

BEYKENT ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

PC TEMELLİ ÇİFT IP DESTEKLİ (IPv4-IPv6)

YÖNLENDİRİCİ BENZETİMİ

YÜKSEK LİSANS TEZİ

Ahmet SÜSLÜ

Tezin Enstitüye Verildiği Tarih : 9 Eylül 2005

Tez Danışmanı : Yrd. Doç. Dr. İbrahim SOĞUKPINAR

Diğer Jüri Üyeleri Prof. Dr. Mahir RESULOV

Yrd. Doç. Dr. Turhan KARAGÜLER

EYLÜL 2005

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜ
YÜKSEK LİSANS TEZ SINAV TUTANAĞI

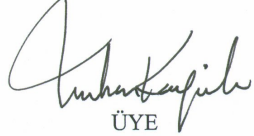
1.6/27/2005

Enstitümüz *Matematik - Bilgisayar Anabilim Dalı Bilgi Teknolojileri Bilim Dalı* yüksek lisans öğrencilerinden BT2251-001 numaralı *Ahmet Süslü'nün* "*Beykent Üniversitesi Lisansüstü Eğitim - Öğretim ve Sınav Yönetmeliği*"nin ilgili maddesine göre hazırlayarak, Enstitümüze teslim ettiği "**PC TEMELLİ ÇİFT IP DESTEKLİ (IPv4-IPv6) YÖNLENDİRİCİ BENZETİMİ**" adlı tezi, Yönetim Kurulumuzun 12.09.2005 tarih ve 2005/11-2 sayılı toplantısında seçilen ve Enstitü binasında toplanan biz jüri üyeleri huzurunda, Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğın 27-c maddesi gereğince (..60..) dakika süre ile aday tarafından savunulmuş ve sonuçta adayın Tezi hakkında *oybirliğı* ile **Kabul** kararı verilmiştir.

İşbu tutanak, 5 nüsha olarak hazırlanmış ve Enstitü Müdürlüğü'ne sunulmak üzere tarafımızdan düzenlenmiştir.


DANIŞMAN
YRD.DOÇ.DR.İBRAHİM SOĞUKPINAR


ÜYE
PROF.DR.MAHİR RESULOV


ÜYE
YRD.DOÇ.DR.TURHAN KARAGÜLER

Önsöz

Internet' in günlük hayatta temel iletişim yolu olması artık bir öngörü olmaktan çıkarak, gerçek haline gelmiştir. Çok yakın bir gelecekte internet kullanımının alışveriş gibi günlük aktiviteler için de geleneksel yöntemlerle giriştiği rekabetten başarıyla çıkacağını tahmin etmek güç değildir. Internet' in varlığı ise en azından başka bir protokol yerini almadığı sürece IP protokolüne bağlıdır. Halen sahip olduğumuz IP iletişimi imkanları geniş ölçüde IPv4 esaslarına dayanmaktadır. Yakın gelecekte ise IPv4' ün eksikliklerini gideren ve bunun yanı sıra belki de şu anda yepyeni iletişim kavramları yaratacak olan IPv6' nın yaygınlaşmasını görebileceğiz. Gerek artmakta olan mevcut IPv4 ağırlıklı internet veri trafiği gerekse ağırlığı artacak olan IPv6 veri trafiği başarılı biçimde yönlendirilmezse halen sahip olduğumuz kullanım konforunda gerileme söz konusu olacaktır. İnterneti canlı bir organizmaya benzetirsek IP yönlendirme damarlara kan akışını sağlamak anlamına gelmektedir. Gerekli veri zamanında ve doğru hedefe yönlendirilmelidir. Bu konuda süregelen tüm araştırmalar daha verimli veri yapıları, daha kısa zamanda daha fazla yönlendirme bilgisini gözden geçirebilen algoritmalar kurmak üzere yapılmaktadır.

Bu çalışmada IPv6' yı, IP yönlendirme kavramı, kimi ilgili veri yapılarını ve algoritmalar araştırılıp anlaşılmaya çalışılmıştır. Üzerine yoğunlaşılacak konu hakkında JAVA kullanarak, çok alışıl gelmiş olmayan bir veri yapısı programlanarak test sonuçları elde edilmiştir.

Beni, bu görece olarak yeni konuda araştırma yapmaya cesaretlendiren, keyifli ve verimli bir çalışma sürdürmeme imkan sağlayan başta danışmanım Sayın Dr. İbrahim SOĞUKPINAR, Sayın Dr. Turhan KARAGÜLER, Sayın Dr. Rifat ÇÖLKESEN' e sonsuz teşekkürlerimi sunuyorum.

Sevgili eşim Saadet, bazen ortak hayatımızı sekteye uğratan tüm çalışma boyunca (aslında yaptığım herşeyde) beni destekledin, teşekkürler.

Ahmet SÜSLÜ

İÇİNDEKİLER

KISALTMALAR	v
TABLO LİSTESİ	vi
ŞEKİL LİSTESİ	vii
SEMBOL LİSTESİ	viii
ÖZET	ix
SUMMARY	x
1. GİRİŞ	1
1.1 Yönlendirme kavramı ve IPv6 kaynaklı sorunlar	2
2. IP YÖNLENDİRME ESASLARI, IPv6' DAKİ ÖZELLİK VE PROBLEMLER, ÇÖZÜM ARAYIŞLARI	5
2.1. IPv4 - IPv6 Geçiş Süreci Stratejileri	5
2.2. Yönlendirme protokolleri, mevcut durum ve IPv6	9
2.3. IP yönlendirme ; şimdi ve gelecek	12
2.4. Sorunlar ve çözüm önerileri	15
2.5 IP karşılaştırma (lookup) işlemi	15
2.5.1. İkili arama ağacı (Binary Search Tree)	16
2.5.2. PATRICIA Trie (PATRICIA veri ağacı)	16
2.6. PATRICIA' nın gerçekleşmesi ve JAVA	17
3. PATRICIA VERİ YAPISI TABANLI YÖNLENDİRİCİ BENZETİMİ	18
3.1. PATRICIA algoritması	18
3.2. Çift yığın (Dual Stack) IP yönlendirmede PATRICIA Trie tasarımı	20
3.3. Tipik yönlendirici mimarisi ve PC benzetimi	22
3.4. RIPng PC benzetimi ve PATRICIA	23
3.5. RIPng algoritması kaba kodu (pseudo code)	24

4. PATRICIA VERİ YAPISI TABANLI YÖNLENDİRİCİ BENZETİMİ, UYGULAMA VE SONUÇLAR	26
4.1. Örnek PATRICIA Trie uygulaması	26
4.2. Test platformları ve test sonuçları	26
4.2.1. Senaryo 1 : 64 bit yapıda tek ağacın kullanımı	27
4.2.2. Senaryo 2 : 64 bit yapıda iki ağacın birbiri ardısıra kullanımı ile 128 bit' lik ağaç elde edilmesi	27
4.3. PATRICIA Ağacı kurulumuna ilişkin test sonuçlarının analizi	28
4.4. Uygulama esasları ve RIPng benzetimi için gerekli kısıtlamalar	29
4.5. Test yöntemi	30
4.6. Uygulama kaynak kodu	31
4.7. Test sonuçları ve analiz	32
5. SONUÇLAR VE TARTIŞMA	34
KAYNAKLAR	35
EKLER	37
ÖZGEÇMİŞ	40

KISALTMALAR

IP	: Internet Protocol
IPv4	: Internet Protocol Version 4
IPv6	: Internet Protocol Version 6
PATRICIA	: Practical Algorithm To Retrieve Information Coded In Alphanumeric
IETF	: Internet Engineering Task Force
GSM	: Global System Mobile Communication
NAT	: Network Address Translate
PAT	: Port Address Translate
QoS	: Quality of Service
RFC	: Request For Comments
DNS	: Domain Name Service
DHCP	: Dynamic Host Configuration Protocol
RIP	: Routing Information Protocol
RIPng	: Routing Information Protocol next generation
RIPv1	: Routing Information Protocol version 1
RIPv2	: Routing Information Protocol version 2
DSL	: Digital Subscriber Line
xDSL	: Çeşitli uygulama biçimleri itibarı ile DSL
ISDN	: Integrated Services Digital Network
ISS	: Internet Servis Sağlayıcı
MAC	: Media Access Control
ICMP	: Internet Control Message Protocol
ICMPv4	: Internet Control Message Protocol version 4
ICMPv6	: Internet Control Message Protocol version 6
PDA	: Personal Data Assistant
TCP	: Transmission Control Protocol
UDP	: User Datagram Protocol
OSI	: Open Systems Interconnection
TCP	: Transmission Control Protocol
MTU	: Maximum Transmission Unit

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 2.1. Kavramsal olarak RIP yönlendirme tablosu içeriği	10
Tablo 2.2. Bazı ağaç erişim yöntemlerinin karmaşıklık karşılaştırması	14
Tablo 4.1. Senaryo 1 için ölçülen zamanlar	27
Tablo 4.2. Senaryo 2 için ölçülen zamanlar	28
Tablo 4.3. Çeşitli veri konfigürasyonları için (A) platformu test zamanları	33

ŞEKİL LİSTESİ

	<u>Sayfa No</u>	
Şekil 2.1.	Genel IPv4 ağı üzerinde IPv6 tüneli şematik gösterimi	6
Şekil 2.2.	Çift yığın kullanan uçbirimlerin IPv4 ve IPv6 ağlarına erişimi	8
Şekil 2.3.	IPv6 Global Unicast adres yapısı	13
Şekil 2.4.	PATRICIA veri ağacı (PATRICIA Trie) örneği	17
Şekil 3.1.	Örnek PATRICIA kök düğümü	18
Şekil 3.2.	Örnek PATRICIA kök düğümüne yeni düğüm ilavesi	19
Şekil 3.3.	Örnek PATRICIA ağacına yeni düğüm ilavesi	19
Şekil 4.1	IP yönlendirici benzetimi uygulaması ekran görüntüsü	32

SEMBOL LİSTESİ

O	: Karmaşıklık
W	: Bit cinsinden veri uzunluğu
n	: Düğüm sayısı
h	: Ağaç yüksekliği
B	: Band genişliği

ÖZET

PC TEMELLİ ÇİFT IP DESTEKLİ (IPv4-IPv6) YÖNLENDİRİCİ BENZETİMİ

Ahmet SÜSLÜ

Mevcut ağlardaki geçiş zorlukları nedeniyle IPv4' ün yerini IPv6' ya bırakması süreci bir anda gerçekleşmeyecek ve her iki protokol belirli bir süre –bazı görüşlere göre daima- birlikte var olacaklardır. Bugün IPv4 üzerinde çalışan IP yönlendirme mekanizmalarının yakın gelecekte IPv6' nın yaygınlaşması ile yetersiz kalacakları tahmin edilmektedir. IP yönlendirme kararının verilebilmesinde temel girdi, IP adresinden hareketle ilgili yönlendirme birimini bulan IP lookup (karşılaştırma, arama-bulma) eylemidir. Bu eylemin IP yönlendirme işlemini darboğaza sokmayacak bir hızda yapılabilmesi gerekir.

Bu çalışmada IPv6' ya geçiş sürecinde aynı omurga üzerinden hem IPv6 hem de IPv4 paketlerini yönlendirmek ilave olarak bunu kabul edilebilir maliyetlere sahip yazılım ve donanım tasarımları ile gerçekleştirmek gerekeceği fikrinden hareketle, IPv4 ve IPv6 yönlendirici tasarımında çekirdek olabilecek bir arama-bulma algoritması incelenmiş ve çift IP destekleyen bir yönlendirici yazılımı kurgulanmıştır. Kurgulanan yazılımın çeşitli platformlarda çalışmasından elde edilen performans değerleri analiz edilmiştir. Sonuç olarak gerek yönlendirme gerekse gelecekte olası başka tiplerde IP bazlı veri arama-bulma hizmetleri için gerekli yüksek performansın elde edilebildiği anlaşılmıştır.

SUMMARY

PC BASED DUAL LAYER IP ROUTER SIMULATION

Ahmet SÜSLÜ

Depending on the problems of transition over present networks, the progress of IPv6 instead of IPv4 will not happen suddenly and both protocols will be together for a while –maybe forever in some resources-. The routing mechanisms working on IPv4 today are supposed to be deficient in near future. The main input to decide the IP routing is IP lookup process which uses IP address to find the related routing unit. This process have to be done with enough speed which will not create a bottleneck for IP routing job.

In this study, a dual routing model has been proposed based on the clue of necessity of routing IPv4 and IPv6 packets together with reasonable cost software and hardware designs, a lookup algorithm which can be core of such design was inspected and a dual IP supporting routing software was simulated. The performance values from simulation running on different platforms were analyzed. As a result, it is concluded that the high performance criteria which is necessary for IP routing and other possible IP based data lookup processes in the future can be achieved.

1. GİRİŞ

İnternet kullanımının yaygınlığının üstel bir hızla artması, “bağlı” cihaz adedi ile “bağlanan” kişi sayısındaki artış, IPv4’ ün gelecekte kullanılmasının daha fazla mümkün olamayacağını işaret etmektedir. IETF 1990’ lı yılların başlarında, üstelik İnternet kullanım oranı bu kadar yüksek değilken adres uzayının tükenmesi olasılığını öngörerek problemi ortaya koymuş ve çözüm için yaklaşık 10 yıllık bir zaman olduğunu belirtmiştir. Buna karşın sorun geniş anlamı ile son yıllarda ilgi çekmeye başlamıştır. 32 bit uzunluğunda IPv4 adres yapısı ile yaklaşık 4 milyar adres sağlanabilmektedir. Bu rakam çok az olmamakla beraber A sınıfı olarak rezerve edilmiş önemli miktarda adresin muhtemelen kullanılmayacak olması kullanılabilir adres sayısını azaltmaktadır[1]. Dolayısı ile 128 bit adres uzayına sahip olması, güvenlik, gezgin IP kullanımı ve yönlendirmedeki ilave üstünlükleri nedeniyle, kullanıcıların IPv6’ya geçmeleri gerekecektir.

Bir diğer sorun, henüz özellikle ekonomik sebeplerle İnternet kullanım oranı artmamış olan başta Çin ve Hindistan olmak üzere gelişme yolundaki kalabalık nüfuslu ülkelerde kullanım oranının artması sonucu ortaya çıkacak IP adresi ihtiyacını karşılamanın pratikte mümkün olamayabileceğidir. Yaklaşık 1,5 milyar nüfusa sahip olan Çin, İnternet kullanımı oranının düşüklüğüne karşın 70 milyon civarındaki İnternet kullanıcısı sayısı ile ABD’ den sonra dünyanın en büyük İnternet kullanıcı nüfusuna sahiptir. Yine Çin, yaklaşık 200 milyon GSM telefonun kullanıldığı ve bunların da en azından belirli bir kısmının potansiyel olarak İnternet’ e erişebileceği bir ülkedir[2].

Yaklaşık 1 milyar kişilik nüfusu olan Hindistan’ da ise merkezi hükümetin uygulamaya koymakta olduğu ilginç bir planı vardır : Ülkede bilgi teknolojilerinden yararlanma düzeyini yükseltmeyi hedefleyen bu proje, başta kırsal kesimde yaşayan öğrenci, çiftçi v.b. alım gücü düşük insanlar olmak üzere yaklaşık 100 milyon ve üzeri kişinin, düşük fiyatlı satılması öngörülen, “Simputer” diye isimlendirilen, İnternet’ e bağlanabilen bir tür cep bilgisayarı sahibi olmasını öngörmektedir. Bu proje aynı zamanda IP adresi ihtiyacının artması anlamına da gelmektedir. Üstelik bu ihtiyaç zaten kalabalık bir nüfusu temsil eden mevcut bilgisayar ve GSM kullanıcılarının üzerine eklenecektir[3].

Yine de IPv6' ya geçişte adres uzayının tükenmesi sorunu tek gerekçe değildir. Üstelik NAT/PAT gibi tekniklerin kullanımı bu sorunla yüzleşmemizi geciktirmiştir ve bir süre daha geciktirebilir. IPv4' ün özellikle Internet' te şu ana kadarki kullanımından elde edilen tecrübelerin ışığında, IP ağlarının gelecekteki muhtemel kullanım alanları da düşünülerek IPv6' nın bazı beklentileri karşılamak üzere de geliştirildiği söylenebilir. Adres alanının artmasının yanı sıra IPv6 kullanımı ile :

- Özellikle noktadan noktaya (peer-to-peer) IP trafiğinde NAT/PAT gibi teknolojilerin artık gerekmemesi ile hız ve güvenliğin artması söz konusu olacaktır. Bu konu çok yakın gelecekte karşılaşmamız olasılığı olan noktadan noktaya mobil IP iletişimi, özellikle mobil kullanıcıları hedef alan duyuru ve reklam organizasyonları gibi yeni elektronik ticaret uygulamaları için de gereklidir.
- Yine noktadan noktaya IP iletişiminde sağlanacak olan kolaylık ve güvenlik, hızlı biçimde yaygınlaşan dağıtık hesaplama (grid computing, distributed computing) uygulamalarının da çalışma biçimlerini etkileyecek ve muhtemelen yaygınlıklarını da arttıracaktır.

Noktadan noktaya veya bir noktadan çok noktaya özellikle multimedia içeriğinin aktarılması sırasında ihtiyaç duyulan yüksek düzeyde QoS desteğinin de IPv6' nın uygulanması ile temini hedeflenmektedir. Bu konudaki zorlukların aşılması “kişiye veya gruba özel yayıncılık” temelli yeni bir medya pazarının oluşmasını sağlayabilir.

IPv6' nın önce mevcut Internet altyapısı ile beraber prototip kullanımı başlamıştır ve yaygınlaşması ile birlikte IPv4 ve IPv6' nın beraber varolduğu bir geçiş dönemi yaşanacaktır. Bu dönem için yapılan öngörüler IPv6' nın yaygınlaşma oranının artması ile değişen çeşitli birlikte varoluş süreçleri içermektedir[1].

IPv6 trafiğinin genel Internet veri trafiği içindeki payı arttıkça, doğal olarak, yönlendirilmesi gereken IP paketlerinin de çoğunluğu IPv6 paketleri olacaktır. Gerek IPv6 protokolünün yönlendirme konusunda kimi yenilikler içermesi gerekse IPv6 adres alanının 128 bit uzunlukta olması yönlendirme işlemleri konusunda halen kullanılan yaklaşımların hiç değişmeden devamının mümkün olmayacağını ortaya koymaktadır.

1.1 Yönlendirme kavramı ve IPv6 kaynaklı sorunlar

IP paketi, üretildiği ağ dışında bir ağa gönderilmek isteniyorsa, ilgili ağa yönlendirilmesi gerekir. Yönlendirme işini yapan cihaza “yönlendirici”,

yönlendiricinin işini yaparken uyduğu kurallar kümesine de “yönlendirme protokolü” adı verilir.

Yönlendirme işlemi gerçekleştirilirken temel girdi IP paketinin alıcı adresidir. IP paketinin alıcı adresi yönlendirici üzerinde çalışan yazılım tarafından incelenip eldeki veriler ile karşılaştırılır ve yönlendirmenin hangi fiziksel arabirime yapılacağına ilişkin karar oluşturulur.

IP yönlendirme IPv4 için genel olarak çözümlenmiş bir sorun olmakla beraber IPv6 yeni bazı sorunlar doğurmaktadır :

- İnternet kullanımı yaygınlaşmaktadır, gelecekte kullanılacak olan IP adreslerinin çoğunluğu IPv6 adresleri olacaktır. Günümüzde bir merkez yönlendirici 40.000 – 50.000 civarı adres girdisi içeren yönlendirme tablolarına başvurmaktadır. Yakın gelecekte 250.000 – 500.000 arası girdi içeren yönlendirme tabloları yaygın biçimde kullanılacaktır.
- Yönlendirme tablolarındaki girdi miktarının artmasının yanısıra IPv6 adresleri IPv4 adreslerinden 4 kat daha uzundur (32 bit’ e karşılık 128 bit). Temel yönlendirici donanım ve yazılımları yaygın biçimde kullanılan 32 bit işlemciler, 32 bit veri yolu erişimi ve 32 bit hafıza erişimi kullanır. 128 bit uzunluktaki IPv6 verisi ile işlem yapmak halen kullanılan yaygın donanımlar ve bu donanımların üzerinde çalışan mevcut yazılımlar için 4 kat fazla çevrim gerektirir. Bu durum en basit yorumla veriye erişen iş süreçlerini oluşturan algoritmaların karmaşıklığı değişmese bile zaman ihtiyaçları 4 kat artmıştır anlamına gelir.

IPv6 temelinde yönlendirme probleminin tekrar ele alınması gerekmektedir. IPv6 ’ ya geçişin ağırlık kazanması ile birlikte IPv6 kullanımından beklentiler artacaktır. Protokolün başarılı biçimde yaygınlaşması ve beklentileri karşılaması konusunda en önemli aşamalardan birisi yönlendirme işinin sorunsuz yapılmasını sağlayacak ilerlemelerin kaydedilmesidir. Protokolü temel alan ticari ve bilimsel servislerin yaygınlaşabilmesi ancak bu şekilde mümkün olabilir. Bu konuda uzun süredir çeşitli çalışmalar yapılmakta ve deneysel bazı sonuçlara ulaşılmaktadır. Temel olarak önerilenler mevcut durumda başarılı bir şekilde hizmet veren veri yapılarının artacak olan iş yükü ve uzayacak olan veri boyu da göz önüne alındığında yetersiz kalacağından hareketle yeni algoritmaların geliştirilmesi gerekliliğidir [4] [5] [6]. IP karşılaştırma işinin karmaşıklığını azaltma ve verimliliğini artırma amaçlanarak önerilerde bulunulmuştur.

Bu çalışmada da IPv4 yönlendirme konusunda yaygın kullanılan kimi yönlendirme protokollerinde verinin organizasyonunda kullanılan bir veri ağacı biçimi olan PATRICIA Trie incelenmiş, örnek protokol olarak ele alınan RIP ve RIPng için benzer karmaşıklıkta bir simülasyon içeren bir ağ yazılımı programlanarak, IP erişim tablosunun bu ağaçta depolanması sağlanmış, IPv4 paketleri ile kapsüllenmiş IPv4 ve IPv6 paketlerini benzeştiren veri yığınları ile performans testleri yapılarak söz konusu yöntemin verimliliği hakkında hükme varılmıştır. Yapılan testlerde IPv4 adreslerinin benzeştirmek üzere 32 bit, IPv6 adreslerini benzeştirmek üzere de 128 bit uzunlukta tamsayılarla çalışılmış ve her iki IP protokolünden kaynaklanan trafik yükü çeşitli oranlarda gerçekleştirilmiştir. Hızlı biçimde yaygınlaşan mobil IP ihtiyacı ve taşınabilir cihaz sayısındaki artış da göz önüne alındığında, kolay taşınabilen ve platform bağımsız bir uygulama kurgulanması hedeflenmiş, bu amaca uygun olarak da JAVA programlama dili kullanılmıştır. Uygulama JAVA ile işaretçi tipi değişken kullanımına da bir örnektir. Uygulamada elde edilen sonuçlar IP yönlendirme gibi yoğun veri işlemeye dayalı bir işlemin kolay bulunabilir ve kabul edilebilir maliyette donanım ve yazılımlar kullanılarak mümkün olabileceğini göstermektedir. Ayrıca benzeri yoğun veri işleme gerektiren işlemler için de (IP paket filtreleme esaslı güvenlik yazılımları, IP yayın izleme ve kayıt...) bir çözüm zemini vaad etmektedir.

2. IP YÖNLENDİRME ESASLARI, IPv6' DAKİ ÖZELLİK VE PROBLEMLER, ÇÖZÜM ARAYIŞLARI

Wang ve arkadaşları [4] tarafından yapılan çalışmada daha verimli bir IP yönlendirme işi için çeşitli algoritmalar karşılaştırılmış, donanım ve yazılımsal yöntemler incelenmiş, erişim ağacı kullanımının veriye erişim için en verimli metod olduğuna hükmedilmiştir. Söz konusu çalışmada yazarlar IPv4 adreslerinin ortak yönlendirilme özelliklerine göre gruplandırıldığı erişim ağacı temelli bir algoritmayı önermişler ve IPv6 adresleri için de uygulanabilirliğini ortaya atmışlardır. Varghese ve arkadaşları tarafından yapılan çalışmalarda [5] [6] mevcut veri yapısı yaklaşımları incelenerek daha iyi karmaşıklık ve daha az zaman kullanımı vaadeden algoritmalar önerilmektedir.

Genel olarak IP yönlendirme özelde de IPv6 yönlendirme konusunda iyileştirme önerilerinde bulunan bahsi geçen örnek çalışmalar ve benzerlerinde ortak nokta, giderek zorlaşan IP yönlendirme işini başarabilmek için verinin mümkün olan en az karmaşıklıkta erişimine olanak sağlayan bir erişim ağacı üzerinde organize edilmesi ve halen kullanılan erişim algoritmalarının yeni ihtiyaçlar göz önüne alınarak geliştirilip amaca daha iyi hizmet eden yeni algoritmaların türetilmesi gerekliliğidir.

2.1 IPv4 – IPv6 geçiş süreci stratejileri[9]

IPv4' ten IPv6' ya kademeli geçiş isteği bazı temel gerekçelere dayanmaktadır :

- Halen yaygın biçimde ve oldukça başarıyla hizmet eden IPv4 ağlarına büyük miktarda para ve emek yatırımı yapılmıştır. Bu ağlarda sağlanan servisleri devam ettirmek ekonomik olarak verimli bir davranıştır.
- IPv4 ağları üzerinde çalışan ilave hizmetlerin (güvenlik duvarları, sanal-özel ağ...v.b. uygulamalar) IPv6 için adaptasyonu ve testi gerekmektedir. Bu alanda da mevcut yatırımın korunmaya ihtiyacı vardır.

Kademeli geçiş için her iki protokolün birlikte var olacağı bir dönem ön görülmektedir. Bu birlikte var oluş döneminde izlenecek stratejiler ana hatları ile aşağıdaki başlıklarda sıralanabilir :

- IPv6 cihazlarının IPv4 cihazları ile iletişimini sağlayabilmek amacı ile protokolün çevirilmesi (Network address and protocol translating = NAPT)

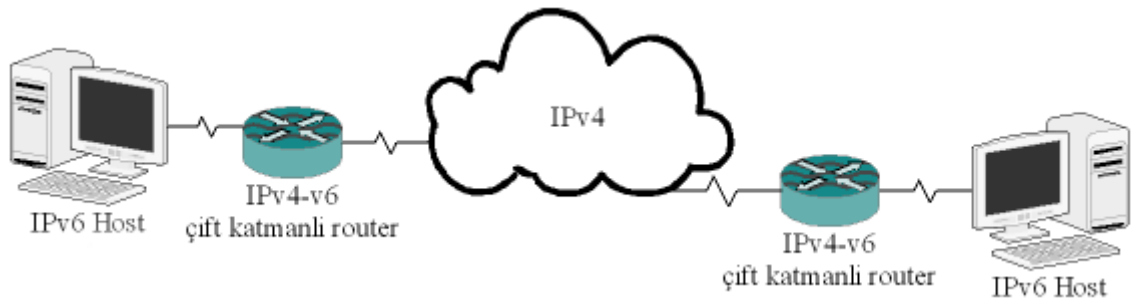
- IPv6 paketlerinin zaten var olan IPv4 alt yapısı üzerinden taşınması (tunnelling)
- IPv6 ve IPv4 protokollerinin her ikisini de kullanabilen çift-yığınlı (dual-stack) cihazların kullanımı.

Bu stratejileri kısaca inceleyecek olursak :

Ağ adresi ve protokolünün çevirisi (NAPT) : RFC 2765 ve 2766 dökümanlarında detaylı biçimde açıklanan bu yaklaşımda temel amaç, IPv6 ağından IPv4 ağına (ve tersi durumda) iletişim gerçekleştirilmek istendiğinde yönlendirmenin sağlanabilmesidir. Bu hedefi gerçekleştirebilmek için ağ cihazı IPv4 ve IPv6 adreslerini eşleştirmeye yarayan bir havuz tutar, her iki protokolde yapılan iletişim isteklerini, diğer protokoldeki eşdeğer ağ adresine yönlendirir.

Tünelleme (Tunneling) : Neredeyse tüm Internet altyapısı ve IP bazlı tüm yerel networklerin IPv4 alt yapısına sahip yatırımlar olduğu düşünüldüğünde, akla gelen bir çözüm metodu, IPv6 verisinin zaten mevcut olan bu alt yapıdan, IPv4 verisi gibi iletilmesidir. Yapılan iş genel olarak IPv6 paketlerinin IPv4 paket başlığı ardından gelen IPv4 veri alanına yerleştirilmesi ve IPv6 verisinin değerlendirileceği noktaya kadar bu şekilde ulaştırılarak, söz konusu noktada IPv6 verisinin ayrıştırılarak kullanılmasıdır.

Tünelleme konusu çeşitli RFC dökümanlarında detaylı olarak ele alınmıştır. Halen kullanılan IPv6 protokol tanımında da tünelleme işine yapısal olarak destek bulunmaktadır. Tünelleme yönteminin detaylı incelenmesi bu çalışmanın kapsamı dışındadır, ancak yönteme ait yapısal bir şema üzerinde açıklama yapmak yararlı olabilir :



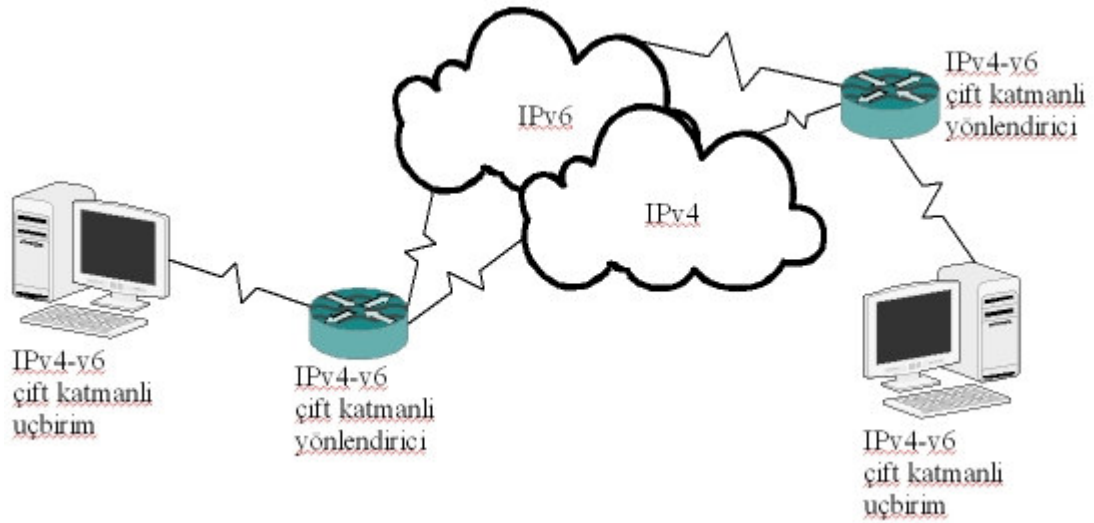
Şekil 2.1 : Genel IPv4 ağı üzerinde IPv6 tüneli şematik gösterimi

Örnek şemada(Şekil 2.1) IPv6 kullanan her iki uçtaki cihaz, alt ağlarını IPv4 ağına bağlayan çift yığımlı yönlendiriciler sayesinde verilerini IPv4 network' ü üzerinden aktarmaktadır.

Her iki uçtaki ağ kullanıcısı birimler (host) sadece IPv6 protokolü ile çalışıyor olabilir. Kullanıcıların yönlendiricilere bağlantısı IPv6 ile, ancak IPv6 tarafından kullanılan bir tünelleme protokolü ile yapılmaktadır. Yönlendiriciler çift katmanlı çalışmaktadır. IPv6 verisini IPv4 başlığı ile donatarak mevcut IPv4 altyapısına yönlendirmektedirler. Tünelleme bir IPv6 yönlendiriciden diğerine bir kurulum gerektirebileceği gibi (manuel tünelleme), IPv6 tarafından yapısal olarak desteklenen otomatik tünel konfigürasyonu mekanizmaları (IPv4 uyumlu IPv6 adreslemesi ve benzeri yöntemler gibi) sayesinde el ile ayarlanan konfigürasyon bilgisi olmadan da IPv4 uyumlu IPv6 yönlendiriciler tarafından gerçekleştirilebilir (otomatik tünelleme).

Tünelleme, çift yığımlı cihaz kullanımını gerekli kılmaktadır. Çift-yığın stratejisinin mevcut yatırımları koruyan bir uygulaması gibi düşünülebilir.

Çift-yığın (Dual-stack) kullanımı : Her iki protokolü de kullanma becerisine sahip ağ cihazı, çift-yığımlı (dual-stack) cihaz olarak adlandırılır. Bu ağ cihazı IPv4 kullanan ağ cihazları ile IPv4, IPv6 kullanan ağ cihazları ile de IPv6 protokolünü kullanarak iletişim kurar. Her iki protokolü de yapısal olarak kullanır. Her iki protokole ait iletişim gereksinimini de yapıdal olarak karşılar. Örneğin IPv4 ağının gerektirdiği DNS/DHCP iletişimi (DNS A kaydına erişim, DHCP yoluyla konfigürasyon gibi) hizmetleri sağlanırken, IPv6 ağına özgü DNS AAAA kaydına erişim, DHCP kullanmadan konfigürasyon gibi eylemleri de yerine getirir. Çift IP yığımlına sahip ağ cihazı (bu cihaz bilgisayar, yönlendirici, güvenlik duvarı.... olabilir) her iki IP protokolü için de tam bir destek vermektedir.



Şekil 2.2 : Çift yığın kullanan uçbirimlerin IPv6 ve IPv4 ağlarına erişimi

Karşılaştırma : Ana hatları ile ortaya konan bu birlikte varoluş stratejileri çeşitli noktalarda birbirleri ile karşılaştırılıp avantaj ve handikapları ortaya konabilir. Sıralayacak olursak :

- **Tünelleme yöntemi**, IPv6 protokolü içinde yapısal olarak desteklenmektedir. Mevcut IPv4 alt yapısı üzerinde çalışması ön görüldüğü için en az yatırımla IPv6 cihazlarını yaygın IPv4 ağları üzerinden iletişime geçirme konusunda ucuz ve kolay bir çözüm olarak kabul edilebilir. İki uzak ağ veya uzak cihaz, aradaki yönlendiriciler veya IP paketinin eriştiği ağlar özel bir şekilde yapılandırılmamışsa bile IPv4 üzerinde tünellenmiş IPv6 ile iletişim kurabilirler. Bu uygulama konusunda bir tür esneklik yarattığı için yöntemin avantajlı yanı olarak kabul edilebilir. Ancak IPv6 paketi ağlar arasında yolculuğu boyunca IPv4 paketi içinde olacağı için yönlendirme kuralları IPv4 uyumlu olacaktır. Bu durum IPv6 için geliştirilmiş yönlendirme mekanizmaları veya IPv6 paketine özgü kullanım kolaylıkları gibi gelişmelerden yoksun kalınması anlamına gelir. Örneğin bu yöntem IPv6' nın yaygınlaşması ile gündeme gelmesi düşünülen yüksek QoS beklentisi hedefi için uygun değildir. Ayrıca IPv6 paketlerinin üretildikleri noktada IPv4 ile sarmalanması (kapsüllenmesi) ve kullanılacakları noktada da IPv4 paketlerinden ayıklanmaları ilave işlemci gücü ihtiyacı ve zaman kaybı anlamına gelmektedir.
- **Çeviri (NAPT)**, tünellemeye benzer biçimde IPv6 protokolünün ön gördüğü yenilik ve geliştirmeleri kullanamamak anlamına gelmektedir. Genel öngörü,

hiçbir alternatif çözümün bulunamadığı durumlarda bu yöneme başvurulmasıdır.

- **Çift yığın (Dual-stack)** barındıran ağ cihazları, şüphesiz en esnek ve gelecekte de en yaygın kabul görececek olan iletişim çözümleridir. Her iki IP protokolünün de ön görülmüş tüm özelliklerinden yararlanmak ancak çift yığınla cihazlarla mümkündür. Ancak bu esneklik ve kullanılabilirlik karşılığında ödenmesi gereken bir bedel vardır : Her iki protokolün aynı cihazda bulunması, ağ cihazının daha fazla işlemci gücü ve hafızaya sahip olması, eğer ağ cihazı bir yönlendirici ise her iki protokol için ayrı yönlendirme tabloları tutulması ve buna benzer biçimde bazı işlerin iki kez yapılması anlamına gelir. Buna karşılık her iki protokole de kusursuz biçimde destek elde edilir.

2.2 Yönlendirme protokolleri, mevcut durum ve IPv6

Hem Internet' in yaygınlaşması hem de IP protokolünün yerel ağlarda da başlıca iletişim protokolü haline gelmesi endüstrinin yönlendirme işi konusunda büyük bir birikim ve tecrübe edinmesine yol açmıştır. Kullanılabilen yönlendirme protokolleri geliştirilmiş, yeni sürümleri oluşturulmuştur. IPv6' nın yaygınlaşması beklentisi ise yönlendirme protokollerine bu yeni perspektif ile bakılarak yeni nesil IP protokolüne uygun biçimde geliştirilmeleri çabalarına sebep olmuştur.

Bir yönlendirme protokolü esas olarak yönlendirici cihazın, elindeki IP paketini doğru rotaya sevk edebilmesi için kullandığı yönlendirme tablosuna ilişkin güncellemeleri elde ederek uygulamasını sağlar. Aynı yönlendirme protokolünün dili ile anlaşan yönlendiriciler, yönlendirme tablolarında olagelen değişiklikleri birbirleri ile değiştirerek birbirlerinin yönlendirme tablosunun güncel tutulmasını sağlarlar.

Örnek protokol, RIPng[9] : RIP (Routing Information Protocol), IPv4 ağları arasında yönlendirme sağlamak amacı ile yaygın biçimde kullanılan bir yönlendirme protokolüdür. Yetkili bir AS (Otonom Sistem)' in kontrol ettiği bir alt ağlar kümesinde, alt ağlar arası yönlendirme işinin gerçekleştirilebilmesi için kullanılır. Bu tür yönlendirme protokolleri "Interior Gateway Protocols" olarak adlandırılır, RIP de bunlardan biridir. RIP' in IPv6 yönlendirmesi için uyarlanmış sürümü RIPng (RIP next generation) olarak adlandırılır.

RIPng (tanımlanışı : RFC 2080, Ocak 1997) Bellman-Ford algoritması olarak da bilinen uzaklık vektörü algoritmasına dayanır ve pek çok kavramsal yönünü öncülleri olan RIPv1 ve RIPv2' den almıştır. RIPng mesajları da IPv6 ile taşınır. RIP

yönlendiriciler duyurularını, alıcı adresinin FF02::9 olduğu, “tüm RIP yönlendiricilerin çoklu yayın adresine” yaparlar[1].

Ana hatları ile RIP(ng) protokolü ile işleyen yönlendiricinin çalışma biçimi şöyle açıklanabilir :

- Yönlendirici RIPng güncelleme mesajlarını kullanarak, yönlendirme bilgisini ve doğrudan bağlı olduğu komşularını periyodik biçimde yayınlamaktadır.
- Benzeri mesajları da almaktadır. Yönlendirici doğrudan bağlı olduğu komşusundan protokolün öngördüğü bir güncelleme mesajı aldığı anda, mesaj işleminden geçirir, yönlendirme tablolarını günceller.

Tipik bir RIPng yönlendirme tablosu aşağıdaki gibidir :

Tablo 2.1 : Kavramsal olarak RIP yönlendirme tablosu içeriği

IPv6 adres ön ek	Protokol	Yönlendirme arabirimi	Sonraki yönlendirici	RIPv6 Metrik
A/B	C	D	E	F
(...Benzer girdiler...)

A : Bir IPv6 adres ön ekidir. 128 bit uzunluğa kadar ulaşabilir, örnek olarak (örnekler gerçektir) : 200a::0000:0109 , fe80:0104::0000:0e07 ... v.b.

B : A ön eki için hedef adresin ön ek uzunluğudur. “Hedef adresin B bit’e kadarki kısmı, içeriği ne olursa olsun. A girdisi kullanılarak yönlendirilebilir” olarak yorumlanabilir.

C : Protokol. Yönlendirme tablosunda ilgili satır için tesbit edilmiş yönlendirme protokolü bilgisini içerir.

D : Yönlendirme arabirimi, veri paketinin hangi arayüze aktarılacağı bilgisini içerir.

E : Sonraki yönlendiricinin IPv6 adresi.

F : RIPv6 metrik değeri, hedefin yönlendiriciye kaç “hop” (atlama) uzaklıkta olduğunu gösterir.

Ayrıca her yönlendirme tablosu girdisi; girdinin tabloya eklendiği ya da güncellendiği andan bu yana geçen zamanı gösteren bir **zamanlayıcı**, tablo girdisinin yakın zamanda değiştiğini gösteren bir **yol değişim bayrağı**, girdinin hangi yolla

(RIP mesajı veya başka bir yönlendirme protokolü mesajı ya da el ile girilmiş) tabloya girdiğini gösterir bir **yol kaynağı bilgisi** de yönlendirme bilgisine eşlik eder.

Geçiş süreci, birlikte yaşama ve IP yönlendirmenin ilişkisi : Gerek mevcut Internet kullanımının hızlı artması gerekse buna eklenecek olan IPv6 trafiği göz önüne alındığında, Internet üzerindeki IP trafiği hakkında bazı çıkarımlarda bulunmak mümkün olmaktadır :

- Hem mevcut durum hem de gelecek, IP adres sayısının artmasını, bu adreslerin önce küçük sonra da çok büyük bir bölümünün IPv6 adreslerinden oluşmasını işaret etmektedir.
- Bu içerikteki IP trafiğinin yönlendirilebilmesi için en uygun çözüm, önceki bölümde de tartışıldığı üzere, her iki IP sürümünün de “dilinden anlayan” çift IP yığınlı yönlendiriciler kullanmaktır.

Özellikle belli tarihten önce devreye alınmış yönlendiricilerin sürece uyum sağlamak üzere yazılım (ve donanım) açısından güncellenmesi gerekecektir ki bu zaman ve para açısından ciddi bir yatırım demektir. Bu yatırım yapılsa bile mevcut arama/bulma algoritmaları kullanılarak IPv6 trafiğini yönlendirmeye çalışmanın verimsiz sonuçlar doğuracağı iddia edilmektedir[4,5].

Göze alınması gereken bir diğer zorluk birim zamanda daha çok miktarda IP paketi üreten geniş bandla Internet’ e bağlı kişisel kullanıcıların sayısında da olağanüstü bir artış olmasıdır. TV kablosu, uydu kanalı, xDSL, ISDN v.b. erişim metotları ile kullanıcılar gittikçe daha uzun süre ve daha önemlisi daha çok veri alış verişi yaparak Internet kullanmaktadırlar. Bu, mevcut durumda da özellikle merkezi noktalardaki (ISS, Telekom Operatörü...) yönlendiricilerin ağır iş yüküne maruz kaldıklarını gösterir. Ayrıca ülkeler ve kıtalararası geçiş linklerinin gittikçe daha yüksek band genişliğini daha ucuza sunuyor olması da IP trafiğine olan talebi arttırmaktadır. Bu durum doğrudan Internet kullanımını hedeflemeyen ancak bu linklerden veri taşınması sonucunu doğuran IP Telephony ve Video Konferans gibi uygulamaların gittikçe daha çok yaygınlık kazanmasının da bir sonucudur (tabii sebebidir de). IPv6’ nın da yüksek etkinlik kazanması ile IP yönlendirme konusunda kritik noktalarda oluşacak olan darboğazı aşmak için yönlendirme işini gittikçe daha hızlı ve daha ucuz yapmanın yolunu araştıran çok sayıda çalışma sürmektedir.

2.3 IP yönlendirme ; şimdi ve gelecek

IP yönlendirme konusunda temel zorluk kısıtlı zamanda yoğun miktarda IP adres bilgisinin eldeki referans bilgi ile karşılaştırılıp karar verilmesi gerekliliğidir. Konuya çözüm önerileri getiren çeşitli çalışmaları inceleyecek olursak ; örneğin Wang ve arkadaşları [4] tarafından yapılan çalışmada daha verimli bir IP yönlendirme işi için çeşitli algoritmalar karşılaştırılmış, donanım ve yazılımsal yöntemler incelenmiş, erişim ağacı kullanımının veriye erişim için en verimli metod olduğuna hükmedilmiştir. Söz konusu çalışmada yazarlar IPv4 adreslerinin ortak yönlendirilme özelliklerine göre gruplandırıldığı erişim ağacı temelli bir algoritmayı önermişler ve IPv6 adresleri için de uygulanabilirliğini ortaya atmışlardır. Varghese ve arkadaşları tarafından yapılan çalışmalarda [5] [6] mevcut veri yapısı yaklaşımları incelenerek daha iyi karmaşıklık ve daha az zaman kullanımı vaadeden algoritmalar önerilmektedir.

Genel olarak IP yönlendirme özelde de IPv6 yönlendirme konusunda iyileştirme önerilerinde bulunan bahsi geçen örnek çalışmalar ve benzerlerinde ortak nokta, giderek zorlaşan IP yönlendirme işini başarabilmek için verinin mümkün olan en az karmaşıklıkta erişimine olanak sağlayan bir erişim ağacı üzerinde organize edilmesi ve halen kullanılan erişim algoritmalarının yeni ihtiyaçlar göz önüne alınarak geliştirilip amaca daha iyi hizmet eden yeni algoritmaların türetilmesi gerekliliğidir.

Konuyu ve zorlukları anlamamanın yolu IPv4 ve IPv6 adres yapılarının kıyaslanmasından geçmektedir.

IPv4 adres yapısı : IPv4 adresi 32 bit (4 byte) uzunluktadır ve yaygın olarak birbirinden . işareti ile ayrılmış 4 adet desimal sayı ile gösterilir. Belirli bir tek adresi veya bir alt ağı işaret etmek üzere bu gösterime alt ağ maskesi ifadesi de eklenebilir.

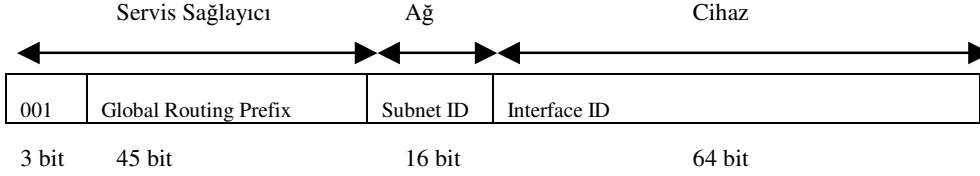
10.0.0.1/255.0.0.0 veya 10.0.0.1/8 gibi

IPv4 adresi A,B,C,D sınıflarından birine ait olacak biçimde yapılandırılır. Kullanılan A,B,C,D sınıfları yönlendirme yapılacak ağı ve alt ağı işaret ederken farklı sayıda bitler kullanılmaktadırlar.

Ancak burada konu açısından temelde dikkate alınması gereken nokta şudur : IPv4 adresi toplamda 32 bitten oluşmaktadır ve yönlendirme bilgisinin saklanacağı bir veri ağacında günümüzün sıradan bilgisayarları tarafından tek bir hafıza erişimi ile okunabilecek kadar yer tutmaktadırlar. Öte yandan IPv6 adresleri 128 bit uzunluğundadır, gösterim biçimi de farklıdır. Çok temel bir farklılık IPv6 adresinin IPv4 adresinden 4 kez daha uzun olması ve modern bir kişisel bilgisayarda 4 hafıza

erişimi ile okunmasının mümkün olmasıdır. Bu ilk bakışta çok ciddi bir handikaptır. [5] numaralı çalışmada da öne sürüldüğü gibi IPv4 adresleri için başarıyla kullanılan IP karşılaştırma algoritmalarının 4 kez daha yavaş çalışmasına da sebep olabilir.

Sorunu yakından incelemek için IPv6 adres yapısını da yakından incelemek gerekir :



Şekil 2.3 : IPv6 Global Unicast Adres Yapısı[1]

Şekil 1’de görülen Global Unicast Adres yapısı, IPv6 ağına bağlı bir cihazı IPv6 ağı üzerinde konumlandırabilmemizi sağlar. Hiyerarşik olarak :

Servis Sağlayıcı

Servis Sağlayıcının hizmet verdiği alt ağlardan biri

Alt ağa bağlı bir cihaz üzerindeki bir ağ arabiriminin adresi

Ağ arabiriminin adresi (Interface ID), arabirimin MAC numarasından türetilmekte ve ilgili alt ağda tek (unique) oluşu garanti edilmektedir.

IPv6 datagramının IPv4 datagramından farklılıkları, IP tabloları üzerinde karşılaştırma (lookup) yöntemlerinden halen kullanılanların, bu yeni perspektif için değerlendirilmesini gerektirmektedir[4,6].

Yönlendirmenin hesabı, yönlendirme tablosundan yararlanmak : IP yönlendirme için elimizdeki temel veri, IP datagramının hedef adresidir. Gerek IPv4 gerekse IPv6 (ve tabii ki diğer yönlendirme katmanı protokolü olan ICMPv4 ve ICMPv6) datagramları paket başlığında bu bilgiyi taşımaktadır. Yönlendirme yazılımı veya donanımı hedef adresini ilgili tablo ile (yönlendirme tablosu) karşılaştırarak hedef ağı bulmaya, hedef ağa ulaşan fiziki yolun bağlı olduğu arabirimi tespit etmeye ve IP/ICMP paketini bu arabirime yönlendirmeye çalışır. Eğer tek bir Geniş Alan Ağ bağlantısı varsa veya Internet omurgasına doğrudan bir bağlantı yoksa (mesela bütün paketleri bir Internet Servis Sağlayıcı(ISS)’ ye yollanıyorsa), bu durum yönlendirme hesabı yapmak gerekmiyor anlamına gelir. Paket ya yerel ağa aittir ya da değildir. Bu durumda yerel ağa ait olmayan her paket varsayılan arabirime yönlendirilebilir. Yapılan iş sadece iletim işidir.

Eğer yönlendirme hesabı yapılacaksa, ciddi miktarda veri ile uğraşılıyor demektir. 40.000 – 50.000 (yakın gelecekte 250.000 ve üzeri) satırlık yönlendirme tabloları olabilir. Böyle bir veri yığını ile uğraşmak için bu konu üzerine optimize edilmiş veri erişim metotlarını kullanmak şarttır.

IP yönlendirme, IP filtreleme ve benzeri işlemler IP paketinin alıcı ve/veya gönderici adresinin bir tablo ile karşılaştırılarak karar verilmesini gerektirir. Karşılaştırmada, uygun en uzun ön ek (longest matching prefix) i bulmaya yarayan bir sayısal sözlük ağacı kullanımı standartlaşmış çözümdür. . Ancak IPv4 adreslerinin 32 bitlik uzunluğuna karşın IPv6 adreslerinin 128 bit uzunlukta oluşu kullanılan veri yapısının uzun anahtar sahaları içermesini zorunlu kılar.

IP adresinden yola çıkarak (key), ilgili verilere (data) ulaşmak için yaygın kabul görmüş bazı ağaç erişim algoritmaları kullanılır. Çeşitli ağaç erişim algoritmaları IPv4 uygulamalarında kullanılmış ve yeterince deneysel sonuç elde edilmiştir. IPv6 uygulamaları içinse öneriler ve denemeler sürmektedir.

Tabii ki çeşitli ağaç erişim algoritmalarının birbirlerine göre üstün ya da zayıf oldukları alanlar vardır. Temel üstünlüğü (ve zayıflığı) başta algoritmanın metodun karmaşıklığı (O) belirler. Algoritmanın ne derece kolay programlanabildiği, patent kısıtlaması altında olup olmadığı da diğer kullanılabilirlik kriterlerindedir.

Aşağıdaki tabloda yaygın kullanılan bazı ağaç erişim algoritmaları için kurulum, arama ve yer ihtiyacı açısından bir karşılaştırma yapılmıştır.

Tablo 2.2 : Bazı ağaç erişim yöntemlerinin karmaşıklık karşılaştırması

Yöntem	Kurulum	Arama	Hafıza
Binary Search Tree	$O(n)$	$O(\log_2 n)$	$O(n)$
Patricia Trie	$O(nW)$	$O(W)$	$O(nW)$
LC Trie	$O(hn)$	$O(h)$	$O(hn)$

n=düğüm sayısı, W=veri boyu (bit), h=ağaç yüksekliği değerlerini ifade etmektedir.

2.4 Sorunlar ve çözüm önerileri

Hızlı paket yönlendirme işlemi ve yönlendirme tablosunun mümkün olan en güncel halde tutulması iki açıdan zamana karşı yarış demektir :

1-Mevcut yönlendirme tablosu üzerinde mümkün olan en yüksek hızda karşılaştırma yaparak eldeki IP adres ön eki için en uygun yönlendirmeyi bulmak ve paketi ilgili arabirime göndermek.

2-Sürekli olarak gerçekleşen yönlendirme tablosu değişimlerini mümkün olan en kısa zamanda uygulamak ve bu sırada paket trafiğini aksatmamak.

Hız, her iki ihtiyaç konusunda da anahtar çözümdür. Çağdaş çalışmalar halen yaşayan uygulama olan IPv4 paket yönlendirme konusunu ele alarak performansı artırıcı çözümler önermekte ve bu önerileri IPv6' yı destekleyecek biçimde uyarlama yolunu göstermektedir[4,5,6].

IPv6, adres uzunluğu sebebiyle arama işinin bir miktar zorlaşmasına sebep olmakla beraber, bazı yeni tanımlar da getirerek yönlendirmeyi kolaylaştırmaktadır da. En devrimsel nitelikteki yeni özellik IANA (=Internet Assigned Numbers Authority) tarafından IPv6 adres ön eklerinin coğrafi bölge ve kıtalara göre ayrılmış olmasıdır. Böylelikle trafiğin iletimi bazen çok basit bir karara bağlı olabilir –ilgili bölgesel linke aktar-[7].

2.5 IP karşılaştırma (lookup) işlemi

Bir IP paketinin küresel Internet ağı üzerinde (veya herhangi bir IP ağında) alıcısına erişmesi için, doğru iletim hatlarına yönlendirilmesi gerekir. Yönlendirme işini yapan düğümlerde IP paketinin alıcı adresi kısmına ve düğümde bulunan IP yönlendirme tablosuna bakılarak, IP paketinin hangi arabirime yönlendirilmesi gerektiği konusunda bir karar verilir. Ağ üzerindeki iletimi sağlayan yolların değişebilir olması –mesela kopan bir bağlantı, devreden çıkan bir yönlendirici veya fiyatı daha ucuz olan yolu kullanma isteği v.b. gibi sebepler- düğümde saklanan IP yönlendirme tablosunun dinamik olması yani güncellenebilmesi gerekliliğini doğurur. Hem yönlendirmeye karar vermek için hem de güncellemeleri uygulayabilmek için sık sık yönlendirme tablosu üzerinde işlem yapmak gerekir.

Kullanılan metotlara genel bakış : IP karşılaştırma işlemi için yaygın biçimde kabul görmüş metotlar esas olarak “ikili arama” (binary search) prensibine dayanır. Ancak en saf haliyle ikili aramanın bazı sakıncaları vardır, öncelikle dolaşılacak her düğümde aranan değer ile bulunan değer karşılaştırılması gerekir, ikili ağacı oluşturma veya güncelleme sırasında ağacın dengesiz hale gelmemesi için algoritmanın uygulanması sırasında önlem alınması gerekir, bu ve benzeri nitelikler arama ağacı için daha optimal algoritmaların seçimini zorunlu kılar :

2.5.1 İkili arama ağacı (Binary Search Tree)

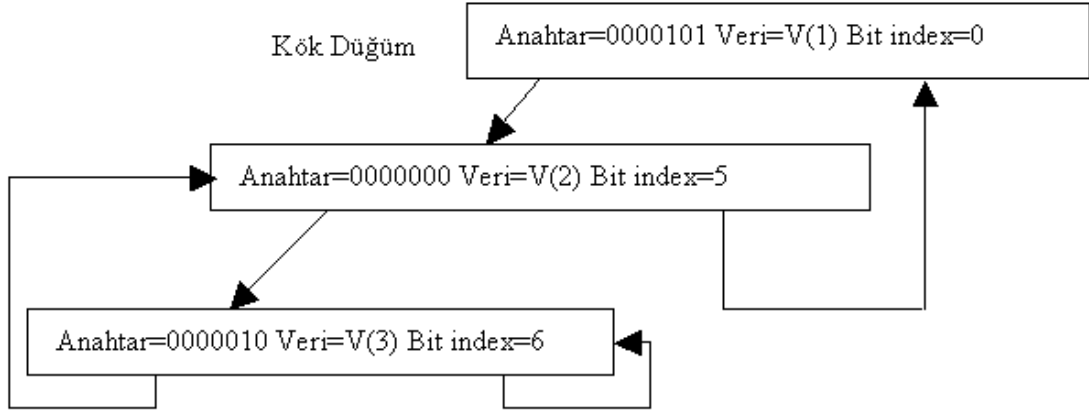
N adet düğüm içeren ikili arama ağacında W uzunlukta aramanın karmaşıklığı :

$O(W \times \log N)$ olacaktır.

Karmaşıklığı azaltmak ve uygulanabilirliği arttırmak için ikili aramayı temel alan ancak optimize edilmiş veri erişim metotları tercih edilir. Genel kanı ağacın kolay dolaşılabilir ve dengeli oluşması sonucunu verecek metotların işe yaradığıdır.

2.5.2 PATRICIA Trie (PATRICIA veri ağacı)

IPv4 yönlendirmesinde yaygın olarak kullanılan Patricia[10] (Practical Algorithm To Retrieve Information Coded in Alphanumeric, Morrison, 1968) veri ağacı(Şekil 2.4), karmaşıklığı düğümlerinde tuttuğu veri büyüklüğünden etkilenmeyen ve IPv4 için endüstriyel olarak kullanılan uygulamalarda da yer bulmuş başarılı bir veri yapısıdır[8]. FreeBSD bünyesindeki ve başka pek çok yazılımsal yönlendirici uygulaması PATRICIA Trie veya türdeşlerini kullanır. Uygulamada PATRICIA Trie’ nin 128 bit’ lik adresleri destekleyebilecek biçimde kurulması, mevcut yaygın ticari mikro işlemcilerin genelde 32 bit, bazı modellerde 64 bit hafıza erişimi sebebiyle performans açısından handikaplıdır. Buna karşın yakın gelecekte IP bazlı ticari multimedia yayını gibi sebeplerle alışlagelmiş yönlendirici veya PC donanımlarının yanı sıra GSM, PDA, taşıt aracı kontrol elektroniği gibi donanımların üzerinde de çalışabilecek, platform bağımsız uygulanabilecek, ölçeklenebilir IP karşılaştırma uygulamalarına ihtiyaç olacaktır.



Şekil 2.4 : PATRICIA veri ağacı (PATRICIA Trie) örneği

2.5 PATRICIA' nın gerçekleştirilmesi ve JAVA

Platform bağımsız, taşınabilir, ölçeklenebilir bir veri ağacı ile, yukarıda sayılan çeşitli cihazlar üzerinde IP karşılaştırma ve karar verme amacına hizmet etmek mümkündür. Platform bağımsızlığı, taşınabilirlik ve ölçeklenebilirlik hedefleri dikkate alındığında algoritmanın uygulaması için geliştirme ortamı olarak JAVA ön plana çıkmaktadır. JAVA, bir yazılım geliştirme ortamı olarak genellikle temel veri yapılarının programlanmasında kullanılmamaktadır. Her bir veri nesnesinin bir diğerine referans verebildiği pointer (işaretçi) tipi değişken kullanabilen programlama dilleri ile (C/C++/C#) temel veri yapıları ve algoritmaların programlanması daha alışlagelmiş olan yoldur.

Ancak JAVA dili de uygun biçimde kullanıldığında koyulan hedef için idealdir. JAVA' da her temel tipte değişken, gerçekte işaretçi tipli bir nesne ile temsil edilmekte ve nesnelerin birbirlerini işaretçi benzeri bir ilişki ile izlemesi sağlanabilmektedir. Uygulamanın arka planında JAVA yorumlayıcı tarafından çöp toplama (artık erişilmeyecek olan değişkenlerin işaret ettiği hafıza alanlarının otomatik temizliği) işinin yapılması, her bir işaretçi için önceden hafıza alanı sorgulama gerekmemesi, bunun yerine JAVA yorumlayıcı platformundan uygulamaya geriye dönen hata mesajının kontrol edilebilmesi ve buna benzer konforlu bazı özellikler de JAVA ortamını program geliştirme açısından ideal kılmaktadır. Herşeyden önemlisi JAVA çeşitli aygıtlarda –cep telefonu, PDA, araç bilgisayarı, kamera, tüketici elektroniği v.b.- gömülü (embedded) çalışabilirliğini kanıtlamıştır. Yakın gelecekte internete erişen cihazların çoğunluğunun bilgisayarlar dışında sözü edilen bu tip aygıtlar olabileceği düşünüldüğünde JAVA uygulama geliştirme platformu olarak gelecek vaad etmektedir.

3. PATRICIA VERİ YAPISI TABANLI YÖNLENDİRİCİ BENZETİMİ

Bu bölümde gerçekleştirilen uygulamada, yapılacak testlerin senaryolarına uygun olarak 64 bit anahtar saha, 64 bit veri sahası, düğüme ait bit index' i içeren, ayrıca sağ ve sol düğüm işaretçilerine sahip karakteristik bir PATRICIA ağacı yapısı [8] incelenmiş ayrıca sözü edilen PATRICIA ağacının veri tabanı olarak kullanıldığı temel yönlendirme kurallarına uyan bir yönlendirici uygulaması benzetimi kurgulanmıştır. Gerek PATRICIA ağacı gerekse yönlendirici simülasyonu 4. bölümde JAVA kullanılarak geliştirilmiş ve farklı senaryolar uyarınca farklı kişisel bilgisayarlarda IPv4 ve IPv6 adres yapısı benzeri verilerle performans testleri yapılmıştır.

3.1 PATRICIA algoritması

PATRICIA (Practical Algorithm to Retrieve Information Coded In Alpha-Numeric, Morrison 1968), yüksek miktardaki alfa-nümerik veriyi en az hafıza alanı sarfiyatı ile ve en düşük ağaç yüksekliği ile organize etmek amacıyla geliştirilmiş bir sözlük ağacı algoritmasıdır. Diğer sayısal arama ağaçlarına göre iki noktada önemli üstünlük sağlar :

1-Tek yönlü gereksiz dallanmaları önleyerek N sayıda değer için N sayıda düğüm içeren ağacı kurmayı olası kılar.

2-Tüm bir arama boyunca sadece 1 kez (karar aşamasında) aranan anahtar değerinin bulunulan düğüm ile kontrolü gerekir.

PATRICIA ağacının kuruluş algoritmasının işleyişini anlamak için aşağıdaki örnek vakayı incelemek yararlı olacaktır :

1.adım : Kök düğümün oluşturulması; ağaç bu kök düğümün sol alt ağacı olarak gelişecektir.

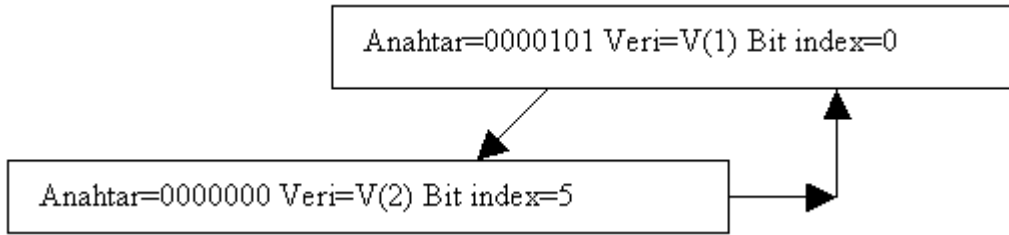
Her düğüm yapısında bir bit index değeri, bir anahtar saha ve sağ ve sol alt düğümlere birer işaretçi bulunur. Ağaç üzerinde dolaşılırken uğranan düğüme ilişkin bit index değeri bir sonraki adımda sağ veya sola dallanma yapılması gerektiğini anlatır. Tüm anahtar saha değerinin aranan değer ile karşılaştırılması gerekmez.

Anahtar=0000101 Veri=V(1) Bit index=0

Şekil 3.1 : Örnek PATRICIA kök düğümü

Örnek kök düğüm. Anahtar saha 5 (ikilik düzende 7 bit uzunlukta, 0000101) değerini taşımaktadır. “Veri” bu anahtar saha ile ilişkilendirilmiş bir bilgi topluluğudur. IP yönlendirme örneğinde anahtar saha IP ön eki, veri sahası ise yönlendirme yapılacak arabirim numarası,... gibi ilave bir bilgi olabilir. Kök düğümün bit index değeri 0’ dır. Ağaç geliştikçe aşağı doğru inen dallarda bit index değeri büyüyecektir.

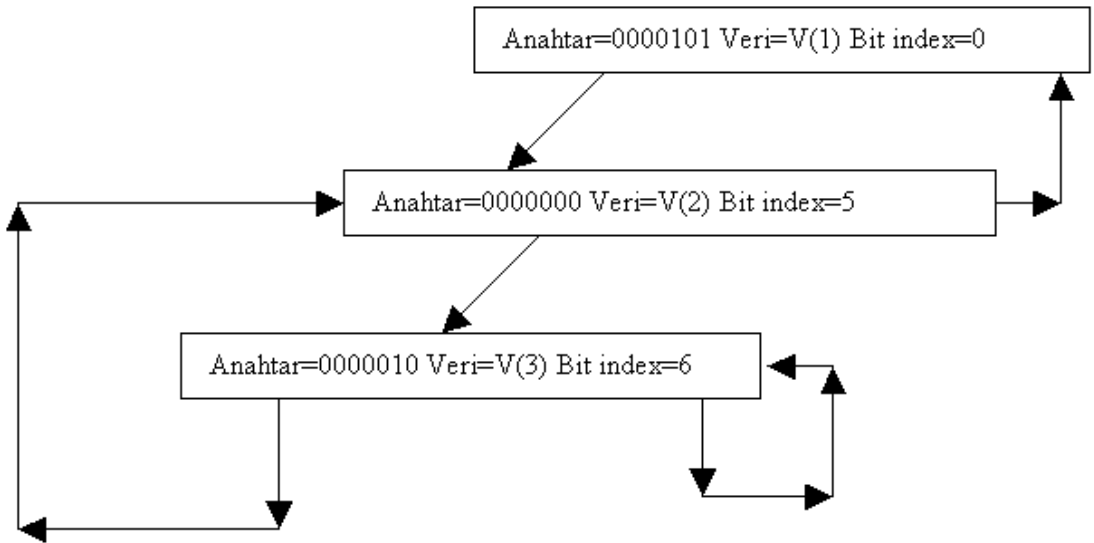
2. adım : Yeni düğümlerin ilavesi: Kök düğümden sonraki ilk düğüm, kökün sol alt ağaç işaretçisine bağlanacaktır. Bu düğüm (ve tabii ki ilave edilen her bir düğüm için) bit index değeri hesaplanarak düğüme yerleştirilecektir. 0 değerli anahtarı ilave edelim:



Şekil 3.2 : Örnek PATRICIA kök düğümüne yeni düğüm ilavesi

Bit index değeri 5 olarak hesaplanır. Çünkü önceki (ata) düğüme ait anahtar ile yeni anahtar değerinin farklılaştığı ilk bit (en soldan, “leftmost bit” değerinden itibaren) 5. bittir.

3. adım : Yeni düğümlerin ilavesi:



Şekil 3.3 : Örnek PATRICIA ağacına yeni düğüm ilavesi

Ağaç aşağı doğru dolaşılır. Kural kök düğümünden başlamaktır. Kök düğümün sol işaretçisi takip edilir (sağ işaretçi boştur). Ulaşılan düğümde bit index değeri okunur ve yerleştirilmeye çalışılan yeni düğümdeki anahtar sahanın ilgili sayılı biti test edilir. Bit 0 ise sola 1 ise sağa dallanılır ve ağaç gezilmeye devam edilir. Sağ veya sol işaretçilerden biri ağacın yukarı düzeyini (küçülen bir bit index değerini) işaret ettiğinde son ziyaret edilen düğüme geri dönülür, işaretçi bağlantıları yapılarak ağaca bir düğüm daha ilave edilmiş olur.

4. adım : Silme ve değişiklik : Ağaç gezilerek ilgili düğüm bulunur, anahtar saha aranan değer ile karşılaştırılır, tam eşitlik sağlanıyorsa düğüm silinir, kopan işaretçi bağlantıları yukarıya doğru onarılır. Değişiklik söz konusu ise (anahtar saha aynı kalıyor fakat veri değişiyorsa) ağaç gezilir, ilgili düğüme ulaşılır, sadece veri alanı güncellenir.

3.2 Çift Yığın (Dual Stack) IP yönlendirmede Patricia Trie tasarımı

İlk uygulamada kullanılacak olan PATRICIA Trie' ye ait veri yapısı aşağıdaki gibidir (ilgili JAVA programından alıntıyla) :

...

...

```
public class PatNode {  
    private int Bitindex;  
    private long Keyvalue;  
    private long Datavalue;  
    private PatNode LFnode;  
    private PatNode RGnode;
```

...

...

Veri yapısı incelenerek, uygulama için esas teşkil eden algoritmaya geçmek daha kolay olabilir. Veri yapısının bileşenlerinin üzerinden geçelim :

Bitindex : 3.1.' de anlatılan PATRICIA uygulaması esasları hatırlanırsa, veri ağacı "Bit index" olarak adlandırılan ve temel olarak eldeki düğümün içerdiği anahtar sahanın kaç numaralı bit' te bir önceki (ata) düğümün anahtar sahasından farklılaştığını işaret eden pozitif bir tamsayıdır. Olası en büyük değeri:

anahtar sahanın bit cinsinden uzunluğu – 1 olacaktır.

Keyvalue : Bit indexin hesaplanıp dallanmaya esas teşkil eden kararın verilmesine yol açan anahtar değerlerin depolandığı alandır. Pratik olarak bizim uygulamamızda IPv4 adres alanını veya IPv6 adres alanının 64 bitlik yarısını tutmaktadır. Olası en büyük değeri IPv6 adres alanı deposu olarak kullanıldığında 64 bittir.

Datavalue : Ağaç düğümlerinde depolanacak ve anahtar saha üzerinde yapılacak aramalar sonucu ilgili düğüme gelindiğinde okunacak veya güncellenecek verinin bulunduğu alandır. IP yönlendirici adres arama motoru olarak kullanılacak olan örnek ağacımızda bu alanda yönlendirici donanımının ilgili arabiriminin tanımı bulunmalıdır. Bu durumda olası en büyük değeri :

Yönlendiricinin toplam arabirim sayısı + n adet durum (yönlendirme yok veya benzeri özel kararlar için n adet durum ön görülebilir, toplam değer adedi < 64 bit)

olacaktır. Bu alanda kullanılan temel veri tipi yeterli seçilirse IP arama bazlı başka işlemler için de veri depolanması sağlanabilir. Örneğin NAT/NAPT uygulamalarında IP veya port numarası karşılıklarının depolanması, dinamik DNS hizmeti sağlanması gibi.

LFnode ve RGnode : Kurulan veri ağacı dinamik hafızada işaretçi kullanımı ile yer aldığı için her bir düğüm kendisinden sonraki sol ve sağ alt düğümleri işaret etmelidir. PATRICIA düğümünde sol ve sağ işaretçi olarak PATRICIA düğümü cinsinden veriler kullanılmıştır. Bu yapı hafızada işaretçilerle çalışmayı da mümkün kılmıştır.

PATRICIA veri ağacını oluşturan uygulamanın kaba kodu aşağıdaki gibidir , tam kaynak kod eklerde yer almaktadır:

```
Ana_Program_Başlangıcı ;
```

```
Veri_Yapılarını_Tanımla ;
```

```
public class PatNode {  
    private int Bitindex;  
    private long Keyvalue;  
    private long Datavalue;  
    private PatNode LFnode;  
    private PatNode RGnode;}...
```

```
Kök_Düğümü_Oluştur() ; // Tek düğümlük ağaç oluşur.
```

```
Sayaç_Başlat();
```

```
Ana_Döngü_Başlangıcı ;
```



```
Keyvalue= Sayaç_Değeri ;

Datavalue=Keyvalue_ile_ilişkili_bir_değer ;

Ağaca_Ekle(Keyvalue, Datavalue, Kök_Düğüm) ;

Sayacı_Arttır() ;

Ana_Döngü_Başlangıcına_Dön() ;

//Fonksiyonlar

Fonksiyon Ağaca_Ekle(Key, Data, Düğüm) ;

    //Bit_index değerini hesapla (Bölüm 3.1)

    //Sağ veya sol işaretçiyi kullanma kararı ver (Bölüm 3.1)

    //Kendini rekürsif olarak çağır

    Ağaca_Ekle(Key, Data, Düğüm) ;

Return ;
```

3.3 Tipik yönlendirici mimarisi ve PC benzetimi

IP ağında yer alan tipik bir yönlendirici, sahip olduğu arabirimler arasında veri trafiğini mümkün kılarken konumuna (=görev yaptığı noktanın ağ üzerindeki özel yerine) uygun olarak seçilmiş bir yönlendirme protokolü kullanır. Halen çeşitli yönlendirme protokolleri IPv4 ortamında kullanılmakta ve IPv6 ortamı için gerekli protokoller de genellikle bu protokollerden türetilmektedir. Yönlendiriciler üzerlerinde çalışan protokollerin ön gördüğü biçimde işlerini gerçekleştirmektedir[9].

Tipik olarak yönlendiriciler :

- Sahip oldukları ara birimlere erişen veri paketlerini inceleyerek iş döngülerine başlarlar :

1-Veri paketi iletilmek üzere yönlendiriciye erişmiş bir IP paketi olabilir

2-Veri paketi yönlendiricinin yararlanacağı bilgiler içeren bir TCP/UDP datagramı olabilir

- Yönlendirici veri paketini ne yapacağına karar verir :

1-Yönlendirme protokolünün gereği olan veri iletişimi söz konusu olabilir : Komşu yönlendiriciden veya yönlendiricilerden yönlendirme tablosu bilgileri gelebilir (bu bilgiler kullanılarak sahip olunan yönlendirme tablosu güncellenecektir), bu ve benzer durumlar yönlendirme tablosu üzerinde güncelleme yapılmasını gerektirir.

2-İletilmesi gereken paketler söz konusudur, bu durumda yönlendirme tablosu gözden geçirilerek paket ilgili arabirime yönlendirilir veya hiçbir ilgili arabirim bulunamıyorsa yok edilir.

- Yönlendirici, üzerinde çalışan protokolün gereği olarak bazı veriler yayınlabilir : Devreden çıkan veya devreye giren bir arabirim olabilir, komşu yönlendiricilerden devreden çıkanlar veya devreye girenler olabilir,..... bu değişiklikler tespit edildiğinde hem yönlendirme tablosunda güncellemeler yapılması hem de yapılan güncellemelerin yönlendirme protokolünün ön gördüğü biçimde komşu yönlendiricilere duyurulması gerekir.

Tüm bu sayılan işler yönlendirici devrede olduğu sürece sürekli tekrarlanır. Genellikle her eylem yönlendirme tablosu üzerinde bir kontrol veya güncelleme içerir. Yönlendirme tablosu üzerinde hızlı çalışma gereğinin temelini bu gerçek oluşturur.

3.4 RIPng PC benzetimi ve PATRICIA

RIPng yönlendirme protokolü esas olarak Bellman-Ford algoritması olarak bilinen ve maliyetli graflarda yol bulmayı mümkün kılan bir algoritmayı esas alır[9].

Yönlendirici yönlendirme protokolü ile ilgili bir paket aldığı anda paketin içeriğindeki bilgiyi algoritma uyarınca değerlendirir. Değerlendirme sonucunda yönlendirme tablosu üzerinde değişiklik yapılması ihtiyacı belirebilir. Yönlendiricinin çalışmasının bu kısmı tabloya çok hızlı erişim ve güncelleme gerektirir çünkü güncelleme ihtiyacının yanı sıra yönlendiricinin paket yönlendirme işi de devam etmektedir. Pek çok yönlendirici mimarisi güncelleme sırasında olası gecikmelerin etkisini en aza indirmek için yönlendirme tablosunun iki kopyasını bulundurmakta

ve güncellemeyi bir kopya üzerinde yapıp hemen akabinde güncellenmiş kopyayı kullanıma almaktadırlar[4].

Yönlendirici sadece yönlendireceği bir paket aldığı anda ise paketin alıcı adresini yönlendirme tablosu ile karşılaştırır. Bu karşılaştırma sonucu ya uygun bir yönlendirme bulunur ya da paket yok edilir. Bu işlem de olabilecek en yüksek hızda yapılmalıdır çünkü veri akışı süreklidir ve hat üzerinde gecikme yaratılmasına tolerans gösterilemez.

4. bölümde konu edilen uygulamalar, bir yönlendiricinin tablo başvuruları için kullanabileceği bir veri ağacını kurmayı, bu ağaç üzerinde hızlı arama-bulma yapmayı hedeflemektedir. Uygulama sonucu elde edilen ham değerler ile hesaplama yapıldığında çeşitli donanımlar üzerinde yüksek band genişliklerine hizmet edilebileceği anlaşılmıştır.

Uygulamanın bu derece yüksek hızlarda çalışabilmesi verinin özel bir formda organize edilmesi ile mümkün olabilmektedir. Bu tür yüksek performans gerektiren bir uygulamada alışılmış disk dosyaları kullanımı, veritabanı erişimi gibi yöntemler kullanılamaz. Bu yöntemler veriye ulaşırken zaman kaybına yol açar. 3. bölümde de anlatıldığı üzere ağaç veri yapıları yönlendirme tabloları üzerinde işlem yapılırken tercih edilen yapılardır. Veriyi en hızlı erişilebilecek biçimde organize etmeyi mümkün kılarlar. PATRICIA ağacı endüstriyel yönlendiriciler de dahil olmak üzere yönlendirici yazılımlarında geniş biçimde kullanılmaktadır. Gerçekleştirilen uygulama RIP protokolünün tüm davranışlarını benzeştiremez. Bu ayrı bir çalışma konusu olabilir. Ancak temel olarak veri ve protokol paketlerinin ayrımı, IP arama, bulma işlemleri benzeştirilmiştir.

3.5 RIPng algoritması kaba kodu (pseudo code)

RIPng protokolü esasları IETF RFC 2080 (G. Malkin, R. Minnear, Ocak 1997) dökümanında önerilmiştir.

RIPng' nin önce kaba kod üzerinden akış diyagramının kurgulanması anlaşılmasını kolaylaştırmak açısından yararlı olacaktır :

```
Ana_Program_ Başlangıcı ;  
Başlangıç_Ayarlarını_ Yap() ;  
Ana_Gövde_Döngüsü ;  
    Arabirimleri_izle ;  
        IP_paketi_bekle ;
```

```

Eğer
    Protokol_mesajı_geldi //UDP paketi,
                                //Alıcı adres IPv6 tüm yönlendiriciler
                                //duyuru adresi ya da yönlendiricinin
                                //IPv6 adresi, port 521
O halde
    Mesajı_kontrol_et(IP_paket_içeriği) ;
        Eğer
            Mesaj_geçerli ;
                O halde
                    Arama_ağacını_güncelle(Protokol_mesaj_içeriği) ;
                Ya da
                    Protokol_mesajı_ile_istenen_bilgileri_gönder(Protokol_mesaj_içeriği);
        Eğer Sonu
    Değilse
        Arama_ağacını_incele_yönlendirme_yap(IP_paket_alıcı_adresi) ;
    Eğer Sonu

    Ana_Gövde_Başına_Dön ;

Ana_Gövde_Sonu ;

//Fonksiyonların iç yapısı

Function Mesajı_Kontrol_Et() ;
    /* Mesaj, geçerli bir RIPng mesajı olmalıdır,
       Bu durumun kontrolü için ;
       Zaten port no kontrol edilmiştir,
       Mesajın komşu bir yönlendiriciden gelip gelmediği,
       İçeriğinin (yönlendirme tablosunun kısmen veya tamamen
           güncellenmesi veya kısmen veya tamamen
           gönderilmesi v.b. isteği) doğru formda olması,...
       gibi kontroller yapılmalıdır
    */

Function Arama_ağacını_güncelle(mesaj_içeriği) ;
    /* Protokol mesajının içeriğini yorumlayıp
       PATRICIA ağacına yeni düğüm eklemek, varolan
       düğümleri güncellemek veya düğüm silmek gibi
       işlemler desteklenmelidir
    */

Protokol_mesajı_ile_istenen_bilgileri_gönder(mesaj_içeriği) ;
    /* PATRICIA ağacı okunup ilgili içerik
       protokol mesajı paketleri haline getirilip
       tüm IPv6 router' ları çokluduyuru (multicast)
       adresine, ilgili port (521) üzerinden duyurulmalıdır
    */

Arama_ağacını_incele_yönlendirme_yap(alıcı_adresi) ;
    /* PATRICIA ağacı alıcı adresi anahtar saha yapılarak
       incelenmeli ve en yakın eşleşme bulunmalıdır.
       Bulunan eşleşme yönlendirme için kullanılacaktır
    */

```

4. PATRICIA VERİ YAPISI TABANLI YÖNLENDİRİCİ BENZETİMİ, UYGULAMA VE SONUÇLAR

4.1 Örnek Patricia Trie uygulaması

Bölüm 3.2' de açıklanan PATRICIA Trie veri yapısı kullanılarak gerçekleştirilmiş örnek veri ağacı uygulamasına ait JAVA ile yazılmış kaynak kod EK A' da bulunabilir. Bu çalışmada kullanılan test platformlarının MS Windows işletim sistemine sahip PC' ler olmasına karşın kod herhangi bir işletim sistemi veya donanım platformuna özel değildir ve JAVA 1.4 ve üzeri versiyonlarla çalıştırılabilir. Kod üzerinde, yapılan işlemlere ait detaylı açıklama satırları da yer almaktadır.

Uygulamanın oluşturulmasında [8] numaralı kaynakta yer verilen tipik ve temel PATRICIA uygulaması örneklerinden yararlanılmıştır. Ancak kullanılan veri yapısı çalışma konusuna uygun olarak uzun tamsayı (long integer) veri değerlerini destekleyecek ve yönlendirici uygulaması için veri sağlayacak biçimde tasarlanmıştır.

4.2 Test platformları ve test sonuçları

Kurulan senaryoların test edilmesi için kullanılan platformlar aşağıdaki gibidir :

A-Notebook PC : Intel PIII-1000 Mhz Mobile işlemci, 256 Mb RAM, Windows XP Pro., JAVA 1.5 Beta 2

B-Desktop PC : Intel PIII-550 Mhz işlemci, 256 Mb RAM, Windows XP Pro., JAVA 1.5 Beta 2

C-Notebook PC : AMD K6/II 400 Mhz işlemci, 96 Mb RAM, Windows 98, JAVA 1.5 Beta 2

Kurulan veri ağaçlarında 100.000 adet düğüm oluşturulmuştur. Bu rakam günümüz IPv4 yönlendiricilerinin yönlendirme tablosunu temsil etmek için yüksektir ancak IPv6' nın yaygın kullanımı durumunda tipik bir değer olacağı beklenebilir. Kurulan ağaçlarda 100.000 düğüm için hafıza ihtiyacı gerek PC donanımı gerekse hedef kitleyi oluşturan cep telefonu, PDA, kamera, araç kontrol elektroniği v.b. donanım

için ihmal edilebilecek kadar düşüktür, performans kriteri olarak dikkate alınmamıştır.

4.2.1 Senaryo 1 : 64 bit yapıda tek ağacın kullanımı

Senaryoda 64 bit veriyi destekleyen düğümlere sahip tek ağaç oluşturulmuş, sonra da ağaca baştan sona içerdiği tüm değerler için erişilmiş ve değişik platformlar için elde edilen değişik zaman performansları incelenmiştir.

IPv4 adresi için arama, IPv6 adresi için ilk 64 bitlik arama (Global Routing Prefix + Subnet ID' nin karşılaştırma işlemi, böylece kendi alt ağımızı hedeflemeyen bir IPv6 paketini ilgili global linke göndermek mümkündür) bu senaryoda kurulan veri ağacı ile desteklenebilir. Bu senaryoda elde edilen test sonuçları Tablo 3.1' de gösterilmiştir.

Tablo 4.1 Senaryo 1 için ölçülen zamanlar.

PLATFORM	KURMA ZAMANI (ms)	ARAMA ZAMANI (ms)
A	721	151
B	1883	390
C	3080	490

4.2.2 Senaryo 2 : 64 bit yapıda iki ağacın birbiri ardısıra kullanımı ile 128 bit' lik ağaç elde edilmesi

Sonuçları Tablo 4.1'de gösterilen senaryo 2'de 64 bit veriyi destekleyen iki adet ağaç oluşturulmuş ve ilk ağaçtan okunan veri ikinci ağaç için anahtar yapılarak ikinci ağaca erişilmiş, böylelikle toplam 128 bit' lik bir veri elde edilmesi sağlanmıştır. Bu senaryoda benzeştirilen, tüm IPv6 adresinin çözümlenmesidir. Toplam 200.000 adet düğüm içeren 64 bit' lik iki ağaç kurulmuş ve bu ağaçlar üzerinde arama yapılmıştır. Böylelikle 100.000 adet düğüm içeren 128 bit' lik bir ağaç benzeştirilmiştir.

Tablo 4.2 Senaryo 2 için ölçülen zamanlar.

PLATFORM	KURMA ZAMANI (ms)	ARAMA ZAMANI (ms)
A	1422	330
B	3716	941
C	6200	1050

4.3 PATRICIA Ağacı kurulumuna ilişkin test sonuçlarının analizi

2. senaryoda A platformu kullanılarak 100.000 adet düğüm içeren 128 bitlik (2 adet 100.000 düğüm içeren 64 bitlik) PATRICIA ağacı kurulmuştur. Ağacın toplam oluşturulma zamanı 1422 ms. olarak gerçekleşmiştir. Bu sonuç çağdaş çalışmalarda rastlanabilen sonuçlarla rekabet edebilecek durumdadır[4,5,6].

Yine 2. senaryoda A platformu ile 100.000 adet 128 bitlik düğüm içeren PATRICIA ağacının tüm düğümlerine erişen bir arama yapılmış ve 330 ms.' lik bir zaman harcanmıştır. Bu performans :

303.030 Adet Arama/Saniye (=100.000 / 330 x 1000)

anlamına gelmektedir.

Buradan ortalama 1,5 Kbyte uzunluktaki IPv6 paketleri ile çalışıldığı kabulü yapılarak olası bir yönlendirici için:

$303.030 \times 1,5 = 454.545$ Kbyte/Saniye

teorik band genişliğine ulaşılabilir. Tabii ki gerçek yönlendirici yazılımı sadece ağaç üzerinde karşılaştırma yapmayıp alternatif pek çok işlemle uğraşacağı için bu performansın pratikte gerçekleşmesi beklenemez. Ancak yine de sözü edilen band genişliği, performanslı bir PC bazlı 128 bit yönlendirici yazılımının gerçekleşmesinin mümkün olduğunu işaret etmektedir.

En düşük hızdaki platformdan alınan veriler üzerinde yorum yapmak daha gerçekçi bir yaklaşım olabilir. C platformu kullanılarak elde edilen arama performansı :

95.238 Adet Arama/Saniye (=100.000 / 1050 x 1000)

anlamına gelmektedir.

Bu platformun çalışma hızının da neredeyse günümüzdeki ve yakın gelecekteki GSM, PDA v.b. cihazlara yakın olduğu göz önüne alındığında uygulamanın gömülü çalışacak IP karşılaştırma mekanizmaları için çekirdek görevini üstlenebileceği anlaşılabilir. En yavaş platform olan C platformu :

$$95.238 \times 1,5 = 142.857 \text{ Kbyte/Saniye}$$

teorik band genişliği vaad etmektedir. Her ne kadar gerçek IP karşılaştırma (yönlendirme, filtreleme, multimedia...amaçlarıyla) uygulamasında yerine getirilecek olan ek görevler nedeniyle bu band genişliğini elde etmek mümkün görülme de düşük frekanslı uygulama platformları için elde edilen bu sonuç umut vericidir.

Uygulama programının çalıştırılması sonucunda elde edilen veriler gözden geçirildiğinde genel performansın test platformlarının hızına çok bağlı olduğu görülmektedir. Ancak çalışma platformunun hızı her ne olursa olsun uygulamanın karmaşıklığının her platformda aynı olduğu konusu gözden kaçırılmamalıdır.

Gerek ağaç oluşturulurken gerekse ağaç üzerinde arama yapılırken elde edilen en yüksek hız doğal olarak A platformuna aittir. Bu hızı internet' in gerçek yaşam koşulları ile karşılaştırıp fikir yürütmek gerekir. Uygulamanın optimize edilmesi karmaşıklığı aynı kalmasına rağmen çalışma hızını artırabilir.

4.4 Uygulama esasları ve RIPng benzetimi için gerekli kısıtlamalar

RIPng prensipleri ve PATRICIA ağacının arama motoru olarak beraber çalışmasını amaçlayan uygulama esas olarak yönlendirici davranışını benzeştirmeyi ve bu esaslar ile çalışan olası bir yönlendiricinin performans değerleri hakkında bir tahminde bulunabilmeyi amaçlamaktadır.

Yönlendirici benzetiminin gerçek bir IPv6 network' ü olmadan da IPv4 ağı imkanları ile kullanılabilmesi gerekir. Bunun için uygulama iki bölüme ayrılmıştır. Esas uygulama bağlı olduğu ağlardan gelen çağrılar dinlemekte ve cevaplamaktadır. Esas uygulamanın yanı sıra bir istemci uygulama kurgulanmıştır. İstemci uygulama belirli kurallar ve ön görülebilir bir rastgelelik ile IP paketleri üretmekte ve esas uygulamayı sorgulamaktadır.

Uygulamanın istemci-sunucu mimarisine benzeyen şekilde ikiye ayrılmış olmasına karşın gerçek bir istemci-sunucu kurgulamasından kaçınılmıştır. Çünkü gerçek yönlendirici çalışması istemci-sunucu tasarımında değildir. Yönlendirici ancak kendi adresini ve iletişim port' unu hedefleyen mesajlar karşısında tepki verir. Bu durumda

sunucu gibi davranır. Yönlendirici ihtiyaç duyduğu bilgileri ise komşu yönlendiricilerden talep eder, bu durumda da istemci gibi davranır. Toplamda yönlendirici davranışı alışlagelmiş istemci-sunucu formunda değildir.

Tam bir istemci-sunucu tasarımından kaçınmak için, esas programda JAVA 1.4 sürümünden beri var olan ve bir sunucunun eş zamanlı istek yapan çok sayıda istemciye cevap vermesini sağlayan kilitlemeyen bağlantı (non-blocking connection) ağ alt yapısı kullanılmamıştır. Gerçek sunucu davranışına daha yakın olan çalışma şekli, bir arabirimden gelen IP paketlerinin, geliş sıraları ile hizmet almaları, eğer yönlendirici uygulamasının performansı bu hizmeti vermeye yeterli değilse ağ üzerinde aksaklıklar çıkması, gecikmeler olması, paket kayıplarına bağlı olarak kaynak uçların isteklerini tekrarlamalarıdır.

Şu anda yaygın olan, temininde ve ayarlamasında zorluk çekilmeyen IPv4 ağ alt yapısı uygulama testleri için yeterlidir. Yönlendirici benzetiminin gerçekleştiği esas uygulama sık kullanılmayan özel bir portu dinler biçimde tasarlanmıştır. Bu port üzerinden gelen IPv4 paketlerinin veri kısmı (payload) IPv6 adres yapısına benzer biçimde 128 bit alıcı adres alanını taklit eden rastgele seçilmiş 4 adet integer sayı içerecek biçimde tasarlanmıştır. Eğer tüm IPv6 başlık yapısı benzeştirilirse, yapılan iş bir tür IPv4 içinde IPv6 kapsüllemesi (encapsulating) gibi düşünülebilir. Bu yöntem test ortamı kurmayı çok kolaylaştırır. Ancak IPv4 paket başlıklarının ayıklanması için bir miktar ilave güç harcanır. Bu da test kolaylığı avantajına karşılık ödenen bir bedeldir.

4.5 Test yöntemi

Yukarıda da bahsedildiği gibi uygulama esas ve yardımcı olmak üzere iki bölümlüdür. Esas bölüm arabirimine gelen IP paketlerini değerlendirmektedir. Yardımcı bölüm aynı PC üzerinde veya başka bir PC' de çalışabilir. Aynı PC üzerinde çalışacaksa "localhost" özel adresine verilerini göndermelidir. Başka bir PC' de çalışacaksa arada bir IP ağı olmalıdır. IPv4 öngörülmüştür ancak uygulama, yapısı gereği IPv6 ağına da çalışacaktır.

Yardımcı uygulama IP paketleri üretmekte ve esas uygulamanın ilgili bağlantı noktasına göndermektedir. Bu IP paketleri bir IP ağına rastlanacak olan ve çeşitli göndericilerden çeşitli alıcılara gidecek olan datagramlardır. Ağ üzerinde yönlendirilmek üzere oluşturulmuş IP paketlerinin toplam adedini daha yardımcı uygulama tarafından oluşturulurlarken tespit edebiliriz ve limitleyebiliriz.

Üretilen IP paketleri yönlendirici davranışını benzeştiren uygulamaya erişmekte ve uygulama da bu paketlerin içeriğinde göre bir iş yapmaktadır. Bu sırada harcanan zaman sebebiyle de uygulamanın hizmet edebileceği paket miktarı bir sınıra erişecektir. Bazı paketlerin hizmet alamayabileceği, paket tekrar ve kayıplarının olabileceği göz önünde bulundurulmalıdır. Bir şekilde hizmet alan paketleri esas uygulamanın alt rutinlerinde ilgili noktalarda sayarak bulabiliriz. Bu paketlerin adedine PH diyelim.

Bu durumda ;

Yönlendiri Arama Bulma Performansı = PH / Zaman

olacaktır.

Bu performans kriteri bize IP arama ve bulma işini yapan yönlendirici çekirdeğinin birim zamanda hizmet edebileceği IP paket adedini verecektir.

4.6 Uygulama kaynak kodu

Yönlendirici benzetimini gerçekleyen 2 ayrı uygulamanın kaynak kodları ekler bölümünde bulunabilir. Sürekli olarak paket üretilip bağlı olduğu ağa gönderen ilk uygulama (IPPacketSender.Java) yönlendirici üzerinde oluşacak iş yükünü benzeştirmektedir. Uygulama tipik IPv4 ve IPv6 adreslerini benzeştirmek için, 32 bit ve 128 bit (4x32 bit) uzunlukta veriler hazırlamakta ve bunları sunucu gibi davranan yönlendirici benzetimi uygulamasına göndermektedir. Ayrıca RIPng veya olası başka bir yönlendirme protokolünde yer alacak olan, yönlendirme yapılmayacak ancak yönlendiricinin bir iş yapmasına sebep olacak yönlendirme protokolü mesajlarını benzeştirmek amacı ile ilki 100. milisaniyede ve sonra her 5 saniyede 1 kez alıcı uygulama tarafından protokol mesajı olarak kabul edilen bir paket yollanmaktadır (zamanlar parametrik ve değiştirilebilir). Bu uygulamaya ait kaynak kod EK B' de bulunabilir.

Yönlendiricinin IP arama ve bulma davranışını benzeştiren 2. uygulamaya ait kaynak kod EK C' de bulunabilir. Bu uygulama bağlı olduğu ağ veya ağlardan gelen veri paketlerinin içerdiği bilgi uyarınca 2 adet IP karşılaştırma tablosu (PATRICIA veri ağacı) kurmakta veya tablolar üzerinde arama yapmakta, sabit zaman aralıkları ile gelen ve IP karşılaştırma tablosu ile ilgili bilgi içermeyen paketleri de ekrana basmaktadır (hedeflenen protokol mesajları ile gerçekleşmesi kaçınılmaz olan zaman ve performans kaybını benzeştirmektedir).

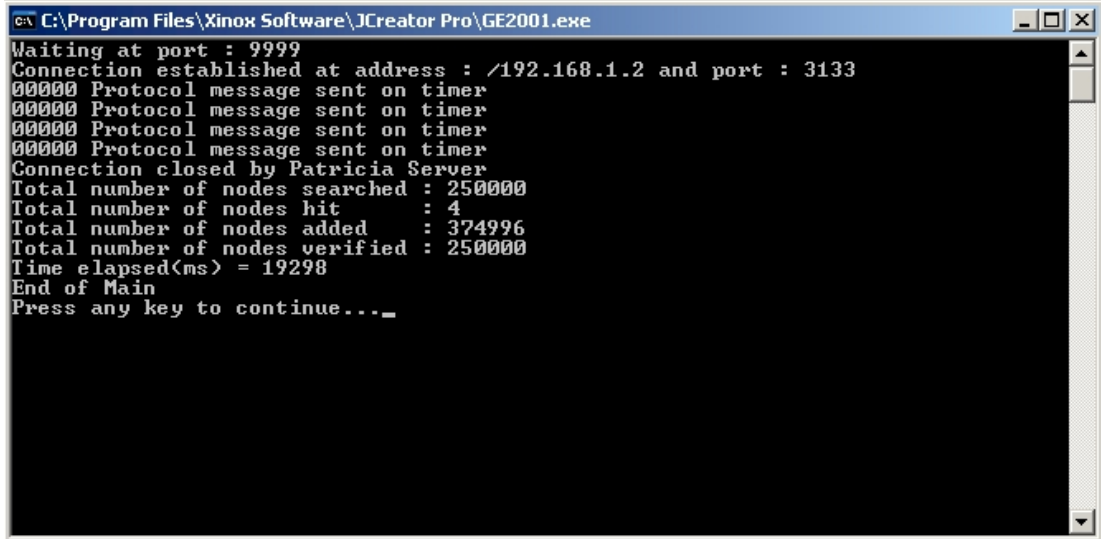
Her iki uygulama da OSI referans modeline göre uygulama katmanında çalışmaktadır. Bu nedenle yapısı bir istemci/sunucu uygulamasını andıran testler sırasında IP paket trafiğinin uygulama programları tarafından denetlenmesi gerekmez. Bu görev ağ katmanında TCP protokolü tarafından yerine getirilmektedir.

4.7 Test sonuçları ve Analiz

Daha önce PATRICIA ağacı kurulurken kullanılan donanımlardan (A) platformu uygulamada sunucu (yönlendirici IP arama ve bulma davranışının benzetimi) için seçilmiş ve üzerinde 2. uygulama çalıştırılmıştır. IP paketlerinin üretilmesi ve dinleyen sunucuya gönderilmesi işi için başka bir donanım platformu kullanılmıştır. Bu iş zaman açısından kritik olmadığı ve karmaşıklığı da düşük olduğu için daha basit donanımlarla da halledilebilir. Bu yeni platformun özellikleri aşağıdadır :

D-Desktop PC : Intel Celeron 1000 Mhz işlemci, 256 Mb RAM, Windows XP Pro., JAVA 1.5 Beta 2

Uygulamanın ekran görüntüsü aşağıdaki gibidir :



```
C:\Program Files\Xinox Software\JCreator Pro\GE2001.exe
Waiting at port : 9999
Connection established at address : /192.168.1.2 and port : 3133
00000 Protocol message sent on timer
00000 Protocol message sent on timer
00000 Protocol message sent on timer
00000 Protocol message sent on timer
Connection closed by Patricia Server
Total number of nodes searched : 250000
Total number of nodes hit : 4
Total number of nodes added : 374996
Total number of nodes verified : 250000
Time elapsed(ms) = 19298
End of Main
Press any key to continue..._
```

Şekil 4.1 : IP yönlendirici benzetimi uygulaması ekran görüntüsü

D platformundan A platformuna yollanan belli sayıdaki paketin, A platformu tarafından işlenmesi için geçen süre önemlidir. Uygulama, işlemlerin sonunda yaptığı işin sayısal büyüklüklerini rapor etmektedir. Yukarıdaki örnek ekran görüntüsü uyarınca uygulama 250.000 adet paket değerlendirmiştir (bu paketlerin yarısı 32 bit kalan yarısı da 128 bit rastgele üretilmiş integer veriler taşırlar). Uygulama gelen paketleri önce mevcut PATRICIA ağaçlarında (2 adet, 1 adet 32 bit adres karşılaştırması ve 128 bit' lik adreslerin ilk 64 biti için, 1 adet de 128 bitlik adreslerin

kalan 64 bitlik bölümü için) arar. Veri yoksa eklenir. Rastgele üretilmiş olan veriler içinde tekrar edenler olabilir. Bu tekrarlar ilk aramada bulunduğu için eklenmez. Teorik olarak 2 ağaca toplam eklenecek veri : 375.000 adet düğümdür. Ancak örnek çıktıda 4 adet arama sonucu bulunma (hit) gerçekleşmiştir. Arama ağacının bu çalışma koşullarında performansı şöyle hesaplanabilir :

Birim zamanda yapılabilen lookup = toplam lookup adedi / geçen zaman.

$(250.000 + 375.000 + 250.000)$ adet / 19.298 sn = 45.341 adet/sn. Veri ağacı işlemi performansı 45.341 adet/sn olarak hesaplanmaktadır. Yönlendirme protokolünün tüm gereklerinin yerine getirildiği bir simülasyonda bu değerin daha küçük olması beklenmelidir. Testin başka platformlarda da tekrarlanması yararlıdır. Tablo 4.1’de çeşitli veri konfigürasyonları için yapılmış testlerin sonuçları yer almaktadır, test (A) platformunda yapılmıştır (Hem istemci hem de sunucu olarak aynı platform kullanılmıştır) :

Tablo 4.3 : Çeşitli veri konfigürasyonları için (A) platformu test zamanları :

Paket Adedi	32 bit (adet)	128 bit (adet)	Toplam işlem	Zaman (ms)	Performans
150.000	75.000	75.000	525.000	13.960	37.607 adet/sn.
150.000	100.000	50.000	500.000	11.046	45.265 adet/sn.
150.000	50.000	100.000	550.000	15.713	35.002 adet/sn.
300.000	150.000	150.000	1.050.000	32.246	34.954 adet/sn.
300.000	200.000	100.000	1.000.000	31.095	32.159 adet/sn.
300.000	100.000	200.000	1.100.000	25.016	43.971 adet/sn.
450.000	225.000	225.000	1.575.000	35.511	44.352 adet/sn.
450.000	300.000	150.000	1.500.000	45.576	32.912 adet/sn.
450.000	150.000	300.000	1.650.000	40.498	40.742 adet/sn.

5. SONUÇLAR VE TARTIŞMA

Kurgulanan ve gerçekleştirilen uygulama sonucunda elde edilen test sonuçları incelendiğinde aşağıdaki çıkarımları yapmak mümkündür:

Öncelikle, lineer olarak artan miktarlarda ve değişen ağırlık oranlarına sahip artan miktarda veri ile yapılan testlerde toplam test zamanının da yaklaşık doğrusal olarak artış gösterdiği ve hesaplanan performansın artan veri miktarından olumsuz biçimde etkilenmediği görülmektedir:

Test uygulamasının yapısı gereği test verisi bileşimindeki 32 ve 128 bit ağırlık oranı 2 adet veri ağacına farklı oranlarda kullanımına yol açmaktadır. Test performansını esas etkileyen verinin bileşimidir. Çünkü bu bileşim arda arda kullanılan iki adet 64 bitlik veri ağacına erişim sayısını belirlemektedir.

Test sonuçları göstermektedir ki, en kötü performans bile aynı karmaşıklıkta tam bir yönlendirici yazılımının bir kenar yönlendirici görevini görebileceğini kanıtlamaktadır. En kötü performans için hesaplanan band genişliği (MTU=en büyük veri paketi boyu 1.500 byte olarak alınmıştır) :

$$B = 32.159 \text{ adet/sn} \times 1.500 \text{ byte} = 48,2 \text{ Mbyte/sn.}' \text{ dir.}$$

Bu performans ihtiyacın çok üzerinde görülmekle beraber tam bir yönlendirici yazılımında karmaşıklık değişirse de yapılan iş miktarı artacağı için paket yönlendirici performansı düşecektir.

Yapılan uygulamanın üzerinde durulması gereken bir diğer yönü de taşınabilir ve platform bağımsız olmasının hedeflenmiş oluşudur. Yakın gelecekte gezgin internet kullanımının artması ve IPv6 kaynaklı yeni tüketim mecralarının oluşması ile sadece geleneksel anlamda bilgisayarlarda değil, başta GSM aygıtları olmak üzere, ev elektroniği, araç kontrol elektroniği ve bunlar gibi elektronik aygıtlarda, donanım platformu ve işletim sisteminden bağımsız olarak internet' e erişim ihtiyacı şu ana göre defalarca kere artacaktır. Tüm bu cihazlarda IP trafiğinin kontrolü, güvenlik, konfor gibi nedenlerle bu çalışmadaki uygulamaya benzer temelleri kullanan yazılımlara ihtiyaç olacaktır. Bu çalışmada elde edilen sonuçlar özellikle küçük cihazlarda çalışabilecek gezgin IP temelli uygulamalar konusunda umut verici olmuştur.

KAYNAKLAR

- [1] **Cisco Systems IOS Learning Services, Understanding the Essentials Series**, 2002, ABCs of IP Version 6
- [2] **Chongrong Li, Xing Li, Jianping Wu, YueYou, Maoke Chen**, 2002, IPv6 Development in China, China Education and Research Network (CERNET) Center
- [3] **BBC NEWS**, Wednesday, 18 July, 2001, 11:12 GMT
<http://news.bbc.co.uk/1/hi/sci/tech/1442000.stm> , 01.03.2005, “Indian handheld to tackle digital divide”
- [3] **BBC NEWS**, 29 March, 2004, 09:36 GMT
<http://news.bbc.co.uk/2/hi/technology/3578309.stm> , 27.12.2004,
“Simputer for poor goes on sale”
- [4] **Chia-Tai Chan, Pi-Chung Wang, Shuo-Cheng Hu, Chung-Liang Lee, Rong-Chang Chen**, 2003, High-performance IP forwarding with efficient routing-table update, *Computer Communications*, **26**, 1681-1692.
- [5] **Priyank Warkhede, Subhash Suri, George Varghese**, 2004, Multiway range trees: scalable IP lookup with fast updates, *Computer Networks*, **44**, 289–303
- [6] **Butler Lampson, Venkatachary Srinivasan, George Varghese**, 1999, IP Lookups Using Multiway and Multicolumn Search, *IEEE/ACM Transactions on Networking*, **Vol. 7, No. 3**
- [7] **Internet Assigned Numbers Authority**, 2004, IPv6 Top Level Aggregation Identifier Assignments, <http://www.iana.org/assignments/ipv6-tla-assignments>
- [8] **Robert Sedgwick**, 2002, Algorithms in Java Third Edition Parts 1-4 Fundamentals Data Structures Sorting Searching, Chapter 15-3, Addison Wesley
- [9] **Silvia Hagen**, 2002, IPv6 Essentials, Chapter 8,10, O'REILLY

- [10] **Donald R. Morrison**, 1968, PATRICIA-Practical Algorithm To Retrieve Information Coded in Alphanumeric, Journal of the Association for Computing Machinery, Vol. 15, No. 4
- [11] **Malkin, G ve Minnear, R**, 1997, IETF Request For Comments : 2080

EK A – PATRICIA VERİ AĞACI JAVA UYGULAMASI

Kaynak kod, çalışmanın arka iç kapağında sunulan CD’ de bulunabilir
(PATRICIA\PatNode.Java).

EK B – IP PAKETLERİ ÜRETECİ JAVA UYGULAMASI

Kaynak kod, çalışmanın arka iç kapağında sunulan CD’ de bulunabilir
(\LOOKUP\IPPacketSender.Java)..

EK C – YÖNLENDİRİCİ İP ARAMA BULMA PERFORMANSI BENZETİMİ JAVA UYGULAMASI

Kaynak kod, çalışmanın arka iç kapağında sunulan CD’ de bulunabilir
(\LOOKUP\PatNode.Java).

ÖZGEÇMİŞ

Ahmet SÜSLÜ, 1969 İstanbul doğumludur. Vefa Lisesi (1986) ve Yıldız Teknik Üniversitesi Meslek Yüksek Okulu Bilgisayar Programcılığı Bölümü (1988) mezunudur. 1988-1991 yılları arasında Yıldız Teknik Ü. Elektrik Müh. Bölümü' ne devam etmiş fakat yoğunlaşan çalışma koşulları nedeni ile bu eğitimi tamamlayamadan bölümden ayrılmıştır. 1998 yılında Eskişehir Anadolu Ü. İktisat Bölümü' nü bitirmiş ve 2002 yılında Beykent Ü. Fen Bilimleri Enstitüsü' nde Lisansüstü eğitimine başlamıştır.

1987 yılından 1995 yılına kadar bilgi işlem sektöründe ve çeşitli büyük organizasyonların bilgi işlem merkezlerinde donanım teknisyeni, yardımcı ve uzman programcı, uzman analist ve proje yöneticisi olarak çalışmıştır. 1995 yılından bu yana kendi firmasında PC bazlı donanım ve yazılımların satışı, özel yazılım geliştirme, PC bazlı otomasyonların iş analistliği ve danışmanlığı konularında hizmet vermektedir.