

1. Introduction

Υπολογιστικό Νέφος

- η αποθήκευση, επεξεργασία και χρήση δεδομένων από απομακρυσμένους υπολογιστές στους οποίους εξασφαλίζεται πρόσβαση μέσω του Διαδικτύου
- λόγος αναγκαιότητας: εκθετική αύξηση του όγκου δεδομένων

Χαρακτηριστικά των Υπολογιστικών Νεφών

- Κοινή χρήση: Οι εξυπηρετητές τίθενται στη διάθεση πολλών χρηστών μέσω τεχνολογιών εικονικοποίησης
- Δέσμευση πόρων κατά απαίτηση: Επιπλέον πόροι τίθενται στη διάθεση του χρήστη όποτε απαιτείται
- Τιμολόγηση ανάλογα με τη χρήση: Ο κάθε χρήστης χρεώνεται με βάση τους πόρους που χρησιμοποιήσει πραγματικά

Μοντέλα Ανάπτυξης Νέφους

Δημόσιο Νέφος

- οι εγκαταστάσεις υποδομής και οι προσφερόμενες υπηρεσίες, παρέχονται από εταιρείες παρόχους (π.χ. Google) προς το ευρύ κοινό
- Πλεονέκτημα: χαμηλό κόστος - ο χρήστης χρειάζεται μια οικονομική συσκευή - Δυνατότητα χρήσης και διαχείρισης εφαρμογών χωρίς την αγορά, κατοχή, διαχείριση και συντήρηση κάποιας υποδομής, το κόστος αυτό μετακυλίεται στον πάροχο
- Μειονέκτημα: ασφάλεια και ιδιωτικότητα των δεδομένων (δεν έχουμε εικόνα για τον τρόπο διαχείρισης των δεδομένων μας, πρέπει να εμπιστευτούμε τον πάροχο)

Ιδιωτικό Νέφος

- οι εγκαταστάσεις υποδομής και οι προσφερόμενες υπηρεσίες, παρέχονται από έναν οργανισμό σε χρήστες/υπαλλήλους του οργανισμού (π.χ. τράπεζες, πανεπιστήμια)
- Πλεονέκτημα: λύνει το πρόβλημα της ασφάλειας - μεγαλύτερη εμπιστοσύνη, καθώς οι υπηρεσίες παρέχονται από τον ίδιο τον οργανισμό
- Μειονέκτημα: υψηλότερο κόστος, καθώς ο οργανισμός πρέπει να εγκαταστήσει και να διαχειριστεί τα δικά του κέντρα δεδομένων (datacenters)

Υβριδικό Νέφος

- Συνδυασμός δημόσιου και ιδιωτικού νέφους: Μέρος των δεδομένων (μικρού όγκου, σημαντικά και απόρρητα δεδομένα) αποθηκεύεται στο ιδιωτικό νέφος και τα υπόλοιπα (μεγάλου όγκου) στο δημόσιο νέφος
- Υψηλότερο επίπεδο ασφάλειας και εμπιστοσύνης στο ιδιωτικό νέφος
- Χαμηλότερο κόστος χρήσης στο δημόσιο νέφος

2018-06

Να αναφέρετε τα πλεονεκτήματα και τα μειονεκτήματα του δημόσιου και ιδιωτικού νέφους, όσον αφορά το κόστος υπηρεσιών και την ιδιωτικότητα/ασφάλεια. Υπάρχει κάποια δυνατότητα να συνδυαστούν τα πλεονεκτήματα των δύο αυτών μοντέλων ανάπτυξης?

Πλεονεκτήματα Υπολογιστικών Νεφών

- Μείωση του κόστους: Δυνατότητα χρήσης και διαχείρισης εφαρμογών χωρίς την αγορά, κατοχή, διαχείριση και συντήρηση κάποιας υποδομής
- Ευέλικτη δέσμευση πόρων: Δυνατότητα δέσμευσης και χρήσης υπολογιστικών πόρων ανάλογα με τις ανάγκες του χρήστη

- Ευκολία πρόσβασης: Μέσω του Διαδικτύου, οι χρήστες έχουν πρόσβαση στα δεδομένα και στις εφαρμογές τους οπουδήποτε και από οποιαδήποτε συσκευή
- Αξιοπιστία: Η δυνατότητα μετακίνησης εικονικών μηχανών αλλά και διατήρησης αντιγράφων των δεδομένων διασφαλίζουν την αξιόπιστη και απρόσκοπτη εκτέλεση εφαρμογών ακόμη και όταν παρουσιάζονται σφάλματα στο υλικό ή λογισμικό

Μειονεκτήματα Υπολογιστικών Νεφών

- Ασφάλεια δεδομένων: Κακόβουλοι χρήστες ενδέχεται να αποκτήσουν πρόσβαση σε δεδομένα που έχουν αποθηκευτεί στο νέφος ή σε δεδομένα που μεταφέρονται από τη συσκευή του χρήστη προς το νέφος (ή και αντίστροφα)
- Πτώση απόδοσης: Η διαχείριση της υποδομής από τον πάροχο (π.χ. μετακίνηση εικονικών μηχανών) μπορεί να έχει επίπτωση στην απόδοση των εφαρμογών

Μοντέλα Υπηρεσιών Νέφους

Λογισμικό ως Υπηρεσία (Software as a Service - SaaS)

- Ο χρήστης έχει πρόσβαση σε μια εφαρμογή λογισμικού (π.χ. email, κειμενογράφος), η οποία φιλοξενείται και διαχειρίζεται από τον πάροχο
- Οι δυνατότητες ελέγχου από τον χρήστη περιορίζονται στην αλληλεπίδραση με την ίδια την εφαρμογή (συνήθως μέσω ενός φυλλομετρητή browser)

Πλατφόρμα ως Υπηρεσία (Platform as a Service - PaaS)

- Ο χρήστης έχει πρόσβαση σε ένα περιβάλλον ανάπτυξης και διαχείρισης εφαρμογών (π.χ. MS Azure), το οποίο φιλοξενείται και διαχειρίζεται από τον πάροχο
- Ο χρήστης δεν έχει άμεση πρόσβαση στον υπολογιστή όπου φιλοξενείται το συγκεκριμένο λογισμικό

Υποδομή ως Υπηρεσία (Infrastructure as a Service - IaaS)

- Ο χρήστης έχει πρόσβαση σε υπολογιστικούς πόρους που συνήθως παρέχονται με τη μορφή εικονικών υπολογιστών (virtual machines), η υποδομή και οι πόροι διαχειρίζονται από τον πάροχο
- Ο χρήστης έχει άμεση πρόσβαση στο Λ/Σ της εικονικής μηχανής και δυνατότητα εγκατάστασης και διαχείρισης λογισμικού

Lab1 Openstack

Πλατφόρμα ανοικτού κώδικα για τη διαχείριση clouds. Επιτρέπει τη δημιουργία και διαχείριση εικονικών μηχανών, παρέχει περιορισμένη υποστήριξη για containers. Ανήκει στην κατηγορία υπηρεσιών υποδομής (IaaS) (παρόμοια υπηρεσία με oceanos, το openstack έχει πιο ευρύ φάσμα υπηρεσιών)

2. Κέντρα Δεδομένων

Χαρακτηριστικά Σύγχρονων Κέντρων Δεδομένων

- **Ευρεία κλίμακα:** ένα κέντρο δεδομένων συνήθως περιλαμβάνει χιλιάδες εξυπηρετητές
- **Εικονικοποίηση:** Ευρεία χρήση εικονικών μηχανών για τη βέλτιστη αξιοποίηση των υπολογιστικών πόρων ενός κέντρου δεδομένων
- **Υλικό ευρείας χρήσης:** Τα κέντρα δεδομένων απαρτίζονται από (συνήθως ομογενές) υλικό ευρείας χρήσης και σχετικά χαμηλού κόστους (commodity hardware)

2019-06, 2020-06

Θέλουμε να αυξήσουμε την επεξεργαστική ισχύ ενός DataCenter. Ποιά είναι η καλύτερη εναλλακτική, να αναβαθμίσουμε τους υπάρχοντες Η/Υ ή να προσθέσουμε περισσότερους και γιατί?

Επικρατεί η τάση για Scale-Out, δηλαδή αντί να έχουμε λίγους και πολύ ισχυρούς υπολογιστές, χρησιμοποιούνται περισσότεροι υπολογιστές με μέτρια υπολογιστική ισχύ και κοινά χαρακτηριστικά (commodity hardware). Υπάρχει η δυνατότητα κλιμάκωσης της διαθέσιμης υπολογιστικής ισχύος, προσθέτοντας περισσότερους εξυπηρετητές. Για το σκοπό αυτό χρησιμοποιούνται αρχιτεκτονικές δικτύων (fat-tree) οι οποίες επιτρέπουν την προσθήκη περισσότερων εξυπηρετητών για την αύξηση της υπολογιστικής ισχύος. Αυτό συμβαίνει γιατί ο ρυθμός αύξησης του όγκου των δεδομένων ξεπερνά κατά πολύ τον ρυθμό αύξησης της υπολογιστικής ισχύος ανά υπολογιστή, **άρα οι ανάγκες επεξεργασίας των δεδομένων απαιτούν τη συνδρομή πολλών υπολογιστών.**

Υπερκάλυψη Δικτύου Κέντρων Δεδομένων (Oversubscription) Δ2 30'

Βαθμός υπερκάλυψης (O): Η αναλογία της συνολικής κίνησης ενός rack προς την κίνηση που βγαίνει εκτός του rack

Παράδειγμα: Έστω συνολική κίνηση 30Gbps, εντός του rack τα 20 και εκτός τα 10. Άρα $O = 30:10 = 3:1$

2022-09

Παράδειγμα για την υπερκαλυψη δικτυου κεντρου δεδομενων

Τοπολογία Fat-Tree Δ2 37'

- $k^2/4$ root switches με k θύρες το καθένα
- k pods, το καθένα με δύο στρώματα (top of the rack, aggregation) από $k/2$ switches το στρώμα καθε pod άρα $(k/2) * k$ switches σε κάθε στρώμα συνολικά και $k^2/4$ εξυπηρετητές καθε pod
- $k^3/4$ εξυπηρετητές συνολικά

2018-06, 2019-06, 2022-07

Να υπολογίσετε πόσα root, aggregation, και Top-of-the-Rack switch χρειάζονται για τη δικτύωση 16000 (και για 6750) servers σε τοπολογία fat-tree τριών επιπέδων. Να υπολογιστούν:

- αριθμός root switches με πόσες θύρες το καθένα
- αριθμός pods με πόσα switches και servers το καθένα

Αρχιτεκτονική Δικτύου Portland

Περιορισμοί Ethernet σε Κέντρα Δεδομένων

- Μεγάλοι πίνακες δρομολόγησης (αντιστοιχίζουν δ/ση MAC με port του switch): Το μέγεθος του πίνακα δρομολόγησης στα switch είναι ανάλογο του αριθμού των εξυπηρετητών. Τα switches χρησιμοποιούν μια ειδική μνήμη (TCAM) για να αποθηκεύουν τους πίνακες δρομολόγησης, η οποία είναι πιο ακριβή και καταναλώνει πολλή ενέργεια, άρα αυξάνεται το κόστος εγκατάστασης του datacenter.
- Κίνηση δικτύου λόγω εκπομπής (broadcast): Οι εκπομπές ARP και DHCP καταναλώνουν δικτυακούς και υπολογιστικούς πόρους από τους εξυπηρετητές και τα switch

- Spanning trees: εξαλείφει βρόχους από μία τοπολογία (απενεργοποιεί θύρες switches). Χρησιμοποιούν μόνο ένα μικρό αριθμό συνδέσεων, οπότε ένα μεγάλο μέρος του διαθέσιμου bandwidth παραμένει αναξιοποίητο. Επιπλέον επιβαρύνουν τις συνδέσεις του switch που έχει επιλεγεί ως root από τον αλγόριθμο σχηματισμού του spanning tree

Portland Δ3 15'

- έχει εφαρμογή σε ιεραρχικές τοπολογίες κέντρων δεδομένων π.χ. τοπολογίες fat-tree
- τροποποιεί τα switches, όχι τους servers
- Χρησιμοποιεί διευθύνσεις IP για την ταυτοποίηση των εξυπηρετητών και εισάγει τις διευθύνσεις pseudo MAC (PMAC) για την κωδικοποίηση της θέσης των εξυπηρετητών και τις χρησιμοποιεί για τη δρομολόγηση των πακέτων στο δίκτυο (μείωση μεγέθους πινάκων δρομολόγησης λόγω ιεραρχίας των διευθύνσεων PMAC).
Μορφή: pod.position.port.vmid (16.8.8.16)bits

Ποια είναι η διεύθυνση PMAC μιας εικονικής μηχανής με ID 1, που συνδέεται στο 0ο PoD, στη θύρα 3 του 1ου ToR switch? 00-00-01-03-00-01

2022-07

Ποια είναι η διεύθυνση PMAC μιας εικονικής μηχανής με ID 3, που συνδέεται στο 10ο PoD, στη θύρα 25 του 5ου ToR switch? 00-0A-05-19-00-03

Εκχώρηση PMAC

1. Μόλις ο server συνδεθεί στο ToR switch, το switch σχηματίζει την PMAC (π.χ 00-00-01-03-00-00)
2. Το ToR switch στέλνει στον επόπτη δικτύου την πληροφορία αυτή, ο οποίος την αποθηκεύει στον πίνακα που διατηρεί (IP, PMAC)
3. Το ToR switch αποθηκεύει σε πίνακα (IP, MAC, PMAC) την πληροφορία για κάθε server (μόνο) ο οποίος συνδέεται σε αυτό

Αναζήτηση Φυσικής Διεύθυνσης (ARP) και Δρομολόγηση Πακέτων Δ3 33'

Έστω ένας server θέλει να στείλει πακέτα στην IP 172.23.192.120. Χρειαζόμαστε τη διεύθυνση MAC του παραλήπτη!

1. Ο server στέλνει ένα αίτημα ARP (broadcast) για την IP 172.23.192.120
2. Το ToR switch αναχαιτίζει το αίτημα ARP, και το προωθεί προς τον επόπτη δικτύου
3. Ο επόπτης δικτύου αναζητά τη διεύθυνση PMAC στον πίνακα (IP, PMAC) και στέλνει το αντίστοιχο ζεύγος (IP, PMAC) στο ToR switch
4. Το ToR switch γνωστοποιεί την PMAC στον server που έστειλε το αίτημα (ο server θεωρεί ότι έλαβε δ/ση MAC)
5. Ο server στέλνει πακέτα με προορισμό IP 172.23.192.120, PMAC 00-02-01-02-00-00
6. Το ToR switch δρομολογεί τα πακέτα στον παραλήπτη με βάση την PMAC
7. Το ToR switch του παραλήπτη αντικαθιστά την PMAC με την MAC (από τον πίνακα IP, MAC, PMAC) και στέλνει τα πακέτα στον παραλήπτη (αν δεν αντικαθιστούσε, ο παραλήπτης θα απέρριπτε τα πακέτα. Η προώθηση μέσω PMAC δεν γίνεται αντιληπτή από τους εξυπηρετητές)

2023-01, 2023-06, 2024-06

Στην αρχιτεκτονική Portland, ποιος είναι ο λόγος που το κάθε ToR switch αποθηκεύει τη MAC διεύθυνση του κάθε εξυπηρετητή που συνδέεται σε αυτόν?

Αυτοματοποίηση του Προσδιορισμού Θέσης

Τα switch στην αρχιτεκτονική Portland ανταλλάσσουν μεταξύ τους μηνύματα (Location Discovery Messages LDM). Περιλαμβάνουν στοιχεία: Switch ID, Αριθμός pod, Θέση εντός του pod, Επίπεδο δέντρου.

Πρωτόκολλο Προσδιορισμού Θέσης Δ3 59'

Ανακάλυψη επιπέδου του δέντρου: το switch στέλνει LDM μηνύματα προς όλες τις θύρες

- Αν είναι ToR switch (Επίπεδο 0) τότε δε θα πάρει απάντηση από κάτω (τους servers), άρα καταλαβαίνει ότι είναι στο Επίπεδο 0
- Αν είναι aggregation switch (Επίπεδο 1) τότε θα λάβει κάποια μηνύματα από τους ToR switches που θα λένε ότι βρίσκονται στο Επίπεδο 0, άρα θα καταλάβει ότι βρίσκεται στο Επίπεδο 1
- Αν είναι root switch (Επίπεδο 2) τότε θα λάβει μηνύματα μόνο από aggregation switches που θα λένε ότι βρίσκονται στο Επίπεδο 1, άρα θα καταλάβει ότι βρίσκεται στο Επίπεδο 2

Ανάθεση θέσης (μέσα στο pod), αφορά μόνο τα ToR switches: στέλνει LDM μηνύματα προς τα aggregation switches του pod και προτείνει μια θέση (για τον εαυτό του)

- Αν η θέση που προτείνει είναι διαθέσιμη παίρνει απάντηση ΝΑΙ
- Αν δεν είναι διαθέσιμη παίρνει απάντηση ΟΧΙ και προτείνει τον επόμενο αριθμό για θέση

Ανάθεση αριθμού pod: **το ToR switch με θέση 0** στέλνει αίτημα για αριθμό pod στον επόπτη δικτύου ο οποίος απαντά με τον επόμενο διαθέσιμο αριθμό. Στη συνέχεια το switch στέλνει μηνύματα κοινοποίησης του αριθμού pod μέσα στο pod

2018-06, 2022-09

Περιγράψτε τον τρόπο με τον οποίο διενεργούνται αυτόματα οι παρακάτω λειτουργίες στην αρχιτεκτονική Portland: (α) ένα switch ανακαλύπτει σε ποιο επίπεδο της τοπολογίας δικτύου βρίσκεται, (β) ένα Top-of-the-Rack switch λαμβάνει ένα μοναδικό αριθμό που προσδιορίζει τη θέση του στο Pod, και (γ) ένα Pod λαμβάνει ένα μοναδικό αριθμό που το προσδιορίζει σε όλο το δίκτυο.

Αρχιτεκτονική Δικτύου SEATTLE

Κατανεμημένοι Πίνακες Κατατεμαχισμού Δ3 72', Δ4 1'

2024-06

Θεωρήστε έναν κατανεμημένο πίνακα κατατεμαχισμού (DHT), όπου από κάθε κόμβο διατηρούνται δείκτες προς όλους τους υπόλοιπους κόμβους του DHT. Περιγράψτε τα πλεονεκτήματα και μειονεκτήματα ενός τέτοιου DHT. Σε ποιά περίπτωση ενδείκνυται η χρήση ενός τέτοιου DHT?

Το πλεονέκτημα ενός τέτοιου DHT είναι η **ελαχιστοποίηση του χρόνου αναζήτησης κλειδιού**, καθώς απαιτείται μόνο ένα μήνυμα για την εύρεση του κλειδιού που αναζητούμε - από τον πρώτο κόμβο απευθείας στον κόμβο που κατέχει το κλειδί. Ένα από τα μειονεκτήματα είναι ότι **αυξάνεται ο όγκος πληροφορίας για τη δρομολόγηση στο DHT**, καθώς κάθε υπολογιστής πρέπει να αποθηκεύει πληροφορία για όλους τους υπόλοιπους υπολογιστές. Επιπλέον, τα δίκτυα αυτά είναι δυναμικά, δηλαδή έχουμε πολύ συχνά συνδέσεις και αποσυνδέσεις χρηστών. Για κάθε κόμβο που συνδέεται ή αποσυνδέεται πρέπει να ενημερωθούν όλοι οι υπόλοιποι κόμβοι, κατά συνέπεια έχουμε **μεγάλη διάχυση**

πληροφορίας. Για τους λόγους αυτούς, η χρήση ενός τέτοιου DHT ενδείκνυται μόνο στην περίπτωση που έχουμε ένα μικρό δίκτυο (δεκάδες κόμβοι).

Αρχιτεκτονική SEATTLE Δ4 39'

- Επίπεδο (flat - χωρίς υποδίκτυα) δίκτυο αρχιτεκτονικής Ethernet. Δρομολόγηση βάσει MAC.
- Τροποποιήσεις μόνο στα switch (όχι στους servers).
- Βασικές λειτουργίες του δικτύου με τη χρήση συναρτήσεων hash.
- Στόχοι:
 - Μείωση του μεγέθους του πίνακα δρομολόγησης στα switch
 - Αποφυγή της εκπομπής (broadcast) πακέτων για ARP και DHCP
 - Διατήρηση των προτερημάτων της αρχιτεκτονικής Ethernet: Απλότητα στη διαχείριση του δικτύου, μετακίνηση τερματικών χωρίς την αλλαγή διεύθυνσης IP

Λειτουργία SEATTLE Δ4 43'

1. **Εγγραφή νέου τερματικού x στο δίκτυο:** (κανονικά με το Ethernet θα ξεκινούσε η διαδικασία ώστε να αποθηκευτούν οι διαδρομές προς τον x σε όλα τα switches του δικτύου, άρα αύξηση μεγέθους των πινάκων δρομολόγησης)
 - a. το switch A εφαρμόζει τη **συνάρτηση Hash** πάνω στη δ/ση MACx, η οποία δίνει ως αποτέλεσμα ένα από τα switches, στην προκειμένη περίπτωση έστω το B
 - b. στο switch B **αποθηκεύεται το <MACx, A>** δηλαδή ως κλειδί η δ/ση MACx και ως τιμή η θέση του x, που είναι το switch A
2. **Αποστολή πακέτου από y στον x:**
 - a. ο y στέλνει ένα πακέτο με δ/ση παραλήπτη τη MACx
 - b. το πακέτο φτάνει στο switch D. Το switch D δε ξέρει σε ποιο switch συνδέεται ο x, εφαρμόζει τη συνάρτηση Hash πάνω στη δ/ση MACx, και μαθαίνει ότι η πληροφορία για τη θέση του x βρίσκεται στο B, οπότε **προωθεί το πακέτο προς το switch B**
 - c. το B γνωρίζει τη θέση του x, άρα προωθεί το πακέτο προς το switch A το οποίο με τη σειρά του το προωθεί προς τον x

Το πακέτο προωθείται στον παραλήπτη μέσω ενός ενδιάμεσου κόμβου (B) ο οποίος ξέρει τη θέση του παραλήπτη, έτσι πετυχαίνεται η μείωση των πινάκων δρομολόγησης των υπόλοιπων switches.

Παρατήρηση: μεγαλύτερη διαδρομή (D-C-B-A ενώ γινόταν D-C-A).

Caching

Μετά την αποστολή του πρώτου πακέτου μέσω της διαδρομής D-C-B-A, προαιρετικά ο B μπορεί να ενημερώσει τον D για τη θέση του x (**Ενημέρωση <MACx,A> στο D**), άρα ο D μπορεί να στείλει τα επόμενα πακέτα μέσω της πιο σύντομης διαδρομής (**caching** - προσωρινή αποθήκευση στον D, πχ. όσο διαρκεί η μετάδοση από το y προς το x)

2024-02

Αναφέρετε ένα πλεονέκτημα και ένα μειονέκτημα της λειτουργίας caching στην αρχιτεκτονική SEATTLE. Σε ποιά περίπτωση ενδείκνυται η ενεργοποίηση αυτής της λειτουργίας?

2023-09

Εξηγήστε το βασικό πλεονέκτημα της αποθήκευσης της θέσης ενός υπολογιστή σε ένα μόνο switch, όπως διενεργείται από το SEATTLE (συγκρίνεται με το Ethernet). Υπάρχει η δυνατότητα από το SEATTLE να αποθηκευτεί η θέση ενός υπολογιστή σε περισσότερα από ένα switch και, αν ναι, σε ποιά περίπτωση συμβαίνει αυτό?

Διαχείριση Μετακίνησης Τερματικών Δ4 56'

Σε συνέχεια του προηγούμενου παραδείγματος έχουμε τη θέση του server x αποθηκευμένη σε 3 switches, στο A, B (hash), και D (caching). Έστω ότι ο x μετακινείται από το switch A στο switch G, τα βήματα για την ενημέρωση του x στα switches είναι:

1. το switch G εφαρμόζει τη συνάρτηση hash πάνω στη δ/ση MACx, παίρνει το ίδιο αποτέλεσμα (switch B)
2. το switch G ενημερώνει το switch B για τη νέα θέση του x στο δίκτυο (<MACx, G>). Το switch B πριν διαγράψει την παλιά θέση ενημερώνει το switch A για τη νέα θέση του x
3. όταν χρησιμοποιηθεί η πιο σύντομη διαδρομή, δηλαδή το switch D στείλει ένα πακέτο προς τον x, θα το στείλει μέσω του A και το switch A θα ενημερώσει το D για τη νέα θέση του x

Διαχείριση Αιτημάτων ARP Δ4 60'

1. **Εγγραφή server α:** ο α συνδέεται στο switch Sa
2. **Υπολογισμός $F(IPa) = ra$:** Εφαρμόζεται η συνάρτηση hash πάνω στην δ/ση IP του α (ένα αίτημα ARP αναζητεί ποια δ/ση MAC αντιστοιχεί σε μια δ/ση IP). Παίρνουμε ως αποτέλεσμα το switch ra
3. **Αποθήκευση $(IPa, MACa, Sa)$:** Αποθήκευση του κλειδιού IPa, του MACa, και της θέσης του α (Sa) στο switch ra
4. **Broadcast αίτημα ARP προς α:** Αν ο b θέλει να στείλει ένα πακέτο στον α με βάση την IP, για να μπορέσει να στείλει το πακέτο προς τον Sb θα πρέπει να μάθει και τη δ/ση MAC του παραλήπτη. Άρα θα στείλει ένα αίτημα ARP (broadcast) προς α για να μπορέσει να μάθει τη MAC του παραλήπτη
5. **Υπολογισμός $F(IPa) = ra$:** Το Sb εφαρμόζει τη συνάρτηση hash πάνω στη δ/ση IP του παραλήπτη, παίρνει ως αποτέλεσμα το switch ra
6. **Unicast αίτημα ARP προς ra:** Το Sb διοχετεύει την κίνηση προς το switch ra
7. **Unicast απάντηση ARP $(IPa, MACa, Sa)$:** Η πληροφορία $(IPa, MACa, Sa)$ κοινοποιείται προς το Sb, ο b μαθαίνει τη MACa και το Sb μπορεί να στείλει ένα πακέτο προς τον α (έχει και τη δ/ση MACa και τη θέση του α - Sa)

2018-06

Να περιγράψετε πως η αρχιτεκτονική SEATTLE αποφεύγει την εκπομπή μηνυμάτων σε όλο το δίκτυο (broadcast) κατά την εξυπηρέτηση μηνυμάτων ARP.

Διαχείριση Αιτημάτων DHCP Δ4 64'

1. **Εγγραφή DHCP server:** ο d συνδέεται στο switch Sd
2. **Υπολογισμός $F(MACd) = r$:** Εφαρμόζεται η συνάρτηση hash πάνω στην δ/ση MAC του d. Παίρνουμε ως αποτέλεσμα το switch r
3. **Αποθήκευση $(MACd, Sd)$:** Αποθήκευση του κλειδιού MACd και της θέσης του d (Sd) στο switch r
4. **Broadcast αίτημα DHCP:** Όταν ο h συνδεθεί στο δίκτυο και θέλει να πάρει μια δ/ση IP, στέλνει broadcast αίτημα DHCP

5. **Υπολογισμός $F(0x\text{DHCP}) = r$:** Το Sd εφαρμόζει τη συνάρτηση hash πάνω στη δ/ση MACd, παίρνει ως αποτέλεσμα το switch r
6. **Unicast αίτημα DHCP προς r**
7. **Αίτημα DHCP προς Sd:** Το r διοχετεύει την κίνηση προς το switch Sd
8. **Παράδοση DHCP αιτήματος στο d**

Εξισορρόπηση Φορτίου

Εξισορρόπηση φορτίου είναι η ισοκατανομή της κίνησης μεταξύ των συνδέσεων σε μια τοπολογία fat-tree. Αν αξιοποιηθούν οι πολλαπλές διαδρομές επιτρέπουν την αποστολή δεδομένων με πιο γρήγορο ρυθμό.

Δικτυακές Ροές

- Τα πακέτα κάθε διαφορετικής εφαρμογής αποτελούν μια διαφορετική ροή.
- Ορισμός ροής γίνεται με το 5-tuple: [srcIP, destIP, sourcePort, destPort, protocol (TCP,UDP)]
- Πακέτα που ανήκουν στην ίδια ροή πρέπει να ακολουθούν την ίδια διαδρομή (για να μην φτάνουν με λάθος σειρά και έχουμε καθυστερήσεις)

ECMP (Equal Cost Multiple Paths)

- στατική εξισορρόπηση φορτίου - δε χρειάζεται κάποια άλλη πληροφορία πέρα από τις επικεφαλίδες των πακέτων. Είναι μια τεχνολογία που υλοποιείται πολύ εύκολα
- κάνει ανάθεση διαδρομών του δικτύου σε δικτυακές ροές
 - **port = hash(packet_5-tuple) mod N** (N = πλήθος επιλογών στις διακλαδώσεις)
- έχει εφαρμογή στα **ToR** και στα **aggregation** switches (εκεί έχουμε διακλαδώσεις των συνδέσεων)
- δε λαμβάνει υπόψη το μέγεθος των ροών, **διαδρομές έχουν κοινές συνδέσεις**, αν δρομολογηθούν σε μια σύνδεση δύο ροές με μεγάλο όγκο δεδομένων (elephant flows), είναι πιθανό το bandwidth που απαιτείται για τις δύο ροές να υπερβαίνει τη χωρητικότητα της σύνδεσης, με αποτέλεσμα να έχουμε **συγκρούσεις πακέτων** (καθυστερήσεις και απώλειες πακέτων)
- η απόδοση του περιορίζεται όταν υπάρχει διαφοροποίηση στο μέγεθος των ροών, οδηγεί σε ανισοκατανομή του φορτίου, δηλαδή σε κάποιες συνδέσεις έχουμε πολύ μεγάλο φορτίο ενώ σε άλλες μικρό. Έχουμε **κακή αξιοποίηση (σπατάλη) διαθέσιμου bandwidth** μεταξύ root και aggregation switches.

2019-06, 2022-07

Σε ποιά επίπεδα (root, aggregation, ToR) μπορεί να χρησιμοποιηθεί το ECMP για κατανομή φόρτου? Σε ποιά περίπτωση περιορίζεται η απόδοση αυτής της τεχνικής?

2024-06

Εξηγήστε τα πλεονεκτήματα και μειονεκτήματα της τεχνικής εξισορρόπησης φορτίου ECMP. Σε ποιά περίπτωση ενδείκνυται η χρήση αυτής της τεχνικής?

Δυναμική Εξισορρόπηση Φορτίου (Hedera) Δ4 99'

1. **Προσδιορισμός ροών elephant:** Περιοδική καταγραφή τιμών στους μετρητές των switch
 - a. Ροές που υπερβαίνουν κάποιο προκαθορισμένο όριο (bps) θεωρούνται elephant

- b. Οι ροές με μικρό όγκο δεδομένων (mice) κατανέμονται στο δίκτυο με βάση το ECMP
- Υπολογισμός απαιτήσεων ροών:** Βάσει ενός πίνακα ροών, υπολογίζεται η απαίτηση της κάθε ροής στον αποστολέα και παραλήπτη επαναληπτικά, μέχρι να επέλθει σύγκλιση. Επαναλήψεις $O(|F|)$, F : # elephant flows
 - Ανάθεση ροών:** Ανάθεση διαδρομών του δικτύου σε ροές elephant με βάση τις απαιτήσεις των ροών

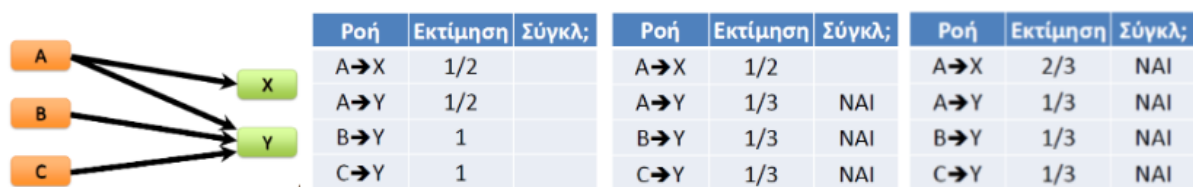
2023-09

Σε ένα δίκτυο υπάρχει χαμηλή διακύμανση στις δικτυακές ροές. Θα προτιμήσετε το ECMP ή το Hedera και γιατί;

Το ECMP είναι αποτελεσματικό για τις πολλές ροές μικρού μεγέθους, οπότε δεν υπάρχει λόγος να χρησιμοποιηθούν οι πιο πολύπλοκοι μηχανισμοί του Hedera

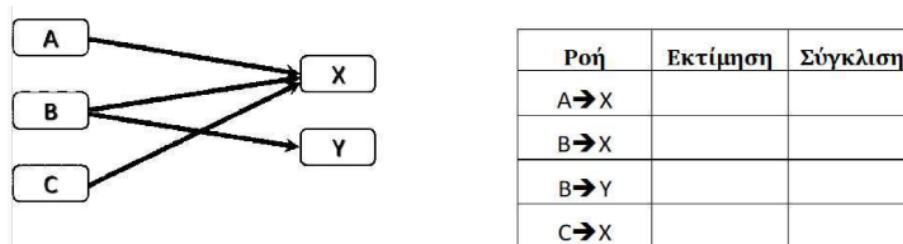
Υπολογισμός απαιτήσεων ροών Δ4 103'

Τεχνική max-min fairness



2023-06, 2024-02

Να υπολογίσετε τις απαιτήσεις των 4 ροών με βάση το παρακάτω σχήμα, σύμφωνα με την τεχνική που εφαρμόζεται στο Hedera. Θεωρήστε ότι κάθε αποστολέας και παραλήπτης έχει μία μονάδα bandwidth. Νά δείξετε αναλυτικά τη διαδικασία υπολογισμού με βάση τους περιορισμούς που υπεισέρχονται σε αποστολείς και παραλήπτες, συμπληρώνοντας τον παρακάτω πίνακα για κάθε βήμα μέχρι να επέλθει σύγκλιση.



Lab2 Mininet

Δημιουργεί πιστές αναπαραστάσεις μιας δικτυακής τοπολογίας

Ποιος είναι ο ρόλος των παραμέτρων mac και arp στο mininet;

- mac:** εκχώρηση απλοποιημένων διευθύνσεων MAC στους servers (έχουν μηδενικά, καταλήγουν στο id του server)
- arp:** δημιουργία ολοκληρωμένων πινάκων ARP σε όλους τους servers με τις MAC διευθύνσεις του κάθε server

3. Εικονικοποίηση

Εικονικοί Εξυπηρετητές

Εικονικοποίηση (Virtualization) Δ5 23'

Διαμερισμός των πόρων (υλικό) ενός υπολογιστή και ανάθεση τους σε εικονικές μηχανές που λειτουργούν σαν ξεχωριστοί υπολογιστές. Κάθε εικονική μηχανή έχει διαφορετικό ΛΣ. Ο επόπτης διαχειρίζεται τον διαμοιρασμό των πόρων

Κατηγορίες Εικονικοποίησης Δ5 30', Δ6 1'

1. Πλήρης Εικονικοποίηση (Full Virtualization) π.χ. KVM, VMWare, VirtualBox

Ο επόπτης εξομοιώνει εξ ολοκλήρου το υλικό του υπολογιστή (π.χ. ΚΜΕ, μνήμη, συσκευές E/E, BIOS) επιτρέποντας την εκτέλεση ενός μη τροποποιημένου φιλοξενούμενου ΛΣ.

- Το κάθε φιλοξενούμενο ΛΣ έχει την ψευδαίσθηση ότι διαχειρίζεται φυσικούς πόρους.
- Κάθε εικονική μηχανή τρέχει μια πλήρη έκδοση του ΛΣ (περιλαμβάνει kernel και userspace).
- Επιτρέπεται στα φιλοξενούμενα ΛΣ να εκτελούν κλήσεις συστήματος.
- Υψηλή ευελιξία: Δυνατότητα χρήσης διαφορετικών και μη τροποποιημένων ΛΣ
- Υψηλή απομόνωση: ο πυρήνας του ΛΣ είναι μέσα στην εικονική μηχανή και ο επόπτης διασφαλίζει το διαμοιρασμό των πόρων
- Χαμηλή απόδοση
 - Οι κλήσεις συστήματος συλλαμβάνονται από το λογισμικό ελέγχου του επόπτη και προσομοιώνονται, δηλαδή ο επόπτης κάνει μετάφραση αναφορών (απο id του εικονικού υλικού (πχ cru1) σε id του υλικού (πχ cru7))
 - Κατά την εκτέλεση λειτουργιών του πυρήνα, απαιτείται να ελεγχθεί αν οι λειτουργίες αφορούν πόρους που έχουν εκχωρηθεί σε αυτή την εικονική μηχανή, αλλιώς θα πρέπει να παρεμποδιστούν από τον επόπτη

2. Παραεικονικοποίηση (Paravirtualization) π.χ. Xen, Denali

Ο επόπτης παρέχει σε κάθε εικονική μηχανή μία διεπαφή προγραμματισμού (API) για την εκτέλεση κλήσεων συστήματος.

- Οι κλήσεις συστήματος εκτελούνται από τον επόπτη (μέσω του API) αντί από τον πυρήνα του φιλοξενούμενου ΛΣ (το ΛΣ ξέρει ότι είναι σε εικονική μηχανή).
- Κάθε εικονική μηχανή τρέχει μια πλήρη έκδοση του ΛΣ (περιλαμβάνει kernel και userspace)
- Απαιτεί τροποποίηση του πυρήνα του ΛΣ
- Υψηλότερη απόδοση σε σχέση με την πλήρη εικονικοποίηση καθώς δε χρειάζεται μεταφραση αναφορών και έλεγχος των πόρων από τον επόπτη
- Αποδεκτό επίπεδο απομόνωσης: ολόκληρο το ΛΣ φιλοξενείται εντός μιας εικονικής μηχανής
- Χαμηλότερη ευελιξία: Δυνατότητα χρήσης διαφορετικών ΛΣ, όμως απαιτείται τροποποίηση του πυρήνα του φιλοξενούμενου ΛΣ

3. Εικονικοποίηση ΛΣ (OS based Virtualization) π.χ. OpenVZ, LXC, Docker

Ένα στρώμα εικονικοποίησης επιτρέπει την τμηματοποίηση του ΛΣ σε περιέκτες και διασφαλίζει τον διαμοιρασμό των πόρων μεταξύ των περιεκτών.

- Κάθε περιέκτης παρέχει χώρο μόνο για την εκτέλεση εφαρμογών (userspace).
- Όλοι οι περιέκτες διαμοιράζονται τον πυρήνα του ΛΣ.

- Απευθύνεται σε χρήστες που δεν θέλουν πρόσβαση στο ΛΣ, θέλουν μόνο να τρέχουν εφαρμογές
- Υψηλή απόδοση: άμεση προσπέλαση του υλικού από τον πυρήνα - δεν μεσολαβεί επόπτης
- Χαμηλή ευελιξία: Αδυναμία χρήσης διαφορετικών ΛΣ, τα containers πρέπει να έχουν τον ίδιο πυρήνα
- Χαμηλή απομόνωση: λόγω του κοινού πυρήνα ο διαμοιρασμός των πόρων είναι πιο δύσκολος

2022-09

Γιατί σε πλήρη εικονικοποίηση υπάρχει καθυστέρηση με την χρήση αναφορών?

2023-01

Εξηγήστε τις διαφορές μεταξύ της πλήρους εικονικοποίησης και της παραεικονικοποίησης εξυπηρετητών. Ποιά τεχνική εξασφαλίζει υψηλότερη απόδοση και γιατί?

2023-09

Γιατί το επίπεδο απομόνωσης στους containers είναι πιο χαμηλό από ότι στις εικονικές μηχανές;

2018-06, 2019-06

Έστω ότι θέλετε να νοικιάσετε μια εικονική μηχανή με δυνατότητα παρέμβασης στον πυρήνα του λειτουργικού συστήματος της εικονικής μηχανής. Τι θα επιλέγατε μεταξύ μιας εικονικής μηχανής XEN και ενός container? Να δικαιολογήσετε την απάντησή σας.

Θα έπρεπε να επιλέξουμε μια εικονική μηχανή Xen καθώς σε αυτή την περίπτωση, κάθε φιλοξενούμενη εικονική μηχανή έχει το δικό της ΛΣ με ξεχωριστό πυρήνα, τον οποίο μπορούμε να τροποποιήσουμε. Αντιθέτως, τα containers μοιράζονται τον πυρήνα του ΛΣ και επομένως η τροποποίηση του πυρήνα του δεν είναι εφικτή.

Παραεικονικοποίηση με το Xen

- Dom0: εικονική μηχανή που χρησιμοποιεί ο διαχειριστής του υπολογιστή, ξεκινάει μαζί με τον υπολογιστή - είναι πάντα διαθέσιμη. Υπεύθυνη για διαχείριση συσκευών και εικονικών μηχανών (δημιουργία, μετακίνηση, τερματισμός εικονικών μηχανών, παρακολούθηση χρήσης πόρων)
- DomU: εικονικές μηχανές των πελατών/χρηστών, περιέχουν τις εφαρμογές που εκτελούν

2022-09, 2023-06

Εξηγήστε τη διαφορά μεταξύ του Dom0 και του DomU στο Xen. Σε τι είδους χρήστες απευθύνεται ένα Dom0 και ένα DomU?

Σύνδεση Εικονικών Μηχανών Xen στο Δίκτυο

- Η δικτυακή κίνηση περνάει μέσα από τον Dom0
- Υπάρχουν front end interfaces στα DomU και back end interfaces στο Dom0, τα οποία συνδέονται μέσω ειδικών καναλιών επικοινωνίας (δημιουργούνται αυτόματα από το Xen)
- Χρειάζεται σύνδεση της back end διεπαφής με τη φυσική διεπαφή (π.χ. μέσω γέφυρας Linux)

Εικονικά Δίκτυα και Συνδέσεις

Εικονικά Τοπικά Δίκτυα (VLAN) Δ6 10'

- Ένα τοπικό δίκτυο μπορεί να διαχωριστεί σε πολλά εικονικά τοπικά δίκτυα (VLANs). Το κάθε VLAN είναι απομονωμένο από τα υπόλοιπα
- VLAN tag στην επικεφαλίδα του κάθε πλαισίου Ethernet, περιέχει ένα μοναδικό ID (12 bits) για κάθε VLAN (δυνατότητα δημιουργίας έως και 4096 ξεχωριστών VLANs)
- Το switch διατηρεί ένα πίνακα με αντιστοιχίσεις θυρών που ανήκουν σε ένα VLAN και VLAN IDs
- Ένα μήνυμα broadcast περιορίζεται στις θύρες του VLAN στο οποίο ανήκει το πακέτο (απομόνωση της κίνησης)

Διαμόρφωση VLAN με περισσότερα Switches Δ6 26'

- Ξεχωριστή σύνδεση για κάθε VLAN
 - Σπατάλη θυρών σε κάθε switch
 - Δυσκολία να γίνουν αλλαγές στη διαμόρφωση (π.χ. προσθήκη ή αφαίρεση VLANs απαιτεί αλλαγές στην καλωδίωση μεταξύ των switches)
- Χρήση ενός trunk link (σύνδεση κορμού)
 - Όλα τα πλαίσια (ανεξαρτήτως VLAN ID) προωθούνται στη σύνδεση κορμού και φτάνουν στο άλλο switch (το switch ελέγχει την προώθηση στις θύρες του σωστού VLAN μέσω του πίνακα που διατηρεί)
 - Δεν απαιτούνται αλλαγές στην καλωδίωση αν αλλάξει η διαμόρφωση των VLANs

Εικονικές Συνδέσεις (Tunnel) Δ5 69'

Ένα Tunnel συνδέει δύο κόμβους (μεταξύ των οποίων συνήθως μεσολαβούν πολλές φυσικές συνδέσεις - δρομολογητές), δημιουργώντας την ψευδαίσθηση ότι οι δύο κόμβοι είναι απευθείας συνδεδεμένοι μεταξύ τους. Η δημιουργία εικονικών συνδέσεων συνήθως απαιτεί την ενθυλάκωση πακέτων (τεχνολογίες IP in IP, MPLS, GRE)

IP in IP Δ5 72'

Η τεχνική IP in IP ενθυλακώνει πακέτα IP εντός άλλων πακέτων IP (v4 ή v6). Μέσα στο tunnel η δρομολόγηση γίνεται βάσει της εξωτερικής επικεφαλίδας IP

- Ευρεία υποστήριξη, λόγω της καθολικής διάδοσης του IP
- Επιβάρυνση (overhead) λόγω της ενθυλάκωσης - επιπλέον επικεφαλίδα (20 bytes με το IPv4)

Multi Protocol Label Switching (MPLS) Δ6 37'

Η τεχνολογία MPLS ενθυλακώνει πακέτα IP σε πλαίσια (π.χ. Ethernet, PPP)

- Εισάγεται ένα επιπλέον στρώμα που παρεμβάλλεται μεταξύ των επικεφαλίδων IP και Ethernet, η προώθηση γίνεται βάσει πληροφορίας της επικεφαλίδας του MPLS (όχι του IP)
- Ταχεία αναζήτηση στον πίνακα δρομολόγησης με τη χρήση ενός label (20 bits) σταθερού μήκους (αντί για αναζήτηση του IP prefix/subnet που ταιριάζει καλύτερα)
- Υποστηρίζεται από τους περισσότερους παρόχους υπηρεσιών διαδικτύου (ISPs)

Προώθηση Πακέτων με το MPLS

Ο κάθε MPLS δρομολογητής (Label Switch Router LSR) διατηρεί **πίνακες δρομολόγησης** με την πληροφορία <in label, out label, prefix, output interface>

Η δρομολόγηση σε ένα LSR γίνεται ως εξής:

1. Εξάγεται η τιμή του label από την επικεφαλίδα MPLS του πακέτου
2. Εντοπίζεται η γραμμή του πίνακα δρομολόγησης που περιέχει το label του πακέτου στο πεδίο in label
3. Αντικαθιστάται η τιμή του label στο πακέτο με την τιμή του out label της γραμμής του πίνακα
4. Το πακέτο προωθείται στη θύρα που αντιστοιχεί στο output interface της γραμμής του πίνακα

Η πρώτη εισαγωγή της κεφαλίδας MPLS

Στην όρια του δικτύου MPLS τοποθετούνται δρομολογητές (Label Edge Routers LERs), διατηρούν ίδιους πίνακες δρομολόγησης με τους LSRs (το in label είναι κενό) και προωθούν πακέτα ως εξής (κατά την είσοδο σε ένα δίκτυο MPLS)

- Εξάγεται η διεύθυνση IP προορισμού από την επικεφαλίδα IP του πακέτου
- Εντοπίζεται η γραμμή του πίνακα δρομολόγησης της οποίας το prefix αντιστοιχεί στη διεύθυνση IP προορισμού του πακέτου
- Γίνεται ενθυλάκωση του πακέτου IP σε ένα πακέτο MPLS με το label να παίρνει την τιμή του out label της γραμμής του πίνακα
- Το πακέτο προωθείται στη θύρα που αντιστοιχεί στο output interface της γραμμής του πίνακα

2022-07

Σε ποιά περίπτωση χρησιμοποιείται το prefix στον πίνακα προώθησης ενός δρομολογητή MPLS κατά την προώθηση πακέτων?

Στα όρια του δικτύου MPLS, στους Label Edge Router, εξάγεται η διεύθυνση IP από το πακέτο και χρησιμοποιούνται τα prefixes του πίνακα προώθησης για να βρεθεί με ποιο prefix αντιστοιχεί η IP που εξήχθη. Αφού εντοπιστεί η σωστή γραμμή το πακέτο ενθυλακώνεται σε πλαίσιο MPLS με label την τιμή του out label της γραμμής του πίνακα και τέλος προωθείται στη θύρα που αντιστοιχεί στο output interface της γραμμής του πίνακα. Επιπλέον, το prefix στον πίνακα προώθησης μπορεί να χρησιμοποιηθεί σε περίπτωση που έχουμε κάποιο corruption σε κάποια επικεφαλίδα του MPLS, μπορεί να γίνει αναζήτηση με το IP.

Generic Routing Encapsulation (GRE) Δ6 76'

Πρωτόκολλο για ευέλικτη ενθυλάκωση πακέτων - επιτρέπει ενθυλάκωση οποιουδήποτε πρωτοκόλλου σε οποιοδήποτε πρωτόκολλο

- **Protocol Type:** Πληροφορία σχετικά με το πρωτόκολλο ή το πακέτο που έχει ενθυλακωθεί
- Διευκολύνει την αποπολυπλεξία μέσω **κλειδιού** (key) **4** bytes. Σε περίπτωση πολλαπλών εικονικών μηχανών σε έναν server, οι οποίες μοιράζονται ένα κοινό φυσικό interface (καθεμία έχει το δικό της εικονικό interface), το key αποθηκεύει την πληροφορία που συνδέει το πακέτο με κάποια εικονική μηχανή, πχ. ID της εικονικής μηχανής
- Μικρή επιβάρυνση (overhead): 4 bytes χωρίς τη χρήση των επιπλέον πεδίων της επικεφαλίδας

2023-01

Να αναφέρετε ένα παράδειγμα χρήσης του πεδίου κλειδιού της επικεφαλίδας του πρωτοκόλλου GRE

Virtual Private Networks (VPN) Δ7 8'

Υπηρεσία σύνδεσης σε ένα ιδιωτικό δίκτυο. Χαρακτηριστικά:

- Απομακρυσμένη πρόσβαση μέσω τεχνολογιών εικονικών συνδέσεων (tunnels)
- Ασφαλής πρόσβαση μέσω της κρυπτογράφησης των δεδομένων αποστολής και λήψης
- Διαχωρισμός της κίνησης από το υπόλοιπο Διαδίκτυο

Μοντέλα Υπηρεσιών για VPN Δ7 30'

Μοντέλο πελάτη: Η εγκατάσταση και διαχείριση των VPN servers γίνεται από έναν οργανισμό (π.χ. εταιρεία, πανεπιστήμιο) και η υπηρεσία απευθύνεται στους χρήστες/υπαλλήλους του οργανισμού. Το κόστος και τη διαδικασία εγκατάστασης και διαχείρισης τα αναλαμβάνει ο οργανισμός.

Μοντέλο παρόχου (PPVPN): Η εγκατάσταση και διαχείριση των VPN servers γίνεται από κάποιον πάροχο (π.χ. ISP), στο δίκτυο του και η υπηρεσία απευθύνεται προς το ευρύ κοινό. Ο χρήστης χρειάζεται μόνο κάποια credentials και την IP του VPN server για να έχει πρόσβαση στο δίκτυο VPN. Χρήση εικονικών δρομολογητών με ξεχωριστό πρωτόκολλο και πίνακα δρομολόγησης για τη δρομολόγηση της κίνησης από το ένα κομμάτι του VPN στο άλλο.

2023-09

Ποιες είναι οι διαφορές μεταξύ του μοντέλου παρόχου και του πελάτη στο VPN; Ποια είναι τα πλεονεκτήματα και σε ποιους χρήστες απευθύνεται το καθένα;

Φιλτράρισμα Διαδρομών σε VPN Δ7 34'

Ανακοινώσεις διαδρομών από κόμβους του VPN διαδίδονται στους υπόλοιπους κόμβους του VPN (και όχι στο υπόλοιπο διαδίκτυο - **απομόνωση** του VPN). Επιτυγχάνεται με ρυθμίσεις στους δρομολογητές (blocks, permits)

Εικονικές Δικτυακές Διεπαφές

Πολλές εικονικές μηχανές μπορεί να χρειαστεί να διαμοιραστούν μία δικτυακή διεπαφή: Τα πακέτα θα πρέπει να προωθηθούν στην εικονική μηχανή προορισμού. Η αποπολυπλεξία της κίνησης μπορεί να γίνει μέσω του GRE Key (VMid), μέσω του VLAN ID (VMId) ή dst_IP ή dst_MAC.

Ταξινόμηση Πακέτων μέσω Λογισμικού Δ7 49'

- Διαθέσιμες πλατφόρμες λογισμικού (Click Modular Router, OpenvSwitch)
- Δυνατότητα εφαρμογής εξειδικευμένων αλγορίθμων χρονοπρογραμματισμού (π.χ. weighted RR)
- Δυσκολία στην αξιοποίηση πολυπύρηνων επεξεργαστών

Click Modular Router Δ7 54'

Πλατφόρμα για την **προώθηση και επεξεργασία πακέτων**:

- Διευκολύνει την υλοποίηση π.χ. δρομολογητών, switches, firewalls σε υπολογιστές
- Παρέχει μία πληθώρα λειτουργιών (elements), π.χ. μεταφορά πακέτων από/προς δικτυακές θύρες, εισαγωγή σε ουρές, ταξινόμηση κ.ά., οι οποίες συνδυάζονται με τη μορφή γράφου
- Δυνατότητα τροποποίησης ή δημιουργίας νέων λειτουργιών επεξεργασίας πακέτων

Ταξινόμηση Πακέτων μέσω Λογισμικού στο Click Δ7 62'

1 TD per output

- Ένα TD element διαμοιράζεται μεταξύ των N εικονικών μηχανών
- Το TD χρησιμοποιεί έναν πυρήνα
- Περιορισμός λόγω επεξεργαστικής ισχύος του ενός πυρήνα

1 TD per core

- Ένα TD element ανά εικονική μηχανή
- Τα TD εκτελούνται σε N διαφορετικούς πυρήνες
- Περιορισμός λόγω ταυτοχρονισμού (απαιτείται συγχρονισμός - κλειδωμά NIC)

Ταξινόμηση Πακέτων μέσω Υλικού Δ7 68'

- Ταξινόμηση πακέτων από την κάρτα δικτύου χωρίς τη συνδρομή του επεξεργαστή (κάθε εικονική μηχανή έχει τη δική της εικονική ουρά στην είσοδο και στην έξοδο) (ελεύθερος επεξεργαστής)
- Περιορισμένη υποστήριξη αλγορίθμων χρονοπρογραμματισμού (π.χ. μόνο round robin)
- Περιορισμένος αριθμός εικονικών ουρών ανά δικτυακή επαφή (π.χ. 32 ουρές)

Τεχνολογίες Ταξινόμησης Πακέτων μέσω Υλικού Δ7 73'

Intel VMDq

Η δικτυακή διεπαφή περιλαμβάνει πολλές **εικονικοποιημένες δικτυακές διεπαφές** (vif) (εικονικές ουρές) που αντιστοιχούν σε εικονικές μηχανές. Η ταξινόμηση πακέτων σε εικονικές ουρές γίνεται βάσει της διεύθυνσης MAC προορισμού τους. Τα πακέτα προωθούνται από τις vif προς ένα **εικονικό switch** (διαμεσολάβηση επόπτη), το οποίο τα προωθεί προς την αντίστοιχη εικονική μηχανή.

- Ενδεχόμενη αλλαγή της σειράς των πακέτων σε μία δικτυακή ροή
- Απαιτεί τη μεσολάβηση CPU (το εικονικό switch εκτελείται στη CPU)

SR IOV (μετεξέλιξη του VMDq)

Η δικτυακή διεπαφή περιλαμβάνει πολλές **εικονικοποιημένες λειτουργίες** (VF) (εικονικές ουρές) που αντιστοιχούν σε εικονικές μηχανές και συνδέονται απευθείας με αυτές. Η ταξινόμηση πακέτων σε εικονικές ουρές γίνεται βάσει της διεύθυνσης MAC προορισμού τους. Υπάρχει ένας **VF Driver** σε κάθε εικονική μηχανή που επιτρέπει την απευθείας μεταφορά πακέτων από την εικονική ουρά στο κομμάτι της μνήμης όπου έχει πρόσβαση η εικονική μηχανή.

- **Απευθείας μεταφορά πακέτων** στην εικονική μηχανή (χωρίς τη διαμεσολάβηση του επόπτη)
- Απαιτείται υποστήριξη από τον επόπτη

Microsoft Receiver side Scaling (RSS)

- Ταξινόμηση πακέτων σε εικονικές ουρές με τη χρήση συνάρτησης hash
- Κάθε δικτυακή ροή προωθείται σε ξεχωριστή εικονική ουρά π.χ. μια εικονική ουρά μπορεί να οριστεί μέσω του 5-tuple

2019-06

Τι θα επιλέγατε μεταξύ VMDq και RSS για ταξινόμηση πακέτων μεταξύ εικονικών μηχανών σε έναν server? Γιατί?

Το RSS δεν ταιριάζει σε εικονικές μηχανές, καθώς είναι δύσκολο να συνδεθούν flows με εικονικές μηχανές. Μας ενδιαφέρει περισσότερο η ταξινόμηση πακέτων με βάση τη διεύθυνση MAC του παραλήπτη. Για το λόγο αυτό χρησιμοποιείται περισσότερο το VMDq

του οποίου η ταξινόμηση ταιριάζει περισσότερο στο **διαμοιρασμό μιας διεπαφής** μεταξύ εικονικών μηχανών.

2023_06

Εξηγήστε τη διαφορά μεταξύ των τεχνολογιών εικονικοποίησης δικτυακών διεπαφών VMDq και SR-IOV.

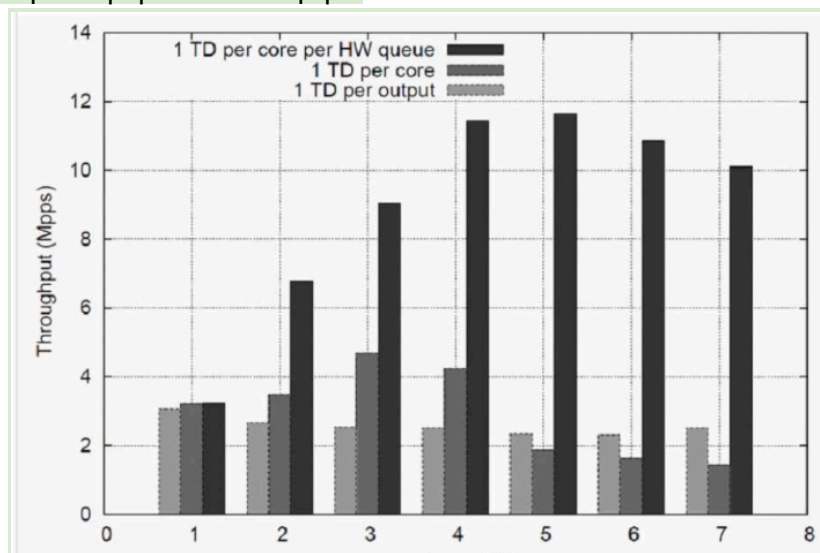
Απόδοση κατά την Ταξινόμηση Πακέτων Δ7 87'

2023_09

Θεωρείστε την παρακάτω τεχνική διαμοιρασμού (σε λογισμικό) μιας φυσικής δικτυακής διεπαφής μεταξύ πολλαπλών εικονικών μηχανών. Θεωρείστε ότι το ToDevice (TD) της κάθε εικονικής μηχανής εκτελείται σε ένα διαφορετικό πυρήνα CPU. Εξηγήστε τι πρόβλημα θα δημιουργηθεί όταν χρησιμοποιηθούν πυρήνες από διαφορετικές CPU (στον ίδιο εξυπηρετητή).

2024_02

Εξηγήστε που οφείλεται η πτώση της απόδοσης της 2ης τεχνικής ταξινόμησης (1TD per core) μετά την προσθήκη του 5ου πυρήνα



Έχουμε 2 επεξεργαστές από 4 πυρήνες ο καθένας. Η τιμή του lock (0,1) όταν χρησιμοποιούμε μόνο 4 πυρήνες (1 επεξεργαστής) αποθηκεύεται στην μνήμη cache του επεξεργαστή (κοινή για τους 4 πυρήνες). Μόλις μπει ο 5ος πυρήνας, αναγκαστικά η τιμή του lock θα αποθηκευτεί στη ram (ο δεύτερος επεξεργαστής δεν έχει πρόσβαση στην cache του πρώτου επεξεργαστή). Η πρόσβαση στη RAM είναι πολύ πιο αργή από ότι στην cache, άρα μεγαλύτερη καθυστέρηση που επηρεάζει την απόδοση.

Lab3 Tunnel Δ5 87'

2018-06

Έστω οι παρακάτω εντολές (που δίνονται στο PC1) για τη δημιουργία ενός tunnel μεταξύ των PC1 και PC2:

```
ip tunnel add tun0 mode ipip local 10.10.10.1 remote 10.10.10.2
ifconfig tun0 20.20.20.1 netmask 255.255.255.0 pointopoint 20.20.20.5
```

Ποιά θα είναι η διεύθυνση IP που θα εκχωρηθεί στη διεπαφή tun0 και eth0? Περιγράψτε σύντομα τη διαδικασία ενθυλάκωσης ενός πακέτου που στέλνεται από τη διεπαφή tun0 προς το PC2.



eth0 10.10.10.1

tun0 20.20.20.1

Στέλνω πακέτο με δ/σεις αποστολέα και παραλήπτη 20.20.20.1 και 20.20.20.5. Όταν φτάσει στο eth0 ενθυλακώνεται σε ένα άλλο πακέτο και εισάγεται μια εξωτερική επικεφαλίδα με δ/σεις αποστολέα και παραλήπτη 10.10.10.1 και 10.10.10.2.

Lab4 XenCenter Δ6 89'

Lab5 Kubernetes

2024-06

Εξηγήστε την έννοια του autoscaling στο Kubernetes. Τι παράμετροι εισάγονται κατά τον ορισμό μιας ενέργειας autoscaling?

Το autoscaling στο Kubernetes επιτρέπει την **αυτόματη προσαρμογή** του αριθμού των pods (HPA) ή των **πόρων που ανατίθενται** σε pods (VPA) ανάλογα με τον **φόρτο εργασίας**. Οι βασικές παράμετροι που εισάγονται περιλαμβάνουν τον **ελάχιστο και μέγιστο αριθμό των pods**, την **επιθυμητή χρήση CPU/μνήμης**, και τους **κανόνες scaling** (π.χ., scaling steps, cooldown periods).

4.Δίκτυα Οριζόμενα Μέσω Λογισμικού

Προβλήματα λόγω των Δρομολογητών Δ7 110'

Ενσωμάτωση του control plane στη δικτυακή συσκευή:

- δύσκολη τροποποίηση κάποιου υλοποιημένου πρωτοκόλλου δρομολόγησης
- αργή εξέλιξη των δικτύων
- εμπόδιο στην εφαρμογή καινοτόμων μηχανισμών για δρομολόγηση και διαχείριση δικτύων

Κατανεμημένη δρομολόγηση:

- Οι δικτυακές συσκευές δεν έχουν πλήρη εικόνα του δικτύου
- Ο υπολογισμός διαδρομών καθυστερεί σημαντικά

OpenFlow

- Το OpenFlow είναι ένα API για την κεντρική διαχείριση switch από έναν εξωτερικό ελεγκτή
- Το control plane αποσυνδέεται από το υλικό και μεταφέρεται στον ελεγκτή
- Το switch περιλαμβάνει ένα πίνακα ροών (flowtable), όπου ο ελεγκτής εισάγει κανόνες προώθησης, σύμφωνα με τους οποίους το switch αποφασίζει πως θα μεταχειριστεί κάθε εισερχόμενο πακέτο (προώθηση, αποστολή στον ελεγκτή, απόρριψη)

Βασικά Μηνύματα OpenFlow Δ9

Controller↔Switch

- Hello: Εγκαθίδρυση σύνδεσης
- Echo: Μέτρηση καθυστέρησης/ταχύτητας σύνδεσης, έλεγχος αν παραμένει ενεργή η σύνδεση

Controller→Switch

- Packet out: Υπόδειξη στο switch ενέργειας που θα πρέπει να εφαρμόσει σε ένα πακέτο
- Flow mod: Εισαγωγή/διαγραφή/τροποποίηση κανόνα προώθησης στον πίνακα ροών του switch

Switch→Controller:

- Packet in: Προώθηση επικεφαλίδας πακέτου προς τον controller, κάθε φορά που δεν υπάρχει αντιστοίχιση του πακέτου στον πίνακα ροών του switch (table miss)
- Flow removed: Ειδοποίηση controller κάθε φορά που έχει λήξει κάποιος κανόνας προώθησης και θα πρέπει να αφαιρεθεί από τον πίνακα ροών
- Port status: Ειδοποίηση κατά τη μεταβολή ρύθμισης ή κατάστασης σε κάποια θύρα του switch
- Error: Κοινοποίηση σφαλμάτων προς τον controller

Εισαγωγή Κανόνων Προώθησης

Εισαγωγή κανόνων **εκ των προτέρων** από τον controller

- Καμία καθυστέρηση κατά την προώθηση των ροών
- Απαιτείται εκ των προτέρων γνώση της κίνησης που θα διέλθει από κάθε switch
- Ενδεχόμενη σπατάλη χώρου στον πίνακα ροών

Εισαγωγή κανόνων **κατά την άφιξη των ροών** στο switch

- Αποδοτική χρήση του πίνακα ροών
- Επιφόρτιση του controller
- Καθυστέρηση κατά την προώθηση των ροών, καθώς το πρώτο πακέτο θα πρέπει να σταλεί στον controller

Software Defined Networks (SDN)

Δίκτυα Οριζόμενα μέσω Λογισμικού (SDN) Δ10 0'

Σε ένα SDN, το control plane τοποθετείται σε ένα εξωτερικό ελεγκτή, ο οποίος διαχειρίζεται ένα αριθμό από switches με τη χρήση πρωτοκόλλων, όπως το OpenFlow. Βασικά χαρακτηριστικά ενός SDN:

- Κεντρική διαχείριση του δικτύου
- Ορατότητα σε όλο το δίκτυο
- Απεικόνιση του δικτύου με διαφορετικές όψεις (για διευκόλυνση λειτουργιών του δικτύου)

Πλεονεκτήματα

- Ευκολία στην εγκατάσταση / αναβάθμιση πρωτοκόλλων δρομολόγησης
- Απλούστερη διαχείριση του δικτύου μέσω απλοποιημένων όψεων του δικτύου
- Σύγκλιση σε ένα πρότυπο (OpenFlow), το οποίο παρέχει πολλές νέες δυνατότητες

Μειονεκτήματα

- SDN ευρείας κλίμακας απαιτούν διαλειτουργικότητα (interoperability) μεταξύ πολλών ελεγκτών
- Ετερογενές υλικό μεταγωγής μπορεί να χρειαστεί χρονοδρομολόγηση κατά την εισαγωγή κανόνων προώθησης στους πίνακες ροών

FlowVisor

Τεμαχισμός SDN Δ10 22'

Το FlowVisor αποτελεί μία πλατφόρμα για τον τεμαχισμό δικτύων SDN. Επιτρέπει σε πολλούς χρήστες να αναλάβουν τη διαχείριση τμήματος ενός δικτύου μέσω του πρωτοκόλλου OpenFlow. Ο κάθε διαχειριστής:

- Χρησιμοποιεί ένα ξεχωριστό ελεγκτή
- Καθορίζει το τεμάχιο (slice) του δικτύου που θα διαχειριστεί
- Παρακολουθεί την κίνηση και παρεμβαίνει στον τρόπο προώθησης των πακέτων μόνο εντός του slice του

Λειτουργία του FlowVisor

Το FlowVisor λειτουργεί μεταξύ του switch και του ελεγκτή:

- Δεσμεύει τμήμα του πίνακα ροών και πόρους από τον επεξεργαστή του switch για κάθε ελεγκτή
- Ελέγχει τα δικαιώματα πρόσβασης σε κάθε μήνυμα ελέγχου
- Κάνει μετάφραση αναφορών στα μηνύματα ελέγχου, μετατρέπει εικονικά στοιχεία, στα αντίστοιχα φυσικά

Ορισμός Τεμαχίων (Slices) Δ10 36'

Κάθε slice περιλαμβάνει:

- Τμήμα της τοπολογίας (θύρες και switches) του δικτύου
- Τμήμα του πίνακα ροών
- Ποσοστό του διαθέσιμου bandwidth (των θυρών που διαμοιράζονται μεταξύ slices πχ με VMDo)
- Ποσοστό της επεξεργαστικής ισχύος του κάθε switch
- Τμήμα των ροών του δικτύου (flowspace) (εύρος MAC, IP Addresses και TCP ports - ευελιξία στη διαφοροποίηση μεταξύ υπηρεσιών)

Έλεγχος Πολιτικής (Policy Check) Δ10 45'

- Άφιξη **πακέτου** στο FlowVisor (από OpenFlow switch προς controller) χωρίς κάποια καταχώρηση στον πίνακα ροών
 - Το FlowVisor προωθεί το πακέτο στον Controller C γιατί σε αυτόν ανήκει η **ροή**
- Άφιξη **μηνύματος ελέγχου** στο FlowVisor (από controller προς OpenFlow switch)
 - Το FlowVisor προωθεί το πακέτο στο αντίστοιχο switch αν ο Controller C έχει **δικαίωμα πρόσβασης στη ροή**, αλλιώς το απορρίπτει

2022-09

Openflow με πολλους ελεγκτες, πως λειτουργει?

Lab6 OpenFlow Δ8 43'

Προσθήκη κανόνων προώθησης. Λειτουργία ως:

- **hub**: sudo ovs-ofctl add-flow s1 **actions=FLOOD**
- **switch**: sudo ovs-ofctl add-flow s1 **dl_dst=00:00:00:00:00:01,actions=output:1**
 - **ARP broadcast**: sudo ovs-ofctl add-flow s1 **dl_type=0x806,actions=flood**
- **router**: sudo ovs-ofctl add-flow s1
dl_type=0x800,nw_dst=10.0.0.1,actions=output:1

2022-09

Lab6 OpenFlow

2022-07

Ποιές δυνατότητες του OpenFlow (σε επίπεδο αντιστοίχισης και ενεργειών) θα αξιοποιούσατε για την υλοποίηση των βασικών λειτουργιών ενός δρομολογητή IP.

Προσθήκη κανόνων προώθησης από τον ελεγκτή στον πίνακα ροών του δρομολογητή, για την δρομολόγηση πακέτων IP. Οι κανόνες προώθησης θα πρέπει να περιλαμβάνουν την εξής πληροφορία:

- **dl_type**: πληροφορία για το ενθυλακωμένο πακέτο (IP)
- **nw_dst**: διεύθυνση IP προορισμού
- **action**: θύρα εξόδου στην οποία θα προωθηθεί το πακέτο

Lab7 POX Δ9

2023-09

Ποια είναι η λειτουργία του μηνύματος Packet_in στο OpenFlow; Από ποιον στέλνεται και προς τα που;

2024-06

Εξηγήστε τη χρησιμότητα του πεδίου buffer_id στο OpenFlow μήνυμα PACKET_IN Δ9 35'

2023-06

Εξηγήστε για ποιό λόγο γίνεται ο διαχωρισμός της επικεφαλίδας και των δεδομένων του πακέτου, όταν χρειαστεί να αποσταλεί κάποιο μήνυμα PACKET_IN από ένα OpenFlow switch προς τον Controller. Που διατηρούνται τα δεδομένα και πως διασφαλίζεται η ανάκτηση τους? Δ9 35'

2022-07

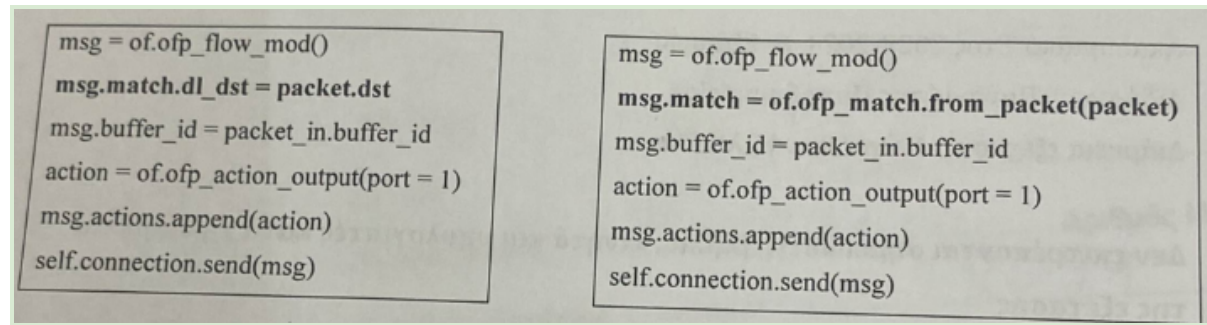
Θεωρήστε ένα PACKET_IN event, όπου η επικεφαλίδα ενός πακέτου προωθείται από ένα OpenFlow switch σε ένα OpenFlow controller, καθώς ο πίνακας ροών του switch δεν περιέχει κάποια πληροφορία για την προώθηση του συγκεκριμένου πακέτου. Υποθέτοντας ότι στον Controller εκτελείται κάποια εφαρμογή που είναι σε θέση να αποφασίσει τι θα συμβεί στο πακέτο, εξηγήστε πως θα γίνει η προώθηση ολόκληρου του πακέτου (επικεφαλίδα + δεδομένα) μέσω ενός μηνύματος PACKET_OUT. Δ9 47'

Αφού αποφασίσει ο controller το action που θα εφαρμοστεί στο πακέτο, δημιουργεί ένα μήνυμα PACKET_OUT το οποίο στέλνει στο switch. Το μήνυμα PACKET_OUT αντιγράφει τις τιμές των πεδίων data (επικεφαλίδα) και buffer_id από τα αντίστοιχα πεδία που υπήρχαν στο PACKET_IN. Επιπλέον, το πεδίο action(s) περιλαμβάνει την ενέργεια που θα πρέπει να εφαρμοστεί στο πακέτο. Όταν το switch λάβει το PACKET_OUT, επανασυνθέτει το packet το οποίο έχει σαν επικεφαλίδα το πεδίο data από το PACKET_OUT και σαν payload τα δεδομένα που υπάρχουν αποθηκευμένα στο buffer του switch και στα οποία δείχνει το πεδίο buffer_id. Αφού ολοκληρωθεί η σύνθεση του πακέτου, το switch εφαρμόζει στο πακέτο την ενέργεια που υπαγορεύει το πεδίο actions.

2024-06

Παρακάτω σας δίνονται δύο εκδόσεις κώδικα Python για τη δημιουργία ενός κανόνα προώθησης OpenFlow από τον ελεγκτή POX. Για διευκόλυνσή σας έχει επισημανθεί η γραμμή που διαφοροποιείται μεταξύ των δύο εκδόσεων κώδικα. Να εξηγήσετε ποιά είναι η

διαφορά στο περιεχόμενο των κανόνων προώθησης OpenFlow που θα δημιουργηθούν μεταξύ των δύο αυτών εκδόσεων κώδικα. Δ9 111'



Ο πίνακας που θα δημιουργηθεί από το δεξί θα έχει πολλές εγγραφές γιατί κάθε κανόνας προώθησης θα έχει τιμές σε όλα τα πεδία - όχι μόνο στο dl_dst (MAC προορισμού) (αντιγράφονται οι τιμές όλων των πεδίων του packet στο match και αυτό δημιουργεί πολλούς συνδυασμούς). Ο πίνακας που θα δημιουργηθεί από το αριστερό θα έχει λιγότερες εγγραφές γιατί οι κανόνες προώθησης θα έχουν τιμές μόνο στο dl_dst (MAC προορισμού)

5. Διαχείριση Εικονικών Μηχανών

Εικονικό Switch Δ10 63'

2019-06

Ποιά είναι τα πλεονεκτήματα χρήσης εικονικών switch σε εικονικοποιημένους servers για σύνδεση μηχανών στο διαδίκτυο?

- ευέλικτη σύνδεση των εικονικών μηχανών με το δίκτυο. Μπορούμε να προσθέσουμε κανόνες προώθησης που θα καθορίζουν με ποιον τρόπο θα γίνει η σύνδεση ενός εικονικού interface με ένα φυσικό interface. Δε χρειάζεται να ρυθμίζουμε ή να τερματίζουμε Linux bridges όταν ξεκινάμε/μετακινούμε/διαγράφουμε μια εικονική μηχανή, αλλά μπορούμε απλά και εύκολα να προσθέτουμε/αφαιρούμε κανόνες προώθησης μέσα στο virtual switch
- Μπορούμε να προγραμματίσουμε την απευθείας σύνδεση εικονικών μηχανών μέσω του virtual switch
- Επιπλέον υπάρχει η δυνατότητα ένας εξωτερικός controller να αναλάβει τη διαχείριση του vSwitch παρέχοντας κεντρικό έλεγχο και ορατότητα σε όλο το δίκτυο. Διευκολύνει τη διαχείριση των εικονικών μηχανών και τη σύνδεσή τους στο δίκτυο

OpenvSwitch Δ10 68'

Ποια διευκόλυνση παρέχει η συνεργασία του OpenvSwitch με το Xen?

Ο controller **διαβιβάζει αιτήματα** για προγραμματισμό του OpenvSwitch για τη σύνδεση θυρών εικονικών μηχανών με τις φυσικές θύρες του υπολογιστή ΚΑΙ αιτήματα στο XEN μέσω του API. Έχουμε **συντονισμό ενεργειών** πχ δημιουργία εικονικής μηχανής και σύνδεσή της με το δίκτυο. Διευκολύνει τη διαχείριση των εικονικών μηχανών και τη σύνδεσή τους στο δίκτυο.

Μετακίνηση Εικονικών Μηχανών

Οι **επόπτες** εικονικών εξυπηρετητών επιτρέπουν τη μετακίνηση εικονικών μηχανών μεταξύ εξυπηρετητών.

Λόγοι Μετακίνησης Εικονικών Μηχανών Δ11 5'

- **Αντιμετώπιση προβλημάτων** στην εικονική μηχανή (**σφάλματα** στο υλικό ή λογισμικό, **έλλειψη** υπολογιστικών πόρων, **χαμηλή** ποιότητα παροχής υπηρεσίας)
- **Εφαρμογή πολιτικών** του παρόχου υπολογιστικού νέφους (**εξισορρόπηση φορτίου**, συγκέντρωση των εικονικών μηχανών στο μικρότερο δυνατό αριθμό εξυπηρετητών για την **μείωση** της κατανάλωσης ενέργειας)
- **Συντήρηση/αναβάθμιση** εξυπηρετητή

Μετακίνηση Εικονικών Μηχανών Δ11 13'

Η μετακίνηση εικονικών μηχανών μπορεί να οδηγήσει σε διαστήματα **εκτός λειτουργίας** (downtime). Απαιτεί τη μεταφορά:

- των δεδομένων που είναι αποθηκευμένα στην κύρια μνήμη
- του συστήματος αρχείων (συνήθως αρκετά GB)
- των δικτυακών συνδέσεων

Μετακίνηση Χωρίς Κοινό Αποθηκευτικό Μέσο Δ11 18'

Τεχνική freeze and copy. Διαδικασία μετακίνησης: Αναστολή της λειτουργίας της εικονικής μηχανής, μεταφορά των δεδομένων της μνήμης και του συστήματος αρχείων στον εξυπηρετητή προορισμού, επανεκκίνηση της εικονικής μηχανής στο νέο εξυπηρετητή

- **Μεγάλος** χρόνος εκτός λειτουργίας
- Διασφαλίζεται η **συνοχή** και **εγκυρότητα** των δεδομένων στη μνήμη και του συστήματος αρχείων της εικονικής μηχανής.

Μεταφορά δεδομένων κατ'απαίτηση

Δεδομένα (π.χ. αρχεία) μεταφέρονται μόνο όταν χρειαστούν από κάποια εφαρμογή που εκτελείται στον εξυπηρετητή προορισμού (μεταφέρονται μόνο τα αρχεία που είναι απαραίτητα για την εκκίνηση της εικονικής μηχανής και τα αρχεία εφαρμογών ή υπηρεσιών θα μεταφερθούν όταν χρειαστούν)

- **Σύντομος** χρόνος εκτός λειτουργίας
- **Χαμηλή απόδοση εφαρμογών** λόγω της καθυστέρησης κατά τη μεταφορά των δεδομένων
- Ενδείκνυται για εφαρμογές που δεν εκτελούνται συχνά

Μεταφορά δεδομένων εκ των προτέρων (pre-copy)

Όλα τα δεδομένα (μνήμη, αρχεία) μεταφέρονται στον εξυπηρετητή προορισμού εκ των προτέρων, και αφού μεταφερθούν γίνεται η **μετάβαση** στην εκτέλεση της εικονικής μηχανής στον προορισμό

- **Σύντομος** χρόνος εκτός λειτουργίας
- **Καλύτερη απόδοση** εφαρμογών σε σχέση με τη μεταφορά κατ'απαίτηση
- **Πρόβλημα συνοχής** - τμήματα αρχείων που έχουν τροποποιηθεί (μετά την αρχική μεταφορά τους) πρέπει να μεταφερθούν ξανά στον εξυπηρετητή προορισμού (η τεχνική **ενδείκνυται όταν δεν έχουμε συχνές τροποποιήσεις** στα αρχεία)

2023-09

Θέλουμε να μετακινήσουμε μία εικονική μηχανή χωρίς αποθηκευτικό μέσο. Προτείνετε έναν τρόπο μείωσης του downtime (πχ. σε συνδυασμό με κάποια άλλη τεχνική μετακίνησης) της τεχνικής freeze-and-copy.

2022-07

Ποια τεχνική μετακίνησης εικονικών μηχανών θα χρησιμοποιούσατε για μια εικονική μηχανή που εκτελεί μια εφαρμογή με πολύ συχνές προσπελάσεις στο σκληρό δίσκο. Να δικαιολογήσετε την απάντησή σας.

Σε περίπτωση που έχουμε κάποια εφαρμογή που πραγματοποιεί συχνές προσπελάσεις στο δίσκο είναι καλύτερο να χρησιμοποιηθεί το freeze and copy - διασφαλίζει εγκυρότητα και συνοχή

Μετακίνηση με Κοινό Αποθηκευτικό Μέσο Δ11 29'

- Η ύπαρξη κοινού αποθηκευτικού μέσου (π.χ. NFS) **διευκολύνει** τη μετακίνηση εικονικών μηχανών. Δεν απαιτείται η μεταφορά του συστήματος αρχείων - είναι ήδη αποθηκευμένο στο κοινό αποθηκευτικό μέσο.
- **Ελάχιστος χρόνος αναμονής** που συνήθως δεν έχει κάποια επίπτωση στις εφαρμογές που τρέχουν στην εικονική μηχανή
- **Χαμηλότερη απόδοση** κατά τη μεταφορά δεδομένων από και προς το αποθηκευτικό μέσο (συγκριτικά με ένα τοπικό αποθηκευτικό μέσο)

Επίδραση Downtime σε Εφαρμογές

- Αν η εικονική μηχανή εκτελεί **εφαρμογές χρήστη**: Πιθανές διαταραχές κατά την εκτέλεση γίνονται αντιληπτές από το χρήστη της εικονικής μηχανής
- Αν η εικονική μηχανή εκτελεί **εφαρμογές εξυπηρετητή** ή peer to peer: Μεγαλύτερο αντίκτυπο. Προκαλούνται διαταραχές σε κάθε χρήστη που εξυπηρετείται από την εικονική μηχανή π.χ. διακοπή στην αναπαραγωγή βίντεο στην περίπτωση που στην εικονική μηχανή φιλοξενείται ένας streaming server

Κατανεμημένη Διαχείριση

Κεντρική Διαχείριση Δ11 41'

- Ευκολία στην υλοποίηση (Κεντρικός διαχειριστής με πλήρη γνώση και ορατότητα της υποδομής)
- Περιορισμένη κλίμακα
- Δύσκολη ανάκαμψη από σφάλματα

Κατανεμημένη Διαχείριση Δ11 44'

Έχουμε agents υλοποιημένα σε κάθε συσκευή, τα οποία επικοινωνούν μεταξύ τους και από κοινού λαμβάνουν αποφάσεις σχετικά με τη διαχείριση του δικτύου, π.χ. την επιλογή εξυπηρετητών για τη φιλοξενία μιας εικονικής μηχανής

- Πολυπλοκότητα στην υλοποίηση
- Κατανομή του φορτίου διαχείρισης - Ευκολία στην κλιμάκωση
- Ευρωστία κατά την εμφάνιση σφαλμάτων

Ομαδοποίηση Εξυπηρετητών Δ11 48'

Εξυπηρετητές με κοινές ιδιότητες ταξινομούνται σε μία ομάδα (**cluster**) (πχ servers που διαθέτουν GPU)

Αναζήτηση Εξυπηρετητή για Φιλοξενία VM Δ11 50'

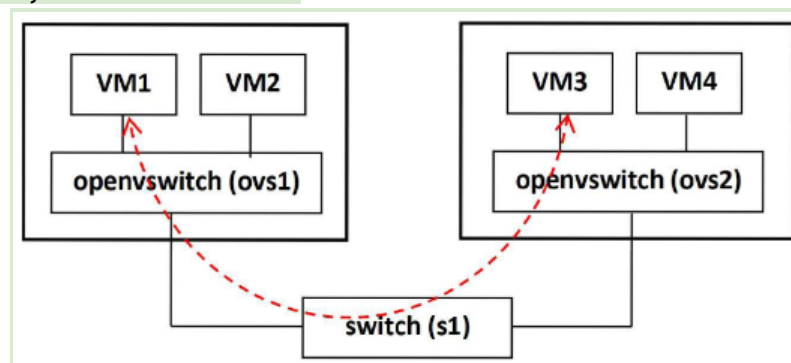
1. **Ειδοποίηση** όλων των πρακτόρων της ομάδας
2. **Υπολογισμός** του Δείκτη Ανομοιότητας ΔΑ (αξιολογηση καταλληλότητας server για τη φιλοξενία της εικονικής μηχανής)
3. **Ανταλλαγή των ΔΑ** μεταξύ των πρακτόρων της ομάδας

4. Ο κάθε πράκτορας **συγκρίνει** το δικό του ΔΑ με τους υπόλοιπους που έλαβε
5. Ο εξυπηρετητής με το μικρότερο ΔΑ **επιλέγεται** για τη φιλοξενία της εικονικής μηχανής

Lab8 OpenvSwitch Δ10 76'

2018-06, 2019-06, 2024-02

Έστω 2 servers καθένας από τους οποίους φιλοξενεί δύο εικονικές μηχανές. Οι εικονικές μηχανές συνδέονται μέσω δύο OpenvSwitch και ενός OpenFlow switch, όπως φαίνεται στο σχήμα. Θεωρώντας ότι όλοι οι πίνακες ροών είναι αρχικά κενοί, γράψτε τους κανόνες προώθησης που θα πρέπει να εισαχθούν στα ovs1, ovs2, και s1, ώστε να επιτρέπεται η επικοινωνία μεταξύ των VM1 και VM3.



ovs1:

```
sudo ovs-ofctl add-flow ovs1 dl_src=MAC(VM1),dl_dst=MAC(VM3),action:output=3
sudo ovs-ofctl add-flow ovs1 dl_src=MAC(VM3),dl_dst=MAC(VM1),action:output=1 ...etc...
```

6.Δρομολογητής ως Υπηρεσία

Software Routers

Δρομολογητής υλοποιημένος σε ένα υπολογιστή (π.χ. εξυπηρετητή)

- Προγραμματιζόμενο **data (forwarding) plane** (π.χ. με το Click Modular Router)
- Δυνατότητα επέκτασης του **control plane** (π.χ. με το XORP)
- Απόδοση περιορίζεται λόγω καθυστέρησης προσπέλασης στη μνήμη

Μέγιστη Ταχύτητα Προώθησης Πακέτων Δ12 21'

2019-06, 2023-06

Έστω ένας σύγχρονος εξυπηρετητής με διαύλους PCIe, κύρια μνήμη με πραγματική συχνότητα 500 MHz και κόστος 5 κύκλων ρολογιού για CAS. Θεωρώντας 10 προσπελάσεις στην κύρια μνήμη για κάθε πακέτο και ότι η κάθε προσπέλαση στη μνήμη σχετίζεται μόνο με CAS, υπολογίστε τη μέγιστη ταχύτητα προώθησης πακέτων από τον εξυπηρετητή. 10 Mrps

2022-07, 2023-01, 2024-06

Έστω ένας σύγχρονος εξυπηρετητής με διαύλους PCIe, κύρια μνήμη με πραγματική συχνότητα 500 MHz και κόστος 4 κύκλων ρολογιού για CAS και 6 κύκλων ρολογιού για RAS. Θεωρώντας 5 προσπελάσεις στην κύρια μνήμη για κάθε πακέτο και ότι η κάθε προσπέλαση στη μνήμη προϋποθέτει αλλαγή στήλης CAS και αλλαγή γραμμής RAS, υπολογίστε τη μέγιστη ταχύτητα προώθησης πακέτων από τον εξυπηρετητή

$f=500$, άρα $T = 1/500 = 0.002 \text{ s} = 2 \text{ ns}$ ο κάθε κύκλος ρολογιού

$2 \cdot (4+6) = 20 \text{ ns}$ η κάθε προσπέλαση στη μνήμη, $20 \cdot 5 = 100 \text{ ns}$ η κάθε προώθηση πακέτου

$1/(100 \cdot 10^9) = 10.000.000$ πακέτα ανά s = 10 Mpps

Εικονικοί Δρομολογητές

Εικονική μηχανή που εκτελεί λογισμικό δρομολόγησης (εκχωρούνται σε χρήστες). Χρήση τεχνολογιών εικονικοποίησης εξυπηρετητών (π.χ. Xen) για το **διαμοιρασμό πόρων** και την **απομόνωση** (isolation)

Εικονικοί Δρομολογητές με το Xen Δ12 35'

Πρώθηση μέσω Καναλιών E/E του Xen

Η κίνηση περνάει μέσα από το management domain (Dom0) - τα Forwarding Planes εκτελούνται στους πυρήνες για υψηλότερη απόδοση, τα control planes στα userspace των εικονικών δρομολογητών

- Υψηλό επίπεδο απομόνωσης (κάθε δρομολογητής παραμένει υλοποιημένος εντός μιας ξεχωριστής εικονικής μηχανής και ο επόπτης διασφαλίζει το διαμοιρασμό πόρων)
- Χαμηλή απόδοση (λόγω καθυστέρησης του επόπτη κατά την αλλαγή μεταξύ domains)

Πρώθηση εντός ενός Κοινού Domain

τα Forwarding Planes εκτελούνται σε ένα **κοινό** Forwarding Domain (εικονική μηχανή)

- Υψηλή απόδοση (αποφεύγουμε τις εναλλαγές domain)
- Χαμηλό επίπεδο απομόνωσης (λόγω του κοινού Forwarding Domain) - είναι εκτός της εποπτείας του επόπτη - απαιτείται χρονοπρογραμματισμός στο Forwarding Domain της χρήσης της CPU από τους δρομολογητές

Απευθείας Σύνδεση στο Δίκτυο

το Forwarding Plane κάθε domain συνδέεται απευθείας με τη δικτυακή θύρα - Χρειάζονται κάρτες δικτύου με πολλές εικονικές ουρές για τη φιλοξενία πολλών εικονικών δρομολογητών

- Υψηλή απόδοση (η κίνηση δεν περνάει μέσα από το management domain)
- Υψηλό επίπεδο απομόνωσης

Μετακίνηση Εικονικών Δρομολογητών Δ12 49'

Η μετακίνηση ενός εικονικού δρομολογητή προϋποθέτει τη μεταφορά:

- του data plane
- του control plane
- των συνδέσεων δικτύου στο δρομολογητή

Μεταφορά Data Plane

- περιέχει μία βάση (Forwarding Information Base - FIB) όπου αποθηκεύεται ο πίνακας πρώτωσης
- το data plane δεν λειτουργεί κατά τη μεταφορά, οπότε δεν θα είναι δυνατή η πρώτωση πακέτων (με αποτέλεσμα την απώλειά τους)

Μεταφορά Control Plane

- Απαιτεί τη μεταφορά:
 - του image, δηλ. τα προγράμματα δρομολόγησης και τις ρυθμίσεις
 - της βάσης (Routing Information Base - RIB) όπου αποθηκεύεται ο πίνακας δρομολόγησης
- το control plane δεν λειτουργεί κατά τη μεταφορά, οπότε όλες οι ενημερώσεις δρομολόγησης που λαμβάνονται θα χαθούν, άρα τα πρωτόκολλα δρομολόγησης θα χρειαστεί να επανυπολογίσουν τις διαδρομές (σημαντικό αντίκτυπο σε περιπτώσεις

που ο εικονικός δρομολογητής συμμετέχει σε κάποια συνεδρία BGP, καθώς αυτή μπορεί να τερματιστεί)

Virtual Routers on the Move (VROOM) Δ12 55'

Κύριοι στόχοι:

- Ελαχιστοποίηση των διαταραχών κατά τη μετακίνηση
- Ελάχιστες ή και καθόλου απώλειες πακέτων και μηνυμάτων δρομολόγησης
- Διατήρηση της τοπολογίας του δικτύου

Μετακίνηση Δρομολογητών με το VROOM Δ12 56'

1. Δημιουργία **tunnels** για την επαναπροώθηση των μηνυμάτων δρομολόγησης από τον αρχικό server στον server προορισμού

2. **Μετακίνηση του control plane** στον προορισμό

- Μετακίνηση με pre copy των αρχείων ρυθμίσεων του διαχειριστή του δρομολογητή και της μνήμης (RIB) (Τα υπόλοιπα αρχεία θεωρείται ότι υπάρχουν στον υπολογιστή προορισμού)
- μετακίνηση με freeze and copy όλων των τροποποιημένων σελίδων μνήμης

Σύντομο downtime κατά τη μεταφορά του control plane - χάνονται κάποια μηνύματα δρομολόγησης που θα έπρεπε να δρομολογηθούν στο control plane. Το data plane λειτουργεί κανονικά - προωθεί πακέτα προς τους υπόλοιπους κόμβους του δικτύου

3. **Επαναπροώθηση των μηνυμάτων δρομολόγησης** στον προορισμό (το control plane στον προορισμό είναι σε θέση να ενημερώνει τη FIB στον αρχικό υπολογιστή, κάθε φορά που λαμβάνεται ένα νέο μήνυμα δρομολόγησης)

4. **Επαναδημιουργία του data plane** από το control plane στον προορισμό. Ενδεχόμενη μετακίνηση του επιπέδου δεδομένων στον προορισμό είναι περιττή, η FIB μπορεί να σχηματιστεί απευθείας από τη RIB στον προορισμό (Διαφορετικές τεχνολογίες μνήμης (π.χ. TCAM, SRAM) μεταξύ των δύο υπολογιστών μπορούν να περιπλέξουν ενδεχόμενη μετακίνηση)

5. **Μεταφορά των συνδέσεων δικτύου** στον προορισμό. Το διπλό data plane επιτρέπει την **ασύγχρονη** μεταφορά τους (η εισερχόμενη και εξερχόμενη σύνδεση μπορούν να μεταφερθούν ξεχωριστά)

6. **Κατάργηση του data plane και των tunnels** στον αρχικό υπολογιστή. Μετά την ολοκλήρωση της μετακίνησης το παλιό data plane και τα tunnels δεν χρειάζονται πλέον

2023-01, 2023-06

Σε ποιο στάδιο της μετακίνησης δρομολογητών με το VROOM υπάρχει περίπτωση να χαθούν πακέτα και γιατί?

2024-06

Ποιά είναι η χρησιμότητα των tunnel ανακατεύθυνσης κατά τη μετακίνηση εικονικών δρομολογητών με το VROOM? Εξηγήστε τι είδους μηνύματα ή πακέτα μπορούν να μεταφέρονται σε κάθε κατεύθυνση

- Για τον καταμετρημένο υπολογισμό διαδρομών απαιτείται η ανταλλαγή μηνυμάτων μεταξύ των δρομολογητών, **επικοινωνούν για να μάθουν την τοπολογία του δικτύου** και μετά μέσω πρωτοκόλλων δρομολόγησης υπολογίζουν διαδρομές
 - Δρομολόγηση εντός ενός δικτύου: OSPF, RIP, IS-IS
 - Δρομολόγηση διαμέσου διαφορετικών δικτύων: BGP

- Κάθε πρωτόκολλο δρομολόγησης **ενημερώνει τον πίνακα προώθησης** του data plane **για κάθε νέα διαδρομή** (ή υπάρχουσα διαδρομή που έχει υποστεί μεταβολή)
- Μέσω των **ανακοινώσεων διαδρομών** ο κάθε δρομολογητής μαθαίνει διαδρομές τις αποθηκεύει στον πίνακα δρομολόγησης
- το control plane στον προορισμό **ενημερώνει τη FIB** στον αρχικό υπολογιστή, κάθε φορά που λαμβάνεται ένα νέο μήνυμα δρομολόγησης (ανακοίνωση διαδρομής)

7. Δικτυακή Λειτουργία ως Υπηρεσία Νέφους

Δικτυακές Λειτουργίες σε Λογισμικό

Δικτυακές λειτουργίες μπορούν να υλοποιηθούν σε λογισμικό:

- Μειωμένο κόστος
- Εγκατάσταση δικτυακών λειτουργιών κατά απαίτηση του πελάτη
- Δυνατότητα επιλογής μεταξύ διαφόρων λειτουργιών που προσφέρει ο πάροχος

Εικονικοποίηση Δικτυακών Λειτουργιών (NFV) Δ12 74'

Η εικονικοποίηση δικτυακών λειτουργιών επιτρέπει τη φιλοξενία πολλών δικτυακών λειτουργιών σε ένα εξυπηρετητή. Κάθε δικτυακή λειτουργία τοποθετείται σε μία ξεχωριστή εικονική μηχανή. Η δικτυακή κίνηση περνάει μέσα από τη Δικτυακή Λειτουργία και φιλτραρεται πχ firewall. Τεχνολογίες εικονικών εξυπηρετητών (π.χ. Xen) επιτρέπουν:

- Το **διαμοιρασμό** των πόρων μεταξύ των εικονικών δικτυακών λειτουργιών
- Την **απομόνωση** μεταξύ των εικονικών δικτυακών λειτουργιών

Source Routing

- Υπολογισμός διαδρομής: Κεντρικός ελεγκτής (π.χ. OpenFlow)
- Εισαγωγή της διαδρομής στα ToR switch
- Προώθηση πακέτων: Εναλλαγή θυρών στα ενδιάμεσα switch

Απαιτήσεις σε Χώρο Επικεφαλίδας

2018-06, 2022-07, 2023_01, 2023_09, 2024_02

Έστω ότι θέλετε να εφαρμόσετε μια τεχνική source routing με την ενθυλάκωση της ακολουθίας των θυρών στο πακέτο και τη χρήση ενός δείκτη στην επόμενη θύρα. Υπολογίστε το χώρο που απαιτείται στην επικεφαλίδα του πακέτου, αν

- τα switch έχουν ως 64 θύρες και το μέγιστο μήκος διαδρομής στο δίκτυο είναι 10 hops (**64 bits**)
- τα switch έχουν έως 32 θύρες και το μέγιστο μήκος διαδρομής στο δίκτυο είναι 8 hops (**43 bits**)
- τα switch έχουν έως 64 θύρες και το μέγιστο μήκος διαδρομής στο δίκτυο είναι 16 hops (**100 bits**)

Να αναφέρετε ένα πεδίο επικεφαλίδας του IP ή Ethernet, όπου θα μπορούσε να αποθηκευτεί αυτή η πληροφορία (43 bits) (θύρες + pointer)

Τα 43 bits μπορούν να χωρέσουν στο πεδίο επικεφαλίδας MAC αποστολέα ενός πακέτου Ethernet

Σύγκριση Τρόπων Εγκατάστασης Δικτυακών Λειτουργιών

Εγκατάσταση σε κέντρα δεδομένων

- Η κάθε δικτυακή λειτουργία λαμβάνει **μόνο** τα πακέτα προς επεξεργασία

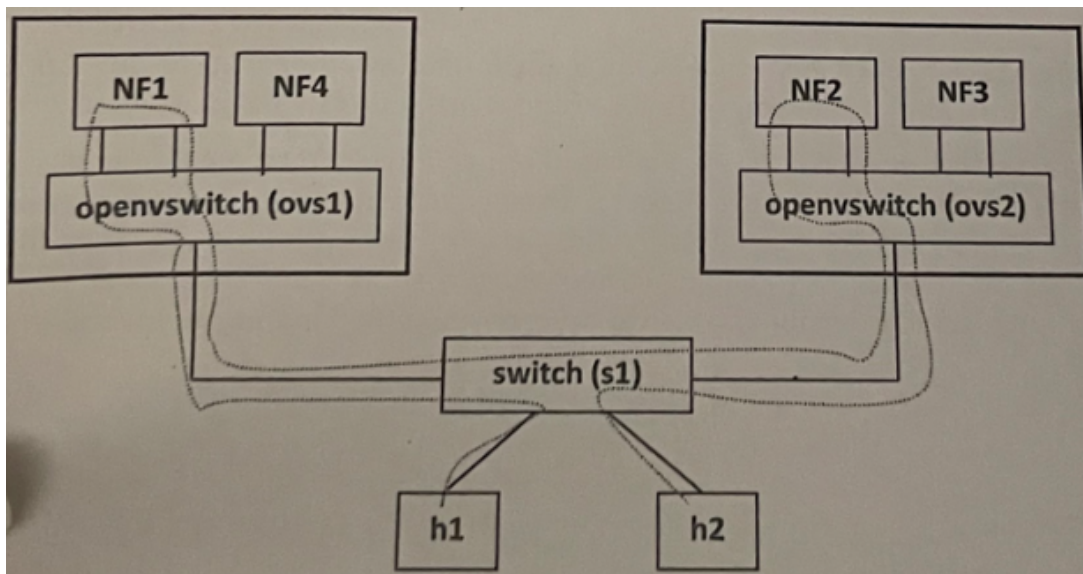
- Απαιτούνται **τεχνικές για δρομολόγηση** των πακέτων στα κέντρα δεδομένων. Μεγαλύτερη καθυστέρηση λόγω μακρύτερης διαδρομής
- Υψηλό φορτίο στις γραμμές πρόσβασης προς τα κέντρα δεδομένων

Εγκατάσταση κατά μήκος της διαδρομής

- Δεν απαιτείται **ανακατεύθυνση** των πακέτων
- **Σπατάλη πόρων** - Η κάθε δικτυακή λειτουργία μπορεί να λάβει και πακέτα που δεν χρειάζονται (από αυτήν) επεξεργασία και πρέπει απλώς να προωθηθούν

2019-06, 2023_01, 2023_06

Έστω οι παρακάτω δύο εξυπηρετητές που φιλοξενούν δικτυακές λειτουργίες (NF). (α) Εξηγήστε ποιά δυσκολία υπάρχει για τη δρομολόγηση της κίνησης από τον h1 προς τον h2, διαμέσου των NF1 και NF2, όπως φαίνεται στο παρακάτω σχήμα. (β) Προτείνετε κάποια λύση για την αντιμετώπιση αυτού του προβλήματος



(α) Το ίδιο πακέτο θα χρειαστεί να διασχίσει τα switches πολλές φορές και σε κάθε πέρασμα θα χρειαστεί να εφαρμοστεί διαφορετική ενέργεια. Εάν το πακέτο έχει κάθε φορά τα ίδια στοιχεία, τα switches δε μπορούν να ξεχωρίσουν τις διαφορετικές διελεύσεις του πακέτου. Χρειάζεται εισαγωγή επιπλέον πληροφορίας (tagging) στο πακέτο, καθώς δεν αρκούν τα στοιχεία στα υπάρχοντα πεδία επικεφαλίδας

(β) Source routing με labels και pointer