**T.C.**
**GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ**
**MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ**

# AKTİF AĞLAR İÇİN IPSEC TABANLI GÜVENLİK SİSTEMİ TASARIMI

**Aydın KOÇAK**

**YÜSKEK LİSANS TEZİ**
**BİLGİSAYAR MÜHENDİSLİĞİ**
**ANABİLİM DALI**

**GEBZE**
**2006**

**T. C.**
**GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ**
**MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ**

# AKTİF AĞLAR İÇİN IPSEC TABANLI GÜVENLİK SİSTEMİ TASARIMI

**Aydın KOÇAK**

**YÜKSEK LİSANS TEZİ**
**BİLGİSAYAR MÜHENDİSLİĞİ**
**ANABİLİM DALI**

**TEZ DANIŞMANI**
**Yrd. Doç. Dr. İbrahim SOĞUKPINAR**

**GEBZE**
**2006**

# ÖZET

Günümüzde ağ uygulamalarının sayısının oldukça artmasının doğal sonucu olarak bant genişliği ihtiyacı da sürekli artmaktadır. Artık neredeyse ağ'a bağlı olmayan bir bilgisayar düşünülemez diyebiliriz. Tarihsel sürece bakıldığı zaman bant genişliği ihtiyacı sürekli olarak hat kapasitelerinin artırılması yöntemi ile giderilmeye çalışılmıştır. Örnek vermek gerekirse bundan beş altı yıl önce 10 Mbps. lik bağlantı hızı yerel bağlantılar için yeterli idi ağ üzerinde koşan uygulama sayısı ve çeşidi arttıkça bu hız sırasıyla 100 Mbps, 1.000 Mbps(1Gbps), 10.000 Mbps(10Gbps) ve en son 40.000 Mbps(40Gbps)' lere kadar çıktı ve daha da çıkacak.

Bu da gösteriyor ki bant genişliği ihtiyacını hat kapasitesini artırarak çözmeye çalışmak tek başına yeterli bir yöntem değil. İşte bu nokta da hesaplama üzerinde de değişiklik ve iyileştirme yapmak fikri ortaya çıkıyor. Bu fikir bağlamında "Aktif Ağlar"; Üzerlerinden geçen trafik üzerinde uygulama tipine göre özelleştirilmiş hesaplamalar uygulayabilen düğümler içeren yeni bir "Ağ Modeli" olarak karşımıza çıkıyor. "Aktif Ağlar" kısa sürede araştırmacıların üzerlerinde en çok zaman harcadığı ilgilerin yoğunlaştığı bir alan oldu. Bu çalışma alanı ticari üreticileri de cezp etmekle birlikte güvenlik başta olmak üzere aşılması gerekli birçok zorluk ve belirsizlik içerdiği için laboratuarlardan çıkamamıştır.

Bu çalışmada, önce kısaca mevcut ağ teknolojilerine değinilmiş, daha sonra aktif ağlar tanıtılmış, aktif ağ bileşenleri ve yapısı ayrıntılı bir şekilde incelenmiştir. Ardından aktif ağlar ile mevcut ağ teknolojileri karşılaştırılmıştır. Bunu takiben aktif ağların laboratuar ortamından çıkamamasına neden olan zorluk ve belirsizliklere kısaca değindikten sonra bu zorluklar ve belirsizlikler ayrıntılı bir şekilde ortaya konmuştur. Son olarak mevcut zorluklardan bir kısmını ortadan kaldırmak üzere önerilen "Aktif Ağlarda IPSEC ile Güven Yönetimi" ile verilerin şifrelenmesi ve bir güven mekanizmasına dayalı model ve uygulanması anlatılmış, modelin sınandığı testbed tanıtılmıştır. Testbed üzerinde uygulanan model ile ilgili test sonuçları verilmiştir. Son olarak modelin bu sınamalara göre bir değerlendirilmesi yapılmış ve ileri de yapılması gerekli eklentiler ve eksiklikler belirlenmiştir.

# SUMMARY

In the recent years, bandwidth requirements have increased continuously in parallel with the rice in the number of network application. Today nearly all the current computers worldwide are connected to a network.

The general approach to bandwidth problem has usually been by increasing the line capacities. For example until 5 years ago, 10 Mbps. connection speeds were enough for local area networks. With the dramatic rise in the number and type of applications running on local area networks, line speeds were progressively increased to 100Mbps., 1000Mbps. (1 Gbps.), 10000 Mbps.(10Gbps.) and finaly 40000(40Gbps.).

However, even this connection capacity will not be sufficient for future bandwidth requirements. Thus, upgrading the speeds is not a sufficient method to overcome the bandwidth bottleneck problem.

An alternative idea is changing and improving the calculation methods. In this context, active networks emerge as a new network model containing nodes which can perform application specific calculations on the traffic passing through.

For this reason, active networks rapidly became a widely researched field. However, despite its conformed to laboratories, as it involved many difficulties and uncertainties, particularly security.

In this thesis, firstly introduced to current LAN technologies have been given, followed by a detailed analysis on active networks, with their nodes and structure. Then after listing the difficulties and uncertainties that limit the commercial implementation of active networks, these are analyzed one by one in detail.

Finally, “Trust Management in Active Networks by IPSEC” based on encryption of data and trust management model and application, which is proposed in order to overcome some difficulties, is explained and test bed on which the model is implemented is introduced.

# TEŞEKKÜR

Tez konumu belirlememden bitirmeme kadar sürekli yol gösteren ve destek olan danışmanım Sayın Yrd. Doç. Dr. İbrahim Soğukpınar' a, yardımlarından dolayı arkadaşlarım Araş. Gör. Türker Akyüz, Araş. Gör. Rahim Karabağ ve Araş. Gör. H. Türker Şahin'e teşekkürü bir borç bilirim.

# İÇİNDEKİLER

# SİMGELER VE KISALTMALAR DİZİNİ

| | |
|---|---|
| AN | Active Network |
| ANTS | Active Node Transfer System |
| SANTS | Secure Active Node Transfer System |
| JANOS | Java-oriented Active Network Operating System |
| ABONE | Active Network Backbone |
| IP | Internet Protocol |
| IPSEC | Internet Protocol Security |
| JNODEOS | Java Based Network Operating System |
| DOS | Denial Of Service |
| SSL | Secure Socket Layer |
| ASIC | Application Specific Integrated Circuite |
| LAN | Local Area Network |
| WAN | Wide Area Network |
| MAC | Media Access Control |
| OSI | Open System Interconnection |
| UTP | Unshielded Twisted Pair |
| IPX | Internetwork Packet Exchange |
| SNA | Systems Network Architecture |
| DECNET | Digital Equipment Corporate Network |
| DLCI | Data Link Connection Identifier |
| STP | Spanning Tree Protocol |
| ESP | Encapsulating Security Payload |
| IKE | Internet Key Exchange |
| AH | Authentication Header |
| PGP | Pretty Good Privacy |
| SSH | Secure Shell |
| DNS | Domain Name System |

# ŞEKİLLER DİZİNİ

# ÇİZELGELER DİZİNİ

**Çizelge** **Sayfa**

# 1 GİRİŞ

Aktif ağlar bugünkü pasif iletim gerçekleştiren ve protokol seviyesinde soyutlama yapabilen ağların yeni ağ kaynaklarını programlamak için API seviyesinde soyutlama yapabilen çok temel bir programlanabilir ağa gelişimidir.[ Tennenhouse,et all, 1997]

Geleneksel veri ağlarında, veriler bir yerden başka bir yere aktarılırken üzerlerinde herhangi bir değişiklik yapılmaz. Pasif iletim de sadece temel başlık işlemleri uygulanabilir. Aktif Ağlarda ise herhangi bir ağ bileşeni üzerinden geçen trafik üzerinde uygulamaya veya kullanıcıya özel hesaplamalar yapabilir.[ Karnouskos, 2001]

Bu tür aktif ağların kullanımı ile birlikte işe özel hesaplamalar kullanılarak ağ iletimi hem en uygun hale gelecek hem de bilgi kalitesi artacaktır. Fakat aynı zamanda çok çeşitli zorluklarda ortaya çıkacaktır. Böyle bir altyapıda temel zorluk; esneklik, performans, sağlamlık, kullanılabilirlik ve güvenlik gereksinimleri ile ilgili doğru dengeyi bulmaktır. Çok güvenli bir ağ tasarlanabilir. Ancak yukarıda ki denge tutturulamazsa yani performans, kullanılabilirlik iyi değilse güvenliğin bir anlamı yoktur. Bu çalışmada sistemi güvenli hale getirirken bu denge mümkün olduğunca gözetilmeye çalışılmıştır.

Aktif ağlar özellikle güvenlik ve standartlaşma eksikliği gibi temel zorluklar nedeni ile laboratuar ortamından üretim ortamına girememiştir. Bu çalışmada güvenlik zorluklarının bir kısmını şifreleme, yetkilendirme tekniklerinin kullanıldığı ve laboratuar ortamında test edildiği “Aktif Ağlarda IPSEC ile Güven Yönetimi” modeli ile aşılmıştır.

Aktif ağlarındaki zorlukların büyük bir kısmını ileriki bölümlerde de anlatılacağı gibi güvenlik zorluk ve belirsizlikleri oluşturmaktadır. En temel zorluklar bir düğümün kendisine ait olmayan verilere yetkisiz olarak ulaşabilmesi, kötü niyetli bir kodun düğüm üzerinde kendisini kopyalayarak veya çalıştırarak servis durdurma, kötü niyetli bir düğümün kendisine ait olmayan bir kodu alıp değiştirip göndermesi gibi zorluklardır.

Bu çalışmada önerilen model temel olarak düğümler arasında IPSEC kullanılarak, üçüncü katmanda verilerin şifrelenmesi ve güven yönetimi sayesinde düğümlerin kimliklerinin ve güvenilecek düğümlerin adım adım tanımlanmasına dayanmaktadır. Önerilen model ile yukarıda bahsedilen kodun çalınması durumu tehdit olmaktan çıkmıştır. Şifrelenmiş kod ele geçirilse bile bir anlam ifade etmez. Aktif düğümler arasında güven mekanizması kullanıldığı için aktif düğümler sadece güvendikleri düğümlerden gelen paketlerin düğümü programlamasına izin verecektir. Böylece bir sorun oluşsa bile bu sorunun kaynağı belirlenecek ve ileriye dönük önlemler alınabilecektir.

Bu çalışma ile aktif kodun güvenliği şifreleme algoritmalarının gücü seviyesinde neredeyse %100 sağlanmıştır. Aktif düğüm ile ilgilide yukarıda anlatılan savunma mekanizmaları ile güvenlik sağlanmaya çalışılmıştır. Ancak bu çalışma ile aktif ağlar üzerindeki bütün güvenlik zorluklarının ortadan kaldırıldığını söylenemez. Ama bu çalışmaya ek birkaç yöntemle güvenlik ile ilgili zorlukların büyük bir bölümü ortadan kalkacaktır. Yukarıda değinilen üreticilerin aktif cihazlar üretmesindeki en büyük engelin güvenlik olmadığı belirli bir standardın oluşturulmaması olduğu düşünülmektedir.

Önerilen model oluşturulan test bed üzerinde test edilmiş ve sonuçları ayrıntılı olarak anlatılmıştır. Güvenlik önlemlerinin artırılması ile performans da önemli kayıplar söz konusu olmaktadır. Önerilen modelde bu kıstas göz önünde bulundurularak güvenlik ile performans dengelenmiştir. Bununla ilgili ayrıntılı bilgi sayısal sonuçlara ileriki bölümlerde değinilmiştir.

# 2 GENEL KAVRAMLAR VE MEVCUT ÇALIŞMALAR

Aktif ağları ve getirilerini iyi anlayabilmek için mevcut ağ teknolojileri kabaca incelenmelidir. Mevcut ağ teknolojilerindeki en temel kavramlar olan anahtarlar, yönlendiriciler ve bunların yaptıkları anahtarlama ve yönlendirme işlemleri ayrıntılı olarak incelenmiştir. Aktif ağlar içinde yönlendirme ve anahtarlamaya katılabilecek özellikler tartışılmış, aktif düğümler ile ilgili genel kavramlara değinilmiştir.

## 2.1 Geleneksel Veri Ağları

Geleneksel veri ağlarında eskiden çok sayıda standart ve teknoloji olmakla birlikte artık günümüzde Ethernet teknolojisi ve TCP/IP protokol süiti artık standart hale gelmiştir. Genel olarak geleneksel veri ağlarında yerel ağlar için, anahtarlar, geniş alan ağlarında ise anahtar ve yönlendiriciler kullanılmaktadır.

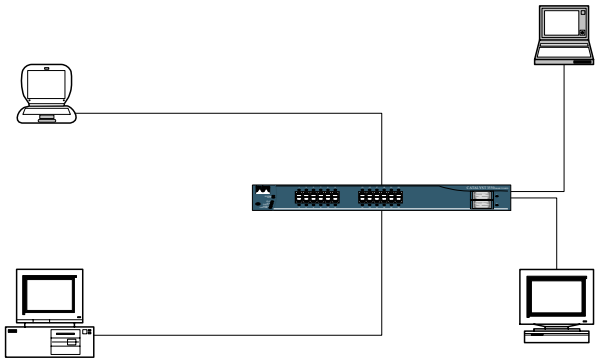
Günümüzde yerel ağlarda iletim ortamı olarak fiber veya bakır ( UTP CAT – 5 ) kablolar veya hava (kablosuz iletim) , geniş alan ağları için genel olarak kiralık hatlar, uydu veya hava ( kablosuz iletim) kullanılmaktadır.

Kampus ağları, orta veya büyük ölçekli yerel ağlar içinde sadece anahtar kullanılıyor demek yanlış olur genel de yönlendirici özelliği de bulunan anahtarlar veya anahtar ve yönlendiriciler sıklıkla bir arada kullanılmaktadır.

### 2.1.1 Anahtar ve Anahtarlama Teknolojileri

Anahtarlama genel olarak devre anahtarlama, paket anahtarlama ve hücre anahtarlama olmak üzere üçe ayrılır. Geniş alan ağlarında kiralık hatlar için devre anahtarlama, yerel ağlarda ise paket anahtarlama, hücre anahtarlama ise hem geniş ağlarda hem de yerel ağlarda kullanılır.

Anahtarlar, yerel ağlar için Ethernet ağlarında **ASIC** yongalar kullanılarak MAC adresine göre iletim gerçekleştiren cihazlardır. Anahtarlama işlemi OSI Referans Modelinde ikinci katmanında (veri bağı) gerçekleşmektedir. Anahtarlar, her bağlantı noktası üzerinde bulunan cihazın MAC adresini tuttuğu bir MAC adres tablosu sayesinde hangi bağlantı noktasında hangi adresin olduğunu bilip veri paketlerini bu tabloya göre iletme işini yaparlar. Anahtarlardan oluşan ağlar bir broadcast ağ oluşturur.



**Şekil 2.1 Yerel Alan Anahtarlama Örnek Ağı**

| Hedef Adres | Kaynak Adres | Uzunluk | Data | FCS |
|---|---|---|---|---|
| 1111.1111.1c | 1111.1111.1a | 1514 | 10110…1 | 4 Byte |

**Şekil 2.2 İkinci Katman Çerçeve Formatı ve Örnek Çerçeve**

| MAC Adres Tablosu I. Durum | |
|---|---|
| MAC Adresi | Port No |
| | |
| | |
| | |
| | |
| … | … |

| MAC Adres Tablosu II. Durum | |
|---|---|
| MAC Adresi | Port No |
| 1111.1111.1a | 2 |
| 1111.1111.1c | 7 |
| | |
| | |
| … | … |

| MAC Adres Tablosu Final | |
|---|---|
| MAC Adresi | Port No |
| 1111.1111.1a | 2 |
| 1111.1111.1c | 7 |
| 1111.1111.1b | 4 |
| 1111.1111.1d | 5 |
| … | … |

**Şekil 2.3 Örnek Yerel Ağ MAC Adres Tabloları**

Yerel ağ üzerinde Şekil – 2.1 deki "A" bilgisayarı "C" bilgisayarına bilgi göndermek istesin bunun anahtarlama işleminin nasıl geliştiğini Şekil – 2.2 ve Şekil – 2.3' ü kullanarak açıklanırsa;

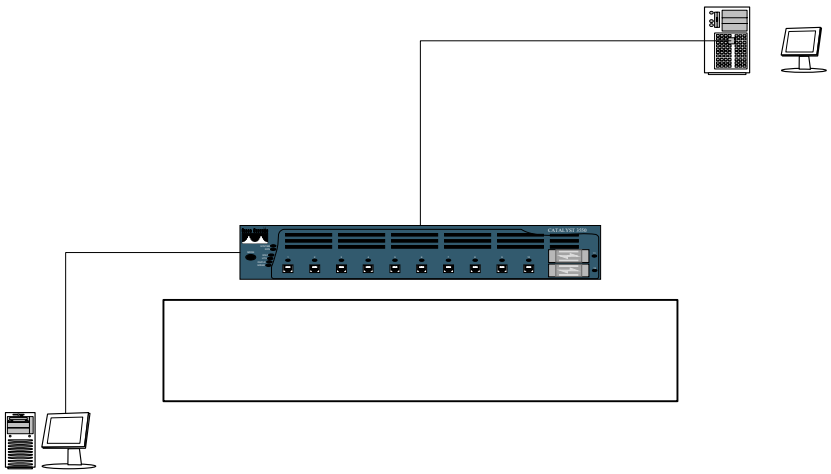
İlk önce anahtarın MAC Adres Tablosu boştur. "A" bilgisayarı Şekil –2.2' deki gibi çerçeveyi anahtara gönderir anahtar ilk durum da yeni açıldığı için boş olan tablosuna "A" bilgisayarından gelen çerçeveden "A" bilgisayarının MAC adresini ve hangi porta bağlı olduğunu bulup tabloya ekliyor. Hedef adres olan "C" makinesinin adresi henüz tabloda olmadığı için anahtar flooding denilen özel bir broadcasting yaparak çerçevenin geldiği port dışında bütün portlara bu çerçeveyi iletir. "C" makinesi dışında buna hiçbir makine cevap vermeyecektir. "C" ' nin cevap vermesi ile "C" ' nin de MAC adresi ve bağlı olduğu port belirlenip MAC Adres Tablosu'na eklenmiştir. Bu şekilde mevcut bütün portların da ki makinelerin adreslerini bularak final MAC Adres Tablosunu oluşturmaktadır. MAC Adres Tablosunda olan kayıtlar için flooding söz konusu değildir direkt olarak veri çerçevesi sahibine gönderilir.

Yerel ağ üzerinde yaygın olan üç tür anahtarlama kullanılmaktadır. Bunlar "cut through", "store and forward", "fragment free" olarak isimlendirilmektedir.

| Hedef Adres | Kaynak Adres | Uzunluk | Data | FCS |
|---|---|---|---|---|
| 123a.1111.0a | 123a.2222.1c | 1514 | 111…0 | 4 Byte |

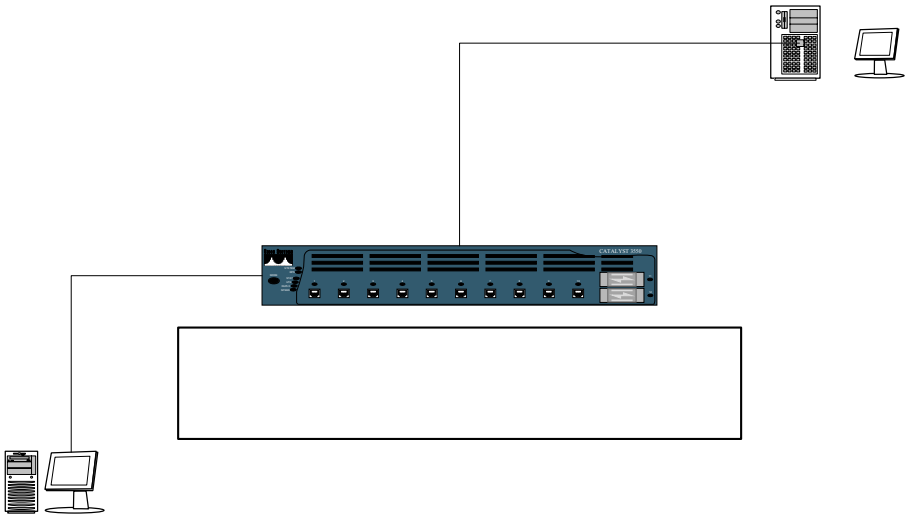
**Şekil 2.4 Örnek Katman – 2 Veri Çerçevesi**

Birincisi en hızlı olanıdır. Bu tür anahtarlamada sadece veri çerçevesi hedef adres kısmına bakılır ve ona göre hedefe iletilir. Herhangi bir hata denetimi ve düzeltme yoktur. Bilgi bütünlüğü ve güvenliğinin önemli olduğu ağlarda uygun değildir.



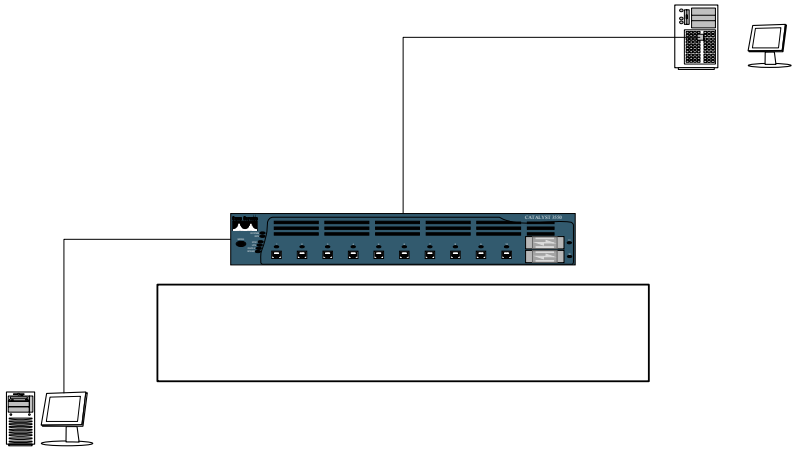
**Şekil 2.5 Cut – Through Anahtarlama Tekniği**

İkincisi en yavaş olanıdır. Önce iletilmek üzere gönderilen bütün veri anahtar cihaz üzerine kopyalanır, hata denetimleri yapılarak bütünlük ve doğruluk kontrol edilir daha sonra hedefine iletilir. Hız gerektiren işlemlerin olduğu ağlarda uygun değildir.



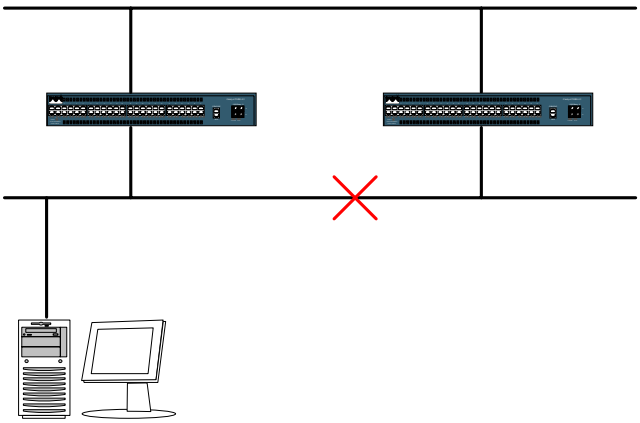
**Şekil 2.6 Store & Forward Anahtarlama Tekniği**

Sonuncusu, ikinciden hızlı ve birinciden daha güvenlidir. Bu anahtarlama yönteminde veri çerçevesinin ilk 64 Byte'ı anahtar üzerine kopyalanır ve hata denetiminden geçirildikten bütünlük ve doğruluğu kontrol edildikten sonra hedefe iletilir. Burada ilk 64 Byte'ın alınmasının sebebi genellikle veri çerçevelerindeki hataların ilk 64 Byte üzerinde olmasıdır.



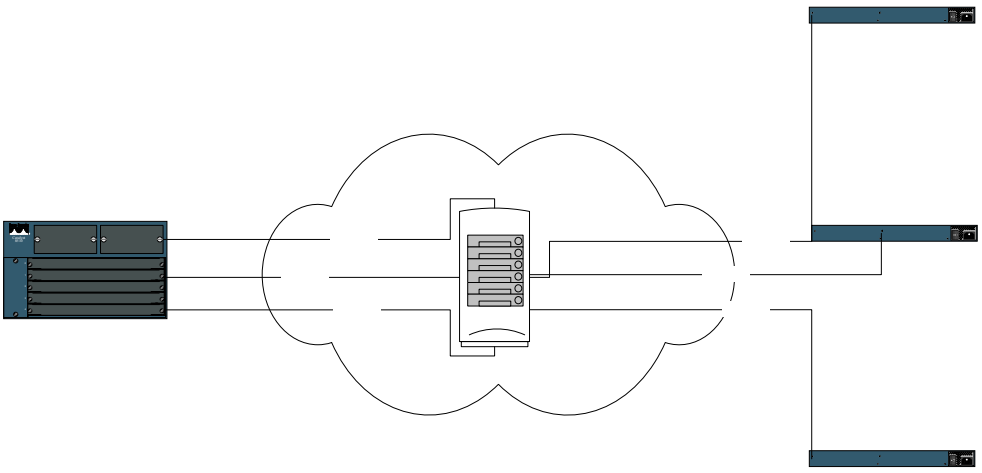
**Şekil 2.7 Fragment Free Anahtarlama Tekniği**

Anahtarlama işleminde dikkat edilmesi gereken MAC Adres Tablosunda bir MAC adresi için birden fazla bilgi olmasıdır. Bu durum da yerel ağda sonsuz döngü oluşur. Bundan korunmak için yerel ağlarda anahtarlar üzerinde STP protokolü kullanılır. Bu protokol birden fazla olan bağlantılardan düşük kapasiteye ait olanı otomatik olarak kapatacaktır.



**Şekil 2.8 STP ve Çalışma Prensibi**

Geniş alan ağlarında hem devre anahtarlama hem de paket anahtarlama tekniklerinin kullanıldığı teknolojiler vardır. Kiralık hat'lar genellikle devre anahtarlama, frame – relay, X.25 gibi teknolojiler ise paket anahtarlama teknolojileridir. Şekil 2.7'de Telekom Altyapısı içerisinde görülen Frame – Relay anahtar yerel alan ağlarındaki anahtarlama için kullanılan MAC adresine benzer DLCI numarası kullanarak paket anahtarlama gerçekleştirir. Merkez ve ofis bağlantıları sabit bir fiziksel devreye sahip değildir, her seferinde farklı yollardan anahtarlanabilir. X.25 de benzer bir yapı içerir. Kiralık hatlarda ise devre anahtarlama söz konusudur. Burada merkez ve ofis arasındaki bağlantı sabit bit yol üzerinden devre anahtarlama yapılarak iletilir.



**Şekil 2.9 Frame – Relay Anahtarlama**

ATM ise hücre anahtarlama tekniğini kullanır ve hem yerel alan ağlarında hem de geniş alan ağlarında kullanılır. Geniş alan ağlarında ATM frame – relay, X.25 veya kiralık hat teknolojileri üzerinden gerçekleştirilir.

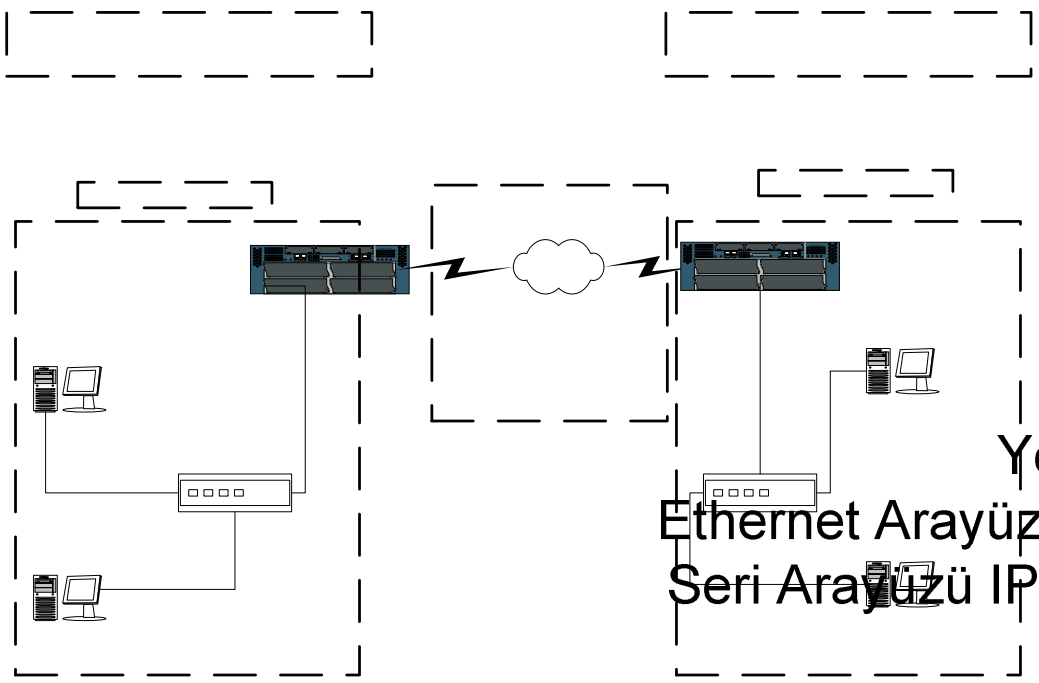
Yukarıda bütün anahtarlama teknolojilerine değinilmeye çalışılmıştır. Görüleceği gibi anahtarlama teknolojilerinde anahtarların yaptığı tek iş veri çerçevelerindeki kaynak ve hedef adresi belirlemek ve iletmektir. Kullanıcı verisi üzerinde herhangi bir işlem yapılmaz.

### 2.1.2 Yönlendirme ve Yönlendiriciler

Yönlendirme işlemi OSI Referans Modelinin üçüncü katmanında gerçekleşir. Genellikle küçük yerel ağlarda kullanılmaz. Ancak geniş alan ağlarında ve kampus ağlarında kullanılır. Yönlendirme işlemi yönlendirilebilen protokollerin kullandığı sanal adreslere göre gerçekleştirilir.

Anahtarlama işlemindeki anahtarların MAC Adres Tablosu'na benzer yönlendiriciler de Yönlendirme Tabloları oluştururlar ve yönlendirme işlemini bu tabloya göre gerçekleştirirler. Yönlendirme Tablolarını yönlendiriciler dinamik yönlendirme protokollerini kullanarak veya direkt olarak kullanıcıların girdiği bilgilerle oluşturur.

En yaygın yönlendirilebilir protokol internetin de temelini oluşturan IP'dir. Ayrıca IPX, SNA, DECNET gibi eski ve çok yaygın olmayan yönlendirilebilir protokoller de vardır.




**Şekil 2.10 Örnek Yönlendirme Ağı**

| Kaynak IP Adresi | Hedef IP Adresi | Protokol | Veri | Segment |
|---|---|---|---|---|
| 172.16.1.2 | 172.16.2.2 | TCP | 10101…1 | … |

Ağ - 1

**Şekil 2.11 Örnek IP Paket Bilgileri**

Şekil – 2.10'daki Ağ – 1'de bulunan bir bilgisayarın üzerinde TCP/IP yapılandırmasının olduğunu ve varsayılan ağ geçidi olarak Yönlendirici – 1'in Ethernet ara yüzüne ait 172.16.1.1 IP adresi belirlenmiş olduğunu düşünülürse. Ağ – 2' deki bir bilgisayara veya internete bağlanmak istediğinde bu isteği varsayılan ağ geçidi olan yönlendirici – 1'e gelir. Yönlendirici – 1 Yönlendirme Tablosuna bakarak veri paketinin gideceği yolu belirler.

| Yönlendirme Tablosu | | |
|---|---|---|
| Hedef Ağ | Çıkış IP'si | Çıkış Arayüzü |
| 172.16.1.0 | | Ethernet 0 |
| 192.168.100.0 | | Serial 0 |
| 172.16.2.0 | 192.168.100.3 | Serial 0 |
| … | … | … |

Tower PC

172.16.1.0 /

**Şekil 2.12 Örnek Yönlendirme Tablosu**

An

Anahtarlama yerel alan içerisinde kalmakta, yerel alan dışında bir yerlerle iletişim kurma işini yönlendiricilerin yapması gerekmektedir. Eğer yerel alan ağlarını birbirine bağlamak gerekirse mutlaka yönlendirme teknolojileri kullanılmalıdır. Düşünüldüğü zaman Internet'te en geniş ağdır milyonlarca yerel ağın birleşmesinden oluşur o zaman internettin temelini de yönlendirme ve yönlendiriciler oluşturmaktadır.

Yönlendirme işlemi yukarıda da değinildiği gibi iki şekilde gerçekleştirilebilir. Birincisi statik olarak veri paketlerinin gideceği yolun belirlenmesi ikincisi ise veri paketinin gideceği yolun dinamik yönlendirme protokolleri aracılığıyla belirlenmesidir. Dinamik yönlendirme protokolleri yönlendirilebilen protokoller üzerinde değişik yönlendirme algoritmaları kullanarak veri paketlerinin rotalarını bulmalarını sağlar.

En sık kullanılan yönlendirme protokolleri RIP, OSPF, IS – IS, vs… dir. Yönlendirme protokolleri çalışma mantıklarına göre uzaklık vektörü ve hat durumu olmak üzere temel iki kategori içerisinde bulunur.

Yönlendirme protokolleri Yönlendirici Tablolarının değişimi esasına dayanmaktadır. Daha basit bir şekilde her yönlendirici bildiği yolları bir birlerine söyler. Bu söyleme şekilleri, zamanları, vs. gibi durumlar yönlendirme protokollerini oluşturur.

Uzaklık vektörü temelli yönlendirme protokolleri ( örneğin RIP ) Yönlendirme Tablolarını belirli aralıklarla günceller ve değişirler. Hat durumu temelli yönlendirme protokolleri ( örneğin OSPF ) bir bağlantının durumunun değişmesi halinde Yönlendirme Tablolarını günceller ve değişirler.

Yönlendirme işleminde anahtarlama işleminin tersine olarak bir bağlantı için birden fazla yol aynı anda kullanılabilir. Anahtarlamanın tersine bu bağlantılardan birisi kapatılmaz burada eğer dinamik yönlendirme protokolleri kullanılıyorsa yük

paylaşımı otomatik olarak veya belirlenen oranlarda yapılabilir. Statik yönlendirmede de el ile bir hedef için birden fazla yol belirlenebilir.

Yönlendirme işlemlerinde de dikkat edilecek olursa kullanıcı verileri üzerinde herhangi bir değişiklik söz konusu değildir. Sadece yönlendirici bir paketi başka bir ağa gönderirken kaynak adres yerine kendi adresini yazar ve dönüş paketinde tekrar düzenler bunun dışında veriler tamamen sabittir.

## 2.2 Aktif Ağlar ve Genel Kavramları

Aktif ağlar üzerlerinden geçen veri paketleri üzerinde uygulamalara özel hesaplamalar yapabilen düğümlerden oluşan ağdır.

Bir ağ iki şekilde aktif olabilir;

Düğüm üzerinden geçen bütün paketlere bir kod ekler ( kod eklendikten sonra paket + kod yapısına **kapsül** denir.) ve aktif düğüme iletir. Bu yaklaşıma **bütünleşik** veya **in-band aktif ağlar** yaklaşımı denir.

Düğüm üzerine kullanıcı kendi işlemleri yapılacağı zaman uygulanmasını istediği hesaplama algoritmalarını girebilir ve kendi paketleri iletilirken bu hesaplama algoritmaları uygulanabilir. Burada ki hesaplama algoritmaları, özel bir sıkıştırma algoritması, videotranscoding algoritması, vs. olabilir. Bu tür yaklaşıma da **ayrık** veya **out-band aktif ağlar** yaklaşımı denir.

Aktif ağlardaki yönlendirme ve anahtarlama işlemleri uygulama seviyesinde gerçekleştirilmektedir. Mevcut ağ teknolojilerinde olduğu ayrı ayrı anahtar yönlendirici güvenlik duvarı gibi cihazlar bulunmakta idi, ancak aktif ağlarda bütün bu cihazların yaptığı işleri aktif düğüm gerçekleştirebilmektedir. Çünkü aktif düğümler uygulama seviyesinde çalışan cihazlardır.

Bu nedenle yönlendirme ve anahtarlama işlemleri alt katmanlarda aynen mevcut ağ teknolojilerinde olduğu gibi gerçekleşir. Ancak uygulama katmanında kod

çalıştırıldıktan sonra yapılan programlamaya göre yönlendirme ve anahtarlama bilgileri yeniden düzenlenebilir ve yeni paketler oluşturulabilir.

Aktif bir anahtarı mevcut altyapılarda kullanılan bir anahtardan ayıran en önemli fark bütün özelliklerinin anahtarlama algoritmaları dahil yeniden programlanabilmesidir. Bunu bir örnek ile açıklayabiliriz hatta bu örnek bu konu ile ilgili ileride yapılacak bir çalışmada olabilir. Aktif olmayan anahtarımızı düşünelim bu anahtar üzerinde en çok kullanılan üç adet anahtarlama algoritması mevcuttu ve bunlardan birisi seçilerek paketler anahtarlanıyordu. Bölüm 2.1.1 de ayrıntılı olarak anlatıldığı gibi cut-through hızlı, store & forward güvenli, fragment free cut-through'dan güvenli store & forward'dan hızlı idi görüldüğü gibi hangi anahtarlama yöntemi seçilirse seçilsin bir şeylerden kayıp vardır.

Aktif anahtarda her bir bağlantı için ilk başta fragment free ile anahtarlama başlatılıp belirli bir süre bağlantının kalitesi ile ilgili istatistiksel veriler tutulup bu veriler üzerinde karar verme algoritmaları kullanılarak bağlantının kaliteli, normal, kötü gibi etiketlenmesi ve en uygun anahtarlama algoritmasının seçilmesi sağlanabilir. Bağlantıdaki bozukluk geçici olabileceği için tutulan istatistikler bilgiler ve karar verme algoritmaları belirli aralıklarla tekrarlanabilir. Normal bir anahtarda her kullanıcı ve bağlantı için bir anahtarlama algoritması vardır. Aktif anahtar üzerinde her bir bağlantı için dinamik olarak değişen ve en uygun olan anahtarlama algoritması kullanılabilir.

Yönlendiricilerde de durum farklı değildir. Bir yönlendirme algoritması belirlenir ve statik olarak çalışır. Örneğin OSPF gelişmiş ve güçlü bir yönlendirme algoritmasıdır ancak çok aşırı kaynak kullanır. RIP, OSPF kadar gelişmiş bir yönlendirme protokolü değildir. Ancak daha az kaynak kullanır. Aktif bir yönlendiricide de en uygun yönlendirme algoritması dinamik olarak seçilebilir. Yine anahtarlarda olduğu gibi yönlendiricilerde de yönlendirme algoritmaları bütün cihaz içerisinde veya belirlenen ara yüzlerdeki bütün bağlantılar için geçerlidir. Aktif bir yönlendiricide ise her bağlantı için dinamik olarak değişebilecek farklı algoritmalar kullanılabilir.

## 2.3 Geleneksel Ağ Ve Aktif Ağların Karşılaştırılması

Geleneksel veri ağlarında anahtarlarda veya yönlendiricilerde çok fazla işlemci gücü, hafıza veya depolama alanı gerekmez çünkü yapılan iş birkaç başlık işleminden ibarettir. Ancak aktif ağlarda bir aktif düğüm üzerinde bütün aktif kodlar üzerine kopyalanacağı için ve üzerinde bir işletim sistemi, çalıştırma ortamı bulunacağı için daha fazla disk alanına, program derleyeceği ve çalıştıracağı için daha fazla hafıza ve CPU gücüne gereksinim duyulacaktır. Yani günümüz anahtar ve yönlendiricilerinin aktif ağ yönlendirici ve anahtarları olabilmeleri için donanım olarak baya güçlendirilmeleri gerekmektedir.

Geleneksel veri ağlarında bir anahtar veya yönlendirici üzerinde hiçbir özelleştirme yapamazsınız, tamamen üreticinin çizdiği sınırlarda kalmak durumundasınız. Ancak aktif ağlar kullanıcı ile ağ düğümleri arasındaki bu sınırı kaldırıp ağ düğümlerinin özel amaçlara yönelik programlanabilmelerini sağlamaktadır.

Ağ düğümleri üzerinde programlama yapabilmek ağ teknolojilerinin hızlı gelişmesine de olanak sağlayacaktır. Yeni teknolojiler geliştirme işi sadece üreticilerdeki insan gücüyle sınırlı kalmayıp daha geniş bir geliştirici kitlesine yayılacağı için yeni teknolojilerin geliştirilme hızları ve sayıları artacaktır.

Burada aktif ağ düğümlerinin kullanıcılara işletim sistemleri gibi bir programlama ara yüzü sunması iyidir. Ancak bu noktada hangi programlama dilinin kullanılacağı hangi çalıştırma ortamının kullanılacağı gibi standartların belirli olması gerekmektedir. Aksi halde her şey çok karmaşık olur ve Pandora'nın Kutusuna döner işler ki şu an öyle olduğu söylenebilir.

Güvenlik zorluklarının yanında temel bir standardın oturtulmamsı aktif ağların üreticiler tarafından hemen kabul edilip aktif ağ ürünlerini piyasaya sürmelerini geciktirmektedir.

## 2.4 Aktif Ağ Altyapısı Ve Güvenlik Zorlukları

Bir aktif ağ altyapısında üç temel bileşen bulunur:

### 2.4.1 Çalıştırma Ortamı

Aktif kodun çalıştırıldığı yerdir. Çalıştırma ortamı ana düğümün kaynaklarına bir güvelik – kontrol şeması dahilinde erişimi sağlar. Birden fazla çalıştırma ortamı bir arada bulunabilirler. Örneğin ANTS, ALIEN vs...

### 2.4.2 Aktif Kod ve Özellikleri

Aktif bir düğümün çalıştırma ortamında çalıştırılabilen koddur. Aktif kod düğüm üzerinde kurulmuş genel amaçlı birçok dille (JAVA, C, vs...) yazılmış olabilir. Çalıştırma ortamı tarafından çalıştırılan kod kullanıcının tercihleri doğrultusunda aktif düğümü programlar.

Aktif kod değişik karakteristiklere sahip olabilir. Aktif kod:

**Konumsuz (Stateless) :** Aktif kod basit olarak bir düğümden bir düğüme transfer edilen ve orada başlangıçtan itibaren her zaman çalışan aptal bir programdır.

**Konumlu (Statefulll) :** Burada aktif kod ağ üzerinde gezerken konum bilgisini tutar ve bu gezinti sırasında konumuna göre karar verebilir. Bu aktif koda çevresel durumlara göre bir düğümde çalışmasını durdurma bir düğümde çalışmasını devam ettirme kararı ile dinamik bir yapı kazandırır.

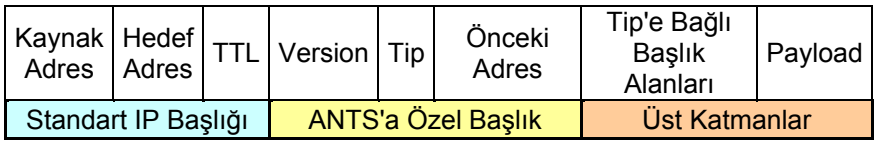
**Sabit (Stationary) :** Burada aktif kod bir düğüm için kalıcı olarak durur özel bir ara yüz üzerinden istek olduğunda aktif düğümü programlar.

**Taşınabilir (Mobile) :** Aktif kod içsel amaçları ve istekleri doğrultusunda düğümler arasında serbestçe dolaşır.

Konumlu ve taşınabilir özellikler birleştirildiği zaman Aktif kodumuz Mobile agent karakterini almış olur. Mobile agent ağ üzerinde dolaşır ve dinamik olarak değiştirilebilen amaçları doğrultusunda kendi kendine çeşitli araçları çalıştırır. Dahası mobile agent'ın akıllı olması Aktif ağlar tarafından daha da ilgi çekici olmasını sağlamaktadır.

### 2.4.3 Aktif Kod Taşıyıcı

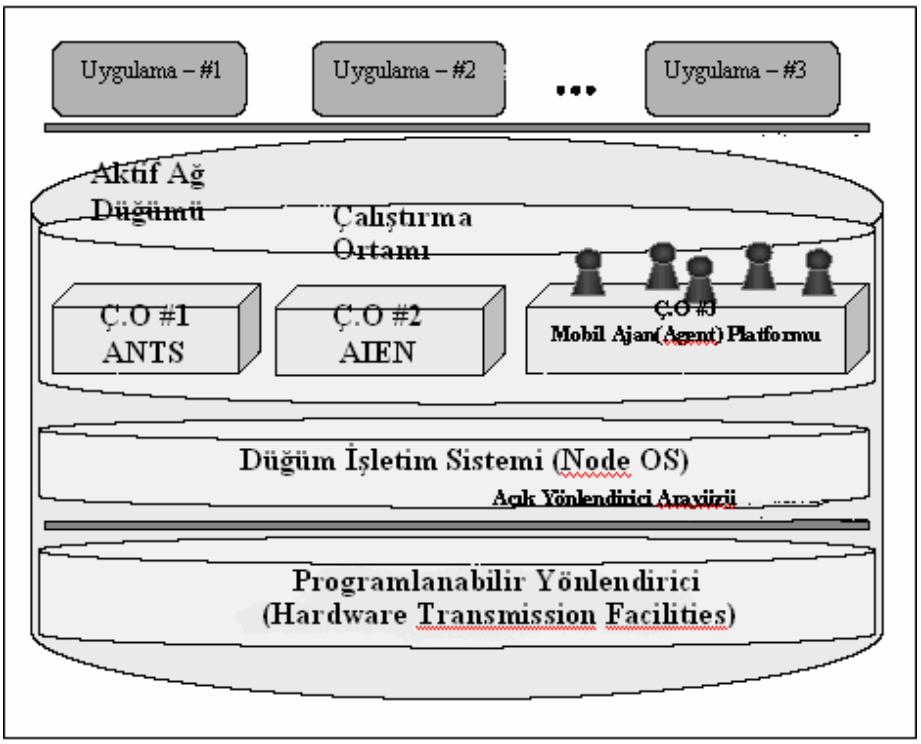
Aktif kod kaynak düğümden hedef düğüme taşınır. Bu taşıma işlemi giriş bölümünde değindiğimiz gibi bütünleşik sistemde kapsül sistemi ile, ayrık sistemde ise düğüm üzerine yerleştirilmiş uygulama özel kullanıcı algoritmaları ile gerçeklenmektedir.




**Şekil 2.13 Aktif Kod Taşıyıcı Olarak Kapsül Yapısı**

### 2.4.4 Çoklu Çalıştırma Ortamlı Aktif Ağ Mimarisi

Çoklu çalıştırma ortamlı aktif düğüm mimarisi aşağıdaki Şekil – 2.14'de gösterilmiştir. Dikkat edilmesi gereken birden çok çalıştırma ortamının bir Düğüm işletim sistemi üzerinde olabileceğidir. Böyle bir mimari aşağıdaki ana bileşenlerden meydana gelmektedir.




**Şekil 2.14 Çoklu Çalıştırma Ortamlı Aktif Düğüm Mimarisi**

### 2.4.4.1 Programlanabilir Yönlendirici veya Anahtar

Yönlendirici kaynaklarını dinamik olarak programlamayabilmek için API üzerinden erişilebilen yönlendiricidir. Açık düğüm ara yüzü CPU, hafıza gibi bilgisayar kaynaklarından bant genişliği, buffer gibi iletim kaynaklarına kadar değişen yönlendirme kaynaklarının yapısını temsil eder.( Programlanabilmelerini sağlar. ) . Bu aslında çoğu aktif ağ projesi için Linux, Unix temelli bir işletim sistemini + donanımın bize sunduklarıdır.

### 2.4.4.2 Düğüm İşletim Sistemi

Aktif ağ üzerindeki her bir düğüm üzerinde çalışan işletim sistemidir. Moab, JNodeOS gibi düğüm işletim sistemleri vardır. Bunlarda kullanılan genelde JVM tarzı bir sanal makinedir. Bu sanal makine alttaki işletim sistemleri ile iletişim kurarak donanım kaynaklarını aktif çalıştırma ortamlarına ve kodlara sunar.

#### 2.4.4.3 Çalıştırma Ortamı

Düğüm işletim sisteminin üstündedir. İşletim sistemi tarafından verilen servisleri kullanır.

Aktif ağ düğümü, Düğüm İşletim Sistemi, Çalıştırma Ortamları ve aktif uygulamalar olmak üzere bölümlendirilir. Bu mimaride birden fazla çalıştırma ortamı aynı anda bulunabilir ve bir aktif kodu ortak olarak işleyebilecekleri gibi ayrı ayrı da işleyebilirler. Her çalıştırma ortamı bir programlama ara yüzü veya sanal makine sunarak üçüncü parti kodlar ile programlanabilir ve kontrol edilebilir. Düğüm İşletim Sistemi, düğümün kaynaklarının yönetimini yapar. Uygulamalar çalıştırma ortamının sunduğu bütün servislere erişebilirler. Genellikle bir uygulama bir çalıştırma ortamı üzerinde çalıştırılır. Ancak ileriye dönük bir öngörü ile çeşitli servisleri çeşitli çalıştırma ortamlarından alan uygulamalar düşünülebilir.

## 2.5 Aktif Ağlardaki Tehditler Ve Çözüm Önerileri

Mevcut ağlarda kaynaklar sadece gelen paketlerin geçici olarak bir kısmının veya tamamının hafızada tutulması ve CPU'nun paketin gideceği yönü belirlemesi için kullanılıyor. Bu tür altyapılar da aradaki bir düğüm için katı kaynak yönetimi kritik değildir.

Fakat bir aktif ağ uygulaması çok daha fazla kaynak tüketmenin yanında hızlıdır da. Eğer kaynak yönetimi yoksa veya iyi değilse DOS ( Denial Of Service ) atağı kolaylıkla gerçekleştirilebilir. Ayrıca aktif ağlarda aktif kodun sahibi, düğüme yükleyen, düğüm donanımının sahibi, çalıştırma ortamının sahibi ve hatta çalıştırılacak alanın sahibi gibi farklı güvenlik politikaları gerektiren farklı durumlar söz konusudur[Gabrijelĉiĉ, et all, 2004]. Böyle heterojen ağlarda güvenlik son derece önemlidir.

Bir aktif ağda, aktif paket, aktif düğümü, ağ kaynaklarını ve diğer aktif paketleri birçok yolla kötüye kullanabilir. Bunun yanında aktif düğüm ve çalıştırma ortamı da aktif kodu kötüye kullanabilir.

Genel olarak aktif ağlardaki güvenlik sorunları:

- Aktif Kod tarafından, çalıştırma alanının suiistimal edilmesi,

- Aktif kodun başka bir aktif kod tarafından suiistimal edilmesi,

- Aktif kodun çalıştırma ortamı tarafından suiistimal edilmesi,

- Ağ altyapının veya içerisinde bulunulan teknolojinin eksikleri sonucu Çalıştırma ortamı ve aktif ağ ile ilgili suiistimaller. ( Unix sisteminin güvenlik eksiklikleri gibi)

## 2.6 Aktif Ağlardaki Güvenlik Zorlukları

**Hasar :** Bir aktif paket bir düğümün kaynaklarını veya servislerini, tekrar düzenleyerek, değiştirerek veya hafızadan silerek yok edebilir veya değiştirebilir. Veya bir düğüm aktif kodu işini bitirmeden önce silebilir. Son olarak aktif paketler aynı hesaplama ortamını paylaştıkları için bir birlerine zarar verebilirler.

**DoS ( Denial Of Service – Servis Durdurma) :** Bir aktif paket sürekli ağ bağlantısı kurarak veya çok büyük CPU kullanımı gerektiren kodlar kullanarak kaynak veya servislerin aşırı yüklenmelerine sebep olabilir. Bu durumda düğüm düzgün olarak fonksiyonlarına devam edemez ve diğer aktif paketler çalıştırılamaz veya iletilemez.

**Hırsızlık :** Aktif paket bir düğümün özel bilgilerine ulaşıp bunları çalabilir. Diğer taraftan bir aktif paket içindeki özel bilgiler, ziyaret ettiği düğümler tarafından çalınabilir. Aktif paket içerisindeki bilgiler şifrelense bile tamamen güvenli değildir, çünkü çalıştırılmak için çözülmesi gerekir.

**Bütünleşik Atak :** Mevcut en büyük tehdit yukarıda sayılan saldırı tekniklerinin birlikte kullanıldığı karmaşık ataklardır.

## 2.7 Genel Saldırı Tipleri

### 2.7.1 Çalıştırma Ortamının Aktif Kod Tarafından Suiistimali

Kötü niyetli bir aktif kod çalıştırma ortamında güvenlik zaaflarına yol açacak aşağıdaki ataklara yol açabilir.

**Maskeleme: Bir** aktif kod kendi kimliğini gizleyerek başka bir kod gibi davranabilir. Böylece o kod ile ilgili yetkilere sahip olur.

**Servis Durdurma (Denial Of Service) :** Kötü niyetli aktif kod sistem kaynaklarını aşırı yükleyerek sistemin performansının düşmesine hatta servis dışı kalmasına sebep olabilir.

**Yetkisiz Giriş: Aktif** kod yazımında dil ile ilgili veya kod ile ilgili eksiklikleri kullanarak aktif kod yetkilendirme adımını atlayarak çalıştırma ortamına zarar verebilir.

**Karmaşık Ataklar: Bir** den çok aktif kod birlikte koordineli ve planlı bir şekilde aktif düğüme atak yapabilir bu tür atakların belirlenmesi ve bu tür ataklardan korunması oldukça zordur.

### 2.7.2 Aktif Kodun Başka Bir Aktif Kod Tarafında Suiistimali

Kötü niyetli bir aktif kod bir aktif koda çok değişik türde ataklar gerçekleştirebilir.

**Masquerading :** Bir aktif kod diğer aktif kodu ile iletişiminde kimliğini tamamen gizleyerek veya başka kimliğe girerek diğer aktif koda zarar verebilir.

**Servis Durdurma (Denial Of Service) :** Bir aktif kod diğer aktif koda spam mesajlar göndererek cpu kullanımı disk kullanımı gibi kaynakların yükünü artırarak servisi çalışmaz hale getirebilir.

**Yetkisiz Erişim** **:** Yetkisiz bir şekilde kötü niyetli bir aktif kod diğer aktif kodun görevlerini durumunu değiştirebilir aynı zamanda içerdiği bilgileri de çalabilir.

### 2.7.3 Aktif Kodun Çalıştırma Ortamı Tarafından Suiistimali

Bir aktif kodun çalıştırılması ile ilgili bütün kontrol Çalıştırma ortamının elindedir. Bu yüzden aşağıdaki saldırıları gerçekleştirebilir.

**Masquerading** **:** Kendisini başka bir çalıştırma ortamı gibi göstererek aktif kodun durum bilgisini kötüye kullanabilir.

**Denial Of Service** **:** Kötü niyetli bir çalıştırma ortamı aktif bir kodun istekleri sürekli erteler veya görmezden gelebilir hatta aktif kodu silebilir.

**Yetkisiz ve Gizli Dinleme** **:** Aktif kod ile ilgili bütün trafiği ve aktif kodun yapısını monitör edebilir. Eğer veriler şifrelenmemiş ise bütün verilere ulaşıp elde edebilir.

**Klonlama** **:** Aktif kodu klonlayarak klonladığı aktif kod üzerinden aktif kod ile ilgili bilgilere ulaşabilir.

### 2.7.4 Altyapının Aktif Kod ve Çalıştırma Ortamını Suiistimali

Aktif kod ağ üzerinde düğümden düğüme dolaşırken çeşitli tehditler söz konusudur. Dışarıdan bir saldırgan yukarıda saydığımız bütün atak türlerini ( masq, Denial Of Service vs…) içeren bir atak yapabilir. Bir senaryo üretecek olursak Çalıştırma ortamı bir unix düğüm üzerinde olsun ve kullanıcı güvenlik ile ilgili diskte duran dosyayı yanlış ayarlamış olabilir böylece dışarıdan bir saldırgan bu dosyaya erişebilir ve dolayısı ile her şeye erişebilir.

## 2.8 Aktif Kodun Güvenlik Gereksinimleri ve Çözümler

Aktif ağların amacı kullanıcı veya herhangi bir uygulamanın ağa kod ekleyebilmesi ve özel ihtiyaçları doğrultusunda davranışı sağlamasına olanak vermektir.

Bu da doğal olarak güvenlik sorunu ortaya çıkarmaktadır ki aktif ağların araştırma alanlarından çıkıp gerçek dünyaya girebilmesi ve yaygınlaşabilmesi için bu problemlerin ortadan kaldırılması gerekmektedir.

Maalesef güvelik Boolean değişkenler gibi, basit test edilebilir bir özellik değildir. Güvenlik bir çok parametre göz önüne alınarak tasarlanmalıdır herhangi basit bir güvenlik açığı tüm sistemi tehlikeye atar. Aktif ağların genel güvenlik gereksinimleri şunlardır.

**Gizlilik ve Güvenirlik :**

Aktif kod içinde ki gizli veri gizli kalmalıdır. Veriye sadece yetkili kullanıcılar ulaşabilmelidir. Yetkisiz kullanıcılar bilgilere ulaşamamalıdır. Bu aktif kodun iletimi sırasında dolaştığı bütün düğümler için geçerli olmalıdır.

**Bütünlük :**

Aktif kod ağ üzerinde dolaşırken yetkisiz erişim veya kazara oluşacak değişikliklerden korunmalıdır.

**Bulunurluk :**

Aktif ağ kötü niyetli olarak yetkili kullanıcıların isteklerini reddetmemelidir. Dahası kaynak yönetimi, birlikte kullanımın kontrolü, deadlock yönetimi, çoklu – erişim, hata durumlarının taranması ve kurtarılması, sonsuz döngü sorunları giderilmelidir.

**Yetkilendirme :**

Yetkilendirme güvelik politikasının ilk adımını oluşturmaktadır. Heterojen ağlarda özel kodları düğümlere yetkili kullanıcılar tarafından yerleştirilmesi kritik ve önemlidir.

### 2.8.1 Aktif Düğüm Kaynaklarının Aktif Paketlere Karşı Korunması

Bu konuda ki temel nokta Aktif paketlerin bir birlerine veya aktif düğümlere müdahale etmelerini engellemektir. Bu durumda en iyi çözüm bir güvenlik yöneticisidir. Burada güvenlik yönetici implementasyonunda birçok geleneksel teknik kullanılabilir.

Bu teknikler;

Kriptografik teknikler, Aktif düğüm ve Aktif paketin yetkilendirilmesi teknikleri, güvelik olayları ile ilgili akıllı filtreler ve kayıtlar tutarak sistem kaynaklarına erişim ve süreçleri bir birlerinde ayıran teknikler içermelidir.

Genel Yöntemler şöyle sıralanabilir :

**İmzalı Kod :** Bildiğimiz dijital imza uygulaması kullanılarak gelen aktif kodun kaynağı kesin olarak belirlenebilir.

**Konum Değeri :** Aktif paket içerisinde taşınan bir aktif kod düğümler arasında gezinirken konum bilgisini tutuyorsa. Buradaki durum değerine göre özel bir çalıştırma ortamına belirli yetkiler verilmesine aktif kod karar verebilir. Buradaki durum değeri sabit yapılarak bunun değiştirilmesi engellenebilir.

**Güvenli Kod Yorumlanması :** Bu aktif kodun yazıldığı dil ile ilgili bir özelliktir. Bu konuda kod güveliği diller arasında farklılık göstermektedir. Bu doğrultuda bir güvenlik önlemi alınarak çok fazla güvenli olmayan yorumlayıcılar kullanılmalıdır.

Burada ayrıca hata izolasyonu, yol geçmişi, kaynak yönetimi gibi teknikler de kullanılmaktadır.

### 2.8.2 Aktif Paketlerin Korunması

Aktif paketlerin korunması ile ilgili iki önemli yöntem **Fault Tolerance** (Hata Toleransı) teknikleri ve **encryption** ( Şifreleme) tekniklerdir.

**Şifreleme** aktif paketlerin düz metin kod ve data içermemesidir. Şifreleme genellikle kod ve data çalıştırılmadan transit geçiyorsa kullanılır.

**Hata Tolerans** teknikleri; Replication, persistence ve redirection dır.

**Replication,** aktif paketler her düğümde geçici bir süre için kopyalanır bir sorun olduğu durumda ilgili düğüm tarafından tekrar alınmak üzere.

**Persistence,** ise geçici kopyalamaya karşın düğümde herhangi bir çökme ve hataya karşı kalıcı olarak aktif düğüme kopyalanır.

**Redirection,** Aktif paket kendi ön tanımlı yolunda ( default route ) hata varsa alternatif bir yol arayabilir. Replication ve Persistence ağ paketlerinin büyük bir çoğunluğu için kabul edilebilir değildir, çünkü bunlar çok fazla hafıza ve bandgenişliği tüketir. Bu nedenle sadece çok önemli aktif paketler için kabul edilebilir teknikler olabilirler. Örneğin; bütün aktif düğümlere yüklenecek yeni yönlendirme algoritmaları gibi.

Redirection ve şifreleme daha makul CPU kullanımı gerektirdiği ve hafıza gereksinimi çok fazla olmadığı için her aktif paket için uygulanabilirliği mümkün gibi gözüküyor.

### 2.8.3Güvenli Bir Aktif Ağ Altyapısının Özellikleri

Şimdi yukarıda bahsettiğimiz güvenlik problemlerini çözme tekniklerinin uygulandığı bir çözüm üretebilirsek aktif bir kod düğüme geldiği zaman sistem şunları yapmalıdır:

- Paket güven belgesi ( kimlik, anahtar ) için yetkilendirmeyi kabul etmeli yani paketin geçerli bir güven belgesi olmalı.
- Gönderen ağ elemanının kimlik bilgilerinin tanımlanmış olmalı.
- Gönderen kullanıcı kimliği tanımlanmış olmalı.
- Belirlenen kimlikler ve güven belgeleri dikkate alınarak uygun kaynaklara erişim izni verilmeli.
- Yetkilendirme ve güvenlik kuralları dahilinde çalıştırmanın kabulü.
- Çalıştırma anında sistem kaynaklarının kontrol ve monitör edilmesi.
- Eğer paket o düğümde çalışmayacak ise ve şifrelenmesi gerekiyorsa şifrele ve transit geçir.
- Eğer paket uygun tanımlara ve kimliğe sahip değilse, kısıtlı bir alanda çalışması kabul edilebilir veya çalıştırılması kabul edilmeyebilir.

# 3 AKTİF AĞLAR İÇİN IPSEC İLE GÜVEN YÖNETİMİ MODELİ

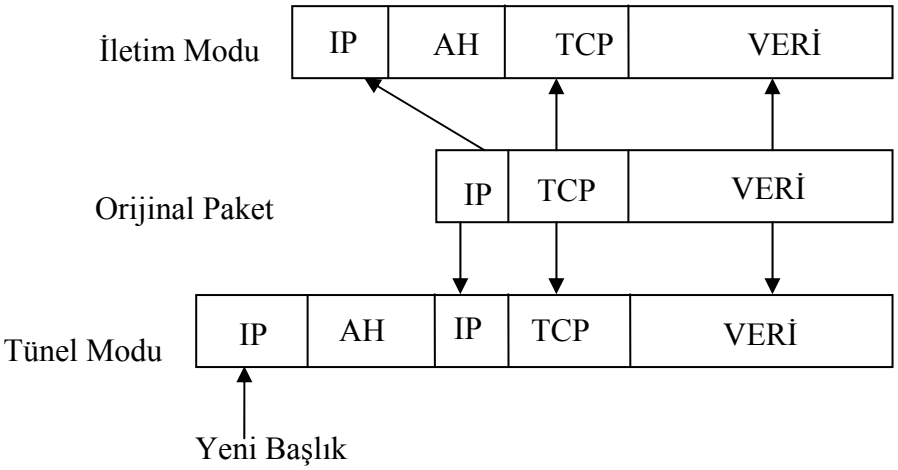
Buraya kadar aktif ağlar üzerindeki mevcut güvenlik eksikliklerini ve bu konunun giderilmesi ile ilgili öneriler üzerinde duruldu. Şimdi ise önerilen GYTE Aktif ağ test laboratuarı üzerinde IPSEC kullanarak aktif düğüm ve kod için güvenlik sistemi modeli incelenecektir.

Burada uyguladığımız model OSI referans modelinin üçüncü (ağ) seviyesinde bir güvenlik önlemidir. Güvenlik önlemlerini ne kadar alt seviyede alınırsa, o kadar etkili olacaktır. Buna örnek vermek gerekirse en etkin şifrelemeyi OSI Referans modelinin ilk (donanım) katmanında yer alan donanımsal şifreleyiciler sağlar. Bu nedenlerle bu çalışmada IPSEC kullanılmıştır.

## 3.1 IPSEC Protokolü ve Modelin Uygulanması

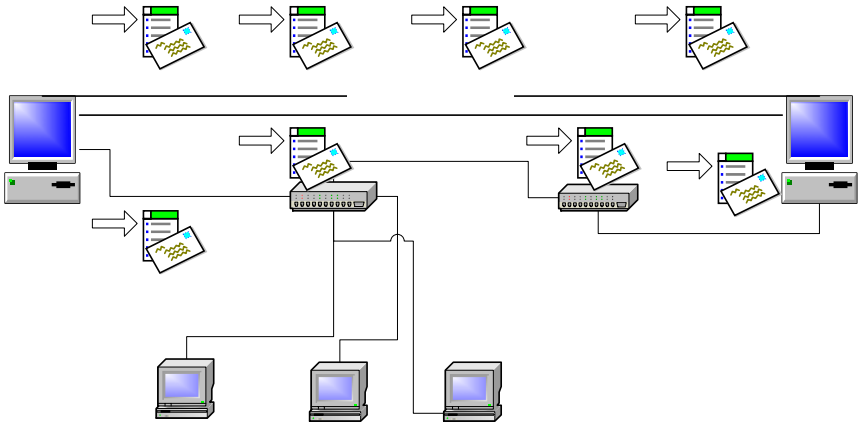
IPSEC IP protokol yığını üzerinde şifreleme ve yetkilendirme yapmayı destekleyen bir servistir. IPSEC bu seviyede çalışarak IP trafiği üzerinden taşınan bütün trafiği koruma altına alabilir. Diğer şifreleme teknikleri daha üst katmanda genelde özel uygulamalar için tasarlanmıştır. Örneğin; PGP mail, SSH uzak bağlantı, SSL web uygulamaları için kullanılmak üzere tasarlanmıştır. IP seviyesinde bir güvenlik sistemi kurarak zaten bunun üzerindeki seviyelerde ki bütün iletimi güvenli hale getirilmiş olunur.

IPSEC, kurulan bağlantının şifrelenmesi ve yetkilendirilmesi için ESP, bağlantının kurulması ve anahtar değişimi işlemi için ise IKE, paket yetkilendirme için AH protokollerini kullanır. IPSEC ile bütün IP datagramını veya sadece üst-seviye protokolleri koruma altına alınabilir. Bu IPSEC' in iki kullanım şeklidir. Birinci kullanım şekline **tünel modu** ikincisine ise **taşıma modu** denir. Tünel modunda IPSEC protokolü kullanılarak bütün IP datagramı yeni bir IP datagramına encapsule edilir. Taşıma modun da ise sadece IP datagramına IPSEC başlığı eklenir.




**Şekil 3.1 IPSEC IP Datagram Yapısı**

Önerilen model için her iki modda kullanılmıştır. Tünel modu daha çok hibrit bir yapı içerisinde ki iki aktif düğümün güvenli bir şekilde iletişimi için kullanılmıştır.



**Şekil 3.2 Aktif Ağ IPSEC Tünel Modu (VPN)**

Şekil – 3.2'den de anlaşılacağı gibi eğer IPSEC tüneli kurmazsak iki Aktif düğüm arasında aktif kodumuzu korumamız paylaşılmış ortamların kullanılması sonucu zorlaşmaktadır. Ama IPSEC'i böyle hibrit bir yapıda tünel modda kullanırsak

daha önce bahsetmiş olduğumuz çoğu zorluğu IPSEC'in şifreleme ve yetkilendirme servisleri sayesinde aşmış oluruz.

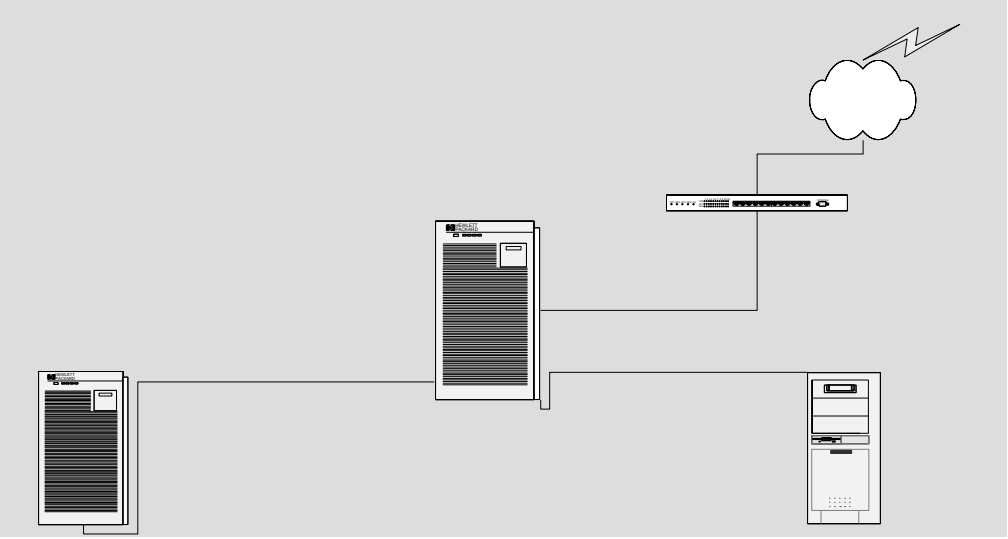
Taşıma modunda ise aktif düğümlerden oluşan yalıtılmış bir durum için kullandık. Burada mevcut IP datagramına sadece IPSEC' in başlığı ekleniyor yani şekil – 3.1' deki AH ( Authentication Header ).

**Şekil 3.3 Aktif Ağ IPSEC Taşıma Modu**

Burada da IPSEC kullanarak şifreleme yapılmış ve düğüm – 1' den düğüm – 3' e gidecek paket düğüm – 2 ' den geçeceği için düğüm – 2' nin bu paketleri suistimal etmesi önlenmiş olmuştur.

Her iki mod'da da önerilen modelde simetrik anahtar mimarisi kullanılmıştır. Yani her düğümün bir adet anahtarı vardır. Kapsüller gönderilirken bu anahtarla şifrelenip açılırken tekrar bu anahtar ile açılmaktadır. Önerilen model için düğüm – 1 ile düğüm – 3 arasında gidecek trafik transport modun da ara düğümden geçeceği için ara düğüm, düğüm – 1 ve düğüm – 3 için güvenilen bir düğümdür. Çünkü her iki düğümünde anahtarları düğüm – 3 de bulunmaktadır. Bu durumda düğüm – 3 güvenilen düğüm olmak zorundadır. Önerilen model daha geniş bir yapıya uygulanırsa anahtar dağıtımı yapılması gerekmektedir. Bunun için IPSEC'in IKE protokolü kullanılarak iletişimden önce anahtar dağıtımı gerçekleştirilmektedir. Diğer bir yöntem ise bir güvenlik merkezi SA ( Security Authority ) kurulması ve gerekli anahtarların bu merkez tarafından sağlanmasıdır. Her iki durumda da anahtarların geçerlilik süreleri makul seviyelerde ayarlanarak güvenlik gereksinimleri sağlanmış olur.

## 3.2 Testbed İle İlgili Ayrıntılı Bilgi



**Şekil 3.4 GYTE Aktif Ağ Test Ortamı**

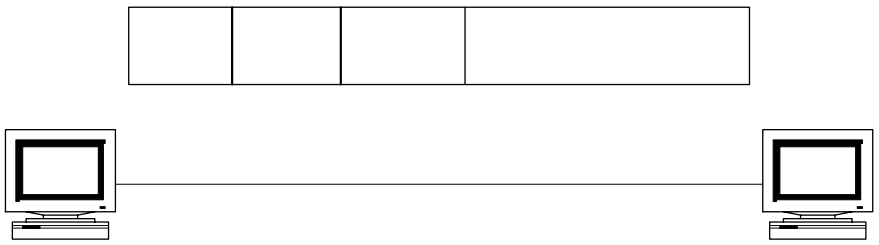
Şekilde 3.4'de görülen test laboratuarını biraz açıklarsak; Düğüm **– 2** HP Workstation xv8000 serisi 1 GB. Ram'e, 3 adet 1Gbit, 1 adet 100Mbit Ethernet kartına sahip. **Düğüm – 3** OEM P – II 300Mhz işlemci. 256Mb. Ram'e, 1 adet 100Mbit Ethernet kartına sahip. **Düğüm – 1** HP Workstation xv5000 serisi 512Mb. Ram'e, 1 adet 100Mbit Ethernet kartına sahip.

Düğüm – 1 direkt çapraz ağ kablosu ile Düğüm – 2' in Ethernet 3' üne Düğüm – 3 ise yine çapraz ağ kablosu ile Düğüm – 2' in Ethernet 2' sine bağlıdır. Düğüm – 2 ethernet0 ara yüzü üzerinde GYTE lokal ağına ve oradan da internete bağlıdır. Düğüm – 2 ve Düğüm – 3 üzerinde Fedora Core2 Linux işletim sistemi Düğüm – 1 üzerinde ise SuSE 9.2 Linux işletim sistemi kuruludur. Ara düğüm ( Düğüm – 2 ) üzerinde Düğüm – 1 ve Düğüm – 3' ün bir birlerine ve GYTE ağına bağlanabilmeleri için Düğüm – 2 üzerinde "IP Forwarding" ve "IP Masquerading" aktif edilmiştir. Düğüm – 2'nin ethernet3 arayüzü ve Düğüm – 1'in IP adresleri aynı bloktan reserved adresler, Düğüm – 2'in ethernet2 arayüzü ve Düğüm – 3' ün IP adresleri de başka bir bloktan reserved adresler olarak verilmiştir.

HP WS xv5000
Düğüm - 1

Bu altyapıda bütün düğümler üzerine JNodeOS [Janos Project, 2003] temelli Ants – 2.0.3 versiyonu çalıştırma ortamı olarak kurulmuştur. Bu çalışmada mevcut altyapı üzerinde IPSEC güven sistemini kullanarak aktif kod ve aktif düğümü saldırılardan korumak için bir sistem önerilmiştir. Yukarıda da belirtildiği gibi kriptolama ile paketin veya düğümün tam olarak zararsız olup olmadığı bilgisine ulaşılamaz ama en azından paketin kaynağı ve pakete kim tarafında güven belgesi verdiği belirlenebilir.

Burada Linux işletim sistemli düğümlerin çekirdeklerine IPSEC desteği eklenmiştir. Her düğüm için anahtarlar ve şifreleme ile ilgili bilgiler el ile "/etc/setkey.conf" dosyası içerisinde tutulmuştur. Örnek setkey.conf dosyası önerilen modelin uygulandığı test bed için aşağıdaki gibidir.



**Şekil 3.5 Taşıma Modu Örneği**

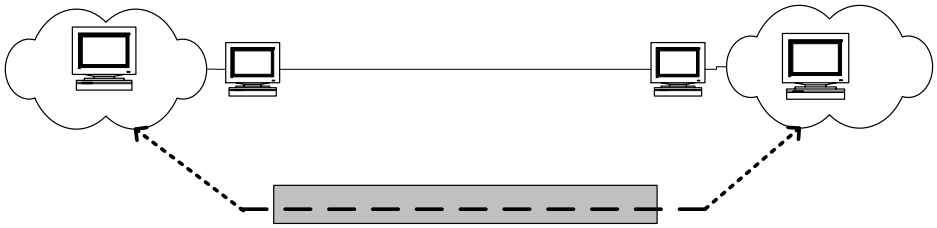
```
# AH SAs using 128 bit long keys
add 192.168.1.100 192.168.2.100 ah 0x200 -A hmac-md5 \
0xc0291ff014dccdd03874d9e8e4cdf3e6;
add 192.168.2.100 192.168.1.100 ah 0x300 -A hmac-md5 \
0x96358c90783bbfa3d7b196ceabe0536b;

# ESP SAs using 192 bit long keys (168 + 24 parity)
add 192.168.1.100 192.168.2.100 esp 0x201 -E 3des-cbc \
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;
add 192.168.2.100 192.168.1.100 esp 0x301 -E 3des-cbc \
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df;

# Security policies
spdadd 192.168.1.100 192.168.2.100 any -P out ipsec
        esp/transport//require
        ah/transport//require;

spdadd 192.168.2.100 192.168.1.100 any -P in ipsec
        esp/transport//require
        ah/transport//require;
```

**Şekil 3.6 Taşıma modu için setkey.conf Dosyası**



**Şekil 3.7 Tünel Modu Örneği**

```
#!/usr/sbin/setkey -f
flush;
spdflush;
# ESP SAs doing encryption using 192 bit long keys (168 + 24 parity)
# and authentication using 128 bit long keys
add 192.168.1.100 192.168.2.100 esp 0x201 -m tunnel -E 3des-cbc \
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831 \
-A hmac-md5 0xc0291ff014dccdd03874d9e8e4cdf3e6;

add 192.168.2.100 192.168.1.100 esp 0x301 -m tunnel -E 3des-cbc \
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df \
-A hmac-md5 0x96358c90783bbfa3d7b196ceabe0536b;
# Security policies
spdadd 172.16.1.0/24 172.16.2.0/24 any -P out ipsec
        esp/tunnel/192.168.1.100-192.168.2.100/require;

spdadd 172.16.2.0/24 172.16.1.0/24 any -P in ipsec
        esp/tunnel/192.168.2.100-192.168.1.100/require;
```

**Şekil 3.8 Tünel modu için setkey.conf Dosyası** 172.16.

Ayrıca bu tek seçeneğimiz değildir modelin uygulandığı test bed ağı küçük olduğu için bu daha ideal ama geniş bir aktif ağ için bu işlemler dinamik olarak da yapılabilir.

Burada hem aktif kod hem de aktif düğüm imzalı olacağı için en azından aktif kodun güven belgesini kimin sağladığı bilgisine ulaşılabilir veya güven belgesi olmayan aktif kodun düğüme alınarak zarar vermesi veya güven belgesi olmayan düğümün aktif koda her hangi bir güvenlik ihlalinde bulunamaması sağlanmış olur.

Ayrıca bu sisteme anetd ( Active Network Daemon)' de kurulmuştur. Düğüm – 2 ( aradüğüm) üzerinde anetd aktif edilmiş ve sistem başlangıcına eklenmiştir.

ABONE (Active Network Backbone) [Whu J.,et all, 2002] test omurgasına kayıt yapılmak istenmiştir ancak maalesef ABONE kayıt sayfası çalışmamaktadır.

Yine de merkezi olmasa da bir anetd çalıştırılarak lokal bir test omurgası oluşturulmuş olacaktır.

Ayrıca burada ki test ortamı için çalıştırma ortamı olarak sadece ants değil, sara, asp, vs.. gibi çalıştırma ortamları da desteklenecektir.

Böylece bu çalışmanın amacı olan şifreleme tekniği ile güven belgesi kaynağının belirlenmesi, güven belgesi olmayan aktif kod ve düğümlerin sisteme zarar verememesi, aktif kodun başka bir aktif kod gibi veya aktif düğümün başka bir aktif düğüm gibi davranamaması **(masquerading)** , aktif düğümün aktif kod üzerindeki özel bilgilere ulaşamaması gibi güvenlik gereksinimleri sağlanmış olacaktır.

# 4 DENEYSEL SONUÇLAR VE ANALİZ

Düğümler arasında ANTS çalıştırma ortamında java dili ile yazılmış ping uygulaması denenmiştir. Bu denemelerde önce güvenlik içeren model kullanılmamış daha sonra ise model uygulanarak sonuçlar değerlendirilmiştir.

## 4.1 Deneysel Sonuçlar

Ping uygulaması Düğüm – 1 üzerinden çalıştırılmış ve Düğüm – 2 ve Düğüm – 3 makinelerine 1000 adet ping uygulaması içeren kapsül gönderilmiştir. Deneme için kullanılan "setkey.conf" aşağıdaki gibidir.

```
#!/usr/sbin/setkey –f
#Aradüğüm üzerinde Düğüm – 1 ve Düğüm – 3 ayarları
#İlk önce mevcut SAD ve SPD bilgiler siliniyor…
flush;
spdflush;
#AH SA'ları için 128 – bit uzun anahtarlar

#Düğüm – 1 için AH SA'ları ve 128-bitlik anahtarlar
add 172.16.2.1 172.16.2.2 ah 0x200 –A hmac-md5
0xa275eccdd014003874d9e8ecdf5a3
add  172.16.2.2 172.16.2.1 ah 0x300 –A hmac-md5
0x7eff023456adcc55598edffaac4452

#Düğüm – 3 için AH SA'ları ve 128-bitlik anahtarlar
add 172.16.3.1 172.16.3.2 ah 0x400 –A hmac-md5
0xecca27540001dd4d9387a35ecfd8e
add  172.16.3.2 172.16.3.1 ah 0x500 –A hmac-md5
0x5244cffaaed55985aaeb654321cca

#ESP SA'ları için 192 – bitlik anahtarlar
#Düğüm – 1 için ESP SA'ları ve 192 – bitlik anahtarlar
add 172.16.2.1 172.16.2.2 esp 0x201 –E 3des-cbc
```

```
0x83726237abde87ac8867affe78123477770acde0000001
add 172.16.2.2 172.16.2.1 esp 0x301 –E 3des-cbc
0x34567adefff347aaffdca00912ace980009cccfee8a8e002

#Düğüm – 3 için ESP SA'ları ve 192 – bitlik anahtarlar
add 172.16.3.1 172.16.3.2 esp 0x401 –E 3des-cbc
0xaa3334087ccfffdea8a6ec660981267aabbced7a780003
add 172.16.3.2 172.16.3.1 esp 0x501 –E 3des-cbc
0xaf5ce907ae25cbf39994f7892222221caaa7a80021cc8

#Güvenlik Politikaları

#Düğüm – 1 İçin güvenlik Politikaları
spdadd 172.16.2.1 172.16.2.2 any –P out ipsec
        esp/transport//require
        esp/transport//require;
spdadd 172.16.2.2 172.16.2.1 any –P in ipsec
        esp/transport//require
        esp/transport/require;

#Düğüm – 3 İçin güvenlik Politikaları
spdadd 172.16.3.1 172.16.3.2 any –P out ipsec
        esp/transport//require
        esp/transport//require;
spdadd 172.16.3.2 172.16.3.1 any –P in ipsec
        esp/transport//require

        esp/transport/require;
```

**Şekil 4.1 Ping Uygulaması İçin Ara Düğüm "setkey.conf"**

```
#!/bin/bash

CONFIG_FILE=ping.config

NODE1_IP=172.16.2.2
NODE1_LABEL= “Düğüm – 1”

NODE2_IP=172.16.2.1
NODE2_LABEL= “Ara Düğüm”

NODE3_IP=172.16.3.2
NODE3_LABEL= “Düğüm – 3”

NODECT=3
```

**Şekil 4.2 Ping Uygulamasını Çalıştıracak “ping” Scripti**

“ping” scripti ile düğüm - 1’den düğüm–3 pinglenecektir. Burada sadece düğüm IP’leri, isimleri ve NODECT=3 ile hangi düğümün pingleneceği belirlenmiştir. Burada CONFIG_FILE=ping.config ile aşağıda verilen ping.config dosyasının kullanılacağı belirtilmiştir.

```
node 172.16.2.1 –routes ping.routes

nchannel 172.16.2.1 aradugum.testbedgyte:*

application 172.16.2.1 apps.ping.PingApplication –target 172.16.3.2 –iter 1000 \
-interv 300 –lossage 1

node 172.16.2.2 –route ping.routes
```

```
nchannel 172.16.2.2 dugum2.testbedgyte:*

node 172.16.3.2 –routes ping.routes

nchannel 172.16.3.2 dugum3.testbedgyte:*

connect 172.16.2.2 172.16.2.1

connect 172.16.2.1 172.16.3.2
```

**Şekil 4.3 "ping.config" Dosyası**

"ping.config" dosyasında düğümlerimiz "node" deyiminden sonra ip adresi ve "–route" parametresi ile takip edilecek rotaları belirlenir. "nchannel" ile düğümün IP adresi ve DNS ismi girilir. "application" ile uygulamanın çalıştığı IP adresi hangi uygulamayı kullanacağı "-target" ile hangi düğümün pingleneceği belirleniyor. "-iter" ile kaç tane ping kapsülünün gönderileceği "-interv" ile ne kadar aralıkla (micro second olarak) kapsül gönderileceği, "-lossage" ile 1 sn. geçtikten sonra kapsül'ün iletilmeyeceği ve kayıp olacağı belirlenmiştir. Test ile ilgili bilgiler Tablo 4.1 de verilmiştir.

**Tablo 4.1 1.000 Kapsül ve 1ms Aralık İçin Model Performans ve Gecikme Bilgileri Tablosu**

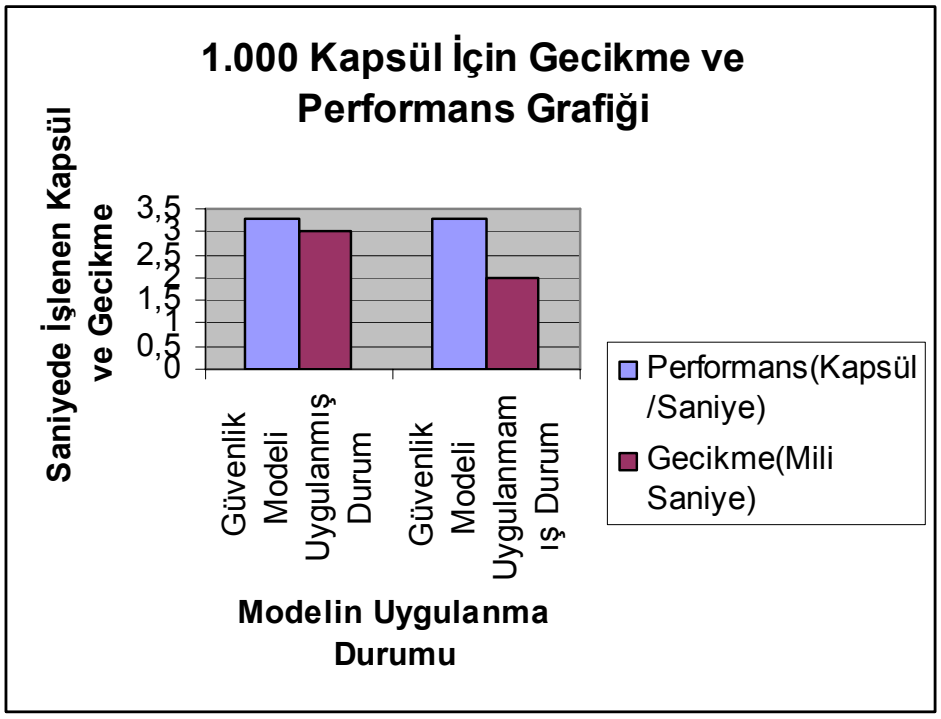
| | Güvenlik Modeli Uygulanmış Durum | Güvenlik Modeli Uygulanmamış Durum |
|---|---|---|
| Performans(Kapsül/Saniye) | 3,3 | 3,35 |
| Gecikme(Mili Saniye) | 3ms | 2ms |

**Tablo 4.2 10.000 Kapsül ve 0,2ms Aralık İçin Model Performans ve Gecikme Bilgileri Tablosu**

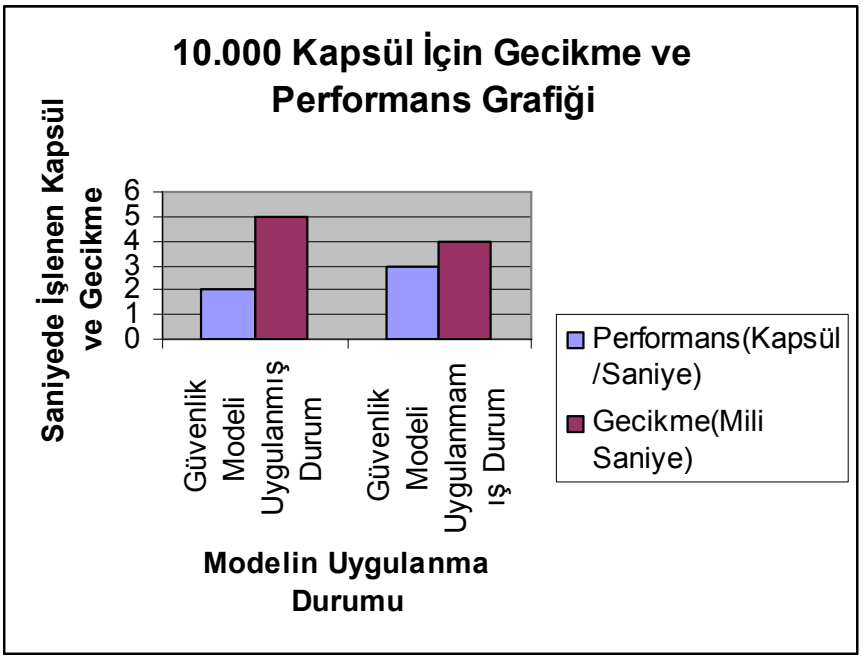
| | Güvenlik Modeli Uygulanmış Durum | Güvenlik Modeli Uygulanmamış Durum |
|---|---|---|
| Performans(Kapsül/Saniye) | 2 | 3 |
| Gecikme(Mili Saniye) | 5ms | 4ms |

**Tablo 4.3 100.000 Kapsül ve 0,2ms Aralık İçin Model Performans ve Gecikme Bilgileri Tablosu**

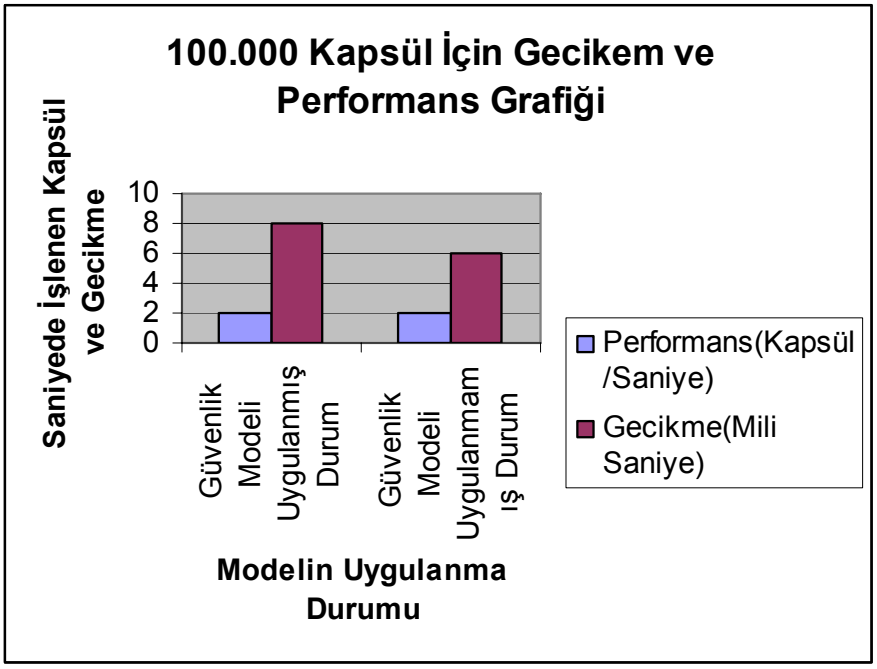
| | Güvenlik Modeli Uygulanmış Durum | Güvenlik Modeli Uygulanmamış Durum |
|---|---|---|
| Performans(Kapsül/Saniye) | 2 | 2 |
| Gecikme(Mili Saniye) | 8ms | 6ms |




**Şekil 4.4 1.000 Kapsül İçin Model Performans ve Gecikme Testi Grafiği**




**Şekil 4.5 10.000 Kapsül İçin Model Performans ve Gecikme Testi Grafiği**




**Şekil 4.6 100.000 Kapsül İçin Model Performans ve Gecikme Testi Grafiği**

Test sonuçlarından ve grafiklerden de anlaşılacağı gibi modelin uygulanması sonucunda çok dramatik bir kapsül işleme performansı ve hız kaybı söz konusu değil

hatta neredeyse hiçbir kayıp söz konusu değil. Çünkü kapsül işleme performansı neredeyse aynı binde 3 lük bir fark var sadece gecikmeye gelince de sadece saniyenin binde biri kadar bir kayıp söz konusu bunlarda normal şartlarda görmezden gelinebilecek kayıplardır. Ancak yüzde olarak gecikmedeki artış %50' yi buldu bunun daha sağlıklı anlaşılabilmesi için paket sayısını 10.000'e paket gönderme aralıklarını da 0,2 mili saniyeye düşürüldüğü bir test daha yapılmıştır. Bu test önerilen model uygulanmadan yapıldığında saniyede işlenebilen kapsül sayısı 3'e düşmüş gecikme ise 4ms olmuştur. Önerilen model uygulandığında ise saniyede işlenebilen kapsül sayısı 2'ye düşmüş gecikme ise 5ms olmuştur. Kapsül sayısını 100.000'e çıkarıp aynı test uygulandığında saniyede işlenebilen kapsül sayısı her iki durum içinde 2 olarak gerçekleşmiştir. Gecikmeler model uygulanmadan 6ms, uygulandıktan sonra 8ms'e kadar çıkmıştır. Performans kayıpları %25' i bulmaktadır. Donanım gücü artırıldığında bu kayıp %5 - %10'lara düşürülebilir. Bu performans kaybı güvenlik göz önüne alındığı zaman normal şartlarda kabul edilebilirdir. Günümüz bilgisayarlarında artık işlemci ve diğer kaynaklar açısından sorun olmadığı için model güncel bilgisayar sistemlerinde uygulanabilir.

## 4.2 Diğer Çözümlerle Karşılaştırma

Diğer çözümlerin hepsi şifreleme, yetkilendirme ve izleme tekniklerini kullanmaktadır. SANTS güvenlik çözümü merkezi bir güvenlik stratejisi izliyor şifreleme içinde uygulama seviyesinde çalışan KeyNotes kullanmaktadır. Bu yüzden performans ve gecikme oranları önerilen modelden daha kötü değerlere sahip güvenliğin merkezi olması da aslında iyi bir çözüm değil sonuçta bütün anahtarlar bir bilgisayarda bulunur ve bu bilgisayarın çalışmaması bütün ağı durduracaktır. Önerilen modelde güvenlik adım adım ( hob by hop ) tekniği ile uygulanmaktadır. Her düğümün sadece kendisine direkt bağlı olan düğümlerle ilgili güvenlik bilgilerine sahip olması yeterlidir. Bilmediği bir düğümden direkt bir iletişim talebi kabul edilmeyecek güvendiği bir düğüm üzerinden gelirse bu istek işlenecektir. Burada anahtarların ve güvenlikle ilgili bilgilerin tutulduğu dosyanın güvenliği önemlidir. Bu dosyalarda oluşturulurken unix sistemlerde ki "chmod 400" komutu ile ve root kullanıcısının sahipliğinde yetkilendirilirse sisteme ulaşılsa bile bu dosyalara ulaşılamayacaktır. Ek bir önlem olarak tripwire **[**Tripwire, 2005**]** yazılımı kullanılarak sistem bütünlüğü sağlanmış ve sistemde meydana gelecek en ufak

değişiklikler bile tespit edilebilir hale gelmiştir. Bununla birlikte de işletim sistemi kaynaklı güvenlik sorunları minimuma indirilmiştir.

SANTS çözümü ayrıca düğüm üzerinde çalışan kodları takip ederek zararlı sonuçlara neden olan kodları ve sahiplerini belirleyip bir daha ki erişimlerini engelleyebilmektedir.

SANTS önerilen modele göre ek güvenlik önlemleri içermektedir. Ancak güvenlik ve performans dengesini yakalayamadığı açıktır.

Test sonuçlarına göre önerilen modelin uygulanabilirliği denenmiş, performans ve gecikmeler kabul edilebilir seviyede gerçekleşmektedir.

# 5 SONUÇ VE ÖNERİLER

Aktif ağlar günümüzde birçok araştırmanın yapıldığı gelecek vadeden bir teknoloji olarak görülmekte ve endüstri tarafından da takip edilmektedir. Ancak laboratuar ortamlarından çıkıp piyasalar da bir standart olması ve gerçek dünyada kullanılmaya başlanması için çok kritik güvenlik zorluklarının aşılması gerekmektedir. Bu zorlukların aşılması ile ilgili birçok teknik üzerinde durulmuştur bunların birini veya bir kaçını içeren bir güvenlik sistemi tasarımı ile aktif ağlardaki güvenlik sorunlarının tamamı olmasa bile büyük bir kısmı giderilebilir. Burada aktif paketin oluşturulduğu programlama kaynaklı sorunlara değinilmemiştir. Zaten programlama dili olarak bilinen genel amaçlı programlama dilleri yaygın olarak kullanılmaktadır, aktif ağlara özel programlama dilleri ise pek rağbet görmemektedir. Ancak burada başta da bahsedildiği gibi aktif ağların en temel katkısı amaca yönelik ağların oluşturulabilmesidir. Bu nedenle amaç belirlenirken güvenlik, performans gerekleri de belirlenmelidir. Açıktır ki güvenlik önlemi arttıkça performansın düşmesi kaçınılmazdır. Önerilen model ile aktif ağlar üzerindeki bütün güvenlik sorunlarını çözmek amaçlanmamaktadır. Ama önemli bir kısmı aşılmıştır.

Güvenlik başlangıçta da belirtildiği gibi boolean bir değişken değildir. Performans ve güvenliğin dengesi iyi sağlanmalıdır. Çünkü kullanılabilirliği bu denge belirlemektedir. Bir sistem çok güvenli olabilir ama performansı çok kötü ise kullanılamaz.

Önerilen model bütün güvenlik zorluklarının üstesinden gelememektedir. Sadece giden gelen kapsüllerin güvenli bir şekilde iletilmelerini sağlamaktadır. Aktif düğüm ile ilgili olarak da güvenilecek kaynaklar belirlenerek bilinmeyen ve güvenilmeyen kaynaklarla iletişim kurulmamaktadır. Bilinen kaynaklardan gelen saldırılar ise sistem kayıtlarından belirlenip gerekli önlemler alınabilmektedir. Ancak aktif düğüm üzerinde zararlı kodların otomatik olarak belirlenip çalıştırılmaması veya zarar verdiği gözlenen bir aktif kodun bir daha ki sefere engellenebilmesi gibi önlemleri alamamaktadır.

Bu çalışma sonucunda gözlenmiştir ki aslında aktif ağlarda ki güvenlik ve kaynak yönetimi çok önemli bir sorun olmakla birlikte asıl sorun standartların olmamasıdır. Bu yüzden üreticiler bu teknoloji üzerine eğilememektedir. Standart bir çalıştırma ortamı, aktif kod programlama dili ve işletim sisteminin belirlenmesi çoğu sorun ve belirsizliği ortadan kaldıracaktır. Bir ağ ortamında çeşitli üreticilere ait aktif düğümler ve bunların her birinin üzerinde farklı çalıştırma ortamı, farklı aktif kod programlama dili ve farklı işletim sistemi olması büyük bir karışıklık ve belirsizliğe yol açmaktadır.

İleriki çalışmalarda önerilen modele bir monitör sistemi eklenebilir. Bu sayede aktif kodların davranışları izlenerek zararlı olma olasılığı olan aktif kodlar baştan hiç çalıştırılmayarak güvenlik açısından büyük bir artı sağlanabilir. Ancak daha önceki çözümlerdeki örneğin SANTS'daki gibi performans ve güvenlik dengesinin iyi olmadığı bir sonuçta çıkabilir.

# KAYNAKLAR

David L. T., Jonath M. S., W.David S., David J. W., Gary J. M. "A Survey of Active Network Research" IEEE Communication Magazine, Vol. 35, No. 1, pp80-86. January 1997

Stamatis Karnouskos "Security implications of implementing active network infrastructures using agent technology", Computer Networks 36(2001) 87 – 100

Dusan G., Borka J. B., Jurij T. "Future active ip networks security architecture", Elsevier Computer Communication , Article in press. (December 2004)

Stuart E., Osman N. E., Dan N., Suresh V. "Commercially viable active networking", ACM SIGOPS Operating Systems Review, January 2002, Volume 36 Issue 1

Wen – Shyen E. Chen , Chih – Lin Hu "A mobile agent – based active network architecture or intelligent network control" Elsevier Information Science 141 ( 2002) 3 – 35

David L. T. ve David J. W. "Towards an Active Network Architecture", Computer Comm. Review, Vol. 26, No. 2, April 1996.

Tomas S. Ve Christion F. T, "Towards Mobile Cryptography, ICSI technical report 97 - 049 ", November 2001.

AN Security Working Group, "Security Architecture for Active Nets", November 2001

The Janos Project http://www.cs.utah.edu/flux/janos/, University Of Utah, USA

Jian-Guo W., Zeng-Zhi L., Ya-Nan K. "Research and implementation of a scalable secure active network node" Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on Volume 1, 4-5 Nov. 2002 Page(s):111 - 115 vol.1

Youngsoo K., ; Jungchan N, Seungwon S**. "**A secure method for transferring active packet using digital signature schemes" Telecommunications, 2003. ICT 2003. 10th International Conference on Volume 1, 23 Feb.-1 March 2003 Page(s):66 - 69 vol.1

Tullmann P., Hibler M., Lepreau, J. **"**Janos: a Java-oriented OS for active network nodes" Selected Areas in Communications, IEEE Journal on Volume 19, Issue 3, March 2001 Page(s):501 - 510

Tang Y., Gong Y. **"**A survey of authorization based Active Networks security" Communications, Circuits and Systems, 2004. ICCCAS 2004. 2004 International Conference on Volume 1, 27-29 June 2004 Page(s):22 - 24 Vol.1

Sterne D., Djahandari K., Balupari R., La Cholter W., Babson B., Wilson B., Narasimhan P., Purtell A., Schnackenberg,D., Linden S., **"**Active network based DDoS defense" DARPA Active NEtworks Conference and Exposition, 2002. Proceedings 29-30 May 2002 Page(s):193 – 203

Iqbal A., Khiyal M.S.H., Sher M., "A simple practical active network architecture" Computer and Information Technology, 2004. CIT '04. The Fourth International Conference on 14-16 Sept. 2004 Page(s):664 - 667

Zhongwen L., Shui Y., Leming L., **"**A new safety mechanism of active networks" Info-tech and Info-net, 2001. Proceedings. ICII 2001 - Beijing. 2001 International Conferences on Volume 2, 29 Oct.-1 Nov. 2001 Page(s):779 - 785 vol.2

Kulkarni A.B., Minden G.J., Hill R., Wijata Y., Gopinath A., Sheth S., Wahhab F., Pindi H., Nagarajan A.; **"**Implementation of a prototype active network" Open Architectures and Network Programming, 1998 IEEE 3-4 April 1998 Page(s):130 - 142

Tarantola C.; **"**Dynamic Active Network Services" Mobile Data Management, 2004. Proceedings. 2004 IEEE International Conference on 2004 Page(s):173

Merugu S., Bhattacharjee S., Zegura E., Calvert K.; **"**Bowman: a node OS for active networks**"** INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE Volume 3, 26-30 March 2000 Page(s):1127 - 1136 vol.3

Peichu S, Aronhime P.; **"**Active networks and node addition" Circuits and Systems, 1996., IEEE 39th Midwest symposium on Volume 3, 18-21 Aug. 1996 Page(s):1115 - 1118 vol.3

Law K.L.E., Leung R.; "A design and implementation of active network socket programming**"** Computer Communications and Networks, 2002. Proceedings. Eleventh International Conference on 14-16 Oct. 2002 Page(s):78 - 83

http:/sourceforge.net/projects/tripwire , 2005

# ÖZGEÇMİŞ

1978 yılında Yozgat'ta doğdu. İlk ve orta öğretimini 1984 – 1995 yılları arasında Yozgat'ta tamamladı. 1995 yılında girdiği Muğla Üniversitesi Matematik Bölümü'nden 1999 yılında mezun oldu. 2002 Yılında Gebze Yüksek Teknoloji Enstitüsü Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'nde yüksek lisans eğitimine başladı. Halen bir kamu kuruluşunda Sistem ve Network Sorumlusu olarak çalışmaktadır.