

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**GENİŞ ÖLÇEKLİ AĞLAR İÇİN YENİ BİR DAĞITIK  
AYRIK OLAY TABANLI BENZETİM YAKLAŞIMI VE  
UYGULAMASI**

**DOKTORA TEZİ**

**Bülent ÇOBANOĞLU**

**Enstitü Ana Bilim Dalı** : **ELEKTRONİK - BİLGİSAYAR EĞİTİMİ**  
**Tez Danışmanı** : **Prof. Dr. Hüseyin EKİZ**  
**Ortak Danışman** : **Yrd. Doç. Dr. Ahmet ZENGİN**

**Aralık 2011**

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**GENİŞ ÖLÇEKLİ AĞLAR İÇİN YENİ BİR DAĞITIK  
AYRIK OLAY TABANLI BENZETİM YAKLAŞIMI VE  
UYGULAMASI**

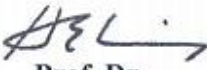
**DOKTORA TEZİ**

**Bülent ÇOBANOĞLU**

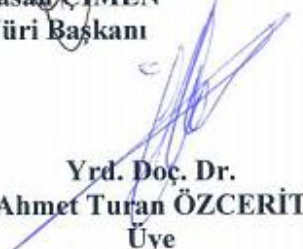
**Enstitü Anabilim Dalı : ELEKTRONİK-BİLGİSAYAR EĞİTİMİ**

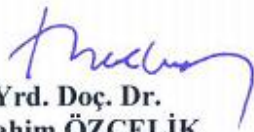
**Bu tez 23/12/2011 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.**

  
**Prof. Dr.  
Hasan ÇİMEN  
Jüri Başkanı**

  
**Prof. Dr.  
Hüseyin EKİZ  
Üye**

  
**Prof. Dr.  
İsmail ERTÜRK  
Üye**

  
**Yrd. Doç. Dr.  
Ahmet Turan ÖZCERİT  
Üye**

  
**Yrd. Doç. Dr.  
İbrahim ÖZÇELİK  
Üye**

## ÖNSÖZ

Aksiyoner bir akademisyenin nasıl olması gerektiğini şahsında bizlere gösteren, doktora çalışmam boyunca bana destek olan, ümidi kestiğimiz anlarda tekrardan tetikleme sinyali verip yol gösteren, çok kıymetli hocam ve danışmanım Sayın Prof. Dr. Hüseyin EKİZ' e teşekkürü bir borç bilirim.

Tez konum ile ilgili yaptığı yönlendirmelerle tezin gerçekleştirilmesinde, başlangıcından sonuna kadar karşılaştığım problemlerin çözümünde, tecrübesini, ilmini, sabrını benden esirgemeyen ve hep tünelin ucundaki ışığı gösteren, danışmanlık yaptığı öğrencilere bütün bilgi birikimini aktarmaya çalışan ortak danışmanım Sayın Yrd. Doç. Dr. Ahmet ZENGİN' e sonsuz şükranlarımı sunarım.

Doktora tez döneminde altı aylık zaman dilimlerinde tez çalışması ile ilgili değerli görüşleri ile bu teze katkılarını esirgemeyen Sayın jüri üyelerine ayrıca teşekkür ederim.

Belki de benden çok tezin bitmesini arzulayan, maddi-manevi desteklerini esirgemeyen isimleri bende mahfuz uzak-yakın bütün arkadaşlarıma, aile fertlerime, özellikle de eşime ve kızıma teşekkür ederim.

Bu tez çalışması, Sakarya Üniversitesi Bilimsel Araştırma Projeleri (BAP) Komisyon Başkanlığı tarafından desteklenmiştir [proje numarası:2010-50-02-009].

# İÇİNDEKİLER

ÖNSÖZ .....	ii
İÇİNDEKİLER .....	iii
SİMGELER VE KISALTMALAR LİSTESİ .....	vii
ŞEKİLLER LİSTESİ .....	x
TABLolar LİSTESİ .....	xii
ÖZET .....	xiii
SUMMARY .....	xiv

## BÖLÜM 1.

GİRİŞ .....	1
1.1. Geniş Ölçekli Ağ Modelleme Problemleri .....	2
1.2. Önerilen Yöntem .....	4
1.3. Literatürde Yapılan Çalışmaların Özeti .....	5
1.4. Tez Çalışmasının Bilime Katkısı .....	9
1.4. Tez Düzeni .....	11

## BÖLÜM 2.

MODELLEME VE BENZETİM TEORİSİ .....	14
2.1. Giriş .....	14
2.2. Modelleme ve Benzetim .....	15
2.3. Modelleme ve Benzetim Çerçevesi .....	20
2.3.1. Deneysel çerçeve .....	22
2.4. Modelleme ve Benzetimin Olumlu ve Olumsuz Yönleri .....	22
2.5. Modelleme Yöntemleri .....	24
2.5.1. Ayrık olaylı modelleme .....	24
2.6. DEVS Modelleme ve Benzetim (M&S) Yaklaşımı .....	27
2.6.1. Klasik DEVS yaklaşımı .....	29

2.6.2. Paralel DEVS yaklaşımı .....	31
2.7. DEVS-Suite Benzetim Ortamı.....	32
BÖLÜM 3.	
PARALEL ve DAĞITIK BENZETİM .....	37
3.1. Giriş .....	37
3.2. Dağıtık Teknolojiler ve Dağıtık Benzetim .....	40
3.2.1. İstemci / Sunucu mimarisi .....	41
3.2.2. Java Uzak Metot Çağrısı (Java Remote Method Invocation- JAVA RMI) .....	43
3.2.3. Ortak Nesne İstem Aracı Mimarisi (Common Object Request Broker Architecture- CORBA).....	43
3.2.4. Yüksek Seviyeli Yapı (High Level Architecture -HLA).....	44
3.3. Dağıtık Ağ Modelleme ve Benzetim Süreci .....	44
BÖLÜM 4.	
YÖNLENDİRME PROTOKOLLERİ .....	47
4.1. Giriş .....	47
4.2. Yönlendirme Algoritma ve Protokolleri.....	48
4.3. Yönlendirme Protokollerinin Sınıflandırılması .....	51
4.3.1. Uzaklık Vektörü (Distance-Vector) protokolleri.....	52
4.3.1.1. Yönlendirme Bilgilendirme Protokolü (RIP).....	53
4.3.1.2. Dahili Ağ Geçidi Yönlendirme Protokolü (IGRP).....	54
4.3.2. Bağlantı Durumu (Link-State) protokolleri .....	55
4.3.2.1. İlk Açık En Kısa Yol (OSPF) Protokolü.....	56
4.3.2.2. Sınır Geçit Protokolü (Border Gateway Protocol- BGP)....	59
BÖLÜM 5.	
AĞ BENZETİM ARAÇLARI .....	61
5.1. Giriş .....	61
5.2. Ağ Tasarım ve Benzetim Araçlarının Sınıflandırılması .....	62
5.3. Topoloji Üretim Araçları .....	63
5.4. Benzetim Araçları .....	65

5.4.1. OMNET++.....	67
5.4.2. ns-2 (Network Simulator-2).....	69
5.4.3. OPNET (Optimised Network Engineering Tool).....	70
5.4.4. DEVS-Suite .....	71

## BÖLÜM 6.

YENİ BİR DAĞITIK BENZETİM ALGORİTMASININ GELİŞTİRİLMESİ VE D-DEVSNET AĞ BENZETİM ARACININ TASARIMI.....	73
6.1. Giriş .....	73
6.2. D-DEVSNET Ağ Benzetim Aracı Geliştirme Süreci.....	74
6.3. Ağ Bileşenlerinin Tanımlanması ve Tasarımı .....	78
6.3.1 Atomik modeller.....	80
6.3.1.1. Sunucu düğüm atomik modeli .....	81
6.3.1.2. İstemci düğüm atomik modeli.....	82
6.3.1.3. Yönlendirici Atomik Modeli.....	84
6.3.2. Birleşik Modeller .....	87
6.3.2.1. Otonom Sistem modeli.....	87
6.3.3. D-DEVSNET deneysel çerçevesi .....	88
6.4. OSPF Protokol Modeli .....	90
6.5. Yönlendirme Modülü.....	94
6.6. BRITE Topoloji Üretim Aracı.....	96

## BÖLÜM 7.

D-DEVSNET BENZETİM ARACININ DOĞRULANMASI ve GEÇERLENMESİ .....	98
7.1. Giriş .....	98
7.2. D-DEVSNET Doğrulama ve Geçerleme Süreci.....	98
7.3. D-DEVSNET Kavramsal Modelinin Geçerlenmesi .....	102
7.4. D-DEVSNET Protokol ve Benzetim Aracı Geçerlenmesi .....	104

## BÖLÜM 8.

BENZETİM SONUÇLARI.....	109
8.1. Giriş .....	109

8.2. Başarım Değerlendirmesi .....	111
8.2.1. İş çıkarma yeteneği .....	111
8.2.2. Uçtan uca ortalama paket gecikmesi .....	112
8.2.3. Kayıp paket sayısı.....	113
8.2.4. Hız ve bellek tüketimi.....	113
BÖLÜM 9.	
SONUÇLAR VE DEĞERLENDİRME .....	115
9.1. Sonuçlar .....	115
9.2. Tartışma ve Öneriler .....	119
KAYNAKLAR .....	122
EKLER.....	131
Ek A. D-DEVSNET Sınıf Diyagramı.....	131
ÖZGEÇMİŞ .....	132

## SİMGELER VE KISALTMALAR LİSTESİ

ALSP	:Tümleşik Seviye Benzetim Protokolü (Aggregate Level Simulation Protocol)
AM	:Atomik model
API	:Uygulama Ara yüzü
ARPANET	:Amerikan Gelişmiş Savunma Araştırmaları Dairesi Ağı ( Advanced Research Projects Agency Network)
AS	:Otonom Sistem
BGP	:Sınır Geçit Protokolü ( Border Gateway Protokol )
BRITE	:The Boston University Reperetative Internet Topology Generator
CM	:Birleşik Model (Coupled Model)
CORBA	:Common Object Request Broker Architecture
CPU	:Merkezi işlem birimi
CSV	:Virgülle Ayrılmış Değerler (Comma Seperated Values)
DESS	:Diferansiyel Denklemler Sistem Tanımı (Differential Equation System Specification)
DES	:Ayrık olaylı sistemler
DEVS	:Ayrık Olaylı Sistem Tanımı (Discrete Event System Specification)
DIS	: Distributed Interactive Simulation (Dağıtık Etkileşimli Benzetim)
D-DEVSNET	:DEVS tabanlı dağıtık mimarili ağ benzetim aracı
EF	:DeneySEL Çerçeve (Experimental Frame)
DTE	:DEVS izleme ortamı (DEVS Tracking Environment)
EGP	:Harici Ağ Geçidi Protokolü (Exterior Gateway Protocol)
EIC	:Harici giriş bağlantısı
EIGRP	:Enhanced Interior Gateway Routing Protocol
EOC	:Harici çıkış bağlantısı
EF	:DeneySEL Çerçeve (Experimental Frame)
FIFO	:İlk Giren İlk Çıkar (First-In, First-Out)
GloMoSim	:Global Mobile System Simulator



GUI	:Grafiksel kullanıcı ara yüzü (Graphical User Interface)
HLA	:Yüksek Seviyeli Yapı (High Level Architecture)
IC	:Dâhili bağlantılar
ID	:Kimlik
JRE	:Java Çalışma Ortamı (Java Runtime Environment)
J-SİM	:JavaSim ağ benzetim aracı
IEEE	:Elektrik Elektronik Mühendisleri Enstitüsü (Institute of Electrical and Electronical Engineers)
IETF	:İnternet Mühendisliği Görev Gücü (Internet Engineering Task Force)
IGP	:Dâhili Ağ Geçidi Protokolü (Interior Gateway Protocol)
I/O	:Giriş ve Çıkış
IP	:İnternet Protokolü
ISC	:İnternet Sistemleri Konsorsiyumu (Internet Systems Consortium)
IS-IS	:Intermediate System To Intermediate System
JNS	:Java Network Simulator
JVM	:Java Sanal Makinesi (Java Virtual Machine)
KB	:Kilobayt
LAN	:Yerel alan ağı (Local Area Network)
LSA	:Bağlantı Durumu İlanları (Link State Advertisement)
LSDB	:Bağlantı Durumu Veri tabanı (Link State Database)
MAC	:Ortam Erişim Denetim Katmanı (Media Access Control)
MAN	:Kentsel alan ağı (Metropolitan Area Network)
M&S	:Modelleme ve Benzetim
MFVC	:Model Façade View Controller
MTU	:Maksimum aktarım birim değeri ( Maximum Transmission Unit )
n	:Düğüm sayısı
NIC	:Ağ arabirim kartı
ns-2	:Network Simulator - 2
ns-3	:Network Simulator - 3
OS	:İşletim Sistemi (Operating System)
OMNET++	:Objective Modular Network Test-bed in C++
OPNET	:OPTimised Network Engineering Tool
ORB	:Nesne İstek Aracı (Object Request Broker)

OSPF	:Açık En Kısa İlk Yol (Open Shortest Path First)
QoS	:Hizmet Kalitesi (Quality of Service)
RFC	:Açıklama İsteği (Request For Comments)
RIP	:Yönlendirme Bilgilendirme Protokolü (Routing Information Protocol)
RIPv1	:RIP Sürüm-1
RIPv2	:RIP Sürüm-2
RMI	:Uzak metot çağrısı (Remote Method Invocation)
RTI	:Çalışma Anı Altyapısı (Run-Time Infrastructure)
SPF	:En Kısa İlk Yol (Shortest Path First)
TCP	:İletim Kontrol Protokolü (Transmission Control Protocol)
TTL	:Yaşama Zamanı (Time to Live)
UDP	:Kullanıcı Veri bloğu İletişim Kuralları (User Datagram Protocol )
UML	:Birleşik Modelleme Dili (Unified Modeling Language)
V&V	:Doğrulama ve Geçerleme (Verification & Validation)
PDNS	:Paralel / Dağıtık ağ benzetim aracı (Parallel / Distributed ns)
SSFNet	:Ölçeklenebilir benzetim çerçevesi (Scalable Simulation Framework)
WAN	:Geniş Alan Ağı (Wide Area Network)

## ŞEKİLLER LİSTESİ

Şekil 1.1.	İnternete bağlı düğüm / bilgisayar sayısı .....	1
Şekil 2.1.	Sistem, modelleme ve benzetim ilişkisi.....	14
Şekil 2.2.	Modelleme yöntemleri .....	16
Şekil 2.3.	Benzetim çalışmasının adımları [34, 41] .....	19
Şekil 2.4.	Modelleme ve Benzetim (M&S) Çerçevesi: Varlıklar ve İlişkileri .....	21
Şekil 2.5.	Deneysel çerçevenin yapısı .....	22
Şekil 2.6.	Ayrık olaylı sistem modelleme örneği: kasiyer / kuyruk sistemi .....	26
Şekil 2.7.	Ayrık olaylı benzetim çekirdeği [48]. .....	27
Şekil 2.8.	Benzetim sınıflandırması içerisinde DEVS yönteminin yeri.....	28
Şekil 2.9.	Paralel DEVS birleşik modeli .....	32
Şekil 2.10.	DEVS-Suite MFVC Paket Yapısı .....	35
Şekil 3.1a.	Dağıtık bir sistem .....	38
b.	Dağıtık bir sistem. ....	38
c.	Paralel bir sistem. ....	38
Şekil 3.2.	İstemci ve sunucu arasındaki iletişim .....	41
Şekil 3.3.	Java RMI mekanizması.....	43
Şekil 3.4.	CORBA mekanizması.....	43
Şekil 3.5.	Dağıtık ağ modelleme süreci algoritması .....	45
Şekil 4.1.	Örnek bir ağın yönlendirme tablosu .....	50
Şekil 4.2.	BGP protokolünün işlevi.....	60
Şekil 5.1.	Ağ Tasarım ve Benzetim Araçlarının Sınıflandırılması .....	63
Şekil 5.2.	OMNET++ Modül Yapısı.....	68
Şekil 5.3.	Yönlendirici Düğümleri İçeren Basit Bir Ağ Yapısının OMNET++ ile Gerçekleştirimi .....	68
Şekil 5.4.	Yönlendirici Düğümleri İçeren Basit Bir Ağ Yapısının NAM Editörde Görüntüsü .....	69
Şekil 5.5.	Yönlendirici Düğümleri İçeren Basit Bir Network Yapısının OPNET	

Görüntüsü.....	71
Şekil 5.6. D-DEVSNET Ağ Benzetim Aracının Görünümü.....	72
Şekil 6.1. İstemci - Sunucu mimarili dağıtık DEVS tabanlı ağ modelleme ve benzetim süreci.....	75
Şekil 6.2a. D-DEVSNET ağ benzetim aracının sunucu tarafı ekran görüntüsü .....	77
b. D-DEVSNET ağ benzetim aracının istemci tarafı ekran görüntüsü .....	78
Şekil 6.3. D-DEVSNET benzetim aracının kavramsal modeli ve bileşenleri.....	79
Şekil 6.4. Geliştirilen sunucu atomik düğümün D-DEVSNET ekran çıktısı .....	81
Şekil 6.5. Geliştirilen istemci atomik düğümün D-DEVSNET ekran çıktısı.....	83
Şekil 6.6. Yönlendirici atomik modelinin D-DEVSNET ekran çıktısı .....	84
Şekil 6.7. Yönlendirici atomik modelin kavramsal yapısı .....	85
Şekil 6.8. D-DEVSNET otonom sistem birleşik modeli.....	88
Şekil 6.9. D-DEVSNET grafiksel arayüzünde olay dönüştürücüde anlık izlenen veriler .....	89
Şekil 6.10. D-DEVSNET ağı deneysel çerçeve bağlantısı. ....	89
Şekil 6.11. OSPF protokolü durum geçişleri .....	93
Şekil 6.12. Bir yönlendirme tablosunun D-DEVSNET ekranındaki görünümü. ....	95
Şekil 6.13. D-DEVSNET BRUTE topoloji üretici ekran görüntüsü .....	97
Şekil 6.14. 1000 düğüm için BRUTE görüntüleyici.....	97
Şekil 7.1. D-DEVSNET OSPF protokolü doğrulama ve geçirme süreci .....	101
Şekil 7.2. OSI referans modelinin D-DEVSNET eşdeğeri .....	102
Şekil 7.3. D-DEVSNET ağ katman yapısının kavramsal modeli .....	103
Şekil 7.4a. 4 düğümlü ns-2 OSPF ağı .....	105
b. DEVS ağı .....	105
Şekil 7.5. Test Edilen Ağ Benzetim Araçlarının Ortalama Ağ Çıkışı .....	106
Şekil 7.6. Test edilen ağ benzetim araçlarına ait uçtan uca ortalama gecikme ....	107
Şekil 7.7. 3000 düğümlü bir D-DEVSNET ağı için paket kaybı .....	107
Şekil 8.1. 1000 yönlendiricili örnek bir ağda deneysel çerçeve üzerinde izlenen veriler .....	110
Şekil 8.2. Farklı ölçeklerde D-DEVSNET iş çıkarma yeteneği.....	112
Şekil 8.3. Farklı ölçeklerde D-DEVSNET uçtan uca ortalama gecikme .....	112
Şekil 8.4. Farklı ölçeklerde D-DEVSNET kayıp paket sayısı .....	113

## TABLolar LİSTESİ

Tablo 2.1.	M &S şemasındaki temel varlıkların açıklanması. ....	21
Tablo 3.1.	Paralel ve dağıtık uygulamaların farklılıkları .....	39
Tablo 4.1.	OSPF ile ilgili bazı RFC Tanımlamaları.....	57
Tablo 5.1.	Topoloji üretim araçlarının karşılaştırılması.....	64
Tablo 5.2.	Yaygın olarak kullanılan ağ benzetim araçları ve özellikleri .....	66
Tablo 7.1.	D-DEVSNET ve ns-2 ağ benzetim parametreleri.....	105
Tablo 7.2.	Karşılaştırılan benzetim araçlarının olay sıklığı .....	108
Tablo 8.1.	Yapılan deneylerde uygulanan benzetim parametreleri.....	110
Tablo 8.2.	Ağ benzetim araçlarının hız ve bellek tüketimi karşılaştırması.....	114

## ÖZET

Anahtar kelimeler: Paralel ve Dağıtık Benzetim, İstemci / Sunucu Mimarisi, Geniş Ölçekli Ağ, DEVS, DEVS-Suite, D-DEVSNET, OSPF

Modelleme ve benzetim teorisi ağ tasarım ve protokollerini test etme ve dinamik ağ davranışını anlama da önemli bir araçtır. En büyük ağ olan İnternetin baş döndürücü bir hızla büyümesi, yeni benzetim tekniklerini geliştirmeyi zorunlu kılmaktadır.

Yapılan çalışmada geniş ölçekli ağlar için ayrıık olay tabanlı yeni bir paralel ve dağıtık benzetim yaklaşımı ve ağ benzetim aracı geliştirilmiştir. DEVS modelleme yaklaşımı kullanılarak istemci / sunucu tabanlı, ölçeklenebilir, platformdan bağımsız, esnek 'D-DEVSNET' isimli yeni bir ağ benzetim aracının modelleme ve tasarımı gerçekleştirilmiş ve başarıml analizleri yapılmıştır.

D-DEVSNET ağ benzetim aracının başarımlını ve DEVS yaklaşımının paralel ve dağıtık uygulamalardaki gücünü göstermek amacıyla belirli bir zaman periyotunda farklı ölçeklerdeki ağlarda iş çıkarma yeteneđi, uçtan uca ortalama paket gecikmesi ve kayıp paket miktarı gibi testler yapılmıştır. Yapılan testler sonucunda D-DEVSNET benzetim aracının iyi bir ölçeklenebilirliğe ve yüksek bir performansa sahip olduđu, benzetim aracı performansının, ağdaki düğüm / yönlendirici sayısından (artmasından veya azalmasından) ve donanım kapasitesinden fazla etkilenmediđi görülmüştür.

Geliştirilen benzetim aracının, açık kaynak kodlu, paralel ve dağıtık mimaride, web tabanlı çalışabilme yeteneđi, benzetim sonuçlarının her aşamada kolay izlenebilirliđi, görselliđi, kolay kurulumu gibi özellikleri ile uzaktan eğitim çalışmaları başta olmak üzere eğitim amaçlı kullanımı hedeflenmiştir. Bu kapsamda geliştirilen D-DEVSNET ağ benzetim aracı, dağıtık, ölçeklenebilir, adaptif ve güçlü ağ uygulamalarının modellenmesi ve tasarımı için örnek bir çerçeve oluşturmuş, DEVS yaklaşımının geniş bir uygulama alanına sahip olduđu gösterilmiştir.

# **DESIGN AND IMPLEMENTATION OF A NEW DISCRETE EVENT BASED DISTRIBUTED SIMULATION APPROACH FOR LARGE-SCALE NETWORKS**

## **SUMMARY**

Key Words: Parallel and Distributed Simulation, Client / Server Architecture, Large-Scale Network, DEVS, DEVS-Suite, D-DEVSNET, OSPF

Modeling and simulation theory is an important tool for testing network design and protocols as well as understanding the dynamic behavior of the network. Growth of the biggest network, Internet's dazzling speed requires new simulation techniques.

In this study, a tool was developed for design and implementation of a new discrete event based distributed simulation for large-scale networks. By using DEVS as a modeling approach, client / server based, scalable, platform-independent, flexible new network modeling and simulation tool called 'D-DEVSNET' was designed and performance analyzes were conducted.

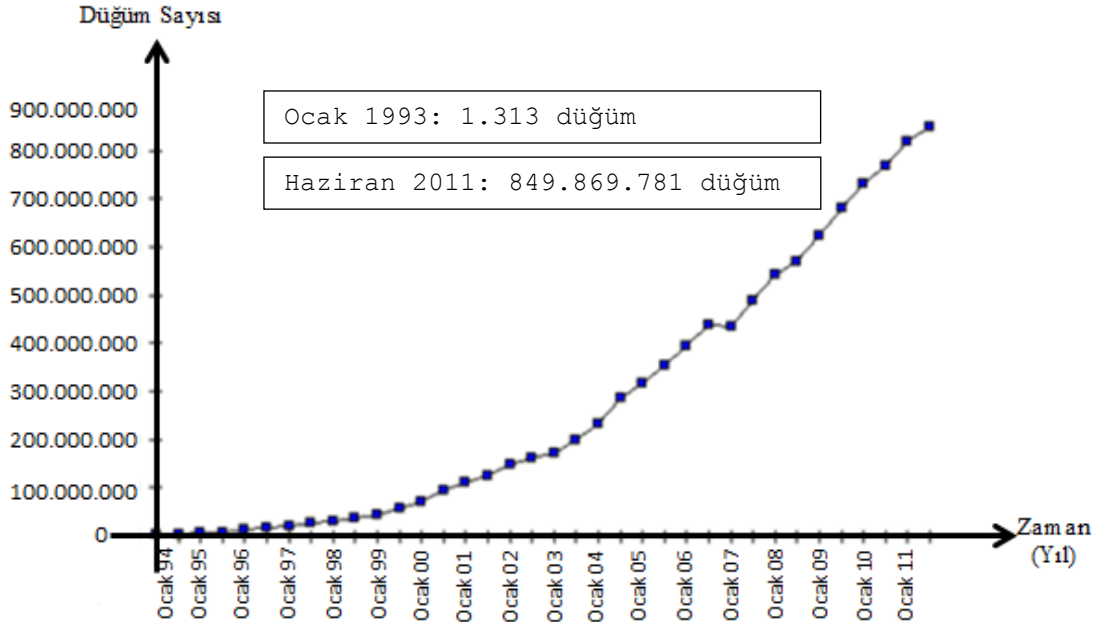
In order to show the performance of the network simulation tool, D-DEVSNET and the power of the DEVS approach on parallel and distributed applications, some tests were carried out such as the measurement of throughput capability of different sized networks in a certain period of time, the average end-to-end packet delay and the amount of packet loss. According to the tests that were conducted, D-DEVSNET has been proven a well-scalable and high performing simulation tool. In addition, it was seen that the performance of the simulation tool was not much affected by the network node / router number (increase or decrease of) and the hardware capacity.

Due to its being open sourced, parallel and distributed architecture, easy-to-follow simulation results at every step, visual quality also ability to work in a web based environment and easy installation, the developed simulation tool, D-DEVSNET was mainly aimed to use for distance education studies. The D-DEVSNET network simulation tool developed in this context has created a sample framework for distributed, scalable, adaptive and robust modeling and design of network applications. In the meantime, DEVS approach has been shown to have a wide range of application.

## BÖLÜM 1. GİRİŞ

Başlangıçta dört üniversite arasında test ağı olarak tasarlanan ilk ağ yapısı (ARPANET) [1], İnternet Sistemleri Konsorsiyumu'nun (ISC) altı ayda bir yayınlanan verilerine göre önümüzdeki birkaç yıl içerisinde bir milyar düğümlü İnternet yapısına erişecektir (Şekil 1.1) [2]. Ağların bu şekilde hızla büyümesi, ağlarda ölçeklenebilirlik konusunun yeni anlamlar kazanması yanında, yeni kavramların / konuların teknolojiye eklenmesi ile sonuçlanmıştır.

Günümüzde her yerden erişilebilirliği sağlayan küresel bir iletişim aracı olan İnternet, modern yaşam aktivitelerinin merkezinde yer almaktadır. İnternet ve teknolojilerinin gelişimi; mühendislik başta olmak üzere tüm bilim dallarını, eğitim ve ticaret iletişimini etkilemiş, bu alanda yeni bilim dallarının ve teknolojilerinin ortaya çıkmasına sebep olmuştur.



Şekil 1.1. İnternete bağlı düğüm / bilgisayar sayısı



En büyük ağ olan İnternetin baş döndürücü bir hızla büyümesi, mümkün olan en büyük ölçekte benzetim tekniklerini geliştirmeyi zorunlu kılmaktadır. Ağ benzetimi sadece ağ tasarım ve protokollerini test etmek için değil aynı zamanda dinamik ağ davranışını anlamamıza da yardımcı olabilecek en önemli araçlardan biridir.

Yapılan çalışmada, geniş ölçekli ağların benzetimi için yeni bir paralel ve dağıtık benzetim algoritması geliştirilmiş ve DEVS-Suite altında modellenmiştir. Geliştirilen benzetim aracı 'D-DEVSNET' ismi ile adlandırılmış ve yaygın İnternet protokollerinden OSPF ve BGP örnek alınarak dağıtık ve paralel bir şekilde çalıştırılmıştır. Ayrıca D-DEVSNET ortamının kapasitesini belirlemek ve DEVS yaklaşımının paralel ve dağıtık uygulamalardaki gücünü göstermek amacıyla, farklı ölçekteki ağlar incelenerek benzer ağ benzetim araçları ile kıyaslaması yapılmıştır.

### **1.1. Geniş Ölçekli Ağ Modelleme Problemleri**

Bilgisayar ağları günümüzde çok çeşitli alanlarda çeşitli amaçlarla yaygın olarak kullanılmaktadır. Kamu hizmetlerinden basit eğlence ve sohbet ağlarına kadar çeşitli sektörlerde ağ sistemleri hizmet ve faaliyet alanı olarak benimsenerek, bilgisayar ağlarını kullanmaktadır. Ağ hızında ve işlem yapma gücündeki hızlı artış gereksinimi, ağ içerisinde gerçekleştirilen yönlendirme ve yönetim işlemlerinin merkezi bir yapıdan dağıtık bir yapıya doğru kaymasını zorunlu kılmaktadır. Performans / maliyet oranının göz önünde tutulması zorunluluğu, araştırmacıları ağ üzerinde yeni işlem kapasitelerinin eklenmesi ve yeni yöntemlerin tasarlanması konusuna daha fazla yöneltmektedir [3, 4].

Ağ tasarım süreci, maliyet ve kapasite ile birlikte kullanıcı gereksinimlerinin de karşılandığı zor bir süreçtir / görevdir. Bu tasarım sürecini kolaylaştırmak için araştırmacılar ve üretici firmalar farklı ağ modelleme ve benzetim (M&S) araçları geliştirmişlerdir ve hâlihazırda geliştirmeye devam etmektedirler. Bu ağ modelleme ve benzetim araçları pratik amaçlar için kullanılabilirliği gibi eğitim ve araştırma amaçlı da kullanılmaktadırlar. Çok çeşitli sistem şartlarında, protokolleri, bunların diğer protokollerle etkileşimini araştırmak ve diğer yaklaşımlarla karşılaştırmasını

yapmak, protokollerin davranış ve özelliklerini incelemek açısından oldukça önemlidir [5].

Ağ sistemleri bu kadar geniş alanda kullanılmasına karşılık, ağ sistemlerinin ve kullanılan teknolojilerin eğitim altyapısı henüz yeterince oluşmamıştır. Eksiklik özellikle; teorik kavramların yerleşmesine kaktı sağlayacak eğitim ve araştırma araçlarının yetersizliğinden, anlaşılmasızlığından, kullanım / gelişim zorluğundan ve açık kaynak kodlu olmayışından kaynaklanmaktadır. Bunun yanında, İnternetin yakın gelecekte bir milyar düğüme erişeceği düşünülürse, ölçeklenebilirlik kavramının ağların modellenmesinde ve benzetiminde yeni boyutlar kazanacağı açıktır. Ağ araştırmalarında yaygın olarak kullanılan ns-2 [6], OPNET [7], OMNET++ [8], vb. birçok benzetim aracı küçük boyutlu ağları çalışmak için ideal platformlarken; bu yazılımların hiçbirisi geniş ölçekli sistemleri modelleme yeteneğine sahip değildir. Diğer bir deyişle, bahsedilen benzetim araçlarının hepsi, günümüzde üstel olarak artan ağ sistemlerini modellemede ve değişken yapılu ağ sistemlerinin performansını test etmede yetersiz kalmaktadırlar. Karmaşık yapıya sahip ağlar ve bunlar üzerinde çalışan dağıtık sistemleri geliştirmek mümkün olsa da, bu sistemleri modellemek ve yönetmek çok zor bir problem olarak karşımıza çıkmaktadır [3, 9].

Belirtilen kısıtlamaları / yetersizlikleri bir ölçüde karşılayabilen ve binlerce düğümü modelleyebilen GloMoSim, PDNS, vb. benzetim araçları bulunmakla birlikte, bunların çoğu dinamik, gelişebilir, yeniden boyutlanabilir ve değişik trafik şartlarına uyarlanabilir bir ağı modelleyememektedir. Diğer taraftan, bu tür benzetim araçlarının bileşenleri modüler ve hiyerarşik bir yapıda olmadığından bileşenlerin yeniden kullanımı, değişik uygulamalara uyarlanabilirliği ve hiyerarşik tasarımı zordur. Klasik benzetim araçlarının çalıştırıldıkları bilgisayarlarda oldukça yüksek kaynak kullanım gereksinimleri, çok büyük hesaplama maliyeti oluşturmaları, gelişmiş ve büyük uygulamaların meydana getirilmesini zorlaştırmaktadır [9].

Geniş ölçekli, İnternet benzeri ağ ortamlarının davranış modellerini oluşturmak için yeni tekniklerin, veri ve analiz yöntemlerinin geliştirilmesi, muhtemel kritik

hataların, yapısal zayıflıkların tespit edilebilmesi ve gerekli önlemlerin alınmasında anahtar rol oynayabilir. Geniş ölçekli ağlarda, hatalara karşı hassasiyet derecesini tespit etmek, daha sağlam yönlendirme algoritmalarını tasarlamak ve test etmek amacıyla deney yapmak günümüzde mümkün olmamakta, bütün ağ sisteminin davranışını çözmek veya hatalar karşısında ağın çökmesini önlemek gibi problemleri çözme konusunda mevcut benzetim araçları yetersiz kalmaktadır. Bu sebeple araştırmaların geniş ölçekli ağları modellemek için yeni tekniklerin geliştirilmesine odaklanması gereği açıktır [9].

Milyarlarca insanın, küresel ağ içerisinde çalışan uygulamalara günlük yaşamlarının bir parçası olarak düzenli olarak erişmesi, ağ uygulamalarının üç özelliğe sahip olması sonucunu doğurmaktadır:

- Büyük boyutlu talepleri karşılayacak bir ölçekte olması yanında heterojen yapıda (kablolu / kablosuz iletişime olanak sağlamalı) ve veri / ses / çoklu ortam trafiğine imkân verebilmelidir,
- Dinamik kullanıcı taleplerine ve ağ şartlarına kolayca uyum sağlamalıdır,
- Kısmi hatalar karşısında ayakta kalabilmeli ve kullanıcılara hizmet vermeye devam edebilmelidir.

Yukarıda yapılan açıklamalardan, ağ benzetim sistemlerinde, karmaşıklık, ölçeklenebilirlik, uyum yeteneği, beka, vb. problemler ile karşılaşılacağı sonucu çıkarılabilir. Ağ sistemlerinin esneklik, karmaşıklık, ölçeklenebilirlik, vb. problemlerinin çözülmesi amacıyla gelişmiş yönlendirme sistemlerini tasarlamak / test etmek için bileşenlerin hangi seviyede soyutlanması gerektiğini, hangi elemanlar arasında ne tür bir ilişki olduğunu belirlemek amacıyla kullanılacak gerçek dünya verisi, modelleme / benzetim desteği ve geniş ölçekli sistemleri tasarlama yaklaşımı gerekmektedir.

## 1.2. Önerilen Yöntem

Dağıtık ve geniş ölçekli bir ağ sisteminin modellenmesi işlemi;

1. Gereksinimlerin belirlenmesini,
2. Paralel ve dağıtık ayrık olay tabanlı modelleme yaklaşımına sahip bir benzetim ortamının geliştirilmesini, diğer bir ifade ile ağ bileşenlerinin, bu bileşenlerde çalışacak yazılım nesnelere ve nesnelere arasındaki etkileşimlerin tanımlanmasını,
3. Yazılım nesnelere işlem yapan farklı coğrafik alanlardaki düğümlere dağıtılmalarını; sunucu-istemci uygulama alt yapısının kurulmasını,
4. Ağ topolojileri ile yönlendirme protokollerinin entegrasyonunu,
5. Değişken ağ boyutu ve trafik şartları altında test edilmesini içermektedir.

Yapılan çalışmada ayrık olay tabanlı, modüler ve hiyerarşik bir modelleme yaklaşımı olan DEVS yaklaşımı kullanılarak sunucu / istemci tabanlı yeni bir dağıtık ve paralel mimaride çalışan bir ağ benzetim aracı geliştirilip; geliştirilen araç, 'D-DEVSNET' olarak adlandırıldı. Bu kapsamda 'DEVS-Suite [10]' modelleme ve benzetim ortamına 'BRITE [11]' topoloji üretim aracı eklendi ve böylece farklı ölçeklerdeki ağ topolojilerinin tasarımı ve izlenmesi kolaylaştırıldı.

Paralel DEVS modelleme ve benzetim yaklaşımının ve Java programlama dilinin sağladığı esneklik, paralel ve dağıtık mimari tasarımını kolaylaştırmaktadır. DEVS-Suite modelleme ve benzetim ortamının nesneye yönelik yapısı, bir ağı oluşturan düğümlerin / yönlendiricilerin, yazılım varlıklarının ve deneysel çerçevelerin modüler bir yapıda tasarımını, yeniden kullanılabilirliğini ve sistemlerin sistemini oluşturmayı kolaylaştırmaktadır [10].

Geliştirilen D-DEVSNET ağ benzetim aracı ile esnek, ölçeklenebilir geniş ölçekli bir ağ benzetim aracı geliştirilmiş ve küçük ölçekli ağlar üzerinde elde edilen benzetim sonuçları geniş ölçekli ağlar üzerinde de doğrulanmıştır.

### **1.3. Literatürde Yapılan Çalışmaların Özeti**

M.A. Rahman ve arkadaşları, analitik, benzetim, ağ keşif ve topoloji üretim araçları olmak üzere dört sınıfa ayırdıkları yaklaşık 100 adet ağ benzetim aracı üzerinde bir

inceleme çalışması yapmış ve ağ araştırmacıların doğru aracı seçmelerine yardımcı olmayı hedeflemişlerdir. Araştırmacılar bu araçlardan bazılarını test etmiş ve değişik kaynaklardan elde ettikleri bilgileri sunmuşlardır. Yapılan çalışmada, özellikle geniş ölçekli ağ benzetim araçları üzerinde yeni teknoloji ve çalışmalara olan ihtiyaç vurgulanmıştır [12].

B. P. Zeigler ve S. Mittal' in hazırladıkları raporda, çok geniş ölçekli ağların geleceği noktasından benzetim araçları incelenmiş ve geniş ölçekli ağların geliştirilmesi için modelleme ve benzetimin gerekliliği vurgulanmıştır. Mevcut modelleme ve benzetim araçlarının sorunları ve çözüm yolları belirtilerek, özellikle ağ araştırmalarında yeni yaklaşımlara olan ihtiyaç vurgulanmıştır [9].

M. Malowidzki, ns-2, J-Sim ve OPNET benzetim araçlarını, benzetim modu ve programlama ara yüzü açısından karşılaştırmıştır. Ağ benzetim aracı seçiminde benzetim modu ve programlama ara yüzünün çok da önemli bir kistas olmadığı, seçimin problemin doğasına göre yapılması gerektiği vurgulanmıştır. Buna göre; protokol analizleri ve modelleme geçerliliği testleri için küçük ölçekli, yüksek performans testleri için ise belirli bir soyutlamaya sahip geniş ölçekli ağların ideal platformlar olduğu belirtilmiştir [13].

Li ve Qian, gerçekleştirdikleri Paralel Network (ns-2 tabanlı) benzetim aracı ile benzetim araçlarının bellek gereksinimi ve benzetim zamanı dezavantajlarını çözmeyi hedeflemişlerdir. Paralel ayırık olaylı benzetim (PDES) teknikleri kullanılarak, benzetim zamanı kısaltılmıştır. Bunun için geniş ölçekli benzetim modeli paralel çalışan alt modellere ayrılmıştır [14].

L. Begg ve arkadaşları tarafından benzetim araçları üzerine hazırlanan kapsamlı bir teknik raporda, gelecek nesil ağ benzetim araçlarında servis erişilebilirliği ve esneklik çalışmalarında ayırık olaylı benzetimin önemli bir metot olacağı belirtilmiştir. En uygun benzetim araçlarının seçiminde ölçüm, servis erişilebilirliği, ağ bileşen seçimi, prototip geliştirme gibi kriterlerin değerlendirilmesinde yardımcı olacak bir çalışma yapmışlardır [15].

Taekyu Kim ve arkadaşlarının çalışmasında, DEVS ve ns-2'nun birbirlerine göre avantajlarından faydalanılarak DEVS/NS-2 isimli yeni bir benzetim aracı geliştirilmiştir. Kablosuz algılayıcı ağların modellendiği bu benzetim aracının OPNET ile karşılaştırması yapılmıştır. Ayrıca bu çalışmada geniş ölçekli ağlar için geliştirilecek benzetim ortamında, dağıtık benzetim teknoloji ve yaklaşımlarından yararlanılabileceği vurgulanmıştır [16].

K. Shaukat ve H. Sarjoughian, hazırladıkları teknik raporda ns-2 ve DEVS modelleme yaklaşımları, tasarlanan bir sıcaklık kontrol sistemi ile karşılaştırılmıştır [17].

M. Zang, 2007 yılında tamamlanan doktora tezinde DEVSJAVA ve JAVA RMI yazılım / teknolojilerini birleştirerek DEVS/RMI ismi ile yeni bir DEVS modelleme ve benzetim çerçevesi geliştirmiştir [18].

Inthira Srivrunyoo, 2007 yılında tamamlanan yüksek lisans tezinde DEVS yaklaşımının öğrenilmesi için web tabanlı bir eğitim ortamı geliştirmiş ve trafik ışıkları benzetimini gerçekleştirmiştir [19].

Lee ve arkadaşları [20] DEVS yaklaşımı ile HLA teknolojisini birlikte kullanarak, geniş ölçekli homojen trafik ağı için bir zeki ulaşım sisteminin modelleme ve benzetimini gerçekleştirmişlerdir. Zacharewicz ve arkadaşları [21] ise yine DEVS/HLA yaklaşımını kullanarak RFID tabanlı lojistik sistemlerin optimizasyonu için bir model geliştirmişlerdir.

Waupotitsch ve arkadaşları tarafından yapılan çalışmada, ulusal çapta (Los Angeles ağı) geniş ölçekli bir ağda ölçeklenebilirlik üzerinde durulmuş ve 11225 düğümlü dağıtık paket seviyeli bir ağın benzetimi gerçekleştirilmiştir [22].

Er. Monika Gupta ve arkadaşı uzak Java RMI kullanarak minimum maliyet, en kısa yol hesaplamaları yapan basit bir OSPF benzetim aracı gerçekleştirmişlerdir. Java RMI ile farklı yönlendirici ve ara yüzdeki nesnelere haberleşme sağlanmıştır [23].

Talal Mohamed Jaafar, 2007 yılında tamamlanan doktora tezinde, geniş ölçekli ağların (5476 düğüm) bellek yetersizliği problemi için EIGRP yönlendirme protokolünde misafir düğüm yaklaşımını önermiş ve çeşitli analizler yapmıştır [24].

Jerome Vernez ve arkadaşları tarafından Python dilinde yazılmış, grafiksel ara yüze sahip 'Nessi' isimli bir ağ benzetim aracı geliştirilmiştir. Eğitim amaçlı olarak tasarlanan bu benzetim aracı ile değişik protokol modellerinin tasarım ve analizleri yapılmış ve ns-2 ile karşılaştırması yapılmıştır [25].

Liu ve arkadaşları geniş ölçekli ağlarda (3.157 yönlendiriciye sahip) dağıtık uygulamaların davranışlarını incelemek üzere PRIME isimli gerçek zamanlı bir ağ benzetim aracı geliştirmişlerdir. Yaptıkları çalışmada Emulab küme bilgisayar test ortamını kullanarak web trafiği altında eş düzeyli (peer-to-peer) içerik dağıtım sistemini incelemişlerdir [26].

Jorge L. Justiniano Magraner, 2007 yılında tamamlanan yüksek lisans tezinde, Java programlama dili kullanarak eğitim amaçlı grafiksel ara yüze sahip bir ağ benzetim aracı geliştirilmiş ve OSPF protokol testleri yapılmıştır [27].

Tüncel, 2010 yılında tamamlanan doktora tezinde gezgin ağ yönlendirme protokolleri için MANET-DEVS isimli bir benzetim aracı geliştirmiştir. Dağıtık ve karmaşık dinamiklere sahip MANET'lerin karakteristikleri üzerine, ölçeklenebilir, adaptif, farklı topolojilere sahip, sağlıklı ve kolay benzetim çalışmaları yapılmasına imkân sağlayacak DEVS tabanlı bir benzetim aracının tasarımı yapılmıştır [28].

Victorina Sanz ve arkadaşları Paralel DEVS yaklaşımını kullanarak ayrık olaylı sistem modelleme için yeni bir DEVSLib isimli Modelica Kütüphanesi geliştirerek ATM makinesini (bankamatik) ve bir tank sistemini modellemişlerdir [29].

Azzedine Boukerche ve arkadaşlarının çalışmasında geniş ölçekli dağıtık benzetim ortamları için HLA/RTI özel yazılımının baskın olarak kullanıldığı vurgulanarak, genel amaçlı yeni bir DEVS tabanlı RTI tasarımı gerçekleştirilmiştir [30].

James Byrne ve arkadaşları, web tabanlı benzetim modelleri ve araçları üzerine geniş çaplı bir araştırma yaparak, klasik (masaüstü) benzetim modelleri ve araçlarına göre üstünlük ve zayıflıklarını karşılaştırmışlardır. Gelişen teknolojilerle birlikte web tabanlı benzetim ve araçlarına talebin artarak devam edeceğini belirtmişlerdir. Ayrıca web tabanlı benzetim uygulamalarının birlikte çalışabilmesi için soketler, CORBA, Java RMI gibi ara yazılımların ve dağıtık teknolojilerin önemi vurgulanmıştır [31].

#### **1.4. Tez Çalışmasının Bilime Katkısı**

Bu tez sürecinde yapılan çalışmalar aşağıdaki şekilde özetlenebilir:

- Yeni bir dağıtık ve paralel mimaride çalışan DEVS protokolü /algoritması geliştirildi.
- Geliştirilen DEVS tabanlı algoritma, geniş ölçekli ağ benzetim çalışmaları için DEVS-Suite ortamında modellendi. Modellenen ağ benzetim aracına 'D-DEVSNET' ismi verildi.
- Geliştirilen D-DEVSNET ağ benzetim aracının performansını test etmek üzere OSPF protokolü entegre edildi [32, 33].
- Otomatik olarak topoloji üretmek amacıyla BRITE topoloji üretici ağ benzetim aracına eklendi. Böylece geniş ölçekli ağların kolayca kurulup analizi sağlandı.
- Özellikle eğitim amaçlı kullanılan ağ benzetim araçları karşılaştırmalı olarak incelenerek, üstünlük ve zayıflıkları sunuldu.
- D-DEVSNET' in güvenilirliğini tesis etmek üzere diğer benzetim araçları ile (ns-2, OPNET) doğrulama ve geçerleme testlerine tabi tutuldu.
- Gerçekleştirilen son uygulamada; farklı ölçeklerde, farklı topolojilerde ve farklı ağ trafikleri altında üretilen senaryolar sonucunda elde edilen veriler ve



gözlemlenen çıkışlar grafikler halinde sunuldu ve D-DEVSNET ağ benzetim aracının geniş ölçekli dağıtık yapıdaki klasik bir ağı, modelleme ve benzetim kabiliyetine sahip olduğu gösterilmeye çalışıldı.

Bu yapılan çalışmaların bilime katkılarını ise şöyle sıralayabiliriz:

- Yeni bir ayrık olaylı paralel ve dağıtık mimaride çalışan DEVS tabanlı bir yönlendirme protokolünün modellenmesi yapılmıştır.
- Paralel ve dağıtık mimaride geniş ölçekli ve sistem teorisi tabanlı modüler ve hiyerarşik yeni bir ağ benzetim aracı (D-DEVSNET) tasarlanmıştır. Geliştirilen ağ benzetim aracının, DEVS modelleme yaklaşımına sahip ve %100 nesne yönelimli olması ile yeniden kullanılabilir bileşenlere sahip modellerin geliştirilmesine imkân sağlanmıştır.
- Geliştirilen ağ benzetim aracı ile yapılan benzetim çalışmalarının her aşamada kolay izlenmesi ve iyi bir görselliğe sahip olması sağlanmıştır. Böylece incelenen ağ sisteminin karakteristikleri hakkında bilgi sağlayacak sonuçların üretilmesi ve elde edilen sonuçların değerlendirilmesi ve yorumlanması işlemleri oldukça kolaylaşmıştır.
- Geliştirilen ağ benzetim aracı, platformdan bağımsız ve kolay kurulabilen bir altyapıya sahiptir. Ayrıca benzetim aracının İnternet üzerinden kullanımı da mümkündür. Uzaktan erişilebilen benzetim araçlarına ihtiyacın her geçen gün arttığı günümüzde gerçekleştirilen benzetim aracı, uzaktan eğitim uygulamalarında da rahatlıkla kullanılabilir. Modelleme ve benzetim de web tabanlı benzetimlerin artması ve daha da artacak olması dikkate alınırse yapılan çalışma web tabanlı geniş ölçekli ağ benzetimlerinin tasarımına da öncülük edecektir.
- Açık kaynak kodlu, modüler yapıdaki ağ benzetim aracına rahatlıkla yeni bileşenler ve araçlar eklenebilir. İlave edilecek farklı bileşenlerle, geliştirilen

benzetim aracı sadece geniş ölçekli dağıtık ağ sistemlerini değil, aynı zamanda farklı teknolojilerin (kablosuz algılayıcı ağlar gibi) modelleme ve benzetiminin yapıldığı bir ortama dönüştürülebilir.

- Geniş ölçekli ağların kolayca kurulup, analiz edilebilmesi için kod yazmadan topoloji üretebilmeyi sağlayan BRITE topoloji üretici ağ benzetim aracına eklenmiştir. BRITE topoloji üretici ile güvenilir, gerçekçi topolojilerin oluşturulması sağlanmıştır.
- Geliştirilen ağ benzetim aracı, paralel ve dağıtık çalışabilme yeteneğine sahiptir. DEVS yaklaşımı ve dağıtık istemci / sunucu mimarisi ile ağ bileşenleri kolayca coğrafik olarak farklı ağlar üzerine bölünebilir ve böylece çok geniş ölçekli ağ modellerinin benzetim çalışması yapılabilir.
- Geliştirilen ağ benzetim aracının iyi bir ölçeklenebilirliğe sahip olması hedeflenmiştir. D-DEVSNET, ağdaki düğüm sayısından (artmasından veya azalmasından) çok fazla etkilenmemektedir.

Özet olarak, geliştirilen ağ benzetim aracının, paralel ve dağıtık, web destekli çalışabilme yeteneği, platform bağımsızlığı, benzetim sonuçlarının her aşamada kolay izlenebilirliği, görselliği, kolay kurulumu gibi özellikleri ile eğitim amaçlı kullanımı hedeflenmiştir. Bu kapsamda geliştirilen D-DEVSNET benzetim aracı, dağıtık, ölçeklenebilir, adaptif, farklı topolojilere sahip geniş ölçekli ağlar için, sağlıklı ve kolay benzetim çalışmaları yapılmasına imkân sağlayarak, geniş ölçekli ağ uygulamalarının modellenmesi / tasarımı için örnek bir çerçeve oluşturmuş ve DEVS yaklaşımının geniş bir uygulama alanına sahip olduğu gösterilmiştir.

#### **1.4. Tez Düzeni**

Bölüm 1’de problemin tanımı, yapılan çalışmanın amacı, literatürdeki çalışmalar ele alınmış, yapılan tez çalışmasını diğerlerinden farklı kılan yönler, bilime katkısı ve tez düzeni hakkında bilgi verilmiştir.

Bölüm 2’de modelleme ve benzetimin tanımı ve kullanılan temel kavramlar ve bilgiler verilmiş, iyi bir modelleme benzetim çalışmaları için gerekli süreç detaylandırılarak adımlar halinde anlatılmış, modelleme ve benzetimin avantajları ve dezavantajları sıralanmıştır. DEVS modelleme ve benzetim yaklaşımı detaylı olarak anlatılmış, atomik ve birleşik DEVS kavramları klasik ve paralel DEVS başlıkları altında incelenmiştir. Ayrıca DEVS-Suite benzetim aracı tanıtılmıştır.

Bölüm 3’te dağıtık teknolojiler ve istemci / sunucu mimarisi incelenmiş, paralel ve dağıtık sistemlerin farkları açıklanmıştır. Dağıtık ağ sistemlerinin tasarım ve analizini sistematik bir şekilde gerçekleştirmek için dağıtık ağ modelleme ve benzetim süreci adım adım açıklanmıştır.

Bölüm 4’de yönlendirme algoritma ve protokollerinin sınıflandırması yapılarak incelenmiş ve RIP, OSPF ve BGP protokollerinin çalışma mantıkları detaylandırılmıştır.

Bölüm 5’de, araştırmacılara en uygun ağ benzetim aracının seçiminde yardımcı olmak gayesiyle ağ modelleme ve benzetim araçları üzerinde kısa bir inceleme yapılmış, özellikle eğitim amaçlı olarak kullanılan ağ benzetim araçlarının üstünlük ve zayıflıkları vurgulanmış ve D-DEVSNET benzetim aracı ile karşılaştırması yapılmıştır.

Bölüm 6’da geliştirilen D-DEVSNET ağ benzetim ortamını oluşturan atomik ve birleşik model bileşenleri ve ağ benzetim ortamını geliştirme süreci anlatılmıştır. OSPF protokolünün DEVS tabanlı modellenmesi ve ağdaki davranışı açıklanmıştır. Ayrıca BRUTE topoloji üretim aracı ve DEVS-Suite ortamına entegrasyonu da anlatılmıştır. Anlatılan teorik bilgiler, DEVS-Suite platformundan alınan ekran görüntüleri ile desteklenmiştir.

Bölüm 7’de, modellenen ve DEVS-Suite ortamında gerçekleştirilen dağıtık, geniş ölçekli D-DEVSNET ağ benzetim aracının doğrulama ve geçerleme testlerine yer verilmiştir. Bu test ve deneyler; geliştirilen ağ benzetim aracının kavramsal model ve

davranış geerlemesini, ns-2 ađ benzetim aracı ile benzer senaryolarda test edilmesini ve iş ıkarma yeteneklerinin karşılaştırılmasını kapsar. Deneysel ereve yoluyla elde edilen ıkışlar grafikler halinde sunulmuştur.

Bölüm 8’de istemci / sunucu taraflı, dağıtık, geniş ölekli D-DEVSNET ađ benzetim aracının test sonuçları yer almaktadır. Yapılan alışma ile farklı öleklerde, farklı topolojilerde ve farklı ađ trafikleri altında üretilen senaryolar sonucunda elde edilen veriler ve gözlemlenen ıkışlar grafikler halinde sunulmuş ve D-DEVSNET ađ benzetim aracının geniş ölekli klasik bir ađı, modelleme ve benzetim kabiliyetine sahip olduđu gösterilmeye alışılmıştır.

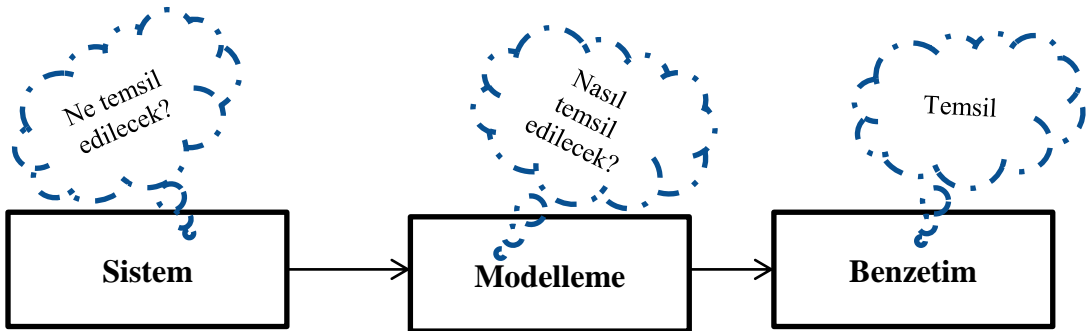
Bölüm 9’da yapılan alışmalardan elde edilen sonuçlar genel hatlarıyla özetlenmekte, tez alışmasının bilime katkıları tartışılmakta ve ileride bu alışmanın devamı olarak yapılabilecek alışmalara ışık tutabilecek önerilerde bulunmaktadır.

## BÖLÜM 2. MODELLEME VE BENZETİM TEORİSİ

### 2.1. Giriş

Benzetim, gerçek bir dünya süreci veya sisteminin işletilmesinin zamana bağlı olarak taklit edilmesidir [34]. Şekil 2.1'deki akış şemasında da görüleceği üzere benzetim, söz konusu sistemi temsil edecek bir modeli içerir. Dolayısıyla modelleme ve benzetim, temsil edilecek / tasarlanacak sistemin, gerçek hayatta gerçekleştirilmesinin tehlikeli, pahalı ya da zaman gerektiren deneylerinin, inceleme ve çalışmalarının gerçekleştirilmesine / yürütülmesine imkân sağlayan vazgeçilmez bir araçtır.

Bir başka ifade ile benzetim; teorik ya da gerçek bir sistemin modelinin tasarımı, bu model kullanılarak sistemin işletilmesi ve sistemin davranışının izlenmesi sürecidir. Benzetim işleminin sayısal ortamda bilgisayarlarla gerçekleştirilmesi, 'bilgisayar benzetimi (simülasyon)' olarak adlandırılır [35]. Günümüzde, taklit edilen gerçek sistemler (uçuş simülatorü, tank simülatorü, deprem simülatorü gibi) genelde bilgisayar yardımıyla modellenmektedir.



Şekil 2.1. Sistem, modelleme ve benzetim ilişkisi

Farklı bilim dallarındaki modelleme ve benzetim çalışmalarında kullanılan birçok kavramın (model tanımlama, basitleştirme, geçerleme, benzetim, vb.) birbirine benzerliği, modelleme ve benzetim olgularının ayrı bir disiplin olarak ele alınmasına sebep olmuştur [36].

Bu bölümde herhangi bir uygulama alanından bağımsız olarak modelleme ve benzetim teorisine ilişkin temel kavramlar verildikten sonra bir ayrık olaylı benzetim yaklaşımı olan DEVS' ten bahsedilecektir.

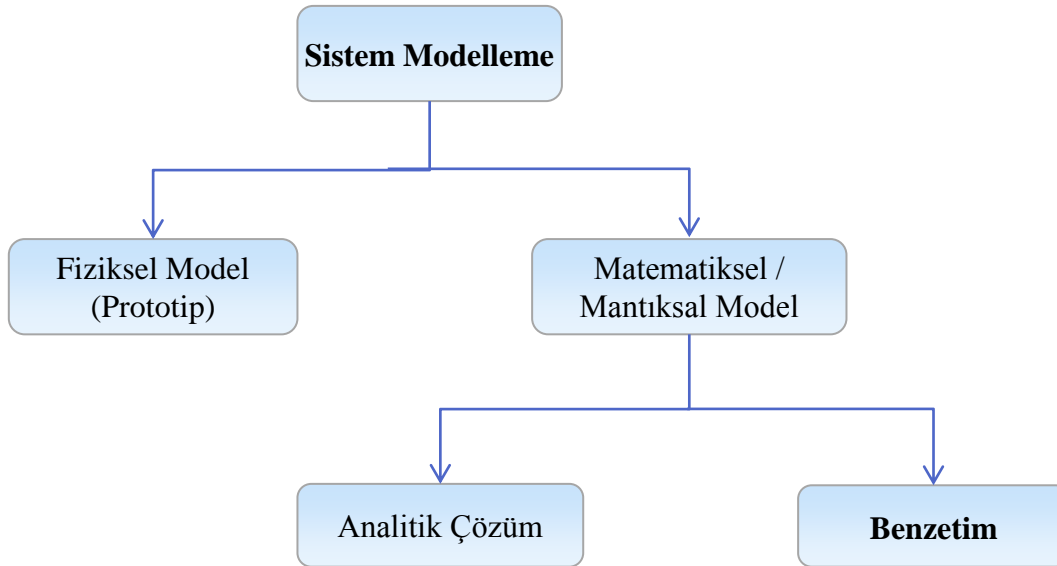
## 2.2. Modelleme ve Benzetim

Modelleme, gerçek dünya nesnesinin yaklaşık olarak ifade edilmesidir. Modelden beklenen, gerçek sistemden alınacak sonuçlara benzer sonuçlar vermesidir. Modelleme, bir sistemi incelemek üzere o sistemin basit bir örneğinin yapılması anlamına gelir. Bu örnek gerçek sistemin yardımcısı ve basitleştirilmiş bir şeklidir. Fakat modelin de gerçek sistemden alınacak sonuçlara benzer sonuçlar verecek kadar detaylı olması beklenir [28]. Bir sistemi modellemek için üç farklı yöntem kullanılır(Şekil 2.2):

1. Fiziksel modelini ya da ilk örneğini (prototipini) oluşturma,
2. Matematiksel analitik çözüm,
3. Benzetim (Simülasyon)

Fiziksel model yönteminde; sistemin performansı, mevcut sistem ya da sistemin prototipi değişik koşullar altında incelenerek elde edilebilir. Fiziksel model, en güvenilir ve en doğru yöntem olmasına rağmen, özellikle karmaşık iletişim sistemleri için planlama ve tasarım aşamaları gibi çeşitli konfigürasyonların denenmesinin zorunlu olduğu birçok durumda gerçekleştirilmesi oldukça zordur. Prototip ya da gerçek bir sistemle çalışma, pratik olmayan yüksek maliyet ve uzun zaman gerektiren bir yöntemdir [35, 37, 38].

Analitik çözüm, diğer yöntemler arasında basitlik, ekonomiklik gibi avantajlara sahiptir ve genellikle basitleştirilmiş varsayımlar ve ideal kabuller üzerine kurulur. Bu yüzden kesin sonuçlar istendiğinde analitik modeli oluşturmak, karmaşıklık ve zaman tüketimi açısından sistemin prototipini oluşturmak kadar zordur [38].



Şekil 2.2. Modelleme yöntemleri

Benzetim yöntemi, gerçek bir sürecin veya sistemin işletilmesinin zamana bağlı olarak taklit edilmesidir. Diğer bir ifade ile benzetim; bir sisteme ait sebep - sonuç ilişkilerinin bilgisayar ortamına aktarılarak sistemin davranışlarının bilgisayarda izlenmesini sağlayan bir modelleme tekniğidir [39]. Benzetim ister elle, isterse bilgisayar ile yapılsın, bir sistemin yapay kayıtlarının oluşturulması ve gerçek sistemin işletim karakteristikleriyle ilgili sonuçlarının elde edilmesinde bu yapay kaydın incelenmesini kapsamaktadır. Benzetim genellikle başka testlerle gerçekleştirmediğimiz karmaşık bir sistemin davranışını tahmin etmede kullanılan bir yöntemdir [34].

Benzetim; gerçek sistemin modelinin tasarımı ve bu model ile sistemin işletilmesi amacına yönelik olarak, sistemin davranışını anlayabilmek veya değişik stratejileri değerlendirebilmek (ölçütler kümesinin verdiği sınırlar içinde) için deneyler yürütme sürecidir. Artık bilimsel araştırmalarda teorik ve deneysel yaklaşımları tamamlayıcı

nitelikte benzetim yöntemlerinin kullanımı bir zorunluluk halini almıştır ve her geçen gün daha karmaşık ve daha detaylı benzetimlere ihtiyaç duyulmaktadır [5, 40].

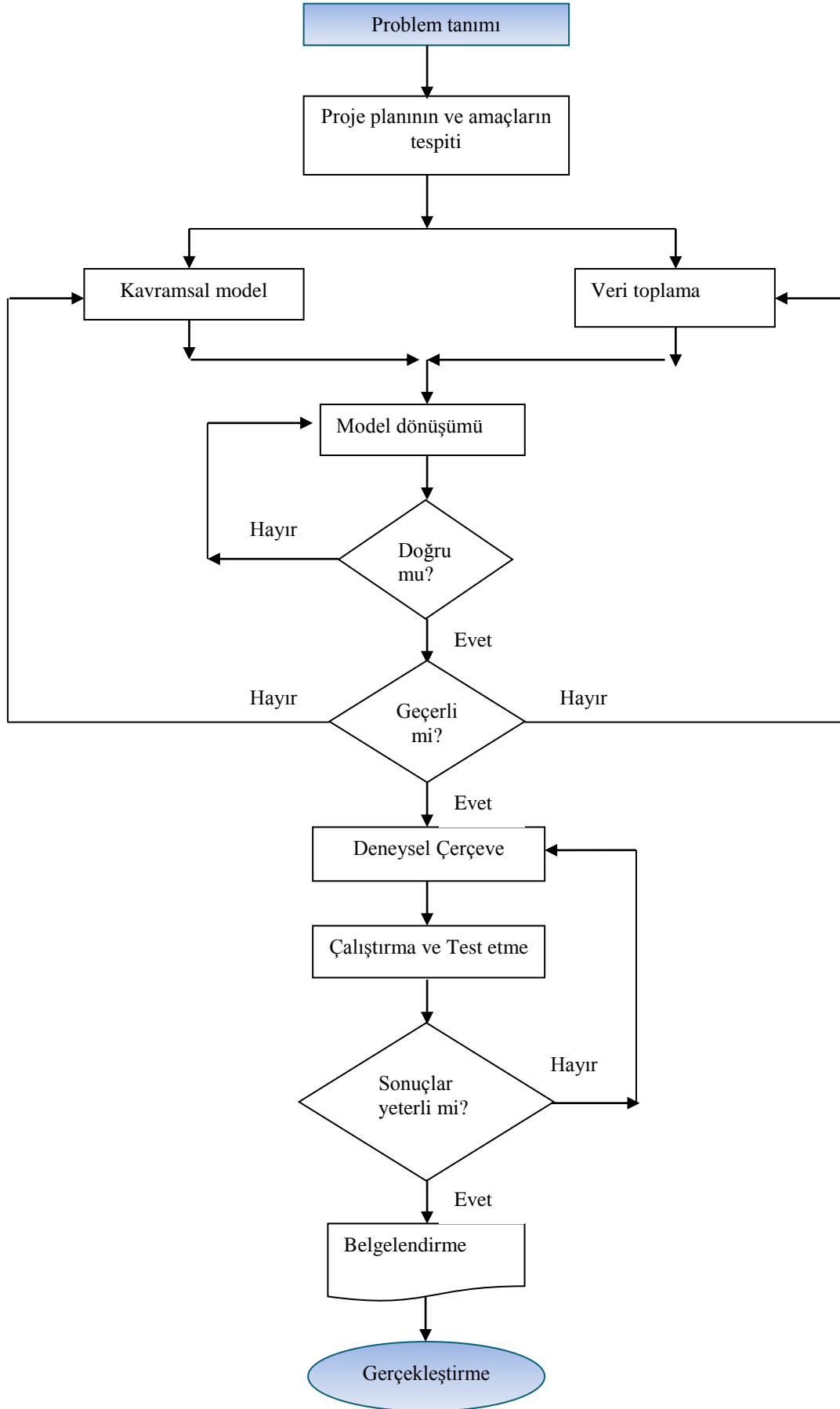
Şekil 2.3'te bir benzetim çalışmasının işlem basamakları gösterilmektedir. İşlem basamaklarını, Mackulak [41], Shannon [42], Law [43] ve Banks [34]'ın bu konudaki çalışmalarından faydalanarak kısaca açıklayacağız.

1. Problem tanımı: Her program veya her benzetim aracı, bir probleme çözüm üretmek amacıyla geliştirilir. Burada önemli olan nokta, problemin açık ve seçik bir şekilde ifade edilmesidir. Problemi bu nedenle, 'X sisteminin performans çalışması' yerine 'A, B, C faktörleri altında X sisteminin M metriğinin performans çalışması' şeklinde tanımlamak daha doğru olacaktır.
2. Tüm proje planının ve amaçların tespiti: Benzetim çalışmasının hangi sorulara ne tür cevap[lar] verdiği vurgulanır. Bu noktada projenin her aşamasının sonunda beklenen çıktı, maliyet ve süre gibi tüm proje planı detaylandırılır.
3. Kavramsal Model: Geniş ölçekli sistemlerin karmaşıklık düzeyleri yüksek olmaktadır. Bu nedenle önce sistemin basit bir modeli oluşturulur daha sonra detaylandırılır.

Kavramsal model, gerçek dünya probleminin ilk soyutlamasıdır ve sistem nesnelere ve aralarındaki ilişkilerin açıklamalarını içerir. Sistemin ana bileşenleri ve bu bileşenlere ilişkin temel özellikler ile bileşenler arasındaki ilişki ve etkileşimler ortaya konularak basit model (abstract model) ve bundan sonra her bir bileşenin ayrıntılı modeli oluşturulur. Bu çerçevede modellemenin ilk basamağı olan basit model, bu aşamada ortaya çıkarılır. Kavramsal modellemede sistemin ön plana çıkan bileşenleri ile bunlara ilişkin temel işlevsel, yapısal, davranışsal ve etkileşimsel özellikleri tanımlanmakta ve böylece sistemin âdeta bir taslak çizimi oluşturulmaktadır [36].



4. Veri Toplama: Benzetim için gerekli olan giriş verileri toplanır. Sistemin ayrıntılı verileri (bant genişliği, vb.), var olan sistemin başarımı, giriş verileri ve sistemin kaynakları tanımlanır. Olası her bir giriş için uygun olasılık dağılımı ve ilgili parametreler seçilir. Pek çok benzetim yazılım paketi birçok özellik yanında rastgele sayı üreteçleri ve uygun dağılım türleri içermektedir. Deneysel dağılımlar, standart dağılımlar veya mevcut sistem verisi uygun olmadığında kullanılırlar [28]. Şekil 2.3'ten de anlaşılacağı üzere verilerin toplanma aşaması devam ederken model dönüşümü yapılabilir.
5. Model dönüşümü: Üçüncü adımda açıklanan kavramsal model bu adımda bir program vasıtası ile bilgisayar ortamında işlenebilecek formata / koda dönüştürülür.
6. Doğrulama: Doğrulama, yazılan program kodunun kavramsal modelde tanımlanan davranışı tam olarak yansıttığı kabulü ile soyut tanımlamadan program koduna geçişin doğruluğu ile ilgilendir. Bu aşamada giriş verilerinin ve model yapısının mantıksal hata içerip içermediği test edilerek model doğrulanır.
7. Geçerleme: Geçerleme, modelin kullanım amacı dikkate alınarak ürettiği verilerin modellenen sistemle ve ölçülmüş verilerle olan benzerlik derecesinin belirlenmesi işlemidir. Benzetim çalışmasının en önemli adımlarından biridir. Geçerlemede, tasarlanan modelin modellediği sistemle olan benzerlik derecesi ölçülürken doğrulamada, oluşturulan modelin, modeli oluşturan tasarımcının öngördüğü kavramsal tanımlamalar ve gereksinimler ile hangi düzeyde benzeştiği ölçülmektedir. Doğrulanmış ve geçerlenmiş modeller ile elde edilen sonuçlar güvenilir addedilir. Modelin geçerliliği / doğruluğu kabul edilene kadar işlemler tekrar edilir [4, 14].
8. Deneysel Çerçeve: Gerçek dünya nesnesini temsil eden sistemle, sisteme karşılık gelen modellerin çalıştırılacağı deneysel şartları / ortamı temsil eder. Bir başka ifade ile bir modelin deney yapılacağı ve gözlemleneceği şartları tanımlar. Tipik bir deneysel çerçeve bir üreteç ve dönüştürücüden oluşur [3, 39].



Şekil 2.3. Benzetim çalışmasının adımları [34, 41]

9. Çalıştırma ve test etme: Farklı benzetim senaryoları için programın çalıştırıldığı ve sonuçlar üzerinde analizlerin yapıldığı aşamadır. Uygun etkiye sahip giriş değişkenleri seçilerek testler yapılır. Örnek sayısının testlere güvenilebilecek çoklukta olması önemlidir.
10. Belgelendirme: Belgelendirme, programı kullanacak kişi ve geliştiriciler için bir ‘kullanıcı elkitabı’ biçiminde hazırlanır. Problemin tanımı, çözüm yöntemi, giriş parametreleri ve buna bağlı çıkış değerleri, sonraki çalışmalar için öneriler, kaynak program listesi, uygulama koşulları ve kullanım amaçları ayrıntılı olarak yer alır.
11. Gerçekleştirme: Başarılı bir gerçekleştirim, süreçteki tüm adımların başarılı olmasına bağlıdır.

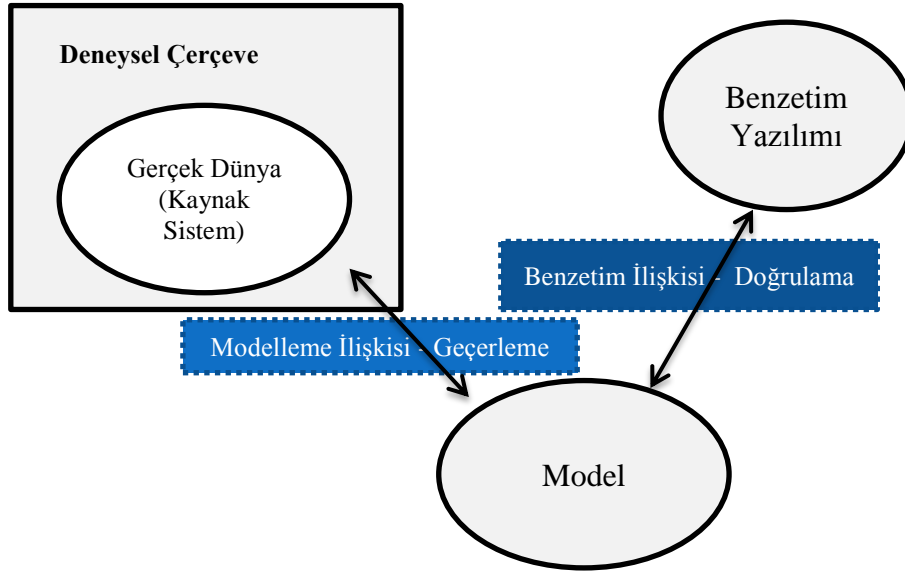
Benzetim çalışmasındaki bu işlem basamaklarının tamamı her zaman mümkün veya gerekli olmayabileceği gibi, burada bahsedilmeyen ek adımlar da gerekebilir.

Bilgisayar ağı araştırmalarında modelleme ve benzetim yönteminin, yeni ağ teknolojilerinin etkin bir şekilde geliştirilmesi ve test edilmesi, değişik ağ koşulları ve senaryoları altında iletişim protokollerinin geliştirilmesinde ve değerlendirilmesinde önemli bir rolü bulunmaktadır [44]. Bir sistemi modellemek için kullanılan yöntemler karşılaştırıldığında bilgisayar ağlarının başarımlarında analizinde olay tabanlı (event-driven) bilgisayar benzetimi en iyi çözüm olarak görülmektedir [38].

### **2.3. Modelleme ve Benzetim Çerçevesi**

Modelleme ve benzetimin amacı, gerçek dünyada bulunan ya da bulunması öngörülen sistem veya sistemlerin benzerlerinin oluşturulması ve işletilmesidir. Modelleme ve benzetim (M&S) çerçevesi ise varlıkları ve onların ilişkilerini tanımlar [3, 45]. Şekil 2.4'ten görüleceği üzere temel varlıklar; gerçek dünya (modellenecek kaynak sistem- örneğin İnternet ağı), model (gerçek dünya

davranışını üreten yapının matematiksel modeli), benzetim aracı (modeli işleten araç-örneğin bilgisayar) ve deneysel çerçeve ile bunların birbirleri ile olan ilişkilerinden oluşmaktadır. Bu ilişkiler Tablo 2.1’ de kısaca açıklanmıştır



Şekil 2.4. Modelleme ve Benzetim (M&S) Çerçevesi: Varlıklar ve İlişkileri

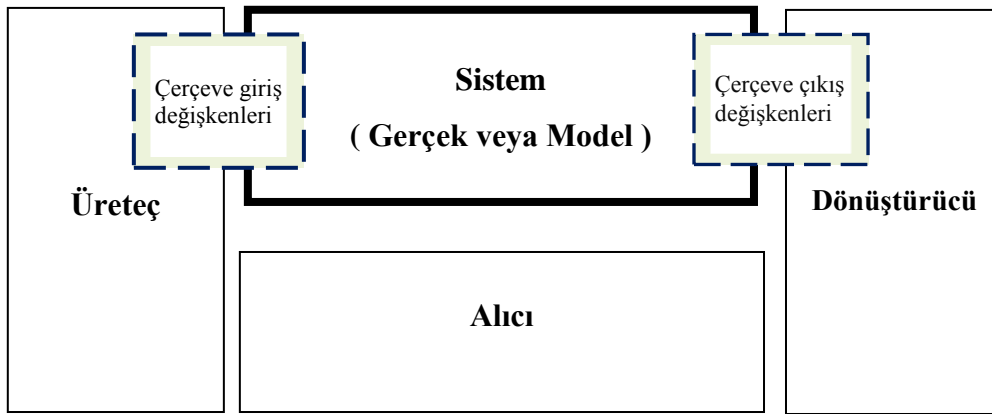
Tablo 2.1. M & S şemasındaki temel varlıkların açıklanması.

Temel Varlıklar	Açıklama
<b>Kaynak Sistem</b>	Modellemeyi düşündüğümüz gerçek veya sanal ortamı belirtir.
<b>Deneysel Çerçeve</b>	Deney yapılan veya gözlemlenen sistemin belirli koşullar altındaki tanımlamasıdır. Diğer bir ifade ile deneysel çerçeve, bir benzetim aracı ile model üzerinde deneyler yapan kişinin hedeflerini yansıtır.
<b>Model</b>	Gerçek dünya davranışını üreten yapının matematiksel modelini tanımlayan komutlar kümesidir. Bir sistemin yapısının ve / veya davranışının belirgin özellikleri belirli bir doğruluk dâhilinde model tarafından yansıtılmalıdır.
<b>Doğrulama ve Geçerleme</b>	Doğrulama, benzetim programının tutarlılığını türetildiği modele göre kontrol işlemidir. Geçerleme ise benzetim ortamında elde edilen sonuçlarla belirlenmiş deneysel çerçeve ortamında elde edilen sonuçların karşılaştırılması işlemidir [46].
<b>Benzetim Yazılımı</b>	Modelin işletilmesi / çalıştırılması aşamasında kullanılan benzetim aracıdır.

### 2.3.1. Deneysel çerçeve

Deneysel çerçeve (Experimental Frame - EF) deney yapılan veya gözlemlenen sistemin belirli koşullar altındaki çalışmasını tarif eder. Başka bir ifade ile deneysel çerçeve, gerçek bir sistem veya bir benzetim aracı ile model üzerinde deneyler yapan kişinin / modelleyicinin hedeflerini yansıtır.

Şekil 2.5'te gösterildiği gibi bir deneysel çerçeve, sistemin / modelin giriş ve çıkış uçlarına bağlanan çerçeve giriş / çıkış değişkenleri, üreteç, dönüştürücü ve alıcı bileşenlerinden oluşmaktadır. Bu bileşenleri işlevleri bakımından bir elektronik devre deneyindeki giriş / çıkış bağlantıları, sinyal üretici ve osiloskoba benzetilebiliriz.



Şekil 2.5. Deneysel çerçevenin yapısı

Üreteç deney esnasında sistemi / modeli uyarmak için, dönüştürücü ise sistemden gelen sonuçlardan mantıklı bir yorum yapmak için sisteme / modele uygulanacak dönüşümleri tanımlar. Üreteç ve dönüştürücü, giriş / çıkış değişkenleri ile birlikte deneysel çerçeve içindeki üreteç girişleri ile dönüştürücü çıkışlarını karşılaştıran bir alıcıdan oluşur. Alıcı, sistemin (gerçek veya model) deneysel çerçevesinin deney yapan kişinin hedefleriyle uyumlu olup-olmadığını belirler [39].

### 2.4. Modelleme ve Benzetimin Olumlu ve Olumsuz Yönleri

Modelleme ve benzetimin olumlu yönlerini aşağıdaki gibi sıralayabiliriz [47].

- Üzerinde çalışılan sistemin matematiksel bir modeli geliştirilerek sistemin daha anlaşılır olması sağlanır ve çok uzun zaman dilimlerinde sistemin çalışması gözlemlenebilir,
- Fizibilite için sistem hakkındaki hipotezlerin testine imkân sağlar,
- Detaylardaki karmaşık olguları gözlemlemek için zamanı sıkıştırmaya veya genişletmeye imkân sağlar,
- Sistemin modeli değiştirilerek sistemin uygulanmasındaki etkiler gözlemlenebilir. Gerçek sistem ile yapılacak uygulamaların önemli risklerini gerçek sistemi bozmadan azaltır,
- Yetersiz bilginin olduğu yeni veya bilinmeyen durumlar hakkında test imkânı sağlar,
- Başarım ölçümlerinin en hassas olduğu sürücü değişkenleri ve birbirleri ile olan ilişkilerini saptar,
- Varlıkların veya bilginin akışındaki tıkanıklıkların belirlenmesini (materyal, insan vb.) kolaylaştırır,
- Sistem yapılandırması için çoklu başarım ölçümleri kullanılabilir,
- Problem çözümüne sistematik yaklaşım sağlar,
- Kısa sürede iyi ve sağlam bir sistem geliştirilmesini sağlar.

Modelleme ve benzetimin olumsuz yönlerini ise aşağıdaki gibi sıralayabiliriz;

- Benzetimin zaman alan karmaşık alıştırmalar içermesi,
- Analitik çözümün mümkün olduğu ortamlarda benzetimin kullanılması,
- Hatalı model tasarımı,
- Benzetim modelinin fazla karmaşık veya basit olması,
- Hatalı varsayımlar yapılması,
- Kayda geçirilmemiş varsayımların ortaya çıkması, yeterli belgelendirmenin yapılmaması,
- Giriş verilerinde yanlış olasılık dağılımı,
- Başarımın yanlış ölçülmesi,
- Benzetim programının mantıksal hata içermesi,
- Çıkış verisinde başlangıçtaki önyargı,
- Konfigürasyon için tek benzetim çalıştırma,

- Eksik çalışma takvimi veya bütçe planlaması,
- Benzetim çalışması ile ilgisi olan personel arasındaki iletişimsizlik.

## 2.5. Modelleme Yöntemleri

Modelleme açısından bakıldığında, tüm sistemler amaçlarına göre farklılaşan bir soyutlama seviyesinde ele alınıp, incelenebilirler. Modelleme işlemlerinde; çözünürlük, soyutlama seviyesi ve deneysel çerçevenin kapsamı referans alınarak, bir sistem en karmaşıktan en basite / soyuta doğru sürekli, ayrık zamanlı ve ayrık olaylı olarak incelenebilir.

Sürekli sistem modellerini işletebilen benzetim araçları, diferansiyel denklem çözümlerini yapabilen entegre devrelerden oluşurken; ayrık zamanlı sistem modellerini işletebilen benzetim araçları, fark denklemlerini çözebilen özyinelemeli yapıdaki algoritmalarından oluşmaktadır.

Ayrık olaylı sistem modellerini işletebilen benzetim araçları, olay denklemlerinin yapısına göre düzenlenen ve zaman boyutunda durum çözümlemesi yapabilen algoritmalarından oluşmaktadır. Modelleme işleminde amaç sistem davranışını en düşük çözünürlükte ve soyutlama seviyesinde incelemek olması durumunda ayrık olaylı modelleme yaklaşımı kullanılır [36].

### 2.5.1. Ayrık olaylı modelleme

Zaman ekseninin sürekli olduğu, ancak sadece sınırlı bir zaman periyodunda sonlu sayıda olayların meydana geldiği sistemler, ‘ayrık olaylı sistemler’ olarak adlandırılır ve yüksek bir soyutlama seviyesinde ele alınırlar.

Ayrık olaylı sistemlerde, sadece eş zamanlı olmayan ayrık olayların anlık durum geçişlerinde sistemin durumu değişirken, olaylar arasında durum değişikliği olmaz. Bu sebeple ‘ayrık olaylı modeller’, sistem durumunun zaman üzerinde sürekli değiştiği sürekli modellerden ayrılırlar. Olaylar arasında sistemin durumunun

değişmemesi sebebi ile ayrıık olaylı benzetim araçlarında olay olmayan zamanlarda sistemin durumu işlenmez. Bu da, durum bilgisinin sürekli bir zaman ekseninde her noktada gösteriminin gerektiği benzetimler ile karşılaştırıldığında, daha verimli bir benzetime sebep olur [28, 39].

Ayrıık olaylı modelleme yaklaşımları yüksek bir soyutlama düzeyinde ele alınır. Soyutlama, belirli bir amaç için gerekli olan bilgi veya verilerin daha rahat elde edilebilmesi için bir kavram veya sistem davranışının bilgi içeriğini azaltma veya basitleştirme sürecidir. Soyutlama, basit ilişkiler tanımlamak için önemsiz ayrıntıların yok edildiği, modelleyicinin gerçek dünyadaki sistem davranışının belirleyici özelliklerini alıp, farklı bir yapıda temsil edilmesini sağlayan bir tekniktir.

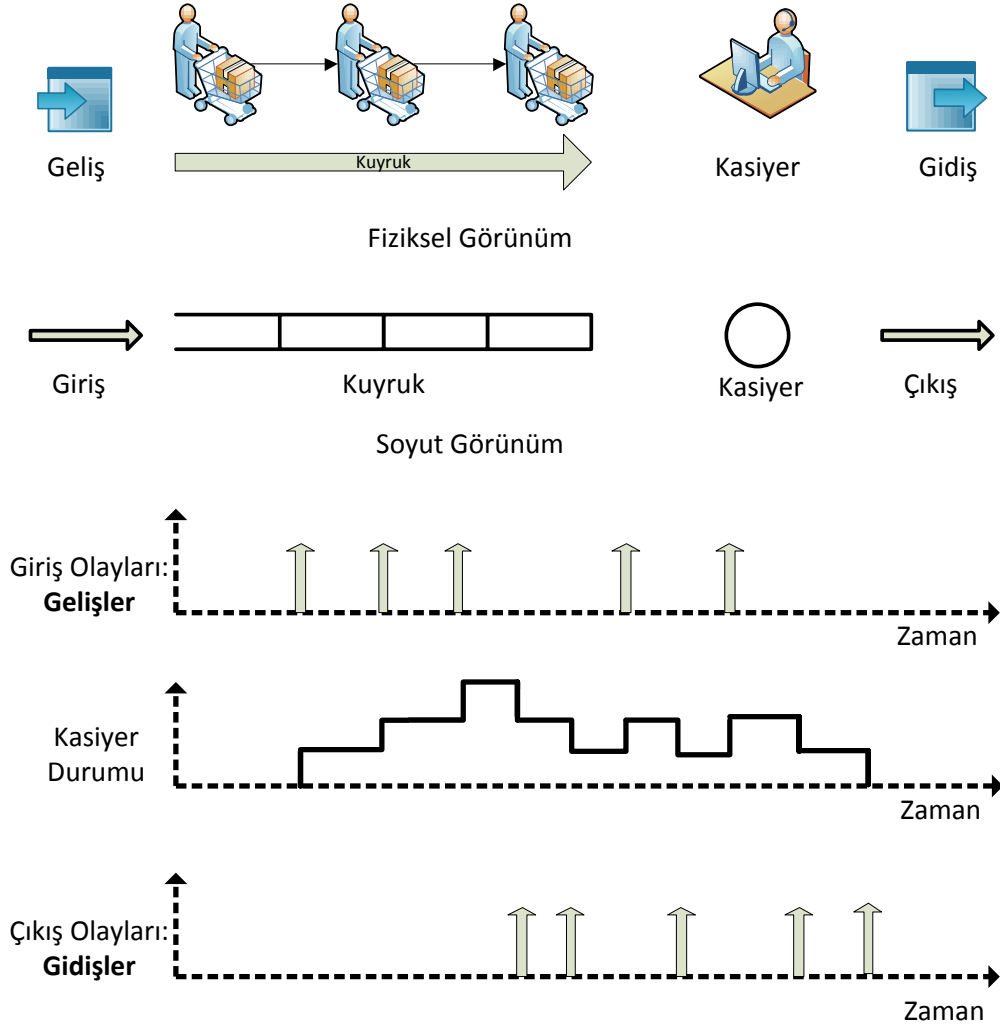
Yüksek bir soyutlama düzeyi, gerçek dünya davranışıyla ilgisi olmayan benzetim araçlarını getirebilir. Özellikle birden fazla olayın aynı zaman diliminde meydana geldiği olay çakışmaları, ayrıık olaylı modelin yeteri kadar detaylandırılmamasından kaynaklanabilir. Yeni ayrıık olay yöntemleri, oluşan olay çakışmalarını başarılı bir şekilde yönetirler ve bu durumların ortadan kaldırılması konusunda güvenilir çözüm yolları sunarlar [39].

Ayrıık olaylı modellemeye örnek olarak bir kasiyer / kuyruk sisteminin davranışını gösterebiliriz. Şekil 2.6'dan da görüleceği üzere fiziksel seviyede kasiyer aynı anda sadece bir müşteriye hizmet vermektedir. Kasiyerin meşgul olduğu durumda müşteriler kuyrukta beklemektedir. Burada sistem durumu müşterilerin özelliklerine bakmaksızın meydana gelir (yaş, alınan ürün adedi, vb.). Böylece model kuyruk durumu doğal sayılardan oluşan bir kuyruk uzunluğu manasına gelir. Kasiyer 'boş veya 'meşgul' durumunda olabilir. Sistem dinamikleri kuyruk ve kasiyer durumları ile kararlaştırılır. Kuyruk yönetimi, 'ilk giren ilk çıkar (First In First Out - FIFO)' yöntemine göre ele alınır [48].

Ayrıık olaylı modelleme ve benzetim stratejisinde model; her bir olay için olayın durum üzerindeki ve sistemin gelecekteki davranışı üzerindeki etkisini tanımlar. Bu, yeni olayların gelecekteki bir zamana programlanmasıyla yapılır. Bütün olaylar



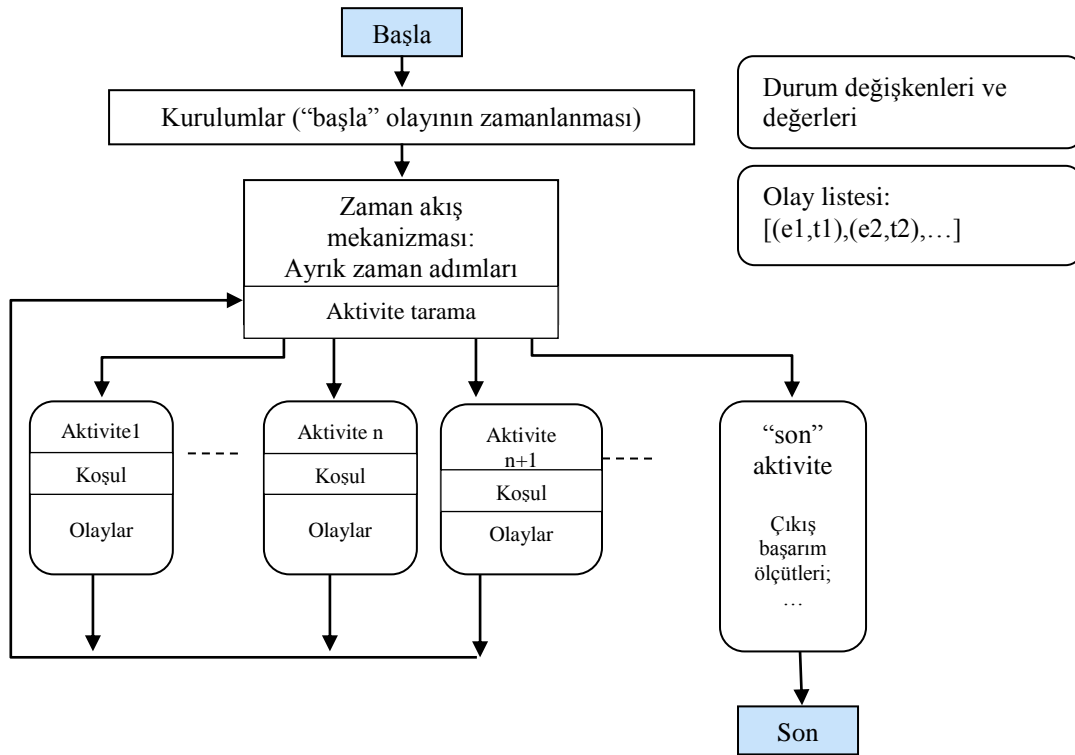
önceden planlanır. Bir olayın planlanması, olayın gerçekleşmesi için gerekli bütün şartların önceden bilindiği durumlarda mantıklı olur [39].



Şekil 2.6. Ayrık olaylı sistem modelleme örneği: kasiyer / kuyruk sistemi

Bir ayrık olaylı benzetim modeli, benzetim zamanını ilerleten, olay listesini ve sistem durumunu güncelleyen tekrarlamalı benzetim işlemleri ile gösterilebilir. Şekil 2.7’de görüldüğü gibi bir ayrık olaylı benzetim çekirdeği iki adet veri yapısı kullanır. Veri yapılarından bir tanesi modelde bildirilen durum değişkenlerini, diğeri ise artan zamanla ve azalan öncelikte, sıralı bir olay listesinde zamanlanmış olay bildirimlerini içerir. Bir olay programlandığı zaman listede en sona eklenir. Öncelikler aynı zamanda gerçekleşen (çakışan) olaylar arasında seçim yapmak için kullanılır. Bir ayrık olaylı benzetim çekirdeği, olayları artan zaman adımlarına göre ‘olay listesi’

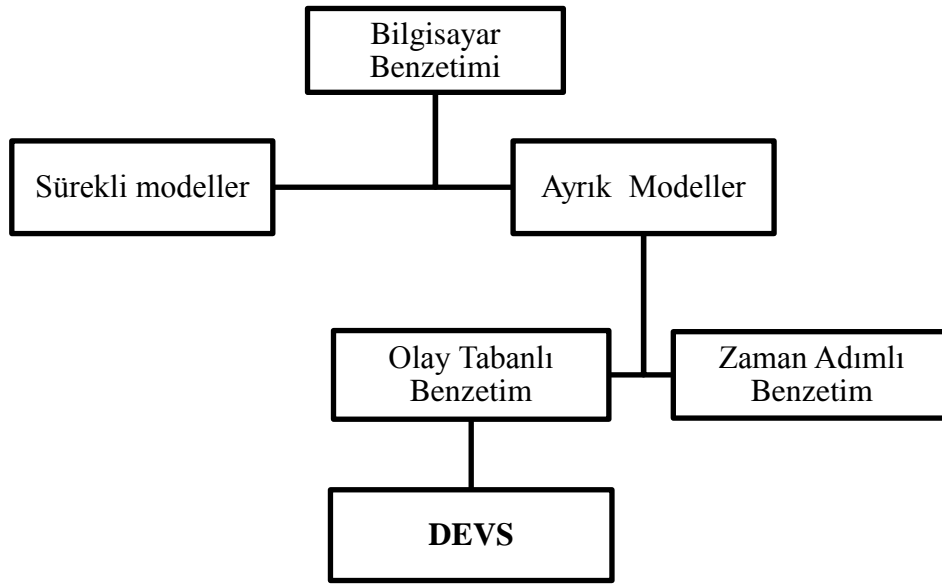
şeklinde düzenler ve liste tamamen boşalana kadar listenin en başındakini alıp işler. Olay bildirimindeki olay zamanı, benzetim zamanını ilerletmek için kullanılır. Olay bildirimindeki olay türüne bağlı olarak, uygun olay bildirimini yapılabilir. Bu işlem, olay bildirimlerini olay listesine ekleyerek sistemin durumunu değiştirebilir ve gelecekte yeni olayları zamanlayabilir. Olay, durum ve zaman tanımlamaları için DEVS modelleme ve benzetim yaklaşımı önerilmektedir [36, 48].



Şekil 2.7. Ayrık olaylı benzetim çekirdeği [48].

## 2.6. DEVS Modelleme ve Benzetim (M&S) Yaklaşımı

Ayrık Olaylı Sistem Tanımı (Discrete Event System Specification – DEVS ) formalizmi / yaklaşımı, sistem olarak adlandırılan bir matematiksel nesneyi tanımlama aracıdır. DEVS yaklaşımı, ayrık olaylı sistemlerin modellenmesi ve analizi için ilk olarak Dr. Bernard P. Zeigler tarafından 1976 yılında 'Theory of Modeling and Simulation' isimli kitabı ile ortaya atılmıştır [44].



Şekil 2.8. Benzetim sınıflandırması içerisinde DEVS yönteminin yeri

Bilgisayar benzetimi, sürekli ve ayrık modeller olarak ikiye ayrılır, ayrık modelleri de olay tabanlı ve zaman adımlı olmak üzere ikiye ayırabiliriz (Şekil 2.8). DEVS, ayrık olay tabanlı, modüler ve hiyerarşik bir benzetim yaklaşımıdır ve son zamanlarda diğer yaklaşımlardan daha fazla öne çıkmaktadır [49, 50].

DEVS yaklaşımının sadece ayrık olaylı modeller için değil, aynı zamanda ayrık zamanlı ve diferansiyel denklemlerle ifade edilen davranışları uyarlamak için de bir hesaplama temeli oluşturması, modelleme ve benzetim aktivitelerinde DEVS yaklaşımının yaygın olarak kullanılmasını sağlamaktadır.

Ayrık olaylı yaklaşımlar sınıfı içinde modeller, zamanın sürekli olduğu ancak sınırlı bir zaman periyodunda sonlu sayıda olayın meydana geldiği bir soyutlama seviyesinde tanımlanır. Bu olaylar, sistemin durum değiştirmesine neden olabilir. Olaylar arasında sistemin durumu kesinlikle değişmez. Sistemin durumunun zaman içerisinde sürekli değiştiği sürekli modellerden bu noktada farklıdır [39].

DEVS yaklaşımı, sistemleri bileşenlerin birbirine bağlanması yoluyla tanımlar ve iletişim portlarını kullanarak bileşenler arasında karşılıklı etkileşimi mümkün kılar.

DEVS M&S yaklaşımının avantajlarını şöyle sıralayabiliriz;

- Bileşenler arası sağlam bağlantı,
- Modüler ve hiyerarşik tasarım,
- Olay tabanlı ve yüksek performanslı benzetim,
- Nesne yönelimli uygulama geliştirme,
- Düşük tasarım zamanı,
- Gelişmiş test altyapısı ve daha kaliteli model tasarımı,
- Paralel ve dağıtık çalışabilme yeteneği,
- Doğrulama ve geçерleme kolaylığı,
- Birlikte çalışabilirlik ve yeniden kullanılabilirlik.

Bu avantajlar dikkate alınarak DEVS yaklaşımı ile aşağıdaki problemler çözülebilir;

- Melez sistemlerin tanımlanması ve modellenmesi,
- Birlikte çalışabilirlik ve yeniden kullanım,
- Otomatik görevler yerine getiren araçlar,
- Yüksek performanslı paralel ve dağıtık benzetim.

DEVS formalizminin ADEVS [51], DEVS-Suite [10], DEVS/C++ [52], DEVSJAVA [53], JAMES II [54] ve SmallDEVS [55] olmak üzere birçok uyarlaması mevcuttur. Bu genel DEVS ortamlarının yanında; sürekli ve ayrık olaylı sistemler için DESS/DEVS, paralel DEVS için P-DEVS, gerçek zamanlı sistemler için RT-DEVS, hücresel sistemler için Cell-DEVS, fuzzy (bulanık) sistemler için Fuzzy-DEVS gibi belli bir amaca yönelik DEVS uyarlamaları da mevcuttur [9, 56].

### **2.6.1. Klasik DEVS yaklaşımı**

Klasik DEVS yaklaşımında, sistem davranışı atomik DEVS ve birleşik / tümleşik DEVS olmak üzere iki farklı seviyede tanımlanır:

Atomik DEVS modeli; DEVS formalizminin alt parçalara ayrılamayan temel modelidir ve temel yapısal dinamikleri içerir. En düşük seviyede, sıralı durumlar arasındaki geçişler gibi ayrık olaylı sistemin otonom davranışını, harici giriş olaylarına nasıl tepki verdiğini ve çıkış olaylarını nasıl hesapladığını tanımlar. Bir atomik model, giriş-çıkış portlarına, durum değişkenlerine, başlangıç durumuna ve dâhili ve harici durum geçiş fonksiyonlarına sahiptir.

Bir atomik DEVS modeli matematiksel olarak;

$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$  şeklinde tanımlanır. Burada;

X: giriş olayları kümesini,

S: Olası tüm durumlar kümesini,

Y: çıkış olayları kümesini,

$\delta_{int}$ :  $S \rightarrow S$  Dâhili durum geçiş fonksiyonunu,

$\delta_{ext}$ :  $Q \times X \rightarrow S$  Harici durum geçiş fonksiyonu,

Burada;  $Q = \{(s,e) | s \in S, 0 \leq e \leq ta(s)\}$  toplam durum kümesini,

e, en son olan geçişten bu yana geçen süreyi,

ta:  $S \rightarrow R^+_{0,\infty}$  Zaman artış fonksiyonunu, (0 ve  $\infty$  arasındaki pozitif reel sayılar kümesidir.)

$\lambda$ :  $S \rightarrow Y$  Çıkış fonksiyonunu ifade etmektedir.

Birleşik DEVS modeli; daha yüksek bir düzeyde, bir sistemi bileşenler ağı olarak tanımlar. Bileşenler, atomik DEVS modelleri ve diğer birleşik DEVS modelleri olabilirler. Bağlantılar, bileşenlerin birbirini nasıl etkilediğini gösterir. Özellikle, bir bileşenin çıkış olayları, ağ bağlantısı aracılığıyla bir diğer bileşenin giriş olayları olabilir. Birleşik DEVS modeli, bir veya daha fazla atomik ve/veya birleşik modelden oluşabilir. Her bir birleşik DEVS için bir atomik DEVS tasarlanabileceği gibi, atomik veya birleşik olan bir DEVS modeli bir atomik DEVS ile gösterilebilir. Birleşik DEVS, başka birleşik DEVS bileşenlerine sahip olabildiği için hiyerarşik modelleme yapısı desteklenir [39].

Birleşik bir DEVS modelinin matematiksel ifadesi ise;

$M = \langle X, Y, D, \{M_i\}, \{I_i\}, EIC, EOC, Secim \rangle$  şeklinde tanımlanır. Burada;

X: Dış girdiler kümesini,  
 Y: Dış çıktılar kümesini,  
 D: DEVS bileşen modelleri kümesini,  
 Her bir  $i \in D$  için,  
      $M_i$ : DEVS bileşen modeli,  
      $I_i$ :  $i$  'yi etkileyen ve  $i$  'den etkilenen diğer bileşenler kümesi,

EIC: Harici girdi olaylarıyla iç bileşen girdilerinin bileşenler kümesini,  
 EOC: Harici çıktılarla iç bileşen çıktılarının bileşenler kümesini,  
 Secim:  $\{M_i\} - \emptyset \rightarrow \{M_i\}$ , eşitlik fonksiyonunu ifade etmektedir.

### 2.6.2. Paralel DEVS yaklaşımı

Klasik DEVS yaklaşımının en önemli problemi aynı anda meydana gelen olayları yönetememesidir. Bu durum paralel DEVS yaklaşımı ile çözülmüştür. Paralel DEVS (P-DEVS), çakışma problemlerinin üstesinden gelindiği klasik DEVS modelinin gelişmiş bir sürümüdür. Paralel DEVS atomik modelinin klasik modelden farkı, çoklu giriş / çıkış portlarına ve değerler kümesine, ayrıca çakışma geçiş fonksiyonuna sahip olmasıdır. Paralel birleşik DEVS modelinin klasik modelden farkı ise seçim (select) fonksiyonuna sahip olmamasıdır.

Paralel bir atomik DEVS modelinin matematiksel ifadesi;

$M = \langle XM, YM, S, \delta_{int}, \delta_{ext}, \delta_{conf}, \lambda, ta \rangle$  şeklinde tanımlanır. Burada;

$XM = \{(p,v) \mid p \in IPorts, v \in X_p\}$  giriş portları kümesi ve değerleri,  
 $YM = \{(p,v) \mid p \in OPorts, v \in Y_p\}$  çıkış portları kümesi ve değerleri,  
 $S$ , durumlar kümesi,  
 $\delta_{int} : S \rightarrow S$  dahili geçiş fonksiyonu,  
 $\delta_{ext} : Q \times X \rightarrow S$  harici geçiş fonksiyonu,  
     burada;  $Q = \{(s,e) \mid s \in S, 0 \leq e \leq ta(s)\}$  toplam durum kümesi,  
      $e$ , en son olan geçişten bu yana geçen süredir.  
 $\delta_{conf} : Q \times X \rightarrow S$  çakışma (confluent) geçiş fonksiyonu,  
 $\lambda : S \rightarrow Y$  çıkış fonksiyonu,  
 $ta : S \rightarrow \mathbb{R}^+, \infty$  zaman ilerleme fonksiyonu, 0 ve  $\infty$  arasındaki pozitif reel sayılar kümesini ifade etmektedir.

Paralel bir birleşik DEVS modelinin matematiksel ifadesi ise;

$M = \langle X_M, Y_M, D, \{M_i\}, EIC, EOC, IC \rangle$  şeklinde tanımlanır. Burada;

$X_M = \{(p,v) | p \in IPorts, v \in X_p\}$  giriş portları kümesi ve değerleri,

$Y_M = \{(p,v) | p \in OPorts, v \in Y_p\}$  çıkış portları kümesi ve değerleri,

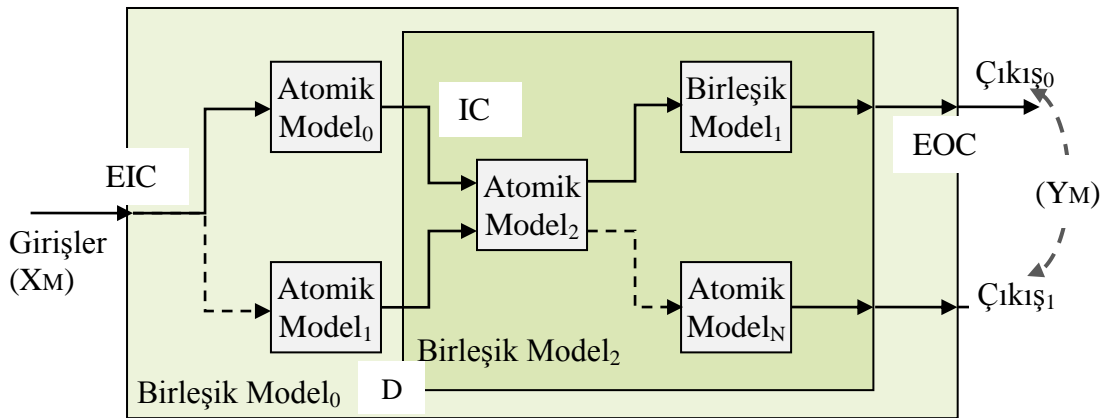
D: Birleşik modelin bileşenler kümesi,

Her  $i \in D$  için,  $M_i$ : i bileşenin DEVS modelidir;

EIC: Harici giriş bağlantısı: Dış girdi olaylarıyla iç bileşen girdilerinin bileşenler kümesi,

EOC: Harici çıkış bağlantısı: Dış çıktılarla iç bileşen çıktılarının bileşenler kümesi,

IC: Dâhili bağlantıları ifade etmektedir.



Şekil 2.9. Paralel DEVS birleşik modeli

Şekil 2.9’da paralel bir DEVS birleşik modelinin hiyerarşik yapısı görülmektedir. Şekilden de görüldüğü üzere bir birleşik model, bir veya daha fazla atomik ve / veya birleşik modelden oluşabilir.

## 2.7. DEVS-Suite Benzetim Ortamı

Daha öncede ifade edildiği gibi, DEVS yaklaşımı sistem davranışını atomik (atomic) DEVS ve birleşik (coupled) DEVS olmak üzere iki farklı seviyede tanımlar. Atomik DEVS; en düşük seviyede, sıralı durumlar arasındaki geçişler gibi ayrık olaylı sistemin otonom davranışını, harici bir girişe (olaylar) nasıl tepki verdiğini ve çıkışı

(olaylar) nasıl hesapladığını tanımlar. Birleşik DEVS, daha yüksek bir seviyede, bir sistemi bileşenler ağı olarak tanımlar. Birleşik DEVS, başka birleşik DEVS bileşenlerine sahip olabildiği için hiyerarşik modelleme yapısı desteklenir. Bileşenler, atomik DEVS modelleri ve birleşik DEVS modelleri olabilirler. Bağlantılar, bileşenlerin birbirini nasıl etkilediğini gösterir. Özellikle, bir bileşenin çıkış olayları ağ bağlantısı aracılığıyla bir diğer bileşenin giriş olayları olabilir [9, 10, 39, 53].

DEVS yaklaşımının çok sayıda yazılım uyarlaması vardır. DEVSJAVA, DEVSIM++, phytonDEVS, ADEVS, DEVS/C++ ve DEVS-Suite bunlardan birkaçıdır [56].

DEVS-Suite ve DEVSJAVA, paralel DEVS ve onunla ilişkili teknolojilerin nesne yönelimli bir gerçekleştirimidir. Java programlama dilinin gelişmiş özelliklerini ve nesne yönelimli programlama tekniklerini kullanarak karmaşık sistemlerin ve ağ sistemlerinin davranışlarını DEVS yaklaşımını kullanarak görüntülerler.

DEVSJAVA; DEVS yaklaşımını / metodolojisini kullanan, nesne yönelimli yapısı ile bir ağı oluşturan düğümlerin, yazılım varlıklarının ve deneysel çerçevelerin modüler bir yapıda tasarımını, yeniden kullanımını sağlayan tamamen Java sınıf ve paketlerinden oluşan bir modelleme ve benzetim ortamıdır. Örneğin GenCol, veri yapılarının / varlıkların sunumu için gerekli sınıfların yer aldığı bir pakettir. genDevs, formalizm / yaklaşım prensiplerinin çekirdeğini oluşturan bir pakettir. simView ise benzetim sonuçlarının grafiksel olarak sunumunu yapan bir pakettir.

DEVS-Suite; Java programlama dili ile geliştirilmiş ve DEVSJAVA benzetim aracının yeni bir sürümü olan genel bir modelleme ve benzetim aracıdır. Bölüm 5' de anlatılan 'benzetim araçlarının karşılaştırılması' başlığı altında sunulacak bilgilerden görüleceği üzere, DEVS-Suite benzetim aracını diğer araçlardan öne çıkaran birçok özelliğin arka planında, Java programlama dili ile tasarlanmış olması yatmaktadır.



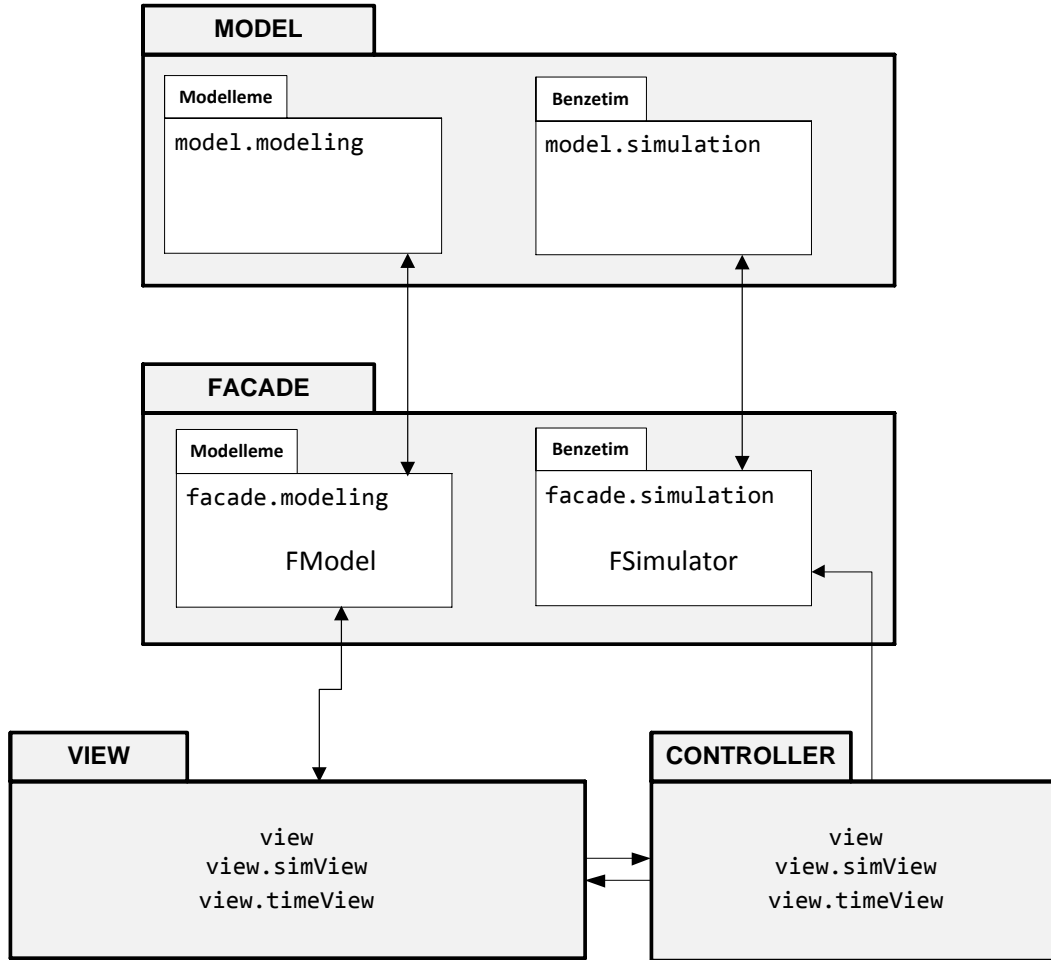
Java, Sun Microsystems mühendislerinden James Gosling tarafından geliştirilmeye başlanmış %100 nesne yönelimli, platformdan bağımsız, yüksek performanslı, çok işlevli, dağıtık, yüksek seviyeli ve ücretsiz bir programlama dilidir. İlk sürümü 1996 yılında çıkmış olup, takip eden yıllarda 1.1, 1.2, 1.3, 1.4, 5.0, 6.0 ve 7.0 gibi sürümleri geliştirilmiştir [57]. Java programlama dili C/C++ programlama dili ile aynı gramer yapısına sahip olmakla birlikte ondan daha yüksek bir başarıma sahiptir [58].

Java tabanlı ayrık olaylı sistem modelleme araçlarının kullanımı gittikçe artmaktadır. Basitlik, %100 nesne yönelimli, dağıtık çalışabilme, taşınabilirlik, platform bağımsızlığı ve ücretsiz olması gibi pek çok özelliğinden dolayı Java dili modelleyiciler tarafından tercih edilmektedir [28, 59 - 61].

DEVS-Suite, DEVS yaklaşımına dayalı, açık kaynak kodlu, ayrık olaylı genel amaçlı bir benzetim ortamıdır ve modüler, hiyerarşik, ayrık olaylı sistem ortamı, nesneye dayalı Java programlama dili kullanılarak gerçekleştirilir. DEVS-Suite, paralel DEVS tabanlıdır ve benzetim sonuçlarının daha iyi izlenebilmesi için bazı eklentiler içeren (simview, timeview gibi) DEVSJAVA benzetim aracının yeni bir sürümüdür. DEVS Suite ile gerçekleştirilen benzetim, adım adım veya sürekli çalıştırılabilir ve bir sonraki ( $T_N$ ) veya en son ( $T_L$ ) olay zamanı izlenebilir. DEVS Suite'in ana bileşenlerini, simview, DEVS izleme ortamı (DEVS Tracking Environment-DTE) ve timeview oluşturmaktadır [10, 33, 62].

Şekil 2.10'dan da görüleceği üzere DEVS-Suite benzetim ortamı, Model View Control (MVC) mimarisine Façade tasarım şablonunun eklenmesi ile geliştirilmiştir. DEVS-Suite paket yapısı Model, Façade, View ve Controller paket ve alt paketlerinden oluşmaktadır.

Model; arka planda çalışan birbiri ile etkileşen sınıfları barındıran bir pakettir. Kod sayfasında gösterilen veriler aslında arka planda model kısmında tutulur.



Şekil 2.10. DEVS-Suite MFVC Paket Yapısı

Façade; bileşenlerin dış dünya ile iletişimde tanımlanmış giriş / çıkış noktalarını / portlarını kullanmasını sağlayan pakettir. Paketin içeriğini, dış dünyadan tamamen gizler.

View; kullanıcı ara yüzünü barındıran pakettir. Ekranda görünen ara yüzü ve grafiksel öğeleri içerir.

Controller; Model ve View arasındaki etkileşimi sağlayan pakettir.

DEVS-Suite, platformdan bağımsız, bir benzetim dili öğrenme gereksinimini ortadan kaldıran yüksek performanslı ve kullanıcı dostu grafiksel bir ara yüze sahiptir. DEVS-Suite benzetim aracı kişisel bilgisayarlar üzerinde çalıştırılabileceği gibi

DEVS-Suite Web Start [10] ile çevrimiçi olarak web ortamında da çalıştırılabilir. Böylece uzaktan eğitim amaçlı da kullanılabilir.

## **BÖLÜM 3. PARALEL ve DAĞITIK BENZETİM**

### **3.1. Giriş**

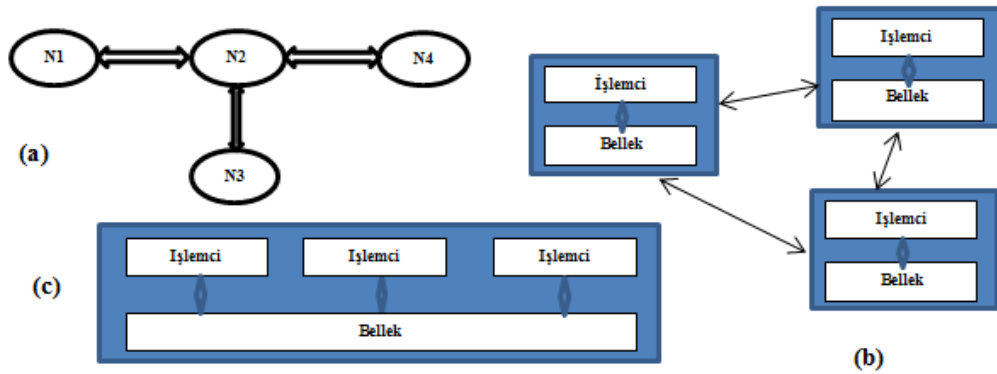
İnternet ve sosyal ağ uygulamalarının hızlı bir şekilde artışı ve yeni teknolojilerin gelişimi, uzaktan eğitimin yaygınlaşmasını, beraberinde benzetim uygulama ve araçlarının da web tabanlı (online) olmasını zorunlu kılmaktadır. Her geçen gün daha karmaşık hale gelen gerçek sistemlere paralel olarak benzetimlerde daha karmaşık hale gelmiş, dolayısı ile benzetimlerden beklenen yetenekler / işlevler de artmıştır. Gerçek sistemin modeli karmaşıklaştıkça benzetim için ihtiyaç duyulan işlem miktarı da artmış ve tek bir bilgisayarın işlem kapasitesini aşmıştır. Dolayısıyla paralel ve dağıtık çalışan benzetim teknolojilerine ihtiyaç ortaya çıkmıştır.

Eğitim amaçlı geliştirilen benzetim araçlarına uzaktan erişilmesi gerekliliği birden çok benzetim ögesinden oluşan ve birbirleri ile haberleşen benzetim sistemlerini doğurmuştur. Bu tip benzetim sistemlerinin gerçekleştirilmesi ise paralel ve dağıtık sistemler ile gerçekleştirilebilmektedir (Şekil 3.1).

Dağıtık çalışan bir sistem, birkaç bağımsız işlemciden ve bu işlemcileri destekleyen veri depoları veya veri tabanlarından oluşur [3]. Bunlar bütün bir hedefi elde etmek üzere işbirliği yapmak için etkileşim halindedirler. İşlemler kendi aktivitelerini koordine ederler ve bir haberleşme sistemi aracılığıyla bilgiyi karşılıklı değişirler. İşbirliğini tesis etmek için sistemin bileşenleri arasında çok yakın bir senkronizasyon olması gerekir.

Dağıtık benzetim, dağıtık bilgisayar teknolojileri ile geleneksel sıralı benzetim tekniklerini birleştirmektedir. Bu işlem, çok işlemcili bir bilgisayarda gerçekleştirilebileceği gibi bir ağ ile birbirine bağlanmış bilgisayarlar arasında da

gerçekleştirilebilmektedir. Dağıtık benzetim işlem süresinde elde edilen kazanç yanında zaman tasarrufu da sağlamaktadır. Coğrafik olarak farklı yerlerdeki kullanıcıları bir araya getirebilmek ve farklı veri tabanlarını kullanabilmek dağıtık benzetim ile mümkün olabilmektedir. Dağıtık benzetimde modeli kurulacak sistem alt öğelerine ayrılmakta ve birbirleriyle bağlantılı iş istasyonlarında benzetim modeli çalıştırılmaktadır [63, 64].



Şekil 3.1. (a)–(b) Dağıtık bir sistem. (c) Paralel bir sistem.

‘Dağıtık benzetimi’; belirli bir amaca yönelik olarak birbirine coğrafik olarak dağıtık bağlanmış bilgisayar topluluğu olarak da düşünebiliriz. Burada önemli olan, ortak amacın gerçekleştirilmesidir. ‘Paralel benzetim’ ise; benzetim programlarının / uygulamalarının çok işlemcili ortamlarda çalıştırılması ile ilgilidir. Çok işlemcili birden fazla bilgisayar üzerinde çalışan benzetim uygulamaları ise, ‘paralel ve dağıtık benzetim’ olarak tanımlanmaktadır. Esasen her iki durumda da (paralel veya dağıtık) tek bir benzetim uygulaması çok işlemcili bilgisayarlarda gerçekleşmektedir.

Paralel benzetimlerde benzetim öğeleri genel olarak birbiri ile aynı özelliklerde (özdeş) olup bir işlemin parçalara bölünerek daha hızlı bir şekilde sonuç alınması hedeflenmektedir. Dağıtık benzetimlerde ise farklı özellik ve işlevlerdeki (heterojen) benzetim öğeleri farklı modellerin gerçekleşmesini sağlayarak benzetim sistemini oluşturmaktadır. Dağıtık benzetimlerde herhangi bir benzetim öğesi paralel çalışan alt öğelerden oluşabilir. Böylece dağıtık ve paralel benzetimler iç içe kullanılabilir. Paralel ve dağıtık benzetim, günümüzde yüksek seviyede işlem gücü ve bellek gerektiren benzetim problemlerini çözmeye çok daha önemli hale gelmiştir [18].

Dağıtık benzetim sistemleri sayesinde eş zamanlı mühendislik uygulamalarının da dağıtık olarak gerçekleştirilmesi mümkündür [65].

Günümüz bilgisayarlarının yüksek grafik yetenekleri sayesinde benzetim, eğitim amaçlı da kullanılmaktadır [66].

Paralel ve dağıtık uygulamalar çoğu kez birbirine karıştırılmaktadır. Tablo 3.1, yukarıda ifade edilen paralel ve dağıtık sistemler arasındaki farklar konusunda bizlere kısa bir fikir verecektir.

Tablo 3.1. Paralel ve dağıtık uygulamaların farklılıkları

Paralel Uygulama	Dağıtık Uygulama
Paylaşımli bellek sistemi	Dağıtık bellek sistemi
Çok işlemcili	Çok bilgisayarlı
İşletim sisteminde simetrik çoklu işlem desteği	Çoklu kaynak paylaşımı
Daha basit bir yapı	Daha karmaşık bir yapı
Sınırlı ölçeklenebilirlik	Daha iyi bir ölçeklenebilirlik
Sınırlı alanda uygulanabilme	Dünya ölçeğinde uygulanabilme

Bir benzetim aracını tanımlamak için ‘paralel’ veya ‘dağıtık’ terimlerinin kullanılıp kullanılmaması büyük ölçüde tasarımdaki hedeflerle ilgilidir. Bu hedefleri ise şöyle sıralayabiliriz:

- Hız: Eğer tasarım iyi yapılırsa, bir benzetim aracının çalışması için gereken zaman ona tahsis edilen işlemcilerin sayısına göre azaltılabilir.
- Belleğin verimli kullanılması: Tüm işlemcilerin bütün hafıza kapasiteleri kullanılabilir.

- Veri işleme: Özel düğümler tarafından sağlanan daha büyük veri işleme ve grafik kapasitesinden faydalanılabilir.
- Kaynak paylaşımı: Karşılıklı / birlikte çalışabilirlik ve kaynakların paylaşımı sağlanabilir.

Eğer bir benzetim mimarisinde ana hedef, uygulama veya çalışma zamanını düşürmek ve coğrafik olarak farklı yerlerde konumlanmış benzetim araçlarını birlikte çalıştırmak ise benzetim aracı mutlaka paralel ve dağıtık mimaride tasarlanmalıdır. Bu tez bu hedefi gerçekleştirmek amacıyla DEVS-Suite üzerinde çalışan paralel ve dağıtık bir benzetim altyapısı geliştirmek ile ilgilidir.

### 3.2. Dağıtık Teknolojiler ve Dağıtık Benzetim

Bilgisayar ağları gibi birçok yazılım bileşeninden oluşan bir sistemi modellemek amacıyla, ayırık olaylı sistemlerin sistem teorisi ve modelleme kavramlarını daha farklı ve özgün bir biçimde ifade eden DEVS modelleme ve benzetim yöntemi kullanılır. Bölüm 2’de de anlatıldığı gibi DEVS modelleme yöntemi, sistem teorisi temellerini kullanması nedeniyle nesneye yönelik uygulamalar için elverişlidir. DEVS yaklaşımının modüler ve hiyerarşik modeller kurmaya elverişli esnek yapısının sağladığı avantajlar, bilgisayar ağları gibi ayırık olaylı sistemlerin modellenmesinde sistem teorisi tabanlı bir yaklaşım sağlamaktadır. Dolayısıyla uygun ara birimli hiyerarşik modüllere sahip sistemlerin tasarımı mümkün olmakta ve modern hesaplama teorisinde çok önemli bir yere sahip olan paralelliğin meydana getirilmesi sağlanmakta ve sonuç olarak bilgisayarların sıralı çalışan mimarileri nedeniyle ortaya çıkan sınırlandırmaları ortadan kalkmaktadır [39, 45].

Diğer taraftan, geniş ölçekli bir benzetim sistemini çalıştırmak için tek bir bilgisayar kullanmak çoğu kez pahalı veya imkânsızdır. Bunun yerine birden fazla bilgisayar kullanarak, benzetim bileşenlerini farklı bilgisayarlara atayıp ortak bir hedef doğrultusunda çalışan dağıtık bir sistem tasarlamak çok daha ucuz ve elverişlidir. Bu çerçevede, yüksek dereceli etkileşimli benzetim faaliyetleri için gerçeğe uygun ve

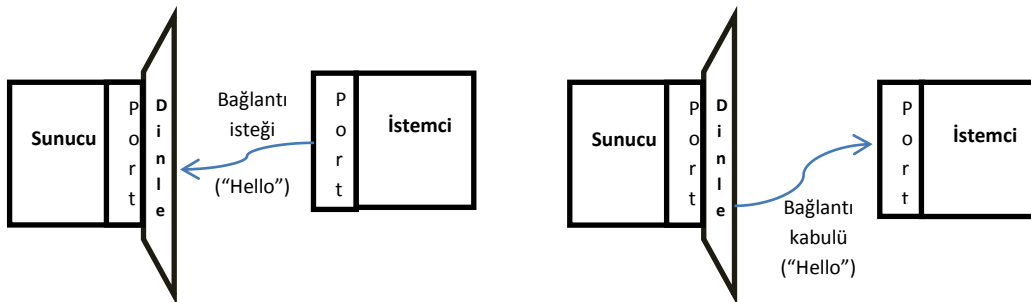
karmaşık ‘sanal dünyalar’ oluşturmak amacıyla dağıtık teknolojiler (İstemci / Sunucu, Java RMI, CORBA, HLA) kullanılarak farklı yer ve farklı türdeki benzetim araçlarının birbirine bağlanması sağlanabilir.

Karmaşık ve geniş ölçekli İnternet uygulamalarını gerçekleştirmek ve yönetmek için nesne yönelimli Java teknolojilerinden (İstemci / Sunucu, kanallar, Java-RMI, CORBA, HLA, kümeleme, web servisleri gibi) faydalanılabilir. Aşağıdaki bölümlerde bu teknolojilerden bazıları kısaca açıklanmaktadır.

### 3.2.1. İstemci / Sunucu mimarisi

İstemci / Sunucu mimarisi, diğer mimarilere (merkezi, zaman paylaşımı, eş düzeyli) oranla kullanılabilirlik, esneklik, birlikte çalışabilirlik ve ölçeklenebilirlik özellikleri iyileştirilmiş çok yönlü, mesaj tabanlı ve modüler bir yapıya sahiptir [7].

Java programlama dilinde istemci / sunucu sistemi soketler ve özel bir protokol (TCP, UDP gibi) aracılığı ile kurulur. Soket programlama, en alt seviyeli programlama tekniğidir. Soketler bir ağ üzerinde çalışan programlar (istemci-sunucu) arasında çift yönlü iletişim sağlarlar. Diğer bir ifade ile uygulamaların kendi aralarında bağlanabilmeleri ve haberleşebilmeleri için kanal işlevi görürler.



Şekil 3.2. İstemci ve sunucu arasındaki iletişim

İstemci, herhangi bir adresten veri almak için sunucunun port numarasını belirterek istekte bulunur. Sunucu ise istemcinin istediği verileri sağlar. Bu iletişimin nasıl gerçekleştiği Şekil 3.2’de görülmektedir.



Dağıtık bir Java uygulamasında istemci tarafının soket yapısının algoritması;

---

```

try {
//Sunucu ile iletişim için bir soket oluşturulur
soket = new Socket(adres, Server.PORT);

o_s = new PrintWriter(new BufferedWriter(new OutputStreamWriter(
soket.getOutputStream())), true);
i_s = new BufferedReader(new InputStreamReader(
soket.getInputStream()));
} catch (IOException e) {
System.err.println('Client server ile iletisim kuramadi!');
}

```

---

Burada, 'adres' değişkeninde sunucu bilgisayarın IP adresi tutulur ve port numarası ile birlikte 'Socket' sınıfından oluşturulan 'soket' isimli bir nesneye aktarılır. Sunucuya veri aktarma ve sunucudan veri okumak için giriş-çıkış kanalları oluşturulur. Okunan veriler / paketler, 'BufferedReader' sınıfı aracılığı ile 'i\_s' değişkeninde, gönderilen veriler / paketler ise 'PrintWriter' sınıfı aracılığı ile 'o\_s' isimli değişkende tutulur.

Dağıtık bir Java uygulamasında Sunucu tarafının soket yapısının algoritması;

---

```

try {
//Sunucu soketi oluştur ve boşteki bir portu bağla
soket = new ServerSocket(PORT);
client = soket.accept();
i_s = new BufferedReader(new InputStreamReader(
client.getInputStream()));
o_s = new PrintWriter(new BufferedWriter(new OutputStreamWriter(
client.getOutputStream())), true);
} catch (IOException ex) {
System.err.println('Hata...:' + ex);
}

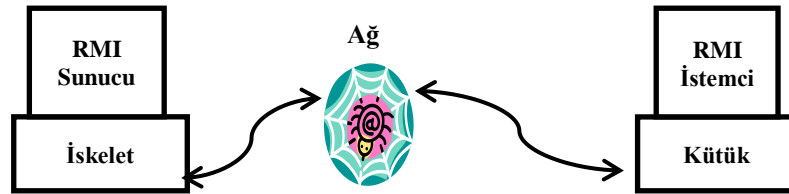
```

---

Burada, 'ServerSocket' sınıfının 'accept()' metodu ile bekleyen istemci bağlantı istekleri kabul edilir. İstemci tarafında olduğu gibi giriş ve çıkış kanalları oluşturularak veri / paket aktarımları gerçekleştirilir.

### 3.2.2. Java Uzak Metot Çağrısı (Java Remote Method Invocation- JAVA RMI)

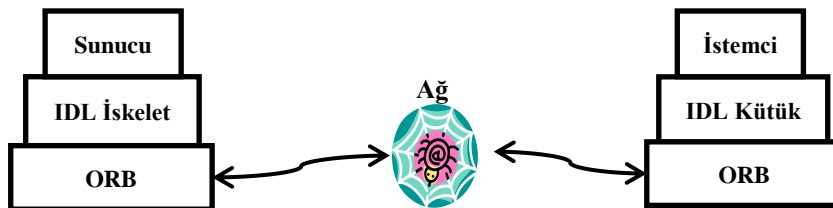
RMI, Java programlama diline özel olarak dağıtık uygulamalarda uzaktaki nesnelerle yerel ortamdaki esneklikte çalışma hedeflenir. Java RMI ve CORBA, ağ uygulamaları için kullanıcıya soketlere göre daha üst seviyeli bir uygulama ara yüzü (API) sunmaktadır. Bu sebeple, RMI ile dağıtık uygulamalar gerçekleştirmek soketlere göre daha kolaydır. Şekil 3.3'te görüldüğü gibi bu mekanizmada istemci / sunucu temelli çalışır. Bir nesne hem istemci hem de sunucu olabilir. Kütük (Stub) istemci, iskelet (skeleton) ise sunucu üzerinde çalışan RMI nesneleri için birer komşu (proxy) işlevi görürler.



Şekil 3.3. Java RMI mekanizması

### 3.2.3. Ortak Nesne İstem Aracı Mimarisi (Common Object Request Broker Architecture- CORBA)

CORBA, OMG (Object Management Group-Nesne Yönetim Grubu)'nun 1991 yılında İnternet / ağ üzerinde birlikte çalışan farklı programlama dilleri için tasarladığı altyapının adıdır. RMI'dan farkı, platform bağımsızlığıdır, fakat CORBA'yı destekleyen dil, kendi 'ara yüz tanımlama dili-IDL' eşleştirmesine sahip olmalıdır.



Şekil 3.4. CORBA mekanizması

Şekil 3.4’de basitleştirilmiş CORBA mekanizması görülmektedir. Şekilde görüldüğü üzere, sunucu ve istemci arasında iletişim ORB (Object Request Broker - Nesne İstek Aracı) arayüzü ile gerçekleşir. Java RMI ile CORBA arasında hız bakımından çok büyük bir fark olmamakla birlikte bütün nesnelere Java dilinde tanımlanmış ise RMI, değilse CORBA’nın tercih edilmesi daha mantıklıdır [67].

### 3.2.4. Yüksek Seviyeli Yapı (High Level Architecture -HLA)

Yüksek Seviyeli Yapı (HLA), etkileşimli benzetim modelleri ve ortamlarının gelişmesine katkıda bulunacak nesneye-yönelik bir yaklaşımdır. HLA, birbirlerine bağlanabilme ve işbirliği yapabilme yeteneğine sahip benzetimlerin karşılıklı çalışmasını sağlayan bir çerçevedir. Bu çerçeve içerisinde işbirliği yapan benzetimler, ‘HLA federasyonu’ olarak isimlendirilir. HLA, genellikle uygulanabilir bir takım kurallar yoluyla karşılıklı çalışabilirliği (interoperability) ve yeniden kullanılabilirliği (reusability) kolaylaştırır. HLA’ nın oluşturduğu yapay ortamda benzetim türündeki uygulamalar, birbirleriyle gerçek zamanlı olarak karşılıklı veri alışverişi yapabilecekleri gibi, standart bilgisayar iletişim servisleri kullanılarak birbirine bağlanmış benzetim araçları ve donanımlarla da karşılıklı bilgi alışverişi yapabilirler.

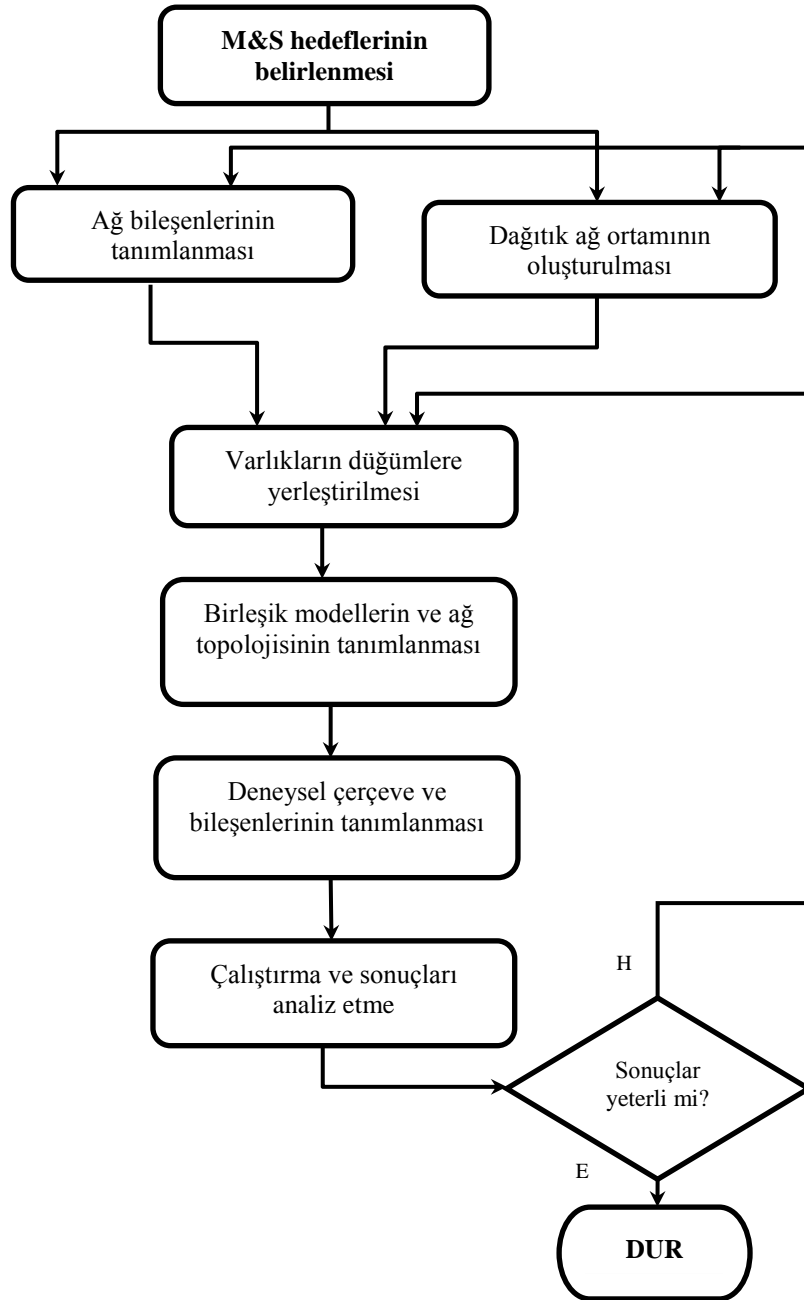
HLA teknolojisi; farklı amaçlar için inşa edilen sistemleri, farklı dönemlerdeki teknolojileri, değişik ürünleri ve çeşitli servis platformlarını bir araya getirir ve bunların yapay bir ortamda karşılıklı olarak çalışmalarına olanak tanır. Bir başka ifadeyle HLA, kendisinden daha önceki standartları (DIS, ALSP, gibi) kullanarak bir araya gelmiş benzetim gruplarıyla da veri alışverişi yapabilir [68].

### 3.3. Dağıtık Ağ Modelleme ve Benzetim Süreci

Dağıtık ağ sistemlerinin tasarım ve analizini sistematik bir şekilde gerçekleştirmek için bu sistemlerin modellenmesi ve benzetim işlemleri bir takım aşamalara bölünür. Bu aşamaların algoritması, maddeler halinde aşağıda sıralanmış ve akış şeması Şekil

3.5'te gösterilmiştir. Verilen algoritma incelendiğinde dağıtık bir ağ benzetim ortamının geliştirilmesinde altı aşamalı süreç olduğu gözlemlenir [44].

İlk adım olarak; modelleme ve benzetim (M&S) hedefleri belirlenir. Hedeflerin belirlenmesi ile geliştirilen sistemin hangi problemler üzerine odaklandığı açıklanır. Önceden hedefler bilirse, bu hedeflere uygun deneysel çerçeveler geliştirilebilir.



Şekil 3.5. Dağıtık ağ modelleme süreci algoritması

İkinci adımda; benzetim bileşenlerinin tanımlanması işlemi gerçekleştirilir. Bu aşamada benzetimi oluşturan bütün bileşenler ve bu işlemlere ilişkin parametreler tanımlanır. Düğümler ve varlıklar ayrı ayrı ele alınarak dağıtık mimaride (istemci-sunucu taraflı) bir ağ ortamı oluşturulur.

Üçüncü adımda; geliştirilen varlıklar ve düğümler birleştirilirler.

Dördüncü adımda; bileşenlerin benzetim ortamında birbirlerine bağlanması ile değişik ağ topoloji ve bileşenleri meydana getirilebilir.

Beşinci adımda; M&S hedeflerine ve geliştirilen modellere uygun bir deneysel çerçeve tanımlanır. Bu deneysel çerçeve modelin bir takım şartlar altında test edilmesine ve gözlem yapılmasına imkân sağlar.

Son adımda ise; geliştirilen benzetim aracı test edilir ve sonuçlar gözlemlenir. Eğer sonuçlar kabul edilebilir bir aralıkta ise benzetim işlemi sonlandırılır. Aksi halde, tekrar başa dönülür ve geliştirilen modellerin parametreleri yeniden ayarlanır. Böylece algoritma adımlarında ileri-geri hareket edilerek en uygun model geliştirilmiş olur [28].

## **BÖLÜM 4. YÖNLENDİRME PROTOKOLLERİ**

### **4.1. Giriş**

Çoğu durumda aynı bilgisayar ağı veya farklı bilgisayar ağları üzerindeki iki bilgisayar / düğüm arasında bir veri paketini göndermek için birden fazla yol bulunur. Bilgisayar ağları üzerinde yer alan bir bilgisayarın aynı ya da farklı bir ağ üzerinde bulunan başka bir bilgisayara nasıl ulaşacağına karar verme işlemine ‘yönlendirme (routing)’, bilgisayarlar arasındaki iletişimi sağlamak amacıyla verileri düzenlemeye yarayan, standart olarak kabul edilmiş kurallar dizisine ise ‘protokol’ adı verilir. Ağ iletişiminin tümü protokollere dayanır. Diğer bir ifade ile protokol, bir mesajın, nasıl hazırlanacağını, gönderileceğini, iletişim kanallarının nasıl düzenleneceğini ve yönetileceğini tanımlayan bir dizi kurallar manzumesidir.

Yönlendirme işlemini gerçekleştirmek için yönlendiricinin hedef ağ adresini ve o ağın nerede olduğunu, hedef ağa gidilebilecek olası tüm yolları ve bu yollardan en iyisini bilmesi gerekir [69]. Yönlendirme işinin gerçekleşmesinde ana görev yönlendirme protokollerine düşer. Birçok yönlendirme protokolü bulunmasına karşılık temel işlevleri aynıdır. Yönlendirme protokollerinin temel işlevleri;

- Ağın durumunu, olası değişiklikleri izlemek ve duyurmak,
- Hedef düğüme giden en kısa yolu hesaplamak,
- Paketleri kaynaktan hedefe doğru yönlendirmek.

Paketi gönderme işlemi tüm yönlendirme protokolleri için aynıdır. Bu protokolleri birbirinden ayıran temel özellik; ağdaki değişikliklerden (topolojinin değişikliklerinden) nasıl haberdar olacakları ve her yön için en kısa yol hesabını nasıl yapacaklarıdır.

Yönlendirme protokolleri hedef düğüme ulaşmada en iyi yolu belirlemek için ölçütleri (yola ulaşılabilirliği ölçen bir değer) kullanır. En uygun yolun bulunması ve yönlendirme kararının verilmesi, yol uzunluğu, yolun bant genişliği, gecikme, güvenlik, ara düğümlerin sayısı gibi göz önüne alınan birçok parametreye / ölçüte bağlıdır. Yönlendirme protokolleri, türüne göre bu parametrelerden birine veya birkaçına bakarak görevlerini yerine getirirler.

Yönlendirme protokollerinin temeli, yönlendirme algoritmalarına dayanır. Bu algoritmalar ile gönderilecek verinin hangi yoldan gideceği, yönlendirme elemanları tarafından hesaplanır. Yönlendirme protokolleri, yönlendiricilerin yol bilgisini paylaşmasını sağlar [70].

Yönlendiriciler, yönlendirme protokollerini kullanarak hangi ağların erişilebileceğini belirler ve en iyi yolu kararlaştırmak için aralarında bilgi alışverişinde bulunurlar. Bu nedenle; yönlendirme protokolleri, ağ içindeki yönlendiricileri bulmak / tutmak için gereklidir ve sadece yönlendiriciler arasında kullanılır. RIP, OSPF, EGP, IGP, BGP vb. protokoller, yaygın olarak kullanılan yönlendirme protokollerinden bazılarıdır.

Mevcut İnternet yönlendirme altyapısı da otonom sistemler (Autonomous Systems-AS) arasında uzaklık vektörü yönlendirme protokollerini (örneğin BGP) kullanırken; otonom sistemler içerisinde ise bağlantı durumu yönlendirme protokollerini (örneğin OSPF, IS-IS vs.) kullanır [71-73].

## **4.2. Yönlendirme Algoritma ve Protokolleri**

Yönlendirme temel olarak, paketlerin bir ağdan diğerine iletilmesi işlemidir ve OSI referans modelinin [74] üçüncü katmanı olan ağ katmanında gerçekleşir. Yönlendirme işlemini gerçekleştirmek için yönlendiricinin hedef ağ adresini ve o ağın nerede olduğunu, hedef ağa gidilebilecek olası tüm yolları ve bu yollardan en kısa olanını bilmesi gerekir.

Yönlendirme algoritmaları, yönlendiriciler üzerinde tutulan ve en kısa yolun belirlenmesinde kullanılan tabloların güncellenmesi için kullanılır. Yönlendirme algoritmalarını, ‘statik’ (adaptif olmayan) ve ‘dinamik’ (adaptif) yönlendirme algoritmaları olmak üzere ikiye ayırabiliriz. Statik yönlendirmede, yönlendirme tabloları belli bir algoritmaya dayanarak önceden oluşturulur ve bir daha değiştirilmez. Bu durumda, bir düğümden diğer düğümlere ulaşmak için kullanılacak yollar önceden bellidir ve ağdaki trafik değişiminden etkilenmez. Dinamik yönlendirmede ise yönlendirme tablolarının zaman içerisinde, ağ trafiğinde ya da bağlantılarda meydana gelen değişimlerle, güncellenmesi hedeflenir.

Dinamik bir ağdaki trafik değişkendir. Her yönlendirici, periyodik olarak doğrudan bağlantılı olduğu komşularına olan uzaklığı ölçmekte ve değişiklikleri bildiren bir güncelleme paketini ağda bulunan tüm yönlendiricilere göndermektedir. Böylece ağda bir değişiklik olduğunda tüm bilgilerin gönderilmesi yerine sadece değişen/güncellenen kısım ile ilgili bilgi gönderilmektedir.

Statik algoritmalar yol tercihlerini yaparken yolun trafiği, uzunluğu gibi kıstasları göz önüne almadan, daha önceden kendisine verilen plan dâhilinde karar verirlerken, dinamik algoritmalar bunun tersine her veri paketi için trafiği, yolun uzunluğunu tekrar gözden geçirerek karar verirler.

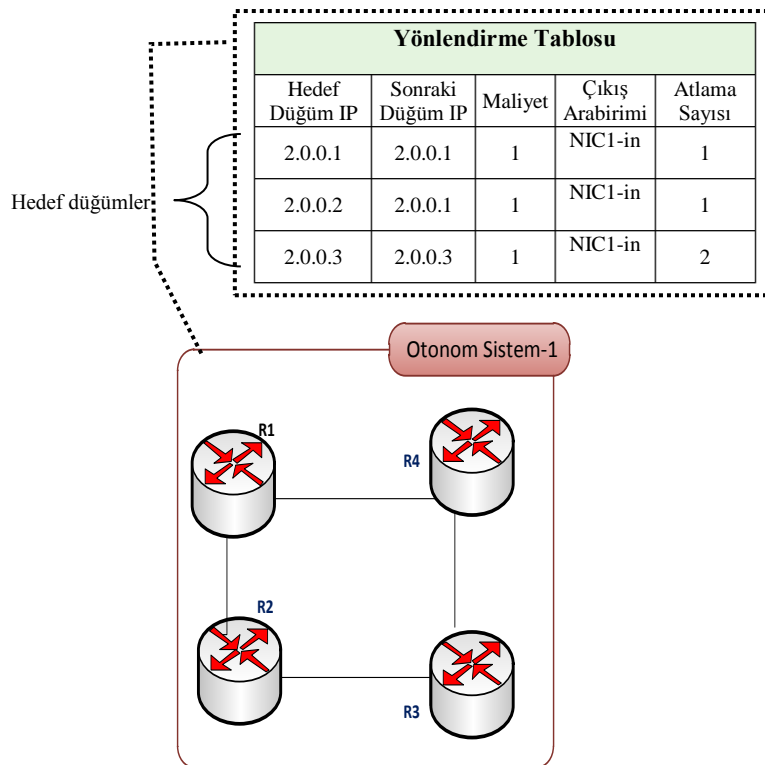
Yönlendiriciler, en iyi yolu bulmak için yönlendirme tablolarını kullanırlar. Yönlendirme işinin gerçekleştirilmesi için sistemler üzerinde yönlendirme tablosu tutulur. Yönlendirme protokollerinin ve algoritmalarının amacı, yönlendirme tablolarını oluşturup bu tabloları kullanıcı protokollerinin hizmetine sunmaktır. Yönlendirme tablolarını, kısaca ağdaki yol durum bilgisini tutan tablolar olarak tanımlayabiliriz. Ağdaki her bir bilgisayar / düğüm yönlendirme tablolarına bakarak hangi veri paketini, nereye göndereceğine karar verir.

Yönlendirme tablolarının oluşturulması / güncellenmesi yönlendirme algoritmalarının görevidir ve yönlendirme algoritmasının başarısını belirleyen bir etkidir. Yönlendirme tabloları, gönderilecek veri paketlerinin alıcısına ulaşması için hangi



yolun izleneceğini belirten yol bilgilerini tutar. Bu bilgiler, ağ yöneticisi tarafından statik olarak elle girilir veya kullanılan yönlendirme algoritması tarafından dinamik olarak doldurulur ve güncellenir. Yol bilgilerinin tabloya elle girilmesi, zahmetli bir iş olması yanında, hata oranını artırır ve güncelleştirmeyi zorlaştırır. Bu dezavantajlar yüzünden yönlendirme tablolarının statik olarak elle oluşturulması yerine bu bilgilerin otomatik olarak girilmesi ve dinamik olarak güncellenmesi için çeşitli yönlendirme protokolleri kullanılır.

Bir yönlendirme tablosu statik algoritmalarda ağ kurulum aşamasında (ağda yük yokken), dinamik algoritmalarda ise ağın yük durumuna göre dinamik bir şekilde uyarlanarak kurulabilir. Dinamik yönlendirme algoritmalarında, ağın genel durumuna bağlı olarak süreç içerisinde tabloya yeni kayıtlar eklenebilir ve mevcut kayıtlar kaldırılabilir [28].



Şekil 4.1. Örnek bir ağın yönlendirme tablosu

Şekil 4.1’de ki örnek yönlendirme tablosundan da görüleceği üzere bir yönlendirme tablosu, hedef düğüm IP adresleri, bu hedefe ulaşmak için hangi düğüme

başvurulması gerektiği, hedefe ulaşmak için gerekli atlama sayısı, çıkış arabirimi gibi bilgileri içerir. Ayrıca geniş ölçekli ağlarda bu bilgilere ek olarak, otonom sistem numarası, kuyruk durumu, gönderilen, hedefe ulaşan ve kaybolan paket sayıları, komşuluk tablosu ve topoloji veri tabanı gibi yönlendirme bilgilerini de içerebilir.

Yönlendirme tablolarının oluşturulma, güncelleme ve yakınsama (convergence) hızı ağ trafiğinin dengeli olmasını (ağda boşa ve aşırı yüklü bir birimin olmaması) doğrudan etkilediğinden, yönlendirme algoritmalarının kalitesini ölçme aracı olarak kullanılabilir [39].

Yönlendirme tablosunun hesaplanması işlemleri de ayrı bir çalışma alanıdır. Örneğin H. El-Sayed ve arkadaşlarının bir çalışmasında [75], Dijkstra en kısa yol algoritması kullanılarak, yönlendirme tablosu hesaplaması optimize edilmiş, yanıt verme ve yakınsama süresi azaltılmış ve ağ büyüdükçe daha fazla işlemci gücü ihtiyacını azaltan yönlendirme motoru ölçeklenebilirliği geliştirilmiştir.

### **4.3. Yönlendirme Protokollerinin Sınıflandırılması**

Yönlendirme protokolleri, kullanım alanlarına veya işleyişlerine göre sınıflandırılabilir. Kullanım alanlarına göre otonom sistem içi yönlendirme protokolleri (Interior Gateway Protocol-IGP) ve otonom sistemler arası yönlendirme protokolleri (Exterior Gateway Protocol – EGP) olmak üzere ikiye ayrılabilir.

Otonom sistem, bir yönetimsel merkezden yönetilen ağları tanımlar. İnternete bağlı her servis sağlayıcının kendi adına kayıtlı bir otonom sistem numarası vardır ve kendi ağlarında bu otonom sistem numarasını kullanırlar.

Otonom sistem içi yönlendirme protokolleri, bir otonom sistemin içinde kullanılacak yönlendirme protokolleridir. Bu yönlendirme protokollerine RIPv1, RIPv2, IGRP, EIGRP, OSPF ve IS-IS protokolleri örnek olarak gösterilebilir [71]. Otonom sistemler arasında çalışan yönlendirme protokollerine ise BGP protokolü örnek olarak gösterilebilir.

Daha yaygın bir sınıflandırma şekli olarak yönlendirme protokolleri, yönlendiricilerin en uygun yolu hesaplamada kullandıkları algoritma esas alınarak işleyişlerine göre uzaklık vektörü, bağlantı durumu ve karma olmak üzere üç alt sınıfa ayrılabilir. Örneğin IP ağlarda oldukça fazla kullanılan RIP, uzaklık vektörü algoritmasına, OSPF ise bağlantı durumu algoritmasına örnek protokollerdir [76-78].

#### 4.3.1. Uzaklık Vektörü (Distance-Vector) protokolleri

Uzaklık vektörü protokolleri, uzaklık vektörü algoritmalarına (örneğin Bellman Ford yönlendirme algoritması) dayanır ve yönlendiriciler arasında uzaklık bilgisi (veya atlama sayısı) ölçüt değeri olarak kullanılır. Her yönlendiricide paketlerin gönderilebileceği diğer komşu yönlendiriciler için uzaklık vektör tablosu (bilinen her bir ağa ilişkin atlama - hop sayısını içeren çizelge) oluşturulur. En kısa yola, bu vektöre dayanarak karar verilir. Uzaklık vektörü, yönlendiricilerin hemen komşusu olan yönlendiricilere göre hesaplanır. Yönlendiriciler, kendi taraflarındaki yönlendirme tablosu bilgilerini, diğer tüm komşu yönlendiricilere yayma yolu (broadcast) ile bildirir ve her yönlendirici kendisine gelen yeni durumları tuttuğu tabloya yansıtır. Buda dolayısıyla yüksek bir bant genişliği gerektirir.

Yönlendirme tablosunun transfer işlemi sonucunda, tüm yönlendiriciler, ağdaki diğer yönlendiricilere olan uzaklık vektörü değerlerini öğrenmiş olurlar. Yönlendiriciler, paketleri, bir noktadan diğer bir noktaya iletirken yönlendirme tablolarındaki en küçük uzaklık vektörü değerine sahip olan yolları seçer ve bu şekilde ağdaki iki nokta arasındaki en uygun yol belirlenmiş olur. Yönlendirme tablosundaki bu yol bilgileri sayesinde de yönlendiriciler paket iletimini gerçekleştirirler. Uzaklık vektörü protokolleri, bir ağda i kaynak düğümünden j hedef düğümüne giderken en kısa yolu ‘Bellman-Ford’ algoritmasına göre aşağıdaki şekilde hesaplarlar [52, 76] .

Eğer i ve j direkt olarak bağlı ise,  $d_{ij}$  aradaki bağlantının ölçütü olarak tanımlanabilir. Eğer i ve j direkt bağlı değil ise  $d_{ij}$  sonsuz olarak kabul edilir. En kısa yolun bulunması için yapılacak işlem, yoldaki tüm bağlantıların maliyetlerinin /ölçütlerinin toplanmasıdır.

$D(i, j)$  ise  $i$  ile  $j$  arasındaki en iyi yol olsun. Bu durumda;

$$D(i, i) = 0, \quad \text{her } i \text{ için,}$$

$$D(i, j) = \min_k [d(i, k) + D(k, j)], \quad i \neq j \text{ için,}$$

Görüldüğü gibi ikinci eşitlik  $i$  nin komşuları olan  $k$  lar ile sınırlanabilir, çünkü diğerleri için  $d(i, k)$  değeri sonsuzdur [76].

Bu algoritmaya göre çalışan uzaklık vektörü protokollerinde her yönlendirici ( $i$ ), tüm komşularından ( $k$ ) yönlendirme tablolarını alır. Bu tablolarda  $j$  düğümü için mevcut ölçüt değerine kendisi ile tabloyu aldığı komşusu arasındaki bağlantının ölçüt değerini ekleyerek  $j$  düğümüne gitmek için kullanılacak yolun maliyetini bulur. Daha sonra hesapladığı tüm yollardan en düşük ölçüte / maliyete sahip olanı seçer. Bulunan bu en iyi yol ağın topolojisi değişmediği sürece sabit kalacaktır.

Uzaklık vektörü protokolüne örnek olarak RIP (Yönlendirme Bilgisi Protokolü - Routing Information Protocol) ve IGRP (İç Ağ Geçidi Yönlendirme Protokolü - Interior Gateway Routing Protocol) gösterilebilir.

#### **4.3.1.1. Yönlendirme Bilgilendirme Protokolü (RIP)**

RIP (Routing Information Protocol), bir TCP/IP ağındaki yönlendiricilerin birbirini otomatik olarak tanımasında kullanılan bir protokoldür. Aynı zamanda uzaklık vektör algoritmasına dayanır.

RIP'in mevcut iki sürümü vardır. RIP'in ilk sürümü (RIPv1) 1988 yılında RFC 1058 belgesi ile ikinci sürümü (RIPv2) ise 1994'te RFC 1723 belgesi ile tanımlanmıştır. RIP'te en fazla izin verilen düğüm atlama sayısı 15'dir. Yani 16. adımdan sonraki ağlar erişilemez (unreachable) olarak değerlendirilir. RIPv1, sadece sınıflı yönlendirmeyi kullanır. Güncellemeler esnasında alt ağ maskesi bilgisi gönderilmez. Yani bu sürümde ağdaki tüm cihazlar aynı alt ağ maskesini kullanmak zorundadır. RIPv2 ise sınıfsız yönlendirme olarak adlandırılır ve güncellemeler esnasında alt ağ

maskesi bilgisi gönderilir. Diğer bir fark ise RIPv2'de güncellemeler, broadcast olarak değil multicast olarak gönderilir.

Bu protokolü çalıştıran yönlendiriciler, kendi yönlendirme tablolarının tamamını periyodik olarak (30 saniye aralıklarla) bütün komşu yönlendiricilere gönderirler. Ayrıca en iyi yolu seçerken üzerinden geçilmesi / atlanması gereken adım sayısını (hop count) ölçüt olarak alırlar.

RIP'de, bir yönlendiricinin yönlendirme tablosunu komşu yönlendiricilere duyurmak için bekleme süreleri, 30, 180 ve 240 saniyelik aralıklarla üç farklı seviyede ayarlanabilir. Dolayısı ile RIP'de yakınsama süresi yavaştır. Örneğin 100 yönlendiriciye sahip bir ağda, en baştaki ağın bilgisinin en sondaki yönlendiriciye gitmesi (30 sn aralıklarla yayın yapıldığı düşünülürse) 50 dakikayı bulabilir. Bu süre de ağda sonsuz uzun süre (kısırdöngü) anlamına gelir.

Küçük ağlar için etkili bir biçimde çalışan RIP yönlendirme protokolü, diğer protokollerden daha büyük bir trafik oluşturduğu için büyük ağlarda tercih edilmez.

#### **4.3.1.2. Dahili Ağ Geçidi Yönlendirme Protokolü (IGRP)**

IGRP (Interior Gateway Routing Protocol), Cisco tarafından geliştirilmiş bir uzaklık-vektörü algoritmasıdır. Bu yüzden bir ağda IGRP çalıştırmak için tüm yönlendiricilerin Cisco marka olması gerekir. IGRP'de üzerinden atlanması gereken maksimum düğüm sayısı değeri 255'dir. Bunun haricinde IGRP, RIP'ten farklı olarak en iyi yolu seçerken kullanılan ölçüt için varsayılan olarak, gecikme ve bant genişliğini kullanır. Bunun haricinde güvenilirlik, yük ve MTU (Maximum Transmission Unit) değerleri de ölçüt hesabında kullanılabilir.

IGRP'nin yapılandırılması RIP'e çok benzese de önemli bir fark vardır. O da otonom sistem (AS) numarasıdır. Aynı otonom sistem de bulunan tüm yönlendiriciler aynı AS numarasına sahip olmalıdırlar.

### 4.3.2. Bağlantı Durumu (Link-State) protokolleri

Bağlantı durumu yönlendirme protokolleri, en kısa yol (Shortest Path First -SPF) protokoller olarak da bilinirler. Bağlantı durumu protokolünde en kısa yol, Dijkstra algoritması [79] kullanılarak bulunur. Dijkstra algoritması, kaynak düğüm ile ağdaki diğer düğümler arasındaki en kısa yolu belirlemek üzere tasarlanmıştır.

Bağlantı durumu algoritması, en kısa yolun belirlenmesi için kullanılan metrik değeri, uzaklık bilgisinin yanı sıra yönlendiricilere yapılmış olan bağlantıları da göz önüne alarak hesaplar. Bu algorithmada, ağ içindeki her bir yönlendirici, ağın tüm topolojisi hakkında bilgi sahibidir. Herhangi bir yönlendirici, kendisine olan bağlantıda bir değişiklik olduğunu anladığı zaman, bu değişikliği tüm ağa yayma yoluyla bildirir. Ancak bu yayma işlemi tüm yönlendirme tablosunun gönderilmesi şeklinde olmayıp, yalnızca algılanan değişikliğin bildirilmesi şeklinde olur. Uzaklık vektörü algoritmasında ise komşu düğümlere gönderilen yönlendirme tablosu bilgileri daha fazladır. Genelde, yönlendirme tablosunun tamamı veya büyük bir kısmı gönderilir. Bu durum, uzaklık vektörü algoritmasının ağı daha fazla yüklemesi anlamına gelir. Ancak gönderme işlemi yalnızca komşu düğümlere yapılır. OSPF protokolü, bağlantı durumu algoritmasına dayanan bir protokoldür.

Uzaklık vektöründen farklı olarak yönlendiriciler, birbirlerine nasıl bağlı olduklarını, bağlantı durumu bildirimleri / ilanları (Link State Advertisements - LSAs) ile ağdaki diğer yönlendiricilere bildirirler. Bağlantı durumunda algılanan bir değişiklik, ağdaki yine bu algoritmayı koşturan tüm yönlendiricilere bildirilir.

Her yönlendirici, ağ topolojisini ve bağlantı uzaklıklarını içeren bir veri tabanı içermektedir. Bu veri tabanı akış (flooding) yöntemi ile gönderenin dışında ağda bulunan diğer yönlendiricilere olan en kısa mesafeyi 'en kısa yol' algoritmasını kullanarak hesaplamaktadır [80].

Bağlantı durumu yönlendirmesinin çalışma mantığı kısaca aşağıdaki şekilde özetlenebilir ve bu algoritmayı kullanan her yönlendirici bu adımları yürütür [74]:

- Komşularını keşfeder ve onların ağ (IP) adreslerini öğrenir. Bunun için ağdaki tüm yönlendiricilere ‘hello’ mesajı göndererek selam verir.
- Her bir yönlendirici, bütün komşularına olan maliyet, IP adresi, bağlantı durumu, gecikme gibi bilgilerin yer aldığı bağlantı durumu ilanlarını (LSA) üretir.
- Öğrendiklerini bağlantı durumu veri tabanına kaydeder ve ağda bir durum değişikliği olduğunda tabloları günceller.
- Bu paketleri diğer yönlendiricilere gönderir. Bunun için akış (flooding) algoritmasını kullanır.
- Tüm yönlendiricilere olan en kısa yolu Dijkstra algoritmasını kullanarak hesaplar.

Karma yönlendirme; uzaklık vektörü ve bağlantı durumu yönlendirmelerinin bir karışımı olup, klasik uzaklık vektörü protokollerinden çok daha iyi bir yakınsama (convergence) değerine sahiptirler ve bağlantı durumu güncellemelerindeki ek yükleri önlerler. Yakınsama, yönlendiricilerin ortamdaki tüm ağlardan(yollardan), haberdar olma durumudur. Bu protokollere örnek olarak IS-IS ve EIGRP protokolleri gösterilebilir [76 – 78, 81, 82].

#### **4.3.2.1. İlk Açık En Kısa Yol (OSPF) Protokolü**

OSPF (Open Shortest Path First), RIP’deki bazı problemleri ortadan kaldırmak için ‘İnternet Mühendisliği Görev Gücü’ (Internet Engineering Task Force – IETF) tarafından geliştirilmiş bir bağlantı durumu protokolüdür.

OSPF isminden de anlaşılacağı üzere iki temel özelliğe sahiptir. Birincisi açık (Open-O) kaynak kodlu olması, ikincisi ise ‘İlk En Kısa Yol’ (Shortest Path First - SPF) algoritmasını kullanmasıdır [69]. SPF, algoritmayı bulan kişinin adını belirten Dijkstra algoritması olarak da bilinmektedir [79]. İlk olarak 1989 yılında RFC 1131 numarası ile, en güncel sürümü ise RFC 2328 ile tanımlanan OSPF tanımlamaları / belirtimi sürekli geliştirilmektedir [83 - 85].

OSPF protokolü, geniş ölçekli otonom sistemler (AS) içinde kullanılmak üzere tasarlanmış bir dâhili ağ geçidi protokolüdür. Otonom sistemler arası yönlendirme için tasarlanmamıştır. İnternet üzerindeki otonom sistemler arasında çalışan temel harici ağ geçidi protokolü ise sınır geçit protokolüdür (Border Gateway Protokol-BGP). OSPF, ortak bir yönetim altındaki, kampüs, şirket ve bölgesel ağlar gibi, IP ağlarında Dijkstra algoritmasını kullanarak hedefe gidecek en kısa yolu bulmaya çalışır.

Tablo 4.1. OSPF ile ilgili bazı RFC Tanımlamaları

RFC No	Tarih	Durum/Amaç	Açıklama
1131	1991	Önerilen Standart	OSPF Tanımlaması
1245	1991	Bilgi	OSPF Protokol Analizi
1247	1991	Taslak Standart	OSPF Sürüm 2
1370	1992	Önerilen Standart	OSPF için uygulanabilir ifadeler
1584	1994	Önerilen Standart	OSPF Çoklu yayın eklentisi (MOSPF)
2178	1997	Taslak Standart	OSPF Sürüm 2
2328	1998	Standart	OSPF Sürüm 2
2740	1999	Önerilen standart	IPv6 için OSPF Sürüm 3

OSPF protokolünün nasıl çalıştığı RFC (Request For Comments) olarak isimlendirilen İnternet'teki doküman arşivlerinde ayrıntılı olarak açıklanmıştır. Bir RFC, bir İnternet standardını veya sadece önerileri, fikirleri, istatistiksel bilgileri veya bir standart için gerekli kuralları içeren bir metin belgesidir. Dokümanlar, yayınlanma tarihlerine göre sıraya konular. Bu yüzden konulara göre bir sıralama söz konusu değildir. RFC dokümanları İnternette arşivlenmiştir ve İnternette bunlara erişmek mümkündür. Tablo 4.1'de görüldüğü gibi OSPF protokolü ile ilgili bilgiler, RFC 1131, RFC 1247, RFC 1584, RFC 2178 ve RFC 2328, RFC 2740 gibi dokümanlarda ayrıntılılarıyla verilmiştir [83 - 85].

OSPF protokolü, bağlantı durumu algoritmasının en yaygın kullanılan örneklerinden biridir [71]. Buna göre OSPF, RIP' ten farklı olarak yol bilgisini hızlı bir şekilde



öğrenme (hızlı yakınsama), büyük ve karmaşık ağlarda daha iyi çalışabilme ve güvenilirlik konularında oldukça başarılıdır.

OSPF bağlantı durum ilanlarını makul derecede hızlı bir yakınsama ile birlikte diğer komşuların tümüne aktararak topolojik değişiklikleri kolayca tespit edebilen bir bağlantı durum yönlendirme protokolüdür. OSPF protokolü, uzaklık vektörü protokolleri gibi ölçüt kullanmaz. Yani RIP’de olduğu gibi herhangi bir düğüm sayısı sınırlaması yoktur. Ölçüt hesaplamasında maliyet adı verilen bant genişliği ile ters orantılı değerleri kullanır (Maliyet =  $10^8$  / bant genişliği).

Öte yandan, OSPF bir ağa kaç adımda ulaşabildiğini değil bir ağa ne kadar zamanda (veya hangi hızda) ulaşabildiği bilgisini kullanır. Dolayısıyla, RIP’te görülebilen sonsuza kadar sayma sorunu ortadan kalkmış olur. OSPF’nin RIP’e göre üstünlükleri aşağıdaki gibi özetlenebilir:

- OSPF büyük veya çok büyük ağlar arası ölçeklendirme yapabilir,
- Ağ topolojisi değişikliklerinin yeniden yapılandırılması daha hızlı gerçekleşir,
- OSPF hesaplamalı yollar, her zaman döngüsüzdür,
- Daha güvenilirdir.

OSPF protokolünün çalışma mantığı aşağıdaki şekilde özetlenebilir:

1. Yönlendiriciler içinde bulunduğu ağı öğrenebilmek için 10 saniye aralıklara ‘multicast’ (gruba özel çoğa gönderim) yoluyla ‘hello’ paketleri gönderir ve bu paketlerin içerisinde bulunan alan adı, kimlik doğrulama, ağ maskesi gibi çeşitli değerlerin aynı olup olmamasına bakılarak yönlendiricilerin komşu olup olmadığına karar verilir. Komşuluk tablosunda her bir yönlendiriciye ait farklı bir yönlendirici ID değeri vardır. Bu tabloda ağ adresleri ile yönlendirici ID değerleri eşleştirilir. Bu değer yönlendiricinin kendisine ait bir ‘loopback’ (geridöngü) adresi var ise, o adres üzerinde bulunan en büyük IP adresidir. Eğer herhangi bir geri döngü tanımlanmamışsa yönlendiricinin sahip olduğu en büyük IP değeri ‘Yönlendirici ID’ olarak atanır.

2. OSPF protokolünde ‘hello’ paketlerine gelen cevaplara göre ‘bağlantı durumu ilanı’ (Link State Advertisement – LSA) adı verilen paketler gönderilmeye başlanır. LSA paketleri içerisinde yönlendiricilerin bağlantıları, ara yüzleri ve hat durumu bilgileri yer almaktadır. LSA paket alışverişini yapan her yönlendirici kendisine ait bir LSA tablosu bulundurur ve oluşturulan bu LSA tablosu diğer yönlendiricilere gönderilerek, ağ içerisindeki bütün yönlendiricilerin birbirlerinin LSA tablosunu öğrendiği bir veri tabanı oluşması sağlanır. Oluşturulan bu veri tabanı sayesinde ağ içerisindeki yol bilgisi ve mesafe hesaplanır.
3. SPF (Dijkstra) algoritması yardımıyla ağ topolojisi çıkarılır ve her düğüm belirli bir zaman diliminde (30 sn) ya da olay tetiklemeli olarak LSA paketlerini gönderirler. Ağ üzerinde herhangi bir değişiklik yoksa güncelleme yapılmaz ve ‘hello’ paketleri dışında ağda herhangi bir trafik oluşturulmaz [33].

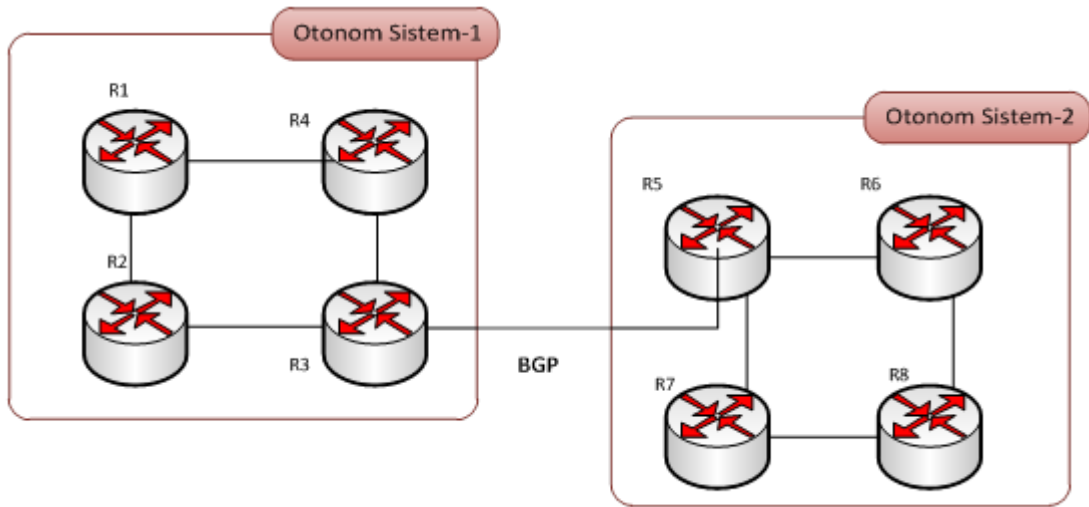
OSPF protokolünde 3 farklı tablo bulunmaktadır: Komşuluk Tablosu (Neighbor Table), Topoloji Tablosu (Topology Table) ve Yönlendirme Tablosu (Routing Table).

- Komşuluk tablosunda, yönlendiricilerin komşularının listesi tutulmaktadır.
- Topoloji tablosunda, otonom sistem içerisindeki bütün yönlendiricileri ve onların bağlantıları tutulmaktadır. LSA, bu tabloda yer almaktadır.
- Yönlendirme tablosunda ise hedef düğümlere giden en kısa yolların bilgisi tutulur. Yönlendirme veri tabanı olarak da isimlendirilir.

#### **4.3.2.2. Sınır Geçit Protokolü (Border Gateway Protocol- BGP)**

Günümüz İnternetinin trafik altyapısında otonom sistem içerisinde OSPF protokolü kullanılırken otonom sistemler arasında ise BGP protokolü [86] kullanılır.

İnternet trafik altyapısı, birbirine farklı türde ara bağlantılarla bağlanmış otonom sistemlerden oluşmaktadır. BGP, otonom sistemleri birbirine bağlamak amacı ile otonom sistemler arası bağlantıda veri trafiğinin hangi yönlendiriciler üzerinden gerçekleştirileceğine karar verilmesini sağlayan bir yönlendirme protokolüdür. Otonom sistemler, kendi içerisinde özerk olan sistemlerdir. Şekil 4.2’de görüldüğü üzere R3 ve R5 düğümleri otonom sistemleri, birbirlerine bağlayan sınır düğümlerdir. Bu düğümler arasındaki veri / paket trafiğini BGP protokolü sağlar.



Şekil 4.2. BGP protokolünün işlevi

## **BÖLÜM 5. AĞ BENZETİM ARAÇLARI**

### **5.1. Giriş**

Bilgisayar ağlarının temel amacı; kullanıcıların, ağıın kaynaklarına ulaşması ve diğer kullanıcılarla iletişimde bulunmasını sağlamaktır. Günümüzde bilgisayar ağları, işletim sistemleri, iletişim protokolleri, bağlantı teknolojileri, trafik akışı, yönlendirme algoritma ve protokolleri gibi çok çeşitli kavramların / uygulamaların karışımından oluşan oldukça karmaşık bir bilim dalı halini almıştır.

Bilgisayar ağları üzerine yapılan araştırmalar hız, altyapı, maliyet, kullanıcı gereksinimlerindeki çeşitlilik ve karşılıklı çalışabilirlik açısından zor bir süreçtir. Ağ tasarımı, maliyet ve kapasite ile birlikte kullanıcı gereksinimlerinin de karşılandığı zor bir görevdir. Bu süreci kolaylaştırmak ve hızlandırmak için ağ modelleme ve benzetim araçları yaygın olarak kullanılmaktadır.

Bilgisayar ağı araştırma ve tasarımlarında modelleme ve benzetim yönteminin, gerçek bir ağıın modellenmesi, yeni ağ teknolojilerinin etkin bir şekilde geliştirilmesi ve test edilmesi, değişik ağ koşulları ve senaryoları altında iletişim protokollerinin ve algoritmalarının, bağlantı teknolojilerinin geliştirilmesinde ve değerlendirilmesinde önemli bir rolü bulunmaktadır [5, 49, 87].

Tasarım aşamasında, zaman ve maliyet avantajları nedeniyle modelleme ve benzetim bilim dalının sağladığı imkânlar yaygın bir şekilde araştırmacılar tarafından kullanılmaktadır. Tasarım sürecini kolaylaştırmak ve hızlandırmak maksadıyla farklı ağ benzetim araçlarının geliştirilmesi ve üretilmesi aktif bir araştırma konusudur.

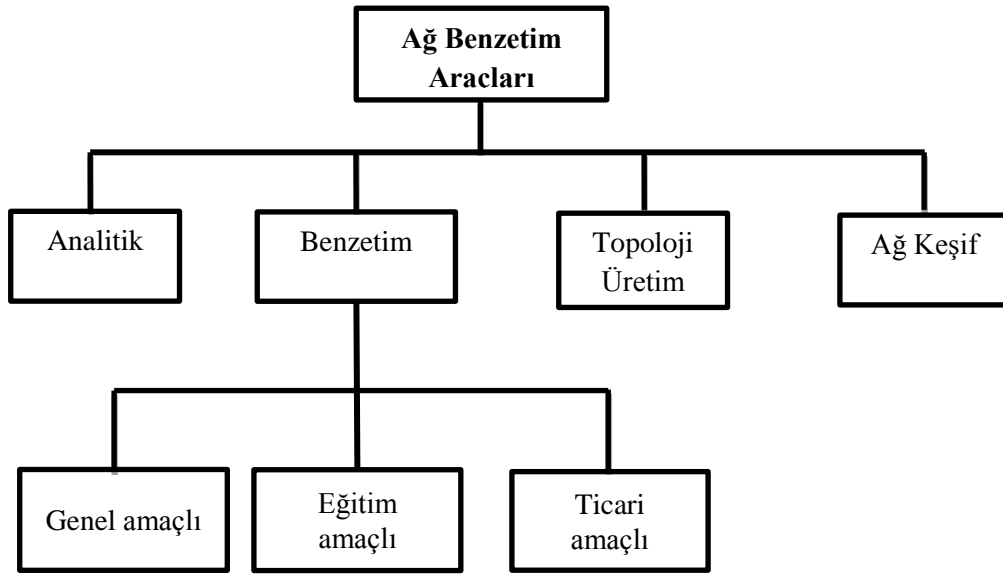
Günümüzde bilgisayar ağları araştırma ve eğitiminde kullanılan birçok benzetim aracı bulunmaktadır. Bu bölümde, bilgisayar ağları konusunda çalışan araştırmacıların araç seçiminde yardımcı olmak amacıyla iyi bilinen benzetim araçları karşılaştırılmıştır. Özellikle eğitim amaçlı olarak kullanılan ağ benzetim araçlarının üstünlük ve zayıflıkları incelenmiş, güçlü ve zayıf yönleri tartışılmıştır.

## 5.2. Ağ Tasarım ve Benzetim Araçlarının Sınıflandırılması

Günümüzde kullanılan çok çeşitli ağ modelleme ve benzetim aracı bulunmaktadır. Bu araçlar; analitik, topoloji üretim, ağ keşif ve benzetim araçları olmak üzere dört sınıfa ayrılabilir (Şekil 5.1).

- Analitik araçlar, bir ağ modelinin tasarım ve hesaplamalarına (örneğin, güvenilirlik, kullanılabilirlik gibi) yardım eder. Analitik model, diğer yöntemler arasında basitlik avantajına sahiptir ve genellikle basitleştirilmiş varsayımlar ve ideal kabuller üzerine kurulur. Bu yüzden kesin sonuçlar istendiğinde analitik modeli oluşturmak karmaşıklık ve zaman tüketimi açısından sistemin prototipini oluşturmak kadar zordur. Delite, Cappuccino, XnetMod, TND-Tool, NetRule, XNP ve NPEST gibi araçlar, analitik araçlara örnek olarak verilebilir [12, 49].
- Topoloji üretim araçları, farklı algoritmaları temel alarak küçük veya geniş ölçekli topolojiler üretmeye ve araştırmacıların ağ performans analizlerine yardım eder. Waxman, Tiers, Transit-stub, GT-ITM, Inet, PRLG, BRITE ve KOM ScenGen gibi araçlar, topoloji üretim araçlarına örnek olarak verilebilir [5, 12].
- Ağ keşif araçları ise mevcut bir sistemin gerçek ağ bileşenlerini almak ve bunları grafiksel ya da metinsel olarak göstermek için kullanılır. Ayrıca ağ keşif araçları, mevcut ağları izleme, planlama ve değişiklikleri uygulamadan önce ağ benzetimi için giriş verilerini üretmede yardımcı olabilir. Bu araçlara örnek olarak Fremont, Scotty, NetMap, Big Brother, LANsurveyor, NetView, Nessus, Nmap, Open view ve Intermapper gösterilebilir [5, 12].

- Benzetim araçları modellemenin ötesinde bir ağ modelinin dinamik davranışının (paket geçişi, bağlantı hataları, TCP protokolü gibi) benzetiminde kullanılırlar. Ağ operasyonlarının gerçekçi senaryolarla analizi için ilk yıllarda analitik yöntem ve araçlar (örneğin ağ kuyruk teorisi gibi) yeterli olurken, günümüzde ağ donanımlarının karmaşıklığı ve hızı, protokollerin karmaşıklığı, geniş ölçekli ağ yapısı, benzetimi gerekli kılmaktadır [5, 12, 49].



Şekil 5.1. Ağ Tasarım ve Benzetim Araçlarının Sınıflandırılması

Bilgisayar ağlarının tasarım ve analizinde artık vazgeçilmez bir araç olan ağ benzetim araçlarının sayısı günümüzde yüzleri bulmuştur ve bu benzetim araçlarını, Şekil 5.1’de gösterildiği gibi akademik / eğitim (ücretsiz) amaçlı, ticari amaçlı ve genel amaçlı olmak üzere üç başlık altında toplamak mümkündür.

### 5.3. Topoloji Üretim Araçları

Ağ topoloji yapısının iyi modellenmesi, iletişim tekniklerinin analizi ve geliştirilmesi için esas teşkil etmektedir. Ağ iletişimde, verimli protokol tasarımında, problemlerin (yönetim, yönlendirme, kaynakların kullanılması vb.) çözümünde,

benzetim için doğru modelin kurulmasında ve hata toleransı çalışmalarında topolojinin önemi çok büyüktür [12].

Topoloji üretim araçları, farklı algoritmaları temel alarak küçük veya geniş ölçekli topolojiler üretmeye ve araştırmacıların ağ performans analizlerine yardım eder. Waxman, Tiers, Transit-stub, GT-ITM, Inet, PRLG, BRITE ve KOM ScenGen gibi araçlar, topoloji üretim araçlarına örnek olarak verilebilir. Bu araçlardan bazıları (KOM ScenGen, BRITE gibi) ağ topoloji üretimi ve trafik oluşumu için otomatik yazılım bileşenlerine sahip olduğu gibi diğer araçların ürettiği topolojileri de kullanabilir. Tablo 5.1’de topoloji üretim araçlarının karşılaştırılması sunulmuştur [12, 88].

Tablo 5.1. Topoloji üretim araçlarının karşılaştırılması

Araçlar	Topoloji Ölçeği	İşletim Sistemi	Programlama Dili
<b>Waxman</b>	Büyük	Unix	C
<b>Tiers</b>	Büyük	Unix	C++
<b>Transit-stub</b>	Büyük	Unix	C
<b>GT-ITM</b>	Çok Büyük	Unix	C
<b>Inet</b>	Çok Büyük	Unix	C
<b>PLRG</b>	Büyük	Unix	C
<b>BRITE</b>	Çok Büyük	Unix, Windows	C++, Java
<b>KOM ScenGen</b>	Büyük	Windows, Unix	Java

Ağ topoloji üretim araçları, gerçekçi ağ benzetim senaryoları, ağ yönetimi, ağ konum belirleme ve performans karakteristikleri için yaygın olarak kullanılmaktadır [89]. Geniş ölçekli benzetim deneylerinde topoloji üretimi önemlidir. Çünkü

- Geniş ölçekli gerçek ağların tasarımı çok zordur,
- İnternet ölçekli modeller için algoritma geliştirme işlemi uzun ömürlü ve daha verimli ağ tasarımları için gereklidir [90].

Genel topoloji üretim araçlarından beklenen özellikler aşağıdaki gibi sıralanabilir [91]:

- Temsil edilebilirlik: Gerçekçi sanal topolojiler üretmeli ve İnternet topolojilerini farklı açılardan doğru yansıtmalıdır.
- Kapsamlılık: Tek bir topoloji üretisinde mümkün olduğunca fazla üreteç modeli barındırmalıdır.
- Esneklik: Geniş ölçekte topoloji üretebilmeli, minimum ve maksimum düğüm sınırlaması getirmemelidir.
- Verimlilik: Makul seviyede CPU ve bellek kullanımı ile çok büyük ölçeklerde topoloji üretebilmelidir.
- Genişletilebilirlik: Kullanıcı tarafından kolaylıkla yeni modellerin eklenebilmesine izin vermelidir.
- Kullanıcı dostu: Standart kullanıcı ara yüzü kullanım ilkelerini takip etmelidir. Kullanıcılar üretim aracını kolaylıkla kullanabilmelidir.
- Birlikte çalışabilirlik: Yaygın olarak kullanılan ağ benzetim araçları (ns-2, OPNET, OMNET++ gibi) için ara yüzlere sahip olmalıdır ve ağ benzetim araçlarına kolay entegre edilebilmelidir.
- Sağlamlık: Geniş hata algılama yeteneğine sahip olmalı ve verimli çalışmalıdır.

#### **5.4. Benzetim Araçları**

Ağ benzetim araçları, araştırmacılara ağ tasarım ve geliştirme sürecinde yardımcı olmak üzere geliştirilmişlerdir. Çok sayıdaki ağ benzetim aracı arasında karşılaştırma yapılmış ve elde edilen sonuçlar Tablo5.2'de sunulmuştur [5, 4, 7, 12, 28, 44, 92].



Tablo 5.2. Yaygın olarak kullanılan ağ benzetim araçları ve özellikleri

Özellik	ns-2	PDNS	OPNET	OMNeT++	Qual Net	SSF Net	J-Sim	Glomo sim	D-DEVSNET
<b>Amaç</b>	Eğitim, araştırma	Eğitim, araştırma	Ticari, araştırma	Eğitim, araştırma	Ticari	Ticari, araştırma	Eğitim	Genel kablosuz ağlar	Eğitim, araştırma
<b>Model kütüphanesi</b>	İyi	İyi	İyi	İyi	Orta	Zayıf	Orta	Orta	Zayıf
<b>Analiz</b>	Orta	Orta	<b>Çok İyi</b>	Zayıf	Zayıf	Zayıf	Zayıf	İyi	<b>Çok İyi</b>
<b>Esneklik</b>	Orta	Orta	İyi	<b>Çok iyi</b>	İyi	İyi	İyi	Orta	<b>Çok iyi</b>
<b>Belgelendirme</b>	Orta	Orta	<b>Çok İyi</b>	İyi	Zayıf	Zayıf	Zayıf	Orta	Orta
<b>Kullanım kolaylığı</b>	Zor	Zor	<b>Kolay</b>	Orta	Orta	Zor	Zor	Zor	<b>Kolay</b>
<b>Kullanıcı arayüzü</b>	Zayıf	Zayıf	İyi	İyi	İyi	İyi	Orta	İyi	İyi
<b>Ölçeklenebilirlik</b>	Orta	<b>Çok iyi</b>	Orta	İyi	<b>Çok iyi</b>	<b>Çok iyi</b>	Orta	İyi	İyi
<b>Performans</b>	İyi	<b>Çok İyi</b>	Orta	Orta	İyi	<b>Çok İyi</b>	İyi	Orta	<b>Çok İyi</b>
<b>Ağ modeli</b>	WAN	<b>Büyük ölçek</b>	LAN, WAN, Uydu	LAN, MAN, Kablosuz	Kablosuz, Uydu	<b>Büyük ölçek</b>	LAN, Kablosuz	Kablosuz	LAN, MAN, WAN
<b>Lisans</b>	<b>Açık kaynak</b>	<b>Açık kaynak</b>	Ticari	<b>Açık kaynak</b>	Ticari	<b>Açık kaynak</b>	<b>Açık kaynak</b>	<b>Açık kaynak</b>	<b>Açık kaynak</b>
<b>Program. dili</b>	C++, Tcl	C++, Tcl	C++	C++	C++	Java, C++	Java, Tcl	Java, C	Java
<b>Platform</b>	Unix, Linux	Unix	<b>Linux, Win.</b>	<b>Linux, Mac, Win.</b>	<b>Linux, Win.</b>	<b>Linux, Win.</b>	<b>Linux, Mac, Win.</b>	<b>Win., Unix</b>	<b>Linux, Win.</b>
<b>Nesne Yönelimli</b>	Orta	Orta	İyi	Orta	Orta	<b>Çok İyi</b>	<b>Çok İyi</b>	Orta	<b>Çok İyi</b>
<b>Web Erişim</b>	Yok	Yok	Yok	Yok	Yok	Yok	İyi	Yok	İyi

Tablo 5.2’de benzetim araçlarının öne çıkan (avantajlı) özellikleri koyu renkte gösterilmiştir. Farklı özellikler / amaçlar doğrultusunda, tablodaki benzetim araçları incelendiğinde, bütünüyle mükemmel bir benzetim aracının olmadığı söylenebilir. Ayrıca her benzetim aracı farklı bir amaç doğrultusunda geliştirilmiş veya geliştirilmektedir. Örneğin OPNET ve QualNet’te ticari amaç, PDNS [93] ve SSFNet [94]’te geniş ölçekli İnternet trafiği, GloMosim ve ns-2’da kablosuz ağ benzetimi çalışmalarının daha fazla öne çıktığı görülmektedir.

Begg ve arkadaşları [15] tarafından yapılan çalışmada, özel amaçlı geliştirilen benzetim araçlarının kullanıcı taleplerini karşılayabileceği vurgulanmıştır. DEVS-Suite benzetim aracında ağ benzetimleri için belli amaçlar doğrultusunda özelleştirilerek kullanıcı gereksinimlerini karşılayabilecek bir araca dönüştürülebilir.

Bu bölümde ağ benzetim araçları kullanım amaçlarına göre eğitim, ticari ve genel amaçlı olmak üzere kısaca tanıttıktan sonra, özellikle yaygın olarak kullanılan ağ benzetim araçları (OPNET, ns-2 ve OMNET++) detaylandırılmıştır.

Eğitim ve araştırma amaçlı olarak akademik çevreler (üniversiteler) tarafından geliştirilen eğitim amaçlı ağ benzetim araçlarına; ns-2, PDNS, OMNET++, Netsim (M.I.T Üniversitesi, Network Simulator), GTNetS (Georgian Tech Üniversitesi, Network Simulator), WIPSIM (Aalborg Üniversitesi, Wireless IP Simulator) ve J-Sim [95] örnek olarak gösterilebilir.

Ticari amaçlı ağ benzetim araçlarına; OPNET, QualNet (Glomosim'in ticari sürümü), COMNET III, REAL (REalistic And Large) ve SSFNet örnek olarak gösterilebilir.

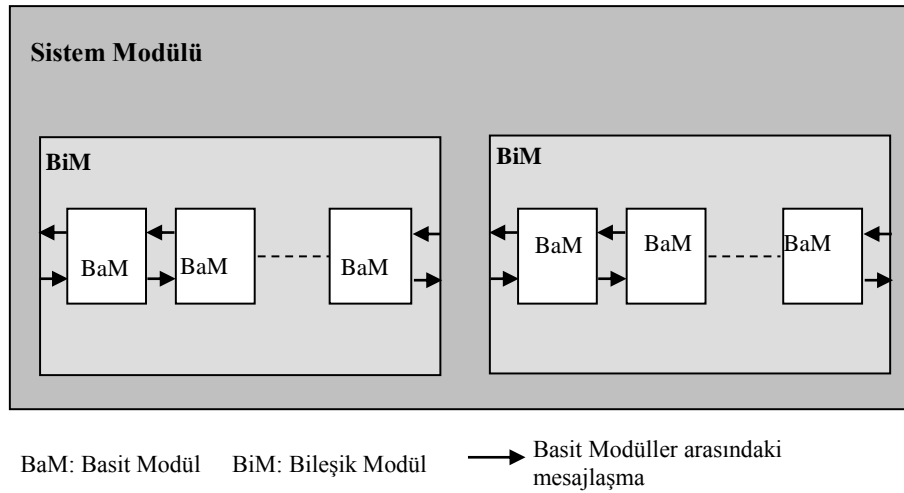
#### 5.4.1. OMNET++

OMNET++ (Objective Modular Network Testbed in C++), nesne yönelimli (object-oriented), modüler bir ayrık olay ağ benzetim aracıdır ve bu yazılım aşağıda maddeler halinde verilen süreçlerin benzetiminde kullanılabilir.

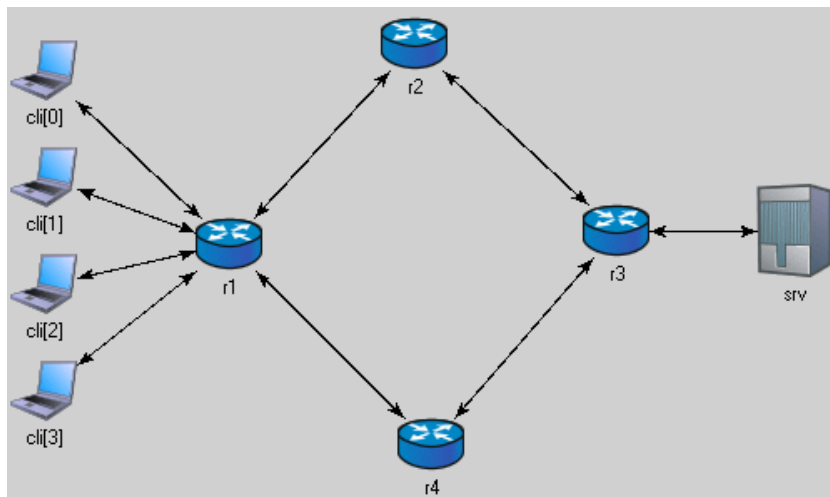
- Haberleşme trafiğinin modellenmesi,
- İletişim protokollerinin modellenmesi,
- Çok işlemcili ve diğer dağıtık donanım sistemlerinin modellenmesi,
- Donanım yapılarının incelenmesi,
- Karmaşık sistemlerin başarımlarının analizlerinin değerlendirilmesi,
- Ayrık olay yaklaşımının elverişli olduğu diğer sistemlerin modellenmesi.

OMNET++ yazılımında bir ağ modeli, Şekil 5.2'de görüldüğü gibi iç içe geçmiş modüllerin birleşiminden meydana gelmektedir. En üst seviyedeki modül, sistem modülü ya da ağ olarak isimlendirilir. İç içe geçen modüllerin derinliği kullanıcıya bağlıdır ve bu sayede karmaşık sistemlerin modelleri kolaylıkla gerçekleştirilebilir. Modüller, basit ve bileşik olmak üzere iki kategoriye ayrılmaktadır. Bir basit modül,

modellenmek istenen parçanın davranışlarını tanımlayan C++ dosyasıyla ilişkilendirilir ve bu dosya kullanıcı tarafından OMNET++ benzetim sınıf kütüphaneleri kullanılarak yazılmaktadır. Bileşik modüller ise basit modüllerin birleşiminden meydana gelmektedir ve doğrudan bir C++ dosyasıyla ilişkili değildir. Modüller kendi aralarından mesajlar yardımıyla haberleşmekte ve benzetim zamanı, bir modül kendisinden veya başka bir modülden mesaj aldığı anda ilerlemektedir.



Şekil 5.2. OMNET++ Modül Yapısı

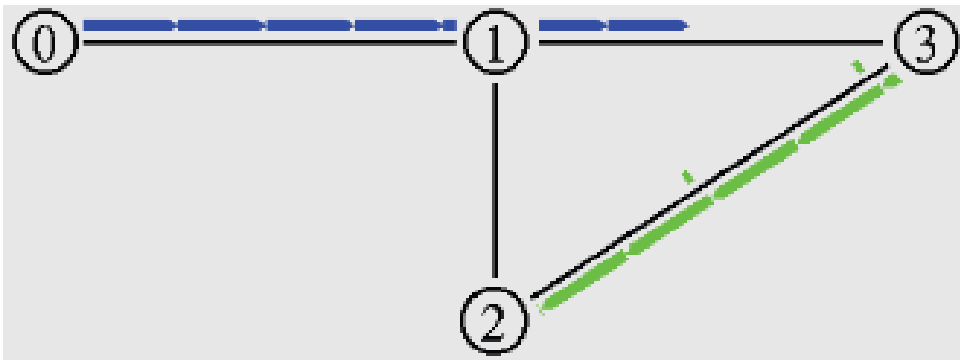


Şekil 5.3. Yönlendirici Dğümleri İçeren Basit Bir Ağ Yapısının OMNET++ ile Gerçekleştirimi

OMNET++, zamanlama işlemlerini ‘self message’ adı verilen ve düğümün kendisine gönderdiği mesajlar yardımıyla gerçekleştirmektedir. Modüllerin yapısı ve arabirimleri, Ağ Tanımlama Dili (Network Description Languages – NED) ile oluşturulmakta ve benzetim parametreleri bir başlangıç dosyası (.ini) ile kolaylıkla ayarlanabilmektedir [8, 96]. Şekil 5.3’te dört düğümlü basit bir ağ yapısının OMNET++ ile gerçekleştirimi gösterilmiştir.

#### 5.4.2. ns-2 (Network Simulator-2)

NS, İlk olarak 1989 yılında REAL ağ benzetim aracının bir türeği olarak geliştirilmeye başlasa da, 1995 yılında DARPA tarafından U.C.Berkeley / LBL / Xerox PARC işbirliği ile bir VINT (Virtual Inter-Network Testbed) projesi olarak desteklendi. Günümüzde ise ISI (Information Sciences Institute) ve Southern California Üniversitesi tarafından geliştirilmektedir. ns-2, ağ araştırmaları için geliştirilen IP tabanlı protokolleri (TCP, UDP), trafik kaynaklarını (FTP, Telnet, Web, CBR, gibi), DropTail, RED, CBQ gibi yönlendirici kuyruk yönetimini ve farklı yönlendirme algoritmalarının benzetiminde kullanılan açık kaynak kodlu, ayırık olay tabanlı bir ağ benzetim aracıdır. Hem C++ hem de Tcl/Otcl (Tool Command Language / Object Oriented Tcl) dillerini kullanır. Ayrıca ayrı bir görsel editör (NAM-Network Animatör) kullanır [6]. Şekil 5.4’de dört düğümlü basit bir ağ yapısının ns-2 ile gerçekleştirimi gösterilmiştir.



Şekil 5.4. Yönlendirici Düğümleri İçeren Basit Bir Ağ Yapısının NAM Editörde Görüntüsü

Yaygın olarak kullanılan ns-2 benzetim aracının zayıflıklarını, geniş ölçekli ağları modelleyememesi, kullanım / öğrenim zorluğu, gerçek zamanlı benzetim yapılamaması, platform bağımsızlığının olmaması şeklinde sıralayabiliriz.

#### **5.4.3. OPNET (OPTimised Network Engineering Tool)**

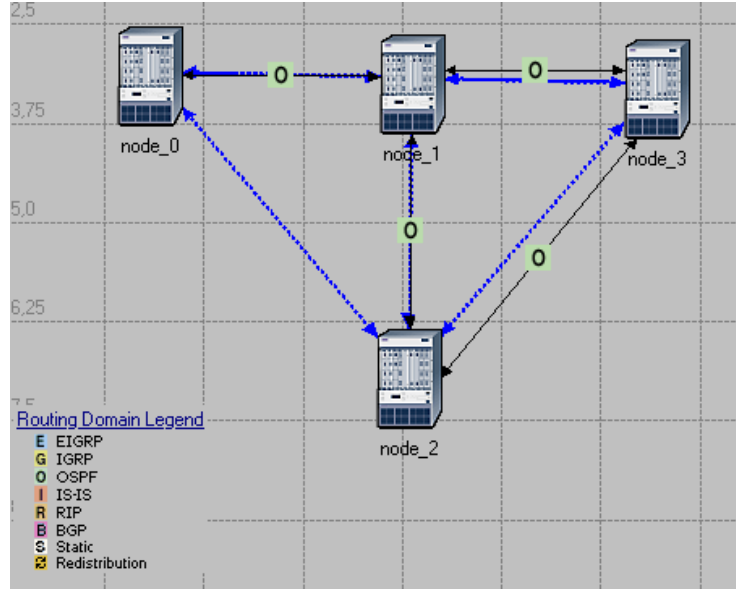
OPNET, 1987 yılında geliştirilen ilk ticari ağ benzetim aracıdır. Çok kolay ağ modellerinin kurulabildiği, grafiksel ara yüze sahip, kullanışlı, endüstride yaygın olarak kullanılan çok güçlü bir ayrık olay ağ benzetim aracıdır. Ticari bir yazılım olmasına rağmen, OPNET programının akademik çalışmalar için İnternette ücretsiz olarak temin edilebilen ‘OPNET IT Guru Academic Edition (AE)’ sürümü de mevcuttur.

OPNET programı ile sistemin davranışı ve ayrık olay benzetimi gerçekleştirilerek analiz yapılabilir. OPNET benzetim programında üç seviye bulunmaktadır. Bunlar; ağ, düğüm ve süreç şeklindedir. Bu seviyeler görsel düzenleyiciler kullanılarak geliştirilebilmektedir. Her bir seviye için düzenleyici editörler bulunmaktadır. Program aynı zamanda benzetim parametrelerini düzenlemek ve veri analizi yaparak grafik oluşturmak için de araçlar içermektedir.

Ağ yapısı, düğüm ve süreç modelleri bir proje dosyasına dâhil olan senaryolar halinde oluşturulmaktadır. Tasarım tamamlandığında benzetim aracı yardımıyla toplanacak istatistikler belirlenerek çalıştırılır. Program dâhilindeki analiz aracı sayesinde elde edilen veriler, istenen grafik türünde görüntülenebilmektedir. Birden fazla senaryoya ait verilerin aynı grafik üzerinde gösterilerek karşılaştırılması da mümkündür. Düğüm modeli ve süreç modeli oluşturma amaçlı editörler yardımıyla kullanıcı tanımlı düğümler ve protokoller oluşturulabilmektedir. Profil tanımlamaları ve uygulama tanımlamaları editörler yardımı ile değiştirilebilmektedir [10].

OPNET’in uygulama alanları olarak, standart LAN-WAN performans modelleme, ağ planlaması, iletişim mimarileri ve protokol araştırma geliştirme, ayrık duyurga ve

kontrol ağı, mobil paket radyo ağı, uydu ağı gösterilebilir. Şekil 5.5'te dört düğümlü basit bir ağ yapısının OPNET ile gerçekleştirimi gösterilmiştir.



Şekil 5.5. Yönlendirici Düğümleri İçeren Basit Bir Network Yapısının OPNET Görüntüsü

#### 5.4.4. DEVS-Suite

Genel amaçlı diğer bir ifade ile özelleştirilebilen araçlara, GloMoSim [92] ve DEVS-Suite [10] gösterilebilir. GloMoSim, genel kablosuz mobil ağ sistemleri için ölçeklenebilir bir benzetim ortamı sunar.

Özellikle açık kaynak kodlu genel amaçlı benzetim araçları belli bir amaca yönelik olarak özelleştirilebilir. Bizde bu çalışmamızda genel amaçlı DEVS-Suite benzetim aracını, dağıtık geniş ölçekli bir ağ benzetim aracı olarak geliştirdik. Genel amaçlı DEVS-Suite benzetim aracı Bölüm 2'de anlatılmıştı. Burada özelleştirilmiş DEVS-Suite ağ benzetim aracının (D-DEVSNET) görünümü ve açıklaması Şekil 5.6'da verilmiştir. DEVS-Suite benzetim ortamı kullanılarak geliştirilen dağıtık, geniş ölçekli ağ benzetim aracının protokol modellemesi, analizleri ve geçerlilik testleri Bölüm 7.4'te açıklanacaktır.

The image displays the D-DEVSNET simulation environment. The main window shows a network topology with several components:

- Server Distributed Network AS-2:** Contains a Server (Idle,  $\sigma = 1,000$ ) and an Experimental Frame (EventGenr Initialization,  $\sigma = \text{infinity}$ ; EventTransd listCreating,  $\sigma = 20,000$ ).
- Network:** Consists of 10 routers (gRouter0 to gRouter9) and 10 NICs (NIC0 to NIC9). Each router is in a "flooding" state with  $\sigma = 1,000$ .
- Control Panels:**
  - Topoloji üretici erişim butonları:** Buttons for generating server and client topologies.
  - Benzetim de yer alan atomik ve birleşik modeller:** A list of components in the simulation.
  - Benzetim ortamının çalıştırılması / durdurulması veya adım adım takip edilebilmesine yardımcı olan butonlar:** Buttons for running, pausing, and stepping through the simulation.
  - Benzetim aracındaki her bir yönlendirici düğüm üzerine fare odaklanması ile çıkan açıklama menüsü:** A detailed view of a router's state, including AS number, queue state, and BGP table.
- Console:** Displays simulation logs, including messages from the Panser router and the completion of topology generation.
- Simulator Control:** A panel with buttons for Run, Step, Request Pause, Reset, and Enable Governor.
- Bottom Panel:** Shows the current simulation state (Pause), time of last event (0.0), and time of next event (1.0).

Şekil 5.6. D-DEVSNET Ağ Benzetim Aracının Görünümü

## **BÖLÜM 6. YENİ BİR DAĞITIK BENZETİM ALGORİTMASININ GELİŞTİRİLMESİ VE D-DEVSNET AĞ BENZETİM ARACININ TASARIMI**

### **6.1. Giriş**

Bilindiği üzere modelleme ve benzetim, özellikle analitik metotların uygulanamadığı geniş ölçekli ağ sistemleri ve bu ağ sistemlerinde çalışan yönlendirme protokollerinin tasarım ve analizinde önemli rol oynamaktadır [44, 92].

Ağ tasarımı (topolojik yapısı, protokolleri, trafik akış denetimi, bağlantı teknolojisi, yönlendirme algoritmaları, vb. ) oldukça karmaşık ve zor bir süreçtir. Tasarımcılar bu karmaşıklığı azaltmak için, belli bir soyutlama seviyesinde modelleme ve benzetim tekniklerinden yararlanırlar [12].

İnternet teknolojilerinin baş döndürücü hızla büyümesi onun ölçeklenebilirliği ve performans değerlendirmesi gibi sorunlarını da beraberinde getirmiştir. Düğüm sayısındaki fazlalık benzetim performansını, benzetim çalıştırma zamanını ve ölçeklenebilirliği zorlaştırmaktadır. Daha öncede ifade edildiği gibi hâlihazırda kullanılan ve geliştirilen birçok ağ benzetim aracı vardır, fakat bunların boyut, teorik altyapı eksikliği ve modellenen ağın karmaşıklığı gibi pek çok sorunları vardır.

Bu çalışmanın öncelikli amacı, DEVS yaklaşımını kullanan DEVS Suite benzetim aracı üzerinde paralel ve dağıtık çalışabilen, yüksek performanslı ve ölçeklenebilir bir ağ benzetim aracını tasarlamak ve gerçekleştirmektir.

Bu çalışmada büyük modelleri çalışabilmek için yeni bir dağıtık benzetim algoritması geliştirildi. Geniş ölçekli ağların benzetiminin ölçeklenebilirliği ve performansı için paralel ve dağıtık, ölçeklenebilir, ayrık olaylı bir ağ modeli



tanımlandı ve DEVS yaklaşımı kullanılarak geliştirilen benzetim modelleri ile önemli İnternet protokolleri birleştirildi. Geliştirilen modeller ve İnternet protokolleri geliştirilen D-DEVSNET ağ benzetim ortamı altında uygulanmıştır.

Paralel DEVS birleşik model tanımı kullanılarak, DEVS-Suite modelleme ve benzetim ortamı altında gerçekleştirilen geniş ölçekli, dağıtık mimarili, DEVS tabanlı ağ benzetim aracı ‘D-DEVSNET’ olarak isimlendirilmiştir. Bölüm 5.4’te diğer ağ benzetim araçları ile genel bir karşılaştırması yapılan D-DEVSNET’in, doğrulama ve geçerlilik testleri ise Bölüm 7.4’te yapılmıştır. Geliştirilen aracın, ölçeklenebilirlik, esneklik, taşınabilirlik ve kullanım kolaylığı gibi özellikleri ile özellikle ağ protokollerinin eğitim ve araştırmasında benzerlerine üstünlük sağladığı görülmüştür.

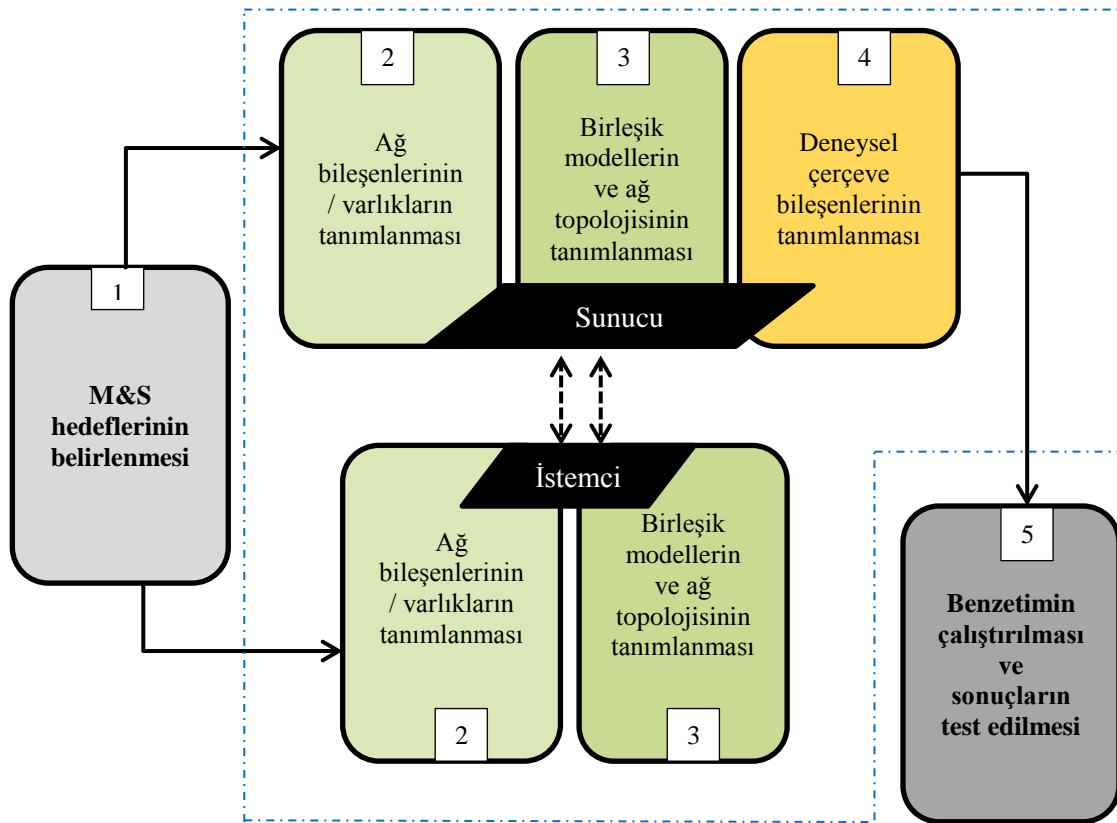
DEVS-Suite, bir benzetim dili öğrenme gereksinimini ortadan kaldıran yüksek performanslı ve kullanıcı dostu grafiksel bir ara yüze sahiptir. DEVS-Suite benzetim aracı kişisel bilgisayarlar üzerinde çalıştırılabileceği gibi DEVS-Suite Web Start ile çevrimiçi web ortamında da çalıştırılabilir. Böylece gerçekleştirilen benzetim aracı, uzaktan eğitim uygulamalarında da rahatlıkla kullanılabilir.

## **6.2. D-DEVSNET Ağ Benzetim Aracı Geliştirme Süreci**

DEVS modelleme yaklaşımı, son zamanlarda birçok mühendislik (donanım tasarımı, haberleşme sistemleri, üretim sistemleri) ve bilim (biyoloji, sosyoloji) dallarında kullanılmaya başlamıştır [20, 21, 28, 97, 98]. DEVS modelleme yaklaşımının, hiyerarşik / modüler bir yapıyı ve dağıtık çalışmayı desteklemesi, karmaşık geniş ölçekli sistemlerin (atomik ve bileşik modellerden oluşan) modellenmesinde kolaylıklar sağlamaktadır.

Bir ağ benzetim aracı, kullanıcılara ağ topolojisini görsel olarak izlemeyi, değişik senaryolar hazırlamayı, düğümleri özelleştirmeyi ve sonuçları analiz etmeyi sağlamalıdır. Grafiksel kullanıcı ara yüzüne sahip olan benzetim aracı, kullanıcılarına üzerinde çalışılan ağ modelini görselleştirme ve izleme imkânı verir. Bölüm 5’te

ayrıntılı olarak anlatıldığı üzere ns-2, OPNET, SSFNet, GloMoSim ve OMNeT++ gibi benzetim araçları bilgisayar ağ arařtırmalarında yaygın olarak kullanılmaktadır. Bu araçların karmařıklığı yönetebilme, performans, ölçeklenebilirlik, görselleřtirme ve teorik sistem tasarım eksikliđi gibi bazı dezavantajları vardır. Mesela ns-2'nin görselliđi zayıf olmasına rađmen ölçeklenebilirliđi iyidir, OPNET'in ise ölçeklenebilirliđi zayıf olmasına rađmen görselliđi iyidir. OMNeT++ i geniřletmek zordur, yüksek oranda özelleřtirme ister ve teorik sistem yaklařımı eksiklikler içerir. Ayrıca tüm bu ağ benzetim araçları tek bir bilgisayar üzerinde çalıřabildiđi için geniř ölçekli ağları modellemek imkânsızdır.



Şekil 6.1. İstemci - Sunucu mimarili dađıtık DEVS tabanlı ağ modelleme ve benzetim süreci

Dađıtık geniř ölçekli ağ sistemlerinin tasarım ve analizini sistematik bir şekilde gerçekleřtirmek için bu sistemlerin modelleme ve benzetimi aşamalara bölünür. D-DEVSNET olarak adlandırılan dađıtık geniř ölçekli DEVS tabanlı ağ benzetim ortamının gelişim/tasarım süreci Şekil 6.1'de gösterilen aşamalardan oluşmaktadır.

Başlangıç aşamasında, modelleme ve benzetim hedefleri belirlendi. Modelleme hedefleri, modellenen sistemin tasarımında, yönetiminde ve kontrolünde tanımlanan rollerle ilişkilidir. Hedefler ifadesi, model tasarlama işleminin belirli problemler üzerine odaklanması vazifesini görür. Önceden hedeflerin bilinmesi, bu hedeflere uygun deneysel çerçevenin hazırlanmasını kolaylaştırır.

İkinci aşamada, ağ tasarımı konusunda karar vermeyi sağlayan bir model ve ağ benzetim aracı yazılımı DEVS-Suite ve Java programlama dili ile geliştirildi. Bu aşamada benzetimi oluşturan temel ağ bileşenleri, varlıkları ve parametreleri istemci ve sunucu taraflı olmak üzere ayrı ayrı tanımlanır. Temel ağ bileşenlerinin tanımlanmasında, bir bağlantı durumu protokolü olan OSPF düğümlere yerleştirildi. OSPF protokolünü seçmemizin sebebi geniş ölçekli ağlarda RIP yerine OSPF yönlendirme protokolünün kullanılmasıdır. Ayrıca otonom sistemler arası yönlendirme protokolü olarak BGP seçilmiş ve uygulanmıştır.

Üçüncü aşamada, geliştirilen varlıklar ve düğümler / yönlendiriciler birleştirilir. Bu bileşenler DEVS-Suite modelleme ve benzetim ortamında birbiriyle bağlanarak değişik topolojiler ve ağ yapıları meydana getirilebilir.

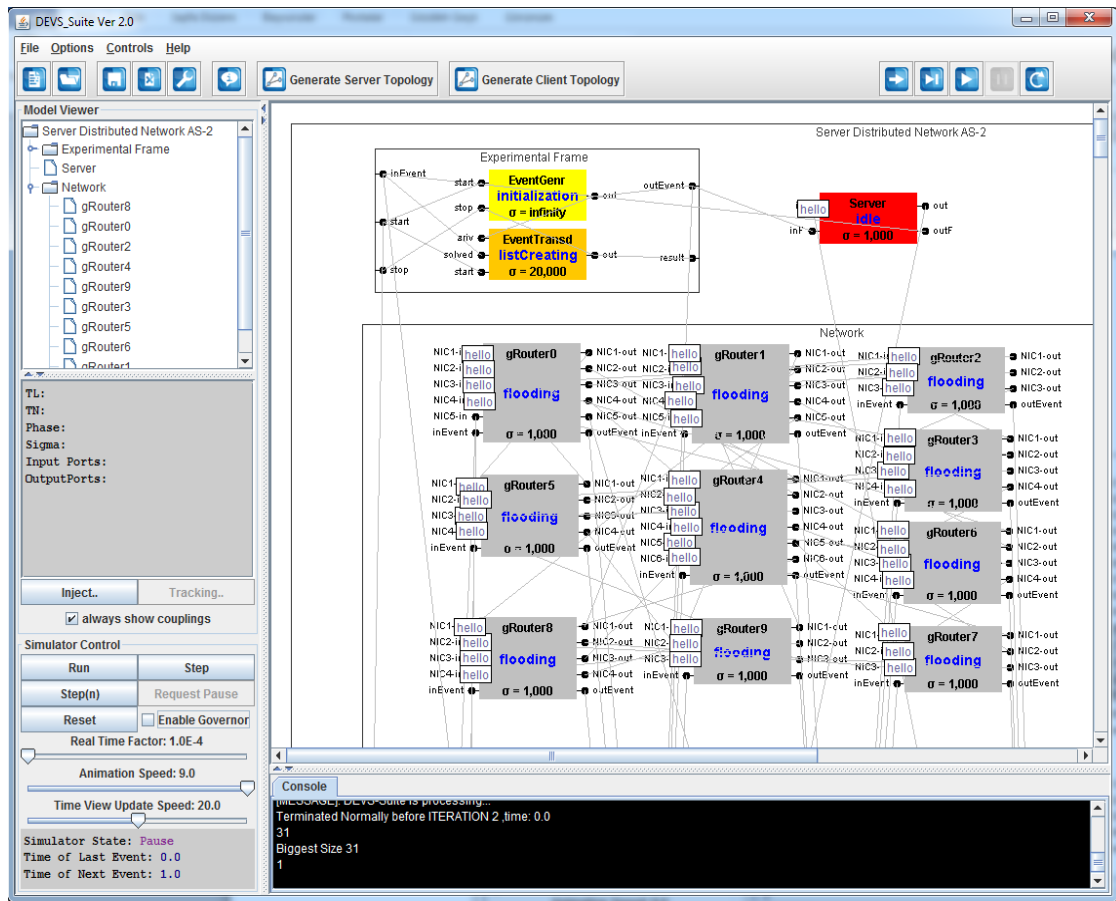
Dördüncü aşamada, modelleme ve benzetim hedeflerine ve geliştirilen modellere uygun bir deneysel çerçeve geliştirildi. Deneysel çerçeve, geliştirilen modelin bir takım koşullar altında test edilmesine ve gözlem yapılmasına yardımcı olur.

Programlama aşamasında öncelikle istemci-sunucu modelleri ve genel bir ağ modelinin temel bileşenleri (düğümler / yönlendiriciler) belli bir soyutlama seviyesinde DEVS yaklaşımı kullanılarak oluşturuldu. OSPF ve BGP yönlendirme protokollerinin çalıştığı yüksek performanslı, modüler ve hiyerarşik yapıda bir ağ benzetim aracı gerçekleştirildi. Daha sonra ise geliştirilen benzetim aracına BRITE topoloji üretim aracı ile entegre edilerek büyük ölçeklerde topolojiler geliştirildi.

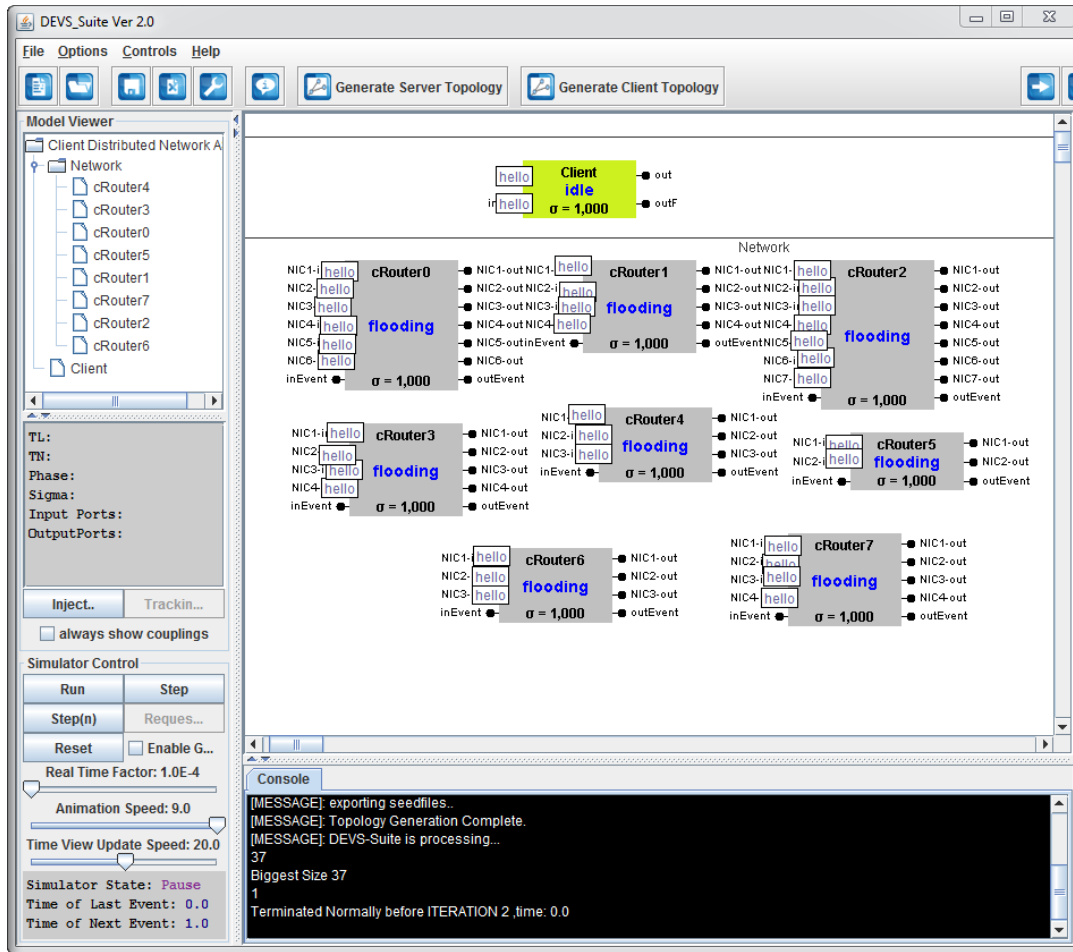
Son aşamada, benzetim aracı çalıştırılır ve sonuçlar DEVS-Suite ortamında gözlemlenir / izlenir. Sonuçlar kabul edilebilir bir aralıkta ise benzetim işlemi

sonlandırılır. Aksi halde, tekrar başa dönlür ve geliştirilen modellerin parametreleri ayarlanır. Böyle bir süreçte ileri geri hareket edilerek en uygun model geliştirilmiş olur [39].

Geliştirilen ağ benzetim aracının değişik ölçeklerdeki farklı ağ benzetim araçları (ns-2, OPNET gibi) ile karşılaştırması yapılarak geçerlilikleri test edildi. Tasarlanan ve gerçekleştirilen dağıtık, geniş ölçekli, D-DEVSNET ağ benzetim aracının ölçeklenebilirlik, esneklik, taşınabilirlik ve kullanım kolaylığı gibi özellikleri ile özellikle ağ protokollerinin eğitim ve araştırmasında benzerlerine nazaran birçok üstün özelliklere sahip olduğu söylenebilir. Şekil 6.2a ve 6.2b' de bu ağ benzetim aracının sunucu ve istemci tarafının ekran görüntüleri verilmiştir.



Şekil 6.2a. D-DEVSNET ağ benzetim aracının sunucu tarafı ekran görüntüsü



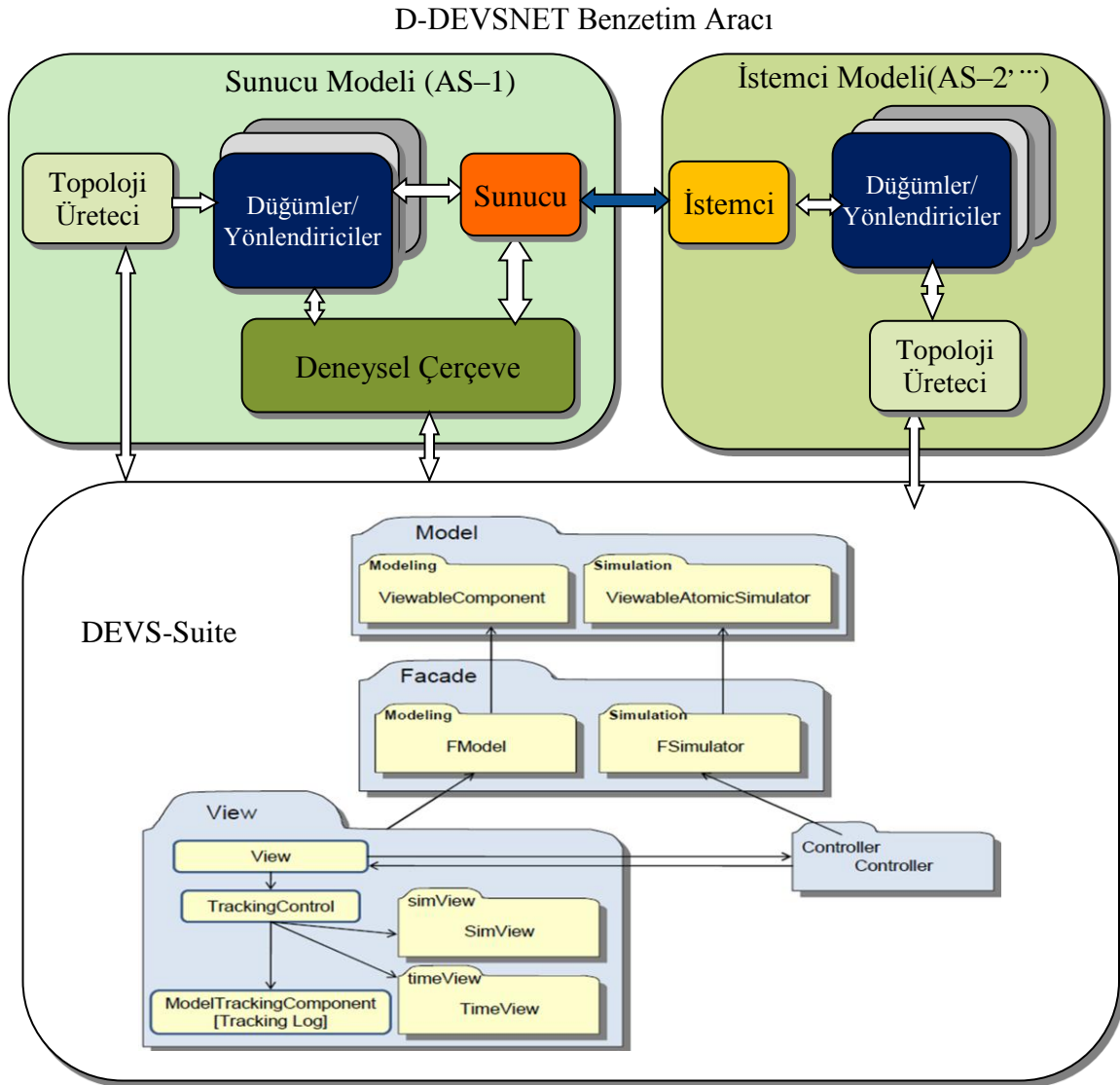
Şekil 6.2b. D-DEVSNET ağ benzetim aracının istemci tarafı ekran görüntüsü

### 6.3. Ağ Bileşenlerinin Tanımlanması ve Tasarımı

Basit bir ağ modeli sadece düğümler ve veri hatları ile temsil / karakterize edilebilir. Geniş ölçekli DEVS tabanlı ağ benzetim modelindeki temel bileşenlerin tasarımına, paralel DEVS atomik modeli kullanılarak başlanmıştır. Ağ benzetim aracını modelleme amacına yönelik olarak oluşturulan düğümler ve bu düğümlerin haberleşmesini sağlayan diğer nesnelere (IP paketleri, vb.), ‘temel ağ bileşenleri’ olarak tanımlanır. Düğüm ve diğer ağ bileşenleri, ağ modelinin davranışını (çıkış zamanı gibi) belirlemek için kullanılabilir. Veri hatları(linkler) ise ağdaki anahtarlama elemanları arasında bir haberleşme yolu sağlar. Tanımlanan bu düğüm ve dinamik bağlantılar ile değişik ağ arabirim araçları (yönlendirici, dağıtıcı gibi) ve

topolojileri geliştirilebilir. Daha sonra bu bileşenlerin bir araya gelmesi ile ‘DEVS birleşik ağ modeli’ oluşturulur [28, 39].

Burada kullanılan atomik ve birleşik modeller, Bölüm 2’de tanımlanan ‘Paralel DEVS yaklaşımı’ kullanılarak tanımlandı ve ‘DEVS-Suite’ modelleme ve benzetim ortamında tasarlandı. Geliştirilen ağ benzetim aracında yapılan test ve deneyler için, bir modelin deney yapılacağı ve gözlemleneceği şartları tanımlayan ‘DEVS deneysel çerçeve’ kavramından istifade edilmiştir. Bölüm 2’de ayrıntılı olarak açıklandığı üzere geliştirilen ağ benzetim aracının deneysel çerçevesi bir paket üretici ve dönüştürücüden oluşmaktadır.



Şekil 6.3. D-DEVSNET benzetim aracının kavramsal modeli ve bileşenleri

Geliştirilen geniş ölçekli istemci / sunucu tabanlı dağıtık mimariye sahip D-DEVSNET ağ benzetim aracının sınıf yapısı Ek A'da verilmiştir. D-DEVSNET ağ benzetim aracının kavramsal modeli ise, sunucu ve istemci modeli içerisindeki ağ bileşenleri ile birlikte Şekil 6.3'te gösterilmiştir. Şekil 6.3'te görüldüğü gibi sunucu modelinin ağ bileşenlerini, otonom sistem (AS-1) içerisindeki düğümler / yönlendiriciler, sunucu (server), deneysel çerçeve ve topoloji üretici oluşturmaktadır. İstemci modelinin ağ bileşenleri ise otonom sistem (AS-2) içerisindeki düğümler / yönlendiriciler, istemci (client) ve topoloji üretici oluşturmaktadır. Aynı zamanda hem sunucu tarafı hem de istemci tarafında temel ayrık olay işlemcisi olan DEVS çekirdeği yer almaktadır.

Geliştirilen ağ modeli üç farklı tipte atomik modele, bir birleşik modele ve deneysel çerçeveye sahiptir. Atomik düğüm modelleri; sunucu düğüm atomik modeli, istemci düğüm atomik modeli ve yönlendirici atomik modelinden oluşurken, birleşik düğüm modelleri; otonom sistemden oluşmaktadır. Ayrıca deneysel çerçeve modeli de üreteç ve dönüştürücü atomik modellerine sahiptir.

Ağ bileşenlerini atomik modeller, birleşik modeller ve deneysel çerçeve olmak üzere ayrı ayrı başlıklar altında inceleyeceğiz.

### 6.3.1 Atomik modeller

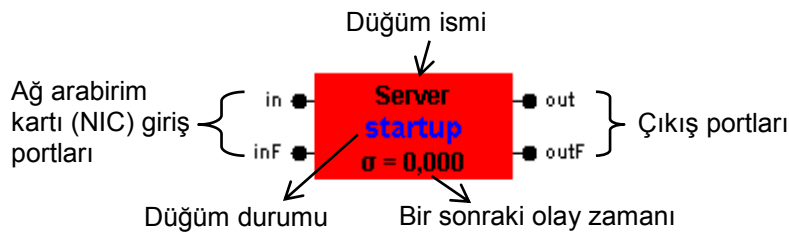
Geliştirilen ağ modelindeki her bir düğüm, her iki yönde de trafik akışına izin veren çift yönlü bağlantıya sahiptir ve paketleri işleme, onları uygun hedeflere yönlendirebilme yeteneğine sahip bir anahtarlama elemanı olarak modellenmiştir. Modellenen düğümler iki veya daha fazla ağ bağlantısıyla birbirine bağlanan DEVS atomik modelleridir. Düğümler / Yönlendiriciler, trafiğin işlenmesi için belli bir bant genişliğine (bit/sn), işlem yapma hızına ve trafiği yürütebilecek kapasitede bir tampon belleğe sahiptir. Düğümlerin bu özellikleri ile oynanarak farklı kapasite değerlerine sahip ağ birimleri oluşturulabilir ve ağ senaryoları geliştirilebilir [39].

D-DEVSNET ağ benzetim aracında üç farklı modelde düğüm / yönlendirici yer almaktadır. Bunları sunucu atomik modeli, istemci atomik modeli ve yönlendirici atomik modeli olmak üzere üç başlık altında inceleyeceğiz.

### 6.3.1.1. Sunucu düğüm atomik modeli

Sunucu ve istemci düğümler üzerinden bağlı buldukları otonom sistemlerin trafik geçişlerine imkân sağlayacak, bir anlamda transit geçiş hizmeti veren düğümler şeklinde tasarlanmışlardır. Sunucu düğüm atomik modeli, istemci tarafı ağıla iletişimi sağlamakla görevli bir yönlendirici olarak tasarlanmıştır. Geliştirilen sunucu düğüm ile istemci düğüm arasındaki yönlendirme işleminde BGP protokolü çalışmaktadır.

Şekil 6.3'ten de görüleceği üzere sunucu düğümün istemci düğümden tek farkı deneysel çerçeveye sahip olmasıdır. Dolayısı ile bütün trafik sonuçları sunucu tarafındaki tek bir dosyada tutulmaktadır. Sunucu düğüm içerisinde ağ arabirimi (Network Interface Card - NIC), DEVS varlıkları, veri ve kontrol paketleri yer almaktadır. Bölüm 3'te anlatıldığı üzere sunucu düğüm, aynı zamanda istemci ile bağlantıyı sağlayan soket yapısına sahiptir. Sunucu düğüm atomik modelinin D-DEVSNET ekranındaki görüntüsü Şekil 6.4'te verilmiştir.



Şekil 6.4. Geliştirilen sunucu atomik düğümün D-DEVSNET ekran çıktısı

DEVS tabanlı bir sunucu düğüm atomik modelinin ( $M_{\text{server\_node}}$ ) davranışını algoritmik olarak aşağıdaki gibi ifade edebiliriz;

$$M_{\text{server\_node}} = \langle X, Y, S, \delta_{\text{ext}}, \delta_{\text{int}}, \delta_{\text{con}}, \lambda, t_a \rangle$$



// Giriş portları ve değerleri

$X = \text{inport} \times \text{invalues}$   
 $\text{invalues} : \{\text{packet}, \text{packet\_HELLO}, \text{packet\_LSA}\};$   
 $\text{inports} : \{\text{in}, \text{inF}\}; // \text{düğüm ağ ara yüzü giriş portları}$

// Çıkış portları ve değerleri

$Y = \text{outport} \times \text{outvalues}$   
 $\text{outvalues} : \{\text{packet}, \text{packet\_HELLO}, \text{packet\_LSA}\};$   
 $\text{outports} : \{\text{out}, \text{outF}\}; // \text{düğüm ağ ara yüzü çıkış portları}$

// Durum değişkeni değerleri

$S = \text{phase} \times \sigma$   
 $\text{phase} : \{\text{'startup'}, \text{'idle'}, \text{'p\_prep\_to\_server'}\}$   
 $\sigma = \mathcal{R}^+_{0,\infty}$

// Harici durum geçiş fonksiyonu

$$\delta_{\text{ext}}((\text{phase}, \sigma), e, X) = \begin{cases} \text{if } \text{packet} \neq \text{LSA and output } x \leftarrow (\text{in}), \\ \text{packetDeAssembly} \\ \text{else if output } x \leftarrow (\text{inF}), \\ \text{packetDeAssembly} \\ // \text{paketi istemciye gönder} \end{cases}$$

// Dâhili durum geçiş fonksiyonu

$$\delta_{\text{int}}(\text{phase}, \sigma, Q) = \begin{cases} \text{if } q_{\text{FromNet}} \neq \text{isEmpty}, \\ \text{packet.remove and} \\ s \leftarrow (\text{'p\_prep\_to\_server'}, \sigma', x) \\ \text{else} \\ s \leftarrow (\text{'idle'}, \sigma', x) \\ \text{where } s \in S \end{cases}$$

// Çakışma geçiş fonksiyonu

$\delta_{\text{con}}((\text{phase}, \sigma), e, X) = \delta_{\text{int}}(\delta_{\text{ext}}(\text{phase}, \sigma), 0, X)$   
 //çakışma durumunda önce  $\delta_{\text{int}}$  çalışır

// Çıkış fonksiyonu

$\lambda(\text{phase}, \sigma, Q) =$   
 $\begin{cases} \text{if } \text{inPort} = \text{isReady}(), r_{\text{Packet}} \leftarrow (\text{inPort.readLine}()) \\ \text{packetAssembly} \\ \text{if } r_{\text{Packet}} \neq \text{null, and } \text{getPort}_{\text{name}} = \text{'outEvent'} \parallel \text{'inF'}, \text{output } y \leftarrow (\text{outF}, \text{null}) \\ \text{else} \\ y \leftarrow (\text{out}, \text{null}) \\ \text{where } y \in Y \end{cases}$

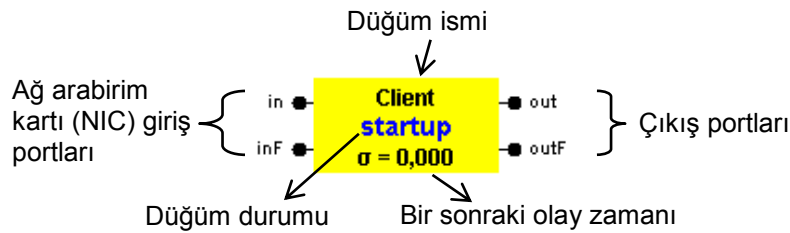
//Zaman ilerleme fonksiyonu

$\text{ta}(s) = \sigma$

### 6.3.1.2. İstemci düğüm atomik modeli

İstemci düğümü, sunucu tarafı ağla iletişimi sağlamakla görevli bir yönlendirici olarak tasarlanmıştır. İstemci düğüm içerisinde ağ arabirimi (Network Interface Card - NIC), DEVS varlıkları, veri ve kontrol paketleri yer almaktadır. Bölüm 3'de

anlatıldığı üzere istemci düğüm ile sunucu arasındaki veri iletiminde Java soket yapısı kullanılmıştır. Sunucu ve istemci düğümleri otonom sistemleri birbirlerine bağlayan sınır düğüm işlevine sahiptir. İstemci düğüm atomik modelinin D-DEVSNET ekranındaki görüntüsü Şekil 6.5’te verilmiştir.



Şekil 6.5. Geliştirilen istemci atomik düğümün D-DEVSNET ekran çıktısı

DEVS tabanlı bir istemci düğüm atomik modelinin ( $M_{client\_node}$ ) davranışını algoritmik olarak aşağıdaki gibi ifade edebiliriz;

$$M_{client\_node} = \langle X, Y, S, \delta_{ext}, \delta_{int}, \delta_{con}, \lambda, ta \rangle$$

// Giriş portları ve değerleri

$$X = inport \times invalues$$

*invalues* : {packet, packet\_HELLO, packet\_LSA };

*inports* : { in, inF }; // yönlendirici ağ arayüzü giriş portları

// Çıkış portları ve değerleri

$$Y = outport \times outvalues$$

*outvalues* : {packet, packet\_HELLO, packet\_LSA };

*outports* : {out, outF } ; // yönlendirici ağ arayüzü çıkış portları

// Durum değişkeni değerleri

$$S = phase \times \sigma$$

*phase* : { 'idle', 'startup', 'p\_prep\_to\_server', 'p\_prep\_to\_network' }

$$\sigma = \mathcal{R}_{0, \infty}^+$$

// Harici durum geçiş fonksiyonu

$$\delta_{ext}((phase, \sigma), e, X) = \begin{cases} \text{if } packet \neq LSA \text{ and output } x \leftarrow (in), \\ \quad packetDeAssembly \\ \text{else if output } x \leftarrow (inF), \\ \quad packetDeAssembly \\ //paketi sunucuya gönder \end{cases}$$

// Dâhili durum geçiş fonksiyonu

$$\delta_{int}(phase, \sigma, Q) = \begin{cases} \text{if } rpacket \neq null, \\ \quad s \leftarrow ('p\_prep\_to\_net', \sigma', x) \\ \text{else if } qFromNet \neq isEmpty, \\ \quad packet.remove \text{ and} \\ \quad s \leftarrow ('p\_prep\_to\_server', \sigma', x) \\ \text{else} \\ \quad s \leftarrow ('idle', \sigma', x) \\ \text{where } s \in S \end{cases}$$

// Çakışma geçiş fonksiyonu

$\delta con ((phase, \sigma), e, X) = \delta int (\delta ext (phase, \sigma), 0, X)$  //çakışma durumunda önce  $\delta int$  çalışır

// Çıkış fonksiyonu

$\lambda (phase, \sigma, Q) =$

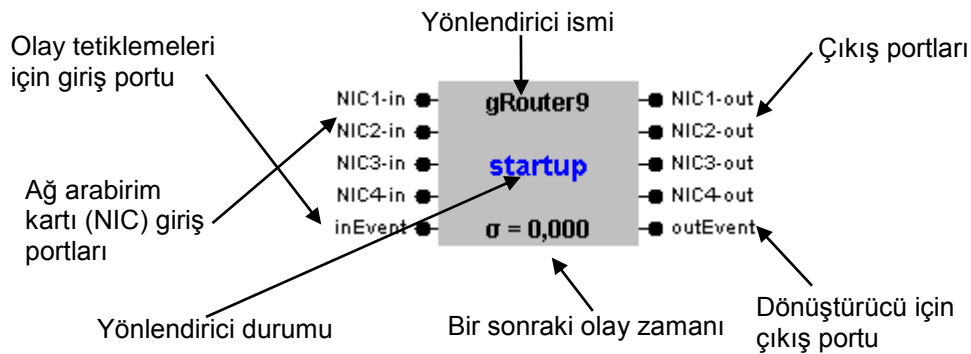
$$\left\{ \begin{array}{l} \text{if } inPort = isReady(), rPacket \leftarrow (inPort.readLine( )) \\ \text{packetAssembly} \\ \text{if } rPacket \neq null, \text{ and } getPort_{name} = 'outEvent' || 'inF', output y \leftarrow (outF, null) \\ \text{else} \\ y \leftarrow (out, null) \\ \text{where } y \in Y \end{array} \right.$$

//Zaman ilerleme fonksiyonu

$ta(s) = \sigma$

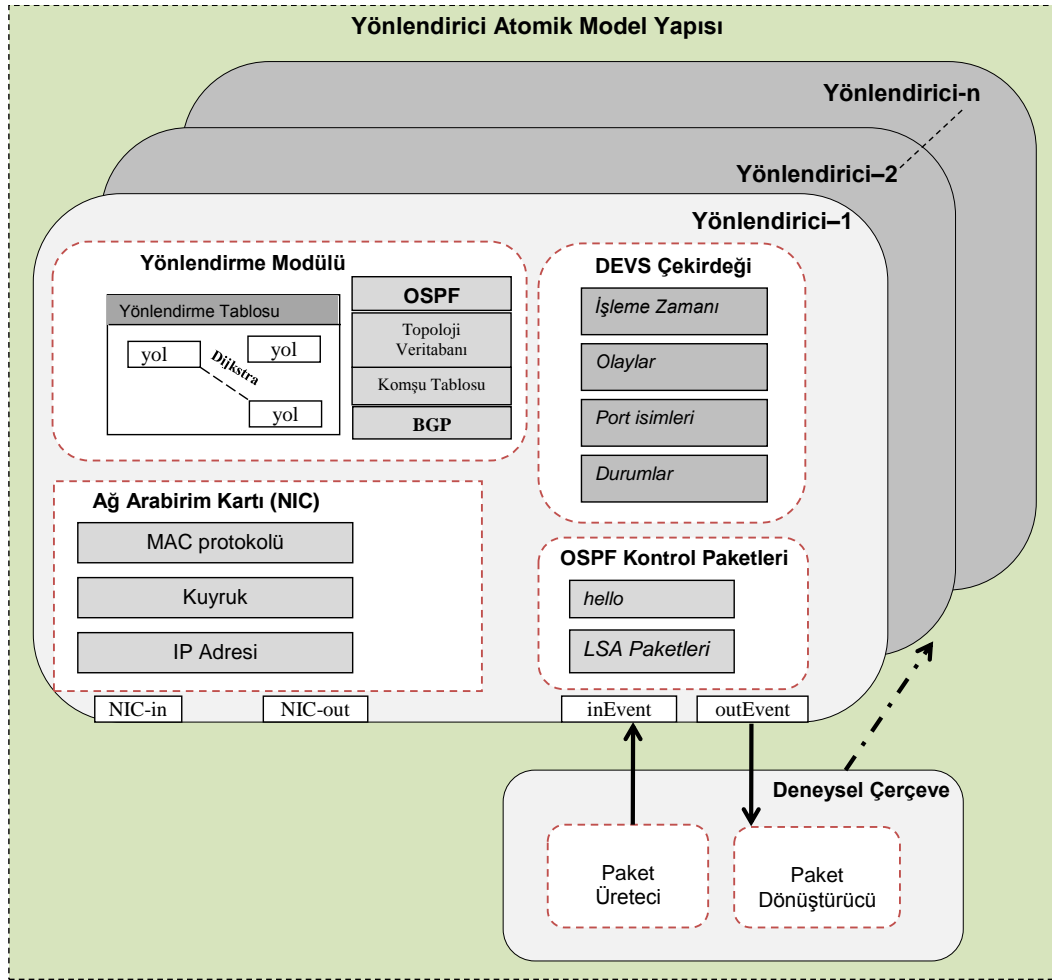
### 6.3.1.3. Yönlendirici Atomik Modeli

Şekil 6.7, ağ içerisinde yer alan bir yönlendiricilerin kavramsal modelini ve yapısını göstermektedir. Buna göre her bir atomik yönlendirici içerisinde yönlendirme modülü, ağ arabirimi (Network Interface Card - NIC), DEVS varlıkları, BGP, OSPF haberleşme ve kontrol paketleri yer almaktadır. Yönlendirme modülü, yönlendirmenin yapıldığı birimdir. Her düğüm, paketlerini hedef düğüme iletirken bu yönlendirme modülü içerisindeki verileri kullanır. Yönlendirme modülü paket yönlendirme işlemini, yönlendirme tablolarını ve algoritmalarını kullanarak gerçekleştirir. Yönlendirme modülü içerisinde yönlendirme tablosu, komşuluk tablosu ve topoloji veri tabanı yer almaktadır. Bu yönlendirme tabloları ağın mevcut durumunu yansıtırlar. Yönlendirici atomik modelinin D-DEVSNET ekranındaki görüntüsü ise Şekil 6.6'da verilmiştir.



Şekil 6.6. Yönlendirici atomik modelinin D-DEVSNET ekran çıktısı

Yönlendiriciye, işleyebileceğinden daha fazla paket geldiğinde paketleri kuyruğa yerleştirir ve kuyruk dolduğunda gelen paketleri imha eder. Tipik bir ağ arabirimi (NIC), gelen ve giden mesajların basit bir şekilde kuyruklama işlemine alınmasını yürüten çok basit bir MAC protokolüne sahiptir. Ana amacımız geniş ölçekli ağ koşulları altında geliştirilen modeli ve aracı test etmek olduğu için MAC protokolü çok basit tutulmuştur.



Şekil 6.7. Yönlendirici atomik modelin kavramsal yapısı

Otonom sistemler içerisindeki yönlendiriciler üzerinde OSPF protokolü, otonom sistemler arası ise BGP protokolü çalışmaktadır. Dolayısıyla, yönlendirme modülü içerisinde OSPF (Sürüm 2) [84] ve BGP protokolleri uygulanmıştır. Bölüm 4’de de ifade edildiği gibi OSPF, otonom sistem içerisinde çalışan bir protokol iken BGP, otonom sistemler arası çalışan bir İnternet protokolüdür. Yönlendirme modülünde, protokol yönetim ve organizasyonlarını içeren farklı protokol yığınları saklanır.

Örneğin, ‘komşu tablosunda’ komşu tablosu hakkındaki bilgiler tutulur. Düğümler, her bir komşusu için sanal bir ağ arabirim kartına(NIC) sahiptir. Kontrol mesajlarını ise ‘hello’ ve ‘LSA’ paketlerini içeren mesajlar oluşturmaktadır [32].

DEVS tabanlı OSPF protokolüne sahip bir yönlendirici atomik modelin ( $M_{OSPF\_atomic\_router}$ ) davranışını algoritmik olarak aşağıdaki gibi ifade edebiliriz;

$M_{OSPF\_atomic\_router} = \langle X, Y, S, \delta_{ext}, \delta_{int}, \delta_{con}, \lambda, ta \rangle$

**// Giriş portları ve değerleri**

$X = inport \times invalues$   
 $invalues : \{packet, packet\_DATA, packet\_HELLO, packet\_LSA\};$   
 $inports : \{NIC1\_in, inEvent\}; // yönlendirici ağ arayüzü giriş portları$

**// Çıkış portları ve değerleri**

$Y = outport \times outvalues$   
 $outvalues : \{packet, packet\_DATA, packet\_HELLO, packet\_LSA\};$   
 $outports : \{NIC1\_out, outEvent\}; // yönlendirici ağ ara yüzü çıkış portları$

**// Durum değişkeni değerleri**

$S = phase \times \sigma \times Q$   
 $phase : \{‘idle’, ‘startup’, ‘queuing’, ‘flooding’, ‘congested’, ‘newLSAadded’, ‘gettingRoute’, ‘subNetting’\}$   
 $\sigma = \mathcal{R}^+_{0,\infty}$   
 $Q = Queue \times Qneighbour\_table \times Qtopology\_database$   
 $// Queue gelen ve giden paketlerin tutulduğu bir pakettir,$   
 $// Qneighbour\_table komşulukları gösteren (ad ve bağlantı durumu) bir veri tabanıdır,$   
 $// Qtopology\_database LSA paketlerini içeren bir topoloji veri tabanıdır$

**// Harici durum geçiş fonksiyonu**

$\delta_{ext}((phase, \sigma, Q), e, X)$   
 $= \left\{ \begin{array}{l} \text{if } packet = LSA, \text{ its source} = \text{address and it is included in LSA history,} \\ \quad \text{discard the packet} \\ \text{else if } packet = DATA \text{ and queueu has enough space for the packet,} \\ \quad \text{enqueue the packet and } s \leftarrow ('queuing', \sigma', x) \\ \quad \text{else } s \leftarrow ('congested', \sigma', x) \\ \text{where } s \in S \end{array} \right.$

**// Dâhili durum geçiş fonksiyonu**

$\delta_{int}(phase, \sigma, Q)$   
 $= \left\{ \begin{array}{l} \text{if } queue = \emptyset, s \leftarrow ('idle', \sigma', x) \\ \quad \text{else dequeue packet} \\ \quad \text{if } packet \neq null \text{ and } packet = HELLO, \\ \quad \quad \text{add neighbour and } s \leftarrow ('addingNbor', \sigma', x) \\ \text{else if neighbour table size} = \text{number of NIC's, } s \leftarrow ('flooding', \sigma', x) \\ \quad \text{else if } packet = LSA, \\ \quad \quad \text{add to topology database and } s \leftarrow ('newLSAadded', \sigma', x) \\ \quad \quad \text{else if } packet = DATA \text{ and its destination} \neq \text{address,} \\ \quad \quad \quad s \leftarrow ('gettingRoute', \sigma', x) \\ \text{else if } packet \text{ is } DATA \text{ and its destination} = \text{address } s \leftarrow ('subnetting', \sigma', x) \\ \text{where } s \in S \end{array} \right.$

**// Çakışma geçiş fonksiyonu**

$$\delta_{con}((phase, \sigma, Q), e, X) = \delta_{ext}(\delta_{int}(phase, \sigma, Q), 0, X)$$

//çakışma durumunda önce  $\delta_{ext}$  çalışır

**// Çıkış fonksiyonu**

$$\lambda(phase, \sigma, Q) =$$

$$\left\{ \begin{array}{l} phase = 'flooding', \text{ create LSA packet and output } y \leftarrow (NIC1\_out, LSA) \\ phase = 'startup', \text{ create HELLO packet and output } y \leftarrow (NIC1\_out, hello) \\ \quad \text{when } phase = 'newLSAadded', y \leftarrow (NIC1\_out, LSA) \text{ and} \\ \quad \quad s \leftarrow ('forwardingLSA', \sigma', x) \\ \quad \text{when } phase = 'gettingRoute', \text{ get outgoing interface, NIC and} \\ \quad \quad y \leftarrow (NIC\_out, packet) \\ \quad \text{when } phase = 'subNetting', y \leftarrow (outEvent, packet) \\ \quad \text{where } y \in Y \end{array} \right.$$
**//Zaman ilerleme fonksiyonu**

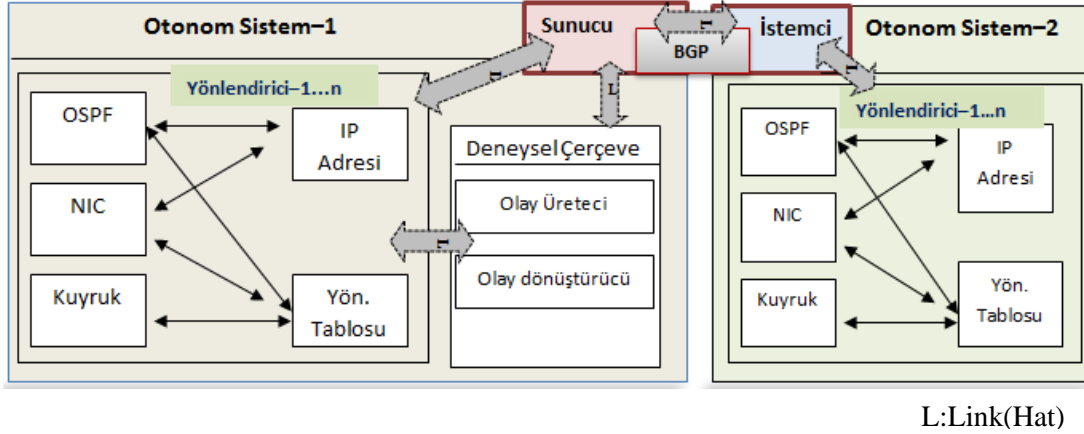
$$ta(s) = \sigma$$
**6.3.2. Birleşik Modeller**

Birleşik modeller, atomik bileşenlerin birleşiminden oluşmaktadır. İstemci tarafındaki düğüm ve yönlendiriciler ile sunucu tarafındaki modeller otonom sistemler olarak birleştirilir ve tüm ağ çıktıları deneysel çerçeve ile yönetilir ve izlenir. Otonom sistemlerin birleştirilmesinde BGP protokolü kullanılır.

**6.3.2.1. Otonom Sistem modeli**

D-DEVSNET benzetim ortamı istemci ve sunucu taraflı olmak üzere iki otonom sistemden oluşmaktadır. Ancak istenildiğinde daha fazla istemci oluşturularak ikiden fazla otonom sistem rahatlıkla oluşturulabilir. Otonom sistemler, kendi içerisinde özerk olan sistemlerdir. Günümüz İnternetinin trafik altyapısında otonom sistem içerisinde OSPF protokolü kullanılırken otonom sistemler arasında ise BGP protokolü kullanılır. Benzer şekilde D-DEVSNET benzetim ortamında da otonom sistem içerisinde DEVS tabanlı OSPF protokolü koşturulurken, otonom sistemler arasında BGP protokolü koşturulur / çalıştırılır. BGP, otonom sistemleri birbirine bağlamak amacı ile otonom sistemler arası bağlantıda veri trafiğinin hangi yönlendiriciler üzerinden gerçekleştirileceğine karar verilmesini sağlayan bir yönlendirme protokolüdür. Şekil 6.8'te D-DEVSNET otonom sistem birleşik modeli

görülmektedir. Şekilde görüldüğü üzere Sunucu ve İstemci düğümleri otonom sistemleri birbirlerine bağlayan sınır düğümlerdir. Bu düğümler arasındaki veri/paket trafiğini BGP protokolü sağlar.



Şekil 6.8. D-DEVSNET otonom sistem birleşik modeli

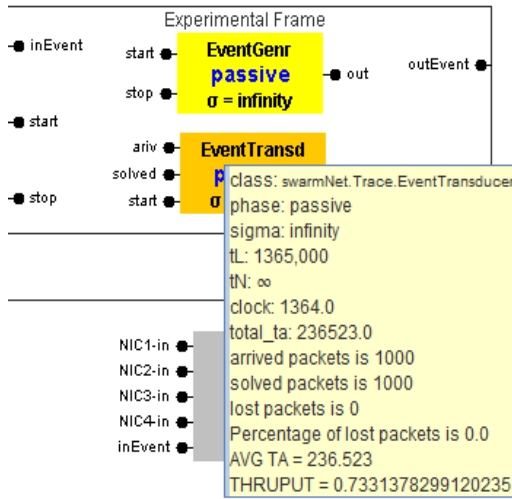
### 6.3.3. D-DEVSNET deneysel çerçevesi

Gerçek dünyada çalışıldığında, deneysel çerçeve (Experimental Frame - EF) sistemin veya modelin çalıştırılacağı deneysel şartları / ortamları tarif eder. Gerçek dünyadan farklı olarak yazılım ortamında ise kurulan sistemin / modelin test edilebilmesi için Bölüm 2’de anlatılan DEVS içerisindeki deneysel çerçeve kavramının oluşturulması / tanımlanması gerekmektedir [28].

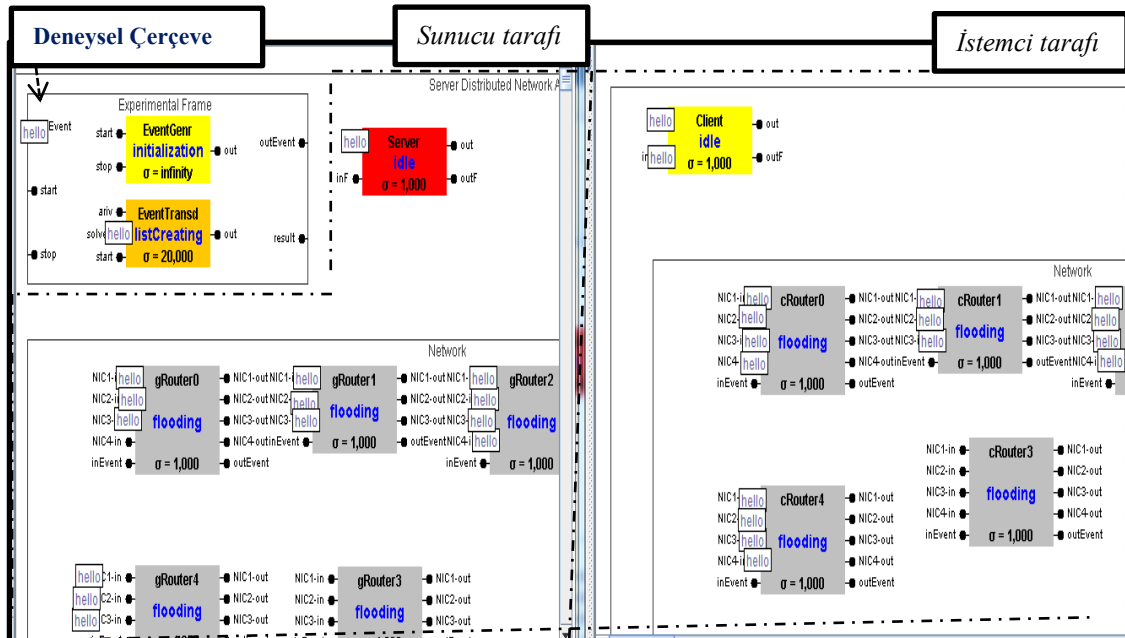
Bizim çalışmamızda deneysel çerçeve iki bileşenden oluşmaktadır;

- Olay Üreteci (Event Generator): Sistemin / modelin giriş uçlarına, sisteme / modele bir tetikleme sinyali vermek için bağlanan bir üretectir ki bu üretet benzetim aracımızda ‘EventGenr’ ismi ile temsil edilmektedir.
- Olay Dönüştürücü (Event Transducer): Sistemin / modelin çıkış uçlarına, sistemden / modelden gelen sonuçların değerlendirilebilmesi için bağlanan bir dönüştürücüdür ki bu dönüştürücü benzetim aracımızda ‘EventTransd’ ismi ile temsil edilmektedir. Olay dönüştürücü, yapılan benzetim çalışmasının

sonuçlarının değerlendirilmesi ve analizinde kullanılan bir araçtır. Ağın çalışması boyunca elde edilen / üretilen çıktıların (paketlerin) analiz edilmesi ve ağın değerlendirilmesinde yardımcı olacak anlamlı sonuçların elde edilmesinde önemli işleve sahiptir.



Şekil 6.9. D-DEVSNET grafiksel arayüzünde olay dönüştürücüde anlık izlenen veriler



Şekil 6.10. D-DEVSNET ağı deneysel çerçeve bağlantısı.



D-DEVSNET’ de olay dönüştürücü tarafından gözlemlenen / izlenen ağ ortamına ait veriler hem anlık izlenebilir hem de bir CSV dosyasında (Comma Seperated Values - CSV) saklanabilir. Ağın başarımları analizi ve değerlendirilmesinde üretilen çeşitli grafikler bu kaydedilen verilerden istifade ile çizilir / elde edilir (Şekil 6.9). Geliştirilen benzetim aracının deneysel çerçeve gösterimi ve ağ bileşenleri ile bağlantısı Şekil 6.10’da gösterilmiştir. Şekilde de görüleceği üzere deneysel çerçeve sunucu tarafında çalışmaktadır. Sunucu tarafındaki otonom sistemin ağ bileşenleri ile doğrudan bağlantısı varken istemci tarafındaki otonom sistemle bağlantısı sunucu – istemci düğümleri üzerinden dolaylı olarak gerçekleşmektedir.

#### 6.4. OSPF Protokol Modeli

Bu tezde geliştirilen dağıtık benzetim ortamını test etmek üzere Zengin [32] tarafından geliştirilen OSPF modeli kullanılmıştır. Aşağıdaki bölümler bu modeli ayrıntılı olarak anlatmaktadır.

OSPF protokolü bağlantı durumu algoritmasının en yaygın kullanılan örneklerinden biridir [71]. OSPF bağlantı durum ilanlarını makul derecede hızlı bir yakınsama ile birlikte diğer komşuların tümüne aktararak topolojik değişiklikleri kolayca tespit edebilen bir bağlantı durum yönlendirme protokolüdür. Dağıtık DEVS tabanlı OSPF protokolünün gerçekleştirilme aşamaları aşağıda verilmiştir. Buna göre;

1. Adım- Ağ keşif aşaması: Her düğüm/yönlendirici başlangıçta kurulum ‘startup’ safhasındadır. Bu aşamada yönlendiricilerin birbirlerini tanıması, yani komşuluk tesisi için her yönlendirici birbirlerine ‘hello’ paketi gönderir. Ağ etki alanındaki(domain) yönlendiricileri bulmanın arkasındaki temel fikir, o an bilinen yönlendiriciden yola çıkarak komşu yönlendiricileri bulmak ve tüm yönlendiriciler bulunana kadar bu işleme devam etmektir. Böylece aynı otonom sistem içerisindeki tüm yönlendiriciler arasında komşuluk ilişkisi kurulmuş ve diğer bir ifade ile yakınsama gerçekleşmiş olur.

2. Adım- Akış (Flooding) aşaması: Komşuluk ilişkisi kurulduktan sonra yönlendirme kapsama alanı içerisindeki LSA'ların dağıtıldığı 'flooding' (sel baskını / akış) denilen aşamaya geçilir. Komşu düğümlerin listesi komşu tablosunda (neighbor table) saklanır. Hello mesajlarının alınmasıyla her düğüm/yönlendirici komşu tablosunu günceller ve bağlantı durumu ilan paketlerini (LSA) bütün komşularına gönderir. Eğer ağda ( yol veya bağlantı durumuna bağlı olarak ) topolojik bir değişim meydana gelirse bu komşudan diğer yönlendiricilere değişiklikleri bildirmek için anlık güncelleme bilgisi yollanır. Sadece değişiklikten etkilenen yolun önceki değiştirilir ve sadece güncellenmiş değişikliğe sahip LSA bilgisi yollanır. Geliştirilen çalışmada bağlantı durumlarını taşımak için DEVS varlıkları kullanıldı. Sel baskınının benzetimine önemli bir ek yük getirmesi sebebiyle, sabit zaman güncellemeleri yerine olay tetikleme yaklaşımı seçildi.

Güvenilir bir sel baskını algoritması veri tabanlarının makul bir yakınsama periyodu sonrasında senkronize olacağını garanti eder ve LSA'ların dağıtımını hızlı bir şekilde yapar ve yakınsama zamanını küçük tutar [39].

3. Adım- LSA paketlerinin oluşturulması: Her düğüm belirli bir zaman diliminde (30 sn) ya da olay tetiklemeli olarak birbirine 'bağlantı durumu ilanı- LSA' paketlerini gönderirler. Sabit bir zaman dilimi yerine olay tetikleme yaklaşımı kullanıldı. 'flooding' aşaması tamamlandıktan sonra olay tetiklemesi ile düğümler 'new LSA added' aşamasına geçer. Eğer gelen LSA paketi, topoloji veritabanında varolan paketlerden daha küçük sürüme sahip ise daha ileri işleme geçilmeden kuyruktan düşürülmelidir. OSPF protokolünün otonom sistemlerine ait topoloji veri tabanları, hem komşuluk tablosu hem de topoloji veri tabanından oluşmaktadır. Topoloji veri tabanında LSA paketleri, komşuluk tablosunda ise komşu yönlendiricilerin ve bağlı oldukları ara yüzlerin isimleri saklanmaktadır.
4. Adım- Kuyruğa alma (queuing) aşaması: OSPF benzeticisinde kuyruk, FIFO (İlk Giren İlk Çıkar- First In First Out) mantığına göre çalışan bir bağlı liste (Linked List) olarak modellenmiştir. Kuyruğa alma işlemi 'queuing' durumu ile ifade

edilmektedir. Kuyruk yapısında iki temel işlem söz konusudur. Birincisi kuyruğa paket ekleme (enqueue) , diğeri ise kuyruktan paket alma (dequeue) işlemleridir. Bu iki işlemin yanında, kullanılan kuyruğun dolu ya da boş olup olmadığının kontrol edilmesi gereklidir. Ayrıca kuyruğa paket ekleme ve çıkarmanın nereden yapılacağını gösteren ön (front) ve son (rear) tamsayı değişkenlerinin tanımlanması gerekir. Paketlerin kuyruğa eklenmesinde bir öncelik değeri (0 en küçük, 7 en büyük değer olmak üzere) kullanılmıştır. Kuyruğa paket ekleme ve silme işlemleri bu öncelik değere bakılarak yapılır. Kuyruk yapısı şu mantığa göre çalışır;

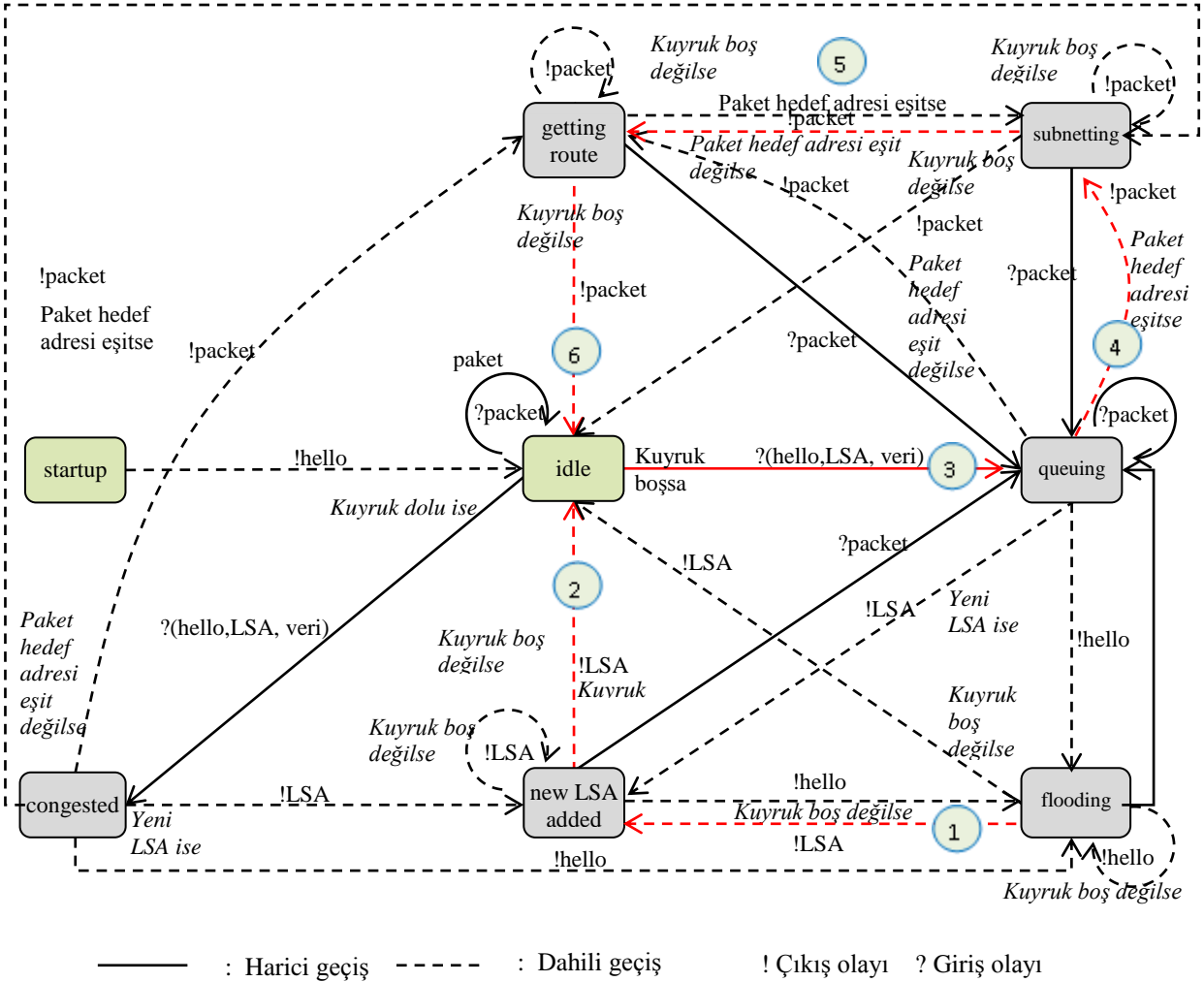
- En yüksek öncelik değerine sahip paket, kuyruğun en önüne eklenir.
- Kuyruğa eklenmek istenen paket, kuyruğun azami boyutundan büyükse paket reddedilir.
- Kuyruğa eklenmek istenen paket, kuyruğun azami boyutundan küçük fakat kuyruk boş değilse(dolu ise) paketler düşürülür.
- Kuyruk boş ise paketler kuyruk sonuna eklenir.

Yönlendirici kuyruğu boş olmadığı sürece, kuyruktaki paketler sırasıyla ‘subnetting’ durumunda değerlendirilir ve eğer paket hedef adresi eşit değilse en kısa yolun hesaplanması için bir sonraki duruma ‘getting route’ geçilir.

5. Adım- En kısa yolun bulunması: Kuyruğa alma işleminden sonra paketin hedef adresi bulması için en kısa yolun bulunması ‘getting route’ aşamasına geçilir. Her yönlendirici, paketi hedefine yönlendirirken en kısa yolu bulmak için Dijkstra algoritmasını çalıştırır. Meydana gelen topolojik değişiklikler Dijkstra algoritması kullanılarak yeniden hesaplanır ve yönlendirme tablosu içerisine aktarılır [3, 33].

Bu adımlara / aşamalara göre geliştirilen OSPF protokolü ile birlikte yönlendirici atomik modelinin algoritması Bölüm 6.3.1.3’te verilmişti. Bu algoritmaya göre çalışan yönlendirici atomik modelin sonlu durum diyagramı Şekil 6.11’de verilmiştir.

Şekil 6.11'deki kesikli çizgiler dâhili, düz çizgiler ise harici durum geçişlerini göstermektedir.



Şekil 6.11. OSPF protokolü durum geçişleri

Özetle tüm düğümler başlangıçta 'başlangıç-startup' durumdadırlar. Düğümler bu durumdan 'hello' paketi göndererek çıkarlar, 'hello' paketleri ile komşuluk ilişkisi kurulduktan sonra yönlendirme kapsama alanı içerisindeki LSA'ların dağıtıldığı 'flooding' denilen aşamaya geçilir. 'flooding' aşamasından sonra LSA paketlerinin oluşturulduğu 'new LSA added' aşamasına geçilir. Bu aşamadan sonra bütün düğümler 'idle' durumuna geçer. 'idle' durumunda iken bir olay üreticiden alınan bir tetikleme sinyali ile veri paketleri üretilir. Düğüme gelen her türlü paket ve düğüm tarafından oluşturulan paketler 'queuing' durumunda kuyruğa eklenir. Düğümün kuyruğu boş olmadığı sürece, kuyruktaki paketler sırasıyla 'subnetting' durumunda

değerlendirilir ve eğer paket hedef adresi eşit değilse en kısa yolun hesaplanması için ‘getting route’ durumuna geçilir. Üretilen paketler, ağda hedefine ulaşan ve kaybolan paket sayısına eşit olduğunda tekrar ‘idle’ durumuna geçilir. Ayrıca ‘idle’ durumundaki bir düğümden bir paket geldiğinde, kuyruk dolu ise durumunu ‘congested’ olarak değiştirir. Benzer şekilde bir düğüme gelen paketler, kuyruk boyutunun yetersizliği, önceden tanımlı yaşam süresinin dolması ve o linke ait bant genişliği kapasitesinin yetersizliği gibi sebeplerle kaybolur ve düğümün durumu ‘congested’ olarak ayarlanır.

### 6.5. Yönlendirme Modülü

Yönlendirme modülü içerisinde yer alan yönlendirme tablolarını kısaca ağdaki yol durum bilgisini tutan tablolar olarak tanımlayabiliriz. Ağdaki her bir bilgisayar/düğüm yönlendirme tablolarına bakarak hangi veri paketini, nereye göndereceğine karar verir.

Bizim çalışmamızda ağdaki her bir düğüm, bir yönlendirme tablosuyla donatılmış ve böylece veri ve kontrol paketlerinin sistematik olarak ağ boyunca yönlendirilmesi sağlanmıştır. Yönlendirme tablosunun boyutu düğümün keşfedebildiği komşu düğümü / komşu sayısı ile ilişkilidir. Bir düğüm bir paketi belirli bir hedefe doğru göndermeye ihtiyaç duyduğunda, hangi çıkış portunun kullanılacağı kararı yönlendirme tablosunda belirtilen bilgiye göre verilir.

Herhangi bir yönlendirici, komşularıyla olan bağlantıda bir değişiklik olduğunu anladığı zaman, bu değişikliği tüm ağa / komşularına ‘çoğa gönderim (multicast)’ olarak bildirir. Ancak bu bildirme tüm yönlendirme tablosunun gönderilmesi şeklinde olmayıp, yalnızca algılanan değişikliğin bildirilmesi şeklinde olur. Bu durum OSPF protokolünün RIP protokolüne göre en belirgin farkıdır.

Yönlendirme modülü içerisinde, yönlendirme tablosu, komşuluk tablosu ve topoloji tablosu yer almaktadır. Bu yönlendirme modülü, ağın mevcut durumunu OSPF protokolüne göre oluşturur. OSPF protokolünde, komşuluk tablosu, topoloji veri

tabanı ve yönlendirme tablosu olmak üzere 3 farklı veri yapısı bulunmaktadır. Ayrıca yönlendirme modülü içerisinde komşu otonom sistemlerin adresini tutan bir BGP tablosu da yer almaktadır.

Şekil 6.12’deki örnek yönlendirme tablomuzdan da görüleceği üzere;

- Komşuluk Tablosunda, yönlendiricilerin komşularının listesi tutulmaktadır. Hedef düğüm IP adresi, hedefe ulaşmak için hangi düğüme başvurulması gerektiği ve çıkış arabirim ismi ve maliyet bilgileri yer almaktadır.
- Topoloji Veri Tabanında, otonom sistem içerisindeki bütün yönlendiriciler ve onların bağlantıları tutulmaktadır. LSA’larda sürüm numaraları ile birlikte bu tabloda yer almaktadır.
- Yönlendirme Tablosunda ise düğüm IP adresi, otonom sistem (AS) numarası, kuyruk durumu, gönderilen, hedefe ulaşan ve kaybolan paket sayıları tutulur. Hedef düğümlere giden en kısa yola, Dijkstra algoritması kullanılarak karar verilir ve paketler bu rota üzerinden gönderilir.

```

IPAddress: 2.0.0.0
AS number: 2
QueueState: 1 packets(0KB)
Number of Routed Packets: 189
Number of Destined Packets: 53
Number of Discarded Packets: 389
0 of 2 LSA discarded
Neighbor Table :
=====
Number of Neighbors: 2
[dest:2.0.0.2 nextHop:2.0.0.2 Out Port:NIC1-in cost:1
, dest:100.0.0.0 nextHop:100.0.0.0 Out Port:NIC2-in cost:1
]
BGP Table :
=====
Number of networks: 1
{100=100.0.0.0}
Topology Database:
=====
Number of LSA's: 2
[LSA_v0ofgRouter2
, LSA_v0ofgRouter1
]

```

Şekil 6.12. Bir yönlendirme tablosunun D-DEVSNET ekranındaki görünümü

Yönlendirme tablosu D-DEVSNET içerisinde ‘RoutingTable’ sınıfı ile temsil edilmiştir. ‘RoutingTable’ sınıfı içerisinde komşuluklar ‘neighbour\_table’ isimli tabloda, topoloji veri tabanı ‘td\_base’ isimli vektörde tutulmaktadır. Yollar ise ‘Route’ sınıfı ile gösterilmiştir / açıklanmıştır. ‘RoutingTable’ sınıfı ve diğer sınıfların kaynak kodları ekteki CD’de verilmiştir.

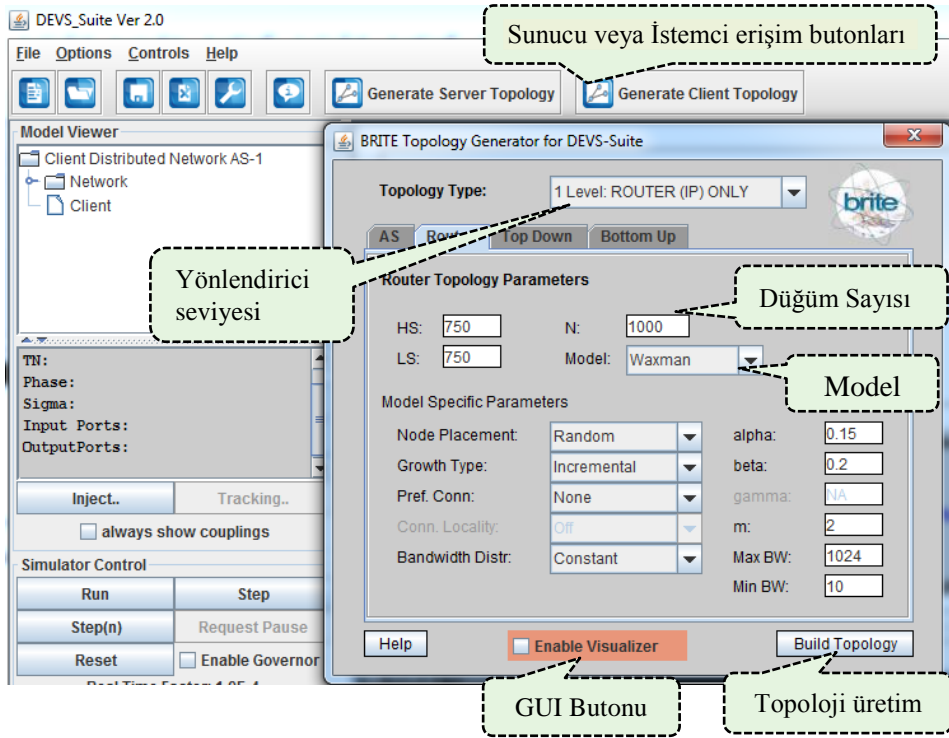
## 6.6. BRITE Topoloji Üretim Aracı

Topoloji üretim araçları ve modelleri ile ilgili geniş çaplı inceleme, M. A. Rahman ve arkadaşları tarafından 2009 yılındaki ‘Network Topology Generation and Discovery Tools’ isimli bir makalede yayınlandı [12]. Bölüm 5.3’te incelenen topoloji üretim araçları arasında BRITE (The Boston University Representative İnternet Topology Generator) [11] topoloji üretim aracını öne çıkaran başlıca özelliklerini;

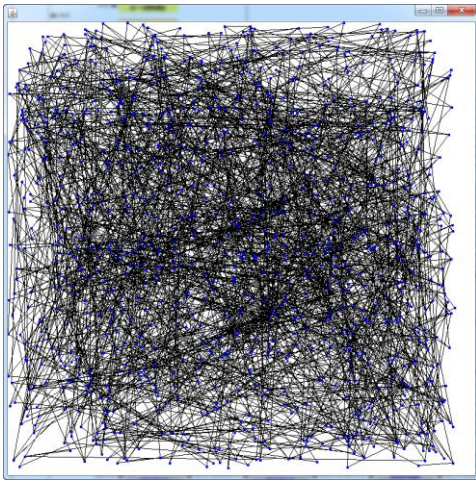
- Hem Java hem de C++ tabanlı olması,
- Waxman [99], Barabasi [100] gibi birçok topoloji modelini desteklemesi,
- ns-2, OMNeT++ ve SSFNet gibi ağ benzetim araçları ile çalışabilmesi,
- Açık kaynak kodlu olması,
- Kolay ve kullanışlı grafiksel ara yüze sahip olması,

şeklinde sıralayabiliriz.

Bizde bu çalışmamızda BRITE topoloji üretim aracını D-DEVSNET benzetim aracına entegre ettik. BRITE topoloji üretim aracının D-DEVSNET benzetim aracına entegre edilmiş hali ve ekran görüntüsü Şekil 6.13’te verilmiştir. 1000 düğümlü bir ağ topolojisinin ‘BRITE görüntüleyici’ sindeki ekran görüntüsü ise Şekil 6.14’te gösterilmiştir.



Şekil 6.13. D-DEVSNET BRITE topoloji üretici ekran görüntüsü



Şekil 6.14. 1000 düğüm için BRITE görüntüleyici



## **BÖLÜM 7. D-DEVSNET BENZETİM ARACININ DOĞRULANMASI ve GEÇERLENMESİ**

### **7.1. Giriş**

Bu bölüm, modellenen ve DEVS-Suite ortamında gerçekleştirilen paralel ve dağıtık, geniş ölçekli D-DEVSNET ağ benzetim aracının doğrulama ve geçleme testlerini içermektedir. Bu test ve deneyler; geliştirilen deneysel çerçeve yardımıyla D-DEVSNET ağ benzetim aracının kavramsal model ve davranış geçlemesini, ns-2 ağ benzetim aracı ile benzer senaryolarda test edilmesini ve iş çıkarma yeteneklerinin karşılaştırılmasını kapsar.

Bölüm 8’de yapılan çalışma ile farklı ölçeklerde, farklı topolojilerde ve farklı ağ trafikleri altında üretilen senaryolar sonucunda elde edilen veriler ve gözlemlenen çıkışlar grafikler halinde sunulmuş ve D-DEVSNET ağ benzetim aracının geniş ölçekli klasik bir ağı, modelleme ve benzetim kabiliyetine sahip olduğu gösterilmeye çalışılmıştır.

### **7.2. D-DEVSNET Doğrulama ve Geçleme Süreci**

Doğrulama ve geçleme (Verification and Validation – V&V) deneyleri yazılım mühendisliğinin temel konularından biri olduğu gibi modelleme ve benzetim sistemlerinin de güvenilirliğini ölçmede vazgeçilmez bir araçtır. Doğrulama, modellenen sistemin gereksinimlere uygun ve hatasız gerçekleştirilmesi üzerine odaklanırken geçleme, geliştirilen sistemin hedeflere uygunluğunu ve ihtiyacı karşılayabilmesi üzerine odaklanır. Boehm [101]’in ifadesi ile geçleme ‘doğru ürünü mü geliştiriyoruz?’, doğrulama ise ‘ürünü doğru mu geliştiriyoruz?’ sorularına cevap arama sürecidir. Geçleme ile tasarlanan modelin, modellediği sistemle olan

benzerlik derecesi, doğrulamayla ise oluşturulan modelin, modeli oluşturan tasarımcının öngördüğü kavramsal tanımlamalarla hangi düzeyde benzeştiği ölçülmektedir.

Bilgisayar ağı araştırmalarında modelleme ve benzetim (M&S) yönteminin, yeni ağ teknolojilerinin etkin bir şekilde geliştirilmesi, test edilmesi, değişik ağ koşulları ve senaryoları altında mevcut protokollerin iyileştirilmesi ve analizinde önemli bir rolü vardır. Modelleme ve benzetim araçları ile yapılan çıkarımların, elde edilen sonuçların kabul edilebilirliğinin güvence altına alınması gerekmektedir. Bu güvencenin sağlanabilmesi için doğrulama ve geçерleme adı verilen çalışmalar / testler yapılır [39, 44].

Doğrulama, yazılan program kodunun kavramsal modelde tanımlanan davranışı tam olarak yansıttığı kabulü ile soyut tanımlamadan program koduna geçişin doğruluğu ile ilgilenir. Bu aşamada giriş verilerinin ve model yapısının mantıksal hata içerip içermediği test edilerek, model doğrulaması yapılır. Geçerleme, modelin kullanım amacı dikkate alınarak ürettiği verilerin modellenen sistemle ve ölçülmüş verilerle olan benzerlik derecesinin belirlenmesi işlemidir. Bir başka ifade ile deneysel çerçeve ortamında elde edilen sonuçların karşılaştırılması işlemidir. Doğal olarak hiçbir modelin % 100 doğru olması da mümkün değildir [23].

Bölüm 2’de açıklanan benzetim çalışmasının işlem basamaklarında doğrulama ve geçerleme aşaması önemli bir işleve sahiptir. Şekil 7.1’de deneysel çerçeveye sahip ayrık olaylı bir OSPF modelleme ve benzetim çalışmasının doğrulama ve geçerleme süreci gösterilmiştir. Tüm süreç, D-DEVSNET modeli ve deneysel çerçevesi kullanılarak örneklenmiştir. Şekil 7.1’den de görüleceği üzere, modelleme aktivitesi problem tanımı ve hedeflerin belirlenmesi ile başlar. Geniş ölçekli dağıtık DEVS tabanlı ağ benzetim aracının gelişim sürecinde belirleyici olan ana hedefleri şöyle sıralayabiliriz;

- Sistem teorisine dayalı bir tasarım,
- Modüler ve hiyerarşik bir yapı,

- Kolay kurulum,
- Yüksek performans,
- Yüksek izlenebilirlik,
- İyi görsel takip,
- Geniş test çerçevesi,
- Platform bağımsızlığı,
- Dağıtık çalışabilmesi,
- Ölçeklenebilirlik,
- Uzaktan erişilebilirlik.

Yapılan testlerde geliştirilen benzetim aracının yukarıdaki özelliklere sahip olduğu görülmüştür.

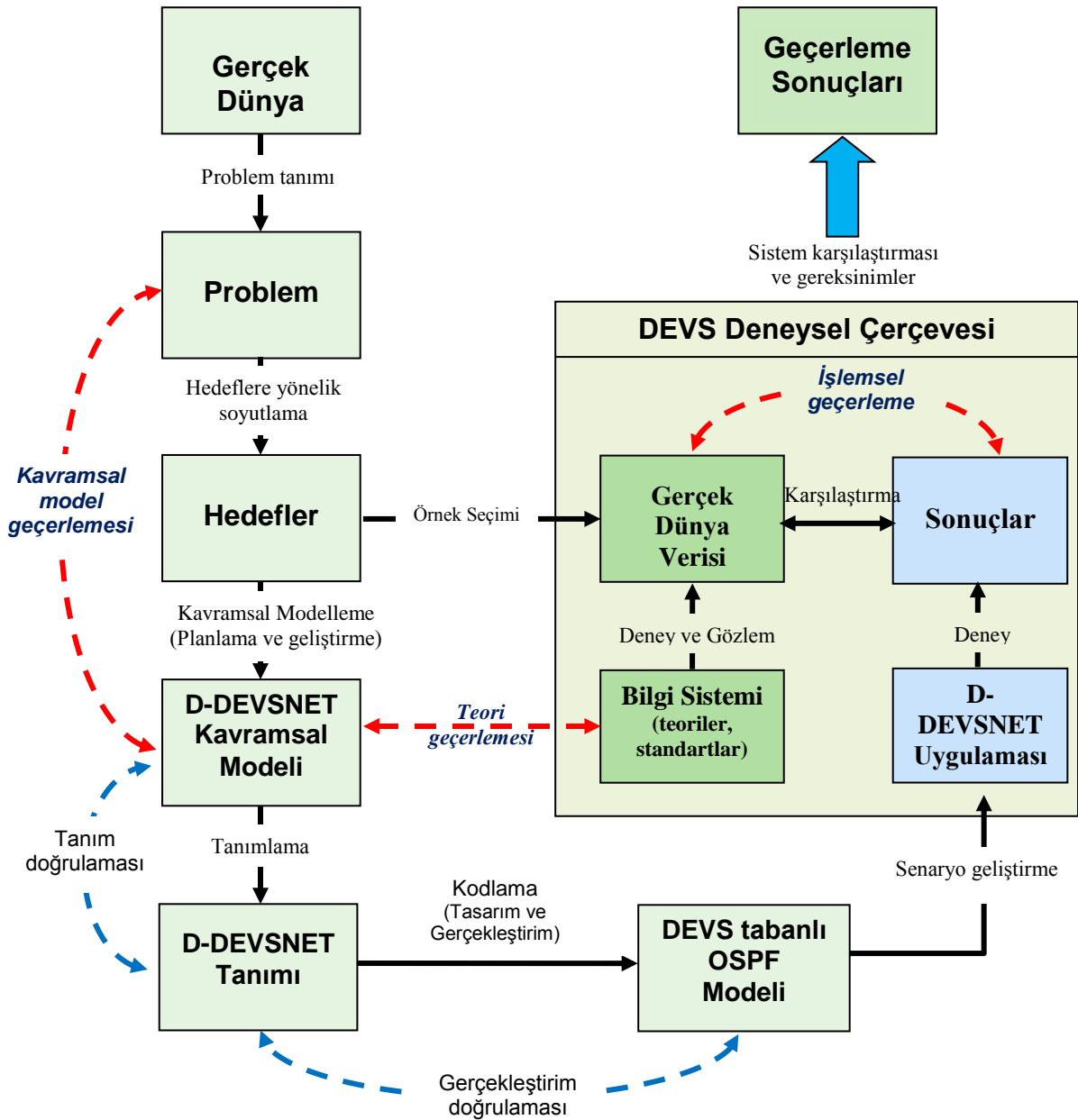
D-DEVSNET protokol geçерleme süreci; kavramsal model geçерlemesi, teori geçерlemesi ve işlemsel geçерleme olmak üzere üç aşamalı / katmanlı gerçekleştirilmiştir.

Kavramsal model, geliştirilen benzetim aracının geniş ölçekli bir ağın gerçek dünyadaki davranışını veya teorik tanımlamasını karşılayacak bir soyutlamayı içerir. Önerilen model soyutlamasının yeterli, yapılan varsayımların doğru olduğu kabulü kavramsal model geçерlemesi ile gösterilir. D-DEVSNET kavramsal modeli düzeyinde yapılan doğrulama ve geçерleme, bu modeli esas alarak gerçekleştirilen benzetim aracının da doğru ve geçerli olmasını sağlayacaktır.

Teori geçерlemesi, geliştirilen modelin teknik detayları ile ilgilendir. Teori geçерlemesinde OSPF protokolünün RFC 2328 dokümanında açıklanan standartları ve teorik yaklaşımları karşıladığı, yani algoritmanın doğruluğu test edilir.

İşlemsel geçерleme ile de RFC dokümanlarında açıklanan OSPF sürümüne olan benzerliği bilinen ağ benzetim araçları (ns-2, OPNET) ile karşılaştırılarak geçerliliği test edilmiştir.

Yapılan çalışmada hem teori hem de tasarım doğrulaması olmak üzere iki yönlü bir doğrulama söz konusudur. Bu doğrulama testleri, doğru ve kabul edilebilir model tasarımı için önemlidir. Teori doğrulaması kavramsal modelde tanımlanan model davranışının doğrulamasını, tasarım doğrulaması ise özelleştirilen DEVS-Suite benzetim aracının doğrulamasını içermektedir. Yani hem protokol hem de benzetim aracının doğruluğu geçerleme testleri ile gösterilmeye çalışılmıştır.

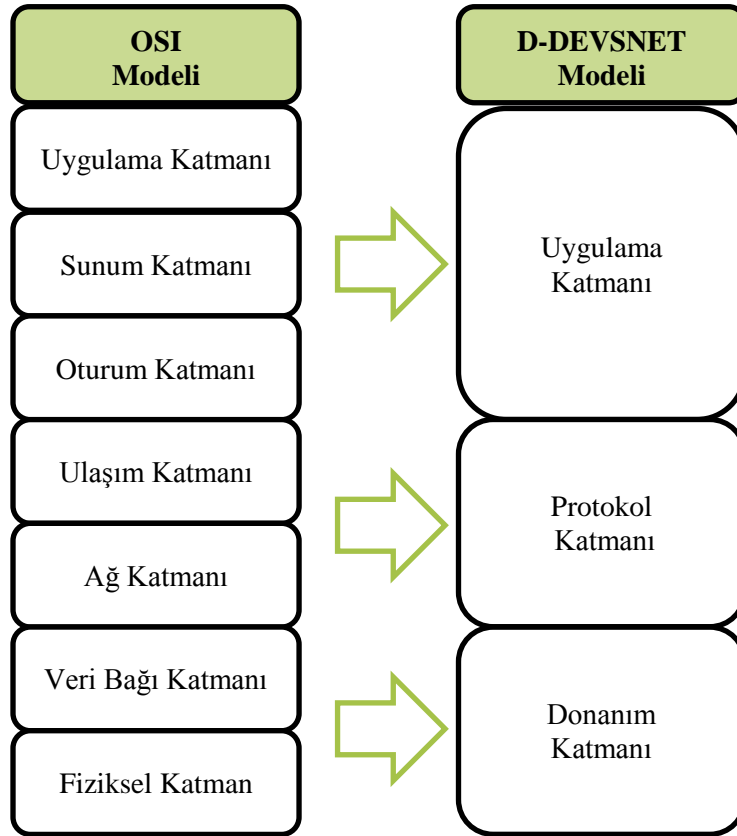


Şekil 7.1. D-DEVSNET OSPF protokolü doğrulama ve geçerleme süreci

Deneysel hedefler ile geçişlemin gereksinimleri arasında kuvvetli bir ilişki vardır. Geçerleme her zaman hedefler ve planlar ile ilişkilidir. Şekil 7.1’de kesik çizgiler karşılaştırma, değerlendirme veya ölçümleri göstermektedir. Ayrıca model ile deneysel çerçevenin uyumlu olması gerekir. Bunun için genellikle model, kendi deneysel çerçevesiyle eş zamanlı olarak tasarlanır [102].

### 7.3. D-DEVSNET Kavramsal Modelinin Geçerlenmesi

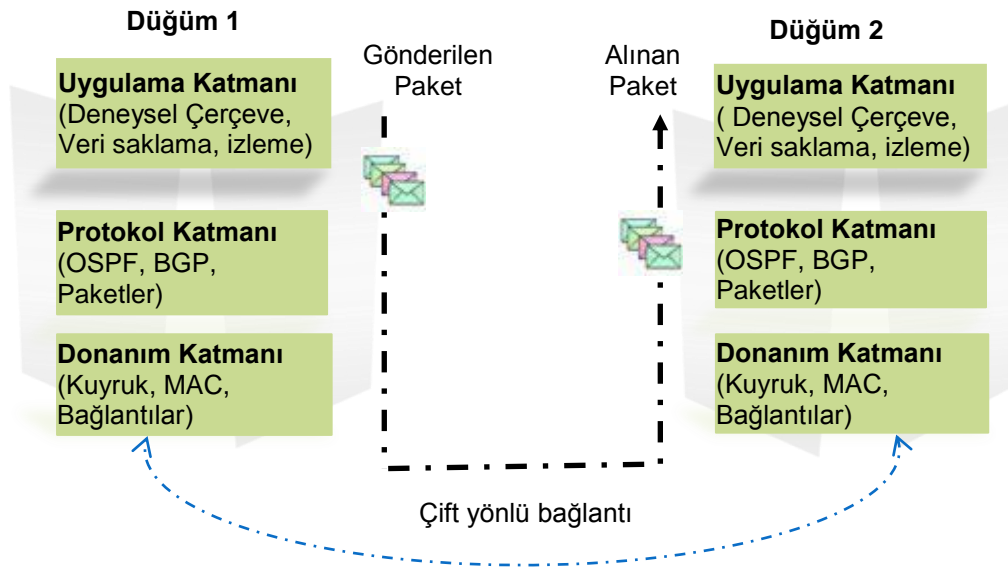
Modelleme ve benzetim sisteminin kavramsal modeli, geliştiricinin dilediği herhangi bir soyutlama seviyesinde hazırlanabilir [103]. Geliştirilen benzetim aracının geniş ölçekli bir ağı gerçek dünyadaki davranışını veya teorik tanımlamasını karşılaması beklenir.



Şekil 7.2. OSI referans modelinin D-DEVSNET eşdeğeri

Geliştirilen benzetim aracının yedi katmandan oluşan OSI referans modeline göre çalışması hedeflenmiştir. D-DEVSNET benzetim aracının ana hedeflerine (protokol gerçekleştirimi ve eğitim) uygun olarak yedi katmanlı OSI referans modeli, üç katmanda soyutlanmıştır (Şekil 7.2). Buna göre;

- Donanım katmanı, OSI modelinin fiziksel ve veri bağı katmanlarına,
- Protokol katmanı, OSI modelinin ağ ve ulaşım katmanlarına,
- Uygulama katmanı, OSI modelinin oturum, sunum ve uygulama katmanlarına, karşılık gelmektedir.



Şekil 7.3. D-DEVSNET ağ katman yapısının kavramsal modeli

Şekil 7.3'te geliştirilen ağ benzetim aracının OSI referans modeline karşılık gelen ağ katman yapısının kavramsal modeli görülmektedir. Uygulama katmanı verilerin saklanması ve izlenmesini içeren deneySEL çerçeve ile Java soket yapısını içermektedir. Protokol katmanı yönlendirme işlemini yapan OSPF, BGP protokollerini, IP ve veri ve kontrol paketlerini içerirken, donanım katmanı ağ arabirim kartı (kuyruk yapısı, giriş – çıkış portları gibi) ve bağlantılarını içermektedir.

#### 7.4. D-DEVSNET Protokol ve Benzetim Aracı Geerlemesi

Genellikle bir ađın bařarımı lölürken iř ıkarma yeteneđi, utan uca ortalama gecikme ve kayıp paket miktarı gibi bazı lütlere bakılır [28].

İř ıkarma yeteneđi (throughput): Birim zamanda bir ađda gelen-giden paketlerin sayısı, bir bařka ifade ile bilgi miktarıdır. Bir ađın iř ıkarma yeteneđi ařađdaki formül ile hesaplanır;

$$\text{Toplam iř} = \text{Hedefe ulařan paket sayısı} / \text{zaman};$$

Utan uca ortalama gecikme: Utan uca ortalama gecikme, hedefine ulařan bir paketin, hedefe ulařma süresini belirlemede kullanılan ve ađın genel durumu hakkında bilgi veren önemli bir parametredir. Ortalama gecikme, kaynak düđümünden ıkan veri paketlerinin hedef düđümlere varıř sürelerinin ortalamasıdır. Ađdaki her bir düđümün gönderdiđi veri paketlerinin, kaynaktan ıktıkları zaman ile hedef düđüme ulařtıkları zaman farkı belirlenir. Belirlenen zaman farklarının ortalaması alınarak, utan uca paket gecikmesi hesaplanır.

Kayıp paket sayısı: Kayıp paket sayısı, ađda hedefine ulařamayan paketlerin sayısıdır. Kaynak düđümlerden ıkan tüm veri paketlerinden hedefine ulařamayanların yüzde cinsinden oranı hesaplanarak belirlenir. Ađda düđümlerin kuyruklarında bekleyen paketlerin toplam sayısı, ađ trafiđinin seviyesini belirlemeye ve tıkanıklıkların tespitine yardımcı olur.

Kayıp paket oranı ařađdaki formül ile hesaplanır:

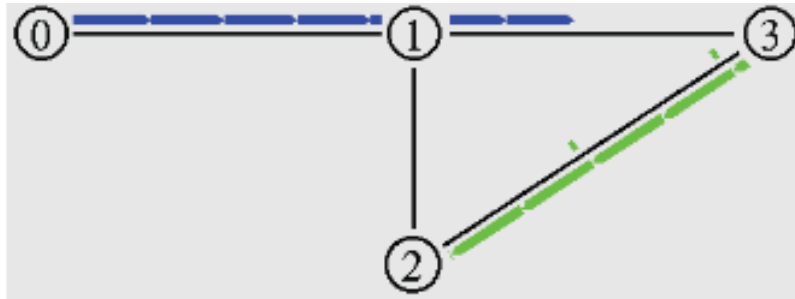
$$\text{Üretilen paket sayısı} - \text{Hedefine ulařan paket sayısı} / \text{Üretilen paket sayısı} * 100;$$

Geliřtirilen benzetim aracının dođrulanması ve geerlemesi için geliřtirilen model, ns-2 ile karřılařtırıldı. DEVS yaklařımından yararlanılarak OSPF protokolünün tasarlandıđı 4 düđümlü bir DEVS modeli (řekil 7.4b), yine 4 düđümlü ns-2 modeli

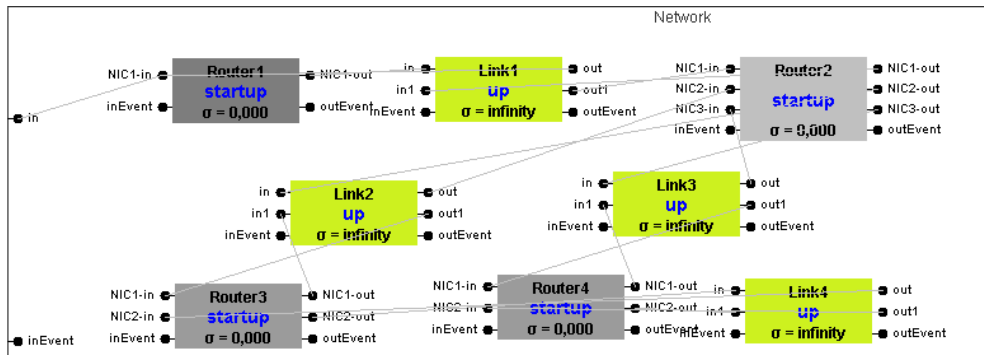
(Şekil 7.4a) ile Tablo 7.1'deki parametreler dikkate alınarak karşılaştırıldı ve iş çıkarma yetenekleri, ortalama gecikme ve kayıp paket sayıları ölçülerek test edildi.

Tablo 7.1. D-DEVSNET ve ns-2 ağ benzetim parametreleri

Benzetim Parametreleri	D-DEVSNET modeli	ns-2 modeli
Topoloji	4 adet yönlendirici ve çift yönlü bağlantı	4 adet yönlendirici ve çift yönlü bağlantı
Protokol	OSPF	Bağlantı durumu
Düğüm İşlem Süresi (msn)	1	-
Paket Boyutu (byte)	552	552
Bant Genişliği (Mbps)	2	2
Gecikme (msn)	1	1
Trafik	Rasgele	TCP üzerinden FTP
Kuyruk Limiti (paket)	362	20
Süre (sn)	10	10



(a)

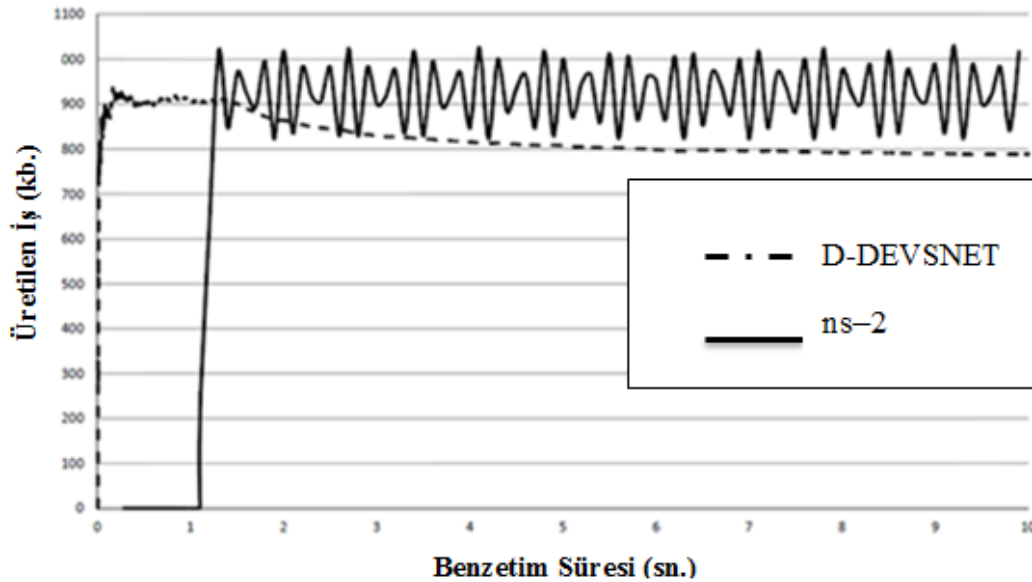


(b)

Şekil 7.4. (a) 4 düğümlü ns-2 OSPF ağı, (b) DEVS ağı



Şekil 7.5'te görüldüğü üzere, kısa süreli bir kararlı geçiş zamanından sonra (yaklaşık 2 saniye), ns-2 için yaklaşık ortalama iş çıkarma yeteneği 822.4 KB/sn, D-DEVSNET için ise 1489 paket \*552 byte = 821.9 KB/sn' dir. Ağ çıkışına hızlı bir şekilde ulaşılması ve çıkışın çalışma boyunca sabit kalması, OSPF algoritmasının yük dengeleme davranışının göstergesidir. Bununla birlikte, düğümlerin sınırlı sayıda bağlantı elemanı kullanılarak birbirine bağlı olmasına ve ağır trafik şartlarına rağmen OSPF uygulamasında herhangi bir paket kaybı yaşanmamaktadır. Aşırı trafik artışlarına sistem hızlı cevap verdiği için dolayı ns-2 ağ çıkışında görülen aşırı dalgalanmalar D-DEVSNET'de görülmemektedir [33].

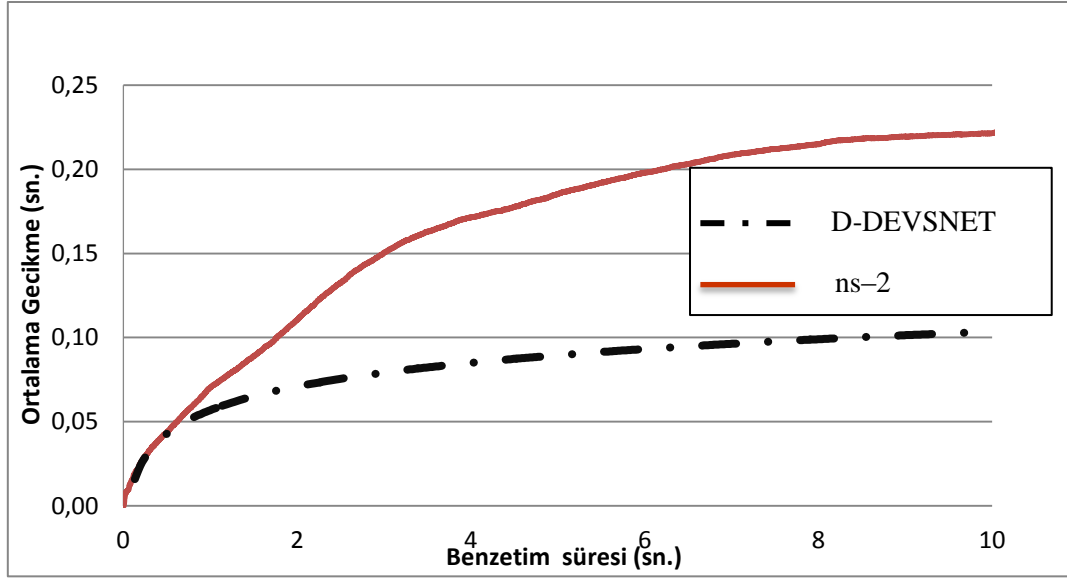


Şekil 7.5. Test Edilen Ağ Benzetim Araçlarının Ortalama Ağ Çıkışı

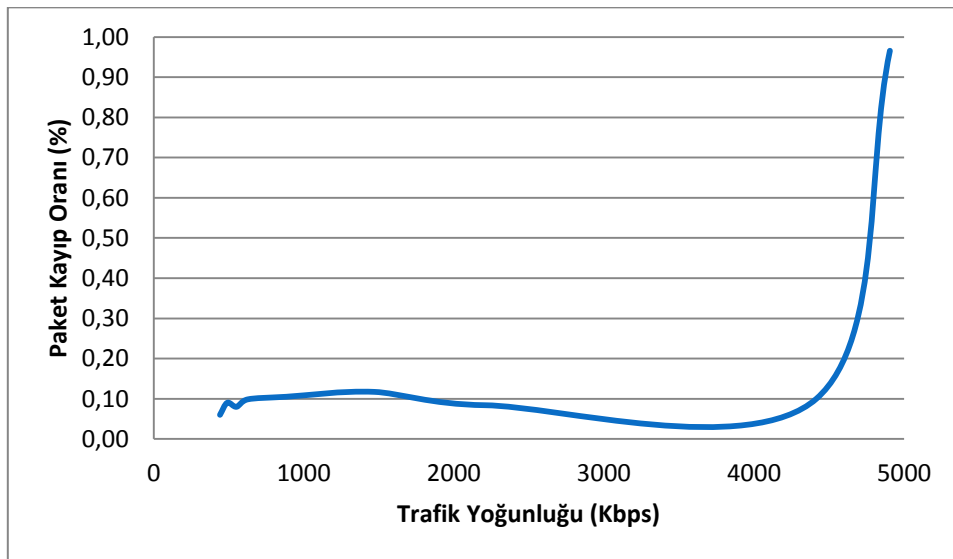
Şekil 7.5'teki ortalama iş çıkarma yeteneği grafiğinden de görüldüğü üzere yapılan testlerde ağ benzetimi için geliştirilen D-DEVSNET benzetim aracının, diğer ağ benzetim araçları ile benzer ağ çıkışlarına sahip olduğu görülmüştür.

Bir başka önemli başarımlı ölçütü uçtan uca gecikmedir. Şekil 7.6'da 10 saniye boyunca ağ üzerinden geçen paketlerin maruz kaldığı ortalama gecikmenin zaman üzerinde değişim grafiği görülmektedir. Ortalama gecikme; paket üreticinin paketi ürettiği zaman ile paketin hedefe vardığı zaman arasındaki fark (paketin yaşam süresi) olarak hesaplanmaktadır. Ortalama gecikme grafiğinden de görülebileceği gibi ns-2 ve D-DEVSNET benzetim araçlarındaki gecikme değerleri arasındaki fark

büyükür. Bu durum, yönlendiricilerin işlem süresi deęerinden kaynaklanmaktadır. Bu deęer ns-2 düęümleri için sıfır iken D-DEVSNET’de sıfırdan büyük pozitif bir deęerdir (1 msn). Bu sebeple ortalama gecikme farklı olarak ortaya çıkmıştır. Gecikmenin tutarlı ve kabul edilebilir bir seviyede kalması, aęın tıkanıklıkları yönetebildięinin bir göstergesidir.



Şekil 7.6. Test edilen aę benzetim araçlarına ait uçtan uca ortalama gecikme



Şekil 7.7. 3000 düęümlü bir D-DEVSNET aęı için paket kaybı

Şekil 7.7’de geliştirilen benzetim aracının uç / sınır değerlerdeki davranışını test etmek amacıyla 3000 düğümlü farklı trafik koşullarındaki (442Kbps–4,5Mbps arasındaki) bir ağda paket kaybı incelenmiştir. Bu grafiğe göre D-DEVSNET modelinde, yaklaşık 4,4 Mbps’ da trafik yoğunluğu kararlı hale geliyor. Bu değerden sonra model doğruluğunu ve kararlılığını kaybediyor.

Tablo 7.2. Karşılaştırılan benzetim araçlarının olay sıklığı

Benzetim Aracı	D-DEVSNET	ns-2	OPNET
Olay Sayısı	1000	28388	15942

Tablo 7.2’de karşılaştırılan benzetim araçlarının olay sıklığı verilmiştir. Buna göre 10 saniyelik benzetim süresinde D-DEVSNET’de 1000 olay olurken ns-2’da 28388, ve OPNET’de ise 15942 olay kaydedilmiştir [5]. D-DEVSNET, gereksiz ayrıntıların göz ardı edildiği yüksek soyutlama seviyesine (daha az karmaşıklığa) sahip bir ağ benzetim aracı olduğu için olay sayısı oldukça düşüktür (ns-2’ye göre 30 kat daha az). Tablo 7.2’den görüleceği üzere aynı işi yapmak için D-DEVSNET, ns-2 ve OPNET’e göre çok daha az olay üretir (olaya gereksinim duyar), böylece büyük boyutlu uygulamalarda doğruluktan fazlaca ödün verilmeden bellek ve işlemci kullanımı minimize edilmiş olur.

## **BÖLÜM 8. BENZETİM SONUÇLARI**

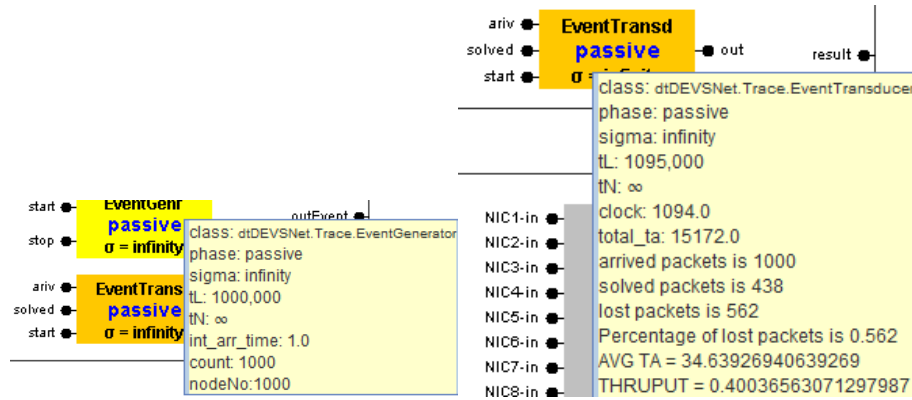
### **8.1. Giriş**

Bu tez çalışmasında, paralel DEVS mimarisi ile oluşturulmuş DEVS-Suite modelleme ve benzetim ortamında, Java programlama dili kullanılarak, istemci ve sunucu taraflı çalışan klasik bir ağ sistemi üzerinde benzetim çalışmalarına imkân veren paralel ve dağıtık yapıda D-DEVSNET isimli bir ağ benzetim aracı geliştirilmiştir. Geliştirilen bu benzetim aracı üzerinde OSPF ve BGP protokollerinin gerçekleştirilmesi yapılmıştır. Geniş ölçekli bir ağı oluşturan, dağıtık mimariye sahip ağ bileşenleri DEVS modelleme ve benzetim yaklaşımı kullanılarak tasarlanmış ve elde edilen modelleme ve benzetim ortamına ‘D-DEVSNET’ adı verilmiştir.

Geliştirilen D-DEVSNET ağ benzetim aracının doğrulama ve geçerlemesine yönelik benzetim çalışmaları bir önceki bölümde anlatılmıştır. Bu bölümde ise D-DEVSNET ağ benzetim aracının başarımları analizleri ve test sonuçları yer almaktadır. Yapılan çalışma ile farklı ölçeklerde, farklı topolojilerde ve farklı ağ trafikleri altında üretilen senaryolar sonucunda elde edilen veriler ve gözlemlenen çıkışlar grafikler halinde sunulmuş ve D-DEVSNET ağ benzetim aracının geniş ölçekli ağları, modelleme ve benzetim kabiliyetine sahip olduğu gösterilmeye çalışılmıştır.

Benzetim deneyleri Tablo 8.1’ de açıklanan parametreler esas alınarak, Intel Core i7, 1.73GHz, 4GB RAM, 64 bit Windows 7 / Linux ve Intel Core i5, 2.0GHz, 2GB RAM, 64 bit Windows 7 / Linux bilgisayarlarda, JRE 1.7 ile test edildi. Bilgisayarlardan biri istemci diğeri sunucu olarak dağıtık mimaride çalıştırıldı. Gerçekleştirilen benzetim sürecinde, oluşturulan benzetim ortamına ait parametreler, elde edilen veriler (istatistikler, ortalama ağ çıkışı, paket kaybı, vb.) ve ağda iletilen mesajlara ait bilgiler olay dönüştürücü tarafından yorumlanarak CSV (virgülle

ayrılmış değerler - comma seperated values) uzantılı dosyada (output.csv) kaydedildi ve grafikler halinde sunuldu. Ayrıca D-DEVSNET ağ benzetim ortamında kaydedilen tüm bu istatikselsel verileri, benzetim aracı çalışırken yönlendirici veya olay dönüştürücü üzerine farenin odaklanması ile anlık olarak izleyebiliriz. Şekil 8.1'deki ekran görüntüsünde 1000 yönlendiriciye sahip örnek bir ağda üretilen paket sayısı, kayıp paket sayısı, ortalama gecikme ve üretilen iş miktarlarını görmekteyiz.



Şekil 8.1. 1000 yönlendiricili örnek bir ağda deneysel çerçeve üzerinde izlenen veriler

Tablo 8.1. Yapılan deneylerde uygulanan benzetim parametreleri

Benzetim Parametreleri	D-DEVSNET	İNTERNET
Harici yönlendirme protokolü	BGP	BGP
Dâhili yönlendirme protokolü	OSPF	OSPF, RIP
IP Adres modeli	IPv4	IPv4, IPv6
IP Adres boyutu (byte)	4	4
Mesaj Tipi	Geliştirilmiş IP Paketi	IP Paketi
MTU (byte)	552	IPv4 için 576 IPv6 için 1280
Paket başlık boyutu (byte)	20	20
Maksimum LSA paket ömrü	Sonsuz	1 Saat
Atlama sayısı	15	15
Düğüm işlem zamanı	4.5 Kb/s	1 Mb/s - 40 Gb/s
Kuyruk Uzunluğu (paket)	362	32-500
Bağlantı Gecikmesi (msn)	3	3-16
Otonom Sistem Sayısı	2	~40000
Ölçek	~5.000 yönlendirici	~900.000 düğüm
Benzetim süresi (sn)	1000	-

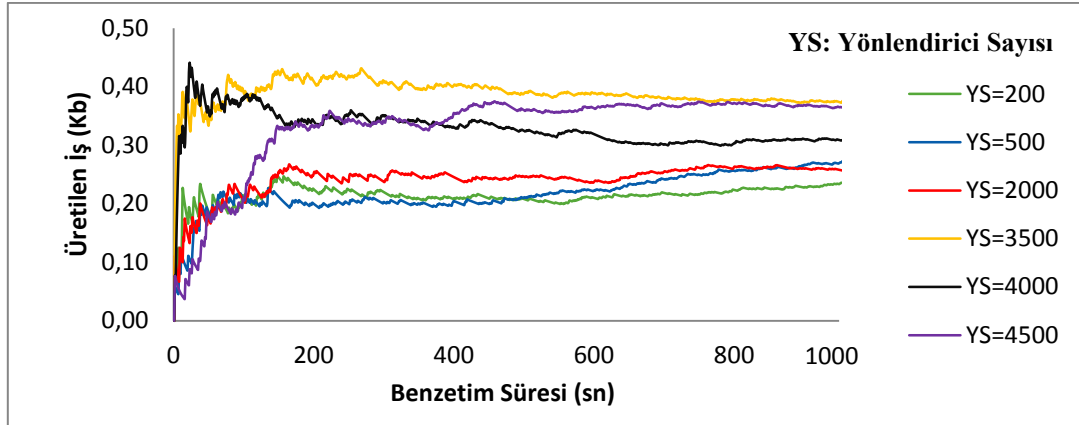
Tablo 8.1’de geliştirilen istemci / sunucu taraflı, paralel ve dağıtık ve geniş ölçekli D-DEVSNET ağ benzetim aracının, benzetim parametreleri ile İnternet sistemi karşılaştırılmıştır [104]. Örneğin, kuyruk uzunluğu, gecikme ve bant genişliği İnternet ölçeği ile aynı seçilmiştir. Özellikle IP adresi, protokoller ve veri paketleri gerçek ölçeklerde seçilmeye çalışılmıştır. Bununla birlikte benzetim ortamının bütün parametrelerinin gerçek dünya parametreleri ile birebir aynı olmasını beklemekte çok gerçekçi olmayacaktır. Bu parametreler kullanılarak D-DEVSNET ağ benzetim aracının başarımları analizleri yapılmıştır.

## 8.2. Başarım Değerlendirmesi

D-DEVSNET ağ benzetim aracının, Bölüm 7.4’de verilen değerlendirme kriterlerinden iş çıkarma yeteneği, paket kayıp oranı, uçtan uca ortalama paket gecikmesi ve bellek tüketimi açısından başarımları ölçümleri yapılmıştır. Başarım ölçümleri farklı ölçeklerdeki (50, 100, 200, 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000 ve 4500 yönlendiricili) ağlar üzerinde Tablo 8.1’deki benzetim parametrelerine göre yapılmıştır.

### 8.2.1. İş çıkarma yeteneği

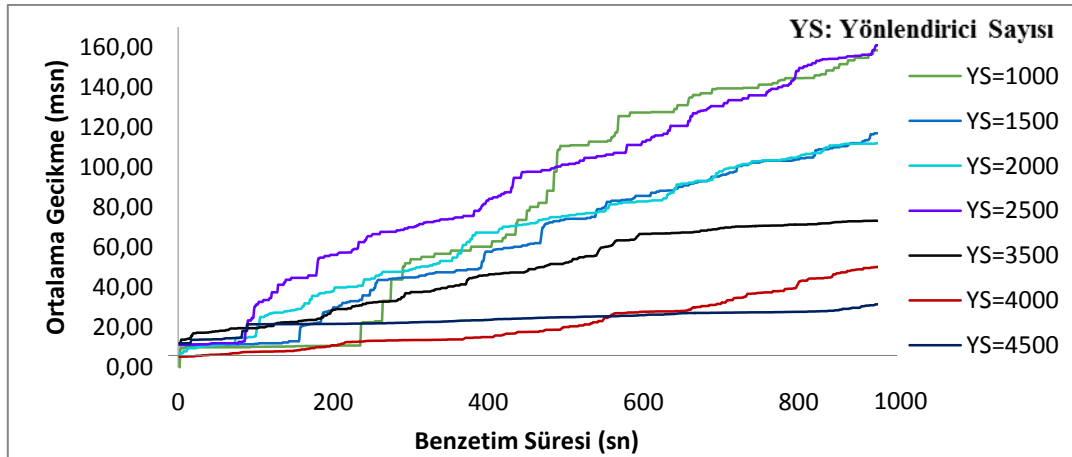
Yönlendirme algoritmalarının test edilmesi işleminde, birim zamanda bir ağda gelen-giden paketlerin sayısını gösteren iş çıkarma grafiği önemli bir ölçüt olarak kullanılır. Farklı düğüm / yönlendirici (50, 100, 200, 500, 2000, 3500, 4000, 4500) sayısına sahip, dağıtık bir ağa ait iş çıkarma yeteneği grafiği Şekil 8.2’de görülmektedir. Şekilde düğüm / yönlendirici sayısı artarken iş çıkarma yeteneğinin de kademeli olarak arttığı görülmektedir. Özellikle 3000 düğüme kadar ki ağlarda üretilen iş daha azken 3500 düğüm ve üzerinde üretilen iş artmaktadır. Bunun sebebi farklı topolojilerde kullanılan trafiğin farklı olmasının yanında bellek ve işlemci kapasitesinin aynı kalmasıdır. Doğal olarak ağın işlem kapasitesinin artması iş çıkarma yeteneğinin de artmasına yol açmaktadır.



Şekil 8.2. Farklı ölçeklerde D-DEVSNET iş çıkarma yeteneği

### 8.2.2. Uçtan uca ortalama paket gecikmesi

Uçtan uca ortalama paket gecikmesi, kaynak düğümünden çıkan veri paketlerinin hedef düğümlere varış sürelerinin ortalamasıdır. Ağdaki her bir düğümün gönderdiği veri paketlerinin, kaynaktan çıktıkları zaman ile hedef düğüme ulaştıkları zaman farkı belirlenir. Belirlenen zaman farklarının ortalaması alınarak, uçtan uca paket gecikmesi hesaplanır.

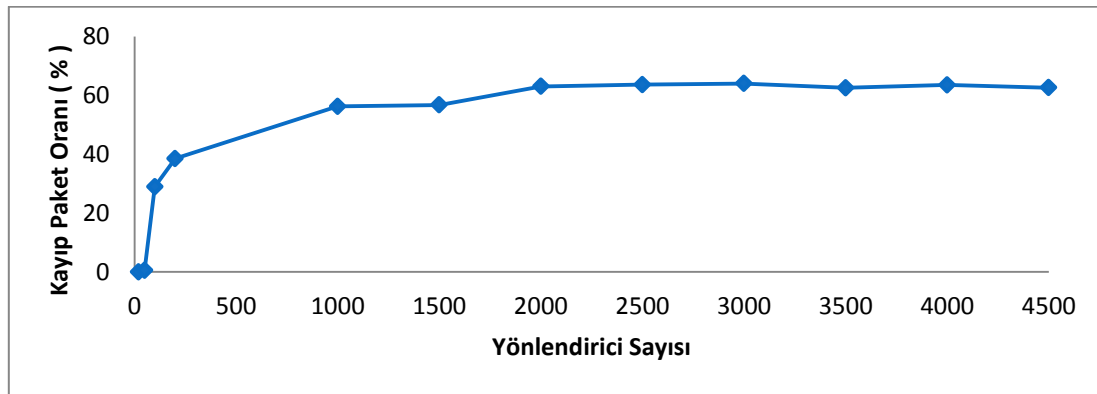


Şekil 8.3. Farklı ölçeklerde D-DEVSNET uçtan uca ortalama gecikme

Şekil 8.3'te 1000 saniyelik zaman periyotunda farklı ölçeklerdeki ağlarda D-DEVSNET benzetim aracının uçtan uca ortalama paket gecikmesi verilmiştir. Buna göre ağın ölçeği büyüdükçe D-DEVSNET' de dikkate değer bir başarımlı kaybı oluşmadığı görülmektedir.

### 8.2.3. Kayıp paket sayısı

Kayıp paket sayısı, ağda hedefine ulaşamayan paketlerin sayısıdır. Kaynak düğümlerden çıkan tüm veri paketlerinden hedefine ulaşamayanların yüzde cinsinden oranı hesaplanarak belirlenir. Ağda düğümlerin kuyruklarında bekleyen paketlerin toplam sayısı, ağ trafiğinin seviyesini belirlemeye ve tıkanıklıkların tespitine yardımcı olur. Şekil 8.4'te düğüm sayılarına bağlı olarak kayıp paket sayısı verilmiştir. Bütün ağlarda üretilen paket sayısı 1000 dir. Şekil 8.4'ten de görüleceği üzere düğüm sayısı ile kayıp paket sayısı arasında doğru bir orantı vardır.



Şekil 8.4. Farklı ölçeklerde D-DEVSNET kayıp paket sayısı

### 8.2.4. Hız ve bellek tüketimi

Ağ benzetim araçlarının hız ve bellek tüketimi karşılaştırmasına yönelik birçok çalışma vardır. Bu konudaki literatür araştırmasında genelde farklı ağ benzetim araçlarına ait hız ve bellek tüketimi karşılaştırmaları yapılmıştır. Nicol [105], ns-2, JavaSim (J-Sim) ve SSFNet benzetim araçlarını 1000 saniyelik sürede bağlantı sayısını dikkate alarak (10-10000 arası) karşılaştırmıştır. Bağlantı başına harcanan bellek tüketimi Tablo 8.2'de verilmiştir. Nicol bu çalışmasında, ns-2 benzetim aracının daha hızlı fakat daha fazla bellek tüketimine sahip olduğunu belirtmiştir. Weingartner ve arkadaşlarının yapmış olduğu çalışmada [106] ns-2, ns-3, OMNeT++, SimPy ve JİST benzetim araçları 600 saniyelik sürede düğüm sayıları (4-3025 düğüm) dikkate alınarak karşılaştırılmıştır. Bu çalışma sonucunda ns-2, OMNET++ ve SimPy araçlarının benzer bellek tüketimine sahip olduğu, JİST'in



daha hızlı fakat daha fazla bellek tüketimine sahip olduğu vurgulanmıştır. Albeseder ve arkadaşlarının çalışmasında [107] OMNET++, ns-2, QualNet ve SimPy araçları çalışma zamanı süreleri dikkate alınarak karşılaştırılmıştır. Bu çalışmada da benzer sonuçlar elde edilmiş, özellikle OMNET'in hız bakımından başarımının iyi olmadığı vurgulanmıştır. Fujimoto ve arkadaşları [49], ns-2, PDNS ve GTNeTS benzetim araçlarını karşılaştırmış ve ns-2'nin daha hızlı çalıştığını belirtmiştir. Tüm bu çalışmalardan, daha hızlı çalışan benzetim araçlarının daha fazla bellek kapasitesine ihtiyaç duyduğunu söyleyebiliriz.

Tablo 8.2. Ağ benzetim araçlarının hız ve bellek tüketimi karşılaştırması

<b>Benzetim aracı</b>	<b>Bağlantı başına bellek tüketimi</b>
ns-2	93,3 KB
J-Sim	21,7 KB
SSFNet	53,3 KB
D-DEVSNET	42,8 KB

Tablo 8.2'deki verilerden de görüleceği üzere D-DEVSNET benzetim programı çalışırken ('hello' paketleri ağda dolaşırken) elde edilen bağlantı başına bellek tüketimi miktarı, ns-2 ve SSFNet'e göre daha iyidir. Yine yaptığımız testlerde bir bilgisayarda ns-2 ile en fazla 1500 düğüme çıkılabilirken D-DEVSNET ile yaklaşık 3000 - 3500 düğüme çıkılabilmektedir. Bu da D-DEVSNET'in iyi bir ölçeklenebilirliğe sahip olduğunu göstermektedir.

## BÖLÜM 9. SONUÇLAR VE DEĞERLENDİRME

### 9.1. Sonuçlar

Bu çalışmada, geniş ölçekli ağlarda çalışabilecek ayrık olay tabanlı yeni bir modelleme ve benzetim yaklaşımı ve ağ benzetim aracı geliştirilmiştir. Geliştirilen bu benzetim aracına ‘D-DEVSNET’ ismi verilmiş ve diğer ağ benzetim araçları ile geçerliliği test edilmiştir.

Sonuç olarak; geliştirilen D-DEVSNET ağ benzetim aracının, klasik heterojen yapıdaki geniş ölçekli bir ağ üzerinde, sağlıklı ve kolay benzetim çalışmaları yapılmasına imkân sağladığı görülmüştür. Geliştirilen D-DEVSNET ağ benzetim aracı aşağıdaki özelliklere sahiptir;

- Web destekli çalışabilme yeteneği: D-DEVSNET, ‘java webstart’ teknolojisine sahiptir.
- Benzetim adımlarının kolay takibi: Benzetim sonuçları her aşamada kolay izlenebilir.
- Görsellik: İyi bir görsel ara yüze sahiptir.
- Kolay kurulum: Java yüklü herhangi bir bilgisayarda “Çalıştır-Run” tuşuna basmak yeterlidir.
- Açık kaynak kodlu: Bütün bileşenleri Java programlama dili ile yazıldığı için ücretsiz, açık kaynak kodlu ve sıfır maliyete sahiptir.

- Modüler yapısı: Ağ bileşenleri modüler, ayrı ayrı sınıf (class) olarak tanımlanmıştır.
- Gelişmiş izleme ve analiz desteği: Sonuçlar anlık izlenebildiği gibi, kalıcı olarak da ‘csv’ uzantılı dosyalarda saklanabilmektedir.
- Ölçeklenebilirlik: Binlerce düğümlü ağların benzetimi yapılabilir. Düğüm sayısı artarken önemli bir başarımlı kaybı yaşanmamaktadır.
- Sistem teorisi tabanlı yaklaşım altyapısı: DEVS modelleme ve benzetim yaklaşımını kullanır.
- Paralel ve dağıtık çalışabilme yeteneği: Ağ bileşenleri birden fazla bilgisayar ve işlemci üzerinde eş zamanlı çalıştırılabilir.
- Esneklik: Rahatlıkla yeni ağ bileşenleri eklenebilir.
- Tümüleşik topoloji üretici desteği: BRITE topoloji üretici, gerçekçi modellerin oluşturulmasına imkân verir
- Yüksek Performans: Düşük güçlü bilgisayarlarda bile yüksek performanslı olarak çalıştırılabilir.
- Platform bağımsızlığı: Java Sanal Makinesinin (JVM) yüklü olduğu her ortamda çalışabilir.

Ayrıca geliştirilen D-DEVSNET benzetim aracı, dağıtık, ölçeklenebilir, adaptif ve güçlü ağ uygulamalarının modellenmesi / tasarımı için örnek bir çerçeve olmuş, DEVS yaklaşımının geniş bir uygulama alanına sahip olduğunu göstermiştir.

Geniş ölçekli, dağıtık D-DEVSNET benzetim aracının bilime katkılarını aşağıdaki şekilde sıralayabiliriz:

- Geniş ölçekli ağların benzetimi için dağıtık ve paralel mimaride ayrık olay tabanlı yeni bir yönlendirme algoritması geliştirilmiştir. Geliştirilen algoritma ile benzetim aracının ölçeklenebilirliğinin artırılması hedeflenmiştir.
- Dağıtık, geniş ölçekli ve sistem teorisi tabanlı modüler ve hiyerarşik bir benzetim aracı geliştirilmiştir. Bölüm 5’de genel bir sınıflandırması yapılan ağ benzetim araçlarının birçoğu sistem teorisi tabanlı değildir. Herhangi bir formal tanımdan yoksun bir benzetim aracı düşük başarıma kötü bir ölçeklenebilirliğe ve yeniden kullanılmayan yazılım mimarisine sahiptir. Benzetim aracı tasarımının formalizasyonu, sistem tasarımının uygun bir şekilde onaylanmasını ve geçerlemesini kolaylaştırmaktadır.
- Benzetim aracı, DEVS modelleme yaklaşımına sahip, %100 nesne tabanlı yapısı ile yeniden kullanılabilir bileşenlere sahip modeller geliştirilmesine imkân sağlar, böylece hazırlanan modellerden yeni modeller geliştirilerek aynı modelin birçok kez kullanıldığı (ağdaki düğüm, yönlendirici, vb) uygulamalarda büyük kolaylık sağlamaktadır. D-DEVSNET ortamını oluşturan bileşenlerin ve varlıkların, modüler bir yapıda tasarımını, yeniden kullanımını olanaklı kılmaktadır.
- D-DEVSNET benzetim aracı ile yapılan benzetim çalışmalarının her aşamada kolay izlenmesi, incelenen ağ sisteminin karakteristikleri hakkında bilgi sağlayacak sonuçların üretilmesini ve elde edilen sonuçların değerlendirilmesi ve yorumlanması işlemlerini kolaylaştırmaktadır.
- Java programlama dili ile geliştirilen D-DEVSNET benzetim aracı platformdan bağımsız (örneğin ns-2 benzetim aracı, iki programlama dili altyapısına sahiptir ve sadece Linux işletim sistemi altında çalışır) ve kolay kurulabilen bir altyapıya sahiptir.

- D-DEVSNET, 'Java Webstart' teknolojisine sahiptir. Bu sayede, benzetim aracını kullanan kişi, bir web sayfasını tıklama kolaylığında benzetimi çalıştırabilir, değişiklikleri izleyebilir. Ayrıca benzetim aracının İnternet üzerinden yayımlatılmasını ve kullanılmasını kolaylaştırır. Uzaktan erişilebilen benzetim araçlarına ihtiyacın her geçen gün arttığı günümüzde uygulamaların paylaşılmasını kolaylaştırır ve uzaktan eğitim için elverişli bir teknolojik altyapı sunar. Dolayısıyla gerçekleştirilen benzetim aracı, uzaktan eğitim uygulamalarında da rahatlıkla kullanılabilir. Bölüm 5'de ifade edilen J-Sim haricindeki benzetim araçlarının bu özellikten yoksun olması geliştirilen benzetim aracını benzerlerine kıyasla bir adım öne çıkarmaktadır.
- D-DEVSNET, özel amaçlı, ticari bir yazılım olmadığı için, son derece esnek bir mimariye sahiptir. Açık kaynak kodlu modüler bir yapıdaki benzetim aracına rahatlıkla yeni bileşenler ve araçlar eklenebilir. İlave edilecek farklı bileşenlerle, geliştirilen benzetim aracı sadece geniş ölçekli dağıtık ağ sistemlerini değil, aynı zamanda farklı teknolojilerin (kablosuz algılayıcı ağlar gibi) modelleme ve benzetiminin yapıldığı bir benzetim ortamına dönüştürülebilir.
- D-DEVSNET ortamı modellenen sistemin yapısal ve davranışsal durumunun izlenmesini, kullanıcıyı yönlendiren mesajlarla ve hiyerarşik bileşen yapılarının ekranda anlık adım adım takip edilmesi ile sağlamaktadır. Bu da geliştirilen benzetim aracında kurulan modellerin, daha iyi izlenmesini / analizini mümkün kılarak iyi bir eğitim aracı olmasını sağlamaktadır. Örneğin ns-2, D-DEVSNET ile karşılaştırıldığında kötü bir grafiksel kullanıcı arabirimine sahiptir.
- D-DEVSNET ortamı model üzerinde gelişmiş test ve deneylere imkân sunmaktadır. Ağ protokollerinin daha iyi başarımların analizinin yapılabilmesi için, detaylı izleme son derece önemlidir. Benzetim sonuçları anlık izlenebildiği gibi, kalıcı olarak da 'csv' uzantılı dosyalarda saklanabilmektedir. 'Time view' seçeneği, zaman üzerinde modellerin davranışının izlenmesini sağlar. Bölüm 6'da detaylandırıldığı gibi, geliştirilen benzetim aracı, benzetim sonuçlarının kolay bir şekilde değerlendirilmesini mümkün kılmaktadır. Örneğin, PDNS

benzetim aracı ürettiği ASCII dosyaları üzerinde Perl kodları çalıştırarak benzetim sonuçlarının yorumlanmasını güçleştirmektedir.

- D-DEVSNET ortamına entegre edilen, BRITE topoloji üretici ile kod yazmaksızın geniş ölçekli topolojiler üretilebilmektedir. DEVS BRITE eklentisi geniş ölçekli ağların kolayca kurulup analizine olanak tanımaktadır. BRITE topoloji üretici, otonom sistemlerin güç kanunlarına dayanır ve gerçekçi modellerin oluşturulmasına imkân verir. Bölüm 7’de D-DEVSNET benzetim aracının başarımını ölçmek amacıyla, farklı ölçeklerde ağ modelleri oluşturulup, OSPF ve BGP protokolleri test edilmiştir.
- D-DEVSNET, paralel ve dağıtık çalışabilme yeteneğine sahiptir. DEVS yaklaşımı ve dağıtık istemci/sunucu mimarisi ile ağ bileşenleri kolayca coğrafik olarak farklı ağlar üzerine bölünebilecek ve böylece oldukça büyük ölçekte ağ modellerinin benzetim çalışması yapılabilecektir. Ayrıca, DEVS modelleme yaklaşımının, birleşim altında kapalılık özelliği; modüler ve hiyerarşik modeller kurmaya imkân sağlaması nedeniyle, oluşturulan ağların, uygun ara birimlerle başka ağlara bağlanarak hiyerarşik modüllere sahip bir ağ sisteminin tasarlanmasına imkân vermektedir.
- D-DEVSNET, iyi bir ölçeklenebilirliğe sahiptir ve başarım testlerinden de görüleceği üzere, benzetim aracı performansı, ağdaki düğüm sayısından (artmasından veya azalmasından) çok fazla etkilenmemektedir.

## 9.2. Tartışma ve Öneriler

Geliştirilen D-DEVSNET modelleme ve benzetim ortamı, klasik heterojen yapıdaki ağlarının yönetimi ve protokol testleri ile ilgili konuları çalışmaya uygundur. D-DEVSNET benzetim ortamında bir ağı oluşturan bileşenler belirli bir soyutlama seviyesinde tutulmuş ve kabullenmeler yapılarak sistemler tasarlanmıştır. Tasarlanan bu sistemler üzerinde, DEVS-Suite ortamının ve Java dilinin gelişmiş, esnek yapısı

kullanılarak yeni uygulamalar geliştirilebilir ve yeni çalışmalar yapılabilir. Geliştirilebilecek uygulamaların / çalışmaların bazıları aşağıda sunulmaktadır.

DEVS yaklaşımı ve istemci-sunucu mimari yaklaşımının avantajlarını kullanarak yeni bir D-DEVSNET benzetim aracı geliştirilmiştir. Geliştirilen benzetim aracında OSPF protokol uygulaması yapılmış ve geniş ölçekli ağda yönlendirme modellemesi yapılmış, diğer bazı özellikleri basit kabullenmeler ve soyutlamalara tabii tutulmuştur. Bu soyutlamaların gerçeğe uygun bir şekilde modellenmesi ile geniş ölçekli bir ağın tüm yönlerini değerlendiren eksiksiz bir D-DEVSNET benzetim aracı geliştirilebilir.

D-DEVSNET benzetim aracının geçerlemesi ve doğrulaması geniş ölçekli ağlarda kullanılan bir yönlendirme protokolü ile yapılmıştır. Daha sağlıklı bir geçeleme ve doğrulama için geliştirilen benzetim aracının soyutlama ve kabullenme seviyelerini günümüzde çok yaygın kullanılan benzetim araçları ile eşit düzeye getirerek, eşit şartlarda yapılacak deneyler sonucunda, çıkan sonuçların karşılaştırması, geliştirilen benzetim aracının doğrulama ve geçerlemesine katkı sağlayacaktır. Sadece OSPF değil diğer yönlendirme protokollerinin (RIP gibi), kablosuz ve sensor ağlarda kullanılan protokollerin benzetim ortamına eklenmesi ile genel bir ağ benzetim ortamı gerçekleştirilebilir. Böylece D-DEVSNET benzetim aracının kullanımının yaygınlaşması sağlanabilir.

D-DEVSNET ortamına ilave edilecek değişik bileşenlerle, geliştirilen benzetim aracı, sadece klasik ağ sistemlerini değil, kablosuz algılayıcı ağlar gibi farklı paralel ve dağıtık çalışan sistemlerin modelleme ve benzetiminde kullanılabilir.

Bilgisayar kaynaklarının, kişi ve kuruluşlar tarafından İnternet üzerinden esnek, güvenli ve eşgüdömlü paylaşımı olarak tanımlanan Grid (yüksek başarılı bilgisayar) altyapısı kullanılarak, D-DEVSNET ortamında oluşturulan modellerin yüzlerce işlemci üzerine bölünmesi ve böylece çok geniş ölçekli (100 bin düğümlü) ağ modellerinin benzetim çalışması yapılabilir.

Java programlama dili kullanılarak gerçekleştirilen uygulamaların, web tarayıcılar altında rahatlıkla çalışabilmesi, D-DEVSNET benzetim aracının İnternet üzerinden yayınlanabilmesini / kullanılmasını kolaylaştırmakta, İnternet kullanılarak uygulamaların paylaşılması, uzaktan eğitim amaçlı uygulamalar için ideal bir altyapı oluşturmaktadır. Geliştirilen D-DEVSNET benzetim aracı, web ortamına taşınarak popüler tarayıcılar üzerinden modelleme ve benzetim çalışmalarına imkân sağlanabilir. Modelleme ve benzetim uygulamalarında web tabanlı araçların kullanım gereksinimi göz önüne alındığında yapılan çalışma web tabanlı geniş ölçekli ağ benzetim araçlarının tasarımına da öncülük edecektir.



## KAYNAKLAR

- [1] ROBERTS, L.G., The ARPANET and Computer Networks, A History of Personal Workstations, A. Goldberg, Adele, ed. New York: ACM, 1986
- [2] İnternet Sistemleri Konsorsiyomu, <http://www.isc.org/solutions/survey>, Erişim tarihi: Kasım 2011
- [3] ZENGİN, A., EKİZ, H., ÇOBANOĞLU, B., TÜNCEL, S., Ağların Eğitimi ve Araştırılması için DEVS Tabanlı Simülatör Tasarımı ve Uygulaması, 5. Uluslararası İleri Teknolojiler Sempozyumu (İATS'09), Karabük, Türkiye, 13-15 Mayıs 2009
- [4] FUJIMOTO, R.M., PERUMALLA, K.S., PARK, A., WU, H., AMMAR, M.H., RILEY, G.F., Large-Scale Network Simulation: How Big? How Fast?, Modeling, Analysis and Simulation of Computer Telecommunications Systems, 11th IEEE/ACM International Symposium on, MASCOTS 2003
- [5] ÇOBANOĞLU, B., ZENGİN, A., EKİZ, H., TÜNCEL, S., Ağ Benzetim Araçlarının Karşılaştırmalı Olarak İncelenmesi, Sayfa: 70-76, International Science and Technology Conference, Famagusta, October 27-29, Kıbrıs, 2010
- [6] NS-2 ağ benzetim aracı, <http://www.isi.edu/nsnam/ns>, Erişim tarihi: Kasım 2011
- [7] OPNET Modeler, <http://www.opnet.com/products/modeler/home.html>, Erişim tarihi: Kasım 2011
- [8] OMNET++ ağ benzetim aracı web adresi, <http://www.omnetpp.org>, Erişim tarihi: Ekim 2011
- [9] ZEIGLER, B.P., MITTAL, S., Modeling and Simulation of Ultra-large Networks: A Framework for New Research Directions, supported by NSF Grant ANI-0135530, ULN Workshop, July 2002
- [10] KIM, S., SARJOUGHIAN H. S., ELAMVAZHUTHI, V., DEVS-Suite: A Simulator Supporting Visual Experimentation Design and Behavior Monitoring, Spring Simulation Multiconference, Article no 161, San Diego, California, 2009
- [11] BRITE Web adresi: <http://www.cs.bu.edu/brite>, Erişim tarihi: Ekim 2011

- [12] RAHMAN, M.A., PAKŠTAS, A., WANG, F. Z., Network Modelling and Simulation Tools, *Simulation Modelling Practice and Theory*, 17, pp. 1011-1031, 2009
- [13] MALOWIDZKI, M., Network Simulators: A Developer's Perspective, *Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'04)*, pp. 412-419, July 2004
- [14] LI, Y., QIAN, D., A Practical Approach For Constructing A Parallel Network Simulator, presented at *Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies*, pp. 655-659, 27-29 Aug. 2003
- [15] BEGG, L., LIU, W., PAWLIKOWSKI, K., PERERA, S., SIRISENA, H., Survey of Simulators of Next Generation Networks for Studying Service Availability and Resilience, *Technical Report TR-COSC 05/06*, University of Canterbury, Christchurch, New Zeland, 2006
- [16] KIM, T., HWANG, M.H., KIM, D., DEVS/NS-2 Environment: An Integrated Tool for Efficient Networks Modeling and Simulation, *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 5, pp 33-60, 2008
- [17] SHAUKAT, K., SARJOUGHIAN, H., A Comparative Study on DEVS and ns-2 Simulation and Modeling, *Technical Report Arizona State University*, Tempe, AZ, 2006
- [18] ZANG, M., Toward A Flexible And Reconfigurable Distributed Simulation: A New Approach To Distributed DEVS, *PhD Thesis, The University Of Arizona*, 2007
- [19] SRIVRUNYOO, I., A Contextualized Web-Based Learning Environment for DEVS Models, *Master's thesis, Georgia State University*, 2007
- [20] LEE, J., LEE, M., CHI, S., DEVS/HLA-Based Modeling and Simulation for Intelligent Transportation Systems, *New Age International Simulation*, (DOI: 10.1177/0037549703040233), 2003
- [21] ZACHAREWICZ, G., ET AL., Distributed Simulation Platform To Design Advanced RFID Based Freight Transportation Systems, *Comput. Industry* (DOI:10.1016/j.compind.2011.04.009), 2011
- [22] WAUPOTITSCH, R., EIDENBENZ, S., KROC, L., SMITH, J., Multi-Scale Integrated Information and Telecommunications System (MIITS): First Results From A Large-Scale End-To-End Network Simulator, *wsc*, pp.2132-2139, *Proceedings of the 2006 Winter Simulation Conference*, (DOI:10.1109/WSC.2006.323013), 2006

- [23] GUPTA, E.M., KAUR, V., Open Shortest Path First Simulator By Using Java RMI, *International Journal of Engineering Science and Technology*, Vol. 2(6), pp. 2155-2163, 2010
- [24] JAAFAR, T.M., *Simulation-Based Routing Protocols Analyses*, PhD Thesis, Georgia Institute of Technology, 2007
- [25] VERNEZ, J., EHRENSBERGER, J., ROBERT, S., Nessi: A Python Network Simulator for Fast Protocol Development, *Computer-Aided Modeling, Analysis and Design of Communication Links and Networks*, pp: 67-71, 11th International Workshop, 2006
- [26] LIU, J., LI, Y., HE, Y., A Large-Scale Real-Time Network Simulation Study Using Prime, in *Proceedings of the Winter Simulation Conference (WSC'09)*, 2009
- [27] MAGRANER, J.L.J., *A Network Simulator Based on OSPF Routing Protocol for Network Design and Educational Purposes*, Yüksek Lisans Tezi, University Of Puerto Rico, 2007
- [28] TÜNCEL, S., *Gezgin Ağ Yönlendirme Protokolleri için DEVS Tabanlı Benzetim Aracı Tasarımı*, Doktora Tezi, Fen Bilimleri Enstitüsü, Sakarya Üniversitesi, 2010
- [29] VICTORINO, S., ALFONSO, U., CELLIER, F.E., SEBASTIAN D., *System Modeling Using The Parallel DEVS Formalism and The Modelica Language*, *Simulation (Modelling) Practice and Theory*, Volume 18, pp. 998-1018, 2010
- [30] BOUKERCHE, A., ZHANG, M., SHADID, A., *DEVS Approach to Real-time RTI Design for Large-Scale Distributed Simulation Systems*, DOI: 10.1177/0037549708092223, *SIMULATION* 84: 231, 2008
- [31] BYRNE, J., HEAVEY, C., BYRNE, P.J., *A Review of Web-Based Simulation and Supporting Tools*, *Simulation Modelling Practice and Theory* Volume 18, Issue 3, pp. 253-276, 2010
- [32] ZENGİN, A., *Large-Scale Integrated Network System Simulation with DEVS-Suite*, *KSII, Transactions on Internet and Information Systems* Vol. 4, No. 4, 2010
- [33] ZENGİN, A., EKİZ, H., ÇOBANOĞLU, B., TÜNCEL, S., *A Simulation Study Of The OSPF Protocol in DEVS-Suite*, *ETAI*, 26-29 September, Ohrid, Macedonia, 2009
- [34] BANKS, J., CARSON II, J.S., NELSON, B.L., NICOL, D.M., *Discrete-Event System Simulation*, 4rd edition, Prentice-Hall Inc., 2001

- [35] BAYILMIŞ, C., IEEE 802.11B KLAN Kullanarak CAN Segmentleri Genişleten Arabağlaşım Birimi Tasarımı, Doktora Tezi, Fen Bilimleri Enstitüsü, Kocaeli Üniversitesi, 2006
- [36] FIRAT, C., Simülasyon Sistemlerinde DEVS Tabanlı Kavramsal Modelleme ve Değerlendirme, Doktora Tezi, Fen Bilimleri Enstitüsü, Sakarya Üniversitesi, 2004
- [37] ÖZÇELİK, İ., CAN/ATM ve PROFIBUS/ATM Yerel Köprülerinin Tasarımı, Doktora Tezi, Fen Bilimleri Enstitüsü, Sakarya Üniversitesi, 2002
- [38] ERTÜRK, İ., Internetworking Between ATM LANs and Legacy LANs over ATM Networks, PhD Thesis, The School of Engineering and Information Technology, Sussex University, England, 2000
- [39] ZENGİN, A., Dağıtık Simulasyon Sistemleri için Yeni Bir Yönlendirme Algoritması ve Uygulaması, Doktora Tezi, Fen Bilimleri Enstitüsü, Sakarya Üniversitesi, 2004
- [40] ERÇELEBİ, T., Yüksek Düzeyli Mimarının, Koşut İşlem Gereksinim Olan Bilimsel Benzetim Uygulamaları İçin Uygunluğunun İncelenmesi, Yüksek Lisans Tezi, Hacettepe Üniversitesi, 2008
- [41] MACKULAK, G.T., COCHRAN, J.K., SHUNK, D.L., Generic System Simulation Development on the PC (Phase I-III), Tempe, Arizona: Systems Simulation Laboratory, Arizona State University, 1987
- [42] SHANNON, R.E., Systems Simulation the Art and Science. Englewood Cliffs, New Jersey: Prentice-Hall., 1975
- [43] LAW, A.M., KELTON, W.D., Simulation, Modeling and Analysis, 3rd Edition, McGraw-Hill, 1999
- [44] ZEIGLER, B.P., PRAEHOFER, H., KIM, T.G., Theory of Modelling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems, Academic Press, Second Edition, 2000
- [45] ZEIGLER, B.P., KRIEGER, R.E., Theory of Modelling and Simulation, Malabar, Florida, 1984
- [46] BALCI, O., Principles and Techniques of Simulation Validation, Verification and Testing, Proceedings of the 1995 Winter Simulation Conference, pp. 147-154, 1995
- [47] MARIA, A., Introduction to Modeling and Simulation, Proceedings of the Winter Simulation Conference, pp. 7-13, 1997
- [48] VANGHELUWE, H., Multi-Formalism Modelling and Simulation, PhD Thesis, Universiteit Gent Faculteit Wetenschappen, 2001

- [49] FUJIMOTO, R.M., PERUMALLA, K.S., RILEY, G.F., Network Simulation, Morgan & Claypool, 2007
- [50] PARK, S., KIM, S.H., HUNT, C.A., PARK, D., DEVS Peer-to-Peer Protocol for Distributed and Parallel Simulation of Hierarchical and Decomposable DEVS Models, pp. 91-95, Univ. of California, San Francisco, ISITC, 2007
- [51] NUTARO, J., Adevs: A Discrete Event System Simulator, <http://www.ornl.gov/~lqn/adevs/index.html> , Erişim tarihi: Kasım 2011
- [52] ZEIGLER, B.P., MOON, Y., KIM, D., KIM, J.G., DEVS-C++: A High Performance Modelling and Simulation Environment, Twenty-ninth Hawaii International Conference on System Sciences, pp. 350-359, 1996
- [53] DEVS-Java Reference Guide: Artificial Intelligence and Simulation Research Group Department of Electrical and Computer Engineering The University of Arizona, <http://www.cs.gsu.edu/~fbai1/paper/4560.pdf>, Erişim tarihi: Kasım 2011
- [54] HIMMELSPACH, J., EWALD, R., UHRMACHER, A.M., A Flexible and Scalable Experimentation Layer for JAMES II, Proceedings of the Winter Simulation Conference. Edited by S. Mason, R. Hill, L. Moench, and O. Rose, pp. 827–835, Winter Simulation Conference, December 2008
- [55] SmallDEVS, [www.fit.vutbr.cz/~janousek/smalldevs](http://www.fit.vutbr.cz/~janousek/smalldevs), Erişim tarihi: Kasım 2011
- [56] WAINER, G.A., MOSTERMAN, P.J., Discrete-Event Modeling and Simulation: Theory and Applications, ISBN: 9781420072341, pp. 120-200, Taylor and Francis, 2010
- [57] ÇOBANOĞLU, B., Java ile Programlama ve Veri Yapıları, ISBN: 978-9944-711-10-4, Pusula Yayıncılık, 2. Baskı, Sayfa: 11-15, 2010
- [58] REINHOLTZ, K., Java Will Be Faster Than C++, ACM SIGPLAN Notices, Volume 35, Issue 2, pp 25 – 28, 2000
- [59] NICOL, D.M., Scalability of Garbage Collection in Java-Based Discrete-Event Simulators, Conference on Computer Simulation, Cambridge England, 2003
- [60] KULJIS, J., PAUL R.J., An Appraisal of Web-Based Simulation: Whither We Wander?, Simulation Practice and Theory, 9, pp. 37-54, 2001
- [61] KILGORE, R. A., KLEINDORFER, G. B., The Future Of Java-Based Simulation, Winter Simulation Conference, pp. 1707-1712, 1998

- [62] HELSER, E.J., Design and Analysis of View Synchronization in DEVS-Suite, Master's thesis, Computer Science and Engineering Department, Arizona State University, 2009
- [63] SAAD, S.M., PERERA, T., WICKRAMARACHCHI, R., Simulation of Distributed Manufacturing Enterprises: A New Approach, Winter Simulation Conference, pp. 1167- 1173, 7-10 December, New Orleans, LA., 2003
- [64] FUJIMOTO, R. M., Parallel and Distributed Simulation Systems, John Wiley and Sons, ISBN: 0471183830, pp. 140-150, New York, 2000
- [65] HIBINO, H., FUKUDA, Y., YURA, Y., MITSUYUKI, K., KANEDA, K., Manufacturing Adapter of Distributed Simulation Systems Using HLA, Winter Simulation Conference, pp.1099-1107, 8-11 December, San Diego, CA, 2002
- [66] SCHUMANN, M., BLUEMEL, E., SCHULZE T., STRASSBURGER, S., RITTER, K.C., (1998), Using HLA for Factory Simulation, 1998 Fall Simulation Interoperability Workshop. Orlando, Florida, USA, 1998
- [67] TÜRKSEVER, M., ERDUR, R.C., Farklı Bakış Açılarında Java RMI ve CORBA'nın Karşılaştırılması, Pamukkale Üniversitesi, Mühendislik Bilimleri Dergisi, 7 (1), Sayfa: 63-70, 2001
- [68] UYGUN Ö., Yüksek Seviyeli Mimari (HLA) Temelli Dağıtık İmalat Benzetimi, Doktora Tezi, Fen Bilimleri Enstitüsü, Sakarya Üniversitesi, 2006
- [69] KARAMAN, Ö., IP Şebekelerinde Yönlendirme Protokolleri, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, İstanbul Üniversitesi, 2006
- [70] KÜÇÜKARSLAN, T., ÖZMEN, B., ÜNVERDİ, N.Ö., Ağ Omurgalarında RIP ve OSPF Uygulamaları, 2. Mühendislik ve Teknoloji Sempozyumu, Çankaya Üniversitesi, Ankara, 30 Nisan -1 Mayıs 2009
- [71] ERICSSON, M., RESENDE, M.G.C., PARDALOS, P.M., A Genetic Algorithm for the Weight Setting Problem in OSPF Routing, Journal of Combinatorial Optimization, pp. 299-333, Kluwer Academic Publishers. Manufactured in The Netherlands, 2002
- [72] YAUN, G.R., BAUER, D., BHUTADA, H.L., CAROTHERS, C.D., YUKSEL, M., KALYANARAMAN, S., Large Scale Network Simulation Techniques: Examples of TCP and OSPF Models, SIGCOMM Comput. Commun. Rev., Vol. 33, No. 3, pp. 27-41, 2003
- [73] HALSALL, F., Computer Networking and the Internet, pp. 359-373, 5th edition, Addison Wesley, ISBN: 0321263588, 2005
- [74] TANENBAUM, ANDREW S., Computer Networks, 4th edition, Pearson Education, Inc., New Jersey, 2003

- [75] EL-SAYED, H., AHMED, M., JASEEMUDDIN, M., PETRIU, D., A Framework for Performance Characterization and Enhancement of The OSPF Routing Protocol, the 9th IASTED International Conference on Internet and Multimedia Systems and Applications (EuroIMSA 2005), Grindelwald, Switzerland, Feb 21-23, 2005
- [76] STEENSTRUP, M., Routing in Communications Network. Prentice-Hall, ISBN: 0-13-185059-8, 1995
- [77] CHOWDHURY A., LUSE P., FRIEDER O., WAN P., Network Survivability Simulation of the Commercially Deployed Dynamic Routing System Protocol, IEEE Workshop on Fault-Tolerant Parallel and Distributed Systems, May 2000
- [78] APOSTOLOPOULOS, G., GUERIN, R., KAMAT, S., Implementation and Performance Measurements of QoS Routing Extensions to OSPF, in Proc. of IEEE Infocom, March 1999
- [79] DIJKSTRA, E.W, A note on two problems in connexion with graphs, <http://jmvidal.cse.sc.edu/library/dijkstra59a.pdf>, Numerische Mathematik 1: 269–271, 1950
- [80] BOZKURT, B., Bilgisayar Ağlarında Yönlendirici Algoritmaları için Simülatör, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Marmara Üniversitesi, 2001
- [81] MCQUILLAN, M.J., RICHER, I., The New Routing Algorithm for The ARPANET, IEEE Transactions on Communications, Vol. Com-28, no 5, pp. 711-719, May 1980
- [82] BUCHANAN, W., Distributed Systems and Networks, McGraw- Hill, pp. 377-391, 2000
- [83] I. E. T. Force, OSPF (RFC 1131) Internet Standards Track Protocol, <http://www.ietf.org/rfc/rfc1131.pdf>, Erişim tarihi: Kasım 2011
- [84] MOY. J., OSPF Sürüm 2, RFC, <http://www.ietf.org/rfc/rfc2328.txt>, Erişim tarihi: Kasım 2011
- [85] İnternet RFC arşivi, <http://www.faqs.org/rfcs>, Erişim tarihi: Kasım 2011
- [86] BGP Protokolü (RFC 4277), <http://tools.ietf.org/rfc/rfc4277.txt>, Erişim tarihi: Kasım 2011
- [87] BRESLAU, L., ESTRIN, D., FALL, K., FLOYD, S., HEIDEMANN, J., HELMY, A., HUANG, P., MCCANNE, S., VARADHAN, K., XU, Y., YU, H., Advances in Network Simulation. IEEE Computer, 33(5):59–67, May 2000

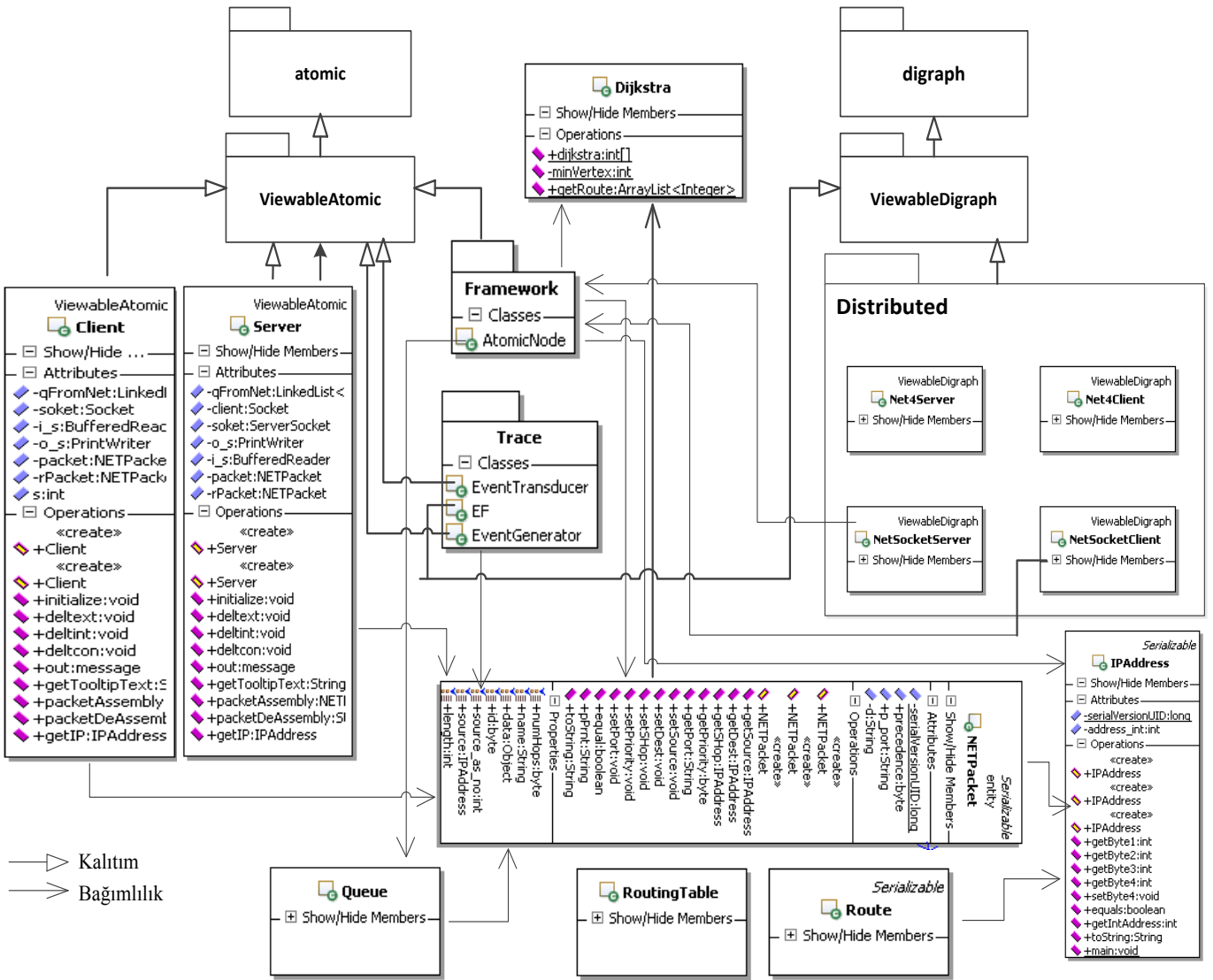
- [88] RAHMAN, M.A., PAKSTAS, A., WANG, F.Z., Network Modelling and Simulation Tools, in: Proceedings of the Eighth EPSRC Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (EPSRC PGNet 2007), Liverpool John Moores University, June 28–29, Liverpool, UK, 2007
- [89] SIAMWALLA, R., SHARMA, R., KESHAV, S., Discovering Internet Topology, <http://www.cs.cornell.edu/skeshav/papers/discovery.pdf>, Erişim tarihi: Kasım 2011
- [90] YOOK, S.H., JEONG, H., BARABASI, A.L., Modeling the Internet's Large-Scale Topology, DOI:10.1073/pnas.172501399, 2001
- [91] MEDINA, A., LAKHINA, A.I., MATTA, I., BYERS, J., BRITE: Universal Topology Generation from a User's Perspective, Technical Report, Boston University, Boston, MA, USA, 2001
- [92] ZENG, X., BAGRODIA, R., GERLA, M., GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks, Technical report, University of California, <http://www.scalable-networks.com/pdf/glomosim.pdf>, Erişim tarihi: Kasım 2011
- [93] Parallel/Distribute NS, <http://www.cc.gatech.edu/computing/compass/pdns/>, Erişim tarihi: Ekim 2011
- [94] SSFNet, [www.ssfnet.org](http://www.ssfnet.org), Erişim tarihi: Kasım 2011
- [95] J-SIM, <http://sites.google.com/site/jsimofficial>, Erişim tarihi: Kasım 2011
- [96] ÇAKIROĞLU, M., Kablosuz Algılayıcı Ağlar için Dinamik Kanal Atlamalı Güvenlik Sistemi Tasarımı, Doktora Tezi, Fen Bilimleri Enstitüsü, Sakarya Üniversitesi, 2008
- [97] ZENGİN, A., SARJOUGHIAN, H. S., EKİZ, H., Honeybee Inspired Discrete Event Network Modeling, in 16th European Simulation Symposium, Budapest, Hungary, pp. 176–182, 2004
- [98] SARJOUGHIAN, H. S., CHEN, Y., BURGER, K., A Component-based Visual Simulator for MIPS32 Processors, in Frontiers in Education, Saratoga Spring, New York, October 2008
- [99] WAXMAN, B.M., Routing of Multipoint Connections, IEEE Journal of Selected Areas in Communication 6, pp: 1617–1622, 1988
- [100] BARABASI, A. L., ALBERT, R., Emergence of Scaling in Random Networks, Science, pp. 509–512, 1999
- [101] BOEHM, B. W., Software Engineering Economics, Prentice Hall, ISBN 0-13-822122-7, May 1980



- [102] PACE, D.K., Modeling and Simulation Verification and Validation Challenges, Johns Hopkins Apl Technical Digest, 25, 2, pp. 163-172, 2004
- [103] BİLGİN, S., DEMİRÖRS, O., KARAGÖZ, A., Kavramsal Modelleme ve Simülasyon Projelerinde Önemi, Savunma Sanayii Gündemi Dergisi, Sayfa 32-37, 2010
- [104] BATES T., SMITH P., HUSTON G. CIDR Report, Web report, <http://www.cidr-report.org/as2.0>, Erişim tarihi: Kasım 2011
- [105] NICOL, D. M., Scalability of Network Simulators Revisited, In Proceedings of the 2003 Conference on Networked and Distributed Systems (CNDS), Orlando, FL, January 2003
- [106] WEINGARTNER, E., LEHN, H., WEHRLE K., A Performance Comparison of Recent Network Simulators, ICC 2009: IEEE International Conference on Communications, Dresden, Germany, 2009
- [107] ALBESEDER D., FUEGGER M., Small PC-Network Simulation – A Comprehensive Performance Case Study, Research Report, TU Wien, Institut für Technische Informatik, 2005

# EKLER

## Ek A. D-DEVSNET Sınıf Diyagramı



## ÖZGEÇMİŞ

Bülent ÇOBANOĞLU, 1974 yılı Divriği doğumludur. İlk ve ortaöğrenimini memleketi Erzincan'da tamamladı. Erzincan Fatih Endüstri Meslek Lisesi 'Elektronik' bölümünü bitirdikten sonra lisans eğitimini; Gazi Üniversitesi Teknik Eğitim Fakültesi 'Elektronik - Bilgisayar Öğretmenliği' bölümünde, yüksek lisans eğitimini ise; 'Implementation of Taylor method by the use of microcontrollers' isimli tez çalışması ile Bahçeşehir Üniversitesi Bilgisayar Mühendisliğinde tamamladı. 1997 yılında Gaziosmanpaşa Üniversitesi, Niksar Meslek Yüksekokulunda Öğretim Görevlisi olarak göreve başladı. Askerliğini 2002 yılında yedek subay olarak tamamladı. 2007 yılında ise Sakarya Üniversitesi Adapazarı Meslek Yüksekokuluna geçiş yaptı. Gaziosmanpaşa ve Sakarya üniversitelerinde birçok ulusal ve Avrupa Birliği projelerinde görev aldı. Halen Sakarya Üniversitesi Adapazarı Meslek Yüksekokulunda Öğretim Görevlisi olarak görevine devam etmektedir. Evli ve bir kız çocuğu babasıdır.