©2019 Published in 7th International Symposium on Innovative Technologies in Engineering and Science 22-24 November 2019 (ISITES2019 SanliUrfa - Turkey)
https://doi.org/10.33793/acperpro.02.03.99

5G-destekli Mobil Uç Hesaplama Çerçevesi

Ramazan Akkurt ve M. Fatih Tuysuz
Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Harran Üniversitesi, Şanlıurfa

Abstract

Nowadays, mobile devices have increased rapidly due to the fact that devices connected to the Internet become more portable and can respond to user requests remotely. With this increase, the workload of mobile devices has also increased. However, it is sometimes impossible for mobile devices with limited resources to evaluate some critical and large transactions locally on their own and hence, it may adversely affect the user experience. The MEC (Mobile Edge Computing) system, which enables these processes to be evaluated and returned to the device faster and more efficiently, is now made possible with the 5G-network technology. The MEC concept differs from the previous MCC (Mobile Cloud Computing) system. In this study, the content of the basic MEC system such as the comparison of the MEC system with the MCC, the advantages it offers to mobile devices, usage cases, offloading scenarios, effective resource allocation and mobility management are discussed. In addition, in the scope of the study, high performance studies using MEC system effectively in the literature were examined.

Key words: *MEC, Offloading decisions, MCC, Cellular Networks*

Özet

İnternete bağlanan cihazların günden güne daha taşınabilir hale gelmesi ve kullanıcı isteklerine mobil olarak cevap verebilmesi ile birlikte mobil cihazlar artmıştır. Bu artışla birlikte, cihazların sahip olduğu iş yükü de buna bağlı olarak artmıştır. Ancak kısıtlı kaynaklara sahip mobil cihazların, bazı kritik ve büyük işlemleri local olarak kendi bünyesinde değerlendirebilmesi bazen olanaksız bazen de kullanıcı deneyimini kötü etkilemektedir. Mobil cihazların sahip olduğu bu tarz işlemlerin daha hızlı ve etkili bir şekilde değerlendirilip cihaza tekrar döndürülmesini sağlayan MEC (Mobile Edge Computing) sistemi günümüzde 5G ağ teknolojisi ile mümkün hale geliştirilmiştir. MEC kavramı daha önceki MCC (Mobile Cloud Computing) sistemine gore farklılıklar içermektedir. Bu çalışmada, MEC sisteminin MCC ile kıyası, mobil cihazlara sunduğu avantajlar, kullanım durumları, offloading senaryoları, etkili kaynak tahsisi ve hareketlilik yönetimi gibi temel MEC sisteminin içeriği ele alınmıştır. Ayrıca çalışma kapsamında literatürde MEC sistemini etkin kullanan yüksek performanslı çalışmalarda irdelenmiştir.

Anahtar Kelimeler: *MEC, Yükleme kararları, MCC , Hücresel Ağlar*

1. Giriş

Gelişen teknoloji ile birlikte kullanıcı isteklerindeki veri oranı ve servis kalitesi düzeyi artış göstermektedir. Bu gelişmeler ile birlikte mobil cihazların işlemci (CPU) gücü de artmaktadır. Yüksek pil tüketimi ihtiyacı ise kullanıcıların kendi cihazlarında büyük uygulamaları uzun sure kullanmalarının önüne geçmektedir. Bu yüzden mobil cihazlar, kısa zamanda büyük işlemler

*Corresponding author: Address: Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Harran Üniversitesi, Şanlıurfa, Türkiye. E-mail address: tr.rakkurt@gmail.com, Phone: +905534590712

gerektiren bazı uygulamaların üstesinde gelemeyebilirler. Gözetleme, sanal gerçeklik ve gerçek zamanlı trafik izleme gibi ağ destekli uygulamalar hızlı işlem ve hızlı tepki süresi gerektirmektedir. Mobil Bulut Hesaplama (MCC) çerçevesi bu tarzdaki uygulamaların değerlendirilmesi için az kaynakla çok iş yapabilme mantığına uygun olarak mobil cihazlara kaldıraç görevi sağlamaktadır. Mobil ağlarda son kullanıcıya hizmet sağlayan MCC, son kullanıcı olan mobil cihazların iş yükünü azaltmayı hedefleyen bir sistemdir. Yalnız, MCC sistemindeki veri merkezlerinin son kullanıcıya olan uzaklığı, uzun bir gecikme süresi anlamına gelmektedir. Bu gecikme kötü bir kullanıcı deneyimine ve kötü bir servis kalitesine neden olabilmektedir. Günümüzde 5G ağ teknolojisi kullanılarak son kullanıcıya uzak olan bulut kaynaklarının daha yakın bir konuma taşınması ile Mobil Uç Hesaplama (MEC) kavramı ortaya çıkmıştır. MEC son kullanıcının iletişim halinde olduğu erişim noktalarının (Baz istasyonu, modem, vb.) arkasında çalışan bir sistem olarak tasarlanmıştır.

2. MEC ve Sağladığı Avantajlar

MEC sistemi ile ek işlemci ve depolama gücünün, ağ senaryosunda kullanıcıya mümkün olan en yakın konumda olması sağlanmıştır. Bu sayede, mobil cihazlar kendi lokal kaynaklarında çalıştıramadığı veya yüksek pil gücü ve işlemci gereksinimi nedeniyle zorlanarak/verimsiz çalıştırabildiği uygulamaları kritik cevap süresi içerisinde MEC sisteminin kaynaklarından yararlanarak (MEC sunucularında değerlendirilen uygulama verilerinin tekrar mobil cihaza gönderilmesiyle) sağlayabilmektedirler. MEC sisteminin MCC'ye göre yakınlık, düşük gecikme süresi, yüksek bant genişliği, gerçek-zamanlı trafik yönetimi ve konum farkındalığı gibi temel avantajları bulunmaktadır.

MEC, erişim noktalarının arkasında çalışmasından dolayı MCC'ye göre son kullanıcıya daha yakındır. Bu yakınlık sayesinde MCC'ye göre çok daha küçük bir gecikme süresine sahiptir. Gelişmiş MEC kaynakları son kullanıcıya daha iyi bir bant genişliği de sağlamaktadır. Gerçek zamanlı trafik yönetimi sayesinde, ağ üzerindeki tüm durumlara karşı eylem planı oluşturulabilmektedir. Son olarak sağlanan avantajlardan biri de kullanıcının kendisine en yakın fiziksel lokasyondaki erişim noktasına ulaşmasına ve bu nokta üzerinden iletişim kurulmasına olanak sağlanmasıdır. Mobil bulut hesaplama çerçevesi ile mobil uç hesaplama çerçevesi arasındaki temel farklılıklar Tablo 1'de gösterilmektedir. Tablodan da görülebileceği üzere, MEC sağladığı dağıtık çalışma yapısı nedeniyle tekil hata noktası ihtimaline karşı daha ölçeklenebilir ve uç cihazlara daha hızlı veri transferi gerçekleştirebilir bir yapıda çalışmaktadır. Bu sayede kullanıcıların talep ettiği servis kalitesi düzeyleri MEC ile artırabilmektedir.

Tablo 1. *MCC ve MEC karşılaştırılması*

Özellikler	MCC	MEC
Yayılma	Merkezi	Dağıtık
Son kullanıcıya uzaklık	Yüksek	Düşük
Gecikme	Yüksek	Düşük
İşlem Gücü/Kapasitesi	Yüksek	Düşük
Depolama Kapasitesi	Yüksek	Düşük

Mobil cihazlardaki verilerin MEC sisteminde değerlendirilmesi için verilerin MEC sistemine iletilmesi gerekmektedir. Bu verilerin iletilme şekli boşaltma (offloading) olarak gerçekleşmektedir. Offloading veri iletimi için yarı-offlading ve tam-offloading olmak üzere 2 farklı senaryo üzerinde durulmaktadır. Bu iki senaryonun ne olduğu ve konu hakkında literatürde ne tarz çalışmalar yapıldığı Bölüm II'de özetlenmiştir. MEC sistem kaynaklarının cihazlara nasıl tahsis edildiği ve literatürde ne tarz çalışmalar yapıldığı ise Bölüm III'te aktarılmıştır. Son olarak, mobil kullanıcılarının hareketliliğinin sistem içinde nasıl yönetildiği ve bu konu için literatürdeki temel çözüm önerilerinin neler olduğu Bölüm IV'te incelenmiştir.

3. Offloading Veri İletimi

Offloading veri iletimi Küçük Hücre Ağları (SCN), Wifi offloading ve Ad-Hoc temelli offloading olarak üç şekilde gerçekleştirilmektedir. Veri iletimi SCN için iç mekanlarda kapsama alanı sağlayan küçük hücrelere sahip femtocell'ler ile gerçekleştirilir. MEC sisteminde offloading veri iletimi yukarda bahsedildiği gibi iki şekilde yapılmaktadır.

A. Tam-Offloading

Büyük işlemler veya hızlı cevap süresi gerektiren uygulamaların iş yükünün mobil cihazda mı yoksa MEC serverlarında mı değerlendirileceği ile ilgili bir karar verilmelidir. Bu kararın verilmesinde etki sahibi olan çok fazla etmen vardır. En temel etmenlerden ikisi gecikme süresi ve harcanacak enerji miktarıdır. Eğer tüm işlemlerin MEC sisteminde yapılması daha verimli ise tüm veriler tam-offloading olarak MEC serverlara iletilir. En verimli ve uygun offloading kararlarının verilmesi kullanıcı deneyimi ve servis kalitesi açısından oldukça kritiktir. Bu kapsamda daha verimli offloading kararları için literatürde gecikme süresini minumuma yaklaştırmayı amaçlayan bazı çalışmalar önerilmiştir.

Çalışma [1]'de tek değişkenli bir arama algoritması önerilmiştir. Algoritma yürütme gecikmesini (execution delay) minimize etmek amacıyla uygulamanın buffer kuyruk durumu, MEC server ve cihazın mevcut işlem gücü, MEC server ve cihaz arasındaki kanalın karakteristiğine göre en uygun offloading karar verme modülünü bulmayı hedeflemektedir. Cihaz, offloading kararlarını bu modüle göre kendi kendine yapar. Bu modül her zaman slotunda bufferda bekleyen uygulamanın gecikme süresini minimize edecek şekilde, uygulamanın lokal cihazda mı , yoksa

MEC'te mi çalıştırılacağına karar verir. Çalışmadaki simülasyon sonuçlarına göre lokalde çalıştırılması kararı verilen bir uygulamanın gecikme süresinin %80 azaldığı, MEC'te çalıştırılmasına karar verilen bir uygulamanın gecikme süresinin ise %44 azaldığı gösterilmiştir.

Çalışma [2]'de ise yürütme gecikmesini azaltmak amacıyla, tam-offload olarak karar verilmiş hatalı uygulamaları azaltmayı hedefleyen bir yaklaşım önerilmiştir. Yazarlar düşük-karmaşıklığa sahip Lyapunov optimization based dynamic computation offloading (LODCO) algoritması önermiştir. LODCO ile, eğer işlemler cihazda yapılıyorsa, her bir zaman slotunda bir offloading kararı verilir ve daha sonra cihaz için CPU cycle'ı tahsis edilir. Aksi durumda eğer işlemler MEC'te yapılıyorsa bu işlem için bir iletim gücü tahsis edilir. Önerilen LODCO algoritmasının simülasyon sonuçlarına göre MEC'teki yürütme gecikmesi %64 oranında azalma göstermektedir.

Çalışma [3]'te ise UE (User Equipment) için uygun bir yürütme gecikmesinde, enerji tüketimini minimize etmek amacıyla bir optimizasyon problemi formüle edilmiştir. Bu optimizasyon problemi Markov chain process min-max aralığında formüle edilmiştir. Bu optimizasyon problemini çözmek için 2 adet kaynak tahsis stratejisi tanıtılmıştır. İlk stratejide ağ, online bir öğrenmeye dayalı olarak UE'de çalışan programa dinamik bir şekilde uyum sağlar. İkinci strateji, uygulamaya ilişkin belirli bir bilgi seviyesinden (her slotta paket cinsinden ölçülen iletim oranı, radyo kanalı durumu, vb.) yararlanan önceden hesaplanmış offline bir stratejidir. Simülasyon sonuçlarına göre önceden hesaplanmış offline strateji, online öğrenme stratejisine göre , düşük ve orta iletim oranı için %50 daha iyi performans sağlamıştır.

B. Yarı-offloading

Daha önceden de belirtildiği gibi, etkili bir offloading kararı için en önemli nokta, kullanıcı deneyimini ve servis kalitesini en üst seviyede tutabilmektir. Bazı durumlarda tüm işlemlerin MEC sunucularında değerlendirilmesi yerine, bir kısmının lokal cihazlarda, geriye kalan kısmının ise MEC sunucularında değerlendirilmesi daha verimli bir strateji olabilir. Bu tarz yarı/kısmi offloading kararları için literatürde bazı çalışmalar önerilmiştir.

Çalışma [4]'te TDMA (Time Division Multiple Access) yapısına dayalı çoklu UE sistemine uygun bir offloading modülü önerilmiştir. T saniye süre boyunca, zaman slotlara bölünmüştür. Her zaman slotunda, UE'ler kanal kalitesine veya lokal cihazdaki enerji tüketimine göre verilerini parçalar halinde MEC'e offload edebilirler. Bu bağlamda , optimal bir kaynak tahsis politikası önerilmiştir. Bu politika, lokalde optimal bir gecikme süresini sağlayabilen uygulamaları daha düşük bir derece ile öncelik sırasına koyma politikasıdır. Eğer uygulama , gereken sürede lokalde çalıştırılabiliyorsa düşük önceliğe sahip olur. Kendi lokal kaynağında optimal bir gecikme süresini garanti edemeyen uygulamalar ise yüksek önceliğe sahip olur. Bu şekildeki bir öncelik-sonralık sırasına sokulan uygulamalar için, bir eşik değere göre kaynak tahsis politikası önerilmiştir. Eğer UE önceliği verilen eşik değerden daha büyükse tam-offloading, eğer daha küçükse uygulamanın optimal gecikme süresine uygun olarak minimum sayıda yarı/kısmi offloading yapılabilir. Simülasyon sonuçlarına göre bu kaynak tahsisi modülü diğer çalışmalara

göre ihmal edilebilir oranda bir enerji tüketimine sahiptir.

Çalışma [5]'te DVS (Dynamic Voltage Scaling) teknikleri kullanılarak, optimal gecikme süresine uygun enerji tüketimini minimize etmeyi amaçlayan bir offloading kararı yaklaşımı önerilmiştir. Bu yaklaşımda en optimal enerji tüketimi baz alınarak kısmi/yarı offloading şeması önerilmiştir. Uygulamanın izin verilen maksimum gecikmesi Lmax olarak temsil edilmiştir. Önerilen bu şemanın amacı, uygulamanın gerçek gecikme süresi olan Lmax'a daima eşit olduğunu garanti etmektir. Bu sayede kullanıcının Servis Kalitesi (QoS) düzeyi etkilenmeden enerji minimizasyonu gerçekleştirilerek offloading kararlarının etkili bir şekilde gerçekleşmesi sağlanmıştır. Simülasyon sonuçlarına göre servis kalitesi sabit kalarak, enerji tüketiminin önceki duruma göre azaldığı gösterilmiştir.

4. Kaynak Tahsisi

MEC sisteminde çalıştırılacak uygulamalar için veri boşaltma işlemi haricinde ayrıca MEC tarafından bir kaynak tahsisi de yapılmalıdır. Bu kapsamda çalışma [6]'da önerilen yaklaşımın temel amacı, gecikme süresine uygun bir şekilde MEC tarafından hizmet verilen uygulama sayısının maksimize edilmesidir. Uygulamalar için gereken optimal gecikme sürelerine göre, uygulamaları bir öncelik sırasına sokmak ve kaynak tahsisini buna göre yapmaktır. Veri iletimi gerçekleştirilmiş uygulamalar ilk olarak MEC içiresindeki scheduler'a gönderilir. Scheduler hesaplama kaynağına sahip bir düğüm olup olmadığını kontrol eder. Eğer yeterli kaynağa sahip bir düğüm varsa o düğüme bir sanal makine (VM) tahsis edilir, ardından bu MEC düğümünde uygulama yürütülür. Son olarak sonuçlar UE'ye geri gönderilir. Eğer yeterli bir MEC kaynağı (sunucu) yoksa, scheduler uygulamayı uzaktaki merkezi buluta (Centralized Cloud) gönderir. Önerilen çalışma kapsamında gerçekleştirilen simülasyon sonuçları, uygulamanın tamamının yürütülmesi için gereken toplam sürenin %25 oranında azaldığını göstermiştir. Uygulamaların gereksinimi olan optimal gecikme süresine göre, MEC'teki uygulama sayısını maksimize etmeyi amaçlayan bu yaklaşımın bozulmaması için yazarlar ayrıca önceliklendirme bazlı bir işbirliği politikası önermişlerdir. Bu politika, her bir öncelik seviyesi için buffer eşik değeri belirlemektedir. Eğer buffer tamamen dolu ise, uygulamalar buluta gönderilir. Bu çalışmada, buffer eşik değeri büyüklüğü düşük karmaşıklığa sahip yinelemeli bir algoritma tarafından hesaplanmaktadır.

Çalışma [7] ise yoğun şekilde UE'ler bulunan kablosuz erişim alanları için önerilmiş bir çalışmadır. Birden fazla MEC sunucuya erişebilen UE'lerin, kendisi için en optimal gecikme süresini ve güç tüketimini garanti eden MEC sunucudan servis alması öngörülmüştür. Önerilen bu optimal politikada Markov karar işlem çerçevesi kullanılmıştır. Ancak kullanılan bu metot, MEC sunucuların sayısı arttıkça iletişim yükünün artmasına ve işlem karmaşıklığına sebep olmuştur. Bu sorunu aşmak için yeni bir indeksleme politikası geliştirilmiştir. Her sunucu kendi durumuna (kaynak durumu) göre kendi indeksini hesaplar ve bu indeksler tüm sunucular tarafından yayılır. UE'ler kendisi için gecikme ve güç tüketimini minimize eden en uygun sunucuyu bu indekslere bakarak seçer.

Çalışma [8]'de MEC kaynaklarının femtocell gibi daha iç-mekansal erişim noktalarında bulunmasını öneren bir yaklaşım sunulmuştur. Femto-clouds adı verilen bu yaklaşım femtocell'lerin işbirliği içinde çalışması ilkesine dayanan bir oyun teorisi yaklaşımıdır. Her femto-clouds kendi kapsama alanındaki UE'lere hizmet vermekle sorumludur. Eğer boş ve herhangi bir UE'ye hizmet vermeyen bir femto-clouds varsa bunu diğer femto-clouds ile paylaşır ve diğer femto-clouds'ın kapsama alanındaki UE'ye servis sağlar. Bu durumda kendi kaynağını başka femto-clouds ile paylaşarak işbirlikçi bir şekilde hizmet verir. Sayısal sonuçlara göre, yalnızca tek bir femto-cloud'ta servis alan bir UE'nin yürütme gecikmesinin %50 oranında azaldığı gösterilmiştir. Ayrıca, ekstra işbirliği içinde birden fazla femto-clouds ile %25 daha düşük bir yürütme gecikme süresine sahip olduğu da çalışma kapsamında gösterilmiştir.

Çalışma [15]'te yazarlar SceNB'lerin hesaplama ve iletişim yükülerine göre en uygun SceNB'nin seçilmesini sağlayan bir algoritma önermişlerdir.Application Considering Algorithm (ACA) bunu yaparken offloaded uygulamaların bilgilerini kullanarak yapar.Bu bilgiler genellikle MEC server'a transfer edilen uygulamanın byte sayısı ve uygulama tarafından kabul edilen maksimum gecikme süresi gibi kritik bilgilerdir.Hesaplama için SCeNB'lerin seçimi,VM migrasyonunun maliyetini önlemek için offloading öncelikli statik bir yoldan yapılır.Ayrıca iki ayrı backhaul, düşük iş yüklü ADSL ve yüksek kaliteli gigabit passive optical network (GPON) için performans değerlendirmesi yapılır.Şuana kadar önerilen bu ACA algoritması UE'lerin saniye başına offloaded görev sayısının maximum 6 tane olduğu durumlarda %100 olarak uygun kaynak tahsisi yapılabildiği gösterilmiştir.

Çalışma [16]'da UE uygulamalarının yürütülme gecikmesi , hem iletişim hemde hesaplama kaynaklarındaki aşırı yüklenmeyi ve VM migrasyon maaliyetini minimize etmeyi amaçlayan bir yaklaşım önerilmiştir.Problemin tamamı MDP kullanılarak çözülmüş ve VM tahsisi olarak formüle edilmiştir.Bu senaryoda aynı konumda bulunan 2 UE için birinci UE'ye SCeNB1'in VM'i tahsis edilir.Bu bağlamda SCeNB1 üzerinde oluşacak aşırı yükün önlenmesi için diğer UE için SCeNB1'ın komşusu olan ve yüksek backhaul kalitesine sahip olan SCeNB2'den bir VM tahsis edilir.Aynı konumda bulunmalarına rağmen bu 2 UE'de farklı SCeNB'lerden hizmet alırmış olurlar.Simülasyon sonuçlarına göre eğer yeterli bir hesaplama gücüne sahip bir SCeNB varsa yüksek VM migrasyon maaliyetine sahip olsa bile bu SCeNB'den hizmet almanın tercih edildiği gösterilmiştir.

5. Hareketlilik Yönetimi

Geleneksel bir hücresel ağda, kullanıcıların konum değiştirmeleri durumunda, servis kalitesi ve devamlılığı garanti edilmelidir. Handover adı verilen el-değiştirme prosedürü ile herhangi 2 veya daha fazla kapsama alanı arasında konum değiştiren UE'lerin servis migrasyonu da bulundukları kapsama alanlarına göre değişmektedir. Offload veri iletim servisini kullanan bir cihazın, aynı şekilde offloading işlemi kesintisiz olarak devam etmelidir.

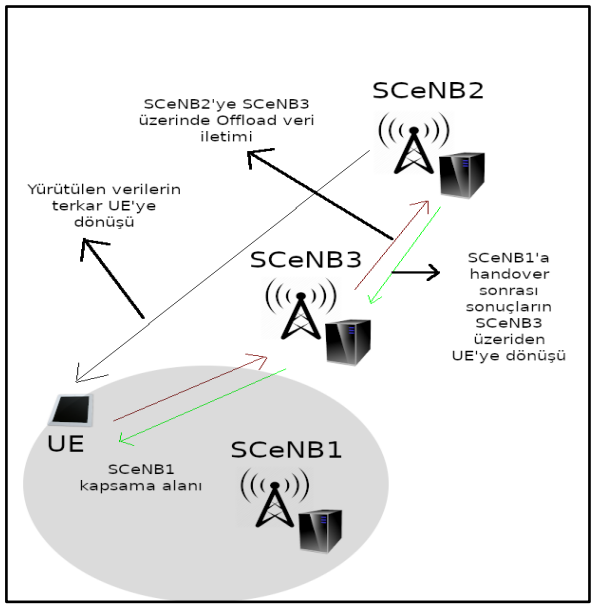
Hareketlilik senaryoları için literatürde bazı çözüm önerileri geliştirilmiştir. UE'nin hareketinin düşük ve sınırlı olduğu durumlarda (örneğin düşük bir hızda AVM içerisinde hareket eden bir UE) düzgün ayarlanan bir iletim gücü ile ya da komşu bir baz istasyona el-değiştirme ile servis devamlılığı sağlanabilir. Bu kapsamda çalışma [9]'da hassas gecikme gereksinimi olan gerçek zamanlı uygulamaların offloading iletimini yönetmeye yardımcı olan cloud-aware power control (CaPC) algoritması önerilmiştir. CaPC'nin amacı belli bir gecikme sınırında MEC'e offloading iletim yapan uygulamaların sayısını maksimize etmektir. Algoritma mümkün olan durumlarda düzgün bir güç kontrol ayarlaması ile kapsama alanını genişleterek hareketli olan UE'nin el-değiştirme yapmasına gerek kalmadan hizmet almaya devam etmesini sağlayabilmektedir. CaPC algoritması baz istasyonlarının iletim güçlerini karşılaştırarak baz istasyonları için default (varsayılan) bir iletim gücü hesaplar. Bu değer ortama en çok sinyal gönderen baz istasyonundan alınan güç seviyesine ya da bir den fazla baz istasyonu tarafından gönderilen sinyal seviyesine bağlıdır. Düşük bir SINR (Signal to interference and noise ratio) seviyesinde offload iletim gerçekleşemeyeceğinden dolayı iyi ayarlanmış bir iletim gücü hesabı kritiktir. Mümkün olan durumlarda, CaPC gecikmeye uygun bir şekilde MEC'te yürütülen uygulamaların %95'ini başarılı bir şekilde tekrar UE'lere iletmektedir. Aksi durumlarda, sinyal gücü kontrolü ile aşağı yukarı %80 oranında başarılı bir şekilde iletim gerçekleşebilmektedir. Bu kapsamda yazarlar tarafından CaPC algoritmasının yaklaşık %15 daha fazla verim sağladığı bildirilmiştir.

Yukardaki çalışma sınırlı hareket alanındaki UE için önerilen bir yaklaşımdır. Aksi durumlarda, yani hareketin sınırsız olduğu durumlarda ise güç kontrolü sağlanan servisin devamlılığı için yeterli olmayabilir. Bu tarz durumlarda kullanılan servisin devamlılığı için VM migrasyonu düşünülmüştür. VM migrasyonu bir maliyete sahiptir. Bu maliyet düğümler arasındaki VM'in iletimi için harcanan backhaul kaynakları ve migrasyon için gereken süreyi temsil eder. Bu kapsamda çalışma [10]'da VM migrasyonunun UE'lerin performansına nasıl etki ettiği ele alınmıştır. Yazarlar Markov Chain'e dayalı bir analitik model önermiştir.VM migrasyonu olmadan, UE'nin optimum MEC'e bağlanma olasılığı, UE ve baz istasyonu arasındaki artan hop sayısı ile azalır. Bu aynı zamanda gecikmenin artmasına da sebep olur. Ancak yüksek maliyetli VM migrasyonunda, UE'nin optimal MEC'e bağlanması en düşük gecikme süresi ile mümkün olduğu gösterilmiştir. Doğal olarak düşük bir gecikme istenen durum olduğu için her bir el-değiştirme durumunda VM migrasyonu gerçekleştirilmelidir.

Yukardaki çalışmada daha genel bir ön hazırlık olarak VM migrasyonun avantaj veya dezavantajlı olduğu durumlara odaklanılmıştır. Çalışma [11]'de ise VM migrasyonun yapılıp yapılmayacağına karar veren farklı bir politika önerilmiştir. Bu karar politikası VM migrasyonunun maliyeti ve kazancı üzerinden yapılmıştır. Yazarlar el-değiştirme işlemi gerçekleştikten sonra optimal bir eşik-değeri bulmayı amaçlayan, continious time MDP (CTMDP) bazlı bir karar politikası formüle etmiştir. Optimal eşik-değeri karar politikası VM migrasyonu olup olmayacağına karar vermektedir. Örnek olarak A baz istasyonundan servis almaya başlayan kullanıcı A'dan çıkıp B baz istasyonunun kapsama alanına geçerse, el-değiştirme durumu gerçekleşir. Bu karar politkasına göre eğer ki migrasyon kazanç değeri, migrasyon maliyetinden daha küçük ise VM migrasyonu diğer baz istasyonuna iletilmeye başlar. Aksi durumda eski MEC sunucudan servis sağlanmaya devam eder. Simülasyon sonuçlarına göre

önerilen bu politika VM migrasyonu içermeyen stratejiye göre beklenen maksimum kazancı oldukça başarılı bir şekilde garanti etmektedir.

Çalışma [12] ise VM migrasyonu için gereken toplam maliyeti minimize etmeyi amaçlamaktadır. VM migrasyonu için optimal eşik-değeri bulan iteratif bir algoritma önerilmiştir. Algortimanın zaman karmaşıklığı O(M|N)'dir. M ve N maksimum negatif offset değerleridir. Sonuçlara göre algoritma VM migrasyonu olacak veya olmayacak tüm kararlar için daha iyi performans göstermektedir. Düğümler arasında migrasyonu gereken büyük miktardaki veriler için VM migrasyonu yeterli değildir. Bu tarz büyük verilerin işlem süresi uzun dakikalar hatta saatler alabilir. Bu migrasyon saniyeler sürse bile, gerçek zamanlı uygulamalar offload edilemez. Çünkü işlemler devam ederken kullanıcı hareketi de devam edebilir. Sonuç olarak cihaza tekrar gönderilmesi gereken veriler kullanıcının hareketliliği sebebiyle optimal gecikme süresini aşabilir. Böyle durumlarda yeni bir iletim yolu bulmak ya da yolu optimize etmek daha iyi bir seçenektir.
Çalışma [13]'te MEC'de yürütülen uygulamaların tekrar UE'ye teslim edilmesi için bir yol seçimi algoritması önerilmiştir. Algoritmanın amacı hem backhaul hem de radyo link kalitesine göre iletim gecikmesini minimize etmektir. Bu yaklaşımdaki çözüm gecikmeyi minimize eden yeni bir SceNB'ye el-değiştirme gerçekleşmesini sağlamaktır. Çalışmaya ait senaryo Şekil 1'de gösterilmiştir. Uygulama SCeNB2 de yürütülmektedir. Yürütmesi biten veriler direkt olarak SCeNB2 tarafından kullanıcıya geri gönderilir. Yalnız, SCeNB2 verileri kullanıcıdan SCeNB3 aracılığıyla alınır. Kullanıcı hareketine devam eder ve SCeNB3 kapsama alanına girer ve mevcut MBS'ye(SCeNB3) el-değiştirme gerçekleşir. Son durumda tüm tamamlanan veriler ise radyo linkleri sayesinde SCeNB3'den alınır.Bu şekildeki bir yaklaşımla büyük yürütme zamanına sahip uygulamalar, gecikme süresini aşmadan tekrar kullanıcıya gönderilir. Sonuçlara göre bir cihazın ilk başta servis aldığı MEC'ten sonuçları dolaylı yoldan da olsa tekrar geri alması durumu için iletim gecikmesinin %9 azaltıldığı gösterilmiştir.



Şekil 1. Yol seçim algoritmasının çalışma prensibi

Çalışma [14]'te ise daha genel bir VM migrasyonu düşünülmüştür.Yazar VM migrasyonu için MDP kullanarak sıralı karar verme problemi ve en optimal karar politikasını bulan bir algortima tanımlamıştır.Algortimanın karmaşıklığı O()'tür.N değeri eNB ve UE arasındaki hop sayısını ifade eder.Algortimanın zaman karmaşıklığı ise O() olarak ifade edilmiştir.
Önerilen bu yaklaşımın temel amacı minumum hop sayısını bularak optimal bir gecikmede VM migrasyonun yapılmasını sağlamaktır.Sayısal sonuçlara göre önerilen bu migrasyon stratejisi diğer çalışmalarla karşılaştırıldığında kabaca maaliyeti %35 azalttığı gösterilmiştir.

6. Sonuçlar

MEC sistemi mobil cihazlara daha yakın noktalarda konumlandırıldıklarından, bulut kaynaklarına göre işlem ve gecikme süreleri bakımından daha avantajlıdır. Bu sayede yüksek işlem yükü olan uygulamaların optimum gecikmelerde, mobil cihazlarda çalıştırabilmesine olanak sağlanır. Bu işlem ayrıca cihazların enerji tüketimini azaltarak, pil ömürlerinin uzun olmasına imkan sağlar. İrdelenen çalışmalar ışığında bu sistemin taşınabilir cihazlara büyük avantajlar getireceği öngörülmektedir. Tam olarak olgunlaşmamış olan bu teknolojinin gelecek çalışmalarda büyük ilgi göreceği aşikardır. Daha kısa gecikme süreleri veya daha az enerji tüketimi gibi mobil cihazların karşı karşıya kaldığı zorluklara çözüm getiren yeni çalışmalar ile birlikte, yıllar içinde MEC sisteminin daha olgun ve gelişmiş şekilde, mobil sistemlerde büyük bir araştırma konusu olacağı tahmin edilmektedir. Araştırma konusu olarak offloading kararları, hesaplamalı kaynak tahsisi ve mobilite yönetimi gibi temel MEC sisteminin işleyişini doğrudan ilgilendiren konuların gelişimi gelecek çalışmalarda MEC için kritik önem taşımaktadır.

Kaynaklar

[1] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Barcelona, Spain, 2016, pp. 1451–1455.

[2] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," IEEE J. Sel. Areas Commun., vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[3] M. Kamoun, W. Labidi, and M. Sarkiss, “Joint resource allocation and offloading strategies in cloud enabled cellular networks,” in Proc. IEEE Int. Conf. Commun. (ICC), London, U.K., 2015, pp. 5529–5534.
[4] C. You and K. Huang, “Multiuser resource allocation for mobileedge computation offloading,” in Proc. IEEE Glob. Commun. Conf. (GLOBECOM), Washington, DC, USA, 2016, pp. 1–6.
[5] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, “Mobile-edge computing: Partial computation offloading using dynamic voltage scaling,” IEEE Trans. Commun., vol. 64, no. 10, pp. 4268–4282, Oct. 2016
[6] T. Zhao, S. Zhou, X. Guo, Y. Zhao, and Z. Niu, “A cooperative scheduling scheme of local cloud and Internet cloud for delay-aware mobile cloud computing,” in Proc. IEEE Globecom Workshops (GC Wkshps), San Diego, CA, USA, 2015, pp. 1–6.
[7] X. Guo, R. Singh, T. Zhao, and Z. Niu, “An index based task assignment policy for achieving optimal power-delay tradeoff in edge cloud systems,” in Proc. IEEE Int. Conf. Commun. (ICC), Kuala Lumpur, Malaysia, 2016, pp. 1–7.
[8] S. M. S. Tanzil, O. N. Gharehshiran, and V. Krishnamurthy, “Femtocloud formation: A coalitional game-theoretic approach,” in Proc. IEEE Glob. Commun. Conf. (GLOBECOM), San Diego, CA, USA, 2015, pp. 1–6.
[9] P. Mach and Z. Becvar, “Cloud-aware power control for cloud-enabled small cells,” in Proc. IEEE Globecom Workshops (GC Wkshps), Austin, TX, USA, 2014, pp. 1038–1043.
[10] T. Taleb and A. Ksentini, “An analytical model for follow me cloud,” in Proc. IEEE Glob. Commun. Conf. (GLOBECOM), Atlanta, GA, USA, 2013, pp. 1291–1296.
[11] A. Ksentini, T. Taleb, and M. Chen, “A Markov decision process-based service migration procedure for follow me cloud,” in Proc. IEEE Int. Conf. Commun. (ICC), Sydney, NSW, Australia, 2014, pp. 1350–1354.
[12] S. Wang et al., “Mobility-induced service migration in mobile microclouds,” in Proc. IEEE Mil. Commun. Conf., Baltimore, MD, USA, 2014, pp. 835–840.
[13] Z. Becvar, J. Plachy, and P. Mach, “Path selection using handover in mobile networks with cloud-enabled small cells,” in Proc. IEEE Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC), Washington, DC, USA, 2014, pp. 1480–1485.
[14] S. Wang et al., “Dynamic service migration in mobile edge-clouds,” in Proc. Netw. Conf. (IFIP Networking), Toulouse, France, 2015, pp. 1–9.
[15] M. Vondra and Z. Becvar, “QoS-ensuring distribution of computation load among cloud-enabled small cells,” in Proc. IEEE Int. Conf. Cloud Netw. (CloudNet), 2014, pp. 197–203.
[16] V. Di Valerio and F. L. Presti, “Optimal virtual machines allocation in mobile femto-cloud computing: An MDP approach,” in Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW), Istanbul, Turkey, 2014, pp. 7–11.