NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCES
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

PROGRAM OF POSTGRADUATE STUDIES

PhD THESIS

Multi-layer IoT Resource Management

Kyriaki P. Panagidi

ATHENS

December 2019

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Πολυεπίπεδη διαχείριση πόρων σε δίκτυα IoT

Κυριακή Π. Παναγίδη

ΑΘΗΝΑ

Δεκέμβριος 2019

PhD THESIS

Multi-layer IoT Resource Management

Kyriaki P. Panagidi

SUPERVISOR: Eftathios Hadjiefthymiades, Professor Department of Informatics and Telecommunications, UoA

THREE-MEMBER ADVISORY COMMITTEE:

Eftathios Hadjiefthymiades, Professor UoA
Dimitris Varoutas, Associate Professor UoA
Lazaros Merakos, Professor UoA

SEVEN-MEMBER EXAMINATION COMMITTEE

(Signature) (Signature)

Eftathios Hadjiefthymiades,
Professor UoA

Dimitris Varoutas,
Associate Professor UoA

(Signature) (Signature)

Lazaros Merakos,
Professor UoA

Evangelos Zervas,
Professor UNIWA

(Signature) (Signature)

George Xylomenos,
Associate Professor AUEB

Kimon Kontovasilis,
Researcher A, NCSR "Demokritos"

(Signature)

Christos Douligeris,
Professor University of Piraeus

Examination Date 23/12/2019

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Πολυεπίπεδη διαχείριση πόρων σε δίκτυα IoT

Κυριακή Π. Παναγίδη

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Ευστάθιος Χατζηευθυμιάδης, Καθηγητής Τμήμα Πληροφορικής και Τηλεπικοινωνιών, ΕΚΠΑ

ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ:
Ευστάθιος Χατζηευθυμιάδης, Καθηγητής ΕΚΠΑ
Δημήτρης Βαρουτάς, Αναπληρωτής Καθηγητής ΕΚΠΑ
Λάζαρος Μεράκος, Καθηγητής ΕΚΠΑ

ΕΠΤΑΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

(Υπογραφή)

Ευστάθιος Χατζηευθυμιάδης,
Καθηγητής ΕΚΠΑ

(Υπογραφή)

Δημήτρης Βαρουτάς,
Αναπληρωτής Καθηγητής ΕΚΠΑ

(Υπογραφή)

Λάζαρος Μεράκος,
Καθηγητής ΕΚΠΑ

(Υπογραφή)

Ευάγγελος Ζέρβας,
Καθηγητής ΠΑΔΑ

(Υπογραφή)

Γεώργιος Ξυλωμένος,
Αναπληρωτής Καθηγητής ΟΠΑ

(Υπογραφή)

Κίμων Κοντοβασίλης,
Ερευνητής Α΄ ΕΚΕΦΕ Δημόκριτος

Χρήστος Δουληγέρης,
Καθηγητής ΠΑΠΕΙ

Ημερομηνία εξέτασης 23/12/2019

ABSTRACT

Internet of Things is one of the most promising paradigms in the current decade characterized by the use of smart and self-configured objects, like sensors, actuators, wearables etc., that are connected to a network and exchange data by sensing, reacting to events, and interacting with the environment. In addition unmanned mobile devices, e.g. drones, were introduced to users' daily life the last decade and become a part of the whole of "objects" participating in the IoT as long as they carry sensing equipment and on-board computing elements. These sensing and computational capabilities enhance the network embedded intelligence and allow complex tasks to be realized in a highly distributed fashion, thus, balancing load across the infrastructure and rendering communications much more energy efficient in newly introduced mobile Internet of Things. Their main characteristic of movement to space and time add a new degree of freedom to the needs of monitoring in IoT creating mobile IoT networks. All the IoT objects generate huge amount of data imposing a great demand on processing in order to transform the data into useful information or services.

In this new dynamic landscape, it is necessary to have an adequate architecture that can integrate heterogeneous information streams and provide services with an acceptable quality to the users. The realization of an IoT framework needs to take into account many constraints related to the device (power consumption, network processing, battery lifetime etc.), to the stochastic nature of the underlying network (delay, bandwidth utilization, latency) and to the middleware overlay that is necessary to fuse big volumes of information streams and deliver a service to the user. Therefore it is needed the use of a resource management architecture that can monitor the performance of the units involved in different layers in an IoT system and decides to take actions based on optimal resource use in order to support reliable delivery of information with an acceptable Quality of Service to the users.

This thesis proposes the design of a resource management framework which can monitor with no prior knowledge information streams produced by IoT devices, can predict changes with online mechanisms that can disrupt the performance of the IoT framework and can take actions to retain acceptable Quality Of Service while trying to save resources. The online, time optimized and distributed decision making models are based on Optimal Stopping Theory and Change Detection Theory. Starting from the edge we present our research of a dynamic encoder adaptive to changes in video sequences. This encoder proposes a compressing method in order to minimize the error produced to the transmitted multimedia sequence without harming the content. Afterwards, we study the performance of the underlying wireless network and how the changes in network environment can affect and disrupt the mission of unmanned robotic devices and the telemetry received. We study and analyze a time-optimized decision making model adaptive to network quality changes applied in unmanned vehicles as long as the communication between the devices and the fixed control stations can be obstructed, overloaded or can suffer from high packet loss rate due to network variations. Last but not least a stochastic optimization framework of on-line control unit applied on a data distributed platform is proposed because these platforms are necessary to IoT infrastructures to process the enormous volumes of data exchanged between IoT devices.

The findings of such decision making models are promising and solidly supportive to a vast spectrum of real-time and latency-sensitive applications with QoS requirements in IoT environments.

SUBJECT AREA: Optimal Stopping Theory applied on Decision-making in IoT

KEYWORDS: Real-time Decision Making, Mobile IoT, Optimal Stopping Theory, Scene Detection, Group-of-Pictures, Change-point Detection, Unmanned Vehicle, Distributed streaming platform, Prioritization

ΠΕΡΙΛΗΨΗ

Το Διαδίκτυο των πραγμάτων (Internet of Things - IoT) είναι ένα από τα πιο ελπιδοφόρα παραδείγματα της τρέχουσας δεκαετίας που χαρακτηρίζεται από τη χρήση έξυπνων και αυτόνομα διαμορφωμένων αντικειμένων, όπως αισθητήρες, actuators, wearables, κλπ., που συνδέονται με το Διαδίκτυο και ανταλλάσσουν δεδομένα με στόχο την έγκαιρη ανίχνευση συμβάντων, αντίδραση σε αυτά, και την αλληλεπίδραση τους με το περιβάλλον. Επιπλέον, μη επανδρωμένες κινητές συσκευές, π.χ. drones, εισήχθησαν στην καθημερινή ζωή των χρηστών την τελευταία δεκαετία και έγιναν μέρος του συνόλου των "αντικειμένων" που συμμετέχουν στο IoT εφόσον φέρουν εξοπλισμό ανίχνευσης και ενσωματωμένα στοιχεία υπολογιστών. Αυτές οι δυνατότητες ανίχνευσης και υπολογιστικής ικανότητας ενισχύουν την ενσωματωμένη στο δίκτυο ευφυΐα και επιτρέπουν την πραγματοποίηση σύνθετων εργασιών με πολύ κατανεμημένο τρόπο, εξισορροπώντας το φορτίο σε όλη την υποδομή και καθιστώντας τις επικοινωνίες πολύ πιο ενεργειακά αποδοτικές στο νεοεισαχθέν κινητό Διαδίκτυο των πραγμάτων. Το χαρακτηριστικό της κίνησης τους στο χώρο και στο χρόνο προσθέτει έναν νέο βαθμό ελευθερίας στις ανάγκες της παρακολούθησης στο IoT δημιουργώντας τα κινητά δίκτυα IoT. Όλα τα αντικείμενα IoT συμβάλουν στη δημιουργία ενός τεράστιου όγκου δεδομένων που απαιτούν σύνθετη επεξεργασία προκειμένου να μετατραπούν από δεδομένα σε χρήσιμες πληροφορίες ή υπηρεσίες.

Σε αυτό το νέο δυναμικό τοπίο, είναι απαραίτητο να υπάρχει μια επαρκής αρχιτεκτονική που να μπορεί να ενσωματώνει ετερογενείς ροές πληροφοριών και να παρέχει υπηρεσίες με αποδεκτή ποιότητα στους χρήστες. Η υλοποίηση ενός συστήματος IoT πρέπει να λαμβάνει υπόψη πολλούς περιορισμούς που σχετίζονται με τη συσκευή (κατανάλωση ρεύματος, επεξεργασία δικτύου, διάρκεια ζωής της μπαταρίας κ.λπ.), με το στοχαστικό χαρακτήρα του υποκείμενου δικτύου (καθυστέρηση, χρήση εύρους ζώνης, λανθάνουσα κατάσταση κ.λπ.) και το επικαλυπτόμενο midddleware που είναι απαραίτητο για τη συνένωση μεγάλων όγκων ροών πληροφοριών και την παροχή υπηρεσιών στους χρήστες. Ως εκ τούτου απαιτείται η χρήση αρχιτεκτονικής για τη διαχείριση των IoT πόρων η οποία να μπορεί να παρακολουθεί την απόδοση των μονάδων που συμμετέχουν σε διαφορετικά επίπεδα στο σύστημα και να αποφασίζει να αναλάβει δράσεις με βάση τη βέλτιστη χρήση των πόρων προκειμένου να υποστηριχθεί η αξιόπιστη παροχή πληροφοριών με αποδεκτή ποιότητα υπηρεσίας προς τους χρήστες.

Η διατριβή αυτή προτείνει το σχεδιασμό ενός πλαισίου διαχείρισης πόρων το οποίο μπορεί να παρακολουθεί χωρίς προηγούμενη γνώση πηγές πληροφοριών που παράγονται από συσκευές IoT, μπορεί να προβλέψει αλλαγές που διαταράσσουν την απόδοση του συστήματος και μπορεί να τις αντιμετωπίσει μέσω μηχανισμών απόφασης ώστε να διατηρήσει ένα επίπεδο αποδεκτής ποιότητας εξυπηρέτησης ενώ ταυτόχρονα εξοικονομεί πόρους. Τα προτεινόμενα κατανεμημένα μοντέλα λήψης αποφάσεων βασίζονται στη θεωρία βέλτιστης παύσης και στη θεωρία ανίχνευσης αλλαγών. Ξεκινώντας από την οπτική της συσκευής, παρουσιάζουμε την έρευνά μας για έναν δυναμικό κωδικοποιητή που προσαρμόζεται στις αλλαγές σε ακολουθίες πολυμεσικών ροών. Αυτός ο κωδικοποιητής προτείνει μια μέθοδο συμπίεσης προκειμένου να ελαχιστοποιηθεί το σφάλμα που παράγεται στη μεταδιδόμενη αλληλουχία πολυμέσων χωρίς να βλάπτεται το περιεχόμενο. Στη συνέχεια, μελετάμε τις επιδόσεις του υποκείμενου ασύρματου δικτύου και πώς οι αλλαγές στο περιβάλλον δικτύου μπορούν

να επηρεάσουν και να διαταράξουν την αποστολή μη επανδρωμένων συσκευών και την τηλεμετρία που μεταδίδουν. Μελετάμε και αναλύουμε ένα μοντέλο λήψης απόφασης βελτιστοποιημένο κατά το χρόνο που προσαρμόζεται στις αλλαγές ποιότητας του δικτύου και εφαρμόζεται στα μη επανδρωμένα οχήματα, εφόσον η επικοινωνία μεταξύ των συσκευών και των σταθερών σταθμών ελέγχου μπορεί συχνά να παρεμποδιστεί, να υπερφορτωθεί ή να υποστεί υψηλό ποσοστό απώλειας πακέτων λόγω παραλλαγών του δικτύου. Τέλος, προτείνεται ένα στοχαστικό πλαίσιο βελτιστοποίησης της μονάδας ελέγχου σε πραγματικό χρόνο το οποίο εφαρμόζεται σε μια κατανεμημένη πλατφόρμα δεδομένων, καθώς οι κατανεμημένες πλατφόρμες διάδοσης δεδομένων είναι απαραίτητες στις υποδομές IoT για τη διαχείριση και την επεξεργασία των τεράστιων όγκων δεδομένων που ανταλλάσσονται μεταξύ συσκευών IoT.

Τα ευρήματα των προτεινόμενων μοντέλων λήψης αποφάσεων σε αυτή τη διδακτορική διατριβή είναι πολλά υποσχόμενα και υποστηρίζουν σταθερά ένα ευρύ φάσμα εφαρμογών ευαίσθητων σε καθυστερήσεις με απαιτήσεις ποιότητας υπηρεσίας σε περιβάλλοντα IoT.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Λήψη Αποφάσεων Βέλτιστης Παύσης σε δίκτυα IoT

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Δυναμική Λήψη Απόφασης σε πραγματικό χρόνο, Ανίχνευση Αλλαγής, Θεωρία Βέλτιστης Παύσης, Διαδίκτυο των Πραγμάτων, Κινητό Διαδίκτυο των Πραγμάτων, Ομάδα Πλαισίων, Μη επανδρωμένα οχήματα, Κατανεμημένη Πλατφόρμα μηνυμάτων, Απόδοση Προτεραιότητας

ΣΥΝΟΠΤΙΚΗ ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΔΙΔΑΚΤΟΡΙΚΗΣ ΔΙΑΤΡΙΒΗΣ

Το Διαδίκτυο των Πραγμάτων (Internet of Things - IoT) είναι ένα από τα πλέον ελπιδοφόρα παραδείγματα σήμερα το οποίο χαρακτηρίζεται από τη χρήση έξυπνων συσκευών όπως αισθητήρες, κινητά τηλέφωνα, RFIDs κ.λπ. οι οποίες συνδέονται με το Διαδίκτυο και ανταλλάσσουν δεδομένα με στόχο την παρακολούθηση του περιβάλλοντος και την ανί- -χνευση συμβάντων, όπως φωτιά. Η ιστορία του IoT συνδέεται με την ιδέα που πρότεινε ο Mark Weiser [51] για ”ένα φυσικό κόσμο που είναι αόρατα συνυφασμένος με αισθητήρες, οθόνες και υπολογιστικά στοιχεία ενσωματωμένα σε καθημερινά αντικείμενα της ζωή μας και συν- -δεδεμένα συνέχεια στο δίκτυο”. Η υλοποίηση αυτής της ιδέας πραγματώνεται αρχικά στα Ασύρματα Δίκτυα Αισθητήρων (Wireless Sensor Networks - WSNs) τα οποία έχουν ως στόχο τη δημιουργία ασύρματων συστημάτων με φτηνά υπολογιστικά στοιχεία, τα οποία ονομάζονται κόμβοι αισθητήρων και λειτουργούν ασύρματα, μικρά σε μέγεθος, βάρος και κόστος για την υλοποίηση ενός κοινού στόχου. Καθώς η αντικατάσταση μπαταρίας δεν είναι εφικτή σε πολλές περιπτώσεις, θα πρέπει οι βασικές λειτουργίες, όπως η επεξερ- -γασία και η ανταλλαγή πληροφορίας, αυτών των συσκευών να βασίζονται σε τεχνικές βέλτιστης κατανάλωσης ενέργειας.

Με την εξέλιξη της τεχνολογίας και τη δημιουργία έξυπνων συσκευών με αναβαθμισμένα χαρακτηριστικά ως προς την υπολογιστική ισχύ και τις δυνατότητες παρακολούθησης τα Ασύρματα Δίκτυα Αισθητήρων επεκτάθηκαν σε αυτό που ονομάζεται Διαδίκτυο των Πραγμάτων όπου κάθε φυσική συσκευή που συνδέεται με το Διαδίκτυο είναι ικανή να μεταφέρει δεδομένα χωρίς να απαιτεί αλληλεπίδραση ανθρώπου με άνθρωπο ή άνθρωπο με υπολογιστή. Ένα παράδειγμα έξυπνων συσκευών του IoT εκτός από τα κινητά τηλέφωνα (smartphones) αποτελούν και οι ρομποτικές συσκευές όπως τα drones, εφ ’όσον φέρουν εξοπλισμό παρακολούθησης, όπως κάμερες και αισθητήρες, και υπολογιστικές μονάδες όπως Raspberry Pis. Χαρακτηρίζονται επιπλέον από την ικανότητά τους να μετακινούνται, προσθέτοντας ένα νέο βαθμό ελευθερίας βελτιώνοντας και μετασχηματίζοντας την παραδο- -σιακή αντίληψη των σταθερών συσκευών IoT σε κινητό IoT. Το IoT λοιπόν ενσωματώνει ένα σύνολο ετερογενών αντικειμένων που αλληλοεπιδρούν με το περιβάλλον τους και χρησιμοποιούνται από ένα εύρος διάχυτων εφαρμογών (pervasive applications). Οι περισ- -σότερες εφαρμογές εγείρουν προκλήσεις στη χρήση λόγω περιορισμένων πόρων σε αυτά τα μικροσκοπικά και μη επιτηρούμενα αντικείμενα.

Οι τεχνικές λοιπόν προκλήσεις του IoT μπορούν να καθοριστούν σε διάφορες περιοχές όπως:

- **Ετερογένεια:** Το πλήθος και η ετερογένεια των συσκευών δημιουργεί δυσκολίες στην σύνδεση τους καθώς πρέπει να συμπεριληφθούν διαφορετικές φυσικές διασυν- -δέσεις και αρχιτεκτονικές συστημάτων. Αυτές οι διαφορές μπορούν να προκαλέσουν προβλήματα σε ορισμένες επικοινωνίες όπως, στα ασύρματα δίκτυα που υλοποιούνται σε έξυπνες πόλεις παρουσιάζεται συνήθως μεγάλη καθυστέρηση μεταφοράς (latency).

- **Περιορισμένοι πόροι:** Ένα χαρακτηριστικό μιας διάχυτης συσκευής IoT είναι οι περιορισμένοι πόροι της. Μια τυπική συσκευή IoT που λειτουργεί με μπαταρία διαθέτει λειτουργίες αποθήκευσης, επεξεργασίας και συνδεσιμότητας. Δεδομένου ότι αυτοί οι πόροι είναι περιορισμένοι σε εκτεταμένες εφαρμογές, επομένως, εφαρμό-
-ζονται διάφοροι ενεργειακά αποδοτικοί αλγόριθμοι και πρωτόκολλα για την αποθήκευ-
-ση, επεξεργασία και μεταφορά δεδομένων σύμφωνα με τις απαιτήσεις εφαρμογής.

- **Διαλειτουργικότητα:** Οι IoT συσκευές κατασκευάζονται από πολλούς διαφορετικούς προμηθευτές χρησιμοποιώντας διάφορες τεχνολογίες. Η ομαλή ενσωμάτωσή τους μπορεί να επιτευχθεί μόνο εάν τα συστήματα IoT είναι χτισμένα πάνω από τα ανοικτά πρότυπα (open standards). Μπορεί να υπάρχουν πολλαπλά πρότυπα για τις ίδιες περιοχές όπως διαφορετικά πρότυπα ασύρματης δικτύωσης, αλλά πρέπει να δημιουρ-
-γηθεί διαλειτουργικότητα μεταξύ τους π.χ. πύλες μεταξύ διαφορετικών φυσικών δικτύων.

- **Ποιότητα Υπηρεσιων - QoS:** Με την πρόοδο των ενσωματωμένων συσκευών, η ισχύς επεξεργασίας των συσκευών IoT αυξάνεται καθημερινά, αλλά αυτό έχει ως αποτέλεσμα την αυξημένη κατανάλωση ενέργειας. Μια πιθανή λύση είναι οι συσκευές IoT να μπορούν να βασίζονται σε πιο ισχυρές συσκευές ή σε διακομιστές για την επεξεργασία δεδομένων, αλλά αυτό προσθέτει καθυστέρηση στην επεξεργασία δεδομένων, αυξάνει την καθυστέρηση του δικτύου και το κόστος.

- **Υπολογιστική πολυπλοκότητα και πολυπλοκότητα αποθήκευσης:** Οι συσκευές IoT παράγουν τεράστιες ποσότητες δεδομένων. Αυτά τα δεδομένα μπορούν να παράγονται σε συνεχή ροή ή κατά ριπές και να είναι σε δομημένη ή αδόμητη μορφή. Προκειμένου να αντληθούν τα μέγιστα από αυτά τα δεδομένα, πρέπει να μεταφερθούν, να αποθηκευτούν και να αναλυθούν. Οι ενέργειες αυτές έχουν μεγάλες απαιτήσεις στη δικτύωση, την αποθήκευση και την υπολογιστική υποδομή των IoT.

- **Ασφάλεια, εμπιστοσύνη και ιδιωτικότητα:** Η διείσδυση του Διαδικτύου στην καθη-
-μερινή ζωή υπογραμμίζει την ανάγκη για σωστές και ασφαλείς λύσεις. Ο μεγάλος αριθμός των συμμετεχόντων συσκευών καθιστά δύσκολη τη σχεδίαση ενός εντελώς ασφαλούς συστήματος, καθώς υπάρχουν πολλά σημεία πιθανής επίθεσης. Στη συνέχεια, οι λύσεις πρέπει να είναι φορητές σε ένα ευρύ σύνολο συσκευών, παρά τις εγγενείς διαφορές τους.

Η παρούσα διδακτορική διατριβή προτείνει το σχεδιασμό μιας αρχιτεκτονικής διαχείρισης πόρων που αντιμετωπίζει τις προκλήσεις της ετερογένειας, της ποιότητας της υπηρεσίας, των περιορισμών πόρων και της υπολογιστικής πολυπλοκότητας η οποία εφαρμόζεται σε διαφορετικά επίπεδα αρχιτεκτονικής, όπως στο επίπεδο συσκευής (Edge Layer), στο επίπεδο της επικοινωνίας (Communication Layer) και στο επίπεδο του middleware, όπως φαίνεται στην εικόνα ref fig: Architecture. Τα προτεινόμενα μοντέλα λήψης αποφάσεων μπορούν να παρακολουθούν τις ροές πληροφοριών που παράγονται από συσκευές IoT χωρίς πρότερη γνώση, μπορούν να προβλέψουν βάσει της Θεωρίας Βέλτιστης Πάυσης και Ανίχνευσης Αλλαγών αλλαγές οι οποίες μπορούν να διαταράξουν την απόδοση του

συστήματος IoT και να λάβουν μέτρα για να διατηρήσουν μια αποδεκτή ποιότητα υπηρεσίας στον τελικό χρήστη ενώ προσπαθούν να εξοικονομήσουν πόρους, στοιχείο απαραίτητο για την μετάβασή μας στην νέα εποχή του Διαδικτύου.

Η διατριβή αυτή χωρίζεται σε τρία βασικά μέρη. Το πρώτο μέρος αφιερώνεται στη μελέτη της εφαρμογής της θεωρίας Βέλτιστης Παύσης στη δημιουργία δυναμικού μεγέθους ομάδας πλαισίων (GOPs) εικονοροών προκειμένου να μεταδίδονται ακολουθίες video σε αποδεκτή ποιότητα σε ασύρματα δίκτυα αισθητήρων. Μια κατηγορία ασύρματων δικτύων αισθητήρων, στα οποία μεταδίδονται δεδομένα όπως η φωνή, η εικόνα και το βίντεο ονομάζονται Ασύρματα δίκτυα Πολυμεσικών Δεδομένων Αισθητήρων (Wireless Multimedia Sensor Networks - WSMNs). Τα WSMN έχουν ευρεία ανάπτυξη λόγω της ποικιλίας των εφαρμογών στις οποίες χρησιμοποιούνται, όπως η παρακολούθηση της κυκλοφοριακής συμφόρησης, του περιβάλλοντος, η παρακολούθηση ασθενών, οι υπηρεσίες εντοπισμού και η καταγραφή ασυνήθιστων συμβάντων, κα. Μία από τις προκλήσεις των WSMN είναι η διάρκεια ζωής του δικτύου, δεδομένου ότι οι κόμβοι λειτουργούν κυρίως με μπαταρία και η εικονοροή πολυμέσων ειδικά με υψηλή ποιότητα καταναλώνει αρκετή ενέργεια από τους κόμβους. Η εικονοροή πολυμέσων είναι η διαδικασία αποστολής και παράδοσης περιεχομένου video στους τελικούς χρήστες ή σε κάποιο σταθμό βάσης, όπου θα σταλεί για περαιτέρω επεξεργ--ασία. Η επιλογή της ποιότητας σύνδεσης εξαρτάται από το περιεχόμενο που διανέμεται. Για παράδειγμα, η ελάχιστη συνιστώμενη ευρυζωνική σύνδεση βίντεο είναι 2,5 MB/s, η οποία καταναλώνει μεγάλη ισχύ, ιδιαίτερα εάν το δίκτυο χρησιμοποιείται για παρακολούθηση και επίβλεψη σε πραγματικό χρόνο. Κατά συνέπεια, η αποτελεσματική συμπίεση του περιεχομένου πολυμέσων είναι ζωτικής σημασίας. Ωστόσο η κωδικοποίηση μιας εικονοροής πολυμέσων θα πρέπει να χρησιμοποιείται για τη συμπίεση των δεδομένων χωρίς να προκα--λείται βλάβη στην ποιότητα των δεδομένων. Ως εκ τούτου, στα πλαίσια της διδακτορική διατριβή μελετήθηκε και σχεδιάστηκε ένα μοντέλο λήψης αποφάσεων βελτιστοποιημένο με βάση το χρόνο, το οποίο δημιουργεί διαφορετικό μέγεθος ομάδας πλαισίων (GOPs) προκειμένου να μεταδίδει ακολουθίες εικονοροών πολυμέσων σε αποδεκτή ποιότητα με τη χρήση πόρων ασύρματων δικτύων. Η εφαρμογή και η αξιολόγηση της απόδοσης του μοντέλου με τα αποτελέσματα των πειραμάτων δημοσιεύτηκε στο περιοδικό Computer Communications της Elsevier [43].

Στο δεύτερο μέρος παρουσιάζονται τα αποτελέσματα της μελέτης εφαρμογής της θεωρίας Βέλτιστης Παύσης στο δυναμικο χειρισμό μηνυμάτων που ανταλλάσουν μη επανδρωμένα οχήματα (Internet of Drones) σε συνθήκες αβεβαιότητας της δικτυακής υποδομής. Το περιβάλλον λειτουργίας των συσκευών βασίζεται σε ασύρματες συνδέσεις όπου αντιμετω--πίζονται οι παρακάτω τεχνικές και ερευνητικές προκλήσεις:

- Πρόκληση 1: Παρακολούθηση των συσκευών σε πραγματικό χρόνο. Οι εφαρμογές παρακολούθησης σε πραγματικό χρόνο, όπως η ανίχνευση πυρκαγιών, απαιτούν την παράδοση μηνυμάτων ελέγχου μεταξύ των μη επανδρωμένων οχημάτων και του συστήματος ελέγχου με την ελάχιστη καθυστέρηση και με υψηλή ακρίβεια. Οι αποστολές αυτές συνήθως εξελίσσονται σε αγροτικές ή απομακρυσμένες περιοχές όπου η συνδεσιμότητα του δικτύου αναμένεται να είναι χαμηλή και μη αξιόπιστη. Επιπλέον η σύνδεση μεταξύ των κινητών κόμβων και του κόμβου ελέγχου ενδέχεται να είναι υπερφορτωμένη ή να διακοπεί ή να υποστεί μεγάλη απώλεια πακέτων.

Ως εκ τούτου μελετήθηκε και προτάθηκε ένας μηχανισμός λήψης απόφασης στο μη επανδρωμένο όχημα/σταθμό ελέγχου του, ο οποίος αποφασίζει αυτόματα τη διακοπή ή την μετάδοση των μηνυμάτων τηλεμετρίας/ελέγχου αντίστοιχα. Ο μηχανι--σμός αυτός εφαρμόζει τη θεωρίας βέλτιστης παύσης βάσει στατιστικών μετρήσεων πραγμα- -τικού χρόνου του υφιστάμενου ασύρματου δικτύου μέσα στο οποίο κινείται το μη επανδρωμένο όχημα.

- Πρόκληση 2: Ασφαλής έλεγχος των κινητών κόμβων. Η συνδεσιμότητα μεταξύ των μη επανδρωμένων συσκευών και των σταθμών ελέγχου πρέπει να λάβει υπόψη τον παράγοντα της κινητικότητας. Η κίνηση των οχημάτων προσθέτει έναν ακόμα βαθμό ελευθερίας στη λειτουργίας τους καθώς το μη επανδρωμένο όχημα κινείται στο χώρο με τις εντολές και τη καθοδήγηση του σταθμού ελέγχου. Τα μηνύματα ελέγχου μαζί με τις επιβεβαιώσεις (acknowledgments) είναι απαραίτητο να παραδίδονται πάντα ώστε να διασφαλιστεί η ασφάλεια και ολοκλήρωσης μια αποστολής. Η συνήθης προσέγγιση σε περίπτωση απώλειας δικτύου είναι η αυτόματη επιστροφή του κινητού κόμβου στην αρχική του θέση ακυρώνοντας την αποστολή. Αν το μη επανδρωμένο όχημα είναι κοντά στην ολοκλήρωση ή στον στόχο της αποστολής, η ακύρωση οδηγεί σε σημαντική σπατάλη χρόνου και πόρων.

Για να αντιμετωπισθουν οι προαναφερθείσες προκλήσεις, αναπτύχθηκε ένας δυναμικό και στοχαστικό σύστημα λήψης απόφασης για τη μετάδοση ή την παύση μηνυμάτων τηλεμετρίας και ελέγχου κατά τη διάρκεια μιας αποστολής βάσει των στοιχείων ποιότητας του δικτύου. Η διαχείριση των διαλόγων τηλεμετρίας και ελέγχου σχεδιάστηκε σύμφωνα με τις αρχές βέλτιστης παύσης (Optimal Stopping Theory - OST) ως πρόβλημα άπειρου ορίζοντα (infinite horizon). Ο προτεινόμενος μηχανισμός αυτός δημοσιεύτηκε στο συνέδριο IEEE WiMob 2018 [44] . Πρόσθετα, το σύστημα λήψης αποφάσεων ενισχύθηκε με ένα μηχανισμό αυτόματης ενεργοποίησης της μετάδοσης των μηνυμάτων παρακολουθώντας την ποιότητα του δικτύου. Καθώς η παύση της τηλεμετρίας ενός οχήματος που κινείται δεν μπορεί να εκτείνεται άπειρα, το πρόβλημα μελετήθηκε σύμφωνα με τις αρχές της βέλτιστης παύσης αλλά ως πρόβλημα περιορισμένου ορίζοντα. Ο ολοκληρωμένος μηχανισμός ελέγχου που βελτιστοποιείται με το χρόνο δημοσιεύτηκε στο περιοδικό ACM Transactions on Internet Technology [42] και επιτυγχάνει τη βέλτιστη παράδοση κρίσιμων πληροφορίων από τα μη επανδρωμένα οχήματα στα συστήματα ελέγχου και αντίστροφα.

Το τρίτο μέρος μελετά την εφαρμογή της θεωρίας Βέλτιστης Παύσης στη διαχείριση ουρών και την προτεραιοποίηση (prioritization) μηνυμάτων που διαχειρίζονται κατανεμημένη πλατ--φόρμα ροής μηνυμάτων (distributed message streaming platform). Η χρήση των πολυμε--σικών εφαρμογών αναπτύσσεται τα τελευταία χρόνια ραγδαία σε διάφορους τομείς όπως video-on-demand, εκπαίδευση εξ αποστάσεως, παρακολούθηση συμβάντων κ.α. Σύμφωνα με τη Cisco, η κυκλοφορία των βίντεο που παράγονται και καταναλώνονται μέσω του Διαδικτύου θα φθάσει το 80% της συνολικής κίνησης στο Διαδίκτυο έως το 2021. Ειδικά στην παραγωγή και τη διάδοση πολυμεσικών δεδομένων εταιρείες όπως η Netflix χρησιμο--ποιούν κατανεμημένες πλατφόρμες ροής δεδομένων (Distributed data streaming platforms) ώστε να ανταποκριθούν στην τεράστια παραγωγή/κατανάλωση πολυμεσικών ροών πληροφορίας. Μια από τις βασικές προκλήσεις των κατανεμημένων πλατφόρμων ροής

δεδομένων είναι η εξασφάλιση υψηλής ποιότητας παροχής περιεχομένου έγκαιρα και αξιόπι-
-στα ακόμα και αν οι χρήστες βρίσκονται σε περιβάλλον με περιορισμένη συνδεσιμότητα (π.χ. λόγω απώλειας καναλιών και υπερκορεσμένου εύρους ζώνης). Στα πλαίσια της διδακτορικής διατριβής προτείνουμε ένα στοχαστικό μοντέλο ελέγχου της ροής δεδομένων σε πραγματικό χρόνο, το οποίο εφαρμόζεται στους brokers της πλατφόρμας Apache Kafka. Στην πολιτική ιεράρχησης των μηνυμάτων που ανταλλάσσονται μέσω της πλατφόρμας προσθέτουμε ένα μηχανισμό βέλτιστης παράδοσης των μηνυμάτων στις ουρές βάσει στοι-
-χείων της ποιότητας της εφαρμογής (Quality of Service). Στόχος του μηχανισμού είναι να αποφευχθούν προβλήματα συγχρονισμού και μεγάλες καθυστερήσεις σε υψηλής προτερα-
-ότητας κατηγορίας μηνυμάτων είτε στις τελικές συσκευές είτε στο cloud. Η στρατηγική παράδοσης μηνυμάτων προέρχεται από δύο βέλτιστες πολιτικές βέλτιστης παύσης. Η εφαρμογή και η αξιολόγηση της απόδοσης του μοντέλου με τα αποτελέσματα των πειραμά-
-των έχει υποβληθεί για δημοσίευση στο συνέδριο IEEE WOWMOM 2020 [45].

Η διδακτορική διατριβή δομείται ως εξής:

- Κεφάλαιο 1: Εισάγονται οι σχετικές έννοιες, περιγράφονται οι προκλήσεις, διατυπώ-
-νονται τα βασικά ερωτήματα που αντιμετωπίζει η διατριβή και συνοψίζονται οι κύριες συνεισφορές.

- Κεφάλαιο 2: Παρουσιάζεται το σχετικό υπόβαθρο σχετικά με τη θεωρία βέλτιστης διακοπής, την ανίχνευση αλλαγών και τα βέλτιστα μοντέλα λήψης αποφάσεων για το διαδίκτυο.

- Κεφάλαιο 3: Αναλύονται τα αποτελέσματα ενός δυναμικού κωδικοποιητή που προσαρ-
-μόζεται στις αλλαγές στις ακολουθίες βίντεο βάσει της Θεωρίας Βέλτιστης Πάυσης.

- Κεφάλαιο 4: Παρουσιάζεται ο σχεδιασμός και η πειραματική αξιολόγηση της απόδο-
-σης και η σύγκριση της αξιολόγησης ενός μοντέλου λήψης αποφάσεων βελτιστοποι-
-ημένου χρόνου, προσαρμοσμένου στις αλλαγές ποιότητας δικτύου. Αυτό το μοντέλο εφαρμόζεται σε μη επανδρωμένα οχήματα και στις αντίστοιχες μονάδες ελέγχου τους.

- Κεφάλαιο 5: Παρουσιάζεται ο σχεδιασμός και τα αποτελέσματα ενός στοχαστικού πλαισίου βελτιστοποίησης ενος μηχανισμού λήψης απόφασης στην πλατφόρμα κατα-
-νεμημένης ανταλλαγής δεδομένων Apache Kafka.

- Κεφάλαιο 6: Συνοψίζονται τα κύρια σημεία της διατριβής και παρουσιάζονται κατευ-
-θύνσεις για μελλοντική έρευνα.

- Παράρτημα Α: Παρουσιάζονται Βασικές έννοιες της Θεωρίας Βέλτιστης Παύσης.

- Παράρτημα Β: Παρατίθενται οι αποδείξεις του Κεφαλαιου 4.

- Παράρτημα C: Περιέχει τους αλγορίθμοιυς υλοποίησης του Κεφαλαίου 5.

To my parents, Panagiotis and Sofia,
to my sister, Dimitra,
to my Nikos and
our daughter.

ACKNOWLEDGEMENTS

I want to express my sincere gratitude to my supervisor and advisor Stathes Hadjiefthymiades for his tremendous support on my PhD studies and research. He introduced me to the scientific field of pervasive computing in IoT and the marvelous process of academic communication. His guidance has proven crucial, especially in the difficult times of thesis topic reassessments and he taught me the valuable lesson of hard work. I also thank Professor Lazaros Merakos as my supervisor in my first steps in my BSc Thesis and his continuous support in my academic years. I thank Professor Dimitris Varoutas for his help during all these years and his patience to listen my concerns. I would also like to thank the members of the examination committee, Professors Evangelos Zervas, George Xylomenos, Christos Douligeris and Kimon Kontovasilis for accepting to review my thesis.

I am grateful to Professor Christos Anagnostopoulos as colleague and co-author for working with me on mathematical problems and teaching me many important skills that will be valuable in my career. I would like to thank Christos for our lengthy conversations, brainstorming and mathematical sessions. Christos, with his passion in applied machine learning and Optimal Stopping Theory, has greatly shifted my interest to this research area during the time we were working together several years before getting into the PhD program. This dissertation was shaped by our collaboration! I thank Kostas Kolomvatsos for his patience and guidance in my first steps of my PhD Thesis. Kostas was my supervisor of my MSc Thesis and without his contributions and ideas in decision making systems this journey might never had begun for me.

During my journey as a PhD student, my research was influenced significantly by communication I had with: Kostis Gerakos for our interesting discussions about Apache Kafka who, very patiently, provided very valuable feedback on my early steps in performance evaluation. Additionally, I would like to thank Thanos Chalvatzaras for the excellent and fruitful scientific collaboration we had. I would like to thank my colleagues for always being very enthusiastic about discussing technical and social aspects of PhD life! Vassilis Papataxiarhis, Babis Andreou, Dimitris Zampouras and Fanis Kontos. They all made our lab a more interesting and joyful place to work.

In addition, I would like to thank my mother Sofia for inspiring me the virtue of ambition and my father Panagiotis (as a mathematician) for the introduction to mathematics from my early stage of my childhood. Moreover, I thank my sister Dimitra for always being there for me and listening my conference presentations and for her support and love . I thank my husband, Nikos, for his unconditional support and interest in my progress during this period. His clear way of thinking was always an inspiration for me.

Finally, I thank my childhood friend Anna for her life-long company and for her moral support over the course of my PhD studies. Last but not least, I am deeply grateful to my friends and my colleagues for their unparalleled support and encouragement to complete this dissertation.

CONTENTS

1 INTRODUCTION **31**

1.1 Scope and Problem Statement 31

1.2 Contribution and Methodology 32

1.3 Structure and overview of the thesis 35

2 BACKGROUND AND RELATED WORK **37**

2.1 Related Work on scene detection and adaptive encoding policies 37

2.2 Related Work on contextual information flow delivery policies and routing protocols 38

2.3 Related Work on distributed message platforms 40

2.3.1 Relevant Technologies used as message bus 40

3 Optimal Stopping Theory applied on Time-optimized Grouping-of-Pictures **43**

3.1 Motivation 43

3.1.1 Contribution 44

3.2 System Design of Time-optimized Grouping-of-Pictures 45

3.2.1 Group-Of-Pictures structure 45

3.2.2 Rationale 45

3.2.2.1 Gamma Distribution 47

3.2.3 Solution Fundamentals 48

3.2.3.1 Normal Distribution 52

3.2.4 Complexity & Model Design Parameters 53

3.3 Performance Evaluation 54

3.3.1 Simulation setup 54

3.3.2 Discussion of Simulation results 55

3.4 Conclusions 58

4 Real-time stochastic control mechanism adaptive to changes in network quality based on optimal sequential decision making rules. **59**

4.1 Motivation . . . 59
4.1.1 Contribution . . . 61
4.2 System Design of Time-Optimized Decision Making Model for Unmanned vehicles . . . 62
4.2.1 Rationale . . . 62
4.2.2 Time-Optimized Change-Point Decision Making Process . . . 63
4.2.2.1 Solution for TOCP . . . 65
4.2.3 Discounted Reward Decision Making Process . . . 67
4.3 Performance Evaluation . . . 71
4.3.1 Experimental Platform & Methodology . . . 73
4.3.2 Model Parameters & Real Datasets . . . 73
4.3.3 Experiments: Performance & Comparative Assessment . . . 77
4.3.3.1 Expected Performance in Mission M1 . . . 80
4.3.3.2 Expected Performance in Mission M2 . . . 80
4.3.3.3 Expected Latency in Missions M1 & M2 . . . 81
4.4 Conclusions . . . 82

5 Time-optimized prioritization of Kafka Message Scheduling for Unmanned Vehicles in IoT networks 83
5.1 Motivation . . . 83
5.1.1 Contribution . . . 84
5.2 System Design . . . 84
5.2.1 Distributed Communication Layer on UAVs . . . 85
5.2.1.1 Apache Kafka Clients . . . 85
5.2.1.2 Topics and Partitions . . . 85
5.2.2 Message Priorities in Apache Kafka . . . 88
5.2.3 Optimal Stopping Delivery Strategy . . . 89
5.3 Performance Evaluation . . . 92
5.4 Conclusions . . . 96

6 CONCLUSIONS AND FUTURE WORK 97

ABBREVIATIONS - ACRONYMS 99

APPENDICES **100**

A Preliminaries in Optimal Stopping Theory & Change Detection Theory **101**

A.1 Optimal Stopping Theory 101

A.2 Change Point Detection Theory 102

B REAL-TIME STOCHASTIC CONTROL MECHANISM ADAPTIVE TO CHANGES IN NETWORK QUALITY BASED ON OPTIMAL SEQUENTIAL DECISION MAKING RULES **105**

B.1 Proof 105

B.2 CUSUM algorithm 105

C TIME-OPTIMIZED PRIORITIZATION OF KAFKA MESSAGE SCHEDULING FOR UNMANNED VEHICLES IN IoT NETWORKS **107**

REFERENCES **112**

LIST OF FIGURES

1.1 Multi-layer Decision making Architecture 33

1.2 Contribution of the decision making models applied on Edge, Communication and Middleware layer 34

3.1 GOP in the H.263 video flow 45

3.2 MPEG Encoder 46

3.3 Fitting probability distribution functions based on the actual SATD values. . 47

3.4 Graphs of functions $\mathbb{I}_1$ (continuous line) and W (dotted line), T=30 50

3.5 α, β values of Γ distribution for the set of videos used in our experiments . . 54

3.6 SATD between classic approach and OST- football video 56

3.7 SATD between classic approach and OST - waterfall video 57

3.8 SATD between classic approach and OST - bridge video 57

3.9 SATD between classic approach and OST - hall video 58

4.1 UxV State Transition Model 63

4.2 (Upper) a. The probability density function f_0 and (lower) b. the f_1 model fitting for *good* and *bad* quality of QNI values, respectively. 64

4.3 The behavior of the QNI and the cumulative log-likelihood ratio corresponding to a change from a 'good' network state to a 'bad' network state. 67

4.4 Analysis of LDF based on different values of discount factor γ. 70

4.5 The TOCP-DRP proposed architecture for the UxV Management. 73

4.6 The Turtlebot UGV with XBOX Kinect and Raspberry Pi computing modules. 74

4.7 The detection delay function D_n vs. (a) different α values; (b) different γ values; and (c) different observations r). 75

4.8 The behavior of (up) produced and (down) consumed packets in a saturated network 77

4.9 The differences between produced and consumed data with No policy applied (black area) and with OST model, $\alpha \in [0, 0.002]$ (grey area). 78

4.10 Real time monitoring of the robot while executing "Mission 1-Path exploration" and "Mission 2-Scanning of an area" 79

4.11 The QNI for all the compared policies regarding the mission M1: Exploration of a Path. 80
4.12 The QNI for all the compared policies regarding the mission M2: Scanning of an unknown Area. 81
4.13 The latency (ms) measured during the no-policy and the TOCP-DRP policy in mission M1 (a) and mission M2 (b). 82

5.1 Main components of Kafka 86
5.2 Decision Making Processes applied on Kafka Priorities Queues. 87
5.3 Joint probability Density function of RTT and packet loss in normal and saturated state. 90
5.4 Experimental setup for simulated scenario. 93
5.5 KPI performance metrics measured during the simulation. 94
5.6 Decicion making between states in S_n to S_s and then to S_n. 95
5.7 Delay in two priority queues between states in S_n to S_s and then to S_n. . . . 96

B.1 Monotony analysis of L 105

LIST OF TABLES

2.1 Comparison of ActiveMQ, ActiveMQ and Apache Kafka 40

3.1 α and β values for different types of video 53
3.2 Size of bitstreams transmitted in network 56

4.1 Rules of State Transitions 63
4.2 NLogL and BIC metrics for the probability distributions. 65
4.3 Model Parameters for the Experiments. 72
4.4 Packets Produced and Consumed 76

1. INTRODUCTION

1.1 Scope and Problem Statement

Internet of Things (IoT) is one of the most promising paradigms nowadays characterized by the use of smart and self-configured objects, like sensors, actuators, wearables etc, that are connected to a network and exchange data by sensing, reacting to events, and interacting with the environment. The history of the IoT can be traced in the area of Ubiquitous computing and Wireless Sensor Networks (WSNs). Mark Weiser [51] proposed the idea of a smart environment: ”a physical world that is richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives, and connected through a continuous network”. This idea is explored in the area of WSNs, where the goal is to build a system of many cheap computational components, called sensor nodes, wirelessly connected and jointly working towards a common goal. Sensor nodes have specific and, usually, small size, weight and cost.

Going a step further and with the technological evolution, new physical devices with enhanced characteristics at both hardware and software parts are introduced daily, e.g. smartphones, wearables, unmanned devices etc, extending WSN paradigm to a ”network”, i.e. IoT, where every physical device is connected to the Internet ready to transfer data without requiring human-to-human or human-to-computer interaction. Especially robotic devices take part in IoT as long as they carry sensing equipment and on-board computing elements. IoT embodies a vision of merging heterogeneous objects while utilizing the Internet as a backbone of communication to establish interaction among physical and virtual entities. These seamless interactions among heterogeneous objects enable ubiquitous and pervasive applications. Most of these applications pose many challenges due to constrained resources in these miniature and unattended objects.

The technical challenges of the IoT can be identified in several areas:

- **Heterogeneity:** Connecting trillions of devices in the same network is not an easy task. The heterogeneity of the involved devices makes it even more difficult, since many different physical interconnections and system architectures can be expected. These differences can cause problems to certain communications. For instance, city wide ad-hoc wireless networks typically have large latency, which break timing perspectives of current Internet protocols.
- **Constrained resources:** An important aspect of pervasive IoT device is its constrained resources. A typical battery-operated IoT device possesses storage, processing, bandwidth, and energy as its resources. Since these resources are limited and the battery replacement is not frasible in many cases, therefore, various energy-efficient lightweight algorithms and protocols shall be being implemented to store, process and transfer the data as per application requirements.
- **Interoperability and integration:** The IoT is built by many distinct vendors, using

various technologies. Their seamless integration can only be possible if IoT systems are built on top of open standards. There may be multiple standards for the same areas (e.g. different wireless networking standards), but interoperability between them has to be established (e.g. gateways between different physical networks).

- **Quality of Service**: With the advancements in embedded devices, the processing power of IoT devices is increasing day by day, but this results in increased energy consumption. To overcome that, IoT devices can rely on more powerful devices or servers for processing of data, but this introduces a delay in data processing and increases network delay and cost.
- **Computational and storage complexity:** The devices that comprise the IoT generate massive amounts of data. These data can be continuous or in bursts, and be in structured or unstructured form. In order to extract the most from these data, they have to be transported, stored and analyzed. These operations put enormous pressure on networking, storage and computational infrastructure.
- **Security, Trust and Privacy:** The penetration of the IoT in daily lives emphasizes the need of proper secure solutions. The large number of devices involved makes the design of a completely secure system difficult, as there are many points of potential attack. Then, any solutions have to be portable to a wide set of devices, despite their intrinsic differences.

This thesis proposes the design of a resource management architecture that addresses the challenges of Heterogeneity, Quality of service, Resource constraints and Computational complexity applied on the edge, communication and middleware layer as shown in Figure 1.1. The proposed decision making models can monitor information streams produced by IoT devices, can predict changes based on Optimal Stopping and Change Detection theory that can disrupt the performance of the IoT system and can take actions to retain acceptable Quality Of Service while trying to save resources which is essential for progressing into a new era of IoT.

1.2 Contribution and Methodology

A Multi-layer IoT Resource Management Architecture is studied during this thesis taking into consideration different QoS parameters such as delay, bandwidth utilization, power consumption, multi-media in-network processing etc., for applying decision making models based on Optimal Stopping Theory. As shown in Figure 1.2 the main layers of architecture are: the Edge, Communication layer and the Processing layer. The Edge Layer is referred to the device and the computational complexity of different tasks. One daily energy demanding task in IoT is multimedia streaming, which causes the energy drainage to network resources and lifetime. Therefore efficient compressing methods are needed in order to minimize the consuming power but without harming the content of the distributed data. The Communication layer is based on wireless network technologies in order to

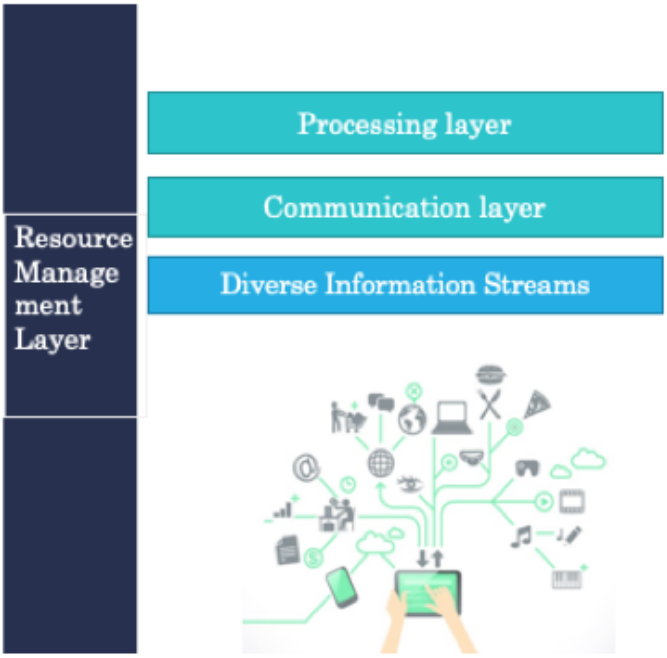


Figure 1.1: Multi-layer Decision making Architecture

enable interactions between various heterogeneous devices and information streams. At this layer information streams produced by heterogeneous sources are gathered in real time while taking into account the rational use of IoT devices. This mere data need to be combined in order to extract knowledge. At Processing layer distributed data streaming solutions are targeted because they are extensively used to manage the big data flows of generated information streams by IoT devices. It is necessary these platforms to support reliable and timely communication despite poor performance of underlying units like lossy channels and failed components. At this these we design and implement online decision making models based on Optimal Stopping Theory in order to monitor the performance of units in different layers and predict disruptive changes. For example in edge layer during the multimedia compressing task a change can be defined as a scene change during the transmission of a multimedia sequence or an unknown object shown suddenly in the frame; a change in communication layer can be defined as a network blind spot of communication link during a flight of a drone in an unknown area. Changes trigger actions like the reconfiguration of the input system at Resource management layer. The main challenge is with no prior knowledge to monitor, predict changes and proactively act to sustain the continuous performance of a task efficiently without the energy drain of IoT devices.

This thesis consists of three main parts. At the first part we include our study of a content driven model applied to infrastructures with restricted resources like Wireless Sensor Multimedia Networks (WSMNs) in order to support multimedia application in such infras-

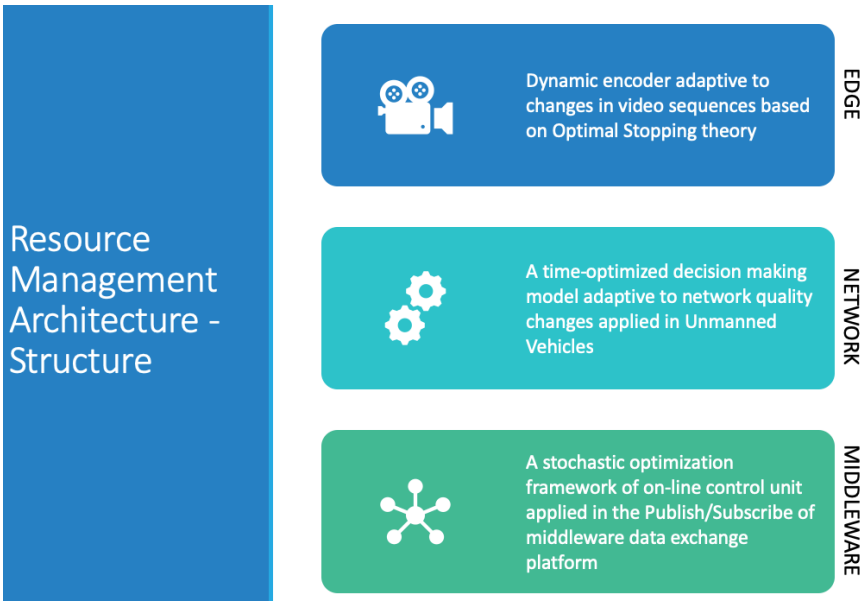


Figure 1.2: Contribution of the decision making models applied on Edge, Communication and Middleware layer

tructures. Currently WSMNs are attracting significant attention due to the variety of applications in which can be applied such as traffic congestion, environmental habitat patient monitoring, etc. Although providing better quality for images and videos is necessary, it shortens the network lifetime as the energy battery operated sources are rapidly drained. Going inside the device, we propose a dynamic video encoding model that detects scene changes and tunes the synthesis of MPEG Group of Pictures (GOP) to meet Quality of Service objectives, i.e. transmit video sequences in acceptable quality, with a rational use of the IoT resources. The decision making process is based on Optimal Stopping Theory decision rule for the conclusion of the GOP in the encoder and the transmission of intra-coded frames. The quantitative findings show that the proposed scheme performs quite efficiently while dispatching video with different characteristics.

The second part of the thesis presents a study of performance changes in communication links between IoT devices. The idea behind this research is the efficient monitor and control of unmanned devices operating in critical missions like natural disasters. How can we secure the real time monitoring of the devices in order to acquire useful information e.g. the detection of a fire while in the same time we secure the unsupervised control of these devices? In this chapter, we propose a real-time control mechanism to *adapt* to changes in network quality by dynamically *pausing* control telemetry and control messages based on *optimal sequential decision making rules*. This is expected to ensure the trouble-free delivery of critical information subject to the dynamic network status that unmanned devices encounter while dispatching a certain mission. Our rationale is that should the network be performing properly, then the transmission control can be 'relaxed' to exploit the available resources in the resource-constrained IoT device. Our model introduces two sequential optimal stopping time decision making mechanisms based on the Change De-

tection theory and an application-specific discounted reward process. The experimental performance evaluation and comparison assessment showed the successful delivery of messages in poor network conditions and the moderate production of messages so as not to burden an already saturated network.

At the final part of this thesis we study the performance of a distributed message platform implemented as a middleware in an IoT system. The huge amount of data generated by sensor-instrumented objects of the real world in an IoT environment impose a great demand on processing and storage to transform the data into useful information or services. Some applications can be latency sensitive, while other applications can require complex processing including historical data and time series analysis. Therefore, considering the typical resource constraints of IoT devices, it is difficult to envision a real world IoT ecosystem without including a cloud platform or at least a distributed data streaming platforms. Distributed data streaming solutions manage big data flows of relevant data to/from devices, services and micro-services and are critical centerpiece of IoT deployments. These platforms are necessary in IoT infrastructures to process such enormous volumes of data against resource constrained IoT devices. The key challenges arise when supporting reliable and timely communication over constrained networks (e.g. due to lossy channels and failed components). To overcome these challenges, we propose a stochastic optimization framework of on-line control unit applied in the Publish/Subscribe of middleware data exchange platform, in our case Apache Kafka, adaptive to changes in performance of the studied platform. We enhance our messaging distribution platform by applying prioritization policy of different types of messages when key performance indicators change. The optimality of the proposed mechanism is achieved by applying optimal delivery decision making policy in different priority queues. Optimal delivery decisions involves whether a consumer/producer in the device edge shall pause the pull/push requests in order not to overload a saturated message bus, to cause synchronization issues or to risk to completely lose the messages.

1.3 Structure and overview of the thesis

In the following, we present the structure of the thesis and a a brief overview of the Chapters.

- Chapter 2: The relative background of scene detection discussed in Chapter 3, the relative background on contextual information flow delivery policies and distributed message platforms are extensively discussed.
- Chapter 3: The study on a dynamic encoder adaptive to changes in video sequences based on Optimal Stopping theory is analysed.
- Chapter 4: The design, the experimental performance evaluation and comparison assessment of a time-optimized decision making model adaptive to network quality changes is presented. This model is applied on unmanned vehicles and their respective control units

- Chapter 5: A discussion on the design of a stochastic optimization framework of on-line control unit applied in the Publish/Subscribe of middleware data exchange platform is presented.
- Chapter 6: The concluding remarks of this thesis are presented along with directions for future work.
- Appendix A: The preliminaries of Optimal Stopping Theory are included
- Appendix B: Proofs of Chapter 4 are presented.
- Appendix C: The algorithm shown in Chapter 5 for the Discounted Reward Function is presented.

2. BACKGROUND AND RELATED WORK

A Multi-layer IoT Resource Management Architecture is studied during this thesis taking into consideration different QoS parameters such as delay, bandwidth utilization, power consumption, multi-media in-network processing etc., for applying decision making models based on Optimal Stopping Theory. We present the related works studied at this thesis in three main subsections. The first subsection includes the literature on scene detection and adaptive encoding applied on multimedia streaming. An extensive discussion of contextual information flow delivery policies and routing protocols applied on WSNs and robotic devices is presented in the second subsection. Finally we present the state of the art distributed platforms making a summary of the technologies applied as message bus for the managing of big heterogeneous data flows.

2.1 Related Work on scene detection and adaptive encoding policies

Scene change detection is the main criterion which defines GOP length in many research approaches. Therefore we present below related works on scene detection and adaptive GOP structuring. All the following approaches are based on the following steps: authors extract some statistics from consecutive frames like color histograms or block differences and then compare this information with a specific threshold. Especially in compressed videos we can use several well studied indicators like discrete cosine transform (DCT) coefficients [53],[36], and block modes/types [52] and motion vectors [33],[11] and [**?**].

Authors in [29] study the scene detection problem. The use of texture variation indicators, like interframe variations combined with a parallel processing method is proposed for video encoding with adaptive GOP structure. They detect both types of scene changes, i.e. gradual and abrupt scene changes, at less computation effort and the creation of new GOP is based on this detection. They also propose also balanced frame-level parallel scheduling algorithms that first determine frame priority, followed by the thread priority assignment. However this approach is mostly focused in the parallelization of video processing and not on the implementation of more sophisticated algorithms for scene detection. Scene detection can be based on other approaches like the pixel-based method in [26].The differences between the pixel values of two sequential frames is measured and if this value is higher than a specific threshold a change is detected. The disadvantage for the pixel-based method is that it is sensitive to object motion in the scene. Histogram comparison is proposed in [35] where the difference between histograms of two sequential frames is computed in order to determine the scene change. It should be mentioned that histograms are not sufficient information for scene change detection as long as different scenes can have similar histogram values. Scene change detection by using Markov Chain Monte Carlo (MCMC) algorithm [54] and k-means clustering-based [14] approaches also provide feasible solutions. The posterior probability calculation of the MCMC algorithm is computed based on the data likelihood of the video and it requires important computation effort. However statistical techniques, e.g. pixel-based and block-

based luminance difference approaches, involve lower complexity than clustering-based approaches as shown in [47].

Adaptive GOP size is also a well-known problem in the related literature but most approaches follow intuitive processes. Dumitras and Haskell [6] developed a frame type decision algorithm, which employs the motion similarity information. Authors show that the optimal number of B frames between reference frames must be between 0 and 2. In [34] the author proposes to place I frames to the positions of detected cuts during the process of video encoding. Our model follows a different approach compared to these research efforts. Mainly these methods compare two or more consecutive frames and not by taking the advance of interframe result. This approach requires a significant computational effort. In addition histograms and other traditional methods based on statistics cannot be applied on real-time fast flow video in which changes can occur stochastically. Our model lets the encoder decide the GOP size by autonomously determining the appropriate time to conclude the GOP.

Methods derived from the optimal stopping theory have been applied to information dissemination in ad-hoc networks. The data delivery mechanisms in [8] and [9]deal with the delivery of quality information to context-aware applications in static and mobile ad-hoc networks respectively assuming epidemic-based information dissemination schemes. The mechanism in [8] is based on the probabilistic nature of the ”secretary problem” [23] and the optimal online problem. In [10] authors make optimal stopping decisions on the collection of contextual data from WSNs. Authors try to determine the best time to switch from decision to learning phase of Principal Component-based Context Compression (PC3) model while data inaccuracy is taken into account. If data inaccuracy remains at low levels, then any deterministic switching from compression to learning phase leads to unnecessary energy consumption. OST rules are applied between compression and learning phases of the observations.

2.2 Related Work on contextual information flow delivery policies and routing protocols

The challenge of optimizing contextual information flow delivery among UxVs is non trivial given the network circumstances and status. To our knowledge, there is no prior holistic work addressing the problem of time-optimized information flow. In the literature, research has been extensively focused on message-routing protocol employed on UxVs. Opportunistic networks have been proposed as long as they are capable of maintaining efficient operation in a wide range of network density and mobility conditions [46],[37]. By classifying the diversity of topological conditions in networking environments, one end of the spectrum corresponds to almost static dense topologies. In this case, conventional topology-based protocols [38] function best by using node labels / identities. As the nodal density decreases and / or the mobility increases, and up to a point where the connectivity status between pairs of nodes remains stable, position-based families of protocols [37], [20] become more suitable.

Additionally, in networks of low nodal density, intense mobility becomes a prerequisite for the creation of contact opportunities. For such topologies, protocols based on the 'carry' action [46], [25], i.e., the spatial transposition of the message due to the physical movement of the carrier node, perform efficiently. The aforementioned routing protocols have been designed to accommodate a restricted set of possible network conditions, corresponding to a particular sub-range and yield satisfactory performance only under these conditions.

Opportunistic Networking is also an open and an active field of research where OST can be applied at routing delivery protocols. A proposal for opportunistic networks (OppNet) [16] is studied in which the authors present Softwarecast as a general delivery scheme for group communications based on mobile code. This software code and a delivery state is the main input to persist refined delivery-decision making methods based on OST to implement complex decisions. In [19], the authors present the Relcast, a composite routing-delivery scheme that used OST-based delivery strategies to route messages to profiles which are defined by delivery functions such as best maximum and over-the-average. If we go a step further, we define a routing delivery protocol to social OppNet like influencers' networks. The [18] refer to an OST-based solution to deliver messages in highly connected networks. However, the proposed solutions are based on metrics like low latency, while the authors in [17] proposed a solution of broadcast protocols for OppNet based on efficiency, preventing unrestrained propagation of messages.

All the proposed delivery routing protocols are based on variations of the Secretary Problem [23] like the called rank-based selection and cardinal payoffs variation of the secretary problem [15]. However a unique strategy cannot be applied to sequences with abrupt changes where each state shall be treated differently. Other research efforts are focused on delay-tolerant methodologies, where mobile sinks (e.g., data aggregation nodes) 'patrol' a number of static sensor nodes and collect data [32], [50]. Nonetheless, due to their delay-tolerant principle for data delivery, they cannot be directly applied to real-time applications like disaster management.

Methods based on the principles of dynamic stochastic optimization frameworks, like Optimal Stopping Theory, have been successfully applied to information dissemination in ad-hoc networks. The authors in [21] add mobility into wireless network infrastructure, i.e., WiFi access points (AP) on wheels, which move to optimize user performance. The Roomba devices equipped with network interfaces move independently around areas in order to maximize the wireless capacity in this area. However, the mobile devices are moving based on a grid at the floor to predefined paths. In [**?**], researchers apply Optimal Stopping Theory based on Change Detection only in-network statistics. This method is only applied to pause the generation of telemetry messages. Pausing period stops when a time threshold is reached and for this period framework is agnostic to network state. Researchers' method is compared with our proposed model in Section 4.3.3 applied on real UxVs. Contextual data delivery mechanisms have been studied in the literature though from a different 'perspective' in mobile ad-hoc networks. The contextual data delivery mechanisms in [10], [8], [43] and [9] deal with the delivery of quality information to context-aware applications in static and mobile ad-hoc networks, respectively, assuming

epidemic-based information dissemination schemes. In [10], the authors propose optimal decision making approaches on the collection of contextual data from WSNs. The mechanism in [8] is based on the probabilistic extension of the well-known Secretary Problem introduced in [23] merged with an optimal on-line stochastic optimization problem. The authors in [7] tackle the task offloading decision making problem by adopting the principles of optimal stopping theory (OST) to minimize the execution delay in a sequential decision manner. Their approach significantly minimizes the execution delay for task execution and the results are closer to the optimal solution than other deterministic offloading methods. The authors in [43] study a dynamic video encoder that detects scene changes and tunes the synthesis of Groups-of-Pictures (GoP) accordingly based on an 'Black-Jack' like application of Optimal Stopping Theory. The proposed MPEG encoder tracks the error between the sequential frames in a Group-of-Pictures (GOPs) and optimally creates GOP sizes which are content-based with the minimum waste of the resources.

2.3 Related Work on distributed message platforms

In this section we study the state of the art of the distributed data streaming platforms. We make a summary of the different technologies applied as message bus. Additionally, a brief survey on articles that use Optimal Stopping Theory in routing and delivery of data is presented. Finally we present our contribution in the delivery of messages in a cloud-based infrastructure.

2.3.1 Relevant Technologies used as message bus

The available software solutions that suit the requirements in terms of asynchronous communication between the components, using a Publish/Subscribe or Publisher/Consumer communication model are described. This communication model is typically realized by means of a Message Broker, which connects different applications that can simultaneously act as Publishers and/or Subscribers (resp. Publishers and/or Consumers).

Funct.Techn.	Active MQ	RabbitMQ	Apache Kafka
High Throughput	-	-	+
Scalability	+	+	+
Push-based Model	+	+	-
Pull-based Model	-	-	+

Table 2.1: Comparison of ActiveMQ, ActiveMQ and Apache Kafka

The Apache ActiveMQ [2] Message Broker provides an open source implementation of the Java Message Service (JMS) specifications. It acts as a reliable hub in any message oriented enterprise application, and integrates perfectly with Java EE containers, ESBs, and other JMS providers. It is designed for high performance clustering, client-server and

peer-to-peer based communication. It uses a specific protocol, called Open Wire, to allow access to Active MQ brokers using different programming languages and protocols. For enabling cross-language/ platform communication of different clients with ActiveMQ, the STOMP protocol [5] is also supported. STOMP is a simple text orientated messaging protocol, which provides an interoperable wire format which enables messaging interoperability among many languages, platforms and brokers. In addition, ActiveMQ provides support for different messaging protocols, transport options and interfaces, such as:

- AMQP protocol [1] - a platform-agnostic protocol, suitable for real-time data streams communication, and business transactions between applications, across distributed cloud computing environments. AMQP is an OASIS standard, thus avoids the need to use proprietary technologies and would be an interesting solution for interoperability and ease of integration and extension of our platform
- REST software API - ActiveMQ implements a RESTful API to messaging, allowing any web capable device, or web application, to publish or consume messages using a regular HTTP POST or GET
- TCP transport - through the TCP transport Apache ActiveMQ also provides clients with the possibility to connect to a remote ActiveMQ server by using a simple TCP socket interface
- MQTT protocol [3] – protocol specifically designed to allow connections and communication in an IoT environment.

RabbitMQ [4] is another message broker implementation which supports several messaging protocols, directly and through the use of plugins. Several RabbitMQ servers on a local network can be clustered together, forming a single logical broker. Like Apache ActiveMQ, supported protocols include STOMP, AMQP and MQTT. Further, it provides:

- HTTP API to send and receive messages from a web browser (management plugin)
- STOMP messaging to the browser (Web-Stomp plugin)
- JSON-RPC lightweight remote procedure call protocol (synchronous protocol) to the browser (channel plugin)

The decision for utilizing Apache Kafka acting as message broker applied on UAVS was taken by comparing the features and capabilities of other popular technologies such as RabbitMQ and ActiveMQ. While the advanced routing capabilities and the maturity of both projects was a beneficial factor features such as exceptional scalability and high throughput made Apache Kafka more fitting from the competition. Message persistence was also an important factor for choosing Apache Kafka since RabbitMQ and ActiveMQ have inadequate policies regarding the retention period. RabbitMQ brokers keeps track of the consumed messages and if the number of consumers that received the message is considered sufficient then it deleted the message. ActiveMQ can support persistent messaging which however is slower and for message replication purposes uses a Master/Slave

model that can only guarantee data recovery and not on the fly re-transmission. An additional layer of message storage could be applied in both solutions but this would have an unnecessary increase in complexity. Another disadvantage of RabbitMQ in terms with Apache Kafka is the increased number of bindings while Apache Kafka can support multiple consumption for a message topic and difficulties to increase the cluster size by adding new nodes while in Apache Kafka the re-partitioning and replication of messages are done automatically in background [22]. Finally Kafka Connect, a framework with already developed and supported connectors to other systems such as databases, stream processing platforms etc was an unique feature that was considered a supporting factor for the development of the different modules. In this thesis, we propose a decision making modules adaptive to performance changes applied on distributed streaming platform that allows in an optimal way to deliver high and low priority messages in the network. This problem can seen as an optimization problem of deciding how to deliver messages in the network with different priorities. Optimal Stopping Theory deals exactly with this type of problems, where it is requested to choose a time to take a particular action, in order to maximise an expected reward or minimise an expected cost. The two well known Optimal stopping problems are the secretary problem/wedding problem and the blackjack problem. In secretary problem a person must interview a number of candidates n that can be ranked from the best to worst in order to select the best candidate. If a candidate is not selected then there is no option for recall. The optimal solution studied in [23] is to discard $\frac{n}{e}$ candidates and choose after this interval the best candidate shown of the previous ones. Blackjack problem comes from the well known card game where a user observes sequentially the values of an infinite sequence of non negative random variables. After each observation, the user decides whether to stop or to continue. If the user decides to stop at a given moment, the obtains a payoff dependent on the sum of already observed values. The greater the sum, the more the user gains, unless the sum exceeds a given positive number. If so, the decision maker loses all or part of the payoff. Studies on data delivery mechanisms were extensively presented in the previous sections.

3. OPTIMAL STOPPING THEORY APPLIED ON TIME-OPTIMIZED GROUPING-OF-PICTURES

3.1 Motivation

Staring from the edge, we study a dynamic video encoder that detects scene changes and tunes the synthesis of Groups-of-Pictures accordingly. Such dynamic encoding can be applied to infrastructures with restricted resources, like Wireless Sensor Multimedia Networks (WSMNs) [13] facilities where multimedia streams are of use. Currently, WSMNs are attracting significant attention due to the variety of applications in which they can be applied such as traffic congestion, environmental, habitat and patient monitoring and recording unusual events. One of the challenges of WSMNs is the lifetime of the network, since the nodes are mostly battery-operated. Although providing better quality for images and videos is necessary, it shortens the network lifetime as the energy sources are rapidly drained. One of the features, which is energy consuming in WSMNs, is multimedia streaming. Multimedia streaming is the process of sending and delivering multimedia content to end users or to the fixed infrastructure, where it will pass through further processing. Multimedia streaming requires efficient compressing methods which minimize the consuming power without harming the content of the distributed data.

The most popular standard for motion compensated video compression is MPEG. Even though it was originally designed for digital storage media, its capabilities have been increased to support a high spectrum of bit rates in order to be used in streaming multimedia applications over the Internet or over lossy wireless networks. Although in this thesis we assess the performance of our scheme using the MPEG-2 standard our technique is also applicable to the MPEG-1 and 4 standards as well as the H.26* family of standards. This wide applicability is based on the intra-frame calculations that we undertake in order to throttle our decision making process. In this paragraph we briefly present the broader MPEG video compression technique. A key feature of MPEG is the ability to compress a video signal to a fraction of the original size by coding only the differences between two sequential frames instead of an entire frame. This compression method is called differential encoding. MPEG uses three types of frames, i.e. I, P and B frames to implement different compression methods and exploit inter-frame dependencies within the video stream. Typically, the repeated sequence of I, P, and B frames is known as Group of Pictures (GOP). Each GOP is characterized by a specific number of I, P and B frames. I frame means an intra-coded frame and can be treated as a standalone image. I frames are often used as a reference point to a new scene or a big change to the already transmitted sequence of frames. A P frame contains only predictive information. P frame is generated by looking at the deltas between the present and the previous frame. B frames are created by examining the differences between the previous and the next reference frame, i.e. either I or P, in a sequence of frames. P and B frames do not contain sufficient information to view the related video frame but they have the advantage of requiring significantly less resources when stored or transmitted. P and B frames can be decoded in the context of

GOP. Ideally a GOP should represent a similar continuous related scene. The encoders mostly use fixed GOP size to encode video sequences. A fixed encoder can operate with different size of GOPs but once a target size for the GOPs is selected, the same size is applied to the whole coded sequence. Fixed encoders are easy to implement but they prevent the encoding process to be adaptive to changes in video sequences due to i) scene cuts (abrupt/gradually), ii) changes of video capturing settings e.g. camera focus and iii) degradation of frame quality based on transmission noise.
Challenge 1: Bandwidth is limited: Imagine a video from a surveillance camera of a parking lot. Except from the movement of a car or a passenger all the remaining scene remains static over times. It is expected that the surveillance video demonstrates ”limited” activity thus frequent transmission of I frames is not needed, which in turn require network resources and energy. In contrast a football match contains many scene changes because the camera or the objects in the scene are constantly in movement, which logically corresponds to frequent I frames. If scenes with small video content variance, e.g. parking lot, are coded with the same GOP structure frequency with high rate changing frames, e.g. football match, this would lead to a considerable waste of network resources. Constant rate of I frame generation from fixed encoders requires significantly more bandwidth than the actually needed to support the considered multimedia applications.

Challenge 2: Video Streaming in 'accepted' quality: Scene changes can be divided into two categories: abrupt and gradual. The difference between abrupt and gradual scene changes lies in the number of frames needed to conclude the change. If the change is contained only in one frame it is defined as a abrupt scene change. Gradual scene change involves several frames to complete the transition from one scene to another. Encoding process is influenced by GOP structure because it is based on predictive coding techniques along the temporal axis such as motion prediction and compensation. If I-frames are created independently of scene changes then encoding efficiency will suffer from severe error drifting on video transmission. Again high-rate changing frames should be shorter than slow motion videos in order to achieve better coding efficiency.

3.1.1 Contribution

We propose a model of dynamic encoder adaptive to changes in video sequences by dynamically adjusting GOP size based on an Optimal Stopping Theory (OST) rule in order to transmit video sequences in an acceptable quality with the simultaneous rational use of WSMN resources. More specifically we report:

1. what it is defined as a scene change problem and quantify this event
2. the optimal stopping rule for the discussed problem and how it is applied
3. the performance evaluation of the proposed scheme.

This chapter is organized as follows. In Section 3.2 we present the preliminaries needed for both GOP structure, OST formulation problem and the description of our solution. Sec-

tion 3.3 presents the experiments performed and the corresponding discussion followed by the conclusions in section 3.4. Experiments were conducted by using several types of slow and fast motion video samples from a media library [28].

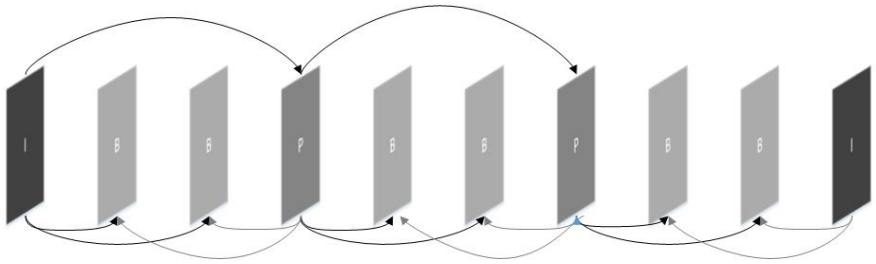


Figure 3.1: GOP in the H.263 video flow

3.2 System Design of Time-optimized Grouping-of-Pictures

3.2.1 Group-Of-Pictures structure

The main goal of the MPEG standard is to compress a video sequence to a fraction of the original prior to transmission or storage. This is achieved by transmitting the changes between frames, which are sampled at specific time intervals, and not the whole sequence of frames. The basic processing blocks in the encoder, shown in Figure 3.2, are Discrete Cosine Transform (DCT) coefficient quantizer, run-length amplitude / variable length coder, and block-based motion compensated prediction, using motion estimation.

Starting with the first frame of a Group-Of-Pictures (GOP), an I (intracoded) frame is created. The encoder can predict a target frame. This is commonly referred to as a P (Predicted) frame, and it may also be predicted from other P frames, although only in a forward-time manner. Each P frame in a sequence is predicted from the frame immediately preceding it, whether it is an I frame or a P frame. Note that, I frames are autonomously compressed spatially with no reference to any other frame in the sequence. The temporal prediction technique used in MPEG video is based on motion estimation. The basic assumption of motion estimation is that, in most cases, consecutive video frames will be similar except for changes induced by objects moving within the frames. In the trivial case of zero motion between frames (and no other differences caused by noise), the encoder predicts the current frame as a duplicate of the prediction frame. When this is done, the only information necessary to transmit to the decoder becomes the syntactic overhead, which is necessary to reconstruct the picture from the original reference frame.

3.2.2 Rationale

A prediction scheme inside the encoder is used in order to foresee any scene changes. In our case, it is assumed that each GOP structure can be large and finite. Each next

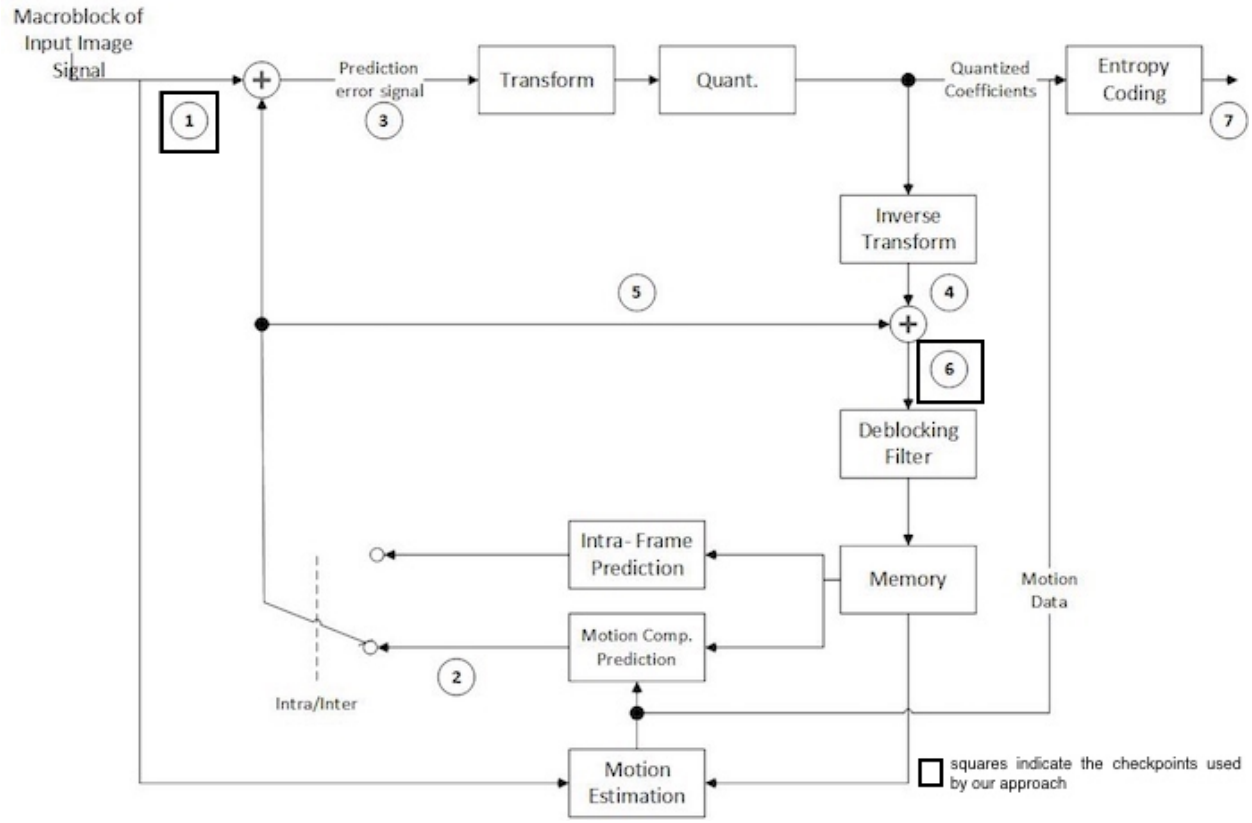


Figure 3.2: MPEG Encoder

frame is encoded as a P frame at discrete time step $t \in \{1, \ldots, n\}$. At time instance $t = 1$ an I frame F_I is constructed and in $t = N$ the last frame F_{P_N} is created. The main goal is to continue to add P frames into the same GOP sequence, if and only if a scene change does not occur. At this point we must 'quantify' a scene change. Based on the definition provided in [29] let us consider a video frame F_{C_t} coming inside the encoder encoder at the checkpoint (1) of Figure 3.2. A P frame F_{P_t} is encoded using the motion vectors between the previous reference frame and the current frame F_{C_t} inserted in the MPEG encoder and the output is an encoded frame F_{P_t-enc} which mainly contains the differences between F_{C_t} and the previously I or P frame as shown in Figure 3.1. This bit-stream is sent to the decoder. The decoder based on these differences creates the new frame F_{DC_t} exiting from checkpoint (6) in figure 3.2. The metric indicating a possible scene change is defined as the sum of absolute differences [29] between the two frames (SATD) F_{C_t} and F_{DC_t} where $F_{i,j}^{F_{C_t}}$ is the pixel value at location (i, j) of frame F_{C_t} and I_w and H_h are the width and height of a frame, respectively:

$$SATD(F_{C_t}, F_{DC_t}) = \sum_{i=0}^{I_w-1} \sum_{j=0}^{H_h-1} |F_{i,j}^{F_{C_t}} - F_{i,j}^{F_{DC_t}}| \tag{3.1}$$

As a decision maker, we desire to get as close as possible to a given limit in which a scene

change occurs but the limit should not be exceeded. Given the incoming values of SATD between the incoming and the outgoing frame, i,e, checkpoints 1 and 6 from the encoder encoder in Figure 3.2, we would like to find the closest distribution which fits the data. We used a distribution comparison function which returns the fit of all valid parametric probability distributions to the input data and plot the Probability Density Functions (PDFs) to compare them graphically. In our case we can see the results of the function in Figure 3.3. In this work we will deal with the two most prevalent distributions, i.e. gamma and normal distributions.

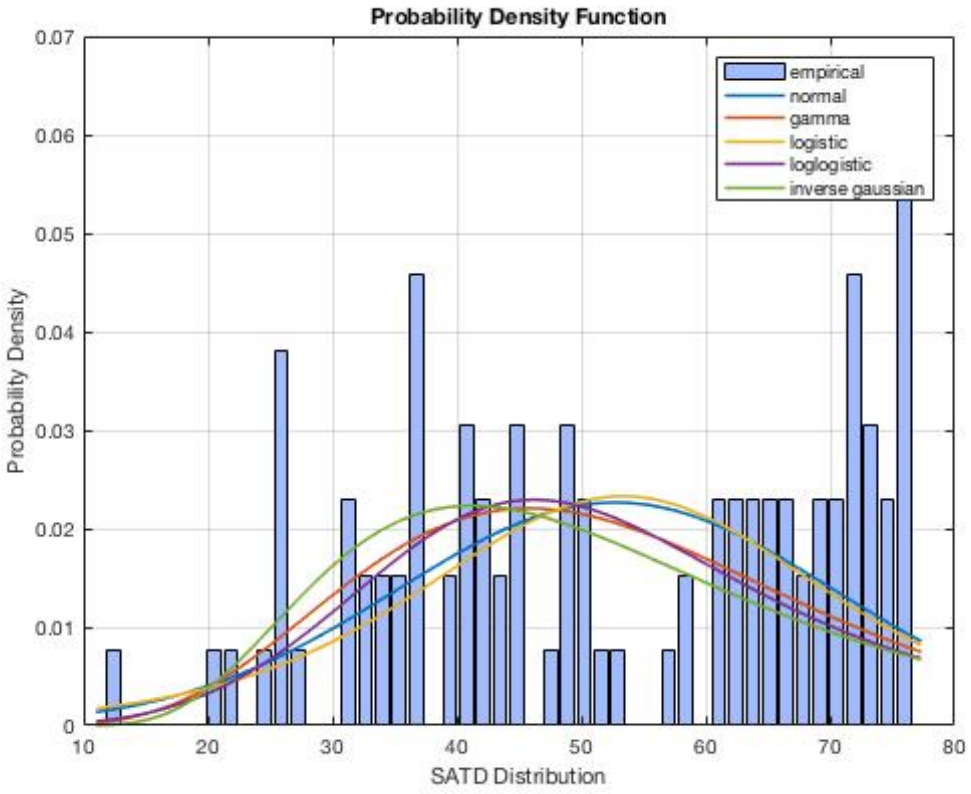


Figure 3.3: Fitting probability distribution functions based on the actual SATD values.

3.2.2.1 Gamma Distribution

Specifically, let us consider that $S_1, S_2, \cdots, S_n$ be a sequence of sequentially observed random variables having a gamma distribution $\Gamma(\alpha, \beta)$ where $\alpha, \beta > 0$ and each corresponds to $S_n = SATD(F_{C_n}, F_{DC_n})$ at time instance (step) $t = n$:

$$f(s \mid \alpha, \beta) = \frac{\beta^{\alpha} s^{\alpha-1} e^{-s\beta}}{\Gamma(\alpha)} \tag{3.2}$$

The encoder observes the random sequence $\{S_1, \ldots, S_n\}$ and decides whether to 'stop'

or to 'continue'. The encoder wants to pull as many frames as possible. If the encoder decides to stop at the moment *n*, then it will gain a real-valued pay off $(y + \sum_{i=1}^{n} S_i)$, if the sum $\sum_{i=1}^{n} S_i$ is not greater than a specified threshold *T*. The threshold T corresponds to cumulative error when a scene changes occurs. If the encoder passes the limit T, then the gain is zero. A given nonnegative real number *y* appearing in the above gain definition is another characteristic of the problem and may be interpreted as an initial state of the process of observations. Formally, we consider a Markov chain $(Y_n, \mathbb{F}_n)$ for $n = 1, \ldots, N$ with

$$Y_n = y + \sum_{i=1}^{n} S_i, \tag{3.3}$$

with $\mathbb{F}_n$ being generated by the observations $S_1, S_2, \ldots, S_n$ and $y > 0$. We define as pay off the real-valued function $W(y) \in \mathbb{R}$ such that:

$$W(Y_n; y, T) \;=\; \begin{cases} (y + \sum_{i=1}^{n} S_i) & \text{, if } y + \sum_{i=1}^{n} S_i \leq T \\ 0 & \text{, otherwise,} \end{cases} \tag{3.4}$$

with error tolerance threshold $T > 0$. The threshold T indicates the tolerance of the encoder to *delay* the cumulative sum of the frame variations in light of pulling as many fames as possible. However, the sum of those variations is stochastic, thus, the encoder has to find an optimal rule for stopping the surge of the random sum just before reaching its maximum tolerance value T. Based on the pay off $W(Y_n)$ with initial state $y > 0$ and tolerance threshold T, we define our optimal stopping time problem for the encoder:

Problem 1. Given observations of SATD values $\{S_1, \ldots, S_n\}$ and tolerance cumulative sums $Y_1 = y + S_1, Y_2 = y + S_1 + S_2, \ldots, Y_n = y + \sum_{i=1}^{n} S_n$, find the optimal stopping time t^* to maximize the expected pay off $\mathbb{E}[W(Y_{t^*})|Y_1]$ where the pay off is defined in (3.4).

3.2.3 Solution Fundamentals

Before proceeding with a solution of Problem 1, we refer to the Proposition 1 to analyze the expectation of the optimal value $V_n(y)$.

Proposition 1. If there exists a real number t^*, $0 \leq t^* \leq T$ such that the conditions of Theorem 1 hold true, the optimal value $V_n(y)$ is calculated for $y < t^*$ as follows, where $n = 2, \ldots, N$:

$$V_n(y) = \int_0^{t^*-y} V_{n-1}(y+s) f(s) ds + \int_{t^*-y}^{\infty} W(y+s) f(s) ds \tag{3.5}$$

with the initial condition $V_1(y) = \int_0^{\infty} W(y+s) f(s) ds$.

Proof: This derives immediately from the principle of optimality in Theorem 1 by taking the expectation of the optimal value of $V_n(y)$.

Let us now provide a solution to Problem 1. We need first to find the form of $V_n(y) = \mathcal{Q}^n(y)$, $n = 1, \ldots, N$. By definition of the operator $\mathcal{Q}$, we have for every $y \in (0, T]$:

$$\begin{aligned} \mathcal{Q}W(y) &= \max\{W(y), \mathbb{E}[W(Y_1)]\} = \max\{W(y), \mathbb{E}[W(y + S_1)]\} \\ &= \max\{W(y), \int_0^\infty W(y+s) f(s \mid \alpha, \beta) ds\} \\ &= \max\{W(y), \mathbb{I}_1(y)\} \end{aligned}$$

For $y < T$ and given Proposition 1, the integral function $\mathbb{I}_1(y) = \int_0^\infty W(y+s) f(s \mid \alpha, \beta) ds$ is expressed as follows:

$$\begin{aligned} \mathbb{I}_1(y) &= \int_0^\infty W(y+s) f(s) ds = \\ &= \int_0^{T-y} (y+s) f(s) ds + \int_{T-y}^\infty 0 f(s) ds \\ &= \int_0^{T-y} (y+s) \frac{\beta^\alpha s^{\alpha-1} e^{-s\beta}}{\Gamma(\alpha)} ds \\ &= \int_0^{T-y} y \frac{\beta^\alpha s^{\alpha-1} e^{-s\beta}}{\Gamma(\alpha)} ds + \int_0^{T-y} s \frac{\beta^\alpha s^{\alpha-1} e^{-s\beta}}{\Gamma(\alpha)} ds \\ &= \frac{1}{\Gamma(a)} (y \beta^\alpha (T-y)^\alpha (\beta(T-y))^{-\alpha} (\Gamma(\alpha) - \Gamma(\alpha, \beta(T-y))) \\ &+ \beta^{\alpha-1} (T-y)^\alpha (\beta(T-y))^{-\alpha} (\Gamma(\alpha+1) - \Gamma(\alpha+1, \beta(T-y)))) \end{aligned} \tag{3.6}$$

Figure 3.4 shows an exemplary graph of the pay off function $W(y)$ and the integral function $\mathbb{I}_1(y)$ for $y \leq T$.

It is easy to verify that for any given tolerance threshold T the functions W and I_1 have equal values at $t_1 \in (0, T]$ at which the function $\mathbb{I}_1$ takes its only maximum on the interval $(0, T]$ because $\mathbb{I}_1(y) > W(y)$ for $y \in (0, t_1)$ and $\mathbb{I}_1(y) < W(y)$ for $y \in (t_1, T]$. Then the value of t_1 is estimated by solving the following equation:

$$\mathbb{I}(t_1) = \int_0^{T-t_1} W(t_1 + s) f(s) ds = W(t_1), \tag{3.7}$$

and depends on the probability density function $f(s)$ and tolerance threshold T. Given the pay off function in (3.4), we obtain t_1 by solving the equation:

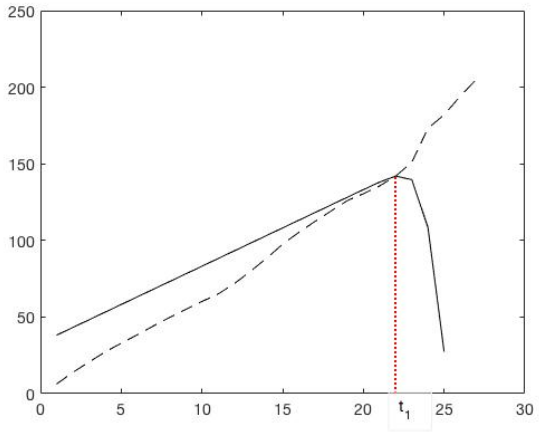


Figure 3.4: Graphs of functions $\mathbb{I}_1$ (continuous line) and W (dotted line), T=30

$$\begin{aligned} \int_0^{T-t_1} (t_1+s)f(s)ds &= t_1 \Leftrightarrow \\ t_1 \int_0^{T-t_1} f(s)ds + \int_0^{T-t_1} sf(s)ds &= t_1 \Leftrightarrow \\ t_1 \frac{(1-F_S(T-t_1))}{F_S(T-t_1)} &= \mathbb{E}[S|S \leq T-t_1], \end{aligned} \tag{3.8}$$

where $F_S(x) = P(S \leq x) = \int_0^x f(s)ds$ is the cumulative probability function of S and $\mathbb{E}[S|S \leq T-t_1]$ is the conditional expectation of S given that $S \leq T-t_1$. The optimal value function $V_1 = \mathcal{Q}W$ is the maximum of the two ones presented in Figure3.4. Based on the optimality in Theorem 1, one step before the end of the observations the decision maker should continue the observations if it is at any state y which is less than t_1 and should stop otherwise. Obviously the functions $\mathbb{I}_1$, *W* and V_1 are equal to 0 for arguments greater than T.

Given that $\mathbb{I}_1(y)$ denotes the expectation $\mathbb{E}[V_{n-1}(y+S_1)], n = 1 \cdots N$, we provide Proposition 2 that holds true for any integral function $\mathbb{I}_n(y)$, $n = 1 \cdots N$ by induction.

Proposition 2. For any natural number n, for t_1 derived from (3.7), and for every $T > 0, \alpha > 0, \beta > 0$ the integral function $\mathbb{I}_n(y)$ satisfies the following conditions:

1. $\mathbb{I}_n(y) > W(y)$ for $y \in (0, t_1)$
2. $\mathbb{I}_n(y) < W(y)$ for $y \in (t_1, T]$
3. $\mathbb{I}_n(y) = 0$ for $y > T$

Proof. The conditions (1),(2) and (3) for $n = 1$ derive from Proposition 1. Now, let us assume that the conditions (1)–(3) hold for $\mathbb{I}_{n-1}(y)$. Then, by definition of $V_{n-1}(y)$ and by induction assumption for $y \in (0, t_1)$ we obtain:

$$\begin{aligned}
\mathbb{I}_n(y) &= \int_0^{\infty} V_{n-1}(y+s)f(s)ds \\
&= \int_0^{t_1-y} I_{n-1}(y+s)f(s)ds + \int_{t_1-y}^{T-y} W(y+s)f(s)ds + \int_{T-y}^{\infty} 0 \cdot f(s)ds \\
&\geq \int_0^{t_1-y} W(y+s)f(s)ds + \int_{t_1-y}^{T-y} W(y+s)f(s)ds = \mathbb{I}_1(y) > W(y)
\end{aligned}$$

Hence the condition (1) is satisfied. In addition, condition (2) is satisfied when $y \in [t_1, T)$ since we obtain that:

$$\mathbb{I}_n(y) = \int_0^{\infty} V_{n-1}(y+s)f(s)ds = \int_0^{T-y} W(y+s)f(s)ds = \mathbb{I}_1(y) < W(y) \tag{3.9}$$

The condition (3) is obvious, thus, the proof of Proposition 2 is completed.

It follows from Proposition 2 immediately that for $n = 1, \ldots, N$, the optimal values $V_n(y)$ have the form:

$$V_n(y) = \mathbb{I}_n(y)\mathbf{1}_{(0,t_1]}(y) + W(y)\mathbf{1}_{(t_1,T]}(y), \tag{3.10}$$

where the value of t_1 is provided in (3.7). Based on this, we provide the optimal stopping rule for the Problem 1:

Proposition 3. Given a sequence of SATD realizations $S_1, \ldots, S_N$ with probability density function $f(s)$ and pay off function $W(Y_n; y, T)$ defined in (3.4) with cumulative sum $Y_n = y + \sum_{i=1}^{n} S_i$, the optimal stopping rule t^* for the Problem 1 with initial state y is given by:

$$t^* = \min\{0 \leq k \leq N : Y_k = y + \sum_{i=1}^{k} S_i \geq t_1\}, \tag{3.11}$$

where t_1 is estimated in (3.7).

Proof: The result follows directly from Theorem 1 and Proposition 2.

From Proposition 3, the optimal stopping rule model is interpreted as follows: the encoder continues to observe, i.e. add P frames in the GOP sequence, as long as the sum of the initial state y and the sum of already observed values s_i do not exceed the value t_1. Hence, we have to compute the threshold value t_1 which requires the estimation of the probability density function $f(s)$ given a tolerance threshold T. In case that SATD follow the gamma distribution $\Gamma(\alpha, \beta)$, then the integral function $\mathbb{I}_1$ is directly provided in (3.6) and t_1 is obtained by solving the equation in (3.7).

Remark 1. It is worth mentioning that the optimal value $V_N(y)$ of Problem 1 is inductively calculated for $y < t_1$ from the recursive equation:

$$V_n(y) = \int_0^{t_1-y} V_{n-1}(y+s)f(s)ds + \int_{t_1-y}^{T-y} (y+s)f(s)ds,$$

where the initial condition is given by $V_1(y) = \mathbb{I}_1(y)$ for any $N > 0$ and $n = 2, 3, \ldots, N$.

3.2.3.1 Normal Distribution

The normal distribution is investigated as the probability distribution that fits the actual SATD values. However, error figures are limited to values greater than zero. Let $S \sim N(\mu, \sigma^2)$ follow a normal distribution and lie within the interval $S \in [0, +\infty)$. Then, the random variable S conditioned on the interval $[0, \infty)$ is described by the *truncated* probability function:

$$f(s \mid \mu, \sigma, 0, \infty) = \frac{\frac{1}{\sqrt{2\pi}} e^{\frac{-1}{2}(\frac{s-\mu}{\sigma})^2}}{\sigma(\Phi(\infty) - \Phi(\frac{0-\mu}{\sigma}))}, \text{with } \Phi(x) = \frac{1}{2}(1 + erf(x/\sqrt{2})). \tag{3.12}$$

By definition, $\Phi(\infty) = 1$ and, thus, the probability function is re-written as:

$$f(s \mid \mu, \sigma, 0, \infty) = \frac{1}{\sqrt{2\pi}\sigma(1 - \Phi(\frac{-\mu}{\sigma}))} e^{\frac{-1}{2}(\frac{s-\mu}{\sigma})^2} \tag{3.13}$$

According to the truncated normal distribution and Preposition 1 for $y < T$, the integral function $\mathbb{I}_1(y) = \int_0^{\infty} W(y+s)f(s \mid \mu, \sigma)ds$ is expressed as follows:

$$\begin{aligned}
\mathbb{I}_1(y) &= \int_0^{\infty} W(y+s)f(s)ds = \\
&= \int_0^{T-y} (y+s)f(s)ds + \int_{T-y}^{\infty} 0f(s)ds \\
&= \int_0^{T-y} (y+s)\frac{1}{\sqrt{2\pi}\sigma(1 - \Phi(\frac{-\mu}{\sigma}))} e^{\frac{-1}{2}(\frac{s-\mu}{\sigma})^2} ds \\
&= (\frac{1}{\sqrt{2\pi}\sigma(1 - \Phi(\frac{-\mu}{\sigma}))})(\int_0^{T-y} ye^{\frac{-1}{2}(\frac{s-\mu}{\sigma})^2} ds + \int_0^{T-y} se^{\frac{-1}{2}(\frac{s-\mu}{\sigma})^2} ds) \\
&= y\sqrt{\frac{\pi}{2}}\sigma(erf(\frac{\mu}{\sqrt{2}\sigma}) - erf(\frac{\mu - T + y}{\sqrt{2}\sigma})) + \sigma\frac{1}{2}(-\sqrt{2\pi}\mu erf(\frac{\mu - T + y}{\sqrt{2}\sigma}) + \\
&+ \sqrt{2\pi}\mu erf(\frac{\mu}{\sqrt{2}\sigma}) + 2\sigma(e^{\frac{-\mu^2}{2\sigma^2}} - e^{\frac{-(\mu-T+y)^2}{2\sigma^2}}))
\end{aligned} \tag{3.14}$$

Type	α	β	μ	σ	T
Slow motion Video	16.50761	0.07891	0.9766	0.6694	45
Medium motion Video	4.516779	2.99732	7.5131	2.2424	25
Fast motion	7.5712	6.96713	22.6879	4.7797	12

Table 3.1: α and β values for different types of video

It is easy to verify that, for any given tolerance threshold T, the functions W and I_1 have equal values at $t_1 \in (0, T]$ at which the function $\mathbb{I}_1$ takes its only maximum on the interval $(0, T]$ because $\mathbb{I}_1(y) > W(y)$ for $y \in (0, t_1)$ and $\mathbb{I}_1(y) < W(y)$ for $y \in (t_1, T]$. Then, the optimal stopping rule t^* is derived from Proposition 3.

3.2.4 Complexity & Model Design Parameters

The complexity of the encoder for triggering the optimal stopping rule as derived from Proposition 3 is based on the calculation of the current SATD value. Specifically, the SATD value calculation requires $O(I_w H_h)$ time since I_w and H_h are the width and height of a frame. The encoder then increases the current summation Y_n at step n by the new S_n SATD value, which is achieved in $O(1)$. If this sum exceeds the optional threshold t_1 provided by Proposition 3, then the encoder is triggered. Hence, the overall complexity for the decision making requires $O(I_w H_h)$.

The design parameters of the problem are the following: the limit (threshold tolerance) T of the cumulative sum of inter-frame deviations, and α, and β fitted parameters of the Gamma distribution. Let us assume that the initial state y of the process equals to 0, i.e., after each triggering of the encoder, and let us confine ourselves to this situation where the value of the problem $V_N(0)$ is positive, i.e., the decision maker (encoder) should make at least one observation (receives at least one frame). Using sample videos from the Test Media Library [28], we have tried to evaluate the design parameters of our approach in different MPEG streams with different needs. For example, a video containing only one shot of a waterfall from a stable camera reception can be characterized as a *slow* motion video. In contrast, a sequence from a football match can be considered as a *fast* motion video. A *medium* motion vector can be defined as a man who is talking to the camera by moving his head. For these three different examples of motion videos α, and β values of the Gamma distribution were computed and presented in Table 3.1.

3.3 Performance Evaluation

3.3.1 Simulation setup

The simulation setup has as follows: we have used an MPEG-2 simulator. This simulator is based on the work presented in [27]. MPEG-2 simulators is enhanced with additional functions in order to support the creation of GOPs with dynamic size based on an OST rule.

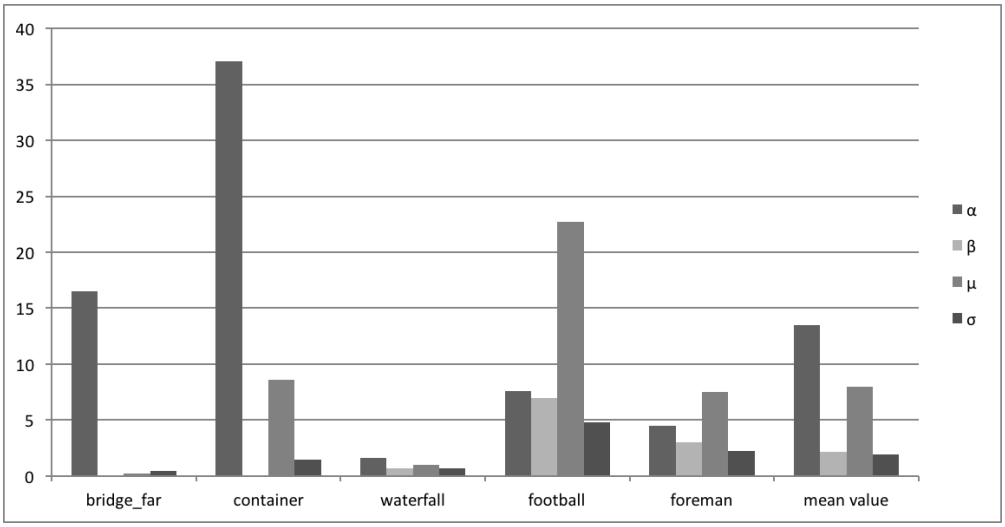


Figure 3.5: α, β values of Γ distribution for the set of videos used in our experiments

The performance metrics of the proposed encoder with dynamic grouping of pictures of GOPs adapted to stream behavior are i) the produced error of the encoding process and ii) the size of generated video stream in bits. In this way we are trying to map the two challenges referred to Section 1 related to limited bandwidth and the 'quality' of the derived video stream with the experimental results. The metric related to the quality is SATD shown in equation 3.1 and was measured between the control points (1) and (6) as depicted in MPEG encoder at Figure 3.2. The dynamic grouping of pictures method is compared with a classic fixed-length version of an MPEG-2 encoder which creates a GOP with one I frame and then adds a constant number of P frames e.g. IPPPPPPPPPP. In our case the length of P frames is equal to 10. The pool of videos is downloaded from [28]. Every video was examined in a sequential stream of 100 frames. A short description of the videos follows to illustrate the dynamic character of streams:

1. bridge-far: a slow motion video showing a bridge from remote;
2. waterfall: a slow motion video with the constant recording of a waterfall;
3. hall-objects: a fast motion video from a camera in an office corridor. At some point two people walk in;

4. highway: a shooting of a vacant highway recorded by a camera in a car - medium motion video;
5. foreman: a person talking to camera - medium motion video;
6. football: a fast motion video from a football match;
7. container: a fixed camera showing the course of a tanker - fast motion video;

For each of these videos α, β, μ and σ values are presented in Figure 3.5. The presented values are generated by a single GOP with one I frame and an "infinite" number of P frames. A single GOP of a video stream can provide us with a holistic overview of the SATD of frames. The approach that we follow in order to configure α, β, μ and σ values is the following. A pool of twenty different kind of videos was analyzed and the mean values of the aforementioned parameters were extracted. These mean values are the initial α_{mean}, β_{mean}, μ_{mean} and $\sigma mean$ values when the encoder starts to operate, i.e. $t = 0$. The OST rule for the first incoming frames is based on these initial values. User can select gamma or normal functionality for implementing the dynamic encoding module. We use the abbreviation DGPE describing the dynamic grouping of pictures encoder for the gamma distribution and NDGPE describing the normal distribution. The time when the first GOP concludes, α and β or μ and σ values are re-calculated fitting in the cumulative SATD of the already processed video stream, i.e. GOP=1.

3.3.2 Discussion of Simulation results

The results of the simulations are described below. The classic encoder (CE) created 10 fixed length GOPs. The number of GOPS created by DGPE and NDGPE are depicted in table 3.2. In the same table we compare the total size transmitted for each video $(inbits)$ from the CE and DGPE encoders. We can notice that in slow motion videos the GOP size is extended in order to avoid unnecessary transmissions of I frames. For example in the waterfall video the number of GOPs is reduced to 2 and 3 per 60 frames in DGPE and NDGPE respectively. In contrast in fast motion video the GOPs created are increased while the size of the generated bitstream stays belows the generated bitstream of CE in average. It can be noticed that the volume transmitted in most of the cases from dynamic encoder is smaller than classic encoder. This is expected as fixed encoders are not content-driven and lead to waste of bits and resources. By comparing the dynamic encoders, we may notice that DGPE is more "sensitive" in fast-motion videos by capturing more scene changes than DGPE while DGPE shows tolerance to the medium motion videos.

In addition, through Figures 3.6, 3.7, 3.8 and 3.9 we provide a comparison overview of SATD measured between CE and DGPE. In Figure 3.6 it is observed that the error values coming from CE are higher than the dynamic encoders DGPE and NDGPE. The median value of SATD corresponds to 107.4 for CE. The median value of DGPE is 27.56 and 23.52 of NDGPE. The fewer GOPs created by truncated normal encoder also corresponds to

Video	$DGPE_{GoPs}$	$NDGPE_{GoPs}$	$DGPE_{Size}$	CE_{Size}	$NDGPE_{Size}$
bridge	8	7	832477	876182	841583
waterfall	2	3	1499705	1642998	1500144
hall	9	15	1019986	1032109	1098023
container	15	11	2158422	2017824	2084643
foreman	6	9	2705621	2819314	2847462
football	27	13	6608428	6510336	6288473

Table 3.2: Size of bitstreams transmitted in network

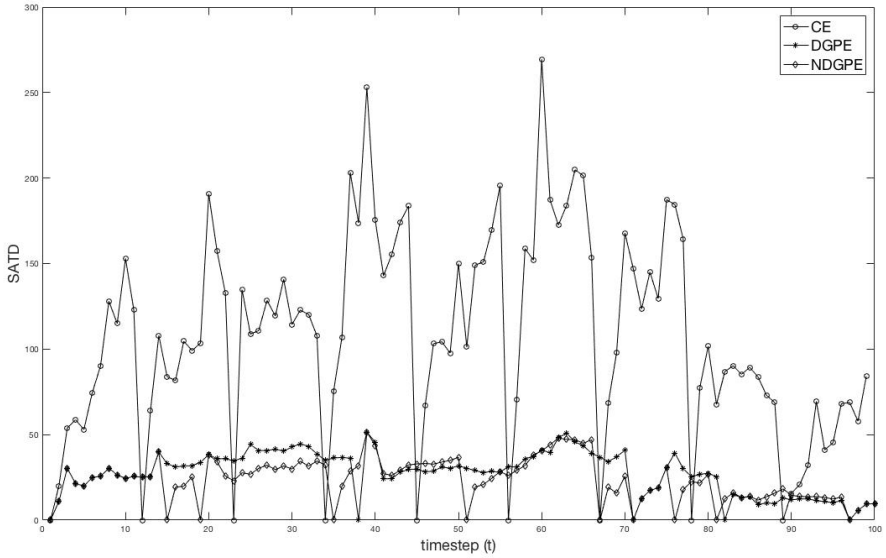
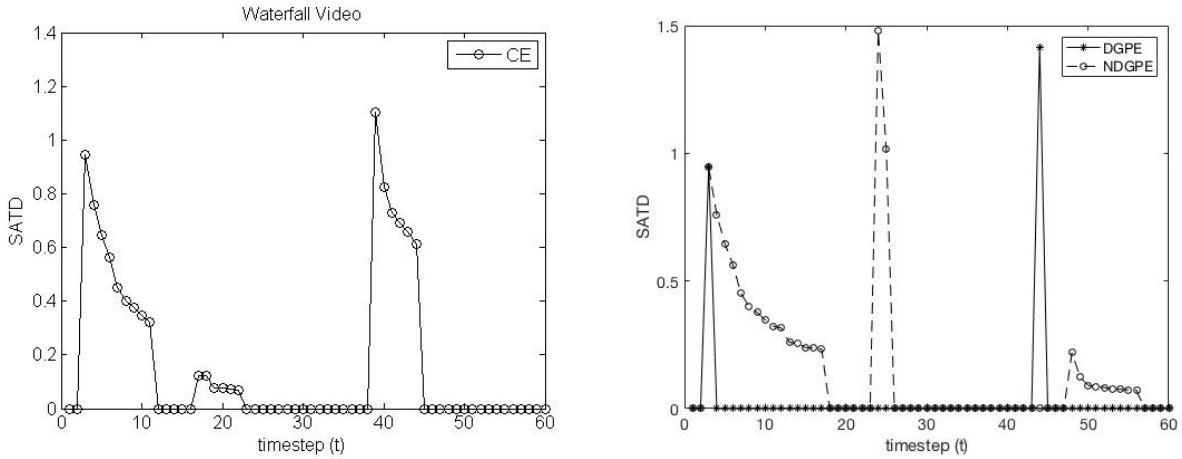


Figure 3.6: SATD between classic approach and OST- football video

a reduction of 4% in the total transmitted volume of bits as shown in table 3.2. In Figure 3.7, DGPE has the best video stream performance. The error remains close to the zero values. The first GOP is based on initial mean values of α and β and the next GOPs are based on the refitting of the design values to the incoming data distribution. NGOE needs time to fit μ and σ values to the slow motion video distribution. The mean and std values of the output error are the following: DGPE$\{0.0394, 0.2177\}$ and NDGPE$\{0.1625, 0.2934\}$.

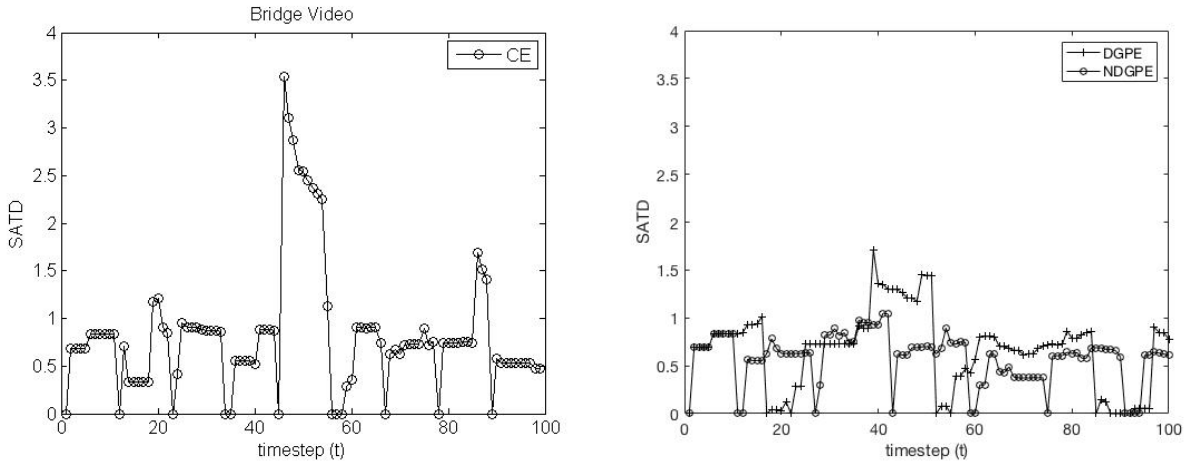
From the results in Figures 3.8 and 3.9 the encoder which uses normal distribution to compute t^* performs better than the other assessed encoders. We can notice that NDGPE needs more time to be adaptive to the changes of the incoming distribution but then SATD error values generated between the frames in the GOP created correspond to small values. For example at hall video the error values after the first 30 frames are quite low when compared with DGPE and CE methods

From the description above, it shown that the dynamic encoders perform better than the



(a) SATD of CE (b) SATD of DGPE and NDGPE

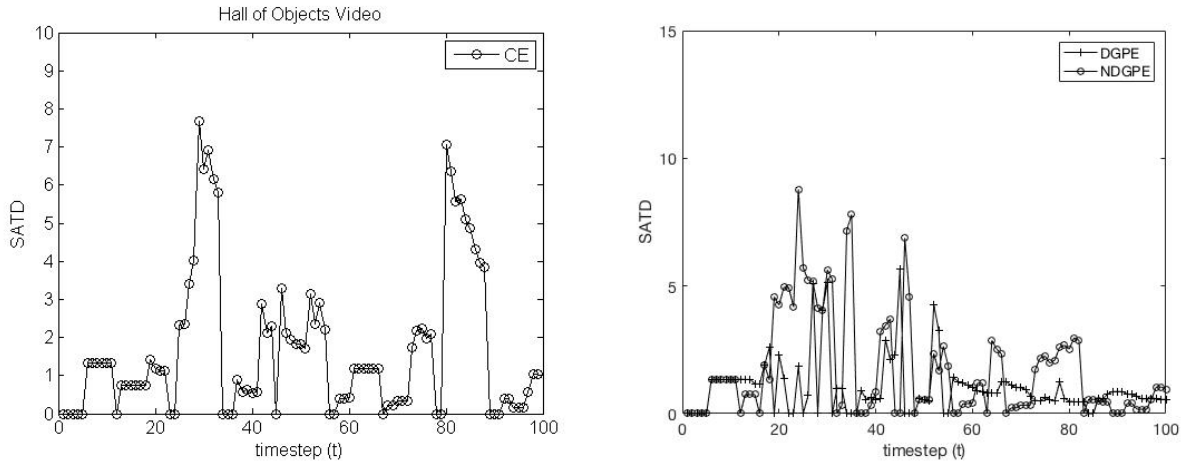
Figure 3.7: SATD between classic approach and OST - waterfall video



(a) SATD of CE (b) SATD of DGPE and NDGPE

Figure 3.8: SATD between classic approach and OST - bridge video

fixed length encoder. The notion of adoption to video content is important as I frames are depended on scene changes and thus the encoding efficiency suffers from the error drifting on video transmission. The NDGPE shows better performance but a number of training incoming frames are required to fit the data distribution. If video streaming is quite short then the gamma-based encoder DGPE is the better candidate.



(a) SATD of CE

(b) SATD of DGPE and NDGPE

Figure 3.9: SATD between classic approach and OST - hall video

3.4 Conclusions

In this chapter, we focus on content-based MPEG encoder and propose an OST decision rule for the conclusion of GOP and the transmission of intra- coded frames. Dynamic encoding applied to infrastructures with restricted resources, like IoT camera networks, is needed in order to support media-rich applications in such infrastructures. Limited bandwidth and battery lifetime require nowadays content-driven transmission rates and processing of the video sequences. One major contribution is the adaptation to video changes; I frames are created when scene changes are detected which leads to significant resource savings while retaining equal quality levels. Our encoder can be applied to facilities with restricted resources like WSMNs in order to transmit video sequences in an acceptable quality. The aim is twofold: to create different size of GOPs adaptive to the transmitted video streams and to try to save resources with a small SATD error. Experiments show that the GOP size was extended in order to avoid unnecessary transmissions. We observe that the stream volume transmitted in most of the cases is smaller than the CE created bitstream which justifies that fixed encoders which are not content-driven lead to waste of network resources. The encoder focuses on the transmitted video content and, thus, the values of SATD stay lower than the classic approach.

4. REAL-TIME STOCHASTIC CONTROL MECHANISM ADAPTIVE TO CHANGES IN NETWORK QUALITY BASED ON OPTIMAL SEQUENTIAL DECISION MAKING RULES.

4.1 Motivation

Going a step further to the edge, we continue our research study to robotic devices and how changes in wireless network conditions can affect the control and the exchange of telemetry messages. We can consider a *drone* as a mobile computing and sensing node deployed to different locations tailored to specific tasks. The fundamental features that 'transform' Unmanned Vehicles to popular mobile IoT nodes are the ability to autonomously make decisions (i.e., without human intervention), the capability of carrying additional application-specific payloads, the endurance, capability of re-programmability, and capacity to stream locally sensed/captured multimedia content. As Unmanned Vehicles become more advanced in terms of computational capabilities, they are expected to present greater value in application cases of e.g., environmental surveillance and monitoring, and supporting crisis management activities. For instance, consider the use case where drones equipped with video camera and various sensors, like air-quality, humidity and temperature, are programmed to cruise over forests and spot fires at an early stage.

The ultimate target of an Unmanned Vehicle, also coined as UxV, where 'x' can stand for either 'A' aerial, or 'G' ground, or 'S' surface vehicle, is the successful execution of a pre-programmed mission. A mission is often described as a trajectory with specific way-points that the UxV is tasked to approach and collect various measurements, e.g., from on-board sensors, or capture images or video, e.g., from on-board cameras. The way-points along with the various commands are determined from a control unit, i.e., a Ground Control Station (GCS). A GCS is a remote coordinator (master) node responsible for contextual data acquisition and real-time control and monitoring of the progress of the UxVs missions. The communication between UxV and GCS is realised in a wireless manner. The UxVs themselves can be either involved in a mission as single/individual units or as groups, i.e., swarm of UxVs. A swarm of UxVs forms a remote sensing system and can be treated as Mobile Wireless Sensor Network (MWSN) of highly dynamic topology. More importantly, the on-board computing & sensing elements of the UxVs enhance the in-network embedded intelligence of the swarm. This allows complex local computational and analytics tasks to be realized in a highly distributed fashion, thus, balancing computational load across the infrastructure and render communications much more energy efficient. In this Mobile IoT (MIoT) environment of UxV-driven distributed computing, we are facing the following research and technical challenges:

Challenge 1: Real-time Monitoring. Real-time surveillance and monitoring applications, e.g., detection of forest fires, require control messages to be delivered from a *swarm* of UxVs to the GCS with the *minimal delay and high accuracy*. These missions typically involve rural areas, where the network connectivity is expected to be poor [21]. Moreover, radio paths between the UxVs and GCS are anticipated to be obstructed, overloaded or

to suffer from high packet loss rate. It is challenging to predict these network variations in these environments. Hence, it is deemed crucial, during a mission, an UxV to autonomously decide *when to pause* telemetry/control measurements that are not currently prioritized as 'important' and save network resources.

Challenge 2: Secure UxV Control & Actuation. The connectivity among UxVs and GCS needs to take into consideration the *mobility* factor. This factor adds up a new degree of freedom to their operation, since the GCS sends control commands to UxVs while UxVs are moving for further local actuation. The control messages and their acknowledgements must be securely delivered in order to guarantee safe and successful missions. The usual approach to emergency cases, when a UxV loses the connection to GCS, is that the UxV returns to its initial position abandoning the mission. This means that the mission is cancelled, even if the UxV could be really close to the mission's end or objective leading to significant waste of time and resources.

In this thesis work, we cope with the above-mentioned challenges by proposing an on-line stochastic-driven decision making scheme that leverages the transmission functionality of UxVs and GCS by being adaptive to changes in network quality. This is designed and developed by our novel suppression control of telemetry and control messages model based on the principles of the Optimal Stopping Theory (OST). Our time-optimized control mechanism achieves the *optimal delivery* of critical information from UxVs to GCS and vice-versa. Our rationale is that should the network be performing properly, then the transmission control can be 'relaxed' to exploit the available resources in the resource-constrained UxV. Our model introduces two sequential optimal stopping time decision making mechanisms based on the Change Detection theory and an application-specific discounted reward process.

We consider the case where a UxV operator desires to execute a mission and consider the setting where two main components are provided: a GCS and an Unmanned Ground Vehicle (UGV). The mission instructions could be consolidated in a domain-specific script, e.g., the mission scripts compiled through our experimentation platform for UxVs RAWFIE [31]. The RAWFIE[1] platform is briefly presented in Section 4.3.3. The mission script defined by the operator includes (among others) the UxV trajectory way-points in the field area to control the device in space and time and the sensing components involved (sensors) to collect in-field measurements. The main goal of the two components is the monitoring of an area to detect fire based on camera stream and on-board environmental sensors. This use case was also conducted during the RAWFIE project lifetime.

The baseline solution/establishment for the UGV's mission is as follows: The GCS sends specific commands (directives) to the UGV as indicated in an experimentation script, e.g., "Go-to-Point", "Pause" on a specific point, or "Abort" the mission and return home (RTL). The UGV sends sensor measurements streams, e.g., temperature, humidity, video and its geo-spatial position (GPS) to GCS with a predefined frequency. Both GCS and UGV have as a goal the successful completion of the monitoring of the area. Both UGV and GCS monitor the *quality of the network*. The quality of the network can be classified as

[1] http://www.rawfie.eu/about

proposed in [24] and is crucial for the mission because significant commands (down-link from GCS to UGV) or sensor values/measurements (up-link from UGV to GCS) can be occasionally lost due to the stochastic network behavior.

We propose a real-time control mechanism to *adapt* to changes in network quality by dynamically *pausing* control telemetry and control messages based on *optimal sequential decision making rules*. This is expected to ensure the trouble-free delivery of critical information subject to the dynamic network status that UxVs encounter while dispatching a certain mission.

Remark 1. *Overall, our scheme can be applied in all cases where connections are competing for stochastically varying network resource and optimally manage their relative priorities.*

This chapter is organized as follows: In Section 4.2, the proposed optimized information flow model and our two optimal stopping problem solutions. Section 4.3 presents our comprehensive experiments with real UxV settings, where our mechanisms performances are followed by the conclusions in Section 4.4.

4.1.1 Contribution

The studied problem deals with poor network performance during a UxV predefined mission. The online control of UxVs mission is highly connected with two types of paths: geo-spatial and network. The union of localization and network factors concludes to safe mission with accurate data reports. It is apparent that the mobility factor adds up new complexity to the aforementioned solutions in literature that handle message forwarding or routing topologies for stationary sensor networks.

Furthermore, our framework is independent of the UxVs technologies and can be applied to different kind of UxVs (aerial, sea, ground) and to their on-board software like ROS [49] or Ardupilot [12]. Mostly in literature, the UxV solutions are targeted to problems with a specific type of UxVs. However, our work in this thesis does not depend on the type of UxV. Our decision making process handles the control of contextual flow in a mission based on the quality network statistics with no-prior knowledge of the environment and the category of the device, i.e., aerial, ground or surface vehicles. This real-time decision making framework is based on two Optimal Stopping Time Policies that optimally schedule context delivery (control messages and values) and deliver messages with minimum loss of packets in poor or saturated networks.

Our specific technical contribution of this work is:

1. *A stochastic optimization mechanism for on-line network quality change detection*;
2. *A hybrid sequential decision making mechanism for optimal control commands from the GCS to UxV based on the Optimal Stopping Theory*;
3. *Proof of optimality of the two proposed mechanisms in UxV MIoT environments*;

4. *Comprehensive performance evaluation, sensitivity analysis of the major parameters, and comparative assessment of the proposed mechanisms in a real-testbed UxVs platform*.

4.2 System Design of Time-Optimized Decision Making Model for Unmanned vehicles

4.2.1 Rationale

The main contribution of this research work is to establish an in-network/on-device lightweight sequential Decision Making Process (DMP) that leverages the on-line derived network statistics to efficiently control the progress of a UxV mission. Each UxV is equipped with a number of sensors and at least one network interface. Our DMP is capturing network related information, e.g., packet error rate, and controls the transmission of messages on both UxV to GCS and GCS to UxV, dynamically. Fundamentally, based on the real-time captured network statistics, our DMP makes transitions during the UxV mission between two states: *active* and *passive* state as shown in Figure 4.1. The time duration for staying in each state and the transition from one state to another are *optimally* determined by two real-time decision making mechanisms as will be discussed in the following paragraphs.

All messages exchanged between UxV and GCS are categorized in 'high' and 'low' priority. High priority message is considered (i) the minimum necessary systemic instructions to carry out a mission and (ii) sensor data defined by the UxV operator as highly important. When the UxV/GCS is in active state then the DMP sends constantly messages for telemetry and control. In the passive state, the DMP sends only high priority messages. For instance, the position reporting from the UxV is a prerequisite for the safe execution of the mission. In this case, high-priority commands are being sent constantly. Low priority messages, e.g., temperature values captured locally from the UxV sensors, can be delayed until the network exhibits *better performance*. The message priority at the GCS is the inverse, i.e., significant messages are to be delayed in order to safely reach the UxV. The described rules of state transitions based on the network state, the UxV and GCS are shown in Table 4.1.

The DMP runs locally on the UxV and on the GCS, enriched with a Time-Optimized Change-Point Decision Making Process (TOCP). The TOCP is triggered when a change on network performance occurs; the TOCP is discussed extensively in Section 4.2.2. This will enable the UxV and the GCS to transit from the *active* state to the *passive* state. When the DMP concludes on the 'passive' state, then a Discounted Reward Decision Making Process (DRP) is activated, as will be discussed in Section 4.2.3. The rationale is that the DRP *sequentially* ranks the network quality measurements from the relatively *worst* to the relatively *best* and, then *optimally*, it delays its pause interval for the (stochastically) globally best network observation to resume from the pausing period as dictated by the TOCP. The pausing period has a maximum deadline, hereinafter referred to as the pausing horizon Th_{max}. This indicates the maximum time interval the UxV waits *without*

Table 4.1: Rules of State Transitions

Component	Network State 'Good'	Network State 'Bad'
UxV - High priority sensors	ON	ON
GCS - High Priority Messages	ON	OFF
UxV - Low priority sensors	ON	OFF

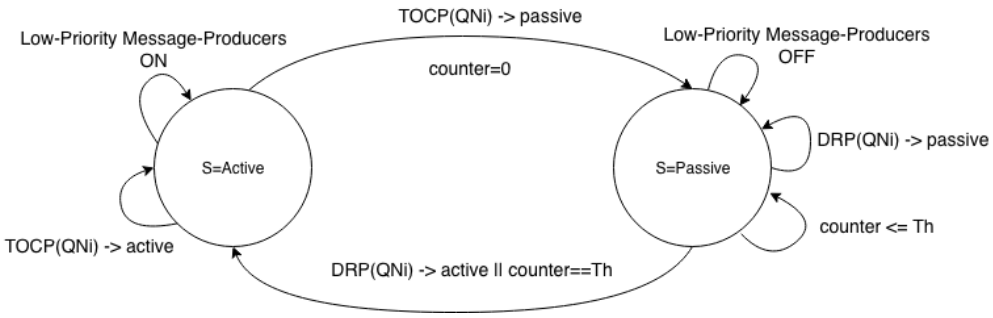


Figure 4.1: UxV State Transition Model

receiving any command and ACK messages from the GCS. To sum up, we propose a mechanism for temporal control of the transmission of the messages to and from the UxV. This mechanism is based on a network condition model that transits from good to bad and vice versa. All these transitions are monitored and validated through our system using the principles of the change detection and optimal stopping theory.

4.2.2 Time-Optimized Change-Point Decision Making Process

In this section we introduce the TOCP, which reflects the behavior of the UxV being in the active state. Specifically, consider the network quality readings $x_1, x_2, \ldots, x_n$ as a discrete random signal with independent and identically distributed (i.i.d.) random variables observed sequentially in real time. Consider also that the network readings follow a probability density function $p(x_n, f_i)$. In our case, f_i expresses the normal distribution with mean value μ_i and variance σ_i. To estimate $p(x_n, f_i)$, a probability density function comparison method has been adopted to derive the *closest* distribution to our Quality Network Indicator (QNI) values.

The QNI derives from the normalization of the basic network metrics: Packet Error Rate (PER), Signal-to-Noise Ration (SNR), and the interference quality indicator (Q). The SNR is defined as the ratio of signal power to the noise power. The PER is calculated as the rate between the lost packets and the total packets sent through the network. The interference quality indicator Q is exported by an access point in the scale $[0, 100]$ and depends on the level of contention or interference, like the bit or frame error rate, or other hardware metric. The holistic QNI at time n indicates the *quality* of the current network connectivity defined

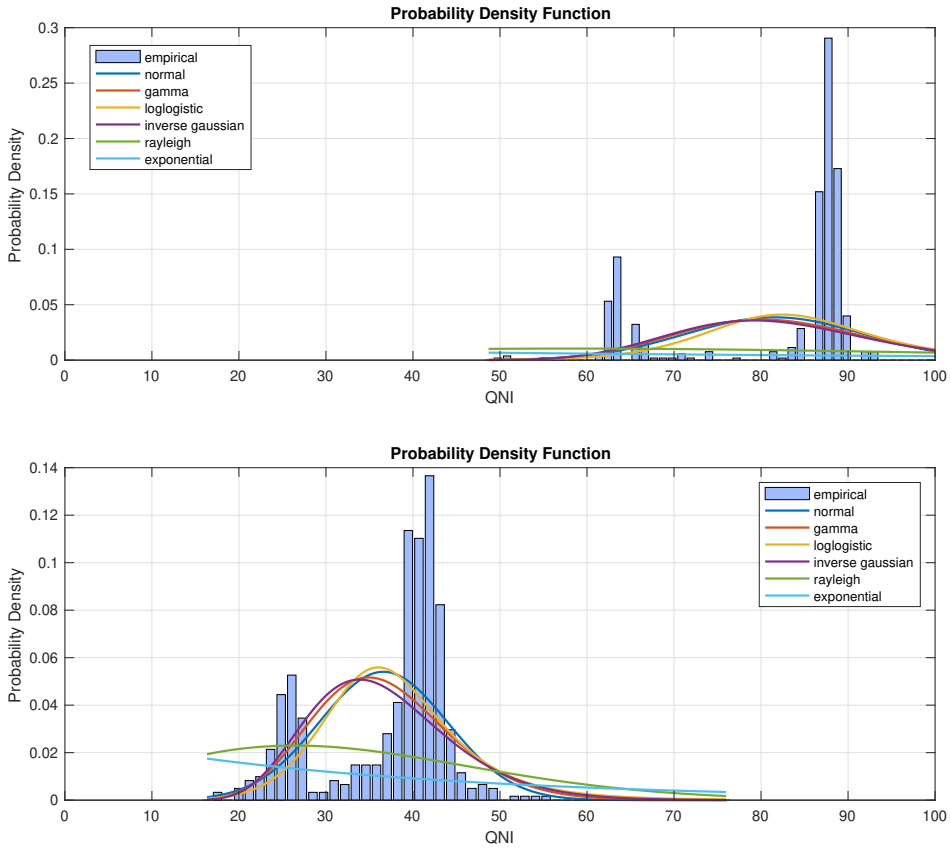


Figure 4.2: (Upper) a. The probability density function f_0 and (lower) b. the f_1 model fitting for *good* and *bad* quality of QNI values, respectively.

as the weighted sum of the (normalized) quality indicators:

$$QNI_n \quad = \quad a_1 P\hat{E}R_n + a_2 S\hat{N}R_n + a_3 \hat{Q}_n, \tag{4.1}$$

where the QNI is affine combination of PER, SNQ and Q in $[0, 100]$ such that $\sum_{i=1}^{3} a_i = 1$, $a_i \in [0, 1], \forall i$.

We consider the incoming QNI values as an adapted strong Markov process $(X_n)_{n \leftarrow 0}$ defined by the filtered probability space $p(x_n, f_0)$. The estimation of the $p(x_n, f_i)$ is based on model fitting of all the parametric probability distributions to the QNI. The output of this model fitting is shown in Figure 4.2a for $p(x_n, f_0)$ and Figure 4.2b for $p(x_n, f_1)$. The list of examined probability distributions is extensive. We are based our decisions and reasoning on the fundamental NLogL (Negative of the Log Likelihood) and the BIC (Bayesian Information Criterion) metrics. For each distribution examined, we derived the corresponding NLogL and BIC values provided in Table 4.2. As it is shown in Figure 4.2a and Figure 4.2b, the best distribution fitting to our experimental data is the Normal Distribution.

We further studied an abrupt change from good to bad network conditions. In this case, we performed experiments in which the network conditions changed at time m. As shown in Figure 4.3, before time m, the QNI follows the distribution $p(x_n, f_0)$, and after time m,

Table 4.2: NLogL and BIC metrics for the probability distributions.

Examined Distribution	NLogL	BIC
Normal ($\mathcal{N}$)	1876.5	3765.5
Gamma (Γ)	1904.2	3820.8
Log-logistic	1909.3	3831
Inverse Gaussian (IG)	1919.4	3851.2
Rayleigh	2366.6	4739.5
Exponential (Exp)	2700.7	54076

the QNI follows $p(x_n, f_1)$. Under these experimental observations, the QNI distribution observed between the first sample x_0 and the current x_k sample takes two forms, where H_0 represents No-Change-Point Hypothesis and H_1 represents the Change-Point Hypothesis:

$$p(x) = \begin{cases} \prod_{n=0}^{k} p(x_n, f_0), \text{ No-Change-Point Hypothesis } H_0; \\ \prod_{n=0}^{m-1} p(x_n, f_0) \prod_{n=m}^{k} p(x_n, f_1), \text{ Change-Point Hypothesis } H_1 \end{cases} \tag{4.2}$$

The challenge is to decide between the two hypotheses H_0, H_1 w.r.t. QNI, and to approximate efficiently and timely the potential change point time m. A feasible solution derived by the change-point detection theory adopts the minmax approach in [41].

Let us define the conditional expected detection delay by

$$\mathbb{E}_{H1}[(N_d - m + 1)^+ | n = 0, 1 \ldots, m-1], \tag{4.3}$$

as defined in [41], where the expectation is taken under one change hypothesis H_1. The minimax performance criterion is given by its supremum taken over. Specifically, the *worst-case detection delay* is estimated as:

$$D_n(\tau) = \sup_{n \geq 1} ess \sup \mathbb{E}_k[(\tau - k + 1)^+ | \mathbb{F}_{k-1}], \tag{4.4}$$

with $x^+ = max\{x, 0\}$. Based on this objective, we formulate the change-point detection Problem 1:

Problem 1. *The UxV should determine an optimal change-point detection time τ that minimizes the worst-case detection delay in (4.4).*

4.2.2.1 Solution for TOCP

Let us first denote the FAR defined as [30]:

$$\text{FAR}(\tau) = \frac{1}{\mathbb{E}_\infty[\tau]}.$$

Based on our examined distribution fitting, we introduce the instantaneous log-likelihood ratio at time n by:

$$L_x(n) \quad = \quad \ln \frac{p(x(n), f_0)}{p(x(n), f_1)} = \ln \frac{\sigma_1^2}{\sigma_0^2} + \frac{(x-\mu_1)^2}{2\sigma_1^2} - \frac{(x-\mu_0)^2}{2\sigma_0^2}, \tag{4.5}$$

and its cumulative summation of the ratios from 0 to n:

$$S(n) = \sum_{k=0}^{n} L_x(k). \tag{4.6}$$

The expectation $\mathbb{E}_\infty[\tau]$ defines the expected time between false alarms. A false alarm in our case is defined when the DMP mechanism detects a change for state transition to passive, while the network quality is characterized as good. Under the Lorden criterion, our objective is to find the stopping rule that minimizes the worst-case delay subject to an upper bound on the FAR. The decision function in our problem in a change between good and severe network conditions is shown in Figure $4.3b$.

The optimal solution to (4.4) was determined in [39], which is provided by the Cumulative Sum (CUSUM) test [40]. A presentation of the CUSUM approach applied to our problem can be found in Appendix B and its description is shown in Algorithm 1. The optimal stopping time for detecting the change point is given by:

$$\tau^* = \min\{n \geq 1, \max_{1 \leq k \leq n} \sum_{i=k}^{n} L_x(i) \geq \alpha\} \tag{4.7}$$

Let the detection threshold α be chosen such that the ARL to false alarm derives $FAR \geq \alpha > 0$. Clearly, this condition is equivalent to limit the rate of false detection by a given maximum value. When $\alpha \to \infty$, the CUSUM algorithm minimizes the worst case detection delay $\mathbb{E}_{H1}[N_d]$. The value of this delay can be approximated by using Kullback-Leibler (KL) divergence. The KL captures the discrimination between the post and pre-change hypotheses and measures the *detectability* of the change, which is proved to be:

$$\mathbb{E}_{H1}[N_d] = \frac{\ln \alpha}{\ln(\frac{\sigma_1}{\sigma_0}) + \frac{\sigma_0^2 + (\mu_0 - \mu_1)^2}{2\sigma_1^2} - \frac{1}{2}}. \tag{4.8}$$

Proof. The KL divergence captures the discrimination between the post and pre-change hypotheses and is a measure of the tractability of the change:

$$\mathbb{E}_{H1}[N_d] = \frac{ln\alpha}{I(p_{f0}, p_{f1})} = \frac{ln\alpha}{E_{f0}[ln(\frac{p(x_n, f0)}{p(x_n, f1)})]}, \tag{4.9}$$

where

$$
\begin{aligned}
I(p_{f0}, p_{f1}) &= E_{f0}(ln(\frac{p(x_n, f0)}{p(x_n, f1)})) = \int [(ln(p(x_n, f0)) - ln(p(x_n, f1))]p(x_n, f0)dx \\
&= \int [\frac{-1}{2}ln(2\pi) - ln(\sigma_0) - \frac{-(x-\mu_0)^2}{2\sigma_0^2} + \frac{1}{2}ln(2\pi) + ln(\sigma_1) + \frac{(x-\mu_1)^2}{2\sigma_1^2}] \\
&\quad \times \frac{1}{\sqrt{2\pi\sigma_0^2}} exp[-\frac{(x-\mu_0)^2}{2\sigma_0^2}]dx \\
&= \int \{ln(\frac{\sigma_1}{\sigma_0}) + \frac{1}{2}[\frac{(x-\mu_1)^2}{\sigma_1^2}] - \frac{1}{2}[\frac{(x-\mu_0)^2}{\sigma_0^2}]\} \times \frac{1}{\sqrt{2\pi\sigma_0^2}} exp[-\frac{(x-\mu_0)^2}{2\sigma_0^2}]dx \\
&= E_0\{ln(\frac{\sigma_1}{\sigma_0}) + \frac{1}{2}[\frac{(x-\mu_1)^2}{\sigma_1^2}] - \frac{1}{2}[\frac{(x-\mu_0)^2}{\sigma_0^2}]\} \\
&= ln(\frac{\sigma_1}{\sigma_0}) + \frac{1}{2\sigma_1^2}E_0\{(X-\mu_1)^2\} - \frac{1}{2\sigma_0^2}E_0\{(X-\mu_0)^2\} \\
&= ln(\frac{\sigma_1}{\sigma_0}) + \frac{1}{2\sigma_1^2}[E_0\{(X-\mu_1)^2\} + 2(\mu_0-\mu_1)E_0(X-\mu_0) + (\mu_0^2-\mu_1^2)] - \frac{1}{2} \\
&= ln(\frac{\sigma_1}{\sigma_0}) + \frac{\sigma_0^2 + (\mu_0-\mu_1)^2}{2\sigma_1^2} - \frac{1}{2}
\end{aligned}
$$

□

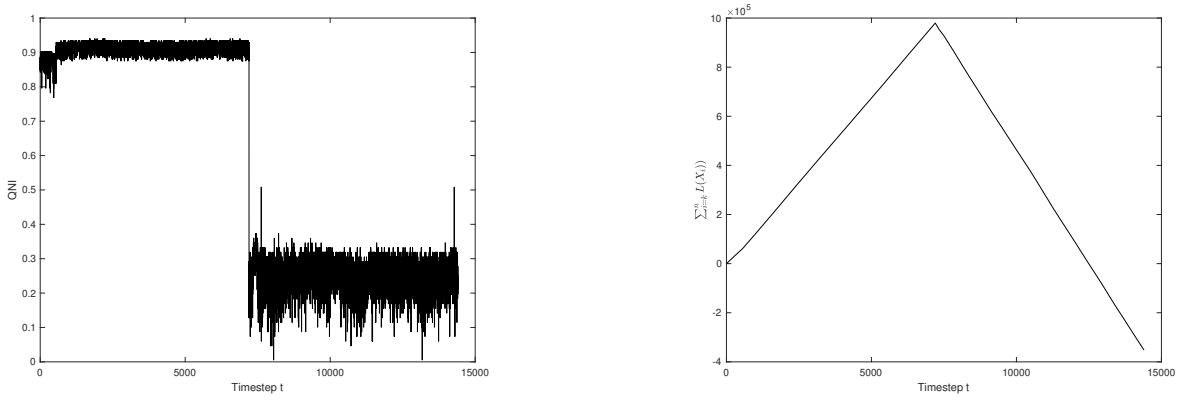


Figure 4.3: The behavior of the QNI and the cumulative log-likelihood ratio corresponding to a change from a ‘good’ network state to a ‘bad’ network state.

4.2.3 Discounted Reward Decision Making Process

We propose a hybrid solution based on the change-point detection and a Discounted Reward Process (DRP) with Linear Discount Function (LDF). The reason is that the UxV cannot pause forever to send commands or to send telemetry messages. The UxV has

a hard limit for sending message to GCS in order to report that it is alive and active. The same stands for GCS, i.e., the GCS cannot leave a UxV with no control messages. Therefore, the pausing period when a UxV decides whether to start again the streaming of commands can be treated as a finite horizon problem as will be described here.

It is assumed that when the pausing period starts, the UxV receives a QNI value x_k at a time instance k. The objective is to seek a stopping rule that will *maximize the probability of choosing the best (maximum) QNI value* x_k indicating the best possible network condition.

Let us define a random variable u_k, which represents the LDF reward if the kth QNI observation is chosen, that is:

$$u_k = \begin{cases} 1 - \frac{\gamma}{N}k & \text{if } x_k = \max\{x_l, l = 1, \ldots, k-1\} \\ 0 & \text{otherwise} \end{cases} \tag{4.10}$$

The parameter $\gamma \in [0, 1]$ denotes the *discount factor*. The discount factor γ represents the modeling abstraction where UxV focuses on selecting the best QNI value of the N received QNI values. The LDF in (4.10) indicates that the UxV has to report at least one QNI value observing at most N QNI values. The higher the discount factor γ is, the higher the penalty gets until a reception of a better QNI value. The UxV receives the reward u_k if the kth observation is chosen and refers to the highest QNI value among all N QNI values; otherwise u_k is zero.

Problem 2. *Given a fixed time horizon N, the UxV has to determine a optimal stopping rule $r, 1 \leq r \leq N$, which maximizes the expectation $\mathbb{E}[u_r]$.*

For solving Problem 2, consider first receiving the kth observation of the QNI value x_k. We can then define the random variable $z_k = j$ $(1 \leq j \leq k)$, which denotes the *relative ranking* of the QNI value x_k among the first k observations of the UxV. The assignment $z_k = 1$ means that the kth QNI value refers to the highest QNI value among the first k QNI values seen. We state then the optimal policy for a UxV w.r.t. LDF in (4.10) as follows in our optimal policy.

Remark 2. *Optimal Policy: There exists a time $r^*(1 \leq r^* \leq N)$ such that the UxV observes the QNI values of the first $r^* - 1$ QNI values without accepting any of them. Then for $r^* \leq k \leq N$ the UxV accepts x_k if $z_k = 1$. In case of $z_k > 1, \forall r^* \leq k < N$, or $r^* = N$, then the UxV accepts x_N, which is the last observed QNI value, with $u_N = 1 - \gamma$.*

Let $\omega_k(j)$ be the conditional expected reward of the kth observation given that $z_k = j$, that is, $\omega_k(j) = \mathbb{E}[u_k | z_k = j]$. The probability of finding the maximum QNI value x_k, i.e., $(j = 1)$, at the kth observation is:

$$P(u_k = 1 | z_k = j) = \frac{P(u_k = 1, z_k = j)}{P(z_k = j)} = \begin{cases} \frac{k}{N} & \text{if } j = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Hence, we have for the $\omega_k(j)$ that:

$$\omega_k(j) = \begin{cases} \frac{k}{N}(1 - \frac{\gamma}{N}k) & \text{if } j = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{4.11}$$

The value $\omega_k(j) = 0$ for $j \neq 1$ indicates that there is no reward if the best quality network state is not chosen.

For each $r = 1, \ldots, N$ let $\xi(r)$ denote a stopping rule, that is the first $r-1$ QNI values are observed and the next QNI value, which exceeds all of its predecessors, is accepted. If none of the first $N-1$ QNI values is reported then the last one is reported. Then, we obtain that:

$$P(\xi(r) = k) \quad = \quad \frac{r-1}{k(k-1)}, \tag{4.12}$$

thus, the corresponding expected payoff $\phi(r; \gamma, N)$ w.r.t. to the reward function in (4.10) is

$$\phi(r; \gamma, N) = \mathbb{E}[u_{\xi(r)}] = \sum_{k=r}^{N} \omega_k(1) P(\xi(r) = k) = \frac{r-1}{N} \sum_{k=r}^{N} \left(\frac{1 - \frac{\gamma}{N} k}{k-1} \right) \tag{4.13}$$

It follows that the r^* of the proposed optimal policy that maximizes the expected payoff $\phi(r; \gamma, N)$ in (4.13) is the optimal stopping rule. The $\phi(r^*; \gamma, N)$ is the maximum probability of finding the best QNI value on the UxV.

Theorem 1. *There exists a* r^* $(1 \leq r^* \leq N)$ *which maximizes* $\phi(r; \gamma, N)$ *over* $1, 2, \ldots, N$. *Then, the optimal stopping rule* r^* *satisfies the following:*

$$r^* = r^*(\gamma, N) = \min \left\{ r \geq 1 | \lambda(r; \gamma, N) = \sum_{k=r}^{N-1} \frac{1}{k} + r \frac{2\frac{\gamma}{N}}{1 - \frac{\gamma}{N}} - \frac{1+\gamma}{1-\frac{\gamma}{N}} \leq 0 \right\} \tag{4.14}$$

Proof. The expected payoff of the stopping rule r is $\phi(; \gamma, N)$. Hence, we find the first optimal stopping rule r^* for which it holds true that $\phi(r; \gamma, N) - \phi(r+1; \gamma, N) \geq 0$, to stop at r given the conditional expectation of the reward at $r+1$ after observing the relative rankings up to r. Specifically, since the conditional expectation at $r+1$ is

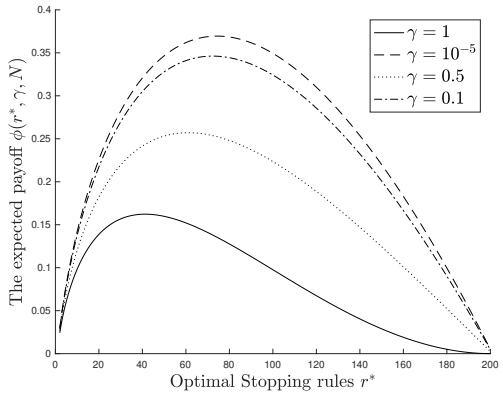
$$\phi(r+1; \gamma, N) = \frac{r}{N} \sum_{k=r+1}^{N} \frac{1 - \frac{\gamma}{N} k}{k-1} \tag{4.15}$$

we can derive that:

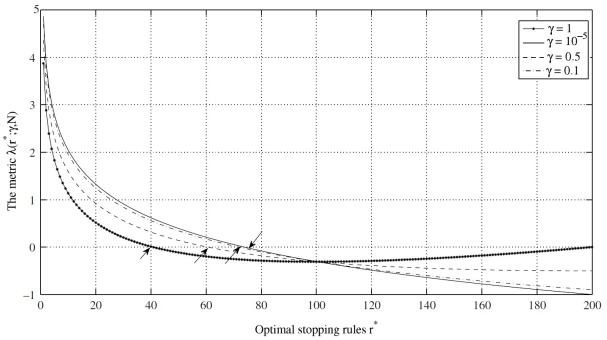
$$\phi(r; \gamma, N) \quad = \quad \frac{r-1}{N} \sum_{k=r}^{N} \frac{1 - \frac{\gamma}{N} k}{k-1} = \frac{1 - \frac{\gamma}{N} r}{N} + \phi(r+1; \gamma, N) - \frac{1}{N} \sum_{k=r+1}^{N} \frac{1 - \frac{\gamma}{N} k}{k-1}.$$

Hence, in order to stop at the first r, which satisfies that:

$$\phi(r; \gamma, N) - \phi(r+1; \gamma, N) \geq 0, \tag{4.16}$$



(a) The expected payoff $\phi(r^*, \gamma, N)$ for different values of discount factor γ and N=200.



(b) The optimal stopping rules for different values of discount factor γ and $N = 200$. The arrows depict the earliest (optimal) stopping times r^* such that $\lambda(r^*; \gamma, N) \leq 0$.

Figure 4.4: Analysis of LDF based on different values of discount factor γ.

we obtain that:

$$(1 - \frac{\gamma}{N}r) + \frac{\gamma}{N}(N - r) - (1 - \frac{\gamma}{N}) \sum_{k=r}^{N-1} \frac{1}{k} \geq 0 \tag{4.17}$$

which concludes

$$\sum_{k=r}^{N-1} \frac{1}{k} + r\frac{2\frac{\gamma}{N}}{1 - \frac{\gamma}{N}} - \frac{1 + \gamma}{1 - \frac{\gamma}{N}} \leq 0. \tag{4.18}$$

Hence, the optimal stopping time r^* is obtained at the first $r \geq 1$, where the above equation turns non-positive. □

The implementation of the optimal stopping time r^* is shown in Algorithm 2. For $\gamma = 0$ and a large N, we obtain the classical optimal stopping rule $r^* = \frac{N}{e}$. Figure 4.4b depicts the value $\lambda(r; \gamma, N)$ and the optimal stopping rules r^* for which $\lambda(r^*; \gamma, N) \leq 0$ for different values of γ and $N = 200$. As $\gamma \to 0$ then $r^* \to \frac{N}{e}$ as illustrated in Figure 4.4b (for $\gamma = 10^{-5}, r^* = 74 \approx N/e$). The UxV reports y at observation $k \geq r^*$ for which $x_k >$ max$\{x_l : l = 1, \ldots, r^*\}$.

In Figure 4.4a we illustrate the value of the maximum probability of choosing the best QNI value $\phi(r^*; \gamma, N)$. For $\gamma = 0$ we obtain the classical secretary problem, i.e., $\phi(r^*; 0, N) \approx 1/e = 0.3678$ for large N. As the discount factor increases the maximum expected payoff decreases for large N. This indicates that we obtain a *low* likelihood (close to 0.161 for $N = 200$) in accepting the *best* QNI value once $\gamma = 1$, and this is the highest probability of achieving this. Moreover, in Figure 4.4b we show the optimal stopping rules for different values of discount factor γ and $N = 200$. The arrows depict the earliest (optimal) stopping times r^* such that $\lambda(r^*; \gamma, N) \leq 0$.

Remark 3. *For $0 < \gamma_1 < \gamma_2 \leq 1$ the corresponding optimal stopping rule $r_1^* > r_2^*$. This indicates that the UxV finds a QNI value earlier (stops the process earlier) when the dis-*

count factor is higher. Furthermore, as the discount factor is low then the UxV accepts a QNI value later in N; note also that the initial value of $r^* \to \frac{N}{e}$ as $\gamma \to 0$ for all N.

Algorithm 1 TOCP-DRP Algorithm

$n \leftarrow 0$
$T_h \leftarrow$ maximum threshold interval
$r \leftarrow$ number of observations
$\alpha \leftarrow$ Change point detection threshold
$active \leftarrow TRUE$
$counter \leftarrow 0$
$(x^*, r^*) = LDS_{OST}(r, \gamma)$
while the algorithm is not stopped **do**
    if $active$ **then** /* CUSUM Algorithm described in Appendix C */
        measure the current QNI x_n
        $s_n = \ln \frac{p(x(n), f_0)}{p(x(n), f_1)}$
        $S_n = \sum_{k=0}^{n} s_k$
        $G_n = S_n - \min_{1 \leq k \leq n} \{S_{k-1}\}$
        if $G_n > \alpha$ **then** /*A change point is detected; DRP is activated*/
            $N_d \leftarrow n$
            $\hat{n} \leftarrow \arg\min_{1 \leq k \leq n} S_{k-1}$
            Change occurs
            $active \leftarrow FALSE$
            Reset
        $n = n + 1$
    else
        if $n == T_h$ **then** /* maximum pausing time is reached T_h */
            $active \leftarrow TRUE$;
            break
        else$[x^*, stopped, m] = LDSF(n, r^*, x_n, x^*)$ /* invocation of DRP*/
            if $stopped == TRUE$ **then**
                $active \leftarrow TRUE$;
                break
        $n = n + 1$

4.3 Performance Evaluation

We evaluate a complete functional ground UxV that operates on two different missions, i.e., scanning search for a specific value and exhaustive scan of a certain location. We focus on the latency and the quality of the network during the mission and the impact of various parameters like mobility. We begin with a brief description of our experiment methodology.

Algorithm 2 DRP Procedures

function $LDS_{OST}(r, \gamma)$
    for $1 < n < r$ **do**
        $y(n) = LDS(n, \gamma, r)$
    return y

function $LDSF(k, r^*, x, x^*)$
    $stopped \leftarrow FALSE$
    $position \leftarrow -1$
    if $k < r^*$ **then**
        if $x > x^*$ **then**
            $x^* = x$
    else
        if $x > x^*$ **then**
            $x^* = x$
            $stopped \leftarrow TRUE$
            $position = k$
    return $x^*, stopped, position$

function LDS(x, r, γ)
    $s \leftarrow 0$
    for $x < i < \gamma$ **do**
        $s = s + \frac{1}{x}$
    $y = y + r\frac{2\frac{1}{N}}{1-\frac{\gamma}{N}} - \frac{1+\gamma}{1-\frac{\gamma}{N}}$
    return y

Table 4.3: Model Parameters for the Experiments.

Parameter Names	Values
Change point detection threshold α	[0,1]
DRP discount factorγ	[1 10]
Maximum pause horizon T_h	60

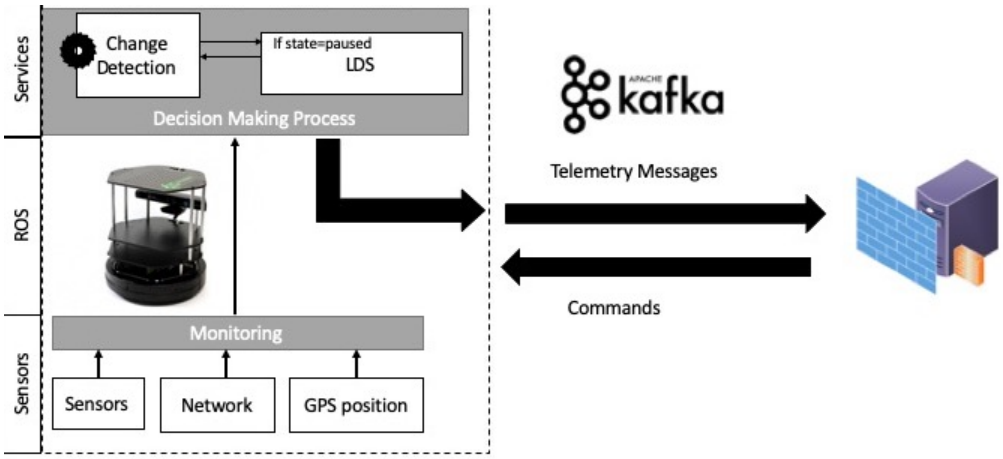


Figure 4.5: The TOCP-DRP proposed architecture for the UxV Management.

4.3.1 Experimental Platform & Methodology

The open-source TurtleBot device was used as ground UxV, i.e. UGV, in our experiments as shown in Figure 4.6. The TurtleBot uses a camera with depth sensor, i.e., XBOX Kinect for mapping purposes. ROS (Robotic Operating System) is the main operating system, which is an open-source, meta-operating system executing on a Raspberry Pi, as shown in Figure 4.6. UGV receives movement commands from the GCS in order to approach the given trajectory's way-points and finally reaches the objective waypoint. The UGV creates a map of the environment and, simultaneously, localizes itself in it, which is commonly known as the SLAM (Simultaneous Localization and Mapping) technology. This is also required to safely navigate within open spaces and proceed with informed decisions about the exploration targets. The Rviz [48] software was used to illustrate the mapping instance created by the UGV in Figure 4.10.

The communication spine between GCS and UxV is a message bus platform based on Apache Kafka, as shown in 4.5. The ROS publish-subscribe message pattern facilitates the interoperability with Apache Kafka, which is basically a messaging system where clients publish messages and from where consumers 'consume' them. The main advantages of the Apache Kafka are i) the high performance in delivering messages and ii) the ability to scale out by distributing the workload among different servers, therefore, supporting a cluster-based architecture. As such, it can be used for transmitting UGV measurements that will be routed from producers i.e., UxVs, to the consumers i.e., the GCS for monitoring, control, etc.

4.3.2 Model Parameters & Real Datasets

Prior to the real experimentation of our DMP and TOCP mechanisms, we consider a large-scale experiment generated randomly as a combination of real-life datasets. The real-life datasets were generated after multiple runs of different network conditions. We can

(a) (b)

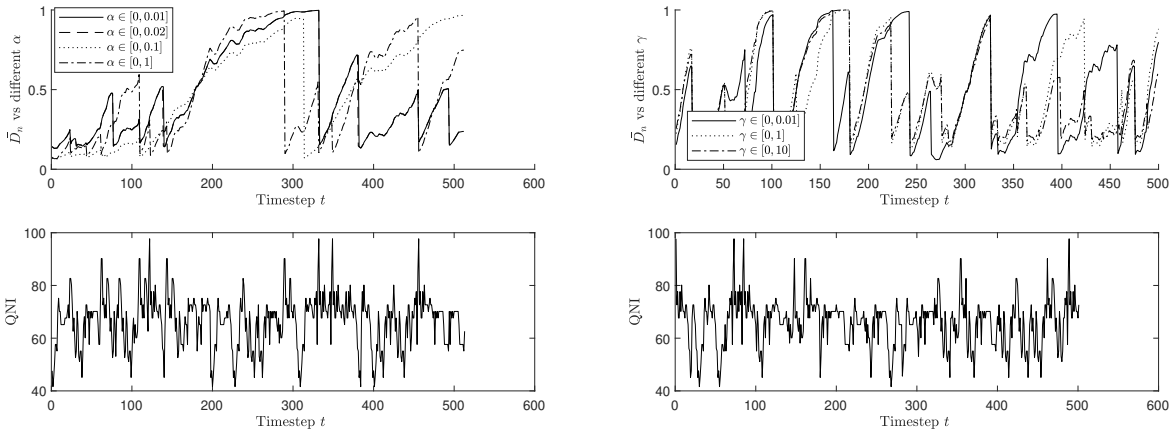
Figure 4.6: The Turtlebot UGV with XBOX Kinect and Raspberry Pi computing modules.

categorize the scenarios as follows:

1. Good dataset, experiencing no disconnections, i.e., QNI values range in $(60, 100]$;
2. Medium dataset, indicating a saturated network where the QNI values range in $[40, 70]$;
3. Bad dataset with several disconnections experienced, i.e., the QNI values range in $[20, 50]$.

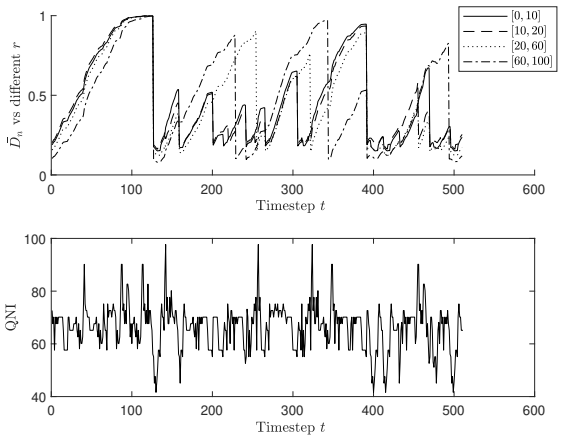
The randomly selected blocks of all the three datasets are producing a dynamic QNI for each run of the experiment. Based on the produced dynamic QNI, we run a number of experiments in order to study the three design parameters of TOCP and DRP optimal model, i.e. α, γ, and r number of observations. We consider equal weights in equation 4.1 for all network parameters, i.e. $a_i = \frac{1}{3}$. We run 100 experiments with specific threshold Th_{max} and the maximum number of C_{max}. Figure 4.7a shows the detection delay function D_n against different α values. The D_n is more adaptive to QNI changes while α values are decreasing. The detected changes in the interval [0,0.02] are 50% more than that of $\alpha \geq 0.1$. For the DRP model, γ is a discount factor, i.e. D_n stops earlier with higher γ values as shown in Figure 4.7b. DRP adopts the LDF function in the 'passive' state. Therefore we can observe frequent changes from passive to the active state as expected.

We further investigate the behavior of the detection delay function D_n as r approaches infinity. As shown in Figure 4.7c, for small values of r, the D_n is sensitive even to small changes in network. While r is working in higher intervals, D_n is more reluctant to DRP



(a)

(b)



(c)

Figure 4.7: The detection delay function D_n vs. (a) different α values; (b) different γ values; and (c) different observations r).

Table 4.4: Packets Produced and Consumed

$ModelName$	$PacketsProduced$	Packets Consumed	Lost Packets
$Nopolicy$	3716	2253	1463
$HeuristicPolicy$	3719	2074	1645
TOCP $\alpha = 0.002$	2550	2478	72
TOCP $\alpha = 0.001$	1914	1707	207

phase. However, the probability of waiting for a large number of observations r to report a change in network quality tends to zero as shown in Figure 4.7c.

In addition we would lilke to study the performance of TOPC applied on a UAV simulator. Taking α values in the interval $[0, 0.02]$, D_n function was studied similarly for $N = 100$ experiments. Four different models were evaluated as follows

- A non policy model, i.e GCS and UxV continuously send and receive messages having indifference to network conditions
- A heuristic theshold-based model, in which the transmission of messages is paused when QNI falls under the interval of $[0.2, 0.3]$ and messages are stored in a queue.
- TOCP model with α values $\in [0, 0.02]$
- TOCP model with α values $\in [0, 0.01]$

The performance metrics of the proposed model with dynamic adoption of transmission of exchanged messages taking into account the real-time networks conditions are i) the number of produced messages by the producers exist in GCS and UxV, ii) the consumed messages by the consumers in both units and iii) the lost number of messages. In this way we are trying to map the differences between the different approaches related to saturated networks with the upper goal the successful execution of a mission. It is evident that in table 4.4 the OST model outperforms related with the rest models. The heuristic model and the original version of receiving and transmitting messages lost messages close to 24%, in other words close to 1 to 4 messages were lost, while only 4% of the messages were lost in OST model with $\alpha \in [0, 0.001]$ and this percentage is minimized to less of 2% in OST model with $\alpha \in [0, 0.002]$.

In figures 4.8a and 4.8b an overview of the messages produced and consumed by the four different models. It is expected that ”network-agnostic” models produce a great amount of exchanged models and especially in threshold model the messages are displaced due to the latency of queue. In case of OST models, producers and consumers have a more moderate and normal distribution of messages in time hence the messages are delivered even if the existence of network disconnections.

The latency issues is more evident in figure 4.9. The difference in each timestep of the messages produced and consumed by the two units is studied in the main cases, i.e. in

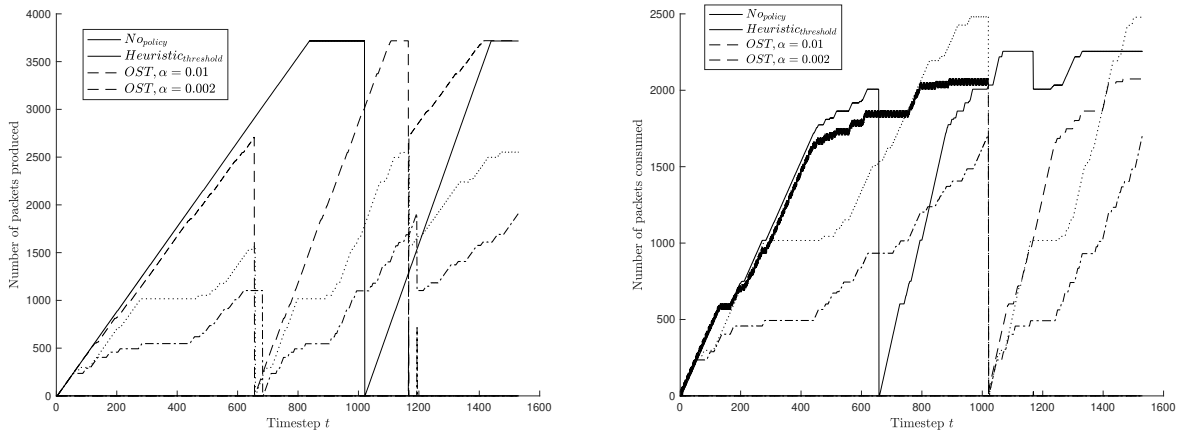


Figure 4.8: The behavior of (up) produced and (down) consumed packets in a saturated network

the original case where no algorithm is implemented and in the case of the OST model. It should be noted that we have 3 main network disconnections in the timestep $t = 600$, $t = 1000$ and $t = 1200$. Until $t = 600$ the OST model, although it works in saturated network, the differences between the produced and consumed messages are zero. After the first disconnection it can be noticed a small shift of the messages paused and consumed in a later time. This latency imported by the stopping time of the producers is balanced in the next time steps and finally close to all messages are delivered successfully. This is not the case for the original model in which the producers continually and burdens the network and consumers with messages that are not delivered.

4.3.3 Experiments: Performance & Comparative Assessment

We report on the experimental evaluation of our framework and mechanisms to examine their performance. We also provide a comparative assessment with models found in the literature. The UxV and GCS are part of the Road-, Air- and Water based Future Internet Experimentation (RAWFIE) platform, which offers an experimentation framework for interconnecting numerous testbeds over which remote experimentation can be realized.

The RAWFIE platform has been developed in the context of H2020 EU-funded (FIRE+ initiative) project, which focuses on the MIoT paradigm and provides research and experimentation facilities through the ever growing domain of UxVs. The RAWFIE platform is device agnostic, promoting the experimentation under different technologies of UxVs that are equipped with different sensors, cameras and network interfaces. Any UxV is managed by a central controlling entity which is programmed per case and fully overview/drive the operation of the respective mechanisms (e.g., auto-pilots, remote controlled ground vehicles), as shown in Figure 4.5. The basic requirement is that each UxV shall be able to receive/send and decode/encode the incoming/outgoing messages from the testbed and

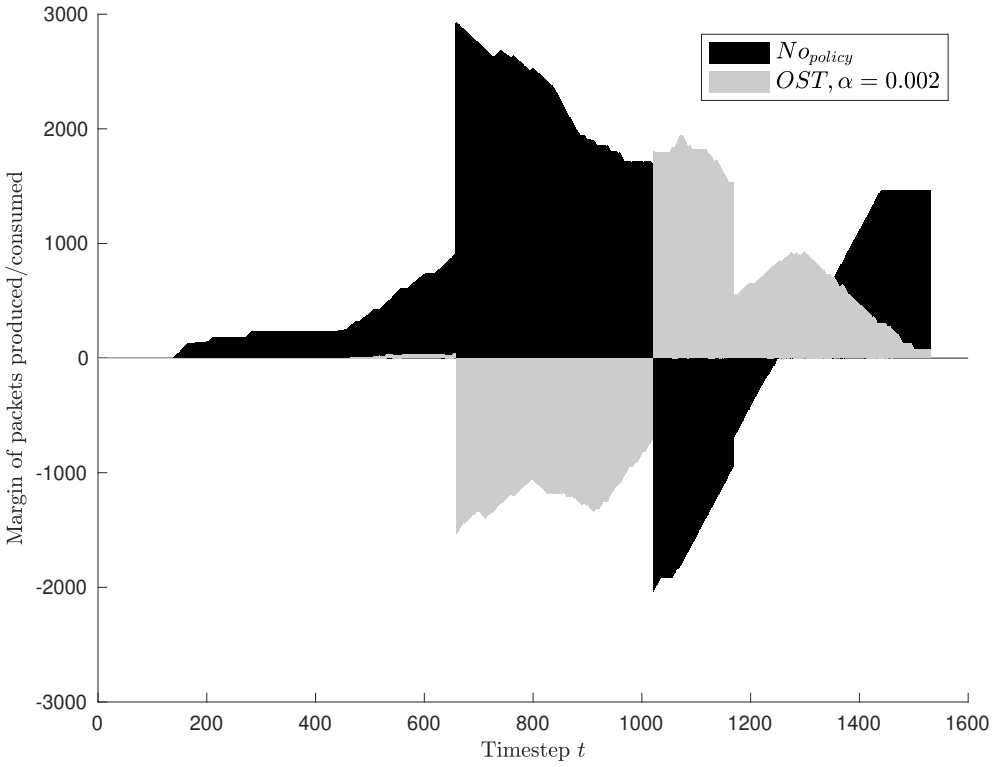


Figure 4.9: The differences between produced and consumed data with No policy applied (black area) and with OST model, $\alpha \in [0, 0.002]$ (grey area).

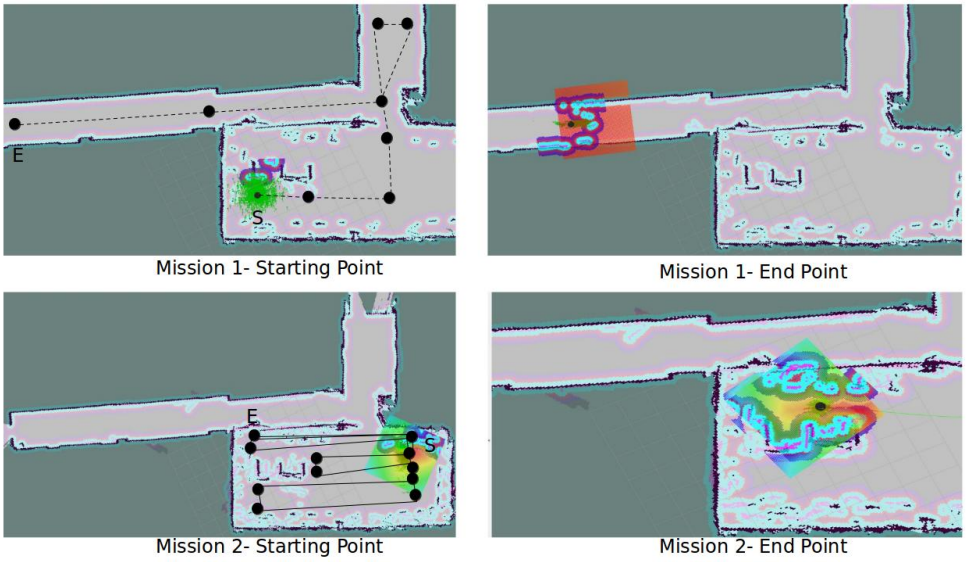


Figure 4.10: Real time monitoring of the robot while executing ”Mission 1-Path exploration” and ”Mission 2-Scanning of an area”

deliver them to the relevant on-board component.

Our TOCP-DRP optimal mechanisms extended the functionalities of RAWFIE and can be applied to any MIoT device, i.e. UAV, UGV and USV. The used UGV in our experiments offers the convenience to make multiple repetitions of the same experiment in the campus of the University of Athens, Greece, unaffected from weather conditions and with real users.

The UGV was used in two real case applications: (1) scanning search for a specific sensor value or a detection of an event designed by a user (mission 1-M1) and (2) exhaustive scan of a room (mission 2-M2). In both missions, the user creates a path as shown in Figure 4.10 and the UGV should follow the way-points in order to reach the final destination. The depicted area is an amphitheater of the Department of Informatics & Telecommunications of the University of Athens and a corridor outside. During the execution of the experiments, the area is used from students and staff members that are moving around and their mobile devices are connected to the same WiFi network.

We performed 100 runs of 10 mins duration each, where each run involves sampling for more than $N = 100$ observations for every sensor integrated on UGV. The comparative assessment is based on four different policies of decision making: (i) the no-policy model, (ii) the heuristic threshold-based model, in which the transmission of messages is paused when QNI falls below a threshold, (iii) TOCP model based on [44], which applies a change detection policy triggering the ‘pause’ mode operation (the passive mode lasts for Th and then it is activated again) and (iv) the hybrid TOCP-DRP model applied on both UGV and GCS. The performance metrics are QNI measured, Packet Error Rate (PER), based on packets sent and packets lost, and the end-to-end message latency.

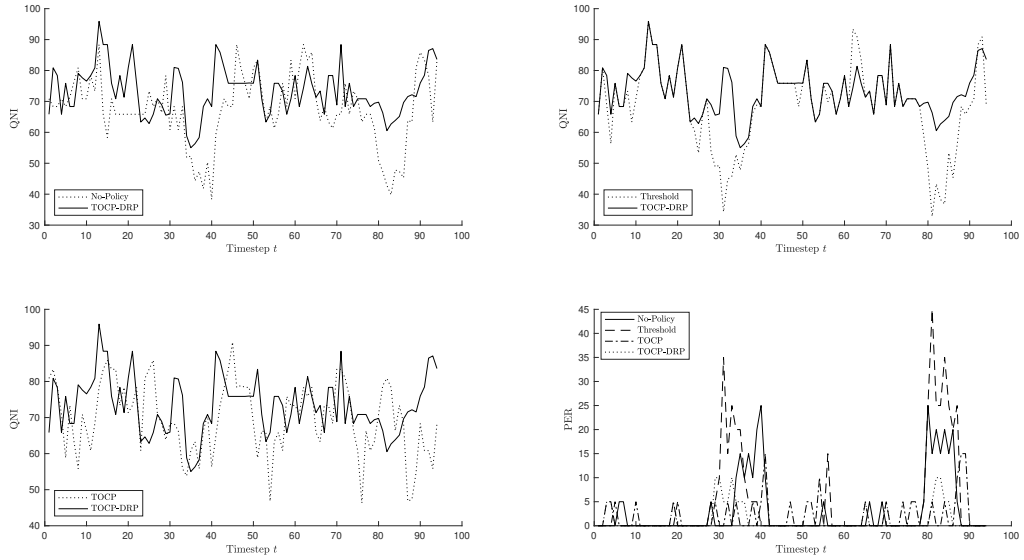


Figure 4.11: The QNI for all the compared policies regarding the mission M1: Exploration of a Path.

4.3.3.1 Expected Performance in Mission M1

Figure 4.11 plots the QNI performance of the four policies. We can observe that in mission M1, two areas of poor connectivity exist in time-steps [35-45] and [75-90]. The no-policy, the threshold-based policy and TOCP policy reach QNI values less than 50%, while our TOCP-DRP policy has a mean value close to 68%. In addition, for $N > 60$ the TOCP-DRP is more intolerant to network changes with mean values around [70-85].

The PER maximum values are for all the policies: $\{no-policy, threshold-based$ policy, $TOCP$, and $TOCP-DRP\}$ are $\{25, 45, 15, 10\}$, respectively, with TOCP-DRP achieving the minimum PER, i.e., we obtain up to 20% less PER compared with the other policies. The TOCP-DRP has better performance than the TOCP policy because TOCP overviews network data only in active mode and TOCP-DRP monitors QNI in both active and passive mode. The deactivation of passive mode in TOCP happens when the threshold is reached and this means that the algorithm is triggered in random time-steps independently of the network status. This is the reason for observing relatively small PER values every 50 steps when the algorithm recognizes a change detection.

4.3.3.2 Expected Performance in Mission M2

Figure 4.12 shows the QNI performance of the four comparison policies for scanning missions. The M2 mission is performed indoors where areas of low connectivity and objects exist as obstacles to the UGV. The QNI has greater fluctuation in this mission relative to

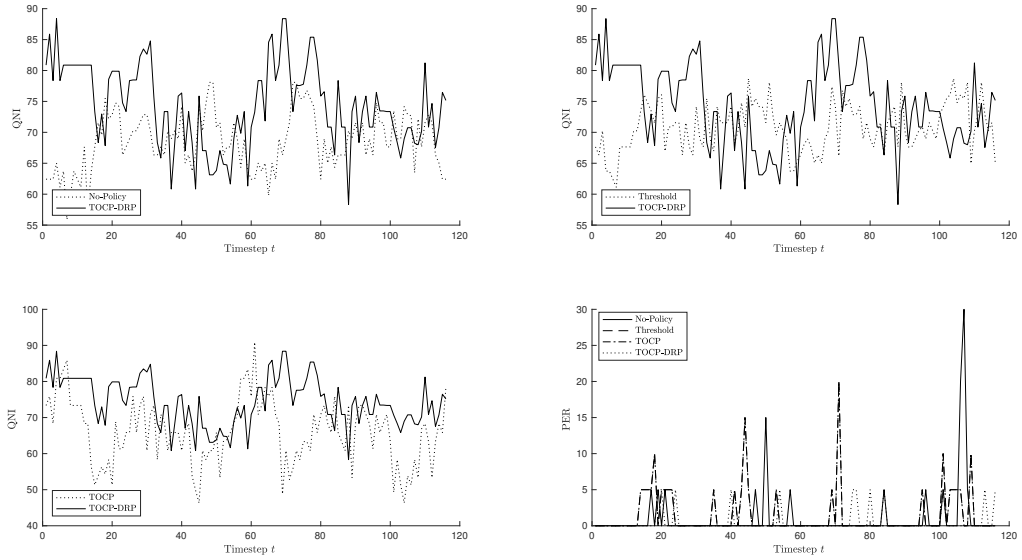


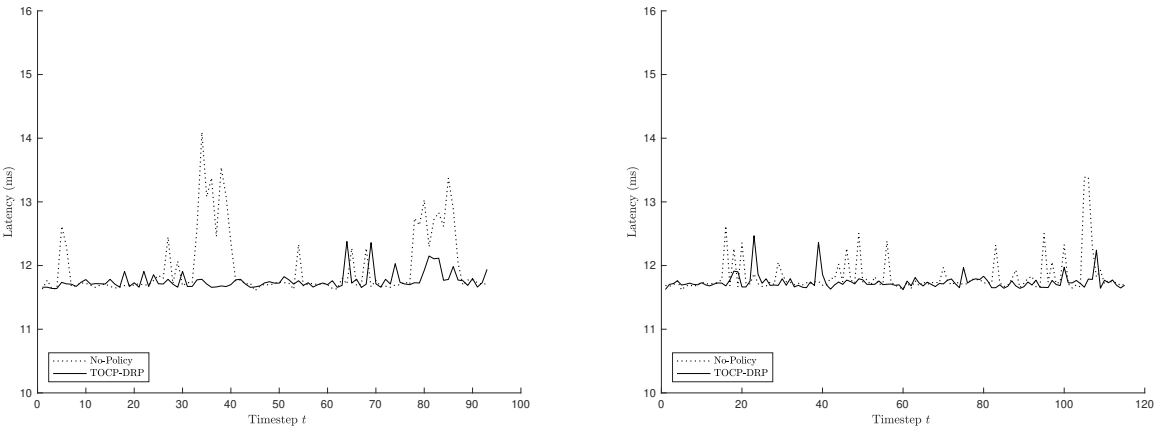
Figure 4.12: The QNI for all the compared policies regarding the mission M2: Scanning of an unknown Area.

the M1 mission. Our TOCP-DRP mechanisms from the early beginning of mission M2, where UGV is positioned in one random corner of an amphitheater, outperforms the other policies. The average values of QNI for all policies: $\{no-policy, threshold-based$ policy, $TOCP,$ and $TOCP-DRP\}$ are $\{68.4446, 70.8197, 65.8525, 76.3498\}$, respectively.

The performance of the PER is similar to the M1 mission. The PER is minimized in our TOCP-DRP policy, where the maximum value is 10% in observations. In the remaining policies, the PER achieve values between 20% and 30% .

4.3.3.3 Expected Latency in Missions M1 & M2

We plot the latency of the no-policy and our TOCP-DRP policy in Figure 4.13a(a) and Figure 4.13a(b) for the missions M1 and M2, respectively. The TOCP-DRP policy is considered more efficient than the no-policy for all the observations in both missions. In particular, in M1 we can measure 24% less end-to-end message latency compared to the original no-policy decision making of UGV. Moreover, the TOCP-DRP policy achieves systematically a message latency value which is close to 9% less of the original message latency. We can conclude that the double hybrid optimal stopping model in the two phases of the network, i.e., active and passive, based on the network assessment monitoring results to missions with low end-to-end latency and low expected PER.



(a) (b)

Figure 4.13: The latency (ms) measured during the no-policy and the TOCP-DRP policy in mission M1 (a) and mission M2 (b).

4.4 Conclusions

We propose an in-network/on-device time-optimized decision making model of real-time control adaptive to changes of the network quality. This adaptive model dynamically pauses telemetry and control messages based on derived optimal stopping rules in order to assess in real-time the trade-off between the delivery of the messages and the network quality statistics. Our DMP policy optimally schedules critical information delivery to a back-end system. This policy uses two optimal stopping theory mechanisms based on change-detection theory and the linear discounted secretary problem. When the quality of the network significantly changes, the UxV and the GCS can decide in real-time to pause/start the transmission of telemetry in order not to overload a saturated network, or to risk to lose completely the messages. Our experimental performance evaluation and comparison assessment showed the successful delivery of messages in poor network conditions and the moderate production of messages so as not to burden an already saturated network.

5. TIME-OPTIMIZED PRIORITIZATION OF KAFKA MESSAGE SCHEDULING FOR UNMANNED VEHICLES IN IOT NETWORKS

5.1 Motivation

Internet of Things is one of the most promising paradigms in the current decade characterized by the use of smart and self-configured objects like sensors, actuators, wearables etc, that are connected to a network and exchange data by sensing, reacting to events, and interacting with the environment. In addition unmanned mobile devices like drones were introduced to users'daily life the last decade and become a part of the whole of "objects" participating in the IoT as long as they carry sensing equipment and on-board computing elements. The huge amount of data generated by sensor-instrumented objects of the real world in an IoT environment impose a great demand on processing and storage to transform the data into useful information or services. Some applications can be latency sensitive, while other applications can require complex processing including historical data and time series analysis.

Therefore, considering the typical resource constraints of IoT devices, it is difficult to envision a real world IoT ecosystem without including a cloud platform or at least a distributed data streaming platforms. Distributed data streaming solutions manage big data flows of relevant data to/from devices, services and micro-services and are critical centerpiece of IoT deployments. These platforms are necessary in IoT infrastructures to process such enormous volumes of data against resource constrained IoT devices. Especially in case of mobile IoT where devices exchange except their telemetry, multimedia content captured in real time and their geolocation in space distributed data streaming platforms are critical parts for a succesful execution of a mission. The key challenges arise when supporting reliable and timely communication over constrained networks (e.g. due to lossy channels and failed components).

To overcome these challenges, we propose a stochastic optimization framework of online control unit applied in the Publish/Subscribe of middleware data exchange platform, in our case Apache Kafka, adaptive to changes in perfomance of message bus. The proposed middleware solution brings together heterogeneous endpoints but also covers a vast spectrum of devices, sensors and control subsystems. We enhance our messaging distribution platform by applying prioritization policy of different types of messages when Key Performance Indicators (KPIs) change. The optimality of the proposed mechanism is achieved by making optimal delivery decision in different priority queues. Optimal delivery decisions involves whether a consumer/producer in the device edge shall pause the pull/push requests in order not to overload a saturated message bus, to cause synchronization issues or to risk to loose completely the messages. The message delivery strategy is coming from two optimal policies [42] based on optimal stopping time series search algorithms.

This chapter is organized as follows: In Section 5.2, the proposed optimized distributed

communication layer is presented. Section 5.3 presents our comprehensive experiments with a drone simulator, where our mechanisms performances are followed by the conclusions in Section 5.4.

5.1.1 Contribution

Let us assume that a swarm of edge devices, i.e unmanned aerial vehicles, can be virtualized and be used in a cloud-based infrastructure. A mission of a swarm of unmanned aerial vehicles will take place in an outdoor geofenced area named as testbed gathering measurements from predefined waypoints. A Ground Control Station (GCS) will control and manage the trajectories and the telemetry produced by UAVs. A message bus, i.e. Apache Kafka, will be deployed at GCS and UAVs; message bus clients will be deployed in the devices. Messages will be exchanged between clients and the central infrastructure (GCS). All streaming requests will be pushed inside different queues of the Apache Kafka message bus. Data preprocessing can be performed inside the queues of the Apache Kafka utilizing the KSQL Rest interface. A local controller in the devices will consume messages upon validating their importance about Key Performance Indicators (KPIs) and will select to forward the streaming process in a different priority queue (high and low). Hence a distributed message bus architecture adaptive to the changes in the performance of the messages exchanged is the main core of the architecture in order to ensure the real time control and safety of UAVs. Our contribution is as follows

- *presenting a distributed platform on the UAVs for the exchange of messages*,
- *definition of optimal stopping rule for the discussed problem and how it is applied in our context*,
- *comprehensive performance evaluation of our selected architecture schema*.

5.2 System Design

In this section the establishment of a lightweight sequential Decision Making Process (DMP) applied on Apache Kafka priorities scheme is presented. In our model we assume a swarm of UAVS operating in unknown network conditions. The swarm of UAVs is controled by a ground control unit (GCU) in real-time. A UAV can be considered as an edge device equipped with a number of sensors, a raspberry Pi and at least one network interface. The swarm of UAVs in a specific testbed is handled as micro-cloud that interacts independently with the central infrastructure. Multiple brokers are installed in the cluster close to the GCS and edge devices send telemetry messages to specific topic names. With this approach all messages avoid the excessive round trip delay of contacting a central Broker. UAVs consume command messages coming from the GCS in specific topics.

5.2.1 Distributed Communication Layer on UAVs

The message bus interconnects edge devices and control units. It is used for asynchronous notifications and method calls / response handling and for transmitting measurements that will be routed from producers (e.g., edge devices) to the consumers pertaining to control side and to any other application/service. Apache Kafka has been chosen for implementing the message bus, by taking into consideration the following aspects / advantages:

- capability to automatically spread data and, consequently, ability to assign workloads across a cluster of machines, thus allowing fine-tuning, load balancing and scalability of a clustered environment,
- capability to automatically replicate data over multiple servers (brokers), thus ensuring fault tolerance,
- built-in persistence mechanisms, which allows the system to easily deal with issues like the temporary overload of the network connection, or temporary disconnections
- high throughput, in terms of messages per seconds.
- build in security mechanisms that can be enabled during message exchange

In the following subsections we describe the main components of Apache Kafka that are used in our framework as shown in Figure 5.1.

5.2.1.1 Apache Kafka Clients

Apache Kafka Clients are software client implementations for producing and consuming messages using Apache Kafka as message broker. Besides the Java Client, clients in other several languages are provided, such as Python and a native C/C++ client, which are suitable for running the software in embedded devices with very specific requirements (e.g. limited memory). Currently, for most of the platform components the Java client implementation is utilized, while the C/C++ client has been adopted and adapted on the UAVs side.

5.2.1.2 Topics and Partitions

There are two main concepts that need to be taken into account for realizing specific communication work flows in a message bus: ”topics” and ”partitions”. Each message is assigned by the producer to a specific topic, and therefore written by Apache Kafka into this topic. In general terms, each topic can be seen as a container of particular type of information, therefore allowing information consumers to clearly identify the information they are interested in, and to subscribe only to the related topics.

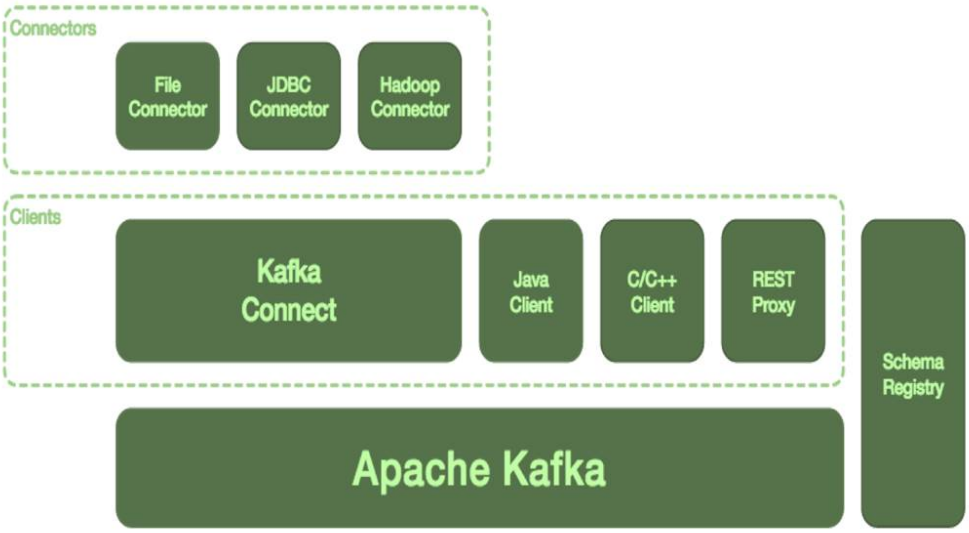


Figure 5.1: Main components of Kafka

The concept of **partitions** is, on the other hand, mainly used for load balancing across different Apache Kafka clients: message logs belonging to a specific topic can be assigned to different partitions in the same or different brokers across the Apache Kafka cluster, so that write and read operations are distributed among different servers based on the workload. Partitions are identified by producers and consumers using specific keys.

In order to achieve data redundancy in an Apache Kafka cluster copies of the different partitions are saved in different nodes as replicas. Apache Kafka permits $n - 1$ broker failures where n is the number of brokers. Of course the number of replicas is dependable with the number of brokers in the Apache Kafka cluster. It is not possible to define a topic with replication factor greater than the number of brokers that you have.

In addition, each message stored in Apache Kafka is addressed by its logical offset in a given partition. By maintaining information about the offset, the consumer can keep track of the last messages consumed within a topic partition. A Kafka broker stores all messages received in a ring buffer for a configurable amount of hours (until disk is full or the max log size is reached). So messages could also be read hours later. Producers are implemented in a way so that they buffer messages locally, until they can be sent to a Apache Kafka broker.

An important parameter to consider when analyzing the different distribution patterns in design of our platform is the latency defined as the amount of time a message takes to reach the receiver/s, after it has been sent by the publisher. For this purpose, the following main configuration and deployment principles were applied for reducing the latency in the publish/subscribe communication mechanism with Apache Kafka between UAVs and

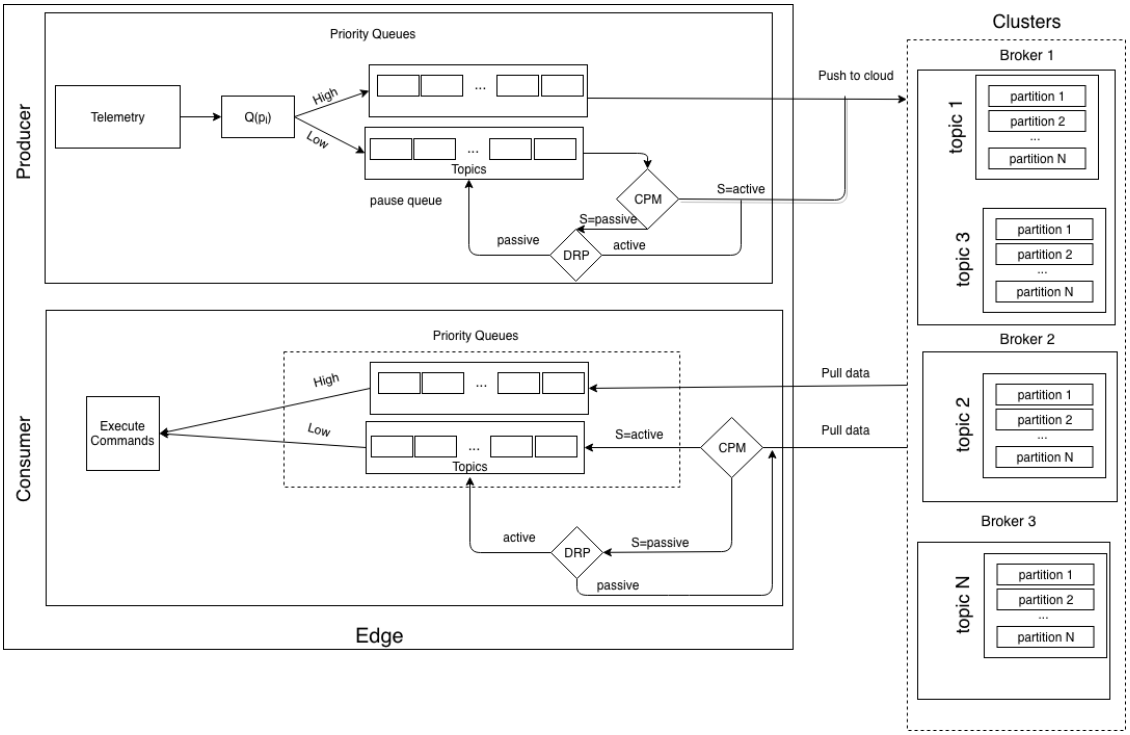


Figure 5.2: Decision Making Processes applied on Kafka Priorities Queues.

control units:

1. use of different partitions (a partition in Apache Kafka is the equivalent of a message stream for other messaging systems, which can be spread across different servers for scalability) for the different UxVs. This ensures that the messages of the various UxVs do not intermix, and it provides much shorter message bus queues dedicated to a particular UxV and much faster response times. Consumers can also be organized in consumer groups allowing multiple UxVs to read from multiple partitions in a topic.

2. use of a local Message Bus (message broker) installation within each local infrastructure. This way, the internal communication between e.g. the software Ground Control Systems and the UxVs will be performed in a local, controlled network environment, thus reducing the impact of the network in the latency of the communication. The overall workload in the message bus will be reduced (since each broker will just handle its own messages). Messages from each local broker can be anyway mirrored to a centralized Kafka broker deployed in the Cloud, so that Middle Tier components which need to access specific messages (e.g. logs or other data for experiments' control) will directly access the central message broker rather than each of the local ones.

5.2.2 Message Priorities in Apache Kafka

To implement prioritization in Apache Kafka as shown in Figure 5.2, we had to make some improvements in the clients and also to tweak the usual behavior of topic handling inside a broker. Apache Kafka usually treats the messages inside a topic equally by trying to receive and deliver them the same, agnostic per the importance of every individual value. In a scenario, with low performance of distributed platforms, we have to make two main considerations regarding the prioritization of messages. First, a device has to limit the messages that producing to the absolute minimum. Those are the important messages that can lead to a successful mission accomplishment or at least the messages that can make the device returning to a situation where the network connectivity has recovered. Second, the messages received from the device have to be those that can help it to exit the critical part of a path that can lead to the device being unresponsive and consequently to the mission failure. That messages can be an alternate waypoint, the command to continue the scheduled plan, an attempt to establish an ad-hoc network with neighboring devices or even the temporal hibernation of the device until the situation is improved. The first issue can be addressed by providing the Kafka Producers the ability to make decisions depending on the KPIs performance metrics whether to send a message or not. This is achieved by categorizing the Kafka topics regarding their importance to these with high priority and to these with low priority. This can be achieved by simply renaming the topics (or creating new ones) adding the suffixes "_$HIGH$" and "_LOW" to their names. The following algorithm can be applied.

Algorithm 3 Producing with prioritization

$n \leftarrow$ current network quality

$t \leftarrow$ topic name

$m \leftarrow$ message

$k \leftarrow$ network quality threshold

function DecideProduce(n, t, m)

    if $n < k$ AND $t = HIGH$ **then**

        $produce(m)$

The second issue can be addressed by utilizing methods from the Apache Kafka Consumer API. Apache Kafka topics are log files and every message gets an id depending on the size of the file the moment that this message was produced. When an Apache Kafka consumer fetches data from a topic then it can get only the new messages by calculating the size of the file the last time that a message was consumed and comparing it with the current size of the file. If the later is greater than the first then it starts from the first new message to consume i.e. the message that is currently having the id that equals the last size of the log file. To consume with priorities we follow a similar procedure but we target the high priority topics. A consumer will fetch data from low priority topics only if the high

priority topics haven't received new messages thus ensuring that the first messages will be received, are the ones with higher importance. The methods of the consumer API that were used were pause(), resume(), position() and endOffsets() and the following procedure was implemented in the consumers running on the edge.

Algorithm 4 Consuming with prioritization

$h \leftarrow$ high priority topic

$n \leftarrow$ current network quality

$k \leftarrow$ network quality threshold

$e \leftarrow$ endpoint of high priority topic

$o \leftarrow$ offset of high priority topic

$l \leftarrow$ low priority topic

function DecideConsume

    while $true$ **do**

        $consume(h)$

        if $n < k$ AND $e < o$ **then**

            $consume(l)$

5.2.3 Optimal Stopping Delivery Strategy

The prioritization of messages as described above can be managed by a decision making process in both edge devices and control units. DMP is monitoring performance metrics from the message distributed platform or the underlying communication network and decide if a change occurs from an "acceptable" performance to a critical situation. If a change occurs, then DMP can decide how to control the delivery (producing and consuming) of Kafka queue prioritization messages on both edge to control units and vice versa dynamically. KPIs can be various metrics such as the packet loss of kafka message bus, average round trip time (RTT) of the packets and the total time needed between the packets transmitted and received. In other words we are interested in finding a stopping time that detects a change to the performance of Kafka message bus with the minimum delay based on random KPIs coming to our system. This description can be approached as authors proposed in [42] change detection problem among two states normal S_n and saturated S_s. The possible states of our analysis are coming form a vector combination of all features in normal and saturated network. For simplicity reasons we describe the possible states of two KPIs, i.e. packet loss and RTT: good packet loss- good RTT, good packet loss - bad RTT, bad packet loss - good RTT and bad packet loss - bad RTT. As shown in Figure 5.3 two dominant areas exist described as a good packet loss - good RTT

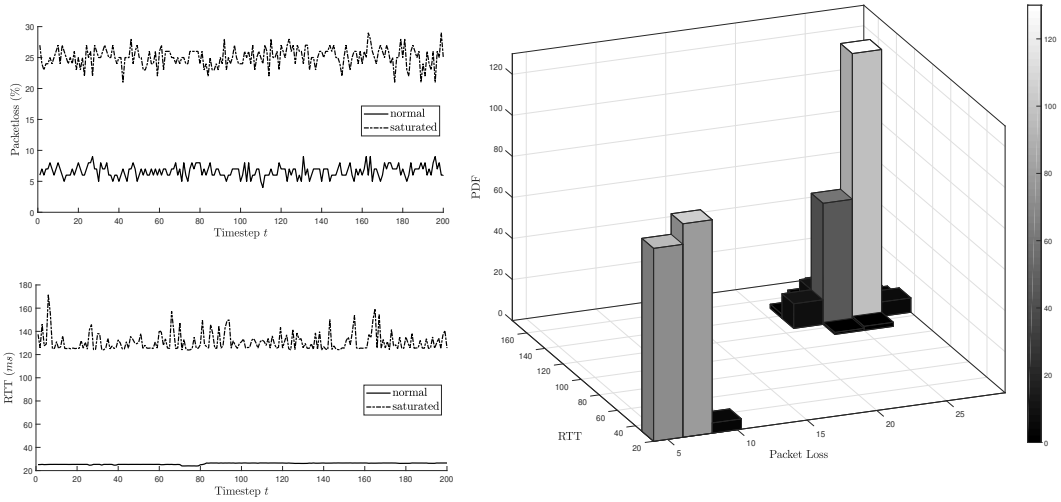


Figure 5.3: Joint probability Density function of RTT and packet loss in normal and saturated state.

$\in S_n$ and bad packet loss - bad RTT $\in S_s$. Hence change point detection theory can be applied on the two distinct states normal S_n and saturated S_s.

1. Detection Step

We consider a sequence of discrete random signal with independent and identically distributed (i.i.d.) random variables observed sequentially in real time y_k with a probability density $p_\theta(y)$, i.e. KPI indicators. Before an unknown change t_m the parameter θ is equal to θ_0 and after the change it is equal to θ_1 where $\theta_0 \neq \theta_1$. In our case, θ_i expresses the normal distribution with mean value μ_i and variance σ_i. To estimate $p_{\theta_i}(y)$, a probability density function comparison method has been adopted to derive the *closest* distribution to KPI values. At each timestep t_i a decision rule is applied between two states about the parameter θ:

$$P(y) = \begin{cases} \prod_{n=0}^{k} p(y_n, \theta_0), \ s \in S_n; \\ \prod_{n=0}^{m-1} p(y_n, \theta_0) \prod_{n=m}^{k} p(y_n, \theta_1), s \in S_s, \text{ change} \end{cases} \tag{5.1}$$

A solution proposed by Lorden [41] use the so-called likehood ratio $L_y(n)$ and the cumulative summation of the log-likehood ratio $S(n)$:

A feasible solution derived by the change-point detection theory adopts the minmax approach in [41][42]. For the change point detection we introduce the cumulative summation

of the the log-likelihood ratios at time n by $L_x(n)$ and its :

$$L_y(n) = \ln \frac{p(y(n), \theta_0)}{p(y(n), \theta_1)} \tag{5.2}$$

$$S(n) = \sum_{k=0}^{n} L_x(k) \tag{5.3}$$

$$= \sum_{k=0}^{n} \ln \frac{p(y(n), \theta_0)}{p(y(n), \theta_1)} \tag{5.4}$$

When the change time m is unknown, the standard statistical approach consists of estimating it by using the maximum likelihood principle, which leads to the following decision function :

$$G(n) = \max_{1 \leq k \leq n} \sum_{i=k}^{n} L_y(i) \tag{5.5}$$

The optimal stopping time for detecting the change point is given by:

$$\tau^* = \min\{n \geq 1, \max_{1 \leq k \leq n} \sum_{i=k}^{n} L_y(i) \geq \alpha\} \tag{5.6}$$

Then decide if we are in state S_s if $G(n) \geq \alpha$ (else we are in S_n), where α is a threshold set by the user and normally express the rate of false alarms. Clearly, this condition is equivalent to limit the rate of false detection by a given maximum value. Following CUSUM optimal solution for change detection we compute the maximum like-likelihood estimate, which is the value of τ^* maximizing the likelihood $p(y_m, \theta_1)|S_s$

$$\tau^* = arg \max_{1 \leq m \leq n} \sum_{i=m}^{n} L_y(i) \tag{5.7}$$

$$= arg \max_{1 \leq m \leq n} \sum_{k=m}^{n} \ln \frac{p(y(n), \theta_0)}{p(y(n), \theta_1)} \tag{5.8}$$

$$\tag{5.9}$$

From the above the Equation 5.3 can be re-written for the time change m as

$$S(n|m) = S(n) - S(m-1) \tag{5.10}$$

and then we can extract the expressions

$$G(n) = S[n] - \min_{1 \leq m \leq n} S[m-1] \tag{5.11}$$

$$\tau^* = arg \min_{1 \leq m \leq n} S[m-1] \tag{5.12}$$

Equation 5.11 shows that the decision function $G(n)$ is the current value of the cumulative sum $S(n)$ minus its current minimum value. Equation for τ^* shows that the change time estimate is the time following the current minimum of the cumulative sum. Therefore, each step composing the whole algorithm relies on the same quantity: the cumulative sum $S(n)$ which this explains the name of cumulative sum or CUSUM algorithm.

2. Recall Step

When the change point detection model (CPM) is activated then only the messages belonging in the high priority queue are produced and consumed as shown in Figure 5.2. However the pausing period cannot be infinite as long as it is applied in UAVs during mission. Going a step further from applying a heuristic threshold activator we introduce in this phase a discounted reward optimal stopping model. CPM is a finite horizon solution similar to secretary problem:

- The goal is to find the best KPI metric that will restore the management (producing/consuming) of all priorities queues before time reaches the maximum th threshold T_h
- There is a limited and known time to deliver the message.
- When the maximum T_h is reached or a decision is coming from CPM then producers and consumers in the devices are sending and receiving the data constantly from both categories of queues.
- KPIs can be ordered from the worst to the best according to their value
- A reward value is assigned to each observation y_i received.
- Reward value has a penalty factor as long as i-th observation gets closer to T_h
- There is no recall to value of KPI that was rejected in a specific timeslot

CPM problem is quite similar to the standard secretary problem. The main difference is that the payoff obtained is discounted with a factor as long as ”the time flies” and the decision shall be selected as soon as possible. The basic idea is to enforce the decision making process while messages are not exchanged in low priority queue. It is obvious that in case of unmanned vehicles we cannot freeze the monitoring of telemetry values for a long period of time. The optimal stopping time τ^* is equal to the minimum value of the following $\min(\sum_{i=r}^{N-1} \frac{1}{i} + r\frac{2\frac{\gamma}{N}}{1-\frac{\gamma}{N}} - \frac{1+\gamma}{1-\frac{\gamma}{N}})$ as shown in Algorithm 5 in Appendix C. For $\gamma = 0$ and a large N, we obtain the classical optimal stopping rule $r^* = \frac{N}{e}$.

5.3 Performance Evaluation

We report on an experimental evaluation to compare the performance of our framework. We have used a UAV simulator and the role of control station was handled by a fixed

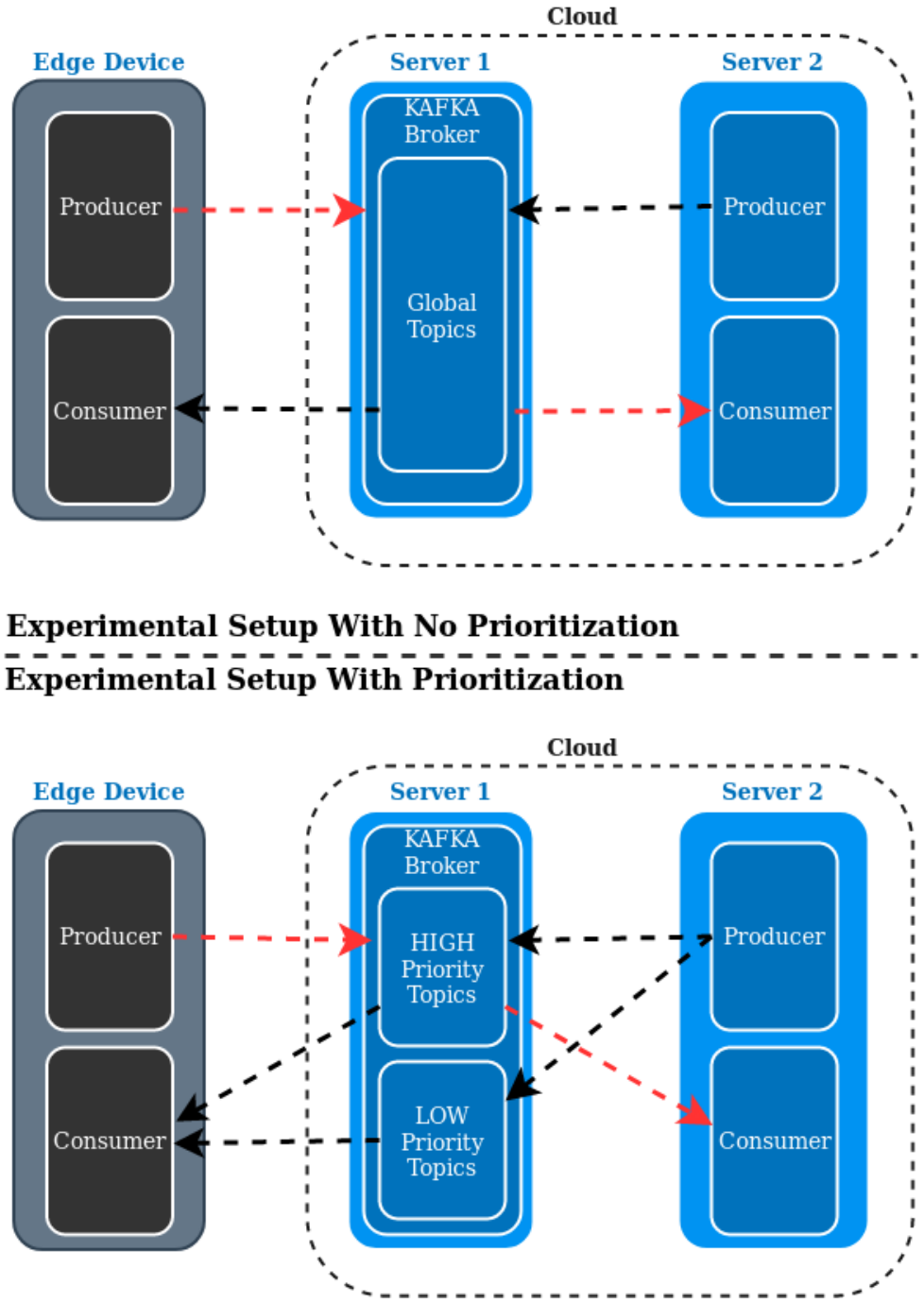


Figure 5.4: Experimental setup for simulated scenario.

server. UAV simulator and GCS are part of the *Road-, Air- and Water-based Future Internet Experimentation* (RAWFIE) [1] platform which offers a framework for interconnecting numerous testbeds over which remote experimentation can be realized. RAWFIE platform originates in a H2002 EU-funded (FIRE+ initiative) project which focuses on the mobile IoT paradigm and provides research and experimentation facilities through the ever growing domain of unmanned networked devices (vehicles). We performed 100 simulation runs with duration 10 minutes. In each second 2000 messages were produced and distributed in the message platform for each priority queue. The change point detection thresholds is randomly selected in the range $[0, 1]$ while for CPM γ is randomly selected in the interval $[1, 10]$ and the maximum threshold coming from unmanned manufacturers to safely freeze the messaging of the devices is equal to 60 seconds.

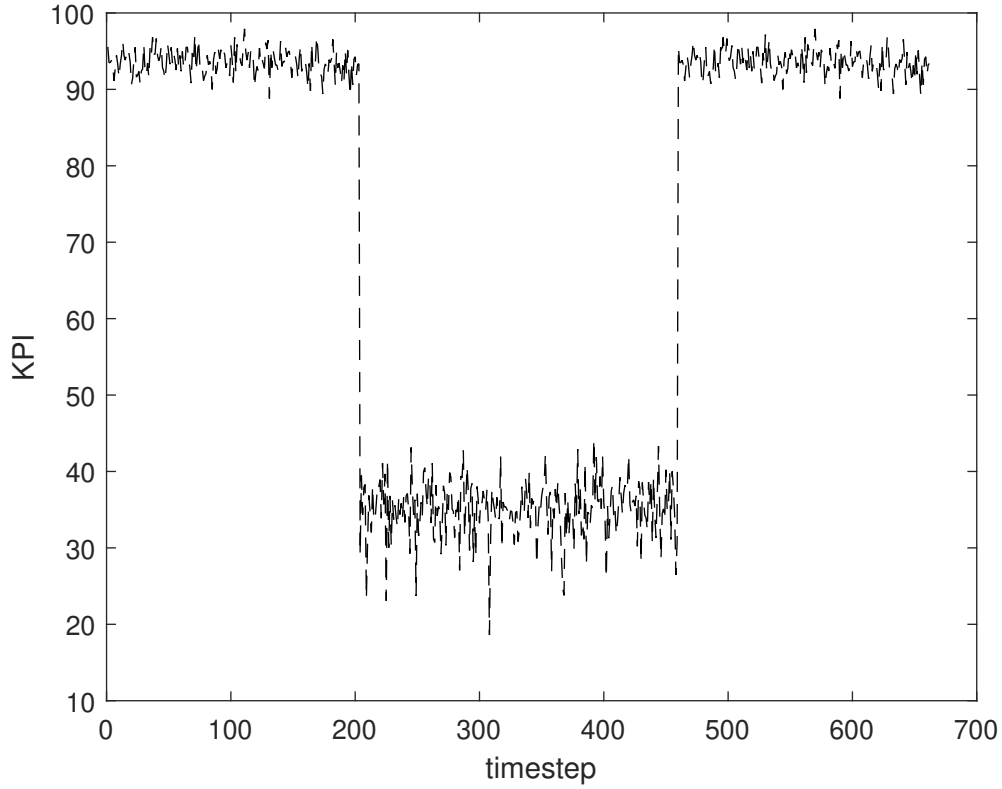
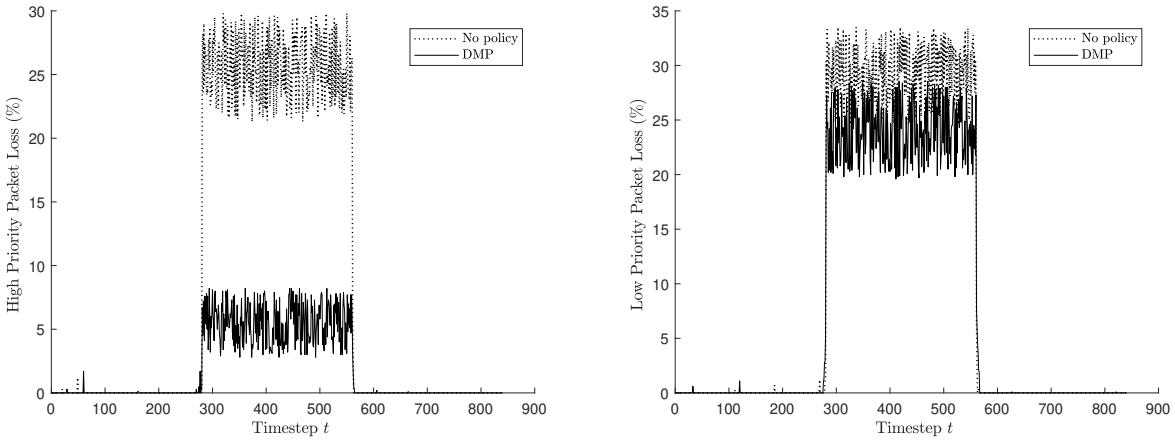


Figure 5.5: KPI performance metrics measured during the simulation.

To perform experimental evaluation two kinds of Apache Kafka clients were tested in both good and saturated network performance conditions. The first set (no-policy) of producers and consumers were simple Java clients producing and consuming constantly messages unaware of the network conditions to measure the performance of a non-priority Apache Kafka installation.As shown in figure 5.4 a pair of producers and consumers were deployed in the cloud emulating a service that can be in the same data center with the Apache Kafka thus having a small delay by requesting or transmitting messages to the broker. Another

[1] www.rawfie.eu



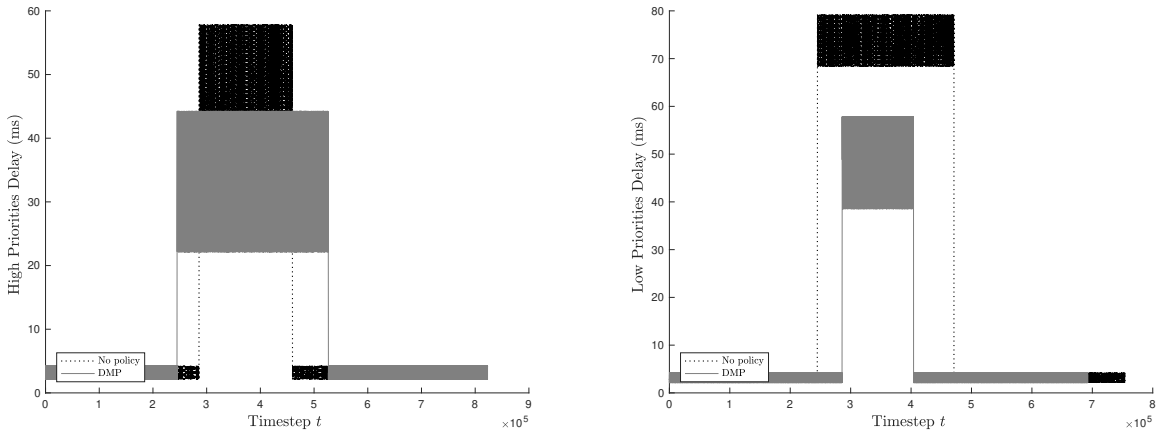
(a) (b)

Figure 5.6: Decicion making between states in S_n to S_s and then to S_n.

pair of producer/consumer was placed on an edge device that was vulnerable to network conditions. Both producers and consumers were sending and receiving messages equally from all the topics without distinguishing the ones with high or low priority. In the opposite scenario (DMP), a pair of enhanced clients aware of the network conditions were installed on the edge device to measure the performance of priority queues. The KPI performance of the distributed message bus was measured as the affine combination of the RTT and the Packet loss as depicted in Figure 5.5.

Figures 5.6a and 5.6b plot the packet loss measured in all the priority queues. In both Low and High priority queues the DMP policy outperforms against no-policy. Especially in the High priority queue, in which essential information is exchanged for the user, the packet loss is less than 10% while in the no-policy the mean value of the packet loss is in range $[25\%, 30\%]$. The small improvement for DMP policy in the Low priority queue can be explained from the brokers' side. Brokers in DMP handle less "bursts" of messages in total that cannot handle due to bad network performance.

The latency issues is more evident in Figures 5.7a and 5.7b. Applying the DMP decision making model the average value of delay in the messages successfully transmitted in Kafka message bus in saturated state S_s is between for High Priority Queue $[25 - 45]$ ms and for Low Priority Queue between $[45, 60]$ ms. This is not the use case for the non policy model where the delay metrics is twice the delay of DMP. In the S_s applying the DMP policy has the result of balancing the traffic in both Producers and Consumers in the Apache Kafka message bus.



(a) (b)

Figure 5.7: Delay in two priority queues between states in S_n to S_s and then to S_n.

5.4 Conclusions

This research work proposes a stochastic optimization framework that monitors the performance KPIs of a middleware data exchange platform trying to balance the traffic of exchanged messages with the minimum loss. We propose a model of dynamic decision making, adaptive to changes by dynamically adjusting the transmission in different priority queues in order not to overload a saturated message bus, to cause synchronization issues or to risk to loose completely the messages. The performance evaluation showed the successful delivery of messages in poor performance conditions and the moderate production of messages of High Priority messages so as not to burden an already saturated queue which leads to loose completely the messages.

6. CONCLUSIONS AND FUTURE WORK

In this thesis decision making models were investigated which can monitor with no prior knowledge information streams produced by IoT devices, can predict changes with online mechanisms that can disrupt the performance of the IoT framework and can take actions to retain acceptable Quality Of Service while trying to save resources. The online, time optimized and distributed decision making models are based on Optimal Stopping Theory and Change Detection Theory appied applied on the Edge, Communication and Middleware layer of a multi-layer resource management architecture.

Starting from edge layer in chapter 3, a content-based MPEG encoder using OST decision rule for the conclusion of GOP and the transmission of intra- coded frames was presented. Dynamic encoding applied to infrastructures with restricted resources, like IoT camera networks, is needed in order to support media-rich applications in such infrastructures. Limited bandwidth and battery lifetime require nowadays content-driven transmission rates and processing of the video sequences. One major contribution of this thesis is the adaptation to video changes; I frames are created when scene changes are detected which leads to significant resource savings while retaining equal quality levels. The proposed encoder can be applied to facilities with restricted resources like WSMNs in order to transmit video sequences in an acceptable quality. The aim is twofold: to create different size of GOPs adaptive to the transmitted video streams and to try to save resources with a small SATD error. Experiments showed that the GOP size was extended in order to avoid unnecessary transmissions. We observe that the stream volume transmitted in most of the cases is smaller than the CE created bitstream which justifies that fixed encoders which are not content-driven lead to waste of network resources. The encoder focuses on the transmitted video content and, thus, the values of SATD stay lower than the classic approach. Our future agenda includes the expansion of our study toward the inclusion of bidirectional (B) frames in the OST controlled video stream. B frames are created by examining the difference between the previous and the next reference frame and this surely imposes changes in the OST strategy applied for the GOP inclusion. However B frames require less resources when stored or transmitted and this can further lead to savings on the resources employed for video transmission. Additionally, our future work includes applying OST policies to block matching algorithms that are used to identify the matching blocks in video sequences for motion estimation. Motion estimation is used to discover temporal redundancy between two images or two video frames by comparing blocks of pixels. OST policies can stop the exchaustive search of a search region in a block of pixels reducing estimation time and complexity.

In chapter 4, we investigate the performance of the underlying wireless network and how the changes in network environment can affect and disrupt the mission of unmanned robotic devices and the telemetry received. A decision making model adaptive to network quality changes applied in mobile IoT domain was proposed. The Mobile IoT domain has been significantly expanded with the proliferation of drones and unmanned robotic devices. In this new landscape, the communication between the resource-constrained device and the fixed infrastructure is similarly expanded to include new messages of vary-

ing importance, control, and monitoring. To efficiently and effectively control the exchange of such messages subject to the stochastic nature of the underlying wireless network, we studied and designed a time-optimized, dynamic, and distributed decision making mechanism based on the principles of the Optimal Stopping and Change Detection theories. When the quality of the network significantly changes, the UxV and the GCS can decide in real-time to pause/start the transmission of telemetry in order not to overload a saturated network, or to risk to lose completely the messages. The findings from our experimentation platform were promising showing the successful delivery of messages in poor network conditions in different unmanned vehicle's mission. Our future research agenda includes the adoption of our TOCP-DRP model in a swarm of UxVs in order to handle the offloading of the services / tasks, e.g., generation of telemetry, between the swarm entities. We also plan to apply our TOCP-DRP mechanism adding the navigation of the robot in space. A change detection framework can predict the possible chnages in network and this will trigger changes in the navigation waypoints in space. It can explore in real time alternative paths or it can return to a previous state in order to successfuly deliver critical information.

Last but not least a stochastic optimization framework of on-line control unit applied on a data distributed platform was proposed as long as these platforms are necessary to IoT infrastructures to process the enormous volumes of data exchanged between IoT devices. The proposed stochastic optimization framework monitors the performance KPIs of a middleware data exchange platform trying to balance the traffic of exchanged messages with the minimum loss. It was designed a model of dynamic decision making, adaptive to changes by dynamically adjusting the transmission in different priority queues in order not to overload a saturated message bus, to cause synchronization issues or to risk to loose completely the messages. The performance evaluation showed the successful delivery of messages in poor performance conditions and the moderate production of messages of High Priority messages so as not to burden an already saturated queue which leads to loose completely the messages. Our future agenda includes the support of the on-site deployment of a local broker depending on the conditions met by the decision making procedure. Upon estimating that the network conditions will cause problems and delay to the message delivery, the framework will deploy automatically a message broker on-site to serve low priority messages so that it will not get lost in the process. With the network improvement, all the messages that were transmitted will be transferred to the main message broker. This mechanism could also potentially lower the pressure on the network and the delivery of high priority messages could also benefit from the smaller quantity of network traffic. The clients and the decision-making model described in this thesis will be the basis for this new architecture.

ABBREVIATIONS - ACRONYMS

IoT	Internet of Things
WSN	Wireless Sensor Networks
WMSN	Wireless Multimedia Sensor Networks
GoP	Group of Pictures
OST	Optimal Stopping Theory
DCT	Discrete Cosine Transform
MCMC	Markov Chain Monte Carlo
SATD	sum of absolute differences between the two frames (inter-frame variation
$\Gamma(a, b)$	Gamma distribution
$W(y)$	pay off of the real-valued function
DGPE	dynamic grouping of pictures encoder for the gamma distribution
NDGPE	dynamic grouping of pictures encoder for the normal distribution
DMP	Decision Making Process
TOCP	Time-Optimized Change Point DMP
DRP	Optimal Discounted Reward DMP
QNI	Quality Network Indicator
$p(x_n, f)$	Probability Density Function with parametric density f
f_i	Normal distribution $\mathcal{N}(\mu_i, \sigma_i)$
H_0	No-Change-point Hypothesis
H_1	Change-point Hypothesis
FAR	False Alarm Rate
N_d	Detection Change time
α	Detection threshold: $0 \leq \alpha \leq FAR$
$\gamma \in [0, 1]$	Discounted factor at DRP policy
$r(\gamma, N) \in \{1, \ldots, N\}$	Stopping time in DRP policy up to time N
t^*, τ^*, r^*	Optimal stopping times in generic OST, Change-point Detection,

	and DRP policy, respectively
CPM	Change point detection model
$L_x(n)$	Log-likelihood ratio at time n for random variable X
KPI	Key Performance Indicators
JMS	Java Message Service
RTT	Round Trip Time
S_n	Normal State
S_s	Saturated State
UAV	Unmanned Aerial Vehicle
GCS	Ground Control Station
DRP	Optimal Discounted Reward DMP
$\mathcal{N}(\mu_i, \sigma_i)$	
T_h	Maximum horizon where a UxV can be paused

APPENDIX A. PRELIMINARIES IN OPTIMAL STOPPING THEORY & CHANGE DETECTION THEORY

Before elaborating on our problem formulation and the proposed time-optimized mechanisms, we provide the fundamentals and principles adopted from the OST and the change-detection theory.

A.1 Optimal Stopping Theory

The first studied optimal stopping problem is related with the problem of choosing a time to take a given action based on a sequentially observed random variables in order to maximize an expected payoff. In addition our stopping time problem has a finite horizon, i.e., there is an upper bound on the number of stages at which we may stop.

Let $\mathbb{F}_n$ be defined as the σ-algebra generated by the random variables $Y_1, Y_2, \cdots, Y_n$ in a probability space $(\Omega, \mathbb{F}, P)$. We envisage $\mathbb{F}_n$ as the filtration (information) observed up to (discrete) time instance n by collecting the realization values of the random variables up to n. For instance, in our context $Y_1, Y_2, \cdots, Y_n$ are considered the observed Quality of Network Indicator (QNI) values in discrete timesteps $t = 1, \ldots n$. A *stopping rule* or *stopping time* is defined as the random variable τ with realization values in a set of natural numbers such that $\{\tau = n \in \mathbb{F}_n\}$ for $n = 1, 2, \ldots$ and probability $P(\tau < \infty) = 1$. We denote with $\mathbb{M}(n, N)$ the class of all stopping rules τ in which $P(n \leq \tau \leq N) = 1$ for any $n = 1, 2, \ldots$ and $N > 0$. The real-valued pay off function is then defined as the mapping $W : \mathbb{R} \rightarrow \mathbb{R}$ being a Borel measurable function which values $W(y)$ interpret the pay off of a decision maker when it stops the Markov chain $(Y_n, \mathbb{F}_n)$ at the state $y \in \mathbb{R}$. In our case, the reward can be defined as the selection of the best network metric (QNI value) reached so far.

Assume now that for a given state y and for a given stopping rule τ, the expectation of the reward (pay-off) function is $\mathbb{E}[W(Y_\tau)|Y_1 = y]$ exists. Then, the expected pay off $\mathbb{E}[W(Y_\tau)|Y_1 = y]$ corresponding to a chosen stopping rule τ exists for all states $y \in \mathbb{R}$, which refers to the *reward value* of the stopping problem. Based on the *principles of optimality* the *reward value* $V_N(y)$ is the supremum of the expected pay off of *all* the stopping rules belonging to $\mathbb{M}(1, N)$, i.e.,

$$V_N(y) \;=\; \sup_{\tau \in \mathbb{M}(1,N)} \mathbb{E}[W(Y_\tau)|Y_1 = y], \tag{A.1}$$

where the supremum is taken for all stopping rules $\tau \in \mathbb{M}(1, N)$ for which the expectation $\mathbb{E}[W(Y_\tau)|Y_1 = y]$ exists for all $y \in \mathbb{R}$. Based on the optimal value $V_N(y)$, where the supremum in (A.1) is attained, the *optimal stopping rule* $t^* \in \mathbb{M}(1, N)$ should satisfy the condition:

$$V_N(y) \;=\; \mathbb{E}[W(Y_t^*)|Y_1 = y], \forall y \in \mathbb{R}. \tag{A.2}$$

It is then clear that the optimal value $V_N(y)$ is the maximum possible excepted reward to be obtained observing the random variables $Y_1, \ldots, Y_N$ up to the N-th observation.

Consider also that the expectations $\mathbb{E}[W(Y_\tau)|Y_1 = y]$ exist for all $y \in \mathbb{R}$ and, based on the principles of optimality. Let us then introduce the operator $\mathcal{Q}$ over the reward function $W \in \mathbb{R}$ such that:

$$\mathcal{Q}W(y) \quad = \quad \text{max}\{W(y), \mathbb{E}[W(Y_t^*)|Y_1 = y]\}. \tag{A.3}$$

Then, the optimal stopping rule t^*, which attains the optimal value in (A.2), is estimated by the Theorem 2:

Theorem 2. *Assume that $W \in \mathbb{R}$. Then:*

- $V_n(y) = \mathcal{Q}^n W(y)$*,* $n = 1.2, \ldots$*;*
- $V_n(y) = \text{max}\{W(y), \mathbb{E}[V_{n-1}(Y_1)]\}$*, where* $V_0(y) = W(y)$*;*
- *the optimal stopping time t_n^* is evaluated as:*

$$t_n^* \quad = \quad \text{min}\{0 \leq k \leq n : V_{n-k}(y) = W(y)\}, \tag{A.4}$$

This refers to an optimal stopping rule in $\mathbb{M}(1, n)$. If $\mathbb{E}[|W(Y_k)|] < \infty$, for $k = 1, \ldots, n$, then the stopping rule t_n^* in (A.4) is optimal in the class $\mathbb{M}(1, n)$.

Proof. Please refer to [23]. □

A.2 Change Point Detection Theory

The second category of the optimal stopping problem is the detection of a change point. Consider that we are monitoring a sequence of a sequence random variables, like values of the QNI, $\{Y_1, Y_2, \ldots Y_n\}$ with a known distribution f_0. At some point m in time, unknown to us, the distribution changes to another known distribution f_1. Our goal is to detect the change as soon as it occurs. Let $\mathbb{F}_n$, $n \geq 1$ be the σ-algebra generated by the random variables $\{Y_1, Y_2, \ldots Y_n\}$. A sequential *change point detection rule* is then derived by the stopping time τ of the observed values. The stopping time τ for the change point detection has the following characteristics:

- Average Run Length (ARL): ARL, proposed in [40], is defined as the expected number of observed values before a change decision is taken, where N_d is the detection time and f is assumed to be constant, i.e., $ARL = \mathbb{E}[N_d]$
- The Detection Delay D_n is the average detection delay corresponding to the observed $\{Y_1, Y_2, \ldots Y_n\}$ needed before a detection change occurs. Therefore, this quantity has to be as small as possible to minimize the reaction time of the algorithm.

- The False Alarm Rate FAR [30] is calculated as the ratio between the number of negative events wrongly categorized as changes.

In the following, we describe the two in-network/on-device optimal stopping rule mechanisms running on the UxV; the same mechanisms also run on the GCS.

APPENDIX B. REAL-TIME STOCHASTIC CONTROL MECHANISM ADAPTIVE TO CHANGES IN NETWORK QUALITY BASED ON OPTIMAL SEQUENTIAL DECISION MAKING RULES

B.1 Proof

Proof. The function $L^*(\cdot)$ of the log-likelihood ratio between $\bar{f}_0$ and $\bar{f}_1$ is continuous over the support of $\bar{f}_1$ and has an extremum. The proof is based on the first derivative test:

$$\frac{dL}{dx} = \frac{2(x-\mu_1)}{2\sigma_1^2} - \frac{2(x-\mu_0)}{2\sigma_0^2} = x(\frac{\sigma_0^2-\sigma_1^2}{\sigma_0^2\sigma_1^2}) + \frac{\mu_0\sigma_1^2-\mu_1\sigma_0^2}{\sigma_0^2\sigma_1^2}. \quad \text{(B.1)}$$

For $\mu_0 > \mu_1$ and $\sigma_1 > \sigma_0$, we obtain that $x^* = \frac{\mu_1\sigma_0^2-\mu_0\sigma_1^2}{\sigma_0^2-\sigma_1^2}$.

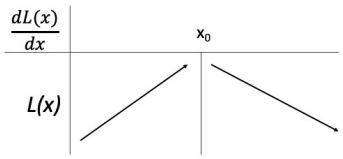


Figure B.1: Monotony analysis of L

□

B.2 CUSUM algorithm

In the CUSUM algorithm, we further define the generalized log-likelihood ratio G_x

$$\begin{aligned} G_x[k] &= max_{1\leq m\leq k} L_x[k,m] = max_{1\leq m\leq k} \sum_{n=m}^{k} ln\frac{p(,x(n)f_0)}{p(,x(n)f_1)}, \\ &= S[k] - min_{1\leq m\leq k} S[m-1], \end{aligned}$$

where $\hat{m}$ is defined as

$$\hat{m} = \arg\min_{1\leq m\leq k} S[m-1] \quad \text{(B.2)}$$

Equation B.2 shows that the decision function $G[k]$ is the current value of the cumulative sum $S[k]$ minus its current minimum value. Equation B.2 shows that the change time estimate is the time following the current minimum of the cumulative sum. Therefore, each step composing the whole algorithm relies on the same quantity: the cumulative sum $S[k]$. This explains the name of cumulative sum or CUSUM algorithm.

APPENDIX C. TIME-OPTIMIZED PRIORITIZATION OF KAFKA MESSAGE SCHEDULING FOR UNMANNED VEHICLES IN IOT NETWORKS

Algorithm 5 CPM

$T \leftarrow$ maximum threshold
$c \leftarrow$ counter in CPM
$\gamma \leftarrow$ discount factor
$r \leftarrow$ number of observations
$r^* = OST_optimal(r, \gamma)$
$decision \leftarrow false$
while c<T AND decision==False **do**
    measure the current y
    if c< r^* **then**
        if $y > y^*$ **then**
            $y^* \leftarrow y$
    else
        if $y > y^*$ **then**
            $y^* \leftarrow y$
            $decision \leftarrow True$
    $c \leftarrow c + 1$

Algorithm 6 $OST_{optimal}$

function $OST_{optimal}$(r,γ)
    $Y \leftarrow List$
    for $1 < n < r$ **do**
        $s \leftarrow 0$
        for $n < k < \gamma$ **do**
            $s \leftarrow s + \frac{1}{n}$
            $y \leftarrow y + r\frac{2\frac{\gamma}{T}}{1-\frac{\gamma}{T}} - \frac{1+\gamma}{1-\frac{\gamma}{T}}$
        Y.add(y)
    return min(Y)

REFERENCES

[1] Amqp - advanced message queuing protocol.

[2] Apache activemq message broker.

[3] Mqtt- message queuing telemetry transport.

[4] Rabbitmq.

[5] Stomp - simple text oriented message protocol.

[6] A.Dumitras and B.G. Haskell. I/p/b frame type decision by collinearity of displacements. In *IEEE Int. Conf. Image Process*, volume 4, pages 2769–2772, Oct. 2004.

[7] Ibrahim Ahmed I Alghamdi, Christos Anagnostopoulos, and Dimitrios P. Pezaros. On the optimality of task offloading in mobile edge computing environments. In *IEEE Global Communications Conference 2019*, Hawaii, USA, 09-13 Dec 2019.

[8] C. Anagnostopoulos and S. Hadjefthymiades. Delay-tolerant delivery quality information in ad hoc networks. *Journal of Parallel and Distributed Computing*, 71(7):974–987, 2011.

[9] C. Anagnostopoulos and S. Hadjefthymiades. Optimal quality-aware scheduling of data consumption in mobile ad hoc networks. *Journal of Parallel and Distributed Computing*, 72(10):1269–1279, Oct. 2012.

[10] C. Anagnostopoulos and S. Hadjiefthymiades. Advanced principal component-based compression schemes for wireless sensor networks. *ACM Trans. Sen. Netw.*, 11(1):7:1–7:34, July 2014.

[11] A.G. Nguycn Anthony Y. Lan and J-N Hwang. Scene context-dependent reference-frame placement for mpeg video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(3):478–489, 1999.

[12] ArduPilot Open Source Autopilot. Open source. `http://ardupilot.org/`, Accessed: 2019-06-29.

[13] Jiang X. Azim M. *Wireless Sensor Multimedia Netwoks: Architectures, Protocols and Applications*. CRC Press, October 27,2015.

[14] A. Ferman B. Gunsel and A. Tekalp. Temporal video segmentation using unsupervised clustering and semantic object tracking. *Electronic Imaging*, 7:592–604, 1998.

[15] Neil Bearden. A new secretary problem with rank-based selection and cardinal payoffs. *Journal of Mathematical Psychology - J MATH PSYCHOL*, 50:58–59, 02 2006.

[16] Carlos Borrego, Gerard Garcia-Vandellós, and Sergi Robles. Softwarecast: A code-based delivery manycast scheme in heterogeneous and opportunistic ad hoc networks. *Ad Hoc Networks*, 55:72–86, 2017.

[17] Carlos Borrego Iglesias, Joan Borrell, and S Robles. Efficient broadcast in opportunistic networks using optimal stopping theory. *Ad Hoc Networks*, 88, 05 2019.

[18] Carlos Borrego Iglesias, Joan Borrell, and S Robles. Hey, influencer! message delivery to social central nodes in social opportunistic networks. *Computer Communications*, 137, 02 2019.

[19] Carlos Borrego Iglesias, Adrián Sánchez-Carmona, Zhiyuan Li, and S Robles. Explore and wait: A composite routing-delivery scheme for relative profile-casting in opportunistic networks. *Computer Networks*, 123, 05 2017.

[20] Y. Cao and Z. Sun. Position based routing algorithms for ad hoc networks: A taxonomy. *Ad Hoc Wireless Networking, Kluwer*, 2003.

[21] A. Dhekne, M. Gowda, R. R. Choudhury, and S. Nelakuditi. If wifi aps could move: A measurement study. *IEEE Transactions on Mobile Computing*, 17(10):2293–2306, Oct 2018.

[22] Philippe Dobbelaere and Kyumars Sheykh Esmaili. Kafka versus rabbitmq: A comparative study of two industry reference publish/subscribe implementations: Industry paper. In *Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems*. ACM, 2017.

[23] Thomas S. Ferguson. *Optimal Stopping and Applications*. Mathematics Department, UCLA, Accessed May, 2015.

[24] F.Fu and M. van der Schaar. Dependant optimal stopping framework for wireless multimedia transmission. In *Accoustics speech and Signal Processing (ICASSP)*, pages 1–6. IEEE, 2010.

[25] Ai-Chun Pang Hao-Min Lin, Yu Ge and J. S. Pathmasuntharam. Performance study on delay tolerant networks in maritime communication environments. In *OCEANS 2010 IEEE-Sydney*, 2010.

[26] A. Kankanhalli H.J. Zhang and S.W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems*, 1:10–28, 1993.

[27] Steve Hoelzer. Mpeg-2 overview and matlab codec project. `http://users.cs.cf.ac.uk/Dave.Marshall/Multimedia/Lecture_Examples/Compression/mpegproj/`, 2005.

[28] Steve Hoelzer. Xiph org. `https://media.xiph.org/video/derf/`, 2010 (Accessed: 2017-02-24).

[29] Hsu-Feng Hsiao and Chen-Tsang Wu. Balanced parallel scheduling for video encoding with adaptive gop structure. *IEEE Trans. Parallel Distrib. Syst*, 24:2355–2364, 2013.

[30] V. V. Veeravalli J. Unnikrishnan and S. Meyn. Least favorable distributions for robust quickest change detection. In *2009 IEEE International Symposium on Information Theory , IEEE*, 2009.

[31] Kostas Kolomvatsos, Michael Tsiroukis, and Stathes Hadjiefthymiades. An experiment description language for supporting mobile iot applications. In Martin Serrano, Niklaos Isaris, Hans Schaffers, John Domingue, Michael Boniface, and Thanasis Korakis, editors, *Building the Future Internet through FIRE*, River Publishers series in information science and technology, pages 461–460. River Publishers, Gistrup, Denmark, 2017.

[32] C. Konstantopoulos, G. Pantziou, D. Gavalas, A. Mpitziopoulos, and B. Mamalis. A rendezvous-based approach enabling energy-efficient sensory data collection with mobile sinks. *IEEE Transactions on Parallel and Distributed Systems*, 23(5):809–817, May 2012.

[33] I. Koprinska and S. Carrato. Detecting and classifying video shot boundaries in mpeg compressed sequences. In *IX European Signal Processing Conf. (EUSIPCO)*, pages 1729–1732, 1998.

[34] Krulikovska L. A novel method of adaptive gop structure based on the positions of video cuts. In *ELMAR*, pages 67–70, 2011.

[35] J. Niu-Y. Xia Z. Feng L. Wu, X. Huang and Y. Zhou. Fdu at trec2002: Filtering, qa, web and video tasks. In *11th Text Retrieval Conference*, 2002.

[36] H.C. Liu and G.L. Zick. Automatic determination of scene changes in mpeg compressed video. pages 764–767, 1995.

[37] S. Mao-Y. Xiao I. Chlamtac M. Chen, V. Leung. Hybrid geographical routing for flexible energy-delay trade-offs. *IEEE Trans. Veh. Technol*, 58(9):4976–4988, 2009.

[38] A. B. McDonald. Survey of adaptive shortest-path routing in dynamic packet-switched networks. *Technical report at the Dept of Information Science and Telecommunications*, April 1997.

[39] G. V. Moustakides. Optimal stopping time for detecting changes in distributions. *Ann. Statist.*, 14(4):1379–1387, 1986.

[40] E. S. Page. Continuous inspection schemes. *Biometrica*, 41:100–115, 1954.

[41] E. S. Page. Procedures for reacting to a change in distribution. *Ann. Math. Statist.*, 42(6):1897–1908, 1971.

[42] K. Panagidi, , C. Anagnostopoulos, T. Chalvatzaras, and S. Hadjiefthymiades. To transmit or not to transmit: Controlling communications in the mobile iot domain. *ACM Transactions on Internet Technology (TOIT) - In press*, 2019.

[43] K. Panagidi, C. Anagnostopoulos, and S. Hadjiefthymiades. Optimal grouping-of-pictures in iot video streams. *Computer Communications*, 118:185–194, March 2018.

[44] K. Panagidi, I. Galanis, C. Anagnostopoulos, and S. Hadjiefthymiades. Time-optimized contextual information flow on unmanned vehicles. In *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 185–191, Oct 2018.

[45] K. Panagidi, K. Gerakos, and S. Hadjiefthymiades. Time-optimized prioritization of kafka message scheduling for unmanned vehicles. In *21st IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM 2020)*, volume under submission, June 2020.

[46] L. Blazevic S. Giordano, I. Stojmenovic. Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. *IEEE Communications Surveys and Tutorials*, 15(2):654–677, 2013.

[47] J. Holler S. Lefevre and N. Vincent. A review of real-time segmentation of uncompressed video sequences for content- based search and retrieval. *Real-Time Imaging*, 9:73–98, 2003.

[48] Rviz ROS Software. Open source. `http://wiki.ros.org/rviz`, Accessed: 2019-06-29.

[49] Robot Operating System. Open source. `https://www.ros.org/`, Accessed: 2019-06-29.

[50] Y. Wang, W. Peng, and Y. Tseng. Energy-balanced dispatch of mobile sensors in a hybrid wireless sensor network. *IEEE Transactions on Parallel and Distributed Systems*, 21(12):1836–1850, Dec 2010.

[51] Mark Weiser. The computer for the 21st century. *IEEE Pervasive Computing*, 1:19–25, 1999.

[52] B. Yeo and B. Liu. Novel error concealment method with adaptive prediction to the abrupt and gradual scene changes. *IEEE Trans. Multimedia*, 6(1):158–173, 2004.

[53] B. Yeo and B. Liu. Rapid scene analysis on compressed videodynamic vision. *IEEE Trans. Circuits and Systems for Video Technology*, 5(6):533–544, Dec. 1995.

[54] Y. Zhai and M. Shah. Video scene segmentation using markov chain monte carlo. *IEEE Trans. Multimedia*, 8(4):686–697, 2006.