

**T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**TÜRKİYE ÖRNEĞİNDE İNTERNET AĞ  
ALTYAPISININ TIKANIKLIK ANALİZİ**

**YÜKSEK LİSANS TEZİ**

**Hakan Can ALTUNAY**

**Enstitü Anabilim Dalı : ELEKTRONİK VE BİLGİSAYAR  
EĞİTİMİ**  
**Tez Danışmanı : Doç. Dr. Ahmet Turan ÖZCERİT**

**Mayıs 2014**

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**TÜRKİYE ÖRNEĞİNDE İNTERNET AĞ  
ALTYAPISININ TIKANIKLIK ANALİZİ**

**YÜKSEK LİSANS TEZİ**

**Hakan Can ALTUNAY**

Enstitü Anabilim Dalı : **ELEKTRONİK VE BİLGİSAYAR  
EĞİTİMİ**

Bu tez 14 / 05 / 2014 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.

Doç. Dr.  
**Ahmet Turan ÖZCERİT**

Jüri Başkanı



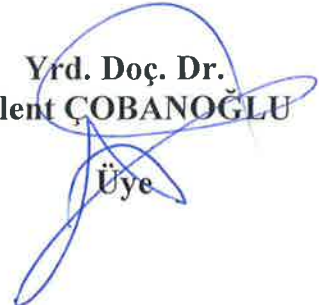
Doç. Dr.  
**Ahmet ZENGİN**

Üye



Yrd. Doç. Dr.  
**Bülent COBANOĞLU**

Üye



## **TEŐEKKÜR**

Tez alıŐmam sűresince deęerli fikir ve gűrűŐlerini benimle paylaŐan danıŐman hocam Do. Dr. Ahmet Turan ŐZCERİT'e, yűksek lisans eęitimi boyunca birlikte olduęum Selim BAKIRCILAR ve Ferdi DOęAN'a, eęitim hayatım boyunca alıŐmalarımı sabırla destekleyen aileme, yűksek lisans alıŐmam sűresince yardımlarını esirgemeyen sevgili eŐime en iten teŐekkűrlerimi sunarım.

# İÇİNDEKİLER

TEŞEKKÜR.....	ii
İÇİNDEKİLER .....	iii
SİMGELER VE KISALTMALAR LİSTESİ.....	v
ŞEKİLLER LİSTESİ .....	vii
ÖZET.....	ix
SUMMARY .....	x
BÖLÜM 1. GİRİŞ .....	1
BÖLÜM 2. BİLGİSAYAR AĞLARININ SİMÜLASYONU .....	5
2.1. OPNET Yazılımı .....	13
2.2. OPNET’te Bulunan Editörler.....	15
2.2.1. Proje editörü .....	15
2.2.2. Node editörü .....	15
2.2.3. Proses editörü .....	16
2.2.4. Link model editörü .....	17
2.2.5. Paket format editörü .....	17
2.2.6. İci editörü (Interface control information).....	18
2.2.7. Anten şablonu editörü.....	18
2.2.8. Modülasyon eğrisi editörü .....	18
2.2.9. PDF editörü.....	18
2.2.10. Probe editörü.....	19
2.2.11. Simülasyon aracı.....	19
2.2.12. Filtre editörü .....	19
2.2.13. Analiz aracı.....	19

BÖLÜM 3. TCP TIKANIKLIK ALGORİTMALARI.....	20
3.1. TCP Reno.....	21
3.2. TCP Tahoe.....	21
BÖLÜM 4. TÜRKİYE ÖRNEĞİNDE İNTERNET AĞ ALTYAPISININ TIKANIKLIK ANALİZİ .....	23
BÖLÜM 5. SONUÇLAR VE ÖNERİLER.....	32
KAYNAKLAR.....	33
EKLER .....	36
ÖZGEÇMİŞ .....	41

## **SİMGELER VE KISALTMALAR LİSTESİ**

ARP	: Address Resolution Protocol
ATM	: Asynchronous Transfer Mode
BITNET	: Because it's Time Network
DNS	: Domain Name System
FTP	: File Transfer Protocol
GPRS	: General Packet Radio Service
GSM	: Global Special Mobile
HTTP	: Hyper Text Transfer Protocol
ICMP	: Internet Control Message Protocol
IEEE	: Institute of Electrical and Electronical Engineers
IP	: Internet Protocol
IPv4	: Internet Protocol Version4
IPv6	: Internet Protocol Version6
IrDa	: Infrared Data Association
Kbps	: Kilo bit per second
LAN	: Local Area Network
LLC	: Logical Link Control
MAC	: Media Access Control
MAN	: Metropolitan Area Network
Mbps	: Mega bit per second
NCP	: Network Control Protocol
NNTP	: Network News Transfer Protocol
NSFNET	: National Science Foundation Network
RARP	: Reverse Address Resolution Protocol
SMTP	: Simple Mail Transfer Protocol
SNMP	: Simple Network Management Protocol

TCP	: Transmission Control Protocol
TFTP	: Trivial file Transfer Protocol
UDP	: User Datagram Protocol
WAN	: Wide Area Network
WCDMA	: Wide Band Code Division Multiple Access
Wi-Fi	: Wireless Fidelity
WiMAX	: World Wide Interoperability for Microwave Access
WLAN	: Wireless Local Area Network
WMAN	: Wireless Metropolitan Area Network
WPAN	: Wireless Personal Area Network
WSN	: Wireless Sensor Network
WWAN	: Wireless Wide Area Network
WWW	: World Wide Web

## ŞEKİLLER LİSTESİ

Şekil 2.1. TCP/IP Katmanları.....	6
Şekil 2.2. Proje Editörü .....	15
Şekil 2.3. Node Editörü .....	16
Şekil 2.4. Proses Editörü .....	16
Şekil 2.5. Link Model Editörü.....	17
Şekil 2.6. Paket Format Editörü .....	18
Şekil 4.1. NoDrop senaryoya ait grafik.....	26
Şekil 4.2. Download Response Time Grafiği.....	27
Şekil 4.3. Sent Sequence Number Grafiği .....	27
Şekil 4.4. Tahoe senaryosuna ait grafik .....	28
Şekil 4.5. Tahoe senaryosuna ait gecikme grafiği.....	29
Şekil 4.6. Reno senaryosuna ait grafik.....	30
Şekil 4.7. Genel ağ görünümü.....	31



## TABLolar LİSTESİ

Tablo 2.1. Kablosuz Ağ Yapılarının Sınıflandırılması .....	11
Tablo 2.2. En yaygın kullanılan modelleme yazılımları .....	14

## ÖZET

Anahtar kelimeler: İnternet Altyapısı, OPNET, Modelleme, Bilgisayar Ağları

Bilgisayarlar iletişim aracı olarak kullanılmaya başlandıktan sonra bilgisayar ağı kavramı ortaya çıkmıştır. Genel olarak bilgisayar ağları kapsama alanlarına göre yerel alan ağları (LAN-Local Area Network), kentsel alan ağları (MAN-Metropolitan Area Network) ve geniş alan ağları (WAN-Wide Area Network) olmak üzere üç gruba ayrılır. Bilgisayarların ağ yapısı üzerinde birbirleriyle haberleşmesini sağlayan kavrama İnternet denir.

İnternet kavramı 1970’li yıllarda ortaya çıksa da 1990’lı yılların sonlarında yaygın bir biçimde kullanılmaya başlanmıştır. Günümüzde ise İnternet teknolojisi yüksek veri hızı ve geliştirilmiş altyapısı ile hizmet sunmaktadır.

İnternet kavramının genişlemesi veri aktarımı sağlanan bilgisayar ağının da genişlemesi demektir. Genişleyen bilgisayar ağlarının yüksek maliyeti, farklı coğrafi bölgelerdeki kurulum zorlukları ve ağ yönetiminin güçlüğü bilgisayar ağlarının modellenmesi ihtiyacını doğurmuştur.

Bu tez çalışmasında Türkiye örneğinde İnternet ağ altyapısının tıkanıklık analizi incelenmiştir. OPNET programının eğitim amaçlı kullanılan OPNET Modeler 14.5 Educational Version ile Türkiye örneğinde İnternet ağ altyapısının tıkanıklık analizi modellenmesi yapılmıştır. Modelleme sırasında ağ üzerindeki mevcut durum değerlendirilmiş ve ağ yapısının geliştirilmesi için ileriye dönük öneriler sunulmuştur.

Modelleme alanı üzerine uygulama ve profil nesnelere eklenerek modelleme çalışmasına NoDrop, Tahoe ve Reno olmak üzere üç farklı senaryo eklenmiştir. Bu senaryolar üzerinde farklı paket kayıpları gözlenmiş ve ortaya çıkan grafikler karşılaştırılmıştır.

# **CONGESTION ANALYSIS OF INTERNET NETWORK INFRASTRUCTURE IN THE CASE OF TURKEY**

## **SUMMARY**

Key Words: Internet infrastructure, OPNET, Modelling, Computer Networks

After computers were started to be used as communication means, computer network concept has come up. Computer networks are divided into 3 groups in general according to their coverage areas as local area networks (LAN-Local Area Network), metropolitan area networks (MAN-Metropolitan Area Network) and wide area networks (WAN-Wide Area Network). The concept which enables the communication between the computers through network structure is called Internet.

Although Internet concept arose in 1970s, it started to be commonly used in the late 1990s. Today internet technology provides service with its high data speed and improved infrastructure. Extension of the Internet concept means the extension of the computer network on which data migration is performed. High cost, installation difficulties in various geographical areas and difficulty in network management of extended computer networks has created the need for modelling of the computer networks.

In this study, congestion analysis of Internet network infrastructure in the case of Turkey has been examined. Internet network modelling has been performed with the Modeler 14.5 Educational version of OPNET. Program which is used for training purpose. During the modelling, existing situation on networks has been assessed and prospective recommendations have been provided in order to improve the network structure. Inserting application and profile objects on modelling area, three different scenarios have been attached on modeling study as NoDrop, Tahoe and Reno. Different package losses have been observed on these scenarios and the graphics showed up have been compared.

## **BÖLÜM 1. GİRİŞ**

Günümüzde bilgi ve bilgiye erişimin önemi hızla artmaktadır. Teknolojinin ilerlemesi ile birlikte bilgiye ulaşmak için en büyük araçlardan birisi olan bilgisayarlar, hayatımızın pek çok alanına girmiş bulunmaktadır. Fakat bunların içerisinde en önemli olanı iletişimidir.

Ülkemizde İnternetin önemi çok kısa sürede fark edilip, hızlı bir ivme ile kullanım alanı ve kullanıcı sayısı artmıştır. 2012 Kasım ayı verilerine göre, ülkemizde İnternet kullanıcı sayısı 36 milyonu geçmiştir. Ancak İnternet abone sayısı, toplam kullanıcı sayısına göre farklılık göstermektedir. Bu durumu İnternet kafeler veya bir abonelik üzerinden birden fazla kullanıcının İnternete girmesi etkilemektedir. Yine 2012 Kasım verilerine göre, ülkemizde toplam abone sayısı 19 milyonu geçmiştir [1].

2013 Ağustos ayı verilerine göre ise ülkemizde ortalama İnternet hızı 3.1Mbps seviyesindedir. Bu hız oranı ile Türkiye dünya sıralamasında 61. sıradadır [2].

Bilgisayarlar, iletişim aracı olarak kullanılmaya başlandıktan sonra bilgisayar ağı kavramı ortaya çıkmıştır. Temel olarak iki veya daha fazla bilgisayarın birbirleriyle iletişim halinde olup veri aktarımı yapabilmesi için tasarlanan yapılara bilgisayar ağı denir. Genel olarak bilgisayar ağları kapsama alanlarına göre, yerel alan ağları (LAN - Local Area Network), kentsel alan ağları (MAN - Metropolitan Area Network) ve geniş alan ağları (WAN - Wide Area Network) olmak üzere üç gruba ayrılır [3].

Bilgisayarların ağ yapısı üzerinde birbirleriyle haberleşmesini sağlayan kavrama İnternet denir [4]. İnternet kontrol edilemeyen bir hızla genişleyip büyümektedir. İnternet kavramının genişlemesi, veri aktarımı sağlanan bilgisayar ağının da genişlemesi demektir. Genişleyen bilgisayar ağlarının yüksek maliyeti, farklı coğrafi

bölgelerdeki kurulum zorlukları ve ağ yönetiminin güçlüğü bilgisayar ağlarının modellenmesi ihtiyacını doğurmuştur [5].

Gerçek bir sistemin bilgisayar ortamında özel yazılımlar kullanılıp, temsili olarak gösterilmesine modelleme denir [6]. Modelleme yazılımları, genel olarak yeni iletişim yazılımlarının geliştirilmesi, var olan protokollerin güncellenmesi, değişik trafik tiplerinin ve ağ topolojilerinin performanslarını ölçmek için kullanılırlar [7].

Bilgisayar ağlarının modellenmesinde OPNET, OMNET, NetworkII.5, Network Simülator 2 ve Network Simülator 3 gibi pek çok modelleme yazılımı kullanılabilir. OPNET yazılımı uygulama performanslarının yönetilmesi ve ağ kapasitesinin genişliği ele alındığında sıklıkla tercih edilen bir modelleme yazılımıdır.

Bilgisayar ağları geniş coğrafyalara yayıldıkça ağ yapıları da incelenmeye başlanmıştır. Ülkemizde Hatice Develi tarafından, Süleyman Demirel Üniversitesi Kampüs Ağı OPNET yazılımı kullanılarak modellenmiş ve bu ağdaki mevcut trafik incelenmiştir [7].

2009 yılında Hakan Çetin tarafından, Türkiye'nin otonom sistem seviyesinde İnternet haritası çıkarılmış ve ağ trafiği yorumlanmıştır. Hakan Çetin tarafından, yapılan bu çalışmada ülkemizdeki otonom sistem noktaları tespit edilerek, birbirleriyle olan bağlantıları ortaya konulmuştur [8].

2003 yılında Tommy Svensson ve Alex Popescu tarafından yapılan tez çalışmasında, farklı şehir arasında dört ayrı algoritma ile oluşturulan TCP tıkanıklık kontrolü işleyişi ele alınmıştır. Bu dört farklı algoritma tıkanıklık kaçınma, yavaş başlangıç, yeniden hızlı ve hızlı kurtarma algoritmalarıdır. Bu algoritmalar üzerinde üç farklı senaryo kurularak ağ yapısı analiz edilmiş ve ağ trafiği hakkında yorumlar yapılmıştır [9].

Bilgisayar ağlarında fazla yükten kaynaklanan paketlerin bekleme sürelerinin artması, paket kayıplarının yaşanması ve ağın etkinliğinin azalması gibi sorunlar o

ağın tıkanıklık yaşandığının göstergesidir. Ağ etkinliği azaldıkça kaybolan paketler tekrar tekrar gönderilecek bu da ağ trafiğinin artması anlamına gelecektir.

Bilgisayar ağlarında tıkanıklığı gidermek için temel olarak rezervasyon tabanlı ve isteğe bağlı olmak üzere iki çeşit tıkanıklık önleme metodu vardır. Rezervasyon tabanlı tıkanıklık önleme metodunda önceden kullanıcı için ağ üzerinde kaynak ayrılır, kullanıcı gerektiğinde bu kaynak üzerinden bilgi transferini gerçekleştirebilir. İsteğe bağlı tıkanıklık önleme yönteminde ise kullanıcılar sürekli ağı gözlemleyerek ağın durumunu değiştirmelidirler.

TCP tıkanıklık denetim algoritmaları Slow Start, Congestion Avoidance, Fast Retransmit ve Fast Recovery olmak üzere dört grupta toplanır.

Türkiye örneğinde İnternet ağ altyapısının tıkanıklık analizi bu çalışmada ele alınmıştır. Gelişen teknoloji ile birlikte ağ yapılarında kullanıcı sayısı ile ağ cihazlarının artması tıkanıklık sorununu arttıracak düşünölmüş ve TCP tıkanıklık denetim algoritmalarının bu sorunu çözmek için nasıl cevap verdiği üzerinde durulmuştur. Farklı senaryolarda elde edilen paket kaybı, gecikme zamanı ve verim gibi grafikler yorumlanarak geleceğe dönük tahminler yapılmıştır.

Bu tez çalışması 5 bölümden oluşmaktadır. Birinci bölüm olan giriş kısmında tezin ortaya çıkmasına neden olan faktörler üzerinde durularak tez düzeni ortaya konulmuştur.

Tez çalışmasının ikinci bölümünde, İnternet, İnternetin dünyada ve ölkemizdeki gelişimi hakkında bilgi verilmiş, bilgisayar ağları ve sınıflandırılması gösterilmiş ve günümüzde sıklıkla kullanılan ağ topolojileri hakkında bilgi verilmiştir.

Ayrıca ikinci bölümde, bilgisayar ağlarının modellenmesinde kullanılan yazılımlar hakkında bilgi verilmiş ve OPNET yazılımının tercih edilmesinin nedenleri üzerinde durularak OPNET yazılımı açıklanmıştır.

TCP tıkanıklık önleme algoritmaları üçüncü bölümde açıklanarak bu algoritmalar üzerinde kullanılan yöntemler ve bu algoritmaların birbirlerine karşı olan üstünlükleri gösterilmiştir.

Tez çalışmasının dördüncü bölümünde Türkiye örneğinde İnternet ağ altyapısının tıkanıklık analizi için hazırlanan modellemede izlenen uygulama adımları açıklanmıştır. Tez çalışması NoDrop, Tahoe ve Reno adı verilen üç farklı senaryo ile hazırlanarak Slow Start (yavaş başlangıç), Fast Retransmit(yeniden hızlı), Fast Recovery(hızlı kurtarma) ve Congestion Avoidance (tıkanıklık kaçınma) algoritmaları incelenmiştir. Ayrıca ağın verimi üzerinde durularak, paket kaybı ve gecikme zamanlarının grafikleri çıkartılmıştır. Uygulama adımlarında her bir ağ elemanının özellikleri hakkında bilgi verilmiştir. Son olarak ise modellemenin sonucunda simülasyon çalıştırılarak grafikler analiz edilmiştir.

Beşinci bölümde ise elde edilen simülasyon sonuçlarına göre ağda tıkanıklık durumunda yapılabilecek adımlar ve geleceğe dair öneriler yer almaktadır.

## **BÖLÜM 2. BİLGİSAYAR AĞLARININ SİMÜLASYONU**

Günümüzde, bilgisayarlar gelişen teknolojiye bağlı olarak hayatımızda çok önemli bir yer tutmaktadır. Bununla birlikte bilgisayarların birbirleriyle haberleşmesi, bilgisayar ağlarının ortaya çıkmasına neden olmuştur. İki ya da daha fazla bilgisayarın yazılımsal ve donanımsal olarak birbirine bağlanmasına bilgisayar ağı denir [10]. Binlerce bilgisayar ağının bir araya gelmesiyle oluşan yapıya ise internet adı verilir [10].

Bilgisayar haberleşmesi, bilgisayar ağları ve İnternet teknolojileri kavramları sıkça birbirine karıştırılmaktadır. Fakat teknik açıdan incelendiğinde iki ya da daha fazla sayıda bilgisayarın iletişimine bilgisayar haberleşmesi denir [10]. Bilgisayar ağı tanımı incelendiğinde ise, bilgisayar ağının bilgisayar haberleşmesini kapsadığı ortaya çıkmaktadır. İnternet teknolojisi ise TCP / IP protokol kümesine bağlı olan bir bilgisayar ağı uygulamasıdır [10].

Bilgisayar ağları, günümüzde hemen hemen her alanda oldukça yaygın bir şekilde kullanılmaktadır. Gerek kamu hizmeti gerekse eğlence ve sohbet ortamları olsun pek çok sektörde ağ uygulamaları ve hizmetleri gözükmektedir [11].

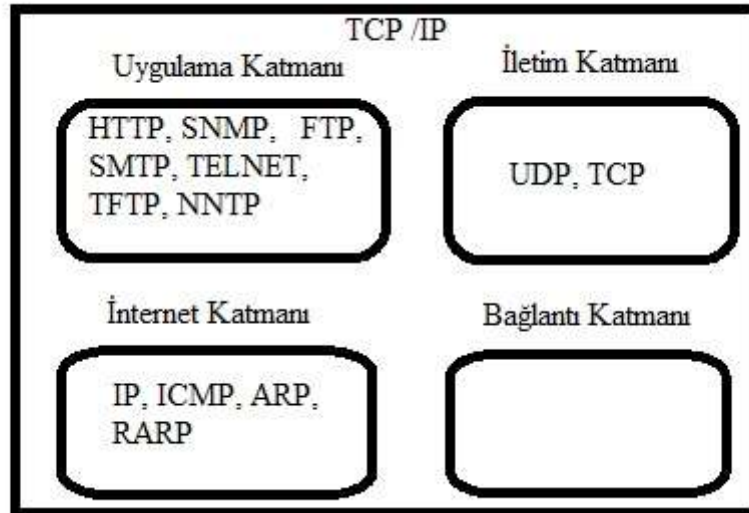
Dünya teknolojisi, son yüzyılda pek çok alanda büyük seviyelerde ilerleme göstermiştir. Bu ilerlemenin en önemli kollarının başında ise İnternet teknolojisi gelmektedir. İnternet teknolojisi ile birlikte gelen hızlı iletişim, günlük hayatta birçok faydayı da beraberinde getirmiştir [12].

İnternet ilk olarak 1962 yılında J.C.R. Licklider'in tartışmaya açtığı Galaktik Ağ kavramı ile doğmuştur. Licklider, bu kavram ile isteyen herkesin nerede olursa olsun herhangi bir bilgi ya da programa erişebilmesini ifade etmiştir. 1965 yılında ise Lawrence Roberts ile Thomas Merrill ilk defa bilgisayarların birbirleri ile



haberleşmesini sağlamışlardır. 1966 yılında Roberts, DARPA'da çalışmaya başlamış ve ARPANET adında bir proje yapmıştır. 1969 yılında ARPANET projesi çerçevesinde ilk ağ bağlantısı dört farklı merkez ile ana bilgisayarlar arasında sağlanmış ve internetin ilk uygulaması ortaya çıkmıştır. Bu uygulamadan sonra kısa süre içerisinde birçok merkezdeki bilgisayarlar ARPANET ağına bağlanmıştır. 1971 yılında Ağ Kontrol Protokolü (NCP – Network Control Protocol) ortaya çıkmış, 1972 yılına gelindiğinde ise ARPANET'in NCP ile bir uygulaması gerçekleştirilmiştir. Yine aynı yıl içerisinde ARPANET üzerinde ilk defa elektronik posta kullanılmaya başlanmıştır. 1 Ocak 1983 tarihinde ise kullanıcılara yeni olanaklar sağlayan İletişim kontrol Protokolü (Transmission Control Protocol / Internet Protocol – TCP / IP) ARPANET üzerinde kullanılmaya başlanmıştır.

TCP / IP modeli uygulama katmanı, iletim katmanı, internet katmanı ve bağlantı katmanı olmak üzere dört farklı katmandan oluşur. Bu katmanlar üzerinde farklı protokoller bulunmaktadır. Uygulama katmanı üzerinde HTTP, SNMP, FTP, SMTP, TELNET, TFTP, NNTP protokolleri, iletim katmanı üzerinde UDP ve TCP protokolleri, internet katmanı üzerinde IP, ICMP, ARP, RARP protokolleri bulunur. Bağlantı katmanı üzerinde ise herhangi bir protokol yoktur [13].



Şekil 2.1. TCP/IP Katmanları

TCP / IP bugün var olan İnternet ağının ana halkası olarak geçerliliğini sürdürmektedir. Günümüz İnterneti genel olarak IPv4 ve IPv6 sürümleri üzerine kurulmuştur [14].

Dünya üzerindeki İnternet ağına bağlı olan bilgisayar sayısının artması ve teknolojiye hız kavramının öne çıkması ile birlikte İnternet protokolünün dördüncü sürümü olan IP version 4 (IPv4) yetersiz kalmaya başlamıştır. Artan hız gereksinimi ile birlikte hizmet kalitesi, mobilite, güvenlik gibi özellikler ön plana çıkmış ve yeni bir İnternet protokolü geliştirilmesine ihtiyaç duyulmuştur. IPv4'te karşılaşılan sorunların giderilmesi amacıyla IPv6 veya diğer bir isimle Yeni Nesil İnternet Protokolü geliştirilmiştir [15].

IPv6, IPv4 ile uyumlu çalışabilen fakat zamanla onun yerini alacak bir IP sürümüdür. Bu sürüm IETF tarafından IPv4'te karşılaşılan adres uzayının tükenmesi, başlık yapısı, yönlendirici ve güvenlik gibi konularda karşılaşılan problemlerin çözülmesi amacıyla tasarlanmıştır. Tabi ki bu problemlerin başında adres uzayının tükenmesi gelmektedir [15].

Bu nedenle IPv4'teki adresleme formatı veri alanlarında da çok önemli değişiklikler IPv6'ta yapılmıştır. IETF, bu yeni nesil İnternet protokolünü tasarlariken bazı hedefler belirlemiştir. Bunlar;

- IPv4 sürümünden çok daha fazla sunucuyu adresleyebilmek,
- Protokolün basitleştirilmesi ve yönlendirme tablolarının boyutlarının küçültülmesi,
- Güvenliği daha yüksek seviyeye çekebilmek,
- Gerçek zamanlı uygulamalardaki hizmet kalitesine daha fazla önem verilmesi,
- Mobil kullanım ve dolaşımın problemsiz gerçekleşebilmesi,
- Protokolün gelişim ve güncellemelere açık olması,
- Eski ve yeni protokollerin bir arada kullanılabilmesi olanağı olarak belirlenmiştir.

IPv4 ile IPv6 sürümlerinin karşılaştırılmasını ise maddeler halinde aşağıdaki gibi gösterebiliriz.

- IPv6, 128 bit adresleme avantajına sahiptir. Yani değişik bir tabirle IPv4 ile gelen  $2^{32}$  potansiyel adres kapasitesi IPv6 ile  $2^{128}$  seviyesine yükseltilmiştir.
- IPv6 ile NAT'a olan gereksinim bitmiş ve bu sayede uçtan uca adresleme olanağı doğmuştur. Bu sebeple de maliyet ve karmaşıklık azalmıştır.
- IPv4 'te güvenlik protokolü olan IP-Sec seçime bırakılmıştır. Fakat bu özellik IPv6 ile zorunlu hale getirilmiştir.
- IPv6, akış etiketi ve uzantı başlıkları sayesinde daha gelişmiş seviyede hizmet kalitesi sunmaktadır.
- IPv6 ile otomatik yapılandırma olanağı doğmuştur. Bu sayede yerel adresleme, çoklu gönderim, herhangi birine gönderim, geliştirilmiş tak ve çalıştır özelliği, komşu düğümlerin, yönlendiricilerin ve sunucuların bulunması özellikleri sağlanır.
- IPv6 ile yeni ve farklı adresleme çeşitleri ortaya çıkmıştır. Bunlar yerel ve herhangi birine gönderim seçenekleridir.
- Ağ ara yüzlerine birden fazla adres atanabilmesi özelliği sayesinde bu adresler gereksinimlere göre güvenlik, yük dengeleme, güvenilirlik ve hizmet kalitesi amacıyla kullanılabilir.
- IPv4 sürümünde başlık yapısında bulunan zorunlu olmayan alanlar IPv6 ile çıkartılmış ve sade bir başlık yapısı kazandırılmıştır.
- IPv6 başlığına ek bilgi eklenebilmektedir. Örneğin IPsec ve mobil IPv6 protokolleri ile birlikte uzantı başlıkları, temel IPv6 protokolünün üzerine eklenmektedir.
- IPv6 mobilite, güvenlik, hizmet kalitesi, P2P uygulamaları gibi güçlü ağ hizmetleri sunmaktadır [15].

Türkiye'ye İnternetin gelişine baktığımızda ise, 1986 yılında tesis edilen EARN (European Academic and Research) / BITNET (Because It's Time Network) bağlantılı bir geniş alan ağı olan TÜVEKA (Türkiye Üniversiteler ve Araştırma Kurumları Ağı) görülmektedir [16].

İlerleyen yıllarda TÜVEKA ağı hat kapasitesinin yetersiz kalmaya başlamasıyla ODTÜ ve TÜBİTAK işbirliğinde 1991 yılında yeni bir ağ projesi başlatılmıştır. Bu proje üzerinde ilk bağlantı 1991 yılının Ekim ayında X.25 üzerinden Hollanda'ya yapılmış, bu gelişmeyi takiben 1993 yılında 64Kbps kapasiteli kiralık hat üzerinden ODTÜ Bilgi İşlem Daire Başkanlığındaki yönlendiriciler kullanılarak ABD'de NSFNET'e TCP / IP üzerinden sağlanmıştır. Bu hat uzun bir süre ülkenin tek çıkış hattı olmuştur. Sürecin devamında 1994 yılında Ege Üniversitesi, 1995 yılında Boğaziçi ve Bilkent Üniversiteleri, 1996 yılında İstanbul Teknik Üniversitesi bağlantıları sağlanmıştır. Yine aynı yıl TÜBİTAK bünyesinde Ulusal Akademik Ağ ve Bilgi Merkezi (ULAKBİM) kurulmuştur. ULAKBİM, en yeni teknolojileri kullanarak ülkemizdeki bütün eğitim ve araştırma kurumlarını birbirine bağlayan Ulusal Akademik Ağ (ULAKNET)'i kurmuş hızlı bir iletişim ağı ve bilgi hizmeti sağlamıştır [17].

İnternet üzerindeki cihazların adreslenmesinin sonucunda bilgisayarlar arasında iletişimi sağlayan internet protokolü (IP) adında bir ağ katmanı protokolü ortaya çıkmıştır [17].

İnternet protokolü, İnternet ağının en temel elemanıdır. Çünkü İnternet bağlantısı bulunan iki veya daha fazla bilgisayarın birbiriyle haberleşmesi, İnternet protokolü sayesinde gerçekleşmektedir. İnternet protokolü, TCP ve UDP olarak adlandırılan taşıma katmanı protokollerinin altında, Ethernet ve ATM gibi ağ katmanı protokollerinin üzerinde yer alır. İnternet protokolünün görevi, ağdaki bilgisayarlar arasında veri alışverişi gerçekleşirken, bilgisayarların adreslenmesini ve veri paketlerinin yönlendirilmesini sağlamaktır [18].

Bilgisayar ağları için kaynaklarda farklı sınıflamalar olsa da geleneksel sınıflama yaygın olarak kullanılmaktadır. Geleneksel sınıflamaya göre bir bilgisayar ağı üç sınıfa ayrılır [19].

- Yerel Alan Ağı (Local Area Network - LAN)
- Kentsel / Kampüs Alan Ağı (Metropolitan Area Network - MAN)
- Geniş Alan Ağı (Wide Area Network - WAN)

Ağ üzerinde bulunan bilgisayarların ve ağ ekipmanlarının birbirlerine bağlanma şekline ağ topolojisi denir. Yerel bilgisayar ağlarında genel olarak kullanılan başlıca topolojiler aşağıda gösterilmiştir [20].

- Doğrusal Topoloji (Bus Topology)
- Yıldız Topoloji (Star Topology)
- Halka topoloji (Ring Topology)
- Ağaç Topoloji (Tree Topology)
- Karmaşık Topoloji (Mesh Topology)

Herhangi bir fiziksel bağlantı olmadan kablosuz haberleşme yeteneği bulunan cihazların birbirleriyle haberleşmesini sağlayan yapıya kablosuz ağ denir [20]. Şimdilerde kullanım alanı oldukça yaygınlaşan kablosuz ağlar, ilk olarak 1971 yılında, kablosuz ağların atası olarak kabul edilen ALOHANET ağının kurulması ile başlamıştır. Bu ağ, bilinen ilk kablosuz yerel alan ağı (WLAN-Wireless Local Area Network)'dir. Bu ağda sadece yedi adet bilgisayar olup, çift yönlü yıldız topoloji kullanılmıştır [20].

Kablolu bilgisayar ağları iletim ortamı açısından çift burgulu, koaksiyel ve fiber optik kablo olmak üzere üç grupta incelenir. Bu üç iletim ortamı da geniş band, taşıyıcı band ve temel band iletim metodu ile veri alışverişi sağlarlar.

Kablosuz ağlardaki tarihsel gelişime baktığımızda, 1981 yılında Global Special Mobile(GSM)'in başlaması, 1994 yılında Ericsson tarafından ilk Bluetooth'un tanıtılması, 1998'de WLAN standardı IEEE802.11 Legacy sürümünün yayınlanması, 1999'da yeni kablosuz LAN standartları 802.11b ve 802.11a sürümlerinin yayınlanması, 2001'de kablosuz geniş band standardı olan IEEE802.16 yani WİMAX'ın tanıtılması, 2003'te IEEE802.16a adında geliştirilmiş WİMAX sürümünün yayınlanması, yine aynı yılda yayınlanan IEEE802.11g ve IEEE802.15.4 diğer bir adıyla ZigBee standardının yayınlanması ve 2008 yılında IEEE802.11n standardının yayınlanması görülmektedir [21].

Gelişen teknoloji ile birlikte, kablosuz ağların daha yaygın bir şekilde kullanılmasının nedenleri kablo maliyetinden kurtulma, kablolardan bağımsız serbest dolaşımın sağlanması, genişletilebilme, ağ kurulumun kolaylığı ve istenilen durumlarda kablolu ağlarla bütünleşme olarak sayılabilir[22].

Kablosuz ağların sınıflandırılması aşağıdaki tabloda gösterilmiştir.

Tablo 2.1. Kablosuz Ağ Yapılarının Sınıflandırılması

Kablosuz Kişisel Alan Ağları	Kablosuz Yerel Alan Ağları	Kablosuz Anakent Alan Ağları	Kablosuz Geniş Alan Ağları
Bluetooth HomeRf IrDA ZigBee	HiperLAN	WIMAX	3G GPRS GSM 4G

Bu kablosuz ağ yapılarından en önemlilerinden birisi de WIMAX teknolojisidir. WIMAX, dar alanda kablosuz internet ağı oluşturan WI-FI teknolojisinden sonra kablosuz İnterneti çok daha geniş alanlara yaymaya çalışan bir teknolojidir [23]. WIMAX teknolojisinin ülkemizdeki durumuna bakıldığında test ve deneme çalışmaları olarak Ankara, İstanbul, Kocaeli, Eskişehir ve Yozgat illerinde çeşitli denemeler gerçekleştirilmiştir.

Uzun mesafelerde yüksek hızda kablosuz erişimin sağlanmasını amaçlayan bir diğer teknoloji ise 4G teknolojisidir. 4. Nesil cep telefon teknolojilerine verilen genel ada 4G denir. Bu teknoloji ile kapsama alanı artmakta aynı zamanda mevcut şebekelerde bulunmayan yüksek hızda görüntü transferi sağlanmaktadır. [24]

Teknolojinin ilerlemesi ve İnternetin kontrol edilemeyecek şekilde büyümesi sonucunda bilgisayar ağlarının benzetimi ihtiyacı doğmuştur. Belirli bir zaman çerçevesinde herhangi bir sistemin yapısının taklit edilmesine simülasyon ya da benzetim denir [25]. Benzetim, mevcut işlemler hakkında geçmişten günümüze hatta geleceğe dair gerçekçi bilgiler sunan bir araçtır. Simülatör kullanmak, işlemler üzerindeki testlerin daha ucuz ve daha kısa sürede bitirilmesi olanağını sağlar. Bu

sebeple günümüzde teknolojik gelişmelerin etkisiyle sanayi ve endüstride benzetim kullanımını artmıştır.

Gün geçtikçe genişleyen bilgisayar ağları ve İnternet, simülasyon kullanma zorunluluğunu ve gerekliliğini ortaya çıkarmıştır [26]. Bilgisayar ağlarının gerçek hayatta kurulması maliyetli ve zaman alıcı olması bakımından bu alanda tasarlanmış OPNET, OMNET++, NetworkII.5, Network Simulator 2 ve Network Simulator 3 gibi bazı programlar tasarlanmıştır.

Ağ simülasyonu, bir ağ ortamının bilgisayarda modellenmesi ve fiziksel kurulum yapılmaksızın çalışma düzeninin test edilmesi anlamına gelmektedir. Bu test sürecinde TCP ve OSI referans modelinin tüm özellikleri uygulamaya dahil edilirse alınan sonuçlar o derecede gerçeğe yakın olur. Bu sayede gerçek hayattaki ağların kurulumu herhangi bir sorunla karşılaşmadan başarıyla gerçekleştirilebilir [26].

Ağ modelleme yazılımlarının yararlarından bazıları şunlardır:

- Gerçek hayattaki anlaşılması zor olan ağ yapıları iyi bir öğretim ortamı sağlanarak rahat bir şekilde kavranır.
- Birden çok kullanıcı gerçek zamanlı bir ortamı aynı anda kullanır.
- Kullanıcılar ortam araçlarını kullanarak ağ yapısı hazırlayıp test edebilirler.
- Maliyetler minimum seviyeye indirilir.

Bununla birlikte ağ modelleme yazılımlarının bazı dezavantajları da bulunmaktadır.

- Karmaşık bir sistemin modellenmesi yorucu olur.
- Benzetim modelinin programlanmasında kullanılan programlama dilin zorluğu.
- Benzetim modelinin yapısı gerçek sistemle ilgili ancak tahminde bulunmayı sağlar.
- Benzetim modelleri soruna en iyi çözümü bulmak yerine alternatif çözümleri karşılaştırır.

Ağ modelleme yazılımları, açık kaynak kodlu ve kapalı kaynak kodlu yazılımlar olmak üzere 2 grupta toplanır. Açık kaynak kodlu yazılımlara Ns-2, Ns-3, OMNET++, kapalı kaynak kodlu yazılımlara ise QualNet ve OPNET örnek olarak verilebilir.

Ns2, açık kaynak kodlu bir ayırık olay simülasyon programıdır ve ilk olarak 1989 yılında ortaya çıkmıştır. C++ ve OtCl dilleri ile yazılmıştır. Senaryolar OtCl ile yazılmaktadır. Bu senaryolar alt modüllerde C++ ile yazılan kodları çağırılmaktadır. Ns2 geniş alan ağları, yerel alan ağları ve kişisel alan ağlarında kullanılabilir [27].

Ns3 yazılımı ise kesinlikle Ns2'nin devamı değildir. C++ ve Python dilleri ile yazılmış nesne tabanlı bir yazılımdır. Ns3 başka yazılımlarla tümleştirilebilir, esnek ve geliştirilebilir bir dildir. Zengin bir belgelemeye sahip olan Ns3 ayrıca aktif bir mail grubu da içerir [28].

OMNET++, nesneye yönelik modüler yapıda ayırık olay simülasyon yazılımıdır. OMNET++ yazılımı haberleşme ağlarının trafik modellemesinde, ağ kuyruklarının modellenmesinde, karmaşık yazılım sistemlerinin performans durumlarının ölçülmesinde ve dağıtık donanım sistemlerinin modellenmesinde kullanılır [29].

Qualnet yazılımı 2000-2001 yıllarında ortaya çıkmıştır. Bu yazılım ağ performansını yüksek oranda doğru ölçebilmektedir. Büyük heterojen ağları yönetmek ve dağıtık uygulamalar için kullanılmaktadır [30].

## 2.1. OPNET Yazılımı

OPNET, ağ tasarımı, ağ üzerinde modelleme, trafik izleme, performans analizi gibi birçok işlemi ayrıntılı olarak yapmamızı sağlayan bir yazılımdır. Bu yazılım üzerinde kendi geliştirdiğimiz ağ protokolünü dahi test edebiliriz. Günümüzde pek çok simülasyon programı olmasına rağmen en yaygın kullanılanlar aşağıdaki tabloda gösterilmiştir.



Tablo 2.2. En yaygın kullanılan modelleme yazılımları

AĞ MODELLEME YAZILIMLARI	
OPNET	<ul style="list-style-type: none"> <li>- Geniş bir kütüphaneye sahiptir.</li> <li>- Editörleri sayesinde yeni model ve ürünler oluşturulabilir ve bunlar kütüphaneye eklenebilir.</li> </ul>
NETWORK SIMULATOR 2 (NS-2)	<ul style="list-style-type: none"> <li>- Açık kaynak kodlu, ayrık olay simülasyon yazılımıdır.</li> <li>- Senaryolar OtCl dili ile yazılmaktadır.</li> <li>- Alt modüllerde C++ dili ile yazılan modüller senaryolar tarafından çağırılır.</li> </ul>
NETWORK SIMULATOR 3 (NS-3)	<ul style="list-style-type: none"> <li>- C++ ve Phyton dilleri ile yazılmış nesne tabanlı bir yazılımdır.</li> <li>- Başka yazılımlarla tümleştirilebilir, esnek bir yazılımdır.</li> <li>- Aktif bir mail grubu içerir.</li> </ul>
OMNET++	<ul style="list-style-type: none"> <li>- Nesneye yönelik ayrık olay simülasyon programıdır.</li> <li>- Modüler yapıda tasarlanmış bir yazılımdır.</li> </ul>
QUALNET DEVELOPER	<ul style="list-style-type: none"> <li>- Heterojen ağlarda kullanılır.</li> <li>- Ağ performansını büyük oranda doğru ölçer.</li> </ul>

OPNET yazılımı, haberleşme ağlarının modellenmesi için kullanıcıya görsel olarak bir benzetim ortamı sağlayan nesne tabanlı bir programdır. Sistemler modellendikten sonra davranış ve başarımları analizleri ayrık olay (discrete event) benzetim metodu ile gerçekleştirilir. OPNET geniş bir kütüphaneye ve hiyerarşik bir modelleme yapısına sahiptir. OPNET yazılımının diğer yazılımlara göre en önemli avantajı editörlerin yardımı sayesinde yeni protokol ve ürünlerin modellerinin oluşturulabilmesi ve bunların model kütüphanesine eklenebilmesidir [31].

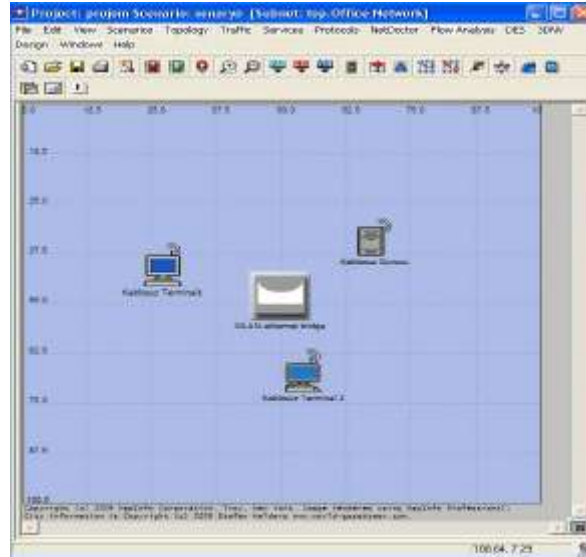
OPNET genel olarak LAN ve WAN yapılarının modellenmesinde, ağlar arası iletişim (internetworking), dağıtık algılayıcı ve kontrol ağlarının geliştirilmesinde, uydu ağlarda ve kablosuz ağ yapılarında kullanılır [31].

## 2.2. OPNET'te Bulunan Editörler

OPNET modelleme ve simülasyon yazılımında 13 adet editör bulunmaktadır.

### 2.2.1. Proje editörü

Proje editörü fiziksel topolojiyi belirlemek, iletişim ağı konumunu tanımlamak, düğümler arası bağlantıların sağlandığı, simülasyonun çalıştırıldığı ve sonuçların izlendiği editördür. Bu editörde her düğümün temelinde bulunan yetenekler belirlenir. Düğümün veya düğümlerin davranış parametreleri veya özellikleri değiştirilerek, sadece o düğüme özgü olması sağlanabilir.

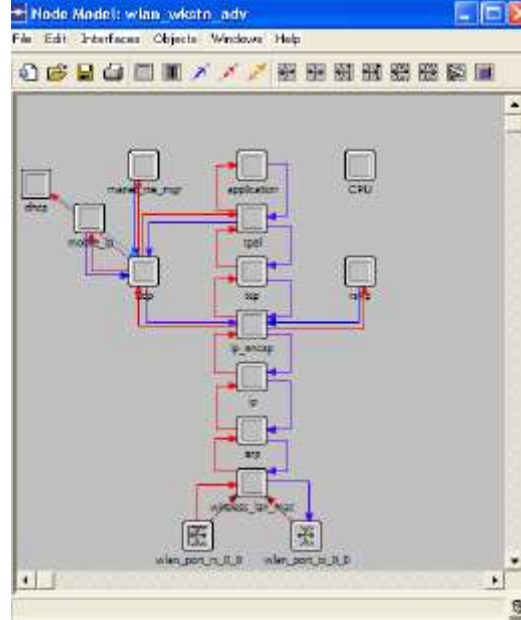


Şekil 2.2. Proje Editörü

### 2.2.2. Node editörü

Node editörde, ağ üzerindeki iletişim cihazlarının dahili özellikleri kullanılarak node modeller oluşturulur. Node modeller birbirlerine bağlantılı olarak tanımlanmıştır.

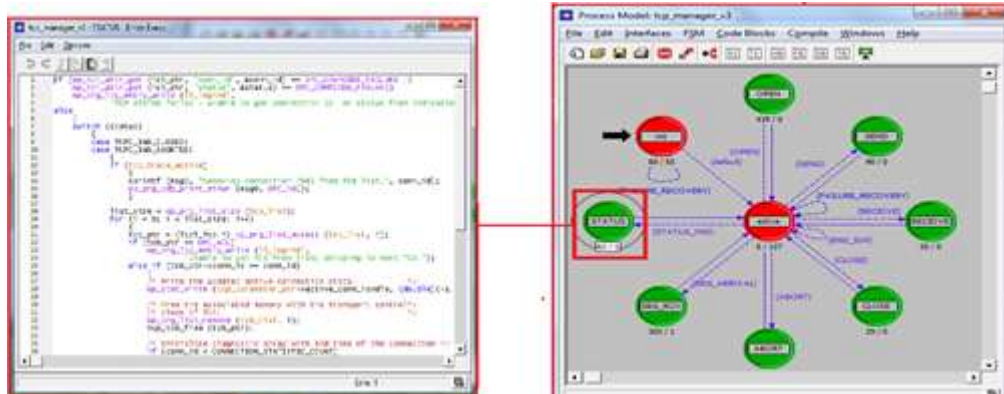
Node modeller, parametre modülleri ve program modülleri olmak üzere iki ayrı grupta incelenirler.



Şekil 2.3. Node Editörü

### 2.2.3. Proses editörü

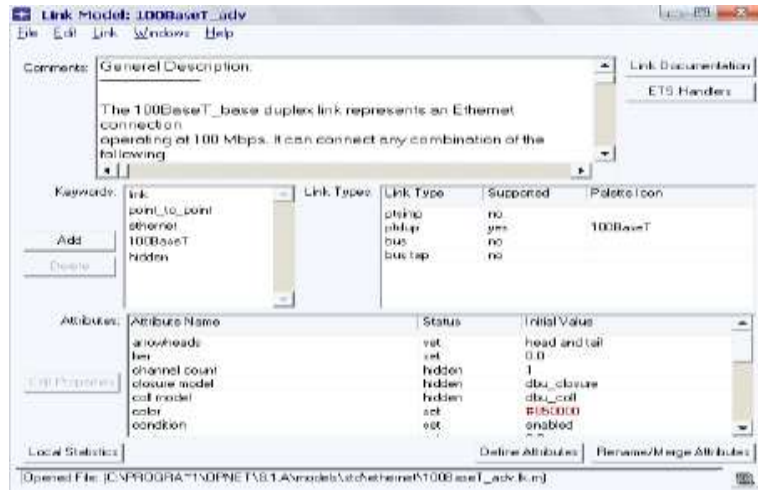
Proses editör, ağın akış mantığını, kuyruk modüllerini açıklamak ve işlemcinin davranışını geliştirmek için kullanılır. Prosesler arasındaki iletişim, kesmeler tarafından desteklenmektedir. Proses modeller C programlama dili ile tanımlanır [31].



Şekil 2.4. Proses Editörü

## 2.2.4. Link model editörü

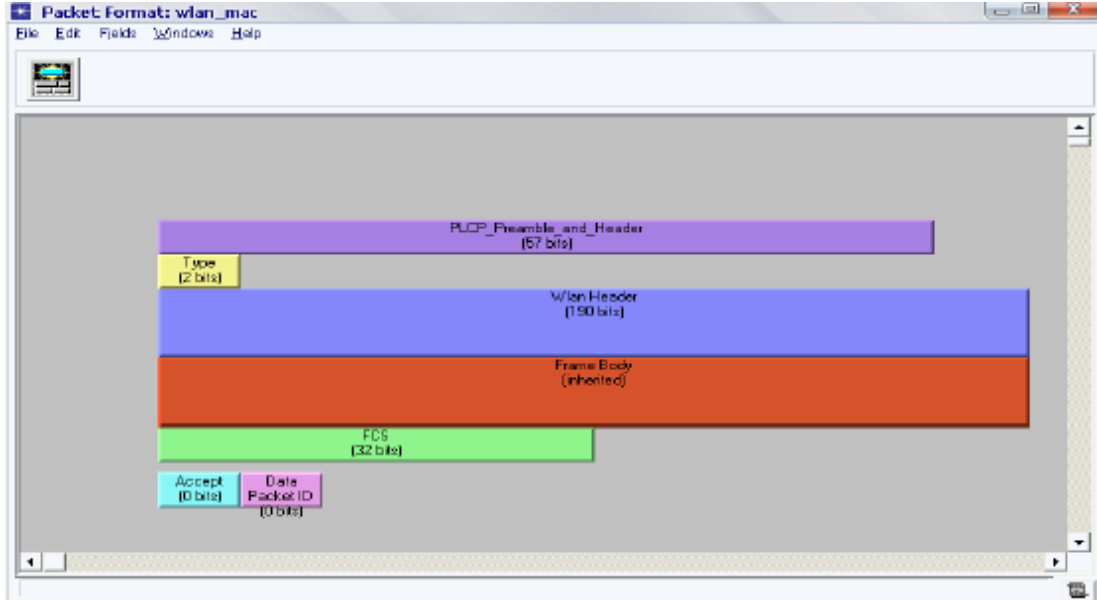
Bağlantı modellerinin oluşturulduğu, düzenlendiği ve görüntülediği editördür. Ayrıca bu editörde ağ cihazlarının iletişimini sağlayan bağlantı modelleri oluşturulup düzenlenir. Veri iletim hızı, kanal sayısı ve hata modeli gibi bağlantı hattı özellikleri bu editörde tanımlanır.



Şekil 2.5. Link Model Editörü

## 2.2.5. Paket format editörü

Paket formatlarının oluşturulduğu, paketlerin sıralarının belirlendiği, paket içerisindeki verilerin tiplerinin ve boyutlarının ayarlandığı editördür.



Şekil 2.6. Paket Format Editörü

### 2.2.6. Ici editörü (Interface control information)

Ara yüz kontrol bilgilerinin düzenlenerek oluşturulduğu ve görüntülediği editördür.

### 2.2.7. Anten şablonu editörü

Alıcı ve vericiler için anten şablonlarının oluşturulup ayarlarının yapıldığı editördür. Kablosuz ağ modelindeki alıcı ve vericiler için anten örnekleri oluşturmak ve düzenlemek için kullanılır.

### 2.2.8. Modülasyon eğrisi editörü

Vericiler için modülasyon eğrilerinin düzenlenerek gösterildiği editöre modülasyon eğrisi editörü denir. Sadece kablosuz modül desteği eklenmiş OPNET modeller yazılımlarında bulunur.

### 2.2.9. PDF editörü

Tahmini yoğunluk fonksiyonlarının oluşturulduğu editördür [31].

### **2.2.10. Probe editörü**

Simülasyon süresince büyük miktarlarda çıkış üretme yeteneğine sahip opnet modellerini içeren editördür.

### **2.2.11. Simülasyon aracı**

Ağ yapısının tüm modelleri tanımlandıktan sonra sistem performansı ve davranışını incelemek amacıyla kullanılan araçtır.

### **2.2.12. Filtre editörü**

Analiz penceresindeki verilere uygulanacak olan sayısal işlemlerin tanımlandığı editördür.

### **2.2.13. Analiz aracı**

Simülasyonlar sonucunda üretilen sayısal değerleri işleme ve görüntüleme işlemlerinin yapılabildiği ayrıca bu değerler üzerine yorumların yapılabildiği araçtır [31].

### **BÖLÜM 3. TCP TIKANIKLIK ALGORİTMALARI**

TCP protokolü bulunan ağlarda, bilgi alışverişinin düzenli olarak sağlanabilmesi için temel olarak kullanılan tıkanıklık önleme algoritmaları bulunur. Bu algoritmalar TCP Reno ve TCP Tahoe algoritmaları olarak adlandırılır. Bu algoritmalar üzerinde slow start (yavaş başlangıç), fast recovery (hızlı devam), fast retransmit (yeniden hızlı iletim) ve congestion avoidance (tıkanıklık kaçınma) yöntemleri kullanılır.

Yavaş başlangıç, TCP protokolü için geliştirilen bir tıkanıklık önleme yöntemidir. Bu yöntemde amaç, ağ üzerinde paket gönderilirken paket kayıplarını asgariye indirmek için hattı test ederek paket gönderim hızını arttırmaktır. Doğrusal ve üssel olmak üzere iki farklı yaklaşımı vardır. Doğrusal yaklaşımda tıkanıklık penceresi içerisindeki bir bilginin onaylanması durumunda pencerenin kapasitesi, onaylanmış paketler kadar veya sabit bir sayı kadar arttırılır. Eğer bu artış miktarı sabitse doğrusal artım, değişkense üssel artım olarak değerlendirilir. Bu nedenle her başarılı transfer sonrasında pencere değişken miktarda büyütülerek anlık aktarım miktarı arttırılmış olur [35].

Hızlı devam yöntemi yavaş başlangıç yönteminin bir alt uygulamasıdır. Bu uygulamada paket onaylarındaki gecikmeden kaynaklanan pencere boyutunu düşürme işlemi, pencerenin boyutunu daha yavaş azaltmakla olmaktadır.

Yeniden hızlı iletim yöntemi özellikle Tahoe algoritmasında zaman aşımı problemi yaşandığı zaman kullanılan bir yöntemdir. Alıcı tarafta sürekli bir paket alımı söz konusu ise gönderici tarafta paketlerin yinelenmesi görünecektir [36].

Tıkanıklık kaçınma yönteminde paketi gönderen tarafın paket kaybı ve noktasal gecikmeler gibi ağdaki tıkanıklık sebeplerini hesaba katarak gönderim hızını azaltması veya arttırması esasına dayanır. TCP ağlar için kullanılan tıkanıklık önleme

yöntemi istatistiksel olarak ağdan yararlanacak azami kullanıcının hesaplanarak bu sayıdan fazlasını engelleyip veya tıkanıklığı önceden anlayıp aktif sıra yönetimi yaparak çalışmaktadır.

### 3.1. TCP Reno

TCP tıkanıklık önleme algoritmalarından birisi olan Reno algoritması her başarılı onay paketinden sonra tıkanıklık penceresini (congestion window) bir arttırır. Şayet ağ üzerinde paket kaybı olursa, tıkanıklık penceresinin değeri başlangıç değerine geri döndürülür. Ağ üzerinde tıkanıklık olduğunda, tıkanıklık penceresinin boyutu dolu ise ve ağ üzerinden onay paketi alınmadıysa, tıkanıklık penceresinin boyutu yarıya indirilir. Aynı zamanda bir paket ağ üzerinde zaman aşımına uğrarsa yine tıkanıklık pencere boyutu yarıya indirilir [35].

Eğer ağ üzerinde gönderilmekte olan her paket sorunsuz bir şekilde alınırsa, tıkanıklık pencere boyutu arttırılacaktır.

Son olarak ise gönderilen paketler göndericinin beklediği zamandan daha yavaş bir hızda gönderildiğinde tıkanıklık penceresinin boyutu azaltılacaktır.

### 3.2. TCP Tahoe

TCP Tahoe algoritması bir tıkanıklık önleme algoritması olup, Reno algoritmasından farklı olarak ağ üzerinde bir tıkanıklık olduğunda bu durum algılanır ve çözüm üretilir. Tahoe algoritmasında, paket kaybı onay paketinin alınması sırasındaki zaman aşımında algılanır. Paket kaybı algılandığında Tahoe algoritmasında tıkanıklık pencere boyutu 1 mss'e (maximum segment size) indirilir. Ardından iletişim sıfırlanarak yavaş başlangıç yöntemi uygulanarak ve bilgi iletişimine devam edilir [36].

Aynı durum, yeni paket kaybının algılanması, Reno algoritmasında üç adet tekrarlı onay paketinin alınması ile olur. Reno algoritmasının bu durumda ürettiği çözüm ise Tahoe algoritmasından farklı olarak tıkanıklık pencere boyutunu yarıya indirmek ve



hızlı kurtarma yöntemini uygulamaktır. Bu nedenle hızlı kurtarma yönteminin sadece Reno algoritmasında kullanıldığı anlaşılmaktadır. Ayrıca zaman aşımı durumunda her iki algoritma da tıkanıklık pencere boyutunu 1mss indirmektedir.

## **BÖLÜM 4. TÜRKİYE ÖRNEĞİNDE İNTERNET AĞ ALTYAPISININ TIKANIKLIK ANALİZİ**

Türkiye örneğinde İnternet ağ altyapısının tıkanıklık analizini incelemek için kullanılan modelleme yazılımı OPNET'tir. OPNET yazılımı iletişim protokollerinde, ağ yapılarının modellenmesinde, simülasyon ve performans işlemlerinin gerçekleştirilmesinde kullanılan bir yazılımdır. Bilgisayar ağlarının modellenmesinde yaygın olarak kullanılan OPNET hem Windows hem de Unix ortamında çalışabilen bir programdır [32].

OPNET yazılımı nesne tabanlı bir yazılımdır ve düzenlenebilir nesnelere bünyesinde barındırır. OPNET yazılımı hiyerarşik bir yapıya sahiptir. OPNET yazılımının grafik editörü bulunmaktadır. Gerektiği yerde grafik editörü ile modeller oluşturulabilir. OPNET yazılımı yüksek seviyeli bir dildir. Bu nedenle detaylı modellemeye imkan sağlayan esnek bir yapısı vardır. Model özellikleri C dilinde derlendikten sonra OPNET'te benzetimler otomatik olarak gerçekleşir. OPNET modelleri ve veri dosyaları program arayüzü ile değiştirilebilir. OPNET yazılımında uygulama sırasında pek çok istatistik, otomatik olarak kullanıcıya sunulur. Simülasyon sonrası sonuçlar analiz edilir. Simülasyon sonuçlarının işlenmesi ve grafiksel olarak kapsamlı sunumu için OPNET'te içerik yönünden zengin bir araç menüsü mevcuttur. Ayrıca OPNET yazılımında, simülasyonun çeşitli seviyelerinde animasyonlar oluşturulabilir.

OPNET yazılımı, yukarıda bahsedilen özelliklerinden dolayı bu tez çalışmasında tercih edilmiştir.

İnternet ortamında güvenilir bilgi akışı, TCP protokolü sayesinde gerçekleşir. TCP protokolü paketlerin güvenli ve sıralı bir şekilde hedefe teslim edileceğinin garantisini veren protokoldür [33].

Gönderici düğümün, gönderdiği veri miktarı alıcı düğüm tarafından sınırlanabilir. Bu mekanizmaya akış kontrolü adı verilir. Bu mekanizmadaki amaç göndericinin hızını ayarlayarak ağın aşırı yüklemelerden korunmasıdır.

Ayrıca TCP protokolü yüksek seviyede tıkanıklık kontrol mekanizması uygulamaktadır. Bu mekanizmadaki temel amaç, veri iletiminin yüksek hızda güvenilir bir şekilde gerçekleştirilmesidir. TCP tıkanıklık kontrolü, her bir veri kaynağı için ilk önce ağdaki kullanılabilir kapasite miktarını belirler ve böylece kaç tane paketin güvenli bir şekilde iletilebileceği öğrenilir. Daha sonra tıkanıklık penceresi adlı durum değişkeni her bağlantı üzerinde tanımlanır. Tıkanıklık penceresi, belirli bir zamanda ne kadar verinin iletilebileceğini sınırlamaktadır.

Mantık olarak, ağdaki tıkanıklık arttığı zaman tıkanıklık penceresi küçültülür, tıkanıklık azaldığı zaman ise tıkanıklık penceresi büyütülür. Fakat zaman aşımaları TCP tarafından bir tıkanıklık olarak görülür. Her bir zaman aşımında veri kaynağı tıkanıklık penceresini bir önceki değerin yarısı olacak şekilde günceller.

Bu tez çalışmasında, Türkiye örneğinde İnternet ağ altyapısının tıkanıklık analizi çıkartılmıştır. Bu nedenle TCP işleyişi ve dört farklı iç içe tıkanıklık kontrolü için kullanılan algoritmalar üzerinde durulmuştur. Bu algoritmalar yavaş başlangıç, tıkanıklık kaçınma, hızlı düzelme ve yeniden hızlı tekrar iletim olarak oluşturulmuştur. OPNET yazılımı ile NoDrop, Tahoe ve Reno senaryoları kurularak elde edilen grafiklerin analizleri yapılmış ve bu grafiklerin sonuçları karşılaştırılmıştır.

Uygulama sırasında ilk önce yavaş başlangıç ve tıkanıklık kaçınma algoritmalarının davranışları incelenmiştir. Daha sonra ise hızlı düzelme ve yeniden hızlı iletim algoritmaları üzerinde tıkanıklık kaçınma davranışlarındaki değişiklikler incelenmiştir.

TCP protokolünün eski sürümlerinde alıcı tarafından belirlenen pencere boyutu kapasitesine kadar çok sayıda segment, ağın içerisine yerleştirilerek sistem

başlatılırdı. Bu işlem, ana düğümle aynı LAN üzerine konularak gerçekleştirilir. Fakat gönderici ve alıcı ile yönlendiriciler arasındaki bağlantı yavaş ise ağ üzerinde farklı sorunlar ortaya çıkar.

Bazı ara yönlendirme paketlerinin kuyruğu vardır ve bu paketler yönlendirici kuyruk alanlarının dışında da çalışmaktadır. Bu durumu önlemek için kullanılan algoritmaya yavaş başlangıç denir [24].

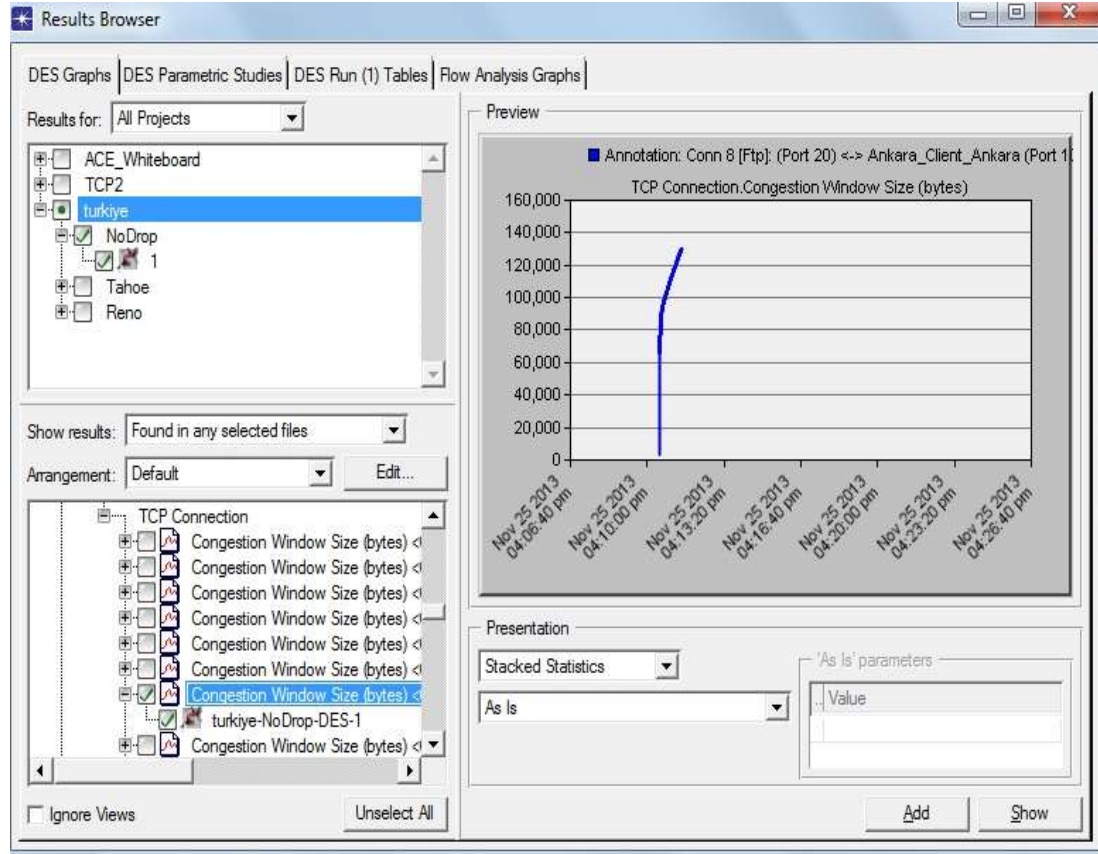
Ayrıca tıkanıklık penceresi boyutu tek bir paket boyutunun altına hiçbir zaman düşmez. Buna ek olarak tıkanıklık penceresinin boyutunu arttırmak için slow start (yavaş başlangıç) algoritması kullanılır. Bu algoritmaya göre tıkanıklık penceresi boyutu hızlı bir şekilde üssel olarak arttırılır.

Veri iletimi başlarken gerçekleşebilecek bilinmeyen durumlar ile ağın mevcut kapasitesinin yavaş yavaş belirlenmesi işlemi TCP tarafından gerçekleşir. Bu yüzden ağ trafik yükü azaltılır.

Yavaş başlatma algoritması gönderici TCP'ye başka bir pencere ekler. Buna tıkanıklık kaçınma penceresi denir [24]. Başka bir ağdaki ana bilgisayar ile yeni bir bağlantı kurulduğunda tıkanıklık penceresi bir segmentte genel olarak 512 byte büyüklüğünde başlatılır.

Tıkanıklık kaçınma ve yavaş başlangıç algoritmalarının kurulması için OPNET yazılımı üzerinde bir TCP projesi ve NoDrop adında yeni bir senaryo oluşturulmuştur. Bu senaryoda profil nesnesinin FTP özelliği üzerinde durulmuştur. Ayrıca çalışma alanı üzerindeki subnetin içerisinde temel anlamda Ethernet server, ethernet4\_slip8\_gtwy, Ethernet Workstation ve IP32\_Cloud nesnesi yerleştirilir. Bu senaryoda server nesnesinin TCP parametresi altında yer alan fast recovery ve fast retransmit özellikleri disable olarak ayarlanır. NoDrop senaryosunda server nesnesinin TCP connection ve congestion window size özellikleri incelenmiştir. Bu ağ kurulumu TCP'nin End to End iletim protokolü olarak adlandırılır. Tıkanıklık pencere boyutu farklı bir mekanizma ile kontrol edilmiştir. Bu ağ yapısının paket

kaybı açısından mükemmel olduğu varsayılır. Senaryonun simülasyonu belirli bir zaman aralığında çalıştırıldığında aşağıdaki grafik sonucunu vermiştir.

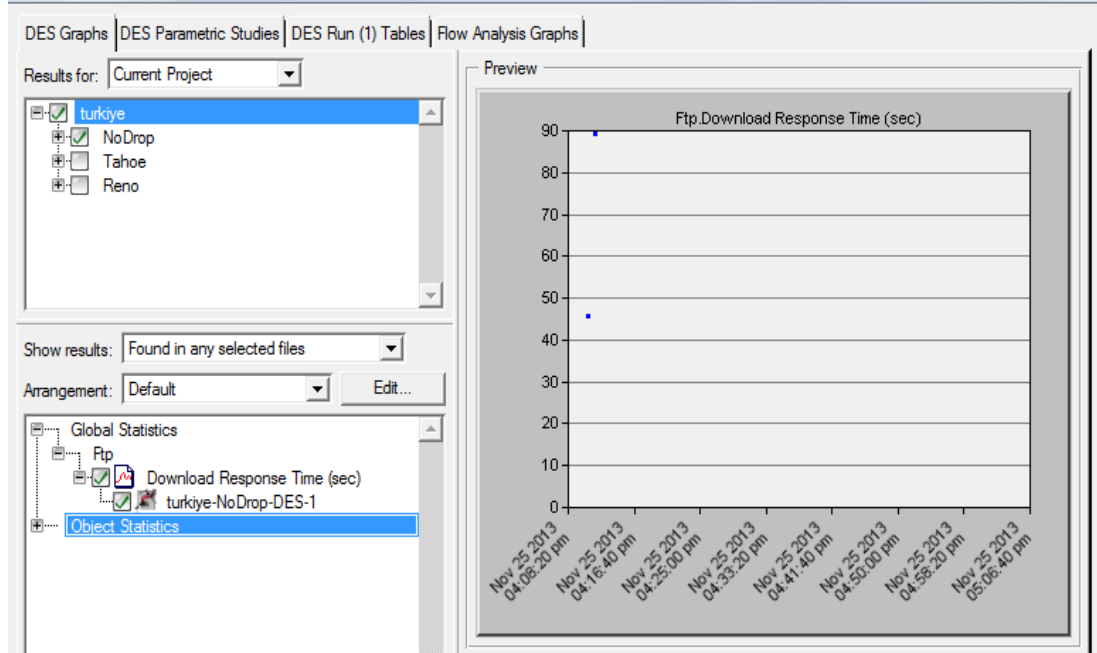


Şekil 4.1. NoDrop senaryoya ait grafik.

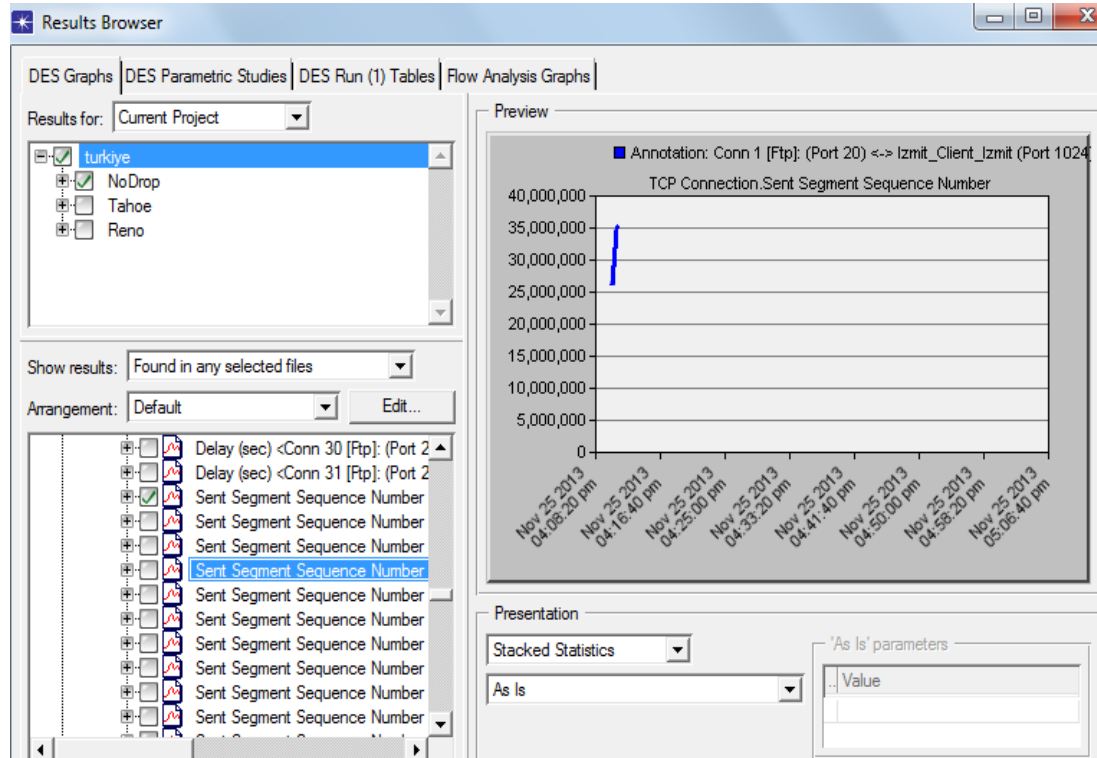
Yukarıdaki grafik NoDrop senaryoya ait simülasyon sonucunu göstermektedir. Ve bu senaryoda paket kaybı yaşanmaktadır. Aynı senaryoya ait throughput(verim) özelliğine bakıldığında ise FTP uzunluğunun üzerinde bir throughput ölçümü gözükmektedir. Bu grafiğe göre bulut sunucu ve istemcinin DS-3 hatlarını daha az tükettiği görülmektedir.

FTP Server ve TCP Connection nesnelerinin grafikleri uzaltılıp Sent Sequence number özelliği seçilirse, sıra numaralarının kaynağında kullanılan hız görülebilir. Bu oran direk olarak bağlantının verimi ile ilgilidir. Sunucu ne zaman aktif olarak bilgi gönderirse bu bilgileri dönüşümlü olarak dönemler arasında görebiliriz. Bu bağlantının verimi düşüktür çünkü pencere boyutu yeterince büyük değildir. TCP

kaynak pencerelerini tam bir pencere boyutu içerisinde gönderir fakat tekrar iletimde bir bekleme süresi karşımıza çıkmaktadır.



Şekil 4.2. Download Response Time Grafiği

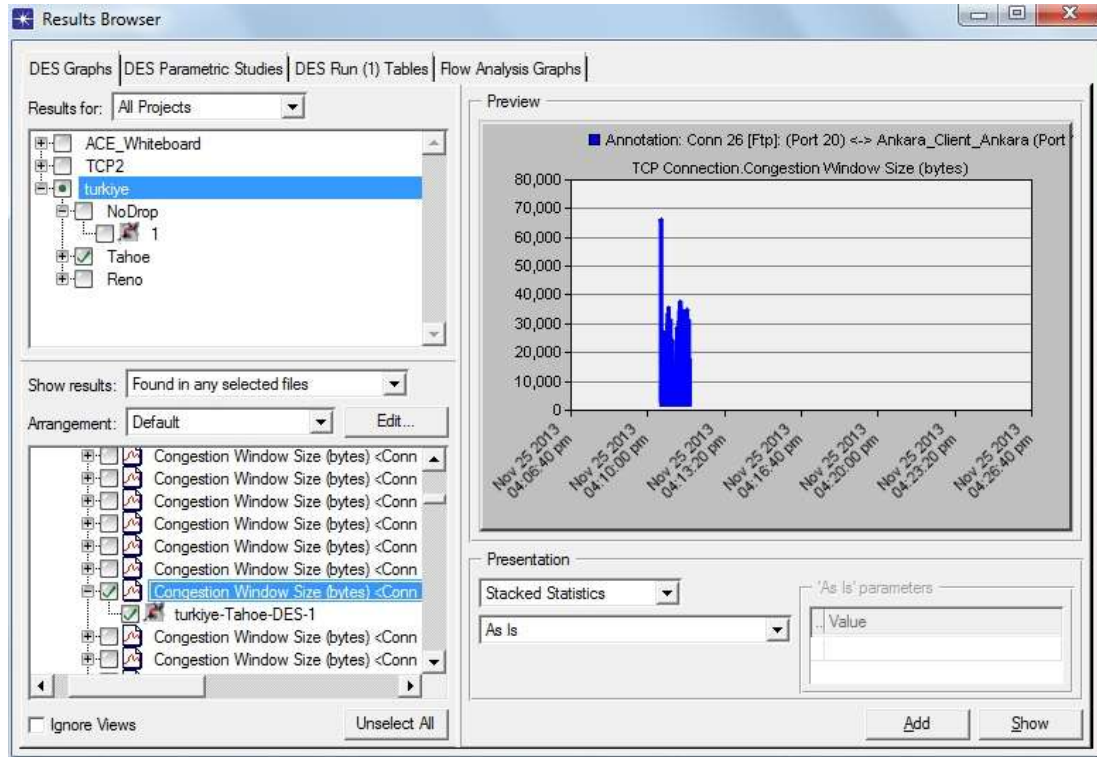


Şekil 4.3. Sent Sequence Number Grafiği

Yavaş başlangıç algoritmalarının sonucunda hızlı tekrar iletim (fast retransmit) ve hızlı düzelleme (fast recovery) algoritmaları gerçekleştirilebilir. Hızlı tekrar iletim algoritmasında kaybolmuş ya da bozulmuş bir paketin zaman aşımı süresinden önce tekrar iletilmesi söz konusudur.

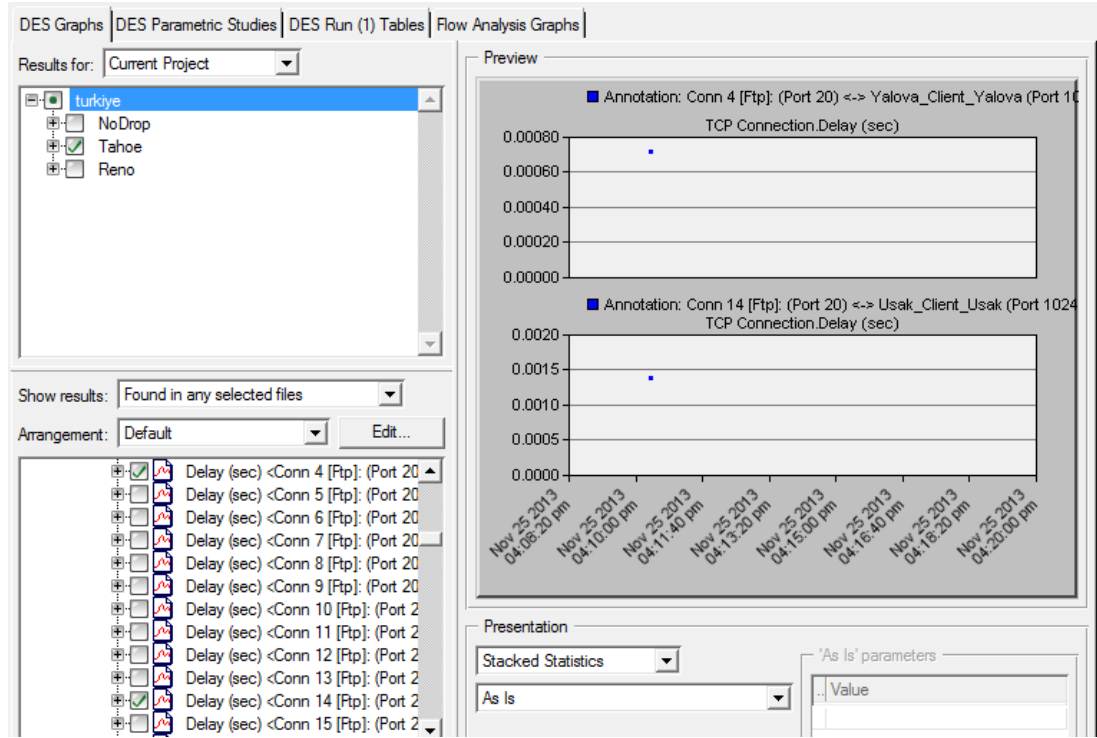
Tıkanıklık kaçınma algoritmasında, bazı değişiklikler yapılarak hızlı tekrar iletim algoritması elde edilir. Kayıpların tespiti ve onarımı için hızlı tekrar iletim algoritması kullanılır. Bu algoritma bir önceki senaryonun kopyalanması ile elde edilir. TCP iletim zamanlayıcısının dolmasını beklemeden ağ üzerinde eksik bölüm gibi görünen yerlerde yeniden bir iletim gerçekleştirir.

Nodrop senaryosu kopyalanarak adı Tahoe olarak düzenlenir. Oluşturulan senaryo üzerinde bulunan IP32\_Cloud nesnesinin Fast Retransmit özelliği Enable olarak değiştirilir. Ayrıca IP32\_Cloud nesnesinin Packet Discard Radio özelliği %0.5 olarak değiştirilir. Son olarak ise server nesnesinin TCP parametresi altında yer alan Fast Retransmit özelliği Enable olarak ayarlanır.



Şekil 4.4. Tahoe senaryosuna ait grafik.

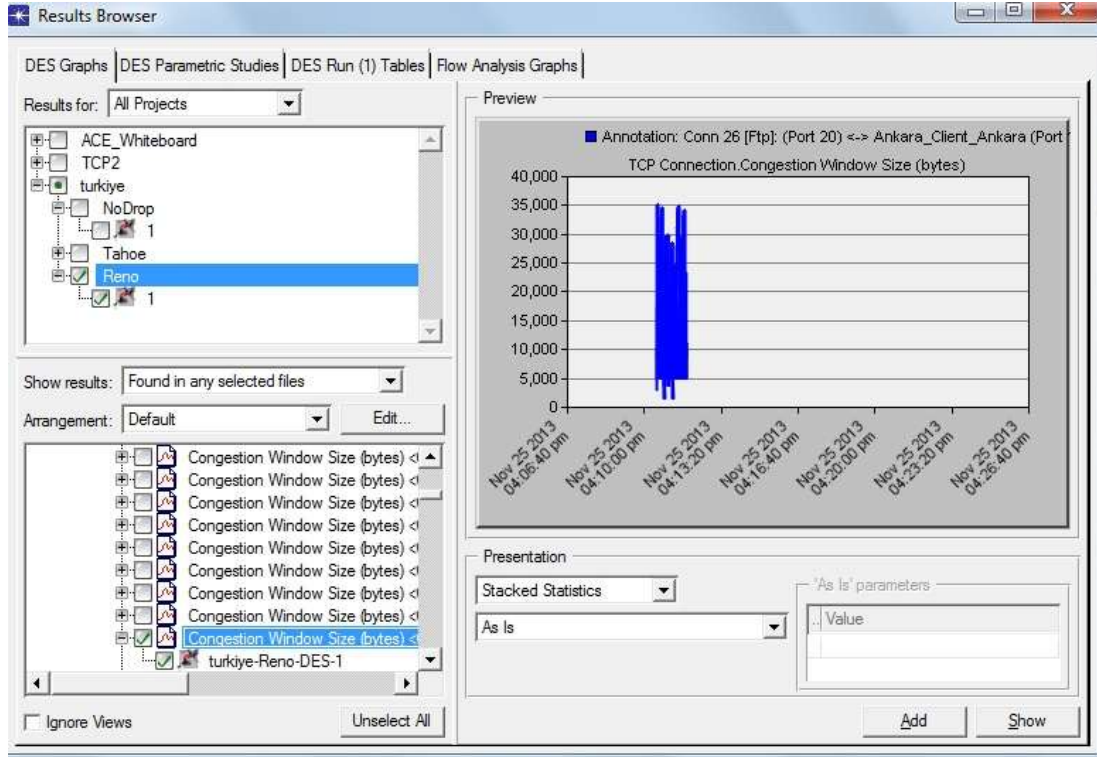
İkinci grafik Tahoe senaryosuna ait grafikdir. Ve %0.5 paket kaybı yaşanmaktadır. Yavaş başlangıç performanslı çalışmaktadır.



Şekil 4.5. Tahoe senaryosuna ait gecikme grafiği

Son senaryo olan Reno isimli senaryo da ise server nesnesinin TCP parametresi altında bulunan Fast Recovery özelliği Enabled olarak ayarlanır. Bu senaryoda yeniden hızlı algoritmanın eksik bölümü gibi görünen ağ yapısı, tıkanıklık kaçınma olmaksızın yavaş başlangıç gibi performans gösterir. Hızlı kurtarma özellikle geniş pencereler için orta seviye tıkanıklık altında yüksek performans sağlayan bir uygulamadır.

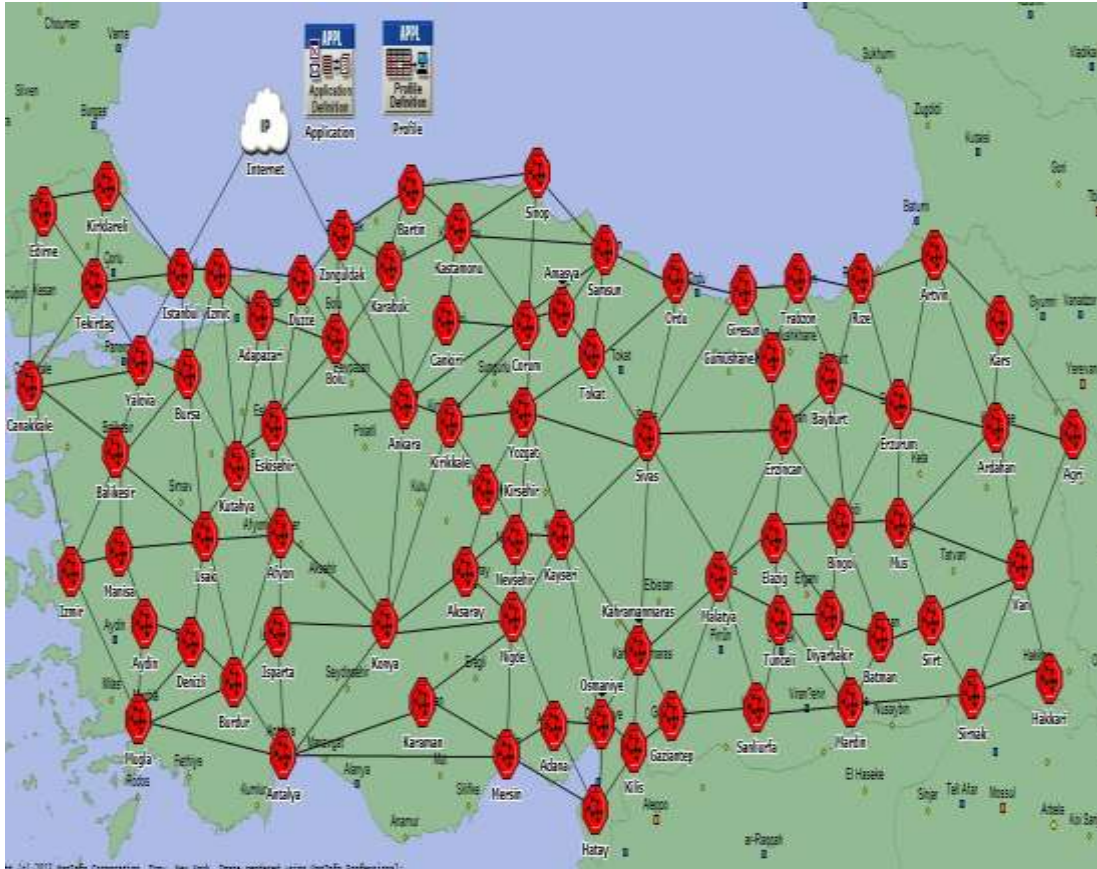




Şekil 4.6. Reno senaryosuna ait grafik.

Üçüncü grafik ise Reno senaryosuna ait grafik olup, bu senaryoda da %0.5 paket kaybı yaşanmaktadır. Yalnız bu senaryoda Congestion Window Size (tıkanıklık pencere boyutu) özelliği Tahoe senaryoda olduğu gibi sıfırın altına düşmez. Hızlı kurtarma yavaş başlangıca göre daha performanslı olarak çalışmaktadır.

Tüm algoritmaların ve senaryoların kullanıldığı genel ağ görünümü ise aşağıdaki şekilde gösterilmiştir.



Şekil 4.7. Genel ağ görünümü.

## **BÖLÜM 5. SONUÇLAR VE ÖNERİLER**

Sonuç olarak, OPNET yazılımı sayesinde kolaylıkla bilgisayar ağlarının modellenmesi ve ağ istatistiklerinin tutulması işlemleri gerçekleştirilir. OPNET yazılımında birçok farklı istatistik seçenekleri değerlendirilir. Bu istatistiklerden sadece belirli bir kısmı bu tez çalışmasında değerlendirilmiştir.

Bu çalışmada, Türkiye örneğinde İnternet ağ altyapısının tıkanıklık analizi için bir modelleme geliştirilmiştir. Bu modellemede üç farklı senaryo hazırlanmış ve bu senaryolar üzerinde Fast Retransmit, Fast Recovery, Slow Start ve Congestion Avoidance algoritmaları incelenerek simülasyon grafikleri elde edilmiştir.

Elde edilen grafikler yorumlandığında, Fast Recovery (hızlı kurtarma) özelliğine sahip simülasyonun daha yüksek performansta çalıştığı görülmüştür. Diğer özelliklerde ise paket kayıpları ile düşük performans gözlenmiştir.

Yapılan çalışma sonucunda ortaya çıkan analiz sonuçları, daha önce bu alanda çalışma yapan Tommy Svensson ve Alex Popescu'nun elde ettikleri paket kaybı değerleri ve tıkanıklık kontrolünün sonuçları ile örtüşmektedir.

Bu bilgiler ışığında bir ağ yapısındaki tıkanıklık ve paket kaybının giderilmesi için hazırlanan algoritmalar ortaya konmuştur. Ülkemizdeki mevcut İnternet ağının her geçen gün genişlediği düşünüldüğünde ise bu çalışmada kullanılan algoritmaların ülkemizin ağ yapısını rahatlatması açısından yol gösterici olacağı düşünülmektedir.

## KAYNAKLAR

- [1] MESTÇİ A., Türkiye İnternet Raporu 2007, XII. Türkiye’de İnternet Konferansı, 8 – 10 Kasım 2007.
- [2] AKTAŞ M., SAĞIROĞLU Ş., IPv6: Uluslararası Çalışmalar ve Türkiye’de Durum, Ulusal IPv6 Konferansı 2011.
- [3] ERDİN M. E., İnternet Protokolü Sürüm 6 (IPv6) Mimarisi Üzerinde Servis Sınıflarının Önceliklendirilmesi, Y. Lisans, Ankara Üniversitesi, Fen Bilimleri Enstitüsü, Şubat 2010, 79.
- [4] [http://www.ipv6.net.tr/docs/IPv6\\_Gecis\\_Projesi\\_makale.pdf](http://www.ipv6.net.tr/docs/IPv6_Gecis_Projesi_makale.pdf). (Ulusal IPv6 Protokol Altyapısı Tasarımı ve Geçiş Projesi), Erişim Tarihi: 10.11. 2013.
- [5] <http://www.hasanbalik.com/yayinlar/d/18.pdf>. (TCP/IP’nin Dünü, Bugünü, Yarını), Erişim Tarihi: 10.08. 2013.
- [6] PARLAK A., İnternet ve Türkiye’de İnternetin Gelişimi, Lisans, Fırat Üniversitesi, Mühendislik Fakültesi, Temmuz 2005, 76.
- [7] ZİHNİ H. G., TCP/IP ile Kablosuz Algılayıcı Ağlarına Erişim, Y. Lisans, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Haziran 2011, 73.
- [8] ÇÖLKESEN R., ÖRENCİK B., Bilgisayar Haberleşmesi ve Ağ Teknolojileri, Papatya Yayıncılık, Mayıs 2008.
- [9] OSMAN O., UÇAN O. N., Bilgisayar Ağları ve Ağ Güvenliği, Nobel Yayın Dağıtım, 2006.
- [10] ÖZSEVEN T., Bilgisayar Ağları, Murathan Yayınevi, 2013.
- [11] YILDIRIMOĞLU M., Her Yönüyle İnternetin Altyapısı TCP/IP, Pusula Yayıncılık ve İletişim, 2013.
- [12] ZENGİN A., EKİZ H., ÇOBANOĞLU B., TUNCEL S., Ağların Eğitimi ve Araştırılması için Devs Tabanlı Simülatör Tasarımı ve Uygulaması, 5. Uluslararası İleri Teknolojiler Sempozyumu, 13-15 Mayıs 2009.
- [13] TAŞKIN C., Ağ Teknolojileri ve Telekomünikasyon, Pusula Yayıncılık ve İletişim, 2009.

- [14] ÇETİN H., Türkiye'nin Otonom Sistem Seviyesinde İnternet Haritasının Çıkarımı ve İncelenmesi, Y. Lisans, Muğla Üniversitesi, Fen Bilimleri Enstitüsü, 81, 2009.
- [15] KUZU A., Bilgisayar Ağları ve İletişim, Nobel Yayın Dağıtım, 2010.
- [16] GEZGİN D. M., Kablosuz Ağ Teknolojileri ve Şifreleme, Doktora, Trakya Üniversitesi, Fen Bilimleri Enstitüsü, 149, 2011.
- [17] HENDERSON T.R., LACAGE M., RILEY G.F., Network Simulations with the Ns-3 Simulator, USA, 2008.
- [18] SİRAJ S., GUPTA K.A., BADGUJAR R., Network Simulation Tools Survey, International Journal of Advanced Research in Computer and Communication Engineering, 2012.
- [19] Geier J., Designing and Deploying 802.11n Wireless Networks, Cisco Press, 2010.
- [20] YANG F., ZHOU H., ZHANG L., FENG J., An Improved Security Scheme in WMAN Based on IEEE Standard 802.16, 2005.
- [21] VARGA A., Using the OMNET++ Discrete Event Simulation System in Education, IEEE Transactions on Education, 1999.
- [22] ARKUT İ. C., ARKUT R. C., İnternet ve Ağlarda Kaotik Büyüme, Journal of Kultur University, 3pp., 135-139, 2006.
- [23] CHANG X., Network Simulations With Opnet, Proceedings of the 1999 Winter Simulation Conference, 307 – 313, 1999.
- [24] SVENSSON T., POPESCU A., Development of Laboratory Exercises Based on Opnet Modeler, Master, Blekinge Institute of Technology, Electrical Engineering, 268, 2003.
- [25] DEVELİ H., Süleyman Demirel Üniversitesi Kampüs Ağının Opnet ile Modellenmesi, Y.Lisans, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, 60, 2009.
- [26] ADAMOS V., Greek Business Network, Master, University of Portsmouth, Department of Electronic & Computer Engineering, 120, 2004.
- [27] <http://www.slideshare.net/mooncrown/trkiye-internet-raporu-2013>, (Türkiye İnternet Raporu 2013), Erişim Tarihi: 05.12.2013.
- [28] <http://www.webrazzi.com/2013/08/23/dunyada-en-hizli-internet-kullanan-ulkeleri-ve-turkiyenin-durumu/> (Dünyada En Hızlı İnternet Kullanan Ülkeleri ve Türkiye'nin Durumu), Erişim Tarihi: 05.12.2013.

- [29] RAKESH K.J., IDRIS Z.B., UPENA D.D., Location Based Performance of WIMAX Network for Qos with Optimal Base Stations (BS), *Wireless Engineering and Technology*, 135-145, 2011.
- [30] VENKATA NP., LAKSHMINRAYANAN S., An Investigation of Geographic Mapping Techniques for Internet Hosts, USA, 2001.
- [31] Alice S.L.K., Modeling Simulation and Performance Evaluation of Telecommunication Networks, Master, University of Manitoba, Department of Electrical and Computer Engineering, 225, 1999.
- [32] NALBATÇI İ., BAYILMIŞ C., İSKEFİYELİ M., KIRBAŞ İ., WIMAX Ağlarda Çoklu Ortam Trafiklerinin OPNET Kullanılarak Başarım Analizi, *APJES*, 26-40, 2013.
- [33] LUCIO F.G., FARRERA P.M., JAMMEH E., FLEURY M., REED M.J., OPNET Modeler and Ns2: Comparing the Accuracy of Network Simulators for Packet-Level Analysis using a Network Testbed, 2003.
- [34] MARCIS I., *Computer Networks – Performance And Quality Of Service*, April 2010.
- [35] ALLMAN M., PAXSON V., STEVENS W., *TCP Congestion Control*, April 1999.
- [36] STEVENS V.R., *TCP/IP Illustrated, Volume:1: The Protocols*, Reading, Massachusetts: Addison-Wesley, 1994.

## EKLER

### OPNET’te Gerçekleştirilen Modelleme Adımları

- 1- OPNET’te File – New menüsünden Project komutu verilerek yeni bir proje oluşturulur.
- 2- Projeye (TCP) ve senaryoya (NoDrop) isim verilerek OK butonuna basılır.
- 3- Create Empty Scenario seçeneği seçilir ve Next butonuna basılır.
- 4- Choose from maps seçeneği tıklanır ve Next butonuna basılır.
- 5- Haritadan Avrupa seçeneği seçilir ve Next butonuna basılır.
- 6- OK butonuna basılarak proje alanına erişilir.
- 7- Bir adet Application Config nesnesi eklenir.
- 8- Applications nesnesi üzerinde sağ tıklanarak Edit Attributes seçeneği seçilir.
- 9- Karşımıza gelen pencerede Application Definitions satırına tıklanarak Edit özelliği seçilir.
- 10- Satır değeri 1 olarak ayarlanır.
- 11- Application Name özelliği FTP\_Application olarak değiştirilir.
- 12- Application Definitions satırına gidilerek Row değeri sıfır, Description değeri Edit olarak seçilir.
- 13- Karşımıza gelen değerler aşağıda gösterildiği gibi ayarlanır.

Attribute	Value
Command Mix (Get/Total)	100%
Inter Request Time (seconds)	Constant(3600)
File Size (bytes)	Constant(9000000)
Symbolic Server Name	FTP Server
Type of Service	Best Effort (0)
RSVP Parameters	None
Back – End Custom Application	Not Used

- 14- OK butonuna basılarak Application Attributes penceresi kapatılır.

- 15- Çalışma alanına bir adet Profile Config eklenir.
- 16- Profile nesnesinin üzerinde sağ tıklanarak Edit Attributes özelliği seçilir.
- 17- Profile Configuration satırındaki value değerinden Edit özelliği seçilir.
- 18- Rows değeri 1 olarak ayarlanır.
- 19- Profile Name özelliği FTP\_Profile olarak değiştirilir. Operation mode seçeneği Serial (Order) olarak düzenlenir.
- 20- Start Time değeri için constant (100), Duration değeri için End of Simulation seçilir.
- 21- Repeatability özelliği için Once At Start time seçilir.
- 22- Applications sütunundan Edit seçilir.
- 23- Rows değeri 1 olarak ayarlanır.
- 24- Name özelliği FTP\_Application olarak değiştirilir.
- 25- Start Time Offset özelliği için constant (5), Duration özelliği için End of Profile seçilir.
- 26- Repeatability özelliği Once At Start Time olarak kurulur.
- 27- OK butonuna basılarak Applications Table kapatılır.
- 28- OK butonuna basılarak Profile Configuration Table kapatılır.
- 29- OK butonuna basılarak Profile Attributes kapatılır.
- 30- Çalışma alanına bir subnet yerleştirilir.
- 31- Subnetin Name özelliği İstanbul olarak ayarlanır.
- 32- İstanbul subnetinin üzerinde çift tıklanarak içerisine girilir.
- 33- Bir adet Ethernet\_server eklenerek name özelliği Server\_Istanbul olarak değiştirilir.
- 34- Çalışma alanına Ethernet4\_slip8\_gtwy eklenerek name özelliği Router\_Istanbul olarak değiştirilir.
- 35- Server ve Router 100BaseT kablo ile birbirlerine bağlanır.
- 36- Server\_Istanbul nesnesi üzerinde sağ tıklanır ve Edit Attributes özelliği seçilir.
- 37- Application:Supported Services özelliğine seçilerek Edit özelliği tıklanır.
- 38- Karşımıza gelen yeni pencerede Rows değeri 1 olarak ayarlanır.
- 39- Name özelliği için FTP\_Application seçilir ve OK butonuna basılır.
- 40- Server Address özelliğine gidilir ve value değeri Server\_Istanbul olarak ayarlanır.



- 41- TCP Parameters deęerinin yanındaki “+” iřaretine tıklanarak uzatılır.
- 42- Fast Retransmit ve Fast Recovery için Disable seęilir.
- 43- OK butonuna basılır.
- 44- File – Save ile proje kaydedilir.
- 45- Go to the nest higher level butonuna tıklanarak bir üst seviyeye geçilir.
- 46- Çalıřma alanına bir subnet daha yerleřtirilir ve Name özellięi Ankara olarak deęiřtirilir.
- 47- Ankara subnetine çift tıklanarak ierisine girilir.
- 48- Çalıřma alanına bir adet Ethernet\_wkstn eklenir. Name özellięi Client\_Ankara olarak deęiřtirilir.
- 49- Ardından çalıřma alanına bir adet ethernet4\_slip8\_gtwy eklenir ve name özellięi Router\_ankara olarak ayarlanır.
- 50- Client ve Router 100BaseT kablo ile birbirine baęlanır.
- 51- Client\_Ankara nesnesi üzerinde saę tıklanarak Edit Attributes özellięi seęilir.
- 52- Application:Supported Profiles özellięine gidilerek Edit özellięi seęilir.
- 53- Rows deęeri 1 olarak ayarlanır.
- 54- Profile Name özellięi FTP\_Profile olarak deęiřtirilir.
- 55- Ok butonuna basılır.
- 56- Client Address özellięindeki value deęeri Client\_Ankara olarak ayarlanır.
- 57- Application:Destination Preferences özellięine gidilir, Edit özellięi seęilir.
- 58- Rows deęeri 1 olarak ayarlanır.
- 59- Symbolic Name deęeri FTP\_Server olarak deęiřtirilir.
- 60- Actual Name sütununa tıklanır.
- 61- Name özellięi için Server\_Istanbul girilir ve OK butonuna basılarak Actual Name Table penceresi kapatılır.
- 62- OK butonuna basılarak Application:Destination Preferences penceresi kapatılır.
- 63- OK butonuna basılarak Client\_Ankara Attributes penceresi kapatılır.
- 64- File – Save özellięi ile proje kaydedilir.
- 65- Go to the nest higher level butonuna basılarak bir üst seviyeye çıkılır. Harita üzerine dięer iller için subnet’ler eklenerek Ankara subnet’in de olduęu gibi ayarlar tekrarlanır.
- 66- Çalıřma alanına bir adet ip32\_Cloud eklenir.

- 67- Name özelliği internet olarak değiştirilir.
- 68- Istanbul subneti ile internet adlı ip32\_Cloud birbirine PPP\_DS3 kablo ile birbirine bağlanır.
- 69- Bu esnada karşımıza gelen pop-up menüde Istanbul.Router\_Istanbul seçeneği seçilir.
- 70- Ankara subneti ile internet adlı ip32\_Cloud birbirine PPP\_DS3 kablo ile bağlanır.
- 71- Bu esnada karşımıza gelen pop-up menüde Ankara.Router\_Ankara özelliği seçilir.
- 72- Istanbul subnetine girilir.
- 73- Server\_Istanbul nesnesinin üzerinde sağ tıklanarak Choose Individual Statistics özelliği seçilir.
- 74- Node Statistics özelliği uzatılarak TCP Connection ve Congestion Window Size (bytes) özellikleri seçilir.
- 75- Congestion Window Size(bytes) özelliğine sağ tıklanarak Change Collection Mode seçilir.
- 76- Advanced seçim kutusu işaretlenir.
- 77- Capture Mode özelliği All Values olarak değiştirilir.
- 78- OK butonuna basılarak diyalog penceresi kapatılır.
- 79- OK butonuna basılarak Choose Results penceresi kapatılır.
- 80- File-Save özelliği ile proje kaydedilir.
- 81- Simulation özelliğinden Configure Discrete Event Simulation özelliği seçilir.
- 82- Duration özelliği için 15 minutes değeri girilir.
- 83- Run butonuna basılarak simülasyon başlatılır.
- 84- Simülasyon bittiğinde Close butonuna basılır.
- 85- Wiev Results özelliği üzerinde sağ tıklanır.
- 86- Object Statistics değeri uzatılarak Choose from maps networks özelliğinin altından Istanbul - Server\_Istanbul – TCP Connection – Congestion Window Size özellikleri seçilir.
- 87- Show butonuna basılır.
- 88- Scenarios menüsünden Duplicate Scenario özelliği seçilir.
- 89- Senaryo adı Tahoe olarak değiştirilir.
- 90- IP32\_Cloud'un üzerinde sağ tıklanır ve Edit Attributes özelliği seçilir.

- 91- Packet Discard Ratio oranı 0.5% olarak ayarlanır.
- 92- OK butonuna basılır.
- 93- Istanbul subnetine girilir.
- 94- Server\_Istanbul nesnesi üzerinde sağ tıklanır ve Edit Attributes özelliği seçilir.
- 95- TCP Parameters özelliği uzatılır.
- 96- Fast Retransmit değeri Enabled olarak değiştirilir.
- 97- OK butonuna basılarak File-Save işlemi ile proje kaydedilir.
- 98- Scenarios menüsünden Duplicate Scenarios özelliği seçilir.
- 99- Senaryo adı Reno olarak değiştirilir.
- 100- Istanbul nesnesine girilir.
- 101- Server\_Istanbul nesnesi üzerinde sağ tıklanır ve Edit Attributes özelliği seçilir.
- 102- TCP Parameters özelliği uzatılarak Fast Recovery özelliği Enabled yapılır.
- 103- Ok butonuna basılarak File-Save işlemi ile proje kaydedilir.
- 104- Scenarios menüsünden Manage Scenarios özelliği seçilir.
- 105- Results değerleri collect ya da recollect olarak değiştirilir.
- 106- OK butonuna basılır.
- 107- Sistem kendi kendine simülasyonu başlatır.
- 108- Simülasyon bittiğinde Close butonuna basılır.
- 109- Çalışma alanında sağ tıklanarak View Results komutuna tıklanır.
- 110- Object Statistics değeri uzatılarak Choose From Maps Network – Istanbul - Server\_Istanbul – TCP Conneciton özelliklerinin altında yer alan Congestion Window Size (bytes) seçilir.
- 111- Statistics Stacked değeri seçili olmalıdır.
- 112- Show butonuna basılarak sonuçlar görsel olarak izlenir.

## ÖZGEÇMİŞ

Hakan Can ALTUNAY, 16.06.1983' de Samsun' da doğdu. İlk, orta ve lise eğitimini Samsun'da tamamladı. 2001 yılında Atakum Anadolu Teknik Lisesi, Elektronik Bölümünden mezun oldu. 2002 yılında başladığı Sakarya Üniversitesi Teknik Eğitim Fakültesi Elektronik – Bilgisayar Eğitimi bölümünü 2006 yılında bitirdi. 2006 – 2010 yılları arasında Özel bir Eğitim Kurumunda Bilgisayar Öğretmeni olarak çalıştı. 2010 yılından beri Ondokuz Mayıs Üniversitesi Çarşamba Ticaret Borsası Meslek Yüksekokulu Bilgisayar Teknolojileri Bölümü'nde Öğretim Görevlisi olarak çalışmaktadır.