RADA NAUKOWA DYSCYPLINY
INFORMATYKA TECHNICZNA I TELEKOMUNIKACJA POLITECHNIKI WARSZAWSKIEJ

zaprasza na

OBRONĘ ROZPRAWY DOKTORSKIEJ

mgr. inż. Grzegorz Piotr PANEK

która odbędzie się w dniu **18 listopada 2024 roku,** o godzinie 14**:00** w trybie stacjonarnym

Temat rozprawy:

,,Application relocation in an Edge-enabled 5G-system"

Promotor: dr hab. inż. Halina Tarasiuk – Politechnika Warszawska

Recenzenci: dr hab. inż. Róża Goścień - prof. uczelni – Politechnika Wrocławska

dr hab. inż. Krzysztof Grochla – prof. uczelni – Instytut Informatyki Teoretycznej i Stosowanej PAN

dr hab. inż. Jacek Rak, prof. uczelni – Politechnika Gdańska

Obrona odbędzie się w Audytorium Centralnym Wydziału Elektroniki i Technik Informacyjnych PW. Osoby zainteresowane uczestnictwem w obronie proszone są o zgłoszenie chęci uczestnictwa w formie elektronicznej na adres sekretarza komisji: dr hab. inż. Marcin Kowalczyk , email: marcin.kowalczyk@pw.edu.pl do dnia 14.11 godz. 16:00.

Z rozprawą doktorską i recenzjami można zapoznać się w Czytelni Biblioteki Głównej Politechniki Warszawskiej, Warszawa, Plac Politechniki 1.

Streszczenie rozprawy doktorskiej i recenzje są zamieszczone na stronie internetowej: https://www.bip.pw.edu.pl/Postepowania-w-sprawie-nadania-stopnia-naukowego/Doktoraty/Wszczete-po-30-kwietnia-2019-r/Rada-Naukowa-Dyscypliny-Informatyka-Techniczna-i-Telekomunikacja/mgr-inz.-Grzegorz-Panek

Przewodniczący Rady Naukowej Dyscypliny
Informatyka Techniczna i Telekomunikacja
Politechniki Warszawskiej
prof. dr hab. inż. Jarosław Arabas

Relokacja aplikacji w systemie 5G z przetwarzaniem brzegowym

STRESZCZENIE

Rozwój sieci mobilnych oraz związane z nim możliwości realizacji nowych, zaawansowanych usług, otwierają nowe obszary badawcze w zakresie projektowania systemów sieciowych, które umożliwiają realizację trzech głównych założeń sieci 5G, tj. bardzo duża szybkość bitowa, bardzo małe opóźnienia oraz masowa łączność pomiędzy urządzeniami. W szczególność, dynamiczny rozwój usług wymagających bardzo małych opóźnień komunikacyjnych, sprawił że przetwarzanie na brzegu sieci (ang. Edge Computing) stało się kluczowym rozwiązaniem w trwającej transformacji sieci. Integracja systemu przetwarzania danych na brzegu sieci z architekturą sieci 5G stawia nowe wyzwania takie jak np. efektywne zarządzanie cyklem życia aplikacji brzegowych. Celem rozprawy było zaprojektowanie systemu oferującego wysoką jakość usług (ang. Quality of Service) realizowanych na brzegu sieci, jednocześnie zapewniając ciągłość świadczenia usługi w przypadku mobilności użytkownika końcowego.

W rozprawie skupiono się na procesie relokacji aplikacji brzegowych, który polega na przeniesieniu aplikacji pomiędzy serwerami brzegowymi, aby zagwarantować ciągłość świadczenia usługi podczas mobilności użytkownika. Zaproponowane zostało rozwiązanie dla sieci 5G bazujące na przetwarzaniu w chmurze oraz zgodne ze standardami organizacji ETSI i 3GPP. Przedstawiono trzy główne obszary badań prowadzonych w ramach tej rozprawy.

Początkowo zaprezentowano oryginalne rozwiązanie, demonstrujące procedurę relokacji aplikacji brzegowej w środowisku 5G wzbogaconym o funkcjonalność przetwarzania na brzegu sieci. Zbudowany w tym celu demonstrator umożliwia przetestowanie w praktyce mechanizmu zapewniania ciągłości świadczenia usług w przypadku mobilności użytkownika końcowego lub problemów z infrastrukturą serwerów brzegowych. Wdrożenie procedury relokacji obejmuje wykorzystanie i rozszerzenie prekomercyjnych rozwiązań o otwartym kodzie źródłowym: Kubernetes jako narzędzie uruchomieniowe aplikacji brzegowych w środowisku chmurowym, oraz EMCO jako orkiestrator i zarządca aplikacji brzegowych.

Następnie zaproponowane zostały dwa oryginalne algorytmy mające na celu wybranie odpowiedniego serwera brzegowego do relokacji aplikacji w przypadku mobilności użytkownika końcowego. Pierwszy z algorytmów reprezentuje podejście heurystyczne, które dzieli topologię serwerów brzegowych na podzbiory, a następnie analizuje je aż do momentu znalezienia odpowiedniego serwera. Natomiast drugi algorytm został opracowany z wykorzystaniem metody uczenia maszynowego ze wzmocnieniem (ang. Reinforcement Learning) do trenowania modelu decyzyjnego. Przeprowadzono oraz przestawiono szczegółowe porównanie zaproponowanych algorytmów relokacji aplikacji, pokazując mocne i słabe strony obu rozwiązań, a także dostarczając szereg spostrzeżeń do wykorzystania przez operatorów telekomunikacyjnych rozważających wdrożenie takich rozwiązań do systemów 5G z przetwarzaniem na brzegu sieci.

Słowa kluczowe: *przetwarzanie na brzegu sieci, sieć mobilna 5G, infrastruktura chmurowa, orkiestracja aplikacji brzegowych, algorytm heurystyczny, algorytm uczenia maszynowego przez wzmacnianie*

Recenzja rozprawy doktorskiej

Imię i nazwisko kandydata: Grzegorz Piotr Panek

Tytuł rozprawy doktorskiej:

Application relocation in an Edge-enabled 5G-system

Promotor: dr hab. inż. Halina Tarasiuk

1. Wybór tematu pracy

Rozprawa dotyczy bardzo aktualnego problemu badawczego, jakim jest wybór obliczeniowych węzłów brzegowych, na których mają zostać uruchomione aplikacje działające w modelu edge computing. Problem wynika z rozwoju współczesnych sieci komórkowych oraz zapewnienia przez architekturę systemów 5G możliwości przetwarzania danych użytkownika nie tylko w chmurze obliczeniowej, ale także w węzłach położonych w bezpośredniej bliskości stacji bazowych, w celu minimalizacji czasu transmisji danych. Doktorant skupił się w pracy na wybranym zagadnieniu związanym z zarządzaniem aplikacjami brzegowymi - procesie relokacji, a więc przeniesieniu pomiędzy serwerami w celu zapewnienia ciągłości przetwarzania, gdy użytkownik się przemieszcza.

Temat podjęty w pracy doktorskiej jest aktualny i wynika bezpośrednio z rozszerzeń standardu 5G zaproponowanych przez organizację 3GPP i ETSI. Model przetwarzania brzegowego jest nowym wyzwaniem, z którym muszą się zmierzyć operatorzy sieci 5G. W chwili obecnej jest to tematyka, która jest intensywnie badana przez naukowców na świecie, a zagadnienia przetwarzania brzegowego w sieciach komórkowych są przedmiotem wielu projektów naukowych o zasięgu Europejskim i światowym. Badania te są bardzo istotne w celu umożliwienia użytkownikom sieci komórkowych korzystania z aplikacji wymagających niskiego opóźnienia pakietów, jak np. pojazdy autonomiczne, rzeczywistość rozszerzona (AR), czy operacje na odległość. Mają również duże znaczenie ze względu na skalę potencjalnego zastosowania – opracowane algorytmy znajdą zastosowanie w zarządzaniu serwerami obsługującymi aplikacje milionów użytkowników. Tematyka pracy została bardzo dobrze ulokowana w kontekście rozszerzeń standardu 5G, w którym opracowano architekturę i zasady migracji aplikacji pomiędzy serwerami brzegowymi, ale nie określono algorytmów decydujących o sposobie migracji, a więc rozwiązań, których dotyczy praca. Świadczy to, iż doktorat dotyczy bardzo aktualnego i istotnego problemu badawczego, mającego także duże zastosowanie praktyczne.

2. Ocena układu rozprawy

Rozprawa składa się z 9 rozdziałów oraz bibliografii. W pierwszym rozdziale doktorant przedstawił wstęp i uzasadnienie wyboru tematyki badawczej. W drugim zawarł wstęp do technologii 5G oraz standardów definiujących działanie sieci komórkowych. W rozdziale 3 opisał wyzwania związane z zastosowaniem modelu przetwarzania brzegowego. W rozdziale 4 zawarł przegląd literatury. W kolejnym rozdziale omówił demonstrator możliwości realizacji relokacji aplikacji, a w rozdziale 6 w sposób formalny zdefiniował problem badawczy. Najważniejsza część wyników pracy została zawarta w rozdziałach 7 i 8, w których doktorant zaprezentował 2 opracowane algorytmy: heurystyczny i oparty o uczenie maszynowe wraz z wynikami badania ich efektywności. Pracę zamyka zwięzłe podsumowanie w rozdziale 9.

Podział rozprawy na rozdziały jest klarowny i bardzo dobrze prowadzi czytelnika od zdefiniowania problemu, poprzez opis stanu wiedzy i opis środowiska badawczego, do propozycji nowych algorytmów opracowanych przez autora i analizy ich działania. Praca jest napisana w sposób czytelny. Należy zwrócić uwagę na bardzo dogłębną analizę i precyzyjny opis sposobu realizacji przetwarzania brzegowego w sieciach 5G. Doktorant także bardzo klarownie opisał opracowane algorytmy, definiując każdy z nich w postaci pseudokodu. Także wyniki badań własnych doktoranta są jasno opisane. Przyjęta metoda organizacji pracy w sposób jednoznaczny pozwala na wyróżnienie wkładu własnego doktoranta od opisu stanu wiedzy. Ostatni rozdział dobrze podsumowuje uzyskanie wyniki i prezentuje je w kontekście badań w obszarze systemów przetwarzania brzegowego.

3. Metodologia badawcza

Doktorant w ramach realizacji pracy doktorskiej przygotował środowisko emulacyjne, złożone z serwerów maszyn wirtualnych i środowiska kubernetes pozwalającego odzwierciedlić działanie serwerów brzegowych sieci, kontrolera rdzenia sieci free5GC i symulatora części radiowej sieci 5G UERANSIM. Przygotowanie i konfiguracja tego środowiska wymagały wiele pracy, a jego działanie pozwala zweryfikować działanie narzędzi do sterowania serwerami brzegowymi i relokacji aplikacji w środowisku maksymalnie zbliżonym do docelowej sieci 5G. Środowisko to zostało wykorzystane do udowodnienia poprawności koncepcji relokacji aplikacji brzegowych, jednak w rozprawie nie zawarto pomiarów lub analiz wydajności działania takiej relokacji nawet dla prostych przypadków, co jest pewną wadą rozprawy.

Metodologia badawcza oceny wydajności opracowanych algorytmów koncentruje się na analizie efektywności działania zaproponowanych algorytmów przenoszenia aplikacji za pomocą symulacji sieci komputerowych. Doktorant przygotował środowisko, które nazwał „Edge-Enabled 5G network simulator", składające się z symulatora zachowania użytkownika końcowego odzwierciedlającego mobilność klientów, modułu sterowania działaniem serwerów brzegowych „Edge Orchestrator" oraz emulatora sieci 5G i topologii sieci „Network and Edge Topology". Doktorant poprawnie odzwierciedlił w modelu symulacyjnym architekturę badanej sieci oraz metody przenoszenia aplikacji pomiędzy serwerami brzegowymi. Z dużą starannością także opracował model opóźnień pomiędzy transmisjami na poziomie miasta, regionu i sieci międzynarodowej. Problem do rozwiązania został poprawnie opisany za pomocą notacji matematycznej, a przepływ komunikacji pomiędzy poszczególnymi elementami architektury został bardzo szczegółowo opisany za pomocą diagramów przepływu. Badania za pomocą modelu symulacyjnego zostały wykonane poprawnie pod względem metodologicznym. Doktorant dobrze zaplanował eksperymenty symulacyjne, dla każdego punktu pomiarowego wykreślając średnią ze 100 uruchomień symulacji i odznaczając na wykresie przedziały ufności, co pozwala oszacować zakres błędów. W rozprawie nie opisano wystarczająco precyzyjnie w jaki sposób dokonano walidacji środowiska symulacyjnego np. poprzez porównanie z pomiarami wykonanymi w środowisku emulacyjnym opisanym w rozdziale 5, jednak wyniki przedstawione w rozprawie nie wskazują na błędy w jego działaniu.

Istotnym elementem badań w pracy doktorskiej jest zastosowanie uczenia maszynowego do sterowania działaniem aplikacji w sieci brzegowej. Doktorant dobrze dobrał metodę uczenia ze wzmocnieniem, tworząc model oparty na algorytmie Proximal Policy Optimization (PPO). Wybór tego algorytmu jest adekwatny do problemu rozwiązywanego w rozprawie i doktorant rzetelnie przeanalizował sposób doboru hiper-parametrów. Metodologia doboru zbiorów uczących i sposobu uczenia poprzez wykorzystane przez doktoranta wskaźniki KPI jest także metodologicznie poprawna.

4. Analiza źródeł i stanu wiedzy

Bibliografia rozprawy obejmuje odwołania do 107 artykułów naukowych, książek oraz standardów sieci 5G. Są to prace opisujące tło prowadzonych badań, w tym podstawowe koncepcje związane z architekturą sieci 5G, modelowaniem tych sieci oraz analizą efektywności sieci bezprzewodowych. Doktorant w rozdziałach 2 i 3 bardzo szczegółowo opisał sposób realizacji przetwarzania w modelu „edge” w ramach standardów 5G, odwołując się do dobrze dobranych prac w dziedzinie. Również tło w zakresie badań nad systemami przetwarzania brzegowego zostało dobrze oddane w rozprawie, od odniesienia się do nowych przypadków użycia dla sieci 5G proponowanych przez organizacje standaryzacyjne, po przegląd prac związanych z zastosowaniem optymalizacji do sterowania przetwarzaniem w systemach brzegowych. Cytowane prace są związane z tematem rozprawy, a odwołania zostały umieszczone adekwatne do treści pracy. Przegląd literatury został opisany w sposób wyczerpujący i świadczy, że autor dogłębnie przeanalizował stan wiedzy i przed przystąpieniem do tworzenia własnych algorytmów dobrze zapoznał się z informacjami na temat opisanych wcześniej metod optymalizacji i sterowania działaniem aplikacji na serwerach brzegowych sieci bezprzewodowych.

5. Poprawność redakcyjna rozprawy

Rozprawa została przygotowana w sposób bardzo staranny i nie zawiera znaczących błędów językowych lub redakcyjnych. Rozprawa jest napisana w sposób czytelny i zrozumiały. Sposób organizacji treści w pracy nie budzi zastrzeżeń.
Za niewielką wadę redakcyjną rozprawy można jedynie uznać sposób przygotowania części z rysunków, w których umieszczono część z opisów w sposób nieczytelny lub mało zrozumiały. Np. diagram przepływu wiadomości na rycinie 4.1 zawiera nazwy (np. „4. Nudr_DM_Notify”) które nie zostały omówione w prace i pochodzą wprost ze standardu, z którego został zaczerpnięty, a jego zrozumienie wymaga sięgnięcia do dokumentu źródłowego. Opis części obiektów na rysunku 8.3 jest tak mały, że odczytanie liter jest możliwe jedynie przy użyciu lupy. W/w drobne wady jednak nie umniejszają wysokiego poziomu redakcyjnego całości rozprawy.

6. Wartość naukowa rozprawy

Praca dotyczy bardzo aktualnej tematyki naukowej i stanowi istotny wkład w rozwiązanie problemu efektywnego doboru węzłów brzegowych do realizacji przetwarzania w modelu Edge Computing oraz opracowania algorytmów sterujących migracją aplikacji. Doktorant dobrze zidentyfikował niszę w prowadzonych badaniach nad systemami przetwarzania brzegowego. Koncentrując się na algorytmach sterujących relokacją aplikacji podjął badania o nowatorskim charakterze, w obszarze, w którym liczba dostępnych metod opisanych w literaturze jest niewielka. Praca doktorska nie jest prostą adaptacją znanych rozwiązań do nieznacznie zmodyfikowanego problemu, lecz stanowi odpowiedź na realną potrzebę badawczą. Analiza literatury naukowej zawarta w pracy potwierdza, że doktorat stanowi nowe rozwiązania problemu naukowego.

Doktorant w rozprawie przedstawił 2 metody rozwiązania problemu relokacji aplikacji brzegowych w odpowiedzi na mobilność użytkowników: klasyczny algorytm heurystyczny (zaimplementowany w kilku wersjach) oraz metodę opartą na uczeniu ze wzmocnieniem. Potwierdza to, że doktorant dysponuje odpowiednim warsztatem naukowym i jest w stanie opracować algorytmy o różnym charakterze. Analiza wydajności dla wszystkich algorytmów została przeprowadzona bardzo rzetelnie. Doktorant przeprowadził badania dla różnych parametrów związanych z opóźnieniem transmisji i dla różnych wielkości sieci oraz ocenił skalowalność opracowanych metod w zależności od liczby aplikacji. W doktoracie także oszacowano złożoność obliczeniową (czas wykonania) poszczególnych algorytmów, co pozwala oszacować wpływ realizacji obliczeń na wydłużenie procesu relokacji. Wszystkie te elementy świadczą o dużej rzetelności prowadzonych badań oraz wiarygodności przedstawionych w rozprawie wyników.

Prace przedstawione w rozprawie doktorskiej były elementem 3 publikacji naukowych, w tym jednej opublikowanej w renomowanym czasopiśmie IEEE Communication Magazine oraz 2 artykułach opublikowanych w materiałach konferencyjnych uznanych konferencji międzynarodowych: IEEE International Conference on Communications (ICC) i IEEE Global Communications Conference (Globecom). Świadczy to o wysokiej jakości prowadzonych badań.

7. Możliwość praktycznego zastosowania wyników badań

Wyniki badań zrealizowanych w ramach pracy doktorskiej mają bezpośrednie zastosowanie w praktycznym zarządzaniu sieciami komórkowymi 5 generacji. Istotny aspekt praktyczny mają algorytmy relokacji aplikacji w systemie przetwarzania brzegowego, które można wdrożyć w systemach zarządzania węzłami brzegowymi. Wyniki analizy porównawczej algorytmów w wersjach kładących nacisk na opóźnienie, równoważenie obciążenia lub metod opartych o uczenie maszynowe pozwalają dopasować algorytm do specyfiki zastosowania danej sieci i dobrać odpowiednią metodę pod kątem minimalizacji prawdopodobieństwa odmowy przełączenia aplikacji lub pod kątem minimalizacji częstości przełączeń. Bardzo istotny aspekt praktyczny ma także opracowane przez autora i opisane w rozdziale 5 rozprawy środowisko „proof of concept”, w którym można uruchomić i przetestować w warunkach laboratoryjnych działanie metody przenoszenia aplikacji pomiędzy serwerami brzegowymi. Operator sieci 5G może wykorzystać opisaną w pracy metodę budowania środowiska emulacyjnego do weryfikacji i przetestowania funkcji przetwarzania brzegowego przed ich wdrożeniem w całej sieci.

Możliwość praktycznego zastosowania wyników badań została potwierdzona licznymi prezentacjami wyników pracy doktoranta na konferencjach branżowych, m.in.: KubeCon 2022 w Valencii, "Telco at Edge Days" podczas konferencji KubeCon 2023 w Amsterdamie oraz podczas Orange Open Tech Days w Paryżu w 2023 roku. Należy zwrócić uwagę, że doktorat został opracowany we współpracy z jednym z największych operatorów sieci komórkowych w Polsce – firmą Orange, a jego tematyka jest bezpośrednio powiązana z nowymi usługami rozwijanymi w sieciach komórkowych przyszłości. Dlatego istnieje bardzo duża szansa, że wyniki prac doktoranta zostaną w praktyce zastosowane w sieciach obsługujących miliony użytkowników.

8. Uwagi krytyczne

Istotna część badań efektywności algorytmów przeniesienia aplikacji pomiędzy serwerami brzegowymi zależy od przyjętego sposobu odwzorowania mobilności użytkowników. W środowisku symulacyjnym opisanym w rozdziale 6 rozprawy wskazano, że zaimplementowano jedynie prosty model mobilności użytkowników zakładający, że mogą oni przemieszczać się pomiędzy sąsiednimi komórkami z równomiernym prawdopodobieństwem. Może to prowadzić do nierównomiernego obciążenia komórek, z większym prawdopodobieństwem przełączenia użytkownika do komórek znajdujących się w centrum sieci, podobnie jak ma to miejsce w modelu „Random Waypoint”. Czy zweryfikowano równomierność obciążenia sieci i czy ma to wpływ na wyniki przeprowadzonej analizy np. prawdopodobieństwa odmowy przełączenia aplikacji?

Doktorant używa modelu symulacyjnego w znacznej mierze bazującego na opracowanych samodzielnie elementach. W jaki sposób zweryfikowano poprawność działania symulatora i czy spróbowano porównać działanie modelu symulacyjnego dla prostych topologii z działaniem środowiska „proof of concept” opisanego w rozdziale 5?

W przeprowadzonych analizach skoncentrowano się na algorytmach dokonujących próby przełączenia jedynie pojedynczej aplikacji, a parametrem użytym do oceny jakości działania algorytmu jest współczynnik odrzuconych prób przeniesienia (ang. relocation rejection rate). Naturalną odpowiedzią na ten problem wydaje się przeniesienie innych aplikacji w celu zwolnienia miejsca na serwerach brzegowych położonych w miejscu docelowym, lub równoważenie obciążenia poprzez przełączenia aplikacji realizowane nie tylko w odpowiedzi na mobilność użytkowników, ale także w sposób proaktywny, np. poprzez okresowe uruchomienie w celu przeniesienia aplikacji możliwych do przeniesienia do serwerów o niższym obciążeniu. Dlaczego w planach dalszych prac w rozdziale 9.2 skoncentrowano się na praktycznych aspektach (np. wsparcie migracji aplikacji stanowych), a pominięto aspekt możliwej poprawy metody zarządzania migracją aplikacji poprzez w/w rozszerzenia?

9. Podsumowanie i ocena końcowa

Praca doktorska pt. „Application relocation in an Edge-enabled 5G-system" została przygotowana przez Pana magistra Grzegorza Piotra Panek rzetelnie i wykazuje zdolność kandydata do prowadzenia pracy naukowej w sposób samodzielny. Praca jest poprawna pod względem metodologicznym, a opracowane algorytmy relokacji aplikacji pomiędzy węzłami brzegowymi w odpowiedzi na mobilność użytkowników są nowatorskie i zostały rzetelnie przeanalizowane. Badania w zakresie zastosowania uczenia maszynowego ze wzmocnieniem do optymalizacji przełączeń aplikacji w systemie przetwarzania brzegowego sieci 5G wykraczają poza stan wiedzy, ale mają także bezpośrednie zastosowanie praktyczne. Badania zostały poprawnie zaplanowane i zrealizowane, a ich wyniki szczegółowo i precyzyjnie opisane. Praca tym samym potwierdza, iż kandydat posiada wymaganą wiedzę teoretyczną w dyscyplinie Informatyka Techniczna i Telekomunikacja.

Stwierdzam, że recenzowana rozprawa doktorska Pana Grzegorza Panek spełnia warunki określone w art. 187 ust. 1 i 2 Ustawy z dnia 20 lipca 2018r. Prawo o szkolnictwie wyższym i nauce (Dz.U. 2018 r., poz. 1668 z późn. zmianami) i wnioskuję do Rady Dyscypliny Informatyka Techniczna i Telekomunikacja Politechniki Warszawskiej o dopuszczenie Pana Magistra Grzegorza Panka do dalszych etapów przewodu doktorskiego.

Krzysztof Grochla; IITiS PAN
2024.09.25 12:59:00 +02'00'

25.09.2024
....
data sporządzenia recenzji

. ...
podpis recenzenta

POLITECHNIKA
GDAŃSKA

WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI I INFORMATYKI

Gdańsk, 29.08.2024 r.

Dr hab. inż. Jacek Rak, prof. PG
Wydział Elektroniki, Telekomunikacji i Informatyki
Politechnika Gdańska

Recenzja rozprawy doktorskiej

Tytuł: **Application Relocation in an Edge-Enabled 5G-System**

Autor: **Mgr inż. Grzegorz Piotr Panek**

Promotor: **Dr hab. inż. Halina Tarasiuk**

Rozprawa doktorska mgra inż. Grzegorza Panka dotyczy zagadnienia relokacji aplikacji brzegowych (ang. edge applications), tj. zmiany lokalizacji instancji aplikacji pomiędzy węzłami systemu brzegowego (ang. edge hosts) w celu spełnienia silnych wymagań jakościowych oraz nieprzerwanego świadczenia usług dla mobilnego użytkownika systemu 5G. Rozpatruje ona zatem łącznie kwestie jakości usług oraz niezawodności transmisji, koncentrując się w szczególności na mechanizmach nieprzerwanej dostępności usług (ang. uninterrupted service). Zagadnienia te są istotne oraz aktualne w odniesieniu do systemów 5G, w stosunku do których sformułowane wymagania w zakresie przepustowości, opóźnień transmisji oraz generalnie dostępności usług są wysokie. Stopień trudności analizowanego zagadnienia podnosi założenie dotyczące mobilności użytkowników końcowych.

Rozprawa doktorska mgra inż. Grzegorza Panka została napisana w języku angielskim. Struktura pracy jest ogólnie poprawna i obejmuje dziewięć rozdziałów, kolejno: wprowadzenie (Rozdział 1) ukazujące motywację pracy, opis najważniejszych osiągnięć oraz organizację rozprawy; omówienie w Rozdziale 2 kluczowych dla rozprawy zagadnień związanych z systemami 5G; dyskusję rozwiązań dotyczących przetwarzania na brzegu sieci (ang. edge computing) – Rozdział 3; przegląd literatury w zakresie istniejących metod rozwiązania problemu relokacji aplikacji (Rozdział 4); szczegółowy opis rozważanej w rozprawie architektury systemu 5G umożliwiającej relokację aplikacji (Rozdział 5); formalne zdefiniowanie problemu relokacji aplikacji (Rozdział 6); opis proponowanego algorytmu heurystycznego wyboru węzła docelowego relokacji aplikacji (Rozdział 7); opis autorskiego algorytmu relokacji wykorzystującego koncepcję uczenia maszynowego (Rozdział 8); podsumowanie pracy (Rozdział 9) oraz trzy części nienumerowane: bibliografia, wykaz rysunków i wykaz tabel. Bibliografia obejmuje 107 adekwatnie dobranych publikacji.

Należy podkreślić, że oceniana rozprawa jest wynikiem realizacji tzw. doktoratu wdrożeniowego i jest wynikiem współpracy Orange Innovation Poland z Politechniką Warszawską, a Doktorant jest zatrudniony w w/w firmie na stanowisku inżyniera badawczego. Wyniki pracy Doktoranta zostały zaprezentowane w szczególności w jednym artykule w czasopiśmie IEEE Communications Magazine (200 pkt MNiSW) oraz dwóch artykułach (po 70 pkt MNiSW każdy) opublikowanych w materiałach konferencji IEEE ICC 2023 w Rzymie oraz IEEE Globecom 2023 w Kuala Lumpur. Doktorant zaprezentował wyniki swoich prac również na innych konferencjach branżowych z ramienia Orange, m.in. KubeCon 2022 w Walencji.

1. Jaki jest cel naukowy rozprawy i czy został on trafnie i jasno sformułowany?

Celem pracy Doktoranta było opracowanie efektywnych mechanizmów relokacji aplikacji w zintegrowanej architekturze obejmującej system 5G oraz rozwiązania mobile edge computing (MEC). Cel ten jest jasno naświetlony i umotywowany w pierwszych czterech rozdziałach rozprawy, mimo że rozprawa nie zawiera bezpośrednio sformułowanej tezy.

W mojej opinii cel ten jest niewątpliwie trafnie sformułowany z uwagi na duży stopień mobilności użytkowników systemu 5G decydujący o dynamicznym charakterze architektury systemu oraz implikujący potrzebę wdrożenia mechanizmów elastycznej konfiguracji w celu podtrzymania ciągłości dostępności usług przy zadanych charakterystykach QoS.

Zakres zrealizowanych przez Doktoranta prac opisanych w rozprawie obejmuje opracowanie algorytmów relokacji aplikacji, projekt i implementację demonstratora, jak i weryfikację charakterystyk właściwości proponowanych rozwiązań. Zakres ten jest w ogólności właściwy. Również poziom trudności powyższych prac należy uznać za adekwatny dla badań naukowych na poziomie doktorskim.

2. Na czym polega oryginalny dorobek Autora i jakie jest znaczenie poznawcze lub przydatność praktyczna dla nauki bądź techniki?

Do oryginalnego dorobku Autora można moim zdaniem zaliczyć:

1) projekt zgodnej z założeniami ETSI i 3GPP architektury systemu 5G uwzględniającej mechanizmy edge computing i pozwalającej na relokację aplikacji brzegowych oraz opis demonstratora (ang. proof-of-concept – PoC) wykorzystującego m.in. platformę Kubernetes zaprezentowany w rozdziale 5 rozprawy;
2) zdefiniowanie problemu relokacji aplikacji w rozdziale 6 rozprawy;
3) propozycję heurystycznego algorytmu EAR-Heuristic relokacji aplikacji brzegowych uwzględniającego wymagania aplikacji oraz zgodnego ze strategią równoważenia obciążenia opisanego w rozdziale 7 rozprawy;
4) propozycję algorytmu wyboru węzła docelowego relokacji aplikacji wykorzystującego strategię uczenia maszynowego do rozwiązania postawionego problemu opisanego w rozdziale 8 rozprawy,

a także:

5) zaprezentowany w rozdziale 2 rozprawy opis najważniejszych elementów składowych architektury systemu 5G; zobrazowanie scenariuszy, w których przetwarzanie na brzegu sieci odgrywa kluczową rolę dla użytkowników mobilnych; wyjaśnienie zasad integracji technologii MEC z systemem 5G; szczegółowa prezentacja motywów (w tym mobilność użytkowników, degradacja poziomu QoS, problemy serwera MEC w zakresie dalszego wspierania aplikacji) oraz scenariuszy (np. pojazdy autonomiczne, drony czy strumieniowanie ruchu wideo), w których koncepcja relokacji aplikacji ma kardynalne znaczenie w podtrzymywaniu ciągłości świadczenia usług przy zadanych parametrach QoS;

6) zawarta w rozdziale 4 analiza rozwiązań dostępnych w literaturze dotyczących problemu relokacji aplikacji.

Dorobek zaprezentowany w ocenianej rozprawie ma istotne znaczenie praktyczne poparte realnym zaangażowaniem Doktoranta we współpracę z jednostką otoczenia gospodarczego (w tym przypadku Orange Innovation Poland) podczas realizacji zadań rozprawy.

Problem rozpatrywany w rozprawie przez Doktoranta należy uznać za rozwiązany. Sam sposób rozwiązania postawionego problemu także nie budzi większych zastrzeżeń. Zastosowana przez Doktoranta metodologia jest adekwatna dla badań naukowych na poziomie doktorskim.

3. Jakie są słabsze strony rozprawy?

Uwagi natury ogólnej

a) Praca została napisana ogólnie dobrze pod względem językowym (praca w języku angielskim). Jednakże czasami występują w niej pewne niedociągnięcia językowe, jak np.:

- pisownia rozłączna wyrażenia „three fold" zamiast typowej łącznej ("three-fold" lub „threefold") – str. 7 rozprawy;
- niewłaściwe użycie sformułowania „proceedings" w odniesieniu do czasopisma IEEE Communications Magazine na str. 17;
- nadmiarowe użycie rodzajnika „the" np. przed odwołaniami do rozdziałów (przykładowo, zamiast „the chapter 4" na str. 17 powinno być „chapter 4");
- „dedicated for" → „dedicated to" na str. 18;
- "represents a notable improvements" → usunąć "a" na str. 19;
- "The main MEO responsibility is to ...maintain, ...select, ...trigger..." (a nie "maintaining", "triggering" na str. 22-23);
- "Indeed, its reduce the number..." → "Indeed, it reduces the number..." (na str. 89).

b) Dla skrótu "UE" używanego np. na str. 24 pełna nazwa jest podana dopiero na str. 28 rozprawy.

c) W pracy bardzo często występują określenia pisane z wielkie litery, mimo, że co najmniej w przypadku niektórych z nich wydaje się to nieuzasadnione (np. „Life-Cycle" na str. 28).

d) Zauważalna jest niespójność formatowania pełnych nazw: wyrazy rozpoczynają się raz z wielkiej litery, innym razem z małej (np. „ultra-high definition" – str. 31). Trudno jest wychwycić zasadę, którą Autor się w tym kontekście posługuje.

e) Tytuł rozdziału 2 („Background") jest zbyt krótki i z tego powodu nie odzwierciedla jego zawartości.

f) Zawartość rozdziału 3 dotycząca wyzwań w zakresie wdrażania rozwiązań edge computing stanowi naturalne rozszerzenie treści rozdziału 2. Z tego powodu wydaje się, że lepiej pozycjonowałaby się ona jako sekcja rozdziału 2, a nie jako oddzielny rozdział.

g) Zawartość pól tekstowych rysunku 5.3 jest mało czytelna (czcionka o zbyt małym rozmiarze). W tym kontekście lepszym rozwiązaniem byłoby zaprezentowanie rys. 5.3 w układzie pionowym jako obiektu zajmującego całą stronę.

h) Użycie symbolu *V* w kontekście łączy systemu w rozdziale 6 nie jest fortunne, gdyż typowo symbol *V* jest zarezerwowany dla wierzchołków grafu. W przypadku łączy zwyczajowo stosuje się symbol *E* (od ang. edges) dla łączy dwukierunkowych lub symbol *A* (ang. arcs – skierowane łuki) w przypadku łączy jednokierunkowych.

i) W celu zwiększenia poziomu czytelności modelu optymalizacyjnego z rozdziału 6, wskazane byłoby pogrupowanie opisu składowych modelu w części obejmujące symbole, zmienne, stałe, a dopiero w dalszej kolejności funkcję kryterialną i warunki ograniczające.

j) Sekcja 6.5 zatytułowana „Edge Relocation Simulator" nie pasuje tematycznie do rozdziału 6 omawiającego aspekty modelowania matematycznego dla problemu optymalizacyjnego.

k) W rozdziale 6 w przypadku iloczynu liczb powinien być zastosowany prosty operator · zamiast operatora ×. Ten drugi ma zastosowanie w przypadku iloczynu kartezjańskiego zbiorów.

l) W rozdziale 7 zastosowanie przedrostka „O-" w nazewnictwie wariantów algorytmu oraz sam tytuł rozdziału 7.2.5: „Heuristic and Optimal comparison" mogą być dla czytelnika mylące, gdyż mogą sugerować rozwiązania uzyskane w wyniku rozwiązania zadania optymalizacyjnego z rozdziału 6 (podczas gdy w pracy brakuje jednoznacznej informacji o wykorzystaniu konkretnego narzędzia (tzw. solvera) w tym zakresie oraz, w konsekwencji tego, przeprowadzenia analizy porównawczej).

m) Znaczna część treści zawartej na rysunku 8.3 jest trudna do odczytania z uwagi na bardzo mały rozmiar czcionki.

Uwagi szczegółowe

1. Pomimo, że rozdział 4.1 relatywnie szczegółowo prezentuje najważniejsze cechy rozwiązań dostępnych w literaturze (ang. state-of-the-art – SoTA), pozostawia pewien niedosyt w kontekście szerszej identyfikacji otwartych problemów, czy też analizy stopnia możliwości dalszej poprawy charakterystyk istniejących rozwiązań. Rozszerzenie prezentacji o powyższe aspekty podkreśliłoby lepiej rolę osiągnięć Doktoranta w stosunku do rozwiązań SoTA.
2. W rozprawie brakuje jednoznacznej informacji o tym, czy zdefiniowany w rozdziale 6 model optymalizacyjny (dotyczący problemu określenia najlepszych lokalizacji dla relokacji aplikacji) został (czy też nie został) użyty w dalszej części rozprawy w analizie porównawczej. Wątpliwości w tym zakresie są uzasadnione m.in. brakiem jednoznacznej informacji odnośnie użytego środowiska obliczeniowego (solvera) dla problemu optymalizacyjnego z rozdziału 6 oraz jego konfiguracji. Zasadne jest więc pytanie o komentarz w zakresie stopnia optymalności (ang. optimality gap) w odniesieniu do jakości rezultatów (mierzonych wartością funkcji kryterialnej) uzyskanych w wyniku użycia algorytmów z rozdziałów 7 i 8.
3. Dodatkowe uzasadnienie wydaje się niezbędne w kontekście modelowania w rozprawie wartości opóźnień end-to-end (rozdział 6.5.2) bazując jedynie na wartościach opóźnień na łączach systemu.
4. Analiza efektywności algorytmu heurystycznego omówionego w rozdziale 7 byłaby pełniejsza, gdyby obejmowała ocenę złożoności obliczeniowej rozpatrywanego algorytmu. Pozwoliłoby to np. na dokładniejszą analizę stopnia skalowalności algorytmu.

Powyższe uwagi nie wpływają na moją pozytywną łączną ocenę rozprawy.

4. Do której z następujących kategorii Recenzent zalicza rozprawę:

a/ nie spełniająca wymagań,
b/ wymagająca wprowadzenia poprawek i ponownego recenzowania,
c/ spełniająca wymagania,
d/ wykraczająca ponad poziom zadawalający (spełniająca wymagania z nadmiarem),
e/ wybitna?

Podsumowując, moja recenzja pracy doktorskiej mgra inż. Grzegorza Piotra Panka jest **pozytywna**. Wartość merytoryczna rozprawy jak i sposób prezentacji wyników kwalifikują w mojej ocenie rozprawę mgra inż. Grzegorza Piotra Panka do kategorii **c/ spełniająca wymagania** ustawowe stawiane rozprawom doktorskim.

Z powodów powyższych, wnoszę o dopuszczenie rozprawy doktorskiej mgra inż. Grzegorza Piotra Panka do publicznej obrony.

Jacek Rak

Wrocław, 02.09.2024

Recenzja rozprawy doktorskiej

Recenzentka:

Dr hab. inż. Róża Goścień, prof. PWr

Katedra Systemów i Sieci Komputerowych

Wydział Informatyki i Telekomunikacji

Politechnika Wrocławska

Recenzowana rozprawa:

Tytuł: Application Relocation in an Edge-Enabled 5G-System

Autor: mgr inż. Grzegorz Piotr Panek

Promotor: dr hab. inż. Halina Tarasiuk

1. Zawartość rozprawy

Na przełomie ostatnich kilkunastu lat, sieci teleinformatyczne stały się nieodzownym elementem codziennego życia społeczeństwa. Ułatwiają one, a czasem nawet i umożliwiają, funkcjonowanie tak kluczowych dla nas obszarów jak praca, nauka, finanse, służba zdrowia, rozrywka czy też życie społeczne. Z roku na rok obserwujemy rosnącą liczbę użytkowników sieciowych, podłączonych urządzeń oraz zainteresowanie usługami wymagającymi wysokich przepustowości, niewielkich opóźnień oraz ciągłej dostępności zasobów sieciowych. Co więcej, zauważamy rosnącą popularność sieci mobilnych, które aktualnie są źródłem i celem większości ruchu sieciowego.

Główną cechą tych sieci jest mobilność, która jest kluczowym udogodnieniem dla użytkowników końcowych. Mobilność jest równocześnie wysokim wyzwaniem dla operatorów telekomunikacyjnych, gdyż przynosi ona wiele nowych problemów związanych z zarządzeniem sieciami oraz obsługą użytkowników. Aby sprostać tak dużym wymaganiom, sieci te muszą się bardzo szybko rozwijać, w szczególności w zakresie ich architektur i systemów sterowania. Aktualnie systemy mobilne bazują głównie na technologii piątej generacji, zwanej w skrócie 5G. Aby dodatkowo usprawnić świadczenie wielu usług pożądanych przez użytkowników sieciowych (w szczególności usług wrażliwych na opóźnienia), zaproponowana została architektura z przetwarzaniem brzegowym (ang. edge computing), w której kluczowe obliczenia dla użytkownika końcowego przeniesione zostały znacznie bliżej, czyli na urządzenia brzegowe (z którymi użytkownik może się połączyć bezpośrednio lub za pośrednictwem jedynie kilku urządzeń). Rozwiązanie takie pozwala znacząco zmniejszyć opóźnienia transmisji, co jest kluczowe dla wielu popularnych usług. Aby usprawnić realizację usług wrażliwych na opóźnienia, sieci 5G można zintegrować z technologią przetwarzania brzegowego. Wymaga to jednak dodatkowych prac nad zdefiniowanie szczegółów połączenia oraz współpracy tych dwóch rozwiązań. Dodatkowo, rozwiązanie to przynosi nowy oraz istotny problem relokacji usług, czyli przenoszenia użytkownika wraz ze świadczoną dla niego usługą pomiędzy rożnymi węzłami-serwerami sieci. Kluczowym elementem tego problemu jest wybranie nowego węzła do obsługi klienta (gdy poprzedni węzeł nie jest w stanie świadczyć usług lub nie jest stanie robić tego na odpowiednim poziomie), z uwzględnieniem wymagań realizowanych usług, obciążenia sieci oraz dostępności zasobów sieciowych. Warto zauważyć, iż problem relokacji jest problemem dynamicznym, gdyż stan sieci mobilnej i jej zasobów zmienia się bardzo szybko. Niniejsza rozprawa doktorska porusza dwa wspomniane powyżej wyzwanie – zdefiniowanie architektury oraz sposobu współpracy w sieci 5G implementujące przetwarzanie brzegowe oraz efektywne rozwiązanie problemu relokacji poprzez dedykowane metody rozwiązania. Stąd też, recenzowana rozprawa doktorska wpisuje się w aktualne trendy i potrzeby badawcze. Należy więc zaznaczyć, iż porusza ona tematy aktualne i istotne badawczo.

Rozprawa składa się z 9 numerowanych rozdziałów uzupełnionych (na końcu rozprawy) spisem literatury, rysunków, tabel oraz akronimów. Rozprawa liczy 141 numerowanych stron i napisana jest w języku angielskim. Lista literatury składa się z 107 odpowiednio dobranych pozycji.

Rozdział 1 jest wprowadzeniem do rozprawy. Przedstawia krótką definicję rozważanego problemu wraz z jego motywacją, następnie cel, zakres oraz strukturę dalszej części pracy. Zawiera również skrócony opis głównego dokonania Autora (które jest podstawą rozważanej rozprawy) wraz z opisem jego dorobku publikacyjnego.

Rozdział 2 jest rozdziałem teoretycznym, wprowadzającym czytelnika w tematykę sieci 5G, mobilności w sieciach 5G oraz wynikającej z tego potrzeby relokacji usług pomiędzy węzłami sieci 5G (czyli głównego problemu optymalizacyjnego rozwiązywanego w rozprawie). W rozdziale tym Autor przedstawia również 5 przykładowych scenariuszy obsługiwanych przez sieć 5G, w których występuje konieczność relokacji usług.

Rozdział 3 jest kolejnym rozdziałem teoretycznym, w którym przedstawiana jest architektura sieci z przetwarzaniem brzegowym z uwzględnieniem powiązanych wyzwań oraz otwartych pytań badawczych.

Rozdział 4 stanowi przegląd literatury związanej z tematyką rozprawy. Rozdział ten podzielony został na trzy podrozdziały dotyczące (odpowiednio): (*i*) problemu relokacji usług w sieciach 5G oraz migracji usług w sieciach w architekturze z przetwarzaniem brzegowym, (*ii*) problemu synchronizacji przy relokacji usług w sieciach o architekturze z przetwarzaniem brzegowym, (*iii*) podsumowania. W pierwszej części przeanalizowane zostały metody oparte o modele programowania liniowego, heurystyki oraz metody oparte o uczenie maszynowe.

Rozdziały 5-8 prezentują dokonania Autora. W rozdziale 5 zaproponowana została architektura systemu sieci 5G z przetwarzaniem brzegowym wspierającego dynamiczną relokację usług. Architektura ta została również zaimplementowana w autorskim symulatorze, który został zintegrowany z rzeczywistym oprogramowanie dostępnym dla węzłów sieci 5G. Rozdział 5 przedstawia w skrócie przygotowany symulator. Rozdział 6 przedstawia model matematyczny programowania liniowego dla głównego problemu optymalizacyjnego rozważanego w rozprawie – problem relokacji usług. Rozdział 7 przedstawia opis autorskiego algorytmu heurystycznego rozwiązania problemu relokacji oraz jego porównanie z metodami referencyjnymi. Rozdział 8 przedstawia opis kolejnego algorytmu zaproponowanego przez Autora dla problemu relokacji. Jest to algorytm oparty o uczenie maszynowe, a dokładniej o uczenie ze wzmocnieniem.

Rozdział 9 jest podsumowaniem rozprawy oraz dotychczasowej pracy wykonanej przez Autora.

Zaproponowany układ pracy jest poprawny, w sposób logiczny prezentuje osiągnięcia Autora. Przedstawione rozdziały teoretyczne w sposób jasny i wystarczający przedstawiają tło oraz motywację dla rozważanych problemów oraz architektur sieciowych. Dobrze umotywowane oraz opisane są również dokonania Autora, będące podstawą rozważanej rozprawy.

2. Opinia o rozprawie

Dynamiczny rozwój sieci mobilnych oraz ich rosnąca popularność wśród społeczeństwa skutkują potrzebą opracowania nowych rozwiązań pozwalających świadczyć pożądane prze użytkowników usługi w sposób efektywny. Jednym ze zidentyfikowanych przez Autora problemów kluczowych aktualnie dla sieci mobilnych jest ich integracja z architekturą przetwarzania brzegowego oraz zagadnienie efektywnej relokacji usług świadczonych użytkownikom końcowym pomiędzy węzłami-serwerami znajdującymi się różnych strefach administracyjnych (geograficznych) sieci. Relokacja jest niezbędna aby zapewnić użytkownikom ciągłość świadczenia usług (przy przemieszczaniu się) lub usługę na odpowiednim poziomie (przy zmieniającym się obciążeniu sieci). Problem relokacji w sieciach 5G z przetwarzanie brzegowym jest zagadnieniem nowym w literaturze i wymagane jest opracowanie dla niego dedykowanych metod rozwiązania. Właśnie ta potrzeba stała się podstawą recenzowanej rozprawy doktorskiej, w której Autor skupia się na opracowaniu szczegółów architektury pozwalającej na relokację usług w sieciach 5G z przetwarzaniem brzegowym, a następnie proponuje dedykowane algorytmy rozwiązania tego problemu. Jeden z proponowanych algorytmów wykorzystuje uczenie maszynowe, co dodatkowo odpowiada na aktualne trendy w rozwoju sieci teleinformatycznych – czyli wykorzystanie uczenia maszynowego do optymalizacji ich działania.

W rozważanej rozprawie doktorskiej Autor prezentuje cztery elementy stanowiące jego główne osiągniecie naukowego: (*i*) propozycję architektury sieci 5G z przetwarzaniem brzegowym wspierającej proces relokacji, (*ii*) definicję (wraz z modelem matematycznym) nowego problemu optymalizacyjnego dotyczącego relokacji usług w sieciach 5G z przetwarzaniem brzegowym, (*iii*) algorytm heurystyczny dedykowany do rozwiązania zadanego problemu oraz analizę jego efektywności, (*iv*) algorytm bazujący na uczeniu maszynowym dedykowany do rozwiazania zdefiniowanego problemu oraz analizę jego efektywności.

Pierwszym elementem osiągnięcia jest zaproponowanie architektury sieci mobilnej 5G z przetwarzaniem brzegowym, jej dokładny opis oraz zaproponowanie rozwiązań pozwalających przeprowadzać w tej architekturze operacje relokacji usług. Należy podkreślić, iż jest to główny element współpracy z firmą Orange Innovation Polska, która została zawiązania celem przygotowania doktoratu wdrożeniowego.

W ramach drugiej części osiągnięcia, Autor zdefiniował nowy problem optymalizacyjny – relokacja usługi w sieci 5G z przetwarzaniem brzegowym. Funkcja celu w problemie określona jest jako średnia ważona uwzględniająca opóźnienie realizacji transmisji (klient – serwer), wykorzystanie pamięci oraz procesora przez usługę. Wśród ograniczeń problemu znalazły się warunki kontrolujące dostępność zasobów sieciowych na różnych serwerach oraz uwzględniające pozostawienie części zasobów niewykorzystywanych jako margines bezpieczeństwa przy świadczeniu usług. Problem został w rozprawie zamodelowany za pomocą techniki programowania liniowego.

Trzecia część osiągnięcia to propozycja algorytmu heurystycznego do rozwiazania zagadnienia relokacji. Algorytm bazuje na procedurze zachłannej, która iteracyjnie analizuje kolejnych kandydatów do relokacji (potencjalne węzły, które mogą obsłużyć usługę), wzbogaconej o mechanizm przeszukiwania lokalnego. Dla każdego rozważnego kandydata sprawdzana jest dostępność zasobów oraz powiązana z nim wartość funkcji celu. Zaproponowany algorytm zostaje następnie porównany z czterema metodami referencyjnym biorąc pod uwagę: liczbę żądanych zgłoszeń relokacji, liczbę nieobsłużonych relokacji, średnie wykorzystanie pamięci i procesora w sieci. Porównanie przeprowadzone zostaje dla sieci różnego poziomu (miastowe, regionalne oraz międzynarodowe), różnego typu oraz dla różnej liczby usług. Zaproponowany algorytm wypada na tle wszystkich badanych metod zadowalająco, chociaż nie wykazuje najwyższej jakości dla wszystkich badanych scenariuszy i kryteriów.

Czwarty element osiągniecia Autora to propozycja algorytmu rozwiązania zadania relokacji bazującego na uczeniu maszynowym, a dokładniej na uczeniu ze wzmocnieniem (ang. reinforcement learning). W ramach tego osiągnięcia Autor zaproponował sposób modelowania stanu sieci oraz usługi jak również funkcję oceny akcji podejmowanych przez agenta. Następnie, zaproponowany algorytm został porównany z czterema metodami referencyjnymi oraz zaproponowanym algorytmem heurystycznym. W badaniach wykorzystane zostały te same kryteria porównania oraz scenariusze testowe, co przy analizowaniu efektywności samego algorytmu heurystycznego. Zaproponowany algorytm wypada na tle wszystkich badanych metod zadowalająco, chociaż nie wykazuje najwyższej jakości dla wszystkich badanych scenariuszy i kryteriów.

Podsumowując, osiągnięcie (oraz wszystkie jego składowe) zaproponowane w rozprawie odpowiada na ważny i aktualny problem dotyczący nowoczesnych sieci mobilnych świadczących szereg usług użytkownikom końcowym. Dodatkowo, sposób zaadresowania zagadnienia (czyli wykorzystanie uczenia maszynowego do rozwiazania zidentyfikowanego problemu optymalizacyjnego) wpasowuje się w aktualne trendy rozwoju sieci teleinformatycznych oraz narzędzi optymalizacyjnych. Przedstawione w rozprawie osiągnięcie stanowi więc oryginalne rozwiązania dla problemu integracji sieci 5G z przetwarzaniem brzegowym oraz relokacji usług w sieciach mobilnych 5G z przetwarzaniem brzegowym, stąd spełnia wymogi stawiane rozprawom doktorskim.

3. Uwagi krytyczne i dyskusyjne

Po lektorze pracy nasuwają się następujące uwagi dyskusyjne:

a.) Zaprezentowany model problemu opisany jest w sposób niepoprawny, przez co jest bardzo trudny w interpretacji. Pojedyncze oznaczenia (litery) zmieniają swoje znaczenie (interpretację) na przestrzeni opisu modelu. Nie ma również jednej konwencji powiązania oznaczenia z indeksami – raz używane są przypisy (dolne/górne), innym razem notacja z nawiasami. Nieznane są typy zmiennych oraz zakresy przyjmowanych przez nie wartości. Wszystkie używanego oznaczenia powinny być najpierw wprowadzone z podziałem na sekcje indeksów, stałych oraz zmiennych (wskazując również ich charakter – czyli typ i zakres przyjmowanych wartości). Dzięki takiemu podejściu Autor uniknąłby istniejącej aktualnie kolizji oznaczeń. Model nie uwzględnia również skali czasu, która w sieci mobilnej jest niezwykle istotna (stan sieci i dostępność zasobów zmieniają się bardzo dynamicznie).

b.) Brak analizy złożoności problemu optymalizacyjnego (np. na podstawie przedstawionego modelu matematycznego).

c.) Problem opisany modelem matematycznym oraz problemy rozwiązywane przez algorytmy heurystyczne nie są ze sobą tożsame (mamy przede wszystkim inne funkcje celu). Warto byłoby przedstawić model problemu rozwiązywanego przez algorytmy heurystyczne i porównać ich wyniki z wynikami optymalnymi (dostarczanymi przez metodę dokładną na bazie modelu).

d.) W pracy niepoprawnie używane jest określenie *optymalny*. *Optymalny* znaczy *najlepszy* (w zadanym środowisku) i jest przymiotnikiem niestopniowalnym. Natomiast Autor pisze o metodach (tutaj tłumaczenie z języka angielskiego) mniej i bardziej optymalnych. Dodatkowo, nazywa metody referencyjne metodami optymalnymi, gdzie nie istnieje żaden dowód na to, że te algorytmy są metodami optymalnymi (zwłaszcza, że wykazują często niższą efektywność niż metody zaproponowane przez Autora).

e.) Wątpliwości budzi również opis tuningu (czyli strojenia) metod. Strojeniu podlegają sterowalne parametry samego algorytmu (lub hiperparametry w przypadku algorytmów uczenia maszynowego), nie współczynniki funkcji celu. Tuning parametrów funkcji celu prowadzi to definiowania różnych problemów optymalizacyjnych, a co za tym idzie – do rozwiązywania przez algorytmy innych problemów. Dodatkowo, przy opisie tuningu należy zaznaczyć jakie wartości parametrów były badane (na jakiej podstawie takie zostały wybrane), jak zmieniała się efektywność metody w funkcji zmiany wartości rożnych parametrów oraz jakie wartości zostają rekomendowane do badań docelowych (i na jakiej podstawie).

f.) W pracy brakuje szerszej analizy statystycznej otrzymanych wyników, w szczególności przy porównaniu różnych algorytmów i wnioskowania, który jest lepszy od którego. Do takiej analizy należy wybrać odpowiednie testy statystyczne i sprawdzić czy różnice efektywności są statystycznie istotne.

g.) Wątpliwości budzi również użycie określania *reliability* (i *reliable*) do opisu stworzonego systemu. Określenie to kojarzone jest przede wszystkim z metodami ochrony sieci (jej elementów) przed różnego rodzaju atakami i awariami bądź metodami podnoszącymi wiarygodność systemu. Jednak zaproponowany system nie posiada takich mechanizmów. Określenie *reliability* wprowadza więc czytelnika w błąd.

h.) Przegląd literaturowy dotyczący metod relokacji zdaje się być opisany bardzo pokrótce. Powiązanych prac jest dużo więcej, zwłaszcza że problem relokacji (nie dokładnie w takiej formie, ale podobnej) rozważany jest nie tylko w sieciach 5G.

i.) W pracy doktorskiej brak jest tezy badawczej.

W rozprawie zauważono liczne błędy edytorskie, które nie uniemożliwiają jednak odbioru pracy:

- Błędy językowego. W szczególności: niepoprawnie stosowane (lub brak zastosowania) przedimków a/an/the, niepoprawne stosowanie czasów (niepoprawna odmiana czasowników) oraz brak konsekwencji stosowania tego samego czasu w opisie.
- Umiejscowienie rysunków zbyt daleko od miejsca odwołania się do nich (w szczególności odwoływanie się w rozdziale *x* do rysunku zamieszczonego w rozdziale *x*+1.
- Brak konsekwencji w oznaczaniu operacji mnożenia. Operacja mnożenie raz oznaczana za pomocą znaku „x", raz ze znakiem „∘", a raz bez żadnego oznaczenia.
- Niekompletne opisy tabel i rysunków. Np. tytuł Tabeli 7.2 to „czas zbieżności", natomiast w tabeli przedstawione są również inne dane (poza czasem).

4. Wnioski końcowe

Recenzowana rozprawa doktorska potwierdza szeroką wiedzę teoretyczną mgra inż. Grzegorza Panka w zakresie sieci mobilnych 5G oraz sieci z przetwarzaniem brzegowym. Rozprawa realizowana było jako doktorat wdrożeniowy i należy podkreślić, iż dokonania Autora mają charakter mocno aplikacyjny oraz odpowiadający na potrzeby rynku telekomunikacyjnego. W ramach rozprawy zaimplementowany został autorski symulator, wykorzystujący rozwiązania i algorytmy używane przez rzeczywistych operatorów. Do rozwiązywania problemu relokacji usług zaimplementowane zostały natomiast autorskie propozycje Autora. Dokonanie Autora oraz przygotowane publikacje naukowe potwierdzają również jego umiejętność prowadzenia samodzielnych badań naukowych na wysokim poziomie (o czym świadczy również dorobek publikacyjny Autora, w którym znajdują się prace opublikowane w tak prestiżowym czasopiśmie jak *IEEE Communications Magazine* oraz podczas renomowanych konferencji jak *IEEE Global Communications Conference (GlobeCom)* oraz *IEEE International Conference on Communications (ICC)*). Zaprezentowane w rozprawie osiągnięcia i propozycje Autora stanowią oryginalne rozwiązania dla istotnego oraz aktualnego problemu badawczego, a uzyskane wyniki badań (w tym opracowany symulator) są rozwojem wiedzy pozwalającym ulepszać istniejące lub projektować nowe protokoły sieciowe, algorytmy i mechanizmy sterowania sieciami mobilnymi.

Praca spełnia więc ustawowe wymagania stawiane rozprawom doktorski, stąd wnioskuję o jej dopuszczenie do publicznej obrony. Dodatkowo, praca ma charakter aplikacyjny, a w ramach jej realizacji powstał system (autorski symulator), który może zostać bezpośrednio wykorzystany do zarządzania węzłami sieci 5G. Stąd też, praca spełnia wymagania stawiane doktoratom wdrożeniowym.

WARSAW UNIVERSITY OF TECHNOLOGY

DISCIPLINE OF SCIENCE INFORMATION AND COMMUNICATIONS TECHNOLOGY
FIELD OF SCIENCE ENGINEERING AND TECHNOLOGY

Ph.D. Thesis

Grzegorz Piotr Panek, M.Sc.

Application relocation in an Edge-enabled 5G-system

Supervisor
Halina Tarasiuk, D.Sc., Ph.D.

WARSAW 2024

to my wife Justyna, son Fryś, and my parents.

especially to my late Dad...

a wish left unspoken yet ever-present...

Mojej żonie Justynie, Frysiowi oraz moim Rodzicom.

W szczególności mojemu zmarłemu Tacie...

Marzenie niewypowiedziane, jednak zawsze obecne...

Application relocation in an Edge-enabled 5G-system

ABSTRACT

With the growing development of 5G networks and its new services, Edge Computing is becoming the cornerstone of the ongoing network transformation. Its integration into 5G network brings new research opportunities related to the design and implementation of high-performance system, enabling the accomplishment of the three main promises of 5G network: very high throughput, low latency, and massive connectivity. This development has generated strong interest in realizing effective life cycle management of low latency-sensitive Edge applications in order to achieve a high level of QoS while ensuring the service continuity in the case of user mobility. This thesis deals with the relocation of Edge applications, commonly called Edge Relocation, which aims to relocate Edge application instances between Edge Hosts in order to ensure uninterrupted service in user-mobility scenario. To achieve our objective, a cloud-native Edge-enabled 5G system compliant with ETSI and 3GPP standards has been proposed. The contribution of this thesis is three fold.

Firstly, a proof of concept is presented. It demonstrates how Edge Relocation can be implemented on the top of integrated 5G and Edge system to ensure service continuity in the case of end user mobility or Edge infrastructure unavailability. Implementation of relocation mechanisms involves utilizing and expanding open-source technologies contributing to the industrial aspect of the thesis. Kubernetes, recognized as the standard for cloud-native application orchestration, is utilized and Edge Multi-Cluster Orchestrator (EMCO) solution that is providing the capability of orchestrating Edge applications in a multi-cluster environment.

Next, two original algorithms are introduced. These algorithms are designed to identify a suitable Edge Host for application relocation. The first algorithm adopts a heuristic approach, consequently dividing the edge topology into sub-topologies and exploring them until an appropriate Edge Host is identified. This approach is then compared with the final contribution of this thesis, which use Reinforcement Learning to train a decision model. A detailed comparison is conducted, revealing the strengths and weaknesses of both solutions, providing valuable *insights for telecommunication operators considering the deployment of Edge-enabled 5G system.*

Key words: *Edge Computing, 5G network, cloud-native, Management and Orchestration, heuristic algorithm, Reinforcement Learning algorithm*

Relokacja aplikacji w systemie 5G z przetwarzaniem brzegowym

STRESZCZENIE

Rozwój sieci mobilnych oraz związane z nim możliwości realizacji nowych, zaawansowanych usług, otwierają nowe obszary badawcze w zakresie projektowania systemów sieciowych, które umożliwiają realizację trzech głównych założeń sieci 5G, tj. bardzo duża szybkość bitowa, bardzo małe opóźnienia oraz masowa łączność pomiędzy urządzeniami. W szczególność, dynamiczny rozwój usług wymagających bardzo małych opóźnień komunikacyjnych, sprawił że przetwarzanie na brzegu sieci (ang. Edge Computing) stało się kluczowym rozwiązaniem w trwającej transformacji sieci. Integracja systemu przetwarzania danych na brzegu sieci z architekturą sieci 5G stawia nowe wyzwania takie jak np. efektywne zarządzanie cyklem życia aplikacji brzegowych. Celem rozprawy było zaprojektowanie systemu oferującego wysoką jakość usług (ang. Quality of Service) realizowanych na brzegu sieci, jednocześnie zapewniając ciągłość świadczenia usługi w przypadku mobilności użytkownika końcowego.

W rozprawie skupiono się na procesie relokacji aplikacji brzegowych, który polega na przeniesieniu aplikacji pomiędzy serwerami brzegowymi, aby zagwarantować ciągłość świadczenia usługi podczas mobilności użytkownika. Zaproponowane zostało rozwiązanie dla sieci 5G bazujące na przetwarzaniu w chmurze oraz zgodne ze standardami organizacji ETSI i 3GPP. Przedstawiono trzy główne obszary badań prowadzonych w ramach tej rozprawy.

Początkowo zaprezentowano oryginalne rozwiązanie, demonstrujące procedurę relokacji aplikacji brzegowej w środowisku 5G wzbogaconym o funkcjonalność przetwarzania na brzegu sieci. Zbudowany w tym celu demonstrator umożliwia przetestowanie w praktyce mechanizmu zapewniania ciągłości świadczenia usług w przypadku mobilności użytkownika końcowego lub problemów z infrastrukturą serwerów brzegowych. Wdrożenie procedury relokacji obejmuje wykorzystanie i rozszerzenie pre-komercyjnych rozwiązań o otwartym kodzie źródłowym: Kubernetes jako narzędzie uruchomieniowe aplikacji brzegowych w środowisku chmurowym, oraz EMCO jako orkiestrator i zarządca aplikacji brzegowych.

Następnie zaproponowane zostały dwa oryginalne algorytmy mające na celu wybranie odpowiedniego serwera brzegowego do relokacji aplikacji w przypadku mobilności użytkownika końcowego. Pierwszy z algorytmów reprezentuje podejście heurystyczne, które dzieli topologię serwerów brzegowych na podzbiory, a następnie analizuje je aż do momentu znalezienia odpowiedniego serwera. Natomiast drugi algorytm został opracowany z wykorzystaniem metody uczenia maszynowego ze wzmocnieniem (ang. Reinforcement Learning) do trenowania modelu decyzyjnego. Przeprowadzono oraz przestawiono szczegółowe porównanie zaproponowanych algorytmów relokacji aplikacji, pokazując mocne i słabe strony obu rozwiązań, a także dostarczając szereg spostrzeżeń do wykorzystania przez operatorów telekomunikacyjnych rozważających wdrożenie takich rozwiązań do systemów 5G z przetwarzaniem na brzegu sieci.

Słowa kluczowe: *przetwarzanie na brzegu sieci, sieć mobilna 5G, infrastruktura chmurowa, orkiestracja aplikacji brzegowych, algorytm heurystyczny, algorytm uczenia maszynowego przez wzmacnianie*

Contents

Chapter 1

Introduction

With the growing development of 5G networks and its new services, Edge Computing is becoming the cornerstone of the ongoing network transformation. Its integration into 5G network brings new research opportunities related to the design and implementation of high-performance systems, enabling the accomplishment of the three main promises of 5G network: very high throughput, low latency, and massive connectivity. The convergence of 5G network and Edge Computing has changed the technology landscape, ushering in a new era of innovative use cases and accelerating the implementation of an intelligent, fully connected digital world at an unprecedented pace. However, the stringent requirements coupled to the high dynamicity of these new applications make their management and orchestration extremely challenging. The mobility of end-users is a critical factor expected to significantly impact Edge operations. As a result, Edge-enabled 5G systems face the daunting task of tracking moving users and meeting their Quality of Experience (QoE) demands simultaneously, which presents a formidable technical challenge that must be addressed to obtain a truly seamless and ubiquitous user experience. Telco stakeholders are urged to design innovative distributed systems implementing disruptive operations in order to fulfill the dream of a fully connected, intelligent digital world. Such new Edge-enabled 5G system should be capable of ensuring the orchestration of the deployed Edge applications while maintaining their Quality of Service (QoS). Specifically, these aforementioned systems must guarantee an uninterrupted communication with Edge Hosts when the users are moving, by providing orchestration mechanisms such as Edge application relocation.

According to the 3rd Generation Partnership Project (3GPP) [10], Edge Relocation is one of the main issues that are also addressed by the European Telecommunication Standardization Institute (ETSI) in the context of integrated MEC and the 5G network [8]. **By Edge Relocation, we refer to the ability to relocate the Edge application running on a source MEC Host to a target MEC Host.** Several Edge applications, leveraging 5G (such as: autonomous vehicles, cloud gaming, eXtended Reality or Autonomous UAVs), may require to guarantee QoS, specifically, very low latency communication and high availability. Hence, the Edge infrastructure will be highly stressed observing high load, or highly mobile users. In this perspective, it is very important to support the migration of applications in order to ensure service continuity during the mobility of the end-user or in case of source Edge Host performance degradation (e.g., lack of resources, failure, etc.).

In this dissertation, we deal with the mobility impacts on operations in Edge-enabled 5G system. We propose an original Edge Relocation solution that provides the capability to follow moving users while jointly respecting their Edge applications' latency and infrastructure resource requirements. The contribution of this dissertation is three-fold:

- Firstly, Edge Relocation framework named `5G-Edge Relocator` is designed and implemented. It leverages Kubernetes, the de-facto standard for cloud-native application orchestration and Edge Multi-Cluster Orchestrator (EMCO) solution [4] providing the capability of orchestrating Edge applications in a multi-cluster environment. Proposed framework relies on an ETSI and 3GPP compliant architecture, leveraging cloud-native Edge-enabled 5G system [38, 1, 74]. It is responsible for the relocation of Edge applications between Edge clusters. It besides ensures the observability of the Edge and access network infrastructures in order to select new Edge cluster destinations.

- Secondly, a relocation algorithm is proposed, named EAR-Heuristic (Edge Application Relocation Heuristic), which supports the selection of the destination cluster while jointly responding to the application requirements and load balancing the resource consumption of the Operator infrastructure.

- Finally, another novel algorithm based on Reinforcement Learning is proposed to provide the selection of new Edge Host called EAR-RL (Edge Application Relocation Reinforcement Learning). Our proposed solution aims to achieve a balance between two key objectives: maintaining QoS for Edge applications and obtaining load-balancing of Edge infrastructure. By achieving this balance, algorithm can ensure that Edge-enabled 5G system operate efficiently, with minimal resource wastage and maximal QoE satisfaction for end-users.

The PhD process has been carried out in an industrial mode as a collaboration between Orange Innovation Poland and Warsaw University of Technology, with support from the expertise of Orange France. The PhD candidate holds a position of Research Engineer at Orange Innovation Poland and is pursuing their PhD at Warsaw University of Technology.

During the PhD process, three research papers related to Edge Relocation were prepared and published them in: proceedings of IEEE Communication Magazine 2022 [74]; proceedings of IEEE International Conference on Communications (ICC) held in Rome in 2023 [75] and IEEE Global Communications Conference (Globecom) held in Kuala Lumpur in 2023 [73]. Additionally, we presented our industrial contribution and Edge Relocation implementation solution at several industrial conferences, including: KubeCon 2022 in Valencia (demonstrating the Proof of Concept of the EMCO feature for Edge Relocation prepared with Intel, which is the originator of EMCO); "Telco at Edge Days", a co-located event of KubeCon 2023 in Amsterdam and during Orange Open Tech Days in Paris in 2023. The first contribution (described in Section 5.4.1) that presents `5G-Edge Relocator` has been published in IEEE Communication Magazine and at IEEE ICC Conference. The evaluation of heuristic algorithm, that states a second contribution (described in Section 7.2.5) has been presented at IEEE ICC Conference, while the algorithm based on Reinforcement Learning (third contribution described in Section 8.3.4) has been published at IEEE Globecom Conference. This dissertation also contains an extensive evaluation of the above mentioned algorithms that has not been published yet (such as scaling evaluation described in Section 7.2.7 or RL non-masked approach introduced in Section 8.3.4).

This thesis is organized as follows: the chapter 2: Background introduces 5G and MEC system, while indicating the complexity of integrating both systems. Next, in the chapter 3, we describe the technology challenges observed in Edge Computing that we focus in this work. Next, the chapter 4 presents related works that provides an analysis of the application migration in Edge Computing problem, based on current state of the art. Chapter 5 describes the Edge Relocation procedure within the Edge-enabled 5G system and offers a perspective on the proposed demonstrator. This Section represents the first industrial contribution of our work. In chapter 6, we formulate the problem and present modelling of the environment. Next, in the chapter 7 we introduce heuristic algorithm for selecting new Edge Host for user equipment in mobility scenarios and present the evaluation results. This is the second contribution of our work. Chapter 8 is dedicated for describing the third contribution, which is a Reinforcement Learning-based approach for selecting Edge Host. Finally, in chapter 9, we provide a comprehensive summary of all the activities considered in this thesis.

Chapter 2

Background

This chapter introduces the background of 5G network connectivity, emphasizing its capability to provide ultra-reliable low-latency communication. Next, we deep dive into architectural view on 5G network and its integral enablers like Edge Computing. We clearly introduce the challenge of life-cycle management for Edge Applications and demonstrate its usage in an integrated 5G and MEC System. Finally, we present an overview of our motivation, driven by growing number of use-cases, which requires mechanisms and algorithms to support session and service continuity in proposed system.

2.1 5G network empowered mobile connectivity

The dynamic evolution of mobile communication since the 1980s has been driven by the growing importance of mobile networks in modern industries. The societal dynamics of each generation push forward the development of next generation mobile communication standards. In this era of wired connectivity, mobile communication has become a part of our daily lives, what facilitated plenty of services and allowed transforming the way how we interact with surrounding world. The continuous improvement of mobile networks has played a key role to enable the explosion of new innovative services, that together with 5G technology, are ready for changing the telco landscape. The 5G mobile network, represents a notable improvements in terms of available throughput, capacity, and reliability. 5G network has promised to realize a set of game-changing capabilities, including:

- Enhanced Mobile Broadband (eMBB) that provides significantly higher throughput compared to 4G technology, targeted 20 Gbit/s for downlink and 10 Gbit/s for uplink [49].
- Massive Machine-Type Communications (mMTC) to guarantee the capacity to connect a massive number of devices to the network [27].
- Ultra-Reliable Low-Latency Communication (URLLC) which ensures extremely low communication latency in the range of few milliseconds [76].

This is a significant advancements for applications that require real-time or near real-time responses, such as remote surgery, autonomous vehicles, industrial automation, and augmented/virtual reality. All mentioned 5G network features state key enablers for several new applications across various sectors, such as smart cities and Industry 4.0, healthcare, entertainment services like virtual reality (VR), and beyond.

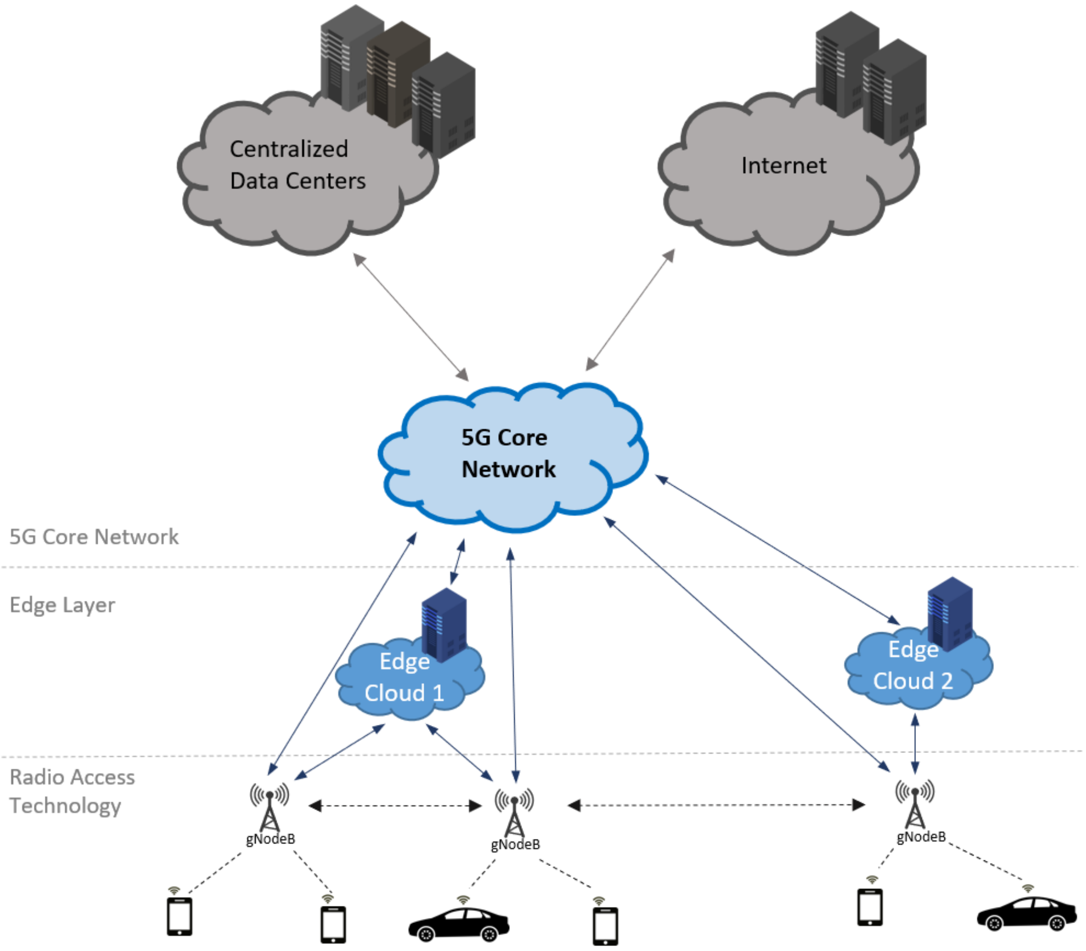


Figure 2.1: Edge-enabled 5G system

5G technology introduces and integrates new techniques to achieve promised low-latency communication. One of the key enabler for 5G network to realize URLLC is Edge Computing - technology that has been standardized by the European Telecommunication Standards Institute (ETSI) in a way to be widely used in mobile systems and finally named Multi-Access Edge Computing (MEC) [40]. The MEC aims to process user's data as close as possible to the end-user, typically at the network's edge. This entails attaching more computing servers closer to the end-users at: an edge local data centers/clouds, or co-located within gNodeBs as presented in Figure 2.1. Edge Computing plays a crucial role in enabling URLLC, while it offers as well numerous advantages to telco operators, not limited to just reduce network latency:

- **Bandwidth Efficiency:** Edge Computing also reduces the volume of data transmitted over long distances to reach centralized data centers, what results in bandwidth usage savings and minimize associated costs.
- **Reliability:** MEC can improve service reliability by providing local failover options. In case of network disruptions or failures in one location, Edge Hosts can continue to operate, ensuring uninterrupted service for users.
- **Low-Latency:** It reduces the round-trip time for data transport to reach the processing point due to the lower distance to network infrastructure, facilitating efficient near-real-time data processing, essential for URLLC services.
- **Security:** MEC enables more granular control over user data. Data can be processed locally instead of sending it to a central cloud, allowing for better control over data privacy and reducing the risk of data exposure.

2.2 ETSI based Multi-Access Edge Computing

Edge Computing has been standardized by ETSI to make it accessible within different mobile systems. This section presents the main concepts of MEC and its functional split based on final reference architecture that has been published in 2022 [40]. MEC System, as presented in Figure 2.2 is composed of three functional levels that can be grouped into MEC System level, MEC Host level, and Access Network level.

The **MEC Host level** comprises the **MEC Host** entity, which provides virtualization infrastructure (VI). VI delivers compute, storage, and network resources for **MEC Applications**. The MEC Host includes a data plane responsible for data transport among applications, services, the 3GPP network, and other local or external access networks.
The **MEC Platform** provides a set of supportive functions for MEC Applications, such as: configuration of internal DNS, instructing the MEC Host Data plane, or creating an environment where MEC Applications can discover, advertise, consume, and offer other MEC services. Additionally, MEC Host is managed by an entity responsible for tasks such as allocating, managing, and releasing virtualized resources in the virtualized infrastructure or preparing the VI to run a software image.

The **MEC System level** management includes the **MEC Orchestrator (MEO)** as a core component that maintains an overall view of the MEC System. The main MEO responsibility is to:

- maintaining an overall view of the MEC System based on deployed MEC Hosts, while

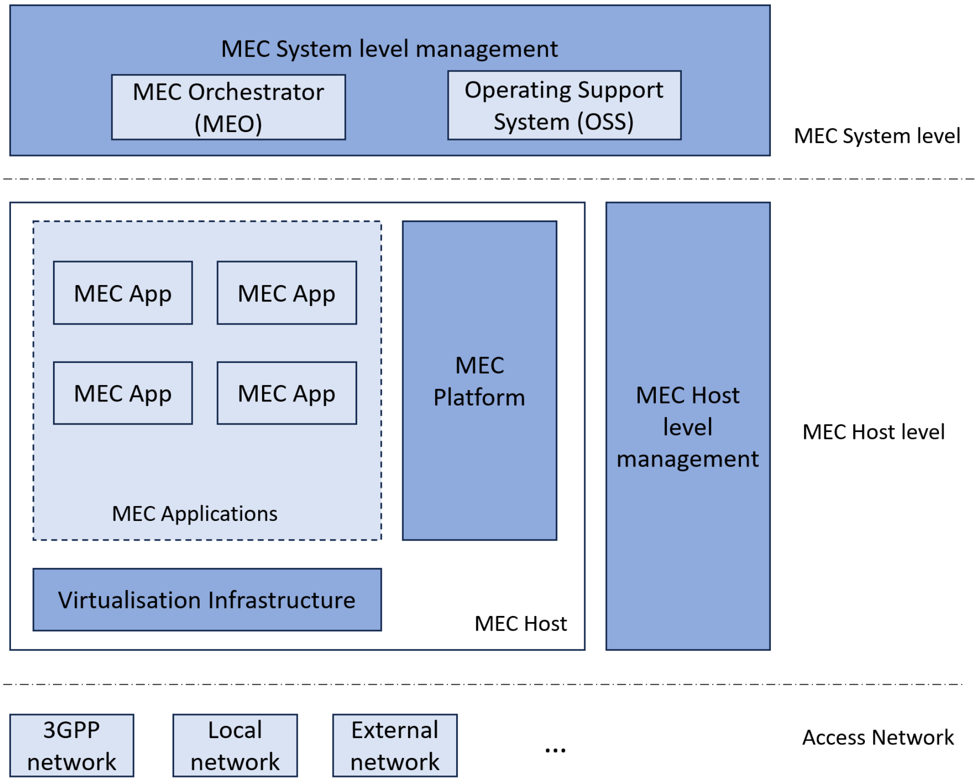


Figure 2.2: ETSI based MEC reference architecture. Based on [40]

monitoring available resources, available MEC services, and the topology of MEC Hosts;

- onboarding of application packages and communication with VI managers to instantiate applications;
- selecting appropriate MEC Host(s) for application instantiation and/or relocation based on a set of constraints, such as latency or available resources;
- triggering application instantiation, termination, relocation, and other life-cycle operations.

In addition to MEO, the management system layer also includes Operations Support System (OSS) of an operator, which receives high-level request (often service-level requests), translate them and finally transmit for execution to MEO.

The last, but not the least is the **Access Network** level. In order to allow end-users to communicate within MEC System, particularly with MEC Applications installed at MEC Hosts, they need to access the system through external Access Network, that can be either 3GPP based 4G/5G network, or any other type, including private, local, external networks, or even WiFi or non-3GPP access.

It is worth noting that MEC System has been developed in complement to Network Function Virtualization (NFV) concept [19], which offers several deployment options. One of them is to to deploy it together with Virtual Network Function (VNF) on the same, common Virtualized Infrastructure. Next, Management and Orchestration (MANO) [13] of VNFs can execute some of the MEC management and orchestration tasks. It proves a strong synergy between the 5G system deployment model (relying on NFV) and the potential deployment of MEC in similar manner.

2.3 5G network architecture

This section aims at providing comprehensive overview of 5G network architecture and its key components. It has been standardized by the 3rd Generation Partnership Project (3GPP) and recently published in Release 17 (2022) [11]. The architecture of the 5G network represents an evolution from previous generation of mobile networks. From its beginning, 5G network has been designed in a microservice based approach [101]. Such an approach imposes mainly in decomposition of a large monolithic architecture of 4G network into Network Functions (NFs) each performing, single, specific logical function. Similar to previous system of mobile communication, 5G network architecture can be divided into following architectural parts: Control Plane (called 5G Core), User Plane and Radio Access Network (New Radio in 5G standard).

The **Control Plane** (referred as 5G Core) has been designed with the principle of a Service-Based Architecture (SBA) [82]. This design principle is depicted in Figure 2.3, where the central point of 5G Core contains a common message bus which allows any NF to communicate any other NF. This is useful for the future evolution of 5G Core, allowing new NFs to be easily added to existing core functions. The architecture of the 5G Core, depicted in Figure 2.3, highlights only the Network Functions that are either considered or utilized in our study.

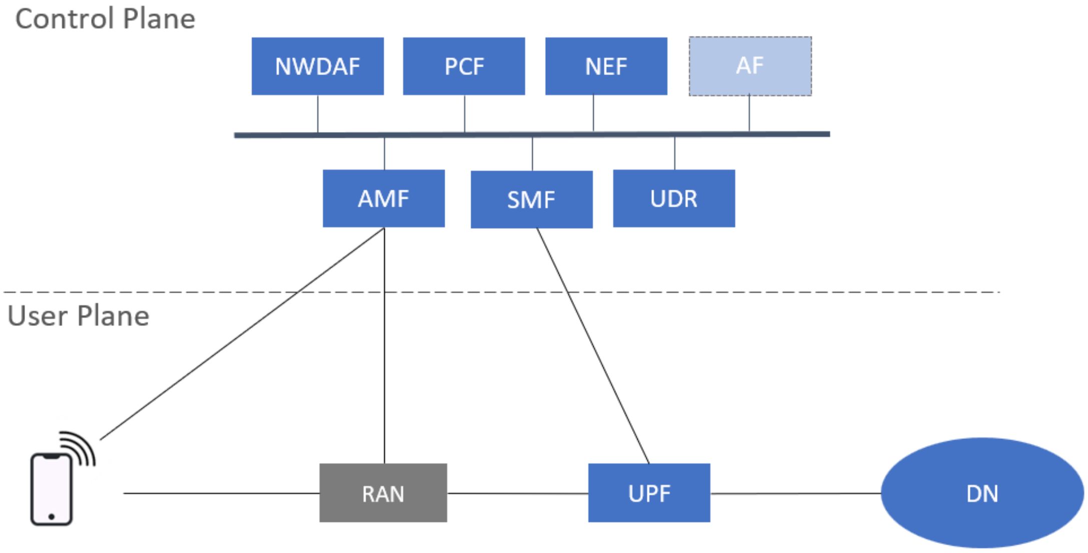


Figure 2.3: 3GPP 5G Network Architecture. Based on [11]

- **Access and Mobility Management Function (AMF)** handles the UE's management of connection and mobility tasks such as authentication, authorization, registration and primarily mobility management. It is also responsible for the selection of the session management function (SMF) which manages UE's Packet Data Unit (PDU) session(s) [45].

- **Session Management Function (SMF)** is responsible for the creation, modification and deleting of the PDU session and the allocation of the IP addresses to the end-user. Besides, it selects and controls the User Plane Function.

- **User Plane Function (UPF)** performs the packet routing and forwarding to the suitable data network. Additionally, it provides local breakout capabilities for the traffic to be routed to applications in edge locations.

- **Policy Control Function (PCF)** provides policy rules to the AMF and SMF functions to ensure a fine granularity control of the authorized end-user's flows. Precisely, the PCF influences the traffic routing by passing new policies to the SMF in order to update the PDU session based on the location of the end-user. SMF configures the traffic rules on the serving UPF accordingly.

- **Network Exposure Function (NEF)** securely exposes services' capabilities provided by the control network functions (e.g., SMF, AMF, PCF) to the MEC System [58]. In doing so, the latter can interact with the core network in the process of the PDU session update by traffic steering rules reconfiguration. It also enables external AF an authentication capability [58].

- **Application Function (AF)** AF is a generic term for any function that a telco operator can connect to the 5G Core message bus to communicate with all other NFs. If the AF is authenticated, it can communicate directly with other Core NFs; otherwise, it needs to transmit all messages through the NEF, which acts as an authorization gateway to external AFs.

- **Unified Data Repository (UDR)** is a centralized data repository for users' information. It plays role of a database where information such as: subscription data, subscriber policy data, sessions, contexts, SIM identities are stored.

- **Network Data Analytics Function (NWDAF)** stands for a new approach for data collection and analysis internally in mobile network control plane. It is mainly responsible for subscribing for any data coming from NFs as stated in [50, 106], analyse according to the defined policies and propose insights [28].

The architecture contains as well DN what stands for Data Network and represents any type of Service Provider services, internet access or 3rd party services. In our study DN is considered as MEC entity.

The final element of the architecture is the Radio Access Network (RAN). It enables wireless communication for users to connect 5G infrastructure and other part of network, such as the internet, or any Data Networks. In the presented architecture RAN (Figure 2.3) is specified as a single component without differentiation between multiple gNodeBs (5G NR base stations) or the splitting of a Radio Access Functions, since it is not the focus of this thesis.

2.4 Integration of 5G network and MEC technology

As mentioned at the beginning of this chapter, the integration of MEC technology into the 5G network architecture is a key enabler for implementing URLLC service use-cases. This section demonstrates how to interconnect both systems, which have been standardized by two independent standardization bodies.

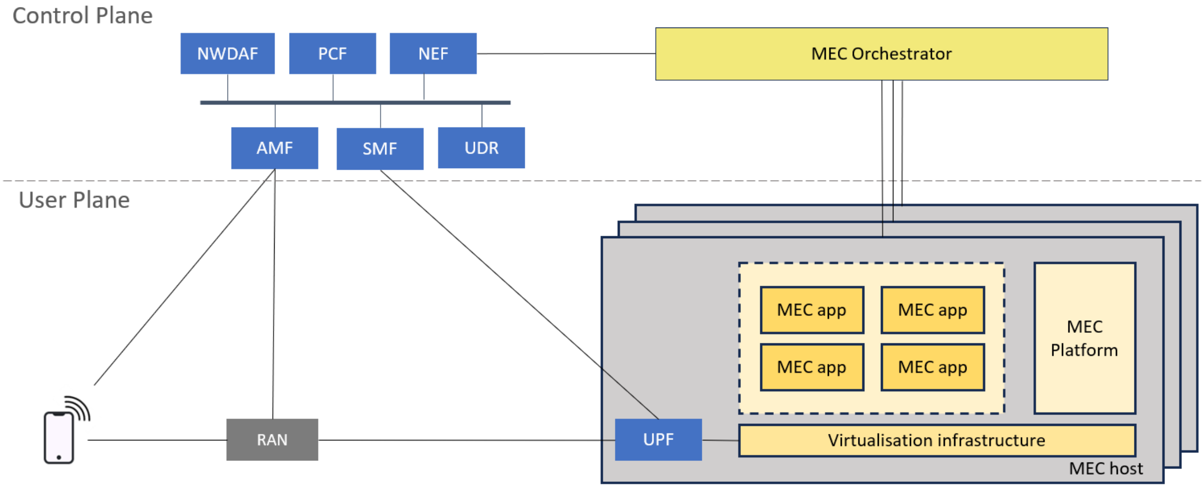


Figure 2.4: Integration of MEC technology and 5G architecture. Based on [6]

Two dimensions of both systems integration can be observed. First, integrating the 5G Control Plane and MEC System management requires interconnect of MEC Orchestrator and other 5G Core Network Functions. According to ETSI [6], a MEC Orchestrator can be served as an Application Function to 5G Core. It means, if MEC Orchestrator is a trusted entity, it can be directly configured as an AF. However, if the MEC Orchestrator is an untrusted entity, establishing an interconnection between the MEC Orchestrator and the Network Exposure Function is necessary to grant authentication. The NEF serves as a gateway for communication when exposing data from any 5G Core services.

In the User Plane, user data is forwarded to the external Data Network, which, in assumed architecture (Figure 2.4), is the MEC Host. This forwarding is possible by mechanisms implemented in 5G system as functionalities of the User Plane Function (UPF), such as local breakout or an Uplink Classifier. These mechanisms route traffic to specific Data Networks within the Edge Infrastructure. Moreover, ETSI has defined several deployment models [6] for UPF in MEC solution. As suggested, UPF has been deployed at Edge Provider's infrastructure to minimize latency by being deployed as close to the MEC Host as possible.

Both 3GPP and ETSI have developed several procedures to facilitate the above-mentioned integration. These procedures encompass: a) MEC Orchestrator's influence on 5G network traffic steering to affect traffic routing and proper UPF (re)-selection, and b) MEC Orchestrator requesting various resources from 5G NFs. AMF can provide MEC Orchestrator with a data related to User Equipment (UE) mobility, while SMF can deliver PDU Session information. Further details on these procedures are described in the following chapter 4.

2.5 Motivation for Edge Relocation

The deployment of 5G network is driving the growth of the Edge Computing market, which is estimated to be $176 billion in 2022 [17]. Indeed, Telco stakeholders are investing in Edge Computing to run new disruptive use cases related to industry 4.0, autonomous vehicles, extended reality (XR), cloud gaming, etc. Therefore, they aim to harness the benefits of 5G network alongside the Edge in order to monetize their highly performing infrastructures. However, the proliferation of these new use cases leads to an exponential increase of mobile data traffic with data rates in the magnitude of terabits per second, latency of milliseconds, and mobility speed reaching 500 km/h [49]. To cope with this unprecedented growth driven by the emergence of new services with more stringent requirements, Telco stakeholders are urged to design innovative distributed systems implementing disruptive operations in order to fulfil the dream of a fully connected, intelligent digital world.

Such new Edge-enabled 5G systems should be capable of ensuring the Management and Orchestration (MANO) of the deployed Edge services while maintaining their Quality of Service (QoS) [24, 34]. Specifically, these aforementioned systems must guarantee an uninterrupted communication with Edge Hosts when the users are moving. The radio-access handover procedure at 5G system side is well known and standardized by 3GPP [12], while a service continuity for users' connected to Edge architecture is a main subject of this work. Mobility of end users requires set of Life-Cycle management operations done on Edge services in order to guarantee uninterrupted communication between end-users and Edge Applications. This is where Edge Relocation concept needs to be introduced.

According to the 3GPP [10], and ETSI, the Edge Relocation is one of the main issues addressed in the context of Integrated MEC and the 5G network [8].

By Edge Relocation, we refer to the ability to relocate the Edge Application running on a source MEC Host to a target MEC Host [74].

Edge Relocation procedure may be triggered by several events, which can be either driven by the 5G Core or the MEC Host, e.g.:

- (5G Core) The UE moves out of coverage area of serving MEC Host (source MEC Host);
- (5G Core, MEC) The QoS level decreases due to radio connection degradation;
- (MEC) The MEC Host is no longer able to host the application due to the lack of resources (e.g., MEC Host overload, MEC Host failure).

The Edge Relocation is a key enabler of Edge-Enabled 5G system. Several Edge Applications, leveraging 5G network (e.g., autonomous vehicles, cloud-gaming, etc.), may require strict QoS, specifically, very low latency and high availability. In this perspective, it is very important to support the migration of applications in order to ensure service continuity during the mobility of the end-user across the system or in case of source MEC Host performance degradation (e.g., lack of resources, failure, etc.). It is worth noting that such a problem differs with Edge Application offloading [99, 105, 104, 79], where the goal is to offload the application workloads from User Equipment (UE) to Edge Hosts to allow better autonomy while improving the application performance.

2.6 Potential use-cases

The advent of 5G technology has driven the emergence of new use cases sensitive to latency. Some of them such as e-health, video streaming, unmanned aerial vehicles, autonomous vehicles, cloud-gaming and extended reality have, in addition, mobility needs [42]. To deal with the aforementioned challenging requirements, it is important to use the MEC solution to perform data processing or content delivery as near to the end-user as possible. Indeed, provided by the operator through its infrastructure, the MEC can offer an ecosystem for efficient and seamless application mobility [32]. Envisioning an intelligent Edge-enabled 5G system is crucial to achieve the targeted performances. As depicted in Figure 2.5, this system relies on several geo-distributed MEC Hosts. MEC Hosts are part of the Edge cloud infrastructure which is connected to the 5G Core network located in the Centralized Cloud.

Hereafter, we review the characteristics and requirements of use cases to which the Edge Relocation could bring a real added-value.

2.6.1 Autonomus vehicles

Autonomous vehicles are seen as the most relevant service that 5G network will deliver. Self-driving cars embed a set of sensors and cameras that are constantly processing the nearest surrounding. The autonomous central car management system is characterized by low latency communication ($\approx$ 15 ms) in high mobility scenarios (C-V2X) [69]. Therefore, it requires to be located as close as possible in order to guarantee an uninterrupted service continuity with quick response time. In this perspective, the support of application relocation and user data synchronization between MEC Hosts in order to manage and coordinate autonomous cars is crucial [85].

In a smart city, autonomous vehicles rely on a network of sensors, cameras, and other nearby infrastructure to traverse through their environment. These vehicles use advanced algorithms and real-time data to interact with various entities in the city, to determine the most efficient path to a particular destination. By analyzing data on traffic patterns, road conditions, and other factors, the autonomous vehicle can choose the fastest or nearest route to avoid potential hazards or disruptions, such as accidents or road closures due to construction. The communication delay between nearby entities and the vehicles must be minimal to avoid any disruption in the vehicle's

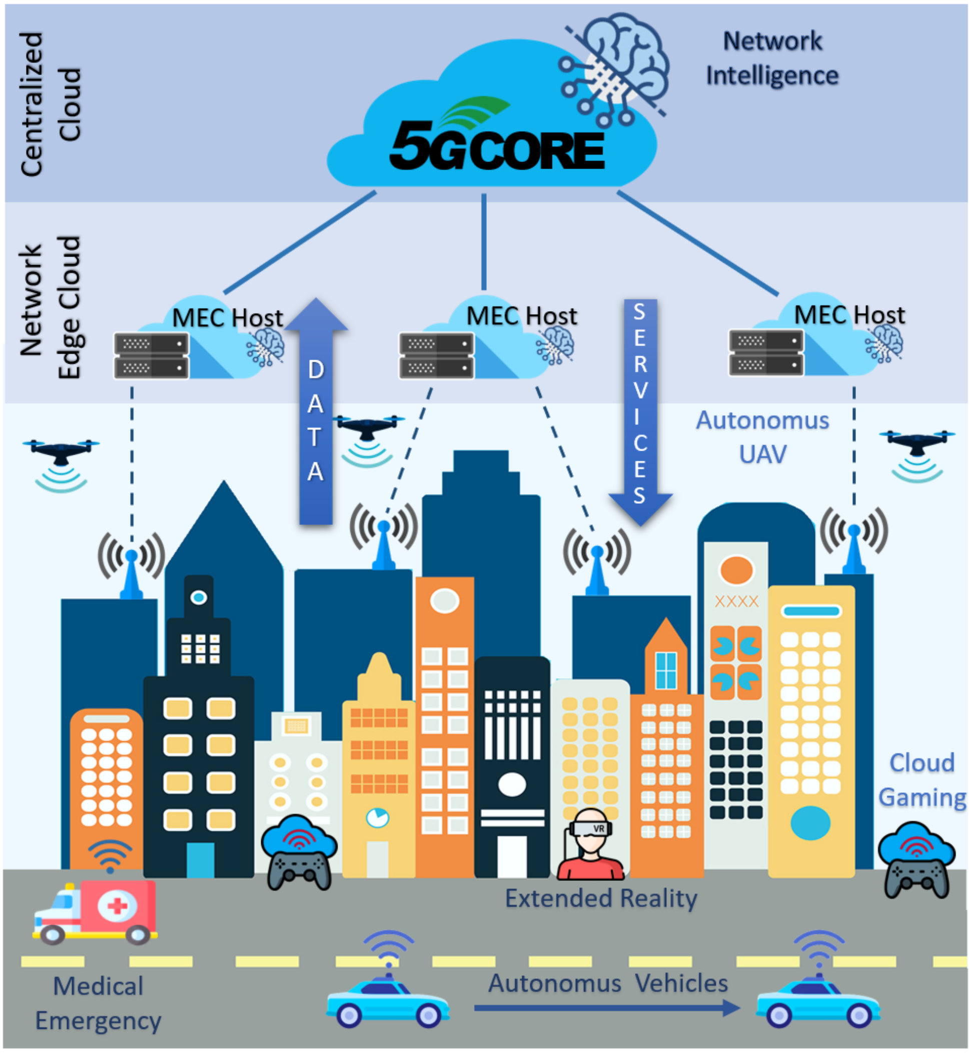


Figure 2.5: Edge-enabled 5G system with exemplary use-cases

operation during its mobility. The autonomous central car management system is characterized by low latency communication (≈ 15 ms) in high mobility scenarios (C-V2X) [69].

This communication enables the vehicle to maintain situational awareness of potential events that may occur during its journey, and as needed, dynamically adjust its route to account for any unforeseen circumstances. To minimize the delays that autonomous vehicles must tolerate when dealing with both planned and unplanned events, one approach for such vehicles is to delegate a portion of their computation to nearby external resources in the city to obtain real-time traffic updates. Autonomous vehicles can deploy software modules on the available near entities to gather, parse, and transmit pertinent information for navigation. These modules must ensure that communication delays between the car and the remote modules are minimal to avoid any disruption in the vehicle's operation, especially that the vehicle is mobile continuously. To address the communication challenge of autonomous vehicles in a smart city, 5G-coupled Edge Computing is essential. Edge Combined to 5G network offer additional hosting and execution resources for applications in close proximity to both data sources and end-users, in contrast to the internet backbone, where cloud providers are located at a distance. By leveraging Edge Hosts, autonomous vehicles are supervised by the infrastructure reducing hence the congestion and making the traffic more smoothly. In this perspective, the support of application relocation and user data synchronization between MEC Hosts in order to manage and coordinate autonomous cars is crucial [85].

Among others use cases, we leverage autonomous vehicles to validate the feasibility of our solution, where we select the distant supervision module while considering the mobility of vehicles.

2.6.2 Drones

Unmanned aerial vehicles (UAV) are among the most beneficiary services of MEC. Drones steering system requires short response time ($\approx$ 30 ms) to be guaranteed by computation resources at the Edge of the network. Edge Relocation is a key enabler MEC feature to achieve full automation of drones piloting. It makes it as useful as never before and gives opportunity to offer new services based on autonomous flying vehicles [86, 68]. UAVs are used in military sector for remote reconnaissance in high-risk areas. Drones are also widely used for security sector, to monitor cities, protection of properties and continuous boarder patrolling. Edge Computing allows for continuous images analysis in MEC Host in order to detect emergency cases.

2.6.3 Video Streaming

Video streaming services is experiencing significant growth in popularity. The continuous development of mobile networks allowed to transmit high quality video on demand (VoD). MEC Hosts allows to store the video content at the Edge of network in order to reduce large amounts of data to be transferred through the entire network and to guarantee latency constraints ($\approx$ 100 ms) even for ultra-high definition (UHD) video [69]. Edge Relocation is a support feature to utilize the 'follow me' procedure [71] for streaming services to keep video content as close as possible to the end-users. It is especially relevant in mobility scenarios such as watching videos in high-speed rails or in fast moving cars.

2.6.4 Cloud-Gaming

Cloud-gaming is shaping up to be the disruption of the computer game market. The main idea behind this emerging service is to run the game on a cloud server and stream the rendered video to the client [23]. In doing so, the end-users do not need to have high-end performance equipment to run the games anymore. However, such a service is extremely sensitive to network latency. Indeed, some game types (e.g. First Person Shooter) [7] require a delay of 10 ms or less from user operation to screen display. However, during the end-user mobility, this delay constraint can be easily violated since the MEC Host, hosting the cloud-gaming application, becomes increasingly far. Such a distance, especially in the case of high end-user speed, may cause additional latency, inducing QoS degradation.

2.6.5 Extended Reality (XR)

eXtended Reality (XR) is a concept that allows to combine real, physical view of environment with additional digital objects. Special headset or glasses facilitate daily routine by displaying visual objects or playing audio messages. The following examples of XR usage are strongly related with user mobility and require enabling Edge Relocation in order to guarantee service continuity. XR technique may support people with visual or hearing disabilities to simplify moving around in public spaces, making shopping or just sightseeing [93]. Augmented Reality may navigate people around city, or translate and display phrases from the mouth movement by recording and processing in real-time [61]. Extended Reality may also help drivers by notifying about road hazards, or support people working at huge industrial territories, e.g. docks, airports.

Taking into account the heterogeneity of the latency requirements from 10 ms to 100 ms for the discussed use cases, it is necessary to propose smart Edge Relocation solutions that will ensure service continuity.

2.7 Conclusions

Integration of MEC and 5G network technologies represents a significant advancement in modern telco networks. This synergy enables low-latency and reliable communication, opening the stage for development of a wide range of innovative applications across various sectors. Use-cases such as autonomous vehicles and drones, video streaming and cloud-gaming in high-speed railways, or extended reality will benefit greatly from this technology convergence. These use-cases demand low latency and high availability, which can be achieved through the efficient utilization of Edge resources and the support of Edge Relocation.

The proposed integration architecture of MEC technology and 5G network does not implement (therefore does not satisfy) all the requisite MEC interfaces as specified by ETSI. To avoid any misuse of the term "MEC system", the alternative term "Edge System" is used in this context in the rest of this thesis. If we are describing a system based on ETSI standards, we would use the term "MEC", however, when referring to our specific implementation, we would use "Edge Computing".

The concept of Edge Relocation assumes seamless migration of Edge Applications between Edge Hosts to provide Service and Session Continuity (SSC). This capability is crucial enabler for ensuring uninterrupted and reliable services in mobility scenarios. Such a mechanism has been designed and implemented as a part of industrial contribution of this thesis and is presented in section 5.

It is expected that coupling Edge Computing and 5G network should ensure Quality of Service for latency-critical services. This expectation brings new research opportunities for 5G network and beyond that are described in next chapter 3.

Chapter 3

Edge Computing: technological challenges and open research issues

The current status of Edge Computing development and its standardization (Multi-Access Edge Computing) brings several challenges related to the service continuity of Edge Applications [16]. ITU-T presents the strategy for Network 2030 that specifies new use-cases enabled by MEC and new services implemented at the edge [91]. Most of these use-cases consider user mobility. In this perspective, Edge Relocation is a key enabler. Hereafter, we give an overview about challenges and open research issues to be addressed to support service continuity in the Edge-enabled 5G system. As explained in the previous chapter, we use the term "Edge" instead of "MEC" to refer to the broader technology concept of Edge Computing.

3.1 Granularity of Edge Hosts distribution

Designing of Edge-enabled 5G system requires careful planning in terms of location density of Edge Host. To assure end-to-end latency for latency-sensitive applications it is desirable to position Edge Hosts in close proximity to gNodeBs. However, the real-world implementation of a geo-distributed Edge Computing system involves significant capital expenditure (CAPEX) for telecom operators when considering the deployment of individual Edge Hosts everywhere. The solution lies in finding a balance between system performance and effectiveness and investment possibilities. This challenge can be divided into two sub-challenges. First, is a multi-level classification, this means categorizing Edge Hosts into near-edge, far-edge, and central-edge categories in order to increase its deployment efficiency. Secondly, the capacity planning for each Edge infrastructure level is essential. It allows to determine the number of needed hosts and their overall capacity. Such a specification should follow numbers of expected users and estimated number of applications to be deployed.

3.2 Observability of Edge-enabled 5G system infrastructure

Observability is a cornerstone of the Edge Relocation approach ensuring insightful analysis of the Edge-enabled 5G system in order to provide an adequate Quality of Experience (QoE) [95]. Observability refers to the activities involving the measurement, collection and analysis of the various diagnostic signals that are fed back in real-time from both the cloud infrastructure and the applications running on it [53]. We recall that the Edge Relocation can be triggered mainly by two kinds of events: i) the mobility of users and ii) the degradation of the hosting Edge Host performances. That is why a joint observability of Edge-enabled 5G system infrastructure is required to provide an in-depth understanding of the network and application behaviors anticipating any QoS/QoE degradation. In this context two-level observability is required: i) Infrastructure-level including hardware (computing and memory resources, network topology) and software (Hypervisor, OS, containers) and ii) application-level including Key Performance Indicator (KPI) metrics and UE-mobility in order to derive statistical information about the UE mobility, and generate predictive information about future events. The development and design of a complementary observability system for Edge Computing systems presents a significant technological challenge as stated in [95].

3.3 Distributed Edge Multi-cluster Networking

The geographically distributed Edge Hosts (called Edge Clusters in the implementation perspective), spanning across different regions, data centers, or even infrastructure providers, present a research challenge in establishing efficient connectivity to guarantee uninterrupted communication among Edge services. To address this, a potential solution is proposed in the implementation of a multi-cluster (each cluster representing single Edge Host) service mesh [100, 36] — an infrastructure layer that contains service-to-service communication, observability, security capabilities across multiple Edge Clusters. This solution provide possibility of exploring features such as: effective management and orchestration of network connectivity, service discovery, traffic routing, and load balancing among services operating in geo-distributed Edge clusters.

3.4 Edge Multi-cluster Orchestration

The lifecycle management of Edge Applications is extremely challenging. Specifically, the placement, the scaling, the relocation and the observability of container-based instances are complex operations [63]. Unfortunately, the Vanilla version of Kubernetes [3] is not capable of efficiently orchestrating composite applications deployed on thousands of Edge Hosts. Achieving a zero downtime [83] application relocation between Edge Hosts needs several adaptions of Kubernetes to make it possible. Various Kubernetes based solutions for Edge Computing have born such as KubeEdge[1], Fleet[2] and EMCO[3]; however, none of them supports the operation of Edge Relocation.

3.5 Smart Edge Relocation decision

Coupling 5G network and Edge technology opens the opportunity to make use of 5G network functions or Edge Orchestrator to provide efficient Edge Relocation decisions. Specifically, NWDAF can leverage Artificial Intelligence (AI)-based approaches [90, 65] to run predictive user-mobility analytics, anomaly detection and trends analysis based on a mobility data provided by SMF/AMF functions [15]. Therefore, two key challenges to utilize potential coming from NWDAF is to define AI algorithms to follow and predict user mobility behavior (e.g. autonomous vehicles or unmanned aerial vehicles) and specify a set of user-mobility and/or QoS metrics to be measured and provided as an AI algorithm inputs. User mobility patterns may specify the next sector of user localization with a high probability and provides detailed information about timing constraints. Despite the complexity of single user mobility behavior

[1] https://github.com/kubeedge/kubeedge

[2] https://github.com/rancher/fleet

[3] https://github.com/open-ness/EMCO

predictions, the Edge Relocation system may be highly effective and desirable especially for emergency sensitive use-cases.

The self-learning ML-based algorithms are able to create user-mobility patterns to make Edge Relocation decision in-advance. In consequence it provides time advantages for executing zero downtime Edge Relocation process what can make Edge services ultra high reliable.

3.6 Edge synchronization

The use-cases presented in this thesis are considered to be handled as a both: statefull and stateless services and might require a careful context transfer management and synchronization across source and target instances. Telecom organizations need mechanisms to handle data replication, synchronization, and consistency across clusters to prevent data inconsistencies, conflicts, or stale information [67, 103]. Indeed, moving applications will have to be potentiality available or synchronized over several Edge Hosts. Since data will be distributed, Edge synchronization will ensure Edge Application data to be always consistent and tolerant to data distribution [39]. Efficiently relocating the user context is challenging. To relocate at once a whole package of user data during the Edge Relocation process what increase process duration, or to migrate only major part of user context before Edge Relocation and to update the remaining fresh part in incremental mode.

3.7 Conclusions

This section concludes both: research and industrial challenges for making Edge Computing for mobile systems industrialized and efficient in integrated 5G system.

The contribution of this thesis is threefold:

- To address the challenges of Edge Multi-cluster Management and Orchestration, we focus on key operations related to the life-cycle management of Edge Applications, as depicted in the blue circle in Figure 3.1. To do so, we have designed and developed a seamless procedure for relocating stateless applications between Edge Hosts. Our solution has been pushed to the open-source Edge Orchestrator - EMCO (Edge-Multi Cluster Orchestrator). Additionally, we have created a real demonstrator of an Edge-enabled 5G system, using open-source technologies in order to validate implemented procedure as shown in yellow circle in Figure 3.1. It worth underlying that EMCO stands as a pre-commercial solution provided by Orange for effective management of Edge infrastructure and services [41]. This represents and proves the industrial application of the thesis, particularly as the implemented Edge Relocation procedure has been integrated into the open-source Edge Orchestrator and is currently in active use by EMCO. It is worth mentioning that our demonstrator includes an observability framework, which allows for ongoing monitoring of Edge Clusters to address the observability challenges in Edge Computing.

- To meet the challenge of making smart Edge Relocation decisions, as highlighted in the red circle in Figure 3.1, we designed and implemented the "EAR" (Edge Application Relocation) heuristic algorithm. It is aimed at selecting the best Edge Host while considering resource usage optimization and minimizing latency. The evaluation of this algorithm in-

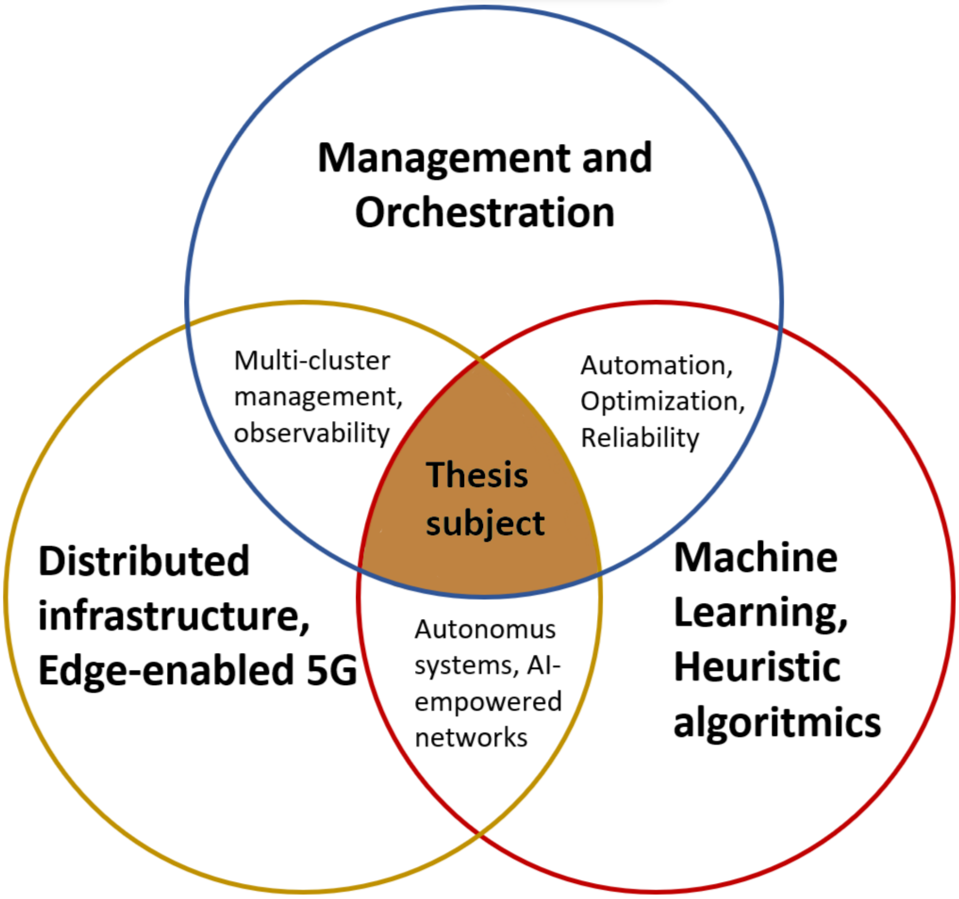

Management and Orchestration
Multi-cluster management, observability
Automation, Optimization, Reliability
Thesis subject
Distributed infrastructure, Edge-enabled 5G
Machine Learning, Heuristic algoritmics
Autonomus systems, AI-empowered networks

Figure 3.1: Composition of thesis subject

cludes two parts: a) Testing it on the demonstrator mentioned in point 1. b) Conducting performance tests on a original simulator of an Edge-enabled 5G system and comparing the results with other optimal strategies.

In preparing exemplary Edge Host topologies for algorithm evaluation purposes, we carefully considered the challenge of granularity in Edge Host distribution.

- Finally, we design, implemented, and trained a Machine Learning model (Reinforcement Learning) based on the Proximal Policy Optimization (PPO) algorithm to strengthen smart Edge Relocation decisions. The objective of this algorithm is as well to determine the best Edge Host. To evaluate this model, we also utilized the simulator mentioned in the previous point and we made a comparison with the `EAR-Heuristic` and `Optimal` approaches.

The first contribution of this thesis is related to the industrial mode of conducted research, while the second and third contributions have a more scientific focus, nevertheless, they still remain highly valuable for operational use.

Chapter 4

Related work

In this chapter, we provide an analysis of the related work that addresses the support of Edge services continuity focusing on Edge Relocation issue. Specifically, we concentrate on the areas related to facilitating the deployment and management of cloud-native applications in geo-distributed environments while considering the Edge Computing and mobile network context. By examining the existing literature on this topic, we shed light on the various approaches and solutions proposed in the literature to tackle this complex problem. We also deep dive into related work of 5G standardization bodies to analyse the current state in defining procedures that support Edge Relocation use cases.

4.1 Application relocation and service migration in Edge Computing

Following the literature, the relocation problem can be resolved through three main approaches, i) mathematical optimization models based on Integer Linear Programming (ILP) or Mixed Integer Linear Programming (MILP), which are solved using optimization solvers [94, 21], ii) heuristic methods [57, 86], and iii) Machine Learning (ML) approaches. Although exact methods are capable of achieving optimal results, they often struggle to resolve large scale and complex problem instances due to their time-consuming nature. In such scenarios, heuristics and meta-heuristics approaches can be used to provide relatively "faster" but "sub-optimal" solutions. However, the recent emergence of ML-based approaches with their interactive learning and decision-making abilities have garnered recognition for providing both accurate and rapid solutions, making them a promising alternative to traditional optimization techniques.

4.1.1 Heuristic and Linear Programming optimisation

The work presented in [21] addresses challenges in Unmanned Aerial Vehicle (UAVs) communication using MEC technology as an enabler for low-latency 5G mobile network connectivity. It proposes a proactive relocation approach using Linear Integer Programming based on UAVs' predefined flight plans. The goal is to maximize the availability of UAV applications in the serving MEC Host while considering relocation time. The paper introduces decision variables (e.g., time required for the relocation and the distance between the current and optimal MEC Hosts), constraints (e.g., elapsed relocation time), and linearization techniques for optimization.

Next, in [94] authors discuss the deployment and relocation of synchrophasor-based applications (PDC) in power grids. It introduces a cloud-edge architecture, virtual PDCs and an optimization algorithm based on binary integer linear programming. The goal is to meet latency and data completeness requirements of target applications in dynamic power grids. The proposed framework monitors network performance (latency mainly), triggering optimal vPDC relocations to respect latency requirements.

In the category of the meta-heuristic methods, paper [71] presents 'follow-me' approach in MEC system as an optimization solution for dynamic service placement due to user mobility, where the application is "following" end user. It proposes a mobility-aware framework aiming to minimize user-perceived latency while considering a predefined long-term migration cost budget. Heuristic based on the Markov approximation technique is proposed to handle the NP-hardness problem. The key metrics considered for relocation decisions include: capacity required by users at different MEC nodes, network propagation delay and data transmission delay, and operational cost (bandwidth usage and energy consumption).

Another heuristic approach is presented in [57], in which the issue of limited resources in user devices is presented. Authors consider MEC technology as a solution for large-size and latency-sensitive applications, while analyzing user mobility and different users' tasks dependency for workloads offloading to MEC Host. They proposed heuristic algorithm that aims to minimize workloads completion time in MEC Host by jointly considering these factors. The study evaluates the algorithm's performance through simulations, demonstrating its superiority over existing approaches.

4.1.2 ML-based methods

The category of ML-based methods have been the subject of several surveys that have examined the use of ML techniques for Edge Application relocation. Notably, [25, 35] have mentioned that the use of ML for distributed edge-cloud application migration remains an open area that requires further research and evaluation. In this research work [89], authors try to solve the problem of container migration. It was successfully tackled by using of Deep Q-Learning (DQL) solution, where each node is equipped with its own DQL agent. Their system's state is composed of delay metric, power consumption, and migration cost, while the action space is consisting of the set of nodes to migrate to. The action space was optimized by dividing the fog nodes into three groups: under-, normal-, and over-utilization. Furthermore, the training process proposed by authors was enhanced using Dueling Deep Q Networks (DDQN), which in simple words assigns different priorities to transitions in the experience memory, what leads to faster learning and greater stability. The really decent results of this solution demonstrate that the DQL approach facilitates swift decision-making and significantly outperforms existing baseline approaches, mainly in terms of delay, power consumption, and migration cost.

De Vita et al. in [96] developed a cutting-edge RL framework, that takes into account the interaction between an intelligent agent and an IoT environment. The proposed algorithm was designed to learn optimal policies during system development, with the ultimate goal of optimizing application relocation across various Edge servers within the network slice. The focus of proposed algorithm is primarily on improving latency and resource utilization in order to make it a highly effective tool for optimizing IoT systems.

The authors of [43] have developed a highly effective solution to the problem of service placement using a distributed deep reinforcement learning (DRL) approach. In particular, authors utilized an Importance Weighted Actor-Learner Architecture (IMPALA) that relies on actor-critic techniques. IMPALA addresses and fixes issue related to rapid adaptation and generalization existing in centralized DRL techniques. IMPALA is using an adaptive off-policy correction mechanism that enhances convergence. To further improve performance, the researchers has used several recurrent layers to process temporal behaviors and utilized a replay buffer to optimize the input sample quality. Additionally, in this paper autohrs utilize Directed Acyclic Graphs (DAGs) approach to formulate IoT services, which supports the dependencies between IoT services and reduces the complexity of the problem.

IMPALA addresses issues in centralized DRL techniques by employing an adaptive off-policy correction mechanism, boosting convergence despite challenges in quick adaptation and generalization.

In [98], authors put forth a novel hierarchical placement strategy for Edge Computing service that is optimized for reducing network congestion. To achieve this goal, they utilized Q-learning, a type of RL algorithm, to map tree services onto the physical network.

In paper [87] authors introduce the ML-enhAnced Edge Service OrchesTRation (MAESTRO) algorithm, utilizing NFV and ML for automated management and orchestration (MANO) operations in V2X services. They are focusing on ensuring QoS for V2X services by Edge services relocation, addressing challenges in 5G network and beyond. Real-life experiments on Smart Highway and Virtual Wall testbeds (that relies on Kubernetes Edge infrastructure) in Belgium validate the proposed algorithm. The MAESTRO algorithm combines Multi-Criteria Decision-Making (MCDM) and Support Vector Regression (SVR) to make proactive ML-driven decisions for Edge service relocation. It is important to note that authors relies on KPIs coming from both: infrastructure (CPU, Memory utilization) and network (latency, bandwidth).

The focus of next paper [33] is on optimizing QoS in a mobile scenario with heterogeneous services and resource limits. The proposed Cyclic Deep Q-network-based Edge Service Placement and Request Scheduling (CDSR) framework aims to find a long-term optimal solution despite future information unavailability. The key contributions include investigating a three-tier MEC network with vertical and horizontal cooperation, formulating the optimization problem using Markov Decision Processes, and proposing a DRL-based framework to decouple Edge service placement and request scheduling decisions. State representation contains information including the availability of services, the location of end users, and the current load on Edge servers.

Finally, the paper [56] addresses challenges in Edge Computing caused by the mobility of end devices (vehicles) that leads to QoS degradation and interruptions in Edge services. The paper proposes a framework for joint optimization of service migration and resource management in Edge Computing using reinforcement learning. Both proactive and reactive migration is considered, while Multi-Armed Bandit (MAB) methods are presented for solving the optimization problem. The proposed framework is considering average latency and energy consumption as a KPIs to make migration decision.

Service migration in Edge Computing and its complement like context migration have been as well a study of several novel approaches beyond analysis of this study, like, e.g. blockchain-based migration using deep reinforcement learning [81, 80] or service and data compression-based migration [77].

4.1.3 Comparative analysis

Table 4.1 provides a summary of existing research work focusing on Edge services migration. It shows various approaches in terms of type of algorithms, among which majority works relies on machine-learning based solutions, what is seen as a tendency in recent years. However, still classical linear programming and various heuristic approaches are also applied. It is worth noting that none of listed research work comprehensively compares reinforcement learning and heuristic approaches, considering aspects such as pros, cons, design, performance complexity, and usage recommendations for telco operators, while it is a case of our work. The table also outlines main criteria for relocation/migration decision, what allows us to indicate the consensus based on a frequently used metrics. Additionally, table contains technology-related information, such as integration of proposed Edge solution with access mobile network (4/5/6G) and considering end-to-end procedural complexity. This enables us to assess completeness of the proposed solution. Finally, we examine the presence of cloud-native Edge infrastructure in the referenced works. It can easily answer the question, whether the work is theoretical, or more pragmatical including real-environment testing. This fact also specify whether the migration procedure itself was considered, and if feedback was applied into the design of decision systems, which is a key aspect of telco environment.

Additionally, it is important to note that the migration of Edge workloads/applications is also addressed as an issue of task offloading in Edge Computing, where workloads can be offloaded between UE and Edge servers depending on the scenario. Several frameworks have been proposed in this context [59][60][26][97]. Since Edge Relocation is different life-cycle management operation that primarily handles migration between Edge server instances, and offloading is not a key topic of thesis, we do not include it into our analysis.

Table 4.1: Comparison of Edge Relocation methods

paper	**type of algorithm**	**metrics**	**integration with mobile network (4/5/6 G)**	**cloud-native edge infra**
[21]	linear integer programming	user location (mobility)	only mentioned no integration	only mentioned no implementation
[94]	binary linear integer programming	distance (with respect to latency)	No	only mentioned no implementation
[86]	Lyapunov optimization	network delay data transmission delay bandwidth usage energy consumption	only mentioned no integration	No
[57]	heuristic	resource usage (task dependency) user mobility.	No	No
[89]	Deep Q-Learning (DQL)	communication delay power consumption mobile users movement	No	Yes, based on Docker
[96]	deep RL	number of UEs attached to the eNB (that MEC Host is associated with) location of the UE	integration with 4G simulator	No
[43]	deep RL	CPU (cores, utilization, speed) access bandwidth, data rate of servers access latency of servers power consumption of IoT device	No	No
[98]	Deep Q-Learning (DQL)	link traffic loads remaining edge node capacities	No	No
[87]	ML: Support Vector Regression (SVR)	CPU, Memory utilization latency, bandwidth	Yes	Yes
[33]	Cyclic DeepRL	user location current load on Edge servers	No	Yes
[56]	Multi-Armed Bandit (RL)	average latency energy consumption	No	No

4.2 Standardization enablers for Edge Relocation

In the context of Edge Relocation we also analyzed related work of ETSI (already described in Section 2.4) and 3GPP standardization. The 3GPP has introduced 3 types of providing Service and Session Continuity (SSC) in 5G network. As stated in [11] the most advanced and efficient for both user and network is SSC Mode 3, where the changes to the user plane are visible to the UE, but the network ensures no loss of connectivity. A new PDU Session Anchor is established before terminating the previous connection to guarantee service continuity. Satisfying SSC mode 3 has become one of our design principals.

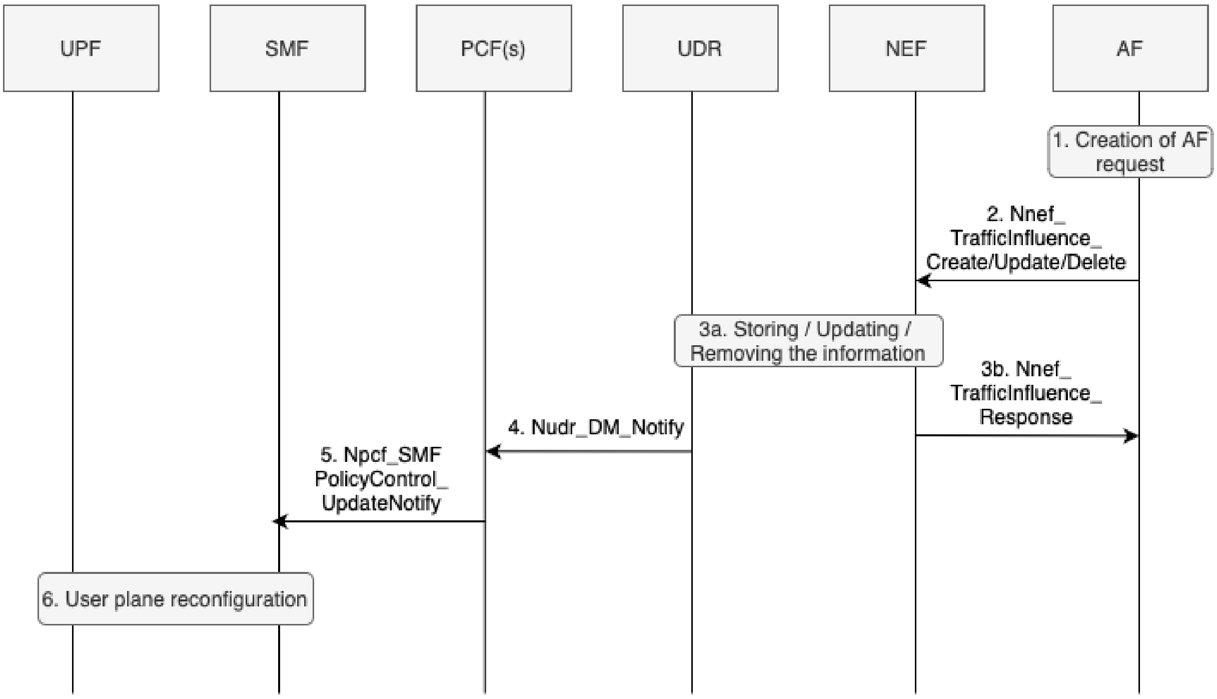

UPF
SMF
PCF(s)
UDR
NEF
AF
1. Creation of AF request
2. Nnef_ TrafficInfluence_ Create/Update/Delete
3a. Storing / Updating / Removing the information
3b. Nnef_ TrafficInfluence_ Response
4. Nudr_DM_Notify
5. Npcf_SMF PolicyControl_ UpdateNotify
6. User plane reconfiguration

Figure 4.1: "Application Function influence on traffic routing" procedure, based on [12]

Additionally, we identified another enabler, which is 3GPP standardized procedure "Application Function influence on traffic routing" that has been introduced in the Release 16 of "Procedures for the 5G System" [12]. The term Application Function is a general term for any function that can connect to the 5G Core message bus to communicate with all other NFs. We assume Edge Orchestrator to be an exemplary AF. Mentioned procedure enables external entities (first authorized through the Network Exposure Function), such as Edge Orchestrator to influence on traffic routing in 5G network data plane. Edge Orchestrator can influence on 5G Core to reestablish or modify existing PDU session(s) to route traffic towards new application instance, what ensures session continuity of end-to-end Edge Relocation procedure. As shown in Figure 4.1, after the NEF authentication, the message is transmitted through 5G Data repository (URD). Proper rules are created within PCF and next transmitted to Session Management Function to update traffic rules accordingly in UPFs. We utilized the concept of "Application Function influence on traffic routing" in the design process of our Edge Relocation procedure.

Next, we identified the 5G system's capability to expose network function capabilities to external AFs, as defined in [11]. As stated in 3GPP document, "The Network Exposure Function (NEF) supports external exposure of capabilities of network functions. External exposure can be categorized as Monitoring capability, Provisioning capability, Policy/Charging capability and Analytics reporting capability. The Monitoring capability is for monitoring of specific event for UE in 5G system and making such monitoring events information available for external exposure via the NEF". We took profit from this capability of 5G standard for the Edge Relocation case. The Edge Orchestrator is subscribed for AMF and SMF events, such as UE mobility events and Incoming Handover events, what triggers initiation of relocation decision process. Furthermore, if 5G system would be able to measure end-to-end user latency in communication with data network (Edge Host), this information can also be transmitted from 5G Core towards Edge Orchestrator.

The capabilities defined by 3GPP that were discovered we found really useful for the Edge Relocation use case, since an external Application Function (like Edge Orchestrator) now has the ability to directly interact with 5G Core, what is a key enabler for integration two independent architectures of MEC and 5G network. It allows as well to design end-to-end procedures for Edge Computing.

4.3 Conclusions

The problem of Edge Relocation has been already identified by several research work. However, no one has yet considered an end-to-end procedure that encompasses two integrated architectures: ETSI-based MEC architecture and 3GPP-based 5G system. Several papers deal with decision-making regarding relocation using different techniques. Our work focus on a comparison between different approaches for solving the Application Relocation problem for Multi-Access Edge Computing, specifically heuristic and machine learning approaches. This is not the case for any existing paper. Identifying the most relevant data from both infrastructure metrics coming from the MEC system and end-user position and latency measurement considered from the 5G network side is as well not a case for mentioned papers. Our work deals with a detailed perspective on the collecting needed data aligned with standards. Last but not least, our Application Relocation process was validated in a PoC environment, including a cloud-native Kubernetes-based Edge infrastructure, a multi-cluster application manager, and containerized Edge services which is rare in existing works. Finally, since the beginning, we have designed the system to be aligned with the 5G system procedures described in 3GPP standards.

Chapter 5

Architecture and Industrial PoC

This chapter presents the first contribution of the PhD thesis. It primarily describes the results obtained in a cooperation with Orange Innovation Poland emphasizing the industrial mode of PhD process, adopting practical implementation in a form of the proof of concept (PoC). The main goals to obtain were set as follows:

- Developing the architecture of an integrated Edge-enabled 5G system.
- Managing and orchestrating multiple distributed Edge Hosts based on pre-commercial EMCO (Edge-Multi Cluster Orchestrator) system [4].
- Implementing seamless relocation of Edge Applications across Edge Hosts to provide missing functionality identified as a key gap to industrialize and implement the EMCO system into operational network.
- Building a PoC based on the given EMCO-managed Edge-enabled 5G system to confirm operationalization of proposed Edge Relocation method.

All of the above-mentioned goals were recognized as significant industry challenges for telecom operators to provide a resilient and high-performing Edge system, while ensuring support for session and service continuity in the context of enhanced user mobility. The design and implementation considered all architectural and procedural recommendations from standardization bodies such as ETSI and 3GPP.

5.1 Edge-enabled 5G system architecture

The proposed architecture of Edge-enabled 5G system is illustrated in Figure 5.1. This architecture leverages 3GPP 5G network elements and the proposed Cloud-Native Edge System to establish a versatile Edge-enabled 5G system architecture. The integration was done following the recommendations outlined in ETSI and 3GPP architectures and deployment models [8][10], as previously discussed in Section 2.4.

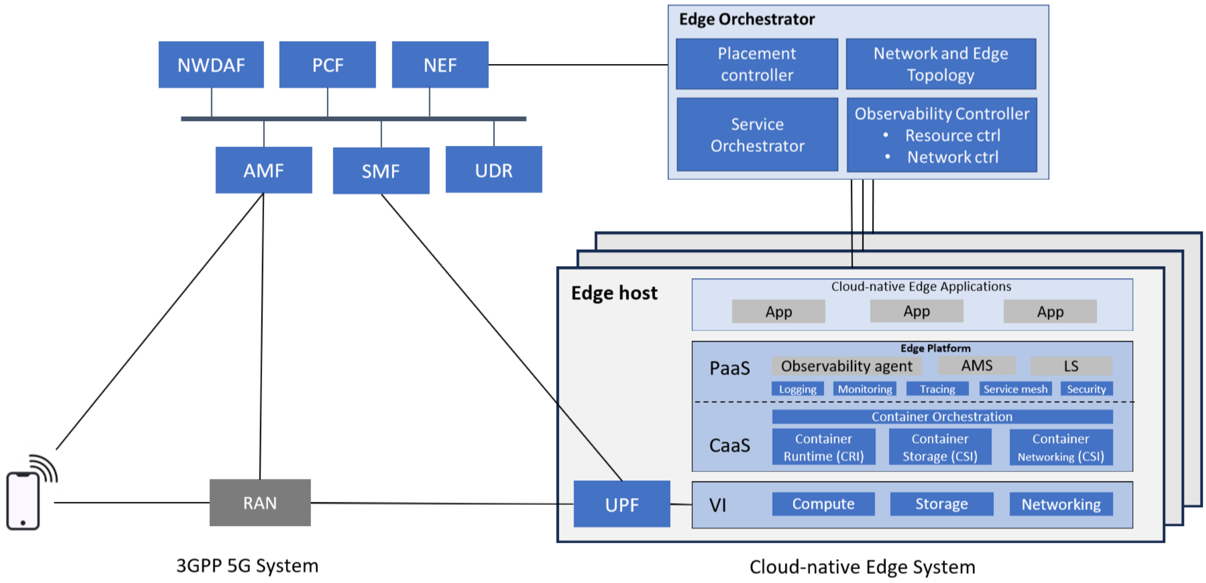


Figure 5.1: Architecture of the Edge-enabled 5G system

The functional entities of the architecture are divided into two groups:

- Edge-level entities enabling the lifecycle management of Edge Applications and specifically performing the Edge Relocation process.
- 5G network - level entities corresponding to the network functions deployed in the Radio Access and Core Networks (i.e. RAN and 5G Core). In the following, we highlight only the main network functions which are directly involved in the Edge Relocation process.

Edge-level entities: the cloud-native Edge System consists of an Edge Orchestrator and one or more Edge Hosts.

Edge Orchestrator is responsible for the life-cycle management of Edge Applications and the Edge platform itself. Specifically, the orchestrator manages application's state, keeps the local topology of Edge Hosts and takes Edge Relocation decisions. Orchestrator is hosted by the 5G Core network and treated as an Application Function (AF). If the Edge Orchestrator is considered as trusted by the Operator, it is, then, allowed to interact directly with the relevant 5G Core Network Functions (PCF, AMF and SMF). Otherwise, it communicates only with the NEF function which handles communication with the network functions. The orchestrator requires to retrieve various parameters from 5G Core network functions (e.g. SMF, AMF), such as UE mobility events, PDU session events, QoS parameters to be able to make the Edge Relocation decision. Then, the orchestrator can influence UE's traffic routing by interacting with the PCF and SMF to provide information on the new desired state.

The Edge Host consists of i) the Virtualization Infrastructure (VI), which offers compute, storage, and network resources to the Edge Applications and ii) the Edge platform which provides several services to ensure an efficient deployment of Edge Applications on Edge Hosts. Among mentioned services there are: `Observability agent` which monitors infrastructure and informs Edge Orchestrator about current resources utilization, while supporting orchestration/placement decisions. The `Application Mobility Service (AMS)` supports application data/context synchronization between multiple Edge Application instances run on different Edge Hosts.

The Edge Applications are cloud-native [55]. They are implemented while making use of i) stateless architecture, ii) microservices and iii) containers technology. They are disaggregated into a set of small individual services, where each one is packaged and running in its own container.

To provide a cloud-native environment, the Edge Platform relies on two layers: Container as a Service (CaaS) and Platform as a Service (PaaS) [14]. The CaaS offers a complete framework for deploying and managing container based Edge Applications. It incorporates several interface plugins such Container Network Interface (CNI), Container Runtime Interface (CRI) and Container Storage Interface (CSI) to support multiple implementations. The PaaS hosts the Edge services including the Observability Controller, Topology Controller. Besides, it provides the set of tools and common services which can be required by Edge Applications and services such as service mesh to expose various traffic capabilities (e.g., telemetry, policy, etc.), monitoring, logging and tracing to ensure the observability of Edge Applications, security, etc.

5G network-level entities were already introduced in Section 2.3. Here, we have included additional information specific only to Edge Relocation.

Both SMF and AMF can provide several Key Performance Indicator (KPI) metrics and end-user mobility events in order to derive statistical information about the UE mobility, and generate predictive information about future events. SMF provides information related to the UE handover and session state, while AMF provides more detailed information about user mobility and QoS degradation. Mentioned metrics can be passed to the entity responsible for determining wheter to relocate end users' application or not.

Optionally, we have considered a Network Data Analytics Function (NWDAF) as a native 5G Core service for data collection and analytics. It retrieves data gathered from: i) other 5G Core NFs, ii) applications, and iii) UE, so that it processes and ensures data analytics using defined algorithms [46]. In doing so, it can trigger or suggest actions when necessary. In consequence, an Edge Relocation decision can be performed, achieving a guaranteed UE's QoS. The NWDAF is subscribed for user mobility events, QoS indicators and other metrics exposed by SMF/AMF. The data analytics module of NWDAF constantly analyses gathered data and monitors defined connection parameters in order to detect given patterns.

In the designed and implemented architecture, we have decided to shift the responsibility for data collection, aggregation, analysis, and ultimately the decision-making process for relocation to the Edge Orchestrator. It is more in line for the Edge Orchestrator to make decisions regarding the selection of the best Edge Host for relocation Edge Applications within the Edge system. This approach avoids overburdening the 5G system with the task of making decisions for each specific application deployed within the Edge System. As a consequence the 5G system's role is limited to two main functions: a) gathering information about UE (User Equipment) mobility and b) reconfiguration data plane as a part of the Edge Relocation procedure.

Aligning more closely with the designed system, the Edge Orchestrator is now tasked with making decisions regarding the selection of the Edge Host for relocating an Edge Applications within the Edge system.

5.2 Edge Relocation

As depicted in Figure 5.2, in an Edge-enabled 5G system, the Edge Relocation refers to the capability of relocate a running Edge Application instance (and user context, in case of a stateful applications) from one source Edge Host to a target Edge Host to deal with QoS degradation.

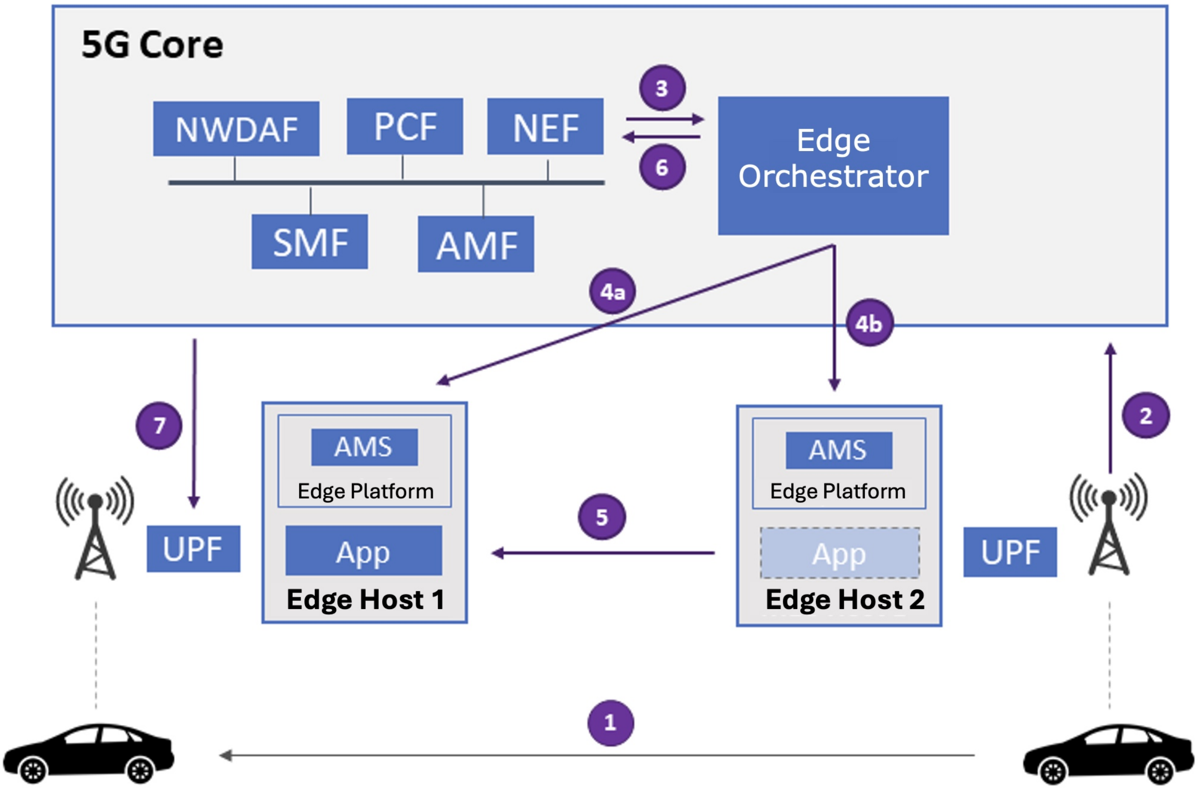


Figure 5.2: Edge Relocation procedure

Besides, to ensure the service continuity of UEs, the Edge Applications which can be either statefull or stateless, should be considered during the Edge Relocation procedure [52]. In particular, stateless applications do not store any session-related information; consequently, no data synchronization is needed between instances during the Edge Relocation process. However, stateful applications store user or session related information. Hence, the replication and synchronization of this data across multiple Edge Hosts is crucial to ensure service continuity.

Scenario depicted in Figure 5.2, covers an Edge Relocation procedure triggered to maintain UE service continuity. In this scenario, the UE changes both its location area and its attached RAN. Once a radio handover procedure is triggered, the new attached RAN (i.e., target RAN) will transmit data throughout a UPF to a target Edge Host.

As illustrated in Figure 5.2, scenario can be divided into three main phases: i) mobility detection ii) application relocation and iii) data plane update. In particular:

- **Mobility detection**: Once the user changes its location and a handover is initiated (1), the base station sends a location update to the AMF which forwards it to the SMF (2). Next, SMF informs the MEO about user mobility and the handover trigger (3).
- **Application relocation**: The Edge Orchestrator takes the application relocation request, while jointly considering UE mobility information and the Edge Hosts topology, and then, optionally selects a target Edge Host (3). Next, Orchestrator informs the source and target Edge Hosts (4a, 4b) of the incoming relocation actions. The relocation process is performed (5) and if the application is stateful, the migration of the application's current state is assisted by the AMS of the source and target Edge Hosts. Then, the new application running on the target Edge Host is synchronized with the latest state, and hence, becomes ready to handle UE traffic. The Edge Orchestrator confirms to the SMF that the Edge Relocation procedure on the 'Edge side' has been completed (6).
- **Data plane update**: Once the finalization of the Edge Application relocation on the Edge side is completed, SMF can complete the handover. SMF assisted by PCF takes in charge i) the reselection of a new UPF through which the traffic will be routed to the target Edge Host and ii) the re-establishment or modification of the PDU session (7).

5.3 Edge Relocation Procedure in Edge-enabled 5G system

This section aims at detailed description of Edge Relocation procedure in integrated 5G network and Edge Computing System.

As illustrated in Figure 5.3, the Edge Relocation of an Edge Application performs as follows. First, the Edge Orchestrator (EO) pre-subscribes to the SMF for User Plane path management events notification. Once a handover is initiated, the base station sends a location update to the AMF (0a) which forwards it to the SMF (0b). Second, as soon as the notification trigger is met, the SMF informs the Edge Orchestrator about the user mobility and the handover trigger. This message is sent through NEF (1) and it is communicated to the EO (2). The SMF performs operations in a sequential mode. It allows, first, the finalization of the Edge Application relocation on the Edge side and, then, the completion of the handover once the confirmation is received. In the next step, the EO recognizes the application handling the UE session. The EO takes the application relocation checks request, while considering the UE mobility information and the Edge Hosts topology, and then, optionally selects a target Edge Host (3). The positive decision triggers the main part of the Edge Relocation process. The EO checks that the Edge Application is not already running on the target Edge Host. If it is the case, it deploys it (4). If the application is statefull, the EO informs the source Edge Host (5) and then asks the AMS (6) to prepare and transfer the application's current state to the target Edge Host. It is worth noting that the AMS of the source Edge Host communicates directly with the AMS of target Edge Host. Next, the target Edge Host confirms the reception of the application state to the EO (7). Then, the application running on the target Edge Host is synchronized with the latest state, and hence, becomes ready to handle the UE traffic. The EO confirms to the SMF that the Edge Relocation procedure at the Edge side has been completed by sending a traffic influence request to reselect the UPF. This request goes through the NEF (8), is stored by the local UDR (9) and transmitted to the PCF. The latter formulates new policy rules (10), which are applied to the SMF (11). The SMF takes in charge i) the reselection of a new UPF though which the traffic will be routed to the target Edge Host and ii) the re-establishment of the PDU session (12). Finally, the EO checks whether the application is still needed on the source Edge Host. If it is the case, the EO triggers the application's uninstall and release resources (13).

5.4 Demonstrator perspective - Proof of Concept

In this section, we give insights on our `5G-Edge Relocator`, an innovative PoC framework for Edge Application relocation. `5G-Edge Relocator` leverages Kubernetes [3] and Edge Multi-Cluster Orchestrator (EMCO) [4].

Let's recall that Kubernetes is an open-source container orchestration platform that automates the deployment, scaling, and management of applications across clusters of hosts [2]. It provides a robust system for organizing and running containerized applications, enabling efficient resource utilization and high availability.

EMCO is also an open-source project for intent-based deployment of cloud-native applications [51] to a set of Kubernetes clusters spanning numerous edge locations. It aims to simplify

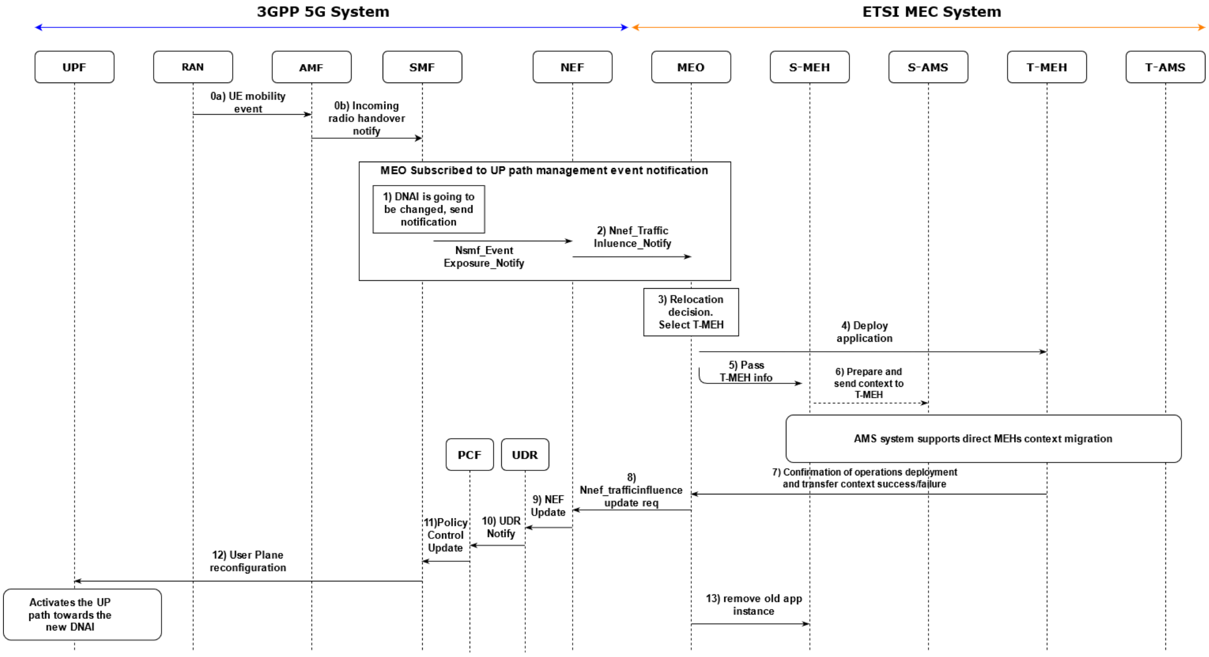

3GPP 5G System
ETSI MEC System
UPF
RAN
AMF
SMF
NEF
MEO
S-MEH
S-AMS
T-MEH
T-AMS
0a) UE mobility event
0b) Incoming radio handover notify
MEO Subscribed to UP path management event notification
1) DNAI is going to be changed, send notification
Nsmf_Event Exposure_Notify
2) Nnef_Traffic Inluence_Notify
3) Relocation decision. Select T-MEH
4) Deploy application
5) Pass T-MEH info
6) Prepare and send context to T-MEH
AMS system supports direct MEHs context migration
PCF
UDR
7) Confirmation of operations deployment and transfer context success/failure
8) Nnef_trafficinfluence update req
9) NEF Update
10) UDR Notify
11)Policy Control Update
12) User Plane reconfiguration
Activates the UP path towards the new DNAI
13) remove old app instance

Figure 5.3: Edge Relocation workflow for intra MNO - Scenario #3

the deployment and lifecycle management of distributed applications across multiple Edge clusters (this is the implementation name of Edge Hosts), enabling efficient resource utilization and improved scalability at the Network Edge.

Both solutions have been selected as a pre-commercial solutions for Orange to manage Edge infrastructure and to orchestrate and monitor of Edge Applications. The aim of this contribution is three fold: i) to integrate EMCO with multiple Edge clusters, while validating EMCO functionalities. Next, ii) the implementation of missing procedure of seamless Edge Application relocation, and iii) finally integration with 5G Network and validation of implemented procedure.

5.4.1 `5G-Edge Relocator` Framework

`5G-Edge Relocator` is a proposed modular framework coupling Edge and 5G network to execute the relocation of Edge Applications on geo-distributed Kubernetes clusters (Edge Hosts). To achieve its goal, our framework relies on an Edge-enabled 5G system to allow performing an end-to-end procedure of Edge Application relocation on the top of the proposed system. `5G-Edge Relocator` extends EMCO to ensure the relocation of applications while jointly considering their requirements and the underlying Edge infrastructure status.

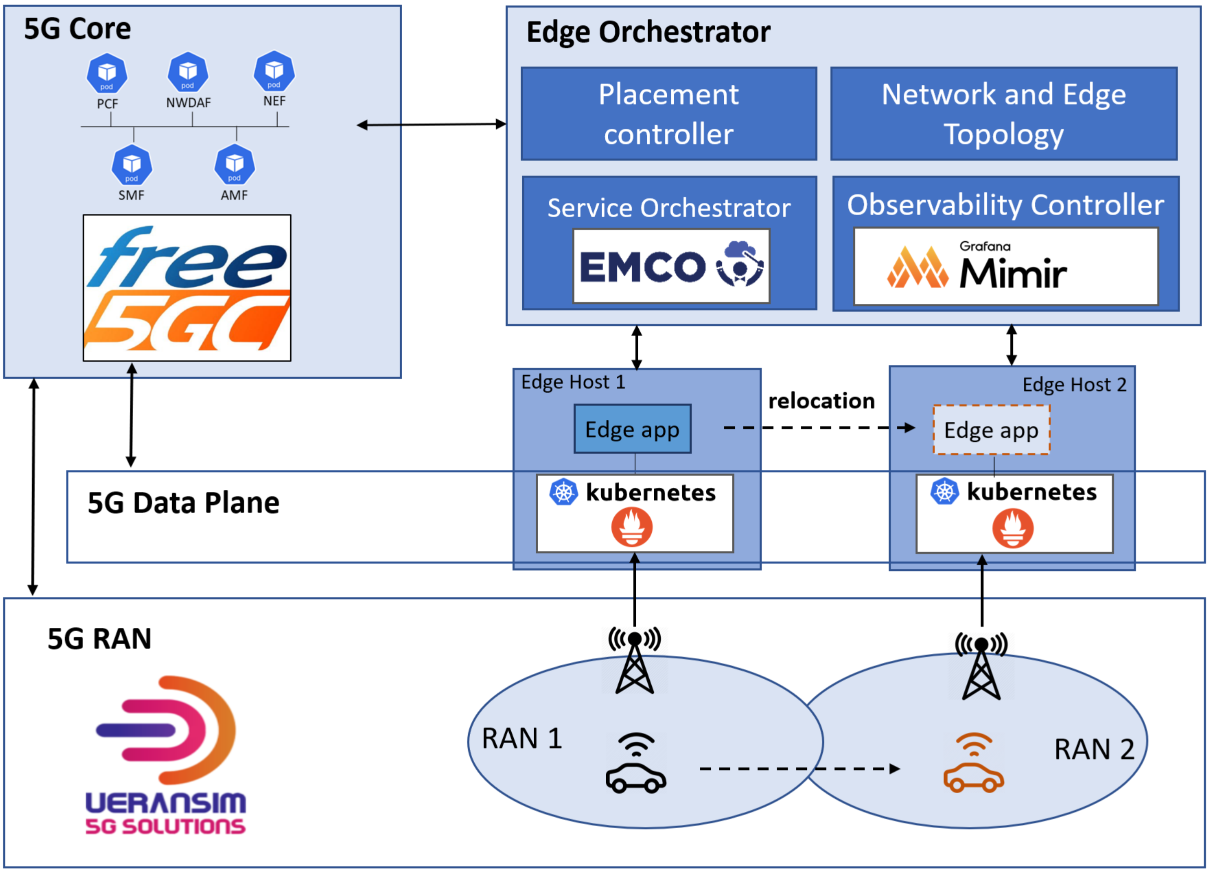


Figure 5.4: 5G-Edge Relocator framework: Technology mapping

`5G-Edge Relocator` relies on the following building blocks:

- ***5G system*** includes 5G control plane, 5G data plane and Radio Access Network (RAN). It enables users to reach applications located at Edge Hosts. The 5G RAN and data plane are responsible for transmitting data to the proper Edge Application, while 5G control plane is in charge of managing and control the data plane. For implementation of 5G system we used Towards5GS project that is containerized version of two projects: Free5GC - an open-source implementation of 5G Control Plane and UERANSIM - an simulator of Access Network integrated with simulator of end user.

- ***Edge Orchestrator*** - the cornerstone of our architecture. It acts as a management plane over the set of Edge Hosts. Its main responsibility is to manage the life cycle of applications located at Edge Hosts, including instantiation, scaling, healing and relocation between Edge Hosts, to cite few.

To achieve its goals, *Edge Orchestrator* relies on the following blocks:

Service Orchestrator - a key controller of the Edge Orchestrator. It is in charge of interacting with Edge Hosts to execute LCM (Life Cycle Management) operations directly on Edge Applications. This functional component has been represented by EMCO.

Placement Controller is empowered with an effective algorithm to find an appropriate Edge Host based on (i) various constraints related either to the end user or the application and (ii) information retrieved from the topology components, such as the Edge Host load. This component is more described in the next section related to heuristic decision algorithm. This component has been developed in the scope of this PhD thesis as an extension to EMCO.

Edge-network Topology represents, as depicted in Figure 6.1, jointly the Edge Hosts' topology coupled to the cells' topology, the whole completed by the connectivity between them. This component as well has been developed internally as a extension to EMCO. The Edge topology is organized into three levels: City, Regional and International and more describe in next section 6. The Edge-Network topology is fed by the Observability Controller with up-to-date information about network performance, such as latency at links, as well as state of used and available computing resources at Edge Hosts.

Observability Controller encompasses two sub-controllers called network performance controller and resource controller. i) The network performance controller observes network-related metrics such as latencies at Edge-Network Topology links. Note that the Edge-Network Topology and Observability Controller can provide the shortest path between two given nodes of the topology based on Dijkstra's algorithm, when requested by the Placement Controller. ii) While the resource controller provides real-time measurements of current utilization of Edge cluster computing resources (CPU and Memory). Resource controller was implemented based on centralized Grafana Mimir solution, that is subscribed for resource load changes at each of distributed Edge clusters where Prometheus agents are responsible for exporting such data.

- ***Edge Host*** corresponds to the virtualized environment hosting Edge Applications. It has the capability to offload data to the destination Edge Applications. Moreover, it allows to route data to other Edge Hosts [8]. Edge Hosts were implemented using Kubernetes clusters, where a single, one-node cluster represents one Edge Host. Starting from this place we use "Edge Cluster" name for Edge Host in terms of implementation use. Kubernetes and its accompanying CNCF (Cloud-native Computing Foundation) projects become de-facto a standard for delivering an Edge cloud-native infrastructure (Rancher, ClusterAPI) and management of workloads (docker, containerd), as described in following papers [44, 102, 37, 62].

5.4.2 Edge Relocation workflow

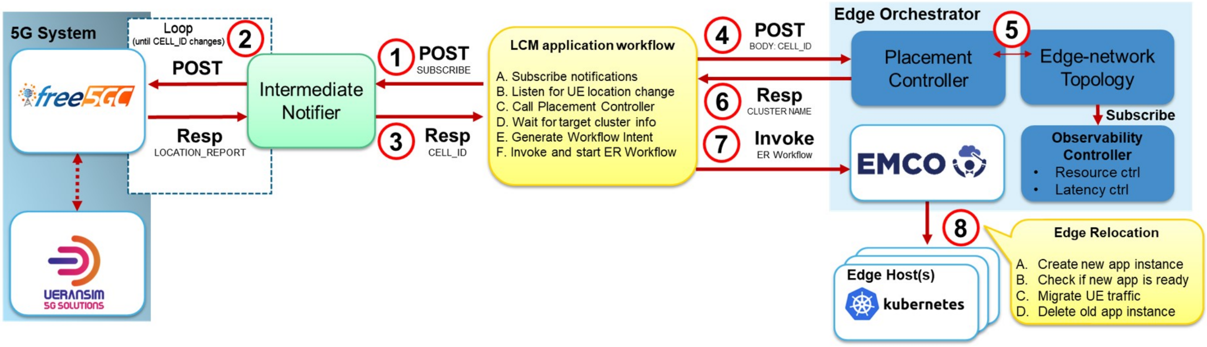


Figure 5.5: 5G-Edge Relocator workflow

Once a new Edge Application is deployed, a dedicated LCM workflow is started. First, the workflow is subscribing to an `Intermediate notifier` (Figure 5.5, **step 1**), which allows to subscribe for user mobility events (e.g., handover) directly from the Access and Mobility Management Function (AMF). Once the AMF is detecting an end user handover, a location report with the destination network cell identifier is passed to the LCM workflow (**steps 2-3**). This information is next passed to the Placement Controller so that it selects the new Edge Host to host the application (**step 4**). The Placement Controller runs the algorithm to identify the Edge Host based on (i) the new end user position, (ii) the application requirements and (iii) the Edge-Network topology status (**step 5**). If a new Edge Host has been identified, this information is returned to the LCM workflow (**step 6**). Then, a new relocation intent for a given application is prepared and provided to the Edge Orchestrator (**step 7**). The latter triggers the application relocation procedure (**step 8**).

5.4.3 Application relocation execution

The procedure of application relocation has been designed in a way to ensure the service continuity. Based on the relocation intent, the relocation workflow is triggered. This intent includes the information required to successfully execute the operation, such as the application identifier and the destination Edge Host. At the beginning, the new application instance is created, while the old one is maintained until it becomes safe to delete it. Subsequentially, the process checks the readiness of the new application instance leveraging the monitor agent located at each Edge Host. Once, the new application is ready, a DNS system recovers the new application instance and updates the DNS entry. In doing so, the end user traffic will be sent to the new application instance. Finally, the old instance of the application is deleted, which allows resources saving on limited Edge Hosts. The procedure of containerized application relocation from origin Edge Host (Kubernetes cluster) to target one using top-level EMCO orchestrator has been entirely designed, implemented and upstreamed to open-source EMCO project [4], as a response to the industrial needs and telco operator requirements for Edge Application orchestrator.

5.4.4 Experimental environment

Our experimental platform relies on four R630 DELL servers with a pre-installed cloud manager system - OpenStack Ocata. On the top, we have created 28 VMs with Ubuntu 20.04 LTS image to host Kubernetes clusters. Then, we used *kubeadm* in order to deploy 25 Edge clusters on the top of previously created VMs. 3 management clusters characterized by 1 master and 1 worker (4vCPU and 4GB RAM each) and 22 workload clusters characterized by one single node playing jointly the role of master and worker. The management clusters have been dedicated to host `5G-Edge Relocator` components as depicted in Figure 5.5 while the workload clusters host Edge Applications. Note that we are considering an Edge-Network topology composed of only one coverage zone (as marked in red in figure 6.1). Among the workload clusters, (i) 16 belong to the City-level with a defined capacity of 4GB RAM and 4 vCPU, (ii) 5 belong to the Regional-level characterized by 8GB RAM and 8 vCPU, and (iii) 1 belongs to the International-level with 12GB RAM and 12 vCPU.

As described in Section 5.4.1, `5G-Edge Relocator` is a modular framework relying on micro-service based components. Some of them are fully implemented and others are adapted from open source projects. Specifically, the Placement Controller, Edge-network topology, Observability Controller (based on Prometheus tool [9]), Intermediate Notifier are implemented using GO language. The aforementioned components are enriched with various open source projects: Kubernetes version 1.19.0, EMCO version 22.06 [4] and *Towards5GS* [5] projects. We recall that Kubernetes is a cloud-native application orchestrator. Its clusters correspond to the Edge Hosts of our framework. EMCO is an Edge Application orchestrator enabling the deployment of applications in a multi-cluster infrastructure. *Towards5GS* is our open source Kubernetes-based implementation of a 5G system. Further details about the implementation of 5G service can be found in our previous work [47].

5.5 Conclusions

The proposed Edge-enabled 5G system allows us to define an end-to-end Edge Relocation procedure, including the interactions between the 5G control and data plane and Edge system. We integrated and evaluated the open-source solutions for cloud-native infrastructure and workloads management: EMCO and Kubernetes, both of which are Orange pre-commercial solutions. The performed functional evaluation has been described in magazine paper [74]. Next, we designed and implemented the lacking procedure of seamless migration of containerized applications across multiple Edge Hosts (Kubernetes clusters), and upstreamed our code to EMCO open-source repository. Lastly, we built Edge-Relocator Proof of Concept Framework (demonstrator), integrating additionally open-source projects: Free5GC, UERANSiM, and Prometheus. It allows us to perform end-to-end functional validation of the proposed procedure in 5G-network environment.

The above-mentioned efforts have helped Orange Poland to industrialize Edge Computing technology by:

- possibility of commercializing and industrializing the pre-commercial EMCO Orchestrator for the effective management of the distributed Edge Hosts infrastructure.
- reusing the implemented Edge Relocation procedure to support SSC in Edge Computing.
- reusing additional enhancements into EMCO, including Edge Topology registry and Placement Controller.

Additionally, considering the complexity of the targeted challenge, which involves building an algorithm for selecting an Edge Host to relocate Edge Applications, we have acknowledged that executing full-scale performance evaluations of proposed algorithms using the actual implementation of an Edge-Relocator Framework (demonstrator) might be inefficient. The real demonstrator faces several limitations such as: a) Topology scaling (limited infrastructure for constructing larger, multi-cluster Edge topologies), b) Convergence time (real application relocation consumes seconds, making performance execution tests lengthy). Given these limitations of the PoC implementation, we have decided to additionally develop an Edge Relocation simulator, as introduced in the next chapter.

Chapter 6

Edge Relocation: Problem modelling

This chapter provides a presentation of Edge Relocation problem modeling and the problem statement. We are going through: mathematical modeling of system elements, set of assumptions, requirements, constraints, and finally, objective functions. The proposed modelling is an introduction to the algorithms investigated in the subsequent part of the thesis. Based on the proposed modeling, we are presenting elements of Edge Relocation simulator and its assumptions.

6.1 Edge topology model

We model the Edge topology as an undirected graph G=(N,V) as depicted in Figure 6.1. N represents the Edge Hosts and V the links between them. Each node $i \in N$ is characterized by its (i) CPU capacity, Cap, (ii) available CPU, CPU_i, (iii) Memory capacity, M (iv) available memory, Mem_i and, (v) a cost μ_i depending on its level. Each link $l \in V$ is characterized by latency $\psi(l)$. It is worth noting that Edge Hosts are in charge of hosting Edge applications and forwarding traffic to others Edge Hosts [8].

We distinguish between three node classifications:

- Levels: each Edge Host belongs to a j level. We recall that a level could be a City-level ($j = 1$), Regional-level ($j = 2$), or International-level ($j = 3$).
- Zones: Edge Hosts at the same level are grouped into Z zones (e.g., Warsaw, Gdansk, etc.) depending on their geo-locations. We assume that all Edge Hosts belonging to the same zone

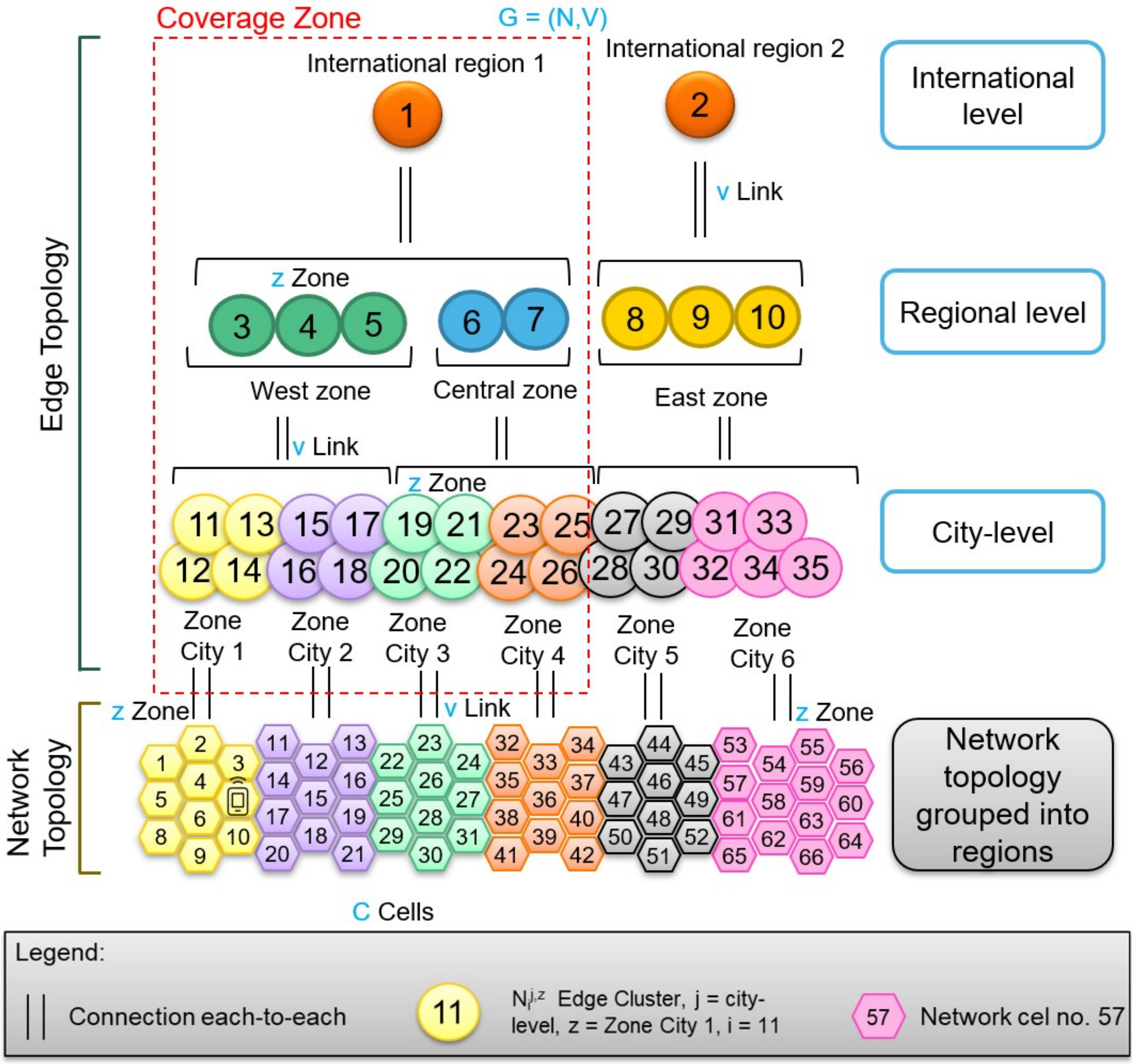


Figure 6.1: Network and Edge Topology model

are directly connected (e.g., Edge Host 13 is connected to Edge Hosts 11, 12 and 14). In addition, we assume that they are connected to their neighbors belonging to a different zone within the same level (e.g., at City-level, Edge Host 13 is connected to Edge Hosts: 15 and 16). The links between levels are connecting Edge Hosts belonging to the same coverage zone with the respect to their levels' order. As depicted in Figure 6.1, the City-level is connected to the Regional-level and the Regional-level is connected to the International-level.

A City-level is directly controlling a set of cells located within the same geographic area. Recursively, Edge Hosts at level j are controlled by the ones at levels $\{l \mid l > j\}$.

- Coverage zones: A coverage zone corresponds to a hierarchy of zones belonging to different levels. We call such as subset of Edge Hosts $K|K \subset N$.

The Edge topology G is augmented by a Mobile Network Topology, $G_a(N_a, V_a)$. N_a corresponds to the set of cell nodes. V_a corresponds to the links between $n_a \in N_a$ and $n_b \in N$. Note that v_a^b exists only if the Edge Host n_b belongs to the City zone serving the cell n_a.
Each link $l \in V_a$ is characterized by its latency $\psi(l)$.

6.2 Edge application

As aforementioned, Edge Hosts are able to host one or more dedicated Edge applications. Dedicated application is serving to the single user. Each application a is characterized by its requirements: CPU, memory and latency denoted $req\_CPU(a)$, $req\_Mem(a)$, and $req\_\psi(a)$, respectively. We assume that an end user is characterized by a communication, $e \in C$ between itself and the Edge Host hosting its application. Note that for simplification purposes, we consider that a communication is between a cell c, where the end user is located and the Edge Host hosting its application.

6.3 Problem statement

We aim to find the appropriate Edge Host to relocate a given application while respecting its requirements. For such a problem, we are considering the following constraints:

- Placement constraint: The placement decision is modelled by a binary variable $x_{a,i}$, where $x_{a,i}$ is fixed to 1 if the Edge application a is placed at the Edge Host $i \in K$, and 0 otherwise. This constraint is formally defined as follows:

$$\sum_{i=1}^{k} x_{a,i} = 1 \tag{6.1}$$

- Resources and load constraint:

  The available resources $CPU(i)$ and $Mem(i)$ of the selected Edge Host $i \in K$ has to be greater than the requested resources by the Edge Application a. Also, the used resources by the hosted applications should not exceed a certain threshold X_i in order to always leave an amount of computation power for critical services such as emergency services. To achieve that, we need to deduct t_i amount of resources such as $X_i = Cap_i - t_i$, where Cap_i represents the capacity of the node i in terms of the aforementioned resource. We assume that τ corresponds to the maximum of rate of resource usage which is expressed as follows: $\tau = \frac{X_i}{Cap_i}$. Note that for simplification purposes, we express X_i for the CPU. But it will be the same reasoning for the Memory. Hence, we define the following constraints:

$$\begin{cases} \sum_{i=1}^{k} x_{a,i}.req\_CPU(a) \leq CPU_i - t_i \\ \sum_{i=1}^{k} x_{a,i}.req\_Mem(a) \leq Mem_i - t_i \end{cases} \tag{6.2}$$

- Latency constraint: Each application has defined end-to-end latency constraint that has to be guaranteed by placing the application in an Edge Host, where the path to reach it is below the end-to-end latency constraint assured for the application. Lets define the following notations:
  - P_{ij} expresses the set of paths from a node c (i.e., cell) to another node (i.e., Edge Host) j, $(c, j) \in N_a \times K$.
  - P denotes the set of all admissible paths. Formally, $P = \bigcup_{\{c,j\} \in N_a \times K} P_{ij}$.
  - β_{ep} is a binary variable indicating whether a communication e passes through the path $p \in P$.
  - Δ_{lp} is a binary coefficient determining whether the physical link $l \in V \cup V_a$ belongs to the path $p \in P$ or not.
  - $(s_s(e), s_d(e)) \in N_a \times K$ denotes the source and destination of a communication e, $e \in C$.

  We assume that a communication e between an end user $s_s(e)$ and an Edge Host $s_d(e)$ is embedded in a physical path $p \in P_{cj}$ between the cell $c \in N_a$, where the end user is located and the Edge Host $j \in N$. Formally,

$$\sum_{p \in P_{cj}} \beta_{ep} = 1, \tag{6.3}$$

  A communication $e \in C$ must be hosted in a single path $p \in P_{cj}$. Such as $s_s(e) = c$ and $s_d(e)$ is hosted in a node $j \in N$. Formally,

$$\forall p \in P_{cj},\ \beta_{ep} \leq x_{aj} \tag{6.4}$$

Each physical path $p \in P$ is characterized by an end-to-end latency, $\psi(p)$. The latter corresponds to the sum of delays of its forming $l \in V \cup V_a$. Formally,

$$\psi(p) = \sum_{l \in V \cup V_a} \Delta_{lp} \times \psi(l), \forall p \in P \tag{6.5}$$

Finally, a communication $e \in C$ must be hosted in a path p, ensuring an end-to-end latency lower than that required by the application.

$$\sum_{p \in P} \psi(p) \times \beta_{ep} \leq req\_\psi(a),\ \forall e \in C \tag{6.6}$$

The aim is to find the Edge Host i to relocate an application a in order to satisfy application latency and resources constraints, as well as load-balance among Edge Hosts in the coverage zone K.

6.4 Objective function

Our aim is to minimize the total cost of Edge Application placement decision with the respect to the all defined constraints above. We formulated the objective function as follows:

$$min(\alpha \times \phi_l + \mu \times (\beta \times \phi_c + \sigma \times \phi_m)) \tag{6.7}$$

Where ϕ_l, ϕ_c, ϕ_m reflect the cost of the latency to reach Edge Host, selected CPU and Memory, respectively.

$$\begin{cases} \phi_c = \sum_{i=1}^{k} x_{a,i}(C_i - CPU_i) \\ \phi_m = \sum_{i=1}^{k} x_{a,i}(M_i - Mem_i) \\ \phi_l = \sum_{p \in P_{cj}} \beta_{ep} \times \psi(p) \end{cases} \tag{6.8}$$

α, β, σ represent weights for: end-to-end latency associated with Edge Host, its current load of CPU, and memory, respectively. These weights are defined according to the adopted strategy. μ is an additional weight associated to the physical infrastructure to differentiate Edge Hosts located at different levels (City, Regional, and International).

6.5 Edge Relocation Simulator

To perform an evaluation of proposed heuristic (section: 7) and reinforcement-learning (section: 8.1) algorithms the Edge-Enabled 5G network simulator was developed and is presented in this section. It has been designed to allow gauge effectiveness of proposed algorithms compared to other strategies.

The simulator shares the same implementation of some components from the demonstrator described in the previous chapter 5.4.1, such as: Placement Controller, Topology Controller. The main advantages of the simulator are possibilities to perform studies for more complex network topology: The Kubernetes clusters that represented Edge Hosts in the demonstrator have been replaced by data structures in the simulator.

Our initial performance evaluation of the heuristic solution was presented in [75]. The evaluation was done based on demonstrator that we introduced in section 5.4. Since we wanted to perform extensive experiments, including topology scaling, we have transformed demonstrator into simulator and perform more advanced experiments, described in next chapters.

6.5.1 Simulation model

The simulator consists of three components as depicted in Figure 6.2:

- End-User Simulator has replaced Free5G Core and UERANSIM. This components of the demonstrator represent multiple end-users that are connected to Edge infrastructure. The applied mobility model assumes that users can move across neighbour network cells with uniform distribution. The mobility model also assumes that user cannot return to the cell from which he last came.

- ERC: Edge Orchestrator has inherited all logical blocks. Placement Controller and Network and Edge Topology registry shares the same implementation with demonstrator. Observability Controller watches for resources utilization, however the difference with simulator is that it observes virtualized Edge Hosts while demonstrator watches real Kubernetes clusters. The service orchestrator has been developed from scratch as a replacement for EMCO and it's role is to instantiate, delete and relocate instances of Edge Hosts defined in Network and Edge Topology component.

- Network and Edge Topology (NMT) faithfully reproduces Kubernetes clusters, while providing unlimited possibility of scalling the number and capacity of Edge Hosts.

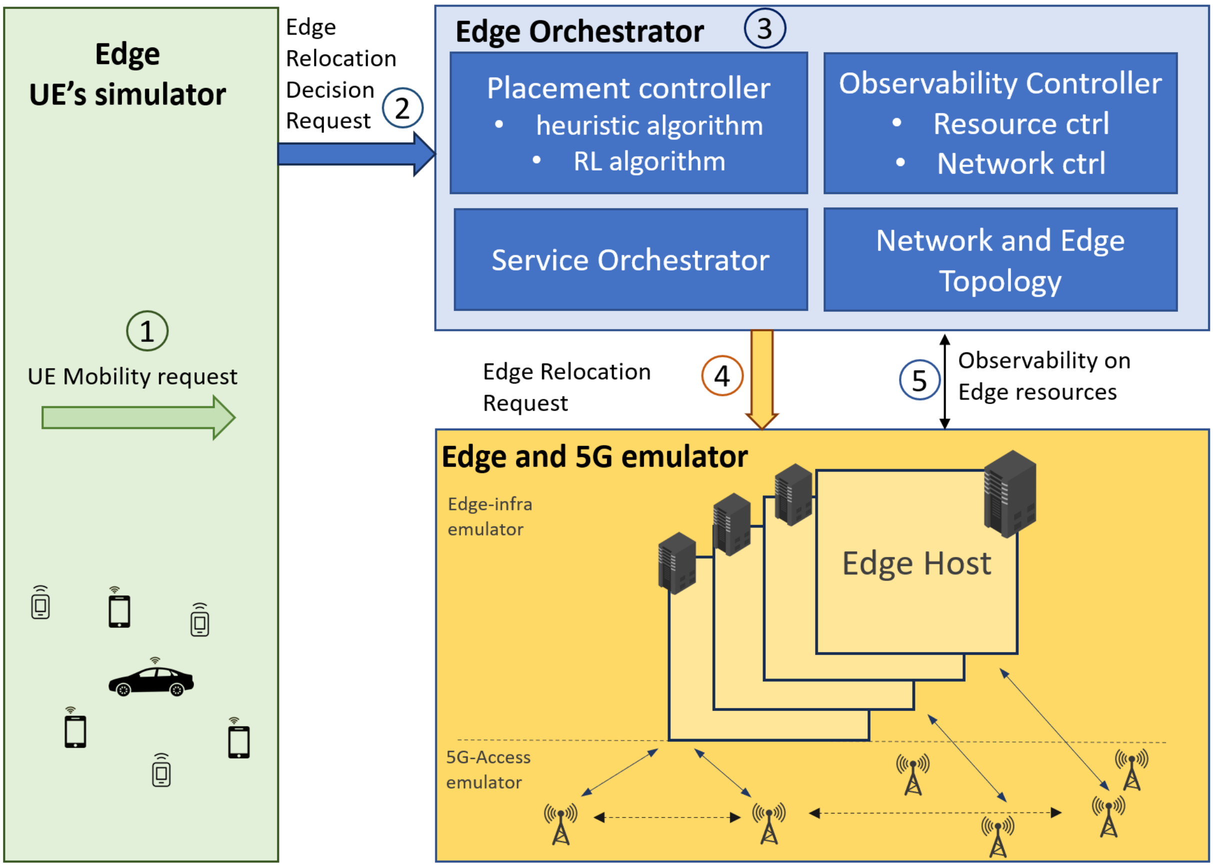


Figure 6.2: Edge Relocation simulator

The relocation cycle in simulator is presented as follows. First, Edge UE's Simulator, based on pre-defined mobility model selects next cell, where UE is moving (only neighbour cells are allowed, avoiding returns). The geographical movement of UE to the next cell triggers Edge Orchestrator to consider relocation of Edge Application. Placement Controller is checking with Observability Controller to determine the necessity of relocating UE's dedicated Edge Application. If so, the Service Orchestrator initiates relocation procedure in Edge and 5G emulator and migrate Edge Application, next updates resources utilization in NMT, what finalizes relocation procedure. After the relocation in NMT, Edge UE's simulator receives notification regarding the status of finalized operation.

Relocation requests for various UEs are iteratively executed for each tested algorithm multiple times in order to validate it's effectivnes according to the defined KPIs.

6.5.2 Latency modelling

One of the main challenges in selecting an Edge Host for hosting an application is to satisfy latency constraints. At this point it's worth recalling that end users can access any node of the Edge Host topology, as each Edge Host provides both computation and forwarding capabilities. The topology has been modeled as a graph, including two types of nodes: Edge Host nodes and mobile network cells. Each cell represents the user's current position in the network. The links allow to transfer data between end user and target Edge Host. Each link is characterized by latency. The end-to-end latency to reach the target Edge Host is the sum of the individual latencies on the links along the path as shown in Figure 6.4.

This section provides a zoomed perspective on a part of the topology, which is representative

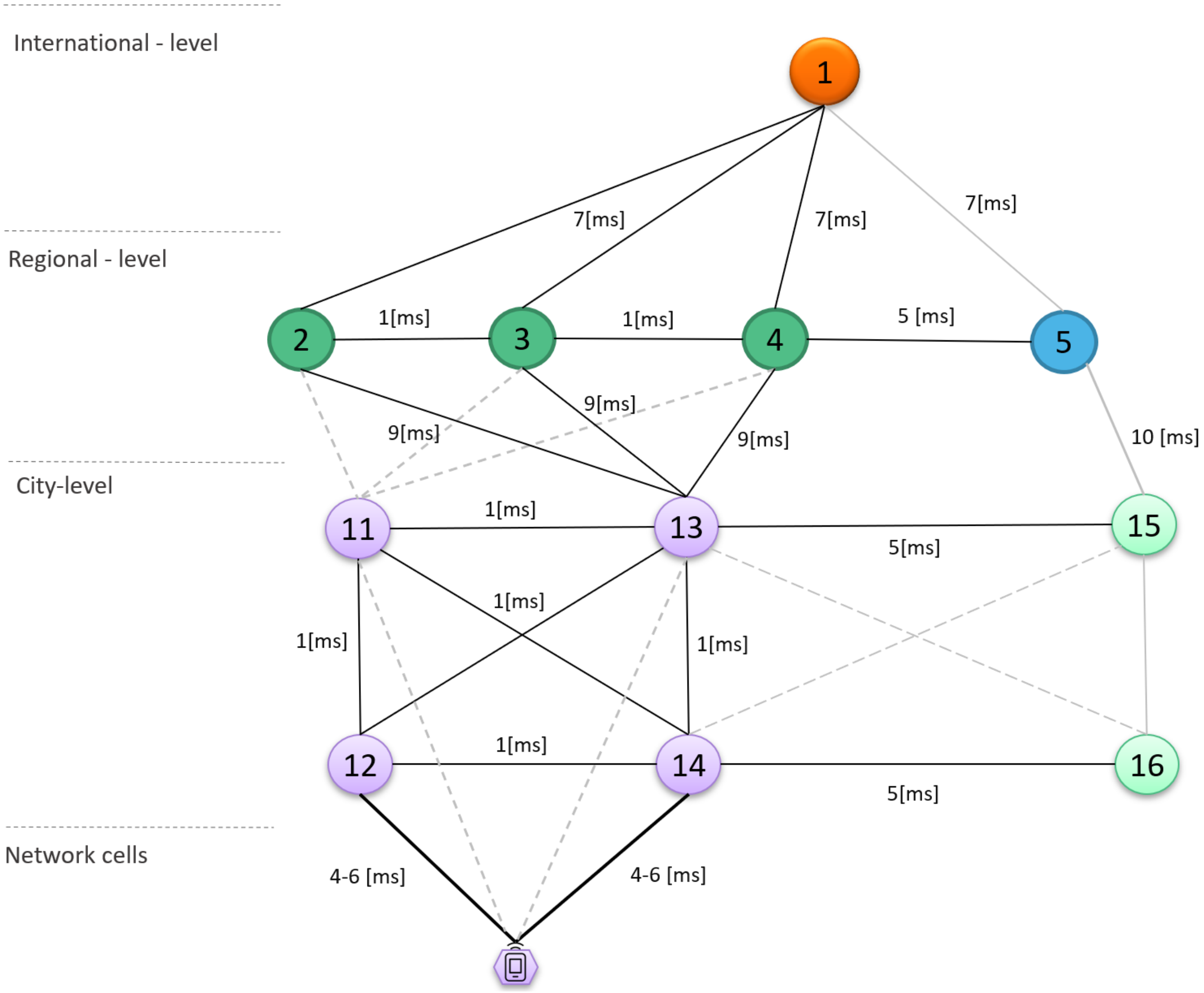


Figure 6.4: Latency modelling

of the entire structure. It demonstrates how the modeling of latency between Edge Hosts and end users has been designed. Additionally, the next chapter details how the algorithm searches for Edge Hosts and paths that meet the latency constraints.

Figure 6.4 provides detailed information about a part of the topology presented earlier in Figure 6.1. First, let's note that the latency between any network cell and City-level Edge Hosts (please recall that each network cell in the same zone is connected to each Edge Host in the corresponding City-level zone) is randomly selected at the beginning of experiment within a range of 4 to 6 ms. Then, the latency between Edge Hosts in the same City-level zone is 1 ms, while between Edge Hosts belonging to two different zones at the City-level, it is 5 ms. To access any Regional-level Edge Host from any City-level Edge Host, the additional introduced latency would be 9 or 10 ms. Similarly, at the Regional-level, each host in the same zone is connected via a link characterized by 1 ms latency, while between different zones, it is 5 ms. Finally, to access an International-level Edge Host from any Regional-level Edge Host, the additional introduced latency would be 7 ms. The latency modeling is designed consistently across the entire topology. The main rule that's stands behind our modeling is "the farther the node, the higher latency". We also included some other thoughts related to the signal processing and propagation time.

6.6 Conclusions

The Edge Relocation problem statement considers its modelling, topology design, Edge Host selection constraints, and objective. This modelling allows for design algorithm as a solution for a raised problem. In next chapters we are utilizing the above-described problem modelling to address mentioned constrains and objectives by our proposed algorithms. The proposed original latency modeling for the topology simplifies building reliable simulator for Edge topology to perform evaluation of proposed algorithm under the conditions similar to the real environments. The generic latency model utilize building large-scale topologies, enabling us to validate scallability of proposed algorithms in configurations comparable to those in real environments.

Chapter 7

Edge Application Relocation Heuristic decision algorithm

To address the issue of Edge application relocation as described in the previous chapter, two solutions have been proposed. The first is a heuristic-based algorithm, called `Edge Application Relocation Heurisitc` or in shortcut `EAR-Heuristic` and the second is Rainforcement-Learning (RL) based approach, named `EAR-RL`. This chapter focuses on a heuristic approach, where next is dedicated to Reinforcement Learning solutions.

7.1 `Edge Application Relocation` Heuristic algorithm

This section describes end-to-end operating principle of the proposed heuristic presented in the proposed Algorithm 1. Additionally, subsequent subsections deep dive into more detailed perspective. In order to identify Edge Host to relocate an Edge Application the proposed `EAR-Heuristic` proceeds as follows:

- Firstly, it checks if the current host is still responding to the application requirements (line 3). If so, no Edge Relocation is triggered and the application is maintained in the current Edge Host, and it ends algorithm (lines 4-5).
- Otherwise, `EAR-Heuristic` search for a new Edge Host. To do so, it iteratively divides

Algorithm 1 EAR-Heuristic

```
Input: $n_s$, Topo, currentHost
Output: bestHost
if checkHostSatisfyConstraints(currentHost) then
    bestHost = currentHost
    break
end if
First_Edge = Topo.getEdge_City($n_s$)
Current_Edges[] = Topo.getEdge(First_Edge.Att_city, First_Edge.Att_Reg, First_Edge.Att_Int)
candidateEdgeHosts, toEvalNeighEdges = FindCondidates(Current_Edges)
while candidateEdgeHosts.empty == True do
    checkedEdgeHosts.append(Current_Edges)
    Current_Edges :=[]
    if toEvalNeighEdges.empty == True then
        Return bestHost=null
    end if
    for each edge ∈ toEvalNeighEdges do
        if (!checkedEdgeHosts.contains(edge) &
                !Current_Edges.contains(edge) &
                edge.Att_Int == First_edge.Att_Int) then
            Current_Edges.append(edge)
        end if
    end for
    toEvalNeighEdges := []
    candidateEdgeHosts, toEvalNeighEdges = FindCondidates(Current_Edges)
end while
if (candidateEdgeHosts.count != 0) then
    bestHost = bestEdgeHost(candidateEdgeHosts)
else
    bestHost = null
end if
Return bestHost
```

the coverage zone into several search areas (described at section 7.1.1) in order to greedily explore the Edge Hosts search space.

- Secondly, the previously resulted local search areas are sequentially explored in order to find an appropriate Edge Host. Specifically, our algorithm inspects the local search area to find Edge Hosts that satisfy all application's constraints and therefore consider them as Edge candidates (line 9, or next 24; details in subsection 7.1.2).
- If eligible candidates are found, our algorithm selects the Edge cluster that minimizes the objective function (line 27, described in subsection 7.1.3) expressed in the previous section. Otherwise, it explores the next local search area.
- Finally, if no eligible solution is found in all local search zones, our algorithm will reject the Edge Relocation request and keeps the application in the current Edge Host (lines 13-14, 29).

To summarize, the `EAR-Heuristic` algorithm is presented as Algorithm 1. It iterates over next search area one by one (subsection 7.1.1), while exploring it in the following way: it invokes procedure 2 named "Find Candidates" inspecting given search area to identify Edge Hosts that satisfy all application's constraints and therefore consider them as Edge Candidates (subsection 7.1.2). However, if procedure 2 does not identify any Candidates, algorithm 1 explore next search area in coverage zone, till it identifies any Candidate. Finally, if any of Candidates exist, Algorithm 1 invokes the procedure 3 called "bestEdgeHost" to perform a classification method of choosen Edge Host among Candidates and select the best one according to taken strategy (subsection 7.1.3).

7.1.1 Local search areas construction

As explained previously, `EAR-Heuristic` triggers the search phase only if the current Edge Host no longer respects the application's requirements.
The structure of the local search areas within the defined coverage zone is described in Algorithm 1 and presented more detailed at right side of Figure 7.1. First, the algorithm will build the primary local search zone. To do so, it identifies a reference Edge Host at the City-level. The latter corresponds to any cluster which is directly connected to the network cell where the

end user is located (line 7 of Algorithm 1). Then, all Edge clusters belonging to the same zone will be added to athe local search area. Next, the hierarchical controllers of the reference Edge Host will be selected to be added to the local search area. Note that hierarchical controllers correspond to Edge Hosts belonging to the Regional and International levels and which are controlling the geographic zone to which belongs the reference Edge cluster (line 8). Let's take the Edge-network topology depicted in Figure 7.1 as an example. For the cell number 7 where the UE is located, the reference Edge Host is one of "Zone City 1" Edge Hosts (i.e., 7 to 10). Hence, the first local search zone will be composed of the following Hosts: 7, 8, 9, 10, 2, 3, 4, 1. If no eligible candidate is found, next local search areas are iteratively constructed with respect to the previous local search area. Specifically, a new search area is composed of clusters that are direct neighbors (directly connected) of clusters of the previous local search area (lines 10-25). Consequently, based on the reference topology in Figure 7.1, the second local search zone will be composed of the following Edge clusters: 11, 12, 5 while the third local search space will be composed of clusters: 13, 14, 6, the fourth: 15, 16, and so on. The procedure will be re-executed until the whole coverage zone of "International region 1" is explored or eligible Edge Host candidates will be identified.

7.1.2 Filtering phase and Edge Host selection

As described in Algorithm 2 called "Find Candidates" the filtering phase aims to explore all Edge Hosts in a given search area in order to check whether they are eligible to relocate application or not. The Edge Host can be considered as an eligible candidate only if it satisfies all constraints. Two types of constraints are considered:

- Application-specific (lines 4-8):
  - the targeted latency which is compared with the latency offered by the shortest path between the network cell $n_a \in N_a$ and the current investigated Edge Host (line 4). The shortest path calculation returns also a path how to reach Edge Host that can be used as an instruction for routing protocols if considered host will be selected.
  - the requested virtualized resources which are compared with the available resources of the considered Edge Host (line 7-8)

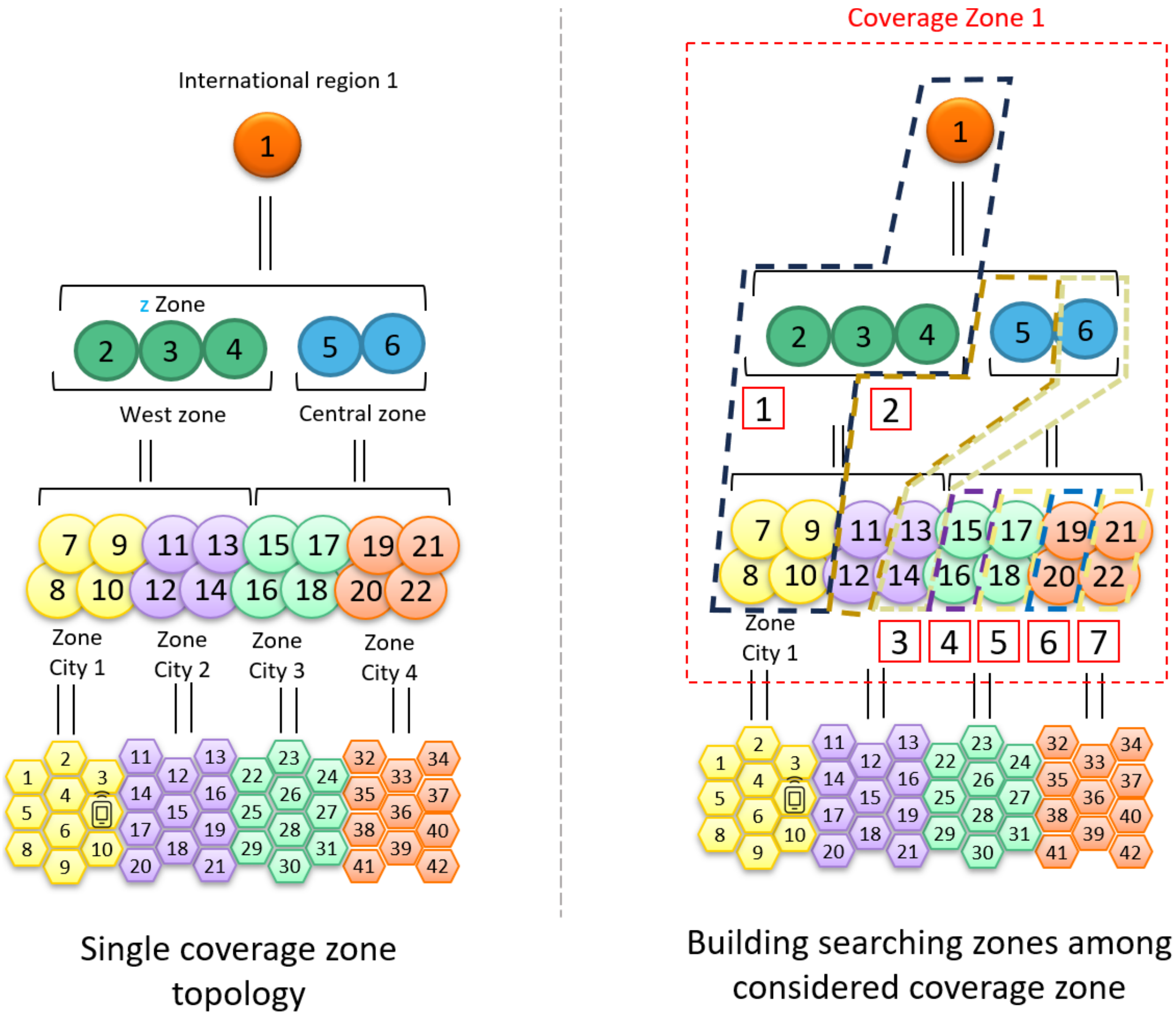


Figure 7.1: Initial placement attempts

- Infrastructure-specific (lines 9-10):
  - the load threshold which is compared with the load of the current Edge Host to which are added the application's requested resources.

7.1.3 Edge Host classification

Finally, among the identified Edge Candidates the Edge Host that minimizing the objective function (described in Section 6.4) will be selected to relocate the application. Among all selected Edge Candidates, the Algorithm 3 is searching for best one according to defined objective function. Best means the Edge Host with the least value of cost calculated according to

Algorithm 2 FindCandidates

```
Inputs: App, $n_s$, Current_Edges
Outputs: candidateEdgeHosts, toEvalNeighEdges
for each edge $\in$ Current_Edges do
    To_EdgeHostLat, path = Shortest_path($n_s$, edge)
    if To_EdgeHostLat $<=$ App.Lat then
        toEvalNeighEdges.append(edge)
        if (edge.CPU $>$ App.CPU &
            edge.MEM $>$ App.MEM &
     (edge.usedCPU $+$ App.CPU) $\leq$t$_i$ &
     (edge.usedMem $+$ App.Mem) $\leq$t$_i$)  then
            candidateEdgeHosts.append(edge, path)
        end if
    end if
end for
Return candidateEdgeHosts, toEvalNeighEdges
```

the defined objective function. It is important to mention here, that the values of latency and resources' utilization are first normalized (to make fair comparison between different physical quantities) before calculating objective function.

7.2 Evaluation

In this section, we assess the performance of proposed `EAR-Heuristic` decision algorithm evaluated in an experimental simulator platform that was described in section 6.5. To gauge the effectiveness of proposed *EAR-Heuristic* algorithm, it has been compared to four related strategies: (i) *O-Latency*, (ii) *O-LoadBalancing*, (iii) *O-Hybrid* and, (iv) *H-Hybrid.* Finally, the obtained results are analyzed and the effectiveness of proposed solution is discussed.

7.2.1 Experiments' assumptions

- **Topology**: Primarily, a single-coverage zone topology is considered, as presented in Figure 7.1 This configuration involves 42 network cells grouped into 4 zones. Additionally,

Algorithm 3 bestEdgeHost

```
Input: $n_s$, candidateEdgeHosts, $\alpha$, $\beta$, $\sigma$
Output: bestHost
Cost = $\infty$
eg = null
for each edgeHost $\in$ candidateEdgeHosts do
    N_Lat, N_CPU , N_Mem = Normalized(Lat($n_s$, edgeHost), Edge.CPU, Edge.Mem)
    Cost_Edge = $\alpha \times N\_Lat + (\beta \times N\_CPU + \sigma \times N\_Mem) \times Static\_cost$
    if Cost_Edge < Cost then
        Cost := Cost_Edge
        bestHost := Edge
    end if
end for
Return bestHost
```

22 Edge Hosts were considered. Among them, (i) 16 belong to the City-level with a defined capacity of 4GB RAM and 4 vCPU, (ii) 5 belong to the Regional-level characterized by 8GB RAM and 8 vCPU, and (iii) 1 belongs to the International-level with 12GB RAM and 12 vCPU.

- **The** τ: has been set to 80% of total capacity of given clusters. Let us recall that τ value specify the maximum tolerated load of Edge Hosts, which in this case is 80% of total capacity (total capacity - t_i).

- **The level cost**: μ is set as follows: 3 for City-level, 2 for Regional-level and 1 for International-level. Let's remind here, that μ value maps the unitary cost of application placement depending on the level. That means it is cheaper to instantiate application in a big international-level centralized data center, rather than in a small Edge Host distributed in a city.

- **Application requirements**: We consider D number of different Edge applications, of which: 33% represents cloud-gaming applications with target latency requirements of 10ms, 33% corresponds to autonomous vehicle autopilot with target latency requirements of 15 ms, and 33% of UAV autopilot applications with requirements of 30 ms. We assume that each Edge application is characterized by a CPU request varying in [0.5–1] vCPU and RAM request varying in [0.5–1] GB RAM. It is worth to remind here, that we are considering dedicated Edge Application, it means that each application is responsible (and closely coupled) for handling single end user.

- **Initial application placement**: A set of D generated Edge Applications with its attributes (type, resources' requirements, initial end user location) is initially deployed randomly across the coverage zone while respecting their requirements. To do so, an initial procedure presented in algorithm 4 is executed. It proceed as follows: for each application, the randomly selected Edge Host is checked. First, checking for enough resources (with a respect to τ value) of the selected Edge Host to host the considered Edge Application is proceed. Additionally, it checks whether the latency (from initial end-user location) to reach the selected Edge Host is lower than the target latency for the application. If either of these conditions is not met, another Edge Host is selected for validation. If none of the Edge hosts can host the Edge Application, the entire procedure is finished with failure. Then, the procedure might be repeated several times, until proper configuration will be identified, since each procedure execution, starts with various order of applications and randomly selected Edge Hosts.

Algorithm 4 Initial placement of Edge Application Algorithm

```
for each app ∈ edge application list do
    while app is not yet allocated do
        Select a random Edge Host
        if Edge Host has enough resources for the app &
           Edge Host latency is lower than app's latency request then
                Allocate the edge app to the Edge Host
        end if
        if all Edge Hosts already explored then
            Return
        end if
    end while
end for
```

- **Number of Applications**: To determine the reference number D of Edge Applications to be deployed in asssumed topology, an prerequisite experiment has been conducted. It aims at finding efficient number of applications. By "efficient" we mean a number that can be deployed relatively easily while still simulate reasonable load on the infrastructure. This allows for efficient evaluation of Heuristic algorithm.

  Figure 7.1 presents the results of initial placement algorithm of Edge Application executed in the assumed topology 100 times. It illustrates how frequently the algorithm can successfully determine initial placement of all Edge Applications, depending on the number of applications. The threshold of 85% marked at chart represents reference efficiency value of initial placement, that has been selected in a heuristic manner.

  To facilitate an efficient performance comparison, the number of applications D reaching at least assumed threshold, has been set to 50. This choice generates a substantial load on the given topology with relatively efficient number of attempts, while providing room for the algorithm to find the initial placement.

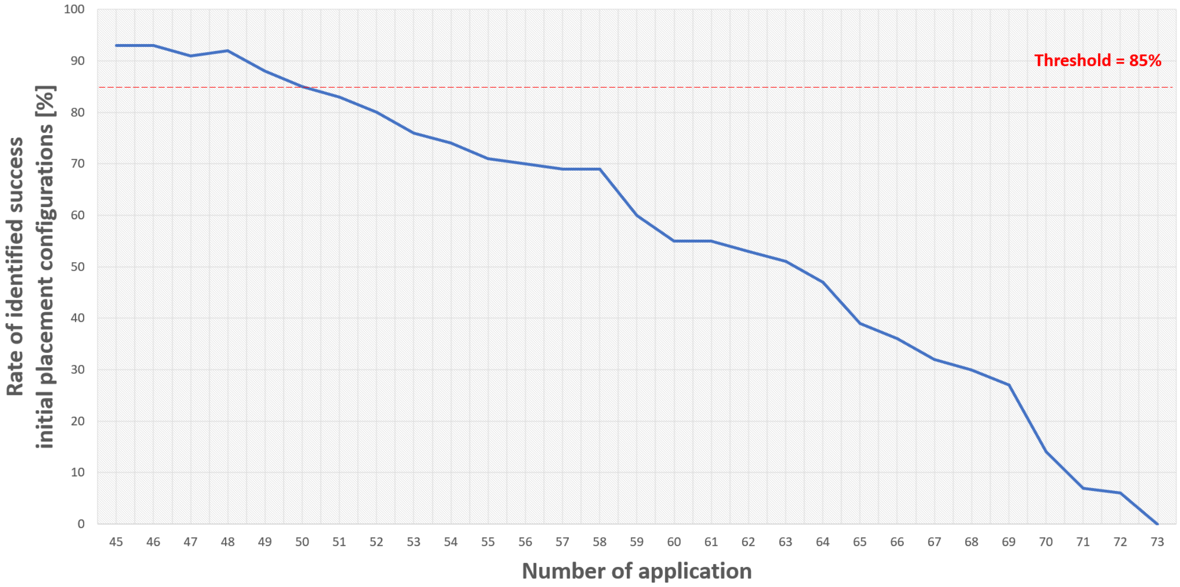


Figure 7.2: Initial placement attempts

The selected number of application needs to be treated as a reference value assumed in a heuristic manner for conducting first performance evaluation and comparison between optimal and heuristic approaches. Subsequently, the experiment described in Section 7.2.6

investigates how the algoirthm's performance varies depending on the number of applications.

- **Relocation requests:** A relocation request is composed of two information, firstly, it specifies the end user to whom this request applies to (user who is associated with dedicated application), and secondly, it indicates the destination cell where the end user of this application is moving to. The first value is generated randomly, while the second relies on a mobility state machine defined for the assumed network cell topology, as shown in Figure 7.2.

  For fair comparison of algorithms, an equal number of relocation requests must be performed in each single experiment. We need specify the number of requests to be analyzed by each of algorithms. For that second prerequisite experiment was conducted. The goal of this experiment was to asses the impact of relocation requests number on stability of obtained performance results of tested algorithm. For this purpose, we deployed 50 Edge Applications in the given topology and executed testing experiment, while validating ratio of relocation rejections of `EAR-Heuristic`. The number of sent relocation requests has been defined between 20 and 400 with a granularity of 20.

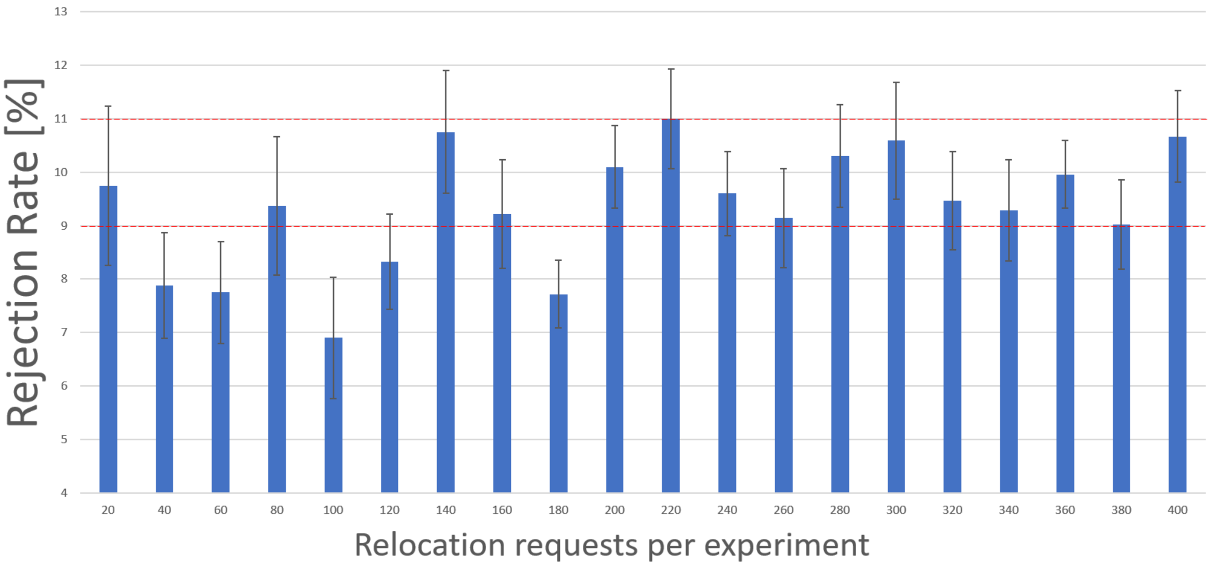


Figure 7.3: Number of iterations tunning

As depicted in Figure 7.3, while assuming more than 200 requests, the obtained results seem to stabilize and achieve similar values, considering the confidence intervals. There is no difference in how many relocation requests are executed for each algorithm after crossing number 200. To conclude this point, a reference number of 250 relocation requests has been selected in a heuristic manner. 250 relocation requests mean, on average, each of the 50 end users (application) will move 5 times.

- **Additional assumptions**

  To make a fair comparison between our proposed algorithm and other strategies, for all tested algorithms we keep the same: set of applications, its initial placement, the same mobility paths (the same relocation requests). Single testing of all algorithms is called a 'single iteration.' In our particular case: single iteration consists of six experiments, each testing a different strategy.

  To ensure the reliability of results, these iterations are repeated 100 times to calculate confidence intervals. Each next iteration covers: different set of Edge application (various application requirements), different initial placement and different set of relocation requests, while keeping the same: topology configuration and number of Edge application D.

7.2.2 Performance metrics

We define the following performance metrics to assess the efficiency of our solution:

- T_r**:** The rate of triggered relocations. It corresponds to the ratio of executed relocations following the end user mobility.

- R_r**:** The rate of relocation rejection. It corresponds to the ratio of rejected relocation requests due to the failure of the algorithm to find an Edge Host to which the application can be migrated.

- CPU_r**:** The rate of average usage of vCPU at Edge Hosts aggregated per level.

- Mem_r**:** The rate of average usage of Memory at Edge Hosts aggregated per level.

7.2.3 Scenario description

We compare our algorithm to the following strategies:

- *O-Latency* strategy selects the optimal cluster that minimizes the latency between the end user and its application ($\alpha = 1$ & $\beta = 0$ & $\sigma = 0$).
- *O-LoadBalancing* strategy selects the optimal cluster that will balance the load of clusters at the considered Edge-network topology ($\alpha = 0$ & $\beta = 0.5$ & $\sigma = 0.5$).
- *O-Hybrid* strategy selects the optimal cluster taking into account a mix of above strategies with a respect to defined weights ($\alpha = 0.5$ & $\beta = 0.25$ & $\sigma = 0.25$).
- *H-Hybrid* is a variant of EAR-Heuristic. This strategy aims to always find better Edge clusters to host Edge application following the mobility of end user. The main difference compared to EAR-Heuristic is *H-Hybrid* is not skipping looking for new Edge Host, while current Edge host is satisfying application requirements ($\alpha = 0.5$ & $\beta = 0.25$ & $\sigma = 0.25$).

Metrics and performance results are calculated, when relevant, with a confidence interval equals to 95% based on 100 repetitions.

7.2.4 EAR-Heuristic parameters weights tuning

In order to identify the most performing variant of `EAR-Heuristic` algorithm, in the first step the parameters of objective function were subjected to evaluation. The tested variants with different objective function weights have been presented in the Table 7.1. The experiments were performed based on assumptions done in previous sections.

Table 7.1: EAR-algorithm objective function weights variants

Parameter	Var I	Var II	Var III	Var IV	Var V
Latency (α)	0.5	0	1	0.7	0.3
CPU (β)	0.25	0.5	0	0.15	0.35
Memory (σ)	0.25	0.5	0	0.15	0.35

According to the obtained results presented in Figure 7.4, Variants II and V achieved the lowest average rejection rate among all the variants. This is because Variant II considers only resource utilization, meaning that minimizing the objective function leads to the selection of the least-loaded Edge Host among currently considered by heuristic a set of Edge Hosts. Similarly

Variant V is mainly focused with finding a load-balancing for Edge topology. This, in turn, increases the capacity of the entire Edge system and results in the fewest rejected relocations. On the other hand, second variant means that objective function is not trying to optimize (minimize) latency, but only to satisfy application requirements. In terms of Triggered rate all variant are performed similarly, so it is hard to specify candidate based on this results. Apart of Variant II, the Variant I has been selected to further analysis, since it is a straight hybrid version, where it should not only balance a load, but as well try to minimize latency. Let's call Variant II as `EAR-LB` (Load-balancing), and the variant I as `EAR-Hybrid`.

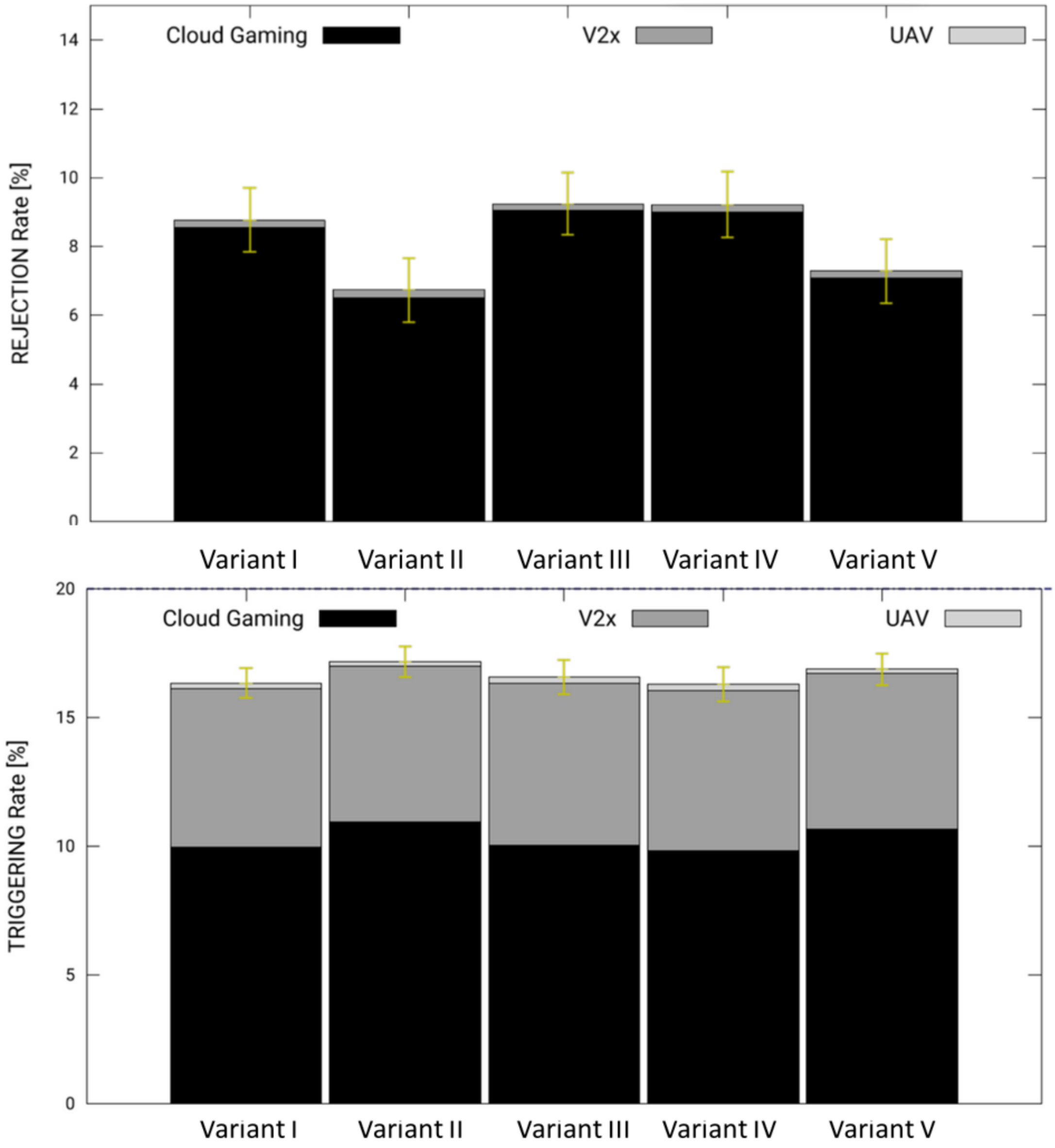


Figure 7.4: Objective function parameters tuning results

7.2.5 Heuristic and Optimal comparison

This subsection presents obtained performance results for all optimal strategies and both variants of `EAR-Heuristic` algorithm. The Figure 7.5 depicts the rate of triggered relocations while Figure 7.6 presents the rate of rejected relocation for each type of Edge Application (i.e, Cloud gaming, V2X and UAV) throughout the experiment. As expected, both variants: `EAR-LB`

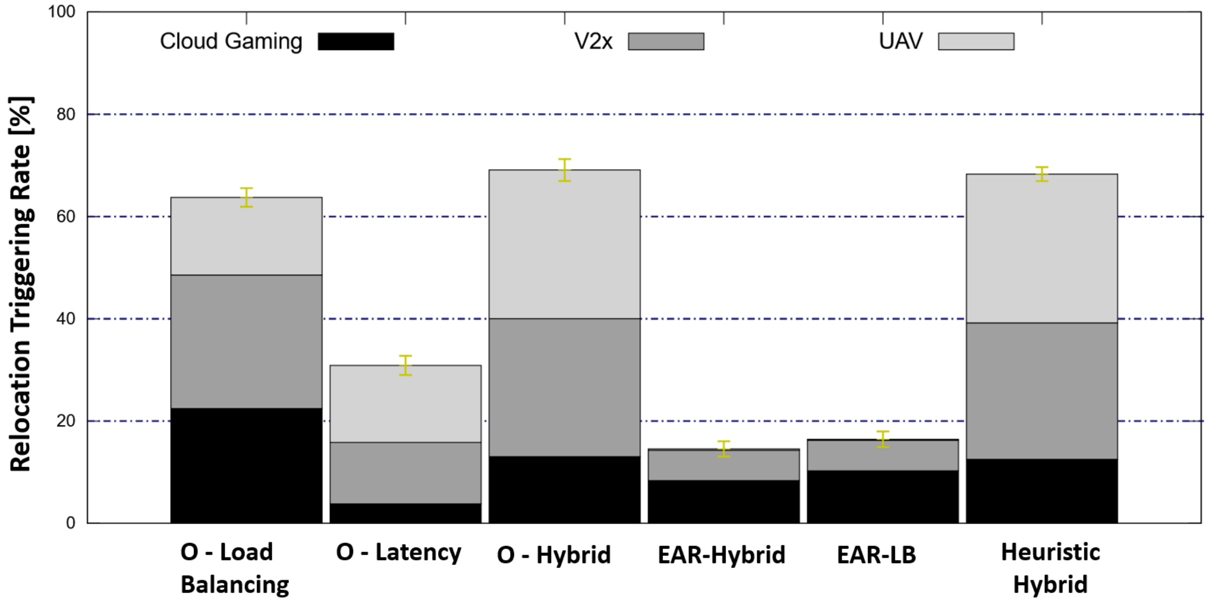


Figure 7.5: Triggering rate for 50 Edge application

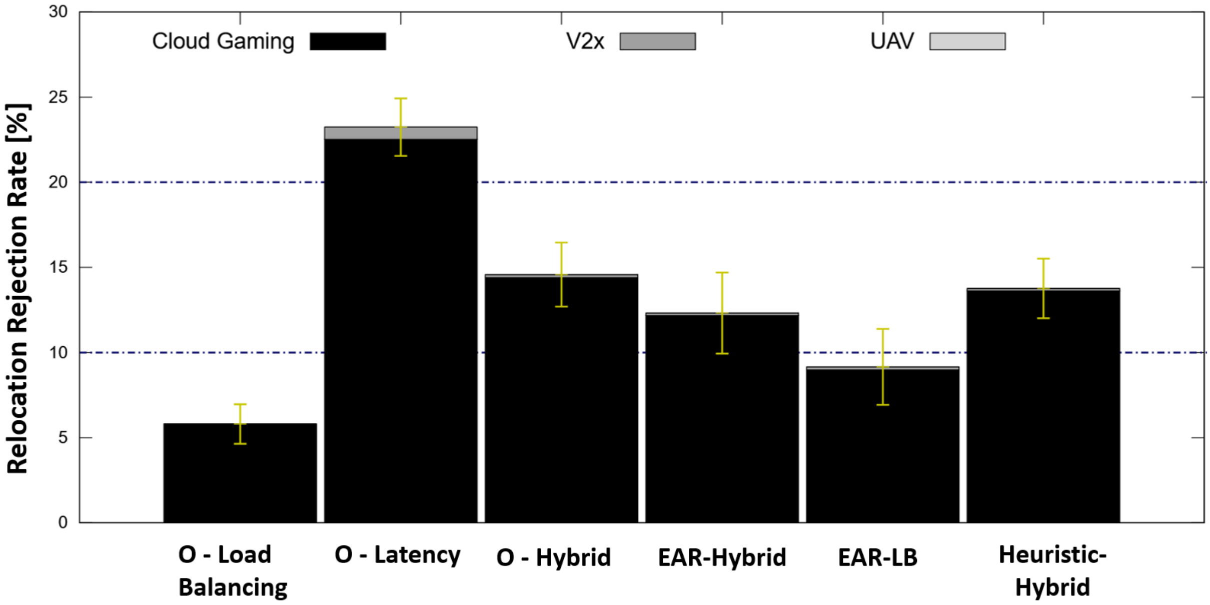


Figure 7.6: Rejection rate for 50 Edge application

and `EAR-Hybrid` of EAR-Heuristic algorithm considerably minimizes relocations rate. Indeed, its reduce the number of triggered relocation by at least 3 times compared to other strategies, while achieving similar rejections rate. In doing so, the applications can be kept in the same Edge cluster if the latter respects its requirements in terms of target latency and resources (i.e., CPU and Memory). Unfortunately, other methods trigger much more Edge Relocations that may inducing hence a service interruption.

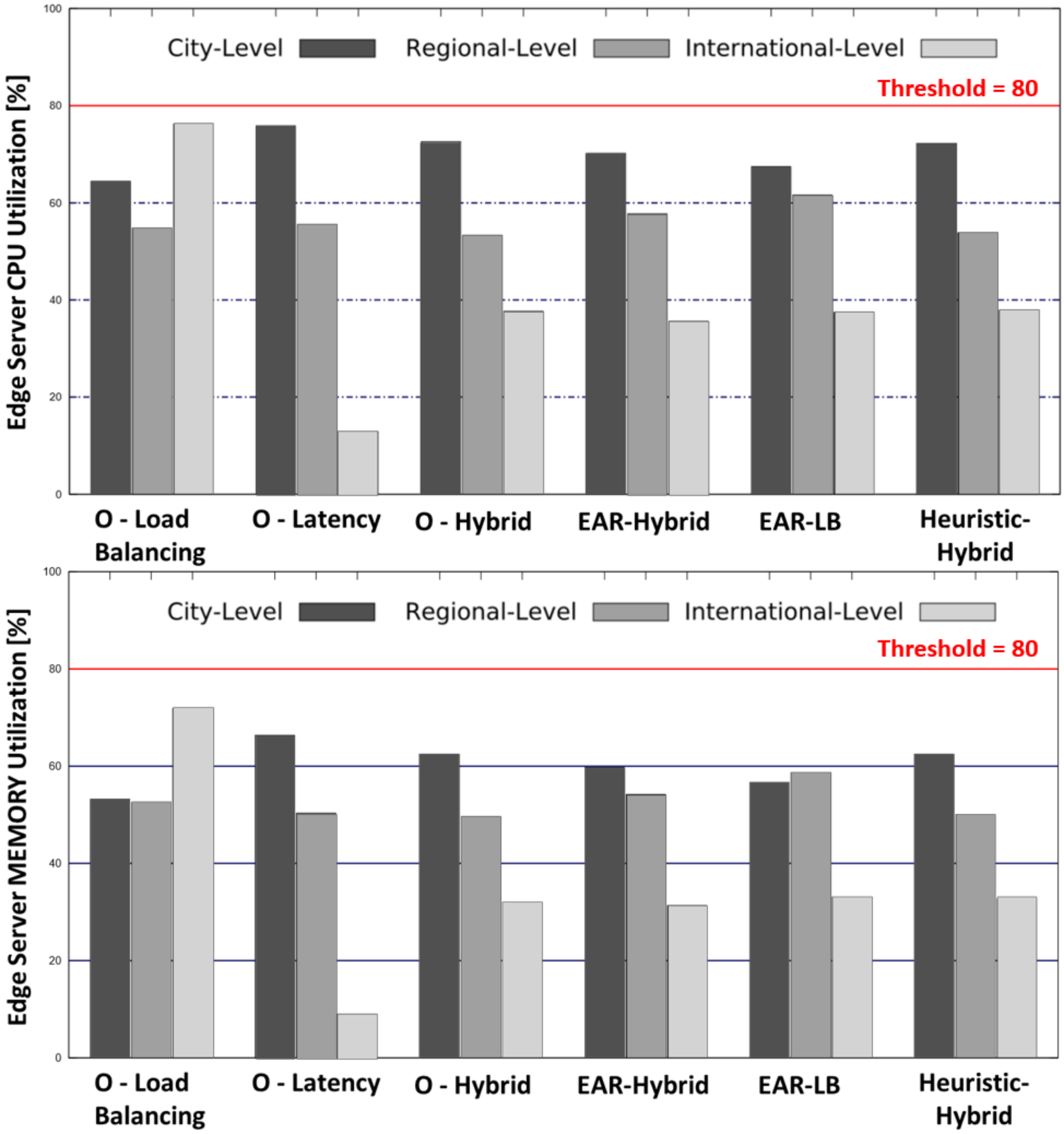


Figure 7.7: Average resources (vCPU, Memory) utilization per level

Next, the Figure 7.7 presents the average aggregated load of vCPU and Memory at the whole Edge topology per level. As expected, the O-Load Balancing strategy evenly distributes the load across levels, while achieving the lowest rejection rate however with a high triggering rate. The O-Latency strategy is trying to first load City-level and next Regional-level clusters in order to offer the lowest latency, but it induces the highest rejection rate due to overloaded clusters. Our EAR-Heuristic algorithm ensures a trade-off between clusters load balancing and latency minimization. Indeed, it maintains a satisfactory clusters load while offering a reasonable latency to end users.

We can observe that O-Latency strategy is trying to place applications starting with the closest levels of Edge topology as depicted in Figure 7.7, however it represents the highest rejection rate among all strategies, since it overload city-level clusters first and than cannot identify more resources for most demanding (in terms of latency) applications.

O-LoadBalancing strategy presents the best results in terms of rejection rate, however it introduces more relocations than our proposed algorithm. Both variants of heuristic algorithm receives low rate of search rejections while maintaining lowest number of Edge Relocations as illustrated at charts 7.5 and 7.6 what is expected value, since first of all our proposed algorithm aims at minimizing the number of Edge Relocation operation, while maintaining low rejection rate.

Table 7.2 depicts the convergence time of the different relocation strategies. The times might be dependent on computational power, however we wanted to illustrate certain trends. It is straightforward to see that `Heuristic-Hybrid` algorithm (special variant of heuristic) receives similar results in terms of rejection and successful relocations rate to the O-Hybrid strategy, while due to our proposed heuristic it minimizes algorithm execution times more than 3 times as presented in Table 7.2, while both variants of `EAR-Heuristic` algorithm minimizes the computation time 6 times in comparison to all optimal searching. The fast and efficient convergence time favor end user QoE by minimizing the service interruption during the relocation procedure.

Table 7.2: Convergence Time

	O-Load Balancing	O-Latency	O-Hybrid	EAR-Heuristic Hybrid	EAR-Heuristic LB	H-Hybrid
Average Execution Time [ms]	77.00	76.64	76.76	14.60	12.97	31.37
Relocation Rejection Rate [%]	5.8 ±1.16	23.24 ± 1.69	14.58 ± 1.88	12.32 ± 2.38	9.16 ±2.33	16.76 ± 1.75
Relocation Triggering Rate [%]	65.56 ± 1.82	32.76 ± 1.88	71.24 ± 2.14	16.05 ± 1.49	17.97 ±1.53	69.67 ± 1.37

7.2.6 End users scaling

In the next experiment, the main objective is to assess the impact of scaling the number of end users on the algorithm's performance. To remind, each end user is linked to a **dedicated** Edge Application, which implies that increasing the number of end users will also increase the number of Edge Applications deployed on Edge infrastructure. All the assumptions for this experiment remain the same with those in the previous section, except for the number of end users (and Edge Applications), which has been adjusted. For comparison purposes, four distinct values of *D* have been chosen to be tested: 30 applications, 40 applications, 50 applications, and 60 applications. Moreover, the same distribution of application types was maintained. Specifically, for each number of applications, $\frac{1}{3}$ were of the Cloud-gaming type, $\frac{1}{3}$ were of the V2X type, and $\frac{1}{3}$ represented UAVs, with precision up to divisibility by 3.

The Figures 7.8a) to 7.8d) presents triggering rates, while charts 7.9a) to 7.9d) depict rejection rates for all tested algorithm variants, corresponding to specific values of "D" representing the number of applications. Presented charts also take into account the different types of applications.

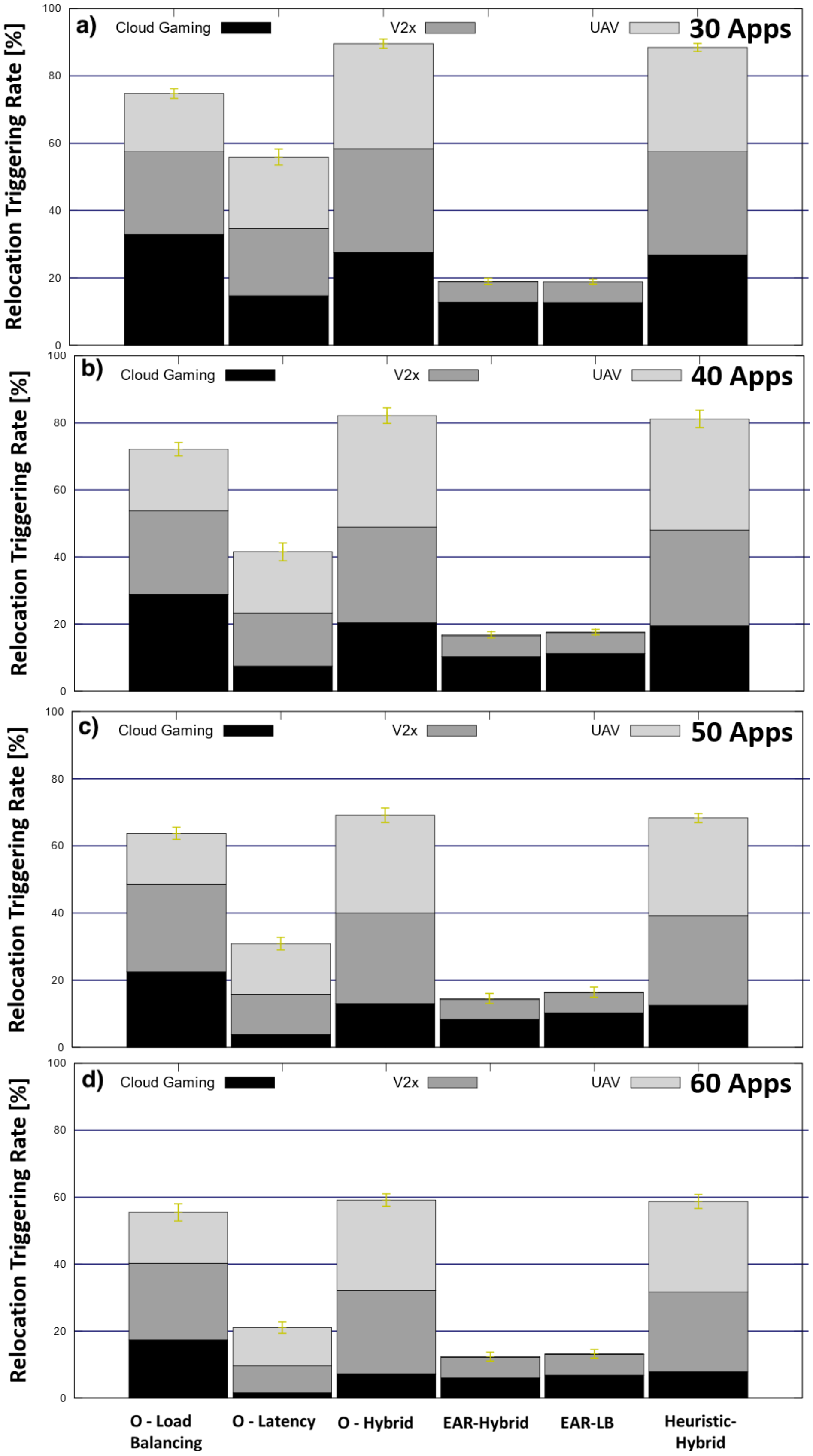


Figure 7.8: Relocation Triggering Rate for different initial number of applications

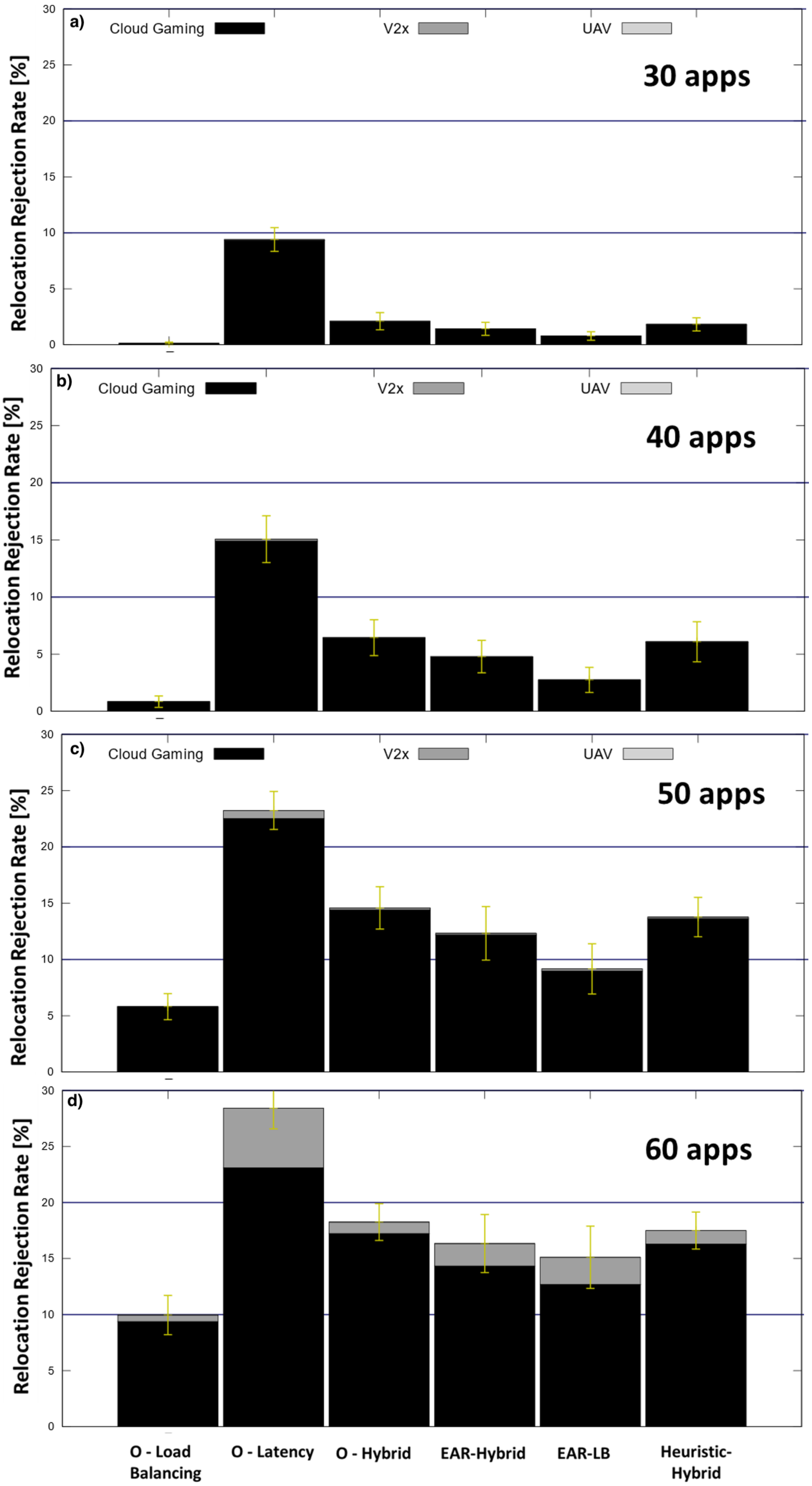


Figure 7.9: Relocation Rejection Rate for different initial number of applications

The initial observation is that the relative ratios of the results obtained for different algorithms remain consistent. The same tendency is observed across all algorithms, regardless of the number of applications. For example, independently on the number of applications, the most effective algorithm in terms of minimizing rejection rate remains `EAR-LB`, while the least effective one is `O-latency`. Furthermore, concerning triggered relocations, `EAR-LB` consistently outperforms the other algorithms by minimizing the number of triggered relocations.

Another noticed trend observed during analysis of rejection rates is that as the number of applications (users) deployed on the infrastructure increases, the rejection rates also rise for all algorithms. This occurs because a higher number of applications generates more load on the infrastructure and reduces the available space on other Edge Hosts for application relocation. Consequently, the algorithms struggle to find optimized Edge Hosts and end up rejecting requests.

Furthermore, in terms of the triggered relocation rate, increasing the number of applications results in a decreasing the relocation triggering ratio. The reason for such a behavior is similar to that for rejection rates. More applications lead to a higher infrastructure load, which, in turn, means less available resources. Consequently, less available Edge Hosts to relocate application directly reduces successful relocations. As mentioned, higher rejection rate minimizes relocation rates, as both KPIs are interconnected.

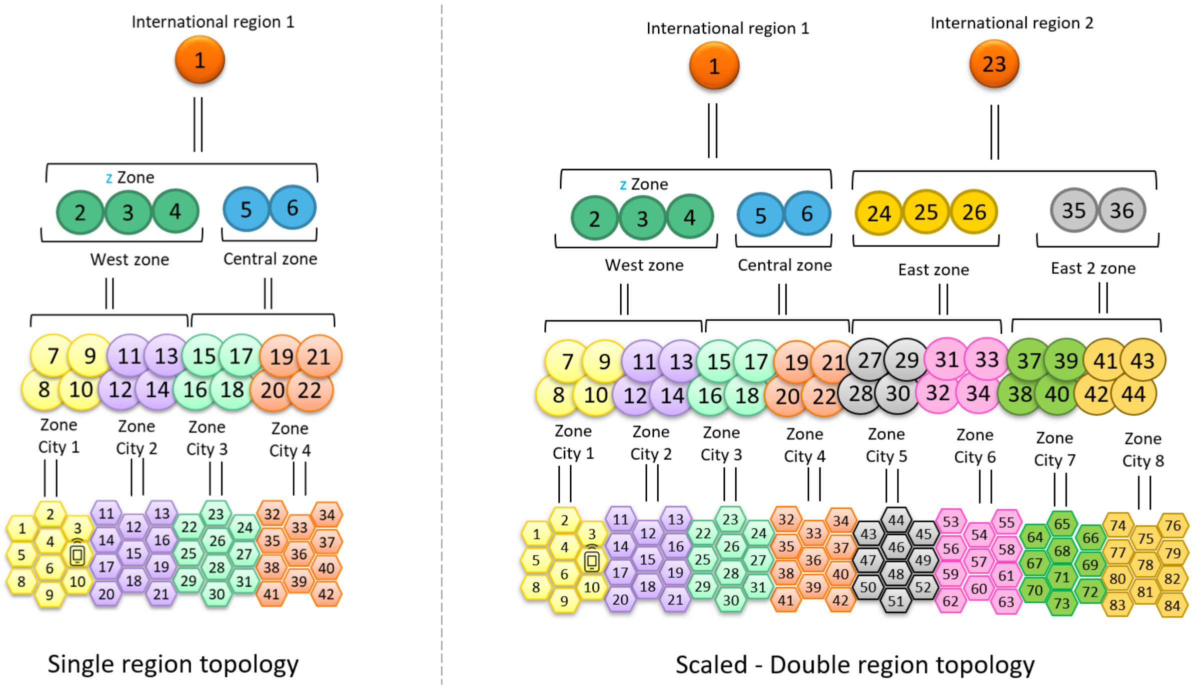
The key outcome of this experiment is `EAR-Heuristic` outperforms other optimal algorithm variants independently of the application number. This allows to scale out algorithm for larger and more 'crowded' infrastructures. Regardless of the application number, the observations and conclusions from the analysis of previous experiment remain valid for all tested algorithms.

7.2.7 Topology scaling

The next experiment aims to validate the impact of scaling the topology on the obtained results. To achieve this, we keep all the assumptions from the original experiments unchanged, except for topology size. The topology has been doubled compared to the primary experiments. Figure 7.10 illustrates both topologies, on the left, there is a regular topology, which was previously explored in last section's experiment. On the right, there is the new topology, which is an expanded coverage zone, mirroring the first coverage zone. This makes the topology twice as large as in the previous experiments, also in the capacity size and composition of Edge Hosts across levels and zones. In addition, the number of network cells were doubled as well. It's important to note that other assumptions, such as the number of users, number of mobility events, and so forth, remained consistent.

In the tested scenario, involving 50 Edge end users, (which correspond to 50 dedicated Edge Applications) were traversing 84 network cells while triggering relocation requests. Table 7.3 presents comparison of convergence times for different algorithm variants for one-coverage (previous section experiment) and two-coverage zones.

Observing the results, it becomes evident that all variants of the Heuristic algorithm consistently achieve comparable processing times across various topology scales. In contrast, all variants of the Optimal algorithms require at least twice the time to converge in doubled scale topology. The reason is the fact that optimal algorithm, no matter on its variant has to consider, check and analyze all Edge Hosts, while heuristic algorithm is focusing solely on a subset of closer located Edge Hosts leading to a consistent convergence time regardless of topology size. In this particular case, doubling the number of Edge Hosts results in a proportional doubling



Single region topology

Scaled - Double region topology

Figure 7.10: Single region topology vs scaled double region topology

Table 7.3: Convergence Time of scalled topology comparison in [ms]

Zones number	O-Load Balancing	O-Latency	O-Hybrid	EAR-Heuristic	EAR-LB	H-Hybrid
Single Zone Average Execution Time [ms]	77.00	76.64	76.76	14.60	12.97	31.37
Two Zones Average Execution Time [ms]	162.68	162.39	160.96	15.37	14.24	34.21

of processing time for the Optimal algorithm. To sum up, heuristic algorithm demonstrates increased effectiveness for larger scale topologies, while maintaining convergence time at similar level. In constrast, the processing time for Optimal algorithm grows linearly in proportion to the Edge Host number. This underscores the ability of scalling the EAR algorithm for extensive and large-scale Edge systems.

7.3 Conclusions

This chapter introduces the `EAR-Heuristic` algorithm and explains its principle of operations. Next, several assumptions of performance evaluation were presented. We have shown tuning of related algorithm parameters. Testing of performance of the proposed solutions was conducted under various conditions, including topology and application scaling, and then compared them with other optimal strategies.

In summary, conducted experiments prove that the `EAR-Heuristic` in its `EAR-LB` variant outperforms other optimal strategies in terms of triggering relocations. This positively impacts both user QoS and QoE, as it results in rare service interruptions caused by Edge Application relocations. `EAR-LB` also performs satisfactorily in terms of minimizing the number of rejected relocations by effectively balancing the load within local topology areas.

Most importantly, all `EAR-Heuristic` solutions remains scallable, considering higher application numbers and larger topology scales while in addition maintaining a short convergence time. This rapid decision-making minimizes service interruption time, enhancing the overall user experience.

Nevertheless, recently, we are observing more and more interest around Machine Learning-based approaches to solve network decision challenges. Especially Reinforcement Learning techniques are gaining a momentum, where the agent is learning by interacting with simulated environment. Thus, we decided to compare the performance of our proposed heuristic solutions with Reinforcement Learning-based models, what is presented in the next chapter.

Chapter 8

Reinforcement Learning decision algorithm

This chapter aims to address the Application Relocation while leveraging the Reinforcement Learning (RL) techniques. It walks us through the entire process of i) modeling Edge-enabled 5G system as an RL Environment, ii) an agent training and iii) a model evaluation. Finally, a comparative study with previous, analytical (heuristic) and optimal solution is presented.

Reinforcement Learning is a type of machine learning where an agent learns to make decisions by interacting with a testing environment in a closed loop. The main concept behind Reinforcement Learning is to enable an agent to learn optimal behaviour (actions) through trials and error, while being guided by environment's feedback system of rewards and penalties [88].

Our proposed Reinforcement Learning based framework, presented in this chapter, was already introduced and evaluated in our research work [73]. Hereafter, we gives insights into description of the environment and conduct various experiments including deep analysis of the obtained results.

8.1 Modeling Reinforcement Learning environment for Edge Relocation

The proposed Reinforcement Learning framework, illustrated in Figure 8.1, empowers users to move across geographies, switch between cells, and trigger relocation requests while ensuring a high-quality user experience. The model makes use of Proximal Policy Optimization (PPO) algorithm to determine optimal Edge Host to relocate Edge Applications while ensuring the desired application-requested latency. The application placement is optimized by maximizing load-balancing at the Edge infrastructure. By ensuring resource utilization and a balanced distribution of workloads, the system's overall performance is enhanced. This sections aims to guide how we model, learn and evaluate Edge Relocation solution using Reinforcement Learning.

In the first phase, let's consider the modeling of Edge Relocation environment. In the context of Reinforcement Learning, an environment aims at describing and converting the real scenario into a testing environment where the agent can play to learn optimal policies. The primary objective of the environment is to map the chosen scenario as precise as possible, to give the agent all necessary information for taking proper actions. The environment is defined by the following assets: State, Action and Reward system.

8.1.1 State Observation Space and Action Space

When we started the modeling of the Edge Relocation RL Environment, we established a set of assumptions, which will be detailed in the following sections. These assumptions primarily aimed to simplify the representation of the real scenario, characterized by high dynamics, where multiple users can freely move simultaneously, changing their positions and triggering applications relocations. Moreover, the simultaneous mobility behaviour of multiple users (and simultaneous relocations) may result in inconsistencies and discontinuities in the environment state representation. To address these challenges, the first simplification involves restricting the environment to handle a single user at a time. In consequence, the observability state have been limited to contain information only about a single user (or application) during a relocation decision. Despite these simplifications, the original problem modeling and objectives remain unchanged, as stated in Section 6.3.

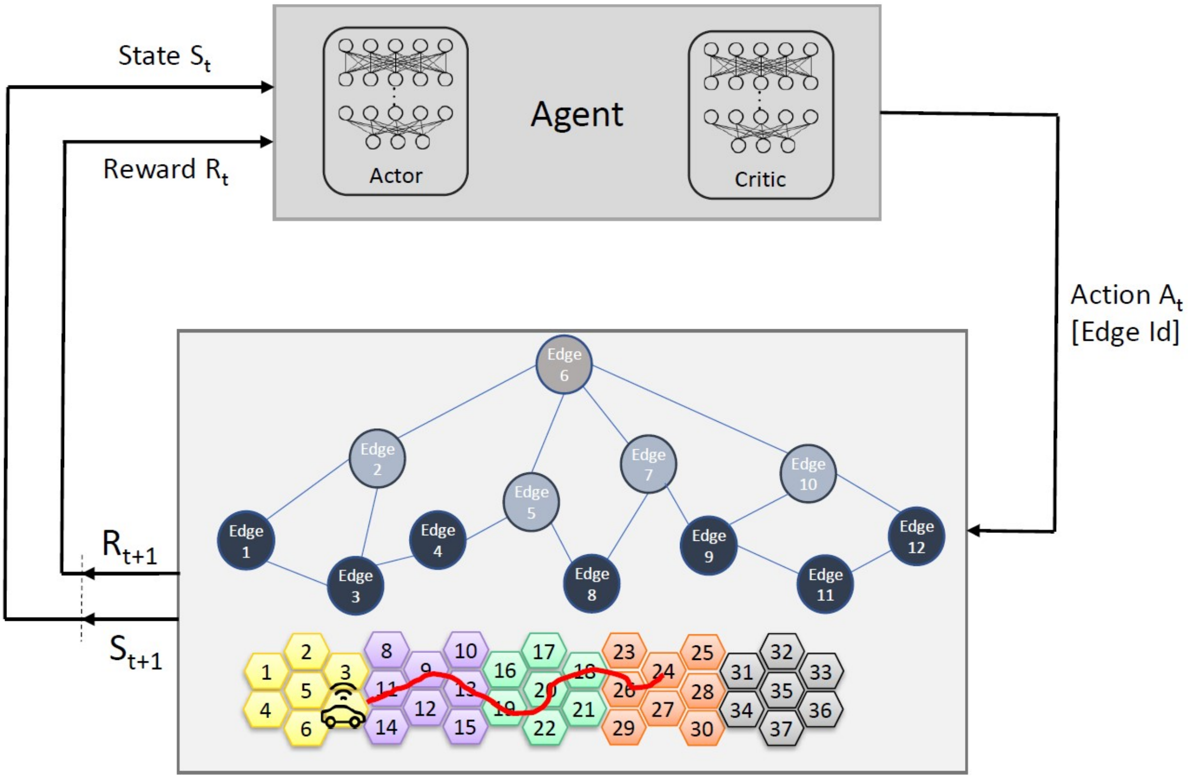


Figure 8.1: Reinforcement Learning framework for Edge Relocation

The state of environment is represented by two vectors:

- The first vector describes the Edge Application $\vec{p}_a$ as shown in Table 8.1. It contains all the crucial information about the currently considered Edge Application, including required CPU, required memory, target latency.

Table 8.1: State representation for Edge Relocation RL environement 1/2

App Attribute	App Required CPU [mvCPU]	App Required MEM [Mb]	App Target Latency [ms]
Value	900	850	10
Range	[500 - 1000]	[500 - 1000]	[10; 15; 30]

- The second vector describes the Edge Hosts, expressed as an array of vectors $(c^r_{h_i})_{r \in R}$ as shown in Table 8.2. Each row represents attributes of a single Edge Server and contains essential information about Edge Host such as: CPU capacity, and available CPU, memory capacity, and available memory, the information whether the currently considered application is deployed at the given Edge Host, and finally, offered latency toward the new network cell of the currently considered end user.

Table 8.2: State representation for Edge Relocation RL environement 2/2

Edge server ID	Available CPU [mvCPU]	CPU Capacity [mvCPU]	Available Memory [Mb]	MEM Capacity [Mb]	Is app Instantiated	Offered latency towards new cell [ms]
Edge 1	9000	12000	8800	12000	False	18
Edge 2	3600	8000	3700	8000	True	14
Edge X	800	4000	900	4000	False	6
Range	[0-12k]	[4k-12k]	[0-12k]	[4k-12k]	[0; 1]	[4-30]

By combining these two vectors, we obtain a comprehensive set of data that can be used to recognize a given state, analyze it and optimize Edge Relocation decision.

Additionally, both Tables: 8.1 and 8.2 present special bottom row called "Range", which is not part of the state representation. Instead, it provides information about the ranges that subsequent values can assume. These ranges for all values make up the Observability State Space. Agent is exploring states (flexible combination of values within their respective ranges) withing State Space, to efficiently recognize a given state and optimize its decision.

Looking ahead, during the evaluation of agent learning, we observed some unexpected behaviour that required a slight modification in the Observability Space. The agent was unable to learn latency constraints due to the fact that the values of latency (both offered and target) were too small compared to other values related to offered and required resources amount. As a solution, we decided to rescale both latency-related values multiplying by *100* to bring them within similar scale compared to other values. This adjustment resulted in a more consistent behaviour.

In addition to the Observation Space, we defined the Action Space, which comprises all Edge Hosts that can be accessed within the considered zone.

8.1.2 Reward Function

The reward function serves as direct feedback from the environment to the agent, assessing the effectiveness of the agent's decisions regarding the aim of maximizing the mean reward over episodes. The modeling of the reward function aligns with our multi-constraint problem modeling. However, based on the results of the heuristic, we adjusted a specific goal for the reward function: optimize load-balancing in order to minimize rejection rate, while respecting target latency requirements.

The reward function is then based on two factors. First, we proposed to modify the objective function of the heuristic solution by not explicitly optimizing the latency. Instead, our focus is on ensuring the required latency constraint. Second, we aim to increase the system capacity through load balancing.

First, to assess the Edge Relocation action in terms of load balancing, we need to find a reliable indicator. We chose to represent it as the delta of the standard deviation of resource utilization, stated as $\Delta\sigma$. This is the difference between the standard deviation before relocation, σ_1, and the standard deviation after relocation, σ_2.

$$\{\Delta\sigma = \sigma_1 - \sigma_2 \tag{8.1}$$

The standard deviation σ of resource utilization is the sum of two standard deviations: one for CPU utilization and another for memory utilization across N Edge Hosts. Here, μ_{CPU} and μ_{MEM} represent the mean CPU and memory utilization, respectively.

$$\begin{cases} \sigma_i = \sqrt{\frac{\sum_{j=1}^{N}(CPU_{util}^{j} - \mu_{CPU})^2}{N}} + \sqrt{\frac{\sum_{j=1}^{N}(MEM_{util}^{j} - \mu_{MEM})^2}{N}} \end{cases} \tag{8.2}$$

In practice, this modeling approach leads the agent to learn directly to select the Edge Hosts with the lowest percentage load. This differs from heuristic solutions, as the focus is not on learning to respect resource constraints while optimizing given objective function. Instead, the objective is to consistently select the best-performing Edge Host each time (Edge Host that is impacting the most on improving load standard deviation). This approach indirectly represents a 'best effort' mode to align with meeting application requirements.

In addition to resources reward, another signal is necessary for our agent to learn ensuring the required latency. To address this, a simple reward system is implemented. A reward r is assigned for selecting Edge Host that satisfies the latency constraint, otherwise a penalty p is imposed. In our case r equals 0, while penalty p is set to -1.

The final step is to determine the appropriate ratio and balance between the weights of both factors in reward function (because load-balancing and latency rewards are summed into final reward) since we want the agent to learn both simultaneously. To achieve this, we have to adjust the scales for $\Delta\sigma$, rescaling to ensure that the results fall within the range $[0,3]$.

8.2 Deep reinforcement learning using PPO

To train our Edge Relocation decision agent that we called `ER-RL` agent, we integrated our custom environment with a Proximal Policy Optimization algorithm [84]. PPO is a deep RL algorithm that iteratively updates a policy to maximize the expected cumulative reward. It achieves this by ensuring conservative policy updates to maintain stability and prevent drastic policy changes. The selection of PPO as the training algorithm for the agent was based on its distinct characteristic. Firstly, it is a model-free algorithm [29], meaning it focuses on learning the pair of State-Action rather than modeling state tansitions in our environment. This is aligned well with the requirements of our problem modeling. Secondly, PPO uses deep neural network, which is an efficient way to express policy. Neural networks are as well a powerful function approximators and it is important to note that the ability to approximate complex and continuous value functions or policies is crucial for RL. The agent is leveraging neural network, so we can specified our learning as "Deep Learning". Last but not least, PPO is known for its stability and robustness, making it a suitable choice for training policies in complex environments such as the Edge Relocation environment. For all these reasons, PPO algorithm has been recognized as a decent solver for a telco-specific issues related to: resource allocation for network slices [54] [31], resource management for 5G network services [72], as well as life-cycle management of 5G Network Functions [70].

8.2.1 Learning hyper-parameters

The learning hyper-parameters in machine learning are external configuration settings that influence learning dynamics during the training [20]. During the training of our agent, we repeatedly monitored the learning parameters and consistently tuned the parameters given below in order to achieve an efficient learning process as suggested in [66]. Finally, we updated several learning hyper-parameter for PPO agent compared to default values [48], especially:

- **Learning Rate** controls the step size during the optimization process. In practice it influences how quickly or slowly a model learns. This parameter has been set to $1e-4$, which is the lowest limit of the learning rate range [48]. Setting it low makes increasing stability of learning and allows to avoid learning sub-optimal policy. We required stability of learning, since we introduced quite high amount of exploration in the learning process. A low learning rate in practice also extends the duration of the learning process.

- **Entropy Coefficient** is a parameter that controls the amount of exploration during the training. During the initial training sessions, we observed the agent has learned a sub-optimal policy, selecting only one or two Edge Hosts. To solve this and push the agent to explore more actions and learn a more optimal policy, we decided to increase the entropy in the training process. This adjustment aims to favor greater exploration, what allows the agent to discover better strategies and improve overall performance. Enthropy has been finally set to 0.9.

- **Number of steps** hyperparameter refers to the number of steps used to collect experiences before performing a policy update. **Number of steps** is set to a value that allows the algorithm to make reasonably policy updates without being too sensitive to short-term fluctuations in the environment. The value was set to 10000 steps per update, which is a quite high value, what means rarer policy updates, enhancing learning stability.

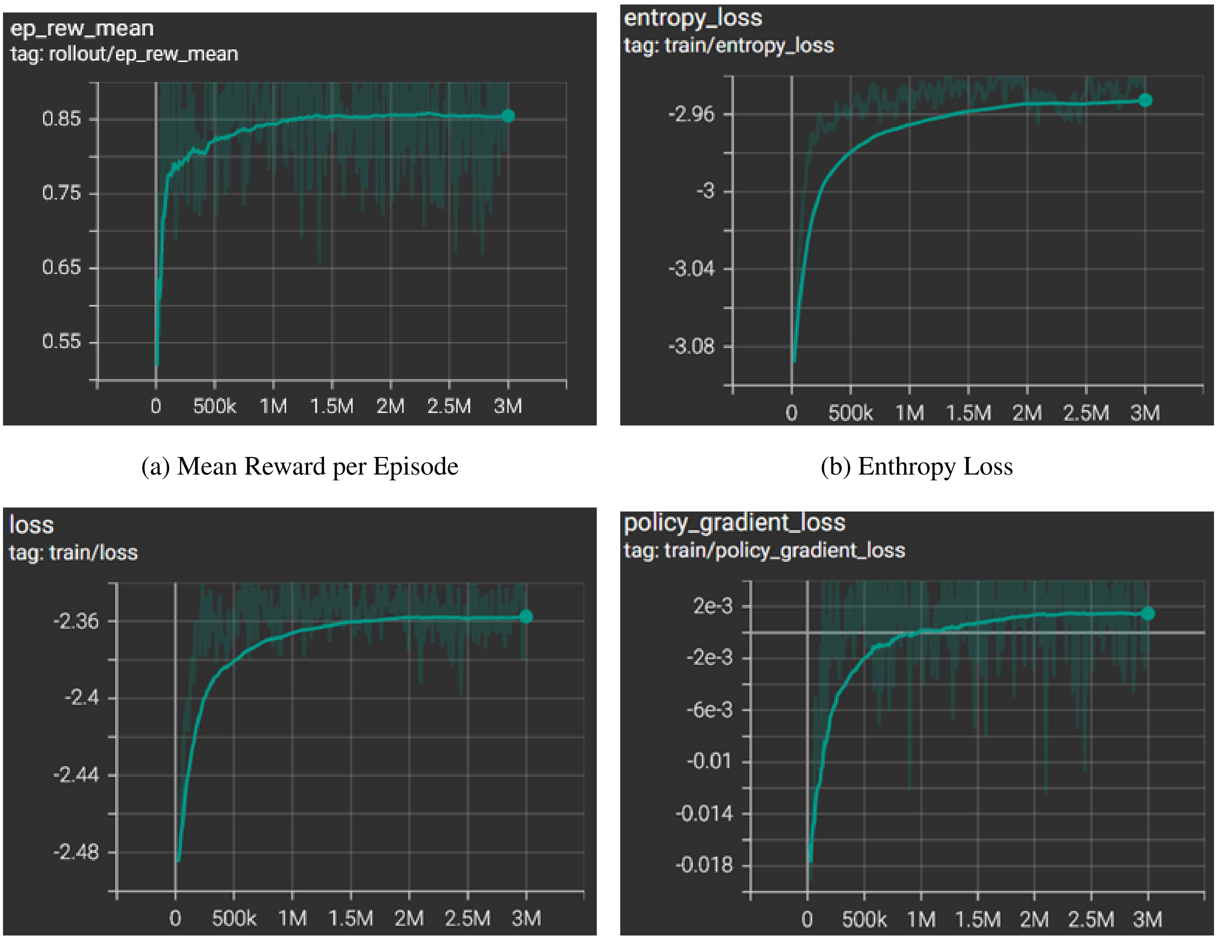
- **Batch size** specifies the number of samples (timesteps) from the collected experiences that are used in each policy update. Batch size is a parameter closely related to previous parameter - Number of steps. The collected experiences are divided into batches, and the policy is updated based on these smaller batches. We set Batch size value to 2000. In practice, after these 10000 timesteps, the agent collects experiences and updates its policy. Since we set batch size to 2000, the collected experiences will be divided into five batches of 2000 samples each, and the policy will be updated five times using these batches. Similarly to number of steps, the bigger batch size makes learning process more stable without being too sensitive.

8.2.2 `ER-RL` training sandbox

To train our `ER-RL` model, we developed a custom RL environment using Python and OpenAI Gym library [22] as described in section 8.1. Each episode in the training process involved generating a new single end-user and application dedicated for this user and than simulating user movement through the cell topology, as shown in Figure 8.1. The end-user is performing a single movement per episode, with the goal as specified in reward function: to optimize resource load-balancing in order to minimize the rejection rate, while aiming to respect target latency requirements. Each episode introduces a new, varied initial load at each Edge Hosts in the topology, simulating the load of other applications.

To achieve this, a unique application is created for each single episode with specific requirements within the range: [0.5-1] CPU, [0.5-1] GB RAM, and latency requirements depending on its type. Specifically, we differentiate three types of applications based on latency requirements: 10 ms representing cloud gaming applications, 15 ms acting autonomous car steering systems, and 30 ms for autonomous drones steering systems. Furthermore, at the beginning of each episode, a UE movement is generated while specifying the first Edge to host the application so that its requirements (i.e. CPU, RAM and latency) are met.

The learning process for the agent is designed as follows: when a user triggers a handover due to its mobility, the agent is activated to determine the optimal Edge Host for placing the Edge Application. Subsequently, the agent takes actions in the environment and receives a reward based on the effectiveness of its decision-making. By leveraging this approach, the system can dynamically adapt to changing user requirements and mobility patterns, resulting in a better overall user experience.



(a) Mean Reward per Episode

(b) Enthrophy Loss

(c) Loss

(d) Policy Loss

Figure 8.2: RL Learning KPIs

8.2.3 Learning evaluation

Reinforcement Learning process is continuously monitored throughout training. Agents learning process is described by the set of KPIs that are used to assess and measure the effectiveness and progress of the learning process. Major learning KPIs have been presented below:

- **Mean reward per episode** is presenting mean reward for agent's decisions per episode. It is is expected to grow over time and finally stabilize at a given high possible value. As presented in Figure 8.6a we can observe high reward increase in the initial phase and than gradually smaller increments until a stable value is obtained.

- **Entropy Loss** is a measure of how much the agent is encouraged to explore and not finish too quickly in the current policy. In Figure 8.2b, during the first phase, we are observing high entropy loss reduction and than gradual stability, as the agent learns and becomes more confident in its policy. Finally, the entropy loss is becoming stabilizing due to finding a balance between exploration and exploitation leading to a consistent level of entropy stabilization.

- **Policy Loss** is a metric that shows how much the current policy of an agent needs to be adjusted during the training process. The desired state is stabilizing or decreasing policy loss over the learning time. As presented in Figure 8.2d, policy learning has stabilized what suggests that the agent is updating its policy in a controlled manner.

- **Loss** presented in Figure 8.6b, indicates how much the agent's predictions differ from the desired outcomes during the training process. The lowest value the better, so once again we can observe high descrease at the beginning and gradually smaller decrease until a stable value is obtained.

All the aforementioned metrics achieved expected "shapes" by growing over the time and learning from history, and than stabilizing at target values. This indicates that agent has efficiently learned some policy. However, this is still not enough to asses accuracy of agent's behaviour. Currently, we cannot be certain whether policy learned by the agent aligns with the desired policy defined by reward function. For assessing whether our environment modeling and reward function have been designed properly, we need to validate the performance of agent's decisions in real-life scenario, as described in the next section.

8.2.4 Masking in Reinforcement Learning

Before comparing our `ER-RL` model to other heuristic and optimal strategies, it is important to compare it to an optimal Reinforcement Learning agent. This comparison will help to asses the efficiency of our agent's policy. Potential reasons for sub-optimal performance may include a) inefficient training (bad hyper-parameters configuration or/and insufficient training) or/and potential mistakes in the environment modeling that may have inaccurately described the Edge Relocation problem as discussed in previous sections. Such a comparison will analyse if the mistaken decisions are a results of sub-optimal policy or simply limited choices in selecting an Edge Host that satisfies constraints.

To conduct this comparison, we implemented Reinforcement Learning with "bad action" masking algorithm, known as Maskable PPO [78][30]. This algorithm masks inefficient actions during the training phase. Before taking an action, the agent checks the environment to identify Edge Hosts that do not meet the constraints (as described in Section 6.3) and selects only those that ensure a successful relocation. This mechanism minimizes the number of unfavorable choices. Nevertheless, if all actions are masked and the application remains at the same cluster while violating the latency constraints, this action will lead to a penalty.

The `ER-RL-masked` is a hybrid solution between analytical method (as agent is checks and masks inappropriate actions in advance) and Reinforcement Learning (selecting action that maximize reward function among the appropriate ones). We have adjusted reward function to masked actions only. In the new reward function, the agent is assessed only for Load-Balancing, with latency considerations excluded since latency is always guaranteed due to the masking.

The `ER-RL-masked` agent represents an optimal version of the Reinforcement Learning agent and allows us to verify efficiency of our previously defined `ER-RL` agent.

8.3 Performance Evaluation

In this section, we assess the performance of our trained `ER-RL` model and compare it with: `ER-RL-Masked`, `Optimal` and `EAR-heuristic` algorithms proposed previously in section 7. To conduct the performance evaluation, we integrated our proposed `ER-RL` and `ER-RL-masked` agents into the existing Edge Relocation simulator introduced in section 6.5. Placement Controller has been enriched by new sub-component called: `RL-Edge Relocator` that is responsible for RL-based decision for Edge Applications relocation. This integration allowed us to validate performance of all proposed algorithms using simulator under consistient conditions.

8.3.1 Edge Relocation evaluation environment

To evaluate `ER-RL` trained model, we leveraged simulator under conditions similiar to those considered in evaluation of heuristic algorithm in Section 7.2. Let's remind the key assumptions of experiments.

First, we consider single zone again, the one on the left side as illustrated in Figure 8.3. The simulator relies on an Edge infrastructure consisting of 22 nodes divided into three levels: one International Edge Host with 12 vCPU and 12 GB RAM, 5 Regional Hosts with 8 vCPU and 8 GB RAM each, and 16 City-level Edge Hosts with 4 vCPU and 4 GB RAM each.

Additionally, the simulation environment includes as well 42 network cells that enable the attachment of end-users to the Edge-enabled 5G system. In doing so, the cells allow users to move within the simulated network. These network cells and Edge Hosts are grouped into zones (highlighted with a common color in Figure 7.1) based on the distance and location. The distance allows to generate latency between network cells and Edge Hosts as already introduced in Figure 6.4.

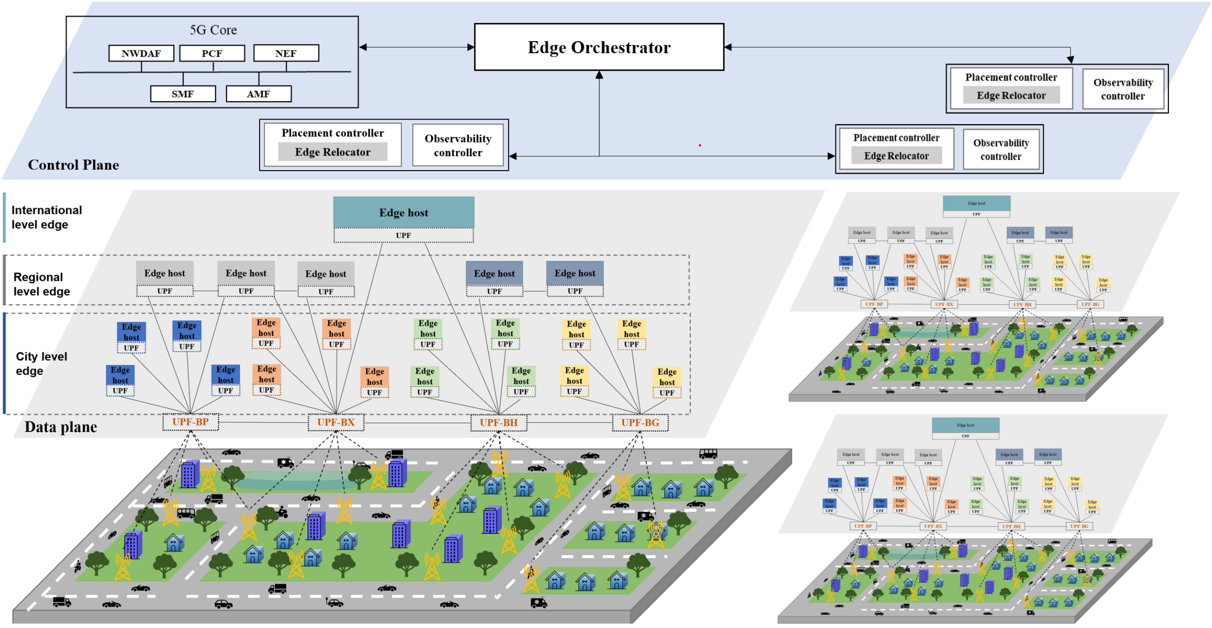


Figure 8.3: Edge Topology

8.3.2 Scenarios Description

Similarly, we consider 50 Edge Applications equitably shared between cloud-gaming, autonomous vehicle and UAV autopilot applications (i.e. 33% of each type). Additionally, we set a capacity threshold X to 80% of the total available resources of the Edge Host. This threshold ensures that a reserve of resources is available for emergency services in case of unexpected traffic spikes or failures as stated in Section 6.3. Not satisfying this constraint will classify such a decision as a failure one for any type of used algorithm: RL-based, heuristic or optimal.

To ensure realistic user movement patterns, we generated the same 250 movement events for each experiment type using a random selection of UEs. The experiment was designed to select the first moving UE randomly, and then generate the subsequent cell destinations based on their current location. This approach allows the simulation of user mobility patterns that reflect real-world scenarios.

We recall that `EAR-Heuristic` is a heuristic-based algorithm capable of optimizing the selection of the destination host while jointly responding to the application requirements and balance the resource consumption of the Edge infrastructure. `Optimal algorithm` selects the Edge Host that optimizes selection of lower latency Host while trying to load-balance resources. For each experiment, we used the same trajectory for the UE and the same initial placement for a set of applications. This allowed to evaluate algorithms on a consistent basis and draw meaningful comparisons between them.

8.3.3 Performance KPIs

We have established the following metrics for evaluating the effectiveness of our solution's:

- T_r**:** the rate of triggered relocations. It corresponds to the ratio of executed relocations following the end user mobility.

- R_r**:** the rate of relocation rejection. It is the ratio of rejected relocation requests due to the failure of the algorithm to find an Edge Host to which the application can be migrated.

- CPU_r**:** the rate of average usage of vCPU at Edge Hosts aggregated per level.

- Mem_r**:** the rate of average usage of Memory at Edge Hosts aggregated per level.

Metrics and performance results were calculated with a confidence interval of 95%, wherever applicable. To ensure robustness and accuracy of the results, we repeated the entire experimental procedure 20 times, allowing us to achieve statistically significant confidence intervals with the t-Student distribution.

8.3.4 Experimental Results

Firstly, it is crucial to compare and explain the performance of both version of RL-based algorithms. As anticipated, the masked variant demonstrate better results than normal variant in terms of minimizing rejections and triggering decisions. The masked-agent covers all non-feasible actions through constraints verification, and ensure the selection only available Edge Hosts, effectively minimizing relocations. This makes it a kind of optimally trained model for RL-based algorithm.

The non-masked variant has the potential to achieve similar performance, however, this was achieved only partially in our training. As previously explained, the training environment we established relies on simplified assumptions, impacting the final performance. Additionally, tuning the learning hyper-parameters turned out to be significant challenge, leaving opportunity for further improvements. Nonetheless, the non-masked variant still operates efficiently compared with other strategies. Consequently, we can confirm that our proposed RL-based solution successfully addresses Edge Relocation problem. Let's take a deeper look at comparing RL-based algorithms with the previously introduced optimization algorithm.

In our comparative study, we focus on algorithms that shares design approach and can be fairly compared. Both ER-RL variants, `EAR-LB` and `Optimal-LB` are specifically designed in the same manner to ensure low latency and load balancing, and we are focusing on comparing them. The remaining variants including i) `Optimal:Latency`, ii) `Optimal:Hybrid`, iii) `EAR-Hybrid`) involve a slightly different design approach, considering the optimization (minimization) of latency as well. We decided to keep all the variant to maintain all available options.

Figures 8.4 and 8.5 illustrate the summary of rejection and relocation rates for heuristic, optimal and RL-based algorithms. Firstly, the rejection rate evaluates how often the algorithm fails to find an Edge Host that meet the requirements of the application. It is straightforward to see that `ER-RL Masked` outperformed all other algorithms, especially our previously proposed `EAR-LB` (2.47 times lower rejection rate than `EAR-LB`). `ER-RL-Masked` algorithm achieves the lowest rejection rate of all solutions resulting in rate of approximately $3.16\% \pm 0.87\%$ for cloud-gaming, while not rejecting neither V2x nor UAV applications. At the first sight it might seems also bit non-intuitive, that `EAR-RL-Masked` achieved lower rejection rate than optimal approach, however, still it is a matter of slight design difference, Optimal approach in objective function takes into account load-balancing among different levels of Edge Hosts topology, while `ER-RL-Masked` makes load-balancing without distinguishing servers into different levels, what makes `ER-RL-Masked` a bit better in terms of rejection rate than `optimal-LB`.

Secondly, the triggering rate evaluates how often the algorithm triggers Edge Relocation operation to maintain the QoS. It is clear that the `EAR-Hybrid` remains unrivaled among all algorithms while the `ER-RL` algorithm outperforms the `Optimal:Load Balancing` algorithm. The contrasting behaviour between the `EAR-LB` and `ER-RL-Masked` algorithms can be attributed to their respective decision-making criteria. `EAR-LB` algorithm uses a simple approach that avoids relocation if the current Edge Host meets the application's requirements, while the

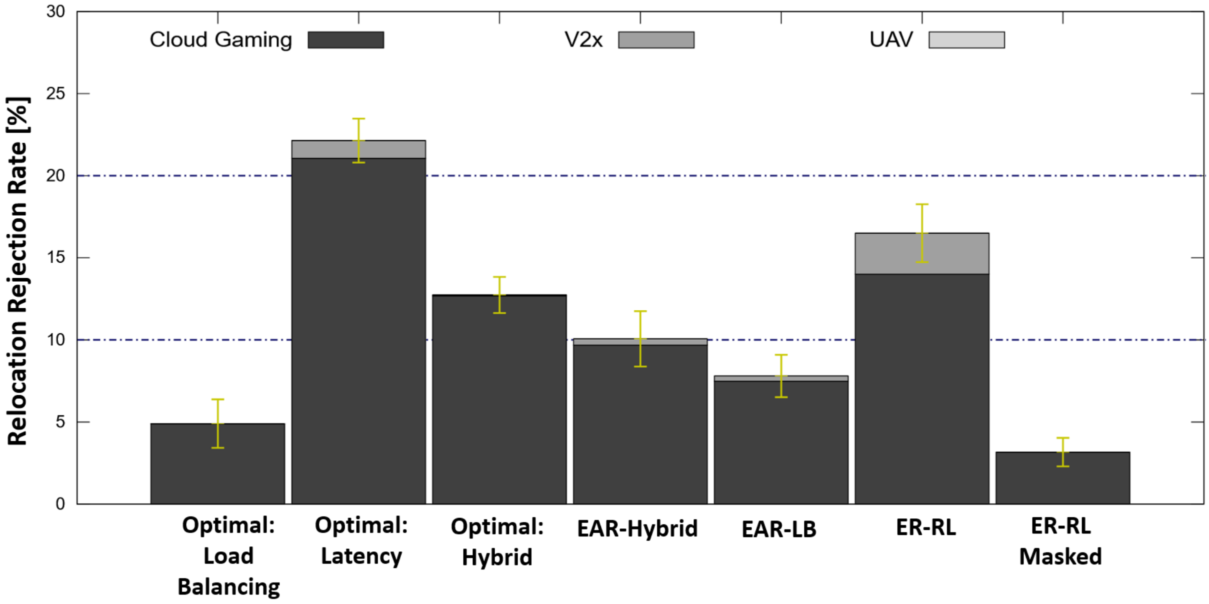


Figure 8.4: Rejection rate summary including RL-based algorithms

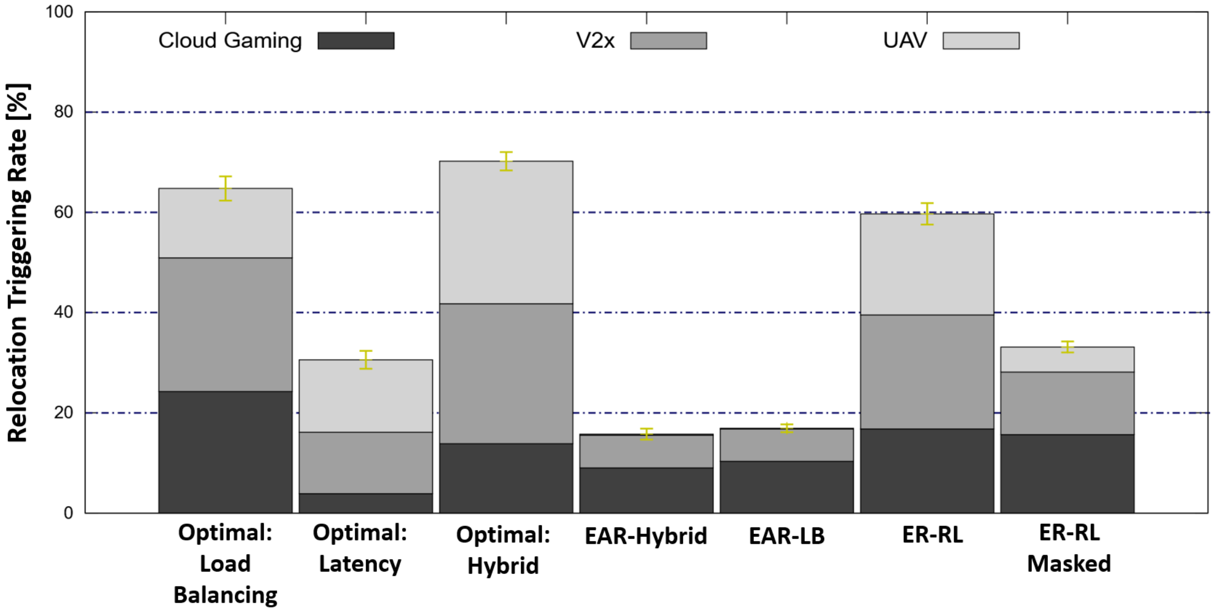
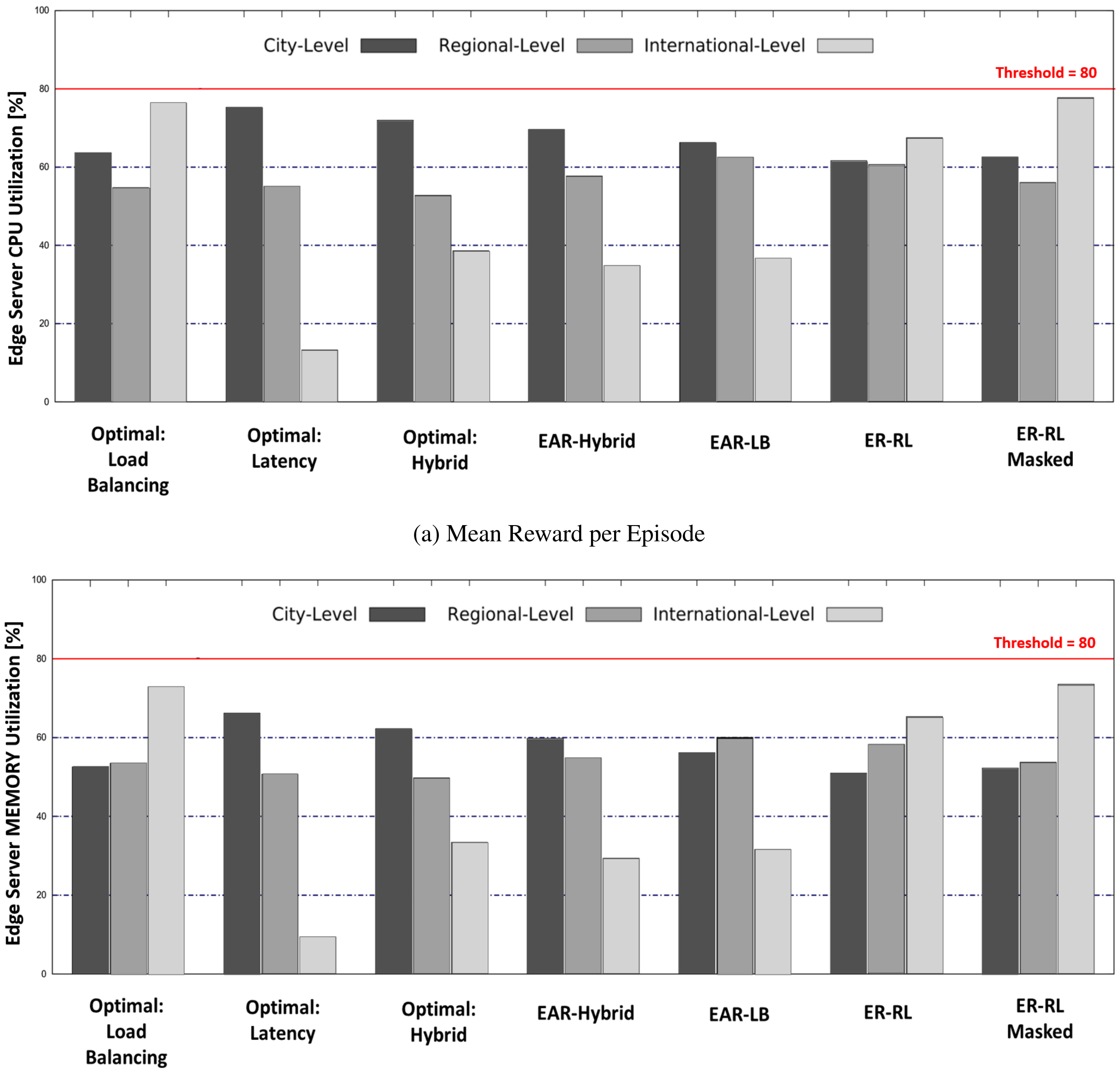


Figure 8.5: Triggering rate summary including RL-based algorithms

ER-RL-Masked algorithm utilizes a more advanced reward function that prioritizes load balancing, even if the highest reward can be obtained by staying with the current Edge Host. ER-RL-Masked algorithm aims to achieve optimal performance in dynamic environments, where resource availability and utilization may vary over time. Overall, ER-RL-Masked takes a more global perspective by considering the future capacity of the system, while EAR-heuristic optimizes only for the current situation.

Additionally, Figure 8.6 shows that both variants of ER-RL algorithm demonstrates de-



(a) Mean Reward per Episode

(b) Loss

Figure 8.6: Load distribution for Edge Relocation RL-methods

cent load-balancing capabilities across different levels of Edge infrastructure compared to the EAR-heuristic algorithm. This efficient load-balancing is achieved due to the ER-RL-Masked algorithm's ability to allocate resources based not only on the current situation but also in the context of previous and future decisions. Well load-balancing enables ER-RL-Masked to minimize the rejection rate.

Next, we observed that both Optimal-LB and ER-RL-Masked approaches are resulting in nearly identical load distributions (CPU and Memory) across levels, as depicted in Figure 8.6. It is worth to recall that the optimal approach relies on the calculation of objective function, selecting the least loaded Edge Host considering levels, while for the RL solution we trained a model to select the one that minimizes the most standard deviation of CPU and Memory utilization across the topology. Consistent results in both cases suggests that delta of standard deviation is a solid KPI to asses load-balancing, which allowed us to train well-performing model in terms of load-balancing. To continue this observation, even though both solutions are well balancing a load, the ER-RL-Masked approach has achieved lower convergence time than Optimal:LB as shown in Table 8.3.

Table 8.3: Convergence Time for RL-based algorithms

	O-Load Balancing	EAR-Heuristic LB	ER-RL	ER-RL-Masked
Average Execution Time [ms]	88.24	15.49	70.69	72.95
Relocation Rejection Rate [%]	4.9 ±1.48	7.8 ±1.29	16.5 ±1.76	3.16 ±0.87
Relocation Triggering Rate [%]	64.76 ± 1.82	16.92 ± 0.78	59.7 ±2.14	33.14 ± 1.09

Table 8.3 provides a summary of the convergence times for all tested algorithms. Both RL variants have achieved better convergence time than Optimal approach however the `EAR-LB` still stands out as the fastest solution. The difference of execution times for ER-RL and ER-RL-Masked is due to the additional time required for mask computation. In general, the execution time for RL models is not significantly influenced by the topology size (except the constant time for mask calculation or gathering the current environmental state). This suggests that RL-models can be efficiently applied to large topologies, as a performing and quick solution.

8.3.5 Conclusions

In this chapter, we introduced a novel application relocation method based on reinforcement learning techniques. We implemented a training environment based on OpenAI's Gym library [92], and trained RL agents using PPO and MaskablePPO algorithms. In conclusion, the comparison between `EAR-heuristic` and `ER-RL-Masked` algorithms performances show that both perform better than the `Optimal` algorithm. The selection of which algorithm to use depends on the specific requirements of a telco operator. If operator's goal would be to minimize the number of relocations to ensure uninterrupted communications, `EAR-heuristic` or `EAR-LB` is recommended. On the other hand, if the goal is to achieve the lowest rejection rate, to load-balance resources and always provide an Edge Host for the application, resulting in higher system capacity for a larger number of end-users, `ER-RL-Masked` is recommended. This trade-off is discussed in the 6G NGMT white paper [107], which considers the design of future networks. Choosing the appropriate relocation algorithm is crucial for creating efficient networks for 6G and beyond.

In terms of practical insights coming from our research, our hands-on experience with Reinforcement Learning has shown that efficient training of RL model including all prerequesties, such as modeling of training environment is really demanding task. Especially, tasks like hyperparameters tunning and adjusting state/modeling can be time-consuming, which may be considered as a drawback compared to analytical solutions. On the other hand, we found RL techniques to be highly practical. Once a model is trained, there is no need for runtime analysis when a relocation request occurs. The RL model simply evaluates the current state and matches it with a predefined decision, providing the appropriate Edge Host.

Chapter 9

Summary

The convergence of 5G network and Edge Computing is fostering the development of innovative use cases making the dream of a fully connected, intelligent digital world almost true. However, the stringent requirements coupled to the high dynamicity of these new applications make their orchestration extremely challenging. Specifically, the mobility of end users will undeniably impact Edge operations. Indeed, the Edge-enabled 5G systems need to provide the capability to follow moving users while respecting their latency requirements.

In this dissertation, we comprehensively address the support of session continuity in Edge-enabled 5G system. Relocation between Edge Hosts is required during the mobility of end user in 5G network, leading to degradation of QoS. To tackle this challenge, we designed, implemented and evaluated an Edge Application relocation procedure by leveraging and expanding pre-commercial systems of Orange, including: Kubernetes, the de-facto standard for cloud-native application orchestration and Edge Multi-Cluster Orchestrator (EMCO) solution which provides the capability of orchestrating Edge applications in a multi-cluster environment. The implementation of Edge Relocation procedures enables Orange to deploy these tools in Edge Computing production environment. In a research context, we addressed smart relocation decision challenge by introducing two novel multi-criteria algorithms named: a) heuristic, and b) Reinforcement Learning. Both aims at selecting a new Edge server to relocate Edge application, which state unique contribution to the research area of Edge services continuity in 5G system.

9.1 Dissertation contributions

In chapter 2, we introduce the background information of 5G network connectivity with a focus on ultra reliable low-latency communication. We introduced ETSI-based Multi-Access Edge Computing architecture, a crucial enabler for 5G system in achieving promised low-latency communication. Then, we highlighted the main 5G network components that interact with Multi-Access Edge Computing and presented the possible manner of both systems integration. Moreover, we identified a functional gap for the support of service continuity withing integrated 5G and MEC system. Both business and research motivation to address this issue were provided. Finally, we presented a set of use-cases that are awaiting for proposed mechanisms for industrialization, including autonomous vehicles steering systems, UAV steering systems, cloud-gaming or extender reality.

In chapter 3, we pointed out several open research challenges and technological gaps of Edge Computing with a particular focus on advanced services management mechanisms. We identified gaps related to the design and deployment of Edge systems for telco operators. These include issues such as the granularity of Edge Hosts; an efficient observability of both infrastructure and application level for triggering smart life-cycle management operations. Moreover, we highlighted implementation gaps, such as multi-cluster connectivity in a multi-cloud environment and application state synchronization. Finally, we positioned contribution of this thesis as point bridging distributed systems, Edge-enabled 5G systems, Management and Orchestration, and Application Relocation through heuristic and Machine Learning techniques.

The related work presented in chapter 4 provides insights into various research and implementation perspectives on supporting service continuity in Edge Computing. We conducted a comparative analysis of existing solutions, and pointed out how the related work tries to solve relocation decision problem in incomplete manner, often considering dummy metrics, or neglecting the influence of 5G access network. Additionally, we highlighted the originality of our approach by introducing decision making algorithms that relies on metrics coming from both Edge infrastructure and 5G core network. Finally, we identified set of 3GPP core network procedures from releases 16 and 17 that we found useful for the proposed end-to-end relocation workflow.

In chapter 5, we provided the architecture of Edge-enabled 5G system and the `5G-Edge`

`Relocator` framework, which was implemented as PoC, taking into consideration cloud-native principles and a microservices based 5G network architecture. We presented interfaces for interconnecting both the control plane and data plane of Edge Computing and 5G system. Then, we implemented the proposed system, while leveraging a set of open-source projects such as free5GC, UERANSIM for 5G network and Kubernetes, and EMCO for Edge infrastructure implementation. Additionally, we implemented our own code extensions to the Edge platform by implementing missing components, like Edge Topology, Placement Controller or Observability Controller. Next, at the top of our system Edge-enabled 5G system, we designed an end-to-end Edge Relocation workflow, encompassing: a) observability of end-user mobility events, b) triggering the relocation decision procedure, relying on real-time infrastructure measurements and 5G network control messages, and finally c) the execution of a zero-downtime containerized Edge Application relocation across Kubernetes clusters. The implementation outlined in this chapter stands as an industrial contribution of this thesis, forming part of the evaluation and industrialization project for Orange, a telecommunications operator. The aim of this project is to enhance pre-commercial solutions for Edge Computing management and orchestration, specifically focusing on Kubernetes and EMCO (Edge Multi-Cluster Orchestrator) [4].

Next, in chapter 6, we introduced the Edge Relocation problem statement, providing a clear overview of problem modeling, the problem statement, and the simulation model. The key components include: i) Edge and network topology modeled as a graph, considering different classifications such as levels, zones, and coverage zones; ii) representation of Edge Application; iii) problem statement where the objective is to find an appropriate Edge Host to relocate a given application while respecting its requirements and considering constraints such as resources utilization and latency. iv) definition of objective function; v) simulation model based on demonstrator presented in previous chapter; and vi) latency modeling in the environment. The formalized problem modeling is a prerequisite for solution presented in following chapters.

In chapter 7, we proposed a heuristic algorithm called EAR-Heuristic that aims at selecting a new Edge Host for Edge Application relocation. The algorithm divides the Edge Hosts topology into sub-topologies and analyze, subset of Edge Host that satisfies defined constraints such as latency or resource utilization. Among the servers that meet constraints, algorithm is calculating proposed objective-function, and the highest-ranking Edge Host is selected. First, we conducted a set of tunning experiments to adjust algorithm parameters. Next, we performed a series of experiments to compare our proposed algorithm with different variants of optimal searching. Our performance comparison aims at validating algorithm efficiency in terms of defined KPIs such as: convergence time, failed and successful relocation executions. Additionally, we examined the influence of Edge topology and user number scaling on algorithm effectiveness.

Finally, in chapter 8, we introduced a new decision algorithm called ER-RL that relies on Reinforcement Learning approach, where the RL agent is learning by experience, playing with the environment, making mistakes, and adjusting policies until it learns the satisfied policy. We go through the entire process of creating an RL-based solution: Firstly, we defined and implemented a training environment that shares similar assumptions as the demonstrator. Among the environment elements, we defined: i) the observation space, which represents the entire set of possible states that the Edge-enabled 5G network environment can be achieve, and ii) a possible action spaces, which include all Edge Hosts in topology. Finally, we defined iii) a reward function, that is a feedback from the environment to the agent that is asessing the agent's decisions. Than, we go through the learning phase, during which we fine-tune the learning hyper-parameters. We developed two versions of Reinforcement Learning agents based on PPO algorithm. The first is a non-masked: native `ER-RL` model, while the second is a hybrid of RL and a heuristic approach, called `ER-RL-Masked`. Masked version allowed us to build an "optimal" model of Reinforcement Learning in the context of fault decisions. This is performed to compare, the effectiveness of non-masked version with some reference, ideal RL algorithm. Finally, we evaluated both RL variants, considering all previously examined heuristic and optimal algorithms. The results provide valuable insights for telecom operators, helping them to determine which algorithm proves more efficient under different assumptions and policies.

9.2 Future work

We conclude this dissertation with a discussion of potential next steps and highlight areas for further research in the area of support of service continuity for Edge services.

Support for stateful application migration in a multi-cloud Edge Computing environment. As already mentioned, the cloud-native systems relies on a container based application deployment were envisioned as a mechanism for running stateless applications. Management systems like Kubernetes are primarily designed to support stateless applications. However, the state of the application, particularly concerning end-user data, is crucial for edge services. The support for state migration along with applications cannot be omitted. There are mechanisms that application developers can use to keep the state of containerized applications in Kubernetes. Whether developers choose to utilize new native mechanisms of Kubernetes such as persistent volumes or decided for alternative options such as distributed and independently synchronized database, that might be deployed outside of Kubernetes cluster, support for state maintenance is crucial.

Another challange related to the execution phase of Edge Relocation is **support for a multi-cloud environment**. Specifically, operators want to deploy Edge infrastructure in a hybrid mode – partially on on-premise servers co-located with gNBs and partially on public cloud providers. This interconnection introduces a range of new research and implementation opportunities related to multi-cluster connectivity, service mesh for observability in a multi-cloud environment, and the management of multiple distributions of Kubernetes, as each cloud provider supports its own custom version. The support for stateful application migration appears to be a crucial point and should be addressed in the short term, while the multi-cloud issue requires more in-depth study and can be set as a mid-term challenge.

Reinforcement Learning for Edge Relocation decision model.

As stated in the research outcomes of the conducted experiments, RL turns out to be a promising choice for the decision model in telco Edge Relocation case. However, the learning process consumes a significant amount of time, and an ideal training involves numerous tuning experiments to adjust proposed learning hyper-parameters, enhance the design of the environment, and redefine the reward function. In this work, following the initial analysis, we chose to focus on Proximal Policy Optimization (PPO) algorithms. Nevertheless, alternative algorithms such as A2C, A3C, SAC, or DDPG exist [18]. A thorough evaluation of each algorithm would provide a comprehensive answer to the question of selecting RL as a method. Since there are many open-source implementations of these algorithms, we can categorize this task as a mid-term challenge. Achieving a solid understanding of each variant of RL is crucial to avoid getting lost in the complexities of hyperparameter.

Another open research challenge related to RL can be formulated as a question: **how to design an ideal model to represent the real environment for training purposes**. Such a model should ensure that no assumptions are missed and all possible behaviors are taken into account. It seems to be a set of promising new technologies in long-term perspective that may support such a modeling. Recently the concept of a digital twin appears, it represents a reliable digital copy of real objects. These objects could include Edge infrastructure as has been discussed in [64], end-users, and their mobility. Use of digital twins into RL training environments could significantly simplify the design process and guarantee high-quality learning by providing an accurate and comprehensive representation of the real-world mapping.

Bibliography

[1] 5g system overview by 3gpp. https://www.3gpp.org/technologies/5g-system-overview, Accesed: 04.03.2024.

[2] Kubernetes architecture. https://kubernetes.io/fr/docs/concepts/architecture/, Accesed: 04.03.2024.

[3] Kubernetes documentation. https://kubernetes.io/docs/home/, Accessed on: 4.03.2024.

[4] Open-Source tools and solutions summary, The Linux Networking Foundation. https://project-emco.io, Accessed on: 4.03.2024.

[5] Orange Open-Source projects. https://github.com/Orange-OpenSource/towards5gs-helm, Accesed: 04.03.2024.

[6] ETSI White Paper No. 28 MEC in 5G networks, June 2018.

[7] AI-Integrated Communications Use Case Interim Report. Technical report, Innovative Optical Wireless Network (IOWN) Global Forum, 2020.

[8] ETSI Multi-access Edge Computing: MEC 5G Integration, Oct. 2020.

[9] Prometheus - observability tool - documentation. https://prometheus.io, Accessed on: 4.03.2024.

[10] 3rd Generation Partnership Project (3GPP). Study on enhancement of support for Edge Computing in 5G Core network (5GC). Technical Specification Group Services and System Aspects 23.748, Dec. 2020.

[11] 3rd Generation Partnership Project (3GPP). 3GPP TS 23.501: System architecture for the 5G System; Stage 2. Technical Specification 23.501, 3rd Generation Partnership Project (3GPP), Feb. 2021.

[12] 3rd Generation Partnership Project (3GPP). 3GPP TS 23.502: Procedures for the 5G System; Stage 2. Technical Specification 23.502, 3rd Generation Partnership Project (3GPP), Dec. 2021.

[13] 3rd Generation Partnership Project (3GPP). Management and orchestration; architecture framework. Technical Specification 28.533, August 2022.

[14] 5G-PPP Cloud Native and 5G Verticals' services, Feb. 2020.

[15] Khizar Abbas et al. Network data analytics function for ibn-based network slice lifecycle management. In *2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 148–153, 2021.

[16] H. Abdah, J. P. Barraca, and R. L. Aguiar. QoS-Aware Service Continuity in the Virtualized Edge. volume 7, pages 51570–51588, 2019.

[17] Amr Ahmed. How edge computing takes on a key role in data-driven business, ey insights. April 2022.

[18] Fadi AlMahamid and Katarina Grolinger. Reinforcement learning algorithms: An overview and classification. In *2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, September 2021.

[19] Gabriele Baldoni et al. Edge computing enhancements in an nfv-based ecosystem for 5g neutral hosts. In *2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pages 1–5, 2018.

[20] Eva Bartz, Thomas Bartz-Beielstein, Martin Zaefferer, and Olaf Mersmann, editors. *Hyperparameter Tuning for Machine and Deep Learning with R - A Practical Guide*. Springer, 2023.

[21] Oussama Bekkouche et al. Toward proactive service relocation for uavs in mec. In *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021.

[22] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. https://arxiv.org/abs/1606.01540, 2016. Accessed: 19 Apr 2023.

[23] Tuo Cao, Zhuzhong Qian, Kun Wu, Mingxian Zhou, and Yibo Jin. Service placement and bandwidth allocation for mec-enabled mobile cloud gaming. In *2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 179–188, 2021.

[24] Hung-Ming Chen, Yung-Feng Lu, Chun-Hung Tsai, and Che-Jung Chang. Design of a mec-integrated 5g mano platform. In *2023 Sixth International Symposium on Computer, Consumer and Control (IS3C)*, pages 210–213, 2023.

[25] Jiasi Chen and Xukan Ran. Deep learning with edge computing: A review. *Proceedings of the IEEE*, 107, 2019.

[26] Xianfu Chen et al. Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning. volume 6, pages 4005–4018, 2019.

[27] Xiaoming Chen et al. Massive access for 5g and beyond. volume 39, pages 615–637, 2021.

[28] Ali Chouman, Dimitrios Michael Manias, and Abdallah Shami. Towards supporting intelligence in 5g/6g core networks: Nwdaf implementation and initial analysis. In *2022 International Wireless Communications and Mobile Computing (IWCMC)*, pages 324–329, 2022.

[29] Mostafa Zaman Chowdhury, Md. Tanvir Hossan, and Yeong Min Jang. Applying model-free reinforcement learning algorithm in network slicing for 5g. In *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pages 1–4, 2019.

[30] Luobin Cui and Ying Tang. Comparing the effectiveness of ppo and its variants in training ai to play game. In *2023 International Conference on Cyber-Physical Social Intelligence (ICCSI)*, pages 521–526, 2023.

[31] Carlos Ruiz De Mendoza, Cristina Cervelló-Pastor, and Sebastià Sallent. Optimal resource placement in 5g/6g mec for connected autonomous vehicles routes powered by deep reinforcement learning. In *2023 IEEE 48th Conference on Local Computer Networks (LCN)*, pages 1–4, 2023.

[32] F. De Vita et al. Using deep reinforcement learning for application relocation in multi-access edge computing. *IEEE Communications Standards Magazine*, pages pages 71–78, 2019.

[33] Yuxuan Deng, Xiuhua Li, Chuan Sun, Jinlong Hao, Xiaofei Wang, and Victor Leung. Deep reinforcement learning for joint service placement and request scheduling in mobile edge computing networks. pages 637–642, 07 2023.

[34] Sabelo Dlamini, Joyce Mwangama, Neco Ventura, and Thomas Magedanz. Design of an autonomous management and orchestration for fog computing. In *2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC)*, pages 1–6, 2018.

[35] Thang Le Duc, , et al. Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey. volume 52, pages 1–39. ACM New York, NY, USA, 2019.

[36] Van-Binh Duong and Younghan Kim. A design of service mesh based 5g core network using cilium. In *2023 International Conference on Information Networking (ICOIN)*, pages 25–28, 2023.

[37] Daniil Ermolenko, Claudia Kilicheva, Ammar Muthanna, and Abdukodir Khakimov. Internet of things services orchestration framework based on kubernetes and edge computing. In *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, pages 12–17, 2021.

[38] ETSI. White paper no. 28 mec in 5g networks, June 2018.

[39] ETSI Multi-access Edge Computing (MEC); Application Mobility Service API. Number DGR/MEC-0021, 1 2020.

[40] ETSI Industry Specification Group for Multi-access Edge Computing (ISG MEC). Multi-access Edge Computing (MEC); Framework and Reference Architecture. Group Specification GS MEC 003, 03 2022.

[41] Linux Networking Foundation. Orange leverages emco to guide autonomous vehicles. *EMCO User Story*, 2024. Accessed on 10th January 2024.

[42] M. Giordani et al. Toward 6G Networks: Use Cases and Technologies. volume 58, pages 55–61, Mar. 2020.

[43] Mohammad Goudarzi, Marimuthu Palaniswami, and Rajkumar Buyya. A distributed deep reinforcement learning technique for application placement in edge and fog computing environments. volume 22, pages 2491–2505, 2023.

[44] Francesc Guim et al. Autonomous lifecycle management for resource-efficient workload orchestration for green edge computing. volume 6, pages 571–582, 2022.

[45] Jeounglak Ha and Young-Il Choi. Support of a multi-access session in 5g mobile network. In *2019 25th Asia-Pacific Conference on Communications (APCC)*, pages 378–383, 2019.

[46] Xiaowu He, Zhiming Yang, Yong Xiang, and Shuang Qian. Nwdaf in 3gpp 5g advanced: A survey. In *2023 3rd International Conference on Electronic Information Engineering and Computer Science (EIECS)*, pages 756–761, 2023.

[47] Baudouin Herlicq et al. NextGenEMO: an Efficient Provisioning of Edge-Native Applications. In *ICC 2022 - IEEE International Conference on Communications*, pages 1924–1929, 2022.

[48] Ashley Hill et al. Stable baselines. https://github.com/hill-a/stable-baselines, accessed: 3.04.2024, 2018.

[49] International Telecommunication Union Radiocommunication Sector (ITU-R). Minimum requirements related to technical performance for IMT-2020 radio interface(s). Report M.2410-0, ITU-R, 11 2017. Version 0.

[50] Youbin Jeon et al. A distributed nwdaf architecture for federated learning in 5g. In *2022 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–2, 2022.

[51] Qingsong Jiao, Botong Xu, and Yunjie Fan. Design of cloud native application architecture based on kubernetes. In *2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pages 494–499, 2021.

[52] Paulo Souza Junior et al. Stateful Container Migration in Geo-Distributed Environments. In *2020 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 49–56, 2020.

[53] Abderaouf Khichane, Ilhem Fajjari, Nadjib Aitsaadi, and Mourad Gueroui. 5gc-observer demonstrator: a non-intrusive observability prototype for cloud native 5g system. In *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, pages 1–3, 2023.

[54] Yohan Kim and Hyuk Lim. Multi-agent reinforcement learning-based resource management for end-to-end network slicing. *IEEE Access*, 9:56178–56190, 2021.

[55] Nane Kratzke and Peter-Christian Quint. Understanding cloud-native applications after 10 years of cloud computing - a systematic mapping study. volume 126, pages 1–16, 2017.

[56] Dapeng Lan et al. Joint optimization of service migration and resource management for vehicular edge computing. pages 155–160, 06 2023.

[57] Eunhee Lee et al. Offloading dependent tasks with mobility in mobile edge computing. In *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6, 2022.

[58] Pengyu Li and Yanxia Xing. Capability exposure vitalizes 5g network. In *2021 International Wireless Communications and Mobile Computing (IWCMC)*, pages 874–878, 2021.

[59] Yiquan Li, Chenxi Yang, Miaoxin Deng, Xue Tang, and Wenzao Li. A dynamic resource optimization scheme for mec task offloading based on policy gradient. In *2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC)*, volume 6, pages 342–345, 2022.

[60] Shanni Liang, Haibin Wan, Tuanfa Qin, Jun Li, and Wen Chen. Multi-user computation offloading for mobile edge computing: A deep reinforcement learning and game theory approach. In *2020 IEEE 20th International Conference on Communication Technology (ICCT)*, pages 1534–1539, 2020.

[61] Soule Issa Loutfi, Ufuk Tureli, and Ibraheem Shayea. Augmented reality with mobility awareness in mobile edge computing over 6g network: A survey. In *2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pages 1–6, 2023.

[62] Chenyu Lu, Zhaowu Huang, Caishan Weng, Feng Jiao, Xiaolin Guo, and Fang Dong. A kubernetes-oriented edge network orchestrator for heterogeneous environment. In *2022 Tenth International Conference on Advanced Cloud and Big Data (CBD)*, pages 300–305, 2022.

[63] Anshita Malviya and Rajendra Kumar Dwivedi. A comparative analysis of container orchestration tools in cloud computing. In *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 698–703, 2022.

[64] Bomin Mao, Jiajia Liu, Yingying Wu, and Nei Kato. Security and privacy on 6g network edge: A survey. volume 25, pages 1095–1127, 2023.

[65] Abdelkader Mekrache, Karim Boutiba, and Adlen Ksentini. Combining network data analytics function and machine learning for abnormal traffic detection in beyond 5g. In *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, pages 1204–1209, 2023.

[66] Yuan Meng, Yang Yang, Sanmukh Kuppannagari, Rajgopal Kannan, and Viktor Prasanna. How to efficiently train your ai agent? characterizing and evaluating deep

reinforcement learning on heterogeneous platforms. In *2020 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–7, 2020.

[67] S. Naganandhini and D. Shanthi. Optimizing replication of data for distributed cloud computing environments: Techniques, challenges, and research gap. In *2023 2nd International Conference on Edge Computing and Applications (ICECAA)*, pages 35–41, 2023.

[68] Mozhgan Navardi, Edward Humes, and Tinoosh Mohsenin. E2edgeai: Energy-efficient edge computing for deployment of vision-based dnns on autonomous tiny drones. In *2022 IEEE/ACM 7th Symposium on Edge Computing (SEC)*, pages 504–509, 2022.

[69] M. Newman and T. McElligott. How to build and operate at the edge. *TM Forum*, Nov. 2020.

[70] Hai T. Nguyen, Tien Van Do, and Csaba Rotter. Scaling upf instances in 5g/6g core with deep reinforcement learning. volume 9, pages 165892–165906, 2021.

[71] Tao Ouyang, Zhi Zhou, and Xu Chen. Follow Me at the Edge: Mobility-Aware Dynamic Service Placement for Mobile Edge Computing. volume 36, pages 2333–2345, 2018.

[72] Chandrasen Pandey, Vaibhav Tiwari, Agbotiname Lucky Imoize, and Diptendu Sinha Roy. Deep reinforcement learning-based resource management for 5g networks: Optimizing embb throughput and urllc latency. In *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, pages 1–6, 2023.

[73] G. Panek, N. El-houda Nouar, I. Fajjari, P. Matysiak, and H. Tarasiuk. Rl-edge relocator: A reinforcement learning-based approach for application relocation in edge-enabled 5g system. In *Proceedings of the IEEE Global Telecommunications Conference - GLOBECOM 2023*, pages 1–6, 2023.

[74] Grzegorz Panek et al. Application relocation in an edge-enabled 5g system: Use cases, architecture, and challenges. *IEEE Communications Magazine*, 60(8):28–34, 2022.

[75] Grzegorz Panek, Piotr Matysiak, Nour El-houda Nouar, Ilhem Fajjari, and Halina Tarasiuk. 5g-edge relocator: A framework for application relocation in edge-enabled 5g

system. In *ICC 2023 - IEEE International Conference on Communications*, pages 4885–4891, 2023.

[76] Petar Popovski, Kasper Fløe Trillingsgaard, Osvaldo Simeone, and Giuseppe Durisi. 5g wireless network slicing for embb, urllc, and mmtc: A communication-theoretic view. volume 6, pages 55765–55779, 2018.

[77] Chengxi Pu, Huiming Xu, Hua Jiang, Daigang Chen, and Pengfei Han. An environment-aware and dynamic compression-based approach for edge computing service migration. In *2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pages 292–297, 2022.

[78] Antonin Raffin et al. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268), 2021.

[79] Syed Salman Raza Naqvi, Wei Liu, Manzoor Ahmed, Muhammad Anwar, Muhammad Mirza, Qibo Sun, and Shangguang Wang. An efficient task offloading scheme in vehicular edge computing. *Journal of Cloud Computing*, 9, 06 2020.

[80] Shuyang Ren and Choonhwa Lee. Blockchain-based service migration for multi-access edge computing. In *2023 International Conference on Information Networking (ICOIN)*, pages 51–55, 2023.

[81] Yinlin Ren, Xingyu Chen, Song Guo, Shaoyong Guo, and Ao Xiong. Blockchain-based vec network trust management: A drl algorithm for vehicular service offloading and migration. volume 70, pages 8148–8160, 2021.

[82] Hans Christian Rudolph, Andreas Kunz, Luigi Lo Iacono, and Hoai Viet Nguyen. Security challenges of the 3gpp 5g service based architecture. volume 3, pages 60–65, 2019.

[83] Chaitanya K. Rudrabhatla. Comparison of zero downtime based deployment techniques in public cloud infrastructure. In *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pages 1082–1086, 2020.

[84] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[85] Syed Danial Ali Shah, Mark A. Gregory, and Shuo Li. Cloud-native network slicing using software defined networking based multi-access edge computing: A survey. volume 9, pages 10903–10924, 2021.

[86] Samir Si-Mohammed et al. UAV mission optimization in 5G: On reducing MEC service relocation. In *IEEE Global Communications Conference*, 2020.

[87] Nina Slamnik-Kriještorac, Miguel Camelo Botero, Luca Cominardi, Steven Latré, and Johann M. Marquez-Barja. An ml-driven framework for edge orchestration in a vehicular nfv mano environment. In *2023 IEEE 20th Consumer Communications Networking Conference (CCNC)*, pages 728–733, 2023.

[88] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, Massachusetts and London, England, 2015. Accessed on 15 January 2023.

[89] Zhiqing Tang et al. Migration modeling and learning algorithms for containers in fog computing. *IEEE Transactions on Computing*, 2018.

[90] ITU-T Unified architecture for machine learning in 5G and future networks, Jan. 2019.

[91] ITU-T Network 2030 Architecture Framework, June 2020.

[92] Mark Towers et al. Gymnasium, An API standard for single-agent reinforcement learning environments. https://zenodo.org/record/8127025, accessed: 3.04.2024, March 2023.

[93] Bao Trinh and Gabriel-Miro Muntean. A deep reinforcement learning-based resource management scheme for sdn-mec-supported xr applications. In *2022 IEEE 19th Annual Consumer Communications Networking Conference (CCNC)*, pages 790–795, 2022.

[94] Nikolaos Tzanis et al. Optimal relocation of virtualized pdc in edge-cloud architectures under dynamic latency conditions. In *2022 International Conference on Electrical, Computer and Energy Technologies*, 2022.

[95] Muhammad Usman, Simone Ferlin, Anna Brunstrom, and Javid Taheri. A survey on observability of distributed edge container-based microservices. volume 10, pages 86904–86919, 2022.

[96] De Vita et al. Using deep reinforcement learning for application relocation in multi-access edge computing. IEEE, 2019.

[97] Jin Wang, Jia Hu, Geyong Min, Albert Y. Zomaya, and Nektarios Georgalas. Fast adaptive task offloading in edge computing based on meta reinforcement learning. *IEEE Transactions on Parallel and Distributed Systems*, 32(1):242–253, 2021.

[98] Yimeng Wang et al. A reinforcement learning approach for online service tree placement in edge computing. In *2019 IEEE 27th International Conference on Network Protocols (ICNP)*, pages 1–6. IEEE, 2019.

[99] Ziying Wu and Danfeng Yan. Deep reinforcement learning-based computation offloading for 5g vehicle-aware multi-access edge computing network. volume 18, pages 26–41, 11 2021.

[100] Xiaojing XIE and Shyam S. Govardhan. A service mesh-based load balancing and task scheduling system for deep learning applications. In *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, pages 843–849, 2020.

[101] Ou Xinjian et al. Research on 5g microservices capability open architecture and deterministic bearing technology. In *2021 IEEE 21st International Conference on Communication Technology (ICCT)*, pages 492–496, 2021.

[102] Ying Xiong, Yulin Sun, Li Xing, and Ying Huang. Extend cloud to edge with kubeedge. In *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 373–377, 2018.

[103] Jiexiong Xu et al. Lightpool: A nvme-of-based high-performance and lightweight storage pool architecture for cloud-native distributed database. In *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 983–995, 2024.

[104] Fangxiaoqi Yu, Haopeng Chen, and Jinqing Xu. Dmpo: Dynamic mobility-aware partial offloading in mobile edge computing. volume 89, pages 722–735, 2018.

[105] Wenhan Zhan et al. Mobility-aware multi-user offloading optimization for mobile edge computing. volume 69, pages 3341–3356, 2020.

[106] Hua Zhang, Sen Xu, Jincan Xin, and Shangkun Xiong. Artificial intelligence in mobile communication network. In *2022 IEEE 8th International Conference on Computer and Communications (ICCC)*, pages 1017–1021, 2022.

[107] David Lister Narothum Saxena Javan Erfanian Zhao, Quan. "6g requirements and design considerations". In *NGMN Alliance*, 2023.

Acronyms

3GPP 3rd Generation Partnership Project.

5GC 5G Core.

AF Application Function.

AMF Access and Mobility Management Function.

CaaS Container as a service.

CAPEX capital expenditure.

CNCF Cloud Native Computing Foundation.

CNI Container Network Interface.

CRI Container Runtime Interface.

DN Data network.

DNS Domain Name System.

EAR Edge Application Relocation.

eMBB Enhanced Mobile Broadband. 20,

EMCO Edge Multi-Cluster Orchestrator.

EO Edge Orchestrator.

ER Edge Relocation.

ERC Edge Relocation Controller.

ETSI European Telecommunications Standards Institute.

ITU Internationl Telecommunication Union.

KPI Key Performance Indicators.

LCM Life-cycle Management.

MANO Management and Orchestration.

MEC Multi-Acces Edge Computing.

MEO MEC Orchestrator.

ML Machine Learning.

mMTC Massive Machine-Type Communications.

NEF Network Exposure Function.

NF Network Function.

NFV Network Function Virtualization. 23,

NGMT Next Generation Mobile Network.

NMT Network and MEC Topology.

NWDAF Network Data Analytics Function.

OS Operating System.

OSS Operations Support System.

PaaS Platform as a service.

PCF Policy Control Function.

PDU Packet Data Unit.

PPO Proximal Policy Optimization.

QoS Quality of Service.

RAN Radio Access Network.

RL Reinforcement Learning.

SBA Service-based Architecture.

SMF Session Management Function..

SSC Service and Session Continuity.

UAV unmanned aerial vehicle.

UDR Unified Data Repository.

UE User Equipment.

UHD Ultra high definition.

UPF User Plane Function.

URLLC Ultra Reliable Low Latency Communications.

V2X Vehicle-to-Everything.

vCPU virtual central processing unit.

VI Virtualized Infrastructure.

VNF Virtual Network Function. 23,

VoD Video on Demand.

VR Virtual Reality.

XR Extended Reality.

List of Figures

List of Tables