EV AĞLARINDA OTOMATİK IPv6 YÖNLENDİRİCİ YAPILANDIRILMASI

Reha Oğuz ALTUĞ
Yüksek Lisans Tezi

Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı
Ağustos – 2006

JÜRİ VE ENSTİTÜ ONAYI

Reha Oğuz ALTUĞ'un "**Ev Ağlarında Otomatik IPv6 Yönlendirici Yapılandırılması"** başlıklı **Bilgisayar Mühendisliği** Anabilim Dalındaki, Yüksek Lisans tezi 04.08.2006 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

	Adı-Soyadı	**İmza**
Üye (Tez Danışmanı)	**: Yard.Doç. Dr. CÜNEYT AKINLAR**	**...……**
Üye	**: Yard.Doç. Dr. YUSUF OYSAL**	**...……**
Üye	**: Yard.Doç. Dr. EMİN GERMEN**	**...……**

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun …… tarih ve …… sayılı kararıyla onaylanmıştır.

Prof. Dr. Altuğ İFTAR

Enstitü Müdürü

ÖZET

Yüksek Lisans Tezi

EV AĞLARINDA IPv6 OTOMATİK YÖNLEDİRİCİ YAPILANDIRILMASI

Reha Oğuz ALTUĞ

Anadolu Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Yard. Doç. Dr. Cüneyt AKINLAR
2006, 52 sayfa

Bu çalışmada, IPv6 protokolünün eksik bir bölümü olan ev ve küçük işyeri ağlarında otomatik yönlendirici yapılandırılmasının nasıl gerçekleştirileceğinin belirlenmesi amaçlanmıştır. Ağ üzerinde öncelikle her bir bağ üzerine eşsiz bir alt ağ adresi atanması veya ağdaki tüm bağlar için tek bir alt ağ adresi atanması çözümleri üretilmiştir. Belirlenen çözümler öncelikle tek yönlendiricinin daha sonra birden fazla yönlendiricinin bulunduğu senaryolarda ayrıntılandırılmıştır. Tek yönlendiricinin bulunduğu ağlarda yapılandırma işlemi daha basit görünse bile, çok yönlendiricili ağlarda düğümlerin aranması ve alt ağ adresi çakışmaları gibi problemler ortaya çıkmaktadır. Bahsedilen sorunların aşılması için bu çalışmada bağ-durum algoritmalarından faydalanılmıştır.

IPv6 protokolünün istemci kısmında otomatik yapılandırılması olgunluğa erişmesine rağmen yönlendirici kısmında henüz olgunlaşmış bir yaklaşım bulunmadığı görülmüştür. Bu sebeple ağ üzerindeki tüm yönlendiricilerin el ile yapılandırılması gerekmektedir. Ev kullanıcılarının yeterli teknik bilgiye sahip olmadıkları ve küçük işletmelerin teknik eleman çalıştırmak için mali güçlerinin bulunmadığı göz önüne alındığında, otomatik yönlendirici yapılandırılması daha çok önem kazanmaktadır.

Yapılan çalışma, ev ağları veya küçük işletme ağları için çözüm getirmesinin yanında, daha karmaşık olan büyük ağlarla ilgili yapılandırma sorunlarına da uygulanabilir. Bu sayede IPv6 protokolünün gerçek tak-kullan mantığında çalışabilmesi sağlanabilir.

Anahtar Kelimeler: IPv6, Otomatik Yönlendirici Yapılandırılması, Çoklu-bağ Alt Ağı, Sıfır-Yapılandırmalı Ağlar, Ev Ağları

ABSTRACT

Master of Science Thesis

IPv6 ROUTER AUTO CONFIGURATION FOR HOME NETWORKS

Reha Oğuz ALTUĞ

Anadolu University
Graduate School of Sciences
Computer Engineering Program

Supervisor: Assist. Prof. Dr. Cüneyt AKINLAR

2006, 52 pages

The aim of this study is to enable IPv6 router auto-configuration in IPv6 networks. IPv6 host auto configuration is part of IPv6 specification from the start but IPv6 routers still needs manual configuration. This situation is not only unacceptable for home and small office networks; it also complicates the management of complex corporate networks. To enable easy and ubiquitous deployment of future IPv6 networks, there is a need for an IPv6 router auto configuration protocol to complement IPv6 host auto configuration protocol to make IPv6 networks truly plug-and-play.

In this study two approaches are proposed. First one is assigning unique subnets to all links in the network. Second approach is assigning a single subnet to all links. Both approaches are firstly applied to single-router networks. Then these solutions are extended to multi-router networks. Single-router auto configuration is straightforward but in multi-router configuration there are more challenging points like host awareness and subnet conflicts. These concerns are solved with the use of link-state routing algorithms as proposed in the study.

Keywords: IPv6, Router Auto Configuration, Multilink Subnet, ZeroConf Networks, SOHO Networks

TEŞEKKÜR

Bu çalışmayı yöneten, çalışma boyunca sürekli ilgi ve desteğini esirgemeyen, olumlu eleştiri ve önerileri ile çalışmama büyük katkıda bulunan danışman hocam Sn. Yrd. Doç. Dr. Cüneyt AKINLAR'a özveri ve nezaketi için, yine çalışma boyunca yardımlarını esirgemeyen Sn. Yard. Doç. Dr. Hakan ŞENEL'e ve bu çalışmanın meydana gelebilmesi için gösterdiği manevi destekten dolayı eşim Hicran ALTUĞ'a teşekkürü borç biliyorum.

Hazırlanan çalışmanın konu ile ilgilenenlere yararlı olmasını dileyerek saygı ve sevgilerimi sunuyorum.

Reha Oğuz ALTUĞ

Ağustos 2006

İÇİNDEKİLER

Sayfa

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

KISALTMALAR DİZİNİ

ARP	:	Address Resolution Protocol
CIDR	:	Classless Inter-Domain Routing
DHCPv6	:	Dynamic Host Configuration Protocol Version 6
DNS	:	Domain Name System
FDDI	:	Fiber-Distributed Data Interface
IANA	:	Internet Assigned Numbers Authority
ICMPv6	:	Internet Control Message Protocol version 6
IETF	:	Institute of Electrical and Electronics Engineers
IETF	:	Internet Engineering Task Force
IGMP	:	Internet Group Management Protocol
IHL	:	Internet Header Length
IPv4	:	Internet Protocol version 4
IPv6	:	Internet Protocol version 6
ISP	:	Internet Service Provider
ISDN	:	Integrated Services Digital Network
MAC	:	Media Access Control
MTU	:	Maximum Transfer Unit
NA	:	Neighbor Advertisement
NAT	:	Network Address Translation
NS	:	Neighbor Solicitation
OSI	:	Open Systems Interconnection
OSPF	:	Open Shortest Path First
OUI	:	Organizational Unit Identifier
PDU	:	Protocol Data Unit
RFC	:	Request For Comments
RIP	:	Routing Information Protocol
RS	:	Router Solicitation
RA	:	Router Advertisement
RP	:	Rendezvous Point
SOHO	:	Small Office Home Network
WiFi	:	Wireless Fidelity
xDSL	:	Digital Subscriber Line

1. GİRİŞ ve AMAÇ

Internet ağı 1990lı yılardan itibaren kuruluş amacını aşmış, ticari ve kişisel kullanım ağırlıklı bir yapıya geçmiştir. Internet'in yaygınlaşmasının sebepleri arasında "Word Wide Web" ve haberleşme servisleri gibi Internet ağının lokomotifleri olarak görülen yazılımsal etkenlerin yanında donanım teknolojisinin gelişmesi, haberleşme altyapısının yaygınlaşması ve yeni bağlantı teknolojilerinin uygulanmaya başlanması da önemli yer tutar. Şu an bir çok ev kullanıcısı kabul edilebilir ücretler karşılığında hızlı Internet bağlantısı hizmeti alabilmektedir.

Son yıllarda her ne kadar Kuzey Amerika ve Avrupa'daki Internet kullanıcılarının sayısındaki artış lineer bir görünüm kazandıysa da, gelişmekte olan bölgeler olarak adlandırabilecek Asya, Güney Amerika, Afrika ve Orta Doğu'da kullanıcıların sayısında geometrik bir artış görülmektedir. Kullanıcı sayısındaki artış karşısında mevcut Internet altyapısı yeterli olsa bile, Internet'in işlemesinde kullanılan Internet protokolü versiyon 4 (IPv4) ihtiyaca cevap verememektedir. Internet üzerinde her bir kullanıcının bir Internet numarası olması gerektiğinden kullanıcı sayısı arttıkça kullanılabilecek Internet numarası sıkıntısı yaşanmaktadır.

Internet kullanıcılarındaki artışın yanında kullanılan uygulamalar da çeşitlilik kazanmıştır (www, e-mail, görüntü ve ses transferi). Bu uygulamalar daha fazla veri transfer hızları, hizmet kalitesi (QoS : Quality of Service) gerektirmektedir. IPv4 protokolünün bu servisler düşünülerek tasarlanmamış olması nedeniyle, söz konusu uygulamalar için yetersiz kalmaktadır.

Internet'in ticari bir yapı kazanması ile birlikte IPv4 protokolünün eksikliklerinden birisinin de güvenlik konusunda olduğu görülmüştür. Mevcut protokolün üzerine eklentiler yapılarak oluşturulan güvenlik sistemleri hem protokolün işleyişini yavaşlatmakta hem de tüm istemcilerde bu eklentiler için desteğin bulunmasını gerektirmektedir.

Tüm bu sebepler Internet ağının bel kemiğini oluşturan yönlendiricilerin (routers) iş yüklerini arttırmaktadır. Bunun sonucunda ağlar arasındaki veri iletişiminde yavaşlama ve yönetimsel zorluklar ortaya çıkmaktadır.

Yukarıda anlatılan problemler göz önüne alınarak, IETF (Internet Engineering Task Force) tarafından IPv4 protokolünün yerine geçecek IPv6 protokolü tasarlanmıştır. IPv6 protokolünün çözdüğü sorunlar şunlardır:

- Internet kullanımının yaygınlaşması ile tükenen Internet adresleri,
- IPv4 protokolünün yönlendiriciler üzerine getirdiği yük,
- Daha basit yapılandırma ihtiyacı,
- Güvenlik,
- Hizmet Kalitesi desteği.

Günümüzde Internet ağının tümüne yakını 1981 yılında standartlaştırılan IPv4 protokolü tabanlıdır. IPv6 protokolünün yaygınlaşmasının önündeki en önemli etkenler mevcut yapının değiştirilmek istenmemesinden kaynaklanmaktadır. IPv4 tabanlı yapının değiştirilmesi ile ortaya çıkabilecek sorunlar şunlardır :

- Yapı içerisindeki istemci, sunucu, yönlendirici gibi aygıtlarda yazılımsal ve donanımsal güncellemelere gerek duyulması,
- Geçiş aşamasında IPv4 ve IPv6 protokollerinin her ikisinin de desteklenmesi gerekliliği,
- Yapının işleyişinden sorumlu teknik personelin eğitimi için kaynak ve zaman harcanması.

WWW, IPv4 için nasıl bir tetikleyici uygulama (killer application) olduysa IPv6 kullanımına geçiş için de bir tetikleyici uygulama beklenmektedir. Durum kontrolsüz otomatik yapılandırmanın (stateless autoconfiguration) IPv6 protokolü için bir tetikleyici uygulama olabileceği, fakat birden fazla uygulama alanına ihtiyaç duyduğu görülmüştür.

IPv6 ağları tek yönlendiricili ağlar ve çok yönlendiricili ağlar olmak üzere iki grupta sınıflandırılabilir. Tek yönlendiricili ağlar bir yönlendiricinin bulunduğu ve bu yönlendiricinin farklı bölütleri yıldız topolojisinde birleştirip Internet bağlantısı sağladığı ağlardır (örn. ev ve SOHO ağları). Çok yönlendiricili ağlar ise birden fazla yönlendiricinin bulunduğu ve farklı bölütlerin birbirlerine bağlandığı ağlardır.

Her ne kadar tek yönlendiricili ağların otomatik yapılandırılması basit bir durum olsa da çok karmaşık yönlendiricili ağların otomatik yapılandırılması güç

bir iştir. Bu tip ağların el ile yapılandırılması ve yönetilmesi IPv6 kullanımının yaygınlaşmasını engellemektedir. Bir çok farklı bağ katmanı teknolojilerinin kullanılması sebebiyle küçük ev ve SOHO ağlarında birden fazla yönlendirici bulunabilir ve bu yönlendiricilerin el ile müdahale edilmeden otomatik yapılandırılmasını sağlayacak bir protokole ihtiyaç duyulmaktadır.

IPv6 düğümleri için otomatik yapılandırma IPv6 protokolünün tasarımından itibaren düşünülüyor olmasına rağmen, IPv6 yönlendiricilerinin halen el ile yapılandırılması ve yönetilmesi gerekmektedir. Bir site veya organizasyon ISP'den (Internet Servis Sağlayıcısı) bağlantı hizmeti aldığı durumlarda her ne kadar bir evrensel önek (global prefix) alsa da organizasyon içi yapılandırma hala ağ yöneticilerine kalmaktadır [18-20]. Ağ yöneticisi ağdaki her bir bağ (link) veya bölüt (segment) için el ile yapılandırma yapacak ve ağ büyüdükçe yapılan iş karmaşıklaşacaktır. Daha kötü bir senaryo olarak ağ yapısı değiştiğinde tüm ağın tekrar gözden geçirilmesi ve tekrar yapılandırılması gerekebilir. Bu durum büyük ağların yönetimini daha karmaşık hale getirmekle birlikte ev ve küçük işyeri ağları için kabul edilebilir bir durum değildir. Teknik personel çalıştırma olanağı bulunmayan küçük ev ve SOHO ağlarında [14] gerçek anlamda IPv6 tak-çalıştır mimarisinin gerçekleştirilebilmesi için yönlendiricilerin otomatik yapılandırılması gerekmektedir. Standart hale getirilecek bir IPv6 otomatik yönlendirici yapılandırma protokolüyle, IPv6 ağları daha kolay kurulabilir ve yönetilebilir hale gelebilir. Bu standart protokol IPv6 düğümleri için otomatik yapılandırma mekanizmasını tamamlayacak ve IPv6 ağlarını gerçek tak-çalıştır kullanım kolaylığına kavuşturacaktır.

Bu çalışmada IPv6 yönlendiricileri için otomatik yapılandırma çözümleri geliştirilmiştir. Üretilen çözümler üçüncü ve dördüncü bölümlerde anlatılmıştır. Üçüncü bölümde açıklanan yöntemin ana fikri bir ağda bulunan her yönlendiricinin her bir ara yüzünün (interface) başlangıç durumunda eşsiz (unique) bir alt ağ adresi (subnetid) alması ve topolojik değişmelerde birbirleri ile işbirliği yaparak bu eşsizliği devam ettirmesidir. Bu bölümde hem tek ve hem de çok yönlendiricili IPv6 ağlarının eşsiz alt ağ adresleri kullanılarak otomatik yapılandırılmasının nasıl sağlanacağı ve yapılacak iş için gerekli algoritma açıklanmış, bahsedilen fikrin temel iç-alan topolojisi tüme gönderim (intra-

domain topology broadcast) algoritmaları üzerinde yapılacak basit değişikliklerle nasıl gerçekleştirilebileceği gösterilmiştir.

Dördüncü bölümde ise, ağdaki tüm bağların aynı alt ağ ön ekini kullandığı bir yapı düşünülmüştür [27]. Bu amaçla düşünülen yöntemde, yönlendiricilerin tüm bağlarda tek alt ağ ön eki yayımlayarak IPv6 katmanında bir çoklu-bağ alt ağı (muti-link subnet) oluşturmaları ve IPv6 ağlarında bahsedilen çoklu-bağ desteğinin nasıl gerçekleştirilebileceği açıklanmıştır. Bu amaçla öncelikle ev ağları gibi tek-yönlendiricili (single-router) basit ağlar göz önüne alınarak çoklu-bağ alt ağı desteğinin bu ağda nasıl destekleneceğini gösteren iki ayrı yöntem sunulmaktadır. Anlatılan yöntemlerin daha karmaşık olan çok yönlendiricili (multi-router) ağlara nasıl genişletileceği üzerinde de durulmaktadır. Bu yapılandırma yöntemi ile tüm ağ için tek bir alt ağ ön eki yeterli olmasının yanında el ile yönetim sorununu gidererek, otomatik adres yapılandırmasını kolaylaştırmaktadır. Önerilen yöntemlerin yeni olduğu ve standart otomatik düğüm yapılandırma protokolünü tamamlayarak gerçek tak-çalıştır IPv6 mimarisini oluşturacağı düşünülmektedir.

Beşinci bölümde eşsiz alt ağ adresleri ile yapılandırma metodunun gerçeklemesi anlatılmıştır. Gerçekleme için bir program yazılmış ve yapılandırma yöntemindeki her bir basamak modellenmiştir.

Bu çalışma içerisinde küçük ev ve işyeri ağlarına SOHO ağı ismi verilmiştir. Ayrıca IP adresi tanımı aksi belirtilmedikçe IPv6 protokolü üzerindeki Internet adreslerini ifade etmektedir.

2. IPv6 PROTOKOLÜ

Kullanıcı sayısındaki artış ve hali hazırda kullanılmakta olan IPv4 protokolünün tasarımındaki sınırlamalar nedeni ile Internet Ağı ile ilgili çok sayıda sorun yaşanmaktadır. Bu sorunlardan en büyüğü giderek azalan IP numaralarıdır. IP Numaralarının tükenmesi ile birlikte daha fazla kullanıcının veya sunucunun Internet'e bağlanması olanaksız hale gelmektedir. Bu sorunun aşılması için düşünülen Ağ Adres Dönüşümü (Network Address Translation - NAT) adı verilen kullanıcılara evrensel (public) adres yerine özel (private) adres verilmesi ve bu kullanıcıların evrensel adreslere sahip bilgisayarların arkasına saklanması yöntemi yetersiz kalmaktadır. Ağ Adres Dönüşümü metodunun kullanıcıları dışarıdan ulaşılamaz hale getirmesi ve bir çok protokolü desteklememesi gibi eksileri bulunmaktadır [1].

IPv6'nın getirdiği çok daha büyük adres alanının (address space) bu sorunu ortadan kaldıracağı düşünülmektedir. IPv6 ile her Internet kullanıcısının evrensel bir adresi olacak böylece Ağ Adres Dönüşümü gibi yöntemlere gerek duyulmayacaktır.

IPv4 protokolünün sorunlarından bir diğeri Internet omurgasını oluşturan yönlendiricilerin yönlendirme tablolarının alarm verecek boyutlara ulaşmasıdır. Bu sorun hem yönlendirme işlemini yavaşlatmakta hem de yönlendiriciler arasında yönlendirme bilgilerinin alışverişini sağlayan yönlendirme protokollerini yetersiz kılmaktadır [3]. IPv6 getirdiği daha efektif, hiyerarşik ve yalın yönlendirme yapısı sayesinde yönlendirme problemini ortadan kaldıracaktır.

IPv4'ün diğer bir sorunu da otomatik yapılandırma olarak görülmektedir. Deneyimsiz bir Internet kullanıcısı DHCP (Dinamik İstemci Yapılandırma Protokolü) sunucusunun bulunmadığı bir ağa giriş yaptığı zaman adres, ağ geçidi ve DNS (Internet Ad Sistemi) yapılandırması gibi bir çok kavramı bildiği varsayımıyla karşılaşabilmektedir. IPv6, getirdiği otomatik yapılandırma yöntemleri sayesinde bu sorunları çözmekte ve teknik bilgi sahibi olmayan kullacıların dahi kolayca Internet ağına bağlanmasına imkan vermektedir.

IPv4'te güvenlik problemi IPSec protokolünün eklenmesi ile çözülmüş gibi görünse de, iki istemci haberleşirken güvenliğin sağlanabilmesi için IPSec

protokolü her iki istemcide de kurulu olmalıdır. IPv6, doğrudan IPSec desteği vererek uyum sorunlarını ortadan kaldırmıştır.

Internet'in yaygın olarak kullanıldığı alanlardan biri, görüntü ve ses iletimidir. Diğer haberleşme türlerinden farklı olarak, görüntü ve ses iletimi zaman-hassas (time-critical) uygulamalardır. IPv4 protokolü bu tür uygulamalar için sınırlı destek vermektedir. IPv6 protokolünün tasarlanması sürecinde hizmet kalitesi önemli bir yer tutmakta ve üst katmanda kullanılan uygulamaların ihtiyaç duyduğu öncelikler karşılanmaktadır.

IPv6'ya geçiş yapmak için tetikleyici uygulamalardan biri olarak görülen mobil cihazlara verilen destek otomatik konfigürasyon yetenekleri ile arttırılmıştır.

IPv6'ya geçiş için sayılan engellerden biri teknolojik alt yapının hazır olup olmadığıdır. Altyapının en önemli ayağı sunucuların ve son kullanıcıların ihtiyaç duydukları işletim sistemi ve yazılımlardır.

IPv6 destekleyen işletim sistemleri

- Macintosh,
- Unix / Linux,
- Windows,
- OS/X

olarak sıralanabilir.

Çok sayıda yazılım gerek yeni versiyonlarında gerekse güncellemelerle IPv6 desteğini sağlamaktadır. Internet omurgasını oluşturan yönlendiriciler için de IPv6 desteği tamamlanmış bulunmaktadır. Cisco, Hitachi, Nortel Networks gibi sektörün önde gelen üreticileri yeni ürünlerinde ve yazılım güncellemesi ile eski ürünlerinde bu desteği sağlamışlardır [2].

Sonuç olarak IPv6'ya geçiş için teknolojik altyapıyı oluşturan öğeler olan işletim sistemi, yazılım ve yönlendiricilerin uyumu sorununun ortadan kalktığı görülmektedir.

2.1 IPv6 Genel Özellikleri

IPv6 protokolü, IPv4 protokolünün geçirdiği evrimsel süreç ve eksikleri göz önüne alınarak tasarlanmıştır. Protokolün tasarlanmasındaki göz önüne alınan

önemli parametreler esneklik, kolay kullanım ve IP katmanın üst ve alt katmanlarında minimum değişiklik yapılması gereği olarak sıralanabilir. IPv6 protokolü bu özelliklerin desteklenebilmesi için getirdiği yenilikler ve geliştirmeler aşağıda açıklanmaktadır.

2.1.1. Yeni Başlık Formatı

IPv6'da başlık formatı başlık (header) işleme yükünü (overhead) azaltmak amacı ile tamamen değiştirilmiştir. Başlık içerisinde temel bazı temel alanlar belirlenmiş, daha önemsiz ve seçimli (optional) alanlar başlığın arkasında bulunacak şekilde tasarlanmıştır. IPv4 ile IPv6 paketleri birbiri ile uyumsuzdur, diğer bir değişle IPv6 geriye yönelik destek vermez. Eğer bir sistem iki başlık formatını da kullanmak istiyorsa hem IPv4 hem de IPv6 protokolünü kullanmalıdır. IPv6 adresi IPv4 adresinden dört kat daha büyük olmasına rağmen IPv6 başlığı IPv4 başlığından sadece iki kat daha büyüktür [4].

2.1.2. Geniş Adres Alanı

IPv4'te kullanılan 32 bitlik adresler yerine IPv6 adresleri 128 bitten (16 bayt) oluşur. 128 bitlik adres kullanımı $3{,}40 \times 10^{38}$ farklı adres türetilmesine olanak verir. 128 bitin tamamı adresleme için kullanmak yerine bir kısmı hiyerarşik bir yapı oluşturmak için farklı seviyelerde alt ağların oluşturulmasında kullanılacak, bu sayede büyük yönlendirme tablolarına ve Ağ Adres Dönüşümü gibi metotlara gerek kalmayacaktır [4].

2.1.3. Durum Kontrollü ve Durum Kontrolsüz Adres Yapılandırması

IPv6 protokolü, durum kontrollü (stateful) ve durum kontrolsüz (stateless) olmak üzere iki tip otomatik adreslendirme yöntemi kullanmaktadır. Durum kontrollü yapılandırmada, istemciler yönlendirici mesajlarına bakarak IP adresi alabilecekleri DHCPv6 ana makinelerini öğrenirler. Daha sonra DHCPv6 ana makineleri ile iletişime geçerek gerekli adres ve konfigürasyon bilgilerini alırlar.

Durum kontrolsüz yapılandırmada ise istemciler bulundukları bağ üzerinde bulunan yönlendiricilerden aldıkları önekleri (prefix) kullanarak kendileri için IP adresleri oluştururlar. İstemciler, bir yönlendiricinin bulunmadığı durumlarda bile kendilerini bağ-içi (link-local) adresler ile otomatik yapılandırarak haberleşebilirler [4]. IPv4 ile IPv6 arasındaki temel farklılıklar Çizelge 2.1'de sunulmaktadır.

2.1.4. Tümleşik Güvenlik Yapısı

IPv6'da güvenlik bir opsiyon değil gereklilik olarak görülmüştür ve IPSec protokolü IPv6'da tümleşik olarak gelmektedir. Bu sayede tüm IPv6 istemcilerinde güvenlik protokolü bulunmakta ve uyumsuzluklar ortadan kaldırılmaktadır.

2.1.5. Hizmet Kalitesi

IPv6 paket başlığı içerisinde bulunan akış etiketi (flow label) alanını kullanılarak ile IPv6 paketleri çeşitli gruplara ayrılabilmektedir. Yönlendiriciler tarafından bu gruplar üzerinde uygulanacak politikalar (policy) sayesinde, IPv6 ile hizmet kalitesi uygulamaları kolaylıkla gerçekleştirilebilir. Ayrıca IPv6 trafik akışını başlık bilgisinden elde ettiği için şifrelenmiş veri taşıyan paketlerle ilgili sorunu da çözmektedir.

2.1.6. Mobil Cihaz Desteği

IPv6'da mobil düğümler (node) herhangi bir ek protokol veya yönlendirici desteği olmadan bir ağdan diğerine geçiş yapabilirler. Bu geçiş sırasında herhangi bir adres değişikliğine ihtiyaç duyulmamaktadır. Bu özellikleri sayesinde IPv6 mobil uygulamalar için de uygun bir ortam oluşturmaktadır [5].

Çizelge 2.1 IPv4 ile IPv6 arasındaki temel farklar [4]

IPv4	**IPv6**
Hedef ve Kaynak Adresleri 32 bittir.	Hedef ve Kaynak Adresleri 128 bittir.
IPSec desteği isteğe bağlıdır.	IPSec desteği bir gerekliliktir.
Yönlendiriciler tarafından hizmet niteliği için kullanılacak paket akışı ayracı yoktur.	IPv6 başlığında bulunan ve yönlendiriciler tarafından hizmet niteliği için kullanılacak paket akış etiketi vardır.
Parçalanma (fragmentation) hem yönlendiricilerde hem de istemcilerde yapılır.	Parçalanma sadece istemcilerde yapılır.
Sağlama toplamı (checksum) başlık içerisindedir.	Başlık sağlama toplamını içermez.
Başlık opsiyonları içerir.	Bütün opsiyonel veriler uzantı başlıklarının içerisinde yer alır.
Veri Bağı Katmanı (Data Link Layer) adresi ile IP adresi eşleştirmesi için Adres Çözme Protokolü (Address Resolution Protocol - ARP) kullanılır.	Adres Çözme Protokolü yerine Komşu Sorgulama mesajları (Neighbor Solicitation Message - NS) getirilmiştir.
Alt ağ grup üyeliğini yönetmek için Internet Grup Yönetim Protokolü (Internet Group Management Protocol - IGMP) kullanılmaktadır.	IGMP yerine Çoğa Gönderim Dinleyici Keşif mesajları (Multicast Listener Discovery messages) getirilmiştir.
En iyi varsayılan ağ geçidini bulmak için opsiyonel olan ICMP Yönlendirici Keşif (ICMP Router Discovery) protokolü kullanılmaktadır.	ICMP Yönlendirici Keşif protokolü yerine ICMPv6 Yönlendirici Sorgulama (Router Solicitation - RS) ve Yönlendirici Cevap (Router Adversisement - RA) mesajları kullanılmaktadır.
Bir mesajı alt ağda bütün düğümlere göndermek için tüme gönderim (broadcast) adresi kullanılmaktadır.	IPv6'da tüme gönderim adresi bulunmamaktadır. Bunun yerine bağ-içi alanı tüm-düğümler çoğa gönderim (link-local scope all-nodes multicast) adresleri kullanılmaktadır.
El ile veya DHCP aracılığı ile yapılandırma yapılabilir.	El ile veya DHCP aracılığı ile yapılandırmaya ihtiyaç yoktur.
Parçalanma ihtimali olan 576 baytlık paket büyüklüğünü destekler.	Parçalama olmadan 1280 baytlık paket büyüklüğünü destekler.

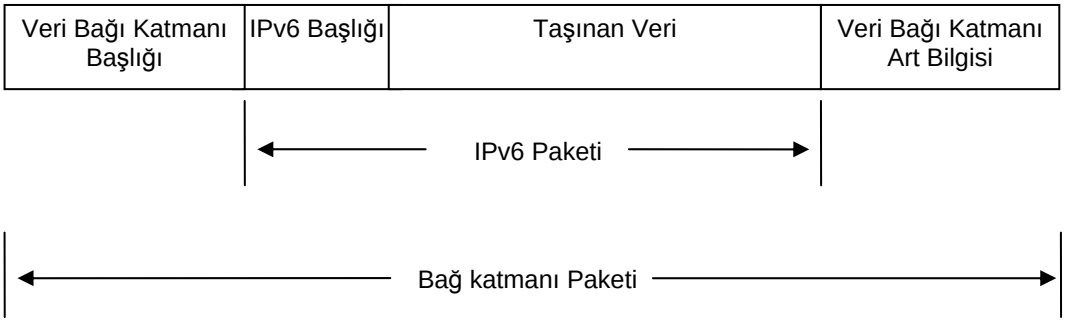
2.1.7. Genişletilebilirlik

IPv4 paketleri sadece 40 bayt büyüklüğünde uzantılara destek vermektedir. IPv6 paketi ise, uzantı başlıklarının (extension headers) IPv6 paket başlığının arkasından gelmesi sebebi ile, boyutu IPv6 paket boyutuna kadar olan büyüklükteki uzantılara destek vermektedir [4].

2.2. Ağ İletim Ortamlarında IPv6 Paketi

OSI (Open Systems Interconnection) modeline göre, IPv6 üçüncü katman olan Ağ Katmanında (Network Layer) bulunmaktadır. IPv6 paketleri oluşturulduktan sonra bir alttaki katman olan Veri Bağı Katmanına gönderilmektedir. Veri Bağı Katmanında IPv6 paketleri ikinci katman paketleri içine yerleştirilmektedir. Bu işleme sarmallama (encapsulation) denilmekte ve IPv6 paketinin başına Veri Bağı Katmanı Başlığı, sonuna ise Veri Bağı Katmanı Art Bilgisi eklenmektedir.

Veri Bağı Katmanı'nda IPv6 paket yapısı Şekil 2.1'de gösterilmektedir.

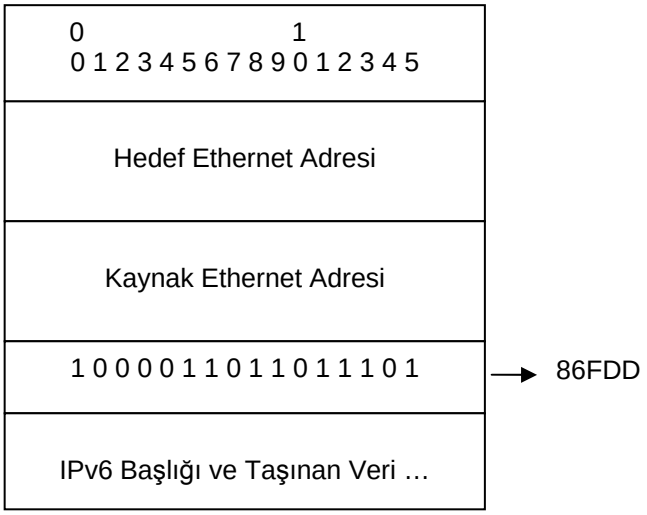


Şekil 2.1. Veri bağı Katmanı'nda IPv6 paket yapısı [4]

Internet'e bağlı olan istemcilerin ve sunucuların tamamına yakını IP protokolünü kullansalar da Veri Bağı Katmanı'nda farklı teknolojiler kullanılabilirler. Bunun sonucu olarak, genelde Şekil 2.1'e uyan fakat içerik olarak farklı paket yapıları bulunabilmektedir.

2.2.1. Ethernet Sarmalaması

IPv6 paketleri standart Ethernet Çerçevesi (Ethernet Frame) kullanmaktadır. Ethernet başlığında onaltılık 86DD değerini içermesi gereken Ethernet kodu, hedef ve kaynak Ethernet adresleri bulunur. IPv6 başlığını içeren veri alanından hemen sonra taşınan veri gelir. Ethernet bağının en az çerçeve boyutundan daha küçük bir paket varsa dolgulama (padding) baytlarıyla boyut artırılır. Ethernet çerçeve yapısı Şekil 2.2'de gösterilmiştir [6].



Şekil 2.2. Ethernet çevre yapısı [6]

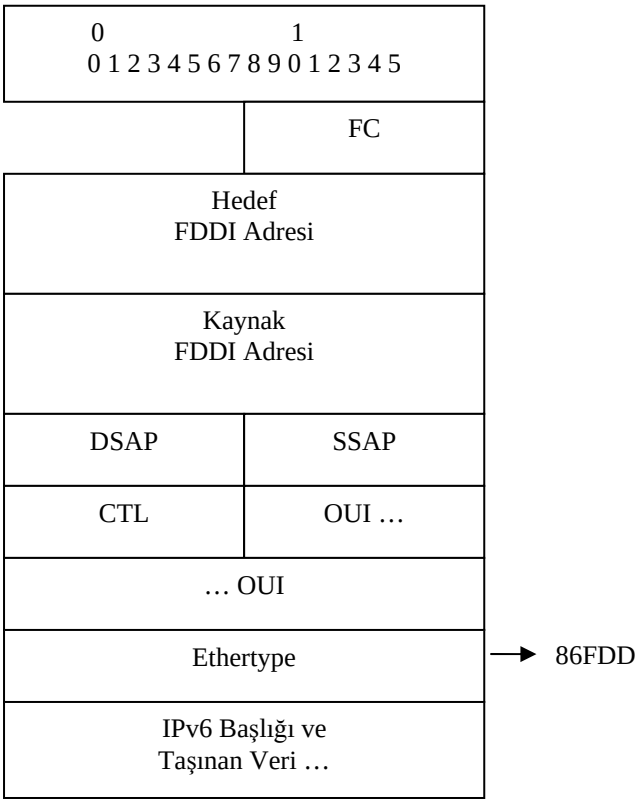
2.2.2. FDDI Sarmalaması

FDDI en fazla 4500 baytlık çerçeve büyüklüğüne izin verir. Veri bağı sarmallaması (22 bayt), LLC/SNAP başlığı(8 bayt) çıkarıldığında teorik olarak IPv6 paketi için 4470 baytlık bir alan kalır. Fakat genişletilebilirlik kaygılarıyla bu değer 4352 bayta indirilmiş ve varsayılan MTU (Maximum Transfer Unit) boyutu olarak atanmıştır.

IPv6 paketleri, FDDI ağı üzerinde uzun-biçimli (long-format) 48 bitlik adresler kullanılarak LLC/SNAP çerçevelerinde iletilir. Veri alanı, IPv6 başlığını ve taşınan veriyi içerir. Veri alanını FDDI çerçeve denetim dizisi (frame check sequence), bitiş ayracı (ending delimiter) ve çerçeve durum sembolleri (frame status symbols) izler. FDDI çerçeve yapısı Şekil 2.3'te gösterilmektedir.

FDDI çerçevesinde bulunan alanlar şunlardır :

- FC : Çerçeve Kodu. 50 ile 57 arasında değer almalıdır.
- DSAP/SSAP : SNAP sarmalamasını ifade eden onaltılık AA değerlerini taşımalıdır.
- CTL : Kontrol Alanı. Numaralandırılmamış bilgiyi ifade eden onaltılık 03 değerini taşımalıdır.
- OUI : Onaltılık 000000 değerini taşımalıdır.
- Ethertpye : Ethernet protokol tipi onaltılık 86DD değerini almalıdır [7].



Şekil 2.3. FDDI çevre yapısı [7]

IPv6 paketlerinin Andaçlı Halka (Token Ring) ağlarında iletimi için RFC 2470, PPP ağlarında iletimi için RFC 2472 ve Frame Relay ağlarında iletimi için RFC 2590 dokümanları çıkarılmıştır.[1]

[1] RFC (Request For Comments.) : IETF tarafından çeşitli konularda oluşturulan Internet standartları ve bu standartların dokümanları.

2.3 IPv6 Başlık Yapısı

IP protokollerinde başlık kısmı yönlendirme bilgisi içerme ve paketin geri kalanı hakkında bilgi verme gibi görevleri yerine getirir. Bu bilgileri saklamak için tanımlanmış belirli alanlar ve bu alanlar için belirlenmiş özel değerler bulunmaktadır. Bir IP düğümü, bu alanlara bakarak paketi işlemektedir.

IPv6 başlığı, IPv4 başlığında az kullanılan veya kullanılmayan alanların çıkarılması ile sadeleştirilmiştir. IPv4 başlığında bulunan isteğe bağlı bu alanlar, uzantı başlıkları olarak IPv6 başlığının arkasında yer almaktadır. Bu sayede IPv6 başlığı 40 bayt ile sabitlenmiştir.

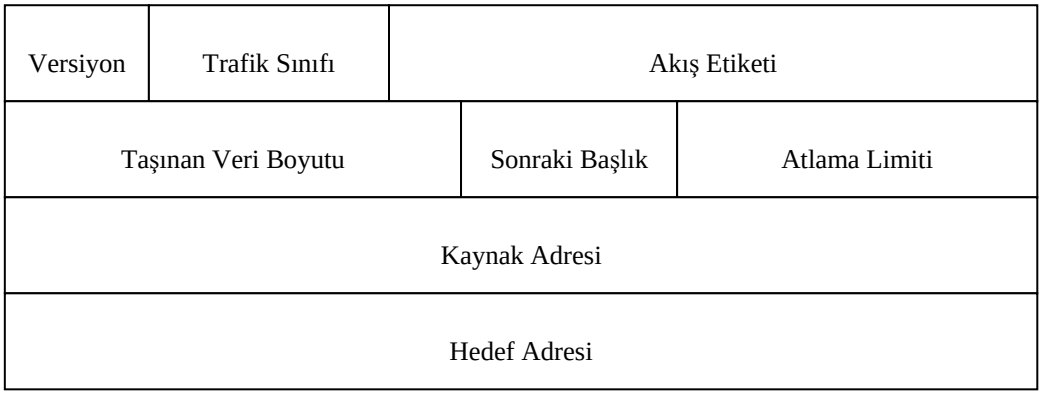
Basit bir IPv4 paketinin 20 bayt olduğu düşünüldüğünde 40 baytlık bir IPv6 paketinin işlenmesi daha zor gibi görünse de, IPv4 paketinin daha çok alana ve değişken boyuta sahip olması IPv6 paketine göre işlenmesini daha zor bir hale getirmektedir [4].

IPv6 ve IPv4 paketlerinin yapıları Şekil 2.4 ile Şekil2.5'te görülmektedir.

<table>
<tr><td>Versiyon</td><td>IHL</td><td>Servis Tipi</td><td colspan="2">Toplam Uzunluk</td></tr>
<tr><td colspan="3">Tanımlama</td><td>İmler</td><td>Parçalanma Konumu</td></tr>
<tr><td colspan="2">Yaşama Süresi</td><td>Protokol</td><td colspan="2">Başlık Sağlaması</td></tr>
<tr><td colspan="5">Kaynak Adresi</td></tr>
<tr><td colspan="5">Hedef Adresi</td></tr>
<tr><td colspan="4">Opsiyonlar</td><td>Dolgu</td></tr>
</table>

Şekil 2.4. IPv4 paketi başlık yapısı [9]

Bir IPv4 paketi, opsiyonları ve dolgulama bitleri kullanılmadığı taktirde en az 12 farklı alandan oluşmaktadır. Opsiyon alanı genişletilebilir olduğu için bir IPv4 paketi çok daha karmaşık olabilir. IPv4 paketi yapısı ile ilgili daha fazla bilgi için RFC 971'e bakılabilir.



Şekil 2.5. IPv6 paketi başlık yapısı [8]

Şekil 2.5'te görüldüğü gibi IPv6 paketi çok sade bir yapıda tasarlanmıştır. IPv6 paket yapısında bulunan alanlar şunlardır :

- Versiyon : 4 bittir. IPv6 için 6 değerini alır.
- Trafik Sınıfı : 8 bittir. IPv6 paketini işlenmesinde dikkat edilecek sınıf veya önem sırasını içerir.
- Akış etiketi : 20 bittir. IPv6 paketi eğer zaman-kritik bir iletişimde kullanılıyorsa bu değer yönlendiricilerde paketin akış içerisinde ki sırasını saptamaya yarar. Eğer istemciler arasında servis kalitesi gerektirmeyen bir iletişim varsa bu alanın değeri sıfırdır.
- Taşınan Veri Boyutu : 16 bittir. Üst katman PDU'su (Protocol Data Unit) ve tüm uzantı başlıkları dahil olmak üzere taşınan verini boyutunu içerir. Eğer taşınan veri boyutu 16 bitle ifade edilebilecek 65.535 bayttan daha büyükse bu alan sıfır değerini alır ve Jumbo Taşınan Veri Opsiyonu (Jumbo Payload Option) kullanılır.
- Sonraki Başlık : 8 bittir. Başlıktan sonra gelen alanın uzantı başlığı tipi veya üst katman protokolü tipi (TCP, UDP vb.) ile ilgili bilgi içerir.
- Dolaşma Limiti : 8 bittir. IPv6 paketinin dolaşabileceği en fazla düğüm sayısını (hop limit) verir. IP paketinin geçtiği her bir düğümde değer bir azaltılır ve sıfıra eşit olduğunda paket atılır.

- Kaynak Adresi : 128 bittir. IP paketini oluşturan düğümün adresidir.
- Hedef Adresi : 128 bittir. IP paketinin gönderildiği adrestir. Bu alan tek bir düğüm adresi olabileceği gibi çoğa-gönderim gibi metotlarda farklı değerler alabilir[8].

2.4. IPv6 Adresleri

IPv6 adresleri, 32 bitlik adres kullanan IPv4'te artan adres sıkıntısını gidermek amacıyla, 128 bitten oluşmuştur. 128 bitlik adres alanı 2^{128} diğer bir değişle $3{,}4x10^{38}$ farklı adresin kullanılmasına izin verir. Bu sayı dünyadaki her bir metrekareye $6{,}5x10^{23}$ tane adres düşmesine olanak verir. Ancak tüm adres alanının adreslendirme için kullanılması yerine hiyerarşik yönlendirme alanlarına (domain) bölünmesi ile daha efektif bir yönlendirme yapısı oluşturulacaktır [4].

2.4.1. IPv6 Adres Gösterimi

IPv6 adres gösterimi için üç farklı yöntem bulunmaktadır :

- Tercih edilen format x:x:x:x:x:x:x:x şeklindedir. Bu gösterimde *x* değeri dört adet onaltılık sayıdan oluşmaktadır.
  Örn.
  ABCD:EF01:2345:6789:ABCD:EF01:2345:6789
  2001:DB8:0:0:8:800:200C:417A
  Bu formatta bir bölümün içerisinde birbirini takip eden sıfırlar yazılmak zorunda değildir.
- Belirli tiplerdeki IPv6 adreslerinin ayrılma yöntemlerinden dolayı adresler arka arkaya gelen uzun sıfır dizilerinden oluşmaktadır. Bu tip adreslerde yazım kolaylığı sağlamak amacı ile sıfırları sıkıştırarak elde edilen bir gösterim düşünülmüştür. Bir yada birden fazla onaltılık alanın sıfır değeri taşıdığını göstermek amacıyla "::" işareti kullanılır. Bir adreste "::" işareti yalnız bir kere kullanılabilir.

Örn.

FF01:0:0:0:0:0:0:101 yerine sıkıştırılmış hali

FF01::101 gösterimi kullanılabilir.

- IPv6 ve IPv4 ağlarının iç içe bulunduğu ortamlarda adresin ilk 6 parçasının onaltılık *x* değerleri tarafından, son 4 parçasının ise onluk *d* değerleri tarafından oluşturulduğu x:x:x:x:x:x:d.d.d.d gösterimi kullanılabilir. Bu gösterimde *d* ile ifade edilen kısım standart IPv4 gösterimidir.

  Örn.

  0:0:0:0:0:0:13.1.68.3 veya sıkıştırılmış hali

  ::13.1.68.3 gösterimi kullanılabilir [15].

2.4.2. IPv6 Adres Ön Eki Gösterimi

Ön ekler adreslerin değişmeyen veya ağ ayracına ait olan bitlerinden oluşurlar. Ayraçlar, yollar ve adres alanları için kullanılan IPv6 öneklerinin gösteriminde IPv4 Sınıfsız Alan-İçi Yönlendirme gösterimi (Classless Inter-Domain Routing notiation - CIDR) kullanılır. IPv6 öneki *adres/önek-uzunluğu* formatında gösterilir.

Örn.

21DA:D3::/48 bir yol ön ekini

21DA:D3:0:2F3B ise bir alt ağ ön ekini gösterir [4].

IPv4'te kullanılan alt ağ adresi maskesi (subnet mask) IPv6'da kullanılmamaktadır. Alt ağ adresi maskesi yerine adres ön eki kullanılmaktadır.

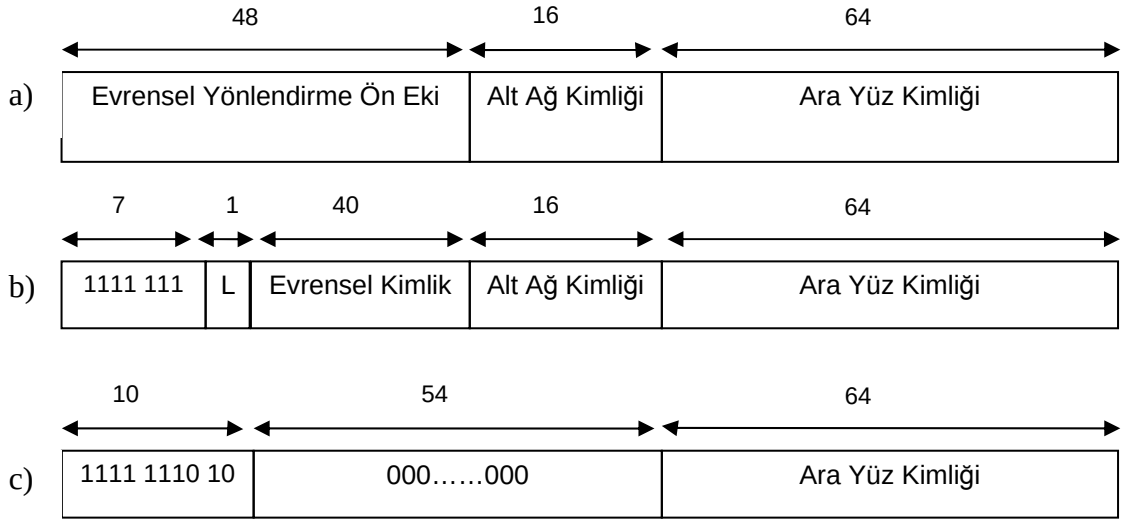
2.4.3. Teke-Gönderim IPv6 Adresleri

Teke-gönderim adresleri sadece bir ara yüzü ayırt emek için kullanılan adreslerdir. Teke-gönderim adresine gönderilmiş paketler sadece o adresi taşıyan ara yüzlere iletilir [15-17]. Teke-gönderim adresleri üç grupta sıralanabilir :

- *Evrensel teke-gönderim adresleri* : IPv4 genel (public) adresleri ile aynı işlevdedir. Bu adresler Internet üzerinden doğrudan erişilebilir ve

yönlendirilebilirler. Bu sebeple Internet üzerinde eşsiz olmaları gerekir. Evrensel teke-gönderim adresi 48 bitlik evrensel yönlendirme ön eki (global routing prefix), 16 bitlik alt ağ kimliği ve 64 bitlik ara yüz kimliğinden oluşur. Bir organizasyon evrensel yönlendirme ön ekini başlangıç durumunda bir servis sağlayıcıdan DHCPv6 [18,19], ICMPv6 [20] protokollerini veya başka bir protokol kullanarak alır. 16 bitlik alt ağ kimliği ağ içerisine IPv6 alt ağlarını ayırt etmek için kullanılır ve ağ yöneticisi tarafından atanır. Ara yüz kimliği bir alt ağdaki ara yüzü ifade eder [15]. Evrensel teke-gönderim adresinin yapısı Şekil 2.6(a)'da gösterilmektedir.

- *Yerel teke-gönderim adresleri* : Site-içi (site-local) adreslerin [15] yerini alması için tasarlanan yerel teke-gönderim adresleri standart hale gelmiştir [17]. Bu adresler Internet bağlantısı kesilmesi durumunda (diğer bir değişle servis sağlayıcı tarafından verilen evrensel yönlendirme ön eki yoksa) veya site içi trafiğin site içerisine mahsus olmasını sağlamak için bir ön ek olarak kullanılırlar [17]. Yerel teke-gönderim adresleri evrensel yönlendirilebilir adreslere benzer yapıdadır: İlk 7 bit sabit FC00::/7 değerini alır, arkasından L adı verilen ve 1 değerini taşıyan bir bit gelir ve daha sonra 40 bitlik bir evrensel kimlik bulunur. 40 bitlik bir evrensel kimliği 16 bitlik alt ağ kimliği izler. Son 64 bit ise ara yüz kimliğidir. Yerel teke-gönderim adreslerinin evrensel olarak eşsiz olması beklenir ancak yalnızca site içi iletişimde kullanılırlar. Yerel teke-gönderim adresinin yapısı Şekil 2.6(b)'de verilmektedir.
- *Bağ-içi teke-gönderim adresleri* : Bağ-içi teke gönderim adresleri sadece tek bir bağ içinde kullanılabilir. Bu adresler komşu sorgulama, otomatik adres yapılandırma veya bağın bağlı olduğu herhangi bir yönlendirici bulunmadığı durumlarda düğümler arasında iletişimin sağlanması gibi amaçlarla kullanılırlar. Yönlendiriciler bu tip adresleri bağ içerisinden dışarıya yönlendirmemelidirler [4]. Bağ-içi teke-gönderim adresinin yapısı Şekil 2.6(c)'de gösterilmektedir.

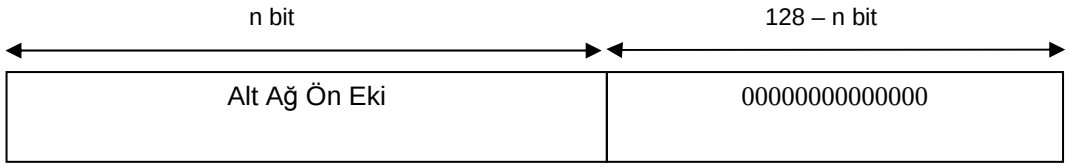


Şekil 2.6. Teke gönderim IPv6 adres yapıları. (a) Servis sağlayıcı bazlı evrensel teke-gönderim adresi, (b) yerel teke gönderim adresi, (c) yerel-bağ (link-local) adresi

Bu çalışmada yerel veya evrensel teke-gönderim adresleri "p.s.i" şeklinde ifade edilmiştir. Bu tanımda *p* 48 bitlik ön eki, *s* 16 bitlik al ağ kimliğini, *i* ise 64 bitlik ara yüz kimliğini ifade etmektedir.

2.4.4. Herhangi-Bire-Gönderim IPv6 Adresleri

Herhangi-bire-gönderim (anycast) adresleri genellikle farklı düğümlere ait bir grup ara yüzü ifade etmek için kullanırlar. Herhangi-bire-gönderim adresine gönderilmiş olan bir paket bu adrese sahip (genellikle yönlendirme algoritmalarına göre en yakındakine) tek bir ara yüze iletilir. Herhangi-bire-gönderim adresleri günümüzde yalnızca hedef adresi olarak kullanılmakta ve sadece yönlendiricilere atanmaktadır. Herhangi-bire-gönderim adresinin ilk n biti ara yüzün ait olduğu alt ağ adresini ifade ederken geri kalan 128-n bitine sıfır değeri atanır. Herhangi-bire-gönderim adresinin yapısı Şekil 2.7'de gösterilmiştir [15].

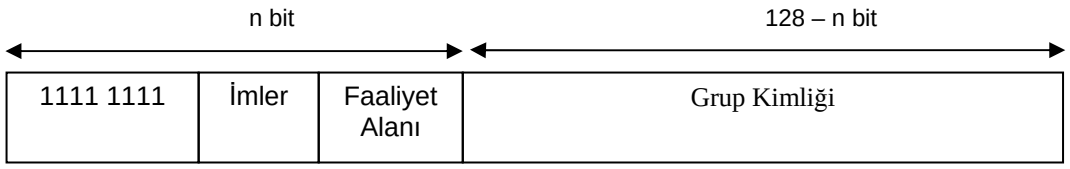


Şekil 2.7. Herhangi-bire-gönderim IPv6 adres yapısı

2.4.5. Tüme-Gönderim IPv6 Adresleri

Tüme-gönderim adresleri, herhangi-bire-gönderim adresleri gibi genellikle farklı düğümlere ait bir grup ara yüzü ifade etmek için kullanır. Ancak herhangi-bire-gönderim adreslerinden farklı olarak bir tüme-gönderim adresine gönderilen paket bu adrese sahip tüm ara yüzlere iletilir [15]. Düğümler ara yüzlerini yapılandırarak herhangi bir anda bir tüme-gönderim adres grubuna dahil olabilirler veya aynı gruptan ayrılabilirler. Ayrıca, tüme-gönderim adresleri yalnızca hedef adresi olabilirler. Tüme gönderim adreslerinin ilk sekiz biti FF değerini taşıdığı için kolaylıkla diğer adreslerden ayrılırlar [4]. Tüme-gönderim adresini yapısı Şekil 2.8'de gösterilmiştir [15].

İmler alanı dört ayrı imden oluşmuştur. *O* imi gelecekteki kullanım için ayrılmıştır [15]. *P* imi atanan tüme-gönderim adresinin ağ ön kimliğine bağlı olup olmadığını belirler [10]. *R* imi buluşma noktası bilgisinin (RP) tüme-gönderim adresine gömülü olup olmadığını belirler [11]. *T* imi ise atanan tüme gönderim adresinin ön tanımlı (IANA[2] tarafından) olup olmadığını ifade eder. Faaliyet alanı (scope) bölümü ise tüme-gönderim grubunu sınırlandırmak için kullanılan 4 bitlik bir alandır [15].



Şekil 2.8. Tüme-gönderim IPv6 adres yapısı

[2] IANA : A.B.D.'de bulunan, Internet numaraları ve adlarının yönetimini gerçekleştiren kuruluş.

2.5. Komşu Keşif Mesajları

Düğümler, aynı bağ üzerindeki komşularının bağ-katmanı adreslerinin öğrenilmesinde ve hafızaya alınmış eski bilgilerin geçerliliğinin kontrol edilmesinde komşu keşif mesajlarını (Neighbor Discovery messages – ND messages) kullanırlar. Ayrıca, düğümler bağ üzerindeki yönlendiricileri sorgulamak ve bu yönlendiricilerden adres, önek ve diğer yapılandırma bilgilerini almak için de ND mesajlarını kullanırlar. Yönlendiriciler ise bu mesajları paket yönlendirme işleminde hedef düğümün tespiti ile varlıklarını ve yapılandırma bilgilerini düğümlere yayımlamada kullanırlar [28]. IPv6'daki ND mesajları, IPv4'teki ARP, ICMP Yönlendirici Keşif ve ICMP Yönlendirme mesajlarının yerini almıştır [4]. ND mesajları sadece tek bir bağ içinde iletilmek üzere tasarlandığı için bu mesajlarda atlama limit değeri 255 olarak atanır. Eğer mesaj farklı bir bağa geçerse ND mesajının atlama limiti bir azaltılarak 254 değerini alır ve bu mesaj diğer düğümler tarafından göz ardı edilir. Tanımlanmış ND mesajları şunlardır :

- *Yönlendirici Sorgulama (Router Solicitation - RS)* : IPv6 düğümleri tarafından bağdaki yönlendiricilerin bulunması için kullanılır.
- *Yönlendirici Bildirisi (Router Advertisement - RA)* : IPv6 yönlendiricileri tarafından RS mesajına karşılık veya periyodik olarak gönderilir. Bağ ön ekleri, MTU büyüklüğü, otomatik yapılandırmanın kullanılıp kullanılmayacağı ve oluşturulan adreslerin geçerlilik süresi gibi bilgileri içerir.
- *Komşu Sorgulama (Neighbor Solicitation – NS)* : IPv6 düğümleri tarafından komşu düğümlerin bağ-katmanı adresini öğrenmede ve bağ üzerinde çift adreslerin tespitinde (duplicate address detection) kullanılır.
- *Komşu Bildirisi (Neighbor Advertisement - NA)* : NS mesajına karşılık olarak sorgulanan düğüm tarafından kendi bağ-katmanı adresini bildirmek için kullanılır.
- *Yönlendirme (Redirect)* : Yönlendiriciler tarafından hedefe giden yolda daha iyi bir ilk-atlama (first-hop) düğümü bulunduğunu belirtmek amacı ile kullanılır.

2.6 IPv6 Düğümü Otomatik Yapılandırma Süreci

IPv6 düğümlerinin otomatik yapılandırılması süreci RFC 2462'de [12] açıklanmaktadır. Bir IPv6 düğümü sadece yönlendirici cevap mesajlarının kullanıldığı ve bir veya daha fazla ön ek bilgisinin bu mesajlardan elde edildiği durum kontrolsüz adres yapılandırması (1), yönlendirici cevap mesajı içerisindeki Yönetilen Adres Yapılandırma imine (Managed Address Configuration flag) 1 değeri atanması ile DHCPv6 (2), her iki mekanizmanın da beraber kullanılması (3) yollarıyla yapılandırılabilir. IPv6 düğümünün otomatik yapılandırma basamakları şunlardan oluşmaktadır :

- FE80::/64 değerini taşıyan yerel-bağ ön eki ve 64 bitlik ara yüz kimliği kullanılarak geçici bir yerel-bağ adresi oluşturulur ve komşu sorgulama mesajları kullanılarak bağ üzerinde bu adresin eşsizliği test edilir.
- Yerel-bağ adresinin yapılandırılmasından sonra düğüm belirli bir sayıda (varsayılan olarak 3) Yönlendirici Sorgulama Mesajı gönderir.
- Eğer herhangi bir Yönlendirici Cevap Mesajı alınmazsa, düğüm yapılandırma için DHCPv6 kullanır.
- Eğer Yönlendirici Cevap Mesajı alınırsa;
    - Özerk imi (Autonomous Flag) 1 değerini taşıyan her bir Ön Ek Bilgi Opsiyonu (Prefix Information Option) için düğüm, içerilen ön ek bilgisini kullanarak bir adres oluşturur.
    - Eğer Yönetilen Adres Yapılandırma imi 1 değerini taşıyorsa, düğüm adres yapılandırması için DHCPv6 kullanır.

Düğümlerin otomatik yapılandırması şu şekilde özetlenebilir; Yerel-bağ adresinin otomatik yapılandırılmasından sonra geri kalan otomatik yapılandırma işlemi yönlendirici cevaplarına bağlıdır. Yönlendirici cevapları düğümün IPv6 adresi oluşturması için kullanacağı ön ek bilgisini taşıyabildiği gibi düğümün IPv6 yapılandırma bilgisini DHCPv6'dan alması gerektiğini de ifade edebilir.

3. EŞSİZ ALT AĞ ADRESLERİ İLE SIFIR-YAPILANDIRMALI AĞLARIN OTOMATİK YAPILANDIRILMASI

IPv6 düğümleri için otomatik adres yapılandırması IPv6 tanımlamasının bir parçası iken IPv6 yönlendirici yapılandırması hala el ile yapılandırma ve yönetim gerektirir [4]. Bir sitenin veya organizasyonun ISP'den bağlantı hizmeti aldığı durumlarda her ne kadar bir evrensel önek (global prefix) alsa da organizasyon içi yapılandırma hala ağ yöneticilerine kalmaktadır [18-20]. Ağ yöneticisi ağdaki her bir bağ (link) veya bölüt (segment) için el ile yapılandırma yapacak ve ağ büyüdükçe yapılan iş de karmaşıklaşacaktır. Daha kötü bir senaryo olarak ağ yapısı değiştiğinde tüm ağın tekrar gözden geçirilmesi ve tekrar yapılandırılması gerekecektir. Ayrıca, teknik personel çalıştırma olanağı olmayan küçük ev ve SOHO ağlarında [14] gerçek anlamda IPv6 tak-çalıştır mimarisinin gerçekleştirilebilmesi için yönlendiricilerin otomatik yapılandırılması gerekmektedir.

IPv6 ağları tek yönlendiricili ağlar ve çok yönlendiricili ağlar olmak üzere iki şekilde sınıflandırılabilir. Tek yönlendiricili ağlar bir yönlendiricinin bulunduğu ve bu yönlendiricinin farklı bölütleri yıldız topolojisinde birleştirip Internet bağlantısı sağladığı ağlardır (örn. küçük ev ağları). Çok yönlendiricili ağlar ise birden fazla yönlendiricinin bulunduğu ve farklı bölütlerin birbirlerine bağlandığı ağlardır.

Her ne kadar tek yönlendiricili ağların otomatik yapılandırılması basit bir durum olsa da çok karmaşık yönlendiricili ağların otomatik yapılandırılması güç bir iştir. Bu tip ağların el ile yapılandırılması ve yönetilmesi IPv6 kullanımının yaygınlaşmasını engellemektedir. Bir çok farklı bağ katmanı teknolojilerinin kullanılması sebebiyle küçük ev ve SOHO ağlarında birden fazla yönlendirici bulunabilir ve bu yönlendiricilerin el ile müdahale edilmeden otomatik yapılandırılmasını sağlayacak bir protokole gereksinim duyulmaktadır.

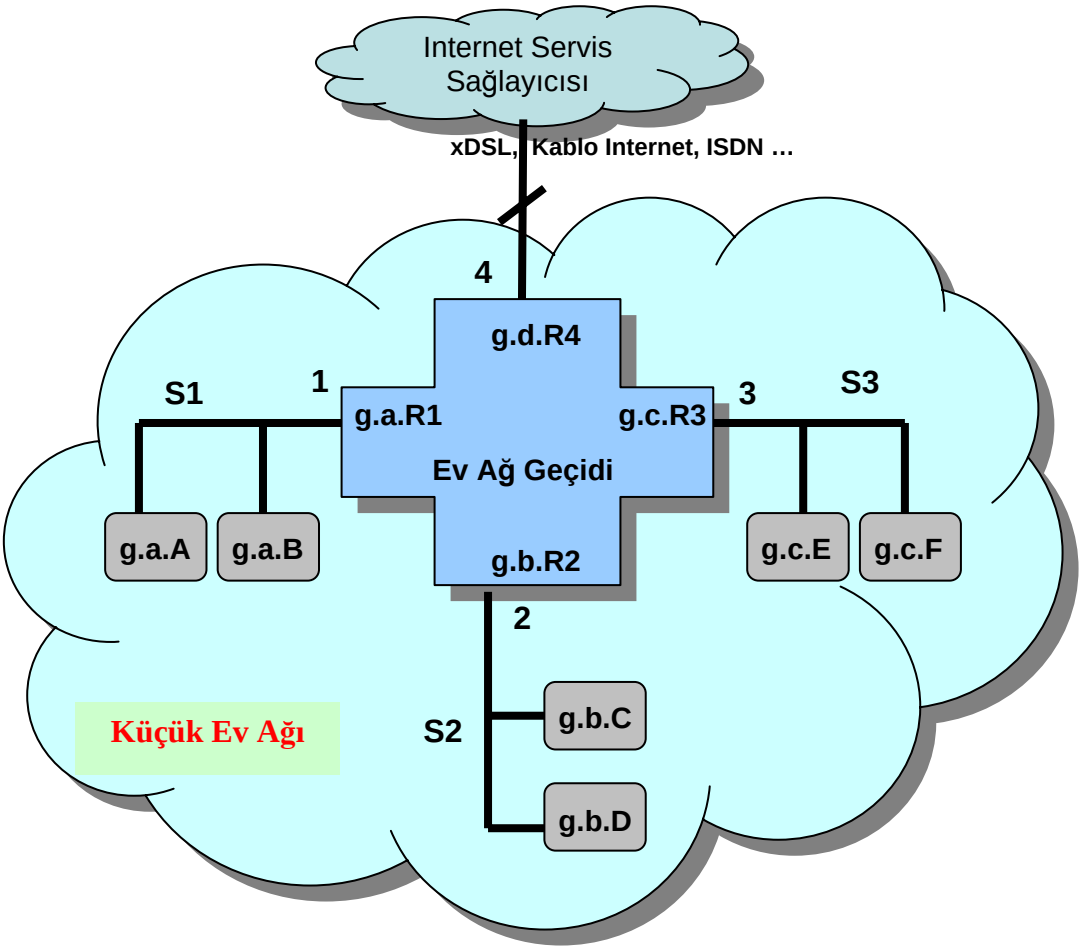
Bu çalışmada tek ve çok yönlendiricili IPv6 ağlarının eşsiz alt ağ adresleri kullanılarak otomatik yapılandırılması hedeflenmiştir. Amaç, başlangıçta her yönlendiricinin her bir ara yüzüne (interface) rastsal bir adres ataması ve sonrasında ağdaki diğer yönlendiriciler ile işbirliği yaparak atanan adreslerin

eşsizliğini (unique) garantilemesi ve topolojik değişmelerde yine birbirleri ile işbirliği yaparak bu eşsizliği devam ettirmeleri ilkesine dayanmaktadır. Yerel adres ataması sırasında bir ağın farklı bölütlerindeki bir veya daha fazla yönlendiricinin aynı adresi ataması sebebiyle IPv6 alt ağ adresi çakışması yaşanabilir. Bu tip çakışmaların yönlendiriciler tarafından tespiti ve çözümü için mevcut iç-ağ yönlendirme algoritmasına yapılacak eklentiler ve bu eklentilerin nasıl çalışacağı belirtilmiştir. Önerilerinin algoritmaların düğümlerdeki IPv6 otomatik yapılandırmasını tamamlayarak IPv6 ağ mimarisine gerçek tak-çalıştır mekanizması getirmesi beklenmektedir.

3.1 Tek Yönlendiricili IPv6 Ağlarının Otomatik yapılandırılması

Tek yönlendiricili ağlar (Sekil 3.1), farklı bölütleri yıldız topolojisinde birleştirip Internet bağlantısı sağlayan bir yönlendiricinin bulunduğu ağlardır. Bu tip ağlara örnek olarak Ethernet, WiFi (802.11), HomePNA, IEEE 1394, Bluetooth gibi birden fazla farklı bölütlerden oluşan ve xDSL, Kablo Internet veya ISDN gibi aldığı Internet bağlantı hizmetini iç ağa paylaştıran ev ağları verilebilir.

Şekil 3.1'de gösterilen yönlendirici 3 iç bölütü 1, 2 ve 3 numaralı ara yüzleri vasıtasıyla birleştirmekte ve 4 numaralı ara yüzü ile Internet bağlantısı sağlamaktadır. Bu ağın otomatik yapılandırılması oldukça basittir. Yönlendirici öncelikle servis sağlayıcıdan *g* evrensel yönlendirme ön ekini alır daha sonra 16 bitlik eşsiz alt ağ kimlikleri olan a, b ve c değerlerini sırasıyla S1, S2 ve S3 iç bölütlerine atar (örn. g.a.::/64, g.b::/64 ve g.c::/64). Yönlendirici daha sonra ilgili bölütlerine atanan ön ekleri bildiren mesajları gönderir. Her bir bölütteki düğümler otomatik IPv6 yapılandırma algoritmasını kullanarak adreslerini yapılandırabilirler. Şekildeki ağda A düğümü *g.a.A* IPv6 adresi ile yapılandırılmıştır. Bu yapılandırılmada *g* 48 bitlik evrensel yönlendirme ön ekini, *a* bulunulan bölüt için yönlendirici tarafından atanmış 16 bitlik alt ağ kimliğini, *A* ise düğümün 64 bitlik ara yüz kimliğini ifade etmektedir.



Şekil 3.1. Tek yönlendiricili IPv6 ağı örneği

Internet bağlantısının olmadığı, örneğin isteğe bağlı olarak izole edilmiş ağlarda yönlendirici evrensel yönlendirme ön eki alamaz ve yerel teke gönderim adresleri kullanmak zorundadır. Bu çalışmada yönlendiricinin öncelikle [17]'de anlatılan algoritmayı kullanarak 40 bitlik bir evrensel kimlik üretmesi ve bu değeri FD00::/8'e ekleyerek 48 bitlik yerel g1 adresini oluşturması öngörülmüştür. Daha sonra yönlendirici 16 bitlik eşsiz alt ağ kimlikleri olan a, b ve c değerlerini sırasıyla S1, S2 ve S3 iç bölütlerine atar (örn. g1.a::/64, g1.b::/64 veg1.c::/64) ve otomatik yapılandırmaya evrensel ön ek olduğu durumdaki gibi devam eder. Eğer daha sonradan bir evrensel yönlendirme ön eki alınırsa sadece *g1* yerel adres ön ekini *g* ile değiştirerek ve yeni oluşturulan g.a::/64, g.b::/64 ve g.c::/64 ön eklerini mesajlarla bölütlere bildirerek aynı ağ alt kimliğini kullanmaya devam eder. Yönlendirici aynı zaman da ön ek yaşam süresini (prefix lifetime) sıfır atayarak

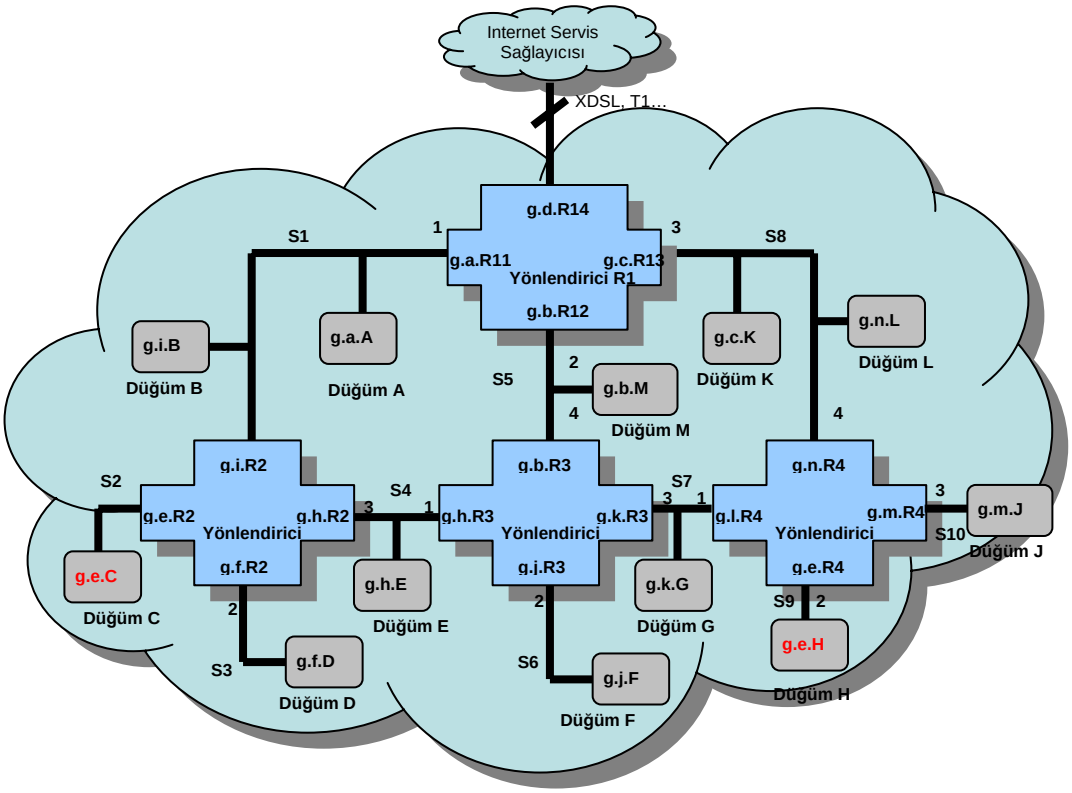
yapacağı yerel teke gönderim mesajlarıyla *g1* ön ekinin kullanımını durdurabilir veya *g1* ön ekini yerel haberleşmede kullanmaya devam edebilir.

3.2 Çok Yönlendiricili IPv6 Ağlarının Otomatik Yapılandırılması

Çok yönlendiricili ağlar birden fazla yönlendiricinin bulunduğu ve farklı bölütlerin birbirlerine bağlandığı ağlardır. Bu ağlarda bir veya birden fazla yönlendirici Internet bağlantı hizmeti verebilir. Herhangi orta büyüklükte bir ağ birden fazla yönlendiriciye sahip olabilir. Şekil 3.2'de 10 ayrı bölütü birbirine bağlayan 4 ayrı yönlendiricinin bulunduğu bir IPv6 ağı örnek olarak verilmiştir. Tipik bir senaryoda (örn. Mühendislik fakültesi) R1 bilgi işlem odasında konumlandırılarak xDSL veya T1[3] bağlantısı üzerinden Internet bağlantı hizmeti vermek üzere yapılandırılabilir. Diğer iç yönlendiricilerin her biri farklı bölümlerde konumlandırılabilir (örn. R2 Bilgisayar Mühendisliğinde, R3 Elektrik-Elektronik Mühendisliğinde ve R4 Çevre Mühendisliğinde). Şekil 3.2'deki gibi sistemin sağlamlığı (robust) ve sürekliliğinin sağlanması için yönlendiricilerin kendi aralarında da bağlantılar olmalıdır. Büyük ölçekli şirketler, üniversiteler ve kamu kuruluşları gibi organizasyon ağlarında onlarca yönlendirici bulunabilir.

Bu bölümde Bölüm 3.2.1.'de anlatılan tek yönlendiricili ağların çok yönlendiricili ağlara nasıl genişletileceği anlatılmaktadır. Şekil 3.2'de görüldüğü gibi, ağın Internet'e bağlı olduğu ve en az bir yönlendiricinin servis sağlayıcı tarafından atanan 48 bitlik bir evrensel yönlendirilebilir ön ek aldığı kabul edilmektedir. Ayrıca bu 48 bitlik evrensel yönlendirilebilir ön ekinin ağdaki tüm yönlendiricilere alan-içi yönlendirme algoritmasıyla yayınlandığı kabul edilmektedir. Otomatik yapılandırma için gereken işlem ise tüm bölütler için 16 bitlik eşsiz ve tutarlı birer alt ağ kimliğinin atanmasıdır. Bu işlemi otomatikleştirmek için bu tezde sunulan yöntemin arkasındaki temel fikir ise yönlendiricilerin başlangıçta her bir ara yüzüne rastsal birer yerel alt ağ kimliği ataması ve ağdaki diğer yönlendiricilerle iş birliği yaparak atanan alt ağ kimliklerinin tüm ağ üzerinde eşsiz olduğunun sağlamasının yapılmasıdır.

[3] T1 : Telefon hattı üzerinden yapılan hızlı Internet bağlantı türü.



Şekil 3.2. Çok yönlendiricili IPv6 ağı örneği

Her bir yönlendirici yerel alt ağ kimliği atamasını kendisi yaptığı için iki veya daha fazla yönlendiricinin farklı bölütler için aynı alt ağ kimliğini ataması olasıdır ve bu durum *Alt Ağ Kimliği Çakışması* (subnetid conflict) olarak adlandırılır. Alt ağ kimliği çakışması çalışır durumdaki iki farklı ağın bir göbek (hub) veya köprü (bridge) ile birleştirilmesi durumunda da gerçekleşebilir. Bir ağın düzgün çalışması için alt ağ kimliği çakışmaları tespit edilmeli ve düzeltilmelidir. Şekil 3.2.'de R1 yönlendiricisinin S1 bölütüne g.e::/64 alt ağ adresi ataması ve R4 yönlendiricisinin de aynı IPv6 alt ağ adresini S9 bölütüne ataması sebebi ile oluşan bir alt ağ kimliği çakışması vardır. Bu çakışmanın tespit edilmesi ve yönlendiricilerden birinin IPv6 alt ağ kimliğini ağ içersinde eşsiz bir değerle değiştirmesi gerekmektedir. Unutulmamalıdır ki faklı yönlendiricilerin bir bölüte aynı IPv6 alt ağ kimliğini ataması durumunda alt ağ kimliği çakışması gerçekleşmez. Aslında bu istenilen bir durumdur. Örneğin şekilde verilen R1 ve R3 yönlendiricileri S5 bölütü için aynı g.b::/64 alt ağ adresini atamışlardır. Ayrıca R1 yönlendiricisinin S1 bölütüne g.e::/64 alt ağ adresi ataması ve R2 yönlendiricisinin

aynı bölüte g.i::/64 alt ağ adresi ataması durumdan oluşan aynı bölüte iki yönlendiricinin farklı alt ağ kimlikleri ataması işlemi bir çakışma sayılmamaktadır. Sorun bir veya daha fazla yönlendiricinin farklı bölütlere aynı alt ağ kimliği ataması durumunda ortaya çıkmaktadır.

Alt ağ adresinin yayınlanması, alt ağ adresi çakışması ve çözümü için tümüyle yeni bir protokol tanımlanabilir. Ancak bu çalışmada var olan bir alan-içi yönlendirme protokolünün bu görevi üstlenmesi düşünülmüştür. Bu seçimin sebebi ise bir sitenin alt ağ adresi atama mekanizmasının el ile veya otomatik olmasına bakılmazsızın site-içi yönlendirme protokolü kullanmak zorunda olmasıdır. Sonuç olarak sisteme yönlendiricilerin kullanacağı yeni bir protokol daha eklemek yerine zaten var olan yönlendirme protokolünü kullanmanın daha akıllıca olduğu görülmüştür. Eşsiz alt ağ adresi atamasında uzaklık-vektör algoritması [23, 24] kullanılabileceği gibi bu çalışmada öneriler genel bağlantı-durum (link-state) veya topolojik tüme gönderim algoritmaları [25, 26] kullanılarak gösterilmiştir.

Bu tez çalışmasında genel topolojik tüme gönderim yönlendirme algoritması, IPv6 protokolü için otomatik yapılandırma eklenerek geliştirilmiştir (Şekil 3.4). Algoritma bir yönlendiricinin (düğüm n olarak ifade edilen) tek bir ara yüzü için tasarlanmıştır. Yani yönlendirici her bir ara yüzü için bahsedilen algoritmayı çalıştırmalıdır. Şekil 3.3'te I.1 ve I.2 basamaklarında yönlendirici başlangıçta ara yüzüne basitçe yerel-eşsiz (locally-unique) alt ağ adresi atamaktadır. Sonraki IV.1 ve IV.2 basamakları komşu listesini ve ilgili bağ için atanmış alt ağ adresini diğer yönlendiricilere yayan periyodik bağ-durum tüme gönderim aşamasını içermektedir. Yönlendirici alt ağ adresi çakışması tespiti ve çözümünü V.1'den V.4'e kadar olan basamaklarda gerçekleştirmektedir. Alt ağ adresi çakışması tespiti ve çözümü şu şekilde gerçekleştirilmektedir :

Yönlendirici diğer bir yönlendirici ile doğrudan bağlı olmadığı *m* ara yüzünden örneğin diğer bir yönlendiricinin $subnetid_m$ alt ağ adresli *n* bağına bağlı bir ara yüzünden bir bağ-durumu paketi aldığında öncelikle V.1.3 basamağında alt ağ adresi çakışması kontrolü yapar. Bir çakışma varsa daha büyük ayraca sahip olan yönlendirici atamış olduğu alt ağ adresini değiştirir yani daha küçük bir ayraca sahip olan yönlendirici kazanan taraf olur.

Algoritmada kullanılan semboller :

s_n : düğüm n'de ki sıra numarası – uçucu olmayan bellekte (non-volatile) saklanır.

subnetid_n : düğüm n'nin alt ağ adresi kimliği – uçucu olmayan bellekte saklanır.

N_n : *n*'ye komşu olan düğümler kümesi.

$w_{n,m}$: $m \in N_n$ için (n,m) bağının ağırlığı.
L_n : $\{(m, w_{n,m}) : m \in N_n\}$.

L_n : *n* tarafından bilinen komşuların listesi.

s_n^m : s_m tarafından düğüm *n*'nin görünümü.

L_n^m : L_m tarafından düğüm *n*'nin görünümü.

subnetid_n^m : subnetid_m tarafından düğüm *n*'nin görünümü.

Düğüm n için eklenti yapılan Topolojik Tüme Gönderim Yönlendirme Algoritması

I. Düğüm n gelirse :
  I.1. Eğer ilk defa geliyorsa
    I.1.1. $s_n \leftarrow 0$, $\text{subnetid}_n \leftarrow$ A eşsiz alt ağ kimliği.
  I.2. $N_n \leftarrow \emptyset$, $L_n \leftarrow \{n\}$
  I.3. Tüm çalışan komşu bağları getir.
II. Komşu bağ (n,m) giderse :
  II.1. N_n'den *m*'i sil.
III. Komşu bağ (n,m) gelirse :
  III.1. $w_{n,m} \leftarrow (n,m)$ için ölçülen ağırlık.
  III.2. N_n'e *m*'yi ekle.
IV. Periyodik olarak :
  IV.1. $s_n \leftarrow s_n + 1$
  IV.2. N_n'nin bütün komşularına $(n, s_n, \text{subnetid}_n, L_n)$ gönder.
V. Düğüm n, $(m, s, \text{subnetid}, L)$ mesajı alırsa :
  V.1. Eğer ($m \notin L_n$ veya $s_n^m < s$) ise
    V.1.1. Eğer ($m \notin L_n$) ise *m*'yi L_n'ye ekle.
    V.1.2. $(m, s_n^m, \text{subnetid}_n^m, L_n^m) \leftarrow (m, s, \text{subnetid}, L)$
    V.1.3. Eğer (subnetid = subnetid_n ve $m \notin N_n$ ve $m < n$) ise
      V.1.3.1. subnetid_n = yeni bir alt ağ adres kimliği /*Çakışma*/
    V.1.4. N_n içerisindeki tüm komşulara $(m, s, \text{subnetid}, L)$ mesajını gönder.

Şekil 3.3. IPv6 yönlendirici otomatik yapılandırması için topolojik tüme gönderim yönlendirme algoritması.

IV.1 basamağına göre kaybeden yönlendiricinin göndereceği sonraki paket daha büyük sıra (sequence) numarasına sahip olacağı için ağdaki bütün düğümler yeni atanan alt ağ adresini kullanmaya başlayacaklardır. Daha fazla topolojik değişikliklerin oluşmayacağı düşünülen, toplam *k* tane ara yüzün, toplam *e* bölütün bulunduğu çok yönlendiricili bir ağın, hiçbir alt ağ adresi çakışması bulunmayan ve bütün yönlendiricilerin döngüsüz bir şekilde tüm IPv6 düğümlerine ulaşabildiği bir yapıya kavuşması için O(*k* x *e*) adet mesaj iletimi gerekmektedir. Hesaplanan sayı şu şekilde açıklanabilir: Her hangi bir IPv6 alt adres çakışması durumunda en küçük ayraca sahip olan yönlendirici ara yüzünün her zaman kazandığı görülür. En küçük ayraca sahip olan ara yüzden gönderilen mesajların diğer tüm ara yüzlere ulaşması için *e* adet mesaj gönderilmesi gerekmektedir. Her bir yönlendirici ara yüzü bu mesajı aldığında her hangi bir çakışma durumunu tespit eder ve sorunu giderir. Bu işlem tamamlandıktan sonra en küçük ayraca sahip olan ara yüzün yapılandırması bir daha değişmeyecektir. Bir sonraki basamakta sıra en küçükten bir sonraki ayraç değerine sahip olan yönlendirici ara yüzüne gelir. Bu ara yüzde yine *e* tane mesaj göndermek sureti ile ağ üzerindeki yapılandırmasını gerçekleştirecektir. Diğer kalan ara yüzlerin yapılandırılmasından sonra sıra son olarak en büyük ayraca sahip olan ara yüze gelir. Sonuçta tüm çakışmalar başlangıçta kaç adet alt ağ adres çakışması olduğuna bakılmaksızın O(*k* x *e*) tane mesaj iletimi ile çözülecektir. Bunun yanında O(*k* x *e*) tane mesaj iletiminden sonra bütün yönlendiriciler ağın tüm topolojik yapısını öğrenmiş olacaklar ve bütün IPv6 adresleri için en kısa yolları kolaylıkla hesaplayabileceklerdir.

Alt Ağ Adresi Çakışma İhtimali : n tane yönlendirici ara yüzü olan, L-bitlik alt ağ kimliği değeri kullanan bir ağ için alt ağ adreslerinin çakışma ihtimali [17]'de verilen

$$p = 1 - e^{\frac{-n^2}{2^{L+1}}}$$

eşitliği kullanılarak bulunabilir. Örnek olarak L=16 ve n = 20 alındığında çakışma ihtimali $p = 3{,}05 \times 10^{-3}$ olarak hesaplanır.

Evrensel Yönlendirme Ön Ekinin Bulunamaması : Ağın Internet'e bağlı olmadığı durumlarda, ağ evrensel yönlendirme ön ekinden de yoksun olacaktır ve bu durumda yerel adresler kullanılmalıdır. Bu durumda iki seçenek vardır : (1) Her

bir yönlendirici kendi 48 bitlik ön ekini oluşturacak ve ağa yayacaktır. Bu seçenekte *n* farklı yönlendiricinin bulunduğu bir ağda *n* tane faklı ön ek olmalıdır. (2) Tüm yönlendiriciler ortak bir ön ekte karar kılacaklar ve tüm ağda bu ön ek kullanılacaktır.

İlk seçenekte her bir *i* yönlendiricisi, ara yüzlerine 16 bitlik yerel eşsiz alt ağ adresi (s_i) atamasını yapmadan önce kendi 48 bitlik evrensel kimlik ön ekini (g_i) oluşturacaktır. Böylece *k* tane ara yüze sahip bir *i* yönlendiricisi her bir *j* ara yüzü için $g_i.s_j$::/64, $1 < j \leq k$ 64 bitlik eşsiz ön eklerini atayacaktır. Bu 64 bitlik ön ekler ağ içerisinde büyük ihtimalle eşsiz olacaktır. Eğer yerel teke gönderim adreslerinde 40 bitlik bir evrensel kimlik alanı kullanılarak 40 bitlik 10 değişik ön ekler oluşturulursa, alt ağ adresi çakışma ihtimalinin $4{,}54 \times 10^{-11}$ olduğunu gösterilmiştir [17].

Anlatılan otomatik yapılandırma yönteminin kullanılmasıyla, ağ içerisinde herhangi bir IPv6 alt ağ adres çakışması ihtimalinin bulunmamaktadır. Ancak bahsedilen eşsiz ön eklerin Şekil 3.3'teki gibi bir yönlendirme protokolü ile yönlendiricilere dağıtılması çözülmesi gerekilen bir problem olarak görülmektedir. Eğer sonradan bir evrensel ön ek *g* kullanılabilir hale gelirse, her bir *j* ara yüzü için her bir yönlendirici $g.s_j$::/64 şeklinde 64 bitlik bir yeni ön ek ataması yaparak IPv6 alt ağlarını yayımlamak için Şekil 3.3'teki algoritmayı tekrar çalıştırır. Bu sayede yeni oluşacak IPv6 alt ağ adresi çakışmalarını da önlemiş olur.

İkinci seçenekte, her bir *i* yönlendiricisi önce kendi 48 bitlik evrensel kimlik ön ekini (g_i) oluşturacak sonra ara yüzlerine 16 bitlik yerel eşsiz alt ağ adresi (s_i) atamasını yapacaktır. Ancak bu sefer yönlendiriciler 16 bitlik alt ağ adresleri üzerinde anlaşmalarının yanında, aynı 48 bitlik yerel ön ekin de tüm ağ içerisinde kullanılması üzerinde anlaşmalıdır. Bu ön ek seçimi Şekil 3.3'teki algoritmaya kolaylıkla dahil edilebilir. Beşinci basamakta yönlendirici bağ-durumu mesajını aldığında öncelikle 48 bitlik ön ek seçimine gider. Yine alt ağ adresi çakışmasının çözümüne benzer bir şekilde, daha küçük ayraca sahip olan yönlendirici kazanır. Ağ durağan hale geldiğinde en küçük ayraca sahip olan yönlendiricinin seçmiş olduğu yerel ön ek tüm sitede kullanılacak ön ek olmuş olur. Sonrasında alt ağ adresi çakışması ve çözümü yeni Şekil 3.3'te gösterildiği

gibi çözülür. Bu yaklaşımın bir avantajı: 48 bitlik evrensel ön ek kullanılabilir olduğunda tüm ağın aynı 16 bitlik alt ağ adresi atamalarını ve 48 bitlik evrensel ön eki kullanarak yeni 64 bitlik ön eklerini oluşturabilmesidir. Böylece Hinden ve Haberman [17] tarafından tavsiye edildiği gibi aynı 16 bitlik alt ağ adresleri hem yerel ön eklerde hem de evrensel ön eklerde kullanılabilir.

4. TEK VE ÇOK YÖNLENDİRİCİLİ IPv6 AĞLARININ ÇOKLU-BAĞ YÖNTEMİ İLE OTOMATİK YAPILANDIRILMASI

Ipv6 ağ yapılandırmasında genel yönelim eşsiz alt ağ adresleri atamaktır [4]. Diğer bir değişle her bir fiziksel bağ için bir ağ adresi kullanılır. Ağ yöneticisi her bağ için eşsiz bir alt ağ adresi ataması yapmalı ve bu alt ağ adreslerinin doğrudan bağlı olan bağlara yayınlanması için yönlendiricileri el ile yapılandırmalıdır. Daha sonra düğümler standart IPv6 durum kontrolsüz otomatik yapılandırma protokolünü [12,13] ve yayınlanan IPv6 alt ağ ön ekini kullanarak IPv6 adreslerini oluştururlar. Şekil 4.1(a)'da bu yönteme bir örnek oluşturmaktadır. Yönlendirici servis sağlayıcıdan g::/48 evrensel yönlendirme ön ekini aldıktan sonra 16 bitlik *a*, *b* ve *c* alt ağ kimliklerinin g.a::/64, g.b::/64 ve g.c::/64 ön eklerini oluşturarak sırasıyla L1, L2 ve L3 bölütlerinde yayınlar. Daha sonra düğümler IPv6 durum kontrolsüz yapılandırma algoritmasını kullanarak kendi IPv6 adreslerini oluştururlar. Şekil 4.1(a)'da A düğümü verilen bilgileri kullanarak kendisine g.a.A IPv6 adresini oluşturur. Bu adresi oluşturan parçalardan *g* : 48 bitlik evrensel yönlendirme ön eki, *a* : L1 için yönlendirici tarafından atanmış 16 bitlik alt ağ adresi ve *A* : düğümünü ara yüz kimliği olarak tanımlanır.

Günümüzde kullanılmakta olan yapılandırma yöntemi el ile müdahale gerektirdiği gibi her bir bağ için eşsiz bir alt ağ ön eki de gerektirmektedir. Bu soruna çözüm olarak ağdaki tüm bağların aynı alt ağ ön ekini kullandığı bir yöntem de düşünülebilir [27]. Bu bölümde eşsiz alt ağ adresleri atama yöntemine alternatif olarak çoklu-bağ yöntemi ile IPv6 ağlarında yapılandırma metodu önerilmektedir. Bu yapılandırma yöntemi ile tüm ağ için tek bir alt ağ ön eki yeterli olmasının yanında el ile yönetim sorununu gidererek, otomatik adres yapılandırmasını kolaylaştırmaktadır.

Bu bölümde öncelikle ev ağları gibi tek-yönlendiricili basit ağlar göz önüne alınarak çoklu-bağ alt ağı desteğinin bu ağda nasıl destekleneceğini gösteren iki ayrı metot anlatılacaktır. Sonrasında ise anlatılan metotların daha karmaşık yapıda olan çok yönlendiricili ağlara nasıl genişletileceği üzerinde durulacaktır. Anlatılan metotların yeni olduğu ve standart otomatik düğüm yapılandırma protokolünü tamamlayarak gerçek tak-çalıştır IPv6 mimarisini oluşturacağı düşünülmektedir. Böyle bir tak-çalıştır yapısındaki yapılandırma metodu ev ve SOHO adresleri için

gereklilik olmanın yanında büyük ağlar için de yönetimsel kazanç sağlayacaktır [14].

Tek bir bağın tek bir alt ağ ön ekine ihtiyaç duyduğu düşünülürse, bağ-katmanında (L2) köprüleme (bridging) yapmanın tek bir bağ oluşturacağı öne sürülebilir. Ancak bu yöntemin iki dezavantajı bulunmaktadır:

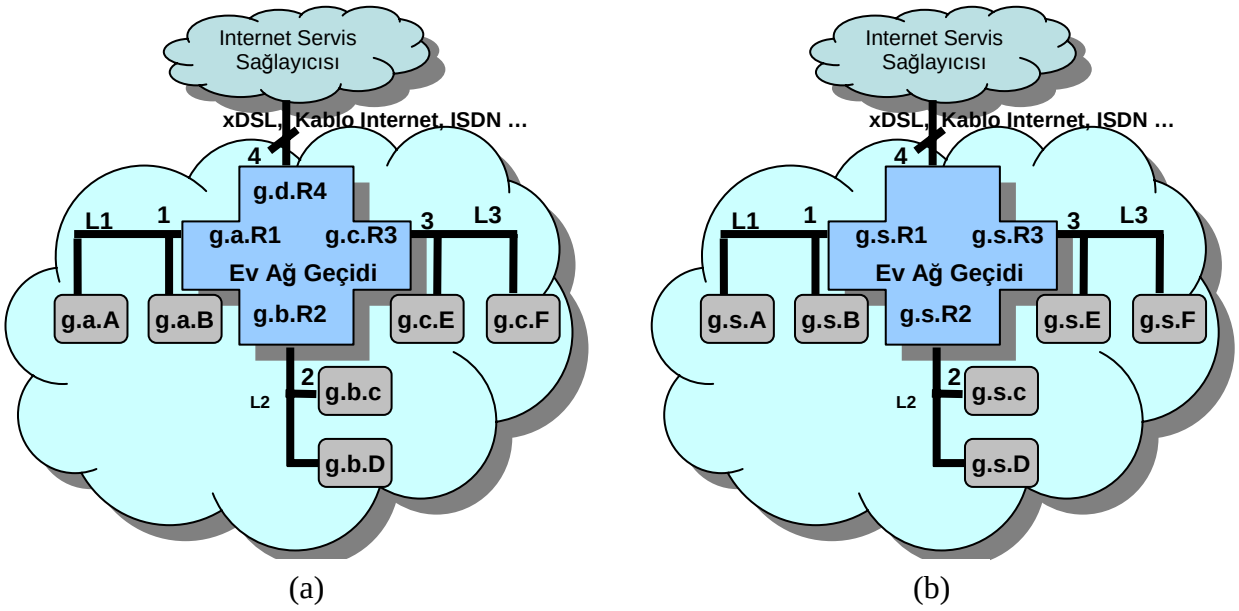
1.) Tüm bağ teknolojileri bağ-batmanında köprülenemez. Örneğin IEEE 802 standardını desteklemeyen iki bağ teknolojisi kullanıldığında veya bu iki bağ farklı MTU'lar kullanılıyorsa köprüleme yapılamayacaktır [27].
2.) L2 köprüleme tek bir çoğa gönderim alanı oluşturur. Bu ise büyük ağlar için istenmeyen bir durumdur. Alternatif bir yaklaşım ise birden fazla bağın üçüncü katmanda örn. IPv6 katmanında köprülenmesidir.

IPv6 katmanındaki çoklu-bağların (multi-link) aynı alt ağ ön eki kullanılması ile oluşturulan ağa çoklu-bağ alt ağı (multi-link subnet) adı verilir. Bu yapılandırma yönteminde yönlendiriciler tüm bağlarda tek alt ağ ön eki yayınlayarak IPv6 katmanında bir çoklu-bağ alt ağı oluştururlar.

Tüm avantajlarına rağmen çoklu-bağ alt ağı desteğinin de getirdiği sorunlar vardır. Özellikle L2 köprülerinde olduğu gibi ağ içerisinde paket yönlendirmesi (packet forwarding) yönlendiriciler tarafından saydam (transparent) olarak gerçekleştirilmelidir. Düğümler çoklu-bağ alt ağında olduklarını bilmemeli ve paket alış verişlerini standart düğüm paket yönlendirme algoritmasını kullanarak gerçekleştirmelidir.

4.1. Tek Yönlendiricili IPv6 Ağlarının Çoklu-Bağ Yöntemi ile Otomatik Yapılandırılması

Tek yönlendiricili ağlar bir yönlendiricinin farklı bölütleri yıldız topolojisinde birleştirmesiyle oluşur. Tek yönlendiricili ağlara örnek olarak Şekil 4.1'deki yapılar verilebilir. Bu tip ağlar birden fazla farklı bölütlerden (örn. Ethernet, WiFi (802.11), HomePNA, IEEE 1394, Bluetooth gibi) oluşur ve xDSL, Kablo Internet veya ISDN gibi Internet bağlantı hizmetleri iç ağa paylaştırılır (örn.



Şekil 4.1. Tek yönlendiricili IPv6 ağları (örn. ev ağı). (a)Yönlendirici L1, L2 ve L3 bağlarına eşsiz g.a::/64, g.b::/64 ve g.c::/64 IPv6 alt ağ adreslerini atamıştır. (b) Aynı ağ yapısı kullanılarak yönlendiricinin tüm bağlara tek g.s::/64 IPv6 alt ağ adresi ataması ile oluşturulan çoklu-bağ alt ağı.

ev ağları). Şekil 4.1(b)'de yönlendirici 4 numaralı ara yüzü ile Internet bağlantısını, 3 iç bölüdü 1,2 ve 3 numaralı ara yüzleri aracılığıyla birleştirerek paylaştırmaktadır.

Tek yönlendiricili IPv6 ağlarında çoklu-bağ alt ağı yapılandırma yöntemi, ağdaki tüm bağları kapsayacak şekilde tek bir IPv6 al ağ adresi ataması kavramını ortaya koymuştur. IPv6 çoklu-bağ fikri Thaler ve Huitema [27] tarafından tanıtılmıştır. Çoklu bağ yapısının bir örneği Şekil 4.1(b)'de gösterilmiştir. Yönlendirici g::/48 evrensel yönlendirilebilir ön ekini aldıktan sonra 16 bitlik alt tek ağ adresi *s*'yi g.s::/64 ön ekini oluşturarak atar. Sonrasında ağdaki belirtilen çoklu-bağa ait tüm düğümler otomatik düğüm yapılandırma algoritmasını kullanarak IPv6 adreslerini oluştururlar.

Bu yöntem tüm ağ için aynı IPv6 alt ağ ön ekini kullanılması avantajının yanında farklı bağlardaki düğümlerin nasıl haberleşeceği gibi sorunları da beraberinde getirmektedir. Düğümlerin çoklu-bağa ait olduklarının farkında olmayıp sorunsuz olarak çalışması en önemli sorundur. Bu sorunun çözümü için üç alternatif sunulmaktadır.

4.1.1 Yönlendiricilerin ND-Proxy gibi Davrandığı Off-Link Modeli

Bu modelde yönlendirici RA'larda yayınlanan g.s::/64 ön eki için otomatik adres yapılandırma imini (autonomous configuration flag) (A) ve bağ-üstünde imini (on-link flag) (L) kapalı konuma (set off) getirir. Böylelikle düğüm dağıtım için bütün paketlerini varsayılan yönlendiriciye gönderir. Bu noktada paketlerin hedef düğüme yönlendirilmesi sorumluluğu yönlendiriciye aittir. Yönlendirici paket yönlendirme işlemini yerine getirebilmek için bütün bağlara NS (Neighbor Solicitement messages) mesajları gönderir ve gelen NA (Neighbor Advertisements messages) mesajlarını kullanarak yönlendirme tablosunu (routing table) günceller. Bu yolla yönlendirici ağdaki bütün düğümlerin konumlarını öğrenir ve bu düğüm yollarını (host routes) kullanarak basit paket yönlendirmesi yapar. Anlatılan yöntemi somutlaştırmak için A düğümünün B düğümüne bir paket gönderdiği senaryoda gerçekleşecek olası hareketler şu şekildedir :

(1) A düğümü paketi yönlendiriciye gönderir.
(2) Yönlendirici L1, L2 ve L3 bölütlerine NS mesajı gönderir.
(3) B düğümü yönlendiriciye NA mesajı ile cevap verir.
(4) Yönlendirici yönlendirme tablosunu (g.s.B, Intf1) bilgisi ile günceller.
(5) Yönlendirici paketi B düğümüne gönderir.
(6) Yönlendirici A düğümüne ICMPv6 Yönlendirme (Redirect) mesajı gönderir.
(7) A düğümü bundan sonraki paketlerini doğrudan B düğümüne gönderir.

Yönlendirici B düğümüne nasıl ulaşacağını da bu süreçte öğrenmiş olur. Bundan sonra ağdaki herhangi bir düğüm B düğümüne paket göndermek istediğinde yönlendirici paketi L1 üzerinden doğrudan B düğümüne yönlendirebilir.

Eğer A düğümü paketi L2 veya L3 bölütlerinde bulunan bir düğüme (örn. C düğümüne) göndermiş olsaydı, C düğümü 3. basamaktaki gibi cevap verecekti. Böylelikle yönlendirici C düğümün konumunu öğrenecek ve paketi basitçe L2 üzerinden C düğümüne yönlendirecekti. Bunda sonra C düğümüne gönderilecek

paketler ise bir daha ND mesajına ihtiyaç duyulmadan doğrudan yönlendirilecektir.

4.1.2. Yönlendiricilerin ND-Proxy gibi Davrandığı On-Link Modeli

Bu modelde yönlendirici RA'larda yayınlanan g.s::/64 ön eki için otomatik adres yapılandırma imini (A) kapalı konuma ve bağ-üstünde imini (L) açık konuma (set on) getirir. Bu durum bir düğüm aynı bağdaki başka bir düğüme bir mesaj göndermek isteğinde karşılık gelen talep edilen çoğa gönderim düğüm adresine (solicited node multicast address) NS mesajları göndermesini gerektirir. Eğer hedef düğüm aynı bağda ise NA mesajsı ile cevap verir ve paketin doğrudan iletimi sağlanır. Ancak düğüm aynı bağda değil ise yönlendirici NS mesajını alır ve karşılık olarak NA mesajı gönderir. Bu durum yönlendiricinin ağdaki tüm düğümlerin konumunu bilmesini ve konum tespitleri için gönderilen tüm NS'leri dinlemesini gerektirir. Yönlendiricinin ağdaki tüm düğümlerin konumlarını bildiğini kabul edilirse, A düğümü C düğümü için bir NS mesajı gönderdiğinde yönlendirici L1 üzerinden kendi MAC adresini kullanarak cevap verecektir. Bu yöntem teorik olarak çalışabilir olarak görünse de anlatılan yönteme iki nedenden dolayı önerilmemektedir: (1) yönlendiricinin ağdaki bütün düğümlerin konumlarına ilişkin bilgiyi nasıl oluşturacağı açık değildir, (2) yönlendiricinin bu bilgiyi elde ettiği kabul edilse bile bütün ara yüzlerinde gelen talep edilen çoğa gönderim düğüm adresi mesajlarını dinlemesi ve bunlara cevap vermesi pratikte olası değildir.

4.1.3. Yönlendiricilerin DHCPv6 Sunucusu Çalıştırdığı Off-Link Modeli

Yönlendiricinin bir istek gelmesi durumunda düğümün adresini öğrendiği 4.1.2.'deki yöntem yerine yönlendiricinin DHCPv6 sunucusu çalıştırdığı bir yapı önerilebilir. Bu yöntemde yönlendirici RA mesajlarındaki yönetilebilir adres yapılandırması imini açık konumuna getirerek, RA yayınını alan tüm düğümlerin yapılandırmalarını DHCPv6 sunucusu kullanarak gerçekleştirmelerini sağlar. Düğümler DHCPv6 tarafından yapılandırılırken yönlendirici de yönlendirme

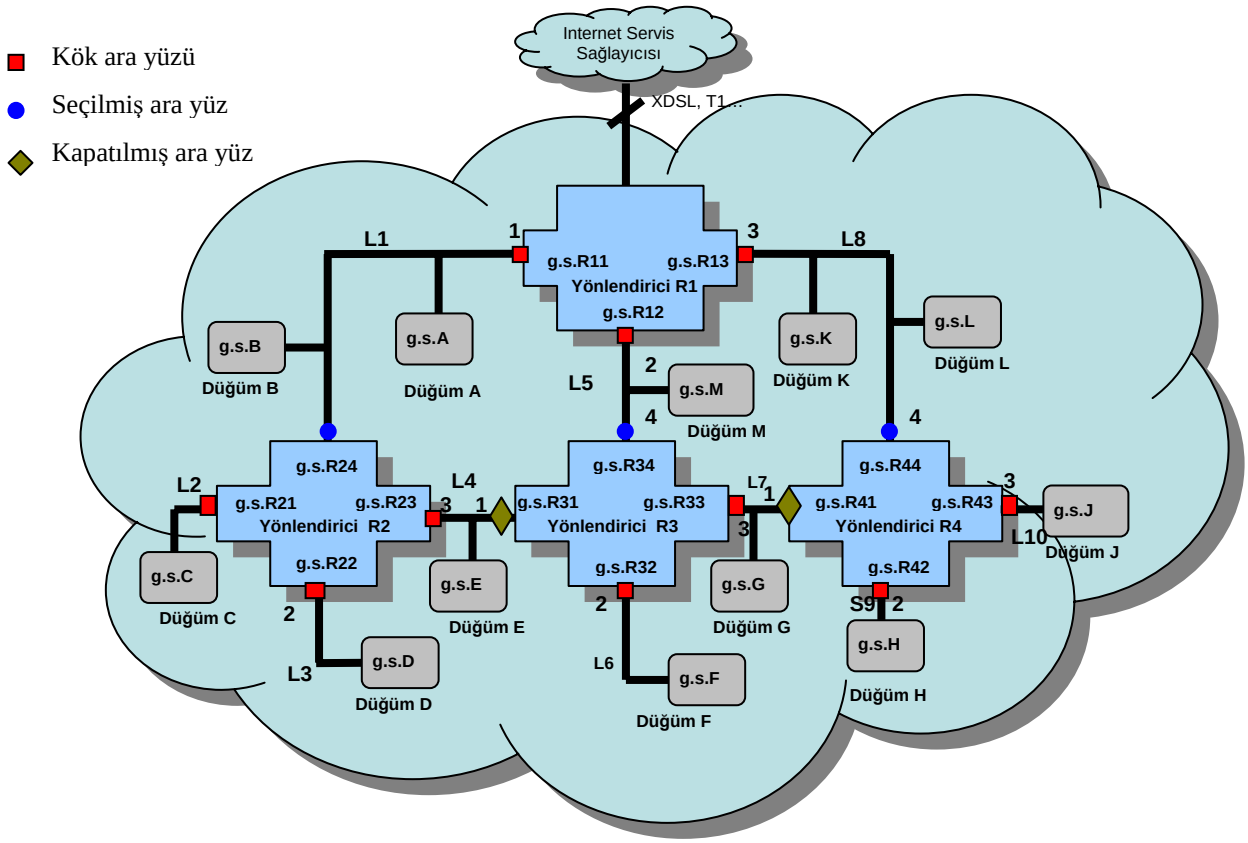
tablosunu günceller. Bu sayede yönlendirici A ve B düğümlerini L1 bölütünde, C ve D düğümlerinin L2 bölütünde ve E ve F düğümlerinin L3 bölütünde bulunduğunu öğrenir. Kullanılan g.s::/64 ön eki bağ-dışı (off-link) olarak işaretlendiği için tüm paketler yönlendiriciye gelir ve yönlendirici yönlendirme tablosundan yararlanarak paketleri hedeflerine iletir. Ayrıca, eğer bir düğüm bir bağdan diğerine geçerse DHCPv6 sunucusu tarafından tekrar yapılandırılmalıdır. Böylece yönlendirici değişikliği kendi yönlendirme tablosunda da güncelleyebilir.

4.2. Çok Yönlendiricili IPv6 Ağlarının Otomatik Yapılandırılması

Birden fazla yönlendiricinin bulunduğu ve farklı bölütlerin birbirlerine bağlandığı ağa çok yönlendiricili ağ adı verilir. Bu tip ağlarda bir veya birden fazla yönlendirici Internet bağlantı hizmeti verebilir ve herhangi bir orta büyüklükte ağ birden fazla yönlendiriciye sahip olabilir. Farklı bağ teknolojilerinin kullanılması sonucunda ev ve SOHO ağlarında bile bu bağları birbirine bağlayacak birden fazla yönlendirici bulunabilir.

Şekil 4.2'de 10 ayrı bölütü bir birine bağlayan 4 ayrı yönlendiricinin bulunduğu bir ağ örnek olarak verilmiştir. Bu senaryoda g.s:://64 çoklu-bağ adresi ağdaki tüm bağları içerecek şekilde yapılandırılmıştır. Ağdaki her bir alt ağa eşsiz alt ağ adresleri atayarak yapılandırma gerçekleştirilebilir. Ancak, bu yöntem birden fazla alt ağ adresinin yanında el ile yönetim gerektirmektedir. Bu nedenle çok yönlendiricili ağlar için çoklu-bağ adresi desteği önerilmiştir.

Çoklu-bağ adresi desteğinin çok yönlendiricili ağlara genişletilmesi ağdaki yönlendiriciler arasında özel bir işbirliği gerektirmektedir. Bu bölümde ND-Proxy (bkz. bölüm 4.1.1) ve DHCPv6 (bkz. bölüm 4.1.3) yöntemlerinin çok yönlendiricili ağlara genişletilmesi açıklanmaktadır.



Şekil 4.2. Tüm ağın g.s::/64 çoklu-bağ alt ağ adresi ile yapılandırıldığı çok yönlendiricili IPv6 ağı örneği

4.2.1. Yönlendiricilerin ND-Proxy Gibi Davrandığı Off-Link Modeli

Bu modelde yönlendiriciler g.s::/64 ön eki için RA mesajlarını otomatik adres yapılandırma imini (A) ve bağ-üstünde imini açık konuma getirerek yayınlarlar. Böylece bütün düğümler paketlerini dağıtım için yönlendiriciye gönderirler. Yönlendirici ise paketlerin dağıtım işini üstlenir.

Tek yönlendiricili ağlarda paket dağıtımı kolay olmasına karşı çok yönlendiricili ağlarda yönlendiricilerin özel bir işbirliği yapması gereklidir. Tek yönlendiricili ağlarda yönlendirici tüm bağlarına NS mesajı gönderir ve gelen NA mesajları vasıtasıyla düğümün konumunu bulur. Eğer NA mesajı gelmezse, bu hedef düğümün ilgili bağda bulunmadığını ifade eder. Çok yönlendiricili ağlarda ise NS mesajının gelmemesi hedef düğümün doğrudan bağlı olan bağlarda konumlanmadığını ifade eder. Ancak hedef düğüm, ağ içerisindeki başka bir bağ içerisinde olabilir. Örnek olarak Şekil 4.2'de A düğümünün H düğümüne bir

paket gönderdiği varsayılsın. A düğümü paketi varsayılan yönlendiricisi olan R1 yönlendiricisine iletir. R1 yönlendiricisi L1, L5 ve L8 bölütlerine H düğümünün talep edilen çoğa gönderim düğüm adresine NS mesajı gönderir. R1 yönlendiricisi NS mesajına karşılık hiçbir NA mesajı almamasına rağmen, H düğümünün diğer bölütlerde (örnekte L bölüdünde) konumlanmış olma ihtimali nedeniyle H düğümünün varolmadığından emin olamaz. Bu durumda R1 yönlendiricisi NS mesajının ağdaki tüm bölütlerde yayınlandığından emin olmak için komşu yönlendiricilerin hepsine aynı NS mesajını gönderir. R1 yönlendiricisi tarafından gönderilen ilk NS mesajının, H düğümüne ait talep edilen çoğa gönderim düğüm adresine gönderilen mesajları dinlemedikleri için R2, R3 ve R4 yönlendiricileri tarafından işlenmeyeceğine dikkat edilmelidir. Bu sebeple R1 yönlendiricisi H düğümünün MAC adresini soran diğer bir NS mesajını R2, R3 ve R4 yönlendiricilerine özellikle göndermelidir.

NS mesajlarının ağ üzerinde dağıtımına özellikle dikkat edilmelidir. Eğer her yönlendirici NS mesajlarını basitçe komşularına gönderir ve aynı algoritmayı NS mesajlarının kabulünde de kullanırsa, NS mesajları ağ içerisinde sonsuz döngüye girebilirler. Bu tür döngülerin yok edilmesi için bu çalışmada Şekil 4.2'de görüldüğü gibi Perlman'nın Kapsayan Ağaç (Spanning Tree) algoritması kullanılmaktadır.

Kapsayan Ağacın Algoritmasının Hesaplanması : Yönlendiriciler öncelikle kendi aralarında bir kök düğüm (root node) seçerler. Bu seçim genelde en küçük ayraca sahip olan yönlendiricinin kök olarak seçilmesine dayanır. Örnekte R1 yönlendiricisi kök düğüm olarak seçilmiştir. Diğer kök olmayan yönlendiriciler kök yönlendiriciye en kısa yollarına ait olan ara yüzlerini kök ara yüzü olarak işaretlerler. Örnekte R2, R3 ve R4 yönlendiricilerinin 4 numaralı ara yüzlerini R1 kök yönlendiricisinden sadece bir atlama (hop) uzaklıkta oldukları için kök ara yüz olarak seçmişlerdir. Bir sonraki basamakta ağdaki her bir bölüt için sadece bir yönlendirici atanmış yönlendirici olarak tayin edilir. Atanmış olan yönlendirici bölütteki paket trafiğinin yönlendirilmesinden ve kök yönlendiriciye en kısa yolun sağlanmasından sorumludur. Atanmış yönlendirici seçiminde yine en küçük ayraca sahip olan yönlendirici kazanır. Örnekte R2 ve R3 yönlendiricilerinin her ikisi de L4 bölütüne bağlı olmasına karşın daha küçük ayraca sahip olan R2

yönlendiricisi atamış yönlendirici olarak belirlenmiştir. Diğer tüm bağlardaki atanmış yönlendiriciler aynı yöntemle tayin edilir. Yönlendiriciler daha sonra atanmış yönlendirici olarak belirlendikleri bölütlere bağlı ara yüzlerine atanmış ara yüzler olarak işaretlerler. Geri kalan bütün ara yüzler o ara yüzden her hangi bir paket trafiğinin kabul edilmeyeceği anlamına gelen "Kapatılmış Ara Yüz" olarak işaretlenirler.

Otomatik Yönlendirici Yapılandırması ve Paket Yönlendirme: Açılış aşamasında tüm yönlendiriciler yukarıda anlatılan kapsayan ağaç algoritmasını çalıştırırlar. Internet bağlantısına sahip olan yönlendiriciler ayraçlarına en küçük değer atayarak kök yönlendirici olmayı garantilerler. Ayracın en ön önemli biti (most significant bit) Internet bağlantısını ifade etmektedir. Sıfır değeri Internet bağlantısının olduğunu yani sınır yönlendiricisini (edge router), bir değeri ise ağ içi yönlendiriciyi ifade etmektedir. Bu kuraldan hareketle sınır yönlendiricisi her zaman ağ içi yönlendiriciden daha küçük bir ayraca sahip olacaktır. Kapsayan ağaç algoritmasının tamamlanmasından sonra bile tüm yönlendiricilerin yönlendirme tabloları boş olacaktır. Yönlendiriciler algoritmanın tamamlanmasından sonra kapalı olamayan ara yüzlerine g.s::/64 ön ekini kullanarak IPv6 adresleri atayacaklardır. Yönlendiriciler ayrıca atanmış yönlendirici olarak belirlendikleri bölütlerde g.s::/64 ön ekini bağ-içi imi açık konumda olarak yayınlayacaklardır. Böylelikle atanmış yönlendiriciler düğümler için varsayılan yönlendirici haline gelirler ve bağdaki düğümler paketlerini atanmış yönlendiriciye gönderirler. Yönlendirici bir düğümden paket aldığında öncelikle kendi yönlendirme tablosuna bakar. Eğer yönlendirici tablosunda hedef için bir giriş varsa paket bu girişe göre yönlendirilir. Eğer bir giriş yoksa yönlendirici kapalı olmayan ara yüzlerinden NS mesajı göndermeye başlar. Eğer bir NA mesajı gelirse yönlendirici basitçe yönlendirme tablosunu güncelleyerek paketin iletimini gerçekleştirir. Bir NA mesajı gelmezse NS mesajı kapsayan ağaç sınırlarından (spanning tree edge) komşu yönlendiricilere gönderilir. Komşu yönlendiriciler NS mesajı tüm bağlara ulaşıncaya kadar aynı algoritmayı çalıştırırlar. Bir NS cevabı geldikten sonra mesajı alan yönlendirici yönlendirme tablosunu güncelleştirdikten sonra cevap verir. Bir örnek bu konuyu daha iyi

açıklayacaktır. A düğümünün H düğümüne bir paket göndermek istediği senaryo göz önüne alınırsa, iletişim şu basamaklardan oluşacaktır :

(1) A düğümü paketi R1 yönlendiricisine iletir.
(2) R1 yönlendiricisi L1, L5 ve L8 bölütlerine NS mesajı gönderir.
(3) Eğer bir NA mesajı gelmezse R1 yönlendiricisi H düğümü için NS mesajını R2, R3 ve R4 yönlendiricilerine gönderir.
(4) R2 yönlendiricisi L2, L3 ve L4, R3 yönlendiricisi L6 ve L7, R4 yönlendiricisi L9 ve L10 bölütlerine NS mesajı gönderir.
(5) H düğümü R4 yönlendiricisine NA mesajı gönderir. R4 yönlendiricisi yönlendirme tablosunu (g.s.H, Intf2) girdisi ile günceller ve R1 yönlendiricisine NA mesajı gönderir.
(6) R1 yönlendiricisi yönlendirme tablosunu (g.s.H, Intf3, NextHop = R4) girdisi ile günceller ve paketi R4 yönlendiricisine iletir.
(7) R4 yönlendiricisi paketi H düğümüne iletir.
(8) Bir sonraki sefere R1 yönlendiricisi H düğümüne gönderilecek olan paketleri ND mesajlarını kullanmadan doğrudan R4 yönlendiricisine iletir.

4.2.2 DHCPv6 Sunucusu Kullanılan On-Link Modeli

Yönlendiricinin NS mesajlarını inceleyerek hedef düğümün konumunu öğrendiği yöntem yerine yönlendiricinin düğümlerin DHCPv6 sunucusu ile otomatik yapılandırılması sırasında düğüm konumlarını öğrendiği bir yapı önerilebilir. Bu yapıya göre öncelikle yönlendiriciler Bölüm 4.2.1'de anlatıldığı gibi kapsayan ağ algoritmasını çalıştırırlar. Sonrasında kök yönlendirici DHCPv6 sunucusunu, diğer yönlendiriciler ise DHCPv6 vekillerini (proxy) çalıştırırlar. Yönlendiriciler yönetilebilir adres yapılandırma imini set ederek düğümlerin IPv6 yapılandırmalarını DHCPv6 aracılığı ile gerçekleştirmelerini zorunlu kılarlar.

Bir düğümün IPv6 adresi yapılandırması içim DHCPv6 kullandığı düşünülürse öncelikle *dhcpdiscover*[4] mesajı göndermesi gerekir. Eğer düğüm kök yönlendiriciye doğrudan bağlı olan bir bölütte ise (örn. düğüm A) R1 yönlendiricisi bu mesajı alacak ve adres yapılandırması genel DHCPv6 protokolü

[4] İstemcilerin ağdaki DHCP sunucularını sorgulamak için gönderdiği mesaj.

mesajları iletimi ile sağlanacaktır. Adres yapılandırmasının sonucunda R1 yönlendiricisi A düğümünün L1 bölütünde bulunduğunu bilecek ve yönlendirme tablosunu buna göre güncelleyecektir. Eğer düğüm kök yönlendiriciye doğrudan bağlı olmayan bir bölütte ise (düğüm H gibi) *dhcpdiscover* mesajı DHCPv6 vekil sunucusu çalıştıran bir yönlendirici tarafından alınacak ve kapsama ağacı üzerinden kök yönlendiriciye yönlendirilecektir. Mesaj DHCPv6 sunucusuna geldiğinde düğüm için bir adres ayrılacaktır. Sonrasında R1 yönlendiricisi yönlendirme tablosunu güncelleyerek DHCPv6 vekil sunucusu çalıştıran R4 yönlendiricisine *dhcppack*[5] mesajı gönderecektir. R4 yönlendiricisi ise kendi yönlendirme tablosunu güncelledikten sonra *dhcppack* mesajını H düğümüne iletecektir. Böylelikle adres yapılandırması sırasında kök yönlendirici ile düğüm arasındaki bütün yönlendiriciler yönlendirme tablolarını güncelleyerek düğüme ulaşım yolunu öğrenmiş olacaklardır. Bütün düğümler adres yapılandırmalarını bitirdiklerinde R1 yönlendiricisi tüm düğümlere ulaşım yolunu bilecektir. Diğer yönlendiriciler ise atanmış yönlendirici olarak seçildikleri ağaçlarda bulunan düğümlere ulaşım yollarını öğrenmiş olacaklardır.

Paket Yönlendirme : Yönlendirici bir düğümden paket aldığında öncelikle yönlendirme tablosuna bakar. Eğer hedef düğüm için bir girdi varsa paket doğrudan yönlendirilir. Aksi halde yönlendirici paketi kök yönlendiriciye giden yoldaki bir sonraki yönlendiriciye iletir. Paketin iletileceği sonraki yönlendirici kaynak yönlendiricinin kök ara yüzü ile aynı bölütte bulunan atanmış yönlendiricidir. Bu yolla paket ya tüm düğümlerin konumlarının bilindiği kök yönlendiriciye yada hedef düğümün bilindiği bir ara yönlendiriciye kadar iletilecektir. Sonuçta paket hedef düğüme ulaşacaktır. Bu yöntemin daha rahat anlaşılması için bir örnek yararlı olabilir : H düğümünün C düğümüne bir paket gönderdiği senaryoda yönlendiricilerin yapacağı işlemler şöyle sıralanabilir :

(1) H düğüme paketi R4 yönlendiricisine iletir.
(2) R4 yönlendiricisi C düğümüne nasıl ulaşılacağını bilmediği için paketi R1 yönlendiricisine iletir.
(3) R1 yönlendiricisi kök yönlendirici olarak C düğümüne R2 yönlendiricisi

[5] DHCP sunucularının istemcilere yapılandırma bilgisini gönderdiği cevap mesajı.

üzerinden ulaşılabileceğini bilir ve paketi R2 yönlendiricisine iletir.

(4) R2 yönlendiricisi C düğümünün L2 bölütünde olduğunu bilir ve paketi C düğümüne iletir.

Bu yöntem doğru paket iletimi sağlasa da, paketler kapsayan ağacın sınırlarında dolaşmak zorundadırlar. Bu ise, bir paketin hedefe ulaşmak için var olan en kısa yolu seçmek yerine daha uzun olan yolu seçme olasılığı olduğu anlamına gelir. Örneğin, H düğümü G düğümüne bir paket gönderdiğinde paket H-R4-R1-R3-G yolunu izler. H düğünden G düğümüne en kısa yol H-R4-G olmasına rağmen R4 yönlendiricisi 1 numaralı ara yüzü kapsayan ağaç hesaplaması tarafından kapatılmış olarak işaretlendiği olduğu için paketi R1 yönlendiricisine iletir. Bu noktada kapsayan ağaç hesaplaması yapılmadan paket yönlendirme işlemi gerçekleşmesi olasıdır. Bu durum bir sonraki bölümde incelenmektedir.

4.2.3 Tüm Yönlendiricilerin DHCPv6 Sunucusu Çalıştırdığı On-Link Modeli

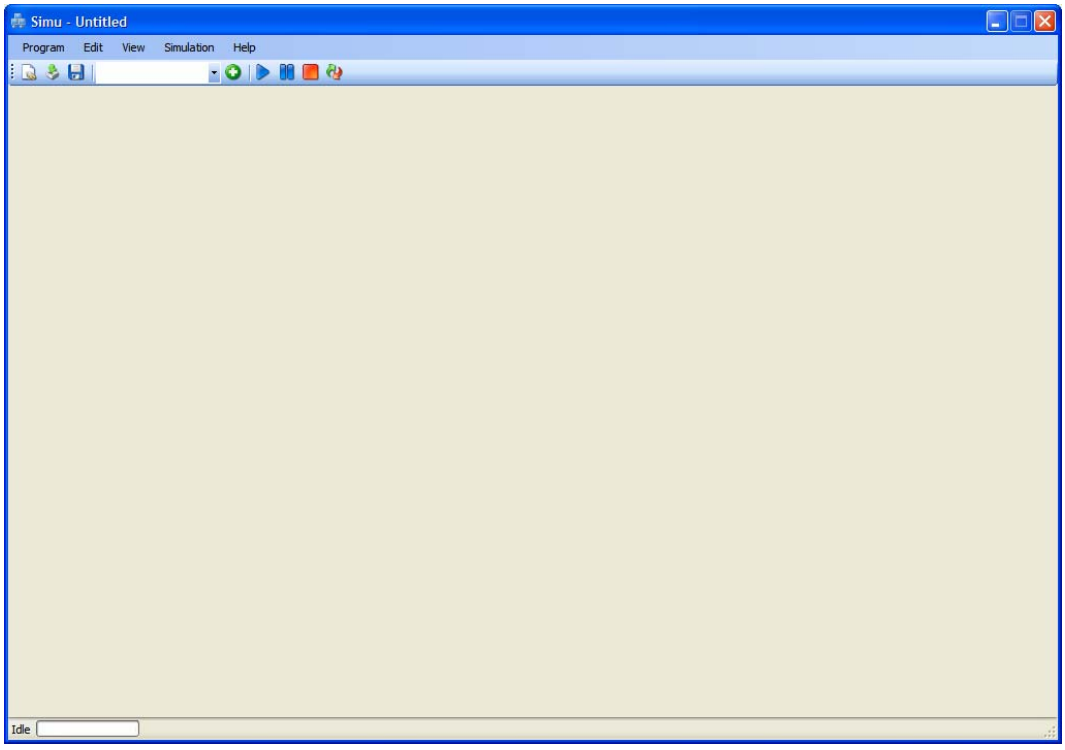
Çoklu-bağ alt ağlarını gerçeklemenin bir yolu da tüm yönlendiricilerin DHCPv6 sunucusu çalıştırmaları ve ağdaki düğümlerin bu sunucuları kullanarak yapılandırmalarını sağlamaktır. Düğümlerin IPv6 yapılandırması sona erdiğinde yönlendiriciler sadece kendilerine doğrudan bağlı olan bölütlerdeki düğümleri öğrenirler ve yalnızca doğrudan bağlı olan düğümlere ait olan paket trafiğini yönlendirebilirler. Tüm ağ içinde yönlendirmenin yapılabilmesi için yönlendiriciler RIP [21] ve OSPF [22] gibi alan-içi yönlendirme algoritmaları çalıştırarak düğüm yollarını (host routes) dağıtmalıdırlar. Bu yöntemle tüm yönlendiriciler ağdaki herhangi bir düğüme en kısa yolu kullanarak nasıl ulaşılabileceğini bilirler. Bu yöntemde yönlendiricilerin kapsayan ağaç hesaplaması yapması gerekmediği unutulmamalıdır.

5. GERÇEKLEME

IPv6 ağları için önerilen otomatik yapılandırma yöntemlerinden eşsiz alt ağ adresleri ile yapılandırma metodu için C# 2.0 dili ile görsel bir gerçekleme (simulation) programı yazılmıştır. Programın amacı dinamik olarak tasarlanan bir ağın otomatik olarak yapılandırılmasının sağlanması ve ardından paket iletimi ile bu yapılandırmanın çalışmasının gözlenmesidir. Programın çalışması için .Net 2.0 uygulama çerçevesinin (framework) kurulu olması gereklidir.

5.1. Program Ara Yüzü

Program çalıştırıldığında Şekil 5.1'de görüldüğü gibi boş bir tasarım ekranı ile başlayacaktır.



Şekil 5.1. Gerçekleme programı başlangıç ekranı

Şekil 5.2'de yeni bir tasarıma başlamak, yapılan tasarımların kaydetmek ve kaydedilmiş tasarımları açmak için kullanılan *Program* menüsü görülmektedir.

Şekil 5.2. Program menüsü

Şekil 5.3'te görülen *Edit* menüsünde tasarımda kullanılan ISP, yönlendirici ve düğüm gibi ağ öğelerini eklemek ve öğeler arasındaki bağları yönetmek için kontroller bulunmaktadır. Tasarım aşamasında öncelikle bu menüden ağ elemanları eklenecek ve eklenen elemanlar arasında istenilen bağlantılar oluşturulacaktır. Eklenen elemanların ve oluşturulan bağların silinmesi veya yerlerinin değiştirilmesi tasarım ekranından gerçekleştirilebilmektedir.

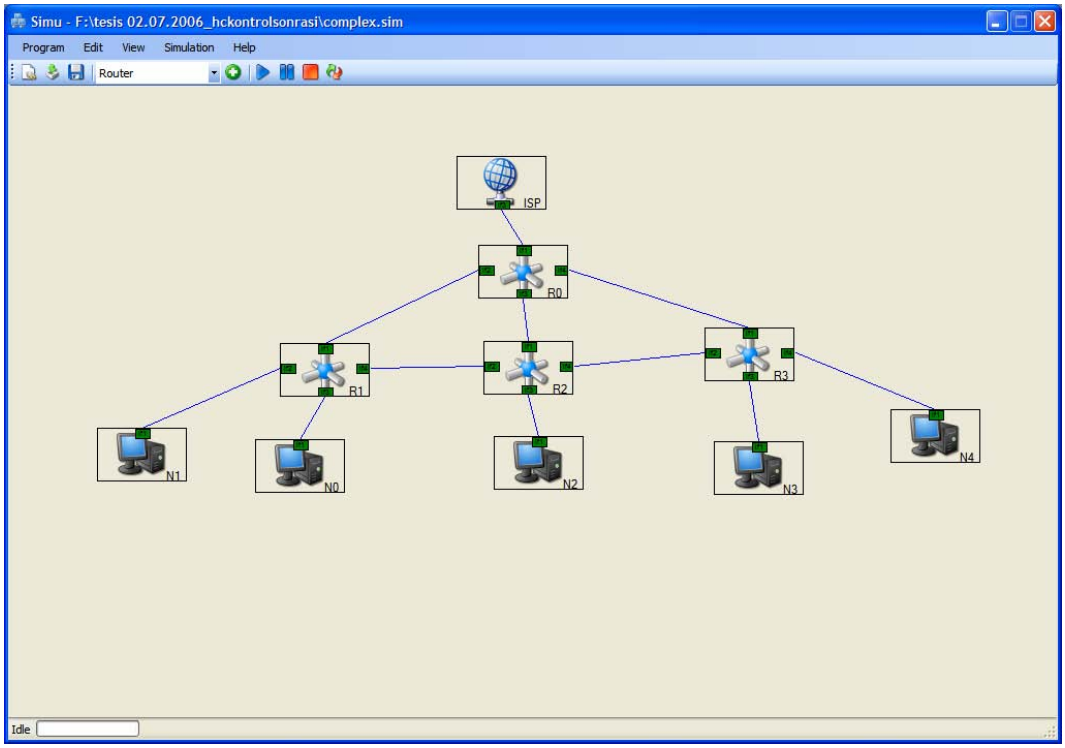
Şekil 5.3. Edit menüsü

Şekil 5.4'te tasarım ekranındaki ağ elemanlarının görünümü ile ilgili ayarların yapılacağı *View* menüsü görülmektedir. Bu menü yardımıyla kısa yol çubuğunun ve tasarım ekranındaki elemanların bağlantı bilgilerinin görüntülenmesi kontrol edilebilir.

Şekil 5.4. View menüsü

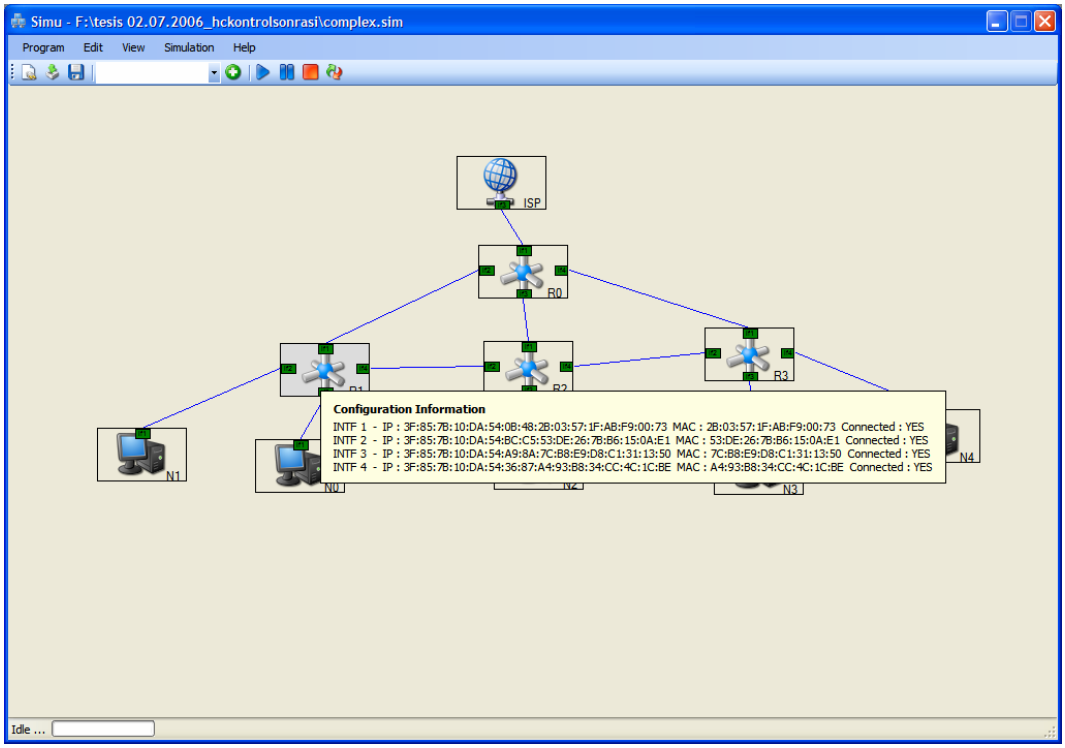
5.2. Programın Kullanılması

Yeni bir tasarım başlandığında öncelikle *Edit* menüsünden veya hızlı erişim çubuğundan ağ elemanları seçilerek tasarım ekranına eklenirler. Daha iyi bir görünüm için eklenen elemanların ekran üzerindeki konumları sürükle-bırak yöntemi ile ayarlanabilir. Eklenen elemanlar arasındaki bağlantı kurmak için ilk elemanın istenilen bağlantı noktasına fare ile tıklanır. Bu nokta bağlantılın ilk ucunu belirlemektedir. Bağlantının diğer ucunu belirlemek için diğer bir elemanın bağlantı noktasına fare ile tıklamak yeterlidir. Bu işlem sonucunda iki ağ elemanı arasında bir bağ oluşturulmuş olur. Tam olarak sağlıklı bir haberleşmenin gerçekleşebilmesi için bir ISP, bir veya birden fazla yönlendirici, yönlendiricilere bağlı düğümlerin tasarım ekranın eklenmesi ve bu elemanlar arasındaki bağlantıların oluşturulması gerekmektedir. Şekil 5.5'te tasarımı tamamlanmış fakat henüz yapılandırılmamış bir ağ yapısı görülmektedir.



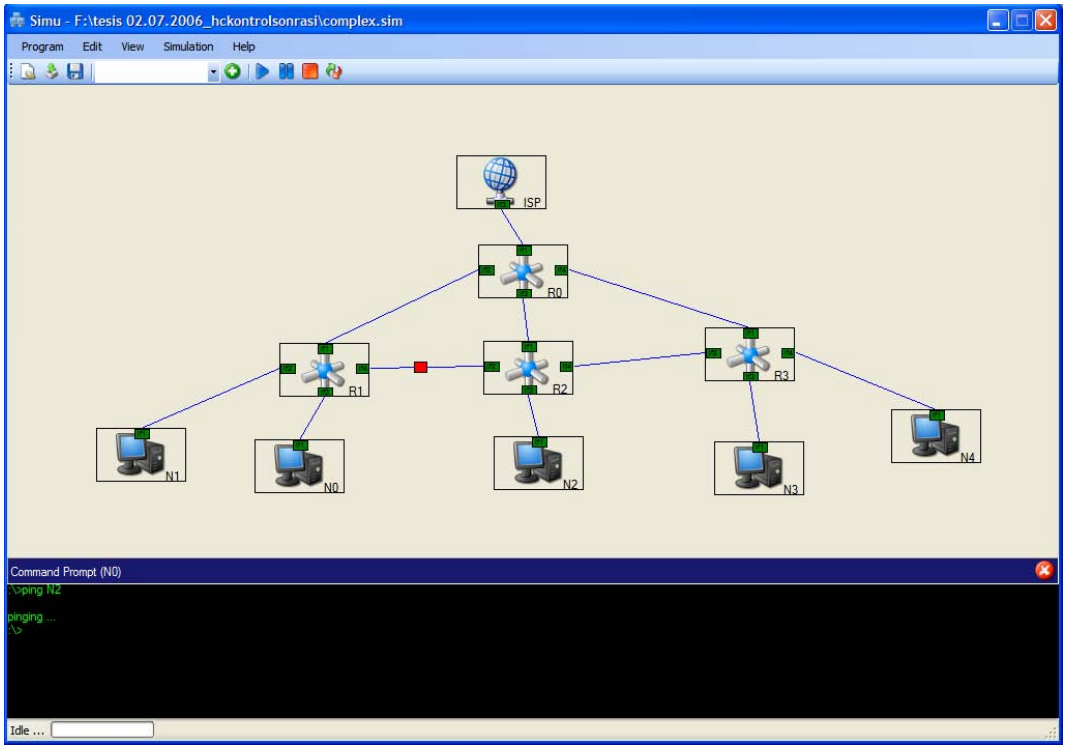
Şekil 5.5. Tasarımı tamamlanmış ağ yapısı

Tasarım işlemini bitirdikten sonra *Simulation* menüsünden *Run* düğmesine basarak yapılandırma gerçeklemesi başlatılmalıdır. Gerçeklemede öncelikle ISP'den elde edilen evrensel yönlendirme ön ekinin yönlendiriciler tarafından paylaşılması ve her bir ara yüz için alt ağ adreslerinin oluşturulması basamakları görüntülenir. Sonrasında her bir yönlendiricinin kendisine bağlı düğümlerin otomatik adres yapılandırmalarını gerçekleştirebilmeleri için yönlendiriciler tarafından RA paketlerini yayınlaması görüntülenir. Düğümler yapılandırmalarını tamamladıktan sonra yönlendiriciler kendi aralarında herhangi bir alt ağ adres çakışması olup olmadığını kontrol ederler. Gerçeklemenin hangi aşamada olduğu tasarım ekranının sol altında görüntülenmektedir. Bütün basamaklar tamamlandığında ağ durağan hale gelir. *View* menüsünden *Show Pop-Up Config Info* seçeneği aktif hale getirildikten sonra ağdaki yönlendiricilerin ve düğümlerin yapılandırması ilgili elemanın üzerine fare imlecini getirerek şekil 5.6'daki gibi görüntülenebilir.



Şekil 5.6. Ağ elemanın yapılandırmasının görüntülenmesi

Adres yapılandırması tamamlandıktan sonra herhangi iki düğüm arasındaki bağlantının testini gerçekleştirmek için istenilen bir düğüm üzerine farenin sağ tuşu ile tıklayarak o düğüme ait konut satırı (command prompt) açılır. Komut satırından *ping* komutu ile görsel olarak paketin yönlendiriciler üzerindeki iletimi izlenebilir. Komut satırından *ping* komutunun çalıştırılması ve paketin iletimi şekil 5.7'de görülmektedir.



Şekil 5.7. Komut satırı ile iki düğüm arasındaki iletişimin kontrolü

6. SONUÇLAR, TARTIŞMA VE ÖNERİLER

IPv4 protokolü Internet üzerindeki baskın protokol olmasına rağmen bir çok konudaki yetersizlikleri nedeni ile yerini IPv6 protokolüne bırakacaktır. IPv4 protokolünden IPv6 protokolüne geçiş süreci birtakım sorunları beraberinde getireceği için kar amacı gütmeyen kurumlar ve ticari şirketler tarafından yavaşlatılmaktadır. Bu geçiş sürecini hızlandırmak ve zorunlu kılmak için bir tetikleyici uygulama beklenmektedir.

Otomatik yapılandırmanın IPv6 protokolünün yaygınlaşması için önemli bir parametre olduğu düşünülmektedir. İstemcilerin otomatik yapılandırılması son kullanıcılar için büyük bir kolaylık getirmekle beraber, yönlendiricilerin otomatik yapılandırmasının standartlaştırması IPv6 protokolünün tak-çalıştır kullanım kolaylığına kavuşmasını sağlayacaktır.

Ev ve SOHO ağlarında farklı bağ teknolojilerinin kullanılması, küçük boyutta olsa bile bu ağlarda birden fazla yönlendirici bulunmasını gerektirebilir. Bu tip ağlarda teknik bilgi gerekmeden otomatik yönlendirici yapılandırma yöntemleri bir gereklilik haline gelmektedir.

Yapılan çalışmada küçük ev ve SOHO ağlarından tek ve çok yönlendiricili ağlarda otomatik yönlendirici yapılandırması için birden fazla yöntem önerilmiştir. Bu yöntemler kullanılarak el ile müdahale olmadan ağ içerisindeki yönlendiricilerin ağ adreslerini paylaşmaları ve gerekli durumlarda çakışmaları tespit ederek gidermeleri olanağı bulunmaktadır.

Yönlendiricilerin ağdaki her bir bölüte eşsiz alt ağ adresini ataması yöntemi tek ve çok yönlendiricili ağlarda otomatik yapılandırmayı kolaylaştıracaktır. Ayrıca bu yöntemde kullanacak algoritma, yeni bir protokol önermek yerine varolan yönlendirme protokollerine yapılacak eklentilerle çalışacaktır. Yönlendiricilerin ağdaki tüm bölütlere aynı alt ağ adresini ataması yönteminin tek bir alt ağ adresi kullanmasının yanı sıra istemcilerin otomatik yapılandırmasını kolaylaştıracağı da düşünülmektedir. Bu yöntemler küçük ev ve SOHO ağları için tasarlanmakla birlikte, isteğe göre bazı değişiklikler ve eklentiler yapılarak daha büyük ve karmaşık iş ve organizasyon ağlarında da kolaylıkla uygulanabilir.

MANET olarak bilinen özel amaçlı mobil ağların dinamik yapısı düşünüldüğünde her bir istemcinin bir yönlendirici gibi ağ içerisindeki paketleri hedefe doğru yönlendirmesi gereklidir. Ayrıca bu ağlarda istemcilerin yapılandırılması başka bir sorundur. Bu nedenlerle yapısal olarak farklı olsa da işlevsel olarak bu çalışmanın uygulama alanı olan otomatik yönlendirici yapılandırması MANET ağlardaki istemci yapılandırması için bir taban oluşturabilir.

KAYNAKLAR

[1] Holdrege, M. ve Srisuresh, P., *Protocol Complications with the IP Network Address Translator*, RFC 3027, 2001.

[2] Feyrer, H., *Introduction to IPv6*, 2001. http://www.onlamp.com/pub/a/onlamp/2001/05/24/ipv6_tutorial.html?page=1

[3] Loshin, P., *IPv6 theory, protocol, and practice*, 2nd edition, Morgan Kaufmann Publishers, San Francisco, 2004.

[4] Microsoft Corporation, *Introduction to IP Version 6*, 2004. http://www.microsoft.com/technet/itsolutions/network/ipv6/introipv6.mspx

[5] Johnson, D., Perkins ve C., Arkko J., *Mobility Support in IPv6*, RFC 3775, 2004.

[6] Crawford, M., *Transmission of IPv6 Packets over Ethernet Networks*, RFC 2464, 1998.

[7] Crawford, M., Transmission *of IPv6 Packets over FDDI Networks*, RFC 2467, 1998.

[8] Deering, S. ve Hinden, R., *Internet Protocol, Version 6 (IPv6)*, RFC 2460, 1998.

[9] Information Sciences Institute, University of Southern California, *Internet Protocol Darpa Iternet Program Protocol Specification*, RFC 791, 1981.

[10] Haberman, B. ve Thaler, D., *Unicast-Prefix-based IPv6 Multicast Addresses*, RFC 3306, 2002.

[11] Savola, P. ve Haberman, B., *Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address*, RFC 3956, 2004.

[12] Thomson, S. ve Narten, T., *IPv6 Stateless Address Autoconfiguration*, RFC 2462, 1998.

[13] Narten, T., *Neighbor Discovery and Stateless Autoconfiguration in IPv6*, IEEE Internet Computing, pp. 54-62, August 1999.

[14] Binet D., *Home Networking: The IPv6 killer application?*, 2002.

http://aristote1.aristote.asso.fr/Presentations/IPv6-2002/P/Binet/Binet.pdf.

[15] Hinden, R. ve Deering, S., *Internet Protocol Version 6 (IPv6) Addressing Architecture*, RFC 3513, 2003.

[16] Hinden, R., Deering, S. ve Nordmark, E., *IPv6 Global Unicast Address Format*, RFC 3587, 2003.

[17] Hinden, R. ve Haberman, B., *Unique Local IPv6 Unicast Addresses*, RFC 4193, 2005.

[18] Troan, O. ve Droms, R., *IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6*, RFC 3633, 2003.

[19] Miyakawa, S. ve Droms R., *Requirements for IPv6 Prefix Delegation*, RFC 3769, 2004.

[20] Thulasi, A. ve Raman, S., *IPv6 Prefix Delegation Using ICMPv6*, draft-arunt-prefix-delegation-using-icmpv6-00.txt, 2004.

[21] Malkin, G., *RIP Version 2*, RFC 2453, 1998.

[22] Moy, J., *OSPF Version 2*, RFC 2328, 1998.

[23] Jaffe, J.M. ve Moss, F.H., *A Responsive Distributed Routing Algorithm for Computer Networks*, IEEE Transactions on Communications, **(7)**, 1758-1762, 1982.

[24] Merlin, P.M. ve Segall, A., *A Failsafe Distributed Routing Protocol*, IEEE Transactions on Communications, **(9)**, September 1979.

[25] Humblet, P.A. ve Soloway, S.R., *Topology Broadcast Algorithms*, Computer Networks and ISDN Systems, **(16)**, 179-186, 1989.

[26] Spinelli, J.M. ve Gallager, R.G., *Event Driven Topology Broadcast Without Sequence Numbers*, IEEE Transactions on Communications, vol. 37, number 5, 1989.

[27] Thaler, D. ve Huitema, C., *Multi-link Subnet Support in IPv6*, Work in progress, 2003.

[28] Narten, T., Nordmark, E. ve Simpson, W., *Neighbor Discovery for IP Version 6 (IPv6)*, RFC 2461, 1998.