

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

**GEZGİN AĞ YÖNLENDİRME PROTOKOLLERİ İÇİN
DEVS TABANLI BENZETİM ARACI TASARIMI**

DOKTORA TEZİ

Sinan TÜNCEL

Enstitü Anabilim Dalı : ELEKTRİK-ELEKTRONİK MÜH.

Enstitü Bilim Dalı : ELEKTRONİK

Tez Danışmanı : Prof. Dr. Hüseyin EKİZ

Ekim 2010

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GEZGİN AĞ YÖNLENDİRME PROTOKOLLERİ İÇİN
DEVS TABANLI BENZETİM ARACI TASARIMI

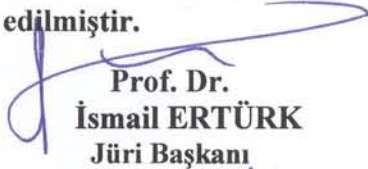
DOKTORA TEZİ

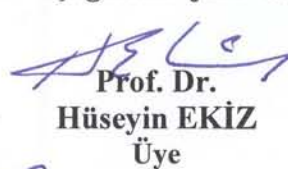
Sinan TÜNCEL

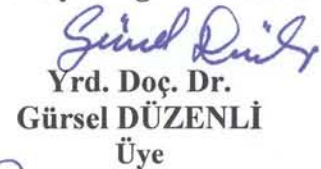
Enstitü Anabilim Dalı : ELEKTRİK ELEKTRONİK MÜH.

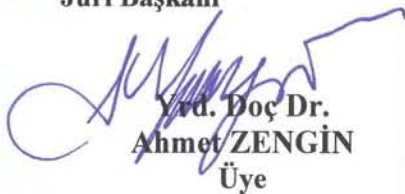
Enstitü Bilim Dalı : ELEKTRONİK

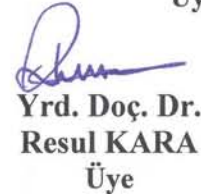
Bu tez 25/10/2010 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.


Prof. Dr.
İsmail ERTÜRK
Jüri Başkanı


Prof. Dr.
Hüseyin EKİZ
Üye


Yrd. Doç. Dr.
Gürsel DÜZENLİ
Üye


Yrd. Doç. Dr.
Ahmet ZENGİN
Üye


Yrd. Doç. Dr.
Resul KARA
Üye

TEŐEKKÜR

Eđitimim süresince olduđu gibi bu doktora alıőmasındada bana destek olan, hocam ve danıőmanım Prof. Dr. Hüseyin EKİZ'e teőekkürü bir bor bilirim.

Tez konum ile ilgili yaptıđı motivasyon ve yönlendirmelerle destek olan, tezin gerekleőtirilmesinde, baőlangıcından sonuna kadar karőılaőtıđım problemlerin özümünde, tecrübesini, fikirlerini, sabrını benden esirgemeyen Yrd. Do Dr. Ahmet ZENGİN'e sonsuz őükranlarımı sunarım.

Ayrıca bölümdeki arkadaşlarıma ve alıőmam boyunca maddi ve manevi desteklerini aldıđım, moral motivasyonlarını her zaman yanımda bulduđum aileme teőekkürü bir bor bilirim.

İÇİNDEKİLER

TEŞEKKÜR.....	ii
İÇİNDEKİLER	iii
SİMGELER VE KISALTMALAR LİSTESİ	vii
ŞEKİLLER LİSTESİ	x
TABLolar LİSTESİ.....	xiii
ÖZET.....	xiv
SUMMARY	xv
BÖLÜM 1.	
GİRİŞ	1
1.1. Ağ Benzetimleri ve Problemleri	2
1.2. Benzetimler İle İlgili Problemlere Çözüm Önerisi	5
1.3. Benzetimler ve MANET'lerle İlgili Literatürde Yapılan Çalışmalar	6
1.4. Yapılan Tez Çalışması ve Bilime Katkısı.....	11
1.5. Tez Düzeni.....	15
BÖLÜM 2.	
MODELLEME VE BENZETİM	17
2.1. Giriş	17
2.2. Modelleme ve Benzetim	18
2.3. Temel Kavramlar	18
2.4. Benzetim Süreci.....	24
2.5. Kavramsal Modelleme.....	29
2.6. Modelleme ve Benzetimin Faydaları / Olumsuzlukları.....	30
2.7. Doğrulama, Geçerleme ve Akreditasyon.....	31
2.7.1. Doğrulama ve geçerleme prensipleri	34
BÖLÜM 3.	
AYRIK OLAY TABANLI MODELLEME VE BENZETİM.....	35

3.1. Giriş	35
3.2. Sistemlerin Tanımlanması	36
3.2.1. Sistem tanımlama düzeyleri hiyerarşisi	38
3.2.3. Sistem sınıflama özeti	39
3.3. Dinamik Sistemler Formalizminde Temel Yaklaşımlar	41
3.4. Ayrık Olaylı Sistemler ve DEVS Modelleme ve Benzetim Teorisi	42
3.4.1. Ayrık olaylı modelleme yaklaşımı	43
3.4.2. Ayrık olaylı sistemlerde kullanılan terimler	45
3.4.3. Ayrık olaylı benzetim stratejileri	47
3.4.3.1. Olay zamanlama benzetim stratejisi	47
3.4.3.2. Aktivite tarama benzetim stratejisi	49
3.4.3.3. Süreç etkileşim benzetim stratejisi	50
3.4.4. Ayrık olaylı benzetim stratejileri arasındaki ilişkiler	50
3.5. Ayrık Olaylı Sistem Tanımlama DEVS Modelleme Yaklaşımı	51
3.6. Birleşik DEVS Modelleme Yaklaşımı	53
3.7. Hiyerarşik Model Tasarımı: DEVS Birleşim Çerçevesi	55

BÖLÜM 4.

HAREKETLİ AD HOC AĞLAR (MOBILE AD HOC NETWORKS – MANET) ve YÖNLENDİRME PROTOKOLLERİ	57
4.1. Giriş	57
4.2. Uygulama Alanları	58
4.3. Başarım Parametreleri	58
4.4. Hareketlilik	60
4.5. MANET Yönlendirme Protokolleri	61
4.5.1. Yol bulma yöntemleri	62
4.5.2. MANET'lerin sınıflandırılması	63
4.6. AD HOC İsteğe Bağlı Uzaklık Vektörü Yönlendirme Protokolü (AODV)	66
4.6.1. AODV yönlendirme protokolü terminolojisi	67
4.6.2. Yönlendime tablosu ve öngösterge listesi	68
4.6.3. AODV çalışma süreci ve mesaj paketlerinin hareketi	69
4.6.3.1. HELLO paketi	70
4.6.3.2. RREQ Mesajı	71
4.6.3.3. RREP mesajı	73

4.6.3.4. RRER mesajı.....	75
---------------------------	----

BÖLÜM 5.

MANET - DEVS MODELLEME VE BENZETİM ORTAMININ GELİŞTİRİLMESİ.....	77
5.1. Giriş	77
5.2. Java Programlama Dili ve DEVS-Suite ortamı	78
5.3. MANET-DEVS Modelleme Süreci	79
5.4. MANET Modelleme Yaklaşımı ve Ağı Oluşturan Bileşenlerin Tasarımı	81
5.4.1. Düğüm atomik modeli	85
5.4.2. Topografya atomik modeli	86
5.4.2.1. Topolojideki değişimlerin modellenmesi.....	87
5.4.2.2. Hareketlilik modeli.....	89
5.4.2.3. Topografyada düğümlerin görsel takibi	90
5.4.3. Ağ paketleri	90
5.4.4. Yönlendirme tabloları.....	92
5.4.5. Birleşik (Coupled) model	94
5.4.6. MANET-DEVS deneysel çerçevesi ve ağ trafik modeli	95
5.5. Topoloji Üretici.....	97
5.6. Görsel Takip	99

BÖLÜM 6.

MANET-DEVS BENZETİM DENEYLERİ VE SONUÇLARI.....	102
6.1. Giriş	102
6.2. AODV Yönlendirme Algoritması Uygulaması	102
6.2.1. AODV yönlendirme algoritması	103
6.2.2. MANET-DEVS'te uyarılama işlemi	104
6.2.2.1. Düğüm atomik modelinin davranışı ve parametreler.....	105
6.2.2.2. Ağ modeli.....	107
6.2.2.3. Senaryoların oluşturulması ve yapılan kabuller.....	109
6.2.2.4. Yönlendirme veritabanının kurulması.....	110
6.2.3. Benzetim sonuçları	111
6.2.3.1. Benzetim parametreleri	111
6.3. MANET-DEVS Doğrulama ve Geçerleme Deneyleri	116
6.3.1. Hedeflerin belirlenmesi	117

6.3.2. MANET-DEVS kavramsal model geçerlemesi.....	118
6.3.3. Model davranış geçerlemesi	118
BÖLÜM 7.	
SONUÇLAR VE DEĞERLENDİRME	122
7.1. Sonuçlar	122
7.2. Tartışma ve Öneriler	125
KAYNAKLAR	127
EKLER	134
ÖZGEÇMİŞ	141

SİMGELER VE KISALTMALAR LİSTESİ

4G	:Dördüncü nesil
ACM	:Association for Computing Machinery
AD HOC	:Geçici
AI	:Yapay Zeka (Artificial Intelligence)
AM	:Atomik model
AODV	:Ad Hoc İsteğe Bağlı Uzaklık Vektörü Yönlendirme Protokolü (Ad hoc On-demand Distance Vector Routing)
BRITE	:The Boston University Representative Internet Topology Generator
CM	: Birleşik Model (Coupled Model)
CPU	:Merkezi işlem birimi
CSV	:Virgülle Ayrılmış Değerler (Comma Separated Values)
d	:Düğümler arası uzaklık
D&G	:Doğrulama ve geçerleme
DESS	:Diferansiyel Denklemlili Sistem Tanımı (Differential Equation System Specification)
DES	:Ayrık olaylı sistemler
DEVS	:Ayrık Olaylı Sistem Tanımı (Discrete Event System Specification)
DG&A	:Doğrulama, geçerleme ve akreditasyon
DSDV	:Varış Sıralı Uzaklık Vektörü Protokolü (Destination Sequence Distance Vector)
DSR	:Değişken Kaynak Yönlendirme Protokolü (Dynamic Source Routing)
DTSS	:Ayrık Zamanlı Sistem Tanımı (Discrete Time System Specification)
EF	:DeneySEL Çerçeve (Experimental Frame)
FIFO	:İlk Giren İlk Çıkar (First-In, First-Out)
FSM	:Sonlu Durum Mekanizması (Finite State Machine)
FSR	:Balıkgözü Durum Yönlendirme Protokolü (Fisheye State Routing)
GloMoSim	:Global Mobile Information System Simulator

GUI	:Grafiksel kullanıcı arayüzü (Graphical User Interface)
h	:Yükseklik,
HLA	:Yüksek Seviyeli Yapı (High Level Architecture)
IC	:Entegre (Integrated Circuit)
ID	:Kimlik
IEEE	:Elektrik Elektronik Mühendisleri Enstitüsü (Institute of Electrical and Electronical Engineers)
IETF	:Internet Engineering Task Force
I/O	:Giriş ve Çıkış
IP	:İnternet Protokolü
JNS	:Java Network Simulator
JVM	:Java Sanal Mekanizması (Java Virtual Machine)
LAN	:Yerel alan ağı (Local Area Network)
M&S	:Modelleme ve Simülasyon
MAC	:Ortam erişim kontrolü (Media access control)
MANET	:Mobile AD HOC Network (Gezgin Ad Hoc Ağ)
n	:Düğüm sayısı
Ns-2	:Network Simulator 2
Ns-3	:Network Simulator 3
OS	:İşletim Sistemi (Operating System)
OMNET	:Objective Modular Network Test-bed in C++
QoS	:Hizmet Kalitesi (Quality of Service)
r	:Kapsama yarı çapı
RMI	:Uzak metot çağırısı (Remote Method Invocation)
RREQ	:Route Request (Yol İsteği)
RREP	:Route Replay (Yol İstek Cevabı)
RRER	:Route Error (Yol Hatası)
SPF	:En Kısa İlk Yol (Shortest Path First)
TORA	:Geçici Sıralı Yol Atama Protokolü (Temporally Ordered Routing Algorithm)
TTL	:Yaşama Zamanı (Time to Live)
UDP	:Kullanıcı Veribloğu İletişim Kuralları (User Datagram Protocol)
UML	:Birleşik Modelleme Dili (Unified Modeling Language)

VHDL	:VHSIC donanım tanımlama dili (VHSIC Hardware Description Language)
VV&A	:Verification, Validation & Accreditation
PAN	:Kişisel ağlar (personal area network)
PDNS	:Paralel/Dağıtık ağ benzeticisi
SSFnet	:Ölçeklenebilir benzetim çerçevesi (Scalable Simulation Framework)
TCP	:İletim Kontrol Protokolü (Transmission Control Protocol)
QSS	:Nitel Sistem Tanımlama (Qualitative System Specification)
w	:Genişlik,
WLAN	:Kablosuz yerel alan ağı
WRP	:Telsiz Yönlendirme Protokolü (Wireless Routing Protocol)
ZRP	:Bölge Yönlendirme Protokolü (Zone Routing Protocol)

ŞEKİLLER LİSTESİ

Şekil 2.1. Modelleme ve benzetim kavramlarının birbiriyle ilişkisi.....	19
Şekil 2.2. Model türleri	20
Şekil 2.3. Deneysel çerçevenin yapısı.....	22
Şekil 2.4. Modelleme ve benzetim süreci	26
Şekil 2.5. Benzetim kavramsal modeli.....	29
Şekil 2.5. Doğrulama ve geçerlemenin, benzetim süreci, sistem teorisi ve gerçek dünya ile olan ilişkisi.....	33
Şekil 3.1. Temel bir sistemin blok şeması.....	35
Şekil 3.2. Hiyerarşik sistem ayrışımı	37
Şekil 3.3. Giriş çıkış sistemi.....	39
Şekil 3.4. Genel sistem sınıflandırma şeması.....	40
Şekil 3.5. Sistem tanımlama formalizmleri.....	42
Şekil 3.6. Kasiyer / kuruk sisteminin davranış örneği	44
Şekil 3.7. Olay zamanlama benzetim çekirdeği	48
Şekil 3.8. Aktivite tarama benzetim stratejisi	49
Şekil 3.8. Benzetim stratejilerinin sınıflandırılması.....	51
Şekil 3.9. DEVS işleyiş mekanizması.....	52
Şekil 3.10. Ayrık olaylı sistemde giriş, durum, geçen süre ve çıkışlar	53
Şekil 3.11 Birleşik DEVS yaklaşımında bağlantılar	54
Şekil 3.12. DEVS birleşik modeli	56
Şekil 4.1. Basit bir Ad Hoc ağ görünümü	57
Şekil 4.2. MANET protokol başarımlar parametreleri.....	59
Şekil 4.3. Hedef sayısına göre yönlendirme tipleri	63
Şekil 4.4. MANET protokollerinde sınıflandırma	65
Şekil 4.5. HELLO mesaj paketi bileşenleri.....	70
Şekil 4.6. HELLO mesajları sonrası yönlendirme tabloları	70
Şekil 4.7. AODV mesaj gönderme süreci	71
Şekil 4.8. RREQ mesaj paketi içeriği.....	72

Şekil 4.9. RREQ mesaj iletimi	72
Şekil 4.10. RREQ mesaj iletimi	73
Şekil 4.11. RREQ mesaj iletimi	73
Şekil 4.12. RREP mesaj bileşenleri.....	74
Şekil 4.13. RREP mesaj iletimi.....	74
Şekil 4.14. RREP mesaj iletimi.....	75
Şekil 4.15. RRER mesaj içeriği.....	75
Şekil 4.16. RRER mesaj iletimi	76
Şekil 5.1. MANET'in modelleme ve benzetim süreci	80
Şekil 5.2. DEVS-Suite ve MANET-DEVS benzeticisi kavramsal modeli gösterimi	82
Şekil 5.3. MANET-DEVS benzetim ortamı sınıf diyagramı	83
Şekil 5.4. Atomik düğüm sınıf diyagramı.	84
Şekil 5.5. Geliştirilen düğümün kavramsal modeli.....	86
Şekil 5.6. Düğüm atomik modelinin ekran çıktısı.....	86
Şekil 5.7. Topografya atomik modelinin kavramsal modeli (a) ve ekran çıktısı (b)..	87
Şekil 5.8. Hareketli düğümlerde bağlantı kurulumu	88
Şekil 5.9. Düğümler arası bağlantıların kurulması / kaldırılması	89
Şekil 5.10. Bir paketin MANET-DEVS ortamında bileşenler arasında hareketi.....	92
Şekil 5.11. Bir yönlendirme tablosu örneği.	93
Şekil 5.12. Bir yönlendirme tablosunun DEVS-Suite altında ekran görünümü.	94
Şekil 5.13. Düğümlerin bir biri ile bağlantı kurması	94
Şekil 5.14. MANET-DEVS ortamında bir ağ modeli birleşimi.....	95
Şekil 5.15. Deneysel çerçeve ve bileşenleri	96
Şekil 5.16. Bir ağın deneysel çerçeveye bağlantısı.....	97
Şekil 5.17. DEVS-Suite BRITE topoloji üretici ekran görüntüsü.....	99
Şekil 5.18. MANET görüntüleyici için kullanıcı seçeneği	100
Şekil 5.19. MANET-Görüntüleyicide 500 X 500 lük bir alanda topoloji dağılımı .	100
Şekil 5.20. 7 düğümlü bir ağın takibi için tasarlanan ekran çıktısı	101
Şekil 6.1. AODV yönlendirme protokolü mesajlaşması.....	103
Şekil 6.2. Bir düğüme ait sadeleştirilmiş durum diyagramı.....	106
Şekil 6.3. Basit bir ağın deneysel çerçeve ile birlikte DEVS-Suite ekran görünümü	108
Şekil 6.4. Basit ağda başlangıçta yönlendirme tablolarının oluşturulması.	110
Şekil 6.5. İş çıkarma yeteneği grafiği.....	112
Şekil 6.6. Düğüm sayısı ve ortalama paket gecikmesi arasındaki ilişki	113

Şekil 6.7. Düğüm ve mesaj sayısı arasındaki ilişki	114
Şekil 6.8. Ağ trafiği ve veri paketi trafiği arasındaki ilişki.....	114
Şekil 6.9. Düğüm sayısı ve benzetim süresi arasındaki ilişki	115
Şekil 6.10. Bellek kullanımı ve düğüm sayısı arasındaki ilişki	116
Şekil 6.11. MANET-DEVS ağ katmanları yapısı	118
Şekil 6.12. Ns-2 ve MANET-DEVS topoloji ekran görüntüleri.....	120
Şekil 6.13. Ns-2 ve MANET-DEVS iş çıkarma yeteneklerinin karşılaştırma grafiği	121
Şekil A.1. Ana program akış şeması	134
Şekil A.2. Veri gönderme isteği akış şeması.....	135
Şekil A.3. Mesaj alma süreci akış şeması	136

TABLolar LİSTESİ

Tablo 1.1. Ağ benzeticilerinin karşılaştırılması	4
Tablo 2.1. Zaman tasnifi / kavramı	23
Tablo 2.2. Doğrulama, geçirme ve test sürecinde cevap verilmesi beklenen sorular	32
Tablo 3.1. Sistem Tanımlama Düzeyleri Hiyerarşisi	38
Tablo 4.1. MANET'lerin uygulama alanları	58
Tablo 4.2. Yönlendirme türlerinin karşılaştırılması	64
Tablo 6.1. Benzetim senaryoları için ihtiyaç duyulan parametre ve formüller.....	107
Tablo 6.2. Yapılan deneylerde uygulanan senaryo parametreleri	111

ÖZET

Anahtar Kelimeler: Modelleme, Benzetim, DEVS, MANET, Ad Hoc, AODV

Ad Hoc ağlar; geçici formlarda, merkezi yönetimin bulunmadığı veya geleneksel ağlardaki düzenli servis desteklerinin olmadığı gezgin düğümlerden oluşan ağlardır. Düğümler rastgele ve keyfi olarak kendilerini organize ederek hareket etmekte serbesttir. Bu yüzden ağın topolojisi hızlı ve öngörülemeyen bir şekilde değişebilir.

Ad Hoc ağlar gibi dağıtık sistemler, çeşitli teknolojiler kullanarak birbirleriyle iletişim yapan birimlerden oluşur. İletişim içerisinde bulunan sistemin uyulanabilirlik, ölçeklenebilirlik, güvenilirlik gibi bir takım niteliklere sahip olması gerekmektedir. Artan karmaşıklık ve boyut nedeniyle ortaya çıkan sorunların üstesinden gelmek amacıyla geliştirilen çeşitli yöntemler bilgisayar ağlarının ihtiyaçlarına göre kullanılmaktadır. Bu yöntemlerin tasarım ve geliştirilmesinde modelleme ve benzetim araçlarının yeri büyüktür.

Kablosuz sistemler için statik topoloji üreten benzeticiler (Qualnet, Omnet++, Ns-2, Opnet, vb.) küçük ağları çalışmak için ideal platformlarken, günümüzde üstel olarak artan ağ sistemlerini modellemede ve değişken yapıları ağ sistemlerinin başarımını test etmede yetersiz kalmaktadırlar. Ayrıca, bu benzeticilerin mimarilerinin birçoğu soyutlama ve hiyerarşiden yoksun olmaları yanında çok büyük hesaplama maliyeti oluşturmaktadırlar.

Sunulan tezde, gezgin sistemlerin karmaşıklık, ölçeklenebilirlik, uyum yeteneği gibi sorunlarının incelenmesi, tasarım alternatiflerinin araştırılması ve farklı çözüm yaklaşımlarının test edilebilmesi için Ayrık Olaylı Sistem Tanımı (DEVS) kullanılarak bir MANET modeli ve benzeticisi geliştirildi.

Geliştirilen benzeticinin üstünlüklerini ve başarımını göstermek amacıyla farklı ölçeklerden oluşan ağlar farklı trafik yükleri altında çalıştırılarak, en yaygın yönlendirme protokollerinden olan AODV'nin başarımı incelendi. Topoloji üretici özellikle büyük modelleri otomatik oluşturabilmek için tasarlandı ve model geçeleme testleri gerçekleştirildi. Gerçekleştirilen uygulamalardan, geliştirilen benzeticinin son derece paralel, esnek ve hızlı çalıştığını, değişik teknolojileri barındıran uygulamaları geliştirebilme yeteneğine sahip olduğu gözlemlendi. Ayrıca kullanılan yöntemin bu gibi sistemleri modellemede üstünlükleri gösterilmeye çalışıldı.

DESIGN AND APPLICATION OF A DEVS-BASED NETWORK SIMULATOR FOR MOBILE ROUTING PROTOCOLS

SUMMARY

Keywords: Modeling & Simulation, DEVS, MANET, Ad Hoc, AODV

Mobile ad hoc wireless networks (MANETs) are autonomous collections of mobile nodes communicating via wireless antennas. Formal verification of routing protocols for MANETs requires modeling of every part correctly to verify. The characteristics of MANETs that bring challenge to the task of modeling are node mobility, scalability and broadcast. Consequently, to cope with management of such networks in presence of ever increasing complexity, various decentralized and centralized approaches are being used to address private and public organizations demands.

To bring solutions to MANET challenges , simulators generating static topology such as ns-2, Glomosim, Omnet++, Qualnet and Opnet are ideal platforms for studying small MANET networks, but incapable of modeling and testing large-scale and dynamic structured networks. Furthermore, due to lack of hierarchy and abstraction in their structure, it is difficult to create, manage and compute large models.

In this study, a parallel and distributed network simulator is developed to bring solutions to MANET systems by using Discrete Event System Specification (DEVS) formalism. Developed simulator is called MANET-DEVS in which network model is defined with its components (e.g. nodes and links) and their hierarchical structure.

Using the DEVS hierarchical model composition concept, we develop simulation models of networks with varying topologies and scales. For example, we will use clusters to study its impact on reducing communication and increasing performance. The explicit and hidden behaviors of these networks are observed under various experimental configurations e.g., nodes and links are assigned different capacities. Topology generator is developed and validation experiments are also done.

BÖLÜM 1. GİRİŞ

İlk olarak Nikola Tesla'nın 1883'de başarılı bir şekilde kablosuz bilgi iletimini gerçekleştirmesi ile başlayan kablosuz iletişim; ilk kullanıldığı günden günümüze kadar kullanılıyor olsada, özellikle son 10 yılda daha önce hiç olmadığı kadar çok kullanılmaktadır (cep telefonu, kablosuz internet, vb.). Mobil sistemlerin avantajları; geleneksel kablolu iletişim ile kıyaslandığında, altyapı gereksinimlerinin düşük olması ve gezgin iletişim desteğidir. Cep telefonlarının iletişimdeki başarısından etkilenen araştırmacılar sabit altyapı olmadan haberleşme üzerine odaklanmışlardır. Bu hedef, cep telefonları sistemleriyle benzer başarıya sahip herhangi bir altyapı desteği olmadan geleneksel kablolu ağlar kurmak olarak ifade edilebilir. Bu yaklaşımın altında “gezgin Ad Hoc ağlar” (Mobile Ad Hoc Network - MANET) oluşturulması fikri yatmaktadır. Askeri amaçlı olarak uzun süredir çok atlamalı (multi hop) Ad Hoc ağlar kullanılsa da, MANET'ler için çok fazla ticari uygulama olmamasına rağmen nihai hedef tamamen altyapısız gezgin ağlar kurmaktır. Devletler, endüstri şirketleri ve akademik çevreler zamanlarının büyük kısmını altyapısız gezgin ağları gerçekleştirmek için harcamaktadır [1]. Geliştirilecek MANET'lerin 4G mimarisinin önemli bir parçası olması beklenmektedir [2].

Hareketlilik özelliğine sahip kablosuz ağlar iki farklı yapıya sahiptir: İlk yapıda baz istasyonu olarak adlandırılan bir tür köprü bulunmakta ve gezgin düğüm baz istasyonu çevresinde bağlantı / iletişim kurmaktadır. İkinci yapı ise, genel olarak ‘Ad Hoc’ ağlar olarak bilinen altyapısız, gezgin, sabit bir yönlendiricisi olmayan ağlardır [2]. MANET'ler “düğüm” olarak adlandırılan keyfi ve rastgele hareket edebilen platformlardan oluşur ve kablolu ağlardan farklı olarak yönlendiricinin yaptığı yönlendirme görevini de yerine getirirler. Veri, kablolu ağlardan farklı olarak düğümden düğüme yönlendirilerek iletilir [3].

MANET ve algılayıcı ağlar kaynakları sınırlı kablosuz Ad Hoc ağlardır. MANET'ler tipik olarak yüksek kapasiteli hareketliliğin ve işlemin bir arada olduğu aygıtlardan oluşur. Algılayıcı ağlar ise tipik olarak belirli coğrafik bölgelerde izleme ve algılama amaçlı olarak görev yaparlar. Bu her iki kablosuz ağ, Ad Hoc doğalarına göre karakterize edilmiştir. Her ikisinde de ağ topolojisinin sabit olmaması, enerjinin değerli olması, düğümlerin biri birine kablosuz bağlantılarla bağlanması gibi bazı karakteristikleri ortaktır [4].

Başta endüstriyel ve ticari olmak üzere birçok alanda kullanılan MANET'lerin sahip olduğu karakteristik özellikler aşağıdaki şekilde sıralanabilir [3]:

Değişken topoloji: Düğümlerin rastgele ve keyfi hareket etmesi sonucu ağ topolojisinde değişiklikler oluşur.

Sınırlı / değişken bant genişliği: Kablosuz bağlantılar kablolulara göre oldukça düşük bağlantı kapasitesine sahiptir. Buna ek olarak çoklu erişim, sönümleme (fading), gürültü ve etkileşim nedeniyle gerçekleşen bağlantı kapasitesi radyo frekans kapasitesinden düşük kalır.

Kısıtlı enerji: MANET'lerdeki düğümlerin bazıları veya tamamı enerji kaynağı olarak batarya benzeri tükenebilir enerji kaynakları kullanırlar. Düğümler için en önemli sistem tasarım kistası, enerjinin muhafaza edilmesi olmalıdır.

Sınırlı fiziksel güvenlik: Mobil ağlar, kablolu ağlara göre yüksek fiziksel tehlike ile karşı karşıyadır (dinlenme, hizmet dışı bırakma saldırıları, vb).

1.1. Ağ Benzeticileri ve Problemleri

İletişim protokolleri, bağlantı teknolojileri, trafik akışları ve yönlendirme algoritmalarının birlikte şekillendirdiği ağların tasarım süreci oldukça karmaşıktır [5]. Karmaşıklığın azaltılması ile ilgili açık bir yaklaşım, modelleme ve benzetim tekniklerinin kullanılmasıdır [5]. Donanım eksiklikleri, ekipmanların

geliştirilmesindeki zorluklar, gerçek dünyadaki karmaşık yapılandırma, güvenilir istatistiklerin toplanması, vb. sebeplerden dolayı, yeni fikirleri, ağın davranışını ve başarımını analiz etmede ağ benzetimi araştırmacılar ve mühendisler için önemli bir araçtır [6].

İnternet'in mevcut boyutuna ve karmaşıklığına erişmeden önce, küçük homojen ağlarda yönlendirme algoritmaları tasarlamak, test etmek ve modelleme / benzetim yoluyla prototip ağları incelemek kısmen mümkündür. İnternet gibi büyük ölçekli ağlar üzerinde hatalara karşı hassasiyet derecesini tespit etmek, daha sağlam yönlendirme algoritmaları tasarlamak / 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. İnternet ortamının davranışının modellerini oluşturmak amacıyla yeterince veri ve analiz yöntemi bulunsa idi, kritik hataların ve yapısal zayıflıkların tespit edilebilmesi yanında olası hatalara karşı da önlemler alınabilirdi [7].

Günümüzde çeşitli modelleme ve benzetim araçları ağ tasarımı için şirketler / akademik araştırma grupları tarafından planlanan amaca göre pratik veya eğitim amaçlı olarak kullanılmaktadır. Bu araçların işlevselliği, avantajları, süreçleri ve özellikleri açısından detaylı olarak incelenmesi gereksinimi bulunmasına rağmen hemen hemen tüm ağ benzetim araçlarının benzer yolla (bazen tamamlayıcı hedefler) çalıştığı açıktır [5].

İletişim ağlarını doğrudan modellemek için kullanılacak yazılımlar (COMNET, OPNET, Ns-2, Ns-3, GloMoSim, OMNET++, JNS, vb.) yanında donanım bileşenlerini modellemek için kullanılacak araçlar (VHDL, Verilog, SystemC, vb.) bulunmaktadır [8][9]. En yaygın kullanılan ağ benzetimcileri olan Ns-2, Ns-3, OPNET, OMNET++, vb. benzetimciler küçük boyutlu ağları çalışmak için ideal platformlardır. Bu programları büyük ölçekli sistemlerin modellenmesi ve benzetiminide kullanmanın zor olmasının yanında, değişik teknolojilerin içine katıldığı sistemleri modelleme yeteneğindedirler [10]. Ağ sistemlerinin

boyutlarının süreç içerisinde üstel olarak artması nedeni ile statik topoloji üreten bu benzetimler yetersiz kaldılar ve gelişen sistemlerin başarımını doğru bir şekilde test edemez / ölçemez duruma düştüler. Belirtilen eksikliklerin ortaya çıkmasının sebepleri olarak; basit benzetimlerin yapısal sınırlamaları ve büyük ölçekli karmaşık yapıları ağların yetersiz bir şekilde analiz edilmesi sıralanabilir. Belirtilen kısıtlamaları / yetersizlikleri bir ölçüde karşılayabilen ve binlerce düğümü modelleyebilen GloMoSim, PDNS, vb. benzetimler bulunsa da, bunların hiçbiri dinamik, gelişebilir, yeniden boyutlanabilir ve değişik trafik şartlarına uyarlanabilir (adaptif) bir ağı modelleyememektedir [10][11][12]. Ayrıca, bu tür benzetimlerin bileşenleri modüller ve hiyerarşik bir yapıda olmadığından bileşenlerin yeniden kullanımı, değişik uygulamalara uyarlanabilirliği ve hiyerarşik tasarımı zordur. Klasik benzetimlerin çalıştırıldıkları bilgisayarlarda oldukça yüksek kaynak kullanım gereksinimleri, gelişmiş ve büyük uygulamaların meydana getirilmesini zorlaştırmaktadır.

Tablo 1.1. Ağ benzetimlerinin karşılaştırılması

Karşılaştırma Ölçütleri	Ns-2	Pdns	OPNET	OMNeT++	J-sim	SSFnet	GloMoSim	DEVS-Suite
Nesne Yönelimli	orta	Orta	güçlü	orta	çok güçlü	çok güçlü	Orta	çok güçlü
Ağ model kütüphanesi	güçlü	güçlü	güçlü	güçlü	orta	Zayıf	orta	zayıf
Sonuçların analizi	orta	Orta	çok güçlü	zayıf	zayıf	Zayıf	güçlü	çok güçlü
Genişletilebilirlik	orta	Orta	güçlü	güçlü	çok güçlü	çok güçlü	çok güçlü	çok güçlü
Uzman ihtiyacı	çok güçlü	çok güçlü	zayıf	çok güçlü	zayıf	Strong	zayıf	orta
Kurulum Kolaylığı	zayıf	zayıf	güçlü	orta	çok güçlü	Strong	güçlü	çok güçlü
Dokümantasyon	orta	Orta	çok güçlü	güçlü	zayıf	Zayıf	orta	orta
Erişilebilirlik	çok güçlü	güçlü	zayıf	çok güçlü	çok güçlü	çok güçlü	zayıf	çok güçlü
Görsellik	zayıf	zayıf	güçlü	güçlü	orta	çok güçlü	çok güçlü	çok güçlü
Kullanıcı tabanı	çok güçlü	zayıf	güçlü	güçlü	orta	Zayıf	zayıf	zayıf
Ölçeklenebilirlik	zayıf	çok güçlü	orta	orta	güçlü	çok güçlü	çok güçlü	çok güçlü
Başarım	güçlü	çok güçlü	orta	orta	güçlü	çok güçlü	orta	çok güçlü
Rastgelelik	çok güçlü	çok güçlü	zayıf	güçlü	zayıf	Zayıf	zayıf	çok güçlü
Hata Modelleme	çok güçlü	çok güçlü	çok güçlü	orta	zayıf	orta	zayıf	çok güçlü
Web Erişim	yok	yok	yok	yok	güçlü	yok	yok	çok güçlü

Benzeticilerin farklı açıdan karşılaştırılması ile elde edilen özellikler, Tablo1.1'de görülmektedir. Farklı özellikler açısından tablodaki benzeticilerin incelenmesinden, mükemmel bir benzetim aracı olmadığı görülmektedir. Begg ve arkadaşları [13] tarafından yapılan çalışmada, maliyet ve süre göz ardı edildiğinde özel amaçlı hazırlanan benzeticilerin kullanıcı taleplerini karşılayabileceği vurgulanmıştır.

1.2. Benzeticiler İle İlgili Problemlere Çözüm Önerisi

MANET sistemlerinin hareketlilik, karmaşıklık, ölçeklenebilirlik, vb. problemlerinin çözülmesi amacıyla gelişmiş yönlendirme sistemlerini tasarlamak ve 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 büyük ölçekli sistemleri tasarlama yaklaşımı gerekmektedir. Yapılan çalışma, yukarıdaki gereksinimleri karşılamak üzere MANET'lere yönelik modelleme ve benzetim araçlarının tasarlanmasını ve uygulanmasını içermektedir.

Yapılan tez çalışması; bir MANET ağ sistemini oluşturan bileşenlerin DEVS yöntemi kullanılarak tanımlanmasını, tanımlanan bileşenlerin davranışlarının detaylarının belirlenmesini, ortaya çıkan modelle bir takım örnek çalışmaların ve deneylerin yapılmasını kapsamaktadır.

Bir MANET sisteminin modellenmesi işlemi; ağ bileşenlerinin tanımlanmasını, bu bileşenlerde çalışacak yazılım nesnelere ve nesnelere arasındaki etkileşimlerin tanımlanmasını, yazılım nesnelere işlem yapan düğümlere dağıtılmalarını ve ağ topolojileri ile iletişim protokollerinin tanımlanmasını içermektedir. Yönlendirme algoritmalarının test edilmesi, büyük ölçekli MANET'lerin analizi, modellenmesi, vb. ileri MANET uygulamalarını gerçekleştirmek amacıyla modellenen sistemi oluşturan parçalar ve bileşenler 'Java' programlama dili kullanılarak gerçekleştirildi. DEVS 'birleşik model' (coupled model) kavramı kullanılarak sistem bileşenleri birbirine bağlanıp, benzetim deneyleri ile model davranışı üzerinde gözlem yapmak amacıyla deneysel çerçeve [10] aracı kullanıldı. Java dilinde yazılan DEVS-Suite [14] modelleme ve benzetim ortamında, DEVS metodolojisi kullanılarak özellikle

MANET yönlendirme algoritmalarını çalışmak üzere geliştirilen ortam ‘MANET-DEVS’ olarak adlandırıldı.

Kolay tasarım ve anlaşılır arabirimiyle model tasarım ve eğitiminde etkili bir araç olan DEVS-Suite ortamı; nesneye-yönelik modellemeyi, eş zamanlı paralel çalışan benzetimleri, etkileşen benzetim nesneleri arasında uyumluluğu ve web tabanlı benzetimleri olanaklı kılmaktadır [10]. DEVS modelleme ve benzetim yaklaşımının ve Java programlama dilinin sağladığı esneklik, değişen ortama adapte olabilen zeki bileşenlerin tasarımını kolaylaştırmaktadır. DEVS-Suite’in nesneye yönelik yapısı, bir ağı oluşturan düğümlerin, linklerin, yazılım varlıklarının ve deneysel çerçevelerin modüler bir yapıda tasarımını, yeniden kullanımını ve sistemlerin sistemini oluşturmayı kolaylaştırmaktadır [15].

1.3. Benzeticiler ve MANET’lerle İlgili Literatürde Yapılan Çalışmalar

MANET’ler ile ilgili yapılan çalışmalar yukarıda detayı verilen karakteristikleri üzerine yoğunlaşmaktadır. Yönlendirme protokolleri üzerine yapılan çalışmalar verim ve enerji kullanımı üzerine doğrudan etkisi nedeniyle ön plana çıkmaktadır. Yapılan çalışmalarda benzeticilerden faydalanma önemli bir yer tutmakta ve bu nedenle benzeticiler ile ilgili çalışmalar ayrıca önem arz etmektedir. Aşağıda, MANET’lerle ilgili literatürdeki bazı çalışmalar sıralanarak özetlenmiştir:

B. P. Zeigler ve S. Mittal’in hazırladıkları raporda, ultra geniş ağların geleceği noktasından benzeticiler incelenmiş ve bu ağların geliştirilmesi için modelleme ve benzetim çevrimi ele alınmıştır. Mevcut modelleme ve benzetim araçlarının kapasitesini belirlemek ve ultra geniş ağların araştırılmasında yardımcı olması hedeflenmiştir [7].

M.A. Rahman ve arkadaşları, araştırmacıların deneylerinde doğru aracı seçmeleri için 100 adet modelleme ve benzetim aracı üzerine bir araştırma yapmış ve analiz edilen araçları karakteristiklerine göre sınıflandırmışlardır. Diğer yapılan çalışmalara göre daha geniş bir yelpazeyi ele alarak ağ keşif araçları ve topoloji üreteçlerini de

analiz ve incelemeye dâhil etmişlerdir. Araştırmacılar bu araçlardan bazılarını test etmiş ve değişik kaynaklardan elde ettikleri bilgileri sunmuşlardır. Yapılan çalışmada, gelecekte yapılacak çalışmalar için açık olan problemlerin tanımlaması da yapılmıştır [5].

C. R. Dow ve arkadaşları, IEEE / IEE Elektronik kütüphanesinden elde ettikleri 1998-2003 yılları arasındaki 1380 makaleden yola çıkarak MANET'ler hakkında yapılan çalışmaları 15 kategoriye indirerek özetlemişlerdir. Çalışmada yönlendirme ve enerji üzerine olan çalışmaların hızlı bir şekilde arttığı, IP adresleme, hata giderme ile ilgili çalışmaların yeterli dikkati çekmediği bu nedenle gelecekte bu boşlukların önemli çalışma alanları olabileceği üzerinde durulmuştur. Ayrıca MANET deneyleri için benzetim ölçütlerine değinilmiştir. [16].

Stuart Kurkowski, MANET'lerin araştırılmasında benzetimin her geçen gün daha fazla kullanılmaya başlarken yapılan benzetim çalışmalarının güvenilirliğinin azaldığı üzerine durmaktadır. Çalışmasını desteklemek amacıyla 2000-2005 arasında ACM Uluslararası Gezgin AD Hoc Ağlar ve Bilgi İşleme Sempozyumunda (International Symposium on Mobile Ad Hoc Networking and Computing-MobiHoc) yer alan yaklaşık 150 benzetim çalışmasını değerlendirmektedir. Değerlendirmede yapılan hatalar ve eksiklikler belirtilmiş ve tartışılmış, benzetim tabanlı MANET çalışmalarının güvenilirliğini artırmak için sonuçlar ortaya konmuştur [17].

L. Hogie ve P. Bouvry tarafından yapılan çalışmada, MANET'lerin modelleme ve benzetimini yapabilen benzeticilerin durumları ve ortak benzetim teknikleri üzerinde durulmuştur. Bu benzeticilerin ana karakteristikleri ve özellikleri incelenmiş, her bir benzeticinin öne çıkan yönleri belirtilerek benzeticisi seçiminde faydalanılması hedeflenmiş, hangi ihtiyaç için hangi benzeticinin kullanılacağına dair ipuçları verilmiştir. Ayrıca MANET'lerin benzetiminde öne çıkan sorunlar özetlenmiştir [18].

D. Orfanus ve arkadaşları basit topoloji kontrol algoritması ile kablosuz ağ benzeticilerinin karşılaştırılmasını yapmışlardır [19].

L. Begg ve arkadaşları tarafından benzeticiler üzerine hazırlanan kapsamlı bir teknik raporda, gelecek nesil ağ benzeticilerinde servis erişilebilirliği ve esneklik çalışmalarında ayrık olaylı benzetimin önemli bir metot olacağı belirtilmiştir. Ağ benzeticilerinin en popüler olanları, özel amaçlı benzeticilerde dâhil edilerek yapılan araştırmanın sonucu, gelecek nesil ağlar ile ilgili projelere katılan diğer çalışma gruplarının önerileri ile birlikte sunulmaktadır. En uygun benzetim araçlarının seçiminde ölçüm, servis erişilebilirliği, ağ bileşen seçimi, prototip geliştirme gibi kriterlerin değerlendirilmesinde yardımcı olacak bir çalışmadır [13].

R. Ben-El-Kezadri ve F. Kamoun çalışmalarında MANET benzeticilerin analizi, karşılaştırması ve geçerlemesi için hazırlanan bir çerçeve önerilmiştir. Sistem farklı benzeticiler ve benzetici sürümleri tarafından üretilen çıktı dosyalarını işlemeye dayanır. Ağ aktivitelerinin zamanda temsiline, ağ ölçütlerinin grafiklendirilmesine ve 802.11 standart tanımlamasına göre geçerlemesi önerilmektedir [20].

K. Shaukat ve H. Sarjoughian, hazırladıkları teknik raporda ns-2 ve DEVS modelleme yaklaşımı tasarlanan bir sıcaklık kontrol sistemi ile karşılaştırılmıştır [21].

Taekyu Kim ve arkadaşları DEVS ve Ns-2'nun birbirlerine göre avantajlarından faydalanılarak daha iyi bir benzetim çalışması yapılmak istenmiştir [22].

M. Malowidzki Ns-2, J-Sim, OPNET benzeticilerini, benzetim modu ve programlama ara yüzü açısından incelemeye tabi tutmuştur [6].

T. Antoine, kablosuz algılayıcı ağların mimarisini iki farklı protokol ile test etmiş ve yapılan benzetim çalışmasının geçerlemesini yapmıştır [23].

U. Farooq ve arkadaşları kablosuz Ad Hoc ağlarda hareketlilik ve yönlendirme işlevini Cell-DEVS kullanarak modellemiş, örnek olarak AODV protokolü başarılı bir şekilde uygulamıştır [24].

Wolfgang Kiess, MANET'lerin gerçek dünyadaki uygulamaları üzerine yapılan araştırmasında, Ad Hoc ağların davranışının benzetim çalışmaları ile gösterdiği farklılıkları sunmuştur [25].

G. Jayakumar, MANET yönlendirme protokollerinin birbirine olan üstünlük ve dezavantajlarını sıralamıştır [2].

M. Abdolhasan'nın çalışmasında yönlendirme protokolleri tabloya dayalı, isteğe bağlı ve karma olarak sınıflandırılmıştır. Her bir sınıflandırma grubundaki protokoller için temel karakteristikleri ve karmaşıklıklarının karşılaştırılması yapılarak tablolar ile özetlenen bir gözden geçirme çalışmasıdır [26].

C. Liu, çalışmasında MANET'lerde kullanılan protokollerin genel özelliklerini anlatarak tek yönlü ve çok yönlü olarak sınıflandırdığı protokoller kendi aralarında karşılaştırılmaya tabi tutmuştur [27].

P. Kuosmanen, Ad Hoc yönlendirme protokolleri üzerine yapılan sınıflandırma çalışmasında, farklı protokollerin uygunluk ve zıtlıklarının iyileştirilmesi hedeflenmiştir [28].

A.V. Biradar, yapılan sınıflandırma çalışmasında diğerlerinden farklı olarak Ad Hoc ağlarda kullanılan yönlendirme protokollerini genetiğine göre sınıflandırılmıştır [29].

J. Haerri, düğüm seviyesindeki davranışlara göre (topoloji kontrol, konum yönetimi, geniş yayın vb.) protokollerin sınıflandırılması üzerine kapsamlı bir çalışma yapmıştır. MANET yönlendirme protokollerinin geliştirilmesi ve tasarımı için benzerlikleri ortaya çıkarmaya yönelik bir çalışmadır [30].

L. Junhai, MANET'lerde çok yönlü yayın yapan protokolleri incelemiş ve MANET karakteristiklerinin iyileştirilmesine yönelik önerilerde bulunmuştur [31].

N. Meghanathan, Ad Hoc ağlar üzerine yapılan çalışmalara destek olması amacıyla yek yönü yönlendirme protokollerini incelemiştir [32].

G. Jayakumar, AODV ve DSR yönlendirme protokollerinin Ns-2 ile başarımlarını karşılaştırılmasını gerçekleştirmiştir [33].

T. Camp, Ad Hoc ağların benzetiminde kullanılan hareketlilik modellerini, başarımların ölçümlerinde kullanılacak hareketlilik modeli seçiminde yardımcı olması amacıyla incelemiştir. Hareketlilik modelinin değişmesiyle Ad Hoc protokol başarımlarında büyük değişiklik olduğu gösterilmiştir [34].

S. Sesay, yönlendirme, ortam erişim, çoklu yayım, hizmet kalitesi, TCP başarımları, enerji, güvenlik, Bluetooth gibi ağ kavramları üzerine tartışmalarda bulunmaktadır [35].

N. Sivakumar çeşitli başarımların kriterlerine göre Ad Hoc yönlendirme protokollerinin değerlendirilmesini yapmıştır [36].

A. Rasheed AODV protokolünün farklı sürümlerinin uygulamalarını inceleyerek her bir uygulamanın avantaj ve dezavantajlarını tartışmış AODV-UU'nun kurulum / yapılandırılması Ns-2 benzetim ortamında detaylı olarak anlatmıştır [37].

N.S. Yadav, isteğe bağlı Ad Hoc protokollerin Ns-2 ile başarımlarını karşılaştırılmasını yapmıştır [38].

C. Hongsong, çalışmasında AODV yönlendirme protokolüne karşı DoS ve Black hole ataklarını analiz etmiş ve çözüm önerilerinde bulunmuştur [39].

C. E. Perkins, MANET'lerde kullanılan protokollerin Ns-2 benzetim ortamında farklı ağ yükü, hareketlilik ve ağ büyüklüğü ile detaylı bir başarımların ölçüm çalışması yapmış ve başarımların artırılabilmesi için önerilerde bulunmuştur [40].

A. Boukerche, Ns-2 benzetim ortamında Ad Hoc ağlarda kullanılan protokollerin farklı yük ve büyüklükteki ağlarda benzetimini gerçekleştirmiştir. Paket türlerinin başarımlar üzerindeki etkisine değinmiştir [41].

G. Campos 4 farklı Ad Hoc protokolünün 8 farklı ağda 3 farklı hareketlilik türünde Ns-2 benzetim ortamında videophone uygulamasının başarımlar ölçüm çalışmasını yapmıştır [42].

I. Broustis ve arkadaşları Ns-2 ve QualNet benzetimcilerini kullanarak MANET'lerde, seçilmiş protokoller kapsamlı başarımlar ölçümüne tabi tutmuşlardır [43].

1.4. Yapılan Tez Çalışması ve Bilime Katkısı

Tez sürecinde yapılan çalışmalar ve yapılan çalışmaların bilime katkısı aşağıdaki şekilde özetlenebilir:

- Farklı MANET yönlendirme algoritmalarının incelenmesine olanak tanıyan bir ortam oluşturulması amacıyla örnek bir ağ modelinin DEVS (Discrete Event System Specification) kullanılarak modellenmesi ve benzetimi işlemleri gerçekleştirildi.
- Geliştirilen MANET benzetici üzerinde yaygın bir yönlendirme algoritması olan AODV modellenerek başarımlar analizi yapıldı.
- Geliştirilen benzetiminin büyük ölçekli ağlarda kullanılabilirliğini göstermek amacıyla, çeşitli boyutlarda ağ modelleri oluşturularak başarımlar incelendi.
- Otomatik olarak topoloji üretmek amacıyla BRITE topoloji jeneratörü sisteme entegre edildi.
- Geliştirilen benzetiminin güvenilirliğini tesis etmek üzere Ns-2 ile geçirme testlerine tabi tutuldu.

Yapılan çalışmada, ‘MANET-DEVS’ modelleme ve benzetim ortamının çalışmasını ve yönlendirme algoritmalarının bu ortamda gerçekleşmesini göstermek amacıyla bir takım örnek uygulamalar gerçekleştirilmiştir. Uygulamada, en yaygın kullanılan isteğe bağlı yönlendirme protokollerinden biri olan “Ad Hoc Distance Vector (AODV) yönlendirme algoritmasının MANET-DEVS içinde modellenmesi ve benzetim işlemleri gerçekleştirildi. Kablosuz ağlarda uzaklık vektörü sınıfı yönlendirme algoritmalarının en yaygın kullanılanlardan biri olan AODV algoritması uygulanırken, MANET-DEVS ortamının klasik algoritmaları modelleme yeteneğinde olduğu gösterilmeye çalışıldı.

Gerçekleştirilen son uygulamada; DEVS-Suite ortamının yüksek başarımlı yeteneği kullanılarak, MANET-DEVS ortamının kapasitesini belirlemek ve DEVS yaklaşımının paralel / dağıtık uygulamalardaki gücünü göstermek amacıyla farklı ölçekte ağlar incelendi ve topoloji analizi yapıldı. Büyük ölçekli ağları incelemede başlıca hedeflerimizden birisi; İnternet türü ağların analizi ve yönetimidir.

Geliştirilen MANET-DEVS benzetim ortamının kavramsal model ve davranış geçerlemesi yapıldı. Ns-2 ve MANET-DEVS benzetim araçları benzer senaryolarda test edilerek iş çıkarma yetenekleri karşılaştırıldı.

MANET-DEVS modelleme ve benzetim ortamının bilime katkıları aşağıdaki şekilde özetlenebilir;

- Sistem teorisi tabanlı, modüler, hiyerarşik yapılı, yüksek başarımlı, ölçeklenebilir ve yeniden kullanılabilir yazılım mimarisine sahip bir benzetici tasarımı gerçekleştirilmiştir.
- MANET’ler gibi dağıtık, karmaşık, dinamik sistemlerin ihtiyacı olan, sistem tasarımının uygun bir şekilde onaylanmasını ve geçerlenmesini kolaylaştıran, benzetici tasarımı formalizasyonuna sahip bir benzetici geliştirilmiştir.

- Sistem problemlerinin ortaya çıkarılmasını kolaylaştıran, model geliştirme zamanı düşük bir benzetici tasarımı gerçekleştirilerek, aynı modelin birçok kez kullanıldığı (ağdaki düğüm, yönlendirici, vb) uygulamalarda büyük kolaylık sağlamakta, oluşturulan bileşenlerin ve varlıkların, modüler bir yapıda tasarımını, yeniden kullanımını mümkün kılmaktadır.
- DEVS yaklaşımının matematiksel altyapısı, durum değişkenlerinin değişimlerine ve sabit-parçalı bir formatta grafiklerin üretilmesine imkân sağlamaktadır. Böylece geliştirilen MANET-DEVS benzeticisi ile yapılan benzetim çalışmalarını gözlemlemek, sistemin karakteristikleri hakkında bilgi sağlayacak sonuçları üretmek ve elde edilen sonuçları değerlendirmek / yorumlamak kolaylaşmıştır.
- Geliştirilen MANET-DEVS benzeticisi, izleme ve analiz modülünden (Tracking Environment, Time View) yararlanarak, oluşturulan model üzerinde gelişmiş test ve deneylere imkân sunmakta, yapılan benzetim çalışmasının sonuçlarını hem 'log' (günlük) dosyalarında, hem de 'csv' dosyalarında saklayarak, benzetim sonuçlarının kolay bir şekilde değerlendirilmesini mümkün kılmaktadır.
- Tamamen nesneye yönelik bir programlama dili kodlanmış olması, MANET-DEVS benzeticisini platformdan bağımsız kolay kurulabilir bir altyapıya sahip kılmaktadır.
- Java programlama dili kullanılarak gerçekleştirilen uygulamaların, web tarayıcılar altında rahatlıkla çalışabilmesi, MANET-DEVS benzeticisinin İnternet üzerinden yayınlanabilmesini / kullanılmasını kolaylaştırmakta, İnternet kullanılarak uygulamaların paylaşılması ve çağımızın eğitim teknolojileri (web tabanlı, uzaktan eğitim, vb.) için elverişli bir altyapı oluşturmaktadır.
- Geliştirilen MANET-DEVS ortamına ilave edilecek değişik bileşenlerle, geliştirilen benzetici, sadece ağ sistemlerini değil, farklı paralel ve dağıtık

sistemlerin modelleme ve benzetiminin yapıldığı bir benzetim ortamına dönüştürülebilir (örnek: kablosuz algılayıcı ağlar).

- DEVS-Suite ortamından destek alan MANET-DEVS ortamı, sistemin ve modelin yapısal ve davranışsal durumunu, zaman tabanlı eğrilerle, mesaj animasyonlarıyla, bileşen yapılarının adım adım takip edilerek kurulan modellerin, daha iyi izlenmesini / analizini mümkün kılan bir benzetim ortamı sağlamaktadır.
- MANET-DEVS ortamında, entegre edilen BRITE topoloji üretici ile, kod yazmaksızın büyük ölçekli topolojiler üretilebilmektedir, böylece büyük ölçekli ağların kolayca kurulup analizine olanak tanımaktadır.
- DEVS / HLA teknolojisi ve Java RMI teknolojileri kullanılarak, MANET-DEVS ortamında oluşturulan modeller kolayca birden fazla işlemci üzerine bölünebilecek ve böylece oldukça büyük ölçekte ağ modellerinin benzetim çalışması yapılabilecektir. Oluşturulan ağların, uygun arabirimlerle başka ağlara bağlanarak modüler ve hiyerarşik bir ağ sisteminin tasarlanmasına imkân vermektedir.

Özet olarak, gerçekleştirilen MANET-DEVS benzeticisi, dağıtık ve karmaşık dinamiklere sahip MANET'lerin karakteristikleri üzerine, ölçeklenebilir, adaptif, farklı topolojilere sahip, sağlıklı ve kolay benzetim çalışmaları yapılmasına imkân sağlayarak, gezgin ağ uygulamalarının modellenmesi / tasarımı için örnek bir çerçeve olmuş, DEVS metodolojisinin geniş bir uygulama alanına uygulanabilirliğini desteklemiştir. Geliştirilen benzeticide, Java dilinin kullanılması, uygun işletim sistemi seçme, altyapı programı yükleme / güncelleme vb. işlemleri ortadan kaldırmaktadır.

1.5. 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 organizasyonu hakkında bilgi verilmektedir.

Bölüm 2’de modelleme ve benzetimin günümüzdeki önemi hakkında bilgi verilmiş, modelleme ve benzetimin tanımı ve kullanılan temel kavramlar ve bilgiler verilmiş, iyi bir modelleme benzetim çalışmaları için gerekli süreç detaylandırılarak adımlar halinde anlatılmıştır. Modelleme ve benzetimde deney ve deneysel çerçevenin önemine değinilmiş, kavramsal modelleme özetlenmiş, modelin sistem karşılığına kabul edilir bir uygunlukta olup olmadığının denetlenmesi işlemi olan doğrulama ve geçерleme konuları üzerinde durulmuştur. Son olarak modelleme ve benzetimin avantajları ve dezavantajları sıralanmıştır.

Bölüm 3’te sistemin tanımlanması, sistemlerin çözümlenmesindeki kavramlar, sistem tanımlanmasında takip edilmesi gereken adımlar ve sistemlerin sınıflanması özetlenmektedir. Sistem sınıflamalarından biri olan dinamik sistemlerin formalize edilmesinde kullanılan yaklaşımlar tanımlanmakta ve Ayrık Olaylı Sistem Tanımında (Discrete Event System Specification – DEVS) kullanılan temel kavramlar ele alınıp, ayrık olaylı benzetim stratejileri ve birbirleri ile olan ilişkilerine değinilmektedir. DEVS yaklaşımı detaylandırılıp, atomik, birleşik DEVS kavramları tanımlanmakta ve hiyerarşik model tasarımından bahsedilmektedir.

Bölüm 4’de, tezde sunulan benzetici çalışmasında kullanılan MANET tanımı, uygulama alanları, MANET’lerin başarımlar parametreleri anlatılmakta ve MANET ağları farklı kılan başlıca özellik olan “hareketlilik” konusu ele alınmaktadır. Ayrıca MANET’lerde yönlendirme protokolleri ve bu protokollerin sınıflandırılmasında ele alınmaktadır. Son olarak MANET’lerde yaygın olarak kullanılan yönlendirme protokollerine kısaca değinilerek tez çalışmasında uygulaması gerçekleştirilen AODV (Ad Hoc On-Demand Distance Vector Routing – AODV) yönlendirme

protokolü detaylandırılarak örnek bir senaryo üzerinde protokolün temel mesaj trafiği izah edilmektedir.

Bölüm 5’de bir MANET ağı oluşturan bileşenlerin DEVS yaklaşımı ile tanımlanması ve ağı oluşturan bileşenlerin davranışları açıklanmakta, MANET-DEVS platformunun modelleme süreci ve bu platform için oluşturulan nesnelere anlatılmaktadır. MANET’lerde analiz ve ölçeklenebilirlik testlerinin yapılması için bu nesnelere Java programlama dili yardımıyla DEVS-Suite platformunda kod ortamına aktarılmıştır. Bu teorik bilgiler DEVS-Suite platformundan alınan ekran görüntüleri ile desteklenmektedir.

Bölüm 6’da MANET-DEVS modelleme ve benzetim ortamında AODV yönlendirme protokolünün modellenmesi ve benzetimi sunulmakta, DEVS modelleme ve benzetim yaklaşımı kullanılarak gezgin Ad Hoc ağ sistemleri ile bu ağlarda kullanılan yönlendirme protokolünün gerçek bir ağ benzeticisi gibi çalışıp çalışmadığının testi yapılmaktadır. Oluşturulan deneysel çerçeve yoluyla elde edilen çıkışlar grafikler ile sunulmakta ve yorumlanmaktadır.

Bölüm 7’de yapılan çalışmalardan elde edilen sonuçlar genel hatlarıyla özetlenmekte, tez çalışmasının bilime katkıları tartışılmakta ve ileride bu çalışmanın devamı olarak yapılabilecek çalışmalara ışık tutabilecek önerilerde bulunmaktadır.

BÖLÜM 2. MODELLEME VE BENZETİM

2.1. Giriş

Modelleme ve benzetim modern hayatta gittikçe artan bir rol oynamakta, yeni sistem ve ürünlerin değerlendirmesini, çalışmasını, etkili tasarımını ve nasıl çalıştıklarını anlamamıza katkı sağlamaktadır. Modelleme ve benzetim sonuçları pek çok alanda karar verme ve çözüm için hayati bilgiler sunmaktadır [44].

Modelleme ve benzetim alanı, oldukça karmaşıktır. Modelleme ve benzetimle ilgili her bilim dalı kendisine özgü modeller, yöntemler ve bu modelleri çalıştırmak için gerekli araçlarla birlikte gelişmiştir veya gelişmektedir [45].

Modelleme ve benzetim uygulamaları yaygın bir şekilde kullanılmakla birlikte, belirli bir bilim dalına özgü olmayan model tanımlama, basitleştirme, onaylama (validation), benzetim, araştırma (expolaration) vb. kavramlar bulunmaktadır. Bu ifadeler herkes tarafından kabul edilmiş olmalıdır. Ancak, genel olarak kullanışlı bir biçimde ayrıştırılabilen (isolated) ve soyutlanabilen kavramlar söylenmelidir [45].

Farklı bilim dallarında modelleme ve benzetim alanında yapılan çalışmalarda kullanılan birçok kavramın (model tanımlama, basitleştirme, geçerleme, benzetim vb.) birbirine benzediği görülmüştür. Bu durum, modelleme ve benzetim olgularının ayrı bir disiplin olarak ele alınmasına sebep olmuştur [46].

Aşağıda modelleme ve benzetimin tanımının ardından benzetim süreci ve temel kavramlar açıklanmaktadır.

2.2. Modelleme ve Benzetim

Modelleme, bir model oluşturma sürecinin adıdır [47]. 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.

Benzetim ise bir sisteme ait neden - sonuç ilişkilerinin bilgisayar ortamına aktarılarak sistemin davranışlarının bilgisayarda izlenmesini sağlayan bir modelleme tekniğidir. [48].

Aşağıda modelleme ve benzetim sürecinde kullanılan temel kavramlar ve modelleme benzetim sürecinin aşamaları anlatılmaktadır.

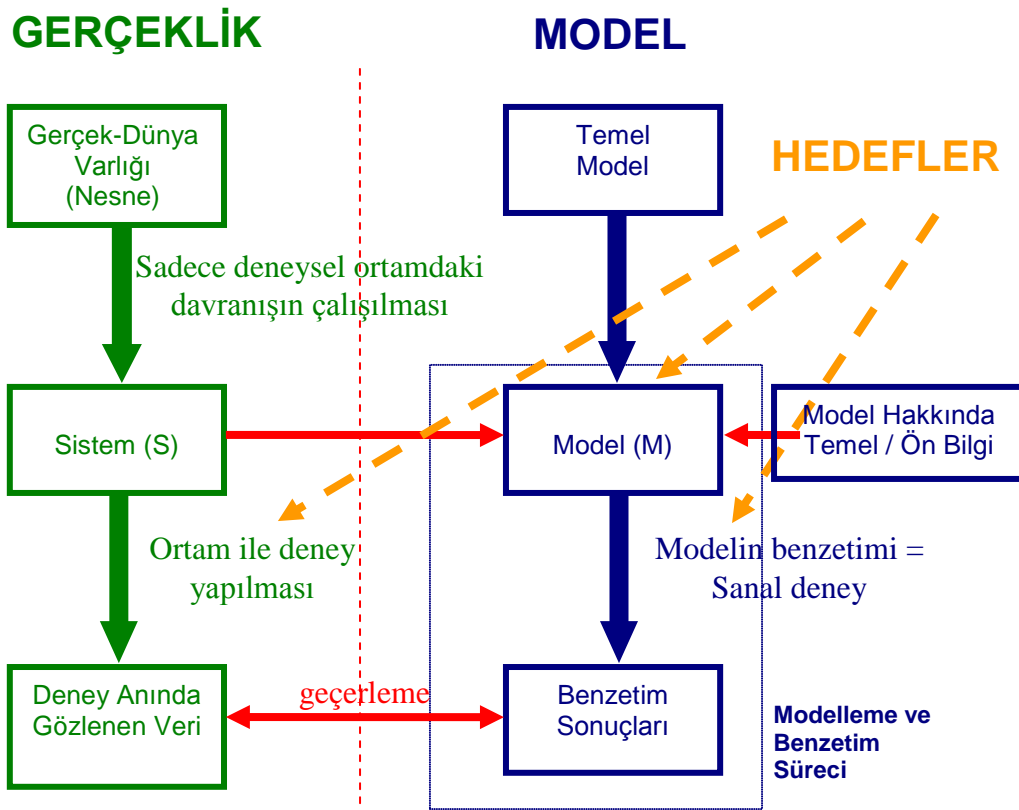
2.3. Temel Kavramlar

Şekil 2.1’de modelleme ve benzetimde kullanılan temel kavramların birbiriyle ve gerçek dünya ile olan ilişkisi gösterilmektedir. Bu kavramlar şu şekilde tanımlanabilir;

Nesne: modelleme ve benzetim sürecinde çalıştığı ortama göre değişken bir davranış sergileyen gerçek dünyadaki bir varlıktır.

Temel Model: Varsayımsal / farazidir. Geçerli tüm durumlar ve yüzeysel görünüşler için nesnenin özelliklerinin soyut gösterimidir. Pratikte bir nesnenin bütün özellikleri / davranışları / gösterimlerini bir modelde oluşturmak ise farazidir [49].

Sistem: Girdileri, durumları, davranışları ve çıktıları olan bir birim olarak tanımlanır [49]. Sadece yapı ve davranış açısından dikkate alınarak belirli şartlar altında gerçek dünyada tanımlanmış nesnedir [45].



Şekil 2.1. Modelleme ve benzetim kavramlarının birbiriyle ilişkisi

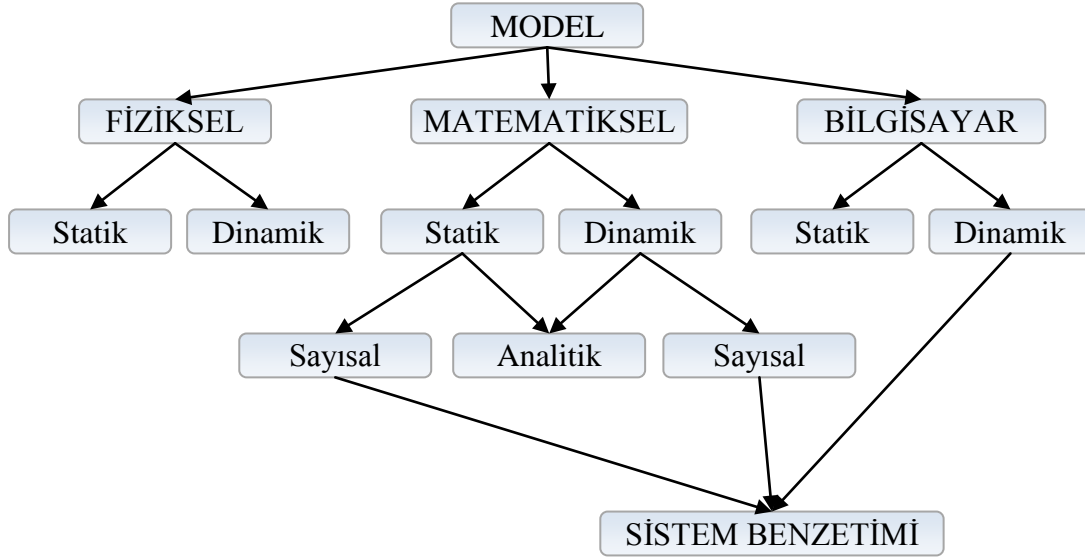
Model: Model bir sistemin çalışmasını ve yapısını temsil eder. Model temsil ettiği sisteme benzerdir, fakat onun basitleştirilmiş halidir. Model sisteme yakın ve dikkat çeken özelliklerini içermekle beraber deney gerçekleştirecek ve anlaşılabilir kadar basit olmalıdır. İyi model gerçeklik ve basitlik arasında dengeyi sağlar [47].

Model türleri Şekil 2.2' de görülmektedir [50].

Fiziksel modeller: Fiziksel modeller dinamik ve statik olmak üzere iki türe ayrılır. Statik fiziksel model sistemin küçük ölçekli bir modelidir ve zamanla değişmez (bir binanın maketi, deniz yerine küçük bir su tankı, vb.).

Dinamik fiziksel modeller zamanla değişen veya zamanın bir fonksiyonudur. Rüzgâr tünelineki bir uçak modeli, modele bütünleşmiş üreteç yardımıyla üflenen hava

yoluyla farklı sürat ve basınç değerleri ölçümleri yapılabilir. Burada rüzgâr hızı zamanla değişir ve dinamik modele bir örnektir [50].



Şekil 2.2. Model türleri

Matematiksel model: Pek çok sistem matematiksel denklemlere dönüştürülebilir. Bu denklemler sistemin matematiksel modeli olarak adlandırılır. İlk çağlardan beri bilim adamları matematik yardımı ve gözlem yoluyla doğanın gizemini çözmeye çalışmışlardır. Kepler'in kuralları güneş sisteminin dinamik bir gösterimidir. Akışkan mekaniğindeki denklemler akışkanların modelinin dinamik bir temsilidir [50].

Statik model sistem dengede olduğu zaman değişkenlerin ilişkilerini tanımlar. Sistemin matematiksel modelinin dengedeki hali Statik Matematiksel Model olarak adlandırılır [50].

Bilgisayar modeli: Bilgisayarların ortaya çıkışı ile modelleme ve benzetim kavramı tamamen değişmiştir. Sürekli matematiksel modeller bilgisayarların sayısal metotları kullanılarak çözümlenebilir. Bu şekilde problem çözümlenmesine bilgisayarla modelleme denir. Burada akla gelen soru, çözümün benzetim çözümünden farkı nedir? Benzetimin gerçek anlamı sistem veya olgunun çalışma davranışını benzetmek veya kopyalamaktır. Gerçekte benzetim matematiksel hesaplamayı, bilgisayar grafiklerini

ve hatta ayırık modellemeyi içeren bir bilgisayar modelidir. Fakat bilgisayara dayalı model bundan farklıdır. Matematik kadar grafiklerden yardım alınır ve gerçek senaryolar yardımı ile benzetim yapılır [50].

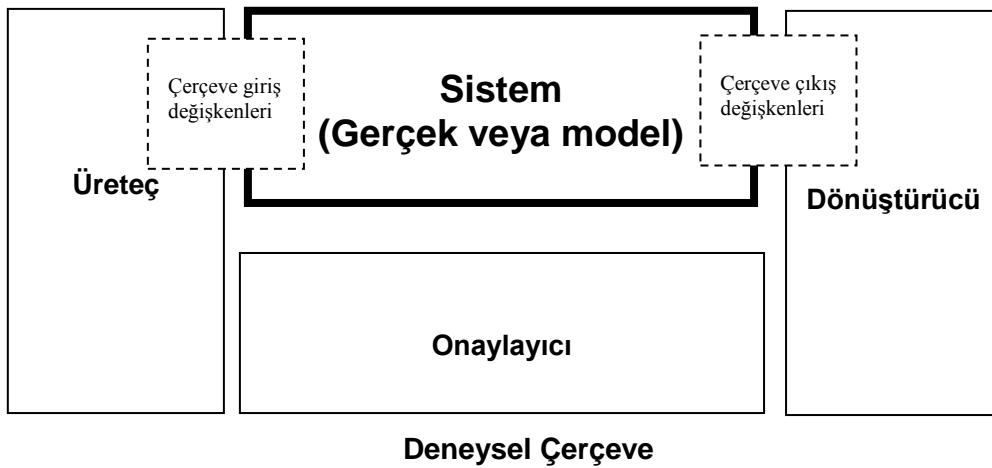
Yazılım mühendisliğinde model, modelleme diliyle formüle edilir. Örnek olarak UML (Unified Modelling Language - "Birleşik Modelleme Dili") verilebilir. UML bir programlama / yazılım geliştirme dili olmaktan ziyade, sistemlerin nasıl modellenebileceğini belirleyen ve açıklayan yöntemlerin bir araya toplanmış halidir. Daha çok yazılım geliştiriciler tarafından kullanılıyor olsa da UML ile yapılan modellemeler sadece yazılım projelerinde kullanılmak zorunda değildir. Çeşitli diyagramlar yardımı ile sistem geçişlerini açıklamaya yardımcı olur [51].

Bileşik (Lumped) Model: Belirli bir deneysel çerçevenin kapsamında eksiksiz bir sistem tanımlamasını verir. "Tam tanım" ("accurate description") kesin bir şekilde tanımlamaya gerek duyar. Genellikle sistem yapısının kesin özellikleri ve / veya davranışları belirli bir doğrulukta modeli yansıtmalıdır [49].

Doğrulama ve Geçerleme: 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 [52].

Deney: Sistemi fiziksel olarak test etme işlemidir. Sistemin çalışmasını giriş ve parametrelerini değiştirerek etkileyebilir. Deney ortamı ayrı bir sistem olarak görülebilir ve dolayısıyla birleşik bir model yoluyla ayrıca modellenebilir. Deney gözlem işlemini içerir, gözlem ise ölçümleri elde etmeyi sağlar [49].

Deneysel Çerçeve: Herhangi bir sistemle gerçek dünyada çalıştığımızda, deneysel çerçeve (Experimental Frame - EF) sistem ve sisteme karşılık düşen modellerin çalıştırılacağı deneysel şartları / ortamları tarif eder. Diğer bir ifadeyle deneysel çerçeve, gerçek bir sistem veya bir benzetim aracılığıyla model üzerinde deneyler yapan kişinin / modelleyicinin hedeflerini yansıtır.



Şekil 2.3. Deneysel çerçevenin yapısı

Deneysel çerçeve sistemin / modelin giriş ve çıkış uçlarına bağlanan ‘Çerçeve Giriş Değişkenleri’ ve ‘Çerçeve Çıkış Değişkenlerinden’ oluşur (Şekil 2.3). Üreteç (generatör) deney esnasında sistemi / modeli uyararak için, dönüştürücü (transducer) ise sistemden gelen sonuçlardan mantıklı bir yorum yapmak için sisteme / modele uygulanacak dönüşümleri tanımlar.

Çıkış kavramı ile fiziksel sistem ve gözlemci tarafından ölçülen model içi durumlar biçimindeki yapay değerler kastedilmektedir. Üreteç ve dönüştürücü, giriş / çıkış değişkenleri ile birlikte deneysel çerçeve içindeki üreteç girişleri ile dönüştürücü çıkışlarını karşılaştıran bir onaylayıcıdan (acceptor) oluşur. Onaylayıcı, sistemin (gerçek veya model) deneysel çerçevesinin deney yapanın hedefleriyle uygun olup olmadığını belirler [48].

Benzetim: Belirli bir modelleme yaklaşımında (DEVS, Petri Net, Diferansiyel Eşitlikler, Bond Grafikleri, vb) tanımlanan bir modelin, benzetim sonuçlarını (dinamik giriş / çıkış davranışı) üretir. Benzetim, bir sistemin davranışı ile ilgili sanal bir deney yapma işlemi olarak görülebilir. Benzetim hem sembolik hem de nümerik teknikleri kullanabilir. Modellemenin hedefi bilgiyi sunan sistemi anlaşılır ve yeniden kullanılabilir bir şekilde mantıksal olarak tarif etmek iken, kullanılan tekniğin önemli olmadığı benzetimin amacı; olabildiğince hızlı olmak ve modelin işlevlerini doğru bir şekilde yansıtmaktır. Sembolik teknikler tek bir çözümden daha

çok çözümler sınıfının oluşmasını sağlamaları nedeniyle, çoğu kez sayısal tekniklere göre daha çok tercih edilmektedir. Ayrıca sembolik optimizasyonlar doğaları sayesinde nümerik çözümlerden daha büyük bir etkiye sahiptirler. Model ve sistem arasında eşbiçimli (homomorphic) bir bağıntının olması Sistem – Deney / Model – Sanal Deney düzeneği için çok önemlidir: gerçek bir sistemin modelini geliştirmek ve geliştirilen modelin davranışının benzetimini yapmak, deneysel sonuçları gözlemlemeyi ve sistematik bir şekilde düzenlemeyi takiben yapılan gerçek bir deneyle aynı sonucu verir [48].

Zaman: Gerçek zamanın akışını modelleyen olayları sıraya koyup düzenlemek için bir zaman eksenini kullanılır. Eğer böyle bir zaman ekseninin yorumu gerçeklikten uzak kalırsa, ondan mantıksal zaman olarak söz edilir.

Gerçek dünyada, gerçek zamanda meydana gelen olaylar dikkate alındığında, gerçek bir saat tarafından gösterilen bir zaman değişkenine başvurulur. Bu sebeple mantıksal zaman, bir şekilde modelin içine yerleştirilmiş bir saat tarafından ölçülürken, metrik zaman veya duvar saati zamanı olarak adlandırılan fiziksel zaman, fiziksel bir saat tarafından ölçülür. Aynı zamanda, izafiyet teorisinin gösterdiği gibi, farklı yerlerdeki gözlemciler tarafından algılanan zaman farklı olabilir. Bu ayrıntıya dayanarak, zaman hem lokal hem de global olabilir. Birincisi, sadece bir sistemin bir bileşeninin içerisinde geçerli iken; ikincisi, bütün sistem içinde geçerlidir. Bu nedenle, en azından iki tane zamanı sınıflandırma boyutu bulunur: mantıksal/fiziksel eksen boyunca ve diğeri lokal /global eksen boyunca. Sonuç olarak, zaman kavramı, Tablo 2.1’de gösterilen dört türden herhangi birine uygun getirilerek yorumlanabilir [45].

Tablo 2.1. Zaman tasnifi / kavramı

		Mantıksal / Fiziksel	
		Mantıksal Zaman	Fiziksel Zaman
Global / Yerel	Global Zaman	Global, mantıksal: Tüm bileşenler benzer kuramsal zaman ekseninde çalışır	Global, fiziksel: Tüm bileşenler benzer sistem saatinde çalışır
	Yerel Zaman	Yerel, mantıksal: Bileşenler kendi zaman ekseninde çalışır	Mantıksal, fiziksel: Bileşenler kendi sistem saatinde

Geleneksel olarak, modelleme ve benzetimin çoğunlukla ilk türde (global, yerel) olduğu düşünülür. Modellenen bir sistemin bütün bileşenlerinin aynı zaman çerçevesi referansına sahip olduğu varsayılır ve zaman bir soyut nicelik olarak dikkate alınır. Ancak, bir model, bir ağdaki bilgisayarlar arasında dağıtık olabilen ve aynı zamanda gerçek dünya ile etkileşim yapabilen bir benzetici içinde çalıştırıldığında, bu kurguyu sürdürmek zordur. Zaman esasları arasındaki senkronizasyonun, ikisinin arasındaki bir benzerliği sürdürmeyi gerektirdiği önemlidir. Mesela, dağıtık bir benzetim protokolü, özel benzetici düğümleri tarafından sürdürülen yerel, mantıksal zamanlarını senkronize eder. Başka bir senkronizasyon örneği gerçek zamanlı, insan kapalı devreli (human-in-the-loop) benzetim tabanlı eğitimde ortaya çıkar. Burada benzetici, bir pilotun fiziksel olarak algıladığı zaman esası ve benzetimi yapılan uçak modelinin mantıksal zamanı arasında bir uyum oluşturmak için bir fiziksel zaman esası (bilgisayar sisteminin saati vb.) kullanır [45].

2.4. Benzetim Süreci

Benzetim projelerinin yönetimi söz konusu olduğu zaman dikkatlice hazırlanmış sistematik bir plan ve uygulaması gereklidir. Benzetim konularında “40-20-40” kuralı uygulanır [53].

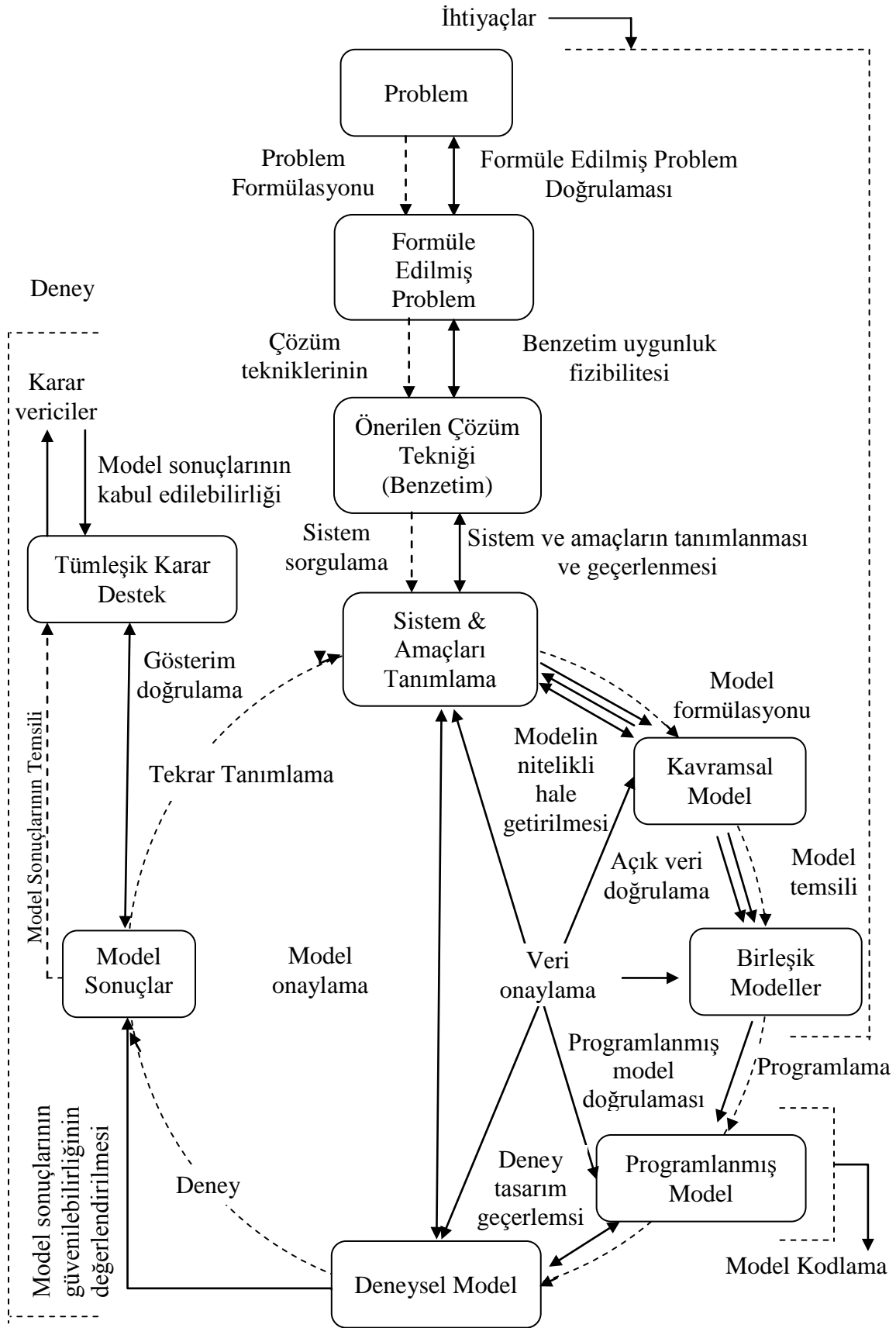
- %40 problem tanımlanması, ihtiyaçların tanımlanması, proje planlaması, sistem tanımlanması, kavramsal modelin formülasyonu, başlangıç deneylerinin tasarımı ve giriş verilerinin hazırlanması,
- %20 modelin çevrilmesi,
- %40 model geçirme, doğrulama, deneye son halinin verilmesi, analiz, yorumlama, uygulama ve dokümantasyon içindir.

Bu aşamaların tamamen birbirinden bağımsız olduğunu düşünmek mümkün değildir. Örneğin ilk aşamada, gereksinimlerin temininde programlama da göz önüne alınmalıdır. Model geliştiricileri kolay ve doğru deney kadar gerçekleştirecek benzetim programı için de gayret etmelidirler. Şekil 2.4 benzetim modelinin içerdiği görevleri detaylı bir şekilde göstermektedir. Bu görevlerin pek çoğu kodlama aşamasından önce yer almaktadır ve model gözden geçirmelerine bağlı olarak farklı aşamalarda tekrar etmektedir.

Benzetim süreci araçları projelerin kodlama aşamasına destek için kullanılmaktadır. Bununla beraber bu araçların hiçbirisi özellikle benzetimi desteklemek için geliştirilmemiştir. [53]

Benzetim modeli geliştirme sürecinde gerekli adımlar şu şekilde sıralanabilir [47].

1. Problemin tanımlanması (hedefin tayini)
2. Problemin formüle edilmesi
3. Gerçek sistem verilerinin toplanması ve işlenmesi
4. Modelin geliştirilmesi ve formüle edilmesi
5. Modelin geçerlenmesi
6. Gelecekte kullanım için dokümantasyon hazırlanması
7. Uygun deneysel tasarımın seçilmesi
8. Çalıştırma için deneysel durumların kurulması
9. Benzetimin çalışmasının gerçekleştirilmesi
10. Sonuçların sunulması ve yorumlanması
11. Başka hareket tarzlarının tavsiye edilmesi



Şekil 2.4. Modelleme ve benzetim süreci

Benzetim çalışmasındaki bu mantıksal sıralamaya rağmen benzetimin hedefleri gerçekleşmeden pek çok alt evreler gerekebilir. Bu adımların tamamı her zaman mümkün veya gerekli olmayabileceği gibi ek adımlarda gerekebilir. Aşağıda üç aşamada bu adımlar açıklanmaktadır.

i. Benzetim modelinin geliştirilmesi

Benzetim modelini oluşturan bileşenler, sistem varlıkları, giriş değişkenleri, başarımlar ölçümü ve fonksiyonel ilişkilerdir. Hemen hemen tüm benzetim yazılımları her bir bileşen için model yapısını destekler. Modelin tartışılması benzetim çalışmasının en önemli parçasıdır.

Benzetim modelinin geliştirilmesi süreci aşağıdaki adımları içermektedir (Şekil 2.4):

1. Adım: Problemin tanımlanması: Mevcut sistem ile ilgili problemler sıralanır. Önerilen sistem için gereksinimler belirlenir.
2. Adım: Problemin formüle edilmesi: Çalışmanın hedefleri ve problemler ile karşılaştırılacak farklı sistem konfigürasyonları için başarımlar ölçütleri belirlenir. Sistem başarımı hakkında hipotezler ortaya konur ve çalışma-zaman çerçevesine karar verilir. Benzetimin son kullanıcıları belirlenir.
3. Adım: Gerçek sistem verilerinin toplanması: 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 üreticileri ve uygun dağılım türleri içermektedir. Deneysel dağılımlar, standart dağılımlar veya mevcut sistem verisi uygun olmadığında kullanılırlar.
4. Adım: Modelin formüle edilmesi ve oluşturulması: Sistemin şematik ve ağ diyagramları geliştirilir (varlıkların sistemdeki akışı gibi). Kavramsal model

benzetim yazılımının kabul edebileceği forma dönüştürülür. Modelin beklenen şekilde çalıştığı doğrulanır.

5. Adım: Model geçerlenmesi: Gerçek sistemin başarımı ile modelin başarımı bilinen durumlar altında karşılaştırılır. İstatistiksel çıkarım testleri gerçekleştirilir ve sistem incelenir.

6. Adım: Dökümantasyon: Gelecek kullanımlar için model hakkında yardımcı bir belge hazırlanır. Hedefler, varsayımlar, giriş değişenleri ve ayrıntıları kayda geçirilir.

ii. Benzetim deneyinin tasarlanması

Benzetim deneyi ciddi testler içerir. Bu testler gözlem ve başarımlar ölçümlerindeki değişikliklerin sebeplerini tanımlamak için giriş değişkenlerinin değiştirilmesini içerebilir. Benzetim çalışmasında deneyin sayısı model boyunca sorulan soruların sayısından fazladır veya eşittir. Takip eden adımlar deney tasarımını anlatmaktadır.

7. Adım: Uygun deneysel tasarımı seçilmesi: Uygun etkiye sahip birkaç giriş değişkeni için başarımlar ölçümü yapılır.

8. Adım: Çalıştırma için deneysel koşullar kurulması: Çalışma süresi, uygun başlangıç koşulları ve gerekli ise hazırlık süresi seçilir. Çıkış veri büyüklüğüne bakarak bağımsız deney sayısı belirlenir (her bir çalışma benzer durum fakat farklı rastgele sayı akışı). Örnek sayısının başarımlar ölçümlerine güvenilebilecek çoklukta olması önemlidir.

9. Adım: Benzetimi çalışmasının gerçekleştirilmesi: 7. ve 8. adımlara uygun olarak tekrar çalıştırılır.

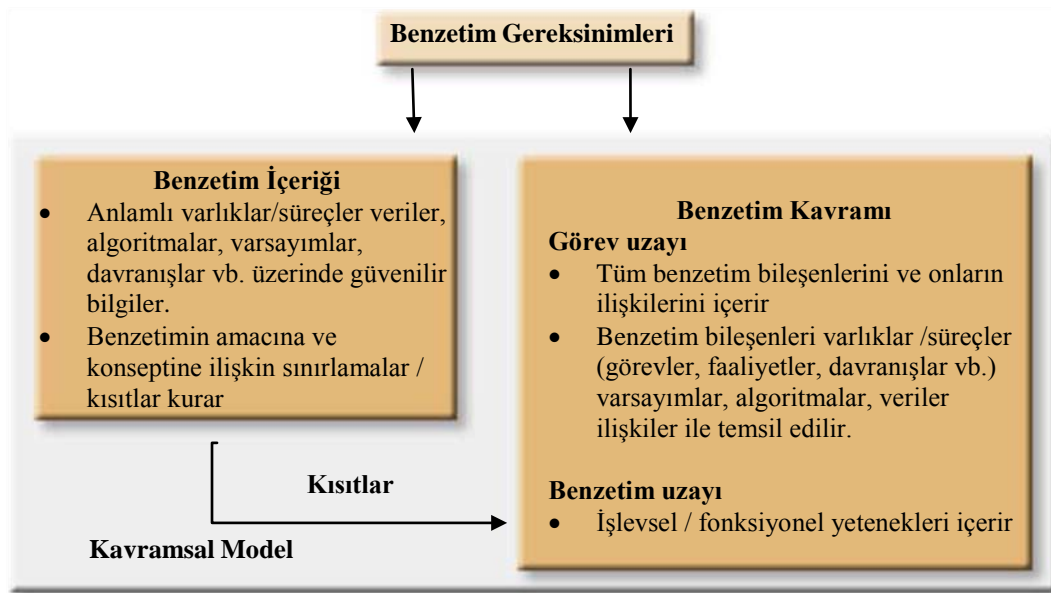
iii. Benzetimin başarımlarının analizi

10. Adım: Sonuçların yorumlanması: İlgili her bir durum için istenen başarımlar ölçümlerine göre sayısal hesaplama yapılır. Sistem başarımları ile ilgili varsayımlar test edilir. Çıkış verileri ile ilgili grafiksel gösterimler oluşturulur.

11. Adım: Doğruluğu artırmak, tereddütleri azaltmak, daha hassas analizler için daha farklı deneyler yapmayı içerir.

2.5. Kavramsal Modelleme

Kavramsal modellemenin ilk aşaması benzetim geliştiriciler, tasarımcılar (sistem analizcileri, sistem mühendisleri, yazılım tasarımcıları, kod geliştiriciler, test elemanları vb) ve uygulama personeli arasındaki geniş kapsamlı iletişim olmalıdır. Bir benzetim sistemine ilişkin kavramsal model Şekil 2.5’de gösterildiği gibi üç ana bileşenden meydana gelir [54].



Şekil 2.5. Benzetim kavramsal modeli.

Birinci bileşen olan benzetim içeriği, benzetim alanı ile ilgili bilgi ve bilgi kaynaklarını ifade eder. Modellenmesi öngörülen varlıklar ve süreçler ile, ilgili kaynaklar / indeksler ve benzeri bilgiler yer almaktadır.

Benzetim geliřtiricilerin tüm uygulama hakkındaki kavram ve kullanıcı isterlerini karşılayacak bir sistemin nasıl gerçekleştirileceğinin öngörüsünü içermektedir. Örnek olarak benzetimi gerçekleştirilecek bir askeri hava operasyonunda fizik ve aerodinamik kuralları benzetim içeriğİ, momentumun korunumu benzetim kavramıdır.

Benzetim bileşenleri, benzetimde yer alacak her varlığın, işlemin veya bunların gruplarının modellerine ilişkin bilgileri içermektedir. Benzetim bileşenleri, bileşenin muhtemel durumları, davranışları, başarımı, öznitelikleri parametreleri gibi özelliklerini tanımlayan veriler ve ilişkilerini içerir. Varlıklar gibi işlemler ve süreçler ve süreci etkileyen çevresel etkiler benzetim bileşeni olabilir [54].

2.6. Modelleme ve Benzetimin Faydaları / Olumsuzlukları

Uygulayıcılara göre modelleme benzetim ve analiz en sık kullanılan araştırma tekniğidir. Modelleme ve benzetimin faydalarını aşağıdaki gibi sıralayabiliriz [47].

- Çalışılan sistemin matematiksel bir modeli geliştirilerek sistemin daha anlaşılır olmasını sağlar ve çok uzun periyotlarda sistemin çalışması gözlemlenebilir
- Fizibilite için sistem hakkındaki hipotezlerin testine imkân sağlar,
- Detaylardaki karmaşık olguları gözlemlmek için zamanı sıkıştırmaya veya genişletmeye imkân sağlar,
- Sistemin modelini değiştirerek sistemin uygulanmasındaki etkiler gözlemlenebilir. Gerçek sistem ile yapılacak uygulamaların önemli risklerini gerçek sistemi bozmadan azaltır,
- Yetersiz bilginin olduğu yeni veya bilinmeyen durumlar hakkında deneme imkânı verir,
- Başarım ölçümlerinin en hassas olduğu sürücü değişkenleri ve birbirleri ile olan ilişkilerini saptar,
- Varlıkların veya bilginin akışındaki tıkanıklıkların belirlenmesi (materyal insan vb.),
- Sistem konfigürasyonu için çoklu başarım ölçümleri kullanılabilir,

- Problem çözümüne sistematik yaklaşım sağlar ve
- Kısa sürede iyi ve sağlam bir sistem geliştirilmesini sağlar.

Modelleme ve benzetimin olumsuz yönleri;

- Benzetim zaman alan karmaşık alıştırmalar olabilir,
- Açık olmayan nesnelere içerebilir,
- Analitik çözüm mümkün iken benzetim kullanmak,
- Hatalı model tasarlamak,
- Benzetim modelinin fazla karmaşık veya basit olması,
- Hatalı varsayımlar yapılması,
- Kayda geçirilmemiş varsayımlar. Benzetim modelleme ve analiz çalışmalarının her safhasında dokümantasyon çok önemlidir,
- Girişte yanlış olasılık dağılımı kullanmak,
- Bir dağılımı (stochastic) kendi ortalamasıyla (deterministic) değiştirmek,
- Başarımın yanlış ölçülmesi,
- Benzetim programındaki hatalar,
- Benzetim çıkış analizinde bağımsız farz edilen standart istatistik formülleri kullanmak,
- Çıkış verisinde başlangıçtaki önyargı,
- Konfigürasyon için tek benzetim çalıştırmak,
- Eksik çalışma takvimi veya bütçe planlaması ve
- Benzetim çalışması ile ilgisi olan personel arasında eksik iletişimidir.

2.7. Doğrulama, Geçerleme ve Akreditasyon

Benzetim modeli herhangi bir hedefe (tasarım, analiz, kontrol, optimizasyon vb.) ulaşmak için bir araçtır. Bu nedenle temel / ön şart, modelleme ve benzetim (araçları) aracılığıyla yapılan çıkarımların / elde edilen sonuçların rahatlıkla kabul edilebilmesinin güvence altına alınmasıdır. Bu güvenin kurulabilmesi için iki ayrı çalışma gerçekleştirilebilir: doğrulama ve geçerleme (D&G / Verification and Validation) [48].

Doğrulama bilgisayar ortamına dönüştürülmüş model tasarımının istenen seviyede olup olmadığının kontrol işlemidir. Geçerleme ise eldeki amaçlar için modelin yeterli doğrulukta olup olmadığının kontrol işlemidir. Anahtar kavram yeterli doğruluktur. Hiçbir modelin % 100 doğru olması mümkün değildir ve bu durumun geçerli sebepleri vardır. Bununla beraber D&G'nin hedefi modelin yeterince doğruluğunu garanti altına almaktır [55].

Benzetim süreci boyunca doğrulama, geçerleme ve model akreditasyonunun (DG&A) cevap aradığı sorular Tablo 2.2'de özetlenmektedir [56].

Tablo 2.2. Doğrulama, geçerleme ve test sürecinde cevap verilmesi beklenen sorular

Doğrulama	Çalıştırılabilir modelin programlama kodu kavramsal modele doğru bir şekilde uygulanabildi mi?	Kavramsal model, modelin planlanan kullanımına cevap veriyor mu?	Çalıştırılabilir model ihtiyaç olduğunda ve istenildiği formatta sonuç üretiyor mu?
Geçerleme	Kavramsal model benzetim ortamında doğru temsil ediliyor mu?	Üretilen sonuçlar benzetim ortamının davranışlarına ne kadar yakın?	Hangi aralıkta girişler model sonuçları için kullanışlı ve güvenilir?
Akreditasyon	Modelin kapasitesi ve planlanan uygulamaların gereksinimleri uygun mu?	Doğrulama ve geçerlemenin sonuçları eğer planlanan uygulama için kullanılırsa doğru sonuçları verir mi?	Planlanan uygulama için eksik model uygulanırsa sonuçları ne olur?

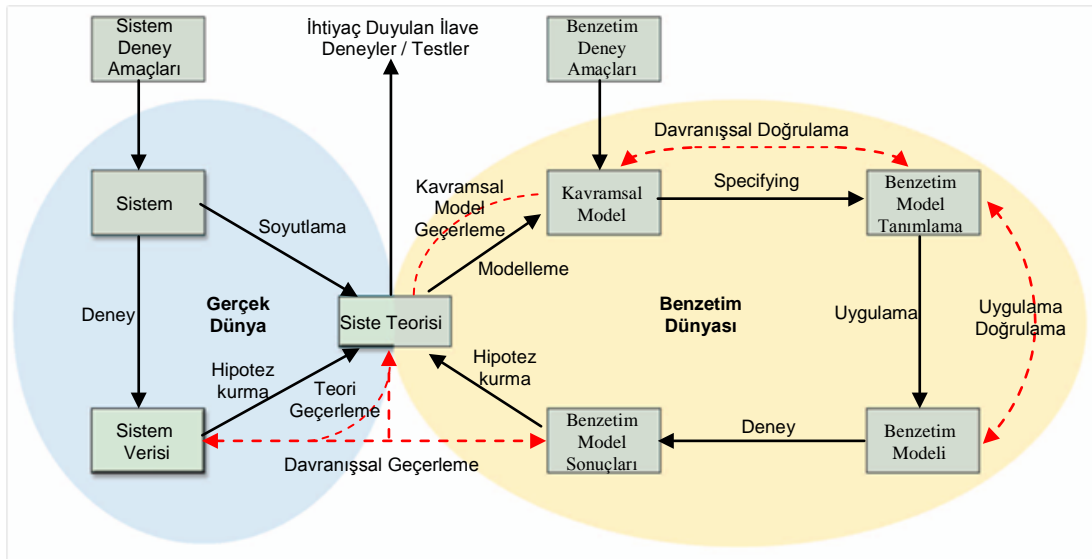
Deneysel hedefler ile geçerlemenin gereksinimleri arasında kuvvetli bir bağ vardır. Geçerleme her zaman hedefler / istekler / planlar ile ilişkilidir. Şekil 2.5'de kesik çizgiler karşılaştırma, değerlendirme veya ölçümleri göstermektedir [44].

Çeşitli geçerleme türleri bulunmaktadır: kavramsal model geçerleme, yapısal geçerleme ve davranışsal geçerleme.

Kavramsal model geçерleme; kavramsal model altında yatan teori ve varsayımların sistem teorileri ile uyumlu olduğunu ve sistemin model gösteriminin benzetim modelinin amacı için makul olduğunu belirleme olarak tanımlanır.

Yapısal geçерleme; kavramsal modeli belli bir bilgisayar sistemi üzerinde programlama ve uygulama için yazılım tasarımının ve özelliğinin yeterli olduğunun belirlenmesi olarak tanımlanır.

Davranışsal geçerleme; model çıkış davranışının uygulanabilirlik boyutunda modelin hedeflenen amacı için yeterli doğruluğa sahip olup olmadığını belirlemek olarak tanımlanır [57].



Şekil 2.5. Doğrulama ve geçerlemenin, benzetim süreci, sistem teorisi ve gerçek dünya ile olan ilişkisi

Doğrulama ve geçerleme aktivitelerinin bir özeti Şekil 2.5'de görülmektedir [44]. Burada, bir sistem ve o sistemin modeli arasında üretilen davranış açısından uygunluk sadece sınırlı bir deneysel çerçeve içinde değerlendirilecektir. Diğer taraftan, ölçümlerin ve benzetim sonuçlarının büyük oranda örtüşmesi benzetime olan güveni artırmasına rağmen modelin geçerliliğini ispatlamaz. Bu sebepten dolayı, tahrif (falsification) kavramı ortaya atılmıştır. Tahrif, tam olarak bir modeli çürütmek ve bozmak işlemidir. Sonuç olarak, bilgiyi karşılıklı olarak değiştiren modeller

kullanıldığında, model ile deneysel çerçeve uyumlu olmalı ve bir model kendi deneysel çerçevesiyle eş zamanlı olarak tasarlanmalıdır. Bu gereksinim bir model tasarım dilinin geliştirilme gereksinimini ortaya çıkarmıştır [49].

2.7.1. Doğrulama ve geçeleme prensipleri

Doğrulama ve geçelemeyle ait prensipler aşağıda özetlenmiştir [52].

1. D&G testleri benzetim sürecinin tamamını kapsar ve süreç boyunca çalışmaya eşlik eder.
2. Benzetim modelinin sonuçlarının kesin doğru veya yanlış olduğunu ortaya koymaz; model sistemin özetidir. Mükemmel bir model asla beklenemez.
3. Benzetim modeli çalışmanın hedeflerine uygun olacak şekilde inşa edilir ve bu hedeflere göre güvenilirliğine karar verilir; modelleme ve benzetim sürecinin sistem ve hedeflerin tanımlanması aşamasında açıkça ve anlaşılır bir şekilde ortaya konur. Çalışmanın hedeflerinin doğru belirlenmesi benzetim çalışmasının başarısı için çok önemlidir.
4. Benzetim modeli DG&A, çalışmaların kötü etkilenmesini önlemeyi içerir.
5. DG&A sezgisel yaratıcılık istediğinden güçtür.
6. Benzetim modelinin güvenilirliği önceden belirlenip test edilen durumlar için iddia edilebilir.
7. Benzetim modelinin tamamının testi mümkün değildir; modelin tam testi mümkün olan tüm girişler için test edilmesi gereklidir.
8. Benzetim modelinin doğrulanması, geçeleme ve testi planlanmış ve yazılmış olmalıdır.
9. Hatalar modelleme sürecinde mümkün olduğunca erken ortaya çıkarılmalıdır.
10. Çoklu cevap problemi sağlıklı bir şekilde tespit edilmeli ve çözümlenmelidir.
11. Her bir alt modelin testi, tüm modelin güvenilirliğini göstermez.
12. Benzetim modelinin doğrulanması, benzetimin sonuçlarının güvenilirliği ve kabul edilirliliği manasına gelmez.
13. Formüle edilmiş problemin doğruluğunun test edilmesi benzetimin kabul edilebilirliğini ve güvenilirliğini büyük ölçüde etkiler.

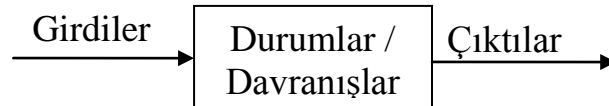
BÖLÜM 3. AYRIK OLAY TABANLI MODELLEME VE BENZETİM

3.1. Giriş

İlk olarak 1960'larda geliştirilen matematiksel sistem teorisi, dinamik sistemlerin çözümü için kesin matematiksel formüllerle desteklenmiştir. İki temel bakış açısı vardır:

1. Sistem tanımlama seviyeleri; Sistemlerin davranışını tarif etmede kullanılan düzeyler ve çalıştıran mekanizmalardır.
2. Sistemleri tanımlama formalizmleri; Modelleri inşa etmek için kullanılan sürekli ve ayrık modelleme türleridir.

Modelleme ve benzetim teorisinde yer alan metotların pek çoğu sistem teorisine dayalı olarak geliştirilmiştir. Temel olarak sistem, girdileri, durumları, davranışları ve çıktılara sahip bir birim olarak tanımlanmaktadır [45]. Şekil 3.1'de temel bir sistemin yapısı, blok şeması görülmektedir.



Şekil 3.1. Temel bir sistemin blok şeması

3.2. Sistemlerin Tanımlanması

Sistem teorisi, 'sistemin yapısı' (sistemin dâhili oluşumu) ile 'sistemin davranışını' (dış göstergesi) birbirinden ayırır. Bir sistemin giriş / çıkış davranışı, gerçek bir sistem veya modelden toplanan veri kayıtlarından oluşur. Sistemin içyapısı, hem durum çıkışlarını hem de durumdan duruma geçiş mekanizmalarını içerir. Sistem yapısının bilinmesi, sistemin davranışını (analiz, benzetim) ortaya çıkarmaya imkân sağlar. Genellikle, davranıştan yapıyı çıkarma işlemi kolay olmamakla beraber, gözlenen davranışın geçerli bir temsilini bulmak, modelleme ve benzetim sürecinin önemli aşamalarından birisidir [48] [45].

Sistemler çoğunlukla birçok alt sistemin bir araya gelmesi ve etkileşmesiyle oluşan karmaşık yapılardır. Bu yapılar sistem davranışı, sistem yapısı, sistem çözümlemesi ve sistem sentezi kavramlarıyla çözümlenmektedir. Sistem ile ilgili kavramlar aşağıdaki gibi açıklanabilir.

Sistem davranışı: Bir sistemin girdileri ile çıktılarının ilişkilendirilmesinde ortaya koyduğu karakteristik olarak tanımlanmaktadır.

Sistem yapısı: Sistemin sahip olduğu durumlar ve durumlar arasındaki geçişler ile bu geçişlerin sistem çıktılarını üretme mekanizmasıdır.

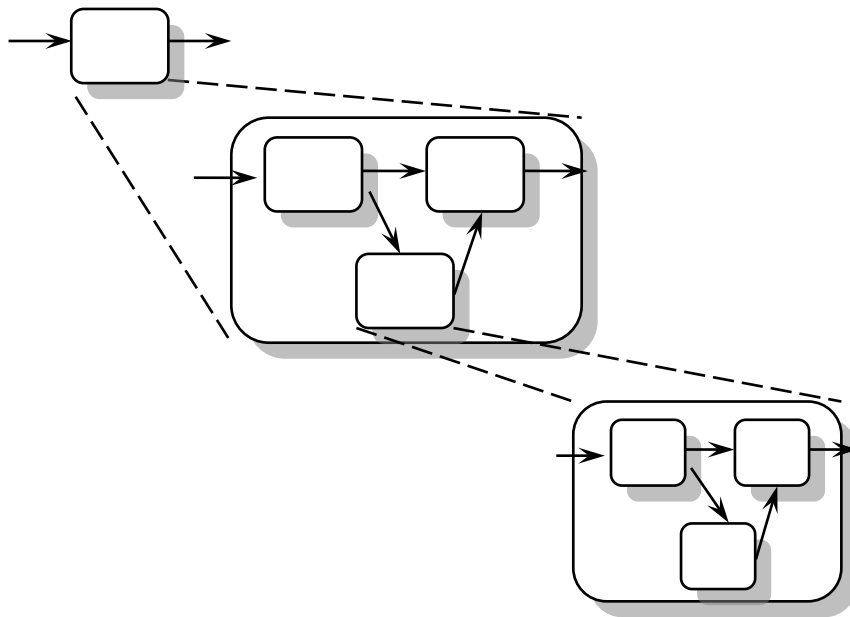
Sistem çözümlemesi: Sistemin ayrıştırılarak alt parçalarının belirlenmesidir.

Sistem sentezi: Alt sistemlerin (bileşen sistemler) bir araya getirilerek daha büyük sistemlerin oluşturulmasıdır [46].

Modelleyiciler gözlemlere dayanarak bir sistemin daha karmaşık modellerini tasarlayabilirler. Hiyerarşi, karmaşık modellerin tasarlanmasında, modellerin sınıflandırılmasını ve yönetimini kolaylaştıran önemli bir kavramdır. Hiyerarşi içindeki her bir yapı sistem hakkında daha detaylı bilgi sağlayarak bir öncekini detaylandırır. Sistemin yapısı ve davranışı ile ilgili bazı sorunların hiyerarşideki en

alt seviyelerinde daha iyi cevap bulması nedeniyle, daha üst seviye bir modelden daha az detaylı bir modele doğru ters yönde hareket etmek mümkündür. Özellikle en alt seviyede tanımlı eğriler biçimindeki açık davranış çoğu durumda gereklidir.

Önemli bir yapı kavramı, sistemin sistem parçalarına ayrılması olarak adlandırılan ‘ayrışmadır’ (decomposition) (Şekil 3.2). İkinci bir kavram; sistemlerinin bir araya gelip birleşerek daha büyük bir sistemi meydana getirmesi olarak ifade edilebilecek ‘birleşimdir’ (composition). Sistemlerin bileşiminin yapısı ve davranışı temel sistem teorisi terimleriyle ifade edilebildiği için, sistemler teorisi bileşim altında kapalıdır. Daha önceden tasarlanan bileşenlerden daha büyük sistemleri oluşturma ve oluşturmaya devam etme işlemi ‘hiyerarşik yapıyı’ ortaya çıkarır. Bileşim altında kapalılık, böyle bir birleşimin iyi tanımlı yapı ve davranışa sahip bir sistem (bileşke – ‘resultant’ olarak adlandırılır) ile sonuçlanmasını garanti eder. Modüler sistemler, çevre ile tüm etkileşimlerin yapıldığı giriş ve çıkış portlarına sahiptir. Modüler sistemler, giriş portları çıkış portlarına bağlanarak ‘birleşik’ (coupled) hale getirilebilirler ve daha büyük bir sistemi oluşturmak için bir araya getirilen sistemler hiyerarşik bir yapıya sahip olabilirler [48].



Şekil 3.2. Hiyerarşik sistem ayrışımı

3.2.1. Sistem tanımlama düzeyleri hiyerarşisi

1970’lerde George Jiri Klir tarafından geliştirilen kuramlara paralel olarak modelleme ve benzetim teorisi, ‘benzer hiyerarşi’ ile formüle edilmiştir. Bu çerçevede dinamik sistemlerin genel kavramları ve sistem tanımlamasında kullanılacak yollar tanımlanmıştır. Bu yollar Tablo 3.1’de açıklanmaktadır.

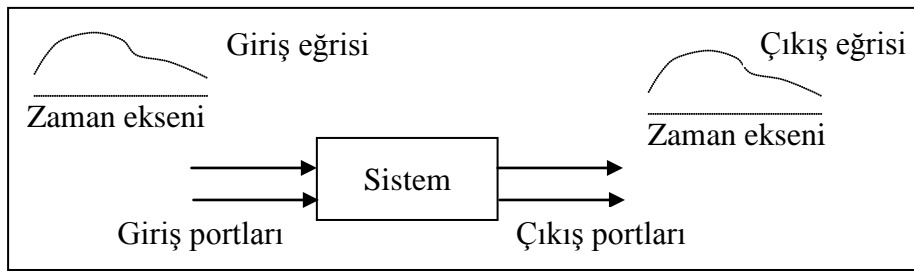
Bu yollar arasındaki temel fark, dinamiklerle, sistemlerin zaman üzerindeki davranışıyla ilgilenen benzetimi tanımlayan Sistem Tanımlama Hiyerarşisidir. Bu nedenle, zaman, sıralanan veya düzenlenen bütün olayların üzerinde kurulmuştur [45][23].

Tablo 3.1. Sistem Tanımlama Düzeyleri Hiyerarşisi

Düzyey	Tanımlama Tipi	Klir’deki Karşılığı	Açıklama
0	Deneysel Çerçeve	Kaynak Sistem	Girişlerle sistem benzetimi nasıl yapılır; hangi değişkenler ölçüm için ve zaman boyutunda gözlem nasıl yapılır?
1	Girdi / Çıktı Davranışı	Veri Sistemi	Kaynak sistemden toplanan zaman sıralı veri topluluğu; giriş ve çıkış parçalarından oluşur
2	Girdi / Çıktı Fonksiyonu		Başlangıç durum bilgisi; verilen başlangıç durumu, her bir değer tek bir çıkış üretir.
3	Durum Geçişi	Üretici Sistem	Durumlar girişlerden nasıl etkilenir? Giriş değerlerinden sonra durum ve çıkışa etkisi nedir?
4	Bileşik Bileşen	Yapı Sistemi	Bileşenler ve birbirleriyle etkileşimi. Bu tanımlamalar farklı çözünürlük seviyelerinde olabilir.

Sistemlerin, diğer sistemler ile etkileşimine aracılık eden giriş ve çıkış arabirimlerine sahip olduğunu varsayabiliriz. Şekil 3.3’te gösterildiği gibi [45], sistemler, kendi giriş portları aracılığıyla zaman sıralı girişleri alırlar ve kendi çıkış portlarından

cevap verirler. “Port” terimi, ister girişi benzetim ederek veya ister çıkışı gözlemleyerek olsun sistem ile özel etkileşim yapma araçları anlamına gelir. Sistemlerde zaman-indeksli girişler, giriş eğrileri, aynı şekilde, zaman-endeksli çıkışlar, çıkış eğrileri olarak adlandırılır. Portlar, sadece bir tanesi aracılığıyla sistem ile etkileşim yapılabilen kanallardır. Bu sistemin modüler olduğu anlamına gelir. Klir’in çerçevesi dinamikleri, giriş/çıkış portlarını ve modüleriteyi içerse de, o bu kavramları açıklamaz. Fakat bu kavramlar modelleme ve benzetim içinde ortaya çıkan problemlerin etkin bir şekilde çözülmesinde önemli rol oynarlar [45].



Şekil 3.3. Giriş çıkış sistemi

3.2.3. Sistem sınıflama özeti

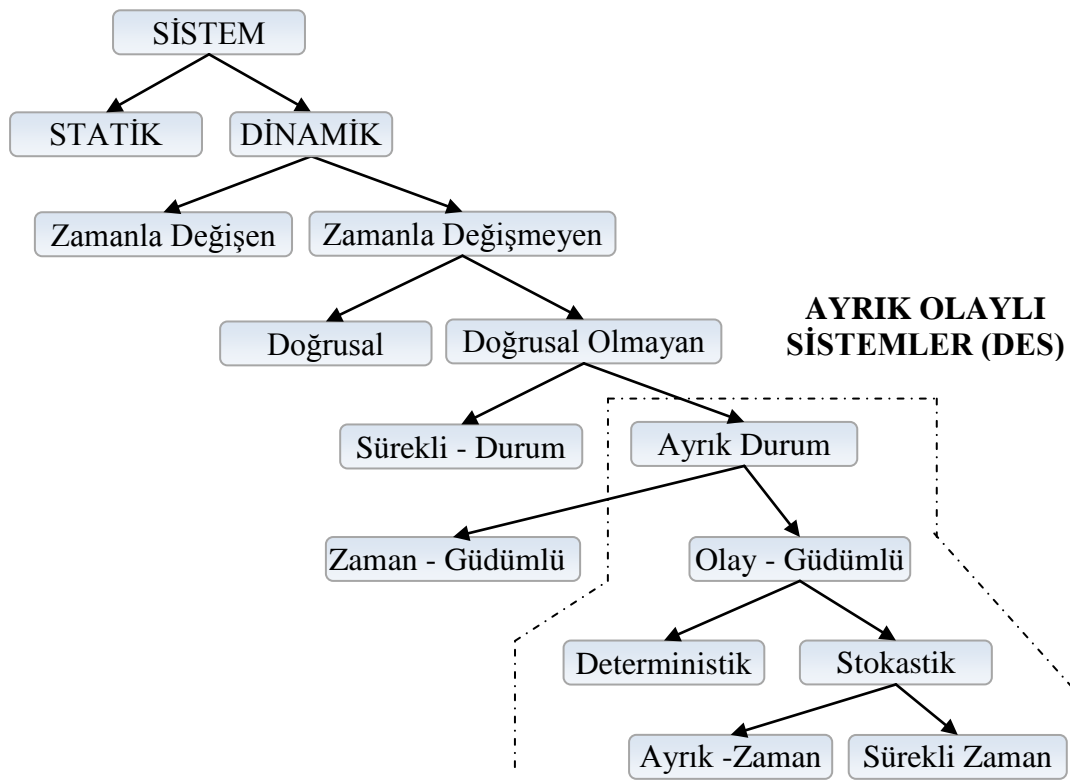
Aşağıda tanımlanan genel sistem sınıflandırması kesin olmamakla birlikte, sistem ve kontrol teorisini farklı bakış açılarından açıklamamıza ve ayrık olaylı sistemlerin anahtar özelliklerini tanımlamamıza yardımcı olacaktır. Şekil 3.4’de vurgulanmak istenen sınıflandırmayı desteklemek açısından bir referanstır [58].

Statik ve Dinamik Sistemler: Statik sistemlerde çıkış girişin geçmiş değerlerinden bağımsız iken; dinamik sistemlerde çıkış girişin geçmiş değerlerine bağımlıdır / dayanır. Diferansiyel veya fark denklemleri genellikle dinamik sistemlerin davranışlarını tarif etmekte kullanılır.

Zamanla Değişen ve Değişmeyen Sistemler: Bu tür sistemlerin davranışları zamanla değişmez. Bu özellik, durağanlık (stationarity) olarak adlandırılır. Bu tür sistemlerde sistem girişine belirli bir giriş verince, hep sabit bir çıkış alınır.

Doğrusal ve Doğrusal Olmayan Sistemler: Doğrusal sistemler $g(a_1u_1 + a_2u_2) = a_1g(u_1) + a_2g(u_2)$ ile tanımlanır, burada g çıkış a_1, a_2 gerçekte sayılar ve u_1, u_2 giriş vektörleridir.

Sürekli ve Ayrık Durumlu Sistemler (Continuous-State and Discrete-State): Sürekli durum sistemlerinde durum değişkenleri gerçekte veya karmaşık bir sayıdır. Ayrık durumlu sistemlerde ise durum değişkenleri ayrık kümelerin elemanlarıdır (pozitif tamsayı, vb.).



Şekil 3.4. Genel sistem sınıflandırma şeması

Zamana Dayalı ve Olaya Dayalı Sistemler: Zamana dayalı sistemlerde, zaman değiştikçe durumda değişmektedir. Olaya dayalı sistemlerde olaylar, sadece eş zamanlı olmayan ayrık olayların anlık durum geçişlerinde değişmektedir. Olayların gerçekleştiği aralıkta durum değişmez.

Deterministik ve Stokastik Sistemler: Rastgele bir veya daha fazla çıkış değişkeni olduğu zaman sistem olasılıklı (stochastic) olarak tanımlanır. Bu durumda sistem durumları olasılıklı işlemler ile açıklanır ve olasılıklı çerçeve sistem davranışının tanımlanmasını içerir

Ayrık Zamanlı ve Sürekli Zamanlı Sistemler: Sürekli zamanlı sistemlerde tüm girişler, durum ve çıkış değişkenleri muhtemel tüm zaman değerleri içindir. Ayrık zamanlı sistemlerde bu değişkenlerin bir veya daha fazlası genellikle örnekleme işlemleri sonucu zamanda ayrık noktalarda tanımlanır [58].

3.3. Dinamik Sistemler Formalizminde Temel Yaklaşımlar

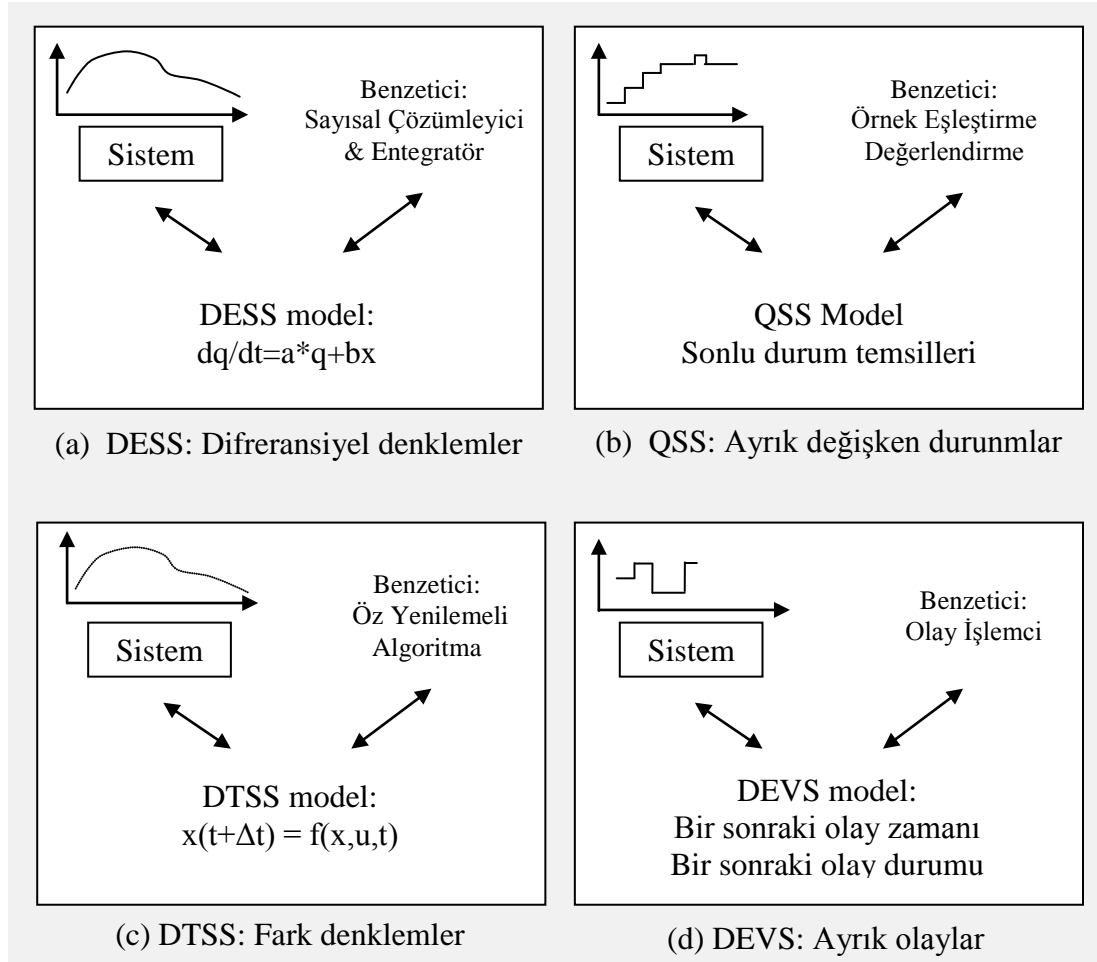
Diferansiyel Denklemlerli Sistem Tanımlama (Differential Equation System Specification - DESS): zaman ekseninin sürekli olduğu ve sistem veri tabanındaki eğrilerin zaman fonksiyonlarının da sürekli olduğu kabul edilen sistemlerdir. Modeller (sistem tanımlamaları) durum değişkenlerinin değişim oranları belirtilen diferansiyel denklemler ile tanımlanır. Bu görüş sayısal çözümler, sayısal toplayıcılar ve diferansiyel cebirsel denklem (DAE) çözümleridir [59].

Ayrık Zamanlı Sistem Tanımlama (Discrete Time System Specification - DTSS): Zaman eksenini ayrık, sistem veri tabanındaki eğriler buna bağlı olarak zamanda sıra ile işaretlenmiştir. Modellerin bir durumdan diğer duruma nasıl geçecekleri fark denklemleri ile ifade edilir [59].

Nitel Sistem Tanımlama (Qualitative System Specification - QSS): Zaman eksenini sürekli ve sonlu bir durum değişkenleri kümesi varsayar. Modeller sistem durumlar kümesine göre türetilir. Benzeticiler kümelerine göre sistem çıkış ve cevapları için girişleri belirler.

Ayrık Olaylı Sistem Tanımlama (Discrete Event System Specification - DEVS): Bu sistemlerde, zaman eksenini sürekli ve eğriler sistem veri tabanında parçalı tanımlıdır. Durum değişkenleri zamanın farklı periyotları için sabit kalır. Durumun değişmesi

“olay” olarak adlandırılır. Modeller olayların nasıl planlanacağını ve durum geçişlerine nelerin sebep olacağını tanımlar. Ortak benzetimler olay sürecini model tarafından belirtildiği gibi yönetirler [59].



Şekil 3.5. Sistem tanımlama formalizmleri

Aşağıdaki kısımda, yukarıda tanımlı sistemlerden ayrık olaylı sistemler detaylandırılmaktadır.

3.4. Ayrık Olaylı Sistemler ve DEVS Modelleme ve Benzetim Teorisi

DEVS (Discrete Event System Specification - DEVS), ayrık olaylı sistemler için sistem teorisi ve modelleme kavramlarını daha özel bir biçimde ifade eden ayrık olaylı sistem tanımlama yaklaşımıdır. Sadece ayrık olaylı modeller için değil aynı

zamanda ayrık zamanlı ve diferansiyel denklemlerle ifade edilen davranışları uyarlamak için bir hesaplama temeli oluşturması nedeni ile önemlidir [48].

Paralel DEVS, modern hesaplama teorisinde çok önemli bir yere sahip olan paralelliğin faydalı bir şekilde kullanılmasını engelleyen ve ilk zamanlardaki bilgisayarların sıralı operasyonları nedeniyle ortaya çıkan sınırlandırmaları ortadan kaldırmaktadır. Bu bölümde daha sonraki kısımlarda tartışılacak DEVS'in nesneye-yönelik çerçeveye uygulanması işlemine hazırlık amacı ile DEVS modelleme yaklaşımı konusu tanıtılmaya çalışılacaktır.

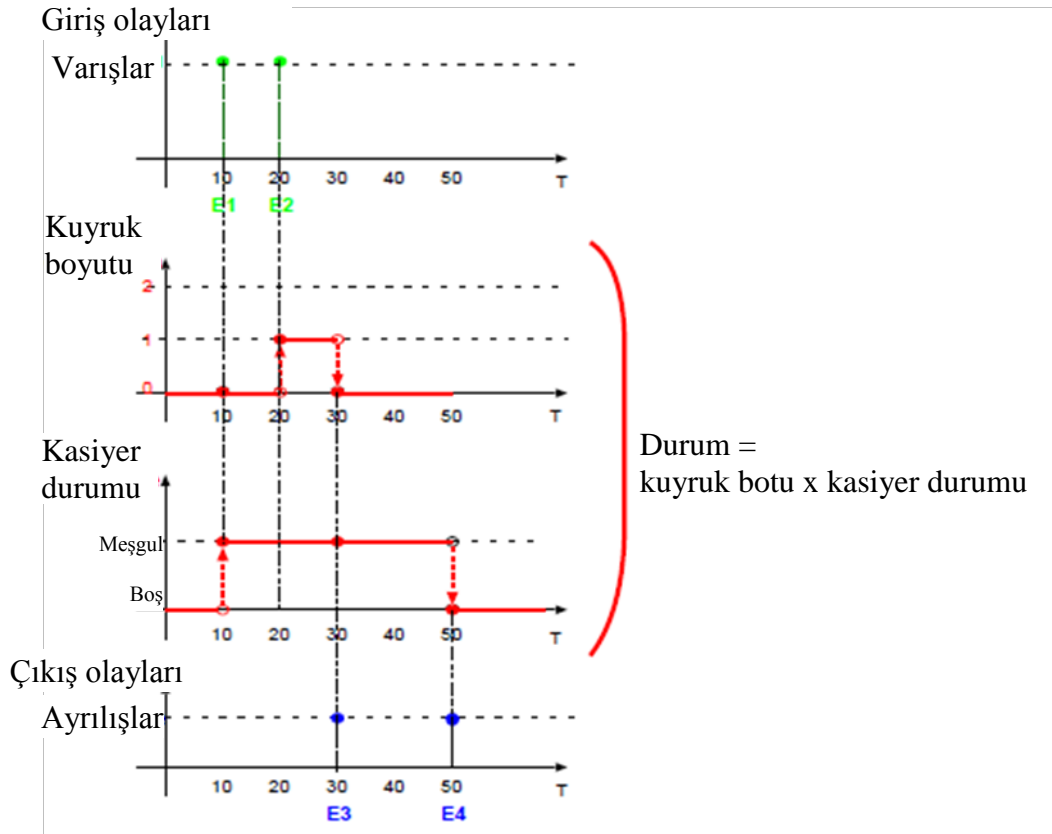
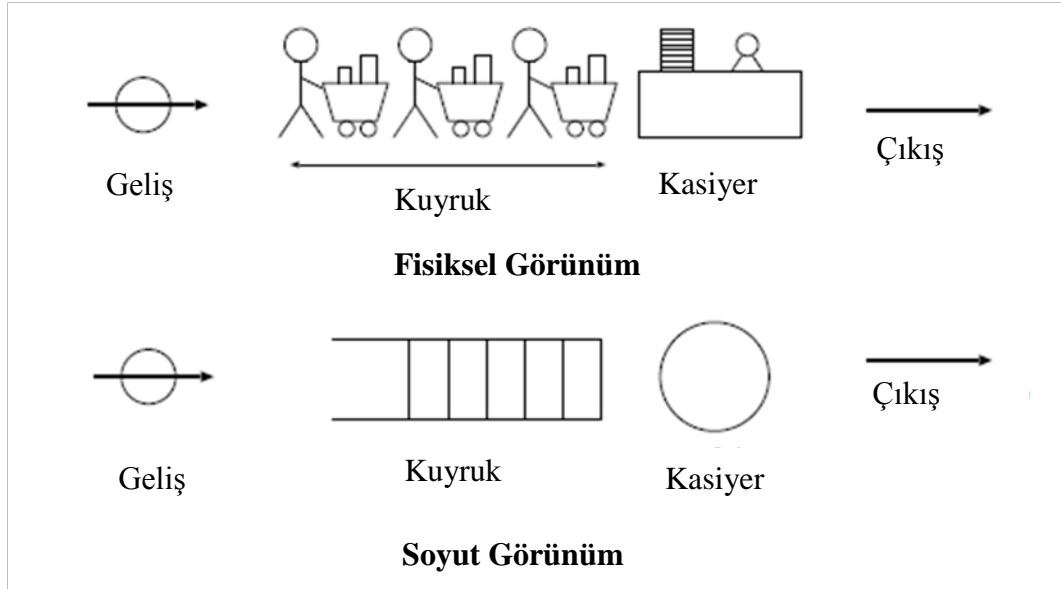
3.4.1. Ayrık olaylı modelleme yaklaşımı

Zaman ekseninin sürekli olduğu, ancak sadece sınırlı bir zaman periyodunda sonlu sayıda olayların meydana geldiği sistemler, 'ayrık olay' olarak adlandırılan bir soyutlama seviyesinde ele alınmıştır. Bu olaylar sistemin durumunda bir değişikliğe neden olabilmelerine karşılık, olaylar arasında sistemin durumu değişmez. Bu sebeple 'ayrık olaylı modeller', sistemin durumunun zaman üzerinde sürekli değiştiği sürekli modellerden ayrılırlar [48].

Ayrık olaylı modelleme yaklaşımları yüksek bir soyutlama düzeyinde ele alınır. Bu soyutlama, sistem davranışının gerçeğe uygun bir şekilde tanımlanması için uygundur. Diğer taraftan, olaylar arasında sistemin durumunun değişmemesi nedeni ile ayrık olaylı benzetici olay olmayan zamanlarda sistemin durumunu işlemeye gerek duymaz. Bu, durum bilgisinin sürekli bir zaman ekseninde her noktada gösterilmesi gerektiği sürekli benzetim ile karşılaştırıldığında, son derece verimli bir benzetime neden olur.

Yüksek bir soyutlama düzeyi, gerçek dünya davranışıyla ilgisi olmayan benzetim araçlarını getirebilir. Özellikle birden fazla olayın aynı zaman diliminde meydana geldiği olay çakışmaları, ayrık olaylı modelin yeteri kadar detaylandırılmamasından kaynaklanabilir. DEVS modelleme yaklaşımı ve onun türevleri, oluşan olay

çakışmalarını başarılı bir şekilde yönetir ve bu durumların ortadan kaldırılması konusunda güvenilir bir çözüm yolu sunar [48].



Şekil 3.6. Kasiyer / kuyruk sisteminin davranış örneği

Şekil 3.6’da kasiyer/kuyruk sisteminin örnek bir davranışı görülmektedir. Müşteri gelişi ile geçecek işlem tarif edilmiştir.

Fiziksel seviyede kasiyer bir anda 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 karşılaştırılır. Kuyruk idaresi, ‘ilk giren ilk çıkar’ (First In First Out - FIFO) yöntemine göre ele alınır [49].

3.4.2. Ayrık olaylı sistemlerde kullanılan terimler

Ayrık olaylı benzetimde kullanılan terimler aşağıda tanımlanmaktadır [49][48].

An: Bir nesnenin en az bir karakteristiğine atanabilen bir zaman değeridir.

Aralık: Birbirini izleyen iki an arasındaki süredir.

Zaman Periyodu (time-span): Bir veya daha fazla aralığının ardı ardına gelmesidir.

Nesnenin Durumu: Belirli bir anda bir nesnenin tüm karakteristik değerlerinin listesidir.

Aktivite: Bir aralık boyunca bir nesnenin durumudur.

Olay: Bir nesnenin durumunda bir anda olan bir değişimdir ve o ana kadar olması engellenen bir aktiviteyi / işlemi başlatır. Olayın oluş şartı yalnızca sistem zamanına bağlıysa, bu olay ‘belirlenmiş’ bir olaydır ve bu özelliğe sahip modeller, benzetim modellemede, ‘zaman olayı’ olarak isimlendirilir. Zamana bağımlı olmadan sistemin şartlarına bağlı olarak gelişen / oluşan olaylar tesadüfidir (contingent) ve bu tür olaylar modellemede, ‘durum olayı’ olarak isimlendirilir.

Nesne aktivitesi: Durum deęişimlerine sebep olan birbirini izleyen iki olay arasındaki nesnenin durumudur. Dięer olaylar, dięer nesnelerdeki durum deęişimlerine göre meydana gelebilir.

İşlem (process): Bir zaman periyodu boyunca bir nesnenin birbirini takip eden durumlarıdır. Bir veya daha fazla nesne aktivitesinin ardı ardına gelmesi, 'işlem' olarak adlandırılabilir.

Belirli bir problem için, olayların ne olduğunu belirlemek / anlayabilmek için aşağıdaki adımlar takip edilebilir:

- Nesnelerin ve nesne karakteristiklerinin tespit edilmesi,
- Sistemin karakteristiklerinin belirlenmesi ve
- Bir olay olarak, herhangi bir karakteristik deęerinde deęişimin nedenlerinin tanımlanmasıdır.

Çoęu zaman sayıcılar, minimum deęerler, maksimum deęerler, ortalamalar, deęişkenlerin frekans dağılımları, vb. başarımların hesaplanmasına yardımcı olacak ilave durum deęişkenleri modele eklenir. Ayrık olaylı benzetimde, ortalama kuyruk uzunluğu ve kaynakların kullanımı gibi başarımların ölçütleri sık sık kullanılır. Başarımların ölçütleri bir benzetim çalışmasının sonucunda elde edilen çıktılarıdır.

Bir benzetim modelinin 'statik' ve 'dinamik' olmak üzere iki farklı yapısı bulunmaktadır. Statik yapı; modelin olası durumlarını tanımlarken, dinamik yapı; durumun zaman üzerindeki deęişimini tanımlar. Statik yapı genellikle nesnelere ve nesne karakteristikleriyle ifade edilirken, bir modelin dinamik yapısının tanımlanmasında 'dünya görüşleri' olarak bilinen farklı yöntemler kullanılır.

Aşağıda, ayrık olaylı benzetimin kaynağını oluşturan çeşitli dünya görüşleri sunulmaktadır [49][48].

3.4.3. Ayrık olaylı benzetim stratejileri

Benzetim dillerinde yaygın olarak kullanılan üç farklı ayrık olaylı benzetim stratejisi bulunmaktadır [45].

- Olay zamanlama (event scheduling),
- Aktivite tarama (activity scanning) ve
- Süreç etkileşimi (process interaction) .

“Dünya görüşleri” olarak da adlandırılan bu stratejilerin her biri diğerlerinden daha doğal bir şekilde ifade edilebilir bir model tanımlama biçimi ortaya koyar. Bir başka deyişle, her bir benzetim stratejisi dünyanın çalışma şeklini ele alırken orijinal yaklaşımlara sahiptirler. Yukarıda isimleri verilen stratejilerden sonuncusunun ilk ikisinin bir kombinasyonu olması yanında günümüzde geliştirilen benzetim ortamlarının çoğu bu stratejilerin bir kombinasyonunu kullanmaktadır [48].

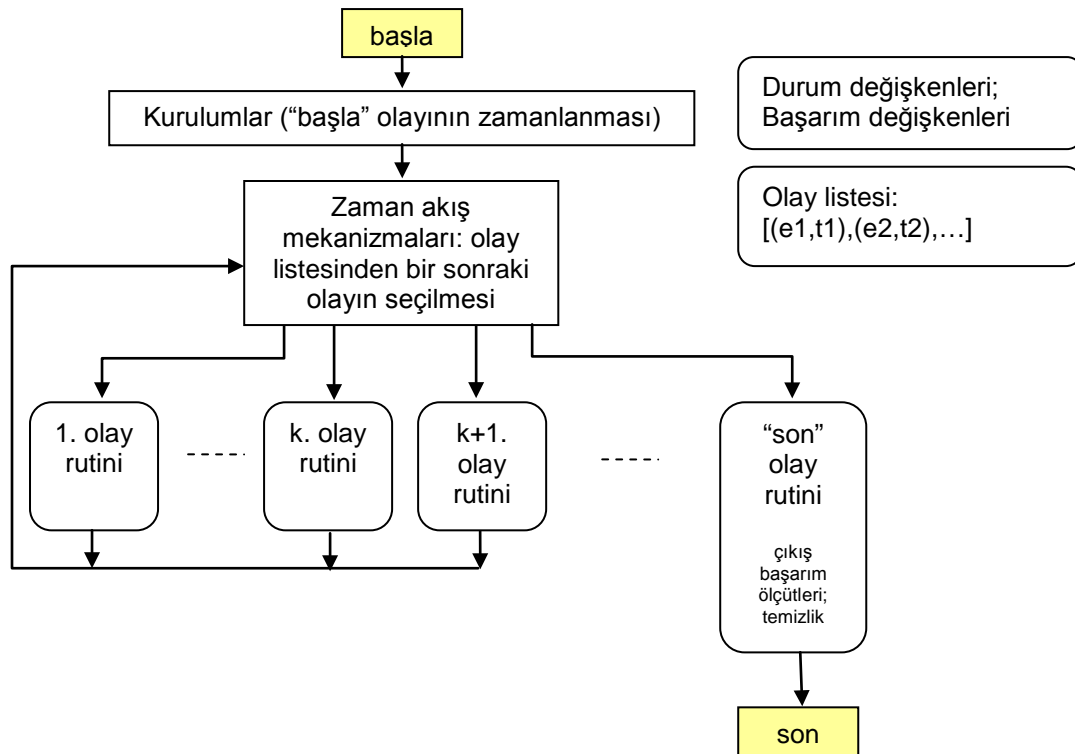
3.4.3.1. Olay zamanlama benzetim stratejisi

Olaya yönelik modelleme ve benzetimde bütün olaylar önceden planlanır. Bir olayın planlanması, olayın gerçekleşmesi için gerekli bütün şartlar önceden bilinebildiği durumlarda mantıklı olur. Olay zamanlama 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 [48].

Şekil 3.7’de görüldüğü gibi bir olay zamanlama benzetim çekirdeği iki (global) 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. Durum değişkenleri, durum değişkenlerinin ve kombinasyonlarının minimumunu, maksimumunu, ortalamasını ve standart sapmasının hesaplanması için ilave başarımlar değişkenleriyle artırılabilir. Bir olay zamanlama çekirdeği, olayları

'olay listesi' şeklinde düzenleyerek (artan zaman sırasına göre), listenin en üstündekini kaldırıp işleyerek ve bunu liste tamamen boşalana kadar tekrarlayarak çalışır. Olay bildirimindeki olay zamanı, benzetim zamanını ilerletmek için kullanılır. Olay bildirimindeki olay türüne bağlı olarak, uygun olay bildirim yapılabılır. Bu işlem, olay bildirimlerini olay listesine koyarak sistemin durumunu değiştirebilir ve gelecekte yeni olaylar zamanlayabilir. Olay zamanlama yaklaşımında:

- Sistemin kurulumu ve ön zamanlamalı olaylar bir '*başlangıç*' olayına konulabilir. Bu olay otomatik olarak olay listesine yerleştirilir ve daha sonra normal bir olay gibi işlenir.
- Belirli bir benzetim zamanında benzetimin sonlandırılması, benzetim prosedürü tarafından son olay olarak işlenecek olan özel bir 'son' olayının zamanlanmasıyla halledilir. Bu olay başarımlar ölçümleri çıkışını ve elde edilen diğer istatistikleri üreten işlem yönergelerini içerir.

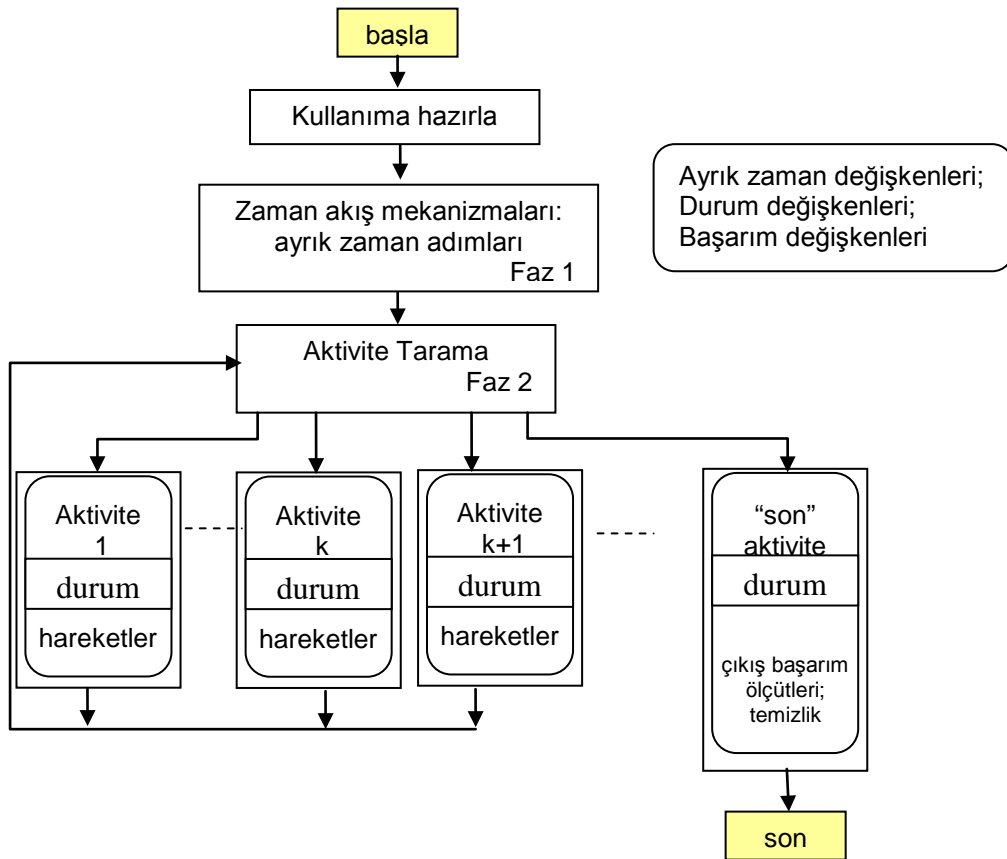


Şekil 3.7. Olay zamanlama benzetim çekirdeği

Bir olay zamanlama modeli, benzetim zamanını ilerleten, olay listesini ve sistem durumunu güncelleyen yinelemeli bir benzetim prosedürüyle benzetim edilir.

3.4.3.2. Aktivite tarama benzetim stratejisi

Aktivite tarama yönteminde, bir takım aktiviteleri etkinleştiren şartlar tanımlanır. Bu yaklaşım, Prolog gibi bildirimli yapay zekâ (Artificial Intelligence - AI) dillerinde kullanılan yaklaşımla benzerlikler taşır. Bir aktivite tarama benzetimi zamanı ilerletmek için ayrık zaman adımını kullanır. Aktivite tarama safhasında, çözücü (solver) aktivitenin şartı sağlayıp sağlamadığını kontrol eder ve eğer şart doğruysa işler. Bu tarama işlemi en azından bir aktivite şartı doğru olarak değerlendirilene kadar devam eder. Eğer aktivitelerin hiçbiri şartı sağlamazsa, zaman akış safhası zamanı ilerleterek tekrar çalıştırılır. Petri ağları, aktivite tarama tabanlı bir modelleme yaklaşımına örnektir [48].



Şekil 3.8. Aktivite tarama benzetim stratejisi

3.4.3.3. Süreç etkileşim benzetim stratejisi

Aktivite tarama sabit bir zaman adımı kullanır ve bu pek çok durumda verimli değildir. Davranışı doğru bir şekilde modellemek için iki olay arasındaki zaman aralığı olabildiğince küçük seçilmelidir. Diğer taraftan, bazı olaylar zaman bakımından çok seyrek olabilir. Bu gibi uzun zaman aralıklarında, aktivite tarama benzeticisi, şartlar değişmemesine rağmen zamandaki her noktada tüm şartları gereksiz bir şekilde kontrol eder.

Süreç etkileşim benzetim stratejisinde, aktivite tarama yaklaşımı olay zamanlama yaklaşımıyla birleştirilir. Aktiviteler, olay zamanlama benzetim stratejisindeki gibi gelecekteki bir zamana programlanır. Buna ek olarak, olay zamanlarında, bütün aktivite şartları aktivite tarama benzetim stratejisindeki gibi kontrol edilir.

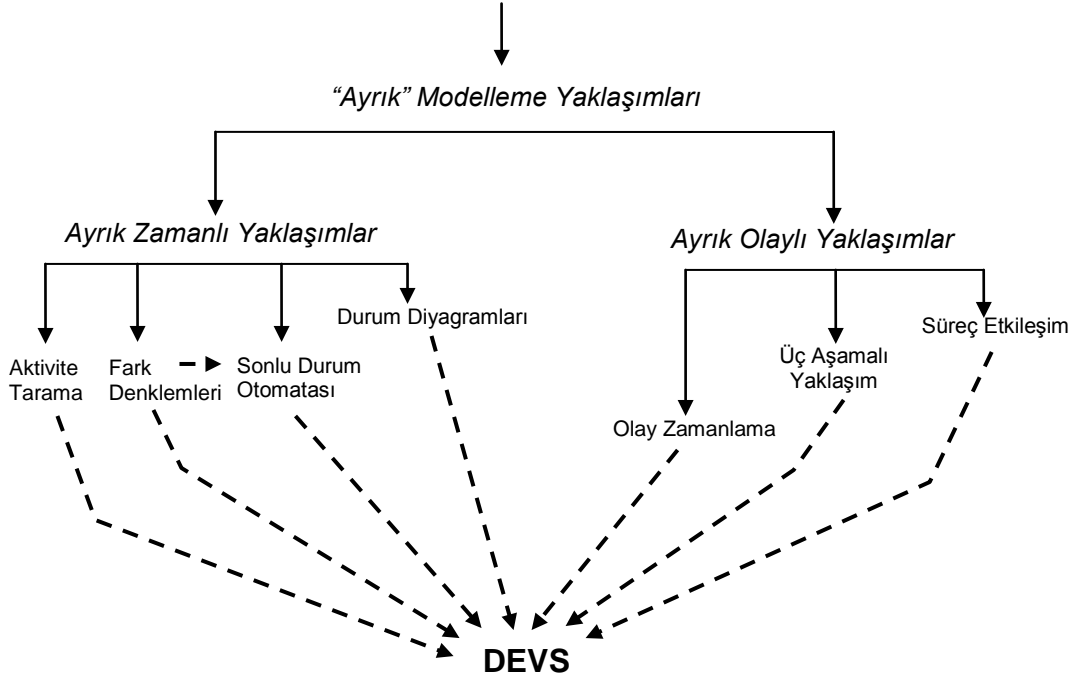
3.4.4. Ayrık olaylı benzetim stratejileri arasındaki ilişkiler

Şekil 3.8, farklı ayrık yaklaşımlar arasındaki ilişkilerin bir özetini göstermektedir. Şekilde sol tarafta zaman akış mekanizmasının sabit bir zaman ilerlemesi şeklinde olduğu yaklaşımlar sağ tarafta ise, ‘ayrık olaylı’ zaman akışına (saat, sadece olayların olduğu zamanlarda ilerler) sahip yaklaşımlar görülmektedir.

Ayrık olaylı modelleme yaklaşımları modüler olmayan bir formdadırlar. Yani olay işleyicisi (event handler), aktiviteler, işlem blokları, vb. model bileşenleri bir takım arabirimler aracılığıyla kendi ortamıyla etkileşen varlıklar değildir. Bundan çok, direk olarak global durum değişkenleri ile birlikte diğer bileşenleri etkilerler.

Şekil 3.8’deki kesikli çizgiler, bir kaynak modelleme yaklaşımında tanımlı bir modelin hedef yaklaşımda tanımlı aynı modele dönüşümünü göstermektedir. Orijinal ve dönüşmüş modeller, aynı başlangıç şartlarında benzetim edildiğinde aynı durum eğrilerini üretirlerse “denk” olarak adlandırılırlar. Şekilde tüm dönüşümler bir sonraki bölümde tanımlanacak olan DEVS yaklaşımına yönelmiştir. Tüm bu dönüşümlerin özü, modüler olmayan tanımlamalardan modüler bir ‘DEVS Birleşik

modelinin' kurulmasıdır. DEVS birleşik modeli, modüler bileşenler arasındaki bağlantılar aracılığıyla bağımlılıklar açık bir şekilde gösterilerek elde edilir [48].



Şekil 3.8. Benzetim stratejilerinin sınıflandırılması

3.5. Ayrık Olaylı Sistem Tanımlama DEVS Modelleme Yaklaşımı

Atomik DEVS modelleme yaklaşımı, sistemin ayrık olaylı davranışının değişik yönlerini tanımlayan bir yapıdadır. DEVS atomik model tanımı, değişken zaman periyotlarına sahip parçalı sabit eğriler şeklinde durumları (değişken değerler) ve ilişkili zaman eksenini tanımlar. Atomik model tanımı yeni durum değerlerinin nasıl üretileceğini ve ne zaman bu yeni değerlerin etkin olacağını da açıklar. Paralel bir atomik DEVS modeli aşağıdaki yapıdadır [10][23]:

$$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \delta_{conf}, \lambda, ta \rangle$$

Burada;

X, giriş değerleri kümesi,

S, durumlar kümesi,

Y, çıkış değerleri kümesi,

$\delta_{int} : S \rightarrow S$ dahili geçiş fonksiyonu,

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

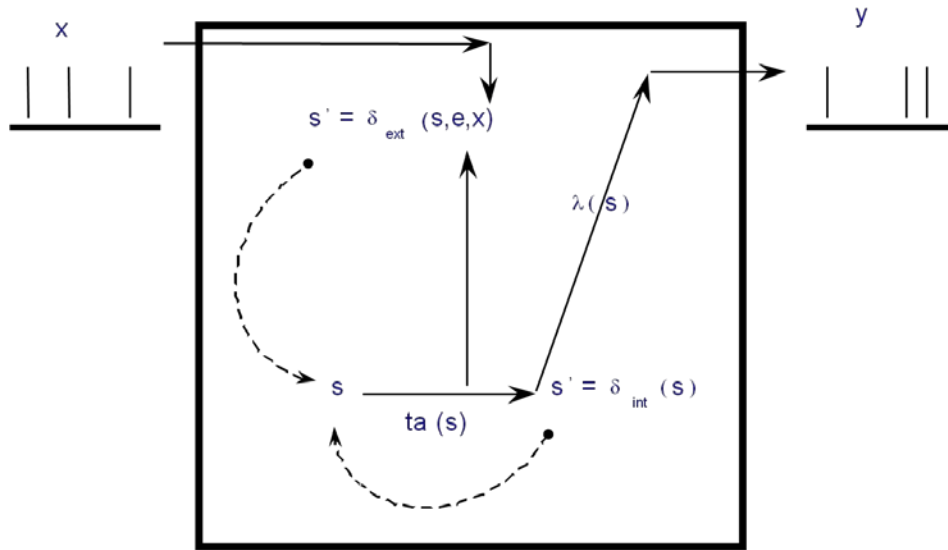
burada; $Q = \{(s,e) | s \in S, 0 \leq e \leq ta(s)\}$ toplam durum kümesi,

e , en son olan geçiştten bu yana geçen süredir.

$\delta_{ext} : Q \times X \rightarrow S$ çakışma (confluent) geçiş fonksiyonu,

$\lambda : S \rightarrow Y$ çıkış fonksiyonu,

$ta : S \rightarrow \mathbb{R}^+_{0,\infty}$ zaman ilerleme (time advance) fonksiyonu, 0 ve ∞ arasındaki pozitif reel sayılar kümesidir.



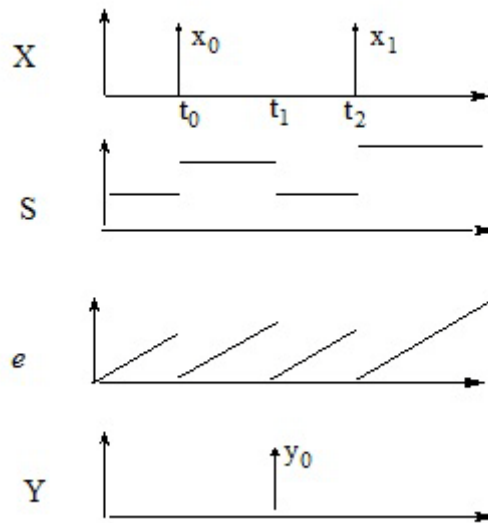
Şekil 3.9. DEVS işleyiş mekanizması.

Bu elemanların yorumu Şekil 3.9’da görülmektedir. Herhangi bir anda sistem bir ‘s’ durumundadır ve hiçbir harici olay meydana gelmemesi durumunda, sistem $ta(s)$ zamanı süresince ‘s’ durumunda kalır. Yukarıda yapılan tanımdan anlaşılacağı üzere ‘ $ta(s)$ ’ bir reel sayı olmasına rağmen ‘0’ ve ‘ ∞ ’ değerlerini de alabilmektedir. Zaman ilerleme fonksiyonu $ta(s) = 0$ olduğunda, ‘s’ durumunun süresi araya başka olaylar giremeyecek kadar çok kısadır – bu durumda ‘s’ durumunun bir geçici durum olduğunu belirtmek yanlış olmaz. İkinci durumda ($ta(s) = \infty$ olduğunda), harici bir olay bu durumu bozmadıkça sistem sonsuza kadar ‘s’ durumunda kalır ve bu durumda ‘s’ pasif bir durum olarak tanımlanır. Bir durumda kalma süresi dolduğu zaman (geçen süre $e=ta(s)$ olduğu zaman), sistem $\lambda(s)$ değerini çıkış olarak verir ve

$\delta_{int}(s)$ durumuna geçer. Bu anda, çıkışın dâhili geçişlerden hemen önce üretildiğine dikkat edilmelidir.

Eğer bir $x \in X$ harici olayı bitiş zamanından önce meydana gelirse (sistem $e \leq ta(s)$ ile (s,e) durumundaysa), sistem $\delta_{ext}(s,e,x)$ durumuna geçer. Bu nedenle dâhili geçiş fonksiyonu, en son geçişten itibaren hiçbir olay olmadığı zaman sistemin yeni bir duruma geçmesine neden olur. Bu durum, 'x' girişi, 's' şimdiki durumu ve sistemin bu durumda ne kadar kaldığını gösteren 'e' tarafından belirlenir. Her iki durumda da sistem, yeni bir $ta(s')$ sükûnet zamanı ile yeni bir s' durumundadır ve aynı olay devam eder.

Aşağıda Şekil 3.10'da ayrık olaylı sistemde giriş, durum, geçen süre ve çıkışların zamana bağlı gösterimi görülmektedir [45].



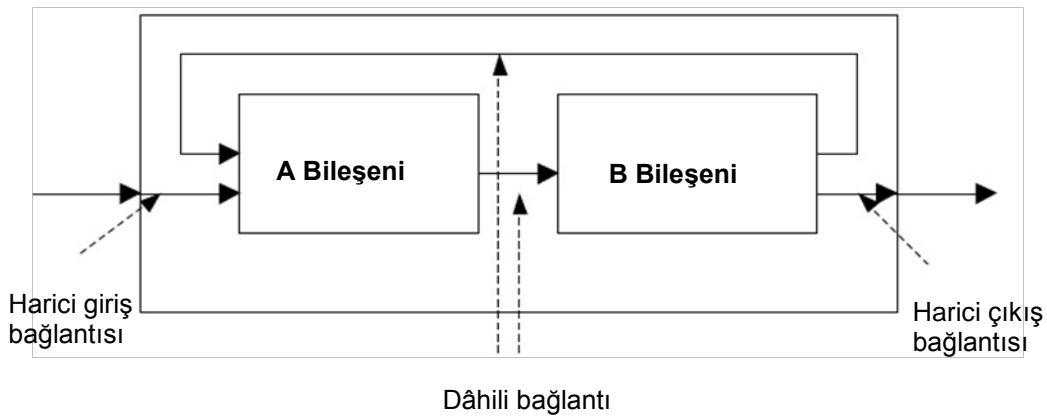
Şekil 3.10. Ayrık olaylı sistemde giriş, durum, geçen süre ve çıkışlar

3.6. Birleşik DEVS Modelleme Yaklaşımı

Birleşik model, bir takım bileşen modellerden ve bunların birbirleriyle olan bağlantılarından oluşur. Daha öncede ifade edildiği üzere bileşenler 'atomik' veya 'birleşik' olabilir. Birleşik modelin davranışı bileşenlerin davranışıyla ve / veya

bağlantısıyla tanımlanabilir. Bağlantı üç türe ayrılır: Harici giriş bağlantısı, harici çıkış bağlantısı ve dâhili bağlantı (Şekil 3.11).

Harici giriş bağlantısı; birleşik modelin kendisi ve bileşenlerin bir veya birden fazlasının arasında olur. Harici çıkış bağlantısı ise; bileşen çıkışları ile birleşik modelin çıkışları arasındaki bağlantıdır. Dâhili bağlantı; bileşen çıkışları ve bileşen girişleri arasında yapılır.



Şekil 3.11 Birleşik DEVS yaklaşımında bağlantılar

Birleşik bir DEVS modelleme yaklaşımı aşağıdaki yapıdadır;

$$CM = \langle X, Y, D, \{M_i\}, EIC, EOC, IC, Select \rangle$$

Burada;

X, Y : giriş ve çıkış kümeleridir

D : birleşik modelin bileşenler kümesidir

Her $i \in D$ için,

M_i : atomik veya birleşik olabilen bir bileşenin DEVS modelidir;

$EIC \subseteq X \times \cup_i X_i$, harici giriş bağlantı ilişkisi;

$EOC \subseteq \cup_i Y_i \times Y$, harici çıkış bağlantı ilişkisi;

$IC \subseteq \cup_i Y_i \times \cup_j X_j$, dâhili bağlantı ilişkisi;

$Select: 2^{\{M_i\}} - \emptyset \rightarrow \{M_i\}$, eşitlik fonksiyonudur.

DEVS formalizmi temel olarak ayrık olaylı sistemlerin tanımlanmasında kullanılır. DEVS formalizminin önemli bir özelliği de modelleme ve benzetim bölümleri arasında açık bir ayrıma izin vermesidir. Bunun anlamı herhangi bir sistemin davranışını fazından bağımsız modelleyebiliriz [23].

Modelin davranışının tanımlanması kadar sistem modelinde çeşitli düzeylerde tanımlayan güçlü bir modelleme ve benzetim aracıdır. DEVS formalizminin önemli bir özelliği her bir modelin otomatik olarak benzetimini desteklemesidir. DEVS modelleme ve benzetimi birbirinden ayırır. Bu yüzden her bir model özel benzetici olmaksızın çalıştırılabilir. Her bir model, davranışını yöneten bir benzeticiye, her bir birleşik modelde, bileşenlerinin uyumlu çalışmasını sağlayan bir koordinatöre sahiptir. DEVS benzetim sistemlerinin mimarisi hiyerarşik ve modüler DEVS formalizmi ile benzetim kavramlarından gelir [23].

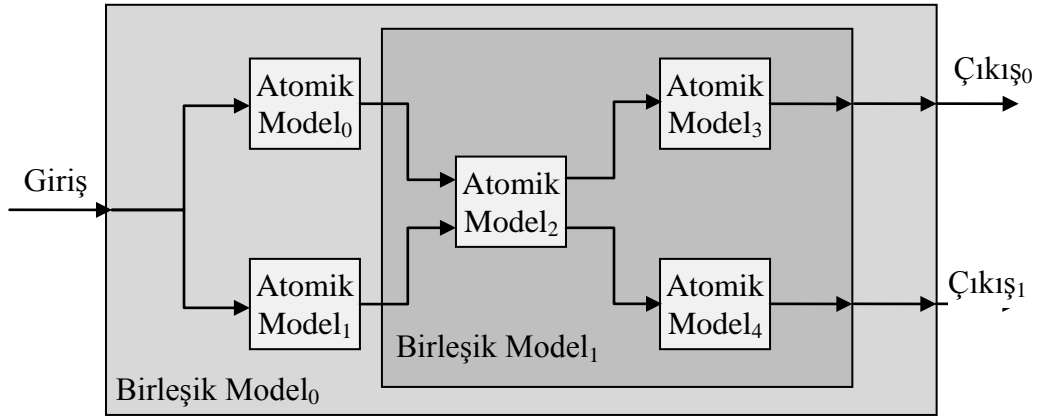
3.7. Hiyerarşik Model Tasarımı: DEVS Birleşim Çerçevesi

Birleşik bir model DEVS modelleme yaklaşımında temel bir model olarak ifade edilebilir. Bu temel model daha büyük bir birleşik model içinde kullanılabilir. Bu, DEVS yaklaşımının hiyerarşik model yapısı için gerekli olan birleşim altında kapalı olduğunun göstergesidir. Bir birleşik modeli denk bir temel model olarak ifade etmek, tüm davranışı vermek üzere bileşenlerin etkileşmesi yoluyla yapılır. Daha önce de ifade edildiği gibi, birleşik bir modellerin atomik modüller gibi daha büyük sistemlerde bir bileşen olabilir ve böylece hiyerarşik bir yapı olur [49][48].

Bir bağlantı tanımının bir takım modellere uygulanması yoluyla nasıl bir birleşik model üretildiği Şekil 3.12'de gösterilmektedir. Bu modeli daha büyük bir sistemde yeni bileşenlerle birlikte bir bileşen olarak kullanarak ve bağlantı bilgisini de ekleyerek, hiyerarşik bir birleşik model elde edilir.

Şekil 3.12'de birleşik modelin hiyerarşik yapısı görülmektedir. BirleşikModel₀ bir giriş portu, iki çıkış portu, iki atomik model (AM₀,AM₁) ve bir birleşik modele sahiptir. Birleşik Model₁ atomik model AM₂ AM₃ ve AM₄ ü kapsamaktadır. Dahili

bağlantılar ($AM_0 \rightarrow CM_1$ ve $AM_1 \rightarrow CM_1$), harici giriş bağlantıları ($IN \rightarrow AM_0$ ve $IN \rightarrow AM_1$), harici çıkış bağlantıları ($CM_1 \rightarrow OUT_0$ ve $CM_1 \rightarrow OUT_1$) ve alt modelin etkisi ($CM_1 = \{AM_0, AM_1\}$ veya AM_0 and AM_1 CM_1 etkiler [23]).



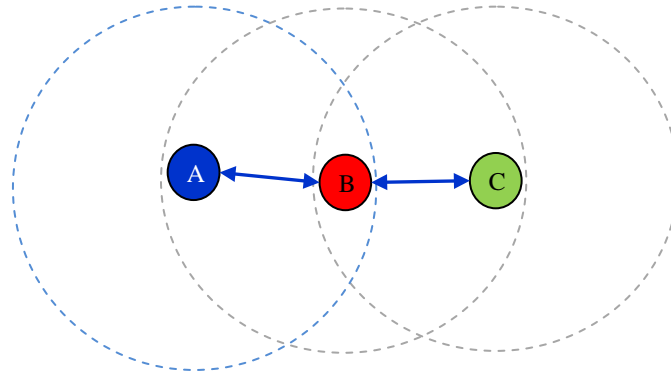
Şekil 3.12. DEVS birleşik modeli

BÖLÜM 4. HAREKETLİ AD HOC AĞLAR (MOBILE AD HOC NETWORKS – MANET) ve YÖNLENDİRME PROTOKOLLERİ

4.1. Giriş

Kablosuz veri iletişim aygıtları ve gezgin işlemlerde yakın zamandaki gelişmeler, gezgin kullanıcılar arasında kablosuz iletişimi daha önce olmadığı kadar popüler kılmıştır [3] [60].

Ad Hoc ağlar; geçici formlarda, merkezi yönetimin bulunmadığı veya geleneksel ağlardaki düzenli servis desteklerinin olmadığı gezgin düğümlerden oluşan ağlardır. Düğümler rastgele ve keyfi olarak kendilerini organize ederek hareket etmekte serbesttir. Bu yüzden ağın topolojisi hızlı ve öngörülemeyen bir şekilde değişebilir [2].



Şekil 4.1. Basit bir Ad Hoc ağ görünümü

Şekil 4.1'de basit bir Ad Hoc ağı görülmektedir. Şekilde A, B ve C ağdaki düğümleri, kesikli çizgilerle oluşturulmuş daireler düğümlerin radyo kapsama alanlarını, ok işaretleri ise bağlanabilirliği temsil etmektedir. Burada A düğümü ile B

düğümü ve B düğümü hem A hem C düğümü ile, C düğümü B düğümü ile iletişim halindedir. A ve C düğümler birbiriyle iletişim kurmak istediklerinde bunu ancak B üzerinden gerçekleştirebileceklerdir.

4.2. Uygulama Alanları

MANET'lerin uygulama alanları Tablo 4.1'de özetlenmiştir [2].

Tablo 4.1. MANET'lerin uygulama alanları.

Ağ Türü	Uygulama Alanı
Taktik ağlar	Askeri iletişim, operasyon, otomatize edilmiş savaş alanları
Algılayıcı ağlar	Yerel veya uzaktan akıllı algılayıcı düğümlerin kontrol edilmesi, çevresel uygulamalar: hayvanların hareketlerinin izlenmesi, çevresel kirlenme sulama vb.
Acil servisler	Arama ve kurtarma operasyonları, hasta kayıt, durum, veri vb. iletimi
Ticari	Her yerden elektronik ödeme (taxi vb.), gezgin ofis, servis araçları, haberlerin iletimi (yol drumları hava muzik vb.), Araç kazaları kılavuzu için yerel Ad Hoc ağlar.
Ev ve iş ağları	Ev/iş kablosuz ağ (WLAN), kişisel ağlar (personal area network - PAN) paylaşımlı beyaz tahta uygulamaları, fuarlar vb.
Eğitim uygulamaları	Sanal sınıf ve konferans sınıflarının oluşturulması ve Ad Hoc iletişimin konferans toplantı veya ders boyunca sürdürülmesi
Eğlence	Robot evcil hayvanlar, dışarıdan internet erişimi vb.
Konuma duyarlı servisler	Servis takibi, otomatik arama yönlendirme bulunulan yere yönelik işlerin iletimi. Bilgi servisleri (yere özel reklam servisleri, turizm klavuzu vb.).

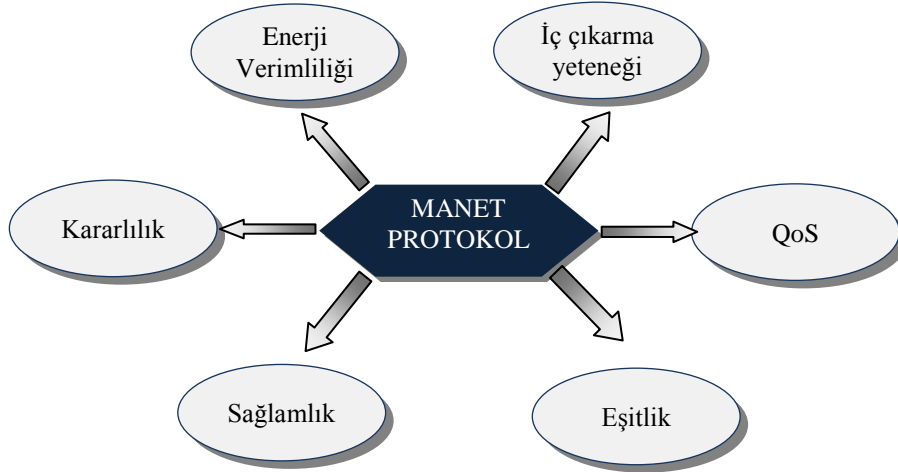
4.3. Başarım Parametreleri

MANET'lerin kendine özgü özellikleri başarım kriterlerine de yansımıştır. MANET protokollerin başarımlarının belirlenmesinde kullanılan parametreler Şekil 4.2'de özetlenmekte ve bu parametreler aşağıda tanımlanmaktadır [1].

İç çıkarma yeteneği (throughput): İletişim ağlarında bant genişliğinin veri transferi için kullanılan parçasıdır. Ek yük (paket başlıkları, kontrol paketleri, güvenlik vb.) için kullanım nedeniyle veri iletimi için bant genişliğinin 100% kullanımı mümkün

değildir. MANET protokollerinin amacı ek yük için kullanılan bant genişliğini diğer hedeflerden kısıtlamaya gitmeden mümkün olduğunca küçük tutmaktır.

Servis kalitesi (QoS): Düşük gecikme yüksek paket teslim oranı ve garanti edilmiş bant genişliği QoS'in tanımlanmış bazı ölçüm parametreleridir.



Şekil 4.2. MANET protokol başarımları parametreleri

Eşitlik (Fairness): Çıkış maksimum değeri tek düğümün süresiz kullanımına izin verilerek elde edilir. Bu ağdaki diğer düğümler için haksızlık olur. Eşitlik ağ kaynaklarının eşit dağılımı ile sağlanır (bant genişliği vb.). MANET protokolleri kullanıcılar için kanal erişiminin dinamik durumda eşit bir şekilde kullanılmasından sorumludur.

Sağlamlık (Robustness): Kablosuz iletişimde veri veya kontrol paketlerinin kaybolması istisnai bir durum değildir. Bazı MANET protokolleri koordinatör düğümler tarafından merkezi kontrole dayaması o düğümlerde problemlere yol açabilir (bataryalarının bitmesi, vb.). Bu nedenle sağlam MANET protokolleri düğümlerin paket kayıpları veya düğüm zayıflıkları olmaksızın kararsız olmadan normal işleyişi için tasarlanmıştır.

Kararlılık (Stability): MANET protokolleri dinamik sistemleri kontrol eder ve başarımları dinamik sistemlerde olduğu gibi değişkendir. Gerçekte pek çok protokol (IEEE 802.11, ALOHA vb.) eğer kanal erişim talebi belirli bir eşik değerinden büyük ise kararsızdır. Kararlı (stable) MANET protokolleri, kararsızlıktan kaçınmalıdır.

Enerji tüketimi: Sınırlı enerjinin gereksiz tüketilmesinden kaçınmak için enerjinin verimli kullanımı çok önemlidir. Ağın verimliliğini etkilemeyecek şekilde uyku zamanı boyunca MANET protokollerinde alıcı veya verici çalıştırılmaz.

4.4. Hareketlilik

Kullanıcılar için MANET'lerin en önemli avantajı hareketliliğidir. Diğer taraftan güvenilir yüksek başarımlı iletişim sağlamak amacı ile üstesinden gelinmesi gereken bir sorundur [1].

MANET protokollerindeki ek yük yığılması oluşmasında en fazla etkisi olan hareketliliğidir. Bu yüzden iletişimin verimliliğini düşürmektedir. Bununla beraber yakın geçmişte kablosuz Ad Hoc ağların kapasitesinin artırılmasında hareketliliğin etkisi olmuştur. MANET ağlar gezgin ve sabit düğümlerin karışımına sahip olabilir. Düğümlerin hareketliliği üç kategoride incelenebilir [61].

- Rastgele hareketlilik: Düğümlerin analitik uygunluk için keyfi olarak rastgele şablonlarda hareket ettiği (düzenli "Brown devinimi" olarak modellenen) kabul edilmektedir.
- Tahmin edilebilir hareketlilik (Predictable Mobility): Hareketli düğümlerin hareket kümeleri bilindiği varsayılabilir ve bu bilgi rota verisi için kullanılabilir. Hareketli düğümler veri transfer etmek amacıyla hareket etmezler ve bu yüzden yolları yönlendirme gereksinimleri ile çatışmaz.

- Kontrollü hareketlilik: Düğümlerin hareket örneği tamamen ağın kontrolü altındadır. Bu altyapı ağa mantıksal destek vermek gibi amaçlar için kullanılabilir.

4.5. MANET Yönlendirme Protokolleri

İletişim protokolü düğümler arası bilgi alış verişi için iyi tanımlanmış yapılar ve önceden belirlemiş kuralları veya eğilimleri ile etkin ve verimli iletişim sağlar. Yönlendirme protokolleri sözdizimsel (paket tipleri, formatları, vb.) ve anlamsal (bozuk paketler, vb.) bileşenlere sahiptir [1]. Ad Hoc ağlarda yönlendirme protokolünün başlıca amacı düğümler arasında doğru ve verimli yolun kurulması ve mesajların zamanında yerine ulaştırılmasıdır [62].

MANET’lerde kaynak ve hedef genelde doğrudan iletişim halinde değildir. Mesajlar ara düğümler yardımıyla iletilirler. Gerçekte çok atlamalı iletim MANET’lerin en önemli avantajıdır. Bu şekilde çok atlamalı olarak dolaylı iletişim yönlendirme olarak adlandırılır ve kaynak ile hedef arasındaki yolun belirlenmesi ve yönetiminden ağ katmanı sorumludur. Ağ katmanı uçtan uca (çok atlamalı) veri iletişimini sağlamak için, tek adımlı veri iletişimi kullanan MAC katmanından faydalanır [1].

MANET’lerde kaynakların sınırlılığı verimli ve güvenilir yönlendirme stratejilerinin tasarlanmasında büyük bir problem oluşturmaktadır. Zeki yönlendirme stratejileri ağ durumundaki değişikliklere adapte olurken (ağ büyüklüğü, trafik yoğunluğu, vb.) aynı zamanda sınırlı kaynakları verimli kullanmalıdır. Buna paralel olarak farklı kullanıcı ve uygulama tipleri için farklı QoS seviyelerini desteklemelidir. Kablosuz ağlara ilginin artmasından önce kablolu ağlarda iki temel algoritma kullanılmaktadır [26].

Geleneksel yönlendirme protokolleri bağlantı durumu (link-state) veya uzaklık vektörü (distance vector) algoritmalarına dayanır. Ağdaki her bir düğüm için en uygun rotayı ve ağdaki topoloji değişikliklerini tüm ağ boyunca yayılan mesajlarla

periyodik olarak güncelleyerek elde eder. Bu tür protokoller Ad Hoc ağlar için uygun değildir. Rotanın her bir düğüm için bulunması ve yönetimi hem masraflı (adım fazlalığı) hem de çoğunlukla gereksizdir. Bunun dışında Ad Hoc ağlarda sık değişen topolojiler nedeniyle düzenli güncelleme yeterli olmaz, çok fazla ulaştırılmayan paket ve kötü iletişime yol açar [26].

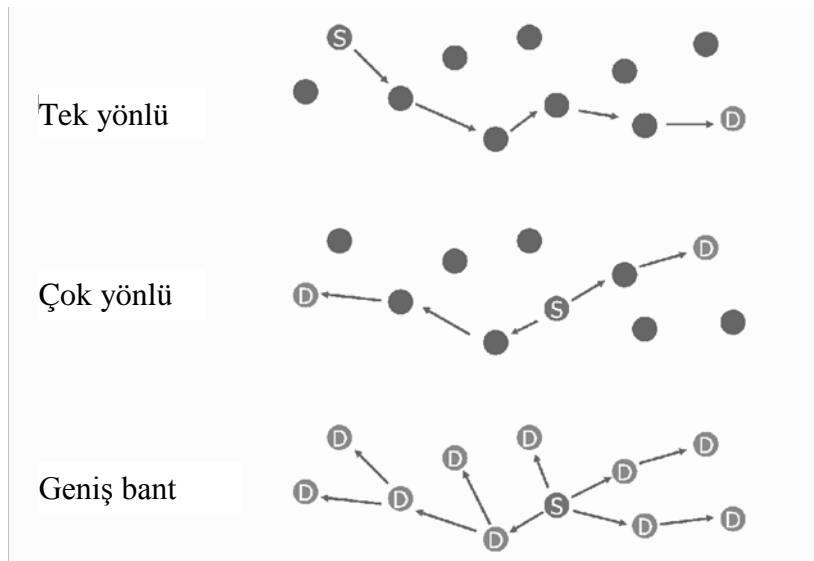
Bu problemlerin üstesinden gelmek için MANET çalışma gurubu 'Internet Engineering Task Force (IETF)' içinde Ad Hoc ağlarda IP tabanlı yönlendirme çerçevelerini geliştirmek için çalışmaktadır. Günümüzde gezgin Ad Hoc kablosuz ağlar için pek çok yönlendirme protokolü uzaklık vektörü (distance vector) ve bağlantı durum (link state) yönlendirme algoritmalarından geliştirilerek önerilmiştir [41] [26].

4.5.1. Yol bulma yöntemleri

Verimli yönlendirmeyi sağlamak için ağ katmanı iletişim ağının topolojisi hakkında yeterli bilgiye ihtiyaç duyar. Bunun için ağ katmanında iki temel fonksiyon vardır: yol keşfi ve yol yönetimi. Yol keşfinin hedefi, kaynak ve hedef arasındaki yolu belirli kriterlere göre bulmaktır (en kısa atlama, minimum maliyet). Yolun yönetimi ise var olan yolun korunmasıdır (hareketlilik nedeniyle topolojideki değişiklikler, vb.) [1].

Hedef sayısına göre 3 temel yönlendirme tipi vardır (Şekil 4.3) [1]:

- Tek yönlü; sadece tek hedef vardır.
- Çok yönlü; birden fazla hedef düğüm vardır.
- Geniş bant; Ağdaki tüm düğümler hedeftir.



Şekil 4.3. Hedef sayısına göre yönlendirme tipleri

4.5.2. MANET'lerin sınıflandırılması

MANET'lerde farklı bakış açılarına göre (yolun keşfi, karmaşıklık, türetildiği protokol, ölçeklenebilirlik, vb.) sınıflandırmak mümkündür. Yönlendirme protokollerinin sınıflandırması, bazen yapılan çalışmalar içinde ele alınmış bazense sadece sınıflandırmaya yönelik çalışmalar yapılmıştır [2] [28] [29] [30] [32] [38] [40] [62] [63] [64].

Hareketli Ad Hoc ağları anlamada en popüler metotlardan biri yönlendirme bilgisinin nasıl elde edildiği ve nasıl korunduğu / yönetildiğidir. Bu metodu kullanırken gezgin Ad Hoc ağ yönlendirme protokolleri; 'tabloya dayalı', 'isteğe bağlı' ve 'hibrit yönlendirme' olarak ayrılmaktadır [26] [27] [41] [42].

Tabloya dayalı, isteğe bağlı ve karma Ad Hoc yönlendirme protokollerine genel bir bakış Tablo 4.2'de görülmektedir [26].

Tabloya dayalı (table driven) yönlendirme protokolleri kullanırken gezgin Ad Hoc ağlardaki düğümler, ulaşılacak düğümleri, rotaların tutarlılığının sürdürülmesi ve yönlendirme bilgisinin güncelliği için düzenli olarak değerlendirir. Bu yüzden

kaynak bir rotaya ihtiyaç duyarsa hemen elde edebilir [27]. Güncelleme prosedürü bant genişliğine büyük yükler getirdiği için ölçeklenebilirliği zayıftır [26].

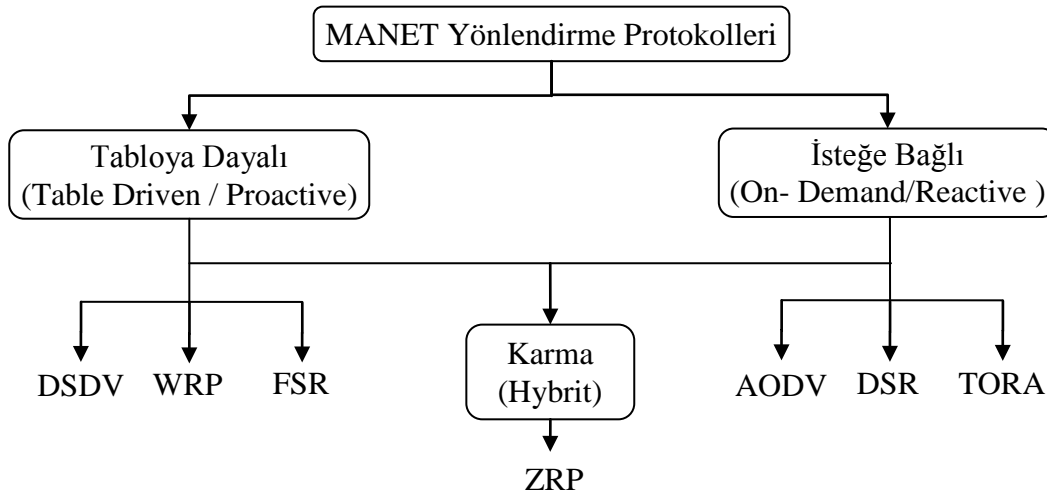
Tablo 4.2. Yönlendirme türlerinin karşılaştırılması

Yönlendirme sınıfı	Tabloya Dayalı	İsteğe bağlı	Karma
Yönlendirme yapısı	Düz ve hiyerarşik yapı olabilir	Genelde düz	Genelde hiyerarşik
Yolun erişilebilirliği	Düğüm aktif ise her zaman erişilebilir.	İhtiyaç duyulduğunda tanımlanır.	Hedefin konumuna bağlıdır
Kontrol trafik seviyesi	Genelde yüksek	Genel yönlendirmeye göre daha az	Genelde tabloya dayalı ve isteğe bağlı protokollerden az
Düzenli güncelleme	Evet. İstisnai olarak duruma bağlı olabilir.	Gereksiz. Bazı protokollerde isteğe bağlıdır	Geçitler arasında veya bir alan içinde
Hareketliliğin etkisi	Genelde belirli aralıklarla güncellenir. DREAM hareketliliğe bağlı olarak günceller	ABR LBQ tanıtır. ROAM eşik güncellemesi kullanır AODV yerel yol keşfi kullanır	Genellikle birden fazla yol mevcut olabilir. Hatalar grup çalışması ile azaltılır.
Kayıt gereksinimleri	Yüksek	İhtiyaç duyulan veya saklanması gereken yol sayısına göre değişir. Genelde tabloya dayalı protokollerden düşüktür	Eğer kümeler büyük ise her bir kümenin büyüklüğüne bağlı, her bir küme tabloya dayalı kadar büyük olabilir
Gecikme seviyesi	Küçük yollar önceden belirlenir	Tabloya dayalıya göre daha yüksek	Yerel (aynı alan veya küme) hedef için küçük kümeler arası isteğe bağlı protokoller kadar yüksek
Ölçeklenebilirlik seviyesi	Genelde 100 düğüm üzeri	Kaynak yönlendirme protokolleri birkaç yüz düğümden fazla. Noktadan noktaya daha fazla. Aynı zamanda trafik seviyesi ve atlama sayısına bağlı	1000 ve daha yüksek seviyeler için tasarlanmıştır

İsteğe bağlı yönlendirme protokolleri aynı zamanda “on-demand” olarak da adlandırılmaktadır. Bu tür protokollerde yollar sadece ihtiyaç duyulduğunda aranır. Rota keşif operasyonu rota-belirleme prosedürü ile talep edilir. Keşif prosedürü tüm yol permütasyonlarının denenmesinden sonra rota hazır değil veya rota bulundu durumlarından birisi gerçekleştiikten sonra sonlanır [27]. Genelde isteğe bağlı yönlendirme protokollerinin çoğu en kötü durum senaryosu dikkate alınırsa benzer yönlendirme maliyetine sahiptirler. Bu onların benzer, rota keşif ve sürdürme gibi doğal yapılarından kaynaklanmaktadır [26].

Karma yönlendirme protokolleri tabloya dayalı ve isteğe bağlı yönlendirme protokollerinin esaslarını bir araya getirilmesini ve eksikliklerinin üstesinden gelmeyi önermiştir. Normalde karma yönlendirme protokolleri gezgin Ad Hoc ağların hiyerarşik ağ yapılarının üstesinden gelmek içindir [26][27]. Karma yönlendirme protokolleri isteğe bağlı ve tabloya dayalı olanlara göre daha büyük ölçekli ağları destekleme potansiyeline sahiptir. Bu protokoller tanımlı yapıları sayesinde düğümlerin tekrar yayınlama sayılarını minimize ederler. Düğümlerin beraber çalışmasını mümkün kılar ve sırayla yönlendirmenin nasıl gerçekleştirileceğini organize ederler. Beraber çalışırken en iyi veya en uygun düğüm yol keşfini gerçekleştirir [26].

MANET'lerde kullanılan birçok yönlendirme protokolü vardır. Aşağıda bu protokollerden çok kullanılanları ve bu protokollerin sınıflandırılması görülmektedir (Şekil 4.4).



Şekil 4.4. MANET protokollerinde sınıflandırma

- Varış Sıralı Uzaklık Vektörü Protokolü (Destination Sequence Distance Vector - DSDV)
- Telsiz Yönlendirme Protokolü (Wireless Routing Protocol - WRP)
- Balıkgözü Durum Yönlendirme Protokolü (Fisheye State Routing - FSR)
- Bölge Yönlendirme Protokolü (Zone Routing Protocol - ZRP)

- Ad Hoc İsteğe Bağlı Uzaklık Vektörü Yönlendirme Protokolü (Ad hoc On-demand Distance Vector Routing - AODV)
- Değişken Kaynak Yönlendirme Protokolü (Dynamic Source Routing - DSR)
- Geçici Sıralı Yol Atama Protokolü (Temporally Ordered Routing Algorithm - TORA)

4.6. AD HOC İsteğe Bağlı Uzaklık Vektörü Yönlendirme Protokolü (AODV)

AODV (Ad Hoc On–Demand Distance Vector Routing) yönlendirme protokolü C. Perkins (Nokia Araştırma Merkezi), E. Belding-Royer (Kaliforniya Üniversitesi, Santa Barbara) ve C. S. Das (Cincinnati Üniversitesi) tarafından geliştirildi [65]. AODV yönlendirme algoritması dinamik, kendi kendine başlayan (self starting), çok atlamalı gezgin düğümler arası Ad Hoc ağların yönetimi için kullanılmaktadır. AODV mobil düğümlerin yeni hedefler için hızlı yol kurulmasını sağlar ve hedef düğümlerle rotanın yönetimi için aktif bir iletişim altyapısına sahip olmak zorunda değildir. AODV zamanla ağ topolojisindeki oluşan değişikliklere ve bağlantı kopmalarına düğümlerin cevap vermesine imkân sağlar. AODV kısır döngü (loop–free) ve Ad Hoc ağ topolojisinde değişiklik olduğu zaman oluşabilecek Bellmand-Ford sonsuza sayma (counting to infinity) problemlerinin üstesinden gelebilir.

AODV'nin tasarlanma sebepleri:

- AODV yönlendirme protokolü küçük veya büyük ölçekli gezgin Ad Hoc ağlar için tasarlanmıştır.
- AODV veri trafiğindeki düşük orta ve yüksek hareketliliği desteklemektedir.
- AODV düğümlerin birbirine güvendiği ağlarda kullanmak için tasarlanmıştır.
- AODV ölçeklenebilirlik ve başarımla kontrol trafiğinin azaltılması ve veri trafiğindeki paketlerin fazlalıkların elimine edilmesi için tasarlanmıştır.

AODV'nin ayırt edici özelliklerinden biri her bir yol için hedef sıra numarası kullanmasıdır. Hedef sıra numarası hedef tarafından isteği gönderen düğüm için üretilir. Hedef sıra numarasının kullanılması ile sonsuz döngü engellenmiş olur. Bir

düğüm istekte bulunacağı zaman elinde bulunan en büyük hedef sıra numarasını kullanır [65][42].

4.6.1. AODV yönlendirme protokolü terminolojisi

AODV’de kullanılan terminoloji aşağıda tanımlanmıştır

Aktif yol; Yönlendirme tablosunda geçerli olarak işaretlenmiş hedef rotasıdır. Sadece aktif yol veri paketlerinin iletilmesi için kullanılabilir.

Broadcast; IP sınırlı yayın 255.255.255.255. Yayınlanan paketler rastgele iletilmez fakat AODV mesajlarının Ad Hoc ağ boyunca iletilmesinde kullanışlıdır.

Hedef düğüm; Veri paketlerinin iletildiği düğümdür. Düğümler hedef olduklarını AODV protokolü tarafından sağlanan yol keşfinde kullanılan mesajlardan ve tipik veri paketinde IP başlığında belirlenen alandaki adresten öğrenir.

İleten düğüm; bir sonraki düğüm için paketleri kabul edip ileten düğümdür. Yönlendirme kontrol mesajları ile kurulur.

İleri yol; İstenen hedefe veri paketlerinin gönderilmesi için kurulan rotadır.

Geçersiz yol; Rota sona erdiğinde yönlendirme tablosunda işaretlenir. Önceki geçerli rota bilgisi uzatılmış zaman periyodu için saklanır. Geçersiz rota veri paketlerinin iletilmesi için kullanılır, fakat rota onarımları için ve gelecekte kullanılacak RREQ (Route request) mesajları için destek bilgi olarak kullanılır.

Kaynak düğüm; AODV yol keşfi başlatan ve muhtemel geri yayınlama yapan düğümlerdir. Örneğin bir düğüm yol keşfi başlattığında RREQ mesajı yayar böylece RREQ mesajının kaynak düğümü olur.

Tersine yol; Hedef düğüm veya hedefe ait rota bilgisine sahip olan ara düğüm tarafından kaynak düğüme gönderilen cevap RREP (Route Replay) paketinin izlediği yoldur.

Sıra numarası; Üreten düğüm tarafından yönetilen düzenli olarak artırılan sayıdır. Böylece AODV yönlendirme protokolünde mesajlar diğer düğümler tarafından üreten düğüme ait bilginin tazeliğinin doğruluğu için kullanılır.

Geçerli yol; Yönlendirme tablosunda geçerli olarak işaretlenmiş hedef rotasıdır. Aktif yol terimi ile aynı tanıma sahiptir

4.6.2. Yönlendime tablosu ve öngösterge listesi

AODV bir yönlendirme protokolüdür ve yönlendirme tablosu kullanır. Yönlendirme tablosu kısa süreli olarak rotaları saklar. AODV yönlendirme tabloları aşağıdaki bilgileri tutar.

- Hedef IP Adresi
- Hedef sıra numarası
- Geçerli hedef sıra numarası bayrağı
- Diğer durumlar ve yönlendirme bayrakları (geçerli, geçersiz, onarılabilir, onarıldı vb.)
- Ağ arayüzü
- Adım sayısı (hedefe ulaşmak için gerekli hop sayısı)
- Sonraki adım
- Ön gösterge listesi
- Yaşam süresi (yolun son veya silme süresi)

Her bir düğüm hedef düğüm IP adresi için ulaşılmış son sıra numarasını içermek zorundadır. Bu sıra numarası “hedef sıra numarası” olarak adlandırılmaktadır. Düğüm RREQ, RREP ve RERR mesajlarından herhangi birini aldığı anda yönlendirme tablosunda ilgili hedefin sıra numarasını günceller. AODV sonsuz

döngüye sebep olmamak için her bir düğümün kendi sıra numarasına sahip olması ve bu sıra numarasının yönetimine dayanır. Düğüm sıra numarasını iki durumda artırır.

- Rota keşfi için mesaj üretmeden önce kendi sıra numarasını artırır. Bu daha önce kurulmuş rotalar ile çakışmayı engeller.
- RREQ mesajına cevap olarak RREP göndermeden hemen önce artırır.

Düğüm, komşularından bir AODV mesajı aldığı zaman veya hedef için rota oluşturduğu / güncellediği zaman, yönlendirme tablosunu hedef düğüm için kontrol eder. Hedefe karşılık gelen giriş yok ise oluşturulur. Sıra numarası kontrol paketinin sıra numarası alanı “0” değilse bu alandan elde edilir.

Yönlendirme tablosundaki sıra numarası sadece aşağıdaki durumlarda değiştirilir / güncellenir:

1. Hedef sıra numarası yönlendirme tablosundakinden büyük ise,
2. Sıra numarası eşit fakat yeni bilginin atlama sayısı (hop count) küçük ise,
3. Sıra numarası yok ise.

Düğüm tarafından yönetilmiş her geçerli yol için “ön gösterge” listesi yönetilir. Bu yol mesajların iletiildiği bir yol üzerinde olabilir. Bu liste düğümden alınan uyarılardır (bir sonraki düğüm linkinin kaybolup olmadığı ve RREP'nin gönderilip gönderilmediği vb).

4.6.3. AODV çalışma süreci ve mesaj paketlerinin hareketi

Yol isteği (Route Request - RREQ), Yol cevabı (Route Reply - RREP), Yol hata (Route Error - RERR) AODV için tanımlanmış mesaj tipleridir ve bu mesaj tipleri UDP vasıtasıyla normal IP başlığı işlemine tabi tutularak iletilir. Mesajın yayın alanı IP başlığındaki TTL ile belirlenir. Bu sayede mesajın tüm ağ boyunca yayını sağlanabilir.

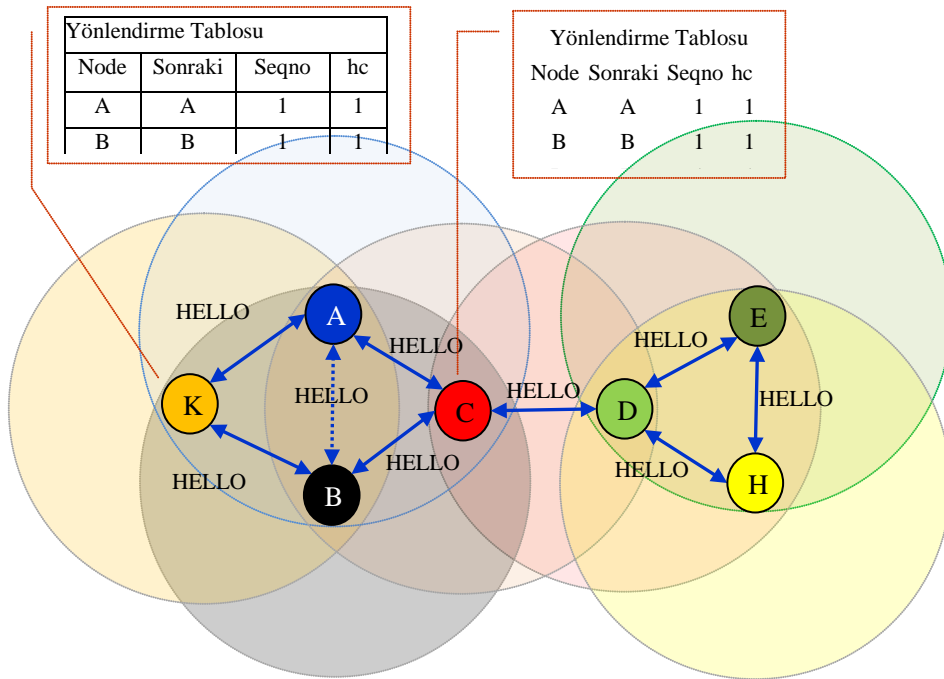
Aşağıda yedi düğümden oluşan örnek bir ağ üzerinden AODV yönlendirme algoritmasının çalışması anlatılmaktadır.

4.6.3.1. HELLO paketi

Düğüm bağlanabilirlik bilgisini HELLO mesajları yayarak elde edebilir. Şekil 4.5’de HELLO mesaj paketi oluşturmada kullanılan değerler, Şekil 4.6’da ise örnek bir ağ üzerinde HELLO paketlerinden sonra düğümlerin yönlendirme tablosundan örnekler görülmektedir.

Hedef IP adresi (Düğümün IP adresi)
Hedef Sıra Numarası (Düğümün en son sıra numarası)
Hop Sayısı (0)
Yaşam Süresi ($ALLOWED_HELLO_LOSS * HELLO_INTERVAL$)

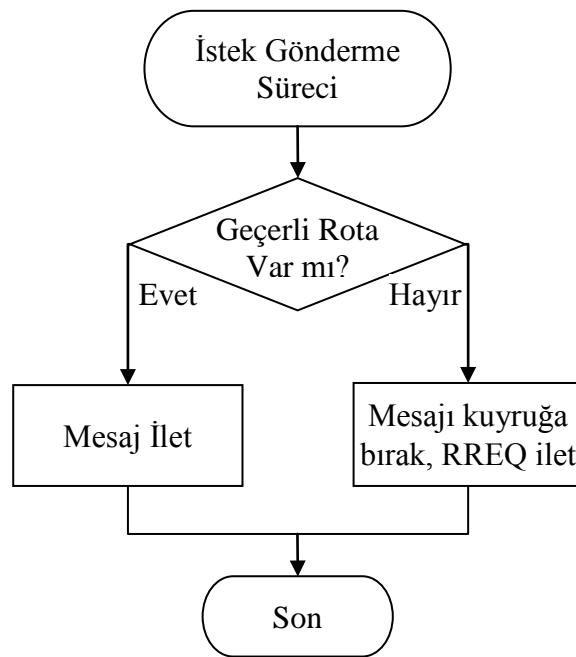
Şekil 4.5. HELLO mesaj paketi bileşenleri



Şekil 4.6. HELLO mesajları sonrası yönlendirme tabloları

4.6.3.2. RREQ Mesajı

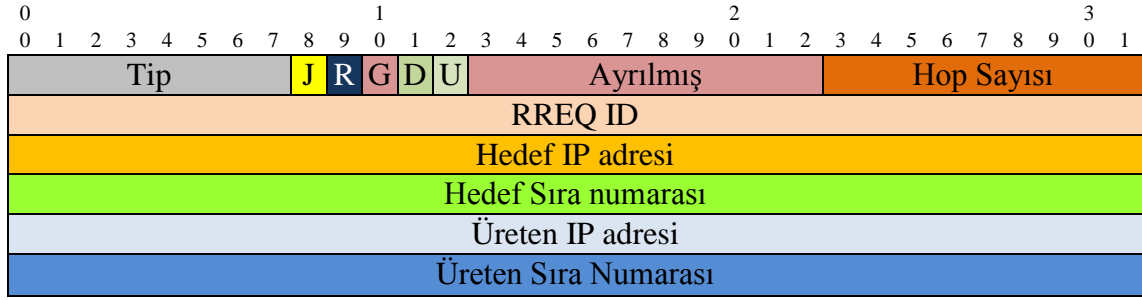
Bir düğüm bir hedef için rota bilgisine ihtiyaç duyduğu zaman RREQ (Route Request) mesajı yayınlar (broadcast). Hedef veya hedefe ait yeterince güncel bilgiye sahip düğüm tarafından gönderilen (unicast) mesajın alınmasıyla hedefe ait rota elde edilir. Bu iletişim sırasında, ara düğümler aldıkları mesajlardan kendi yönlendirme tablolarını güncellerler [65]. Bir veri gönderme isteği olduğunda düğümün sergilediği davranış Şekil 4.7'deki akış şemasında gösterilmektedir.



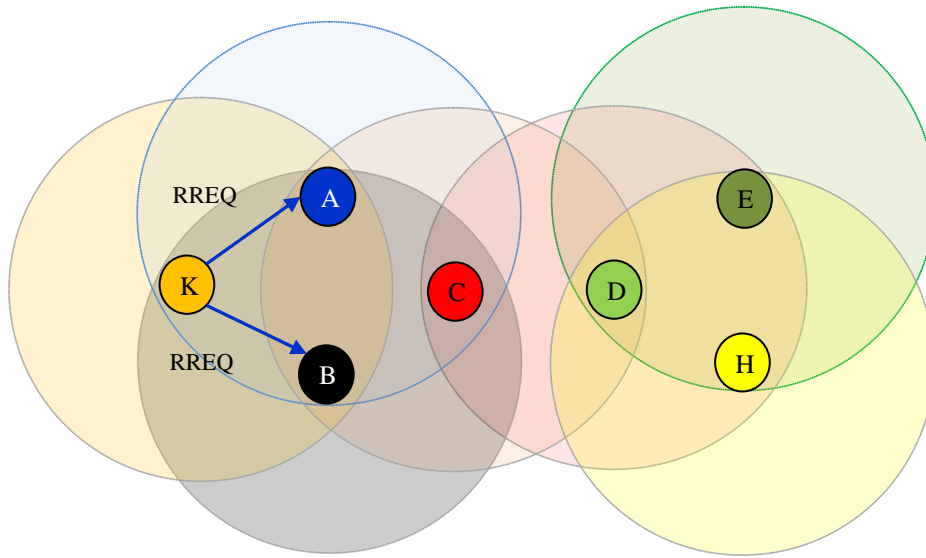
Şekil 4.7. AODV mesaj gönderme süreci

Düğüm geçerli bir rotaya sahip olmadığı zaman rota kurmak üzere Şekil 4.8'de gösterilen biçimde bir RREQ mesajı yayınlar.

Kaynak düğüm (K) hedef düğüme (H) bir veri göndermek istemektedir. Yönlendirme tablosunda H düğümü için bir rota bilgisi mevcut değildir. Bu nedenle RREQ mesajı yayınlar (Şekil 4.9).

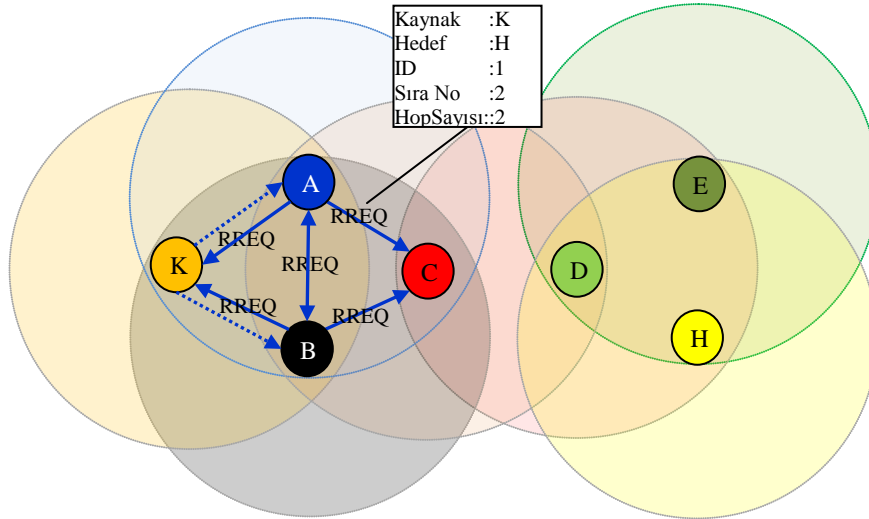


Şekil 4.8. RREQ mesaj paketi içeriği

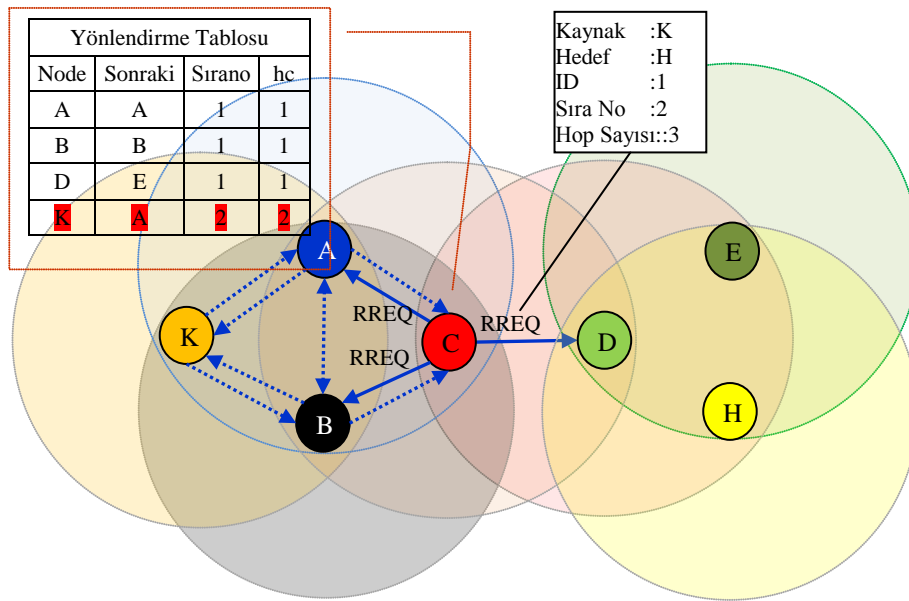


Şekil 4.9. RREQ mesaj iletimi

Mesajı alan A ve B düğümleri öncelikle K düğümüne ait rota bilgilerini güncellerler. Bir düğüm RREQ aldığı zaman iki seçeneği vardır: (1) Eğer hedef kendisi ise veya hedefe ait geçerli bir rota bilgisine sahip ise RREP yayınlar, (2) değil ise RREQ mesajını iletir. Burada A ve B düğümleri H düğümüne ait bir geçerli rota bilgisine sahip olmadıkları için RREQ mesajını iletirler (Şekil 4.10 ve Şekil 4.11). RREQ mesajında bulunan RREQ_ID sayesinde düğümler kendilerine ulaşan mesajlarından sadece biri için işlem yaparlar. Böylece sonsuz döngüler engellenmiş olur.



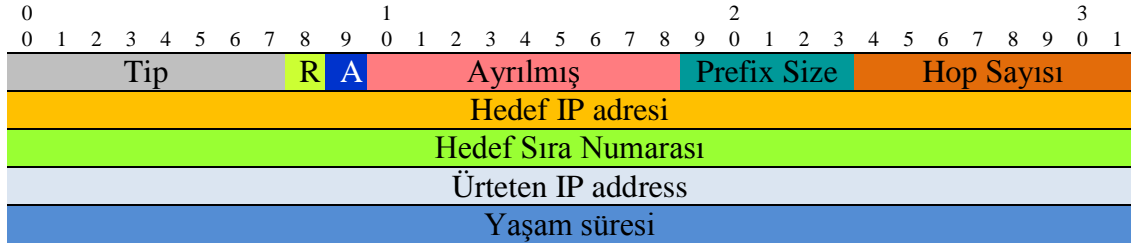
Şekil 4.10. RREQ mesaj iletimi



Şekil 4.11. RREQ mesaj iletimi

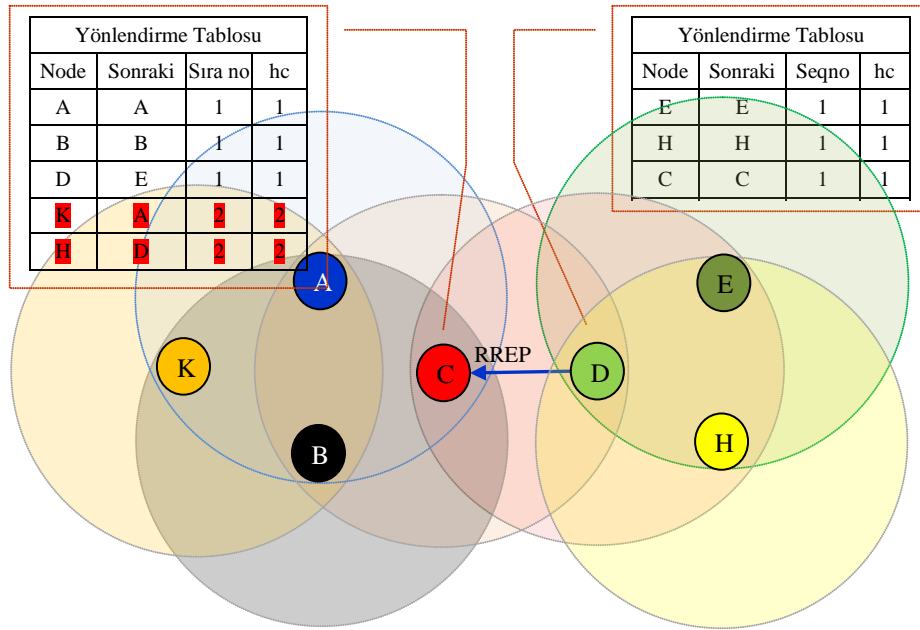
4.6.3.3. RREP mesajı

Bir düğüm kendisine gelen RREQ mesajının hedefi ise veya RREQ mesajının hedefine ait geçerli bir rotaya sahip ise, hedef için düğümde var olan yönlendirme tablosundaki hedef sıra numarası geçerli ve RREQ mesajındaki hedef sıra numarası büyük veya eşit ise Şekil 4.12'deki gibi bir RREP mesajı yayınlar.

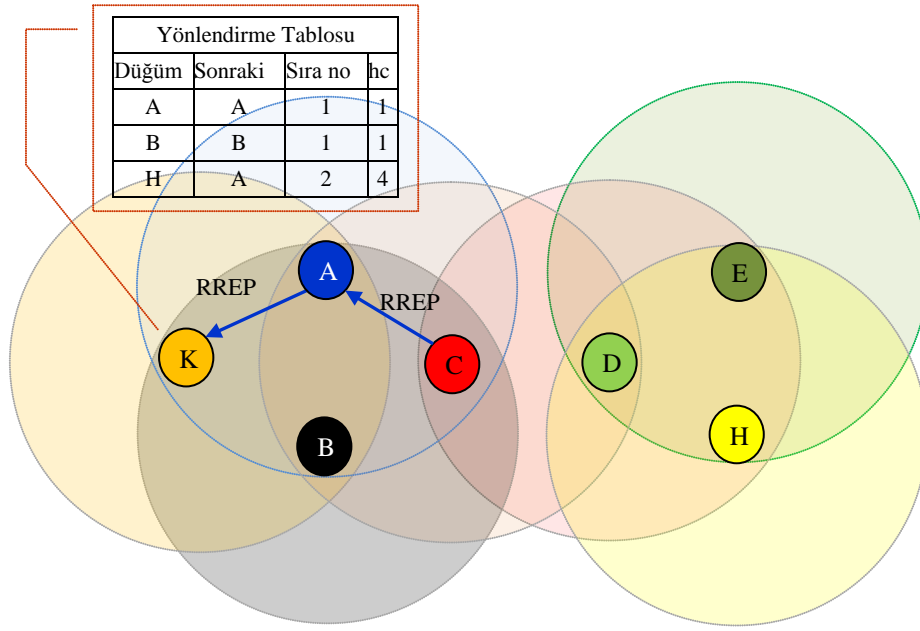


Şekil 4.12. RREP mesaj bileşenleri

Düğüm RREP üretirken, hedef IP adresini, RREQ mesajını üretenin IP'sini alarak RREP mesajında karşılık gelen alanlara koyar. Hedef kendisi ise yönlendirme tablosunda mesajı gönderene ait bilgileri günceller ve bir sonraki düğüme adım sayısını artırarak gönderir. Böylece üreten hedefin kaç adım uzaklıkta olduğunu öğrenir. Eğer hedef kendisi ise kendi sıra numarasını artırmalı aksi halde hedef RREP üretmeden sıra numarasını değiştirmez.



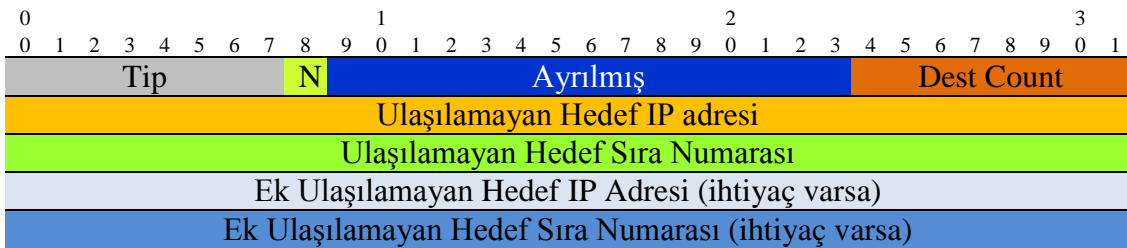
Şekil 4.13. RREP mesaj iletimi



Şekil 4.14. RREP mesaj iletimi

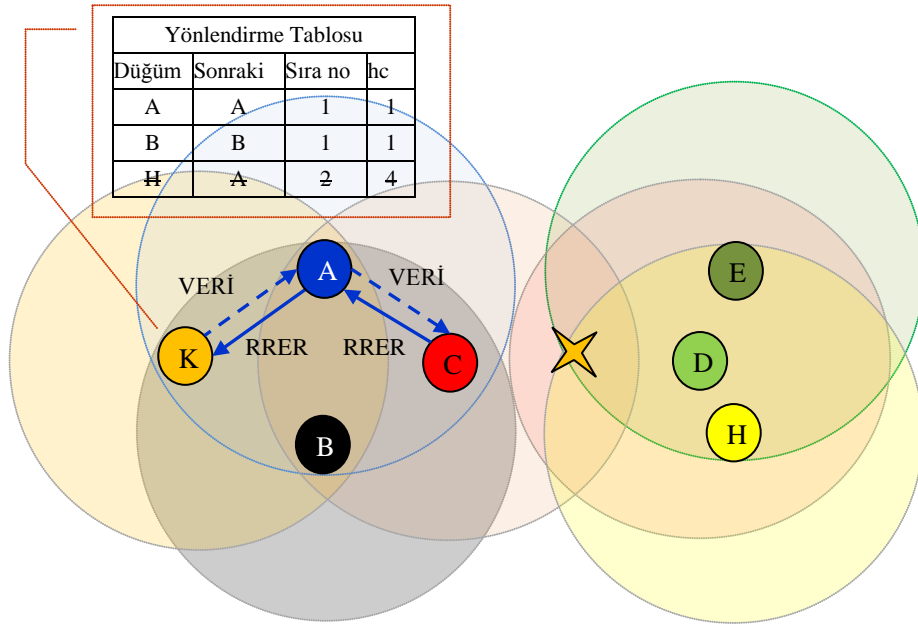
4.6.3.4. RRER mesajı

Düğümler bir sonraki düğümün bağlantı durumunu takip ederler. Aktif bir rotanın bağlantısında bir kırılma olduğu zaman RRER mesajı yayınlarlar (Şekil 4.15). Bu mesaj belirtilen hedefin artık ulaşılamaz olduğunu belirtir. Bu mekanizmayı mümkün kılmak amacıyla her bir düğüm “ön işaretçi” listesi tutar.



Şekil 4.15. RRER mesaj içeriği

Şekil 4.16’da veri paketi iletimi sırasında C ve D düğümleri arasında bağlantının kopması nedeniyle C düğümü RRER mesajı yayınlar mesajı alan düğümler rota bilgisini günceller.



Şekil 4.16. RRER mesaj iletimi

BÖLÜM 5. MANET - DEVS MODELLEME VE BENZETİM ORTAMININ GELİŞTİRİLMESİ

5.1. Giriş

Bu bölümde, MANET ağların problemlerine çözüm getirmek amacıyla Bölüm 3’ de detayı verilen DEVS [10][59] modelleme ve benzetim yaklaşımı kullanılarak, gelişmiş kullanıcı dostu arabirimine sahip, tasarım, modelleme, benzetim ve eğitim amaçlı kullanılabilen DEVS-Suite ortamı [15] üzerinde geliştirilen bir MANET modelleme ve benzetim çerçevesinin gelişim safhaları ele alınacaktır. Sistemi oluşturan bileşenler tanımlanarak, tanımı yapılan bileşenler ve bu bileşenlerin davranışları detaylandırılacaktır.

Tipik bir MANET sisteminde gezgin düğümler, düğümlerin iletişimini sağlayan bağlantılar ve destek birimleri bulunmaktadır. Sistem bileşenlerinin davranışlarının modellenmesinden sonra, MANET ağların analizi ve yönlendirme algoritmalarının test edilmesi uygulamalarını gerçekleştirebilmek amacıyla, bileşenler, Java programlama dili kullanılarak kodlanmıştır. Daha sonra geliştirilen model ile deney yapabilmek için Bölüm 2’de detaylandırılan deneysel çerçeve geliştirilmiştir. DEVS birleşik model tanımı kullanılarak, DEVS-Suite modelleme ve benzetim ortamı altında gerçekleştirilen ortam “MANET-DEVS” olarak isimlendirilmiştir.

MANET gibi dinamik, paralel ve dağıtık bir sistemi DEVS-Suite ortamında gerçeklemek; nesneye-yönelik modellemeyi, eş zamanlı paralel çalışan benzetimleri, etkileşen benzetim nesneleri arasında tam uyumluluğu, web tabanlı benzetimleri, model üzerinde detaylı testler yapabilmeyi ve eksiksiz bir model takibini olanaklı kılmaktadır.

MANET-DEVS ortamının, modellenme ve benzetim aşamalarında oluşturulan bileşenler, nesnelere, kavramlar ile ilgili teorik bilgi ve DEVS-Suite ortamından elde edilen ekran çıktıları bölüm içerisinde detaylandırılmaktadır.

5.2. Java Programlama Dili ve DEVS-Suite ortamı

Java programlama dili C ve C++' programlama dilleri ile benzer söz dizimine sahip nesneye dayalı bir programlama dilidir. Java dilinin taşınabilirliği ve web sayfalarındaki Java appletleri nedeniyle sık sık 'World Wide Web' ile birlikte anılmaktadır [66]. Java programlama dili C++ programlama dilinden daha yüksek başarıma sahiptir. [67].

Java programlama dili, Java sanal makinesi (Java Virtual Machine –JVM) ve java programlama dili ile ilişkili sınıf kütüphaneleri yardımıyla 3 farklı alanda taşınabilirlik özelliği sunar [68] . Bunlar;

- Kaynak kodu taşınabilirliği: Bir Java programı, temel CPU, işletim sistemi veya Java derleyicisi dikkate alınmaksızın aynı sonuçları üretmelidir.
- CPU mimarisi taşınabilirliği: JVM CPU mimarisi taşınabilirliğini sağlar. Java derleyicisi, JVM için nesne kodunu (J-code) üretir ve belirli bir CPU üzerinde JVM ilgili CPU için J-kodunu derler.
- OS/GUI taşınabilirliği: OS/GUI taşınabilirliğini tesis etmek için, Java sanal OS ve GUI için bazı paketler (awt, util ve lang) sağlar.

Java tabanlı ayrık olaylı sistem modelleme araçlarının kullanımı gittikçe artmaktadır [69][70][71]. Basitlik, %100 nesneye yönelim, hız, dağıtık çalışabilme, taşınabilirlik, vb. pek çok faydadan dolayı modelleyiciler tarafından tercih edilmektedir [71].

DEVS-Suite modelleme ve benzetim ortamı paralel DEVS formalizmine dayanır ve modüller, hiyerarşik, ayrık olaylı sistem ortamı, nesneye dayalı Java programlama dili

kullanılarak gerçekleştirilir [72][70]. MANET-DEVS benzetim ortamının modellenmesi ve benzetimi DEVS-Suite ortamında Java'nın imkânlarından faydalanılarak hazırlanmıştır.

5.3. MANET-DEVS Modelleme Süreci

Dağıtık ağ sistemlerinin tasarım ve analizini sistematik bir şekilde gerçekleştirmek için bu sistemlerin modellenmesi ve benzetimi beş aşamaya bölünebilir:

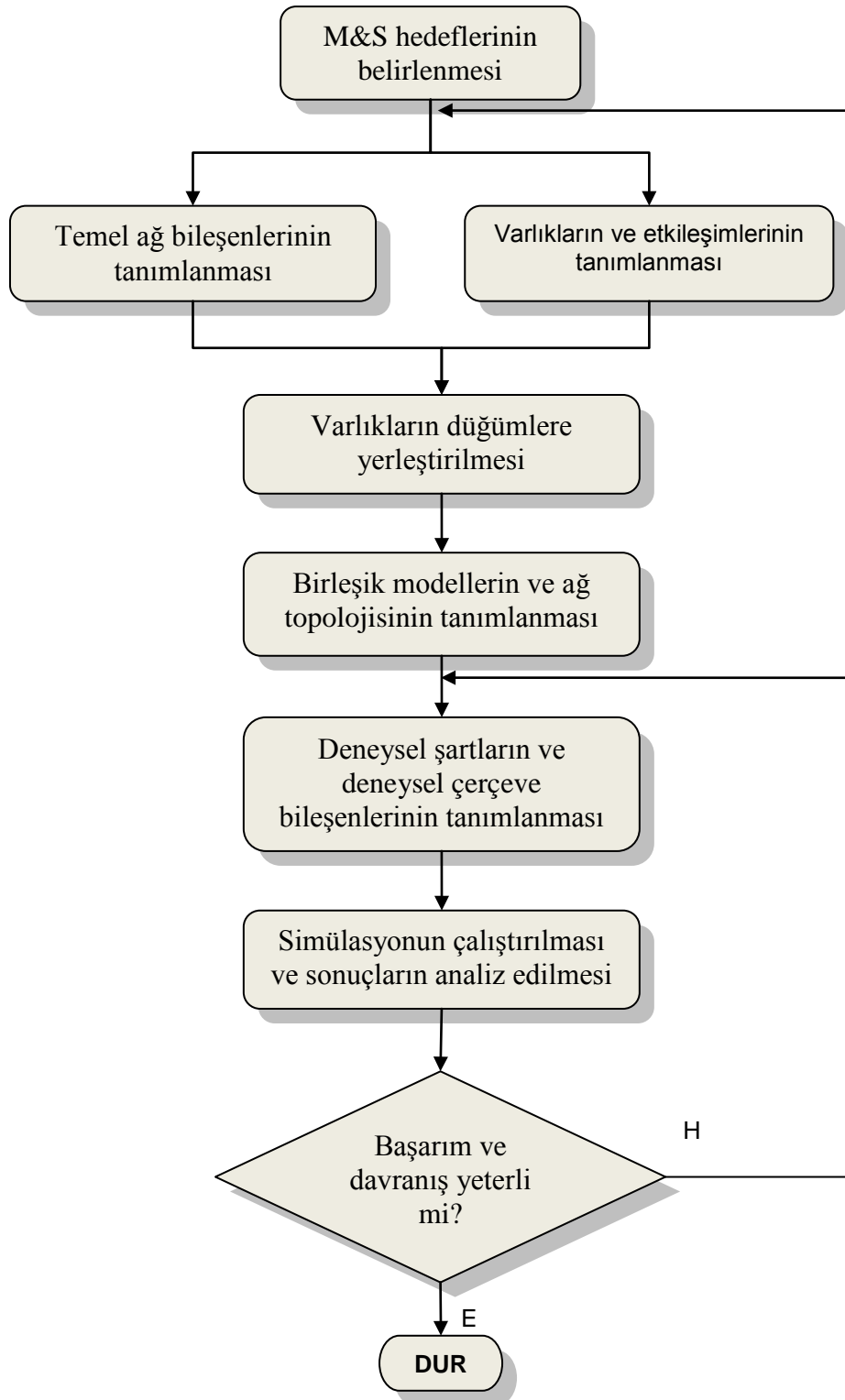
İlk aşama, modelleme ve benzetim hedeflerinin belirlenmesidir (Şekil 5.1). Modelleme hedefleri, modelin sistemlerin tasarımında, yönetiminde ve kontrolünde tanımlanan rollerle ilişkilidir. Hedefler ifadesi, model tasarlama işleminin belirli problemler üzerine odaklanması vazifesini görür. Önceden hedefler bilinirse, bu hedeflere uygun deneysel çerçeveler geliştirilebilir.

İkinci aşama, benzetim bileşenlerinin tanımlanması işlemidir. Bu aşamada benzetimi oluşturan bütün bileşenler ve bu işlemlere ilişkin parametreler tanımlanır. Bu aşamada düğümlerin ve varlıkların ayrı ayrı ele alındığına dikkat edilmelidir.

Üçüncü aşama, geliştirilen varlıkların ve düğümlerin birleştirilmesi aşamasıdır. Bu bileşenler DEVS-Suite modelleme ve benzetim ortamında birbiriyle bağlanarak değişik topolojiler ve ağ konfigürasyonları meydana getirilebilir.

Dördüncü aşama, modelleme ve benzetim hedeflerine ve geliştirilen modellere uygun bir deneysel çerçeve tanımlanması işlemidir. Bu deneysel çerçeve modelin bir takım şartlar altında test edilmesine ve gözlem yapılmasına olanak tanır.

Son aşamada, benzetim çalıştırılır ve sonuçlar DEVS-Suite ortamında gözlemlenir. Sonuçlar kabul edilebilir bir aralıkta ise benzetim işlemi sonlandırılır. Aksi halde, tekrar başa dönülür ve geliştirilen modellerin parametreleri ayarlanır. Böyle bir süreçte ileri geri hareket edilerek en uygun model geliştirilmiş olur.



Şekil 5.1. MANET'in modelleme ve benzetim süreci

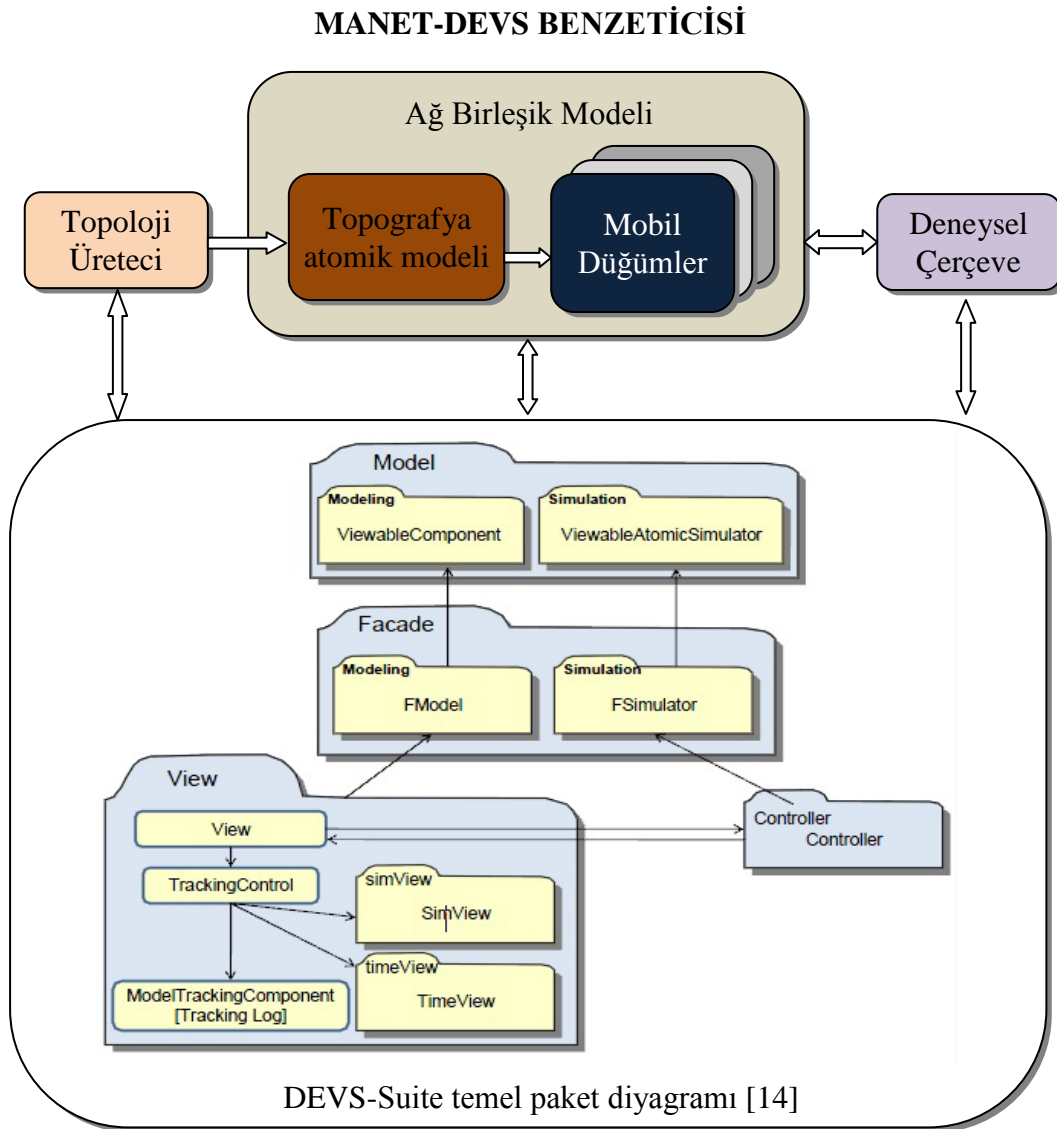
5.4. MANET Modelleme Yaklaşımı ve Ağ Oluşturan Bileşenlerin Tasarımı

Bir MANET ağ topolojisi, gezgin düğümlerden (bilgisayar, cep telefonu, vb.) oluşur. Düğümler, ağ üzerinde dolaşan varlıklar için bir hesaplama altyapısı oluşturması yanında diğer düğümlerle mesajlaşma ve veri trafiği için bir takım arabirimlere ve araçlara sahiptir.

MANET sistemini modelleme amacına yönelik oluşturulan düğümler, bu düğümlerin haberleşmesini sağlayan diğer nesnelere (IP paketleri vb.), ‘temel ağ bileşenleri’ olarak tanımlanır [63]. Tanımlanan düğümler ve dinamik bağlantılar ile çeşitli ağ konfigürasyonları ve topolojileri geliştirilebilir. Daha sonra bu bileşenlerin bir araya gelmesinden oluşan bir ağ modeli ‘DEVS birleşik ağ modeli’ olarak adlandırılır. Burada kullanılan atomik ve birleşik modeller, Bölüm 3’te tanımlanan ‘Paralel DEVS yaklaşımı’ kullanılarak tanımlanır ve ‘DEVS-Suite’ modelleme ve benzetim ortamında tasarlanılır. Bu yöntemde, düğümler ve diğer ağ bileşenleri, ağ modelinin davranışını (çıkış zamanı gibi) belirlemek için kullanılabilir.

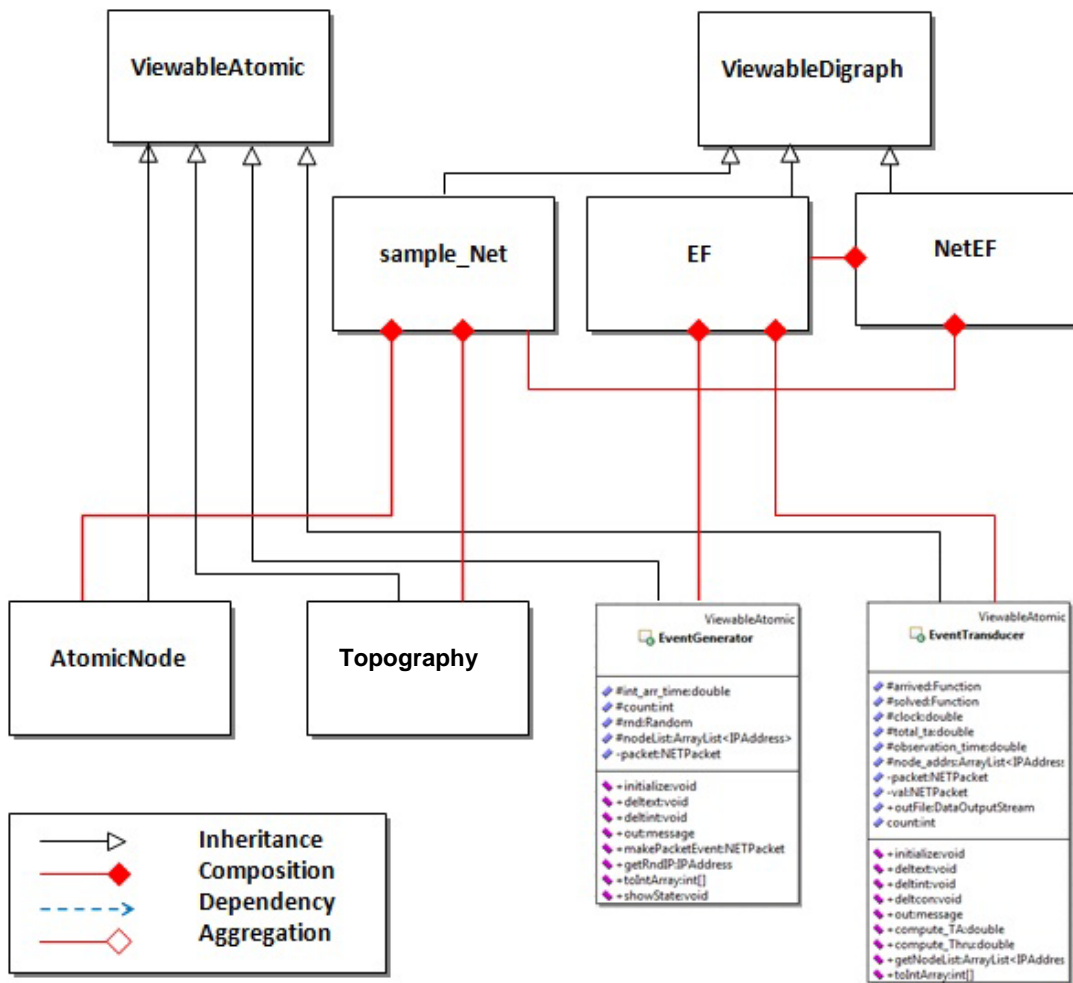
Yapılan çalışmada geliştirilen modellerle benzetim deneyleri gerçekleştirmek için, bir modelin deney yapılacağı ve gözlemleneceği şartları tanımlayan ‘DEVS deneysel çerçeve’ kavramından faydalanıldı. Bölüm 2’de detaylı olarak ifade edildiği gibi, tipik bir deneysel çerçeve bir üreteç ve bir dönüştürücüden oluşur. Ağ trafiğini üretmek ve düğüm gibi ağ bileşenlerinin arızalanmaları ve hizmet verememeleri gibi ağın çalışması sırasında meydana gelebilecek muhtemel olayları önceden programlamak ve zamanlamak maksadıyla ağdaki ilgili bileşenlere DEVS mesajları gönderen bir ‘olay üretici’ kullanıldı.

Geliştirilen MANET-DEVS benzeticisinin kavramsal modeli en temel haliyle Şekil 5.2’de görülmektedir. Şekilde görüldüğü gibi MANET-DEVS benzeticisindeki bileşenler DEVS çekirdeği ile beraber çalışmaktadır.



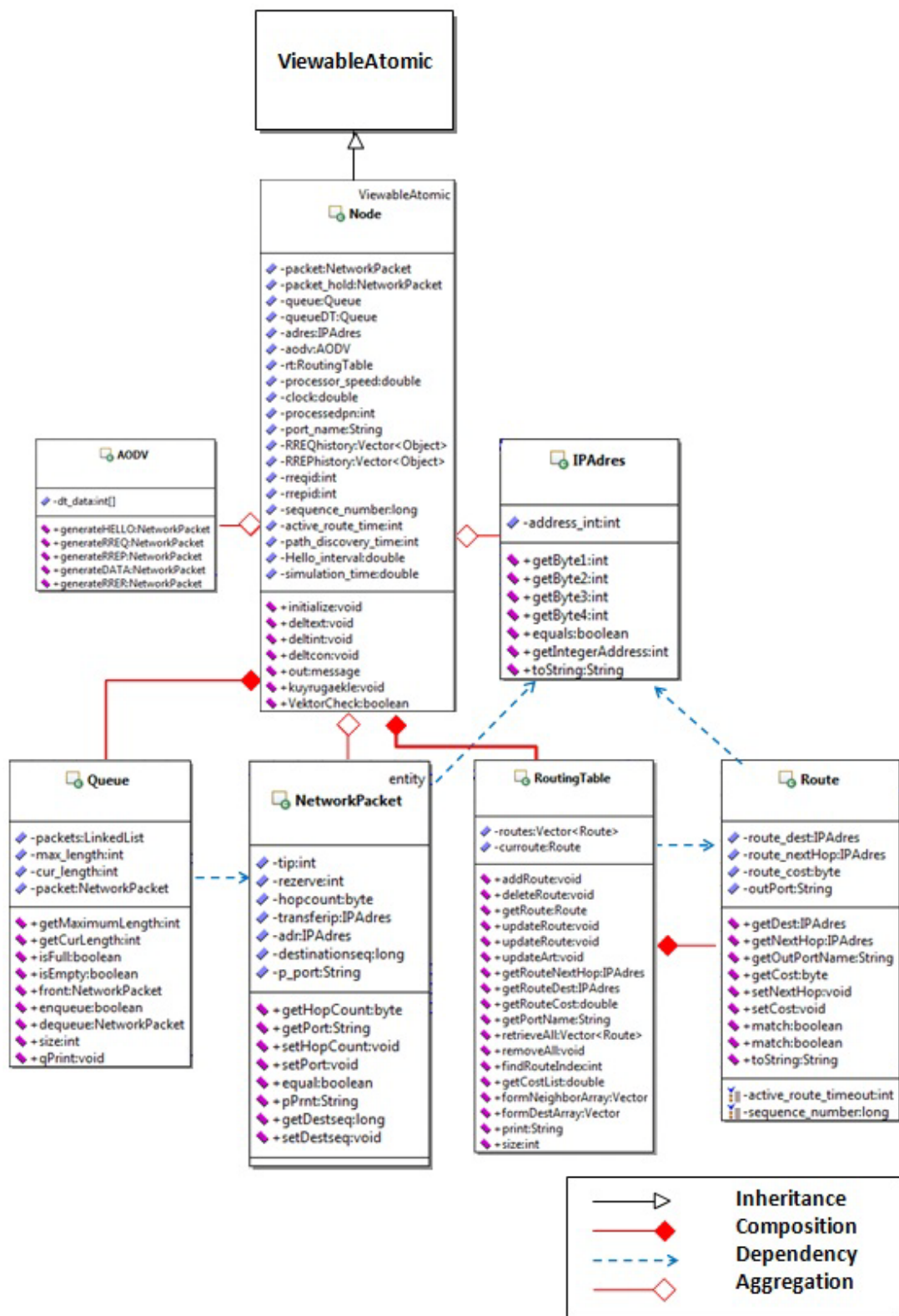
Şekil 5.2. DEVS-Suite ve MANET-DEVS benzetici kavramsal modeli gösterimi

Nesne yönelimli analiz, tasarım ve programlamada sistemin statik yapısı sınıf diyagramları ile modellenir. Sınıfları en iyi şekilde temsil etmeye yarayan sınıf diyagramları, UML'nin en çok kullanılan diyagram türlerinden biridir. Sınıf diyagramları gerçek örneklere göre oluşturulan sınıflar, sınıflara ait özellikler ve sınıfların davranışlarının çizilmesi ile oluşturulur. Hazırlanan sınıflar arasında ilişkiler olabilir, bu ilişkiler ilişkinin türünü, şeklini temsil eden çizgi ve sembollerle belirtilir.



Şekil 5.3. MANET-DEVS benzetim ortamı sınıf diyagramı

Sınıf diyagramlarında, içi boş ok işareti ile bir sınıfın başka bir sınıftan türetilmesi (inheritance) temsil edilmektedir. Birden fazla parçadan oluşan sınıflar arasında ‘sahiptir / içerir’ türünden bir ilişki var ise (Aggregation); içerim bağlantısı, içeren sınıf tarafına yerleştirilen içi boş eşkenar dörtgen ile gösterilmektedir. İçi dolu eşkenar dörtgen ise, bir sınıfın başka bir sınıftan ibaret olması (oluşum -composition) temsil edilmektedir. ‘Oluşum’, ‘içerir’ türünün daha güçlü bir halidir. Oluşumda parçalar bütün olmadan olmazlar Oluşturulan sınıf başka bir sınıfı kullanıyor ise bu durumda sınıflar arası ilişki kesikli çizgi ile gösterilmektedir. Bu bilgiler ışığında MANET-DEVS ortamı için hazırlanan sınıf diyagramları Şekil 5.3 ve 5.4’de görülmektedir.



Şekil 5.4. Atomik düğüm sınıf diyagramı.

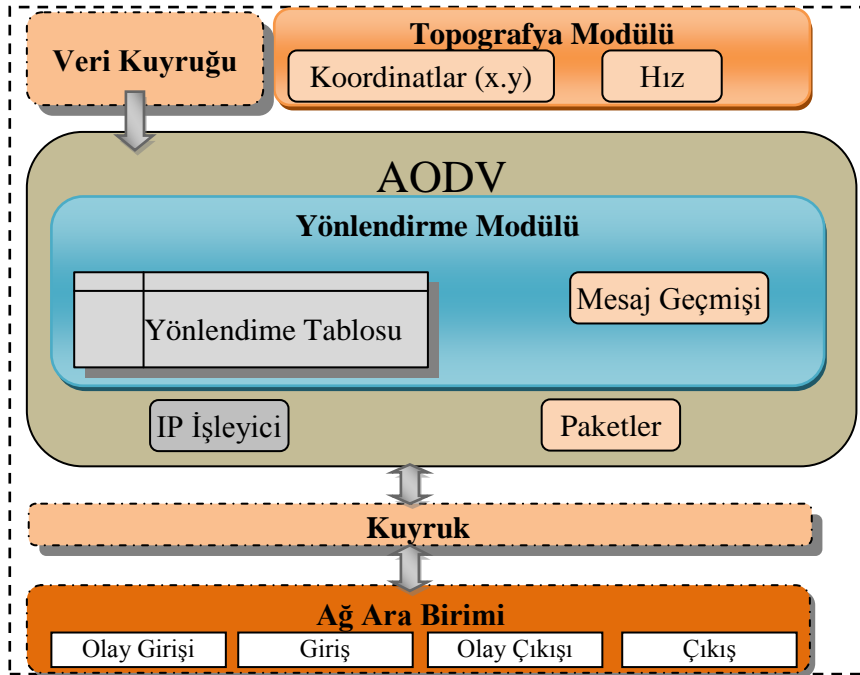
5.4.1. Dügüm atomik modeli

Ağdaki her bir düğüm, aşağıda yapısı detaylı bir şekilde tanımlanacak olan paketleri işleyebilme ve paketleri uygun hedeflere yönlendirebilme yeteneğine sahip bir anahtarlama birimi olarak modellenmiştir. Modellenen düğümler iki veya daha fazla ağ bağlantısıyla (link) birbirine bağlanan DEVS atomik modelleridir. Düğümlerin davranış karakteristikleri, trafiğin işlenmesi için bant genişliği, işlem yapma hızı ve trafiği işleyebilecek kapasitede tampon boyutudur. Tanımlanan karakteristiklerle oynanarak değişik kapasiteye sahip ağ birimleri oluşturulabilir ve farklı ağ senaryoları geliştirilebilir.

Bir MANET ağında normal bir ağda olduğu gibi yönlendiriciler yoktur. Gelen mesajların yönlendirme işlemi düğümler tarafından yapılmaktadır. Yönlendirme işlemine de sahip olan düğümler, bu işlemleri sahip oldukları karar mantığı ile sağlamaktadırlar. Yönlendirici modelleri belirli bir hedef noktası için gidiş bağlantısını belirlemek amacıyla bir yönlendirme karar mantığına ihtiyaç duyar. Bunu gerçekleştirmek için düğüm modeline yönlendirme tabloları yerleştirilmiştir (Şekil 5.5).

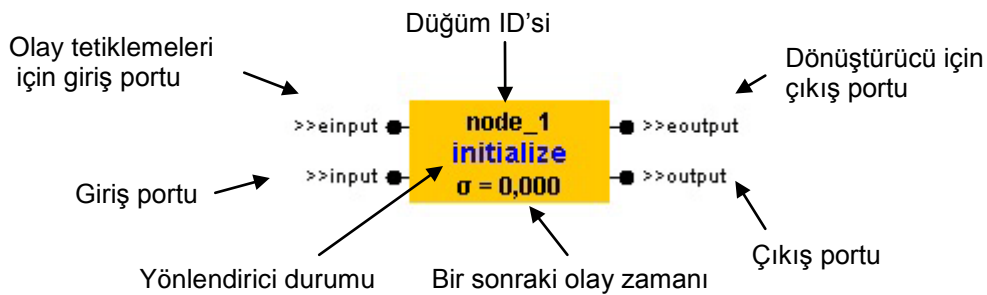
Şekil 5.5’de bir düğüm mimarisi görülmektedir. Burada yönlendirme modülü, bir başka deyişle IP İşleyici (IP Handler), yönlendirmenin yapıldığı birimdir. Her düğümde, paketler hedef düğümlerine yönlendirme modülü kullanılarak iletilir. Yönlendirme modülleri paket yönlendirme işlemi yönlendirme tablolarını ve yönlendirme algoritmalarını kullanarak gerçekleştirir. Daha sonraki bölümlerde de ifade edileceği gibi yönlendirme tabloları ağ içerisinde haberleşme sırasında elde edilen bilgilerden muhtemel hedef düğümler için verilere sahiptir. Bu veriler paketin belirli bir hedef için kullanacağı yollardır.

Oluşturulan düğümler bir ağ ara birimine sahiptir. Ağ ara birimi gelen ve giden mesajların basit bir şekilde kuyruklama işlemine (drop-tail) alınmasını yürüten çok basit bir MAC protokolü uygulamaktadır.



Şekil 5.5. Geliştirilen düğümün kavramsal modeli

Bir düğümün DEVS-Suite ekranında görsel olarak görünümü Şekil 5.6'da görülmektedir.

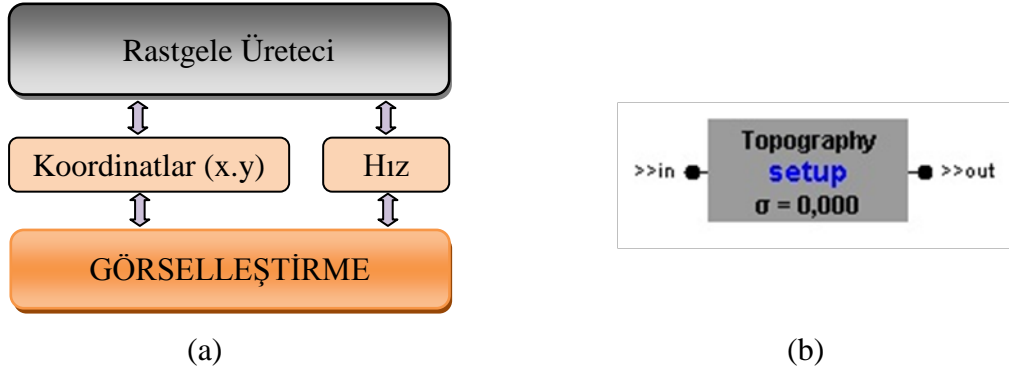


Şekil 5.6. Düğüm atomik modelinin ekran çıktısı.

5.4.2. Topografya atomik modeli

Düğümün hareket kabiliyetine sahip olması birbiri ile olan bağlanabilirliğinin sürekli değişmesine sebep olmaktadır ve düğümler arasında mesaj trafiği için kullanılan bağlantıların dinamik olmasını zorunlu kılmaktadır. Bu dinamizmin

kontrolünü her bir düğüme bırakmak gereksiz kod yüküne sebep olacağından dolayı MANET ağ ortamında bağlantıların kurulması ve kaldırılması işlemi Topografya adı verilen atomik modeli tarafından kontrol edilmektedir (Şekil 5.7).



Şekil 5.7. Topografya atomik modelinin kavramsal modeli (a) ve ekran çıktısı (b)

Oluşturulan atomik modelin belirlenen aralıkta periyodik olarak yerine getirdiği / yaptığı görevler aşağıdaki şekilde özetlenebilir;

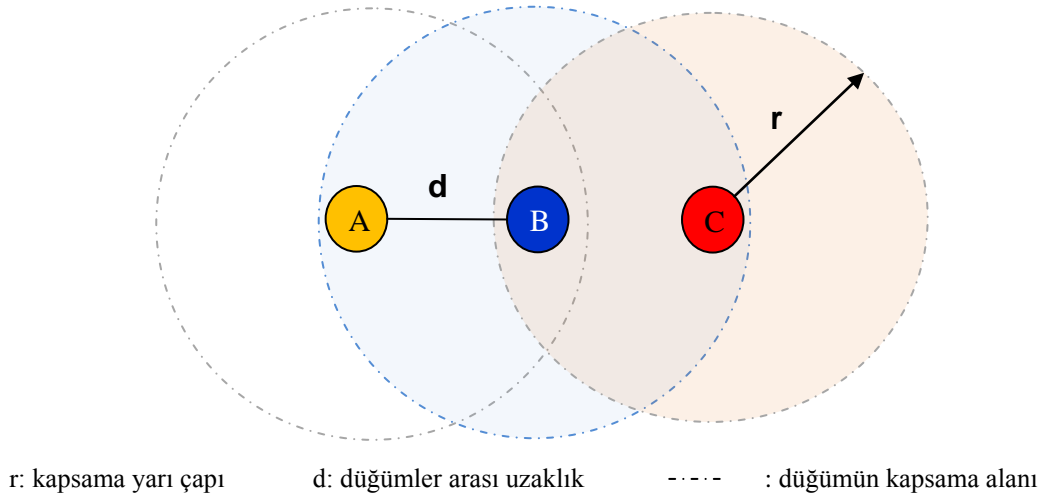
- Düğümleri belirlenmiş hız ve yön bilgisine göre yeni koordinatlarına konumlandırır.
- Konumu değişen düğümün diğer düğümler ile kapsama alanı içinde olup / olmadığını kontrol ederek bağlantılarını tekrar kurar / kaldırır.
- Düğümlerin hareketleri ve bağlantıları görsel olarak ekranda güncellenir

Aşağıda bu dinamizmin nasıl sağlandığı detaylandırılmaktadır.

5.4.2.1. Topolojideki değişimlerin modellenmesi

Hareketli düğümler kablosuz oldukları için bir başka düğümlerle iletişimi antenleri vasıtasıyla coğrafik olarak birbirlerinin kapsama alanı içinde ise kurabilir. Şekil 5.8'de A, B ve C düğümlerinden A düğümü sadece B düğümü ile iletişim kurabilirken, B düğümü A ve C düğümü ile iletişim kurabilmektedir.

MANET-DEVS ortamında bir düğüm ile başka bir düğüm arasındaki iletişime iki düğüm arasındaki uzaklığın (d) hesaplanması ile karar verilir. Düğümler için belirlenen kapsama alanı hesaplanan uzaklıktan büyük ise düğümler arasında iletişim var kabul edilir ve bağlantı kurulur.

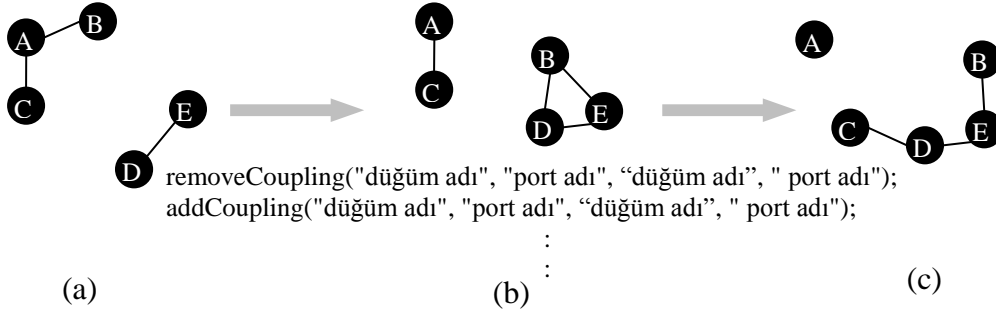


Şekil 5.8. Hareketli düğümlerde bağlantı kurulumu

Tasarlanan benzetiminin mobil ağlar için olması sürekli konumu değişen düğümlerin birbiri ile olan bağlantılarının da dinamik olmasını gerektirmektedir. Zamanın belirli bir anında iletişim halinde olan düğümler bir başka zaman diliminde farklı koordinatlarda oldukları için aralarındaki bağlantıyı kaybedebilmektedirler. Buda ağın topolojisinin devamlı değişmesine sebep olmaktadır. Değişken topoloji altında düğümlerin arasındaki bağlantıların kurulup kaldırılması DEVS-Suite ortamının sahip olduğu dinamik yapı (variable structure) sayesinde mümkün olmaktadır [73] [74].

Dinamik yapı DEVS, benzetim esnasında bağlantıların ve modellerin eklenip kaldırılmasını sağlar. DEVS bileşen tabanlı modelleme ve benzetimi desteklemektedir. Bu destek sadece ortamın modellenmesi ve benzetimi için değil hiyerarşik modüler modelleme için de geçerlidir. DEVS ortamında model, giriş ve çıkış portları olarak adlandırılan ara yüzlerle tanımlanmış bir bileşendir. Bir atomik veya ya birleşik modelin çıkış / giriş portlarını başka bir atomik veya bileşik modelin

giriş / çıkış portu ile bağlayarak mesajların bir bileşenden diğer bileşene geçişi sağlanır.



Şekil 5.9. Düğümler arası bağlantıların kurulması / kaldırılması

Düğümler Şekil 5.9 (a) ile belirtilen konumdan, Şekil 5.9 (b) ile belirtilen konuma geçtiklerinde B düğümünün ağdaki diğer düğümler ile olan bağlantıları ve bunların değişimi görülmektedir.

Düğümler arası iletişimi etkileyen solma (fading), fiziksel ortamın şartları ve mesajın ortamda hedefine erişmeden önceki kaybettiği süre göz ardı edilmiştir.

5.4.2.2. Hareketlilik modeli

Bir gezgin Ad Hoc ağda hareketlilik modelinin değişmesi başarımlar ölçümlerinin önemli miktarda değişmesine neden olacağı için seçilecek olan hareketlilik modeli önemlidir. Normalde gerçek hayattaki hareketliliğin yazılım ortamına aktarılmasındaki güçlükten dolayı yapay hareketlilik modellerini kullanmak kaçınılmazdır. Ad Hoc ağlar için birçok hareketlilik modeli vardır (Random Walk Mobility Model, Random Waypoint Mobility Model, Random Direction Mobility, A Probabilistic Version of the Random Walk Mobility, vb.).

Yapılan çalışmada, hareketlilik modellerinden Brownian devinimi prensibine göre çalışan “Rastgele Yürüyen Hareketlilik Modeli” (Random Walk Mobility Model) kullanılmıştır [34]. Kullanılan model 1827’de botanikçi Robert Brown tarafından

gözlemlenmiştir ve daha sonra matematiksel tanımlaması ilk olarak Albert Einstein tarafından 1926 yılında yapılmıştır.

MANET-DEVS ortamında ‘topografya’ adlı atomik modeli tarafından düğümlerin sınırları belirlenmiş 2 boyutlu düzlem üzerinde hareket etmeleri sağlanmaktadır. Düğümlerin hareketlerini rastgele bir koordinattan yine rastgele seçilmiş başka bir koordinata doğru rastgele bir hızla yapmaları koordine edilir. Hedefe ulaşan düğüm rastgele belirlenmiş değerlerle, yeni bir hedefe doğru hareketlerine devam etmektedirler.

5.4.2.3. Topografyada düğümlerin görsel takibi

MANET-DEVS ortamında geliştirilen ara yüzle düğümlerin konumu, birbirine olan komşulukları ve hareketleri takip edilebilmektedir. Yüksek düğüm sayısında ek yük getireceği için oluşturulan ara yüzün ekranda görülmesi / görülmemesi seçenek olarak sunulmuştur. Topografyanın yönettiği bu özellik Bölüm 5.6’da detaylı olarak anlatılmaktadır.

5.4.3. Ağ paketleri

Kurulan gezgin ağın bileşenlerinden birisi, bileşenler arasında karşılıklı değişilen mesaj paketleridir. Mesaj paketleri ağ bileşenleri arasında karşılıklı değişilen nesnelere sahiptir. Yapılan çalışmada DEVS-Suite ortamının geliştirildiği Java programlama dilinin esnekliği sayesinde, tanımlanan paketler esnek yapılara ve yeteneklere sahip olacak şekilde modellenmiştir. Bu esneklik sayesinde paketler farklı formlara sokulabilmektedir.

Kullanılan ağın izlenmesini, kontrolünü ve verimli iletişimini sağlayan yönlendirme protokolüne ait kontrol paketleri ve veriyi taşıyan paketler olmak üzere iki temel paket oluşturulmuştur. Bu paketler temelde aynı yapıya sahip olmalarına ve aynı bilgiyi taşımalarına karşılık, farklılıkları tanımlanan davranışlarındadır. Veri

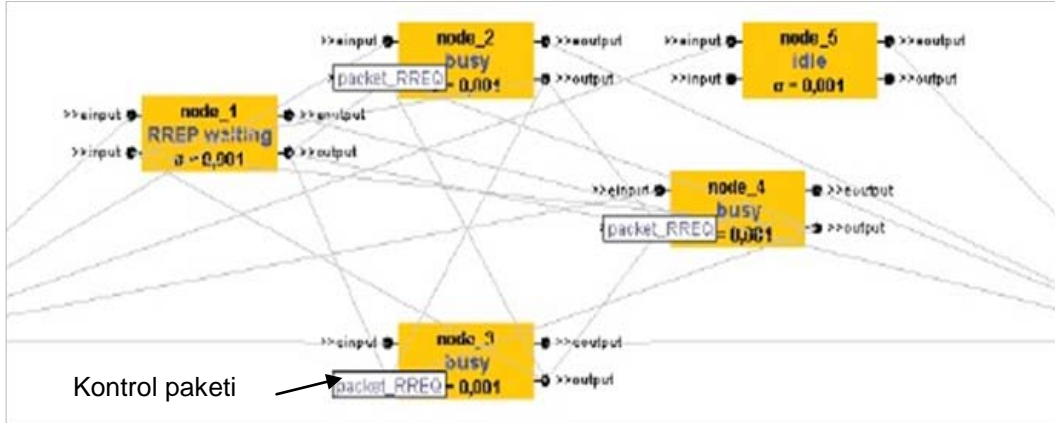
paketleri, bilgi taşıyan temel IP paketleri olarak düşünülebilir. Kontrol paketlerinin özelliklerini sıralayacak olursak;

- Düğümler tarafından oluşturulur,
- Ağda belirlenmiş bir görevi yerine getirmek için iletilirler ve
- Düğümlerde bulunan yönlendirme veritabanları üzerinde ekleme / silme / değiştirme / güncelleme vb. yönlendirme protokolünün ihtiyacı olan işlemlerin yerine getirilmesini sağlarlar.

Bir paket, düğüm atomik modeli tarafından üretildiği zaman paketi üreten düğümün bağlantılı olduğu diğer düğümlere iletilir. Paketi alan düğüm, gelen paketler için oluşturulmuş kuyruğa paketi yerleştirir. Kuyruğa giren paketler FIFO prensibine göre işleme alınır. İşleme alınan paketin nasıl değerlendirileceği, kontrol paketinin türüne göre ve yönlendirme protokolünün kurallarına bağlıdır.

Düğümlerdeki kuyruk nesnelere belirli bir kapasiteye sahiptir. Bu nedenle gelen mesajlar eğer kuyrukta yeterli boş alan var ise alınır, aksi takdirde mesaj atılır ki bu veri kaybına sebep olur. Bunun dışında gelen bir paketin atılmasının yönlendirme protokolünün kurallarına göre farklı sebepleri olabilir (paketin yaşam ömrü (TTL- Time to Live), daha önce benzer paketin gelmiş olması vb).

Gerçek bir ağda paket içerisinde bulunan ve paketin türüne göre değişkenlik gösteren protokol başlığını (Protokol header) oluşturan paket kimlik numarası, türü, yaşam ömrü, kaynak adresi, hedef adresi vb. bilgiler MANET-DEVS ortamında paket nesnesi içerisindeki değişkenlerde saklanır. Paket başlığı için 20 Bayt'lık bir standart bir IP paketi boyutu kabul edilmiştir. Normalde bir paketin boyutu taşıdığı veri boyutuna göre değişmektedir. Hazırlanan benzeticide paketin taşıyacağı veri boyutu paket içindeki bir değişkende tutulur ve iletilen paketin toplam boyutu gerektiğinde buradan hesaplanır. Şekil 5.10'da DEVS-Suite ortamında bir paket nesnesinin düğümler arasında iletilmesi görülmektedir.



Şekil 5.10. Bir paketin MANET-DEVS ortamında bileşenler arasında hareketi.

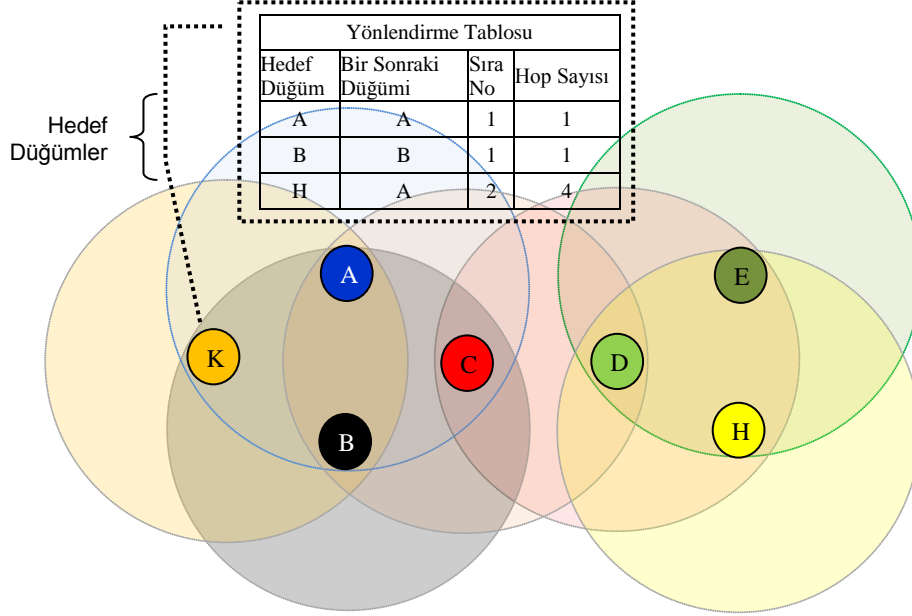
Düğüm tarafından iletilmek istenen veri paketi iletişimin sağlandığı kuyruktan farklı bir kuyruksa bekletilmektedir. Böylelikle veri paketinin iletilmesi için yönlendirme protokolünün gerekli kontrol mesajlarını yayınlaması ve alması sırasında geçen sürede ağdaki diğer düğümler tarafından yayınlanan mesajlar işleme konarak ağın daha hızlı, sağlıklı ve verimli olarak iletişiminin gerçekleştirilmesi sağlanmış olmaktadır.

Yüksek başarılı bir benzetici tasarlamak adına geliştirilen MANET-DEVS ortamında, hata giderme ve yol onarımı için kullanılan paketler dikkate alınmamıştır. Diğer bir amaç MANET ağ katmanlarından sadece yönlendirme katmanına odaklanılmasıdır.

5.4.4. Yönlendirme tabloları

Düğüm, ağdaki muhtemel her bir hedef düğüme ait erişim bilgilerinin tutulduğu bir yönlendirme tablosuna sahiptir. Yönlendirme tablolarının biçimi ağda kullanılan yönlendirme protokolüne göre değişmektedir. Modellenen yönlendirme tablosunda her bir satır muhtemel bir hedefe aittir. Sütunlar ise o hedefe ait bilgileri içerir. Yönlendirme tablosunun boyutu bilgi sahibi olunan düğüm adedi ile orantılı olarak değişmektedir. Bir düğüm bir veri paketini bir hedefe göndermek istediğinde veya

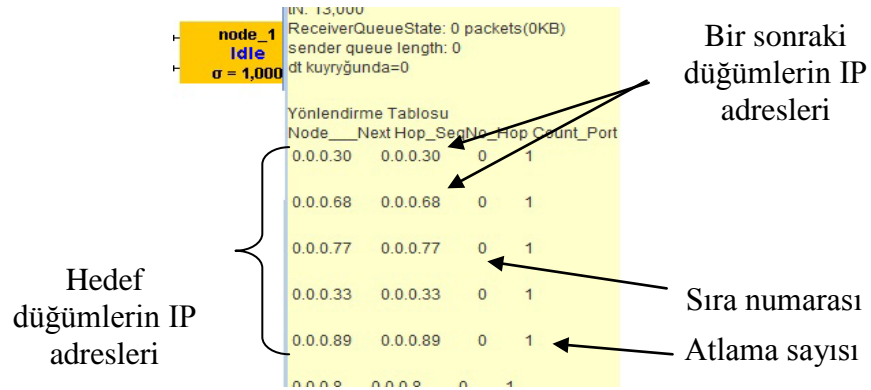
bir paketin iletilmesi işlemini gerçekleştirirken sahip olduğu yönlendirme tablosuna başvurarak hareket eder (Şekil 5.11).



Şekil 5.11. Bir yönlendirme tablosu örneği.

Yönlendirme tablolarının oluşturulması / güncellemesi yönlendirme algoritmalarının görevidir ve yönlendirme algoritmasının başarısını belirleyen bir etkidir. 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 yeni girişler eklenebilir ve mevcut kayıtlar kaldırılabilir.

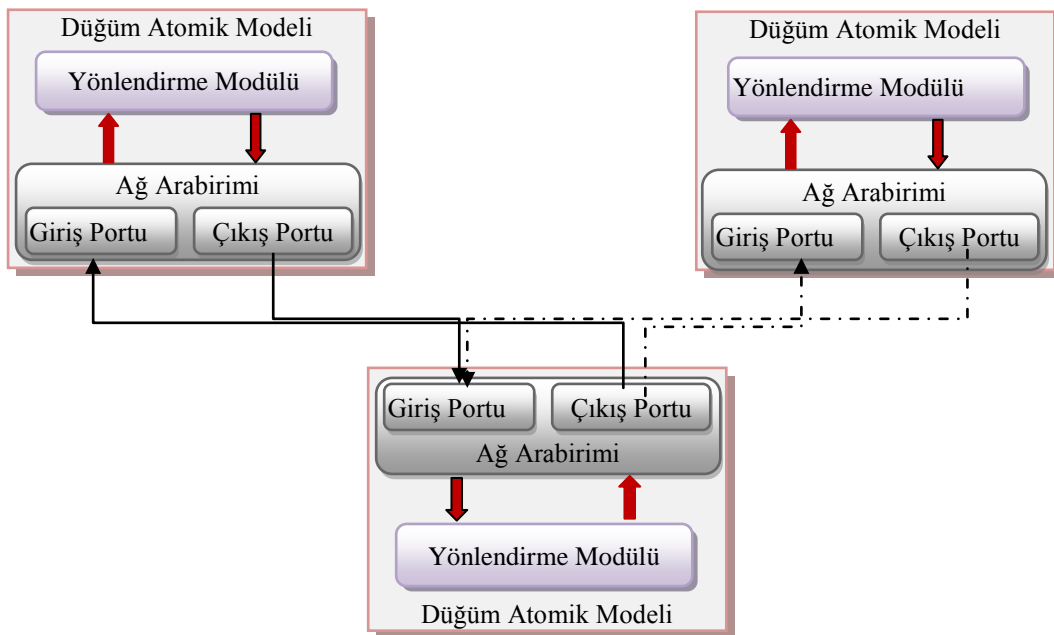
AODV yönlendirme protokolünde örnek bir yönlendirme tablosu Şekil 5.11'de gösterilmektedir. Bir yönlendirme tablosunda hedef, bu hedefe ulaşmak için hangi düğüme başvurulması gerektiği, hedefe ulaşmak için gerekli atlama sayısı ve yolun güncelliğinin takibini sağlayan sıra numarası tutulmaktadır. MANET-DEVS ortamında bir düğümün yönlendirme tablosu Şekil 5.12'de görülmektedir.



Şekil 5.12. Bir yönlendirme tablosunun DEVS-Suite altında ekran görüntüsü.

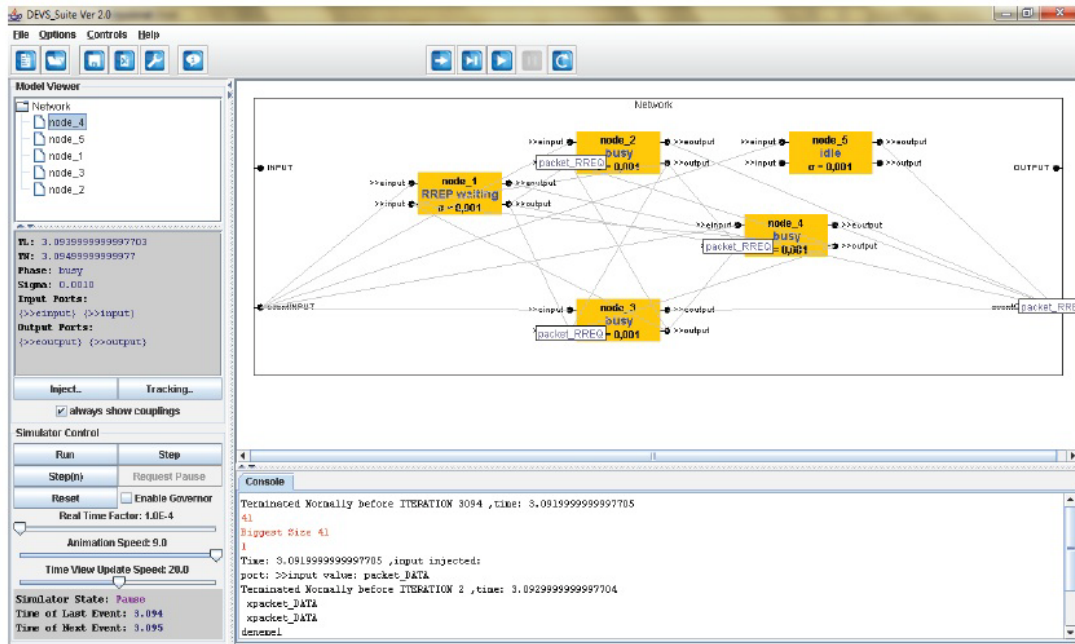
5.4.5. Birleşik (Coupled) model

DEVS birleşik model yaklaşımı kullanılarak, atomik modeller birbirine bağlanarak oluşturulan birleşik modeller sayesinde değişik topolojilerde ve dinamik yapıda ağlar oluşturulabiliriz. Şekil 5.13’de, 3 düğümden oluşan bir birleşik model görülmekte, Şekil 5.14’de ise MANE-DEVS ortamında düğümlerin birbirleri ile olan bağlantıları görülmektedir.



Şekil 5.13. Düğümlerin bir biri ile bağlantı kurması

DEVS'in birleşik model özelliğinden faydalanarak MANET-DEVS ortamında oluşturulan ağ, belirlenen senaryolar ve farklı değişken değerleri ile test edilebilir. Hazırlanan MANET-DEVS ortamı farklı sayıdaki düğümlerden oluşan farklı topolojiye sahip gezgin ağların araştırılmasına imkân sağlamaktadır. Test edilebilecek ağın büyüklüğü benzetimin çalıştırılacağı bilgisayarın donanım kapasitesine bağlı olarak değişmektedir.



Şekil 5.14. MANET-DEVS ortamında bir ağ modeli birleşimi.

5.4.6. MANET-DEVS deneysel çerçevesi ve ağ trafik modeli

Gerçek dünyada çalışıldığında, deneysel çerçeve (Experimental Frame - EF) sistemin / modelin çalıştırılacağı deneysel şartları / ortamları tarif eder. Gerçek dünyadan farklı olarak yazılım ortamında ise kurulan sistemin / modelin test edilebilmesi için Bölüm 3'de anlatılan DEVS içerisindeki deneysel çerçeve kavramının oluşturulması / tanımlanması gerekmektedir.

Deneysel çerçeve üç bileşenden oluşur;

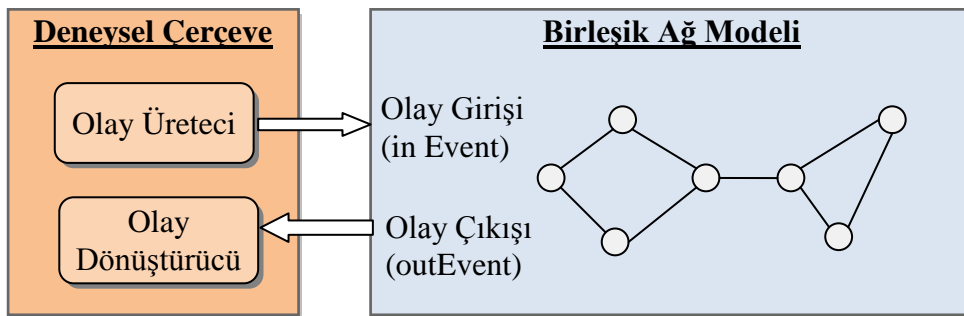
- Sistemin / modelin giriş uçlarına, sistemi / modeli uyarmak için bağlanan bir üreteç (event generatör),
- Sistemin / modelin çıkış uçlarına, sistemden / modelden gelen sonuçların değerlendirilebilmesi için bağlanan bir dönüştürücü (event transducer),
- Giriş / çıkış değişkenleri ile birlikte deneysel çerçeve içindeki üreteç girişleri ile dönüştürücü çıkışlarını karşılaştıran bir onaylayıcıdan (acceptor) oluşur. Onaylayıcı, sistemin (gerçek veya model) deneysel çerçevesinin deneyi yapanın hedefleriyle uygun olup-olmadığını belirler [48].

Genellikle bir ağın başarımı ölçülürken belirli kriterlere bakılır [75]. Bunlar;

İş çıkarma yeteneği (throughput): ağ üzerindeki bir birim zamanda ağdan geçip giden paketlerin sayısıdır veya bilgi miktarıdır.

Ortalama gecikme: Belirli bir zaman aralığında hedefine ulaşan bütün paketler tarafından maruz kalınan ortalama gecikme, ağın genel durumu hakkında önemli bir parametredir.

Bekleyen paketlerin sayısı: Ağda düğümlerin kuyruklarında bekleyen paketlerin toplam sayısı, ağ trafiğinin seviyesini belirlemeye ve tıkanıklıkların tespitine yarar.

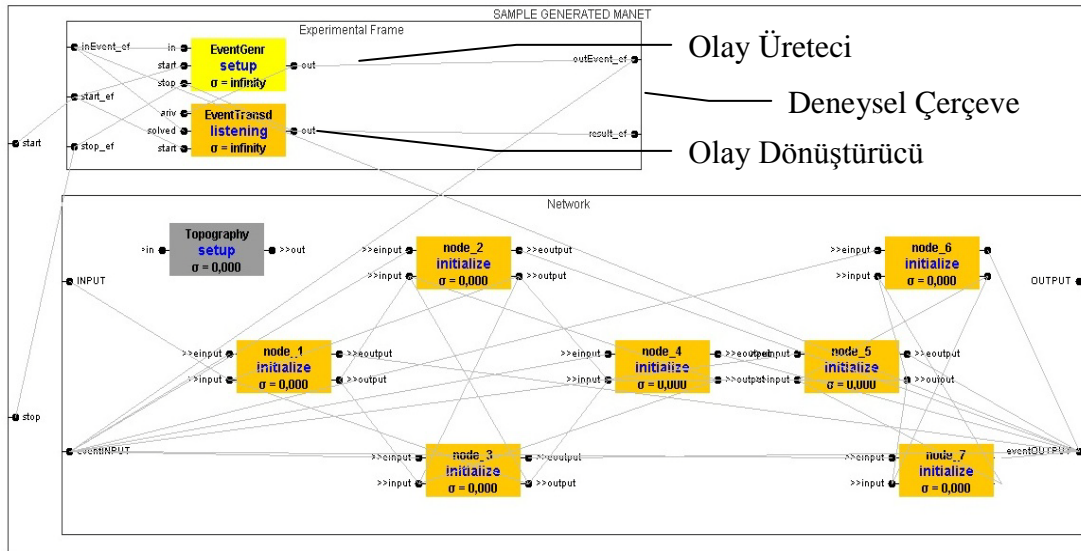


Şekil 5.15. Deneysel çerçeve ve bileşenleri

Şekil 5.15’de yapılan çalışmada bir mobil ağın başarımlarını ölçümlerini yapabilmek için hazırlanan DEVS deneysel çerçeve bileşenleri ve ağa uyarlanması görülmektedir.

Olay dönüştürücü yapılan benzetim çalışmasının sonuçlarını değerlendirmek için kullanılır. Ağın gözlemlenmesi ile elde edilen çıktılar analiz edilerek / hesaplanarak ağı değerlendirmemize yarayacak anlamlı sonuçları elde etmemize yardımcı olur. Olay dönüştürücü tarafından gözlemlenen MANET-DEVS ortamına ait veriler bir CSV dosyasına (Comma Separated Values - CSV) kaydedilir. Saklanan bu dosya istenildiği zaman başarımlarını değerlendirmesi için kullanılacak grafiklerin oluşturulmasında yardımcı olur.

DEVS-Suite ortamında MANET-DEVS benzeticisi için oluşturulan deneysel çerçeve ile ağın oluşturduğu bileşik modelinin ekran çıktısı Şekil 5.16’da görülmektedir.



Şekil 5.16. Bir ağın deneysel çerçeveyle bağlantısı.

5.5. Topoloji Üretici

Ağ topoloji üreticileri gerçeğe yakın benzetim çalışmaları ve ağın başarımlarını karakteristiklerinin incelenmesi için araştırmacılar tarafından sıklıkla

kullanılmaktadır. 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. Özetle ağın başarımı önemli ölçüde topolojiye dayanır [76].

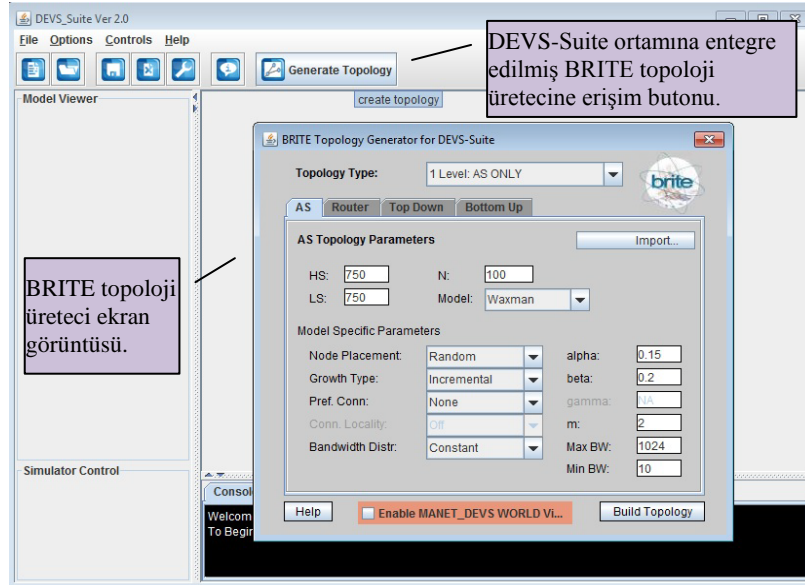
İdeal topoloji üreticileri İnternet topolojilerinin doğru gösterimini üreterek protokol ve algoritmaların doğruluğunu araştıran ve yeni nesil güçlü modeller üzerinde çalışan araştırmacıların kullanımına sunar [77].

Aşağıda topoloji üreticilerinin karakteristik özellikleri listelenmektedir [77].

- Temsil edilebilirlik: Doğru yapay topolojiler üretir. İnternet topolojilerini farklı açılardan doğru yansıtır.
- Kapsamlılık: Tek bir topoloji üreticinde mümkün olduğunca fazla üretic modelini barındırır.
- Esneklik: Geniş ölçekte topoloji üretebilir, mantıklı sayı aralığında minimum ve maksimum düğüm sınırlaması getirmez.
- Verimlilik: Büyük ölçeklerde topoloji üretirken CPU ve bellek tüketiminde cimri davranır.
- Genişleyebilirlik: Kullanıcı tarafından yeni modellerin eklenebildiği mekanizmaları destekler.
- Kullanıcı dostu: Standart kullanıcı arayüzü prensiplerini kullanır. Kullanıcılar tarafından üreticinin mekanizmasının kolay öğrenilmesini sağlar.
- Karşılıklı çalışabilirlik: Benzetim ve görselleştirme araçları için arayüzler sağlar. Geniş kullanıma sahip benzetimciler tarafından kullanılmasına imkân verir.
- Sağlamlık: Geniş hata sezme yeteneğine sahiptir.

Topoloji üreticileri ile ilgili önemli bir çalışma M. A. Rahman ve arkadaşları tarafından yapılmıştır [76]. Bu çalışmada BRIT (The Boston University Representative Internet Topology Generator) topoloji üreticisi geniş ölçekli ağları

desteklemesi, birçok önemli topoloji modelini desteklemesi, hem C++ hemde Java ile kodlanmış sürümlerinin bulunması ve açık kaynak kodlu olması ile diğer topoloji üreticileri arasında öne çıkmaktadır.

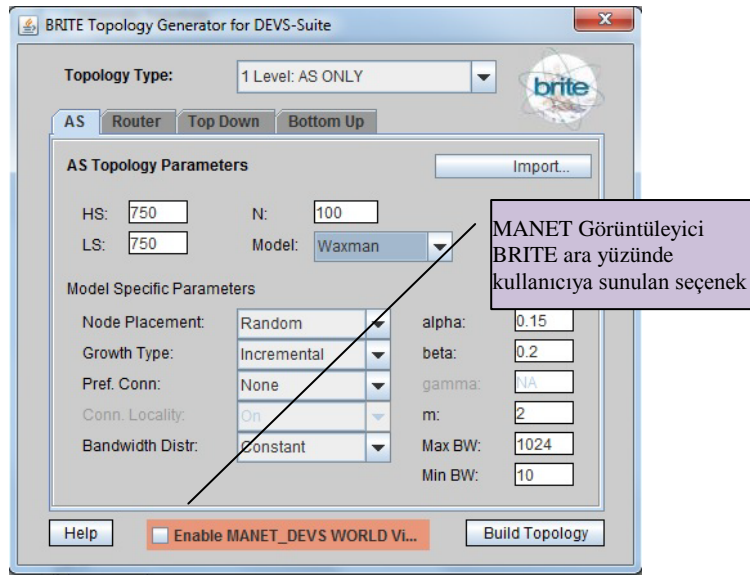


Şekil 5.17. DEVS-Suite BRITE topoloji üretici ekran görüntüsü

BRITE topoloji üretici DEVS-Suite benzetim ortamına entegre edilerek (Şekil 5.17), hazırlanan MANET-DEVS ortamında yapılacak deneylerde kullanılacak ağın topolojisi, BRITE topoloji üreticinden elde edilmektedir.

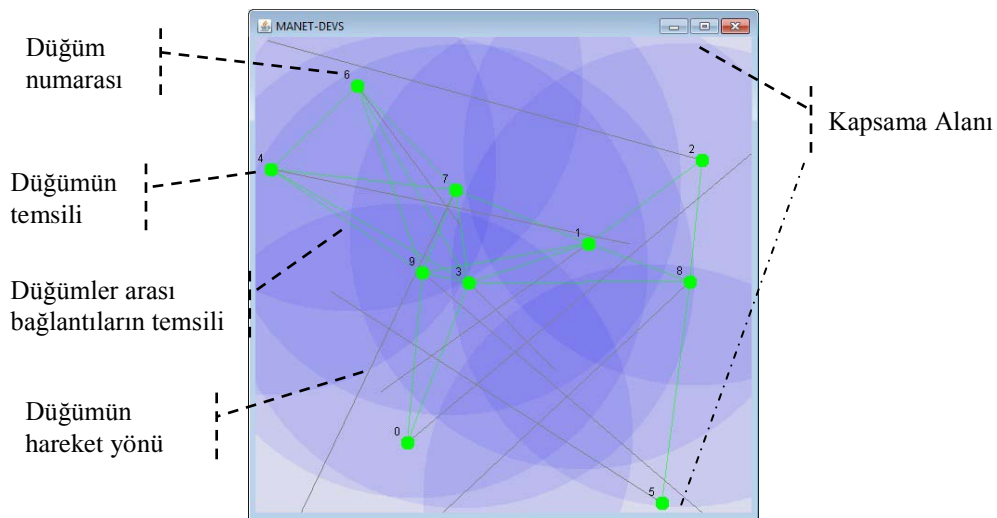
5.6. Görsel Takip

MANET-DEVS ortamında benzetimi yapılan ağı oluşturan düğümlerin ağ üzerindeki konumu, kapsama alanları, hareket yönü ve hareketlerinin görsel takibini yapmak amacıyla “MANET Görüntüleyici” adı verilen bir modül tasarlanmıştır. Hazırlanan görüntüleyici bu hizmeti verebilmek için benzetimin çalıştırıldığı kaynakları kullanmaktadır (RAM vb.). Bu nedenle geniş ölçekli ağların benzetiminde ihtiyaç duyulan kaynakların verimli kullanılabilmesi için bu ara yüzün çalıştırılıp çalıştırılmaması BRITE topoloji üreticisine entegre edilen bir buton ile kullanıcıya seçenek olarak sunulmuştur (Şekil 5.18).



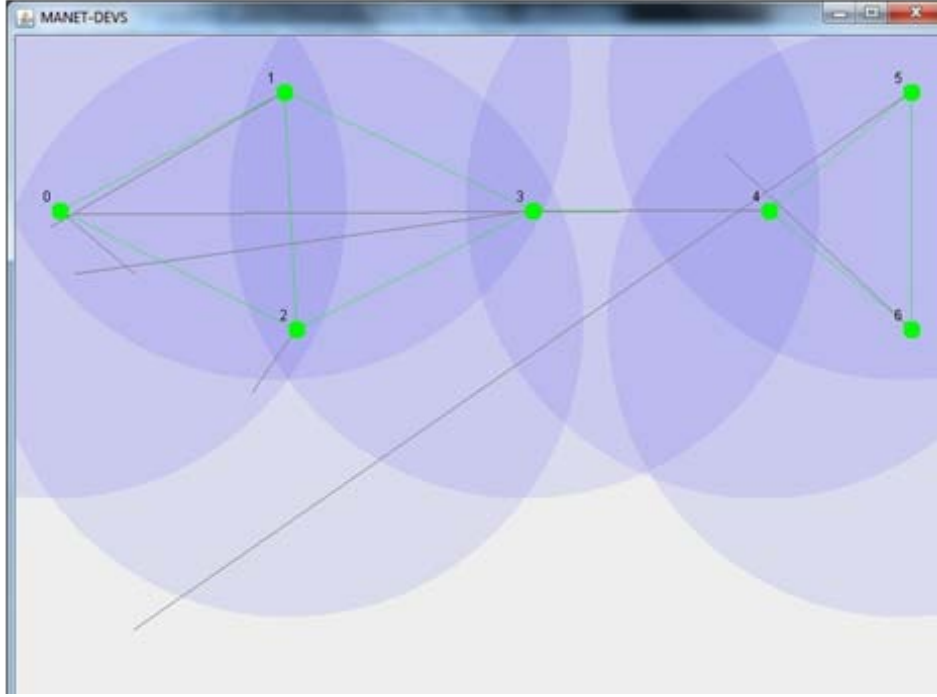
Şekil 5.18. MANET görüntüleyici için kullanıcı seçeneği

MANET-Görüntüleyici ekranında $1m=1$ piksel olarak kabul edilmiştir. Şekil 5.19'da 500 X 500 bir alanda 10 düğümden oluşan bir topolojinin ekran görüntüsü verilmiş ve bu ekran görüntüsü üzerindeki şekillerin ifade ettikleri manalar açıklama baloncuklarında belirtilmiştir

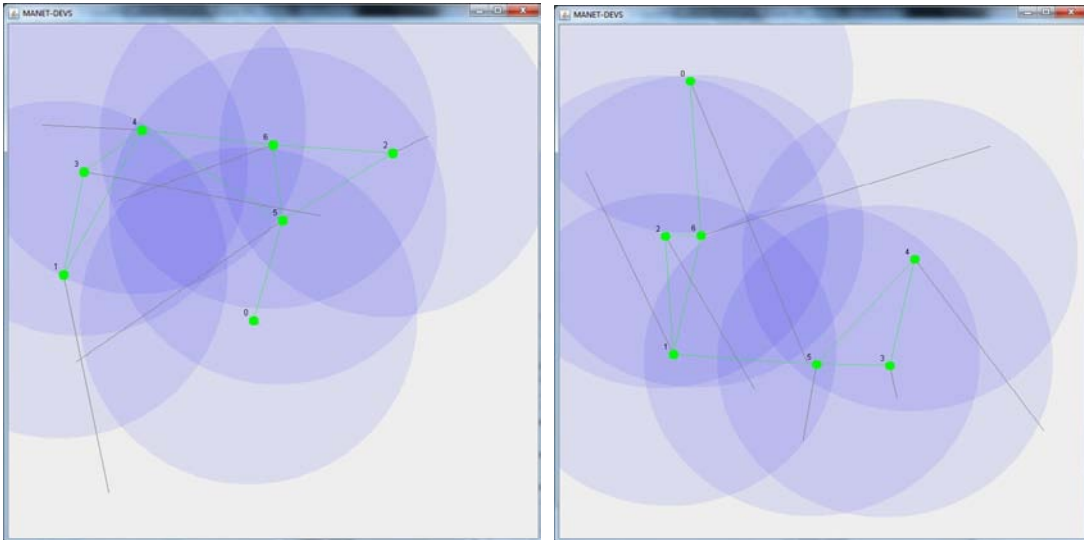


Şekil 5.19. MANET-Görüntüleyicide 500 X 500 lük bir alanda topoloji dağılımı

7 düğümden oluşan hareketli bir ağın 3 farklı zaman dilimindeki konumlarını gösteren ekran görüntüsü Şekil 5.20'de görülmektedir.



(t 1)



(t 2)

(t 3)

Şekil 5.20. 7 düğümlü bir ağın takibi için tasarlanan ekran çıktısı

BÖLÜM 6. MANET-DEVS BENZETİM DENEYLERİ VE SONUÇLARI

6.1. Giriş

Bu bölümde, DEVS-Suite ortamında geliştirilen MANET-DEVS modelleme ve benzetim ortamının kapasitesini ölçmek / test etmek amacıyla benzetim deneyleri gerçekleştirilmiştir. Bu çalışmaları;

- MANET'lere yönelik geliştirilen ve en yaygın kullanılan yönlendirme protokollerinden biri olan AODV yönlendirme algoritmasının MANET-DEVS modelleme / benzetim ortamında modellenmesi,
- Hareketli düğümlerin davranışlarının modellenmesi,
- Geliştirilen deneysel çerçeve yardımıyla farklı ölçeklerde, farklı topolojilerde ve farklı ağ trafiği altında üretilen senaryolar sonucunda elde edilen veriler ve gözlemlenen çıkışların grafikler halinde sunulması,

olarak sıralayabiliriz.

Gerçekleştirilen çalışmalarla, MANET-DEVS modelleme / benzetim ortamının MANET'lerin karakteristiklerini modelleme ve benzetim yeteneğinde olduğu gösterilmeye çalışıldı.

6.2. AODV Yönlendirme Algoritması Uygulaması

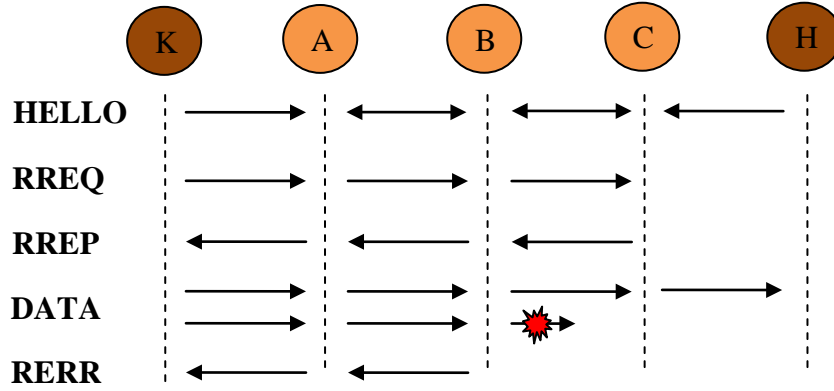
Bölüm 4'de anlatılan yönlendirme konusu; ağlarda hızı, verimliliği, güvenliği etkileyen ve aynı zamanda araştırmacıların üzerinde en çok çalıştığı konulardan birisidir. Aşağıda isteğe bağlı yönlendirme protokolleri arasında en yaygın kullanıma

sahip yönlendirme protokollerinden biri olan AODV yönlendirme protokolünün MANET-DEVS ortamına nasıl aktarıldığı / uygulandığı anlatılmaktadır.

6.2.1. AODV yönlendirme algoritması

Geliştirilen MANET-DEVS ortamında örnek bir uygulama olarak temel bir isteğe bağlı yönlendirme algoritması olan AODV algoritmasının modellenmesi çalışması yapıldı. Bu modelleme çalışması ile MANET-DEVS ortamının Ad Hoc yönlendirme algoritmalarını modelleme ve benzetim yeteneğine sahip olduğu gösterildi.

AODV, özellikle MANET'ler için tasarlanmıştır. Sürekli değişen ağ topolojisi nedeniyle yönlendirme tablosunda düğümlere ait bilgiler belirli bir süreden fazla (active root timeout) saklanamaz. Eğer bir düğüm başka bir düğüme mesaj göndermek isterse, o düğüme ait yol bilgisine sahip olmak zorunda değildir, ihtiyaç duyduğunda yolun keşfi için harekete geçer.



Şekil 6.1. AODV yönlendirme protokolü mesajlaşması

AODV yönlendirme algoritması; RREQ, RREP ve RERR olmak üzere 3 temel kontrol mesajı ve veri paketlerinin gönderildiği mesaj türüne sahiptir (Şekil 6.1). Bir düğüm bir başka düğüme mesaj göndermek istediğinde, önce yönlendirme tablosunda hedef düğüme ait yol bilgisine sahip olup olmadığına bakar. Eğer hedef düğüme ait bir yol bilgisi var ise veri paketini iletir, aksi takdirde düğüm veri paketinin gönderilmek istediği hedefe ait yol bilgisini elde etmek için bir RREQ

mesajı yayınlar. Bir düğüm kendisine komşu olan bir veya daha fazla düğümden aynı RREQ mesajını alabilir. Düğümün kendisine gelen mesajlardan sadece birisini işleme koyması RREQ mesajının ID numarasının kontrolü ile sağlanır. RREQ mesajını ileten düğümler bu mesaja ait kaynak adresini ve ID numarasını tamponda belirlenen süre boyunca saklar. Bu süre zarfında aynı kaynak ve ID'ye sahip RREQ mesajlarını işleme koymayarak sonsuz döngülerin oluşmasını engeller. Mesajı alan düğümler hedef düğüme ait yol bilgisi için kendi yönlendirme tablolarını kontrol ederler, eğer yol bilgisine sahip ise bir düğüm RREP mesajı yayınlar, değil ise RREQ mesajını iletir. RREQ mesajı tüm düğümler tarafından işleme konurken (broadcast), RREP mesajı sadece tek bir düğüm tarafından işleme konur (unicast). Böylece hedef ile kaynak arasında yol bilgisi elde edilmiş olur ve bu yol üzerinden veri paketleri gönderilir.

RRER mesajı, veri iletimi sırasında hedef ile kaynak arasında kurulan yolda bir kopma olması durumunda, mesajı iletemeyen düğüm tarafından kaynağa doğru gönderilir. RRER mesajını alan ara düğümler, kaynağa bu mesajı iletirken kendi yönlendirme tablolarını da güncellerler. Hata mesajını alan kaynak düğüm hata giderme prosedürlerini işletir. MANET-DEVS ortamında hata denetim mekanizması ihmal edilmiştir.

6.2.2. MANET-DEVS'te uyarılama işlemi

Bir MANET ağ üzerinde yönlendirme algoritmalarını uygulamak ve geliştirilen değişik trafik şekilleri altında MANET karakteristiklerini test etmek için, DEVS-Suite altında çeşitli uygulamalar oluşturuldu. Oluşturulan modelde ağı oluşturan bileşenler tanımlandıktan sonra, DEVS birleşik model tanımlamaları kullanılarak değişik topolojilerde ağlar meydana getirildi. Oluşturulan ağın, belirli bir trafik yükü altında davranışı izlenip, elde edilen sonuçlar grafikler halinde bu bölümün sonunda sunuldu.

6.2.2.1. D ğ m atomik modelinin davranış ve parametreler

Geliştirilen  rnek uygulamalarda (yapılan deneylerde), gerek d nya koşulları oluşturularak, b t n bileşenlere ait parametrelerin gerek d nyaya uygun seilmesine alışıldı. Deėişik senaryoları uygulamak amacıyla; aėın davranışını etkileyen benzetim parametrelerin benzetim alışmasından  nce ayarlanabilir veya benzetim sırasında deėiştirilebilir  zelliklere sahip olması saėlandı.

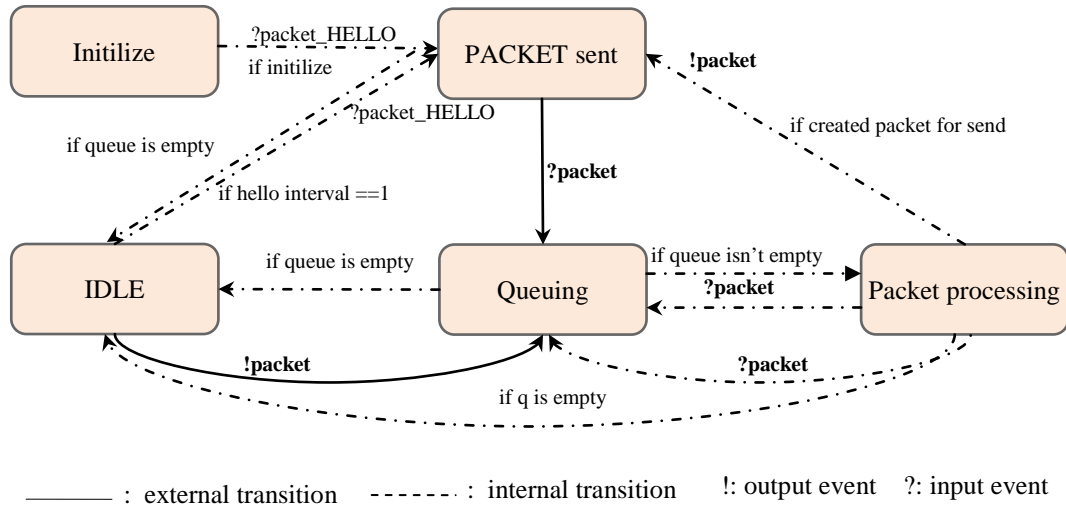
B t n d ğ mler; farklı IP adreslerine, giriş / ıkış portlarına, senaryoların uygulanabilmesi / g zlemlenebilmesi iin olay giriş / olay ıkış portlarına, y nlendirme tablosuna ve eşitli parametrelere (paket iőleme hızı, yol keşif s resi, HELLO mesaj aralıėı vb.) sahiptirler.

Bir d ğ m, baēlangı anında ve belirtilen zaman dilimlerinde (HELLO interval) HELLO mesajı g nderir. HELLO mesajını alan d ğ mler y nlendirme tablolarını g ncellerler. Y nlendirme tablosuna eklenen bir yol, belirlenen s re boyunca tabloda saklanır (active root timeout). Bu s re iinde saklanan yol ile ilgili herhangi bir mesaj gelmez ise; ilgili yol bilgisi silinir aksi takdirde yolun saklama s resi yeniden baēlar.

Daha  nce ifade edildiėi gibi d ğ m bir veri g ndermek istediėinde, verinin g nderilmek istendiėi hedefe ait yol bilgisinin y nlendirme tablosunda olup olmadığını kontrol eder. Yol bilgisinin olmaması durumunda, ise, yol bilgisini elde etmek iin RREQ mesajı yayınlar ve veri paketini RREP mesajını bekleme s resi kadar (path discovery time) bellekte tutar. Eėer RREP mesajı verilen s re ierisinde gelirse, veri paketini g nderilmek  zere kuyruėa bırakır. Bu s re ierisinde baēka g nderilmek istenen veri paketleri bulunursa; bu paketler, veri paketleri iin ayrılmıő kuyrukta bekletilirler. Yol olması durumunda ise paket bekletilmeden g nderilmek  zere kuyruėa bırakılır.

Düğümde kullanılan tüm kuyruklar FIFO prensibine göre çalışır. Tüm kuyruklar, kuyruğa eklenmek istenen paket boyutu kadar boş yer olması durumunda paketi kabul ederken, boş yer olmaması durumunda paketleri atarlar.

Şekil 6.2’de bir düğümün dinamik yapısını modellemekte kullanılan durum diyagramının sadeleştirilmiş hali görülmektedir. Şekilde tüm düğümler başlangıçta ‘kurulum-initilize’ durumdadırlar. Düğümler bu durumdan HELLO paketi göndererek çıkarlar. Düğüm bir paket göndermek istediğinde, paket türüne bakmaksızın ‘PACKET sent’ durumuna geçer. Düğüme gelen her türlü paket ve düğüm tarafından oluşturulan paketler ‘Queuing’ durumunda kuyruğa eklenir. Düğümün kuyruğu boş olmadığı sürece, kuyruktaki paketler sırasıyla ‘packet processing’ durumunda değerlendirilir ve verilen karara göre düğüm yeni durumuna geçer. Düğümün kuyruğu boş olduğu zaman ‘Idle’ durumunda kalır. Düğüm çalıştığı sürece belirlenen ‘HELLO interval’ süresinde periyodik olarak HELLO mesajları göndermeye devam eder.



Şekil 6.2. Bir düğüme ait sadeleştirilmiş durum diyagramı

Düğümde AODV davranışını sergilerken kullanılan algoritmalar EK-A’da, düğüm atomik modelinin DEVS sistem tanımlaması EK-B de görülebilir.

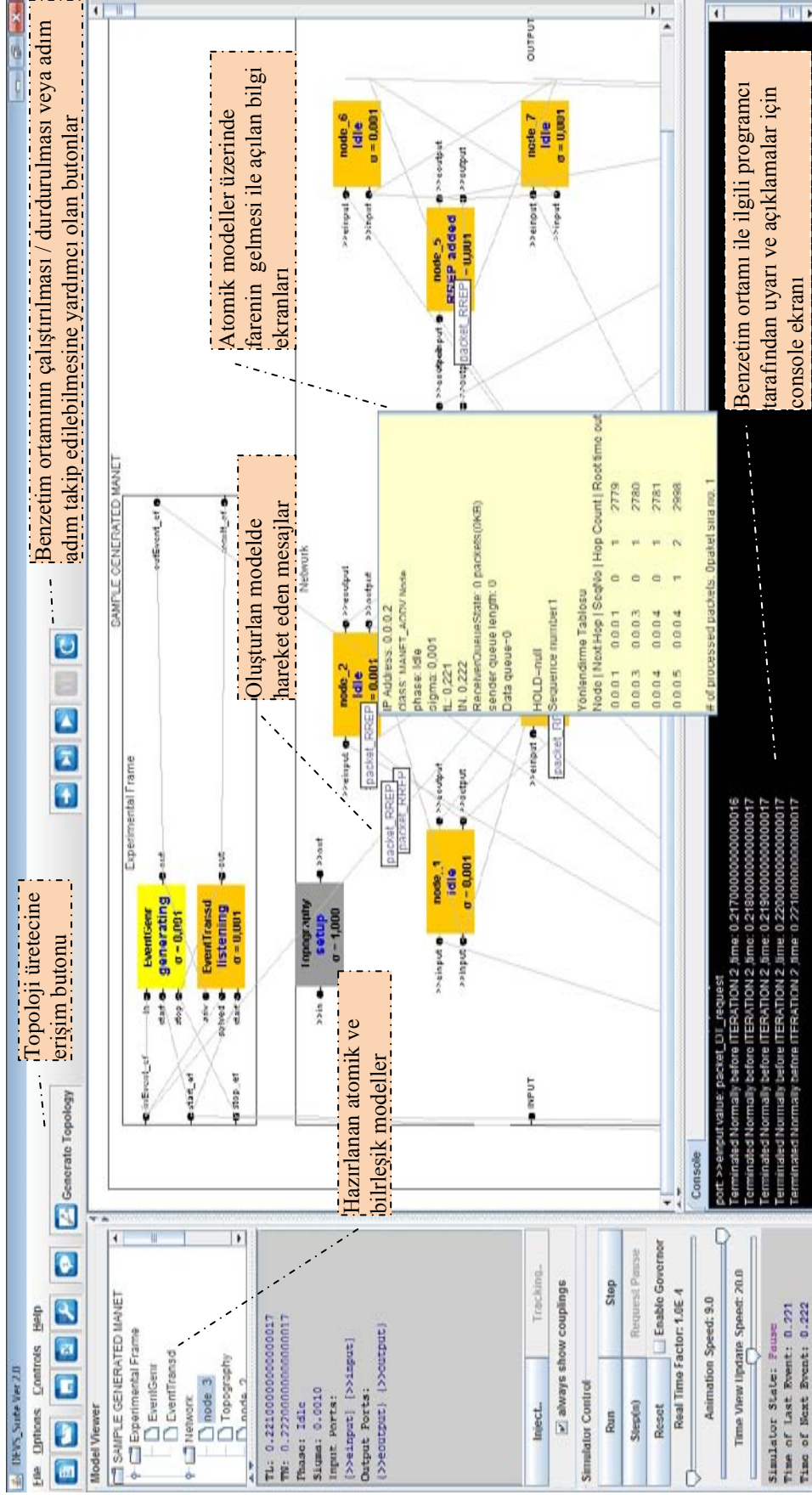
6.2.2.2. Ağ modeli

Geliştirilen çerçevenin verimliliğini değerlendirmek için, çeşitli ağ modelleri oluşturulup, oluşturulan modellerle DEVS-Suite altında çeşitli deneyler gerçekleştirildi. Bu çalışmada modellenen en basit ağ, 7 düğüme sahip bir ağdır. Yapılan çalışmalar sırasında gerçekleştirilen 7 düğüm sayısına sahip “basit ağ” oluştururken temel amacımız MANET-DEVS ortamında ağın çalışmasını ve uygulanan algoritmaların çalışmasını incelemek, modellemek ve davranışlarını gözlemlemektir. Basit ağda her bir düğüm, coğrafik olarak komşu bir düğümlle bağlanmıştır. Çalışmalar sırasında kullanılan basit ağın bir deneysel çerçeve ile birlikte bağlantısı ve DEVS-Suite altında ekran çıktısı Şekil 6.3’de görülmektedir.

Topoloji kurulurken ve ağ ile ilgili senaryo hazırlarken benzeticide kullanılacak parametreler ve formüller MANET’ler için yapılan benzetim çalışmalarının incelendiği bir çalışmada [17] Tablo 6.1’de görüldüğü gibi belirtilmiştir. MANET-DEVS ortamındaki ağın ve senaryoların oluşturulmasında Tablo 6.1’deki parametreler dikkate alınmıştır.

Tablo 6.1. Benzetim senaryoları için ihtiyaç duyulan parametre ve formüller

Parametre	Açıklama	Formül
Benzetim alanı	Topolojinin kurulduğu alan	$w * h$
Düğüm yoğunluğu	Benzetim alanındaki düğüm yoğunluğu	$\frac{n}{w * h}$
Düğüm kapsama alanı	Düğümlerin kapsama alanlarının kapsamı	$\pi * r^2$
Footprint	Düğümlerin benzetim alanında kapsadığı alanın yüzdesi	$\frac{(\pi * r^2)}{w * h} * 100$
Maksimum yol	Kaynaktan hedefe paketin doğrusal olarak aldığı maksimum uzaklık	$\sqrt{w^2 + h^2}$
Ağ çapı	Birbirine en uzak iki nokta arasındaki minimum atlama sayısı	$\frac{\sqrt{w^2 + h^2}}{r}$
Komşu sayısı	Benzetim ve iletim alanına dayalı olarak komşu sayısı	$\frac{\pi * r^2}{\left(\frac{w * h}{n}\right)}$
w= genişlik, h= yükseklik, r= kapsama alanı, n= düğüm sayısı		



Şekil 6.3. Basit bir ağın deneysel çerçeve ile birlikte DEVS-Suite ekran görüntüsü

6.2.2.3. Senaryoların oluşturulması ve yapılan kabuller

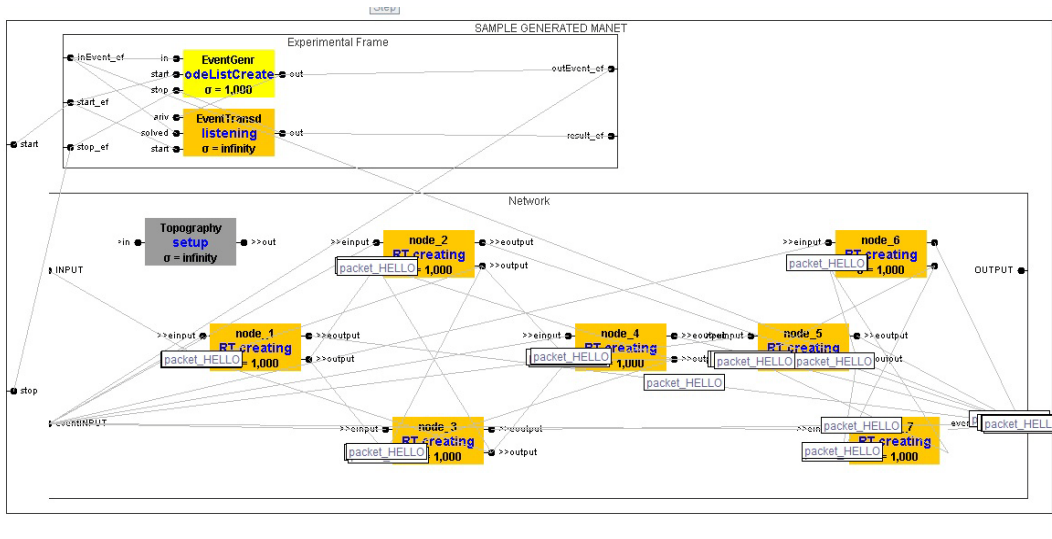
Yapılan çalışmada, ayrık olay yaklaşımının kullanılması nedeni ile herhangi bir düğümün ağa bir paket göndermesi ve bir bağlantının veya düğümün çalışmaz duruma gelmesi, ağda meydana gelen olaylar olarak tanımlandı. Benzetim sırasında ağda meydana gelen olayları önceden programlamak amacıyla kullanılan olay üretici detayları Bölüm 2’de verilen deneysel çerçevenin bir bileşenidir.

Daha öncede belirtildiği şekilde, ağı oluşturan bütün elemanlar olay üreticiyle doğrudan bağlı bir '>>input' girişine sahiptir. MANET-DEVS modelleme ve benzetim ortamında bütün bileşenler '>>input' portundan gelen mesajları diğer portlara göre daha farklı değerlendirir. Dolayısıyla, ağ çalışması sırasında herhangi bir link düğümü önceden programlanabileceği gibi; paket olayları da trafiği üretmek için kullanılabilir. Ağdaki trafiği oluşturan bütün kontrol paketleri düğümler tarafından, veri paketleride trafik üretici tarafından üretilir. Uygulamada, veri paketleri rastgele seçilmiş kaynak / hedef adresleri ile birlikte oluşturulur ve veri paketinin kaynak düğümünün '>>input' portuna gönderilir. Daha sonra paketin takip edeceği yol mevcut yönlendirme tabloları tarafından belirlenir ve paket ağ içerisinde hedefine doğru yoluna devam eder. Paket yol boyunca her bir düğümde mevcut kapasiteyi rastgele seçilmiş bir boyut kadar düşürür. Eğer paketin yolu üzerinde herhangi bir düğüm üzerinde kapasite tamamen doluysa bu paket atılır.

Paket hedefine vardığı zaman, hedef düğümün '>>output' portundan olay dönüştürücüye yollanır. Olay dönüştürücü, olay üreticinden aldığı verilerle paketin verilerini karşılaştırır ve sonuçları bir dosyada saklar. Paketin ağdan atıldığı durumlarda paket olay dönüştürücüye erişemeyeceği için kayıp paketlerin sayısı olay dönüştürücü tarafından kolayca belirlenebilir. Olay üretici tarafından üretilen trafik tek biçimli rastgele bir dağılımla üretilir.

6.2.2.4. Yönlendirme veritabanının kurulması

Ağı oluşturan düğümler birbirine bağlanıp ağ oluşturulduğunda, bileşenler buldukları ortamlar hakkında herhangi bir bilgiye sahip değildirler. Başlangıç tablolarını oluşturma işleminde, ağ içerisindeki bütün bileşenler ‘kurulum’ (initialization) aşamasındadır ve her bir düğüm komşu düğümlere AODV algoritmasına göre birer ‘HELLO’ mesajı gönderir.



Şekil 6.4. Basit ağda başlangıçta yönlendirme tablolarının oluşturulması.

Ağ içerisinde iletilen paketler düğümler tarafından değerlendirilerek, başlangıç yönlendirme tabloları oluşturulur. Şekil 6.4’te basit ağda bağlantılar üzerinde HELLO paketlerin iletilmesi ve başlangıç yönlendirme tablolarının oluşturulması görülmektedir. Düğüm, benzetimin gerçekleştiği süre boyunca her 1 sn’de bir HELLO mesajlarını periyodik olarak üretir. Düğümler bunun dışında ağda dolaşan diğer kontrol paketleri ve veri paketlerinden aldıkları bilgiler ile yönlendirme tablolarını ekleme / güncelleme işlemlerini sürdürürler.

6.2.3. Benzetim sonuçları

AODV yönlendirme protokolünün, geliştirilen MANET-DEVS ortamında Tablo 6.2’de verilen parametrelere göre benzetim deneyleri gerçekleştirildi. Benzetim sırasında, ağ içerisinde gerçekleşen olayları DEVS-Suite ortamının sunduğu imkânlar ve MANET-DEVS ortamına eklenen farklı özelliklerle görsel olarak takip etmek mümkün olmaktadır.

6.2.3.1. Benzetim parametreleri

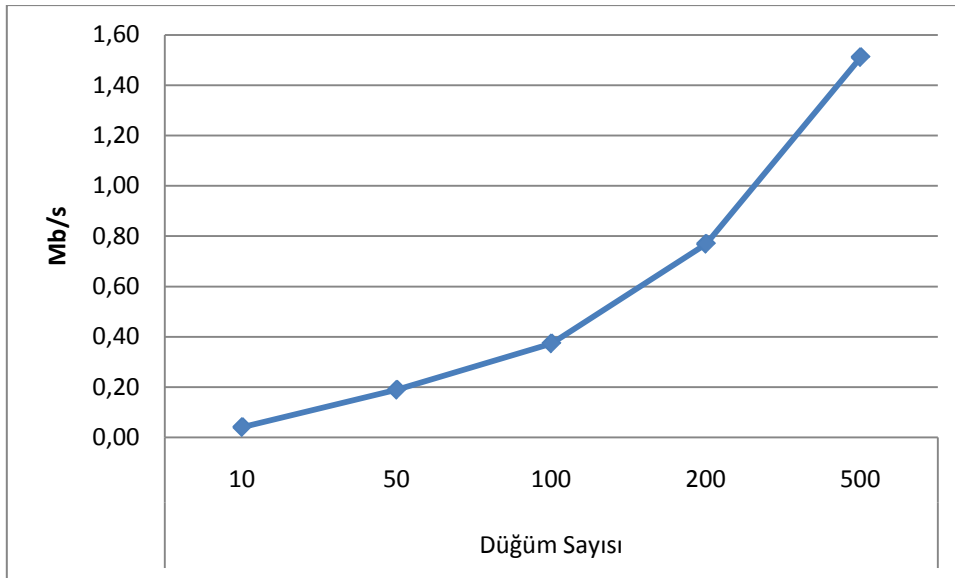
Gerçekleştirilen benzetim sürecinde, oluşturulan benzetim ortamına ait parametreler (istatistikler, bellek kullanımı, geçen süre, vb), elde edilen veriler ve ağda iletilen mesajlara ait bilgiler olay dönüştürücü tarafından yorumlanarak üretilen CSV (virgülle ayrılmış değerler - comma seperated values) dosyalarında kaydedilmektedir.

Tablo 6.2. Yapılan deneylerde uygulanan senaryo parametreleri.

Parametre	Düğüm Adedi				
	10	50	100	200	500
Alan (m)	500 X 500	1000 x1000	1500 x1500	2000 x 2000	3500 x 3500
Kapsama Alanı Çapı (m)	250				
Benzetim süresi (sn)	10				
Üretilen Mesaj Adedi / Düğüm Adedi	100				
Hız (m/sn)	0-10				
Hareket yönü	Rastgele				
Topoloji modeli	WAXMAN				
Kullanılan bilgisayar	Intel Core 2 Duo 2,93 Ghz 3GB RAM				
Platform	Windows 7 64 bit+ 64 bit Java Runtime Environment				

Yapılan deneyler sonucunda, gözlemlerden ve kaydedilen verilerden ağın durumu ve yönlendirme protokolünün başarımı ile ilgili bilgiler grafikler halinde dönüştürülerek aşağıda sunulmaktadır.

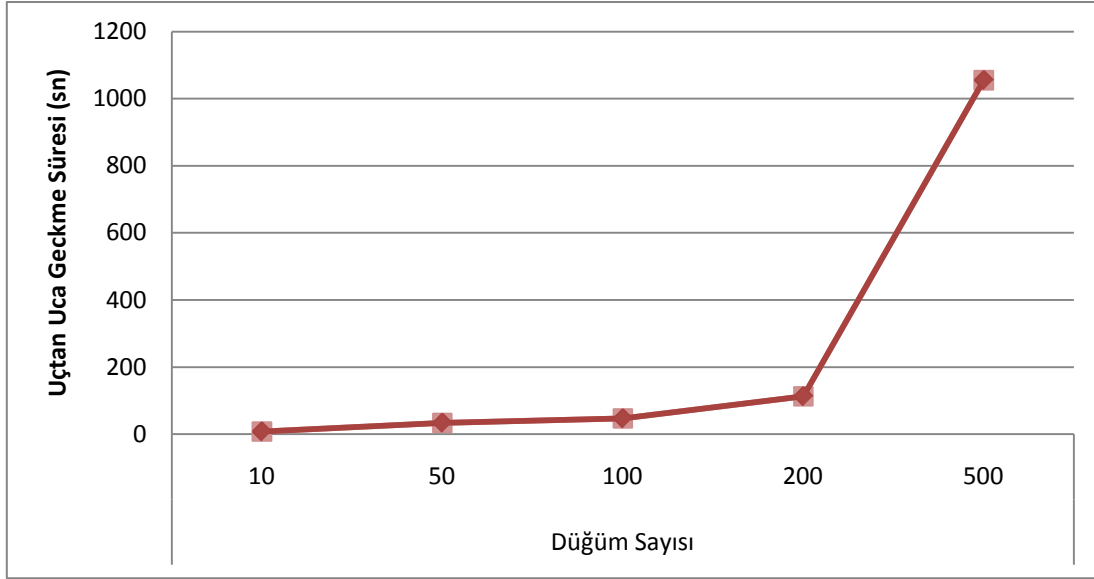
Yönlendirme algoritmalarının test edilmesi işleminde, bir birim zamanda ağ üzerinden geçen paketlerin sayısı (iş çıkarma yeteneği - throughput) önemli bir kriterdir. Farklı senaryolar altında AODV yönlendirme protokolünün uygulandığı ağa ait iş çıkarma yeteneği grafiği Şekil 6.5’de görülmektedir. Şekilde düğüm sayısı artarken iş çıkarma yeteneğinin de kademeli olarak arttığı görülmektedir. Bunun sebebi, farklı topolojilerde kullanılan trafiğinde farklı olmasıdır. Ağın işlem kapasitesinin artması iş çıkarma yeteneğinin de artmasına neden olmaktadır.



Şekil 6.5. İş çıkarma yeteneği grafiği

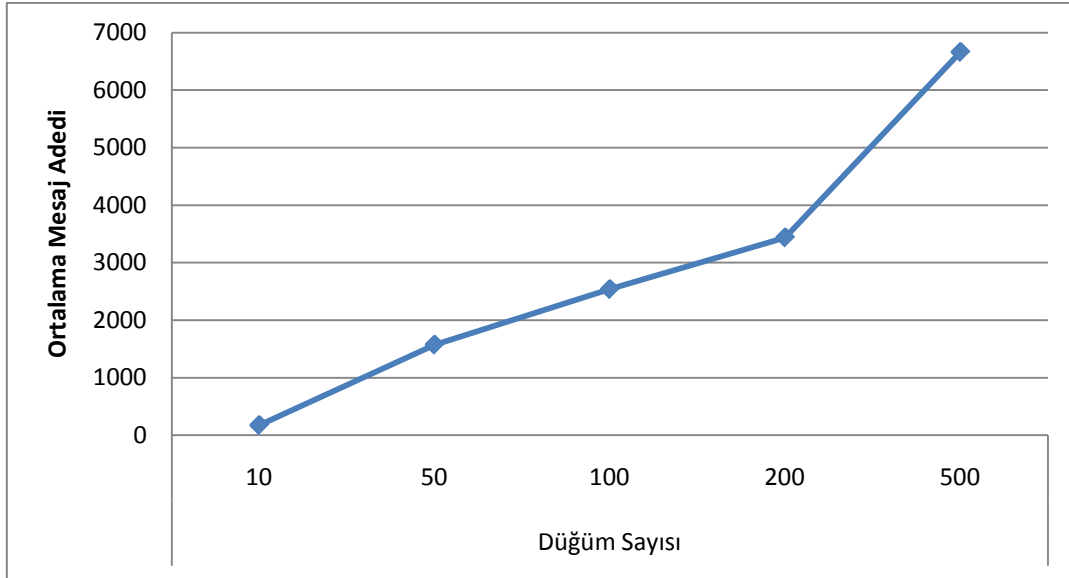
Ortalama paket gecikmesi ağlarda gözlemlenen ve ağın genel durumu hakkında bilgi sahibi olmamızı sağlayan diğer bir kriterdir. Şekil 6.6’da, ağdaki düğüm sayısı artarken ortalama gecikme süreside artmaktadır. Düğüm sayısı arttığında bir paketin hedefine ulaşması sırasında daha çok ara düğüm kat edeceği ve ağda iletilen mesaj sayısının artması, düğümlerin tamponlarında / kuyruklarında meydana gelen birikmeleri artıracığından paketin hedefe varma süresini de geciktirmektedir. Yapılan deneylerde ortalama paket gecikmelerine ait elde edilen sonuçlar Şekil

6.6'da görülmektedir. Artış 200 düğümlü senaryoya kadar normal, 200 den 500'e kadar ki aralıkta yüksek olmuştur. Bunun sebebi 500 düğümden sonra oluşan büyük ölçekli hareketlilik ve trafik şartlarının AODV'nin başarımı üzerinde önemli ölçüde olumsuz etkiye sahip olmasıdır. Ancak yine gecikme 1sn seviyesinde kabul edilebilir bir değerdedir.



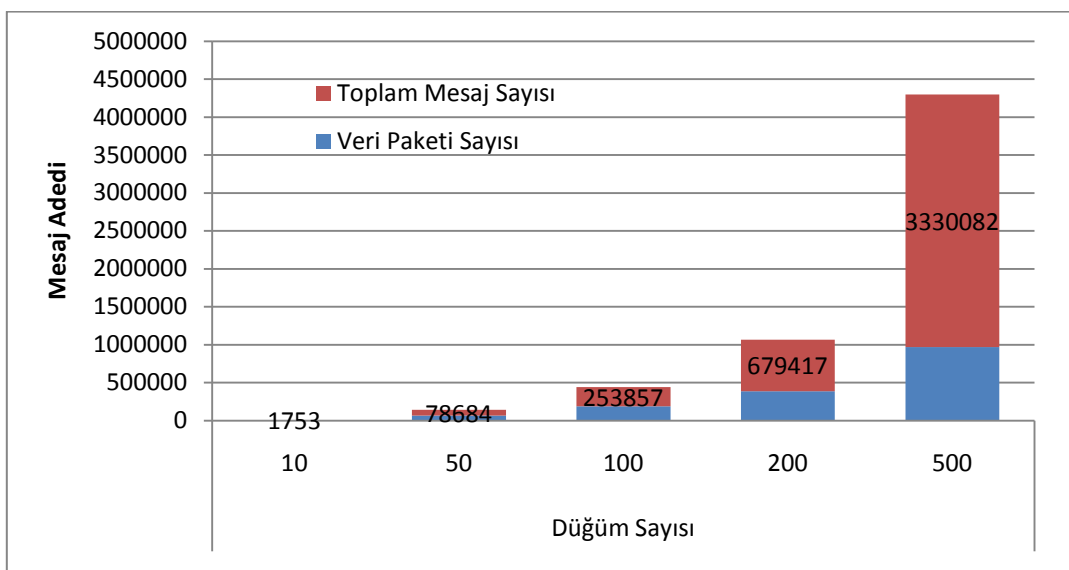
Şekil 6.6. Düğüm sayısı ve ortalama paket gecikmesi arasındaki ilişki

1 sn'de düğüm başına üretilen mesaj sayısı sabit olmak üzere benzetim süresince düğüm sayısındaki artış ile düğüm başına düşen mesaj trafiğindeki artış Şekil 6.7'de görülmektedir. Mesaj trafiğindeki artış, mesajın hedefe ulaşması için gereken sekme (hop) sayısının artması ve kullanılan yönlendirme protokolünün kullandığı kontrol mesajlarının çok yönlü (broadcasting) olarak yayılması ile açıklanabilir.



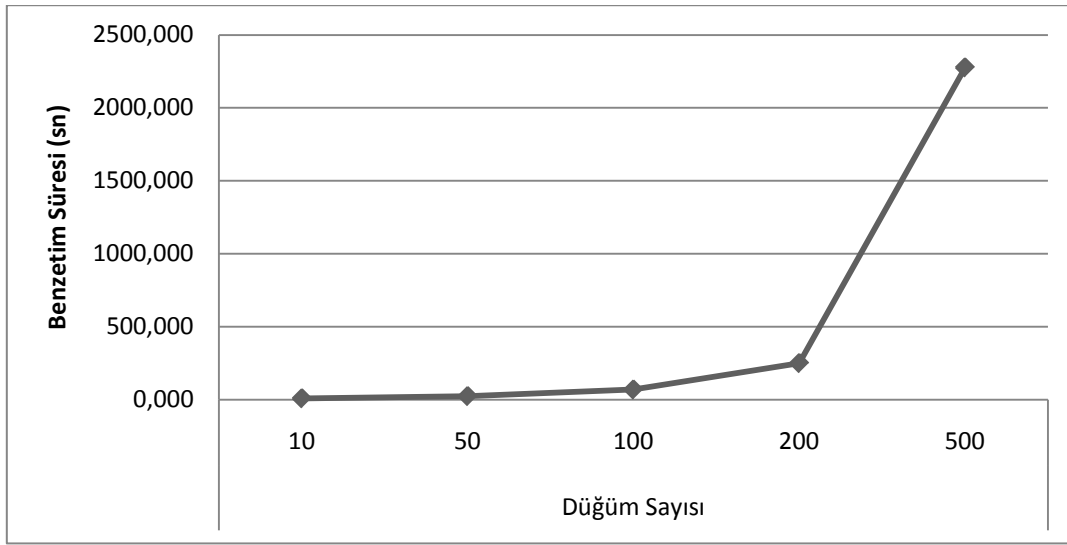
Şekil 6.7. Düğüm ve mesaj sayısı arasındaki ilişki

Düğüm sayısındaki artış ile ağ trafiğindeki artışın sebebini gösteren diğer bir veri ise ağda dolaşan veri paketlerinin tüm mesajlara oranıdır. AODV yönlendirme protokolünün karakteristik özelliği veri göndermeden önce yol keşfi yapılmasıdır ve bu durum ağ trafiğinde artışa sebep olmaktadır. Şekil 6.8’de bu durum grafik halinde görülmektedir. Ancak, kontrol mesajlarının toplam boyutu veri paketlerinin toplam boyutunun çok altında kalmaktadır.



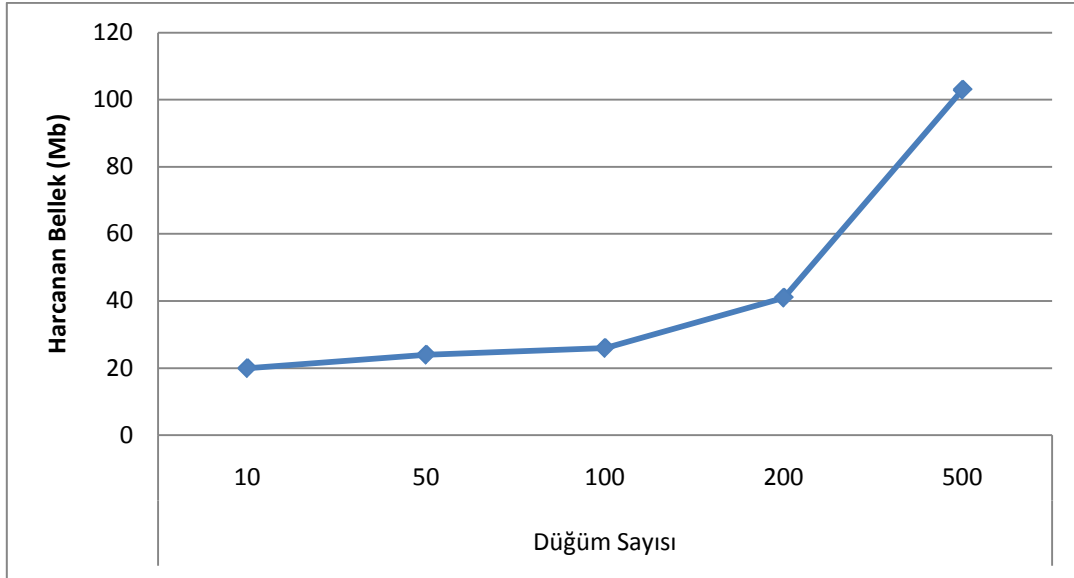
Şekil 6.8. Ağ trafiği ve veri paketi trafiği arasındaki ilişki

Benzetimin yapıldığı bilgisayarın kaynaklarının verimli kullanılması yapılan deneylerdeki ölçeklenebilirlik için önemlidir. Yapılan deneylerde 10 sn süresince her saniye düğüm başına 10 mesaj olacak şekilde trafik oluşturulmuştur. Düğüm başına üretilen mesaj sabit kalmak üzere düğüm sayısındaki artış ile ağ trafiğinde meydana gelen değişim yukarıda görülmektedir. Oluşan bu trafiğin duvar saati zamanı cinsinden yansıması Şekil 6.9'da görülmektedir.



Şekil 6.9. Düğüm sayısı ve benzetim süresi arasındaki ilişki

Ölçeklenebilirliği etkileyen diğer bir faktör de bellek kullanımudur. Rastgele topoloji nedeniyle düğümler arası oluşan bağlantı miktarı ve veri trafiğindeki değişiklik, kullanılan bellek miktarının anlık olarak değişmesine sebep olmaktadır. Yapılan deneylerde MANET-DEVS ortamının kullandığı bellek miktarı Şekil 6.10'da görülmektedir.



Şekil 6.10. Bellek kullanımı ve düğüm sayısı arasındaki ilişki

Sonuç olarak, AODV yönlendirme algoritması uygulamasında benzetim süresince bütün ağ kaynaklarının dengeli bir şekilde kullanıldığı ve yönlendirme tablolarının tutarlı olduğu gözlemlenmiştir. Yukarıdaki grafiklerden kapasite ve kaynakların kullanımının son derece dengeli olduğu anlaşılmaktadır. Grafikler MANET-DEVS ortamının bir ağ benzeticisinin bütün fonksiyonlarını yerine getirdiğinin göstergesidir. Bu sebeple, sonuçlar bir ağ benzeticisi olarak tasarlanan MANET-DEVS modelleme ve benzetim ortamının geçerlenmesi ve doğrulanması işleminin birer parçalarıdır. Ancak aşağıdaki bölümde MANET-DEVS'in geçerleme ve doğrulama süresi ayrı olarak ele alınmıştır.

Geliştirdiğimiz model ölçeklenebilir, adaptif ve sağlam gezgin ağ uygulamalarının modellenmesi ve tasarımı için örnek bir çerçeve olmuştur.

6.3. MANET-DEVS Doğrulama ve Geçerleme Deneyleri

Benzetim modeli herhangi bir hedefe (tasarım, analiz, kontrol, optimizasyon vb.) ulaşmak için bir araçtır. Modelleme ve benzetim (araçları) aracılığıyla yapılan çıkarımların ve elde edilen sonuçların rahatlıkla kabul edilebilmesinin güvence altına

alınması gerekmektedir. Bu güvenin kurulabilmesi için iki ayrı çalışma gerçekleştirilebilir: doğrulama ve geçerleme [48].

Doğrulama bilgisayar ortamına dönüştürülmüş model tasarımının istenen seviyede olup olmadığının kontrol işlemidir. Geçerleme ise eldeki amaçlar için modelin yeterli doğrulukta olup olmadığının kontrol işlemidir. Hiçbir modelin % 100 doğru olması mümkün değildir ve bu durumun geçerli sebepleri vardır. [55].

Bölüm 2’ de detayları verilen geçerleme, modelleme ve benzetim yaşam sürecinde gerekli bir adımdır. Yapılan çalışmada geçerleme süreci aşağıda detaylandırılmaktadır.

6.3.1. Hedeflerin belirlenmesi

MANET-DEVS benzeticisi geliştirilirken ana hedefler aşağıda sıralanmıştır;

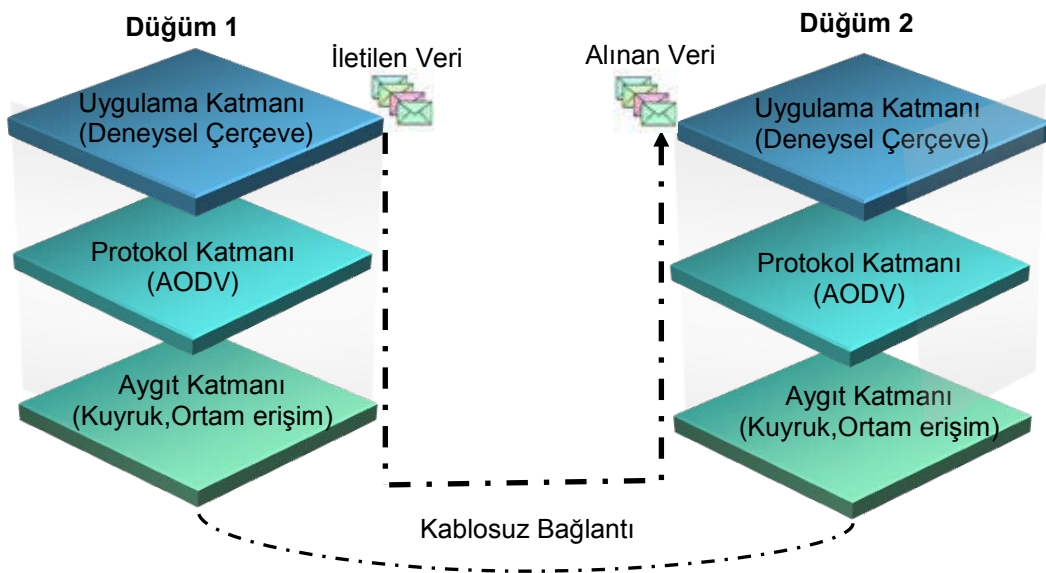
- Sistem teorisine dayalı bir tasarım
- Modüler ve hiyerarşik bir yapı
- Kolay kurulum,
- Yüksek performans ve ölçeklenebilirlik
- Yüksek izlenebilirlik
- İyi görsel takip
- Geniş test çerçevesi
- Platform bağımsız
- HLA yapısına uygun

bir benzetici geliştirilmesi hedeflenmiştir.

6.3.2. MANET-DEVS kavramsal model geçerlemesi

Kavramsal model geçerieme; oluşturulan kavramsal modelin teori ve varsayımlarının sistem teorisi ile uyumlu olup olmadığını ve sistemin model gösteriminin başlangıçta belirlenen amaçları gerçekleştirmek için makul olup olmadığını belirlemektir.

Geliştirilen benzeticinin bir MANET'in gerçek dünyadaki davranışını veya teorik tanımlamasını karşılaması beklenir. Benzeticinin 7 katmandan oluşan OSI standartına göre çalışması hedeflenmiştir. Şekil 6.11'de OSI referans modelinin 3 katmanda özetlenmiş hali ve geliştirilen benzeticide buna karşılık gelen birimler görülmektedir.



Şekil 6.11. MANET-DEVS ağ katmanları yapısı

6.3.3. Model davranış geçerlemesi

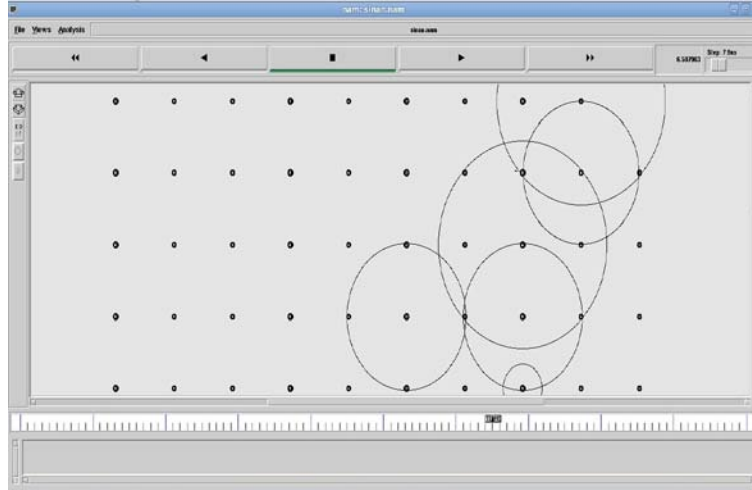
Modelin davranış geçerlemesi; oluşturulan kavramsal modelin belli bir bilgisayar sistemi üzerinde programlama ve uygulama için yazılım tasarımının ve özelliğinin yeterli olduğunun belirlenmesi olarak tanımlanır.

Bilindiği gibi gerçek ortamda her bir düğüm bağımsızdır ve kendine has özellikler barındırır. Bu özelliklerin kimisi dışarıdan görülebilirken kimisinde sadece düğüm tarafından kontrol edilir. Benzetim ortamında bunu sağlamak için nesneye dayalı programlama dilinin önemli özelliğinden biri olan ‘sarma’ (encapsulatin) özelliğinden faydalanılmıştır. Sarma özelliği sadece düğümler için değil aynı zamanda ağda hareket eden veri paketleri gibi varlıklar için de kullanılmıştır. Böylece modeli oluşturan katmanlarda mesaj paketlerinin birbirinden bağımsız olarak hareket etmesi sağlanmıştır.

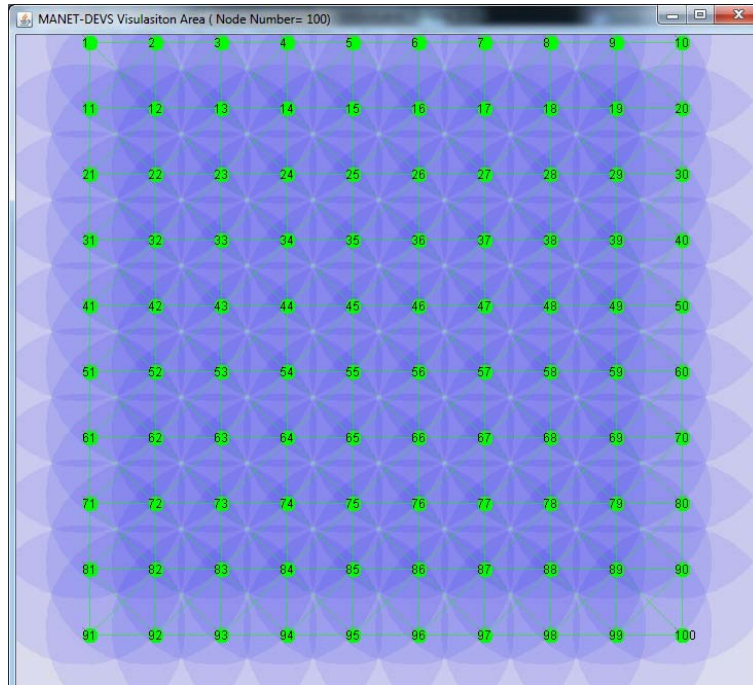
Düğümlerin iletişimde IP adreslerinden faydalanılmıştır. Her bir düğüm sahip oldukları farklı IP adresleri ile gerçek bir ağdaki ortam oluşturulmuştur.

Düğümlerin veri trafiğini nasıl yönetecekleri yönlendirme protokolleri ile sağlanmaktadır. Geliştirilen benzeticide önceki bölümlerde detayları verilen AODV protokolü kullanılmıştır.

Modelin davranışı geçerlemesi için gerçek dünya verisinin kullanılmasındaki güçlükler nedeniyle çok yaygın olarak kullanılan Ns-2 benzetim aracı kullanılmıştır. MANET-DEVS ve Ns-2 benzeticileri için mümkün olduğunca benzer senaryolar ve parametreler kullanılmıştır. Bu benzeticilerde oluşturulan topolojilerin ekran görüntüleri Şekil 6.12 de görülmektedir.



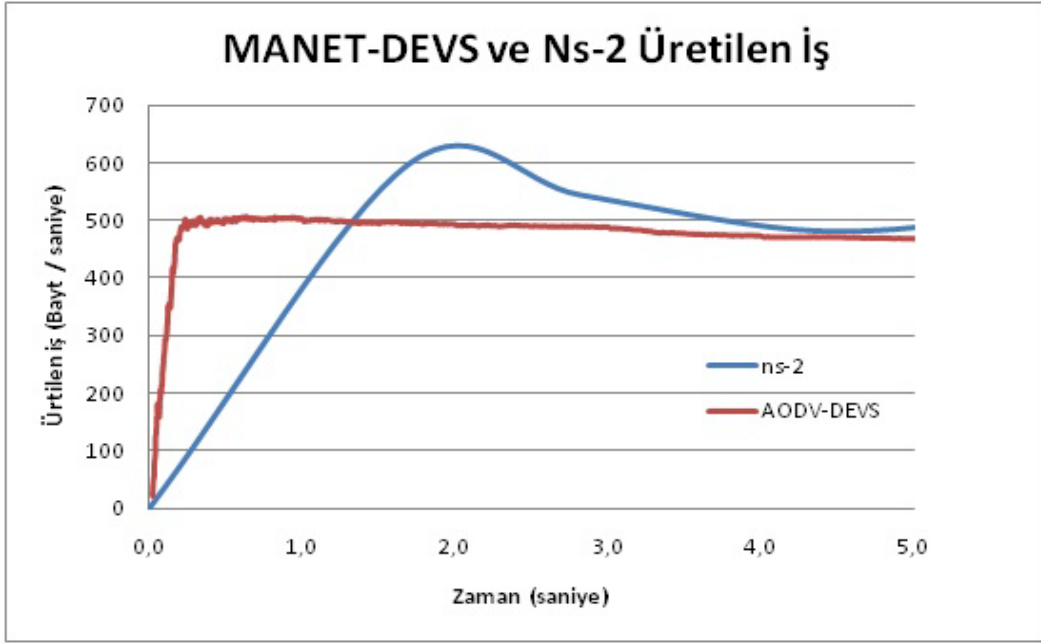
Ns-2 nam editorü ekran görünümü



MANET- DEVS topoloji takip ekranı

Şekil 6.12. Ns-2 ve MANET-DEVS topoloji ekran görüntüleri

Her iki benzeticide benzer şartlarda yapılan benzetim deneyleri sonucunda elde edilen iş çıkarma yeteneği grafiği Şekil 6.13' de görülmektedir. İki benzeticinin kullandıkları rastgele sayı üreticileri göz önüne alındığında benzeticilerin çıkışlarının hemen hemen birbirine yakın olduğu görülmektedir.



Şekil 6.13. Ns-2 ve MANET-DEVS iş çıkarma yeteneklerinin karşılaştırma grafiği

BÖLÜM 7. SONUÇLAR VE DEĞERLENDİRME

7.1. Sonuçlar

DEVS modelleme ve benzetim yaklaşımının paralel sistemler için geliştirilen 'Paralel DEVS', modern hesaplama teorisinde çok önemli bir yere sahip olan paralelliğin gerçekleşmesine imkân sağlamaktadır. Paralel DEVS'in sunduğu bu imkân, bilgisayarların sıralı çalışan mimarilerinden kaynaklanan sınırlandırmaları ortadan kaldırmakta ve bilgisayar ağları gibi yüksek derecede dinamik ve paralel çalışan sistemlerin modellenmesi ve benzetimini mümkün kılmaktadır. DEVS sağladığı semantikler ve ifade biçimleriyle yüksek dinamizme sahip bileşenlerin tanımlanmasını ve yazılım olarak gerçekleşmesini kolaylaştırmıştır

Bu çalışmada Paralel DEVS mimarisi ile oluşturulmuş DEVS-Suite modelleme ve benzetim ortamında Java programlama dili kullanılarak, MANET sistemler üzerinde modelleme ve benzetim çalışmalarına imkân veren bir benzetici geliştirilmiş ve geliştirilen bu benzetici üzerinde modelleme ve benzetim uygulaması gerçekleştirilmiştir. Bir MANET'i oluşturan, dağıtık bir yapıya sahip olan bileşenler DEVS modelleme ve benzetim yaklaşımı kullanılarak modellenmiş, elde edilen modelleme ve benzetim ortamına 'MANET-DEVS' adı verilmiştir. Hazırlanan MANET-DEVS modelleme ve benzetim çalışmasının doğrulama ve geçerlemesine yönelik benzetim çalışmaları gerçekleştirilmiştir.

MANET-DEVS benzeticisinin bilime katkılarını aşağıdaki şekilde sıralayabiliriz;

- Sistem Teorisi Tabanlılık: MANET'ler gibi dağıtık karmaşık ve son derece dinamik sistemler için gerekli olan, sistem teorisi tabanlı modüler ve hiyerarşik bir benzetici geliştirilmiştir. Bölüm 1'de avantajları / dezavantajları verilen ağ

benzeticilerinin çoğu formalize değildir. Herhangi bir formal tanımdan yoksun bir benzeticici düşük başarıma kötü bir ölçeklenebilirliğe ve yeniden kullanılmayan yazılım mimarisine sahiptir. Benzeticici tasarımının formalizasyonu, sistem tasarımının uygun bir şekilde onaylanmasını ve geçerlemesini kolaylaştırmaktadır

- Yeniden Kullanılabilirlik: DEVS modelleme yaklaşımına sahip, model geliştirme zamanı düşük ve tasarımın erken safhalarında kritik sistem problemlerinin ortaya çıkarılmasını mümkün kılan bir benzeticici tasarlanmıştır. Sahip olduğu nesneye yönelik 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. MANET-DEVS ortamını oluşturan bileşenlerin ve varlıkların, modüler bir yapıda tasarımını, yeniden kullanımını olanaklı kılmaktadır.
- Sonuçları Yorumlama Kolaylığı: MANET-DEVS ortamının matematiksel altyapısı durum değişkenlerinin değişimlerine ve sabit-parçalı bir formatta olan grafiklerin üretilmesine imkân sağlamaktadır. Böylece geliştirilen MANET-DEVS benzeticisi ile yapılan benzetim çalışmalarını gözlemlenmek / araştırılmak istenen MANET sisteminin karakteristikleri hakkında bilgi sağlayacak sonuçların üretilmesi ve elde edilen sonuçların değerlendirilmesi / yorumlanması kolaylaştırılmıştır.
- Platform Bağımsız: MANET-DEVS benzeticisi, tamamen nesneye yönelik programlama dili ile zeki bileşenlerin kodlanmasını kolaylaştırmakta, platformdan bağımsız kolay kurulabilen (örneğin Ns-2 benzeticisi iki programlama dili altyapısına sahiptir ve benzeticici paketi sadece Linux işletim sistemi için geliştirilmiştir) bir benzeticici altyapısına sahiptir.
- Uzaktan Eğitim Desteği: MANET-DEVS “java webstart” teknolojisi ile İnternet üzerinden yayınlanabilmesini / kullanılmasını sağlamakta ve İnternet kullanılarak

uygulamaların paylaşılmasını, uzaktan eğitim gibi günümüzün ileri teknolojileri için elverişli bir altyapı oluşturmaktadır. Bölüm 1’de ifade edilen benzeticiler arasından, sadece Jsim benzeticisinin bu özelliği kısmen desteklediği ifade edilmişti. Geliştirilen benzetici modelleme ve benzetim biliminin uygulamalı olarak kavranmasına ve öğrenilmesine katkı sağlamıştır.

- Esneklik: MANET-DEVS modelleme ortamı, özel amaçlı (domain specific) bir yazılım olmadığı için, son derece esnek bir mimariye sahiptir. İlave edilecek farklı bileşenlerle, geliştirilen benzeticiyi sadece ağ sistemlerini değil, aynı zamanda paralel ve dağıtık farklı sistemlerin modelleme ve benzetiminin yapıldığı bir benzetim ortamına dönüştürülebilir.
- Görsellik: MANET-DEVS ortamı sistemin ve modelin yapısal ve davranışsal durumunu, zaman tabanlı eğrilerle, mesaj animasyonlarıyla, hiyerarşik bileşen yapılarının ekranda adım adım takip edilmesine imkân sağlamaktadır. Bu da geliştirilen benzeticide kurulan modellerin, daha iyi izlenmesini / analizini mümkün kılarak iyi bir eğitim aracı olmasını sağlamaktadır (örneğin ns-2 kötü bir grafiksel kullanıcı arabirimine sahiptir).
- Gelişmiş İzleme ve Analiz Desteği: MANET-DEVS ortamı model üzerinde gelişmiş test ve deneylere imkân sunmaktadır. MANET’lerde, ağ protokollerinin daha iyi başarımların analizinin yapılabilmesi için, detaylı izleme son derece önemlidir. Benzetim sonuçları hem ‘log’ (günlük) dosyalarında, hem de ‘csv’ dosyalarında saklanabilmektedir. ‘Time view’ seçeneği, zaman üzerinde modellerin davranışının izlenmesini sağlar. Bölüm 6’da detaylandırıldığı gibi, geliştirilen benzetici, benzetim sonuçlarının kolay bir şekilde değerlendirilmesini mümkün kılmaktadır (örneğin, Pdns benzeticisi ürettiği ASCII dosyaları üzerinde Perl kodları çalıştırarak benzetici sonuçlarının yorumlanmasını güçleştirmektedir).
- Tümüleşik Topoloji Üretici Desteği: Topoloji üretici, MANET-DEVS ortamına entegre edilen, BRITE topoloji üretici ile kod yazmaksızın büyük ölçekli

topolojiler üretilebilmektedir, buda büyük ölçekli ağların kolayca kurulup analizine olanak tanımaktadır. BRITE topoloji üretici, otonom sistem güç kanunlarına dayanır ve gerçekçi modellerin oluşturulmasına imkân verir. Bölüm 6'da geliştirilen MANET-DEVS benzeticisinin başarımını ölçmek amacıyla, farklı ölçeklerde ağ modelleri oluşturulup, AODV yönlendirme algoritması test edilebilmiştir.

- Paralel ve Dağıtık Çalışabilme Yeteneği: DEVS / HLA teknolojisi ve Java RMI teknolojileri kullanılarak, modeller kolayca birden fazla işlemci 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 arabirimlerle başka ağlara bağlanarak hiyerarşik modüllere sahip bir ağ sisteminin tasarlanmasına imkan vermektedir.

Sonuç olarak; MANET-DEVS benzeticisi, bir MANET sistem üzerinde, MANET'lerin karakteristikleri üzerine, sağlıklı ve kolay benzetim çalışmaları yapılmasına imkân sağlamaktadır. Geliştirilen benzeticide, Java dilinin kullanılması nedeniyle, uygun işletim sistemi seçme, program yükleme / güncelleme vb. işlemlere gerek kalmamaktadır. Geliştirilen MANET-DEVS benzeticisi, ölçeklenebilir, adaptif ve güçlü gezgin ağ uygulamalarının modellenmesi / tasarımı için örnek bir çerçeve olmuş, DEVS metodolojisinin geniş bir uygulama alanına uygulanabilirliğini desteklemiştir.

7.2. Tartışma ve Öneriler

Geliştirilen MANET-DEVS modelleme ve benzetim ortamı, gezgin ağlarının yönetimi ve karakteristikleri ile ilgili konuları çalışmaya uygundur. MANET-DEVS modelleme ve 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

özellikleri kullanılarak detaylı / yeni uygulamalar geliştirilebilir, çalışmalar yapılabilir. Geliştirilebilecek uygulamaların / çalışmaların bazıları aşağıda sunulmaktadır.

MANET-DEVS benzeticisi, DEVS metodolojisini avantajlarını kullanarak bir MANET sistemi modellemek üzere geliştirilmiştir. Geliştirilen benzeticide bir MANET'in hareketlilik ve yönlendirme özelliği modellenmiş, diğer bazı özellikleri basit kabullenmeler ve soyutlamalara tabii tutulmuştur. Bu soyutlamaların gerçeğe uygun bir şekilde modellenmesi ile MANET'in tüm yönlerini değerlendiren eksiksiz bir MANET benzeticisi geliştirilebilir.

İnternetin ve mobil cihazların kullanımının her geçen gün artması yapılan benzetim çalışmalarında ölçeklenebilirliği önemli kılmaktadır. Sınırlı donanıma sahip bilgisayarlar ile milyonlar mertebesine ulaşan mobil cihazların oluşturduğu ağların benzetimini mümkün olmamaktadır. Çok büyük ölçekli MANET'leri modellemek için, ortaya çıkan iş yükünün birden fazla bilgisayara paylaştırarak paralel ve dağıtık modelleme ve benzetimi gerçekleştirecek 'Yüksek seviyeli yapı' (High Level Architecture - HLA) benzeticiler kullanılabilir. HLA teknolojisini kullanan benzeticiler büyük ölçekli bir ağı alt modellere bölüp farklı bilgisayarlarda çalışmasını ve elde edilen sonuçların bir merkezde toplanmasını sağlamaktadır.

MANET-DEVS benzeticisinin geçerlemesi ve doğrulaması MANET'lerde kullanılan bir yönlendirme protokolü ile yapılmıştır. Daha sağlıklı bir geçерleme ve doğrulama için geliştirilen benzeticinin soyutlama ve kabullenme seviyelerini günümüzde çok yaygın kullanılan benzeticiler ile eşit düzeye getirerek, eşit şartlarda yapılacak deneyler sonucunda, çıkan sonuçların karşılaştırması, geliştirilen benzeticinin doğrulama ve geçerlemesine katkı sağlayacaktır. Bu durum MANET-DEVS benzeticisinin kullanımının yangınlaşmasına sebep olacaktır.

Geliştirilen MANET-DEVS ortamı DEVS-Suite_{ws} (DEVS-Suite web start) ortamına taşınarak popüler tarayıcılar üzerinden modelleme ve benzetim çalışmalarına imkan sağlanabilir. Web tabanlı ve uzaktan eğitim çalışmalarında faydalanılabilir.

KAYNAKLAR

- [1] TAVLI, B., HEINZELMAN, W., Mobile Ad Hoc Networks Energy-Efficient Real-Time Data Communications, Springer, Netherlands, 2006.
- [2] JAYAKUMAR, G., GOPINATH, G., Ad Hoc Mobile Wireless Networks Routing Protocols – A Review, Journal of Computer Science, 3, 8, pp. 574-582, 2007.
- [3] CORSON, S., MACKER, J., Mobile Ad hoc Networking (MANET), Network Working Group, [http:// http://tools.ietf.org/html/rfc2501](http://tools.ietf.org/html/rfc2501), 2009.
- [4] BIRADAR, R.V., PATIL, V.C., SAWANT, S.R., MUDHOLKAR, R.R., Classification and Comparison of Routing Protocols in Wireless Sensor Networks, Ubiquitous Communication and Computing Journal, 4, pp. 704-711. 2009.
- [5] RAHMAN, M.A., PAKŠTAS, A., WANG, F. Z., Network Modelling and Simulation Tools, Simulation Modelling Practice and Theory, 17, pp. 1011-1031, 2009.
- [6] MALOWIDZKI, M., Network Simulators: A Developer's Perspective, Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'04), pp. 412-419, July 2004.
- [7] 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, http://www.acims.arizona.edu/EVENTS/ULN/ULN_doc2.pdf (addendum to the ULN Workshop 2001)
- [8] Network Simulator-2, <http://www.isi.edu/nsnam/ns/>, 2010.
- [9] Opnet Modeler, <http://www.opnet.com/products/modeler/home.html>, 2010.
- [10] 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.
- [11] ZENG, X., BAGRODIA, R., GERLA, M., GloMoSim: a Library for Paralel Large-scale Wireless Networks, Proceedings of the 12th Workshop on Paralel Simulations -- PADS '98, Banff, Alberta, Canada, 1998.

- [12] Parallel/Distribute NS, <http://www.cc.gatech.edu/computing/compass/pdns/>, 2010.
- [13] 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.
- [14] 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.
- [15] ZEIGLER, B.P., Multifaceted Modelling and Discrete Event Simulation. Academic Press, London, 1984.
- [16] DOW, C.R., LIN, P. J., CHEN, S.C., LIN, J.H., HWANG, S.F., A Study of Recent Research Trends and Experimental Guidelines in Mobile Ad-hoc Networks, Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05), 1, pp. 72-77, Taipei, Taiwan IEEE Computer Society.2005.
- [17] KURKOWSKI, S., CAMP, T., COLAGROSSO, M., MANET Simulation Studies: The Incredibles, Mobile Computing and Communications Review, 9, 4, pp 50-61, 2005.
- [18] HOGIE, L., BOUVRY, P., GUINAND, F., An Overview of MANETs Simulation, Electronic Notes in Theoretical Computer Science, 150,pp. 81–101, 2006.
- [19] ORFANUS, D., LESSMANN, J., JANACIK, P., LACHEV, L., Performance of Wireless Network Simulators: A Case Study, Proceedings of the 3rd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired Networks (PM2HW2N'08), pp. 59-66, 2008
- [20] BEN-EL-KEZADRI, R., KAMOUN, F., Towards Manet Simulators Massive Comparison and Validation, The 18th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC'07), Athens, Greece, 2007.
- [21] SHAUKAT, K., SARJOUGHIAN, H., A Comparative Study on DEVS and ns-2 Simulation and Modeling, Technical Report Arizona State University, Tempe, AZ, 2006 .
- [22] 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.

- [23] ANTOINE-SANTONI, T., SANTUCCI, J.F., DE GENTILI, E., COSTA, B., Discrete Event Modeling and Simulation of Wireless Sensor Network Performance, *SIMULATION*, 84, 2/3, pp. 103-121, 2008,
- [24] FAROOQ, O., WAINER, G., BALYA, B., DEVS Modeling Of Mobile Wireless Ad Hoc Networks, *Simulation Modelling Practice and Theory*, 15, 285–314, 2007.
- [25] KIESS, W., MAUVE, M., A Survey On Real-World Implementations Of Mobile Ad-Hoc Networks, Elsevier, *Ad Hoc Networks*, 5, pp. 324-339, 2007.
- [26] ABOLHASAN, M., WYSOCKI, T., DUTKIEWICZ, E., A Review of Routing Protocols For Mobile Ad Hoc Networks, ELSEVIER, *Ad Hoc Networks*, 2, pp. 1-22, 2003.
- [27] LIU, C., KAISER, J., A Survey of Mobile Ad Hoc network Routing Protocols, University of Magdeburg, TR-4, 2005.
- [28] KUOSMANEN P., Classification of Ad Hoc Routing Protocols, <http://www.netlab.tkk.fi/opetus/s38030/k02/Papers/12-Petteri.pdf>, 2010.
- [29] BIRADAR, R.V., PATIL, V.C., Classification and Comparison of Routing Techniques in Wireless Ad Hoc Networks, *Ad Hoc and Ubiquitous Computing*, 2006. ISAUHC '06. International Symposium on , vol., no., pp.7-12, 20-23 Dec. 2006.
- [30] HAERRI, J., BONNET, C., On the Classification of Routing Protocols in Mobile Ad-Hoc Networks, Institut Eurecom, Department of Mobile Communications, Research Report RR-04-115, Sophia-Antipolis, France, August 25th, 2004.
- [31] JUNHAI, L., LIU, X., DANXIA, Y., A Research on Multicast Routing Protocols for Mobile Ad-hoc Networks, *Computer Networks*, 52, pp. 988–997, 2008.
- [32] MEGHANATHAN, N., Survey and Taxonomy of Unicast Routing Protocols for Mobile Ad Hoc Networks, *The International Journal on Applications of Graph Theory in Wireless Ad hoc Networks and Sensor Networks (GRAPH-HOC)*, 1, 1, pp. 1-21, December 2009.
- [33] JAYAKUMAR, G., GOPINATH, G., Performance Comparison of Two On-demand Routing Protocols for Ad-hoc Networks based on Random Way Point Mobility Model, *American Journal of Applied Sciences*, 5, 6, pp. 659-664, 2008.

- [34] CAMP, T., BOLENG, J., DAVIES, V., A Survey of Mobility Models for Ad Hoc Network Research, *Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2, 5, pp. 483-502, 2002.
- [35] SESAY, S., YANG, Z., HE, J., A survey on Mobile Ad Hoc Wireless Network , *Information Technology Journal*, 3, 2, pp.168-175, 2004.
- [36] SIVAKUMAR, N., KUMAR, J.S., Comparison of Dynamic MANET On-Demand Routing Protocol With Respect To Various Quantitative Performance Metrics, *IDT Workshop on Interesting Results in Computer Science and Engineering, IRCSE '09, Mälardalen University, Sweden*, 2009.
- [37] RASHEED, A., MOHAMMAD, K., Exploration and Comparison of Several AODV Implementations: A Survey, *Communications of the Arab Computer Society* 2,2, 2009.
- [38] YADAV, N.S., YADAV, R.P., Performance Comparison and Analysis of Table-Driven and On-Demand Routing Protocols for Mobile Ad-hoc Networks, *Journal of Network and Computer Applications*, 4, 2, pp. 110-118, 2008.
- [39] HONGSONGA, C., ZHENZHOUA, J., MINGZENGA, H., ZHONGCHUANA, F., RUIXIANGA, J., Design and Performance Evaluation of A Multi-Agent-Based Dynamic Lifetime Security Scheme For Aodv Routing Protocol, *Journal of Network and Computer Applications*, 30, pp. 145–166, 2007.
- [40] PERKINS, C.E., ROYER, E.M., DAS, S.R., MARINA, M.K., Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks, *IEEE Personal Communications*, pp. 16-28, February 2001.
- [41] BOUKERCHE, A., Performance Evaluation of Routing Protocols for Ad Hoc Wireless Networks, *Mobile Networks and Applications*, 9, pp. 333–342, 2004.
- [42] CAMPOS, G., ELIAS, G., Performance Issues of Ad Hoc Routing Protocols in a Network Scenario Used for Videophone Applications, *System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on* , vol., no., pp. 321a- 321a, 03-06 Jan. 2005.
- [43] BROUSTIS, I., JAKLLARI, G., REPANTIS, T., MOLLE, M., A Comprehensive Comparison of Routing Protocols for Large-Scale Wireless MANETs, *Sensor and Ad Hoc Communications and Networks*, 2006. *SECON '06. 2006 3rd Annual IEEE Communications Society on*, pp. 951-956. 2006.
- [44] PACE, D.K., Modeling and Simulation Verification and Validation Challenges, *Johns Hopkins Apl Technical Digest*, 25, 2, pp. 163-172, 2004.

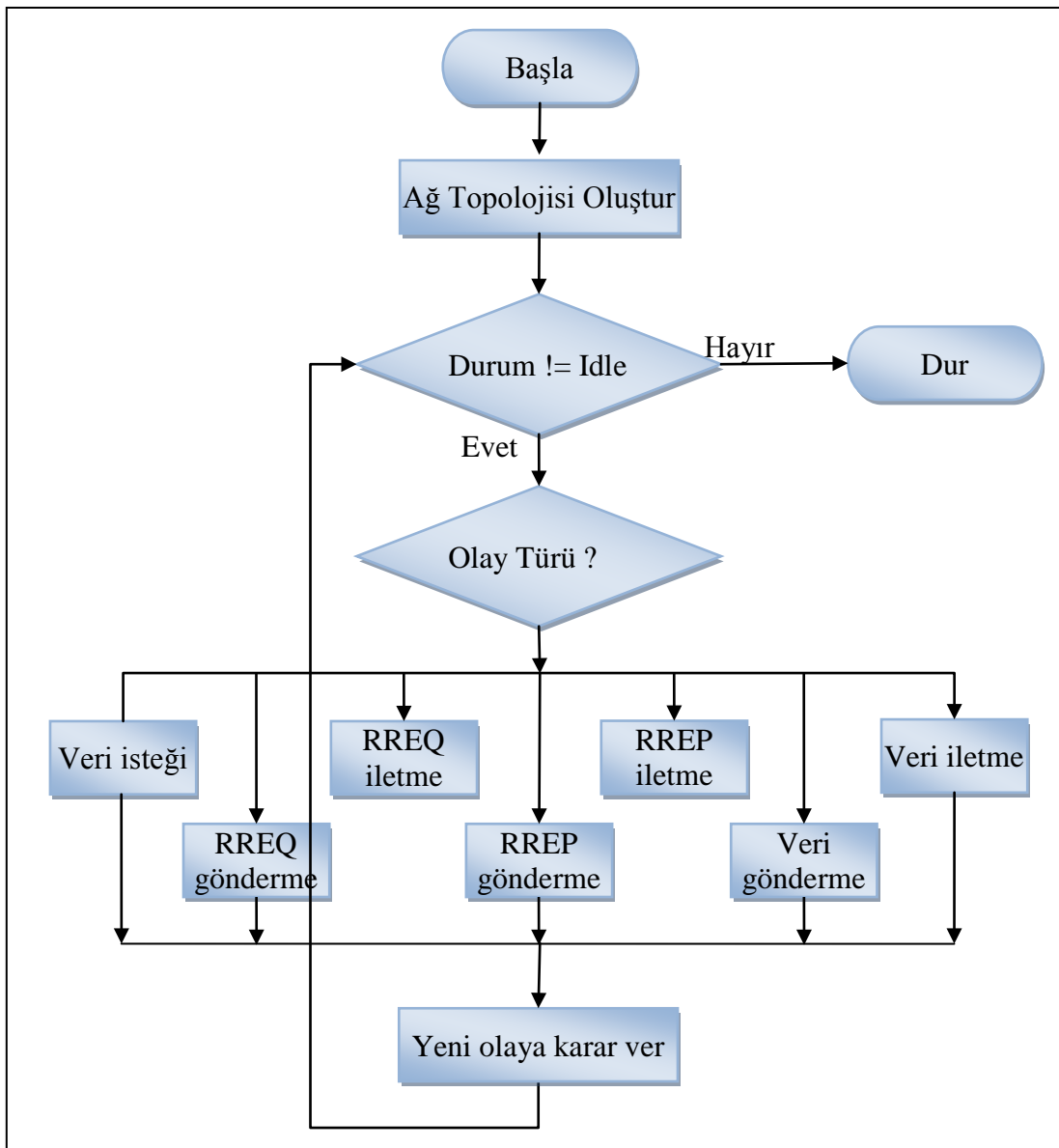
- [45] ZEIGLER, B.P., Theory of Modelling and Simulation. Robert E. Krieger, Malabar, Florida, 1984.
- [46] FIRAT, C., Simulasyon Sistemlerinde DEVS Tabanlı Kavramsal Modelleme ve Değerlendirme, Sakarya Üniversitesi, 2004.
- [47] MARIA, A., Introduction to Modeling and Simulation, Proceedings of the 1997 Winter Simulation Conference, pp. 7-13, 1997.
- [48] ZENGİN, A., Dağıtık Simulasyon Sistemleri İçin Yeni Bir Yönlendirme Algoritması ve Uygulaması, Doktora Tezi, Sakarya Üniversitesi, 2004.
- [49] VANGHELUWE, H., Multi-Formalism Modelling and Simulation, Doktora tezi, Universiteit Gent Faculteit Wetenschappen, 2001.
- [50] SINGH, V.P., System Modeling and Simulation, New Age International (P) Ltd., 2009.
- [51] KUHNE, T., What is a Model, Dagstuhl Seminar Proceedings 04101, <http://drops.dagstuhl.de/opus/volltexte/2005/23>, 2010.
- [52] BALCI, O., Principles And Techniques Of Simulation Validation, Verification, And Testing, Proceedings of the 1995 Winter Simulation Conference, pp. 147-154, 1995.
- [53] RYAN, J., HEAVEY, C., Process modeling for simulation, Computers in Industry, ELSEVIER, 57, pp.437-450, 2006.
- [54] DALE, K.P., Ideas About Simulation Conceptual Model Development, Johns Hopkins Technical Digest, 21, 3 , pp.327-336, 2000.
- [55] ROBINSON, S., Simulation Model Verification And Validation: Increasing The Users' Confidence, Proceedings of the Winter Simulation Conference, pp. 53- 59, 1997.
- [56] SOKOLOWSKI, J. A., BANKS, C.M., Principles of Modeling and Simulation, A John WILEY & SONS, INC., Publication, 2009.
- [57] SARGENT, R.G., Verification And Validation Of Simulation Models, Proceedings of the 2008 Winter Simulation Conference, pp.157-169, 2008.
- [58] CASSANDRAS, C.G., LAFORTUNE, S., Introduction to Discrete Event Systems, Springer, 2008.
- [59] HILD, D.R., Discrete Event System Specification (DEVS) Distributed Object Computing (DOC) Modeling And Simulation, Doktora Tezi, The University of Arizona, 2000.

- [60] ILYAS, M., The Handbook Of Ad Hoc Wireless Networks, CRC Press LLC, USA, 2003.
- [61] BOUKERCHE, A., Algorithms and Protocols for Wireless Sensor Networks, WILEY, 2009.
- [62] ROYER, E.M., A Review of Current Routing Protokols for Ad Hoc Mobile Wireless Networks, IEEE Personel Communications, pp. 46-55, April 1999.
- [63] AL-KARAKI, J.N., KAMAL, A.E., Routing Techniques in Wireless Sensor Networks: A Survey, IEEE Wireless Communications, pp 6-28, December 2004.
- [64] EL-AFANDI, H., HOSSEINI, H., VAIRAVAN, K., A Wireless Ad Hoc Protocols Comparison Study: DSR, AODV and IWAR, Journal of Computational Methods in Sciences and Engineering, 6, pp. 181-190, 2006.
- [65] PERKINS, C., BELDING-ROYER, E., DAS, S., Ad hoc On-Demand Distance Vector (AODV) Routing, <http://tools.ietf.org/html/rfc3561>, 2010.
- [66] CASSEL, R. A., PIDD, M., Distributed Discrete Event Simulation Using the Three-Phase Approach and Java, Simulation Practice and Theory, 8, pp 491-507, 2001.
- [67] REINHOLTZ, K., Java Will Be Faster Then C++, ACM SIGPLAN Notices, Volume 35, Issue 2, pp 25 – 28, 2000.
- [68] PIDD , M., CASSEL, RA., Using Java to Develop Discrete Event Simulations, Journal of the Operational Research Society, 51, 2000.
- [69] NICOL, D. M., Scalability of Garbage Collection in Java-Based Discrete-Event Simulators, Cnference on Computer Simulation, Cambridge England, 2003.
- [70] KULJIS, J., PAUL R.J., An Appraisal of Web-Based Simulation: Whither We Wander?, Simulation Practice and Theory, 9, pp37-54, 2001.
- [71] KILGORE, R. A., KLEINDORFER, G. B., The Future Of Java-Based Simulation, Winter Simulation Conference, pp 1707-1712, 1998.
- [72] www.acims.arizona.edu/SOFTWARE/devsjava.../devsjava-user-ref.pdf, 2010.
- [73] MITTAL, S., MAK, E., DEVS-Based Dynamic Model Reconfiguration and Simulation Control in the Enhanced DoDAF Design Process, The Society for Modeling and Simulation International, pp 239-267, Vol 3 Issue 4, 2006.

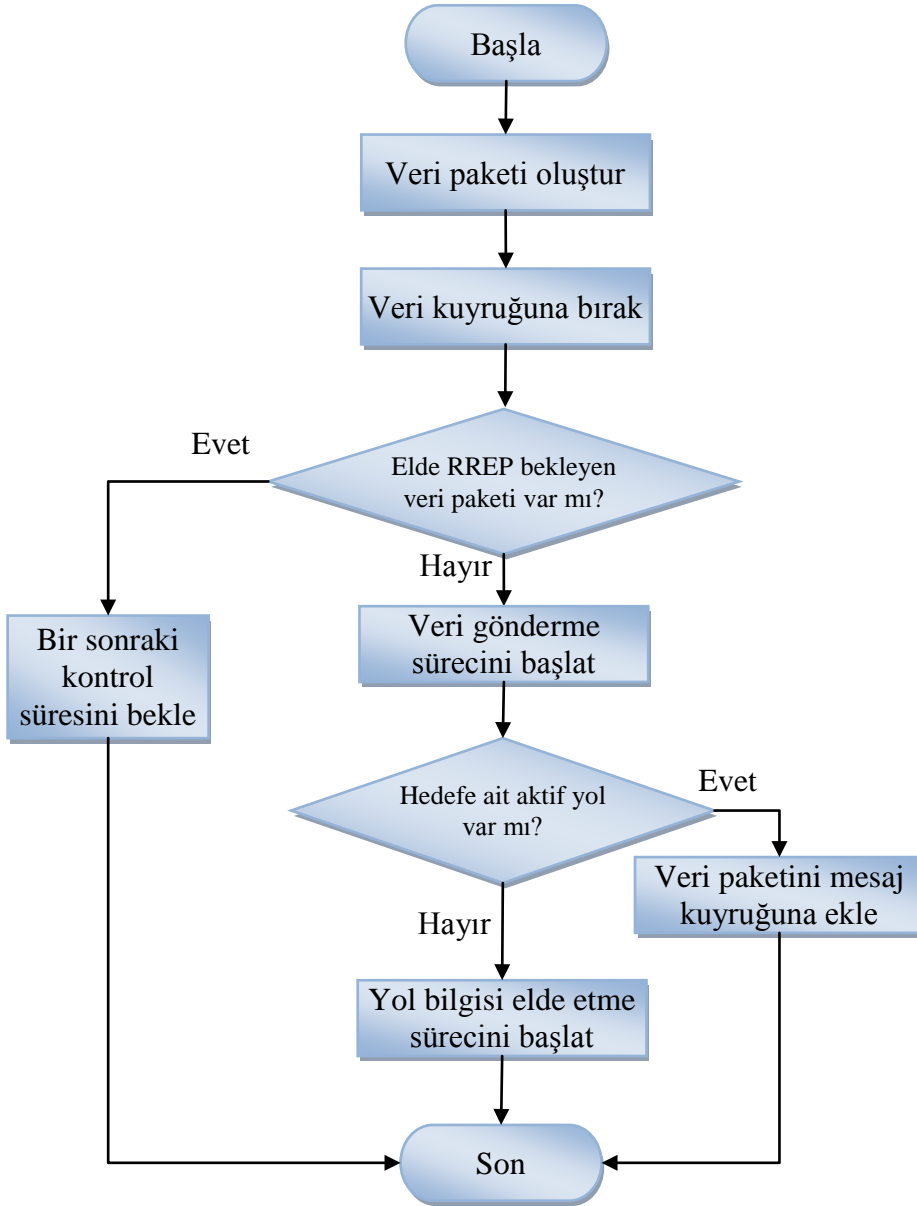
- [74] UHRMACHER, A. M., Dynamic Structures in Modeling and Simulation: A Reflective Approach, *ACM Transactions on Modeling and Computer Simulation*, Vol. 11, No. 2, pp 206–232, April 2001.
- [75] M. HEUSSE., GUERIN, S., SNYERS, D., KUNTZ, P., Adaptive Agent-Driven Routing and Load Balancing in Communication Networks. *Advances in Complex Systems*, 1:234–257, 1998.
- [76] RAHMAN, M. A., PAKŠTAS A., WANG F.Z., Network Topology Generation and Discovery Tools, *Simulation Modelling Practice and Theory*, Vol 17 Issue 6, pp 1011-1031, July 2009,
- [77] 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.

EKLER

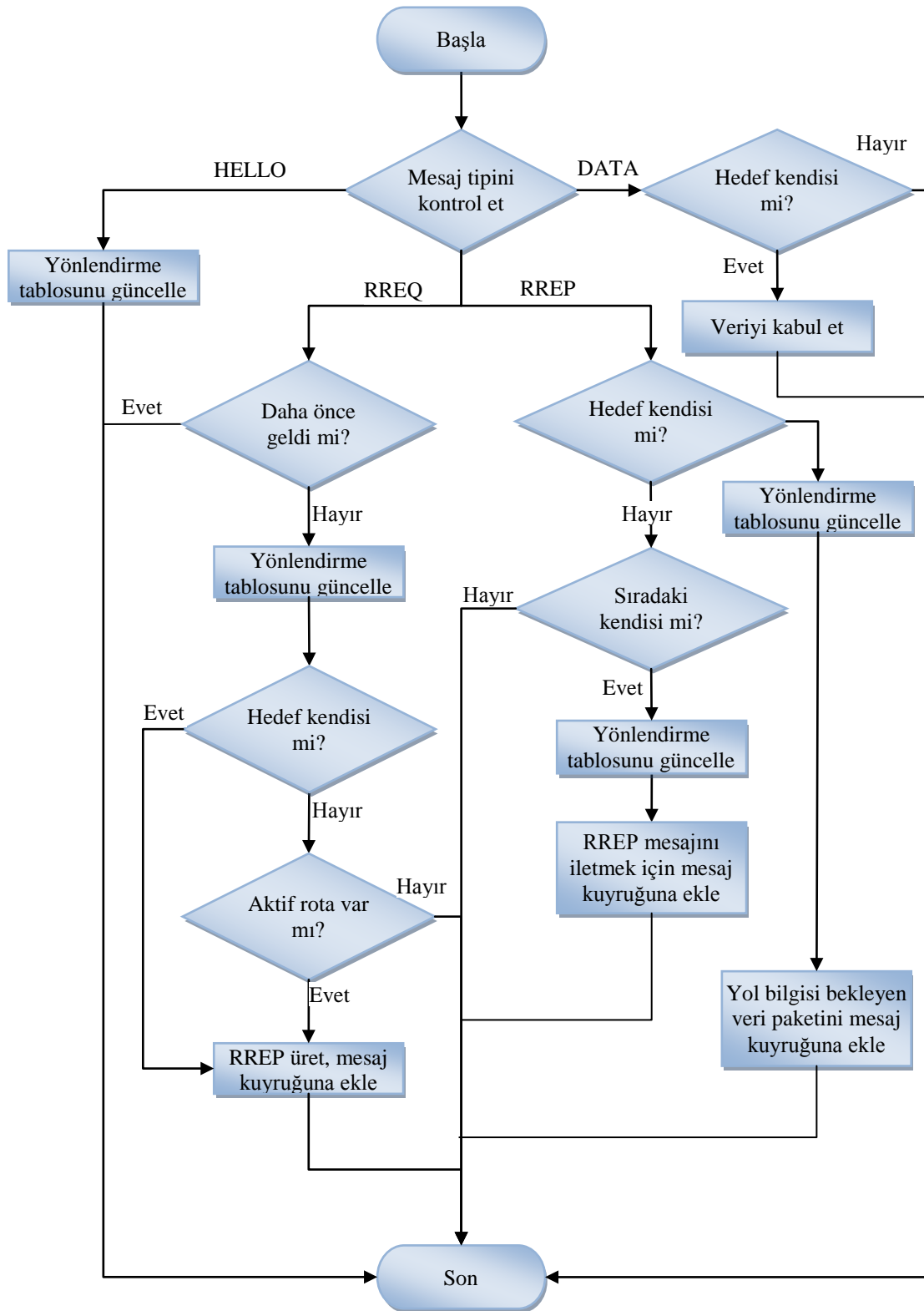
Ek A. MANET-DEVS Ortamında AODV Yönlendirme Protokolü Akış Şeması



Şekil A.1. Ana program akış şeması



Şekil A.2. Veri gönderme isteği akış şeması



Şekil A.3. Mesaj alma süreci akış şeması

Ek B. MANET-DEVS Dügüm Atomik Model Tanımı

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

Burada;

B.1. Giriş portları ve değerleri

$$X = \text{inport} \times \text{invalues}$$

invalues : {DATA, HELLO, RREQ, RREP },

inports : {>>input, >>einput}

B.2. Çıkış portları ve değerleri

$$Y = \text{outport} \times \text{outvalues}$$

outvalues : {DATA, HELLO, RREQ, RREP },

outports : {>>output, >>eoutput },

B.3. Durum kümeleri

$$S = \text{phase} \times \sigma \times Q$$

phase : {“idle”, “initialize”, “HELLO added”, “DT_request added”, “DATA added”, “RREQ added”, “RREP added”, “RT updating”, “DATA received”, “DATA Sending”, “DATA forwarding”, “RREQ sending”, “RREP sending”, “RREP bcs neighbour”, “RREQ forwarding”, “RREP forwarding”, “HELLO sending”, “DATA sent”, “DATA forwarded”, “RREQ sent”, “RREQ forwarded”, “RREP sent”, “RREP forwarded”}

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

$$Q = \text{queue} \times \text{queueDT}$$

Burada, kuyruk gelen ve gönderilen paketleri ve iletilecek veri paketlerini tutar

B.4. Harici geiş fonksiyonu

$\delta_{\text{ext}}(\text{phase}, \sigma, Q, e, X) =$

if packet=HELLO and queue has enough space
 enqueue the packet and $s \leftarrow (\text{"HELLO added"}, \sigma', x)$
 else $s \leftarrow (\text{"congested"}, \sigma', x)$

if packet = RREQ and RREQ history not include
 RREQ ID and address and queue has enough space
 enqueue the packet and $s \leftarrow (\text{"RREQ added"}, \sigma', x)$
 else if packet = RREQ and RREQ history include
 RREQ ID and address reject packet

if packet = RREP and RREP destination IP or next
 hop IP equal address and queue has enough space
 enqueue the packet and $s \leftarrow (\text{"RREP added"}, \sigma', x)$
 else $s \leftarrow (\text{"congested"}, \sigma', x)$

if packet = DATA and its (destination or next hop)=
 address and queue has enough space
 enqueue the packet and $s \leftarrow (\text{"DATA added"}, \sigma', x)$
 else $s \leftarrow (\text{"congested"}, \sigma', x)$

B.5. Dahili Geçiş Fonksiyonu

$\delta_{\text{int}}(\text{phase}, \sigma, Q) =$

```

if queue size = 0 s ← ("idle",  $\sigma'$ , x ) else dequeue packet

if packet=HELLO s ← ("RT updating",  $\sigma'$ , x )

if packet ="RREQ" and source  $\neq$  address
    s ← ("RREQ Sending ",  $\sigma'$ , x );
else if its source  $\neq$  address
    s ← ("RREQ Forwarding ",  $\sigma'$ , x );
else if its destination = address or routing table include route
    s ← ("RREP Sending" ,  $\sigma'$ , x );

if packet =RREP and its destination  $\neq$  address
    s ← ("RREP Forwarding ",  $\sigma'$ , x );
else if destination= address
    DATA enqueue

if packet=DATA and its source=address
    s ← ("DATA sending",  $\sigma'$ , x );
else if packet = DATA and its destination =address
    s ← ("DATA received",  $\sigma'$ , x );
else
    s ← ("DATA forwarding",  $\sigma'$ , x );

where s  $\in$  S
  
```

B.6. Confluent geçiş fonksiyonu

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

B.7. Çıkış fonksiyonu

$$\lambda(\text{phase}, \sigma, Q) = \left\{ \begin{array}{l} \text{if phase = "initialize" generate HELLO and output} \\ y \leftarrow (\gg \text{output, HELLO}) \text{ and } s \leftarrow (\text{"HELLO sending"}, \sigma', x); \\ \\ \text{if phase = "DATA Sending" } y \leftarrow (\gg \text{output, packet}) \\ s \leftarrow (\text{"Data sent"}, \sigma', x); \\ \\ \text{if phase = "DATA forwarding" } y \leftarrow (\gg \text{output, packet}) \\ s \leftarrow (\text{"Data forwarded "}, \sigma', x); \\ \\ \text{if phase = "RREQ sending" } y \leftarrow (\gg \text{output, packet}) \\ s \leftarrow (\text{"RREQ sent "}, \sigma', x); \\ \\ \text{if phase = "RREQ forwarding" } y \leftarrow (\gg \text{output, packet}) \\ s \leftarrow (\text{"RREQ forwarded "}, \sigma', x); \\ \\ \text{if phase = "RREP sending" or phase = "RREP bcs neighbour"} \\ y \leftarrow (\gg \text{output, packet}) \\ s \leftarrow (\text{"RREP sent "}, \sigma', x); \\ \\ \text{if phase = "RREP forwarding" } y \leftarrow (\gg \text{output, packet}) \\ s \leftarrow (\text{"RREP forwarded "}, \sigma', x); \\ \\ \text{where } y \in Y \end{array} \right.$$

B.8. Zaman ilerleme fonksiyonu

$$ta(s) = \sigma$$

ÖZGEÇMİŞ

Sinan Tüncel 1976 Sivrice / Elazığ doğumludur. İlk ve orta öğretimini İstanbul'da tamamladı. Lise öğretimine İstanbul Avcılar Endüstri Meslek Lisesi elektronik bölümünde devam etti. 1995 yılında girdiği Sakarya Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Eğitimi Bölümü Elektronik Programından 1999 yılında mezun oldu. Aynı yıl Sakarya Üniversitesi Bilgi İşlem Dairesi Başkanlığı Yazılım Gurubunda Araştırma Görevlisi olarak göreve başladı. 1999 yılında başladığı yüksek lisans eğitimini “Denetleyici Alan Ağı Endüstriyel İletişim Protokolünün Eğitim Amaçlı Benzetimi” isimli yüksek lisans tez çalışması ile 2002 yılında bitirerek bilim uzmanı unvanını aldı. 2003 yılında askerlik hizmetini OBI subayı olarak Balıkesir Astsubay Meslek Yüksek Okulunda yerine getirdi. 2004 yılında araştırma görevlisi olarak görevine Teknik Eğitim Fakültesi Bilgisayar Sistemleri Eğitimi Bölümünde devam etti. Üniversite destekli, ulusal ve Avrupa Birliği projelerinde görev aldı. Halen Sakarya Üniversitesi Teknik Eğitim Fakültesinde görevine devam etmektedir. Evli ve iki çocuk babasıdır.