



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Smart Healthcare: Σχέδιο εκφόρτωσης RL-Based εργασίας για Edge-Enabled δίκτυα αισθητήρων

Ιωάννης Κυριαζής
icsd18107@aegean.gr
321/2018107

Δίκτυα Αισθητήρων
Δεκέμβριος, 2022

Περίληψη

Η χρήση κόμβων αισθητήρων ή συσκευών Internet-of-Medical-Things (IoMTs), οι οποίοι είναι εξοπλισμένοι με αισθητήρες, είναι ευρεία. Τεράστιες ποσότητες δεδομένων από πολλές έξυπνες εφαρμογές υγειονομικής περίθαλψης συλλέγονται χρησιμοποιώντας αυτούς τους δικτυωμένους αισθητήρες και τα δεδομένα που προκύπτουν χρησιμοποιούνται για να βοηθήσουν στη λήψη αποφάσεων. Μία αποτελεσματική πλατφόρμα για τη συλλογή δεδομένων αισθητήρων είναι το Edge Computing, το οποίο προσφέρει τους απαραίτητους υπολογιστικούς πόρους. Στο μεταξύ, η εστίαση έχει στραφεί στην έξυπνη και ακριβή διαχείριση των πόρων που παρέχεται από την Τεχνητή Νοημοσύνη (Artificial Intelligence - AI), ιδιαίτερα στα συστήματα υγειονομικής περίθαλψης. Η υπολογιστική ταχύτητα και η εμβέλεια των συσκευών υγειονομικής περίθαλψης που βασίζονται στο IoMT θα βελτιωθούν σημαντικά με τη βοήθεια της τεχνητής νοημοσύνης. Ωστόσο, το δύσκολο πρόβλημα με αυτά τα φορητά gadget που απαιτούν ενέργεια, με μικρή μπαταρία και μη ανεκτικότητα σε καθυστερήσεις είναι τα ανεπαρκή και αναποτελεσματικά παραδοσιακά πρότυπα δίκαιης κατανομής πόρων. Προκειμένου να μειωθεί η καθυστέρηση και η κατανάλωση ενέργειας, αυτή η μελέτη προτείνει την προσέγγιση Υπολογιστική Εκφόρτωση με Χρήση Ενισχυτικής Μάθησης (Computation Offloading Using Reinforcement Learning - CORL). Η έλλειψη περιορισμένης χωρητικότητας της μπαταρίας και οι περιορισμοί που αφορούν την καθυστέρηση της υπηρεσίας ικανοποιούνται διατυπώνοντας πρώτα το ζήτημα ως μια πρόκληση που συνδυάζει καθυστέρηση (latency) και μείωση του ενεργειακού κόστους. Επιπλέον, η προτεινόμενη μέθοδος αναζητά τον καλύτερο κόμβο με διαθέσιμους πόρους για την εκφόρτωση εργασίας, προκειμένου να εξισορροπηθεί η καθυστέρηση και η κατανάλωση ενέργειας. Τα πειραματικά ευρήματα καταδεικνύουν τα πλεονεκτήματα του προτεινόμενου σχήματος όσον αφορά την εξοικονόμηση ενέργειας, τη μειωμένη καθυστέρηση και τη βέλτιστη χρήση πόρων του κόμβου σε edge-enabled δίκτυα αισθητήρων. Για να επικυρωθεί το προτεινόμενο σύστημα κάτω από εύλογες υποθέσεις, οι ερευνητές που συνέγραψαν την παρούσα μελέτη χρησιμοποίησαν έναν προσομοιωτή iFogSim (επιτρέπει τη μοντελοποίηση και την προσομοίωση του fog computing για την αξιολόγηση των πολιτικών διαχείρισης πόρων και προγραμματισμού σε πόρους edge και cloud υπό διαφορετικά σενάρια).

Κατάλογος Περιεχομένων

Κεφάλαιο 1	Εισαγωγή.....	4
Κεφάλαιο 2	Σχετική Δουλειά.....	6
2.1	Με βάση την Κινητικότητα (Mobility Based).....	6
2.2	Με βάση την Εκφόρτωση (Offloading Based).....	6
Κεφάλαιο 3	Αρχιτεκτονική και Μοντελοποίηση του Συστήματος.....	8
3.1	Μοντέλο Συστήματος.....	8
3.2	Περιορισμοί.....	9
3.3	Διατύπωση του Προβλήματος.....	9
Κεφάλαιο 4	Υπολογιστική Εκφόρτωση με Χρήση Ενισχυτικής Μάθησης (CORL).....	10
4.1	Αλγοριθμική Λύση.....	10
4.2	Q-Learning.....	12
Κεφάλαιο 5	Αξιολόγηση Απόδοσης.....	13
5.1	Ρύθμιση Πειράματος.....	13
5.2	Ανάλυση Αποτελεσμάτων.....	13
Κεφάλαιο 6	Συμπέρασμα.....	14
	Βιβλιογραφία / Πηγές Πληροφόρησης.....	15

Κεφάλαιο 1. Εισαγωγή

Προκειμένου να δημιουργηθεί ένα βολικό, εξατομικευμένο και αποτελεσματικό σύστημα υγειονομικής περίθαλψης, η έξυπνη υγειονομική περίθαλψη αναφέρεται στη χρήση των IoMTs ή κόμβων αισθητήρων (Sensor Nodes - SN), του cloud computing, του edge computing και της τεχνητής νοημοσύνης (AI). Η χρήση εφαρμογών ιατρικής περίθαλψης σε φορητές συσκευές ή κινητά τηλέφωνα για παρακολούθηση της υγείας σε πραγματικό χρόνο καθίσταται δυνατή χάρη σε τέτοιες τεχνολογίες αιχμής. Οι συσκευές IoMT διαθέτουν αισθητήρες που παράγουν πολλά δεδομένα, τα οποία στη συνέχεια αποστέλλονται μέσω της υποδομής του Διαδικτύου σε μια κεντροποιημένη (centralized) τοποθεσία επεξεργασίας (συνήθως στο cloud) για ανάλυση. Στο τέλος, τα δεδομένα υγείας επιστρέφονται στους τελικούς χρήστες (όπως γιατρούς, νοσηλευτές και ασθενείς) για πρόσθετη ανάλυση. Μπορεί να χρησιμοποιηθεί με την τεχνητή νοημοσύνη για να επιλέξει διάφορες θεραπείες και να κάνει προληπτικούς ελέγχους υγείας και έγκαιρη ανίχνευση ασθενειών. Ωστόσο, η υψηλή καθυστέρηση, έλλειψη της ευελιξίας και η αδυναμία προσαρμογής στην κινητικότητα του τελικού χρήστη είναι μειονεκτήματα για το κεντροποιημένο κέντρο δεδομένων cloud. Μια ιεραρχικά κατανομημένη αρχιτεκτονική υπολογιστών που συνδέεται με τους τελικούς χρήστες και το cloud αναφέρεται ως το παράδειγμα του edge computing. Το περιβάλλον αιχμής του δικτύου αιχμής επεκτείνει τις υπηρεσίες που βασίζονται στο cloud για εφαρμογές IoMT. Παρέχει μια πλατφόρμα και υποδομή για μια ποικιλία υπηρεσιών λογισμικού. Οι μικροί υπολογιστικοί κόμβοι αναφέρονται ως κόμβοι ακμών (Edge Nodes - EN) στο Edge Computing και περιλαμβάνουν συσκευές όπως το Raspberry Pi, smartphone, πίνακες UDOO κ.λπ. Αυτά τα EN έχουν μια ποικιλία υπολογιστικών δυνατοτήτων για την επεξεργασία των δεδομένων αισθητήρων που παράγονται από τα SN. Λαμβάνοντας υπόψη την κινητικότητα του τελικού χρήστη, το edge computing προσφέρει ικανότητα επεξεργασίας κοντά στα SN. Επιτρέπει την έγκαιρη παροχή υπηρεσιών και μειώνει τη συμφόρηση του δικτύου. Λόγω του φυσικού τους μεγέθους και της ικανότητάς τους να αναπτύσσονται σε όλη την άκρη του δικτύου, τα EN, σε αντίθεση με το Cloud Computing, διαθέτουν περιορισμένες δυνατότητες επεξεργασίας και περιορισμένη χωρητικότητα μπαταρίας. Για να μειωθεί η συνολική κατανάλωση ενέργειας των SN και να επιμηκυνθεί η διάρκεια ζωής της μπαταρίας τους, απαιτείται μια προσέγγιση διαχείρισης ενέργειας με δυνατότητα AI. Η μετατόπιση φόρτου απαιτητικών εργασιών σε υποδομές υψηλότερου επιπέδου που μπορούν να προσφέρουν μεγαλύτερους υπολογιστικούς πόρους είναι η νεότερη επιλογή που έχει ληφθεί υπόψη. Το ερώτημα για το πώς, τότε και ποιες απαιτητικές εργασίες θα πρέπει να μεταφερθούν από τους SN στους EN ή στο cloud ήταν ένα εξέχον επίκεντρο πολλών προηγούμενων ερευνών. Αυτό το έγγραφο κάνει την υπόθεση ότι τα SN μπορούν να μεταφέρουν οποιαδήποτε μορφή επιβάρυνσης στο EN χωρίς να λάβουν υπόψη κατηγορίες φόρτου εργασίας (απαιτητικές εργασίες ή εργασίες μεγάλης καθυστέρησης). Μπορεί να είναι δύσκολο να βρεθεί μια στρατηγική εκφόρτωσης εργασίας με δυνατότητα τεχνητής νοημοσύνης που να λαμβάνει υπόψη τόσο την αποδοτική εκφόρτωση ως προς την καθυστέρηση όσο και τον ενεργειακό περιορισμό των SN που τροφοδοτούνται από μπαταρίες. Μετατοπίζοντας μέρος του φόρτου εργασίας τους σε edge ή cloud κόμβους κατά τη διάρκεια πολυάσχολων περιόδων, ο SN μπορεί να ανακουφίσει ορισμένες από τις τοπικές καταπονήσεις και να αποφύγει μια κατάσταση υπερφόρτωσης των περιορισμένων υπολογιστικών τους δυνατοτήτων. Όταν πολλές εργασίες υπόκεινται επεξεργασία σε SN και εκφορτώνουν τις εργασίες τους ταυτόχρονα, ο SN μπορεί επίσης να βελτιστοποιήσει την αντιστάθμιση καθυστέρησης-ενέργειας. Για την από κοινού βελτίωση της συνολικής καθυστέρησης και της χρήσης ενέργειας των SN, σε αυτό το έγγραφο παρουσιάζεται μία μέθοδος CORL. Επίσης, μελετώντας την κατανάλωση ενέργειας κατά την επικοινωνία, την κατανάλωση ενέργειας κατά την εκτέλεση υπολογισμών, την καθυστέρηση κατά την διάρκεια υπολογισμών και την καθυστέρηση επικοινωνίας, εξετάζονται τρόποι βελτιστοποίησης της καθυστέρησης και της κατανάλωσης ενέργειας του περιβάλλοντος δικτύου αισθητήρων με δυνατότητα αιχμής στην περίοδο υψηλής συμφόρησης. Στη συνέχεια διατυπώνεται το ζήτημα της μείωσης της καθυστέρησης και της ενέργειας με end-to-end καθυστέρηση και περιορισμένη ενεργειακή χωρητικότητα. Επιπλέον, παρουσιάζεται η τεχνική Υπολογιστικής Εκφόρτωσης με Χρήση Ενισχυτικής Μάθησης (Computation Offloading Using Reinforcement Learning - CORL), η οποία αναζητά έναν αποτελεσματικό κόμβο για την επεξεργασία διεργασιών που έχουν εκφορτωθεί. Τα οφέλη του

προτεινόμενου σχεδίου περιλαμβάνουν τη μεγιστοποίηση της εξοικονόμησης ενέργειας και την ελαχιστοποίηση της καθυστέρησης. Τέλος, προκειμένου να αποδειχθεί η αποτελεσματικότητα του προτεινόμενου σχήματος CORL, παρουσιάζονται αποτελέσματα προσομοίωσης. Αναλύονται επίσης το κόστος εκτέλεσης, η κατανάλωση ενέργειας και η συνολική καθυστέρηση.

Κεφάλαιο 2. Σχετική Δουλειά

Η ερευνητική κοινότητα είναι αρκετά ενεργή όσον αφορά το IoMT, το οποίο, παρόλο που βασίζεται στο edge computing, βρίσκεται ακόμα στα αρχικά του στάδια. Το Edge computing είναι μια πιθανή στρατηγική για την ανάπτυξη μιας πλατφόρμας προσαρμοστικών λειτουργιών, η οποία επέτρεψε υψηλή διαχειρισσιμότητα του IoMT και μείωσε την καθυστέρηση υπηρεσίας των εφαρμογών IoMT. Με τη βοήθεια έξυπνων συσκευών IoMT (όπως βραχιολάκια, wearables, smartwatches, κ.λπ.), το IoMT που βασίζεται στο cloud συγκεντρώνει ιατρικά δεδομένα σε πραγματικό χρόνο των τελικών χρηστών σε ένα μακρινό κέντρο δεδομένων. Από την άλλη πλευρά, το edge computing προσφέρει μια πιο ευέλικτη και αποτελεσματική πλατφόρμα που υποστηρίζει την επίγνωση του περιβάλλοντος, τη μειωμένη καθυστέρηση και την ενεργειακή απόδοση.

2.1: Με βάση την Κινητικότητα (Mobility Based)

Σε περιβάλλον Fog Computing, η κινητικότητα των εφαρμογών IoMT έχει σημαντικό αντίκτυπο στο Quality of Service (QoS) με τη μορφή καθυστέρησης εξυπηρέτησης. Προτάθηκε ένα σχέδιο μετεγκατάστασης υπηρεσιών βασισμένο στη διαδικασία απόφασης Markov που έλαβε υπόψη το μοτίβο κινητικότητας των συσκευών IoT. Προκειμένου να βρεθούν οι ιδανικές ακολουθίες τοποθέτησης υπηρεσιών με βάση μια μέθοδο χρονικού παραθύρου (time window mechanism), προτάθηκε μια στρατηγική μετεγκατάστασης υπηρεσιών. Ωστόσο, η πλειονότητα των έργων βασίζεται σε πλασματικά μοντέλα και δεν απεικονίζουν με ακρίβεια την κατάσταση του τομέα της Υγείας σήμερα. Η τοποθεσία των τελικών χρηστών έχει επίσης μεγάλο αντίκτυπο στους κανόνες μετεγκατάστασης υπηρεσιών. Επομένως, κατά τη δημιουργία αποτελεσματικών σχεδίων μεταφοράς υπηρεσιών, αυτές οι περιοχές πρέπει να λαμβάνονται υπόψη. Οι στόχοι κατανομής πόρων και βελτιστοποίησης στο Fog Computing έχουν συζητηθεί σε ορισμένες μελέτες. Οι cloud κόμβοι, οι fog κόμβοι και οι συσκευές IoMT αποτελούν τα τρία επίπεδα της ιεραρχικής αρχιτεκτονικής που στηρίζει το Fog Computing. Η βελτιστοποίηση fog πόρων μπορεί να χωριστεί σε δύο κατηγορίες: βελτιστοποίηση πόρων σε fog κόμβους και βελτιστοποίηση πόρων σε κέντρα δεδομένων cloud. Για τη βελτιστοποίηση των πόρων του cloud, προτάθηκε ένας προσαρμοστικός ευρετικός αλγόριθμος κατωφλίου (threshold) (ένας ευρετικός αλγόριθμος είναι αυτός που έχει σχεδιαστεί για να λύνει ένα πρόβλημα με ταχύτερο και αποτελεσματικότερο τρόπο από τις παραδοσιακές μεθόδους θυσιάζοντας τη βέλτιστη, την ακρίβεια, την ακρίβεια ή την πληρότητα για την ταχύτητα). Εφαρμόζοντας ένα gradient descent regression μοντέλο (αλγόριθμος με τον οποίο επιτυγχάνονται προβλέψεις) και μια στατιστική ανάλυση των ιστορικών δεδομένων κατανάλωσης CPU, αυτή η προσέγγιση ανιχνεύει υπερφορτωμένους κεντρικούς υπολογιστές ορίζοντας ένα υψηλό δυναμικό όριο στο κέντρο δεδομένων cloud. Αυτοί οι αλγόριθμοι, ωστόσο, λαμβάνουν υπόψη αποκλειστικά τις ρυθμίσεις του cloud. Προτάθηκε μια τεχνική για την κατανομή πόρων που χρησιμοποιεί τον εγγύς αλγόριθμο (αλγόριθμος για την επίλυση ενός κυρτού προβλήματος βελτιστοποίησης που χρησιμοποιεί τους εγγύς τελεστές των αντικειμενικών όρων) για να μειώσει τη χρήση ενέργειας του κέντρου δεδομένων. Λόγω της έλλειψης ευαισθησίας του εγγύς αλγορίθμου στο μέγεθος του βήματος, της μέτριας ακρίβειας και της γρήγορης σύγκλισης (ένας επαναληπτικός αλγόριθμος λέγεται ότι συγκλίνει όταν καθώς προχωρούν οι επαναλήψεις η έξοδος πλησιάζει όλο και περισσότερο σε μια συγκεκριμένη τιμή), επιλέχθηκε αντί της διπλής αποσύνθεσης (ανάλυση ενός σύνθετου προβλήματος ή συστήματος και διάσπασή του σε μικρότερα μέρη που είναι πιο διαχειρίσιμα και κατανοητά).

2.2: Με βάση την Εκφόρτωση (Offloading Based)

Οι συσκευές IoMT περιλαμβάνουν αισθητήρες ή ενεργοποιητές που τους επιτρέπουν να συλλέγουν σημαντικό όγκο δεδομένων, τα οποία στη συνέχεια αποθηκεύονται, αναλύονται, χρησιμοποιούνται για τη λήψη απόφασης και κοινοποιούνται σε μια πλατφόρμα υγειονομικής περίθαλψης που φιλοξενείται στο cloud. Αντίθετα, τα δεδομένα σε μια πλατφόρμα υγειονομικής περίθαλψης που βασίζεται στο fog αξιολογούνται, κοινοποιούνται, υποβάλλονται σε επεξεργασία και στη συνέχεια αποθηκεύονται στο απομακρυσμένο κέντρο δεδομένων. Ένα σύστημα υγειονομικής περίθαλψης που βασίζεται στο fog αποδίδει καλύτερα από ένα σύστημα υγειονομικής περίθαλψης που βασίζεται στο cloud για διαδραστικές υπηρεσίες σε πραγματικό χρόνο. Προτάθηκε μια αρχιτεκτονική από cloud σε fog για παρακολούθηση U-healthcare για νοσοκομεία και έξυπνα σπίτια. Οι τελικοί χρήστες παρακολουθούνται στο σπίτι και

στα νοσοκομεία χρησιμοποιώντας συσκευές IoMT. Για επείγοντα περιστατικά, τα δεδομένα υποβάλλονται σε επεξεργασία, εξετάζονται και αποστέλλεται μήνυμα στον γιατρό. Ωστόσο, οι πλατφόρμες υγειονομικής περίθαλψης που βασίζονται στο fog έχουν προβλήματα καθυστέρησης και κινητικότητας. Προκειμένου να μειωθεί η κατανάλωση ενέργειας και η καθυστέρηση στο fog περιβάλλον, η υπολογιστική εκφόρτωση (μεταφορά απαιτητικών σε πόρους υπολογιστικών εργασιών σε ξεχωριστό επεξεργαστή ή σε μια εξωτερική πλατφόρμα) είναι απαραίτητη. Για τον σκοπό της βελτίωσης της καθυστέρησης υπηρεσίας και της κατανάλωσης ενέργειας κόμβου, ορισμένες μελέτες έχουν λάβει υπόψη και τις δύο παραμέτρους. Σε μια πλατφόρμα mobile-cloud, αναπτύχθηκαν ενεργειακά αποδοτικές στρατηγικές εκφόρτωσης για εργασίες διακωδικοποίησης (μετατρέπει ένα κωδικοποιημένο ψηφιακό αρχείο ή σύνολο αρχείων σε ένα τροποποιημένο σύνολο ψηφιακών αρχείων που ανταποκρίνονται καλύτερα στις ανάγκες του υλικού). Ο θεμελιώδης στόχος αυτής της προσπάθειας είναι η ικανοποίηση του περιορισμού της καθυστέρησης με ταυτόχρονη μείωση της συνολικής χρήσης ενέργειας. Για να επιλυθεί αυτό, προτάθηκε ένας διαδικτυακός (online) αλγόριθμος εκφόρτωσης εργασιών που καθορίζει εάν μια συγκεκριμένη εργασία πρέπει να γίνει τοπικά ή να μεταφερθεί σε ένα απομακρυσμένο cloud. Προτάθηκε μια προσέγγιση ευρετικής εκφόρτωσης εργασιών για μείωση του χρόνου ολοκλήρωσης της εργασίας σε fog περιβάλλον, ωστόσο, λήφθηκε υπόψη μόνο ένας fog κόμβος.

Κεφάλαιο 3. Αρχιτεκτονική και Μοντελοποίηση του Συστήματος

Το επίπεδο cloud, το επίπεδο edge και το επίπεδο κόμβου αισθητήρα είναι μερικά μόνο από τα τρία επίπεδα που συνθέτουν αυτήν την αρχιτεκτονική. Το ανώτερο στρώμα, που αναφέρεται ως στρώμα σύννεφο, αποτελείται από ετερογενείς πόρους που αποτελούν διακομιστές cloud. Στο χαμηλότερο επίπεδο αυτής της αρχιτεκτονικής δηλαδή το επίπεδο κόμβου αισθητήρα, λαμβάνονται υπόψη συσκευές IoMT και SN. Λόγω περιορισμών καθυστέρησης, ενέργειας και πόρων, αυτό το επίπεδο δημιουργεί τεράστιες ποσότητες δεδομένων, αλλά δεν μπορεί να τα χειριστεί. Το edge computing αποτελείται από όλα τα EN και πιστεύεται ότι είναι ένα ενδιάμεσο επίπεδο μεταξύ του cloud και του επιπέδου κόμβου αισθητήρα. Για να ικανοποιηθούν οι ανάγκες της εφαρμογής IoMT, οι κόμβοι ελέγχουν δεδομένα και διεργασίες στο επίπεδο edge μέσω αμφίδρομης επικοινωνίας μεταξύ τους. Οι πόροι αυτών των κόμβων είναι διαφορετικοί και τα EN ταξινομούνται ιεραρχικά από πάνω προς τα κάτω. Οι κόμβοι που βρίσκονται κοντά στο επίπεδο του κόμβου αισθητήρα έχουν λιγότερη δικτυακή, αποθηκευτική και υπολογιστική ισχύ από εκείνους που βρίσκονται κοντά στο επίπεδο cloud. Με παρόμοιους κόμβους ανερχόμενης ζεύξης και σχεδόν συγκρίσιμη καθυστέρηση ανερχόμενης ζεύξης, δύο ή περισσότεροι κόμβοι συνδέονται στο ίδιο επίπεδο.

3.1: Μοντέλο Συστήματος

Το μοντέλο αποτελείται από τρία επίπεδα για edge-enabled δίκτυα αισθητήρων που αποτελείται από ένα επίπεδο cloud, ένα επίπεδο edge και ένα επίπεδο αισθητήρα. Μια ομάδα EN χρησιμοποιείται για την παροχή υπολογιστικών πόρων και την ανακούφιση του συστήματος από υπερφόρτωση εκφορτώνοντας τις εργασίες όταν το SN κατακλύζεται από υπολογιστικές ανάγκες την στιγμή συμφόρησης. Σε ένα πλαίσιο edge-enabled δικτύου αισθητήρων, παρουσιάζεται μια αποτελεσματική μέθοδος για την ταυτόχρονη μείωση της καθυστέρησης και της χρήσης ενέργειας. Ορίζουμε το F ως μια συλλογή EN που αποτελείται από n κόμβους που είναι γεωγραφικά διασκορπισμένοι, συνδεδεμένοι μεταξύ τους και συνδεδεμένοι με τον cloud κόμβο C . Στο επίπεδο του κόμβου αισθητήρα, υπάρχει ένα σύνολο S , που δηλώνει μία συλλογή από l SNs (δηλ. έξυπνοι μετρητές, αισθητήρες, φορητές συσκευές παρακολούθησης της υγείας, δικτυωμένα οχήματα και ενεργοποιητές). Κάθε SN που συνδέεται με μια εργασία t_j και T είναι μια συλλογή όλων των εργασιών t_j όπου το j ανήκει στο T . Αυτά τα SNs χρησιμοποιούν μια ποικιλία εφαρμογών με ποικίλες απαιτήσεις σε πόρους επικοινωνίας και υπολογισμού. Κάθε SN συνδέεται με μια εργασία, η οποία αναπαρίσταται ως πλειάδα με τα γράμματα $\{D_j, C_j \text{ και } I_j\}$. Όπου D_j είναι το μέγεθος δεδομένων εργασίας, C_j είναι η ποσότητα των πόρων που απαιτούνται για τον υπολογισμό και I_j είναι η καθυστέρηση. Κάνουμε τις ακόλουθες υποθέσεις με βάση την ιδέα ότι ο όγκος των δεδομένων που υποβάλλονται σε επεξεργασία σε μη τυπικά μεγέθη θα πρέπει να κατανέμεται ομοιόμορφα. Δημιουργούμε μια δυαδική μεταβλητή που ονομάζεται $k_{i,j}$ για να αναπαραστήσουμε εάν μια εργασία από το SN_{s_j} μεταφέρεται στον i^{th} κόμβο.

- **Καθυστέρηση:** Η καθυστέρηση μετάδοσης και η καθυστέρηση επεξεργασίας συνθέτουν την καθυστέρηση σε ένα edge περιβάλλον. Η επεξεργασία εργασιών σε SN, EN ή στο cloud προκαλεί την υπολογιστική καθυστέρηση. Η ποσότητα των δεδομένων D_j που πρέπει να κοινοποιηθεί και ο ρυθμός δεδομένων της σύνδεσης $R_{i,j}$ μεταξύ του j^{th} SN και του επιλεγμένου i^{th} EN ή Cloud καθορίζουν την καθυστέρηση μετάδοσης. Το σύμβολο u υποδηλώνει τον αριθμό των hops που πρέπει να διανύσει μια εργασία SN προκειμένου να επεξεργαστεί δεδομένα στο σωστό κόμβο EN ή cloud. Η υπολογιστική καθυστέρηση ($L^{comp}_{i,j}$) μιας εργασίας t_j καθορίζεται από την αναλογία των απαιτήσεων επεξεργασίας της εργασίας προς τη συνολική υπολογιστική ικανότητα (C^{node}_i) του κόμβου. Η συνολική καθυστέρηση μιας εργασίας ($L^{total}_{i,j}$) υπολογίζεται προσθέτοντας την υπολογιστική καθυστέρηση και την καθυστέρηση μετάδοσης.
- **Κατανάλωση Ενέργειας:** Η ενέργεια που καταναλώνεται από τις συσκευές επικοινωνίας, όπως όταν μια εργασία SN εκφορτώνεται στο cloud ή έναν edge κόμβο για επεξεργασία και τα υπολογιστικά αποτελέσματα αποστέλλονται πίσω στις συσκευές SN προς την αντίστροφη κατεύθυνση, και η ενέργεια που καταναλώνεται από την CPU των κόμβων N ($N = \{S \cup F \cup C\}$), προσδιορίζει την κατανάλωση ενέργειας σε ένα περιβάλλον όπου χρησιμοποιούνται edge-enabled δίκτυα αισθητήρων. Ως αποτέλεσμα, το μοντέλο κατανάλωσης ενέργειας υπολογίζεται ως το σύνολο της ενέργειας που χρησιμοποιείται για υπολογισμούς ($E^{comp}_{i,j}$) και επικοινωνία ($E^{tran}_{i,j}$). Η ενέργεια που χρησιμοποιείται από όλους τους κόμβους που ανήκουν στο N καθορίζει τη συνολική κατανάλωση ενέργειας ($E^{total}_{i,j}$) σε ένα edge computing περιβάλλον.

3.2: Περιορισμοί

- Περιορισμός Ανάθεσης: Οι εργασίες που παράγονται από τα SN στο επίπεδο IoMT πρέπει να είναι σε σχέση ένα προς ένα με τους κόμβους που ανήκουν στο N. Υποδεικνύει ότι ένας μεμονωμένος κόμβος ολοκληρώνει κάθε εργασία. Για να προσδιοριστεί εάν η εργασία SN t_j εκτελείται στον i^{th} κόμβο που ανήκει στο N, ορίζεται μια δυαδική μεταβλητή Φ_{ij} . Αν $\Phi_{ij} = 1$, σημαίνει ότι στον i^{th} κόμβο που ανήκει στο N έχει δοθεί η εργασία SN t_j . Εάν $\Phi_{ij} = 0$, ωστόσο, σημαίνει ότι η εργασία SN t_j δεν έχει εκχωρηθεί στον i^{th} κόμβο. Κάθε εργασία SN t_j σε περιορισμό ανάθεσης εκχωρείται σε έναν κόμβο i^{th} κόμβο που ανήκει στο N.
- Προθεσμία Καθυστέρησης: Το L^{\max} είναι η υψηλότερη τιμή καθυστέρησης που μπορεί να ανεχθεί μια εργασία SN t_j . Το L^{\max} είναι το μέγιστο για όλες τις εργασίες SN t_j όπου το j ανήκει στο S. Προκειμένου να διασφαλιστεί ότι η εργασία t_j μπορεί να ολοκληρωθεί χωρίς πρόσθετη καθυστέρηση, θα πρέπει η καθυστέρηση της διεργασίας t_j να είναι μικρότερη ή ίση με την L^{\max} που αναφέρθηκε παραπάνω για κάθε j που ανήκει στο T.
- Περιορισμός Χωρητικότητας: Για να ολοκληρωθεί η εργασία που παρήγαγε το SN, θα πρέπει ο i^{th} κόμβος που ανήκει στο N να διαθέσει έναν συγκεκριμένο αριθμό πόρων δικτύου (N), αποθήκευσης (M) και υπολογισμού (C). Η συνολική ανάγκη πόρων από τον κόμβο δεν μπορεί να είναι μεγαλύτερη από τη χωρητικότητα πόρων του $\Psi_i(C, N, M)$. Η χωρητικότητα πόρων του i^{th} κόμβου που ανήκει στο N αντιπροσωπεύεται από $\Psi_i(C, N, M)$, ενώ η ζήτηση πόρων για την εργασία SN t_j αντιπροσωπεύεται από $\psi_j(C, N, M)$.

3.3: Διατύπωση του Προβλήματος

Ο στόχος αυτής της έρευνας είναι να μειωθεί το συνολικό κόστος C διατηρώντας παράλληλα μια ανεκτή καθυστέρηση. Η ευρετική μέθοδος μπορεί να χρησιμοποιηθεί για την επίλυση του C, ενός προηγμένου bin packing ζητήματος (αντικείμενα διαφορετικών μεγεθών πρέπει να συσκευάζονται σε έναν πεπερασμένο αριθμό κάδων ή δοχείων, καθένα από αυτά σταθερής χωρητικότητας, με τρόπο που να ελαχιστοποιεί τον αριθμό των κάδων που χρησιμοποιούνται) που είναι NP-hard (εάν ένας αλγόριθμος για την επίλυσή του μπορεί να μεταφραστεί σε έναν για την επίλυση οποιουδήποτε NP-problem (μη ντετερμινιστικού πολυωνυμικού χρόνου) προβλήματος). Οι υπάρχουσες ευρετικές τεχνικές, ωστόσο, παρουσιάζουν αστάθεια στο πραγματικό περιβάλλον του δικτύου και είναι ανεπαρκείς για τον χειρισμό πολύπλοκων ζητημάτων που εμποδίζουν τη γρήγορη λήψη αποφάσεων. Το $E^{\text{total}}_{i,j}$ και το $L^{\text{total}}_{i,j}$ σε αυτό το πρόβλημα είναι η συνολική κατανάλωση ενέργειας και η καθυστέρηση της εργασίας t_j , αντίστοιχα. Ρυθμίζοντας ένα κουμπί δ με τιμές 0 και 1, που είναι το βάρος της κατανάλωσης ενέργειας, μπορεί να ερμηνευθεί η αντιστάθμιση μεταξύ της κατανάλωσης ενέργειας και της καθυστέρησης. Ένας περιορισμός C_1 εγγυάται ότι η απαιτούμενη μέγιστη καθυστέρηση L^{\max} περιορίζεται από τη συνολική καθυστέρηση της εργασίας SN t_j . Ένας δεύτερος περιορισμός C_2 εγγυάται ότι οι διαθέσιμοι πόροι του i^{th} κόμβου που ανήκει στο N είναι αρκετοί για την επεξεργασία των εργασιών SN ή ότι δεν γίνεται υπέρβαση των διαθέσιμων πόρων κανενός κόμβου μετά την εκφόρτωση της εργασίας. Η σχέση ένας προς έναν μεταξύ της εργασίας και του κόμβου διασφαλίζεται από τον περιορισμό C_3 . Ο περιορισμός C_4 εγγυάται ότι η συνολική κατανάλωση ενέργειας των συσκευών SN και των edge κόμβων δεν υπερβαίνει τη συνολική χωρητικότητα της μπαταρίας τους (E^{\max}).

Κεφάλαιο 4. Υπολογιστική Εκφόρτωση με Χρήση Ενισχυτικής Μάθησης (CORL)

4.1: Αλγοριθμική Ανάλυση

Υποθετικά, στο προτεινόμενο σχήμα CORL, το επίπεδο cloud είναι υπεύθυνο για τον υπολογισμό του συνολικού κόστους εκφόρτωσης από τον χώρο πιθανών καταστάσεων (S) και από τις ενέργειες (A) για τη δημιουργία ενός πίνακα με χρήση ενισχυτικής μάθησης. Μετά, οι SNs χρησιμοποιούν αυτόν τον πίνακα για να αποφασίσουν εάν θα εκφορτώσουν μια εισερχόμενη εργασία. Στην ενισχυτική μάθηση, ο agent συγκεντρώνει την κατάσταση του συστήματος s_t για κάθε χρονικό βήμα T_t και εκτιμά το reward για το προηγούμενο χρονικό βήμα r_{t-1} . Στη συνέχεια, ο agent επιλέγει μια ενέργεια a_t σύμφωνα με το προκαθορισμένο σχέδιο. Το σύστημα μεταβαίνει στη νέα κατάσταση s_{t+1} στο επόμενο χρονικό βήμα μετά την ανάληψη δράσης. Παρόμοια με αυτό, ο agent υπολογίζει το reward r_t και επιλέγει μια νέα ενέργεια a_{t+1} σύμφωνα με την κατάσταση s_{t+1} .

- **State Space:** Η κατάσταση συστήματος σε αυτήν τη μελέτη καλύπτει όλα τα χαρακτηριστικά πόρων του ίδιου του SN, του edge κόμβου, του cloud και των απαιτήσεων πόρων της εργασίας. Αυτά τα χαρακτηριστικά περιλαμβάνουν τη διαθέσιμη ενεργειακή χωρητικότητα της μπαταρίας, την CPU και το εύρος ζώνης. Η κατάσταση συστήματος συμβολίζεται με τον τύπο $s_t = (R_i, \psi_j(C, M, N))$, όπου το R_i υποδηλώνει τα χαρακτηριστικά των κόμβων για κάθε i που ανήκει στο N και το $\psi_j(C, M, N)$ δηλώνει εκείνα της εργασίας t_j . $R_i = (b, e, r)$, όπου τα b , e και r αντιπροσωπεύουν εύρος ζώνης, επίπεδο μπαταρίας και χωρητικότητα CPU, αντίστοιχα. Οι ανάγκες σε πόρους μιας εργασίας αντιπροσωπεύονται από το $\psi_j(C, M, N)$. Υποθετικά, για λόγους απλότητας, ισχύει ότι ο χώρος κατάστασης αποτελείται από επεξεργαστική ισχύ και διαθέσιμο εύρος ζώνης. Ο χώρος κατάστασης για μια εισερχόμενη εργασία SN μπορεί να παρουσιαστεί ως εξής λαμβάνοντας υπόψη x διάφορους βαθμούς ικανότητας επεξεργασίας και y διαφορετικά επίπεδα διαθέσιμου εύρους ζώνης ($s_{x,y}(\tau)$) όπου ανήκει στο S εκφράζει την κατάσταση με y και x υποδηλώνουν το επίπεδο του διαθέσιμου εύρους ζώνης και το επίπεδο χωρητικότητας της CPU):

$$s_{x,y}(\tau) = \begin{bmatrix} s_{11}(\tau) & \dots & s_{1x}(\tau) \\ \vdots & \ddots & \vdots \\ s_{y1}(\tau) & \dots & s_{xy}(\tau) \end{bmatrix}$$

- **Action Space:** Με βάση την κατάσταση, ο SN προβαίνει σε ένα συγκεκριμένο σύνολο ενεργειών. Οι SNs μπορούν να κάνουν κρίσεις σχετικά με την εκφόρτωση με τέσσερις διαφορετικές ενέργειες. Το σύνολο $A = \{a_1, a_2, a_3, a_4\}$ μπορεί να χρησιμοποιηθεί για να αναπαραστήσει τον χώρο δράσης για το χειρισμό μιας εργασίας σε κάθε χρονική περίοδο απόφασης, όπου a_1 υποδηλώνει ότι ο SN θα περιμένει μια ευκαιρία για να χειριστεί την εργασία, a_2 ο SN θα εκτελέσει την εργασία τοπικά, a_3 αυτή η εργασία θα εκφορτωθεί σε έναν edge κόμβο και a_4 αυτή η εργασία θα εκφορτωθεί στο cloud. Έτσι, κατά τη διάρκεια της χρονικής περιόδου τ , το reward ορίζεται ως εξής:

$$r(\tau) = - \sum_{i \in N} \sum_{j \in T} \Phi_{ij} (\delta E_{i,j}^{total}(\tau) + (1 - \delta) L_{i,j}^{total}(\tau))$$

- **Reward:** Με βάση τις ενέργειες και την παρατήρηση της κατάστασής του, ένας SN λαμβάνει ένα reward. Στόχος αυτής της έρευνας είναι να μειωθεί το συνολικό κόστος με την πάροδο του χρόνου, επομένως η συνάρτηση κόστους και η συνάρτηση reward του προτεινόμενου μοντέλου θα πρέπει να συνδέονται. Υποθετικά λαμβάνεται υπόψη ότι υπάρχει ένας agent στο περιβάλλον όταν χρησιμοποιείται η ενισχυτική μάθηση. Ο agent είναι έτοιμος να λειτουργήσει και να λάβει αποφάσεις εντός της καθορισμένης χρονικής περιόδου τ . Ο agent δίνει στην ακολουθία με μια συγκεκριμένη κατάσταση $s(\tau)$ που ανήκει στο S , οδηγίες για να εκτελέσει μια

συγκεκριμένη ενέργεια $a(\tau)$ που ανήκει στο A . Η κατάσταση του συστήματος στη συνέχεια αλλάζει σε μια νέα κατάσταση $s(\tau+1)$ και ανταμείβεται ($r(\tau)$). Μια δυναμική διαδικασία Markov (Markov Dynamic Process - MDP) δημιουργείται από αυτή τη διαδικασία. Ο βασικός στόχος αυτής της διαδικασίας είναι να διατηρήσει μια πολιτική με παράλληλη αύξηση του συλλεγόμενου reward. Συγκεντρώνοντας όλα τα δεδομένα σχετικά με το reward και την κατάσταση κάθε μεμονωμένου SN για τις αποφάσεις εκφόρτωσης, το επίπεδο cloud στο CORL εκπαιδεύει το μοντέλο εκμάθησης. Τελευταίο αλλά εξίσου σημαντικό, το CORL αποθηκεύει αυτά τα συλλεγόμενα rewards (που επιτυγχάνονται από τον SN για δράση σύμφωνα με τον προβλεπόμενο χώρο καταστάσεων μετά την εφαρμογή της πολιτικής σε κάθε κατάσταση) σε έναν πίνακα Q . Αυτός ο πίνακας Q παρέχεται στον SN, ώστε να μπορεί να αποφασίσει εάν θα εκφορτώσει μια εργασία στο edge, στο επίπεδο cloud ή στον εαυτό του με βάση την τρέχουσα κατάσταση. Ο πίνακας Q ενημερώνεται συνεχώς από το επίπεδο cloud ανάλογα με τις SN ενέργειες εκφόρτωσης. Χρησιμοποιώντας τον πίνακα Q , ο SN επιλέγει μια πορεία δράσης που θα μεγιστοποιήσει τα μελλοντικά rewards. Χρησιμοποιώντας την παρακάτω συνάρτηση, το Q-Learning αναζητά τη βέλτιστη πορεία δράσης για να μεγιστοποιήσει το συνολικό reward για κάθε κατάσταση, όπου η ενημερωμένη τιμή για αυτήν την κατάσταση και ενέργεια εμφανίζεται με $Q(s(\tau), a(\tau))$ στην αριστερή πλευρά. Ενώ, τα $Q(s(\tau), a(\tau))$, $r(\tau)$, και $\max Q(s(\tau + 1), a(\tau + 1))$ υποδεικνύουν, με αυτή τη σειρά, την τρέχουσα τιμή Q , την τιμή του reward που ελήφθη και η μέγιστη τιμή του reward που αναμένεται στο μέλλον. Επιπλέον, γ είναι το discount rate και α είναι μια παράμετρος για το ποσοστό εκμάθησης.

$$Q(s(\tau), a(\tau)) \leftarrow Q(s(\tau), a(\tau)) + \alpha [r(\tau) + \gamma \max Q(s(\tau + 1), a(\tau + 1)) - Q(s(\tau), a(\tau))]$$

Στο παρακάτω στιγμιότυπο φαίνεται ολοκληρωμένος ο αλγόριθμος σε ψευδογλώσσα.

Algorithm 1: Computation Offloading Using Reinforcement Learning (CORL)

```

1 begin
2   Initialize  $Q(s(\tau), a(\tau)) = 0, \forall s \in \mathbf{S}, a \in \mathbf{A}$ ;
3   for (each episode) do
4     Initialize state  $s(0)$ ;
5     for ( $\tau = 1, K$ ) do
6       Generate random number  $\beta$ ;
7       if ( $\beta > \epsilon$ ) then
8         Set  $a(\tau) = \arg\max Q(s(\tau), a(\tau))$  ;
9       end
10      else
11        Set  $a(\tau)$  randomly;
12      end
13      for ( $s(\tau) \in \mathbf{S}$ ) do
14        Performer selected action  $a(\tau)$  ;
15        Obtain reward  $r(\tau)$  using equation (8);
16         $Q(s(\tau), a(\tau)) \leftarrow Q(s(\tau), a(\tau)) + \alpha [r(\tau) +$ 
17           $\gamma \max Q(s(\tau + 1), a(\tau + 1)) - Q(s(\tau), a(\tau))]$ ;
18         $s(\tau) \leftarrow s(\tau + 1)$  ;
19      end
20    end
21 end

```

4.2: Q-Learning

Σε αυτό το σχέδιο, όλοι οι πόροι και τα εισερχόμενα δεδομένα εργασιών από τους SN και τους EN συλλέγονται από το επίπεδο cloud. Καθορίζοντας την κατάσταση με βάση τα χαρακτηριστικά ανά κόμβο και την εργασία τους, το επίπεδο cloud δημιουργεί τον πίνακα Q χρησιμοποιώντας αυτά τα εισερχόμενα δεδομένα ως εκπαίδευση και υπολογίζει το reward $r(t)$ για τις ενέργειες $a(t)$ που γίνονται με την κατάσταση $s(t)$. Σύμφωνα με τις δραστηριότητες εκφόρτωσης εργασιών SN, κατασκευάζεται ο πίνακας Q. Η σχεδόν βέλτιστη πολιτική που θα εφαρμοστεί από το SN έχει χτιστεί χρησιμοποιώντας τη μεθοδολογία Q-Learning. Ο πίνακας Q ξεκινά με μια τιμή 0. Όταν λαμβάνεται μια SN εργασία, χρησιμοποιούμε την εξίσωση « $r(t) \Rightarrow$ » για να υπολογίσουμε το reward $r(t)$ για την τοπική εκφόρτωση της εργασίας στα επίπεδα EN και cloud. Η τελευταία εξίσωση που αναγράφεται στο κεφάλαιο 4.1, η οποία βασίζεται στα $s(t)$, $a(t)$, και $r(t)$, χρησιμοποιείται για την ανάπτυξη της πολιτικής $Q(s(t), a(t))$. Προκειμένου να δημιουργηθεί και να ενημερωθεί ένας πίνακας Q, έχει χρησιμοποιηθεί ο ρυθμός εκμάθησης $\alpha = 0,01$. Επιλέγεται μια ενέργεια $a(t)$ χρησιμοποιώντας την εξερεύνηση και την εκμετάλλευση αφού έχει κατασκευαστεί ο πίνακας Q. Η χρήση της ϵ -greedy (άπληστης) πολιτικής κατά την εκτέλεση της επιλεγμένης διεργασίας είναι μια τεχνική για την επίτευξη ισορροπίας μεταξύ εκμετάλλευσης και εξερεύνησης. Η εξίσωση « $r(t) \Rightarrow$ » όπως αναφέρθηκε παραπάνω χρησιμοποιείται για τον υπολογισμό του reward. Η εξισορρόπηση του φορτίου ϵ μεταξύ εκμετάλλευσης και εξερεύνησης πραγματοποιείται σε όλη τη φάση της διαδικασίας της εκπαίδευσης του αλγορίθμου, ακολουθούμενη από τη δημιουργία του πίνακα Q για τις εισερχόμενες διεργασίες. Το υψηλότερο προβλεπόμενο reward $r(t)$ στο μέλλον για τον SN για την εκτέλεση συγκεκριμένων ενεργειών σε μια συγκεκριμένη κατάσταση, αποθηκεύεται στον πίνακα Q που δημιουργήθηκε. Σύμφωνα με αυτόν τον πίνακα Q, όταν εμφανίζεται μια εισερχόμενη εργασία SN, το SN παρακολουθεί την τρέχουσα κατάσταση $s(t)$ και αποκρίνεται με το μεγαλύτερο αθροιστικό reward.

Κεφάλαιο 5. Αξιολόγηση Απόδοσης

5.1: Ρύθμιση Πειράματος

Για να αξιολογηθεί η αποτελεσματικότητα της προτεινόμενης στρατηγικής εκφόρτωσης εργασιών, πραγματοποιούνται πειράματα χρησιμοποιώντας το προτεινόμενο σχήμα στον προσομοιωτή iFogSim. Είναι ένα σύνολο εργαλείων που επιτρέπει τη μοντελοποίηση και την προσομοίωση edge computing περιβάλλοντος σε συνδυασμό με το cloud. Μπορεί να παρακολουθεί έναν αριθμό δεικτών απόδοσης, συμπεριλαμβανομένης της ενεργειακής κατανάλωσης, της καθυστέρησης, του χρόνου εκφόρτωσης κ.λπ. Τα πειράματα πραγματοποιήθηκαν με χρήση της αρχιτεκτονικής των τριών επιπέδων. Τα χαρακτηριστικά πόρων των κόμβων σε αυτήν την προσομοίωση ποικίλλουν ανάλογα με το επίπεδο. Οι ταχύτητες CPU των κόμβων ορίστηκαν από 1 έως 4 GHz στα διάφορα επίπεδα. Υπάρχει ένας διακομιστής cloud, δεκαπέντε ENs και δέκα έως σαράντα SNs. Το εύρος ζώνης μεταξύ όλων των διαφόρων επιπέδων χωρίζεται προκειμένου να προετοιμαστεί ο χώρος κατάστασης (από τους SNs στους ENs το εύρος ζώνης είναι 512Mbps και από τους ENs στο Cloud επίπεδο το εύρος ζώνης είναι 1024Mbps). Η καθυστέρηση από τους SNs στους ENs είναι 40ms, η καθυστέρηση από τους ENs στο Cloud επίπεδο είναι 75ms, το μέγεθος δεδομένων της SN εργασίας κυμαίνεται από 0,8 έως 1,3Mb, ο περιορισμός καθυστέρησης L^{\max} κυμαίνεται από 0,4 έως 0,8s, η υπολογιστική ενέργεια που καναλώνεται ανά μονάδα είναι 1,35J, η ενέργεια ανά μονάδα που καταναλώνεται για την επίτευξη επικοινωνίας είναι 0,7J, η μεταβλητή ϵ είναι 0,1, ο ρυθμός μάθησης είναι 0,01 και ο discount παράγοντας είναι 0,9. Στο παρόν πείραμα, κάθε κόμβος έλαβε μεταξύ 50 και 400 εισερχόμενες εργασίες SN. Η επεξεργασία εργασιών σταματά όταν ο χρόνος επεξεργασίας υπερβεί το μέγιστο όριο καθυστέρησης. Όταν έχουν ολοκληρωθεί όλες οι εργασίες του SN, το πείραμα φτάνει στο τέλος του.

5.2: Ανάλυση Αποτελεσμάτων

Αποδεικνύεται ότι το προτεινόμενο σύστημα χρησιμοποιεί σταθερά λιγότερη ενέργεια από τις πολιτικές Random, CloudEdge και Cloud. Αυτό είναι κατανοητό αφού το σχήμα CORL επιλέγει έναν αποτελεσματικό κόμβο για την εκτέλεση λειτουργιών SN. Το ενεργειακό κόστος αυξάνεται καθώς αυξάνεται ο αριθμός των εργασιών. Η μειωμένη χρήση ενέργειας για μια ποικιλία εργασιών δίνει στην προτεινόμενη μέθοδο μεγαλύτερη απόδοση από τις τρεις υπάρχουσες μεθόδους. Η συνολική καθυστέρηση αυξάνεται καθώς αυξάνεται ο υπολογιστικός φόρτος ή ο αριθμός των εργασιών. Λόγω της αποτελεσματικής επιλογής κόμβων και της βελτιστοποιημένης κατανάλωσης πόρων, παρατηρείται επίσης ότι το σχήμα CORL αποδίδει καλύτερα από τα σχήματα Random, CloudEdge και Cloud. Καθώς το μέγεθος του συνόλου των εργασιών SN αυξάνεται, οι συνολικές ενεργειακές δαπάνες όλων των σεναρίων αυξάνονται αυτόματα. Το αυξημένο μέγεθος δεδομένων προκαλεί μεγαλύτερη ισχύ επεξεργασίας και κατανάλωση ενέργειας, ωστόσο καθώς αυξάνεται το μέγεθος δεδομένων, η τιμή κατανάλωσης ενέργειας του σχήματος CloudEdge ισούται ουσιαστικά με αυτήν του προτεινόμενου σχήματος. Καθώς αυξάνεται η ποσότητα των εργασιών ή ο υπολογιστικός φόρτος, παρατηρείται ότι η συνολική καθυστέρηση εκφόρτωσης αυξάνεται. Ωστόσο, η απόδοση του προτεινόμενου σχήματος CORL είναι μεγαλύτερη από αυτή των σχημάτων Random, CloudEdge και Cloud. Η συνολική καθυστέρηση αυξάνεται καθώς αυξάνεται το σύνολο δεδομένων των εργασιών. Συγκρίνοντας το προτεινόμενο σχήμα με τις μεθόδους Random, CloudEdge και Cloud, διαπιστώνουμε ότι επιτυγχάνει μειωμένη καθυστέρηση. Η μέθοδος CORL εκφορτώνει αποτελεσματικά εργασίες για επεξεργασία, η οποία είναι η αιτία της παραπάνω σύγκρισης.

Κεφάλαιο 6. Συμπέρασμα

Προκειμένου να εκτελεστούν υπολογιστικές εργασίες σε ένα edge-enabled περιβάλλον δικτύου αισθητήρων με περιορισμούς καθυστέρησης και χωρητικότητας μπαταρίας από άκρο σε άκρο, μελετήθηκε η εκφόρτωση εργασιών SN με στόχο την μείωση της καθυστέρησης και της μείωση της κατανάλωσης ενέργειας. Προκειμένου να μειωθεί η καθυστέρηση και η χρήση ενέργειας κατά τη διάρκεια της εκφόρτωσης υπολογιστικής εργασίας, σχεδιάστηκε το πρόβλημα βελτιστοποίησης. Για να μειωθεί η καθυστέρηση και η κατανάλωση ενέργειας του κόμβου αισθητήρα, προτάθηκε ένα σύστημα CORL. Παρατηρήθηκε ότι μια στρατηγική που βασίζεται στην ενισχυτική μάθηση μπορεί να αποφασίσει πολύ γρήγορα εάν θα εκφορτωθεί η εργασία SN. Εν κατακλείδι, εφαρμόστηκε η ιδέα CORL χρησιμοποιώντας παραμέτρους σε πραγματικό χρόνο και περιβαλλοντικούς παράγοντες. Σε σύγκριση με τα βασικά συστήματα που εκμεταλλεύονται τον περιορισμό της χωρητικότητας της μπαταρίας, αποδεικνύεται ότι το προτεινόμενο σχήμα ελαχιστοποιεί αποτελεσματικά τη συνολική καθυστέρηση και τη χρήση ενέργειας.

Βιβλιογραφία / Πηγές Πληροφόρησης

Yadav, R., Zhang, W., Elgendy, I. A., Dong, G., Shafiq, M., Laghari, A. A., & Prakash, S. (2021). Smart healthcare: RL-based task offloading scheme for edge-enable sensor networks. *IEEE Sensors Journal*, 21(22), 24910-24918.

ΠΕΡΑΣ ΕΡΓΑΣΙΑΣ



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ
ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

UNIVERSITY OF THE AEGEAN
DEPARTMENT OF INFORMATION AND
COMMUNICATION SYSTEMS ENGINEERING

Kyriazis Ioannis

Copyright © 2022 – All Rights Reserved