Gazi Üniv. Müh. Mim. Fak. Der.
Cilt 29, No 2, 353-363, 2014

Journal of the Faculty of Engineering and Architecture of Gazi University
Vol 29, No 2, 353-363, 2014

IPv6 DESTEKLİ SOKET TABANLI SUNUCULAR İÇİN BİR YAZILIM ÖNERİSİ

İbrahim AKŞİT*, Şeref SAĞIROĞLU**
*Bilgi İşlem Daire Başkanlığı, Gazi Üniversitesi, Teknikokullar, 06500, Ankara
**Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, Gazi Üniversitesi, Maltepe 06570, Ankara
iaksit@gazi.edu.tr, **ss@gazi.edu.tr**

(Geliş/Received: 10.07.2012; Kabul/Accepted: 13.03.2014)

ÖZET

Bu çalışmada, mevcut sunucu tabanlı POP3 uygulamalarına IPv6 desteği verilmesi için gerekli olan altyapı yazılım desteğinin neler olması ve konu ile ilgili yapılması zorunlu değişikliklerin neler olması gerektiği ve bunun için örnek bir uygulamanın nasıl geliştirilebileceği adım adım ilk kez anlatılmıştır. İlk kez önerilen çalışmada sonuç olarak; gelen e-postaların IPv6 üzerinden iletilmesini sağlamak amacıyla IPv6'nın nasıl yapılandırılması gerektiği bu makalede anlatılmış, geliştirilen uygulama Microsoft Office Outlook programı kullanılarak test edilmiş ve IPv6 destekli soket tabanlı yazılımların geliştirilmesi noktasında ilk kez çözüm önerileri sunulmuştur.

Anahtar kelimeler: IPv6, IPv6 Yapılandırma, IPv6 Destekli Yazılım, Soket Tabanlı Yazılım

A SOFTWARE SUGGESTION FOR IPv6 SUPPORTED SOCKET BASED SERVERS ABSTRACT

ABSTRACT

In this study, a sample application has been explained step by step the first time to provide IPv6 support necessary infrastructure software support, what changes should be done for an existing server based on POP3 application. IPv6 configuration has been carried out in order to ensure received e-mails transmission over IPv6 with the developed software. The software has been also tested using Microsoft Office Outlook and the solution has been presented to the literature for the first time in the point of development of IPv6 supported socket-based software.

Keywords: IPv6, IPv6 Configuration, IPv6 Supported Software, Socket Based Software

1. GİRİŞ (INTRODUCTION)

Günümüzde yaygın olarak kullandığımız IPv4 (Internet Protocol version 4) 1982 yılında uluslararası bir organizasyon olan IETF (Internet Engineering Task Force) tarafından kabul edilmiştir. Dünya nüfusuna bağlı olarak internetin, teknolojik alt yapıların hızla büyümesi neticesinde kullanılan IPv4 temelli ağların yetersiz kalması, IPv4 adreslerinin tükeneceği ve neredeyse elde IP'nin kalmaması büyük bir sorun teşkil etmektedir [1]. APNIC (Asia Pacific Network Information Centre) ve IANA (Internet Assigned Numbers Authority) verilerine göre en son kalan beş (5) IPv4 adres bloğu ARIN (American Registry For Internet Numbers), AFRINIC (African Network Information Centre), APNIC (Asia Pacific Network Information Centre), LACNIC (Latin America and Caribbean Network Information Centre), ve RIPE NCC (Réseaux IP Européens Network Coordination Centre) arasında eşit olarak dağıtılmış olup artık dağıtılacak bir IPv4 adresi bulunmamaktadır [1-13].

IPv6 için yapılan önerilerden sonra bu protokole geçişler için çalışmalar başlamıştır. Dünya ülkeleri başta Çin, Japonya olmak üzere hızla IPv6'ya geçiş süreçleri ve çalışmaları başlatmıştır. IPv6 protokolünü ilk kez pratiğe geçiren ülke 1998 yılında Çin olmuştur. Japonya Eylül 2000'de, Güney Kore Şubat 2001'de, Avrupa Komisyonu/Birliği Nisan 2001'de

IPv6'yı benimsediklerini duyurmuşlardır [4]. Ülkemizde 2001 yılında başlayan çalışmalar, 15 Şubat 2009 tarihinde TÜBİTAK tarafından desteklenen Ocak 2011 tarihlerinde Ankara'da Ulusal IPv6 Konferansı gerçekleşmiştir. Konferansta IPv6 her yönüyle ele alınmış ve IPv6 hakkında yapılan çalışmalar sunulmuştur. 15 Şubat 2012 tarihinde düzenlenen II. Ulusal IPv6 Konferansında, konferanstan elde edilen, yeni nesil IP'ye geçiş süreci mevcut altyapının IPv6 için eksiklikleri, uzman azlığı, mevcut sistemlerin kısmen veya tamamen değişecek olmasının getirdiği ekonomik yük ve yeni teknoloji geliştirme gibi engelleyici bazı faktörlerden dolayı kolay olmayacağı, internetin ve kablosuz bağlantıların (cihazların) geleceğinin IPv6 protokolüne dayandığı ve internete bağlı cihazların donanım ve yazılımlarının buna göre şekilleneceği ve bu konularda daha çok akademik çalışmaların yapılması gerektiği görülmektedir.

Bu çalışmada, yeni nesil IP olarak bilinen IPv6 hakkında gerekli temel bilgiler takip eden başlıklarda kısaca sunulmuş ve IPv6 kullanacak cihazlar, bilgisayar ve bilgisayar ağlarının ve yazılımların IPv6 desteklemesi için gerekli süreçler, aşamalar ve yapılandırma ile ilgili bilgiler verilmiştir. Var olan POP3 programı için IPv6 desteğini sağlayan öneriler sunulmuş ve bunların yazılımlarının nasıl geliştirilmesi gerektiğine dair örnekler tanıtılmıştır. IPv6 protokolünü kullanan sistemlerin dünyadaki diğer IPv6 ya da IPv4 protokolünü kullanan sistemlerle etkileşiminin, bilgi iletişiminin sağlanabilmesi için yapılması gereken yapılandırmaların ve yönetim tekniklerinin ne olması ve bunların nasıl yapılması gerektiği araştırılmış ve sonuçta bu çalışma kapsamında bir sunucu yazılımı geliştirilerek soket tabanlı IPv6 desteği verilebilmesi için çözüm önerileri sunulmuştur.

2. IPv6 DESTEKLİ UYGULAMA GELİŞTİRİLMESİ İÇİN ÖNERİLER (SUGGESTIONS FOR DEVELOPING IPv6 SUPPORTED APPLICATIONS)

Mevcut uygulamaların çoğunda IPv6 desteği bulunmamaktadır. Uygulamaların IPv6 paketlerinden gelen veriyi işleyebilmeleri için mutlaka IPv6 desteğine sahip olmaları gerekir.

Devam eden satırlarda mevcut uygulamalara IPv6 desteğinin verilmesi, mevcut uygulama kodlarının düzenlenmesi, IPv6 destekli uygulama geliştirilmesi için yöntemler ve öneriler uygun başlıklar altında özetlenmiştir. Daha detaylı bilgi için [1] nolu kaynağa başvurulmalıdır.

2.1. Mevcut Uygulamalara IPv6 Desteği Verilmesi (Providing IPv6 Support For Existing Applications)

Günümüzdeki uygulamaların çoğu henüz IPv6'yı destekler durumda değildir. Bu uygulamalara IPv6 "Ulusal IPv6 Protokol Altyapısı Tasarımı ve Geçişi Projesi" kapsamında yapılan bu ulusal geçiş projesi ile zirveye çıkmıştır [4]. Bu proje kapsamında 12-13 desteği sağlayabilmek için mevcut yazılımlar gözden geçirilmeli veya yeniden hem IPv4 hem IPv6 desteği sağlayan yazılım çözümleri geliştirilmelidir. Böylece geliştirilecek uygulamalar hem IPv4 hem de IPv6 ile çalışabilecek duruma gelirler. IPv4 adresleri uygulamalar tarafından tamsayı (integer) olarak saklanmaktadır. IPv4 adreslerinin 32 bit olmasından dolayı uygulamalarda herhangi bir sıkıntı oluşmamıştır. Fakat yeni nesil internet protokolü - IPv6, 128 bit uzunluğundadır. Bu yüzden mevcut IPv4 adreslerinde kullanılan tamsayı alanı bu protokol için kullanılamaz. IP adreslerinin tutulacağı verinin türü ve alanı IPv4/IPv6 ikilisi için değiştirilmesi, sistem dosyalarından olan **<sys/socket.h>** kütüphanesinde değişiklikler yapılmalıdır [1,6-13].

Uygulama olarak ikili yığın mekanizmasını desteklemek için öncelikle IPv4 ve IPv6 adresleri ele alınmalıdır. Uygulamalarda hem IPv4 hem de IPv6 adreslerinin kullanılabilmesi için *sockaddrs* yapılarından, *sockaddr_in* veya *sockaddr_in6* gibi yapılardan yararlanılmıştır. Böylece, adres sınıfı bilgilerini içeren veriler tanımlı adres türlerine göre veriyi taşıyabilirler.

IPv4 soketlerini belirlemek için soket tabanlı uygulamaların programlama ara yüzleri **AF_INET** sabitini kullanarak tanımlarlar. IPv6 soketlerini belirlemek için ise **AF_INET6** sabiti kullanılır. C programlama dili yapısında yer alan IPv4 akranlarını belirlemek için *sockaddr_in* yapısı kullanılır. IPv6 yapısını belirlemek için ise *sockaddr_in6* yapısı kullanılır [1,6-13].

IPv4'de kullanımı: **sDegiskenIPv4 = socket (AF_INET, SOCK_STREAM, IPPROTO_TCP)** şeklindedir. IPv6'da kullanım şekli ise **sDegiskenIPv6 = socket (AF_INET6, SOCK_STREAM, IPPROTO_TCP)**şeklindedir. *sockaddr_in* ve *sockaddr_in6* yapılarının nasıl tanımlandığı bilgisi detaylı olarak [1] nolu kaynakta verilmiştir.

Soket tabanlı API (Application Programming Interface) ile IPv6 destekli uygulamalar geliştirilebilir. Bunun için IP adres sınıfını **getaddrinfo** ve **getnameinfo** kullanarak adres türünden bağımsız bir hale getirilebilir. Uygulamalarda adresleri temsil etmek için **sockaddr** yapısı kullanılacağı belirtilmiştir. [1] nolu kaynaktan bu protokollere ait bağımsız yeni uygulama program ara yüzlerine (API) erişilebilir. Geleneksel olarak, sadece IPv4 uyumlu programlar IPv4 adreslerini *in_addr* yapısını kullanarak çalıştırırlar. **in_addr** veya **in6_addr** yapısını kullanan soket API' lerinin kullanılmaması gerekir. Aksi takdirde geliştirilecek

uygulamalar hangi adres türünden veri alacağını anlayamaz ve çalışamaz duruma gelirler [5].
IPv6 destekli mevcut uygulama örnekleri gelecekteki çalışmalara katkıda bulunacağından, detaylı bilgiler [1] nolu kaynakta verilmiştir.

2.2. Mevcut Uygulama Kodlarının Düzenlenmesi (Editing Existing Application Codes)

IPv4 tabanlı programlara IPv6 desteği verecek hale getirmek için öncelikle *.c ve *.h uzantılı C programlama dilinin kullandığı dosyalarda IPv4 tarafından kullanılan soket tabanlı uygulama fonksiyonlarının incelenmesi faydalı olacaktır. Bu makale çalışmasının alt yapısını oluşturan bu dosya uygulama örneklerine detaylı bilgi için tez çalışmasına başvurulmalıdır [1].

Unix işletim sistemi ve türevlerinde (Linux gibi) sunucu uygulamalarının çalışabilmesi için temelde iki yol izlenmektedir. Bunlar;

- İnternet süper sunucu olarak da bilinen ***inetd*** modudur. *Inetd* ağdan gelen talepler doğrultusunda en az kaynak kullanımı ile birçok ağ programını yükler. *Inetd(8)* olarak da gösterilebilir.
- Tek başına (stand-alone) çalışan program modudur.

Sunucu tabanlı uygulamaların hem IPv4 hem de IPv6 kullanıcılarının kullanabileceği (yararlanabileceği) bir hizmet sağlamaları için **AF_INET ve AF_INET6** olarak isimlendirilen dinleme soketinin açılması gerekir. Bu soketlerin açılması bir kaç yol ile yapılabilir. Bunlar;

(1) Uygulamayı IPv6 destekler hale getirmek: *Inetd(8)*, süper sunucusunu, hem AF_INET hem de AF_INET6 bağlantılarına cevap verebilecek şekilde ayarlanması gerekir. Genel olarak Linux ve Unix işletim sistemlerinde bulunan **/etc/inetd.conf** dosyasından bu yapılandırma yapılabilir.
(2) Çoklu dinleme soketlerini algılayan bir uygulama çalıştırmak: Bu **select(2)** veya **poll(2)** fonksiyonları/metodları kullanılarak yapılabilir.
(3) Uygulamanın iki durumunu (AF_INET ve AF_INET6) da çalıştırmak: Bu durumlardan birisi IPv4 için AF_INET diğeri ise IPv6 durumu için AF_INET6'dır.

Bahsi geçen durumlardan (1) ve (2) nolu durumlar RFC belgelerinde önerilmektedir. (3) nolu durumda uygulamada soketler arasında geçiş yapabilmek için komut satırından komutlar verilmelidir [1].

2.3. IPv6 Destekli Sunucu Uygulamasının Geliştirilmesi (Developing IPv6 Supported Server Application)

Bu çalışmada, [1] nolu kaynakta yapılan çalışmanın bazı kısımları baz alınarak gerekli IPv6 destekli bir sunucu uygulamasının nasıl geliştirilebileceği anlatılacaktır. Geliştirilen yazılım *inetd (süper sunucu)* ve *tek başına (stand-alone)* gelen çağrılara cevap verebilmektedir.

Ülkemizde bu yapıda yazılım geliştirme için yeterli bilgi birikimi olmadığı için uygulama geliştirme ve test ortamları aşağıda detaylı olarak alt başlıklarda açıklanmıştır.

2.3.1. Uygulama geliştirme ortamı (The application development environment)

[1] nolu kaynakta detaylı olarak anlatılan uygulama Dev C++ editörü kullanılarak geliştirilmiştir. Uygulama Ubuntu 10.10 maverick sunucusu üzerinden çalıştırılmıştır.

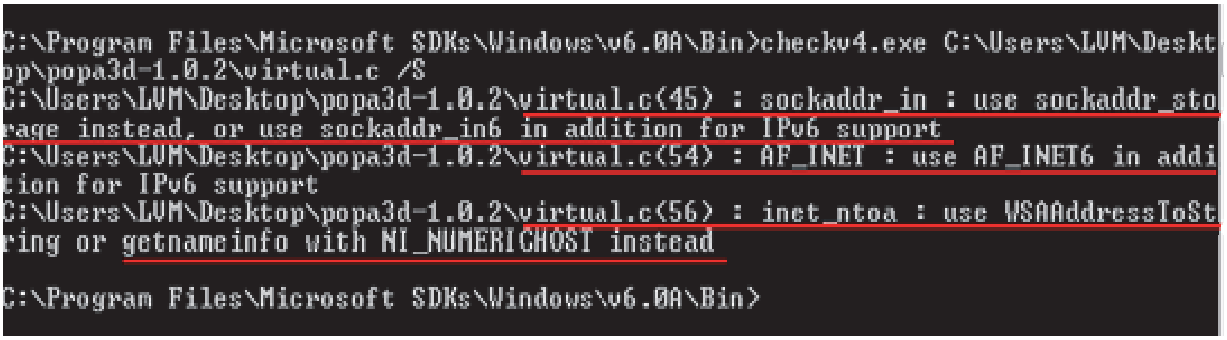
IPv6 desteği verilmesi gereken IPv4 soket dosyalarının hangileri olduğunu ve hangi satırlarda geçtiğini bulmak için [1] nolu kaynakta belirtilen komutlar kullanılmalıdır.

Kaynakta kullanılan komutlar neticesinde IPv6 için değişiklik yapılması gereken dosyalar *virtual.c*, *standalone.c*, *params.h* ve *startup.c* dosyalarıdır. Bu dosyaların içeriği kontrol edilip gerekli düzeltmeler belirtilen dosya başlıkları altında yapılmıştır. Bunlar aşağıda alt başlıklarda açıklanmıştır.

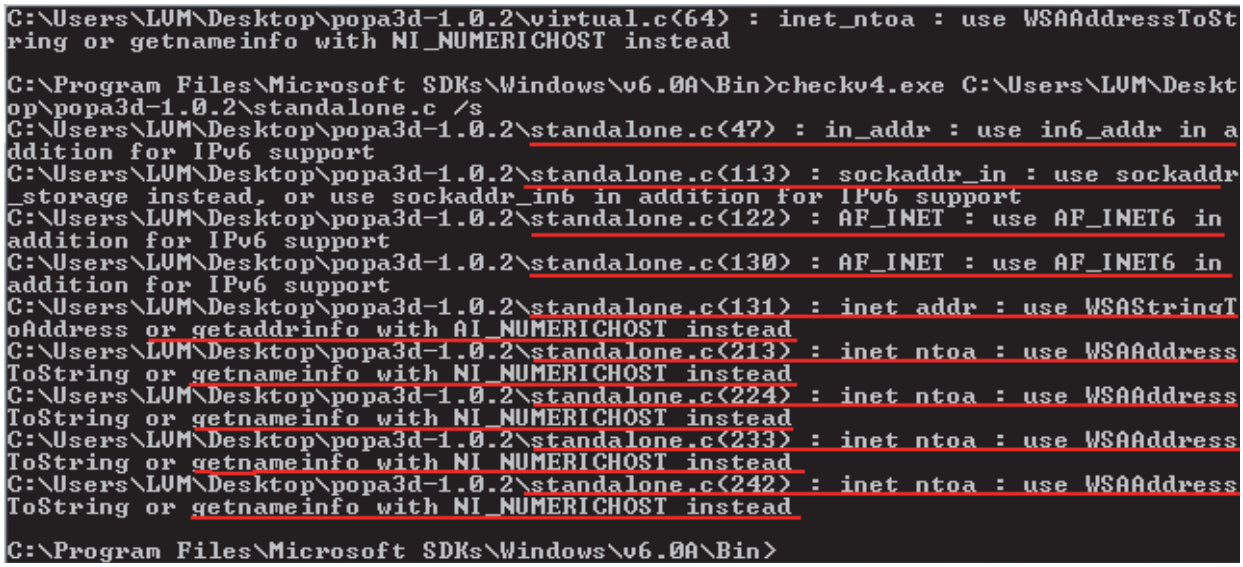
"virtual.c" dosyasının düzeltilmesi için yapılacak işlemler ve açıklamaları

Bu dosya kullanıcı tarafından POP3 sunucunun IP adresine bağlı olarak birbirine bağlı dosyalarını çoklu dosyalara ayırmak için sanal ana dizin oluşturma işlevini sağlar. Dosyanın içerisinde yer alan IPv4 bağımlı olan kod **lookup()** fonksiyonu içerisinde yer alır. Bu kodların adresten bağımsız olarak çalışması için düzeltilmesi gereken alanlar vardır [1]. IPv6 desteği verilmesi gereken dosyalar ayrıca Microsoft tarafından geliştirilen küçük bir uygulama olan checkv4.exe'yi çalıştırarak daha net bir şekilde anlaşılabilir [1]. Şekil-1'de checkv4.exe programı *virtual.c* dosyasında yapılması gereken değişiklikler listelenmiştir. Bu değişiklikler aşağıda sıralanmıştır;

(1) sockaddr_in yerine ***sockaddr_storage*** veya sadece IPv6 desteği için ***sockaddr_in6*** kullanılmalıdır.
(2) AF_INET yerine **AF_INET6** kullanılmalıdır.
(3) inet_ntoa yerine ***getnameinfo()*** kullanılmalıdır [1].



Şekil 1. virtual.c dosyasının checkv4.exe ile kontrolü [1] (Controlling virtual.c file with checkv4.exe)



Şekil 2. standalone.c dosyasının checkv4.exe ile kontrolü için açılan ekran görüntüsü [1] (Screenshot for controlling standalone.c file with using checkv4.exe)

Bu değişikliklerden sonra *virtual.c* dosyamız IPv6 desteğine sahip bir dosya olacaktır. Bu dosya üzerinde yapılan değişiklikler ve açıklamaları [1] nolu kaynakta detaylı olarak verilmiştir.

"standalone.c" dosyasının düzeltilmesi için yapılacak işlemler ve açıklamaları

Popa3d uygulamasının süper sunucu (Inetd) olarak mı yoksa tek başına (standalone) çalışıp çalışmayacağını *params.h* adlı başlık dosyasında yer alan POP_STANDALONE değeri belirler. POP_STANDALONE değeri 0 (sıfır) ise süper sunucu tarafından tetikleneceğini belirtir. POP_STANDALONE değeri 1 (bir) ise ya süper sunucu tarafından ya da tek başına tetikleneceğini belirtir. POP_STANDALONE değeri 1 olarak değiştirilmiştir.

Bu dosya basit olarak tek başına çalışma olarak tetiklendiği durumda Popa3d uygulamasının hizmet vermesini sağlamaktadır. Checkv4.exe programı tarafından kontrol edilen ve Şekil-2'de *standalone.c* dosyasında değiştirilmesi gereken kodların ekran görüntüsü verilmiştir.*standalone.c* dosyasında yapılacak değişiklikler ve açıklamaları [1] nolu kaynakta detaylı olarak verilmiştir. Burada sadece bir kısmı verilmiştir;

(1) Öncellikle **standalone.c** dosyamıza ağ veritabanı işlemlerinin tanımlamalarından sorumlu **netdb.h** başlık dosyası **(#include <netdb.h>),** giriş çıkış arabirimlerinden sorumlu **ioctl.h** sistem başlık dosyası **(#include <sys/ioctl.h>)** ve ağ kartından sorumlu **if.h** ağ başlık dosyası **(#include <net/if.h>)** dahil edilir.

(2) IP adresinin alınacağı aygıt tanımlanır. **(#define DEVICE "eth0")**

(3) Kaynak IPv4 adresinin tanımlandığı **struct in_addr addr** yapısı kaynak IPv6 adresini de alabilmesi için aşağıdaki gibi kod satırı eklenmesi gerekmektedir.

```
#if HAVE_INET6
    struct in6_addr addr6; /* Kaynak IPv6 adresi */
#endif
```

(4) Belirtilen adres IPv4 veya IPv6 adresi olması durumunda **int main (void)** fonksiyonunda, aşağıda gösterilen şekilde gerekli tanımlamalar yapılır.

```
#if !HAVE_INET6
    struct sockaddr_in addr;
    int addrlen;
```

```
#else
    struct sockaddr_in6 addr6;
    struct ifreq ifr;
    char temp[24];
    int addrlen6;
#endif
```

(5) Bu madde ve bundan sonraki maddelere [1] nolu kaynaktan erişilebilir.

Bu değişikliklerden sonra popa3d uygulaması artık komut satırından **-4** ve **-6** parametrelerinden herhangi birini girerek hangi IP türüne göre işlem yapılacağı belirtilir [1,6-13].

"params.h" dosyasının düzeltilmesi için yapılacak işlemler ve açıklamaları

Popa3d uygulamasının süper sunucu (Inetd) olarak mı yoksa tek başına (standalone) çalışıp çalışmayacağını *params.h* adlı başlık dosyasında yer alan POP_STANDALONE değeri belirler. POP_STANDALONE değeri 0 (sıfır) ise süper sunucu tarafından tetikleneceği belirtilir. POP_STANDALONE değeri 1 (bir) ise ya süper sunucu tarafından ya da tek başına tetikleneceği belirtilir.

Bu dosya basit olarak tek başına çalışma olarak tetiklendiği durumda popa3d uygulamasının hizmet vermesini sağlamaktır. *params.h* dosyasında yapılan değişiklikler ve açıklamaları şu şekilde sıralanabilir ([1] nolu kaynakta detaylı olarak verilmiştir);

(1) IPv6 paketlerini de uygulamada kullanabilmek için programa aşağıdaki kod satırı eklenmelidir.

```
//IBRAHIM AKSIT -> IPv6 Desteği sağlamak için tanımlandı
#define HAVE_INET6      1
```

(2) Sadece IPv4 adresleri için bütün arayüzleri dinleyecek olan 0.0.0.0 adres tanımlamasına **(#define DAEMON_ADDR "0.0.0.0" /* INADDR_ANY */)** ek olarak IPv6 adreslerini dinleyecek olan "**::**" adresi için aşağıdaki tanımlama yapılmıştır.

```
//IBRAHIM AKSIT -> IPv6 dinlenecek adres için tanımlandı
#define DAEMON_ADDR6  "[::]"
/* IPv6 için dinlenecek adres - bütün arayüzleri dinler*/
```

Yapılan değişikliklerden sonra popa3d uygulaması artık IPv4 adreslerinin yanında IPv6 adreslerini de dinleyecektir.

"startup.c" dosyasının düzeltilmesi için yapılacak işlemler ve açıklamaları

Popa3d uygulamasının başlangıç dosyasıdır. Komut satırından gelen parametrelerle işlem yapar. *startup.c* dosyasında yapılan değişiklikler ve açıklamaları şu şekilde sıralanabilir;

(1) Temel anlamda soket başlıkları sistemden çağrılarak uygulamaya dahil edilir.

```
/* socket başlık dosyasını uygulamaya dahil ediyoruz */
#include <sys/socket.h>
```

(2) IPv6 adres sınıfını da dinleyebilmek için aşağıdaki kod satırı programa eklenir.

```
/* Adres sınıfına IPv6 ekledim */
int af = AF_INET6;
```

(3) Geliştirilen uygulamanın kullanım fonksiyonuna komut satırından IPv4 ve IPv6 adreslerini parametre olarak belirtilebilmesi için gerekli kodlar eklenir.

```
/* Komut satırı IPv6 desteği için ekledim.*/
fprintf(stderr, "Usage: %s [-D] [-V] [-4] [-6] \n", progname);
```

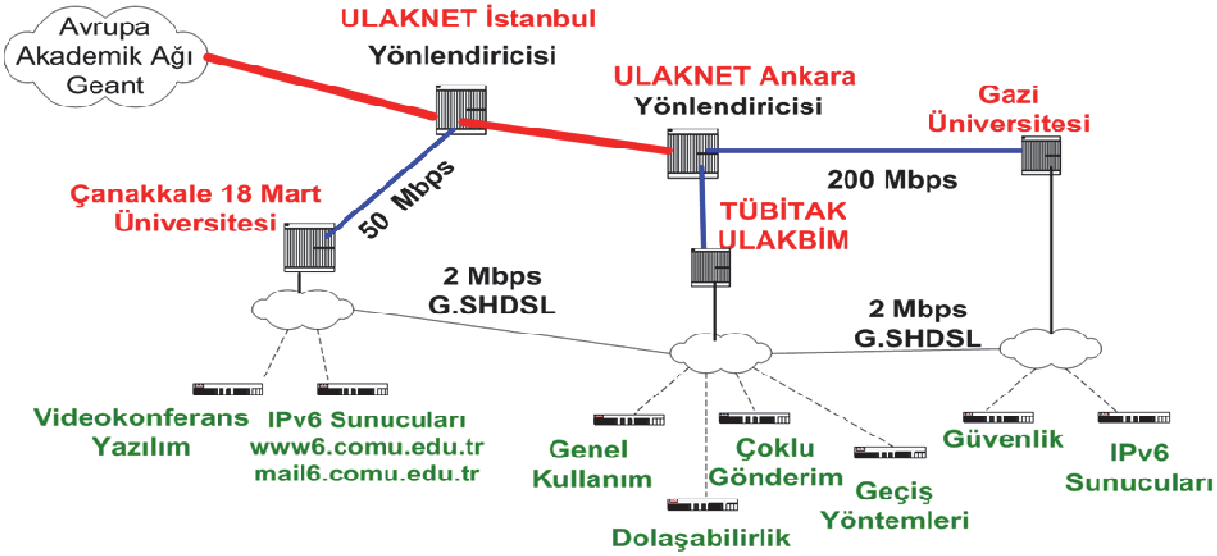
(4) (3) 'de verilen komut satırı desteğini sağlayan yeni kodlar programa eklenir.

```
//DV46 programın Date = D, Version = V, IPv4 = 4, IPv6 = 6 için tanımlandı.
while ((c = getopt(argc, argv, "DV46")) != -1) {
    ......
    // IPv4 ise
    case '4':
    af = AF_INET;
    break;
    // IPv6 ise
    case '6':
    af = AF_INET6;
    break;
    ......
}
```

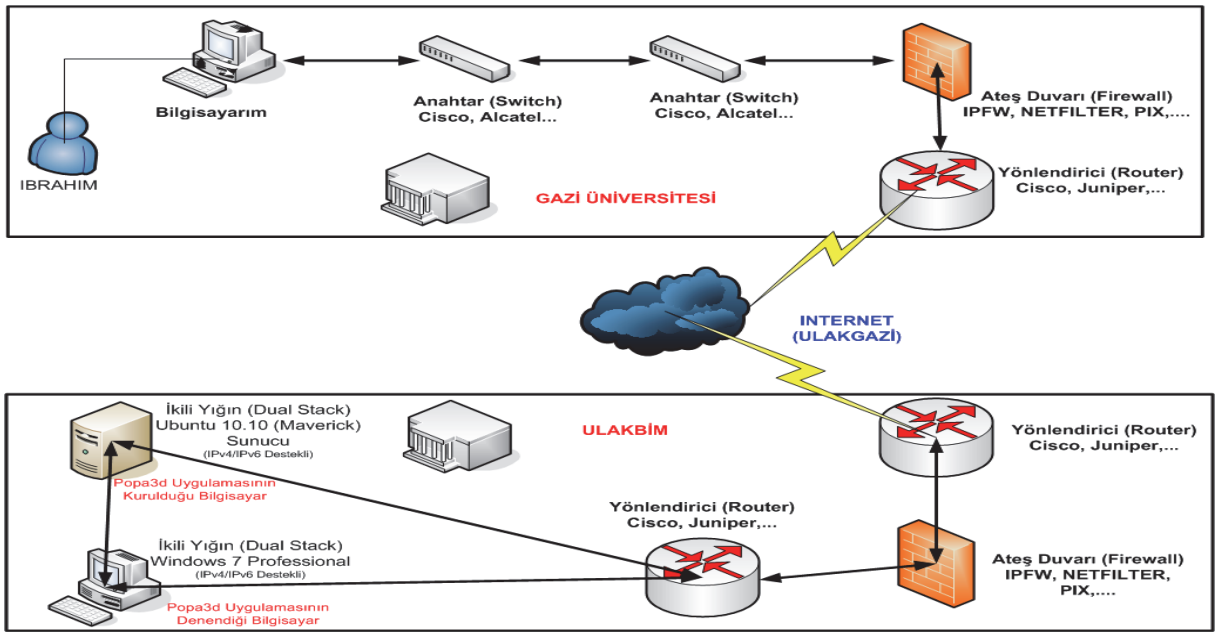
Yapılan değişikliklerden sonra Popa3d uygulaması artık IPv4 adreslerinin yanında IPv6 adreslerini de işleyecek ve komut satırından **-6** parametresini de kabul edecektir [1,6-13].

2.3.2. Uygulama test ortamı (The application test environment)

Geliştirilen IPv6 destekli Popa3d uygulaması IPv6 erişiminin ULAKBİM tarafından sağlandığı "Ulusal IPv6 Protokol Altyapısı Tasarımı ve Geçişi Projesi" kapsamında geliştirilen ve Şekil-3'te genel topolojisi gösterilen IPv6 Test Yatağı ve Geliştirme Ortamı (IPv6-GO)'nda test edilmesi amaçlanmıştır. Fakat, IPv6 destekli Popa3d uygulaması proje süresinde tamamlanamamıştır. Ayrıca, "Ulusal IPv6 Protokol Altyapısı Tasarımı ve Geçişi Projesi" tamamlandıktan sonra Gazi Üniversitesi Bilgisayar Mühendisliği Bölümünde kurulan güvenlik laboratuvarındaki IPv6 bağlantısı maddi sebeplerden dolayı kesilmek zorunda kalmıştır. Bu yüzden uygulama bu test ortamı yerine, ULAKBİM tarafından sağlanan test ortamı ile bir bağlantı ortamı oluşturulmuş ve oluşturulan bu IPv6 destekli ağ ortamında denenmiştir. ULAKBİM tarafından sağlanan ve uygulamaya ait test ortamı ağ topolojisi Şekil-4'te gösterilmiştir [1,4].



Şekil 3. IPv6-GO ağ topolojisi [4] (Topology for IPv6-GO network)

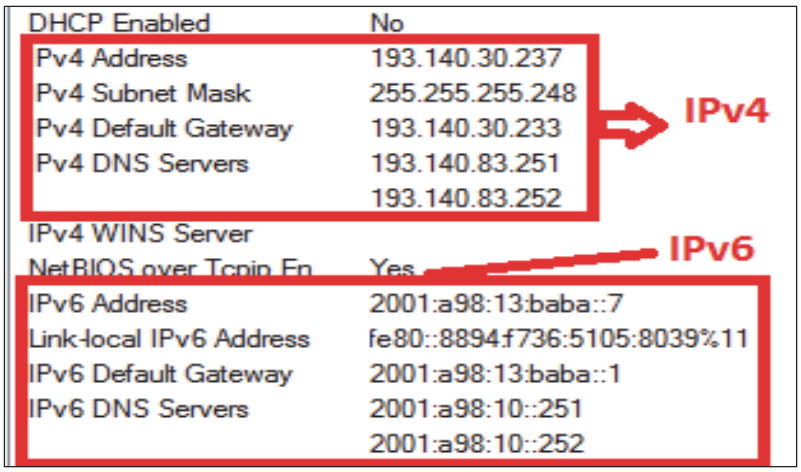


Şekil 4. Uygulama test ortamı genel ağ topolojisi [1] (General network topology for application test platform)

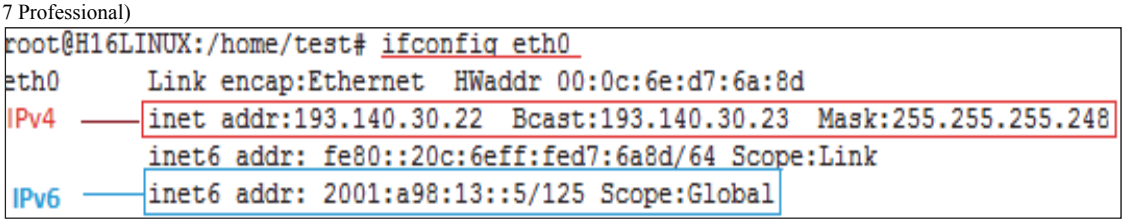
ULAKBİM tarafından IPv6 destekli test ortamı sağlandıktan sonra Gazi Üniversitesi Bilgi İşlem Dairesi Başkanlığı'nda bulunan sadece IPv4 adresine sahip bir bilgisayar üzerinden

ULAKBİM'de bulunan ikili yığın mekanizmasına sahip (IPv4/IPv6) Linux Ubuntu 10.10 Maverick yüklü sunucuya uzaktan erişim sağlanmıştır. Uzaktan erişim sağlanan sunucu üzerine geliştirilen uygulamanın dosyaları aktarılmıştır.

Windows 7 Professional işletim sistemi yüklü istemci bir bilgisayara ait IP bilgileri Şekil-5'te verilmiştir.



Şekil 5. Windows 7 Professional IP bilgilerini içeren ekran görüntüsü [1] (Screenshot having IP information for Windows 7 Professional)



Şekil 6. Linux Ubuntu 10.10 (maverick meerkat) sunucu IP bilgileri [1] (IP information for Linux Ubuntu 10.10 (maverick meerkat) server)

Linux Ubuntu 10.10 (maverick meerkat) yüklü sunucuya ait IP bilgileri Şekil-6'da verilmiştir . Test ortamında bulunan istemci ve sunucu IP adres bilgileri yukarıda verilmiş olup uygulamanın kurulumuna ait aşamalar şu şekildedir;

(1) Sunucuda **/home/test/iaksit/tez/** klasörüne aktarılmış uygulamanın **make** komutu (root@H16LINUX:/home/test/iaksit/tez# **make**) ile derlenip Şekil-7'de derleme sonucu verilmiştir.

(2) Derleme işlemi hatasız yapıldıktan sonra **make install** komutu (root@H16LINUX:/home/test/iaksit/tez# **make install**) ile uygulama sunucuda çalışır duruma getirilir. Şekil-8'de uygulamanın yükleme durumu ve sonucu verilmiştir.

(3) Uygulama yüklendikten sonra, **popa3d -D** komutu (root@H16LINUX:/home/test/iaksit/tez# **popa3d –D**) yazılarak program çalıştırılır. Popa3d uygulamasının çalışıp çalışmadığı **netstat -tapn** komutu ile görülür. Şekil-9'da uygulamanın hizmet verme durumuna ait bilgiler verilmiştir.

(4) Windows 7 Professional yüklü istemci üzerinde bulunan Putty programı kullanılarak sunucu üzerinde bulunan IPv6 destekli hizmet vermeye hazır uygulamaya telnet bağlantısı 110 port numarası sağlanmıştır. Şekil-10'da Windows 7 Professional istemci üzerinden 110 port numaralı POP3 bağlantı durumu verilmiştir. Şekil-11'de ise istemci tarafından gönderilen TCP 110 bağlantısına ait sunucu üzerinde bulunan popa3d uygulamasının hizmet verme durumu verilmiştir.

(5) Windows 7 Professional yüklü istemcide kurulan Microsoft Office Outlook programı aracılığıyla geliştirilmiş olan Popa3d uygulamasının gelen e-postaları alıp almadığı test edilmiştir. Ayrıca e-posta gönderme yazılımı olarak da sunucu üzerine yüklenen Postfix uygulaması kullanılmıştır. Şekil-12'de Microsoft Office Outlook programında e-posta göndermek ve almak için yapılan yapılandırmalar verilmiştir.

(6) Microsoft Office Outlook programı ile POP3 hizmeti yapılandırmasından sonra geliştirilen uygulamayı test etmek amacıyla dosya menüsünden yeni ileti tıklanarak Şekil-13'te gösterilen bilgiler girilmiştir.

(7) Microsoft Office Outlook programı aracılığıyla gönderilen e-posta iletisinin ulaşıp ulaşmadığını test etmek için araç çubuğunda bulunan gönder/al bağlantısına tıklanmıştır. Geliştirilen uygulamanın Microsoft Office Outlook aracılığı ile gelen e-postaları sorunsuz bir şekilde aldığı Şekil-14'te verilmiştir.

```
root@H16LINUX:/home/test/iaksit/tez# make
gcc -Wall -O2 -fomit-frame-pointer -c version.c
gcc -Wall -O2 -fomit-frame-pointer -c startup.c
gcc -Wall -O2 -fomit-frame-pointer -c standalone.c
standalone.c: In function âdo_standaloneâ:
standalone.c:248: warning: pointer targets in passing argument 3 of âacceptâ dif
fer in signedness
/usr/include/sys/socket.h:214: note: expected âsocklen_t * __restrict__â but arg
ument is of type âint *â
standalone.c:204: warning: ignoring return value of âchdirâ, declared with attri
bute warn_unused_result
gcc -Wall -O2 -fomit-frame-pointer -c virtual.c
gcc -Wall -O2 -fomit-frame-pointer -c auth_passwd.c
gcc -Wall -O2 -fomit-frame-pointer -c auth_shadow.c
auth_shadow.c: In function âauth_userpassâ:
auth_shadow.c:58: warning: ignoring return value of âwriteâ, declared with attri
bute warn_unused_result
gcc -Wall -O2 -fomit-frame-pointer -c auth_pam.c
gcc -Wall -O2 -fomit-frame-pointer -c pop_root.c
gcc -Wall -O2 -fomit-frame-pointer -c pop_auth.c
gcc -Wall -O2 -fomit-frame-pointer -c pop_trans.c
gcc -Wall -O2 -fomit-frame-pointer -c protocol.c
gcc -Wall -O2 -fomit-frame-pointer -c database.c
gcc -Wall -O2 -fomit-frame-pointer -c mailbox.c
gcc -Wall -O2 -fomit-frame-pointer -c misc.c
gcc -Wall -O2 -fomit-frame-pointer -c md5/md5.c -o md5/md5.o
gcc -s version.o startup.o standalone.o virtual.o auth_passwd.o auth_shadow.o au
th_pam.o pop_root.o pop_auth.o pop_trans.o protocol.o database.o mailbox.o misc.
o md5/md5.o  -lcrypt -o popa3d
root@H16LINUX:/home/test/iaksit/tez#
```

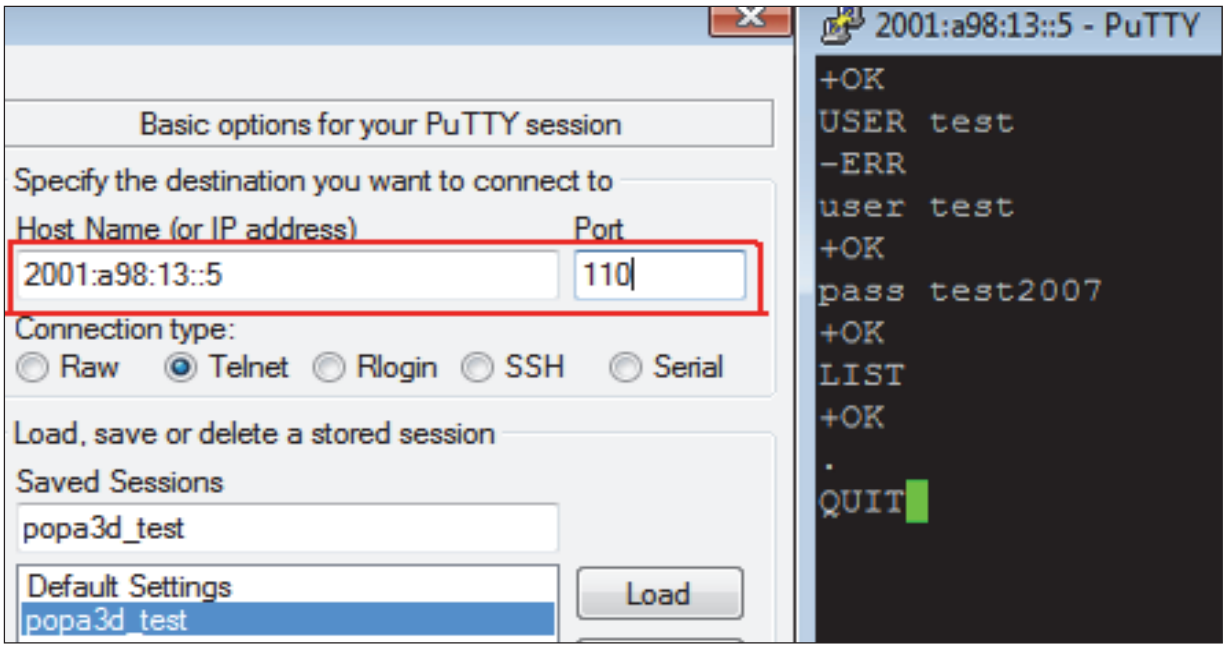
Şekil 7. Uygulama derleme ve sonucu [1] (Application compiling and its result)

```
root@H16LINUX:/home/test/iaksit/tez# make install
mkdir -p -m 755 /usr/local/sbin /usr/local/man/man8
install -c -m 700 popa3d /usr/local/sbin/
install -c -m 644 popa3d.8 /usr/local/man/man8/
root@H16LINUX:/home/test/iaksit/tez#
```

Şekil 8. Uygulama yükleme durumu ve sonucu [1] (Application installation and its result)

```
root@H16LINUX:/home/test/iaksit/tez# netstat -tapn
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      565/sshd
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      751/cupsd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      1226/exim4
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      941/mysqld
tcp        0    240 193.140.30.22:22        194.27.15.253:50091     ESTABLISHED 4008/sshd: test [pr
tcp6       0      0 :::80                   :::*                    LISTEN      1371/apache2
tcp6       0      0 :::22                   :::*                    LISTEN      565/sshd
tcp6       0      0 ::1:631                 :::*                    LISTEN      751/cupsd
tcp6       0      0 ::1:25                  :::*                    LISTEN      1226/exim4
tcp6       0      0 :::110                  :::*                    LISTEN      4492/popa3d
root@H16LINUX:/home/test/iaksit/tez#
```

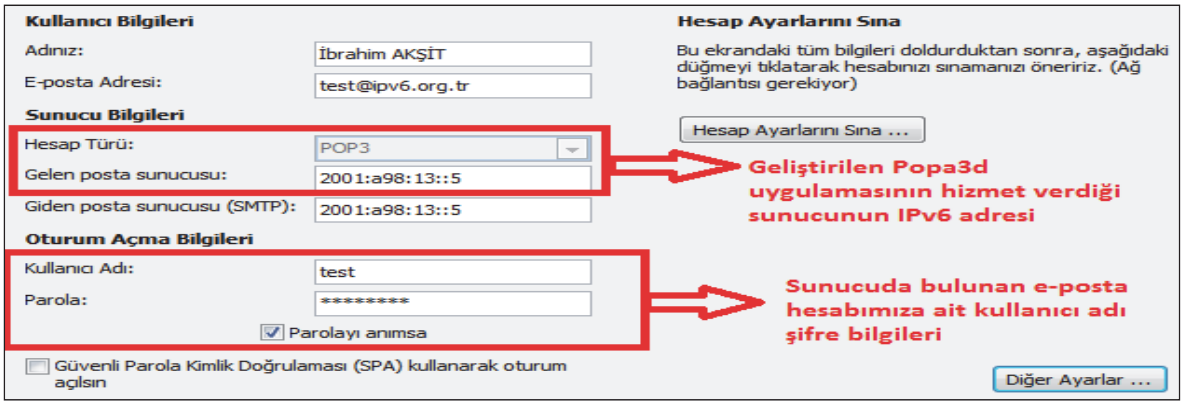
Şekil 9. Uygulama hizmet verme durumu ve sonucu [1] (Application service status and its result)



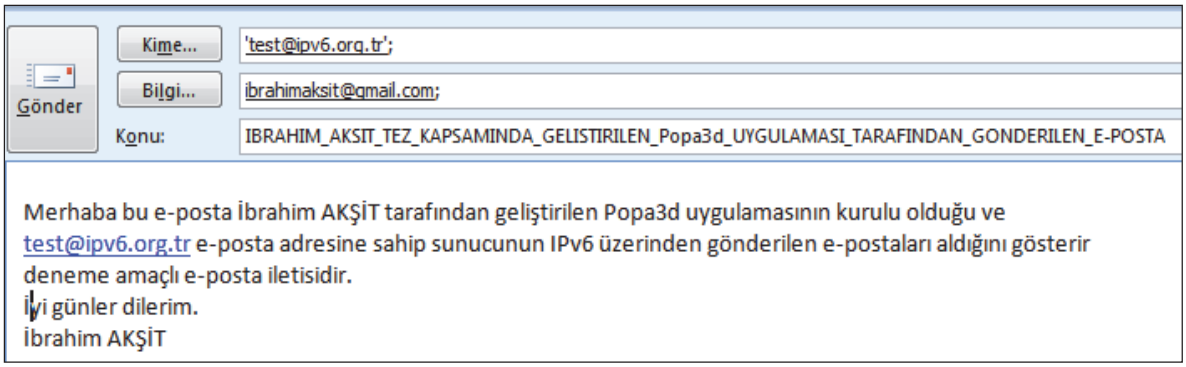
Şekil 10. Windows 7 üzerinden 110 portuna telnet bağlantısı [1] (Telnet connection to Windows 7 via 110 port)

```
root@H16LINUX:/home/test/iaksit/tez# netstat -tapn
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      565/sshd
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      751/cupsd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      1226/exim4
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      941/mysqld
tcp        0     48 193.140.30.22:22        194.27.15.253:50091     ESTABLISHED 4008/sshd: test [pr
tcp6       0      0 :::80                   :::*                    LISTEN      1371/apache2
tcp6       0      0 :::22                   :::*                    LISTEN      565/sshd
tcp6       0      0 ::1:631                 :::*                    LISTEN      751/cupsd
tcp6       0      0 ::1:25                  :::*                    LISTEN      1226/exim4
tcp6       0      0 :::110                  :::*                    LISTEN      4492/popa3d
tcp6       0      0 2001:a98:13::5:110      2001:a98:13:baba::49303 ESTABLISHED 4515/popa3d
root@H16LINUX:/home/test/iaksit/tez#
```

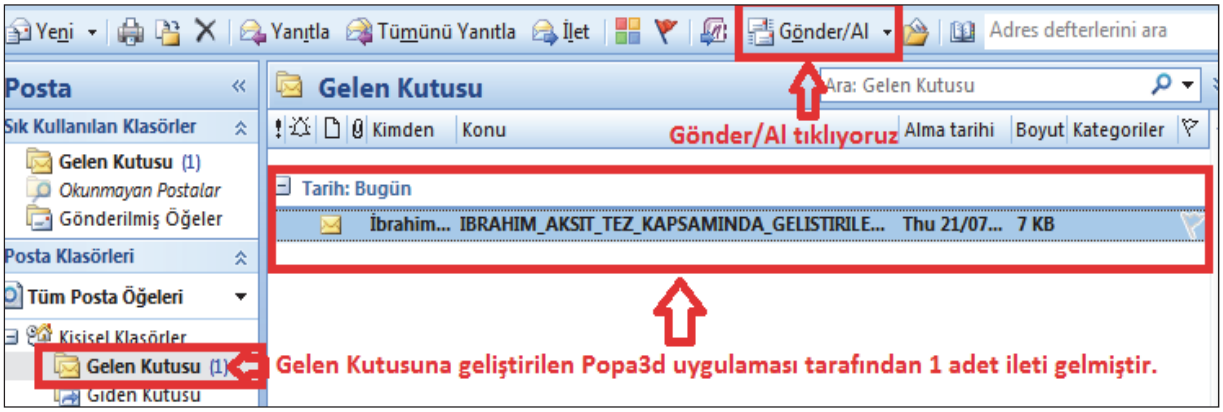
Şekil 11. Uygulamanın 110 portuna hizmet vermesi [1] (Application serving via 110 port)



Şekil 12. Uygulamanın Microsoft Office Outlook ile POP3 yapılandırması [1] (Application setup via POP3 in Microsoft Office Outlook)



Şekil 13. Microsoft Office Outlook ile yeni ileti gönderimi [1] (Sending a new message via Microsoft Office Outlook)



Şekil 14. Microsoft Office Outlook ile gelen iletinin alınması [1] (Receiving a new message via Microsoft Office Outlook)

3. SONUÇLAR VE ÖNERİLER (CONCLUSIONS AND RECOMMENDATIONS)

Bu çalışmada IPv6 paketleri üzerinden gelen e-postaları güvenli bir şekilde almak için gerekli düzenlemeler ve program geliştirilmesi yapılmış, Microsoft Office Outlook programı kullanılarak e-postaların geliştirilen uygulama üzerinden alındığı gösterilmiş, geliştirilen POP3 uygulaması ULAKBİM tarafından sağlanan IPv6 destekli ağ ortamında test edilmiş, geliştirilen uygulamanın IPv6 paketlerine hizmet verme durumu ve sonucu şekillerle gösterilmiştir. Yapılan testler sonucunda, bu çalışma kapsamında önerilen uygulamanın başarılı bir şekilde çalıştığı tespit edilmiştir.

Mevcut uygulamaların IPv6 ile kullanılabilmesi teknolojik olduğu kadar günlük yaşamımız açısından da önemlidir. IPv6'ya geçişin hızla devam ettiği bu günlerde ulusal, bölgesel ve küresel anlamda hizmet verilen ve alınan uygulamaların IPv6 desteğine sahip olması büyük önem arz etmeye başlamıştır. Uygulamaların IPv6 hizmeti verebilecek duruma getirilebilmesi için akademik çalışmaların yapılması, IPv6'ya gerek uygulama gerekse teknoloji olarak hazır olunması ülkemiz için önem arz etmektedir. Uygun altyapı desteğinin sağlanmasından sonra uygulamaların IPv6 desteğine sahip olması gerekmektedir.

Mevcut uygulamaların IPv6 trafiğini kullanabilmeleri için öncelikli olarak soket tabanlı yazılımlarda IPv6 yapılandırılmasının bulunması gerekmektedir. Sunucu ve istemci tabanlı yazılımların, IPv6 desteğinin sağlanması ve IPv6 paketlerinin işlenebilmesi, IPv6 dünyasında dolaşarak gerekli yerlere ulaştırılması ve IPv6 destekli uygulamaların geliştirilmiş olması çalışmanın sağladığı katkılar arasındadır.

Sonuç olarak; mevcut sunucu tabanlı bir POP3 uygulamasına IPv6 desteğinin sağlanması gibi örneklerin ülkemizin bu yeni teknolojiye geçmesi, bu alanda karşılaşılabilecek problemlerin kısa sürede çözülmesi ve en önemlisi bu alanda yapılacak bu gibi yeni çalışmaların, geliştirilecek olan yeni teknolojilerin, uygulamaların ve yapılacak olan araştırmaların artmasına katkılar sağlayacağı değerlendirilmektedir.

Son olarak, Ulusal IPv6'ya geçiş stratejisi gereği tüm alt yapıların buna adapte edilmesi için verilen takvim gereği de en az bir uygulamanın IPv6'ya geçirilmesi zorunluluğuna bu çalışmanın katkı sağlayacağı değerlendirilmektedir.

TEŞEKKÜR (ACKNOWLEDGEMENT)

Bu çalışma Ulusal IPv6 Protokol Altyapısı Tasarımı ve Geçişi Projesi – 108G027 kapsamında geliştirilmiş olup projenin test ortamı kullanılarak geliştirilen uygulama test edilmiştir. Değerli katkı ve

yardımlarından dolayı tüm Ulusal IPv6 Protokol Altyapısı Tasarımı ve Geçişi Projesi çalışanlarına, bu projeye destek veren TÜBİTAK'a teşekkür ederiz.

KAYNAKLAR (REFERENCES)

1. Akşit, İ., **IPv6 Yapılandırması ve Soket Tabanlı IPv6 Destekli Sunucu Yazılımı Geliştirilmesi**, Yüksek Lisans Tezi, Gazi Üniversitesi, Bilişim Enstitüsü, 2011.
2. Sağıroğlu Ş., Karacan H., Alkan M., Ünver M., Orcan S., Yavanoğlu U., "Design of National IPv6 Infrastructure and Transition to IPv6 Protocol: Questionnaire Study", **National IPv6 Conference**, Ankara, 19-25, 2011 (In Turkish).
3. Sağıroğlu Ş., Alkan M., H. Karacan, Parıltı N., Kasap R., Yavanoğlu U., Hacıömeroğlu M. and Akşit İ., "A Study on Cost Analysis For IPv6 Protocol Transition", **Journal of the Faculty of Engineering and Architecture of Gazi University**, Vol 27, No 1, 151-161, 2012 (In Turkish).
4. İnternet: TÜBİTAK-ULAKBİM, Gazi Üniversitesi ve Çanakkale 18 Mart Üniversitesi, "Design of National IPv6 Infrastructure and Transition to IPv6 Protocol", http://www.ipv6.net.tr/, 2012 (In Turkish).
5. Aktaş, M. and Sağıroğlu, Ş., "IPv6: International studies and status in Turkey", **National IPv6 Conference**, Ankara, 5-7, 2011 (In Turkish).
6. Ahrouch, A. A. and Ezzine, S., "IPv4 to IPv6 Migration", Analytical Network Project, Master Program System and Network Engineering, **University of Amsterdam,** 5-17, 2004.
7. Hagino, J. I., "Address-family independent socket programming for IPv6", ***KAME/WIDE Project***, Research Laboratory, Internet Initiative Japan, 2004.
8. İnternet: Morr, D., "IPv6 Programming", IPv6 Deployment, PennState University, https://wikispaces.psu.edu/display/ipv6/Home, 2009.
9. İnternet: Stevens, W., Thomas, M., Nordmark, E. and Jinmei T., "Advanced Sockets Application Program Interface (API) for IPv6", **Internet Engineering Task Force (IETF), RFC 3542,** http://www.ietf.org/rfc/rfc3542.txt, 5-52, 2003.
10. İnternet: Gilligan, R., Thomson, S., Bound, J. and Stevens, W., "Basic Socket Interface Extensions for IPv6", **Internet Engineering Task Force (IETF), RFC 2553,** http://www.ietf.org/rfc/rfc2553.txt, 3-35, 1999.
11. Bieringer, P., "Status of IPv6 (Information & Workshop)", **IEEE CNF, Applications and the Internet Workshops**, 54-57, 2005.
12. Stallings, W., "Computer Networking with Internet Protocols And Technology International Edition", **Prentice Hall**, **Pearson**, USA, 290-300, 2004.
13. Hagen, S., "IPv6 Essentials 2nd Edition", **O'Reilly, O'Reilly Media**, 2006.