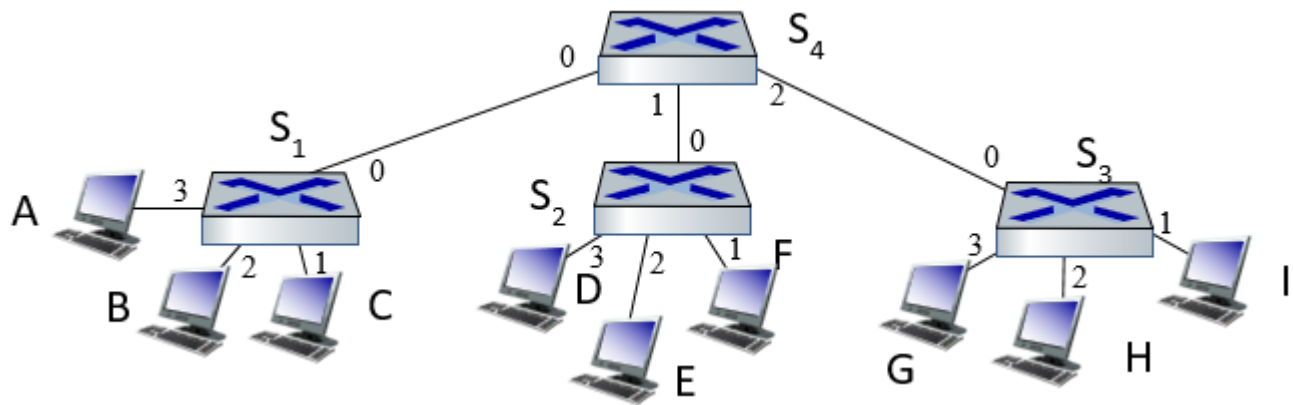


ΤΕΧΝΟΛΟΓΙΕΣ ΚΑΙ ΥΠΗΡΕΣΙΕΣ ΔΙΑΔΙΚΤΥΟΥ

ΕΝΔΕΙΚΤΙΚΕΣ ΑΠΑΝΤΗΣΕΙΣ ΑΣΚΗΣΕΩΝ ΠΡΟΣ ΜΕΛΕΤΗ – 2^ο Μέρος

1. Θεωρήστε το παρακάτω τοπικό δίκτυο (LAN) διασυνδεδεμένων self-learning switches (S1-S4). Υποθέστε ότι οι πίνακες προώθησης των switches είναι αρχικά κενοί. Καταγράψτε τις θύρες των switches στις οποίες θα προωθηθεί κάθε πακέτο στις ακόλουθες διαδοχικές επικοινωνίες μεταξύ των κόμβων που συνδέονται στα switches και δείξτε πως γεμίζουν με πληροφορία οι πίνακες προώθησης.



- i. Ο A στέλνει πακέτο στον D.
- ii. Ο G στέλνει πακέτο στον H.
- iii. Ο F στέλνει πακέτο στον A.
- iv. Ο I στέλνει πακέτο στον G.
- v. Ο C στέλνει πακέτο στον I.

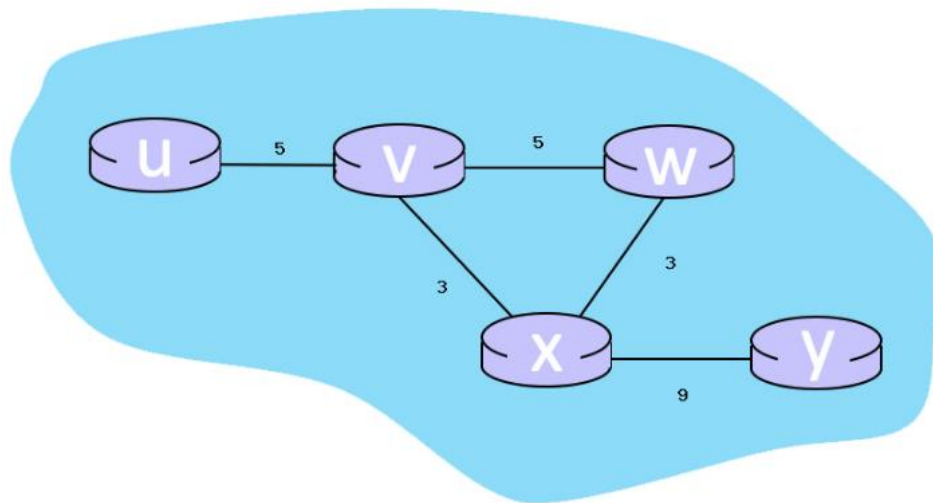
Απάντηση:

Αρχικά οι πίνακες προώθησης και κατακερματισμού των τεσσάρων switches είναι κενοί. Καθώς πραγματοποιούνται οι μεταδόσεις γεμίζουν σταδιακά.

ΜΕΤΑΔΟΣΗ	S1 – ΘΥΡΕΣ ΠΡΟΩΘΗΣΗΣ	S2 – ΘΥΡΕΣ ΠΡΟΩΘΗΣΗΣ	S3 – ΘΥΡΕΣ ΠΡΟΩΘΗΣΗΣ	S4 – ΘΥΡΕΣ ΠΡΟΩΘΗΣΗΣ
i.	0, 1, 2	1, 2, 3	1, 2, 3	1, 2
ii.	1, 2, 3	1, 2, 3	0, 1, 2	0, 1
iii.	3	0	Δεν το λαμβάνει	0
iv.	Δεν το λαμβάνει	Δεν το λαμβάνει	3	Δεν το λαμβάνει
v.	0, 2, 3	1, 2, 3	1	1, 2

ΜΕΤΑΔΟΣΗ	S1 – ΠΙΝΑΚΑΣ ΠΡΟΩΘΗΣΗΣ	S2 – ΠΙΝΑΚΑΣ ΠΡΟΩΘΗΣΗΣ	S3 – ΠΙΝΑΚΑΣ ΠΡΟΩΘΗΣΗΣ	S4 – ΠΙΝΑΚΑΣ ΠΡΟΩΘΗΣΗΣ
i.	A: 3	A: 0	A: 0	A: 0
ii.	G: 0	G: 0	G: 3	G: 2
iii.	F: 0	F: 1	-	F: 1
iv.	-	-	I: 1	-
v.	C:1	C: 0	C: 0	C: 0

2. Θεωρήστε τον ακόλουθο Γράφο ενός δικτύου **6 δρομολογητών**, με τα κόστη των γραμμών σύνδεσης μεταξύ τους. Στο δίκτυο υλοποιείται αλγόριθμος δρομολόγησης διανυσμάτων απόστασης **Bellman-Ford**.

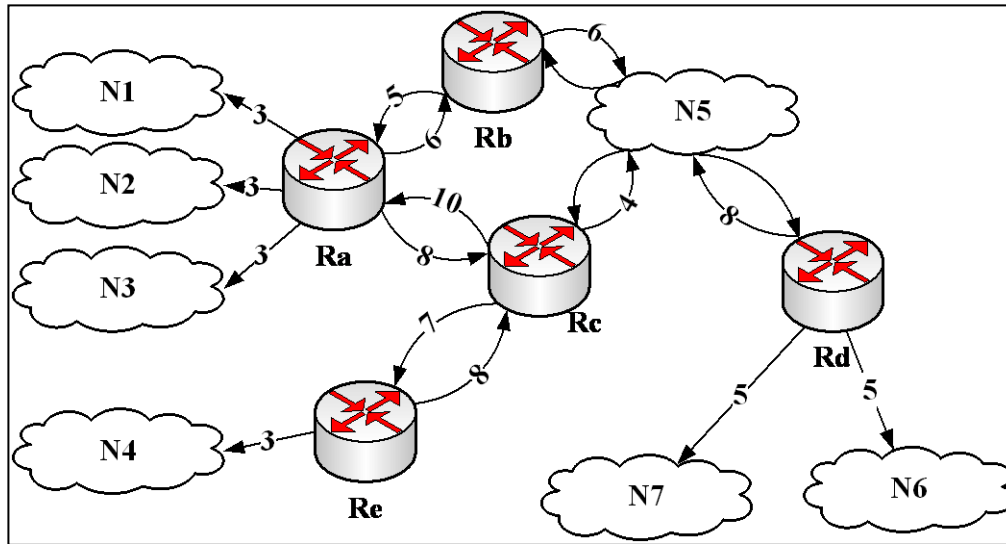


- Όταν ο αλγόριθμος συγκλίνει, ποια είναι τα διανύσματα απόστασης του δρομολογητή **y** από όλους τους υπόλοιπους;
- Ποια είναι τα αρχικά διανύσματα απόστασης από το δρομολογητή **v** προς όλους τους υπόλοιπους;
- Ποιο πρόβλημα δημιουργείται κατά την εκτέλεση του αλγορίθμου όταν προκύπτει αποτυχία σε μία γραμμή σύνδεσης και το κόστος της αυξάνει;

Απάντηση:

- Όταν ο αλγόριθμος **συγκλίνει** τα διανύσματα απόστασης του **y** είναι: $(u, v, w, x, y) = (17, 12, 12, 9, 0)$
 1^ο βήμα: $(u, v, w, x, y) = (\infty, \infty, \infty, 9, 0)$
 2^ο βήμα: $(u, v, w, x, y) = (\infty, 12, 2, 9, 0)$
 3^ο βήμα: $(u, v, w, x, y) = (17, 12, 12, 9, 0)$
- Τα αρχικά διανύσματα απόστασης από το δρομολογητή **v** είναι: $(u, v, w, x, y) = (5, 0, 5, 3, \infty)$
- Το πρόβλημα είναι γνωστό ως “count to infinity”

3. Στο σχήμα δίνεται ο κατευθυνόμενος γράφος ενός δικτύου στο οποίο εφαρμόζεται ο αλγόριθμος **OSPF**. Να σχεδιαστεί το δέντρο Βραχύτερο Μονοπατιού (SPF-Shortest Path First) από τον **Rc** προς το σύνολο των δικτύων N1 έως N7 και να υπολογιστεί το κόστος των αντίστοιχων μονοπατιών.



Απάντηση:

Το πρωτόκολλο OSPF υλοποιεί τον Αλγόριθμο δρομολόγησης του Dijkstra.

Step	N'	D(a),p	D(b),p	D(d),p	D(e),p	D(N1),p	D(N2),p	D(N3),p	D(N4),p	D(N5),p	D(N6),p	D(N7), p
0	Rc	10,Rc	4,Rc	4,Rc	7,Rc	∞	∞	∞	∞	4,Rc	∞	∞
1	RcN5Rb	9,Rb	4,Rc	4,Rc	7,Rc	∞	∞	∞	∞		∞	∞
2	RcN5RbRd	9,Rb			7,Rc	∞	∞	∞	∞		9,Rd	9,Rd
3	RcN5RbRdRe	9,Rb				∞	∞	∞	10,Re			
4	RcN5RbRdReRa					12,Ra	12,Ra	12,Ra				

Rc → N1, Διαδρομή: Rc-(N5)-Rb-Ra-N1, Κόστος: 4+0+5+3=12

Rc → N2, Διαδρομή: Rc-(N5)-Rb-Ra-N2, Κόστος: 4+0+5+3=12

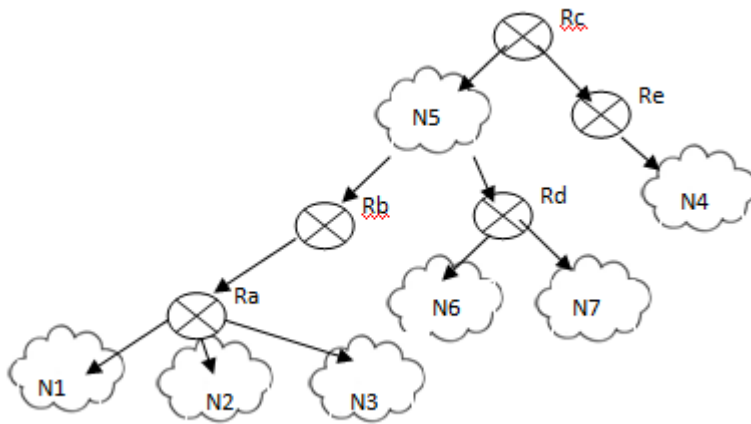
Rc → N3, Διαδρομή: Rc-(N5)-Rb-Ra-N3, Κόστος: 4+0+5+3=12

Rc → N4, Διαδρομή: Rc-Re-N4, Κόστος: 7+3=10

Rc → N5, κόστος: 4

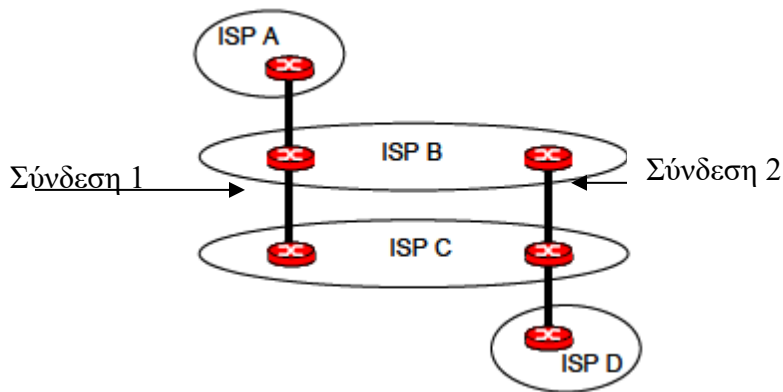
Rc → N6, Διαδρομή: Rc-(N5)-Rd- N6, Κόστος: 4+0+5=9

Rc → N7, Διαδρομή: Rc-(N5)-Rd- N7, Κόστος: 4+0+5=9



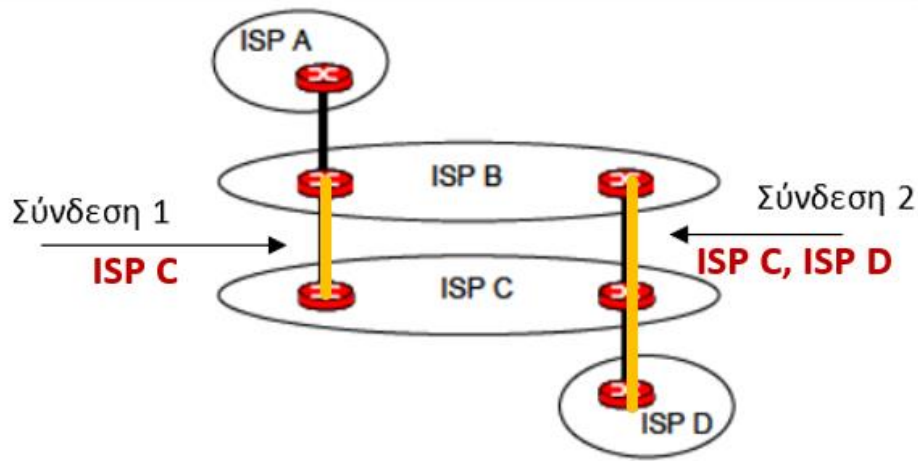
Δέντρο Βραχύτερου Μονοπατιού

4. Θεωρείστε τα δίκτυα του παρακάτω σχήματος. Ο πάροχος ISP B παρέχει δίκτυο κορμού στον πάροχο ISP A. Ο πάροχος ISP C παρέχει δίκτυο κόρμου στον πάροχο ISP D. Τα δίκτυα B και C συνδέονται σε δύο σημεία (Σύνδεση 1 και Σύνδεση 2) και υλοποιούν για τη μεταξύ τους επικοινωνία το πρωτόκολλο **BGP**. Έστω ότι ο ISP A στέλνει κίνηση προς τον ISP D. Ο ISP B θα επιθυμούσε να διεκπεραιώνει αυτή την κίνηση προς τον ISP C μέσω της Σύνδεσης 1 που έχει με τον ISP C (έτσι ώστε να επωμισθεί ο C το κόστος της κίνησης μέσα από το δίκτυό του προκειμένου να φτάσει στον ISP D). Αντίστοιχα ο ISP C θα προτιμούσε η κίνηση που του προωθεί ο B να εισέρχεται στο δίκτυό του μέσω της Σύνδεσης 2 που έχει με τον ISP B. Τί πρέπει να κάνει ο ISP C ώστε να επωμισθεί ο B το κόστος διέλευσης της κίνησης απ' άκρη σε άκρη στο δίκτυό του;

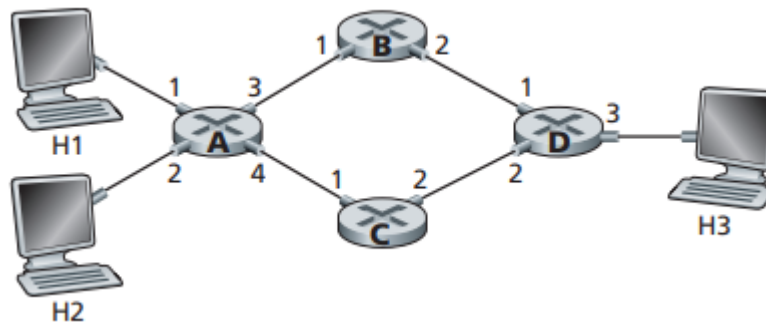


Απάντηση:

Ένας τρόπος να «εξαναγκάσει» τον ISP B να χρησιμοποιήσει τη Σύνδεση 2 είναι να αναγγέλει μόνο τη διαδρομή που περνάει από το δρομολογητή που υλοποιεί τη Σύνδεση 2 ως δυνατή διαδρομή προς τον ISP D.



5. Θεωρήστε το παρακάτω δίκτυο.

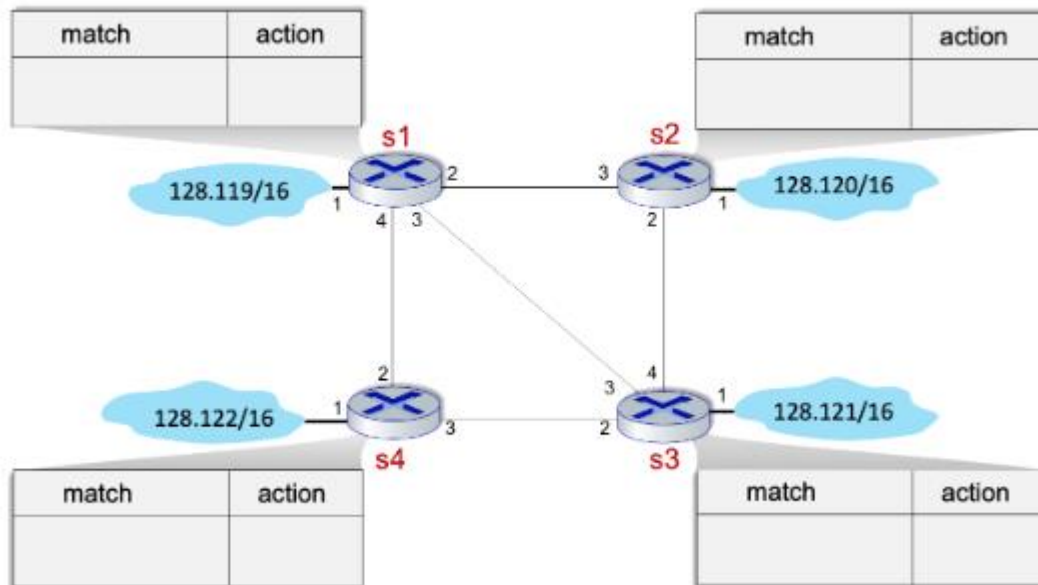


- i. Πώς θα είναι ο πίνακας προώθησης του δρομολογητή A, ώστε όλη η κίνηση η οποία έχει προορισμό τον κόμβο H3 να προωθείται μέσω του interface 3;
- ii. Μπορείτε να προσδιορίσετε τον πίνακα προώθησης του δρομολογητή A, ώστε όλη η κίνηση που προέρχεται από τον κόμβο H1 και έχει προορισμό τον κόμβο H3 να προωθείται μέσω του interface 3, ενώ όλη η κίνηση από τον κόμβο H2 με προορισμό τον κόμβο H3 να προωθείται μέσω του interface 4;

Απάντηση:

- i. Όλα τα πακέτα με IP διεύθυνση προορισμού του H3, προωθούνται στο interface 3.
- ii. Δεν μπορούν να προσδιοριστούν τέτοιου είδους κανόνες, δεδομένου ότι η κλασσική δρομολόγηση βασίζεται μόνο στη διεύθυνση προορισμού.

6. Θεωρήστε ένα δίκτυο 4 δρομολογητών, όπου η προώθηση πακέτων πραγματοποιείται με πίνακες ροής (flow tables), οι οποίοι δημιουργούνται με χρήση **OpenFlow** σε έναν **SDN controller**. Οι IP διευθύνσεις των δικτύων στα οποία συνδέεται κάθε δρομολογητής φαίνονται στο σχήμα, καθώς επίσης και τα interfaces των δρομολογητών.



Υποθέστε ότι επιθυμούμε να υλοποιούνται τα ακόλουθα όσον αφορά την προώθηση των πακέτων:

- Τα TCP πακέτα που φτάνουν στον s2 από το 128.120/16 και έχουν προορισμό το δίκτυο 128.121/16, θα πρέπει να προωθούνται μέσω της διαδρομής: s2 → s1 → s3.
- Τα UDP πακέτα που φτάνουν στον s2 από το 128.120/16 και έχουν προορισμό το δίκτυο 128.121/16, θα πρέπει να προωθούνται μέσω της διαδρομής: s2 → s1 → s4 → s3

Συμπληρώστε τους match+action πίνακες σε κάθε έναν από τους δρομολογητές s1, s2, s3 και s4, έτσι ώστε να υλοποιούνται οι παραπάνω πολιτικές προώθησης πακέτων.

Απάντηση:

s2:

match	action
Ingress port=1, IP_Src=128.120.*, IP_Dst= 128.121.*, Src_Port=Any, Dst_Port=Any, IP_Protocol=TCP	Forward(3)
Ingress port=1, IP_Src=128.120.*, IP_Dst= 128.121.*, Src_Port=Any, Dst_Port=Any, IP_Protocol=UDP	Forward(3)

s1:

match	action
Ingress port=2, IP_Src=128.120.*, IP_Dst= 128.121.*, Src_Port=Any, Dst_Port=Any, IP_Protocol=TCP	Forward(3)

Ingress port=2, IP_Src=128.120.*.*, IP_Dst= 128.121.*.*, Src_Port=Any, Dst_Port=Any, IP_Protocol=UDP	Forward(4)
--	------------

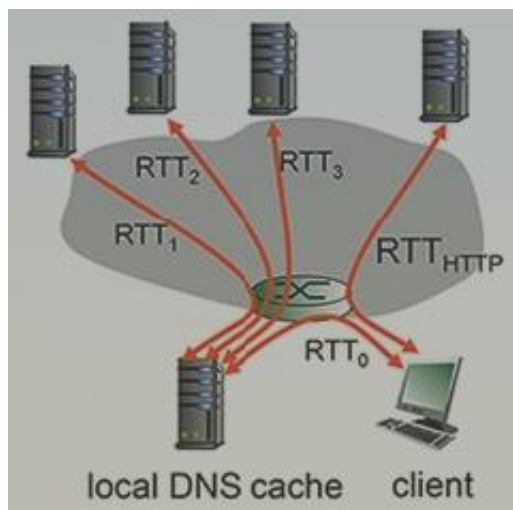
s3:

match	action
Ingress port=3, IP_Src=128.120.*.*, IP_Dst= 128.121.*.*, Src_Port=Any, Dst_Port=Any, IP_Protocol=TCP	Forward(1)
Ingress port=2, IP_Src=128.120.*.*, IP_Dst= 128.121.*.*, Src_Port=Any, Dst_Port=Any, IP_Protocol=UDP	Forward(1)

s4:

match	action
Ingress port=2, IP_Src=128.120.*.*, IP_Dst= 128.121.*.*, Src_Port=Any, Dst_Port=Any, IP_Protocol=UDP	Forward(3)

7. Θεωρήστε ότι κάνετε κλικ σε κάποιο URL καθώς περιηγείστε στο **WWW**. Η IP διεύθυνση του Web Server στον οποίο βρίσκεται το URL αυτό, δεν είναι cached στο μηχανήμά σας, οπότε απαιτείται **DNS** lookup για την εύρεση της IP διεύθυνσης. Έστω ότι χρειάζεται να ερωτηθούν τέσσερις (4) DNS Servers πριν το μηχανήμά σας λάβει από το DNS τη ζητούμενη IP διεύθυνση. Ο πρώτος DNS Server είναι το τοπικό DNS cache και συνεπάγεται μία **RTT** (Round-Trip-Time) καθυστέρηση $RTT_0 = 5$ msec. Ο δεύτερος, ο τρίτος και ο τέταρτος DNS Server, οι οποίοι ερωτήθηκαν, έχουν αντίστοιχα χρόνους RTTs των 31, 42, and 7 msec. Έστω ότι η Web σελίδα που ζητήσαμε πατώντας το URL περιέχει μόνο HTML κείμενο και ότι ο χρόνος RTT μεταξύ του μηχανήματός μας και του Web Server είναι $RTT_{HTTP} = 62$ msec, απαντήστε στα ακόλουθα ερωτήματα:



- i. Θεωρώντας ότι ο χρόνος μετάδοσης του HTML αρχείου είναι αμελητέος, πόσος χρόνος περνάει από τη στιγμή που κάνετε κλικ πάνω στο URL, μέχρι ως ότου ο browser λάβει το αρχείο;
- ii. Υποθέστε τώρα ότι η HTML σελίδα “περιέχει” 9 μικρού μεγέθους αντικείμενα που φιλοξενούνται στον ίδιο server. Θεωρώντας ότι οι χρόνοι μετάδοσης είναι αμελητέοι, πόσος χρόνος περνάει από τη στιγμή που κάνετε κλικ πάνω στο URL της σελίδας, μέχρι ως ότου όλα τα περιεχόμενα της σελίδας να ληφθούν από τον browser, θεωρώντας ότι η έκδοση του HTTP είναι non-persistent και δεν γίνονται παράλληλες TCP συνδέσεις.
- iii. Το ίδιο με το (ii), αλλά ο client είναι ρυθμισμένος ώστε να υποστηρίζει το πολύ 5 παράλληλες TCP συνδέσεις με non-persistent HTTP.
- iv. Το ίδιο με το (ii), αλλά ο client είναι ρυθμισμένος ώστε να υποστηρίζει το πολύ 5 παράλληλες TCP συνδέσεις με persistent HTTP.
- v. Τί παρατηρείτε σχετικά με τη συνολική καθυστέρηση (συγκριτικά) στις παραπάνω περιπτώσεις;

Απάντηση:

- i. Ο συνολικός χρόνος από τη στιγμή της αίτησης της σελίδας μέχρι να την εμφανίσει ο browser είναι:
 $RTT_0 + RTT_1 + RTT_2 + RTT_3 + 2 * RTT_{HTTP} = 5 + 31 + 42 + 7 + 2 * 62 = 209 \text{ msec.}$
 Απαιτούνται 2 RTT_{HTTP} ώστε να γίνει η εγκαθίδρυση της TCP σύνδεσης (1 RTT) και να πραγματοποιηθεί η HTTP GET/response επικοινωνία (1 RTT) πάνω από την TCP σύνδεση.
- ii. Ο συνολικός χρόνος από τη στιγμή της αίτησης της σελίδας μέχρι να την εμφανίσει (με όλα τα αντικείμενα που περιέχει) ο browser είναι:
 $RTT_0 + RTT_1 + RTT_2 + RTT_3 + 2 * RTT_{HTTP} + 2 * 9 * RTT_{HTTP} = 5 + 31 + 42 + 7 + 2 * 62 + 2 * 9 * 62 = 1325 \text{ msec.}$
 Απαιτούνται 2 RTT_{HTTP} (όπως στο (i)) για τη λήψη του βασικού html αρχείου και στη συνέχεια, σειριακά για κάθε ένα από τα 9 ενσωματωμένα αντικείμενα της σελίδας, επίσης καθυστέρηση $2 * RTT_{HTTP}$.
- iii. Ο συνολικός χρόνος από τη στιγμή της αίτησης της σελίδας μέχρι να την εμφανίσει (με όλα τα αντικείμενα που περιέχει) ο browser είναι:
 $RTT_0 + RTT_1 + RTT_2 + RTT_3 + 2 * RTT_{HTTP} + 2 * RTT_{HTTP} + 2 * RTT_{HTTP} = 5 + 31 + 42 + 7 + 2 * 62 + 2 * 62 + 2 * 62 = 457 \text{ msec.}$
 Όμοια καθυστέρηση με το (ii) μέχρι τη λήψη του βασικού html αρχείου. Στη συνέχεια, μπορούμε να έχουμε μέχρι 5 παράλληλες συνδέσεις, λαμβάνοντας ένα ενσωματωμένο αντικείμενο σε κάθε μία. Κάθε παράλληλη σύνδεση απαιτεί δύο RTT_{HTTP} (εγκαθίδρυση της TCP σύνδεσης και πραγματοποίηση HTTP GET/response επικοινωνίας για ένα αντικείμενο πάνω από την TCP σύνδεση). Άρα θα έχουμε λήψη 5 αντικείμενων και 4 αντικείμενων παράλληλα.
- iv. Ο συνολικός χρόνος από τη στιγμή της αίτησης της σελίδας μέχρι να την εμφανίσει (με όλα τα αντικείμενα που περιέχει) ο browser είναι:
 $RTT_0 + RTT_1 + RTT_2 + RTT_3 + 2 * RTT_{HTTP} + RTT_{HTTP} + RTT_{HTTP} = 5 + 31 + 42 + 7 + 2 * 62 + 62 + 62 = 333 \text{ msec.}$
 Όμοια καθυστέρηση με τα (ii) και (iii) μέχρι τη λήψη του βασικού html αρχείου. Όταν έχω persistent HTTP, η TCP σύνδεση παραμένει ενεργή για μελλοντικά HTTP αιτήματα, οπότε δεν έχω άλλη καθυστέρηση για την εγκαθίδρυση TCP σύνδεσης. Επίσης, μπορώ να έχω 5 παράλληλες συνδέσεις,

οπότε χρειαζόμαστε 1 RTT_{HTTP} για τα 5 αντικείμενα της σελίδας και άλλο 1 RTT_{HTTP} για τα υπόλοιπα 4 αντικείμενα, μέσω της ίδιας ενεργής TCP σύνδεσης.

- v. Η συνολική καθυστέρηση είναι μικρότερη στην περίπτωση (iii) που υποστηρίζονται παράλληλες συνδέσεις από ότι στην περίπτωση (ii) που δεν έχω παράλληλες συνδέσεις και ακόμα μικρότερη στην περίπτωση (iv) που η σύνδεση είναι persistent με παράλληλες συνδέσεις.

8. Δίνονται τα **DNS** RR's για κάποιο domain. Απαντήστε στα παρακάτω ερωτήματα:

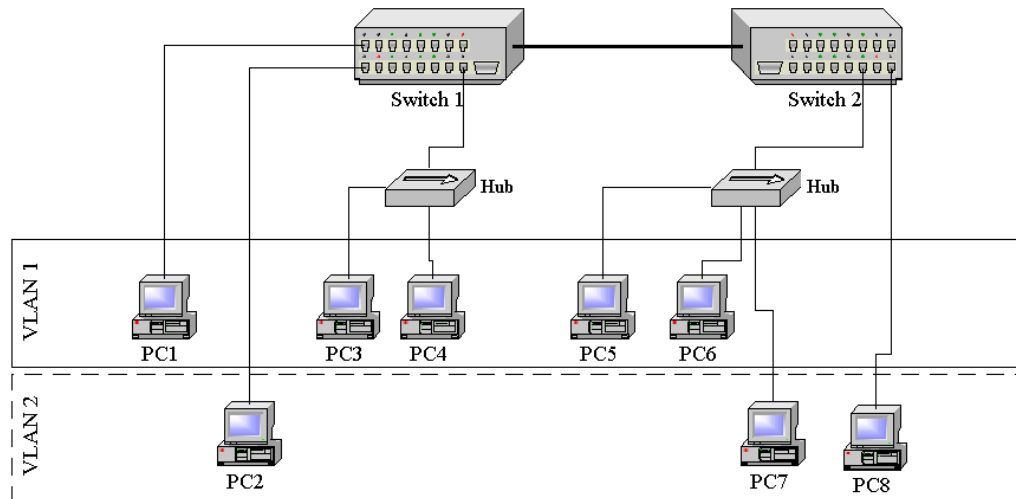
- Σε ποιο domain αναφέρονται;
- Περιγράψτε την χρησιμότητα των δύο πρώτων RR's καθώς και το περιεχόμενό τους.
- Κάποια από τις εγγραφές είναι λάθος. Εντοπίστε την και περιγράψτε το λάθος.

ccslab.aueb.gr.	IN	NS	hermes.aueb.gr.
ccslab.aueb.gr.	IN	MX 10	moria.ccslab.aueb.gr.
silmaril.ccslab.aueb.gr.	IN	A	195.251.252.100
valar.ccslab.aueb.gr.	IN	A	195.251.252.117
www	IN	CNAME	moria.ccslab.aueb.gr
diktya-msc	IN	CNAME	195.251.252.110
lexmark.ccslab.aueb.gr.	IN	A	195.251.252.121
router.ccslab.aueb.gr.	IN	A	195.251.252.126

Απάντηση:

- Domain: ccslab.aueb.gr
- NS: δήλωση Name Server (DNS server) υπεύθυνου για το domain, συντάσσεται με όνομα και όχι IP διεύθυνση.
MX: δήλωση Mail eXchanger (mail server) υπεύθυνου για τους χρήστες του domain, συντάσσεται με όνομα και όχι IP διεύθυνση.
- Λάθος εγγραφή: diktya-msc IN CNAME 195.251.252.110, το CNAME συντάσσεται με όνομα και όχι IP διεύθυνση.

9. Θέλουμε να εγκαταστήσουμε **εικονικά LANs (VLANs)** σύμφωνα με το σχήμα.

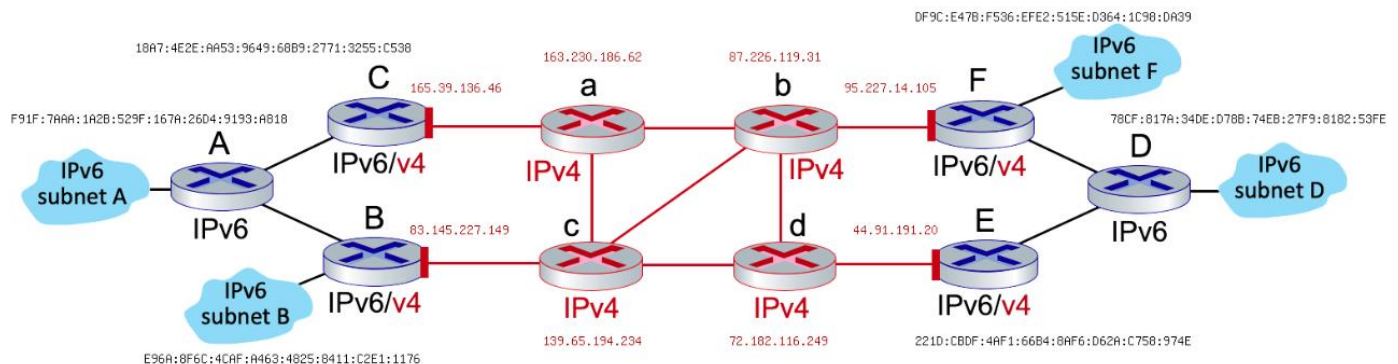


- Αν αγνοήσουμε την επιλογή εκχώρησης ενός «χρώματος» VLAN σε κάθε διεύθυνση επιπέδου 3 (Layer 3-based VLANs), ποια θα ήταν η επιλογή σας μεταξύ της εκχώρησης ενός «χρώματος» VLAN σε κάθε θύρα (port-based) και της εκχώρησης ενός «χρώματος» VLAN σε κάθε διεύθυνση MAC (MAC address-based VLAN) και γιατί;
- Αν το PC7 αντί να συνδέεται στον ομφαλό (hub), συνδεθεί κατευθείαν στο μεταγωγέα 2 (Switch 2), αλλάζει κάτι στην απάντηση (i); Εξηγήστε.

Απάντηση:

- Εφόσον το PC7 συνδέεται σε κοινό hub με τα PC5 και PC6 ενώ θέλουμε να το συνδέσουμε σε διαφορετικό VLAN, δεν γίνεται να χρησιμοποιήσουμε port-based VLANs, θα πρέπει να οδηγηθούμε σε προσδιορισμό των VLANs βάσει MAC διευθύνσεων των κόμβων του VLAN.
- Στην περίπτωση που το PC7 συνδεθεί κατευθείαν στο Switch 2, μπορούμε να επιλέξουμε οποιαδήποτε από τους δύο τρόπους, ο πιο απλός θα ήταν port-based.

10. Θεωρήστε το παρακάτω δίκτυο το οποίο περιλαμβάνει 4 **IPv6** subnets, συνδεδεμένα μέσω δρομολογητών αμιγώς IPv6 (με μπλε χρώμα), δρομολογητών αμιγώς IPv4 (με κόκκινο χρώμα) και υβριδικών δρομολογητών IPv6/IPv4 (μπλε και κόκκινο χρώμα στα IPv4 interfaces).



Έστω ότι ένας host του subnet F θέλει να στείλει ένα IPv6 datagram σε ένα host του subnet B. Θεωρήστε ότι η προώθηση των datagrams μεταξύ των δύο hosts ακολουθεί τη διαδρομή:

F → b → d → c → B

Απαντήστε στις ακόλουθες ερωτήσεις:

- i. Το datagram που προωθείται από τον **F → b** είναι IPv4 ή IPv6 datagram;
- ii. Ποια είναι η διεύθυνση πηγής (source address) και ποια η διεύθυνση προορισμού (destination address) στο datagram από τον **F → b**;
- iii. Το datagram από τον **F → b** ενσωματώνει κάποιο datagram; Αν ναι, ποια είναι η διεύθυνση πηγής και ποια η διεύθυνση προορισμού του ενσωματωμένου datagram;
- iv. Το datagram που προωθείται από τον **b → d** είναι IPv4 ή IPv6 datagram;
- v. Ποια είναι η διεύθυνση πηγής (source address) και ποια η διεύθυνση προορισμού (destination address) στο datagram από τον **b → d**;
- vi. Το datagram από τον **b → d** ενσωματώνει κάποιο datagram; Αν ναι, ποια είναι η διεύθυνση πηγής και ποια η διεύθυνση προορισμού του ενσωματωμένου datagram;
- vii. Το datagram που προωθείται από τον **d → c** είναι IPv4 ή IPv6 datagram;
- viii. Ποια είναι η διεύθυνση πηγής (source address) και ποια η διεύθυνση προορισμού (destination address) στο datagram από τον **d → c**;
- ix. Το datagram από τον **d → c** ενσωματώνει κάποιο datagram; Αν ναι, ποια είναι η διεύθυνση πηγής και ποια η διεύθυνση προορισμού του ενσωματωμένου datagram;
- x. Το datagram που προωθείται από τον **c → B** είναι IPv4 ή IPv6 datagram;
- xi. Ποια είναι η διεύθυνση πηγής (source address) και ποια η διεύθυνση προορισμού (destination address) στο datagram από τον **c → B**;
- xii. Το datagram από τον **c → B** ενσωματώνει κάποιο datagram; Αν ναι, ποια είναι η διεύθυνση πηγής και ποια η διεύθυνση προορισμού του ενσωματωμένου datagram;
- xiii. Ποιος δρομολογητής είναι η “είσοδος στη σήραγγα” (tunnel) και ποιος η “έξοδος στη σήραγγα”;
- xiv. Ποιο πρωτόκολλο ενσωματώνει το άλλο, το IPv4 ή το IPv6;

Απάντηση:

Πριν ξεκινήσει η προώθηση των πακέτων έχει δημιουργηθεί μεταξύ των δρομολογητών που βρίσκονται στα άκρα του IPv4 δικτύου σήραγγα (tunnel) IPv4. Οι δρομολογητές γνωρίζουν τις IPv4 διευθύνσεις που έχουν οι δρομολογητές (gateways) που βρίσκονται στις εισόδους/εξόδους της σήραγγας ώστε να τους χρησιμοποιούν ως αφετηρία/προορισμό των πακέτων για το IPv4 πακέτο που κάνει encapsulate το IPv6 πακέτο.

- i. Το datagram που προωθείται από τον **F → b** είναι IPv4
- ii. Από τον **F → b**, source IP address: 95.227.14.105, destination IP address: 83.145.227.149
- iii. Από τον **F → b**, υπάρχει ενσωματωμένο datagram με
source IP address: DF9C:E47B:F536:EFE2:515E:D364:1C98:DA39 και
destination IP address: E96A:8F6C:4CAF:A463:4825:8411:C2E1:1176
- iv. Το datagram που προωθείται από τον **b → d** είναι IPv4
- v. Από τον **b → d**, source IP address: 95.227.14.105, destination IP address: 83.145.227.149
- vi. Από τον **b → d**, υπάρχει ενσωματωμένο datagram με
source IP address: DF9C:E47B:F536:EFE2:515E:D364:1C98:DA39 και
destination IP address: E96A:8F6C:4CAF:A463:4825:8411:C2E1:1176
- vii. Το datagram που προωθείται από τον **d → c** είναι IPv4
- viii. Από τον **d → c**, source IP address: 95.227.14.105, destination IP address: 83.145.227.149

- ix. Από τον **d** → **c**, υπάρχει ενσωματωμένο datagram με
source IP address: DF9C:E47B:F536:EFE2:515E:D364:1C98:DA39 και
destination IP address: E96A:8F6C:4CAF:A463:4825:8411:C2E1:1176
- x. Το datagram που προωθείται από τον **c** → **B** είναι IPv4
- xi. Από τον **c** → **B**, source IP address: 95.227.14.105, destination IP address: 83.145.227.149
- xii. Από τον **c** → **B**, υπάρχει ενσωματωμένο datagram με
source IP address: DF9C:E47B:F536:EFE2:515E:D364:1C98:DA39 και
destination IP address: E96A:8F6C:4CAF:A463:4825:8411:C2E1:1176
- xiii. Είσοδος σήραγγας: router F, έξοδος σήραγγας: router B
- xiv. Το IPv4 ενσωματώνει το IPv6 προκειμένου να εξασφαλιστεί συμβατότητα μεταξύ της
υπάρχουσας υποδομής IPv4 και του IPv6. Τα IPv6 datagrams ενσωματώνονται μέσα στο payload
του IPv4 datagram. Τα IPv4 datagrams προωθούνται από IPv4 router σε IPv4 router μέχρι να
φτάσουν σε IPv6 router, ο οποίος εξάγει (decapsulates) το IPv6 datagram και το προωθεί σε IPv6
router ή subnet