ανήκει. Η διαδικασία αυτή ονομάζεται «απόπλεξη» (Demultiplexing). Στο TCP/IP υπάρχουν πολλά επίπεδα «απόπλεξης». Οι πληροφορίες που χρειάζονται για να γίνει η απόπλεξη περιέχεται σε μια σειρά κεφαλίδων που περιέχονται στο μήνυμα.

Ως εκ τούτου, το διαδίκτυο μπορεί να οριστεί ως ένα δίκτυο αποτελούμενο από δίκτυα υπολογιστών που επικοινωνούν χρησιμοποιώντας τα πρωτόκολλα TCP/IP.

Όσον αφορά τον χρήστη, το μόνο που χρειάζεται να ξέρει είναι η IP διεύθυνση του παραλήπτη ή όπως θα δούμε παρακάτω ένα «όνομα» που αντιστοιχεί στην IP αυτή.

2.1.2 Internet Protocol (IP)

Το πρωτόκολλο διαδικτύου (IP), αποτελεί την κυριότερη σύμβαση επικοινωνίας για τη μετάδοση πακέτων δεδομένων στο διαδίκτυο. Είναι υπεύθυνο για τη διευθυνσιοδότηση των κόμβων και τη δρομολόγηση των πακέτων δεδομένων μέσω ενός ή περισσότερων δικτύων από έναν υπολογιστή προς έναν τελικό προορισμό.

Ανήκει στο επίπεδο δικτύου (θα αναλυθεί παρακάτω) και καθορίζει τη μορφή που θα έχουν τα πακέτα που στέλνονται και τους μηχανισμούς που χρησιμοποιούνται για την αποστολή των πακέτων. Είναι μια υπηρεσία χωρίς σύνδεση και ανεξάρτητη από την τεχνολογία της υλικής υποδομής του δικτύου.

Κάθε πακέτο, αποτελείται από μια κεφαλίδα και ένα τμήμα δεδομένων. Ανάλογα με την έκδοση του πρωτοκόλλου, η μορφή αυτών των τμημάτων διαφέρει. Στη κεφαλίδα περιέχονται πληροφορίες για τα δεδομένα του πακέτου (metadata) και οι διευθύνσεις αφετηρίας και προορισμού.

Το IP δεν μπορεί να εγγυηθεί για τα πακέτα που προσπαθεί να παραδώσει στον εκάστοτε προορισμό. Απλά κάνει τη βέλτιστη προσπάθεια που μπορεί και άλλα ανώτερα επίπεδα λογισμικού πρωτοκόλλων χρησιμοποιούνται για την διόρθωση τυχόν σφαλμάτων.

Οποιαδήποτε συσκευή συνδέεται σε ένα δίκτυο αποκτά αυτομάτως μία διεύθυνση IP.

Τοπολογία Δικτύου

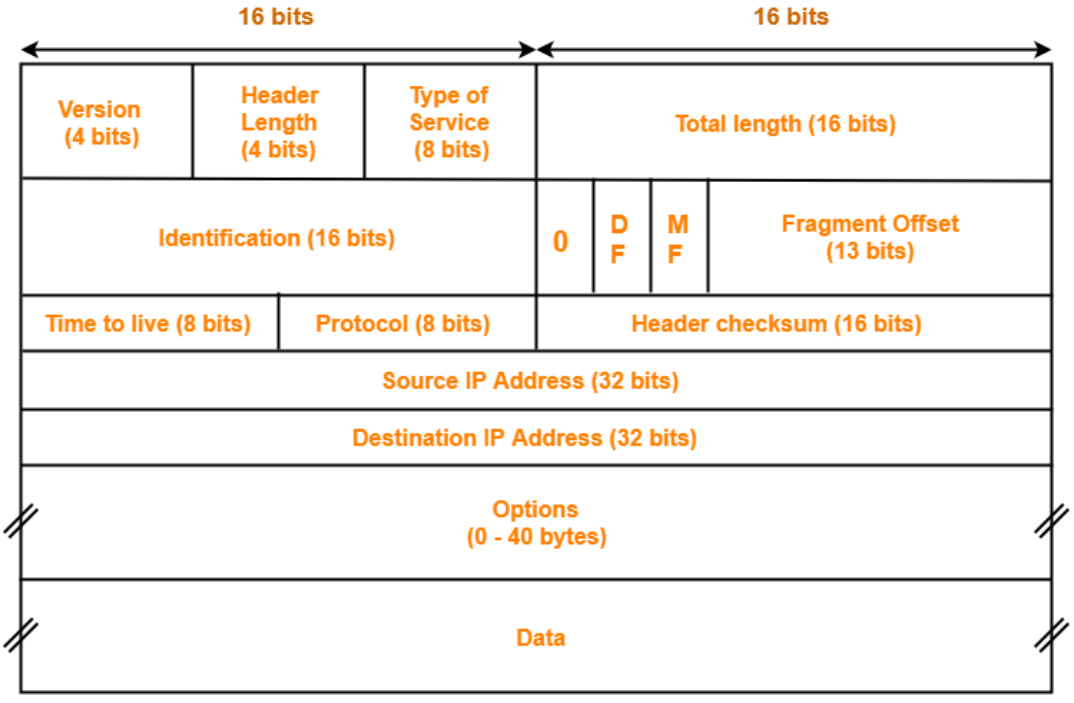
Τα υπό-δίκτυα του διαδικτύου συνδέονται μεταξύ τους με ειδικούς υπολογιστές που λέγονται δρομολογητές (routers) ή πύλες (gateways). Οι δρομολογητές είναι υπεύθυνοι για την δρομολόγηση των πακέτων των δεδομένων μέσα από τα διάφορα δίκτυα ως ότου παραδωθούν στον προορισμό τους.

IPv4

Το IPv4 είναι η τέταρτη έκδοση του πρωτοκόλλου διαδικτύου η οποία εξακολουθεί να χρησιμοποιείται ευρέως. Αποδείχτηκε ένα δυνατό και σταθερό πρωτόκολλο που κάλυπτε τις μέχρι τότε ανάγκες και απαιτήσεις για διευθυνσιοδότηση.

Μορφή πακέτου

Ανάλογα με την έκδοση του πρωτοκόλλου η μορφή του πακέτου διαφέρει. Για την έκδοση 4, η μορφή παρουσιάζεται στην παρακάτω εικόνα:



IPv4 Header

Έκδοση

Είναι το πεδίο της έκδοσης του πρωτοκόλλου, μήκους τεσσάρων bit. Για το IPv4 αυτό έχει την τιμή 4.

Μήκος Επικεφαλίδας (IHL)
Δείχνει το μήκος της επικεφαλίδας σε λέξεις των 32 bit. Επειδή η επικεφαλίδα του IPv4 μπορεί να περιέχει μεταβλητό αριθμό επιλογών, αυτό το πεδίο παρέχει το μήκος της επικεφαλίδας.

Συνολικό Μήκος

Έχει μήκος 16 bit. Καθορίζει το συνολικό μήκος του κομματιού σε bytes (κεφαλίδας και δεδομένων).

Αναγνώριση

Είναι ένα πεδίο ταυτοποίησης και χρησιμοποιείται για τον μοναδικό προσδιορισμό των κομματιών που ανήκουν στο ίδιο αρχικό αυτοδύναμο πακέτο.

Σηματοδότες/Σήματα/Δείκτες

Είναι ένα πεδίο τριών bits που χρησιμοποιείται στον έλεγχο και προσδιορισμό των κομματιών.

Δείκτης εντοπισμού τμήματος

Έχει μήκος 13 bits και προσδιορίζει την θέση ενός συγκεκριμένου κομματιού, από την αρχή του αρχικού μη διασπασμένου αυτοδύναμου πακέτου.

Χρόνος Ζωής

Οριοθετεί το χρόνο ζωής του αυτοδύναμου πακέτου. Έχει μήκος 8 bits και χρησιμοποιείται για την καταστροφή των αυτοδύναμων πακέτων που για διάφορους λόγους περιφέρονται άσκοπα στο Διαδίκτυο.

Αριθμός πρωτοκόλλου

Προσδιορίζει την έκδοση του πρωτοκόλλου που χρησιμοποιείται από το αυτοδύναμο πακέτο.

Άθροισμα ελέγχου επικεφαλίδας

Έχει μήκος 16 bits και χρησιμοποιείται για τον έλεγχο σφαλμάτων της επικεφαλίδας. Μόλις ένα πακέτο φτάσει σε έναν δρομολογητή, ο δρομολογητής υπολογίζει το άθροισμα ελέγχου της επικεφαλίδας και το συγκρίνει με το πεδίο αθροίσματος ελέγχου της επικεφαλίδας. Εάν δεν είναι ίσα, τότε ο δρομολογητής απορρίπτει το πακέτο.

Διεύθυνση πηγής

Είναι η διεύθυνση IPv4 του αποστολέα του πακέτου.

Διεύθυνση προορισμού

Είναι η διεύθυνση IPv4 του παραλήπτη του πακέτου.

Επιλογές

Το πεδίο Επιλογές δεν χρησιμοποιείται συχνά.

Δεδομένα

Τα δεδομένα του πακέτου.

Προβλήματα IPv4

Με το πέρασμα του χρόνου εμφανίζονταν όλο και περισσότερες απαιτήσεις και προβλήματα που δεν μπορούσε να καλύψει η τρέχουσα υλοποίηση. Ορισμένα από τα προβλήματα ήταν:

- **Μικρός αριθμός διευθύνσεων**: Το IPv4 χρησιμοποιεί αριθμούς 32-bit για τις IP διευθύνσεις, που σημαίνει ότι υπάρχουν 4.3 δισεκατομμύρια πιθανές IP διευθύνσεις. Όμως με την χρήση όλων και περισσότερων κινητών συσκευών και την γιγάντωση του Internet of Things (IoT) [27] υπήρξε ανάγκη για περισσότερες διευθύνσεις
- **Μη αποδοτική δρομολόγηση** λόγω αυξανόμενου μεγέθους των δικτύων
- **Ασφάλεια**: δεν διαθέτει ένα ενσωματωμένο σύστημα ασφάλειας. Το IPsec [28] δεν περιέχεται ενσωματωμένο στο IPv4

IPv6

Το IPv6 αποτελεί την έκτη έκδοση του πρωτόκολλου του διαδικτύου και είναι η πιο πρόσφατη αναθεώρηση του πρωτοκόλλου. Είναι το πιο ευρέως διαδεδομένο πρωτόκολλο διαδικτύου μετά το IPv4 και επιλύει τα μεγαλύτερα προβλήματα του IPv4, το οποίο πρόκειται και να αντικαταστήσει.

Το κυριότερο πλεονέκτημα του IPv6, έναντι του IPv4 είναι ο μεγαλύτερος χώρος διευθύνσεων. Το IPv6 χρησιμοποιεί διευθύνσεις 128 bit, το οποίο επιτρέπει 2^{128} διαφορετικές διευθύνσεις σε αντίθεση με το IPv4 που χρησιμοποιεί διευθύνσεις 32 bit και επιτρέπει 2^{32} διαφορετικές διευθύνσεις.

Το IPv6 καθορίζει μία νέα μορφή πακέτου, σχεδιασμένη να ελαχιστοποιεί την επεξεργασία των πακέτων από τους δρομολογητές. Επειδή οι κεφαλίδες των πακέτων του IPv4 και IPv6 διαφέρουν σημαντικά, τα δύο πρωτόκολλα δεν μπορούν να συνεργαστούν.

Το IPv6 διαθέτει ενσωματωμένο στην αρχιτεκτονική του το IPsec σε αντίθεση με το IPv4 και η χρήση του είναι δυνατή σε όλη τη διαδρομή του πακέτου στο δίκτυο.

2.1.3 Πρωτόκολλο ελέγχου μετάδοσης (TCP)

Το πρωτόκολλο ελέγχου μετάδοσης (Transmission Control Protocol – TCP) είναι ένα αξιόπιστο πρωτόκολλο πάνω από το IP. Είναι connection-oriented, δηλαδή η μεταφορά δεδομένων γίνεται μέσω σύνδεσης μεταξύ του client και του server, η οποία προσδιορίζεται από ένα σήμα έναρξης

και ένα σήμα τέλους ή διακοπής. Οι περισσότερες σύγχρονες υπηρεσίες στο διαδίκτυο βασίζονται στο TCP, όπως για παράδειγμα το SMTP [29], το FTP [30] και το HTTP [31].

Το TCP παρέχει τις εξής υπηρεσίες:

- **Εγγυημένη παράδοση πακέτων**: ο παραλήπτης ενημερώνει τον αποστολέα αν τα πακέτα έφτασαν σωστά ή κάποιο χάθηκε ή ήταν αλλοιωμένο
- **Παράδοση πακέτων στην σωστή σειρά**: η σειρά που θα ληφθούν πρέπει να είναι ίδια με την σειρά που στάλθηκαν από τον αποστολέα
- **Έλεγχος ροής**: απαιτεί την επιβεβαίωση λήψης του κάθε πακέτου από τον παραλήπτη πριν στείλει το επόμενο. Αλγόριθμοι που χρησιμοποιούνται επιτρέπουν σε πολλαπλά πακέτα δεδομένων να μεταφέρονται ταυτόχρονα για να χρησιμοποιείται αποδοτικότερα το εύρος ζώνης ενός δικτύου
- **Έλεγχος συμφόρησης**: αναγκάζει τον αποστολέα να μην αποστέλλει πακέτα πιο γρήγορα από τον ρυθμό που τα λαμβάνει και τα διαβάζει
- **Διάσπαση των δεδομένων σε μικρότερα τμήματα (πακέτα)**

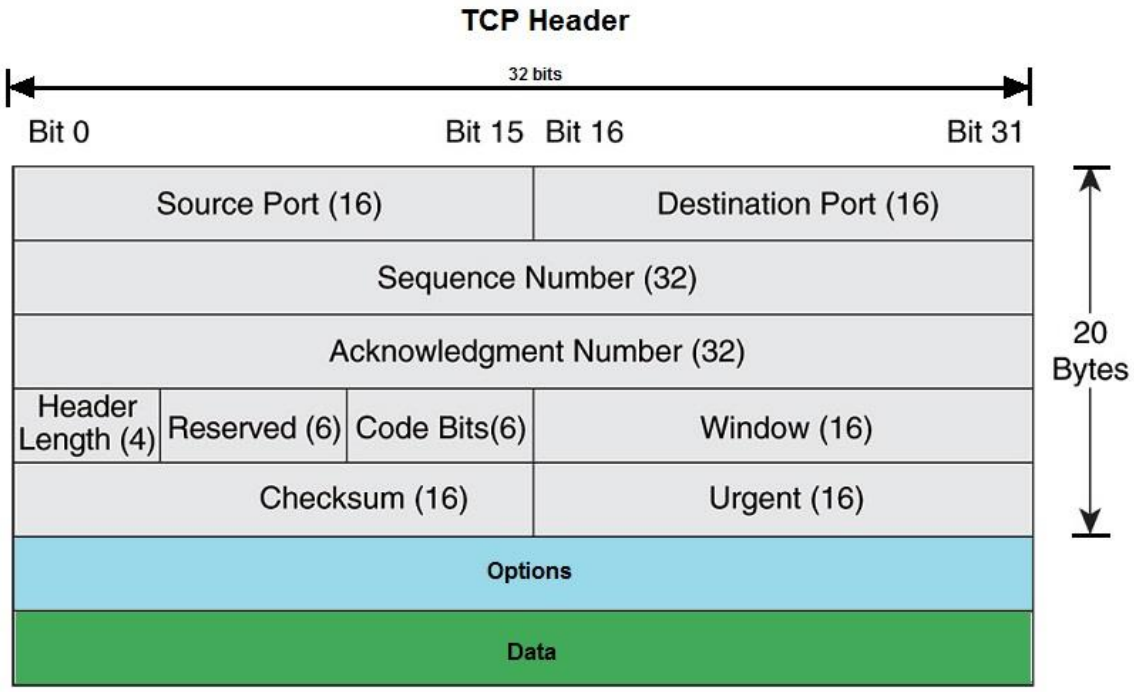
Το TCP δουλεύει ως εξής:

Κάθε πακέτο αριθμείται. Μόνο ο αποστολέας και ο παραλήπτης γνωρίζουν τους αριθμούς των πακέτων (όχι οι ενδιάμεσοι κόμβοι του δικτύου). Ο παραλήπτης λαμβάνει τα πακέτα μέχρι να τελειώσουν. Σε περίπτωση που χαθεί κάποιο πακέτο στο δίκτυο, ο παραλήπτης το ξαναζητάει και ο αποστολέας είναι υπεύθυνος να το ξαναστείλει. Ο παραλήπτης επίσης ελέγχει αν το περιεχόμενο των πακέτων φτάνει σωστά. Τα πακέτα που μεταφέρονται δεν είναι απαραίτητο ότι ακολουθούν όλα το ίδιο δρομολόγιο ως ότου φτάσουν στον παραλήπτη. Η μέθοδος αυτή εξασφαλίζει αξιοπιστία και ταχύτητα καθώς οι ενδιάμεσοι κόμβοι δεν εκτελούν ελέγχους.

Δομή πακέτων TCP

(Για ομάδες των 8 bits διευθύνσεων χρησιμοποιείται ο όρος «octets». Ο όρος “byte” δεν χρησιμοποιείται, επειδή το TCP υποστηρίζεται από ορισμένους υπολογιστές που έχουν μήκος byte διαφορετικό των 8 bits)

Τα πακέτα του TCP ονομάζονται τμήματα (segments). Το τμήμα αποτελείται από την κεφαλίδα (header), η οποία παρέχει συγκεκριμένες πληροφορίες για το TCP και τα δεδομένα. Η κεφαλίδα περιλαμβάνει δέκα υποχρεωτικά πεδία και ένα προαιρετικό πεδίο «επιλογών».



Source Port

θύρα αποστολής

Destination Port

θύρα παραλήπτη

Sequence Number

αριθμός ακολουθίας: αν υπάρχει SYN σημαία τότε είναι ο αρχικός αριθμός ακολουθίας και η πρώτη octet δεδομένων του πακέτου είναι ο ISN+1. Αν δεν υπάρχει η SYN flag, τότε η πρώτη octet δεδομένων είναι ο αριθμός ακολουθίας

Acknowledgment number

Όταν υπάρχει η ACK σημαία, η τιμή αυτού του πεδίου δείχνει τον επόμενο αριθμό ακολουθίας που αναμένει ο αποστολέας

Data offset

καθορίζει το μέγεθος της επικεφαλίδας και επομένως δείχνει και την αρχή των δεδομένων

Reserved

Πεδίο 6 bit δεσμευμένων για μελλοντική χρήση με μηδέν τιμή

Control bits

Περιέχει 6 σημαίες του 1 bit:

- **URG** αν το πεδίο urgent pointer είναι σημαντικό (URGent)
- **ACK** αν το πεδίο επιβεβαίωσης είναι σημαντικό (ACKnowledgment)
- **PSH** Λειτουργία ώθησης (PuSH)
- **RST** Επαναρύθμιση σύνδεσης (ReSeT)
- **SYN** Συγχρονισμός αριθμών ακολουθίας (SYNchronize)
- **FIN** Ο αποστολέας δεν στέλνει άλλα δεδομένα (FINish)

Window

Ο αριθμός από octets δεδομένων που επιθυμεί να δεχτεί ο αποστολέας του πακέτου, αρχίζοντας από εκείνα που δείχνει το πεδίο επιβεβαίωσης

Checksum

Πεδίο μεγέθους 16 bit που χρησιμοποιείται για έλεγχο λαθών στην κεφαλίδα και στα δεδομένα

Options

Μεταβλητή, η οποία καθορίζει ειδικές επιλεγμένες ρυθμίσεις και μπορεί να καταλάβει χώρο στο τέλος της κεφαλίδας. Το μήκος του είναι πολλαπλάσιο των 8 bit

Urgent pointer

Σε περίπτωση που είναι ενεργοποιημένο το URG bit ελέγχου, δείχνει τον αριθμό ακολουθίας της octet που βρίσκεται αμέσως μετά το τελευταίο byte από τα επείγοντα δεδομένα

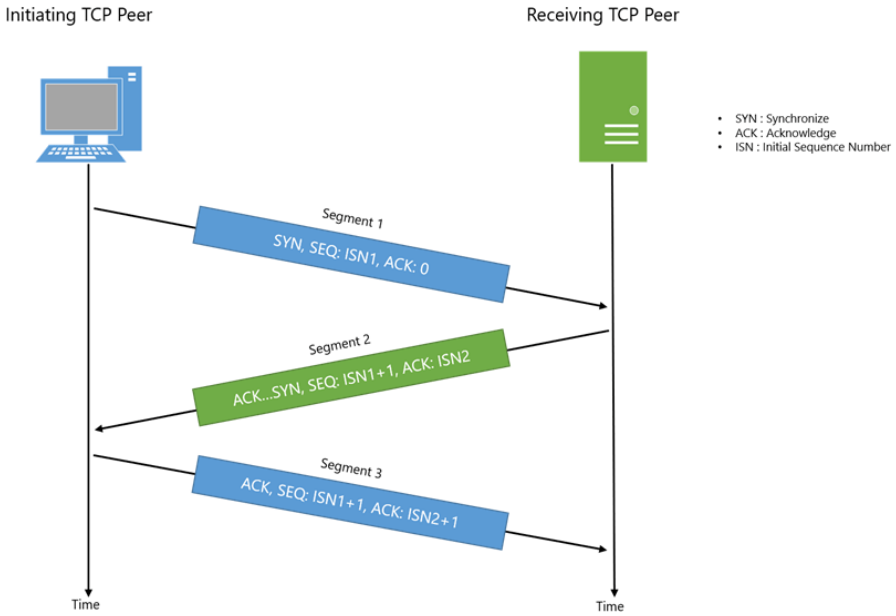
Τρόπος Λειτουργίας

Όπως αναφέρθηκε και πιο πάνω, το TCP είναι connection-oriented, δηλαδή η μεταφορά γίνεται μέσω σύνδεσης μεταξύ του αποστολέα και του παραλήπτη.

Έναρξη

Πριν ξεκινήσει η ανταλλαγή των μηνυμάτων μέσω TCP γίνεται εγκαθίδρυση της σύνδεσης μέσω μιας διαδικασίας που είναι γνωστή ως τριμερής χειραψία (3 way-handshake) και εκτελείται στα τερματικά.

1. Αρχικά ο client στέλνει ένα πακέτο με το SYN bit ενεργοποιημένο. Ο client θέτει το πεδίο αριθμού ακολουθίας στην κεφαλίδα TCP στον αρχικό αριθμό ακολουθίας του (ISN).
2. Ο server απαντάει είτε με SYN και ACK του πρώτου πακέτου του client για να αποδεχτεί τη σύνδεση ή SYN/RST για να ενημερώσει τον client ότι αρνείται την σύνδεση και η διαδικασία σταματά.
3. Όταν ο client πάρει ένα πακέτο SYN/ACK απαντάει αυτή τη φορά με ένα πακέτο ACK. Σε αυτό το σημείο τα δύο μέρη συνδέονται και μπορούν να στέλνουν πλέον τα δεδομένα.



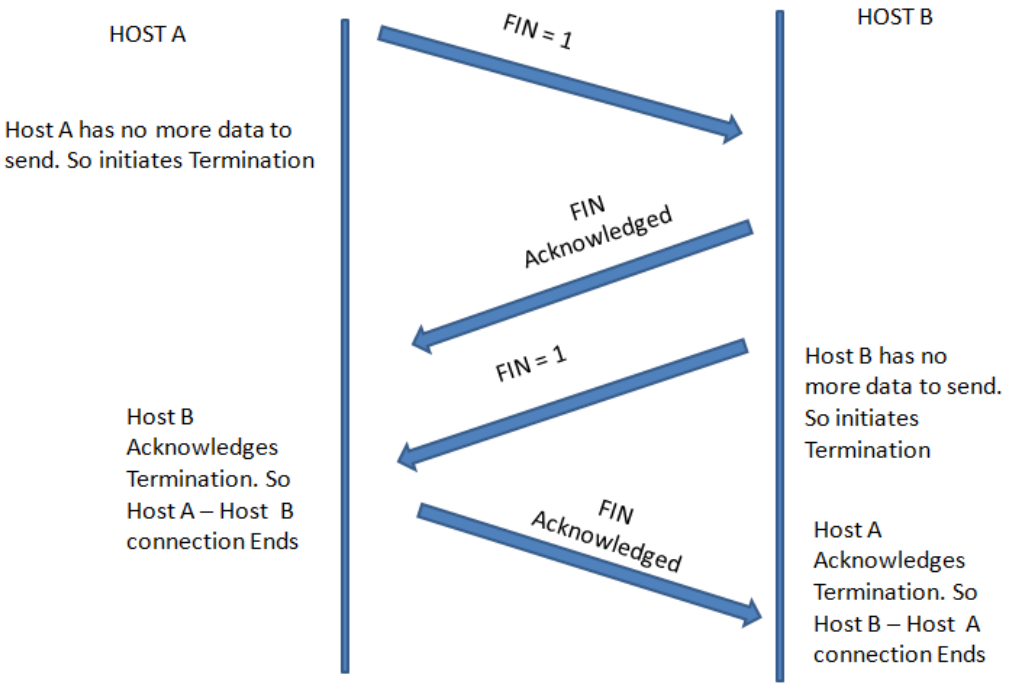
Μεταφορά Δεδομένων

Αφού έχει γίνει η σύνδεση, μπορούν να σταλθούν τα δεδομένα. Πακέτα στέλνονται από τον αποστολέα μέσω των διάφορων δικτύων προς τον παραλήπτη και γίνονται οι έλεγχοι και τα αιτήματα όπως αυτά περιεγράφηκαν σε προηγούμενη ενότητα.

Τερματισμός

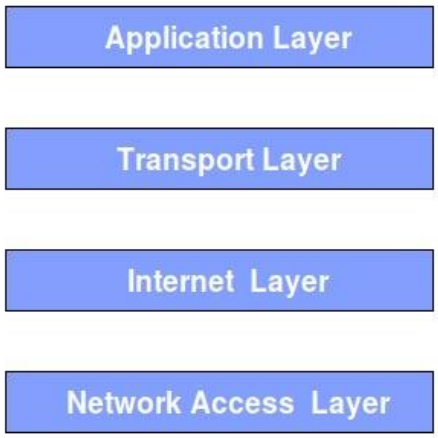
Για τον τερματισμό μια τετραμερής χειραψίας (four-way handshake) πραγματοποιείται με τον τερματισμό κάθε πλευράς ανεξάρτητα.

Όταν κάποιο άκρο επιθυμεί να κλείσει την σύνδεση από πλευράς του, στέλνει ένα πακέτο με το FIN flag ενεργοποιημένο. Τη παραλαβή του πακέτου, επιβεβαιώνει η άλλη πλευρά με ένα ACK και στη συνέχεια στέλνει ένα πακέτο FIN. Η πλευρά που ξεκίνησε τον τερματισμό μπορεί να το επιβεβαιώσει στέλνοντας ένα πακέτο ACK. Μια σύνδεση μπορεί να είναι μισό-ανοιχτή, εννοώντας ότι η μία πλευρά έχει τερματίσει, όχι όμως και η άλλη. Η πλευρά που έχει τερματίσει δεν μπορεί να στείλει πλέον δεδομένα.



2.1.4 Μοντέλο αναφοράς TCP/IP

Αποτελείται από 4 επίπεδα:



Επίπεδο Εφαρμογής

Χειρίζεται αιτήματα για δεδομένα και υπηρεσίες. Περιέχει όλα τα πρωτόκολλα ανώτερου επιπέδου. Κάποια από αυτά είναι το πρωτόκολλο για μεταφορά αρχείων (FTP) , το σύστημα ονομάτων περιοχών (DNS) για την αντιστοίχιση των ονομάτων των υπολογιστών στις διευθύνσεις δικτύου και το HTTP για την προσκόμιση σελίδων από τον παγκόσμιο ιστό και πολλά άλλα.

Επίπεδο μεταφοράς

Έχει σχεδιαστεί να επιτρέπει στους υπολογιστές προέλευσης και προορισμού να συνομιλούν. Υπάρχουν δύο πρωτόκολλα για μεταφορά:

- **TCP**: αξιόπιστο connection-oriented πρωτόκολλο
- **UDP**: αναξιόπιστο connection-less πρωτόκολλο, για εφαρμογές που δεν χρειάζονται την παράδοση των πακέτων με τη σωστή σειρά και πιο ελαστικές σε χάσιμο πακέτων. Το UDP χρησιμοποιείται από εφαρμογές που απαιτούν όσο το δυνατόν πιο μικρή καθυστέρηση. Χαρακτηριστικές εφαρμογές είναι εφαρμογές ήχου και βίντεο.

Επίπεδο δικτύου

Δρομολόγηση πακέτων μέσω ενός δικτύου για τον προορισμό τους.

Επίπεδο διασύνδεσης δικτύου

Μεταφορά πακέτων μεταξύ δύο οντοτήτων. ‘Οταν ένας υπολογιστής θέλει να στείλει πληροφορίες μέσω του TCP/IP πρωτοκόλλουμ πρέπει να γνωρίζει τη διεύθυνση IP στον οποίο θα αποσταλούν οι πληροφορίες αυτές.

2.1.5 Πρωτόκολλο Δυναμικής Καταχώρησης διευθύνσεων (DHCP)

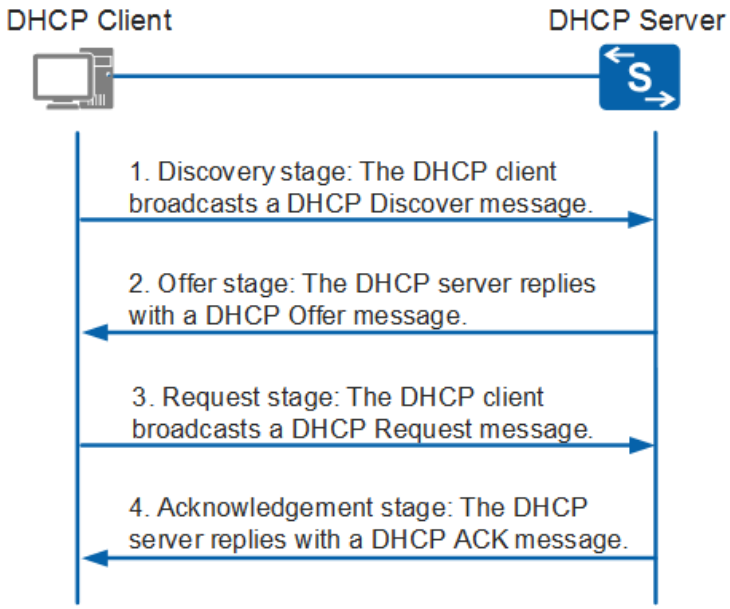
Ο DHCP είναι ένα πρωτόκολλο client-server και επιτρέπει στον χρήστη να αποκτήσει μια IP διεύθυνση αυτόματα.Το DHCP υποστηρίζει τρεις μηχανισμούς για να αντιστοιχίζει διευθύνσεις:

- Αυτόματη αντιστοίχιση (με αντιστοίχιση μόνιμης διεύθυνσης)

- Δυναμική αντιστοίχιση (με διεύθυνση με ημερομηνία λήξης)
- Χειροκίνητη αντιστοίχιση

Ο client και ο server ανταλλάσσουν μεταξύ τους μηνύματα ώστε να πάρει ο χρήστης τις ζητούμενες ρυθμίσεις:

1. Ο client στέλνει ένα DHCPDISCOVER μήνυμα
2. Ο server απαντά με ένα μήνυμα DHCPOFFER
3. Ο πελάτης λαμβάνει το DHCPOFFER και αναμεταδίδει ένα DHCPREQUEST για να ζητήσει ρυθμίσεις.
4. Ο server απαντά με ένα μήνυμα DHCPACK



Το πρωτόκολλο DHCP δεν διαθέτει κανένα μηχανισμό ασφάλειας.

2.1.6 Πρωτόκολλο Μεταφοράς Υπερκειμένου (Http)

Το πρωτόκολλο μεταφοράς υπερκειμένου (HyperText Transfer Protocol, HTTP) είναι ένα πρωτόκολλο επικοινωνίας που τρέχει πάνω από το TCP και καθορίζει τις αιτήσεις (requests) που μπορεί να στείλει ένας client σε έναν server καθώς και τις απαντήσεις (responses) που μπορεί να στείλει ο server.

Κάθε αίτηση περιέχει ένα URL [32], το οποίο είναι μια συμβολοσειρά που προσδιορίζει έναν πόρο, ο οποίος μπορεί να είναι μία εικόνα, μία html [33] σελίδα κλπ.

Το πρωτόκολλο HTTP ορίζει τις ακόλουθες λειτουργίες (ή αλλιώς μέθοδοι) αίτησης:

- **GET**
  Επιστρέφει τον πόρο που υποδεικνύεται από το URL.
- **HEAD**
  Επιστρέφει μόνο τις κεφαλίδες (headers) που υποδεικνύονται από το URL.
- **POST**
  Ο client στέλνει δεδομένα στον server.
- **PUT**
  Αποθηκεύει έναν πόρο στο URL που δίνεται στην αίτηση.
- **DELETE**
  Διαγράφει τον πόρο που υποδεικνύεται από το URL.
- **TRACE**
  Ο client ζητάει από τον server να επιστρέψει την αίτηση. Η μέθοδος αυτή είναι χρήσιμη όταν η επεξεργασία των αιτήσεων δεν γίνεται σωστά και ο πελάτης θέλει να δει ποια αίτηση έλαβε πραγματικά ο διακομιστής.
- **CONNECT**
  Επιτρέπει στον client να πραγματοποιήσει σύνδεση με έναν server μέσω μιας ενδιάμεσης συσκευής.
- **OPTIONS**
  Ο client στέλνει ερωτήματα στον server σχετικά με έναν πόρο και λαμβάνει τις μεθόδους και τις κεφαλίδες που μπορούν να χρησιμοποιηθούν με τον πόρο αυτό.

Κάθε αίτηση λαμβάνει μια απάντηση. Κάθε απάντηση αποτελείται:

- Κωδικό
- Κεφαλίδες (headers)
- Σώμα

Ο κωδικός είναι ένας τριψήφιος αριθμός ο οποίος δηλώνει κατά πόσον εξυπηρετήθηκε η αίτηση και αν δεν εξυπηρετήθηκε, το λόγο. Το πρώτο ψηφίο χρησιμοποιείται για τον χωρισμό των αιτήσεων σε πέντε κατηγορίες:

- **1xx**: ενημερωτική απάντηση. Η αίτηση λήφθηκε και έγινε κατανοητή. Χρησιμοποιείται σπάνια στην πράξη.
- **2xx**: η αίτηση λήφθηκε με επιτυχία και έγινε αποδεκτή. Σε περίπτωση που υπάρχει περιεχόμενο, αποστέλλεται.
- **3xx**: ενημερώνει τον client ότι πρέπει να προβεί σε πρόσθετες ενέργειες για την ολοκλήρωση της αίτησης. Πολλοί από τους κωδικούς χρησιμοποιούνται στην ανακατεύθυνση διευθύνσεων URL.
- **4xx**: η αίτηση απέτυχε λόγω κάποιου σφάλματος που έχει προκληθεί από τον client, όπως ο αιτούμενος πόρος δεν βρέθηκε.
- **5xx**: η αίτηση απέτυχε λόγω κάποιου σφάλματος στον server.

Ενότητα 3: Αρχιτεκτονικές Μελλοντικού Διαδικτύου

3.1. Named Data Network (NDN)

Δίκτυο με βάση τις πληροφορίες (ICN)

Ο όρος «δίκτυο με βάση τις πληροφορίες» εμφανίστηκε γύρω το 2010, πιθανώς εμπνευσμένο από την ομιλία του Van Jacobson [35] σε μια τεχνική συζήτηση στην Google με την ονομασία «A new way to look at networking» [36].

Το δίκτυο με βάση τις πληροφορίες (Information Centric Networking - ICN) είναι μια νέα προσέγγιση για την εξέλιξη υποδομής του διαδικτύου. Η τρέχουσα αρχιτεκτονική επικεντρώνεται στην δημιουργία μιας συνομιλίας μεταξύ δύο μηχανών. Αυτό είναι εμφανές στο σύστημα ονοματολογίας στο οποίο οι διευθύνσεις μεσώ του DNS, υποδεικνύουν ένα μηχάνημα για επικοινωνία προκειμένου να εκτελέσει μία ενέργεια ή να λάβει δεδομένα. Ο στόχος της αρχιτεκτονικής ICN είναι η μετατόπιση της εστίασης από την σύνδεση με ένα μηχάνημα στη λήψη των δεδομένων. Με απλά λόγια στοχεύει στην δημιουργία ενός δικτύου που δεν θα είναι προσανατολισμένο στις IP διευθύνσεις αλλά στη πληροφορία.

Προτείνει την εισαγωγή μοναδικών ονομάτων δεδομένων ως βασική αρχή του διαδικτύου. Τα δεδομένα γίνονται ανεξάρτητα από την τοποθεσία, την εφαρμογή, τον αποθηκευτικό χώρο και τα μέσα μεταφοράς, επιτρέποντας την προσωρινή αποθήκευση και αναπαραγωγή εντός δικτύου. Τα αναμενόμενα οφέλη είναι η βελτιωμένη αποδοτικότητα, η καλύτερη δυνατότητα κλιμάκωσης όσον αφορά τη ζήτηση πληροφοριών / εύρους ζώνης και η καλύτερη ευρωστία σε δύσκολα σενάρια επικοινωνίας.

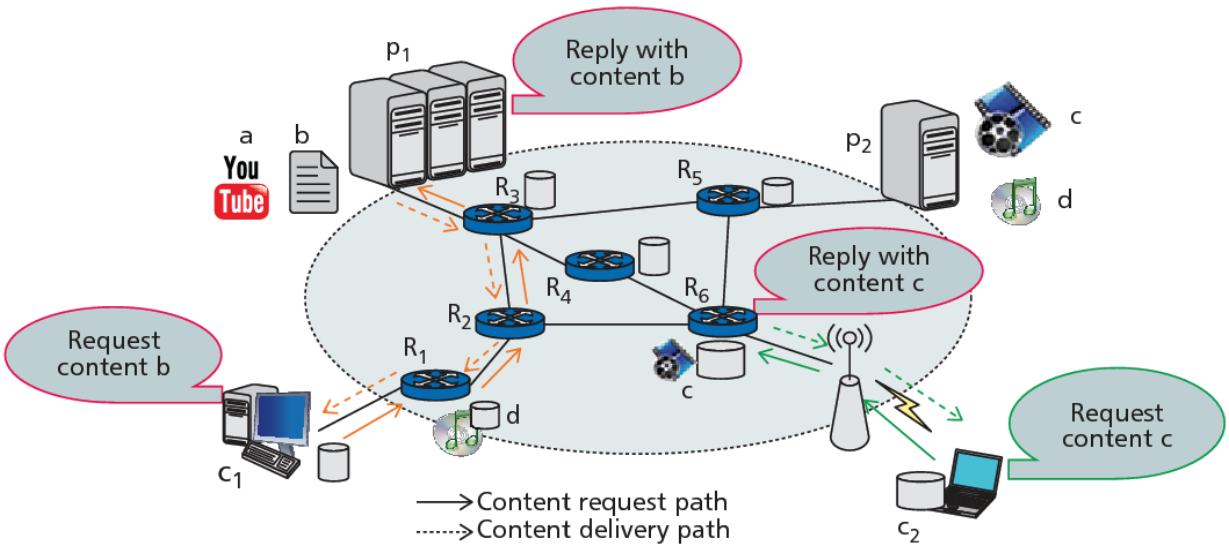
Η βασική ιδέα πίσω από την αρχιτεκτονική αυτή είναι ότι σε περίπτωση αιτήματος από τον χρήστη για κάποια πληροφορία, δεν θα χρησιμοποιηθεί η διεύθυνση στην οποία ενδεχομένως βρίσκεται αυτή η πληροφορία αλλά θα γίνει αίτημα στο ίδιο το περιεχόμενο χρησιμοποιώντας μία περιγραφή του. Στην συνέχεια το δίκτυο θα είναι υπεύθυνο για την δρομολόγηση του αιτήματος ώστε ο χρήστης να πάρει τελικά το αντίγραφο της πληροφορίας που επιθυμεί.

Το πρότυπο ICN αναμένεται επίσης να απαιτεί νέες διεπαφές για εφαρμογές που να αλληλοεπιδρούν με το δίκτυο. Για παράδειγμα, ένα νέο API θα πρέπει να επιτρέπει στους

προγραμματιστές εφαρμογών να επωφελούνται από την ανεξάρτητη ονομασία, την προσωρινή αποθήκευση και τη λειτουργικότητα πολλαπλών προσβάσεων στο ICN.

Με το ICN χρειάζονται νέα μοντέλα ασφάλειας. Το σημερινό μοντέλο εμπιστοσύνης που βασίζεται στον κεντρικό υπολογιστή (ανάκτηση δεδομένων από έναν αξιόπιστο server μέσω ασφαλούς σύνδεσης) δεν ισχύει πλέον. Αντίθετα, οι λειτουργίες ασφαλείας, εμπιστοσύνης και ταυτότητας δεσμεύονται από τα ίδια τα αντικείμενα πληροφοριών, χρησιμοποιώντας υπογεγραμμένα αντικείμενα και εξασφαλίζοντας την ακεραιότητα των δεδομένων ονόματος. Επιπλέον, όλα τα αντικείμενα δεν θα είναι καθολικά προσβάσιμα, απαιτώντας μηχανισμούς εξουσιοδότησης και οριοθέτησης.

Αναμένεται ότι το ICN θα απαιτήσει αλλαγές στο επιχειρηματικό, νομικό και κανονιστικό περιβάλλον. Παραδείγματα περιλαμβάνουν αλλαγές στο πώς γίνονται συμφωνίες ανταλλαγής κίνησης σε ένα δίκτυο που εξαρτάται σε μεγάλο βαθμό από την προσωρινή αποθήκευση για συμμετοχή σε παράδοση αντικειμένων σε τρίτους. Η προσωρινή αποθήκευση αντικειμένων έχει επίσης επιπτώσεις στον τρόπο με τον οποίο το νομικό πλαίσιο ασχολείται με την προσωρινή αποθήκευση. Θα θεωρούνται ότι βρίσκονται στην κατοχή του χρήστη σε σχέση με τα πνευματικά δικαιώματα ή και άλλες νομικές παραβιάσεις;



NDN

Το δίκτυο ονομαστικών δεδομένων ή αλλιώς γνωστό στα αγγλικά ως Named Data Network (NDN) αποτελεί μία προσέγγιση του ICN και είναι ένα από τα πέντε έργα που χρηματοδοτούνται από το εθνικό επιστημονικό ίδρυμα των Ηνωμένων Πολιτειών Αμερικής για το πρόγραμμα «Μελλοντικές Αρχιτεκτονικές του Διαδικτύου».

Το NDN είναι μια εντελώς νέα αρχιτεκτονική, η οποία όμως βασίζεται στις επιτυχίες του σημερινού διαδικτύου αντικατοπτρίζοντας την κατανόηση των δυνατοτήτων και των περιορισμών της τρέχουσας αρχιτεκτονικής του διαδικτύου.

Η IP παρά το γεγονός ότι έχει σχεδιαστεί για επικοινωνία μεταξύ τελικών σημείων, χρησιμοποιείται πλέον με συντριπτική πλειοψηφία για τη διανομή περιεχομένου, τόσο σε σταθερούς υπολογιστές όσο και σε κινητές συσκευές. Όπως γίνεται κατανοητό, η αρχιτεκτονική του διαδικτύου δεν ανταποκρίνεται επαρκώς στην κύρια χρήση της σήμερα.

Επιπλέον όλο και περισσότερες κακόβουλες επιθέσεις γίνονται σε καθημερινή βάση, με αποτέλεσμα να γίνονται συνεχώς προσπάθειες για την ασφάλεια των καναλιών επικοινωνίας, αλλά και πάλι οι παραβιάσεις της ασφάλειας αυξάνονται.

Ο "συνομιλητικός" χαρακτήρας της IP ενσωματώνεται στη μορφή του datagram [37]: Τα IP datagrams μπορούν να ονομάσουν μόνο τελικά σημεία επικοινωνίας (διευθύνσεις IP προέλευσης και προορισμού) σε επίπεδο δικτύου. Τα ονόματα στα datagrams NDN είναι ιεραρχικά δομημένα. Μπορούν να χρησιμοποιηθούν για να ονομάσουν ένα κομμάτι δεδομένων σε μια συζήτηση, όπως κάνει και τώρα το αναγνωριστικό σύνδεσης μεταφοράς TCP / IP (μαζί με τον αριθμό ακολουθίας), αλλά μπορούν επίσης να ονομάσουν ένα κομμάτι δεδομένων από ένα βίντεο απευθείας, αντί να αναγκάσουν να ενσωματωθεί σε μια συνομιλία μεταξύ του client και του server. Αυτή η απλή αλλαγή στο μοντέλο κλεψύδρας, επιτρέπει στη «thin waist» [38] του διαδικτύου να χρησιμοποιεί ονόματα δεδομένων αντί για διευθύνσεις IP για την παράδοση δεδομένων.

Σχεδιαστικές Αρχές Πρωτοκόλλου NDN

Το πρωτόκολλο του NDN διέπεται από τις ακόλουθες σχεδιαστικές αρχές:

- **Ομοιογένεια**
  Το NDN πρέπει να είναι ένα κοινό πρωτόκολλο δικτύου για όλες τις εφαρμογές και τα περιβάλλοντα δικτύου

- **Εστίαση στα ίδια τα δεδομένα**
  Το NDN πρέπει να αντλήσει αμετάβλητα, με μοναδικά ονόματα «πακέτα δεδομένων» τα οποία ζητούνται χρησιμοποιώντας «πακέτα ενδιαφέροντος»

- **Ασφάλεια δεδομένων**
  Η ασφάλεια πρέπει να είναι ιδιοκτησία των δεδομένων και να μην στηρίζεται στην ασφάλεια τρίτων. Στην ιδανική περίπτωση κάθε πακέτο δεδομένων πρέπει να είναι επαληθεύσιμο από μόνο του

- **Ιεραρχική ονομασία**
  Τα πακέτα πρέπει να φέρουν ιεραρχικά ονόματα για να επιτρέπουν την αποπολυπλεξία και να παρέχουν ένα δομημένο πλαίσιο

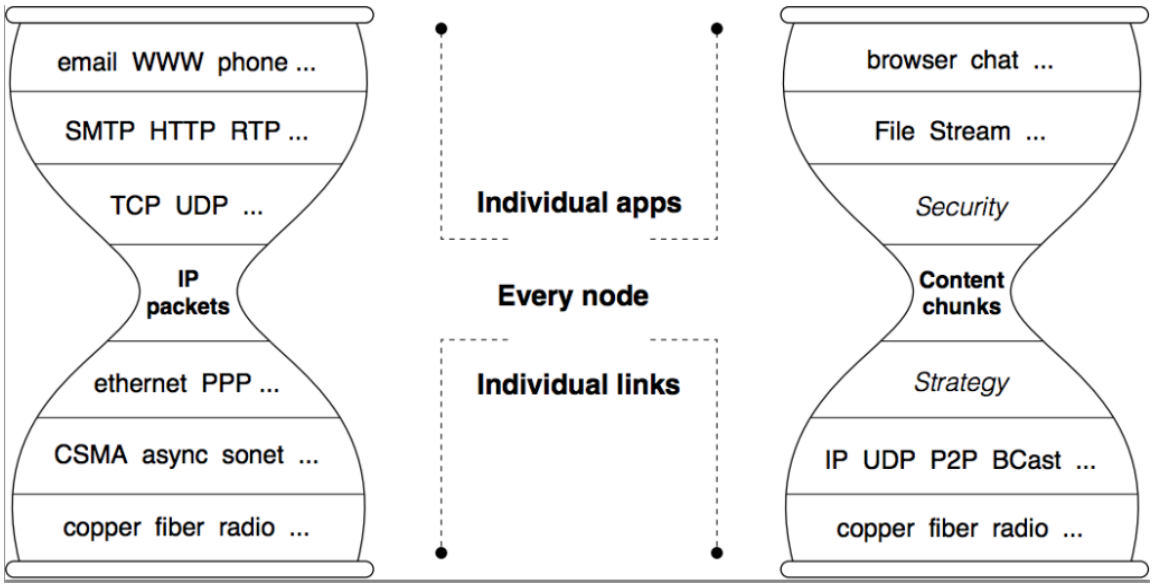
- **Ανακάλυψη ονομάτων στο δίκτυο**
  Τα «πακέτα ενδιαφέροντος» πρέπει να μπορούν να χρησιμοποιούν ημιτελή ονόματα για την ανάκτηση «πακέτων δεδομένων»

- **Ισορροπία ροής μεταξύ συνδέσμων**
  Πάνω από κάθε σύνδεσμο, ένα «πακέτο ενδιαφέροντος» δεν θα πρέπει να φέρνει πίσω περισσότερα από ένα πακέτα δεδομένων.

Αρχιτεκτονικές Αρχές NDN

Παρόλο που το NDN αντιπροσωπεύει μια ολοκαίνουργια πρόταση αρχιτεκτονικής, η μορφή κλεψύδρας το καθιστά συμβατό με το σημερινό διαδίκτυο και οδηγεί σε μια σαφή και απλή στρατηγική ανάπτυξης. Όπως και το IP, το NDN μπορεί να τρέξει σε οτιδήποτε, και οτιδήποτε μπορεί να τρέξει μέσω NDN, συμπεριλαμβανομένης της IP. Οι υπηρεσίες υποδομής IP που χρειάστηκαν πολλά χρόνια για να εξελιχθούν, όπως οι συμβάσεις ονομασίας και η διαχείριση χώρου ονομάτων του DNS μπορούν να χρησιμοποιηθούν εύκολα από το NDN. Πράγματι, επειδή τα ιεραρχικά δομημένα ονόματα του NDN είναι σημασιολογικά συμβατά με τις ιεραρχικά δομημένες διευθύνσεις της IP, βασικά πρωτόκολλα δρομολόγησης μπορούν να χρησιμοποιηθούν όπως είναι.

Η αρχιτεκτονική κλεψύδρα είναι αυτό που κάνει το αρχικό σχέδιο διαδικτύου ισχυρό. Βασίζεται σε ένα παγκόσμιο στρώμα δικτύου (IP) που υλοποιεί την ελάχιστη λειτουργικότητα που απαιτείται για την παγκόσμια διασύνδεση. Αυτή η αποκαλούμενη «thin waist» αποτελεί βασικό παράγοντα της τεράστιας ανάπτυξης του διαδικτύου, επιτρέποντας στις τεχνολογίες χαμηλών και ανώτερων στρώματων να καινοτομούν χωρίς περιττούς περιορισμούς. Το NDN διατηρεί την ίδια αρχιτεκτονική σχήματος κλεψύδρας.



Ο διαχωρισμός επιπέδου δρομολόγησης και προώθησης έχει αποδειχθεί απαραίτητος για την ανάπτυξη του Διαδικτύου. Επιτρέπει την λειτουργία του επιπέδου προώθησης ενώ το σύστημα

δρομολόγησης συνεχίζει να εξελίσσεται με την πάροδο του χρόνου. Το NDN συμμορφώνεται με την ίδια αρχή, επιτρέποντας την ανάπτυξη του NDN με την καλύτερη διαθέσιμη τεχνολογία προώθησης.

Η αρχή «από άκρο σε άκρο» [39] διατηρείται και επεκτείνεται από το NDN καθώς καθιστά δυνατή την ανάπτυξη ισχυρών εφαρμογών εν όψει αποτυχιών δικτύου.

Η κυκλοφορία δικτύου πρέπει να είναι αυτορυθμιζόμενη. Η παράδοση δεδομένων με ισορροπημένη ροή είναι απαραίτητη για τη σταθερή λειτουργία του δικτύου και το NDN σχεδιάζει την ισορροπία ροής στη «thin waist».

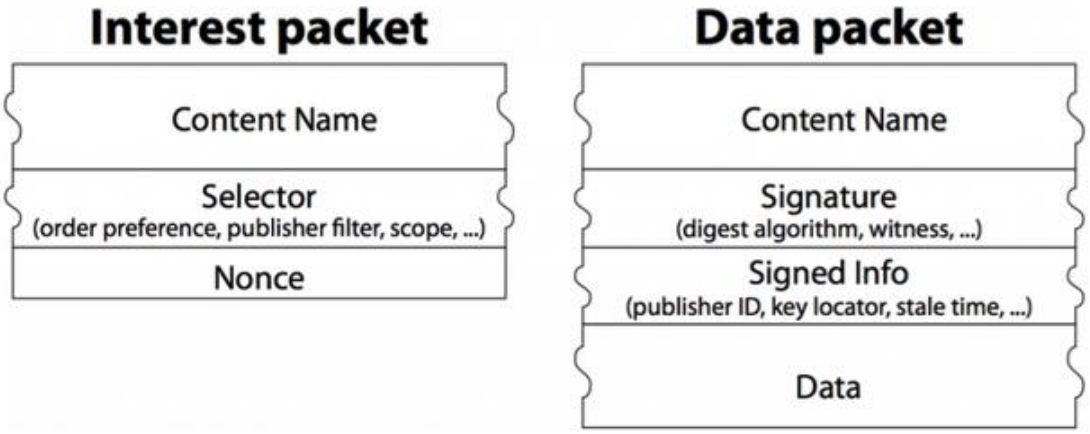
Η ασφάλεια είναι αναγκαίο να ενσωματωθεί στην αρχιτεκτονική. Η ασφάλεια στην τρέχουσα αρχιτεκτονική του διαδικτύου είναι ανεπαρκής και δεν ανταποκρίνεται στις απαιτήσεις του σήμερα. Το NDN παρέχει ένα μια τέτοια ασφάλεια ακριβώς στη «thin waist» υπογράφοντας όλα τα ονόματα δεδομένων.

Αρχιτεκτονική NDN

Όπως αναφέρθηκε προηγουμένως, η «thin waist» είναι το κεντρικό στοιχείο της αρχιτεκτονικής του NDN όπως και της IP. Ωστόσο, επειδή η «thin waist» του NDN χρησιμοποιεί ονόματα δεδομένων αντί για διευθύνσεις IP, αυτή η φαινομενικά απλή αλλαγή οδηγεί σε σημαντικές διαφορές μεταξύ IP και NDN στις λειτουργίες τους για την παράδοση δεδομένων.

Η επικοινωνία στο NDN καθοδηγείται από το δέκτη, δηλαδή από τον καταναλωτή δεδομένων με την ανταλλαγή δύο τύπων πακέτων:

- **Πακέτο Ενδιαφέροντος** (Interest packet)
- **Πακέτο Δεδομένων** (Data packet)

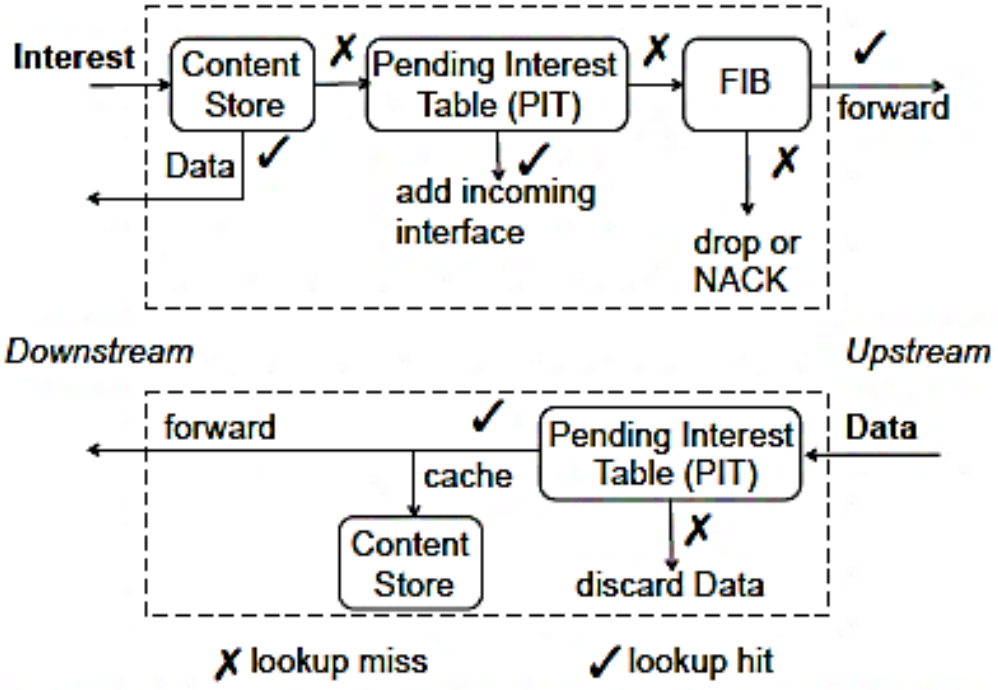


Για να λάβει δεδομένα, ένας καταναλωτής στέλνει ένα πακέτο ενδιαφέροντος, το οποίο φέρει ένα όνομα που προσδιορίζει τα επιθυμητά δεδομένα.

Ένας δρομολογητής θυμάται τη διεπαφή από την οποία έρχεται το αίτημα και στη συνέχεια προωθεί το πακέτο ενδιαφέροντος αναζητώντας το όνομα στη βάση πληροφοριών προώθησης (Forwarding Information Base - FIB), η οποία συμπληρώνεται από ένα πρωτόκολλο δρομολόγησης με βάση το όνομα. Μόλις το πακέτο ενδιαφέροντος φθάσει σε έναν κόμβο που έχει τα ζητούμενα δεδομένα, αποστέλλεται ένα πακέτο δεδομένων, το οποίο μεταφέρει το όνομα και το περιεχόμενο των δεδομένων μαζί με μια υπογραφή από το κλειδί του παραγωγού. Αυτό το πακέτο δεδομένων ακολουθεί αντίστροφα τη διαδρομή του πακέτου ενδιαφέροντος για να επιστρέψει στον καταναλωτή. Τα πακέτα ενδιαφέροντος δρομολογούνται προς τους παραγωγούς δεδομένων με βάση τα ονόματα που μεταφέρονται στα πακέτα ενδιαφέροντος και τα πακέτα δεδομένων επιστρέφονται με βάση τις πληροφορίες κατάστασης που δημιουργούνται από τα ενδιαφέροντα.

Ο δρομολογητής αποθηκεύει σε έναν πίνακα εκκρεμών ενδιαφερόντων (Pending Interest Table - PIT) όλα τα ενδιαφέροντα που περιμένουν να επιστρέψουν πακέτα δεδομένων. Όταν πολλαπλά ενδιαφέροντα για τα ίδια δεδομένα λαμβάνονται, μόνο το πρώτο αποστέλλεται προς την πηγή δεδομένων. Κάθε καταχώριση PIT περιέχει το όνομα του ενδιαφέροντος και ένα σύνολο διεπαφών από το οποίο έχουν ληφθεί τα ενδιαφέροντα για το ίδιο όνομα. Όταν φτάσει ένα πακέτο δεδομένων, ο δρομολογητής βρίσκει την αντίστοιχη καταχώρηση PIT και προωθεί τα δεδομένα σε όλες τις διεπαφές που αναφέρονται στην καταχώρηση PIT. Στη συνέχεια, ο δρομολογητής αφαιρεί την αντίστοιχη καταχώρηση PIT και αποθηκεύει τα δεδομένα στο απόθεμα περιεχομένου (Content Store). Επειδή ένα πακέτο δεδομένων NDN έχει νόημα

ανεξάρτητα από το πού προέρχεται ή από όπου μπορεί να προωθηθεί, ο δρομολογητής μπορεί να το αποθηκεύσει σε μια μνήμη cache [40] για να ικανοποιήσει τα μελλοντικά αιτήματα.



Ονόματα NDN

Ο σχεδιασμός του NDN προϋποθέτει ιεραρχικά δομημένα ονόματα (π.χ. /youtube/video/video1.mkv). Αυτή η ιεραρχική δομή είναι χρήσιμη για εφαρμογές που αντιπροσωπεύουν σχέσεις μεταξύ τμημάτων δεδομένων. Για παράδειγμα, το πέμπτο chunk της δεύτερης έκδοσης του βίντεο μπορεί να ονομάζεται /youtube/video/video1.mkv/2/5.

Οι κοινές δομές που είναι απαραίτητες για να επιτρέψουν στα προγράμματα να λειτουργούν με ονόματα NDN μπορούν να επιτευχθούν με συμβάσεις που συμφωνούνται μεταξύ παραγωγών δεδομένων και καταναλωτών.

Οι συμβάσεις ονόματος είναι συγκεκριμένες στις εφαρμογές αλλά οι δρομολογητές δεν γνωρίζουν την έννοια ενός ονόματος (αδιαφανείς στο δίκτυο). Αυτό επιτρέπει σε κάθε εφαρμογή να επιλέξει το σχήμα ονομασίας που ταιριάζει στις ανάγκες της και επιτρέπει στα σχέδια ονομασίας να εξελίσσονται ανεξάρτητα από το δίκτυο.

Για να ανακτήσουν τα δεδομένα που δημιουργούνται δυναμικά, οι καταναλωτές πρέπει να είναι σε θέση να κατασκευάσουν ντετερμινιστικά το όνομα για ένα επιθυμητό κομμάτι δεδομένων. Αυτό μπορεί να γίνει με δύο τρόπους:

- Χρησιμοποιώντας ένα ντετερμινιστικό αλγόριθμο που επιτρέπει στον παραγωγό και τον καταναλωτή να καταλήξουν στο ίδιο όνομα με βάση τα δεδομένα που είναι διαθέσιμα και στους δύο.
- Χρησιμοποιώντας ένα μέρος των ονομάτων.

Για παράδειγμα, ο καταναλωτής μπορεί να ζητήσει «/youtube/video/video1.mkv» και να πάρει πίσω ένα πακέτο δεδομένων που ονομάζεται «/youtube/video/video1.mkv/1/1» . Ο καταναλωτής μπορεί στη συνέχεια να προσδιορίσει τα μεταγενέστερα τμήματα και να τα ζητήσει, χρησιμοποιώντας έναν συνδυασμό πληροφοριών που αποκαλύπτονται από το πρώτο πακέτο δεδομένων και τη σύμβαση ονομασίας που συμφωνήθηκε.

Δεν χρειάζεται όλα τα ονόματα να είναι μοναδικά παγκοσμίως, εκτός από αυτά που χρησιμοποιούνται για την ανάκτηση δεδομένων σε παγκόσμιο επίπεδο. Τα ονόματα που προορίζονται για τοπική επικοινωνία ενδέχεται να βασίζονται σε μεγάλο βαθμό στο τοπικό πλαίσιο και απαιτούν μόνο τοπική δρομολόγηση για να βρουν αντίστοιχα δεδομένα.

Όπως η διαχείριση χώρου διεύθυνσης IP δεν αποτελεί μέρος της αρχιτεκτονικής IP έτσι και η διαχείριση χώρου ονόματος δεν αποτελεί μέρος της αρχιτεκτονικής NDN. Αλλά όπως είναι κατανοητό αποτελεί το πιο σημαντικό κομμάτι του NDN. Τα ονόματα δεδομένων επιτρέπουν στο NDN να υποστηρίζει αυτόματα διάφορες λειτουργίες, όπως:

- **Διανομή περιεχομένου**: πολλοί χρήστες ζητούν τα ίδια δεδομένα σε διαφορετικές χρονικές στιγμές
- **Multicast**: πολλοί χρήστες ζητούν τα ίδια δεδομένα ταυτόχρονα
- **Κινητικότητα**: χρήστες ζητούν δεδομένα από διαφορετικές τοποθεσίες
- **Ανεκτικότητα σε καθυστερήσεις**: χρήστες ανακτούν δεδομένα μέσω διακεκομμένης σύνδεσης

Ασφάλεια δεδομένων

Σε αντίθεση με το TCP/IP που αφήνει την ευθύνη για την ασφάλεια στα άκρα της επικοινωνίας, το NDN ενσωματώνει την ασφάλεια στα ίδια τα δεδομένα και αυτό επιτυγχάνεται με το να απαιτεί από τους παραγωγούς των δεδομένων να υπογράφουν κρυπτογραφικά κάθε ένα πακέτο δεδομένων που παράγουν. Η χρήση ή όχι ασφάλειας δεν αποτελεί επιλογή. Οι παραγωγοί δεδομένων δεν μπορούν να την παραλείψουν.

Η υπογραφή συνδυασμένη με τις πληροφορίες των δεδομένων των εκδοτών, επιτρέπει τον προσδιορισμό της προέλευσης των δεδομένων, με αποτέλεσμα οι καταναλωτές να μην ενδιαφέρονται για τον τρόπο και την προέλευση των δεδομένων.

Ωστόσο, για να είναι πρακτική, η προσέγγιση ασφάλειας απαιτεί αλλαγές. Η ασφάλεια που βασίζεται στην κρυπτογραφία δημόσιου κλειδιού έχει θεωρηθεί αναποτελεσματική και ακατάλληλη. Εκτός από τις αποτελεσματικές ψηφιακές υπογραφές, το NDN χρειάζεται ευέλικτους και αποτελεσματικούς μηχανισμούς για τη διαχείριση της εμπιστοσύνης των χρηστών. Το NDN προσφέρει ένα υποσχόμενο υπόστρωμα για την επίτευξη αυτών των στόχων ασφάλειας. Τέλος, η προσέγγιση NDN από άκρο σε άκρο ως προς την ασφάλεια διευκολύνει την εμπιστοσύνη μεταξύ εκδοτών και καταναλωτών. Αυτό προσφέρει στους εκδότες και τους καταναλωτές μεγάλη ευελιξία στο να διαλέξουν ή να εξατομικεύσουν τα μοντέλα εμπιστοσύνης τους.

Η ασφάλεια των δεδομένων του NDN μπορεί να επεκταθεί στον έλεγχο προσβασιμότητας περιεχομένου και στην ασφάλεια των υποδομών. Οι εφαρμογές μπορούν να αποκτούν πρόσβαση στα δεδομένα μέσω κρυπτογράφησης και να διανέμουν τα κλειδιά ως κρυπτογραφημένα δεδομένα NDN, περιορίζοντας την περίμετρο ασφάλειας δεδομένων στο πλαίσιο μιας ενιαίας εφαρμογής. Η απαίτηση υπογραφής στα μηνύματα δρομολόγησης και ελέγχου δικτύου παρέχει την απαραίτητη ασφάλεια πρωτοκόλλου δρομολόγησης.

Δρομολόγηση και Προώθηση

Το NDN δρομολογεί και προωθεί τα πακέτα με τη χρήση των ονομάτων, κάτι που εξαλείφει τα προβλήματα που αντιμετωπίζουν οι διευθύνσεις στην αρχιτεκτονική IP:

- Εξάντληση χώρου διευθύνσεων
- NAT μετάβαση [41]
- Κινητικότητα
- Διαχείριση διεύθυνσης

Στο NDN αυτά τα προβλήματα έχουν λυθεί. Δεν υπάρχει θέμα εξάντλησης διεύθυνσης, καθώς ο χώρο ονομάτων είναι απεριόριστος. Δεν υπάρχει θέμα μετάβασης NAT, δεδομένου ότι ο κεντρικός υπολογιστής δεν χρειάζεται να εκθέσει τη διεύθυνσή του για να παρέχει περιεχόμενο. Η κινητικότητα, η οποία απαιτεί αλλαγή διευθύνσεων στην IP, δεν διακόπτει πλέον την επικοινωνία, δεδομένου ότι τα ονόματα των δεδομένων παραμένουν ίδια. Τέλος, δεν απαιτείται πλέον η αντιστοίχιση και διαχείριση διευθύνσεων σε τοπικά δίκτυα.

Τα δοκιμασμένα πρωτόκολλα δρομολόγησης (π.χ. BGP [42]) μπορούν να χρησιμοποιηθούν ως πρωτόκολλα σε δίκτυα NDN. Αντί για IP προθέματα, ένας δρομολογητής NDN αναγγέλλει προθέματα ονομάτων που καλύπτουν τα δεδομένα που ο δρομολογητής είναι πρόθυμος να εξυπηρετήσει.

Ωστόσο, ένας απεριόριστος χώρος ονομάτων θέτει το ζήτημα του τρόπου με τον οποίο διατηρείται ο έλεγχος των μεγεθών του πίνακα δρομολόγησης. Ένα άλλο σημαντικό ερώτημα είναι αν η αναζήτηση σε μεταβλητού μήκους ιεραρχικά ονόματα μπορεί να γίνει γρήγορα και αποτελεσματικά.

Επίσης το NDN συνεισφέρει στη σημαντική βελτίωση της ασφάλειας δρομολόγησης:

- η υπογραφή όλων των δεδομένων, συμπεριλαμβανομένων των μηνυμάτων δρομολόγησης, αποτρέπει να παραβιάζονται ή να αλλοιώνονται.
- η δρομολόγηση πολλαπλών διαδρομών μπορεί να περιορίσει τις επιθέσεις στα προθέματα διότι οι δρομολογητές μπορούν να ανιχνεύσουν την ανωμαλία που προκλήθηκε και να ανακτήσουν τα δεδομένα μέσω εναλλακτικών διαδρομών.
- επειδή τα μηνύματα NDN μπορούν να μιλούν μόνο για δεδομένα και δεν απευθύνονται σε κεντρικούς υπολογιστές, καθιστά δύσκολη την αποστολή κακόβουλων πακέτων σε έναν συγκεκριμένο στόχο.

Για να είναι αποτελεσματικές, οι επιθέσεις κατά του NDN πρέπει να επικεντρωθούν στην άρνηση παροχής υπηρεσιών.

Επίπεδο Δεδομένων

Στο επίπεδο δεδομένων του NDN, ο πίνακας εκκρεμών ενδιαφερόντων (PIT) καταγράφει όλα τα εκκρεμή ενδιαφέροντα και τις εισερχόμενες διεπαφές τους. Κάθε καταχώρηση στον PIT δείχνει την προσδοκία για ένα πακέτο δεδομένων η οποία καταργείται με την λήψη των δεδομένων ή όταν συμβεί timeout. Αυτή η κατάσταση ανά πακέτο είναι μια θεμελιώδης διαφοροποίηση σε σχέση με την IP. Οι πληροφορίες κατάστασης καθιστούν το επίπεδο δεδομένων του NDN προσαρμόσιμο στο χειρισμό τυχών αποτυχιών του δικτύου και αποτελεσματικό στη χρήση των πόρων του.

Ένας κόμβος NDN μπορεί να παρακολουθεί την απόδοση σχετικά με την παράδοση πακέτων σε διαφορετικές διεπαφές και να ανιχνεύει τυχόν απώλεια πακέτων. Επίσης ένα πακέτο ενδιαφέροντος που επιστρέφει πίσω στον ίδιο κόμβο αναγνωρίζεται εύκολα και απορρίπτεται. Έτσι τα πακέτα ενδιαφέροντος όπως και τα πακέτα δεδομένων δεν μπαίνουν σε έναν ατέρμονα βρόχο. Οι μεμονωμένοι δρομολογητές NDN μπορούν να λαμβάνουν τοπικές αποφάσεις σχετικά με την προώθηση των ενδιαφερόντων μέσω πολλαπλών διεπαφών για την επιλογή υπηρεσίας και την εξισορρόπηση φορτίου. Επίσης μπορούν να ανιχνεύουν γρήγορα προβλήματα που υπάρχουν και να επιλέγουν εναλλακτικές διαδρομές για την αποφυγή προβλημάτων.

Caching

Η αυτόματη προσωρινή αποθήκευση στο δίκτυο ενεργοποιείται με την ονομασία δεδομένων. Δεδομένου ότι κάθε πακέτο δεδομένων NDN είναι ουσιαστικά ανεξάρτητο από τον τρόπο και το μέρος που προέρχεται ή όπου μπορεί να προωθηθεί, ένας δρομολογητής μπορεί να το αποθηκεύσει για να ικανοποιήσει μελλοντικά αιτήματα.

Αφού λάβει ένα νέο ενδιαφέρον, ο δρομολογητής ελέγχει πρώτα το απόθεμα περιεχομένου (Content Store). Εάν υπάρχουν δεδομένα που το όνομα τους εμπίπτει με το όνομα που έχει το πακέτο ενδιαφέροντος, τα δεδομένα στέλνονται πίσω ως απάντηση.

Τόσο οι δρομολογητές IP όσο και οι NDN αποθηκεύουν πακέτα δεδομένων. Η διαφορά είναι ότι οι δρομολογητές IP δεν μπορούν να επαναχρησιμοποιήσουν τα δεδομένα μετά την προώθησή τους, ενώ οι δρομολογητές NDN είναι σε θέση να επαναχρησιμοποιήσουν τα δεδομένα αφού αναγνωρίζονται από τα ονόματα των δεδομένων. Για τα στατικά αρχεία, το NDN επιτυγχάνει σχεδόν βέλτιστη παράδοση δεδομένων. Ακόμη και το δυναμικό περιεχόμενο μπορεί να επωφεληθεί από την προσωρινή αποθήκευση στην περίπτωση του mutlicasting.

Η χρησιμοποίηση της cache μπορεί να προκαλέσει ανησυχίες σχετικά με την ιδιωτικότητα. Τα δίκτυα IP προσφέρουν περιορισμένη προστασία της ιδιωτικής ζωής:

- ελέγχοντας την κεφαλίδα ή το ωφέλιμο φορτίο μπορεί να βρεθεί το περιεχόμενο του πακέτου IP
- ελέγχοντας τη διεύθυνση προορισμού, μπορεί να βρεθεί ο υπολογιστής που έκανε αίτηση για τα δεδομένα

Το NDN καταργεί πλήρως τις πληροφορίες σχετικά με το ποιος ζητά τα δεδομένα. Εάν δεν έχει συνδεθεί απευθείας με τον αιτούντα υπολογιστή, ένας δρομολογητής θα γνωρίζει μόνο ότι κάποιος έχει ζητήσει κάποια δεδομένα, άρα δεν θα γνωρίζει ποιος δημιούργησε το αίτημα. Έτσι, η αρχιτεκτονική NDN προσφέρει προστασία της ιδιωτικής ζωής σε ένα θεμελιώδη διαφορετικό επίπεδο σε σχέση με τα δίκτυα IP.

Μεταφορά

Η αρχιτεκτονική του NDN δεν έχει ξεχωριστό στρώμα μεταφοράς (transport layer). Μετακινεί:

- τις λειτουργίες των σημερινών πρωτοκόλλων μεταφοράς στις εφαρμογές και τις βιβλιοθήκες τους
- το στοιχείο στρατηγικής στο επίπεδο προώθησης

Η ακεραιότητα και αξιοπιστία των δεδομένων διαχειρίζονται από τις διαδικασίες των εφαρμογών, όπου μπορούν να πραγματοποιηθούν οι κατάλληλοι έλεγχοι αξιοπιστίας και υπογραφής των δεδομένων.

Όταν ένας δρομολογητής NDN έχει υπερφορτωθεί από την εισερχόμενη κίνηση δεδομένων ενός συγκεκριμένου γείτονα, απλά επιβραδύνει ή σταματά να στέλνει τα πακέτα ενδιαφέροντος σε αυτόν. Αυτό σημαίνει επίσης ότι το NDN εξαλείφει την εξάρτηση από τους τελικούς κεντρικούς υπολογιστές να εκτελούν έλεγχο συμφόρησης. Έτσι, το NDN αποφεύγει την κατάρρευση της συμφόρησης που μπορεί να συμβεί στο σημερινό διαδίκτυο όταν ένα πακέτο χάνεται στους τελευταίους κόμβους και το εύρος ζώνης καταναλώνεται κυρίως από επαναλαμβανόμενες αναμεταδόσεις.

3.2. Delay Tolerant Networking (DTN)

Το Delay Tolerant Networking (DTN) είναι μια προσέγγιση στην αρχιτεκτονική δικτύου υπολογιστών, που επιδιώκει να αντιμετωπίσει τα τεχνικά ζητήματα σε ετερογενή δίκτυα, τα οποία ενδέχεται να μην έχουν συνεχή συνδεσιμότητα στο δίκτυο. Παραδείγματα τέτοιων δικτύων, είναι αυτά που λειτουργούν σε κινητά ή ακραία χερσαία περιβάλλοντα ή δίκτυα στο διάστημα.

Τα υπάρχοντα πρωτόκολλα διαδικτύου δεν λειτουργούν αποτελεσματικά στα παραπάνω δίκτυα λόγω ορισμένων θεμελιωδών παραδοχών που έχουν ενσωματωθεί στην αρχιτεκτονική του σημερινού διαδικτύου:

- υπάρχει end-to-end μονοπάτι μεταξύ πηγής και προορισμού κατά τη διάρκεια της επικοινωνίας
- η end-to-end απώλεια είναι σχετικά μικρή
- όλοι οι δρομολογητές και οι τελικοί σταθμοί υποστηρίζουν τα πρωτόκολλα TCP / IP
- οι εφαρμογές δεν χρειάζεται να ανησυχούν για την απόδοση της επικοινωνίας
- η επιλογή μιας ενιαίας διαδρομής μεταξύ του αποστολέα και του δέκτη είναι επαρκή για την επίτευξη αποδεκτών επιδόσεων επικοινωνίας

Η αρχιτεκτονική του DTN βασίζεται σε μια αφαιρετικότητα των μηνυμάτων μεταγωγής και σχεδιάστηκε για να χαλαρώσει τις περισσότερες από τις παραπάνω παραδοχές.

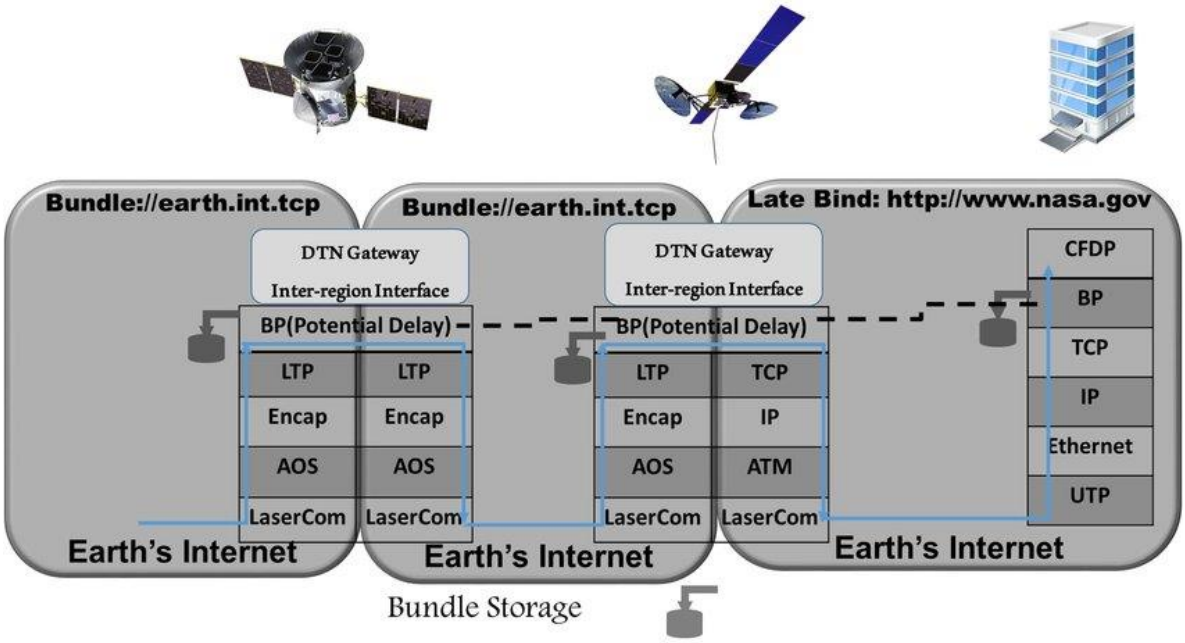
Το DTN σκοπεύει να τρέχει πάνω από τα υπάρχοντες στοίβες πρωτοκόλλων σε διάφορες αρχιτεκτονικές δικτύου και να παρέχει μία «store and forward» λειτουργία στις πύλες ανάμεσα στα δίκτυα όταν ένας κόμβος βρίσκεται από φυσικής απόψεως σε δύο ή περισσότερα δίκτυα.

Για παράδειγμα, στο διαδίκτυο θα λειτουργεί πάνω από το TCP/IP και για συνδέσεις στο διάστημα μπορεί να παρέχει μια υπηρεσία πύλης στο CFDP [43].

Κάθε ένα από αυτά τα περιβάλλοντα δικτύου έχει τις δικές του εξιδεικευμένες στοίβες πρωτοκόλλων και δική του σημασιολογία ονομάτων, τα οποία έχουν αναπτυχθεί για το δικό τους συγκεκριμένο τομέα εφαρμογών. Διαλειτουργικότητα μεταξύ των δικτύων επιτυγχάνεται από ειδικές πύλες DTN, οι οποίες βρίσκονται στα σημεία διασύνδεσης.

Αρχιτεκτονική DTN

Η αρχιτεκτονική DTN περιλαμβάνει τις έννοιες των περιοχών και των πυλών. Στην παρακάτω εικόνα παρουσιάζονται τρεις ανομοιογενείς περιοχές που συνδέονται μεταξύ τους με πύλες DTN:



Η αρχιτεκτονική αυτή προσβλέπει στην εξέλιξη μικρού αριθμού τύπων περιοχών, των οποίων τα στιγμιότυπα του ίδιου τύπου θα υλοποιούν παρόμοια στοίβα πρωτοκόλλων.

Τα όρια των περιοχών χρησιμοποιούνται ως σημεία διασύνδεσης μεταξύ των διαφορετικών ανομοιογενών δικτύων. Δύο κόμβοι βρίσκονται στην ίδια περιοχή όταν μπορούν να επικοινωνήσουν χωρίς να χρησιμοποιούν τις πύλες DTN και με την χρήση των τοπικών

πρωτοκόλλων. Οι πύλες DTN συμβαδίζουν με την έννοια του «ενδιάμεσου σταθμού». Ως ενδιάμεσος σταθμός, περιγράφεται ένα σημείο μέσω του οποίου τα δεδομένα πρέπει να περάσουν ώστε να αποκτήσουν πρόσβαση στην περιοχή. Αυτό το σημείο μπορεί να χρησιμοποιηθεί για την επιβολή ελέγχων και πολιτικών καθώς και την μετάφραση συγκεκριμένων ανά περιοχή κωδικοποιήσεων.

Μία πύλη DTN που βρίσκεται ανάμεσα σε δύο περιοχές αποτελείται από δύο «μισά», καθένα από τα οποία βρίσκεται στις γειτονικές περιοχές πάνω από τα αντίστοιχα πρωτόκολλα μεταφοράς. Η πύλη DTN είναι υπεύθυνη για την αποθήκευση μηνυμάτων όταν η αξιόπιστη παράδοση είναι υποχρεωτική και για τον συσχετισμό διαφορετικών μεταφορών με την «μετάφραση» των καθολικών ονομάτων σε τοπικά ονόματα για την κατεύθυνση της κίνησης στη γειτονική περιοχή. Επίσης, μπορεί να εκτελέσει αυθεντικοποίηση και έλεγχο πρόσβασης στην εισερχόμενη κίνηση για να σιγουρευτεί ότι η προώθηση επιτρέπεται.

Για την δρομολόγηση των μηνυμάτων DTN, ως αναγνωριστικά για αντικείμενα ή ομάδες αντικείμενων, χρησιμοποιούνται ονόματα, τα οποία αποτελούνται από δύο μέρη:

{ Όνομα περιοχής / Όνομα οντότητας }

Πιο συγκεκριμένα, το πρώτο μέρος είναι ένα μοναδικά παγκοσμίως ιεραρχικά δομημένο όνομα περιοχής. Χρησιμοποιείται από τις πύλες DTN για την αναζήτηση του/των μονοπατιού/ων σε μία ή περισσότερες πύλες στην άκρη της συγκεκριμένης περιοχής. Τα ονόματα περιοχών συμπληρώνονται στους DTN πίνακες προώθησης είτε στατικά (από έναν διαχειριστή δικτύου), είτε από ένα ή περισσότερα δυναμικά DTN πρωτόκολλα προώθησης.

Η ιεραρχική δομή των ονομάτων περιοχών παρέχει την δυνατότητα της μείωσης του μεγέθους των DTN πινάκων προώθησης, με τρόπο αντίστοιχο με την συγκέντρωση των διαδρομών του διαδικτύου στο CIDR [44]. Επίσης σημειώνεται ότι παρά τις φαινομενικά ομοιότητες με τα ονόματα DNS, τα ονόματα περιοχών δεν χρειάζεται απαραίτητα να μεταφράζονται σε οποιοδήποτε είδος διεύθυνσης ή να μεταφράζονται σε μια κατανεμημένη ιεραρχία όπως τα ονόματα DNS.

Όσον αφορά το δεύτερο μέρος (όνομα οντότητας), αυτό αναγνωρίζει ένα όνομα μέσα στη συγκεκριμένη περιοχή, το οποίο δεν χρειάζεται να είναι μοναδικό έξω από αυτήν. Μπορεί να

έχει αυθαίρετη δομή και να περιέχει ειδικά σύμβολα και θα μπορεί να αναγνωριστεί μόνο στην συγκεκριμένη περιοχή.

Για παράδειγμα, στην περίπτωση του διαδικτύου μπορούμε να έχουμε το εξής αναγνωριστικό:

{ internet.int, "http://www.test.gr/overview.html" }

Το αναγνωριστικό αυτό αναφέρεται στην περιοχή του διαδικτύου και συγκεκριμένα σε ένα τοπικό αναγνωριστικό (στην περίπτωσή μας σε ένα URI [45]). Καθώς ένα μήνυμα μεταφέρεται μέσα από ένα σύνολο ανομοιογενών περιοχών, μονό το πρώτο μέρος του αναγνωριστικού χρησιμοποιείται για την δρομολόγηση. Όταν φτάνει στα σύνορα της περιοχής προορισμού, το όνομα της οντότητας μεταφράζεται σε τοπικό επίπεδο σε ένα όνομα ή διεύθυνση κατάλληλη για την συγκεκριμένη περιοχή.

Με την μη επιβολή οποιοδήποτε συγκεκριμένης υποδομής όσον αφορά το δεύτερο μέρος του αναγνωριστικού, μπορεί να χρησιμοποιηθεί οποιοδήποτε σχήμα ονομάτων επιθυμεί η εκάστοτε περιοχή ακόμα και ασυνήθιστα.

Η επιλογή για την υιοθέτηση ονομάτων αντί διευθύνσεων στους συμμετέχοντες αυτού του end-to-end σχήματος προώθησης, προέρχεται από παρατηρήσεις που γίνανε σε πρόσφατες τάσεις που αφορούν διαδικασίες του διαδικτύου. Οι διευθύνσεις χρησιμοποιούνται για δρομολόγηση και αναφέρονται σε υπολογιστικούς πόρους, ενώ η ονομασία προστέθηκε ώστε να κάνει την διευθυνσηδότηση ευκολότερη στους ανθρώπους. Σε πολλές περιπτώσεις ένα όνομα αναφέρεται ουσιαστικά σε ένα query για δεδομένα παρά στην αναγνώριση του συγκεκριμένου υπολογιστικού πόρου που το παρέχει.

Η αρχιτεκτονική DTN έχει ως στόχο δίκτυα, όπου η end-to-end διαδρομή δρομολόγησης δεν μπορεί να εγγυηθεί ότι υπάρχει. Οι διαδρομές αποτελούνται από ένα σύνολο από επαφές εξαρτώμενες του χρόνου γνωστές ως «ευκαιρίες επικοινωνίας», οι οποίες χρησιμοποιούνται για την μετακίνηση των μηνυμάτων προς τους προορισμούς τους. Επιπλέον, οι επαφές παραμετροποιούνται από τους χρόνους έναρξης και τέλους, χωρητικότητα, καθυστέρηση και κατεύθυνση.

Οι λεπτομέρειες της επιλογής μιας διαδρομής και του προγραμματισμού των μηνυμάτων αναμένεται να επηρεαστούν σε μεγάλο βαθμό από τα συγκεκριμένα πρωτόκολλα δρομολόγησης και τους αλγορίθμους της εκάστοτε περιοχής.

Η αρχιτεκτονική DTN περιλαμβάνει τους εξής δύο διακριτούς τύπους κόμβων μηνυμάτων δρομολόγησης:

- persistent (P)
- non-persistent (NP)

Οι P κόμβοι περιέχουν μη αμελητέα ποσότητα από αποθηκευμένα μηνύματα, ενώ οι NP κόμβοι μπορεί να μην περιέχουν. Εκτός αν δεν μπορούν ή δεν θέλουν για κάποιο λόγο να αποθηκεύσουν ένα συγκεκριμένο μήνυμα, οι P κόμβοι γενικά συμμετέχουν σε μεταφορά «υπό επιτήρηση» χρησιμοποιώντας τα κατάλληλα πρωτόκολλα μεταφοράς της περιοχής που βρίσκονται.

Μια μεταφορά «υπό επιτήρηση» αναφέρεται στην αναγνωρισμένη παράδοση ενός μηνύματος από έναν DTN hop στον επόμενο, και την μεταφορά της ευθύνης για αξιόπιστη παράδοση. Η έννοια αυτής της μεταφοράς είναι θεμελιώδης για την αρχιτεκτονική για την αντιμετώπιση τυχόν υψηλών ποσοστών απώλειας και την αφαίρεση από ενδεχομένως φτωχούς σε πόρους τερματικούς κόμβους της ευθύνης για διατήρηση μίας end-to-end επικοινωνίας. Συγκεκριμένα, οι τερματικοί κόμβοι δεν χρειάζεται να κρατούν αντίγραφο των δεδομένων, των οποίων έχουν μεταφερθεί υπό επιτήρηση στον επόμενο DTN κόμβο. Για τους τερματικούς κόμβους που επιμένουν για end-to-end αναγνώριση, μια επιβεβαίωση παράδοσης μπορεί προαιρετικά να ζητηθεί, παρόλο που ο τρόπος αντιμετώπισης αυτής της ένδειξης εξαρτάται από την αιτούσα εφαρμογή.

Η μετάβαση από μία end-to-end αξιόπιστη επικοινωνία σε μια αξιόπιστη προσέγγιση hop σε hop, μπορεί να θεωρηθεί ως βελτιστοποίηση της απόδοσης που περιλαμβάνει την μετακίνηση του τελικού σημείου.

Η DTN αρχιτεκτονική απαιτεί ένα επίπεδο συγχρονισμού χρόνου, το οποίο χρησιμοποιείται για την αναγνώριση των πακέτων μηνυμάτων και επίσης την εκκαθάρισή μηνυμάτων, τα οποία έχουν υπερβεί τον χρόνο ζωής τους. Στις πιο πολλές περιπτώσεις, υπάρχουν πολλαπλά επιπρόσθετα οφέλη, που προέρχονται από την επιβολή μεγαλύτερων περιορισμών στον

συγχρονισμό του χρόνου. Επίσης, δίνοντας σε λογικά πλαίσια ακριβές χρόνο συγχρονισμού, οι DTN τεχνικές διαχείρισης συμφόρησης μπορούν να προβλέψουν με ιδιαίτερη επιτυχία πότε μπορεί να συμβεί μία συμφόρηση.

Ασφάλεια

Οι απαιτήσεις ασφάλειας για την DTN αρχιτεκτονική διαφέρουν κάπως από τα παραδοσιακά μοντέλα ασφαλείας δικτύου στο ότι το σύνολο των αρχών περιλαμβάνει τους δρομολογητές δικτύου (DTN πύλες) για την επικοινωνία με τα τελικά σημεία. Στην περίπτωση του DTN, το ενδιαφέρον βρίσκεται στην επιβεβαιωμένη πρόσβαση που αφορά το φορτίο κίνησης μίας συγκεκριμένης κατηγορίας υπηρεσιών καθώς και στην αποφυγή μεταφοράς φορτίου κίνησης για ενδεχομένως μεγάλες αποστάσεις στις οποίες διαπιστώθηκε αργότερα ότι απαγορεύεται.

Για την υλοποίηση του μοντέλου ασφάλειας, κάθε μήνυμα περιλαμβάνει μία αμετάβλητη «σφραγίδα ταχυδρομείου» που περιέχει ένα εξακριβωμένο αναγνωριστικό του αποστολέα, μία έγκριση από την αιτούσα κατηγορία υπηρεσιών που σχετίζεται με το μήνυμα, και ένα ακόμα κρυπτογραφικό υλικό για την επιβεβαίωση της ακρίβειας του περιεχομένου του μηνύματος. Οι δρομολογητές ελέγχουν τα διαπιστευτήρια σε κάθε DTN hop, και όταν η αυθεντικοποίηση αποτύχει, απορρίπτουν την κίνηση όσον το δυνατόν γρηγορότερα.

Η συγκεκριμένη προσέγγιση χρησιμοποιεί κρυπτογραφία δημόσιου κλειδιού ως σημείο εκκίνησης. Στους δρομολογητές και χρήστες εκδίδονται ζεύγη δημοσίων/ιδιωτικών κλειδιών, και ο εντολέας που στέλνει ένα μήνυμα πρέπει να αποκτήσει ένα υπογεγραμμένο αντίγραφο των δημοσιών κλειδιών από μια αρχή ασφαλείας γνωστή στους DTN προωθητές.

Στον πρώτο DTN δρομολογητή, το υπογεγραμμένο δημόσιο κλειδί χρησιμοποιείται για επικύρωση του αποστολέα και την αιτούμενη κατηγορία υπηρεσιών χρησιμοποιώντας μια λίστα πρόσβασης ελέγχου που αποθηκεύεται στη πύλη. Αποδεκτά μηνύματα μετά ξανά-υπογράφονται με το κλειδί της πύλης για διέλευση. Χρησιμοποιώντας αυτή την προσέγγιση, μόνο οι first-hop πύλες χρειάζεται να κρατάνε στην cache τους πιστοποιητικά ανά χρήστη και μετά μόνο για τους γειτονικούς χρήστες.

Έλεγχος ροής και συμφόρησης

Όντας μια hop-by-hop αρχιτεκτονική, ο έλεγχος ροής και ο έλεγχος συμφόρησης για το DTN είναι στενά συσχετισμένες. Ο έλεγχος ροής αναφέρεται στον περιορισμό του ρυθμού αποστολής από έναν DTN κόμβο στον επόμενο κόμβο. Ο έλεγχος συμφόρησης αναφέρεται στον χειρισμό των διαμαχών για την αποθήκευση των μηνυμάτων σε μία πύλη DTN. Οι διαθέσιμοι μηχανισμοί για να αντιμετωπίζουν τέτοια θέματα, μπορούν να κατηγοριοποιηθούν ως προληπτικοί ή αντιδραστικοί. Οι προληπτικοί μέθοδοι γενικά περιλαμβάνουν κάποια μορφή ελέγχου εισόδου, για την αποφυγή της συμφόρησης σε αρχικό στάδιο. Σε πολλές περιπτώσεις, μια περιοχή μπορεί να είναι κάτω από τον διαχειριστικό έλεγχο μια οντότητας, και αυτή η προσέγγιση μπορεί να είναι πρακτική. Ανεξάρτητα με το αν οι προληπτικοί μέθοδοι είναι αναποτελεσματικοί ή μη διαθέσιμοι, οι αντιδραστικοί μέθοδοι πρέπει να χρησιμοποιηθούν, όταν οι πραγματικές λειτουργικές καθυστερήσεις είναι υψηλές.

Η συγκεκριμένη προσέγγιση χρησιμοποιεί μια κοινόχρηστη ουρά προτεραιότητας για την κατανομή του αποθηκευτικού χώρου. Αρχικά, μηνύματα που έχουν λήξει διαγράφονται. Στη συνέχεια, μηνύματα που φτάνουν και δεν έχουν λήξει και είναι υπερβολικά μεγάλα, τους αρνείται η μεταφορά υπό επιτήρηση. Μετά, τα μηνύματα κατηγοριοποιούνται με βάση την προτεραιότητα και τον χρήσιμο χρόνο ζωής.

Δύο ενδεχομένως προβλήματα που μπορούν να προκύψουν όσον αφορά τα ανωτέρω, είναι ο τύπος αντιστροφής προτεραιότητας και το head-of-line blocking [46]. Παράδειγμα αντιστροφής προτεραιότητας είναι όταν μηνύματα με υψηλή προτεραιότητα που φτάνουν να μην έχουν αποθηκευτικό χώρο διαθέσιμο, όταν προηγουμένως έχουν φτάσει μικρότερης προτεραιότητας μηνύματα.

3.3. UMOBILE

Η όλο και αυξανομένη χρήση του διαδικτύου μέσω κινητών συσκευών, καθώς και η συνεχής ανάπτυξη του Internet Of Things (IoT), έχουν εγείρει αμφιβολίες όσον αφορά την επεκτασιμότητα των λύσεων που προσφέρει η παρούσα αρχιτεκτονική διαδικτύου. Αυτό είχε ως αποτέλεσμα, την έρευνα και ανάπτυξη μιας βελτιωμένης αρχιτεκτονικής NDN, η οποία είναι mobile-centric και service-oriented και ονομάζεται umobile [47].

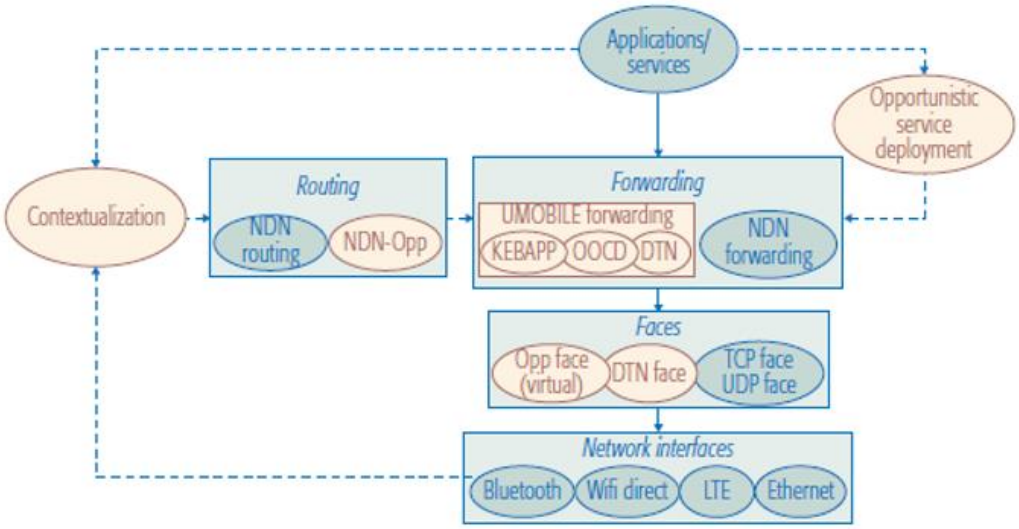
Κύριος στόχος αυτής, είναι η αποτελεσματική παροχή περιεχομένου και υπηρεσιών στους τελικούς χρήστες ακόμα και σε περιοχές που χαρακτηρίζονται ως δύσκολα προσβάσιμες. Το umobile αποσυνδέει τις υπηρεσίες από τις αρχικές τους τοποθεσίες, μετατοπίζοντάς τες σε ένα πρότυπο, το οποίο ενσωματώνει πτυχές τόσο της information-centric όσο και της οπορτουνιστικής δικτύωσης (opportunistic networking), με απώτερο σκοπό την ώθηση των υπηρεσιών του δικτύου και των υπηρεσιών των χρηστών όσο το δυνατόν πλησιέστερα στα άκρα. Με την ώθηση τέτοιων υπηρεσιών κοντά στους χρήστες, πτυχές όπως η χρήση εύρους ζώνης και η διαχείριση πόρων, μπορούν να βελτιστοποιηθούν, ενώ ταυτόχρονα η διαθεσιμότητα υπηρεσιών σε απαιτητικά περιβάλλοντα δικτύου μπορεί να βελτιωθεί.

Όσον αφορά το opportunistic networking, αυτό θεωρείται αναπόσπαστο μέρος της umobile αρχιτεκτονικής, καθώς αυτή στοχεύει κυρίως σε ένα ασύρματο περιβάλλον. Πρόσφατα πρότυπα στο πλαίσιο της ασύρματης δικτύωσης, ενισχύουν τη σχετική device-to-device (D2D) επικοινωνία ως συμπληρωματική στις υπάρχουσες τεχνολογίες πρόσβασης. Αν και δεν είναι ακόμα ώριμο, το WIFI Direct συμπεριλαμβάνεται σε Android OS και υποστηρίζεται από τα περισσότερα Android κινητά. Μια σημαντική αδυναμία των τρεχουσών opportunistic λύσεων είναι το γεγονός ότι η χρησιμότητά τους σε ρεαλιστικά περιβάλλοντα δεν είναι σαφής, καθώς έχουν αξιολογηθεί εκτενώς μόνο μέσω προσομοιωτών και εξομοιωτών. Από την άλλη πλευρά, οι πιο πολλές opportunistic λύσεις, βασίζονται σε μια host-centric προσέγγιση, στέλνοντας πακέτα μεταξύ κόμβων ανεξάρτητα από τα δεδομένα που ανταλλάσσονται.

Umobile Αρχιτεκτονική

Προκειμένου να καλυφθούν οι ελλείψεις της NDN αρχιτεκτονικής (όπως αναλύθηκε σε προηγούμενη ενότητα), γίνανε τροποποιήσεις και βελτιώσεις ώστε να επιτραπεί η υποστήριξη στα άκρα του διαδικτύου, ιδίως για κινητά οπορτουνιστικά ασύρματα περιβάλλοντα.

Αυτό κατέστη εφικτό με τον εντοπισμό των σχετικών απαιτήσεων και την ανάπτυξη διαφόρων μηχανισμών, οι οποίοι κυμαίνονται από νέες στρατηγικές προώθησης προσαρμοσμένες στις συγκεκριμένες ανάγκες, σε έναν οπορτουνιστικό μηχανισμό δρομολόγησης για NDN. Για το σκοπό αυτό, αρχικά αξιοποιήθηκαν όλες οι διαθέσιμες ευκαιρίες επικοινωνίας χρησιμοποιώντας ένα Ethernet interface στο σταθερό τμήμα του δικτύου, καθώς και διασυνδέσεις Wi-Fi/Wi-Fi Direct και LTE στο κινητό μέρος. Επιπλέον, υιοθετείται η αντίληψη περιεχομένου στο πλαίσιο της αρχιτεκτονικής, επιτρέποντας στο δίκτυο να προσαρμοστεί σε διαφορετικές συμπεριφορές χρηστών με την αξιοποίηση πληροφορίων που είναι διαθέσιμες μέσω του wi-fi/wi-fi direct interfaces. Τέλος, γίνεται επέκταση των δυνατότητων των εφαρμογών με την παροχή νέων διεπαφών προγραμματισμού εφαρμογών (API) και σχεδιασμού νέων μηχανισμών QoS για την παροχή υπηρεσιών ανάπτυξης ακόμη και σε απομακρυσμένες, απαιτητικές περιοχές με διαλείπουσα συνδεσιμότητα.



Στη παραπάνω εικόνα, οι κόκκινες ενότητες απεικονίζουν τα συστατικά στοιχεία που έχουν αναπτυχθεί στο πλαίσιο της umobile (και αναλύονται κάτωθι), ενώ οι μπλε απεικονίζουν το αρχικό πλαίσιο NDN.

Forwarding

Ο πυρήνας της προτεινόμενης information-centric επικοινωνίας βασίζεται στη χρήση interests που εκδίδονται από πελάτες ή παρόχους περιεχομένου. Σημειώνεται ότι για την επιτυχή προώθηση, πρέπει να ληφθεί απόφαση σχετικά με το εάν, πότε και σε ποιον πρέπει ένα interest να προωθηθεί. Δεδομένων των ποικιλόμορφων απαιτήσεων της εφαρμογής, καθώς και των

συνθήκων δικτύωσης που μπορεί να εκτείνονται από υψηλής ταχύτητας συνδεσιμότητα σε διαλείπουσες επικοινωνίες, μια συλλογή διαφορετικών στρατηγικών προώθησης πρέπει να υποστηρίζεται από το umobile.

Η προώθηση στο umobile ακολουθεί την by default προώθηση NDN (βέλτιστη διαδρομή, μετάδοση, έλεγχος πελάτη), ενισχυμένη με νέες στρατηγικές προώθησης και μηχανισμούς που επιτρέπουν επικοινωνίες ανεκτές σε καθυστερήσεις και διακοπές. Μια επιλογή ανά namespace για την κατάλληλη στρατηγική, είναι διαθέσιμη για την κάλυψη των αναγκών των εφαρμογών που χρησιμοποιούν διαφορετικά σχήματα ονομασίας.

DTN Tunneling

Μια σημαντική επιλογή προώθησης είναι το DTN tunneling. Για την επίτευξη αυτού, χρησιμοποιήθηκε η υλοποίηση IBR-DTN του bundle πρωτοκόλλου και επίσης η ενίσχυση του NDN με τη δημιουργία ενός νέου DTN face. Διοχετεύοντας πακέτα NDN μέσω του νεοσύστατου DTN face, επεκτείνεται η λειτουργία NDN και παρέχεται δυνατότητα προσέγγισης σε απομακρυσμένες περιοχές στις οποίες δεν υπάρχει τυπική σύνδεση στο διαδίκτυο, καθώς και αξιοπιστία των υπηρεσιών σε αμφισβητούμενα περιβάλλοντα.

Routing

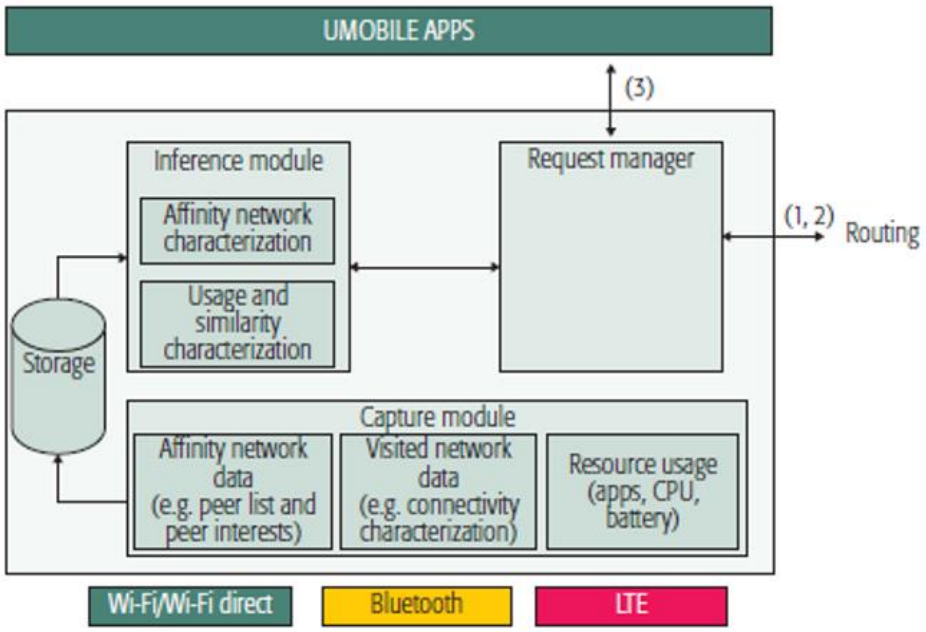
Η δρομολόγηση στο umobile στοχεύει κυρίως opportunistic ασύρματα περιβάλλοντα. Στο πλαίσιο αυτό, έχει αναπτύχθει μια σχετική δρομολόγηση, το NDN framework για opportunistic δικτύα (NDN-Opp), το οποίο είναι σε θέση να αξιοποίησει ασύρματες ευκαιρίες επικοινωνίας. Επίσης, το NDN-Opp παρέχει υλοποίηση της NDN σε Android για opportunistic δίκτυα.

Contextualization

Το umobile αντιμετωπίζει την κοινωνική ευαισθητοποίηση μέσω του context plane, με σκοπό τη καλύτερη διάδοση των πληροφοριών. Η κοινωνική ευαισθητοποίηση έχει αυξηθεί, ιδίως κατά την εξέταση της ικανότητας για εκμετάλλευση της κινητικότητας προσωπικών συσκευών με

στόχο τη μείωση των δεδομένων, καθώς και τη μόχλευση της τοποθεσίας κυκλοφορίας για τη βελτίωση της παροχής υπηρεσιών και περιεχομένου.

Το umobile contextual plane ορίζεται από τον διαχειριστή και αποτελεί μια λειτουργία που εκτελείται στο παρασκήνιο σε έναν τελικό χρήστη ή σε σημείο πρόσβασης. Ο contextual manager (CM) λαμβάνει πληροφορίες σχετικά με το δίκτυο συσχέτισης συσκευών, τις συνήθειες χρήσης και τα ενδιαφέροντα (εσωτερικές πληροφορίες συσκευής). Το κόστος που προκύπτει από το contextualization μεταβιβάζεται, κατόπιν αιτήματος ή περιοδικά, σε άλλες μονάδες umobile για να να βοηθήσει σε διάφορες επιχειρησιακές πτυχές του δικτύου με απώτερο στόχο την επίτευξη αποτελεσματικότερης διάδοσης δεδομένων.



Ενότητα 4: Λειτουργία αναπαράστασης

Για την αναπαράσταση δημιουργήθηκαν δύο Docker containers που επικοινωνούν μεταξύ τους και υλοποιούν τη λογική του πρωτοκόλλου NDN, χρησιμοποιώντας ndn-tools (https://github.com/named-data/ndn-tools).

4.1 Εγκατάσταση Docker

Για την αναπαράσταση χρησιμοποιήθηκε η τεχνολογία του Docker που περιεγράφηκε λεπτομερώς σε παραπάνω ενότητες.

Το Docker παίζει σε Windows, Macintosh και στις διάφορες διανομές του Linux (Ubuntu, Centos, Debian, Fedora κ.λπ.). Στη συγκεκριμένη εργασία παρουσιάζεται η εγκατάσταση σε Microsoft Windows.

Το Docker Desktop είναι η community έκδοση του Docker για τα Microsoft Windows. Η λήψη μπορεί να γίνει από το Docker Hub:
https://hub.docker.com/editions/community/docker-ce-desktop-windows/

Απαιτήσεις Συστήματος

Για να εκτελεστεί με επιτυχία το πρόγραμμα-πελάτη Hyper-V στα Windows 10 υπάρχουν οι εξής απαιτήσεις:

Απαιτήσεις hardware:

- Επεξεργαστής 64 bit
- 4 GB RAM
- Η υποστήριξη εικονικοποίησης υλικού του BIOS πρέπει να είναι ενεργοποιημένη στις ρυθμίσεις του BIOS

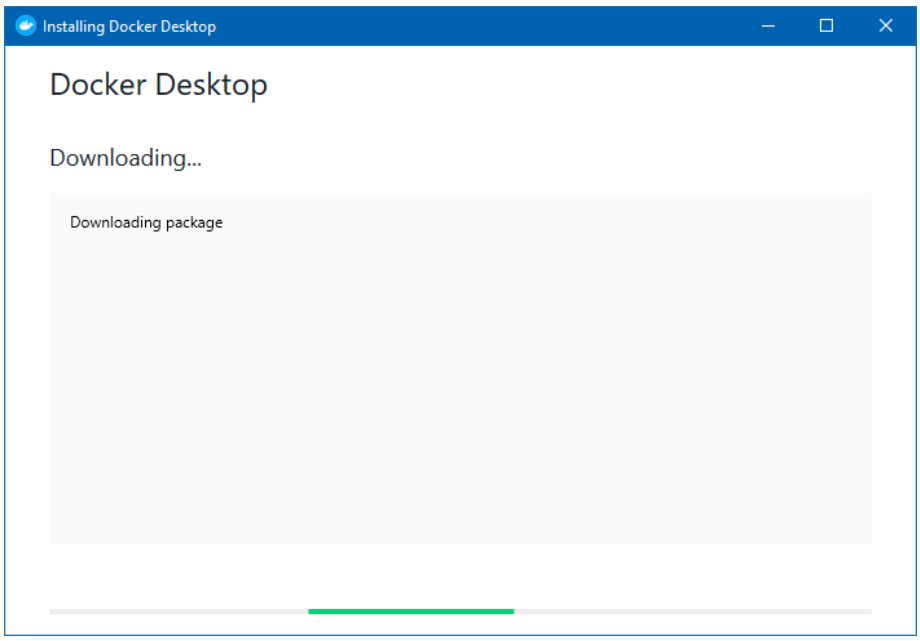
Απαιτήσεις software:

- Windows 10 64-bit: Pro, Enterprise ή Educational (Build 15063 ή νεότερη έκδοση).

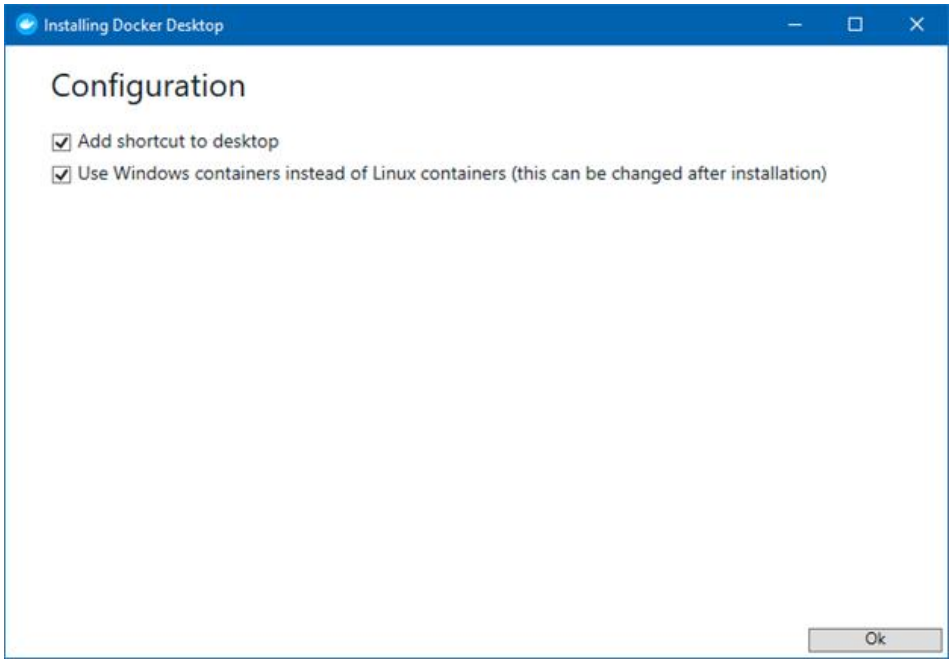
- Hyper-V και Container Οι λειτουργίες των Windows πρέπει να είναι ενεργοποιημένες.

Βήματα Εγκατάστασης

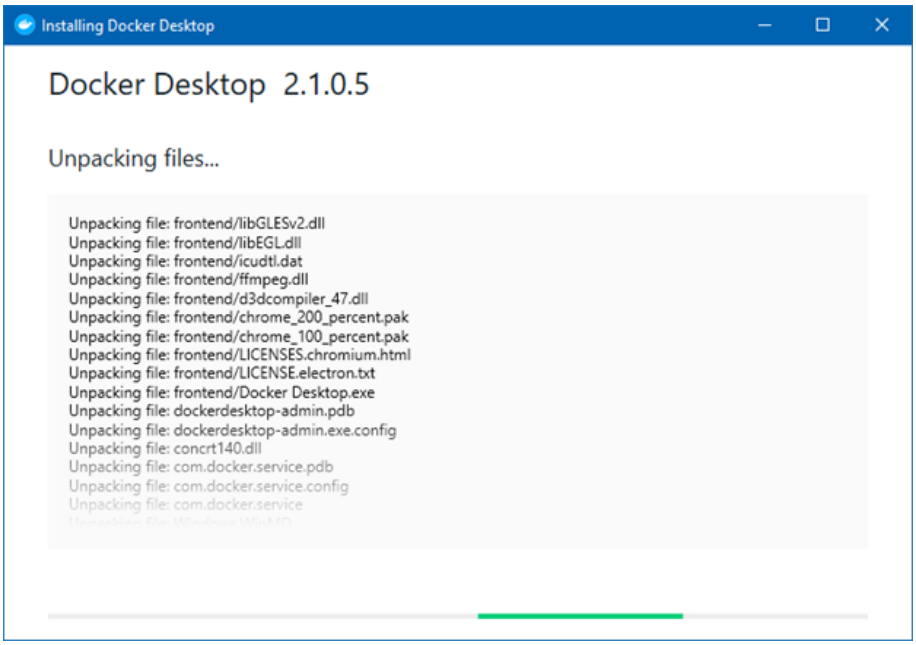
Μόλις κατεβεί το εκτελέσιμο αρχείο από το Docker Hub (Docker Desktop Installer.exe), πατώντας διπλό κλικ αρχίζει η εγκατάσταση.



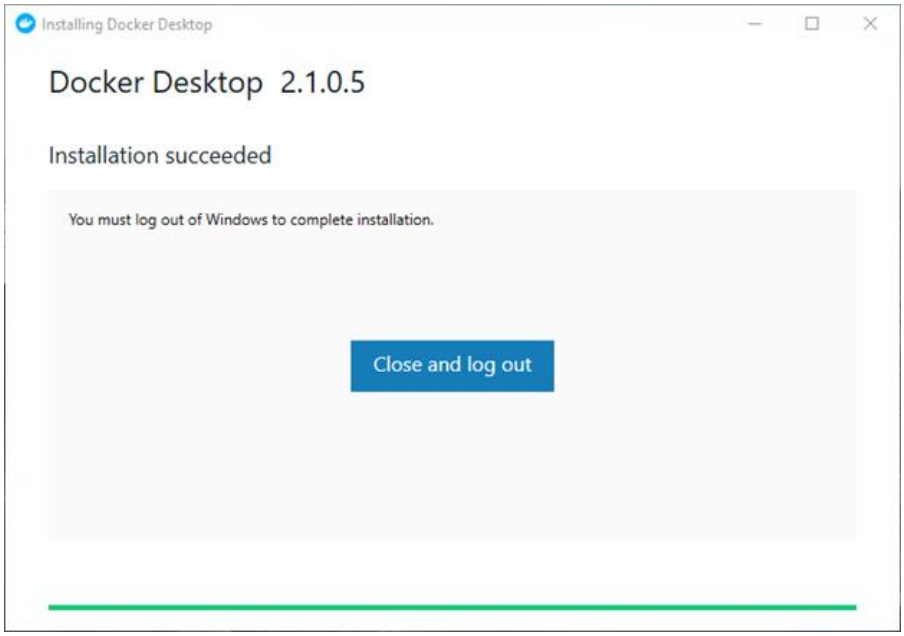
Στην συνέχεια γίνονται οι απαραίτητες ρυθμίσεις:



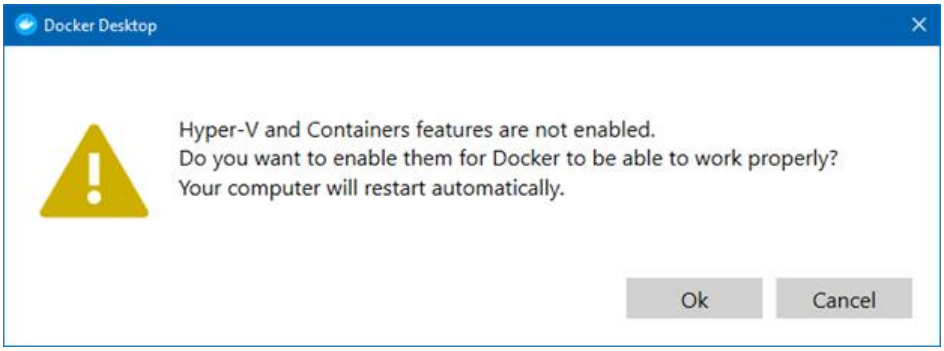
Αφού γίνουν οι ρυθμίσεις, τα αρχεία γίνονται unpack:



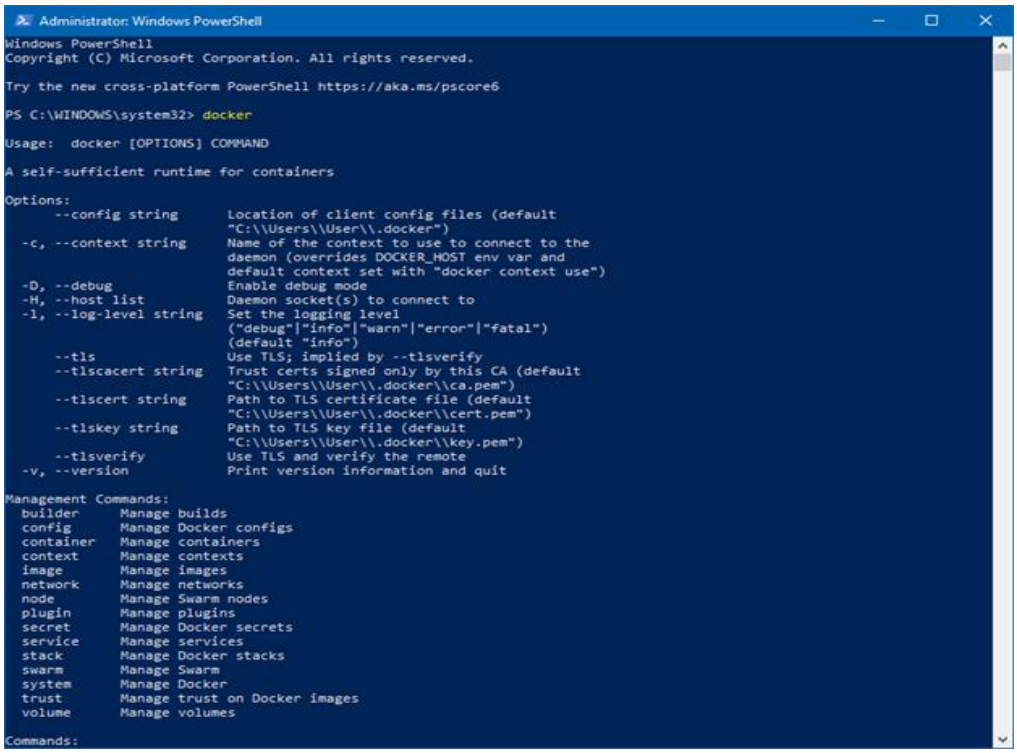
Όταν τελειώσει η εγκατάσταση εμφανίζεται το κάτωθι μήνυμα:



Την πρώτη φορά που θα τρέξει το Docker, σε περίπτωση που οι λειτουργίες Hyper-V και Container δεν έχουν ενεργοποιηθεί, ένα παράθυρο θα εμφανιστεί ειδοποιώντας τον χρήστη ότι πρέπει να ενεργοποιηθούν έτσι ώστε να παίζει σωστά.



Επίσης πρέπει να γίνει επιβεβαίωση ότι η εντολή «docker» είναι διαθέσιμη από τη γραμμή εντολών:



4.2 Δημιουργία Docker εικόνας

DockerFile

Για την δημιουργία της docker εικόνας φτιάχτηκε ένα αρχείο DockerFile. Το DockerFile είναι ένα έγγραφο κειμένου που περιέχει όλες τις εντολές που ο χρήστης μπορεί να εκτελέσει στην γραμμή εντολών για να φτιάξει μια εικόνα. Η χρήση της εντολής «docker build» δημιουργεί μια αυτοματοποιημένη διαδικασία που εκτελεί διαδοχικά πολλαπλές εντολές.

Το DockerFile που δημιουργήθηκε είναι το παρακάτω:

```
from ubuntu:14.04

RUN apt-get update
```

```
RUN apt-get install -y devscripts build-essential cdbs pkg-config debhelper autotools-dev libnl-3-dev libnl-genl-3-dev libnl-route-3-dev libnl-nf-3-dev libnl-cli-3-dev libssl-dev libssl-dev zlib1g-dev libsqlite3-dev libcurl4-openssl-dev libdaemon-dev libvmime-dev libarchive-dev automake autoconf pkg-config libtool libcppunit-dev

COPY ibrcommon-1.0.1.tar.gz /root/
COPY ibrdtn-1.0.1.tar.gz /root/
COPY ibrdtnd-1.0.1.tar.gz /root/
COPY ibrdtn-tools-1.0.1.tar.gz /root/

RUN cd /root && \
    tar -xvf ibrcommon-1.0.1.tar.gz && \
    cd ibrcommon-1.0.1 && \
    ./configure --with-openssl && \
    make && \
    make install && \
    ldconfig

RUN cd /root && \
    tar -xvf ibrdtn-1.0.1.tar.gz && \
    cd ibrdtn-1.0.1 && \
    ./configure && \
    make && \
    make install && \
    ldconfig

RUN cd /root && \
    tar -xvf ibrdtnd-1.0.1.tar.gz && \
    cd ibrdtnd-1.0.1 && \
    ./configure --with-curl && \
    make && \
    make install && \
    ldconfig

RUN cd /root && \
    tar -xvf ibrdtn-tools-1.0.1.tar.gz && \
    cd ibrdtn-tools-1.0.1 && \
    ./configure && \
```

```
    make && \
    make install && \
    ldconfig

COPY ./dnd.conf /root/

RUN apt-get install -y vim build-essential libcrypto++-dev libsqlite3-dev libboost-all-dev libssl-dev pkg-config libpcap-dev doxygen graphviz python-sphinx

COPY ./ndn-cxx_umobile.tar.gz /root/
COPY ./ndn-dtn.tar.gz /root/

RUN cd /root && \
    tar -xvf ndn-cxx_umobile.tar.gz && \
    cd ndn-cxx_umobile && \
    ./waf configure && \
    ./waf && \
    ./waf install

RUN sudo apt-get install -y curl

RUN cd /root && \
    tar -xvf ndn-dtn.tar.gz && \
    cd ./ndn-dtn && \
    mkdir websocketpp && \
    curl -L https://github.com/zaphoyd/websocketpp/archive/0.5.1.tar.gz > websocket.tar.gz && \
    tar zxf websocket.tar.gz -C websocketpp/ --strip 1 && \
    ./waf configure && \
    ./waf && \
    ./waf install

RUN ldconfig
COPY nfd.conf /usr/local/etc/ndn/

ADD ./ndn-tools-ndn-tools-0.3 /root/ndn-tools-ndn-tools-0.3

RUN cd /root/ndn-tools-ndn-tools-0.3 && \
    ./waf configure && \
```

```
    ./waf && \
    ./waf install

ENTRYPOINT /bin/bash
```

Η εικόνα βασίζεται στην διανομή του Linux, Ubuntu και συγκεκριμένα στην έκδοση 14.04. Ο κώδικας για τα εργαλεία NDN - DTN που χρειάζεται να εγκατασταθούνε λειτουργεί σωστά σίγουρα σε αυτήν την έκδοση.

```
from ubuntu:14.04
```

Απαιτείται η εγκατάσταση απαιτούμενων βιβλιοθηκών και εργαλείων χρησιμοποιώντας τον χειρισμό πακέτων του Ubuntu (apt-get).

```
RUN apt-get install -y devscripts build-essential cdbs pkg-config debhelper autotools-dev libnl-3-dev libnl-genl-3-dev libnl-route-3-dev libnl-nf-3-dev libnl-cli-3-dev libssl-dev libssl-dev zlib1g-dev libsqlite3-dev libcurl4-openssl-dev libdaemon-dev libvmime-dev libarchive-dev automake autoconf pkg-config libtool libcppunit-dev
```

Στη συνέχεια πρέπει να εγκατασταθούν κάποια άλλα components που δεν γίνεται μέσω apt-get. Αυτά αφού ληφθούν από τις αντίστοιχες πηγές, πρέπει να αποθηκευτούν στο ίδιο path που βρίσκεται το dockerFile και θα αντιγραφούν μέσα στο Ubuntu χρησιμοποιώντας την εντολή «COPY» στο path «/root/»

```
COPY ibrcommon-1.0.1.tar.gz /root/
COPY ibrdtn-1.0.1.tar.gz /root/
COPY ibrdtnd-1.0.1.tar.gz /root/
COPY ibrdtn-tools-1.0.1.tar.gz /root/
```

Στη συνέχεια, θα πρέπει να γίνει εγκατάσταση τους εκτελώντας την εντολή «RUN» με συγκεκριμένη παραμετροποίηση.

```
RUN cd /root && \
    tar -xvf ibrcommon-1.0.1.tar.gz && \
    cd ibrcommon-1.0.1 && \
    ./configure --with-openssl && \
    make && \
```

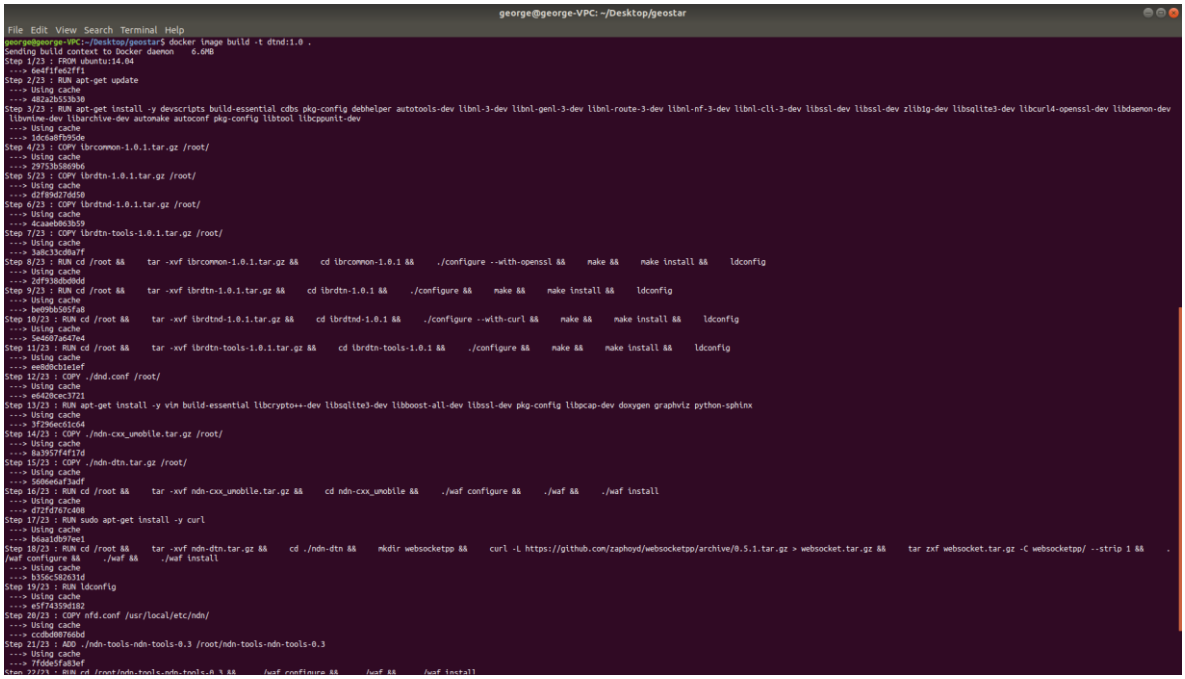
```
make install && \
ldconfig
```

Χτίσιμο Docker Image

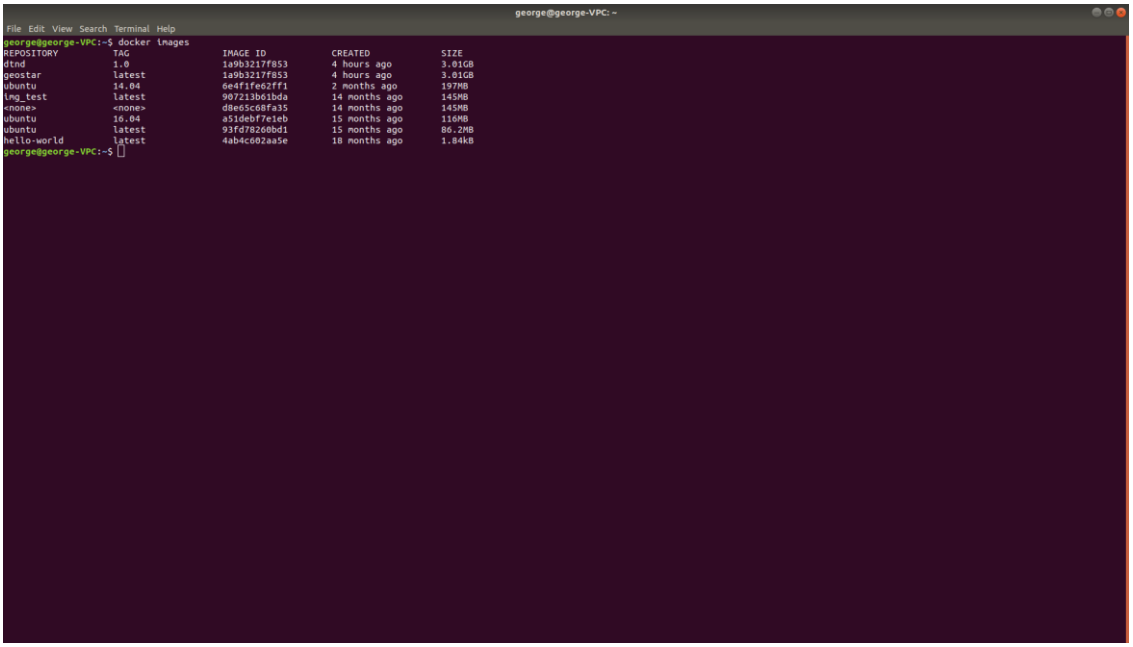
Αφού βεβαιωθούμε ότι το Docker τρέχει στο τοπικό σύστημα και βρισκόμαστε στο path που βρίσκεται το docker file, η εικόνα χτίζεται χρησιμοποιώντας την εντολή:

```
docker image build -t dtnd:1.0 .
```

Συγκεκριμένα δίνεται ένα όνομα και ένα tag στην εικόνα που επρόκειτο να δημιουργηθεί (dtnd:1.0).



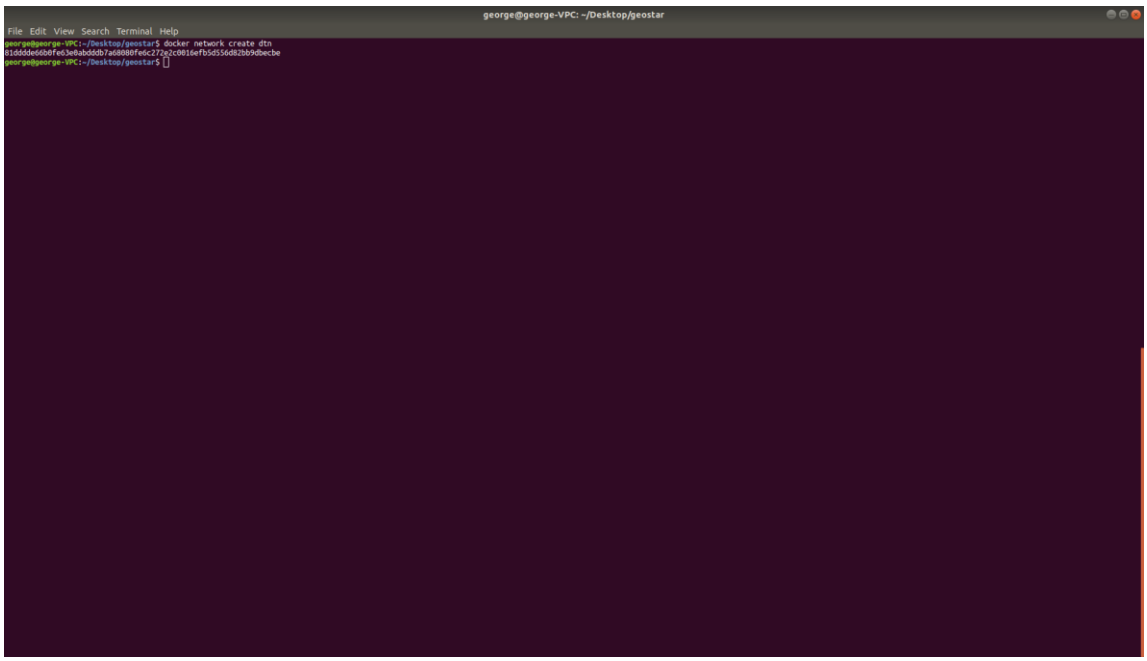
Χρησιμοποιώντας την εντολή «docker images» μπορούμε να δούμε τις εικόνες που έχουν δημιουργηθεί.



4.3 Δημιουργία δικτύου Docker

Για την υλοποίηση της εργασίας απαιτείται ένα container που θα λειτουργεί ως «server» και ένα δεύτερο που λειτουργεί ως «client». Για τον λόγο αυτό πρέπει να δημιουργηθεί ένα “bridge” δίκτυο μεταξύ των containers. Κάθε φορά που ένα νέο container ξεκινάει με την εντολή «docker run», συνδέεται αυτόματα στο bridge δίκτυο. Η δημιουργία του δικτύου γίνεται χρησιμοποιώντας την εντολή:

«docker network create dtn»



4.4 Δημιουργία Containers

Αφού δημιουργήθηκε και το δίκτυο θα πρέπει να δημιουργηθούν και να τρέξουν τα δύο containers. Το καθένα έχει διαφορετικό όνομα που αντικατοπτρίζει την ιδιότητα του (client ή server). Επίσης πρέπει να συνδεθεί στο δίκτυο που φτιάχτηκε προηγουμένως και να γίνει προσβάσιμο στους containers το σύστημα αρχείων που περιέχει τα αρχεία που θα χρειαστούν για την παραμετροποίηση.

Αφού θα λειτουργεί το ένα ως client και το άλλο ως server για την επικοινωνία μέσω του πρωτοκόλλου NDN δημιουργούνται χρησιμοποιώντας διαφορετική παραμετροποίηση. Έχουν φτιαχτεί δύο αρχεία για την παραμετροποίηση αυτή:

- server.conf
- clinet.conf

Η διαφορά ανάμεσα σε αυτά τα 2 αρχεία είναι το local URI που είναι το όνομα που χρησιμοποιεί το NFD για προώθηση πακέτων ενδιαφέροντος/δεδομένων στον daemon. Συγκεκριμένα για το container που θα τρέξει ως server έχει τιμή «dtn://server.dtn» και για το container που θα τρέξει ως client έχει τιμή «dtn://client.dtn».

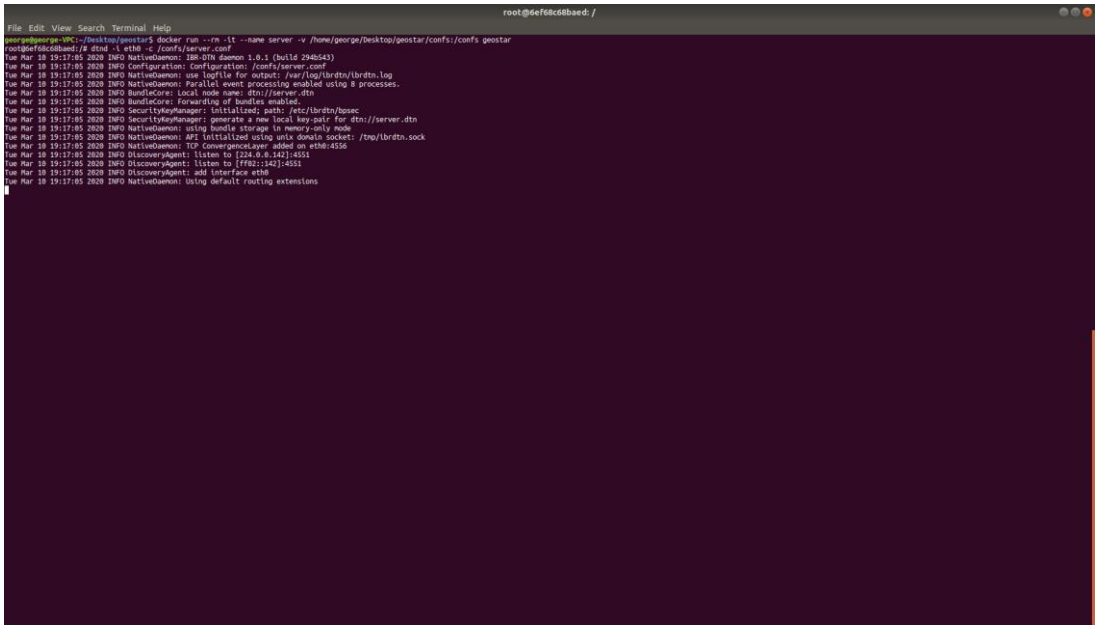
Για την δημιουργία και την εκκίνηση του container που θα λειτουργεί ως server, εκτελείται η εντολή:

docker run –rm -it –net=dtn –name server -v /home/george/Desktop/geostar/confs:/confs geostar

Στη συνέχεια εκτελώντας την εντολή:

«dtnd -i eth0 -c /confs/server.conf»

τρέχει ο daemon με τη συγκεκριμένη παραμετροποίηση.

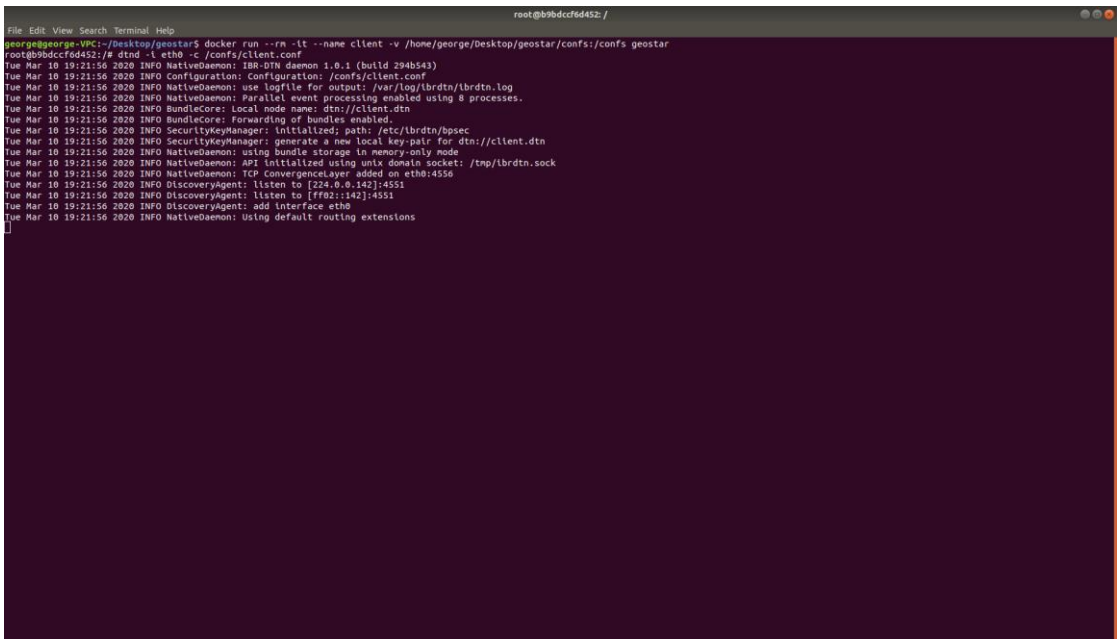


Αντίστοιχα και για τον container-client εκτελείται η εντολή:

docker run –rm -it –net=dtn –name client -v /home/george/Desktop/geostar/confs:/confs geostar

και στη συνέχεια τρέχει ένας νέος daemon με την παραμετροποίηση για τον client με την εντολή:

«dtnd -I eth0 -c /confs/client.conf»



4.5 Επικοινωνία μέσω DTN πρωτοκόλλου

Οι δοκιμές έχουν γίνει χρησιμοποιώντας εργαλεία και βιβλιοθήκες του λογισμικού IBR-DTN (https://github.com/ibrdtn/ibrdtn/wiki/IBR-DTN-Tutorial:-a-step-by-step-guide)

Εκτελώντας την εντολή «docker exec» μια νέα εντολή εκτελείται σε έναν τρέχων container της επιλογής μας. Συγκεκριμένα για τον container-server εκτελείται η εντολή

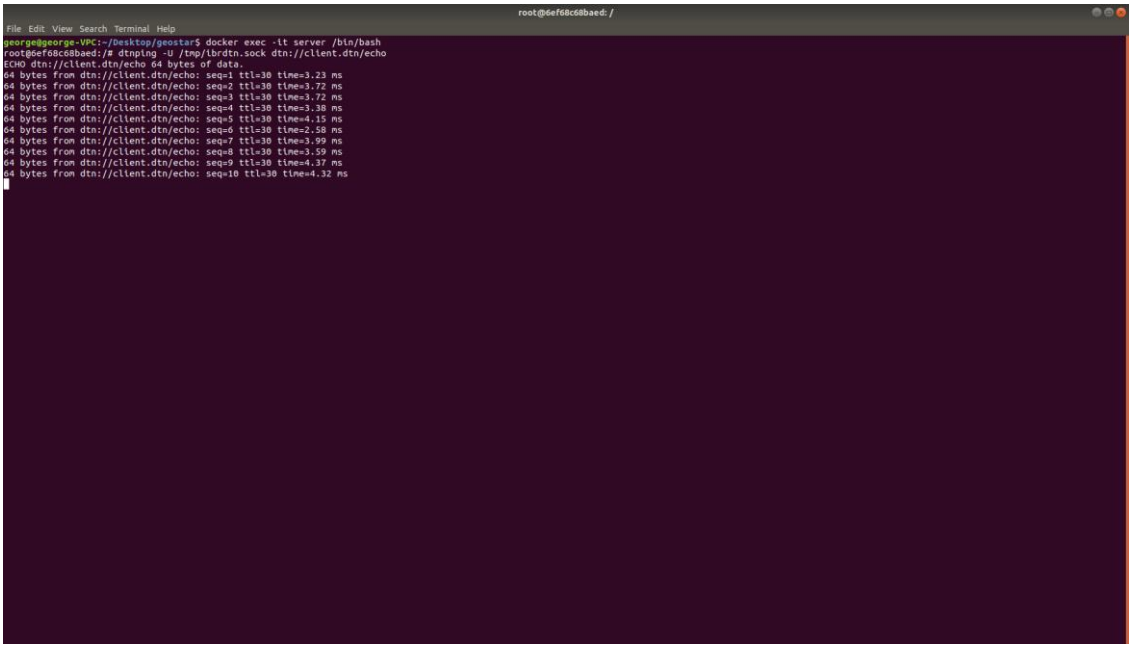
«docker exec -it server /bin/bash».

Δοκιμή Σύνδεσης (Ping)

Θέλουμε τώρα να δούμε αν μπορεί ο server container να επικοινωνήσει με τον client, δηλαδή να δοκιμάσουμε την σύνδεση από την μεριά του server. Αυτό επιτυγχάνεται χρησιμοποιώντας την εντολή:

«dtnping -U /tmp/ibrdtn.sock dtn://client.dtn/echo».

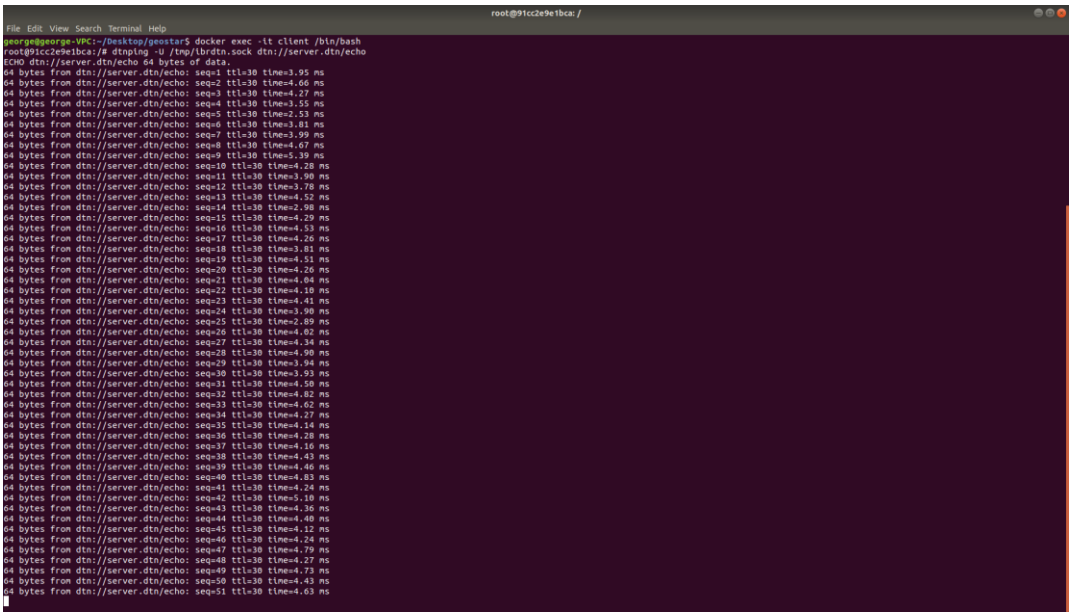
Όπως φαίνεται και από την παρακάτω εικόνα η σύνδεση είναι επιτυχής καθώς τέσσερα πακέτα παρελήφθησαν:



Αντίστοιχα στον container-client εκτελείται η εντολή «docker exec -it client /bin/bash» ώστε στη συνέχεια να τρέξουμε την εντολή:

«dtnping -U /tmp/ibrdtn.sock dtn://server.dtn/echo».

Και εδώ η σύνδεση είναι επιτυχής.



Μεταφορά αρχείων

Χρησιμοποιώντας τις εντολές «dtnsend» και «dtnrecv» μπορούμε να στείλουμε αρχεία μεταξύ των DTN κόμβων.

Αρχικά δημιουργούμε ένα test αρχείο στον server με την εντολή:

«echo GEOSTAR THESIS! > sendfile»

Στην συνέχεια γίνεται register ένας receiver στον client:

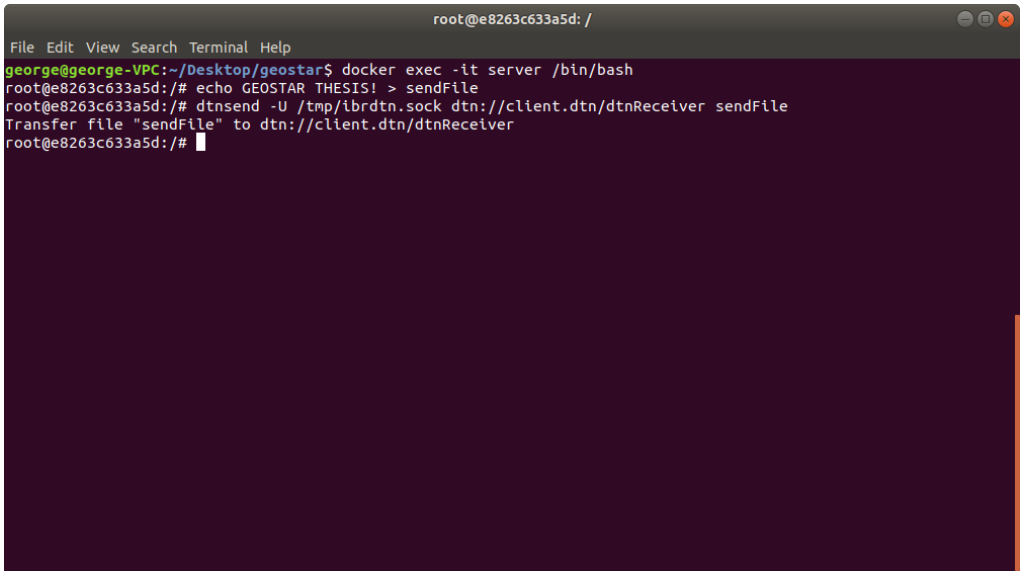
«dtnrecv -U /tmp/ibrdtn.sock –name dtnReceiver».

Αφού γίνει αυτό, ο server στέλνει το αρχείο στον client:

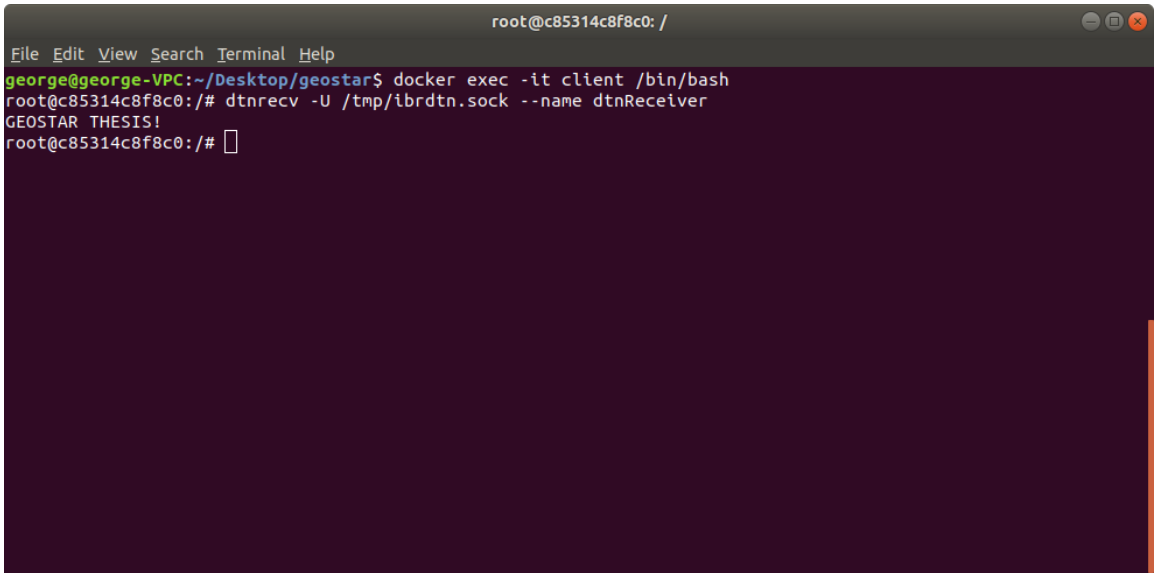
«dtnsend -U /tmp/ibrdtn.sock dtn://client.dtn/dtnReceiver sendFile»

και ο client τυπώνει το περιεχόμενο του αρχείου.

Server:



Client:



4.6 Επικοινωνία NDN-over-DTN

Στην παραπάνω υπό-ενότητα, υλοποιήθηκε η επικοινωνία ανάμεσα σε δύο containers μέσω DTN. Σε αυτή την υπό-ενότητα, παρουσιάζεται η υλοποίηση μίας τοπολογίας, στην οποία δύο containers (umobile1 και umobile2) επικοινωνούν μέσω NDN over DTN και στη μέση υπάρχει ένας τρίτος container (intermediate), ο οποίος επικοινωνεί με τους άλλους δύο μέσω DTN.

(umobile1)---------(intermediate)---------(umobile2)
NDN-DTN DTN NDN-DTN

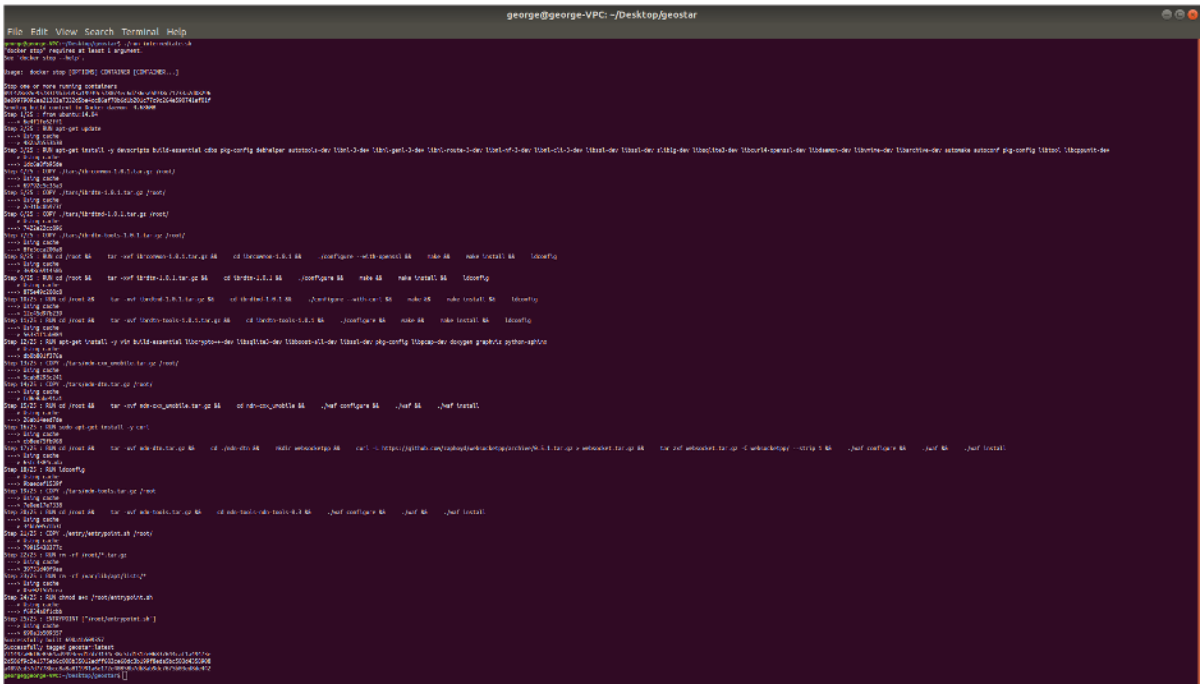
Αρχικά, έχει δημιουργηθεί ένα αρχείο script “intermediate.sh”, το οποίο περιέχει ένα σύνολο από εντολές για τη:

- δημιουργία των απαραίτητων δικτύων, containers και εικόνων εφόσον δεν υπάρχουν
- παραμετροποίηση των containers
- σύνδεση του intermediate container στο network umobile1

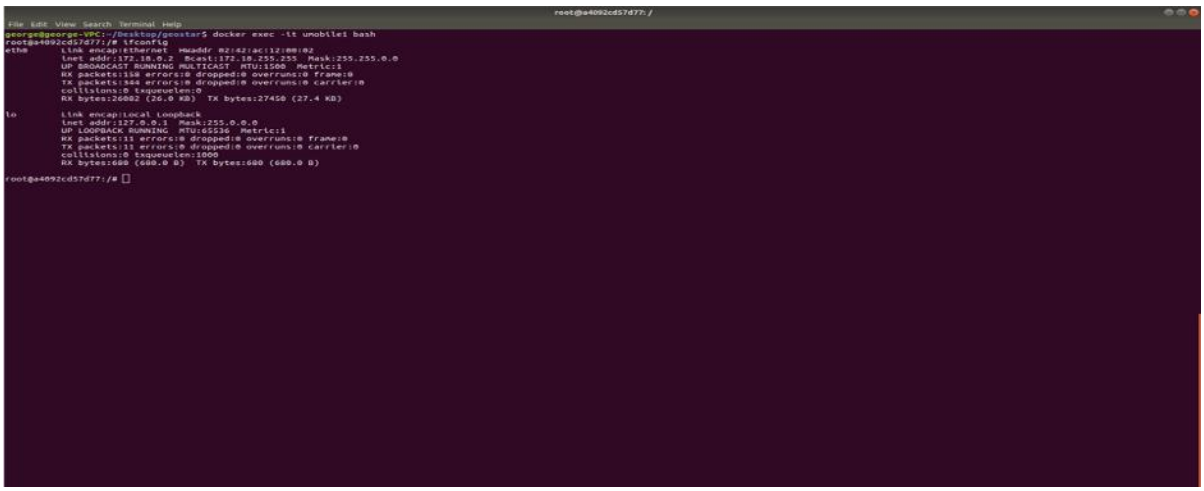
Το αρχείο φτιάχτηκε με σκοπό την αυτοματοποίηση των βασικών διεργασιών, που πρέπει να γίνουν.

```
Open ▾   run_intermediate.sh   ~/Desktop/geostar   Save

#!/bin/bash

name=$1

# stop any other containers with the same name
docker stop $1

# check if containers networks exist and create them, if they do not.
docker network create --subnet=172.18.0.0/16 umobile1
docker network create --subnet=172.20.0.0/16 umobile2

# make sure our base image is already build.
docker build . -t geostar

# Run umobile2
docker run --rm -dit --net umobile2 --ip 172.20.0.2 --entrypoint bash  --name umobile2 -v ${PWD}/confs_intermediate:/confs geostar

docker exec -dit umobile2 /bin/bash -c 'dtnd -i eth0 -c /confs/dtn_umobile2.conf& cp /confs/nfd_umobile2.conf /usr/local/etc/ndn/nfd.conf &&
nfd-start & wait;'

# Run intermediate
docker run --rm -dit --net umobile2 --entrypoint bash --name intermediate -v ${PWD}/confs_intermediate:/confs geostar


# Run umobile1
docker run --rm -dit --net umobile1 --ip 172.18.0.2 --entrypoint bash  --name umobile1 -v ${PWD}/confs_intermediate:/confs geostar

docker exec -dit umobile1 /bin/bash -c 'dtnd -i eth0 -c /confs/dtn_umobile1.conf& cp /confs/nfd_umobile1.conf /usr/local/etc/ndn/nfd.conf &&
nfd-start & wait;'


docker network connect umobile1 intermediate

docker exec -dit intermediate /bin/bash -c 'dtnd -c /confs/dtn_intermediate.conf& wait;'

sh ▾   Tab Width: 8 ▾   Ln 4, Col 1 ▾   INS
```

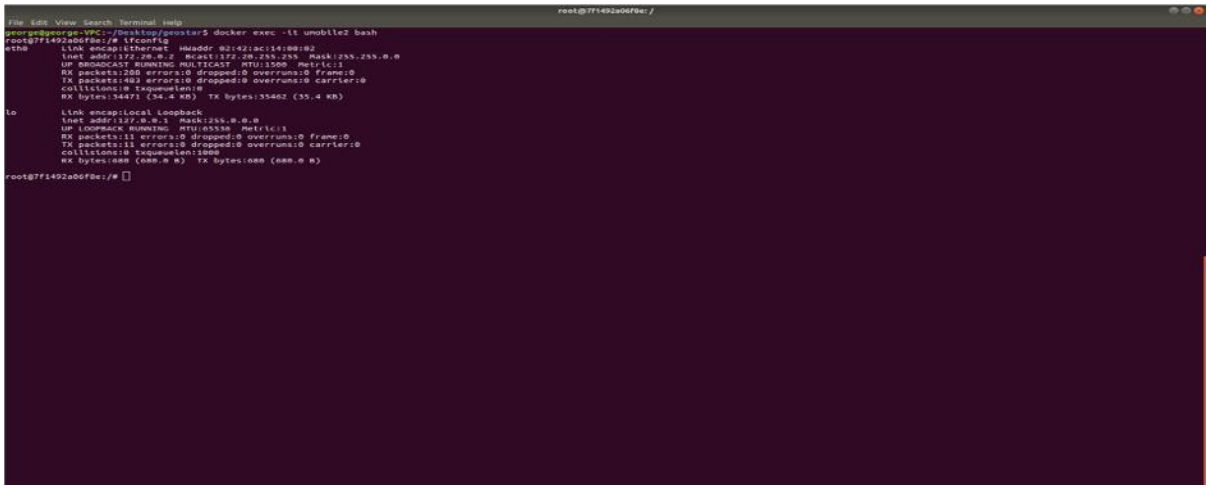
Τρέχοντας την εντολή «sh intermediate.sh”, το παραπάνω αρχείο script εκτελείται:



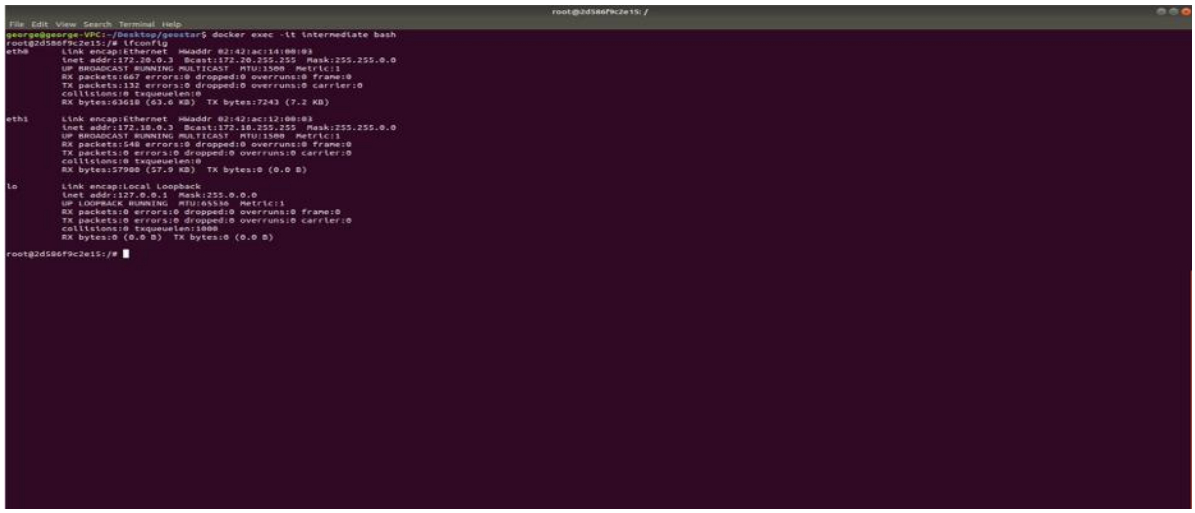
Αφού εκτελεστεί η εντολή «docker exec -it umobile1 bash», τρέχοντας την εντολή «ifconfig» η IP του container umobile1 εμφανίζεται.



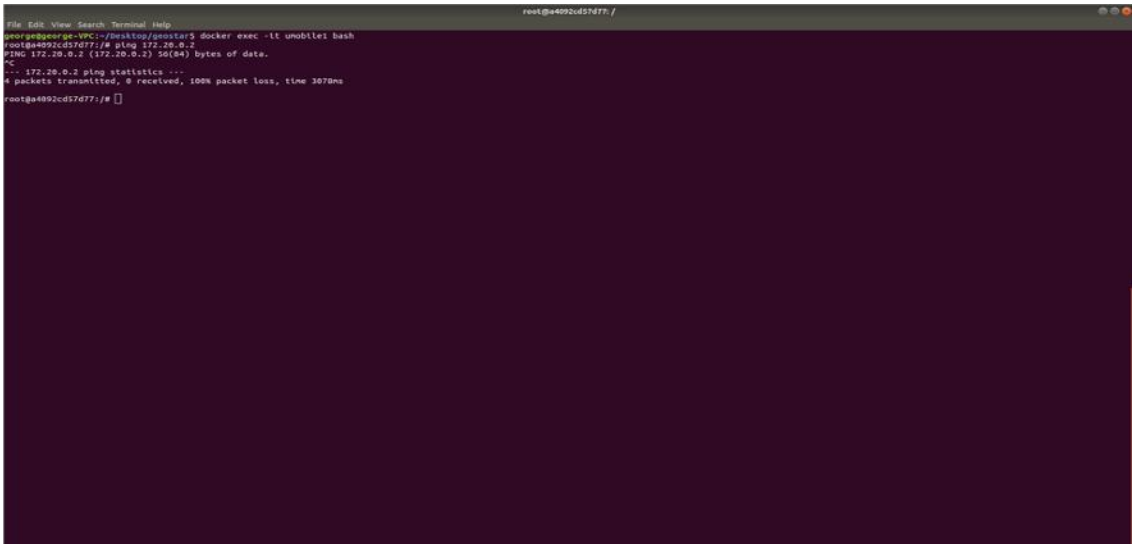
Παρομοίως, εκτελώντας στον container umobile2 τις εντολές «docker exec -it umobile2 bash» και «ifconfig» εμφανίζεται η IP του container.



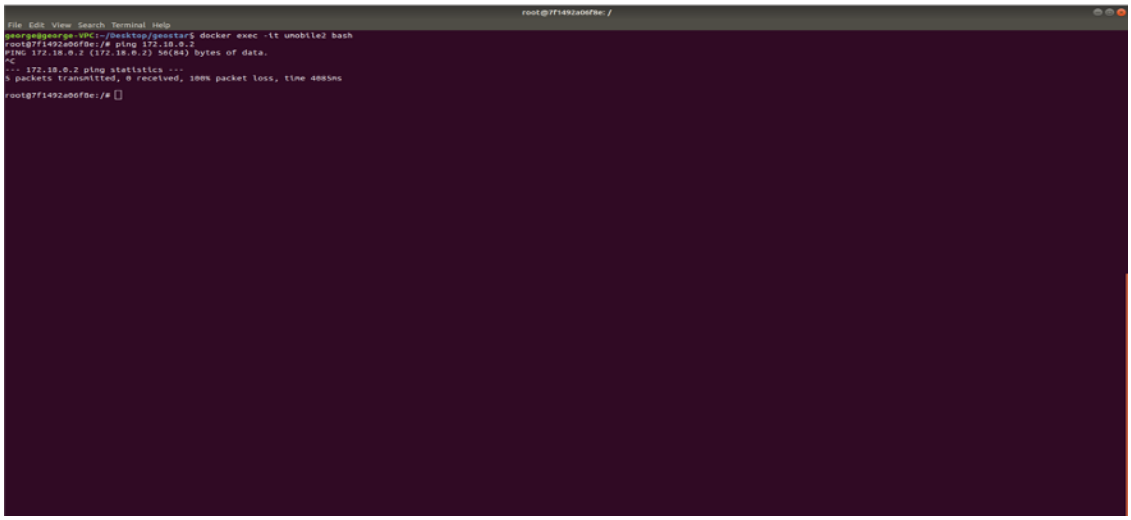
Αντίστοιχα και στον intermediate container με τις εντολές «docker exec -it intermediate bash» και «ifconfig».



Στον container umobile1, εκτελούμε την εντολή «ping 172.20.0.2». Το αναμενόμενο εδώ είναι να μην επικοινωνεί με από ping με τον container umobile2. (η IP που χρησιμοποιήθηκε στην εντολή αντιστοιχεί στον umobile2).



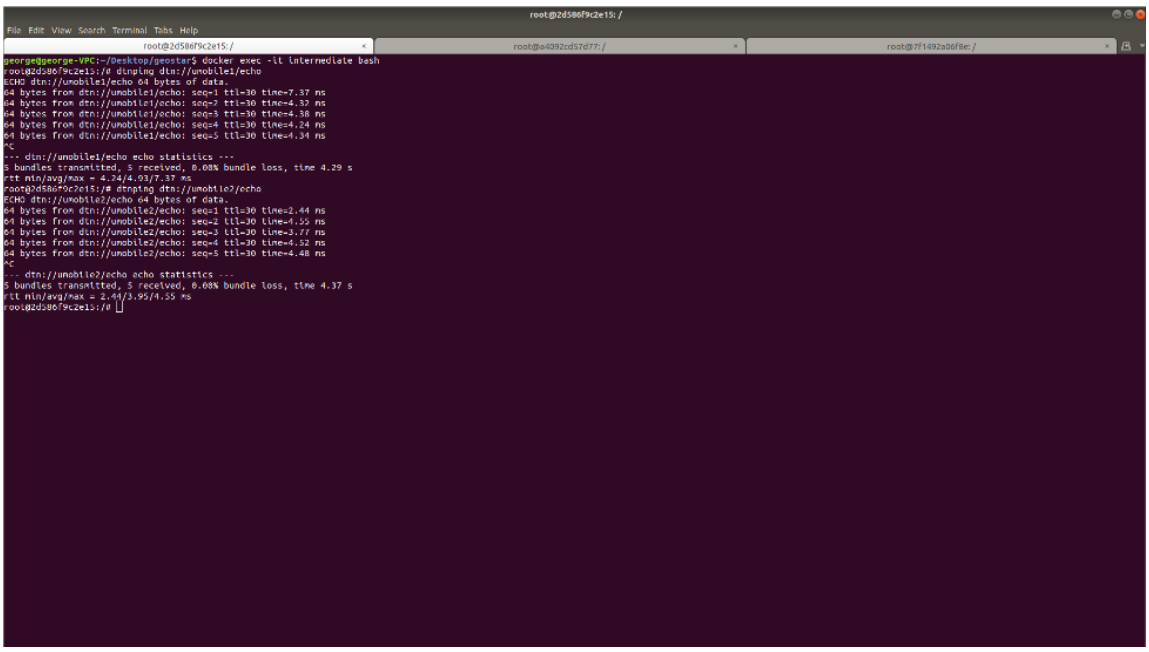
Στον container umobile2 τρέχουμε αντίστοιχα την εντολή «ping 172.18.0.2».



Με την χρήση της εντολής «dtnping» δείχνουμε ότι μπορεί να επικοινωνήσει ο container intermediate με τους άλλους δύο containers. Συγκεκριμένα οι παρακάτω εντολές χρησιμοποιούνται:

dtnping dtn ://umobile1/echo

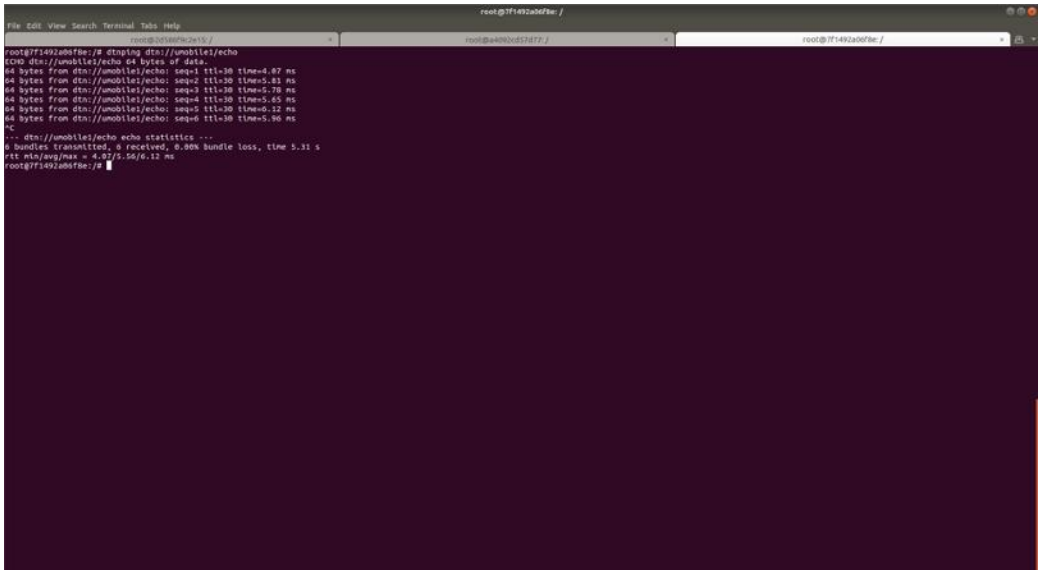
dtnping dtn://umobile2/echo

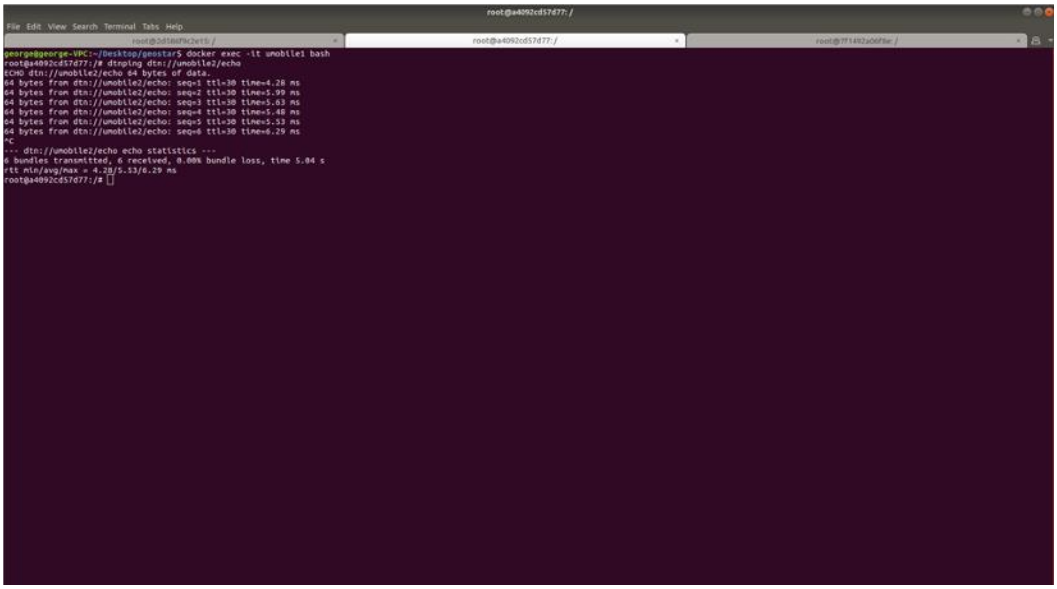


Χρησιμοποιώντας πάλι την εντολή «dtnping» δείχνουμε ότι μπορεί να επικοινωνήσουν μεταξύ τους οι containers umobile1 kai umobile2 με το πρωτόκολλο DTN. Συγκεκριμένα οι παρακάτω εντολές χρησιμοποιούνται:

dtnping dtn://umobile2/echo

dtnping dtn://umobile1/echo

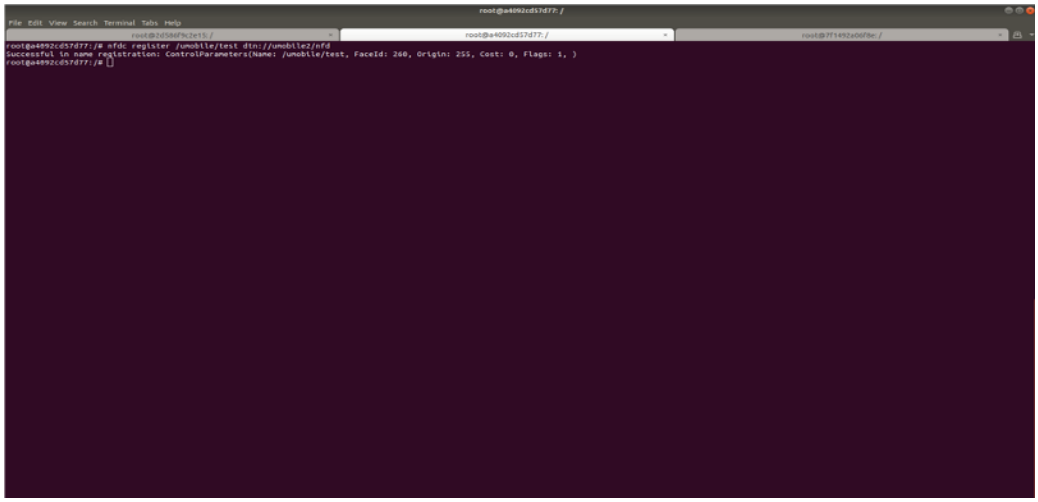




Από τον container umobile1 τρέχουμε την εντολή:

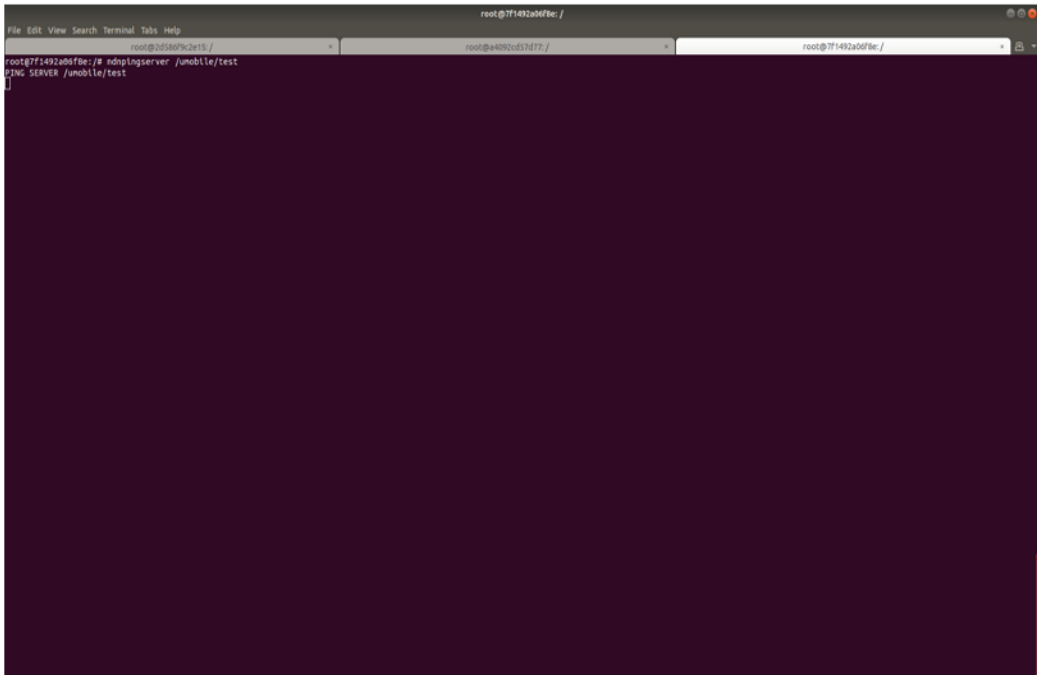
«nfdc register umobile/test dtn://umobile2/nfd»

Συγκεκριμένα, το nfdc είναι ένα εργαλείο που μας επιτρέπει να επηρεάζουμε τα routing information base (RIB), forwarding information base (FIB) και StrategyChoices πίνακα. Η εντολή «register» είτε προσθέτει καινούργια δρομολόγηση είτε ενημερώνει κάποια υπάρχουσα.



Στον container umobile2, τρέχουμε την εντολή:

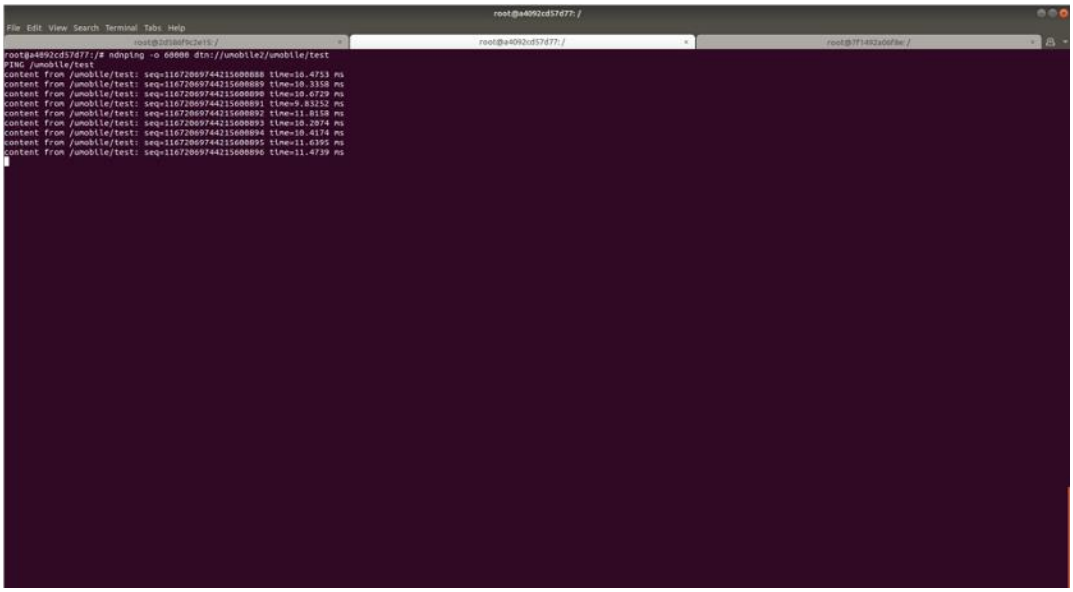
ndnpingserver *umobile/*test

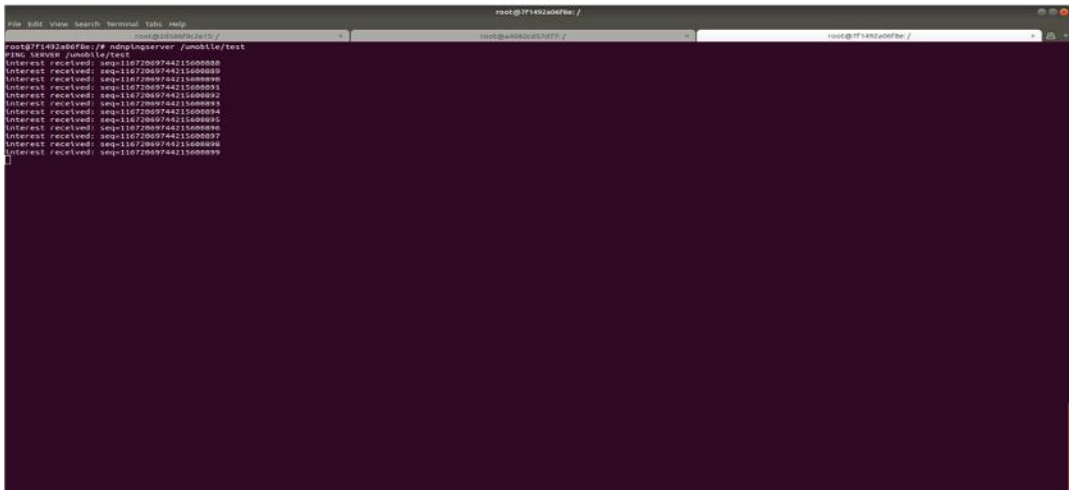


Στον container umobile1 τρέχουμε την εντολή

ndnping -o 60000 dtn://umobile2/umobile/test

Οι δύο container πλέον επικοινωνούν μεταξύ τους με NDN over DTN.

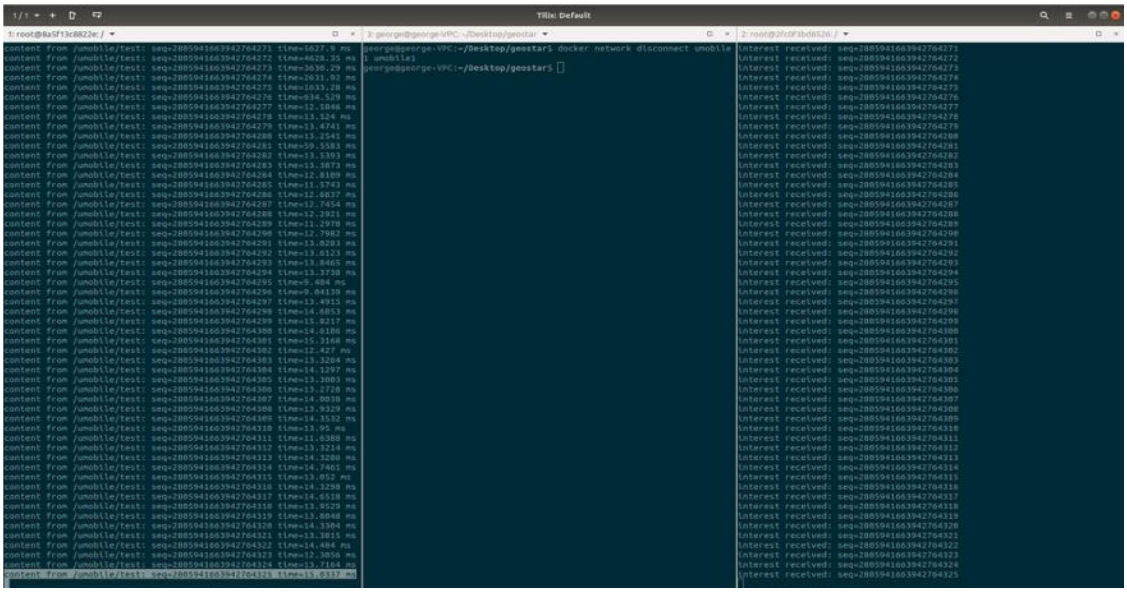


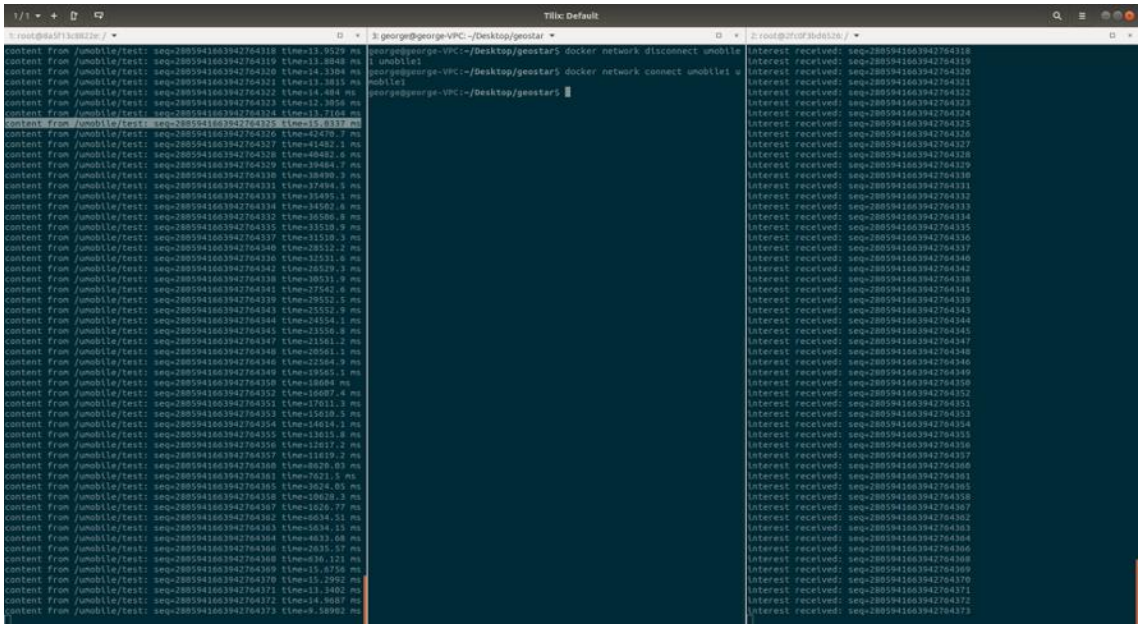


Μπορούμε να αποσυνδέσουμε και να συνδέσουμε αντίστοιχα τον container umobile1 από το δίκτυο umobile1 χρησιμοποιώντας τις εντολές:

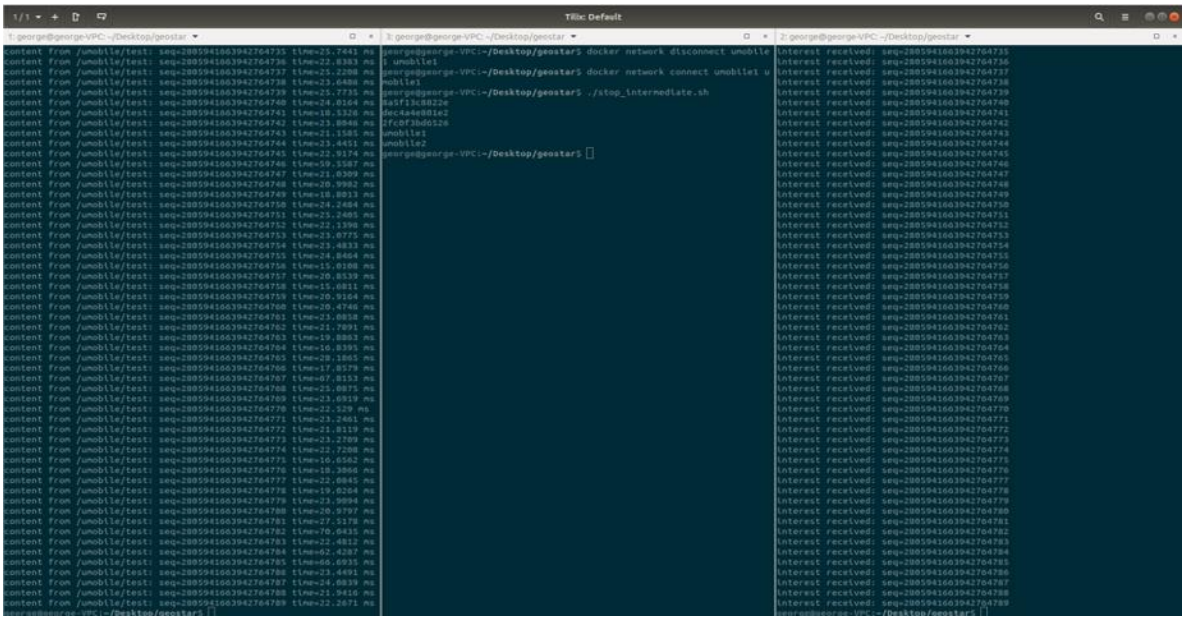
docker network disconnect umobile1 umobile1
docker network connect umobile1 umobile1

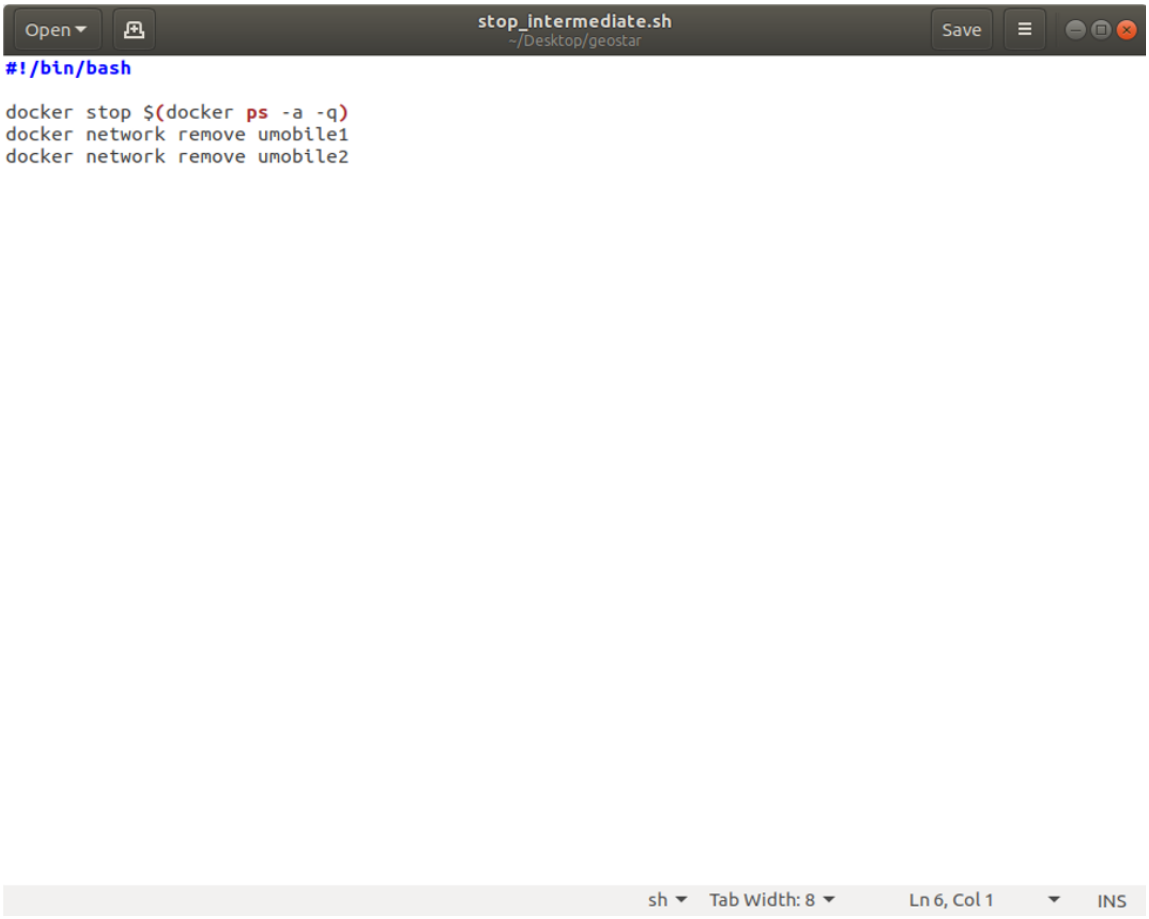
Το περιεχόμενο που θα έστελνε ο container umobile1 το χρόνο που είναι αποσυνδεδεμένος, θα το στείλει αμέσως μόλις συνδεθεί.


Tilix: Default
george@george-VPC: ~/Desktop/geostar
docker network disconnect unobile
unobile1
george@george-VPC:~/Desktop/geostar$


Tilix: Default
george@george-VPC: ~/Desktop/geostar
docker network disconnect unobile
unobile1
docker network connect unobile1 u
mobile1
george@george-VPC:~/Desktop/geostar$

Τέλος τρέχουμε το αρχείο script «stop_intermediate.sh» για την αυτοματοποίηση των εντολών που χρειάζονται να εκτελεστούν μετά την αποπεράτωση του όλου εγχειρήματος.





Ενότητα 5: Συζήτηση

Συνοψίζοντας, κατά τη διάρκεια της πτυχιακής εργασίας μελετήθηκε η τεχνολογία του virtualization, καθώς η τρέχουσα και οι μελλοντικές αρχιτεκτονικές του διαδικτύου. Συγκεκριμένα, αναλύθηκε η λειτουργία των Virtual Machines και Docker, και παρουσιάστηκε η μεταξύ τους σύγκριση. Επιπλέον, έγινε ανάλυση της τρέχουσας υποδομής του διαδικτύου, των πρωτοκόλλων επικοινωνίας (π.χ. TCP/IP) και των μηχανισμών διευθυνσιοδότησης. Στη συνέχεια έγινε ανάλυση των αρχιτεκτονικών NDN, TDN και UMOBILE. Τέλος, δόθηκε ένας οδηγός εγκατάστασης Docker και υλοποιήθηκε ένα δίκτυο με επικοινωνία NDN over DTN.

Εμπνευσμένος από το umobile project, σκοπός της παρούσας εργασίας ήταν να αποδείξει ότι μπορεί να δημιουργηθεί ένα δίκτυο όπου οι συσκευές του (containers) θα επικοινωνούν με πρωτόκολλα NDN/DTN. Σαφώς, με τη χρήση της τεχνολογίας αυτής μπορούν να γίνει μια σειρά από άλλες προσομοιώσεις σεναρίων όπου χαρακτηρίζονται από διαλείπουσα συνδεσιμότητα.

Πηγές

https://docs.docker.com/

https://en.wikipedia.org/wiki/Internet_Protocol

https://en.wikipedia.org/wiki/IPv6

https://en.wikipedia.org/wiki/Transmission_Control_Protocol

https://en.wikipedia.org/wiki/Internet_protocol_suite

https://www.geeksforgeeks.org/tcp-ip-model/

https://searchnetworking.techtarget.com/definition/TCP-IP

https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol

https://www.infoblox.com/glossary/dhcp-server/

http://hermes.di.uoa.gr/RETUDIS/Dhcp/dhcp.html

https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

https://www.webopedia.com/TERM/H/HTTP.html

https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

https://en.wikipedia.org/wiki/Domain_Name_System

https://en.wikipedia.org/wiki/Information-centric_networking

https://www.cs.tufts.edu/comp/150IDS/final_papers/dgriff03.1/FinalReport.html

https://irtf.org/icnrg

https://named-data.net/project/

https://en.wikipedia.org/wiki/NAT_traversal

https://dl.acm.org/doi/pdf/10.1145/2656877.2656887

https://tools.ietf.org/html/rfc4838

https://www.ee.ucl.ac.uk/~ipsaras/files/umobile-commag-connecting-edges.pdf

Βιβλιογραφία

[1] Al-Fares, M. (2008). A scalable, commodity data center network architecture. *ACM SIGCOMM Computer Communication Review*

[2] Yiou, D. P. (24 September 1996). Macintosh Program performs time-series analysis

[3] Wilson, K. (2015). *About Windows.* Apress, Berkeley, CA

[4] *Microsoft*. (n.d.). Retrieved from https://www.microsoft.com/en-us

[5] Hayes, B. (2008). Cloud computing. *Communications of the ACM*, 9-11

[6] Daniel P. Bovet, M. C. (Nov 17, 2005). *Understanding the Linux Kernel: From I/O Ports to Process Management.*

[7] Bovet, D. P., & Cesati, M. (2006). *Understanding the Linux Kernel: From I/O Ports to Process Management third edition.* O' REILLY

[8] Feller, J., & Fitzgerald, B. (2002). *Understanding open source software development*

[9] *CGroups.* (n.d.). Retrieved from Red Hat: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/resource_management_guide/ch01

[10] Rosen, R. (May 2013). *Resource management:Linux kernel Namespaces and cgroups*

[11] *Docker*. (n.d.). Retrieved from Docker Inc: https://www.docker.com/company

[12] *Solomon Hykes*. (n.d.). Retrieved from https://www.docker.com/blog/author/solomon-hykes/

[13] *Sebastien Pahl*. (n.d.). Retrieved from https://www.linkedin.com/in/spahl

[14] Gay, S., & Hole, M. (2002). Types and Subtypes for Client-Server Interactions. (pp. 74-90). Springer, Berlin, Heidelberg

[15] *Daemon Definition*. (2005, August 16). Retrieved from http://www.linfo.org/daemon.html

[16] *restfulapi*. (n.d.). Retrieved from Rest: https://restfulapi.net/rest-architectural-constraints/

[17] *CLI*. (n.d.). Retrieved from Command-line interface: https://www.w3schools.com/whatis/whatis_cli.asp

[18] *Linux Ubuntu*. (n.d.). Retrieved from https://ubuntu.com/

[19] *Java*. (n.d.). Retrieved from https://www.java.com/en/

[20] *Data Base*. (n.d.). Retrieved from https://www.oracle.com/database/what-is-database.html

[21] *Docker Hub*. (n.d.). Retrieved from https://hub.docker.com/

[22] *YAML*. (n.d.). Retrieved from https://yaml.org/

[23] *AUFS*. (n.d.). Retrieved from http://aufs.sourceforge.net/

[24] Rodeh, O., Bacik, J., & Mason, C. (2013). BTRFS: The Linux B-Tree Filesystem. *ACM Transactions on Storage*

[25] *VFS*. (n.d.). Retrieved from https://www.tldp.org/LDP/khg/HyperNews/get/fs/vfstour.html

.
[26] *Device Mapper*. (n.d.). Retrieved from redhat: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/logical_volume_manager_administration/device_mapper

[27] *Google*. (n.d.). Retrieved from https://about.google/

[28] Doraswamy, N. (n.d.). *IPSec.* Prentice Hall PTR.

[29] Hoffman, P. (2002). *SMTP Service Extension for Secure SMTP over Transport Layer Security*

[30] Postel, J., & Reynolds, J. (n.d.). *File Transfer Protocol*

[31] Berners-Lee, T., Fielding, R., & Frystyk, H. (1996). *Hypertext Transfer Protocol*

[32] *URL*. (n.d.). Retrieved from https://www.verisign.com/en_US/website-presence/online/what-is-a-url/index.xhtml

[33] *HTML*. (n.d.). Retrieved from https://www.w3schools.com/html/

[34] *JANA.* (n.d.). Retrieved from https://www.iana.org/

[35] *Van Jacobson*. (n.d.). Retrieved from https://www.internethalloffame.org/official-biography-van-jacobson

[36] A new way to look at networking https://www.youtube.com/watch?v=oCZMoY3q2uM

[37] *User Datagram Protocol*. (n.d.). Retrieved from https://www.hjp.at/doc/rfc/rfc768.html

[38] *Thin Waist*. (n.d.). Retrieved from https://cacm.acm.org/magazines/2019/7/237714-on-the-hourglass-model/fulltext

[39] *Edge to edge*. (n.d.). Retrieved from https://www.youtube.com/watch?v=oCZMoY3q2uM

[40] *Cache Memory*. (n.d.). Retrieved from https://www.geeksforgeeks.org/cache-memory-in-computer-organization/

[41] *Nat*. (n.d.). Retrieved from http://www.internet-computer-security.com/VPN-Guide/NAT-T.html

[42] S.Kent, Lynn, C., & Seo, K. (2000). Secure Border Gateway Protocol. (pp. 582 - 592). IEEE

[43] *CFDP*. (n.d.). Retrieved from https://indico.esa.int/event/145/contributions/817/attachments/888/1068/01_-_CFDP.pdf

[44] *CIDR.* (n.d.). Retrieved from https://whatismyipaddress.com/cidr

[45] Berners-Lee, T., Fielding, R., & Masinter, L. (2005). *Uniform Resource Identifier (URI)*

[46] Scharf, M., & Kiesel, S. (2006). Head-of-line Blocking. *IEEE Globecom 2006.* San Francisco, CA, USA: IEEE

[47] C.A. Sarros, Sotiris Diamantopoulos (2018). Connecting the Edges: A Universal, Mobile-centric, and Opportunistic Communications Architecture. *IEEE Communications Magazine*