İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DOKTORA TEZİ

GELECEK NESİL KABLOSUZ AĞLAR İÇİN YENİ BİR MOBİLİTE YÖNETİM SİSTEMİNİN GELİŞTİRİLMESİ

Bilgisayar. Yük. Müh. Gülsüm Zeynep GÜRKAŞ AYDIN
Bilgisayar Mühendisliği Anabilim Dalı

Danışmanlar
Prof. Dr. A.Halim ZAİM
Doç. Dr. Hakima CHAOUCHI

Mart, 2011

İSTANBUL

Bu çalışma 28/03/2011 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Programında Doktora Tezi olarak kabul edilmiştir.

Tez Jürisi

Prof.Dr.A.Halim ZAİM (Danışman)
İstanbul Üniversitesi
Mühendislik Fakültesi

Prof.Dr. Hakan Ali ÇIRPAN
İstanbul Teknik Üniversitesi
Elektrik-Elektronik Fakültesi

Prof.Dr.İlhami YAVUZ
Maltepe Üniversitesi
Mühendislik ve Doğa Bilimleri Fakültesi

Prof.Dr.Ahmet SERTBAŞ
İstanbul Üniversitesi
Mühendislik Fakültesi

Prof.Dr.Selim AKYOKUŞ
Doğuş Üniversitesi
Mühendislik Fakültesi

ÖNSÖZ

Bu çalışma İstanbul Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında yapılan "Gelecek Nesil Kablosuz Ağlar İçin Yeni Bir Mobilite Yönetim Sisteminin Geliştirilmesi" adlı doktora tez çalışmasını içermektedir.

Bu tez çalışması boyunca gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Sayın Prof. Dr. A.Halim ZAİM'e, tez izleme komitemde yer alan Prof. Dr. İlhami YAVUZ'a ve Prof. Dr. Hakan Ali ÇIRPAN'a, bu çalışmanın yürütülmesinde bilgilerini ve manevi desteklerini esirgemeyen çok değerli çalışma arkadaşlarım Araş. Gör. Özgür Can TURNA ve Araş. Gör. Ergün GÜMÜŞ'e en içten dileklerimle teşekkür ediyorum.

Her türlü desteklerinden dolayı Telecom SudParis Enstitüsü öğretim görevlilerinden ikinci danışmanım Doç.Dr. Hakima CHAOUCHI'ye, manevi ve maddi desteklerini esirgemeyen Prof.Dr. Tülin ATMACA'ya, kendilerinin yanında yürüttüğüm yurtdışı araştırmalarımda bana maddi ve manevi desteklerde bulanan TUBİTAK'a ve Tinçel Kültür Vakfı'na en içten dileklerimle teşekkür ediyorum.

Tüm eğitim hayatım boyunca bana yol gösteren, destek olan, her zaman yanımda olan başta annem ve babam olmak üzere, manevi desteklerini esirgemeyen GÜRKAŞ ve AYDIN ailelerine içten dileklerimle teşekkürlerimi sunuyorum.

Bu tez çalışmasında her türlü desteğini, bilgisini ve zamanını esirgemeyen, tüm sıkıntılarıma benimle birlikte katlanan sevgili eşim M.Ali AYDIN'a teşekkürü borç bilirim.

Bu çalışmamı sevgili eşime ve ailemizin yeni bireyine armağan ediyorum.

Mart 2011 **Gülsüm Zeynep GÜRKAŞ AYDIN**

İÇİNDEKİLER

ŞEKİL LİSTESİ

TABLO LİSTESİ

KISALTMA LİSTESİ

AP : Access Point (Erişim Noktası)
BE : Bağlantı Base Exchange (Temel Mesaj Değişimi)
CN : Corresponding Node (Karşı Düğüm)
DH-HIP : Dynamic Hierarchical HIP (Dinamik Hiyerarşik HIP)
DNS : Domain Name Server (Alan Adı Sunucusu)
e-HIP : Early Update for HIP (HIP için Erken Güncelleme)
ESP : Encapsulating Security Paylod (Kapsüllenmiş Güvenlik Yükü)
EU : Early Update (Erken Güncelleme)
FMIP : Fast Mobile IP (Hızlı Mobil IP)
FU : Finish Update (Güncelleme Bitişi)
H1 : Hierarchy Level 1 (Hiyerarşi Seviyesi 1)
H1H : H1 Handover (H1 Yer Değiştirme)
H2 : Hierarchy Level 2 (Hiyerarşi Seviyesi 2)
H2H : H2 Handover (H2 Yer Değiştirme)
HI : Host Identity (Konuk Kimliği)
HIP : Host Identity Protocol (Konuk Kimlik Protokolü)
HIT : Host Identity Tag (Konuk Kimlik Etiketi)
HL : Handover Latency (Yer Değiştirme Gecikmesi)
HO : Handover/Handoff (Yer Değiştirme)
I : Initiator (Başlatan)
ICMP : Internet Control Message Protocol (Internet Kontrol Mesaj Protokolü)
IETF : Internet Engineering Task Force (Internet Mühendisliği Çalışma Grubu)
INET FW : INET Framework (INET Çerçevesi)
IRTF : Internet Research Task Force (Internnet Araştırmaları Çalışma Grubu)
LRVS : Local Rendezvous Server (Yerel Randevu Sunucusu)
LSI : Local Scope Identity (Yerel Kapsamlı Kimlik)
MIP : Mobile IP (Mobil IP)
MN : Mobile Node (Mobil Düğüm)
NGWN : Next Generation Wiress Networks (Gelecek Nesil Kablosuz Ağlar)
p-eHIP : Predictive eHIP (Tahminli eHIP)
PMIPv6 : Proxy Mobile IPv6 (Proksi Mobil IPv6)
R : Responder (Cevap Veren)
RA : Router Advertisement (Yönlendirici Bilgilendirme)
REGR : Registrar (Kayıt Olunan)
REQR : Requester (Talep Eden)
RVS : Rendezvous Server (Randevu Sunucusu)

SA	: Security Association (Güvenlik Bağlantısı)
SAP	: Service Announcement Packet (Servis Anons Paketi)
TCP/IP	: Transmission Control Protocol/Internet Protocol (İletim Kontrol Protokolü/Internet Protokolü)
UMTS	: Universal Mobile Telecommunications System (Evrensel Mobil Telekomünikasyon Sistemi)

SEMBOL LİSTESİ

$\boldsymbol{\alpha}$	: MN tarafından üretilen toplam trafikteki sinyal mesajlarının oranı
$\boldsymbol{\beta}$	: MN tarafından üretilen toplam trafikteki veri paketlerinin oranı
$\boldsymbol{m_{sig}}$	: Kayıt güncelleme için kullanılan sinyal mesajlarının ortalama boyutu
$\boldsymbol{m_{data}}$	: Kayıt güncelleme için kullanılan veri paketlerinin ortalama boyutu
$\boldsymbol{1/\mu}$	: MN'nin bir alt ağdaki ortalama kalış zamanı
$\boldsymbol{\lambda}$	: Downlink paket iletim zamanı (paket/s)
$\boldsymbol{\Pi_i}$	: MN'nin AP_i alt ağında bulunma olasılığı
$\boldsymbol{N_l}$	: Tüm ağdaki toplam yönlü bağlantı sayısı
$\boldsymbol{D_{max}}$	: Bir VoIP uygulaması için bir etki alanından diğerine geçerken tolere edilebilir maksimum gecikme
$\boldsymbol{T_{start}}$	: MN'nin ilk kayıt paketini yolladığı zaman
$\boldsymbol{T_{end}}$	: MN'nin ilk veri paketini aldığı zaman

ÖZET

GELECEK NESİL KABLOSUZ AĞLAR İÇİN YENİ BİR MOBİLİTE YÖNETİM SİSTEMİNİN GELİŞTİRİLMESİ

Kablosuz ortamlarda IP ağlarının hızla gelişmesi ile mobil düğümlerin yönetimi daha fazla önem kazanmaktadır. Ağ ortamlarında heterojenlik arttıkça, farklı tiplerdeki kablosuz ağların IP katmanında entegrasyonu gerçekleşmektedir. Bu yüzden, ağ katmanı ve daha üst katmanların mobil düğümlerin hareketinden zarar görmemesi beklenmektedir. IP protokollerine dayanan mobilite yönetimi mekanizmaları halen büyük ölçekli servislerin işleyişinde yeterli miktarda etkili olamamaktadır.

Mobilite yönetiminin performansı ile ilgili en önemli noktalardan biri uygulama katmanının mobil düğümün hareket etmesi ve buna bağlı olarak IP adresinin değişiyor olmasıdır. Uygulama katmanında kurulan ve sürdürülen oturumlar mevcut IP adresi ve port numarası ikilisine dayanmaktadır. Bu durumu iyileştirme fikirlerindeki yeni bir akım ise oturum tanımlama ve kimlik tanımlama kavramlarının ayrılması üzerinedir. Bu iki kavrama göre günümüze kadar IP adreslerinin konum ve kimlik tanımlama olmak üzere iki rolü bulunmaktadır. Bu iki rolün ayrılması ile oturumlar IP adresi yerine, bir düğümü benzersiz şekilde tanımlayan yeni bir kimlik belirleyiciye göre tanımlanmaktadır. Böylece, IP adresi değişikliği ile uygulama oturumlarının zarar görmesi önlenmektedir. Bu yeni yaklaşım, TCP/IP protokol kümesine, IP katmanının üstüne, mevcut IP adresleri ve yeni kimlik tanımlayıcıların eşleştirmelerini saklamak üzere yeni bir katman tanımlamaktadır.

Bu kavramlara uygun olarak IETF ve IRTF kurumları tarafından ortaya çıkarılan Konuk Kimliği Protokolü (Host Identity Protocol-HIP) ortaya çıkmıştır. Bu protokol konum/kimlik tanımlayıcı ayrımını aynı zamanda güvenlik desteğini de bünyesinde barındırarak çözmeyi önermektedir. Bu tez çalışmasında HIP protokolü incelenmiş ve önerilen yeni yöntemler bu protokole dayanarak tasarlanmıştır.

Bu tez çalışmasının ilk bölümünde HIP protokolü ile kullanılmak üzere hiyerarşik ağ yapısı ve özelikle mikro mobilite konusunda HIP'in mevcut eksikliklerine çözüm getirmek adına yeni bir yer değiştirme yönetimi mekanizması tasarlanmıştır. Bu yeni yöntem, mobil bir düğümün hareketi esnasından yapması gereken konum güncellemesini erken zamanda başlatarak, yer değiştirme zamanını ve gecikmesini iyileştirmeyi amaçlamaktadır. Önerilen yeni yöntemin HIP'in klasik uçtan uca mobilite yönetimine göre getirdiği avantajlar incelenmiştir.

Çalışmanın ikinci bölümünde önerilen eHIP yöntemi için bir önceden tahmin eklentisi tasarlanmıştır. Buradaki amaç, mobil düğümün erken güncelleme prosedürünün, hareketi esnasında izlediği yolu inceleyerek, eHIP'den farklı olarak daha erken

tetiklenmesidir. Bu yöntemin normal eHIP mekanizmasına entegre edilmesi ve başarılı kararları incelemek suretiyle başarımı, mobil düğümün hızını göz önüne alarak ve almayarak incelenmiştir.

Tez çalışmasının üçüncü bölümünde ise eHIP'de önerilen ağ mimarinse uygun olarak, ağ yapısının bir örgü (mesh) ağ olarak düşünüldüğü bir sistem modellemesine yer verilmiş ve bu model doğrultusunda QoS (servis kalitesi-quality of service) faktörlerini göz önünde bulunduan bir mobilite algoritması önerilmiştir. Önerilen algoritma benzer hiyerarşik yol bulma algoritmaları ile karşılaştırılarak incelenmiştir.

Çalışmanın dördüncü bölümünde ise HIP protokolü, gerçek bir örnek ağ ortamında, mevcut infraHIP implementasyonu kullanılarak test edilmiştir. Çeşitli parametreler iki farklı ağ senaryosu üzerinden incelenmiş ve HIP'in gerçek ağ ortamındaki ağ davranışlarına dair sonuçlar elde edilmiştir.

Tez çalışmasının son bölümünde ise çalışmayla ilgili sonuç değerlendirmelerine yer verilmiştir.

SUMMARY

DESIGN OF A NEW MOBILITY MANAGEMENT SYSTEM FOR NEXT GENERATION WIRELESS NETWORKS

With the rapid growth of IP networks in wireless environments, management of mobile nodes has become a more important issue. As the heterogeneity increases in network environments, the integration of different types of wireless networks occurs in the IP layer. Therefore, it is expected the network layer and above layers to be aware of movement of mobile nodes. Mobility management based on IP protocols is not yet efficient enough to be used for large-scale service deployment.

One of the most important issues related to the performance of mobility management is related to the fact that the application layer suffers from the changing of IP addresses during the movement of the mobile node. In fact, the application layer established and ongoing sessions relies on the current IP address and the port number. New wave in the improvement ideas on this concept is separating the session identification and the location identification. More precisely, up to now the IP address was playing these two roles: the location, and the identification. So, by separating these two concepts, the sessions are not identified according to IP addresses but the new unique identifiers that define a node. This avoids the applications to suffer when the IP address changes during the mobility.. This new approach needs to introduce a new layer in the TCP/IP protocol stack, on top of the IP layer that will handle the new identifiers correspondent with the current IP address.

According to these concepts, Host Identity Protocol (HIP) is proposed by IETF and IRTF. This protocol proposes to solve the locator/identifier split problem by also including the security support. In this thesis study, HIP protocol is examined and new methods based on this protocol have been designed and proposed.

In the first part the study, a hierarchical network structure for HIP protocol and a new handover management mechanism have been designed in order to propose a solution for especially HIP's existing imperfections about mobility management. This new method aims to start the location updates of a mobile node earlier during mobility and so to enhance the handover time and latency. The advantages of this new method have been observed in accordance with HIP's end-to end mobility management.

In this study's second part, a prediction extension has been designed for eHIP method. This extension, aims to trigger the early update of a mobile node by investigating the

path during its mobility earlier than eHIP. The success of this method has been examined with integration of this extension to eHIP method and successful decisions made both with and without taking into account the mobile node's speed.

In the third part of this thesis, a system model and a related mobility algorithm considering QoS factor has been investigated where the network structure is taken into consideration as a mesh network and suitable for network architecture proposed for eHIP. The proposed algorithm has been compared with similar hierarchical path selection algorithms.

In the fourth part of the study, HIP protocol has been tested on a real network testbed and using infraHIP implementation. Various parameters on two different scenarios have been observed and results have been obtained about HIP's behaviors on real network environments.

1. GİRİŞ

Kablosuz haberleşmenin ve iletişimin hızlı yükselişi, mobil bilgi işlemede ve el cihazlarındaki büyük gelişmeler ve internetin çok büyük başarısı sayesinde devrim niteliğinde yaygınlaşan mobil haberleşme dönemi, mevcut mobil haberleşme sistemlerinin doğal bir başarısı olarak ortaya çıkmaktadır. Tüm bu gelişmelerle bağlantılı olarak kullanıcılar, farklı uygulamalara herhangi bir zamanda ve herhangi bir yerde erişmeyi, çok modlu mobil terminallerde mevcut olan birden çok arayüzden en iyisini kullanarak ve arka planda koşan teknolojilerden haberdar olmayı gerektirmeyecek şekilde talep etmektedirler.

Kablosuz ağ ve haberleşme teknolojilerindeki hızlı gelişmeler, kişisel ağlarda Bluetooth, yerel ağlarda IEEE 802.11, geniş alan ağlarında Universal Mobile Telecommunications System (UMTS) ve global ağlarda uydu ağ sistemleri gibi farklı kablosuz haberleşme sistemlerinin oluşmasına sebep olmuştur. Tüm bu ağlar birbirini tamamlamaktadır ve bundan dolayı bu ağların entegrasyonu birleştirilmiş gelecek nesil kablosuz ağları (next generation wireless networks-NGWN) meydana getirmektedir.

Kablosuz erişim ağlarının önlenemez gelişimi, mobil sistemlerdeki büyük gelişmeler ve internetin gelişimi üzerine, mobil kullanıcıların hareket ederken dahi istedikleri servislere her an ve haberleşme kalitelerinde azalma veya kesilme yaşamadan talep etmeleri durumu ortaya çıkmıştır. Yeni nesil IP tabanlı kablosuz ve mobil haberleşme sistemlerindeki en önemli konulardan biri de heterojen erişim teknolojileri arasında dolaşımı yönetmek için akıllı mobilite yönetimi sistemlerinin geliştirilmesidir. Mobilite yönetimi sistemleri hem bağlantı katmanı hem de ağ katmanı ile ilgili çalışmaları barındırmaktadır.

Gelecek nesil kablosuz ağlar, mobil kullanıcılara her zaman en iyi servisi sağlamayı amaçlamaktadır. Gelecek nesil kablosuz ağlarda kullanıcılar her zaman en uygun olan ağlara bağlı olmak durumundadır ve ihtiyaçlarına göre farklı ağlar arasında geçiş

yapmak durumundadır. Bununla birlikte iletişim ortamlarındaki gelişmelere paralel olarak iletilecek veri miktarlarının ve talep edilen servislerinde gelişmekte olduğunu gözlemekteyiz. Yeni nesil ağlarla birlikte kablosuz ortamda sunulacak servisler de gelişecek ve çeşitli servis sınıfları ortaya çıkacaktır. Dolayısıyla, gelecek nesil kablosuz ağlarda arzu edilen kalitede servis için gerekli olan kesintisiz mobilite desteğini sağlamak önemli bir konudur.

TCP/IP, statik bilgisayar ağları için tasarlanmış, mobilite desteği konusunda problemleri olan beş katmanlı bir protokol kümesidir. Ağ katmanında hizmet veren IP protokolüne göre bir IP adresinin iki ana fonksiyonu bulunmaktadır: ilgili düğümü ağ içinde benzersiz şekilde tanımlamak ve iki uç nokta arasındaki trafiği yönlendirmek. IP adresi bir düğümün o anda ait olduğu ağa özeldir. Bir düğüm bulunduğu alt ağı hareket etmesi nedeniyle değiştirdiğinde ise problem ortaya çıkmaktadır. IP adresi değişikliği, sürmekte olan bir IP oturumunun kesintiye uğramasına sebep olmaktadır. Bir düğüm otomatik olarak IP adresini yenilese, bir öndeki oturumda kurulan TCP bağlantıları, IP adresinin değişikliği nedeniyle bozulmaktadır.

İnternet yapısında yıllar boyunca iki adet isim alanı kullanımı (DNS ve IP adresleri), kullanıcıların ihtiyaçlarını gidermek için yeterli olmuştur. İnternet kullanıcılarının sayısı arttıkça, ihtiyaçlar da aynı oranda artmaktadır. Daha fazla gezginlik veya güvenlik gibi ihtiyaçlar ortaya çıkmıştır. Ne Mobile IP (gezginlik için) ne de IPSec (güvenlik için) tüm problemlere çözüm olmuştur, sadece bir yönden çözümler getirmiştir. Ayrıca, tüm çözümler TCP/IP protokolüne dayanmaktadır ve hiçbiri tamamıyla yeni bir çözüm sunmamaktadır. Günümüz internet mimarisinde, IP adreslerinin ikili rolü gezginlik yönetimi ve desteklenen host sayısı gibi konularda birçok problem yaratmaktadır. Bu problemleri çözmek için, Internet Engineering Task Force (IETF) ve Internet Research Task Force (IRTF) tarafından "Host Identity Protocol (HIP)" adında yeni bir protokol önerilmiştir (Nikander ve Moskowitz, 2006). En özet ifadeyle HIP protokolünde, IP adreslerinin konum tanımlama ve kimlik (oturum) tanımlama prosedürlerinin ayrılması önerilmektedir.

2. GENEL KISIMLAR

2.1. MOBİLİTE YÖNETİMİ

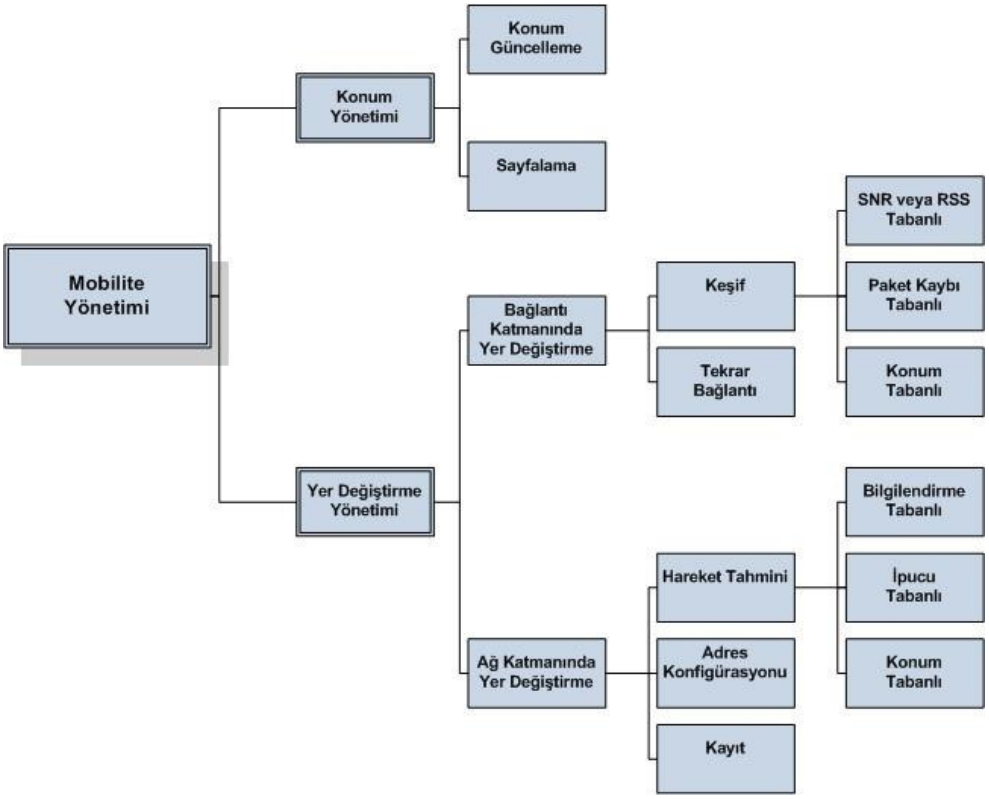
Mobilite yönetimi kavramı temelinde telekomünikasyon ağlarında, hareket halindeki mobil düğümün (Mobile Node-MN) yerini belirlemek ve bağlantı noktası değişen MN ile iletişimi sürdürmek gibi işlevleri yapabilmesini sağlar.

Bağlantı katmanında, internete erişimin kablosuz ağlar üzerinden olması erişim noktalarının sıklıkla değişmesi ihtiyacını beraberinde getirmektedir. Bu değişim kablosuz ağların küçük hücre boyutundan veya kullanıcıların her zaman en iyi servise erişmek için uygun olan kablosuz ağ üzerinden servis alma talepleri yüzünden olabilmektedir. Ancak, sık yer değiştirme sadece zaman gecikmelerini ve paket kaybını değil aynı zamanda güç yönünden hassas mobil terminaller açısından aşırı miktarda güç tüketimini de beraberinde getirmektedir.

Ağ katmanında mobilite yönetimi, temelinde kablolu ve sabit ağlar için geliştirilmiş olan TCP/IP protokolleri tarafından uygun şekilde çözümlenememektedir. IP ağlarında mobilite desteği için en yaygın olarak bilinen mekanizma Mobile IP (MIP)'dir (Perkins, 2002). MIP, IETF tarafından geliştirilen ve mobil düğümlerin bir ağdan diğerine IP adreslerini koruyarak geçişlerini sağlayan bir iletişim protokolüdür.

Bir ağ, minimum servis kesilmesi zamanı, servis kalitesinde (QoS) düşüş olmaması ve en etkili kaynak kullanma durumuyla bir zaman içinde birçok kullanıcıyı yönetme ihtiyacı kısıtları altında, kullanıcı hareketliliğini yönetebilmelidir. Hareket eden kullanıcıların yerini belirleme ve bağlantılarını sürdürme mobilite yönetiminin iki konusu altında incelenmektedir: Konum (Location) ve Yer Değiştirme (Handover/Handoff-HO) yönetimi. Konum yönetimi hareket halindeki mobil kullanıcının haberleşme süresince hareketlerinin kaydolmasını ve güncellenmesini sağlamaktadır. Yer Değiştirme yönetimi ise bir mobil kullanıcının bir kablosuz erişim

noktasından diğerine geçerken bağlantısını aktif durumda tutmasını sağlamaktır. Gelecek nesil sistemlerde, mobil elemanlar için iki tip dolaşım (roaming) mümkündür: etki alanı içinde (intradomain) ve etki alanları arası (interdomain). Etki alanı içinde yapılan dolaşım adından da anlaşılacağı gibi aynı sisteme ait farklı hücreler arasında hareket edilmesi ile gerçekleşir. Etki alanı dışında dolaşım ise farklı protokollere, teknolojilere veya servis sağlayıcılarına geçiş yapmak anlamını taşımaktadır. Şekil 2.1 mobilite yönetimi mekanizmalarının genel yapısını göstermektedir.



Şekil 2.1: Mobilite Yönetimi Mekanizmalarının Sınıflandırılması

Bağlantı katmanında yer değiştirme veya Katman 2 (Layer 2-L2) yer değiştirme, MN'nin yeni bir erişim noktasına (access point-AP) bir bağlantı oluşturması gerektiği için meydana gelmektedir. Bu bağlantı, kullanıcının hareketliliği, MN'nin o anki AP'sinden Alınan Sinyal Gücünün (received signal strength-RSS) veya Sinyal/Gürültü Oranının (Signal to Noise Ratio-SNR) düşmesiyle haberleşmenin aksaması gibi nedenlerle oluşabilmektedir.

Bir MN aynı alt ağ içindeki iki AP arasında hareket ederse, hiçbir IP tabanlı yönlendirme işlemi gerçekleşmez ve oturumu sekteye uğramaz. Ancak, eğer AP'ler farklı IP alt ağlarına bağlıysa, yönlendirme prosedürleri değişikliğe uğrayacağı için L2 yer değiştirmeyi ağ katmanı yer değiştirme (Layer 3-L3) izler.

2.1.1. Konum Yönetimi (Location Management)

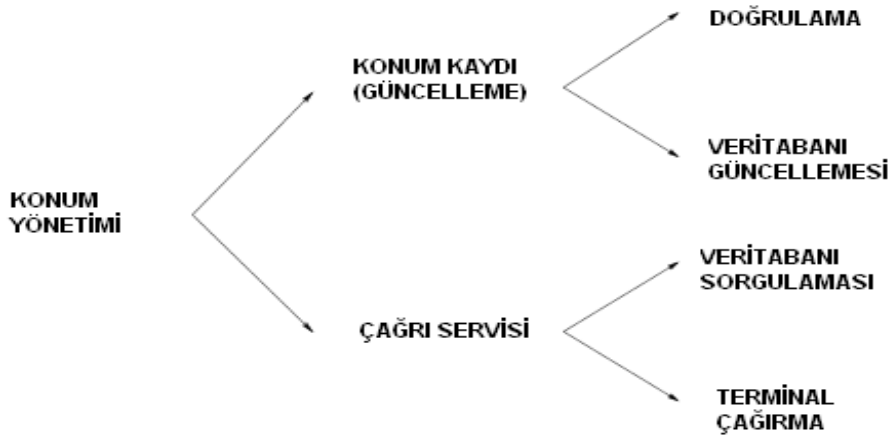
Konum yönetimi, hareket halindeki mobil kullanıcının haberleşme süresince hareketlerinin kaydolmasını ve güncellenmesini sağlamaktadır. İki ana kısımdan meydana gelmektedir:

- Konum kaydı(Konum Güncelleme): MN sistemi o an bulunduğu güncel konumu hakkında bilgilendirir ve konum veritabanlarının güncellenmesi, kullanıcının konum profilini gözden geçirilip düzeltilmesi sağlanır. İkinci aşama çağrı servisi amacı için terminal hakkında konum bilgilerini elde etmek için ağ sorgulanır. Bu aşama veritabanı sorgulamaları içerir.
- Çağrı Yönetimi: Bir MN ile haberleşme sürecinin başlaması için sistemin veritabanlarındaki bilgiler ile mobil elemanın o an bulunduğu konumunun belirlenmesi sürecidir.

Çağrı yönetimi iki ana adımdan meydana gelmektedir:

1. Aranan MN'ye hizmet veren veritabanının belirlenmesi.
2. Mobil elemanın bulunduğu hücre veya alt ağın konumunun belirlenmesi. Bu aşamaya aynı zamanda *paging (çağırma-sayfalama)* adı verilmektedir.

Konum yönetimi ile ilgili operasyonları özet halinde Şekil 2.2'deki gibi bir şema ile gösterebiliriz.



Şekil 2.2: Konum Yönetimi Operasyonları

Etki alanları arası dolaşımda; konum yönetim mekanizmalarının tasarımında aşağıdaki konuların göz önünde bulundurulması gerekmektedir.

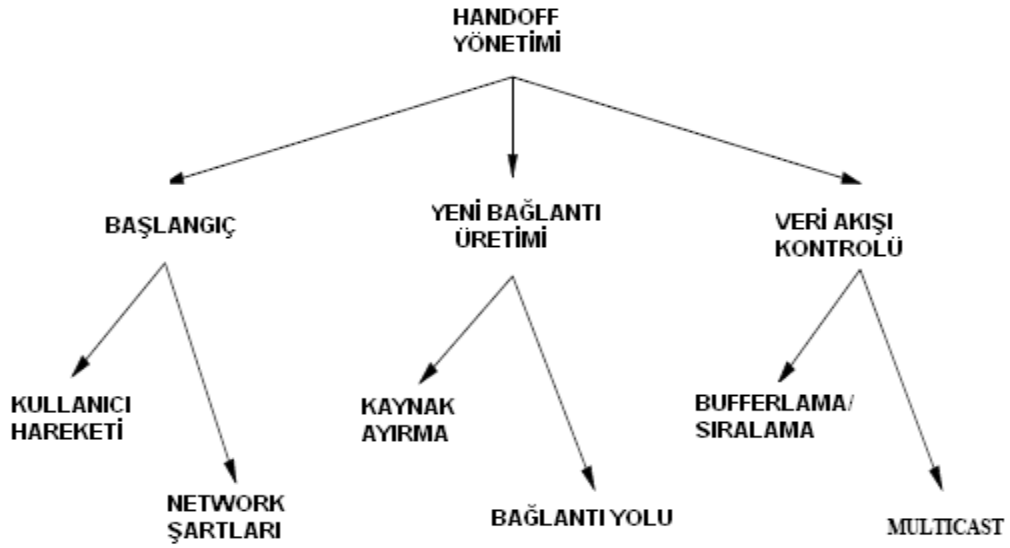
1. Sinyal yükünün azaltılması ve servisin sağlanmasındaki gecikmenin azaltılması
2. Farklı sistemlerdeki farklı QoS garantileri
3. Heterojen kablosuz ağların servis alanları birbiri ile tam olarak örtüşüyorsa;
    a. MN'nin ilgili ağlarda konum güncellemesi gerçekleştirmesi,
    b. Güncel kullanıcı bilgilerinin ilgili ağlarda ve istenilen şekilde saklanması,
    c. Belirli bir zaman kısıtlaması altında MN'in tam olarak yerinin belirlenmesi

2.1.2. Yer Değiştirme Yönetimi (Handoff/Handover Management)

HO yönetimi genel olarak bir mobil kullanıcının bir kablosuz erişim noktasından diğerine geçerken bağlantısını aktif durumda tutmasını sağlamak olarak tanımlanabilir. Kullanıcının hareketi haberleşme kalitesini değiştirebileceği için, kanal ve hücre değişimi söz konusu olur. Bu süreç, kullanıcı pozisyonunu değiştirirken veya ağ şartları zamanla değişirken mobil bağlantıyı sürdürmek için trafiği, yeni inşa edilmiş yollar üzerinden mobil terminale göndermek ile sorumludur.

HO yönetiminde, devam eden aramalar üç şart altında değiştirilmektedir:

- Sinyal şiddetinin kötüleşmesi
- Kullanıcının hareketliliği.
- Servis kalitesi düşüklüğü



Şekil 2.3: HO Yönetimi Operasyonları

HO yönetimi Şekil 2.3'de gösterildiği gibi 3 ana grupta incelenebilir. Radyo kanallarının kötüleşmesi (bir anlamda sinyal gücünün belirli bir eşik değerinin altına düşmesi), çağrıların aynı hücrenin içinde uygun şiddetteki yeni radyo kanallarına transferinin olduğu etki alanı içinde (intradomain-yatay) HO ile veya MN'nin bağlantılarının çok yakın bir hücreye transferinin olduğu etki alanları arası (interdomain-dikey) HO ile sonuçlanır. Her iki durumda da MN'nin bağlantıları eski baz istasyonu ile bağlantısı kesilmeden yeni baz istasyonuna geçebilir. Buna yumuşak (soft) HO adı verilmektedir. Diğer yandan eğer bağlantılar eski baz istasyonunda kesiliyor ve yeni baz istasyonunda yeniden kuruluyorsa buna sert (hard) HO adı verilmektedir. (Akyıldız ve diğ., 1999)

HO protokolleri; yönlendirme, kaynak yönetimi ve veri dağıtım sistemine dayanır. HO'yu gerçekleştirmek için mobil istasyon sürekli sinyal gücünü ve komşu

hücrelerdeki sinyal gücünü kontrol eder. Mobil istasyon tarafından kontrol edilecek hücrelerin listesi baz istasyonu tarafından mobil istasyona iletilir. Ölçümler hangi hücrenin daha kaliteli bir iletişim sağlayabileceği sonucunu verir. İki temel algoritma kullanılmaktadır:

- *Minimum kabul performansı algoritması:* İletişim kalitesi düştüğü zaman güç seviyesi artırılır. Bu artım; artımın kaliteyi etkileyemeyecek olmasına kadar devam eder. Bu aşamada HO gerçekleşir.
- *Güç bütçesi algoritması:* Bu algoritmada güç sürekli yükseltilmek yerine direk olarak HO gerçekleştirilir.

Heterojen ağlarda meydana gelebilecek dikey HO için olası senaryolar şu şekilde olabilir:

1. Kullanıcının hareket ederken o an bulunduğu ağdan çıkıp, o ağ ile kesişen bir ağa kısa bir süre ile geçiş yapması
2. Kullanıcının belirli bir ağa bağlı olması ancak olası ihtiyaçlarını göz önünde bulundurarak kendi ağıyla örtüşen bir ağa veya alt ağlara geçiş yapmayı istemesi
3. Ağın tüm yükünün farklı sistemlere dağıtılması esnasında ihtiyaç duyulması (bu durum her bir ağ için performansın optimize edilmesi esnasında kullanılabilir)

Gelecek nesil ağlarda kullanılacak HO yönetimi tekniklerinin geliştirilmesinde göz önünde bulundurulması gereken unsurlar şunlardır:

1. Sinyal ve tüketilen güç yükünün azaltılması
2. HO süresince QoS garantilerinin sağlanması;
    a. Oldukça düşük yatay ve dikey HO gecikmesi. Bu; gecikme sinyalleme mesajlarının işleme süresi, kaynakların ve yönlendiricilerin ayarlanmasındaki gecikme gibi süreleri barındırmaktadır.
    b. Kullanıcı trafiğinde az miktarda bozulma meydana getirmesi
    c. Sıfıra yakın HO başarısızlığı ve paket kayıp oranı olması
3. Ağ kaynaklarının etkili kullanılması
4. Gelişmiş ölçeklenebilirlik, güvenilirlik ve sağlamlık

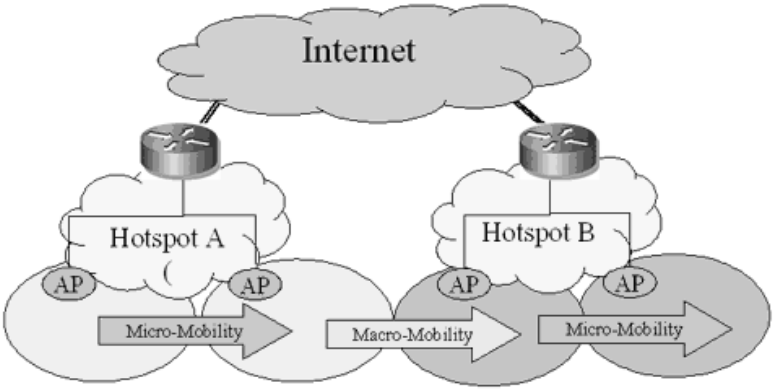
2.1.3. Mobilite Yönetimi Çözümleri

Gelecek nesil IP tabanlı heterojen ağlar ve homojen ağlar için günümüze kadar önerilen mobilite protokolleri TCP/IP protokol mimarisinin farklı katmanlarında çalışmaktadırlar.(Akyıldız ve diğ. 1999- Campbell ve diğ., 2004- Reinbold ve Bonaventure, 2004)

En yaygın olarak kullanılan mobilite teknikleri IP katmanında çözümler sunmaktadır. Daha alt katmanlarda yer alan kablosuz erişim teknolojilerine dayanmazlar ve bağlantıları bulunmamaktadır. Mobilite tekniklerinde kullanılan sinyalleme mesajları IP trafiği aracılığıyla taşınırlar.

Ağ katmanındaki çözümler Şekil 2.4'te görüldüğü gibi iki kategoriye ayrılabilirler: Makro ve Mikro Mobilite teknikleri.

- *Makro Mobilite:* Mobil kullanıcıların iki farklı etki alanı içinde hareket etmesine makro mobilite adı verilmektedir.
- *Mikro Mobilite:* Mobil kullanıcıların bulundukları etki alanı içerisindeki alt ağlar arasındaki hareketine mikro mobilite adı verilmektedir.



Şekil 2.4 :Mikro ve Makro Mobilite

2.1.3.1. Makro Mobilite Çözümleri

Internet haberleşmesinde bir düğüm IP adresi ile ifade edilmektedir ve bu adres de düğümün internette bulunduğu yeri benzersiz bir değer ile ifade etmektedir. Haberleşme esnasında bu düğüme yollanan paketler bu adrese dayanarak yönlendirilmektedir. Bu durumda bir düğümün haberleşmesine devam ettiği esnada hareket etmesini

engellemektedir. Mobile IP bu sorunu hareket halinde olan mobil düğüme ait eski adresine gelen paketleri bu düğümün yeni konumuna yönlendirmesi ile çözmektedir.

Mobile IP; üç adet alt bileşenden meydana gelmektedir. Bu alt bileşenlerin başında keşif (discovery) mekanizması gelmektedir. Bu mekanizma ile mobil kullanıcılar Internet'e bağlı olarak hareket ediyorken etraflarındaki yeni bağlantı noktalarının varlığını tespit edip kendilerine yeni IP adresleri alabilmektedirler. Diğer bir mekanizma ile mobil kullanıcı yeni aldığı IP adresini bağlı bulunduğu yeni yabancı görevli (foreign agent-FA) üzerinden, ev görevlisine (home agent-HA) kaydettirmektedir. Son mekanizma ile de, Mobile IP vasıtası ile bir mobil kullanıcının HA'sına gelen paketler mobil kullanıcının o anda gerçekten bağlı olduğu düğüme yönlendirilmektedir.

Keşif (Agent Discovery) : Bir mobil düğüm yeni bir alt ağa hareket edip etmediğini her bir FA'dan yayımlanan Görevli Bildirimi (Agent Advertisement-AA) mesajlarını dinleyerek tespit edebilmektedir. Ayrıca bir mobil düğüm de istediği zaman Görevli İstek Bildirimi (Agent Solicitation-AS) mesajları yollayarak olası mobilite görevlisi olup olmadığını öğrenebilmektedir.

Kayıt (Registration) : Bir mobil düğüm kayıtlı olduğu ağdan uzakta olduğunda, yeni bir adres (Care of Adress-CoA) alır. Bu adresi, FA mesajlarını dinleyerek veya talep ederek, Dinamik Host Konfigürasyon Protokolü (DHCP) veya Noktadan Noktaya Protokolü (PPP) ile bağlantı kurarak elde edebilmektedir. Mobil düğüm elde ettiği bu yeni adres ile mevcut kendi HA'sına başvuruda bulunur. HA, mobil düğümün kalıcı IP adresi ile yeni geçici adresini ilişkilendirir ve günceller.

Yönlendirme ve Tünelleme (Routing ve Tunnelling) : Bu mekanizma kapsamında kendi evinden uzakta bulunan bir mobil düğüme gelen paketler bu düğümün HA'sı tarafından kendisine yönlendirilir. Bu yönlendirme işleminde kullanılan kapsülleme ile tünelleme gerçekleştirilir. Mobil düğüme hizmet veren FA'ya ulaşan paketler burada dekapsüle edilir ve mobil düğüme yönlendirilir.

Bir mobil düğüm bir alt ağdan diğerine hareket ettiğinde HO süreci aşağıdaki adımları izleyerek uygulanmaya başlamaktadır:

- Yeni bir alt ağa bağlanan MN yeni bir CoA alır.
- MN, bu yeni adresiyle kendi HA'sına kayıt olur. HA eski CoA'nın en uç noktasına kadar kurulu olan tüneli kaldırır ve yeni CoA'ya doğru yeni bir tünel kurar.
- Yeni tünel kurulduktan sonra HA, MN'nin yeni adresini kullanarak paketleri tüneller.

Mobile IP'nin bir takım eksi yönleri de bulunmaktadır. Bir mobil düğümden iletişim kurulan düğüme (corresponding node-CN) yollanan paketler öncelikle HA tarafından yakalanmakta ve daha sonra MN'ye tünellenmektedir. Ancak, MN'den gelen paketler direkt olarak CN'ye yollanır. Bu üçgensel yönlendirme problemi, optimal yollardan daha uzun iletişim yollarına ve paket teslimi için ekstra bir gecikmeye neden olmaktadır. Bir MN bir alt ağdan diğerine hareket ettiğinde, yeni FA eski FA'yı MN'nin hareketi konusunda bilgilendiremez. Bu yüzden o anda eski CoA'ya tünellenmiş olan ve gelen paketler kaybolur.

Mobile IP, çok fazla hareket eden kullanıcılar için uygun bir çözüm değildir. Mobile IP'de, bir MN'nin alt ağlar arasındaki her bir hareketinde HA'ya konum güncelleme mesajı yollanması gerekmektedir. Bu konum güncelleme mesajı, MN hareket ettiği esnada bir iletişimde bulunmuyor olsa dahi yollanması gerekmektedir. Bu sinyal yükü, MN'lerin sayısı arttıkça belirgin bir şekilde artacaktır. Ayrıca, MN'nin kendi ağı ve ziyaret ettiği ağ arasındaki mesafe uzunsa, sinyal gecikmesi de oldukça fazla olacaktır.

Mobile IP mobiliteyi hem homojen hem de heterojen ağlarda desteklemektedir. Makro mobilite için oldukça uygun bir sistem olmasına karşın mikro mobilite için talepleri karşılayamamaktadır.

2.1.3.2. Mikro Mobilite Çözümleri

Mobil terminaller genellikle bir etki alanının alt ağları arasında hareket ederler. HA'ya doğru sinyal yükünü ve gecikmeyi azaltmak için birçok mikro mobilite protokolü geliştirilmiştir. Bu protokoller genel olarak iki kategoriye ayrılabilmektedir: tünel tabanlı ve yönlendirme tabanlı yöntemler:

- Tünelleme tabanlı yöntemler; mobilite ile ilgili sinyal mesajlarının kapsamını sınırlandırmak için yerel veya hiyerarşik kayıt ve kapsülleme yöntemlerini kullanarak toplam sinyal yükünü ve HO gecikmesini azaltmaktadır.
- Yönlendirme tabanlı yöntemlerde; paketleri iletmek için yönlendiricilerde ev sahibi düğümlere özel yollar belirlenir. Bu özel yollar, hareket eden ev sahibi düğümün hareketine bağlı olarak güncellenir.

2.2. KONUK KİMLİĞİ PROTOKOLÜ (HOST IDENTITY PROTOCOL)

Günümüz internet mimarisinde iki adet isim alanı kullanılmaktadır: Domain Name Service (DNS) ve IP adresleri. Bu iki isim alanı, internete dayanan teknolojilerin geliştirilmesinden yıllardır önemli rol oynamaktadırlar. IP adreslerinin hostlar için iki ana görevi bulunmaktadır. Bunlardan biri konumlayıcı (locator) olması diğeri de kimlik tanımlayıcı (identifier) olmasıdır.

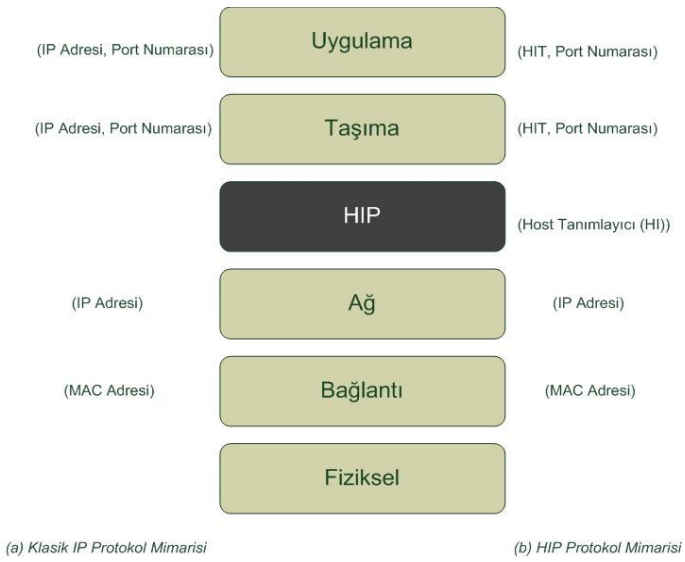
Ağ katmanı açısından bakıldığında bu adresler, konukların (host) ağdaki topolojik konumlarını tanımlamak ve yönlendirme işlemleri için kullanılmaktadırlar. Bir host hareket ettiğinde IP adresi de değişmektedir ve hostun konumu artık bu yeni adres ile belirlenmektedir.

Taşıma ve diğer üst katmanlar açısından düşünüldüğünde ise IP adreslerinin, hostları haberleşme ve bağlantıları boyunca tanımlamak üzere ikini bir rolü daha bulunmaktadır. Bu, IP adreslerinin kimlik tanımlayıcı rolüdür. Bu açıdan bakıldığında ise, IP adresinin haberleşme boyunca değişikliğe uğraması, hostun konumu değişse bile istenmeyen bir durumdur.

Konuk Kimlik Protokolü (HIP), IETF (IETF,2011) ve IRTF (IETF, 2011) tarafından günümüz internet dünyasında karşılaşılan zorlukları çözebilmek için geliştirilen yeni bir protokoldür. Mobile IP'ye alternatif olarak, HIP doğasında güvenlik, gezginlik gibi konular için mimarisinde çözümler barındırır. TCP/IP mimarisinde ağ ve iletim katmanları arasında yer almakadır. HIP'in varlığı ile taşıma katmanı protokolleri IP adresleri yerine kriptografik host tanımlayıcıları kullanabilmektedir. Özellikle son yıllarda artan İnternet kullanımı ve ortaya çıkan yeni ihtiyaçlar, temel TCP/IP

teknolojisinin yeterli olmakta zorlandığı bir noktaya varmıştır. Kullanıcıların farklı ağlar arasında hareket edebileceği ve aynı anda farklı ağlarla bağlantı kurabileceği gerçeği zamanla ortaya çıkmıştır. Ayrıca, internetin kullanım alanı büyüdükçe birçok güvenlik problemi ortaya çıkmıştır. Güvenlik eksikliği de aynı zamanda mevcut IP gezginlik sistemlerinin gelişimini engellemiştir. HIP, bu problemlere çözüm üretmek amacıyla TCP/IP mimarisine tam olarak uyum sağlayabilecek bir protokol olarak geliştirilmiştir (Gurtov, 2008).

IP adreslerinin ikili rolünün getirdiği problemlere çözüm olarak HIP'in önerdiği çözüm Host Identity (HI) kavramıdır. HI kısaca bir açık/gizli anahtar çiftinin açık anahtar kısmıdır. Bu anahtar genel olarak HI'nın 128 bit uzunluğunda bir versiyonu olan Host Identity Tag (HIT) olarak gösterilebilmektedir ve tüm internet üzerinde benzersiz olması gerekmektedir. HI'nin diğer bir gösterim yolu da yerel amaçlarla kullanılabilecek 32 bitlik Local Scope Identity (LSI)'dir. Şekil 2.5'te HIP'in TCP/IP mimarisinin hangi noktasında yer aldığı ve kullanılan değerler bakımından klasik mimariye göre farklılıkları gösterilmektedir (Gurtov, 2008).



Şekil 2.5: HIP Protokol Mimarisi

2.2.1. Internet İsim Alanları ve Konuk Tanımlama Metotları

İsim alanları bir konuğun veya bir servisin internet ortamında benzersiz bir şekilde tanımlanmasına imkan sunarlar. Günümüz internetinde hostlar için iki adet isim alanı kullanılmaktadır: IP adresleri ve Domain Name Server (DNS) alan adları. DNS alan

adları kullanımı ve okunması kolay host isimleri sunmaktadır. Ancak bir hostun var olan IP adresini DNS üzerinde güncellemesi gezginlik anında çok yavaş bir işlem olacaktır. Ayrıca birçok hostun DNS üzerinde değişiklik yapma hakkı olmayabilir. Ayrıca DNS güvenli olmayan ve kolayca bilgileri ele geçirebilir bir servistir.

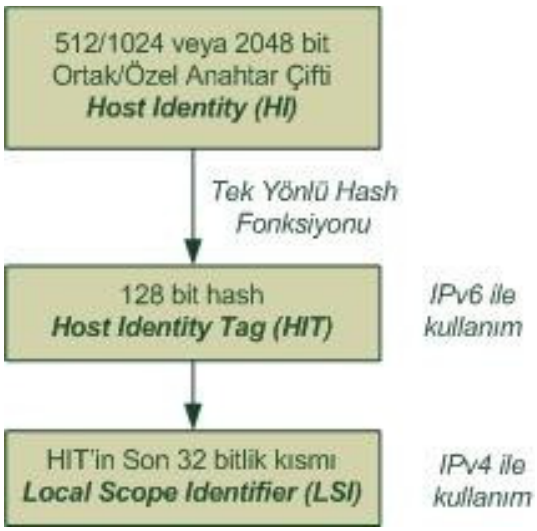
Varolan bu iki isim alanının üç ana dezavantajı bulunmaktadır:

- Taşıma katmanındaki bağlantıları kesmeden host adresini değiştirmenin mümkün olmamas,.
- Hostun doğrulanmasının söz konusu olmaması,
- IP adresleri dinlenerek (spoofing) ele geçirebilmesi ve gizlilik sağlayan (privacy preserving) iletişimin yapılamamasıdır.

HIP, host tanımlayıcılardan (HI) oluşan yeni bir isim alanı (namespace) kavramı getirmektedir. HI'lar üst katmanlarda tanımlayıcı rolünü üstlenmektedir. Açık anahtarın uzunluğu 512, 1024 veya 2048 bit olabilir ve genellikle RSA algoritması tarafından üretilmektedir. Yeni anahtarların üretimi zaman alıcı bir işlem olduğundan dolayı sadece eski anahtarların gizliliği ihlal edildiği zaman üretilirler. Host tanımlayıcı olarak veri paketlerinde ve üst katmanlarda kullanılan açık anahtar büyük ve değişken boyutludur ve uzunluğu da kullanılan kriptografik algoritmanın tipine bağlı olarak değişebilmektedir. Bu durumun üst katmanlarda yol açabileceği problemlere engel olmak adına HIP'te iki adet sabit uzunluklu tanımlayıcı tipi tanımlanmıştır:

- Host Identity Tag (HIT) : HI'nın 128 bitlik gösterimidir. HI üzerinde bir kriptografik hash fonksiyonu ile elde edilir. Hash fonksiyon kullanmanın iki adet avantajı bulunmaktadır. Bunlardan ilki sabit uzunluklu olması ve üst katmanlarda kullanımının daha kolay olmasıdır. İkincisi ise protokolde tutarlı bir formatta temsil ediliyor olmasıdır. HIT'ler HIP paketlerinin gönderici ve alıcılarını temsil ederler.
- Local Scope Identifier (LSI) : LSI ise HI'nın genellikle 32 bitlik bir gösterimidir. Var olan API'ler ve protokoller tarafından kullanılabilir. HIT'den daha kısa olması bir avantajdır ancak sadece yerel kullanıma uygundur. Genellikle halen sadece IPv4 destekleyen uygulamalar tarafından kullanılabilir.

Şekil 2.6'da HI, HIT ve LSI parametrelerinin elde edilmesi blok şema şeklinde gösterilmektedir. HIT, IPv6 adresleri ile aynı uzunlukta olduğu için üst katman uygulamalarında rahatlıkla IP adreslerinin yerine kullanılabilmektedir. HIT'lerin sabit uzunluklu olması, açık/gizli anahtar çiftlerinin üretilmesi için kullanılan kriptografik algoritmayı protokolden bağımsız duruma getirmektedir. LSI ise, HIT'in son 32 biti ile oluşturulabilen bir tanımlayıcıdır. Daha kısa uzunluklu oldukları için, farklı iki ortak anahtardan aynı iki tanımlayıcının oluşma olasılığı HIT'e göre daha fazladır. Bu yüzden LSI'ların kullanımı yereldir ve benzersiz olarak tanımlanamazlar.



Şekil 2.6: HI, HIT ve LSI'nın elde edilmesi

2.2.2. HIP Paket ve Mesajları

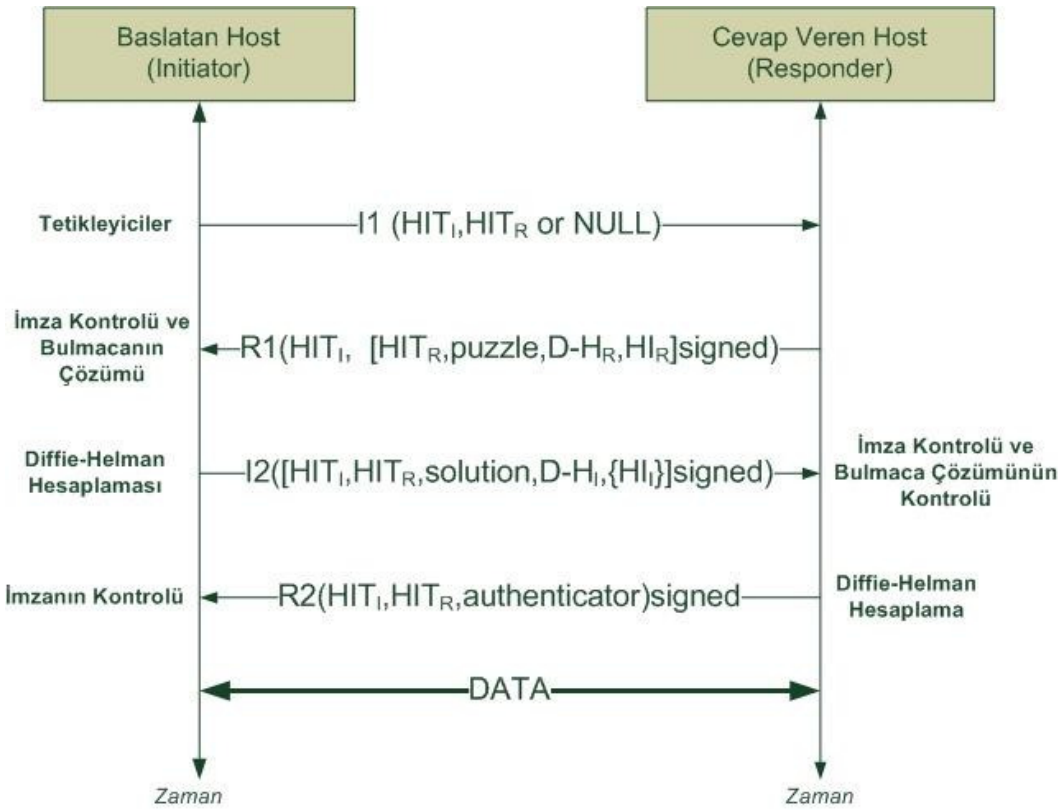
HIP protokolünde kullanılan sekiz adet temel paket bulunmaktadır. Bu paketlerin dört tanesi HIP Base Exchange için, bir tanesi güncelleme için, bir tanesi bilgilendirmelerin yollanması için ve iki tanesi de HIP bağlantılarının kapatılması için kullanılmaktadır. Bu mesajlar Tablo 2.1'de gösterilmektedir. İlk dört paketin detayları ve görevleri bölüm 2.2.3.'de ayrıca belirtilmiştir. UPDATE paketleri HIP'in mobilite yönetimi için kullanılan konum güncelleme paketidir.

Tablo 2.1: HIP Paketleri

I1	HIP Başlatan Paketi
R1	HIP Cevap Veren Paketi
I2	İkinci HIP Başlatan Paketi
R2	İkinci HIP Cevap Veren Paketi
UPDATE	Güncelleme Paketi
NOTIFY	Bilgilendirme Paketi
CLOSE	HIP Bağlantısı Sonlandırma Paketi
CLOSE_ACK	HIP Sonlandırma Onaylama Paketi

2.2.3. Temel Bağlantı Kurulumu (Base Exchange - BE)

HIP Base Exchange (BE), iki host arasında bir HIP bağlantısı kurulmadan önce gerçekleştirilen bir kriptografik anahtar değişim prosedürüdür, bir başka deyişle dört yönlü bir el sıkışmadır (Moskowitz ve diğ., 2008). Bu el sıkışmayı başlatan host Başlatan (Initiator-I), diğer host ise Cevap Veren (Responder-R) olarak tanımlanmaktadır. BE, dört paketten (I1,R1,I2,R2) meydana gelmektedir ve klasik bir Diffie-Hellman anahtar değişimi gerçekleştirir. Şekil 2.7'de dört yönlü BE prosedürü gösterilmektedir.



Şekil 2.7: HIP Base Exchange Prosedürü

I1 Paketi: BE, I1 paketinin bağlantıyı başlatan host tarafından gönderilmesiyle başlar. I1 mesajı bir HIP bağlantısı başlatmak için bir tetikleyici olarak da tanımlanabilir. I1 paketi BE prosedüründe şifrelenmeyen veya imzalanmayan tek pakettir. Kendisinin ve eğer biliniyorsa karşı taraftaki hostun HIT'ini içerir.

R1 Paketi: I1 paketini alan host, paketteki HIT değerinin kendi HIT değeri ile uyup uymadığını kontrol eder ve HIP bağlantısının kurulmasını kabul edip etmeyeceğine karar verir. Eğer kabul edecekse R1 paketini yollar ve bu paket, BE'nin devam etmesi için iletişimi başlatan hostun çözmesi gereken bir bulmaca içerir. Bulmacanın amacı, cevap verecek olan hostun DoS ataklarından korunmasını sağlamaktır. Ayrıca bu paket Diffie-Hellman anahtar değişiminin ilk kısmını da içerir. Bu paketin bulmaca kısmı hariç diğer kısımları imzalanmıştır.

I2 Paketi: BE'yi başlatan host tarafından alınan R1 paketinden sonra, bir önceki adımdaki bulmacanın çözümünü ve Diffie-Hellman anahtar değişiminin bir sonraki kısmını I2 paketi ile karşı hosta yollar. I2 paketinin tamamı imzalı halde gönderilmektedir. Cevap veren host, I2 paketinden başlatan hostun Diffie-Hellman ortak anahtarını elde eder ve Diffie-Hellman oturum anahtarını hesaplar ve böylece bir HIP bağlantısının kurulacağını onaylar.

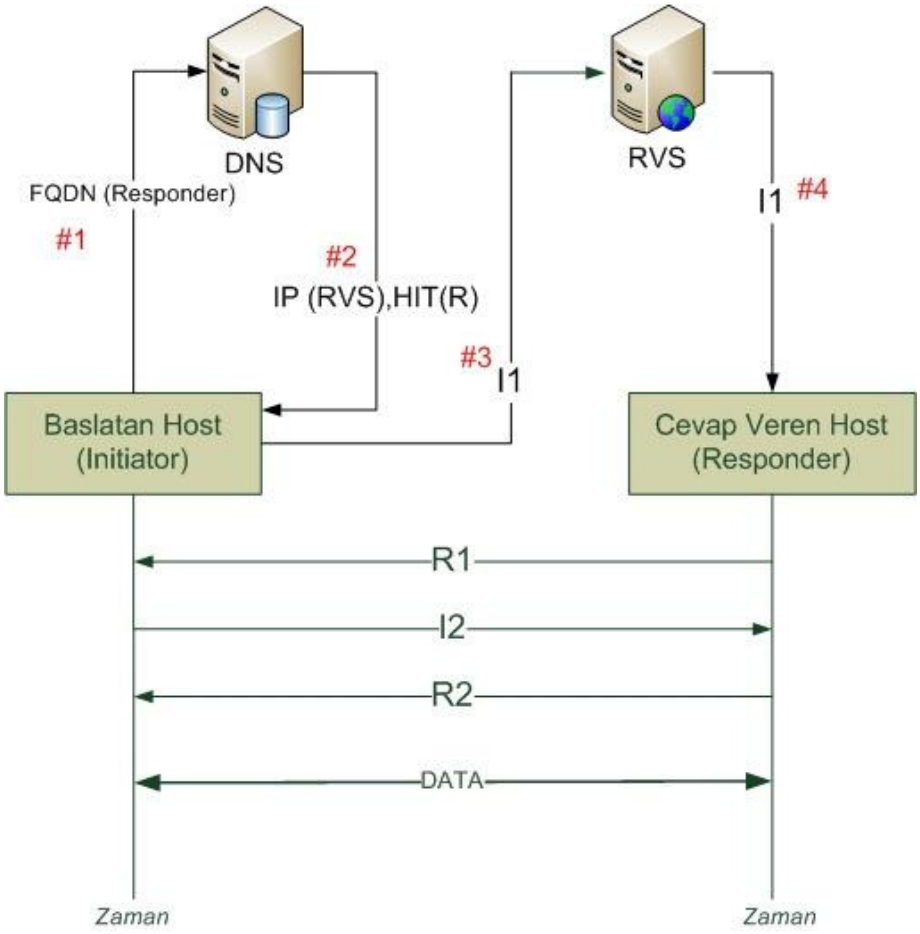
R2 Paketi: BE'yi sonlandıran pakettir. Tüm HIP paketine ait bir imza içerir. Bu pakete bağlantıyı başlatan hostu tekrar (replay) saldırılarından korumak için ihtiyaç duyulmaktadır.

2.2.4. Randevu Sunucusu (RVS)

Bir HIP hostunun erişilebilir olması için IP adresinin bir noktada saklanıyor olması gerekmektedir. Günümüzde genel olarak bu saklama işlemi için DNS sunucuları kullanılmaktadır. DNS sistemiyle ilgili problemler ise daha önceki bölümlerde belirtildiği gibi, gezginlik durumunda konum bilgisinin güncellenmesinde gecikmeler meydana gelmesi ve sistemin yavaş olmasıdır. Bu problemi çözmek amacıyla, HIP protokolü ile beraber yeni bir ağ elemanı tasarlanmıştır. Randevu Sunucusu (Rendezveous Server -RVS) adı verilen bu eleman HIP haberleşmesi ile ilgili her türlü detayı saklamaktadır (Laganier ve Eggert, 2008b). Sadece RVS'nin konum bilgisi DNS

sunucularında saklanmaktadır (Nikander, 2008b). Birbirleri ile HIP protokolü ile haberleşmek isteyen hostlar bağlı oldukları RVS'lere HIT'leri ve o anki IP adresleri ile kayıt olmak zorundadır. Şekil 2.8, RVS'nin varlığında BE prosedürünün nasıl gerçekleştiğini göstermektedir.

RVS tüm hostların ağ içinde erişebileceği ortak bir nokta durumundadır. Bir host (I) iletişim kurmak istediği diğer bir host (R) ile BE başlatmak istediğinde I1 paketini direkt olarak karşı hosta değil RVS'ye yollamaktadır. I, RVS'nin IP adresini DNS aracılığıyla öğrenir ve RVS'ye BE'yi başlatmak için I1 paketini yollar. Bu durumda RVS, gelen paketin HIT'inin kendi kayıtlarında kontrol eder ve eğer varsa bu paketi ilgili IP adresine yollar. Bu adımdan sonra BE'nin diğer mesajları RVS'ye gerek duyulmaksızın direkt olarak I ve R arasında gönderilir. R, R1 mesajı ile gelen isteği cevaplayabilmek için I'nın IP adresini bilmesi gerekmektedir. RVS sunucusu I'nın adresini sakladığı zaman I'nın kaynak adresini içeren bir FROM parametresi ekler. Ayrıca I1 mesajının içerisine RVS_HMAC parametresini de ekler. RVS_HMAC parametresinin HIP HMAC parametresinden tek farkı HIP başlığı içinde FROM parametresinden sonra farklı bir kod numarası bulundurmasıdır. R host, FROM parametresinden elde ettiği I'nın IP adresine direkt cevap verir. Ayrıca I1 paketinin geldiği RVS sunucusunun IP adresini içeren bir VIA parametresi ekler. VIA parametresi HIP işlemlerinin tanımlanmasını sağlar. Aynı zamanda VIA parametresi birkaç RVS sunucusunun adresini tutabilir. Tutulan bu IP adresleri sadece IPv6 veya IPv4-in-IPv6 formatında olabilir.



Şekil 2.8: RVS'nin varlığında HIP Base Exchange

2.2.5. Kayıt Mekanizması (Registration)

HIP'e ait kayıt mekanizmasında kullanılan bir takım kavramlar şunlardır:

- Talep Eden (Requester-REQR); servise kayıt olmak için kayıtlı HIP REGR'den kayıt talep eden HIP düğümüdür.
- Kayıt Eden (Registrar-REGR); diğer düğümlere bir veya birden fazla servise kayıt olmak için kayıt prosedürünü sağlayan HIP düğümüdür.
- Servis ise REQR düğüme sunulan, HIP protokolünde işlenen etkinlikler veya yeni imkanlara bir araç olarak tanımlanabilir.

REGR'nin ve kayıt olmasını sağladığı HIP servisi talep eden tarafın birlikte kullandıkları prosedüre Kayıt Mekenizması (Registration) adı verilmektedir (Laganier ve Koponen, 2008a). Her kaydın sınırlı bir ömrü vardır. REQR bağlanılmış kayıt servislerine tekrar kayıt yapmak suretiyle mevcut kayıt prosedürünü genişletebilirler.

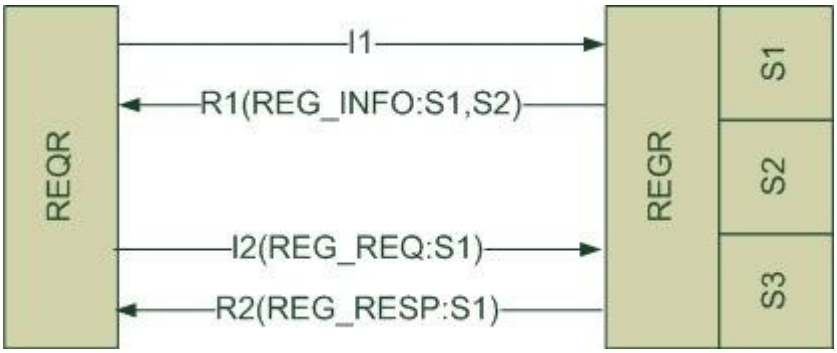
REQR yeni bir REGR keşfettikten sonra aralarında HIP BE başlatırlar ya da REGR'ye bağlı bir mevcut HIP bağlantısını kullanırlar. Her iki durumda da REGR kayıt prosedürünü kabul etmek ya da reddetmek için fazladan parametreleri kullanır. REQR ise sağlanan servise kayıt olmak için gerekli parametreleri kullanır. REQR ve REGR'nin her ikisi de mesajlarında kayıt tiplerini içeren özel HIP parametrelerini kullanabilir.

Bir sunucu REGR gibi davranmak istiyor ise kuracağı tüm BE bağlantıları boyunca göndereceği R1 paketleri içerisinde bir REG_INFO parametresi bulundurmalıdır. Geçici durumda servis sağlayamama durumunda ise R1 paketlerindeki REG_INFO parametresi boş olmalıdır. Tekrardan bu servis kullanılmak istendiği zaman uygun tüm servislerin belirtildiği yeni bir REG_INFO parametresini içeren UPDATE paketleri gönderilmelidir.

Mevcut bir servise kayıt talebi geldiği zaman, REQR bir I2 paketi içerisinde uygun bir REG_REQUEST parametresi oluşturur ya da REGR tarafına bir UPDATE paketi gönderir. Eğer REQR, REGR ile hiç HIP bağlantısı kurmamış ise I2 paketi içerisinde REG_REQUEST parametresini göndermelidir. Böylece REGR tarafına gönderilen paketlerin sayısı en aza indirgenecektir. REGR REG_REQUIRED parametresinin tipini NOTIFY olarak belirtilmiş R2 paketiyle REG_REQUEST parametresini taşımayan bir HIP bağlantısını sonlandırabilir. Bu durumda sunucular arasında hiçbir HIP bağlantısı oluşturulamayacaktır.

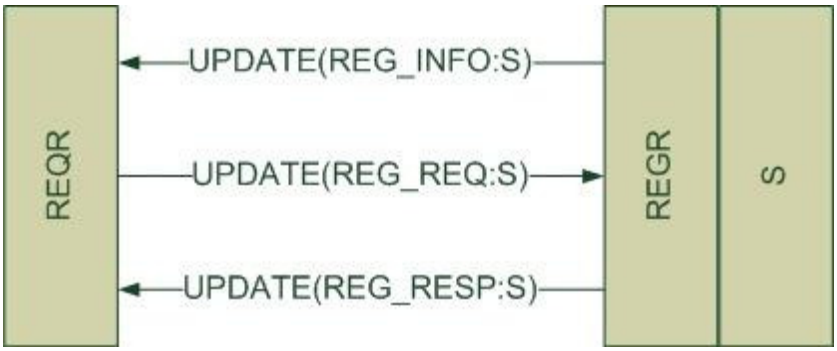
Kayıt prosedürü başladığında, REQR I2 paketinde yer alan sunucu kimliği ile REGR tarafından kontrol edilir. Eğer kimlik doğrulaması gerçekleşirse REQR istediği servislere ulaşabilecektir. Bu kimlik doğrulamasının detayları talep edilen servislere göre değişebilir. Kimlik doğrulaması gerçekleştikten sonra, REGR cevap paketinde bir REG_RESPONSE parametresi gönderilmelidir. Ayrıca bu cevap paketinde kayıt olunan servisin tipleri belirtilmeli ve kayıt esnasında oluşabilecek herhangi bir sorun yüzünden ortaya çıkan REG_FAILED parametresini içermemelidir. Bu cevap paketi mevcut bir BE bağlantısında talep edilen bir R2 mesajı ya da bir UPDATE mesajı olabilir. Özellikle, REG_FAILED gibi hata tipi fazla kimlik gerektiren servis tipleri için

kullanılır. Eğer REGR yeterince kimlik doğrulama prosedürünü gerçekleştirebilecek durumda ve REQR uygun durumda kimliğe sahip ise aralarında HIP bağlantısı kurulduktan sonra REQR tekrar kayıt olmayı deneyebilir.



Şekil 2.9: Arada HIP bağlantısı yok iken kurulan Kayıt Mekanizması

REG_RESPONSE parametresinin başarılı bir şekilde işlenmesiyle REQR tarafında kayıt mekanizması oluşturulur. Aynı zamanda bu işlemin anlamı REGR tarafında da başarılı bir şekilde kayıt işlemini başlatmak anlamına gelmektedir ve servisler kullanılmak üzere uygun duruma gelmiştir. REQR ve REGR taraflarından her ikisi de istedikleri zaman kayıt işlemi tamamlanmadan önce bağlantıyı iptal edebilirler.

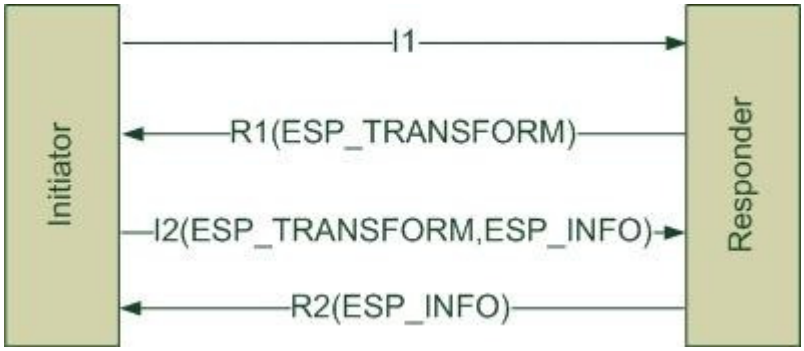


Şekil 2.10: Arada HIP bağlantısı varken kurulan Kayıt Mekanizması

2.2.6. ESP Güvenlik Bağlantısı Kurulumu

HIP kullanarak sunucular arasında ESP (Encaspsulating Security Payload) protokolü aracılığıyla karşılıklı güvenlik bağlantısı kurulumu yapmak BE gibi sunucular arasında mesajlaşma yoluyla sağlanmaktadır. ESP iletişimi süresince HIP sunucuları arasında

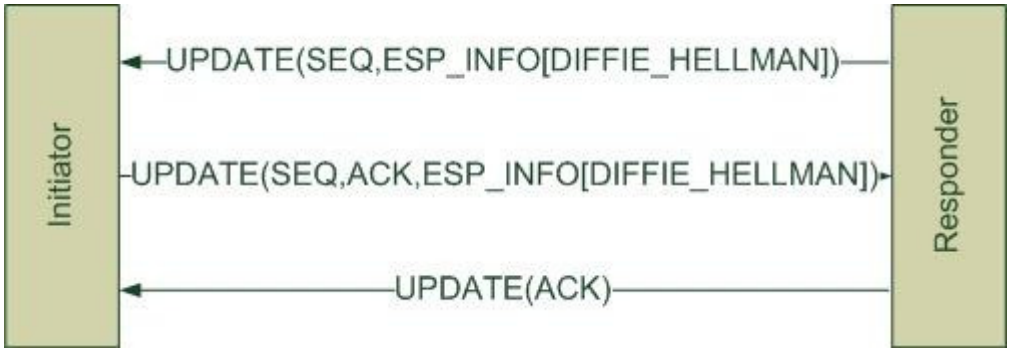
bilgi alışverişi gerekmektedir. Çift taraflı bu iletişim süresince ESP parametreleri bu mesajlardan R1, I1 ve R2 mesajları içerisinde iletilmektedir.



Şekil 2.11: ESP Kullanımı

R1 mesajı ESP_TRANSFORM parametresini içermektedir. Bu parametre karşı tarafa ESP kullanmak için talep edildiğini belirten bir tanımlayıcıdır. I2 mesajı, R1 mesajından alınan ESP_TRANSFORM parametresi için bir cevap döndürür. Gönderici R1 mesajı içerisinde ESP_TRANSFORM parametrelerinden birini seçmiş olmalıdır ve I2 mesajı içerisinde bu seçilen ESP_TRANSFORM parametresi bulunmalıdır. Ayrıca mesajı yanıtlayan sunucu, bağlı olduğu diğer sunucuların kullandığı SPI değerlerini içeren ESP_INFO parametresini de mesaj içerisinde göndermelidir. R2 mesajı ise ESP ayarlamasını sonlandırır. R2 mesajı iletişimi başlatan sunucu için ESP Security Association için gerekli SPI bilgilerini içerir.

ESP güncelleme işlemi ise iki mesaj yoluyla gerçekleştirilmektedir. HIP UPDATE mesajı yoluyla mevcut ESP Güvenlik İlişkisi (Security Association-SA) güncelleme parametrelerini günceller. UPDATE mekanizması ve mesajları Moskowitz ve diğ, (2008) ve Jokela ve Moskowitz (2008) tarafından çalışmalarında tanımlanmıştır.



Şekil 2.12: ESP Güncelleme Prosedürü

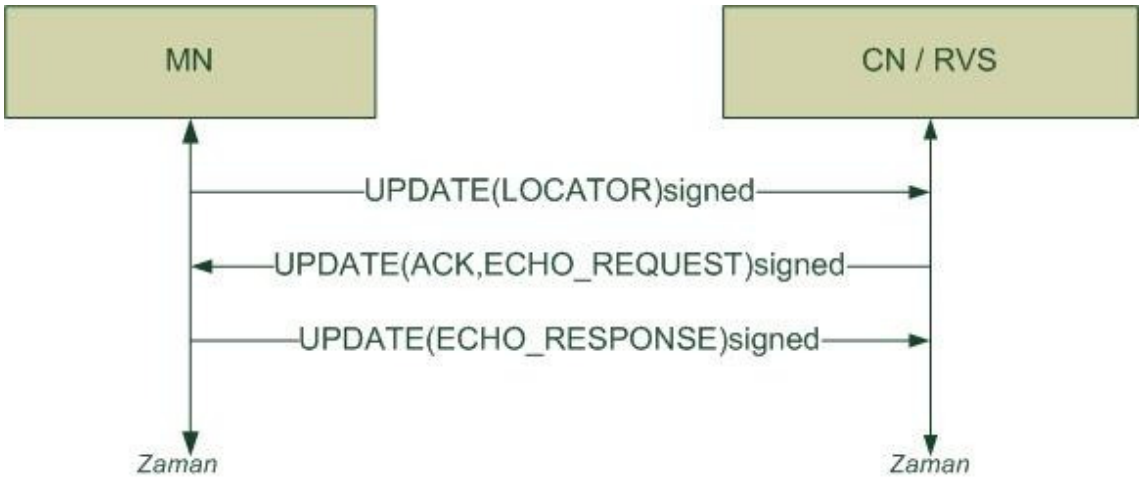
Sunucu ESP SA güncelleme isteği oluşturur ve bir UPDATE mesajı gönderir. Mesaj kullanılan eski SPI değerini, oluşturulan yeni SPI değerini ve bir sonraki adımın hangileri olduğunu içeren indeks değerini içerir. Eğer indeks değeri oluşturulmalı ise UPDATE mesajı Moskowitz ve diğ. (2008) çalışmasında tanımlanan DIFFIE_HELLMAN parametresini de içermelidir.

Karşı taraf gönderilen bu UPDATE mesajına karşı mevcut ESP SA ile bir UPDATE mesajı gönderecektir. Bu mesaj kullanılan eski SPI değerini, yeni SPI değerini ve anahtarlama indeksini içeren ESP_INFO parametresini içermektedir. Eğer gelen UPDATE mesajı DIFFIE_HELLMAN parametresini içeriyor ise cevaplanan pakette mutlaka bir DIFFIE_HELLMAN parametresi bulunmak zorundadır.

2.2.7. HIP'te Mobilite ve Çoklu Konumluluk

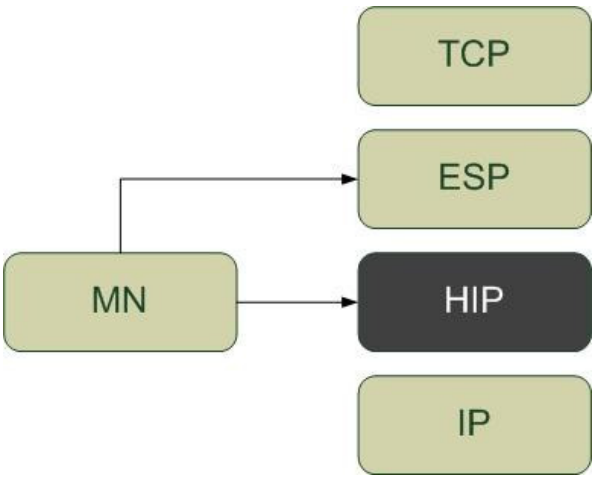
Bir host o an bağlı olduğu ağ içinde veya başka bir ağa hareket ettiğinde IP adresi değişmektedir ve bu adresi bağlı olduğu diğer düğümlere HIP UPDATE paketleri içinde LOCATOR parametresi ile haber vermektedir. Bu UPDATE paketi HIP protokol kurallarına göre gönderilir. ESP protokolünün varlığında ise karşılıklı hostlardan biri karşılıklı güvenlik bağlantısını yeniden kurmak ve hatta yeni bir Diffie-Hellman anahtarı üretmek isteyebilir. Tüm bu işlemler UPDATE paketinin içinde bulunan ek parametrelerle yollanan bilgiler dahilinde tetiklenir. Çoklu Konumluluk (multi homing) özelliğine sahip bir HIP hostu, farklı erişim ağlarına erişebilen farklı ara yüzler için birden çok IP adresine sahiptir. Ağdaki konumunu belirten birden çok adresi olduğu için de, haberleşmek istediği tüm diğer hostlara bu adresleri haber vermelidir. Bunu gerçekleştirmek için, UPDATE mesajları içinde LOCATOR bilgilerini bildirir ve hatta

tercih ettiği adresi konusunda da bilgilendirebilir. Birden çok adres içeren bir UPDATE mesajını alan bir host, olası yanlış güncellemeleri önlemek için bu adreslerinin her birinin erişilebilirliğini kontrol etmek durumundadır (Nikander ve diğ., 2008a)



Şekil 2.13: HIP'te Konum Değişikliği Durumunda Güncelleme Mekanizması

HIP, sunucuları tanımlamak için IP adresi yerine açık/gizli anahtar çiftleri kullanarak taşıma ve ağ katmanlarını birbirinden ayıran bir mimari yapıya sahiptir. Bir sunucu HIP kullandığı zaman, üzerinde yer alan katmanlar(örnek olarak taşıma katmanı soketleri, ESP güvenlik bağı) sunucuları bu kimlikler ile temsil ederler ve IP adreslerini sadece paket gönderimi esnasında kullanırlar. Yine de her sunucu eş sunuculara erişebileceği en az bir adet IP adresini bilmelidir. Bu adresler, BE sırasında kullanılan adreslerdir. Katmanlar arasında bu tarz bir ayrışma ağ katmanı taşınabilirliği ve sunucu çoklu konumluluğuna yönelik yeni çözümler ortaya çıkarmaktadır. Basit uçtan uca yeniden adreslemenin işlevselliğinin yetersiz olduğu birçok durum vardır. Bunlara taşınabilir bir sunucuya ulaşılabilirlik, konum gizliliği, sunucuların eş zamanlı taşınabilirliği ve NAT dolaşımına yönelik bazı durumlar örnek olarak verilebilir. Bu gibi durumlarda ağa işlevsellik açısından bir yardımcı gereklidir. İşte burada devreye HIP RVS girer. HIP bir anahtar belirleme ve parametre uzlaşım protokolüdür. Asıl uygulamaları sunucu kimliklerine dayanan sunucu mesajlarına kimlik doğrulama yapabilme ve ESP taşıma formatı için güvenli bağı oluşturmaya yöneliktir.



Şekil 2.14: ESP kullanan HIP protokolünün katmanlı mimarisi

Şekil 2.14'te ESP taşıma formatını kullanan HIP protokolünün katmanlı mimarisi görülmektedir. HIP'te üst katmanlar IP adresine değil HIT'lere bağlıdır. HIP ara katmanı HIT ve IP adresleri arasındaki bağı sağlamakla görevlidirler. SPI (ESP Security Parameter Index), gelen paketi doğru HIT ile ilişkilendirmek için kullanılmaktadır.

İlk durumda, yani protokolün temelinde olduğu gibi, taşınabilirlik ve çoklu konumluluk söz konusu olmadığında, BE sunucular arasında kullanılacak HIT'leri, ESP için kullanılacak SPI'ları ve IP adreslerini belirler. Gönderilecek her paket için bu tarzda sadece bir bağlantı yapılabilir ve HIP katmanında kullanılabilecek alanlar sadece ESP'de sunulan alanlardır (SPI, HIT). Gelen paketler için ise SPI doğru sunucu içeriğini belirlemede tek gerekli olan malzemedir. ESP yeniden anahtarlamaları HIT ve SPI çiftlerinin çevrim tablosunda değişikliğe neden olurken IP adreslerini etkilemez. Bir diğer durum da taşınabilirlik söz konusu olduğunda, bir sunucu tek bir konuma sahip olmasına rağmen IP adresini taşıyabilmektedir. Bu durumda iki nokta önemlidir: Birincisi, eş sunucu adres değişikliği hakkında HIP UPDATE mesajı ile haberdar edilmelidir. İkincisi ise her sunucu HIP ara katmanındaki yerel bağlarını değiştirmelidir (yeni IP adresi ile). Bu durumda bir UPDATE mesajı ile hem SPI'ler hem de IP adresleri eş zamanlı olarak değişmiş olur. Son olarak çoklu konumluluğa imkan tanıyan bir senaryoda, HIP katmanında birden fazla adrese sahip bir durumu göz önünde bulundurulursa ve bu durumda aynı ara yüzde hem IPv4 hem IPv6 adresleri ya da farklı ISP'lere bağlı birden fazla ara yüzü desteklenmektedir. Bu tarz sunucu çoklu konumlulukları, gelen paketlerin ESP tekrar penceresinden düşmesini önlemek için genellikle her bir ara yüz için ayrı bir ESP bağı gerektirir. Çoklu konumluluk bu sayede

Şekil 2.14'ün sağ tarafında gösterilen bağların birden çoğa şeklinde olmasına izin vermektedir (gönderim esnasında bir HIT birçok SPI ile ve dolayısıyla birçok IP adresiyle birlikte kullanılır). Yine de her bir paket için sadece bir SPI ve IP adres çifti kullanılabilir. Dolayısıyla yukarıda belirtilmiş olan MN bloğunun görevi dinamik olarak bunları değiştirmektir. Bu tarz çoklu bağların yerel olarak yönetilmesinin ötesinde, uçtan uca HIP protokolü istenen HIT, SPI ve IP adresleri arasındaki tabloları tanımlayabilecek ölçüde yeterince esnek olmalıdır.

2.2.7.1. Locator Parametresi

Konumlayıcı (Locator) parametresi, ağa bir bağlantı noktası tanımlar ve HIP ara katmanının altındaki katmanlarda paketlere nasıl davranılacağını belirleyen uçtan uca tünelleme ya da sunucu bazlı tekilleyici (per-host demultiplexing) bağlamını içerir. HIP'in alt katmanlardaki paketlere nasıl davranılacağını belirlemek için yalnızca IP adresleri yeterli değildir. Örneğin sunucu çoklu konumluluğu içeren bir durumda ESP anti tekrar penceresi ihlallerini önlemek için belirli IP adreslerinin belirli ESP SPI'lerine bağlanması gerekebilir. Belirli tünelleme senaryolarında adresler taşıma portları ile birleştirilebilirler. Bu durumda konumlayıcı alanları geleneksel ağ adresleri gibi çalışırlar.

2.2.7.2. Taşınabilirlik

Bir sunucu başka bir adrese taşındığında CN'leri yeni adresi hakkında LOCATOR parametresi içeren bir HIP UPDATE mesajı ile haberdar eder. Bu UPDATE paketi karşı host tarafından onaylanır. Güvenilirliği sağlamak amacıyla, bir paketin kaybolduğu durumda HIP protokol açıklamasında da belirtildiği gibi yeniden gönderilmesi söz konusudur. UPDATE paketindeki veri içeriğini pakette bulunan imza ve özet ile kimlik doğrulamasına tabi tutulabilir.

ESP taşıma formatı kullanımı esnasında sunucu, güvenlik bağını yeniden anahtarlamaya karar verebilir ve yeni bir Diffie-Hellman anahtarı oluşturabilir; tüm bu işlemler UPDATE paketinde ilave parametreler kullanarak tetiklenir. ESP kullanılırken bir sunucu, tüm adreslerden HIP tarafından oluşturulmuş ESP güvenlik bağı ile korunan paketler alabilir. Böylece bir sunucu IP adresini değiştirdikten sonra yeniden anahtarlamaya ihtiyaç duymadan paket göndermeye devam edebilir. Yine de CN'ler, verinin geldiği sunucuyla ilişkili adres kümesini güvenli bir şekilde güncellemeden yeni

adrese veri gönderemezler. Buna ilaveten taşınabilirlik ağ yolu karakteristiklerini değiştirebilir ki böyle bir durumda yeniden sıralama söz konusu olur ve paketler ESP anti tekrar penceresinden düşerler. Bu durum yeniden anahtarlamayı gerekli kılar.

2.2.7.3. Çoklu Konumluluk

Bir sunucu(taşınabilir ya da sabit) bazen birden fazla ara yüze ya da küresel adrese sahip olabilir. Sunucu sahip olduğu ilave ara yüz ya da adresten eş sunucuyu bir LOCATOR parametresi ile haberdar eder. ESP anti tekrar penceresi ile olabilecek sorunları önlemek için bir sunucu, birçok konumlayıcı kullanacağı zaman paket gönderiminde kullanılacak her bir ara yüz ya da adres için farklı bir güvenlik bağı kullanmalıdır. Bu durum paket gönderimlerinin sıralı yerine eş zamanlı olabilmesine izin verir. Eğer eş sunucuya birden fazla konumlayıcı sunuluyorsa bunlardan hangisinin tercihli olduğu belirtilmelidir (sunucunun veri alacağı konumlayıcının belirlenmesi). Varsayılan olarak aksi belirtilmediği takdirde değiş tokuş işleminde kullanılan konumlayıcı tercihli olarak kullanılır.

Çoklu konumluluk durumunda gönderici birçok geçerli konumlayıcıya sahip olabilir. Pratik olarak, çoklu konumlu bir yapıda HIP bağlantıları hem tercihli eş konumlayıcı hem de tercihli yerel konumlayıcı sahibi olabilir. Yine de kaynak adres seçimi için olan mevcut kurallar, hedef konumlayıcısına dayalı kaynak konumlayıcı seçimine hakim olmalıdır.

Biri çoklu konumlu biri ise tek bir konuma sahip iki sunucunun olduğu bir durumu göz önünde bulunduralım. Çoklu konumlu sunucu, tek konuma sahip sunucuyu diğer adresinden haberdar etmeye karar verebilir. Çoklu konumlu sunucunun yeni adres ile kullanacağı yeni bir güvenlik bağı çifti oluşturması gereklidir. Bunun için çoklu konumlu sunucu sorguyu belirtmek için eski SPI değerini sıfır, yeni SPI değerini ise yeni üretilmiş bir değer olarak belirleyip LOCATOR ile birlikte bir ESP_INFO yollar. Yeni adresi yeni SPI ile bağlantılı hale getirmek için konumlayıcı Type “1”e ayarlanır. LOCATOR parametresi bunun yanında bir Type “1” konumlayıcı daha içerir. Bu ise orijinal adres ve SPI çifti içindir. Parametre işlemeyi basitleştirmek ve protokol eklentilerinin konumlayıcıları kaldırmalarını önlemek için LOCATOR parametresi bir bağlantıda kullanımda olan tüm konumlayıcıları listelemek zorundadır. Çoklu konumlu sunucu eşinden bir ESP_INFO ve UPDATE’ine bir onay(ACK) bekler. Taşınabilirlik

durumunda olduğu gibi eş sunucu yeni adresi kullanmadan önce adresini onaylamak zorundadır.

2.2.8. Güvenlik

HIP ile birlikte sunulan en önemli özelliklerden biri de doğasında bulundurduğu güvenlik özellikleridir. Önceki bölümlerde bahsettiğimiz gibi aradaki bağlantının sağlanabilmesi için IPSec (Kent, 2005) yardımıyla kimlik kontrolünün başarıyla gerçekleşmesi zorunludur. Bu prosedür boyunca güvenli bir ESP bağlantısı için gerekli SA mekanizması çalışmaya devam etmektedir. İkinci olarak, HIP tanımlayıcıları açık anahtarlardır. Bu nedenle HIP tanımlayıcıları, bağlantı sırasında gelecek olan saldırılara karşı korunduğu kadar HIP paketlerini tanıyabilme alışkanlığına da sahiptirler. Ayrıca açık anahtarları tanımlayıcılar gibi kullanabilmek demek açık olarak Public Key Infrastructure (PKI) mekanizmasına gerek olmaması demektir. Üçüncü olarak, DoS saldırılarının etkisi azaltılır. Eğer DoS saldırıları birden fazla I1 paketi kullanarak saldırıya geçerse cevap veren taraf sınırlı ölçüde R1 paketi ile cevap verir. HIP hakkında değindiğimiz tüm bu gelişmiş özelliklerine rağmen, hala ele alınması gereken bazı güvenlik sorunları vardır. HIP bağlantısını kurabilmek için kullanılan mekanizma yeni DoS saldırıları için imkan yaratmaktadır. I2 paketi içinde gelen bilgilerin yorumlanması zaman alacağından bu paketler belirli bir süre sonra cevap veren host tarafında tıkanıklığa sebep olacaktır. Bu sorun I2 paketlerinin aynı HI tarafından sabit bir sayıda kabul edilmesiyle çözülebilir. Ayrıca HIP'e karşı kullanılabilecek birçok sayıda ortadaki adam (Man in the Midde) saldırıları vardır. Bu saldırılara karşı en uygun çözüm yöntemi güvenli ve kimliği doğrulanmış bağlantılar kullanmaktır. Ayrıca, HI elemanları sadece belirli bir DNS alanı tarafından alınabilir. Böylece DNS alanında belirlenen bu HI elemanları HIP paketlerini doğrulamak için kullanılabilir.

HIP paketleri tarafından taşınılan verilere erişim yalnızca aynı HIP bağlantısını paylaşan sunucular tarafından gerçeklenmektedir. Bu sayede birbirleriyle bağlantı kuran iki sunucu aralarında güvenli bir iletişim kanalı bulunmasa dahi güvenli bir şekilde birbirlerine veri iletimini gerçekleyebilirler. Verilerin korunmasına yönelik bu güvenlik koruması sadece sunuculara değil HIP protokolünde kullanılan tüm elemanların geçici veya kalıcı olarak karşılaşabileceği hataları tespit edebilme ve önleyebilme potansiyeline sahiptir. Ayrıca sunucuların birbirlerini tanımlayabilmeleri için gerekli

olan isim alanı mekanizması da güvenliğin üzerinde durduğu konulardan biridir. Görüldüğü üzere HIP protokolünde güvenlik, bahsedilen bu mekanizmaların birbirleriyle uyumlu bir şekilde çalışması sonucu sağlanmaktadır ve herhangi birinin aksaması demek tüm güvenliğin tehlikede olması anlamına gelmektedir.

2.3. HIP TABANLI MİKRO MOBİLİTE YÖNTEMLERİ

2.3.1. µHIP

µHIP; HIP protokolünü, bir ağ geçiş yolu bileşenini ve sayfalama eklentisini ağa dahil ederek genişletmektedir. Bu yeni bileşenin adı Lokal Randevu Sunucusu'dur (LRVS) ve bu LRVS, RVS'nin özelliklerini genişletmektedir. µHIP, bir haberleşme ağını farklı yönetimsel alt alanlara (domain) bölmeyi önermektedir ve bu alt alanların her biri bir adet LRVS ile yönetilmektedir. Kısaca, her alan da bir erişim ağı (access network) ve bir LRVS bulunmaktadır. LRVS, mobil düğümlerin yönetiminden ve µHIP destekli erişim ağlarının internete erişiminden sorumludur. Mobil düğümler lokal IP adreslerini LRVS'ye kaydederler. LRVS küresel (global) ve yerel (local) IP adreslerini eşleştirmekle yükümlüdür. LRVS hem RVS'nin rollerini miras alır hem de Internet'e bir geçiş yolu gibi davranır (Novaczki ve diğ., 2006 – Bokor ve diğ., 2007).

2.3.1.1. Başlatma (Bağlantı Kurma) Prosedürü

Bir mobil düğüm (mobile node-MN) yeni bir yönetim alanına giriş yaptığı zaman, bu alan içinde iletişimde bulanabilmek için yeni bir başlatma mekanizması başlatmak durumundadır. Yeni alana giriş yaptıktan sonra MN bir erişim yönlendiriciye (access router-AR) normal şekilde bağlanır. Bağlantı sağlandıktan ve yeni bir yerel IP adresi aldıktan sonra MN ya bir HIP servis keşfi prosedürü başlatır veya LRVS'nin servis anonsu için bekler. Bundan sonra MN LRVS'nin IP adresi ve HIT'i ile ilgili bilgi sahibi olmaktadır. MN haberleşmekte olduğu diğer düğümlere (corresponding nodes-CN) UPDATE paketlerini alır ve I1 mesajındaki kaynak HIT'i kontrol eder, MN'ye R1 mesajını içeren servis anons paketi (service announcement packet-SAP) ve LRVS'nin bilgisi ile cevap verir. Bunların devamında MN, LRVS'ye olan kaydını I2-R2 mesaj çifti ile sürdürür. µHIP'in normal RVS kaydındaki temel fark; servis keşfi ve kayıt prosedürleri boyunca, LRVS'nin sadece MN'nin HIT'i ile yerel IP adresini (IP_{local}) eşleştiren bir kayıt açması değil, aynı zamanda HIT'i küresel olarak erişilebilen IP adresini (IP_{global}) eşleştirmesidir. MN'nin LRVS'ye kaydından sonra başka CN'ler

tarafından erişilebilir olmak için RVS'ye de bir güncelleme veya kayıt işleminin yapılması gerekmektedir. Tüm adımlardan sonra MN HIT_{MN}-IP_{local}-IP_{global} üçlüsü ile LRVS'ye, HIT_{MN}-IP_{global} ikilisi ile RVS'ye kayıt olmaktadır.

2.3.1.2. Etki Alanı içinde Yer Değiştirme

Eğer bir MN aynı etki alanı içerisindeki başka bir bağlantı noktasına hareket ederse, aynı LRVS servis alanı içerisindeki farklı bir AR'den hizmet almaya başlamaktadır. IP adresinin değiştiğini fark eden MN, LRVS'deki kaydını yeni IP adresi ile günceller. MN'nin bağlı olduğu RVS veya haberleşmekte olduğu düğümler bu hareket ve değişiklikler konusunda bilgilendirilmez. LRVS, etki alanı içerisindeki hareketlerden sorumludur. MN'nin bulunduğu alan dışındaki ağ öğeleri hareketler konusunda bilgilendirilmediği için sinyal yükü, paket kaybı ve yer değiştirme gecikmesi azaltılmaktadır.

2.4.1.3. Etki Alanları arasında Yer Değiştirme

Eğer MN farklı yerel etki alanları içinde hareket ederse, alanlar arası prosedürler tetiklenir. Yeni etki alanına varıldığında, MN yeni yerel IP adresi alır ve yeni LRVS hakkında bilgi sahibi olur. MN'nin yeni HIT'ini ve IP adresini LRVS'den öğrenmesinden sonra, yeni kayıt prosedürünü başlatır. MN, bağlı olduğu LRVS'yi değiştirdiği için, bağlantısını devam ettirebilmek için RVS'sini ve haberleşmekte olduğu düğümleri güncellemelidir. Ancak bu noktada ilk yapması gereken, güncelleme prosedürü sonlandırılana kadar kendisinin eski global IP adresine gelen paketleri yönlendirebilmesi için eski LRVS'yi güncellemesidir. Eski LRVS'nin güncellenmesinden sonra MN RVS'yi ve CN'leri günceller. RVS kayıtlarını MN'nin yeni global IP adresi ile günceller. Tüm güncelleme işlemlerinin bitirilmesini takiben MN eski LRVS'den bağlantısını koparır veya belirli bir zaman aşımından sonra bağlantı otomatik olarak sonlandırılır.

2.3.2. Micro-HIP (mHIP)

mHIP, gereksiz sinyalleme ve kontrol mesajlarını azaltmak için bir HIP eklentisi olarak tasarlanmıştır. mHIP, tanımladığı ağ mimarisindeki tüm mHIP destekli bileşenleri mHIP Görevlisi (Agent) adı verilen yeni ağ bileşenleri olarak tanımlamaktadır. İki tip mHIP görevlisi bulunmaktadır: mHIP geçiş yolu (mHIP-G) ve mHIP yönlendirici (mHIP-R). Bu bileşenlerin ana rolleri alan içi yer değiştirme (intra-domain handover) sürecindedir. mHIP'de, yerel IP adresleri ve HIT'ler arasında eşleşmeler tutulmakta ve

ayrıca tüm mHIP alanını temsil eden ortak bir HIT de tanımlanmaktadır. Bir mobil düğüm alan içi bir yer değiştirme gerçekleştirdiğinde, mHIP bu MN'nin UPDATE mesajını alır ve cevap verir. Daha sonra da HIP tabanlı bağlantıyı yeni konuma yönlendirir (Hon So ve Wang, 2008)

mHIP'de, mHIP-G bir kök yönlendirici olarak hizmet vermekte ve özellikle bağlantı başlangıç aşamalarında µHIP ile ortaya çıkan LRVS'ye benzer bir rol üstlenmektedir. MN'lerin etki alanı içerisindeki kayıtlarını tutmaktadırlar. Bir MN bir mHIP-G'ye kayıt olmaktadır. mHIP-G veri veya sinyal paketlerini aldığında, bu paketleri ilgili MN'ye yönlendirir. mHIP-R ise alan içi yer değiştirmeleri yönetebilmekte ve böylece mHIP geçiş yollarının yükünü azaltarak yer değiştirmenin sisteme getirdiği yükü de azaltmaktadır.

2.3.2.1. Başlatma (Bağlantı Kurma) Prosedürü

Bir MN yeni bir yönetim alanına giriş yaptığı zaman, ağda bulunan mHIP-G'ye kayıt olmak için bağlantı kurma mekanizmalarını başlatması gerekmektedir. MN, HIT ve IP bilgilerini ICMP (Internet Control Message Protocol) anons mesajlarından elde eder ve kayıt prosedürünü başlatır. Aynı etki alanı içerisindeki tüm mHIP görevlileri MN'nin HIT'i ve yeni IP adresi hakkında bilgi sahibi olurlar. Son olarak da, MN yeni bilgileri ile RVS'yi günceller.

2.3.2.2. Etki Alanı içinde Yer Değiştirme

MN'nin aynı etki alanı içerisindeki yer değiştirmesi sırasında devam eden bir iletişimi yoksa mHIP-G, bir UPDATE paketi yollanarak yeni IP adresi konusunda bilgilendirilir. Bu UPDATE paketi, mHIP-G ve MN'nin eski konumu arasındaki yol üzerindeki en yakın mHIP-A tarafından (NmHIPA) yakalanır ve işlenir. MN bu paketi aldığında alan içi yer değiştirme prosedürü tamamlanır. Tüm mHIP-A MN'nin HIT ve IP bilgisini öğrenir. NmHIPA aynı zamanda komşu mHIP görevlilerini de MN hakkında günceller.

MN'nin yer değiştirmesi sırasında eğer devam eden bir iletişim söz konusu ise, MN haberleşmekte olduğu düğüme UPDATE paketi yollar. NmHIPA UPDATE paketini CN'den önce yakalar ve gerekli şekilde imzalanmış paketle CN'ye cevap verir. MN, NmHIPA tarafından talep edilen adres kontrolü ile cevap verdikten sonra alan içi yer

değiştirme prosedürü tamamlanır. NmHIPA kendi eşleştirmelerini günceller ve MN'nin IP adresi değişikliği konusunda komşularını haberdar eder.

2.3.3. Dinamik Hiyerarşik HIP (DH-HIP)

DH-HIP (Yang ve diğ., 2007), HIP için tasarlanmış bir konum yönetimi mekanizmasıdır ve RVS'lerin üç seviyeli mimarisini öne sürmektedir. Bunlar geleneksel HIP mimarisinde tanımlanan klasik RVS, Geçiş yolu RVS (GRVS) ve Lokal RVS (LRVS)'dir. Bunlardan en alt seviye olan LRVS'nin yönettiği alt alanın boyutu ise, mobil düğümlerin paket varış hızına ve LRVS seçiminden sonraki mobilite oranına göre belirlenmektedir.

DH-HIP mimarisinde iki tip yönetimsel alan bulunmaktadır. Bunlar otonom (autonomous) ve yönetimsel (administrative) alanlardır. Yönetimsel alanların yönetiminden LRVS sorumluyken, otonom alanların yönetiminden GRVS sorumludur. Otonom alanlar birden fazla yönetimsel alandan oluşabilmektedir GRVS, kayıt ve bağlantı başlatma sürecinde LRVS'ler ve mobil düğümler arasındaki haberleşmeden sorumludurlar. RVS ise LRVS ile mobil düğümler arasındaki iletişimden sorumludur.

DH-HIP'de yönetimsel alanların boyutu bir anlamda aynı LRVS tarafından yönetilen erişim yönlendiricilerinin sayısıdır. Bu sayı mobil düğümlerin paket varış oranına ve mobilite oranlarına (Call-to-Mobility Ratio) göre dinamik olarak sinyal yükünü azaltmak amacıyla ayarlanmaktadır. DH-HIP'in önemli özelliklerinden biri de tüm AR'lerin LRVS'nin rollerini miras alıyor olmasıdır. Bir MN ağa giriş yaptığında, IP adresini ve HIT'ini sırasıyla LRVS'ye, GRVS'ye ve RVS'ye kaydeder. MN, LRVS'ye direk kaydolur ancak GRVS ve RVS'ye kayıt esnasında, alt seviyedeki RVS tarafından ilgili paketler yakalanır ve MN'nin IP adresi ve HIT'i, paketi yakalayan RVS'nin IP adresi ve HIT'i ile değiştirilir. Daha sonra bu paket asıl hedefi olan üst seviye RVS'ye iletilir. Eğer bir CN, söz konusu MN ile haberleşmek istediğinde, DNS sorgusundan sonra, MN'nin bulunduğu alandan sorumlu olan RVS'nin adresini elde eder. Ayrıca mesajların yakalanması ve iletilmesi süreci BE'nin bazı adımları ile devam eder. DH-HIP'in µHIP'den farkı 3 seviyeli hiyerarşik mimarisidir. µHIP'deki gibi bir alandaki AR sayısı sabit değildir, paket varış hızına ve mobil düğümün durumuna göre ayarlanır.

2.3.4. Gelecek Nesil Ağlar için Bir HIP Eklentisi

Toledo ve diğ. (2009) tarafından geliştirilen bu eklenti temel olarak gelecek nesil kablosuz ağlarda, ağdaki ilgili varlıkları erişim teknolojileri hakkında bilgilendirerek yer değiştirme prosedürünü optimize etmeye dayanmaktadır. Önerilen çözüm için göz önünde bulundurulan senaryoda haberleşmekte olan her iki düğümün de hareket halinde olduğu kabul edilmektedir. Bu eklentinin temel amacı, haberleşmekte olan iki mobil düğümün erişim teknolojilerini değiştirdiği, yani bir nevi dikey yer değiştirme gerçekleştirdiği duruma bir çözüm üretmektir.

Bu eklenti güncelleme prosedürü için VHO_NOTIFY adlı yeni bir mesaj tipi kullanmaktadır. Bu mesaj düğümleri haberleşecekleri bir sonraki erişim teknolojisi hakkında bilgilendirmek için kullanılır. Bu mesaj ile düğümler karşılıklı olarak hangi ara yüzü aktive edeceklerini öğrenirler. VHO_NOTIFY mesajı, yer değiştirme sırasında da bazı parametrelerin (yeni IP adresleri gibi) önceden karşı taraf bildirilmesi rolünü üstlenmektedir. Bu eklentide kullanılan bir diğer yeni mesaj türü ise NEW_UPDATE adlı güncelleme mesajlarıdır. Normal HIP UPDATE mesajları ve NEW_UPDATE mesajları arasındaki temel fark LOCATOR parametresinin içeriği ile ilgilidir. Normal HIP'in aksine, NEW_UPDATE mesajında LOCATOR parametresi paketin sahibinin IP adresini içermeyebilir.

Kısaca, VHO_NOTIFY ve NEW_UPDATE mesajlarının eski erişim teknolojisini kullanarak yollanmasıyla karşı düğümler bir sonraki kullanılacak erişim teknolojisi hakkında bilgi sahibi olurlar. Bu şekilde yer değiştirme ile ilgili bilgiler yer değiştirme başlamanda gönderilmiş olmaktadır.

2.3.5. Eş Zamanlı Hareketlilik için Mobilite Eklentisi

Bu eklentinin temel fikri, haberleşmekte olan iki düğümünün aynı anda hareket ederek konum değiştirdiği eş zamanlı mobilite problemini desteklemek üzere RVS'nin rolünü genişletmektir (Hobaya ve diğ., 2009). Eş zamanlı hareketlilik oluştuğunda, her düğüm kendi RVS'lerini yeni adresleri hakkında bilgilendirmektedir. Bu çözümün temel çalışma prensibi UE-PEER ve UE_RVS mesajlarına dayanmaktadır. İlk aşamada MN'ler yeni IP adreslerini kendi RVS'lerine UE_RVS mesajları ile bildirirler. Daha sonra MN'ler birbirlerine RVS'leri kullanarak UE_PEER mesajlarını yollarlar. Bu

noktada RVS MN'lerin yeni konumlardan haberdar değildir ve bu yüzden bu güncelleme mesajları RVS tarafından MN'lerin eski konumlarına iletildikleri için kaybolurlar. Son aşamada ise, UE_PEER mesajları belirli bir zaman aşımından sonra tekrar yollanırlar ve bu kez HIT-IP eşleme işlemi bittiği için doğru konumlara iletilirler. Böylece UE-PEER mesaj değişimi tamamlanmış olur. UE-PEER mesajlarının ikinci kez iletiminde mesajlar RVS tarafından yakalanmaz ancak üçüncü kez değişimde (echo response talebi gibi) karşılık gelen RVS tarafından aktarılırlar. Üç adet UE-PEER mesaj değişiminden sonra MN'ler veri transferine başlayabilirler.

2.3.6. Çoklu Konumlu Düğümlerde HIP-PMIPv6 tabanlı Konum Yönetimi

Bu çalışmada temel olarak HIP ve Proxy Mobile IPv6 protokollerinin kombinasyonuna dayanan bir konum yönetimi mekanizması önerilmektedir (Iapichino ve Bonnet , 2009). Bu çözüm PMIPv6 için çoklu konumluluk ve teknolojiler arası (dikey) yer değiştirme problemlerine çözüm önermektedir. Makro mobilite prosedürü HIP protokolünde olduğu şekliyle kullanılırken, mikro mobilite için HIP ve PMIPv6'nın bir kombinasyonu tanımlanmıştır.

2.3.6.1. Başlatma (Bağlantı Kurma) Prosedürü

HIP-PMIPv6 kombinasyonu çoğunlukla PMIPv6'nın prosedürleri ve ağ elemanlarını (ör. Mobile Acess Gateway-MAG) kullanmaktadır. HIP'in normal RVS güncelleme prosedürünü, PMIPv6'nın güvenli bağlantı için gerekli olan ayarlamaları ve mesaj değişimleri takip eder. Devam etmekte olan iletişimin varlığı durumunda ise, karşılıklı düğümleri güncellemek için normal HIP güncelleme prosedürleri uygulanır. PMIPv6 tarafından kullanılan IP adreslerinin tipinden dolayı yerel RVS (LRVS) fikri mHIP ve μHIP çalışmalarından olduğu gibi kullanılamamaktadır.

2.3.6.2. Aynı Teknoloji içinde Yer Değiştirme

MN'nin yer değiştirme hareketi süresince konum tanımlayıcısında bir değişiklik olmamasından dolayı HIP aynı teknoloji içinde yer değiştirmeyi algılamaz. Bu noktada tüm prosedür PMIPv6'nın kurallarına göre işler. MN, kendisinde herhangi bir ara yüz değişikliği tespit edemediği için RVS'ye ve CN'lere herhangi bir güncelleme işlemi yapamamaktadır.

2.3.6.3. Farklı Teknolojiler arasında Yer Değiştirme

Farklı teknolojiler arası yer değiştirme bir mobil düğümün devam ettirdiği iletişimi sırasında ikinci bir harbeleşme ara yüzüne geçiş yapmasıdır. Bir MN ikinci arayüzüne geçiş yaparsa, eğer halen aynı etki alanı içerisindeyse aynı Home Network Prefix (HNP) değerini alır. Bu durumda MN bağlı olduğu RVS'ye bir UPDATE mesajı yollamaz ancak haberleşmekte olduğu düğümleri bilgilendirmek için ikinci ara yüzünün yeni IP adresini içeren UPDATE mesajını yollar. MAG bu UPDATE paketini yakalar ve CN'lere yollamaz. Gerekli güncelleme işlemlerini mobil düğümün yerine diğer ağ elemanları üzerinden gerçekleştirir.

2.3.7. HIP Tabanlı Mikro Mobilite Optimizasyonu

Muslam ve diğ. (2009) çalışmasında, HIP'in mikro mobilite davranışını geliştirmek adına her alan için Co-Agent (Co-A) adı verilen yeni bir ağ elamanı önerilmektedir. µHIP'de olduğu gibi LRVS bu yöntemde de kullanılmaktadır. Co-A'nın görevi mobil düğümlerin etki alanı içinde ve etki alanları arasındaki yer değiştirmelerini, hem bir MN hem de CN gibi davranarak yönetmesidir. Her alanın LRVS'si MN'lerin yerel ve global adreslerin eşleştirilmesinden sorumludur. Co-A'nin HIT ve IP'si de MN ile eşleştirilmektedir ve Co-A MN için diğer alt alanlardan yerel IP adresi alabilmektedir. Co-A, MN'nin hareketini inceleyebildiği için yer değiştirme prosedürünü optimize etmektedir.

2.3.7.1. Başlatma (Bağlantı Kurma) Prosedürü

Bir MN yeni bir yönetim alanına giriş yaptığı zaman kendisini LRVS'ye kaydeder. RVS'ye kayıt olmasına gerek yoktur and LRVS'nin DNS'de kaydının bulunması gerekmektedir. Bu yaklaşımda, MN AR'lere yönlendirici istek bildirimi (router solicitation-RS) göndererek bilgilendirme mesajı talebinde bulunmaktadır. Yani, MN Co-A'ya karar vermekte ve kendisini ve bu yeni Co-A bilgisini LRVS'ye kayıt ettirir. LRVS'de yapılan işlemlerden sonra MN'nin Co-A'sı ve CN'nin Co-A'sı arasında güvenli bir bağlantı kurulmuş olur.

2.3.7.2. Etki Alanı içinde Yer Değiştirme

Bu yöntemde AP'ler periyodik olarak Co-A'nin HIT ve IP bilgisini taşıyan bilgilendirme mesajlarını yayınlamaktadır. MN, bu bilgileri kullanarak LRVS'ye kaydını gerçekleştirmektedir. Eğer etki alanı içerisinde bir yer değiştirme olursa, MN'nin yerine Co-A işlemleri gerçekleştirir.

2.3.7.3. Etki Alanları arasında Yer Değiştirme

MN bulunduğu alt alanı değiştirirse, etki alanları arasında yer değiştirme oluşur. MN bu değişimi yine RA mesajları aracılığıyla anlamaktadır ve kendisini Co-A aracılığıyla LRVS'ye kayıt ettirmektedir. MN'nin eski Co-A'sı CN'nin LRVS'sini, MN'nin yeni konumunu MN'nin eski LRVS'si aracılığıyla bilgilendirmektedir. Co-A'lar arasındaki mesaj değişiminden sonra, CN'nin LRVS'si veriyi MN'ye yeni LRVS aracılığıyla iletebilmektedir.

2.3.8. Yerel Sınırlandırılmış Mobilite Yönetimi (Localized-HIP)

L-HIP (Hu ve diğ., 2010), bir önceki kısımda belirttiğimiz çalışmaya benzer şekilde PMIP fikrini temel alan, yerel sınırlandırılmış bir mobilite yönetimi mekanizmasıdır. Bu yönteme ağdaki bazı öğeler mobil düğümlerin hareketlerini izlemekle sorumludurlar. Özellikle etki alanı içindeki yer değiştirmelerin yönetimi amacıyla yeni bir ağ elemanı olan Yerel Mobilite Yönetimi Sunucusunun (Local Mobility Management Server-LMMS) kullanımı önerilmektedir. PMIP protokolünde kullanılan MAG'ların kullanımını da dahil ederek PMIP ve HIP tabanlı bir yer değiştirme yönetim mekanizması önermektedir.

3. MALZEME VE YÖNTEM

Bu tez çalışmasında üzerinden durulan ilk nokta HIP'in mikro mobilite esnasında yer değiştirme kalitesini arttırmaya yöneliktir. Bunun için ilk olarak µHIP'e benzer şekilde yer değiştirme gecikmesini iyileştirmek için RVS'lerin ağdaki yerel amaçlarla kullanımına imkan sağlayan hiyerarşik bir ağ mimarisi önermekteyiz. İkinci olarak da önerdiğimiz ağ mimarisinden kullanılmak üzere daha çok mobilite esnasında yer değiştirme kalitesini arttırmak amacına yönelik olarak bir erken güncelleme mekanizması önerilmektedir. (eHIP) (Gurkas Aydin ve diğ., 2009). Bir sonraki adımda, güncelleme işlemlerini iyileştirmek adına ağ mimarimize uyumlu bir hareket tahmini yöntemi ile eHIP yöntemini için bir eklenti önerilmektedir. Son kısımda ise, önerdiğimiz ağ mimarisinin mesh tipi bir ağ olduğu bir senaryo üzerinden servis kalitesini göz önünde bulunduran bir yol seçme algoritması ile ilgili çalışma önerilmektedir.

3.1. AĞ MİMARİSİ

HIP ve HIP tabanlı mikro mobilite çalışmalarının çoğu (mHIP, µHIP, DH-HIP) HIP güncellemelerini hareketin belirlenmesinden sonra veya yeni bir IP adresi elde edildikten sonra gerçekleştirmektedir. Bu bize Hierarchical MIP (Soliman ve diğ., 2005) gibi proaktif yer değiştirme sunan ve Fast Handovers for MIP (Koodli, 2005) gibi yer değiştirme gecikmesini iyileştiren Mobil IP tabanlı çözümleri hatırlatmaktadır. FMIP, MN'nin yeni konumuna varmadan kayıt işlemlerini tamamlaması şeklinde özetlenebilir. Çalışmanın bu kısmında da HIP'in mikro mobilite sıkıntılarına yönelik olarak önerilen erken güncelleme mekanizmasına yönelik olarak tasarladığımız ağ mimarisi yaklaşımı önerilmektedir.

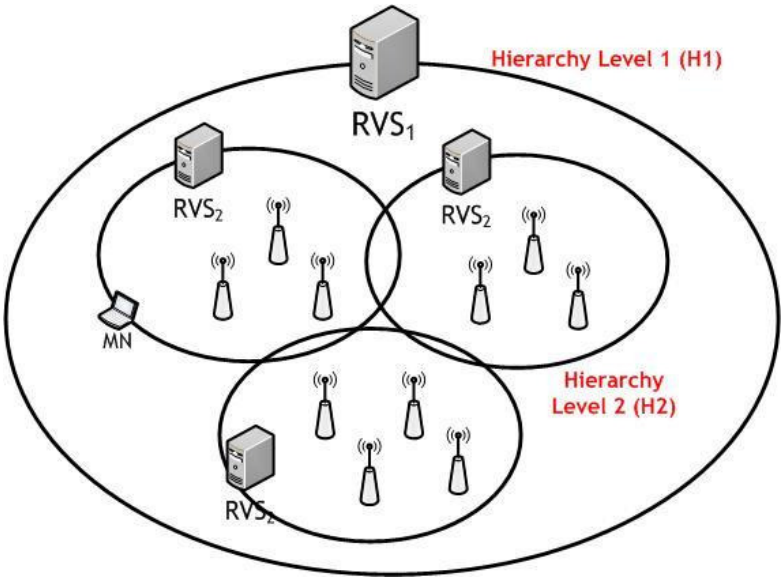
Yer değiştirme için sezinleme (anticipation) ise, yer değiştirmenin, MN'nin bağlı bulunduğu ağ noktasıyla olan bağlantısını tamamen koparmadan, topoloji bilgisi, sinyal gücü, hareket tespiti veya ağ tetikleyicileri gibi bilgilere dayanarak gerçekleşmesidir.

Sezinleme prosedürüne dayanan çoğu çalışmalar genellikle L2 tetikleyicilere dayanır. Ancak sadece bazı teknolojiler L2 tetikleyicilere izin vermektedir. Aslında, L2 tetikleyiciler alındıktan sonra, L3 yer değiştirme zamanı, ağın topolojik durumu veya mobil düğümün hızı gibi birtakım bilgilere bağlı olarak tam ve doğru olarak tespit edilememektedir. Mobil düğümün hızı, yer değiştirmenin sezinlenmesi için çok kritik bir faktördür. Hareket boyunca, L2 tetikleyicinin gerçekleşme noktasına karar vermek, MN'nin bir sonraki hücresini sezinler sezinlemez L3 yer değiştirmenin başlatılması için çok önemlidir.

HIP protokolünün uygulamalarında ve çalışmalarımız için tasarladığımız ağ mimarisinde mikro mobilite sınırlamalarını aşmak amacıyla µHIP'e benzer şekilde LRVS'lerin hiyerarşisi RVS_i şeklinde adlandırılmış ve HIP'in kayıt prosesine ait gecikmeleri minimize etmek amacıyla tasarlanmıştır. Burada "i", hiyerarşi seviyesini göstermektedir. Ağda global ve tek bir ana RVS bulunmaktadır ve bu global RVS, tasarladığımız mimaride RVS_0 olarak adlandırılmıştır. Başlangıçta ve bu çalışmadaki örneklerde i=2 olmak üzere iki seviyeli hiyerarşi olarak kabul edilmektedir. Esasen bu durumda ağda üç seviye RVS bulunmaktadır. Daha sonra ise ağ iki seviyede organize edilen alt ağlara bölünmektedir. İki seviye yerel rol üstlenen RVS bulunmaktadır (RVS_1 ve RVS_2).

3.1.2. Hiyerarşi Seviyeleri

Daha yüksek seviye olan H1 (Hiyerarşi Seviyesi 1), bir veya daha fazla RVS_1 içermektedir ve birçok iç alt alan içerebilen dış alanları yönetmektedir. RVS_1, geniş alanda eHIP kayıt prosesinden sorumludur. Daha düşük seviye olan H2 (Hiyerarşi Seviyesi 2) aynı şekilde bir veya daha fazla RVS_2 içerebilmektedir. RVS_2, bir veya daha fazla kablosuz erişim noktası tarafından hizmet verilen, bir veya daha fazla mobil düğümün kaydından sorumludur. H1 ve H2 hiyerarşi seviyelerinin ağ topolojisindeki yerleşimleri için örnek bir gösterim Şekil 3.1'de bulunmaktadır.



Şekil 3.1. Önerilen Ağ Mimarisinde Hiyerarşi Seviyeleri

Yeni bir RVS'ye yapılması gereken güncelleme esnasında, devam eden haberleşme sırasında ağa fazladan yüksek gecikme ekleyen HIP BE ile kayıt işleminin önceden yapılmış olması gerekmektedir. Tüm hiyerarşi tabanlı çalışmalarda, yeni bir RVS'nin daha önceden kayıtlı olmayan mobil düğümlerin konum güncelleme gecikmeleri konusunda asıl bir yol izlediği ayrıntılı şekilde belirtilmemektedir.

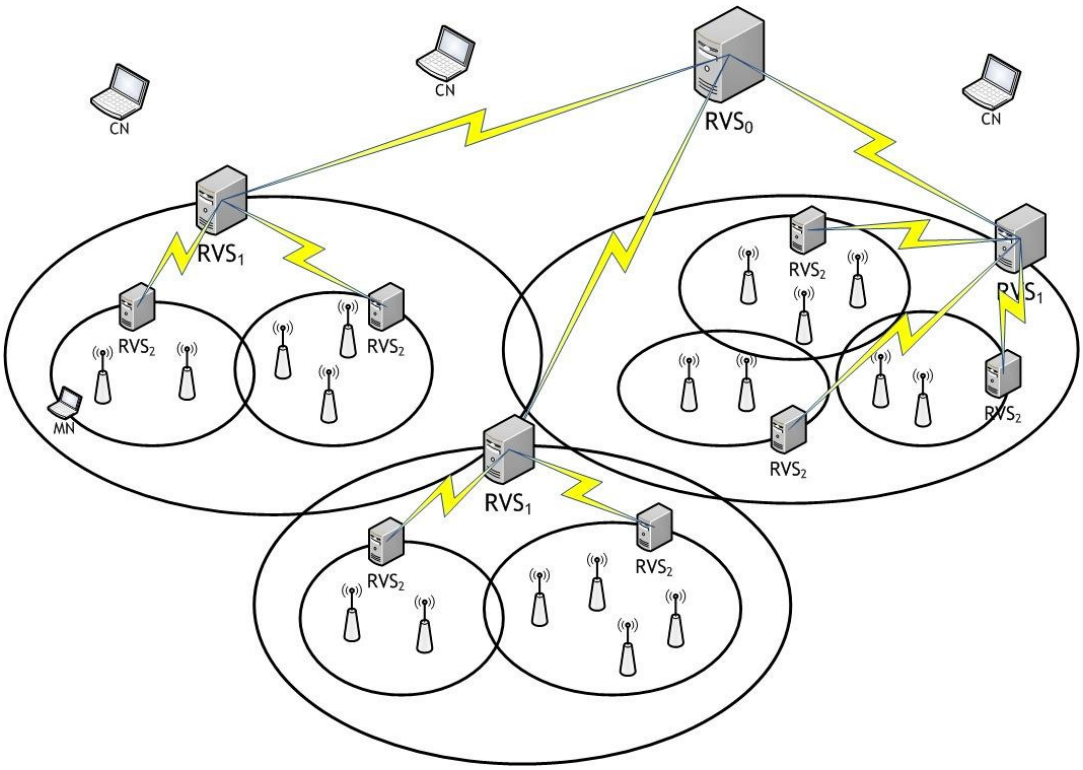
3.1.2. Önceden Kayıt Mekanizması

Çalışmamızda, güncelleme durumunda, yeni RVS'ye kayıt işleminin getireceği ekstra yükten mümkün olduğunca kaçınmak amacıyla, aynı etki alanında yer alan tüm alt (düşük) seviye RVS'lere önceden kayıt olunması önerilmektedir. Tanımladığımız senaryoda, bir mobil düğüm, bir RVS_1 tarafından yönetilen alan girer girmez, o alandaki tüm RVS_2'lere kayıt olmaktadır. Ayrıca, bu kayıt işlemi sırasında mesajlara, mobil düğümün hangi RVS_2'nin alanında "aktif" olarak bulunduğunu belirten bir aktif durum bayrağı eklemek önerilmektedir. Diğer bir deyişle, mobil düğüm o an bağlı olmadığı diğer RVS_2'lere kayıt olurken bu bayrak "pasif" olarak tanımlanmaktadır. Tüm bu tanımlamalara bağlı olarak da, bir mobil düğüm bulunduğu RVS_1 alanını terk ettiğinde, önceden kayıt olduğu tüm RVS_2'ler ile ilgili kayıtlarını kaldırmak durumundadır.

RVS_1 alanına yani H1'e yeni katılan bir MN, ilk kayıt işlemini hizmet aldığı erişim noktasının bağlı olduğu RVS_2'ye başlatır. Ağ mimarisinde yer alan tüm randevu

sunucularının aralarında güvenilir olduğu ve birbirlerinden gelen bilgi ve güncelleme mesajlarını kabul ettikleri varsayılmaktadır.

Hiyerarşik ağ yapısı içeren DH-HIP mimarisinde, sadece bir adet geçiş yolu RVS ikinci bileşen olarak en üst global RVS'nin alt seviyesinde bulunmaktadır. Özellikle bağlantı başlatma sürecinde, DH-HIP mimarisinde tüm bileşenler paketleri yakalar, bazı parametreler ekler ve onları alt seviye bileşenlere iletir. Bu çalışmada BE için çok önemli bir prosedürsel değişiklik önerilmemektedir. Sadece, asıl üzerine değindiğimiz konu olan güncelleme esnasındaki gecikmeyi indirgemek amacıyla aynı alandaki RVS'lere önceden kayıt olunması önerilmektedir.



Şekil 3.2. Önerilen Hiyerarşik Ağ Mimarisi

Mikro mobilite problem göz önünde bulundurulduğunda, normal HIP diğer mikro mobiliteye yönelik çalışmalarda belirtildiği gibi, güncellemek için çok fazla ve uzun süreli sinyal alışverişi gerektirmektedir. Bu da paket kaybını, yer değiştirme gecikmesini (handoff latency-HL) ve yükünü beraberinde getirmektedir. Lokal amaçlı kullanılan RVS'nin temel rolü, mobil düğümün lokal alandaki güncelleme prosesini

yönetmek ve sinyal yükünü minimize etmektir. Global RVS, mobil düğümün hareketini hissetmez ancak RVS_1 tarafından güncellenir. Aslında, düşük seviye RVS (RVS_2), mobil düğümlerin kendi alt ağlarındaki hareketlerinden sorumludur. Yüksek seviye RVS (RVS_1), bir mobil düğüm başka bir RVS_2 tarafından yönetilen başka bir alt ağa geçtiğinde güncellenir. Bu mimarinin avantajı, HMIP'e benzer şekilde, mobil düğümün sıklıkla hareketinde oluşacak sinyal yükünü azaltarak HL'yi de azaltmasıdır.

Eğer bir alt ağ, aynı hiyerarşik alt ağda hareket ederse, aynı RVS_2 tarafından yönetilen ağdaki farklı bir erişim noktasından hizmet almaya başlar. IP adresinin değiştiğini fark eden mobil düğüm RVS_2'deki kaydını yeni adresiyle günceller. Haberleşilmekte olan karşı düğümler (CN) veya RVS_0 bu hareket ve güncellemelerden haberdar edilmez. Mobil düğümün alt ağındaki tüm ağ elemanları haberdar edilmediği için, sinyal yükü azalır ve ayrıca paket kaybı ve HL de normal HIP'den daha iyidir ve böylece HL iyileştirilmiş olur.

3.2. HIP İÇİN ERKEN GÜNCELLEME (EHIP)

Çalışmanın bu kısmında hiyerarşik yaklaşımla beraber HIP için bir erken güncelleme mekanizması tasarlanmıştır. Erken güncelleme (EU) basitçe, bir mobil düğümün hareket etmek istediği yeni ağdaki IP adresi ile ilgili bilgiyi önceden elde etmesi ve yer değiştirme işlemi tamamlanmadan önce kayıt işlemini tamamlamasıdır. EU, mobil düğümün yer değiştirme karar mekanizmasına (handoff decision engine) bağlı olarak farklı parametreler tarafından tetiklenebilir. Bu parametreler, eğer teknoloji tarafından destekleniyorsa, L2 tetikleyiciler olabilir veya FMIP'e benzer şekilde başka parametreler olabilir. Ayrıca, L2 tetikleyicilerin sezinleme prosesindeki dezavantajları ile baş etmek için, Mobil IP'ye yönelik bir erken güncelleme yaklaşımı Kim ve Kim (2007) tarafından önerilmiştir. Bu özellikleri göz önünde bulundurarak HIP Erken Güncelleme eklentisi (eHIP) için temel yer değiştirme prosedürleri tasarlanmıştır.

eHIP'de, Yang ve diğ., (2007) çalışmasında olduğu gibi her erişim noktasının (AP) bir RVS gibi davranmasına gerek bulunmamaktadır. Eğer bir mobil düğüm, erken güncelleme prosedürünü başlatmak isterse, bir sonraki AP'yi kapsayan (hizmet alacağı AP) RVS'nin IP adresi bilgilerini elde etmektedir. Bir sonraki AP'yi ve RVS'nin IP

adresini belirlemek için ise, servis keşfi (service discovery) mekanizması mimarimizi tamamlamaktadır. Bunu yapmanın en kolay ve uygun yolu ise IP adresi bilgilerinin duyurma (advertisement) mesajarının içinde tüm ağa anons edilmesidir. Daha önce belirtildiği gibi, mobil düğümlerin bulunduğu alandaki tüm RVS'lere pasif durumda kayıtlı olduğu unutulmamalıdır.

3.2.1. Kavramlar ve Mesaj Tipleri

Örnek mimari üzerinde ve tasarlanan yer değiştirme mekanizmalarında kullanmakta olduğumuz kavramlar ve eHIP'de kullanılan yeni mesaj tipleri bu bölümde belirtilmektedir.

oAP, mobil düğümün o an bağlı bulunduğu AP'yi, nAP ise hareket etmek istediği bir sonraki AP'yi göstermektedir. Erken güncelleme mekanizması temel olarak iki farklı mesaj tipini kullanmaktadır: Early Update (EU) ve Finish Update (FU). EU mesajları kendi içinde numaralandırılarak hem izledikleri yol belirtilmiş hem de farklı öğelerden gönderilen mesajların anlaşılması sağlanmıştır. Tablo 3.1 kullanılan mesajları ve kavramları kısaca özetlemektedir.

Tablo 3.1. eHIP'de Kullanılan Kavramlar ve Mesajlar

oAP	MN'nin bağlı olduğu, eski AP
nAP	MN'nin hareket etmek istediği bir sonraki AP
oRVS2	MN'nin bağlı olduğu, eski RVS2
oRVS1	MN'nin bağlı olduğu, eski RVS1
nRVS2	MN'nin hareket etmek istediği bir sonraki, yeni RVS2
nRVS1	MN'nin hareket etmek istediği bir sonraki, yeni RVS1
EU	MN'den ilk yollanan Early Update Mesajı
EU1	Eski RVS2'den yeni RVS2'ye yollanan EU bildirim mesajı
FU	Finish Update Mesajı
NEW_HOST_REG	RVS'ler arası yeni kayıt ekleme mesajı
DELETE_HOST_REG	RVS'ler arası kayıt silme mesajı
EU2(OK)	Yerdeğiştirme esnasında yeni RVS2 den gelen doğrulama mesajı
EU2(ERROR)	Yerdeğiştirme esnasında yeni RVS2 den gelen hata mesajı
EU3(OK)	Yerdeğiştirme esnasında eski RVS2 den MN'ye gelen doğrulama mesajı
EU3(ERROR)	Yerdeğiştirme esnasında eski RVS2 den MN'ye gelen hata mesajı
FU1	Yeni RVS2'den CN'lere yollanan güncelleme mesajı

Tablo 3.2, eHIP prosedüründe kullanılan mesajları, içeriklerini ve görevlerini ayrıntılı şekilde göstermektedir.

Tablo 3.2: eHIP'de Kullanılan Mesajlar ve İçerikleri

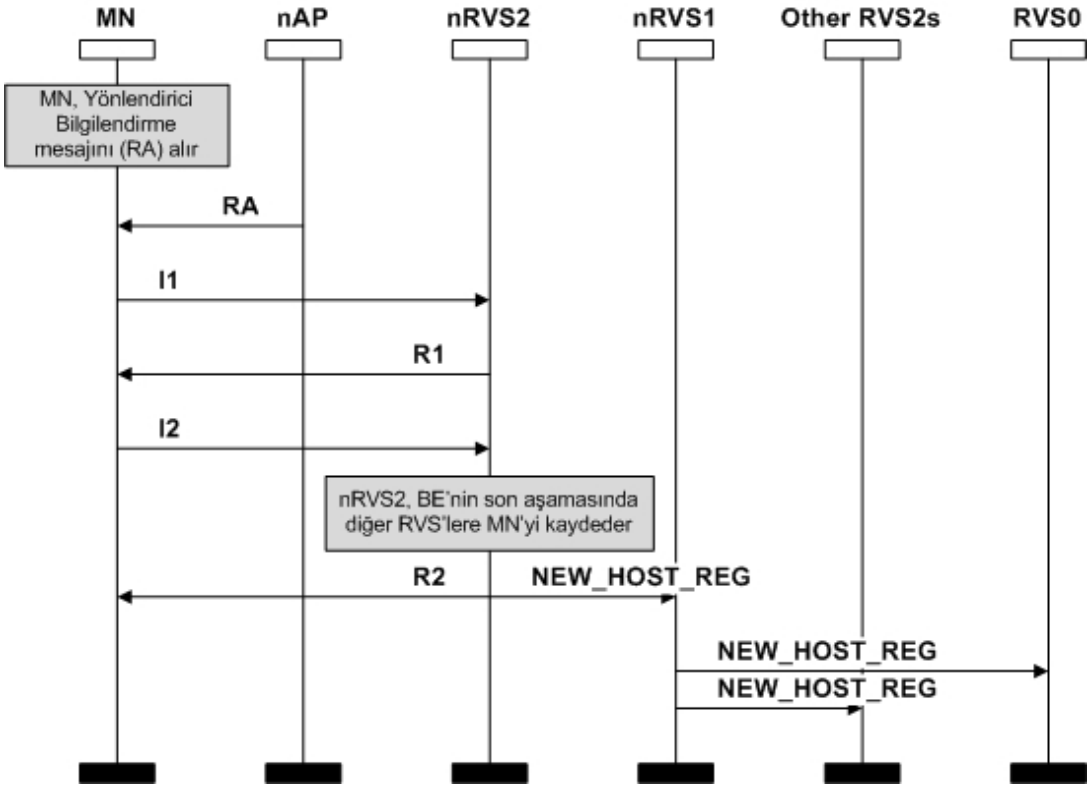
Mesaj Adı	**İçerik**	**Açıklama**
EU	srcHIT srcIP destHIT destIP nRVSHIT newHostIP	MN'nin kapsama alanına girdiği yeni bir AP'den aldığı RA mesajında kayıtlı olduğu RVS'den farklı bir RVS bilgisi geliyorsa, bu RVS'ye erken güncelleme yapmak için oRVS2'ye EU mesajı ile nRVS'nin bilgilerini ve yeni AP'den almak istediği IP bilgisini gönderir.
EU1	srcHIT srcIP destHIT destIP HostHIT HostResvIP	oRVS2'nin MN'nin erken güncelleme isteğini nRVS2'ye bildiren mesajdır. İçerisinde MN'nin HIT bilgisini ve alacağı IP adresi barındırır.
EU2	srcHIT srcIP destHIT destIP AckHostHIT statusFlag	nRVS2'nin oRVS2'den EU1 ile gelen talebe döndüğü cevap mesajıdır. İçerisinde ilgili hostun HIT bilgisini ve işlemin başarılı olup olmadığını belirten durum bayrağını barındırır.
EU3	srcHIT srcIP destHIT destIP AckRvsHIT statusFlag	oRVS2, nRVS2'den gelen cevaba göre MN'ye erken güncelleme işleminin başarılı olup olmadığını bildiren mesajdır. İçerisinde durum bayrağı ile kayıt yapılan nRVS2'nin HIT bilgisini barındırır.
FU	srcHIT srcIP destHIT destIP CnHITList CnIPList	MN'nin nRVS2'ye onun alt alanına geldiğini bildiren ve erken güncelleme işlemini sonlandıran mesajdır. İçerisinde MN'nin bağlı olduğu CN'lerin HIT ve IP listesini barındırır.
FU1	srcHIT srcIP destHIT destIP MnHIT MnIP	nRVS2'nin CN'lere kendisine yeni gelen MN'nin IP bilgisini gönderdiği mesajdır. İçerisinde MN'nin HIT ve IP bilgisini barındırır.
NEW_HOST_REG	srcHIT srcIP destHIT destIP newHostHIT	RVS'lerin kendi alt alanlarına yeni gelen MN'lerin bilgilerini üst seviye RVS'ye bildirdiği mesajdır. İçerisinde yeni hostun HIT bilgisini barındırır. Üst seviye RVS'ler, MN-RVS eşleştirmelerini bu mesaj doğrultusunda gerçeklerler. İçerisinde yeni eklenecek MN'nin HIT bilgisini barındırır.
DELETE_HOST_REG	srcHIT srcIP destHIT destIP oldHostHIT	Üst seviye bir RVS'nin (RVS1 veya RVS0), NEW_HOST_REG mesajını aldığında ilgili MN için farklı bir alt alanda kaydı mevcutsa, bu alt alandaki kendinden alt seviyedeki RVS'ye, ilgili MN kaydının silinmesi için gönderdiği mesajdır. İçerisinde silinecek MN'nin HIT bilgisini barındırır.
RA	RvsHIT RvsIP	IPv6'da bulunan RA mesajının RVS bilgileri içerecek şekilde uyarlanmış halidir. İçerisinde, ilgili IPv6 yönlendiricisinin içerisinde bulunduğu alt alanın RVS'sinin HIT ve IP bilgilerini barındırır.

eHIP mekanizmasını şu şekilde özetleyebiliriz: MN'nin hareketi esnasından bir sonraki RVS_2'nin ve AP'nin sezinlenmesi ve IP adreslerinin keşfedilmesinden sonra, MN o an bağlı bulunduğu $oRVS_2$'ye, ardından $oRVS_2$ de $nRVS_2$'ye EU mesajlarını yollar. $nRVS_2$'nin gerekli işlemleri yapması sonucu MN'ye yine $oRVS_2$ üzerinden cevap mesajları döner. Bu işlemlerden sonra, mobil düğüm yeni konumuna hareket eder ve nAP'nin kapsama alanına ulaşmış olur. Burada yeni $nRVS_2$'ye doğrudan FU mesajını yollar ve nAP'ye vardığını konfirme eder. Bu noktadan sonra nAP'den hizmet almaya başlayabilir. FU mesajını aldıktan sonra $nRVS_2$, FU1 mesajlarını eğer mevcutsa karşı düğümlere yollar. NEW_HOST_REG mesajı ile de $nRVS_1$'e yollayarak tüm hiyerarşinin güncellenmesini tetikler. Mobil düğümden karşı düğümlere veri iletimi ilk FU mesajının yollanmasını takiben başlayabilirken, mobil düğümler, nAP aracılığıyla karşı düğümlerden veri almaya FU1 işlemleri bittikten sonra başlarlar.

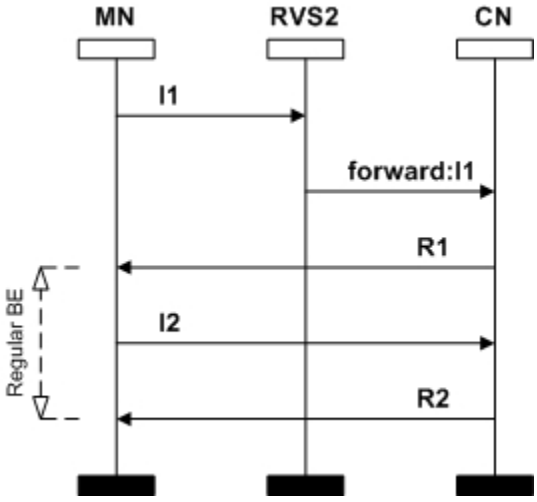
3.2.2. Bağlantı Başlatma Prosedürü

Önerilen ağ mimarisinde, bölüm 3.1.2.'de belirtildiği gibi önceden kayıt mekanizması mevcuttur. Bir mobil düğüm, bir RVS_1 tarafından yönetilen alan girer girmez, o alandaki tüm RVS_2'lere kayıt olmaktadır. Bu süreç de bir anlamda bağlantı başlatma prosedürüdür. Şekil 3.3 önceden kayıt ve bir H1 alt alanına girildiğinde gerçekleştirilen mesaj akış şemasını göstermektedir. Bir MN, yeni bir H1 alanına girdiği zaman ilk I1 mesajını $nRVS_2$'ye yollar. $nRVS_2$ ve MN arasında BE kayıt prosedürü devam ederken, I2 mesajını aldıktan sonra $nRVS_2$, BE'yi sonlandıran R2 mesajı ile aynı anda bir üst seviye RVS_1'e, ağdaki diğer RVS_2'lere ve RVS_0'a bu yeni MN'nin kaydı gerçekleştirilir.

MN'nin bir CN ile haberleşmek istediği durumda BE prosedürünü başlatmak durumundadır. Hiyerarşik yapının bize sağladığı yöntem ile CN'nin farklı seviyelerde olmasına bağlı olarak mesaj akışı sağlanmaktadır. Şekil 3.4 MN ile CN'nin aynı RVS_2'nin yönetimi altında olması durumunda bağlantı kurulumunun nasıl gerçekleşeceğini göstermektedir. MN, BE'yi başlatan I1 mesajını bağlı bulunduğu RVS_2'ye yollar. RVS_2 eğer CN'yi kendi alanında bulursa, I1 mesajını iletir ve BE'nin geri kalan kısmı MN ve CN arasında normal şekilde devam eder.



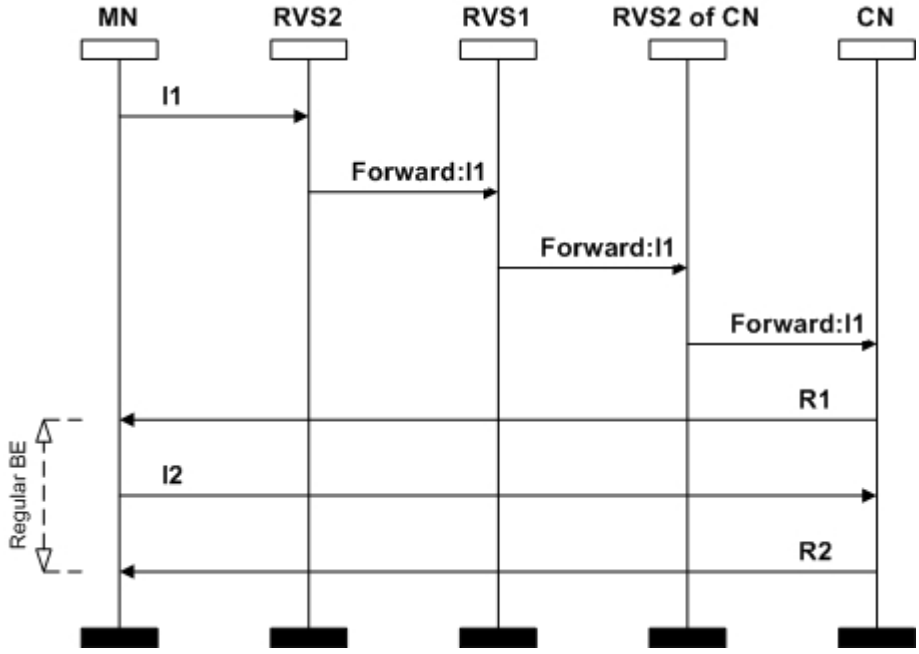
Şekil 3.3: Bağlantı Başlatma/Önceden Kayıt Mekanizması Mesaj Akış Şeması



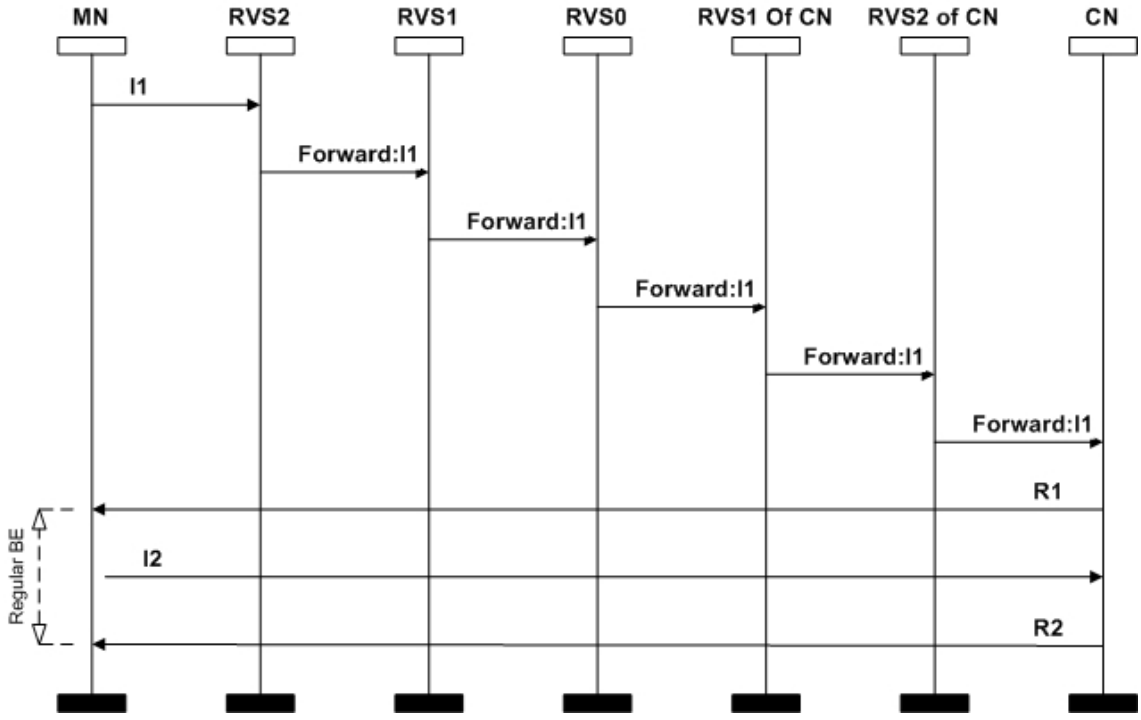
Şekil 3.4: MN ile CN'nin aynı H2'de olması durumunda Bağlantı Kurulumu

Şekil 3.5. ve Şekil 3.6 sırasıyla CN'nin farklı H2'de ve farklı H1'de olması durumunda bağlantı kurulumunun nasıl gerçekleştiğini göstermektedir. MN, I1 mesajını bağlı olduğu RVS_2'ye yolladıktan sonra, RVS_2 kendi alanında CN'nin kaydını bulamazsa I1 mesajını kendisinden bir üst seviyede olan RVS_1'e yollar. RVS_1 eğer CN'nin kendisine

bağlı başka bir RVS_2'de olduğunu tespit ederse I1 mesajını bu RVS_2'ye gönderir. Bu RVS_2'nin de mesajı CN'ye iletmesi sonucu BE'nin geri kalan adımları MN ve CN arasında devam eder.



Şekil 3.5: CN'nin MN'den farklı H2'de olması durumunda Bağlantı Kurulumu

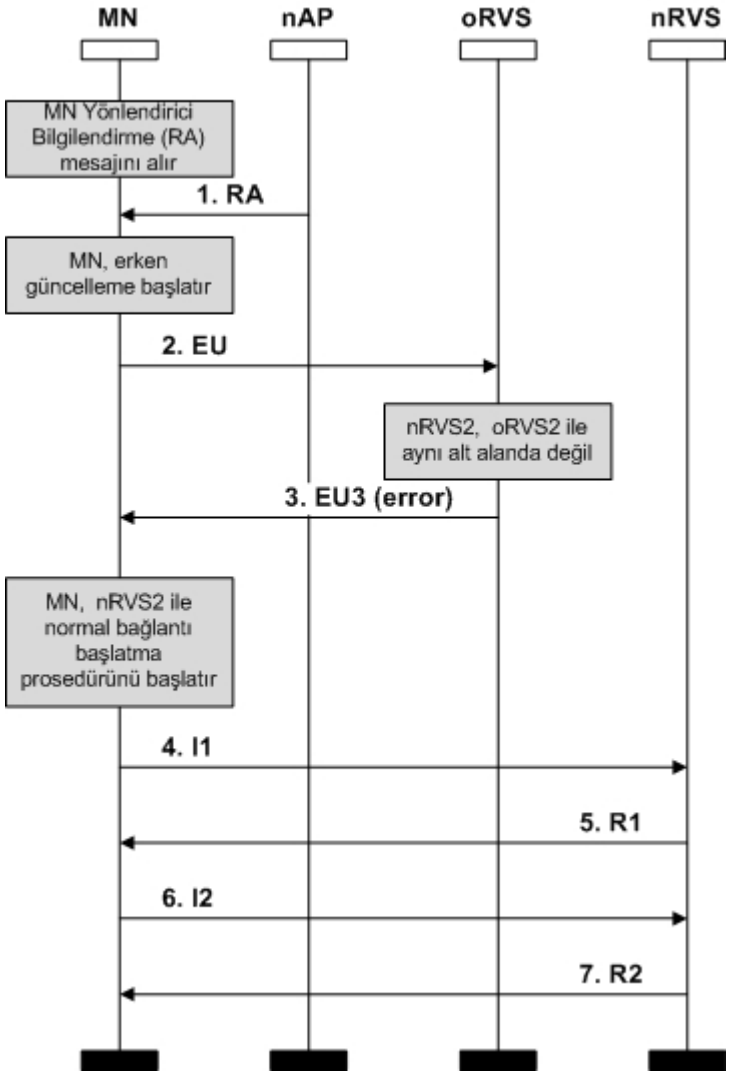


Şekil 3.6: CN'nin MN'den farklı H1'de olması durumunda Bağlantı Kurulumu

CN'nin farklı H1'de olması durumda ise I1 mesajı hiyerarşinin en üst seviyesine kadar gerekli kontroller yapılarak iletilir. Sırasıyla tüm RVS'ler CN'nin kontrolünü yaparlar ve gerekliyse bir üst seviye RVS'ye mesajı iletirler. RVS_0'a ulaşan I1 mesajı yine aynı hiyerarşiyi takip ederek CN'ye ulaşır ve BE'nin geri kalan kısmı normal şekilde decam eder.

3.2.3. Hiyerarşi Seviyesi 1 Yer Değiştirme Prosedürü (H1H)

H1 yer değiştirme (H1 handover-H1H) bir mobil düğümün iki farklı H1 alt alanı arasında yer değiştirme gerçekleştirmesi yani RVS_1 değiştirmesi anlamına gelmektedir. Şekil 3.7 H1H için mesaj akış şemasını göstermektedir.



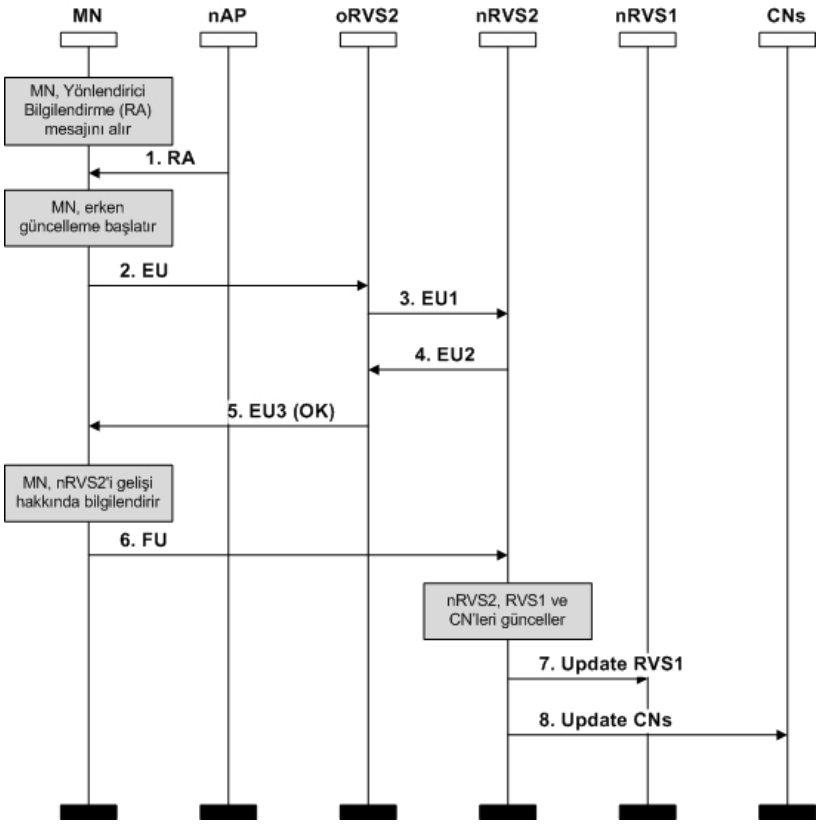
Şekil 3.7: H1H Mesaj Akış Şeması

Bir MN hareket etmeye başladığında ve bağlı olduğundan farklı bir AP'den yönlendirici bildiri mesajı (router advertisement-RA) aldığında, ilk EU mesajını direkt olarak bağlı olduğu RVS'ye (oRVS$_2$) gönderir. Mimarimize göre aynı H1'de olan bir RVS$_2$ birbirlerinden haberdardırlar. EU mesajı, MN'nin HIT'i, yeni AP'nin IP bilgisini ve dahil olunacak yeni nRVS2 bilgisini içermektedir. Bu noktada, bir RVS bir EU mesajı aldığı zaman, talep edilen nRVS$_2$'nin kendisiyle aynı H1 alt alanında olup olmadığını kontrol eder. Eğer talep edilen nRVS$_2$ farklı bir H1 alt alanında ise, oRVS$_2$ hata bilgisi içeren EU3 mesajını MN'ye cevap olarak yollar. Bu mesajı alan MN, farklı H1 alanında olan nRVS$_2$ ile sıfırdan yeni bir kayıt prosedürü başlatır. Bu duruma eHIP'de H1 yer değiştirme (H1H) adı verilmektedir. MN, nRVS$_2$'ye kaydolduktan sonra, nRVS$_2$ üst seviyedeki nRVS$_1$'i ve aynı alt alandaki diğer RVS$_2$'leri daha önce belirtildiği şekilde günceller.

3.2.4. Hiyerarşi Seviyesi 2 Yer Değiştirme Prosedürü (H2H)

H2 yer değiştirme (H2 handover-H2H) bir mobil düğümün aynı H1 alt alanı arasında farklı H2 alt alanları arasında yer değiştirme gerçekleştirmesi yani RVS$_2$ değiştirmesi anlamına gelmektedir. Şekil 3.8 H2H için mesaj akış şemasını göstermektedir.

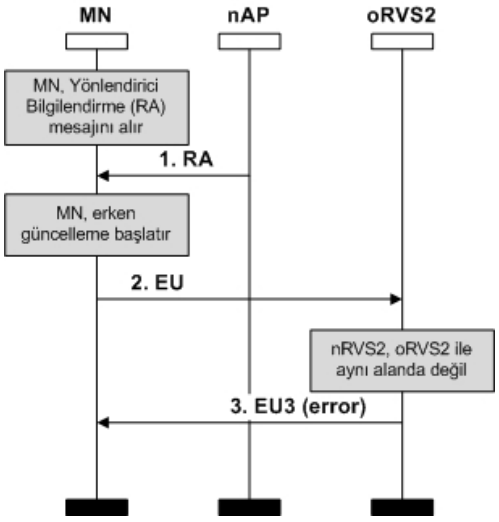
oRVS$_2$ bir MN'den EU mesajı aldığı zaman, eğer talep edilen nRVS$_2$ aynı H1 alt alanı içindeyse, EU1 mesajını nRVS$_2$'ye yeni bir kayıt/güncelleme talep etmek için yollar. Eğer nRVS$_2$ bu mesaja EU2 mesajı ile cevap verirse, nRVS$_2$ MN'nin kendi alt alanına varışıyla ilgili gerekli ön ayarlarıyla başarıyla yapmış ve bu güncellemeyi kabul ediyor demektir. EU2 mesajını alan oRVS$_2$, MN'ye doğrulama bilgisi içeren EU3 mesajı ile cevap verir. EU3 mesajını alan MN, artık FU mesajını nRVS$_2$'ye gönderebilmektedir. MN'nin nRVS$_2$'ye direk olarak mesaj yollayabilmesi, eHIP'in aynı H1 alt alanı içerisindeki tüm RVS$_2$'lere önceden kayıtlı olma özelliğinden kaynaklanmaktadır. Bu noktada eğer MN, eski bağlı bulunduğu alt ağı tamamıyla terk etmeden önce EU3 mesajını alamazsa, yeni nRVS$_2$ ile sıfırdan bir kayıt prosedürü başlatmak zorundadır.



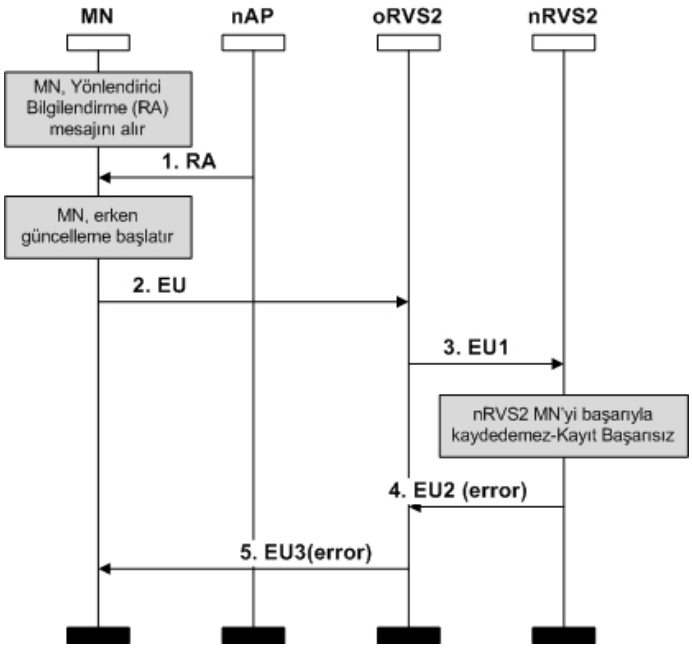
Şekil 3.8: Başarılı durumda H2H Mesaj Akış Şeması

FU mesajı ile MN, eğer varsa haberleşmekte olduğu karşı düğümlerin bilgisini de yollayabilmektedir. $nRVS_2$, üst seviyedeki $nRVS_1$'i ve eğer varsa CN'leri FU mesajını aldıktan sonra güncellemektedir. MN'nin FU mesajından sonra göndermekle yükümlü olduğu bir mesaj yükü bulunmamaktadır. Eğer bir EU olayı tetiklendiğinde ve $oRVS_2$ talepte bulunulan $nRVS_2$'nin farklı H1 alanında olduğunu tespit ettiğinde, daha önce belirtildiği gibi hata bilgisi içeren bir EU3 mesajı ile direk olarak MN'ye cevap verir. Bu durum Şekil 3.9'da da gösterilmektedir. Bundan sonra MN, yeni bir H1N başlatacağını fark eder ve kayıt prosedürünü hazırlar.

H2H ile ilgili oluşabilecek bir diğer senaryo ise $oRVS_2$ tarafından talep edilen güncelleme işleminin başarıyla $nRVS_2$ tarafından tamamlanamaması durumudur. Bu durumda, $nRVS_2$, $oRVS_2$ ile aynı H1 alanında ise, MN'nin gerekli IP detaylarını içeren EU1 mesajını $oRVS_2$'den almasına rağmen, gerekli ön ayarları ve güncellemeleri başarıyla gerçekleştiremeyebilir. Bu durumda $nRVS_2$, $oRVS_2$'ye hata bilgisi içeren EU2 mesajıyla cevap verir ve bu mesajı alan $oRVS_2$ MN'ye aynı bilgiyi içeren EU3 mesajıyla cevap verir. Şekil 3.10 bu senaryoyu gösteren mesaj akış şeması gösterilmektedir.



Şekil 3.9: $nRVS_2$ aynı H1 alanında olmadığında H2H Mesaj Akış Şeması



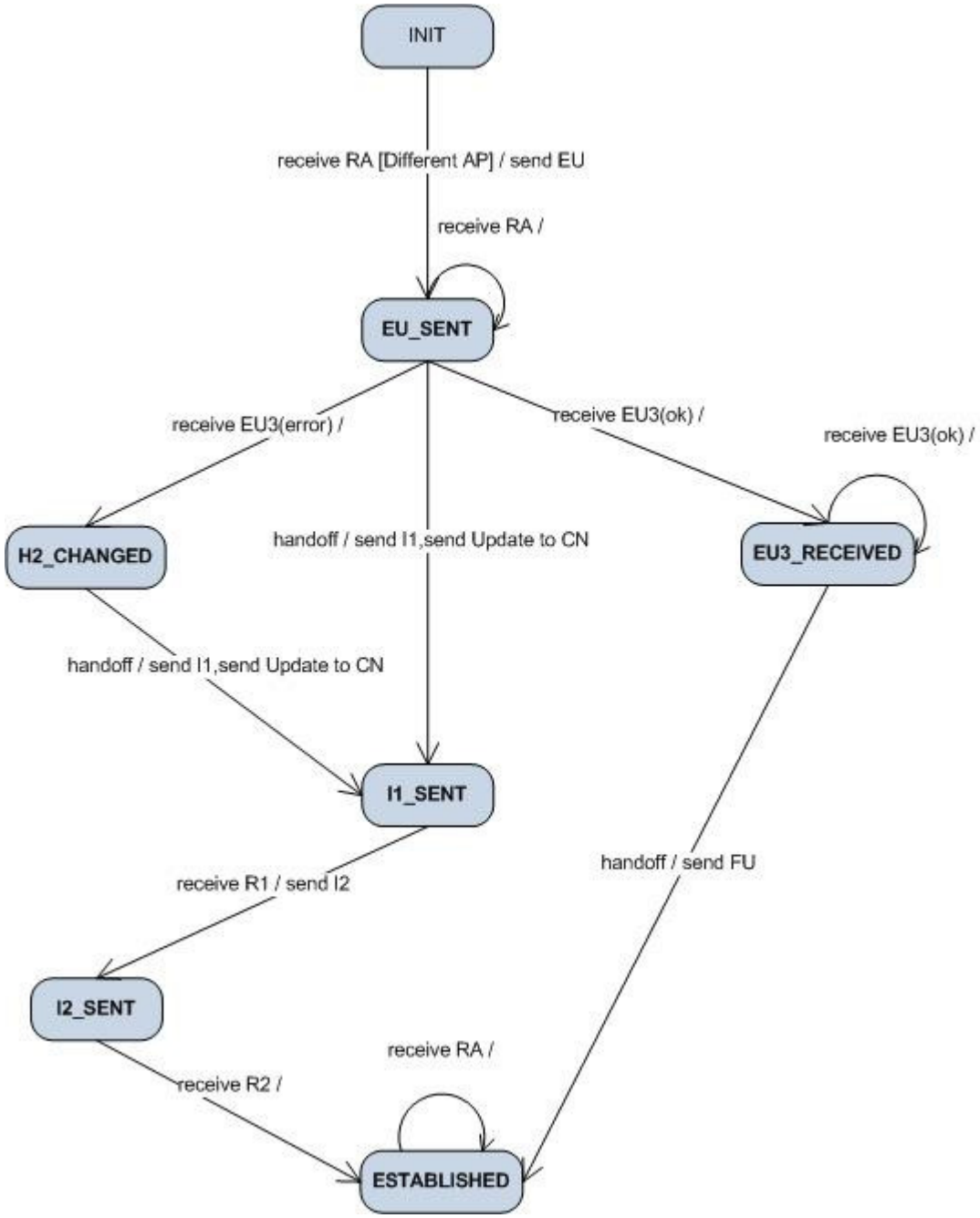
Şekil 3.10: $nRVS_2$'de başarısız güncelleme durumunda H2H Mesaj Akış Şeması

3.2.5. Hiyerarşi Seviyesi 2 içinde Yer Değiştirme

Bir MN hareket etmeye başladığında ve bağlı olduğundan farklı ancak bulunduğu H2'ye ait başka bir AP'den yönlendirici bildiri mesajı (router advertisement-RA) aldığında, EU mesajını direkt olarak bağlı olduğu RVS'ye ($oRVS_2$) gönderir. RVS_2, zaten kendisinde bulunan MN'nin yine kendisine farklı bir IP üzerinden bağlanacağı bilgisini almış olmaktadır. Daha sonra MN'ye kendisine yeni IP ile bağlanabileceğini

bildiren EU3 mesajını göndermektedir. MN adres değişikliğini tamamladığında bağlı olduğu CN bilgilerini FU mesajı ile RVS_2'ye iletir. RVS_2,tüm CN'leri FU1 mesajı ile adres değişikliği hakkında bilgilendirmektedir. Bu sayede üç aşamalı klasik HIP güncelleme işlemi RVS üzerinden gerçekleştirilerek HL düşürülmüş olmaktadır.

Şekil 3.11, eHIP için MN açısından durum geçiş diyagramını göstermektedir.



Şekil 3.11: eHIP durum Geçiş Diyagramı

3.3. EHIP İÇİN HAREKET SEZME EKLENTİSİ (P-EHIP)

eHIP, bir mobil düğümün (MN) hareketi sonucu gereken güncellemenin, MN'nin bağlı olduğu alandan tamamen ayrılmadan önce yapılmasına dayanmaktadır. Early Update (EU) prosedürü, bir MN hizmet almak üzere olduğu AP ile ilgili bilgileri içeren bir Router Advertisement Mesajını (RA) alması sonucu tetiklenir. Çalışmanın bu kısmında eHIP'e yeni bir işleyiş modu olarak bir eklenti önerilmektedir. Bu eklentinin temel fikri, bir sonraki bağlantı noktasından (point of attachement-PoA) sinyal almadan önce erken güncellemenin tetiklenmesi fikrine dayanmaktadır. Burada sinyalden kastımız eHIP prosedüründeki RA mesajlarıdır.

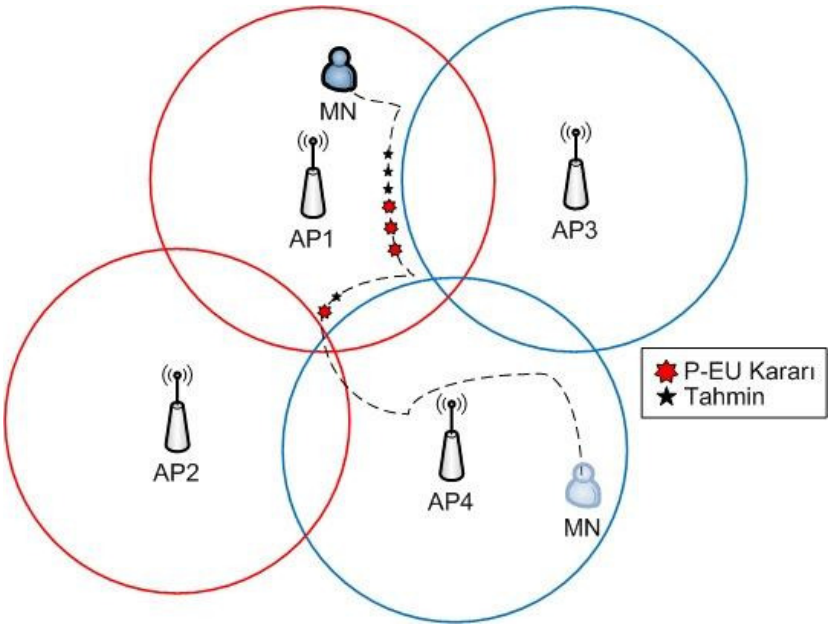
eHIP mimarisinde bu tahmine dayalı modun çalışması için ağ mimarisinde bir takım ek özelliklerin olduğunu varsaymaktayız. Diğer bir deyişle bu eklenti ile beraber eHIP mimarisine de ek özellikler önerilmektedir.

3.3.1. eHIP Mimarisi İçin Önerilen Yeni Özellikler

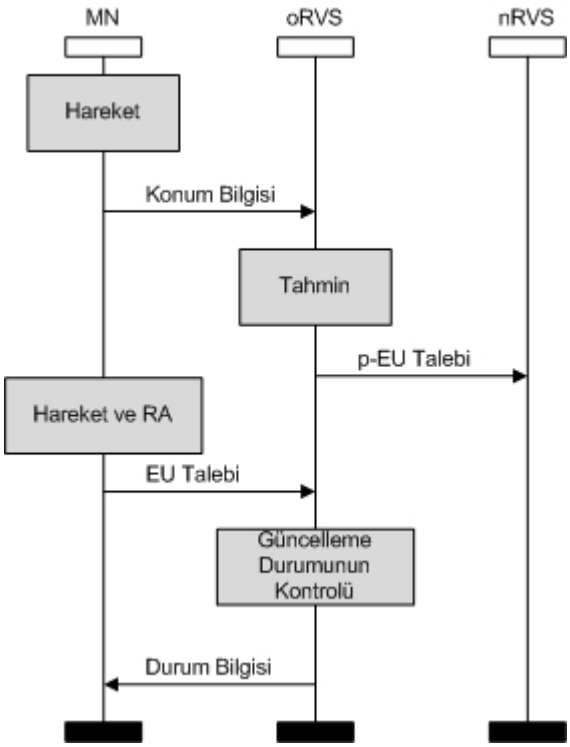
eHIP mimarisinde bu modda çalışabilmek için randevu sunucularının (RVS) ağdaki diğer randevu sunucuları ve erişim noktaları (AP) hakkında yani ağın topolojisi hakkında bilgi sahibi olduğu kabul edilmektedir.

Bu yeni eklentinin tasarımı ile eHIP'in iki modu bulunmaktadır. Bu iki mod, yapılan tahmin ve güncellemenin başarısına göre beraber hareket edebilmekte yani aynı anda işleyebilmektedir. p-eHIP modunun işleyiş sorumluluğu tamamen randevu sunucularına bırakılmaktadır. Bir RVS, tahmin için gerekli şartlar sağlandığında, bu tahmine dayalı olarak, hareket etmekte olan bir MN için erken güncelleme gerçekleştirir. Şekil 3.12 bu tahminlerin örnek bir yol üzerinde ne şekilde gerçekleşebileceğine dair örnek göstermektedir.

Bir MN hareketini sürdürür ve normal eHIP prosedürüne göre EU talebinde bulunursa, RVS kendisine daha önceki tahmini sonucu talep edilen PoA için başarıyla güncellediği bilgisini verir. Eğer tahmine dayalı bir güncelleme yoksa veya tahmin yanlışsa, RVS normal EU prosedürüne geçiş yapar ve geri kalan işlemleri normal şekilde sürdürür. MN, bu durumda prosedürün işleyişinde bir değişiklik hissetmez. Şekil 3.13 p-eHIP'in genel işleyişini göstermektedir.



Şekil 3.12: p-eHIP'de İzlenen Yol Üzerinde Tahminler ve EU Kararları



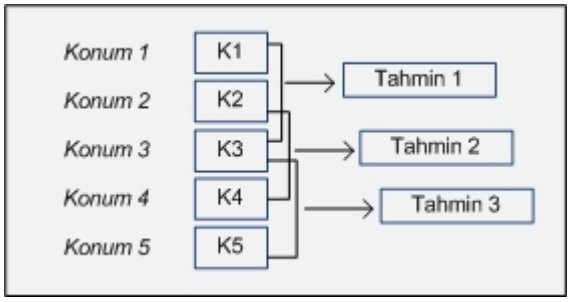
Şekil 3.13: p-eHIP Genel İşleyişi

3.3.2. Tahmin Metodu

p-eHIP yönteminin uygulanabilmesi için bir tahmin metoduna ihtiyaç duyulmaktadır. p-eHIP yönteminde uygulanmakta olan tahmin metoduna göre getirilen ek özellikler şu şekildedir:

i. MN'lerin konum bilgilerini periyodik olarak bağlı oldukları RVS_2'ye yollamaları
ii. RVS'lerin bu konum bilgilerini MN'lerin hareketlerini izlemek amacıyla tablo şeklinde kayıt altına almaları

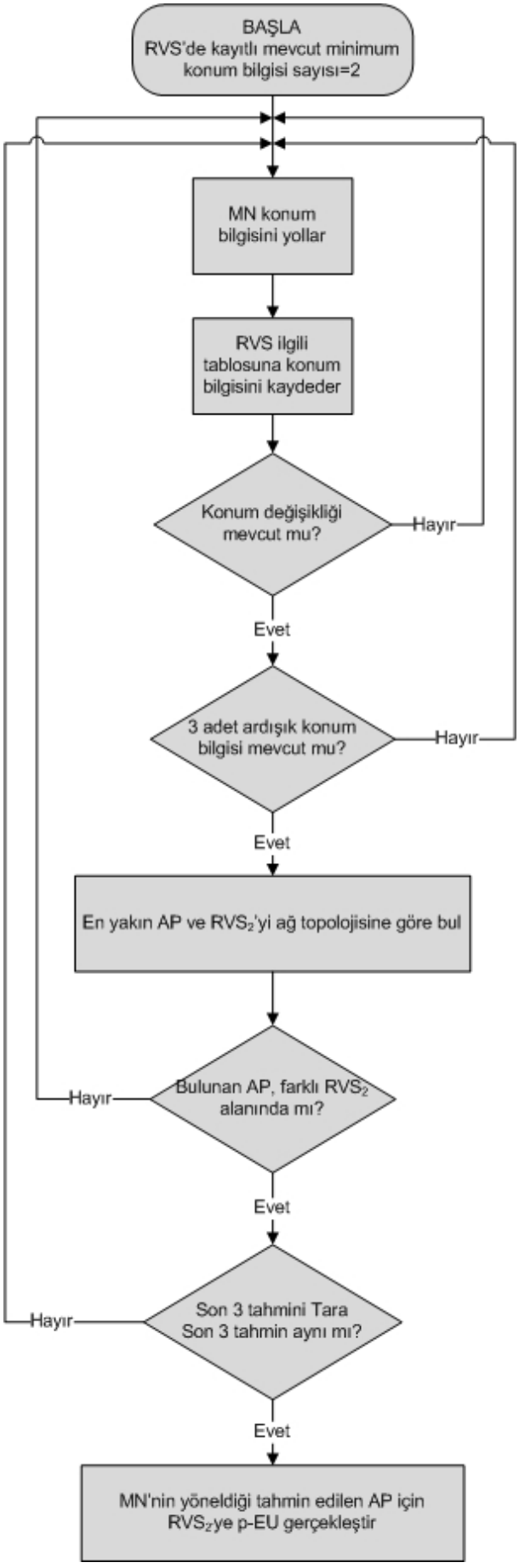
RVS'ler tarafından tutulan kayıtlara göre MN'nin hareket ettiği tespit edilirse tahmin hesaplama metotları tetiklenmektedir. MN'nin hareket ettiğinin tespiti en basit şekliyle konum değişikliğinin tespiti ile gerçekleştirilmektedir. Şekil 3.14 konum bilgilerine göre tahmin mekanizmasının blok şemasını göstermektedir.



Şekil 3.14: Koordinatlara göre Tahmin Şeması

RVS, ağın topolojik bilgisine sahip olduğu için, MN'den alınan son konum bilgilerine bağlı olarak bir sonraki AP ve RVS tahmin edilir. Ancak, bir sonraki AP ve RVS tahmini her bir konum bilgisi ile yapılabildiği halde, erken güncellemenin tetiklenmesi her tahminde yapılmamaktadır. Erken güncellemenin tetiklenmesi için ardı ardına aynı tahmin değerlerinin elde edilmesi beklenmektedir. p-eHIP'de kullanılan ardı ardına üç adet tahminin aynı olması sonucu EU'nun tetiklenmesi gerçekleştirilmektedir. Bunun amacı ise MN için yanlış güncelleme kararlarının en aza indirgenmesidir.

Her bir tahmin için şartları sağlayan EU'lar, gerektiğinde yanlış tahmin durumunda veya herhangi bir sebeple zaman aşımı olduğu durumda iptal edilebilmektedir. Şekil 3.15 p-eHIP için akış şemasını göstermektedir.



Şekil 3.15: p-eHIP Akış Şeması

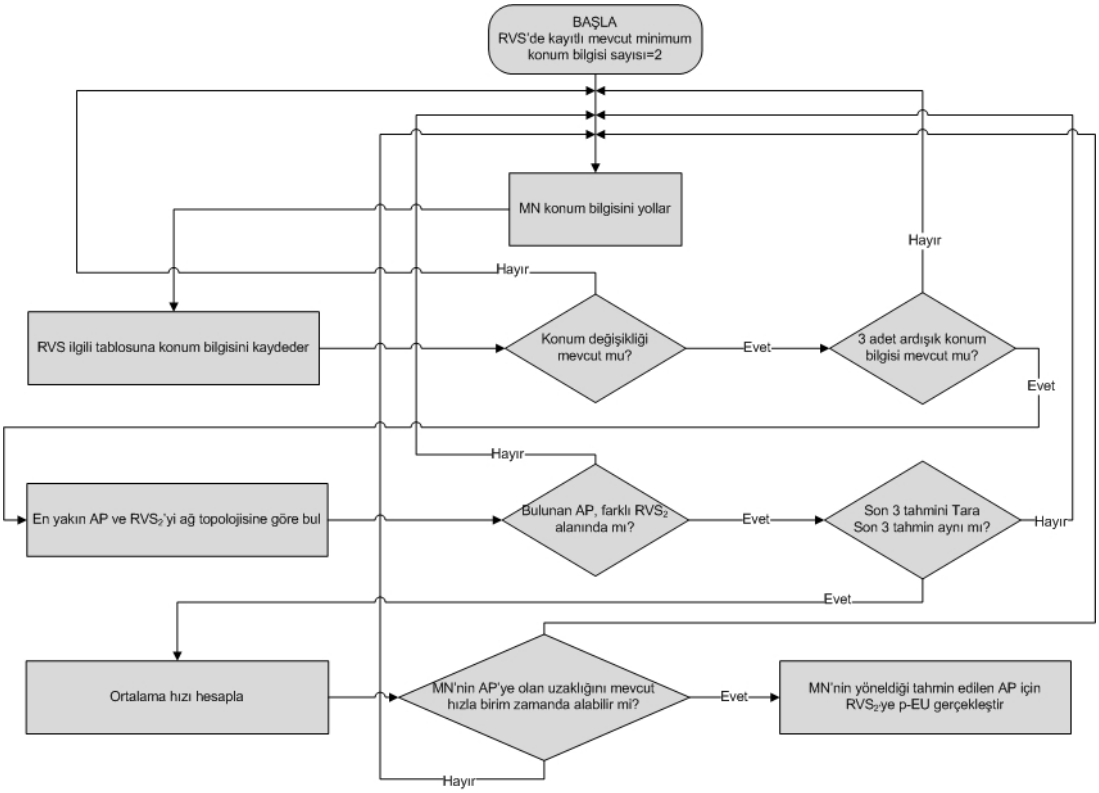
3.3.3. p-eHIP'de Hız Faktörüne Bağlı İyileştirme

p-eHIP'de yapılan tahmine dayalı erken güncellemeleri daha da iyileştirmek amacıyla mobil düğümün hızının bu yöntemdeki kararlara olan etkisi incelenmiştir. p-eHIP'e ilave olarak bu versiyonda yapılan kontrol, birim zamandaki ortalama hızın hesaplanması ve bu hıza göre MN'nin konumunun kendisi için tahminde bulunulan en yakın AP'ye olan uzaklığının bir arada değerlendirilmesi ile oluşmaktadır.

p-eHIP'de yapılan her tahmin, tüm ağ topolojisinde sabit olan AP kapsama alanın çapı ve merkezi göz önünde bulundurularak hesaplanan mesafelere dayanarak yapılmaktadır. Normal olarak eHIP'de uyguladığımız yer değiştirme senaryosuna göre bir MN, kendisinin hizmet aldığı AP ile farklı RVS_2 alt alanında bulunan bir AP'nin kapsama alanlarının kesişim noktasına giriş yaptığı anda EU tetiklenmektedir. Yönü ve hareketi düzenli olarak bir AP'ye yaklaşmakta olan bir MN için normal yer değiştirmeden önce arka arkaya p-EU kararları alınabilmektedir. Ancak; düzenli ve aynı AP'ye doğru hareket eden bir MN'nin kendisi için yapılan p-EU'lardan sonra dahi başka AP için tahmin yapılması mümkündür.

Bu tip durumları (yanlış p-EU) azaltmak amacıyla, tahminde bulunurken MN'nin ortalama hızına bağlı bir karar mekanizması önerilmektedir. Burada temel amaç, MN için çok erken zamanda p-EU kararlarını önlemektir. Böylece daha sonra değişebilecek tahminlerden önce yanlış p-EU kararlarının minimize edilmesi amaçlanmaktadır.

MN'nin bulunduğu konum ile o an kendisi için tahminde bulunulan AP arasındaki mesafe göz önünde bulundurulmaktadır. MN'nin bu mesafeyi, izlediği yol boyunca her adımında hesaplanan ortalama hızıyla bir sonraki hareketinde (konum değişikliğinde) katedip katetmeyeceği kontrol edilmektedir. Şekil 3.16, hız faktörünün göz önünde bulundurulduğu p-eHIP akış şemasını göstermektedir.



Şekil 3.16: Hız Faktörüne bağlı p-eHIP Akış Şeması

3.4. SERVİS KALİTESİ FARKINDALIKLI MOBİLİTE ALGORİTMASI

Bu bölümde önerdiğimiz mimariye uygulanabilecek, toplam radyo kaynak kullanımını (radio resource utilization-RRU), özellikle MN'lerin servis kalitesi ihtiyaçlarını ve gerçek zamanlı uygulamaların gecikmesini göz önünde bulunduran bir yöntem önermekteyiz. Bu yöntemde ağ mimarimizin kablosuz bir ağ yapısı olan mesh tipi ağ yapısında olduğunu varsaymaktayız. Yapılan incelemeler sonucunda, hareket eden MN'lerin devam eden gerçek zamanlı uygulamaları açısından önerilen algoritmanın RRU açısından kazanım sağladığı görülmektedir (Gurkas Aydin ve diğ., 2010).

3.4.1. Sistem Modellemesi ve Problemin Tanımlanması

eHIP yönteminde önerdiğimiz ağ mimarisini yönlendirilmiş bir graf $G(V,E)$, şeklinde ifade ediyoruz. Bu grafa bir bağlanabilirlik (connectivity) grafı adı verilir. Bu graftaki her u düğümü $u \in V = \{1,\ldots,N\}$ olmak üzere bir AP, RVS ve onları internete bağlayan tüm yönlendiricileri temsil edebilir. u ve her bir komşusu v arasında bir çift yönlü kablosuz hat mevcuttur ve $(u,v) \in E$ olan yönlendirilmiş kenarlar ile

gösterilmektedir. Graf bağlanabilirliği bağlılık matrisi ile ifade edilmektedir. Bağlılık matrisi $G(V, E)$, graf kenarları V ile etiketlenmiş satır ve sütunlardan oluşmaktadır. u ve v'nin doğrudan bağlı olup olmadığıyla ilgili olarak (u, v) pozisyonunda 0 veya 1 ile etiketlenmektedirler.

Önerdiğimiz modelde, en alt seviyede MN'lere hizmet veren AP'ler bulunmaktadır. Hareketlilik boyunca, bir MN bir etki alanından diğerine her geçtiğinde sistemi yeni konumu hakkında bilgilendirmek durumundadır. Bunu da ziyaret ettiği etki alanının RVS'si aracılığıyla kayıt güncelleme mesajı yollayarak gerçekleştirir.

Bir MN için RRU iki önemli kavramı içermektedir. Bunlardan ilki veri paketlerinin kaynak kullanımı ile ilgilidir. İkincisi ise kullanıcı hareketliliğini yönetmek için kullanılan sinyal mesajlarının kaynak kullanımı ile ilgilidir. Bu kavramlardan ilki veri teslim maliyeti (data delivery cost) ikincisini ise kayıt güncelleme maliyeti (registration update cost) olarak adlandırılmaktadır.

Bu problem şu şekilde formüle edilmektedir: N düğümden oluşan bir ağ için yer değiştirme boyunca o an aktif olan uygulamanın QoS sınırlandırmalarını sağlamak üzere, toplam radyo kaynak kullanımını minimize eden komşu düğümlerin bulunması gerekmektedir.

Bizim incelediğimiz durumda QoS kısıtlaması devam eden oturum için (VoIP uygulaması) söz konusu olan gecikmedir. Bu durumda RRU maliyeti (3.1)'deki gibi ifade edilebilmektedir (Langar ve diğ., 2009).

$$RRU\_Cost = \alpha.Reg\_Update\_Cost + \beta.Data\_Delivery\_Cost \qquad (3.1)$$

$$\alpha = \frac{2\mu\ m_{sig}}{2\mu\ m_{sig} + \lambda m_{data}} \qquad (3.2)$$

$$\beta = \frac{2\mu\ m_{data}}{2\mu\ m_{sig} + \lambda\ m_{data}} \qquad (3.3)$$

Bu ifadede α ve β, MN tarafından üretilen toplam trafikteki sinyal mesajlarının ve veri paketlerinin oranlarını göstermektedir. m_{sig} ve m_{data} sırasıyla kayıt güncelleme için kullanılan sinyal mesajlarını ve veri paketlerinin ortalama boyutlarını göstermektedir. Ayrıca $1/\mu$ MN'nin bir alt ağdaki ortalama kalış zamanını, λ ise downlink paket iletim zamanını paket/s cinsinden göstermektedir. Kullanılan semboller ve anlamları Tablo 3.3'te bir arada gösterilmektedir.

Tablo 3.3: Sistemin Modellenmesinde Kullanılan Semboller ve Anlamları

α	MN tarafından üretilen toplam trafikteki sinyal mesajlarının oranı
β	MN tarafından üretilen toplam trafikteki veri paketlerinin oranı
m_{sig}	Kayıt güncelleme için kullanılan sinyal mesajlarının ortalama boyutu
m_{data}	Kayıt güncelleme için kullanılan veri paketlerinin ortalama boyutu
$1/\mu$	MN'nin bir alt ağdaki ortalama kalış zamanı
λ	Downlink paket iletim zamanı (paket/s)
Π_i	MN'nin AP_i alt ağında bulunma olasılığı
N_l	Tüm ağdaki toplam yönlü bağlantı sayısı
D_{max}	Bir VoIP uygulaması için bir etki alanından diğerine geçerken tolere edilebilir maksimum gecikme
T_{start}	MN'nin ilk kayıt paketini yolladığı zaman
T_{end}	MN'nin ilk veri paketini aldığı zaman

Kayıt güncelleme maliyeti (Reg_Update_Cost) aşağıdaki gibi gösterilmektedir:

$$\text{Reg\_Update\_Cost} = \frac{1}{N_l} \text{x} \sum_{i=1}^{N} \Pi_i \text{ xUpdate\_Cost}(i) \qquad (3.3)$$

Bu ifadede Π_i, MN'nin AP_i alt ağında bulunma olasılığını, N_l ise tüm ağdaki toplam yönlü bağlantı sayısı göstermektedir. 3.3'te belirtilen ifadede bulunan Update_Cost şu şekilde gösterilmektedir:

$$\text{Update\_Cost}(i) = \sum_{j=1}^{N} (P(i,j) \text{ x} \sum_{k=1}^{N} \min(d(j,k) \text{ x } d(k, RVS_0))) \qquad (3.4)$$

Bu ifadede $P(i,j) = P(AP_i, AP_j)$ olasılığı AP_i'den AP_j'ye geçiş olasılığını ve $d(x,y)$ ise x ve y arasındaki uzaklığı sıçrama sayısı (number of hops) cinsinden göstermektedir.

Yine benzer şekilde veri teslim maliyeti (Data_Delivery_Cost) MN'nin AP_i'ye bağlı olduğundaki downlink trafiğin veri teslim maliyetidir ve şu şekilde ifade edilmektedir:

$$\text{Data\_Delivery\_Cost} = \frac{1}{N_l} \text{x} \sum_{i=1}^{N} \Pi_i \text{ xDelivery\_Cost } (i) \qquad (3.5)$$

Bu ifadede $Delivery\_Cost(i)$ M'nin AP_i'ye bağlı olduğu durumdaki downlink trafiğin teslim maliyetidir ve şu şekilde gösterilmektedir:

$$\text{Delivery\_Cost}(i) = \sum_{j=1}^{N}(P(i,j) \text{ x} \sum_{k=1}^{N} \min(d(j,k) \text{ x } d(k, RVS_0))) \qquad (3.6)$$

Tüm bu ifadelerden yola çıkarak tanımladığımız problemi aşağıdaki hedef fonksiyonu ile ifade etmemiz gerekmektedir.

$$d(AP_i, AP_j) + d(AP_j, RVS_0) < D_{max} \qquad (3.7)$$

ifadesine bağlı olarak

$$\min RRU \qquad (3.8)$$

şeklinde ifade edilmektedir.

Bu ifadedeki D_{max} bir VoIP uygulaması için bir etki alanından diğerine geçerken tolere edilebilir maksimum gecikmeyi göstermektedir ve $D_{max} = T_{end} + T_{start}$ ile ifade edilmektedir. Burada T_{start}, MN'nin ilk kayıt paketini yolladığı an ve T_{end}, MN'nin ilk veri paketini aldığı andır.

3.4.2. QoS Farkındalıklı Mobilite için Önerilen Algoritma

Bu bölümde MN'lerin mobilitesini, QoS kısıtlarına bağlı olarak hesaba katan bir algoritma önerilmektedir. Bu algoritma, MN yeni bir AP'ye her hareket ettiğinde çalışmaktadır. Bu algoritma öncelikle RVS_0'a doğrudan olmayan yolun kayıt maliyetini o anki RVS_i üzerinden hesaplar ve belirli bir eşik değeri olan *Thresh* ile karşılaştırır. Eğer hesaplanan maliyet eşik değerine eşit veya ondan daha düşükse, MN RVS_0'a olan o anki yolu tercih eder. Kayıt prosedüründen sonra veri iletim prosedürü yeni AP'de RVS_0'ın adresini aramak ile başlar. Eğer bulunursa AP RVS_0'a doğru

$$d(AP_i, RVS_i) + d(RVS_i, RVS_0) \text{ X} \leq D_{max} \qquad (3.9)$$

ve

$$Data\_Delivery\_Cost \leq Thresh\_data \qquad (3.10)$$

koşullarını sağlayan en kısa yolu arar. Eğer bu şart doğru ise yeni AP'ye doğru veri iletimi RVS_0'dan başlar. Aksi takdirde, AP, belirtilen koşulu sağlayan yeni bir kısa yol arayacaktır.

Bu kuralları ifade eden algoritmanın pseudo kod parçası şu şekildedir:

1: Eğer MN yeni bir etki alanına girerse
2: RVS_i'ye Reg_Update_Cost'u hesapla
3: RVS_i'nin IP adresini ve RVS_i ve RVS_0 arasındaki kayıt maliyetini içeren kayıt talebini RVS_i'ye yolla
4: Eğer $Data\_Delivery\_Cost \leq Thresh\_Reg$ ise
5: RVS_0 kaydını gerçekleştir
6: Yeni AP o anki RVS_0'ın IP adresini yönlendirme tablosunda olup olmadığını kontrol eder
7: Eğer adres varsa
8: Yeni AP RVS_0'a olan en kısa yolu bulur
9: RVS_0'a Veri İletim Maliyetini hesaplar
10: Eğer $d(AP_i, RVS_i) + d(RVS_i, RVS_0) \text{ X} \leq D_{max}$ ve $Data\_Delivery\_Cost \leq Thresh\_data$ ise
11: Veri teslimi için bu yol seçilir
12: Veri teslimi prosedürü gerçekleştirilir.
13: Diğer durumlarda
14: Daha düşük maliyetli bir yol aranır

3.5. SİMÜLASYON VE ANALİZ

eHIP yönteminin test edilmesi için kullanılan OMNET++ iletişim ağlarını, IT sistemlerini, kuyruk sistemleri ve donanım mimarilerini modellemek için kullanılan ayrık olay tabanlı bir simülasyon ortamıdır (OMNET++, 2011). OMNET++ açık kaynak kodlu, ticari amaç gütmeyen kullanıma uygun bir yazılımdır. Network Simulator (NS) (Ns-2, 2011) gibi açık kaynak ve araştırma odaklı simülatörler ile OPNET (Opnet, 2011) gibi yüksek maliyetli ticari yazılımların arasında yer alan bir çözüm sunmaktadır.

OMNET++, bileşenlerden oluşmaktadır ve modüler bir yapıdadır. Bir simülasyon modeli birbirleriyle haberleşen modüllerden oluşmaktadır.. Bu modüller basit modüller olarak adlandırılmaktadır. C++ diliyle yazılmışlardır ve OMNET++ simülasyon kütüphanesini kullanmaktadırlar. Bu basit modüllerin birleşmesiyle oluşan modüllere de birleşik modüller adı verilmektedir. Topoloji tanımlamalarını gerçekleştirmek için NED (NEtwork Description) adı verilen bir tanımlama dili kullanılmaktadır. Bir .ned dosyasında, basit modül tanımlamaları, birleşik modül tanımlamaları ve ağ tanımlamaları gibi bilgiler bulunmaktadır. Bu yönüyle modellerin davranışlarının tanımlanması ve model topolojisi birbirinden ayrılmaktadır. C++ kodu model davranışlarını tanımlarken, NED dili ile ağ topolojisi tanımlanmaktadır. Simülasyon parametreleri ise C++ NED kodlarından bağımsız olarak INI dosyalarından tanımlanmaktadır.

OMNET++ için bir çok simülasyon modeli, araştırma grupları ve kişiler tarafından geliştirilmektedir. Haberleşme ağları için günümüzde en geçerli ve en yaygın olan model grubu INET Framework'tür (Inet, 2011). INET FW, IPv4, IPv6, TCP, UDP, MPLS, RSVP gibi birçok protokolü, telnet, video akışı gibi birçok uygulamayı ve PPP, Ethernet, 802.11b/g gibi birçok bağlantı katmanı modellerini desteklemektedir.

p-eHIP eklentisinin algoritmik olarak test edilmesi için MATLAB (MATrix LABoratary) simülasyon aracı kullanılmıştır. MATLAB (Matlab, 2011), temel olarak nümerik hesaplama, grafiksel veri gösterimi ve programlamayı içeren bir yazılımdır. Kapsamlı ve hızlı nümerik hesaplama yöntemleri ile hızlıca sonuç alınması mümkün

olmaktadır. p-eHIP yönteminin test edilmesindeki amacımız doğru tahmin oranı ile ilgili nümerik değerler elde etmek olduğu için bu algoritmanın test edilmesinde MATLAB ortamı kullanılmıştır.

Bölüm 3.4'e verilen sistemin modellenmesi ve QoS farkındalıklı algoritma çalışması ise yurtdışında sürdürülen araştırmalarda OPNET simülasyon aracı ile test edilmiştir. OPNET öncelikle ticari amaçlarla kullanılmakta olan bir yazılımdır. Her türlü ağ teknolojisi için kullanılmak üzere protokol modellerini desteklemektedir. Akademik ve araştırmaya yönelik olarak da kullanılabilmektedir. Ağ simülasyonları, yüksek seviyede görsel olarak tanımlanabilmektedir. OPNET'te ağ modellerin geliştirilmesi dört adet hiyerarşik editörün sayesinden yapılmaktadır. Bunlar Proje Editörü, Proses Editörü, Düğüm Editörü ve Kod Editörüdür. Proje Editöründe test edilmesi istenen ağ ortamı tanımlanmakta, Düğüm Editöründe tanımlanan ağ topolojisindeki her bir düğümün genel yapısı tanımlanmakta, Proses Editöründe düğümlerin altındaki her bir prosesin çalışma mekanizması sonlu durum makineleri ile tanımlanmakta ve Kod Editöründe de bu sonlu durum makineleri için ilgili kodlama işlemleri yapılmaktadır.

3.5.1. HIPSIM++

HIPSIM++, INET FW üzerine geliştirilmiş ve HIP protokolünün temel eklentilerini ve özelliklerini gerçekleştiren bir modeldir (HIPSIM++, 2011). HIPSIM++'in temel fonksiyonu, HIP'in çekirdek fonksiyonlarını, mobilite desteğini ve kablosuz davranışını modellemektir. Bu yüzden IPSec ve ona dayanan ilgili algoritmaları gerçeklememektedir. Diffie-Hellman mekanizmaları, RSA yöntemi, kriptografik özet fonksiyonları ve bulmacalar gibi güvenlik algoritmalarını barındırmamaktadır (Bokor ve diğ.2009a – Bokor ve diğ., 2009b).

3.5.1.1. HIPSIM++'in Temel Modülleri

i. HIP Modülü

HIPSIM++'in çekirdek ve HIP katmanını modelleyen modüldür. Her bir yeni HIP oturumu için HIPSM adlı bir geri plan yordamı (daemon) yaratmaktadır. Bu yordam, HIP durum makinesinin (HIPSM) tüm fonksiyonlarından (BE ve mobilite fonksiyonları gibi) sorumludur.

ii. HIPSM Modülü

HIP durum makinesinin ana fonksiyonlarını gerçekleştiren modüldür. HIPSM, paketlerin başarıyla doğrulandığını ve işlendiğini kabul etmektedir. Bir HPSM'nin varlığı, tek bir HIP bağlantısının ve HIP güvenlik bağlantısını (SA) temsil etmekte ve yönetmektedir. HIPSM, BE, RVS kaydı, UPDATE mekanizmaları gibi işlemleri gerçekleştirir ve durum geçişlerini esnasında HIP mesajlarını üretmektedir.

iii. RVSHIP Modülü

RVS fonksiyonlarını destekleyen HIP modüldür. Bu modülde, gelen kayıt mesajları işlenmekte ve I1 mesajlarının uygun HIP-R'ye iletilmesi gerçekleştirilir.

iv. DNSBase Modülü

Temel DNS sunucusu fonksiyonlarının HIP hostlarının isim çözümlemesi ve yeni kaynak kaydı yapılmasını gerçekleştiren basit bir UDP uygulamasıdır. Bu modül, alan adlarını HIT'lere ve IP adreslerine dönüştürmektedir.

3.5.1.2. HIP Düğümleri

i. Kablolu HIP I/R Düğümü (HipHost6)

Bir I ve R hostunun fonksiyonlarını tanımlayan ve INET'in StandardHost6 birleşik modülünden türetilmiştir. İletim ve ağ katmanlarının arasına HIP modülü entegre edilmiştir. Temel bir HIP hostunu ve mekanizmalarını, HIP tabanlı UDP/TCP uygulamalarını mobilite desteği olmadan temsil etmektedir.

ii. Kablosuz HIP I/R (WirelessHipHost6)

Normal HIP hostunun fiziksel arayüzü WLAN olacak şekilde düzenlenmiş halidir. Mobilite operasyonlarını da desteklemektedir.

iii. Çoklu Arayüzlü Kablosuz HIP I/R (WirelessMultihomeHipHost6)

HIP'in çoklu konumluluk özelliklerini desteklemek üzere birden çok fiziksel arayüz bir HIP hostuna eklenmektedir. Bu düğüm tipi, çoklu konumluluğunu destekleyen HIP hostlarını temsil etmektedir.

iv. DNS Sunucusu (StandardHost6 with DNSServer)

Bir DNS sunucusu düğümü, HIP hostlarına isim çözümleme hizmetini yerine getirmekle yükümlüdür. HIPSIM++'da bir simülasyon senaryosunda en az bir adet DNS sunucusuna düğümüne ihtiyaç duyulmaktadır.

v. HIP Randevu Sunucusu (RvsHost6)

Bu düğüm HIP'in randevu sunucusu fonksiyonlarını gerçekleştirmektedir. Kablolu veya kablosuz I düğümlerine gelen I1 mesajlarını uygun ve RVS'de kayıtlı R düğümlerine iletir. Kablosuz HIP düğümleri sürekli olarak konum değişikliklerinden RVS'lerini haberdar etmek durumundadır.

3.5.2. eHIP Simülasyonu

3.5.2.1. eHIP Modülleri

i. HIPEU Modülü

HIPSIM++'in HIP modülünün eHIP fonksiyonlarını ve hiyerarşik yapıyı destekleyecek şekilde genişletilmiş versiyonudur. HIP modülünün fonksiyonlarına ek olarak, RA paketinden RVS bilgisi alma, EU, FU tipinde mesajların işlenişi için ek fonksiyonlar ve durum geçişlerinden sorumlu modüldür.

ii. HIPEUFSM Modülü

eHIP durum makinesinin ana fonksiyonlarını gerçekleştiren modüldür. HIPSM'ye ek olarak, erken güncelleme durumundaki durum değişikliğini ve CN için EU işleminin sonunda RVS_2'den gelecek güncelleme mesajının işlenmesinden sorumludur.

iii. HIPEURVS Modülü

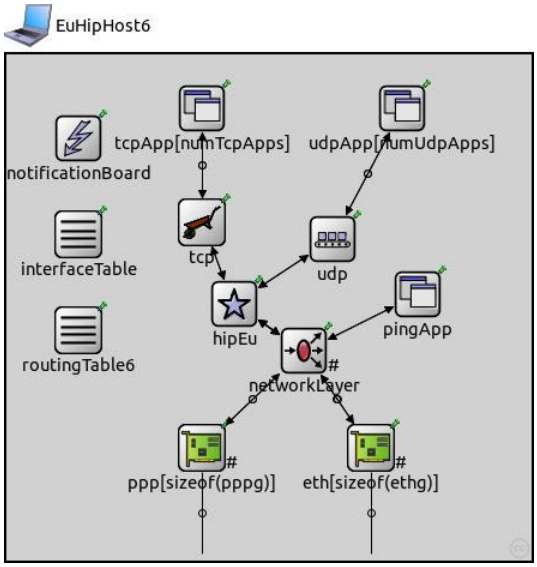
Hiyerarşik yapıda RVS fonksiyonlarını destekleyen HIP modülüdür. Klasik HIP mantığından farklı olarak, seviye bilgisi tutulmaktadır. Bu modülde, en alt seviye RVS'ler HIP hostlarla iletişimden sorumluyken, üst seviye RVS'ler sadece hiyerarşi içerisinde alt seviyesinden RVS'ler ile ve ebeveyn RVS ile haberleşmektedirler. En alt seviye RVS HIP hostunun kaydını yapmanın yanında, HIP hostundan gelen ve hedefi belli olmayan HIP mesajlarını (I1) üst seviyeye iletir. Buna ek olarak, en alt seviye RVS, yeni bir host kaydettiğinde bunu NEW_HOST_REG mesajı ile ebeveyn RVS'ye bildirir. Üst seviye RVS'ler, hostların ürettiği I1 mesajlarını alt seviye RVS'lerden alıp, ya üst seviyeye iletir ya da HIP-R'yi bulduğu alt seviye RVS'ye iletir. Bir host için RVS değişikliği söz konusu olduğunda, üst seviye RVS'ler farklı bir alt alanda aynı host için kayıt tutuyorsa, bu kaydın hem günceller hem de

DELETE_HOST_REG mesajı ile eski alt alandaki RVS'lerdeki ilgili host kaydının silinmesini sağlar.

3.5.2.2. eHIP Düğümleri

i. Kablolu eHIP I/R Düğümü (EUHipHost6)

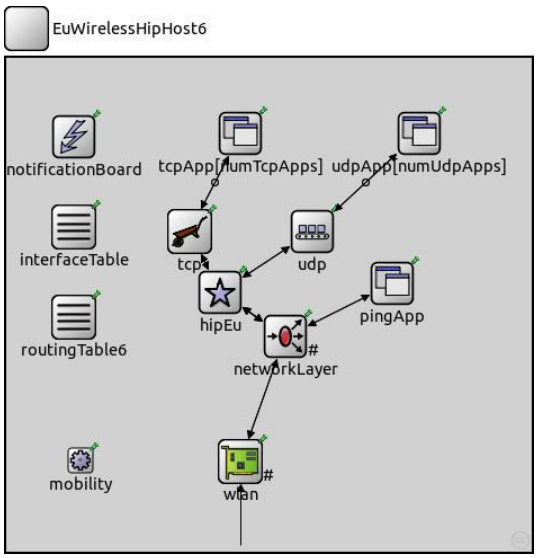
Bir I ve R hostunun fonksiyonlarını tanımlayan ve INET'in StandardHost6 birleşik modülünden türetilmiştir. eHIP fonksiyonlarını destekleyen bir hostu ve mekanizmalarını, HIP tabanlı UDP/TCP uygulamalarını mobilite desteği olmadan temsil etmektedir.



Şekil 3.17: EUHipHost6 düğümünün NED gösterimi

ii. Kablosuz HIP I/R (EUWirelessHipHost6)

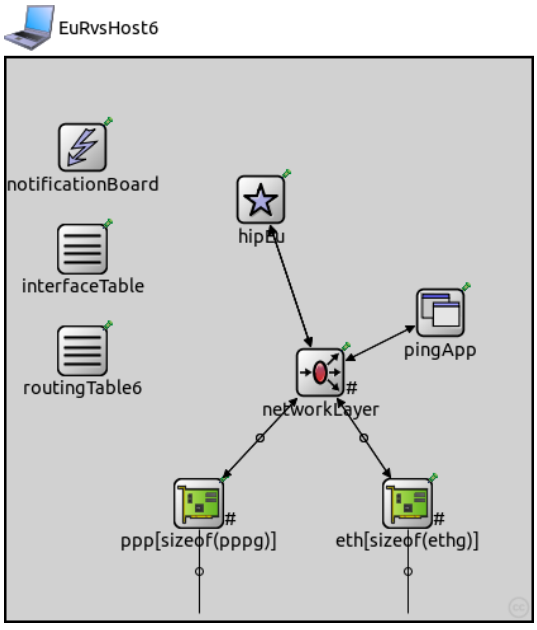
Bir I ve R hostunun fonksiyonlarını tanımlayan ve INET'in WirelessHost6 birleşik modülünden türetilmiştir. eHIP fonksiyonlarını destekleyen bir hostu ve mekanizmalarını, HIP tabanlı UDP/TCP uygulamalarını mobilite desteği ile desteklemektedir. Normal HIP hostunun fiziksel arayüzü WLAN olacak şekilde düzenlenmiş halidir.



Şekil 3.18: EUWirelessHipHost6 düğümünün NED gösterimi

iii. <u>HIP Randevu Sunucusu (EURvsHost6)</u>

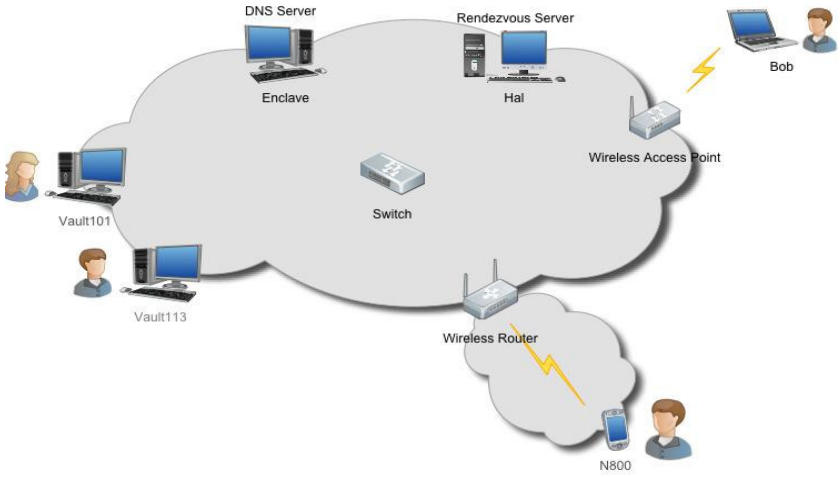
Bu düğüm, eHIP'de kullanılan hiyerarşik RVS yapısına uygun olarak randevu sunucusu fonksiyonlarını gerçekleştirmektedir. Ağdaki tüm seviye RVS'ler, bu düğüm tipinde tanımlanabilmektedir ve tanımlandıkları RVS seviyesine bağlı olarak davranış sergilemektedirler.



Şekil 3.19: EURvsHost6 düğümünün NED gösterimi

3.6. HIP TEST ORTAMI VE GERÇEKLENMESİ

HIP ile ilgili BE ve hareketlilik durumunda güncelleme durumunu performans testlerini gerçekleştirebilmek için gerçek bir test ortamı oluşturulmuştur. Bu test ortamında sabit HIP düğümleri, DNS ve RVS sunucusu olarak masaüstü bilgisayarlar, mobil öğeler olarak da bir dizüstü bilgisayar ve bir tablet bilgisayar kullanılmıştır. Bu öğeler bir adet kablosuz yönlendirici, bir adet kablosuz erişim noktası ve bir adet bluetooth erişim noktası kullanılarak bir test ağı oluşturulmuştur. Şekil 3.20 test ortamının yerleşimini, Tablo 3.4 ise kullanılan ağ elemanlarının detaylarını göstermektedir.



Şekil 3.20: HIP Test Ortamı

Tablo 3.4: HIP Test Ortamında kullanılan cihazlar ve detayları

Test Ortamındaki Rolü	Host Adı	Detay
Kablosuz Yönlendirici	HTBR	NETGEAR KWGR614
Kablosuz Erişim Noktası	HTBAP1	Cisco Aironet 1100
	HTBAP2	ANYCOM EDR-AP
	HTBAP3	
HIP Düğümü	N800	Nokia Internet Tablet N800
	BOB	DELL Latitude D830
	VAULT101	DELL Precision T3400
	VAULT113	
HIP Randevu Sunucusu	HAL	DELL Precision 380
DNS Sunucusu	ENCLAVE	DELL Precision T3400

Konfigüre edilen test ortamında birçok test yapılmıştır ancak bu testlerim temel amacı HIP protokolünün HIP implementasyonu üzerinden doğrulanması ve onaylanmasıdır. Temel HIPL ortamı sağlandıktan sonra HIP'in mobilite yönetimi ile ilgili özelliklerinin analizi ve tasarımı gerçekleştirilmiştir. Test ortamı kurulduktan ve gerekli ayarlamalar

yapıldıktan sonra infraHIP projesinin en son HIP implementasyonu üzerinden testler gerçekleştirilmiştir (InfraHip, 2009).

Testlerin gerçekleştirilmesi için iki ana senaryo tasarlanmıştır:

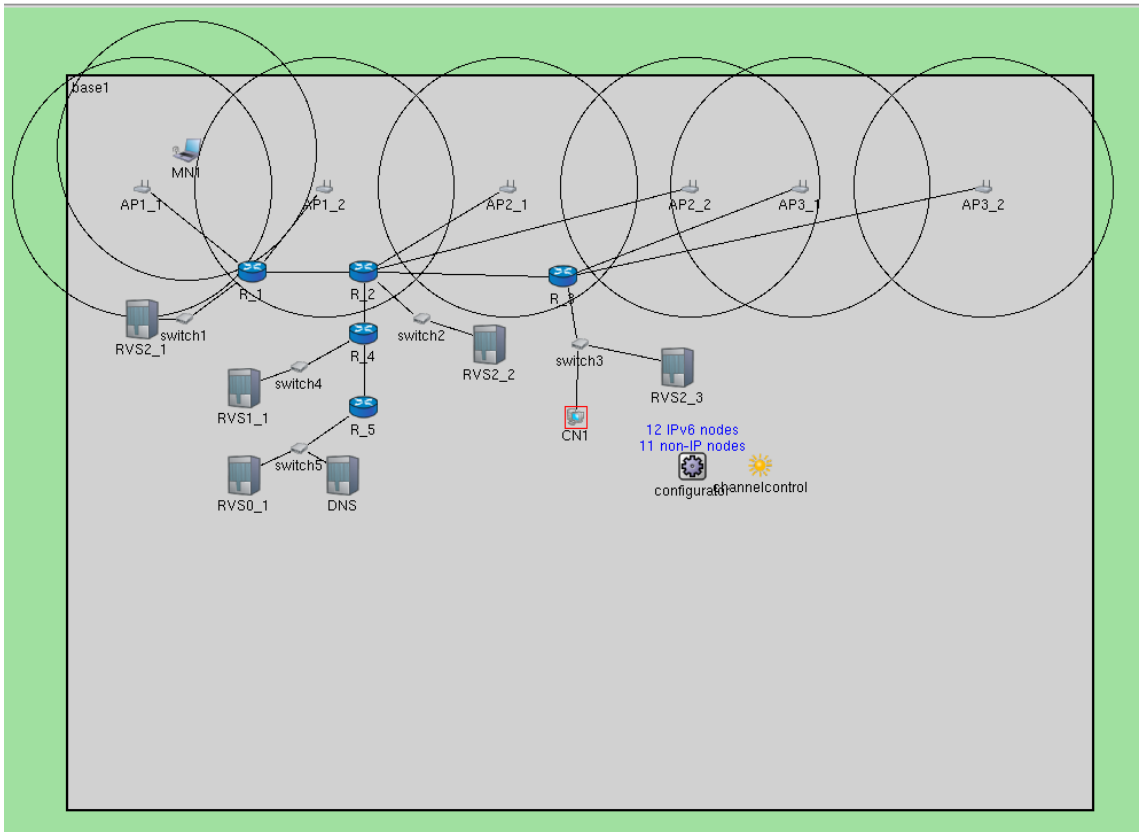
- ***Senaryo 1:*** İlk senaryoda her biri başlatan host (initiator-I) ve cevap veren host (responder-R) olarak görev yapabilen iki adet sabit düğüm (Bob ve Hal) bulunmaktadır. İki düğüm de özel bir LAN üzerinde aynı yönlendiriciye Ethernet üzerinden bağlanmıştır.
- ***Senaryo 2:*** İkinci senaryoda kablosuz yönlendiriciye Ethernet ile bağlı bir sabit düğüm (Hal) ve aynı özel ağa bir kablosuz yönlendirici, bir bluetooth erişim noktası veya kablosuz erişim noktası üzerinden bağlı bir mobil düğüm (N800) bulunmaktadır. Sabit düğüm, mobil düğüm tarafından yollanan farklı mesajlara karşı cevap veren düğüm olarak görev almakta, mobil düğüm ise BE prosedüründe ve mobilite prosedürlerinde başlatan düğüm olarak görev almaktadır.

4. BULGULAR

4.1. EHIP YÖNTEMİ İÇİN PERFORMANS İNCELEMESİ

4.1.1. Ağ Topolojileri ve Senaryolar

Simülasyon sonuçları üç farklı senaryo üzerinde incelenmiştir. Bunlar klasik HIP mimarisi (HIP) (bölüm 2.2), erken güncelleme yöntemini barındırmayan hiyerarşik HIP (Hiyerarşik HIP) mimarisi ve erken güncelleme yöntemini barındıran (eHIP) mimarisidir (bölüm 3.2).



Şekil 4.1: Simülasyonlarda kullanılan ağ topolojisi

HIP senaryosunda ağda tek bir RVS bulunmaktadır. Hiyerarşik HIP ve eHIP senaryolarında Şekil 4.1'de gösterildiği gibi RVS_1 ve RVS_2'lerin hiyerarşisi

kullanılmıştır. Bunların dışında ağ elemanları olarak Ethernet arayüzüne sahip IPV6 yönlendiriciler, anahtarlar (switch) ve erişim noktaları kullanılmıştır.

4.1.2. Simülasyon Parametreleri

OMNET++ Simülasyonlarında kullanılan parametreler INI dosyaları aracılığıyla tanımlanmaktadır. Kullanılan örnek INI dosyaları EK-A'da detaylı olarak gösterilmektedir.

INI dosyasında ağdaki RVS'lerin, MN'nin ve CN'nin HIT bilgileri tanımlanmaktadır. Klasik HIP senaryosunda ağda tek bir RVS bulunmaktadır. Hiyerarşik HIP ve eHIP senaryolarında ise ağa yerleştirilen diğer RVS'lerin HIT tanımlamaları da yapılmaktadır.

MN'nin mobilite esnasından işleyeceği model "Rectangle Mobility" olarak tanımlanmıştır ve tüm senaryolarda ortak olarak kullanılmaktadır. Bu mobilite modelinin seçilmesinin sebebi, senaryoların birbirine yakın davranış sergilemesini sağlamaktır. Bu modelde MN simülasyon boyunca tanımlanan topoloji boyunca dikdörtgensel yapıda bir yol izlemektedir. Bulunduğu konumdan ağın diğer ucuna hareket edip, ağın sınırlarının sonuna geldiğinde yine aynı konuma geri dönmektedir. Bu yolu simülasyon süresi boyunca izlemektedir. Bunların dışında AP'lerin fiziksel özellikleri ve mac adresleri tanımlanmaktadır.

Simülasyonlar üç farklı ağ yükü ve MN'nin beş farklı hızla hareket etmesi üzerinden incelenmektedir. Load1 olarak tanımlanan ağ yükünde ağa verilen trafik 2,5 MBps, Load2 olarak tanımlanan ağ yükünde ağa verilen trafik 5 MBps ve Load3 olarak tanımlanan ağ yükünde ağa verilen trafik 10 MBps olarak belirlenmiştir. Simülasyon zamanı 10000 s olarak belirlenmiş, her bir senaryo ve konfigürasyon için çok sayıda simülasyon koşturularak çıkan sonuçların % 95 güvenilirlik aralığı ile ortalama değerleri alınmıştır.

Ağda MN tarafından üretilen iki farklı türde trafik tipi kullanılmıştır. Bunlar TCP ve UDP tipi trafiktir. TCP için, MN tarafında TCPSessionApp uygulaması ve CN tarafında TCPSinkApp uygulaması kullanımıştır. TCPSessionApp uygulamasında kullanılan

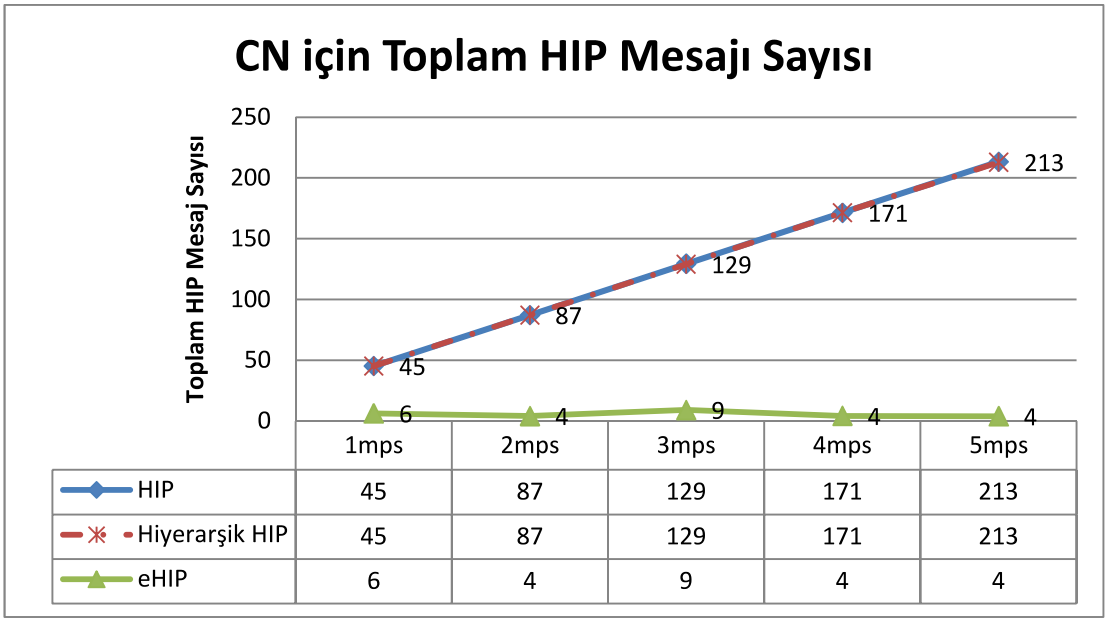
paket boyutu değişken olarak ayarlanmıştır. UDP için ise MN tarafında UDPEchoStream uygulaması ve CN tarafında UDPEchoApp uygulaması kullanılmıştır. UDPEchoStream uygulamasında kullanılan paket boyutu 256 B olarak ayarlanmıştır.

4.1.3. Simülasyon Sonuçları

4.1.3.1. Toplam HIP Mesaj Sayıları

Hiyerarşik yapının ve eHIP yönteminin, ağda işlenmesi gereken toplam HIP mesajı sayısı açısından klasik HIP'e göre getirdiği avantaj ve dezavantajlar incelenmiştir. Hiyerarşik HIP ile eHIP yönteminin farklı en alt seviye RVS'ler ile ve CN'ler ile mesajlaşılması aşamasında oluşmaktadır. Hiyerarşik HIP'de EU mekanizması yer almadığından dolayı, her RVS değişiminde BE prosedürü gerçekleştirilmektedir. Bu BE prosedürü, MN'nin RVS ile olan HO süresini arttırdığı gibi, ikili arasındaki HIP mesaj sayısını da doğal olarak arttırmaktadır. Hiyerarşik HIP'de, aynı H2 içinde AP'ler arası yer değiştirme olduğunda klasik yöntemdeki güncelleme prosedürü ile RVS_2, MN ile ilgili bilgilerini günceller. eHIP'de ise EU mesajı ile bu güncelleme işlemi erken zamanda gerçekleştirilir ve bu sayede hem RVS ile olan mesajlaşma azaltıldığı gibi CN'ler ile gereken güncelleme işlemleri RVS'ye yönlendirilmiş olmaktadır.

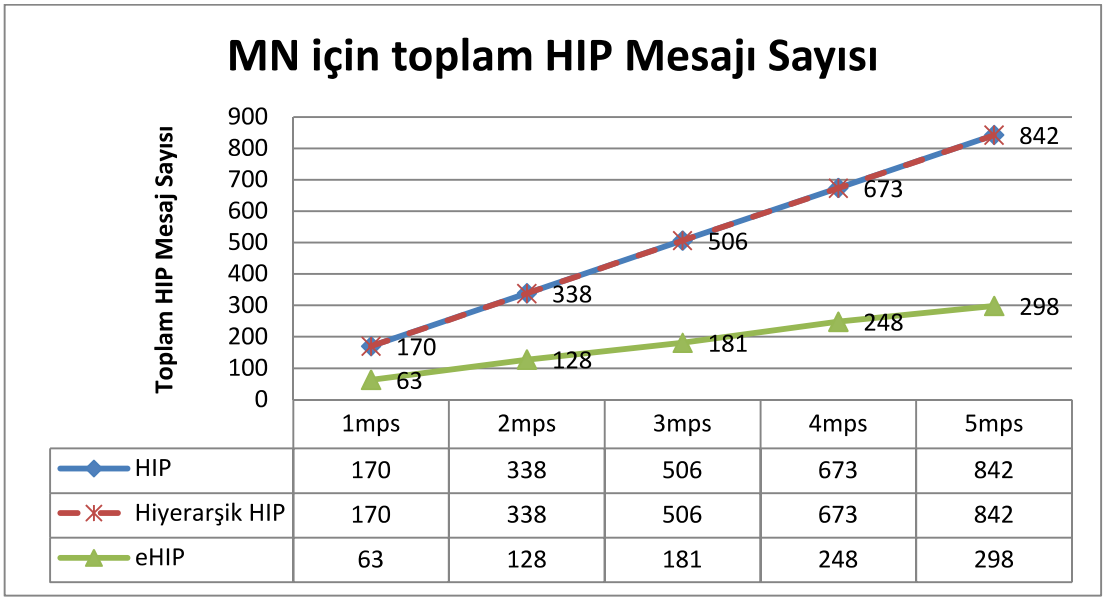
Şekil 4.2 'de her üç yöntem için CN tarafından üretilen toplam HIP mesajı sayısı gösterilmektedir. Grafikte klasik HIP ve Hiyerarşik HIP yöntemlerinde CN tarafından üretilen toplam mesaj sayısının aynı olduğu görülmektedir. Bunun nedeni her iki yöntemde de, CN'nin her yer değiştirmede MN'nin klasik güncelleme prosedürünü (üç adımlı) gerçekleştirmesi, erken güncelleme yapmamasıdır. eHIP'de ise, erken güncelleme işlemi gerçekleştiğinde, CN herhangi bir HIP mesajı üretmemektedir. MN'nin yeni konum bilgisinin CN'de güncellenmesi işlemi MN'nin yeni hareket ettiği RVS_2'den gelen FU1 mesajları ile gerçekleştirilmektedir.



Şekil 4.2: CN'de Üretilen Toplam HIP Mesajı Sayısı

Şekil 4.3 'te her üç yöntem için MN tarafından üretilen toplam HIP mesajı sayısı gösterilmektedir. Grafikte klasik HIP ve Hiyerarşik HIP yöntemlerinde MN tarafından üretilen toplam mesaj sayısının aynı olduğu görülmektedir. MN, klasik HIP'te her yer değiştirme esnasında bağlı bulunduğu RVS ve CN'lere güncelleme işlemi için UPDATE mesajı göndermektedir. Hiyerarşik HIP'te ise aynı H2 içinde AP'ler arası yer değiştirme esnasında UPDATE mesajı göndermektedir. H2 yer değiştirme olduğunda ise CN'ler ile UPDATE, RVS ile BE prosedürlerini gerçekleştirmektedir. Tüm bu güncelleme işlemlerinde MN her bir hedefe iki adet HIP paketi (UPDATE 1 ve UPDATE 3 veya I1 ve I2 mesajları) göndermektedir. Bu sebeple mesaj sayıları birbiriyle örtüşmektedir. eHIP yönteminde, aynı H2 içindeki AP'ler arasındaki yer değiştirme esnasında MN'den sadece EU mesajı gönderilmekte ve RVS_2'den direkt olarak EU3(ok) mesajı gönderilmektedir. Daha sonra yeni IP bilgisinin tamamlanması ile FU mesajı gönderilmektedir. EU1 ve EU2 mesajları başka RVS_2 ile haberleşilmediğinden dolayı kullanılmamaktadır. H2 yer değiştirme durumunda ise MN tarafından RVS'ye bir adet EU ve bir adet FU mesajı gönderilmektedir.eHIP yönteminde yer alan tüm mesaj akışı tamamlanmaktadır. H2 yerdeğiştirme veya H2 içinde yer değiştirme esnasında, eHIP yönteminde MN, CN'lere HIP mesajı göndermemektedir. Gerçeklenen simülasyonlarda sadece bir adet MN-CN haberleşmesi gerçeklenmiştir. MN'nin konuştuğu CN sayısının daha fazla olduğu bir senaryoda, eHIP

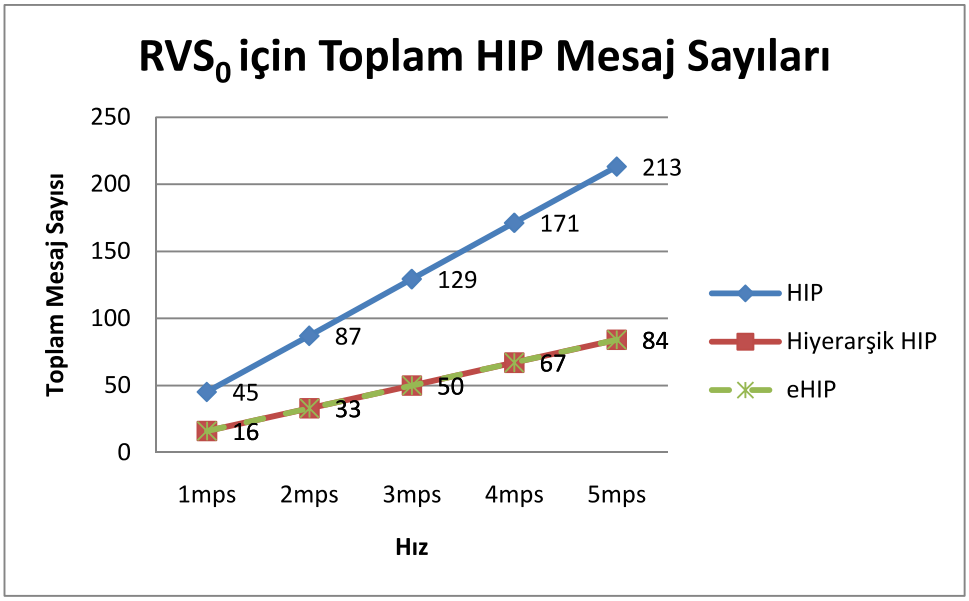
yönteminde MN tarafından gönderilen toplam HIP mesajı sayısı diğer yöntemlere göre oldukça düşük miktarda olacaktır. Bunun sebebi, HIP veya hiyerarşik HIP'te, her bir CN ile güncelleme işlemlerinin ayrı ayrı MN tarafından yapılmasının gerekmesidir. Bu da çoklu CN haberleşmesi olan durumda HIP mesaj sayısını arttırmaktadır. HIP mesaj sayısının fazla miktarda artışı ise, MN'nin aktif haberleşmeye dönme süresini de geciktirecektir. eHIP'de CN'lerin güncellenmesi işlemi MN'lere göre daha güçlü ve yüksek ağ çıkışlı olan RVS'lere devredilmektedir.



	1mps	2mps	3mps	4mps	5mps
HIP	170	338	506	673	842
Hiyerarşik HIP	170	338	506	673	842
eHIP	63	128	181	248	298

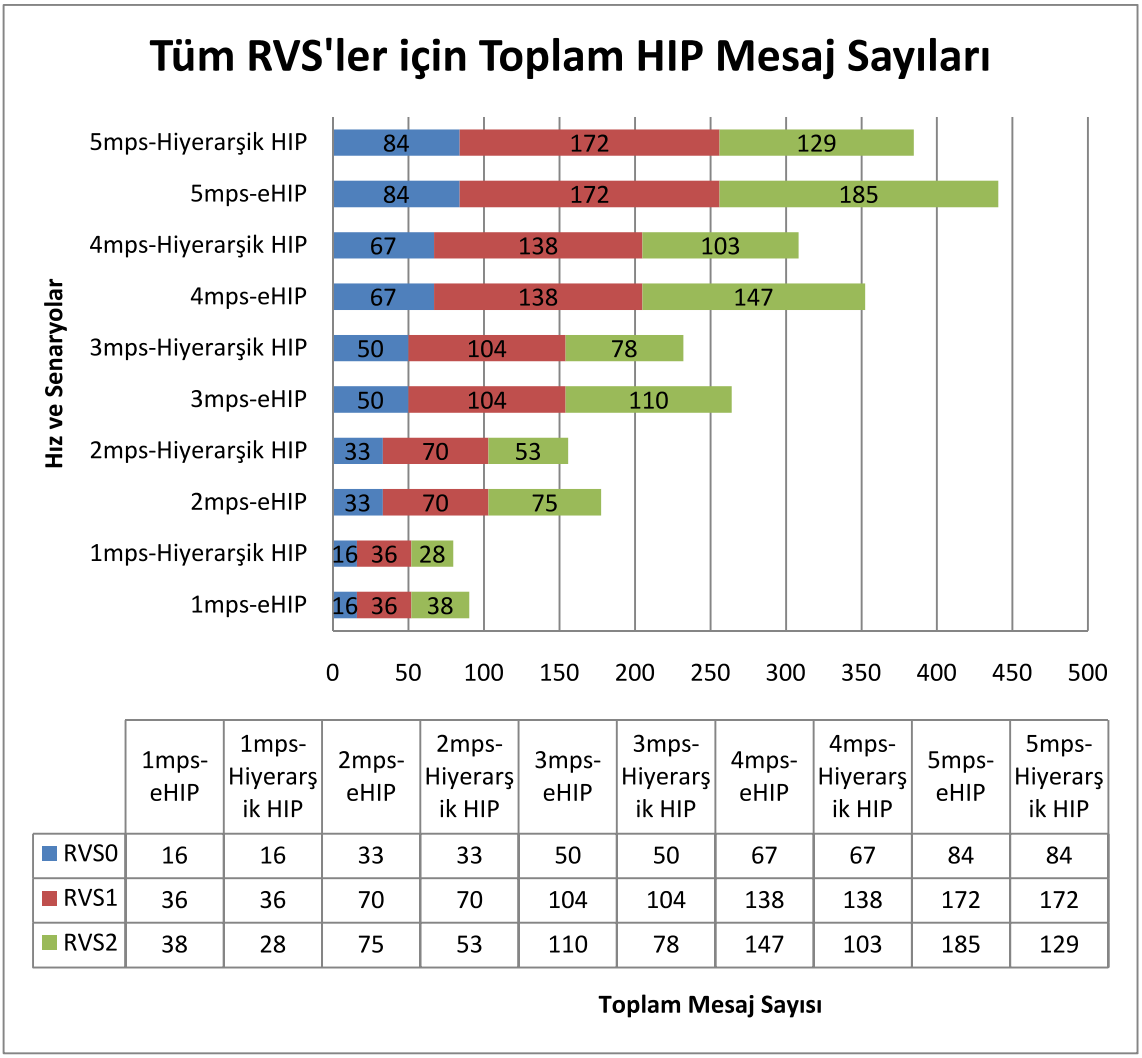
Şekil 4.3: MN'de Üretilen Toplam HIP Mesajı Sayısı

Şekil 4.4'te her üç yöntemde ağın en üst seviyesinde bulunan RVS_0 tarafından gönderilen toplam HIP mesajlarının sayısı gösterilmektedir. Bu grafikte Hiyerarşik HIp ve eHIP'in klasik HIP'e göre en üst seviye RVS bazında getirdiği daha az HIP mesajı avantajı görülmektedir. Klasik HIP'te, ağda tek bir RVS bulunurken, hiyerarşik HIP ve eHIP'te, mesajların işlenme yükü diğer alt seviye RVS'lere bölüştürülmüştür. Klasik HIP'de RVS'lerin hiyerarşisi bulunmadığından dolayı tüm RVS haberleşmeleri RVS_0 seviyesinde olmaktadır. Hiyerarşik HIP ve eHIP'te ise özellikle yer değiştirme esnasında gönderilen mesajlar alt seviye RVS'ler (RVS_1 ve RVS_2) tarafından işlenmekte ve cevaplanmaktadır.



Şekil 4.4: RVS_0 Tarafından Üretilen Toplam HIP Mesajı Sayısı

Şekil 4.5'te, tüm seviye RVS'ler için toplam HIP mesaj sayıları Hiyerarşik HIP ve eHIP yöntemlerinde gösterilmektedir. Hiyerarşik HIP ve eHIP arasında RVS_2'ler tarafından gönderilen mesaj sayısındaki farklılık, CN'lerin güncellenme görevinin eHIP'de en alt seviye RVS'lerin üzerinden olmasıdır. Diğer yöntemlerde bu işlemin yükü MN üzerinde olduğu için, eHIP'de RVS_2 seviyesinde gönderilen HIP mesajlarının sayısı daha fazla çıkmaktadır.



	1mps-eHIP	1mps-Hiyerarşik HIP	2mps-eHIP	2mps-Hiyerarşik HIP	3mps-eHIP	3mps-Hiyerarşik HIP	4mps-eHIP	4mps-Hiyerarşik HIP	5mps-eHIP	5mps-Hiyerarşik HIP
RVS0	16	16	33	33	50	50	67	67	84	84
RVS1	36	36	70	70	104	104	138	138	172	172
RVS2	38	28	75	53	110	78	147	103	185	129

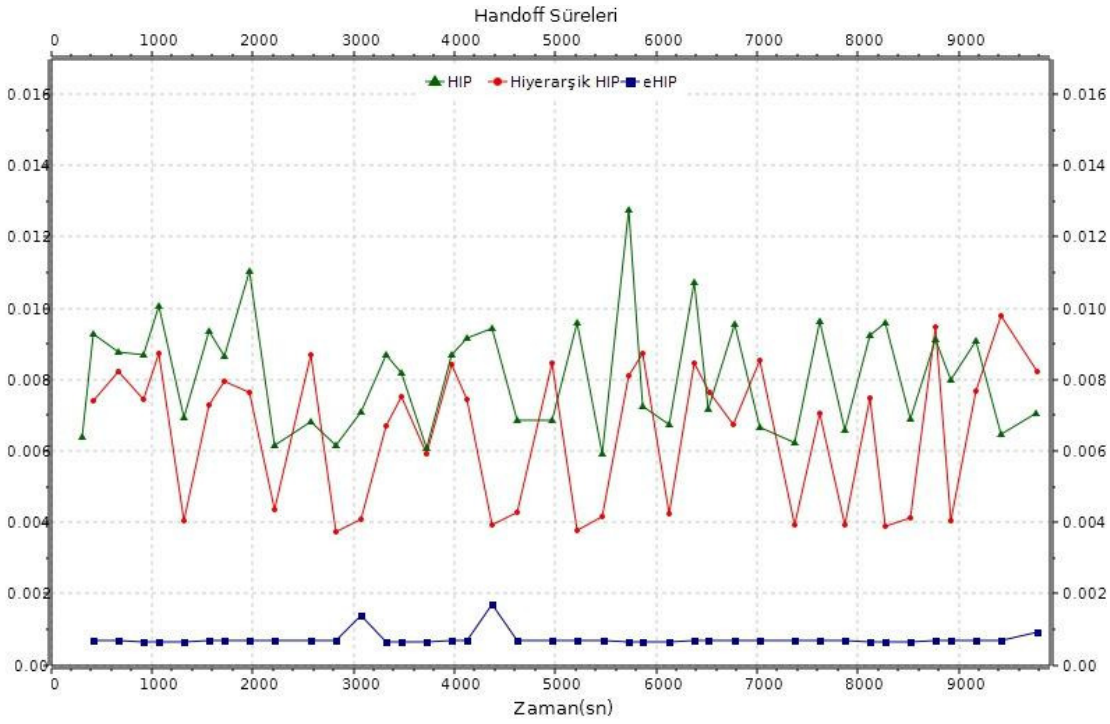
Şekil 4.5: Tüm Seviye RVS'ler için Toplam HIP Mesaj Sayıları

4.1.3.2. HO Süreleri

Simülasyonlarımızda HO süresi, HIP, Hiyerarşik HIP ve eHIP için farklı süreleri ifade etmektedir. HO süresi kısaca, bir MN'nin hareketi sonucu farklı IP adresi elde etme süreci olarak tanımlanabilir. Bir MN'nin hizmet aldığı bir AP'den diğerine geçişte Ip adresi değişmektedir. Klasik HIP'te, yer değiştirme oluştuğunda, MN geçiş yaptığı MN'den hizmet almaya başladığı anda CN ile ve RVS ile üç aşamalı UPDATE prosedürünü gerçekleştirmektedir. HO süresinin bitişi CN'lerin veya RVS'nin son UPDATE mesajını alması ile gerçekleşmektedir. Hiyerarşik HIP'te ise aynı H2 içinde AP'ler arası yer değiştirme esnasında UPDATE prosedürü gerçekleştirilmekte, H2 yerdeğiştirme esnasında ise CN'ler ile UPDATE, RVS ile ise BE prosedürü gerçekleştirilmektedir.

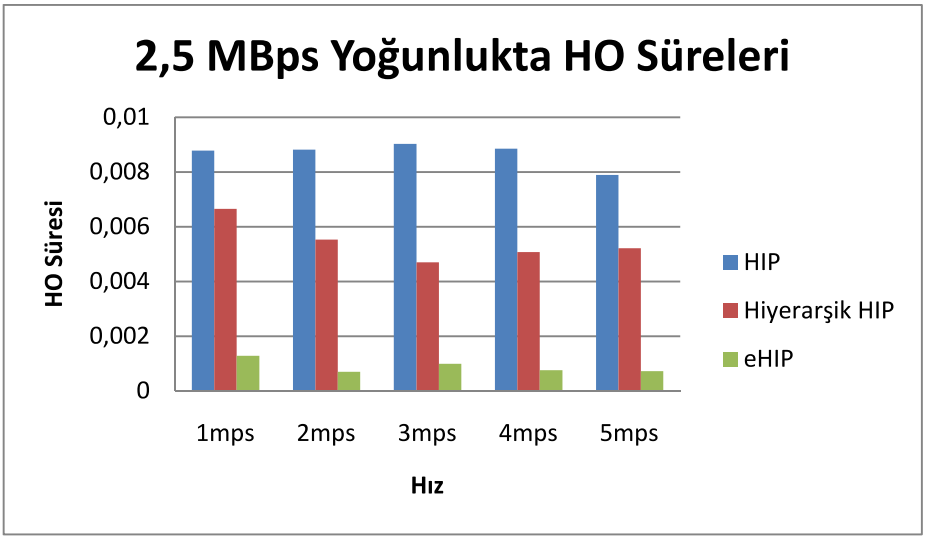
eHIP'te handoff süresi ise, MN'nin geçiş yaptığı AP'den hizmet almaya başladığı an ile FU mesajını yolladığı an arasındaki süre olarak tanımlanmaktadır. Bunun sebebi, erken güncelleme prosedürünü başlatan EU mesajının ve devamındaki kayıt işlemlerinin, henüz MN eski AP'sinden hizmet almaya devam ettiği ancak, yeni AP'den RA mesajı almaya başladığı anda tetiklenmesidir.

Şekil 4.6'da, simüle edilen üç senaryo için zaman çizelgesinde HO sürelerini gösterilmektedir. eHIP yönteminin HO süresi bakımından performansının diğer yöntemlere göre başarımı grafikten görülebilmektedir.

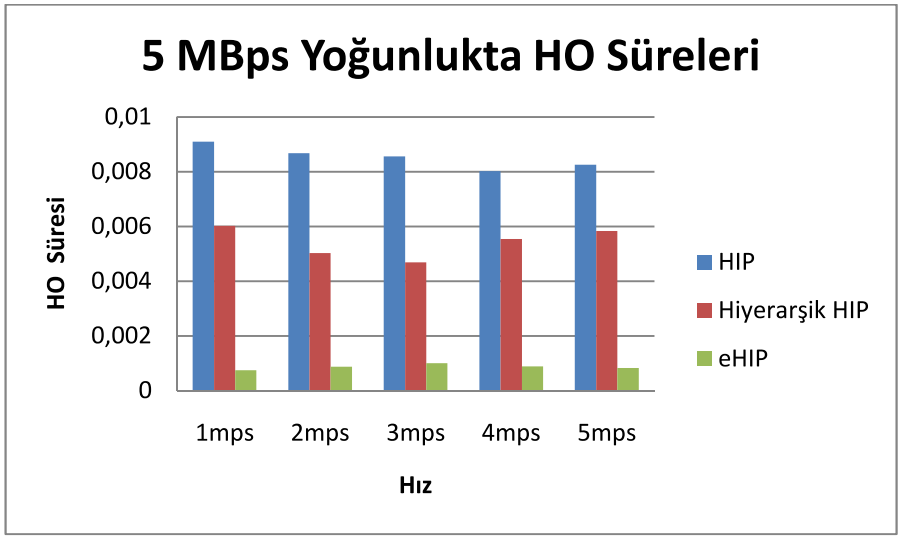


Şekil 4.6: Zamana göre HO süreleri

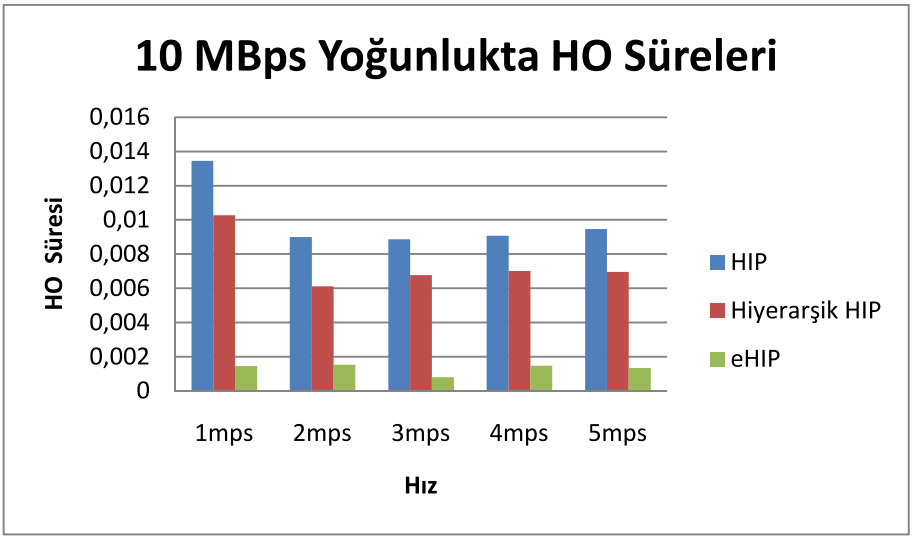
Şekil 4.7, Şekil 4.8 ve Şekil 4.9, simüle edilen farklı trafik yoğunlukları için üç yöntemin karşılaştırmalarını göstermektedir. Belirtilen sonuçlarda HIP'e göre Hiyerarşik HIP'de ortalama %40 zaman kazancı, eHIP'de ise ortalama %90 zaman kazancı sağlanmaktadır.



Şekil 4.7: 2,5 MBps Yoğunlukta HO Süreleri



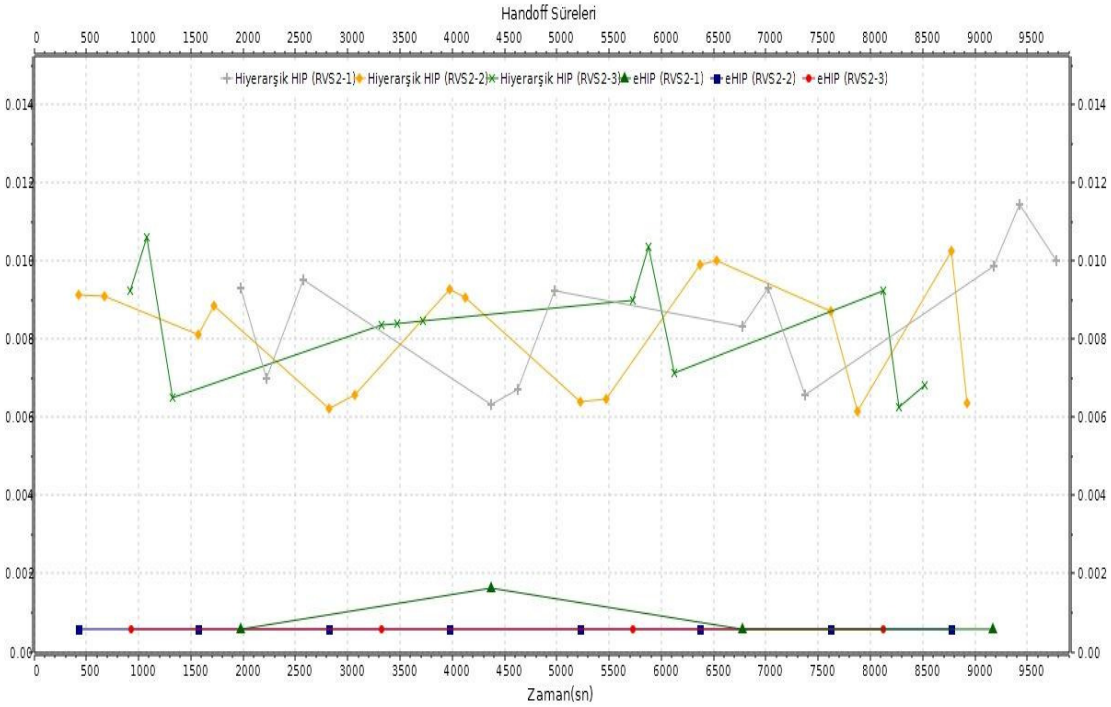
Şekil 4.8: 5 MBps Yoğunlukta HO Süreleri



Şekil 4.9: 10 MBps Yoğunlukta HO Süreleri

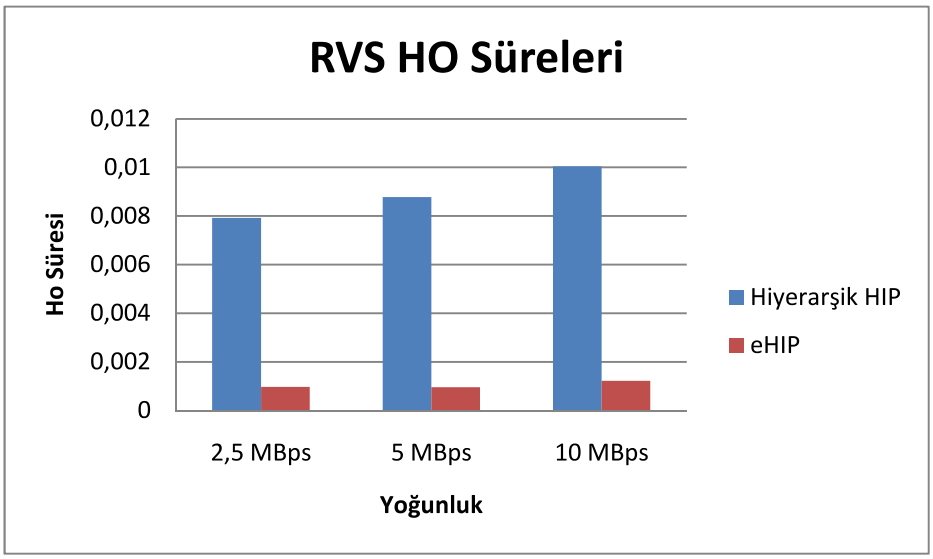
eHIP yönteminde, MN'nin ilk EU mesajını yollaması ile EU3 mesajı arasındaki süre oldukça kısa sürede tamamlanmaktadır. EU3 mesajını alan MN, RA mesajı ile elde ettiği yeni konum bilgileriyle geçiş yapacağı AP'in bağlı olduğu RVS_2'ye kayıt işlemini tamamlamaktadır. Burada bir sonraki aşama ise MN'nin fiziksel geçişini tamamladığı anda FU mesajını yollamasıdır. eHIP'in HO yönteminde MN'nin görevi FU mesajını yollaması ile sona ermektedir. Eğer devam eden iletişimi varsa, bağlı olduğu CN veya CN'lerin bilgilerini FU mesajı ile RVS'ye iletmektedir. CN'lerin güncellenmesi işlemi FU1 mesajları ile RVS tarafından yapılmakta ve bu işlemin yükü MN üzerinden alınmaktadır.

Şekil 4.10'da, simüle edilen üç senaryo için zaman çizelgesinde RVS'ler açısından handoff sürelerini gösterilmektedir. RVS'ye olan HO süresi, MN'nin RVS değişikliği ile ilgili bilgilerinin RVS tarafında güncellemesi için gerekli süre olarak tanımlanmaktadır. eHIP yönteminin, Hiyerarşik HIP'e göre başarımı grafikte gözükmektedir. Burada her ikinci seviye RVS için zaman çizelgesinde ayrı ayrı gösterilmektedir. Klasik HIP senaryosunda ağda tek RVS olduğu ve buna bağlı olarak RVS değişikliği olmadığı için bu yönteme dair sonuçlar verilmemiştir.



Şekil 4.10: Zamana göre RVS HO süreleri

Şekil 4.11'de farklı trafik yoğunlukları için RVS HO süreleri gösterilmektedir. MN'nin farklı hızlardaki hareketindeki değerlerin ortalamaları alınarak ilgili yoğunlukla ilgili ortalama sonuçlar değerlendirilmiştir. Hiyerarşik HIP'te, her RVS değişiminde meydana gelen BE prosedüründen dolayı Hiyerarşik HIP ve eHIP arasındaki fark belirgin şekilde gözükmektedir. eHIP'te kullanılan erken güncelleme mekanizmaları ve RVS'ler arası güvenli kayıt prosedürleri sayesinde, RVS değişimlerinde %90'a varan zaman kazancı sağlayabilmektedir.



Şekil 4.11: Yoğunluğa Göre RVS HO Süreleri

4.1.3.3. eHIP Mesaj Süreleri

Tablo 4.1, Tablo 4.2 ve Tablo 4.3 eHIP yönteminde EU mesajları ve FU mesajları arasındaki süreleri göstermektedir. EU-EU3 mesajları arasındaki süre MN'nin yeni RVS_2'sine kaydının tamamlandığı süredir. Tablolar incelendiğinde EU-EU3 arasındaki sürenin oldukça kısa sürdüğü gözlemlenmektedir. Bunun sebebi, RVS'ler arası karşılıklı güvenli mesajlaşma olması ve birbirlerinden gelen kayıt mesajlarını MN'yi sorgulamadan kabul etmeleri ve gerçekleştirmeleridir.

Tablo 4.1: EU-EU3 Mesajları arasındaki Toplam Süreler

		Ortalama Süre (sn)
2,5 MBps	1 mps	0,026222298
	2mps	0,017052211
	3mps	0,020059064
	4mps	0,019424499
	5mps	0,020226495
5 MBps	1 mps	0,044674033
	2mps	0,045651141
	3mps	0,04736023
	4mps	0,053590974
	5mps	0,053129276
10 Mbps	1 mps	0,148039225
	2mps	0,154485486
	3mps	0,151771204
	4mps	0,157905572
	5mps	0,160419926

Tablo 4.2 ve Tablo 4.3 sırasıyla EU mesajı ile FU mesajı arasındaki süreleri ve EU3 mesajı ile FU mesajı arasındaki süreleri göstermektedir. EU mesajları ve FU mesajları arasındaki sürenin kayıt işlemlerine göre uzun olmasının sebebi, bu geçen sürede MN'nin fiziksel olarak bir AP'den diğerine geçiş yapması ve yeni IP alma prosedürünü tamamlamasıdır. Eski hizmet aldığı AP'den tamamen kopmak ve yeni AP'si ile yeni IP adresini konfigüre etmesi arasındaki geçen süre EU-EU3 süresine göre MN'nin hızına göre değişiklik göstermektedir. MN'nin hızının artması, tüm senaryolarda sabit olan mesafeyi daha hızlı katetmesini sağladığı için ortalama süreyi de azaltmaktadır.

Tablo 4.2: EU-FU Mesajları arasındaki Toplam Süreler

		Ortalama Süre (sn)
2,5 MBps	1 mps	139,1263
	2mps	73,11657
	3mps	50,23085
	4mps	38,23669
	5mps	31,15185
5 MBps	1 mps	142,7654
	2mps	72,9211
	3mps	50,12489
	4mps	38,26546
	5mps	31,05327
10 Mbps	1 mps	140,586
	2mps	73,10213
	3mps	50,08748
	4mps	38,48631
	5mps	31,05704

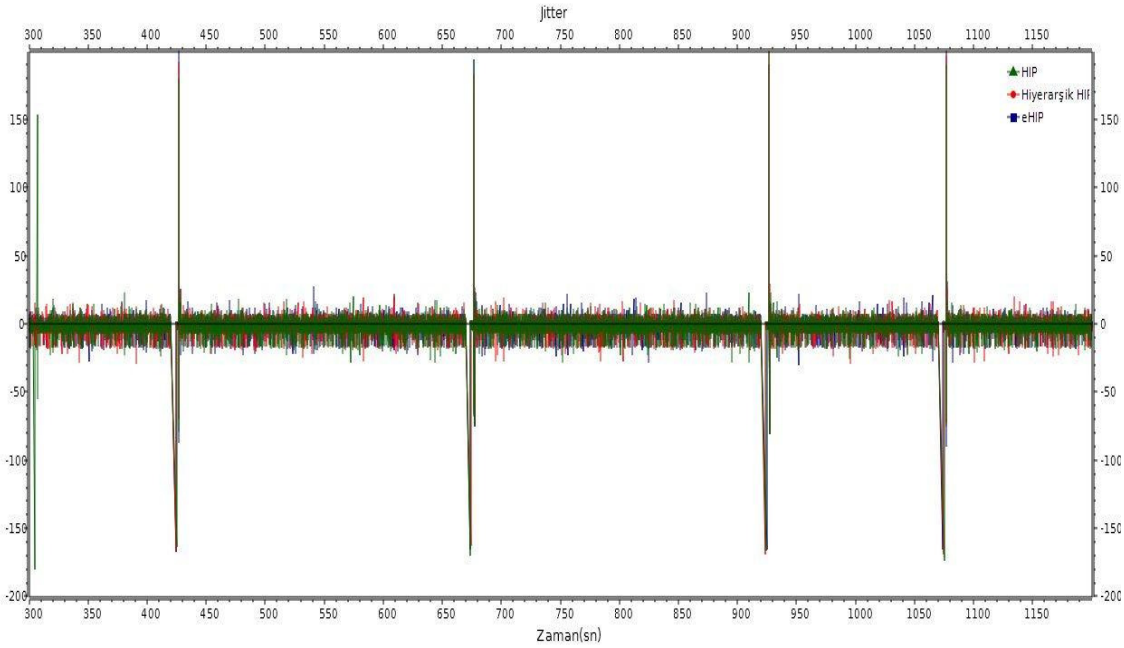
Tablo 4.3: EU3-FU Mesajları arasındaki Toplam Süreler

		Ortalama Süre (sn)
2,5 MBps	1 mps	139,1002
	2mps	73,09906
	3mps	50,20834
	4mps	38,21727
	5mps	31,12982
5 MBps	1 mps	142,718
	2mps	72,87368
	3mps	50,07481
	4mps	38,21169
	5mps	30,99855
10 Mbps	1 mps	140,429
	2mps	72,94212
	3mps	49,93402
	4mps	38,32841
	5mps	30,89591

4.1.3.4. Jitter

Jitter, bir hedefe belirli bir kaynaktan gelen paketlerin gecikme seğirmesini ifade etmektedir. Özellike VoIP gibi gecikmeye duyarlı uygulama tiplerinde düşük jitter değeri önem taşımaktadır. Şekil 4.12'de simüle edilen üç senaryo için zaman

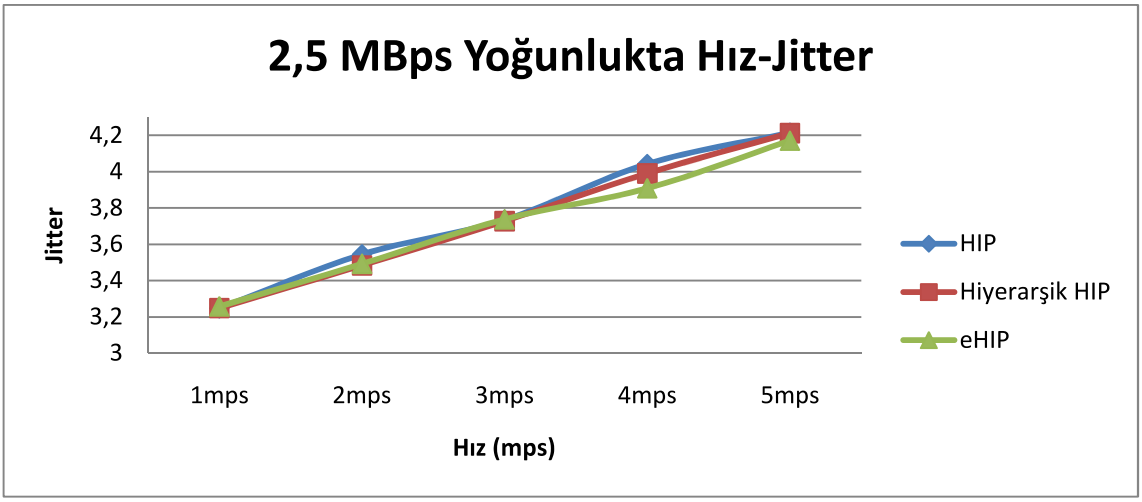
çizelgesinde jitter değerleri gösterilmektedir. Çizelgelerde gözüken sıçramalar her üç yöntemde de HO anlarında olmaktadır.



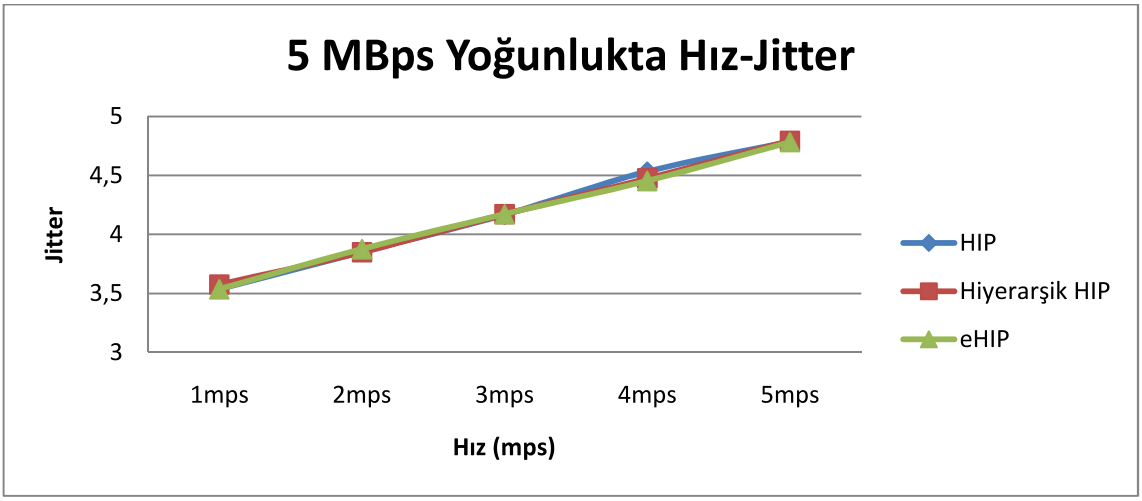
Şekil 4.12: Zamana göre Jitter

Şekil 4.13, 4.14 ve 4.15 farklı trafik yoğunluklarında MN'nin değişen hızına göre jitter değerlerini her üç yöntem için de göstermektedirler. Bu grafikler, jitter değerlerinin standart sapması olarak verilmektedirler.

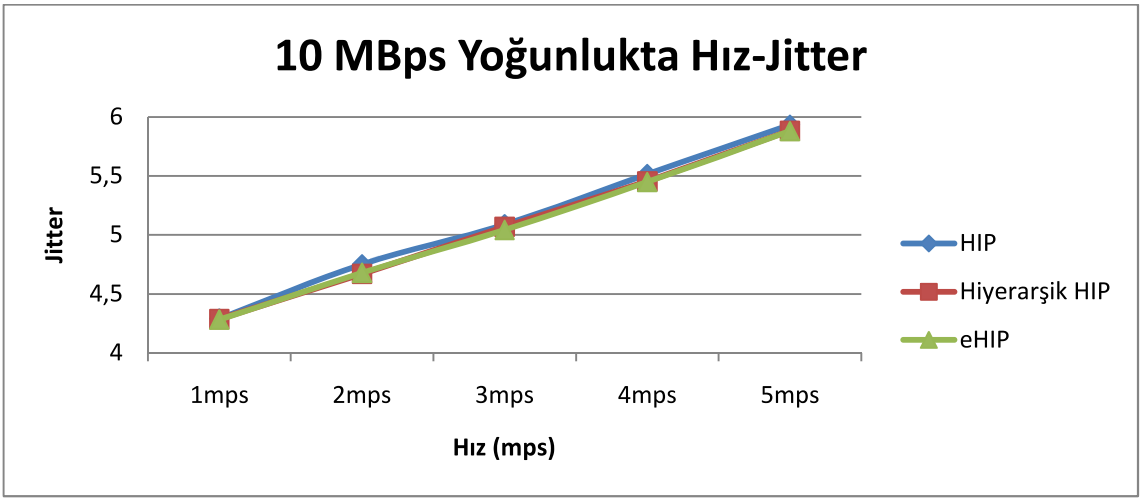
Farklı trafik yoğunluklarında eHIP yönteminin jitter değerlerinin diğer yöntemlere göre çok az miktarda da olsa daha düşük olduğu gözlemlenmiştir. Yöntemin getirdiği ek mesajlaşma yüküne rağmen, HO gecikmesinde elde edilen kazanç sayesinde bu başarım sağlanmaktadır.



Şekil 4.13 : 2,5 MBps Yoğunlukta Hıza Bağlı Jitter



Şekil 4.14: 5 MBps Yoğunlukta Hıza Bağlı Jitter



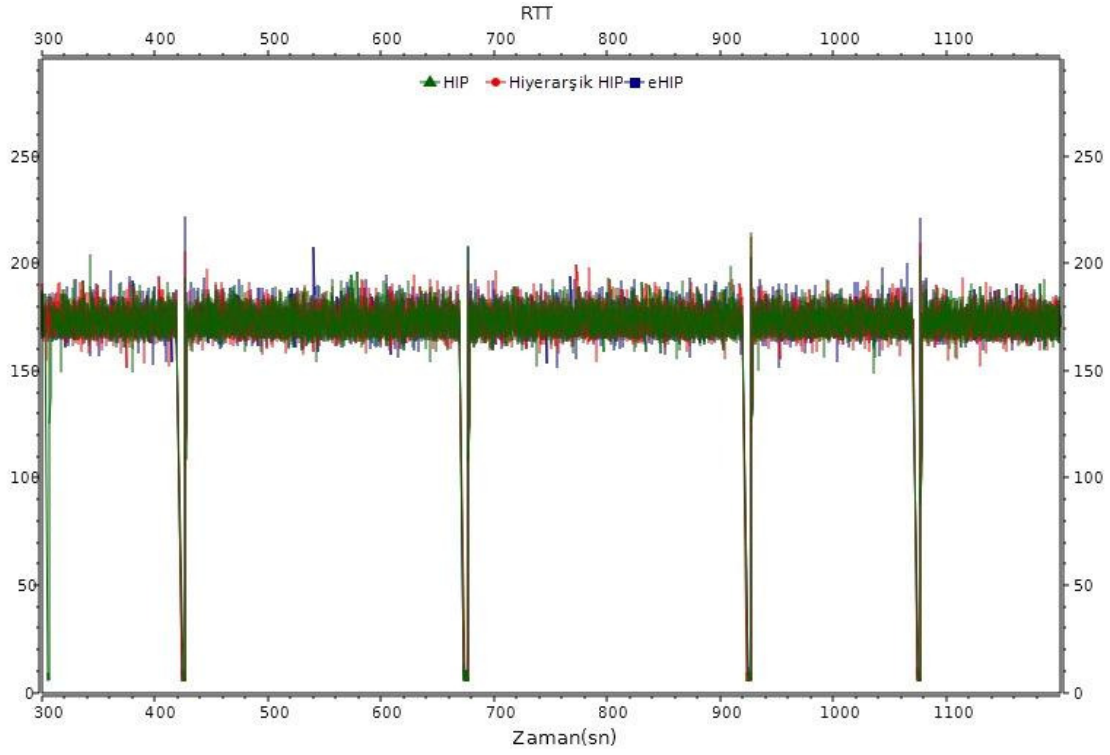
Şekil 4.15: 10 MBps Yoğunlukta Hıza Bağlı Jitter

Sonuç grafiklerinden de görülebileceği gibi, MN'nin hızı arttıkça jitter değerlerinde artış gözlemlenmektedir. Bunun sebebi, simülasyon süresinin tüm çalışmalarda sabit olarak alınması (10000 s) ve hız arttıkça MN'nin gerçekleştirdiği HO miktarının artış göstermesidir. Ağda meydana gelen HO sayısındaki artış, jitter değerinde de artışa neden olmaktadır. Bu durumu Şekil 4.12'de gösterilen zamana göre jitter grafiğinin HO anlarındaki sıçrama miktarının yüksekliğine bakarak da anlayabilmekteyiz.

4.1.3.5. Round Trip Time (RTT)

Simülasyonlarımızda RTT, kaynak TCP ve UDP uygulamalarında ağa gönderilen paketlerin hedefe ulaşması ve hedeften kaynağa cevap dönmesi arasındaki geçen süre olarak tanımlanmaktadır.

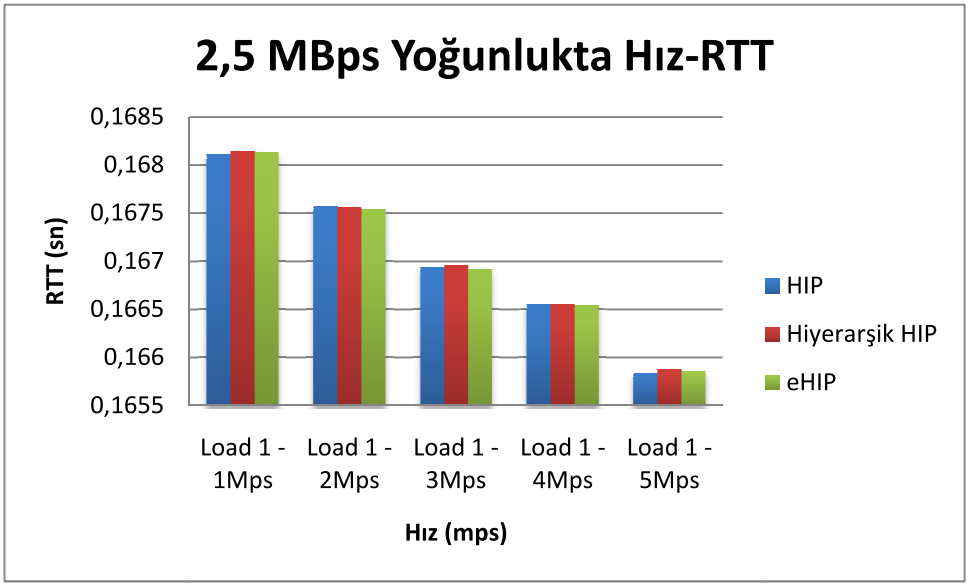
Şekil 4.16'de simüle edilen üç senaryo için zaman çizelgesinde RTT değerleri gösterilmektedir. Grafikte gösterilen sıçramalar, çeşitli nedenlerle meydana gelen paket düşmelerinden dolayı meydana gelmektedir.



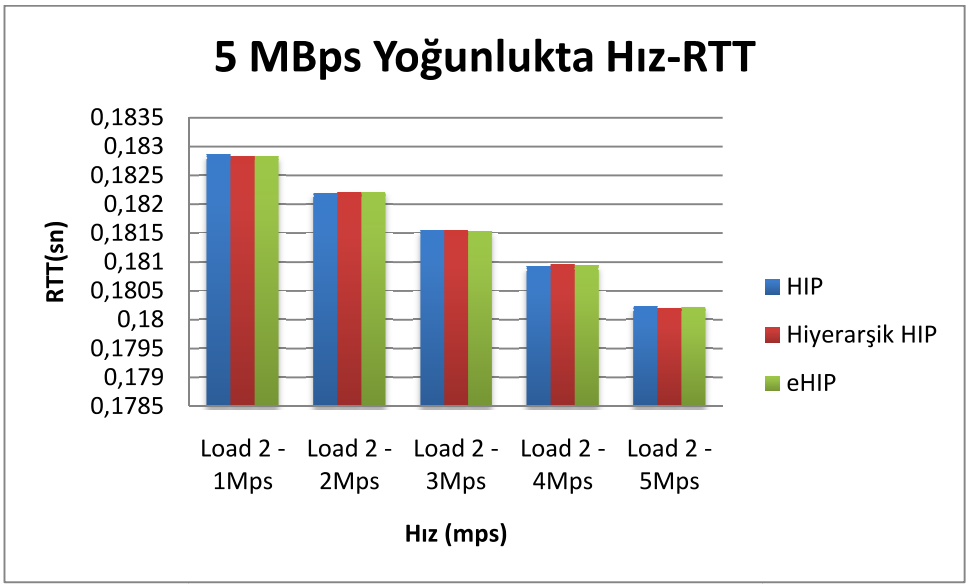
Şekil 4.16: Zaman göre RTT

Şekil 4.17, 4.18 ve 4.19 sırasıyla her üç yöntem için farklı trafik yoğunluklarının hıza bağlı sonuçlarını göstermektedirler. MN'nin hızının RTT değerine gözle görülür

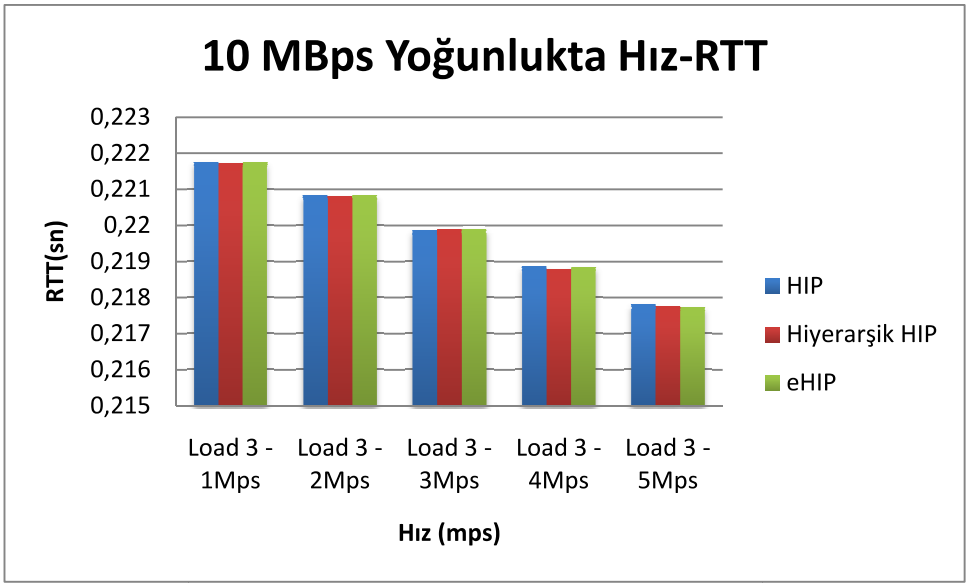
derecede bir etkisi olmadığı grafiklerden gözlemlenmektedir. RTT değerindeki değişim paketlerin kuyrkta bekleme ve düşme durumlarına bağlıdır. Düğümün hızı ve HO süresi dolaylı olarak bu değerleri etkilemektedir. Her üç yöntemde de hıza ve yoğunluğa göre dağılımlar incelendiğinde, aralarında belirgin bir fark olmadığı görülmektedir. eHIP yönteminde ve hiyerarşik HIP yapılarında, ağa dahil edilen ekstra ağ elemanları olan RVS'lerin varlığının toplam mesaj sayısında ağa getirdiği ek yüke rağmen RTT zamanlarını negatif yönde etkilemediği görülmektedir.



Şekil 4.17: 2,5 MBps Yoğunlukta Hıza Bağlı RTT



Şekil 4.18: 5 MBps Yoğunlukta Hıza Bağlı Jitter



Şekil 4.19: 10 MBps Yoğunlukta Hıza Bağlı Jitter

4.2. P-EHIP YÖNTEMİNİN PERFORMANS İNCELEMESİ

4.1.1. Simülasyon Detayları

p-eHIP yönteminin test edilmesi için kullanılan algoritma MATLAB ortamında test edilmiş ve nümerik sonuçlar elde edilmiştir. Bu simülasyonda elde edilen sonuçlar çizilen ağ topolojisi ve MN'nin bu ağ içinde izlediği yola bağlı olarak değişiklik göstermektedir.

Simülasyonda istenirse MN'nin izlediği yol kullanıcı tarafından rasgele aralıklarla çizilebilmektedir ve çizilen her bir adım MN'nin izlediği yolda bir adım/birim zaman olarak kabul edilmektedir. Program tarafından otomatik olarak bir yol çizilmesi durumunda ise birtakım kurallar uygulanmaktadır. Her bir çizilen adımın X veya Y ekseninde, doğu (sağ) veya batı (sol) yönünde olmasına programda ayarlanan olasılıklar doğrultusunda karar verilmektedir. Başka bir deyişle, MN'nin x ekseninde sağa doğru veya sola doğru hareket etme veya hareket etmeme durumuna ayarlanan olasılıklarına göre karar verilmektedir. MN'nin soldan sağa doğru ve dairesel bir yapıda otomatik bir yol izlemesini sağlamak için x ekseninde sağa doğru gitme olasılığı %40, sola doğru gitme olasılığı %20 ve hareket etmeme olasılığı %40 olarak ayarlanmıştır. Yine benzer

şekilde MN'nin y ekseninde yukarı yönlü hareket etme ihtimali %40, aşağı yönlü hareket etme olasılığı %20 ve hareket etmeme olasılığı %40 olarak ayarlanmıştır.

Her iki modda da toplam RVS sayısı, her bir RVS'ye bağlı AP sayısı ve AP'lerin çap bilgileri kullanıcıdan alınmaktadır. Kullanıcı tarafından yapılan yol çizimlerinde 500x500 birim^2 büyüklüğünde bir alan kullanılmakta, program tarafından otomatik olarak yapılan yol çizimlerinde ise kullanıcıdan alınan adım sayısına bağlı olarak kullanılan alan ayarlanmaktadır.

Yöntemin test edilmesinde kullanılan bir diğer değer de tahmini olarak ve normal olarak gerçekleştirilen HO sürelerinin karşılaştırılmasıdır. p-EU süresi olarak adlandırılan tahmine dayalı HO süresi, bir MN'nin kapsama alanına girdiği AP için yapılan son başarılı tahmin ile kapsama alanına girişinden sonraki ilk adımı arasındaki birim zaman olarak kabul edilmektedir. Kapsama alanına girilmesinden bir adım sonrası olarak alınmasının sebebi, MN'nin kapsama alanına girdiği anda normal olarak talep ettiği EU sonrasında zaten önceden kayıtlı olduğu için geri kalan prosedürün gerçekleşmesine gerek kalmayışı ve erken güncellemenin önceden sonlanmasıdır. n-EU süresi ise MN'nin farklı RVS_2'ye ait bir AP'nin kapsama alanına girdiği anda EU mesajını talep etmesi ile eski AP'sinden bağlantısını tamamen kopardığı an olarak kabul edilmektedir.

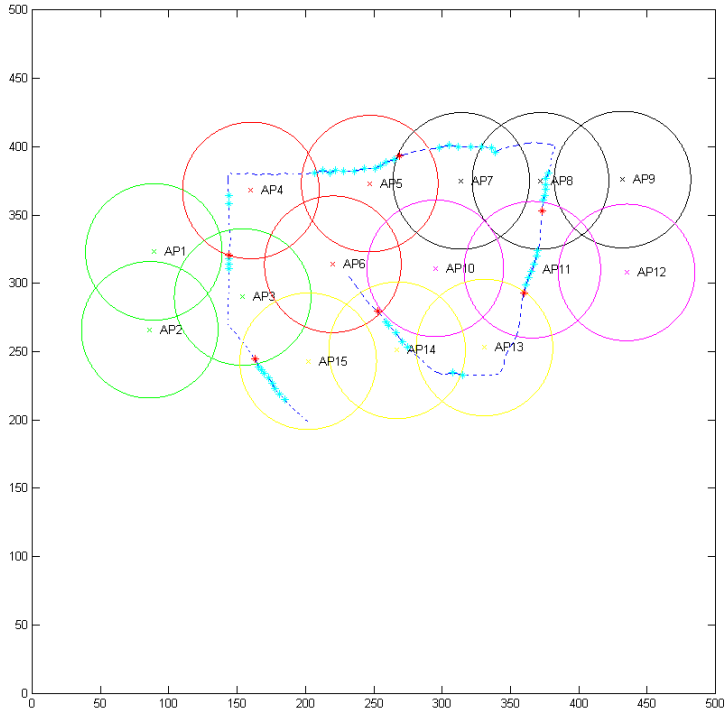
Yapılan tahminler öncelikle MN'nin her adımında RVS'ye güncelleme yaptığı varsayılarak test edilmiştir. Daha sonra periyot değeri değiştirilerek sonuca olumlu veya olumsuz etkileri görülmüş ve bunların nedenleri irdelenmiştir. Periyot değerleri p=1, p=2 ve p=3 olarak alınmıştır. p=2 olduğunda, MN iki birim zamanda bir konum bilgisini RVS'ye yollayarak güncellemekte, p=3 olduğunda ise 3 birim zamanda bir konum bilgisini RVS'ye yollayarak güncellemektedir.

4.1.2. Simülasyon Sonuçları

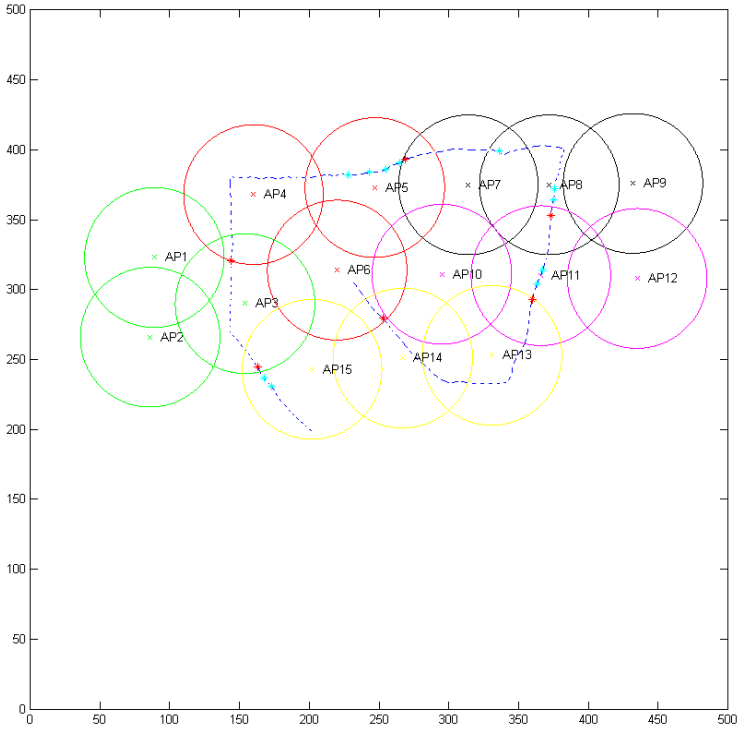
4.1.2.1. Topoloji 1 için Elde edilen Değerler

Yöntemin test edildiği ilk topolojide beş farklı RVS_2, her bir RVS'de üçer adet AP bulunmaktadır. MN'nin hareketi için de AP15'den başlayarak AP6'da sonlanan bir yol kullanıcı tarafından adım adım çizilmiştir. Şekil 4.20, Şekil 4.21 ve Şekil 4.22 bu

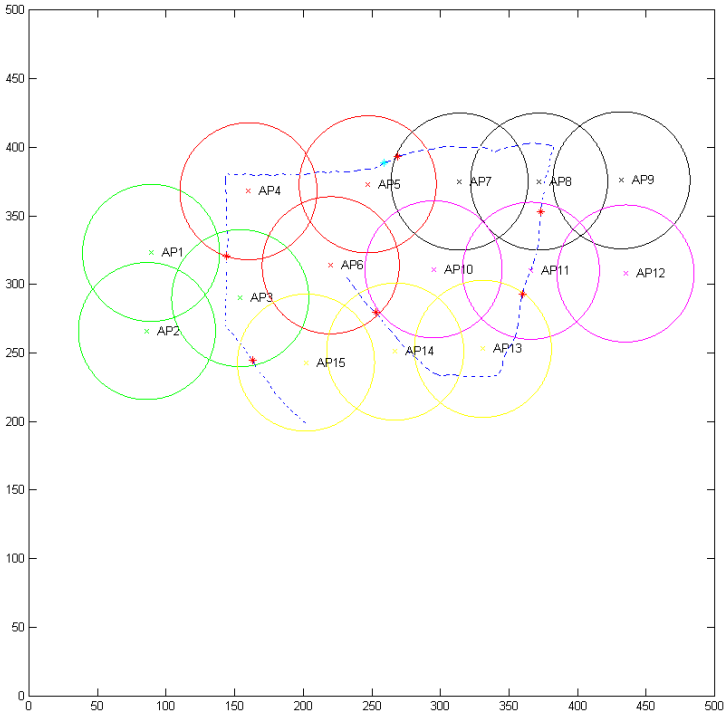
topolojide periyot=1, periyot=2 ve periyot=3 değerleri ile yapılan tahminleri ve EU noktalarını göstermektedir.



Şekil 4.20: p-eHIP Yönteminde Topoloji 1 (p=1)



Şekil 4.21: p-eHIP Yönteminde Topoloji 1 (p=2)



Şekil 4.22: p-eHIP Yönteminde Topoloji 1 (p=3)

Yukarda verilen şekillerden periyot arttıkça yapılan tahmin sayısının p-eHIP'in işleyiş kuralları gereği azaldığı görülmektedir. Bunların her birinde gerçekleştirilen doğru ve yanlış tahmin sayıları Tablo 4.4 'te gösterilmektedir.

Tablo 4.4: p-eHIP Topoloji 1 için Tahmin ve HO Sayıları

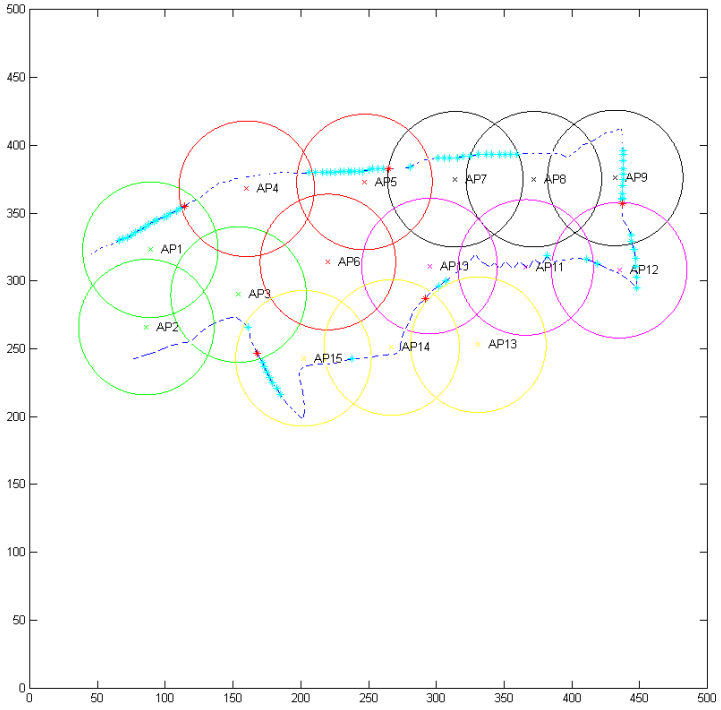
Mod	**Periyot**	**Toplam Tahmin**	**Yanlış**	**Doğru**	**HO Sayısı**	**EU Yapılamayan HO Sayısı**
p-eHIP	1	50	7	43	6	0
p-eHIP	2	11	0	11	6	2
p-eHIP	3	1	0	1	6	5
Hız-p-eHIP	1	9	3	6	6	0
Hız-p-eHIP	2	5	0	5	6	2
Hız-p-eHIP	3	1	0	1	6	5

Daha önce de belirtildiği gibi sonuçlar MN'nin izlediği yola, yol boyunca attığı adımların hızına bağlıdır. Kullanıcı tarafından çizilen bu yolda MN sabit bir hız sergilememekte, her adımda farklı mesafe almaktadır. p-eHIP'de p=1 olduğunda toplamda normalde gerçekleştirilen tüm EU taleplerinden önce tahmin yapılabilmiştir. Bu tahminlerde de % 86 olumlu sonuç elde edilmiştir. Periyot değeri arttıkça

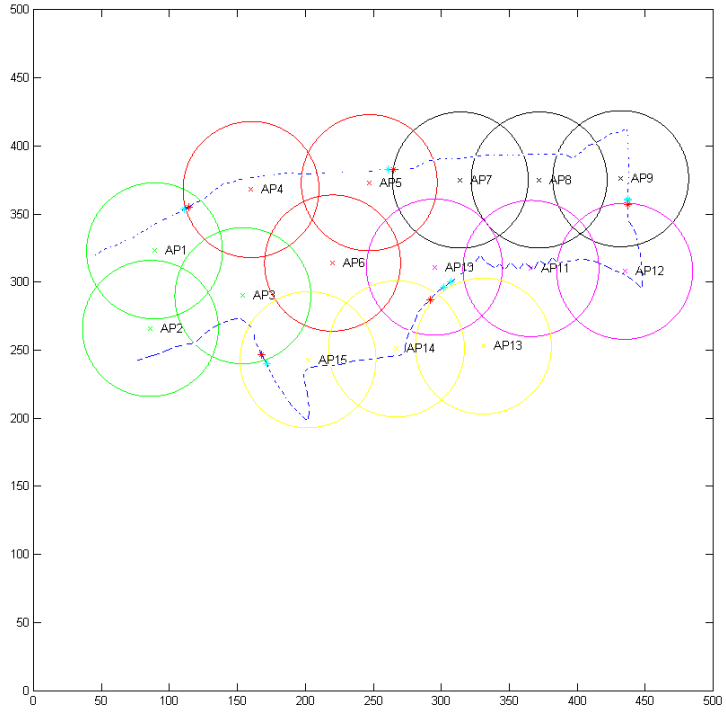
yapılabilen tahmin sayılarının azalması beklendiği gibi izlenmektedir. p=2 ve p=3 değerlerinde toplam tahmin sayılarında belirgin düşüşler gözlemlenmekte ancak bununla beraber yanlış tahmin sayısı da azalmaktadır. Ancak bu periyotlarda yapılan tahminlerin tümü doğru olmasına rağmen, tüm HO'lar için tahmine dayalı EU yapılamadığı gözlemlenmiştir. Hız faktörü katılmış olan p-eHIP modunda amacımız, ağda tahmine dayalı birden fazla aynı değeri veren gereksiz güncelleme ve yanlış tahmin sayısını azaltmaktır. Tablodan gözlemlenebildiği gibi toplam tahmin sayısında yaklaşık %80'lik ve yanlış tahmin sayısında yaklaşık % 70'lik bir düşüş meydana gelmiştir.

4.1.2.2. Topoloji 2 için Elde edilen Değerler

Yöntemin test edildiği ikinci topolojide beş farklı RVS_2, her bir RVS'de üçer adet AP bulunmaktadır. MN'nin hareketi için de AP1'den başlayarak AP2'de sonlanan bir yol kullanıcı tarafından adım adım çizilmiştir. Şekil 4.23 ve Şekil 4.24 bu topolojide periyot=1 için normal p-eHIP ve Hız faktörlü p-eHIP modlarında yapılan tahminleri ve EU noktalarını göstermektedir.



Şekil 4.23: p-eHIP Yönteminde Topoloji 2 (p=1)



Şekil 4.24: Hız Faktörlü p-eHIP Yönteminde Topoloji 2 (p=1)

Tablo 4.5: p-eHIP Topoloji 2 için Tahmin ve HO Sayıları

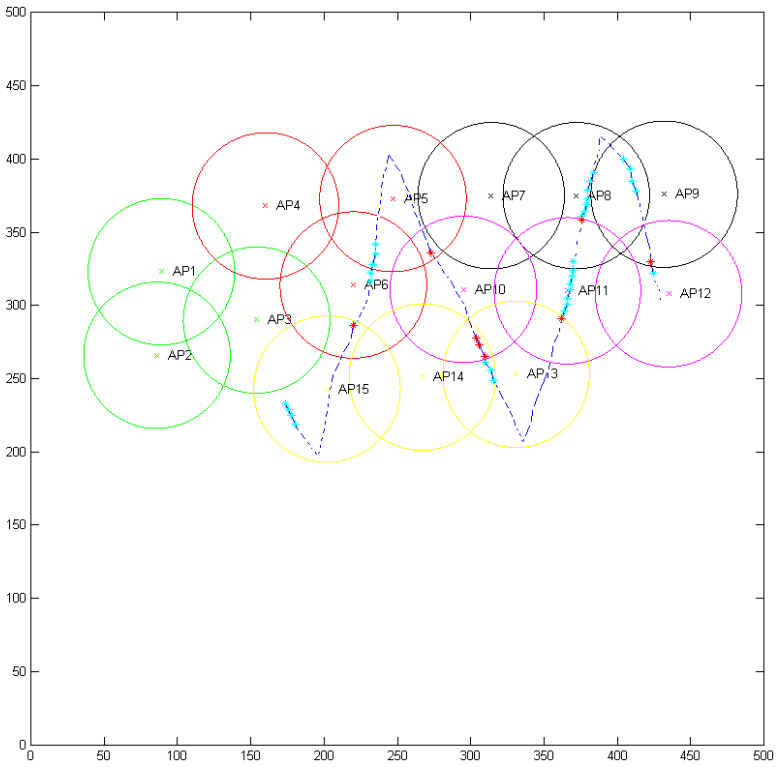
Mod	Periyot	Toplam Tahmin	Yanlış	Doğru	HO Sayısı	EU Yapılamayan HO Sayısı
p-eHIP	1	77	28	49	5	0
p-eHIP	2	24	7	17	5	0
p-eHIP	3	9	3	6	5	2
Hız-p-eHIP	1	7	2	5	5	0
Hız-p-eHIP	2	5	1	4	5	0
Hız-p-eHIP	3	3	0	3	5	2

Topoloji 2 kullanılarak elde edilen doğru ve yanlış tahmin sayıları Tablo 4.5 'te gösterilmektedir. Bu topolojide MN'nin daha dikdörtgensel ve aynı yönde bir yol izlediği görülmektedir. Toplam beş adet HO gerçekleştirilen bu topolojide p=1 için yapılan tahminlerde doğru tahmin oranı %64 civarındayken, aynı periyodun hız faktörlü moduna hem yanlış tahmin sayısı azalmış hem de doğru tahmin oranı %71'e yükselmiştir. Topolojinin yön olarak tutarlığı p=2 olduğunda da doğru tahmin oranının yüksek değerlerde elde edilmesini sağlamıştır. Toplam tahmin sayısı azaldığı gibi yanlış tahmin sayısı da azalarak p=2'de doğruluk oranları %71 ve %80 olarak elde edilmiştir.

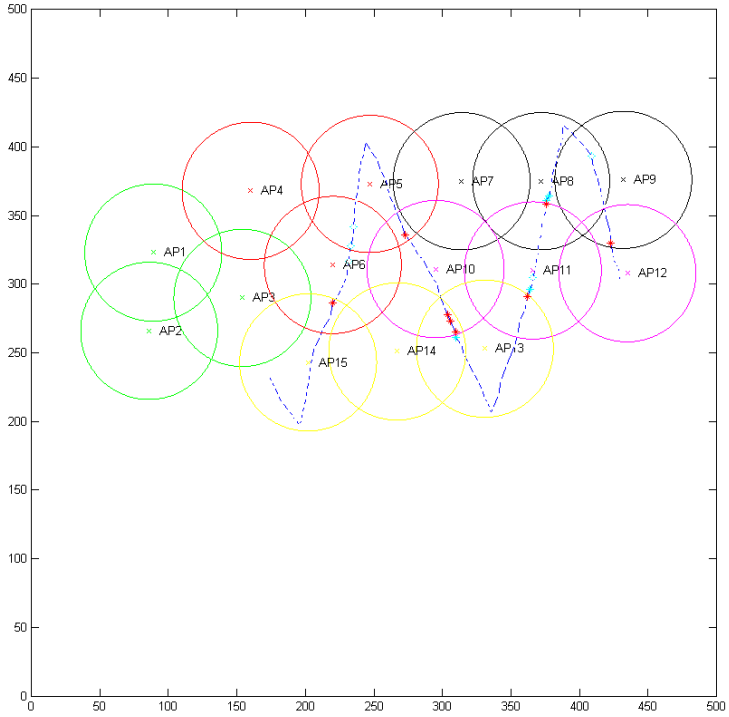
p=3 olduğunda ise EU tahmini yapılamayan HO sayısı ortaya çıkmaktadır. Bu topolojide optimum periyot değerinin maksimum p=2 olduğu gözlemlenmiştir.

4.1.2.3. Topoloji 3 için Elde edilen Değerler

Yöntemin test edildiği üçüncü topolojide beş farklı RVS_2, her bir RVS'de üçer adet AP bulunmaktadır. MN'nin hareketi için de AP12'den başlayarak AP15'te sonlanan, dikey ekseni izleyen bir yol kullanıcı tarafından adım adım çizilmiştir. Şekil 4.25 ve Şekil 4.26 bu topolojide periyot=1 için normal p-eHIP ve Hız faktörlü p-eHIP modlarında yapılan tahminleri ve EU noktalarını göstermektedir.



Şekil 4.25: p-eHIP Yönteminde Topoloji 3 (p=1)



Şekil 4.26: Hız Faktörlü p-eHIP Yönteminde Topoloji 3 (p=1)

Tablo 4.6: p-eHIP Topoloji 3 için Tahmin ve HO Sayıları

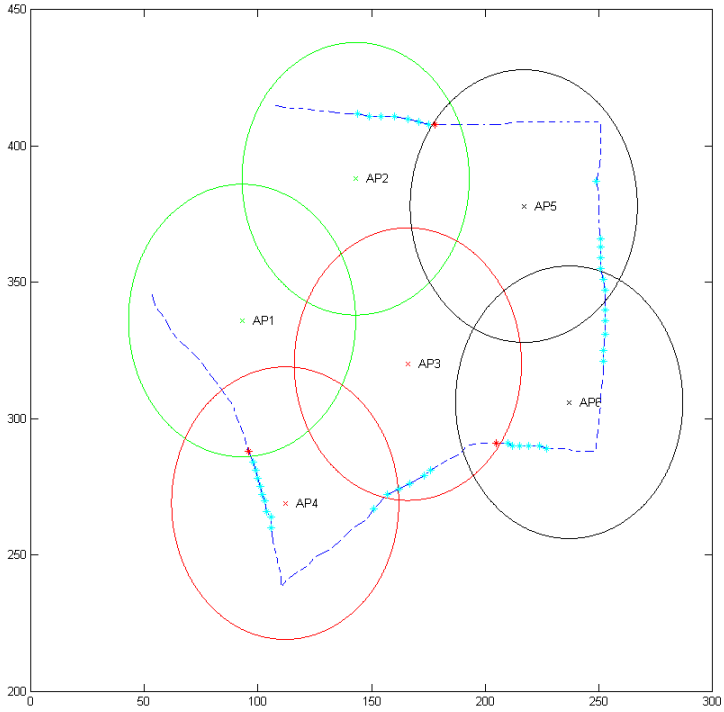
Mod	Periyot	Toplam Tahmin	Yanlış	Doğru	HO Sayısı	EU Yapılamayan HO Sayısı
p-eHIP	1	32	13	19	8	3
p-eHIP	2	7	4	3	8	5
p-eHIP	3	1	0	1	8	7
Hız-p-eHIP	1	4	0	4	8	5
Hız-p-eHIP	2	2	0	2	8	6
Hız-p-eHIP	3	1	0	1	8	7

Aynı topoloji üzerinde farklı bir yol izlenerek gerçekleştirilen üçüncü simülasyonda, MN tarafından dikey eksende hareket eden ve böylece çok sık farklı RVS'ler arasındaki AP'ler arasında geçiş yapılan bir yol izlenmiştir. Bu sık geçişlerin toplam tahmin ve doğru tahmin oranlarına olan olumsuz etkisi Tablo 4.6'da belirtilen değerlerden gözükmektedir. p=1 için %59 oranında bir doğru tahmin oranı sağlanmışken yolun tamamında gerçekleştirilen sekiz HO'nun üç tanesi için tahmin yapılamadığı gözlemlenmiştir. Tahmin yapılamayan HO sayısı periyot arttıkça artmakta ve sonuçta buna da bağlı olarak yapılabilen tahmin sayıları da azalmaktadır. Diğer bir yandan, hız faktörünü içinde barındıran yöntemde yapılan tahminlerin az sayıda ancak doğru

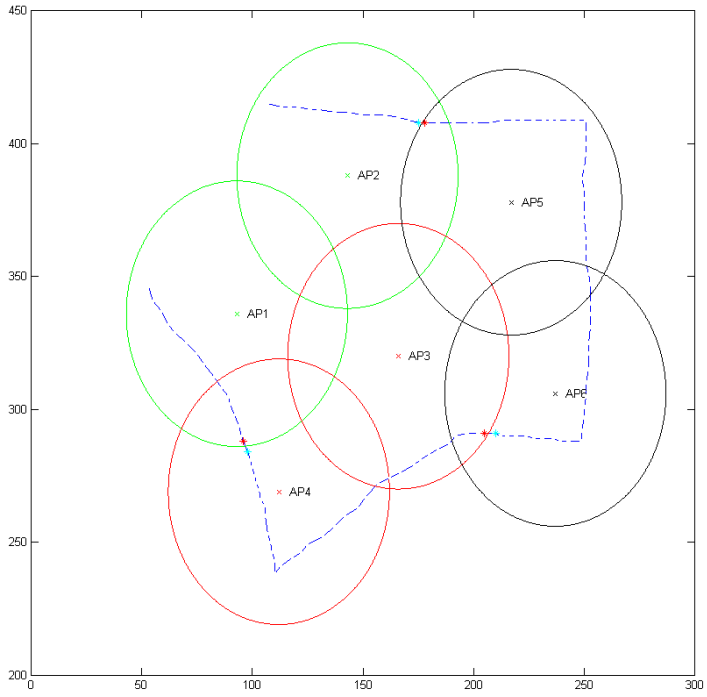
oldukları gözlemlenebilmektedir. Bu da hıza bağlı olarak verilen kararın bu şekilde çok da düzenli olmayan bir hareket esnasında bile başarılı olabildiği görülmüştür.

4.1.2.4. Topoloji 4 için Elde edilen Değerler

Yöntemin test edildiği dördüncü topolojide üç farklı RVS_2, her bir RVS'de ikişer adet AP bulunmaktadır. MN'nin hareketi için de AP2'den başlayarak AP1'de sonlanan, düzgün bir yol kullanıcı tarafından adım adım çizilmiştir. Şekil 4.27 ve Şekil 4.28 bu topolojide periyot=1 için normal p-eHIP ve Hız faktörlü p-eHIP modlarında yapılan tahminleri ve EU noktalarını göstermektedir. Az sayıda ağ elemanı ile incelenen bu topolojide özellikle p=1 değerinde hız faktörünün etkisi açık bir şekilde gözükmektedir.



Şekil 4.27: p-eHIP Yönteminde Topoloji 4 (p=1)



Şekil 4.28: Hız Faktörlü p-eHIP Yönteminde Topoloji 4 (p=1)

Tablo 4.7: p-eHIP Topoloji 4 için Tahmin ve HO Sayıları

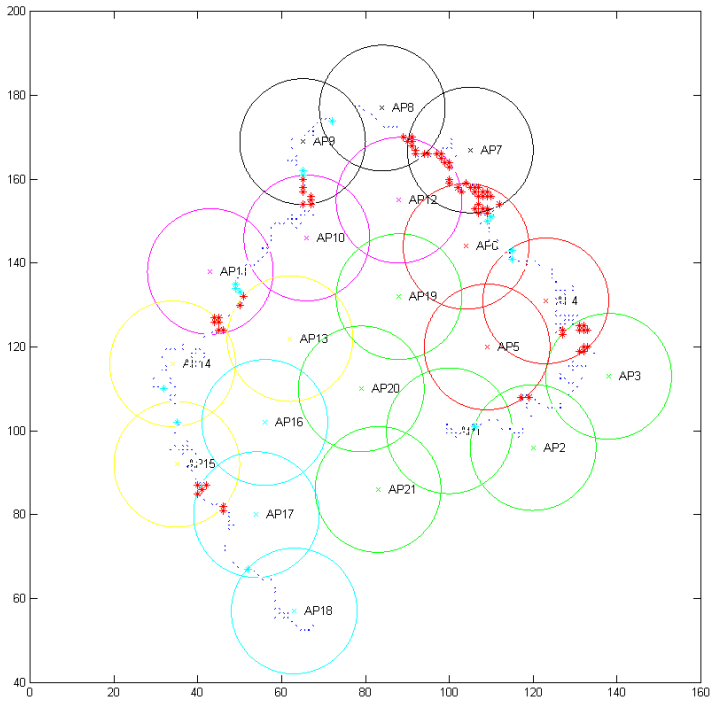
Mod	Periyot	Toplam Tahmin	Yanlış	Doğru	HO Sayısı	EU Yapılamayan HO Sayısı
p-eHIP	1	40	1	39	3	0
p-eHIP	2	8	0	8	3	0
p-eHIP	3	2	0	2	3	2
Hız-p-eHIP	1	3	0	3	3	0
Hız-p-eHIP	2	3	0	3	3	0
Hız-p-eHIP	3	0	0	0	3	3

Tablo 4.7 ve topolojiyi gösteren şekillerden görülebileceği gibi az sayıda ağ elemanı bulunan bu topolojide toplam üç adet HO gerçekleşmektedir. Hız faktörünün olmadığı modda, toplam tahmin sayısı 40 olarak elde edilmiştir. Bunların %98'lik oranı doğrudur ancak gereksiz yere ağda fazla güncelleme işlemine sebep olmaktadır. Hız faktörünün olduğu modda ise p=1 için toplam tahmin sayısında %92,5'lik bir düşüş meydana gelmiş ancak yapılan tahminlerin hepsinin doğru olduğu ve toplam 3 HO için de p-EU yapılabildiği görülmüştür.

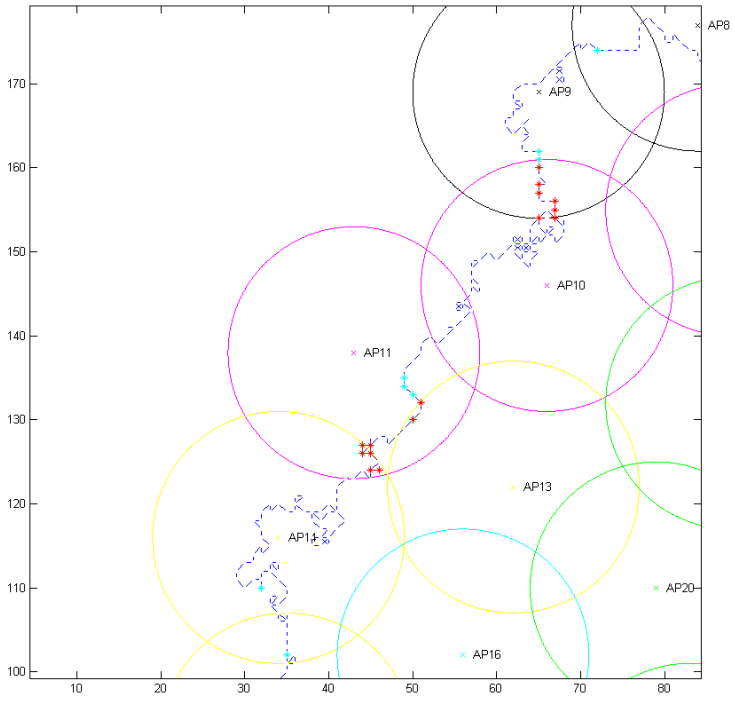
p=2 ve p=3 için toplam tahmin oranları gözle görülür derecede düşmüş ve hatta hız faktörünün olduğu modda sıfırlanmıştır. Bunun sebebi, MN'nin adım aralıklarının çok sık olmaması yani bir anlamda hızlı hareket ediyor olmasıdır. Periyot değeri büyük seçildiğinde p-eHIP işleyişi gereği tahminlerin yapılması ve EU kararlarının verilmesi için gerekli adım sayısı sağlanamamaktadır.

4.1.2.5. Topoloji 5 için Elde edilen Değerler

Yöntemin test edildiği beşinci topolojide altı farklı RVS_2, her bir RVS'de üçer adet AP bulunmaktadır. MN'nin hareketi için de AP1'den başlayarak AP18'de sonlanan, toplam 500 adımdan oluşan bir yol, program tarafından adım adım otomatik olarak çizdirilmiştir. Şekil 4.29 ve Şekil 4.30 bu topolojide periyot=1 için yapılan tahminleri ve EU noktalarını göstermektedir. Şekil 4.30'da otomatik olarak çizdirilen yolda MN'nin hareketinin ayarlanan parametrelere göre sıklıkla değişkenlik gösterdiği görülebilmektedir. Burada izlenen yol sağa, sola, yukarı veya aşağıya doğru bir birim veya $\sqrt{2}$ birim olabilmektedir.



Şekil 4.29: p-eHIP Yönteminde Topoloji 5 (p=1)



Şekil 4.30: p-eHIP Yönteminde Topoloji 4 (p=1) için İzlenen Yol Detayı

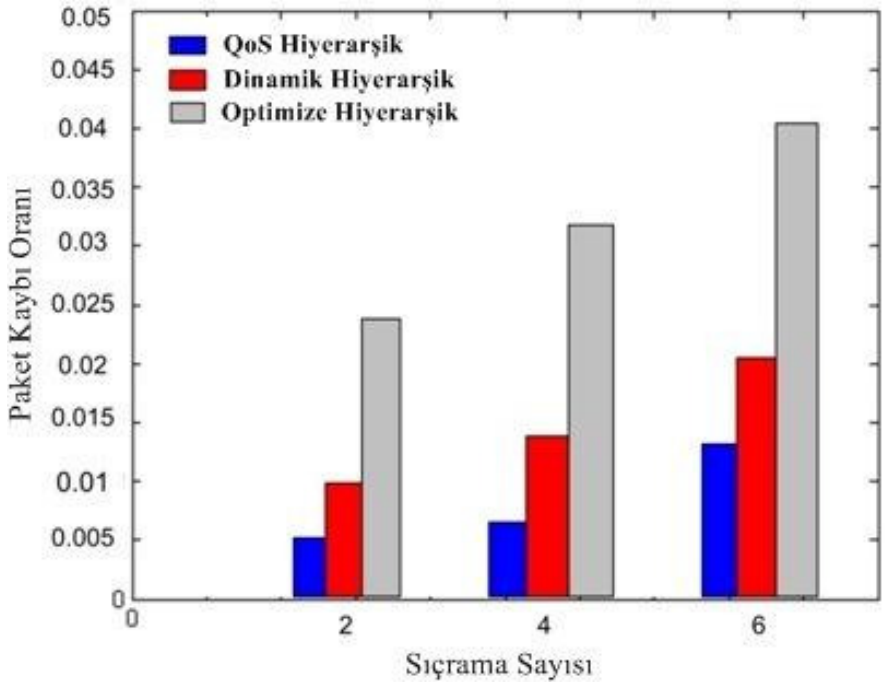
Rasgele olarak program tarafından çizilen yol izlendiğinde MN'nin sık sık HO yaptığı hareketlerinin rasgeleliğinden dolayı p-EU kararlarının sayısının çok yüksek miktarlarda olmadığı gözlemlenmiştir. Ancak p-EU gerçekleştirilebilen HO'larda başarı oranı yaklaşık %80 olarak elde edilmiştir.

n-EU ve p-EU sürelerini karşılaştırmak gerektiğinde ise tüm topolojilerde p-EU zamanın n-EU süresinden daha kısa olduğu gözlemlenmiştir. Süreden yana kazanılan bu avantaj MN'nin hızına ve topolojide AP'lerin yerleşimine göre farklılık göstermektedir. Hızlı hareket eden MN'lerde süreden kazanç %50-%60 aralığında olabiliyorken, MN'nin daha yavaş hareket ettiği topolojilerde bu süre kazancı %80-%85 aralığında oluşabilmektedir. Topoloji 5 gibi MN'nin rasgele olarak düzensiz bir hareket izlediği durumlarda bu süre kazancı daha yüksek değerlere de çıkabilmektedir.

4.3. MODELLENEN SİSTEM VE ALGORİTMANIN PERFORMANS İNCELEMESİ

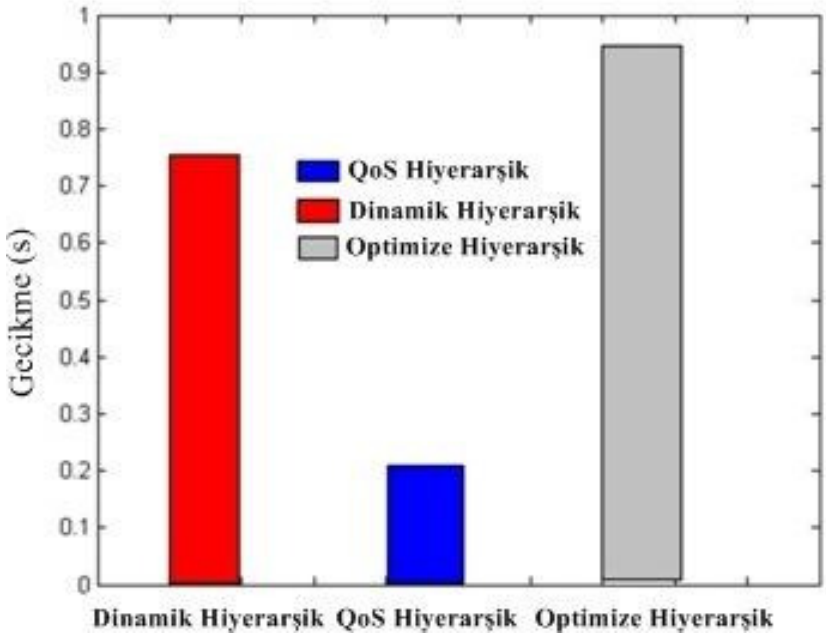
Algoritmamızın performans değerlendirmesi için simülasyon çalışmalarını OPNET simülasyon aracı ile gerçekleştirilmiştir. Bu denemelerde önerdiğimiz algoritmanın performansını, ağın paket kaybı oranı, gecikmesi ve veri iletimi maliyeti cinsinden incelenmiştir. Simülasyonlarda bir MN bir etki alanından diğerine hareket ederken sürmekte olan bir VoIP bağlantısına sahiptir. Önerdiğimiz algoritmanın simülasyon sonuçları RRU maliyeti bulmak için kullanılan iki farklı algoritma olan Dinamik Hiyerarşik (DH) (Yang ve diğ., 2007) ve Optimize edilmiş Hiyerarşik (OH) (Misra ve diğ., 2006) ile karşılaştırılmıştır.

Şekil 4.31, bir oturum boyunca ortalama paket kaybı oranını tüm yöntemler için sıçrama sayısının bir fonksiyonu olarak göstermektedir. Burada önerdiğimiz algoritmanın diğer algoritmalar ile karşılaştırılınca bir oturum boyunca paket kaybının azaldığı gözlemlenmektedir. Bunun sebebi, kaynakları veri ve sinyalleme mesajları cinsinden optimize etmeye çalışan, maliyete dayalı bir algoritma olmasıdır.



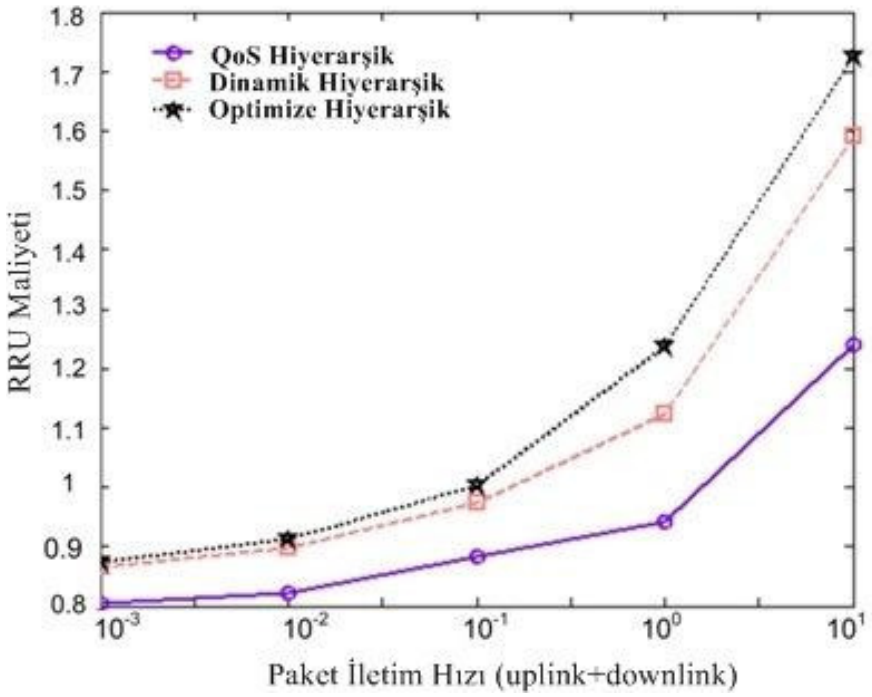
Şekil 4.31: Tüm algoritmalar için paket kaybı oranı ve sıçrama sayısı arasındaki ilişki

Değerlendirilen bir diğer karşılaştırma kriteri ise algoritmanın medyan iletim olasılığıdır. Bu değer bizim algoritmamız için 0.77, dinamik hiyerarşik algoritma için 0.7 ve optimize hiyerarşik algoritma için ise 0.49 değerindedir. .Bundan yola çıkarak elde edilen sonuç ise bizim algoritmamızın, DH ve OH algoritmalarının genellikle hatları sırasıyla %22 veya üstü, %31 veya üstü ve %53 veya üstü kayıp oranları ile kullanmakta olduğudur. Şekil 4.32 yer değiştirme gecikmesinin ölçümlerinde elde edilen sonuçları göstermektedir.



Şekil 4.32: Tüm algoritmalar için yer değiştirme gecikmeleri

Önerdiğimiz algoritmanın oldukça iyi sonuçlar vermektedir ki bu da önerdiğimiz yöntemin yer değiştirme boyunca hedefimiz olan gecikmedeki düşüşü sağladığını göstermektedir. Bu sonuca ulaşılmasının sebebi, maliyet açısından en avantajlı yolu seçerken uygulanan politikanın, özellikle toplam trafiği göz önünde bulundurarak veri mesajları maliyetini minimize etmeyi hesaba katarak mesajların transfer gecikmesini optimize etmesini sağlamasından kaynaklanmaktadır.



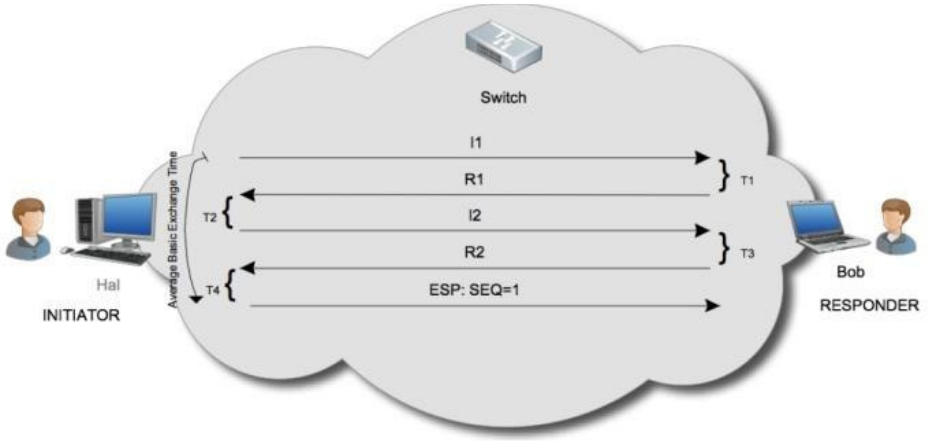
Şekil 4.33: Tüm algoritmalar için radyo kaynak kullanımı (RRU) ve trafik

Şekil 4.33'te önerdiğimiz algoritmanın minimal RRU maliyetini vererek en iyi sonuca vardığı gözlemlenmektedir. Veri iletim yükü ve kayıt güncelleme yükünün minimal olmasından dolayı bu sonuç bu şekilde oluşmaktadır. Bu iki yük arasındaki denge sistemin formülasyonundaki hedef fonksiyonumuz sayesinde başarılmıştır.

4.4. HIP TEST ORTAMI SONUÇLARI

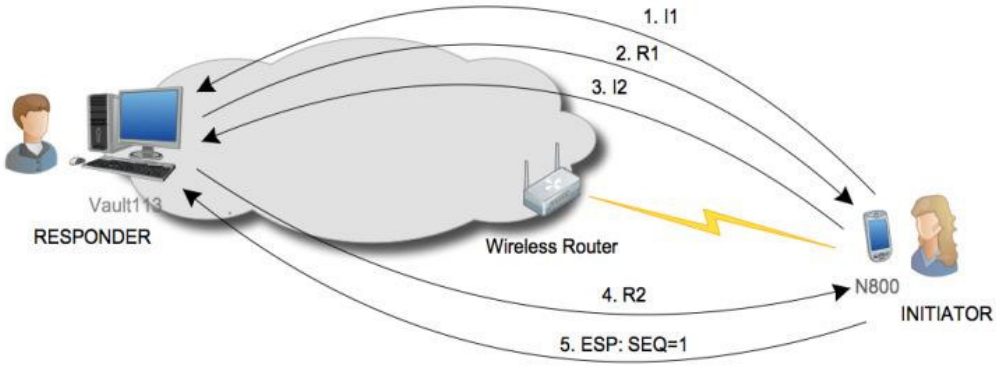
4.4.1. BE Testleri ve Sonuçları

Test ortamında yapılan testlerde öncelikle HIP protokolünün BE süreleri incelenmiştir. Belirtilen senaryolar üzeride BE prosedürüne ait dört temel mesajın (I1, R1, I2 ve R2) doğru akışı ve doğrulanması sağlanmıştır. Şekil 4.34, BE için test edilen ilk senaryoyu, Şekil 4.35, BE için test edilen ikinci senaryoyu göstermektedir.



Şekil 4.34: HIP Base Exchange Test Senaryosu 1 (Dizüstü bilgisayar Ethernet üzerinden bağlı)

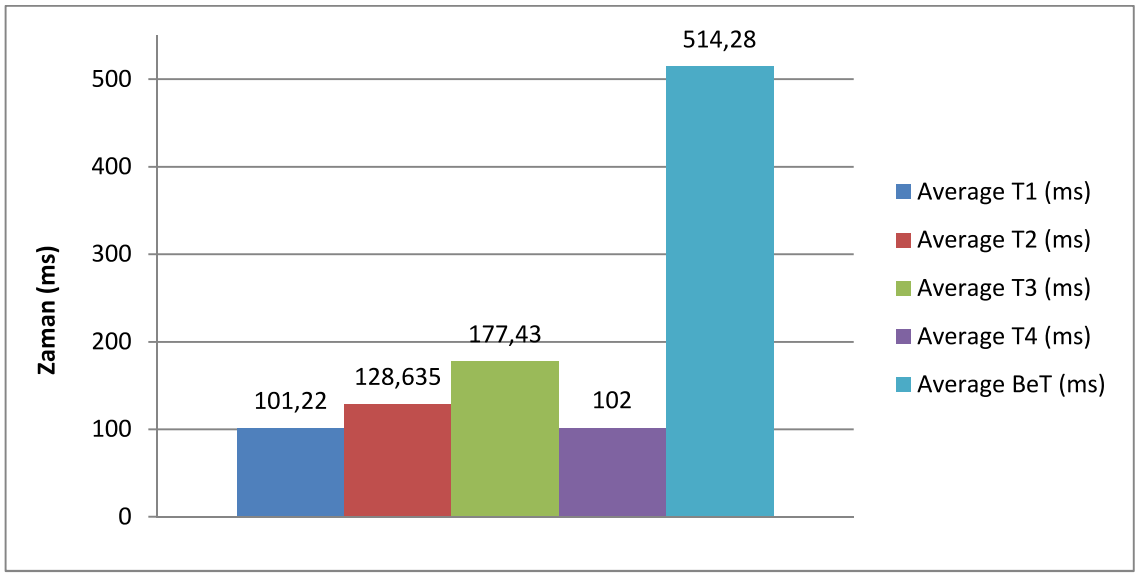
BE'nin akışı ve mesajların işleyiş süreleri şu şekilde tanımlanmaktadır. Dört adet farklı zaman tanımlaması kullanılmıştır. T1 zamanı R düğümünün I1 mesajını alması ve otomatik olarak R1 mesajını yollaması arasında geçen süre, T2 zamanı I düğümü tarafından mesajın alınması, bulmacanın çözülmesi ve R'ye geri yollanması arasında geçen süredir. T3 zamanı R düğümü tarafından bulmacanın cevabının alınması, kayıt talebinin incelenmesi ve I'ya R2 mesajı ile cevap vermesi gereken süredir. T4 zamanı ise I'nın R2 mesajını aldıktan sonra kayıt işlemini gerçekleştirmesi ve IPSec ESP paketlerini R'ye yollamak üzere oluşturduğu süredir.



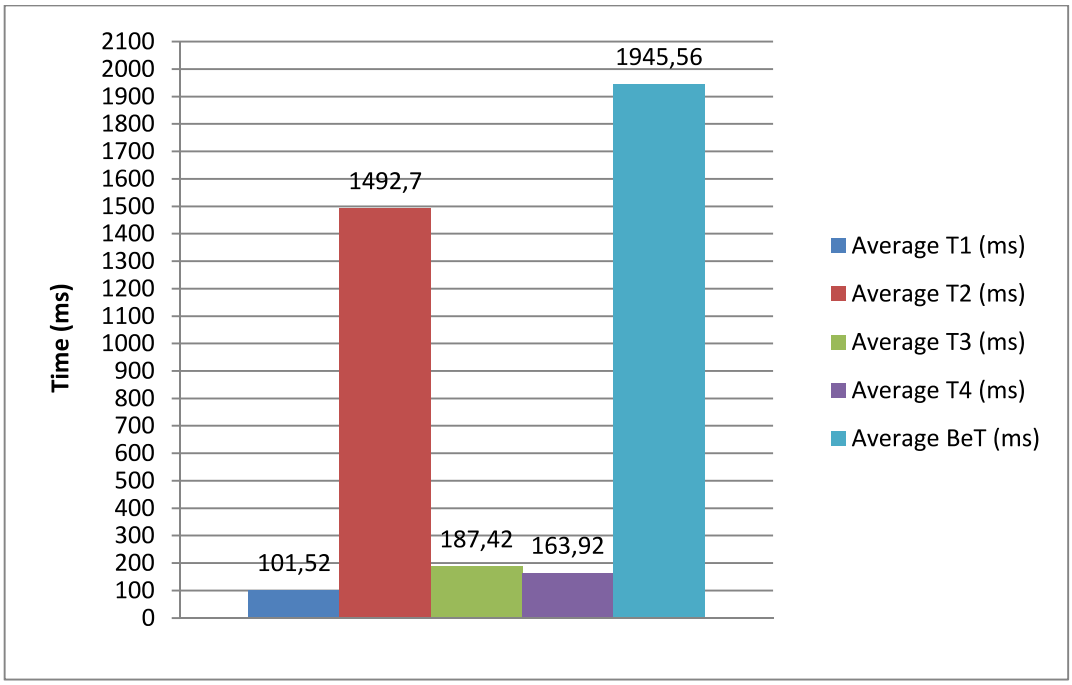
Şekil 4.35: HIP Base Exchange Test Senaryosu 2 (N800 WiFi 802.11g üzerinden bağlı)

Tüm bu değerler göz önünde bulundurulduğunda, T1,T2,T3 ve T4 değerleri için örnekler toplanmış ve BE prosesinin ortalama toplam süresi (BeT= T1+T2+T3+T4) incelenmiştir. Şekil 4.36 ve Şekil 4.37 ortalama olarak T1,T2,T3,T4 ve BeT zamanlarını gösteren sonuçları içermektedir. Her iki senaryoda da görülebileceği gibi en kısa süreler T1 ve T4 zamanlarına aittir ki bunlar önceden tanımlı I1 mesajının işlenmesine ve R2 mesajından sonra ESP Security Association (SA) durumunun ayarlanması için gereken sürelere karşılık gelmektedir. Sonuçlar arasındaki temel farklar mobil düğüm olan N800

için T2 süresinin farklılığından kaynaklanmaktadır. Bunun sebebi de bir ARM işlemciye sahip mobil düğümün gücü normal bir düğümün işlemcisi kadar güçlü olmamasıdır.

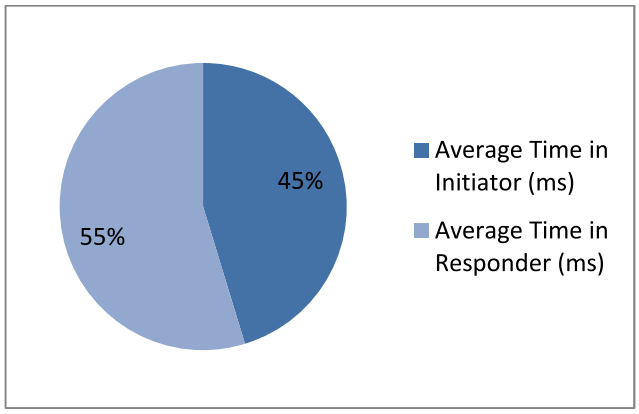


Şekil 4.36: HIP BE için Ortalama Süreler(Senaryo 1)

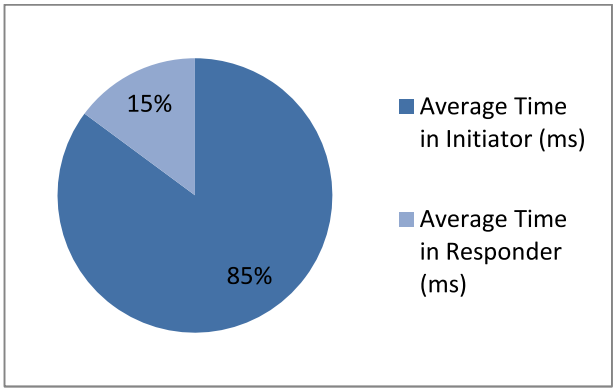


Şekil 4.37: HIP BE için Ortalama Süreler (Senaryo 2)

Şekil 4.38 ve 4.39'dan görülebildiği gibi, BE süresince harcanan zamanın ortalam yüzdelik payı çoğunlukla mobil düğümün donanımsal gücüne dayanmaktadır. Buna bağlı olarak, ilk test senaryosunda, T2 süresinin en fazla zamanı harcaması beklenirken (kriptografik bulmacanın çözüm zamanını içeren), host I %55'lik oranla daha fazla zaman harcadığı gözükmektedir. Bu durum %85'lik oranla ikinci senaryo için de gözükmektedir.



Şekil 4.38: Senaryo 1 için I ve R düğümlerinin HIP BE için harcanan zamanın ortalama yüzdelik değerleri (Dizüstü bilgisayar Ethernet üzerinden bağlı)

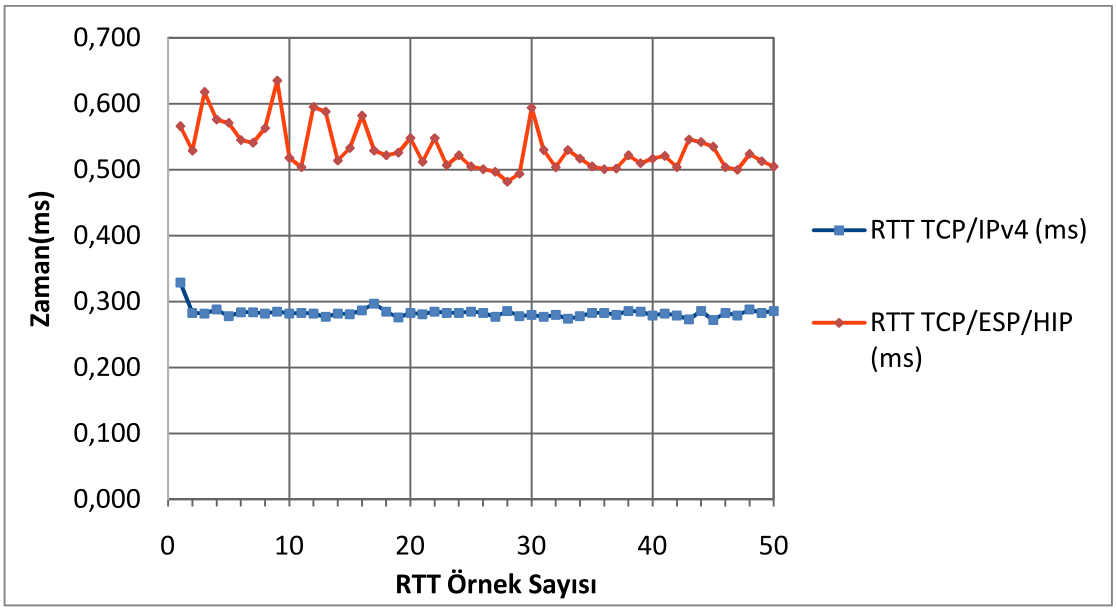


Şekil 4.39: Senaryo 2 için I ve R düğümlerinin HIP BE için harcanan zamanın ortalama yüzdelik değerleri (N800 WiFi 802.11g üzerinden bağlı)

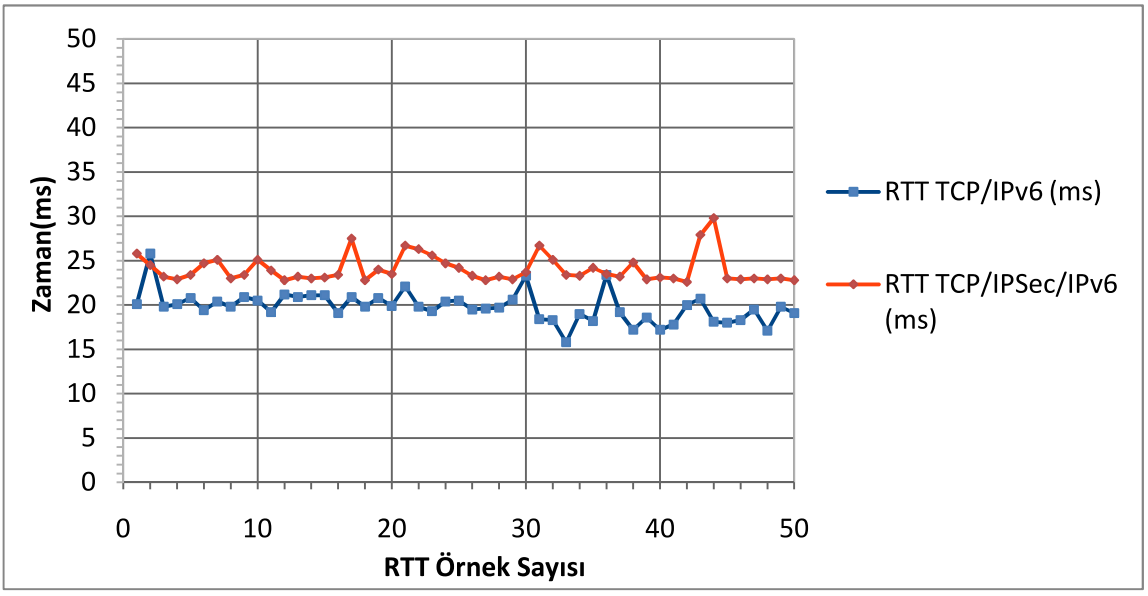
4.4.2. RTT Tahminleri

Ölçülen diğer değerlerden biri de IMPv4 ECHO/RESP ve IPv4 kapsülleme ve HIP kapsüllemesi üzeriden ESP işlemleri için RTT değerlerinin ölçümlerini içermektedir. Her durum için değerlendirilen örnek sayısı 50, her bir senaryoda elde edilen örnek sayısı ise toplam olarak 100'dür. Şekil 4.40 ve 4.41'de de görülebileceği gibi normal IP kapsüllemesi üzerinden yollanan ICMP mesajının ve ESP/HIP kapsüllemesi üzerinden yollanan ICMP mesajlarının RTT değerleri arasındaki fark her iki senaryo için de

görülebilmektedir. Her iki senaryoda da IP üzerinden yollanan mesaj için RTT değeri ESP/HIP üzerinden yollanan mesaj için RTT değerinden fazladır ve dolayısıyla bir ek yük ortaya çıkarmaktadır.



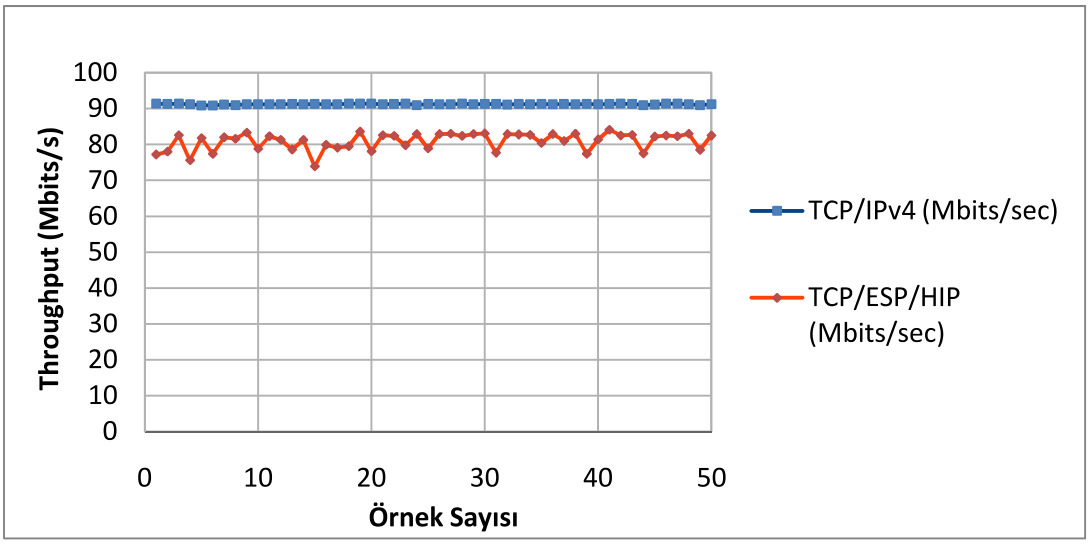
Şekil 4.40: Senaryo 1 için HIP RTT Değerleri (Dizüstü bilgisayar Ethernet üzerinden bağlı)



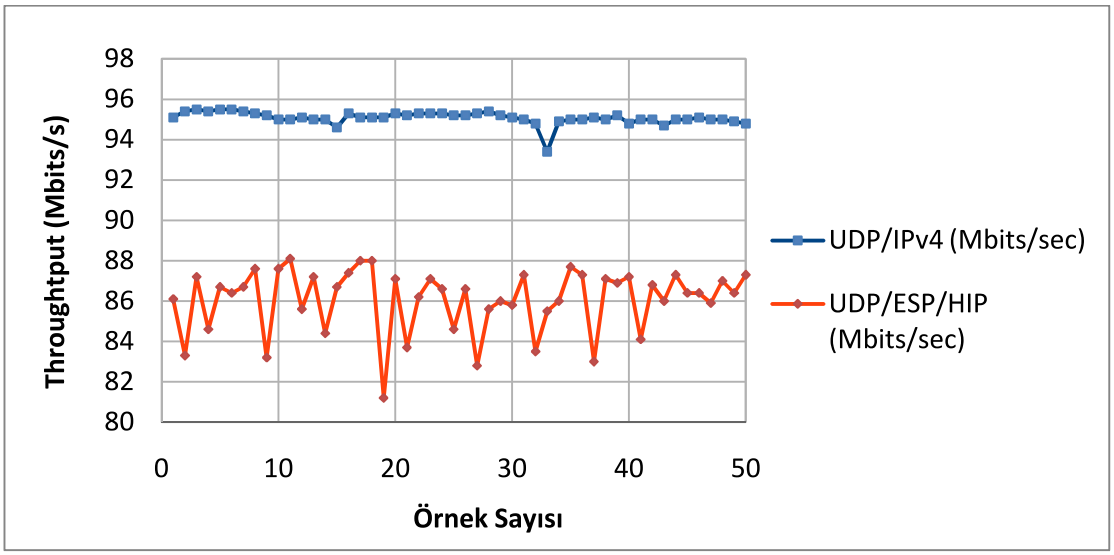
Şekil 4.41: Senaryo 2 için HIP RTT Değerleri (N800 WiFi 802.11g üzerinden bağlı)

4.4.3. Ağın Toplam Yükü (Throughput) Ölçümleri

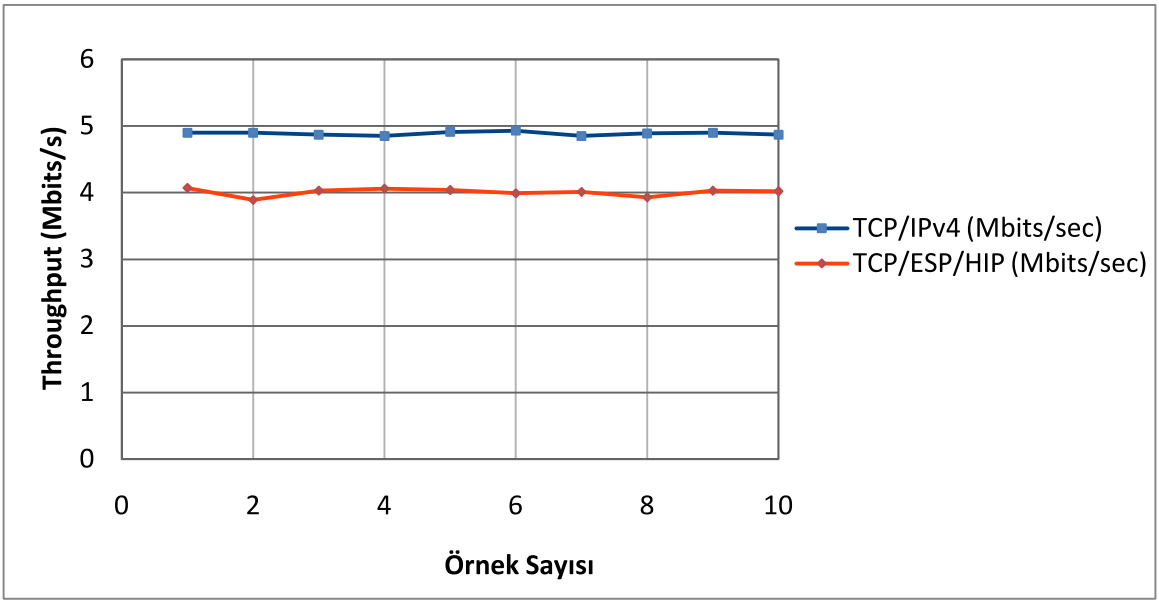
İki HIP düğümü arasındaki haberleşme süresince ortalama toplam yük (throughput) sonuçları da test edilmiştir. Her bir senaryo için 100 saniye boyunca I ve R arasında büyük boyutlu bir dosyanın iletimi süresince ortalama toplam yük değerleri elde edilmiştir. Hem TCP hem de UDP protokolü üzerinde IPv4 ve IPSec ESP/HIP kapsülleme modlarında sonuçlar elde edilmiştir. Elde edilen sonuçlardaki farklılıklar TCP ve UDP protokollerinin doğal olarak farklı işleyişlerinden kaynaklanmaktadır.



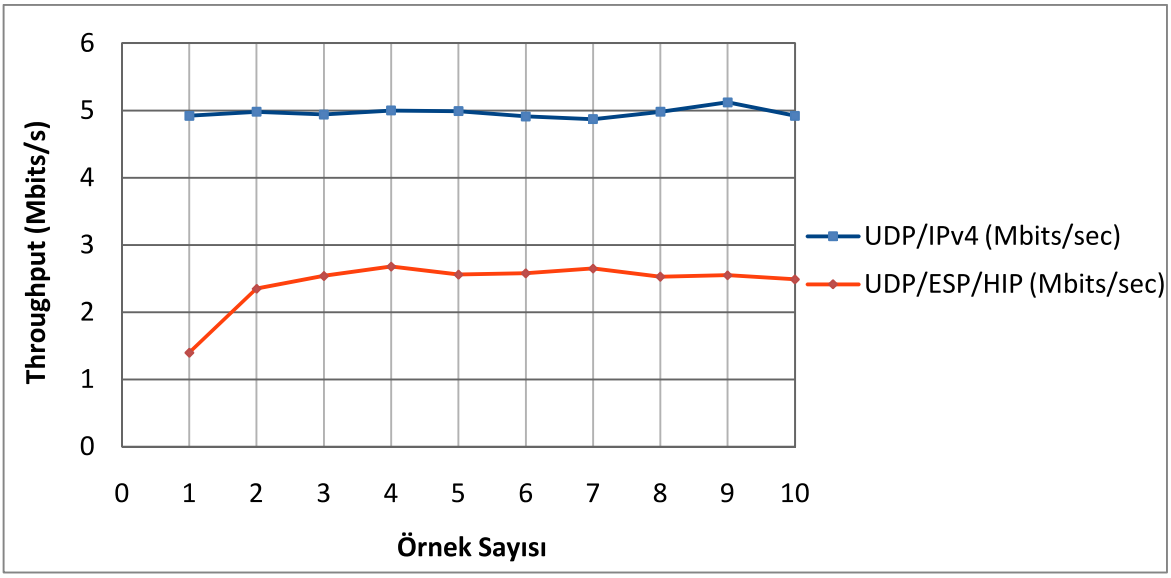
Şekil 4.42: HIP-TCP Throughput Test Senaryosu 1 (Sabit Düğüm)



Şekil 4.43: HIP-UDP Throughput Test Senaryosu 1 (Sabit Düğüm)



Şekil 4.44: HIP TCP Throughput Test Senaryosu 2 (N800 WiFi 802.11g üzerinden bağlı)

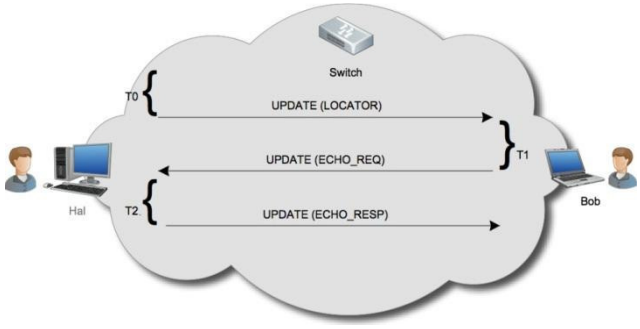


Şekil 4.45: HIP UDP Throughput Test Senaryosu 2 (N800 WiFi 802.11g üzerinden bağlı)

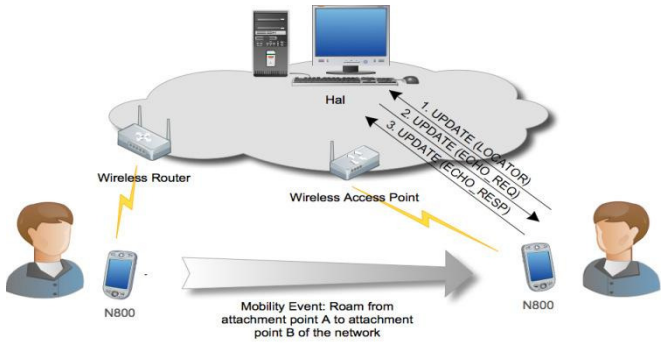
4.4.4. HIP Mobilite Olayları ve Sonuçları

Test ortamı üzerinde elde edilen sonuçlardan bir diğeri de HIP protokolü tarafından tanımlanan farklı UPDATE mesajlarının işlenme süreleri ile ilgili ölçümlerdir. Mesajların doğru akışının doğrulanması ve onaylanması için Şekil 4.46 ve 4.47'te gösterilen test senaryoları mobilite olaylarının test edilmesi için tasarlanmıştır. Senaryo

2'de, mobil HIP düğümünün (N800) ağa bağlı olduğu kablosuz yönlendirici yerine kablosuz erişim noktası veya bluetooth erişim noktası olarak değiştirilmiştir.



Şekil 4.46: HIP Mobilite Olayları için Test Senaryosu 1

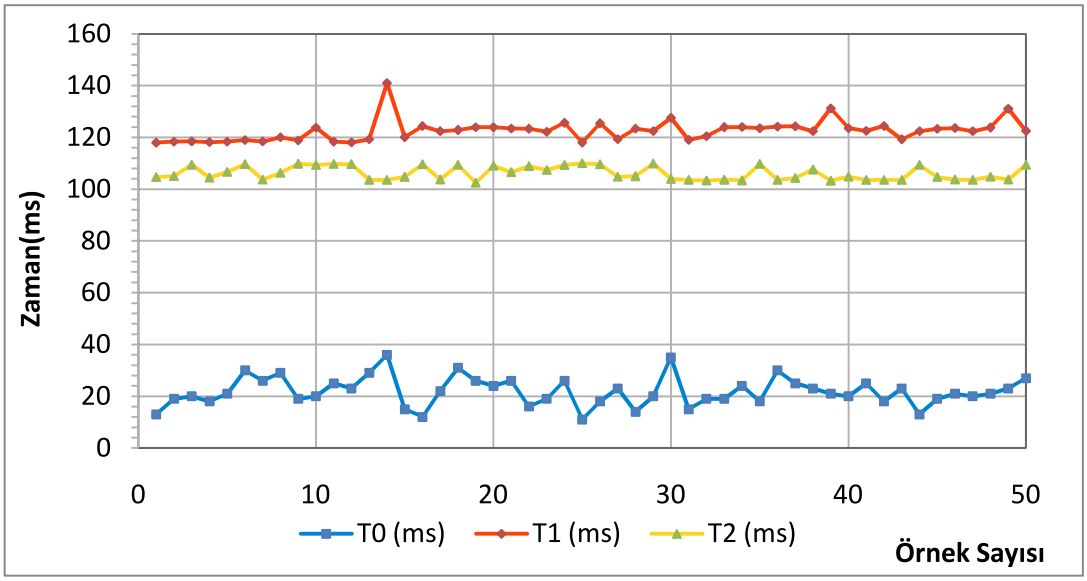


Şekil 4.47: HIP Mobilite Olayları için Test Senaryosu 2 (N800 WiFi 802.11g üzerinden bağlı)

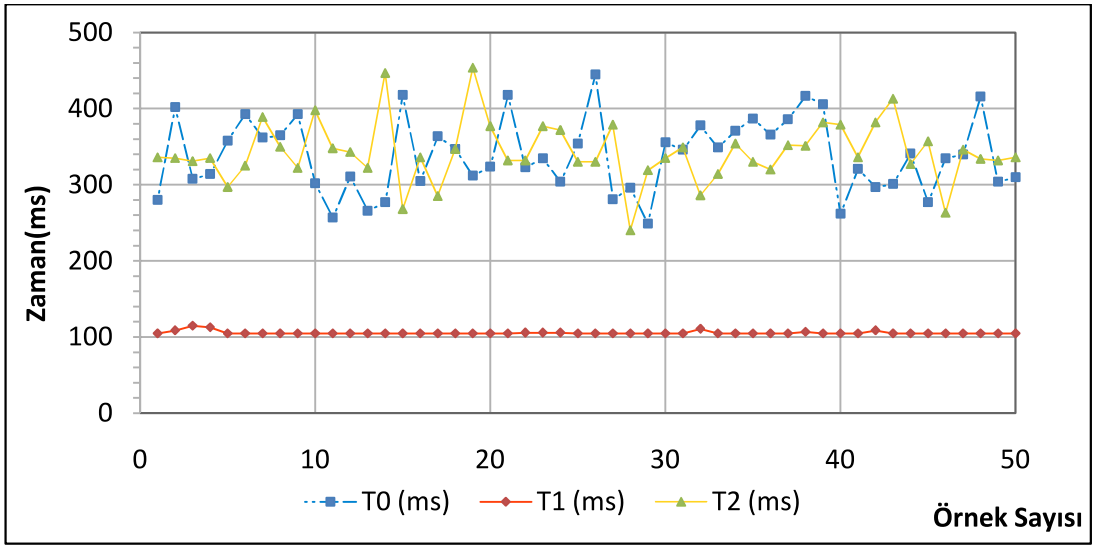
Test senaryosu 1 için, başlatan durumdaki düğümün (Bob) ağ ara yüzünü değiştirmesi için bir kod parçası yeterli olabilecekken, daha gerçeğe uygun bir davranış biçimi olarak fiziksel olarak ağ ara yüzünden kopulduğu ve tekrar bağlanılmasından önce (aynı erişim yönlendiricisindeki ikinci bir bağlantı noktası) belirli bir süre beklendiği durum tercih edilmiştir. Mobilite olaylarının incelendiği testlerde üç adet zaman ölçümü ve süreleri göz önünde bulundurulmuştur. T0 zamanı, I düğümünün mevcut konumunun değiştiğini algılaması, LOCATOR parametresini yeni IP adresi ile değiştirmesi, UPDATE paketini uygun kaynak ve hedef HIT değerleri ile oluşturması ve mesajı süren bir haberleşmesinin bulunduğu tüm HIP düğümlerine göndermesi için geçen süredir.

T1 zamanı, R düğümü tarafından UPDATE mesajının işlenmesi ve bu UPDATE mesajına rastgele bir veri için ECHO(cevap-ses verme) talebi bulunması için geçen süredir. EXHO talebi ile R düğümü, I'nın erişilebilirliğini test etmektedir. T2 zamanı ise, I düğümünün ECHO talebine R tarafından talep edilen veriyi içeren bir UPDATE mesajı ile cevap vermesi için geçen süredir. Bu bölümde yapılan testler ile LOCATOR

parametresinin güncellenmesinin her iki senaryo için HIP mimarisinde ve mobilite eklentilerinde olduğu şekilde yapıldığı test edilmiştir. Şekil 4.48 ve 4.49, Senaryo 1 ve Senaryo 2'de her bir düğüm tarafından (Hal ve Bob) UPDATE paketinin işleniş sürelerini göstermektedir.

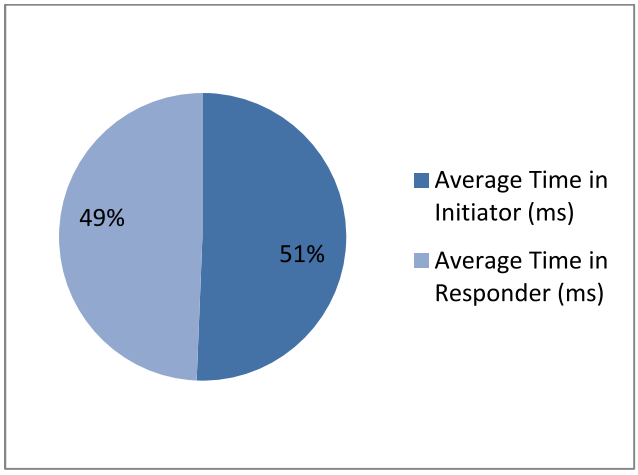


Şekil 4.48: Senaryo 1'de HIP Mobilite Olayları Harcanan Zaman Değerleri (Dizüstü bilgisayar Ethernet üzerinden bağlı)

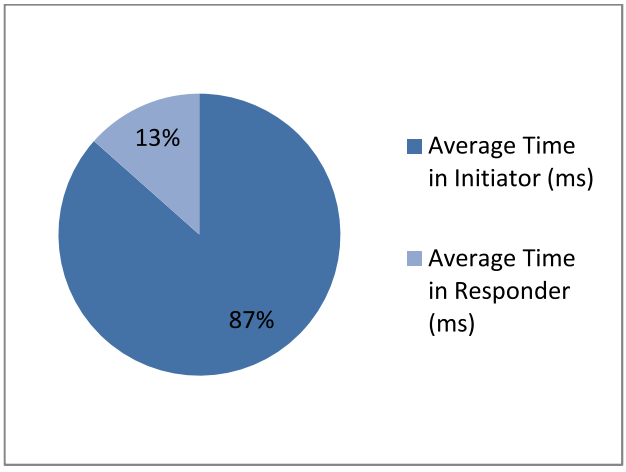


Şekil 4.49: Senaryo 2'de HIP Mobilite Olayları Harcanan Zaman Değerleri (N800 WiFi 802.11g üzerinden bağlı)

Şekil 4.48'de sabit bir düğümün yeni konumunu fark etmesi, mevcut LOCATOR parametresini haberdar etmesi ve diğer uç düğümleri bu değişikliklerden haberdar etmesi için geçen T0 süresinin değeri (ortalama yaklaşık 21 ms) görülebilmektedir. Ancak senaryo 2'nin sonuçlarından da görülebileceği gibi, düşük güce sahip olan mobil cihazın UPDATE işlemleri için geçen süre sabit düğümden elde edilen sonuçlara göre oldukça uzundur. Şekil 4.50 ve 4.51, Her iki senaryo için UPDATE prosedürü boyunca harcanan zamanların I ve R açısından ortalama zamanın yüzdelik değerlerini göstermektedir.



Şekil 4.50: Senaryo 1 için I ve R düğümlerinin HIP Mobilite Olayları için harcanan zamanın ortalama yüzdelik değerleri (Dizüstü bilgisayar Ethernet üzerinden bağlı)



Şekil 4.51: Senaryo 2 için I ve R düğümlerinin HIP Mobilite Olayları için harcanan zamanın ortalama yüzdelik değerleri (N800 WiFi 802.11g üzerinden bağlı)

5. TARTIŞMA VE SONUÇ

Günümüzde gelişen heterojen ağ ortamlarında yeni akım, özellikle TCP/IP mimarisin üst katmanlarında oturum tanımlama ve kimlik tanımlama kavramlarının ayrılması üzerinedir. Bu iki kavrama göre günümüze kadar IP adreslerinin konum ve kimlik tanımlama olmak üzere iki rolü bulunmaktadır. Bu iki rolün ayrılması ile oturumlar IP adresi yerine, bir düğümü benzersiz şekilde tanımlayan yeni bir kimlik belirleyiciye göre tanımlanmaktadır. Böylece, IP adresi değişikliği ile uygulama oturumlarının zarar görmesi önlenmektedir.

Bu tez çalışmasında bu amaçla günümüzde önerilen en kapsamlı çalışmalardan HIP protokolü üzerinde yeni bir mobilite yönetimi mekanizması geliştirilmiştir. eHIP olarak adlandırılan bu yöntem, mevcut HIP tabanlı mobilite yönetimi mekanizmalarından farklı olarak yer değiştirme esnasında erken güncelleme işlemlerine karar verilmesi ve tetiklenmesi esasına dayanmaktadır.

eHIP'i mevcut yöntemlerle kıyaslamak söz konusu olduğunda, mHIP'de, tüm ağ elemanlarına mHIP birimi adı verilmektedir. Bu elemanlar alanlar arası yer değiştirme sırasında mesajları işleyebilmekte ve iletişimi yönlendirebilmektedir. Herhangi bir mHIP birimi yer değiştirme süresinde bir paketi yakalayarak bir CN gibi davranabilir. mHIP ağ geçidi ise ağda mHIP yönlendiricileri yöneten bir kök eleman gibi yer alır. Ağda aynı zamanda tek bir RVS'de bulunmaktadır. HIP'in olağan RVS eklentisindeki gibi, mobil düğümlerde kendilerini mHIP ağ geçiş yollarına kayıt ettirmek zorundadır. mHIP ağ geçidinin rolü, eHIP'deki RVS_i'lerin rolüne benzerdir. Bir mHIP alanında, mHIP birimleri arasındaki bağlantılardan dolayı bir şekilde hiyerarşik olarak da adlandırılabilir. Ancak eHIP mimarisinde, tüm hiyerarşik elemanlar (RVS_i), bir RVS'nin rolünü tamamen üstlenebilmekte ve yerine getirebilmektedir.

eHIP mimarisi, mHIP'den ziyade DH-HIP mimarisi ile hiyerarşik açıdan benzerlik göstermektedir. DH-HIP ağı bir çeşit yerel RVS'ler tarafından yönetilen alt ağlara

bölmektedir. Ancak eHIP'de alt ağların boyutu DH-HIP'de olduğu gibi dinamik olarak değişmemektedir. Alan boyutunun optimizasyonu ise eHIP mimarisinde ölçeklenebilirlik açısından çözülebilecek bir problem olarak tanımlanabilir. eHIP'in diğer hiyerarşik tabanlı yaklaşımlardan önemli bir farkı da bir mobil düğümün ziyaret ettiği ağdaki tüm RVS'lere proaktif olarak kayıtlı olması ve bu kayıt işlemini aktiflik veya pasiflik durumunu belirterek gerçekleştirmesidir.

Bir mobil düğümün yeni konumunu güncellemesi fikri, temelde Fast Handovers for MIP (FMIP)'in hızlı yer değiştirme yöntemine ve yer değiştirmenin sezinlenmesine dayanmaktadır. FMIP'de yer değiştirmenin başlatılması bağlantı katmanından gelen (L2) tetikleyicilere dayanmaktadır ve bu tetikleyiciler aynı zamanda MIPv6'da sezinlemeli yer değiştirmeyi başlatmak için kullanılmaktadır. FMIP'de, mobil düğüm eski erişim yönlendiricisine "solicitation" mesajları yollar ve yeni konumu hakkında bilgilendirir. eHIP'de ise yeni bir konuma hareket etmek isteyen bir mobil düğümün, bir sonraki $nRVS_2$'nin ve nAP'nin IP adresi bilgilerini öğrendiği varsayılmaktadır. Daha önce de belirtildiği gibi, güncelleme işlemindeki gecikmeleri önlemek için tüm RVS'lere proaktif olarak kayıtlı olmak durumundadır. eHIP'de, mobil düğüm ilk EU mesajını o an bağlı bulunduğu $oRVS_2$'ye yollar ve güncelleme işlemini, daha önce pasif olarak kayıtlı olduğu $nRVS_2$ için başlatır.

Klasik HIP mimarisinde farklı olarak, RVS'ler ile ilgili RA mesajı ile MN'nin bağlı olduğu RVS'nin önceden mutlaka biliniyor olması durumu ortadan kaldırılmaktadır. Sistemde bir nevi otomatik RVS ataması yapılmaktadır. Bu sayede çöken bir RVS'nin olması durumunda sistemin stabilizasyonu bozulmamaktadır.

eHIP'de erken güncelleme prosedürünün tetiklenmesi, bir mobil düğümün hareketi esnasında yöneldiği bir diğer erişim noktasından RA mesajı alması üzerine meydana gelmektedir. RA mesajını almak ise ancak diğer AP'nin kapsama alanına girilmeye başlanması ile söz konusu olmaktadır. EU işlemlerinin, mobil düğüm iki AP arasındaki kesişen bölgeden ayrılmadan sonlanması hedeflenmiştir. Tasarlanan tahmin mekanizmasının amacı ise erken güncelleme mekanizmasının bu kesişen alana dahi girilmeden önce tetiklenmesi fikrine dayanmaktadır. Bu erken karar sayesinde mobil

düğüm için güncelleme mekanizmalarının tamamlanma süresi ve dolayısıyla yer değiştirme süresi ve gecikmesi indirgenmiştir.

eHIP ile ilgili simülasyon sonuçları ve performans incelemeleri bölüm 4.1.3'de belirtilmiştir. Bu bölümde elde edilen sonuçlarda genel olarak bakıldığında eHIP mekanizmasının önerilmesinde temel amaç olan HO gecikmesinin azaltılması hedefinin sağlandığı gözlemlenmektedir. Toplam HIP mesajları bakımından sonuçlar bölüm 4.1.3.1'de gösterilmektedir. Hiyerarşik HIP ve klasik HIP mimarilerine göre en büyük avantajı ağırlıklı olarak en alt seviye RVS'lerde gerçekleştirilen mesaj değişimleri ile özellikle yer değiştirme esnasında MN ve CN arasındaki mesajlaşma yükünün azaltılmasıdır. Aynı zamanda klasik HIP mimarisinde yoğun bir mesaj yüküne sahip olan RVS_0'ın mesaj yükü diğer alt seviye RVS'ler arasında dağıtılarak hiyerarşik yapının avantajı gözlemlenmiştir.

Bölüm 4.1.3.2'de gösterilen HO süreleri açısından eHIP'in performansı ise bu çalışmada önerilen ana yöntemin başarımını belirgin şekilde göstermektedir. eHIP yöntemini önerirken ana hedefimiz olan yer değiştirme süresinin klasik HIP ve hiyerarşik HIP mimarilerine göre azaltıldığını gösteren sonuçlar elde edilmiştir. HO sürelerindeki belirgin fark erken güncelleme mekanizmasının başarılı şekilde gerçeklenmesinden ve MN'nin geçiş yaptığı AP'ye olan bağlantısını gerçekleştirmek için klasik HIP'ten farklı olarak UPDATE prosedürünü yeni IP adresini almadan önce gerçekleştirmesinden kaynaklanmaktadır.

Bölüm 4.1.3 altındaki diğer başlıklarda, eHIP yönteminin kendi içerisindeki işlem süreçleri incelenmiştir. Yine HIP'in üst katmanında üretilen TCP ve UDP trafiklerine bağlı olarak bir ağın karakteristik özelliklerini yansıtan kriterlerden jitter ve RTT incelenmiştir ve eHIP'in sağladığı avantajlar gözlemlenmiştir.

eHIP prosedüründe RA mesajlarının elde edilmesine dayalı olarak gerçekleştirilen erken güncelleme prosedürün MN'nin hareketine ve izlediği yola bağlı olarak iyileştirmeye yönelik olarak önerilen p-eHIP yöntemi ile ilgili nümerik sonuçlar bölüm 4.2.2 altında incelenmiştir. Bu yöntemin testi için beş farklı örnek topoloji üzerinde elde edilen sonuçlar değerlendirilmiştir. Önerilen yöntemin MN'nin hareketi esnasında gösterdiği

tutarlılık veya değişkenlik durumu göz önünde bulundurulduğunda çoğunlukla %60-%80 aralığında bir başarı oranı ile EU'dan önce tahmine dayalı p-EU yapabildiğimizi gözlemlemiş bulunmaktayız.

Tez çalışmasının yurtdışında sürdürülen bölümlerinde gerçekleştirilmiş bir çalışma olan sistemin modellenmesine bağlı olarak yol bulma algoritması tasarımı bölüm 4.3'te incelenmiştir. Bu kısımda eHIP'in hiyerarşik mimarisi göz önünde bulundurularak ağın bir örgü ağ tipinde olduğu örnek durum incelenmiş ve sistem modellenmiştir. Elde edilen sonuçlar tasarlanan algoritmanın özellikle RRU maliyeti ve yer değiştirme gecikmesi açısından diğer yöntemlere göre daha başarılı olduğunu ortaya koymaktadır.

Bölüm 4.4'te ise tez çalışmasında incelenen HIP protokolü ile ilgili olarak gerçek test ve ağ ortamında gerçekleştirilen testlere ve ölçümlere yer verilerek bunların literatürde olan sonuçlar ile tutarlılığı incelemiştir. Aynı zamanda çalışmanın diğer kısımlarında önerilen yeni yöntemlerin tasarlanma aşamasında protokolün karakteri hakkında destekleyici sonuçlar elde edilmiştir.

Bu çalışmada, IP adreslerinin konum ve kimlik tanımla rollerinin ayrılması fikrine dayanan ve son yıllarda yoğun olarak yürütülen çalışmalar doğrultusunda yola çıkılmış ve bir yer değiştirme yönetimi yöntemi önerilmiştir. Fikir olarak bu mantığı kullanan başka protokoller üzerinde de gerçekleştirilebilecek olan önerilen erken güncelleme yöntemi, son yıllarda bu fikri benimseyen yeni protokollerden HIP protokolü üzerinde tasarlanmıştır. Bu çalışmada önerilen ve özellikle mikro mobilite yönetimi konusu kapsamına giren yöntemimiz,. HIP HIP için erken güncelleme fikrini benimseyen ilk yöntemdir için mevcut diğer yöntemlerin tam anlamıyla performans incelemeleri bulunmamaktadır. Özellike eHIP'den önce önerilen çalışmalarda ağda kullanılmak üzere yeni ağ elemanları tanımlanmış ve bunlara hem normal HIP haberleşmesinde hem de yer değiştirme süresince yeni görevler yüklenmiştir. Bu yöntemlerinin bazılarında özellikle HIP haberleşmesinde önemli rol oynan BE prosedüründe değişiklik yapılmıştır. eHIP yöntemi BE prosedüründe herhangi bir yapısal değişiklik gerektirmemektedir.

Bu konuyla bağlantılı olarak yapılabilecek çalışmalar arasında eHIP prosedürünün çoklu konumlu düğümlerden oluşan versiyonunun tasarlanması üzerinde çalışılabilir. Aynı zamanda HIP için farklı teknojiler arasında daha aktif ve hızlı şekilde yapılabilecek yer değiştirme prosedürü üzerinden çalışabilir. Aynı zamanda, sistemin analitik olarak modellemesi gerçekleştirilebilir. eHIP yöntemini geliştirmek adına ise yer değiştirme yönetimine kablosuz sensör ağlar veya RFID gibi teknojilerden faydalanıp faydalanılmayacağı incelenebilir. Yine son zamanlarda güncel olarak üzeirnde çalışmalar yapılan Proksi Mobil IP protokolü entegrasyonu ve proksi fikrinin HIP protokolünde kullanımı incelenebilir.

KAYNAKLAR

AKYILDIZ I.F., MCNAIR, J.,HO, J.S.M.,UZUNALİOGLU, H.,WENYE WANG, 1999, Mobility Management for Next Generation Wireless Systems, *Proceedings of IEEE*, vol. 87, no. 8, Aug. 1999, pp. 1347–84.

AKYILDIZ I.F.,XIE J.,MOHANTY S., 2004, A Survey Of Mobility Management In Next-Generation All-IP-Based Wireless Systems, *IEEE Wireless Communications*

ARREZ, L., CHAOUCHI H.,GURKAS AYDIN, Z., Performance Evaluation and Experiments for Host Identity Protocol, *International Journal of Computer Science Issues, Accepted for Volume 8, Issue 2, March 2011*

BOKOR L., NOVÁCZKI, S., IMRE S., 2007, A Complete HIP Based Framework For Secure Micromobility, *MoMM2007, Jakarta, Indonesia,* 111-122

BOKOR, L., NOVÁCZKI, S., TAMÁSL, L. Z., JENEY, G., 2009a, Design and Evaluation of Host Identity Protocol (HIP) Simulation Framework for INET/OMNeT++", *MSWIM 2009, Tenerife, Canary Islands, Spain,* 124-133

BOKOR, L., TAMÁS, L. Z., NOVÁCZKI, S., JENEY, G., 2009b, Protocol Design and Analysis of a HIP-based Per-Application Mobility Management Platform, *MobiWAC 2009, Tenerife, Canary Islands, Spain*, 7-16

CAMPBELL, A.T. GOMEZ, J. SANGHYO KIM CHIEH-YIH WAN TURANYI, Z.R. VALKO, A.G. ,2004, Comparison of IP Micromobility Protocols, *IEEE Wireless Communications,,* Vol.9, 72-82

GURKAS AYDIN, Z., CHAOUCHI, H.,ZAIM, A.H., 2009, eHIP : Early Update for Host Identity Protocol,*ACM Mobility Conference 2009, Article No.: 55, September, Nice,France*.

GURKAS AYDIN, Z., YAHİYA, T.A., CHAOUCHI, H. A., ZAİM H., 2010, QoS Mobility-Aware Algorithm using Early Update for Host Identity Protocol , *IEEE PIMRC 2010, September 2010, Istanbul*, 2014-2018

GURKAS AYDIN, Z., CHAOUCHI, H.,ZAIM, A.H., 2010, A Survey On Micro Mobility Management Of Host Identity Protocol, *Journal of Electrical & Electronics Engineering (IU – JEEE), 2011 (Accepted)*

GURTOV, A. 2008, *Host Identity Protocol (HIP)-Towards the Secure Mobile Internet*, Wiley Publications, ISBN: 978-0-470-99790-1

HIPSIM++, 2011, *HIPSim++: A Host Identity Protocol (HIP) Simulation Framework for INET/OMNeT++* [online], http://www.ict-optimix.eu/index.php/HIPSim

HOBAYA, F., GAY, V. ROBERT, E., 2009, Host Identity Protocol Extension Supporting Simultaneous End-host Mobility, *IWCMC 2009, Cannes-La Bocca*, 261-266

HON SO J. Y., WANG J., 2008, Micro-HIP : A HIP-based Micro-Mobility Solution, *Proceedings of ICC 2008 Workshop*, *Beijing*, 430 – 435

HU, B., YUAN, T., HU, Z., CHEN, S., 2010, L-HIP : A Localized Mobility Management Extension for Host Identity Protocol, *WiCOM 2010, Chengdu,* 1-4

IAPICHINO, G., BONNET, C., 2009, Host Identity Protocol and Proxy Mobile IPv6: A Secure Global and Localized Mobility Management Scheme for Multihomed Mobile Nodes, *GLOBECOM 2009, Honolulu*

IETF, 2011, *Internet Engineering Task Force* [online], http://www.ietf.org/

INET, 2011, *The INET Framework for OMNeT++* [online], http://www.omnetpp.org/doc/INET/neddoc/index.html

INFRAHIP, 2009, *Infrastructure for HIP Implementation [*online], http://infrahip.hiit.fi/

IRTF, 2011, *Internet Research Rask Force* [online], http://www.irtf.org/

JOKELA, P., RINTA-AHO, T., JOKIKYYNY, T., WALL, J., KUPARINEN, M., MAHKONEN, H., MELEN, J., KAUPPINEN, T., KORHONEN, J., 2004, Handover performance with HIP and MIPv6, *Wireless Communication Systems 2004, Mauritius,* 324 - 328.

JOKELA, P., MOSKOWITZ, R. 2008, RFC 5202, *Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP)*

KENT, S., 2005, RFC 4303, *IP Encapsulating Security Payload (ESP)*

KHURRI, A., VOROBYEVA, E., GURTOV, A., 2007, Performance of Host Identity Protocol on Lightweight Hardware, *ACM/IEEE MobiArch '07, New York, USA,*

KIM, H., KIM, Y., 2006, An Early Binding Fast Handover for High-Speed Mobile Nodes on MIPv6 over Connectionless Packet Radio Link, *SNPD'06, Las Vegas, NV,* 237

KOODLI, R., 2005, RFC 4068, *Fast Handovers for Mobile IPv6*

LAGANIER, J., KOPONEN, T., 2008a, RFC 5203, *Host Identity Protocol (HIP) Registration Extension*

LAGANIER, J., EGGERT, L., 2008b, RFC 5204, *Host Identity Protocol (HIP) Rendezvous Extension*

LANGAR, R., NIZAR, B., BOUTABAA, R. 2009, Mobility-Aware Clustering Algorithms with Interference Constraints in Wireless Mesh Networks, *Computer Networks*, Vol. 53, No. 1, 25 – 44

MATLAB, 2011, MATLAB: *The Language of Technical Computing* [online], http://www.mathworks.com/products/matlab/

MISRA, I. S., CHAKRABORTY, M., SAHA, D., MUKHERJEE,A., 2006, An Approach for Optimal Hierarchical Mobility Management Network Architecture, *IEEE VTC 2006, Melbourne, Vic,* 481–485.

MUSLAM, M.M., ANTHONY CHAN, H., VENTURA, N., 2009, HIP Based Micro-Mobility Management Optimization, IWCMC 2009, Cannes-La Bocca, 291-295

MOSKOWITZ, R., NIKANDER, P., HENDERSON, T., 2008, RFC 5201, *Host Identity Protocol*

MOSKOWITZ, R., NIKANDER, P., RFC 4423, 2006, *Host Identity Protocol (HIP) Architecture*

NIKANDER, P., 2008b, RFC 5205, *Host Identity Protocol (HIP) Domain Name System (DNS) Extension*

NIKANDER, P., HENDERSON, T., VOGT, C., ARKKO J., 2008a, RFC 5206, *End-Host Mobility and Multihoming with the Host Identity Protocol*

NOVÁCZKI S., BOKOR L., IMRE S., 2006, Micromobility Support in HIP: a survey and extension of host identity protocol, *IEEE MELECON, Benalmádena (Málaga), Spain* 651 – 654

NS-2, 2011, *The Network Simulator – ns-2* [online], http://nsnam.isi.edu/nsnam/index.php/Main_Page

OMNET, 2011, OMNeT++: *Network Simulation Platform* , http://www.omnetpp.org/ [online]

OPNET, 2011, *OPNET Technologies, Inc. Simulation Platform* [online], http://www.opnet.com

PERKINS, C., 2002, IP Mobility Support for IPv4, *Internet RFC, RFC 3220*

REINBOLD P., BONAVENTURE O., 2003, IP micro-mobility protocols, *IEEE Communications Surveys & Tutorials, Vol.5, 40-57*

SOLIMAN, H., CASTELLUCCIA, C., EL MALKI, K., BELLIER, L. 2005, RFC 4140, *Hierarchical Mobile IPv6 Mobility Management (HMIPv6)*

TOLEDO, N., HIGUERO, M.V., JACOB, E., MATIAS, J., 2009, Extending the Host Identity Protocol for Next Generation Wireless Networks, *IFIP WOCN 2009, Cairo-Egypt,* 1-5

YANG S. , QIN Y. , YANG D. , 2007, Dynamic Hierarchical Location Management Scheme for Host Identity Protocol, *Lecture Notes in Computer Science, Mobile Ad-Hoc and Sensor Networks, Springer Berlin / Heidelberg,* Volume 4864/2007 as proceedings of MSN 2007, Beijing, China

EKLER

EK-A INI DOSYALARI

HIP Senaryosu için Omnetpp.ini

```
network = inet.examples.hip.comp_trys.try_20110226_1.base.base1

#HIP Parameters.-----------------------------
**.RVS0_1.hip.OWN_HIT = "2001:1001:1001:1001:1001:1001:2222:1002"
**.RVS1_1.hip.OWN_HIT = "2001:1001:1001:1001:1001:1001:1111:1001"
**.RVS2_1.hip.OWN_HIT = "2001:2001:2001:2001:2001:2001:1001:1001"
**.RVS2_2.hip.OWN_HIT = "2001:2001:2001:2001:2001:2001:1001:1002"
**.RVS2_3.hip.OWN_HIT = "2001:2001:2001:2001:2001:2001:1001:1003"
**.RVS*.hip.PARTNER_HIT = ""

**.MN1.hip.OWN_HIT = "2001:1001:1001:1001:1001:1001:5555:1001"
**.MN1.hip.PARTNER_HIT = ""
**.MN1.hip.RVSAddr = "2001:1001:1001:1001:1001:1001:2222:1002"
**.MN1.hip.dnsAddress = "DNS"
**.MN1.hip.registerAtRVS = 1 s
**.MN1.hip.REG_StartTime = 10 s

**.CN1.hip.OWN_HIT = "2001:1001:1001:1001:1001:1001:6666:1001"
**.CN1.hip.PARTNER_HIT = ""
**.CN1.hip.RVSAddr = "2001:1001:1001:1001:1001:1001:2222:1002"
**.CN1.hip.dnsAddress = "DNS"
**.CN1.hip.registerAtRVS = 1 s
**.CN1.hip.REG_StartTime = 10 s
```

eHIP ve Hiyerarşik HIP Senaryoları için Omnetpp.ini

```
#HIP Parameters.-----------------------------
**.RVS0_1.hipEu.Child_RVSs = "2001:1001:1001:1001:1001:1001:1111:1001"
**.RVS0_1.hipEu.Neigbour_RVSs = ""
**.RVS0_1.hipEu.OWN_HIT = "2001:1001:1001:1001:1001:1001:2222:1002"
**.RVS0_1.hipEu.PARENT_RVS_HIT = ""
**.RVS0_1.hipEu.RVS_LEVEL = 2
**.RVS0_1.hipEu.ParentRvsName = ""
**.RVS0_1.hipEu.OwnRvsName = "RVS0_1"
**.RVS0_1.hipEu.ChildRvsNames = "RVS1_1"

**.RVS1_1.hipEu.Child_RVSs      =      "2001:2001:2001:2001:2001:2001:1001:1001
2001:2001:2001:2001:2001:2001:1001:1002
2001:2001:2001:2001:2001:2001:1001:1003"
**.RVS1_1.hipEu.Neigbour_RVSs = ""
**.RVS1_1.hipEu.OWN_HIT = "2001:1001:1001:1001:1001:1001:1111:1001"
**.RVS1_1.hipEu.PARENT_RVS_HIT = "2001:1001:1001:1001:1001:1001:2222:1002"
**.RVS1_1.hipEu.RVS_LEVEL = 1
**.RVS1_1.hipEu.ParentRvsName = "RVS0_1"
**.RVS1_1.hipEu.OwnRvsName = "RVS1_1"
**.RVS1_1.hipEu.ChildRvsNames = "RVS2_1 RVS2_2 RVS2_3"
**.RVS1_1.hipEu.NeigbourRvsNames = ""
```

```
**.RVS2_1.hipEu.Child_RVSs = ""
**.RVS2_1.hipEu.Neigbour_RVSs    =    "2001:2001:2001:2001:2001:2001:1001:1002
2001:2001:2001:2001:2001:2001:1001:1003"
**.RVS2_1.hipEu.OWN_HIT = "2001:2001:2001:2001:2001:2001:1001:1001"
**.RVS2_1.hipEu.PARENT_RVS_HIT = "2001:1001:1001:1001:1001:1001:1111:1001"
**.RVS2_1.hipEu.RVS_LEVEL = 0
**.RVS2_1.hipEu.ParentRvsName = "RVS1_1"
**.RVS2_1.hipEu.OwnRvsName = "RVS2_1"
**.RVS2_1.hipEu.ChildRvsNames = ""
**.RVS2_1.hipEu.NeigbourRvsNames = "RVS2_2 RVS2_3"

**.RVS2_2.hipEu.Child_RVSs = ""
**.RVS2_2.hipEu.Neigbour_RVSs    =    "2001:2001:2001:2001:2001:2001:1001:1001
2001:2001:2001:2001:2001:2001:1001:1003"
**.RVS2_2.hipEu.OWN_HIT = "2001:2001:2001:2001:2001:2001:1001:1002"
**.RVS2_2.hipEu.PARENT_RVS_HIT = "2001:1001:1001:1001:1001:1001:1111:1001"
**.RVS2_2.hipEu.RVS_LEVEL = 0
**.RVS2_2.hipEu.ParentRvsName = "RVS1_1"
**.RVS2_2.hipEu.OwnRvsName = "RVS2_2"
**.RVS2_2.hipEu.ChildRvsNames = ""
**.RVS2_2.hipEu.NeigbourRvsNames = "RVS2_1 RVS2_3"

**.RVS2_3.hipEu.Child_RVSs = ""
**.RVS2_3.hipEu.Neigbour_RVSs    =    "2001:2001:2001:2001:2001:2001:1001:1001
2001:2001:2001:2001:2001:2001:1001:1002"
**.RVS2_3.hipEu.OWN_HIT = "2001:2001:2001:2001:2001:2001:1001:1003"
**.RVS2_3.hipEu.PARENT_RVS_HIT = "2001:1001:1001:1001:1001:1001:1111:1001"
**.RVS2_3.hipEu.RVS_LEVEL = 0
**.RVS2_3.hipEu.ParentRvsName = "RVS1_1"
**.RVS2_3.hipEu.OwnRvsName = "RVS2_3"
**.RVS2_3.hipEu.ChildRvsNames = ""
**.RVS2_3.hipEu.NeigbourRvsNames = "RVS2_1 RVS2_2"

**.RVS*.hipEu.REG_StartTime = 10 s
**.RVS*.hipEu.dnsAddress = "DNS"

**.MN1.hipEu.OWN_HIT = "2001:1001:1001:1001:1001:1001:5555:1001"
**.MN1.hipEu.PARTNER_HIT = ""
**.MN1.hipEu.RVSAddr = "2001:2001:2001:2001:2001:2001:1001:1001"
**.MN1.hipEu.dnsAddress = "DNS"
**.MN1.hipEu.registerAtRVS = 0 s
**.MN1.hipEu.REG_StartTime = 10 s

**.CN1.hipEu.OWN_HIT = "2001:1001:1001:1001:1001:1001:6666:1001"
**.CN1.hipEu.PARTNER_HIT = ""
**.CN1.hipEu.RVSAddr = "2001:2001:2001:2001:2001:2001:1001:1003"
**.CN1.hipEu.dnsAddress = "DNS"
**.CN1.hipEu.registerAtRVS = 0 s
**.CN1.hipEu.REG_StartTime = 10 s
```

Tüm Senaryolar için ortak Common.ini

```
**.MN1.numTcpApps = 1
**.MN1.tcpAppType = "TCPSessionApp"
#**.MN1.tcpAppType = "TCPBasicClientApp"
**.MN1.tcpApp[0].port = 1001
#**.MN1.tcpApp[0].active = true
**.MN1.tcpApp[0].active = true
**.MN1.tcpApp[0].address = "2001:1001:1001:1001:1001:1001:5555:1001"
**.MN1.tcpApp[0].connectAddress = "2001:1001:1001:1001:1001:1001:6666:1001"
**.MN1.tcpApp[0].tOpen = 31 s
#**.MN1.tcpApp[0].tSend = 40 s
**.MN1.tcpApp[0].tClose = -1 s
#**.MN1.tcpApp[0].sendBytes = 100000000 B
```

```
#**.MN1.tcpApp[0].requestLength = 200 B
#**.MN1.tcpApp[0].replyLength = 100 MB

**.CN1.numTcpApps = 1
**.CN1.tcpAppType = "TCPSinkApp"
#**.CN1.tcpAppType = "TCPGenericSrvApp"
**.CN1.tcpApp[0].address = "2001:1001:1001:1001:1001:1001:6666:1001"

**.MN1.numUdpApps = 1
**.MN1.udpAppType = "UDPEchoStream"
**.MN1.udpApp[0].port = 2000
# 15 kbps :256B packet 0.13333 sec interval
#**.MN1.udpApp[0].waitInterval = 0.133333 s
**.MN1.udpApp[0].packetLength = 256 B
**.MN1.udpApp[0].destPort = 2001
**.MN1.udpApp[0].startTime = 31 s
**.MN1.udpApp[0].destAddress = "2001:1001:1001:1001:1001:1001:6666:1001"
**.CN1.numUdpApps = 1
**.CN1.udpAppType = "UDPEchoApp"
**.CN1.udpApp[0].localPort = 2001
**.CN1.udpApp[0].destPort = 2000
**.CN1.udpApp[0].messageLength = 0
**.CN1.udpApp[0].messageFreq = 5 s

# mobility----------------------------------------
**.MN1.mobilityType = "RectangleMobility"
**.MN1.mobility.debug = false
**.MN1.mobility.x1 = 100
**.MN1.mobility.y1 = 100
**.MN1.mobility.x2 = 1200
**.MN1.mobility.y2 = 200
**.MN1.mobility.startPos = 0

**.MN1.mobility.updateInterval = 0.1 s
#-------------------------------------------

num-rngs = 2
sim-time-limit = 10000s
warmup-period = 300s
tkenv-default-config = General
repeat = 10

**.gen[*].rng-0 = 1

record-eventlog = false
debug-on-errors = true
cmdenv-express-mode = true

tkenv-plugin-path = ../../../Etc/plugins

#-- Event Log Recording Settings ----------------------
**.MN1.**.module-eventlog-recording = true
**.CN*.**.module-eventlog-recording = true
**.RVS*.**.module-eventlog-recording = true
**.module-eventlog-recording = false
#-----------------------------------------------------------

##########################################################################
# Output vectors # scalars.
##########################################################################
#**.bytesSent.vector-recording = true
**.mac.**.vector-recording = false
**.mac[*].**.vector-recording = false
**.AP*.**.vector-recording = false
**.switch*.**.vector-recording = false
**.R_*.**.vector-recording = false
**.wlan.**.vector-recording = false
```

```
**.networkLayer.**.vector-recording = false
**.vector-recording = true
**.tcp.scalar-recording =  true
**.mac.**.scalar-recording = false
**.AP*.**.scalar-recording = false
**.switch*.**.scalar-recording = false
**.queue.**.scalar-recording = false
**.encap.**.scalar-recording = false
**.wlan.agent.**.scalar-recording = false
**.wlan.agent.L2_HO_DELAY.scalar-recording = true

**.R_1.networkLayer.neighbourDiscovery.HIP_RVS_Address                          =
"2001:2001:2001:2001:2001:2001:1001:1001"
**.R_2.networkLayer.neighbourDiscovery.HIP_RVS_Address                          =
"2001:2001:2001:2001:2001:2001:1001:1002"
**.R_3.networkLayer.neighbourDiscovery.HIP_RVS_Address                          =
"2001:2001:2001:2001:2001:2001:1001:1003"
**.R_1.networkLayer.neighbourDiscovery.HIP_RVS_Name = "RVS2_1"
**.R_2.networkLayer.neighbourDiscovery.HIP_RVS_Name = "RVS2_2"
**.R_3.networkLayer.neighbourDiscovery.HIP_RVS_Name = "RVS2_3"

#-----------------------------------------

# configurator
# = ================================================================
*.playgroundSizeX = 1400 #channel control
*.playgroundSizeY = 1000 #channel control

# channel physical parameters
*.channelcontrol.carrierFrequency = 2.4 GHz
*.channelcontrol.pMax = 2.0  mW
*.channelcontrol.sat = -82 dBm
*.channelcontrol.numChannels = 5

#Access Point Parameters.
**.AP1_1.wlan.mgmt.ssid = "AP1_1"
**.AP1_1.wlan.mac.address = "10:AA:00:00:01:01"
**.AP1_1.eth[0].address = "10:AE:00:00:01:01"
**.AP1_2.wlan.mgmt.ssid = "AP1_2"
**.AP1_2.wlan.mac.address = "10:AA:00:00:01:02"
**.AP1_2.eth[0].address = "10:AE:00:00:01:02"
**.AP2_1.wlan.mgmt.ssid = "AP2_1"
**.AP2_1.wlan.mac.address = "10:AA:00:00:02:01"
**.AP2_1.eth[0].address = "10:AE:00:00:02:01"
**.AP2_2.wlan.mgmt.ssid = "AP2_2"
**.AP2_2.wlan.mac.address = "10:AA:00:00:02:02"
**.AP2_2.eth[0].address = "10:AE:00:00:02:02"
**.AP3_1.wlan.mgmt.ssid = "AP3_1"
**.AP3_1.wlan.mac.address = "10:AA:00:00:03:01"
**.AP3_1.eth[0].address = "10:AE:00:00:03:01"
**.AP3_2.wlan.mgmt.ssid = "AP3_2"
**.AP3_2.wlan.mac.address = "10:AA:00:00:03:02"
**.AP3_2.eth[0].address = "10:AE:00:00:03:02"

#----------------------------------------
**.DNS.numUdpApps = 1
**.DNS.udpAppType = "DNSBase"
**.DNS.udpApp[*].dnsDataFile = "dns.xml"
**.DNS.udpApp[*].startTime = 5 s

**.RVS*.numTcpApps = 0
**.RVS*.tcpAppType = ""
**.RVS*.numUdpApps = 0
**.RVS*.udpAppType = ""
# 10us is the base
#**.R_2.networkLayer.ipv6.procDelay = 10ms
**.ipv6.procDelay = 10us
```

```
# Ethernet NIC configuration
**.eth[*].queue.frameCapacity = 10  # in routers

# wireless channels
**.AP1_1.wlan.radio.channelNumber = 1
**.AP1_2.wlan.radio.channelNumber = 2
**.AP2_1.wlan.radio.channelNumber = 3
**.AP2_2.wlan.radio.channelNumber = 4
**.AP3_1.wlan.radio.channelNumber = 1
**.AP3_2.wlan.radio.channelNumber = 2
**.MN1.wlan.radio.channelNumber = 0 # just initially -- it'll scan

# wireless configuration
**.wlan.agent.activeScan = true
**.wlan.agent.channelsToScan = ""  # "" means all
**.wlan.agent.probeDelay = 0.1 s
**.wlan.agent.minChannelTime = 0.15 s
**.wlan.agent.maxChannelTime = 0.3 s
**.wlan.agent.authenticationTimeout = 5 s
**.wlan.agent.associationTimeout = 5 s

# nic settings
**.mac.address = "auto"
**.mac.maxQueueSize = 14
**.mac.rtsThresholdBytes = 4000 B
**.mac.bitrate = 2e6 bps  # 2Mbps
**.wlan.mac.retryLimit = 7
**.wlan.mac.cwMinData = 7
**.wlan.mac.cwMinBroadcast = 31

**.radio.bitrate = 2E+6 bps #in bits/second
**.radio.transmitterPower = 2.0 mW  #[mW]
**.radio.thermalNoise = -110 dBm
**.radio.sensitivity = -82 mW
**.radio.snirThreshold = 4 dB # in dB

# relay unit configuration
**.relayUnitType = "MACRelayUnitNP"
**.relayUnit.addressTableSize = 100
**.relayUnit.agingTime = 120s
**.relayUnit.bufferSize = 1048576 B  # 1Mb
**.relayUnit.highWatermark = 524288 B  # 512K
**.relayUnit.pauseUnits = 300  # pause for 300*512 bit (19200 byte) time
**.relayUnit.addressTableFile = ""
**.relayUnit.numCPUs = 2
**.relayUnit.processingTime = 2us

#Access Point Shared Parameters.
**.AP*.eth[*].txrate = 100e6 bps
**.AP*.wlan.mgmt.beaconInterval = 0.1 s

#switch Shared Parameters.
**.switch*.mac[*].txrate = 100e6 bps

#ethernet Shared Parameters.
**.eth*.mac.txrate = 100e6 bps

# ip settings
**.routingTableFile = xmldoc("empty.xml")

[Config load1_5mps_ra1_3]
**.neighbourDiscovery.minIntervalBetweenRAs = 1 s
#MinRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.neighbourDiscovery.maxIntervalBetweenRAs = 3 s
#MaxRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.MN1.mobility.speed = 5 mps
```

```
**.MN1.udpApp[0].waitInterval = 0.133333 s
**.MN1.tcpApp[0].tSend = 40 s
**.MN1.tcpApp[0].sendBytes = 100000000 B

[Config load1_1mps_ra1_3]
**.neighbourDiscovery.minIntervalBetweenRAs = 1 s
#MinRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.neighbourDiscovery.maxIntervalBetweenRAs = 3 s
#MaxRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.MN1.mobility.speed = 1 mps
**.MN1.udpApp[0].waitInterval = 0.133333 s
**.MN1.tcpApp[0].tSend = 40 s
**.MN1.tcpApp[0].sendBytes = 100000000 B

[Config load1_2mps_ra1_3]
**.neighbourDiscovery.minIntervalBetweenRAs = 1 s
#MinRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.neighbourDiscovery.maxIntervalBetweenRAs = 3 s
#MaxRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.MN1.mobility.speed = 2 mps
**.MN1.udpApp[0].waitInterval = 0.133333 s
**.MN1.tcpApp[0].tSend = 40 s
**.MN1.tcpApp[0].sendBytes = 100000000 B

[Config load1_3mps_ra1_3]
**.neighbourDiscovery.minIntervalBetweenRAs = 1 s
#MinRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.neighbourDiscovery.maxIntervalBetweenRAs = 3 s
#MaxRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.MN1.mobility.speed = 3 mps
**.MN1.udpApp[0].waitInterval = 0.133333 s
**.MN1.tcpApp[0].tSend = 40 s
**.MN1.tcpApp[0].sendBytes = 100000000 B

[Config load1_4mps_ra1_3]
**.neighbourDiscovery.minIntervalBetweenRAs = 1 s
#MinRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.neighbourDiscovery.maxIntervalBetweenRAs = 3 s
#MaxRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.MN1.mobility.speed = 4 mps
**.MN1.udpApp[0].waitInterval = 0.133333 s
**.MN1.tcpApp[0].tSend = 40 s
**.MN1.tcpApp[0].sendBytes = 100000000 B

[Config load2_1mps_ra1_3]
**.neighbourDiscovery.minIntervalBetweenRAs = 1 s
#MinRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.neighbourDiscovery.maxIntervalBetweenRAs = 3 s
#MaxRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.MN1.mobility.speed = 1 mps
**.MN1.udpApp[0].waitInterval = 0.0666665 s
**.MN1.tcpApp[0].tSend = 40 s
**.MN1.tcpApp[0].sendBytes = 200000000 B

[Config load2_2mps_ra1_3]
**.neighbourDiscovery.minIntervalBetweenRAs = 1 s
#MinRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.neighbourDiscovery.maxIntervalBetweenRAs = 3 s
#MaxRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.MN1.mobility.speed = 2 mps
**.MN1.udpApp[0].waitInterval = 0.0666665 s
**.MN1.tcpApp[0].tSend = 40 s
**.MN1.tcpApp[0].sendBytes = 200000000 B

[Config load2_3mps_ra1_3]
**.neighbourDiscovery.minIntervalBetweenRAs = 1 s
#MinRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
```

```
**.neighbourDiscovery.maxIntervalBetweenRAs = 3 s
#MaxRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.MN1.mobility.speed = 3 mps
**.MN1.udpApp[0].waitInterval = 0.0666665 s
**.MN1.tcpApp[0].tSend = 40 s
**.MN1.tcpApp[0].sendBytes = 200000000 B

[Config load2_4mps_ra1_3]
**.neighbourDiscovery.minIntervalBetweenRAs = 1 s
#MinRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.neighbourDiscovery.maxIntervalBetweenRAs = 3 s
#MaxRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.MN1.mobility.speed = 4 mps
**.MN1.udpApp[0].waitInterval = 0.0666665 s
**.MN1.tcpApp[0].tSend = 40 s
**.MN1.tcpApp[0].sendBytes = 200000000 B


[Config load2_5mps_ra1_3]
**.neighbourDiscovery.minIntervalBetweenRAs = 1 s
#MinRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.neighbourDiscovery.maxIntervalBetweenRAs = 3 s
#MaxRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.MN1.mobility.speed = 5 mps
**.MN1.udpApp[0].waitInterval = 0.0666665 s
**.MN1.tcpApp[0].tSend = 40 s
**.MN1.tcpApp[0].sendBytes = 400000000 B

[Config load3_1mps_ra1_3]
**.neighbourDiscovery.minIntervalBetweenRAs = 1 s
#MinRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.neighbourDiscovery.maxIntervalBetweenRAs = 3 s
#MaxRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.MN1.mobility.speed = 1 mps
**.MN1.udpApp[0].waitInterval = 0.03333325 s
**.MN1.tcpApp[0].tSend = 40 s
**.MN1.tcpApp[0].sendBytes = 400000000 B

[Config load3_2mps_ra1_3]
**.neighbourDiscovery.minIntervalBetweenRAs = 1 s
#MinRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.neighbourDiscovery.maxIntervalBetweenRAs = 3 s
#MaxRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.MN1.mobility.speed = 2 mps
**.MN1.udpApp[0].waitInterval = 0.03333325 s
**.MN1.tcpApp[0].tSend = 40 s
**.MN1.tcpApp[0].sendBytes = 400000000 B

[Config load3_3mps_ra1_3]
**.neighbourDiscovery.minIntervalBetweenRAs = 1 s
#MinRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.neighbourDiscovery.maxIntervalBetweenRAs = 3 s
#MaxRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.MN1.mobility.speed = 3 mps
**.MN1.udpApp[0].waitInterval = 0.03333325 s
**.MN1.tcpApp[0].tSend = 40 s
**.MN1.tcpApp[0].sendBytes = 400000000 B

[Config load3_4mps_ra1_3]
**.neighbourDiscovery.minIntervalBetweenRAs = 1 s
#MinRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.neighbourDiscovery.maxIntervalBetweenRAs = 3 s
#MaxRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.MN1.mobility.speed = 4 mps
**.MN1.udpApp[0].waitInterval = 0.03333325 s
**.MN1.tcpApp[0].tSend = 40 s
**.MN1.tcpApp[0].sendBytes = 400000000 B
```

```
[Config load3_5mps_ra1_3]
**.neighbourDiscovery.minIntervalBetweenRAs = 1 s
#MinRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.neighbourDiscovery.maxIntervalBetweenRAs = 3 s
#MaxRtrAdvInterval (RFC 3775),applicable when MIPv6Support is true
**.MN1.mobility.speed = 5 mps
**.MN1.udpApp[0].waitInterval = 0.03333325 s
**.MN1.tcpApp[0].tSend = 40 s
**.MN1.tcpApp[0].sendBytes = 400000000 B
```

ÖZGEÇMİŞ

Gülsüm Zeynep GÜRKAŞ AYDIN 17 Temmuz 1981 tarihinde Kırklareli'nin Kaynarca beldesinde doğdu. 1999 yılında birincilikle bitirdiği lise öğreniminden sonra aynı yıl İstanbul Üniversitesi Bilgisayar Mühendisliği bölümüne girmeye hak kazandı ve 2003 yılında lisans eğitimini tamamladı. Ekim 2003 tarihinde İstanbul Üniversitesi Bilgisayar Mühendisliği Anabilim Dalında başladığı yüksek lisans eğitimini 2005 yılında tamamlamıştır. Aralık 2003 tarihinde aynı bölümde Araştırma Görevlisi olarak göreve başlamıştır ve halen bu göreve devam etmektedir.