T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MOBİL ROBOT SÜRÜLERİ İÇİN DİNAMİK YÖNLENDİRME ALGORİTMALARI

KHUDAYDAD MAHMOODI

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI

DANIŞMAN
YRD. DOÇ. DR. SIRMA ÇEKİRDEK YAVUZ

İSTANBUL, 2014

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

MOBİL ROBOT SÜRÜLERİ İÇİN DİNAMİK YÖNLENDİRME ALGORİTMALARI

Khudaydad MAHMOODI tarafından hazırlanan tez çalışması 17.06.2014 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Yrd. Doç. Dr. Sırma Çekirdek YAVUZ

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Yrd. Doç. Dr. Sırma Çekirdek YAVUZ

Yıldız Teknik Üniversitesi _____

Yrd. Doç. Dr. Mehmet Fatih AMASYALI

Yıldız Teknik Üniversitesi _____

Yrd. Doç. Dr. Umut Engin AYTEN

Yıldız Teknik Üniversitesi _____

Bu çalışma, Türkiye Bilimsel ve Teknolojik Araştırma Kurumu TÜBİTAK (EEEAG-113E212) ve Yıldız Teknik Üniversitesi Bilimsel Araştırma Projeleri Koordinatörlüğü'nce (2013-04-01-KAP01) desteklenmiştir.

ÖNSÖZ

Bu tez çalışmasının başından sonuna kadar her aşamasında, konu seçiminden içeriğin belirlenmesine kadar benden yardımlarını esirgemeyen, beni yönlendiren ve motive eden değerli hocam ve danışmanım Yrd. Doç. Dr. Sırma Çekirdek YAVUZ'a sonsuz teşekkürlerimi sunarım. Ayrıca, tez çalışma süresince bana yardım ettiği ve yönlendirdiği için Arş. Gör. Dr. Muhammet BALCILAR ve Yrd. Doç. Dr. M. Fatih AMASYALI hocalarıma da içtenlikle teşekkür ederim.

Hayatımın her anında olduğu gibi, tez çalışmalarım sırasında da anlayışları, sevgileri ve sabırlarıyla beni destekleyen sevgili aileme çok teşekkür ederim.

Haziran, 2014

Khudaydad MAHMOODI

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ vii

KISALTMA LİSTESİ viii

ŞEKİL LİSTESİ ix

ÇİZELGE LİSTESİ x

ÖZET xi

ABSTRACT xiii

BÖLÜM 1

GİRİŞ 1

1.1 Literatür Özeti 1
1.2 Tezin Amacı 2
1.3 Hipotez 3

BÖLÜM 2

KABLOLU VE KABLOSUZ AĞLAR 4

2.1 Kablolu Ağlar 4
2.1.1 Kablolu Ağların Avantajları 5
2.1.2 Kablolu Ağların Dezavantajları 5
2.2 Kablosuz Ağlar 5
2.2.1 Kablosuz Ağların Avantajları 6
2.2.2 Kablosuz Ağların Dezavantajları 7

BÖLÜM 3

GEZGİN TASARSIZ AĞLAR 8

3.1 Altyapılı Kablosuz Ağlar 8
3.2 Altyapısız Kablosuz Ağlar 9
3.3 Karma Kablosuz Ağlar 10

3.4 MANET'de Yönlendirme Protokollerinin Sınıflandırılması 11
3.4.1 Proaktif Yönlendirme Protokolleri 11
3.4.2 Reaktif Yönlendirme Protokolleri 12
3.4.3 Mesafe Vektörü Yönlendirme 12
3.4.4 Bağlantı Durumu Yönlendirme 13
3.5 Proaktif ve Reaktif Yönlendirme Protokollerinin Karşılaştırılması 13
3.6 Gezgin Tasarsız Ağlardaki Problemler 13

BÖLÜM 4

YÖNLENDİRME PROTOKOLLERİ 15

4.1 Proaktif Yönlendirme Protokolleri 15
4.1.1 DSDV (Destination Sequenced Distance Vector) 15
4.1.2 WRP (Wireless Routing Protocol) 16
4.1.3 CGSR (Clusterhead Gateway Switch Routing Protocol) 17
4.2 Reaktif Yönlendirme Protokolleri 17
4.2.1 DSR (Dynamic Source Routing) 17
4.2.2 AODV (Ad hoc On-demand Distance Vector) 18
4.2.3 TORA (Temporally Ordered Routing Algorithm) 19
4.2.4 ABR (Associativity Based Routing) 19
4.3 Yönlendirme Algoritmaları 20
4.4 Sel Algoritması ile Yönlendirme 20
4.5 Dijkstra Algoritması ile Yönlendirme 24

BÖLÜM 5

ROBOT GEZİNME 30

5.1 Sezgisel Algoritmalar 30
5.2 A* Algoritması 30
5.3 A* Algoritmasının Özellikleri 31
5.4 A* Algoritmasının Çalışma Mantığı 31
5.5 Maliyet Hesabı (Score Function) 34
5.5.1 G Maliyet Hesabı (G Score) 34
5.5.2 H Maliyet Hesabı (H Score) 34
5.5.3 F Maliyet Hesabı (F Score) 34
5.6 Sezgisel Fonksiyonlar (Heuristic Functions) 35
5.6.1 Manhattan Uzaklığı Yöntemi 35
5.6.2 Diagonal Uzaklığı Yöntemi 35
5.6.3 Öklid Uzaklığı Yöntemi 36
5.7 Engellerden Kaçınma 37
5.8 Deneysel Sonuçlar 38

BÖLÜM 6

ROBOCUP SANAL ROBOT YARIŞMASI VE SİMÜLASYON ARAÇLARI 44

6.1 RoboCup Arama ve Kurtarma Ligi 45
6.2 Simülasyon Ortamı ve Araçlar 46

6.2.1 USARSim (Urban Search and Rescue Simulator) 46
6.2.2 WSS (Wireless Simulation Server) 47

BÖLÜM 7

DENEYSEL SONUÇLAR 49

7.1 Noop Propagation Modeli 49
7.2 Distance ve Obstacle Propagation Modeli 50
7.3 Sonuç 56

ÖZGEÇMİŞ 59

SİMGE LİSTESİ

dBm	Desibel
m	Mesaj Paketi
ms	Milisaniye
Rcur	Şu Anki Robot
Rdest	Hedef Robot
S	Sinyal Gücü
sn	Saniye
th	Eşik Değeri
vb	Ve Benzeri

KISALTMA LİSTESİ

AAAI	Association for the Advancement of Artificial Intelligence
ABR	Associativity Based Routing
AODV	Ad hoc On-demand Distance Vector
AP	Erişim Noktası - Access Point
BF	Blind Flooding Algorithm - Sel Algoritması
CBR	Constant Bit Rate
CGSR	Clusterhead Gateway Switch Routing Protocol
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing
FSR	Fisheye State Routing
GSM	Global System for Mobile Communications
GSR	Global State Routing
IGP	Interior Gateway Protocol
IS-IS	Intermediate System to Intermediate System
MANET	Mobile Ad-Hoc Network - Gezgin Tasarısız Ağlar
OLSR	Optimized Link State Routing
OSPF	Open Shortest Path First
PDA	Personal Digital Asisstant
RF	Radio Frequency - Radyo Frekans
RIP	Routing Information Protocol
RRL	Rescue Robot League - Arama Kurtarma Robot Ligi
RSL	Rescue Simulation League - Arama Kurtarma Simülasyon Ligi
TB	Table Based Approach - Tablo Tabanlı Yaklaşım
TBRPF	Topology Broadcast Based on Reverse Path Forwarding
TORA	Temporally Ordered Routing Algorithm
USARSim	Urban Search and Rescue Simulator
Wi-Fi	Wireless Fidelity - Kablosuz Bağlantı
WRP	Wireless Routing Protocol
WSS	Wirelss Simulation Server

ŞEKİL LİSTESİ

Sayfa

Şekil 3. 1 Altyapılı ağ örneği 9
Şekil 3. 2 Altyapısız ağ örneği 9
Şekil 3. 3 Karma ağ örneği 10
Şekil 3. 4 MANET'lerde yönlendirme protokollerinin sınıflandırılması 11
Şekil 4. 1 DSR protokolü ile rota seçimi 18
Şekil 4. 2 AODV protokolü ile rota seçimi 19
Şekil 4. 3 Sel baskını yönteminde paketlerin ağ içinde yayılması 21
Şekil 4. 3 Sel baskını yönteminde paketlerin ağ içinde yayılması (devamı) 22
Şekil 4. 4 Örnek ağ yapısı 24
Şekil 4. 5 Dijkstra algoritması ile optimum rotaların belirlenmesi 25
Şekil 4. 5 Dijkstra algoritması ile optimum rotaların belirlenmesi (devamı) 26
Şekil 4. 5 Dijkstra algoritması ile optimum rotaların belirlenmesi (devamı) 27
Şekil 5. 1 Manhattan uzaklık yöntemi 35
Şekil 5. 2 Diagonal uzaklık yöntemi 36
Şekil 5. 3 Öklit uzaklık yöntemi 36
Şekil 5. 4 Engellerden kaçınma 37
Şekil 5. 4 Engellerden kaçınma (devamı) 38
Şekil 5. 5 Çalışma zamanın engel sayısına göre değişimi 39
Şekil 5. 6 Bellek kullanımının engel sayısına göre değişimi 41
Şekil 5. 7 Farklı yoğunluktaki deneysel haritalar 41
Şekil 5. 7 Farklı yoğunluktaki deneysel haritalar (devamı) 42
Şekil 5. 7 Farklı yoğunluktaki deneysel haritalar (devamı) 43
Şekil 6. 1 Arama kurtarma işlemi yapan bir robot 45
Şekil 6. 2 Örnek bir felaket ortamları 47
Şekil 6. 3 WSS grafik ara yüzü 48
Şekil 7. 1 Robot sayısına göre mesaj gecikme süresinin kutu grafiği 50
Şekil 7. 2 Test senaryoları 52
Şekil 7. 3 Sel algoritması için başarılı paket gönderim oranı (paket/saniye) 53
Şekil 7. 4 Sel ve Tablo tabanlı yöntemlerin paket gecikmelerinin histogramı 55

ÇİZELGE LİSTESİ

Sayfa

Çizelge 3. 1 Proaktif ve reakif protokollerinin karşılaştırılması 13
Çizelge 4. 1 Komşuluk matrisi 25
Çizelge 5. 1 Zaman analiz sonuçları 39
Çizelge 5. 2 Bellek kullanımı analiz sonuçları 40
Çizelge 7. 1 Maksimum paket gönderim hızı sonuçları 53
Çizelge 7. 2 Paket gecikme sonuçları 54

ÖZET

MOBİL ROBOT SÜRÜLERİ İÇİN DİNAMİK YÖNLENDİRME ALGORİTMALARI

Khudaydad MAHMOODI

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Yrd. Doç. Dr. Sırma Çekirdek YAVUZ

Günümüz iletişiminin önemli bir parçasını oluşturan kablosuz ağların kullanımı üstel olarak artmaktadır. Bu ağların bir kolu olan tasarsız ağlar da akademik dünyada çok ilgi çeken bir çalışma alanıdır. Tasarsız ağlar çok adımlı, alt yapısız ve genellikle hareketli düğümlerden oluşan kablosuz ağlardır. Bu ağlarda düğümler hem diğer düğümlerle iletişim kurarlar hem de paketleri ileterek yönlendirici görevi görürler. Tasarsız ağlar kimyasal atıkların imha edilmesi, nükleer enerjinin işlenmesi, yangın söndürme, askeri veya sivil arama kurtarma görevleri, gezegen keşfi, güvenlik, gözetleme vs. gibi alanlarda kullanılmaktadırlar.

Tasarsız ağlarda bir altyapının mevcut olmaması, düğümlerin hareketli olması, güç kapasitesinin ve bant genişliğinin kısıtlı olması bu ağların en büyük problemleridir. Özellikle düğümlerin hareketliliği topolojinin sık sık değişmesine ve kurulan yolların bozulmasına neden olmaktadır. Bu durum da etkili bir yönlendirme protokolünün kullanımını gerektirmektedir. Tasarsız ağlarda kullanılan yönlendirme protokolleri daha çok en kısa yol bulma esasına dayanmaktadır.

Çoklu robotlu büyük sistemlerde kısıtlı iletişim kapasitelerinden dolayı bütün robotların bilgi alışverişini bir kerede sağlamak zorlaşmaktadır. Bu tarz sistemlerde robotlar arası iletişimi sağlamak için bir tasarısız ağ oluşturmak daha uygundur. Özellikle engebeli arazilerde kurtarma işlemi yapan robotlar için iletişim son derece önemlidir. Büyük felaket ortamında arama kurtarma işlemi yapan robotlar her zaman merkezi yönetimi üstlenen robot ile iletişimde olmayabilir. Böyle durumlarda robotlar arası iletişim kesilmemeli ve mesaj paketleri ara robotlar üzerinden merkezi robota iletilmelidir.

Mesaj paketlerini kaynak robottan hedef robota iletilirken hangi ara robotlar üzerinden gönderileceğini yönlendirme algoritmaları belirler. Bu çalışmada sel algoritması ve tablo tabanlı bir algoritma geliştirilerek USARSim ve WSS simülasyon araçları kullanılarak üç farklı senaryo üzerinde performans ölçümü yapılmış ve test edilmiştir. Yönlendirme algoritmalarının performans ölçümü için mesaj paketinin iletim süresi ve maksimum paket gönderim oranı göz önünde bulundurularak değerlendirilmiştir. Netice itibariyle tablo-tabanlı yönlendirme algoritmasının sel algoritmasına göre çok daha başarılı olduğu gözlenmiştir.

Anahtar Kelimeler: Gezgin tasarsız ağlar, Yönlendirme protokolleri, Proaktif protokoller, Reaktif protokoller, Yönlendirme algoritmaları, USARSim, WSS.

ABSTRACT

DYNAMIC ROUTING ALGORITHMS FOR MOBILE ROBOT SYSTEMS

Khudaydad MAHMOODI

Department of Computer Engineering

MSc. Thesis

Adviser: Asst. Prof. Dr. Sırma Çekirdek YAVUZ

The use of wireless networks which are an essential part of our communication today is growing exponentially. Ad-hoc networks are a class of wireless networks drawing the attention of academic world, industry and governments. Ad-hoc networks are multi hop, infrastructure-less and spontaneous wireless networks which mostly consist of mobile nodes. Nodes in ad-hoc networks both act as a router and communicate with other nodes. Ad-hoc networks are applied to applications such as the disposal of toxic waste, nuclear power processing, firefighting, civilian search and rescue missions, planetary exploration, security, surveillance and reconnaissance tasks have elements of danger.

The absence of infrastructure, the mobile nature of the nodes, the limitations on power capacity and transmission range are great challenges of ad hoc networks that require efficient routing. Especially the mobility of the nodes, which makes topological changes and breaks in paths making it necessary to use an efficient routing protocol. Ad hoc routing protocols are mostly based on finding the shortest path.

It is important that robot teams have an effective communication infrastructure, especially for robots making rescue operations in debris areas. The robots making rescue operation in a large area of disaster are not always directly connected with central operator. In such large areas robots can move around without losing communication with each other only by passing messages from one to another up to the central operator. Routing methods determine from which node to which node the messages are conveyed. In this work blind flooding and table-based routing methods

are tested for three different scenarios to measure their effectiveness using the simulation environment USARSIM and its wireless simulation server WSS. Message delay times and maximum data packet streaming rates are considered for measuring the effectiveness. Although it has some deficiencies, it was observed that table-based approach is more advantageous than blind flooding.

Keywords: Mobile ad-hoc networks, Routing protocols, Proacitve protocols, Reactive protocols, Routing algorithms, USARSim, WSS.

BÖLÜM 1

GİRİŞ

1.1 Literatür Özeti

Teknolojinin gelişmesi ile son yıllarda mobil robotlar ve mobil robot takımlarının uzaktan işletilmesi oldukça önem kazanmıştır. Son zamanlarda iletişim alt yapısının oldukça zayıf veya hiç olmadığı geçilmesi zor ve engebeli arazilerde çalışabilen birçok mobil robot geliştirilmiştir. Bu tarz teknolojik gelişmeler sayesinde robotlar ortak çalışabilen daha zeki ve daha fazla yeteneğe sahip hale gelmişlerdir.

İşbirlikçi mobil robot sistemlerinin geliştirilmesinin arkasındaki itici güç onların tehlikeli uygulamalarda insana olan ihtiyacı düşürme potansiyelleridir. Kimyasal atıkların imha edilmesi, nükleer enerjinin işlenmesi, yangın söndürme, askeri veya sivil arama kurtarma görevleri, gezegen keşfi, güvenlik, gözetleme vs. gibi uygulamalar bu alanda yapılan uygulamalara örnek olarak verilebilir. Kablolu iletişim mobil robotların bu tarz işlemleri düşük maliyet ile gerçeklenmesini sağlar [1].

İletişim alt yapısının oldukça zayıf veya hiç olmadığı yerlerde bağlanabilirliğin tesisine olan ihtiyaç gittikçe artmaktadır. Bu sebepten ötürü tasarsız ağlar ortaya çıkmıştır. Kablosuz ağlar altyapılı ağlar ve altyapısız ağlar (Mobil Ad-hoc Ağlar - MANETs) olmak üzere iki sınıfa ayrılabilir [2].

MANET'ler herhangi bir altyapı gerektirmeksizin otonom olarak kendi kendine organize olabilen ve yapılanabilen ağlardır. Yani geçici bir ağ oluşturmak için merkezi bir yönetime ihtiyaç duymaz. Bu tür ağlarda ağdaki her bir düğüm aynı zamanda bir yönlendirici görevini de üstlenir. Bu ağlarda hareketlilik oldukça fazla olduğundan ağ topolojisinde sık ve tahmin edilemez değişimler gözlenir [2],[3].

Son zamanlarda gezgin tasarsız ağların esnekliğinden ve ağ altyapılarına olan bağımsızlığından dolayı araştırmacıların ilgi konusu olmuştur. Bu ağların altyapıya ihtiyaç duymamaları ve doğaları gereği dinamik olması nedeniyle uçtan uca iletişimin verimli gerçekleştirilmesi için yeni ağ stratejilerinin gerçeklenmesini gerektirir. MANET'ler hızlı bir şekilde az maliyetle gerçekleştirilebilirler ve kolayca yönetilebilirler.

MANET'ler için şimdiye kadar birçok yönlendirme protokolleri önerilmiş ve gerçeklenmiştir. Bunların bant genişliği kullanımının iyileştirilmesi, minimum enerji tüketimi, yüksek verim, paket başına ek yük miktarının azaltılması, gecikmenin azaltılması gibi amaçları vardır. Farklı yönlendirme protokolleri gönderici ve alıcı arasındaki optimal yolu belirlemede farklı ölçüler kullanmaktadır. Her protokolün kendine özgü avantaj ve dezavantajları vardır. Bizim senaryomuzda kritik olan husus paket gecikmesini minimize ederek minimum hop sayısıyla robotlar arası iletişimi en iyi ve hızlı bir şekilde sağlamaktır.

Bu çalışmada robotlar arası paket iletim zamanını minimize etmek için minimum hopla çalışan dijikstra algoritmasını kullanan bir uygulama gerçeklenmiş ve USARSim üzerinde performansı test edilmiştir.

1.2 Tezin Amacı

Bu tez çalışmasının temel amacı çok robotlu bir sistemde robotlar arası iletişimi hızlandırmak ve mesaj paket gecikmelerini en aza indirmek amacı ile dijkstra algoritmasını kullanan tablo-tabanlı bir yönlendirme algoritması geliştirilmesidir. Buradaki amaç herhangi bir altyapı gerektirmeyen tasarsız ağların sağladığı avantajları kullanarak herhangi bir felaket ortamında arama kurtarma görevi yapan işbirlikçi robotların paket kaybı yaşamadan iletişimini sağlamaktır. Gerçekleştirdiğimiz bu yöntem robotlar arası sinyal gücünü kullanarak optimum rotayı belirleyerek mesaj paketini bu rota üzerinden hedef robota iletmektedir.

Gerçekleştirdiğimiz yönlendirme tekniğinin başarısını ölçmek için sel algoritmasını kullanan diğer bir yönlendirme tekniği daha gerçeklenmiştir. Bu çalışma boyunca bu iki yönlendirme tekniğinin başarıları karşılaştırılmış ve robot sayısına göre ideal paket gönderim hızı tespit edilmeye çalışılmıştır. Performans karşılaştırması için birçok farklı

senaryo oluşturularak senaryoların her biri USARSim ve WSS simülasyon araçları üzerinde test edilmiştir. Simülasyon sonuçları analiz edilerek yönlendirme tekniklerinin farklı durumlara göre sergilediği performans başarıları incelenmiştir. Gerçeklenen tüm deneylerde dijkstra algoritması ile gerçeklenen tablo-tabanlı yönlendirme tekniğinin tüm durumlar için makul sonuç veridiğini ve hiçbir mesaj paketi kaybı veya endişe edecek derecede paket gecikmesi olmadığı gözlemlenmiştir.

1.3 Hipotez

Tez çalışması kapsamında gezgin tasarsız ağlarda kullanılan birçok yönlendirme protokolleri karşılaştırmalı olarak incelenmiş ve uyugulamamıza en uygun olan iki farklı yönlendirme algoritması gerçeklenmiş ve test edilmiştir. Sel baskını yöntemi uygulaması basit olması, tamponda taşma olmadığı müddetçe her mimaride paketin hedefe varmasını garanti etmesi, ekstra yapılandırma mesajlarına ihtiyaç duymaması yönü ile az sayıda düğümlerden oluşan ağlarda tercih edilebilir. Fakat düğümler arası bağlantı sayısı artışının karesi oranında mesaj gecikmelerine ve paket iletim hızınında yavaşlamaya neden göz önünde bulundurulmalıdır. Tablo-tabanlı diğer bir yönlendirme yöntemi ise düğümler arası bağlantı artışına göre lineer artışlı bir gecikme sağlaması ve paket iletim hızının yüksek olması önemli bir avantajdır. Fakat yöntem gereği merkezi düğüm (Comstation) tarafından belirli aralıklarla diğer düğümlere gönderilmesi gereken dinamik rotalama tablosu taşıyan mesajlar hat kapasitesinin harcanmasına sebep olur. Bu mesajın sıklığı azaldığında ise robotların rotalama tablosunu öğrenememesi ya da geç öğrenmesine sebep olacağından paket kaybını arttırır. Bu iki oranı dengeleyen gezgin tasarsız ağlar için tablo-tabanlı daha iyi bir yönlendirme yöntemi gerçeklenebilir.

BÖLÜM 2

KABLOLU VE KABLOSUZ AĞLAR

İki veya daha fazla ağ cihazı arasında iletişimi sağlayan bağlantılara ağ denir. Cep telefonunuzdan tutun Wi-Fi'lı bilgisayarınıza kadar her şey bir ağ üzerinden çalışmaktadır. Günlük hayatımızda kullandığımız ağlar genel olarak kablolu ve kablosuz ağlar olmak üzere ikiye ayrılır.

2.1 Kablolu Ağlar

Dünya çapında küresel bir ağ olan Internet yanında, bir kurumun içindeki bilgisayarları birbirine bağlayan yerel ağlar mevcuttur. Bu yerel ağların ilki, Hawaii adalarında 1970'lerin başında ortaya çıkmıştır. Uzaktaki adaları, Honolulu adasındaki Hawaii Üniversitesi ana bilgisayarına bağlayabilmek için telsizler kullanılmıştır, ancak kablo kullanmadan çalışan ağların yaygınlaşması yaklaşık 30 yıl kadar sonra gerçekleşmiştir. Yaklaşık aynı zamanlarda, Hawaii'deki çalışmalardan da etkilenerek, Xerox şirketinde kablolu bir yerel ağ kurulmuştur ve “Ethernet” adı verilmiştir. Bu sistemde uzun bir eş eksenli kabloya (koaksiyel kablo) şirketteki tüm bilgisayarlar bağlanmıştır [4].

Kablolu ağlar, ağdaki cihazların birbirlerine kablo vasıtası ile bağlandıkları ağlardır. Kablolu ağlarda kablo uzunluğunun artması iletişim performansını olumsuz etkilemektedir. Bu bağlantı türünde kullanılan kablo türüne göre bağlantı hızında değişiklikler olabilmektedir. Kablolu ağlar, en hızlı ve en güvenilir ağlardır. Çeşitli kablolama standartları, ek donanım ihtiyacı ve karmaşık bilgisayar yapılandırmalarından dolayı bu tür ağların gerçeklenmesi zordur. Bu nedenden dolayı bu tür ağları gerçekleştirirken tecrübeli bir ağ uzmanına başvurulması gerekebilir.

2.1.1 Kablolu Ağların Avantajları

- **Hız:** Kablolu ağlarda veri iletişim hızı kablosuz ağlara göre çok daha hızlıdır. Genellikle kablolu ağlarda veri iletişim hızı 100 Mbps ile Gbps'ler arasında değişmektedir.
- **Güvenlik:** Kablolu ağlar kablosuz ağlara göre daha güvenlidirler. Çünkü iletişim herkesin erişebileceği radyo frekans (RF) sinyalleri yerine sadece kabloya bağlı olanlar veriye erişebilirler.
- **Konfigürasyon:** Kablolu ağlarda konfigürasyon işlemleri kablosuz ağlara nazaran çok daha kolaydır.
- **Engeller:** Kablo ağlar fiziksel engellerden etkilenmez. Çünkü veri transferi kablo üzerinden olduğu için engellerin etkisi yoktur.

2.1.2 Kablolu Ağların Dezavantajları

- **Mobilite:** Kablolu yerel ağlarda ağdaki cihazların yer değiştirmesi ve sürekli hareketli olması çok zordur. Hatta imkansızdır.
- **Maliyet:** Geniş bir kablolu ağ kurabilmek için çok fazla ağ cihazı ve kabloya ihtiyaç vardır. Dolaysıyla ağ cihazları ve kablo maliyetlerinden dolayı bu tür ağların kurulum maliyeti çok yüksektir.
- **Kurulum Esnekliği:** Kablolu ağ sistemleri kurulumu bazı ortamlarda çok zordur ve uzmanlık gerektirebilir.
- **Genişletilebilirlik:** Kablolu ağlara yeni düğümlerin eklenebilmesi için ek donanıma ve kabloya ihtiyaç vardır. Yani ağın genişletilmesi ek maliyet gerektirir.

2.2 Kablosuz Ağlar

Kablosuz ağlar, sahip oldukları kablosuz alıcı ve vericiler aracılığı ile infrared, mikrodalga ve radyo frekans (RF) gibi kablosuz iletim ortamları üzerinden herhangi bir fiziksel bağlantı olmaksızın birbirleri ile haberleşen ağ cihazlarından oluşur. Günümüzde kullanılmakta olan kablosuz ağ teknolojilerin çoğunluğu RF üzerinden haberleşmektedir. Heterojen bir yapıya sahip olan kablosuz ağlar, kablosuz haberleşme

yeteneğine sahip masaüstü bilgisayarlar, dizüstü bilgisayarlar, yazıcılar, tarayıcılar, PDA'lar, akıllı telefonlar, tablet bilgisayarlar vb. gibi farklı türde birçok cihazın haberleşmesinden oluşmaktadır [5].

Kablosuz ağlar en çok kullanılan ve giderek popüler hale gelen kurulumu kolay ağlardır. Mobil cihazların kullanılmasının giderek çoğalması ile birlikte ev ve ofisler için kablosuz ağlar daha yaygınlaşmıştır. Günümüzde kablosuz ağlar eskiye nazaran daha hızlı, daha güvenli ve daha geniş kapsama alanına sahiptirler. Aynı zamanda içine sızılması kolay ve dış müdahelelere açıktır. Düzgün kurulmadığı takdirde cihazlarda ağ kopuklukları yaşanabilir.

Günümüzde çoğu ev ve küçük ofisler kablosuz ağları az maliyeti ve alt yapısından dolayı daha çok tercih etmektedirler. Fakat kablosuz ağlar tamamen güvenli olmayabilirler ve bu tür ağlarda zaman zaman veri iletim hızının düşmesi ve eksik kurulumlarından dolayı sık sık başarısızlıklar gözlemlenebilir.

2.2.1 Kablosuz Ağların Avantajları

- **Mobilite:** Kablosuz yerel ağlar, ağ kullanıcılarına kapsama alanının hangi noktasında olursa olsunlar, hareket halinde dahi gerçek zamanlı bilgi erişimi sağlar.
- **Maliyet:** Kablosuz ağ kurabilmek için ilk olarak harcanması gereken miktar kablolu bir ağdan daha fazla olmakla birlikte hayat evresi sarfiyatı çok azdır. Uzun vadeli kazançları, çok yer değiştirme gerektiren dinamik ortamlarda kendini belli eder.
- **Kurumlum Esnekliği:** Kablosuz ağ sistemleri kurulumu hızlı ve kolaydır, ayrıca duvar ve tavanlardan kablo çekme zorunluluğu da ortadan kaldırır. Kablosuz ağ teknolojisi kablolu ağın erişemeyeceği yerlere ulaşımı sağlar.
- **Genişletilebilirlik:** Mevcut bir kablosuz ağa yeni düğüm veya kablosuz ağ erişim noktasının eklenmesi çok daha kolaydır. Yani bu tür ağlar az miktarda kullanıcıdan binlerce kullanıcıya kadar kolayca genişletilebilirler.

2.2.2 Kablosuz Ağların Dezavantajları

- **Hız:** Kablosuz ağlarda veri transfer hızı kablolu ağlara göre çok daha yavaştır. Çoğu kablosuz ağlarda veri iletişim hızı 1-54 Mbps arasında iken kablolu ağlarda bu rakam 100 Mbps ile Gbps'ler arasında değişmektedir.
- **Güvenlik:** Kablosuz ağlarda güvenlik riski vardır. Çünkü ağdaki bir saldırganın cihazı bir erişim noktası görevini de üstlenebilir. Bu cihaz üzerinden bağlanan kullanıcıların bilgilerinin çalınma riski vardır.
- **Konfigürasyon:** Kablosuz ağlarda kablolu ağlara göre konfigürasyon işlemleri karmaşık ve zordur. Konfigürasyon işlemleri için uzmanlık gerekebilir.
- **Engeller:** Oda duvarları gibi engeller sinyal kalitesinin büyük miktarda düşmesine yol açar. Ağ içerisindeki cihazların konumları değiştirildiğinde veri aktarım hızında olumsuzluklar yaşanabilir.

BÖLÜM 3

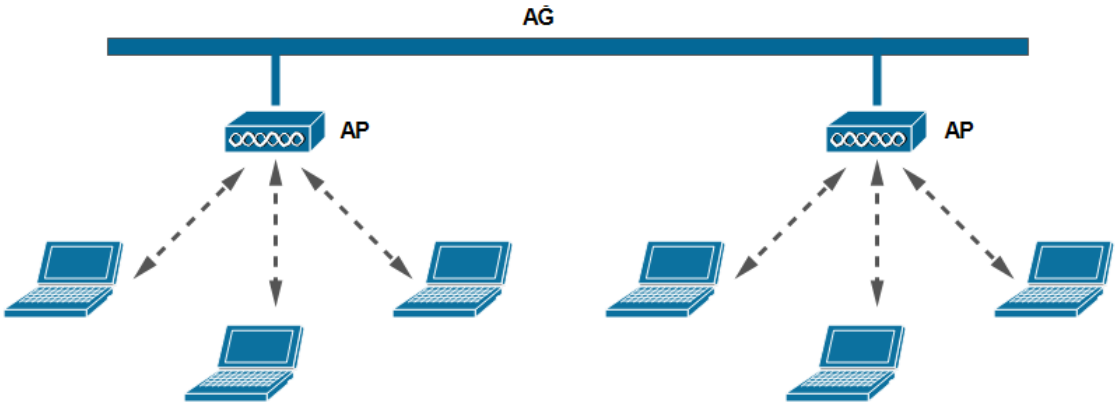
GEZGİN TASARSIZ AĞLAR

İletişim alt yapısının oldukça zayıf veya hiç olmadığı yerlerde bağlanabilirliğin tesisine olan ihtiyaç gittikçe artmaktadır. Bu sebepten ötürü gezgin tasarsız ağlar ortaya çıkmıştır. Kablosuz ağlar, ağ topolojisine göre altyapılı kablosuz ağlar, altyapısız kablosuz ağlar ve karma kablosuz ağlar olmak üzere üç sınıfa ayrılabilir [2]. Bu çalışma altyapısız kablosuz ağları konu edinmiştir.

3.1 Altyapılı Kablosuz Ağlar

Altyapısız ağlar küçük ağlar için kullanışlı bir yapı olsa da büyük ağlar için yetersiz bir yapıdır. Altyapılı kablosuz ağlar (Infrastructure-based Wireless Networks), büyük çaplı ağlar için daha uygundur. Altyapılı kablosuz ağlarda ağı denetlemek için bir erişim noktası (AP - Access Point) bulunur. Erişim noktası (AP) hangi kablosuz cihazın ne zaman konuşacağını denetler. Kablosuz ağ cihazları, alt yapı olarak bir erişim noktası kullanır, bu aygıt ile ağa bağlanarak birbirleriyle iletişim kurar. Altyapılı ağlar ev ve ortamlarında en sık kullanılan ağ topolojisidir [6].

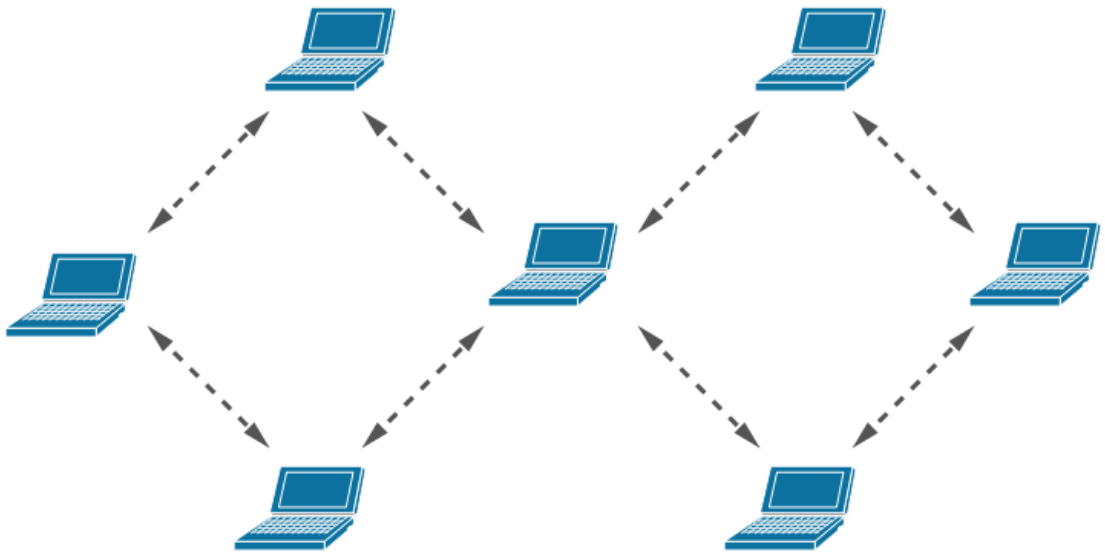
Altyapılı ağlar kapsama alanı problemini çözmek için en yaygın kullanılan yoldur. Günlük hayatta kullandığımız cep telefonları ve PDA'lar baz istasyonları ile temastadırlar. Baz istasyonları da yine internet gibi diğer ağlara bağlıdırlar. GSM (Global System for Mobile Communications) gibi cep telefonu teknolojileri bu tarz ağlar ile çalışır. Baz istasyonları genellikle tüm ülkeyi kapsayacak şekilde tasarlanmış ağlardır. Yoğun nüfuslu bölgelerde baz istasyonların sayısı kırsal alanlara göre daha fazladır. Çünkü yoğun nüfuslu bölgelerdeki yüksek binalar ve diğer engeller radyo dalgalarının yayılmasını ciddi derece de engeller.



Şekil 3. 1 Altyapılı ağ örneği

3.2 Altyapısız Kablosuz Ağlar

Altyapısız kablosuz ağlar veya gezgin tasarsız ağlar (Mobil Ad-hoc Networks - MANET) herhangi bir altyapı gerektirmeksizin otonom olarak kendi kendine organize olabilen ve yapılanabilen ağlardır. Yani geçici bir ağ oluşturmak için merkezi bir yönetime veya alt yapıya ihtiyaç duymaz. Ağdaki her bir düğüm diğer düğümlere mesaj yönlendirme işleminden sorumludurlar. Bu tür ağlarda yönlendirici olmadığı için ağdaki her bir düğüm bir yönlendirici görevini de üstlenir. Bu ağlarda hareketlilik oldukça fazla olduğundan ağ topolojisinde sık ve tahmin edilemez değişimler gözlenir.



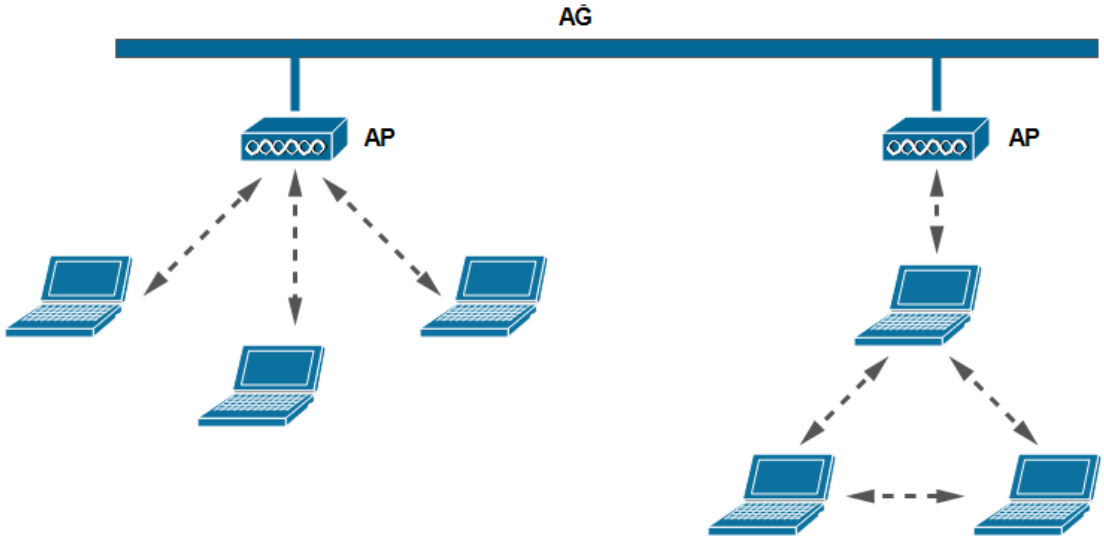
Şekil 3. 2 Altyapısız ağ örneği

İki veya daha fazla kablosuz istemcinin eşler arası (peer-to-peer) bağlantı şekilleri ile birbirine bağlanması yoluyla en küçük kablosuz ağ yapısı oluşturulur. Dolayısıyla oluşturulan ağda erişim noktası (AP) bulunmaz ve geçici olarak kurulmuş bir ağ yapısıdır. Dolaysıyla bu tür ağlara geçici ağ da denir.

Geçici ağın avantajı, ağ denetimi için herhangi bir cihaz (erişim noktası) ve ağ yapılandırma ayarına gerek yoktur. Mevcut bir altyapının olmaması, güç kapasitesinin ve bant genişliğinin kısıtlı olması geçici ağların dezavantajlarıdır. Ayrıca kablosuz cihazların hareketli olması, topolojinin sürekli değişmesine neden olur. Geçici ağda iletişimi denetleyen bir cihaz olmadığı için bağlantı kalitesi düşmektedir [6].

3.3 Karma Kablosuz Ağlar

Karma kablosuz ağlar (Hybrid Wireless Networks) altyapılı ve altyapısız ağların bir yapıda kullanılması ile oluşur. Bu nedenle mobile ağlar baz istasyonlarının erişemediği yerlerde karma ağlar kullanılarak birçok bölgeye genişletilebilir. Baz istasyonları daha sonra internet gibi diğer ağların erişimini sağlar.



Şekil 3. 3 Karma ağ örneği

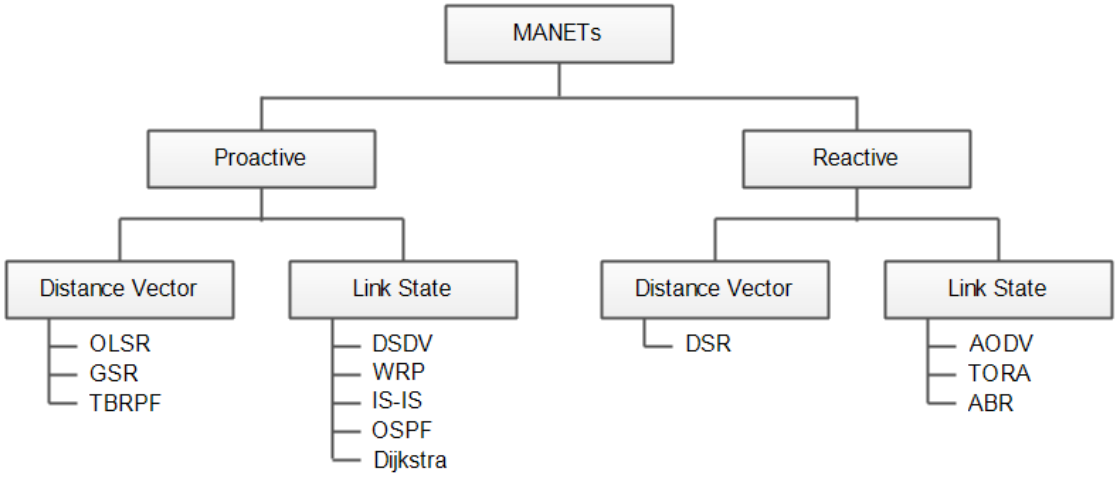
3.4 MANET'de Yönlendirme Protokollerinin Sınıflandırılması

MANET'lerdeki yönlendirme protokolleri, yönlendirme stratejisi ve ağ yapısına göre ikiye ayrılır [3]:

- Proaktif Protokoller (Table-Driven)
- Reaktif Protokoller (On-Demand)

Proaktif ve reaktif yönlendirme protokolleri, ağdaki düğümlerin yönlendiricilerden aldıkları bilgilere ve bu bilgileri kullanarak yönlendirme tablolarını şekillendirmelerine göre Mesafe Vektörü Yönlendirme ve Bağlantı Durumu Yönlendirme olmak üzere ikiye ayrılır. Ağlarda kullanılan çoğu protokoller bu iki kategorinin birisinin altında yer alır.

- Mesafe Vektörü Yönlendirme (Distance Vector Routing)
- Bağlantı Durumu Yönlendirme (Link State Routing)



Şekil 3. 4 MANET'lerde yönlendirme protokollerinin sınıflandırılması

3.4.1 Proaktif Yönlendirme Protokolleri

Proaktif veya table-driven protokolleri yönlendirme bilgisini ihtiyaç duyulmadığı zamanda da tutmaya devam eder. Ağdaki her bir düğüm diğer tüm düğümlere olan yönlendirme bilgisini tutar. Rota bilgisi genellikle yönlendirme tablolarında saklanır ve ağ topolojisi değiştikçe periyodik olarak güncellenir. En önemli özelliği yönlendirme bilgisinin kullanımdan önce hazır olmasıdır. Proaktif protokoller, yönlendirme tablolarının güncel tutulmasının getireceği ek yükün büyük ağlar için oldukça büyük

olacağından dolayı büyük çaptaki ağlar için uygun değildir. Yönlendirme tablolarındaki yükün büyük olması daha fazla bant genişliği kullanılmasına yol açar [7],[8].

3.4.2 Reaktif Yönlendirme Protokolleri

Reaktif veya on-demand yönlendirme protokolleri rota bilgisini ancak bir düğüm tarafından ihtiyaç duyulduğu zaman oluşturur. Bir düğüm hedefe bir yol bulmak isterse ağda rota keşif işlemini başlatır. Rota keşif işlemi sadece ihtiyaç duyulduğu durumlarda başlatılır. Bu işlem kaynak veya hedef tarafından başlatılabilir. Rota bir kere keşfedildikten sonra keşif işlemi sona erer ve bu rota bozulana kadar veya istenmeyene kadar geçerli olur [9]. Paket gönderimleri bu rota üzerinden yapılır. Eğer iki düğüm arasında iletişim yok ise bu iki düğüm arasındaki yönlendirme bilgisine de ihtiyaç yoktur [2]. Bu protokoller genellikle büyük ağlarda proaktif protokollerine göre daha iyi performans sergilerler. Ancak büyük ölçekli ağ trafiği performansın keskin bir şekilde düşmesine neden olur. Çünkü bu tarz protokollerin çoğu rota bulmak için ağda sel gibi yayılır ve bağlantıların tıkanmasına yol açar. Bir başka dezavantajı ise ağda bir rota bulabilmek için gecikmeye ihtiyaç duyar ve bu da bazı uygulamalar için kabul edilemez bir durumdur [3],[8].

3.4.3 Mesafe Vektörü Yönlendirme

Ağdaki her bir yönlendirici komşu yönlendiricilere hedefe nasıl ileşeceği bilgisini mesafe vektörü protokollerini kullanarak gönderir. Ayrıca yönlendiricilere hedef düğüme ne kadar uzakta olduğu ve hedefe ulaşmak için hangi yönü kullanılacağı bilgisi de gönderilir. Ayrıca yönlendiricilere hedefe ne kadar uzakta olduğu ve hedefe ulaşmak için hangi yönü kullanması gerektiğini bildiren iki parça bilgi daha gönderir. Yönlendirici diğer yönlendiricilerden bilgi aldığı zaman, hedef adresleri, mesafeleri ve komşu yönlendiricileri ile ilgili bilgilere göre bir tablo oluşturur ve bu tablodan hedefe olan en kısa rota seçer. Yönlendiriciler uzaklık vektörü protokolünü kullanarak paketi yönlendirme tablosundaki en kısa rotayı kullanarak komşularına veya hedefe iletir. Paketi alan yönlendirici bu noktadan sonra diğerine nasıl ileteceğini bildiğini varsayalım. Buna en iyi örnek olarak Routing Information Protocol (RIP) verebiliriz [7].

3.4.4 Bağlantı Durumu Yönlendirme

Bağlantı durumu protokollerinde yönlendirici hedef ile ilgili bilgiyi sağlamaz. Bunun yerine ağ topolojisi ile ilgili bilgiyi sağlar. Bu bilgi genellikle belirli yönlendiriciye bağlı ağ bağlantılarının aktif veya pasif olduğu durum bilgisini içerir. Bu bilgi ağ boyunca yayılır ve ağdaki her yönlendirici tüm bağlantılarının mevcut durum bilgisini oluşturur [7].

3.5 Proaktif ve Reaktif Yönlendirme Protokollerinin Karşılaştırılması

Proaktif ve reaktif yönlendirme protokollerinin kendine göre avantaj ve dezavantajları vardır. Gerçekleştirecek uygulamaların ihtiyaç durumlarına göre bu protokollerden biri seçilebilir.

Çizelge 3. 1 Proaktif ve reakif protokollerinin karşılaştırılması

	PROAKTİF	**REAKTİF**
YÖNLENDİRME BİLGİSİ	Yönlendirme tablosunda her zaman var	İhtiyaç duyulduğunda oluşturulur
ROTANIN GÜNCELLENMESİ	Periyodik olarak yapılır	İstek olduğunda yapılır
DEPOLAMA İHTİYAÇLARI	Yüksek	Genellikle proaktife göre daha az
EK YÜK	Ağın büyüklük oranına göre değişir	İletimde olan düğüm sayısına göre değişir
GECİKME	Az	Çoğu uygulamalar için gecikme fazla

3.6 Gezgin Tasarsız Ağlardaki Problemler

- **Asimetrik Bağlantılar (Asymmetric Links):** Kablolu ağların çoğu sabit ve simetrik bağlantılar üzerine kuruludur. Fakat bu durum ad-hoc ağlarda düğümlerin sürekli hareketli olduğu ve ağ içerisinde sürekli yer değiştirdiği için söz konusu değildir. Örneğin bir MANET'teki bir A düğümünden B düğümüne olan sinyal kalitesi iyi iken B düğümünden A düğümüne olan sinyal kalitesi iyi olmayabilir. Yani iki düğüm arasındaki bağlantı simetrik olmayabilir.

- **Yönlendirme Yükü (Routing Overhead):** Ad-hoc ağlarda düğümler genellikle ağ içinde sürekli konum değiştirirler. Bu konum değiştirmeler yönlendirme tablosunda bazı gereksiz rotaların oluşmasına neden olur.
- **Girişim veya İletim Çatışması (Interference):** İletim karakteristiklerine göre gelen ve giden bağlantılardan birinin diğeri ile çatışabilmesi ad-hoc ağların en büyük sorunlarından birisidir. Yani bir düğüm başka düğümlerin iletimlerini dinleyebilir ve bu da toplam iletimin bozulmasına sebep olur.
- **Değişken Topoloji (Dynamic Topology):** Ad-hoc ağlarda topolojinin sabit olmaması yönlendirme için önemli bir sorundur. Ağ içerisindeki düğümler hareket edebilirler veya bazı karakteristikleri değişebilir. Ad-hoc ağlarda yönlendirme tabloları bu değişiklikleri ağ topolojisine ve yönlendirme algoritmalarına bir şekilde yansıtmalı ve onlara adapte olmalıdırlar. Örneğin sabit bir ağda yönlendirme tablosunun güncellenmesi 30 saniye aralıklarla gerçekleştirilir. Bu güncelleme sıklığı ad-hoc ağlar için oldukça düşük olabilir.

BÖLÜM 4

YÖNLENDİRME PROTOKOLLERİ

Yönlendirme stratejileri bir ağ üzerinde optimum rotayı belirleyen stratejilerdir. Rota mesaj paketinin düğümler üzerinden gönderilirken izlediği yoldur. Eğer mesaj paketinin hedefe hızlı bir şekilde iletilmesi gerekirse mesajın gönderildiği rotanın mümkün olduğunca kısa olması gerekir. En kısa yolu belirlemek için birçok yönlendirme stratejisi vardır. Bu stratejilerin hepsi tüm uygulamalar için en iyi sonucu vermezler ve uygulamaya göre en uygununu seçmek gerekir.

4.1 Proaktif Yönlendirme Protokolleri

Proaktif yönlendirme protokollerinin en yaygın kullanılanları hakkında detaylı bilgi alt bölümlerde verilmiştir.

4.1.1 DSDV (Destination Sequenced Distance Vector)

DSDV kablosuz ağlar için tasarlanan ilk protokollerden biridir. DSDV protokolünün temeli Bellman-Ford algoritmasına dayanır. Bu yönlendirme protokolünde ağdaki her bir mobil düğüm, diğer düğümler için yolların bulunduğu bir yönlendirme tablosunu tutar. Yönlendirme tablolarının her biri mevcut tüm hedef düğümlerin listesini ve bu düğümlere olan hop saylarını içerir. Tablodaki her bir kayıt başlangıçta kaynak düğüm tarafından bir sıra numarası ile etiketlenmiştir. Sıra numarası o yolun güncelliğini gösterir. Yönlendirme tablolarının periyodik olarak güncellenmesi ağ topolojisi hakkında güncel bilgiyi sağlar. Eğer yönlendirme bilgisi için herhangi bir önemli yeni değişiklik var ise güncellemeler hemen iletilir. Bu yüzden yönlendirme bilgisi güncellemeleri ya periyodik ya da olay güdümlü olabilir. DSDV protokolunda ağdaki her

bir düğümün, yönlendirme tablosunu kendi komşularına bildirmesi gerekir. Bu bildirim yayım (broadcasting) veya çoklu-yayım (multicasting) şeklinde olabilir. Komşu düğümler bu bildirimler sayesinde düğümlerin ağ içerisinde hareket etmesi ile oluşan her türlü ağ değişiklikleri hakkında bilgi sahibi olurlar. Yönlendirme güncellemeleri toptan (full dump) veya artırımlı (inceremental) olmak üzere iki şekilde gönderilebilir. Toptan güncellemelerde yönlendirme tablosunun tamamı komşu düğümlere gönderilir, fakat artırımlı güncellemelerde ise sadece yönlendirme tablosunun değişiklik gösteren kayıtları gönderilir [10].

4.1.2 WRP (Wireless Routing Protocol)

WRP, yol bulma algoritmalarının genel sınıfındandır. Bir dizi dağıtık en kısa yol bulma algoritması olarak tanımlanır. Bu algoritmalar uzunluğa ve her hedef noktaya en kısa yolların 2. den sonuncuya kadar hoplara dair bilgileri kullanarak yolları hesaplar. WRP geçici yönlendirme döngülerinin oluştuğu durumların sayısını azaltır. Her bir düğüm yönlendirme amacıyla dört şeyi muhafaza eder:

1. bir uzaklık tablosu,
2. bir yönlendirme tablosu,
3. bir bağlantı maliyeti tablosu
4. bir mesaj yeniden iletim listesi

WRP, düğümlerin komşularına yapılan periyodik güncelleme mesajı iletimlerini kullanır. Güncelleme mesajının yanıt listesindeki düğümler onay göndermelidirler. Eğer son güncellemeden beri değişim olmazsa, yanıt listesindeki düğümler bağlanırlığın teminatı olarak boşta durduklarını belirten bir "Merhaba" mesajı yollamalıdırlar. Bir düğüm, bir komşusundan güncelleme mesajı aldıktan sonra yönlendirme tablosunun güncellenip güncellemeyeceğine karar verir ve yeni bilgiyi kullanarak daima daha iyi bir yol arar. Eğer bir düğüm daha iyi bir yol elde ederse, bu bilgiyi tablolarını güncelleyebilmeleri için orijinal düğümlere geri gönderir. Onay alımından sonra orijinal düğüm mesaj yeniden iletim listesisini günceller. Böylece her defa da yönlendirme bilgisinin tutarlılığı bu protokoldeki tüm düğümler tarafından kontrol edilir. Bu işlem yönlendirme

döngülerinin ortadan kaldırılmasına yardımcı olur ve daima yönlendirme için ağda en iyi çözümü bulmaya çalışır [10].

4.1.3 CGSR (Clusterhead Gateway Switch Routing Protocol)

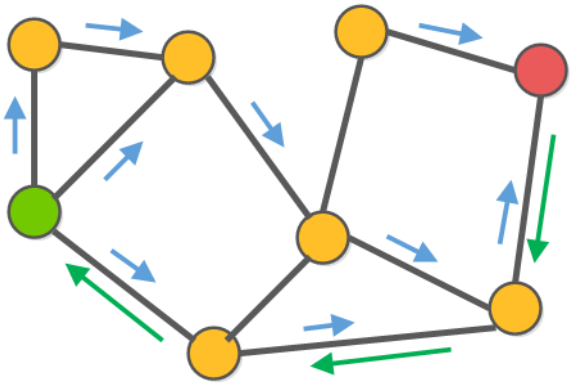
CGSR, düz bir ağ yerine kümelenmiş bir mobil ağlar olduğunu düşünür. Ağı ayrık fakat bağlantılı gruplar şeklinde yapılandırmak için küme başlarını, küme başı seçme algoritması kullanılarak seçilir. Bu protokol, bazı kümeler şekillendirilerek ağda dağıtık bir işleme mekanizması elde eder. Ancak bu protokolün bir dezavantajı ise küme başlarının sıkça değişimi ve seçimi kaynak açlığına sebep olabilir, bu da yönlendirme performansını etkileyebilir. CGSR, yönlendirme düzeni olarak DSDV protokolünü kullanır, bu nedenle ek yükü DSDV ile aynıdır. Yine de DSDV büzerinde bir değişiklik yapar; kanaktan hedefe dağıtım trafiği için hiyerarşik bir "küme başından ağ geçidine" yönlendirme yaklaşımı kullanır. Geçit düğümleri, iki veya daha fazla küme başının iletişim alanında bulunan düğümlerdir. Bir düğüm tarafından gönderilen bir paket ilk önce kümenin başına gönderilir sonra da paket kümenin başı tarafından bir başka küme başına giden ağ geçitine gönderilir. Bu işlem hedef düğümün küme başına ulaşılıncaya kadar devam eder. Paket daha sonra hedef düğümün küme başı tarafından hedef düğüme ulaştırılır [10].

4.2 Reaktif Yönlendirme Protokolleri

Reaktif yönlendirme protokollerinin en yaygın kullanılanları hakkında detaylı bilgi alt bölümlerde verilmiştir.

4.2.1 DSR (Dynamic Source Routing)

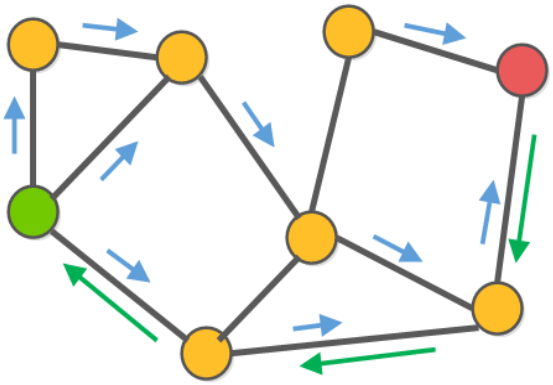
Dinamik Kaynak Dağıtımı (DSR), kaynak rotalama yaklaşımını baz alan reaktif bir protokoldür. Şekil 4.1'deki gibi, DSR protokolü bağlantı durumu algoritmasını baz alır ve rota keşif işlemi talep üzerine kaynak tarafından başlatır. Gönderici kaynaktan hedefe rotayı belirler ve paketteki rota kaydına ara düğümlerin adreslerini dahil eder. DSR protokolü küçük çaplı çok sıçramalı ağlar için tasarlanmıştır. Düğümler arasında kendilerini ağdaki komşularına belirten "Merhaba" mesaj takasları olmadığı için işaretsiz bir protokoldür [11],[10].



Şekil 4. 1 DSR protokolü ile rota seçimi

4.2.2 AODV (Ad hoc On-demand Distance Vector)

AODV temelde DSDV'nin bir iyileştirmesidir. Fakat, AODV proaktif değil reaktif bir protokoldür. Talebe dayalı rotalar yaratarak yayım sayısını minimize eder. Bu durum DSDV için geçerli değildir. Herhangi bir kaynak düğüm bir hedefe bir paket göndermek istediğinde bir rota istek (RREQ) paketi yayar. Komşu düğümler sırayla paketi komşularına iletir; bu işlem hedef düğüme paket ulaşıncaya kadar devam eder. Rota isteğinin taşınma işlemi sırasında aralardaki düğümler, yayım paketinin ilk kopyasının alındığı komşunun adresini kayıt altına alır. Bu kayıt rota tablolarında saklanır ve bu kayıtlar ters yolun oluşturulmasında yardımcı olur. Eğer daha sonra aynı RREQ'in başka kopyaları alınırsa, bu paketler gözardı edilirler. Cevap ters yol kullanılarak gönderilir. Rotanın sürekliliği için, bir kaynak düğüm hareket ettiğinde rota keşif işlemini yeniden başlatabilir. Eğer belli bir rotadaki herhangi bir ara düğüm hareket ederse, kayan düğümün komşusu bağlantıdaki başarısızlığı sezebilir ve yukarısındaki komşuya bir bağlantı başarısızlığı ilgili uyarı gönderir. Bu işlem başarısızlık uyarısı kaynak düğüme ulaşıncaya kadar devam eder. Alınan bilgi baz alınarak, kaynak düğüm rota keşif fazını yeniden başlatmaya karar verebilir [12],[10].



Şekil 4. 2 AODV protokolü ile rota seçimi

4.2.3 TORA (Temporally Ordered Routing Algorithm)

TORA, bazı iyileştirmeler içeren bir reaktif yönlendirme protokolüdür. Kaynaktan hedef düğüme olan rotanın bir "Yönlü Çevrimsiz Graf"ını (Directed Acyclic Graph - DAG) yaratarak düğümler arasında bir bağ tesis edilir. Bu protokol, rota keşif işleminde "geri bağlantı" modelini kullanır. Rota keşif sorgusu ağ boyunca yayımlanarak hedefe veya hedef hakkında bilgi içeren bir düğüme ulaşılıncaya kadar çoğaltılır. TORA "yükseklik" olarak adlandırılan bir parametreye sahiptir. Yükseklik, yanıt veren düğümün gerekli hedef düğüme kadar ki uzaklığını belirten bir uzaklık ölçüsüdür. Rota keşfi aşamasında bu parametre sorgulama yapan düğüme döndürülür. Sorgunun yanıtı geriye doğru yayılırken her ara düğüm kendi TORA tablosunu hedef düğüme olan rota ve yükseklik değeri ile günceller. Kaynak düğüm daha sonra hedefe giden en iyi rotayı seçmek için yükseklik değerini kullanır. Bu protokolün ilginç bir özelliği de en kısa yol yerine genellikle en uygun yolu seçemesidir. Tüm bu denemeler ile TORA, yönlendirme yönetimi ağ trafik yükünü minimize etmeye çalışır [10].

4.2.4 ABR (Associativity Based Routing)

ABR protokolü gezgin tasarsız ağlar için "ilişkilendirme kararlılığı derecesi" şeklinde yeni bir yönlendirme ölçütü tanımlar. Bu yönlendirme protokolünde bir rota, mobil düğümlerin ilişkilendirme kararlılığı derecesi baz alınarak seçilir. Her düğüm ağda varlığını ilan etmek için peryodik olarak işaret mesajı üretir. Bir komşu düğüm işaret mesajını aldığında kendi ilişkilenirlik tablosunu günceller. Alınan her işaret için, alan düğüm ile işaret veren düğüm arasındaki ilişkilenirlik değeri bir artırılır. Herhangi bir

işaret veren düğüm için yüksek ilişkilenirlik değeri, bu düğümün görece durağan olduğu anlamına gelir. İlişkilenirlik değeri, herhangi bir komşu düğüm başka bir düğümün komşuluğuna geçiş yaptığında sıfırlanır [10].

4.3 Yönlendirme Algoritmaları

Yönlendirme algoritmaları, yönlendiriciler üzerinde tutulan ve en uygun yolun belirlenmesinde kullanılan tabloların dinamik olarak güncellenmesi için kullanılır. Temelde, biri uzaklık vektörü, diğeri bağlantı durumu algoritması olarak adlandırılan iki farklı yönlendirme algoritması vardır. RIP, OSPF, IGP gibi birçok yönlendirme protokolü bu iki algoritmadan birine dayanır.

Yönlendirme algoritmaları en iyi yolu seçerken farklı metrik değerleri kullanır. Her algoritmanın kendine göre avantajları vardır. Algoritmalar aşağıdaki farklı değerler baz alınarak işlem yapar:

- Bant genişliği
- Gecikme
- Yük
- Güvenlik
- Atlama sayısı
- Maliyet
- İm sayısı

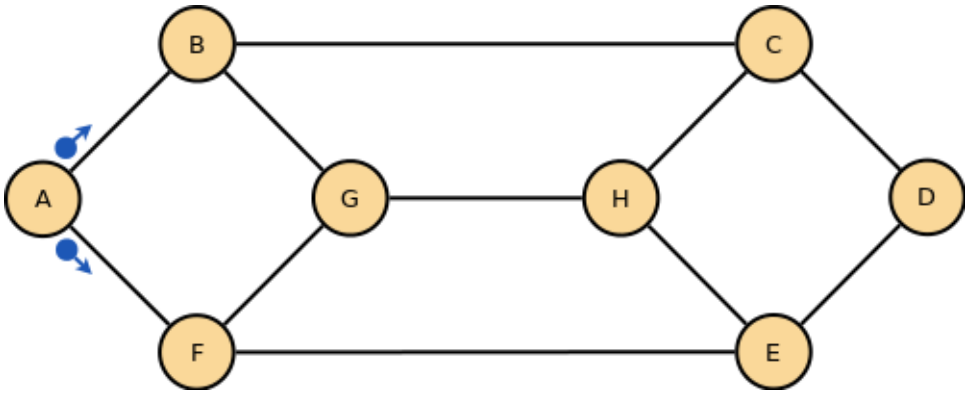
Uygulamamızda paket gecikmesi süresi önemli olduğu için çalışmamız bunu üzerine olmuştur. Bu bölümde sel ve dijkstra yönlendirme algoritmalarını inceleyeceğiz [13].

4.4 Sel Algoritması ile Yönlendirme

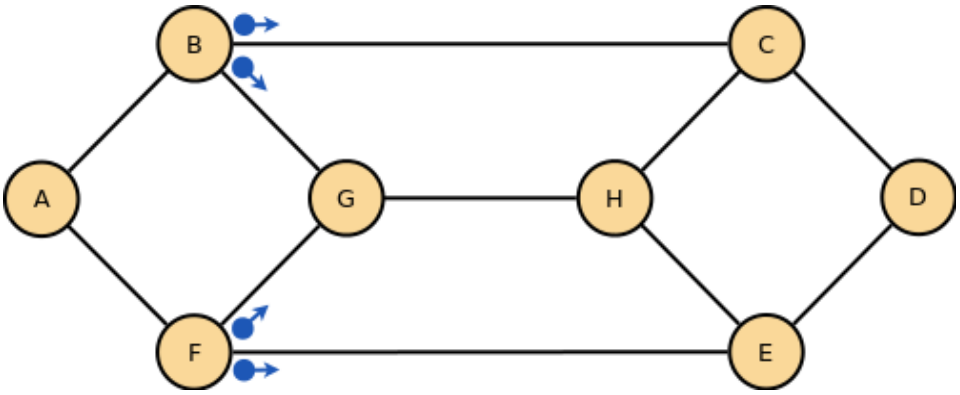
Sel yönlendirme stratejisi ağdaki tüm komşu düğümlere mesaj paketini iletmeyi ve dağıtmayı sağlayan en yaygın kullanılan yönlendirme stratejilerinden biridir. Bu algoritmanın gerçeklenmesi çok kolaydır. Bu algoritmada bir düğüm aldığı mesaj paketini, paketin gönderildiği düğüm hariç ağdaki tüm komşu düğümlere gönderir [14]. Doğal olarak bu yöntem aynı paketin gereksiz pek çok kopyasının yaratılmasına, bu kopyaların ağdaki trafiği aşırı derecede yoğunlaştırmasına ve daha çok bant genişliğinin kullanılmasına neden olacaktır. Mesaj paketlerinin tüm komşu düğümlere gönderilmesi

paketin hedefe ulaşma olasılığını da arttırır. Bu işlem su baskını benzeri akan suyun açıklık bulduğu tüm yollara dağılması gibidir. Sel algoritması kablolu ağlarda arzu edilen davranışı sergileyebilir [13],[15].

Sel baskını yönteminde seçilecek hat için özel hesaplamalar yapılmasına gerek kalmaz. Paket, doğal olarak ek kısa yol üzerinden hedef düğüme erişir. Ancak bu sırada aynı paketin pek çok kopyası yaratılır. Hatta aynı kopyalar pek çok kez aynı düğümlere ulaşır. Şekil 4.3 (a-d), A düğümünden D düğümüne gönderilen bir paketin üç sekme içindeki çoğalmasını ve eriştiği düğümleri gösterir [16],[13].

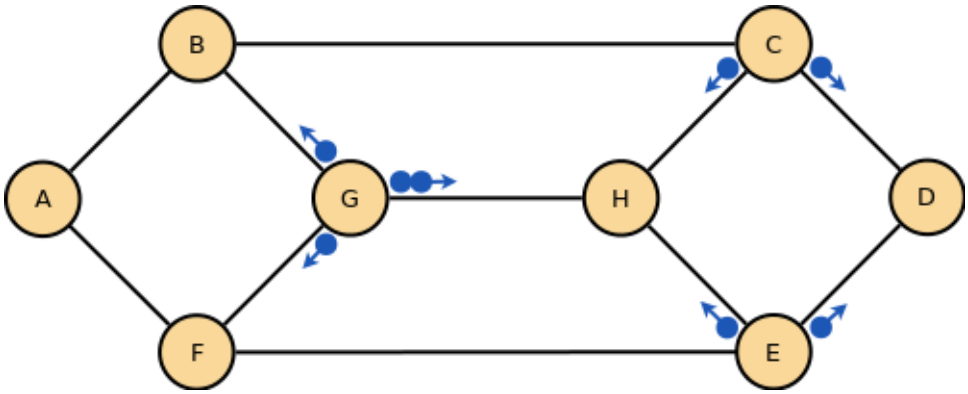


(a) A düğümünden komşu düğümlere mesaj akışı

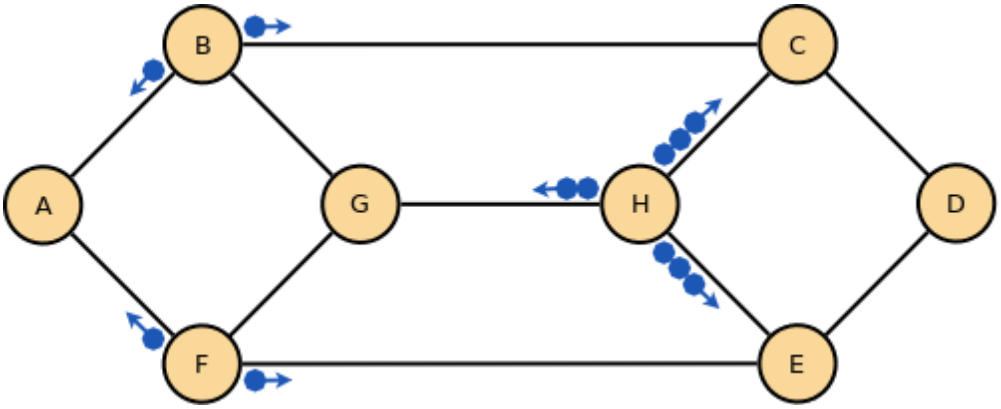


(b) B ve F düğümlerinden komşu düğümlere mesaj akışı

Şekil 4. 3 Sel baskını yönteminde paketlerin ağ içinde yayılması



(c) C, E ve G düğümlerinden komşu düğümlere mesaj akışı



(d) B,D,F ve H düğümlerinden komşu düğümlere mesaj akışı

Şekil 4. 4 Sel baskını yönteminde paketlerin ağ içinde yayılması (devamı)

Şekil 4.3(a) A düğümü mesaj paketinin iki kopyasını üretir ve komşusu olan B ve F düğümlerine gönderir, Şekil 4.3(b) B ve F düğümleri A düğümünden aldıkları mesaj paketinin iki kopyasını üretirler. B düğümü komşusu olan C ve G düğümlerine ve F düğümü komşusu olan E ve G düğümlerine mesaj paketini gönderir, Şekil 4.3(c) C düğümü B düğümünden aldığı mesaj paketini D ve H komşularına, E düğümü F düğümünden aldığı mesaj paketini D ve H komşularına ve G düğümü B düğümünden aldığı mesaj paketini F ve H komşularına ve F düğümünden aldığı mesaj paketini B ve H komşularına gönderir, Şekil 4.3(d) Son olarak B düğümü G den aldığı mesaj paketini A ve C komşularına, F düğümü G den aldığı mesaj paketini A ve E komşularına, H düğümü G ve E den aldığı mesaj paketlerini C komşusuna, G ve C den aldığı mesaj paketlerini E komşusuna ve C ve E den aldığı mesaj paketlerini ise G komşusuna gönderir. Hedef düğüm olan D düğümü ise mesaj paketini C ve E düğümlerine gönderir.

Diğer taraftan bu algoritmanın mesaj paketini tekrar tekrar gönderime ve mesaj paketi kopyasının uzun süre ağda dolaşması gibi bazı eksikleri de vardır. Tekniğin dezavantajlarını önlemek için paketlere sekme sayacı eklenmesi önerilmiştir. Sekme sayacına, paket yaratıldığında, kaynak düğüm tarafından kaynak ve hedef düğümleri arasındaki sekme sayısını gösteren bir değer atanır. Sekme sayacının değeri geçilen her düğümde bir azaltılır. Sekme sayacı sıfır değerine ulaştığında, paket hedef düğüme ulaşmamışsa yok edilir. Paketlerin aynı düğüm tarafından tekrar tekrar kopyalanmasını engellemek için önerilen bir diğer yöntem ise her düğümün yarattığı/kopyaladığı paketlerin kaydını tutmasıdır. Yeni gelen bir paketin kopyalarının yaratılmasından önce bu kayıtlar kontrol edilir ve paket daha önce çoğaltılmadıysa işleme devam edilir aksi durumda paket yok edilir.

Sel algoritmasını uygulamamızda geliştirip ve geliştirdiğimiz diğer yöntemler ile karşılaştırılarak test edilmiştir. Aşağıda sel algoritmasına ait sözde kodu bulunmaktadır. Burada Rcur şu anki robotu, Rdest paketin gideceği hedef robot, S(i,j) i.robot ile j.robot arasındaki sinyal gücü değeri, m mesaj paketi, th sinyal gücü eşik değerini temsil etmektedir.

Algoritma: Sel algoritması ile yönlendirme

1. **Inputs:** R_{cur}, R_{dest}, $S_{(i,j)}$, m, th
2. **while** m mesaj paketi geldi mi **do**
3. **if** $R_{cur} = R_{dest}$ **then**
4. Mesaj paketini al ve işle
5. **else**
6. **if** $S_{(R(cur),R(dest))} >= th$ **then**
7. Mesaj paketini direct olarak R_{dest} robotuna gönder
8. **else**
9. Mesaj paketini kendisi ve mesaji gönderen robot hariç
10. R_{cur} robotunun tüm komşularına gönder
11. **end if**
12. **end if**
13. **end while**

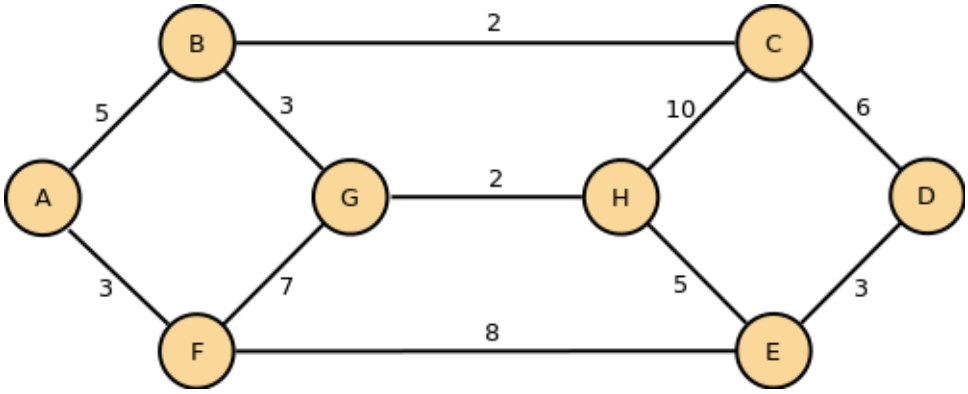
Sel algoritması genellikle pratik değildir ama askeri amaçlar için uygun olabilir. Bir anda çok sayıda yönlendirici bombalama sonucu devreden çıksa bile sel algoritmasının sağlamlığı nedeniyle paketin karşıya ulaşma şansı yüksektir.

4.5 Dijkstra Algoritması ile Yönlendirme

Yönlendirme teknikleri üzerinde düşünmeye başladığınızda akla gelebilecek ilk teknik, iki nokta arasındaki en kısa yolu bulmak ve paketleri o yol üzerinden aktarmak olacaktır. En kısa yolu bulma algoritması, verilen iki düğüm arasındaki en kısa yolu bulan bir algoritmadır [13]. Uygulamada yönlendirme işleminin en optimum bir şekilde gerçekleştirilmesi için dijkstra algoritması seçildi. Dijkstra algoritmasının tercih edilme nedeni amaca uygun ve geliştirmesi kolay olmasıdır.

Dijkstra algoritması 1959'da Hollandalı bilgisayar bilimcisi Edsger Dijkstra tarafından geliştirilmiştir. Dijkstra algoritması, negatif kenarı bulunmayan bir grafta en kısa yol problemini bir kısa yol ağacı üreterek çözen bir graf arama algoritmasıdır. Genellikle ağ protokollerinde yönlendirme işlemleri, oyun programlama, ulaşım, vs gibi alanlarda sıkça kullanılır [17].

Dijkstra algoritması, graflarda verilen bir başlangıç düğümden diğer tüm düğümlere olan en kısa yolu bulur. Düğümler arası tüm kenarların negatif değer almaması gerekir. Dijkstra algoritması, ziyaret edilmiş düğümler kümesi ve tüm diğer ziyaret edilmemiş düğümlere olan mesafeleri sürekli olarak güncellemesi suretiyle çalışır. Her iterasyonda ziyaret edilmemiş en yakın düğüm, ziyaret edilmiş düğümler kümesine eklenir ve bu düğümün ziyaret edilmemiş komşularına olan mesafeler güncellenir.



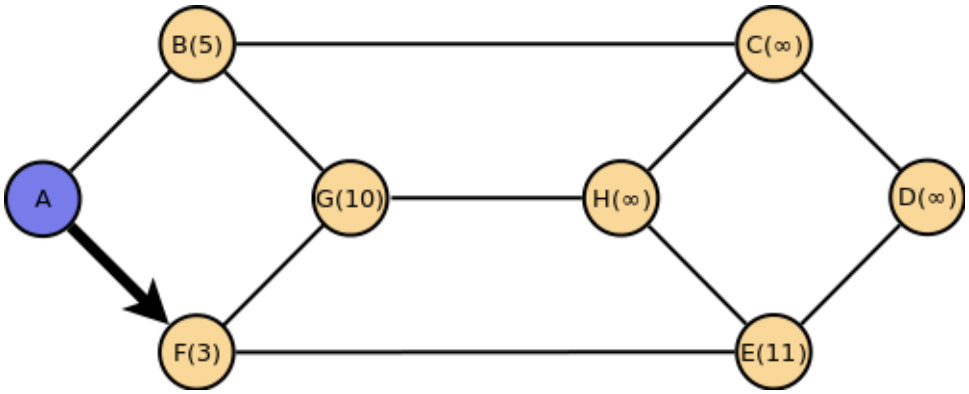
Şekil 4. 5 Örnek ağ yapısı

Şekilde 4.4'de kullanılan ölçütün gecikme değerleri olduğu düşünülerek A düğümünden diğer düğümlere olan en kısa yolu bulunsun. İlk önce komşu düğümler arası ölçütleri gösteren komşuluk matrisi yazılır.

Çizelge 4. 1 Komşuluk matrisi

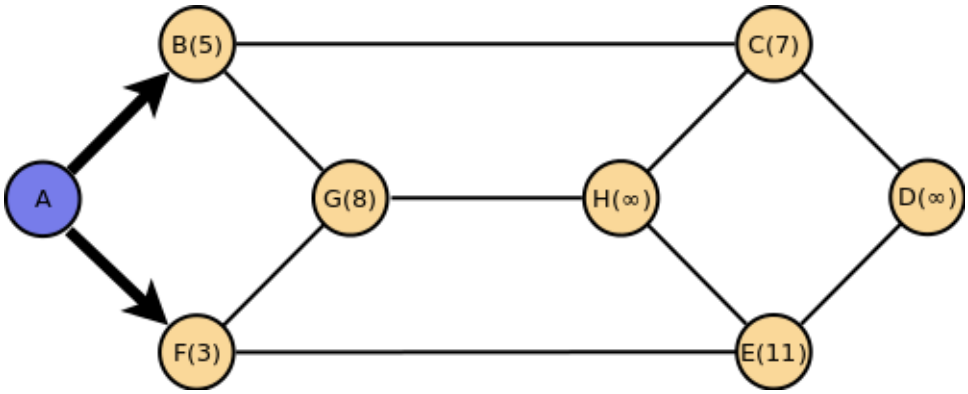
	A	B	C	D	E	F	G	H
A	0	5	∞	∞	∞	3	∞	∞
B	5	0	2	∞	∞	∞	3	∞
C	∞	2	0	6	∞	∞	∞	10
D	∞	∞	6	0	3	∞	∞	∞
E	∞	∞	∞	3	0	8	∞	5
F	3	∞	∞	∞	8	0	7	∞
G	∞	3	∞	∞	∞	7	0	2
H	∞	∞	10	∞	5	∞	2	0

A düğümünden başlanıyor ve ona komşu olan düğümler incelenerek onlara geçici uzaklık değerler atanır. Şekilde 4.4'de B ve F düğümleri için belirlenmiş geçici uzaklık değerleri bulunabilir. Diğer düğümler henüz incelenmediği için onların geçici uzaklık değerleri sonsuz işareti ile gösterilmiştir. Şekil 4.5 (a-g), A düğümünden diğer tüm düğümlere olan optimum rotaların seçilmesi gösterilmiştir [13],[18].

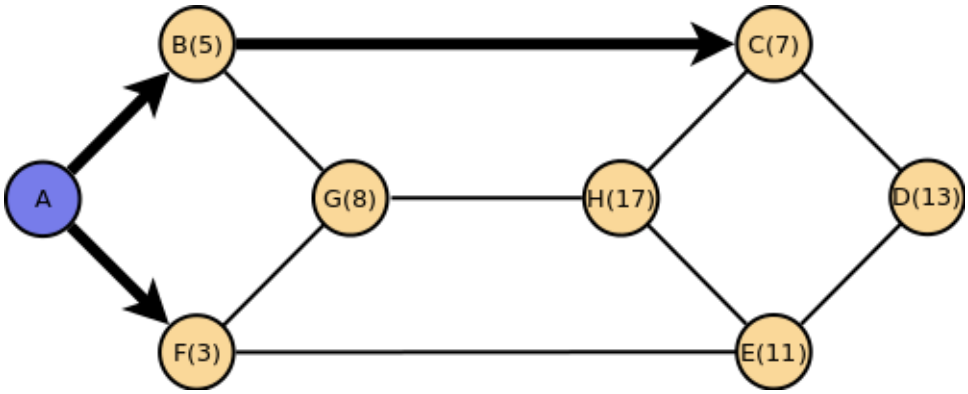


(a)A düğümüne en yakın F düğümü seçilir

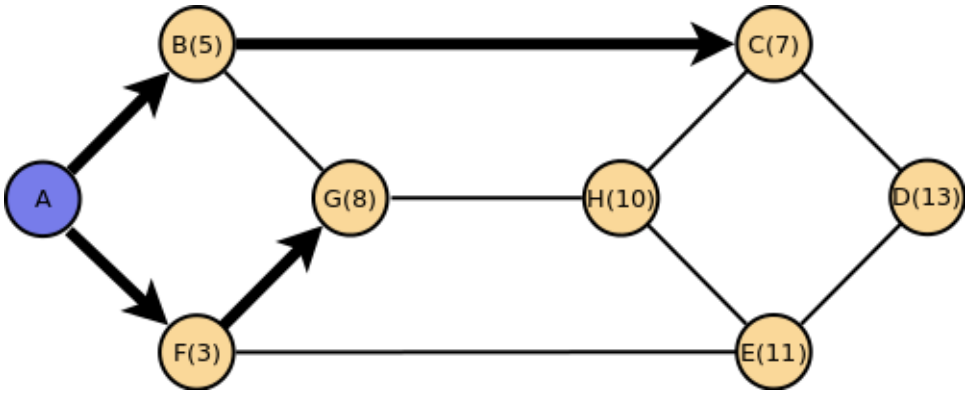
Şekil 4. 6 Dijkstra algoritması ile optimum rotaların belirlenmesi



(b) A düğümüne F düğümünden sonra ikinci en yakın B düğümü seçilir

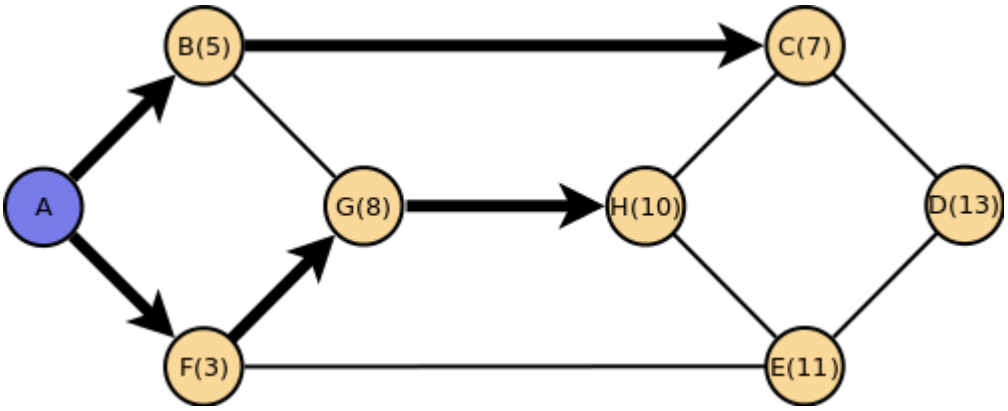


(c) A düğümüne F ve B düğümlerinden sonra en yakın C düğümü seçilir

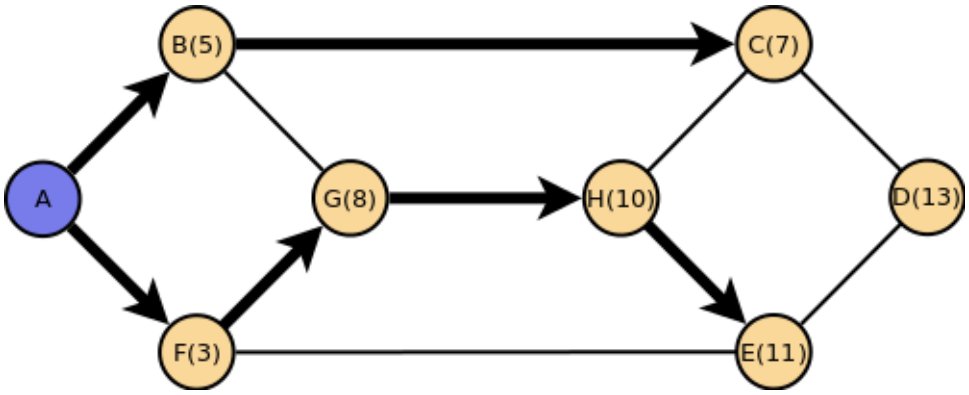


(d) A düğümüne F,B ve C düğümlerinden sonra en yakın G düğümü seçilir

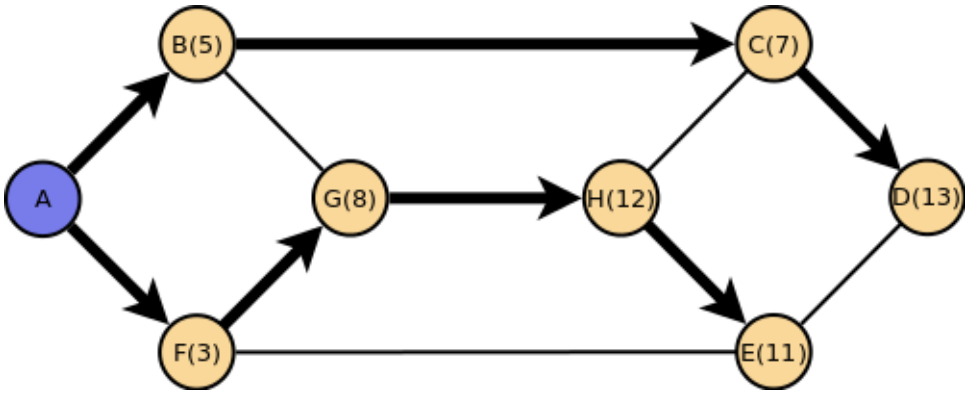
Şekil 4. 7 Dijkstra algoritması ile optimum rotaların belirlenmesi (devamı)



(e) A düğümüne F,B,C ve G düğümlerinden sonra en yakın H düğümü seçilir



(f) A düğümüne F,B,C,G ve H düğümlerinden sonra en yakın E düğümü seçilir



(g) A düğümüne F,B,C,G,H ve E düğümlerinden sonra en yakın D düğümü seçilir

Şekil 4. 8 Dijkstra algoritması ile optimum rotaların belirlenmesi (devamı)

Örnekten de anlaşılacağı gibi bu yöntemde son düğüm dışında tüm düğümler bir kez işlenmiştir. Yukarıdaki örnekte A kaynak düğümünden diğer düğümlere olan en kısa yollar hesaplanmıştır. Aynı yöntem kullanılarak diğer kaynak düğümler için de benzer hesaplamalar yapılabilir.

Farz edelim #robots toplam robot sayısı, N tüm robotlar kümesi, S(i,j) i.robot ile j.robot arasındaki sinyal gücü, Path(n) comstation ile n.robot arasındaki rota, R ulaşılabilen robotlar kümesi, TCost(n) comstation ile n.robot arasındaki toplam rota maliyeti olsun. Buna göre algoritmanın adımları aşağıdaki gibidir.

Algoritma: Dijkstra algoritması

1. **Inputs:** N={1,...,#robots}, $S_{(i,j)}$, i=1,...,#robots, j=1,...,#robots
2. **Outputs:** Path(n), n={2,...,#robots}

3. Denklem(1) kullanarak Cost(i,j) hesapla
4. **for each** robot n in N - {1} **do**
5. Tcost(n) = Cost(1,n)
6. Path(n) = {n}
7. **end for**

8. **while** R ≠ N **do**
9. {N - R} kümesindeki minimum TCost(m) değerine sahip m robotunu bul
10. R = R ∪ {m}
11. **for each** robot n in {N - R} **do**
12. TCost(1,n) = min(TCost(n), TCost(m) + Cost(n,m))
13. **if** TCost(m) + Cost(n,m) < TCost(n) **then**
14. Path(n) = Path(m) ∪ n
15. **end if**
16. **end for**
17. **end while**

Dijkstra algoritması negatif kenar ağırlıklarına sahip graflar için çalışmamaktadır. Ancak uygulamamızda mesafeyi temsil eden robot sinyal güçleri sıfır veya sıfırdan küçük değerler almaktadır. Buna göre sinyal gücü 0 ve -93 değerleri arasında olan robotlar kapsama alanında ve sinyal gücü -93'den küçük olan robotlar kapsama alanının dışında olduğunu gösterir. Algoritmanın uygulamada kullanılabilmesi için bu değerler normalize edildi.

$$Cost_{(i,j)} = \begin{cases} 1 + \left|S_{(i,j)}\right| / 1000 & -93 \leq S_{(i,j)} \leq 0 \\ \infty & S_{(i,j)} < -93 \end{cases} \quad (4.1)$$

Yukarıdaki matematiksel denklemde S(i,j), i.robot ile j.robot arasındaki sinyal gücü seviyesini göstermektedir. Cost(i,j) ise algoritma tarafından kullanılan i.robot ile j.robot arasındaki link maliyetini göstermektedir.

Algoritma: Tablo-tabanlı yaklaşım ile yönlendirme

1. **Inputs:** R_{cur}, R_{dest}, m, Path
2. **while** m mesaj paketi geldi mi **do**
3. **if** $R_{cur} = R_{dest}$ **then**
4. Mesaj paketini al ve işle
5. **else**
6. m mesaj paketindeki rotayı al
7. Rotadaki bir sonraki robot düğümünü bul
8. m mesaj paketini bulduğun robota gönder
9. **end if**
10. **end while**

Uygulamada optimum rota hesaplama işlemi her 5 saniyede bir kez comstation tarafından yapılmaktadır. Comstation bu güncel rota bilgilerini tüm robotlara periyodik olarak göndermektedir. Robotlar ağ içerisinde bu rotaları mesaj alışverişi için kullanmaktadırlar. Bir önceki başlıktaki tanımlara göre robotların mesaj alışverişi için kullandıkları tablo-tabanlı yaklaşımın sözde kodu yukarıda verilmiştir.

BÖLÜM 5

ROBOT GEZİNME

5.1 Sezgisel Algoritmalar

Bilgisayar bilimlerinde, sezgisel ya da buluşsal (heuristic) bir problem çözme tekniğidir. Sonucun doğruluğunun kanıtlanabilir olup olmadığını önemsememektedir. Fakat genelde iyiye yakın çözüm yolları elde eder. Sezgisel algoritmalar ise geçiş süresinde daha verimli hale gelebilmek için en iyi çözümü aramaktan vaz geçerek çözüm zamanını azaltan algoritmalardır.

Sezgisel algoritmalar en iyi sonucu bulacaklarını garanti etmezler fakat makul bir süre içerisinde bir çözüm elde edeceklerini garanti ederler. Genellikle en iyiye yakın olan çözüm yoluna hızlı ve kolay bir şekilde ulaşırlar.

Sezgisel arama algoritmalarına örnek olarak A* Araması (A star), Demet Araması (Beam search), Tırmanış Araması (Hill climbing), En iyi öncelikli arama (Best first search), Açgözlü en iyi öncelikli arama (Greedy best first search) [19] sayılabilir.

5.2 A* Algoritması

A* algoritması, içerisinde barındırdığı buluşsal fonksiyon sebebiyle yapı olarak sezgisel (heuristic) algoritma olarak isimlendirilmektedir. A* algoritmasında kullanılan sezgisel fonksiyonu f(n) ile ifade edecek olursak;

$$f(n) = g(n) + h(n) \tag{5.1}$$

Denklem (5.1)'de g(n), başlangıç düğümünden mevcut düğüme kadar gelme maliyeti, h(n) mevcut düğümden hedef düğüme varmak için tahmin edilen mesafeyi ve f(n) toplam maliyeti göstermektedir.

5.3 A* Algoritmasının Özellikleri

A* algoritmasının başlıca özellikleri aşağıdaki gibidir [20]:

- Tam bir algoritmadır. Eğer bir çözüm varsa her zaman bulur.
- İşlem sürecini önemli derecede hızlandırmak için sezgisel yaklaşımı kullanabilir.
- Düğümden düğüme hareket maliyeti farklı olabilir. Örneğin engebeli ve geçilmesi zor arazilerde hareket yavaş ve çapraz geçişler zor olduğundan farklı maliyetli düz bir yol seçilebilir.
- Arama işlemi istendiğinde birçok farklı yönde yapılabilir.

5.4 A* Algoritmasının Çalışma Mantığı

Dijkstra algoritmasında kullanılan başlangıç düğümünden itibaren bir noktadan dışarıya doğru bütün yönlerde ilerleme metodunun aksine, A* algoritmasında hedef düğüme doğru, doğrudan bir maliyet hesaplanır. Algoritma ilerleme yönünü bu maliyetin artıp azalmasına göre ayarlamaktadır. Böylece karmaşık ortamlarda dahi, algoritma g(n) fonksiyonunu kullanarak hedeften uzaklaştığını anlayabilir. g(n) parametresi aynı zamanda maliyet ile hedefe yaklaşma arasında bir denge unsurudur. Diğer bir deyişle g(n) fonksiyonu, algoritmanın hızı ile doğruluğu arasında belirleyici bir rol oynamaktadır.

Algoritmanın çalışması esnasında öncelikle kaynak düğümün komşu düğümleri ziyaret edilir. Ardından f(n) değeri en düşük olan düğüm öncelikli olmak üzere hedef düğüm bulunana kadar ilerlenmeye devam edilir. Her bir adımda, bir önceki düğüm, gidilen düğümün ebeveyni olarak işaretlenir. Böylece hedef düğüme ulaşıldığında düğümlerin ebeveynleri takip edilerek başlangıç ve hedef düğümleri arasındaki yol elde edilir.

Gidilmesi muhtemel düğümler, algoritma içerisinde "açık liste" isimli bir listede tutulmaktadır. Açık listede bulunan ve ziyaret edilen düğümler tekrar kontrol

edilmemesi için “kapalı liste”ye alınmaktadır. Kapalı listede yer alan bir düğüm, daha kısa bir yol bulunması durumunda tekrar açık listeye alınmaktadır. Açık liste içerisindeki düğümler, başlangıç düğümünden hedef düğüme kadar olan toplam maliyetlerine göre sıralıdırlar. Böylece öncelik sıralamasına sahip bir liste elde edilir. Bahse konu toplam maliyet, mevcut düğüm ile başlangıç düğümü arasındaki gerçek maliyet ile mevcut düğüm ile hedef düğüm arasındaki tahmini maliyettin toplamıdır.

Hedef düğüme ulaşıldığında veya açık listede düğüm kalmayınca algoritma sonlandırılır. A* algoritması, sezgisel fonksiyonun hesapladığı tahmini maliyetin gerçek maliyetten fazla olmadığı durumlarda en kısa yolu verir. Ancak tahmini maliyet gerçek maliyetten fazla olursa bulunan yol, gerektiği kadar kısa değildir.

A* algoritması içerisinde sezgisel maliyet sıfır olarak alınırsa, diğer bir ifadeyle h(n)’in değeri sıfıra eşitlenirse, algoritma Dijkstra algoritması gibi davranış sergileyecektir. Diğer taraftan g(n) fonksiyonunun değeri sıfıra eşitlenirse algoritma, Best-First Search algoritması gibi davranış gösterecektir. İlk durumda taranacak düğüm sayısı ve işlem zamanı artacak, ikinci durumda ise hedefe yaklaşma-uzaklaşma kontrolü yapılamayacaktır. A* algoritmasının daha hızlı çalışabilmesi için geçilmesi istenmeyen düğümlere çok yüksek değerler verilebilir. Ancak en doğru çözüm, bu düğümlerin kapatılması veya çizgeden çıkarılması olacaktır. Ayrıca yoğun bellek kullanımı, A* algoritmasının en önemli dezavantajlarından birisi olduğundan, kullanılan bellek miktarının artırılması algoritmanın daha hızlı çalışmasını sağlayacaktır.

Algoritma içerisinde kullanılan sezgisel yöntemler sebebiyle, kaynak düğümden hedef düğüme ve hedef düğümden kaynak düğüme giden en kısa yollar farklı çıkabilir. Bu nedenle yeterince hızlı sistemlerde, kaynak düğümden hedef düğüme ve hedef düğümden kaynak düğüme giden en kısa yollar hesaplanır ve elde edilen iki sonuçtan en uygunu seçilerek kullanılabilir.

Ayrıca A-Yıldız algoritmasının çeşitli iyileştirmeler içeren Iterative Deepening A* (IDA*), Memory-Bounded A* (MA*), Simplified Memory Bounded A* (SMA*) ve Recursive Best-First Search (RBFS) şeklinde türevleri mevcuttur [21].

Algoritma: A* algoritmasının sözde kodu

```
1.  Açık listeyi ilklendir
2.  Kapalı listeiyi ilklendir
3.
4.  Başlangıç düğüm ile hedef düğümün aynı olmadığından emin ol
5.  Başlangıç ve hedef düğümlerin duvar olmadığından emin ol
6.
7.  Başlangıç düğüm için: G = 0
8.  Başlangıç düğüm için: H sesgisel fonksiyon ile hesapla
9.  Başlangıç düğüm için: F = G + H
10. Başlangıç düğümü açık listeye ekle
11.
12. while açık liste boş olmadığı sürece do
13.     Açık listeden F maliyeti en düşük olanı bul ve mevcut düğüm olarak işaretle
14.
15.     if mevcut düğüm hedef düğüm ise then
16.         Hedefe düğüme varıldı ve rotayı oluştur ve dön
17.     else
18.         Mevcut düğümü açık listeden çıkar ve kapalı listeye ekle
19.         Mevcut düğümün komşularını bul
20.
21.         foreach her bir komşu düğüm için do
22.             if komşu düğüm kapalı listede ise then
23.                 İşlem yapmadan bir sonraki komşuya geç
24.             end if
25.
26.             if komşu düğüm açık listede değil ise then
27.
28.                 if komşu düğüm köşegen bir düğüm ise then
29.                     Yeni düğüm için: G = G + 1.414
30.                 else
31.                     Yeni düğüm için: G = G + 1
32.                 end if
33.
34.                 Yeni düğüm için: H sezgisel fonksiyonu ile hesapla
35.                 Engellerden kaçınmak için mevcut düğümün etrafındaki engel
36.                 sayısını belirli bir kat sayı ile çarpıp H maliyetine ekle
37.                 Yeni düğüm için: F = G + H
38.                 Yeni düğümü açık listeye ekle
39.             end if
40.         end foreach
41.
42.     end if
43. end while
```

5.5 Maliyet Hesabı (Score Function)

A* algoritması, arama yapılacak grafta başlangıç düğümden veya ara düğümlerden sonra hangi düğüme gidilmesi gerektiğini G, H ve F maliyet değerlerini hesaplayarak değerlendirir.

5.5.1 G Maliyet Hesabı (G Score)

G maliyeti basitça başlangıç düğümden gidilecek düğüme olan her adımda birer birer artan maliyet değeridir. Kısacası başlangıç düğümden gidilecek düğüme kadar hareketlerin toplam sayısıdır [21],[20].

$$g(n) = g(n.parent) + cost(n.parent, n) \tag{5.2}$$

Denklem (5.2)'de g(n.parent) ebeveyin düğümünün hareket maliyeti, cost(n.parent, n) ebeveyin düğümünden n düğümüne olan hareket maliyetidir.

5.5.2 H Maliyet Hesabı (H Score)

Sezgisel herbir düğüm ile hedef düğüm arasındaki mesafenin tahmin edilmesi ve kolay hesaplamadır. Her bir düğüm için H maliyeti hedef düğüme ulaşmadan önce en az bir kez hesaplanacağından dolayı bu hesaplama işleminin kolay olması çok önemlidir. H maliyet hesabı arama yapılacak grafin özelliğine bağlı olarak değişebilir. Aşağıda yaygın olarak kullanılan sezgisel hesaplama yöntemleri verilmiştir [20].

Mevcut düğümden hedef düğüme varmak için tahmin edilen mesafe [21]:

- Manhattan Uzaklığı Yöntemi
- Diagonal Uzaklığı Yöntemi
- Öklid Uzaklığı Yöntemi

Yukarıdaki sezgisel fonksiyonların her biri bölüm 5.6 de ayrıntı olarak açıklanmıştır.

5.5.3 F Maliyet Hesabı (F Score)

F maliyet değeri kısaca G ve H maliyet değerleri toplanarak elde edilir. Bu değer mevcut düğümden gidilecek düğüme olan rotanın toplam maliyetini gösterir [20].

f(n) = g(n) + h(n) (5.3)

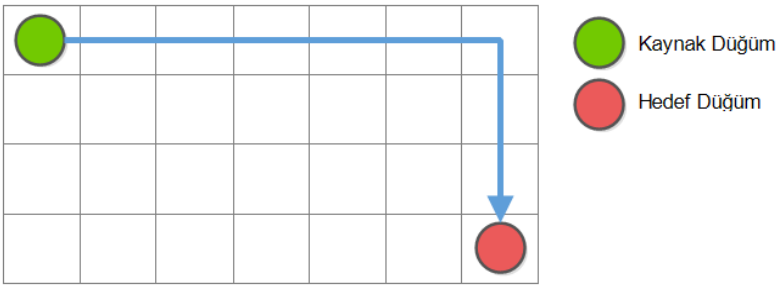
5.6 Sezgisel Fonksiyonlar (Heuristic Functions)

Sezgisel algoritmalar tarafından kullanılan sezgisel fonksiyonlar iki düğüm arasındaki mesafeyi farklı ölçütler kullanarak hesaplarlar. A* algoritması sezgisel fonksiyonları kullanarak en iyi rotayı tahmin etmeye ve algoritmanın performansını arttırmaya çalışır. A* algoritmasında, h(n) sezgisel maliyetin yani hedefe olan uzaklığın hesaplamasında kullanılan yöntemlerden bazıları aşağıda sıralanmıştır.

5.6.1 Manhattan Uzaklığı Yöntemi

Manhattan Uzaklığı Yöntemi, sezgisel maliyet hesaplamalarında kullanılan standart yöntemdir. Mevcut düğüm ile hedef düğüm arasındaki yatay ve dikey mesafelerin toplanmasını esas almaktadır [21],[20].

$$d = |X_2 - X_1| + |Y_2 - Y_1| \quad (5.4)$$



Şekil 5. 1 Manhattan uzaklık yöntemi

5.6.2 Diagonal Uzaklığı Yöntemi

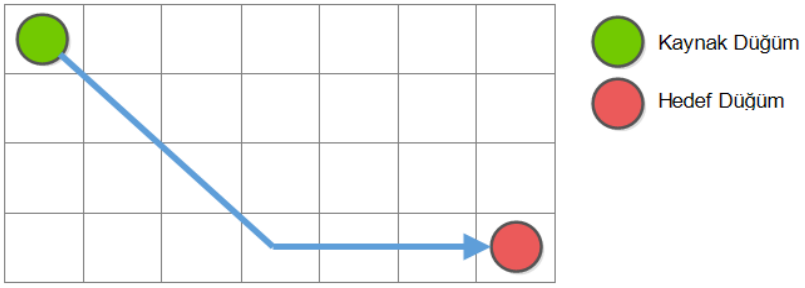
Diagonal Uzaklığı Yöntemi'nde sezgisel maliyeti hesaplama işlemi diğer metotlardan farklı bir şekilde gerçekleşmektedir. Toplam Sezgisel maliyet, yatay ve dikey mesafelerden max ve min olanı bulunur. Aşağıdaki gibi matematiksel işlemlerden geçirildikten sonra toplam sezgisel maliyet elde edilmiş olur. Burada C_d diagonal hareketlerin maliyet ve C_n ise diagonal olmayan hareketlerin toplamıdır [21],[20].

$$d = C_d * d_{min} + C_n (d_{max} - d_{min}) \quad (5.5)$$

$$d_{max} = \max(|X_2 - X_1|, |Y_2 - Y_1|)$$

$$d_{min} = \min(|X_2 - X_1|, |Y_2 - Y_1|)$$

Burada C_d diagonal hareketlerin maliyet ve C_n ise diagonal olmayan hareketlerin toplamıdır.

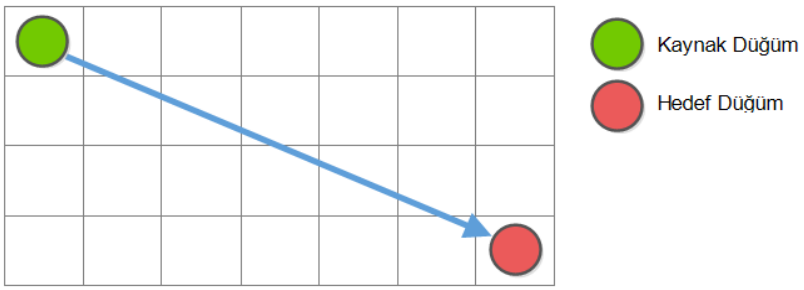


Şekil 5. 2 Diagonal uzaklık yöntemi

5.6.3 Öklid Uzaklığı Yöntemi

Euclidean Uzaklığı Yöntemi, en çok kullanılan sezgisel maliyet hesaplama yöntemlerinden birisidir. Mevcut düğümden hedef düğüme olan yatay ve dikey uzaklıkların karelerinin toplamının karekökü alınarak hesaplanır. Bu uzaklıkların kareleri alınması sebebiyle aykırı değerlerin maliyet fonksiyonuna etkisi Manhattan uzaklığı yöntemine göre daha fazladır [21],[20].

$$d = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \quad (5.6)$$

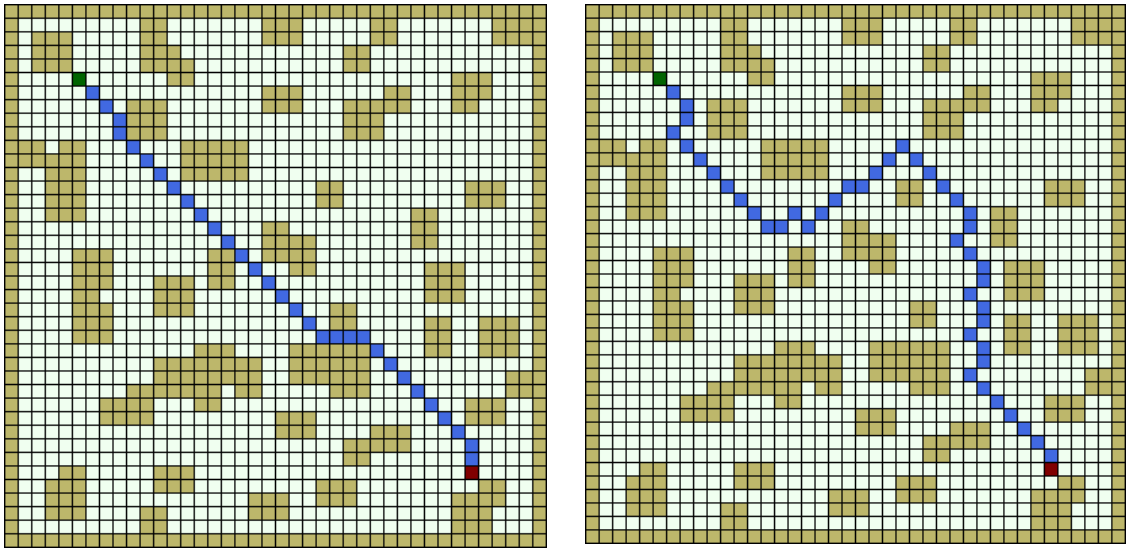


Şekil 5. 3 Öklit uzaklık yöntemi

5.7 Engellerden Kaçınma

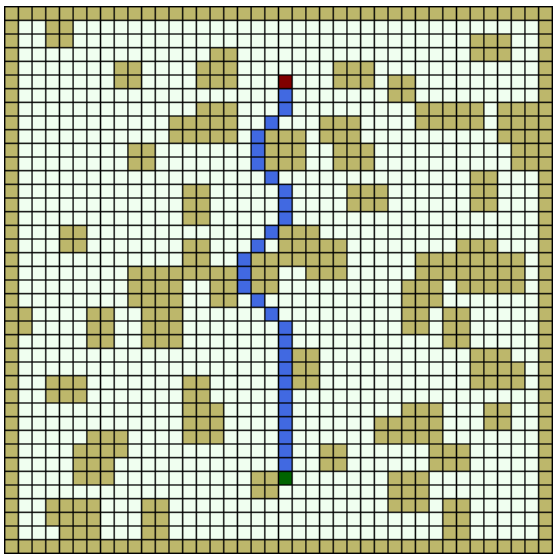
Arama kurtama işlemini yerine getirmek üzere harita üzerinde hedefe noktaya ilerlyen bir robotun engellere çarpmadan hedefe en kısa yoldan gitmesi çok önemlidir. Söz konusu problem, engellerden kaçınma optimizasyon problemi olarak da adlandırılır [22]. A* algoritması engeller bulunan bir haritada belirtilen iki nokta arasında engelleri hesaba katarak en kısa yolu bulur. Bu algoritma çoğu strateji oyunlarında popüler olarak kullanılmaktadır. Örneği starteji oyunlarında düşman askerine sadır komutunu alan bir saldırgan asker en kestirme yoldan düşmana askerine ulaşır ve saldırır. Uygulamamızda, A* algoritmasının bulduğu her yol her zaman en iyi yol değildir. Çünkü A* algoritmasının bulduğu her yoldan robotlar geçemeyebilir. Dolaysıyla robotun gideceği yolun mümkün mertebe dar geçitler olmaması ve yolun uzama ihtimalı olsa bile daha rahat gideceği bir yolun seçilmesi önemlidir. Bu sorunu çözmek için g(n) ve h(n) maliyetlerine ek olarak seçilen noktanın etrafındaki engel sayısını belirli bir k katsayısı ile çarparak engellerden kaçınması sağlandı.

$$f(n) = g(n) + h(n) + k * \text{Engel Sayısı} \tag{5.7}$$

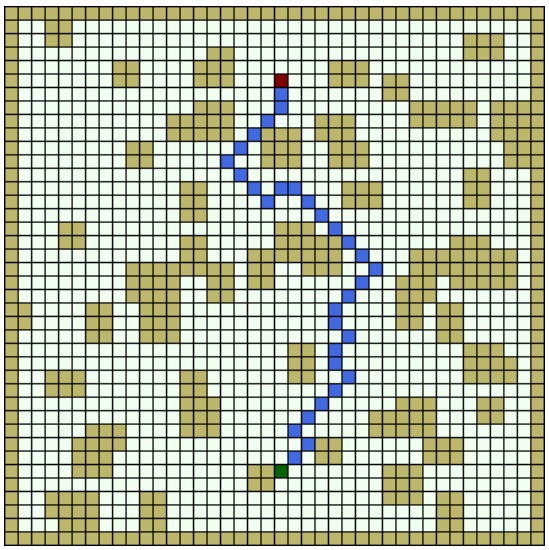


(a) (b)

Şekil 5. 4 Engellerden kaçınma



(c)



(d)

Şekil 5. 5 Engellerden kaçınma (devamı)

5.8 Deneysel Sonuçlar

Astar algoritmasının performansını ölçmek için engellerin bulunma yoğunluğunun farklı olduğu toplam 1000 farklı haritada deneyler gerçeklenmiştir. Haritadaki engeller dörtgen şeklindeki için dolu kutucuklardan oluşur. Bu kutucuklar duvar gibi robotun geçemeyeceği engelleri temsil ederler. Bu kutucukların değişken olup boyutları 5 ile 20 birim arasında değişmektedir. Deneyleri rastgele üretilen sırasıyla 20, 40, 60, 80, 100, 120, 140, 160, 180 ve 200 farklı engel kutucuğundan oluşan 100 farklı harita üzerinde gerçekleştirilmiştir. Yani birinci 100 haritanın her birinde rastgele üretilen 20 faklı engel kutucuğu, ikinci 100 haritanın her birinde rastgele üretilen 40 farklı engel kutucuğu, ... vardır. Her yüzlük gurubun ortalama çalışma zamanı ve ortalama bellek kullanımı ölçülmeye çalışılmıştır. Dolaysıyla deneyler "Zaman Analizi" ve "Bellek Analizi" olmak üzere iki farklı şekilde yapılmıştır.

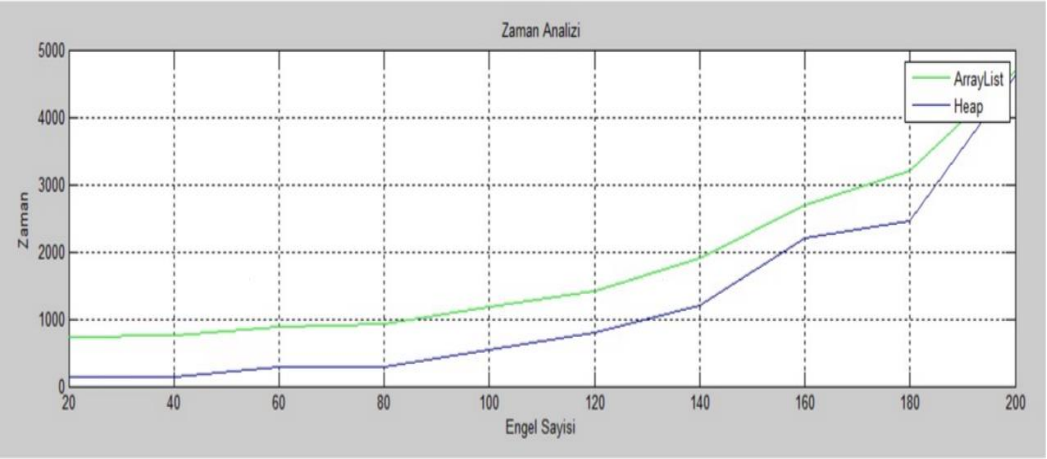
a) Zaman Analizi

Aşağıdaki tabloda zamana göre her deneyin çalışma süresinin performans sonucu verilmiştir. Çizelge 5.1'i ve Şekil 5.9'u incelediğimizde haritadaki engel sayısı seyrek iken heap veri yapısı, array list veri yapısına göre çalışma süresinin çok daha kısa olduğunu ve daha iyi sonuç verdiğini görüyoruz. Engel sayısı haritada arttıkça heap veri yapısının

başarısının array listinkine daha yakın seyrettiğini gözlemliyoruz. Genel olarak bakacak olursak yaptığımız deneylerde engel sayısının çok sık olmadığı durumlar hariç diğer tüm durumlarda heap yapısının en iyi çözüm sunduğunu görüyoruz.

Çizelge 5. 1 Zaman analiz sonuçları

ENGEL SAYISI	HARİTA SAYISI	ARRAY LIST (ms)	HEAP (ms)
20	100	737	130
40	100	756	147
60	100	894	279
80	100	940	292
100	100	1192	540
120	100	1426	797
140	100	1919	1207
160	100	2695	2219
180	100	3208	2470
200	100	4716	4655



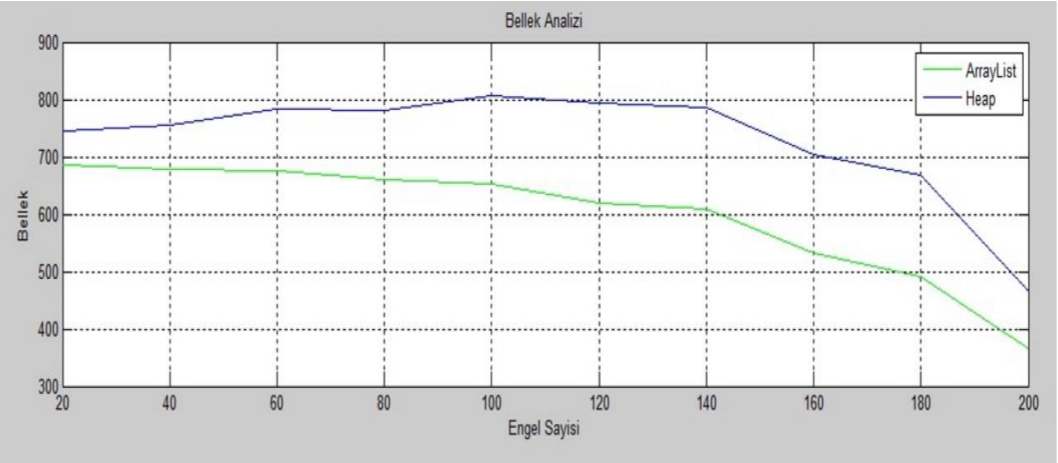
Şekil 5. 6 Çalışma zamanın engel sayısına göre değişimi

b) Bellek Analizi

Aşağıdaki Çizelge 5.2'de her iki yapıya ait bellek kullanım verileri verilmiştir. Çizelge 5.2'i ve Şekil 5.10'u incelediğimizde tüm deneylerde array list veri yapısının heap veri yapısına göre daha az bellek kullandığını görüyoruz. Haritadaki engellerin seyrek olduğu durumlarda her iki yapının bellek kullanımı düşük iken engel sayısı orta seviyede iken maksimum bellek ihtiyaçlarını kullanmaktadırlar. Son olarak haritadaki engellerin çok sık olduğu durumlar için yine her iki yapının bellek kullanımının düştüğü gözlemlenmiştir. Fakat her durumda arraylist, heap'e göre daha az bellek kullanmaktadır.

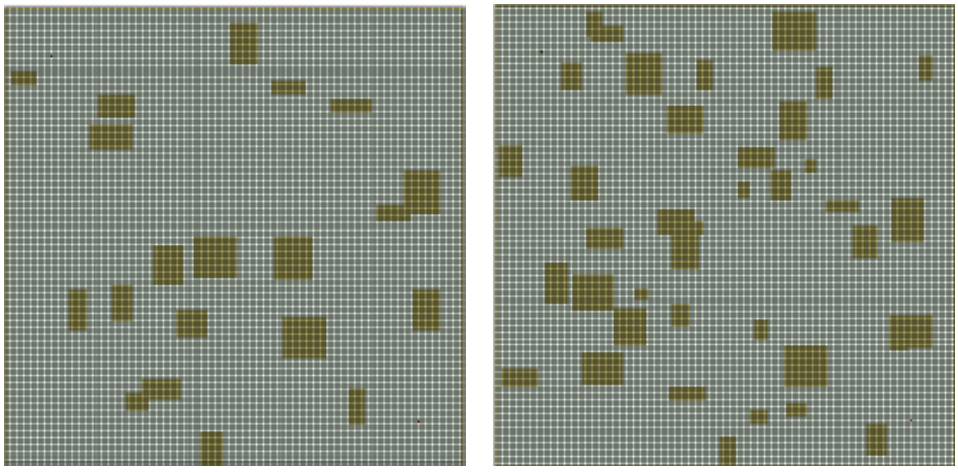
Çizelge 5. 2 Bellek kullanımı analiz sonuçları

ENGEL SAYISI	HARİTA SAYISI	ARRAY LIST (düğüm sayısı)	HEAP (düğüm sayısı)
20	100	687	744
40	100	678	756
60	100	676	784
80	100	661	782
100	100	652	807
120	100	619	793
140	100	608	785
160	100	532	704
180	100	491	669
200	100	366	466



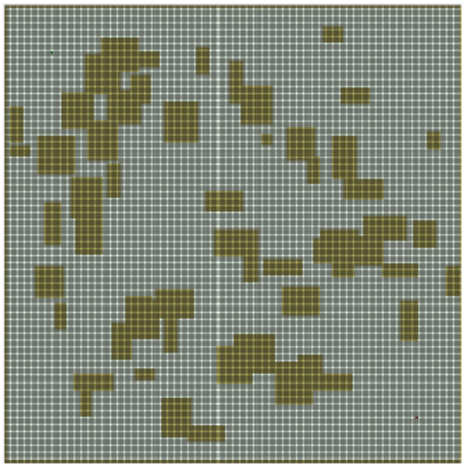
Şekil 5. 7 Bellek kullanımının engel sayısına göre değişimi

Engel sayısına göre bellek kullanım oranı Şekil 5.6'de gösterilmiştir. Haritadeki engel sayısı az iken robotun gidebileceği açık bölgeler daha fazladır. Buna karşılık haritadaki engel sayısı çok fazla olduğunda robotun hareket edebileceği bölgede azdır. Bu nedenle engellerin seyrek bulunduğu haritalarda belleğe eklenecek nokta sayısı çok ve engel yoğun olduğu haritalarda ise belleğe alınacak nokta sayısı azdır. Deney sırasında rastgele üretilen örnek haritalar Şekil 5.7 (a-j)'de gösterilmiştir. Her bir haritada engellerin bulunma yoğunluğu farklıdır. Deneylerin sağlıklı bir şekilde yapılabilmesi ve farklı yoğunlukta haritalarda algoritmanın çalışma süresinin analiz edebilmek için önemlidir.

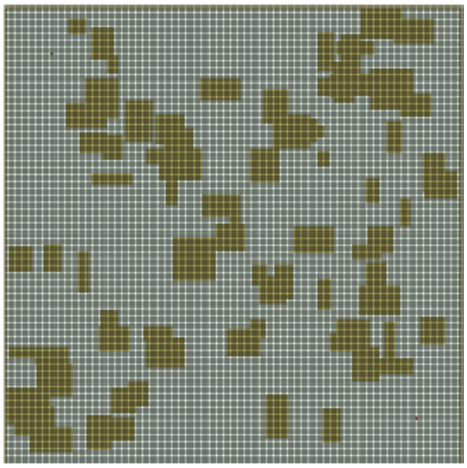


(a) 20 engel kutucuk (b) 40 engel kutucuk

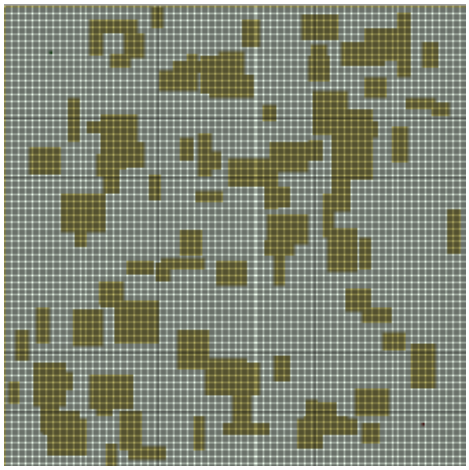
Şekil 5. 8 Farklı yoğunluktaki deneysel haritalar



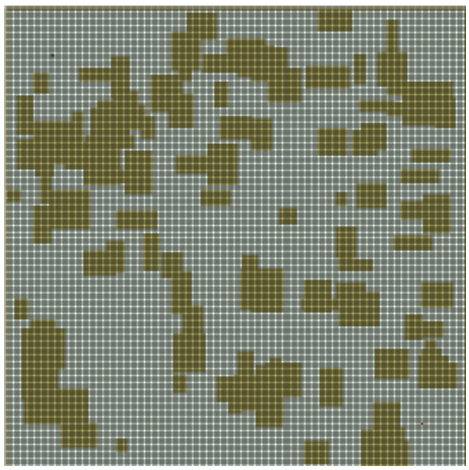
(c) 60 engel kutucuk



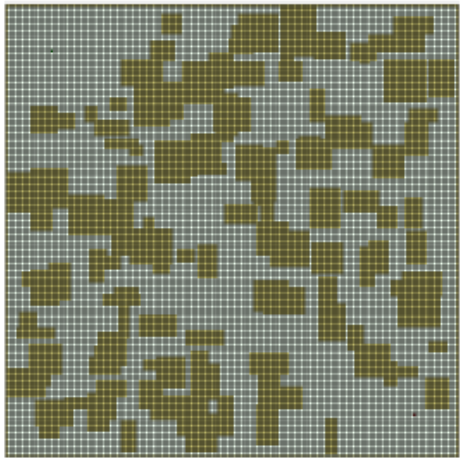
(d) 80 engel kutucuk



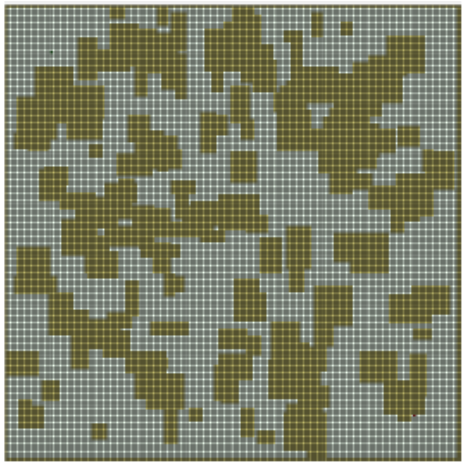
(e) 100 engel kutucuk



(f) 120 engel kutucuk

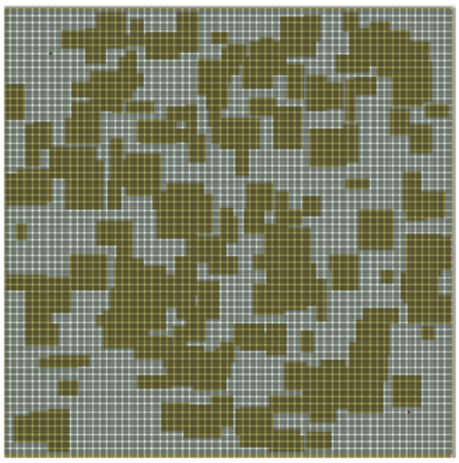


(g) 140 engel kutucuk

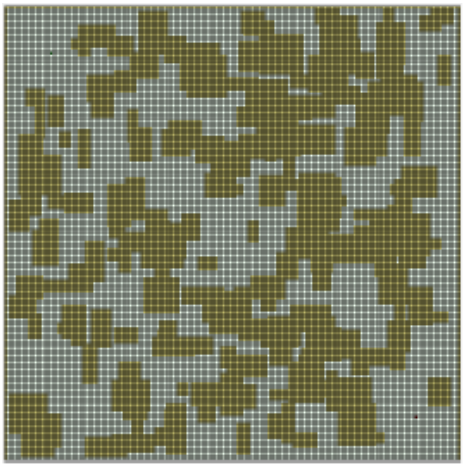


(h) 160 engel kutucuk

Şekil 5. 9 Farklı yoğunluktaki deneysel haritalar (devamı)



(i) 180 engel kutucuk



(j) 200 engel kutucuk

Şekil 5. 10 Farklı yoğunluktaki deneysel haritalar (devamı)

BÖLÜM 6

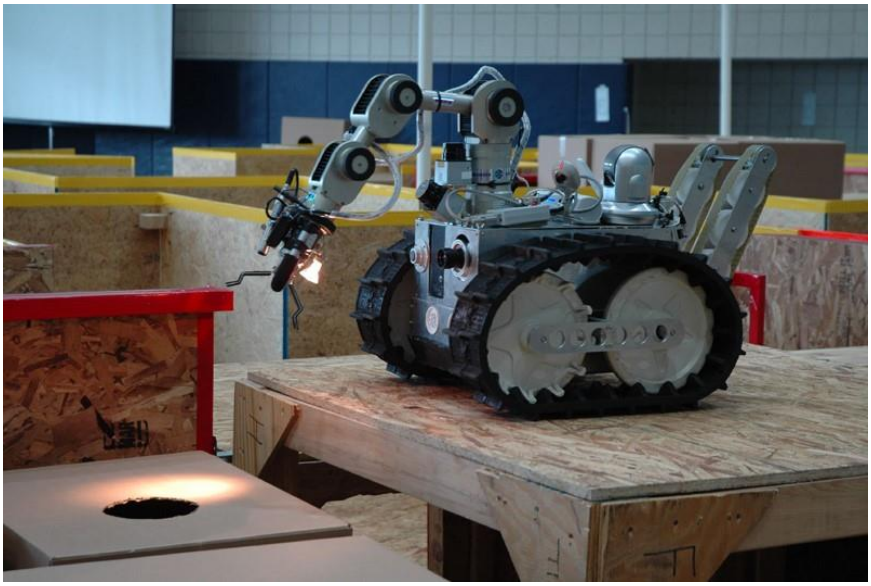
ROBOCUP SANAL ROBOT YARIŞMASI VE SİMÜLASYON ARAÇLARI

RoboCup, 1997 yılında kurulmuş olan uluslararası bir robot yarışmasıdır. Yarışmanın temel amacı araştırmacıların ilgisini çekecek zorlu problemler ortaya koyarak robot bilim ve yapay zeka çalışmalarının ilerlemesini sağlamaktır. Yarışma için gerçekleştirilen araştırmaların hedefi dinamik ortamlarda çalışabilen işbirlikçi çok robotlu ve çok ajanlı sistemlerin geliştirilmesidir. Yarışma kapsamında düzenlenen oyunlar yapay zeka, bilim ve teknoloji eğitimi için katılımcılara büyük bir araştırma fırsatı sağlamaktadır [23].

Yarışmanın nihai hedefi ise 21. yy'ın ortalarına kadar futbol oynayabilen tamamıyla özerk insansı robotların dünya kupasını en son kazanmış olan futbol takımı ile gerçekleştirilecek bir müsabakayı kazanmasını sağlamaktır.

RoboCup Federasyonu başlıca 4 alanda yarışmalar düzenlemektedir ve her bir alan belli sayılarda lig ve alt liglere sahiptir.

- RoboCup Futbol (RoboCup Soccer)
- RoboCup Arama ve Kurtarma (RoboCup Rescue)
- RoboCup Ev (RoboCup @home)
- RoboCup Genç (RoboCup Junior)



Şekil 6. 1 Arama kurtarma işlemi yapan bir robot

Yarışmalar hem bölgesel hem de uluslararası olarak düzenlenmektedir. Uluslararası yarışma her sene farklı bir ülkede gerçekleştirilir. Bölgesel yarışmalar ise Almanya, Çin, İran, Meksika, Amerika Birleşik Devletleri gibi ülkelerde düzenlenmektedir.

6.1 RoboCup Arama ve Kurtarma Ligi

RoboCup arama kurtarma kategorisi Arama Kurtarma Robot Ligi (Rescue Robot League - RRL) ve Arama-Kurtarma Simülasyon Ligi (Rescue Simulation League - RSL) olmak üzere iki alt kategoriye ayrılmaktadır. Çalışmada gerçeklenen uygulama Arama Kurtarma Simülasyon Ligi alt kategorisinde yarışmak üzere geliştirilmiştir. Robot arama ve kurtarma yarışmaları 2000 yılında AAAI Mobil Robot Yarışması ve Sergisi kapsamında başlamış ve daha sonra 2001 yılında RoboCup yarışmasına integre edilmiştir.

1995 yılında, Kobe'deki büyük Hanshin depreminden sonra Japon hükümeti büyük ölçekli felaketlerde arama ve kurtarma işlemlerindeki problemlerin çözümü için ilgili alanlarda çalışma yapılması için araştırmacıları teşvik etmeye karar verir. Bu girişimin sonucu olarak RoboCup Arama ve Kurtarma yarışması ortaya çıkar.

Doğal afetlerin çok sık yaşandığı bölgelerde, zararın en aza indirilmesi önemli bir sosyal konulardan biri haline gelmiştir. Kentsel Arama ve Kurtarma (Urban Search and Rescue - USAR) ile ilgili senaryolar çok ajan ve çoklu robot sistemlerinin araştırılması için büyük

bir potansiyele sahiptir. Gerçek arama ve kurtarma görevlerinin zorlu şartlarda ve ortamlarda güç olduğu için RoboCup yarışmasındaki gibi akıllı robotların yeteneklerinin böylesi yarışmalarda değerlendirilmesi idealdir. Robotların felaket ortamından önceki vaziyette iletişim durumları nasıl ise felaket sonrasında da benzer şekilde iletişim durumunda olmaları gerekir. Benzer şekilde robotlar felaket ortamındaki enkaz altındaki insanları tanıma ve teşhis etme özelliği normal ortamdaki kadar becerikli olması gerekir [24].

6.2 Simülasyon Ortamı ve Araçlar

Gerçek arama kurtarma robotlarının oluşturulması ve test ortamlarının hazırlanması zor ve çok maliyetli olduğundan RoboCup yarışmacılara simülasyon ortamında yarışabilecekleri bir alt kategori ve simülasyon araçları sunmaktadır. Bu bölümde yarışmada kullanılan simülasyon ortamı (USARSim) ve mesaj sunucu aracı (WSS) hakkında bilgiler verilmiştir.

6.2.1 USARSim (Urban Search and Rescue Simulator)

Urban Search and Rescue Simulator (USARSim) yeni geliştirilmekte olan ve oldukça yaygın kullanılan bir mobil robot simülasyon programıdır. Ünlü Unreal 2.0 Motorunun üzerine geliştirilmiştir ve robotların fiziğini güvenilir ve doğru olarak modellemek için Karma fizik motorunu kullanır [25].

USARSim bizim için simülasyon ortamında robotların istenilen pozisyonda oluşturulmasına ve test ortamın hazırlanması için uygun bir zemin hazırlamaktadır. RoboCup yarışmasındaki haritalar, USARSim programında hazırlanan belirli bir felaket ortamı, felaketten etkilenen kurbanlar, aşılması zor yollar vb. bileşenlerden oluşur. Yarışma gereği bu ortamda oluşturulan robotların belirli misyonları vardır. Buradaki amaç robotların verilen misyonu istenilen bir şekilde yerine getirmesidir.

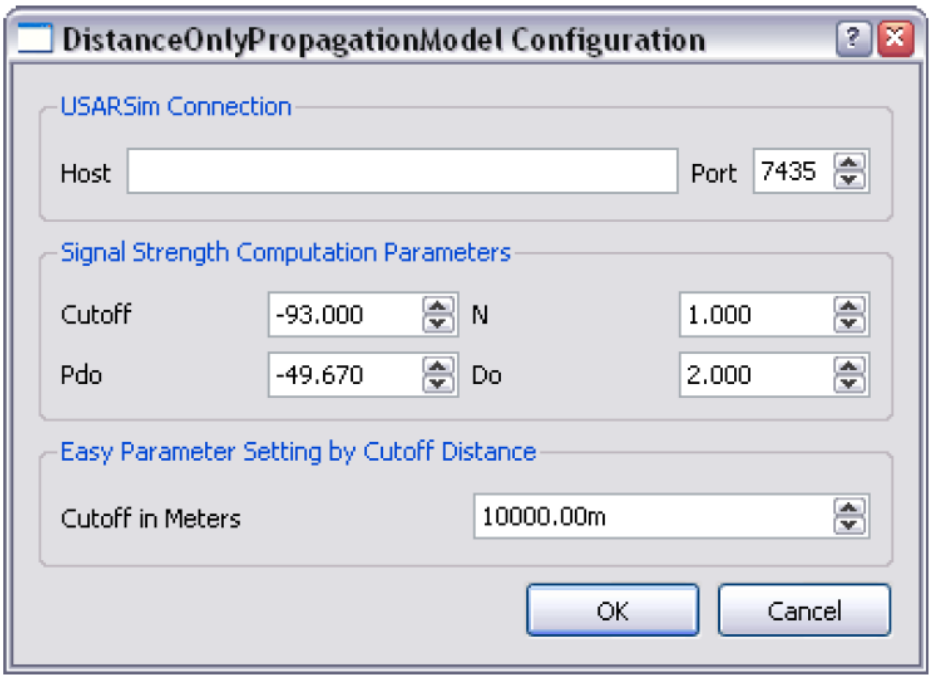
Şekil 6. 2 Örnek bir felaket ortamları

6.2.2 WSS (Wireless Simulation Server)

The Wireless Simulation Server (WSS), RoboCup yarışma ortamındaki sanal robotlar arası iletişimi sağlamak amacı ile Jacobs Üniversitesi tarafından geliştirilen bir yazılımdır. Bu yazılım robotlar ve operatör arası kablosuz iletişimi simüle eder. Yarışma gereği tüm iletişim WSS üzerinden yapılması gerekir. İki düğüm arası sinyal gücünü herhangi bir t anda WSS aracıyla öğrenmek mümkündür. İki düğüm arası bağlantının olup olmadığını öğrenmek için sinyal gücünü belirlenen eşik değeri ile kıyaslamak gerekir. Eğer sinyal gücü eşik (-93 dBm) değerine eşit veya daha büyük ise düğümler arası bağlantı var demektir, aksi durumlarda bağlantının kopmuş olduğunu gösterir [26].

Robotlar birbirleri ile iletişimi sağlamak için önce WSS'in belirli portuna kayıt yaptırması gerekiyor. Bağlantı bir kere kurulduktan sonra WSS robotların mesaj gönderimine ve bağlantısını kapatmasına izin verir. Her mesaj gönderiminden önce kaynak ve hedef düğümleri arası bağlantının olup olmadığına bakar. Eğer sinyal gücü eşik değerinden

daha büyük ise mesaj gönderilir aksi durumlarda mesaj göz ardı edilir ve bağlantı kapatılır [3].



Şekil 6. 3 WSS grafik ara yüzü

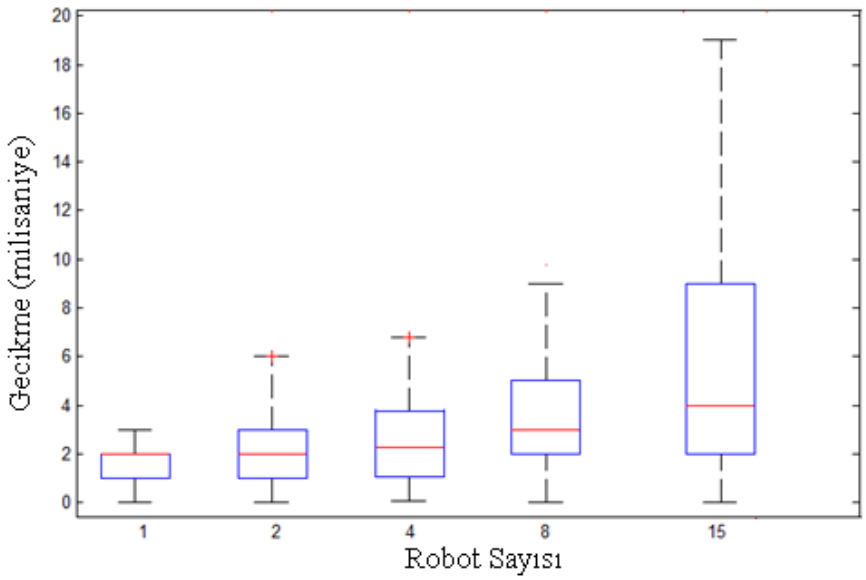
BÖLÜM 7

DENEYSEL SONUÇLAR

Gerçekleştirdiğimiz yönlendirme algoritmaları WSS simülasyon aracının sağladığı iki farklı model üzerinde test edilmiştir. Tüm testlerde USARSim ve WSS simülasyon araçları bir sunucu hizmeti görevini gören bir bilgisayar üzerine kurulmuş ve çalıştırılmıştır. Kodlar ise istemci işlevi gören diğer bir bilgisayarda derlenmiş ve sunucu bilgisayara maksimum 100 Mbps bağlantı hızında bir yönlendirici aracıyla bağlanmıştır.

7.1 Noop Propagation Modeli

WSS simülasyon aracının sağladığı bu model tüm robotların birbirleri ile iletişimde olmasını sağlar. Yönlendirme yöntemlerinin mesajı hedefe ulaştırma hızlarını test etmeden önce tüm robotların ComStation'a direkt bağlı olduğu durumda değişen robot sayılarına göre ortalama mesaj gecikmeleri test edilmiştir. Böylelikle yönlendirme yönteminden bağımsız, en ideal durumda robot sayısı arttığında ortalama mesaj gecikmeleri öğrenilmiş olur. Test senaryosu olarak, Comstation hariç sırasıyla 1, 2, 4, 8 ve 15 robot oluşturuldu. Bu senaryolar için tüm robotlar Comstation ile iletişim halindedir. Her robot 2048 bayt uzunluğundaki mesaj paketini aralıksız olarak ComStation'a göndermekte ve gönderir göndermez mesaj id'sini sistem zamanı ile birlikte loglamaktadır. ComStation ise aldığı mesajın id'si ve alış zamanını loglamaktadır. Her bir robot ComStation'a 20000 mesaj gönderene kadar test sürdürülmüştür.



Şekil 7. 1 Robot sayısına göre mesaj gecikme süresinin kutu grafiği

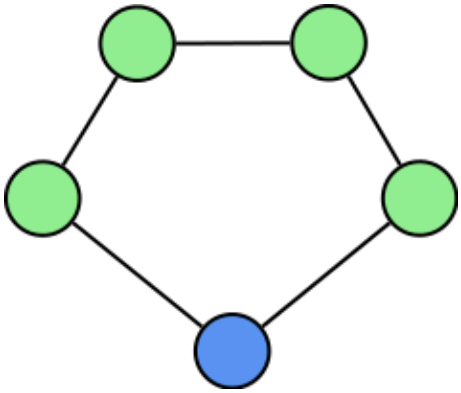
Bu test sonuçlarına göre değişen robot sayısına karşılık mesajın gecikme sürelerinin dağılımının, %50 ve %75 güven aralığındaki yoğunluğu gösteren kutu grafiği şekil 7.1'deki gibi oluşmaktadır. Bu testlere göre hiçbir rotalama yöntemine gerek duyulmadığı durumda, diğer bir ifadeyle noop propagation modeli kullanıldığında mesaj gecikmelerinin robot sayısına göre lineer olarak arttığı, fakat en yüksek sayı olan 15 adet robot kullanıldığında bile mesaj gecikmelerinin 11.04 ms ortalamaya, 22.4 ms standart sapmaya sahip olduğu için robot kontrolünde hiçbir olumsuzlukla karşılaşılmayacağı öngörülmüştür.

7.2 Distance ve Obstacle Propagation Modeli

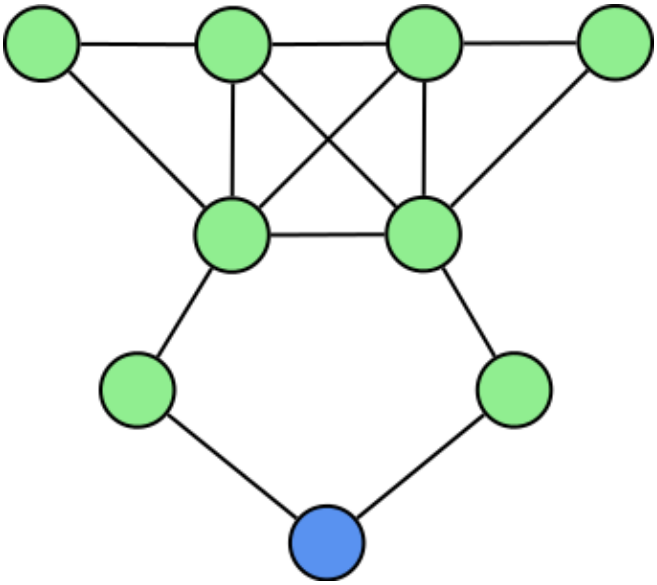
Distance ve Obstacle Propagation Model'lerinde robotlar birbirinden eşik değerinden daha uzakta bulunduğu sürece iletişimde bulunamazlar. Bu durumda robotlar Comstation'a sensör bilgilerini ve ComStation robota kontrol komutlarını diğer robotlar üzerinden ulaştırmalıdırlar. Mesaj paketlerinin hangi rotayı kullanarak hedefe varacağı çok önemlidir. Literatürde baseline yöntem olarak bilinen tüm düğümler kendine gelen paketi iletişimde bulunduğu tüm düğümlere yayması olarak bilinen sel baskını (Blind Flooding - BF) yöntemi ile dinamik olarak robotlar arasındaki en kısa yolu hesaplayıp belirli periyodlarda bu mesajı robotlara yayan tablo-tabanlı (Table-Based - TB) yönlendirme yöntemi kıyaslanmıştır.

Çalışma zamanında robotlar haritanın herhangi bir yerinde bulunabilirler, yani robotların iletişim hatlarını gösteren graf dinamik olarak değişir. Robotların oluşturduğu grafın yapısı değiştikçe test sonuçları da değişeceği açıktır. Adil bir test olabilmesi için ComStation hariç 4, 8, 15 adet robot Şekil 7.2'deki graflardaki gibi bir iletişim hattı oluşturacak şekilde haritaya konumlandırılmıştır.

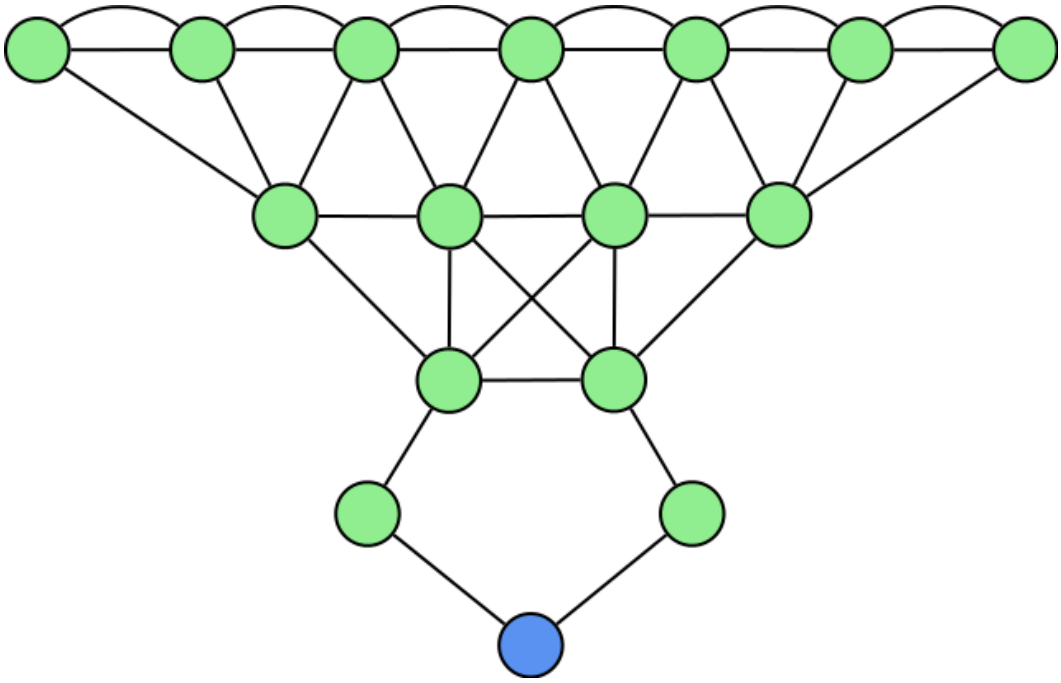
Şekil 7.2'deki graf çift yönlü graf olup, mavi renk ile gösterilen düğüm ComStation'ı temsil eder. Test, sadece robotlardan Comstation'a giden paketlerin birim zamandaki adedi ve gecikmeleri hesaplanmak suretiyle yapılmıştır. Test aşamasında sel algoritması yönteminde ComStation'dan robotlara hiç mesaj gitmezken, tablo-tabanlı yöntemde belirli bir aralıkta ComStation dinamik olarak hesapladığı yönlendirme tablosunu robotlara göndermektedir.



(a) Senaryo I



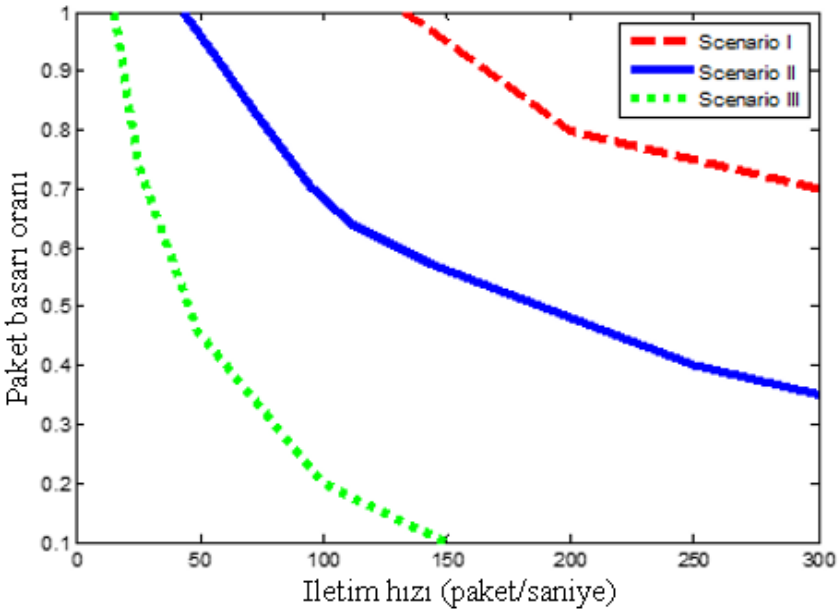
(b) Senaryo II



(c) Senaryo III

Şekil 7. 2 Test senaryoları

Bir felaket ortamında arama kurtarma görevi yapan robot takımının mümkün olduğunca hızlı bir şekilde sensör bilgilerini ComStation'a göndermesi gerekir. Yaptığımız deneylerde her iki yönlendirme algoritması için buffer taşmasından ötürü oluşan ağdaki sistematik gecikmeler ve paket kayıplarını önlemek için optimum veri gönderimi (paket sayısı) araştırılmıştır. Ağdaki buffer taşmalarını incelemek için sıralı mesaj paketlerine farklı gecikme süresi verilmiştir. Şekil 7.3'de her üç senaryo için Sel algoritması ile elde edilen veri gönderim oranına karşılık başarılı paket gönderim oranı gösterilmiştir.



Şekil 7. 3 Sel algoritması için başarılı paket gönderim oranı (paket/saniye)

Şekil 7.3'de sel algoritması birim zamandaki paket gönderim hızına göre başarılı paket gönderme başarısı özetlenmiştir. Örneğin senaryo 1 için yaklaşık 150 paket/sn hızında paket gönderimi yaparsa paket başarı oranı yüzde yüzdür. Fakat bu oran senaryo 2 için yaklaşık olarak 50 paket/sn civarındadır. Son olarak senaryo 3 için 20 paket/sn hızında paket gönderimi yaparsa gönderim başarı oranı yüde yüze yakın olur. Şekil 7.3'deki tüm senaryolar için sistemin sağladığı maksimum paket gönderim oranına karşılık tablo-tabanlı yönlendirme yönteminde herhangi bir sistematik gecikme veya paket kaybı oluşmamıştır.

Çizelge 7. 1 Maksimum paket gönderim hızı sonuçları

SENARYO	YÖNTEM	TOPLAM PAKET HIZI (pk/sn)	HER BİR ROBOT İÇİN PAKET HIZI (pk/sn)
I	Sel	132.45	33.11
I	Tablo Tabanlı	398.32*	99.58
II	Sel	43.52	5.44
II	Tablo Tabanlı	312.76*	39.09
III	Sel	15.16	1.01
III	Tablo Tabanlı	229.11*	15.27

Deneyler her bir robotun ComStationa 5000 mesaj paketini gönderene kadar sürdürülmüştür. Toplam gönderilen paket oranı ve her bir robotun paket gönderim oranı aşağıda verilmiştir.

Çizelge 7.1'de, (*) simgesi sistemin sağladığı maksimum paket gönderim hızını göstermektedir. Tablo-tabanlı yöntem kullanıldığında ne paket kaybı ne de sistematik gecikme ağda gözlenmiştir. (*) işareti ile gösterilen oranlar gerçek oranlar değildir. Bu oran hızlı veri oluşumuna göre artabilir. Sel Algoritması yönteminde gereksiz pek çok mesajlaşmalar bant genişliğinin gereksiz kullanılmasına yol açar. Sel algoritması yönlendirme işlemi için kullanıldığında buffer taşmasını önlemek için her bir robot 1.01 pk/sn den fazla paket üretmemesi ve göndermemesi gerekir. Bu rakam da yaklaşık olarak tablo-tabanlı yöntemin 15 katından daha fazladır.

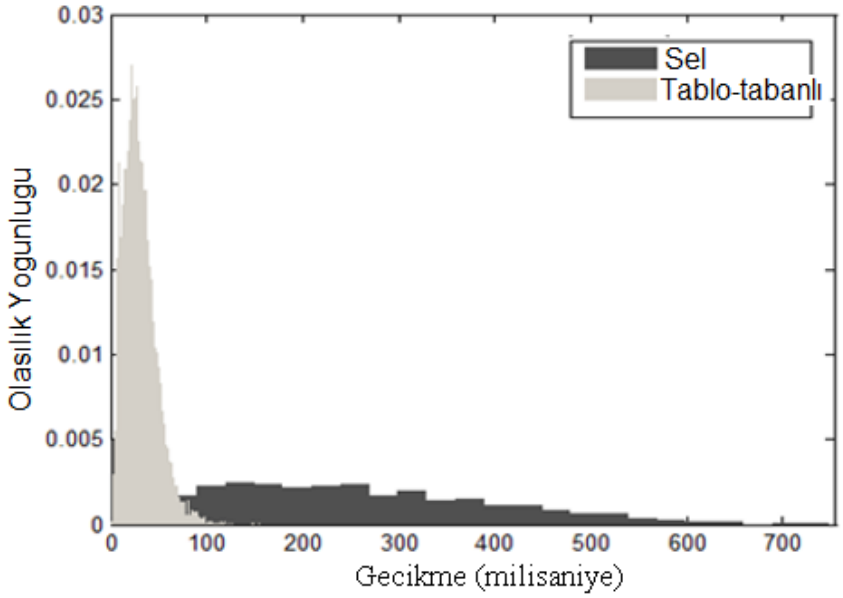
Her bir senaryo için robotların ComStationa gönderdiği paketlerin gecikmesini ölçmek için robotların ürettiği paket oranı daha önce analiz ettiğimiz oranı geçmemesi gerekir. Deneyler her bir robotun ComStationa 5000 mesaj paketi gönderene kadar sürdürülmüş ve her bir mesaj paketinin gecikme süresi hesaplanmıştır.

Çizelge 7. 2 Paket gecikme sonuçları

SENARYO	YÖNTEM	EN KÖTÜ ORT. GECİKME (ms)	EN İYİ ORT. GECİKME (ms)	ORT. GECİKME SÜRESİ (ms)	STANDART SAPMA (ms)
I	Sel	29.75	10.03	18.46	16.80
I	Tablo Tabanlı	10.10	5.27	7.67	6.58
II	Sel	91.39	11.07	52.41	42.45
II	Tablo Tabanlı	20.56	7.27	15.32	10.46
III	Sel	361.87	15.25	218.74	155.77
III	Tablo Tabanlı	42.10	13.16	30.40	17.36

Deney sonuçları Çizelge 7.2'de verilmiştir. Çizelgedeki sonuçlara göre ağdaki düğüm sayısı az iken sel algoritması, tablo-tabanlı algoritmaya benzer sonuç verdiği görülür.

Fakat düğüm sayısının çok ve düğümler arası bağlantının da daha fazla olduğu ağlarda sel algoritmasının gecikme süresi üstel olarak artmaktadır. Buna karşın tablo-tabanlı yönteminde gecikme lineer olarak artmaktadır. Gecikmenin fazla olması bazı uygulamalar için kabul edilemez bir durumdur. Şekil 7.4'de sel algoritması ve tablo-tabanlı yönteminin Senaryo 3 için oluşturduğu paket gecikmelerinin histogramı verilmiştir.



Şekil 7. 4 Sel ve Tablo tabanlı yöntemlerin paket gecikmelerinin histogramı

Şekil 7.4'deki histogram dağılımını incelediğimizde siyah renkli bölge sel algoritması yönteminin ve gri renkli bölge ise tablo-tabanlı yöntemin paketleri hedefe ulaşma olasılığına karşılık paketlerin gecikme süreleri dağılımlarını göstermektedir. Her bir bölgenin toplam olasılığı 1 dir. Dolaysıyla bu dağılıma göre sel yönlendirme yönteminin paket gecikme süreleri yüksek olması nendeniyle paketlerin hedefe ulaşma olasığı da düşüktür. Buna karşın tablo-tabanlı yöntemde ise paket gecikme süreleri az olması nedeniyle paketlerin hedefe ulaşma olasığı da yüksektir. Sel algoritması için paket gecikme aralığı 0-700 ms arasında değişirken, bu aralık tablo-tabanlı yöntem için 0-120 ms civarındadır.

7.3 Sonuç

Gezgin tasarsız ağlar, iletişimin tesisinin zor ve engebeli arazilerde çok robotlu sistemlerde robotlar arası iletişim için umut vadeden bir ağ teknolojisidir. Robotlar arası iletişimde yaygın olarak kullanılan gezgin tasarsız ağlar geliştirilmekte olan uygulama alanıdır. Düğüm sayısının çok az olduğu ve tampon taşma durumunun olmadığı tasarsız ağlarda sel yönlendirme yaklaşımı tercih edilebilir. Çünkü bu yaklaşımın gerçeklenmesi kolay ve tüm mimarilerde mesaj paketlerinin hedef düğüme ulaşma olasılığının yüksek olmasıdır. Bu yaklaşımın en büyük avantajlarından biri de ağ ile ilgili bilgiye ve ekstradan yapılandırma mesajlarına gerek duymamasıdır. Ancak paket gecikmelerinin karesel olarak artmasına dikkat edilmelidir. Çünkü bu gecikmeler paket gönderim hızının düşmesine sebep olur. Düğümler arası bağlantıların artması ile paket gecikmeleri de doğrusal olarak artar. Yüksek paket iletim hızı tablo-tabanlı yönlendirme yönteminin önemli avantajlarından biridir. Ancak yöntemin çalışma prensibi gereği mesajlar ve dinamik yönlendirme tablolarının periyodik olarak diğer düğümlere gönderilmesi daha fazla bant genişliğinin kullanımına yol açar. Ayrıca, eğer mesaj gönderim sıklığı çok düşük olursa paketlerin kayıp olma olasılığı da artabilir. Çünkü yönlendirme tabloları robotlara zamanında iletilmez veya hiç iletilmez. Tablo-tabanlı yönlendirme algoritmasının simülasyon ve gerçek felaket ortamlarında en iyi sonuç vermesi için bu değerleri dengeleyebilmelidir.

KAYNAKLAR

[1] Wang, Z., Zhou, M.C. ve Ansari N., (2003). "Ad-hoc robot wireless communication", Systems, Man and Cybernetics, 4: 4045-4050.

[2] Wahi, C. ve Sonbhadra, S.K., (2012). "Mobile Ad Hoc Network Routing Protocols: A Comparative Study", International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC), 3: 21-31.

[3] Nevatia, Y., (2007). "Ad-Hoc Routing for USARSim", Networks and Distributed Systems Seminar, 15 May 2007, Bremen.

[4] ODTÜ Bilgisayar Topluluğu Elektornik dergisi, Ağ Teknolojileri Tarihçesi, http://e-bergi.com/2007/Ekim/Ag-Teknolojileri-Tarihcesi, 8 Şubat 2013.

[5] T.C. Sakarya Üniversitesi, Kablosuz Ağ teknolojileri, https://dosya.sakarya.edu.tr/Dokumanlar/2013/302/485459235_kablosuz_ag_teknolojileri.pdf, 8 Şubat 2013.

[6] T.C. Milli Eğitim Bakanlığı, (2011). Bilişim Teknolojileri: Kablosuz Ağlar, Ankara.

[7] Gorantala, K., (2006). Routing Protocols in Mobile Ad-hoc Networks, Master Thesis, UMEA University, Sweden.

[8] de Morais Cordeiro, C. ve Agrawal, D.P., (2002). "Mobile ad hoc networking. Center for Distributed and Mobile Computing", OBR Research Center for Distributed and Mobile Computing, University of Cincinnati, Cincinnati.

[9] Wang, Z., Liu, L. ve Zhou, M.C., (2005). "Protocols and applications of ad-hoc robot wireless communication networks: An overview", International Journal of Intelligent Control and Systems, 10: 296- 303.

[10] Dhenakaran, S. S. ve Parvathavarthini, A., (2013). "An Overview of Routing Protocols in Mobile Ad-Hoc Network", International Journal of Advanced Research in Computer Science and Software Engineering, 3.2.

[11] Shrivastava, A., Shanmogavel, A.R., Mistry, A., Chander, N., Patlolla, P. ve Yadlapalli, V., (2005). "Overview of Routing Protocols in MANET's and Enhancements in Reactive Protocols", Lamar University, Beaumont.

[12] Zeiger, F., Kraemer, N. ve Schilling, K., (2008). "Commanding mobile robots via wireless ad-hoc networks - A comparison of four ad-hoc routing protocol

implementations", Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on. IEEE, 19-23 May 2008, Pasadena.

[13] T.C. Milli Eğitim Bakanlığı, (2011). Bilişim Teknolojileri: Yönlendirme Temelleri, Ankara.

[14] Giudici, F., (2008). Broadcasting in Opportunistic Networks, Doctoral Thesis, University of Milan, Milan.

[15] Wikipedia, Yönlendirme algoritmaları, http://tr.wikipedia.org/wiki/Y%C3%B6nlendirme_algoritmalar%C4%B1, 11 Nisan 2013.

[16] Mohapatra, P. ve Krishnamurthy, S.V., (2005). Ad Hoc Networks: Technologies and Protocols, Springer US.

[17] Wikipedia, Dijkstra's algorithm. Dutch scientist Dr. Edsger Dijkstra network algorithm, http://en.wikipedia.org/wiki/Dijkstra's_algorithm, 15 May 2013.

[18] Mathematics & Computer Science, Dijkstra's Algorithm, http://www.mathcs.emory.edu/~cheung/Courses/455/Syllabus/5a-routing/dijkstra.html, 21 May 2013.

[19] Wikipedia, Sezgisel algoritma, http://tr.wikipedia.org/wiki/Sezgisel_algoritma, 22 Mayıs 2013.

[20] Growing with the web, The A* pathfinding algorithm, http://www.growingwiththeweb.com/2012/06/a-pathfinding-algorithm.html, 22 Mayıs 2013.

[21] Bilgisayar Mühendisiyiz, A-star Algoritması, http://www.bilgisayarmuhendisiyiz.biz/a-star-algoritmasi/, 15 Eylül 2013.

[22] Arıcı, V., (2008). Engellerin Bulunduğu Ortamda Gezgin Robotun En İyi Yolu Bulması ve İzlemesi, Yüksek Lisans Tezi, Başkent Üviversitesi, Ankara.

[23] Wikipedia, RoboCup, http://en.wikipedia.org/wiki/RoboCup, 10 Eylül 2013.

[24] Akın, H.L., Ito, N., Jacoff, A., Kleiner, A., Pellenz, J. ve Visser, A., (2013). "RoboCup Rescue Robot and Simulation Leagues", AI Magazine, 34: 78-86.

[25] Stefano, C., Lewis, M., Wang, J., Balakirsky, S. ve Scrapper, C., (2007). "USARSim: a robot simulator for research and education", Robotics and Automation, 10-14 April 2007, Roma.

[26] Pfingsthorn M., (2008). "RoboCup Rescue Virtual Robots: Wireless Simulation Server Documentation", 8 October 2008, Bremen.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : KHUDAYDAD MAHMOODI

Doğum Tarihi ve Yeri : 02.05.1984

Yabancı Dili : FARSÇA, TÜRKÇE, İNGİLİZCE

E-posta : khudaydad.mahmoodi@gmail.com

ÖĞRENİM DURUMU

Derece	**Alan**	**Okul/Üniversite**	**Mezuniyet Yılı**
Lisans	Bilgisayar Müh.	Yıldız Teknik Üniversitesi	2010
Lise	Sayısal	Afgan-Türk Dostluk Lisesi	2002

İŞ TECRÜBESİ

Yıl	**Firma/Kurum**	**Görevi**
2010	Pozitim Teknoloji A.Ş.	Programcı

YAYINLARI

Bildiri

1. Khudaydad Mahmoodi, Muhammet Balcılar, M. Fatih Amasyalı, Sırma Yavuz, “Routing with Dijkstra in Mobile Ad-Hoc Networks”, The 17th annual RoboCup International Symposium, Eindhoven,

Temmuz 2013.

2. Yücel Uzun, Muhammet Balcılar, Khudaydad Mahmoodi, Feruz Davletov, M. Fatih Amasyalı, Sırma Yavuz, “Usage of HoG (Histograms of Oriented Gradients) Features for Victim Detection at Disaster Areas”, 8th International Conference On Electrical and Electronics Engineering (Eleco 2013), Bursa, Kasım 2013.

Proje

1. Afet Robotları İçin Algoritma Tasarımı, YTÜ-BAPK, Proje No: 2013-04-01-KAP01, Yürütücü: Yrd. Doç. Dr. Sırma Yavuz, Proje Ekibi: Yrd. Doç. Dr. Fatih Amasyalı, Arş. Gör. Muhammet Balcılar, Arş. Gör. Erkan Uslu, Khudaydad Mahmoodi, Gürkan Şahin.

ÖDÜLLERİ

1. RoboCup 2013 Dünya Şampiyonası İkincilik Ödülü, Virtual Robot Competition, Rescue Simulation League, Takım Üyesi, Hollanda, 2013.

2. RoboCup İranOpen 2013 Birincilik Ödülü, Virtual Robot Competition, Robocup İranOpen 2013, Rescue Simulation League, Takım Üyesi, İran, 2013.