

Αυτόματη ρύθμιση και επαλήθευση ορθότητας παραμέτρων δικτύων νέας γενιάς

Θεμελιώτης Κωνσταντίνος

Διπλωματική Εργασία

Επιβλέπων: Λιάσκος Χρήστος

Ιωάννινα, Μάιος 2022



**ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA**

Ευχαριστίες

Θέλω να ευχαριστήσω από τα βάθη της καρδιάς μου τους γονείς μου και την οικογένεια μου για την συνεχή στήριξη τους σε μένα σε όλα τα επίπεδα όλα αυτά τα χρόνια. Επίσης να ευχαριστήσω τους φίλους, τους συναδέλφους και τους καθηγητές μου από τους οποίους έμαθα πολλά πράγματα που θα τα κουβαλάω μαζί μου στην μετέπειτα πορεία μου.

1/5/2022

Θεμελιώτης Κωνσταντίνος

Περίληψη

Η όλο και μεγαλύτερη χρήση ηλεκτρονικών συσκευών που έχουν πρόσβαση στο διαδίκτυο – internet και η συνεχής ανάπτυξη εφαρμογών και υπηρεσιών, δημιουργούν όλο και μεγαλύτερα και πιο πολύπλοκα δίκτυα τα οποία όλα μαζί αποτελούν το διαδίκτυο που συνεχώς μεγαλώνει. Αυτή η συνεχής κλιμάκωση και εξάρτηση της καθημερινότητας από το διαδίκτυο απαιτεί από αυτά να παρέχουν υψηλή ποιότητα υπηρεσιών, αξιοπιστία και ασφάλεια. Όσο συμβαίνει όμως αυτό, τόσο η πολυπλοκότητα, η δυσκολία διαχείρισης του και τα σφάλματα που μπορούν να προκύψουν από ανθρώπινο και-μη παράγοντα πληθαίνουν. Για την αποτελεσματικότερη διαχείριση των δικτύων, την ελαχιστοποίηση των πιθανών σφαλμάτων και τον άμεσο εντοπισμό τους έχουν αναπτυχθεί διαδικασίες αυτοματισμών και επαλήθευσης.

Σκοπός αυτής της εργασίας είναι να παρουσιάσει το πεδίο των αυτοματισμών δικτύων ή αλλιώς network automation και να εμβαθύνει στο πεδίο της επαλήθευσης δικτύων ή αλλιώς network validation ή network testing. Θα ξεκινήσουμε βλέποντας τη σημαντικότητα των δικτύων στη σημερινή κοινωνία και την ανάγκη που αυτή δημιουργεί για την ύπαρξη αξιόπιστων δικτύων. Θα παρουσιάσουμε το τι είναι το network automation και το network validation και ποιες είναι οι διαδικασίες του, στη συνέχεια θα παρουσιάσουμε εργαλεία που χρησιμοποιούνται σήμερα στη δικτύωση των υπολογιστών για automation και validation και τέλος θα παρουσιάσουμε το εργαλείο που αναπτύξαμε εμείς για network validation.

Λέξεις Κλειδιά: Internet, Network Automation, Network Validation, Network testing

Abstract

The increasing use of electronic devices that have access to the internet and the continuous app and service development on them, are constantly creating larger and more complex networks which all together are part of the greater Internet. This constant escalation and dependence of everyday life on the internet, requires from networks high quality of service, reliability, and security. But as long networks become bigger and bigger, they get more complex their management becomes harder, and the possibility of an error that can arise from human or non-human factors are increasing. To tackle this problem, automation and validation techniques have been developed.

The aim of this work is to present the field of network automation and to deepen in the field of network validation or network testing. We will start by seeing the importance of networks today and the need it creates for the existence of reliable networks, next, we will present what network automation and network validation are and what its processes are, then we will present tools used today in networking by network admins and engineers and finally, we will present our tool, developed for network validation.

Keywords: Network Automation, Network validation, Network testing

Περιεχόμενα

Κεφάλαιο 1. Εισαγωγή	8
1.1 Το χθες, το σήμερα και το αύριο των δικτύων.....	8
1.2 Σφάλματα Δικτύων.....	9
1.2.1 Λόγοι σφαλμάτων δικτύου.....	10
1.2.2 Επιπτώσεις των σφαλμάτων δικτύου στον πραγματικό κόσμο.	12
1.3 Με τι θα ασχοληθούμε	14
Κεφάλαιο 2. Αυτοματισμοί και επαλήθευση δικτύου.....	15
2.1 Εισαγωγή	15
2.2 Αυτοματισμοί δικτύων	15
2.3 Επαλήθευση δικτύου – network validation.....	17
2.3.1 Τι είναι η επαλήθευση δικτύου – network validation.....	18
2.3.2 Διαδικασίες του Network Validation.....	19
2.4 Network Validation Pipeline.....	24
2.4.1 Validation pipelines	24
Κεφάλαιο 3. Εργαλεία προσομοίωσης και επαλήθευσης δικτύων.	27
3.1 Εισαγωγή	27
3.2 Εργαλεία αυτοματισμών και επαλήθευσης	28
3.2.1 Ansible.....	28
3.2.2 GNS3.....	30
3.2.3 Batfish	31
3.2.4 Suzieq.....	33
3.2.5 EVE-NG.....	34
3.2.6 Cisco packet tracer.....	36
3.2.7 pyATS.....	38
Κεφάλαιο 4. Το δικό μας εργαλείο προσομοίωσης δικτύου	41
4.1 Εισαγωγή	41
4.2 Δυνατότητες του εργαλείου.....	42
4.3 Χαρακτηριστικά και λογική του εργαλείου	45
4.3.1 Χαρακτηριστικά του εργαλείου.....	45

4.3.2 Λογική του εργαλείου	47
4.4 Εργαλεία που χρησιμοποιήθηκαν	60
4.5 Αρχιτεκτονική του λογισμικού	61
4.6 Επεκτασιμότητα.....	64
Κεφάλαιο 5. Παραδείγματα εκτέλεσης.....	66
5.1 Εισαγωγή	66
5.2 Τοπολογία 0.....	68
5.2.1 Οπτικοποίηση τοπολογίας.....	68
5.2.2 Περιγραφή τοπολογίας.....	68
5.2.3 Εκτελέσεις.....	69
5.3 Τοπολογία 1.....	77
5.3.1 Οπτικοποίηση τοπολογίας.....	77
5.3.2 Περιγραφή Τοπολογίας.....	78
5.3.3 Εκτελέσεις.....	79
5.4 Τοπολογία 2.....	87
5.4.1 Οπτικοποίηση τοπολογίας.....	87
5.4.2 Περιγραφή Τοπολογίας.....	88
5.4.3 Εκτελέσεις.....	89
5.5 Τοπολογία 3.....	97
5.5.1 Οπτικοποίηση τοπολογίας.....	97
5.5.2 Περιγραφή τοπολογίας.....	97
5.5.3 Εκτελέσεις.....	99
5.6 Τοπολογία 4.....	106
5.6.1 Οπτικοποίηση τοπολογίας.....	106
5.6.2 Περιγραφή τοπολογίας.....	107
5.6.3 Εκτελέσεις.....	108
Κεφάλαιο 6. Επίλογος.....	126
Κεφάλαιο 7. Βιβλιογραφία.....	127

Κεφάλαιο 1. Εισαγωγή

1.1 Το χθες, το σήμερα και το αύριο των δικτύων

Το χθες και το σήμερα των δικτύων

Από τους πρώτους ηλεκτρονικούς υπολογιστές και τα πρώτα δίκτυα υπολογιστών, μετέπειτα στη πρώτη μορφή του επίσημου Internet και καταλήγοντας σήμερα στις σύγχρονες έξυπνες συσκευές και το σύγχρονο διαδίκτυο, οι τεχνολογικές εξελίξεις είναι τρομακτικά γιγαντιαίου μεγέθους. Παράλληλα μαζί με αυτές τις τεχνολογικές εξελίξεις έχει εξελιχθεί και η καθημερινότητα των ανθρώπων με εφάμιλλους ρυθμούς, όπου πηγαίνοντας πίσω μόνο μία δεκαετία μπορούμε να δούμε ότι η καθημερινότητα ήταν αρκετά διαφορετική. Σήμερα, σχεδόν κάθε πτυχή της καθημερινότητας των ανθρώπων, των επιχειρήσεων των οργανισμών της οικονομίας και των κρατών, βασίζεται σε μεγάλο βαθμό ή και εξαρτώνται αποκλειστικά από ηλεκτρονικές συσκευές και το διαδίκτυο. Η τεχνολογία έχει εισβάλλει στην καθημερινότητα μας, στην επικοινωνία, στην ψυχαγωγία και τον τρόπο που αυτή καταναλώνεται, στην εργασία, στο εμπόριο, στις συναλλαγές, στη γραφειοκρατία και πολλά ακόμη. Ιδιαίτερα έντονο φάνηκε το φαινόμενο της εξάρτησης από τη τεχνολογία σε παγκόσμια κλίμακα, όταν λόγω της πανδημίας τον Μάρτιο του 2020, εν μία νυκτί η εκπαίδευση σε όλες τις κλίμακες της, η εργασία σε μεγάλο της βαθμό, οι υπηρεσίες ολόκληρων κρατών αλλά και η κοινωνική ζωή, μεταφέρθηκαν σχεδόν εξ ολοκλήρου στο διαδίκτυο.

Το αύριο των δικτύων

Όλα αυτά που αναφέραμε παραπάνω αποτελούν πλέον το παρόν της τεχνολογίας και των δικτύων. Το αύριο, με τους ρυθμούς ανάπτυξης που υπάρχουν, έρχεται πολύ γρήγορα και με τις νέες τεχνολογίες δικτύων που ήδη εμφανίζονται και

γίνονται διαθέσιμες στην αγορά, θα φέρουν νέες δυνατότητες όπου με την σειρά τους θα φέρουν σημαντικές αλλαγές στην καθημερινότητα μας. Τεχνολογίες όπως το ασύρματα δίκτυα πέμπτης γενιάς, 5G σήμερα και το 6G αύριο θα φέρουν high bandwidth – low latency αξιόπιστο διαδίκτυο στις κοινότητες το οποίο με την σειρά του θα φέρει πολλές δυνατότητες δημιουργίας και εκμετάλλευσης. Ιδέες όπως η τηλεϊατρική και τα εξ αποστάσεως χειρουργεία, από concept το οποίο κάνει κάποια πρώτα βήματα θα μπορεί να αποτελεί μία καθημερινότητα που θα σώζει ζωές από την μία άκρη του πλανήτη στην άλλη. Η πλήρως αυτόνομη οδήγηση οχημάτων, όπως αυτοκίνητα και φορτηγά, όπου σήμερα βασίζεται σε υπολογιστική όραση και αισθητήρες του οχήματος, με την βοήθεια των δικτύων 5G, θα μπορούσε να δημιουργήσει ένα δίκτυο συνεργατικών οχημάτων τα οποία επικοινωνούν μεταξύ τους και με κεντρικούς κόμβους. Και αυτά χάρις στο υψηλό bandwidth αλλά και σημαντικά χαμηλό latency που θα προσφέρουν αυτές οι τεχνολογίες. Επίσης τεχνολογίες όπως ή πρόσβαση στο διαδίκτυο με την χρήση δορυφόρων, θα μπορεί να δίνει πρόσβαση με υψηλό bandwidth και χαμηλό latency στις πιο απομακρυσμένες περιοχές του πλανήτη. Αυτές είναι μόνο κάποιες από τις μελλοντικές εφαρμογές που μπορεί να προσφέρει η εξέλιξη της τεχνολογίας και των δικτύων.

Βλέπουμε ότι η εξέλιξη της τεχνολογίας έχει φέρει και θα συνεχίσει να φέρνει την ανάγκη ύπαρξης μεγαλύτερων, καλύτερων, ταχύτερων και πιο αξιόπιστων δικτύων αλλά και η εξέλιξη των δικτύων έχει ανοίξει χώρο για την δημιουργία νέων εφαρμογών και τεχνολογιών που θα αλλάξουν την καθημερινότητα μας προς το καλύτερο.

1.2 Σφάλματα Δικτύων

Βλέπουμε ότι το διαδίκτυο αποτελεί πλέον τη ραχοκοκαλιά της σύγχρονης κοινωνίας. Τα δίκτυα και γενικά το διαδίκτυο είναι ένα τεράστιο πολύπλοκο ανθρώπινο κατασκευάσμα που αποτελείται από επιμέρους διαφορετικές τεχνολογίες που η κατάσταση του μεταβάλλεται συνεχώς. Αυτή η σημαντικότητα και η φύση των δικτύων απαιτεί από τους Internet Service Providers (ISP), cloud providers και άλλους οργανισμούς και εταιρείες, να παρέχουν υψηλή ποιότητα υπηρεσιών αλλά και υψηλή ποιότητα ασφάλειας. Τα δίκτυα πρέπει είναι ασφαλή, αξιόπιστα και ανθεκτικά και ευέλικτα ώστε να μπορούν να ανταποκριθούν σε οποιοδήποτε σενάριο μπορεί να προκύψει. Βλάβες όμως γινόντουσαν, γίνονται και είναι βέβαιο ότι θα συνεχίσουν να

γίνονται και στο μέλλον. Από μία μικρή βλάβη σε μία σύζευξη μέσα στο δίκτυο ή μια μεγάλη βλάβη, μπορεί να προκληθεί από πτώση ποιότητας των υπηρεσιών και δυσλειτουργία έως και πλήρη downtime-blackout του δικτύου για μεγάλο χρονικό διάστημα. [1]

1.2.1 Λόγοι σφαλμάτων δικτύου

Οι λόγοι που μπορεί να προκληθεί βλάβη στο δίκτυο που θα οδηγήσει σε υπολειτουργία ή μία πτώση δικτύου – downtime είναι πολλοί και πολλές φορές επικαλυπτόμενοι [2]. Οι πιο σημαντικοί από αυτούς είναι οι εξής :

Ανθρώπινο σφάλμα:

Ίσως ο συνηθέστερος λόγος όπου μπορεί να προκληθεί βλάβη – πτώση σε ένα δίκτυο είναι ο ανθρώπινος παράγοντας. Ένα τέτοιο σφάλμα μπορεί να οφείλεται στην μη επαρκή εκπαίδευση των διαχειριστών του δικτύου, στη βιασύνη ή αφηρημάδα ενός τεχνικού, στην έλλειψη χρόνου, σε μη σαφή κατανόηση της αλλαγής και γενικότερα του δικτύου. Ειδικά λόγου του μεγέθους ενός δικτύου που ένας τεχνικός έχει να διαχειριστεί, τέτοια σφάλματα, όσο πιο μεγάλο και πολύπλοκο είναι ένα δίκτυο, τόσο πιο πιθανό να γίνουν. Επίσης, μία συνήθης περίπτωση, είναι οι επιθέσεις malware μέσω phishing emails, όπου ως στόχο έχουν τους τεχνικούς – προσωπικό ενός οργανισμού, μιας εταιρείας.

Misconfigurations –Σφάλματα στην παραμετροποίηση

Τα σφάλματα στην παραμετροποίηση των συσκευών ενός δικτύου, είναι υπεύθυνα για πολύ μεγάλο ποσοστό των αναπάντεχων βλαβών. Συνήθως τέτοια σφάλματα παρατηρούνται στους δρομολογητές των δικτύων όπου και ένα μικρό σφάλμα μπορεί να προκαλέσει βλάβη σε πολλές υπηρεσίες που βρίσκονται στο διαδίκτυο ταυτόχρονα. Αυτά τα σφάλματα προκύπτουν πολλές φορές από ανθρώπινο παράγοντα, όχι όμως μόνο. Θα επικεντρωθούμε σε αυτά και στις διαδικασίες εντοπισμού τους παρακάτω.

Σφάλματα στο Hardware

Μια ελαττωματική συσκευή, ένας ελαττωματικός σύνδεσμος ή μία παλιά συσκευή είναι πιθανό να υπολειτουργούν ή να καταρρεύσουν και να προκαλέσουν σοβαρές βλάβες σε ένα δίκτυο. Επίσης μια παλιά συσκευή, είναι πιθανόν να προκαλέσει ζητήματα ασυμβατότητας με το πιο πρόσφατο λογισμικό κάτι το οποίο δημιουργεί κι αυτό προβλήματα.

Σφάλματα στο Software

Λογισμικό σε έναν server το οποίο περιέχει bugs, παρωχημένο λογισμικό με κενά ασφαλείας ή και μη συμβατό παρωχημένο firmware σε συσκευές είναι σοβαρές αιτίες ώστε να υπάρχουν πτώσεις του δικτύου αλλά και κενά ασφαλείας τα οποία αφήνουν απροστάτευτο το δίκτυο σε επιθέσεις.

Κυβερνοεπιθέσεις

Κακόβουλες κυβερνοεπιθέσεις γίνονται καθημερινά σε μεγάλους οργανισμούς και εταιρείες όπου συνήθως προκαλούν την υπολειτουργία ενός δικτύου ή και την κατάρρευση του. Κενά ασφαλείας όμως σε ένα δίκτυο, πέρα από την υπολειτουργία ενός δικτύου, μπορεί να οδηγήσουν σε μην εξουσιοδοτημένη πρόσβαση σε δεδομένα ενός οργανισμού, αλλά και των πελατών της

Βλάβες παροχής ηλεκτρικής ενέργειας

Τα δίκτυα και οι υποδομές τους καταναλώνουν μεγάλες ποσότητες ηλεκτρικής ενέργειας για να λειτουργήσουν. Από την μία οι πολλές συσκευές που δουλεύουν 24 ώρες το 24ωρο αλλά και η ψύξη που χρειάζονται αυτές οι συσκευές, φέρνουν την κατανάλωση ρεύματος σε υψηλά επίπεδα. Το δίκτυο παροχής ηλεκτρικού ρεύματος όμως, ειδικά σε ακραίες καιρικές συνθήκες και σε περιόδους υψηλής ζήτησης ενέργειας, είναι δύσκολο να ανταποκριθεί με αποτέλεσμα να συμβαίνουν πολλές διακοπές στην παροχής. Τέτοιες πτώσεις μπορεί να οδηγήσουν στην πτώση ενός δικτύου, αλλά και στην μόνιμη βλάβη συσκευών του.

Έλλειψη προσωπικού

Το μέγεθος και η πολυπλοκότητα των δικτύων απαιτεί ανάλογο μέγεθος σε προσωπικό το οποίο πρέπει να είναι και κατάλληλα εκπαιδευμένο. Πολλές φορές όμως λόγω

δυσκολίας εύρεσης τέτοιου προσωπικού στην αγορά εργασίας ή για λόγους περικοπών της εταιρείας, το προσωπικό δεν είναι αρκετό να διαχειριστεί και να επιβλέπει το δίκτυο με αποτέλεσμα να μεγαλώνει η πιθανότητα σφάλματος από ανθρώπινο παράγοντα, να καθυστερούν σημαντικά οι εργασίες συντήρησης αλλά και η διαδικασία ανάκαμψη μετά από μία βλάβη να γίνεται ακόμα δυσκολότερη.

1.2.2 Επιπτώσεις των σφαλμάτων δικτύου στον πραγματικό κόσμο.

Η πρόκληση μίας βλάβης από τις παραπάνω σε ένα δίκτυο με αποτέλεσμα το downtime -δηλαδή πόση ώρα το δίκτυο υπολειτουργεί ή δεν λειτουργεί καθόλου-, δεν γίνεται να μην έχει επιπτώσεις στον πραγματικό κόσμο. Οι επιπτώσεις αυτές μεταφράζονται αλλιώς για τις επιχειρήσεις και αλλιώς για τους πελάτες – χρήστες.

Για έναν χρήστη το downtime σημαίνει αδυναμία πρόσβασης στις υπηρεσίες όπου χρησιμοποιεί, όπου αυτές μπορεί να είναι υπηρεσίες επικοινωνίας και ψυχαγωγίας αλλά και αδυναμία εργασίας. Αυτές οι βλάβες μπορεί να έχουν σημαντικές επιπτώσεις για ένα χρήστη.

Για μια επιχείρηση που δραστηριοποιείται στο διαδίκτυο ή μία επιχείρηση που βασίζει μεγάλο μέρος της λειτουργία της στο διαδίκτυο, το downtime σημαίνει μη αποτελεσματική εργασία έως πλήρης αδυναμία εργασίας και αδυναμία εξυπηρέτησης. Αυτό μεταφράζεται σε πτώση της παραγωγικότητας μιας επιχείρησης, σε οικονομικές απώλειες και σε ζημία στην εικόνα της επιχείρησης.

Για έναν ISP ή μεγάλο οργανισμό η πτώση των υπηρεσιών τους σημαίνει και πτώση των υπηρεσιών των πελατών τους. Αυτό μεταφράζεται σε βλάβη στην δημόσια εικόνα της επιχείρησης, απώλεια δυσαρεστημένων πελατών, μεγάλες οικονομικές απώλειες, υποτίμηση της αξίας της εταιρείας στο χρηματιστήριο.

1.2.2.1 Παραδείγματα downtime στον πραγματικό κόσμο

Οι περιπτώσεις downtime στον πραγματικό κόσμο είναι αρκετά υπαρκτές και ακόμα και μικρές βλάβες σε εργασίες ρουτίνας, μπορεί να ταλαιπωρήσουν πολύ κόσμο και να προκαλέσουν ζημίες μεγάλης αξίας. Παρακάτω παρουσιάζονται μερικές περιπτώσεις downtime στον πραγματικό κόσμο.

Facebook outage, Οκτώβριος 2021

Μία τέτοια περίπτωση downtime μεγάλης έκτασης έχουμε συναντήσει στην περίπτωση του facebook [3]. Στις 4 Οκτωβρίου του 2021, το δίκτυο της εταιρείας facebook έπεσε, με αποτέλεσμα όλες οι υπηρεσίες της εταιρείας, facebook, messenger, instagram, whatsapp, Mapillary και Oculus να μην είναι προσβάσιμες. Η πτώση αυτή επηρέασε τις υπηρεσίες και τους χρήστες παγκοσμίως για ένα διάστημα περίπου 6 με 7 ωρών. Πέρα από τις ίδιες τις υπηρεσίες, δεν ήταν δυνατή η πρόσβαση σε third-party υπηρεσίες όπου χρησιμοποιούταν ο λογαριασμός του facebook για πρόσβαση. Επίσης, δυνατή δεν ήταν η χρήση εσωτερικών εργαλείων της εταιρείας που χρησιμοποιούνται για την επίλυση τέτοιων προβλημάτων και για την επικοινωνία μεταξύ των εργαζομένων αλλά και προβλήματα υπήρξαν και στην είσοδο στις υποδομές της εταιρείας λόγω του συστήματος ασφαλείας εισόδου στο κτίριο. Αυτός ήταν και ο σημαντικότερος λόγος που η βλάβη κράτησε τόσες ώρες. Η βλάβη συνέβη κατά τη διάρκεια μίας προγραμματισμένης συντήρησης ρουτίνας, όπου μία λάθος παραμετροποίηση στους δρομολογητές του δικτύου του facebook, οδήγησε στο να μην μπορούν να επικοινωνήσουν τα datacenters του facebook με το υπόλοιπο διαδίκτυο. Συγκεκριμένα αυτό το misconfiguration έγινε στο πρωτόκολλο BGP, όπου αυτό σταμάτησε να ανακοινώνει τα DNS prefix routes στους DNS resolvers του υπόλοιπου διαδικτύου.

Από αυτή τη βλάβη υπήρχαν σημαντικές επιπτώσεις για την εταιρεία του facebook αλλά και για τους πελάτες του. Άνθρωποι σε όλον τον πλανήτη που βασίζουν την επικοινωνία τους σε αυτές τις πλατφόρμες, δεν ήταν δυνατόν να επικοινωνήσουν. Επιχειρήσεις που βασίζουν την λειτουργία τους σε αυτές τις πλατφόρμες, δεν ήταν δυνατόν να δουλέψουν. Για την εταιρεία του facebook, αυτή η βλάβη μεταφράστηκε σε υποτίμηση των μετοχών τις κατά 5% στο χρηματιστήριο και σύμφωνα με το περιοδικό Fortune [4] σε απώλεια περίπου 60 εκατομμυρίων δολαρίων από διαφημιστικά έσοδα.

British airways, Μάιος 2017

Το Μάιο του 2017, η εταιρεία συνεργάτης της αεροπορικής εταιρείας για τη διαχείριση των συστημάτων IT, από ανθρώπινο σφάλμα, έκλεισε την παροχή ηλεκτρικής ενέργειας προς datacenter της εταιρείας. Αυτή η πτώση ρεύματος οδήγησε σε φυσικές ζημιές σε συσκευές του datacenter. Αποτέλεσμα αυτής της βλάβης ήταν η ταλαιπωρία περίπου 75000 επιβατών με εκατοντάδες πτήσεις να ακυρώνονται και να καθυστερούν. Για την εταιρεία αυτό μεταφράστηκε σε μεγάλη ζημία για την δημόσια εικόνα της και πολλά εκατομμύρια σε αποζημιώσεις πελατών [5].

Cloudflare outage, Ιούλιος 2020

Στις 17 Ιουλίου του 2020 στο δίκτυο του cloudflare, από το οποίο υπολογίζεται ότι διέρχεται το 18% όλης της κίνησης του διαδικτύου σημειώθηκε outage διάρκειας 27 λεπτών. Το λάθος σημειώθηκε στο backbone του δικτύου όπου έγινε αλλαγή των configurations σε έναν από τους δρομολογητές. Το σφάλμα αυτό, κατεύθυνε όλη την κίνηση του backbone μέσω ενός άλλου δρομολογητή ο οποίος αμέσως είχε να αντιμετωπίσει παραπάνω φορτίο από όσο μπορούσε. Το λάθος αυτό έριξε την κίνηση σε όλο το δίκτυο κατά 50%. Παρόλα αυτά λόγω την αρχιτεκτονικής του backbone δικτύου το σφάλμα δεν επηρέασε όλο το δίκτυο της εταιρείας αλλά συγκεκριμένες γεωγραφικές περιοχές [6].

1.3 Με τι θα ασχοληθούμε

Τα δίκτυα και το διαδίκτυο όπως αναφέραμε παραπάνω, είναι ένα τεράστιο πολύπλοκο ανθρώπινο κατασκεύασμα που αποτελείται από διαφορετικές τεχνολογίες, είναι δυναμικό, η κατάσταση του μεταβάλλεται συνέχεια και είναι εξαιρετικά δύσκολο σε κάθε περίπτωση να δουλεύουν συνέχεια όλα σωστά. Για αυτό και κατά την διάρκεια ζωής ενός δικτύου επιβάλλεται ο προσεκτικός σχεδιασμός, η προσεκτική διαχείριση και η συνεχής παρακολούθηση του και ακόμα πολλά, ώστε να προσφέρει όλο και περισσότερο καλύτερη ποιότητα υπηρεσιών και λιγότερο downtime.

Στην παρακάτω εργασία θα ασχοληθούμε με την διαδικασία αυτοματοποίησης στα δίκτυα. Όπως αναφέρει και ο τίτλος «Αυτόματη ρύθμιση και επαλήθευση ορθότητας παραμέτρων δικτύων νέας γενιάς» θα αναφερθούμε και θα μελετήσουμε την αυτόματη ρύθμιση των παραμέτρων δικτύων και την ανάγκη ύπαρξης της, για το οποίο επικρατεί ο όρος στην αγορά “Network Automation” και θα επικεντρωθούμε στην επαλήθευση ορθότητας παραμέτρων δικτύων νέας γενιάς που ακούει στην αγορά στο όνομα “Network Validation”. Θα δούμε την ανάγκη ύπαρξης του Network Validation, κάποιες βασικές διαδικασίες και όρους, στην συνέχεια θα δούμε εργαλεία, ανοιχτού κώδικα και όχι, που χρησιμοποιούνται αυτή τη στιγμή στην αγορά και τελειώνοντας θα παρουσιάσουμε την δική μας πρώιμη εκδοχή ενός εργαλείου επαλήθευσης παραμέτρων - Network Validation.

Κεφάλαιο 2. Αυτοματισμοί και επαλήθευση δικτύου.

2.1 Εισαγωγή

Κατά την αρχική περίοδο των δικτύων η διαχείριση τους ήταν μία διαδικασία που ήταν αρκετά διαφορετική από ότι είναι σήμερα, όπου οι ελάχιστες διαδικασίες που γινόντουσαν, γινόντουσαν χειροκίνητα από τους διαχειριστές του δικτύου. Η μεγάλη κλιμάκωση των δικτύων και του διαδικτύου, έφερε την ανάγκη για πιο σχολαστική και συστηματική διαχείριση του και των συσκευών που το απαρτίζουν. Πλέον ένα δίκτυο απαιτεί συνεχή παρακολούθηση της κατάστασης του, της κίνησης του όλο το 24ωρο, της κατάστασης ασφάλειας του της απόδοσης του και προσεκτική παραμετροποίηση. Η διαχείριση όλων αυτών των απαιτήσεων, του μεγέθους και της πολυπλοκότητας των δικτύων σήμερα, βασίζονται σε μεγάλο βαθμό σε αυτοματισμούς δικτύων - Network automation

2.2 Αυτοματισμοί δικτύων

Τι ονομάζουμε αυτοματισμούς δικτύων - network automation

Ως αυτοματισμούς δικτύων - network automation, ορίζεται η χρήση λογισμικού και διαδικασιών για διαχείριση, provisioning, συντήρηση, παραμετροποίηση, testing και monitoring των φυσικών και εικονικών συσκευών (VM) μέσα σε ένα δίκτυο και το Στόχος του network automation είναι η αυτοματοποίηση όλων των παραπάνω διεργασιών η ασφαλέστερη, ευκολότερη και εξ αποστάσεως διαχείριση τους που

άλλοτε γινόντουσαν χειροκίνητα από τους διαχειριστές . Σημαντικοί λόγοι που οδηγούν στην ανάγκη ύπαρξης αυτόματων διεργασιών είναι η μεγάλη κλίμακα των σημερινών δικτύων στα οποία συνεχώς ενσωματώνονται νέες συσκευές και εφαρμογές, το πλήθος και η ετερογένεια των συσκευών που οι διαχειριστές έχουν να αντιμετωπίσουν και το πλήθος των σφαλμάτων που μπορούν να προκύψουν και προκύπτουν από ανθρώπινο παράγοντα [7] [8].

Περιπτώσεις χρήσης network automation

Οι διαδικασίες αυτοματισμών δικτύων σήμερα μπορούν να ενσωματωθούν σε ένα εύρος διεργασιών στον κύκλο ζωής ενός δικτύου, κάποια από τα οποία είναι:

- Σχεδιασμός και προγραμματισμός δικτύων, σχεδιασμός σεναρίων και διαχείριση των συσκευών.
- Έλεγχος συσκευών και επαλήθευση των configurations
- Provisioning φυσικών συσκευών, υπηρεσιών και software που υπάρχει στο δίκτυο
- Συλλογή δεδομένων δικτύου από τις συσκευές, συστήματα, software αλλά και για την κίνηση του δικτύου και τις υπηρεσίες σε πραγματικό χρόνο
- Ανάλυση των δεδομένων χρησιμοποιώντας AI και Machine Learning με σκοπό την καλύτερη κατανόηση της τρέχουσας αλλά και των πιθανών μελλοντικών καταστάσεων του δικτύου.
- Αναβάθμιση λογισμικού στις συσκευές του δικτύου.
- Αποσφαλμάτωση και αποκατάσταση εσωτερικών βλαβών δικτύων.
- Συνεχής παρακολούθηση του δικτύου με σκοπό την επιβεβαίωση ότι παρέχει ποιότητα υπηρεσιών.

Πλεονεκτήματα χρήσης εργαλείων αυτοματισμού δικτύων

Τα πλεονεκτήματα ενός δικτύου όπου εφαρμόζονται εργαλεία αυτοματοποίησης είναι πολλαπλά. Παρ'όλο που απαιτούν από τους μηχανικούς και διαχειριστές του δικτύου περισσότερες τεχνικές γνώσεις και ειδίκευση η εξοικείωση με αυτές τις τεχνολογίες, είναι λύση πολλών προβλημάτων για τους επαγγελματίες.

- **Μείωση πιθανότητας σφαλμάτων:** Όπως αναφέραμε στο 1^ο κεφάλαιο, η πιθανότητα να προκληθεί βλάβη σε ένα δίκτυο και δε από ανθρώπινο λάθος είναι πολύ υψηλή. Έτσι εργασίες όπως αυτοματοποίηση των ενημερώσεων των συσκευών και η παραμετροποίηση μπορούν να γίνουν ασφαλέστερα.

- **Γρηγορότερες αλλαγές και ενημερώσεις:** Με αυτοματοποιημένα εργαλεία ενημερώσεων και monitoring, οι διαχειριστές έχουν μια πιο πλήρη εικόνα του δικτύου κάθε στιγμή και μπορούν εύκολα να ενημερώνονται με σφάλματα που συμβαίνουν σε πραγματικό χρόνο.
- **Βελτιστοποιημένη απόδοση:** Το δίκτυο έχοντας δυναμικό χαρακτήρα, με εργαλεία αυτοματοποίησης, μπορεί να ελέγξει και να δρομολογήσει την κίνηση του κατάλληλα για κάθε πιθανό σενάριο και να επιλύσουν προβλήματα τα οποία σε άλλη περίπτωση μπορεί να είχαν προκληθεί
- **Καλύτερη γνώση και απλοποιημένη διαχείριση του δικτύου:** Με εργαλεία αυτοματισμού, όλες οι διεργασίες διαχείρισης ενός δικτύου γίνονται ταχύτερα, ασφαλέστερα και αποτρέπονται πολλά περιστατικά σφαλμάτων, τα οποία θα σε άλλη περίπτωση θα γινόντουσαν. Οι διαχειριστές και το ίδιο το δίκτυο, γνωρίζει καλύτερα τον εαυτό του και σφάλματα που προκύπτουν εντοπίζονται εύκολα και γρήγορα και μπορούν να επιλυθούν και εξ αποστάσεων.

2.3 Επαλήθευση δικτύου – network validation.

Όπως είδαμε παραπάνω, η διαχείριση ενός μεγάλου δικτύου στο σήμερα όχι μόνο είναι ευκολότερο, αλλά και η πολυπλοκότητα του επιβάλλει την ύπαρξη αυτοματισμών για την διαχείριση του. Ένα από τα κύρια πλεονεκτήματα είναι η μείωση της πιθανότητας σφαλμάτων στο δίκτυο από ανθρώπινο παράγοντα. Παρόλα αυτά η αυτοματοποίηση από μόνη της δεν είναι πανάκεια. Σφάλματα ακόμα μπορεί να προκύψουν. Αυτά μπορεί να βρίσκονται στην ίδια την λογική όπου βασίζονται οι μηχανισμοί του εργαλείου καθώς η διαχείριση πολλαπλών ετερογενών συσκευών παραμένει δύσκολο. Μπορεί όμως και να βρίσκονται πάλι στον ανθρώπινο παράγοντα, καθώς κάποιες φορές είναι απαραίτητη η παρέμβαση του ανθρώπου σε κάποιες από τις αυτοματοποιημένες διαδικασίες.

Η σημαντικότητα όμως ενός «αυτοματοποιημένου» λάθους είναι πιθανό να έχει πολύ μεγαλύτερη έκταση και επιπτώσεις από ένα «χειροκίνητο» λάθος. Αυτό βασίζεται στο γεγονός ότι για παράδειγμα ένα λάθος στην λογική του μηχανισμού του αυτοματισμού, δεν είναι τοπικό -δεν έχει εφαρμοστεί τοπικά σε μια συσκευή-, αλλά έχει, μεγάλη έκταση και συνεπώς ένα μικρό λάθος μπορεί άμεσα να έχει μεγάλες επιπτώσεις. Επιπρόσθετα, κάποιες διαδικασίες συντήρησης του δικτύου, παραμένουν ακόμα στα χέρια των μηχανικών-διαχειριστών δικτύου, εισάγοντας άλλη μια φορά τον ανθρώπινο παράγοντα

Από αυτήν την αδυναμία ύπαρξης τέλειων αυτοματισμών την ύπαρξη πιθανότητας σφάλματος και την ανάγκη για όσο πιο δυνατόν ασφαλέστερα και αξιόπιστα δίκτυα, προκύπτει η ανάγκη για επιβεβαίωση και επαλήθευση. Αυτό τον ρόλο αναλαμβάνει να καλύψει το πεδίο της επαλήθευσης δικτύου ή όπως επικρατεί στην αγορά network validation. Παρακάτω θα δούμε το τί είναι η επαλήθευση δικτύου, θα συγκεντρώσουμε τους πυλώνες της και τις γνωστές διαδικασίες και θα τα αναλύσουμε. Στη συνέχεια θα δούμε πως μπορούμε να συνδυάσουμε τις αυτές τις διαδικασίες για να σχεδιάσουμε ένα pipeline επαλήθευσης δικτύου, ανάλογα τις απαιτήσεις του δικτύου.

2.3.1 Τι είναι η επαλήθευση δικτύου – network validation

Ως επαλήθευση δικτύου – network validation περιγράφεται η συνολική διαδικασία επαλήθευσης ενός δικτύου και της ορθότητάς του. Η διαδικασία αυτή αποτελείται από επιμέρους διαδικασίες που πραγματοποιούνται για την επαλήθευση των διαφορετικών συστατικών ενός δικτύου. Μια εύστοχη αναλογία για την περιγραφή του network validation είναι αυτή με το quality assurance και το software testing. Σκοπός του network validation είναι η επαλήθευση ότι όλα «δουλεύουν σωστά» και θα συνεχίσουν να «δουλεύουν σωστά». Αυτό σημαίνει ότι κάθε επιμέρους συσκευή μέσα στο δίκτυο είναι υγιείς και ορθώς παραμετροποιημένη, κάθε σύζευξη μεταξύ των συσκευών είναι ορθή και λειτουργεί και ότι όλες μαζί οι συσκευές σαν σύνολο λειτουργούν και εξυπηρετούν τον σκοπό για τον οποίο έχει στηθεί ένα δίκτυο. Επίσης το δίκτυο πρέπει να είναι δυνατόν να ανταποκριθεί επιτυχώς σε όσο το δυνατόν περισσότερα πιθανά σενάρια μπορεί να προκύψουν. Αυτό μπορεί να σημαίνει ότι μία πιθανή κατάρρευση ενός κόμβου ή μία αυξημένη κίνηση σε ένα δίκτυο δεν θα οδηγήσει σε κάποιο πρόβλημα. Τέλος επίσης σημαντικό το δίκτυο να επιτηρεί και να εκτιμά την συνεχώς την συνεχώς μεταβαλλόμενη κατάσταση του και να είναι σε θέση να επαληθεύσει την λειτουργία του κάθε δεδομένη στιγμή και αν όχι να εντοπίσει ή και να αποσφαλματώσει μόνο του το πρόβλημα [9].

2.3.2 Διαδικασίες του Network Validation.

Σημαντικό είναι να δομήσουμε και να ορίσουμε του «πυλώνες» της επαλήθευσης δικτύου, κάποιες από τις διαδικασίες, τις ενέργειες και τις μεθοδολογίες που πραγματοποιούνται πίσω από τον γενικό όρο network validation [10].

Οι τρεις πυλώνες – κατηγορίες που θα χωρίσουμε είναι:

- 1) Το εύρος της επαλήθευσης
- 2) Πότε γίνεται η επαλήθευση
- 3) Πως γίνεται η επαλήθευση

Το εύρος της επαλήθευσης

Το εύρος της επαλήθευσης είναι ίσως το πιο σημαντικό από τους τρεις πυλώνες. Έχει να κάνει με το εύρος και το βάθος των διαδικασιών επαλήθευσης. Δηλαδή με τα πόσα συστατικά και σε τι βαθμό και λεπτομέρεια αυτά θα ελεγχθούν. Οι διαδικασίες, ανάλογα το βάθος και το εύρος χωρίζονται στις:

- Unit testing
- Functional testing
- Verification

Unit testing – Έλεγχος συστατικών-μονάδων

Όπως και στο software testing, το unit testing σκοπό έχει να ελέγχει και επαληθεύει κάθε αυτόνομο συστατικό, μονάδα και συσκευή που βρίσκεται μέσα στο δίκτυο. Τέτοιοι έλεγχοι επικεντρώνονται στο ότι η συσκευή είναι online και τρέχει, στα ορθά configurations κάθε μίας συσκευής και των πρωτοκόλλων της όπως η ορθή παραμετροποίηση του DNS server της IP διεύθυνσης. Το unit testing είναι σχετικά μια γρήγορη και απλή μορφή ελέγχου, καθώς ελέγχει τοπικά κάθε συσκευή και μπορεί εύκολα να εντοπίσει λάθη και εύκολα να διορθωθούν. Παρ όλα αυτά υστερεί, καθώς δεν ελέγχει την end to end λειτουργία των συσκευών και της συνολικής συμπεριφοράς μεταξύ τους και δεν μπορεί να υπάρξει σαφής εικόνας της κατάστασης του δικτύου και πως λειτουργεί. Είναι δυνατόν δηλαδή, κάθε αυτόνομη συσκευή να είναι ορθώς παραμετροποιημένη, παρόλα αυτά το δίκτυο να αποτυγχάνει να δουλέψει και να επικοινωνήσει. Εδώ έρχεται να καλύψει το κενό η επόμενη κατηγορία.

Functional Testing – Έλεγχος λειτουργικότητας

Το functional testing κάνει ένα βήμα πιο βαθιά από το unit testing και έρχεται να καλύψει περισσότερα κενά. Ρόλος του είναι να ελέγχει την συμπεριφορά των συσκευών από άκρη σε άκρη (end – to – end), συνολικά στο δίκτυο και πως ανταποκρίνονται σε συγκεκριμένα σενάρια. Τέτοια σενάρια είναι το αν μπορούν 2 συσκευές να ανταλλάξουν πακέτα μεταξύ τους, αν έχουν επαφή με τους δρομολογητές του δικτύου. Μπορούμε να δούμε τη διαδικασία του functional testing ως το software integration testing του δικτύου. Το πλήθος όμως των διαφορετικών σεναρίων που μπορεί να προκύψουν σε ένα δίκτυο είναι υπερβολικά μεγάλο και να απαιτεί πολλούς υπολογιστικούς πόρους, έτσι το functional testing υστερεί ως προς την κάλυψη όλων των περιπτώσεων αφήνοντας ένα κενό στην πληρότητα την επαλήθευσης του δικτύου.

Verification – Επαλήθευση

Το verification έρχεται να καλύψει το κενό του functional testing, καλύπτοντας την πληρότητα των διαφορετικών πιθανών σεναρίων που μπορούν να προκύψουν μέσα σε ένα δίκτυο. Αυτό συμβαίνει με την χρήση μαθηματικών προσεγγίσεων και μοντέλων σε ένα καλά καθορισμένο περιβάλλον. Ένα τέτοιο παράδειγμα επαλήθευσης είναι η δυνατότητα ενός δικτύου να μένει ενεργό, παρά την οποιαδήποτε κατάρρευση συνδέσμου ή δρομολογητή στο δίκτυο, όπου μπορεί να προσομοιωθεί κάθε πιθανό σενάριο της κίνησης μέσα στο δίκτυο και να επαληθευτεί αν το δίκτυο είναι ανθεκτικό σε τέτοια σενάρια. Αυτό δίνει την δυνατότητα και την ασφάλεια στους μηχανικούς δικτύου να σχεδιάζουν δίκτυα μεγάλης κλίμακας με γρήγορους ρυθμούς όπου θα παραμένουν αξιόπιστα καθ' όλη την διάρκεια τους.

Πότε γίνεται η επαλήθευση

Ο δεύτερος βασικός πυλώνας της διαδικασίας επαλήθευσης δικτύου είναι χρονική στιγμή όπου πραγματοποιείται η επαλήθευση.

Δύο είναι οι περίοδοι όπου είναι δυνατόν να γίνεται η επαλήθευση:

- Post deployment – Μετά την εφαρμογή κάποιας αλλαγής στο δίκτυο.
- Pre deployment – Πριν την εφαρμογή κάποιας αλλαγής στο δίκτυο.

Post deployment

Η διαδικασία επαλήθευσης μετά την εφαρμογή μιας αλλαγής – Post deployment είναι η πιο απλή από τις δύο χρονικές περιόδους. Αφού γίνει μια αλλαγή στο δίκτυο για παράδειγμα, είτε στα configurations μίας συσκευής, είτε προστεθεί μία καινούργια συσκευή, ελέγχεται αν το δίκτυο παραμένει υγιές και αν όλα λειτουργούν όπως αναμένεται να λειτουργούν. Τα λάθη εντοπίζονται μέσω διαδικασιών αυτόματης ενημέρωσης από συστήματα monitoring, από έλεγχο της κατάστασης του δικτύου και την κίνησης σε αυτό από τους διαχειριστές του είτε μέσω είτε ακόμα και μέσω παραπόνων των χρηστών του δικτύου. Η διαδικασία αυτή όμως από μόνη της δεν είναι αρκετά ευέλικτη. Σε περίπτωση σφάλματος, το λάθος εμφανίζεται αφού έχει περάσει ήδη στο δίκτυο και το έχει επηρεάσει ήδη κάτι το οποίο προσθέτει έξτρα δυσκολία στους διαχειριστές. Επίσης οι διαδικασίες επαλήθευσης βάση της τρέχουσας κατάστασης του δικτύου, περιλαμβάνει σε μεγάλο ανθρώπινο παράγοντα, όπου οι μηχανικοί χειροκίνητα πραγματοποιούν ελέγχους. Και όπως έχει τονιστεί στο Κεφάλαιο 1, τα λάθη από ανθρώπινο παράγοντα αποτελούν το νούμερο 1 παράγοντα downtime σε ένα δίκτυο. Αυτές τις αδυναμίες έρχεται να συμπληρώσει η επαλήθευση πριν την εφαρμογή της αλλαγής.

Pre deployment

Το ενδεχόμενο να περάσει ένα σφάλμα στο δίκτυο, είναι σίγουρα μη επιθυμητό ενώ κιόλας θα ήταν δυνατόν να εντοπιστεί πριν καν ανέβει στο δίκτυο. Αυτόν τον ρόλο έρχεται να καλύψει η pre deployment επαλήθευση. Περιπτώσεις σφαλμάτων που μπορούν να εντοπιστούν μπορεί να προκύψουν από μία λάθος πληκτρολόγηση ενός διαχειριστή του δικτύου, μη καλή κατανόηση του διαχειριστή του δικτύου ή και εσφαλμένη γνώση της τρέχων κατάστασης του. Έτσι, με την επαλήθευση των αλλαγών πριν περάσουν στην παραγωγή, μπορεί πιαστούν και να αποφευχθούν σοβαρές βλάβες στο δίκτυο.

Πως γίνεται η επαλήθευση

Παραπάνω εξετάσαμε το εύρος της διαδικασίας επαλήθευσης, όπου είδαμε το βάθος και την λεπτομέρεια της επαλήθευσης των συστατικών του δικτύου. Στην συνέχεια είδαμε πότε χρονικά εφαρμόζουμε τις διαδικασίες επαλήθευσης. Τώρα θα δούμε τον τρίτο και τελευταίο πυλώνα της συνολικής επαλήθευσης δικτύων που εξετάζουμε το πως γίνεται η επαλήθευση αυτή, με ποια εργαλεία και σε ποια περίπτωση τα εφαρμόζουμε. Αυτές που θα δούμε είναι οι:

- Text analysis
- Emulation
- Operational state analysis
- Model-based analysis

Text analysis – Ανάλυση κειμένου

Κατά την διαδικασία text analysis – ανάλυσης κειμένου γίνεται έλεγχος των παραμέτρων των συσκευών του δικτύου και βασίζεται σε αυτές και μόνο τις πληροφορίες δίχως να γνωρίζει για την κατάσταση του υπόλοιπου δικτύου. Τέτοιοι έλεγχοι είναι για παράδειγμα ο έλεγχος ότι τα απαραίτητα πεδία των configurations έχουν γίνει αρχικοποίηση, ότι οι διευθύνσεις IP και DNS είναι ορθές. Κατά ανάλυση κειμένου, δεν υπάρχει γνώση του υπόλοιπου δικτύου και δεν είναι δυνατόν να γίνουν έλεγχοι για το πως αλληλοεπιδρούν τα συστατικά του δικτύου. Η ανάλυση κειμένου σε μία συνολική διαδικασία επαλήθευσης δικτύου χρησιμοποιείται για να γίνει Unit testing στα configurations στην φάση του pre-deployment .

Emulation - Προσομοίωση

Η προσομοίωση - emulation ενός δικτύου, γίνεται με τη βοήθεια λογισμικού σε ένα περιβάλλον – lab προσομοίωσης με τη χρήση ενός συνόλου εικόνων – virtual machines πραγματικών συσκευών που χρησιμοποιούνται στα δίκτυα. Σκοπός του emulation είναι η προσομοίωση ολόκληρης της τοπολογίας του δικτύου, των συστατικών της και των πρωτοκόλλων του δικτύου όσο το δυνατόν πιο πιστά γίνεται. Πάνω σε αυτήν την τοπολογία είναι δυνατόν να εκτελεστούν και να μελετηθούν πιθανά σενάρια που μπορούν να προκύψουν στο δίκτυο, να εφαρμοστούν διάφορα configurations ή να γίνει η προσθήκη μίας νέας συσκευής και να μελετηθεί η πιθανή συμπεριφορά του πραγματικού δικτύου πάνω σε αυτή την αλλαγή. Με αυτόν τον τρόπο, είναι δυνατόν να επιβεβαιωθεί ή όχι η σχεδίαση ενός πραγματικού δικτύου, να εντοπιστούν ευπάθειες του, να αναδυθούν σφάλματα στο λογισμικό των συσκευών του και πολλά ακόμη.

Παρόλα αυτά, η προσομοίωση ενός δικτύου, είναι μια τεχνική όπου απαιτεί υψηλούς υπολογιστικούς πόρους και πολλές φορές η πλήρης προσομοίωση ενός μεγάλης κλίμακας δικτύου, μπορεί να αποβεί αρκετά δύσκολη. Η προσομοίωση σε μία συνολική διαδικασία επαλήθευσης δικτύου χρησιμοποιείται στην Pre deployment φάση ενός δικτύου με σκοπό την επιβεβαίωση – verification την λειτουργίας του.

Operational state analysis – Ανάλυση κατάστασης λειτουργίας δικτύου.

Με την ανάλυση κατάστασης λειτουργίας δικτύου, οι διαχειριστές λαμβάνουν πληροφορίες για την τρέχουσα κατάσταση του δικτύου από τα συστατικά του. Με αυτές τις πληροφορίες μπορούν να ελέγξουν και να επαληθεύσουν τις επιπτώσεις που μπορεί να επιφέρει μία πιθανή αλλαγή στα configuration η μία προσθήκη συσκευής στο δίκτυο στο δίκτυο σε πραγματικό χρόνο. Παρόλα αυτά η διαδικασία αυτή έχει βασικά μειονεκτήματα. Κάθε έλεγχος που γίνεται, γίνεται αφού περάσει η οποιαδήποτε αλλαγή στο δίκτυο και όχι πριν και έτσι αν περάσει κάποιο σφάλμα στο δίκτυο, οι επιπτώσεις του θα είναι άμεσα φανερές και θα επηρεάσει την λειτουργία του δικτύου. Η ανάλυση κατάστασης λειτουργίας δικτύου πραγματοποιείται στην post deployment φάση της συνολικής διαδικασίας επαλήθευσης δικτύου με στόχο το την επιβεβαίωση - verification του δικτύου.

Model based analysis – Ανάλυση μοντελοποίησης δικτύου

Με την model based analysis χτίζουμε ένα μοντέλο του δικτύου. Αυτό το μοντέλο βασίζεται στις παραμετροποιήσεις – configurations που έχουμε εφαρμόσει στο δίκτυο αλλά σε πραγματικά δεδομένα από τις αλληλεπιδράσεις του δικτύου με υπόλοιπο διαδίκτυο. Αφού χτίσουμε το μοντέλο, κάνουμε αναλύσεις-queries πάνω σε αυτό και τρέχουμε κάθε πιθανό σενάριο με σκοπό την επαλήθευση της ορθής λειτουργίας του. Αυτή η μοντελοποίηση του διαδικτύου βασίζεται σε emulation όπως είδαμε παραπάνω και σε μαθηματικές μεθόδους. Η Ανάλυση μοντελοποίησης δικτύου ως τελικό σκοπό έχει την εξέταση και επαλήθευση όλων των πιθανόν σεναρίων που είναι δυνατόν να προκύψουν, κάτι το οποίο είναι αρκετά δύσκολο καθώς θέλει αρκετή ακρίβεια το μοντέλο και δεδομένα για να είναι όσο πιο κοντά στις πραγματικές συνθήκες. Μια κύρια διαφορά του με την προσομοίωση ενός δικτύου, είναι ότι δεν προσομοιώνει πραγματικές συσκευές με VMs με το λειτουργικό τους και έτσι δεν είναι να εντοπίσει σφάλματα όπως software bugs. Η ανάλυση μοντελοποίησης δικτύου πραγματοποιείται στην φάση pre – deployment με σκοπό τη πλήρες επιβεβαίωση-verification του δικτύου.

2.4 Network Validation Pipeline

Παραπάνω είδαμε τις διαδικασίες, το εύρος την επαλήθευσης, τη χρονική φάση που γίνεται η επαλήθευση και του τρόπους με τους οποίους γίνεται η επαλήθευση. Για την όσο αποτελεσματικότερη επαλήθευση ενός πραγματικού δικτύου, γίνεται χρήση ενός συνδυασμού όλων των παραπάνω διαδικασιών και εργαλείων στα διάφορα στάδια της επαλήθευσης δικτύου. Αυτό το συνδυασμό των διαδικασιών επαλήθευσης του δικτύου θα το ονομάσουμε network validation pipeline. Κατά τον σχεδιασμό όμως ενός network validation pipeline, υπάρχουν 2 σημαντικοί παράγοντες που πρέπει να ληφθούν υπόψη [11]. Αυτοί είναι:

Μέγεθος κάλυψης:

Ως μέγεθος της κάλυψης αναφερόμαστε στο εύρος των διαφορετικών σφαλμάτων που μπορούμε να εντοπίσουμε στο δίκτυο. Το εύρος αυτό εξαρτάται από την επιλογή του τρόπου πως θα γίνει η επαλήθευση. Για παράδειγμα στην Ανάλυση Μοντελοποίησης Δικτύου το εύρος των σφαλμάτων έχει να κάνει με τα όλα τα πιθανά σενάρια που μπορεί να προκύψουν αλλά δεν είναι σε θέση να εντοπίσει σφάλματα που μπορεί να βρίσκονται σε μία συσκευή όπως software bugs. Τέτοια σφάλματα είναι ικανή να καλύψει η Προσομοίωση – Emulation ενός δικτύου και των συσκευών και η Ανάλυση Λειτουργικής Κατάστασης Δικτύου.

Υπολογιστικοί πόροι:

Η ανάγκη για υπολογιστικούς πόρους για κάποια από τους τρόπους επαλήθευσης όπως emulation – simulation και Μοντελοποίηση δικτύου είναι μεγάλοι και ειδικά για μεγάλης κλίμακας δίκτυα και πολλές φορές δεν είναι η πλήρης προσομοίωση και επιλέγεται ένα υπό-σύνολο του δικτύου.

2.4.1 Validation pipelines

Παρακάτω θα παρουσιάσουμε 2 validation pipelines τα οποία έχουν σχεδιαστεί με γνώμονα τα παραπάνω κριτήρια. Το πρώτο θα είναι σχεδιασμένο με γνώμονα το

μέγεθος και εύρος της επαλήθευσης, το δεύτερο με βάση την εξοικονόμηση των υπολογιστικών πόρων προς χρήση.

1ο validation pipeline

Το πρώτο validation pipeline είναι σχεδιασμένο με γνώμονα το μέγεθος και το εύρος κάλυψης. Συνδυάζει τα εργαλεία της Ανάλυση Μοντελοποίησης Δικτύου, Προσομοίωσης Δικτύου και Ανάλυση Λειτουργικής Κατάστασης Δικτύου. Κάθε ένας από τους παραπάνω τρόπους έχει σκοπό να πιάσει διάφορες περιπτώσεις προς επαλήθευση. Με το πέρας όλης των παρακάτω διαδικασιών, παρέχεται η μεγαλύτερη δυνατή εγγύηση κάλυψης σφαλμάτων που μπορεί να προκύψουν.

Ανάλυση Μοντελοποίησης Δικτύου

Το pipeline επαλήθευσης ξεκινάει με την Ανάλυση μοντελοποίησης δικτύου. Εξετάζει ένα σύνολο από σημαντικών configuration και πιθανών σεναρίων και την συνολική εικόνα του δικτύου.

Προσομοίωση Δικτύου

Δεύτερο εργαλείο στο pipeline είναι η πλήρης προσομοίωση του δικτύου. Κάτι τέτοιο όμως μπορεί να αποβεί πολύ ακριβό σε υπολογιστικούς πόρους του προσομοιωτή. Καθώς υπάρχει επικάλυψη περιπτώσεων με την Ανάλυση Μοντελοποίησης Δικτύου του πρώτου βήματος, η προσομοίωση δικτύου μπορεί να προσαρμοστεί ώστε να εντοπίζει λάθη τα οποία δεν αναμένεται να εντοπιστούν στο πρώτο βήμα. Για παράδειγμα bugs στο λογισμικό των συσκευών και γενική συμπεριφορά των συσκευών. Σε περίπτωση όμως. Η χρήση του συνδυασμού των δύο πρώτων τεχνικών σε ίδια σεναρία είναι ικανή να παρέχει μεγαλύτερη σιγουριά και ασφάλεια για την κατάσταση του δικτύου.

Ανάλυση Κατάστασης Δικτύου

Τα δύο πρώτα βήματα στόχευαν στην pre – deployment φάση ενός δικτύου. Η δουλειά της Ανάλυσης Κατάστασης Δικτύου είναι να επαληθεύσει ότι τα configurations και οι αλλαγές που περάσανε στο πραγματικό δίκτυο, όντως ανταποκρίνονται στο επιθυμητό αποτέλεσμα και το δίκτυο είναι υγιές και δουλεύει όπως αναμενόταν.

2^ο Validation Pipeline

Το δεύτερο validation pipeline που θα δούμε έχει ως γνώμονα του υπολογιστικούς πόρους που είναι διαθέσιμοι. Το μεγάλο μέγεθος ενός δικτύου, η πολυπλοκότητα του, αλλά και η αδυναμία των προσομοιωτών δεν είναι δυνατή η πλήρης προσομοίωση του πραγματικού δικτύου, οδηγεί σε περιορισμούς.

Ανάλυση Μοντελοποίησης Δικτύου

Παρόμοια με το πρώτο pipeline, η διαδικασία επαλήθευσης ξεκινάει με την Ανάλυση Μοντελοποίησης Δικτύου. Μέσα από αυτήν την διαδικασία θα επαληθεύσει πιθανά configuration errors και όσο το δυνατόν περισσότερα σενάρια μπορούν να προκύψουν.

Ανάλυση Κατάστασης Δικτύου

Σε αντίθεση με το πρώτο network validation pipeline, σε αυτή την προσέγγιση έχουμε ως γνώμονα τους υπολογιστικούς πόρους και έτσι δεν προχωράμε σε διαδικασία Προσομοίωσης του Δικτύου καθώς κάτι τέτοιο είναι ακριβό υπολογιστικά. Έτσι περνάμε κατευθείαν στην Ανάλυση Κατάστασης δικτύου. Μειονέκτημα της προσέγγισης αυτής είναι ότι δεν είναι δυνατόν να ανιχνευθούν software bugs η να ελεγχθεί πως θα αντιδράσει το δίκτυο σε σφάλματα. Εδώ το πρώτο βήμα την Ανάλυσης Μοντελοποίησης Δικτύου αναλόγως και το πόσο ακριβές είναι έρχεται να καλύψει κάποια από τα κενά και να μειώσει την πιθανότητα σφάλματος. Παρόλα αυτά, με την πιο λεπτομερή Ανάλυση Κατάστασης Δικτύου τρέχοντας functional – integration ελέγχους, μπορούμε να εντοπίσουμε software bugs και ελέγχους ροής δεδομένων που θα κάναμε σε άλλη περίπτωση στην Προσομοίωση.

Παραπάνω παρουσιάσαμε συνοπτικά 2 network validation pipelines το ένα βελτιστοποιημένο προς το εύρος της κάλυψης των ελέγχων και το άλλο βελτιστοποιημένο ως προς το υπολογιστικούς πόρους. Σαφώς, αναλόγως του δικτύου και των εργαλείων που έχει μία ομάδα διαχειριστών και μηχανικών στα χέρια της, σχεδιάζει τις διαδικασίες επαλήθευσης, το εύρος και την ακρίβεια των ελέγχων που θα πραγματοποιήσουν και τις τεχνικές. Στο επόμενο κεφάλαιο θα μελετήσουμε τα εργαλεία επαλήθευσης δικτύων που υπάρχουν αυτή τη στιγμή στην αγορά και χρησιμοποιούνται σε πραγματικά δίκτυα για την αυτοματοποίηση και την επαλήθευση τους.

Κεφάλαιο 3. Εργαλεία προσομοίωσης και επαλήθευσης δικτύων.

3.1 Εισαγωγή

Τα πεδίο του network automation και κυρίως του network validation είναι νεαρά σε σχέση με την ηλικία της δικτύωσης. Με την πάροδο του χρόνου και την εξέλιξη των δικτύων εργαλεία επαλήθευσης δικτύου έχουν γίνει όχι μόνο χρήσιμα αλλά και απαραίτητα. Αφού είδαμε την χρησιμότητα του network validation και αναλύσαμε κάποιες από τις διαδικασίες του και τον συνδυασμό τους σε ένα network validation pipeline, σειρά έχει να δούμε υπαρκτά εργαλεία επαλήθευσης δικτύου.

Σε αυτό το κεφάλαιο θα δούμε μερικά από τα συνηθέστερα εργαλεία που έχουν αναπτυχθεί και χρησιμοποιούνται στον κόσμο του networking για αυτοματισμούς και επαλήθευση δικτύου. Θα δούμε τις δυνατότητες τους, τι προσφέρουν, τα χαρακτηριστικά τους και που χρησιμοποιούνται στον πραγματικό κόσμο. Τα εργαλεία αυτά είναι ανοιχτού κώδικα ή και όχι και καλύπτουν διαφορετικό φάσμα επαληθεύσεων και ακρίβειας.

3.2 Εργαλεία αυτοματισμών και επαλήθευσης

3.2.1 Ansible

Τι είναι το Ansible

Το Ansible είναι ένα cross-platform open-source εργαλείο αυτοματισμών IT με το οποίο γίνεται ο αυτοματισμός λειτουργιών και διαδικασιών στα δίκτυα. Τέτοιες λειτουργίες μπορεί να είναι το software provisioning, η διαχείριση των configurations, το λανσάρισμα μίας εφαρμογής στο δίκτυο (application deployment) . Είναι ένα χρήσιμο εργαλείο που διευκολύνει απλές αλλά και πολύπλοκες λειτουργίες για τη διαχείριση ενός περιβάλλοντος IT.

Πως δουλεύει το Ansible

Στο Ansible, υπάρχουν 2 κατηγορίες κόμβων-υπολογιστών. Ο κόμβος ελέγχου και οι κόμβοι προς διαχείριση. Για να δουλέψει, συνδέεται στους κόμβους προς διαχείριση ενός δικτύου (clients, servers, routers, switches) και στέλνει μικρά προγράμματα, τα οποία ονομάζονται modules. Στη συνέχεια, όντας συνδεδεμένο με τους κόμβους μέσω ssh, εκτελεί αυτά τα modules όπου ύστερα διαγράφονται. Για την σύνδεση μέσω ssh χρειάζονται δικαιώματα πρόσβασης από τον διαχειριστή, συνήθως SSH keys. Για την λειτουργία του Ansible δεν χρειάζεται η ύπαρξη εγκατάστασης του σε κάθε κόμβο προς διαχείριση παρά μόνο στον κόμβο ελέγχου, κάνοντας το ένα agentless εργαλείο.

Με τα modules αυτά είναι δυνατόν να γίνουν διαδικασίες παραμετροποίησης των κόμβων, αναβάθμισης των συσκευών και γενικά διαχείρισής τους

Κύρια συστατικά του Ansible

Το Ansible αποτελείται από ένα σύνολο βασικών συστατικών. Αυτά είναι:

- **Controller Machine** – Κόμβος ελέγχου - συσκευή του διαχειριστή: Αυτή είναι η συσκευή από την οποία διαχειρίζονται οι συσκευές μέσω SSH. Είναι η μόνη συσκευή η οποία απαιτεί την εγκατάσταση του Ansible
- **Inventory**: Περιέχει κάθε μία από τις συσκευές προς διαχείριση (hosts)

- **Playbook:** Αρχεία γραμμένα σε μορφή YAML τα οποία περιγράφουν τις ενέργειες τις οποίες θέλουμε να πραγματοποιήσουμε στους hosts. Αυτές οι ενέργειες αποτελούνται από τα παρακάτω χαρακτηριστικά ακολουθώντας την παρακάτω ιεραρχία:
 - ο **Play:** Περιέχει ένα σύνολο από ενέργειες – tasks
 - ο **Task:** Περιέχει ένα σύνολο από Modules
 - ο **Module:** Τα modules διαχειρίζονται τους πόρους του συστήματος όπως τις υπηρεσίες του, τα πακέτα, τα αρχεία ή εκτελούν εντολές συστήματος. Είναι τα μικρά προγράμματα που πραγματοποιούν τις ενέργειες που θέλουμε να πραγματοποιήσουμε



Εικόνα 3.2.1. Κύρια συστατικά του Ansible

Όπως θα δούμε και παρακάτω, το Ansible είναι ένα εργαλείο το οποίο μπορεί να ενσωματωθεί και με άλλα εργαλεία που χρησιμοποιούνται για την προσομοίωση και επαλήθευση δικτύων. Περισσότερες πληροφορίες για το Ansible αναφορά [12].

3.2.2 GNS3

Τι είναι το GNS3;

Το GNS3 – Graphical Network Simulator 3 είναι ένα δωρεάν cross platform λογισμικό ανοικτού κώδικα προσομοίωσης δικτύων το οποίο παρέχει την δυνατότητα προσομοίωσης πραγματικών πολύπλοκων δικτύων. Μέσω ενός διαδραστικού GUI δίνει τη δυνατότητα στο στήσιμο τοπολογιών δικτύων, χρησιμοποιώντας ένα πλήθος διαφορετικών εικονικών και πραγματικών συσκευών δικτύου.

Συσκευές

Η προσομοίωση-εξομοίωση των συσκευών που χρησιμοποιούνται στο δίκτυο-lab γίνεται με την χρήση images πραγματικών συσκευών (Virtual Machines) οι οποίες προσφέρονται από διάφορους vendors, για παράδειγμα της Cisco, Juniper. Αυτές οι συσκευές μπορεί να είναι:

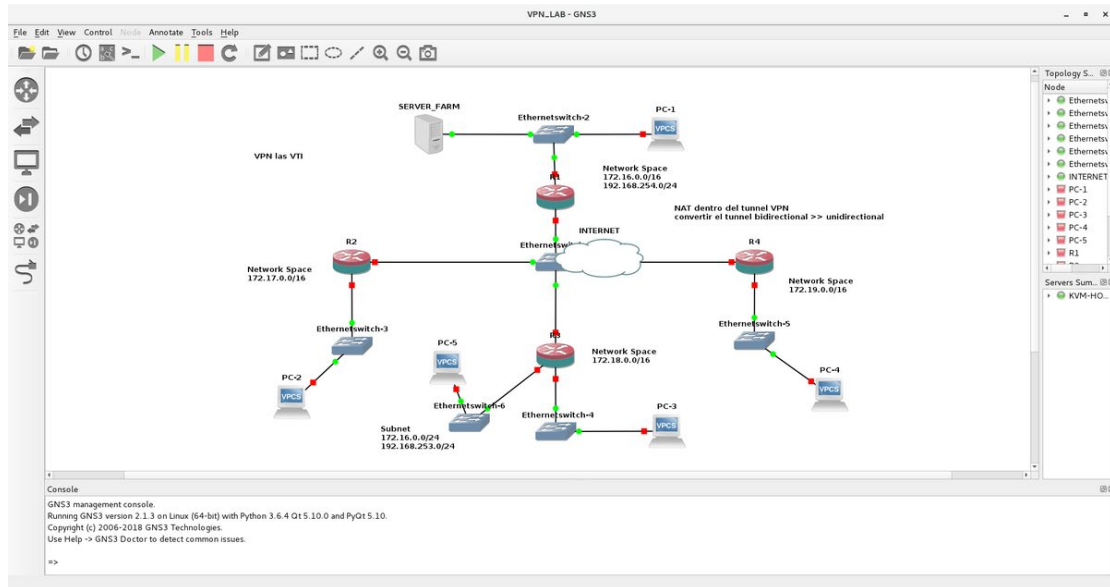
- Routers
- Switches
- Firewalls
- End-Point devices

Δυνατότητες

Το GNS3 δίνει τη δυνατότητα μίας προσομοίωσης ενός πραγματικού δικτύου και των πραγματικών συσκευών του και την παραμετροποίηση τους και των πρωτοκόλλων που χρησιμοποιούνται στα δίκτυα. Μέσα στις δυνατότητες του ενσωματώνει τη χρήση εργαλείων όπως custom python scripts και το λογισμικό Ansible που είδαμε παραπάνω με σκοπό την ανάπτυξη διαδικασιών αυτοματισμών και επαλήθευσης, όπως επίσης και το wireshark για ανάλυση πρωτοκόλλων δικτύου, κίνησης δικτύου κλπ.

Χρησιμότητα του GNS3

Το GNS3 είναι ένα εργαλείο το οποίο επιτρέπει σε επαγγελματίες και ερευνητές στο χώρο της δικτύωσης να σχεδιάσουν και να στήσουν τα δίκτυα τους, να τα παραμετροποιήσουν να ελέγξουν και να επαληθεύσουν τις υποθέσεις τους. Όπως αναφέραμε και στο κεφάλαιο επαλήθευσης δικτύου, είναι ένα εργαλείο που μπορεί να βρει χρησιμότητα στη Pre-deployment φάση επαλήθευσης ενός δικτύου με την χρήση της προσομοίωσης. Ευρέως είναι επίσης και η χρήση του εργαλείου για σκοπούς εκπαίδευσης και εκμάθησης πάνω σε θέματα δικτύωσης.



Εικόνα 3.2.2 Γραφικό περιβάλλον του GNS3

Περισσότερες πληροφορίες για το GNS3 αναφορά [13].

3.2.3 Batfish

Τι είναι το Batfish

Το Batfish είναι ένα open – source εργαλείο ανάλυσης δικτύου με σκοπό την επαλήθευση της ορθότητας του.

Λειτουργεί ως μία containerized υπηρεσία (Docker) και λειτουργεί offline, με αποτέλεσμα να μην χρειάζεται άμεση πρόσβαση στις συσκευές ενός δικτύου. Το batfish λειτουργεί χρησιμοποιώντας ένα σύνολο από configuration files των συσκευών ενός δικτύου -τα οποία ονομάζονται snapshots- τα οποία μετατρέπει σε μοντέλα, ανεξάρτητου κατασκευαστή των συσκευών. Στη συνέχεια κάνοντας αναλύσεις-queries πάνω σε αυτά τα μοντέλα το batfish επαληθεύει την ορθότητα των παραμέτρων ή όχι. Η αλληλεπίδραση με το εργαλείο Batfish γίνεται μέσω του rybatfish, μία βιβλιοθήκη για τη γλώσσα Python. Με τη βοήθεια αυτής της βιβλιοθήκης ο χρήστης μπορεί γράψει python scripts να κάνει τις αναλύσεις-queries πάνω στα μοντέλα με σκοπό την επαλήθευση.

Χρήσεις του Batfish

Το Batfish χρησιμοποιώντας μοντελοποίηση και προσομοίωση του δικτύου παρέχει ένα εύρος διαφορετικών δυνατοτήτων στα χέρια των μηχανικών δικτύου για να πραγματοποιήσουν διαδικασίες επαλήθευσης, που σε άλλη περίπτωση είναι πολύ δύσκολο να γίνουν. Μερικά από τα Use Cases είναι τα εξής:

- **Impact Analysis:** Δίνει την δυνατότητα προσομοίωσης το πως θα ανταποκριθεί το δίκτυο σε ένα σφάλμα, για παράδειγμα κατάρρευση ενός κόμβου, ενός interface
- **Configuration Auditing:** Δίνει τη δυνατότητα επαλήθευσης ότι τα configuration των συσκευών μέσα στο δίκτυο ακολουθούν τα standards που έχουν οριστεί
- **ACL:** Δίνει τη δυνατότητα επαλήθευσης των ACLs (Access Control Lists) ότι είναι ορθές επιτρέποντας η απαγορεύοντας τις ροές.

Χαρακτηριστικά του Batfish

- **Multi vendor Abstraction & support:** Το Batfish υποστηρίζει πληθώρα από διάφορους vendors όπως Arista, Cisco, Juniper. Συλλέγοντας τα δεδομένα από αυτές τις διάφορες συσκευές που υποστηρίζει, χτίζει ουδέτερα από vendor μοντέλα και στην έξοδο του επιστρέφει δεδομένα σε κοινές δομές δεδομένων
- **Packet forwarding Analysis:** Δίνει την δυνατότητα πραγματοποίησης εικονικών traceroutes και ελέγχους προσβασιμότητας μέσα στην τοπολογία δικτύου
- **Configuration Analysis:** Προσφέρει ένα εύρος από υποστηριζόμενα attributes των configuration files με τα οποία μπορούμε να κάνουμε queries πάνω στα μοντέλα, για παράδειγμα BGP, OSPF, ιδιότητες των interfaces κλπ.
- **Control plane Analysis:** Το Batfish αναπαριστά το control plane του δικτύου, δίνοντας τη δυνατότητα της επαλήθευσης των session του, τη γειτνίαση (BGP κ OSFP) κλπ.
- **ACL Analysis:** Παρέχει τη δυνατότητα της επαλήθευσης ορισμένων ροών δεδομένων μέσω ενός ACL.

- **Topology forking:** Παρέχει τη δυνατότητα επεξεργασίας της τοπολογίας, αφαιρώντας κόμβους ή συζεύξεων και να γίνει σύγκριση των πριν και μετά καταστάσεων με σκοπό την μελέτη των επιπτώσεων.

Τέλος, το Batfish παρέχει module το οποίο συνεργάζεται με το Ansible, δίνοντας τη δυνατότητα επαλήθευσης των αυτοματισμών και των configurations πριν περάσουν στο δίκτυο. Περισσότερες πληροφορίες για το Batfish στην αναφορά [14].

3.2.4 Suzieq

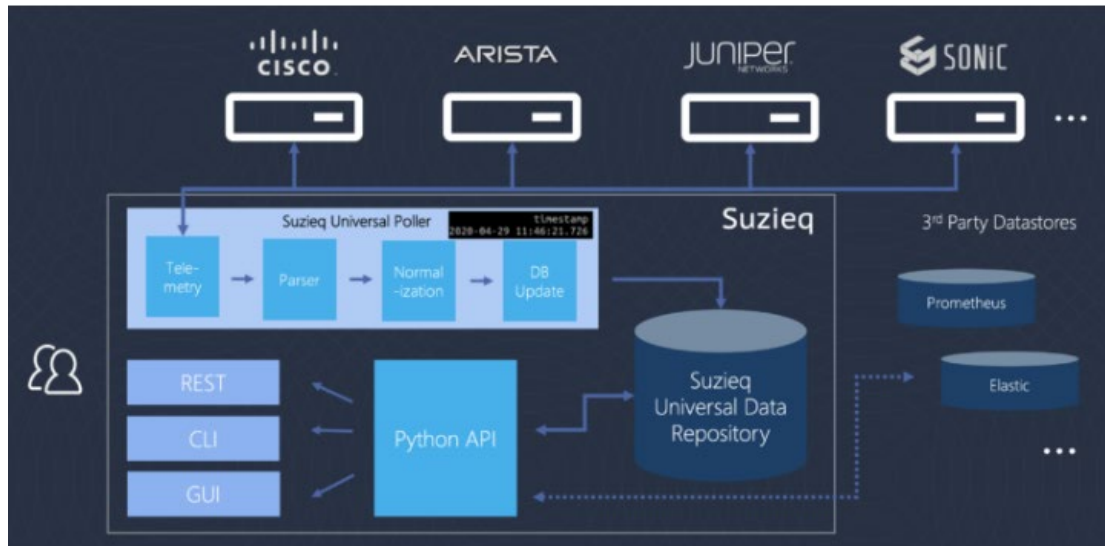
Τι είναι το Suzieq

Το suzieq είναι μία open-source, Multi-vendor πλατφόρμα παρακολούθησης δικτύου. Σκοπός του είναι η καλύτερη γνώση του δικτύου και παρακολούθηση του, και μέσω αυτής να γίνεται έγκαιρος εντοπισμός σφαλμάτων αλλά και προνοητική επαλήθευση αλλαγών σε ένα δίκτυο.

Πως δουλεύει το Suzieq

Χρησιμοποιώντας ένα agentless μοντέλο με τη βοήθεια SSH ή REST APIs, το suzieq συλλέγει δεδομένα σε πραγματικό χρόνο από συσκευές όπως routers, bridges, και Linux Servers, δεδομένα που σε άλλη περίπτωση θα χρειαζόταν log in σε κάθε συσκευή, στην συνέχεια μετατρέπει τα δεδομένα που έχει συλλέξει σε ουδέτερο-ανεξαρτήτου vendor format και τα αποθηκεύει. Ύστερα χρησιμοποιεί αυτά τα δεδομένα και μέσω CLI, GUI, REST API ή Python πραγματοποιεί αναλύσεις για την κατάσταση των συσκευών και του δικτύου, κάνει έγκαιρο εντοπισμό σφαλμάτων και επαλήθευση αλλαγών δικτύου.

Το suzieq, καθώς μαζεύει δεδομένα από συσκευές του δικτύου σε πραγματικό χρόνο και κάνει ανάλυση της κατάστασης του δικτύου αλλά κάνει και επαλήθευση αλλαγών πριν περάσουν στο δίκτυο, είναι ένα εργαλείο που μπορεί να χρησιμοποιηθεί στην Pre-deployment φάση επαλήθευσης ενός δικτύου αλλά και στην Post deployment φάση κάνοντας Operational state analysis.



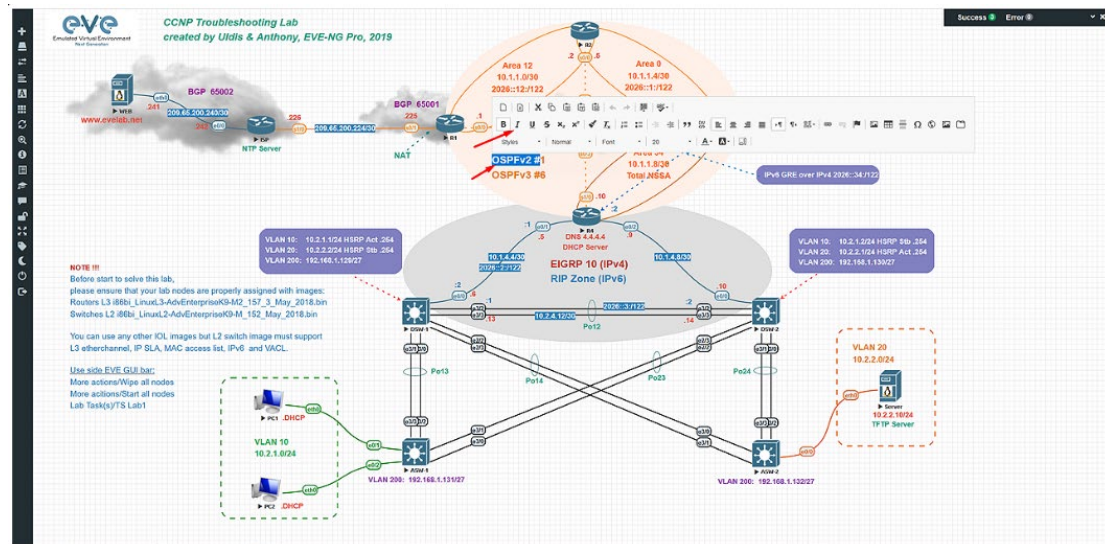
Εικόνα 3.2.3 Αρχιτεκτονική του Suzieq

Περισσότερες πληροφορίες για το Suzieq στην αναφορά [15]

3.2.5 EVE-NG

Τι είναι το EVE-NG

Το Emulated Virtual Environment – Next Generation (EVE – NG) είναι ένας clientless multi-vendor εξομοιωτής δικτύων – network emulator. Προσφέρει ένα γραφικό περιβάλλον (GUI) μέσω browser με το οποίο μπορούμε να δημιουργήσουμε Labs-τοπολογίες δικτύου. Οι τοπολογίες αυτές αποτελούνται από κόμβους-συσσκευές δικτύου όπου είναι εικόνες (VMs) πραγματικών συσκευών. Το εργαλείο τρέχει πάνω σε virtual machine συνεπώς μπορεί να χρησιμοποιηθεί σε διάφορα λειτουργικά συστήματα.



Εικόνα 3.2.4. Γραφικό περιβάλλον του EVE-NG

Χρήσεις του EVE-NG

Η χρήση του εξομοιωτή EVE-NG μπορεί να βρει εφαρμογή σε πληθώρα περιπτώσεων. Μπορεί να χρησιμοποιηθεί για την εκμάθηση της δικτύωσης των υπολογιστών και για μελέτη συσκευών και τεχνολογιών που χρησιμοποιούνται στο χώρο. Μπορεί να βρει εφαρμογή σε πραγματικά δίκτυα και μέσω της προσομοίωσης τους να χρησιμοποιηθεί για network automation και network validation. Μπορεί επίσης να χρησιμοποιηθεί ως proof of concept για πελάτες ή ακόμα και για software testing σε ένα δίκτυο, testing του συστήματος ασφάλειας κλπ.

Συγκεκριμένα, στο πεδίο του network automation, μπορούμε να ενσωματώσουμε το εργαλείο Ansible που παρουσιάσαμε παραπάνω για να εφαρμόσουμε διαδικασίες αυτοματισμών στις συσκευές και στα configurations τους.

Στο πεδίο του network validation, μπορούμε να κάνουμε χρήση των προσομοιώσεων για να δούμε πως συμπεριφέρεται το δίκτυο σε διάφορα σενάρια και configurations τα οποία έχουμε εμείς ορίσει. Επίσης, δυνατή είναι και η χρήση του εργαλείου pyATS που θα παρουσιάσουμε παρακάτω με σκοπό την επαλήθευση της τοπολογίας. Τέλος δυνατή είναι και η χρήση του εργαλείου Wireshark για παρακολούθηση πακέτων και κίνησης των δικτύων.

Περισσότερες πληροφορίες για το EVE-NG στην αναφορά [16].

3.2.6 Cisco packet tracer

Τι είναι το Cisco Packet Tracer

Το Packet Tracer είναι ένα δωρεάν multi-platform εργαλείο προσομοίωσης το οποίο δίνει τη δυνατότητα προσομοίωσης εικονικών τοπολογιών δικτύων. Αυτές οι τοπολογίες δικτύου αποτελούνται από εικόνες πραγματικών συσκευών και εκτελούνε αντίστοιχα πρωτόκολλα δικτύων.

Χρήση του Packet Tracer

Το Packet Tracer είναι ένα εργαλείο που χρησιμοποιείται κατά κόρον στην εκπαίδευση φοιτητών αλλά και μελλοντικών μηχανικών δικτύων. Είναι ιδανικό για εκμάθηση βασικών εννοιών των δικτύων, εξοικείωση με τις συσκευές που απαρτίζουν ένα δίκτυο και παραμετροποίησή τους, τεχνικών σχεδίασης τοπολογιών δικτύου αλλά και πως λειτουργούν τα σημαντικότερα πρωτόκολλα που χρησιμοποιούνται σήμερα στην δικτύωση των υπολογιστών

Υποστηριζόμενα Πρωτόκολλα

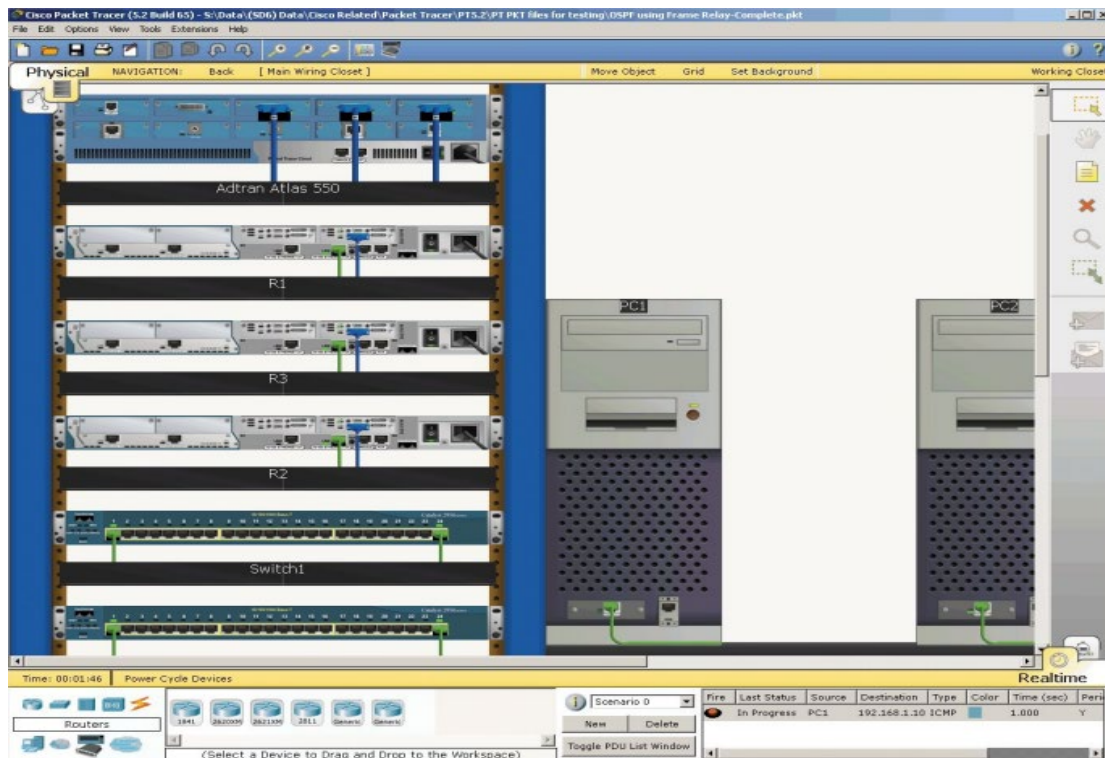
Layer	Cisco Packet Tracer Supported Protocols
Application	▪ FTP, SMTP, POP3, HTTP, TFTP, Telnet, SSH, DNS, DHCP, NTP, SNMP, AAA, ISR VOIP, SCCP config and calls ISR command support, Call Manager Express
Transport	▪ TCP and UDP, TCP Nagle Algorithm & IP Fragmentation, RTP
Network	▪ BGP, IPv4, ICMP, ARP, IPv6, ICMPv6, IPsec, RIPv1/v2/ng, Multi-Area OSPF, EIGRP, Static Routing, Route Redistribution, Multilayer Switching, L3 QoS, NAT, CBAL, Zone-based policy firewall and Intrusion Protection System on the ISR, GRE VPN, IPsec VPN
Network Access/ Interface	▪ Ethernet (802.3), 802.11, HDLC, Frame Relay, PPP, PPPoE, STP, RSTP, VTP, DTP, CDP, 802.1q, PAgP, L2 QoS, SLARP, Simple WEP, WPA, EAP

Εικόνα 3.2.5 Υποστηριζόμενα πρωτόκολλα του εργαλείου ανά επίπεδο.

Κύρια χαρακτηριστικά του packet tracer

Το packet tracer προσφέρει ένα διαδραστικό περιβάλλον οπτικοποίησης και καθιστά εύκολη την προσθήκη των συσκευών και την πραγματοποίηση συνδέσεων μεταξύ τους.

The screenshot displays the Cisco Packet Tracer 5.2 Build 65 interface. The workspace shows a detailed network topology. At the top, a 2811 Router2 is connected to a 3560-24PS Central switch. This central switch is connected to several 2950 switches and 2960 switches. These switches are further connected to various servers (Server-PT, Workgroup Server 1, Workgroup Server 2, Workgroup Server 3) and multiple PCs (A1-A4, B1-B4, C1-D4, E1-F4). The network is interconnected with multiple links, some highlighted in red. The interface includes a top menu bar, a toolbar, and a bottom status bar showing the time as 00:01:50.



Εικόνα 3.2.7. Γραφικό περιβάλλον του *physical workspace* όπου απεικονίζει πραγματικές συσκευές.

Το packet tracer επίσης προσφέρει 2 modes για την οπτικοποίηση της συμπεριφοράς του δικτύου, το real-time mode και το simulation mode. Στο λειτουργία πραγματικού χρόνου real-time mode το δίκτυο και οι συσκευές του συμπεριφέρονται όπως θα συμπεριφέρονταν σε πραγματικό χρόνο με άμεση απόκριση. Στην λειτουργία προσομοίωσης simulation mode ο χρήστης μπορεί να προσδιορίσει τα χρονικά διαστήματα λειτουργίας, να πειραματιστεί με το πως λειτουργεί η μεταφορά δεδομένων, αποτελώντας έτσι ένα πολύ καλό εργαλείο κατανόησης της δικτύωσης των υπολογιστών και των εννοιών της.

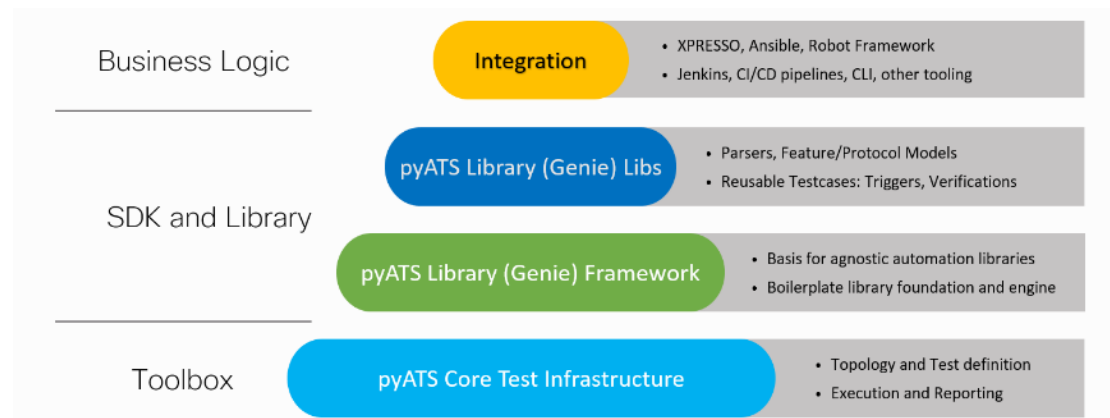
Περισσότερες πληροφορίες για το Cisco Packet Tracer στην αναφορά [17].

3.2.7 pyATS

Τι είναι το οικοσύστημα του pyATS

Το pyATS – Python Automated Test Systems και η βιβλιοθήκη pyATS ορίζουν το οικοσύστημα του pyATS το οποίο έχει σκοπό την δημιουργία αυτοματοποιημένων διαδικασιών παραμετροποίησης, διαχείρισης επαλήθευσης και ελέγχου των δικτύων.

Αναπτύχθηκε από την Cisco, αρχικά για ιδιωτική χρήση της εταιρείας και στη συνέχεια έγινε open-source και χρησιμοποιείται σήμερα από πολλούς network engineers, DevNet engineers και developers. Προσφέρει λύσεις αυτοματισμού, ελέγχων και επαλήθευσης δικτύων σε δίκτυα μεγάλης κλίμακας και υποστηρίζει multi-vendor συσκευές. Τέτοιες συσκευές μπορεί να είναι Routers, switches, access points, firewalls, Linux servers, τηλέφωνα, καλώδια και άλλα πολλά.



Εικόνα 3.2.8. Διάγραμμα οικοσυστήματος του pyATS

pyATS

Το pyATS είναι το test framework του οικοσυστήματος το οποίο εκτελεί data-driver testing και είναι ικανό για rapid development iterations.

Βιβλιοθήκη pyATS

Η βιβλιοθήκη του pyATS η αλλιώς Genie παρέχει όλα τα απαραίτητα εργαλεία για αυτόματο testing όπως:

- Multi-vendor parsers
- Triggers
- APIs
- CLI
- Χρήσιμες βιβλιοθήκες

Χρήσεις της βιβλιοθήκης pyATS

Το pyATS είναι ένα εργαλείο που βοηθάει σε καθημερινές εργασίες DevOps όπως:

- Καταγραφή την τρέχων κατάστασης του δικτύου και δημιουργία εικόνας-snapshot για εκ των υστέρων σύγκριση της κατάστασης του
- Αυτόματο monitoring των συσκευών του δικτύου
- Αυτοματισμός παραμετροποίησης και αναβαθμίσεων συσκευών
- Προσθήκη συσκευών και παρακολούθηση συμπεριφοράς του δικτύου.

Πλεονεκτήματα του ryATS στα χέρια των developers και των μηχανικών δικτύων:

- Parsing εξόδων συσκευών για επαλήθευση της λειτουργίας του
- Επαναχρησιμοποίηση διαθέσιμων test cases
- Δυνατότητα σύνδεσης σε συσκευές με Telnet, SSH, REST, RESTCONF, NETCONF και YANG
- Συλλογή δεδομένων από το δίκτυο για εκτίμηση της κατάστασης του
- Δημιουργία snapshot του δικτύου για παρακολούθηση της πριν και μετά κατάστασης μετά από μία αλλαγή
- Δημιουργία snapshot συσκευών για παρακολούθηση της κατάστασης των συσκευών πριν και μετά

Πως δουλεύει το οικοσύστημα του ryATS

Το ryATS παρέχει ένα οικοσύστημα όπου καθορίζει

- Πως ορίζονται οι τοπολογίες και πως μοντελοποιούνται
- Πως γίνεται προγραμματιστικά η επικοινωνία με τις συσκευές
- Πως ορίζονται και εκτελούνται τα test scripts
- Πως πραγματοποιούνται οι έλεγχοι και πως παράγονται οι αναφορές

Το οικοσύστημα του ryATS προσφέρει δυνατότητα ενσωμάτωσης με άλλα εργαλεία frameworks της αγοράς όπως το Ansible, XPRESSO JENKINS αλλά και τους simulators που παρουσιάσαμε παραπάνω όπως το GNS3 και το EVE-NG.

Περισσότερες πληροφορίες για το ryATS στην αναφορά [18].

Κεφάλαιο 4. Το δικό μας εργαλείο προσομοίωσης δικτύου

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο, αφού παρουσιάσαμε παραπάνω τα εργαλεία που υπάρχουν και χρησιμοποιούνται στον κόσμο της δικτύωσης από επαγγελματίες και μη, σειρά έχει να παρουσιάσουμε το εργαλείο που αναπτύξαμε εμείς. Το δικό μας εργαλείο επαλήθευσης δικτύου, αποτελεί μία εισαγωγική ιδέα για ένα ολοκληρωμένο εργαλείο όπου σκοπός είναι η προσομοίωση μίας τοπολογίας δικτύου, η οπτικοποίησή της, η επαλήθευση και η επαλήθευση της - εντοπισμός λαθών σε αυτήν. Οι έλεγχοι που γίνονται, αποτελούν βασικούς ελέγχους στα configurations της κάθε συσκευής αλλά και της συνολικής συμπεριφοράς και λειτουργικότητας μεταξύ τους μέσα σε ένα δίκτυο.

Παρακάτω θα παρουσιάσουμε τις δυνατότητες που προσφέρει το εργαλείο μας, τα χαρακτηριστικά του, τη λογική του, κάποιες λεπτομέρειες από τα εργαλεία που χρησιμοποιήσαμε για την ανάπτυξή του και τέλος τις μελλοντικές επεκτάσεις που μπορεί να υποστηρίξει στο μέλλον ώστε να καταλήξει να γίνει ένα ολοκληρωμένο εργαλείο προσομοίωσης.

4.2 Δυνατότητες του εργαλείου

Ξεκινώντας, θα δούμε τις δυνατότητες που προσφέρει ο προσομοιωτής για επαλήθευση δικτύου. Σκοπός του εργαλείου είναι μία αρχική προσομοίωση της τοπολογίας ενός δικτύου και των συσκευών που την αποτελούν και της οπτικοποίησης της. Μέσα σε αυτήν την τοπολογία, θέλουμε να επαληθεύσουμε την ορθότητα των παραμέτρων κάθε συσκευής και των interfaces της που έχουν οριστεί, να επαληθεύσουμε την ορθότητα των υπό-δικτύων που αποτελούν το δίκτυο και την συνολική συμπεριφορά του δικτύου και των συσκευών του.

Οι έλεγχοι που πραγματοποιούνται, γίνονται σε δύο φάσεις:

- 1) **Στα configuration files των συσκευών**
- 2) **Στα δίκτυα και στα υπό-δίκτυα της τοπολογίας**

Στα configuration files των συσκευών

Σε αυτή την φάση πραγματοποιούνται έλεγχοι πάνω στα configuration files κάθε μίας συσκευής. Οι έλεγχοι αυτοί γίνονται πριν γίνει η παραμετροποίηση των συσκευών και πριν γίνει η προσομοίωση της τοπολογίας. Συγκεκριμένα γίνονται οι έλεγχοι:

- **Επαλήθευση ορθότητας του format των configuration files:**
 - ο Γίνεται έλεγχος ότι κάθε αρχείο ακολουθεί το format που του netplan που χρησιμοποιούμε για την παραμετροποίηση των συσκευών. Παρακάτω βρίσκεται η περιγραφή για το format αυτό.
- **Επαλήθευση αρχικοποίησης των interfaces**
 - ο Σε αυτή τη διαδικασία εξασφαλίζεται ότι για κάθε interface κάθε μίας συσκευής που χρησιμοποιείται και έχει γίνει αρχικοποίηση στο TGF αρχείο όπου ορίζουμε την τοπολογία, έχει γίνει η αρχικοποίηση του στα configuration files της κάθε συσκευής
- **Επαλήθευση ορθότητας των πεδίων των configuration files.**
 - ο Γίνεται έλεγχος ότι αναλόγως τον τύπο της συσκευής που παραμετροποιούμε, έχουμε και τα κατάλληλα πεδία στο configuration file και δεν λείπει κάποιο από αυτά. Πχ: Το configuration file ενός client pc αναμένεται να έχει πεδίο διεύθυνσης IP ή κάθε δρομολογητής αναμένεται να έχει έναν πίνακα δρομολόγησης. Παρακάτω αναγράφουμε τα απαραίτητα πεδία ανά τύπο συσκευής.

- **Επαλήθευση ορθότητας των τιμών των πεδίων των configuration files.**

Κάθε τιμή των πεδίων στο αρχείο πρέπει να είναι μία έγκυρη τιμή.

- **Έγκυρη διεύθυνση IP:** Μία έγκυρη διεύθυνση IP «x.x.x.x». αποτελείται από τέσσερα πεδία που χωρίζονται μεταξύ τους με τελεία (.) και περιέχουν έναν ακέραιο αριθμό x που ανήκει στο εύρος 0-255 και δεν περιέχει προπορευόμενα μηδενικά.
- **Έγκυρη διεύθυνση MAC:** Μία έγκυρη MAC διεύθυνση αποτελείται από 12 δεκαεξαδικά ψηφία χωρισμένα σε έξι ζεύγη των δύο ψηφίων που χωρίζονται μεταξύ τους με τους χαρακτήρες (-) ή (:).
- **Έγκυρη διεύθυνση gateway:** Μία έγκυρη διεύθυνση IP.
- **Έγκυρη διεύθυνση DNS:** Μία έγκυρη διεύθυνση IP
- **Έγκυρη μάσκα υπό-δικτύου:** Ένας ακέραιος αριθμός /x όπου ανήκει στο εύρος 0-32.
- **Έγκυρη τιμή NAT:** Το NAT που είναι πεδίο στα configuration files των δρομολογητών αποτελείται από 2 πεδία. Από το status όπου παίρνει τις τιμές enabled-disabled και το public ip που πρέπει να είναι μία έγκυρη διεύθυνση IP.

- **Επαλήθευση ορθότητας των πινάκων δρομολόγησης των δρομολογητών.**

Κάθε δρομολογητής πρέπει να έχει έγκυρο πίνακα δρομολόγησης. Αυτό σημαίνει:

- **Έγκυρες τιμές στις εγγραφές:** Κάθε διεύθυνση που βρίσκεται στις εγγραφές του πίνακα πρέπει να είναι μία έγκυρη διεύθυνση IP η οποία ακολουθείται από την μάσκα υπό-δικτύου.
- **Έγκυρο μήκος του πίνακα δρομολόγησης:** Κάθε μία από τις εγγραφές πρέπει να έχει ένα επόμενο άλμα.

Αφού ολοκληρωθεί επιτυχώς η επαλήθευση τους περνάμε στη δεύτερη φάση. Στη δεύτερη φάση, έχουμε κάνει παραμετροποίηση τις συσκευές και μοντελοποίηση το δίκτυο και οι έλεγχοι αφορούν τη λειτουργία του δικτύου και των υπό-δικτύων ως σύνολο. Σε περίπτωση εντοπισμού λάθους, το πρόγραμμα δεν προχωράει στην μοντελοποίηση του δικτύου και τερματίζει την λειτουργία του.

Στα δίκτυα και στα υπό-δίκτυα της τοπολογίας

Σε αυτή τη φάση έχει ολοκληρωθεί επιτυχώς η επαλήθευση των configuration files, έχει γίνει η μοντελοποίηση του δικτύου και εξετάζουμε τη συνολική συμπεριφορά του δικτύου και των υπό-δικτύων του. Οι έλεγχοι που πραγματοποιούνται είναι οι εξής:

- **Έλεγχος μοναδικότητας διευθύνσεων MAC**
 - ο Σε αυτή τη διαδικασία, γίνεται έλεγχος για το αν κάθε μία διεύθυνση MAC που ανήκει στο interface μίας συσκευής, είναι μοναδική για το υπό-δίκτυο που ανήκει. Μία διεύθυνση MAC πρέπει να είναι μοναδική μέσα σε ένα υπό-δίκτυο.
- **Έλεγχος μοναδικότητας IP υπό-δικτύων**
 - ο Σε αυτή τη διαδικασία, ελέγχεται για το αν κάθε ένα υπό-δίκτυο μέσα στην τοπολογία είναι μοναδικό. Δεν είναι δυνατόν να υπάρχουν διπλότυπα IP υπό-δίκτυα εκτός την περίπτωση όπου το υπό-δίκτυο αυτό είναι ιδιωτικό, δηλαδή εκτελεί NAT (Network Address Translation).
- **Έλεγχος ορθά ρυθμισμένης διεύθυνσης gateway**
 - ο Σε αυτή τη διαδικασία, γίνεται έλεγχος ότι το interface μίας συσκευής μέσα σε ένα υπό-δίκτυο είναι ίδιο με το πραγματικό gateway που έχει ρυθμιστεί για το υπό-δίκτυο
- **Έλεγχος ορθά ρυθμισμένης διεύθυνσης IP**
 - ο Σε αυτή τη διαδικασία γίνεται έλεγχος, ότι μία IP διεύθυνση που βρίσκεται στο interface μίας συσκευής μέσα σε ένα υπό-δίκτυο, ανήκει στο υπό-δίκτυο που ανήκει.
- **Έλεγχος μοναδικότητας διεύθυνσης IP**
 - ο Σε αυτή τη διαδικασία γίνεται έλεγχος για διπλότυπες διευθύνσεις IP. Μία διεύθυνση IP πρέπει να είναι μοναδική μέσα σε ένα υπό-δίκτυο. Η ύπαρξη διπλότυπης διεύθυνσης μπορεί να προκαλέσει πολλά προβλήματα στην επικοινωνία.

- **Έλεγχος πινάκων δρομολόγησης**
 - ο Σε αυτή τη διαδικασία γίνονται έλεγχοι επαλήθευσης των πινάκων δρομολόγησης. Συγκεκριμένα:
 - Με τη χρήση των πινάκων δρομολόγησης γίνεται έλεγχος ότι όλες οι συσκευές μέσα σε ένα υπό-δίκτυο μπορούν να επικοινωνήσουν μεταξύ τους.
 - Γίνεται έλεγχος για την ύπαρξη βρόγχων στον πίνακα δρομολόγησης
 - Γίνεται έλεγχος ότι κάθε next hop σε μία εγγραφή του πίνακα δρομολόγησης είναι όντως μία διεύθυνση IP με την οποία ο δρομολογητής είναι συνδεδεμένος.

4.3 Χαρακτηριστικά και λογική του εργαλείου

Παρακάτω θα δείξουμε τα χαρακτηριστικά την λογική με την γίνεται μια εκτέλεση διαδικασίας επαλήθευσης μίας δικής μας τοπολογίας δικτύου. Θα δείξουμε ποιες διαφορετικές συσκευές υποστηρίζονται, τι χαρακτηριστικά υποστηρίζουν τα διάφορα interface των συσκευών, ποια attributes περιλαμβάνει μία σύζευξη μεταξύ δύο συσκευών. Θα δούμε που και πως ορίζεται η τοπολογία, πως και που γίνεται η συγγραφή των configurations files και ύστερα η παραμετροποίηση των συσκευών με αυτά. Τέλος θα δείξουμε πως ξεκινά η διαδικασία επαλήθευσης δικτύου και η οπτικοποίηση του.

4.3.1 Χαρακτηριστικά του εργαλείου

Η τοπολογία του δικτύου αναπαρίσταται ως ένα γράφημα. Σε αυτό το γράφημα κάθε ένας κόμβος είναι μία συσκευή και κάθε ακμή μεταξύ των κόμβων είναι μία σύζευξη-link μεταξύ των συσκευών.

Κόμβοι - Συσκευές

Στο εργαλείο μας επαλήθευσης δικτύου, κάθε συσκευή ορίζεται ως ένας κόμβος. Υποστηρίζονται 5 διαφορετικές συσκευές. Αυτές είναι:

- Hub
- Switch
- Client PC
- Server PC
- Router

Για κάθε μία από αυτές τις συσκευές ορίζεται ένα configuration file που περιέχει όλες τις παραμέτρους απαραίτητες για την αρχικοποίηση των interface των συσκευών. Μία συσκευή είναι δυνατόν να έχει πολλαπλά διαφορετικά interfaces. Ανάλογα τον τύπο της συσκευής έχουμε και διαφορετικές παραμέτρους στα interface. Οι παράμετροι ενός interface μίας συσκευής αναλόγως το είδος της είναι:

- **Hub**
 - ID του interface
- **Switch**
 - ID του interface
 - Διεύθυνση MAC
- **Client PC**
 - ID του interface
 - Διεύθυνση MAC
 - Διεύθυνση IP
 - Gateway
 - DNS
- **Server PC**
 - ID του interface
 - Διεύθυνση MAC
 - Διεύθυνση IP
 - Gateway
 - DNS
- **Router**
 - ID του interface
 - Διεύθυνση MAC
 - Διεύθυνση IP
 - Gateway
 - DNS
 - NAT
 - status
 - public ip

Στην συνέχεια θα δούμε με περισσότερη λεπτομέρεια πως ορίζονται τα παραπάνω στα configuration files και στο αρχείο της τοπολογίας.

Σύζευξη - link - Ακμή

Σύζευξη στην τοπολογία επιτυγχάνεται μεταξύ δύο διαφορετικών συσκευών. Στα άκρα της σύζευξης βρίσκεται το interface της αντίστοιχης συσκευής που παίρνει μέρος. Μία σύζευξη επίσης περιλαμβάνει και attributes-χαρακτηριστικά της.

Αυτά είναι:

- Το ID της σύζευξης
- Το interface της συσκευής που ανήκει στο αριστερό άκρο της σύζευξης
- Το interface της συσκευής που ανήκει στο δεξί άκρο της σύζευξης
- Το bandwidth του συνδέσμου
- Το latency του συνδέσμου

Στην συνέχεια θα δούμε με περισσότερη λεπτομέρεια πως ορίζονται τα παραπάνω στο αρχείο της τοπολογίας.

4.3.2 Λογική του εργαλείου

Συνολικά τα βήματα που ακολουθεί το εργαλείο για το στήσιμο της τοπολογίας του δικτύου και της επαλήθευσης είναι τα εξής:

- Ορισμός Τοπολογίας
- Αρχικοποίηση των configuration files
- Επαλήθευση των configuration files
- Παραμετροποίηση των συσκευών της τοπολογίας με την χρήση των configuration files
- Μοντελοποίηση του δικτύου
- Οπτικοποίηση
- Επαλήθευση δικτύου

Συνολικά για το στήσιμο της τοπολογίας και την επαλήθευση της χρειαζόμαστε τα εξής αρχεία.

- Μη παραμετροποιημένο αρχείο τοπολογίας
- Configuration files
- Παραμετροποιημένο αρχείο τοπολογίας
- Αρχείο οπτικοποίησης

4.3.2.1 Ορισμός Τοπολογίας

Κατά τον ορισμό την τοπολογίας, ορίζουμε την τοπολογία του δικτύου σε ένα αρχείο μορφής `tgf` (trivial graph format). Το Trivial Graph Format [19] είναι ένα απλό text-based format αρχείου με λίστες γειτνίασης που περιγράφουν ένα γράφημα. Εμείς προσαρμόσαμε σε μικρό βαθμό το format αυτό κάνοντας κάποιες προσθήκες ώστε να ταιριάζει με τις ανάγκες μας. Αυτό το αρχείο που ορίζουμε αρχικά την τοπολογία είναι μη παραμετροποιημένο, δηλαδή το χρησιμοποιούμε για να ορίσουμε τις συσκευές και τα συζεύξεις, δίχως παραμέτρους.

Οι κόμβοι-συσκευές του δικτύου ορίζονται ως εξής:

Hub:

```
h1 ~ {'node_type' : 'Hub'};
```

Εικόνα 4.3.1. Δήλωση συσκευής Hub

Switch:

```
sw1 ~ {'node_type' : 'Switch'};
```

Εικόνα 4.3.2. Δήλωση συσκευής switch

Client PC

```
pc1 ~ {'node_type' : 'Client_PC'};
```

Εικόνα 4.3.3. Δήλωση συσκευής Client PC

Server PC

```
sr1 ~ {'node_type' : 'Server_PC'};
```

Εικόνα 4.3.4. Δήλωση συσκευής Server PC

Router

```
r1 ~ {'node_type' : 'Router'};
```

Εικόνα 4.3.5. Δήλωση συσκευής Router

Στον ορισμό των κόμβων – συσκευών το μόνο attribute είναι το 'node_type' όπου είναι ο τύπος της συσκευής που ορίζουμε. Αφού οριστούν όλες οι συσκευές σειρά έχουν οι ακμές. Πριν όμως οριστούν οι ακμές, τοποθετούμε τον χαρακτήρα '#' όπου αποτελεί διαχωριστικό της δήλωσης των κόμβων και των ακμών.

Οι συζεύξεις-ακμές μεταξύ των συσκευών ορίζονται ως:

```
pc1 sw1 ~ {  
    'link_ID' : '1' ,  
    'left_end' : {'if_id' : 'eth1'} ,  
    'right_end' : {'if_id' : 'eth1'},  
    'capacity' : '100',  
    'latency' : '5'};
```

Εικόνα 4.3.6. Ορισμός σύζευξης-ακμής μεταξύ δύο συσκευών στο αρχείο TGF της τοπολογίας.

Η δήλωση της σύζευξης μεταξύ δύο συσκευών αποτελείται από:

- Τη δήλωση της συσκευής-κόμβου που βρίσκεται στο αριστερό άκρο της σύζευξης (pc1)
- Τη δήλωση της συσκευής-κόμβου που βρίσκεται στο δεξί άκρο της σύζευξης (sw1)

Και στη συνέχεια από τα attributes της σύζευξης τα οποία είναι:

- **link_ID**: Το ID της του link-σύζευξης
- **left_end**: Το interface της συσκευής-κόμβου που βρίσκεται στο αριστερό άκρο της σύζευξης (pc1)
- **right_end**: Το interface της συσκευής-κόμβου που βρίσκεται στο δεξί άκρο της σύζευξης (sw1)

- **capacity:** Το bandwidth του συνδέσμου
- **latency:** Η καθυστέρηση διάδοσης του συνδέσμου

Ενδεικτικό Παράδειγμα Τοπολογίας

Παρακάτω θα δείξουμε ένα παράδειγμα ενός μη παραμετροποιημένου αρχείου τοπολογίας το οποίο περιέχει τον ορισμό ενός απλού δικτύου. Το παράδειγμα αυτό θα το χρησιμοποιήσουμε και στα επόμενα στάδια του στησίματος και επαλήθευσης μίας τοπολογίας δικτύου.

```

1 // Client PC definitions
2 pc1 ~ {'node_type' : 'Client_PC'};
3 pc2 ~ {'node_type' : 'Client_PC'};
4 pc3 ~ {'node_type' : 'Client_PC'};
5 pc4 ~ {'node_type' : 'Client_PC'};
6
7 // Switch definitions
8 sw1 ~ {'node_type' : 'Switch'};
9 sw2 ~ {'node_type' : 'Switch'};
10
11 // Router definitions
12 r1 ~ {'node_type' : 'Router'};
13 r2 ~ {'node_type' : 'Router'};
14
15 #
16
17 // --- Subnet 1 ---
18 pc1 sw1 ~ {
19     'link_ID' : '1' ,
20     'left_end' : {'if_id' : 'eth01'} ,
21     'right_end' : {'if_id' : 'eth01'},
22     'capacity' : '100',
23     'latency' : '5'};
24
25 pc2 sw1 ~ {
26     'link_ID' : '2' ,
27     'left_end' : {'if_id' : 'eth01'} ,
28     'right_end' : {'if_id' : 'eth01'},
29     'capacity' : '100',
30     'latency' : '5'};
31
32 sw1 r1 ~ {
33     'link_ID' : '3' ,
34     'left_end' : {'if_id' : 'eth02'} ,
35     'right_end' : {'if_id' : 'eth01'},
36     'capacity' : '100' ,
37     'latency' : '5'};
38
39
40 // --- Subnet 2 ---
41 pc3 sw2 ~ {
42     'link_ID' : '4' ,
43     'left_end' : {'if_id' : 'eth01'} ,
44     'right_end' : {'if_id' : 'eth01'},
45     'capacity' : '100' ,
46     'latency' : '5'};
47
48
49 pc4 sw2 ~ {
50     'link_ID' : '5' ,
51     'left_end' : {'if_id' : 'eth01'} ,
52     'right_end' : {'if_id' : 'eth01'},
53     'capacity' : '100' ,
54     'latency' : '5'};
55
56 sw2 r2 ~ {
57     'link_ID' : '6' ,
58     'left_end' : {'if_id' : 'eth02'} ,
59     'right_end' : {'if_id' : 'eth01'},
60     'capacity' : '100' ,
61     'latency' : '5'};
62
63 // --- Router connections ---
64 r1 r2 ~ {
65     'link_ID' : '7' ,
66     'left_end' : {'if_id' : 'eth02'} ,
67     'right_end' : {'if_id' : 'eth02'},
68     'capacity' : '100' ,
69     'latency' : '5'};
70

```

Εικόνα 4.3.7. Μη παραμετροποιημένο TGF αρχείο τοπολογίας.

4.3.2.2 Αρχικοποίηση των Configuration files

Αφού έχουμε ορίσει την τοπολογία, της συσκευές που περιλαμβάνονται σε αυτήν και τα interface της κάθε συσκευής, σειρά έχει να δημιουργήσουμε τα configuration files. Για κάθε μία συσκευή που έχει οριστεί στο αρχείο τοπολογίας, δημιουργούμε ένα configuration file στο οποίο γίνεται αρχικοποίηση του κάθε ενός interface που έχει η συσκευή. Επίσης, στα configuration files των Router – δρομολογητών, ορίζουμε και τους πίνακες δρομολόγησης του κάθε δρομολογητή. Για τα configuration files χρησιμοποιούμε, δημιουργούμε αρχεία μορφής YAML, και το πρότυπο που χρησιμοποιούμε είναι το Networking Config Version 2 του Netplan [20].

Configuration file format

Για τα configuration files, δημιουργούμε αρχεία μορφής YAML, και το πρότυπο που χρησιμοποιούμε είναι το Networking Config Version 2 Format του Netplan. Το netplan είναι ένα εργαλείο για την εύκολη παραμετροποίηση δικτύων σε συστήματα Linux. Για τη λειτουργία του, χρειάζεται η δημιουργία ενός YAML αρχείου για τα απαιτούμενα interfaces το οποίο ακολουθεί το πρότυπο του Networking Config Version 2 και αυτό θα αναλάβει να παράγει τα απαιτούμενα configuration στις συσκευές. Εμείς χρησιμοποιούμε αυτό το format και έχουμε κάνει μικρές προσθήκες, ώστε να ανταποκριθεί στις απαιτήσεις του εργαλείου μας.

Παρακάτω θα δείξουμε την μορφή που έχουν τα configuration files και για την παρουσίαση θα χρησιμοποιήσουμε κάποια από τα αρχεία των συσκευών που έχουμε κάνει αρχικοποίηση για το παραπάνω παράδειγμα.

Αρχικά, κάθε configuration file ξεκινάει με το πεδίο **'network'** στην αρχή. Στη συνέχεια βρίσκονται τα πεδία:

- **version:** Η έκδοση του Network config format
- **renderer:** Ο renderer που χρησιμοποιεί το εργαλείο του netplan
- **ethernets:** Όλα τα interface που περιέχει μία συσκευή

Configuration αρχείο μίας συσκευής Switch

Η παρακάτω εικόνα αποτελεί το Configuration file μίας συσκευής Switch. Αυτή η συσκευή περιέχει 2 interfaces το eth1 και το eth2 όπου:

- **macaddress:** Η διεύθυνση MAC του κάθε interface

```
1 network:
2   version: 2
3   renderer: networkd
4   ethernets:
5     eth01:
6       macaddress: 'F4-B7-E2-75-56-89'
7     eth02:
8       macaddress: 'F4-B7-E2-75-56-90'
9
```

Εικόνα 4.3.8. Αρχείο sw1_config.yml της συσκευής sw1 του Ενδεικτικού Παραδείγματος Τοπολογίας

Configuration αρχείο μίας συσκευής Client PC

Η παρακάτω εικόνα αποτελεί το Configuration file μίας συσκευής Client PC. Αυτή η συσκευή περιέχει 1 interface, το eth1 όπου:

- **macaddress:** Η διεύθυνση MAC του interface
- **addresses:** Η διεύθυνση IP του interface ακολουθούμενη από την μάσκα υπό-δίκτου στο οποίο βρίσκεται
- **gateway4:** Η gateway του interface
- **nameservers**
 - o **addresses:** Μία λίστα με τους DNS servers του interface

```
1 network:
2   version: 2
3   renderer: networkd
4   ethernets:
5     eth01:
6       macaddress: 'F4-B7-E2-75-56-81'
7       addresses: '21.2.1.10/24'
8       gateway4: '21.2.1.1'
9       nameservers:
10         addresses: [8.8.8.8]
11
```

Εικόνα 4.3.9. Αρχείο pc1_config.yml της συσκευής pc1 του Ενδεικτικού Παραδείγματος Τοπολογίας

Configuration αρχείο μίας συσκευής Router

Η παρακάτω εικόνα αποτελεί το Configuration file μίας συσκευής Router. Αυτή η συσκευή περιέχει 3 interface, το eth1, eth2 και rt.

Τα πεδία των interfaces eth1 και eth2:

- **macaddress**: : Η διεύθυνση MAC του interface
- **addresses**: : Η διεύθυνση IP του interface ακολουθούμενη από την μάσκα υπό-δίκτυου στο οποίο βρίσκεται
- **nat**
 - o **status**: Επιλογή για το αν ο δρομολογητής εκτελεί NAT. Δυνατές τιμές που μπορεί να πάρει: enable || disabled
 - o **public_ip**: Η public IP του υπό-δίκτυου σε περίπτωση όπου εκτελείται NAT. Αν όχι, default τιμή 0.0.0.0
- **gateway4**: Η gateway του interface: Ρυθμισμένη στο 0.0.0.0 καθώς ορίζουμε default διαδρομή στον πίνακα δρομολόγησης
- **nameservers**
 - o **addresses**: Λίστα με τους DNS servers του interface

Το interface rt μέσα στο configuration file ενός δρομολογητή περιέχει τον πίνακα δρομολόγησης του. Τα πεδία αυτού είναι:

- **routes**
 - o **to**: Η λίστα με όλους τους προορισμούς-εγγραφές του πίνακα δρομολόγησης. Ένας προορισμός είναι της μορφής: x.x.x.x./mask όπου x.x.x.x. είναι η διεύθυνση του υπό-δίκτυου και mask η μάσκα του υπό-δίκτυου.
 - o **via**: Το επόμενο άλμα που πρέπει να ακολουθήσει ένα πακέτο μέσα στο δίκτυο για να φτάσει στον προορισμό

```

1  network:
2    version: 2
3    renderer: networkd
4    ethernets:
5      eth01:
6        macaddress: 'F4-B7-E2-75-56-85'
7        addresses: '21.2.1.1/24'
8        nat:
9          status: 'disabled'
10         public_ip: '0.0.0.0'
11         gateway4: '0.0.0.0'
12         nameservers:
13           addresses: [8.8.8.8]
14
15      eth02:
16        macaddress: 'F4-B7-E2-75-56-86'
17        addresses: '21.2.3.1/24'
18        public_ip: '19.2.0.1/24'
19        nat:
20          status: 'disabled'
21          public_ip: '0.0.0.0'
22          gateway4: '0.0.0.0'
23          nameservers:
24            addresses: [8.8.8.8]
25
26      rt:
27        routes:
28          to: ["21.2.1.1/24", "21.2.2.0/24", "0.0.0.0"]
29          via: ["21.2.1.1", "21.2.3.2", "21.2.3.2"]
30

```

Εικόνα 4.3.10. Αρχείο `r1_config.yml` της συσκευής `r1` του Ενδεικτικού Παραδείγματος Τοπολογίας

Configuration αρχείο μίας συσκευής Hub

Η παρακάτω εικόνα αποτελεί το Configuration file μίας συσκευής Hub. Αυτή η συσκευή περιέχει 1 interface, το `eth`. Το Hub αποτελεί μία πολύ απλή περίπτωση συσκευής, όπου δεν πολυχρησιμοποιείται στα σημερινά δίκτυα, και στις περιπτώσεις που εξετάζει εργαλείο μας, δεν ελέγχει κάποια παράμετρο του, συνεπώς το hub δεν έχει πεδία σε αυτή τη φάση του εργαλείου.

```

1  network:
2    version: 2
3    renderer: networkd
4    ethernets:
5      eth01: ""
6

```

Εικόνα 4.3.11. Αρχείο παραμετροποίησης Hub.

Configuration αρχείο μίας συσκευής Server PC

Η παρακάτω εικόνα αποτελεί το Configuration file μίας συσκευής Server PC. Το interface ενός Server PC δεν διαφέρει από το interface ενός Client PC.

```
1  network:
2    version: 2
3    renderer: networkd
4    ethernets:
5      eth0:
6        macaddress: 'F4-B7-E2-75-56-01'
7        addresses: '56.1.1.10/24'
8        gateway4: '56.1.1.1'
9        nameservers:
10         addresses: [8.8.8.8]
11
```

Εικόνα 4.3.12. Αρχείο sr1_config.yml της Τοπολογίας 3 που θα δούμε παρακάτω στα Παραδείγματα Εκτέλεσης.

4.3.2.3 Επαλήθευση των configuration files

Αφού ορίσουμε όλα τα configuration files των συσκευών του δικτύου, σειρά έχει η επαλήθευσή τους. Η επαλήθευση αυτή γίνεται κάνοντας text analysis σε κάθε αρχείο. Συγκεκριμένα γίνονται οι εξής επαληθεύσεις:

- Επαλήθευση ύπαρξης configuration file για κάθε συσκευή: Γίνεται επαλήθευση ότι για κάθε μία συσκευή έχει όντως γίνει αρχικοποίηση ενός αρχείου.
- Επαλήθευση του format του configuration file: Γίνεται επαλήθευση του format το οποίο ακολουθούμε για το γράψιμο των configuration files.
- Επαλήθευση αρχικοποίησης των interfaces: Γίνεται επαλήθευση ότι κάθε ένα από τα interfaces που έχουν οριστεί στο TGF αρχείο όπου ορίσαμε την τοπολογία, έχει γίνει αρχικοποίηση στο configuration file της συσκευής
- Επαλήθευση τιμών πεδίων: Γίνεται επαλήθευση ότι κάθε τιμή στα πεδία των interfaces είναι έγκυρες τιμές

4.3.2.4 Παραμετροποίηση των συσκευών της τοπολογίας

Αφού ολοκληρωθεί το στήσιμο της τοπολογίας και η δημιουργία των configuration files κάθε συσκευής και η επαλήθευσή τους, σειρά έχει η παραμετροποίηση των συσκευών

με αυτά τα αρχεία. Στην διαδικασία της παραμετροποίησης, κάθε ένα interface συσκευής που έχουμε ορίσει στο μη παραμετροποιημένο αρχείο τοπολογίας, αντιστοιχίζεται με την αρχικοποίηση των τιμών του που βρίσκεται στο configuration file της συσκευής. Από αυτήν την παραμετροποίηση, προκύπτει το Παραμετροποιημένο Αρχείο τοπολογίας. Το Παραμετροποιημένο Αρχείο Τοπολογίας, είναι ένα αρχείο Trivial Graph Format (TGF) που περιλαμβάνει τις τιμές του κάθε interface.

Το Παραμετροποιημένο Αρχείο Τοπολογίας που προκύπτει από το Ενδεικτικό Παράδειγμα Τοπολογίας που ξεκινήσαμε παραπάνω είναι το εξής:

```

1  pc1 ~ { 'node_type': 'Client_PC';
2  pc2 ~ { 'node_type': 'Client_PC';
3  pc3 ~ { 'node_type': 'Client_PC';
4  pc4 ~ { 'node_type': 'Client_PC';
5  sw1 ~ { 'node_type': 'Switch';
6  sw2 ~ { 'node_type': 'Switch';
7  r1 ~ {
8    'node_type': 'Router',
9    'routing_table': {
10     'dest': ['21.2.1.1', '21.2.2.0', '0.0.0.0'],
11     'mask': ['255.255.255.0', '255.255.255.0', '255.255.255.255'],
12     'next_hop': ['21.2.1.1', '21.2.3.2', '21.2.3.2']
13   }
14 };
15 r2 ~ {
16   'node_type': 'Router',
17   'routing_table': {
18     'dest': ['21.2.2.1', '21.2.1.1', '0.0.0.0'],
19     'mask': ['255.255.255.0', '255.255.255.0', '255.255.255.255'],
20     'next_hop': ['21.2.2.1', '21.2.3.1', '21.2.3.1']
21   }
22 };
23 #
24
25 pc1 sw1 ~ {
26   'link_ID': '1',
27   'left_end': {'if_id': 'eth1', 'mac': 'F4-B7-E2-75-56-81', 'ip': '21.2.1.10', 'mask': '255.255.255.0', 'gateway': '21.2.1.1', 'dns': '8.8.8.8'},
28   'right_end': {'if_id': 'eth1', 'mac': 'F4-B7-E2-75-56-80'},
29   'capacity': '100',
30   'latency': '5'
31 };
32
33 pc2 sw1 ~ {
34   'link_ID': '2',
35   'left_end': {'if_id': 'eth1', 'mac': 'F4-B7-E2-75-56-82', 'ip': '21.2.1.12', 'mask': '255.255.255.0', 'gateway': '21.2.1.1', 'dns': '8.8.8.8'},
36   'right_end': {'if_id': 'eth1', 'mac': 'F4-B7-E2-75-56-80'},
37   'capacity': '100',
38   'latency': '5'
39 };
40
41 sw1 r1 ~ {
42   'link_ID': '3',
43   'left_end': {'if_id': 'eth2', 'mac': 'F4-B7-E2-75-56-00'},
44   'right_end': {'if_id': 'eth1', 'mac': 'F4-B7-E2-75-56-85', 'ip': '21.2.1.1', 'mask': '255.255.255.0', 'gateway': '0.0.0.0', 'dns': '8.8.8.8', 'nat': 'disabled', 'public_ip': '0.0.0.0'},
45   'capacity': '100',
46   'latency': '5'
47 };
48
49 pc3 sw2 ~ {
50   'link_ID': '4',
51   'left_end': {'if_id': 'eth1', 'mac': 'F4-B7-E2-75-56-83', 'ip': '21.2.2.10', 'mask': '255.255.255.0', 'gateway': '21.2.2.1', 'dns': '8.8.8.8'},
52   'right_end': {'if_id': 'eth1', 'mac': 'F4-B7-E2-75-56-91'},
53   'capacity': '100',
54   'latency': '5'
55 };
56
57 pc4 sw2 ~ {
58   'link_ID': '5',
59   'left_end': {'if_id': 'eth1', 'mac': 'F4-B7-E2-75-56-84', 'ip': '21.2.2.12', 'mask': '255.255.255.0', 'gateway': '21.2.2.1', 'dns': '8.8.8.8'},
60   'right_end': {'if_id': 'eth1', 'mac': 'F4-B7-E2-75-56-91'},
61   'capacity': '100',
62   'latency': '5'
63 };
64
65 sw2 r2 ~ {
66   'link_ID': '6',
67   'left_end': {'if_id': 'eth2', 'mac': 'F4-B7-E2-75-56-92'},
68   'right_end': {'if_id': 'eth1', 'mac': 'F4-B7-E2-75-56-87', 'ip': '21.2.2.1', 'mask': '255.255.255.0', 'gateway': '0.0.0.0', 'dns': '8.8.8.8', 'nat': 'disabled', 'public_ip': '0.0.0.0'},
69   'capacity': '100',
70   'latency': '5'
71 };
72

```

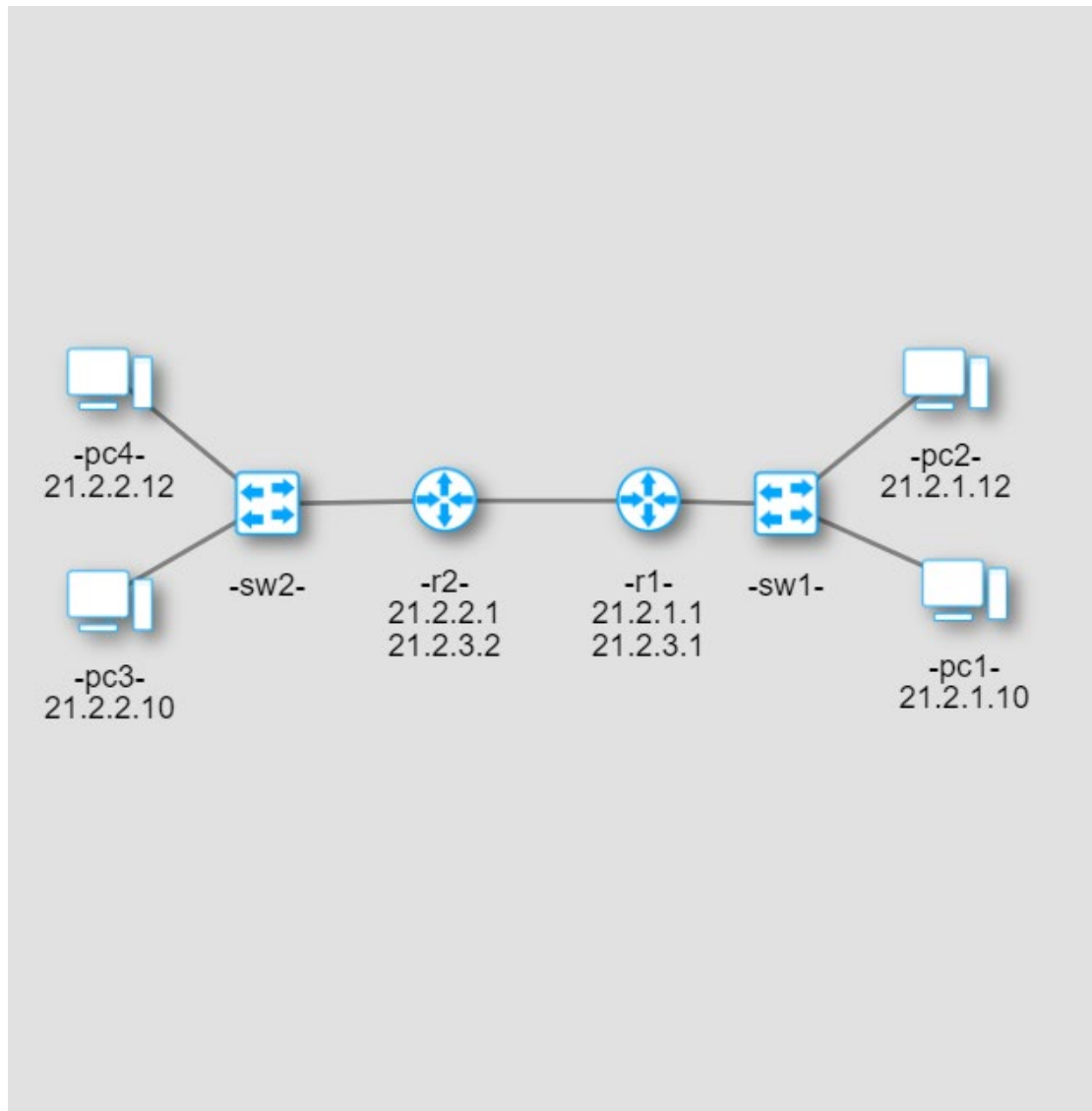
Εικόνα 4.3.13. Παραμετροποιημένο αρχείο τοπολογίας Ενδεικτικού Παραδείγματος

Μοντελοποίηση του δικτύου.

Αφού ολοκληρωθεί η παραμετροποίηση του Αρχείου Τοπολογίας, το πρόγραμμα μας κάνει Parsing το TGF Παραμετροποιημένο Αρχείο Τοπολογίας και ξεκινά την μοντελοποίηση του σε ένα γράφημα. Σε αυτό το γράφημα εκτελούνται queries όπου χωρίζει το δίκτυο σε υπό-δίκτυα. Στη συνέχεια χρησιμοποιείται στην διαδικασία της οπτικοποίησης και της επαλήθευσης του δικτύου.

4.3.2.5 Οπτικοποίηση

Αφού στηθεί και η μοντελοποίηση του δικτύου σειρά έχει η οπτικοποίηση του. Η οπτικοποίηση του δικτύου δημιουργεί από το γράφημα ένα αρχείο μορφής HTML το οποίο ανοίγει μέσω browser. Κάθε ένας κόμβος από τους παρακάτω μπορεί να γίνει drag and drop οπουδήποτε στο παράθυρο.



Εικόνα 4.3.14. Οπτικοποίηση Ενδεικτικού Παραδείγματος Τοπολογίας

Χαρακτηριστικά

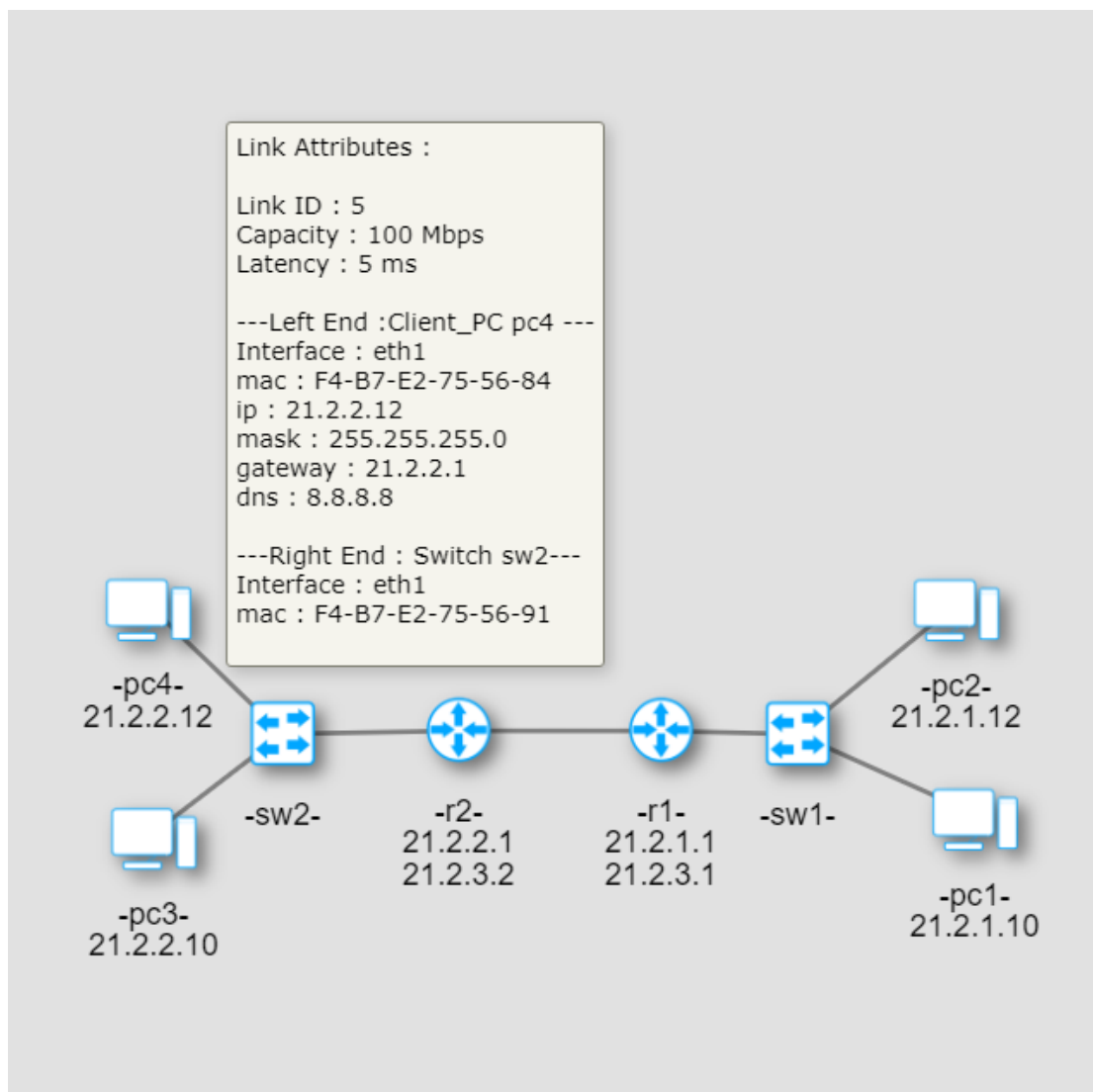
Κάτω από κάθε κόμβο βρίσκεται

- Το ID της συσκευής
- Οι IP διευθύνσεις των interface της συσκευής (εάν έχει, πχ τα Switch δεν έχουν IP).

Φέρνοντας τον κέρσορα πάνω σε μία ακμή εμφανίζονται τα attributes της σύζευξης.

Αυτά είναι:

- Το link ID της σύζευξης
- Το bandwidth της σύζευξης
- Η καθυστέρηση διάδοσης της σύζευξης
- Το αριστερό άκρο-συσσκευή της σύζευξης και τα attributes του interface της συσκευής που παίρνουν μέρος στη σύζευξη
- Το δεξί άκρο-συσσκευή της σύζευξης και τα attributes του interface της συσκευής που παίρνουν μέρος στη σύζευξη



Εικόνα 4.3.15. Οπτικοποίηση Ενδεικτικού Παραδείγματος Τοπολογίας

4.3.2.6 Επαλήθευση δικτύου

Τέλος, αφού ολοκληρωθούν όλες οι παραπάνω διαδικασίες, έχουν επαληθευτεί τα configuration files κάθε συσκευής και έχουν οριστεί όλες οι συσκευές και έχουν επιτυχώς παραμετροποιηθεί, ξεκινά η διαδικασία επαλήθευσης του δικτύου. Οι έλεγχοι που πραγματοποιούνται κατά την διαδικασία επαλήθευσης δικτύου είναι:

- **Επαλήθευση μοναδικότητας διευθύνσεων MAC**
- **Επαλήθευση μοναδικότητας IP υπό-δικτύων**
- **Επαλήθευση ορθά ρυθμισμένης διεύθυνσης gateway**
- **Επαλήθευση ορθά ρυθμισμένης διεύθυνσης IP**
- **Επαλήθευση μοναδικότητας διεύθυνσης IP**
- **Επαλήθευση πινάκων δρομολόγησης**

4.4 Εργαλεία που χρησιμοποιήθηκαν

Έχοντας παρουσιάσει τις δυνατότητες του εργαλείου και στη συνέχεια αναδείξαμε τα χαρακτηριστικά, τη λογική του και την αρχιτεκτονική του, θα παρουσιάσουμε τα σημαντικότερα εργαλεία που χρησιμοποιήθηκαν για το χτίσιμο αυτού του εργαλείου, σε τι γλώσσα αναπτύχθηκε, ποιες βιβλιοθήκες εκμεταλλευτήκαμε κλπ.

Γλώσσα προγραμματισμού:

- Python 3.8.8

Βιβλιοθήκες-Modules-Packages

- NetworkX
- pyvis
- netaddr
- Pandas

NetworkX

Το NetworkX είναι ένα πακέτο-package της python για τη δημιουργία, επεξεργασία και μελέτη της δομής, της δυναμικής και της λειτουργικότητας πολύπλοκων δικτύων –

γραφημάτων. Μας βοήθησε στο να μετατρέψουμε το γράφημα που ορίσαμε στο αρχείο TGF (Trivial Graph Format) σε ένα δίκτυο με δομή το οποίο επεξεργαστήκαμε και εκτελέσαμε queries πάνω σε αυτό. Περισσότερες πληροφορίες για το NetworkX στην αναφορά [21].

Pyvis

Το pyvis είναι ένα open-source module της python το οποίο δίνει την δυνατότητα δημιουργίας γραφημάτων, οπτικοποίησης τους και δυναμικής αλληλεπίδρασης του χρήστη με αυτά. Μας βοήθησε με την οπτικοποίηση του δικτύου, παίρνοντας ως είσοδο το γράφημα που είχαμε δημιουργήσει με τη βοήθεια του πακέτου NetworkX. Ως έξοδο μας έδωσε ένα αρχείο Html, το οποίο με τη βοήθεια της βιβλιοθήκης visjs της javascript οπτικοποιεί το δίκτυό μας. Περισσότερες πληροφορίες για το pyvis στην αναφορά [22].

Netaddr

Το netaddr είναι μία ανεξάρτητη συστήματος βιβλιοθήκη χειρισμού διευθύνσεων δικτύου. Μας βοήθησε με τον χειρισμό των IP διευθύνσεων, διευθύνσεων υπό-δικτύων και MAC διευθύνσεις των συσκευών.

Pandas

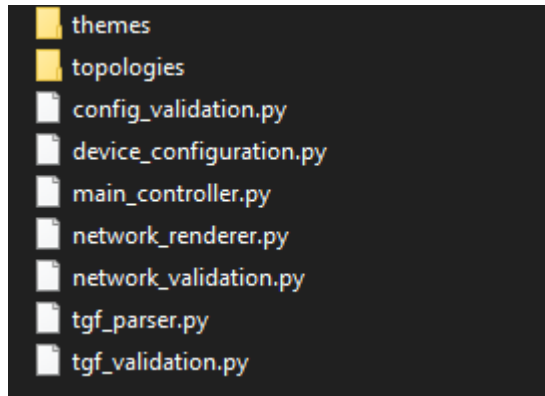
Το pandas είναι μία βιβλιοθήκη της Python με σκοπό τον χειρισμό και την ανάλυση δεδομένων. Μας βοήθησε στην δόμηση της πληροφορίας μέσα στα υπό-δίκτυα και στο δίκτυο ολόκληρο. Περισσότερες πληροφορίες για το pandas στην αναφορά [23].

4.5 Αρχιτεκτονική του λογισμικού

Σε αυτήν την ενότητα θα παρουσιάσουμε την δομή του λογισμικού, τα αρχεία που το αποτελούν, το τι κάνουν αυτά τα αρχεία και τους φακέλους.

Δομή project

Το project αναπτύχθηκε όπως αναφέραμε σε python 3.8 και αποτελείται από 7 αρχεία Python που βρίσκονται στο βασικό directory και 2 directories.



Αρχεία

main_controller.py: Ο main controller που ελέγχει τη ροή και ξεκινάει την διαδικασία επαλήθευσης.

tgf_parser.py: Ο custom parser του Trivial Graph Format (TGF)

tgf_validation.py: Αρχείο υπεύθυνο για την επαλήθευση του μη-παραμετροποιημένου TGF

config_validation.py: Αρχείο υπεύθυνο για τις διαδικασίες επαλήθευσης των configuration files των συσκευών.

network_interface.py: Αρχείο που περιέχει τις κλάσεις που υλοποιούν τα interfaces των συσκευών δικτύου.

device_configuration.py: Αρχείο υπεύθυνο για την παραμετροποίηση των συσκευών

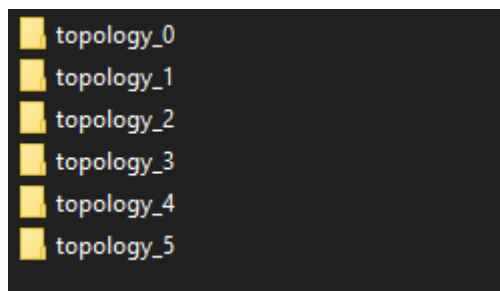
network_renderer.py: Αρχείο υπεύθυνο για την οπτικοποίηση του δικτύου

network_validation.py: Αρχείο υπεύθυνο που περιέχει τις διαδικασίες επαλήθευσης του δικτύου και των υπό-δικτύων του.

Directories

Topologies

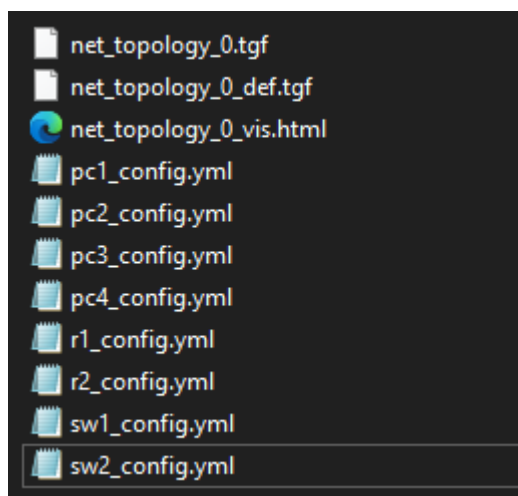
Φάκελος που περιέχει directories με την κάθε μία διαφορετική τοπολογία δικτύου



topology_0:

Φάκελος που περιέχει όλα τα αρχεία απαραίτητα για την τοπολογία 0. Αυτά είναι:

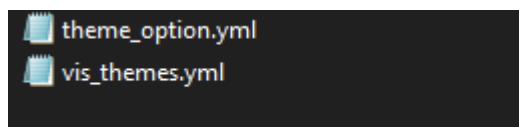
- *net_topology_0_def.tgf*: Μη παραμετροποιημένο αρχείο τοπολογίας
- *net_topology_0.tgf*: Παραμετροποιημένο αρχείο τοπολογίας
- *net_topology_0_vis.html*: Οπτικοποίηση τοπολογίας
- *x_config.yml*: Configuration file κάθε συσκευής, όπου x το ID της συσκευής



Themes

Φάκελος όπου περιέχει αρχεία για τα διαφορετικά θέματα οπτικοποίησης

- *theme_option.yml*: Αρχείο yml όπου περιέχει την επιλογή του θέματος οπτικοποίησης.
- *vis_themes.yml*: Αρχείο yml που περιέχει τα διαφορετικά θέματα οπτικοποίησης.



4.6 Επεκτασιμότητα

Όπως είπαμε στην εισαγωγή αυτού του κεφαλαίου, το δικό μας εργαλείο επαλήθευσης δικτύου, αποτελεί μία εισαγωγική ιδέα για ένα ολοκληρωμένο εργαλείο όπου σκοπός είναι η προσομοίωση μίας τοπολογίας δικτύου. Οι έλεγχοι που πραγματοποιούνται, ανήκουν σε ένα εύρος βασικών ελέγχων επαλήθευσης κάποιων παραμέτρων και λειτουργιών των δικτύων. Πηγαίνοντας το εργαλείο ένα βήμα πιο πέρα, πιο κοντά στα πρότυπα των εργαλείων που παρουσιάσαμε παραπάνω, χρειάζονται επεκτάσεις και ανασκευάσεις που θα το φέρουν στο επίπεδο που απαιτείται ώστε να γίνει ένας ορθός προσομοιωτής-simulator. Αυτές οι αλλαγές χωρίζονται σε δύο επίπεδα.

Το πρώτο επίπεδο επέκτασης του εργαλείου, έχει να κάνει με την ευκολία χρήσης του εργαλείου, την οπτικοποίηση του δικτύου και την αλληλεπίδραση με τον χρήστη. Η ιδέα είναι η ύπαρξη ενός γραφικού περιβάλλοντος GUI όπου ο χρήστης θα αλληλοεπιδρά με το δίκτυο προσθέτοντας συσκευές με το χέρι (drag and drop) και με τον ίδιο τρόπο θα δημιουργεί συζεύξεις μεταξύ των συσκευών. Πλέον, δεν θα χρειάζεται ο ορισμός των συσκευών και των συζεύξεων να είναι χειροκίνητος, κάτι το οποίο είναι ιδιαίτερα χρονοβόρο και μη ευέλικτο. Τέλος, πάνω σε αυτή την οπτικοποίηση, στόχος είναι να προστεθούν και animation τα οποία θα μπορούν να οπτικοποιούν την κίνηση των δικτύων.

Το δεύτερο επίπεδο επέκτασης έχει να κάνει με τα υποστηριζόμενα πρωτόκολλα δικτύου και τις υποστηριζόμενες συσκευές – κόμβους. Η ιδέα είναι το εργαλείο να

υποστηρίζει τη χρήση πραγματικών πρωτοκόλλων που χρησιμοποιούνται στα δίκτυα και στις συσκευές. Μαζί με την υποστήριξη πρωτοκόλλων σημαντικό είναι το εργαλείο να μπορεί να υποστηρίζει χρήση εικονικών συσκευών (Virtual machines) πραγματικών συσκευών, Routers, Switches τα οποία χρησιμοποιούνται σήμερα στα πραγματικά δίκτυα. Ο συνδυασμός αυτών των ενεργειών είναι απαραίτητος έτσι ώστε το εργαλείο μας να γίνει ένα ολοκληρωμένο λογισμικό που προσομοιώνει πραγματικά δίκτυα.

Κεφάλαιο 5. Παραδείγματα εκτέλεσης

5.1 Εισαγωγή

Για την επιβεβαίωση της λειτουργίας του δικού μας εργαλείου επαλήθευσης δικτύου, έχουμε στήσει διάφορες τοπολογίες δικτύου. Κάθε μία από αυτές είναι διαφορετικού μεγέθους και πολυπλοκότητας και είναι σχεδιασμένες με σκοπό να καλυφθούν όσο το δυνατόν περισσότερες περιπτώσεις λάθους μπορούν να προκύψουν. Για την κάθε μία από τις διαφορετικές τοπολογίες, έχουμε τρέξει πολλαπλά παραδείγματα με διαφορετικές παραμέτρους. Σκοπός αυτών των εκτελέσεων είναι να δείξουμε πως συμπεριφέρεται το εργαλείο σε περιπτώσεις όπου έχουμε κάνει αρχικοποίηση τις συσκευές της τοπολογίας με ορθές παραμέτρους αλλά και το πως συμπεριφέρεται σε περιπτώσεις όπου έχουμε κάνει αρχικοποίηση με μη ορθές παραμέτρους και να επαληθεύσουμε την ορθή λειτουργία ή όχι του εργαλείου βάση της εξόδου του προγράμματος.

Παρακάτω, για κάθε μία τοπολογία, παρουσιάζουμε την οπτικοποίηση της και την περιγραφή της όπου έχουν προκύψει από την πρώτη ορθή εκτέλεση της με τα ορθά configurations. Στην συνέχεια παρουσιάζουμε κάθε μία από τις διαφορετικές εκτελέσεις όπου αυτό περιλαμβάνει την περιγραφή της εκτέλεσης, την έξοδο του προγράμματος και τις παρατηρήσεις – σχόλια που προκύπτουν από την έξοδο του προγράμματος.

Συγκεκριμένα:

Παρακάτω παρουσιάζουμε 5 διαφορετικές τοπολογίες και σε αυτές τρέχουμε ένα πλήθος διαφορετικών εκτελέσεων, αυτές είναι:

- Τοπολογία 0, όπου τρέχουμε 10 εκτελέσεις

- Τοπολογία 1, όπου τρέχουμε 8 εκτελέσεις
- Τοπολογία 2, όπου τρέχουμε 8 εκτελέσεις
- Τοπολογία 3, όπου τρέχουμε 7 εκτελέσεις
- Τοπολογία 4, όπου τρέχουμε 12 εκτελέσεις

Στις εκτελέσεις της τοπολογίας 0, σκοπός είναι να δούμε την λειτουργία της διαδικασίας επαλήθευσης των configuration files των συσκευών. Η διαδικασία αυτή ελέγχει:

- Αν κάθε interface που έχει οριστεί στο TGF αρχείο τοπολογίας πρέπει να έχει γίνει αρχικοποίηση και στα configurations της συσκευής
- Αν το configuration file πρέπει να ακολουθεί το Network Config format Version 2 που ακολουθούμε
- Αν κάθε τιμή στα πεδία μίας συσκευής, είναι έγκυρη τιμή.

Σε περίπτωση όπου εντοπιστεί κάποιο λάθος , τερματίζει το πρόγραμμα, δίχως να ολοκληρώσει την αρχικοποίηση των συσκευών και κάνει μοντελοποίηση και οπτικοποίηση του δικτύου.

Στις τοπολογίες Τοπολογία 1 , Τοπολογία 2, Τοπολογία 3, Τοπολογία 4, σκοπός είναι να δούμε την λειτουργία επαλήθευσης του δικτύου, όπου ελέγχει το δίκτυο ως σύνολο. Αφού λοιπόν ολοκληρώνεται η επαλήθευση των configuration files, στην συνέχεια ελέγχονται τα εξής:

- Επαλήθευση μοναδικότητας διευθύνσεων MAC
- Επαλήθευση μοναδικότητας IP υπό-δικτύων
- Επαλήθευση ορθά ρυθμισμένης διεύθυνσης gateway
- Επαλήθευση ορθά ρυθμισμένης διεύθυνσης IP
- Επαλήθευση μοναδικότητας διεύθυνσης IP
- Επαλήθευση πινάκων δρομολόγησης

Σε περίπτωση όπου εντοπιστεί κάποιο λάθος, τερματίζει ανεπιτυχώς η διαδικασία επαλήθευσης δικτύου, αφού εμφανιστεί το κατάλληλο μήνυμα σφάλματος

Σημείωση

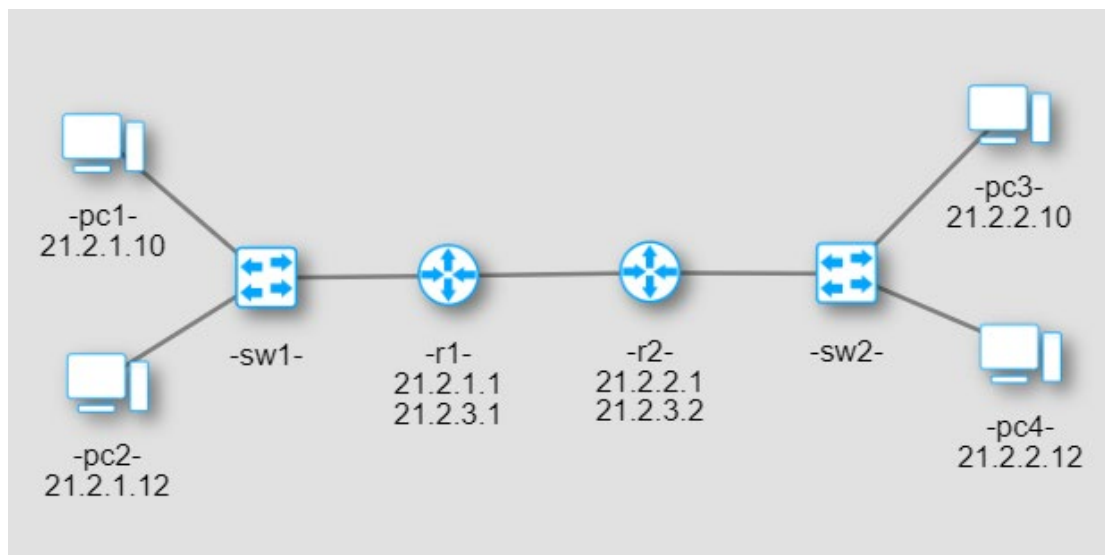
Παρακάτω στην περιγραφή της κάθε τοπολογίας, χαρακτηρίζουμε τα υπό-δίκτυα ως «φυσικά» υπό-δίκτυα και υπό-δίκτυα «μεταξύ δρομολογητών». Φυσικά χαρακτηρίζουμε τα υπό-δίκτυα τα οποία αποτελούνται από υπολογιστές, servers και

switches. Μεταξύ δρομολογητών ορίζουμε τα υπό-δίκτυα στα οποία 2 δρομολογητές είναι συνδεδεμένοι μεταξύ τους.

5.2 Τοπολογία 0

5.2.1 Οπτικοποίηση τοπολογίας

Παρακάτω βρίσκεται η οπτικοποίηση της τοπολογίας στην οποία θα τρέξουμε εκτελέσεις με διάφορα configurations. Η συγκεκριμένη οπτικοποίηση είναι μιας πρώτης εκτέλεσης όπου έχουμε κάνει αρχικοποίηση με ορθά configurations.



Εικόνα 5.2.1. Οπτικοποίηση τοπολογίας 0.

5.2.2 Περιγραφή τοπολογίας

Σε αυτή την τοπολογία, οι εκτελέσεις που θα τρέξουμε έχουν σκοπό να επαληθεύσουν την ορθότητα των configuration files κάνοντας text analysis. Οι έλεγχοι αυτοί δηλαδή επαληθεύουν την ορθότητα του format των configuration files και την ορθότητα των τιμών στα πεδία των configuration files.

Η παραπάνω τοπολογία αποτελείται από 3 υπό-δίκτυα, 2 φυσικά και 2 μεταξύ δρομολογητών. Αυτά είναι:

- Στο router r1: 21.2.1.0/24
- Στο router r2: 21.2.2.0/24

Και μεταξύ των δρομολογητών

- Μεταξύ r1-r2: 21.2.3.0/24

5.2.3 Εκτελέσεις

Εκτέλεση 1

Περιγραφή εκτέλεσης

Η πρώτη εκτέλεση πάνω στην τοπολογία 1 έχει στηθεί με τα κατάλληλα configurations έτσι ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να επαληθεύσει την ορθότητα του format των configuration files κάθε συσκευής. Συγκεκριμένα:

- Δεν ορίζουμε πεδίο για MAC διεύθυνση στο Interface eth01 του pc1
- Δεν ορίζουμε πεδίο gateway στο interface eth01 του pc1

Έξοδος προγράμματος

```
----- Running network topology: 'topology_0' -----  
-Configuration file error at file: 'pc1_config.yml' :  
  -Attribute Error: {'macaddress', 'gateway4'} is missing from interface: eth01
```

Εικόνα 5.2.2 Έξοδος εκτέλεσης 1 στην τοπολογία 0.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι εντοπίστηκε σφάλμα στο configuration file της συσκευής pc1. Συγκεκριμένα εντοπίστηκε λάθος στο format του αρχείου καθώς δεν βρέθηκαν τα πεδία macaddress και gateway4 στο interface eth01. Καθώς εντοπίστηκαν τα λάθη, το πρόγραμμα τερματίζει και δεν συνεχίζει παρακάτω με την ολοκλήρωση της παραμετροποίησης των συσκευών και την επαλήθευση δικτύου.

Εκτέλεση 2

Περιγραφή εκτέλεσης

Η δεύτερη εκτέλεση πάνω στην τοπολογία 1 έχει στηθεί με τα κατάλληλα configurations έτσι ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να επαληθεύσει την αρχικοποίηση των interfaces κάθε συσκευής. Συγκεκριμένα:

- Δεν κάνουμε αρχικοποίηση των interfaces eth01 και eth02 του router r1

Έξοδος προγράμματος

```
----- Running network topology: 'topology_0' -----  
Device configuration Error at file: 'r1_config.yml':  
-Initialaization error: No device initialization for interface: {'eth02', 'eth01'}, at node: 'r1', was found
```

Εικόνα 5.2.3. Έξοδος εκτέλεσης 2, τοπολογία 0.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι εντοπίστηκε σφάλμα στο configuration file της συσκευής r1. Συγκεκριμένα εντοπίστηκε λάθος στα interfaces του αρχείου καθώς δεν βρέθηκαν αρχικοποιήσεις για τα interfaces eth01 και eth02 που ορίστηκαν στο αρχείο τοπολογίας TGF. Καθώς εντοπίστηκαν τα λάθη, το πρόγραμμα τερματίζει και δεν συνεχίζει παρακάτω με την ολοκλήρωση της παραμετροποίησης των συσκευών και την επαλήθευση δικτύου.

Εκτέλεση 3

Περιγραφή εκτέλεσης

Η ένατη εκτέλεση πάνω στην τοπολογία 1 έχει στηθεί με τα κατάλληλα configurations έτσι ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να επαληθεύσει την ορθότητα των τιμών των πεδίων που βρίσκονται στα interface μίας συσκευής. Συγκεκριμένα:

- Ορίζουμε μη έγκυρη διεύθυνση MAC στο interface του κόμβου pc2

Έξοδος προγράμματος

```
----- Running network topology: 'topology_0' -----  
-Configuration file error at file: 'pc2_config.yml' :  
  -Value error: FW-B7-E2-75-56-8W, at interface: eth01, is not a valid mac address
```

Εικόνα 5.2.4. Έξοδος εκτέλεσης 3, τοπολογία 0.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι εντοπίστηκε σφάλμα στο configuration file της συσκευής pc2. Συγκεκριμένα εντοπίστηκε λάθος στην τιμή του πεδίου macaddress του interface eth01 του αρχείου. Η συγκεκριμένη διεύθυνση MAC δεν αποτελεί έγκυρη MAC διεύθυνση. Καθώς εντοπίστηκαν τα λάθη, το πρόγραμμα τερματίζει και δεν συνεχίζει παρακάτω με την ολοκλήρωση της παραμετροποίησης των συσκευών και την επαλήθευση δικτύου.

Εκτέλεση 4

Περιγραφή εκτέλεσης

Η τέταρτη εκτέλεση πάνω στην τοπολογία 0 έχει στηθεί με τα κατάλληλα configurations έτσι ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να επαληθεύσει την ορθότητα των τιμών των πεδίων που βρίσκονται στα interface μίας συσκευής. Συγκεκριμένα:

- Ορίζουμε μη έγκυρη διεύθυνση IP στο interface του κόμβου pc2

Έξοδος προγράμματος

```
----- Running network topology: 'topology_0' -----  
-Configuration file error at file: 'pc2_config.yml' :  
  -Value error: 21.1.12, at interface: eth01 is not a valid ip address.  
    IP address consists of 4 address fields, decimal integer numbers in range 0-255
```

Εικόνα 5.2.5. Έξοδος εκτέλεσης 4, τοπολογία 0.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι εντοπίστηκε σφάλμα στο configuration file της συσκευής pc2. Συγκεκριμένα εντοπίστηκε λάθος στην τιμή του πεδίου της IP διεύθυνσης του interface eth01 του αρχείου. Η συγκεκριμένη διεύθυνση δεν αποτελεί έγκυρη IP διεύθυνση. Καθώς εντοπίστηκαν τα λάθη, το πρόγραμμα τερματίζει και δεν συνεχίζει παρακάτω με την ολοκλήρωση της παραμετροποίησης των συσκευών και την επαλήθευση δικτύου.

Εκτέλεση 5

Περιγραφή εκτέλεσης

Η πέμπτη εκτέλεση πάνω στην τοπολογία 1 έχει στηθεί με τα κατάλληλα configurations έτσι ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να επαληθεύσει την ορθότητα των τιμών των πεδίων που βρίσκονται στα interface μίας συσκευής. Συγκεκριμένα:

- Ορίζουμε μη έγκυρη μάσκα υπό-δικτύου στο interface του κόμβου pc2

Έξοδος προγράμματος

```
----- Running network topology: 'topology_0' -----  
-Configuration file error at file: 'pc2_config.yml' :  
  -Value error: /36, at interface: eth01 is not a valid network mask
```

Εικόνα 5.2.6. Έξοδος εκτέλεσης 5, τοπολογία 0.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι εντοπίστηκε σφάλμα στο configuration file της συσκευής pc2. Συγκεκριμένα εντοπίστηκε λάθος στην τιμή του πεδίου της μάσκας υπό-δικτύου διεύθυνσης του interface eth01 του αρχείου. Η συγκεκριμένη τιμή της μάσκας υπό-δικτύου δεν αποτελεί έγκυρη τιμή μάσκας υπό-δικτύου. Καθώς εντοπίστηκαν τα λάθη, το πρόγραμμα τερματίζει και δεν συνεχίζει παρακάτω με την ολοκλήρωση της παραμετροποίησης των συσκευών και την επαλήθευση δικτύου.

Εκτέλεση 6

Περιγραφή εκτέλεσης

Η έκτη εκτέλεση πάνω στην τοπολογία 1 έχει στηθεί με τα κατάλληλα configurations έτσι ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να επαληθεύσει την ορθότητα των τιμών των πεδίων που βρίσκονται στα interface μίας συσκευής. Συγκεκριμένα:

- Ορίζουμε μη έγκυρη τιμή gateway στο interface της συσκευής pc2

Έξοδος προγράμματος

```
----- Running network topology: 'topology_0' -----  
-Configuration file error at file: 'pc2_config.yml' :  
  
-Value error: 21.2.1.256, at interface: eth01 is not a valid gateway address.  
IP address fields must be decimal integer numbers in range 0-255
```

Εικόνα 5.2.7. Έξοδος εκτέλεσης 6, τοπολογία 0.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι εντοπίστηκε σφάλμα στο configuration file της συσκευής pc2. Συγκεκριμένα εντοπίστηκε λάθος στην τιμή του gateway του interface eth01 της συσκευής. Η συγκεκριμένη διεύθυνση δεν αποτελεί έγκυρη gateway διεύθυνση. Καθώς εντοπίστηκαν τα λάθη, το πρόγραμμα τερματίζει και δεν συνεχίζει παρακάτω με την ολοκλήρωση της παραμετροποίησης των συσκευών και την επαλήθευση δικτύου.

Εκτέλεση 7

Περιγραφή εκτέλεσης

Η έβδομη εκτέλεση πάνω στην τοπολογία 1 έχει στηθεί με τα κατάλληλα configurations έτσι ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να επαληθεύσει την ορθότητα των τιμών των πεδίων που βρίσκονται στα interface μίας συσκευής. Συγκεκριμένα:

- Ορίζουμε εσφαλμένο DNS στο interface του κόμβου pc2

Έξοδος προγράμματος

```
----- Running network topology: 'topology_0' -----  
-Configuration file error at file: 'pc2_config.yml' :  
-Value error: 8.8.8.08, at interface: eth01 is not a valid dns address.  
No leading zeros in IP address fields allowed!
```

Εικόνα 5.2.8. Έξοδος εκτέλεσης 7, τοπολογία 0.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι εντοπίστηκε σφάλμα στο configuration file της συσκευής pc2. Συγκεκριμένα εντοπίστηκε λάθος στην τιμή του DNS του interface eth01 της συσκευής. Η συγκεκριμένη διεύθυνση δεν αποτελεί έγκυρη DNS διεύθυνση. Καθώς εντοπίστηκαν τα λάθη, το πρόγραμμα τερματίζει και δεν συνεχίζει παρακάτω με την ολοκλήρωση της παραμετροποίησης των συσκευών και την επαλήθευση δικτύου.

Εκτέλεση 8

Περιγραφή εκτέλεσης

Η όγδοη εκτέλεση πάνω στην τοπολογία 1 έχει στηθεί με τα κατάλληλα configurations έτσι ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να επαληθεύσει την ορθότητα των τιμών των πεδίων που βρίσκονται στα interface μίας συσκευής. Συγκεκριμένα:

- Δεν ορίζουμε πίνακα δρομολόγησης στο router r1

Έξοδος προγράμματος

```
----- Running network topology: 'topology_0' -----  
-Configuration file error at file: 'r1_config.yml' :  
-Device Error: No routing table is defined in file.
```

Εικόνα 5.2.9. Έξοδος εκτέλεσης 8, τοπολογία 0.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι εντοπίστηκε σφάλμα στο configuration file της συσκευής r1. Συγκεκριμένα εντοπίστηκε λάθος στην δήλωση του πίνακα δρομολόγησης του αρχείου, καθώς δεν έχει δηλωθεί. Κάθε δρομολογητής πρέπει να έχει ορισμένο πίνακα δρομολόγησης στα configurations του. Καθώς εντοπίστηκαν τα λάθη, το πρόγραμμα τερματίζει και δεν συνεχίζει παρακάτω με την επαλήθευση δικτύου.

Εκτέλεση 9

Περιγραφή εκτέλεσης

Η ένατη εκτέλεση πάνω στην τοπολογία 1 έχει στηθεί με τα κατάλληλα configurations έτσι ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να επαληθεύσει την ορθότητα των τιμών των πεδίων που βρίσκονται στα interface μίας συσκευής. Συγκεκριμένα:

- Δεν ορίζουμε επόμενο άλμα για μία από τις διαδρομές του πίνακα δρομολόγησης του δρομολογητή r1

Έξοδος προγράμματος

```
----- Running network topology: 'topology_0' -----  
-Configuration file error at file: 'r1_config.yml' :  
  -Attribute Error: Entries are missing from routing table
```

Εικόνα 5.2.10. Έξοδος εκτέλεσης 9, τοπολογία 0.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι εντοπίστηκε σφάλμα στο configuration file της συσκευής r1. Συγκεκριμένα εντοπίστηκε λάθος στην δήλωση του πίνακα δρομολόγησης του αρχείου. Δεν έχει δηλωθεί κάποια ή κάποιες εγγραφές του πίνακα δρομολόγησης με αποτέλεσμα να μην είναι έγκυρος ο πίνακας. Καθώς εντοπίστηκαν τα λάθη, το πρόγραμμα τερματίζει και δεν συνεχίζει παρακάτω με την ολοκλήρωση της παραμετροποίησης των συσκευών και την επαλήθευση δικτύου.

Εκτέλεση 10

Περιγραφή εκτέλεσης

Η ένατη εκτέλεση πάνω στην τοπολογία 1 έχει στηθεί με τα κατάλληλα configurations έτσι ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να επαληθεύσει την ορθότητα των τιμών των πεδίων που βρίσκονται στα interface μίας συσκευής. Συγκεκριμένα:

- Ορίζουμε εσφαλμένες τιμές στις εγγραφές του πίνακα δρομολόγησης

Έξοδος προγράμματος

```
----- Running network topology: 'topology_0' -----  
-Configuration file error at file: 'r1_config.yml' :  
  
-Value Error: 21.2.2a.0/24 address at routing table -to- list is not a valid ip address.  
IP address consists of 4 address fields, decimal integer numbers in range 0-255
```

Εικόνα 5.2.11. Έξοδος εκτέλεσης 10, τοπολογία 0.

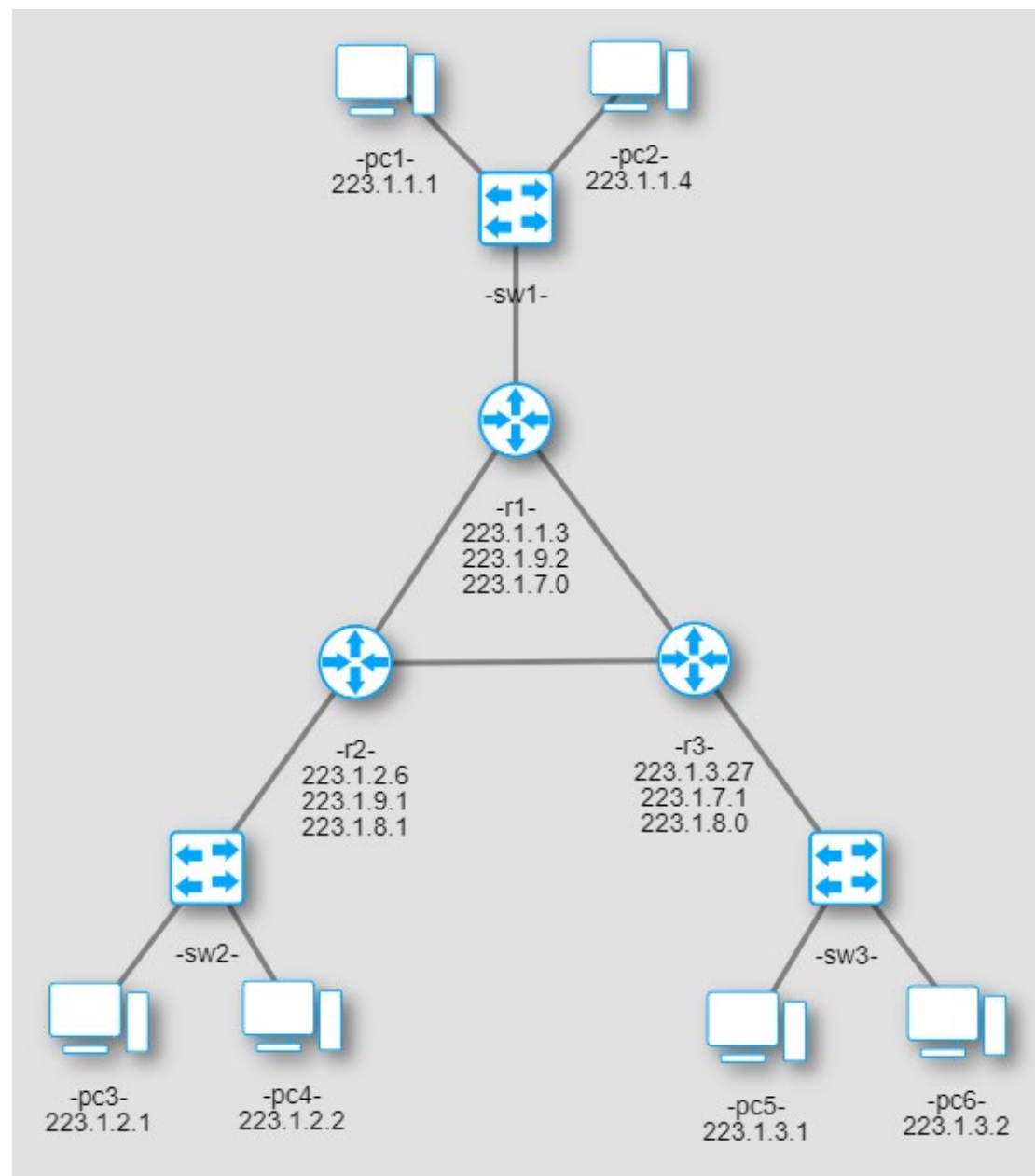
Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι εντοπίστηκε σφάλμα στο configuration file της συσκευής r1. Συγκεκριμένα εντοπίστηκε λάθος στην δήλωση του πίνακα δρομολόγησης του αρχείου. Κάποια από τις εγγραφές του πίνακα δρομολόγησης δεν είναι έγκυρη διεύθυνση IP. Καθώς εντοπίστηκαν τα λάθη, το πρόγραμμα τερματίζει και δεν συνεχίζει παρακάτω με την ολοκλήρωση της παραμετροποίησης των συσκευών και την επαλήθευση δικτύου.

5.3 Τοπολογία 1

5.3.1 Οπτικοποίηση τοπολογίας

Παρακάτω βρίσκεται η οπτικοποίηση της πρώτης τοπολογίας στην οποία θα τρέξουμε εκτελέσεις μια διάφορα configurations. Η συγκεκριμένη οπτικοποίηση είναι της πρώτης εκτέλεσης όπου έχουμε κάνει αρχικοποίηση με ορθά configurations.



Εικόνα 5.3.1. Οπτικοποίηση της τοπολογίας 1.

5.3.2 Περιγραφή Τοπολογίας

Το δίκτυο αποτελείται από 6 υπό-δίκτυα, τρία φυσικά υπό-δίκτυα και τρία μεταξύ των interface των δρομολογητών.

Αυτά είναι:

- Στο router r1: 223.1.1.0/24
- Στο router r2: 223.1.2.0/24
- Στο router r3: 223.1.3.0/24
-

Επίσης αποτελείται από 3 υπό-δίκτυα στα οποία είναι μεταξύ τους συνδεδεμένα οι δρομολογητές. Αυτά είναι:

- Μεταξύ router r1 - r2 : 223.1.9.0/24
- Μεταξύ router r1 - r3 : 223.1.7.0/24
- Μεταξύ router r2 - r3: 223.1.8.0/24

5.3.3 Εκτελέσεις

Εκτέλεση 1

Περιγραφή εκτέλεσης

Η πρώτη εκτέλεση πάνω στην τοπολογία 1 έχει στηθεί με τα κατάλληλα configurations έτσι ώστε να είναι ορθή.

Έξοδος προγράμματος

```
----- Running network topology: 'topology_1' -----  
  
Device configuration completed!  
  
Subnet validation completed!  
  
Routing table validation completed!  
  
Network validation completed!  
  
----- End of running: 'topology_1' -----
```

Εικόνα 5.3.2. Έξοδος εκτέλεσης 1, τοπολογία 1.

Παρατηρήσεις – σχόλια

Με το πέρας της εκτέλεσης του παραδείγματος, η έξοδος του προγράμματος δείχνει ότι όλες οι διαδικασίες αρχικοποίησης και επαλήθευσης ολοκληρώθηκαν επιτυχώς, συνεπώς τα configurations του δικτύου είναι σωστά αρχικοποιημένα όπως και προβλέψαμε και η επαλήθευση δικτύου ολοκληρώνεται επιτυχώς.

Εκτέλεση 2

Περιγραφή εκτέλεσης

Η δεύτερη εκτέλεση πάνω στην τοπολογία 1 έχει στηθεί με τα κατάλληλα configurations έτσι ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Τα λάθη αυτά γίνονται στην επαλήθευση των διευθύνσεων MAC των υπό-δικτύων, όπου ορίζουμε διπλότυπες MAC διευθύνσεις σε κάποια από τα υπό-δίκτυα, συγκεκριμένα:

- Ορίζουμε διπλότυπη MAC μεταξύ των κόμβων pc1 & pc2 στο υπό-δίκτυο 223.1.1.0/24 που βρίσκεται στο router r1
- Ορίζουμε διπλότυπη MAC μεταξύ των κόμβων pc5 & pc6 στο υπό-δίκτυο 223.1.3.0/24 που βρίσκεται στο router r3

Έξοδος προγράμματος

```
----- Running network topology: 'topology_1' -----  
  
Device configuration completed!  
  
-MAC Address Error: Found duplicate MAC addresses at subnet 223.1.1.0/24, host router r1:  
Node Interface      MAC_Address  
pc2    eth01 00-11-22-33-44-04  
pc1    eth01 00-11-22-33-44-04  
  
-MAC Address Error: Found duplicate MAC addresses at subnet 223.1.3.0/24, host router r3:  
Node Interface      MAC_Address  
pc6    eth01 00-11-22-33-44-56  
pc5    eth01 00-11-22-33-44-56  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.3.3. Έξοδος εκτέλεσης 2, τοπολογία 1.

Παρατηρήσεις – σχόλια

Με το πέρας της εκτέλεσης του παραδείγματος, βλέπουμε ότι το πρόγραμμα εντόπισε διπλότυπες διευθύνσεις MAC μεταξύ των pc1, pc2 στο υπό-δίκτυο 223.1.1.0/24 που βρίσκεται στο router r1 και μεταξύ των pc5, pc6 στο υπό-δίκτυο 223.1.3.0/24 που βρίσκεται στο router r3 όπως και αναμέναμε. Μετά των εντοπισμό των λαθών, εμφανίζεται κατάλληλο μήνυμα και τερματίζει ανεπιτυχώς η επαλήθευση του δικτύου.

Εκτέλεση 3

Περιγραφή εκτέλεσης

Η τρίτη εκτέλεση πάνω στην τοπολογία 1 έχει στηθεί με τα κατάλληλα configurations έτσι ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Τα λάθη αυτά γίνονται στην επαλήθευση των IP υπό-δικτύων, όπου ορίζουμε διπλότυπα IP υπό-δίκτυα, συγκεκριμένα

- Ορίζουμε διπλότυπα φυσικά υπό-δίκτυα μεταξύ των router r1 και r2 με διεύθυνση 223.1.1.0/24

Έξοδος προγράμματος

```

----- Running network topology: 'topology_1' -----

Device configuration completed!

Subnet error: Found duplicate subnets:
Host_Router_ID      Address
r1 223.1.1.0/24
r2 223.1.1.0/24

Found network configuration errors. Network validation has ended!

```

Εικόνα 5.3.4. Έξοδος εκτέλεσης 3, τοπολογία 1.

Παρατηρήσεις – σχόλια

Με το πέρας της εκτέλεσης του παραδείγματος, η έξοδος του προγράμματος δείχνει ότι το πρόγραμμα ορθώς εντόπισε τα δύο διπλότυπα IP υπό-δίκτυα. Αυτό είναι το 223.1.1.0/24 όπου βρίσκονται στους κόμβους r1 & r3. Μετά τον εντοπισμό η επαλήθευση δικτύου τερματίζει

Εκτέλεση 4

Περιγραφή εκτέλεσης

Η τέταρτη εκτέλεση πάνω στην τοπολογία 1 έχει στηθεί με μη ορθά configuration με τον συνδυασμό των δύο παραπάνω εκτελέσεων (εκτέλεση 2 και εκτέλεση 3) με σκοπό να δούμε την συμπεριφορά του προγράμματος όταν υπάρχουν δύο διαφορετικών ειδών λάθη στην τοπολογία και το αν η σειρά εκτέλεσης των επαληθεύσεων είναι έτσι όπως την έχουμε ορίσει.

Συγκεκριμένα:

- Ορίζουμε διπλότυπη MAC μεταξύ των κόμβων pc1 & pc2 στο υπό-δίκτυο 223.1.1.0/24 που βρίσκεται στο router r1
- Ορίζουμε διπλότυπη MAC μεταξύ των κόμβων pc5 & pc6 στο υπό-δίκτυο 223.1.3.0/24 που βρίσκεται στο router r3
- Ορίζουμε διπλότυπα φυσικά υπό-δίκτυα μεταξύ των router r1 και r2 με διεύθυνση 223.1.1.0/24

Έξοδος προγράμματος

```
----- Running network topology: 'topology_1' -----  
  
Device configuration completed!  
  
Subnet error: Found duplicate subnets:  
  Host_Router_ID      Address  
0                   r1  223.1.1.0/24  
1                   r2  223.1.1.0/24  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.3.5. Έξοδος εκτέλεσης 4, τοπολογία 1.

Παρατηρήσεις – σχόλια

Με το πέρας της τέταρτης εκτέλεσης παρατηρούμε ότι η έξοδος του προγράμματος είναι ίδια με την έξοδο προγράμματος της εκτέλεσης 4, το οποίο είναι ορθό σύμφωνα με την λογική που σχεδιάστηκε το εργαλείο. Το πρόγραμμα θα κάνει πρώτα επαλήθευση μοναδικότητας των υπό-δικτύων του δικτύου και στη συνέχεια θα κάνει επαλήθευση μοναδικότητας των MAC διευθύνσεων του υπό-δικτύου. Έτσι αφού εντοπίστηκε σφάλμα στην επαλήθευση μοναδικότητας των υπό-δικτύων το πρόγραμμα τερμάτισε και δεν συνέχισε με τις υπόλοιπες διαδικασίες επαλήθευσης.

Εκτέλεση 5

Περιγραφή εκτέλεσης

Η πέμπτη εκτέλεση πάνω στην τοπολογία 1 έχει στηθεί με τα κατάλληλα configurations έτσι ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να ελέγξει αν κάθε συσκευή έχει έγκυρη IP διεύθυνση για το υπό-δίκτυο που ανήκει, συγκεκριμένα.

- Ορίζουμε εσφαλμένη IP διεύθυνση για το interface του κόμβου pc1 στο υπό-δίκτυο 223.1.1.0/24
- Ορίζουμε εσφαλμένη IP διεύθυνση για το interface του κόμβου pc5 στο υπό-δίκτυο 223.1.3.0/24

Έξοδος προγράμματος

```
----- Running network topology: 'topology_1' -----  
Device configuration completed!  
-IP Address Error:  
  Nodes 'pc1', interface: 'eth01', IP Address: '223.1.10.1' , does not belong to its subnet '223.1.1.0/24', host router: r1  
-IP Address Error:  
  Nodes 'pc5', interface: 'eth01', IP Address: '223.1.1.1' , does not belong to its subnet '223.1.3.0/24', host router: r3  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.3.6. Έξοδος εκτέλεσης 5, τοπολογία 1.

Παρατηρήσεις – σχόλια

Μετά το πέρας την πέμπτης εκτέλεσης παρατηρούμε ότι το πρόγραμμα ανίχνευση ότι η IP διευθύνσεις των interfaces του κόμβου pc1 όπως και του κόμβου pc5 είναι εσφαλμένες καθώς δεν ανήκουν στο υπό-δίκτυο. Το πρόγραμμα κάνει ελέγχους επαλήθευσης για όλες τις διευθύνσεις των interfaces και αφού εντοπίζει λάθος, τερματίζει την διαδικασία επαλήθευσης δικτύου.

Εκτέλεση 6

Περιγραφή εκτέλεσης

Η έκτη εκτέλεση πάνω στην τοπολογία 1 έχει στηθεί με τα κατάλληλα configurations έτσι ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να ελέγξει αν κάθε interface της συσκευής έχει έγκυρα ρυθμισμένο gateway για το υπό-δίκτυο που ανήκει. Ως gateway κάθε υπό-δικτύου ορίζεται το Router στο οποίο είναι στημένο το υπό-δίκτυο. Συγκεκριμένα:

- Ορίζουμε εσφαλμένη διεύθυνση gateway για τον κόμβο pc2 στο υπό-δίκτυο 223.1.1.0/24
- Ορίζουμε εσφαλμένη διεύθυνση gateway για τον κόμβο pc4 στο υπό-δίκτυο 222.1.2.0/24

Έξοδος προγράμματος

```
----- Running network topology: 'topology_1' -----  
  
Device configuration completed!  
  
-Gateway error: Invalid gateway at node 'pc2', interface: 'eth01', subnet: 223.1.1.0/24, host router: r1  
-Subnets Gateway: '223.1.1.3'  
-pc2 gateway '223.1.1.5'  
  
-Gateway error: Invalid gateway at node 'pc4', interface: 'eth01', subnet: 223.1.2.0/24, host router: r2  
-Subnets Gateway: '223.1.2.6'  
-pc4 gateway '223.1.2.100'  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.3.7. Έξοδος εκτέλεσης 6, τοπολογία 1.

Παρατηρήσεις – σχόλια

Με το πέρας της έκτης εκτέλεσης της τοπολογίας το πρόγραμμα εντόπισε διεύθυνση gateway στο interface του κόμβου pc2 η οποία δεν ταιριάζει με την διεύθυνση gateway του υπό-δικτύου 223.1.1.0/24 και διεύθυνση gateway στο Interface του κόμβου pc4 η οποία δεν ταιριάζει με την διεύθυνση gateway στο υπό-δίκτυο 223.1.2.0/24. Κάθε interface σε ένα υπό-δίκτυο πρέπει να έχει ως gateway, το gateway του υπό-δικτύου στο οποίο βρίσκεται. Το πρόγραμμα, αφού εντοπίζει σωστά αυτά τα λάθη τερματίζει.

Εκτέλεση 7

Περιγραφή εκτέλεσης

Η έβδομη εκτέλεση πάνω στην τοπολογία 1 έχει στηθεί με τα κατάλληλα configurations έτσι ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να ελέγξει αν κάθε IP διεύθυνση που έχει κάθε συσκευή στα interface της είναι μοναδική μέσα στο IP υπό-δίκτυο η όχι. Συγκεκριμένα

- Ορίσαμε διπλότυπες IP διευθύνσεις στα interfaces των pc3 & pc4 στο υπό-δίκτυο 223.1.2.0/24
- Ορίσαμε διπλότυπες IP διευθύνσεις στα interfaces των pc5 & pc6 στο υπό-δίκτυο 223.1.3.0/24

Έξοδος προγράμματος

```
----- Running network topology: 'topology_1' -----  
Device configuration completed!  
  
-IP Address Error: Found duplicate IP addresses at subnet:223.1.2.0/24 with host router: r2  
Node Interface IP_Address  
pc3      eth01  223.1.2.2  
pc4      eth01  223.1.2.2  
  
-IP Address Error: Found duplicate IP addresses at subnet:223.1.3.0/24 with host router: r3  
Node Interface IP_Address  
pc6      eth01  223.1.3.2  
pc5      eth01  223.1.3.2  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.3.8. Έξοδος εκτέλεσης 7, τοπολογία 1.

Παρατηρήσεις – σχόλια

Με το πέρας της έβδομης εκτέλεσης της τοπολογίας το πρόγραμμα εντόπισε διπλότυπες IP διευθύνσεις στα interfaces των κόμβων pc3 & pc4 στο υπό-δίκτυο 223.1.2.0/24 και στα interfaces των κόμβων pc5 & pc6 στο υπό-δίκτυο 223.1.3.0/24. Καθώς κάθε IP μέσα σε ένα υπό-δίκτυο πρέπει να είναι μοναδική, το πρόγραμμα αφού εντόπισε τα λάθη, τερμάτισε την λειτουργία του.

Εκτέλεση 8

Περιγραφή εκτέλεσης

Η όγδοη και τελευταία πάνω στην τοπολογία 1 έχει στηθεί με τα κατάλληλα configurations έτσι ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να επαληθεύσει τις διαδρομές που ορίζονται από τους πίνακες δρομολόγησης που βρίσκονται στους δρομολογητές - routers του δικτύου. Συγκεκριμένα

- Ορίσαμε μια λούπα στην επικοινωνία ανάμεσα στα υπό-δίκτυα 223.1.1.0/24 και 223.1.3.0/24
- Ορίσαμε μία λούπα στην επικοινωνία ανάμεσα στα υπό-δίκτυα 223.1.2.0/24 και 223.1.3.0/24

Έξοδος προγράμματος

```
----- Running network topology: 'topology_1' -----  
  
Device configuration completed!  
  
Subnet validation completed!  
  
Routing table validation Error:  
-Found loop in routing path from: '223.1.1.3/24', to: '223.1.3.27/24'  
-Node Router Path: -> r1 -> r2 -> r1  
  
Routing table validation Error:  
-Found loop in routing path from: '223.1.2.6/24', to: '223.1.3.27/24'  
-Node Router Path: -> r2 -> r1 -> r2  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.3.9. Έξοδος εκτέλεσης 8, τοπολογία 1.

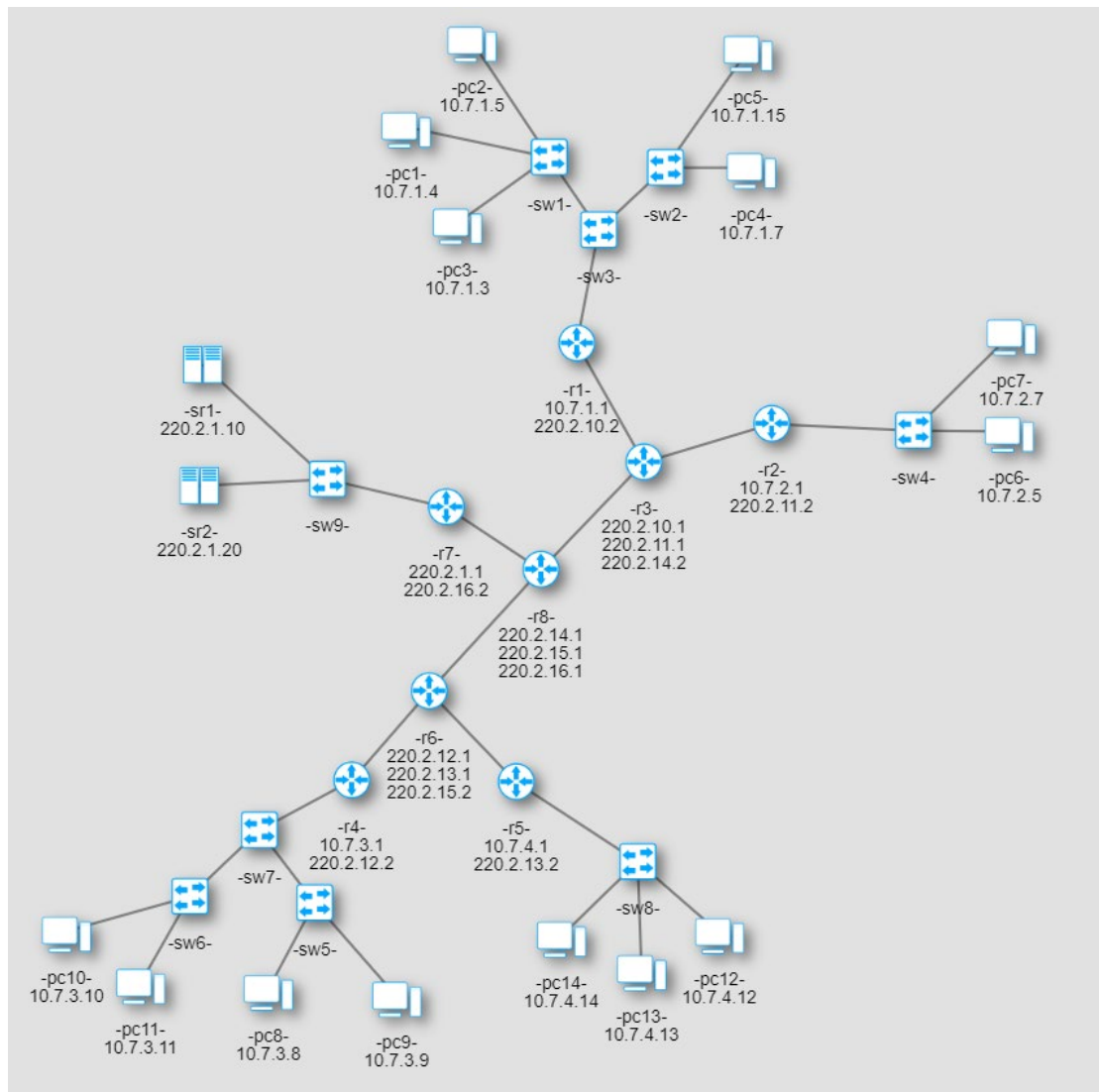
Παρατηρήσεις - σχόλια

Με το πέρας της όγδοης και τελευταίας εκτέλεσης της τοπολογίας η έξοδος του προγράμματος μας δείχνει ότι εντόπισε loop στις διαδρομές μεταξύ του υπό-δικτύου 223.1.1.0/24 που βρίσκεται στον δρομολογητή r1 και του υπό-δικτύου 223.1.3.0/24 που βρίσκεται στον δρομολογητή r3 και μεταξύ του υπό-δικτύου 223.1.2.0/24 που βρίσκεται στον δρομολογητή r2 και 223.1.3.0/24 που βρίσκεται στον δρομολογητή r3. Αυτό συμβαίνει γιατί το r1 για το προορισμό 223.1.3.0/24 προσπαθεί να επικοινωνήσει μέσω r2 και ο δρομολογητής r2 προσπαθεί να επικοινωνήσει με το 223.1.3.0/24 με το r1. Συνεπώς δημιουργείται μία λούπα κανένα από τα 2 υπό-δίκτυα δεν μπορεί να επικοινωνήσει με το τρίτο.

5.4 Τοπολογία 2

5.4.1 Οπτικοποίηση τοπολογίας

Παρακάτω βρίσκεται η οπτικοποίηση της δεύτερης τοπολογίας στην οποία θα τρέξουμε εκτελέσεις με διάφορα configurations. Η συγκεκριμένη οπτικοποίηση είναι της πρώτης εκτέλεσης όπου έχουμε κάνει αρχικοποίηση με ορθά configurations.



Εικόνα 5.4.1. Οπτικοποίηση τοπολογίας 2.

5.4.2 Περιγραφή Τοπολογίας

Το δίκτυο αποτελείται από 12 υπό-δίκτυα, πέντε φυσικά υπό-δίκτυα και 7 μεταξύ των interface των δρομολογητών.

Τα φυσικά υπό-δίκτυα είναι:

- Στο router r1: 10.7.1.0/24 στο οποίο εκτελείται NAT
- Στο router r2: 10.7.2.0/24 στο οποίο εκτελείται NAT
- Στο router r4: 10.7.3.0/24 στο οποίο εκτελείται NAT
- Στο router r5: 10.7.4.0/24 στο οποίο εκτελείται NAT
- Στο router r7: 220.2.1.0/24

Επίσης αποτελείται από 7 υπό-δίκτυα στα οποία είναι μεταξύ τους συνδεδεμένα οι δρομολογητές. Αυτά είναι:

- Μεταξύ router r1 – r3 : 220.2.10.0/24
- Μεταξύ router r2 – r3 : 220.2.11.0/24
- Μεταξύ router r4 – r6 : 220.2.12.0/24
- Μεταξύ router r5 – r6 : 220.2.13.0/24
- Μεταξύ router r3– r8 : 220.2.14.0/24
- Μεταξύ router r6 – r8: 220.2.15.0/24
- Μεταξύ router r7 – r8 : 220.2.16.0/24

5.4.3 Εκτελέσεις

Εκτέλεση 1

Περιγραφή εκτέλεσης

Η πρώτη εκτέλεση στην τοπολογία 2 έχει στηθεί με τα κατάλληλα configurations ώστε να θεωρείται ορθή.

Έξοδος προγράμματος

```
----- Running network topology: 'topology_2' -----  
  
Device configuration completed!  
  
Subnet validation completed!  
  
Routing table validation completed!  
  
Network validation completed!  
  
----- End of running: 'topology_2' -----
```

Εικόνα 5.4.2. Έξοδος εκτέλεσης 1, τοπολογία 2.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι η επαλήθευση των παραμέτρων του δικτύου επαληθεύτηκαν επιτυχώς, όπως και σχεδιάζαμε, συνεπώς το πρόγραμμα με το πέρας της εκτέλεσης τερματίζει επιτυχώς

Εκτέλεση 2

Περιγραφή εκτέλεσης

Η δεύτερη εκτέλεση στην τοπολογία 2 έχει στηθεί με τις κατάλληλες παραμέτρους – configurations ώστε να μην είναι ορθή βάση των αρχών σχεδίασης του δικτύου και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Τα λάθη αυτά γίνονται κατά την διαδικασία επαλήθευσης των διευθύνσεων MAC των συσκευών που βρίσκονται μέσα σε ένα υπό-δίκτυο. Συγκεκριμένα

- Ορίζουμε διπλότυπη MAC διεύθυνση μεταξύ των κόμβων pc2 και pc5 στο υπό-δίκτυο

- Ορίζουμε διπλότυπη MAC διεύθυνση μεταξύ των κόμβων pc12, pc13, pc14

Έξοδος προγράμματος

```
----- Running network topology: 'topology_2' -----  
  
Device configuration completed!  
  
-MAC Address Error: Found duplicate MAC addresses at subnet 10.7.1.0/24, host router r1:  
Node Interface    MAC_Address  
pc2      eth01 F4-B7-E2-75-56-02  
pc5      eth01 F4-B7-E2-75-56-02  
  
-MAC Address Error: Found duplicate MAC addresses at subnet 10.7.4.0/24, host router r5:  
Node Interface    MAC_Address  
pc14     eth01 F4-B7-E2-75-56-02  
pc12     eth01 F4-B7-E2-75-56-02  
pc13     eth01 F4-B7-E2-75-56-02  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.4.3. Έξοδος εκτέλεσης 2, τοπολογία 2.

Παρατηρήσεις – σχόλια

Η έξοδος του προγράμματος μας δείχνει ότι εντόπισε τις διπλότυπες MAC διευθύνσεις στο υπό-δίκτυο 10.7.1.0/24 όπου έχει για host του router r1 και τις διπλότυπες MAC διευθύνσεις στο υπό-δίκτυο 10.7.4.0/24 που έχει ως host το router r5. Παρατηρούμε ότι οι MAC διευθύνσεις είναι ίδιες και στις 5 συσκευές, αλλά δεν εντοπίζει το εντοπίζει το πρόγραμμα. Αυτό συμβαίνει διότι η διαδικασία επαλήθευσης των MAC διευθύνσεων επαληθεύει μόνο τις διευθύνσεις μέσα σε ένα υπό-δίκτυο. Μετά τον εντοπισμό των λαθών, εμφανίζεται κατάλληλο μήνυμα και τερματίζει η εκτέλεση του προγράμματος.

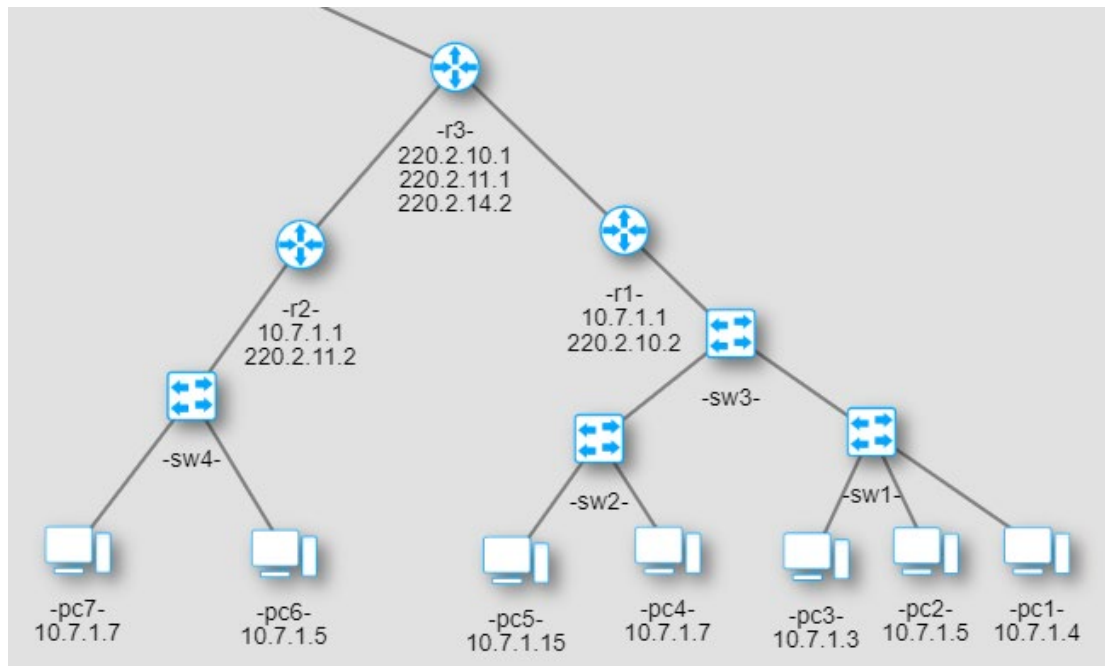
Εκτέλεση 3

Περιγραφή εκτέλεσης

Η τρίτη εκτέλεση στην τοπολογία 2 έχει στηθεί με τις κατάλληλες παραμέτρους configurations ώστε να είναι ορθή βάση των αρχών σχεδίασης του προγράμματος. Σκοπός είναι να δείξουμε μια περίπτωση λειτουργίας. Συγκεκριμένα, έχουμε στήσει δύο ίδια υπό-δίκτυα, το 10.7.1.0/24 στα Router r1 και r2 και θέλουμε να δούμε αν το πρόγραμμα θα εντοπίσει διπλότυπο δίκτυο ή όχι.

Οπτικοποίηση

Τα δύο υπό-δίκτυα 10.7.1.0/24 στα r1 και r2



Εικόνα 5.4.4. Στιγμιότυπο της οπτικοποίησης δικτύου.

Έξοδος προγράμματος

```
----- Running network topology: 'topology_2' -----  
Device configuration completed!  
Subnet validation completed!  
Routing table validation completed!  
Network validation completed!  
----- End of running: 'topology_2' -----
```

Εικόνα 5.4.5. Έξοδος εκτέλεσης 3, τοπολογία 2.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι η επαλήθευση δικτύου ολοκληρώθηκε επιτυχώς και δεν εντοπίστηκε κανένα διπλότυπο δίκτυο. Αυτό συμβαίνει καθώς τα δύο υπό-δίκτυα είναι ιδιωτικά-εκτελούν NAT. Μέσα σε ένα δίκτυο υπολογιστών είναι δυνατό να υπάρχουν πολλά ίδια υπό-δίκτυα NAT.

Εκτέλεση 4

Περιγραφή εκτέλεσης

Η τέταρτη εκτέλεση πάνω στην τοπολογία 2 έχει στηθεί με τα κατάλληλα configurations έτσι ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να ελέγξει αν κάθε συσκευή έχει έγκυρη διεύθυνση IP για το υπό-δίκτυο που ανήκει, συγκεκριμένα.

- Ορίζουμε εσφαλμένη IP διεύθυνση για τα interface των κόμβων pc13 και pc14 στο υπό-δίκτυο 10.7.4.0/24
- Ορίζουμε εσφαλμένη IP διεύθυνση για το interface του κόμβου sr1 στο υπό-δίκτυο 220.2.1.0/24

Έξοδος προγράμματος

```
----- Running network topology: 'topology_2' -----  
Device configuration completed!  
long to its subnet '10.7.4.0/24', host router: r5  
  
-IP Address Error:  
Nodes 'pc13', interface: 'eth01', IP Address: '10.8.4.13' , does not belong to its subnet '10.7.4.0/24', host router: r5  
  
-IP Address Error:  
Nodes 'sr1', interface: 'eth01', IP Address: '223.2.1.10' , does not belong to its subnet '220.2.1.0/24', host router: r7  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.4.6. Έξοδος εκτέλεσης 4, τοπολογία 2.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος της τέταρτης εκτέλεσης βλέπουμε ότι εντοπίστηκαν IP διευθύνσεις στα interfaces των κόμβων pc13, pc14 και sr1 όπου δεν ανήκουν στο τρέχον υπό-δίκτυο που βρίσκονται. Το πρόγραμμα αφού εντόπισε αυτά τα λάθη τερματίζει ανεπιτυχώς την διαδικασία επαλήθευσης δικτύου

Εκτέλεση 5

Περιγραφή εκτέλεσης

Η Πέμπτη εκτέλεση πάνω στην τοπολογία έχει στηθεί με τις κατάλληλες παραμέτρους – configurations ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να ελέγξει αν κάθε interface της συσκευής έχει έγκυρα ρυθμισμένο το gateway για το υπό-δίκτυο που ανήκει. Συγκεκριμένα:

- Ορίζουμε εσφαλμένη διεύθυνση για τον κόμβο pc10 στο υπό-δίκτυο 10.7.3.0/24
- Ορίζουμε εσφαλμένη διεύθυνση για τον κόμβο pc14 στο υπό-δίκτυο 10.7.4.0/24.

Σημείωση

Η εσφαλμένη διεύθυνση που ορίζουμε για τον κόμβο pc14 είναι η public διεύθυνση του υπό-δικτύου που έχει καταχωρηθεί στο router r4.

Έξοδος προγράμματος

```
----- Running network topology: 'topology_2' -----
-Gateway error: Invalid gateway at node 'pc10', interface: 'eth01', subnet: 10.7.3.0/24, host router: r4
-Subnets Gateway: '10.7.3.1'
-pc10 gateway '10.7.3.10'

-Gateway error: Invalid gateway at node 'pc14', interface: 'eth01', subnet: 10.7.4.0/24, host router: r5
-Subnets Gateway: '10.7.4.1'
-pc14 gateway '10.7.5.1'

Found network configuration errors. Network validation has ended!
```

Εικόνα 5.4.7. Έξοδος εκτέλεσης 5, τοπολογία 2.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος της τέταρτης εκτέλεσης βλέπουμε ότι εντοπίστηκαν εσφαλμένα gateway για τον κόμβο pc10 στο υπό-δίκτυο 10.7.3.0/24 και για τον κόμβο pc14 στο υπό-δίκτυο 10.7.4.0/24. Το gateway κάθε ενός κόμβο μέσα σε ένα υπό-δίκτυο πρέπει να είναι το interface του router του υπό-δικτύου. Παρατηρούμε όμως ότι ενώ το pc14 έχει ως gateway την public ip του υπό-δικτύου, το πρόγραμμα αναγνωρίζει ότι έχει εσφαλμένη διεύθυνση gateway. Αυτό συμβαίνει καθώς κάθε συσκευή σε ένα NAT υπό-δίκτυο πρέπει να αναγνωρίζει ως gateway την private ip του router και όχι την public.

Εκτέλεση 6

Περιγραφή εκτέλεσης

Η έκτη εκτέλεση πάνω στην τοπολογία 2 έχει στηθεί με τις κατάλληλες παραμέτρους-configuration έτσι ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να ελέγξει αν κάθε διεύθυνση IP διεύθυνση στα interface μιας συσκευής είναι μοναδική μέσα στο IP υπό-δίκτυο που ανήκει η όχι. Συγκεκριμένα:

- Ορίσαμε διπλότυπη διεύθυνση IP Interfaces του link μεταξύ των router r6 και r8 που είναι συνδεδεμένοι στο υπό-δίκτυο 220.2.15.0/24

Έξοδος προγράμματος

```
----- Running network topology: 'topology_2' -----  
  
Device configuration completed!  
  
-IP Address Error: Found duplicate IP address between router connections:  
  Connection: ('r6', 'r8'), link ID: 31  
  r6 interface: eth03, IP: 220.2.15.1  
  r8 interface: eth02, IP: 220.2.15.1  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.4.8. Έξοδος εκτέλεσης 6, τοπολογία 2.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι το πρόγραμμα εντόπισε διπλότυπη IP διεύθυνση στην σύνδεση μεταξύ των r6 και r8, στο link με id: 31. Το Interface του r6 είναι το eth03 και το Interface του r8 είναι το eth02. Αφού το πρόγραμμα εντόπισε ορθώς αυτά τα λάθη, τερμάτισε τη λειτουργία του ανεπιτυχώς.

Εκτέλεση 7

Περιγραφή εκτέλεσης

Η έβδομη εκτέλεση πάνω στην τοπολογία 2 έχει στηθεί με τις κατάλληλες παραμέτρους-configurations ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να επαληθεύσει τις διαδρομές που ορίζονται από τους πίνακες δρομολόγησης που βρίσκονται στους δρομολογητές-routers του δικτύου. Συγκεκριμένα:

- Έχουμε ορίσει στον πίνακα δρομολόγησης, όλη η κίνηση από το δρομολογητή r1, να ακολουθεί το default gateway του. Παρ' όλα αυτά στην

default διαδρομή, το next hop είναι μία διεύθυνση IP – interface με το οποίο δεν είναι συνδεδεμένο.

Έξοδος προγράμματος

```
----- Running network topology: 'topology_2' -----  
  
Device configuration completed!  
  
Subnet validation completed!  
  
Routing table validation Error:  
-Found error at router connections: Router r1 has no connection with an interface with IP addresss 220.2.10.5  
Route: From: '220.2.10.10/24', to: '220.2.11.10/24'  
  
Routing table validation Error:  
-Found error at router connections: Router r1 has no connection with an interface with IP addresss 220.2.10.5  
Route: From: '220.2.10.10/24', to: '220.2.12.10/24'  
  
Routing table validation Error:  
-Found error at router connections: Router r1 has no connection with an interface with IP addresss 220.2.10.5  
Route: From: '220.2.10.10/24', to: '220.2.13.10/24'  
  
Routing table validation Error:  
-Found error at router connections: Router r1 has no connection with an interface with IP addresss 220.2.10.5  
Route: From: '220.2.10.10/24', to: '220.2.1.1/24'  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.4.9. Έξοδος εκτέλεσης 7, τοπολογία 2.

Παρατηρήσεις

Από την έξοδο του προγράμματος βλέπουμε ότι κατά την διαδικασία επαλήθευσης των διαδρομών το πρόγραμμα εντοπίζει πρόβλημα. Συγκεκριμένα το εντοπίζει όταν το r1 στην επαλήθευση των διαδρομών του, σαν next hop έχει το interface με IP διεύθυνση 220.2.10.100 αλλά δεν έχει καμία σύνδεση με τέτοιο interface. Το λάθος εντοπίζεται πολλές φορές, καθώς πολλαπλές φορές έχει ως next hop το συγκεκριμένο Interface.

Αφού το πρόγραμμα εντοπίζει τα λάθη, τερματίζει ανεπιτυχώς την επαλήθευση δικτύου.

Εκτέλεση 8

Περιγραφή εκτέλεσης

Η όγδοη εκτέλεση πάνω στην τοπολογία 3 έχει στηθεί με τις κατάλληλες παραμέτρους-configurations ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να επαληθεύσει τις διαδρομές που ορίζονται από τους πίνακες δρομολόγησης που βρίσκονται στους δρομολογητές-router του δικτύου.

Συγκεκριμένα:

- Ορίζουμε διπλότυπες εγγραφές για έναν προορισμό στον πίνακα δρομολόγησης του r1 και r8

Έξοδος προγράμματος

```
----- Running network topology: 'topology_2' -----  
  
Device configuration completed!  
  
Subnet validation completed!  
  
-Routing Table Validation Error:  
Found duplicate destination entries at routers r3 routing table  
    dest      mask  next_hop  
220.2.11.2 255.255.255.0 220.2.10.2  
220.2.11.2 255.255.255.0 220.2.11.2  
  
-Routing Table Validation Error:  
Found duplicate destination entries at routers r8 routing table  
    dest      mask  next_hop  
220.2.10.10 255.255.255.0 220.2.14.2  
220.2.10.10 255.255.255.0 220.2.14.2  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.4.10. Έξοδος εκτέλεσης 8, τοπολογία 2.

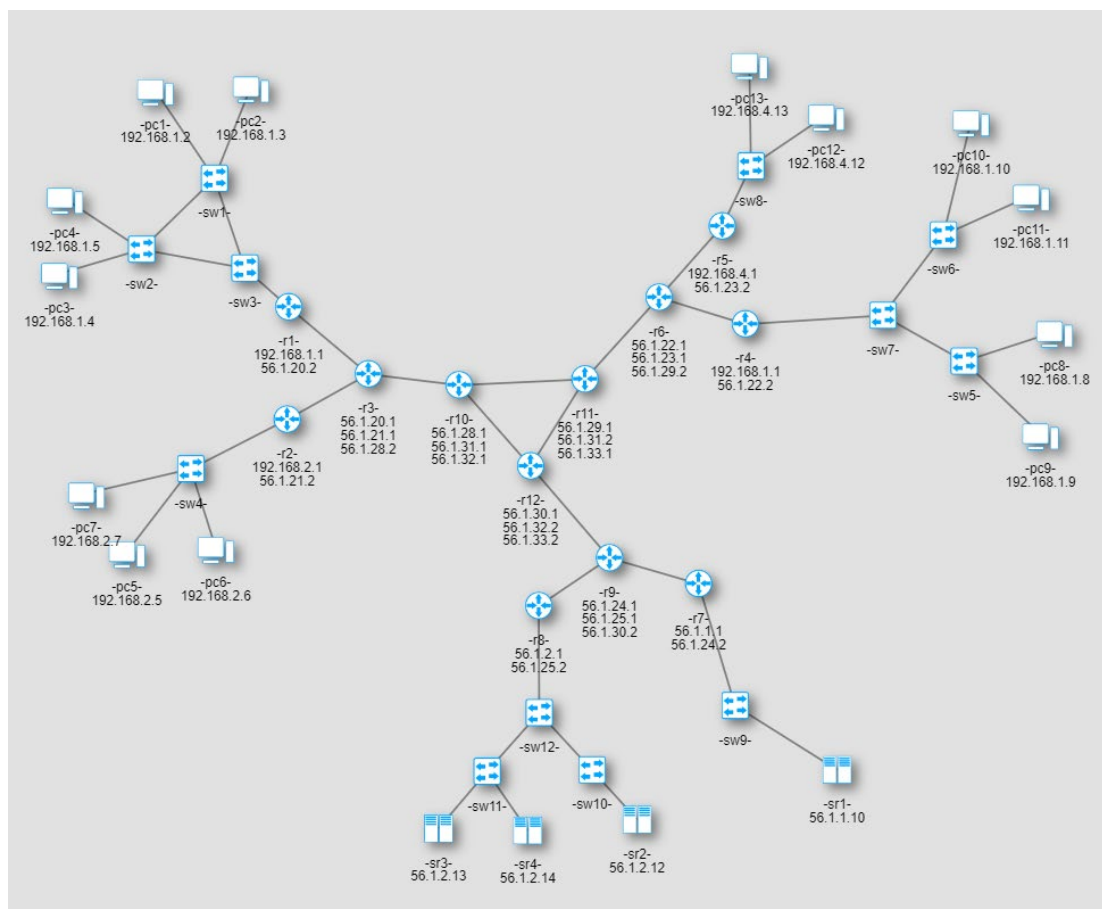
Παρατηρήσεις σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι εντόπισε διπλότυπες εγγραφές στους πίνακες δρομολόγησης του router r1 και r8. Καθώς κάθε εγγραφή σε έναν πίνακα δρομολόγησης πρέπει να είναι μοναδική, το πρόγραμμα ορθώς αναγνώρισε το λάθος και τερματίστηκε η επαλήθευση δικτύου ανεπιτυχώς.

5.5 Τοπολογία 3

5.5.1 Οπτικοποίηση τοπολογίας

Παρακάτω βρίσκεται η οπτικοποίηση της τοπολογίας στην οποία θα τρέξουμε εκτελέσεις με διάφορα configurations. Η συγκεκριμένη οπτικοποίηση είναι μιας πρώτης εκτέλεσης όπου έχουμε κάνει αρχικοποίηση με ορθά configurations.



Εικόνα 5.5.1. Οπτικοποίηση τοπολογίας 3.

5.5.2 Περιγραφή τοπολογίας

Το παραπάνω δίκτυο αποτελείται από 18 υπό-δίκτυα, 6 από τα οποία είναι φυσικά υπό-δίκτυα και τα υπόλοιπα 12 μεταξύ των δρομολογητών.

Τα φυσικά υπό-δίκτυα είναι

- Στο router r1: 192.168.1.0/24 στο οποίο εκτελείται NAT, με public IP: 56.1.20.3/24
- Στο router r2: 192.168.2.0/24 στο οποίο εκτελείται NAT, με public IP: 56.1.21.3/24
- Στο router r4: 192.168.1.0/24 στο οποίο εκτελείται NAT, με public IP: 56.1.22.3/24
- Στο router r5: 192.168.4.0/24 στο οποίο εκτελείται NAT, με public IP: 56.1.23.3/24
- Στο router r7: 56.1.1.0/24
- Στο router r8: 56.1.2.0/24

Τα υπό-δίκτυα μεταξύ των δρομολογητών είναι:

- Μεταξύ r1-r3: 56.1.20.0/24
- Μεταξύ r2-r3: 56.1.21.0/24
- Μεταξύ r4-r6: 56.1.22.0/24
- Μεταξύ r5-r6: 56.1.23.0/24
- Μεταξύ r7-r9: 56.1.24.0/24
- Μεταξύ r8-r9: 56.1.25.0/24
- Μεταξύ r3-r10: 56.1.28.0/24
- Μεταξύ r6-r11: 56.1.29.0/24
- Μεταξύ r9-r12: 56.1.30.0/24
- Μεταξύ r10-r11: 56.1.31.0/24
- Μεταξύ r10-r12: 56.1.32.0/24
- Μεταξύ r11-r12: 56.1.33.0/24

5.5.3 Εκτελέσεις

Εκτέλεση 1

Περιγραφή εκτέλεσης

Η πρώτη εκτέλεση στην τοπολογία 3 έχει στηθεί με τις κατάλληλες παραμέτρους-configurations ώστε να θεωρείται ορθή και το πρόγραμμα να προσπαθεί να το επαληθεύσει

Έξοδος προγράμματος

```
----- Running network topology: 'topology_3' -----  
  
Device configuration completed!  
  
Subnet validation completed!  
  
Routing table validation completed!  
  
Network validation completed!  
  
----- End of running: 'topology_3' -----
```

Εικόνα 5.5.2. Έξοδος εκτέλεσης 1 , τοπολογία 3.

Παρατηρήσεις - σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι η επαλήθευση των παραμέτρων του δικτύου επαληθεύτηκαν επιτυχώς, όπως και σχεδιάζαμε, συνεπώς το πρόγραμμα με το πέρας της εκτέλεσης τερματίζει επιτυχώς

Εκτέλεση 2

Περιγραφή εκτέλεσης

Η δεύτερη εκτέλεση στην τοπολογία 3 έχει στηθεί με τις κατάλληλες παραμέτρους-configurations ώστε να μην είναι ορθή, βάση των επαληθεύσεων που είναι σχεδιασμένο το πρόγραμμα να πραγματοποιεί. Τα λάθη αυτά γίνονται κατά την

διαδικασία επαλήθευσης των διευθύνσεων MAC των συσκευών που βρίσκονται μέσα σε ένα υπό-δίκτυο. Συγκεκριμένα:

- Ορίζουμε την ίδια MAC διεύθυνση στα interface eth01 των κόμβων pc1, pc2, sw1 που ανήκουν στο υπό-δίκτυο 192.168.1.0/24 που σαν host έχει το router r1
- Ορίζουμε την ίδια διεύθυνση MAC στα interface eth01, eth02 του sw5, eth01 του sw6 και eth01 του sw7 που ανήκουν στο υπό-δίκτυο 192.168.1.0/24 με host το router r4

Έξοδος προγράμματος

```
----- Running network topology: 'topology_3' -----  
Device configuration completed!  
  
-MAC Address Error: Found duplicate MAC addresses at subnet 192.168.1.0/24, host router r1:  
Node Interface      MAC_Address  
pc2      eth01 F4-B7-E2-75-56-02  
pc1      eth01 F4-B7-E2-75-56-02  
sw1      eth01 F4-B7-E2-75-56-02  
  
-MAC Address Error: Found duplicate MAC addresses at subnet 192.168.1.0/24, host router r4:  
Node Interface      MAC_Address  
sw5      eth01 00-11-22-33-44-51  
sw5      eth02 00-11-22-33-44-51  
sw6      eth02 00-11-22-33-44-51  
sw7      eth03 00-11-22-33-44-51  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.5.3. Έξοδος εκτέλεσης 2, τοπολογία 3.

Παρατηρήσεις – σχόλια

Η έξοδος του προγράμματος μας δείχνει ότι εντόπισε ίδιες MAC διευθύνσεις σε 2 υπό-δίκτυα. Πρώτα, στο υπό-δίκτυο 192.168.1.0/24 που έχει ως host το r1, μεταξύ των Interfaces των κόμβων pc1, pc2, sw1 και δεύτερον στο υπό-δίκτυο 192.168.1.0/24 που έχει ως host router το r4 μεταξύ των interfaces των κόμβων sw5, sw6, sw7. Μετά τον ορθό εντοπισμό των λαθών, εμφανίζεται κατάλληλο μήνυμα και τερματίζει η εκτέλεση του προγράμματος

Εκτέλεση 3

Περιγραφή εκτέλεσης

Η τρίτη εκτέλεση στην τοπολογία 3 έχει στηθεί με τις κατάλληλες παραμέτρους-configurations ώστε να μην είναι ορθή, βάση των επαληθεύσεων που είναι σχεδιασμένο το πρόγραμμα να πραγματοποιεί. Τα λάθη αυτά γίνονται κατά την διαδικασία επαλήθευσης των IP υπό-δικτύων, όπου ορίζουμε διπλότυπα IP υπό-δίκτυα.

Συγκεκριμένα:

- Ορίζουμε ίδια διεύθυνση υπό-δικτύου μεταξύ του υπό-δικτύου που είναι συνδεδεμένα μεταξύ τους τα r3, r10 και μεταξύ του υπό-δικτύου που είναι συνδεδεμένα μεταξύ τους τα r9, r12
- Ορίζουμε τις ίδιες εσφαλμένες MAC διευθύνσεις με την εκτέλεση 2

Έξοδος προγράμματος

```
----- Running network topology: 'topology_3' -----  
  
Device configuration completed!  
  
Subnet error: Found duplicate subnets:  
Router_ID      Address  
r3 r10 56.1.28.0/24  
r9 r12 56.1.28.0/24  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.5.4. Έξοδος εκτέλεσης 3, τοπολογία 3.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος στην τρίτη εκτέλεση βλέπουμε ότι εντοπίστηκαν διπλότυπα IP υπό-δίκτυα μεταξύ των δρομολογητών. Συγκεκριμένα, εντοπίστηκε διπλότυπο δίκτυο με διεύθυνση 56.1.28.0/24 μεταξύ του υπό-δικτύου των δρομολογητών r3, r10 και του υπό-δικτύου μεταξύ των δρομολογητών r9, r12. Αφού εντοπίστηκαν ορθώς τα λάθη, η επαλήθευση δικτύου τερμάτισε ανεπιτυχώς δίχως να συνεχίσει με τους υπόλοιπους ελέγχους. Αυτό σημαίνει, ότι παρότι έχουμε ορίσει και λάθη σε άλλα σημεία της επαλήθευσης, δεν θα εντοπιστούν καθώς σταματά η εκτέλεση.

Εκτέλεση 4

Περιγραφή εκτέλεσης

Η τέταρτη εκτέλεση πάνω στην τοπολογία 3 έχει στηθεί με τις κατάλληλες παραμέτρους-configurations ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να ελέγξει αν κάθε συσκευή έχει έγκυρη διεύθυνση IP για το υπό-δίκτυο που ανήκει.

Συγκεκριμένα:

- Ορίζουμε μεταξύ της σύνδεσης των κόμβων r9, r12, IP διευθύνσεις διαφορετικών υπό-δικτύων

Έξοδος προγράμματος

```
----- Running network topology: 'topology_3' -----  
Device configuration completed!  
-Router IP Error:  
  Nodes: r9, interface: eth03 and Nodes: r12, interface eth01 at link with ID: '3', does not belong at the same subnet.  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.5.5. Έξοδος εκτέλεσης 4, τοπολογία 3.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι το πρόγραμμα εντόπισε στα interface του link 3 μεταξύ των κόμβων r9 και r12 IP διευθύνσεις που ανήκουν σε διαφορετικό υπό-δίκτυο. Το πρόγραμμα το αναγνώρισε ως λάθος καθώς 2 δρομολογητές που είναι συνδεδεμένοι μεταξύ τους, για να μπορούν να έχουν επικοινωνία πρέπει τα interface της σύνδεσης τους να ανήκουν στο ίδιο IP υπό-δίκτυο.

Εκτέλεση 5

Περιγραφή εκτέλεσης

Η Πέμπτη εκτέλεση πάνω στην τοπολογία έχει στηθεί με τις κατάλληλες παραμέτρους-configurations ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να ελέγξει αν κάθε interface της συσκευής έχει έγκυρα ρυθμισμένη IP διεύθυνση για το gateway του υπό-δικτύου που ανήκει. Συγκεκριμένα:

- Ορίζουμε εσφαλμένη διεύθυνση gateway στα interface των κόμβων pc1, pc5, pc6 και sr1

Έξοδος προγράμματος

```

----- Running network topology: 'topology_3' -----
Device configuration completed!
-Gateway error: Invalid gateway at node 'pc1', interface: 'eth01', subnet: 192.168.1.0/24
  -Subnets Gateway: '192.168.1.1',
  -pc1 gateway '192.168.1.10'
-Gateway error: Invalid gateway at node 'pc6', interface: 'eth01', subnet: 192.168.2.0/24
  -Subnets Gateway: '192.168.2.1',
  -pc6 gateway '192.168.2.10'
-Gateway error: Invalid gateway at node 'pc5', interface: 'eth01', subnet: 192.168.2.0/24
  -Subnets Gateway: '192.168.2.1',
  -pc5 gateway '192.168.2.10'
-Gateway error: Invalid gateway at node 'sr1', interface: 'eth01', subnet: 56.1.1.0/24
  -Subnets Gateway: '56.1.1.1',
  -sr1 gateway '56.1.1.10'
Found network configuration errors. Network validation has ended!

```

Εικόνα 5.5.6. Έξοδος εκτέλεσης 5, τοπολογία 3.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι εντοπίστηκαν εσφαλμένα gateway για τα interfaces των κόμβων pc1 που ανήκει στο υπό-δίκτυο 192.168.1.0/24, pc5 που ανήκει στο 192.168.2.0/24, pc6 που ανήκει στο 192.168.2.0/24 και sr1 που ανήκει στο 56.1.1.0/24. Αποτέλεσμα των μη έγκυρων διευθύνσεων gateway είναι το να μην μπορεί μία συσκευή να έχει επικοινωνία μέσω του δικτύου. Αφού το πρόγραμμα ορθώς εντόπισε τα λάθη, τερματίζει ανεπιτυχώς την επαλήθευση δικτύου.

Εκτέλεση 6

Περιγραφή εκτέλεσης

Η έκτη εκτέλεση πάνω στην τοπολογία 3 έχει στηθεί με τις κατάλληλες παραμέτρους-configurations ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη.

Σκοπός είναι να ελέγξει την διαδικασία επαλήθευσης IP υπό-δικτύων , ελέγχοντας αν κάθε IP διεύθυνση που έχει οριστεί στα interface μιας συσκευής είναι μοναδική για το υπό-δίκτυο που ανήκει. Συγκεκριμένα:

- Ορίζουμε ίδια IP διεύθυνση στο υπό-δίκτυο: 56.1.30.0/24 μεταξύ δρομολογητών στα interface των κόμβων r9 r12, στο μεταξύ τους link με ID 38.

Έξοδος προγράμματος

```
----- Running network topology: 'topology_3' -----  
  
Device configuration completed!  
  
-IP Address Error: Found duplicate IP address between router connections:  
Connection: ('r9', 'r12'), link ID: 38  
r9 interface: eth03, IP: 56.1.30.1  
r12 interface: eth01, IP: 56.1.30.1  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.5.7. Έξοδος εκτέλεσης 6, τοπολογία 3..

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι κατά την διαδικασία επαλήθευσης των IP υπό-δικτύων βρίσκουμε διπλότυπη IP διεύθυνση μεταξύ των interface των κόμβων r9, και r12 που ανήκουν στο ίδιο υπό-δίκτυο δρομολογητών με διεύθυνση 56.1.30.0/24. Καθώς κάθε IP διεύθυνση μέσα σε ένα υπό-δίκτυο πρέπει να είναι μοναδική, το πρόγραμμα αφού ορθώς εντοπίσει το πρόβλημα, τερματίζει ανεπιτυχώς την επαλήθευση δικτύου.

Εκτέλεση 7

Περιγραφή εκτέλεση

Η έβδομη εκτέλεση πάνω στην τοπολογία 3 έχει στηθεί με τις κατάλληλες παραμέτρους-configurations ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη. Σκοπός είναι να επαληθεύσει τις διαδρομές που ορίζονται από τους πίνακες δρομολόγησης που βρίσκονται στους δρομολογητές-router του δικτύου.

Συγκεκριμένα:

- Ορίζουμε στον πίνακα δρομολόγησης του δρομολογητή r9 ότι κίνηση υπάρχει προς το υπό-δίκτυα 56.1.1.0/24 στο router r12.

Έξοδος προγράμματος

```
----- Running network topology: 'topology_3' -----  
  
Device configuration completed!  
  
Subnet validation completed!  
  
Routing table validation Error:  
-Found loop in routing path from: '56.1.20.2/24', to: '56.1.1.1/24'  
-Node Router Path: -> r1 -> r3 -> r10 -> r12 -> r9 -> r9  
  
Routing table validation Error:  
-Found loop in routing path from: '56.1.21.2/24', to: '56.1.1.1/24'  
-Node Router Path: -> r2 -> r3 -> r10 -> r12 -> r9 -> r9  
  
Routing table validation Error:  
-Found loop in routing path from: '56.1.22.2/24', to: '56.1.1.1/24'  
-Node Router Path: -> r4 -> r6 -> r11 -> r12 -> r9 -> r9  
  
Routing table validation Error:  
-Found loop in routing path from: '56.1.23.2/24', to: '56.1.1.1/24'  
-Node Router Path: -> r5 -> r6 -> r11 -> r12 -> r9 -> r9  
  
Routing table validation Error:  
-Found loop in routing path from: '56.1.2.1/24', to: '56.1.1.1/24'  
-Node Router Path: -> r8 -> r9 -> r9  
  
Found network configuration errors. Network validation has ended!
```

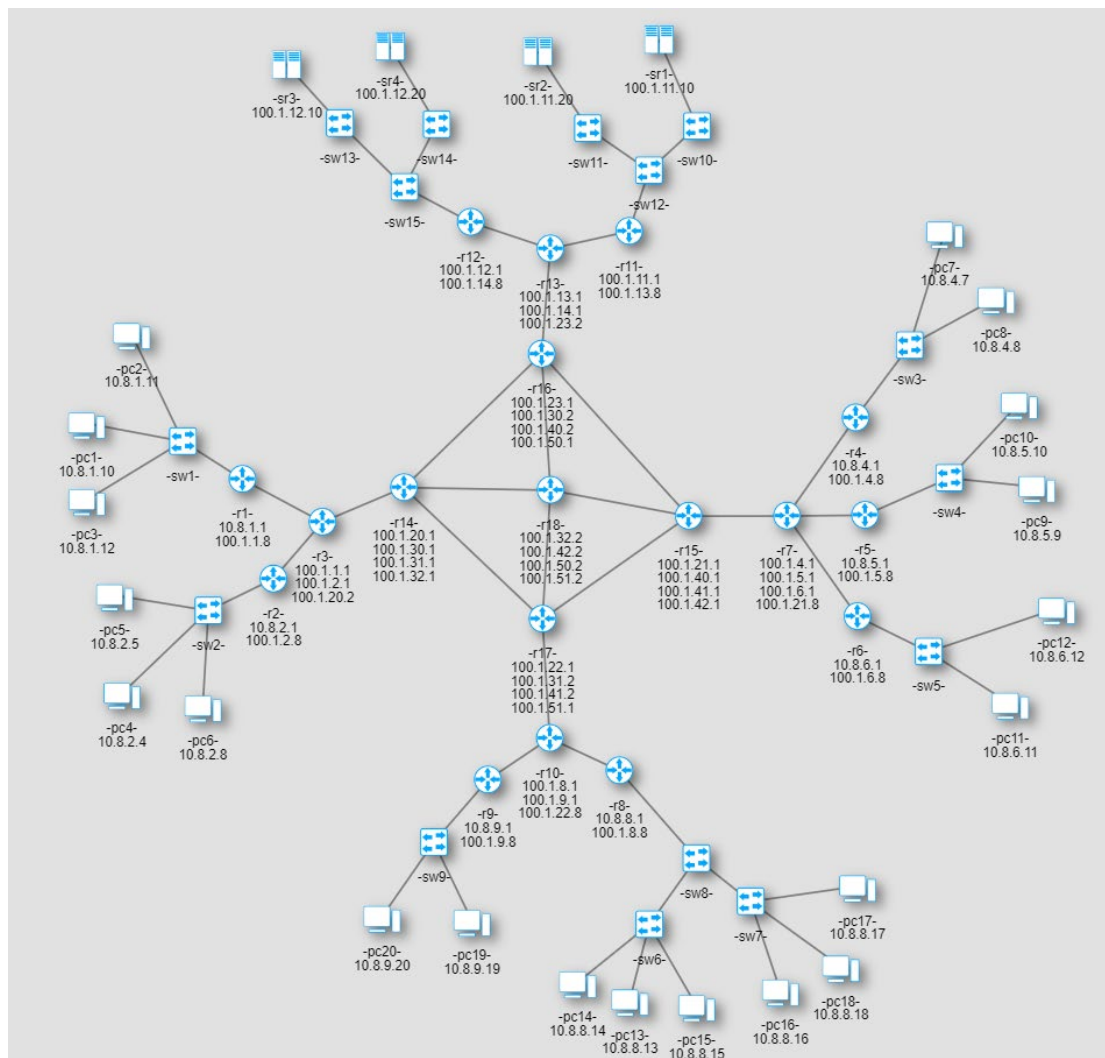
Εικόνα 5.5.8. Έξοδος εκτέλεσης 7, τοπολογία 3.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι εντοπίστηκαν πολλές λούπες στην προσπάθεια επικοινωνίας των υπό-δικτύων με το υπό-δίκτυο 56.1.1.10/24. Αυτό συνέβη γιατί όλη η επικοινωνία με αυτό το υπό-δίκτυο περνάει από το r9. Το r9 όμως με τη σειρά του αντί να προωθήσει την επικοινωνία προς το r7 για να καταλήξει στο υπό-

5.6 Τοπολογία 4

Παρακάτω βρίσκεται η οπτικοποίηση της τοπολογίας στην οποία θα τρέξουμε εκτελέσεις με διάφορα configurations. Η συγκεκριμένη οπτικοποίηση είναι μιας πρώτης εκτέλεσης όπου έχουμε κάνει αρχικοποίηση με ορθά configurations.



104

5.6.2 Περιγραφή τοπολογίας

Το δίκτυο αποτελείται συνολικά από 30 υπό-δίκτυα. Από αυτά τα 9 είναι φυσικά υπό-δίκτυα, δηλαδή υπό-δίκτυα που περιλαμβάνουν συσκευές όπως υπολογιστές, και τα υπόλοιπα υπό-δίκτυα στα οποία είναι συνδεδεμένοι μεταξύ τους οι δρομολογητές του δικτύου.

Τα φυσικά

υπό-δίκτυα είναι:

- Στο router r1: 10.8.1.0/24 στο οποίο εκτελείται NAT με public IP διεύθυνση του router: 100.1.1.8
- Στο router r2: 10.8.2.0/24 στο οποίο εκτελείται NAT με public IP διεύθυνση του router: 100.1.2.8
- Στο router r4: 10.8.4.0/24 στο οποίο εκτελείται NAT με public IP διεύθυνση του router: 100.1.4.8
- Στο router r5 :10.8.5.0/24 στο οποίο εκτελείται NAT με public IP διεύθυνση του router: 100.1.5.8
- Στο router r6 :10.8.6.0/24 στο οποίο εκτελείται NAT με public IP διεύθυνση του router: 100.1.6.8
- Στο router r8 : 10.8.8.0/24 στο οποίο εκτελείται NAT με public IP διεύθυνση του router: 100.8.8.8
- Στο router r9: 10.8.9.0/24 στο οποίο εκτελείται NAT με public IP διεύθυνση του router: 100.8.9.8
- Στο router r11: 100.1.11.0/24
- Στο router r12: 100.1.12.0 /24

Τα υπό-δίκτυα μεταξύ των δρομολογητών είναι:

- Μεταξύ r1 – r3 : 100.1.1.0/24
- Μεταξύ r2 – r3 : 100.1.2.0/24
- Μεταξύ r4 – r7 : 100.1.4.0/24
- Μεταξύ r5 – r7 : 100.1.5.0/24
- Μεταξύ r6 – r7 : 100.1.6.0/24
- Μεταξύ r8 – r10 : 100.1.8.0/24
- Μεταξύ r9 – r10 : 100.1.9.0/24
- Μεταξύ r11 – r13 : 100.1.11.0/24
- Μεταξύ r12 – r13 : 100.1.12.0/24
- Μεταξύ r3 – r14 : 100.1.20.0/24

- Μεταξύ r7 – r15 : 100.1.21.0/24
- Μεταξύ r10 – r17 : 100.1.22.0/24
- Μεταξύ r13 – r16 : 100.1.23.0/24
- Μεταξύ r14 – r16 : 100.1.30.0/24
- Μεταξύ r14 – r17 : 100.1.31.0/24
- Μεταξύ r14 – r18 : 100.1.32.0/24
- Μεταξύ r15 – r16 : 100.1.40.0/24
- Μεταξύ r15 – r17 : 100.1.41.0/24
- Μεταξύ r15 – r15 : 100.1.42.0/24
- Μεταξύ r16 – r18 : 100.1.50.0/24
- Μεταξύ r17 – r18 : 100.1.51.0/24

5.6.3 Εκτελέσεις

Εκτέλεση 1

Περιγραφή εκτέλεσης

Η πρώτη εκτέλεση στην τοπολογία 4 έχει στηθεί με τις κατάλληλες παραμέτρους – configuration ώστε να θεωρείται ορθή και το πρόγραμμα να προσπαθεί να το επαληθεύσει

Έξοδος προγράμματος

```

----- Running network topology: 'topology_4' -----

Device configuration completed!

Subnet validation completed!

Routing table validation completed!

Network validation completed!

----- End of running: 'topology_4' -----

```

Εικόνα 5.6.2. Έξοδος εκτέλεσης , τοπολογία 4.

Παρατηρήσεις - σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι η επαλήθευση των παραμέτρων του δικτύου επαληθεύτηκαν επιτυχώς, όπως και σχεδιάζαμε, συνεπώς το πρόγραμμα με το πέρας της εκτέλεσης τερματίζει επιτυχώς.

Εκτέλεση 1

Περιγραφή εκτέλεσης

Η δεύτερη εκτέλεση στην τοπολογία 3 έχει στηθεί με τις κατάλληλες παραμέτρους – configurations ώστε να μην είναι ορθή, βάση των επαληθεύσεων που είναι σχεδιασμένο το πρόγραμμα να πραγματοποιεί. Τα λάθη αυτά γίνονται κατά την διαδικασία επαλήθευσης των διευθύνσεων MAC των συσκευών που βρίσκονται μέσα σε ένα υπό-δίκτυο. Συγκεκριμένα:

- Ορίζουμε ίδιες MAC διευθύνσεις μεταξύ των interfaces των κόμβων pc1, pc2, sw1, r1 στο υπό-δίκτυο 10.8.1.0/24 με host router το r1
- Ορίζουμε ίδιες MAC διευθύνσεις μεταξύ των interfaces των κόμβων pc11, r6 στο υπό-δίκτυο 10.8.6.0/24 με host router το r6
- Ορίζουμε ίδιες MAC διευθύνσεις μεταξύ των interfaces των κόμβων sr3, sr4, sw15, r12 στο υπό-δίκτυο 100.1.12.0/24 με host router το r12

Έξοδος προγράμματος

```
----- Running network topology: 'topology_4' -----  
  
Device configuration completed!  
  
-MAC Address Error: Found duplicate MAC addresses at subnet 10.8.1.0/24, host router r1:  
Node Interface      MAC_Address  
pc2      eth01 00-D0-56-F2-B5-02  
pc1      eth01 00-D0-56-F2-B5-02  
sw1      eth01 00-D0-56-F2-B5-02  
r1      eth01 00-D0-56-F2-B5-02  
  
-MAC Address Error: Found duplicate MAC addresses at subnet 10.8.6.0/24, host router r6:  
Node Interface      MAC_Address  
pc11     eth01 00-D0-56-F2-B5-11  
r6      eth01 00-D0-56-F2-B5-11  
  
-MAC Address Error: Found duplicate MAC addresses at subnet 100.1.12.0/24, host router r12:  
Node Interface      MAC_Address  
sr4      eth01 00-D0-56-F2-B5-03  
sr3      eth01 00-D0-56-F2-B5-03  
sw15     eth01 00-D0-56-F2-B5-03  
r12      eth01 00-D0-56-F2-B5-03  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.6.3. Έξοδος εκτέλεσης 2, τοπολογία 4.

Παρατηρήσεις – σχόλια

Η έξοδος του προγράμματος μας δείχνει ότι εντόπισε ίδιες MAC διευθύνσεις σε interfaces κόμβων που ανήκουν σε 3 διαφορετικά υπό-δίκτυα του δικτύου. Στο 10.8.1.0/24 με host router το r1, στο 10.8.6.0/24 με host router το r6 και στο 100.1.12.0/24 με host router το r12. Η ύπαρξη ιδίων διευθύνσεων MAC μέσα σε ένα υπό-δίκτυο είναι ικανό να προκαλέσει προβλήματα επικοινωνίας μεταξύ των συσκευών στο data link επίπεδο δικτύου. Μετά τον ορθό εντοπισμό των λαθών, τερματίζεται ανεπιτυχώς η εκτέλεση του προγράμματος.

Εκτέλεση 3

Περιγραφή εκτέλεσης

Η τρίτη εκτέλεση στην τοπολογία 3 έχει στηθεί με τις κατάλληλες παραμέτρους – configurations ώστε να μην είναι ορθή, βάση των επαληθεύσεων που είναι σχεδιασμένο το πρόγραμμα να πραγματοποιεί. Τα λάθη αυτά γίνονται κατά την διαδικασία επαλήθευσης των διευθύνσεων MAC που βρίσκονται στα interfaces που ανήκουν στα δύο άκρα ενός link μεταξύ δύο δρομολογητών. Συγκεκριμένα:

- Ορίζουμε διπλότυπη MAC διεύθυνση μεταξύ των interfaces των κόμβων r6, r7 στο link με ID 44
- Ορίζουμε διπλότυπη MAC διεύθυνση μεταξύ των interfaces των κόμβων r12, r13 στο link με ID 48
- Ορίζουμε διπλότυπη MAC διεύθυνση μεταξύ των interfaces των κόμβων r14, r17 στο link με ID 54

Έξοδος προγράμματος

```
----- Running network topology: 'topology_4' -----  
  
Device configuration completed!  
  
-MAC Address Error: Found duplicate MAC addresses at link between routers with link ID: 44  
Node   Interface  MAC_Address  
r6      eth02      07-D0-56-F2-B5-03  
r7      eth03      07-D0-56-F2-B5-03  
  
-MAC Address Error: Found duplicate MAC addresses at link between routers with link ID: 48  
Node   Interface  MAC_Address  
r12    eth02      12-D0-56-F2-B5-02  
r13    eth02      12-D0-56-F2-B5-02  
  
-MAC Address Error: Found duplicate MAC addresses at link between routers with link ID: 54  
Node   Interface  MAC_Address  
r14    eth03      14-D0-56-F2-B5-03  
r17    eth02      14-D0-56-F2-B5-03  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.6.4. Έξοδος εκτέλεσης 3, τοπολογία 4.

Παρατηρήσεις – σχόλια

Η έξοδος του προγράμματος μας δείχνει ότι εντόπισε διπλότυπες διευθύνσεις MAC στα interfaces των link μεταξύ των router. Συγκεκριμένα εντόπισε διπλότυπη MAC μεταξύ των interfaces των κόμβων r6, r7 στο link με ID 44, στη συνέχεια μεταξύ των Interfaces των r12, r13 στο link με ID 48 και τέλος στα interfaces των κόμβων r14, r17 στο link με ID 54. Η ύπαρξη διπλότυπων διευθύνσεων στις άκρες ενός Link μεταξύ δύο συσκευών είναι δυνατόν να προκαλέσει σοβαρά προβλήματα στην επικοινωνία των Router στο data link layer. Αφού το πρόγραμμα ορθώς εντόπισε τα σφάλματα, τερματίζει ανεπιτυχώς την επαλήθευση δικτύου δίχως να συνεχίσει με τις υπόλοιπες επαληθεύσεις.

Εκτέλεση 4

Περιγραφή εκτέλεσης

Η τέταρτη εκτέλεση στην τοπολογία 3 έχει στηθεί με τις κατάλληλες παραμέτρους – configurations ώστε να μην είναι ορθή, βάση των επαληθεύσεων που είναι σχεδιασμένο το πρόγραμμα να πραγματοποιοει. Τα λάθη αυτά γίνονται κατά την διαδικασία επαλήθευσης των διευθύνσεων MAC των συσκευών που βρίσκονται μέσα σε ένα υπό-δίκτυο. Σε αυτή την εκτέλεση έχουμε ορίσει όλα τα λάθη που ορίσαμε στις προηγούμενες δύο εκτελέσεις (Εκτέλεση 2 και Εκτέλεση 3). Δηλαδή την επαλήθευση των MAC διευθύνσεων των interfaces μέσα σε ένα υπό-δίκτυο αλλά και την επαλήθευση των MAC διευθύνσεων των Interfaces που βρίσκονται στις άκρες ενός link μεταξύ δύο δρομολογητών. Συγκεκριμένα:

- Ορίζουμε ίδιες MAC διευθύνσεις μεταξύ των interfaces των κόμβων pc1, pc2, sw1, r1 στο υπό-δίκτυο 10.8.1.0/24 με host router το r1
- Ορίζουμε ίδιες MAC διευθύνσεις μεταξύ των interfaces των κόμβων pc11, r6 στο υπό-δίκτυο 10.8.6.0/24 με host router το r6
- Ορίζουμε ίδιες MAC διευθύνσεις μεταξύ των interfaces των κόμβων sr3, sr4, sw15, r12 στο υπό-δίκτυο 100.1.12.0/24 με host router το r12
- Ορίζουμε διπλότυπη MAC διεύθυνση μεταξύ των interfaces των κόμβων r6, r7 στο link με ID 44
- Ορίζουμε διπλότυπη MAC διεύθυνση μεταξύ των interfaces των κόμβων r12, r13 στο link με ID 48
- Ορίζουμε διπλότυπη MAC διεύθυνση μεταξύ των interfaces των κόμβων r14, r17 στο link με ID 54
-

Έξοδος προγράμματος

```
----- Running network topology: 'topology_4' -----  
Device configuration completed!  
  
-MAC Address Error: Found duplicate MAC addresses at subnet 10.8.1.0/24, host router r1:  
Node Interface      MAC_Address  
pc1    eth01 00-D0-56-F2-B5-02  
sw1    eth01 00-D0-56-F2-B5-02  
pc2    eth01 00-D0-56-F2-B5-02  
r1     eth01 00-D0-56-F2-B5-02  
  
-MAC Address Error: Found duplicate MAC addresses at subnet 10.8.6.0/24, host router r6:  
Node Interface      MAC_Address  
pc11   eth01 00-D0-56-F2-B5-11  
r6     eth01 00-D0-56-F2-B5-11  
  
-MAC Address Error: Found duplicate MAC addresses at subnet 100.1.12.0/24, host router r12:  
Node Interface      MAC_Address  
sr3    eth01 00-D0-56-F2-B5-03  
sr4    eth01 00-D0-56-F2-B5-03  
sw15   eth01 00-D0-56-F2-B5-03  
r12    eth01 00-D0-56-F2-B5-03  
  
-MAC Address Error: Found duplicate MAC addresses at link between routers with link ID: 44  
Node  Interface      MAC_Address  
r6    eth02           07-D0-56-F2-B5-03  
r7    eth03           07-D0-56-F2-B5-03  
  
-MAC Address Error: Found duplicate MAC addresses at link between routers with link ID: 48  
Node  Interface      MAC_Address  
r12   eth02           12-D0-56-F2-B5-02  
r13   eth02           12-D0-56-F2-B5-02  
  
-MAC Address Error: Found duplicate MAC addresses at link between routers with link ID: 54  
Node  Interface      MAC_Address  
r14   eth03           14-D0-56-F2-B5-03  
r17   eth02           14-D0-56-F2-B5-03  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.6.5. Έξοδος εκτέλεσης 4, τοπολογία 4.

Παρατηρήσεις – σχόλια

Η έξοδος του προγράμματος μας δείχνει ότι εντόπισε ίδιες MAC διευθύνσεις σε interfaces κόμβων μέσα σε φυσικά υπό-δίκτυα αλλά και σε interfaces που ανήκουν στα άκρα ενός link μεταξύ δύο δρομολογητών. Συγκεκριμένα εντόπισε ίδιες MAC διευθύνσεις σε interfaces κόμβων που ανήκουν σε 3 διαφορετικά υπό-δίκτυα του δικτύου, στο 10.8.1.0/24 με host router το r1, στο 10.8.6.0/24 με host router το r6 και στο 100.1.12.0/24 με host router το r12. Επίσης εντόπισε διπλότυπες MAC σε τρία διαφορετικά link μεταξύ δρομολογητών, μεταξύ των interfaces των κόμβων r6, r7 στο link με ID 44, στη συνέχεια μεταξύ των Interfaces των r12, r13 στο link με ID 48 και τέλος στα interfaces των κόμβων r14, r17 στο link με ID 54

Εκτέλεση 5

Περιγραφή εκτέλεσης

Η πέμπτη εκτέλεση στην τοπολογία 4 έχει στηθεί με τις κατάλληλες παραμέτρους – configurations ώστε να μην είναι ορθή, βάση των επαληθεύσεων που είναι σχεδιασμένο το πρόγραμμα να πραγματοποιεί. Τα λάθη αυτά γίνονται κατά την διαδικασία επαλήθευσης των IP υπό-δικτύων, όπου ορίζουμε διπλότυπα IP υπό-δίκτυα. Συγκεκριμένα:

- Ορίζουμε διπλότυπο υπό-δίκτυο μεταξύ των δρομολογητών r3, r14 και r10, r10 με διεύθυνση 100.1.22.0/24
- Ορίζουμε διπλότυπο υπό-δίκτυο μεταξύ των δρομολογητών r7, r15 και r13, r16 με διεύθυνση 100.1.21.0/24

Έξοδος προγράμματος

```
----- Running network topology: 'topology_4' -----  
  
Device configuration completed!  
  
Subnet error: Found duplicate subnets between routers:  
Router_ID      Address  
  r3 r14 100.1.22.0/24  
  r7 r15 100.1.21.0/24  
 r10 r17 100.1.22.0/24  
 r13 r16 100.1.21.0/24  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.6.6. Έξοδος εκτέλεσης 5, τοπολογία 4.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι εντοπίστηκαν διπλότυπα υπό-δίκτυα μεταξύ δρομολογητών. Συγκεκριμένα το υπό-δίκτυο 100.1.22.0/24 που είναι μεταξύ των δρομολογητών r3,r14 και r10, r17 και το υπό-δίκτυο 100.1.21.0/24 που είναι μεταξύ των δρομολογητών r7, r15 και r13, r16. Αφού το πρόγραμμα εντόπισε ορθώς τα λάθη τερματίζει ανεπιτυχώς τη λειτουργία του δίχως να συνεχίσει την επαλήθευση δικτύου.

Εκτέλεση 6

Περιγραφή εκτέλεσης

Η έκτη εκτέλεση πάνω στην τοπολογία 4 έχει στηθεί με τις κατάλληλες παραμέτρους – configurations ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη. Σκοπός είναι να ελέγξει αν οι IP διευθύνσεις των interface που ανήκουν σε ένα link μεταξύ δρομολογητών, ανήκουν στο ίδιο υπό-δίκτυο. Συγκεκριμένα:

- Ορίζουμε IP διευθύνσεις στα interface των r10, r17 ώστε να μην ανήκουν στο ίδιο υπό-δίκτυο
- Ορίζουμε IP διευθύνσεις στα interface των r14, r16 ώστε να μην ανήκουν στο ίδιο υπό-δίκτυο
- Ορίζουμε IP διευθύνσεις στα interface των r12, r13 ώστε να μην ανήκουν στο ίδιο υπό-δίκτυο
- Ορίζουμε IP διευθύνσεις στα interface των r8, r10 ώστε να μην ανήκουν στο ίδιο υπό-δίκτυο

Έξοδος προγράμματος

```
----- Running network topology: 'topology_4' -----  
  
Device configuration completed!  
  
-Router link IP Error:  
  Nodes: r8, interface: eth02, IP: 100.1.9.8/24 and  
  Nodes: r10, interface eth01, IP: 100.1.8.1/24 at link with ID: '45'  
  does not belong to the same subnet!  
  
-Router link IP Error:  
  Nodes: r10, interface: eth03, IP: 100.1.22.8/24 and  
  Nodes: r17, interface eth01, IP: 100.2.22.1/24 at link with ID: '51'  
  does not belong to the same subnet!  
  
-Router link IP Error:  
  Nodes: r12, interface: eth02, IP: 101.1.14.8/24 and  
  Nodes: r13, interface eth02, IP: 100.1.14.1/24 at link with ID: '48'  
  does not belong to the same subnet!  
  
-Router link IP Error:  
  Nodes: r14, interface: eth02, IP: 100.1.36.1/24 and  
  Nodes: r16, interface eth02, IP: 100.1.30.2/24 at link with ID: '53'  
  does not belong to the same subnet!  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.6.7. Έξοδος εκτέλεσης 6 , τοπολογία 4.

Παρατηρήσεις - σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι εντοπίστηκαν διευθύνσεις IP στα interfaces των link μεταξύ των δρομολογητών που δεν ανήκουν στο ίδιο υπό-δίκτυο. Συγκεκριμένα αυτά βρέθηκαν στο link με ID 45 ανάμεσα στους δρομολογητές r8, r10, στο link με ID 51 ανάμεσα στους δρομολογητές r10, r17, στο link με ID 48 ανάμεσα στους δρομολογητές r12, r13 και στο link με ID 53 ανάμεσα στους δρομολογητές r14, r16. Για την επικοινωνία μεταξύ των δρομολογητών χρειάζεται οι δύο άκρες του να ανήκουν στο ίδιο υπό-δίκτυο. Τα παραπάνω λάθη έχουν ως αποτέλεσμα την μη επιτυχή επικοινωνία μεταξύ των δρομολογητών του δικτύου. Αφού το πρόγραμμα εντόπισε ορθώς τα λάθη όπως τα ορίσαμε, τερματίζει ανεπιτυχώς δίχως να συνεχίσει την επαλήθευση δικτύου.

Εκτέλεση 7

Περιγραφή εκτέλεσης

Η έβδομη εκτέλεση πάνω στην τοπολογία 4 έχει στηθεί με τις κατάλληλες παραμέτρους - configurations ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη. Σκοπός είναι να ελέγξει αν οι IP διευθύνσεις των interface που ανήκουν σε ένα link μεταξύ δρομολογητών, είναι διπλότυπες ή όχι. Συγκεκριμένα:

- Ορίζουμε διπλότυπη διεύθυνση IP στα interface που βρίσκονται στα άκρα του link με ID: 44 μεταξύ των κόμβων r6, r7
- Ορίζουμε διπλότυπη διεύθυνση IP στα interface που βρίσκονται στα άκρα του link με ID: 48 μεταξύ των κόμβων r12, r13
- Ορίζουμε διπλότυπη διεύθυνση IP στα interface που βρίσκονται στα άκρα του link με ID: 55 μεταξύ των κόμβων r14 r18
- Ορίζουμε διπλότυπη διεύθυνση IP στα interface που βρίσκονται στα άκρα του link με ID: 58 μεταξύ των κόμβων r15, r18

Έξοδος προγράμματος

```
----- Running network topology: 'topology_4' -----  
  
Device configuration completed!  
  
-IP Address Error: Found duplicate IP address between router connections:  
  Connection: ('r6', 'r7'), link ID: 44  
  r6 interface: eth02, IP: 100.1.6.1  
  r7 interface: eth03, IP: 100.1.6.1  
  
-IP Address Error: Found duplicate IP address between router connections:  
  Connection: ('r12', 'r13'), link ID: 48  
  r12 interface: eth02, IP: 100.1.14.1  
  r13 interface: eth02, IP: 100.1.14.1  
  
-IP Address Error: Found duplicate IP address between router connections:  
  Connection: ('r14', 'r18'), link ID: 55  
  r14 interface: eth04, IP: 100.1.32.1  
  r18 interface: eth01, IP: 100.1.32.1  
  
-IP Address Error: Found duplicate IP address between router connections:  
  Connection: ('r15', 'r18'), link ID: 58  
  r15 interface: eth04, IP: 100.1.42.1  
  r18 interface: eth02, IP: 100.1.42.1  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.6.8. Έξοδος εκτέλεσης 7, τοπολογία 4.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι εντόπισε διπλότυπες IP διευθύνσεις στα interface που βρίσκονται στα άκρα των Link μεταξύ δρομολογητών. Συγκεκριμένα εντόπισε διπλότυπη διεύθυνση στο link με ID 44 μεταξύ των r6 - r7, στο link με ID 48 μεταξύ των r12 - r13, στο link με ID 55 μεταξύ των r14 - r18 και τέλος στο link με ID 58 μεταξύ των r15 - r18. Μετά τον ορθό εντοπισμό των λαθών που ορίσαμε, το πρόγραμμα τερματίζει ανεπιτυχώς την επαλήθευση δικτύου.

Εκτέλεση 8

Περιγραφή εκτέλεσης

Η όγδοη εκτέλεση πάνω στην τοπολογία 4 έχει στηθεί με τις κατάλληλες παραμέτρους – configurations ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη. Σκοπός είναι να ελέγξει αν οι IP διευθύνσεις των interface των συσκευών που ανήκουν σε ένα υπό-δίκτυο είναι διπλότυπες ή όχι.

Συγκεκριμένα:

- Ορίζουμε διπλότυπη διεύθυνση IP στους κόμβους pc1, pc3 και r1 στο υπό-δίκτυο 10.8.1.0/24 στον δρομολογητή r1
- Ορίζουμε διπλότυπη διεύθυνση IP στους κόμβους pc14, pc16 και pc17 στο υπό-δίκτυο 10.8.8.0/24 στο δρομολογητή r8
- Ορίζουμε διπλότυπη διεύθυνση IP στους κόμβους sr1 και sr2 στο υπό-δίκτυο 100.1.11.0/24

Έξοδος προγράμματος

```
----- Running network topology: 'topology_4' -----  
  
Device configuration completed!  
  
-IP Address Error: Found duplicate IP addresses at subnet:10.8.1.0/24 with host router: r1  
Node Interface IP_Address  
pc1      eth01  10.8.1.1  
pc3      eth01  10.8.1.1  
r1       eth01  10.8.1.1  
  
-IP Address Error: Found duplicate IP addresses at subnet:10.8.8.0/24 with host router: r8  
Node Interface IP_Address  
pc14     eth01  10.8.8.14  
pc17     eth01  10.8.8.14  
pc16     eth01  10.8.8.14  
  
-IP Address Error: Found duplicate IP addresses at subnet:100.1.11.0/24 with host router: r11  
Node Interface IP_Address  
sr1      eth01  100.1.11.10  
sr2      eth01  100.1.11.10  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.6.9. Έξοδος εκτέλεσης 8, τοπολογία 4.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι εντοπίστηκαν διπλότυπες IP διευθύνσεις στα interface των κόμβων μέσα σε υπό-δίκτυα. Συγκεκριμένα βρέθηκαν διπλότυπες διευθύνσεις μεταξύ των κόμβων pc1, pc2 και r1 στο υπό-δίκτυο 10.8.1.0/24 που βρίσκεται στον δρομολογητή r1, μεταξύ των κόμβων pc14, pc16 και pc17 στο υπό-δίκτυο 10.8.8.0/24 που βρίσκεται στον δρομολογητή r8 και στους sr1, sr2 που βρίσκονται στο υπό-δίκτυο 100.1.11.0/24 που βρίσκονται στον δρομολογητή r11. Το πρόγραμμα ορθώς εντόπισε τα λάθη έτσι όπως τα ορίσαμε και τερμάτισε ανεπιτυχώς την επαλήθευση δικτύου

Εκτέλεση 9

Περιγραφή εκτέλεσης

Η ένατη εκτέλεση πάνω στην τοπολογία 4 έχει στηθεί με τις κατάλληλες παραμέτρους – configurations ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη. Σκοπός είναι να ελέγξει αν το gateway που είναι ορισμένο στο interface μιας συσκευής που ανήκει μέσα σε ένα υπό-δίκτυο, είναι ορθό για το υπό-δίκτυο στο οποίο ανήκει. Συγκεκριμένα:

- Ορίζουμε εσφαλμένη gateway διεύθυνση στους κόμβους sr3 και sr4 στο υπό-δίκτυο 100.1.12.0/24 που βρίσκεται στο r12
- Ορίζουμε εσφαλμένη gateway διεύθυνση στους κόμβους pc18 και pc15 στο υπό-δίκτυο 10.8.8.0/24 που βρίσκεται στο r8
- Ορίζουμε εσφαλμένη gateway διεύθυνση στους κόμβους pc9 και pc10 στο υπό-δίκτυο 10.8.5.0/24 που βρίσκεται στο r5

Έξοδος προγράμματος

```
----- Running network topology: 'topology_4' -----  
Device configuration completed!  
  
-Gateway error: Invalid gateway at node 'pc9', interface: 'eth01', subnet: 10.8.5.0/24, host router: r5  
-Subnets Gateway: '10.8.5.1',  
-pc9 gateway '10.8.5.10'  
  
-Gateway error: Invalid gateway at node 'pc10', interface: 'eth01', subnet: 10.8.5.0/24, host router: r5  
-Subnets Gateway: '10.8.5.1',  
-pc10 gateway '10.8.5.2'  
  
-Gateway error: Invalid gateway at node 'pc15', interface: 'eth01', subnet: 10.8.8.0/24, host router: r8  
-Subnets Gateway: '10.8.8.1',  
-pc15 gateway '10.8.8.15'  
  
-Gateway error: Invalid gateway at node 'pc18', interface: 'eth01', subnet: 10.8.8.0/24, host router: r8  
-Subnets Gateway: '10.8.8.1',  
-pc18 gateway '10.8.8.16'  
  
-Gateway error: Invalid gateway at node 'sr4', interface: 'eth01', subnet: 100.1.12.0/24, host router: r12  
-Subnets Gateway: '100.1.12.1',  
-sr4 gateway '100.1.12.2'  
  
-Gateway error: Invalid gateway at node 'sr3', interface: 'eth01', subnet: 100.1.12.0/24, host router: r12  
-Subnets Gateway: '100.1.12.1',  
-sr3 gateway '100.1.12.100'  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.6.10. Έξοδος εκτέλεσης 9, τοπολογία 4.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι εντοπίστηκαν εσφαλμένες διευθύνσεις gateway στα interface των κόμβων pc9 και pc10 του υπό-δικτύου

10.8.5.0/24 που βρίσκεται στο r5, των κόμβων pc15 και pc18 που βρίσκονται στο υπό-δίκτυο 10.8.8.0/24 που βρίσκεται στο r8 και τέλος, στα interface των κόμβων sr3 και sr4 που βρίσκονται στο υπό-δίκτυο 100.1.12.0/24 στο r12. Τα λάθη αυτά στα gateway των συσκευών έχουν το αποτέλεσμα να μην μπορούν να επικοινωνήσουν με το υπόλοιπο δίκτυο. Αφού το πρόγραμμα ορθώς εντοπίζει τα λάθη έτσι όπως τα ορίσαμε, τερματίζει ανεπιτυχώς την επαλήθευση δικτύου.

Εκτέλεση 10

Περιγραφή εκτέλεσης

Η δέκατη εκτέλεση πάνω στην τοπολογία 4 έχει στηθεί με τις κατάλληλες παραμέτρους – configurations ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη. Σκοπός είναι να ελέγξει αν η IP διεύθυνση που είναι ορισμένη στο interface μιας συσκευής που ανήκει μέσα σε ένα υπό-δίκτυο, είναι ορθή για το υπό-δίκτυο στο οποίο ανήκει. Συγκεκριμένα:

- Ορίζουμε εσφαλμένη IP διεύθυνση στα interface των pc19 και pc20 που είναι συνδεδεμένα στο υπό-δίκτυο 10.8.9.0/24 του router r9
- Ορίζουμε εσφαλμένη διεύθυνση IP στα interface των pc11 και pc12 που είναι συνδεδεμένα στο υπό-δίκτυο 10.8.6.0/24 του router r6
- Ορίζουμε εσφαλμένη διεύθυνση IP στα interface των sr1 και sr2 που είναι συνδεδεμένα στο υπό-δίκτυο 100.1.11.0/24 του router r11

Έξοδος προγράμματος

```
----- Running network topology: 'topology_4' -----  
Device configuration completed!  
  
-IP Address Error:  
  Nodes 'pc12', interface: 'eth01', IP Address: '10.8.3.12' , does not belong to its subnet '10.8.6.0/24', host router: r6  
  
-IP Address Error:  
  Nodes 'pc11', interface: 'eth01', IP Address: '10.8.8.11' , does not belong to its subnet '10.8.6.0/24', host router: r6  
  
-IP Address Error:  
  Nodes 'pc20', interface: 'eth01', IP Address: '10.89.9.20' , does not belong to its subnet '10.8.9.0/24', host router: r9  
  
-IP Address Error:  
  Nodes 'pc19', interface: 'eth01', IP Address: '10.8.10.19' , does not belong to its subnet '10.8.9.0/24', host router: r9  
  
-IP Address Error:  
  Nodes 'sr1', interface: 'eth01', IP Address: '100.1.1.10' , does not belong to its subnet '100.1.11.0/24', host router: r11  
  
-IP Address Error:  
  Nodes 'sr2', interface: 'eth01', IP Address: '100.1.12.20' , does not belong to its subnet '100.1.11.0/24', host router: r11  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.6.11. Έξοδος εκτέλεσης 10, τοπολογία 4.

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι εντοπίστηκαν διευθύνσεις IP στα interface συσκευών οι οποίες δεν ανήκουν στο υπό-δίκτυο στο οποίο είναι συνδεδεμένες. Συγκεκριμένα, οι διευθύνσεις στα interfaces των pc11 και pc12 δεν ανήκουν στο υπό-δίκτυο 10.8.6.0/24 του r6 που είναι συνδεδεμένα, οι διευθύνσεις των pc19 και pc 20 δεν ανήκουν στο 10.8.9.0/24 του r9 που είναι συνδεδεμένα και τέλος οι διευθύνσεις των sr1 και sr2 δεν ανήκουν στο υπό-δίκτυο 100.1.11.0/24 του r11 που είναι συνδεδεμένα. Αφού το πρόγραμμα ορθώς εντοπίζει τα λάθη όπως τα ορίσαμε, τερματίζει ανεπιτυχώς δίχως να συνεχίσει με την επαλήθευση δικτύου.

Εκτέλεση 11

Περιγραφή εκτέλεσης

Η ενδέκατη εκτέλεση πάνω στην τοπολογία 4 έχει στηθεί με τις κατάλληλες παραμέτρους – configurations ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί να εντοπίσει αυτά τα λάθη. Σκοπός είναι να επαληθεύσει τις διαδρομές που ορίζονται από τους πίνακες δρομολόγησης των δρομολογητών. Συγκεκριμένα:

- Ορίζουμε στον πίνακα δρομολόγησης του r1 ως next hop του default gateway IP ενός interface με το οποίο δεν είναι συνδεδεμένο.

Έξοδος προγράμματος

```
----- Running network topology: 'topology_4' -----  
  
Device configuration completed!  
  
Subnet validation completed!  
  
Routing table validation Error:  
-Found error at router connections: Router r1 has no connection with an interface with IP addresss 100.1.1.2  
Route: From: '100.1.1.8/24', to: '100.1.2.8/24'  
  
Routing table validation Error:  
-Found error at router connections: Router r1 has no connection with an interface with IP addresss 100.1.1.2  
Route: From: '100.1.1.8/24', to: '100.1.4.8/24'  
  
Routing table validation Error:  
-Found error at router connections: Router r1 has no connection with an interface with IP addresss 100.1.1.2  
Route: From: '100.1.1.8/24', to: '100.1.5.8/24'  
  
Routing table validation Error:  
-Found error at router connections: Router r1 has no connection with an interface with IP addresss 100.1.1.2  
Route: From: '100.1.1.8/24', to: '100.1.6.8/24'  
  
Routing table validation Error:  
-Found error at router connections: Router r1 has no connection with an interface with IP addresss 100.1.1.2  
Route: From: '100.1.1.8/24', to: '100.1.8.8/24'  
  
Routing table validation Error:  
-Found error at router connections: Router r1 has no connection with an interface with IP addresss 100.1.1.2  
Route: From: '100.1.1.8/24', to: '100.1.9.8/24'  
  
Routing table validation Error:  
-Found error at router connections: Router r1 has no connection with an interface with IP addresss 100.1.1.2  
Route: From: '100.1.1.8/24', to: '100.1.11.1/24'  
  
Routing table validation Error:  
-Found error at router connections: Router r1 has no connection with an interface with IP addresss 100.1.1.2  
Route: From: '100.1.1.8/24', to: '100.1.12.1/24'  
  
Found network configuration errors. Network validation has ended!
```

Εικόνα 5.6.12. Έξοδος εκτέλεσης 11, τοπολογία 4

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος βλέπουμε ότι εντοπίστηκε σφάλμα στις συζεύξεις του δρομολογητή r1. Συγκεκριμένα στη προσπάθεια για επικοινωνία με τα υπόλοιπα υπό-δίκτυα το r1 στέλνει την κίνηση του προς το default gateway του. Το default gateway του όμως είναι λάθος ρυθμισμένο και δεν έχει σύνδεση με αυτό, με αποτέλεσμα να μην μπορεί να επικοινωνήσει με το υπόλοιπο δίκτυο.

Εκτέλεση 12

Περιγραφή εκτέλεσης

Η ενδέκατη εκτέλεση πάνω στην τοπολογία 4 έχει στηθεί με τις κατάλληλες παραμέτρους – configurations ώστε να μην είναι ορθή και το πρόγραμμα να προσπαθεί

να εντοπίσει αυτά τα λάθη. Σκοπός είναι να επαληθεύσει τις διαδρομές που ορίζονται από τους πίνακες δρομολόγησης των δρομολογητών.

Συγκεκριμένα:

- Ορίζουμε στον πίνακα δρομολόγησης του r7 next hop τέτοια ώστε όποια κίνηση έχει εισερχομένη από οπουδήποτε να πηγαίνει στο r15.

Έξοδος προγράμματος

```
----- Running network topology: 'topology_4' -----  
  
Device configuration completed!  
  
Subnet validation completed!  
  
Routing table validation Error:  
-Found loop in routing path from: '100.1.1.8/24', to: '100.1.4.8/24'  
-Node Router Path: -> r1 -> r3 -> r14 -> r18 -> r15 -> r7 -> r15  
  
Routing table validation Error:  
-Found loop in routing path from: '100.1.1.8/24', to: '100.1.5.8/24'  
-Node Router Path: -> r1 -> r3 -> r14 -> r18 -> r15 -> r7 -> r15  
  
Routing table validation Error:  
-Found loop in routing path from: '100.1.1.8/24', to: '100.1.6.8/24'  
-Node Router Path: -> r1 -> r3 -> r14 -> r18 -> r15 -> r7 -> r15  
  
Routing table validation Error:  
-Found loop in routing path from: '100.1.2.8/24', to: '100.1.4.8/24'  
-Node Router Path: -> r2 -> r3 -> r14 -> r18 -> r15 -> r7 -> r15  
  
Routing table validation Error:  
-Found loop in routing path from: '100.1.2.8/24', to: '100.1.5.8/24'  
-Node Router Path: -> r2 -> r3 -> r14 -> r18 -> r15 -> r7 -> r15  
  
Routing table validation Error:  
-Found loop in routing path from: '100.1.2.8/24', to: '100.1.6.8/24'  
-Node Router Path: -> r2 -> r3 -> r14 -> r18 -> r15 -> r7 -> r15  
  
Routing table validation Error:  
-Found loop in routing path from: '100.1.4.8/24', to: '100.1.5.8/24'  
-Node Router Path: -> r4 -> r7 -> r15 -> r7  
  
Routing table validation Error:  
-Found loop in routing path from: '100.1.4.8/24', to: '100.1.6.8/24'  
-Node Router Path: -> r4 -> r7 -> r15 -> r7
```

Εικόνα 5.6.13. Έξοδος εκτέλεσης 12, τοπολογία 4, Φωτογραφία 1/3

```

Routing table validation Error:
-Found loop in routing path from: '100.1.5.8/24', to: '100.1.4.8/24'
-Node Router Path: -> r5 -> r7 -> r15 -> r7

Routing table validation Error:
-Found loop in routing path from: '100.1.5.8/24', to: '100.1.6.8/24'
-Node Router Path: -> r5 -> r7 -> r15 -> r7

Routing table validation Error:
-Found loop in routing path from: '100.1.6.8/24', to: '100.1.4.8/24'
-Node Router Path: -> r6 -> r7 -> r15 -> r7

Routing table validation Error:
-Found loop in routing path from: '100.1.6.8/24', to: '100.1.5.8/24'
-Node Router Path: -> r6 -> r7 -> r15 -> r7

Routing table validation Error:
-Found loop in routing path from: '100.1.8.8/24', to: '100.1.4.8/24'
-Node Router Path: -> r8 -> r10 -> r17 -> r18 -> r15 -> r7 -> r15

Routing table validation Error:
-Found loop in routing path from: '100.1.8.8/24', to: '100.1.5.8/24'
-Node Router Path: -> r8 -> r10 -> r17 -> r18 -> r15 -> r7 -> r15

Routing table validation Error:
-Found loop in routing path from: '100.1.8.8/24', to: '100.1.6.8/24'
-Node Router Path: -> r8 -> r10 -> r17 -> r18 -> r15 -> r7 -> r15

Routing table validation Error:
-Found loop in routing path from: '100.1.9.8/24', to: '100.1.4.8/24'
-Node Router Path: -> r9 -> r10 -> r17 -> r18 -> r15 -> r7 -> r15

Routing table validation Error:
-Found loop in routing path from: '100.1.9.8/24', to: '100.1.5.8/24'
-Node Router Path: -> r9 -> r10 -> r17 -> r18 -> r15 -> r7 -> r15

Routing table validation Error:
-Found loop in routing path from: '100.1.9.8/24', to: '100.1.6.8/24'
-Node Router Path: -> r9 -> r10 -> r17 -> r18 -> r15 -> r7 -> r15

```

Εικόνα 5.6.14.. Έξοδος εκτέλεσης 12, τοπολογία 4, Φωτογραφία 2/3

```
Routing table validation Error:
-Found loop in routing path from: '100.1.11.1/24', to: '100.1.4.8/24'
-Node Router Path: -> r11 -> r13 -> r16 -> r18 -> r15 -> r7 -> r15

Routing table validation Error:
-Found loop in routing path from: '100.1.11.1/24', to: '100.1.5.8/24'
-Node Router Path: -> r11 -> r13 -> r16 -> r18 -> r15 -> r7 -> r15

Routing table validation Error:
-Found loop in routing path from: '100.1.11.1/24', to: '100.1.6.8/24'
-Node Router Path: -> r11 -> r13 -> r16 -> r18 -> r15 -> r7 -> r15

Routing table validation Error:
-Found loop in routing path from: '100.1.12.1/24', to: '100.1.4.8/24'
-Node Router Path: -> r12 -> r13 -> r16 -> r18 -> r15 -> r7 -> r15

Routing table validation Error:
-Found loop in routing path from: '100.1.12.1/24', to: '100.1.5.8/24'
-Node Router Path: -> r12 -> r13 -> r16 -> r18 -> r15 -> r7 -> r15

Routing table validation Error:
-Found loop in routing path from: '100.1.12.1/24', to: '100.1.6.8/24'
-Node Router Path: -> r12 -> r13 -> r16 -> r18 -> r15 -> r7 -> r15

Found network configuration errors. Network validation has ended!
```

Εικόνα 5.6.15. Έξοδος εκτέλεσης 12, τοπολογία 4, Φωτογραφία 3/3

Παρατηρήσεις – σχόλια

Από την έξοδο του προγράμματος, βλέπουμε πληθώρα από λούπες που έχουν εντοπιστεί στις διαδρομές. Αυτό συνέβη γιατί ορίσαμε στον πίνακα δρομολόγησης του r7 να στέλνει όλη την κίνηση του προς το r15. Αποτέλεσμα αυτού κανένα από τα υπόλοιπα υπό-δίκτυα να μην μπορεί να επικοινωνήσει με τα υπό-δίκτυα που βρίσκονται στους δρομολογητές r4, r5, r6

Κεφάλαιο 6. Επίλογος

Τελειώνοντας, είδαμε λοιπόν ότι τα δίκτυα των υπολογιστών έχουν περάσει και έχουν γίνει ένα αναπόσπαστο ζωτικό κομμάτι της κοινωνίας. Η σημαντικότητα τους φέρνει την επιτακτική ανάγκη της αξιοπιστίας τους και της ασφάλειας τους. Είδαμε τους λόγους που μπορεί να οδηγήσουν σε downtime και τις σημαντικές επιπτώσεις που μπορεί να προκαλέσει στον πραγματικό κόσμο.

Σήμερα για την εξασφάλιση και τη συνολική διαχείριση τους, αναπτύσσονται όλο και περισσότερα εργαλεία και διαδικασίες αυτοματισμού και επαλήθευσης που εφαρμόζονται στην αγορά. Είδαμε ότι επίσημα, το Network validation είναι ένα σχετικά νέο πεδίο με έλλειψη σε αυστηρότητα ορισμών και διαδικασιών, παρόλα αυτά είναι μία αρκετά ενεργή κοινότητα που όλο και μεγαλώνει και οδηγεί το network validation σε πιο επίσημα μονοπάτια.

Τέλος, παρουσιάσαμε το δικό μας εργαλείο επαλήθευσης δικτύου. Το εργαλείο αυτό αποτελεί μία πρώιμη μορφή όπου πραγματοποιεί οπτικοποίηση και βασικούς ελέγχους επαλήθευσης παραμέτρων δικτύων. Λειτουργίες όπως η υποστήριξη πραγματικών πρωτοκόλλων δικτύων και η δυνατότητα προσομοίωσης με τη χρήση εικονικών μηχανών είναι το επόμενο βήμα για την μετατροπή του εργαλείου σε ένα ολοκληρωμένο λογισμικό προσομοίωσης δικτύων.

Κάπου εδώ λοιπόν, ολοκληρώνεται η αναφορά μας για την αυτόματη παραμετροποίηση και επαλήθευση παραμέτρων δικτύων νέας γενιάς, ελπίζοντας να παρουσιάσαμε ικανοποιητικά το πεδίο και το εργαλείο μας αυτό να αποτελέσει την αρχή για ένα ολοκληρωμένο λογισμικό προσομοίωσης δικτύων.

Σας ευχαριστούμε για την προσοχή σας!

Κεφάλαιο 7. Βιβλιογραφία

- [1] «What is network outage,» TechTarget, [Ηλεκτρονικό]. Available: <https://www.techtarget.com/searchnetworking/definition/network-downtime>.
- [2] «Causes of Network Downtime and How to Avoid Them,» axiatp, [Ηλεκτρονικό]. Available: <https://www.axiatp.com/causes-of-network-downtime/>.
- [3] «More details about the October 4 outage,» Meta, [Ηλεκτρονικό]. Available: <https://engineering.fb.com/2021/10/05/networking-traffic/outage-details/>.
- [4] «Facebook's outage cost the company nearly \$100 million in revenue,» [Ηλεκτρονικό]. Available: <https://fortune.com/2021/10/04/facebook-outage-cost-revenue-instagram-whatsapp-not-working-stock/>.
- [5] «British Airways owner loses £170m in value after IT meltdown,» The Guardian, [Ηλεκτρονικό]. Available: <https://www.theguardian.com/business/2017/may/30/british-airways-ba-owner-drops-value-it-meltdown>.
- [6] «Cloudflare outage on July 17, 2020,» cloudflare, [Ηλεκτρονικό]. Available: <https://blog.cloudflare.com/cloudflare-outage-on-july-17-2020/>.
- [7] «What is network automation,» Juniper Networks, [Ηλεκτρονικό]. Available: <https://www.juniper.net/us/en/research-topics/what-is-network-automation.html>.
- [8] «What is network automation,» TechTarget, [Ηλεκτρονικό]. Available: <https://www.techtarget.com/searchnetworking/definition/network-automation>.
- [9] «Network Validation. What does Network Validation mean,» Medium, [Ηλεκτρονικό]. Available: <https://medium.com/the-elegant-network/network->

validation-831bcde03699.

- [10] «The what, when, and how of network validation,» Intentionet, [Ηλεκτρονικό]. Available: <https://www.intentionet.com/blog/the-what-when-and-how-of-network-validation/>.
- [11] «Designing a Network Validation Pipeline,» Intentionet, [Ηλεκτρονικό]. Available: <https://www.intentionet.com/blog/designing-a-network-validation-pipeline/>.
- [12] «Learning Ansible basics,» RedHat, [Ηλεκτρονικό]. Available: <https://www.redhat.com/en/topics/automation/learning-ansible-tutorial>.
- [13] «Getting Started with GNS3,» [Ηλεκτρονικό]. Available: <https://docs.gns3.com/docs/>.
- [14] «An Introduction to Batfish,» [Ηλεκτρονικό]. Available: <https://www.packetcoders.io/batfish-the-what-and-the-why/>.
- [15] «Suzieq,» [Ηλεκτρονικό]. Available: <https://www.stardustsystems.net/documentation/>.
- [16] «EVE-NG Community Cookbook,» [Ηλεκτρονικό]. Available: <https://www.eve-ng.net/index.php/documentation/community-cookbook/>.
- [17] «Cisco Packet Tracer,» [Ηλεκτρονικό]. Available: https://www.cisco.com/c/dam/en_us/training-events/netacad/course_catalog/docs/Cisco_PacketTracer_DS.pdf.
- [18] «Getting Started with pyATS,» [Ηλεκτρονικό]. Available: <https://developer.cisco.com/docs/pyats-getting-started/>.
- [19] «Trivial Graph Format,» Wikipedia, [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Trivial_Graph_Format.
- [20] «Networking Config Version 2,» [Ηλεκτρονικό]. Available: <https://cloudinit.readthedocs.io/en/latest/topics/network-config-format-v2.html>.
- [21] «NetworkX - Network Analysis in Python,» [Ηλεκτρονικό]. Available: <https://networkx.org/>.
- [22] «Interactive network visualizations - pyvis,» [Ηλεκτρονικό]. Available: <https://pyvis.readthedocs.io/en/latest/#>.
- [23] «pandas - Python Data Analysis Library,» [Ηλεκτρονικό]. Available: <https://pandas.pydata.org/>.

