T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SANAL ORTAM ÜZERİNDE OLUŞTURULAN ÖRNEK BİR KURUMSAL AĞ TOPOLOJİSİNİN SNMPv3 İLE TOPOLOJİ KEŞFİ UYGULAMASI

YÜKSEK LİSANS TEZİ

Bilg. Müh. Musa BALTA

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ

Tez Danışmanı : Yrd. Doç. Dr. İbrahim ÖZÇELİK

Ocak 2012

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SANAL ORTAM ÜZERİNDE OLUŞTURULAN ÖRNEK BİR KURUMSAL AĞ TOPOLOJİSİNİN SNMPv3 İLE TOPOLOJİ KEŞFİ UYGULAMASI

YÜKSEK LİSANS TEZİ

Bilg. Müh. Musa BALTA

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ

Bu tez .. / .. /201 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.

Doç. Dr. Ahmet ÖZMEN	Doç. Dr. Celal ÇEKEN	Yrd. Doç. Dr. İbrahim ÖZÇELİK
Jüri Başkanı	Üye	Üye

TEŞEKKÜR

Projenin gerçekleştirilme aşamasında kaynak sıkıntısı çekilmemiştir. Çok sayıda doküman, kaynak materyal, örnek proje ve programlardan yararlanılmıştır. Bu konu üzerinde ileriki zamanlarda, çalışma ve araştırma yapmak isteyecek arkadaşlara büyük yardımı olacağına inandığım bu projeyi hazırlarken bana her konuda yardım eden, desteğini hiç esirgemeyen, değerli fikirleriyle yol gösteren değerli hocam Yrd. Doç. Dr. İbrahim ÖZÇELİK'e ve bana sabırla katlanan tüm mesai arkadaşlarıma ve canım aileme teşekkürü bir borç bilirim.

Bu çalışma SAÜ Bilimsel Araştırma Projeleri Komisyonu tarafından desteklenmiştir. (Proje no: 2011-50-01-072)

İÇİNDEKİLER

SİMGELER VE KISALTMALAR LİSTESİ

AES : Advanced Encryption Standard
AFT : Adress Forwarding Table
ASN.1 : Abstract Syntax Notation One
API : Application Programming Interface
BER : Basic Encoding Rules
CMIP : Common Management Information Protocol
CNLS : Connectionless Network System
CMIS : Common Management Information System
DES : Data Encryption Standard
DMI : Desktop Management Interface
DMTF : Distributed Management Task Force
DMZ : Demilitarized Zone
EGP : Exterior Gateway Protocol
GNS3 : Graphical Network Simulator 3
IAB : Internet Advisory Board
ICMP : Internet Control Message Protocol
IETF : Internet Engineering Task Force
IP : Internet Protocol
LAN : Local Area Network
MAC : Media Access Control
MIB : Management Information Base
MIF : Memory Initalization File
MRTG : Multi Router Traffic Grapher
NAT : Network Address Translation
NMS : Network Management System
OID : Object Identifier
OSPF : Open Short Path First

PDU	: Package Data Unit
RFC	: Request For Comment
SGMP	: Simple Monitoring Gateway Protocol
SMI	: Structure of Management Information
SNMP	: Simple Network Management Protocol
TCP	: Transmission Control Protocol
UDP	: User Datagram Protocol
VPN	: Virtual Private Network
QOS	: Quality of Service
WAN	: Wide Area Network

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

ÖZET

Anahtar kelimeler: Ağ yönetimi, SNMP, GNS3, VWMARE Workstation, Topoloji Keşfi

Günümüzde hızla gelişen bilişim dünyası beraberinde ağ güvenliği, daha hızlı veri iletimi, kolay yönetim gibi yeni gereksinimler getirmiştir. Kurumlar bu yeni gereksinimleri karşılayabilmek için mevcut ağ yönetim sistemlerini revize etmek veya yeni bir ağ yönetim sistemine geçmek zorunda kalırlar. İyi bir ağ yönetim sistemi için ise de ağdaki cihazların özelliklerinin tespiti, aradaki bağlantıların tespiti ve topoloji keşfi gibi kavramlar ön plana çıkar.

Ağ topoloji keşfi ile alakalı bir çok çalışma yapılmıştır. Gerek akademik, gerekse ticari çalışmalar olsun genelde gerçek cihazlar üzerinde olmuştur. Sanallaştırmanın geliştiği bu dönemde, bu tez çalışmasıyla SNMPv3 ile sanal platformlar (GNS3 ve VMWARE Workstation) üzerinde oluşturulan örnek bir kurumsal ağın topoloji keşfi yapılmıştır.

THE TOPOLOGY DISCOVERY OF AN EXAMPLE OF ENTERPRISE NETWORK TOPOLOGY CREATED IN A VIRTUAL ENVIRONMENT WITH SNMPv3

SUMMARY

Key Words: Network management, SNMP, GNS3, VMWARE Workstation, Topology Discovery

Today, along with the rapidly developing information systems have come with new features which are network security, easy management and faster data transmission. So companies has to revise their current network system or replace their system with a new one for solving this network situations. To desing an effective network management system, there are some important topics such as device functionality, connectivity between devices and topology discovery.

There are a lot of workout in this area to develop network management systems. Both academic and commerciel workouts are on the physical real machines. During the period that virsulation has developed, network management systems also start to work on both virtual machines and servers anymore. In this study, an enterprise network topology that was created in virtual environment (GNS3 and VMWARE Workstation) is discovered through an algorithm which uses SNMPv3.

BÖLÜM 1. GİRİŞ

Günümüzde kurumlar kendi alt yapılarında, servis kalitesi, hızlı veri iletişimi, güvenilirlik ve ağ güvenliği gibi hızla önemi artan ağ kriterlerini karşılamak zorundadırlar. Bu yüzden kullanımı daha kolay, işlevselliği daha fazla ve değişen ağ topolojilerine çabuk adaptasyon sağlayabilecek ağ yönetim sistemleri geliştirilmektedir. Geliştirilen bu ağ yönetim sistemleri kurumların ihtiyaçlarına göre şekillenebilir. Bazı yönetim sistemleri sadece sunucu sistemlerin kontrolü üzerine, bazıları algılayıcı/eyleyici sistemler üzerine, fakat genelde ise tüm ağ topolojisi üzerine olurlar. Bu farklılıklar beraberinde farklı protokollerin ortaya çıkmasına neden olmuştur. Günümüzde ağ yönetim sistemlerinde kullanılan en yaygın ağ yönetim protokolü SNMP (Simple Network Management Protocol, Basit Ağ Yönetim Protokolü)'dir. SNMP, kendi çalışma mekanizmasından dolayı ağ ihtiyaçlarını birebir karşılayabilecek özelliklere sahiptir. Bu yüzden ağ yönetim sistemlerinin temelinde yer alır.

Ağ yönetim sistemlerinin geliştirilme ve test edilme süreçleri hem maliyetli hem de uzun bir zaman alan süreçtir. Kendi ağ alt yapılarına çok büyük yatırımlar yapan kurumlar, ufak bir ayrıntıyı gözden kaçırmaları durumunda telafisi olmayan veya çok büyük hasarlı sonuçlarla ortaya karşı karşıya kalabilirler. Bu tarz riskleri önlemek için son yıllarda sanallaştırma konusu büyük önem kazanmıştır. Sanallaştırma ile gerek teknik alt yapı maliyetleri gerekse test ve uygulama süreçlerinde oluşabilecek hata ve riskleri minimize etmek çok kolaylaşmıştır.

İyi bir ağ yönetimi, hata yönetimi, trafik yönetimi, güvenilirlik, esneklik ve güvenlik gibi konu başlıklarını kendi içerisinde barındırmak zorundadır. Bu alanların iyi bir şekilde yönetilebilmesi için ise öncelikle ağ topoloji keşfinin yapılması, ağdaki cihazların aralarındaki bağlantı tipinin tespit edilmesi ve ağdaki cihazların özelliklerinin belirlenmesi gibi işlemlerin yapılması gerekmektedir. Günümüzde

efektif ağ yönetimi büyük kurumlar için çok önem arz ettiğinden, ağ yönetimi ve topoloji keşfi birçok akademik çalışmaya da konu olmuştur. Aşağıda bu çalışmalarla alakalı bilgiler verilmiştir.

[1-9] numaralı akademik çalışmalarda ağ yönetimi ve topoloji keşfi çalışmaları fiziksel cihazlar üzerinde, değişik teknikler kullanılarak yapılmış çalışmalardır. Her bir çalışma gerek kullandıkları teknikler olsun, gerekse kullandıkları algoritmalar olsun birbirleriyle farklılıklar göstermektedirler.

Bu tez çalışmasında diğer akademik çalışmalardan farklı olarak, kurumsal bir ağ topolojisi sanal bir ortam üzerinde modellenerek yine farklı bir sanal ortam üzerinde geliştirilen uygulama sayesinde aynı kurumsal ağın topoloji keşfi yapılmıştır. Uygulamada iki farklı sanallaştırma platformu kullanılmıştır. Bu sanal platformlardan GNS3 (Graphical Network Simulator 3) kurumsal ağı modellemek için, VMWARE Workstation ise geliştirilen uygulama kodunun çalıştırıldığı platform için kullanılmıştır.

Bu tez çalışması 5 ayrı bölümden oluşmaktadır. 2. bölümde, ağ yönetimi ve SNMP kavramları ayrıntılı şekilde anlatılmıştır. Bir ağ yönetim sisteminin ne olduğu, nelerden oluştuğu, hangi mimarilerin olduğu, SNMP protokollerinin temel özellikleri ve çalışma yapısı hakkında detaylı bir bilgi sunulmaktadır. Ayrıca konuyla ilgili verilen bilgiler, şekiller ve tablolar ile desteklenmiştir.

Tezin 3. bölümünde, uygulamada kullanılan teknoloji bileşenlerinin temel bilgileri, bunların çalışma yapıları ve birbirleriyle olan entegrasyonu anlatılmıştır. İki önemli bileşenden bahsedilmiştir; VMWARE Workstation ve GNS3. Gerek ticari gerekse kamu alanında kulanım alanı çok yaygın olan VMWARE Workstation hakkında temel bilgiler verilip, mimarisi ve dosya yapısı üzerinde durulmuştur. Aynı şekilde sanallaştırma alanında kullanılan ve açık kaynak kodlu olan GNS3 hakkında da temel

bilgiler verilmiştir. Bu iki bileşenin kurulumu ile ilgili bilgiler tezin ek kısmında sunulmuştur.

Tezin 4. bölümü, yapılan uygulamayı anlatmaktadır. Uygulamanın kaç aşamadan oluştuğu, uygulamanın algoritmaları, daha önceden aynı konu ile yapılmış benzer çalışmalardan farkı, kullanılan SNMP kütüphaneleri ve yazılım geliştirme ortamları, uygulama kodları ve ağ topolojisinden bahsedilmiştir.

Tezin 5. bölümü ise tezin sonuç kısmını kapsamaktadır. Bu kısımda uygulamanın bize neler kattığı ve uygulamadan çıkarılan sonuçların neler olduğu anlatılmıştır.

BÖLÜM 2. AĞ YÖNETİMİ

2.1. Giriş

Günümüz ağ topolojileri gerek büyüklükleri gerekse karmaşıklıklarından dolayı ağ yönetimini gerekli hale getirmişlerdir. Büyük ağlar, kendi kurumları için maliyet, zaman, performans, güvenilirlik ve güvenlik konularında çok önemli yere sahiptirler. Ancak yine aynı ağlar iyi yönetilemedikleri takdirde aynı başlıklar altında kurumlarına sıkıntı çıkarabilirler. Bu yüzden takip eden bölümlerde hem yapılacak uygulamanın alt yapısını oluşturmak hem de ağ yönetiminin anlaşılabilirliğini sağlamak amacıyla ağ yönetimi ile alakalı önemli kavramlar ele alınacaktır.

2.2. Ağ Yönetiminin Amacı

Aşağıda detaylandırılacak ağ yönetim amaçları genel olarak üç ana başlık altında toplanır [10] :

a. Ağın devamlılığı (çalışır halde tutmak)
b. Ağın performansını yönetmek
c. Maliyeti düşürme

Ağın devamlılığı; sistemlerin ve ağların işlevsel ve çalışır halde olmalarını tanımlar. Bu da ağda herhangi bir duraksamanın veya hatanın oluşmaması gerektiği anlamına gelir. Olası bir hatada veya duraksamada ağın işlevselliği ve oluşan hatanın önemine göre çok büyük hasarlar oluşabilir. Ağ yönetimi sistemleri içerisinde görüntüleme (monitoring) özelliği sayesinde ağ periyodik olarak veya istenildiği anda (manuel) kontrol edilip olası bir hatadan kaçınılmış olur.

Ağın sadece düzgün çalışması işlevsel bir ağ için yeterli olmaz. Aynı zamanda ağ kalitesi de bir ağ için önemlidir. Bundan dolayı ağ yönetiminin büyük bir özelliği de kabul edilebilir bir performans seviyesi sağlamasıdır.

Yeni ağ cihazlarının alımı ve bunların kurulumu, uzun süreçte maliyet ve bakım giderlerine kıyasla daha ucuza gelir. Ağ yönetimi iki ana kategoriye ayrılır: Reaktif ve Proaktif. Kurum maliyetlerini azaltmada Reaktif yönetim daha pahalıdır. Bunun sebebi, bir ağ çöktüğü zaman veya sorunla karşılaştığı zaman maliyet gözetilmeksizin problemin hemen çözülmesi gerekir. Fakat Proaktif yönetim ise bu performans ve hata yönetimlerini öncelikli alanlar olarak belirler ve ona göre işlem yapar.

2.3. Ağ Yönetim Alanları

ISO tarafından kabul görmüş 5 büyük ağ yönetim alanı vardır. Bunlar sırasıyla aşağıda özetlenecektir:

Konfigürasyon Yönetimi: Konfigürasyon yönetimi ağ üzerindeki cihazları güncelleme, parametre değiştirme ve cihazların ayarlarını değiştirme işlemlerini yapar. Cihaz çalışmaya başladığı andan itibaren ağ yönetim sisteminin ilk çalışmaya başlayan fonksiyonudur ve ajan cihazlarında herhangi bir değişiklik olup olmadığını kontrol eder. Kısaca Konfigürasyon Yönetiminin amacı konfigürasyon tanımlamasını, kotrolünü ve durum değişikliklerini kullanarak ürünlerin entegrasyonunu sağlamak ve korumaktır.

Konfigürasyon Yönetiminin Faydaları;

a. Ağ yöneticisinin, ağ araçlarının konfigürasyonu üzerindeki kontrolünü attırır.
b. Konfigürasyonla ilgili verilere hızlı erişim sağlar.
c. Değişikliklerin daha kolay bir şekilde yapılmasını sağlar.

d. Ağ bileşenlerinin envanterini tutarak, ağ yöneticisine daha fazla yardımcı olur.
e. Bu envanter sayesinde, kullanılmakta olan sistem hakkında bir çok rapor alınabilir.

Oturum Yönetimi: Kurulu olan ağdaki cihazların sistem bilgilerinden (cpu, ram, interface …vs), üzerinden geçen ağ trafiğine kadar her türlü bilginin yönetildiği ve saklandığı oturumdur.

Hata Yönetimi: Veri ağında oluşan hataları veya riskleri belirleyen fonksiyondur. Genel olarak şu işlevlere sahiptir: hatayı algılama**,** hatayı izole etme ve eğer mümkünse hatayı düzeltme işlevlerine sahiptir.

Performans Yönetimi: Ağdaki cihazların donanımını, yazılımını ve ortam erişim cihazlarının performanslarını ölçen bir modüldür. İşlem hacmi, kullanım yüzdesi, hata oranları ve cevap süresi gibi parametreler performans yönetimi içerisinde sunulur.

Güvenlik Yönetimi: Ağdaki cihazların güvenliğinden, veri akışı güvenliğine kadar oluşabilecek tüm riskleri önlemeye yönelik bir modüldür. Yönlendirici veya anahtar cihazları kontrol ederek periyodik olarak bilgileri kaydeder. Güvenlik ihlaline sebep olan girişimleri inceler.

Bu çalışma içerisinde yapılacak uygulama, konfigürasyon yönetimi, güvenlik yönetimi ve performans yönetimi konularını içermektedir.

2.4. Ağ Yönetim Mimarileri

Günümüzde ağ yönetiminin daha efektif, kolay ve maliyetinin azaltılması için bazı mimari örnekleri ortaya atılmıştır. Ağ yönetim sisteminin kurulacağı kurumların yapısı da göz önüne alınarak, ağ yöneticileri kurumun ihtiyaçları doğrultusunda ağlarını tasarlarlar. Bu ağ kurulumu aşamasında uyulması gereken bazı kurallar vardır. Bu kurallardan bazıları aşağıdaki gibi sıralanmaktadır [11];

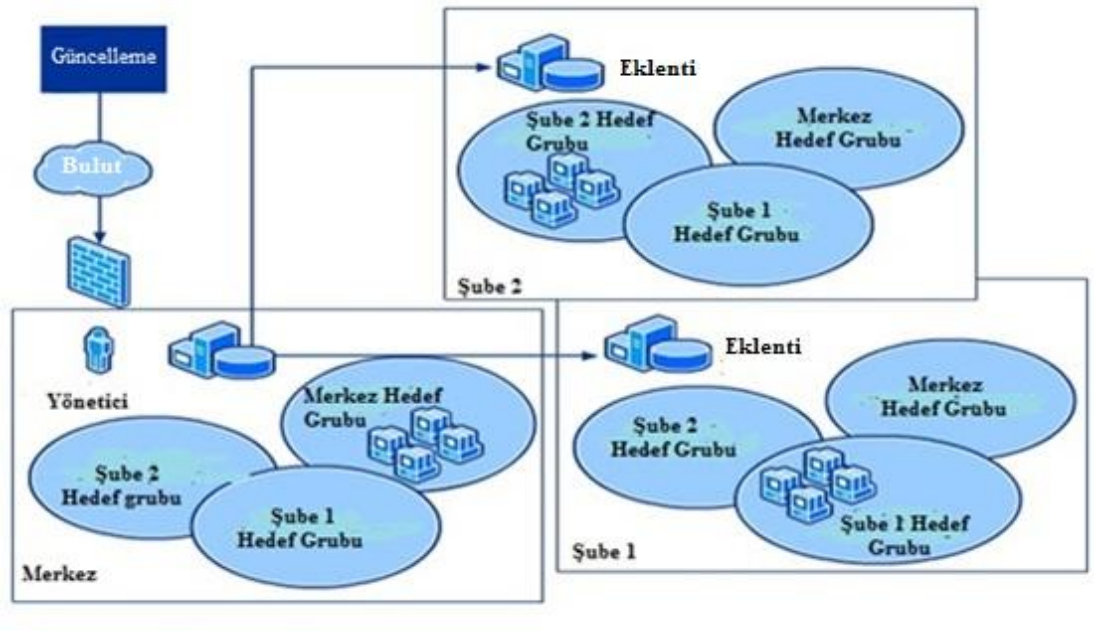
a. Trafik yükünün azlığı
b. Ağdaki cihazlara kolay erişim ve kolay yönetim
c. Yedeklilik
d. Güvenilirlik
e. Güvenlik
f. Maliyet …

Tüm bu özelliklerin hepsinin bir arada sağlandığı ağlar ideal ağlar olarak kabul edilir, ancak gerek şirketin kurumsal yapısı gerekse fiziksel şartların elverişsizliği nedeniyle değişik çözümler bulunmuştur. Genel olarak ağ yönetim mimarilerini 3 ana başlık altında inceleyebiliriz:

2.4.1. Merkezi yönetim

Bu sistemde tüm ağ yönetim platformu basit tek bir bilgisayar sistemi üzerine kurulmuştur. Yedek alınması için tüm bilgisayar sisteminin başka bir sistem tarafından yedeğinin alınması gerekir. Ağdaki cihazlara erişimi sağlayıp, gerekli ayarlamalar yapıldıktan sonra onları istediği gibi yönetebilir. Daha küçük ölçekli firmalarda (küçük şirketler) ve yerel alan ağlarında (LAN, Local Area Network) kullanılır [11].

Şekil 2.1'de merkezi yönetim mimarisi gösterilmiştir. Bu mimaride farklı şubeler tek bir merkezden yönetilmektedir. Şubelerin birbirleriyle direk teması söz konusu değildir.



Şekil 2.1. Merkezi yönetim

Tablo 2.1'de merkezi yönetim mimarisinin avantajları ve dezavantajları verilmiştir.

Tablo 2.1. Merkezi yönetim avantajlar-dezavantajlar

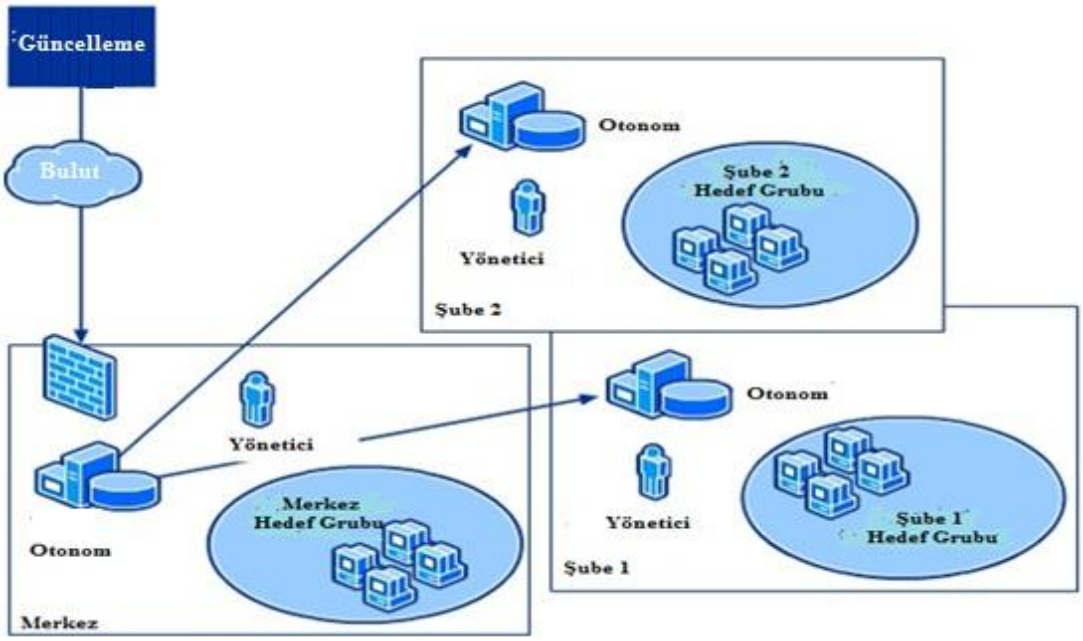
	Avantajlar	Dezavantajlar
1	Hataları ve durumları tek noktandan takip	Tek sistem yedek alma ve hata toleransı için yeterli değil
2	Ağ uygulamalarına ve bilgilere tek noktan erişim	Yeni sistem veya cihaz eklendiğinde, sistemi yenilemek zor olabilir.
3	Güvenlik daha kolay	Tüm istekler bir noktadan geçeceği için trafik yükü

2.4.2. Dağıtık yönetim

Daha çok güvenlik ve güvenilirliğin ön plana çıktığı karmaşık ve kurumsal yapılarda kullanılır. Orta ve büyük çaplı şirketler için tasarlanmış bir ağ yönetim yapısıdır. Fabrikalar, bankalar veya üniversiteler gibi geniş alan ağlarında (WAN, Wide Area Network) kullanılır. Ağ yönetim platformu tek bir sistem üzerine kurulu olmaktan ziyade bölgelere ayrılarak yönetimi kolaylaştırır. Böylelikle;

a. Ağdaki cihaz alarmlarını ve olayları
b. Tüm ağ bilgisini
c. Tüm yönetim uygulamaları bu tip ağlarda kolaylıkla gerçekleştirilebilir [11].

Şekil 2.2'de dağıtık yönetim mimarisi gösterilmektedir. Bu mimaride merkezdeki yönetici sadece şube yöneticileri yönetir. Şube yöneticiler kendi iç yapılarını yönetir.



Şekil 2.2. Dağıtık yönetim

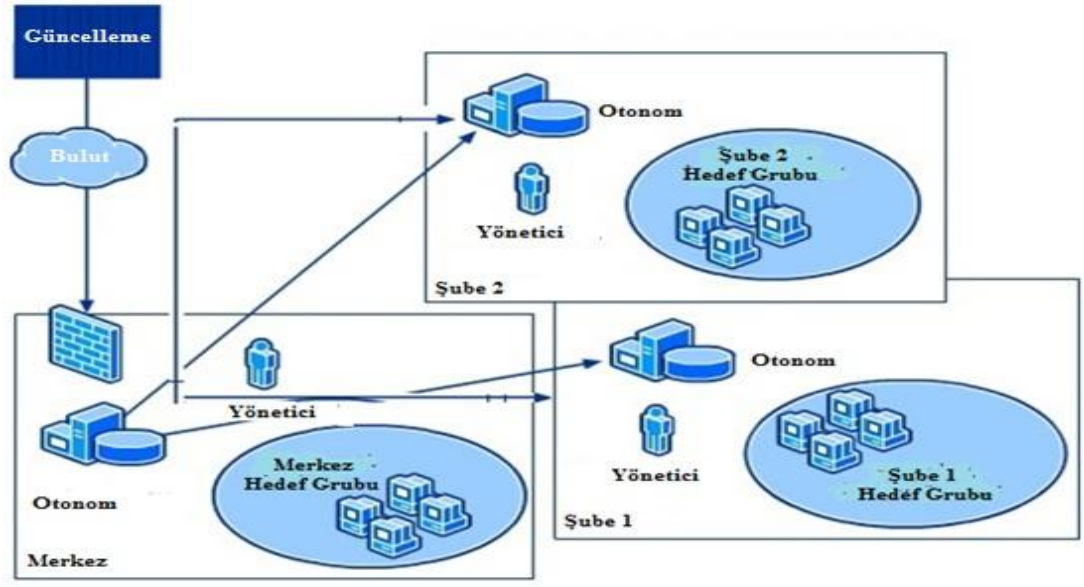
Tablo 2.2'de dağıtık yönetimin avantajları ve dezavantajları verilmiştir .

Tablo 2.2. Dağıtık yönetim avantajlar-dezavantajlar

	Avantajlar	Dezavantajlar
1	Ağı yönetmek için bir çok sistem	Bilgi toplamak daha zor ve zaman kaybı fazla
2	Sistemin yedeği her zaman mevcut	Güvenlik açığı
3	Tek bir sistemdeki trafik yükü azalır.	Maliyet

2.4.3. Hibrid yönetim

Hibrid yönetimde ise yönetimin hem merkezi hem de dağıtık sistemlerin bütünleşik olarak kullanılmasından oluşan bir sistemdir. Böylelikle yedek alma, sistem çökmesi, veri hızında düşme gibi problemler minimize edilmiş olacaktır [11]. Farklı lokasyonlara hizmet veren, altyapı gereksinimi fazla olan büyük kurumlar için kullanılır. Özellikle servis sağlayıcıların tercih ettiği bir modeldir. Şekil 2.3'de hibrid yönetim mimarisi gösterilmektedir. Bu mimaride merkezdeki yönetici hem şube yöneticilerini hem de şube iç yapısını yönetirler.



Şekil 2.3. Hibrid yönetim

Tablo 2.3. Hibrid yönetim avantajlar-dezavantajlar

	Avantajlar	Dezavantajlar
1	Hataları ve durumları tek noktandan veya çok noktadan takip	Tüm sistemin manuel olarak tasarlanması gerekir.
2	Ağ uygulamalarına ve bilgilere kolay erişim	Çok erişim noktası olacağından güvenlik problemi
3	Yönetim daha kolay	Maliyet

Bu çalışmada, merkezi yönetim mimarisini temel alan örnek bir kurumsal topoloji üzerinde çalışılmıştır.

2.5. Ağ Yönetim Protokolleri

Büyüyen ağ yönetim mimarilerini daha iyi kontrol edebilmek için geçmişten günümüze birçok teknik ve protokol ortaya çıkmıştır. Ağ yönetim sistemleri için kullanılan bu yöntemler gerek kullanım alanları olsun gerekse işlevsellikler olsun birbirleriyle büyük farklılıklar göstermektedirler. Bu bölümde ağ yönetim sistemleri içinde en çok tercih edilen bazı ağ yönetim protokolleri anlatılacaktır.

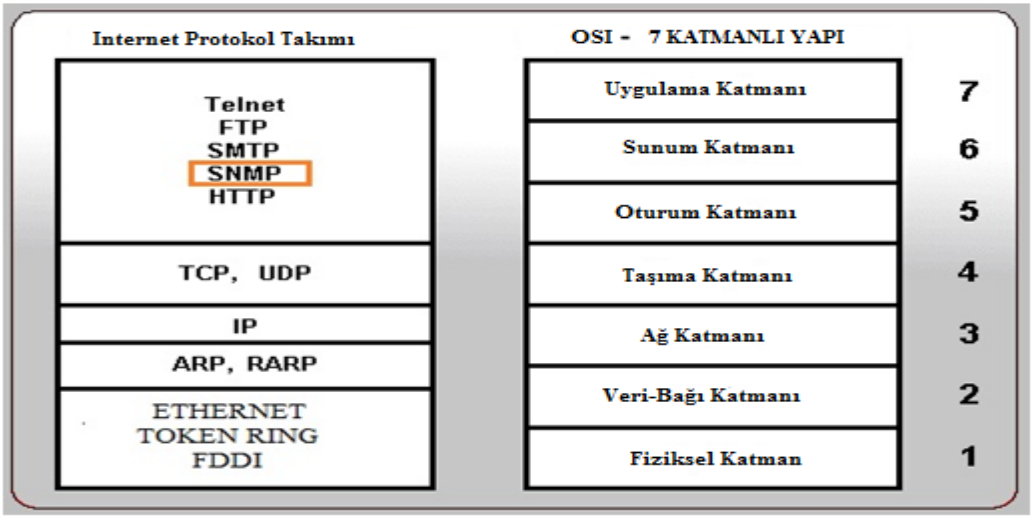
2.5.1. SNMP

SNMP, ağ cihazlarının yönetimini ve izlenmesini kolaylaştıran bir uygulama katmanı protokolüdür. TCP/IP protokol ailesinin bir parçası olan SNMP; ağ yöneticilerinin ağ performansını arttırması, ağ problemlerini bulup çözmesi ve ağlardaki genişleme için planlama yapabilmesine olanak sağlar. Bu protokol sayesinde ağdaki hemen her türlü cihaz izlenebilir hatta yapılandırmaları değiştirilebilir. Genel olarak tanımında ise ağ yöneticilerine ağın performansını kontrol edebilme, ağda oluşan problemleri bulma ve çözme ve ağın verimliliğini artırabilme imkanı sağlar. Şekil 2.4'te SNMP protokolünün avantajları ve dezavantajları verilmiştir.

Tablo 2.4. SNMP avantajlar-dezavantajlar

	Avantajlar	Dezavantajlar
1	Basit dizaynı kolay entegrasyon sağlıyor.	Düşük güvenlik (son sürüm hariç)
2	Kullanım alanı çok geniş.	UDP'yi kullandığından güvenilirlik düşük.
3	Güncelleme işlemi kolay	Çok fazla ağ trafiği yaratır.
4	Artan gereksinimlere kolay adaptasyon	
5	Bir standart olması	
6	Genişletilebilir ve taşınabilir olması	
7	Dağıtık ve merkezi yönetimi desteklemesi	

Şekil 2.4'te, SNMP protokolünün, OSI referans modeli ve TCP/IP protokol kümesi içerisindeki yeri gösterilmiştir.



Şekil 2.4. SNMP'nin OSI referans modelindeki yeri

2.5.2. CMIP

Ağ yönetimi için geliştirilmiş olan başka bir protokoldür. Yönetilebilen cihazlar ve ağ uygulamaları arasında çalışan servislere uyumlu bir uygulamadır. CMIP, OSI referans modeli baz alınarak, ITU-T X.700 tarafından üretilmiştir.

CMIP, yönetilen cihazlardaki yönetim bilgisini modeller ve cihazlar üzerinde hem değişiklik yapmaya hem de performans ölçmeye izin verir. CMIS modelinin kendine özgü sistemi vardır ve bu sistemden türetilen servisleri kullanılır [12].

Yetkilendirme, erişim kontrolü ve güvenlik logları noktasında iyi bir güvenlik sağlar ve hat kopması, cihazın kapanması gibi alışılmadık ağ şartlarına da uyumluluk sağlar. Tablo 2.5'de CMIP protokolünün avantajları ve dezavantajları verilmiştir.

Tablo 2.5. CMIP avantajlar-dezavantajlar

	Avantajlar	Dezavantajlar
1	Protokol değerleri karmaşık görevleri yerine getirebilir.	Sistem büyük ve karmaşık olduğundan sadece iyi ağ cihazları kullanılmalıdır.
2	Daha etkili ağ yönetim sistemi	Oluşan hataya anlık çözümler bulunamaz.
3	Daha güçlü güvenlik (kullanıcı doğrulama,erişim kontrolü..)	

2.5.3. DMI

SNMP ve CMIP'e oranla daha kısıtlı alanda görev yapabilen bir yönetim protokolüdür. Büyük sistemler yerine masaüstü, dizüstü ve sunucu makineler gibi cihazlar üzerinden işlem yaparlar. Bu cihazları yöneten işletim sistemlerinden soyutlayarak yönetim işlemi yapan bir framework destekler. DMTF (Distributed Management Task Force) tarafından geliştirilmiştir. MIF (Memory Initialization File) dosya yapısına sahiptir.

SNMP ile birlikte sorunsuz olarak çalışabilir. Mesela bir SNMP isteği geldiğinde bunu SNMP MIB ile kendi MIF yapısıyla doldurabilir [10]. Tablo 2.6'da DMI protokolünün avantajları ve dezavantajları verilmiştir.

Tablo 2.6. DMI avantajlar-dezavantajlar

	Avantajlar	Dezavantajlar
1	Mimari yönetim uygulamalarıyla donanım arasında standart bir arayüz sağlar.	Büyük sistemler için tercih edilmez.
2	Bugün en çok kullanılan istemci protokolüdür.	

Tablo 2.7'de bu bölümde anlatılan ağ yönetim protokollerinin güvenlik, performans, maliyet gibi alanlardaki karşılaştırması verilmiştir.

Tablo 2.7. Ağ yönetim protokollerinin karşılaştırılması

	SNMP	**DMI**	**CMIP**
Güvenlik	Normal	Normal	Yüksek
Maliyet	Normal	Düşük	Yüksek
Kolay Kurulum	Kolay kurulabilir	Kolay Kurulabilir	Kurulumu zor
Ağ Değişikliği	Adapte olabilmesi için zaman gerekli	Hemen adapte olabilir.	Zor adapte olur.
Trafik Yükü	Yüksek	Normal	Yüksek
Hız	Yüksek	Yüksek	Normal
Kullanım Alanı	Kullanım alanı geniş	Kullanım alanı az	Kullanım alanı az

Bu tez çalışmasında, ağ yönetim protokolü olarak SNMP protokolü kullanıldığından bir sonraki bölümde bu protokol ile alakalı detaylı bilgiler verilecektir.

2.6. SNMP

TCP/IP'nin ilk çıktığı günlerde, ağ yönetimlerini geliştirmek ve dizayn etmek için ufak girişimler olmuştur. Bunlardan biri olan ICMP protokolü, birkaç tane basit ağ yönetim özelliği sunmaktan öteye geçememiştir. ICMP, ping mekanizmasını kullanılarak, TCP/IP makinelerinin arayüzlerinin açık veya kapalı durumda olduğunu, dönen cevaplar sayesinde anlayabilmektedir. Ping mekanizması iyi çalışmasına rağmen, yeterli olamamıştır ve bundan dolayı cihaza gönderilen sorgularda istenen cevaplar alınamamıştır. Ayrıca gerekli bilgi gelse bile bir standart hali olmadığından ağ yöneticisinin gelen bilgiyi nasıl yorumlayacağı konusunda bir netlik oluşmamıştır.

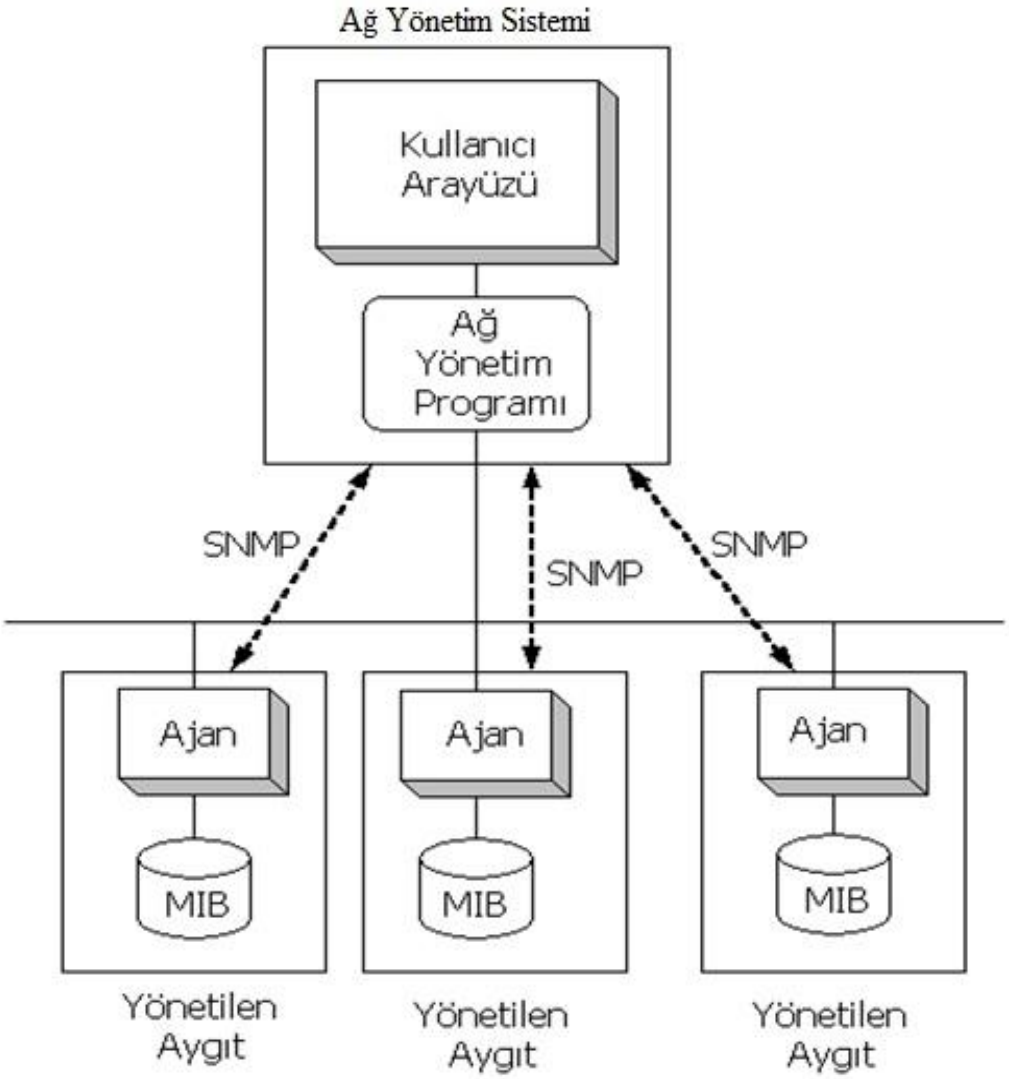
Bütün bunlara bağlı olarak ağ yönetiminin ihtiyaçlarını karşılamak için SNMP ile ilişkili birkaç öneri getirildi. Başlangıçta SNMP, SGMP (Simple Gateway Monitoring Protocol, Basit Geçit İzleme Protokolü) olarak duyuruldu. Farklı yaklaşımlar ve denemeler sonucunda da IAB (Internet Advisory Board) SNMP'yi çıkardı. Şekil 2.5'te SNMP'nin gelişim sürecini görmektesiniz [12].

Şekil 2.5. SNMP zaman çizelgesi

2.6.1. Mimari

SNMP mimarisi ağ yönetim sistemi, ajan ve yönetici kısmından oluşur (Şekil 2.6). Üç temel bileşenden oluşan bu mimaride, mimarinin en alt seviyesinde cihazdan istenilen veriyi çekmeyi sağlayan ajan yazılımı bulunur. Orta seviyede ise ağ yönetim sistemi ile ajan arasındaki iletişimi kuran yönetici kısmı bulunur. Mimarinin

en tepesinde ise tüm yönetim işlemlerinin yapıldığı ağ yönetim sistemi bulunur [11,13].



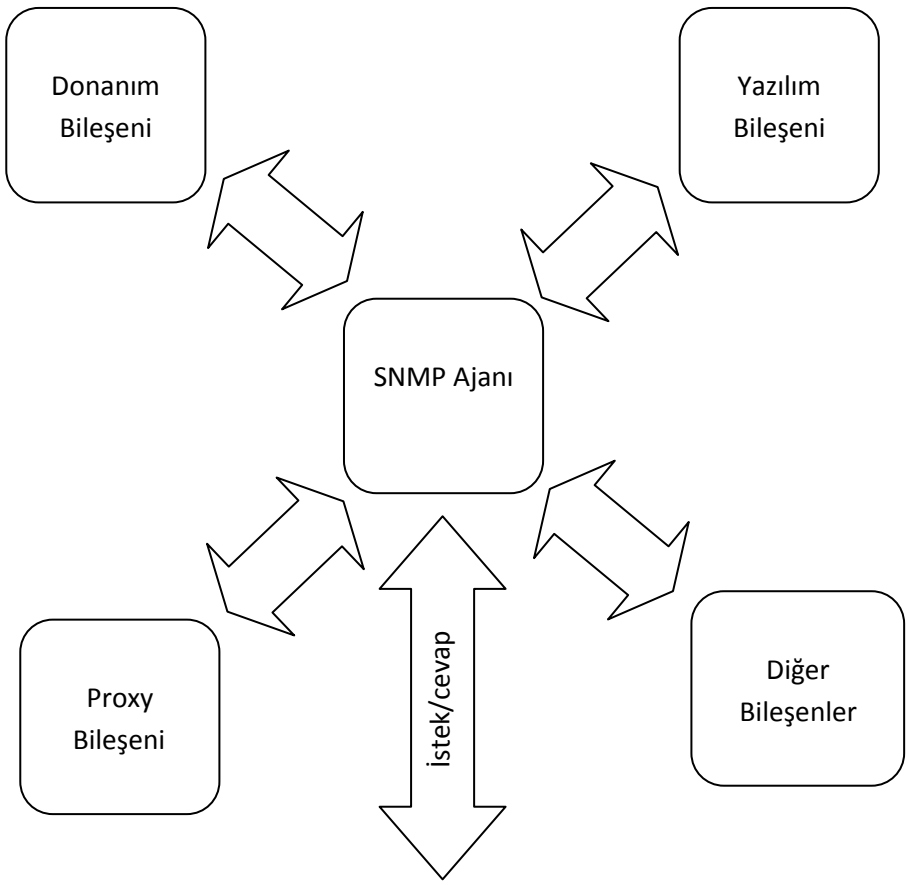
Şekil 2.6. Ağ yönetim sistemi

Bölümün bundan sonraki kısımlarında Şekil 2.6'da gösterilen mimarinin temel bileşenleri, bu bileşenlerin yapısı, birbirleriyle olan bağlantıları, SNMP dilleri ve veri çeşitleri açıklanmıştır.

2.6.1.1. SNMP ajanı (agent)

SNMP ajanı, kontrol veya takip edilen sistem düğümlerinden her birinde aktif edilen bir yazılımdır. Bu yazılım şekillendirmiş yapı içinde öğelerin her birine bir arayüz sağlar. Bu öğeler de yönetim bilgi tabanı dediğimiz MIB (Management Information Base, Yönetim Bilgi Tabanı)'lerde depolanır.

Cihaz, üzerindeki tüm SNMP iletişimini kontrol eder. SNMP ajanı aktif edilmeden önce sistemin ne tarz bir ajana ihtiyacı olduğu tespit edilir. Bu da SNMP sürümü ile alakalıdır. Gereksiz yere sistemden fazla veri çekmek trafiği artırmaya neden olacaktır [12,13]. SNMP ajanının yapısı Şekil 2.7'de gösterilmiştir. Şekilde de görüldüğü gibi üzerinde çalıştığı fiziksel veya yazılımsal tüm bileşenleri, takip eden bölümlerde anlatılacak olan yapılar sayesinde kontrol eder.



Şekil 2.7. SNMP ajanı

SNMP ajanının, yönetici ve ağ yönetim sistemi ile kendi aralarında düzgün iletişim kurabilmeleri ve bunun üretici firmadan firmaya değişiklik göstermemesi için SNMP dilleri ve veri çeşitleri gibi bazı kavramlar ortaya çıkmıştır. Bir sonraki bölümde SNMP dillerinden ve veri çeşitlerinden bahsedilmiştir.

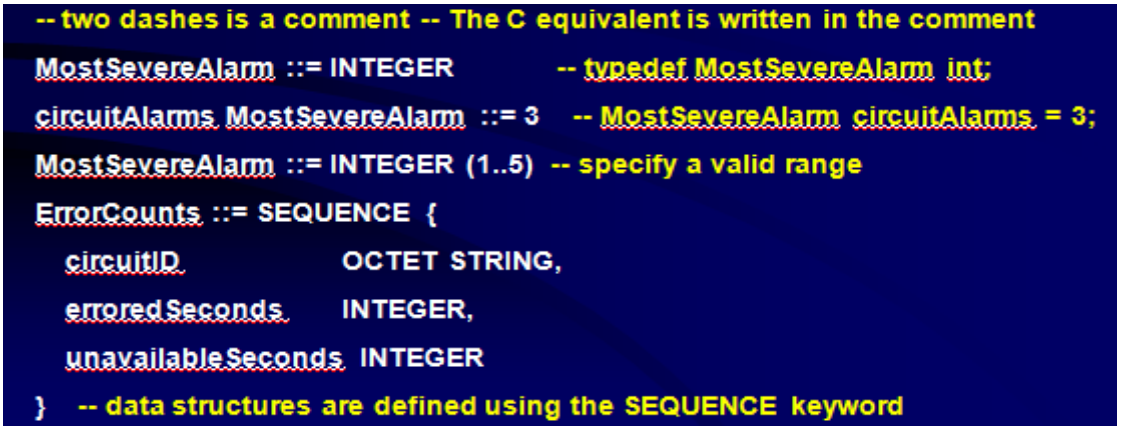
SNMP Dilleri ve Veri Çeşitleri

Tüm SNMP cihazları bir SNMP mesajını anlamak zorundadır. Bu mesajları anlama sırasında bazen sorunlar oluşabilir. İlk problem farklı programlama dillerinin farklı veri tiplerine sahip olmasından kaynaklanır. Mesela Java'da yazılmış SNMP yöneticisi, C dilinde yazılmış SNMP ajanın gönderdiği SNMP mesajlarını anlamayabilir. Bunun bir standart haline getirilmesi gerekmektedir ve bunun için SNMP dilleri ortaya atılmıştır. Böylelikle ajan yazılımının, yönetici yazılımının veya çekirdek kısım olan ağ yönetim sisteminin aynı programlama dilinde ve aynı platformlarda yazılma gereksinimi ortadan kalkmış olur [10]. Bu amaçla geliştirilmiş üç SNMP dili vardır. Bunlar:

a. ASN.1 (Abstract Syntax Notation One)
b. SMI (Structure of Management Information)
c. BER (Basic Encoding Rules)

ASN.1;

ASN.1 bir dil tanımlamasından daha fazlasıdır. C/C++ ve diğer programlama dillerine benzerlik gösterir. Şekil 2.8'de örnek bir ASN.1 sözdizimi gösterilmektedir.



Şekil 2.8. Örnek bir ASN.1 sözdizimi

SMI;

Yönetim Bilgi Yapısı (SMI) , SNMP tarafından manipule edilebilen temel bilgi tiplerini ifade eder. Yönetim verisinin hiyerarşisini ve basit formatını içeren bir iskelet sunar. SMI iki sürüme sahiptir:

SMIv1;

a. MIB modulleri CCITT X.208 ASN.1 veri açıklama dili ile tanımlanmıştır.
b. MIB'lerde kullanılan ASN.1 dilinin alt kümesidir.
c. Tüm ASN.1 yapıları CCITT X.209 BER kullanarak kabloda ilerler.

SMIv2**;** SMIv2, SMIv1'e geriye doğru uyumludur. Tek farklılık Counter64 tipidir.

BER;

ASN.1 ve BER arasındaki ilişki kaynak kod ve makine kodu arasında paralellik kurar.Tüm SNMP mesajları ASN.1'den daha küçük parçalara (BER) çevrilir. BER, her nesne için bir tanımlayıcı atar. Bu tanımlayıcı, her veri çeşidi için özel bir kod anlamına gelir.

SNMP veri çeşitleri;

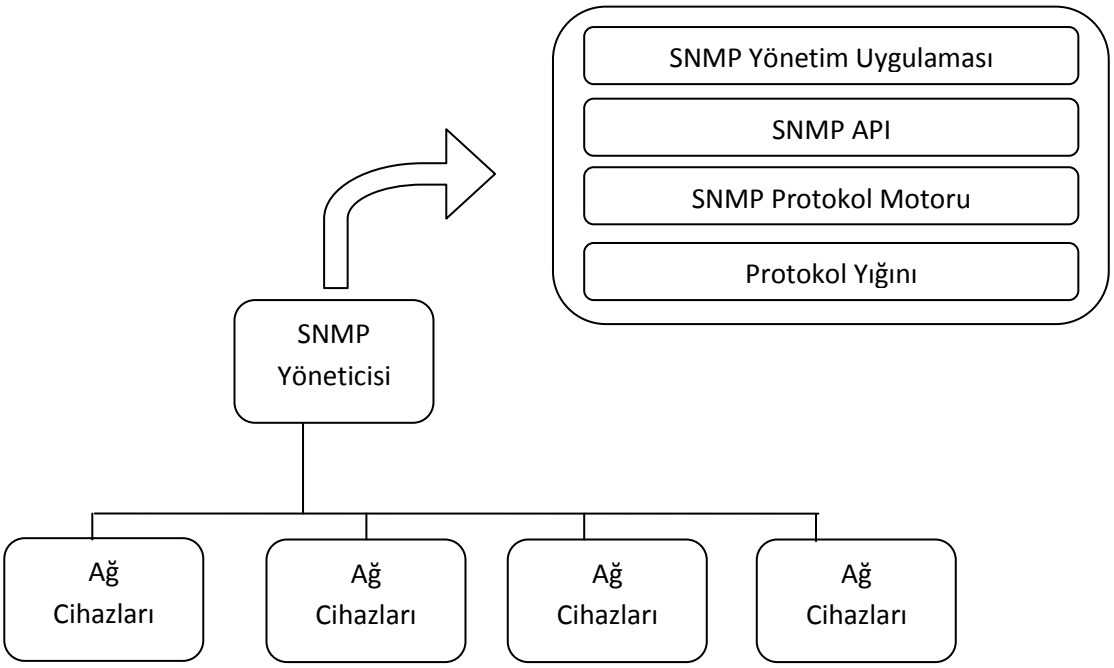
Bir önceki bölümde anlatılan SNMP dillerine bağlı olarak, bu dillerin kullandıkları bazı veri çeşitleri vardır. Bu veri çeşitlerinden en yaygın olanları:

a. Integer - işaretli 32 bit (tamsayı)
b. Octet String (metinsel veriler)
c. Object Identifier (OID, nesne tanımlayıcı)
d. Null - aslında veri çeşidi değil, veri değeridir.
e. IpAddress
f. Counter – işaretlenmemiş 32 bit (negatif olmayan sürekli artan değer)
g. Gauge – işaretlenmemiş 32 bit (artırılabilir veya azaltılabilir değer)
h. Timeticks – işaretlenmemiş 32 bit (son değişiklik zamanı)
i. Opaque (geriye dönük, uyumlu değer)

Bu çalışmada ilerleyen bölümlerde anlatılacak SNMP protokolünün üzerine oturduğu konu olan MIB II ile uyumlu çalışan SMIv2 dili kullanılmıştır. SNMP veri çeşitlerinden ise, “opaque, gauge, counter ve timeticks” hariç tüm veri çeşitleri kullanılmıştır.

2.6.1.2. SNMP yöneticisi (manager)

Ajan uygulamadan ihtiyaç duyulan bilgileri alıp kullanıcıya gösteren ve kullanıcının değiştirmek istediği değerleri cihaza gönderen yazılımdır. SNMP ajanına istek gönderir ve gerekli bildirimleri ve cevapları ajandan alır. SNMP yöneticisi istek gönderirken oturumlar açılır [12,13]. SNMP yöneticisinin yapısı Şekil 2.9’de gösterilmiştir.

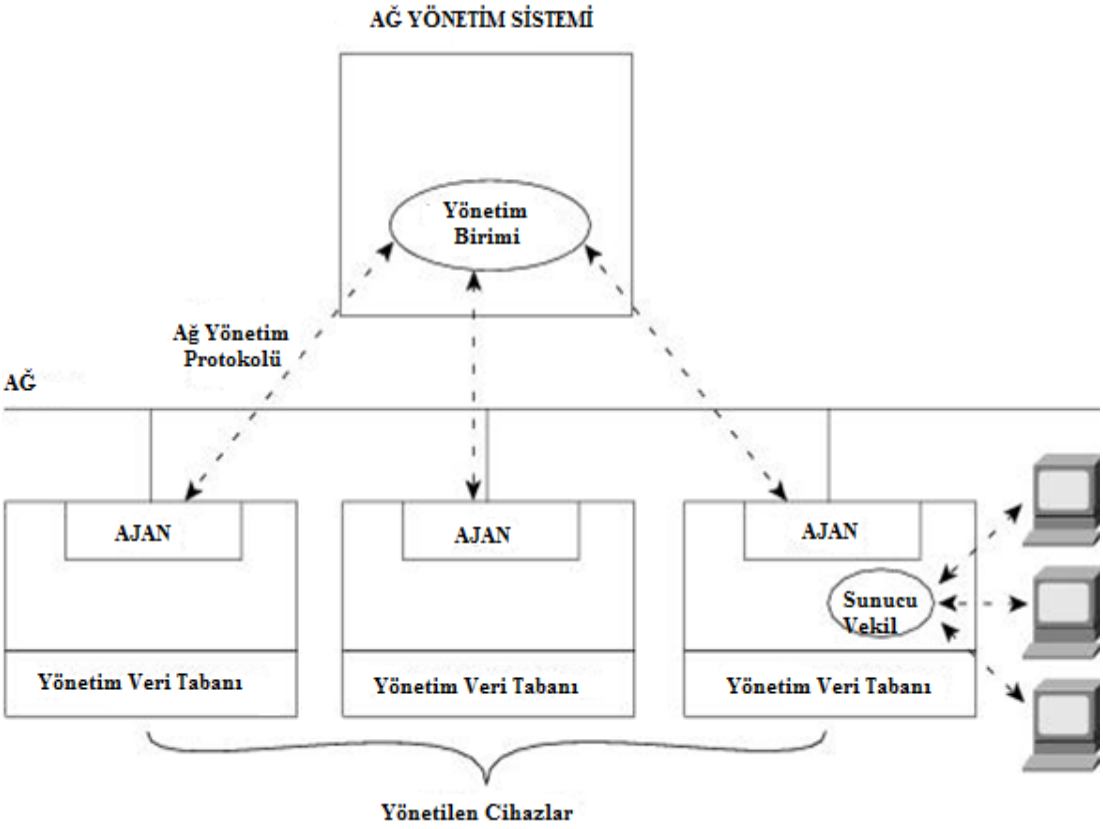


Şekil 2.9. SNMP yöneticisi

2.6.1.3. Ağ yönetim sistemi

Yönetici birimde çalışan ve bir ağa bağlı tüm cihazların izlenmesini ve yönetimini sağlayan uygulamaya verilen isimdir. SNMP ajanı ve SNMP yönetici arasındaki bilgi akışından iletişime kadar her türlü işlemi gerçekleştirir. Şekil 2.10'da ağ yönetim sisteminin yapısı gösterilmektedir [12,13].

Şu anda piyasada en çok kullanılan ağ yönetim sistemleri HP OpenView, Tivoli Netview, Advent Web NMS ve OpenNMS'dir. PRTG, MRTG, What's up ve Nagios'tur.



Şekil 2.10. Günümüzde kullanılan yaygın ağ mimarisi

Bu çalışmada, yukarıdaki mimari içerisinde kullanıcı arayüzüne bir program eklentisi yazarak, yönetilebilen sanal cihazlardan veri çekişi yapılmaktadır.

2.6.2. SNMP sürümleri

2.6.2.1. SNMPv1

İlk SNMP sürümüdür. UDP, IP ve IPX protokolleri üzerinde çalışabilir. Çalışma mantığında ise SNMP, özetle bir sorgu-cevap protokolü olduğu için bu işlem, Get, GetNext, Set ve Trap komutları aracılığıyla olmaktadır. Get, Ağ yönetim sistemi tarafından bir ya da daha fazla nesne bilgisi almak için kullanılır. Eğer yönetilen aygıt üzerinde çalışan ajan, istenen verilerin hepsini cevaplayamıyor ise ağ yönetim sistemine bir cevap yollamaz. Getnext işlemi tabloda yada ajan listesindeki bir sonraki değeri almak için kullanılır. Set işlemi ile yönetilen aygıtın MIB içerisindeki

değerleri değiştirilebilir. Trap işlemi ise ağ yönetim sistemine, yönetilen aygıt tarafından oluşan değişiklikleri bildirmek için kullanılır [13,14].

2.6.2.2. SNMPv2c

SNMPv2'ye, SNMPv1'in evrimleştirilmiş hali diyebiliriz. 1993'de çıkmış bir sürümdur. SNMPv2 ile bazı ek işlemler tanımlanmıştır. Get, GetNext ve Set işlemleri SNMPv1 ile aynı olmasına rağmen SNMPv2'de trap işlemi biraz daha farklıdır.

SNMPv2, v1'e göre iki yeni protokol işlemi daha içermektedir. GetBulk işlemi ile ağ yönetim sistemine büyük miktarda veri yollamak mümkündür. Eğer istenen veri bir paket boyutundan daha fazla ise ajan tarafından ard arda birkaç paket yollanır. Inform işlemi ise bir ağ yönetim sisteminin trap mesajlarını ağdaki başka bir ağ yönetim sistemine yollayabilmesi için kullanılır. SNMPv1'den farklı olarak eğer ajan yazılımı istenen değerlerin hepsini karşılayamıyorsa sisteme geri cevap döndürmemek yerine sadece sağlayabildiği mesajları gönderir [13,14].

2.6.2.3. SNMPv3

SNMPv3 önceki sürümlere göre güvenlik açısından daha gelişmiş olan bir sürümüdür. SNMPv3'te güvenlik düzeyi kavramı ortaya çıkmıştır. Bu düzeyler noAuthNoPriv (Kimlik denetimi ve şifreleme yok), authNoPriv (Kimlik denetimi var, şifreleme yok) ve authPriv (Kimlik denetimi ve şifreleme var) şeklindedir. Bu düzeylerin özellikleri şöyledir [14,15]:

a. noAuthNoPriv: v1 ve v2c'ye karşılık gelen güvenlik düzeyidir. Sadece kullanıcı adı bazlı şifreleme işlemleri yapar. Bu yapısından dolayı güvenli değildir.

b. authNoPriv: Önceki sürümlara göre daha üst seviyede bir güvenlik sağlar çünkü kimlik denetimini kullanıcı adı ve şifre bazlı yapmasının yanı sıra MD5 veya SHA algoritmalarını kullanarak veri bütünlüğü de sağlar.

c. authPriv: Uygulanması tavsiye edilen güvenlik düzeyidir çünkü bir önceki seviyeye ek olarak DES, 3DES ya da AES algoritmasını kullanarak sadece aynı anahtara sahip alıcıların çözebileceği bir şekilde veriyi şifreler.

SNMPv3 güvenlik modeli "authPriv" güvenlik düzeyinde kullanıldığında güvenliğin üç temel bileşeni olan kimlik denetimi, veri bütünlüğü ve gizliliği sağlar. Bu nedenle SNMPv3, IETF tarafından 2004 yılından itibaren güncel SNMP standardı olarak kabul edilmiş, önceki sürümler eski olarak nitelendirilmiştir. Tablo 2.8'de SNMP sürümlerinin karşılaştırmalı tablosu verilmiştir.

Tablo 2.8. SNMP sürümlerini karşılaştırma

	SNMPv1	**SNMPv2**	**SNMPv3**
Kullanılan diller	SMIv1	SMIv2	SMIv2
Protokol işlemleri	Get, GetNext, Set	Get, GetNext, GetBulk, Set	Get, GetNext, GetBulk, Set
Özellikler	İlk standart	SNMPv2-trap, genişletilmiş SMI, yöneticiler arasında iletişim	SNMPv2-trap, güçlü güvenlik desteği
Standart	1991	1999	1999'dan sonra
RFC no	1155,1212,1213,1215	1901~1908,2578~2580	2570~2576

Bu tez çalışmasında SNMPv3 sürümü kullanılarak uygulama geliştirilmiştir.

2.6.3. Paket yapısı ve temel SNMP komutları

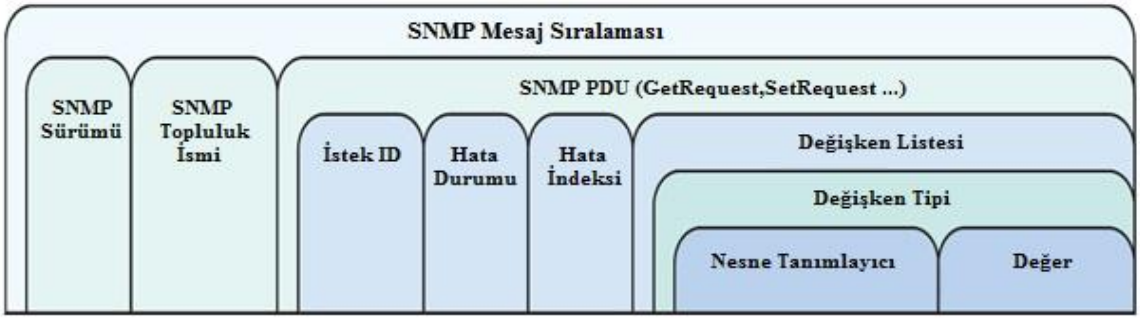
SNMP, paket yapısı açısından genel itibariyle iki yapıdan oluşur: Mesaj başlığı ve PDU (Protokol Data Unit). SNMP paket yapısının genel görünümü Şekil 2.11'de gösterilmiştir. Bu bölümlerin hangi alanlardan oluştukları detaylı bir şekilde Şekil 2.12'de, bu alanların boyutları ve ne iş yaptıkları ise Tablo 2.9'da anlatılmaktadır.

Şekil 2.11. SNMP paket yapısı [15]

SNMP mesaj başlığı

SNMP mesaj başlığı da 2 alan içerir**:** Sürüm numarası ve topluluk (community) ismi

- **Sürüm numarası**: Kullanılan SNMP sürümünü tanımlar**.**
- **Topluluk ismi:** Ağ yönetim sistemleri için erişim alanı tanımlar. Topluluk isimleri doğrulama (authentication) mekanizması gibi çalışır.



Şekil 2.12. SNMP mesaj sıralaması [16]

Tablo 2.9. SNMP mesajı alanları [16]

Alan	Tanımı	Boyutu
Snmp Mesajı Sıralaması	Snmp sürümünü, topluluk ismini ve Snmp PDU'sunu belirten Snmp mesaj sırasını belirtir.	2 byte
Snmp Sürümü	Hangi sürümün kullanıldığını belirtir. Şekil 2.11 ve 2.12'de gösterilmiştir.	3 byte
Snmp Topluluk İsmi	Snmp cihazlarına güvenlik ekleyebilmek ve onlara kolay erişebilmek için tanımlanan "octet string"dir. Şekil 2.11 ve 2.12'de gösterilmiştir.	8 byte

Tablo 2.9. SNMP mesajı alanları [16] (Devam)

Snmp PDU	Snmp mesajının ana bölümünü oluşturur. Farklı protokol veri birimlerini (PDU) tanımlar. Aşağıda PDU çeşitleri ve nasıl bir çerçeve yapısı kullanıldığı ve nasıl bir işlev sunduğu ayrıca anlatılacaktır	2 byte
İstek ID	Belirli Snmp isteklerini tanımlar. Bu index, Snmp yöneticisine uygun isteğe dönen cevapı eşletirme izni verir, Snmp ajan yazılımdan dönen cevabın yansıması gibidir.	3 byte
Hata Durumu	Snmp yöneticisinden gönderilen isteğe 0x00 değeri atanır. Sistemde bir hata varsa Snmp ajanı bu alanı değiştirir. 0x00—Hata yok 0x01—Dönen cevap aktarım için büyük. 0x02—İstenen nesne bulunamadı. 0x03—istekteki veri tipi, Snmp ajanındaki veri tipiyle eşleşmiyor. 0x04—Snmp yöneticisi sadece okuma parametresi atadı. 0x05—Genel Hata	3 byte
Hata İndeksi	Hata olursa, hataya neden olan nesne işaretlenir, diğer taraftan hata indeksi 0x00	3 byte
Değişken Listesi	Bu alanda SNMP PDU çeşidine göre veya uygulama alanına göre farklı değişkenler alabilir.	2 byte
Değişken Tipi	İki alandan oluşur. OID ve OID'nin değeri	2 byte
Nesne Tanımlayıcı (OID)	Snmp ajanındaki parametreleri ifade eder.	12 byte
Değer	SetRequest PDU – Değer, Snmp ajanındaki belirtilen OID'ye atanır. GetRequestPDU – Değeri boştur, dönen verinin izi gibi davranır. GetResponsePDU – Snmp ajanından belirtilen OID'den dönen değerdir.	2 byte

SNMP protokol veri birimi;

Şekil 2.13'de PDU'nun genel çerçeve yapısı verilmiştir. Bu çerçeve yapısı, PDU çeşidine göre farklılıklar gösterebilir.

Şekil 2.13. SNMP PDU yapısı [17]

PDU çeşitleri

a. GetRequest PDU
b. GetNextRequest PDU
c. SetRequest PDU
d. GetResponse PDU
e. Trap PDU

GetRequest PDU: Ağ yönetim sisteminin nesne tanımlayıcıları (OID) çekmek için ajan yazılımına gönderdiği PDU çeşididir.

GetNextRequest PDU: Belirtilen sonraki OID değerini almak için ağ yönetim sisteminden, ajan yazılıma gönderilen PDU çeşididir.

SetRequest PDU: Ağ yönetim sisteminden, ajana OID değerleri atamak için kullanılır.

GetResponse PDU: Ajandan, ağ yönetim sistemine gönderilen cevaplardır.

Trap PDU: Ajandan, ağ yönetim sisteminin seçili olan modülüne gönderilen bildirimlerdir. Cihazda hata varsa, hata bildirimin sadece ağ yönetim sisteminin hata yönetim modülüne gönderilmesi gibi.

Bu bölümde, yukarıda anlatılan PDU çeşitlerinin, hangi SNMP sürümü tarafından desteklendiği ve nasıl bir paket yapısına sahip olduğu Şekil 2.12'daki mesaj sıralaması ve Şekil 2.11'deki genel yapıya bağlı kalınarak gösterilecektir.

SNMPv1 için;

Get/GetNext/Set PDU ayynı paket yapısına sahip oldukları için Şekil 2.14'de aynı PDU çerçeve yapısında gösterilmiştir.

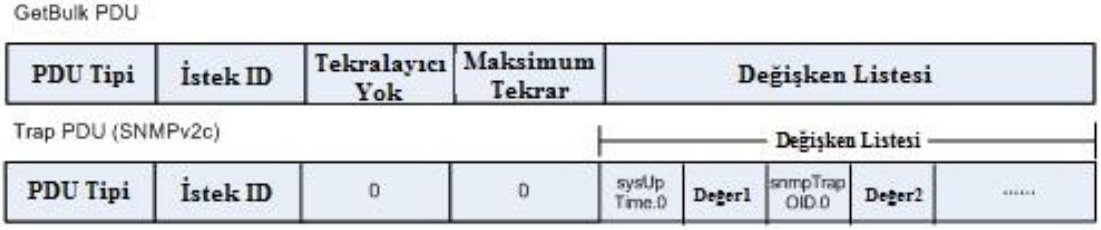
Şekil 2.14. SNMPv1 için PDU çeşitleri [17]

Şekil 2.14'deki PDU Tipi, İstek ID, Hata durumu gibi alanlar Tablo 2.9'da tanımlandığından dolayı burada sadece diğer alanlardan bahsedilecektir.

a. Cihaz Tipi: Trap üreten cihazın tipidir.
b. Ajan adresi: Trap üreten cihazın adresidir.
c. Genel trap: Cihazın başlatılmasında gerekli olan parametreleri içerir.
d. Özel trap: Üreticinin özel trap bilgisini içerir.
e. Geçen süre: Bir işlem yapılması için geçen süredir. Bunu için "sysUptime" kullanılır.

SNMPv2c için;

Şekil 2.14'deki PDU çeşitlerine ek olarak SNMPv2c'nin desteklemiş olduğu PDU çeşitleri şekil 2.15'de gösterilmiştir.



Şekil 2.15. SNMPv2c için PDU çeşitleri [17]

SNMPv3 için;

SNMPv3'ün paket yapısı diğer sürümlere göre değişiklik göstermektedir. Şekil 2.11' deki genel yapıya bağlı olarak güvenlik mekanizması nedeniyle paket yapısına güvenlik modeli, güvenlik parametreleri gibi yeni alanlar eklenir. SNMPv3'ün paket yapısı Şekil 2.16'da gösterilmektedir.

Şekil 2.16. SNMPv3 mesaj yapısı [17]

SNMPv3'ün PDU yapısı diğer sürümlerden farklılık göstermemektedir. SNMPv3'ün mesaj yapısındaki alanlar Tablo 2.10'da gösterilmiştir.

Tablo 2.10. SNMPv3 mesajı alanları [17]

Alan	**Tanımı**
Maksimum Boyut	Gönderilen maksimum mesaj boyutu
Bayraklar	0x0 - Doğrulama ve gizlilik yok 0x1 – Doğrulama var, gizlilik yok 0x3 – Doğrulama ve gizlilik var 0x4 – Bir tane PDU raporu gönderir.
Güvenlik Modeli	Güvenlik modelini ifade eder. 0 – Güvenlik modeli yok. 1-SNMPv1 güvenlik modeli 2-SNMPv2c güvenlik modeli 3-SNMPv3 güvenlik modeli
İçerik Motor ID	Her işlem için özgün bir SNMP girdisi tanımlar.
İçerik İsmi	İçerik ismi tanımlar. Her isim içeri motor ID ile eşleştirilmelidir.

Tablo 2.10. SNMPv3 mesajı alanları [17] (Devam)

Güvenlik Parametreleri	**Yetkilendirme Motor ID:** SNMP motorunu yetkilendirerek SNMP Motor ID'yi özeleştirir. **Yetkilendirme Motor Çalışması:** Yetkilendirilmiş Motor ID'sinin çalışmasını özeleştirir. **Yetkilendirme Motor Zamanı:** Bir zaman değeri atar. **Kullanıcı Adı:** Bir kullanıcı adı atar. Ağ yönetim sistemi ve ajan aynı isimde olmalı. **Doğrulama Parametresi:** Doğrulama mekanizması için bir anahtarlama kullanır. **Gizlilik Parametresi:** Gizlilik mekanizması için parametre kullanır. DES, AES gibi algoritmalar kullanılır.

Temel SNMP komutları;

Bir önceki bölümde anlatılan paket yapılarının ve PDU çeşitlerinin, SNMP ajanı, SNMP yöneticisi ve ağ yönetim sistemi arasında düzgün bir şekilde gönderilebilmesi için kullanılan bazı temel komutlar vardır. Bu temel komutlar bir önceki bölümde anlatılan PDU çeşitlerini yönetir [18].

a. Read komutu: Ağ yönetim sistemi tarafından yönetilen cihazları izlemek için kullanılır.
b. Write komutu: Ağ yönetim sistemi tarafından yönetilen cihazları kontrol etmek, üzerlerinde konfigürasyon yapmak için kullanılır.
c. TRAP: Ajanda bir değişiklik olması durumunda ağ yönetim sistemini bilgilendirmek için kullanılır. Diğer komutların aksine burada ilk SNMP paketini ajan yollar. Trap PDU'sunu yönetir.
d. GET: Belirtilen değişkenin değerini sorgular. GetResponse ve GetRequest PDU'larını yönetir.
e. GETNEXT: Belirtilen değişkenden sonraki değişkenin değerini sorgular. GetResponse ve GetNextRequest PDU'larını yönetir.
f. GETBULK: Sürüm 2 ile gelen bir yeniliktir. Bir seferde birden fazla değişkeni kolay bir şekilde sorgulamaya yarar.
g. SET: Belirtilen değişkenin değerini değiştirmek için kullanılır. GetResponse ve SetRequest PDU'larını yönetir.

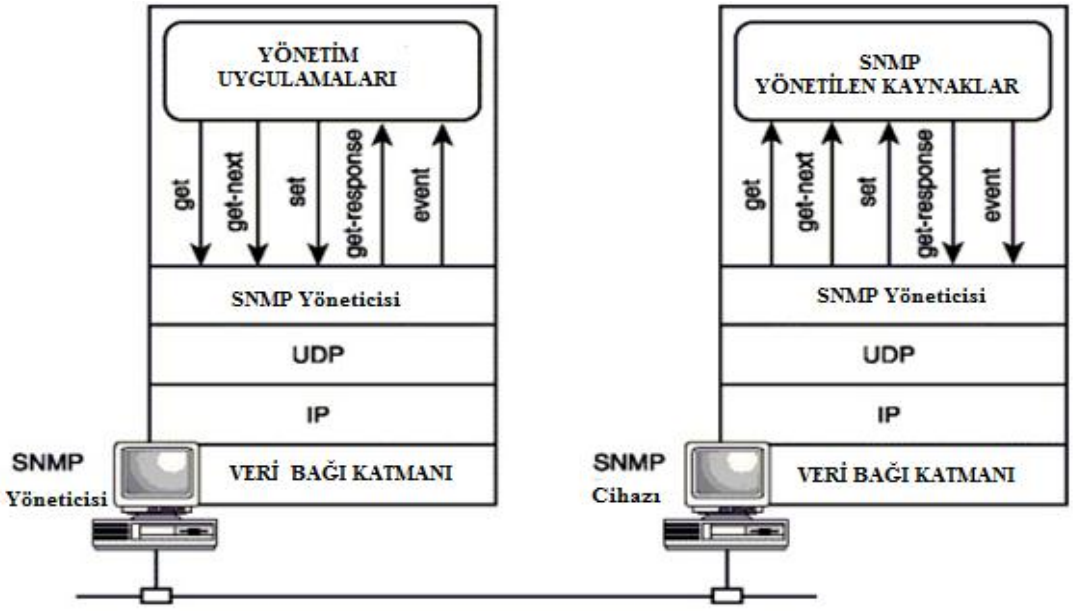
h. INFORM: Sürüm 2 ile gelen bir yeniliktir. TRAP'ten farklı olarak bu komut bir bildirim yaptığında ağ yönetim sisteminden bir onay paketi bekler. Onay paketi gelmezse tekrar INFORM gönderilir.

Bu çalışmada, "read", "get" ve "getnext" komutları kullanılmaktadır.

Çalışma Mekanizması;

SNMP'nin çalışma mekanizması istek gönderme ve isteğe cevap alma şeklindedir ve bunun için taşıma katmanında kullandığı protokol UDP'dir. Ağ yönetim sistemi, istekleri herhangi bir portundan, ajanın 161. portuna gönderir. Ajan geri bildirim için kendisine gelen istekleri 162. portundan gönderdiği cevaplarla sağlar. SNMP ajan yazılımı, cihazda herhangi fiziksel bir sorun oluştuğunda (cihaz üzerindeki fiziksel değerlere atanan değerlerin üzerine çıkıldığı zaman veya periyodik olarak veri gönderimi yapmak için ayarlandığı zaman) iletişimini kendi başlatır [17].

SNMP sayesinde bir cihazdan bilgi alınabileceği gibi, cihazdaki bilgi değiştirilebilir ve cihazda yeni bir yapılandırma uygulanabilir. Örneğin cihaz baştan başlatılabilir, cihaza bir yapılandırma dosyası gönderilebilir ya da cihazdan alınabilir.



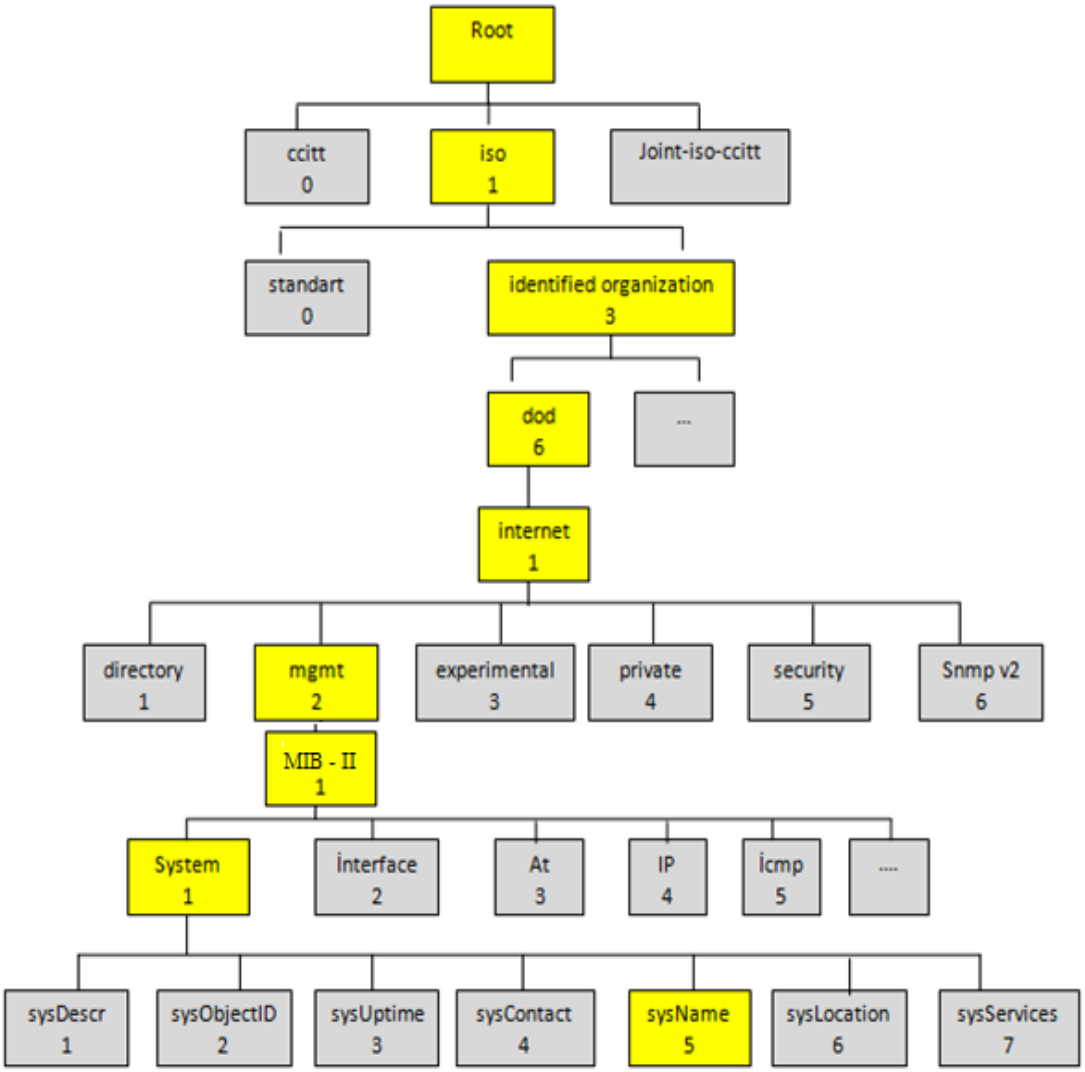
Şekil 2.17. SNMP çalışma yapısı

Şekil 2.17'de, bir önceki bölümde anlatılan SNMP paket yapısı, PDU çeşitleri ve SNMP komutlarına göre bir çalışma yapısı düzenlenmiş olup, SNMP cihazından (ajandan) SNMP yöneticisine veri gönderimi katmansal yapı çerçevesinde gösterilmiştir.

2.6.4. MIB kavramı

MIB kavramı bir ağaç yapısına benzetilebilir. Ulaşılmak istenen değeri tutan değişkene OID (Object IDentifier, Nesne Tanımlayıcısı) adı verilir. MIB yapısındaki sıralamaya göre değer alır. Her kuruluşun, "**Internet Engineering Task Force *(*IETF*)***" tarafından atanan bir değeri vardır, yani belirli bir yere kadar ağaç yapısı evrenseldir, ancak kurumların kendi kullanacakları yönetim nesneleri için bu kodu her kurum kendi tanımlar [13]. Bu değişkenler ağacın dallarının en uç noktasında olup bir cihazla ilgili tek bir değeri tutabileceği gibi kendisinden sonra gelen bütün alt dalları ifade etmek için de kullanılabilir. Kökten ağaç dalına uzanan bu hiyerarşi birbirlerinden nokta ile ayrılmış sayı dizileriyle ifade edilir. Yönetilen nesneler köke bağlı mantıksal gruplar şeklinde öbeklenir [13,14].

MIB'i, SNMP ağ cihazlarının veri nesnelerinin tanımlandığı bir ASCII metin belgesi olarak da tanımlayabiliriz. Mesela SNMP sözlüğü gibi düşünebiliriz. Bu sözlükte her SNMP nesnesinin karşılığı rakamsal olarak tutulur. SNMP cihazı bir bildirimde bulunduğu zaman her veri nesnesini OID'lerle tanımlar.



Şekil 2.18. MIB değerleri ağaç yapısı

Şekil 2.18'de, OID değeri "1.3.6.1.2.1.1.5" olan "sysName" değeri ağaç yapısında gösterilmiştir. Buradaki ilk girdi de sysName.0 olarak adlandırılır. Yani komutta 1.3.6.1.2.1.1.5.0 yerine sysName.0 yazılırsa da aynı işlevi görür. Değişkenin başındaki ilk dört sayı, yani 1.3.6.1 standarttır. Bu noktadan sonra ulaşmak istediğimiz bilgiye göre alt dallara ilerlenir. Örneğin 1.3.6.1.2.1.1 dalı sistemle ilgili sistem adı, sistem tanımı, sistemin ayakta olduğu süre gibi değerleri tutar. Bunun alt dalı olan 1.3.6.1.2.1.1.5.0 değişkeni bunlardan biridir (sistem adı). Şekil 2.18'deki ağaç yapısındaki dalların detaylı açıklaması aşağıdaki gibidir:

a : ISO (International Standart Organization)

b : Org (organization)

c : Dod (Department of defense)

d : Internet

e : Mgmt (Network management entries)

f : SNMP MIB-2

g : system

h : sysName

ı : Dalın sonundaki ilk girdiyi belirtir.

Sistem dalı altındaki diğer değerler ise aşağıdaki gibidir:

1.3.6.1.2.1.1.1 – sysDescr (Cihaz tanımlaması için)

1.3.6.1.2.1.1.2 – sysObjectID (Cihaz ID'si için)

1.3.6.1.2.1.1.3 – sysUpTime (Cihazın ne zaman güncellendiği)

1.3.6.1.2.1.1.4 – sysContact (Cihaz iletişim bilgileri)

1.3.6.1.2.1.1.5 – sysName (Cihaz adı)

1.3.6.1.2.1.1.6 – sysLocation (Cihazın fiziksel yeri)

1.3.6.1.2.1.1.7 – sysServices (Cihaz tipini belirler)

Artan ağ gereksinimlerini karşılayabilmek için MIB değerleri zaman içerisinde üretici firmalar tarafından arttırılmıştır. Daha efektif ağ yönetimi sağlayabilmek için sahadaki cihazlardan daha fazla bilgi çekmek gerekir. İhtiyaç duyulan bu veriler için yeni MIB değerleri ortaya çıkarılmıştır. IETF (Internet Engineering Task Force) tarafından bu MIB değerlerini düzenlemek için farklı RFC (request for comments) dökümanları tanımlanmıştır. Bu MIB değerleri iki ana başlık altında toplanmıştır:

2.6.4.1. MIB v1

Hem hata yönetimi hem de konfigürasyon yönetimi için geliştirilmiş olan MIB v1 için sadece bazı kontrol nesneleri tanımlanmıştır, bu yüzden gelişmiş ağlar için yeterli değildir. MIB v1'de en fazla 100 veri çekişine izin verilmekte olup gereksiz veri çekmeyi engeller.

MIB v1 aşağıdaki alanlarla ilgili veri çekebilir [13]:

a. Sistem
b. Arayüzler
c. Adres çevrimi
d. IP
e. ICMP
f. TCP
g. UDP
h. EGP

2.6.4.2. MIB v2

MIB v1'den farklı olarak; yeni işlemsel gereksinimler için eklentiler sunar. Bunlar SMI/MIB ve SNMP ile ileriye doğru uyumluluk sağlama, çoklu protokolleri destekleyen cihazları destekleme ve açıklık, okunabilirlik için daha anlaşılabilir bir yapı sunmak şeklinde özetlenebilir.

MIB v2 aşağıdaki alanlarla ilgili veri çekebilir [14]:

a. Sistem
b. Fiziksel adresler
c. Arayüzler
d. Adres çevrimleri
e. IP
f. ICMP
g. TCP
h. UDP

i. EGP
j. İletişim
k. SNMP

Bu çalışmada,“System”, “Interface”, “At”, ve “IP” ana başlıkları altındaki MIB II değerleri kullanılmaktadır.

2.7.Sonuç

Bu bölümde anlatılan konu başlıkları içerisinde, kendi uygulamamız için tercih ettiğimiz yöntemler aşağıda verilmiştir:

a. Uygulama, konfigürasyon yönetimi, güvenlik yönetimi ve performans yönetimi konularını içermektedir.
b. Uygulamada, merkezi yönetim mimarisini temel alan örnek bir kurumsal ağ topolojisi tasarlanmıştır.
c. Uygulamada MIB II ile uyumlu çalışan SMIv2 dilini tercih edilmiştir. SNMP veri çeşitlerinden ise, “opaque,gauge,counter ve timeticks” hariç tüm veri çeşitleri kullanlmıştır. Ayrıca uygulamada “read”, “get” ve “getnext” komutları kullanılmıştır.
d. Bu çalışmada, Şekil 2.6’daki mimari içerisinde kullanıcı arayüzüne bir program eklentisi yazarak, yönetilebilen sanal cihazlardan veri çekişi yapılmaktadır. Bu işlemler için, SNMP’nin güvenli sürümü olan SNMPv3 seçilerek, “System”, “Interface”, “At”, ve “IP” başlıkları altındaki MIB II değerlerleri kullanılmıştır.

BÖLÜM 3.

UYGULAMADA KULLANILAN TEKNOLOJİ BİLEŞENLERİ

3.1. Giriş

Bu çalışmada VMWARE Workstation, GNS3 ve Wireshark programları kullanılarak örnek bir kurumsal ağ topolojisi modellenmiştir. Kurumsal ağ kavramı ve modelleme ile ilgili bilgileri vermeden önce bu teknoloji bileşenlerinden bahsedilecektir. VMWARE Workstation ve GNS3 de sanal programlar olduğu için öncelikle sanallaştırma kavramının ne olduğu ve ne gibi faydaları olduğu gibi konular ele alınacak daha sonrasında ise bu sanal programlarla ilgili bilgiler verilecektir.

3.2. Sanallaştırma

Sanallaştırmayı kısaca tanımlamak gerekirse; mevcut bulunan fiziksel donanımın sanal makineler (virtual machines) yardımıyla çok daha verimli kullanılabilmesini sağlayan, çeşitli yazılım ve donanım bağımlılıklarını ortadan kaldıran, bu sayede de yeni ürün ve servis geliştirme maliyetlerinde büyük tasarruflar sağlayan bir yazılım çözümüdür. Çözümde kullanılan sanal makinenin tanımını ilk defa Popek ve Goldberg *"gerçek makinenin etkili, soyutlanmış bir kopyasıydı"* şeklinde yapmıştır [19].

Günümüzde birçok kurum, kendi kurumsal alt yapılarını geliştirebilmek için ve daha iyi yönetebilmek için sürekli kendilerini geliştirmek zorundadır. Bu gelişimleri yapabilmeleri için daha önceden kullandıkları sistemleri revize etmeleri veya yeni bir sisteme geçmeleri gerekmektedir. Her iki durumda da ortaya çıkan maliyet ve zaman kaybı gibi sonuçlar kurumlar için büyük kayıp anlamına gelmektedir. İşte bu nedenlerden ötürü kurumlar gerçek ortamda yapacakları atılımları önce sanal ortamlarda deneyerek alınan performans sonuçlarına göre karar verirler. Bu yüzden

sanallaştırmanın kullanıldığı birçok alan oluşmuştur. Bu alanlar Şekil 3.1'de gösterilmiştir.

Uygulamamızda, teknoloji bileşenleri (araçları) olarak VMWARE Workstation, GNS3 (Graphical Network Simulator 3) ve Wireshark programları kullanılmıştır. Bunlardan VMWARE Workstation ve GNS3 sanallaştırma platformu programlarıdır. Wireshark ise bir ağ analiz (paket yakalama) programıdır.

Şekil 3.1. Sanallaştırma alanları

3.2.1. Sanallaştırmanın avantajları

Sanallaştırma işlemi ve sanal makinelerin kullanımı küçük, orta veya büyük, her ölçekte firmalar için oldukça önemli avantajlar sağlar. Bunlardan bazıları [20,21]:

a. Sunucu kapasitesini yüksek verimle kullanma imkanı sağlar. Kapasitenin gereksiz yere kullanılmasını ve karmaşıklığı engeller.
b. İhtiyaç duyulduğunda sanal olarak çok hızlı bir şekilde yeni sunucu oluşturulabilir.
c. Kurulan sistemin büyüklüğüne göre donanım maliyetlerinde %50'ye varan düşüş sağlanır.
d. Operasyonel kurulum ve bakım maliyetlerinde %80'e varan azalma görülür.

e. Herhangi bir problem anında, sanal sunucuları çok hızlı bir şekilde yeniden çalışır duruma getirebilme imkanı vardır.
f. Merkezi yönetim ile tüm sunucuları tek bir merkezden izleme ve raporlama imkânı sağlanır.
g. Çalışan sistemlerdeki değişiklikleri, sistemleri tamamen durdurmadan taşıma ve müdahele edilmesini sağlar.

3.2.2. Sanallaştırma için kullanılan yazılımlar

Sanallaştırma için kullanılan sanal makine yazılımlarını iki ana başlık altında toplayabiliriz [19]:

a. Açık Kaynak Kodlu Sanal Makine Yazılımlar: VirtualBox, Xen, Bochs, CoLinux, FAUmachine, Hercules emulator , KVM, LilyVM, QEMU, SheepShaver

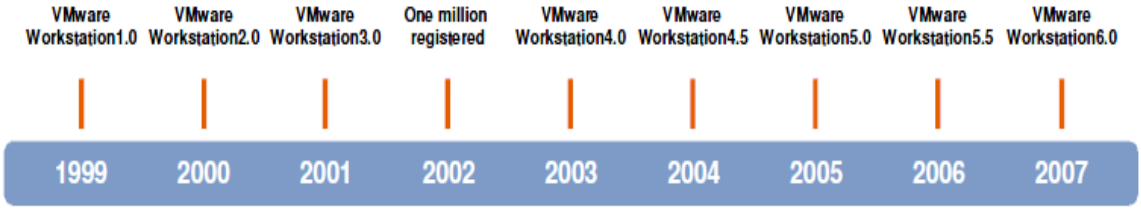
b. Ticari Sanal Makine Yazılımları: VMware, Microsoft Virtual PC, VM/CMS, Parallels Workstation, vThere, Parallels Desktop for Mac, SVISTA, Trango, Virtual Iron Software

3.3. VMWARE Workstation

Sanal makine çözümleri incelediğinde küçük, orta ve büyük her seviyeden şirket ve organizasyonun sanallaştırma ihtiyaçlarını en iyi karşılayabilecek çözümlerin başında VMware ticari çözümleri olduğu görülmektedir. VMware şirketi 1998 yılında ABD merkezli olarak kurulmuştur. İlk çözümlerini pazara 1999 yılında çıkaran şirket şu an pazarı yönlendiren birkaç kuruluştan birisi olarak göze çarpmaktadır [19].

VMware, bir sanal makine yazılımı olup konuk işletim sisteminin ana makine işletim sistemi içinde çalışmasına izin verir. VMware ile konuk işletim sistemini aktif hale getirdikten sonra bu konuk işletim sistemine uygulama programları yükleyebilir ve onun desteklediği servisleri verebiliriz. Sanal bilgisayarımızla asıl gerçek

bilgisayarımız arasında internet, ağ ve dosya paylaşımı yapabiliriz. Kullandığı donanımlar sanal olduğu için makine üzerine işletim sisteminizi kurarken verdiğimiz özelliklere bağlı olarak normalde normal bilgisayardan daha hızlı çalışabilir. Şekil 3.2'de, WMWARE programının gelişim süreci gösterilmiştir.



Şekil 3.2. WMWARE Workstation gelişim süreci

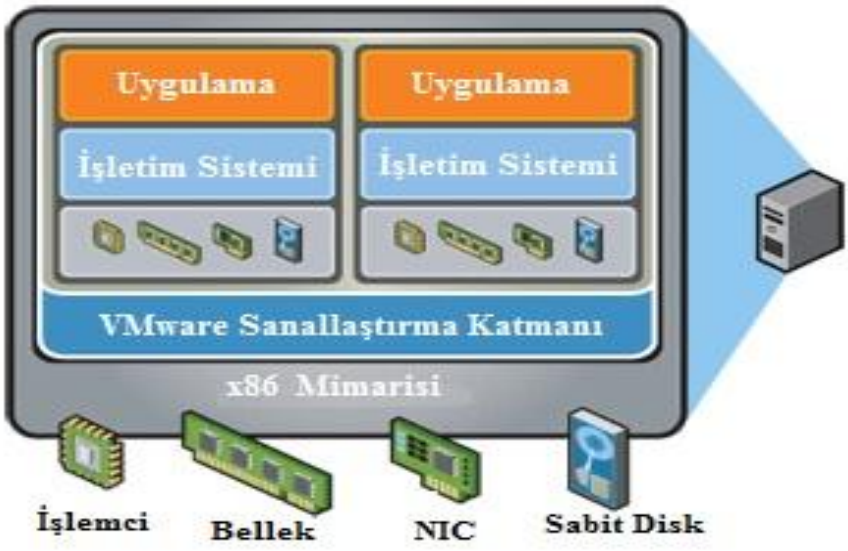
3.3.1. VMWARE katmanlı yapısı ve çalışma mimarisi

Vmware'ın katmanlı yapısına geçmeden önce Vmware terminolojisi olarak şu iki kavramı bilmemiz gerekir [21]:

a. Kaynak İşletim Sistemi (Host Operating System): Vmware Workstation'un çalıştığı ve kendi bilgisayarınızın üzerinde kurulu olan işletim sistemidir.

b. Konuk İşletim Sistemi (Quest Operating System): Kaynak işletim sistemi üzerinde kurulu olan sanal işletim sistemleridir. Sanal makine olarak da adlandırılabilirler.

Şekil 3.3'deki katmanlı yapıya bakıldığı zaman ise en alt katmanda hard disk, bellek, ethernet kartı ve işlemci gibi bazı fiziksel bileşenler görülmektedir. Bir üst katmanda sunucu işletim sistemi bulunmaktadır. Daha sonraki katmanda VMWARE sanallaştırma katmanı yer alır ve artık bu sanallaştırma katmanı üzerine sanal makinelerimizin kurulu olduğu konuk işletim sistemleri eklenebilir. En son katmanda ise sanal makineler üzerinde çalışan uygulamalar görülmektedir.



Şekil 3.3. VWMARE katmanlı yapısı [21]

VMWARE Workstation sanallaştırma programının çalışma yapısını maddeler halinde açıklayabiliriz:

a. Workstation, Linux veya Windows çalışan bir sunucu makine üzerine kurulur ve bir uygulama gibi çalışır.

b. İşletim sistemi ve ilgili uygulamalar daha sonradan Workstation üzerinde çalışacak olan bir sanal makine gibi kapsüllenir.

c. Her sanal makinenin kendine ait CPU'su, hafızası, disk alanı, I/O cihazları bulunur.

d. Her sanal makine fiziksel bir x86 makinesi gibi çalışır.

VMWARE Workstation'da, sanal makine kullanmanın şu avantajları vardır [21]:

a. İzolasyon (Isolotion): Her konuk işletim sistemi, kaynak makineden ve diğer konuk işletim sistemlerinden ayrılmıştır. sunucu makine ve diğer konuk işletim sistemlerinden ayrılmıştır.
b. Verim (Efficiency): Dual-boot'u ve hard disk sürücülerini tekrar tekrar bölümlemeye gerek yoktur.
c. Kapsülleme (Encapsulation): Her sanal makine taşınabilirdir.

d. Gürbüz Ağ İletişimi (Robust Networking): Karmaşık ağ topolojileri çok rahatlıkla oluşturulabilir.
e. Uygunluk (Compatibility): Sanallaştırma donanım uyumsuzluklarını ortadan kaldırır.

3.3.2. VMWARE Workstation için gerekli donanım

VMWARE programını kurabilmek için gereken en düşük fiziksel gereksinim aşağıdaki gibidir [22]:

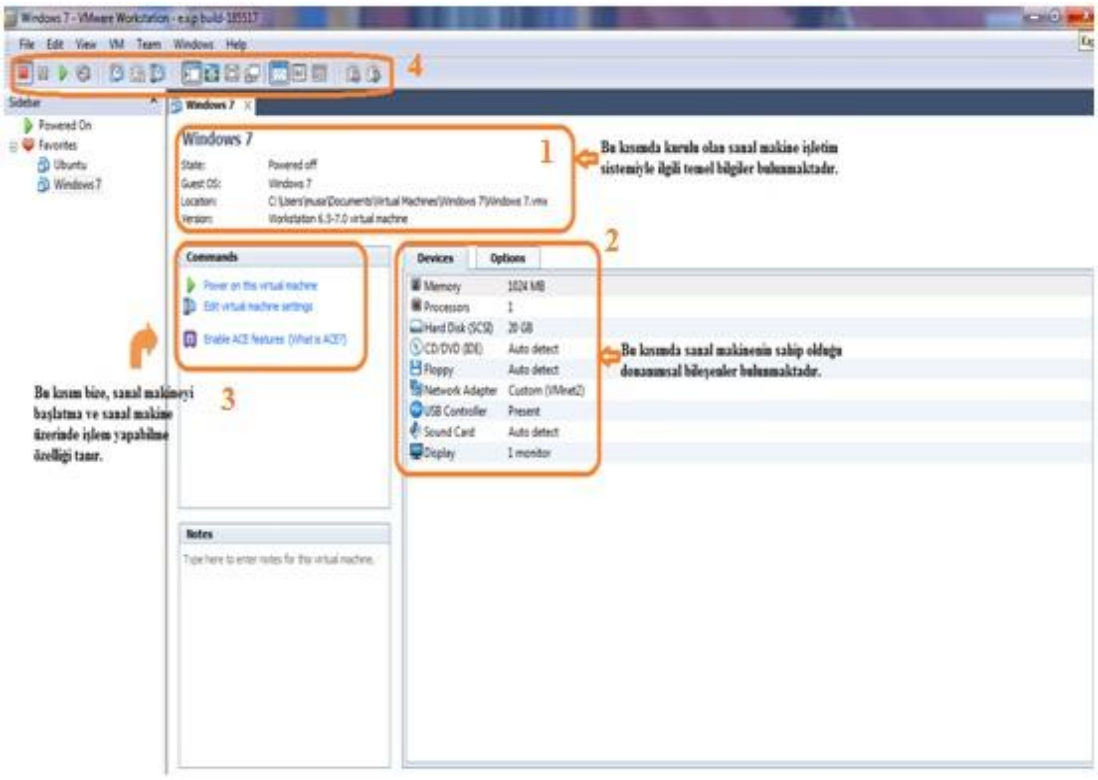
a. PC donanımı için:
   1. Standart x86 veya x86-64 uyumlu bir pc
   2. En düşük 1.3 ghz hızda işlemci
   3. Intel – Pentium 4, Pentium M (with PAE), Core, Core 2, Core i3, Core i5, and Core i7
   4. AMD – Athlon, Athlon MP, Athlon XP, Athlon 64, Athlon X2, Duron, Opteron, Turion X2, Turion 64, Sempron, Phenom, and Phenom II
   5. Çoklu işlemciler desteklenir.
   6. Kurulacak sanal işletim sistemi 64 bitlik ise ancak 64 bitlik işlemci üzerine kurulabilir.

b. Hafıza (memory) için:
   1. Her kurulacak konuk işletim sistemi için minumum 1 gb alan gereklidir.
   2. Her sanal makine için verilebilecek en yüksek hafıza 8 gbdir (32 bitlik makineler için).

c. Sabit diskler (hard disk) için:
   1. Vmware her işletim sistemi için en az 1 gblık alan önerir. Temel kurulum için Linux için 200 mb, windows için ise 1,5 gblık alan gereklidir.

Şekil 3.4'te, VMWARE Workstation'un, tezin EK-A bölümünde anlatılan kurulum bilgilerinden sonraki genel görünümü gösterilmiştir. Şekil 3.4'te 1 nolu kısım kurulu

olan konuk işletim sistemi ile ilgili bilgileri verirken, 2 nolu kısımda ise kurulu olan sanal makinenin donanımsal özellikleriyle ilgili bilgiler verilmektedir. 3 ve 4 nolu kısımlarda ise sırasıyla sanal makineyi başlatma düğmeleri ve menü alanı gösterilmektedir.



Şekil 3.4. VMWARE Workstation genel görünüm

VMWARE firmasının Workstation haricinde başka ürünleri de vardır. Bunlardan bazıları aşağıda verilmiştir:

a. VMWARE ESX Server
b. VMWARE Player
c. WMWARE Server
d. WMWARE View
e. VMWARE ACE
f. VMWARE Lab Manager
g. VMWARE Lifecycle Manager
h. VMWARE Converter

Bu ürünlerden en çok kullanılanları piyasada bir çok kurumun tercih ettiği VWARE ESX Server ve VMWARE Server'dır. VMWARE ESX Server, farklı bir sunucu makineye kurularak birden fazla konuk işletim sistemini, sunucu makine üzerinden yönetebilen VMWARE ürünüdür. VMWARE Server ise, kaynak işletim sistemi üzerinde Windows Server 2008 gibi sadece sunucu işletim sistemleri olduğu sürece kurulabilen ve sunucu özellikleri sunan bir VMWARE ürünüdür. WMWARE Player ve View ürünleri daha önceden kurulu olan sanal makineyi çalıştırmayı sağlayan ürünlerdir.

3.2.3. VMWARE Workstation'da sanal ağ kavramı

Güçlü ağ yapısı özelliği sayesinde karmaşık ağ yapılarını gerçekleyebilme olanağı sağladığından VMWARE Workstation, birçok akademik çalışmada tercih nedeni olmuştur.

Normalde Vmware Workstation, sunucu makine üzerinde 44thernet kartı veya denetçisi bulunmasını şart koşar. Dış dünyaya (Wan, internet veya diğer bir ağa) çıkabilmek için sunucu makinenin 44thernet kartını kullanır. VMWARE Worktation bu sanal ağ özelliği ile, Windows makinelerde 10 tane sanal 44thernet çıkışı, Linux makinelerde ise 255 tane sanal 44thernet çıkışına izin vermektedir. Her iki işletim sistemi içinde 3 tane sanal 44thernet çıkışı "bridged", "host-only" ve "NAT" işlemleri için ayrılmıştır [22]. Bu kavramları sırasıyla açıklayalım:

Bridged Networking;

Kaynak makine bir ağ üzerindeyse ve konuk (sanal) makinenin de bu ağa erişmesini istiyorsak bu ağ tipi kullanılır. Sanal makine aynı ağ üzerinde ayrı bir kaynak makine olarak gözükür. Böylece diğer kaynak ve sanal makineler tarafından görülür ve kaynaklarına ulaşılır. Yine sanal makine ağ üzerindeki yazıcı ve diğer kaynaklara ulaşabilir. Bu ağ tipini oluşturabilmek için kaynak üzerinde fiziksel network kartı bulunmalıdır [22].

Host-Only Networking;

Kaynak makine ve üzerine kurulan sanal makineler arasında kurulan ağ tipidir. Sanal makine oluşturulan ağa, sanal ağ kartı veya anahtar cihaz yoluyla bağlanır. Kullanılan protokolün TCP/IP olması gerekmez, TCP/IP dışındaki protokoller de kullanılabilir. Aynı kaynak üzerine kurulan diğer sanal makineler de birbirleriyle ve kaynak ile iletişim içerisindedir. Kaynak makine üzerine kurulan sanal makineler sadece kaynak üzerinde gözükür, ağ üzerinde gözükmezler [22].

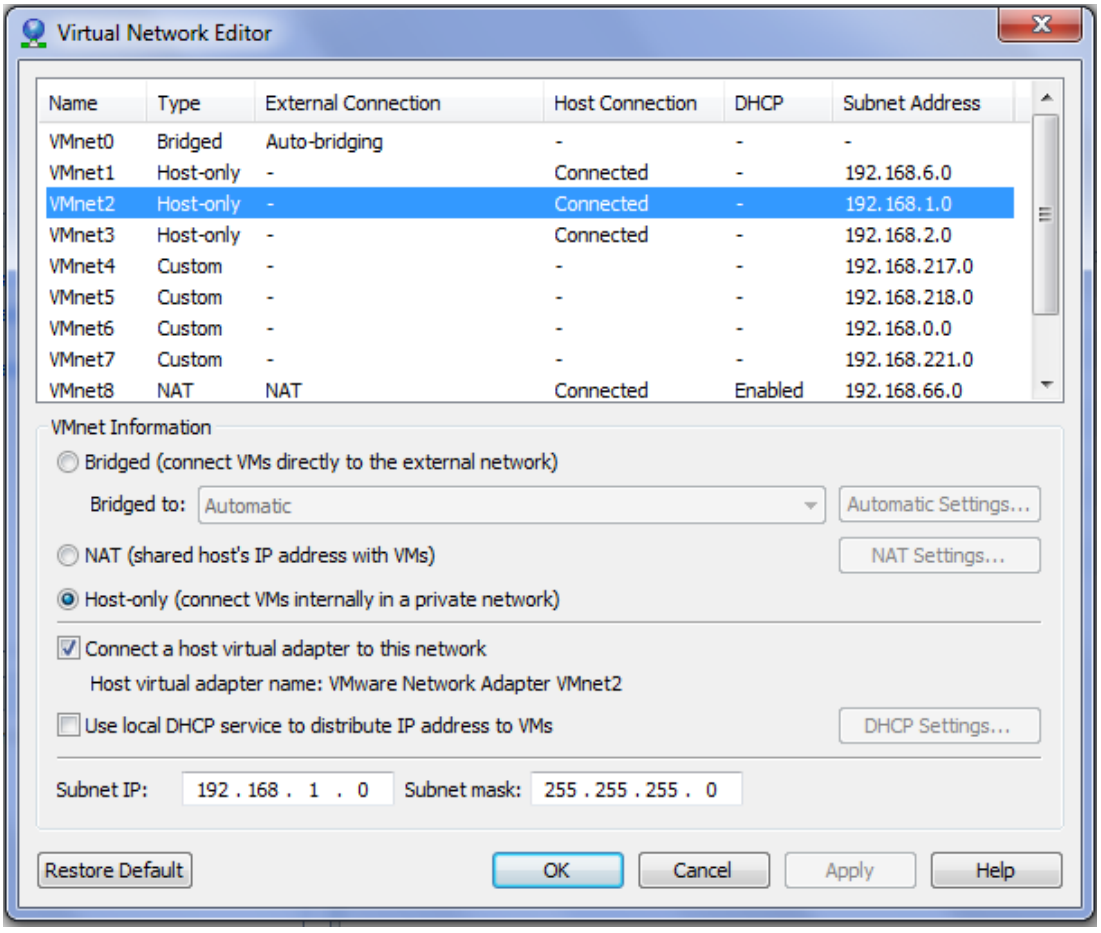
Bu ağ tipini oluşturabilmek için kaynak makine üzerinde ağ kartı olması gerekmez. Kaynak üzerine kurulan sanal makinelerin her biri için sanal ağ kartı tahsis edilir. Sanal makineler IP adreslerini kaynak üzerine kurulan Workstation DHCP sunucusundan otomatik olarak alırlar. Bununla birlikte sanal makinelere elle IP adresi verilebilir.

Network Address Translation (NAT);

Kaynak makinenin çevirmeli-ağ (dial-up) bağlantısını kullanarak internete çıkmak istiyorsak veya kaynak üzerinden dış ağlara çıkmak istiyorsak, sanal makine NAT ağ bağlantı türü ile kurulmalıdır. İnternete ve dış ağlara çıkışlarda sanal makine, kaynak IP adresini kullanır. NAT ile konfigüre edilen sanal makine IP adresini sanal DHCP sunucusundan alır [22].

Ayrıca VMWARE Workstation, TCP/IP, NetBEUI, Microsoft Networking, Samba, Novell NetWare ve Network File System içeren ethernet temelli protokolleri de destekler.

Uygulamamızda farklı iki sanal ortamı haberleştireceğimiz için bulunduğumuz sanal ağı Şekil 3.5'deki gibi "host-only" olarak atadık.



Şekil 3.5. VMWARE Workstation sanal ağ editörü

VMWARE Workstation programının kurulum aşamaları, tezin EK-A bölümünde ayrıntılı bir şekilde anlatılmıştır.

3.4. GNS 3 (Graphical Network Simulator 3)

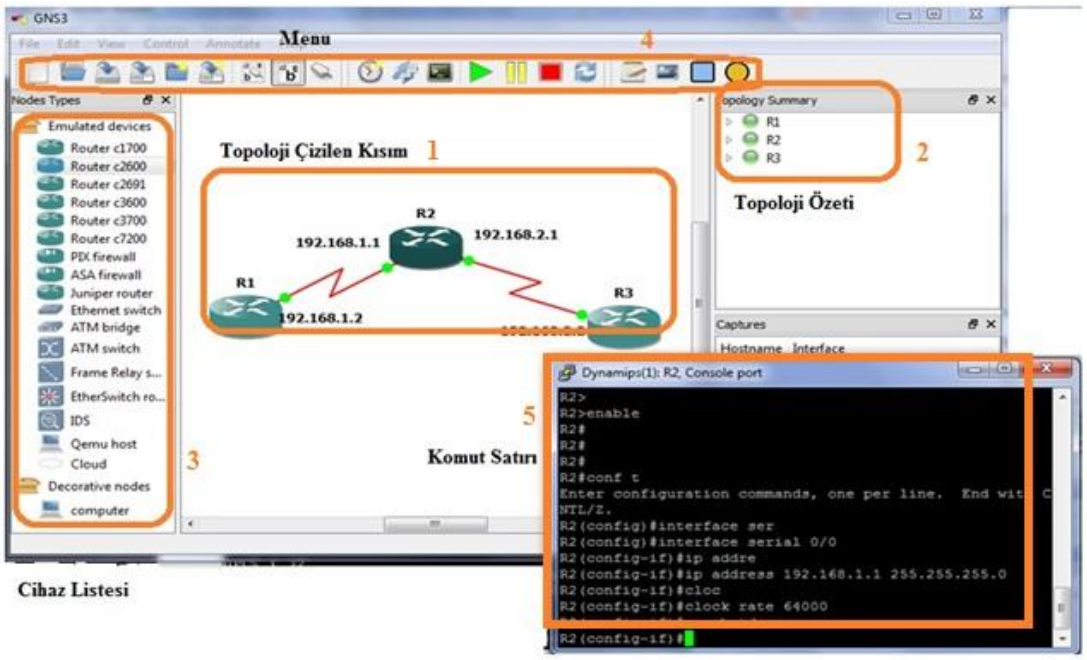
GNS 3, karmaşık ağları modelleyebilen açık kaynak kodlu grafiksel bir ağ modelleme programıdır. GNS 3'ü, VMWARE, Virtual Box veya XEN gibi sanallaştırma programları gibi düşünebiliriz. Bu sanallaştırma programları herhangi bir işletim sistemini sanallaştırırken, GNS 3 ise Cisco cihazlarını IOS'larını (Internetwork Operating System) simüle eder. Kendi çekirdek yapısında Cisco

IOS'larını emule eden Dynamips yazılımı çalışır. Dynamips'in üzerinde ise iyi kullanıcı arayüzü sağlayan dynagen yazılımı çalışır [23].

Ayrıca GNS3, Qemu, Pemu ve Virtual Box gibi diğer emülatör programlarını da destekler. Bu yazılımlar, Cisco ASA ve PIX güvenlik duvarlarını, Cisco IPS'i ve Juniper yönlendiricilerini emule etmek için kullanılırlar.

Piyasada, yönlendirici simülatörleri olmasına rağmen, birçoğu ya Cisco komutlarının hepsini desteklemiyor veyahut da desteklese bile çalışırken hata verebilmektedir. GNS 3'ün güçlü çekirdek yapısı sayesinde bu tarz sıkıntılarla karşılaşılmamaktadır.

Şekil 3.6'da GNS3 programının, tezin EK-B kısmında anlatılan kurulum aşamalarından sonraki genel görünümü verilmiştir. Şekil 3.6'da 1 nolu alan topoloji oluşturulan alanı, 2 nolu alan topolojinin özetini, 3 nolu alan kullanılabilecek cihaz listesini, 4 nolu alan menuyü ve son olarak ise 5 nolu alan da cihazları konfigüre edebilmek için kullanılan komut satırını gösterir.



Şekil 3.6. GNS3 kurulum sonrası genel görünüm

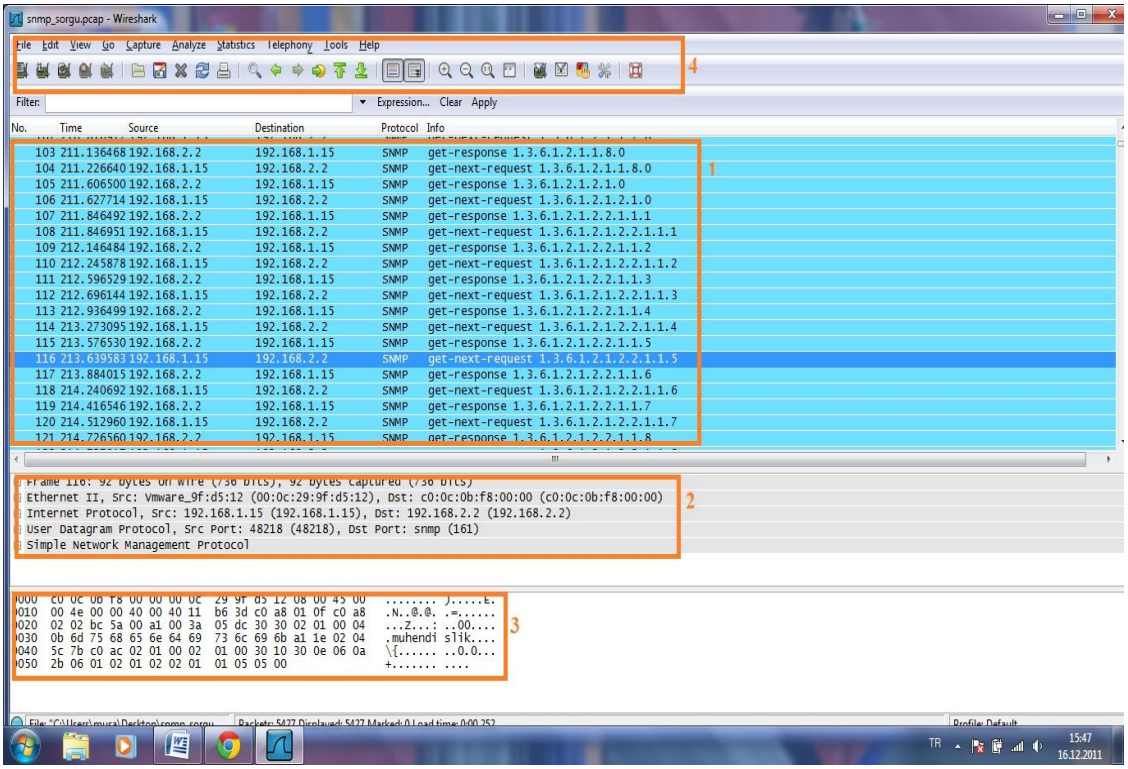
3.5. Wireshark

Wireshark programı günümüzde kullanılan en yaygın ağ analiz (paket yakalama) programıdır. İçerisinde bir ağ paket analiz yazılımı ile ağda iletilen paketleri yakalayıp detaylarıyla görüntülemeyi sağlar. Eski ismi Ethereal da olan Wireshark programı Windows, Linux, MacOS veya Solaris gibi birçok işletim sistemi altında çalışabilmektedir [24].

Wireshark. diğer analiz programlarına göre sağladığı bazı avantajlar sayesinde kullanım alanı çok fazladır. Bu avantajlardan bazılar aşağıdaki gibidir [25]:

a. 750'nin üzerinde protokolü analiz edebilmektedir.
b. Yakalanan paketler kaydedilebilir veya üzerinde değişiklik yapılabilir.
c. Gerçek zamanlı analiz yapar.
d. Yakalanan paketler arasında filtreleme yapabilir.
e. Kullanıcı arayüzü sadedir.

Şekil 3.7'de, uygulamada oluşturulan kurumsal ağdan veri çekilirken Wireshark ile yakalanmış SNMP paketleri gösterilmektedir. Şekil 3.7'de, 1 numarayla gösterilen alan yakalanan paketleri, 2 numaralı alan yakalanan paketin katmanlı mimari karşılığı ve yapısı, 3 numaralı alan veri içeriği ve son olarak da 4 numaralı alan ise de programın menü alanını göstermektedir.



Şekil 3.7. Wireshark programıyla yakalanan SNMP paketleri

BÖLÜM 4.

SANAL ORTAM ÜZERİNDE OLUŞTURULAN ÖRNEK BİR KURUMSAL AĞ TOPOLOJİSİNİN SNMPv3 İLE TOPOLOJİ KEŞFİ UYGULAMASI

4.1. Giriş

Büyük kurumlar artan altyapı masraflarını minimize edebilmek, oluşabilecek riskleri ortadan kaldırmak ve de daha kolay yönetim için kendi altyapılarını sanal ortamlara taşımaktadırlar. Sanallaştırmanın büyük önem kazandığı günümüzde sanallaştırmayla ilgili birçok çalışma yapılmaktadır. Biz de bu tez çalışmasında kurumsal ağ yönetimi için gerekli olan ağ topoloji keşfi ve ağ güvenliği konularını sanal platformlar üzerinde sağlayan bir uygulama gerçekleştirilmiştir.

Bu tez çalışmasıyla hedeflenen amaç, sanal ortam üzerinde oluşturulan kurumsal bir ağ topolojisinin, yine farklı bir sanal ortam üzerinde kullanılan bir algoritma ile topoloji keşfinin SNMPv3 ile yapılmasıdır. Ayrıca bu tez çalışmasıyla, gerçek dünyada karşılaşılan fiziksel gereksinimlerin ortadan kaldırılarak istenilen topoloji örneklerinin kolayca yapılabildiği ve bilgisayar ağları gibi derslerde bu çalışmaların kolayca uygulanabileceğinin gösterilmesi de hedeflenmiştir.

Ağ yönetimi ve ağ topoloji keşfi alanında yapılan birçok akademik ve ticari çalışma vardır. Bu çalışmalar gerek kullandıkları haberleşme protokolleri olsun, gerek oluşturuldukları platformlar olsun, gerekse kullandıkları algoritmalar ve yöntemlerle birbirleriyle farklılıklar arz etmektedirler. Bu çalışmalardan bazıları aşağıda verilmiştir:

[6] numaralı akademik çalışmada, sanal ağların, kullanıcı ve hizmet taleplerinde topolojilerin ve sistem düğümlerinin daha çabuk adapte olduğu dinamik ağ çevreleri ile karakterize edildiğinden bahsedilmiştir. Sanal ağları görüntülemek ve kontrol edebilmek için bilgi yönetimi sistemi diye bir sistem kullanılmıştır. Çalışmada, bu

sistemin sanal ortamlarda bile olsa performans yönetimi gibi bir alanda ne kadar başarılı olduğu gösterilmektedir.

[7] numaralı akademik çalışmada, sanal ağ ortamlarında kullanılan ve gereksinim duyulan kaynak miktarındaki dengeyi ayarlayan çeşitli kaynak ve hizmet yönetim tekniklerinden bahsedilmiştir. Bu tekniklerin kendi içlerinde iş bölümü yaparak bazılarının sadece kaynak yönetimini, bazılarının gerekli kaynak miktarını hesaplaması gibi özelliklerden bahsedilmiştir.

[1] numaralı akademik çalışmada, kurumsal bir ağın cihazlar arasındaki arayüz bağlantılarını kullanarak hem fiziksel hem de mantıksal topolojilerinin bulunmasını sağlayan bir algoritma dizayn edilmesi hedeflenmiştir. Bu amacı gerçekleştirmek için SNMP protokolünün çeşitli MIB değerleri, yönlendirme tablosu, ARP önbellek hafızası gibi yapılardan ilgili veriler çekilmiştir ve bu değerler kullanılmıştır.

[2] numaralı akademik çalışmadaki amaç, ağa mümkün olduğunca az sorgu göndererek, tek bir yetki alanı ile İnternet omurgası içinde otomatik olarak ağ topoloji keşfi yapmaktır. Bu çalışmada bir çok teknik birlikte kullanılmıştır (ping, traceroute, DNS, SNMP).

[3] numaralı akademik çalışmada, tüm ağ bilgisi gerekmeksizin otomatik ağ topoloji keşfi yapılması hedeflenmiştir. Bu çalışma için düşünülen algoritmada, iki bridge cihazı arasındaki üç cihazın iletim girdileriyle ilgili işlem yaparak hedefe gitmek istenmektedir.

[4] numaralı akademik çalışmada ise, “kök ve düğüm” mantığında çalışılmıştır. Çalışmada bir cihaz kök olarak seçilmiş, birkaç tane farklı cihaz ise düğüm olarak atanmıştır. Algoritma bu düğümlerin köke olan uzaklığını hesapladıktan sonra bir işlem yaparak diğer düğümleri bulmayı hedeflemektedir.

[5] numaralı akademik çalışmada ise, ağ topolojisini keşfetmek için kullanılan, veri bağı katmanı için kullanılan algoritma ile ağ katmanında keşif yapan algoritmanın SNMP protokolü bazlı karşılaştırılması yapılarak bazı sonuçlar çıkartılmıştır.

[8] numarada Nmap isimli ticari uygulama incelenmiştir. Nmap açık kaynak kodlu olup ücretsiz bir ağ yazılımıdır. Port bazlı veya IP bazlı arama yaparak topoloji keşfi yapabilmektedir. Bunun için ping ve traceroute mekanizmalarını kullanmaktadır.

[9] numarada ise WhatsUp ticari bir uygulama incelenmiştir. WhatsUp aslında bir ağ yönetim programıdır. Topoloji keşfi için kullandığı direk bir teknik yoktur. Ping mekanizması, ICMP ve SNMP protokollerini birlikte kullanarak topoloji keşfi sağlar.

Bu çalışmada, yukarıdaki çalışmalardan farklı olarak, kurumsal bir ağ topoloji keşfi, farklı sanal ortamlar birbirleriyle entegre halinde çalıştırılarak, SNMP'nin güvenli sürümü olan SNMPv3 ile gerçekleştirilmiştir. Uygulamamızda kullanılan algoritma [1] numaralı akademik çalışmada kullanılan algoritmanın uygulamada kullandığımız sanallaştırma programlarına göre yeniden düzenlenmiş halidir. Uygulamada sanallaştırma platformları olarak GNS3 ve VMWARE Workstation kullanılmıştır. Yazılım geliştirme ortamı olarak Microsoft'un Visual Studio 2010, veritabanı olarak da MySQL tercih edilmiştir.

4.2. Modelleme ve Konfigürasyon

4.2.1. Modelleme ortamı

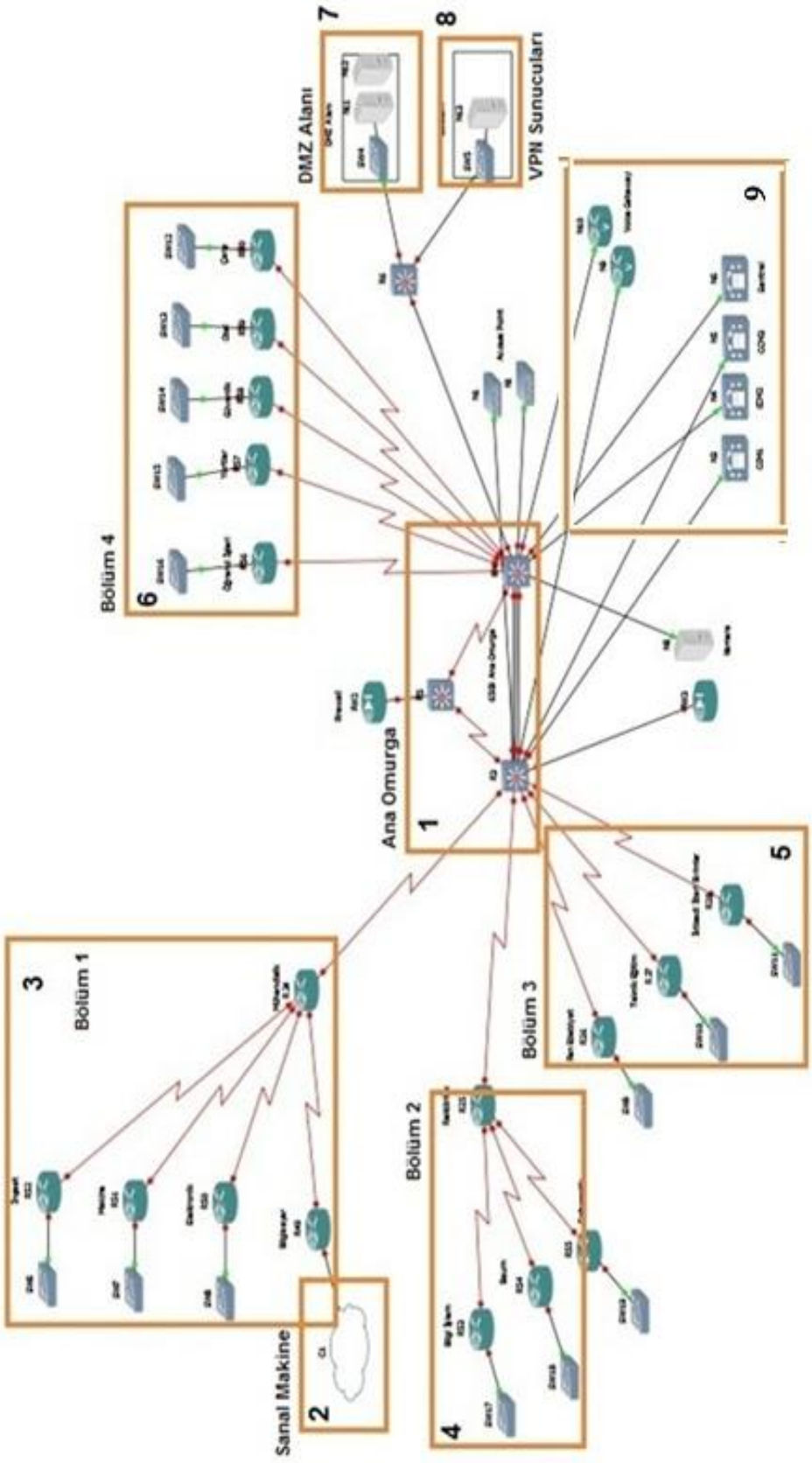
Uygulamada oluşturulan örnek kurumsal ağ topolojisi için iki farklı sanal ortam olan VMWARE Workstation ve GNS3 programları kullanılmıştır. GNS3, kurumsal ağımız modellemek için, VMWARE Workstation ise uygulamamızı çalıştırmak için kullanılmıştır. Bu programların birbirleriyle olan entegrasyonu, kullanılan yönlendirme protokolü bu bölüm içerisinde anlatılacaktır. Oluşturulan kurumsal ağ topolojisinde kullanılan yönlendirici ve anahtar cihazlarının modeli, IP adresleri gibi özellikler Tablo 4.1'de detaylı bir şekilde gösterilmiştir.

4.2.2. Sanal ortam üzerinde oluşturulan örnek bir kurumsal ağ topolojisi

Kurumsal bir ağ, tüm ayrık bölümlerini veya çalışma gruplarını, kurum içindeki tüm bilgisayar kullanıcılarına belli yetkiler dahilinde herhangi bir veriye veya kaynağa erişebilme özelliği vererek tek bir çatı altında toplayan ağlara denir. Aynı zamanda kurumsal bir ağ özerk ve heterojen sistemlerin birlikte çalıştırılabildiği, esneklik, güvenilirlik, performans, güvenlik ve iyi yönetim gibi özellikleri içerisinde

barındıran ağlara da denir. Özetle kurumsal bir ağ tüm sistemlerin bir düzen içerisinde tek bir organizasyon çatısı altında birleştirilmesi anlamına gelir.

Yukarıdaki kurumsal ağ kavramına bağlı kalarak farklı kurumsal ağ yapıları incelenmiş olup, kendi uygulamamız için Şekil 4.1'deki örnek bir kurumsal ağ modeli oluşturulmuştur. İncelenen kurumsal ağ modelleri tezin EK-C kısmında bulunmaktadır.



Şekil 4.1. Oluşturulan örnek kurumsal ağ modeli

Şekil 4.1'de 1 numarayla gösterilen alan kurumsal ağımızın ana omurgasını ifade eder. Tüm kurumsal ağın iskelet yapısı bu bölüm üzerine inşa edilmiştir. Kurumsal ağımızın 2 numarayla gösterilen kısmı ise VMWARE Workstation ile oluşturulan sanal makinemizi ifade eder. Bu kısımda uygulama kodu çalıştırılmaktadır.3, 4, 5 ve 6 numaralarıyla gösterilen kısımlar kurumsal ağımızın farklı bölümlerini ifade eder. 7 numarayla ifade edilen kısım, bir kurumun dış ve iç kaynaklarını kurum dışına açan (genellikle internet) yapı olan DMZ'i (Demilitarized Zone, Sivil Bölge) tanımlar. 8 numarayla gösterilen kısım ise gezici kişilerin veya kurum dışındayken kuruma güvenli bir şekilde bağlanılmasını sağlayan yapı olan VPN'i (Virtual Private Network, Sanal Özel Ağ) olarak tanımlar. Son olarak kurumsal ağımızın diğer kısımları ise çağrı merkezleri ve erişim noktalarını ifade eder. 9 numarayla tanımlanan alan ise kurumsal ağdaki çağrı merkezleri ve erişim noktalarını göstermektedir.

4.2.3. Topolojide kullanılan cihazlar

Şekil 4.1'deki kurumsal ağ topolojisi oluşturulduktan sonra topolojideki tüm cihazların birbirleriye iletişim halinde olabilmesi için konfigürasyonlarının yapılması gerekmektedir. Kurumsal ağ topolojisi büyük bir yapı olduğundan dolayı daha kolay yönetilebilmesi açısından Şekil 4.1'de de görüldüğü gibi ayrı ayrı yapılar halinde gösterilmiştir. Her bir yapıdaki cihazların birbirleriye olan bağlantıları iyi ayarlanmalıdır. Topolojinin fiziksel alt yapı işleri bittikten sonra, topolojimiz için büyük ağ yapıları için kullanılan OSPF (Open Short Path First- İlk Açık Yöne Öncelik) yönlendirme protokolü tercih edilmiştir.

OSPF, bir TCP/IP ağındaki yönlendiricilerin birbirlerini otomatik olarak tanımasında kullanılan bir protokoldür. Dijkstra'nın en kısa yol algoritmasını kullanan bir bağlantı-durum protokolüdür. OSPF ile birlikte, bir yönlendirici, ağın tüm topolojik haritasını oluşturur. Yönlendirici daha sonra yerel olarak tüm ağlara en kısa yol ağacını elde etmek için Dijkstra'nın en kısa yol algoritmasını kullanır ve kendisini bu ağaçta kök olarak belirler. Yönlendiriciler kendi aralarında birbirlerine her on saniyede bir "Hello " paketi yollayarak kendi üzerlerindeki yönlendirme tablolarını güncellerler. Daha kolay yönlendirme yapabilmek için OSPF alan mantığını kullanır.

Normalde tüm cihazlar “area 0” bağlıdır. Ağ yöneticisi ağı istediği gibi alanlara bölebilir [26].

Bizim kurumsal ağımızda da daha kolay yönetim açısından OSPF yönlendirme protokolü kullanılmıştır. Şekil 4.1’de 1 numarayla gösterilen ana omurga alanı “area 0” olarak atanmıştır. 3 numarayla gösterilen Bölüm 1 “area 100”, 4 numarayla gösterilen Bölüm 2 ise “area 200” olarak atanmıştır.

Tablo 4.1. Topolojide kullanılan cihaz listesi

	Alan	**Cihaz Tipi**	**Cihaz Modeli**	**IP**	**OSPF**
1	**Ana Omurga**	EtherSwitchRouter	Cisco 6509	192.168.6.1 192.168.7.1 192.168.8.1 192.168.9.1 192.168.10.1 192.168.15.1 192.168.30.1	Area 0
		EtherSwitchRouter	Cisco 6509	192.168.15.2 192.168.16.2 192.168.17.2 192.168.18.2 192.168.19.2 192.168.20.2 192.168.40.2	Area 0
		EtherSwitchRouter	Cisco 6509	192.168.30.2 192.168.40.1	Area 0
2	**Sanal Makine**	Sanal Makine	Windows 7	192.168.1.3	Area 100

Tablo 4.1. Topolojide kullanılan cihaz listesi (Devam)

3	**Bölüm 1**	Router	Cisco c3700	192.168.2.2 192.168.3.2 192.168.4.2 192.168.5.2 192.168.6.2 →	Area 100 Area 0
		Router	Cisco 2691	192.168.1.1 192.168.2.1	Area 100
		Router	Cisco 2691	192.168.3.1	Area 100
		Router	Cisco 2691	192.168.4.1	Area 100
		Router	Cisco 2691	192.168.5.1	Area 100
		Switch	EthernetSwitch	__	
		Switch	EthernetSwitch	__	
		Switch	EthernetSwitch	__	
4	**Bölüm 2**	Router	Cisco c3700	192.168.7.2 → 192.168.11.1 192.168.12.1 192.168.13.1	Area 0 Area 200
		Router	Cisco 2691	192.168.11.2	Area 200
		Router	Cisco 2691	192.168.12.2	Area 200
		Router	Cisco 2691	192.168.13.2	Area 200
		Switch	EthernetSwitch	__	
		Switch	EthernetSwitch	__	
		Switch	EthernetSwitch	__	
5	**Bölüm 3**	Router	Cisco c3700	192.168.8.2	Area 0
		Router	Cisco c3700	192.168.9.2	Area 0

Tablo 4.1. Topolojide kullanılan cihaz listesi (Devam)

		Router	Cisco c3700	192.168.10.2	Area 0
		Switch	EthernetSwitch	__	
		Switch	EthernetSwitch	__	
		Switch	EthernetSwitch	__	
6	**Bölüm 4**	Router	Cisco 2691	192.168.16.1	Area 0
		Router	Cisco 2691	192.168.17.1	Area 0
		Router	Cisco 2691	192.168.18.1	Area 0
		Router	Cisco 2691	192.168.19.1	Area 0
		Router	Cisco 2691	192.168.20.1	Area 0
		Switch	EthernetSwitch	__	
		Switch	EthernetSwitch	__	
		Switch	EthernetSwitch	__	
		Switch	EthernetSwitch	__	
		Switch	EthernetSwitch	__	
7	**DMZ Alanı**	Switch	EthernetSwitch	__	
		Server	Düğüm Makine	192.168.50.2	
		Server	Düğüm Makine	192.168.50.3	
8	**VPN Sunucu**	Switch	EthernetSwitch	__	
		Server	Düğüm Makine	192.168.60.2	

Oluşturulacak örnek kurumsal ağ topolojisi için gerekli cihazların tespitinden sonra bu cihazların birbirleriyle haberleşebilmeleri için konfigürasyonlarının yapılması gerekmektedir.

4.2.4. Konfigürasyon

Bu bölümde, Şekil 4.1'deki kurumsal ağımızdaki ana omurgada bulunan Bölüm 1, Bölüm 2 ve Bölüm 3'e bağlı olan Cisco c3700 cihazının konfigürasyonu gösterilecektir. Topolojide kullanılan tüm cihazlar aynı mantık doğrultusunda konfigüre edilir. Topolojideki diğer cihazların konfigürasyonları tezin EK-D kısmında yer almaktadır.

4.2.4.1. Arayüzlerin konfigüre edilmesi

Şekilde 4.2'de, cihazın aktif arayüzlerinin ip adres atamaları ve bu arayüzlere "clock rate" değeri atanması gösterilmiştir.

```
R1>enable
R1#configure terminal
R1(config)#hostname  ana_omurga_sol
ana_omurga_sol(config)#interface serial 0/0
ana_omurga_sol(config-if)#no shutdown
ana_omurga_sol(config-if)#ip address 192.168.6.1   255.255.255.0
ana_omurga_sol(config-if)#clock rate 64000
ana_omurga_sol(config-if)#exit
ana_omurga_sol(config)#interface serial 0/1
ana_omurga_sol(config-if)#no shutdown
ana_omurga_sol(config-if)#ip address 192.168.7.1   255.255.255.0
ana_omurga_sol(config-if)#clock rate 64000
ana_omurga_sol(config-if)#exit
ana_omurga_sol(config)#interface serial 0/2
ana_omurga_sol(config-if)#no shutdown
ana_omurga_sol(config-if)#ip address 192.168.8.1   255.255.255.0
ana_omurga_sol(config-if)#clock rate 64000
```

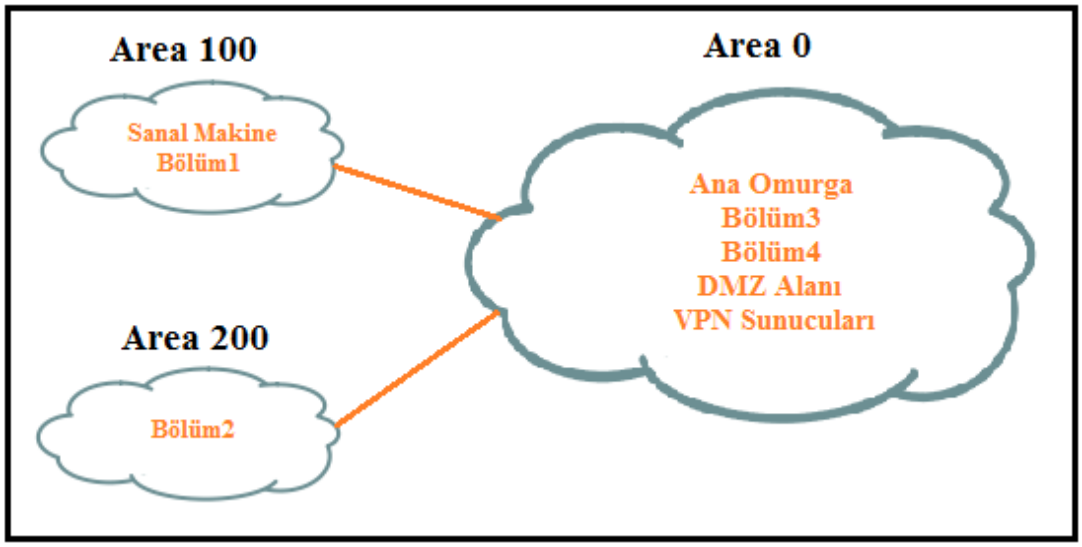
Şekil 4.2. Ana omurgadaki cihazda arayüz konfigürasyonu

```
ana_omurga_sol(config-if)#exit
ana_omurga_sol(config)#interface serial 0/3
ana_omurga_sol(config-if)#no shutdown
ana_omurga_sol(config-if)#ip address 192.168.9.1   255.255.255.0
ana_omurga_sol(config-if)#clock rate 64000
ana_omurga_sol(config-if)#exit
ana_omurga_sol(config)#interface serial 0/4
ana_omurga_sol(config-if)#no shutdown
ana_omurga_sol(config-if)#ip address 192.168.10.1  255.255.255.0
ana_omurga_sol(config-if)#clock rate 64000
ana_omurga_sol(config-if)#exit
ana_omurga_sol(config)#interface serial 0/5
ana_omurga_sol(config-if)#no shutdown
ana_omurga_sol(config-if)#ip address 192.168.15.1   255.255.255.0
ana_omurga_sol(config-if)#clock rate 64000
ana_omurga_sol(config-if)#exit
ana_omurga_sol(config)#interface serial 1/0
ana_omurga_sol(config-if)#no shutdown
ana_omurga_sol(config-if)#ip address 192.168.30.1   255.255.255.0
ana_omurga_sol(config-if)#clock rate 64000
 ana_omurga_sol(config-if)#exit
```

Şekil 4.2. Ana omurgadaki cihazda arayüz konfigürasyonu (Devam)

4.2.4.2. Yönlendirme protokolünün konfigüre edilmesi

Şekil 4.1'deki kurumsal ağın topolojisinin OSPF yapısı hakkında daha önceki bölümlerde ve Tablo 4.1'de bilgiler verilmiştir. Şekil 4.3'de, bu bilgiler doğrultusunda modellenen kurumsal ağın OSPF yapısı gösterilmiştir.



Şekil 4.3. Uygulamada kullanılan OSPF yapısı

Şekil 4.4'de ise, Cisco c3700 cihazının yönlendirme protokolü konfigürasyonu verilmiş olup, bu cihazın "area 0" da olduğu ve hangi ağlara bağlı olduğunun ataması yapılmaktadır.

```
ana_omuga_sol(config)#router ospf 10
ana_omurga_sol(config-router)#network 192.168.6.0  0.0.0.255 area 0
ana_omurga_sol(config-router)#network 192.168.7.0  0.0.0.255 area 0
ana_omurga_sol(config-router)#network 192.168.8.0  0.0.0.255 area 0
ana_omurga_sol(config-router)#network 192.168.9.0  0.0.0.255 area 0
ana_omurga_sol(config-router)#network 192.168.10.0  0.0.0.255 area 0
ana_omurga_sol(config-router)#network 192.168.15.0  0.0.0.255 area 0
ana_omurga_sol(config-router)#network 192.168.30.0  0.0.0.255 area 0
```

Şekil 4.4. Ana omurgadaki cihazda yönlendirme protokolü konfigürasyonu

4.2.4.3. SNMPv3 konfigürasyonu

Uygulamamızda daha önceden de bahsedildiği üzere güvenlik özelliği yüzünden SNMPv3 kullanılmıştır. SNMP protokolünün güvenlik bazlı karşılaştırması Tablo

4.2'de verilmiştir. Çalışmamızda, 5 numaralı SNMP özelliği seçilerek uygulamamız geliştirilmiştir.

Tablo 4.2. SNMP sürümlerinin güvenlik karşılaştırmaları [15]

	Model	Seviye	Kimlik Doğrulama	Şifreleme
1	v1	noAuthNoPriv	Topluluk ismi	Yok
2	v2c	noAuthNoPriv	Topluluk ismi	Yok
3	v3	noAuthNoPriv	Kullanıcı adı	Yok
4	v3	authNoPriv	MD5,veya SHA	Yok
5	v3	authPriv	MD5, veya SHA	DES

SNMPv3 konfigürasyonu için yönlendiriciler ve yönetilebilen anahtar cihazlar aşağıdaki sırada göre konfigüre edilmiştir:

a. Grup oluşturulur: Tablo 4.2'deki güvenlik modeli (v3) ve güvenlik düzeyi seçilir. Aşağıdaki komut satırı ile aynı bölgedeki cihazlar için grup1 isimli grup oluşturmuş olup, güvenlik düzeyi v3 seçilerek bu gruba sadece "read" özelliği atanmıştır.

```
snmp-server group grup1 v3 priv read grup1_oku
```

b. Kullanıcı oluşturulur: Gruba eklenecek kullanıcılar güvenlik esaslarına göre oluşturulur. Bir önceki adımda oluşturulan "grup1"e, "kullanıcı1" atanır. Burada oluşturulan güvenlik kriterleri için kullanıcı doğrulaması için "md5" algoritması, şifreleme için ise de "aes" algoritması kullanılmıştır.

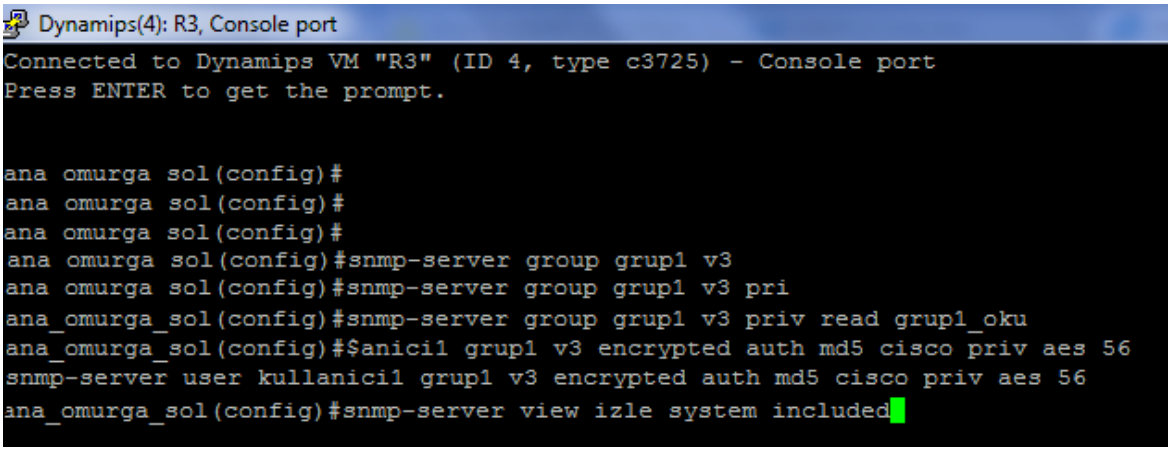
```
snmp-server user kullanici1 grup1 v3 encrypted auth md5 cisco priv aes 256
```

c. Özellikler atanır: Oluşturulan bu gruplara "view, read, write" gibi özellikler atanır. Bizim çalışmamız topoloji keşfi olduğundan dolayı cihazlarımızı "view" olarak ayarlamamız yeterli olacaktır. Bu komuttaki "system" ifadesi, bu gruba

her cihazda MIB II'de tanımlı olan "system" değerlerine erişme imkanı sağlar.

```
snmp-server view izle system included
```

Şekil 4.5'de, kurumsal topolojimizin ana omurga bölümünde bulunan Bölüm1, Bölüm2 ve Bölüm3'e bağlı olan cihazın SNMPv3 konfigürasyonu, yukarıdaki bilgilere bağlı olarak komut satırında gösterilmiştir.



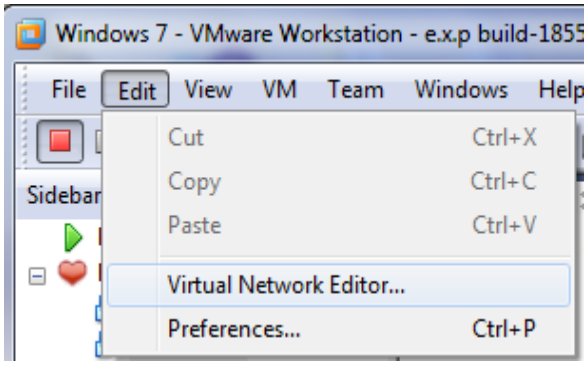
Şekil 4.5. SNMPv3 konfigürasyonu

4.2.4.4. GNS3 ile VMWARE Workstation entegrasyonu

GNS3 ile VMWARE Workstation programlarının kurulum aşamaları ve ilgili özelliklerinin atanması gibi konular tezin EK-A ve EK-B kısımlarında anlatıldığından ve yine uygulamada örnek alınan kurumsal ağ topolojisinin konfigürasyonun tezin uygulama kısmında anlatıldığından dolayı bu bölümde sadece GNS3 ve VMWARE Workstation programlarının birbirleriyle nasıl entegre oldukları anlatılmıştır.

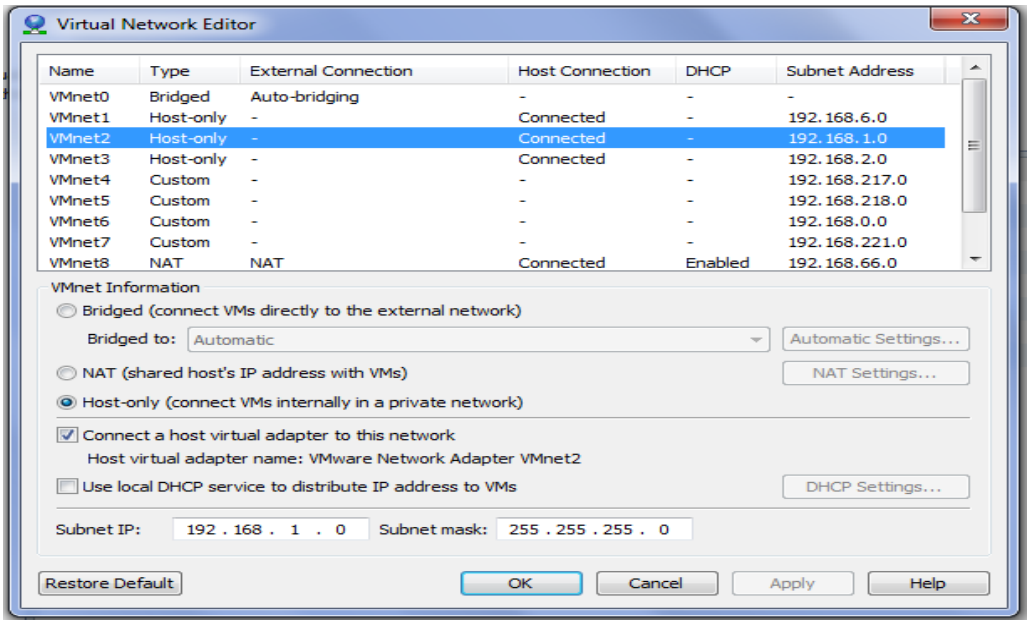
VMWARE Workstation tarafında yapılması gerekenler:

1- Program çalıştırılır ve Şekil 4.6'daki gibi **Edit⟶ Virtual Network Editor** seçeneği seçilir.



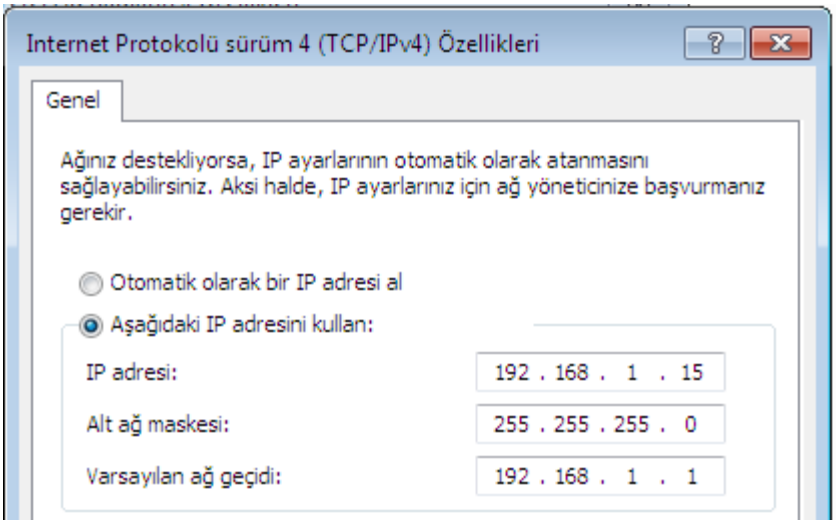
Şekil 4.6. Menüden sanal ağ editörünün seçilmesi

2- Şekil 4.7'de, ekrana gelen pencereden daha önceden sanal makinemize atamış olduğumuz VMnet 2 seçilerek "Host-only" olarak atanır ve son olarak şeklin altında bulunan kısımda bu sanal makinenin kullanacağı altağ alanı verilir.



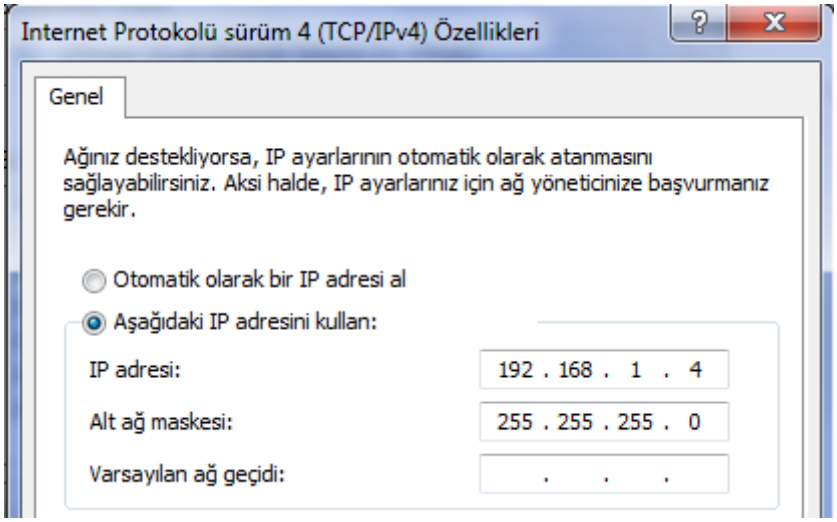
Şekil 4.7. Sanal ağ editöründe ayarlama yapılması

3- Sanal makine çalıştırılır ve Şekil 4.8'deki gibi ilgili IP adresi, ağ maskesi ve varsayılan ağ geçidi ayarlanır. Burada kullanılan varsayılan ağ geçidimiz bağlı olunan yönlendirici cihazını ifade etmektedir.



Şekil 4.8. Sanal makineye IP verme

4- VMWARE Workstation programının kurulu olduğu makine sanal ethernet kartı gerçek bir arayüz olarak görüntülenecektir. Şekil 4.9'da gösterildiği gibi bu kısma sadece IP adresi ve ağ maskesi vermek yeterli olacaktır.



Şekil 4.9. Sanal ethernet kartına IP verme

VMWARE Workstation tarafında yapılacak işlemler tamamlanmıştır. Bundan sonraki kısımda GNS 3 ile ilgili ayarlamalar yapılacaktır.

GNS 3 tarafında yapılması gerekenler:

Tüm topolojinin konfigürasyonu bittikten sonra:

1-GNS3 arayüzünün komut satırına aşağıdaki komut girilir.

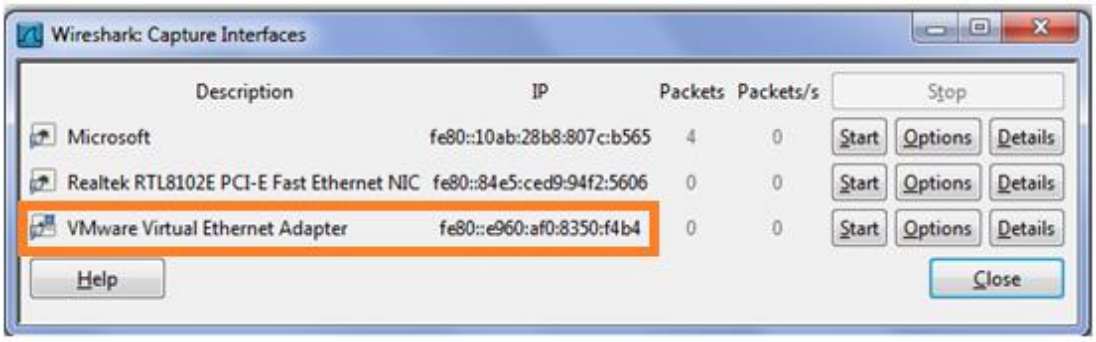
```
=> export /all C:\Lab_Configs
```

2- Bu işlemden sonra tüm konfigüre edilen cihazların konfigürasyonları bu klasöre kaydolmuş olur. Oluşturulan kurumsal ağın topolojisini kaydetmek için ise programın ana penceresinden **File →Save** diyerek çıkan pencerede uygun isim vererek **.net** uzantıyla kayıt işlemini tamamlarız.

Bu adımları da yaptıktan sonra GNS3 programıyla ilgili aşamalar tamamlandığından programı kapatabiliriz. GNS3'deki topolojimize VMWARE Workstation'daki sanal makinemizi eklemek için host-only olarak atamış olduğumuz VMnet 2 isimli sanal Ethernet adaptörünün ID'sini öğrenmemiz gerekir bunun için ise Wireshark programını kullanmamız gerekir.

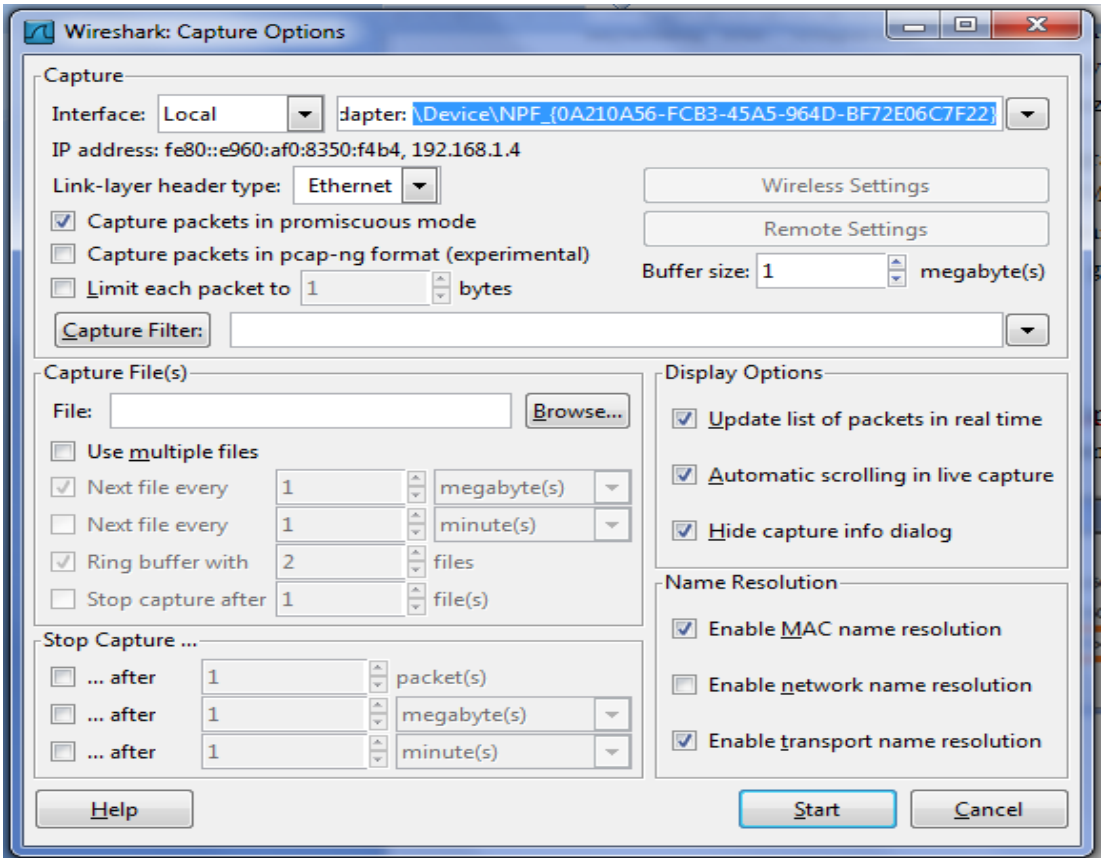
Wireshark tarafında yapılması gerekenler:

1- Program çalıştırılır ve ana menudeki **Capture →Interface** sekmesi tıklanır.
2- Çıkan pencerede **VMware Virtual Ethernet Adapter** seçilir.



Şekil 4.10. Wireshark programında arayüz seçme

3- Şekil 4.10'daki seçili alandaki **Options** sekmesine basılır ve Şekil 4.11'deki seçili alan bir yere kaydedilir.



Şekil 4.11. Sanal adaptör ID'si seçilmesi

Bu işlem de tamamlandıktan sonra Wireshark programı kapatılabilir. Daha önceden

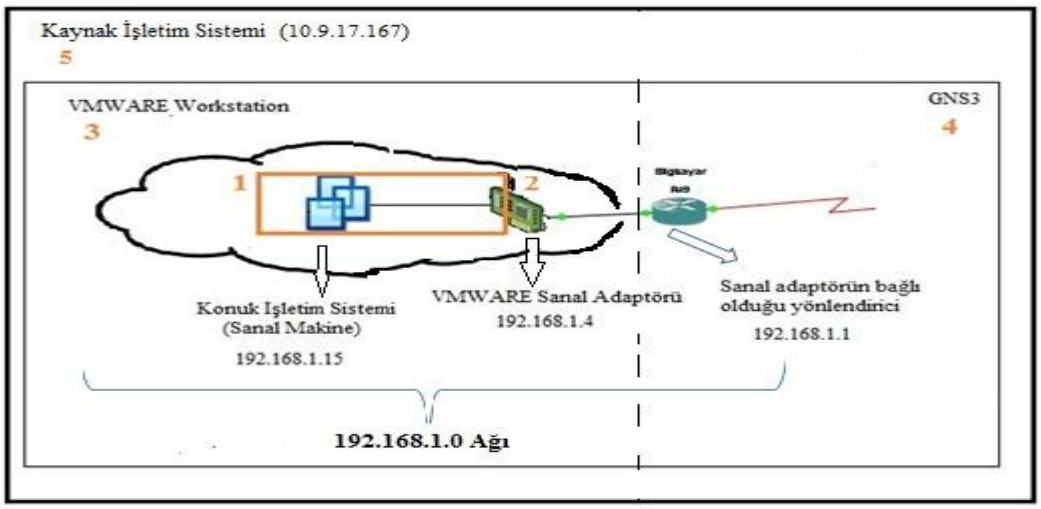
.net uzantılı kaydedilen topoloji dosyası bir metin düzenleyici program ile açılır. Sanal makinemiz topolojideki hangi yönlendiriciye eklenecekse onun altına Şekil

4.11'deki seçili alandaki daha önceden Wireshark ile bulunan ID, Şekil 4.12'deki gibi ilgili alana eklenir ve tekrar kaydedilerek işlem tamamlanır.

```
[[ROUTER R49]]
    model = 2691
    console = 2026
    aux = 2116
    cnfg = R49.cfg
    wic0/0 = WIC-2T
    s0/0 = R24 s2/1
    f0/0 = nio_gen_eth:\Device\NPF_{0A210A56-FCB3-45A5-964D-
BF72E06C7F22}
    x = -901.327127298
    y = -242.61731573
```

Şekil 4.12. Yönlendirici cihazına sanal adaptör ID'sinin eklenmesi

Sonuç olarak bu bölüm içerisinde, GNS3 ve VMWARE Workstation programlarının kurumsal ağ topolojisi içerisinde birbirleriyle olan entegrasyonu anlatılmıştır. Şekil 4.13'de, 1 numarayla gösterilen alan VMWARE Workstation'da oluşturulan konuk işletim sistemini, 2 numarayla gösterilen alan yönlendirici ile konuk işletim sistemini bağlayan WMWARE sanal adaptörünü göstermektedir. Şekil 4.13'de, 3 numarayla gösterilen alan konuk işletim sisteminin kurulu olduğu VMWARE Workstation programını, 4 numarayla gösterilen alan ise kurumsal ağ topolojimizin oluşturulduğu GNS3 programını ifade eder. Şekil 4.13'de, 5 numarayla gösterilen alan ise tüm bu programların ve işlemlerin üzerinde çalıştığı kaynak işletim sistemini gösterir.



Şekil 4.13. Sanal ortamların birbirleriyle olan ilişkisi

4.3. Topoloji Keşfi Uygulaması

4.3.1. Geliştirme ortamı

Modelleme ortamı ile alakalı konfigürasyon işlemleri tamamlandıktan sonra topoloji keşfi uygulamasını geliştirmek için aşağıdaki geliştirme ortamları kullanılmıştır.

a. Uygulama kodu Visual Studio 2010'da C# programlama diliyle yazılmıştır.
b. Veritabanı olarak MySQL kullanılmıştır.
c. Uygulamada, SNMP paketlerinin ağ içerisinde kolaylıkla yönetilebilmesi için kullanımı yaygın olan Net-SNMP kütüphanesinin "snmpwalk" uygulaması seçilmiştir. Net-SNMP sanal makine üzerine kurulmuştur.
d. Sorgu işlemlerini kolaylaştırmak için WebNMS'nin SNMP API'si kullanılmıştır.

Sonraki bölümlerde, yukarıda belirtilen geliştirme ortamı bileşenleri ile alakalı bilgiler verilecektir.

Net-SNMP;

Net-SNMP, SNMP (v1,v2c,v3) protokollerini kullanan bir yazılım takımıdır. IPv4, IPv6, IPX ve diğer ağ katmanı protokollerini destekleyen, SNMP ajanı tarafında genel bir istemci kütüphanesi sunan komut satırı uygulama takımıdır.

Net-SNMP, açık kaynak kod paylaşım sitelerinden SourceForge tarafından Mart 2005 tarihinden itibaren piyasa sunulmuş ve ağ yönetim sistemlerinde sürekli kullanılan bir yazılım takımıdır. Windows, Linux, FreeBSD, OpenBSD, Solaris ve Mac OS gibi işletim sistemlerinde rahatça çalışabilir [27]. Net-SNMP içerisinde bulunan uygulamalar Tablo 4.3'de gösterilmiştir.

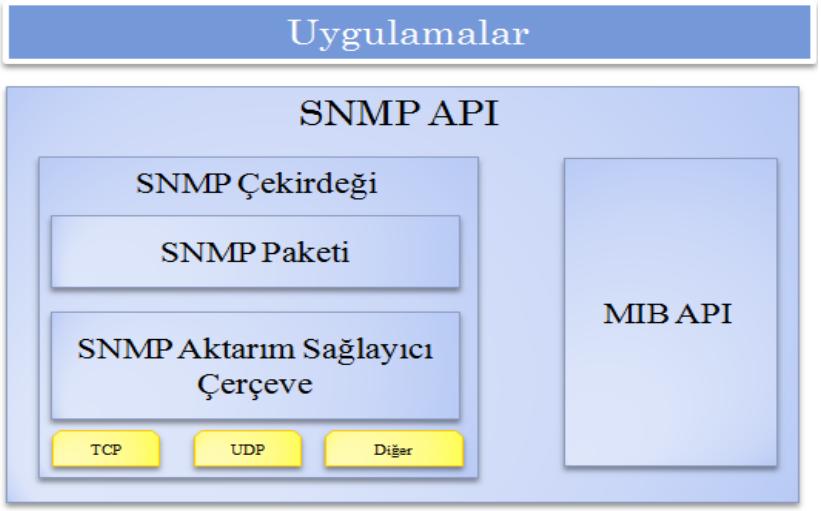
Tablo 4.3. Net-SNMP kütüphanesindeki uygulamalar [24]

Uygulama	Tanımlama
snmptranslate	MIB ve OID isimlerini rakamsal ve metinsel olarak çevirir.
snmpget	SNMP GET isteğini kullanarak bir ağ cihazıyla iletişimi sağlar.
snmpgetnext	SNMP GETNEXT isteğini kullanarak bir ağ cihazıyla iletişimi sağlar.
snmpbulkget	SNMP GETBULK isteğini kullanarak bir ağ cihazıyla iletişimi sağlar.
snmpwalk	SNMP GETNEXT isteğini kullanarak cihazdan veri çeker.
snmpset	SNMP SET isteğini kullanarak bir ağ cihazıyla iletişimi sağlar.
snmptrap	SNMP TRAP ve INFORM bildirim mesajları gönderir.
snmpd	SNMP isteklerine cevap verir.
snmptrapd	SNMP TRAP veya INFORM'larını dinleyen ve bunların logunu tutar.
mib2c	MIB yapılarını, C kodu gibi diğer yapılara dönüştürür.

Uygulamamızda cihazlardan veri çekebilmek için "snmpwalk" sorgu tekniği kullanılmıştır. "snmpwalk" sorgu sıralaması Şekil 4.16'daki gibidir. Ayrıca uygulamada, Net-SNMP'den farklı olarak SNMP sorgularını daha kolay yönetmek için içerisinde sınıflar barındıran WebNMS SNMP API'si bir sonraki bölümde anlatılacaktır.

WebNMS SNMP API .NET;

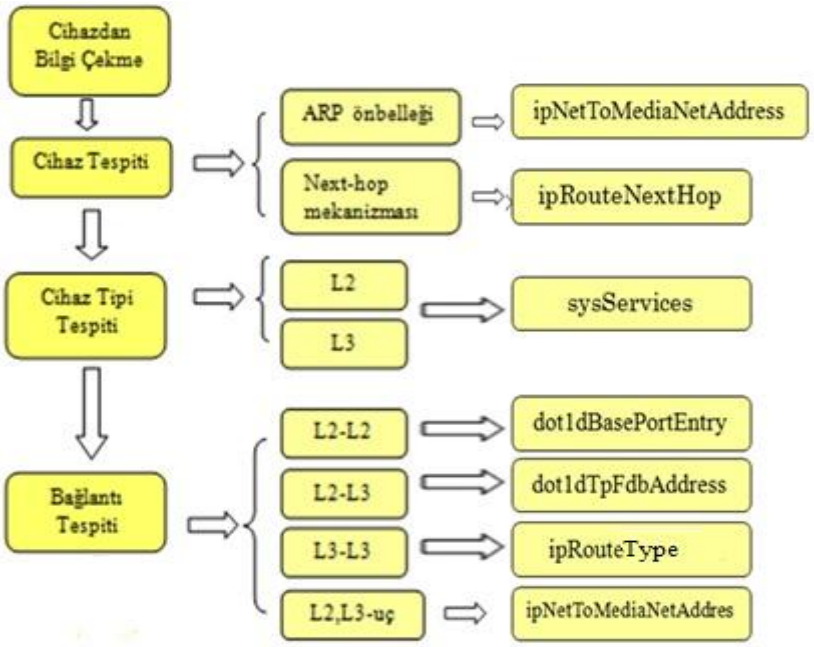
WebNMS SNMP API.NET, SNMP bazlı ağ yönetim uygulamalarında geniş kapsamlı bir uygulama geliştirme ortamı sağlar. Güçlü bir yığın yapısı sağlar. Büyük ağlarda görüntüleme ve yönetim için gerçek zamanlı uygulama olanağı sağlar. GET, GETNEXT, GETBULK, SET komutlarını destekler [28]. WebNMS SNMP API .NET'in katmanlı yapısı Şekil 4.14 'de görüldüğü gibidir.



Şekil 4.14. WebNMS SNMP API'sinin katmanlı yapısı [28]

4.3.2 Kullanılan algoritma

Uygulamamızda kullanılan algoritmanın genel görünümü Şekil 4.15'deki gibidir. Daha önceden de bahsedildiği üzere uygulamamızın genel algoritması [1] numaralı çalışmanın kendi uygulamamıza göre yeniden düzenlenmiş halidir.

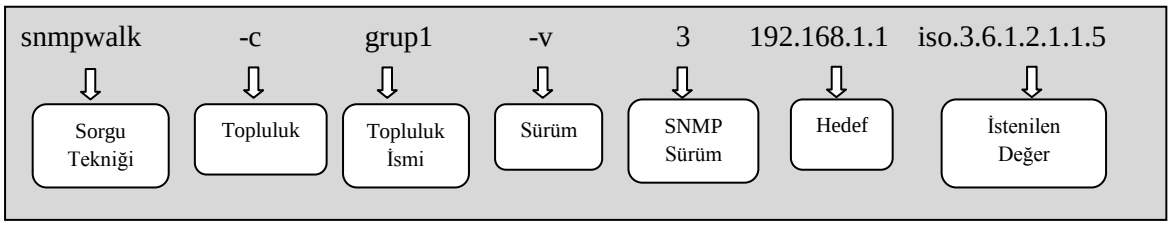


Şekil 4.15. Uygulamanın genel algoritması [1]

Algoritmanın adımları;

Bu bölümde algoritmanın adımları detaylı bir şekilde anlatılmıştır.

a. Cihazdan bilgi çekme: Algoritmanın diğer adımlarında kullanılmak üzere cihazlara sorgular gönderilir. Bu işlem, tezin daha sonraki bölümlerinde anlatılan bir SNMP sorgu tekniği olan "snmpwalk" kullanılarak yapılır. Çekilen sorgudaki parametrelere göre cihazdan istenen veri çekilebilir. Şekil 4.16'da cihaz ismini getiren bir SNMP sorgusu gösterilmiştir. Bu sorgunun haricinde bölüm 2'de MIB ve OID başlıkları altında anlatılan değişkenler (tanıtıcılar) sayesinde istenilen bilgi çekilebilmektedir.

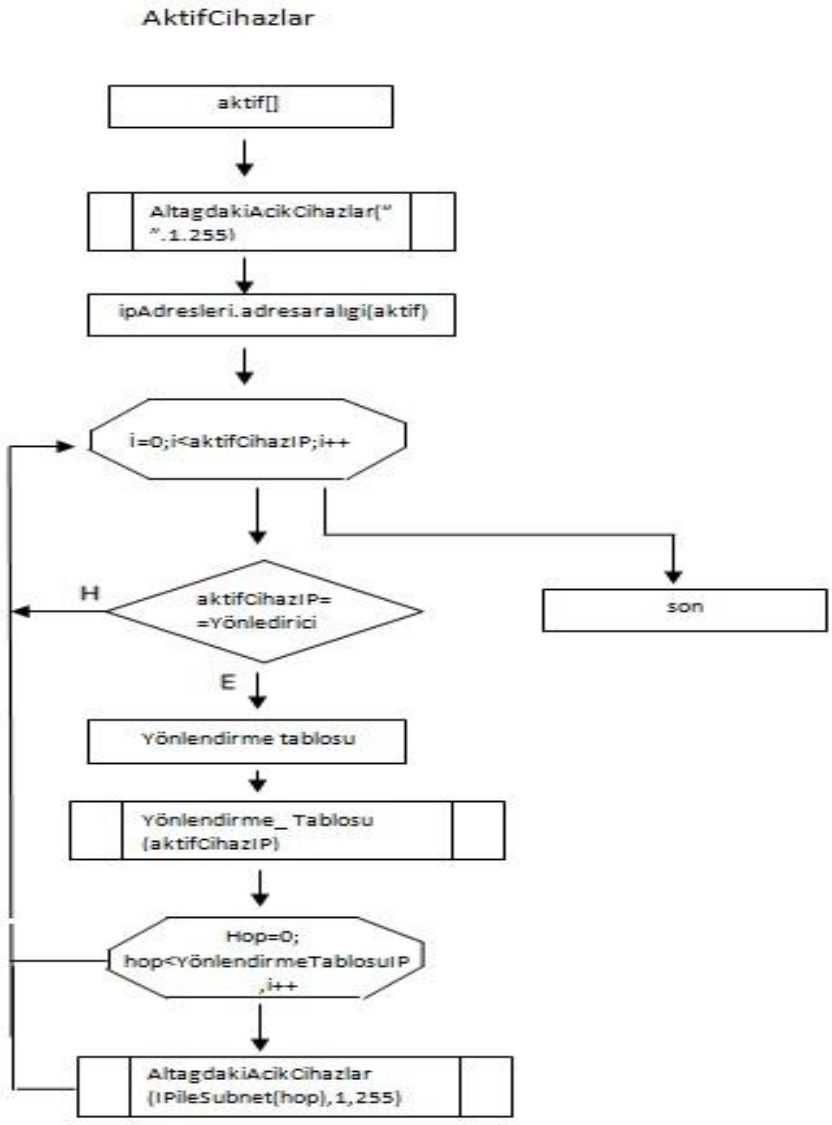


Şekil 4.16. Örnek snmpwalk Sorgusu

b. Cihaz tespiti: Algoritmanın bu kısmında topolojimizdeki aktif tüm cihazların tespiti hedeflenmektedir. Bunun için kullanılan iki teknik vardır: ARP önbellek kayıtları ve Next-hop mekanizması. Bu iki teknik vasıtasıyla, cihazlar üzerinden "ipNetToMediaNetAddress" ve "ipRouteNextHop" değerleri çekilir. "ipNetToMediaNetAddress" değeri katman 2 destekli aktif cihazların bulunmasında, "ipRouteNextHop" değeri ise katman 3 destekli aktif cihazların bulunmasında kullanılan MIB değerleridir.

Şekil 4.17'de, topolojideki tüm aktif cihazların bulunması hedeflenmektedir. Algoritmada ilk önce aktif diye bir dizi tanımlanır. Daha sonra Şekil 4.19'daki "AltagdakiAcikCihazlar" alt fonksiyonu çağrılır. Daha sonra dönen değerler "ipAdresleri" aralığına atanır. Dönen değer kadar bir döngü kurulur. Bu döngü içerisindeki herhangi bir cihaz yönlendirici cihazı ise, o yönlendiricinin yönlendirme tablosu alınır ve o tablodaki alt ağlara yine Şekil 4.19'daki "AltagdakiAcikCihazlar"

fonksiyonu çağrılarak rekürsif bir yapı elde edilir ve tüm aktif cihazlar bulunmuş olur.



Şekil 4.17. Aktif cihaz bulma algoritması

Şekil 4.17'deki algoritmanın C# diliyle yazılmış kod örneği Şekil 4.18'de gösterilmiştir.

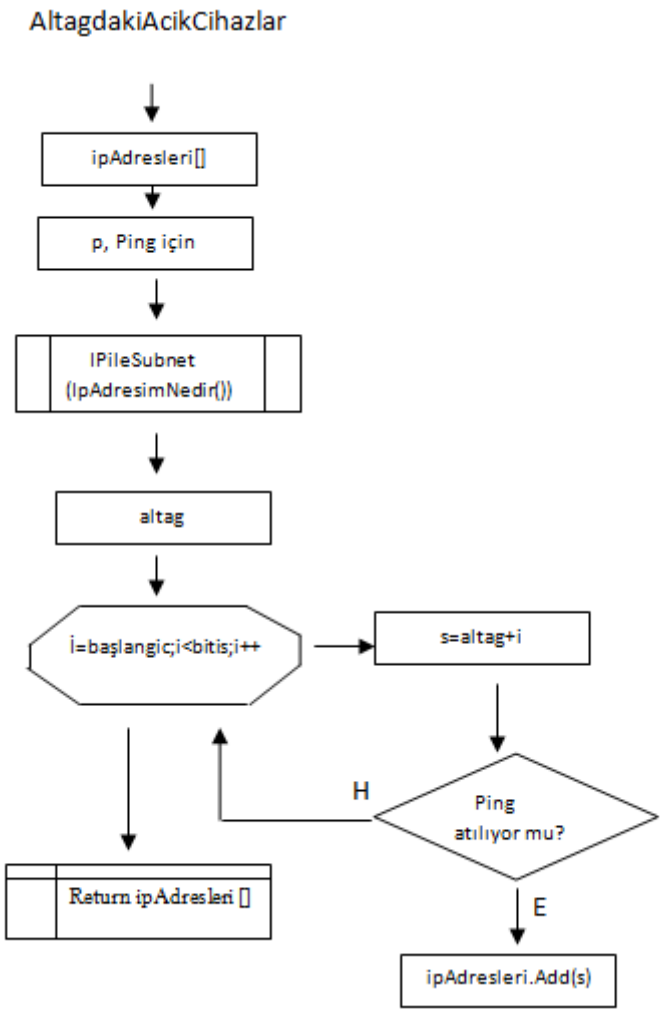
```
//Aktif Cihazları bulur ve ekranda gosterir.

    private void aktifCihazlarClick_Click(object sender, EventArgs e)
    {
       txtSonuc.Clear();
       txtSonuc.Visible = true;
       DateTime dtStart = DateTime.Now;
       List<string>  aktif = Topoloji.SubnetAltindakiAcikCihazlar("", 1, 255);
       _ipAdresleri.AddRange(aktif);
       foreach (string aktifCihazIP in aktif)
       {
          if (Topoloji.CihazTipi(aktifCihazIP) == DeviceType.Router)
          {
             List<string> nextHop = Topoloji.RoutingTablosuNextHopBul(aktifCihazIP);
             foreach (string hop in nextHop)
             {

       _ipAdresleri.AddRange(Topoloji.SubnetAltindakiAcikCihazlar(Topoloji.IPileSubnet
       (hop), 1, 6));
             }
          }
       }
        int dtEnd = DateTime.Now.Subtract(dtStart).Seconds +
        DateTime.Now.Subtract(dtStart).Minutes * 60;
       string result = "";
       foreach (string item in _ipAdresleri)
       {
          result += item + "\r\n";
       }
       sure = dtEnd;
       txtSonuc.Text = result + "\r\n Süre : " + dtEnd.ToString();
    }
```

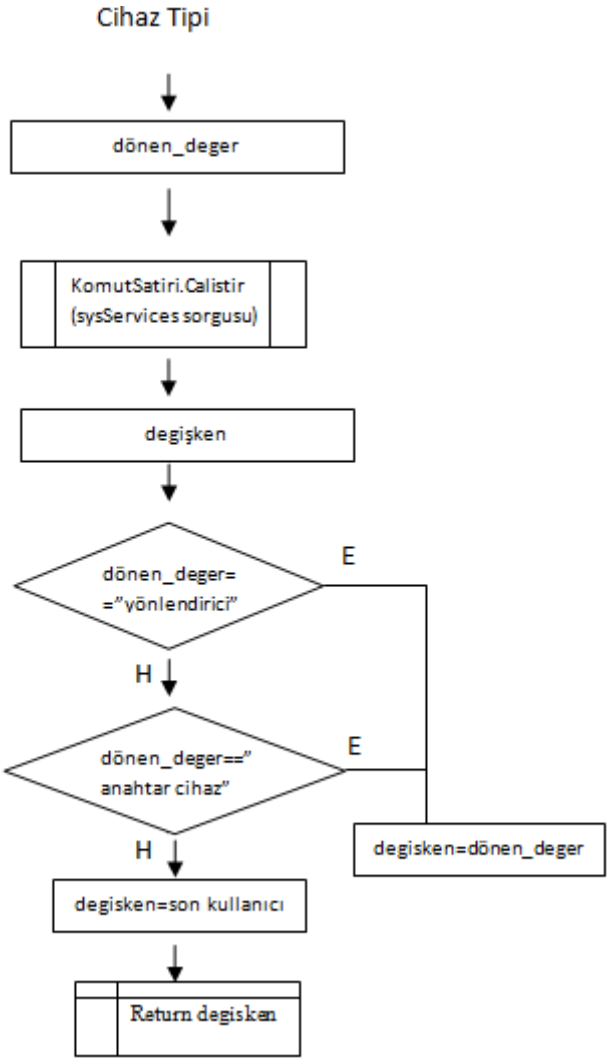
Şekil 4.18. Aktif cihaz bulma algoritmasının C# kodu

Şekil 4.19'da öncelikle "ipAdresleri []" dizisi oluşturulmuş ve ping mekanizması için değişken tanımlanmıştır. Daha sonra kendi cihazımızın bulunduğu alt ağ bulunmuştur. Daha sonrasında bir döngü kurularak ip adresleri birer birer artırılarak hepsine ping atılmıştır. Başarılı olan ip adresleri en baştaki "ipAdresleri []" dizisine eklenmiştir. Bu algoritmada rekürsif bir yapıda çalışmaktadır.



Şekil 4.19. Alt ağ altındaki açık cihazları bulma algoritması

c. Cihaz tipi tespiti: Bir önceki adımda bulunan tüm aktif cihazlara "sysServices" değeri gönderilir. Dönen değere göre cihazın katman 2 (L2) mi yoksa katman 3 (L3) mü olduğu anlaşılır. Bu işlem için kullanılan algoritma Şekil 4.20'de gösterilmiştir. Öncelikle algoritmada "dönen_deger" isimli bir değer tanımlanır. Daha sonra "KomutSatiri.Calistir" isimli fonksiyon çağrılarak "sysServices" sorgusu çağrılır, dönen değer yönlendirici ise katman 3 cihazı diye, anahtar cihazı ise katman 2 cihazı diye, eğer "null" değeri dönerse son kullanıcı cihazı olarak işlem görür.

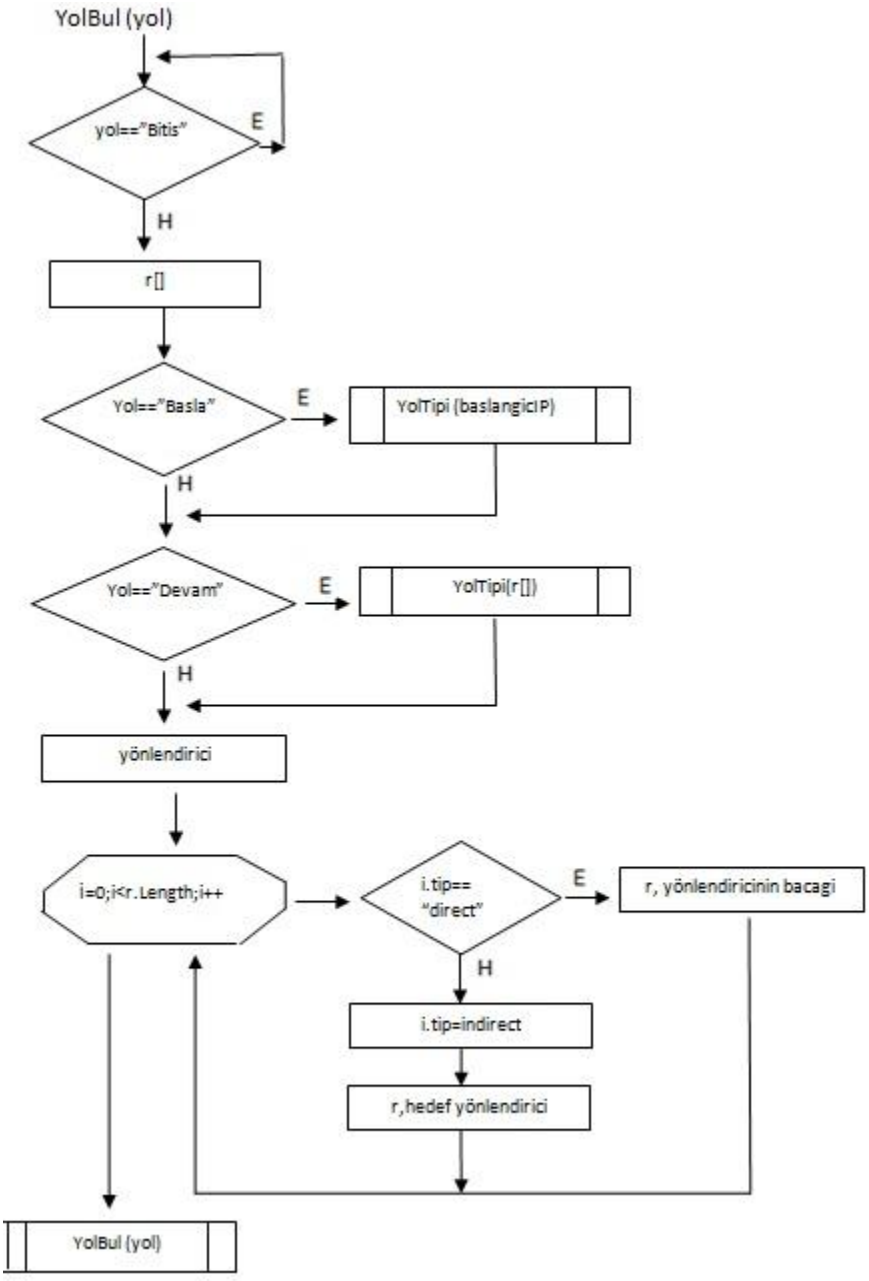


Şekil 4.20. Cihaz tipinin belirlenmesi algoritması

d. Bağlantı tespiti: Cihazlar arasındaki bağlantı tipini bulmak için kullanılan adımdır. Yönlendirici cihazlarından dönen “ipRouteType” değeri sonucuna göre “direct” olan IP’ler o yönlendiriciye, “indirect” olan IP’ler ise bağlı olunan hedefteki yönlendiriciye aittir. Anahtar cihazlardaki bağlantılarıda ise “ipNetToMediaNetAddress” den dönen değerlere ve AFT’e (Address Forwarding Table, Adres İletim Tablosu) bakarak bağlantı tespit edilir.

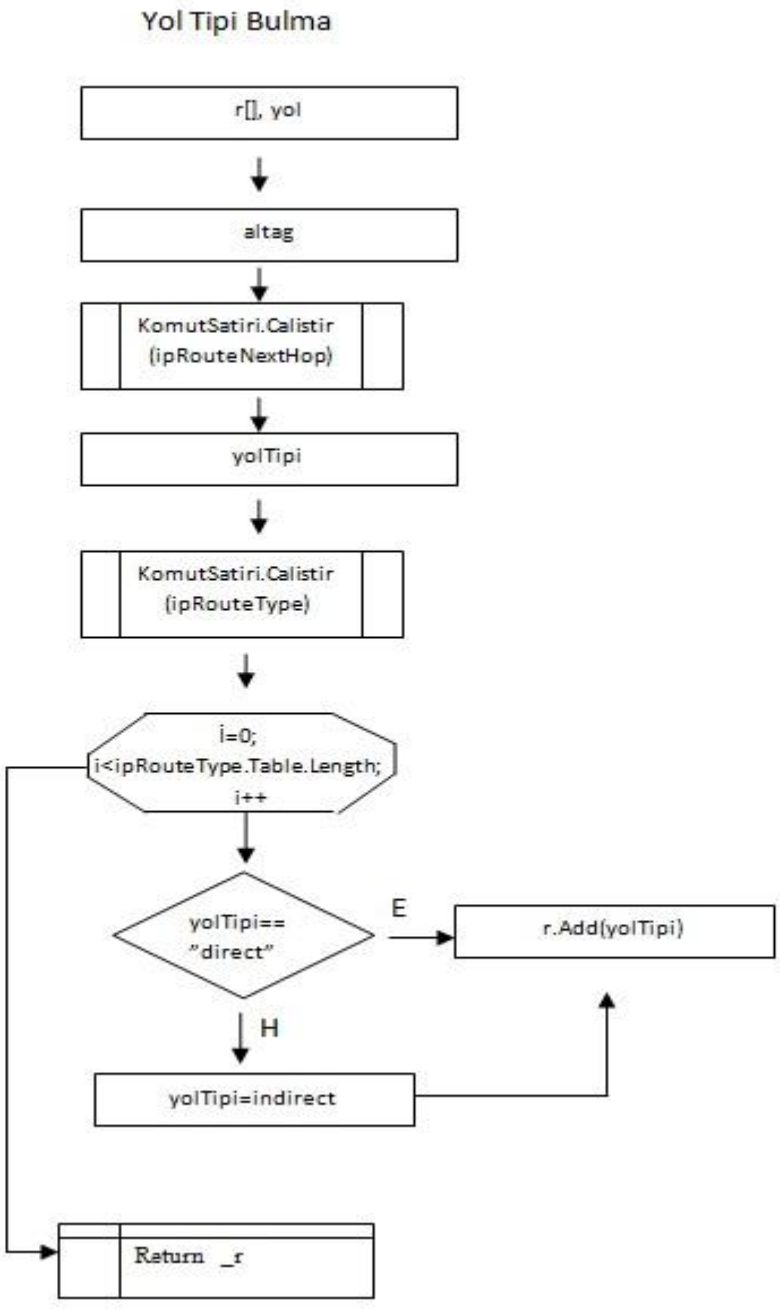
3. Katman cihazları arasında yol bulmak için kullanılan algoritma Şekil 4.21’de gösterilmiştir. Algoritmada, Şekil 4.22’deki algoritma yönlendiricilerin yol tipini belirlemek için bir alt fonksiyon olarak çağrılmaktadır. Şekil 4.22’deki algoritmada öncelikle r[] dizisi tanımlanmaktadır. Bu dizinin elemanları Şekil 4.21’deki algoritma

sonucunda dönen değerlere göre şekillenmektedir. Algoritmada daha sonra, önceden bu algoritmanın çalıştırılıp çalıştırılmadığının kontrolü yapılmaktadır. Algoritma ilk defa başlatılıyorsa Şekil 4.17'daki "Aktif cihaz bulma" algoritmasında bulunan ilk yönlendiriciyi referans alır ve işlemler o IP'ye göre yapılır. Algoritma daha önceden işletildiyse yönlendiriciler Şekil 4.22'deki algoritmadan dönen r dizisi elemanların sonucuna göre birbirlerine eklenir.



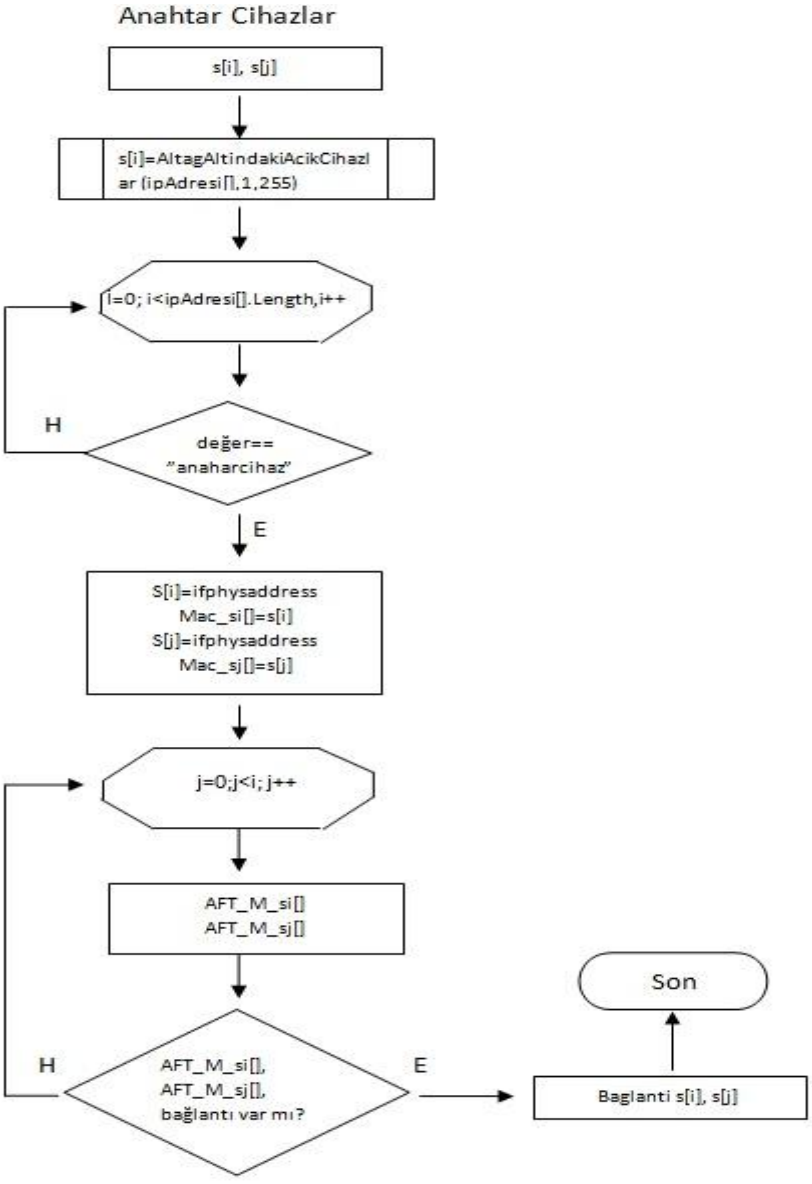
Şekil 4.21. Katman 3 cihazlarda yol bulma algoritması

Şekil 4.22'de gösterilen algoritma, Şekil 4.21'deki algoritmanın bir alt fonksiyonu gibi çalışmaktadır. Bu algoritmada yönlendiricilere gönderilen "ipRouteType" sorgusu sonucunda hangi IP'lerin "direct", hangi IP'lerin ise "indirect" olduğu bulunur.



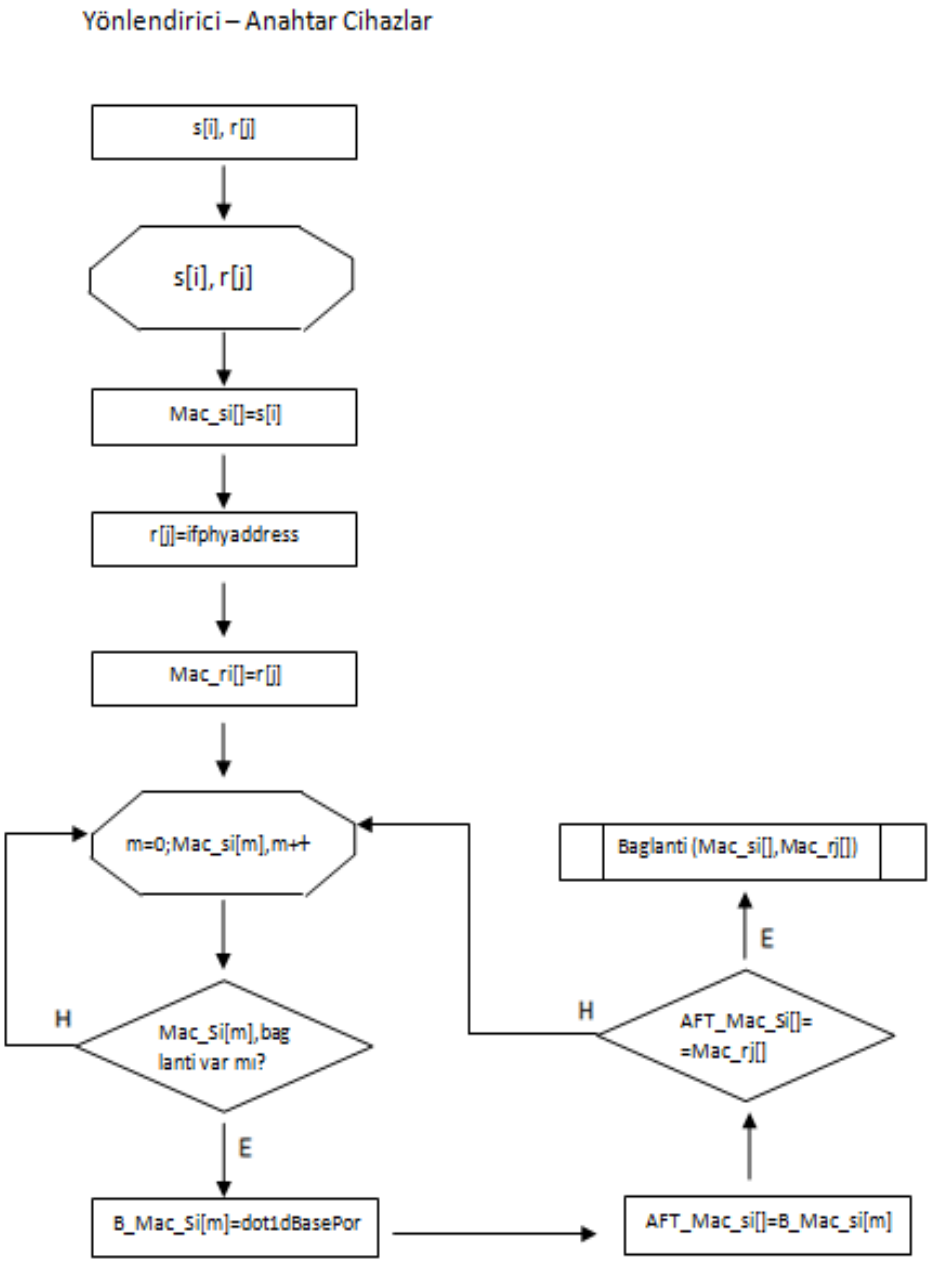
Şekil 4.22. Yol tipi bulma algoritması

Anahtar cihazlar arasındaki bağlantı için AFT tablosu kullanılmaktadır. AFT tablosundaki eşleştirmelere göre anahtar cihazlar arasında bağlantı var mı yok mu tespit edilebilir. Bunun için Şekil 4.23'deki algoritma kullanılmıştır. Algoritmada önce anahtar cihazlar için dizi tanımlanır. Daha sonra Şekil 4.19'daki algoritma çalıştırılır dönen IP değerleri diziye aktarılır. Daha sonra cihazın anahtar cihaz olup olmadığı kontrol edilir. Cihaz anahtar cihazsa arayüzleri bulunarak, döngü kurulur ve bu cihazları AFT tabloları karşılaştırılarak cihazlar arasında bağlantı olup olmadığı tespit edilir.



Şekil 4.23. Anahtar cihazlar arasındaki bağlantıyı bulma algoritması

Son olarak da Şekil 4.24'de yönlendirici cihazlar ve anahtar cihazlar arasındaki bağlantı tespiti için kullanılan algoritma gösterilmiştir. Algoritmada anahtar cihazlar ve yönlendirici cihazlar için diziler oluşturulmuş olup Şekil 4.23'deki algoritmayla aynı mantıkta tasarlanmıştır. Diğer algoritmadan farklı olarak, bu algoritmada yönlendirici ve anahtar cihazların mac adreslerinin birbirleriyle eşleşmesi durumunda bağlantı kurulur.



Şekil 4.24. Yönlendirici ve anahtar cihazlar arasındaki bağlantı bulma algoritması

4.3.3. Kullanılan MIB ve OID değerleri

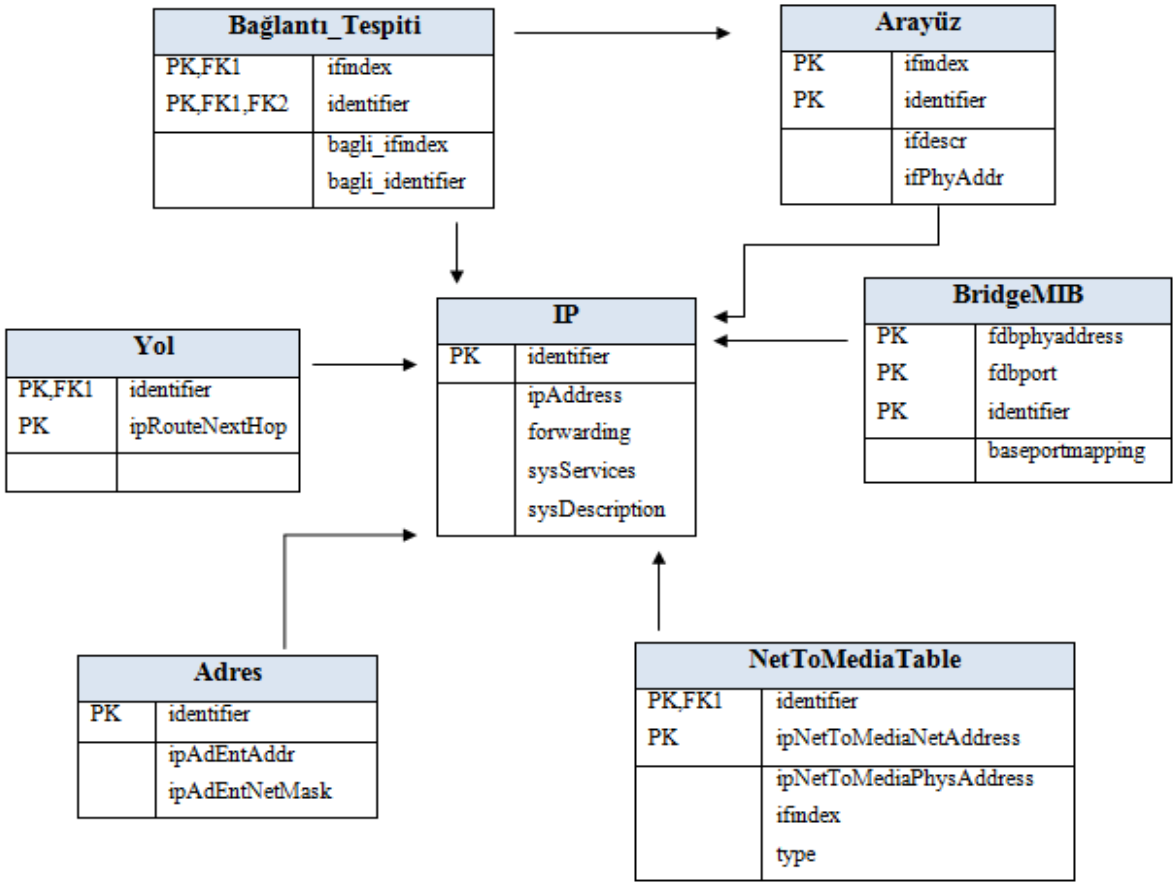
Uygulamada kullanılan MIB ve OID değerleri ve bunların uygulamada ne işe yaradıkları Tablo 4.4'de detaylı bir şekilde gösterilmiştir.

Tablo 4.4. Uygulamada kullanılan MIB nesneleri

MIB	MIB Nesnesi	OID Değeri	Görevi
MIBII (RFC 1213)	**System** → sysName	1.3.6.1.2.1.1.5	Cihaz ismini bulur.
	System → sysServices	1.3.6.1.2.1.1.7	Cihaz tipini bulur.
	System → sysDescr	1.3.6.1.2.1.1.1	Cihaz tanımı
	Iftable → ifIndex	1.3.6.1.2.1.2.2.1.1	Cihaz arayüzü
	İp → ipRouteTable → ipRouteNextHop	1.3.6.1.2.1.4.21.1.7	Bir sonraki atlama noktası
	Ip → ipRouteTable → ipRouteType	1.3.6.1.2.1.4.21.1.8	Yol tipini belirler.
	Ip → ipAddrTable → ipAdEntAddr	1.3.6.1.2.1.4.20.1.1	Bir cihazda çok IP adresi var mı?
	Ip → ipNetToMediaTable → ipNetToMediaNetAddress	1.3.6.1.2.1.4.22.1.3	ARP tablo girdileri
	Ip → ipNetToMediaTable → ipNetToMediaPhysAddress	1.3.6.1.2.1.4.22.1.2	IP adreslerini fiziksel adrese eşleştirir.
BRIDGE-MIB	**dot1dBrige** → dot1dBase → dot1dBasePortTable → dot1dBasePortEntry → dot1dBasePort	1.3.6.1.2.1.17.1.4.1.1	Ifindex girdilerini eşleştiririr.
	dot1dBrige → dot1dTp → dot1dTpFdbTable → dot1dTpFdbEntry → dot1dTpFdbAddress	1.3.6.1.2.1.17.4.3.1.1	Adres İletimTablosunu adresler.
	dot1dBrige → dot1dTp → dot1dTpFdbTable → dot1dTpFdbEntry → dot1dTpFdbStatus	1.3.6.1.2.1.17.4.3.1.3	Arayüzün durumu

4.3.4. İlişkisel veritabanı

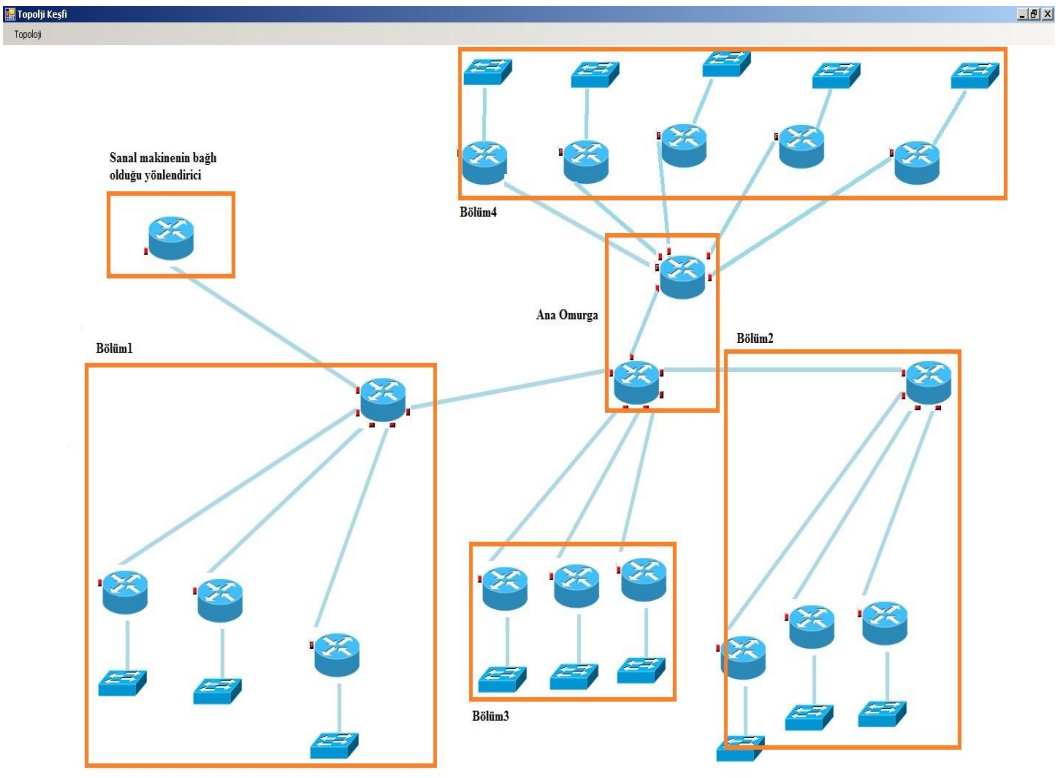
Bu bölümde uygulamada kullanılan veritabanı anlatılmıştır. MySQL ile oluşturulmuş olan veritabanı 7 tablodan oluşmaktadır. Şekil 4.24'de şematize edilen veritabanında tüm tablolar ortadaki IP tablosuyla ilişkilidir. Daha önceki bölümlerde gösterilen algoritmalardan dönen değerler veritabanındaki ilgili alanlara kaydedilir. Anahtar cihazlar için "BridgeMIB", "NetToMediaTable" ve "Adres" tabloları kullanılmıştır. Yönlendirici cihazlar için ise "Arayüz", "Yol" ve "IP" tabloları kullanılmaktadır. "Baglantı_Tespiti" tablosu ise hem yönlendirici hem de anahtar cihazlar için kullanılmaktadır.



Şekil 4.25. Uygulamada kullanılan ilişkisel veritabanı

4.3.5. Program çıktısı

VMWARE Workstation ve GNS3 sanal ortamlarının entegrasyonunun sağlanarak oluşturulan Şekil 4.1'deki örnek kurumsal ağ topolojisinin, uygulama sonucundaki ekran görüntüsü Şekil 4.26'da verilmiştir. Şekil 4.15'deki genel algoritma yapısına bağlı olarak ve ilgili alt algoritmaların da çalıştırılmasıyla gerçeklenen uygulama Visual Studio 2010'da C# programlama diliyle yazılmıştır. Veritabanı olarak MySQL kullanan uygulamada, SNMP sorgu tekniği olarak Net-SNMP kütüphanesinin "snmpwalk" uygulaması seçilmiştir. Ayrıca SNMP sorgularının ve temel komutların daha kolay yönetilebilmesi için WebNMS SNMP API'si de kullanılmıştır. Kurumsal ağ kavramına bağlı kalınarak oluşturulan Şekil 4.1'deki örnek modeldeki ana omurga, sanal makinenin bağlı olduğu yönlendirici, bölüm1, 2, 3 ve 4 alanları, kullanılan algoritmalar ve geliştirme ortamları doğrultusunda Şekil 4.26'daki ekran görüntüsü elde edilmiştir.



Şekil 4.26. Uygulama çalıştırıldıktan sonra elde edilen ekran görüntüsü

BÖLÜM 5. SONUÇLAR

Bu tez çalışmasıyla öncelikle ağ yönetim sistemleri, bir ağ yönetim protokolü olan SNMP, daha sonra da uygulamada kullanılan sanal ortamlar (GNS3 ve VMWARE Workstation) ve son olarak da geliştirilen uygulama anlatılmıştır. Uygulamada kullanılan teknoloji bileşenleri ile daha önceden yapılan akademik çalışmalar ve ticari çalışmalar incelenip, uygulamaya yönelik farklı bir yöntem izlenmiştir.

Bu çalışmada farklı sanal ortamlardan ağ modelleme yazılımı olan GNS3 ile sanal makine yazılımı olan VMWARE Workstation birbirleriyle entegre edilerek bir ağ topoloji keşfi uygulaması yapılmıştır. Uygulamada kullanılan örnek kurumsal ağ SNMPv3 ile modellenerek merkezi yönetim mimarisi kullanılmış, güvenlik yönetimi, performans yönetimi ve konfigürasyon yönetimi gibi ağ yönetim alanları temel alınmıştır. Kurumsal ağın modellenmesi bittikten sonra uygulamayı geliştirirken SNMP'nin çalışma mimarisine bağlı kalarak kullanıcı arayüzüne, MIB II ile uyumlu SMIv2 dilini kullanan bir program yazılmıştır. Visual Studio 2010 ortamında C# ile yazılan programda SNMP sorgu tekniği olarak Net-SNMP kütüphanesinin "snmpwalk" uygulaması tercih edilmiştir. Uygulama sonucunda, bir ağ topolojisi oluşturmak için fiziksel bir ortamın gerekli olmadığı aynı özellikte, aynı büyüklükte ağlarında sanal ortamlarda oluşturularak yönetimlerinin kolay olduğu gözlemlenmiş ve bu tarz çalışmaların bilgisayar ağları gibi derslerde kolayca gerçekleştirilebileceği gösterilmiştir.

Sonuç olarak, sanallaştırmanın kullanım alanın arttığı günümüzde, büyük kurumların dahi kendi sistemlerini sanal ortamlar üzerine taşıyarak fiziksel ortamda oluşabilecek hata risklerine karşı önlem almaktadırlar. Bu yüzden ağ yönetimi ile ilgili sanallaştırma ortamları üzerinde daha fazla uygulama yapılması gerekmektedir.

KAYNAKLAR

[1] PANDEY, S., CHOI, M., WON, Y., HONG, J., SNMP-based enterprise IP network topology discovery, International Journal of Network Management 2011; Volume 21, Issue 3, Pages :169-184, May 2011

[2] SIAMWALLA, R., SHARMA, R., KESHAV, S., Discovering internet topology, Technical report, Cornell University, May 1999

[3] LOWEKAMP, B., O'HALLARON, DR., GROSS, TR., Topology discovery for large Ethernet networks. In ACM SIGCOMM, San Diego, CA, 237–248, August 2001

[4] HOANG, T., BASIN, D., KURUMA, H., ABRIAL, J., Development of a Network topology discovery algorithm, Journal Science of Computer Programming, Volume 74 Issue 11-12, November 2009

[5] XIAOPING, LI., YINXING, LI., JINGHUI, C., QIONG, Xu., The study of network layer topology discovery algorithm for optimization problem based on SNMP, Networked Computing and Advanced Information Management (NCM), 2010 Sixth International Conference, Issue 16-18, Pages: 161-163, Aug. 2010

[6] Monitoring, aggregation and filtering for efficient management of virtual networks, Network and Service Management (CNSM), 2011 7th International Conference , Issue: 24-28, Page :1-7, Oct. 2011

[7] Management of Virtual Network Resources and Services, Systems Engineering (ICSEng), 2011 21st International Conference, Issue: 16-18, Page: 233-238, Aug. 2011

[8] NMAP, http://nmap.org/book/man.html [Ziyaret Tarihi: 15.08.2011]

[9] Product Documentation, http://www.whatsupgold.com/support/guides.aspx [Ziyaret Tarihi: 18.08.2011]

[10] E. MELLQUIST, SNMP++: An Object-Oriented Approach to Developing Network Management Applications, Prentice Hall PTR, 1997; 5:10:20-28

[11] R.MAURO, J.SCHMIDT, Essential SNMP, O'Reilly $ Associates, Inc., 6:10-13:43-47, July 2001

[12] JIZONG LI, WEB-based Network Monitoring Using SNMP, CGI and CORBA, in University of Manitoba, 6:10-14:22, August 1999

[13] A Simple Network Management Protocol (SNMP) http: // www .ietf. org/ rfc/ rfc1157.txt [Ziyaret Tarihi: 22.07.2011]

[14] A Simple Network Management Protocol (SNMP) http://www.ietf.org/rfc/ rfc1157.txt [Ziyaret Tarihi: 22.07.2011]

[15] SNMPv3 Applications http://www.ietf.org/rfc/rfc2273.txt [Ziyaret Tarihi: 24.07.2011]

[16] The SNMP Message Format http://www.rane.com/note161.html [Ziyaret Tarihi: 01.08.2011]

[17] SNMP Technology White Paper http://www.h3c.com/portal/ Products___Solutions/Products/Switches/H3C_S5810_Series_Switches/Whit e_Paper/200912/656293_57_0.htm [Ziyaret Tarihi: 07.08.2011]

[18] SNMP http://www.bidb.itu.edu.tr/?d=1034 [Ziyaret Tarihi: 07.07.2011]

[19] Sanallaştırma ve Sanallaştırmanın Büyük Oyuncusu VMWARE http://www.enderunix.org/docs/Sanallastirma.pdf [Ziyaret Tarihi: 09.08.2011]

[20] The Benefits of Virtualization for Small and Medium Businesses http://www.vmware.com/files/pdf/VMware-SMB-Survey.pdf [Ziyaret Tarihi: 15.08.2011]

[21] Virtualization http://www.vmware.com/virtualization/virtual-machine.html [Ziyaret Tarihi: 09.08.2011]

[22] VMWARE Workstation User's Manuel http://www.vmware.com/pdf/ws71_manual.pdf [Ziyaret Tarihi: 15.08.2011]

[23] GNS3, http://www.gns3.net/documentation [Ziyaret Tarihi: 25.08.2011]

[24] Wireshark Developer's Guide http://www.wireshark.org/docs/wsdg_html_chunked/ [Ziyaret Tarihi: 27.08.2011]

[25] Wireshark http://tr.wikipedia.org/wiki/Wireshark [Ziyaret Tarihi: 27.08.2011]

[26] OSPF Version 2, http://www.ietf.org/rfc/rfc1247.txt [Ziyaret Tarihi: 10.08.2011]

[27] Net-SNMP Tutorials http://www.net-snmp.org/wiki/index.php/Tutorials [Ziyaret Tarihi: 02.09.2011]

[28] WebNMS, http://www.webnms.com/netsnmp/datasheet.html [Ziyaret Tarihi: 09.09.2011]

[29] Selçuk Üniversitesi Ağ Altyapısı http://www.bim.selcuk.edu.tr/teknik_altyapi.html [Ziyaret Tarihi: 09.08.2011]

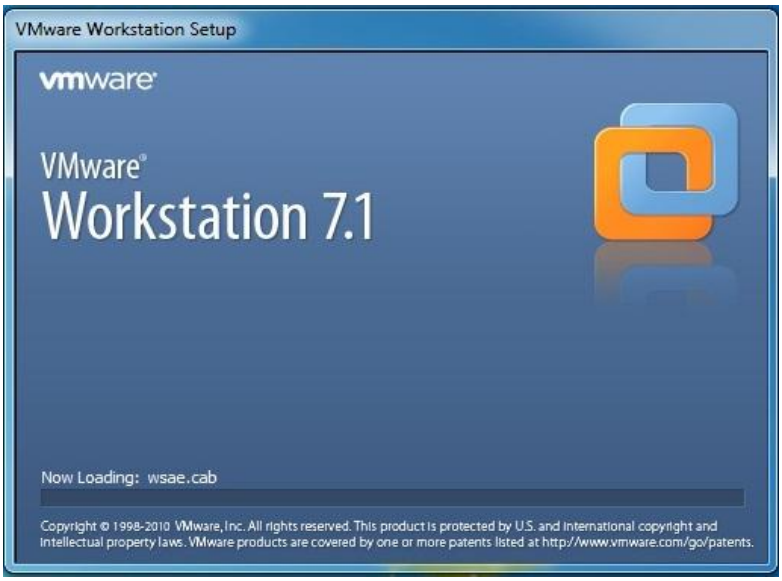
[30] Tübitak Bilişim Müdürlüğü Bilişim Sistemleri Yapısı http://www.tubitak.gov.tr/sid/947/pid/945/cid/7029/index.htm;jsessionid=44F91C95A6AC28A62CADD7CD85EEC08E [Ziyaret Tarihi: 14.08.2011]

EK – A

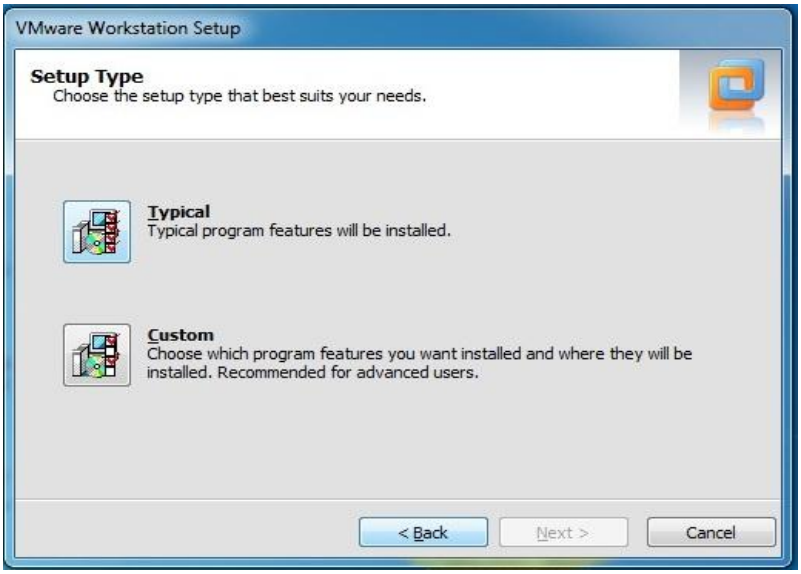
VMWARE Workstation Kurulumu

Tezin bu kısmında VMWARE Workstation programının kurulum aşamalarından ve daha sonrasında ise nasıl sanal makine kurulacağından bahsedilmiştir.

Kurulum işlemi başlatılır, daha sonra kurulum tipi seçilir. Bu aşamalar Şekil A.1 ve A.2'de gösterilmiştir.

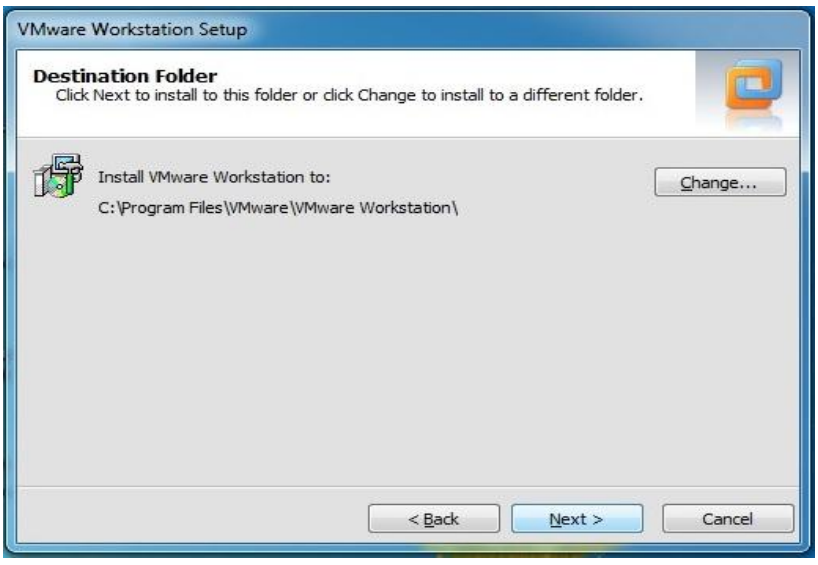


Şekil A.1. WMWARE Workstation programı kurulum başlangıcı



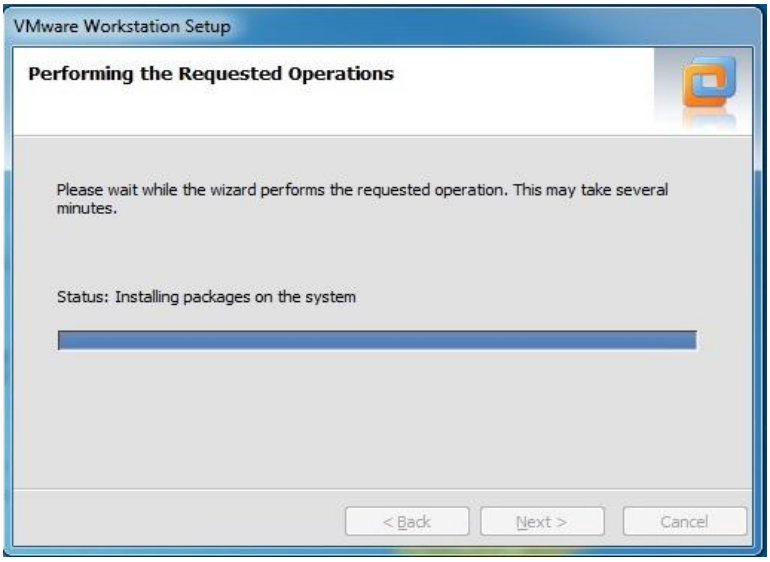
Şekil A.2. Kurulum tipi

Programın kurulum dosyalarının hangi dizinde saklanacağı Şekil A.3'de gösterilmiştir.



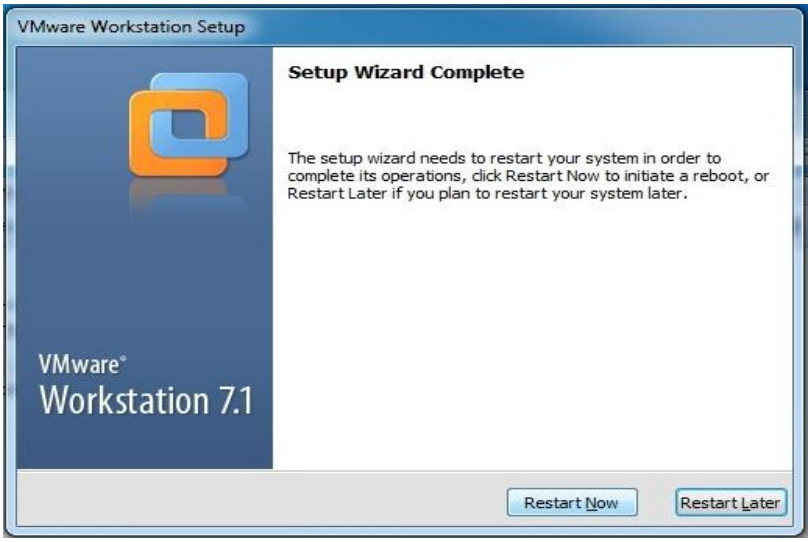
A.3. Yol seçimi

Çıkan pencerelerde "next" sekmelerini tıkladıktan sonra Şekil A.4'deki kurulum aşaması tamamlanmak üzeredir.



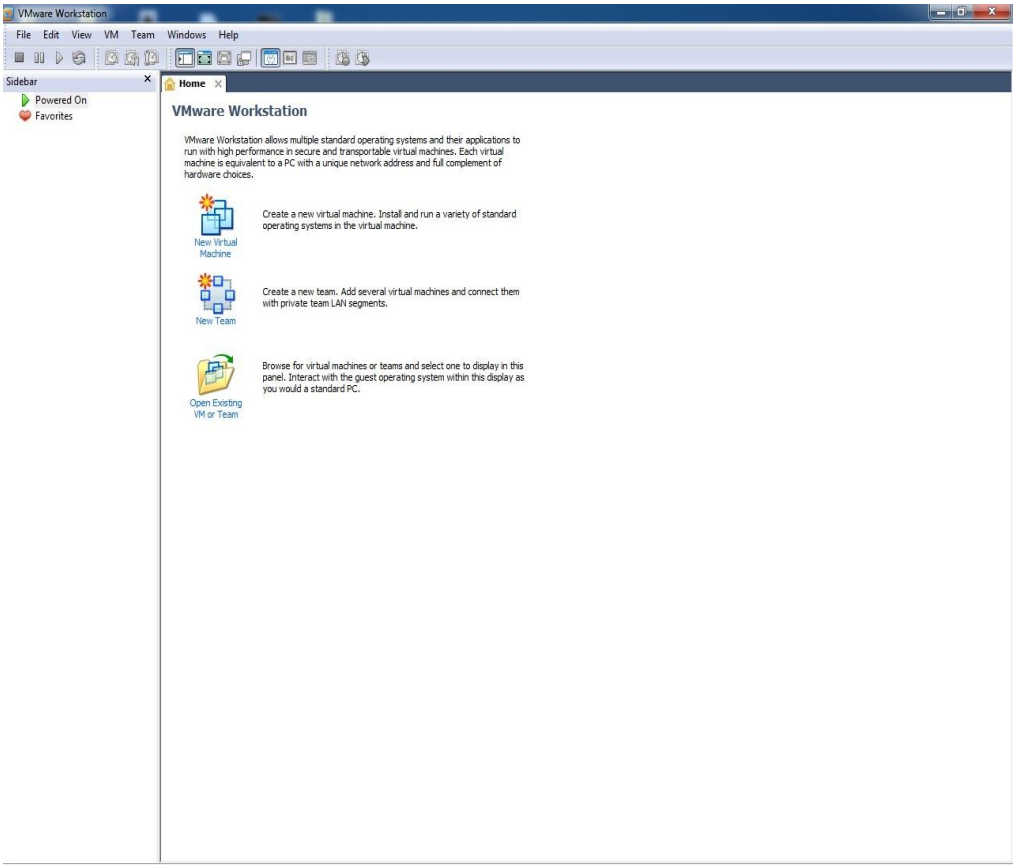
Şekil A.4. Kurulum süreci

Bu aşamadan sonra çıkan pencerede şifre girildikten sonra kurulum adımımız tamamlanmış olup Şekil A.5'deki sistemimizi yeniden başlatmamız gerekmektedir.



Şekil A.5. Yeniden başlatma

Sistemimiz yeniden başlatıldıktan sonra programı açınca karşımıza Şekil A.6'daki gibi bir ekran gelir.



Şekil A.6. Programın genel görünümü

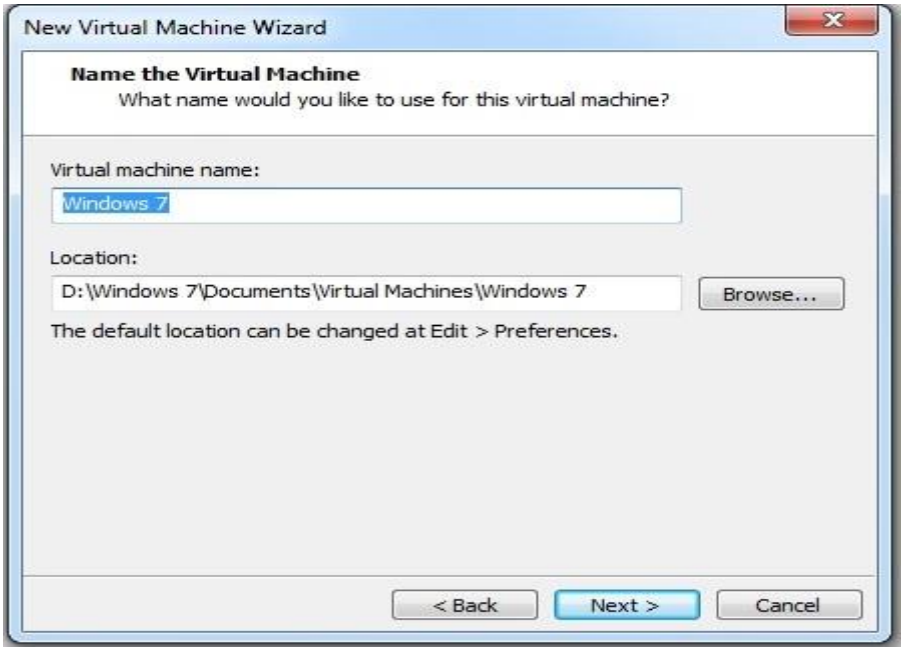
Bu bölümden sonra, kendi uygulamamızda kullandığımız konuk işletim sistemi kurulumu anlatılacaktır. Konuk işletim sistemi kurulumu için gerekli aşamalar sırasıyla anlatılmaktadır. Kurulum işlemine başlamak için Şekil A.6'daki "New Virtual Machine" sekmesine tıklanır ve karşımıza Şekil A.7'deki gibi kurulum tipi penceresi gelir.

Şekil A.7. Kurulum tipi

Şekil A.8'de kurulumun hangi dizinden yapılacağı seçimi gösterilmektedir.

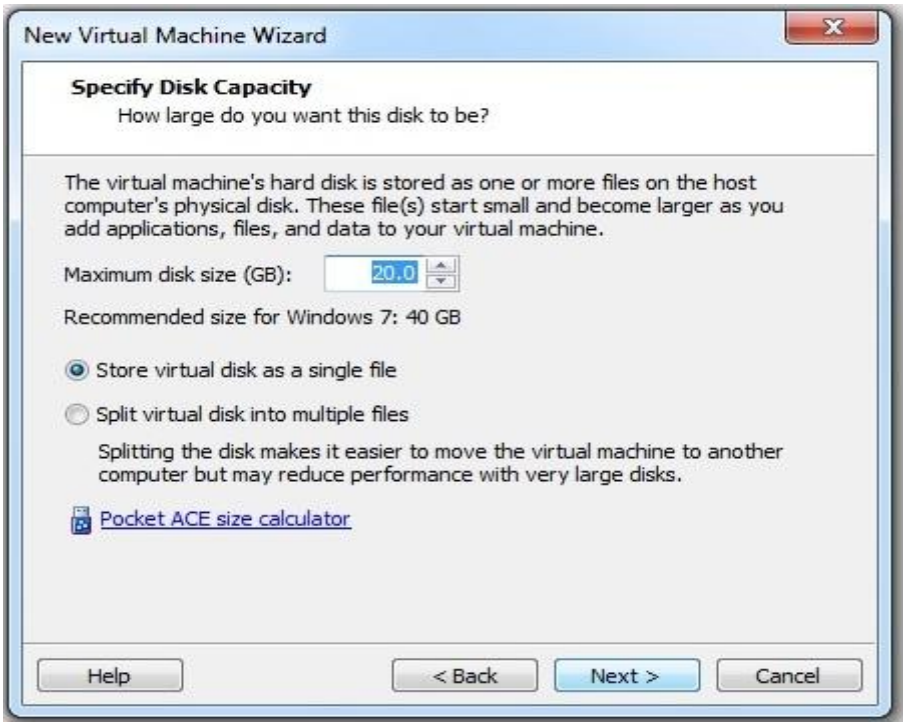
Şekil A.8. Kurulumun hangi dizinden yapılacağının seçilmesi

Şekil A.9'da konuk işletim sistemimize isim verme işlemi gösterilmiştir.



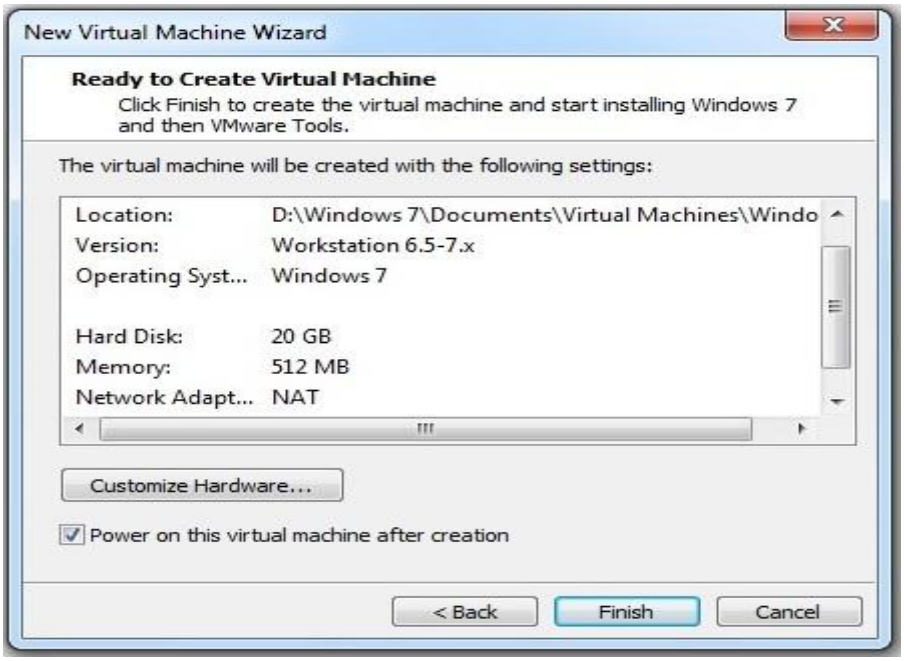
Şekil A.9. Konuk işletim sistemine isim verme

Şekil A.10'da konuk işletim sistemi için sabit disk alanı ayırma işlemi gösterilmektedir.



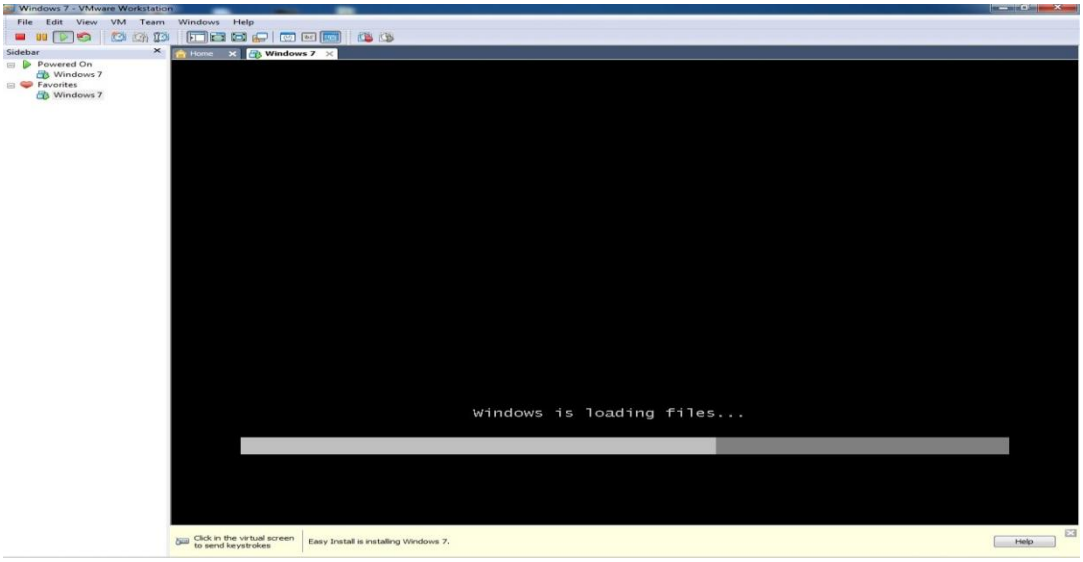
Şekil A.10. Konuk İşletim Sistemi İçin Alan Ayırma

Şekil A.11'de, konuk işletim sisteminin kuruluma geçmeden önce hazır duruma geldiğini gösteren ekran verilmiştir.

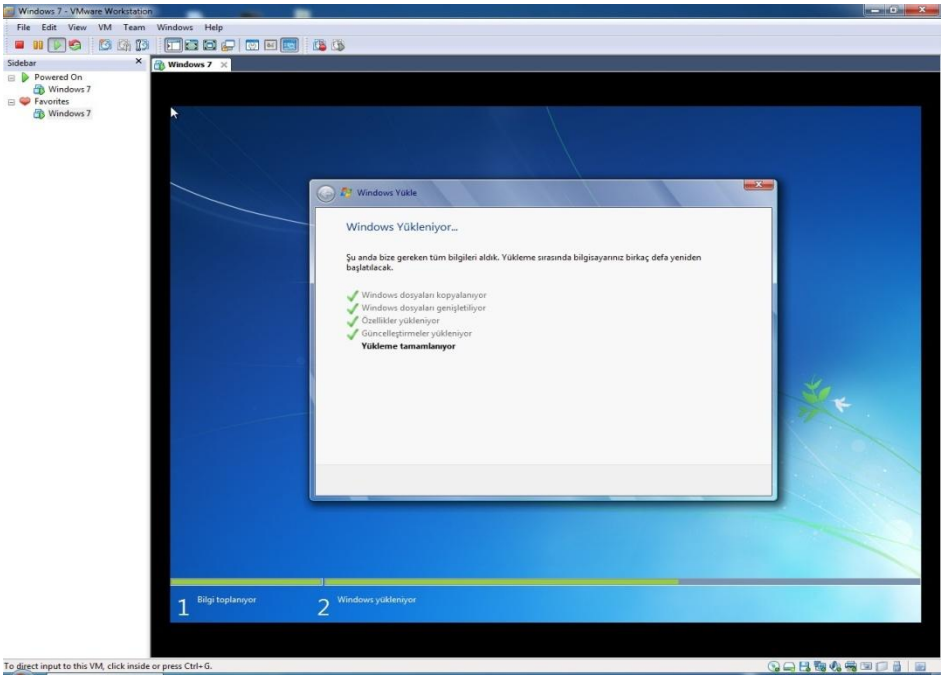


Şekil A.11. Kuruluma hazır ekranı

Bundan sonra konuk işletim sistemi kurulumu başlamaktadır. Şekil A.12'de kurulum başlangıcı ekranı gösterilmektedir.

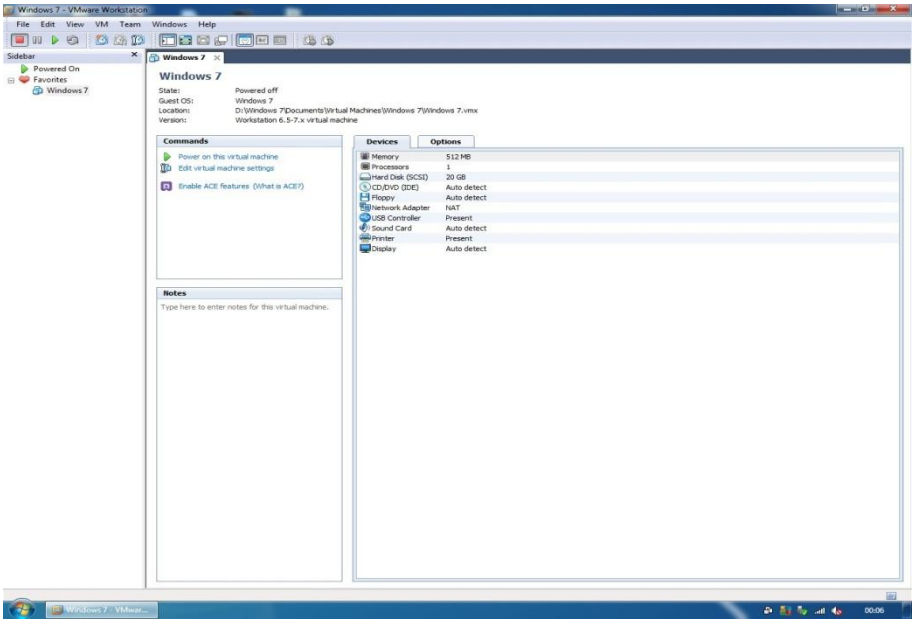


Şekil A.12. Kurulum başlangıcı



Şekil A.13. Kurulum ekranı

Kurulum aşamaları bittikten sonraki ekran görüntüsü Şekil A.14'de gösterilmiştir.



Şekil A.14. Kurulum sonrası genel görünüm

EK – B

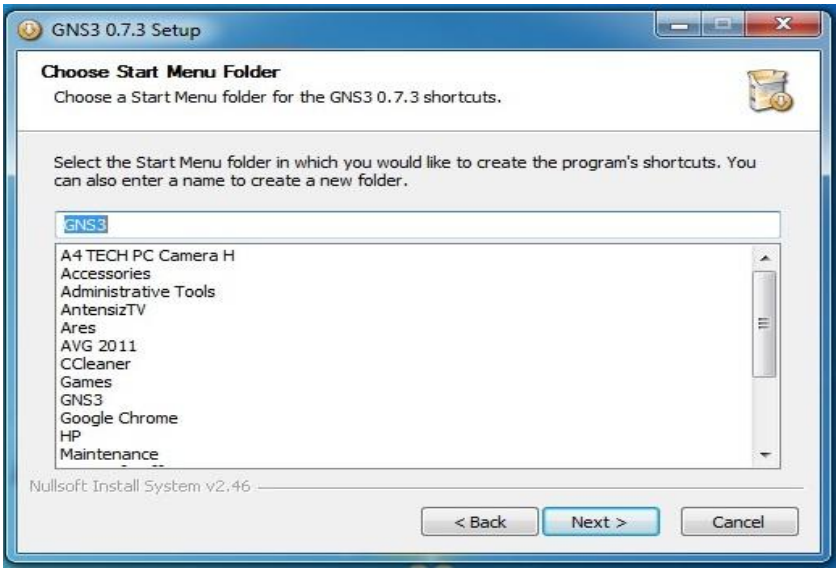
GNS3 Kurulumu

Tezin bu kısmında, uygulamada kullanılan kurumsal ağı topolojisinin oluşturulduğu GNS3 modelleme programının kurulum aşamaları anlatılacaktır.

Şekil B.1'de kurulum işleminin başlatılma ekranı gösterilmiştir.

Şekil B.1. Kurulum işlemi başlatma

Şekil B.2'de başlangıç dosyası seçimi ekranı gösterilmektedir.

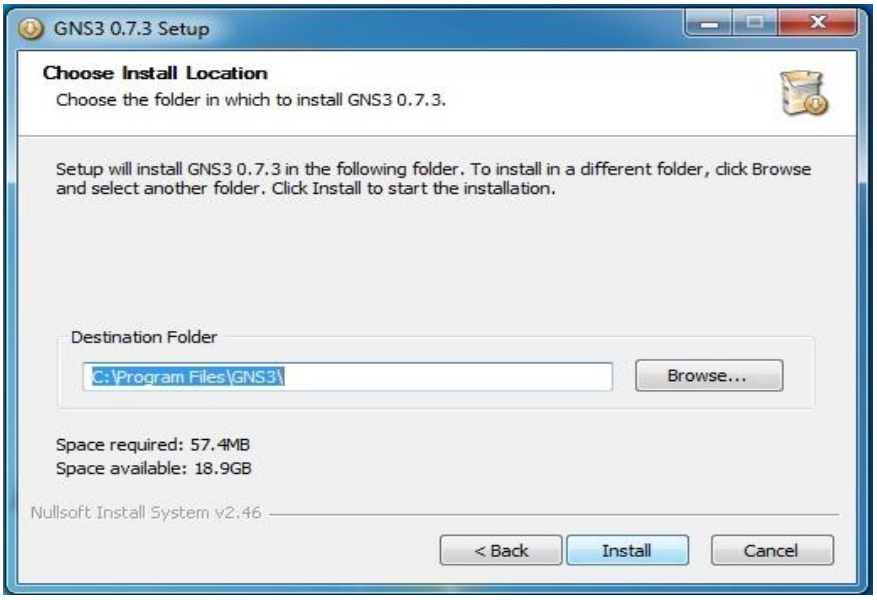


Şekil B.2. Başlangıç dosyası seçimi

Şekil B.3'de kurulum için hangi bileşenlerin seçileceği ekranı gösterilmektedir.

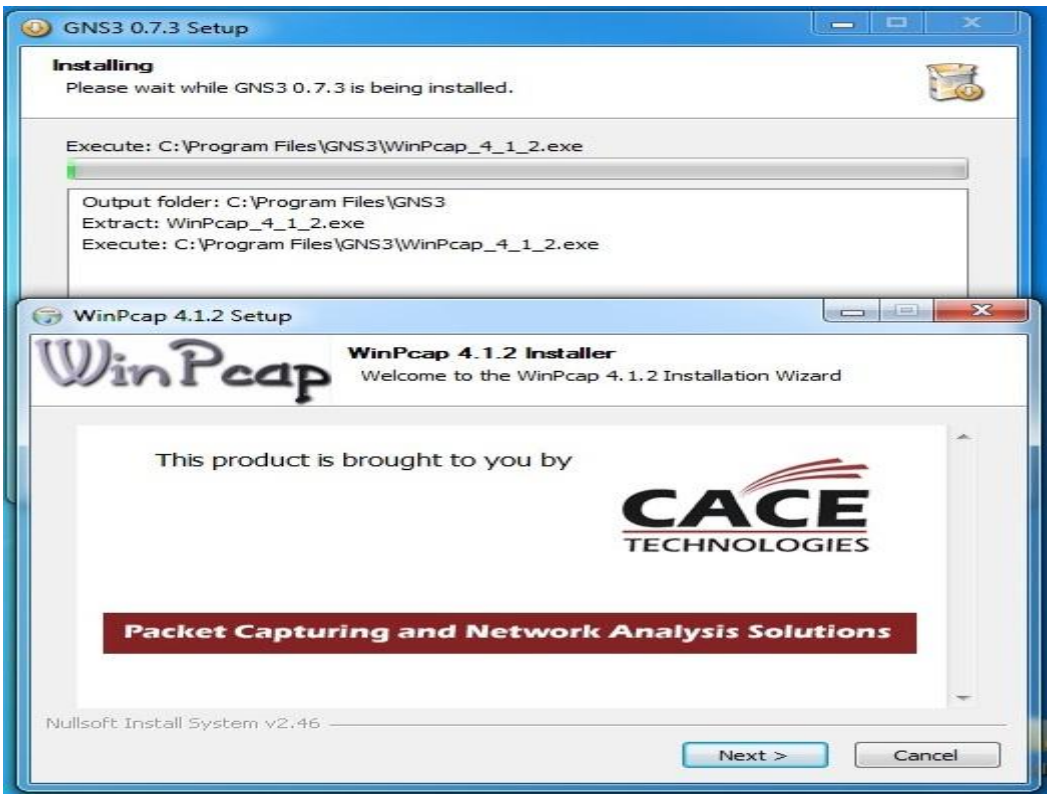
Şekil B.3. Bileşen seçme

Şekil B.4'de kurulumun hangi dizine yapılacağının seçimi gösterilmektedir.



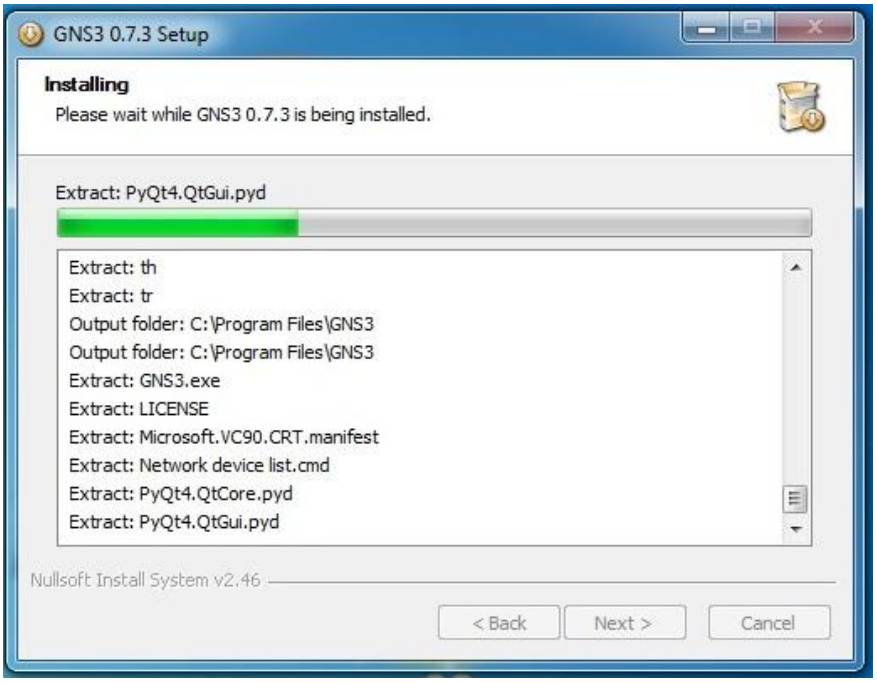
Şekil B.4. Dizin seçme

GNS3'ün kurulabilmesi için önce "WinPcap" yazılımının kurulması gerekmektedir. Şekil B.5'de bu programın kurulum ekranı gösterilmiştir.



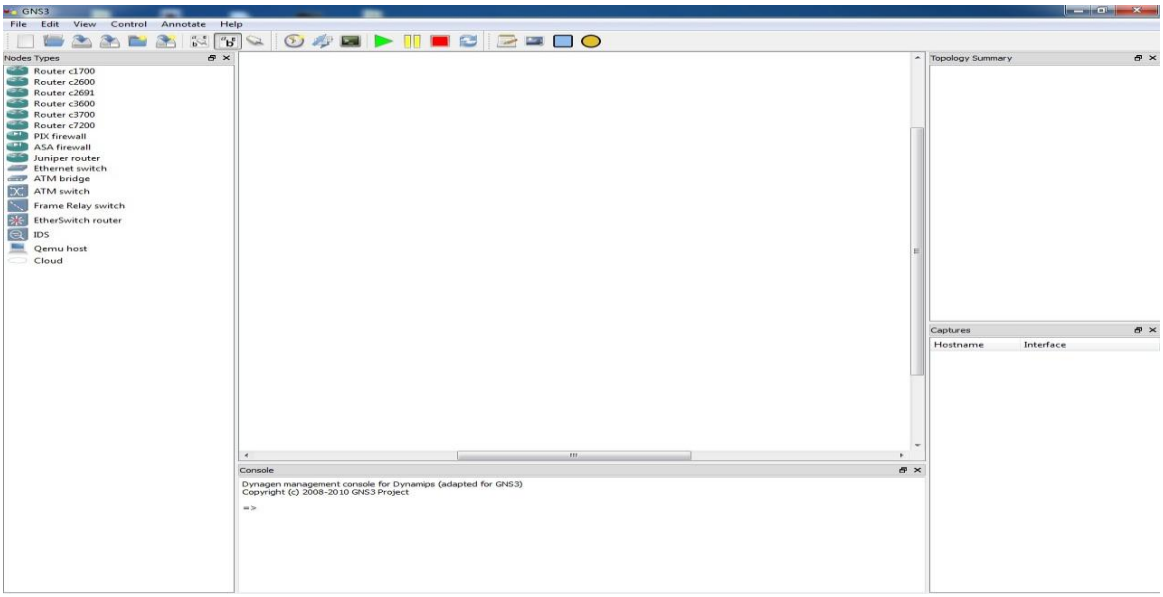
Şekil B.5. WinPcap'in kurulumu

Şekil B.6'da WinPcap programının kurulumundan sonra ,GNS3 programının kurulumunun devam ettiğini gösteren ekran görüntüsü verilmiştir.



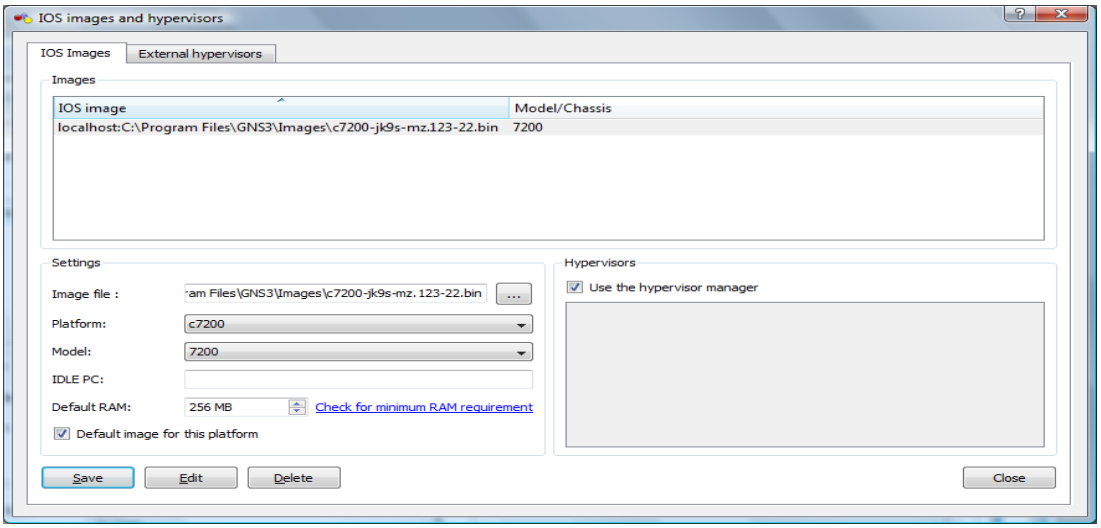
Şekil B.6. Kurulum devam

Kurulum işlemleri tamamlandıktan sonra Şekil B.7'deki gibi bir ekran görüntüsü çıkar.



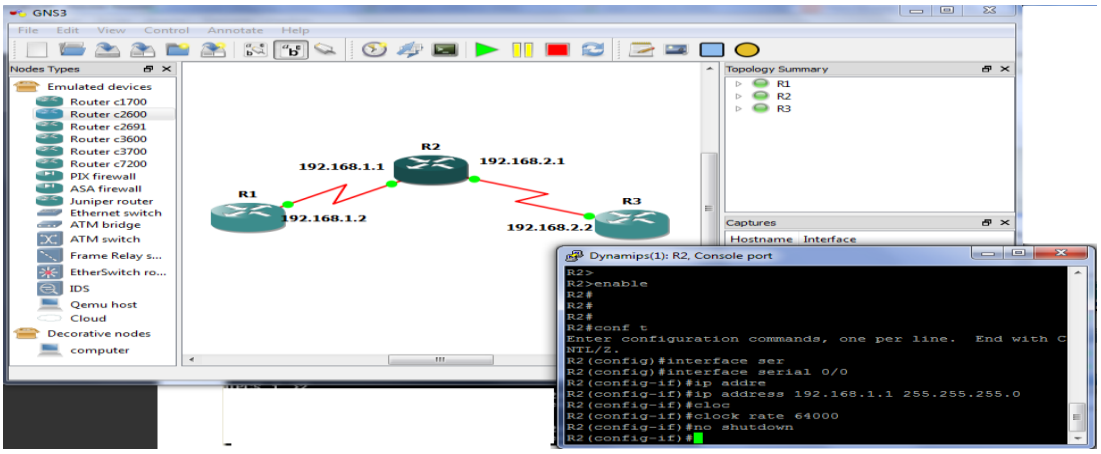
Şekil B.7. Kurulum sonrası genel görünüm

Programın kurulumu tamamlandıktan sonra cihazların IOS'larını (Internetwork Operating System) yüklememiz gerekmektedir. Şekil B.8'de IOS yükleme işlemi gösterilmiştir.



Şekil B.7. IOS yükleme işlemi

IOS işlemi yüklemesi bittikten sonra artık program kullanılmaya hazır hale gelmiştir. Şekil B.8'de kurulum işlemi bittikten sonra oluşturulmuş bir konfigürasyon örneği gösterilmiştir.

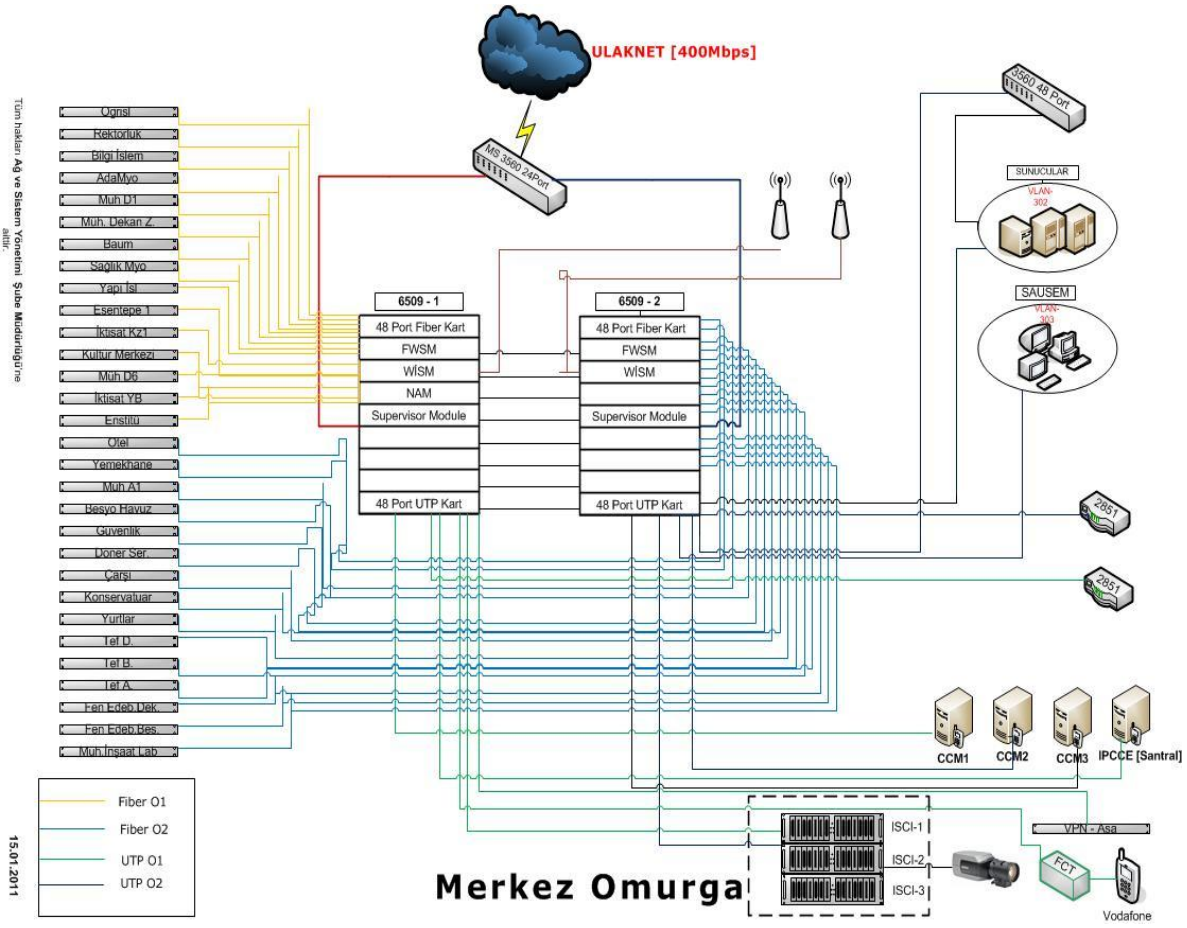


Şekil B.8. Konfigürasyon örneği

EK – C

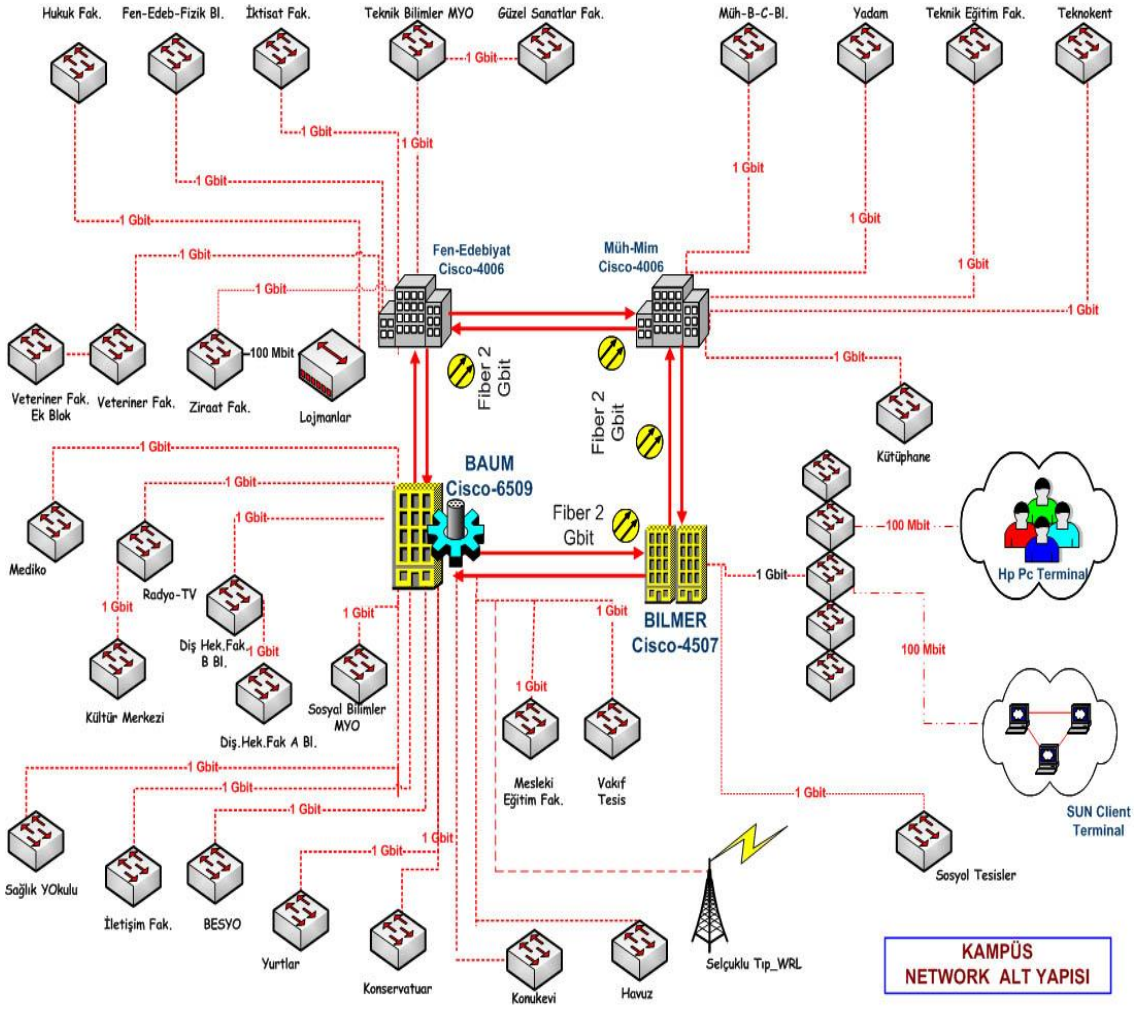
İncelenen Kurumsal Ağ Yapıları

Tezin bu kısmında, uygulamada oluşturulan kurumsal ağ topolojisine örnek teşkil etmesi için incelenen bazı ağ yapıları anlatılmıştır. Şekil C.1'de Sakarya Üniversitesi'nin kurumsal ağ yapısı gösterilmektedir. Ortada iki tane Cisco 6509 cihazının stand-by moda çalışarak diğer cihazları yönetmesiyle oluşan bir yapıda çalışan bu yapı tezimizde kullandığımız yapıya örnek teşkil etmiştir.



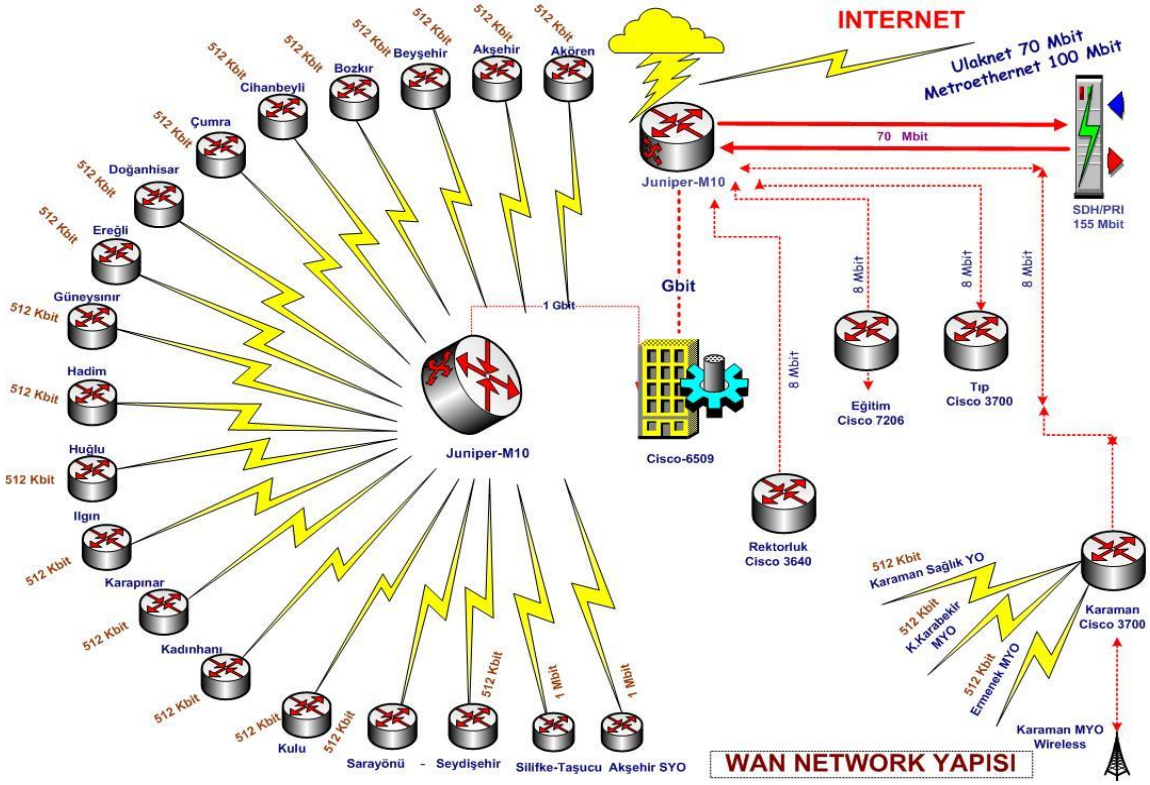
Şekil C.1. Sakarya Üniversitesi kurumsal ağ yapısı

Şekil C.2'de ise Selçuk Üniversitesi'nin hem yerel alan ağındaki yapısı hem de geniş alan ağındaki topoloji yapısı incelenmiştir. Yerel alan ağı, ortada bir tane Cisco 6509 cihaz etrafında toplanan yönlendirici ve anahtar cihazlar ile modellenmiştir.



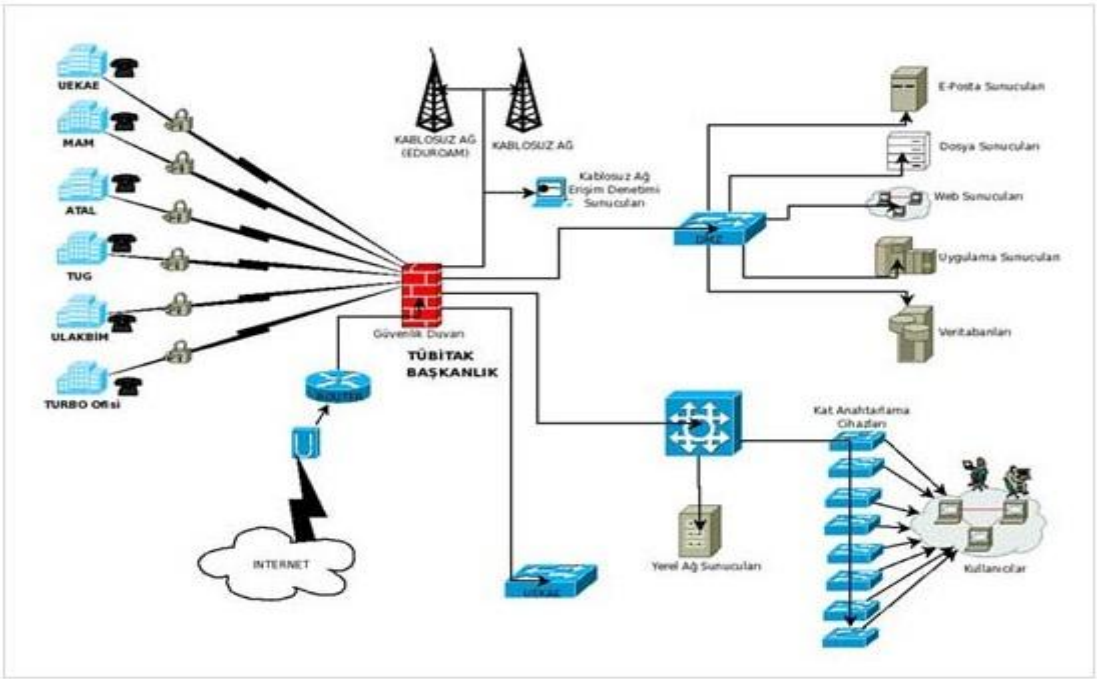
Şekil C.2. Selçuk Üniversitesi yerel alan ağı yapısı [29]

Şekil C.3'de ise Selçuk Üniversitesi'nin farklı kampüsleriyle haberleşebilmesi için oluşturulan geniş alan ağı yapısı gösterilmiştir. Bu yapıda Şekil C.2'deki Cisco 6509 cihazı diğer kampüslerin girişlerindeki yönlendiricilerle bağlantılı şekilde çalışmaktadır.



Şekil C.3. Selçuk Üniversitesi geniş alan ağı yapısı [29]

Şekil C.4'de ise Tübitak'ın Bilişim Müdürlüğü'nün bilişim sistemleri yapısı gösterilmektedir.



Şekil C.4. Tübitak Bilişim Müdürlüğü bilişim sistemleri yapısı [30]

EK – D

Uygulamanın Konfigürasyonu

Tezin bu bölümünde, Şekil 4.1'de verilen kurumsal ağ topolojimizde kullanılan cihazların konfigürasyonları verilecektir. Sırasıyla, Şekil 4.1'deki 1 numarayla gösterilen alandaki cihazlar, 3,4,5 ve 6 numarayla gösterilen alanlardaki ana yönlendiricilerin konfigürasyon bilgileri verilecektir. Şekil 4.1'de, 1 numarayla gösterilen alanda sol tarafta bulunan Cisco 6509 cihazı Şekil 4.2'de verildiği için, Şekil D.1'de sağ taraftaki cihazın konfigürasyonu verilmiştir. Tüm şekillerde 1 numaralı alanlar o cihazın arayüz konfigürasyonunu, 2 numarayla ifade edilen alanlar OSPF konfigürasyonunu, 3 numarayla ifade edilen alanlar ise konfigürasyon bilgisinin kayıt edildiğini ifade eder.

```
R2>enable                                                    1
R2#configure terminal
R2(config)#hostname  ana_omurga_sag
ana_omurga_sag(config)#interface serial 0/0
ana_omurga_sag(config-if)#no shutdown
ana_omurga_sag(config-if)#ip address 192.168.15.2   255.255.255.0
ana_omurga_sag(config-if)#clock rate 64000
ana_omurga_sag(config-if)#exit
ana_omurga_sag(config)#interface serial 0/1
ana_omurga_sag(config-if)#no shutdown
ana_omurga_sag(config-if)#ip address 192.168.16.2   255.255.255.0
ana_omurga_sag(config-if)#clock rate 64000
ana_omurga_sag(config-if)#exit
```

Şekil D.1. Ana omurgadaki sağdaki cihazın konfigürasyonu

```
ana_omurga_sag(config)#interface serial 0/2
ana_omurga_sag(config-if)#no shutdown
ana_omurga_sag(config-if)#ip address 192.168.17.2   255.255.255.0
ana_omurga_sag(config-if)#clock rate 64000
ana_omurga_sag(config-if)#exit
ana_omurga_sag(config)#interface serial 0/3
ana_omurga_sag(config-if)#no shutdown
ana_omurga_sag(config-if)#ip address 192.168.18.2   255.255.255.0
ana_omurga_sag(config-if)#clock rate 64000
ana_omurga_sag(config-if)#exit
ana_omurga_sag(config)#interface serial 0/4
ana_omurga_sag(config-if)#no shutdown
ana_omurga_sag(config-if)#ip address 192.168.19.2   255.255.255.0
ana_omurga_sag(config-if)#clock rate 64000
ana_omurga_sag(config-if)#exit
ana_omurga_sag(config)#interface serial 0/5
ana_omurga_sag(config-if)#no shutdown
ana_omurga_sag(config-if)#ip address 192.168.20.2   255.255.255.0
ana_omurga_sag(config-if)#clock rate 64000
ana_omurga_sag(config-if)#exit
ana_omurga_sag(config)#interface serial 1/0
ana_omurga_sag(config-if)#no shutdown
ana_omurga_sag(config-if)#ip address 192.168.40.2   255.255.255.0
ana_omurga_sag(config-if)#clock rate 64000
ana_omurga_sag(config-if)#exit
```

Şekil D.1. Ana omurgadaki sağdaki cihazın konfigürasyonu (Devam)

```
ana_omurga_sag(config)#router ospf  10                                    2
ana_omurga_sag(config-router)#network 192.168.15.0   0.0.0.255 area 0
ana_omurga_sag(config-router)#network 192.168.16.0   0.0.0.255 area 0
ana_omurga_sag(config-router)#network 192.168.17.0   0.0.0.255 area 0
ana_omurga_sag(config-router)#network 192.168.18.0   0.0.0.255 area 0
ana_omurga_sag(config-router)#network 192.168.19.0   0.0.0.255  area 0
ana_omurga_sag(config-router)#network 192.168.20.0   0.0.0.255 area 0
ana_omurga_sag(config-router)#network 192.168.40.0   0.0.0.255 area 0
ana_omurga_sag(config-router)#exit
```

```
                                                                          3
ana_omurga_sag(config)#exit
ana_omurga_sag# copy running-config startup-config
```

Şekil D.1. Ana omurgadaki sağdaki cihazın konfigürasyonu (Devam)

Şekil D.2'de, kurumsal ağımızın ana omurga isimli alanındaki ortadaki Cisco 6509 cihazının konfigürasyon bilgisi verilmiştir.

```
R3>enable                                                                 1
R3#configure terminal
R3(config)#hostname  ana_omurga_orta
ana_omurga_ortaconfig)#interface serial 0/0
ana_omurga_orta(config-if)#no shutdown
ana_omurga_orta(config-if)#ip address 192.168.30.2   255.255.255.0
ana_omurga_orta(config-if)#clock rate 64000
ana_omurga_orta(config-if)#exit
ana_omurga_orta(config)#interface serial 0/1
ana_omurga_orta(config-if)#no shutdown
ana_omurga_orta(config-if)#ip address 192.168.40.1   255.255.255.0
ana_omurga_orta(config-if)#clock rate 64000
ana_omurga_orta(config-if)#exit
```

Şekil D.2. Ana omurgadaki ortadaki cihazın konfigürasyonu

```
ana_omurga_orta(config)#router ospf 11                                    2
ana_ourga_orta(config-router)#network 192.168.30.0   0.0.0.255 area0
ana_ourga_orta(config-router)#network 192.168.40.0   0.0.0.255 area0
ana_omurga_orta(config-router)#exit
ana_omurga_orta(config)#exit                                              3
ana_omurga_orta# copy running-config startup-config
```

Şekil D.2 Ana omurgadaki ortadaki cihazın konfigürasyonu (Devam)

Şekil D.3'de ise, Şekil 4.1'deki Bölüm1 alanının çıkışında bulunan "muhendislik" isimli Cisco c3700 cihazının konfigürasyonu verilmiştir.

```
R4>enable                                                                 1
R4#configure terminal
R4(config)#hostname  muhendislik
muhendislik (config)#interface serial 0/0
muhendislik (config-if)#no shutdown
muhendislik (config-if)#ip address 192.168.2.2   255.255.255.0
muhendislik (config-if)#clock rate 64000
muhendislik (config-if)#exit
muhendislik (config)#interface serial 0/1
muhendislik (config-if)#no shutdown
muhendislik (config-if)#ip address 192.168.3.2   255.255.255.0
muhendislik (config-if)#clock rate 64000
muhendislik (config-if)#exit
muhendislik (config)#interface serial 0/2
muhendislik (config-if)#no shutdown
muhendislik (config-if)#ip address 192.168.4.2   255.255.255.0
```

Şekil D.3. Bölüm1'deki mühendislik isimli cihazın konfigürasyonu

```
muhendislik (config-if)#clock rate 64000
muhendislik (config-if)#exit
muhendislik (config)#interface serial 0/3
muhendislik (config-if)#no shutdown
muhendislik (config-if)#ip address 192.168.5.2   255.255.255.0
muhendislik (config-if)#clock rate 64000
muhendislik (config-if)#exit
muhendislik (config)#interface serial 0/4
muhendislik (config-if)#no shutdown
```

Şekil D.3. Bölüm1'deki mühendislik isimli cihazın konfigürasyonu (Devam)

```
muhendislik(config)#router ospf 12                                    2
muhendislik (config-router)#network 192.168.2.0   0.0.0.255 area 100
muhendislik (config-router)#network 192.168.3.0   0.0.0.255 area 100
muhendislik (config-router)#network 192.168.4.0   0.0.0.255 area 100
muhendislik (config-router)#network 192.168.5.0   0.0.0.255 area 100
muhendislik (config-router)#network 192.168.6.0   0.0.0.255 area 0
muhendislik (config-router)#exit
muhendislik(config)#exit                                              3
muhendislik# copy running-config startup-config
```

Şekil D.3. Bölüm1'deki mühendislik isimli cihazın konfigürasyonu (Devam)

Şekil D.4'de ise, Şekil 4.1'deki Bölüm2 alanının çıkışında bulunan "rektörlük" isimli Cisco c3700 cihazının konfigürasyonu verilmiştir.

```
R5>enable
R5#configure terminal
R5(config)#hostname  rektorluk
rektorluk (config)#interface serial 0/0
rektorluk (config-if)#no shutdown
rektorluk (config-if)#ip address 192.168.7.2   255.255.255.0
rektorluk (config-if)#clock rate 64000
rektorluk (config-if)#exit
rektorluk (config)#interface serial 0/1
rektorluk (config-if)#no shutdown
rektorluk (config-if)#ip address 192.168.11.1   255.255.255.0
rektorluk (config-if)#clock rate 64000
rektorluk (config-if)#exit
rektorluk (config)#interface serial 0/2
rektorluk (config-if)#no shutdown
rektorluk (config-if)#ip address 192.168.12.1   255.255.255.0
rektorluk (config-if)#clock rate 64000
rektorluk (config-if)#exit
rektorluk (config)#interface serial 0/3
rektorluk (config-if)#no shutdown
rektorluk (config-if)#ip address 192.168.13.1   255.255.255.0
rektorluk (config)#router ospf 13
rektorluk (config-router)#network 192.168.11.0   0.0.0.255 area 200
rektorluk (config-router)#network 192.168.12.0   0.0.0.255 area 200
rektorluk (config-router)#network 192.168.13.0   0.0.0.255 area 200
rektorluk (config-router)#network 192.168.7.0   0.0.0.255 area 0
rektorluk (config-router)#exit
```

Şekil D.4 Bölüm 2'deki rektörlük isimli cihazın konfigürasyonu

ÖZGEÇMİŞ

Musa BALTA, 04.02.1986'da Sakarya'da doğdu. İlk, orta ve lise eğitimini Adapazarı'nda tamamladı. 2004 yılında Sakarya Anadolu Lisesi'nden mezun oldu. 2004 yılında başladığı Sakarya Üniversitesi Bilgisayar Mühendisliği bölümünü 2009 yılında bitirdi. 2007-2008 yılları arasında Erasmus öğrencisi olarak Porto Üniversitesi'nde eğitim aldı. 2009 yılında Sakarya Üniversitesi Bilgisayar ve Bilişim Mühendisliği bölümünde yüksek lisansa başladı. 2009-2010 tarihleri arasında T.C. İstanbul Arel Üniversitesi'nde öğretim görevlisi olarak çalıştı. Aralık 2010 tarihinden itibaren Sakarya Üniversitesi Bilgisayar ve Bilişim Mühendisliği'nde araştırma görevlisi olarak çalışmaktadır.