



**İSTANBUL ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS TEZİ**

**KABLOSUZ ALGILAYICI AĞLARINDA KÜMELEME VE  
YÖNLENDİRME ALGORİTMA ANALİZİ**

**Bil.Müh.  
AHMED M.M.J.ALABBAS  
Bilgisayar Mühendisliği Anabilim Dalı**

**Danışman  
Prof.Dr. Abdül Halim ZAIM**

**Temmuz, 2011**

**İSTANBUL**



**İSTANBUL ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS TEZİ**

**KABLOSUZ ALGILAYICI AĞLARINDA KÜMELEME VE  
YÖNLENDİRME ALGORİTMA ANALİZİ**

**Bil.Müh.  
AHMED M.M.J ALABBAS  
Bilgisayar Mühendisliği Anabilim Dalı**

**Danışman  
Prof.Dr. Abdül Halim ZAIM**

**Temmuz, 2011**

**İSTANBUL**

Bu çalışma 14/09/2011 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği programında Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Jürisi

Prof.Dr. Abdül Halim ZAIM (Danışman)  
İstanbul Üniversitesi  
Mühendislik Fakültesi

Prof. Dr. Ahmet SERTBAŞ  
İstanbul Üniversitesi  
Mühendislik Fakültesi

Prof. Dr. Selim AKYOKUŞ  
Doğuş Üniversitesi  
Mühendislik Fakültesi

Yrd.Doç.Dr.Oğuzhan ÖZTAŞ  
İstanbul Üniversitesi  
Mühendislik Fakültesi

Yrd. Doç. Dr. Derya YILTAŞ KAPLAN  
İstanbul Üniversitesi  
Mühendislik Fakültesi

## **ÖNSÖZ**

Lisans ve yüksek lisans öğrenimim sırasında ve tez çalışmalarım boyunca gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Prof. Dr. Abdül Halim ZAIM'e en içten dileklerle teşekkür ederim .

Bu çalışma boyunca yardımlarını esirgemeyen çalışma arkadaşlarıma Ali Noruzi , Mostafa K.A.Al-Harbawi ve çalışmamın uygulama kısmını destekleyen İstanbul Üniversitesi'ne teşekkürü borç bilirim.

**2011**

**Ahmed M.M.J ALABBAS**

# İÇİNDEKİLER

ÖNSÖZ .....	i
İÇİNDEKİLER .....	ii
ŞEKİL LİSTESİ .....	vi
TABLO LİSTESİ .....	viii
ÖZET .....	ix
SUMMARY .....	x
<b>1. GİRİŞ .....</b>	<b>1</b>
<b>1.1. TANIMLAR .....</b>	<b>1</b>
<b>1.2. ALGILAYICI DÜĞÜM MİMARİSİ .....</b>	<b>2</b>
<b>1.2.1. Platformlar .....</b>	<b>2</b>
<b>1.2.2. Algılayıcı Düğüm Donanımı.....</b>	<b>2</b>
<i>1.2.2.1. Algılayıcı Birim .....</i>	<i>3</i>
<i>1.2.2.2. İşlem Birimi .....</i>	<i>3</i>
<i>1.2.2.3. İletişim Birimi (Alıcı –Verici) .....</i>	<i>4</i>
<i>1.2.2.4. Güç Birimi .....</i>	<i>4</i>
<b>1.2.3. Yazılım.....</b>	<b>5</b>
<b>1.3. KABLOSUZ ALGILAYICI AĞ MİMARİSİ.....</b>	<b>6</b>
<b>1.4. KABLOSUZ ALGILAYICI AĞLARIN (WSN) AVANTAJLARI.....</b>	<b>6</b>
<b>1.5. TASARSIZ(AD-HOC)AĞLAR VE KABLOSUZ ALGILAYICI AĞ(WSN)         PROTOKOLLERİ ARASINDAKİ FARKLAR .....</b>	<b>8</b>
<b>1.6. WSN PROTOKOL YIĞINI .....</b>	<b>9</b>
<b>1.7. KABLOSUZ ALGILAYICI AĞLARININ UYGULAMALARI .....</b>	<b>10</b>
<b>1.7.1. Sağlık Hizmetleri.....</b>	<b>10</b>
<b>1.7.2. Tarımsal Uygulamalar .....</b>	<b>11</b>
<b>1.7.3. Endüstriyel Uygulamalar .....</b>	<b>12</b>
<b>1.7.4. Askeri Uygulamalar .....</b>	<b>12</b>

1.7.5. Çevresel İzleme.....	13
1.7.6. Kamusal Kullanımlar.....	4
<b>2. KÜMELEME .....</b>	<b>15</b>
<b>2.1. KÜMELEME TEKNİKLERİNİN KATEGORİSİ.....</b>	<b>16</b>
2.1.1. Ağ Modeli .....	16
2.1.2. Kümeleme Hedefleri.....	17
2.1.3. Küme Özellikleri.....	18
2.1.4. Küme Başının Yeterlilikleri.....	19
2.1.5. Kümeleme Süreci .....	20
<b>2.2. KABLOSUZ ALGILAYICI AĞLARDA KÜMELEME</b>	
<b>ALGORİTMALARININ SINIFLANDIRMASI .....</b>	<b>20</b>
<b>2.2.1. Birinci Kategori .....</b>	<b>21</b>
2.2.1.1. Deneyimsel Algoritmalar .....	21
2.2.1.2. Ağırlıklı Şemalar.....	25
2.2.1.3. Hiyerarşik Şema.....	26
2.2.1.4. Izgara Şemalar .....	31
<b>2.2.2. İkinci Kategori.....</b>	<b>32</b>
2.2.2.1. Değişken Yakınsama Zaman Algoritmaları .....	32
2.2.2.1. Sabit Yakınsamalı Zaman Algoritmaları .....	33
<b>3. YÖNLENDİRME .....</b>	<b>36</b>
<b>3.1. YÖNLENDİRME TASARIMI İÇİN ÖNEMLİ PARAMETRELER.....</b>	<b>36</b>
<b>3.2. YÖNLENDİRME PROTOKOLARININ KATEGORİSİ .....</b>	<b>39</b>
<b>3.2.1. Ağ Yapısı .....</b>	<b>39</b>
3.2.1.1. Veri Merkezli Protokoller (Düz protokol) .....	39
3.2.1.2. Hiyerarşik Tabanlı Protokoller.....	48
3.2.1.3. Konum Tabanlı(Location based) .....	52
<b>3.2.2. Yönlendirme Keşfi Protokolleri.....</b>	<b>60</b>
3.2.2.1. Proaktif Protokol .....	60

3.2.2.2. <i>Reaktif Protokol</i> .....	60
3.2.2.3. <i>Hibrid Protokol</i> .....	60
<b>3.2.3. Yönlendirme İşlemi .....</b>	<b>60</b>
3.2.3.1. <i>Anlaşma Tabanlı</i> .....	60
3.2.3.2. <i>Sorgu Tabanlı</i> .....	62
3.2.3.3. <i>QoS (Servis Kalitesi) Tabanlı Protokol</i> .....	63
3.2.3.4. <i>Çoklu Yol Yönlendirme</i> .....	66
3.2.3.5. <i>Coherent And Non-Coherent Based Protocol</i> (Tutarlı ve Tutarsız Tabanlı Protokol) .....	67
<b>4. ENERJİ VERİMLİ YÖNLENDİRME PROTOKOLARIN</b> <b>KARŞILAŞTIRILMASI .....</b>	<b>69</b>
<b>4.1. LEACH PROTOKOLÜ .....</b>	<b>70</b>
4.1.1. <i>LEACH'in Avantajları</i> .....	74
4.1.2. <i>LEACH'in Dezavantajları</i> .....	75
<b>4.2. DIRECTED DIFFUSION .....</b>	<b>76</b>
4.2.1. <i>Directed Diffusion'ın Avantajları</i> .....	80
4.2.2. <i>Directed Diffusion'ın Dezavantajları</i> .....	80
<b>4.3. GOSSIPING PROTOKOLÜ.....</b>	<b>81</b>
4.3.1. <i>Gossiping Avantajları</i> .....	82
4.3.2. <i>Gossiping Dezavantajları</i> .....	82
<b>4.4. EESR PROTOKOLÜ.....</b>	<b>85</b>
4.4.1. <i>Algoritmanın Çalışması</i> .....	87
4.4.2. <i>EESR' in Avantajları</i> .....	88
4.4.3 <i>EESR' in Dezavantajları</i> .....	89
<b>4.5. YÖNLENDİRME PROTOKOLLERİNİN</b> <b>KARŞILAŞTIRILMASI .....</b>	<b>91</b>
<b>5. FELGOSSIPING PROTOKOLÜ.....</b>	<b>96</b>
<b>5.1. AĞ BAŞLATMA FAZİ .....</b>	<b>97</b>
<b>5.2. BİLGİ TOPLAMA FAZİ .....</b>	<b>99</b>

<b>5.3. YÖNLENDİRME FAZİ .....</b>	<b>99</b>
<b>5.4. SİMÜLASYON KURULUMU .....</b>	<b>103</b>
<b>5.5. PERFORMANS PARAMETRELERİ .....</b>	<b>106</b>
<b>5.5.1. Genel ağda enerji tasarrufu .....</b>	<b>106</b>
<b>5.5.2. Uçtan-uca gecikme ortalaması .....</b>	<b>106</b>
<b>5.5.3. Kayıp paket sayısı .....</b>	<b>107</b>
<b>5.5.4. Atlama sayısının ortalaması.....</b>	<b>107</b>
<b>5.6. SİMÜLASYON SONUÇLARI .....</b>	<b>107</b>
<b>5.6.1. Genel ağda toplam enerji tasarrufu .....</b>	<b>108</b>
<b>5.6.2. Uçtan uca gecikme ortalaması .....</b>	<b>109</b>
<b>5.6.3. Atlama sayısı ortalaması .....</b>	<b>110</b>
<b>5.6.4. Kayıp Paket sayısı .....</b>	<b>111</b>
<b>6. TARTIŞMA VE SONUÇ .....</b>	<b>112</b>
<b>KAYNAKLAR .....</b>	<b>114</b>
<b>EKLER .....</b>	<b>120</b>
<b>ÖZGEÇMİŞ .....</b>	<b>136</b>



## ŞEKİL LİSTESİ

Şekil 1.1	: Algılayıcı Düğüm Platformu .....	2
Şekil 1.2	: Algılayıcı düğüm mimarisi .....	3
Şekil 1.3	: Genel Kablosuz ağ mimarisi .....	6
Şekil 1.4	: WSN Protokol yığını .....	10
Şekil 1.5	: WSN sağlık uygulaması.....	11
Şekil 1.6	: WSN Tarımsal kullanımı .....	11
Şekil 1.7	: Endüstriyel kontrol kullanımı.....	12
Şekil 1.8	: WSN savaş alanı Kullanımı .....	13
Şekil 1.9	: WSN Ormanda yangın izleme.....	14
Şekil 2.1	: Tek ve Çok sekmeli Kümeleme .....	15
Şekil 2.2	: Küme içi ve kümeler arası bağlantı.....	19
Şekil 2.3	: Yüksek Bağlanabilirlik Küme Algoritması ile kümeleme .....	22
Şekil 2.4	: LID yapısı örneği .....	23
Şekil 2.5	: Min-Max D-algoritmasının yapısı .....	25
Şekil 2.6	: LEACH protokolü .....	28
Şekil 2.7	: TL-LEACH.....	29
Şekil 2.8	: PEGASIS .....	32
Şekil 2.9	: DWEHC .....	34
Şekil 3.1	: Flooding işleyişi .....	40
Şekil 3.2	: Implosion.....	41
Şekil 3.3	: Overlap Problemi.....	41
Şekil 3.4	: Directed diffusion.....	42
Şekil 3.5	: SPIN algoritması.....	43
Şekil 3.6	: Rumor Routing'de ajan kullanımı.....	45
Şekil 3.7	: TEEN.....	49
Şekil 3.8	: PEGASİS.....	51
Şekil 3.9	: GAF algoritmanın çalışma durumları .....	54
Şekil 3.10	: GAF Protokolde sanal ızgara yapısı.....	55
Şekil 3.11	: GEAR Yönlendirme.....	56
Şekil 3.12	:Yayın-aktarım MECN .....	58
Şekil 3.13	: MECN algılayıcı i kapsamı.....	59
Şekil 3.14	: SAR .....	61
Şekil 3.15	: ARRIVE algoritması.....	65
Şekil 4.1	: LEACH protokolü ve TDMA programları .....	70
Şekil 4.2	: LEACH için kurulum faz grafiği.....	72
Şekil 4.3	: LEACH için sabit durum işleminin akış grafiği.....	73
Şekil 4.4	: LEACH protokolünü takma-kod operaysonu açıklar .....	74
Şekil 4.5	: Directed Diffusion protokolü .....	78
Şekil 4.6	: Directed Diffusion protokolünü yöneten sözde-kod.....	79
Şekil 4.7	: Negative reinforcement.....	79
Şekil 4.8	: Flooding ve Gossiping her ikisini yönlendiren mekanizma.....	82
Şekil 4.9	: SGDF Protokolü .....	84
Şekil 4.10	: LGossiping veri yönlendirme şematik şeması.....	84
Şekil 4.11	: ELGossiping yönlendirmesi.....	85
Şekil 4.12	: EESR'da (x,y) 2-boyutlu koordinatlara göre düğüm konumları .....	86
Şekil 4.13	: EESR düzen prosedürü .....	88

<b>Şekil 4.14</b>	: EESR probleminin ilk çözümü .....	90
<b>Şekil 4.15</b>	: EESR probleminin ikinci çözümü .....	90
<b>Şekil 4.16</b>	: EESR gelişimi yönlendirme yöntemi sunan algoritma.....	91
<b>Şekil 5.1</b>	: Ağ Başlatma Fazı.....	98
<b>Şekil 5.2</b>	: Ağ Başlatma Fazı algoritması .....	98
<b>Şekil 5.3</b>	: FELGossiping’te yönlendirme fazı.....	100
<b>Şekil 5.4</b>	: Bilgi toplama ve yönlendirme fazının akış şeması.....	102
<b>Şekil 5.5</b>	: FELGossiping protokolün sözde kodu.....	103
<b>Şekil 5.6</b>	: Radyo modeli.....	104
<b>Şekil 5.7</b>	: 100m ×100m alanında 100 düğüm dağıtımını.....	105
<b>Şekil 5.8</b>	: Simülasyon akışı blok diyagramda açıklanmıştır .....	108
<b>Şekil 5.9</b>	: Toplam enerji tasarrufu .....	109
<b>Şekil 5.10</b>	: Uçtan uca gecikme ortalaması .....	110
<b>Şekil 5.11</b>	: Atlama sayısı ortalaması.....	110
<b>Şekil 5.12</b>	: Kayıp Paket sayısı.....	111

## TABLO LİSTESİ

<b>Tablo 1.1</b>	: Algılayıcı türleri .....	5
<b>Tablo 4.1</b>	: İgi açıklama kullanarak öznitelik ve değer çiftleri (sinekkuşu ile ilgili) .....	77
<b>Tablo 4.2</b>	: Çeyrek daire isimleri, Sektör kimliği ID, Yönetici düğüm isimleri Şekil 4.12'e göre) .....	57
<b>Tablo 4.3</b>	: Düz ve Hiyerarşik protokolleri arasındaki Genel Karşılaştırma .....	93
<b>Tablo 4.4</b>	: LEACH, Directed Diffusion, EESR va Gossiping Protokolleri arasında karşılaştırma.....	94
<b>Tablo 5.1</b>	: Simülasyon parametreleri .....	106

## ÖZET

### **KABLOSUZ ALGILAYICI AĞLARINDA KÜMELEME VE YÖNLENDİRME ALGORİTMA ANALİZİ**

Kablosuz algılayıcı ağları (WSN) bilim dünyasında gelişmekte olan ve hızla büyüyen alanlarından biridir. Düşük maliyet, düşük güç ve çok işlevli algılayıcı düğümler geliştirmeye sebep oldu. Kablosuz algılayıcı ağlardaki son gelişmeler, birçok yönlendirme protokollerin ortaya çıkmasına yol açmıştır. Algılayıcı düğümlerin sınırlı pil kapasitesi sebebinden dolayı, kablosuz algılayıcı ağlarında enerji verimliliği büyük ve önemli sorunlar yaşanmasına neden olur. Böylece, ağ ömrünü en üst düzeye çıkarmak için kablosuz algılayıcı ağların yönlendirme protokollerinde enerji verimli olmalıdır.

Bu tez çalışmasında, kablosuz algılayıcı ağların genel bilgileri, algılayıcı düğümlerin donanım mimarisi, kablosuz algılayıcı ağ uygulamaları, kablosuz algılayıcı ağlar ve Ad-hoc ağ arasında karşılaştırma, kümeleme ve yönlendirme protokolleri, bazı enerji verimli yönlendirme protokolleri arasında derin ve entegre bir karşılaştırmadan söz edildikten sonra yeni bir yönlendirme protokolü öne sürülmüştür.

Önerilen protokol konum-tabanlı yönlendirme Gossiping(LGossiping) protokolü'nün gelişimidir. Yeni protokol adil enerji-verimli konum-tabanlı Gossiping protokolü (FELGossiping) olarak adlandırılır. Simülasyon sonuçları FELGossiping protokolünün uçtan-uca gecikme, yüksek paket kaybı ve genel ağda yüksek enerji tüketimi gibi benzeri yönlendirme protokollerinin (Gossiping, LGossiping and ELgossiping) bazı dezavantajlarını düzeltmiş olduğunu gösterir.

## **SUMMARY**

### **CLUSTERING AND ROUTING ALGORITHM ANALYSIS FOR WIRELESS SENSOR NETWORKS**

The area of wireless sensor networks (WSNs) is one of the developing and fast expanding fields in the scholarly world. This has brought about developing low-power, low cost and multi-function sensor nodes. Recent advances in wireless sensor networks have led to an emergence of many routing protocols. Limited battery capacity of sensor nodes makes energy efficiency a major and challenge problem in wireless sensor networks. Thus, the routing protocols for wireless sensor networks must be energy efficient in order to maximize the network lifetime.

In this thesis, after mentioning to the general information of WSN, hardware architecture of sensor nodes, applications of WSNs, comparison between WSN and Ad-hoc network, clustering and routing protocols, and making a deep and integrated comparison between some energy efficient routing protocols, a new routing protocol has been proposed.

The proposed protocol is an enhancement of Location-based Gossiping (LGossiping) routing protocol. The new protocol has been called **Fair Efficient Location-based Gossiping** protocol (FELGossiping). The simulation results shows that FELGossiping has addressed some drawbacks of counterpart routing protocols (Gossiping, LGossiping and ELgossiping), like the end to end delay, high packet lost and high energy consumption of the network overall.

## 1.GİRİŞ

Kablosuz algılayıcı ağ (KAA) çok sayıda küçük bağımsız algılayıcı düğümler veya parçacıklardan oluşmuş bir bilgisayar ağıdır. Bu küçük bağımsız algılayıcı düğümler ve parçacıklar bilgisayar teknolojisinin geniş bir ağını birleştirirler; donanım, yazılım, ağ iletişimi ve programa metodolojilerini . Bir kablosuz algılayıcı ağ, farklı konumlardaki sıcaklık, ses, titreşim, basınç, hareket veya çevre kirliliğine yol açan maddeler gibi fiziksel veya çevresel durumları işbirliği yaparak izlemek için algılayıcılar olarak kullanılan uzamsal dağılmış otonom araçlardan oluşur [1].

Kablosuz algılayıcı ağın geliştirilmesi başlangıçta muharebe meydanı gözetlemek gibi askeri uygulamalar için harekete geçirildi. Fakat günümüzde kablosuz algılayıcı ağları sağlık hizmetleri uygulamalarını, ev otomasyonlarını, trafik kontrollerini, çevre ve doğal yaşam alanlarını izlemeyi içeren pek çok sivil uygulama alanında kullanılmaktadır.

### 1.1 TANIMLAR

**Algılama:** Algılama, olayların oluşumunu içeren fiziksel bir obje veya süreç hakkında bilgi toplamak için kullanılan bir yöntemdir.

**Algılayıcılar:** Algılayıcılar algıladığı bir görevi yerine getiren nesnelere. Örneğin, insan vücudu çevreden görsel bilgileri (gözler), sesler gibi işitsel bilgileri (kulaklar) ve kokuları (burun) yakalayabilen algılayıcılarla donatılmıştır.

**Algılayıcı düğüm (Sensor network elemanı):** Algılayıcı düğüm, kablosuz algılayıcı ağ için algılama, iletişim, veri depolama, veri sürecinden sorumlu düşük maliyetli ve düşük güçlü bir araçtır. Her bir algılayıcı düğüm, algılayıcılardan, mikro işlemcilerden ve alıcı/vericilerden oluşur.

**Kümeler:** Bir algılayıcı ağ elemanları kümesi, tek bir küme baş elemanı ve çok sayıda küme elemanlarından oluşan kablosuz algılayıcı ağ için birimler haline getirilir.

**Kümenin baş elemanı:** Kümenin baş elemanı küme ve kümenin elemanlarını kontrol eden bir düğümdür. Kümenin baş elemanları üyelerinin düğümlerinin verilerini ve alıcı düğüme ve diğer düğümlere gönderilen verileri birleştirir. Kümenin baş elemanı, kod bölme çoklu erişim, zaman bölümlü çoklu erişim ve diğer zaman çizelgeleri gibi iletişim çizelgelerini organize eder, oluşturur ve temellendirir.

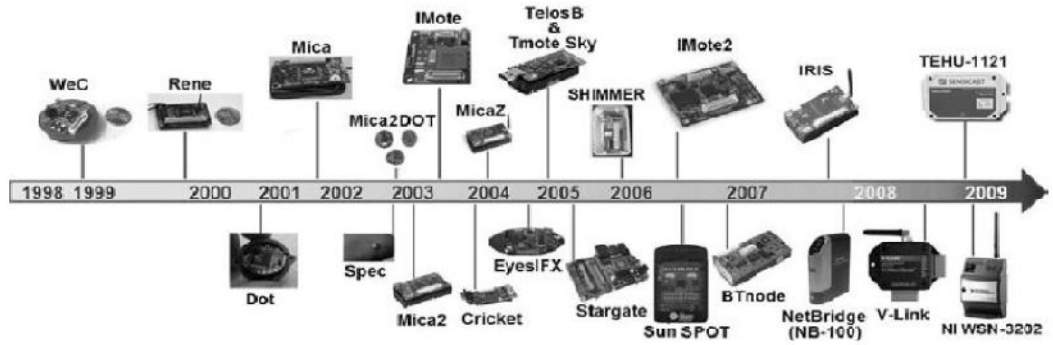
**Baz istasyonu:** Baz istasyonu verilerin bir araya geldiği ve kesildiği yer olan sıraduzensel kablosuz algılayıcı ağ içinde daha üst bir seviyededir. Bu düğüm algılayıcı düğüm ve son kullanıcı arasında transfer bağlantısı sağlıyor. Aynı zamanda, kullanıcı ile ya doğrudan ya da mevcut kablolu ağ üzerinden olan iletişimden sorumludur.

**Son kullanıcı:** Alıcı noktası veya baz istasyonu sadece algılanan veriyi son kullanıcıya gönderecektir. Son kullanıcı gönderilen verileri işleyecek ve sonuca ulaşabilecektir.

## 1.2. ALGILAYICI DÜĞÜM MİMARİSİ

### 1.2.1 Platformlar

Telos, Cricket, MicaZ, Mica2, İris, Sun SPOT, İmote2 vb.[2] dahil olmak üzere son yıllarda pek çok düğüm çeşidi geliştirilmiştir.



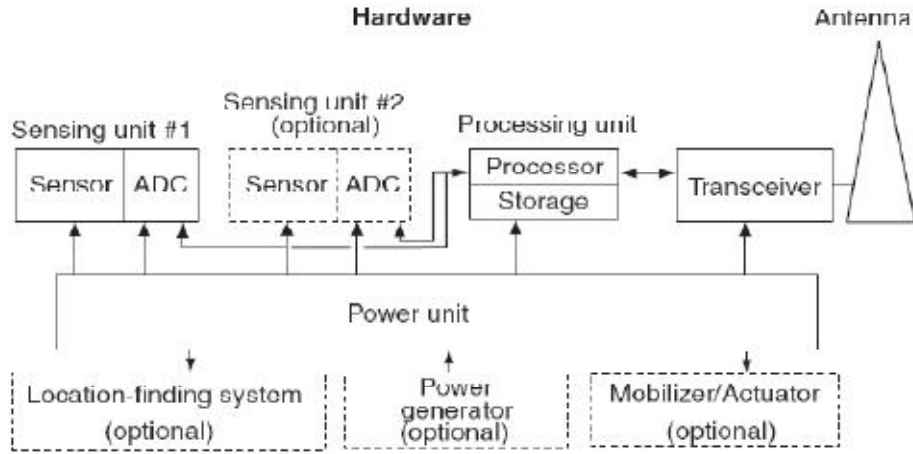
Şekil 1,1: Algılayıcı düğüm platformu[2].

Bu parçacıkların boyutları bir kibrit kutusu büyüklüğünden bir tükenmez kalem ucu boyutuna kadar değişebilir. Mikro Elektromekanik Sistemler (MEMS) teknolojisi parçacığı bir santimetre kübü geçmeyecek bir hacme sığabilen tek bir çipe bir parçacığı yerleştirebilmek amacıyla küçültmek için kullanılmaktadır. Bu parçacıklara “Akıllı Toz” adı verilmiştir.

### 1.2.2. Algılayıcı Düğüm Donanımı

Kablosuz algılayıcı düğümü kablosuz algılayıcı ağların temel elemanıdır. Algılayıcı düğüm, iletişim protokollerini ve veri işleme algoritmalarını depolar ve uygular [3].

Genellikle, algılayıcı düğüm mimarisi Şekil 1,2’de görüldüğü üzere dört bileşenden oluşur: Algılama, İşleme, İletişim, Güç birimi.



Şekil 1.2 : Algılayıcı düğüm mimarisi [4]

Yukarıdaki birimlere ilaveten, bir kablosuz algılayıcı düğüm pek çok uygulamaya özel bileşenler içerebilir. Örneğin; yer belirleme sistemi gibi pek çok ticari algılayıcı düğüm ürünleri genişletme yuvası içerir ve seri iletişimi destekler.

#### 1.2.2.1. Algılayıcı Birim

Entegre halinde bir veya daha fazla algılayıcı düğüm ve bir analog-dijital dönüştürücü. Algılayıcılar çevreyi algılar, verileri toplar ve bu verileri işlenmeleri için göndermeden önce akım ya da voltaj gibi temel verilere dönüştürür. Çevreden algılanan analog verileri dijital verilere dönüştürür ve işlenmesi için mikro denetleyiciye gönderir [3]. İstenen işlemin doğasına bağlı olarak hali hazırda bulunan farklı kategorilerde algılayıcılar vardır. Tipik bir kablosuz algılayıcı düğüm 0.5 Ah ve 1.2 V tan az bir güç kaynağına sahip bir mikro-elektronik düğümdür. Algılayıcıların boyutları ve enerji tüketimleri algılayıcıların seçiminde göz önünde bulundurulmuş anahtar faktörlerdir [4,5, 6].

#### 1.2.2.2. İşlem Birimi

Bu birimin amacı algılama ve iletişimle uyumlu komutların uygulanmasıdır. İşlemci çipi, mikrodenetleyici birim, program komutlarını depolamak için iç hafıza, algılanan verileri depolamak için aktif hafıza ve işlemci - hafıza arasında veri alışverişi için tek yoldan meydana gelir [3].

Bu birim veri toplama, giren ve çıkan bilgileri işleme, iletim performans şartlarını göz önünde bulundurarak yönlendirme bilgilerini ayarlamaktan sorumludur [4, 5, 6]. İşlem birimi yerleşik hafızadan oluşuyor ya da gömülü kart içine entegre edilmiş küçük bir



depolama birimi ile ilişkili olabilir. İşlem birimi algılayıcı düğümün algılama işlemlerini yapmasını sağlayan prosedürleri yönetir, bununla ilişkili algoritmaları çalıştırır ve diğer düğümler ile kablosuz iletişim vasıtasıyla işbirliği kurar.

İşlem birimi Von Neumann, Harvard, and Süper-Harvard (SHARC) olmak üzere üç tip mimariden oluşur[3].

**Von Neumann:** Bu mimarı veri ve program komutlarını depolamak için tek hafızadan oluşur. Ayrıca, hafıza ve işlemci arasında veri alışverişi için tek yola sahiptir. Bu mimarı, her veri transferinin farklı bir saate sahip olmasından dolayı veri işlemede düşük hızla sahiptir.

**Harvard:** Harvard Von Neumann mimarisinin geliştirilmiş halidir. Bu mimarı hafızayı veri hafızası ve komut hafızası olarak ikiye ayırır. Her hafıza bir işlemci ile ayrı bir tek yol vasıtasıyla arayüzlenmiştir. Program ve veri aynı anda depolanır.

**Süper-Harvard:** Bu mimarı aralarında en iyisidir. Bu mimaride iki önemli bileşen eklenmiştir; komut önbelleği ve I/O bileşeni. Komut önbelleği işlemciye entegredir ve işlemcinin performansını artırır. Geçici olarak komutları depolayabilir böylece, program hafızasından getirilen kalıntı komutları azaltır.

#### 1.2.2.3 İletişim Birimi (Alıcı -Verici)

İletişim birimi mikro denetleyiciden gelen bitleri radyo frekansı (RF) dalgaları halinde iletilmesi ve diğer uçtan alınması amacıyla dönüştürülmesi için gerekli işlemleri yapar. İletişim birimi büyük bir radyo dalgası yayabilen ISM (endüstriyel, bilimsel ve tıbbi radyo bandı) bandını kullanır. Kablosuz iletişim ortamında radyo frekansı optik iletişime göre (lazer, İR) daha fazla tercih edilir. Lazer daha az enerji kullanmasına rağmen iletişim için bir görüş alanı gerektirir ve yayın kapasitesi sınırlıdır. Kablosuz algılayıcı ağ 433MHz ile 2.4 GHz arasındaki frekansları kullanır.

#### 1.2.2.4 Güç birimi

Kablosuz algılayıcı düğüm, güç ünitesi tarafından beslenir ve bu birimin sınırlı kapasitesi her birimin enerji açısından verimli olarak çalışmasını gerektirir. Düğümde gereken pil gücü ve enerjiyi çevreden alan güneş pilleri gibi diğer enerji kaynaklarından oluşan enerji ihtiyacını kontrol eder. Düğümdeki temel güç tüketimi hesaplama ve enerji açısından çok pahalı olan iletim esnasında olur. Genellikle, algılayıcı düğümler pille çalışır fakat güneş pilleri vasıtasıyla çevreden de enerji toplayabilirler [4, 5, 6].

Algılayıcıların fiziksel faktörleri pek çok uygulamada kullanıldı[3].Bu faktörlerden bazılarının örnekleri aşağıdaki Tablo 1.1’de gösterilmiştir;

Tablo 1.1: Algılayıcı türleri

<i>Tür</i>	<i>Örnekler</i>
Optik	Foto diyotlar, fototransistorlar, kızılötesi algılayıcılar, CCD (Işığa Hassas Bir Elektronik Levha veya Kamera Çeşidi) Algılayıcılar
Basınç Nem	Basınç, ölçü, barometre, iyonlu basıölçerler Kapasitif ve dirençli algılayıcılar, nem ölçerler, MEMS (Mikro-Elektro Mekanik Sistemler) –neme dayali
Radyasyon Mekanik	İyonizasyon dedektörleri, Geiger-Mueller sayaçları Gerilim ölçerler, dokunsal algılayıcılar, kapasitif diyagramlar, basınç dirençli hücreler
Radyasyon Elektromanyetik Akustik Sıcaklık Hareket, titreşim	İyonizasyon dedektörleri, Geiger-Mueller sayaçları Hall etkisi algılayıcıları, manyetometreler Piezoelektrik rezonatörler, mikrofonlar Termistörler, termo elemanlar İvmeölçerler, ciroskoplar, fotosensörler

### 1.2.3. Yazılım

Donanım altyapılarına ve standartlara ek olarak, çeşitli yazılım altyapıları da kablosuz algılayıcı ağlar için özel olarak geliştirilmiştir. Bunlar arasında en çok kabul gören altyapı, kablosuz gömülü algılayıcı ağlar için tasarlanmış açık kaynaklı bir işletim sistemi olan TinyOS’dur [7]. TinyOS kod boyutlarını minimize eden ve yeni iletişim protokollerinin hayata geçirilmesi için esnek bir altyapı sağlayan bileşen tabanlı bir yapı içermektedir.

*LiteOS* [8], Unix benzeri soyutlamalar sağlayan çoklu kullanımlı işletim sistemidir. TinyOS ile karşılaştırıldığında, *LiteOS* [8] çok kanallı işletim, dinamik bellek yönetimi ve komut satırı arayüzü desteği sağlar.

*Contiki* [9] Telos, Tmote ve Mika ailelerinde kullanılan TI MSP430 ve Atmel AVR gibi mikro denetleyiciler içeren çeşitli altyapılar üzerinde kullanmak için geliştirilmiş çoklu görevli bir işletim sistemidir ve açık bir kaynaktır.

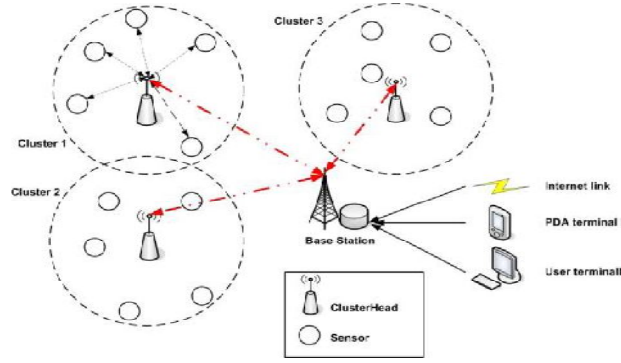
En son (en güncel) *SunSPOT* altyapısı [10] bir işletim sistemi kullanmaz fakat bir Java sanal makinesini (VM); Squawk’i oldukça yetenekli bir Java ME uygulaması olan Bare

Metal ( donanımsal sanallistirma yöntemi ile bir ana bilgisayar üzerinden birbirinden farklı işletim sisteminin eş zamanlı olarak çalıştırmayı sağlayan bir sanallistirma sistemi) üzerinde çalıştırır. Sanal makine (VM) anlık bellek dışında direkt olarak çalışır.

### 1.3 KABLOSUZ ALGILAYICI AĞ MİMARİSİ

Kablosuz Algılayıcı Ağlar (KAA), sayısal ve uygulamalı işlemler yapan, birbirleriyle kablosuz iletişim kuran ve algılama yeteneğine sahip yüzlerce veya binlerce mikro algılayıcı düğümler kümesidir.[11]

Genel kablosuz algılayıcı ağ mimarisini için. Şekil 1.3 bkn .



Şekil 1.3: Genel kablosuz ağ mimarisini[12]

### 1.4. KABLOSUZ ALGILAYICI AĞLARIN (WSN) AVANTAJLARI

Güvenilirlik, doğruluk, esneklik, konuşlandırma kolaylığı ve maliyet verimliliği Kablosuz Algılayıcı Ağların (WSN) temel amaçlarıdır. Kablosuz Algılayıcı Ağların birçok özellikleri ve faydaları vardır, bu nedenle, günümüzde kullandığımız iç mekândan dış mekanlardaki kullanımlarına kadar aşağı yukarı her alanda bu karşılaşılabilir. Bu faydaların birkaçı aşağıda yer almaktadır [13]:

- *Hata toleransı:* Her bir algılayıcının içinde, hata toleransı araç ve bilgi fazlalıkları ile başarılıdır.
- *Çetin koşullarda çalıştırılabilme:* Algılayıcı düğümleri, yüksek seviyedeki hata toleransı sayesinde zorlu ortamlarda konuşlandırılabilirler.

Örneğin, volkanik hareketler ve yangın olaylarının takibi gibi erişimi zor olan yer ve durumlarda kablosuz algılayıcı ağların etkinliği görülür.

- *Kapsama alanı:* Çok sayıdaki algılayıcı düğümler bir fiziksel ortamı izlemek için konuşlandırılabilirler; örneğin tarım uygulamalarında, bu sayılar daha fazla algılayıcı düğüm gerekmesi durumunda ağ maliyetini etkilemeksizin arttırılabilir.
- *Bağlanırlık:* Algılayıcılardan ve alıcılardan oluşan kablosuz algılayıcı ağlar bazı durumlarda bir ağ geçidi oluştururlar. Bu ağ geçidi internet gibi diğer ağları birbirine bağlamak için oluşan bir geçit olabilir. Kablosuz algılayıcı ağların (WSN) kontrolünü kolaylaştırmak için kümeleme kullanılır ve her bir küme belirli bir olaya odaklanır ve bu kümeler (bireysel ağlar) ilişkili olan bilgilerini kolayca paylaşabilir.
- *Algılama duyarlılığı:* Çok sayıda ve farklı çeşitlerdeki algılayıcı düğümler tarafından toplanan bilgiler tek bir algılayıcı veya az sayıdaki algılayıcı düğümünler tarafından toplanan bilgilerden daha doğrudur.
- *Minimal insan etkileşimi:* Yukarıda bahsedildiği gibi bazı uygulamalarda, algılayıcı düğümler erişilmesi zor ve çekin koşullarda kullanılması gerekebilir. Bu da kablosuz algılayıcı ağlarla insanların etkileşimini minimum düzeye indirir.
- *Dinamik algılayıcı zamanlaması:* Uygulamaya göre kablosuz algılayıcı ağlar iletilen veri için bir zamanlama şeması kullanarak öncelik tayin edebilirler.
- *Kablosuz iletişim:* Bazı uygulamalar için hali hazırda var olan kablolu ağları kullanarak algılayıcı ağ inşaa etmek kolay olsa da, birçok uygulama için kablolu algılayıcı ağ büyük bir sorun teşkil eder. Ayrıca, algılayıcı ağların asıl avantajı kablosuz olmasıdır. Normalde algılayıcı düğümler radyo frekansı [RF] kanalı ile kablosuz olarak iletilir. Algılayıcı ağın kablosuz olması için temel nedenler aşağıda sıralanmıştır:

- a) Kablolamanın yüksek maliyetli olması (- \$ 20- 2.000 ayak başına). Bu değere ek olarak, kablolu ağ teknolojisinde kullanılan diğer cihazların maliyeti de yer alanır.
- b) Kablolama kurulum süresinin % 20 - % 80' ini kapsar ve normal iş faaliyetlerinin aksamasına yol açar.
- c) Kablolu ağlarla sadece tek bir yol kullanılır, fazlalık yoktur.
- d) Çok sayıda cihazlar için bakım bir sorun oluşturur.
- e) Ulaşması zor alanlar için kullanılır.
- f) Kablolar düğümlerin hareketliliğini kısıtlar.

- g) Kablolar algılayıcıların denetledikleri olayın yakınında bulunmalarına engel teşkil edebilir. Bu durumda algılayıcılar arasındaki kablosuz iletişim kaçınılmaz bir gerekliliktir.
- h) Kablolardaki karışıklıklar.

### 1.5. TASARSIZ(AD-HOC)AĞLAR VE KABLOSUZ ALGILAYICI AĞ( WSN) PROTOKOLLERİ ARASINDAKİ FARKLAR

Kablosuz özel amaçlı ağ (WANET) acil bir ihtiyacı karşılamak için eş düğümler arasında kurulmuş geçici bir ağdır[14]. Kablosuz ve özel amaçlı ağlar için birçok protokol vardır fakat bu protokoller kablosuz algılayıcı ağların özel gereksinimleri nedeni ile onlara uygun değildir. Akyıldız ve ark göre [5], kablosuz algılayıcı ağlar diğer kablosuz özel amaçlı ağlardan düğüm kimliği, ağ boyutu, düğüm yoğunluğu, düğümün bozulma ihtimali, topoloji değişikliklerinin sıklığı ve düğümlerin kaynak sınırlamaları gibi birçok alanda farklılık gösterir.

Bu farklılıklar aşağıdaki gibi açıklanmıştır:

**Düğüm kimliği:** Binlerce düğümün olduğu bir durumda oluşturulan bir adres şeması çok büyük önem kazanır. Verilerin etkili iletimi için, her düğümün kendine özel bir adresinin olması gereklidir. Bu kadar yüksek düğüm sayılarında global bir adresleme şemasının olması mümkün değildir. Aksine özel amaçlı düğümler, genellikle seçim ve uygulama kriterleri kesin belirlenmiş İP adresleri kullanırlar.

**Düğümlerin Kaynak Sınırlamaları:** Sınırlı enerji, bant genişliği kaynakları, iletim gücü, işleme kapasitesi ve depolama kablosuz algılayıcı ağları yönetirken göz önünde bulundurulması gerek kısıtlayıcı faktörlerdir.

**Kablosuz algılayıcı ağın ağ büyüklüğü;** birkaç düğümden birkaç bin düğüme kadar değişebilir. Öte yandan Kablosuz özel amaçlı ağ (WANET) genellikle yüzden az düğümden oluşur. Maksimum sekiz düğüme kadar oluşabilen Bluetooth Piconet, bir kablosuz özel amaçlı ağ (WANET) örneğidir. Kablosuz yerel alan ağı (WLAN) da kablosuz özel amaçlı ağa (WANET) başka bir örnektir. WLAN Elektrik ve Elektronik Mühendisleri Enstitüsü (İEEE) tarafından geliştirilmiş olan İEEE 802.11b standardına dayanmaktadır. WLAN boyutu erişim noktası başına 32 düğüm ile sınırlıdır[15].

**Düğüm yoğunluğu:** Bir kablosuz algılayıcı ağda düğüm yoğunluğu genellikle yüksektir. WANET'ler genellikle birbirine yakın birkaç düğümden oluşurken, WSN'ler

nispeten ufak bir alandaki yüksek sayıdaki düğümlerden meydana gelir. Bu, düğümlerin büyüklüğü ile ilgilidir. Bir WSN düğümü bir madeni euro kadar küçük olabilirken, diğer WANET'ler genellikle notebook bilgisayarlar, cep telefonları ya da avuç içi bilgisayarlar boyutundadır.

**Topoloji değişikliklerinin sıklığı:** Bir WSN içindeki topoloji değişikliklerinin sıklığı, düğüm hataları, düğüm ekleme, düğüm hareketliliği ve çevre girişimi gibi faktörler nedeniyle yüksektir. Ağ düğümü konumunun ve sayısının bu değişimlere uyum sağlama yeteneğine sahip olması gerekir. Topoloji değişiklikleri birkaç milisaniye kadar sık olabilir. Diğer WANET'lerde düğümler genellikle ağ katılmayı ve birkaç dakika içinde ağdan ayrılmayı talep eder.

**Bozulma olasılığı:** Bir WSN orman veya bir afet bölgesi gibi, uzak veya erişilemeyen alanda konuşlanmış olabilir. Bu gibi şartlarda düğümlerin bozulma olasılığı yüksektir. Bazı düğümler enerji kaynaklarını diğer yollardan daha çok kullanılan bir yönlendirme yolu üzerinde olduğundan diğer düğümlere göre daha çabuk tüketir. Diğer WANET'lerin içindeki düğümlerin şarj edilebilir enerji kaynakları vardır ve işlevine mümkün olmayan ölçüde onlara zarar verebilecek olumsuz çevresel koşullara maruz değildir.

## 1.6. WSN PROTOKOL YIĞINI

WSN protokol yığını için yaygın olarak kabul gören model, IT ve telekomünikasyonda kullanılan yığını Açık Sistemler Bağlantı (OSI) referans protokol yığından uyarlanmıştır. WSN protokol yığını uygulama ve yönlendirme katmanı, güç bilinçliliği, yönlendirme kapasitesi ve düğümler arasında verimli bir kablosuz iletim arasında entegrasyon sağlar. Protokol yığınının içindeki katmanlar Şekil 2,1 de gösterilmektedir [16].

Her katmanın görevi kısaca şu şekilde açıklanabilir:

1. Fiziksel katman güçlü sinyal modülasyonu ve iletim teknikleri uygulayarak verilerin başarılı bir şekilde temin edilmesini ve kablosuz iletişim radyo arayüzü sağlar (mesaj alma/ gönderme).
2. Veri bağlantı katmanı üzerinde uygulanan protokol tipik bir MAC protokolü olarak bilinir. MAC protokollerinin genelde gücü farkında olmaları, gürültüyü filtrelemeleri ve komşu düğüm iletimleri ile mesaj çarpışmasını minimize etmeleri beklenir.

3. Ağ katmanı yönlendirme protokolünü uygular ve bu aynı zamanda bu tezin araştırma alanıdır.
4. Taşıma ve uygulama katmanları genellikle tek bir katman oluşturmak için birleşirler. Bu katman veri akışını sağlar ve algılayıcı gereksinimlerine bağlı olarak uygulama yazılımı için barındırmadan sorumludur.
5. Devam eden akademik araştırma, protokol yığınındaki katmanlar arasındaki sınırları yıkmayı ile ilgilidir. Bu araştırma günümüzde kabul görmüş katmanlar veya bu katmanlardan birkaçının birleşiminden her hangi bir katman arasındaki sınırlara bir köprü için bir “Karma katmanlı” yaklaşımı üretecektir ancak bu tezin odağı bu olmadığı için daha fazla detaylandırılmayacaktır. İlgilenen okuyucular daha fazla bilgi için [17].



Şekil 1.4: WSN Protokol yığını

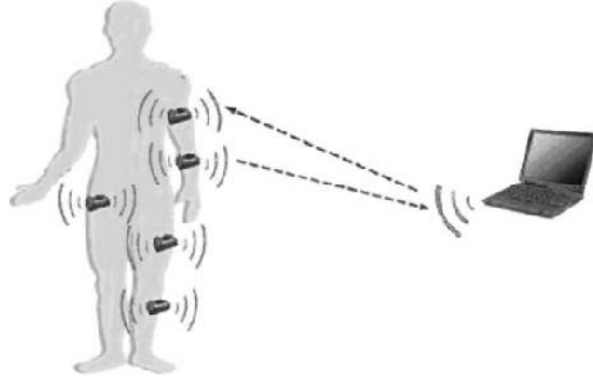
## 1.7. KABLOSUZ ALGILAYICI AĞLARININ UYGULAMALARI

Ucuz algılayıcılara ve kablosuz iletişime olan kolay ulaşılabilirlik, sivil ve askeri alanlarda geniş çaplı uygulamaların ortaya çıkmasına neden oldu. Biz, bu tezde sivil ve askeri alanlardaki ağ uygulamalarının birkaç örneğini tanıtacağız.

### 1.7.1. Sağlık hizmetleri

Kablosuz algılayıcı ağının, sağlık hizmetlerinde birçok kullanım alanı vardır ancak en önemlisi hastanın sağlık durumunun 24 saat izlenebilmesidir. WSN (Kablosuz sensor Ağları), aynı zamanda hastanın sağlık durumunu (nabız, kan basıncı, kalp, sinir v.b.) kontrol etmek ya da doktorların değişimleri takip edebilmesini sağlamak için hasta vücuduna da bağlanabilir[18]. WSN, hastanelerdeki ilaçların tanınmasında da kullanılır. Bu uygulamaların kullanımı, hastanın rahatını sağlamada gerekli masrafların

düşürülmesine de yardımcı olur. Felçli hastaların rehabilitasyonu için kullanılan kablosuz hareket algılayıcı uygulamalarının örnekleri, şekil 1.5'te yer almaktadır.



Şekil 1.5: WSN sağlık uygulaması[18].

### 1.7.2. Tarımsal Uygulamalar

WSN, arazilerde verimli toprak, nem, güneş ısı, su derinliği, toprak-su dengesi ve bitkilerde hava geçirim yüzdesi gibi değerleri bulmanın yanında bitki gelişiminin normal, sağlıklı ve hızlı gerçekleşmesi gibi amaçlar için de kullanılabilir. Bu sonuca ulaştıktan sonra, gezegenin gelişim hızını hesaplayabilir ve tarımsal üretimi arttırabiliriz. WSN, aynı zamanda yönetim uygulamaları için de kullanılır, bu uygulamaların örnekleri COMMONSense[19] uygun bir su idare sistemi sağlama amacını gerçekleştirir. Buna örnek olarak şekil 1,6'da gösterilmektedir.

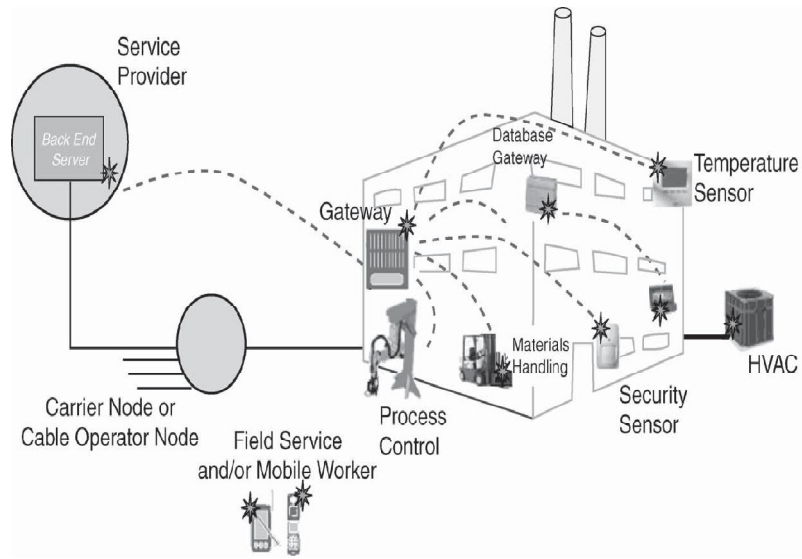


Şekil 1.6 WSN tarımsal kullanımı[18].



### 1.7.3. Endüstriyel Uygulamalar

WSN, çalışma sağlama gibi birçok endüstriyel uygulama için kullanılır, üretimdeki algılayıcı kullanımı ve ürün kontrolü ile ürün çalışma denetimini birleştirme bu uygulamalara örnektir. WSN aynı zamanda kimyasal madde ya da petrol üretiminde de boru hattının hareketlerini denetlemede de kullanılır. Makinelerin insanların ulaşamayacağı bölgelerine yerleştirilen algılayıcılar, makinenin çalışmasını denetler ve herhangi bir bozulma durumunda uyarı verir. İndustrial kullanım örneği aşağıdaki şekil 1.7’de gösterilmiştir.



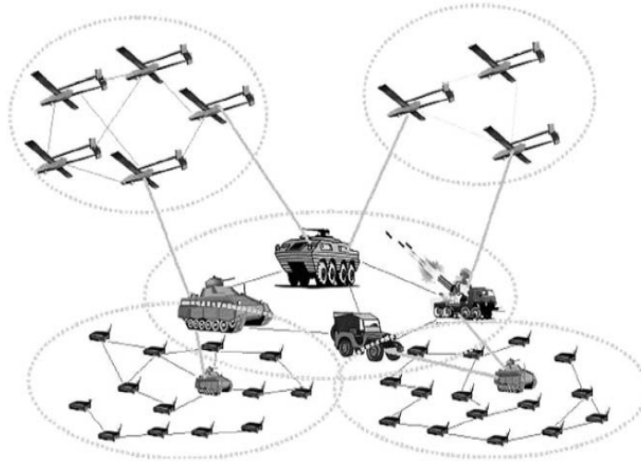
Şekil 1.7 : Endüstriyel kontrol kullanımı[22].

### 1.7.4. Askeri Uygulamalar

WSN, komuta, kontrol, savunma, iletişim, programlama, istihbarat, keşif ve savaş alanında hedef belirleme gibi birçok askeri uygulamada da kullanılır. Savaş alanındaki kablosuz algılayıcı aygıtları, çeşitli hedef nesnelere gelen akustik ve manyetik sinyalleri algılayabilir, bu sistem aynı zamanda kendi kendini otomatik örgütlenme ve ayarlama yetisine de sahiptir[20], savaş alanı kullanımı şekil 1.8’e bkz. Diğer bir özelliği de muhtemel hedefleri tespit ve takip etmek, anlık düşman hareketlerini komuta merkezine bildirmektir.

WSN, çevreden gelen ve diğer ucuz makinelerden çıkan algılama gürültülerini de filtrelemede kullanılabilir. Savaş alanlarındaki bir başka kullanımı da düşman saflarında bulunan keskin nişancıların yerini tespitidir. İstihbarat açısından bakıldığında bu

algılayıcılar; araçların, tankların, savaş uçaklarının, deniz altılarının, misil ve torpidoların iç kısımlarına önlerindeki engellerden (sakınabilmeleri için) yerleştirilir. Kontrol Algılayıcıları, biyolojik, nükleer ve kimyasal silahlarla yapılabilecek muhtemel terörist saldırılarının ortaya çıkarılması ve keşif araştırmaları için konuşlandırılabilir. Bu uygulamanın bir örneği olarak VigilNet'i gösterebiliriz, VigilNet[21], araçların ve manyetik nesnelerin hareketlerinden ortaya çıkan manyetik alanı algılayan manyetik algılayıcılarla donatılı XSM, Micra2 ve Micra2dot platform devrelerine dayalı bir sistemin tasarımı ve uygulamaya geçirilmiş halidir.

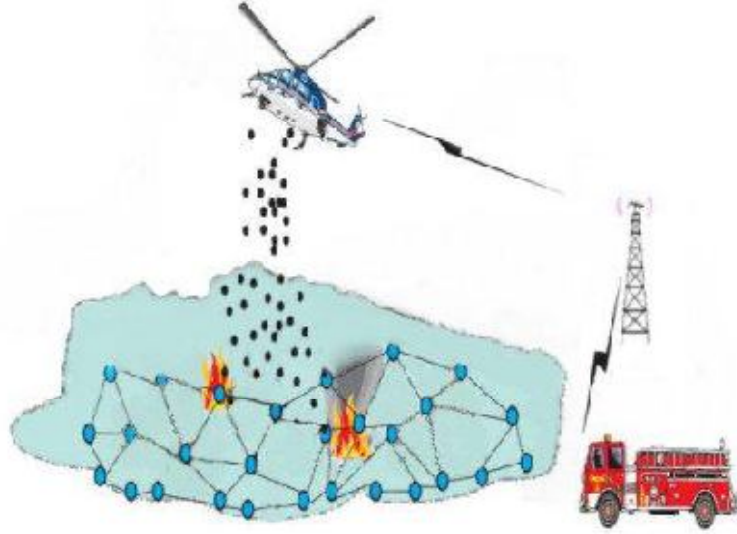


Şekil 1.8: WSN savaş alanı Kullanımı.

### 1.7.5. Çevresel İzleme

Bu uygulama uzun yıllar önce ortaya çıkmıştı. Çevresel denetleme algılayıcıları, habitat izlemesi, afet yardımı ve kaza gözlemleri gibi birçok çevresel denetleme parametresini denetlemek için kullanılır. Habitat denetleme algılayıcıları, orman ve hayvanat bahçelerindeki hayvanları izleyerek onların türlerini, yaşayış biçimlerini öğrenmek ve aynı zamanda onları muhtemel risklerden korumak için kullanılır. Afet yardımı denetleme algılayıcıları ise çoğunlukla, yangın ya da sel yaşanma riski olan bölgelere bu ihtimalleri azaltmak için konuşlandırılır; depremle ilgili algılayıcılar da depremin yönü ve boyutunu öğrenmek, ayrıca binanın güvenlik seviyesini tespit etmek için binaların içine yerleştirilir. Trafik kazalarındaki algılayıcılar, araç sürücüleri ve yayalar için uygun trafik akışını sağlamak için kullanılır; kablosuz algılayıcı ağları, caddelere ve yol kenarlarına trafikle ilgili bilgi toplamak amaçlı yerleştirilen cihazlardan oluşan bu

uygulamalardaki ana rolün sahibidir. Bu algılayıcılar, yollardaki tehlikeleri önlemek amacıyla araçların iç kısımlarına da yerleştirilebilir. Şekil 1.9'a bkz.



Şekil 1.9 : WSN Ormanda yangın izleme.

#### 1.7.6. Kamusal Kullanımlar

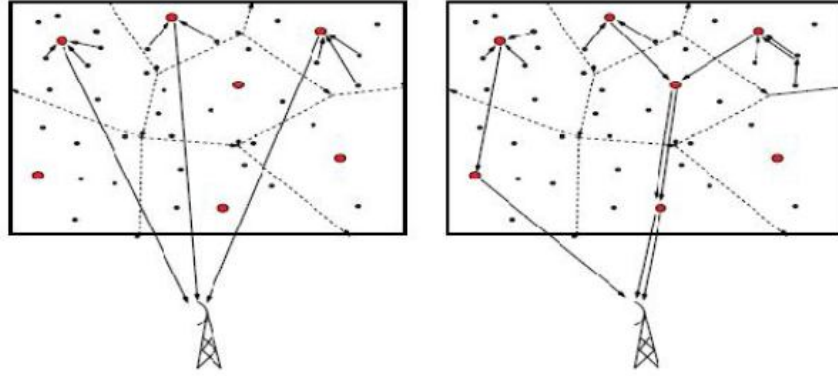
WSN, algılayıcıların güvenliği denetleyebildiği ve kişisel kullanıcıya bağlı otomatikleştirilmiş hizmetler sağlayabildiği ortam denetimli evler gibi diğer birçok alanda kullanılabilir. Bu uygulamanın bir örneği Smart dust'tur. Smart dust, bir binanın etrafına ısı ya da nem gibi değerleri ölçmek, aynı zamanda da rahatsızlıkları ve engelli kişileri tespit etmek gibi amaçlarla kullanılmak üzere binanın etrafına dağıtılmış yüzlerce küçük algılayıcı parçalardan oluşan bir sistemdir.

## 2. KÜMELEME

Kümeleme; ölçeklenebilirliği, enerji tasarrufunu ve dayanıklılığı arttırabilmek amacıyla algılayıcı düğümlerin gruplar halinde bir araya getirilmesidir. Bir grup düğüm, küme başlarının (CHs) kontrol ettiği kümelere yerleştirilir. Kümeler, küme başı ve onun düğüm üyeleri arasındaki bağlantıyla olduğu gibi kümeler ve baz istasyon arasındaki bağlantı ile de ilişki içerisinde.

*Tek-sekmeli: kümelemede* bütün algılayıcı düğümleri üzerlerindeki bilgiyi doğrudan baz istasyona gönderirler, bu uygulama açısından oldukça basittir ancak çok daha geniş boyutlu kablosuz iletim gerektirir. Bu da hem çok daha fazla enerji harcanması, hem de donanım kurulumu anlamına gelir.

*Çok-sekmeli: kümeleme mimarisi* Algılayıcı düğümleri, çok daha az çaplı, bir ya da iki düğümden oluşan kablosuz iletim kullanarak bilgilerini Baz İstasyona gönderir. Tek ve çok sekmeli kümeleme Şekil 2.1’de gösterilmektedir.



Şekil 2.1: Tek ve Çok sekmeli Kümeleme

**Kümeleme kullanımının aşağıda da gösterildiği gibi birçok avantajı vardır[23]:**

- Kümelemenin ağı ölçekleyebilme yeteneği vardır.
- Başlangıçtaki yönlendirme ağını hazırlayabilir ve böylece yönlendirme tablosunun boyutunu tek bir düğümden toplatarak azaltabilir.
- Bir bant genişliği bağlantısını devam ettirebilir çünkü kümeleme, kümeler arası küme başlarının amaçlarını belirler.
- Algılayıcı düğümler arasındaki mesaj alışverişindeki aşırılığı önler.

- Algılayıcı üslerindeki ağ durağanlığını sağlar ve ağın ek yükünü küçültülmesine yardımcı olur.

*Ölçeklenebilirlik:* Fiziksel bir olayı yakalamak için yerleştirilmiş algılayıcı düğümlerin sayısı yüzlerce, binlerce olabilir. Bu nedenle, bu ağlar için geliştirilmiş ağ kurma protokolleri, büyük sayılardaki düğümleri etkili bir şekilde düzenleyebilmelidir.

## 2.1 KÜMELEME TEKNİKLERİNİN KATEGORİSİ

Kümeleme teknikleri, onların WSN'ler üzerindeki özellikleri ve kısa yollarını belirten yayınlanmış projeleri düzenlemek için oluşturulmuştur[23]. Bu kategori oldukça kullanışlıdır ve WSN'in kullanılabilirliğini inceleyebilir. Bu kategorinin bazı bölümleri şunlardır:

### 2.1.1 Ağ Modeli

WSN için olan uygulamalara yardım edecek farklı mimariler ve amaçlar vardır. Biz de bu mimarilerden bazılarını değineceğiz.

- *Ağ Dinamikleri:* Ağ üzerindeki çok daha fazla yapı, algılayıcı düğümlerinin kurulu olduğu bir altyapıda kullanılır. Bunun yanında, bu ağların bazıları, veri istasyonlarının ya da küme başlarının değişkenliğini destekleme amaçlı kullanılır. Düğüm değişkenliği kolay olmayacaktır çünkü diğer Düğüm öğeleri dinamik olarak değişecek ve kümelemenin de zamanında değişmesini sağlayacaktır. Öte yandan, bu değişkenlik, WSN için hazırlanan uygulamalara bağlı olarak, **aralıklarla** ya da **sürekli olarak** algılayıcılar tarafından denetlenebilecektir. Aralıklı denetlemeler, ağın duyarlı bir biçimde çalışmasına olanak verir. Böylece rapor alışverişi sırasında oluşabilecek sinyal trafiğinin akışını sağlar. Buna örnek olarak ormanların yangın ihtimaline karşı önceden izlenmesi verilebilir. Sürekli denetlemelere gelince, bunlar dönemsel raporlama gerektirir. Bu, Alıcıya iletilmek üzere çok daha fazla yoğunluktaki bir trafiğe yol açar. Bu denetlemeler, ağın durağan bir hâl almasını da sağlar. Küme başı, algılayıcı düğümlerinden rastgele seçilmişse, küme başının işlevinde bir rotasyon gerektirebilir.

Bu tarz uygulamaların örnekleri hedefin hareketlerini gösterir. Denetlemelerde çok, büyük oranda dalgalanma yaşanırsa, aralıklı denetleme bu model için daha

uygun olabilir.

- *Şebeke içi bilgi işlem: Farklı* düğümlerden gelen veri içindeki paketler biriktirilebilir ya da sıkıştırılabilir, bu da aktarım sayısında bir azalmaya yol açar. Veri toplama işleminde, farklı algılayıcılardan gelen veriyi biriktirmek için (averaj, Max, Min, kopyasını engelleme) gibi bazı fonksiyonlar kullanılır. Veri hesaplama, iletişim ağından daha az enerji kullanır, bu sayede veri toplamada enerji tasarrufuna gidilebilir. Bu sistem, sinyal işleminde de uygulanabilir, bir algılayıcı düğümün sinyal gürültüsünü azaltması ve ışın oluşturma teknolojisini kullanarak sinyalleri birleştirmesiyle daha uygun bir sinyal yarattığı bu durum ise veri füzyonu olarak adlandırılır.
- *Düğüm yerleştirmesi ve kapasitesi:* Düğüm yerleştirmesi ya belirleyici ya da öz örgütleyicidir. Belirleyici yerleştirmelerde, düğümlerin yerleri önceden manüel olarak tasarlanmıştır. Bu nedenle kümeleme kurulumu gerektirmez ya da önceden kurulmuştur. Enerji tasarrufu ve performans göz önüne alındığında öz örgütleyici sistemleri kullanmak belirleyici sistemleri kullanmaktan daha uygundur. Düğümler aynı şekilde dağıtılmamışsa, enerji tasarruflu ağ işletimi için en uygun kümeleme oldukça zor olur.

### 2.1.2. Kümeleme Hedefleri

WSN uygulamalarının birçoğu çok zordur. Bu uygulamaların gereklerini yerine getirmek için bazı kümeleme amaçları belirlenmiştir. Örneğin, uygulamalar veri gizliliğine, küme içi ve kümeler arası bağlantılığa karşı oldukça duyarlıdır.

- *Yük dengelemesi:* Ağın yaşam süresini uzatmak için eşit büyüklükte kümeler ve düğüm kalitesi sağlanmalıdır çünkü bu, küme başlarının alt gruplarının yüksek oranlardaki enerji kaybını önler. Küme başları veri topladığı zaman veri raporlarının veri istasyonundaki işleme neredeyse aynı zamanda hazır olması için kümelerdeki düğüm sayılarının benzer olması gerekmektedir. Dengelenmiş iş, sistemin durağanlığını artırır ve sistemdeki farklı düğümler arası iletişimi geliştirir.
- *Hata toleransı:* Hata toleransı, zorlu çevrelerdeki düğüm bozuklukları nedeniyle ortaya çıkan kesiklikler olmadan algılayıcı ağının fonksiyonlarının devam edebilmesidir. Küme başlarından kaynaklanan bozukluklar genellikle yeniden kümelemeye gönderilir. Fakat ağ kaynaklarındaki bir yükün yeniden

değerlendirilmesi hem zaman, hem enerji kaybına neden olur. Bu yüzden en iyi yöntem hata toleransı tekniklerini kullanmaktır.

- *Arttırılmış bağlantı ve süresi azaltılmış gecikme:* Uydu bağlantısı gibi bazı uygulamalarda kümeleme başlarının çok uzun mesafelere kadar iletişim yeteneği bulunmaktadır. Küme başları arası bağlantı bu uygulamalarda önemli bir koşuldur. Veri gecikmesi, genellikle küme içi bağlantının tasarım ve sınırlarını ilgilendiren bir konudur. Gecikme, genellikle bir veri yolu üzerinde izin verilen maksimum sayıda “K” sekmesi kurarak hesaplanır.
- *Asgari küme hesabı:* Ağın avantajından faydalanmanın yanında, aynı ağın daha ucuz olanını seçmek için, ağı tasarlayan kişi ağdaki en az sayılı düğümleri tercih eder. Örneğin, küme başları, düğümlerin sınırlı olduğu diz üstü bilgisayarlar ya da cep telefonları olabilir. Bu hedef bir ağın birden fazla kaynağı olduğu zaman gerçekleşir.
- *Azami ağ işlevi:* WSN'nin zorlu uygulamalarında, küme başlarındaki kaynak diğer algılayıcılardan daha fazladır. Küme içi iletişim için enerjiyi en aza indirmek gereklidir, bu nedenle küme başları, kümelerin içlerindeki algılayıcıların birçoğuna yakın yerleştirilmelidir. Algılayıcı düğümleri enerji harcayan yapılar olduğu için onların kullanımı ağın işlev süresini doğrudan etkiler. Birleştirilmiş kümeleme ve yol kurulumu ağın işlev süresini maksimuma çıkarmak için düşünülmeleri gereken yöntemler arasındadır.

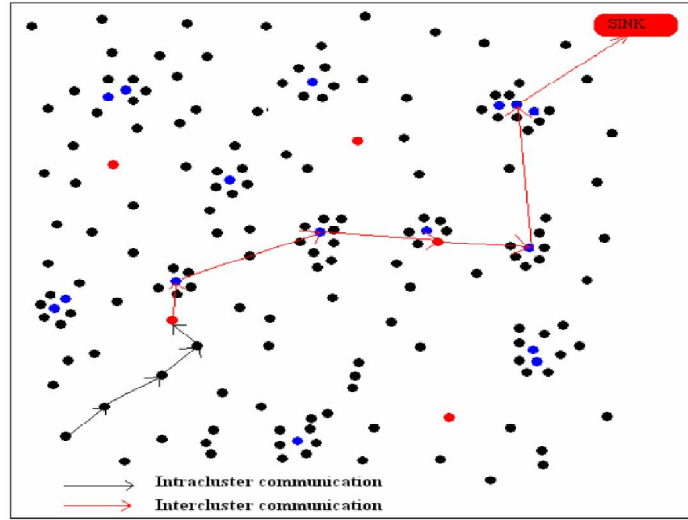
### 2.1.3. Küme Özellikleri

Kümeleri oluşturan bazı özellikler vardır, bu özellikler kümenin dâhili özellikleri arasında da olabilir, diğer kümelerle ilişkili özelliklerde de olabilir.

- *Küme Sayısı:* Yayımlanan bazı makalelere göre, küme başları önceden hesaplanır. Böylece küme sayısı hesaplanabilir. Genellikle, küme başlarının algılayıcı düğümlerinden rastgele seçilmesi farklı sayılarda kümeler verir.
- *İstikrar:* Küme hesabı ve düğümler zamanla farklılaşırsa, bu kümeleme sistemi uyarlanabilir kümeleme olarak adlandırılır. Kümedeki algılayıcı düğümleri aynı kalır, kümelerin sayısı değişmez, ağ süresi boyunca da bir değişikliğe uğramazsa sabit kümeleme olarak adlandırılır.
- *Küme içi topoloji(intra-cluster):* Düğümlerin kendi aralarında ya da küme başlarıyla küme içinde doğrudan bir iletişimi varsa, buna küme içi bağlantı denir.

- *Kümeler arası bağlantısı(inter-cluster)*: Küme başları, iletişimin çok uzak olduğu merkez istasyona doğrudan bağlanamazsa, ya diğer kümelerin bağların bağlanabilir ya da diğer kümeyle bir ağ geçidi yolundan bağlanabilir.

Küme içi ve kümeler arası topolojisi aşağıdaki şekil 2.2’de gösterilmiştir.



Şekil 2.2: Küme içi ve kümeler arası bağlantı[31]

#### 2.1.4. Küme Başının Yeterlilikleri

Kümeler arasındaki küme başlarının yeterlilik dâhilinde şu gibi bazı özellikleri vardır:

- *Mobilite*: Küme başı ve üyeleri sürekli olarak değişiyorsa, küme ya da ağ kendini sürdürebilir ve daha iyi bir ağ performansı sağlar, ancak bu da kümenin istikrarını etkiler. Küme başları durgun bir haldeyken kümelerin içinde istikrar sağlayabilir. Bununla ağın içinde daha az performansla istikrarı koruyabilir.
- *Düğüm Tipi*: Bazı durumlarda, küme başları düğümlerin dağılımıyla hesaplanır. Bu sistem oldukça basittir. Ancak diğer durumlarda bu işlem uzun hesaplamalar ve birçok iletişim kaynakları ile birleştirilerek yapılır.
- *Görevi*: Küme başlarının küme içindeki görevleri içinde aynı küme içindeki algılayıcılardan kaynaklanan trafiği sıralama ve kontrol etmenin yanında veri toplama ve toplanan verinin füzyonunu da tamamlama gibi işlemler yer almaktadır.



### 2.1.5. Kümeleme Süreci

Kümeleme süreçleri ve algoritma özellikleri arasındaki sınıflama, küme başları planıyla büyük ölçüde farklılık gösterir.

- *Yöntem:* Ayrılmış üç bölümden oluşur:
  - *Dağılım:* Algılayıcı düğümleri rastgele dağıtılır ve kümenin başını seçerler.
  - *Merkezileştirme:* küme başı kümenin merkezinden seçilir ve kontrol ettiği diğer küme üyelerine kümenin kontrolünün küme başında olduğunu duyurur.
  - *Hibrit :* Dağıtma ve merkezileştirme metodolojisi vardır.
- *Düğüm gruplandırmasının amacı:* Kümeler kurulurken, hata toleransı, iş dengesi, ağ bağlanabilirliği gibi daha önce de belirttiğimiz birçok amaç mevcuttur.
- *Küme başı seçimi:* Küme başları önceden atanabilir, algılayıcı düğümleri tarafından rastgele seçilebilir ya da diğer yollardan belirlenir.

## 2.2. KABLOSUZ ALGILAYICI AĞLARDA KÜMELEME ALGORİTMALARININ SINIFLANDIRMASI

Kümeleme algoritmaları genelde iki aşamada gerçekleştirilir: kümeleme kurulumu ve kümeleme sürdürülmesi. Kümeleme kurulumu fazında, küme başları ağdaki düğümler arasından seçilir. Küme başları seçildikten sonra, düğüm başları ile ilişkili olan diğer düğümler kümeleri oluşturur. Küme başı olmayan düğümlere üye düğümler denir. Kümenin ucundaki düğümlere *geçit (gateway)* düğümler denir ve bunlar diğer kümelerin gateway düğümleri ile iletişim kurar. Kümeleme sürdürme sürecinde, kümeleme yapılandırması, topolojik değişimler veya düğüm hareketleri yüzünden ilk kümenin kurulmasından sonra değişebilir. Geleneksel kümeleme algoritmaları, güvensiz ve sınırlı bağlantı kapasitesi ve düğüm topolojisindeki değişiklikler sebebiyle gerçek kablosuz ve hoc ortamlarında sınırlarla karşılaşılıyor. Bu sınırlamaları kümeleme başlarını uygun bir şekilde seçerek ve kümeleme sürdürme aşamasında çok fazla hesaplamayı engelleyerek aşmak için birçok deneysel kümeleme algoritmaları sunuldu. Ayrıca, WSN'leri kümelemek için kullanılan dağıtılmış algoritmalar yayınlanmış durumda. Bu ölçeklenebilirliğin ağ kümelenmesinin ana avantajı olarak

görüldüğü düşünülürse, bu algoritmalar yakınsama oranlarına göre değişken ve sabit yakınsama zaman algoritmaları olarak ikiye ayrılırlar.

Bu bölümde, WSN kümeleme algoritmalarını iki kategoriye ayırdık. Birinci kategoride, WSN kümeleme algoritmalarını küme başı seçimi, kümeleme özellikleri ve sürece göre çeşitli şemalara ayırdık. İkinci kategoride, kümeleme algoritmalarını zamanda yakınsama bağlamında iki parçaya ayırıyoruz.

### 2.2.1. Birinci Kategori

#### 2.2.1.1 Deneyimsel Algoritmalar

Şu problemlerden birini veya her ikisini de çözmek için WSN uygulamalarında görülen bir algoritmadır. Bunlardan ilki, en ucuz maliyetle küme oluşturmak ve en uygun çözüm sonuçları elde etmek için gerekli olan çalışma-zamanı (run-time). Bu algoritmadaki belirli ölçüler gerekli değil ve hatırı sayılır performansa dayanır.

Küme başının seçmek için birçok kümeleme algoritması vardır. Herhangi bir kablosuz mobil ağa uygun olmak yerine, bu algoritmaların her biri sadece belirli uygulamalar için uygundur.

*Bağlantılı Küme Algoritması (LCA) :*

LCA ilk gelişmiş kümeleme algoritmasıydı [24,25]. Başta kablolu sensör ağlar için tasarlandı ve daha sonra kablosuz sensör ağlarda kullanıldılar. LCA algoritmasında her düğüme benzersiz bir kimlik numarası verilir. LCA dağıtılmış bir algoritmadır ve belirli bir düğümün varlığına bağımlı değildir.

Bir küme başı oluşturmanın iki yolu vardır:

- Düğüm grubunda en yüksek kimlik numarasına sahip olan düğüm küme başı olarak seçilir.
- Eğer düğümün 1-sekme uzaklıkta komşu düğümü en yüksek ID kimliğine sahip ise ve bu komşu düğümü küme başı değil ise, bu düğüm küme başı olarak seçilir.

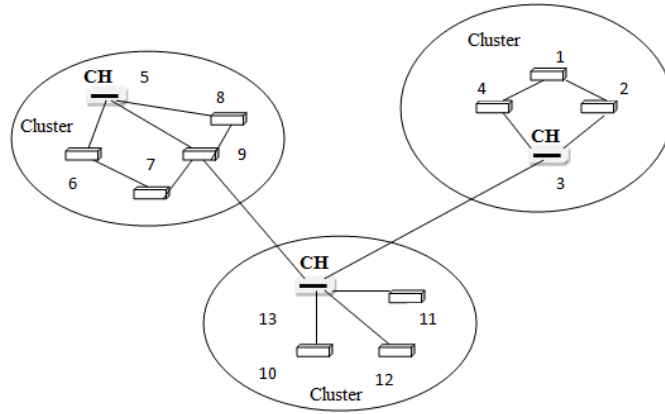
Bu algoritmanın dezavantajı, küme başı yükünün tüm düğümlere eşit olarak dağıtılmamasıdır. LCA'nın başka bir kısıtlaması ise görece yüksek kontrol mesaj gideridir. Ayrıca, LCA düğüm mobilitesini, uyarlanabilir aktarma aralığı ve güç verimliliği sorunlarıyla ilgilenmiyor. LCA algoritması küçük ağlarda(100 düğümden az) kullanmak için tasarlanmıştı.

### *Bağlantılı Küme Algoritması 2(LCA2)*

Bu algoritmada düğüm ikiye ayrılır, açık(non-covered) ve kapalı(covered) [24,25]. Eğer komşularından biri küme başı ise, ona kapalı düğüm denir. Düğüm başları, açık düğümlerden en düşük kimlik numarasına sahip olandan başlanarak seçilir. Seçilen küme başından sonra, yine en küçük ID ve açık düğümlerden olan ikinci küme başı olarak seçilir. Tüm küme başları seçilene kadar işlem devam eder.

### *Yüksek-Bağlanabilirlik Küme Algoritması*

LCA ile karşılaştırıldığında bu algoritma kimlik numaralarına bakmak yerine, bağlanabilirliği kullanıyor[12,26]. Eğer bir düğüm en yüksek sayıda düğüme bağlanırsa ona yüksek bağlanabilir (derece) denir ve çevredeki en yüksek bağlanabilirliğe sahip düğüm, küme başı olarak seçilir, şekil 2.3'e bkz. Bir düğümün bağlanabilirliği, 1-hop komşularına olan bağlantı sayısıdır. Her düğüm duyabildiği düğümlerin bir listesini yayımlar (kendisi de dâhil). Bununla birlikte, eğer düğümler eşit bağlanabilirliğe sahipse, en düşük kimlik numarası (LID – Lowest ID) olan küme başı olarak seçilir.

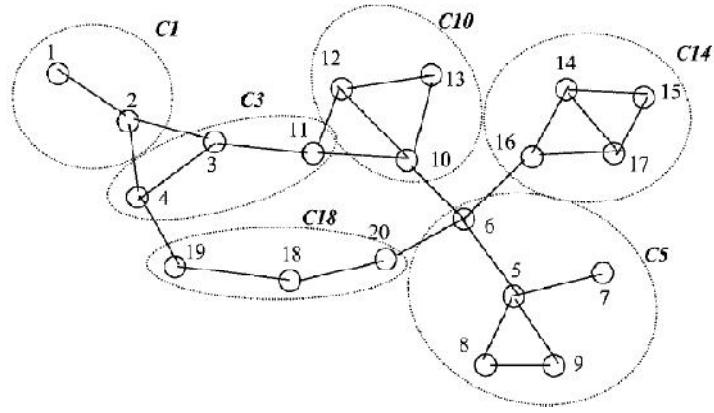


Şekil 2.3 : Yüksek Bağlanabilirlik Küme Algoritması ile kümeleme

LID algoritmasıyla karşılaştırıldığında, HCN daha büyük bir mesaj trafiğine maruz kalıyor, çünkü bağlanabilirlikle ilgili daha çok veri değiştiriliyor. Bununla birlikte, küme başları daha sık değişecek ve böylece yük dağıtımını daha adil olacaktır.

### En Düşük Kimlik(ID) Kümeleme Algoritması

En düşük ID (LID –Lowest ID) kümeleme algoritması 2-sekmeli kümeleme algoritmasıdır [27]. Bu algoritmayı çalıştırırken, bir düğüm periyodik olarak duyabildiği düğümlerin listesini (kendisi de dahil) yayımlar. 1-sekme komşuluktaki düğümlerden sadece ID'si kendininkinden yüksek olan düğüm kendisini küme başı olarak ilan eder. Daha sonra ID'sini ve küme ID'sini yayımlar. Şekil 2,4 LID yapısının bir örneğini gösteriyor, burada 1, 3, 18, 10, 14 ve 5 numaralı düğümler küme başları olarak seçilmiş çünkü kendi kümelerindeki en küçük ID'ye sahipler.



Şekil 2.4 : LID yapısı örneği

### Least Cluster Change Algorithm( LCC)

**En düşük küme değişimi** Algoritması, belli koşullar altında önemli olan stabilite faktörünü sağlayabilmek için küme başı değişiminin sıklığını azaltmaya çalışır [28].

Bu algoritmada, küme başları sadece iki durumda değişebilir:

1. Eğer küme başları diğer küme başlarıyla yayın (aktarım) alanındaysa.
2. Eğer başka bir kümede düğüm küme üyeliğini kaybederse ve yeni bir küme oluşturursa.

LCC ilk kümeler veya küme başlığı seçmek için LID veya HCN kümeleme algoritmalarını kullanır. Küme başlarının değişimi en aza indirildiği için, düğümler kümelerle birleştiğinde veya ayrıldığında küme yapıları değişmeyecektir. LCC kümeleme algoritması, ağ topolojisinin sıkça değiştiği ve düşük yönlendirme masrafı ve gecikmesi olan ortamlarda güçlüdür. Bununla birlikte, yük dağıtımını adaletsiz olacaktır.

### Maks.- Min D-Küme Algoritması

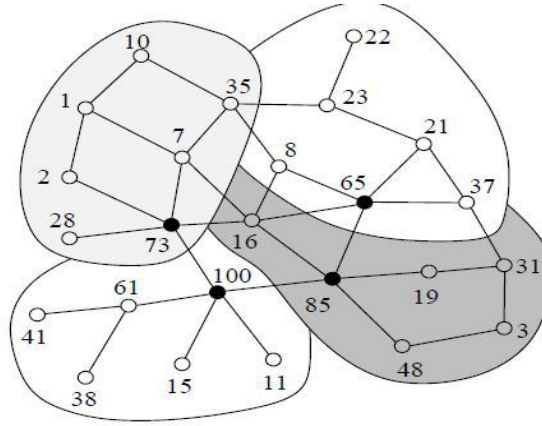
Yeni bir dağıtılmış küme başı seçim süreci olarak sunuldu. Ref [29]'te gösterilen Maks- Min. D-Küme (MMD) algoritması, küme başları arasında adil bir yük dağıtımını sağlamak adına D-hop kümeleri oluşturmak için yük-dengeleme (maks – min) yöntemini kullanıyor. D-hop kümesinde, her düğüm küme başından en fazla D-hop kadar uzakta bulunur. Algoritma saat senkronizasyon ek yükünden kaçınmayı, düğümler arasında gönderilen mesaj sayısını bir  $O(D)$  zaman kompleksine sınırlamayı, küme stabilitesini arttırmayı ve D'nin bir fonksiyonu olarak küme başlarının yoğunluğunu azaltmayı ve LCA algoritmasındaki gibi küme başlarını düğüm ID'sine göre seçmeyi amaçlıyor. Algoritma, bilgilerin bilgi merkezine gönderilmesinde kullanılan enerjiyi azaltmaya yönelik bir amacı yok.

MMD algoritmasında  $2*D$  tur mesaj takası var ve her düğüm, düğüm ID'leri için  $2*D$  boyutunda *WINNER* ve *SENDER* adlı iki dizi tutmak zorundadır. Skl. 6.9'da gösterildiği gibi, *WINNER* ve *SENDER* dizileri sırasıyla kazanan düğüm ID'sini ve her turda kazanan düğüme gönderilen düğüm ID'sini tutar. Önce, her düğüm ID'sini *WINNER*'a ayarlar. Algoritma 4 aşamadan oluşur: *floodmax*, *floodmin*, küme başlarını belirlemek ve kümeleri bağlamak. *Floodmax* aşaması, her düğümden 1-hop komşularına *WINNER* değerlerinin D-tur mesaj takasından oluşur. Her turda, her düğüm kendi *WINNER* değerini tüm 1-hop komşularına iletir ve *WINNER* değerini aldığı en büyük değere günceller. O halde, *floodmax* en büyük düğüm ID'sini D-hop komşularına iletir ve sonda kalan düğüm ID'si küme başı olarak seçilir. Bununla birlikte, bu mekanizma küme başları için dengesiz yükü sonuçlanabilir. *Floodmax*'tan sonra D tur *floodmin* en küçük ID'leri yaymaya başlar. *Floodmax*'ın tersine, her düğüm *WINNER* değerini en küçük düğüm ID'sine günceller. Her iki aşamada da bir düğümden en az bir kere *WINNER* olarak görünen düğüm ID'si **düğüm çifti** olarak adlandırılır. Şekil 2.5'te Maks.- Min D-Küme algoritmasının yapısını göreceksiniz.

Her düğüm şu kurallara göre, *floodmax* ve *floodmin* için *WINNER*'daki girdilere göre küme başını belirler:

- *Kural 1*: Eğer bir düğüm ikinci turda kendi ID'sini alırsa, kendisini küme başı olarak ilan eder ve diğer kuralları geçer. Diğer şekilde ise, Kural 2'ye geçer.

- *Kural 2:* Tüm düğüm çiftleri arasında, bir düğüm küme başı olması için minimum düğüm çiftini seçer. Eğer düğüm için bir düğüm çifti yoksa Kural 3 uygulanır.
- *Kural 3:* Flooding aşamasının ilk d turunda maksimum düğüm ID'sini bu düğüm için küme başı olarak seçilir.



Node	10	1	2	7	35	8	23	22	21	65	37	31	19	85	16	100	73	28	41	61	11	48	3	15	38
Max 1	35	10	73	73	35	65	35	23	65	85	65	37	85	100	85	100	100	73	61	100	100	85	48	100	61
Max 2	35	73	100	100	73	85	65	35	85	100	85	85	100	100	100	100	100	100	100	100	100	100	100	100	100
Max 3	73	100	100	100	100	100	85	65	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Min 1	73	73	100	100	73	100	65	65	85	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Min 2	73	73	73	73	65	73	65	65	65	85	85	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Min 3	65	73	73	65	65	65	65	65	65	65	65	85	100	85	73	100	73	100	100	100	100	100	100	100	100
Result	73	73	73	73	73	65	65	65	65	65	65	85	100	85	100	100	73	100	100	100	100	100	100	100	100

WINNER

Şekil 2.5: Min-Max D-algoritmasının yapısı[29]

Küme başlarını belirledikten sonra, her düğüm küme başını tüm komşularına yayınlar. Eğer bir düğümün tüm komşuları düğümle aynı küme başına sahipse, bu düğüm normal bir düğümdür. Eğer herhangi bir düğüm farklı bir küme başına sahipse, o zaman bu düğüm bir gateway düğümü olur.

LCA ile karşılaştırıldığında, MMD kümeleme algoritması daha az küme başına, daha büyük küme boyuna ve daha iyi küme başı stabilitesine sahiptir.

### 2.2.1.2. Ağırlıklı Şemalar

#### Ağırlıklı Kümeleme Algoritması (WCA)

Küme seçimi için periyodik olmayan bir süreçtir, her istendiğinde ağ topolojisinin yeni bir yapılandırmasını çağırmaya dayanır[30]. Bu kümeleme algoritması ilk küme başı seçiminde uzun süreli bir mimari bulmaya çalışır. Bu algoritma, ideal derece, güç, mobilite ve mobil düğümlerin pil gücü, bir düğümün komşularına uzaklığı, düğüm hızı ve küme başı olarak geçirdiği süre gibi kombine ağırlık ölçülerine dayanır.

Her düğüm ağırlık değerini diğer düğümlere yayınlar. Eğer bir düğüm komşuları arasında en yüksek ağırlığa sahipse, o düğüm küme başı olarak seçilir; değilse komşu kümeye katılır.

Düğüm bir küme başıyla iletişimi kaybettiğinde, yeni bir kümeleme şeması bulmaya çalışır, bu sürece yeniden-seçim süreci denir. WCA'nın sebep olduğu gecikme ve ek yük çok ağırdır çünkü bir düğüm kendi kararını vermeden önce tüm komşularından gelen yanıtları beklemek zorundadır. Algoritmanın diğer dezavantajı da enerji harcamasıdır. Küme başının seçimi kombine ağırlık olarak adlandırılan bazı global parametrelere dayalıdır [30]:

$$W_v = w_1 \Delta_v + w_2 D_v + w_3 M_v + w_4 P_v.$$

$w_1$ ,  $w_2$ ,  $w_3$  ve  $w_4$ , sistem parametreleri için ağırlık faktörleridir. Ağırlık faktörleri ( $w_1$ ,  $w_2$ ,  $w_3$  ve  $w_4$ ) sistem gereksinimlerine göre seçilebilir. Bu faktör özel uygulamaya dayalıdır. Her düğüm için, kombine ağırlıklar ağ üzerinden belirlenir ve iletilir.

Birinci bileşen  $w_1 \Delta_v$  kümedeki maksimum düğüm sayısını belirlemek için MAC işlevselliği için kullanılır. İkinci bileşen  $w_2 D_v$  komşulardan ortalama uzaklıktır.

Üçüncüsü  $w_3 M_v$  düğümlerin mobilitesidir. CH'lerin stabil bir ağ oluşturmak için yavaş hareket etmesi tavsiye edilir. Sonuncusu,  $w_4 P_v$ , düğümdeki enerjidir.

En küçük  $W_v$ 'ye sahip düğüm küme başı olarak seçilir. Küme başı tekrar seçilmez çünkü bu enerji harcanması anlamına gelir.

### 2.2.1.3 Hiyerarşik Şema

#### LEACH

Heinzelman *et al.* tarafından sunulmuştur [32]. LEACH protokolü, küme-temelli operasyonlar WSN'lerdeki enerji tüketimini minimuma indirmeyi amaçlar. LEACH düğümleri yerel kümeler olarak organize eder: her kümede, küme başı olarak adlandırılan bir düğüm var, şekil 2,6'da LEACH in yapısını gösterilmiştir. Küme başı enerji durumuna göre zamanla değişiyor ve her düğüm küme başı olabiliyor. LEACH protokolü şekilde gösterilmiştir (11). Rastgele seçilen küme başları ve üye düğümleri eşit olarak dağıtılabilir. LEACH küme başlarının rastgele rotasyonu ile enerji kullanımını dengeler.

LEACH birçok turda çalışır ve her tur iki aşamaya ayrılır:

- a- İlk aşamada, küme başları belli bir eşik  $T(n)$  değerine göre seçilir ve kümeler yaratılır. Bu faza kurulum fazı denir.

$T(n)$  şu şekilde hesaplanır:

$$T(n) = \begin{cases} \frac{P}{1 - P^{*(r \bmod \frac{1}{P})}} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases}$$

$T(n)$ : Eşik değeri.

P: Kümelerin istenen oranı (olasılığı) = 0.05.

R: Tur sayısı.

G: Düğüm seti.

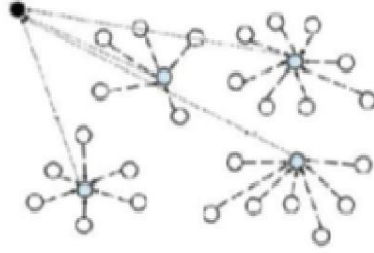
Küme seçimi sırasında, her düğüm 0 ile 1 arasında rastgele bir sayı seçer. Eğer düğümün küme başı için seçtiği rastgele sayı eşik değerden küçükse ( $<T(n)$ ), n sensörü bu tur için küme başı olur.  $T(n)$  parametresi bir sonraki tur için arttırılır çünkü küme başı için aday olan düğüm sayısı daha azdır.

Kurulum aşaması iki safhadan oluşur:

- Duyuru fazı: Küme başını eşik değerine göre seçtikten sonra, Küme başı duyuru mesajını üye düğümlerine gönderir ve küme başı olduğunu anlatır.
- Küme kurulum fazı: Üye düğümler düğümün katılım paket ID'sini gönderir ve küme bunu kurar ve küme başına kümesinin bir üyesi olduğunu ve kümenin kurulduğunu anlatır.
- Program yaratma: Küme başları sensör düğümlerin küme başlarına veri gönderecekleri zamanı belirlerler. Bu seçim, zaman bölümüne çoklu erişim (TDMA – time division multiple Access) yaklaşımı ile yapılır, bu yaklaşım da istikrarlı durum fazı boyunca sürdürülür.

b- İkinci safha, istikrarlı durum fazı olarak adlandırılır: Küme başı ve küme WSN'de kurulduktan sonra üyelerden küme başlarına ve sonra küme başından çıkış düğümüne veri iletimi olur. Bu operasyonu için, LEACH, CSMA tabanlı rastgele erişim şemasının çoklu küme başlarından duyuru çökmelerini engellemesine güvenilir. İstikrarlı durum bittikten sonra, ağ tekrar kurulum safhasına girer ve tekrar küme başları seçer. Ek yükü azaltmak adına, istikrarlı durum fazının süresi kurulum fazından daha uzundur.





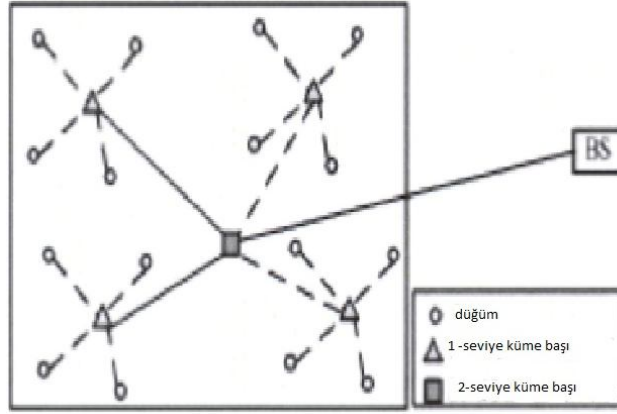
Şekil 2.6 : LEACH protokolü

### *TL-LEACH*

İki-Seviyeli Hiyerarşik LEACH (ya da TLLEACH), LEACH algoritmasına ek olarak sunulmuştur [33]. LEACH'deki bir seviye yerine iki seviye küme başı sunar: Birincil ve ikincil:

- 1) İkincil CH'ler kümelerindeki düğümlerden veri toplarlar. Veri-birleştirmesi bu seviyede yapılabilir.
- 2) Birincil CH'ler kümelerindeki ikincil düğümlerden veri toplarlar. Veri-birleştirmesi bu seviyede de yapılabilir.

Küme TDMA zaman-slotunu kullanarak diğer kümeyle iletişim kurar. TL-LEACH kullanmanın avantajı, baz istasyonuna iletilmesi gereken düğüm sayısını azaltıyor, bu da etkin bir şekilde gereksiz enerji harcanmasını azaltıyor ve orijinal protokole daha uzun yaşam sağlıyor. Sonuç olarak, TL-LEACH protokolü orijinal LEACH protokolünden daha fazla paket iletebiliyor. TL-algoritması aşağıdaki şekil 2.7'de gösteriliyor.



Şekil 2.7 : TL-LEACH

### EECS

Enerji Etkin Kümeleme Şeması (EECS): EECS kümelerin baz istasyonuna uzaklığına bağlı olarak kümeleri dinamik olarak boyutlandırma ile küme yaratarak LEACH algoritmasını ilerletiyor [34]. LEACH küme başlarını, düğümlerin küme başlarına olan minimum uzaklığına göre oluşturuyor. EECS, küme başı adaylarının bir turda küme başı olmak için yarıştığı kümeleme algoritmasıdır. Bu rekabet, adayların rezidüel enerjilerini diğer adaylara yaymamasını içerir. Eğer bir düğüm daha fazla enerjili bir düğüm bulamazsa, küme başı olur.

Sonuç, veri iletmek için baz istasyonuna uzakta bulunan kümelerin yakında olanlardan daha fazla enerjiye gereksinim duyması problemini çözen bir algoritmadır. Bu, enerjinin ağda daha iyi dağıtılmasını sağlayarak daha iyi kaynak kullanımı ve artmış ağ zamanı sağlar.

### HEED

Melez Enerji-Etkin Dağıtımlı Kümeleme (HEED), rezidüel enerji ve iletişim masrafını azaltarak LEACH'i ileri taşır[35]. Eğer küme doğrudan baz istasyonla iletişime geçerse, buna tek-sekmeli(single-hop) kümeleme denir. Eğer baz istasyonla iletişime geçmeden diğer kümelerle de iletişime geçiyorsa, buna çok-sekmeli(multi-hop) kümeleme denir. Her sensor düğüm çoklu güç seviyelerine sahiptir.

HEED'in beş ana hedefi vardır:

- (1) Enerji tüketimini dağıtarak ağ ömrünü uzatmak
- (2) Küme başlarının kurulum safhasındaki enerji tüketimini azaltmak.
- (3) Kontrol aşırı yükünü minimuma indirmek (Düğüm sayısı).
- (4) İyi dağıtılmış küme başları ve yoğun kümeler üretmek.
- (5) Belli bir sabit tekrar aralığında kümeleme işlemini sonlandırmak.

HEED iki kümeleme parametresinin bir melezini üzerinden periyodik olarak küme başlarını seçer:

- **Birincil parametre** her sensor düğümünün rezidüel enerjisidir. Birinci parametre olasılıksal olarak ilk küme başlarını seçmek için kullanılırken, ikinci parametre bağları kırmak için kullanılır.
- **İkincil parametre** komşu yakınlığı veya küme yoğunluğunun bir fonksiyonu olarak küme için iletişim masrafıdır. Küme içi iletişim masrafı, Ortalama Minimum Yeniden Şart Edilebilir Gücü (AMRP) ölçülerek hesaplanır. AMRP, R küme uzaklığındaki her düğümün küme başı ile iletişim kurabilmesi için gereken minimum güç seviyelerinin ortalamasıdır.

$$AMRP = \frac{\sum_{i=1}^M MinPwr_i}{M}$$

MinPwr<sub>i</sub>: bir düğümünü küme başı ile iletişim kurması için gerekli olan minimum güç seviyesini gösterir  $v_i, 1 < i < M$ .

M: Küme aralığındaki düğüm sayısıdır.

Her sensör düğümdeki kümeleme birçok tur gerektiriyor. Her tur, küme alanındaki herhangi bir komşudan mesaj alabilmeye yetecek kadar uzundur. LEACH'te olduğu gibi, ağdaki belirli sayıda küme başları, Cprob, önceden belirlenir. Cprob parametresi sadece ilk küme başı bildirimlerini kısıtlamak için kullanılır ve son küme yapısına bir etkisi yoktur. HEED'de, her sensör düğüm şu şekilde küme başı olma ihtimali CHprob'u şu şekilde hesaplar:

$$CH_{prob} = C_{prob} \times \frac{E_{residual}}{E_{max}},$$

Burada Eresidual sensör düğümdeki tahmini rezidüel enerji, Emax da genelde her düğüm için aynı olan maksimum enerjiye (tam şarj olmuş pile denk gelir) referanstır. CHprob değeri pmin eşik değerinden büyük olmalıdır. Küme başı hem deneysel küme başı, CHprob <1 ise, ya da son küme başı, CHprob 1'e ulaşmışsa, olabilir.

HEED'in her turunda, küme başından haber alamayan her sensör kendisini CHprob ihtimali ile küme başı olur. Yeni seçilen küme başları şimdiki küme başlarının yeni setine eklenirler. Eğer sensör düğümü küme başı olmak için seçilirse, bir deneysel küme başına ya da son küme başına bir duyuru mesajı iletir. Küme başı listesini duyan bir sensör düğüm bu küme başı listesindeki en düşük masraflı olan küme başını seçer. Her küme daha sonra CHprob'unu ikiye katlar ve sonraki adıma geçer.

Eğer bir düğüm kendini bir küme başı olarak seçmeden veya bir kümeye katılmadan HEED uygulamasını tamamlarsa, kendisini son küme başı olarak seçer. Bir deneysel küme başı daha düşük masraflı küme başından haber alırsa sonraki bir tekrarda normal bir düğüm olabilir. Bir düğüm yüksek rezidüel enerji ve düşük masrafa sahipse, peş peşe kümeleme aralıklarında küme başı olarak seçilebilir.

WSN'nin düğümlerin beklenmedik şekilde ölmediği durağan bir ağ olduğu varsayıldığından, düğümlerin komşulukları pek sık değişmez. HEED sabit bir tekrar sayısı O(n) sonra ağ çapından bağımsız olarak sonlandırır.

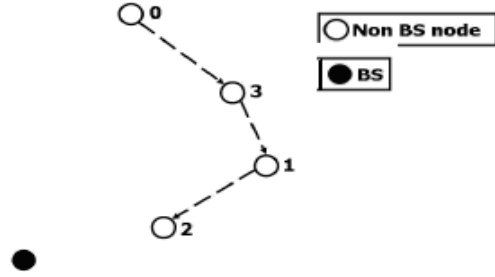
### 2.2.1.2. Izgara Şemalar

#### *Sensör Bilgi Sisteminde Güç-Etkin Toplama (PEGASIS)*

LEACH iyileştirmesi olarak zincir tabanlı protokol sağlar [36]. LEACH'te olduğu gibi, PEGASIS homojen düğümlerin kullanılmasına dayanır, çıkış düğümünün durağan olduğunu varsayar ve mesaj iletimi sırasında harcanan enerji iletimin uzaklığına bağlıdır.

PEGASIS'in ana fikri, sensör düğümleri arasında her düğümün şekil 2.8'deki gibi yakın komşusuna mesaj gönderip ondan mesaj alabileceği bir zincir oluşturmaktır. Toplanan veri düğümden düğüme aktarılır, birleştirilir ve sonra sonunda seçilen bir düğüm (lider

düğüm) veriyi her turda bir baz istasyonuna iletir. PEGASIS veri zinciri oluşturmak için hırslı bir algoritma kullanır ve bu yüzden düğüm sadece yakın komşu düğümleriyle iletişim kurar. Sadece bir düğüm lider veya küme başı olarak seçilir.



Şekil 2.8 : PEGASIS

PEGASIS'in Dezavantajları:

- Protokol operasyonu global ağ bilgisini depolamak için yüksek miktarlarda yerel hafızaya ihtiyaç olabilir.
- Global ağ bilgisi protokol ölçeklenebilirliğine pozitif katkı yapmaz.
- İletim liderinden en uzaktaki kaynağa çok büyük gecikmeler olur.
- Paketler belirgin gecikmeler yaşayabilir, özellikle de zincirdeki uzak düğümlerden kaynaklanıyorsa.

PEGASIS'te görülen gecikme sorununu çözmek için, çok seviyeli zincirleme kullanılabilir. Enerji etkinliğine göre, PEGASIS LEACH'ten daha iyi çalışır (%100'den %300'e kadar). Bu iyileştirme, küme oluşturulmasıyla karşılaştırıldığında zincir iletişimde sınırlı ek yük dolayısıyladır. Sonuç olarak, PEGASIS her tur için enerji miktarını azaltır ve ömrü arttırır.

### 2.2.2. İkinci Kategori

Birçok kümeleme stratejileri vardır. Kümeleme algoritmaları zamanla yakınsamalarına (convergence in time) göre ikiye ayrılırlar:

#### 2.2.2.1. Değişken yakınsama zaman algoritmaları

Zaman, küme algoritmalarının yakınsamalarında önemli bir faktördür, bu algoritmalar  $O(n)$  yakınsama zamanı vardır, burada  $n$  ağdaki sensör düğüm sayısını gösteriyor. Genelde, değişken yakınsama zaman algoritmaları sabit zamana göre küme

özelliklerinde daha fazla kontrole izin verir. Protokollerden bazıları LCA, RCC ve CLUBS'tır.

#### *Rastgele rekabet temelli kümeleme (RCC)*

Bu algoritmanın amacı mobil ağları desteklemektir [37], stabilite ve sadeliği amaçlar. Bu algoritmanın temel fikri, herhangi bir kümeye ait olmayan bir düğümün sensör düğümlere CH olarak adaylığıyla ilgili istekler göndererek bir küme oluşturmasıdır.

Bu algoritmalar topolojik bilgi ya da global ID'lere erişim olmadığı zaman uygulanabilir. Bununla birlikte, kümeler CH'lerini 1-hop mesafede tuttuklarında uygulanabilir değildir çünkü küme başarısız olacak ve yapılanma süreci tekrar başlayacaktır.

Küme başı ve küme seçim sürecine nasıl başlanır?

İlk düğüm paketini komşusuna gönderir ve komşularından bir küme başı seçer. Böyle bir paket duyan düğüm, küme başı olmak için haklarını kaybeder ve kümenin bir üyesi haline gelir. Kümeleri sürdürmek için, periyodik olarak, ağdaki her CH bir CH ilan paketi yollar. Küme başının ilan paketini komşularına gecikmeli göndermesinden kaynaklı gecikme yüzünden, birçok komşu bir düğüme aynı anda küme başı ilan paketini yollar, bu da ağda çatışmaya yol açar. Bu anlaşmazlığı engellemek için, rastgele zaman kullanılır. Eğer bu zamanda bir CH'nin ilan mesajını başka bir düğümden alırsa, düğümlerle iletişimi koparır. Eğer rastgele zamanlama aynı anda yayın yapma sorununu çözemezse, düğüm ID'sini kullanır, en düşük ID'li düğüm CH olur.

#### *2.2.2.2. Sabit yakınsamalı zaman algoritmaları*

Birkaç tekrarda yakınsayan algoritmalar kümenin boyu ile ilgilenmezler. Bu algortmada, algortmadaki her düğüm, yerel uygulama algortmasını bağımsız olarak uygulama stratejisini izler. Ve kendi bölgesi ve komşuluklarındaki üye kümelerinin kararına bağlıdır. Bu protokollerden bazıları LEACH, DWEHC, FLOC, ACE.

#### *Dağıtımli Ağırlık Temelli Enerji Etkin Hiyerarşik Kümeleme (DWEHC)*

Dengeli küme boyutlarına ulaşmayı ve WSN'ler için küme içi topolojileri optimize etmeyi amaçlayan başka bir kümeleme algortmasıdır [38].

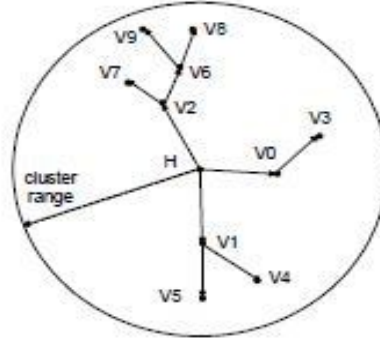
Küme çok seviyeli üye düğümleri içerir.

**Birinci seviye üyeler:** Düğüm doğrudan bağlantı ile CH'ye bağlanır.

**İkinci seviye üyeler:** Düğüm iki hop bağlantısı ile düğüme bağlanır.

**Küme alanı:** Bir küme üyesi düğümden onun küme başına olan en uzak iletim alanı olan parametredir, bu alan tüm ağda sabittir.

DWEHC küme içi yapıda multi-hop kullanılır. Her küme çok seviyeli alt kümeye sahiptir. Şekil 2.9’da gösterilmiştir.



Şekil 2.9 : DWEHC

Küme başı ve küme seçimi:

Küme oluşturma süreci tüm düğümler en enerji-verimli küme içi topolojiyi oluşturana kadar devam eder. Süreç, her düğüm için yedi tekrarda küme oluşturur ve  $O(1)$  zaman karmaşıklığında sonlanır çünkü her düğüm en fazla altı komşuya sahip olabilir [38].

Başlama sırasında, her sensör düğümü, komşuluğunu bulmak için  $(x,y)$  koordinatlarını yayımlar. Bölgesindeki komşu düğümleri bulduktan sonra, ağırlığını şu şekilde hesaplar:

$$w_{\text{weight}}(s) = \left( \sum_{u \in N_{a,c}(s)} \frac{(R-d)}{6R} \right) \times \frac{E_{\text{residual}}(s)}{E_{\text{initial}}(s)},$$

R:Küme aralığı.

$E_{\text{initial}}$ :  $s$  düğümündeki başlangıç enerjisi,

$d$  :  $s$  düğümünden komşu  $u$  düğümüne olan uzaklık.

$N_{a,c}(s)$ :  $s$  düğümünün komşuları.  $A$ , 2 ile 4 arasında bir sayıya eşit olan iletili gücü ve  $c$  sabit sayı.

$E_{\text{Residual}}(s)$  :  $s$  düğümünün rezidüel enerjisi.

Diğerlerinin arasından en yüksek ağırlığa sahip olan düğüm geçici küme başı olur.

Geçici küme başı, komşularından belli bir yüzdedeki düğüm onu küme başı olarak seçerse, gerçek bir küme başı da olabilir. Bu yüzde ilk tekrarda ( $i = 0$ ) %100'dür ve sonraki altı tekrarda ( $i < 6$ ),  $(6 - i) / 6$ 'ya kadar düşer.

DWEHC algoritması, tamamen tüm ağda dağıtılır. Her düğüm ya bir küme başı ya da kümedeki bir alt üyedir. Her küme, yerel olarak optimum olan minimum-güç topolojisini içerir. Her ana düğüm sınırlı sayıda alt düğüme sahiptir, bu da ölçeklenebilirlik açısından önemlidir. Bu algoritma, her düğüm için iyi bir yük dengesi sağlıyor, böylece küme başının ömrünü uzatıyor. Hem DWEHC hem de HEED küme başı seçiminde enerji korunumuna dikkat ediyor ve ağ boyu ile ilgili varsayımlar yapmıyor. Bununla birlikte, DWEHC ile yaratılan kümeler HEED ile yaratılanlardan daha iyi yük dengesine sahiptir.



### 3. YÖNLENDİRME

Kümeleme algoritmasının tanımından sonra, kablosuz algılayıcı verilerinin nasıl yönlendirileceği problemi üzerine çalışacağız. Yönlendirme tekniği, bir kaynak düğümden alıcıya çarpaz kablosuz algılayıcı ağ içindeki bir veya daha fazla geçiş için en iyi yolun tasarlanması ve belirlenmesi sürecidir. Kullanılan çok sayıda WSN algılayıcı düğümleri, düğüm kimliği ağda fazla yüklenen neden olur, geleneksel İP tabanlı protokolleri WSN'ye uygulanamayabilir. Ek olarak, veri almak kaynak gönderenin özel kimliğini bilmekten daha önemlidir.

Ağ üzerinde mesaj iletim sayısını en aza indirmek için en iyi çözüm yönlendirme protokolleridir, çünkü veri paketleri iletilirken enerjinin çoğu tükenir. Pek çok protokolün amacı enerjiyi minimize etmek ve algılayıcı ağların yaşam süresini en üst düzeye çıkarmaktır, çünkü düğümler enerjiyi, bellek ve CPU işlem sürecinin kapasitelerini sınırlarlar.

#### 3.1 YÖNLENDİRME TASARIMI İÇİN ÖNEMLİ PARAMETRELER

WSN yönlendirme protokollerinin tasarımını etkileyen faktörler önemlidir çünkü protokol geliştirme kılavuzluğu sağlamalıdır ([05] ve [39]). Bu parametreler yönlendirme protokolleri arasında analiz ve karşılaştırma amaçları için kullanılabilir. Kablosuz algılayıcı ağlar için iyi protokollerin tasarlanmasına ek olarak, algılayıcı uygulamaları için önemli olan parametreleri anlamak da önemlidir. Protokolleri uygulamada yararlı birçok yol vardır. Söz konusu faktörler şu şekilde özetlenebilir:

**1. Düğüm yoğunluğu:** Düğümler belirli algılama alanları içinde algılama kapsama alanını maksimize etmek için oldukça fazla yoğunlaşmış olabilir. Yönlendirme protokolleri optimal yönlendirme yollarını yoğun dağıtım alanı içinden seçebilmelidir. Düğüm yoğunluğu standart olarak hesaplanabilir:

$$\mu(R) = N \cdot \pi \cdot R / A \dots (1)$$

N, A bölgesindeki algılayıcı düğüm sayısını ve R maksimum iletim alanını ifade etmektedir.

**2. Ölçeklenebilirlik:** Protokolleri tasarlarken göz önünde bulundurulması gereken bir unsur da bir ağ içindeki binlerce düğümüne sahip olmasına imkânını elde edebilmesidir. Bu olasılıkla birlikte, herhangi bir aşamada daha fazla düğüm eklenilebilir, protokoller ve yönlendirme planları ölçeklendirilebilir.

**3. Dağıtım:** Algılayıcı ağları yüzlerce ya da binlerce düğüm içerebilir ve bunlar uzak ve tehlikeli bölgelere dağıtılma ihtiyacı duyabilirler. Eğer bu düğümler yeterince küçük ve ucuz iseler, başka türlü extra bilgi almanın mümkün olmadığı durumlarda uzak veya tehlikeli alanların üzerinde uçan bir uçaktan yüzlerce ve binlerce micro algılayıcıların atılabileceğini hayal edebiliriz.

Topolojiyi tipik olarak üç aşamaya ayrılabilir:

- Düğümünün dağıtımını rastgele dağıtılmış veya ya da önceden belirlenmiş konumlara yerleştirilmiş olabilir.
- Topoloji değişikliklerinin olduğu pozisyon dağılımı aşaması bozulmalara, hareketliliğe ve engellere neden olabilir.
- Topolojinin değiştiği ilave alan için düğümlerin yeniden düzenlenmesi zararlı ağ performansına neden olur.

**4. Enerji değerlendirmesi:** WSN'nin yaşam süresi enerji kaynaklarına ve sensörlerin tüketimine dayandığı zaman, enerji değerlendirmesi yön tasarımı üzerinde büyük bir etkisi vardır. Aktarım sırasında tüketilen güç herhangi bir düğümün enerji tüketiminin büyük kısmıdır.

**5. Güvenilirlik (Hata toleransı):** Düğüm hataları WSN'deki güç azalması, hasar veya engel nedeniyle kaçınılmazdır. Hata durumları için bir yönlendirme protokolünün toleransı ağın kalan işlevselliği üzerindeki düğümlerde olan başarısızlığın etkisine dayalı olarak ölçülür. Algılayıcı düğümün güvenilirliği [39] algılayıcı düğümün 't' aralığı içinde bir arıza ile karşılaşılması olasılığı bir Poisson dağılımı ile modellenmiştir.

$$R_k(t) = e^{-\lambda k \cdot t}$$

Buradaki  $\lambda k$ , k düğümünün başarısızlık oranı ve t zaman periyodudur.

6. **Bağlanabilirlik:** Bir ağın her düğümü etkili iletişim kurabilen herhangi bir komşu ile bir bağlantıya sahiptir. Yüksek düğüm yoğunluğu ağ içinde düğümlerden izole edilmeyi önler. Bu olası bağlantılar WSN topolojisinin ve düğüm hatalarından dolayı olan küçülmelerin değişen doğasına dayanır. Son olarak, bağlanabilirlik ağ içindeki düğümlerinin rastgele dağılma ihtimaline bağlıdır. Eğer ağ bağlantısı düğüm hareketliliği, düğüm hataları veya düğüme müdahale edilmesi nedeniyle koparsa, bağlanabilirlik düzensizdir.

7. **Yaşam süreci:** Amaçlanan uygulama WNS' nin gerekli fonksiyonel ömrünü belirler. Fonksiyonel yaşamın düğüm sağlamlığı ve yönlendirme protokolünün enerji verimliliği üzerinde doğrudan bir etkisi vardır.

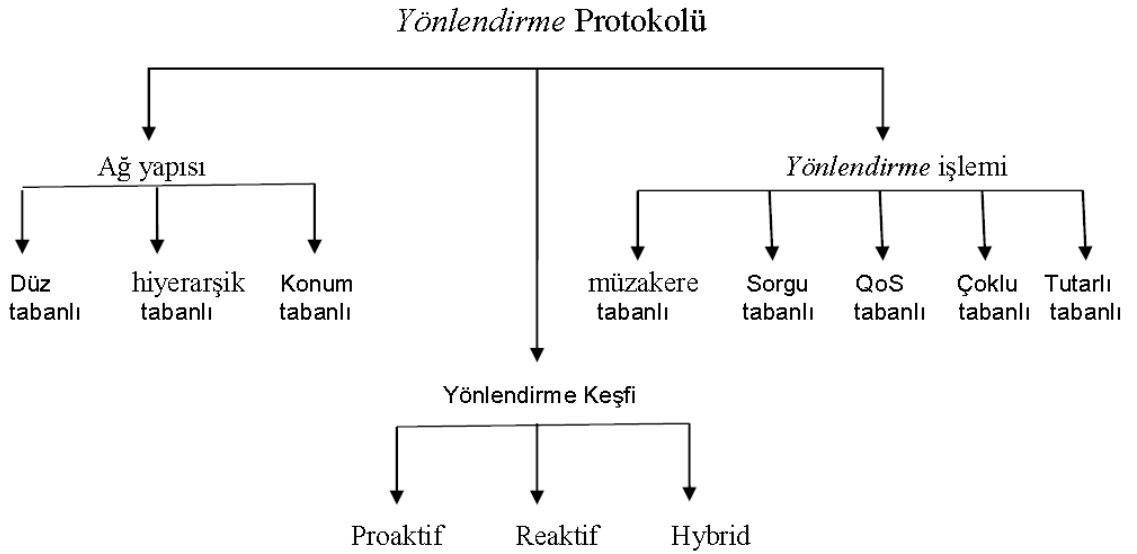
8. **Çalışma ortamı:** Algılayıcı düğümler ve ağlarının, özel amaçlı düğümler ve ağların aksine daha tehlikeli bir ortama yerleştirilmiş olması daha muhtemeldir. Çevreye bağlı olarak, yapı, bileşen ve tolerans seviyesini algılama ile ilgili kaygılar olabilir.

9. **Üretim Maliyetleri:** Tek bir algılayıcı düğümün üretim maliyeti ağın düğümlerinin çok sayıda olması nedeniyle kablosuz algılayıcı ağlar içinde çok önemli bir mesela haline gelir. Tek bir düğümün maliyeti \$1 dan daha az olmalıdır ki, binlerce veya milyonlarca düğümün kablosuz algılayıcı ağı finansal açıdan mümkün olsun [05].

10. **Hareketlilik:** Algılayıcı düğümlerin hareketi uygulamalar tarafından gerekli olabilir veya tesadüfi bir etki söz konusu olabilir ve ağ düğümlerinin herhangi bir yüzdesine uygulanabilir. Hareket sabit bir hızda olabilir veya durağan kalan dönemleri aralıklarla olabilir. Düğümlerin hareket şekli tesadüfi veya önceden seçilen bir model ile belirlenebilir. Örneğin, askerlerin devriye gezmesi bir alanın çevresi ile belirlenebilir.

11. **Donanım Kısıtlamaları:** Bazı tasarım protokolleri yeri, düğüm donanımına gereksinim duyar ve bunun aksine bazı düğüm donanım yerleri protokol katmanları üzerinde kısıtlamalar getirir. Alıcı ile iletişim için yüksek güçteki bir verici buna bir örnek olabilir[32]. Bu kısıtlamaların, belirli mimarilerin tasarımı esnasında göz önünde bulundurulması gerekir.

### 3.2. YÖNLENDİRME PROTOKOLARININ KATEGORİSİ



#### 3.2.1. Ağ yapısı

Ağ yapısının temeli kablosuz algılayıcı ağların içindeki yönlendirme protokolü işlemlerinde önemli bir role sahiptir. Bu kategoriye giren protokoller de üç alt gruba ayrılabilir. Bunlar:

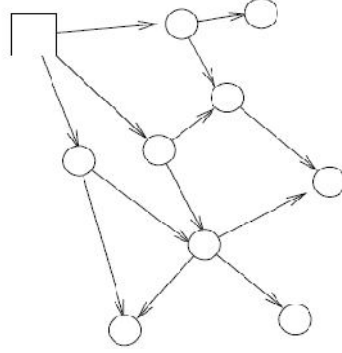
##### 3.2.1.1 Veri Merkezli Protokoller (Düz protokoller):

Bu tip yönlendirmede, kablosuz algılayıcı ağın tüm algılayıcı düğümleri ile eşit bir şekilde iş yaparlar ve genellikle yönlendirme prosedürlerinde eşit hizmet ve aynı göreve atanırlar. Düz yönlendirme, toplanan veriler içerisinde bir sorgu kullanır. Baz istasyonları belirli bölgelerdeki belirli düğümlerin bir grubuna sorgularını aktarır ve bu bölgede yer düğümlerden yanıt bekler. Bu protokoller aynı zamanda Öznitelik tabanlı protokoller olarak adlandırılırlar. Öznitelik-değeri sorgu süresince istenilen veriler esasında, veri özelliklerini belirtmek için önemlidir. Özniteliklerin seçimi WSN uygulanmasına bağlıdır.

##### *Flooding ve Gossiping Protokolleri:*

İki klasik veri aktarma mekanizması, her hangi bir algoritmalar ve topolojiler bakımının yönlendirilmesine gerek duymazlar [05]. Flooding protokolleri içerisinde, veri paketi alan her bir algılayıcı paket bölgesine ulaşıncaya dek komşu düğümlerinden gelenlerin

hepsini yayınlarlar. Bu süreç, başka bir düğümden olan verilerin kabul edilip edilmediğine bakılmaksızın devam eder. Bu protokol küçük ağlar için önerilir. WNS genellikle çok sayıda düğümden oluşur ve oluşturulan veri paketleri çok fazla olabilir aynı zamanda, bu protokolün WSN'yi aktif hala getirmemesi için büyük enerji kullanan ekstra düğümlere paketler transfer edilir. Flooding protokolu Şekil 3.1' de verilmiştir.

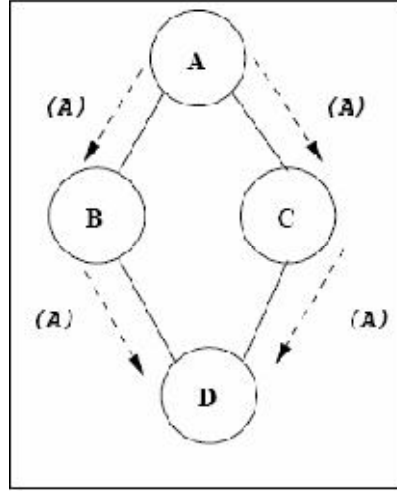


Şekil 3.1: Flooding işleyişi

Göçme sorununu çözmek için yeni bir protokol olan [05,40] 'Gossiping' önerilir.

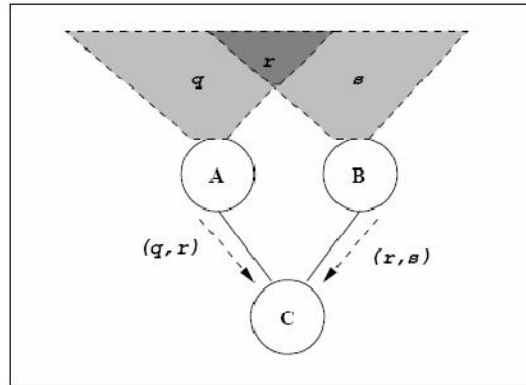
*Gossiping*, *Flooding* protokolünün gelişmiş bir versiyonudur. Süreç şu şekilde ilerler: Kaynak düğümü rastgele bir düğüm olan B varsayalım; komşusunun düğümüne seçim düğümü paketi gönderir. B düğümü de rastgele başka bir düğüm seçer ve ona aynı paket gönderir. Bu süreç hedef düğümüne tüm paketler ulaşmaya kadar devam eder. Böylece belirli bir düğümüne gönderilen bir paketin sadece bir kopyasını almak mümkündür ve bu patlama sorunundan kurtulmaya yol açar. Fakat gossiping operasyonunu bir ağdaki tüm sensörlere bir paket göndermek için çok fazla zamana neden olur.

- **Göçme** (Implosion) : Kopya mesajların aynı düğümlere göndermesi ile oluşur. Şekil 20 deki gibi A düğümü kendi komşuları olan B ve C düğümlerine veriyi gönderir. B ve C düğümleri ise kendilerine gelen veriyi komşuları olan D düğümüne gönderirler. Dolayısıyla D düğümüne aynı verinin birden fazla kopyası ulaşmış olur. Bu durum ise örtüşme olarak adlandırılır



Şekil 3.2: Implosion

- **Overlap** (Örtüşmek): Eğer iki düğüm aynı izleme bölgesinde ise, her iki düğüm aynı anda aynı işi yapıyor olabilir ve Sonuç olarak bir düğüm aynı ileti alır. Burada **A** ve **B** sensörleri aynı bölgeyi  $r$  paylaşırlar bundan dolayı, bu algılayıcılar verileri toplarlar ve her ikisi de topladıkları bilgileri komşuları olan **C** algılayıcısına iletirler.



Şekil 3.3 : Overlap problemi

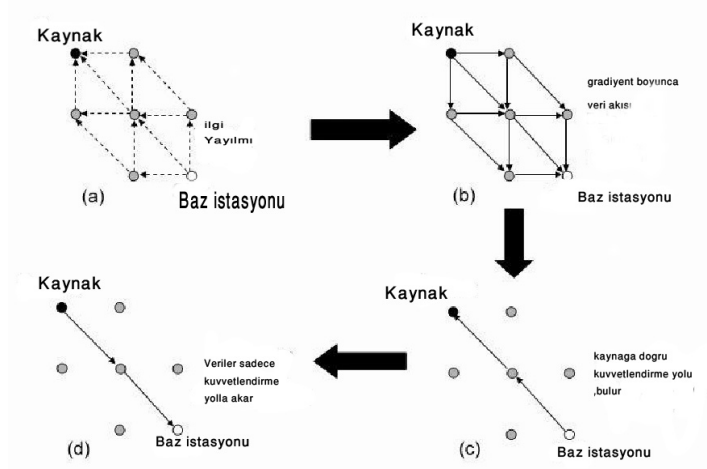
- **Resource blindness** ( Kaynak körlüğü) : Flooding düğümleri kendi enerji miktarını ve belli bir zaman içerisindeki komşularının enerji miktarı hakkında herhangi bir şey bilmiyorlardır. Bu nedenle, mevcut enerjiye göre faaliyetlerini organize edemezler.

### Directed Diffusion

Doğrudan Yayılma, algılayıcılar tarafından toplanan veriler için bir adlandırma şeması kullanarak algılayıcı düğümleri üzerinden veri yayılımı amaçlayan veri ve uygulama merkezli bir paradigmadır.

Doğrudan Yayılma, isimlendirilmiş veriler için algılayıcı ağ boyunca algılayıcı görevleri yaparak çalışır. [41]. Algılayıcılar talep edilen olayları algıladığında olay kaynağa doğru birçok yol üzerinden gönderilir. Depolanan eğimler, bir yön ve veri hızı vermek ve alınacak verileri belirlemek için kullanılır. İlk olarak alıcı ağdan düşük veri hızı talep eder. Algılayıcı verileri ağdan geri almaya başladığında, bir veya birden fazla yoldan daha yüksek veri hızında takviye yapar (Şekil 3.4). İlgili mesaj periyodik olarak güncel ilgiyi tutmak için sensörü yeni bir zaman damgası ile yeniler.

Directed Diffusion algoritması düğüm adresleme ya da küresel bir ağ topolojisinin bakımı gibi sorunları çözer, veri ön bellekleme aynı zamanda enerji tüketimini azaltır. Bölüm 4'te daha detaylı anlatılmıştır.



Şekil 3.4: Directed diffusion

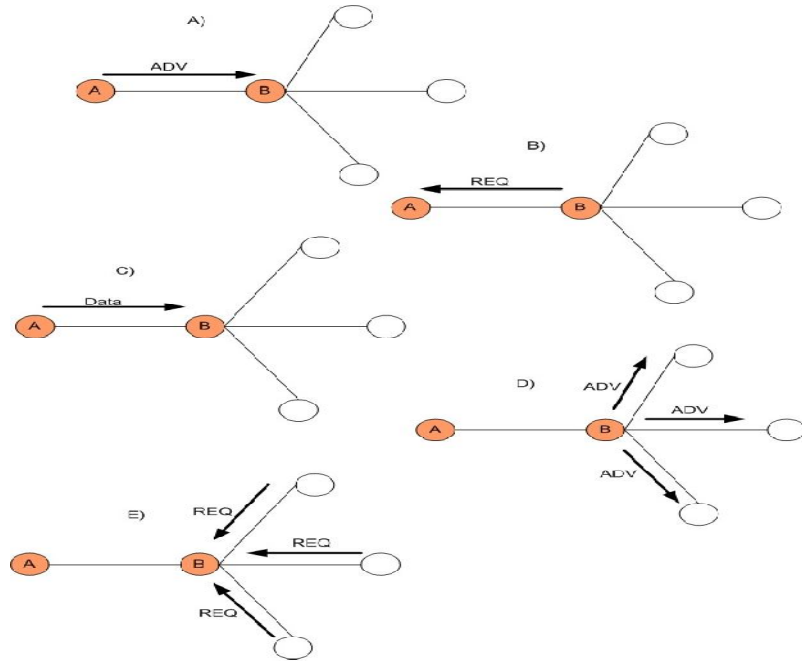
### SPIN (müzakere tabanlı bilgi için algılayıcı protokoller)

SPIN' in başlıca amacı, müzakere ve kaynak adaptasyon tekniklerindeki akımın eksikliğini kaldırmaktır [42]. SPIN, içeriye doğru patlamayı ve örtüşen sorunları çözer. SPIN, düğümlerin veri iletiminden önce birbiriyle müzakere etmelerini önerir. Bu

müzakere, gereksiz veri iletimini önleyerek enerji tüketimini azaltır. Bu şekilde, sadece yararlı bilgiler ağa teslim edilecektir. SPIN tarafından önerilen ana fikir verileri meta-veri adı verilen yüksek seviye tanımlayıcıları kullanılarak adlandırmaktır. Veri üreten bir kaynak, düğüm ağı üzerinden meta veri içeren bir reklam yayar. Bu meta-veriler iletimden önce algılayıcı düğümler arasında değiştirilir. Meta-veri tanımlayıcısı, SPIN algoritmasını yararlı boyutta uygulamak için gerçek verilerin boyutundan çok daha kısa olmalıdır.

Üç tip SPIN mesajı vardır, bunlar: ADV, REQ ve DATA'dır. ADV, yeni veri tanıtımı için, REQ veri istemek için kullanılır ve DATA gerçek mesajın kendisidir. ADV ve REQ mesajları sadece meta-veri içerir, bu yüzden de boyutları ufaktır.

İstenen veri DATA mesajı yoluyla yayılır. SPIN protokol örneği Şekil 3.5'te sunulmuştur.



Şekil 3.5 : SPIN algoritması

SPIN protokol ailesi aşağıdaki protokollerden oluşmaktadır:

SPIN 1: Yukarıda bahsedilen protokolün basit versiyonudur.

SPIN 2: Kaynaklarda eşik kullanan SPIN1'e bir uzantıdır.

SPIN-PP: Noktadan noktaya iletişim içindir.



SPIN-EC: SPIN-PP'ye benzer fakat enerji buluşsal yöntemleri eklenmiştir.

SPIN-BC: Yayın ağı için özeldir.

SPIN'de iki sorun görülmektedir;

- Alıcı ve verici arasındaki algılayıcı veri ile ilgili değilse veri hedefine ulaşamaz.
- İçeri doğru patlama problemi hala REQ safhasında hala mevcuttur.

SPIN ve güdümlü yayılım arasında bazı önemli farklılıklar vardır. Güdümlü yayılımda sorgular talep üzerine gönderilir, bildirilmezler. SPIN'de iş tam tersi uygulanır; algılayıcılar verileri bildirir ve ilgili düğümler bu veriye sorguyu yapar. İletişim komşudan komşuyadır, bu yüzden, veri toplama ve onbelleğe alma uygulanabilir. Son olarak, güdümlü yayılımda küresel ağ topolojisinin korumasına gerek yoktur.

### *Rumor yönlendirmesi*

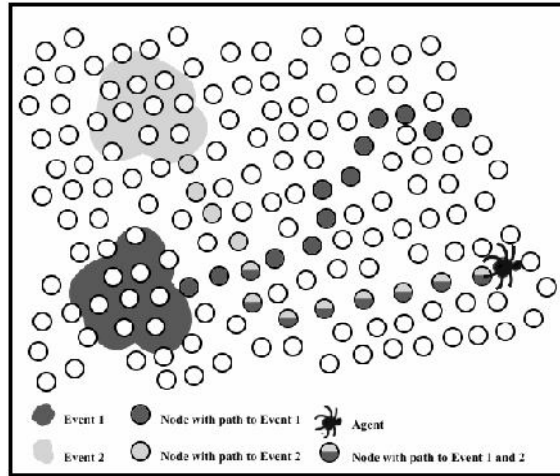
Rumor ile yönlendirme[43] Directed Diffusion için iyi bir alternatiftir. Nitekim birçok durumda değiştirilmesi gereken bilgi miktarı azdır. Güdümlü yayılım algoritmasında, talepler alıcılar tarafından kaynak düğümlere erimesi için iki boyutlu olarak (bir çok yoldan) gönderilir. Diğer taraftan, Rumor ile yönlendirme algoritmasında hem kaynaklar hem alıcılar tek boyutlu bir şekilde rasgele yürüyüşü takip ederek yayırlar. Bir kaynaktan yayılan bir eğri ona karşılık gelen bir alıcıdan gelen bir eğri ile karıştığında ikisi arasında bir yol kurulmuş olur. Rumor ile yönlendirme, sorgu flooding ve olay flooding semaları arasında mantıksal bir uzlaşma olarak kullanılabilir [43]. Rumor ile yönlendirme sorgu sayısı Rumor ile yönlendirme eğrisinin iki kesişim noktasının arasında ise, Rumor ile yönlendirmenin sorgu flooding ve olay flooding ile iki kesişim noktasının arasında ise verimli bir protokoldür.

Genel olarak bu algoritma sorgular ve olay bildirimleri veya gradyan üreten paketleri ile ağ akımından kaçınarak iletişim maliyet miktarını azaltır. Optimuma yakın yollar ve geniş bant aralığı gereksinimleri arasında birbirini dışlayan bir tercih vardır. Rumor ile yönlendirme, olaylar ve sorgulara dayanmaktadır. Düğümler bildikleri olayların listelerini tutarlar. Bir düğüm bir olaya tanıklık ettiğinde olay tablosuna olaya sıfır mesafede bir giriş yapar. Ayrıca belirlenen olasılığa göre; düğüm bir aracı yaratır. Rumor işlemi için Şekil 3.6'ya bkz.

Rumor yönlendirmesi protokolünün bazı dezavantajları vardır. İlk olarak, düzeltme algoritması araçların ve sorguların ağ içinde yayılmasını sağlamakta her zaman etkili değildir. Düğümlerin ve döngülerin yeniden ziyareti engelliyor olsa da, spiral bir düzeni takip edebilir. Böylece, aracı ve sorgu WSN içinde göreceli olarak ufak bir alan içinde kalabilir bu da, başarılı bir sorgu olasılığını düşürür. Ayrıca, büyük bir WSN söz konusu olduğunda, aracının ve sorgunun ziyaret edilen düğümler listesi her iletimde genirler. Sonuç olarak, bu bilgi çok büyük miktarda veri ihtiva eder, bu da her düğümün her ileti iletiminde büyük bir enerji tüketmesine böylece ağın ömrünü kısaltır.

Rumor'ın Directed Diffusion'dan temel farkları şunlardır:

- İlgi yayımı yoktur.
- Özel eylemleri izlemek için sorgu, düğümlere gönderilir.
- Uzun ömürlü paketler, ajanlar, olayların ağa doğru yayılması farklılıkları olarak göze çarpmaktadır.
- Düğüm bir eylemi sezdiğinde, eylemler tablosuna eylemi ekler ve bir ajan üretir.
- Ajanlar lokal eylemler hakkındaki bilgileri yaymak için ağda dolaşırlar.
- Ajanlar TTL (Time-to-Live) içerir.



Şekil 3.6 : Rumor Routing'te ajan kullanımı

#### *ACQUIRE (Algılayıcı Ağda Aktif Sorgu Yönlendirme)*

COUGAR'a benzer şekilde, ağ dağıtılmış bir veri tabanı olarak göz önünde bulundurulur. ACQUIRE [44]: Algoritma, tekrarlanmış veri için kompleks sorgulara yönelik tek seferlik olarak tasarlanmıştır. Flooding, Gossiping, SPIN, Gülümlü yayılım algoritmaları gibi flooding temelli mekanizmalar devamlı birikmiş sorgular için

uygundur, çünkü ilk flooding maliyeti telafi edilebilir ve kaynaktan alıcıya devamlı veri akışı halinde önemsiz hale gelebilir. Bu algoritma, tekrarlanmış veri için tes seferlik karmaşık sorgularda enerji bakımından verimli değildir.

Algoritma ağı dağıtılmış bir veri tabanı olarak görür. Sorgunun alıcıya gönderimi sırasında sorgu yanıtlarını alan her bir düğüm, kullandığı önbellek öncesi bilgilerini yanıtlar ve diğer düğümlere gönderir. Önbellek öncesi bilgiler düzenli olarak güncellenmektedir. Sorgu bir yol boyunca ağı iletirken, karmaşık sorgular basit alt sorgular için yanıtlanırlar. Sorgu tamamen çözüldükten sonra, bilgi alıcıya ters veya kullanılan kısa yola geri gönderilir. Eğer ağ için genel karmaşık sorgular varsa bu ağ için verimli olur.

#### *Gradyan tabanlı yönlendirme*

Gradyan tabanlı yönlendirme (GBR) [45] güdümlü yayılımın bir başka çeşididir. İlgili ağ boyunca yayılırken GBR çok sayıda atlamayı (hop) depolar. Her düğümün yüksekliği keşfedilebilmesi, o düğümden alıcıya minimum sayıdaki atlama olarak tanımlanır. Bir düğüm yüksekliği ve komşunun yüksekliği arasındaki fark ağ üzerindeki gradyan olarak kabul edilir. Bir paket en büyük gradyanlı bağlantı üzerine iletilir. Aynı gradyan ile iki veya daha fazla atlama olduğunda, düğüm rastgele birini seçer. Bir düğümün enerjisi belirli bir eşik altına düştüğünde, düğüm yüksekliğini artırır ve bunun sonucunda, diğer algılayıcılar bu düğüme gönderilen verilerden vazgeçirilir; Bu, enerji verimliliğinin ve ağın yaşam süresinin artmasına izin verir. Çok sayıda yol için bir role olarak hareket eden düğümler çoklu komşulardan iletilen verinin alınmasından önce veri yığını temsil ederler. GBR, iletişim enerjisi verimliliğinin terimlerinde güdümlü yayılımı daha iyi canlandırır.

#### *Minimum Maliyet ile İletim Algoritması (Minimum Cost Forwarding Algorithm)*

MMIA protokolü, ağ içindeki düğümler arasında yollar kurmak için gecikme, üretilen iş ve enerji tüketimi karakteristiklerini birleştirir [46]. Protokol, her bağlantı için, düğümün gecikme, üretilen iş ve enerji tüketimini tutan bir maliyet fonksiyonunu atar. Maliyet fonksiyonu göre, her bir düğümdeki MMIA bir maliyet alanını belirler. Sonuç olarak, paketler en düşük maliyet ile bir sonraki atlamayı belirleyen maliyet alanına doğru akarlar. Minimum maliyet yolu algoritmasının iki aşaması bulunur. Bunlar:

maliyet alanı kurulması ve maliyet yol iletilmesidir. Maliyet alanı herhangi bir düğüm ve alıcı arasındaki minimum maliyeti belirlemek için aşamaların arasını kurar. Alıcı 0 başlangıç maliyeti ile bir reklam (ADV) mesajı yayınlar. Sonra, ADV mesajı maliyeti güncelleyerek iletilir. İ düğümünden ADV mesajı alan her bir J düğümü maliyetini hesaplar.

$Li + C_j, i (7,6)$

Düğüm 'i' 'Li' maliyeti olduğu yerde(= 0 hedef için) , düğüm 'j' den düğüm 'i' ye olan maliyet '  $C_j, i$  ' dir. Daha sonra her bir düğüm, orantılı maliyeti için düğüm i,  $C_j, i$ 'ye bir geri çekilme zamanlayıcısı yerleştirir ve ADV mesajı yayınlar. Geri çekilme zamanlayıcısı, düğümüne minimum maliyetli alıcı seçimi için maliyetini güncellemeye yardım eder.

*EAR( Enerji farkındalığı olan yönlendirim)*

EAR, WSN'nin yaşam süresinin uzaması için olan başka yönlendirme protokolüdür [47]. EAR mümkün olduğunca fazla enerji tasarrufu amaçlayan bir protokoldür. Bu duyarlı bir protokoldür ve hedef önceliklidir. Bu protokolün temelindeki fikir, tüm ağın yaşam süresinin arttırılması gibi belirli bir olasılık ile kaynak düğüm ve veri alıcısı arasında farklı zamanlarda çoklu yolları kullanmaktır. Sonuç olarak her bir yolun enerji daha uzun süreli olacaktır. Bunu başarmak için, bu protokol güdümlü yayılım(directed diffusion) için bir optimal kıyaslama yerine bir çok yolu muhafaza eder. Enerji tüketiminin hesaplanan olasılığına dâhil olan yollar seçilir. Aslında ağın ömrü sadece bu protokol ve bu konularda bir ölçüdür. Düğümlerin her bir çifti arasındaki bütün yönleri keşfetmek için yerleştirilen flooding ler ile başlar, bir yönlendirme tablosu oluşturur ve her bir yönün maliyetini bulur. Daha sonra yüksek maliyetli yolları çıkartır. Bu protokolün önemli bir dezavantajı güdümlü yayılımdan daha fazla zaman alan kurulum aşamasıdır. Protokol, her algılayıcı düğümün, düğümlerin şekillerini ve yerlerini içeren birkaç adresleme şeması ile adreslenebilmesini varsaymaktadır.

Düz yönlendirme protokollerinin avantajları:

Topolojik değişiklikler komşu düğümler ile sınırlıdır. Her düğümdeki yerel ağ bilgisini ve aslında her düğüm fonksiyonu eşit olarak protoçöl ölçeklenebilirliğini ve bilişimsel kolaylığını yükseltir. Karmaşık hesaplamaları belirlemek ya da hiyerarşik bir ağ

topolojisini yönetmeye gerek yoktur. Düz ağ topolojileri, her bir düğüme homojen düğümleri kullanmak için olan dağıtım stratejilerini basitleştirmek için eşit davranır. Belirlenen daha yüksek işlevsellik ve kaynak kapasiteleri ile belirli düğümleri dağılımına ve dahil olması gerek yoktur.

Düz yönlendirme protokollerinin dezavantajları:

Alıcı düğümü bir alıcı düğümü çevreleyen kaynak düğümünün yüksek yoğunluğu ile ağlar içindeki mesaj teslimatında sürenin gecikmesine neden olabilir veya aşırı yüklenme yapabilir. Yüksek hacimli veri mesajının alıcı komşu düğüme çapraz geçişi geçiş lekelerine ve güç kaynağının erkenden başarısız olmasına neden olabilir. Homojen düğümlerin çalıştırılmasında tek kritik düğüm çalışmadığı takdirde düşük yoğunluktaki ağlar ayrılmaya ve ağın parçalarının izolasyonuna sebep olabilir. Bu durumda önerilen ilave alıcı düğüm eklemek olacaktır.

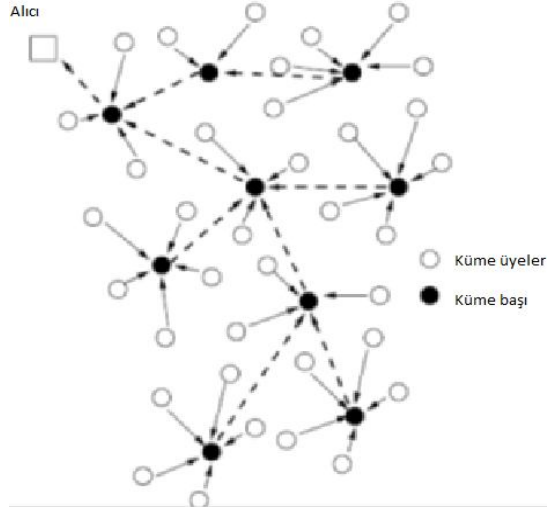
### 3.2.1.2. Hiyerarşik Tabanlı Protokoller

Hiyerarşik yönlendirmenin düz yönlendirmeden farkı ağ içinde statik veya dinamik olabilen farklı rollere sahip olmasıdır. Bu tip protokoller enerji verimliliği, etkin iletişim ve WSN ölçeklenebilirliği ihtiyacını sağlamak için tavsiye edilirler. Düğümler, enerji tüketimini azaltmak için kümeler oluşturmak üzere gruplandırılmıştır. Her küme içinde bir lider veya baş küme seçilir ki düğümler genellikle seçilen bir baş kümenin arta kalan enerjisine sahiptir. Düşük enerji düğümleri amaçlanan nesne içinde ki algılamayı çalıştırmak için kullanılabilir iken, hiyerarşik protokollerde yüksek enerji düğümleri veri işleme ve veri aktarımı için rastgele seçilir. Buna, küme tabanlı yönlendirme de denir. Hiyerarşik yönlendirme iki adımda düzenlenmiştir. Bunlardan birincisi yönlendirme paketleri için baş kümenin seçimi ve diğer adımlardır. Bu bölümde, LEACH, TEEN & APTEEN, PEGASIS gibi bazı protokolleri inceleyeceğiz.

#### *TEEN (Enerji Etkinliği Duyarlı Algılayıcı Ağ Sensörü Protokol Eşiği)*

TEEN, reaktif bir şekilde çalışan ağ içinde kritik zaman uygulamaları için faydalı hiyerarşik bir protokoldür.[48] Yakın düğümler farklı düzeylerde kümelenirler ve küme başlarını seçerler. Her küme başı verilerin alıcıya doğrudan gönderimi ile sorumludur. Kümeler oluştuktan sonra, küme başı düğümler için iki eşik yayınlar. Algılanmış

öznitelikler için yumuşak ve sert eşikler vardır. Sert eşik, bir algılayıcı düğümün iletecisini veya küme başına olan iletimi değiştirmesini tetiklemek için bir özneliğin minimum olasılıklı değeridir. Şekil 3.8’de böylece ilgi sırasındaki öznelik algılandığında, sert eşik düğümlere yalnızca iletim için izin verir böylece iletimlerin sayıları önemli bir derecede azalır. Bir düğüm sert eşikte veya ötesindeki değeri algılar, bu özneliğin değeri eşit bir şekilde veya yumuşak eşikten daha büyük olacak şekilde değiştiğinde veriyi sadece iletir. Sonuç olarak, algılanan öznelik değeri içinde değişim yok ise veya biraz değişim var ise, yumuşak eşik ileride iletim sayısını daha da düşürecektir. Paket iletim sayısını kontrol etmek için hem sert hem de yumuşak eşik değerlerine ayarlama yapılabilir. Ancak eşikler ulaştırılmadıysa, kullanıcı herhangi bir bilgi elde edemediğinde periyodik raporların olduğu yer için TEEN iyi bir uygulama değildir.



Şekil 3.7: TEEN

*APTEEN*(Enerji Etkili Algılayıcı Ağ Protokolüne Duyarlı Uyarlanmış Eşik )

Protokol hem kritik zamanlı olayları hem de periyodik veri toplamayı elde etmek amacıyla olan TEEN'in gelişmiş bir protokolüdür[49]. Ağ yapısı TEEN ile aynıdır. Kümeler ve küme başları oluştuktan sonra aşağıdaki parametreleri yayınlar:

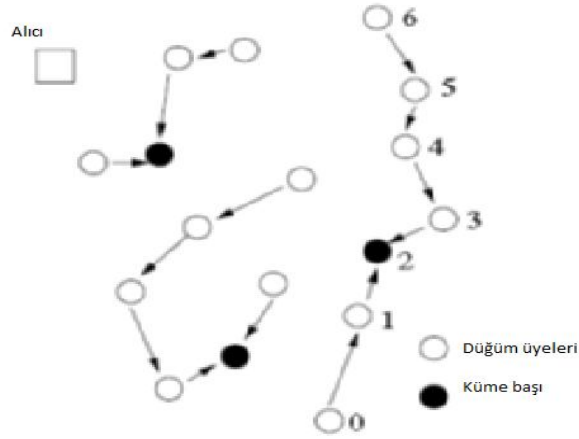
- Özellikleri: Kullanıcın almak istediği veriler için olan fiziksel parametrelerdir.
- Eşikler: TEEN’de sunulan sert eşik ve yumuşak eşiklerdir.

- Zaman çizelgesi : Her düğüm için bir zaman aralığı atamayan TDMA tekniğini takip eden bir zaman çizelgesidir.
- Sayım zamanı: düğüm olan iki başarılı veri iletimi arasındaki zamandır.Eğer düğüm veriyi sayım zamanı ile eşit zamanda göndermez ise, bunu algılamalı ve tekrar iletmelidir.

Küme başı aynı zamanda veri iletiminin tükettiği enerji boyutunu azaltmak için veri toplamaktan da sorumludur. Enerji kaybı ve ağın yaşam süresine göre, TEEN, iletim sayısının azalmasından dolayı Leach ve APTEEN dan daha iyi bir performans gösterir. TEEN ve APTEEN'in başlıca sakıncaları,çoklu seviyelerde kümelerin oluşumunun karmaşıklığı, eşik tabanlı fonksiyonların uygulanması ve oznitelik tabanlı yayılım adlandırmalarıyla ilgili uğraşlardır.

*PEGASİS (Algılayıcı Bilgi Sistemlerinde Güç ve Verimin Bir araya gelmesi)*

PEGASİS protokolü Leach protokolünde iyileştirme sağlamayı amaçlamaktadır. [50] . PEGASİS Şekil 3.9 daki şekliyle, kümeler yerine düğüm zincirleri oluşturarak LEACH küme oluşumunun yol açtığı ek yükü adreslemeyi amaçlamaktadır. Zincir içinde düğümlerin kendilerine en yakın bir sonraki atlama gibi komşularını seçtikleri yerde, ağgözlü bir algoritmaya göre zincir yapı oluşturulur. Zincir yapısının düğümlerin alıcıdan en uzak olduğu noktada başladığı ve düğümlerin global ağ bilgisine sahip olduğu varsayılır. Zincir işlemin bir sonucu olarak, küme oluşumu ve üyelik yerine, zincir içindeki her düğüm kendinden bir önceki ve bir sonraki komşunun izinden gidecektir. Zincir içindeki iletişim, zincir lideri gibi bir algılayıcı düğümde olduğu gibi tüm veriler toplanıncaya kadar, zincir içindeki bir komşudan veri toplayan bir zincir içindeki her düğümün sıralı olarak oluşturulmasıdır. Zincir lideri düğümler arasında bir simgeden geçerek iletişim siparişlerini kontrol eder.



Şekil 3. 8: PEGASIS

Hiyerarşik yönlendirme protokollerinin avantajları:

Genel olarak, hiyerarşik protokoller veri toplama ve veri birleşmesinin kullanımını desteklemektedir. Bu, yinelenen mesajları ortadan kaldırmak için bazı ek işlevler içerebilir. Veri toplama, ağın ek yükünü ve iletilen mesajın miktarını, dolayısıyla da enerji tüketimini azaltır.

Bazı hiyerarşik ağ topolojileri homojen düğüm kullanmak için dağıtım stratejilerini kolaylaştırır. Daha yüksek olarak belirtilen işlevsellik ve kaynak kapasiteleri içeren belirli düğümlerin dağılımına gerek yoktur.

Kümelendirme yaklaşımları içinde tüketilen enerji kümeleri kurmak için olandan, tüm ağda kullanılanlara nazaran çok daha düşüktür. Güç tüketimi küme için belirlenir. Ayrıca veri mesajları küme başına ulaşmak için çok daha az düğüm içinden geçer. Genel hiyerarşik yönlendirme protokolleri bir bütün olarak ağdaki enerjiyi çok daha az tüketir.

Hiyerarşik yönlendirme protokollerinin dezavantajları

Bazı hiyerarşik ağ topolojilerin içerik ve dağıtımını özel düğümlerin belirtilen işlevlerinin çok daha yüksek olmasını ve kaynak yeteneklerini gerektirir. Bu hem dağıtım hem de üretim açısından ağ maliyetini etkileyecektir. Farklı işlevsellikteki düğümlerin kullanımını küme başlarının çevrimini zorlaştıracaktır.



Bu dezavantajı aşmak için bazı hiyerarşik protokollere homojen düğüm kullanımı önerilmektedir. Küme başları yönlendirme stratejilerinde alıcı düğümle doğrudan iletişim kurmak için gereklidir ya da uzun mesafede önemli ölçüde daha fazla enerji tüketecektir. Bu küme başı sıcak noktaların üstesinden gelmek için, protokol uygulamalarında dinamik küme başı seçimi ve rotasyon kullanır.

Küme başlarını homojen ağda rastgele hale getirmek için eklenen ek yük fazladan hesap ve karmaşık getirecektir Algılayıcı düğümler tarafından ek bilginin toplanıp, kuruluma ve ağın başlatılmasına iletilmesi gerekir böylece başlangıç için fazladan ek yük ve mesaj iletimi eklenmiş olur. Ağın başlatılması düz ağ topolojilerinden çok daha fazla enerji tüketiyor olabilir. Küme başlarının seçimi protokol çalışması ile belirlenir. Küme başlarının ölçeklenebilirliği artan miktardaki ağ boyutları ve artan gereksinimi olumsuz etkilemiş olabilir.

### 3.2.1.3. Konum Tabanlı(Location-based)

Konum tabanlı veya coğrafi yönlendirme protokolü adı verilen yöntem kablosuz sensor ağlarda kullanılabilir. Bu ağlarda, sensor düğümleri kendi yerlerini farklı çeşit yerleştirme algoritmalarıyla belirleyebilmektedirler. Komşu sensor düğümlerinin koordinatları komşu düğümler arasındaki bilgi alışverişi ile belirlenmektedir. Bu süreçte sensor düğümleri topoloji bağlanabilirlik bilgisi kullanmak yerine, coğrafi bilgileri kullanmaktadır. Gönderici hem kendi yerini hem de gönderinin ulaşacağı yeri bilmelidir. Düğümlerin yeri gömülü bir düğüm kartına entegre edilmiş bir GPS (Küresel Konumlama Sistemi) sisteminden direkt olarak veya özellikle kablosuz sensör ağlarda kullanılmak üzere geliştirilmiş bir yerleştirme protokolü ile belirtilebilir. Düğümler arasındaki mesafe düğümlerden elde edilen sinyal şiddetinin hesaplanmasıyla ve koordinatlar ise komşu düğümler arasında bilgi alışverişi yapılarak belirlenmektedir. Konum tabanlı yönlendirme protokolleri enerjinin en verimli şekilde kullanılması için çeşitli veri yönlendirme teknikleri sağlamaktadır:

#### *GAF*

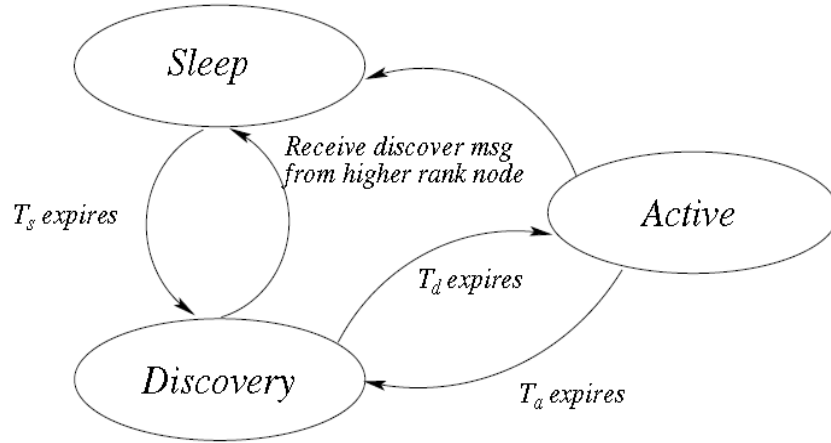
Ağdaki enerji tüketimini, ağdaki konumlarına göre seçilen destek düğümler kullanarak azaltmayı amaçlar. Ana amaç boş dinleme(idle listening)nedeniyle oluşan enerji israfını, ağın bağlanılabilirliğini sürdürecektir az sayıda destek düğüm kullanarak azaltmaktır[51] .

Geographic Adaptive Fidelity (GAF) protokolü öncelikle gezgin düğümlere sahip ağlar için geliştirilmiştir, sabit düğümler için de uygulanabilmektedir [51]. Algoritma bütün ağın alanını Şekil 3.10'da gösterildiği şekilde  $r$  büyüklüğünde bir sanal grid'e ayırır. Her düğüm GPS ya da diğer konumlama sistemleri ile elde ettiği kendi konum bilgisini kullanır ve içinde bulunduğu gridle kendini bağdaştırır. Her grid içerisinde herhangi bir anda yalnızca tek bir cihaz yönlendirme düğümü olarak görev yapar. Bu düğüm diğer tüm düğümler uyku moduna geçebilirken, ana istasyona veri aktarmakla görevlidir. Bu şekilde ağ ömründe, özellikle de yoğun ağlarda, bir uzama görülür. Dahası, GAF iki bitişik A ve B hücresi için, A'daki tüm düğümlerin B içerisindeki düğümlerle maksimum sinyal aralığında çalışabileceğini ve aynısının B içindeki düğümlere bakıldığında A için geçerli olduğunu öne sürmektedir (Şekil 3.10). Bu şekilde, hücre içerisindeki düğümler eşdeğerdir. Grid ve hücre büyüklükleri önceden belirlenmiş olabilir. Bu her düğümün (kendi konumunu bildiğini varsayarsak) ait olduğu hücreyi belirlemesini sağlar. Bu ağdaki düğümlerin çoğunun (sınır hücrelerdeki düğümler dışında) dört yanda da komşu düğümleri olacağı anlamına gelir. Grid büyüklüğü grid içerisindeki her düğümün bitişik gridlerdeki her düğümlerle iletişime geçebileceği şekilde belirlenir. Bu zorunluluk,  $r$  grid büyüklüğü ile  $R$  düğümünün iletişim aralığı arasında şöyle bir ilişkiye neden olmaktadır:

$$r \leq R \sqrt{5}$$

Şekil 3.11'de de görülebileceği gibi, GAF'ın durum değiştirme diyagramı üç durumdan oluşmaktadır, *keşif*, *aktif* ve *uyku*.

- *Keşif (Keşif düğümleri)*: bölgedeki komşu düğümler keşfetmek için kullanılır.
- *Aktif (Aktif düğümler)*: düğümün yönlendirmeye katılıp katılmadığını gösterir.
- *Uyku (Uyuyan düğümler)*: sinyal kapalıdır.

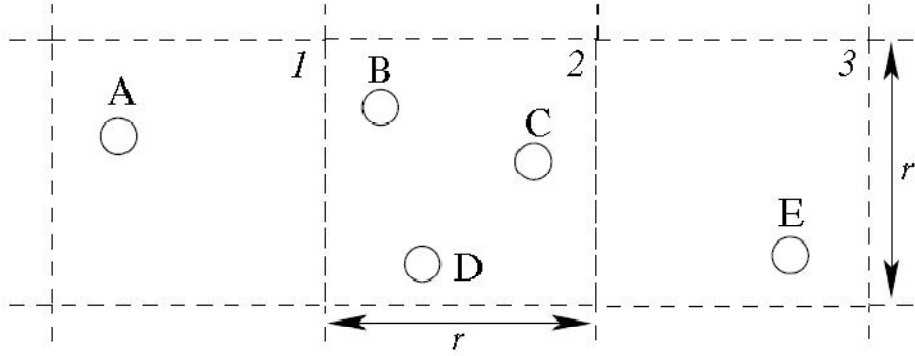


Şekil 3. 9: GAF algoritmanın çalışma durumları[51]

Bir sensör uyku moduna geçtiğinde enerji tasarrufu yapmak için sinyalini kapatır. Keşif modunda, sensör aynı gridda bulunan diğer sensörler hakkında bilgi edinmek için keşif mesajları gönderir ve alır. Aktif moddayken bile sensörler belli aralıklarla eşdeğer sensörleri durumları hakkında bilgilendirmek için keşif mesajları göndermektedir. Bu hallerin her biri için harcanacak süre ihtiyaçlara ve sensör gezginliğine bağlı olarak uygulama tarafından ayarlanabilmektedir. GAF, her gridin sensör derecelendirme kurallarına göre yalnızca bir aktif sensöre sahip olmasını sağlayarak ağ ömrünü uzatmayı hedefler. Sensörlerin derecelendirmesi kalan enerji seviyelerine göre belirlenmektedir. Bu nedenle daha yüksek dereceli bir sensör ilgili gridlerle yönlendirme işlemlerini yürütebilecektir. Örneğin, aktif durumdaki bir sensör, keşif durumundaki bir sensörden daha yüksek bir dereceye sahiptir. Daha uzun bir ömür beklentisi olan bir sensör, daha yüksek bir derece alacaktır.

Herhangi bir sensörü dışarıda bırakmadan bütün sensörleri olabildiğince uzun süre çalışır durumda tutabilmek için GAF sensörlerin sadece uyku durumuna geçmeden önce bir süre aktif durumda kalmalarına dayanan bir yük optimizasyon stratejisi izlemektedir. Bu aynı gridda bulunan diğer sensörlere de aktif duruma gelme ve yönlendirme işlerini yürütme olanağı tanımaktadır. Bu kuralın arkasındaki mantık ise, keşif durumuna geçen sensörlerin, enerji tasarrufunun gerçekleştiği uyku durumundaki komşu sensörlerden daha az enerjileri kalmasıdır. Sensör gezginliğinin bir gridin aktif sensörsüz kalmasına neden olabileceği unutulmamalıdır. Bu sorunu çözmek için, sensör gridi terk edeceği süreyi GPS alıcısı yardımıyla hesaplayarak keşif mesajı içerisinde yayınlamaktadır.

Bu keşif mesajı alındığında, sensörün komşuları uyku durumuna ne zaman geçeceklerini ayarlayarak grid içerisinde yönlendirmeyi yürütecek aktif bir sensörün olmasını sağlarlar.



Şekil 3.10: GAF Protokolde sanal ızgara yapısı

### GEAR

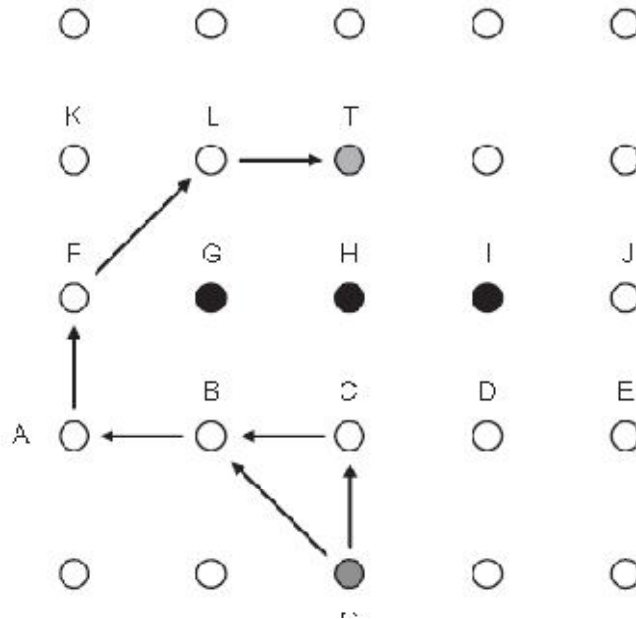
*Yu ve diğerleri.* coğrafi ve enerji farkındalıklı yönlendirme (GEAR) protokolünü önermişlerdir [52]. GEAR, bir paketin ulaşacağı bölgeye enerji farkındalıklı ve coğrafi olarak bilgilendirilmiş komşu seçici bulgulayıcılar aracılığıyla yönlendirilmesini içerir. GEAR, yönlendirilmiş dağıtımlarda uyandırılan bölgeleri, sadece bütün bir ağı uyandırmak yerine belli bir bölgeyi göz önüne alarak kısıtlamayı amaçlar. Bu şekilde GEAR, yönlendirilmiş dağıtımdan daha çok enerji tasarrufu sağlayabilmektedir. Her düğüm komşuları üzerinden hedef noktaya ulaşmanın bedelini öğrenmiştir. Tahmini bedel düğümün kalan enerjisi ile hedef noktaya olan uzaklığın bir kombinasyonudur. GEAR'daki her bir düğüm komşu düğümler üzerinden hedef noktaya ulaşmanın tahmini ve öğrenilmiş bedelinin farkındadır. Tahmini bedel düğümün kalan enerjisi ile hedef noktaya olan uzaklığın bir kombinasyonudur. Öğrenilmiş bedel ise tahmini bedelin ağdaki boşluklar hesaba katılarak yönlendirme için yeniden düzenlenmiş halidir. Boşluklar, bir düğümün hedef bölgeye kendinden daha yakın bir düğümü olmaması sonucunda oluşur. Eğer ağda boşluk yoksa tahmini bedel, öğrenilmiş bedele eşit olur.

GEAR protokolü her kullanılan  $x$  düğümü için şu adımları izler :

- Eğer  $x$  hedef nokta  $d$ 'ye hem öklidyen mesafe hem de öğrenilmiş bedel açısından kendinden daha yakın bir komşulara sahipse, bu komşular arasında en düşük öğrenilmiş bedele sahip olanı seçer ve paketi bu komşuya iletir.

- Eğer yoksa paketi en düşük öğrenilmiş bedele sahip komşuya iletir. Bu durum bir düğümün hedef bölgeye kendinden daha yakın komşu düğüme sahip olmadığı hallerde boşluk oluşmasıyla görülür.

GEAR iki ana aşamadan oluşur. Paketin hedef bölgesine iletilmesi (1. aşama) ve paketin hedef bölgede dağıtımını (2. aşama). 1. aşama sırasında, sensör , hedef bölgeye kendinden daha yakın olan bir komşu düğüm seçerek paketi başka bir sensöre iletmesi için ona iletir. Aksi durumlarda, eğer hedef noktaya kendinden daha yakın bir komşusu yoksa paket sahibi sensör ile hedef bölge arasında bir boşluk oluşur ,şekil 3.12’ye bkz. Bu durumda GEAR öğrenilmiş bedeli minimum olan komşulardan birini seçer. 2. aşamada GEAR gerid önuşümlü coğrafi iletim algoritması kullanarak paketi hedef bölge içerisinde dağıtır. Bu durumda hedef bölge dört alt bölgeye ayrılmıştır. Paket sahibi sensör paketin 4 kopyasını yaratarak bu alt bölgelere her bölgeye bir paket gelecek şekilde dağıtılmasını sağlar. Bu ayırma ve iletim işlemi düğüm alt bölgede tek kalana kadar devam eder ve paket bırakılır. Sensörlerin seyrek olarak bulunduğu bölgelerde GEAR kısıtlanmış paket yollama kullanarak geri dönüşümlü coğrafi iletime göre daha çok enerji tasarrufu sağlar. Bu durumda sensör tüm komşularına sadece bir mesaj yayınlar.



Şekil 3.11: GEAR Yönlendirme

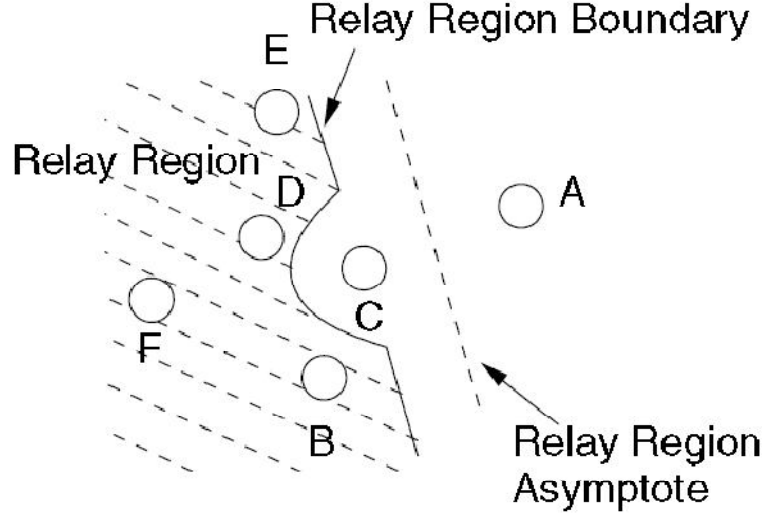
### *MECN*

Minimum enerji iletişim ağı her düğüm için bir aktarım alanı belirler[53]. Düğümlerin aktarım alanları, aktarımın bu düğümler üzerinden gerçekleşmesinin direkt aktarıma göre daha fazla enerji tasarrufu sağlayacağı çevre bölgelerden oluşur. Bir düğümün çevreleme alanı, o düğümün ulaşabileceği tüm aktarım alanları tarafından oluşturulmuş olur.

MECN'in temel prensibi, daha az düğüme sahip olan bir alt ağ bularak, iki düğüm arasındaki aktarımın gerçekleşmesi için daha az enerjiye ihtiyaç duymaktır. Bu şekilde küresel minimum güç yolları, ağdaki tüm düğümler göz önüne alınmak zorunda olmadan bulunmuş olur. Bu, her düğümün aktarım bölgesi için yerel bir arama yapmasıyla gerçekleştirilir.

Düzlem üzerindeki sensörlerin pozisyonlarına dayanmaktadır ve iki ana aşamadan oluşmaktadır. Bu aşamalar, çevreleme grafiği oluşturulması ve bedel dağılımıdır. Bir destek ağı için ilk aşamada (çevreleme grafiği oluşturulması), MECN, sensörlerin yakın çevrelerine dayalı, kapsama grafiği (enclosure graph) adı verilen seyrek bir grafik oluşturur. Bu amaçla, sensör öncelikle direkt olarak iletişim kurabileceği sensörlere göre aktarma alanını belirler. Aktarma alanı, bir mesajı aktarmanın aracı bir sensörle (ya da aktarma sensörü ile) aktarılmasının bu noktalara direkt aktarmadan daha fazla enerji tasarrufu sağlayacağı noktalardır. Bu sensör etrafında, kapsama bölgesi (enclosure region) denilen bir bölge yaratır. Bu bölgenin dışında daha fazla komşu aramak artık enerji tasarrufu sağlamamaktadır. Bir sensörün çevreleme bölgesi içindeki sensörler enerji tasarrufu sağlayan bir aktarım için bu sensörün iletişim kuracağı komşularıdır. Görülebileceği gibi, bir sensörün çevreleme alanı, direkt olarak iletişime geçebileceği sensörlerin aktarım alanlarının kesişimine bağlıdır. Çevreleme grafiği, tüm sensörleri tepe noktaları olarak kabul eden ve köşesi çevreleme bölgeleri içerisinde yer alan sensörlerin ve komşularının arasındaki tüm köşelerin birliğinden oluşan bir yönlendirilmiş grafiğdir. Başka bir deyişle, bir sensör kendi aktarım bölgesindeki sensörleri algıladığı veri için potansiyel iletim adayları olarak görmeyecektir. Şekil 3.13'te yayın-aktarım çifti oluşturan sensörlerin aktarım alanını göstermektedir. Sadece yakın alandaki (örn. Çevreleme alanı) sensörler potansiyel iletim adayları olarak görülmektedir. Dahası, bu grafik seyrek ve kuvvetli bağlardan oluşmaktadır. İkinci aşamada (bedel dağılımı) çevreleme grafiğinin optimal olmayan bağlantıları elenerek, minimum güç topolojisi grafiği oluşturulur. Bu grafik, her sensörden havuza doğru

yönlendirilmiş bir yol içerir ve sensörlerle havuz arasındaki tüm grafiklerden en az güç gerektirenidir. Bedel ölçümünde güç kullanımını temel alan Bellman–Ford en kısa yol algoritması çevreleme grafiğinde en uygun bağlantıların bulunması için kullanılır.



Şekil 3.12 : Yayın-aktarım MECN[53]

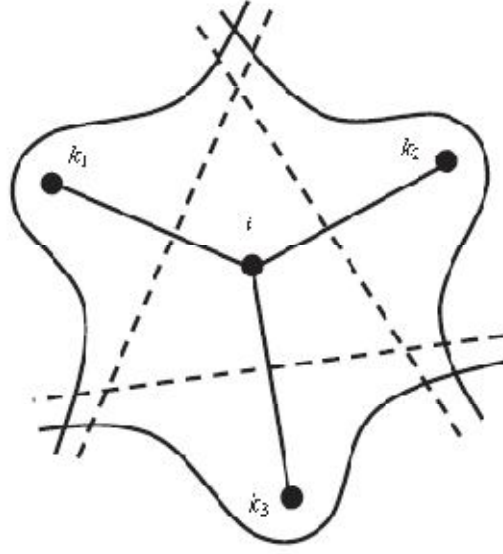
Her sensör kendi bedelini komşularına yayımlar. Bir düğümün bedeli, bu sensörün havuza yönlendirilmiş bir yolda harcayacağı minimum güçtür. Öncelikle, sensör şu formülü uygular:

$$C_{i,n} = \text{Cost}(n) + P_{tx}(i,n) + P_{rx}(n),$$

Burada,  $P_{tx}(i,n)$   $i$  'den  $n$ 'ye aktarım yapmak için gereken güç,  $P_{rx}(n)$   $n$  'in yaygın sensörlerinden gelen bilgileri alması için gereken güç, ve  $\text{Cost}(n)$  ise komşu  $n$  (i.e.,  $n$  sensörü  $i$ 'nin komşusudur) tarafından hesaplanan bedeldir. Sonra,  $i$  sensörünün bedeli şu şekilde hesaplanır:

$$\text{Cost}(i) = \min_{n \in N(i)} C_{i,n}$$

Burada,  $N(i)$   $i$  'ye çevreleme bölgesi konseptine uygun olarak komşu olan komşular takımıdır. Şekil 3.14 bunu açıklamaktadır. Bu aşamanın sonunda, tüm sensörler minimum bedelli komşu bağlantılarını hesaplamış olacaktır. Bu bağlantı algılanan verilerin havuza gönderilmesinde kullanılacaktır. Bütün bağlantılar minimum güç topolojisini oluşturmaktadır.



Şekil 3.13: Sensor i kapsamı

#### *Minimum Enerjili Küçük İletişim Ağları(SMECN)*

Minimum Enerjili Küçük İletişim Ağları (SMECN) protokolü Li Li and Joseph Halpern tarafından geliştirilmiştir[54]. SMECN bir yönlendirme protokolündense bir topoloji kontrol algoritmasıdır. Değişken iletim gücü göz önüne alınmıştır, yani bir düğüm iletim alanını çoğaltıp azaltabilmektedir. Bu özellik iletim alanlarının dar olduğu durumlarda enerji tasarrufu için kullanılabilir. SMECN'nin amacı, SMECN tarafından keşfedilmiş topolojiye uygulanacak bir yönlendirme protokolüyle her düğüm için etkili bir güç düzeyi belirlemektir.

Algoritma tüm sensörlerin maximum güçte iletim yapmalarıyla başlar. Bu düzeyde ağ en yüksek derecede bağlıdır fakat aynı zamanda en fazla enerjiyi de kullanmaktadır. Protokol, iletim gücünün düşürüldüğü ve yüksek enerji bağlantılarının kaldırıldığı bir keşif sürecine girer. Bu sürecin sonunda, ağdaki her bir düğüm kendine has bir enerji düzeyinde iletim yapmaktadır. Gözden geçirilmiş topoloji herhangi iki düğüm arasında minimum enerji gerektiren yolları içerir.

SMECN herhangi iki düğüm arasında birden fazla yol garantilemez. Yalnızca iki düğüm arasındaki minimum enerji yolunun kalacağı kesindir. Eğer minimum enerji yolunda bir düğüm arızalanırsa, bazı düğümler bağlantıyı kaybedebilir. Bu nedenle, sensörler başarısız olduğunda topoloji gözden geçirilmek zorunda kalacaktır ki bu da pahalı bir işler.



### 3.2.2 Yönlendirme Keşfi protokolleri

#### 3.2.2.1 Proaktif Protokol

Bu protokolda daha ihtiyaç duyulmadan bütün yollar hesaplanır. Bu çeşit protokoller düz ve hiyerarşik protokoller için kullanılmaktadır. Bu yöntemle ilgili sorun ise, yolun kullanıma uygun olmaktan çıkabilmesidir. Ağ topolojisindeki bu değişim, bütün ağ içinde yayılmak zorundadır. Bu daha fazla mesaj dolayısıyla da daha fazla güç tüketimi anlamına gelir. Her yol yönlendirme çizelgesinde bir yere sahip ve olası ağ büyüklüğü binlerdeyken, bu büyüklükte yeterli kaynak (hafıza) çizelgeleri olması gerçekdir.

#### 3.2.2.2 Reaktif Protokol

Bu yöntem talep edilen yolu yönlendirme sorgulaması ile yönlendirme kurulumu öncesinde hesaplar. Eğer bir düğümün iletilecek bir mesajı varsa, düğüm mesajın iletileceği yolu oluşturacaktır. Bir diğer deyişle yönlendirme yolları ihtiyaç duyuldukça oluşturulmaktadır.

#### 3.2.2.3 Hibrid Protokol

Bu protokol proaktif ve reaktif protokolleri bir araya getirmektedir. Geniş kablosuz sensör ağlarda kullanılabilir. Hibrid protokoller yığılma yöntemi kullanır. Ağ eğer yönlendirmede yığılar oluşuyorsa proaktif olur. Ağ yığın içerisinde düğüm ekler veya düğüm silerken reaktiftir.

### 3.2.3 Yönlendirme işlemi

Bu protokol ağ yararlılığına bağlı olarak bölünmüştür. Sınıflandırmanın amacı, en yüksek performansı ve en az kaynak kullanımını sağlamaktır. Bu sınıflandırmalar:

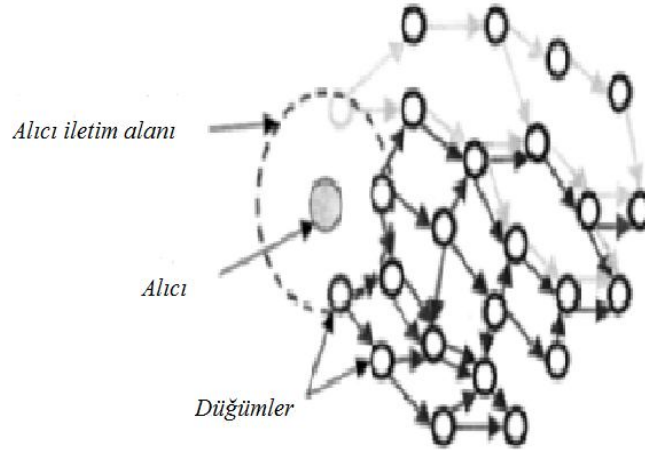
#### 3.2.3.1. Anlaşma Tabanlı

Anlaşma tabanlı protokoller asıl veri transferleri gerçekleşmeden önce komşu sensörler arasındaki anlaşma mesajlarıyla gereksiz veri aktarımlarını azaltmayı amaçlamaktadır. Protokol, ağdaki veri fazlasını yüksek seviyede veri tanımlı şifreleriyle anlaşmada yok edebilir.

- Anlaşma Yoluyla Veri Sensör Protokolleri (Sensor Protocols for Information via Negotiation - SPAN)
- Dizgisel Görev Yönlendirmesi (Sequential assignment routing - SAR)
- Doğrudan Yayılma(Directed Diffusion - DD)

### *Dizgisel Görev Yönlendirmesi (SAR)*

Katayoun Sohrabi, ve diğerleri tarafından öne sürülen sensör ağlar için protokoller grubunun içerisinde yer alan bir yönlendirme protokolüdür [55]. Bağlantı katmanından taşıma katmanına kadar protokoller bu çalışma içerisinde yer almaktadır. Çoklu ağaç yapıları protokol tarafından ağ boyunca inşa edilir. Her ağaç havuza komşu bir düğümde kök sahibidir. Alıcının komşuları, Şekil 3.15'te gösterildiği gibi ağa ağaç oluşturma mesajı içeren paketler yollarlar. Mesaj Alıcıdan daha yüksek dallarda yer alan düğümlere iletilerek yollarlar. Ağaç oluşturma bittiğinde, çoğu sensör birden çok ağaca dahil olacaktır. Dolayısıyla da havuza birden fazla yolla bağlı olacaklardır. Alıcıya yakın sensörler, enerjilerini daha yüksek bir hızda tüketeceklerdir. Birden çok ağaç olması bu düğümler arasında yük optimizasyonunu sağlayacaktır. Kullanılabilecek enerji ve QoS ölçümlerine göre bir yol seçimi yapılacaktır.



Şekil 3.14: SAR

Dizgisel Görev Yönlendirmesi, çok yönlü bir şema kullanıldığı için sensör ağlarda oldukça dayanıklı bir yaklaşımdır. Sensör arızaları oluştuğunda, şemaları uygulanarak bütün ağacın baştan oluşturulmasından kaçınılır. Diğer pek çok protokol gibi, SAR da komşu bilgisini kullanmaktadır. Bu veri çift yönlü bağlantılar gerektirmektedir. Çift yönlü bağlantılar, karmaşık yapı alımlamasını sağlamak için çift yönlü değişimlerde bulunan MAC katman protokolleri tarafından sağlanır. Bu özellikler bazen sensör ağlarda ek yükleri arttırabildikleri için kullanışlı olmamaktadır.

### 3.2.3.2 Sorgu Tabanlı

Bu protokol, sorgu ile ilgi alanının eşleştiği veri düğümü için sorguları alır, transfer eder ve ilk sorgu düğümüne geri iletir. Örneğin (A) baz istasyonu (B) düğümüne bir sorgu gönderir: Bölge 1 deki savaş alanında hareket eden bir obje var mıdır? Düğümler algılama amaçlı sorguların şekillerine sahiplerdir. Bu yaklaşım talep edilmemiş ve gereksiz veri iletimlerini minimize ederek enerjiyi muhafaza eder [56].

### COUGAR

COUGAR algılayıcı veritabanı sistemi, kullanıcı tarafından tanımlanmış sorgular vasıtasıyla bireysel algılayıcı veriler için yayılmış bir veri toplama taslağı sağlar[57]. WSN'deki bilgiler ikiye ayrılır. Bunlar: depolanmış veri ve algılayıcı verileridir. Buna göre, WSN iki sınıf mevcut veriye sahip bir veritabanı sistemi olarak kabul edilir. Depolanmış veri algılayıcıların turu, mümkünse yeri ve ayrıca fiziksel olayların özellikleri gibi bilgileri kapsar. Depolanmış veri alıcılarda mevcuttur ve ilişkiler olarak modellendirilir. Algılayıcı veriler sorgulara göre algılayıcılar tarafından toplanan bilgilere denir. COUGAR'da, algılayıcı veri zaman serisi olarak temsil edilmektedir. Daha spesifik olarak, sensörler senkronize kabul edilir ve algılayıcı veriler kayıt esnasındaki belirli bir yerdeki sinyal işlem fonksiyonu çıkışı olarak kabul edilmektedirler. Her algılayıcı dizi modeli tabanlı veriyi temsil eder. Bunlar, kayıt seti, sıralı bir alan ve alanların sıralanması ile kayıtların sıralanması olarak 3 değişken olarak tanımlanır.

İki sınıfın veri temsiline göre, sorgular iki tür ilişkisel ve sıra operatörlerine dayalı kullanılır. Sıra operatörleri veri sensörü için geçerli iken ilişkisel operatörler, depolanan veriler için uygulanır. Ayrıca, üç operatör ilişkileri ve dizileri birleştirmek için tanımlanır. Buna göre, (1) diziler, bir ilişki elde etmek için konum bilgisi dışında projeye girdi olarak kullanılabilir, (2) bir dizi ve bir ilişki, bir dizi elde etmek için bir ürün operasyonunda kullanılabilir ve (3) bir dizi pozisyon özniteliği haricine göre toplanmış olabilir. Algılayıcı düğümleri, örnekleme, karşılaştırma, entegrasyon gibi, belirli sinyal işleme fonksiyonu bazında toplanan verileri gözlemler. Buna göre, COUGAR içinde, bireysel sensörlerin çıkışları (ADT), işlevleri soyut veri turu olarak temsil edilmektedir.

Sonuç olarak, veritabanında bir ADT nesnesi gerçek dünyada fiziksel bir algılayıcıya karşılık gelir. Bu soyutlamayı kullanarak, COUGAR veritabanı sistemi algılayıcılara sorgu atarken SQL benzeri bir sorgulama dili kullanır. Bu ileri bölümlerde gösterilmiştir.

### 3.2.3.3 QoS (Servis Kalitesi) Tabanlı Protokol

WSNler'deki yönlendirme protoklu ve veri dağıtımındaki enerji tüketiminin azaltılmasının yanında QoS gereklilikleri de gecikmenin, üretilen işin, titreşimin, gizliliğin, bant genişliğinin ve güvenilirliğin tanımında önemlidir. Aynı zamanda protokoller nadir ağ kaynaklarının kullanımını optimizasyonunu amaçlarlar. QoS tabanlı yönlendirme protokollerinde, ağın enerji tüketimi ve veri kalitesi arasında bir denge kütması zorunludur. Kablosuz algılayıcı ağlar dinamik topolojileri, kaynak kıtlığı (güç kısıtlamalarını içerir), radyo kanallarının kalite çeşitlendirmesi, mekrezleşmiş kontrolün eksikliği, ağ araçlarının heterojenliğini içeren QoS'nin tatminini sağlamak amacıyla yüzlerce kanal oluşturur. Bu tür protokollara örnek olarak SAR, SPEED ve ARRIVE. Bu bölümde SPEED ve ARRIVE protokolü incelenecektir.

#### *SPEED*

Bu algılayıcı ağlarda yumuşak gerçek zamanlı iletişim için yersiz protokoldür[58]. Her bir düğüm komşularının hepsi hakkında bilgi tutar ve yollarını bulmak için bir coğrafi yönlendirme tekniğini kullanır. Düğümler arasındaki uçtan uca gecikme ortalaması fikrine sahip olan uygulamalar gibi gönderilen her bir paket için SPEED protokolü belirli bir hızı garanti etmek için uğraşır. Eğer yönlendirme protokolü mevcut paket tahmini hızına ulaşırsa, uygulama sadece kaynak ve tahmini hızı hedef arasındaki mesafeyi böler, önceden bu gecikme hesaplanabilir.

Protokol ayrıca, gerektiğinde tıkanıklık önleyici mekanizma sağlar. SPEED' deki yönlendirme bileşeni olarak adlandırılan yersiz coğrafi belirleyici olmayan iletim, etkin çalışma için aşağıda sunular dört modülü kullanır.

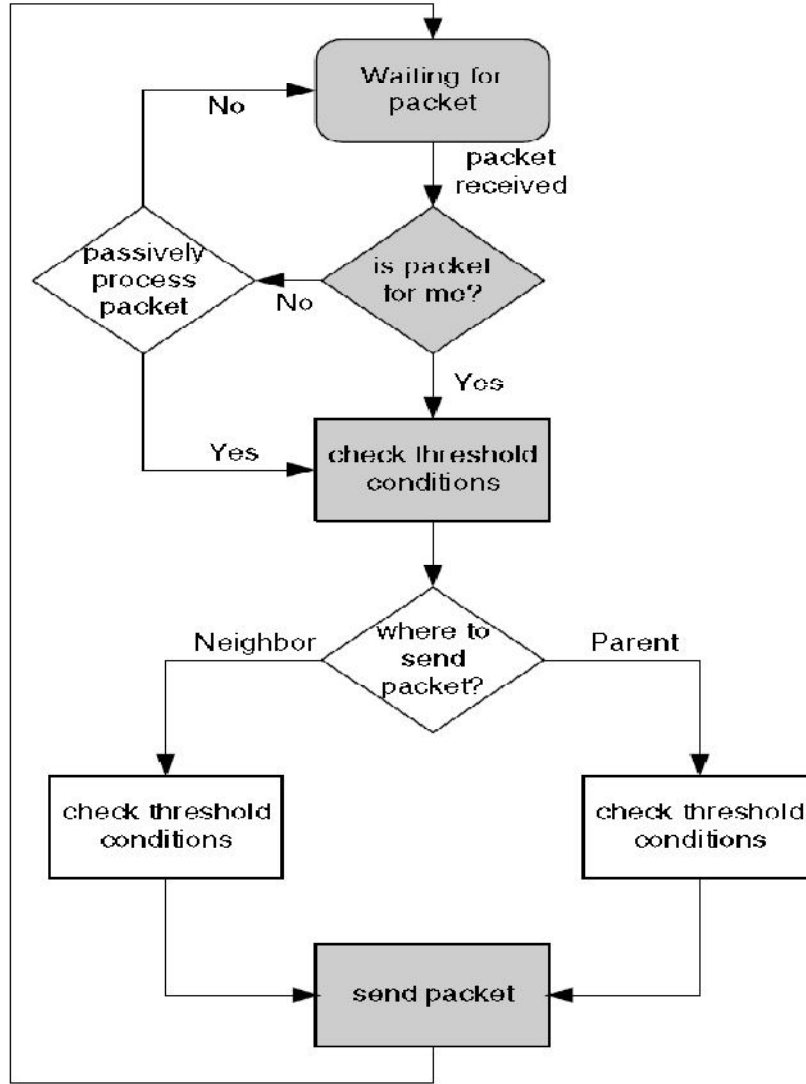
- Uyarı Değişimi: Diğer coğrafi yönlendirme algoritmasına benzer şekilde, konum bilgisi belirli beacon paketleri yayını kullanılarak düğümler tarafından periyodik olarak güncellenir. SPEED içindeki talep işaretleri iki çeşittir. Gecikme tahmini ve geri basınç işaretleri.

- Gecikme Tahmini: Son veri paketinin gönderiminin bilgi kağıdından beri hesaplanan geçen zaman hesaplamalarıyla her bir düğümde gecikme tahmini yapılır. Bu gecikme değerleri gözlemleyerek, SNFG uygulamanın hız ihtiyacına uygun en iyi düğümü seçer.
- Komşu Geri besleme Döngüsü: Uygun bir gecikme değeri ile bir düğüm elde edilirse, bu modül düğümün role oranıtısı için danışmanlık yapmaktadır. Bu oran düğümün bütün komşularının ıskalama oranıdır yani uygulama için yeterli hıza erişemeyen düğümlerdir. Sıfır ve bir arasında rastgele bir sayı üretilir; role oranı oluşturulan sayıdan küçükse paket bırakılır.
- Geri basınç tekrar yönlendirmesi: Bu modül alternatif rotalar tespit edilecek şekilde kaynak düğümlere mesajları göndererek sıkışıklığı azaltmaya yardımcı olur.

SPEED' in yersiz mimarisi nedeniyle, az bellek ve minimal MAC katmanı desteği gerektirir. Aynı zamanda trafik yükü dengeleme sağlar ve algılayıcı ağ uygulamalarında QoS yönlendirmesi ve tıkanıklık yönetimini harekete geçirir.

#### *Geçici Ortamlarda Dayanıklı Yönlendirme için Algoritma (ARRIVE)*

Bu bir olasılıksal algoritmadır, yerel bilgilere dayanarak paket iletimine karar verir ve bu ağın kökü ağın alıcısında olan ağaç benzeri bir topolojisi [59] vardır. İletim yaklaşımı uçtan uca güvenilirlik elde etmek için uygulanmıştır. Paket kaybı, tek bir olaya birden fazla paket göndererek önlenir. Temelde, beklenen paket kaybında üç kaynaktan (i) izole bağlantı, (ii) desenli düğüm hataları ve (iii) kötü niyetli veya kötü davranışlar sergileyebilen düğümler vardır. Şekil 3.16'da ARRIVE algoritmasını incelenmektedir.



Şekil 3.15: ARRIVE algoritması

- \* Etkinlik: Kaynak kimliği, Olay Kimliği ile tanımlanır.
- \* Seviye: Her düğüm kaynak ile alıcı arasındaki uzaklığı belirten eşsiz bir seviyeye sahiptir.
- \* Üstler: Alıcıya bir seviye daha yakın olan düğümlerdir.
- \* Komşular: Birbirleriyle aynı seviyede olan ve birbirlerini duyabilen düğümlerdir.
- \* İtme: Paketleri komşulardan birine iter.
- \* İletim: Üstlerden birine paket iletimidir.
- \* İletim olasılığı: Paket başlığına dâhildir ve itmeyi ya da iletmeyi olasılıksal olarak seçer.

\*İtibar geçmişi: Her düğüm üssü ve komşuları ile ilgili bilgileri tutar.

\*Yakınsama: Aynı etkinliğin birden fazla paketinin aynı hata kaynağına gönderilmesini engeller.

ARRIVE iki şekilde çeşitlilik sağlar: (İ) bir paket olarak, bir sonraki atlama bağlantı güvenilirliği ve düğüm itibarına dayanarak olasılıksal olarak belirlenir (ı) aynı olaydan iki veya daha fazla paket işlemi olduğunda, bu paketler giden farklı bağlantıların takibi için garanti edilir. Her biri aşağıdaki bilgileri bulundurur: seviye, komşuların listesi, üst listesini, komşuların ve üslerin itibar tarihi ve belirli olayların yakınsama geçmişi.

#### 3.2.3.4 Çoklu yol yönlendirme

Çok yollu yönlendirme teknikleri başarısız düğümler veya geçersiz bağlantılar etrafındaki yön için hedef kaynağa giden birden fazla yolu hesaplar. Tek yollu yönlendirme protokollerinde biir bağlantı başarısız olursa, ek kontrol paketleri oluşturulmalı ve yeni yollar keşfetmek için yayın yapılmalıdır. Çoklu yönlendirmede böyle tuzaklardan kaçınan bir orta yol, başarısızlık durumunda hazır bir rota her zaman vardır.

#### *ReInForM*

Çoklu yollar protokolünü kullanan güvenli bilgi iletimi algılayıcı ağlardaki bilgi farkındalığı veri teslimatı ile ilişkili konuları ön plana getirir [60].Gerekli verilerin kayıplı ve güvenilmez yol boyunca gönderilmesi gerçekleşebilir. Bir ormandaki farklı noktalarda bir algılayıcı ağın sıcaklık algılamasını inceleyelim. Belirli bir günde, bir algılayıcı normal bir sıcaklıkta belirli bir noktada 60 ° F sıcaklık algılayabilir. Öte yandan, o anda başka bir sensor uzakta bir yerde 1000 ° F. sıcaklık algılayabilir. Anormal sıcaklık içeren bu paket daha önemlidir ve en düşük gecikme ile teslim edilmesini sağlamak için daha güvenilir bağlantılarla hedefe gönderilmesi gerekmektedir. Temelde önerilen mekanizmalar istenilen güvenilirlik ile veri dağıtımını desteklemektedir. Ancak, güvenilirlik artışı ile bir paket transferi için gerekli ek yük içinde genel bir artış beklenir. Bir paketin güvenilirliği yedek kopyalar tarafından sağlanır iken kaynaktan hedefe doğru olan çoklu yollar keşfedilir. İstenilen güvenilirlik, yerel kanal hata durumları ve her düğüm noktasında komşunun gerekli bilgileri ihtiyaç duyulan yedeklerin derecesini belirler. Başka bir önemli özelliği de verileri önbelleğe alma ve böylece bellek tasarrufunun ara düğümlerde de gerekli olmasıdır.

### 3.2.3.5. Coherent and Non-coherent based protocol( Tutarlı ve tutarsız tabanlı protokol)

- Tutarlı yönlendirmede, minimum işlemden sonra veriler toplayıcılara iletilirler.
- Tutarsız veri işleme yönlendirmesinde, veriler sonraki süreçler için diğer düğümlere gönderilmeden önce düğümler bölgesel olarak ham veriyi işlemeye başlayacaklar.

Veri işleme kablosuz sensor ağların çalışmasında önemli bir bileşendir. Dolayısıyla, yönlendirme teknikleri farklı veri işleme tekniklerini kullanırlar. Genel olarak, düğüm sensörleri ağ alanındaki işlenen farklı data akışlarının her biri ile işbirliği yapacaklardır. WSN'lerde önerilen veri işleme tekniklerine iki örnek olan tutarlı ve tutarlı olmayan veri işlemeye dayalı yönlendirme.[61]. Tutarsız veri işleme yönlendirmesinde düğümler ham veriyi ileriki işlemler için diğer düğümlere göndermeden önce ham veriyi işlerler. Daha ilerideki süreçteki düğümlere 'toplayıcılar' denir. Tutarlı yönlendirmede, asgari veri işlemeyen sonra veri toplayıcılara iletilir.

Minimum işlem genellikle damgalama ve bastırma gibi yinelenen zaman görevleri içerir. Enerji etkili yönlendirmeyi gerçekleştirmek için normal olarak tutarlı işlem yönlendirmesi seçilir. Tutarsız fonksiyonlar oldukça düşük veri trafiğinde yükleme yaparlar. Öte yandan, tutarlı işlem uzun veri akışları üretir, enerji verimliliği uygun yol tarafından sağlanmalıdır.

Tutarlı olmayan veri işleme üç aşamadan oluşur. Bunlar:

- Hedef tespit, veri toplama ve ön işleme
- Üyelik bildirim
- Merkezi düğüm seçimidir.

1. faz sırasında, hedef tespit edilmiş, kendi verileri toplanmış ve on işleme yapılmıştır. Düğüm bir kooperatif işleme katılmaya karar verdiğinde, 2. faz devreye girerek tüm komşulara bu durumu iletir. En kısa sürede her bir algılayıcı yerel bir ağ topolojisi anlayışına sahip olur ki bu mümkün olduğunca yapılmalıdır. 3. faz merkez ise düğümün seçimi olacaktır. Merkez düğüm daha karmaşık bilgi işlem gerçekleştirmek seçildiğinde, yeterli enerji rezervleri ve hesaplama yeteneğine sahip olmalıdır [61].



*Tek kazananla algoritmada (SWE)*

tutarlı ve tutarsız işlemler için tek ve çok kazanan algoritmalar sırasıyla önerilmiştir. Tek kazananlı algoritmada (SWE), karmaşık işleme için tek bir toplayıcı düğüm seçilir. Seçilen düğümün enerji rezervleri bu düğümün hesaplama yeteneğine dayanmaktadır. SWE sürecinin sonunda, minimum atlama mesafesi yapan düğüm ağı tamamen kapsayacaktır. Birden çok kazanan algoritma (MWE) olarak, SWE'nin basit bir uzantısı olarak önerilmiştir. Tüm düğüm kaynakları zaman ve merkezi toplayıcı düğüme bu verileri gönderirken enerji büyük miktarda tüketiliyor olacak; dolayısıyla, bu süreç yüksek bir maliyet gerektirecektir. Enerji maliyetini daha aza indirmenin bir yolu merkezi toplayıcı düğüme veri göndermek ve kaynaklarının sayısını sınırlamaktır. Yerine sadece en iyi aday düğümün (ana toplayıcı düğüm) kaydını tutarak, her bir düğüm adayının n düğüme kadar kaydını tutacaktır. MWE sürecinin sonunda, ağdaki her algılayıcı, her kaynak düğüm (SN) için bir dizi minimum enerji yolları var. Bundan sonra, SWE minimum enerji tüketen düğümleri bulmak için kullanılır. Bu düğüm sonra tutarlı işlem için merkezi düğüm olarak hizmet verebilir. Genel olarak, MWE daha uzun gecikme, daha fazla ek yük ve tutarsız işlem ağlarından daha az ölçülebilirliğe sahiptir.

## 4. ENERJİ VERİMLİ YÖNLENDİRME PROTOKOL ARIN KARŞILAŞTIRILMASI

Önceki bölümlerde de görüldüğü üzere, Kablosuz Sensör Ağları (WSN) tasarım gereksinimleri uygulamaya göre değişen, uygulamaya özgü ağlardır. Bu nedenle yönlendirme protokollerinin gereksinimleri bir uygulamadan diğerine değişir. Örneğin, çevresel uygulamalar için tasarlanmış bir yönlendirme protokolünün gereksinimleri, askeri uygulamalar veya sağlık uygulamaları için olanlardan birçok bakımdan farklıdır. Sonuç olarak yönlendirme protokollerinin gereksinimleri uygulamalar kadar çeşitlidir. Bunlardan bir kısmı; Ölçeklenebilirlik, Gecikme, Verimlilik, Başvuru Farkındalığı, Veri Toplama, En Uygun Rota, Ek Yük ve diğer ölçütlerdir. Bazı uygulamalar bu ölçütlerden bazılarının sağlanmış olmasına ihtiyaç duyarken, diğer uygulamalar diğerlerinin sağlanmış olmasına ihtiyaç duyar. Bununla birlikte, tüm Kablosuz Sensör ağları uygulamadan bağımsız olarak ağ yaşam süresini maksimize etmeye ve tüm ağdaki enerji tüketimini minimize etmeye çalışmak zorundadır veya mecburdur.

Yaygın fenomene dayanarak, WSN'lerde veri birçok sensör tarafından toplanabilir ve bu nedenle veri fazlalığı ihtimali olabilir. Böyle bir fazlalık verimli bir enerji ve bant genişliği kullanımına ulaşabilmek adına kabul edilemezdir. Bu durum yönlendirme protokollerinin kötüye kullanılmasıdır. Dahası, bazı WSN'ler talep edilen verinin nitelik tabanlı adreslemeye dayalı olduğunu algılayan veri-merkezli ağlardır (bir dizi nitelik-değer çifti sorgusundan oluşan). Mesela, [sıcaklık > > 50°F] gibi bir sorgu, yalnızca > 50°F'den yüksek sıcaklık değerleri algılayan sensör düğümlerinin yanıt vermesi ve gösterge değerini raporlaması gerektiği anlamına gelir.

Bu farklılıklar nedeniyle, uygulama ve mimari gereksinimleri ile birlikte WSN'lerin kendilerine özgü özellikleri dikkate alınarak WSN'lerdeki yönlendirme problemleri için birçok yeni algoritma öne sürülmüştür. Bu bölümde bazı iyi bilinen yönlendirme protokollerini (LEACH, Directed Diffusion, Gossiping ve EESR) ve onların açılımlarını (iyileştirmelerini) ve ayrıca veri toplama, ağ-içi işleme, kümeleme ve farklı düğüm rol atamaları ve veri merkezli yöntemler gibi WSN'lere özel taktikleri araştırdık.

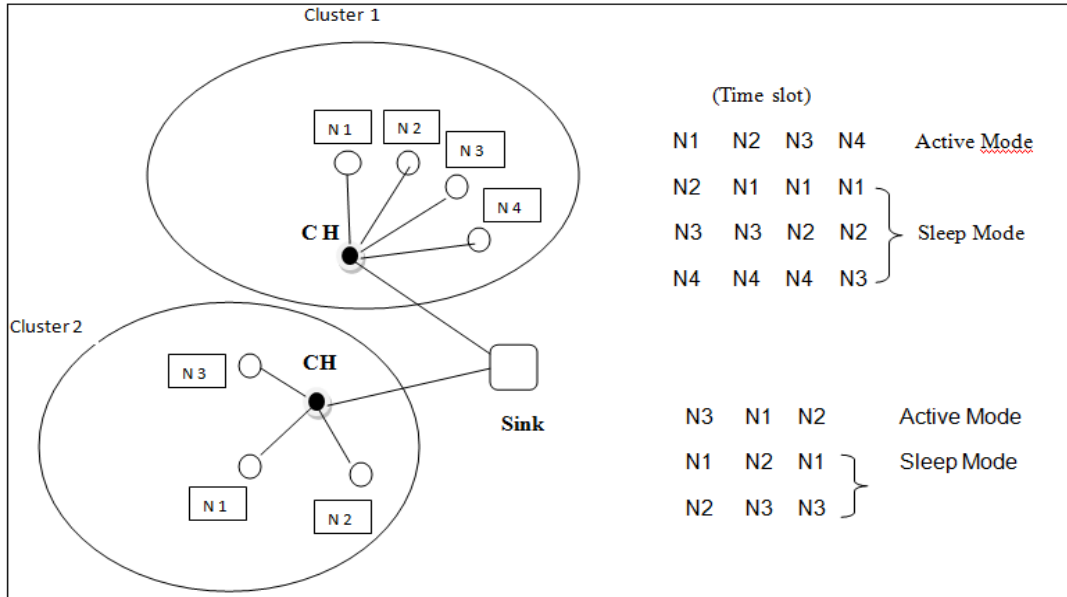
Ondan sonra özelde uygulama gereksinimlerini etkileyen farklı ölçütlere ve genelde WSN'ye dayanan, bu araştırılmış olan yönlendirme protokollerini kıyasladık. Son

olarak, bu bölümün sonunda yeni protokolümüzü öne sürecek ve onu benzerleri ile kıyaslayarak simülasyon sonuçlarımızı göstereceğiz.

#### 4.1 LEACH PROTOKOLÜ

düşük enerjili adaptif kümelendirme hiyerarşisi (LEACH) : kümeleme tabanlı bir protokoldür, yerel küme baz istasyonlarının rasgele dönmesinde kullanılır. LEACH’de düğümler kümelere ayrılır ve her küme Küme üyeleri ve Küme Başı denilen CH adlı koordinatör bir kümeden oluşur. Küme başları ağda çabuk sona ermeye yol açan statik biçimdeki düğümlerden seçilmez. Ancak randomize protokolü ağdaki diğer düğümlere CH rolü dağıtarak düğümler arasında enerji tüketimini dengelemek için kullanılır. Ayrıca, LEACH küme ile kanal erişimini düzenleyen Zaman Bölmeli Çoklu Erişim (TDMA) protokolünü kullanır [32].

TDMA yuvalarında küme üyelerine görev vermek CH’lerin sorumluluğundadır .CH ve üyesi arasındaki eş zamanlı(peer- to- peer) iletişim sadece zaman dilimi üyeye atadığında yapılır ve diğer üyeler uyku durumunda olacaktır. Dolayısıyla enerji kaybı azalır, şekil 4.1’e bkz [62]. Dahası. LEACH kümeler arasındaki girişimi azaltmak için CDMA protokolünü kullanır.



Şekil 4.1 : LEACH protokolü ve TDMA programları[63]

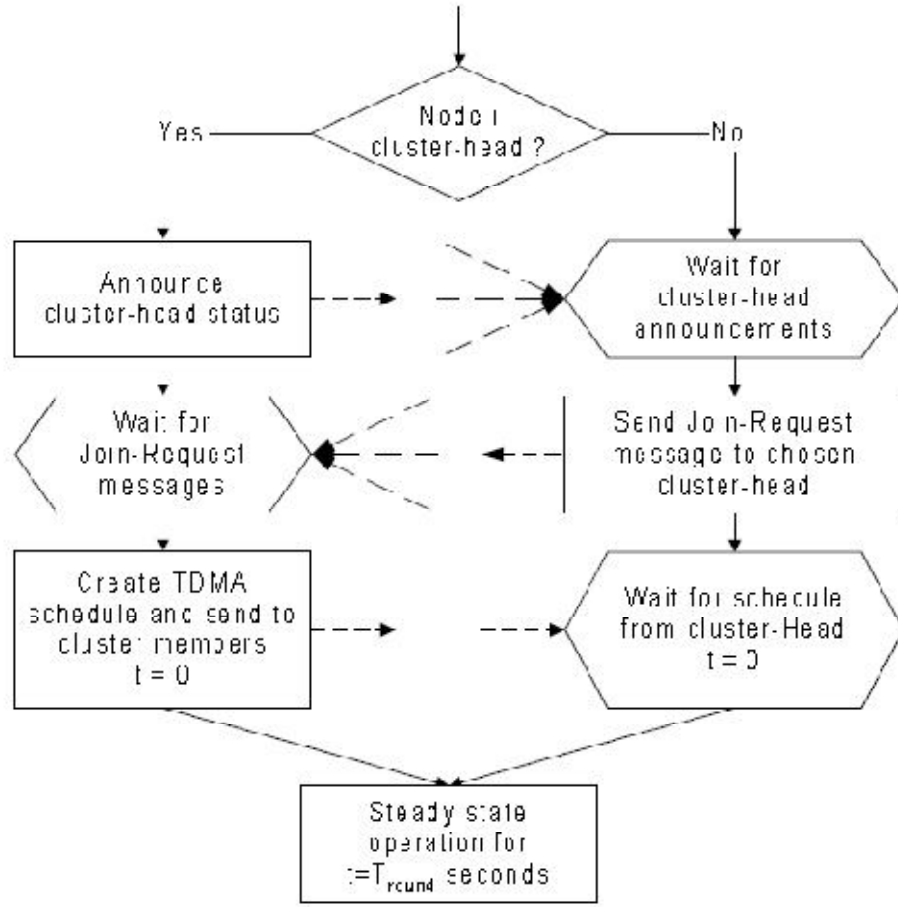
LEACH, daha fazla enerji tüketen düz-mimari protokollerinin dezavantajlarını aşmak için hazırlanmıştır .CH düğümlerden gelen verileri küçük ebatlı ve anlamlı verilerle toplar ve birleştirir ve sonra daha az enerji tüketerek verileri toplayarak baz-stasyona gönderir. Burada enerji tüketimi mesafeye ve veri boyutuna bağlıdır .Bu nedenle her bir yuvada oluşan kümeyi iki parçaya böler ; kurulum aşaması(setup phase) ve sabit durum(steady state) aşaması :

*Kurulum aşaması:*

Bu aşama boyunca kümeler biçimlendirilecek ve her biri düğüm CH olmak veya olmamak için karar verir. Geçerli yuvarlağta bir düğüm (N) (P) 0 ile 1 arasında seçtiği rasgele sayı olasılığı T(n) eşiğinden daha az olursa, bu düğüm CH' olarak seçilir . T(N) aşağıdaki formuldaki gibi hesaplanır :

$$T(n) = \begin{cases} \frac{P}{1-P(r \bmod 1/P)} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases}$$

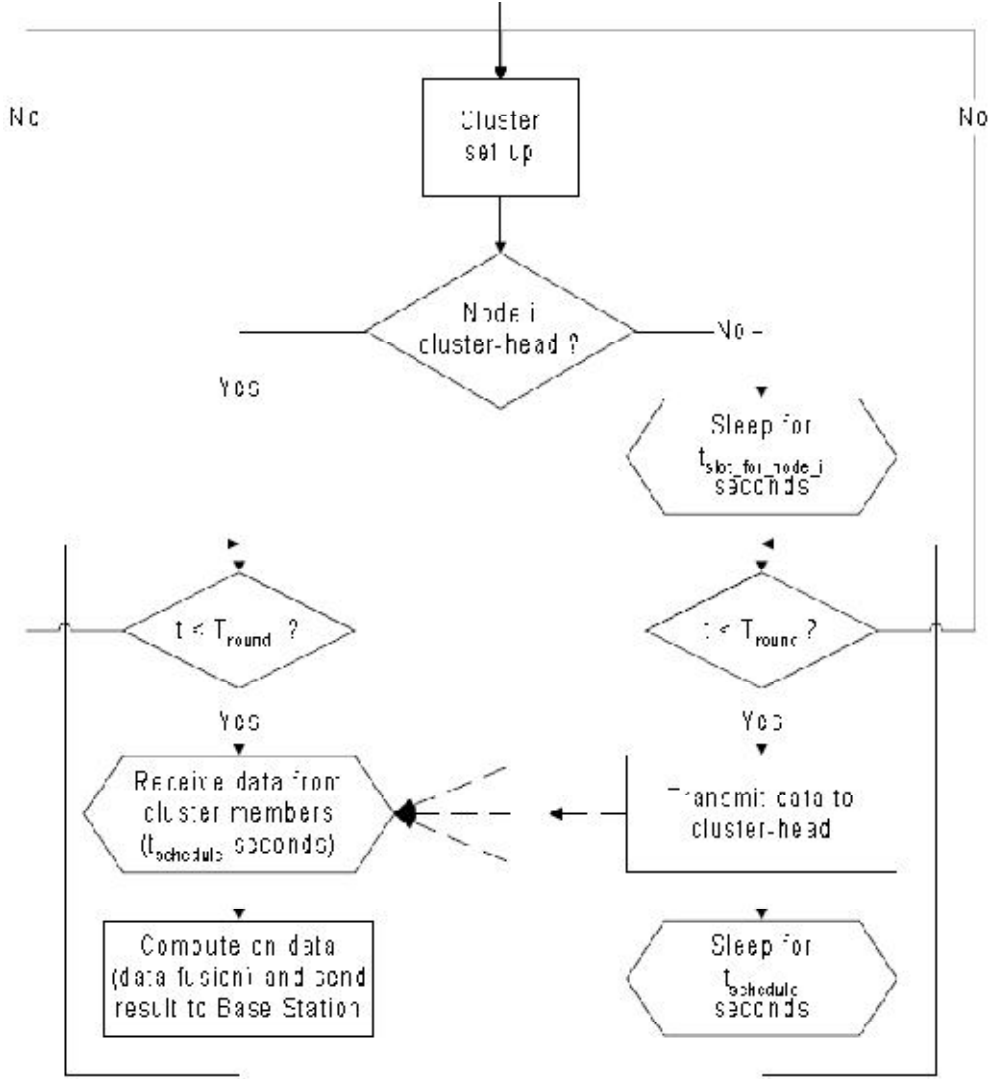
CH birkez seçildikten sonra, CSMA protokolünü kullanarak Yeni CH'lar kendilerini oldukları gibi diğer sensör düğümlerine tanıtırlar. Her bir Düğüm küme başları tarafından güçlü sinyalleri alarak uygun CH'lere katılır. Bu aşamada her non-CH CH tarafından yayılan sinyalleri algılayabilmesi için alıcılarını açık tutmaları gerekir. Benzer türdeki küme başları katılım mesajlarını duyması için alıcılarını açık tutmalıdırlar. Şekil 4.2 akış şekline bkz [63].



Şekil 4.2 : LEACH için kurulum faz grafiği

**Sabit durum aşaması:** Bu aşamada CH'ların kendi üyeleri tarafından veriler algılanacak ve yayılacaktır. CH bu verileri toplayacak ve baz istasyona gönderecektir. Ağ belirli bir zaman sonra kurulum aşamasına gidecektir. Sonuç olarak ek yükü(overhead) azaltmak için sabit durum aşama süresi kurulum süresinden daha uzundur. Şekildeki akış grafiğine bkz.4.3[63].

Bir kaç sayı turlardan sonra küme başı olamamış düğümün enerjisi düşük olacaktır ve CH eşiğinin yene düşük olacaktır. Böylece Bazı CH'ların baz istasyonlarına verilerini iletmeleri mümkün olmayacaktır. Bu nedenle, Bir tur sayıda CH olmayan Düğümlerin değişik düzende çoğaltmak için eşik denklemini bir faktör ekleyerek düğüm eşiğini çoğaltarak düzenleyebilir.



Şekil 4.3 : LEACH için sabit durum işleminin akış grafiği[63]

küme biçimlendirdikten daha sonra ( LEACH protokolünün işleyişini tanımlayan pseudo-kodune şekil 4.4 bkz), küme içindeki düğüm daha az enerji dağıtarak daha kısa mesafelere veri gönderebilecek. Ancak ,CH'lar doğrudan doğruya Baz istasyonla ilettiğinden büyük miktarda enerji tüketirler. Sonuç olarak, LEACH'in en büyük problem ise CHS baz istasyona doğrudan veri göndermesi, özellikle eğer bu CH'ler BS'den uzak ise. Ancak, CH'ler başka CH'lerle yolu ile BS'ne doğru çoklu-Atlama(multi-hop) iletim kullanarak bu sorunu çözecek, CH'ler BS ulaşana kadar sadece birbirine verileri gönderir ve başka CH'lere gelen verileri yeniden toplama zorunda değildir.

LEACH, doğrudan iletişim ve diğer düşük enerji yönlendirme protokoller ile karşılaştırıldığında enerji dağılımında önemli bir azalmayı başarmaktadır. Son olarak LEACH (avantajları ve dezavantajları) ana özellikleri içermektedir. [32,64]:

```

Setup Phase:
In this phase clusters are created---cluster heads (CHs) are
chosen

1.   ForEach (node N)
2.     N selects a random number r between 0 and 1
3.     If (r < Threshold value)
4.       N becomes a CH
5.       N broadcasts a message advertising its CH status
6.     Else
7.       N becomes a regular node
8.       N listens to the advertising messages of the CHs
9.       N chooses the CH with the strongest signal as its
       cluster head
10.      N informs the selected CH and becomes a member of
       its cluster
11.    EndIf
12.  ForEach (clusterhead CH)
13.    CH creates a TDMA schedule for each node to transmit
       data
14.    CH communicates the TDMA schedule to each node in the
       cluster
15.  EndFor

Steady State Phase:
1.  ForEach (regular node N)
2.    N collects sensed data
3.    N transmits the sensed data to the CH in the
       corresponding TDMA time slot
4.  EndFor
5.  ForEach (cluster head CH)
6.    CH receives data from the nodes of the cluster
7.    CH aggregates the data
8.    CH transmits the data to the base station
9.  EndFor

```

Şekil 4.4 : LEACH protokolünü takma-kod operasyonu açıklar

#### 4.1.1. LEACH'in Avantajları

- Kümelerin içindeki iletişimin birçoğunu sınırladığından ağır ölçeklenebilirliğini sağlar
- CH'lar düğümler tarafından toplanan verilere ve oluşturulan ağ trafiğine sınırlama getirir. Dolayısıyla, büyük ölçekli ağ trafiğini aşırı yük olmadan dağıtır ve düz topolojiye göre daha iyi enerji verimliliği elde edebilir.
- Küme başı ile düğümler arası tek-atlama(single-hop) yönlendirmesi kullanır, Dolayısıyla enerji tasarrufu sağlar.

- Dağıtıcı, değer düğümlere CH rolünü dağıtır •
- Ağın ömrü üç şekilde artar. Birincisi, Diğer düğümlere CH rolü dağıtma(normal düğümlerden daha fazla enerji tüketir). İkinci olarak, CH'lar tarafından veri toplayarak. Son olarak, CH tarafından üyelerine atanan TDMA, çoğu algılayıcılarını uyku modunda koyar, özellikle olay-tabanlı(event-based) uygulamalarda. Bu nedenle ağ ömrünü arttırmak ve enerji dağılımında doğrudan iletişime kıyasla yedi kat daha fazla azalma eldeedebilir.
- Düğümleri oluşturmak için konum bilgisi geyektirmez. Bu nedenle, güçlü ve basittir.
- Son olarak, Dinamik kümeleme ve burada tabi sürekli izleme ve merkezi bir konumda periyodik veri toplama işlemi gerçekleştirildiği uygulamalar için uygundur.

#### 4.1.2. LEACH'in Dezavantajları

- Önemli ölçüde küme başlarına itimat edilir ve küme başı kaybı fazla olur, bu nedenle sağlamlılık konusunda iyi olmaz.
- Küme baş değişikliği nedeniyle ekstra yüke(overhead) ve geniş ağlarda dinamik kümelendirmelerde hesapsız enerji verimsizliğine yol açar.
- CH'lar BS ile doğrudan iletişime geçer, kümeler arası(inter-cluster) iletişimi olmaz ve yüksek iletişim gücü gerektirir. Bu nedenle BS ile tek-sekmeli(single-hop) haberleşme ağı büyük ölçekli ağlarda iyi çalışmaz.
- CH'lar eşit şekilde dağıtılmaz, CH' ler küme kenarlarında bulunabilir.
- CH rasgele seçildiğinden enerji tüketimini hesap almaz.
- Son olarak çolu-sekme(multi-hop) kümeler arası iletişimi gerektirdiğinden geniş alanı kaplayacak uygulamalarda iyi çalışmaz.

Bazı LEACH dezavantajları nedeniyle protokolden daha iyi hizmet alabilmek için araştırmalar yapılmıştır. Bu araştırmalardan bazıları; E-LEACH, TL-LEACH, M-LEACH, LEACH-C ve V-LEACH [65].



### **E-LEACH**

Enerji-LEACH Protokolü CH seçim yordamı geliştirir. LEACH gibi turlara(round) ayrıldı, ilk turda CH olmak için tüm düğümlerin aynı olma olasılığı var. Ancak, her bir düğümün ilk turdan sonra kalan enerjileri farklıdır ve en yüksek kalan-enerjili düğüm CH olarak seçilir .

### **TL-LEACH**

LEACH içinde, CH veriyi bir sekmenin içindeki BS'ne gönderir. Ancak, İki aşamalı LEACH de, CH1 küme üyelerinden veri toplar ve CH1 ile BS arasındaki başka bir CH2 aracılığıyla verileri aktarır. [33].

**M-LEACH:** Yukarıda belirtildiği gibi, LEACH de CH veriyi tek-sikmede baz istasyonuna gönderir. Çok-sekmeli (multi-hop) LEACH protokolünde, CH'den gelen veri başka CH'lei geçiş istasyonu yoluyla kullanarak BS'ne aktarır. [66] Bu protokolda baz istasyona uzak olan CH, veri iletimleri sırasında büyük enerji tüketildiği sorunu çözüldü,

**LEACH-C:** LEACH CH'ların yerleri hakkında hiçbir bilgiye sahip değildir. Ancak, Merkezi LEACH protokolü ağ boyunca küme başları dağıtarak daha iyi performans üretebilir. Kurulum aşamasında ,her düğüm BS'ne kalan enerjiyi ve konumunu gönderir. BS bu turda kümeleri belirlemek üzere merkezi küme oluşum algoritması çalıştırır. Ancak, Bu protokol tüm algılayıcı ağ (normalde GPS tarafından sağlanan) için konum bilgilerini gerektirdiğinden sağlam değildir

**V-LEACH:** LEACH protokolünün yeni sürümünde küme içerisindeki CH yok olduğu zaman yerini alacak yedek bir CH bulunur [66]. CH öldüğünde, küme işe yaramaz hale gelir çünkü küme üyeleri tarafından toplanan veriler BS'ne ulaşmayacaktır.

## **4.2. DIRECTED DIFFUSION**

Yönlendirilmiş Dağılım WSN'ler içinde bilgileri toplamak ve yayımlamak için , veri merkezli yönlendirme protokolüdür. BS'nı bu sensörlerden belirli bilgileri istediğinde , BS'den sensörlere doğru akan veri ihtiyacını iyileştirmek için geliştirilmiştir[67] . Ana amacı temel enerji tasarrufu gerçekleştirerek ağın yaşam süresini genişletiyor. Bu amaca

yerine getirmek için, ileti alışverişi yoluyla sınırlı bir ortamda düğümler arasındaki etkileşimleri tutmak zorundadır. Çok-yönlü iletişim sağlayan yerleştirilmiş etkileşimi bu protokolün benzersiz bir özelliğidir. Bu benzersiz özellik düğümlerin yardımı ile BS'nın sorgu sonuçlarına yanıt olarak önemli ölçüde enerji tasarrufu sağlar [67].

DD ana bileşenleri şunlardır , ilgi, veri iletileri, gradyanlar ve takviye bileşenleridir.

**İlgi (Sorgu veya Sorgulama)** tetkik'in (kullanıcı) ne istediğini içerir ve veriler kullanıcı tarafından ilgili bir açıklama belirtir. Genellikle bu tabloda gösterildiği gibi öznitelik – değer çiftleri kullanılarak ifade edilir. 4.1[67].

**Veri iletileri** ya yoplanan veya işlenmiş bir olgu bilgileridir yada ilgi(interest) veya kullanıcı isteğine uygun bir olay dahi olabilir.

**Gradient, ilgiyi aldığı düğümleri gösterir** . BS'ne (tetkik veya kullanıcı) doğru ters yollar (çoklu yönlü ) oluşturmak için kullanılır.

**Takviye(reinforcement)** tetkik'i (BS) belirtilen yol boyunca aynı ilgiyi gradient aşaması kısmında yeniden göndermeyi takviye eden yollardan biridir

Tablo 4,1: İlgi açıklama kullanarak öznitelik ve değer çiftleri (sinekkuşu ile ilgili)

Öznitelik- değer çifti	Açıklama
Türü =Sinekkuşu	Sinekkuşunun yer tespiti
Aralık(Interval) = 20 ms	Her 20 ms'te rapor olayları
Süre = 10 s	Bir sonraki rapor için 10 s
Alan = [(x1,y1),(x2,y2)]	Bu alandaki algılayıcı raporu

BS (tetkik ) ve bağlantının isteğini ilgilendiren algılayıcılar arasında rota oluşturmak için dört aşama;(1) İlgi yayılması, (2) degrade kurulumu, (3) Güçlendirme ve (4) Veri teslimi. Her aşamada ayrıntılı bir açıklama:

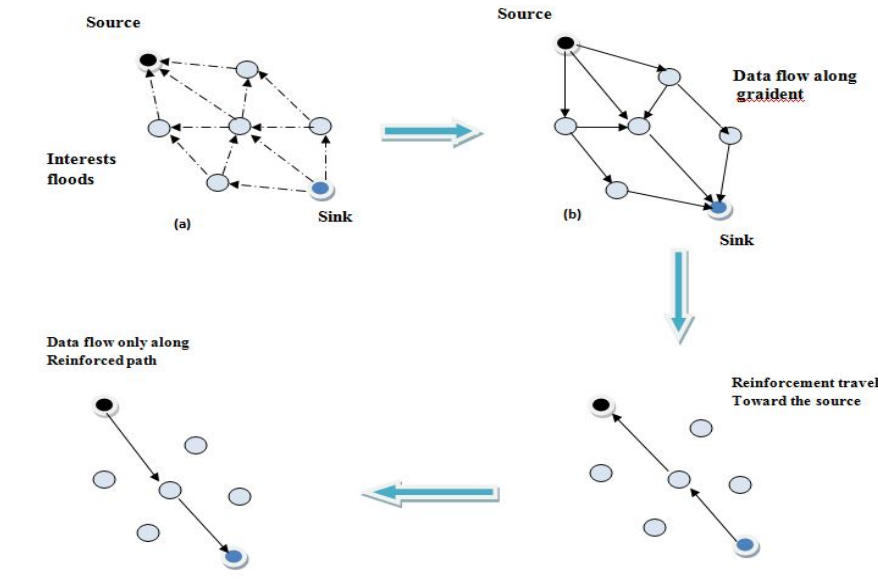
**(1) İlgi yayılması:** Bir BS bir olay istediğinde ya da ilgilendiğinde, ilgi iletimini başlatır ve bu ağdaki tüm düğümler iletir. Bu iletiler keşif iletilerdir, özel görev verileriyle eşleşen düğümleri gösterir. Bu aşama sırasında BS periyodik olarak ilgi iletisi yayınlar. İlgi iletisi alındıktan sonra her bir algılayıcı düğümü, bu ilgiyi ilgi önbelleğine kaydeder. İlgi önbelleği, zaman damgası ( ilgi alma süresi) , *gradient*

(eğim yolundan BS'ne doğru ilgi alınan düğümler) ,aralık ve süre (ilginin önbellekte kalma süreleri) içerir. Sonra bu iletiyi çok sayıdaki düğümlere iletir. İletim kaynağa ulaşana kadar devam eder. Bkz Şekil4.5 (a) Bkz.

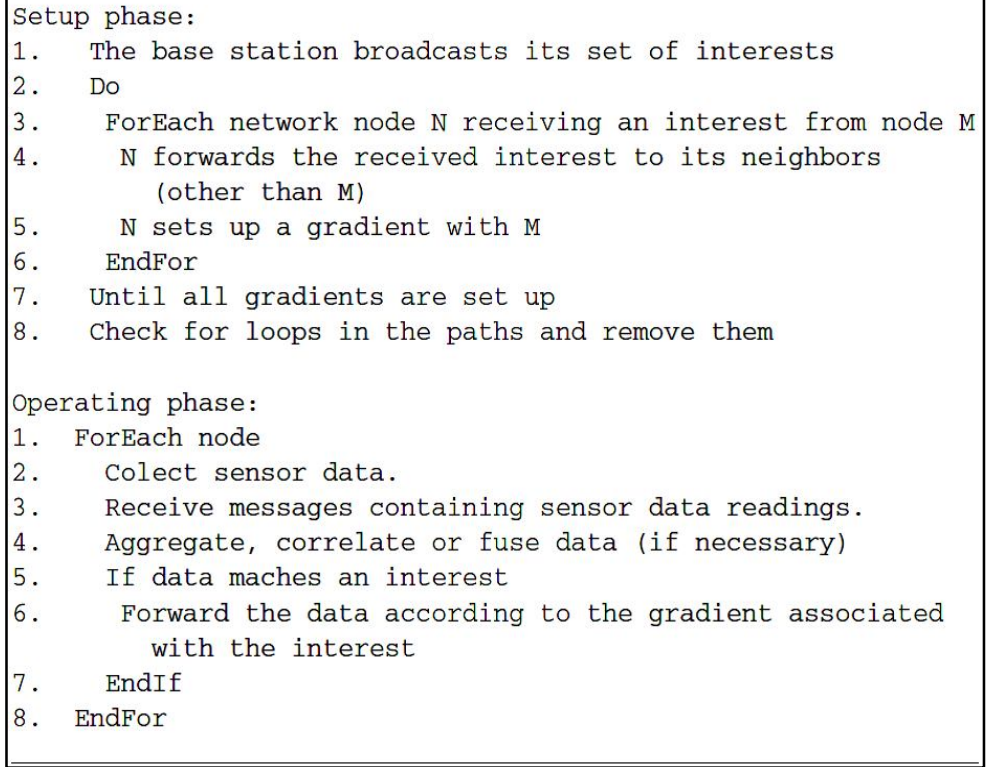
**(2) Eğim kurulumu:** Yerel kurallara dayalı farklı teknikler degrade kurulum için kullanılır. Örneğin, en yüksek kalan enerji ile düğümler eğim kurulumu sırasında seçilebilir. Ağ üzerinden ilgi yayılması sırasında kaynaktan BS'ne geri gradient kurulumunu yapacaktır. Şekildeki 3.5(b) gösterildiği gibi, eğer bir düğümün ilgi iletileri eşleşiyorsa bir kaynak düğüm haline gelir ve kendi verilerini geri BS'ne degrade yoluyla gönderir.

**(3) Güçlendirme:** Degrade kurulum aşamasındaki, alıcı kaynağı doğru birçok yoldan oluşmuştur. Demek ki kaynak verilerini birden çok yolla BS'ne gönderebilir. Ancak, şekildeki gibi 3.5(c) Alıcı, belirli yol boyunca aynı ilgiyi yeniden göndererek belirli bir yolu takviye eder , ve en iyi bağlantı kalitesi , düşük gecikme , komşulardan alınan paket sayısı birçok kurala dayalı seçilir, Bu yolda her düğüm sadece sonraki sekme için güçlendirme iletir [67]. Son olarak çok yönlü teslim sağlamak amacıyla bu aşamada alıcı birden fazla yolu seçilebilir.

**(4) Veri teslimi:** Güçlendirme aşamasından sonra şekilde 3.5(d) gösterildiği gibi kaynak ve alıcı arasındaki yol inşa edilmiştir ve veri iletimi için *Veri teslimi* hazırdır. Yönlendirilmiş dağılım için sözde kod algoritması 4.6[68] bkz.

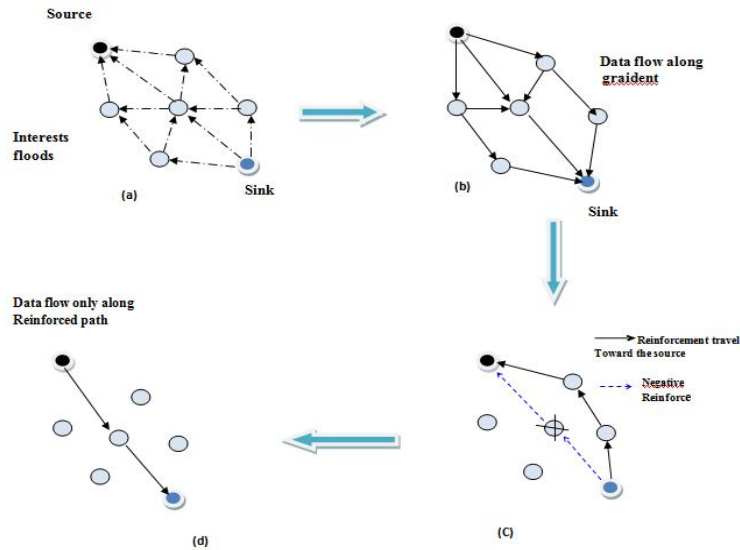


Şekil 4.5 : Directed Diffusion protokolü



Şekil 4.6 : Directed Diffusion protokolünü yöneten sözde-kod

Bazen WSN deki bazı değişiklikler için mevcut veri yolu dinamik olarak değiştirilebilir. Böyle bir durumda alıcı, takviye iletilisini yeni yolu üzerinden gönderir ve mevcut negative takviye Şekilde 4,7 gösterildiği gibidir.



Şekil 4.7 : Negative reinforcement

Bunun sonucunda, DD tarafından aşağıdaki teknik ile karakterize edilmiş diyebiliriz.(Avantajları ve Dezavantajları) :

#### 4.2.1. Directed Diffusion'ın Avantajları

- Tek düğüm veri toplamları almak için tasarlanmıştır.
- Veriler, öznitelik-değer(attribute-value) çiftleri tarafından adlandırılmıştır.
- Bu çok amaçlı sorgu kablosuz algılayıcı ağ ve algılayıcı ağlarda iyi çalışıyor, Örneğin, Belirli alandaki sıcaklık bana ver' yada 'Kim kara ineği gördü'.
- Yönlendirmiş dağılım değer yönlendirme algoritmaları gibi değildir, aynı zamanda birden fazla alıcı sorgular üretir ve veri alır, dolayısıyla eş zamanlı sorguları tek bir ağ içinde elde eder.
- Sonuç olarak, her iki nokta enerji tüketimini azaltmaya katkıda bulunur.
- Kullanıcının ilgilerine dayalı uygulama-bağımlı yollar sağlar.
- Kullanıcının isteklerini karşılamak için, ağ yolları algılayıcı okumalarında ki değişikliklere göre değişir.
- Ne Küresel düğüm adresleme mekanizması ne de bir küresel ağ topolojisi istemez. Ayrıca sadece bir ilgi alanında yolları oluşturulmuştur. Bunun sonucunda enerji verimliliği sağlar.
- İlgi/ Sorgu kaynaklar tarafından değil alıcı tarafından verilir ve sadece istediğinde olur (talep-tabanlı). Ayrıca, tüm iletişimler komşular arasındadır (Gossiping protokolu gibi), adresleme ihtiyacını giderir ve her düğümde veri toplamaya izin verir. Çoğunlukla ilgi için belirli bir yolu seçer. Dolayısıyla ağın enerji tüketimi azalır.
- Eleştirme bilgileri olan düğüm bilgi üretimi söz konusu olan düğümlerdir.

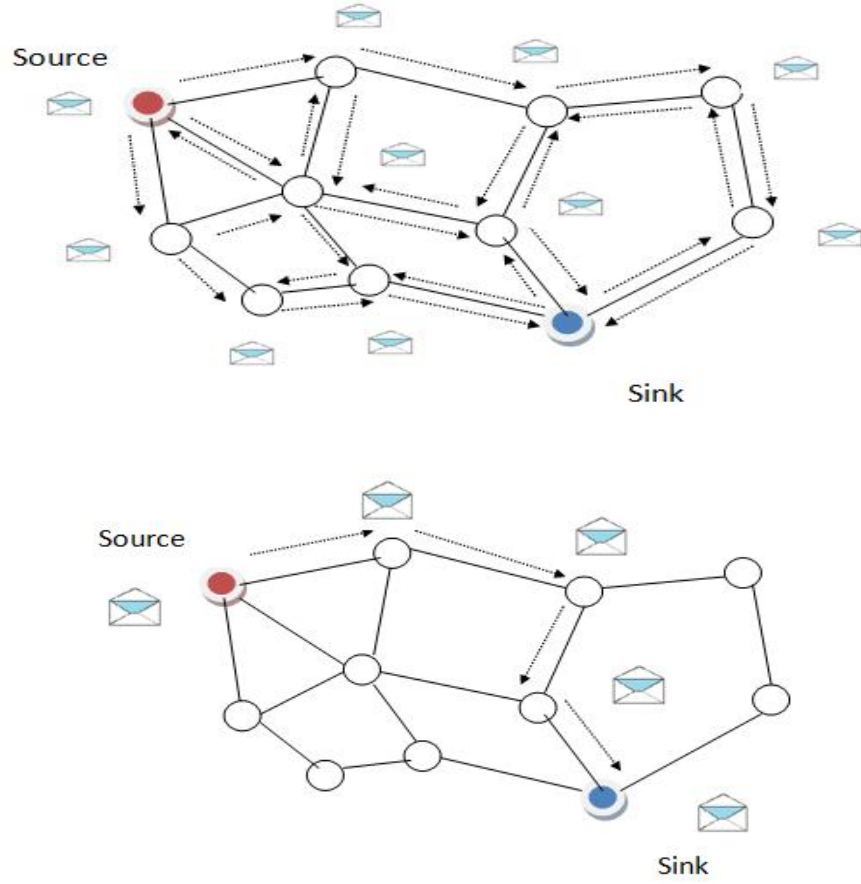
#### 4.2.2 Directed Diffusion'ın Dezavantajları

- Genellikle düz bir topolojiye dayanır. Dolayısıyla, Ölçeklenebilir ve tıkanıklığı (özellikle BS'nin yakınındaki her düğüm) olan sorunlar bulunmaktadır.
- Düğümler arasında ek yük(overhead) problemi ortaya çıkar
- Yönetilmiş Dağılım içinde ilk ilgi iletisi düşük veri hızı içerir. Dolayısıyla, İleti yayılma aşamasında çok sayıda önemli işlem yüklenmesine(overhead) neden olur.

- Her bir düğümde ilgili eğimi kurtarmak için çok sayıda yayılma isteği gerek görülür. Enerji verimliliği ve gerek yüksek miktarda ilgi eğimlei ve alınan iletileri depolamak için bellek optimize edilmiştir.
- Çoğunlukla yolda hızla düğümleri sonlandırmaya neden olan kaynak ve hedef arasındaki en kısa yolu seçermektir, [69].
- Son olarak, Yönlendirilmiş dağılım sorgu tabanlı bir protokoldür. Sürekli veri transferi (dinamik uygulamalar) anında çevresel izleme uygulamaları için gerekli olduğu uygulamada iyi çalışmıyor olabilir.

### 4.3. GOSSIPING PROTOKOLÜ

Komşular arası yayılma [70] veri-aktarma protokolludur, Flooding protokoldaki gibi yönlendirme tabloları ve topoloji bakımı gerektirmez. Flooding geliştirmek ve göçme(impulsion) gibi dizavantajlerini aşmak için üretildi. Flooding'te, düğüm tüm komşularına verileri yayınlar ve alınan düğüm aynı verileri başka bir düğümden alabilir. Veriler Hedef(Destination) tarafından alınana kadar yayın devam eder. Ancak, Gossiping içinde, Bir düğüm packet göndermek için rastgele bir komşu düğümü seçer, bu paketi alan rastgele düğüm, başka bir rasgele komşusu düğümü seçer ve paketi iletir. Bu işlem paket hedefe ulaşana kadar ya da atlama sayısı bitene kadar devam eder. Bunun sonucunda, sadece seçilen düğüm komşuse alıcıyaya alınan paketi iletecektir. Flooding'ten farklı olarak, gossiping protokolu bire-bir(one-one) senaryolarına iyi hizmet eder fakat Flooding bir-çok(one-many) seneryosu hizmet eder. Flooding, hem de Gossiping paket iletimi mekanizması şekil 4,8 gösterilir.



Şekil 4.8 : Flooding ve Gossiping her ikisini yönlendiren mekanizma

Gossiping'n ana amacı güç tüketimini azaltmak ve yönlendirme sistemini mümkün olduğunca basit tutmaktır. Ancak, Verilerin yayılması tarafından gecikmeleri vardır. Gossiping tarafından tüketilen güç, yaklaşım(approximation) aşağıdaki formulda gösterilmiştir[71].

$$O(K^L)$$

*K*: Paket ileten düğümlerinin sayısı

*L*: duraklamalardan önceki sekme sayısı

Gossiping'in en önemli özelliği, uygun K ve L seçerek güç tüketimini kontrol yeteneğidir.

#### 4.3.1 Gossiping Avantajları

- Çok basittir, her hangi bir yönlendirme tablosu ve topoloji bakımı gerektirmez. Bu yüzden çok az enerji tüketir.

- Flooding'in göçme problemini aşacak ortaya çıktı.
- Gossiping'te sadece seçilen düğüm alıcıya verileri göndermede katkıda bulunur.
- Bire- bir iletişim gerektiren uygulamalarda çalışır fakat birçok uygulamalarda çalışmaz.

#### 4.3.2. Gossiping Dezavantajları

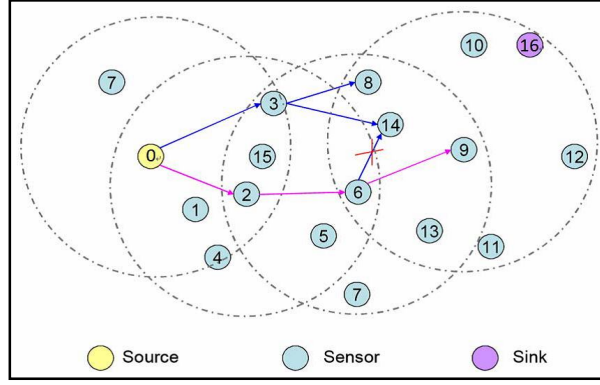
- Paket seçilen düğümden alıcıya' e ulaşana kadar iletilir, bu da paket kaybına sebep olabilir.
- Gossiping'in olağanüstü dezavantajı veri yayılımı nedeniyle gecikme problemi.

Son olarak, Gossiping protokolünü geliştirmek amacıyla birçok Protokol genişletirmiştir. Örneğin; Flossiping, SGDF, LGossiping ve ELGossiping.

***FLOSSIPING protokolü[71]:*** Her iki yaklaşımı Flooding ve Gossiping yönlendirme protokollerini birleştirir. Bir düğüm göndermek üzere bir paketi varsa, bir eşik değerine göre karar verir ve bu değeri paket üst bilgisinde kaydettikten sonra Gossiping mod'ında paket göndermek için rastgele komşu seçer. Başka komşu düğümler ise gelen paketi dinler ve rastgele bir sayı üretir. Komşulardaki oluşturulan rastgele sayılar eşik değerinden daha küçük ise Flooding mod'unda paketi yayınlar. Sonuç olarak, Flossiping, Flooding'teki paket ek yükünü ve Gossiping'teki gecikme sorununu azaltır.

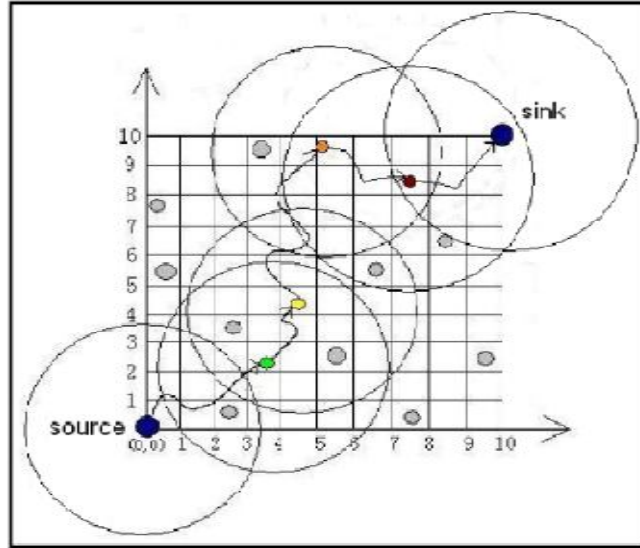
***SGDF Protokolü [72]:*** Tek yönlü Gossiping ile doğrudan Flooding yönlendirme protokolu iki faza ayrılır; Ağ topolojisi başlatma ve yönlendirme aşaması. İlk fazda, her düğüm bir yol ( alıcıya sekme sayısını gösterir) oluşturur. İkinci fazda, Paketi teslim etmek için SGDF Tek yönlü Gossiping ile doğrudan Flooding yönlendirme protokolu kullanır. Sonuç olarak, SGDF yüksek paket teslimat oranıyla düşük ileti karmaşıklığını ve paketlerde kısa gecikmeleri başarmaktadır. Şekil 4,9'a bakınız.





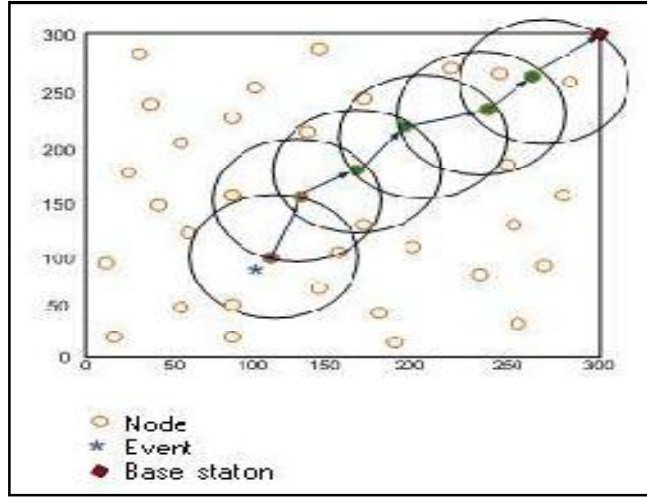
Şekil 4.9 : SGDF Protokolü

**LGossiping Protokolü[73]:** Konum- tabanlı **Gossiping** protokolunda, bir düğüm bir olay yakaladığında, olayı göndermek için rastgele olarak kendi iletim çapında bir komşu düğümü seçer. Komşu düğüm bu olayı alınca da rasgele kendi iletim çapında başka bir düğüm seçer ve olayı gönderir. Bu süreç alıcıya ulaşana kadar devam eder. Sonuç olarak, Gecikme sorunu bir ölçüde çözülür. Şekil 4.10 LGossiping veri yönlendirme şemasını göstermektedir.



Şekil 4.10 : LGossiping veri yönlendirme şematik şeması

**ELGossiping protokolü [74]:** ELGossiping protokolünde, Düğüm bir olay algıladığında ve bu olayı göndermek istediğinde, kendi iletim yarıçapında ve BS/Alicıya en yakın mesafede bir komşu düğüm seçer. Komşu düğüm olayı aldıktan sonara, o da kendi iletim yarıçapında ve BS/Alicıya en yakın mesafede başka bir komşu düğüm seçer. Olay aynı şekilde BS'ne kadar gider. Sonuç olarak, Gecikme ve ulaşma durumu olmayan paketlerin sorununu çözer. Şekil 4.11'e Bkz.



Şekil 4.11: ELGossiping yönlendirmesi

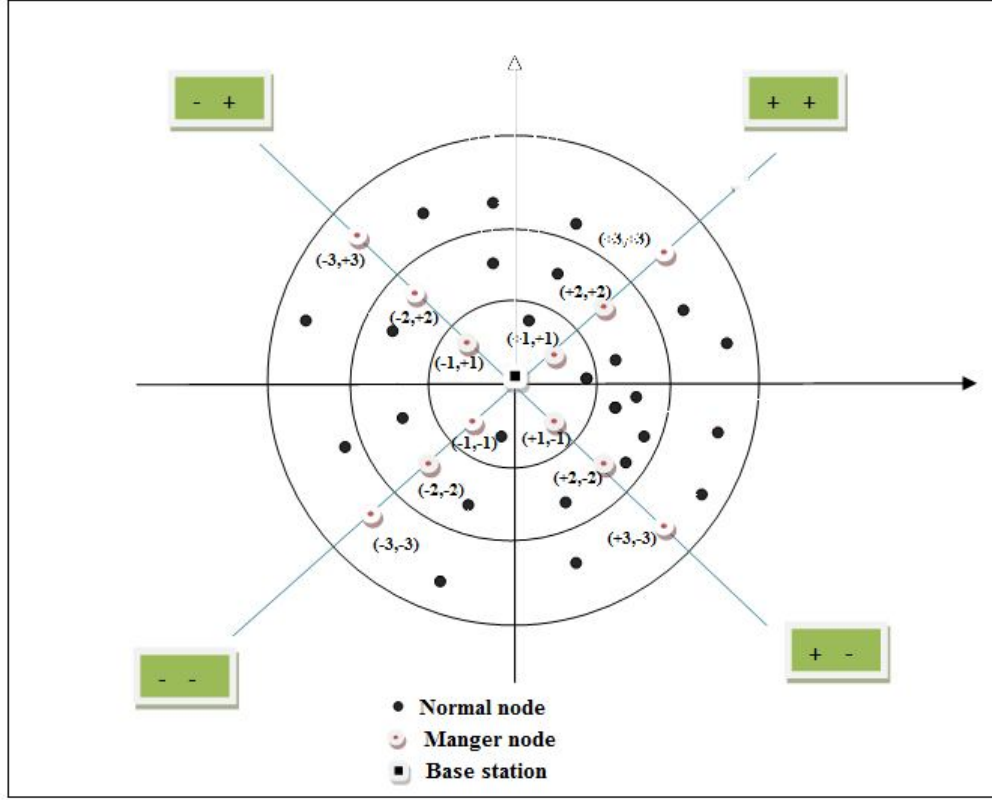
#### 4.4. EESR PROTOKOLÜ

Enerji-tasarrufu sağlayan yönlendirme sensörü(EESR) bir düz yönlendirme algoritmasıdır [75], özellikle veri gecikmesi, enerji tüketiminin azalması ve KSA'nda ölçeklendirilebilirlik için önerildi. Esas olarak Ağ geçidi, Baz İstasyonu, Yönetici düğümler ve sensör düğümlerinden oluşur [75]. Görevleri; **Ağ Geçidi:** Yönetici düğümlerden iletileri teslim alır ya da diğer ağlardan Baz İstasyona gönderir. Baz İstasyonu normal algılayıcı düğümlerinden daha ilave özellikleri vardır. Ağ geçidi'den iletileri alır ve Ağ geçidi'ne iletileri gönderir. Ayrıca, sensör düğümlerinden veri toplamak için sorgu gönderir ve sensör düğümlerden veri alır. Yönetici Düğümler sensör düğümler Ortamdan veri toplar ve Baz istasyona ulaşana kadar 1-atlamada birbirlerine veri gönderir.

Uygulama alanı 2- boyutlu (x,y) koordinatlara göre dört parça koordinatlara (+ +), (+ -), (--), ve (- +) ayrılmıştır ve Baz istasyon koordinat merkezde yer almaktadır. Ayrıca, her koordinat o da sektörlere ayrılmıştır ve Baz istasyonu ortada bulmaktadır. Sayıları, dörtlükte BS'den en uzak noktaya mesajı göndermek için minimum atlama yoluyla hesaplanır. Yönetici düğüm(önceden belirlenmiş),dörtlüğün köşegen hat üzerinde her sektörün merkezinde bulunur ve her biri birbirinden 1-atlama mesafede uzaktır. Sonuç olarak, Diğer düğümler uygulama alanında rasgele olarak dağıtılır; Şekil 4.12 bkz.

Şekilde görüldüğü üzere, her çeyrek dairede üç sektör vardır çünkü Baz istasyonu uzak düğümlerle en çok 3-sekmede(3 hop) iletişim kurabilir. Her sektörün BS'den aldığı ID

kimliği vardır. Dörtlük adı ve BS'den uzaklığına göre belirlenir., Örnek olarak, 1-sekme mesafesindeki sector isimleri (+1+1) sektör, (+1-1)sektör, (-1-1) sektör ve (-1+1) sektör'dır.



Şekil 4.12 : (x,y) 2-boyutlu koordinatlara göre düğüm konumları

Tablo 4,2’de görüldüğü gibi 1-sekme’de “Hello” iletimini yayınlıyarak her bir algılayıcı düğüm EESR tablosunu yapar.

Tablo 4.2 : Çeyrek daire isimleri, Sektör kimliği ID, Yönetici düğüm isimleri (Şekil 4.12'e göre)

BS'den uzaklık mesafesi	Çeyrek daire adı	Sektör kimliği ID	Yönetici düğüm ismi
1 sekme	(+ +)	(+1-1)Sektör	+1 +1M.N
	(+ -)	(+1 -1) Sektör	+1 -1M.N
	(- -)	(-1 -1) Sektör	-1 -1M.N
	(- +)	(-1 -1) Sektör	-1 +1M.N
2 sekme	(+ +)	(+2 +2) Sektör	+2 +2M.N
	(+ -)	(+2 -2) Sektör	+2 -2M.N
	(- -)	(-2 -2) Sektör	-2 -2M.N
	(- +)	(-2 -2) Sektör	-2 +2M.N
3 sekme	(+ +)	(+3 +3) Sektör	+3 +3M.N
	(+ -)	(+3 -3) Sektör	+3 -3M.N
	(- -)	(-3 -3) Sektör	-3 -3M.N
	(- +)	(-3 +3) Sektör	-3 +3M.N

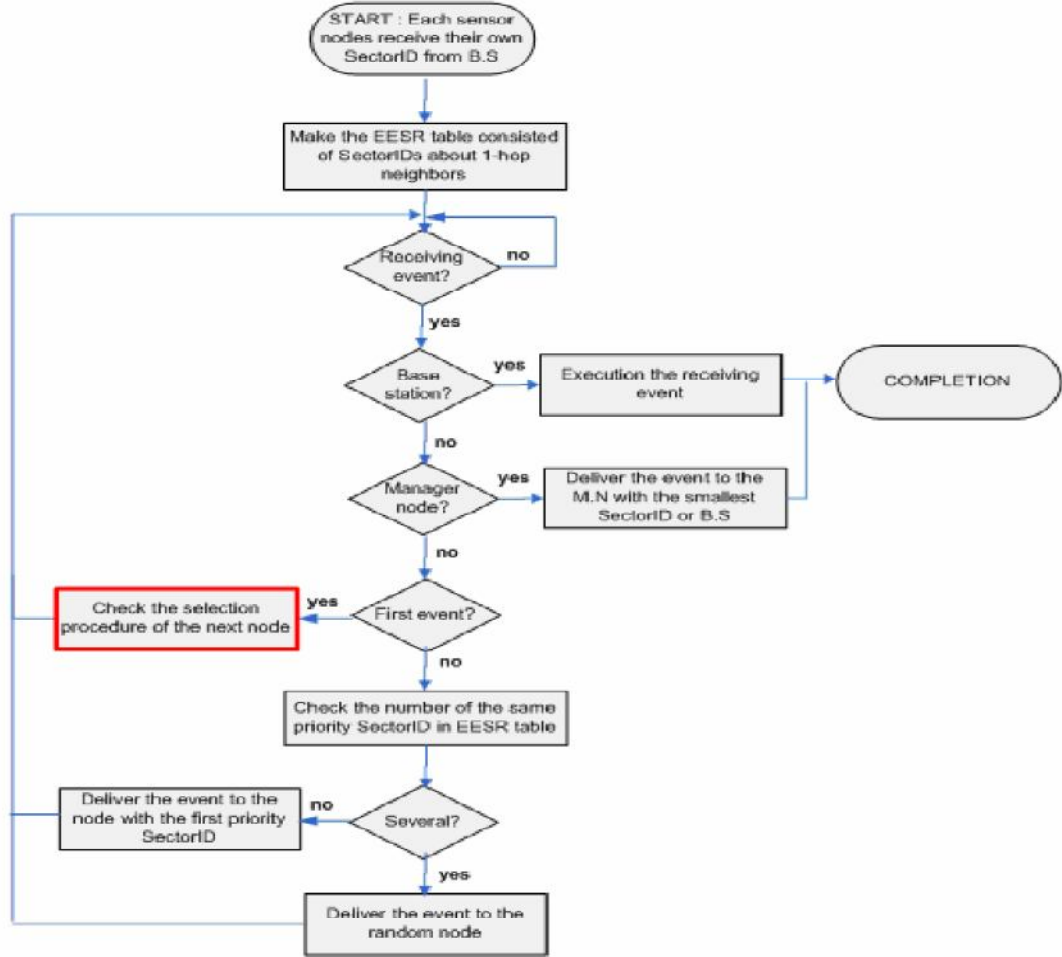
#### 4.4.1. Algoritmanın Çalışması

Düğümleler dağıtıldıktan sonra. Baz İstasyonu göreceli yön bilgileri ve her düğümün ID sektörünü gönderir ve her düğüm kendi EESR tablosunu oluşturur. Düğüm bir olay yakaladıktan sonra. Düğümü olayı teslim etmek üzere bir sonraki düğümü seçmek için EESR tablosundaki 1-sekmeyle tüm komşu düğümlerin sector ID kimliklerini araştırır. Düğüm bir sonraki düğümü seçmek için üç yordamdan birini kullanır

- Yönetici düğüm 1-sekme mesafesinde ise, Diğer sekmeye geçer.
- Yönetici düğüm yok ise ikinci atlamada ki düğümü bulmak için kendisiyle aynı sektörde olan düğümleri ve kendisinden 1-sekme uzaklıkta olan normal düğümü arar.
- Aksi taktirde sektörün dışında yer alan başka bir düğüme bakacak fakat baz istasyona yakın olmalıdır, bu şart altında sonraki atlama düğümü seçer. Aynı çeyrek dairede yatan düğümler tercih edilir.

Sonraki komşu düğüm seçildikten sonra, İlk düğüm olayı yalnızca bu düğüme gönderir. Seçili düğüm olayı alınca sırasıyla ondan sonraki 1-sekme seçmek için aynı yordamı tekrarlar ve olayı gönderir. Olay baz istasyona ulaşana kadar bu işlem devam eder. Ancak, Yönetici düğümü bir olay alırsa, olay baz istasyona ulaşana kadar yönetici-yönetici(manger-manger) işleminde devam eder. Şekil 4.13'de gösterildiği gibi akış şemasını görürüz [75].

İlk olay alındıktan sonra, sonraki sekme (sonraki komşu düğüm) önce veri gönderilen bu düğümleri için kolayca seçilecektir. Her bir düğüm EESR tablousunu depolayarak sektör ID kimliğine göre bir sonraki düğümün seçim prosedürünün(yukarıda bahsedilen üç prosedürden) sonucu olur ve sonra veriyi gönderir.



Şekil 4.13: EESR düzen prosedürü

#### 4.4.2. EESR' in Avantajları

- Uygulama alanını sektörlere böler, bu nedenle ölçeklenebilen sayılır.
- Enerji-tasarrufu sağlar ve üç yolda bu özelliği elde eder; İlki, olayı sadece bir düğüme gönderir, yani taşma(flood)etmez, İkincisi, Yöneticisi Düğümler, önceden tanımlanmış bir kısa yolda (shortest path) veri gönderir. Üçüncüsü, normal düğümler ilk olayı gönderdikten sonra kendi EESR tablolarını kullanarak kolayca sonraki düğümü seçer. Sonuç olarak, Daha az enerji tüketilir ve ağ ömrü uzar.

- Bire-bir iletişimi vardır. Ayrıca, İlk olayı gönderdikten sonra sonraki atlama kolayca bulunabilir. Sonuç olarak, Gecikme süresi düşer.
- Aynı yol üzerinden veri göndermemek ve enerjiyi tüketmemek amacıyla, bazı durumlarda veri göndermek için diğer yolları seçer.

#### 4.4.3. EESR'nin Dezavantajları

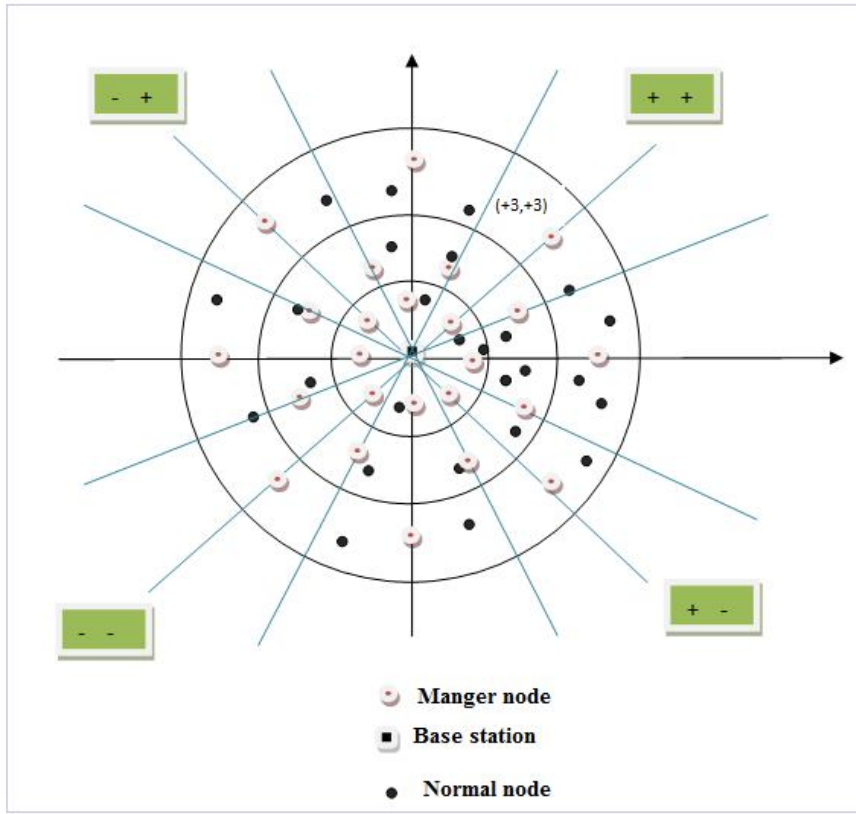
- Olay yakalayan düğüm sonraki düğümü seçmek için kendisine 1-sekmede uzaklıkta olan tüm düğümler aktarım aralığının dışında olabilir. Yani sonraki düğümü seçmek için belirli ölçütlere sahip değildir[76].
- Uzak sektörde bulunan bir olay tespit ederse ve sonraki atlama alt sektöründe yer alırsa. Alçak düğüm enerji durumuna bakarak veri kaybolabilir.
- Uzak sektörde normal düğümler bir olay yakalarsa ve her zaman sonraki sekme yönetici düğümler ise, Yönetici düğümlerin enerjisi erkenden tüketilecektir. Çünkü onlar Baz istasyona kadar olayı yönetici yöneticiye gönderir.
- Enerji tüketimi dengede değildir, bazı düğümlerin başka düğümlerden önce enerjisi tüketirler.

bu protokolün dezavantajlarını gidermek için bunun üzerinde çalışmalar olmuştur [76], birisi adı geçen: Kablosuz algılayıcı ağlarda ortak merkezli sektörler oluşturarak yeni optimal yönlendirme algoritması

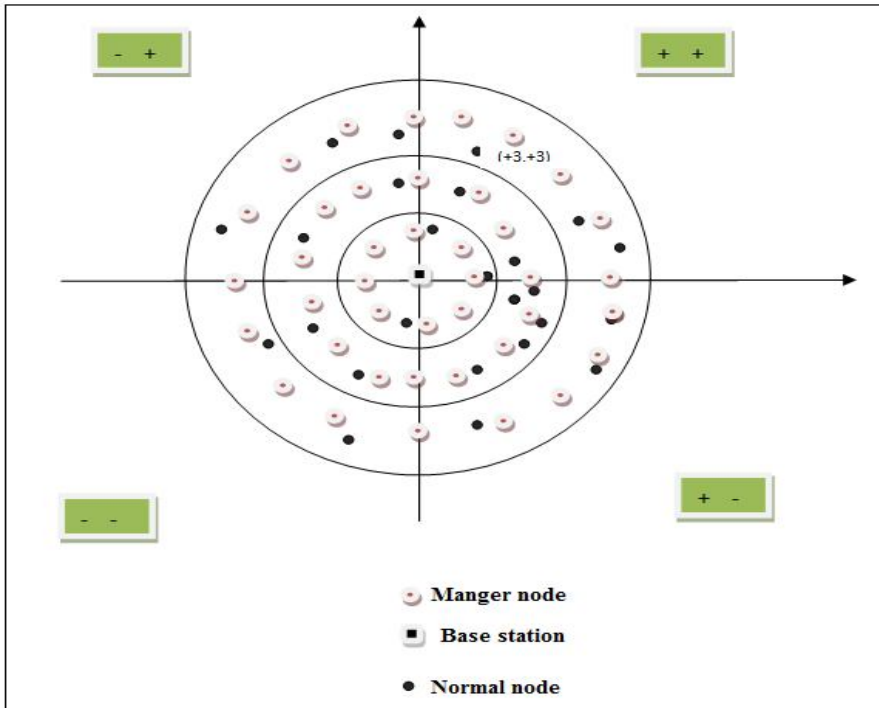
**Kablosuz algılayıcı ağlarda ortak merkezli sektörler oluşturarak yeni optimal yönlendirme algoritması [76]:** Bu çalışma yukarıda bahsedilen EESR'nin zayıflığını aşmayı teklif eder. Yeni protokolde iki çözüm önerirler:

**İlk Çözüm: Şekil 4.14** gösterilen gibi, her çeyrek çemberde anayol(köşegen) sayılarını artırıyor. Bu çözümde ikinci ve üçüncü problem (EESR'nin dezavantajları) çözüldü.. Ancak, İlk sorun çözülmesi gereken en önemli konudur.

**İkinci Çözüm (En uygun olanıdır):** Anayola(köşegen) bakmıyarak, her sektörde eşit bir şekilde yönetici düğümlerin sayılarını hesaplamak. Şekil 4,15 Bkz. Bu çözüm ilk ve son sorunuda çözdü. Daha ayrıntılı bilgi için bkz: [76].

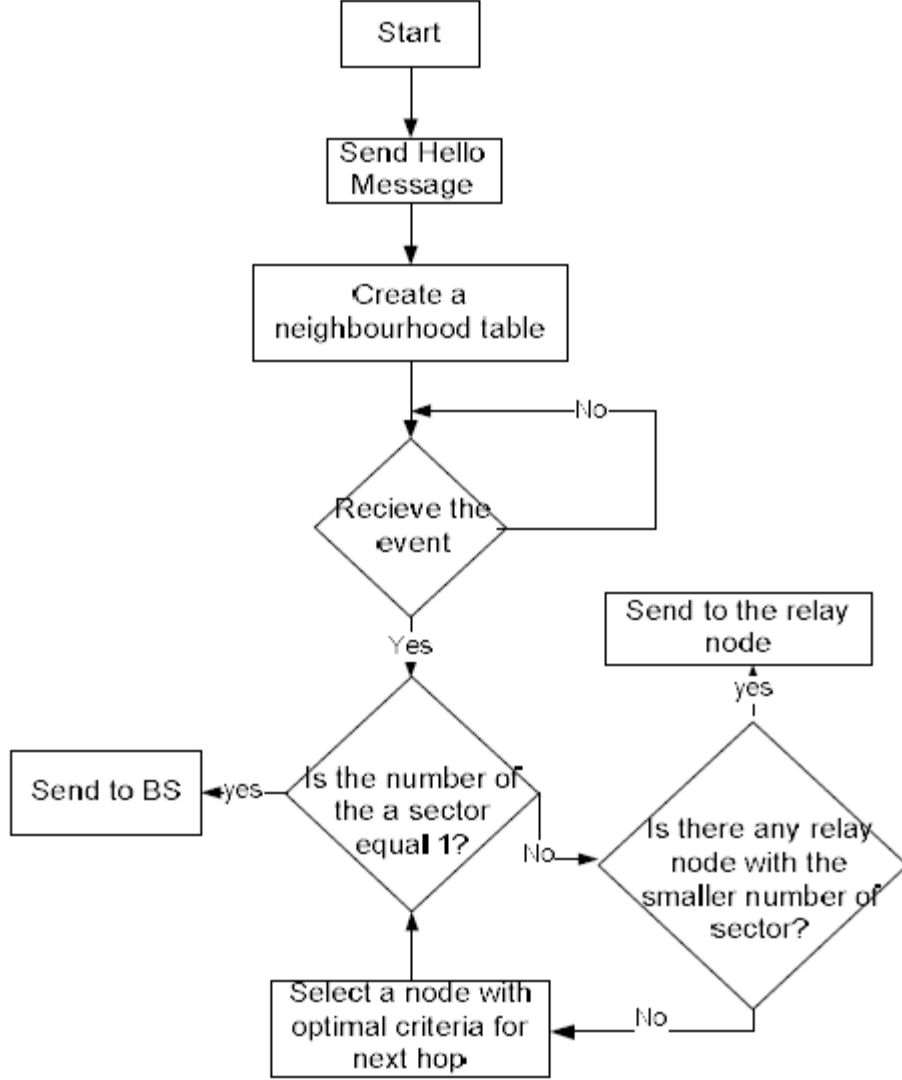


Şekil 4.14: EESR probleminin ilk çözümçü



Şekil 4.15: EESR probleminin ikinci çözümü

Bu iletişim geliştirme protokolü için yönlendirme aşağıdaki Şekilde gösterilen akış sürecidir.4,15 [76].



Şekil 4.16 : EESR gelişimi yönlendirme yöntemi sunan algoritma.

#### 4.5. YÖNLENDİRME PROTOKOLLERİNİN KARŞILAŞTIRILMASI

Bu araştırma sırasında, çok farklılıklar gözlenmiştir, genellikle, düz ve hiyerarşik yönlendirme protokolleri arasında araştırma olmuştur. Gossiping protokolleri diğer protokolleri ile karşılaştırıldığında çok basittir, her hangi bir yönlendirme tablosu ve topoloji yöntemi gerektirmez. Çok yüksek bağlantırlık(connectivity) sağlar, burada, en kısa sürede bir düğüm paketi iletmek için komşularından haberdar olursa. Gossiping



protokolu flooding protokoluna dayanmaktadır. Her paketi tüm komşularına yayınlamak yerine rastgele seçilerek tek bir komşu düğümüne gönderilecektir, bunun anlamı da şudur, her hangi bir zamanda sadece paketin bir kopyası iletimde bulunur.

Paketi almış olan komşu düğüm iletiyi göndermek için başka bir rastgele düğüm seçer. Bu işlem paket hedefe ulaşana kadar ya da paketin maksimum atlama sayısını geçene kadar devam eder. Sonuç olarak, Leach, Directed Diffusion ve EESR Protokoller ile karşılaştırıldığında, Gossiping orta bir miktarda güç kullanır. Gossiping diğer protokolle karşılaştırıldığında, oldukça yüksek gecikme harcar çünkü ağ üzerinde veri yaymak için bir bire iletim kullanır ve düğümleri rastgele seçimden kaynaklanan atlama sayısı çok büyük olabilir. Ağ üzerinde düğüm sayısı artınca, paketin iletim yollarının sayısı artar. Ağın artmasını geçmek için atlama ortalama sayısı alınır. Bu nedenle, paketlerin atlama sayısı maksimum değere ulaştığı zaman paketler düşürülür. Geniş ağlarda paketlerin atlama sayısı maksimum değere ulaşması büyük ihtimale alınır ve bir çok paket kayb olur. Daha Küçük ağlarda gönderilen paketlerin kabaca yarısı kayıp olur ve geniş ağlarda kayb oranı büyük ölçüde artar.

Sonuç olarak, Gossiping protokolu paket kaybolma açısından en kötü protokoldür. Dolayısıyla, Gossiping, Leach, Directed diffusion ve EESR gibi ölçeklenebilirlik değildir. Sonuç olarak, tüm yukarıda bahsedilen iki tabloda özetlenebilir; tablo 4.4[9], düz ve hiyerarşik duyurga ağlar için farklı yönlendirme yaklaşımların genel bir karşılaştırma gösterir.

Tablo 4.5'te nasıl yönlendirme protokolleri (Leach, Yönetmen Difüzyon, EESR ve Gossiping) araştırması farklı kategori altında uygun olduğunu gösterir ve birçok ölçüme göre bu yönlendirme teknikleri karşılaştırır.

Tablo 4.3: Düz ve Hiyerarşik protokolleri arasındaki Genel Karşılaştırma

<b>Hiyerarşik Yönlendirme</b>	<b>Düz Yönlendirme</b>
Rezervasyon tabanlı zaman çizelgelemesi	Çekişme tabanlı zaman çizelgelemesi
Çatışmaların önlenmesi	Çarpışma mevcuttur
Düzenli uyku nedeniyle duty cycle hacmi azalır	Düğümün uyku zamanını kontrol ederek değişken duty cycle yaratır
Küme başı tarafından veri toplama	Çoklu-Sekme yolundaki düğümler komşulardan gelen verileri toplar
Basit ama optimal olmayan yönlendirme	En uygun yönlendirme fakat ek karmaşıklık vardır
Global ve yerel eşzamanlama gerektirir	Linkler senkronizasyon olmadan anında oluşur
Ağ boyunca küme oluşumunu ek yükü(overhead)	Sadece veri iletim bölgede yollar oluşur

Küme başları tarafından düşük gecikme olur	Gecikme ara düğümlerini ve bir çok yol oluşturmada ortaya çıkar
Enerji dağılımı düzenlidir	Enerji dağılımı trafik modülleri üzerine bağlıdır
Enerji dağılımı denetlenemez	Enerji dağılımı trafik düzenine uyum sağlar
Doğru kanallar tahsisi	Garantilenemeyen kanal yerleşimi

Tablo 4.4 : LEACH, Directed Diffusion, EESR ve Gossiping protokolleri arasında karşılaştırma

	LEACH	Directed diffusion	Gossiping	EESR
Snıf	İlyerarşik	Düz	Düz	Düz
Ölçektenebilirlik	Yüksek (Düğümleri Kümeler halinde böler )	Sınırlı (Düz topoloji doğası nedeniyle )	Sınırlı (Düz topoloji doğası nedeniyle )	Yüksek (Uygulama alanını Sektörlere böler )
Yaşam Süresi	Çok iyi (TDMA nedeniyle çoğu algılayıcılar uyku modundadır ve diğer düğümlere CH görevi dağıtır)	İyi (İstek tabanlı ve komşular arası iletişimi )	Orta (Son derece yüksek gecikme)	Çok iyi (Direkt olarak MN yerleşimi önceden belirlenmiştir ve sektör kullanılır)
Enerji Dağıtım	Yüksek (Küme başı ile düğümleri tek-sekmede yönlendirir)	Yüksek (Çoğunlukta ilgi için belirli bir yol seçer ve yollar ilgi olduğu zaman şekillenir )	Orta (LEACH ve EESR den farkı olarak yönlendirme tablosu ve topoloji bakımında yoktur)	Yüksek (Sadece 1-sekme komşulara gönderir ve veri göndermeye farklı yollarla seçer )
Veri Toplama	EVET (Veriler CH'lar tarafından toplanır)	EVET (Her düğüm veri toplar ve sonraki aklamaya gönderir)	HAYIR (Veri iletimine katılan düğümler sadece role/geçiş istasyondur)	EVET
Müzakere-tabanlı	HAYIR Sinyal gücüne göre	EVET (Müzakere Gradient kurulum aşamasında yapılır))	HAYIR (Veri göndermek için komşu düğüm rasgele olarak seçilir)	EVET (EESR tablosu oluşturarak en iyi yol seçilir)
Kaynak Haberdar	EVET	EVET	HAYIR	HAYIR
Sekme İletişimi	Tek-sekme üye- CH ve CH-BS	Çoklu-sekme (kaynaktan BS'a diğer düğümler üzerinden şekillenir)	Çoklu-sekme (kaynaktan BS'a diğer düğümler üzerinden şekillenir)	Çoklu- sekme (B Sulaşana kadar her zaman 1- sekme atlar)

Optimal Yol	HAYIR Üyeler ve CH değişir	EVET (takviye aşaması)	HAYIR (Rastgele seçim)	EVET (Her zaman daha iyi sektör kimliği bulmak için EESR tablosunu kontrol eder)
Gecikme Süresi	Düşük Düğüm doğrudan CH veri gönderir with TDMA schedule	Yüksek (Flooding yoluyla ilgi yayılması)	Çok Yüksek (Veri yayılması nedeniyle)	Düşük (Yüksekte değildir ve ilk olaydan sonar ikinci- sekme kolayca bulunabilir)
Üretilen İş (Throughput )	Çok Yüksek Gecikme yoktur vesadece bir düğüm değiştiği zaman	Uygun	Düşük (Yüksek gecikme nedeniyle)	Yüksek (En iyi yol seçilir, over-head yok ,Gecikme yok)
Ek Yük (over-head )	HAYIR CH birçok düğümlerin verilerini toplar	EVET (Veri işlemleri ve sorgu eşleştirme sırasınca over-head bulunur)	HAYIR (Komşuya doğrudan gönderilir)	HAYIR (Bire bir iletişim)
-Tabanlı	Olay tabanlı Sadece bir olay oluştuğunda algılayıcılar onu yakalar	Sorgu tabanlı (Sorgu arayışı sadece talabta olduğu zaman kullanıcılar tarafından verilir)	Olay Tabanlı Sadece bir olay oluştuğunda algılayıcılar algılar	Olay Tabanlı Sadece bir olay oluştuğunda algılayıcılar algılar
Uygulamalar	İzleme Uygulaması (Dinamik Uygulama.) Bir olay çıkarsa düğüm onu tespit eder	Çok amaçlı Uygulama (Aynı anda birden fazla alıcı veri sorgular ve teslim eder)	Uygulama İhtiyacı (Bire bir iletişim)	İzleme Uygulaması (Dinamik Uygulama.)
Uygulama çeşitleri	Sağlıklı izleme (Yapay retina)	Çevresel izleme (PODS Hawaii)	Çevresel izleme	Çevresel izleme, i.e., Ev Otomasyonu Ağlar

Yukarıda bahsedildiği ve Tablo 4,5'te görüleceği üzere, Gossiping basitliğine rağmen LEACH, Directed Diffusion ve EESR ile kıyaslandığında birçok dezavantajı bulunmaktadır. Bununla ilgili olarak Gossiping Protokolünün eksikliklerinin üstesinden gelmek için birçok yeni protokol öne sürülmüştür. Ancak, bunlardan hiçbiri Gossiping protokolünün sorunlarına tam olarak çözüm bulamamaktadır. Örneğin, Flooding ve Gossiping protokollerini bir araya getiren FLOSSIPNG protokolü, flooding'in ek yük sorununu ve Gossiping'in gecikme sorununu çözmekte ama enerji tüketimi ve paket gecikme süresi problemleri Flooding ve Gossiping yönlendirme protokolleri ile aynı durumdadır. SGDF yüksek paket iletim oranını, düşük mesaj karmaşıklığını ve kısa paket gecikmesini başarmaktadır. Gel gelelim, doğrudan paket gönderimi nedeniyle, paket gönderimi süresince paket miktarı daha büyük hale gelmektedir. Lgossiping gecikme problemini bir dereceye kadar çözerken, Baz İstasyonundaki ulaşamayan paketlerin problemi halen devam etmektedir. Ayrıca, bu protokol (Lgossiping) her bir düğümün konumunu belirlemek için GPS kullanmaktadır. Bu yüzden, ilave donanım ilave masraf anlamına gelmektedir. Sonuç olarak Lgossiping'i iyileştirmek amacıyla ortaya konan ELgossiping, ağ yaşam süresini iyileştirmiş ve paketlerin Baz İstasyonuna iletiminde yaşanan gecikmeyi bir dereceye kadar çözmüştür. Oysa alıcıdaki ulaşamayan paketlerin problemi hâlâ devam etmektedir.

Sonuç olarak, bu çözümlerin hiçbiri Gossiping problemlerini tam olarak çözmemektedir. Bu nedenle, bu bölümün diğer kısmında ELGossiping'i geliştiren yeni bir protokol öne süreceğiz. Öne süreceğimiz bu yeni protokolde, paket yayılmasının neden olduğu gecikmeyi azaltmaya, ulaşamayan paketler problemini çözmeye, ELGossiping ile çalışma tarzını aklımızda tutarak ağ yaşam süresini artırmaya çalışacağız.

## 5.FELGOSSIPING PROTOKOLÜ

Bu bölümde Gossiping problemlerine ve onun uzantılarına çözüm bulmak amacıyla Düzenli Verimli Konum-tabanlı Gossiping protokolünü(FELGossiping) ortaya koyacağız. FELGossiping üç fazdan oluşmaktadır; Başlangıç Fazı (inialization) ,Bilgi toplama fazı (Information Gathering) ve Yönlendirme Fazı (Routing). İlk fazda her bir düğüm alıcı/baz istasyonuna olan yol (gradient) üretmektedir. İkinci fazda ise, gradient kurulduktan sonra, olayı yakalayan düğüm kendi iletim yarıçapında komşu düğümlere bilgi istek mesajını gönderir, bu bilgi mesajın içinde atlama sayısı ve artakalan enerjilerini ister, komşu düğümler sonra atlama sayısı ve artakalan enerjisini olayı yakalayan düğüme gönderir.

Üçüncü fazda, kaynak düğüm atlama sayısını ve artakalan enerjiyi bildikten sonra alıcıya doğru atlama sayısına göre iki düğüm seçer, alıcıya en yakın olan düğümleri seçer, iki düğüm de aynı atlama sayısı olan ya da iki düğümün arasındaki fark 1 olur.

İki düğüm seçildikten sonra, bu algoritma iki düğümün arasındaki artakalan enerjiyi karşılaştırır. Atlama sayısına bakmadan fazla olan artakalan enerjili( residual energy) düğümü seçer. Paketi alıcıya göndermek için seçilen düğüm sonraki atlama olur .

Bu seçilen düğüm aynı iki fazdan geçerek, aynı kaynak düğümün işlemlerini yapar. Paket baz istasyona/alıcıya ulaşana kadar bu işlem devam eder.

Bu protokolü başka protokolarla karşılaştırarak enerjiyi dengeli(balance) ve düzenli(fairli) kullanarak ağda enerji tasarrufunu azalmasını göreceğiz, sonuç olarak ağ yaşam süresini uzatmış oluruz.

Ayrıca yayılım gecikmesini ve paket kaybını azaltmasını göreceğiz. Aşağıda bu üç faz daha detaylı açıklanacaktır.

Aşağıda bazı varsayımlarımız bulunmaktadır:

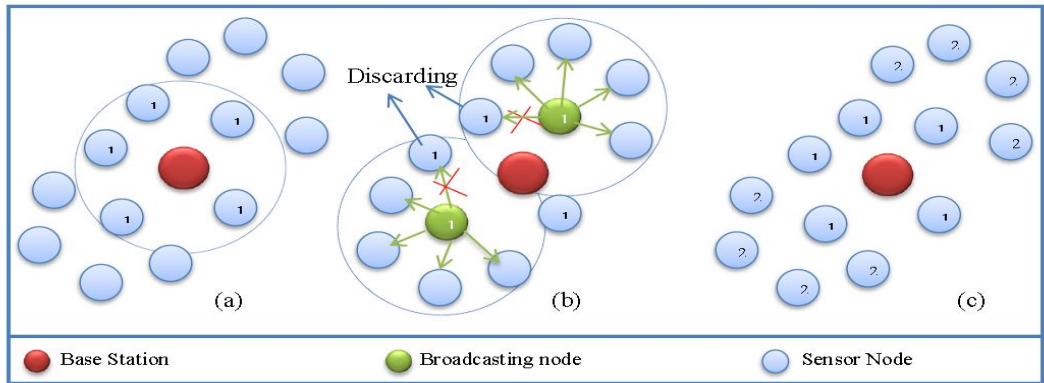
- Başlangıçta tüm düğümler tam enerji doludur ve hepsinde aynı miktarda enerji vardır.
- Her bir düğümün 40 metrelik iletim yarıçapı vardır.
- Her düğüm de GPS sistemi bulunur. Her düğüm kendi yerini tespit eder ve baz istasyondan ne kadar uzak olduğunu tespit eder.

## 5.1. AĞ BAŞLATMA FAZI

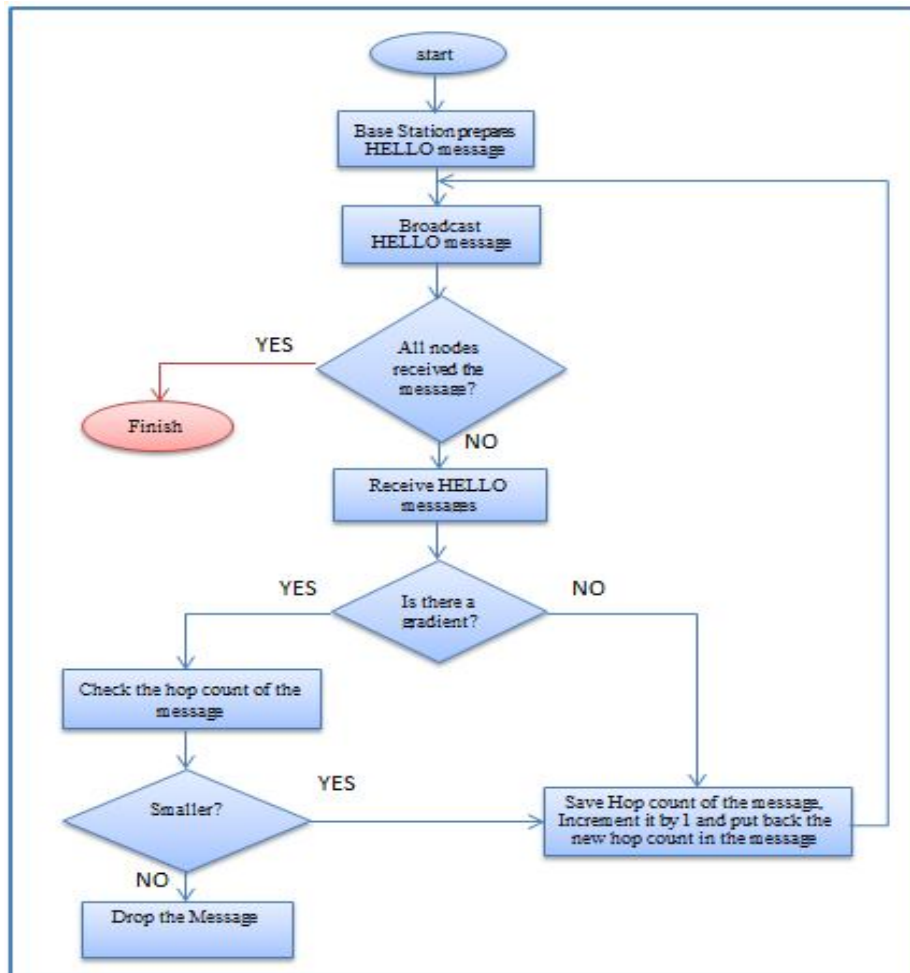
Ağ başlatma fazı, sensor düğümleri uygulama alanına rastgele dağıtıldıktan sonra başlar. En başta, baz istasyonu komşularına bir “MERHABA” mesajı yayınlar. MERHABA mesajı şunları içerir: Baz istasyonunun Adresi (sabit) ve Sekme sayısı (değişken). Sekme sayısı Baz İstasyonuna olan irtifayı ayarlamak için kullanılır, yani, düğümün Baz İstasyonuna olan uzaklığını gösterir. MERHABA mesajı yayımlandıktan sonra tüm 1-sekme komşular bu mesajı alır ve Baz istasyonunun adresini ve sekme sayısını elde ederler. Her bir düğüm sekme sayısını hafızasına kaydeder ve sekme sayısını bir artırır. Sonra yeni sekme sayısı eskisi ile değiştirilir. Bundan sonra MERHABA mesajını alan her düğüm bu mesajı daha uzak düğümlere yayınlamaya devam eder. Şekil 5.1’de de gösterildiği üzere, her bir aşamada sekme sayısı 1 artırılır.

Bir düğüm MERHABA mesajını aldığı anda, hâlihazırda bir irtifa değerinin olup olmadığını kontrol eder. Eğer varsa, mesajın sekme sayısı ile kendisinininkini kıyaslar ve eğer mesajın sekme sayısı kendisinininkinden daha küçükse sekme sayısına bir ekler ve onu yayınlar, ama eğer kendisinininkinin sekme sayısı mesajinkinden daha küçükse veya eşitse mesajı göz ardı eder. Bu durum mesajın düğümlere farklı rotalardan ulaştığı mesaj yayınlama durumlarında meydana gelir. Sonuç olarak irtifa en iyi rotayı saklar. Son olarak, tüm sensorlar MERHABA mesajını almaya dek süreç devam eder, o anda ağ başlatma fazı tamamlanmış olur. Şimdi her düğüm Baz İstasyonuna olan uzaklığını irtifa aracılığıyla bilmektedir.

Şekil 5,1’de MERHABA mesajının ağ boyunca nasıl yayımlandığını göstermektedir. Şekil 5,1(a)’da Baz İstasyonuna 1-sekme komşular sekme sayısını hafızalarında tutacak ve irtifalarını oluşturmaya başlayacaklardır. Her bir düğüm mesajdaki sekme sayısını 1 artıracak ve mesajı yeni sekme sayısı ile yayınlacaktır. Şekil 5,1(b)’de MERHABA mesajının 1-sekme komşulara ulaştığını, bunlardan birinin yayıncı düğüm haline geldiğini ve mesaj yayınlamaya sırayla devam ettiğini görüyoruz. Daha sonra yayıncı düğümün komşuları mesajı alır ve mesajın sekme sayısını kendilerinininki ile kıyaslarlar. Eğer kendilerinininki daha küçükse Şekil 5,1(b)’de gösterildiği üzere mesajı göz ardı ederler. Süreç başlangıç fazı tamamlanmaya kadar Şekil 5,1(c)’de gösterildiği gibi devam eder. Ağ başlatma fazı algoritması şekil 5,2’de gösterilmiştir.



Şekil 5.1 : Ağ Başlatma Fazı



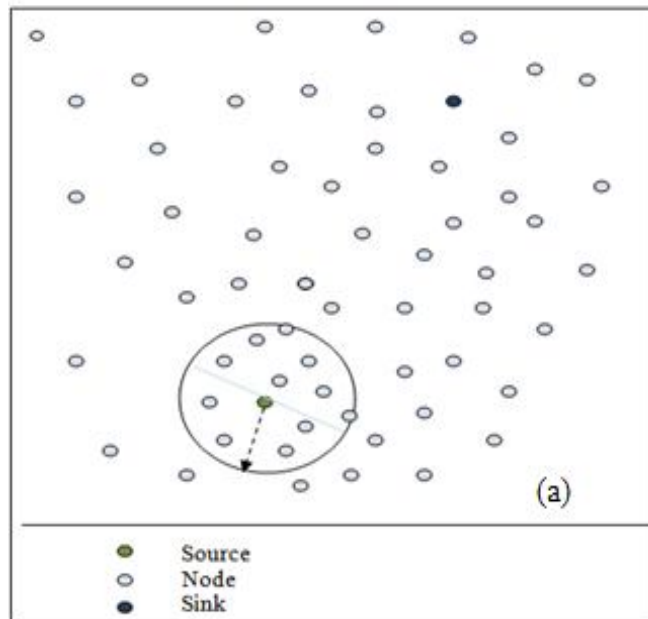
Şekil 5.2: Ağ Başlatma Fazı algoritması

## 5.2. BİLGİ TOPLAMA FAZI

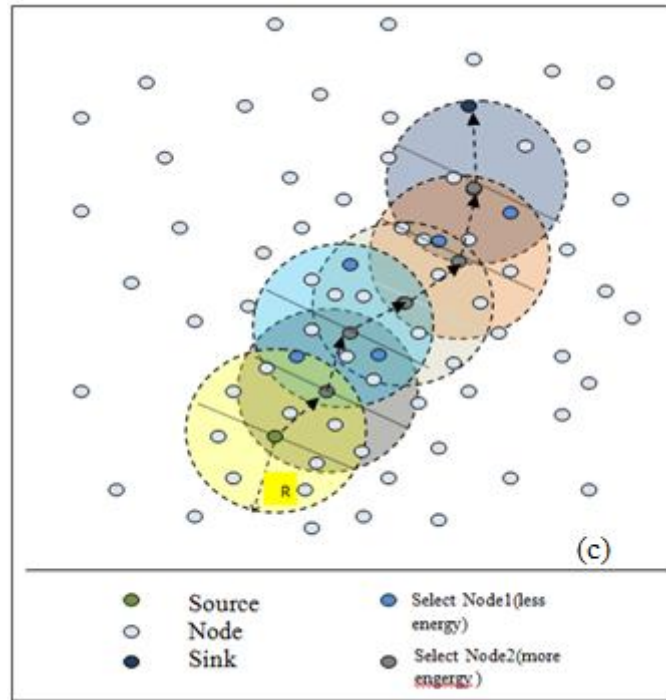
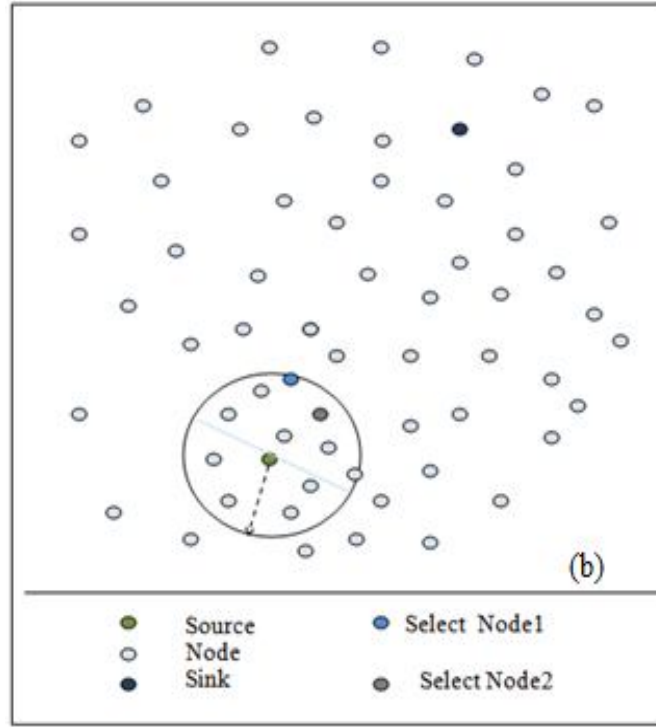
Kaynak düğüm olayı yakaladıktan sonra, 40 m yarıçapında çizgi çizer, bu alanda komşu düğümlere istek(request) mesajı gönderir ve komşu düğümlerden sekme-sayısın ve artakalan enerjiyi ister. Kaynak düğüm istek mesajı gönderdikten sonra, mesajı alan düğüm kaynak düğüme alıcıya doğru sekme-sayısın ve artakalan enerjisini gönderecektir.

## 5.3. YÖNLENDİRME FAZI

Bilgi toplama fazı tamamlandıktan sonra, yönlendirme fazı başlar. Komşu düğümler istek mesajına cevap verdikten sonra, kaynak düğüm baz istasyona yakın olan iki düğüm seçer, bu düğümleri sekme sayısına göre seçer. İki düğümün de sekme sayısı aynı olabilir yada fark bir olabilir. Daha sonra iki düğümün sekme sayısına bakmayarak, iki düğümün artakalan enerjilerine bakarak , hangisi yüksek artakalan enerjiye sahip ise ,o düğüm sonraki sekme olarak seçilir ve o düğüme mesaj iletilir. Eğer iki düğüm de aynı artakalan enerjiye sahipse, baz İstasyona doğru en az sekme sayısı olan düğüm seçilir. Kaynak düğüm mesajı ilettikten sonra. Mesajı alan düğüm aynı kaynak düğümün işlemlerini yapar, iki fazdan(bilgi toplama fazı ve yönlendirme fazı) da geçer. Mesaj baz istasyona ulaşana kadar yada TTL sıfır olana kadar işlem devam eder. İki fazdaki işlemi aşağıda ki şekil 5.3(a,b,c)'de gösterilmiştir.







Şekil 5.3 : FELGossiping'te yönlendirme fazı

Bir düğüm bir olay saptadığında ve bunu baz istasyonuna göndermek istediğinde bir paket üretir ve bu paketin üstbilgisi aşağıda gösterilen altı alanı içermektedir:

<i>Kaynak</i>	Hedef	Şimdiki	<i>Sıradaki</i>	<i>Sekme</i>	<i>TTL</i>
---------------	-------	---------	-----------------	--------------	------------

### Header Paketi

**Kaynak:** Kaynak düğümün (olay düğümünün) kaynak adresi.

**Hedef:** Baz İstasyonunun adresi.

**Şimdiki:** Mevcut düğümün adresi.

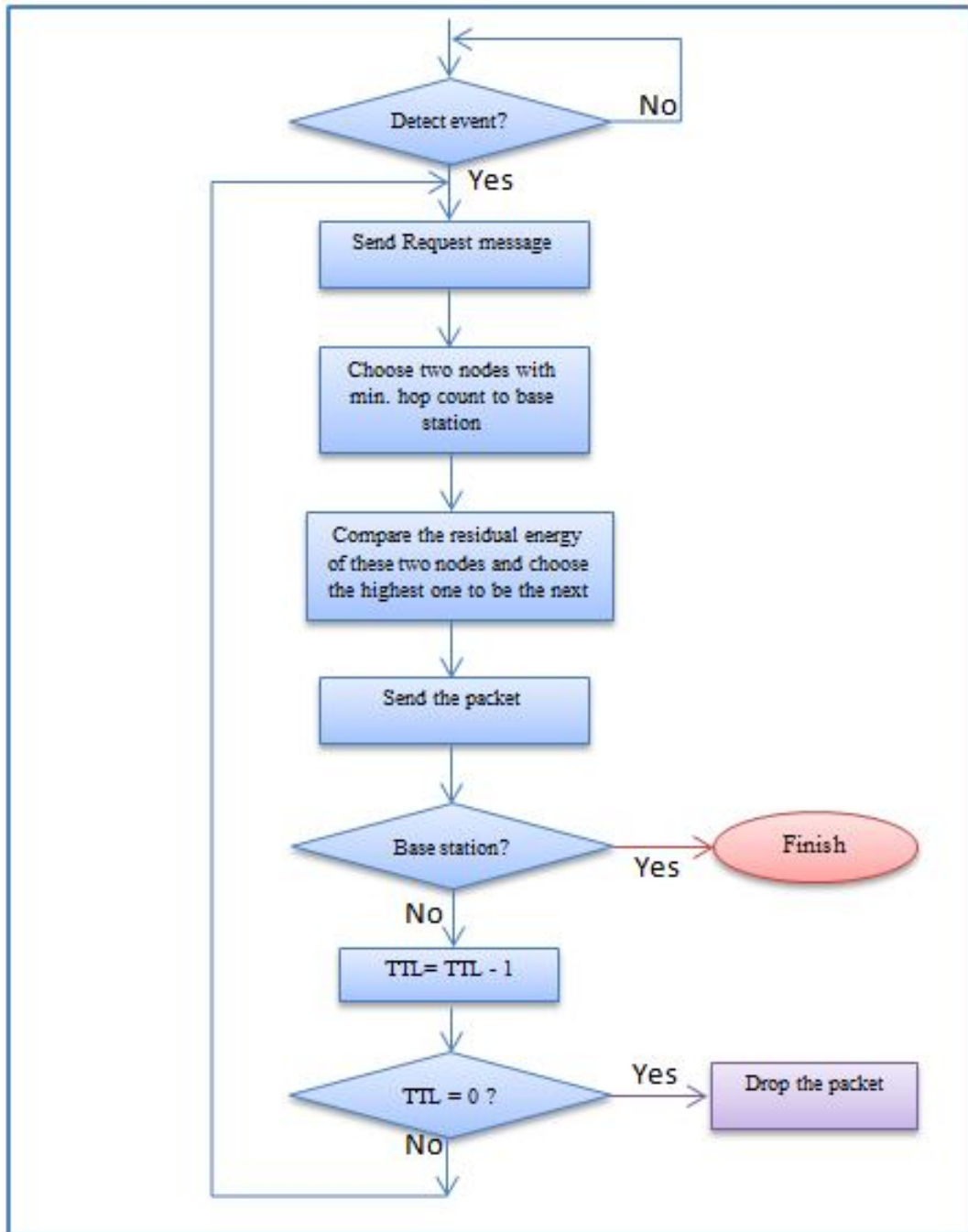
**Sıradaki:** Sıradaki sekmenin adresi.

**Sekme:** Mevcut düğümün Baz İstasyonuna olan sekme sayısı (numara).

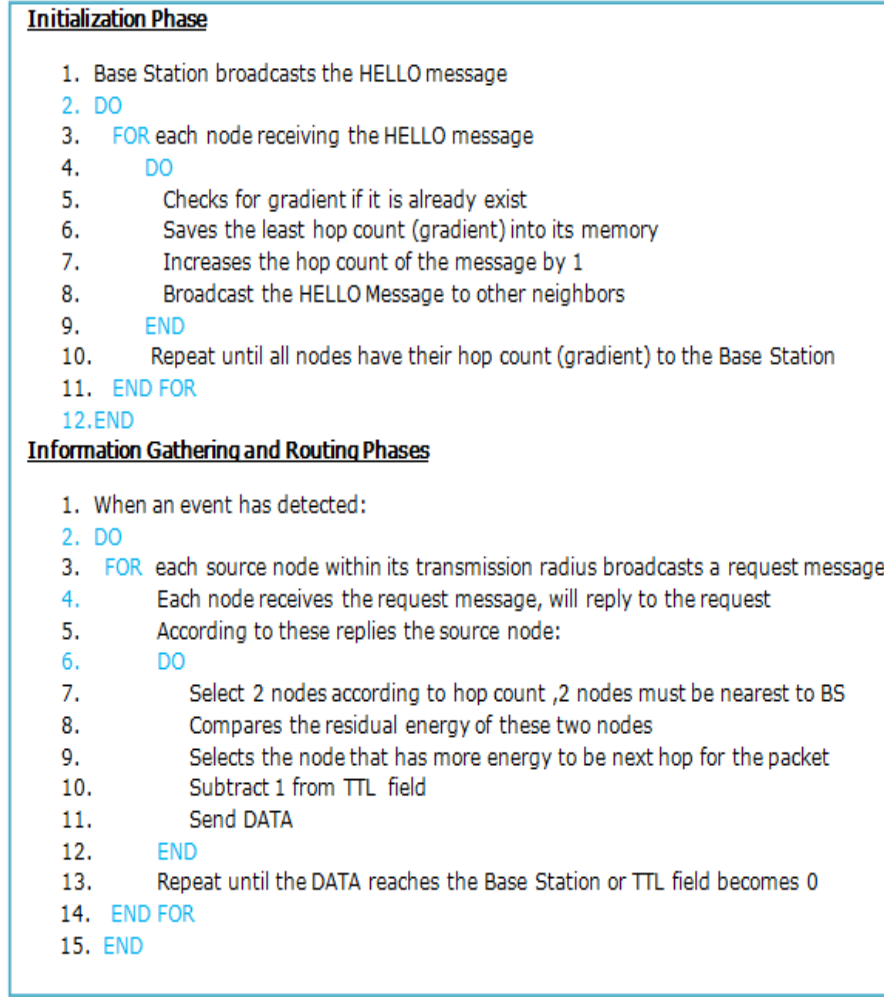
**TTL:** Time To Live (Yaşama Zamanı) paketin TTL'si 2x kaynağın sekme sayısına eşittir. Eğer bu alan 0 olursa paket iptal olur.

Sıradaki sekme seçilir seçilmez adresi üstbilgi'nin **sıradaki alanına** yazılır. Kaynak kendi adresini, **kaynak alanına** ve sadece saptanan olayın düğümü için **şimdiki alanına** yazar. Bundan sonra **kaynak alanı** (olayın saptandığı düğümün adresi) sabit kalacak ama **şimdiki alanı** şimdiki düğümüne göre değişecektir. Üstbilgi'nin **Sekme alanında** sıradaki düğümün sekme sayısı yazacaktır. Son olarak, kaynağın kendi sekme sayısından daha küçük sekme sayısına sahip bir komşu bulmayacağını, aksi halde kendisinininkine eşit sekme sayısına sahip komşu seçeceğini dikkate alarak (2 x kaynağın sekme sayısı) **TTL alanına** yazılacaktır. Paketi alan her bir düğüm, TTL alanından 1 çıkaracak ve paketi Baz İstasyonuna doğru ilerletecektir. Paket Baz İstasyonu paketi alıncaya veya TTL alanı 0 oluncaya kadar ilerlemeye devam edecektir. Bilgi Toplama ve Yönlendirme fazı akış şeması için Şekil 5,4'e bakınız. Başlatma fazı bilgi toplama fazı, yönlendirme fazı sözde kodlarına şekil 5,5'e bakınız.

Sonuç itibariyle, önerdiğimiz protokolde, ilk olarak; baz istasyona yakın olan iki düğüm seçerek gecikmeyi azalttık. Sonra maksimum artakalan enerjiye sahip ve baz istasyonuna en az mesafede olan düğümü seçerek enerjinin adil kullanımı vasıtasıyla ağ yaşam süresini uzattık. Yüksek paket iletim oranına erişmeyi başardık.



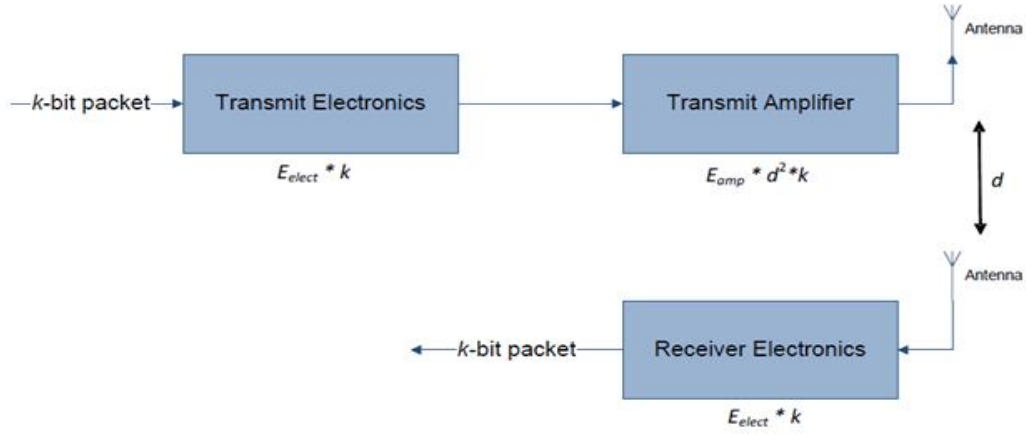
Şekil 5.4: Bilgi toplama ve yönlendirme fazının akış şeması



Şekil 5.5: FELGossiping protokolün sözde kodu

#### 5.4. SİMÜLASYON KURULUMU

Her şeyden önce, kablosuz iletişim bileşenide sensor düğümü enerji süzme faaliyetlerinden sorumlu olduğunu anladıktan sonra, [32] da geçen aynı radyo modelini kullandık. Radyo model şekil 5,5 'ta gösterilmiştir. Birinci derece radyo modelinde, her bir döngüde bir sensor düğümü iletimi veya alımı yapıldığında tüketilen enerji değerlendirmesi sunmaktadır. Radyo da, gerekli olan minimum enerji tüketiminde hedeflenen alıcılara ulaşmak için bir güç kontrolü vardır, radyo modeli şekil 5.6'da gösterilmiştir.



Şekil 5.6 : Radyo modeli

Matematiksel olarak, k-bit mesajı d mesafesi üzerinden iletilir, radyonun ihtiyaç duyduğu gerekli düğüm enerjisi aşağıdaki formül ile hesaplanır

$$E_{Tx} = E_{elect} \cdot k + E_{amp} \cdot d^2 \cdot k$$

Alıcının l-bit uzunluğunda bir mesajı almak için ihtiyaç duyduğu enerji ise aşağıdaki formül ile belirlenir

$$E_{Rx} = E_{elect} \cdot k$$

$E_{amp}$  ise amplifikatör tarafından kullanılan enerjidir

$E_{elec}$ : alıcı ya da verici devreyi çalıştırmak için gerekli enerji.

Kablosuz Sensor Ağların davranışının benzetimini yapmak için, NS-2,OMENT, MATLAB, OPNET ve değerleri gibi çeşitli simülasyon araçları vardır. Ancak, önerilen protokolda NS2 ve MATLAB benzetimi seçtik.

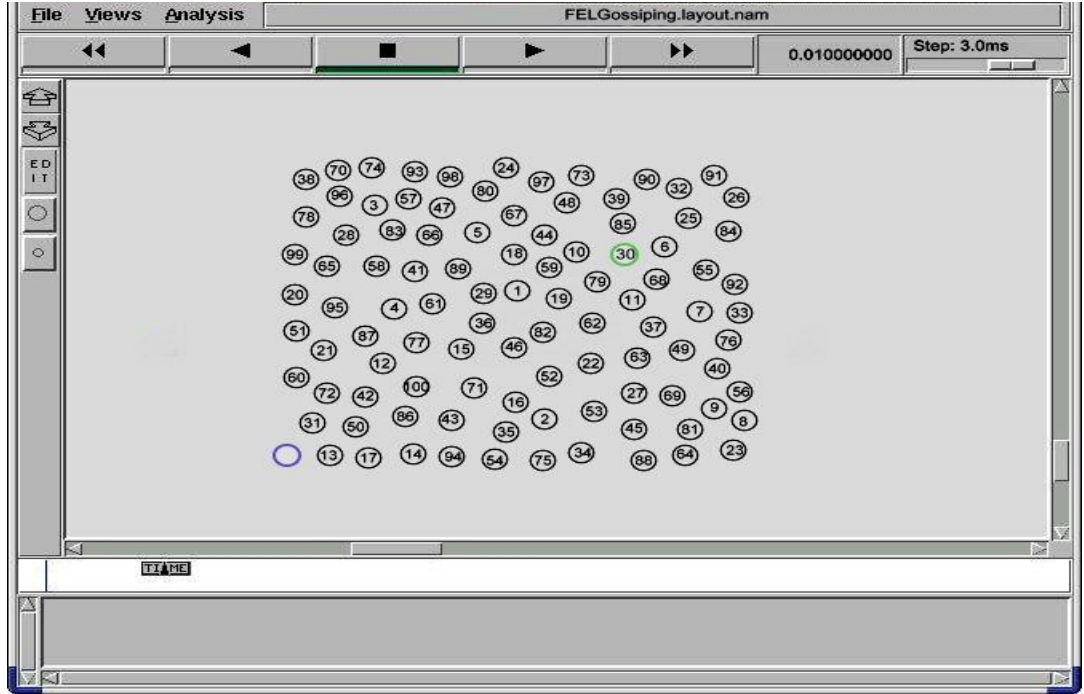
uçtan uca gecikme, ortalama enerji tüketimi veya tüm ağ üzerinden enerji tasarrufu, atlama sayısı ortalaması ve paket kaybı sayısının ortalaması açısından trace dosyası sonuçlarını analiz etmek için Perl programlama kullanır. Sonuç olarak perl programların sonuçlarını MATLAB aracı kullanarak çizilir.

NS-2, ayrık olay ve nesne-yönelimli ağ simülatördür. Ad-hoc ağ yönlendirme protokolları, çok noktaya yayın protokolü ve çeşitli ağ bileşenleri simüle etmek için kullanılır.

NS-2 trace dosyası simülasyon süresi boyunca, ağda tüm olayları içerir. Örneğin, bağlantılar, düğümleri ve paket izleri.

Ağ Animator (NAM), ağda trace dosyası bilgilerine bağlı olarak meydana gelen olayları izlemek için bir animasyon aracıdır. Bu tezde daha esnek ve yeterli değişiklikler NS-2 kullanılarak yapılabilir.

Bu simülasyonda, 100m x 100 m alanında 100 tane düğüm rastgele olarak dağıtıldı. Şekil 5.7 simüle edilen bir ortamda (topolojisi) 100 düğüm dağıtımını göstermektedir.



Şekil 5.7: 100m × 100m alanında 100 düğüm dağıtımını

Düğümleri dağıtıldığında, bazı varsayımlar düğüm özellikleri ile ilgili yapılmış ve bu varsayımlar aşağıdaki noktalarda gösterilmiştir:

- Tüm düğümler homojendir.
- Düğümler statiktir.
- Tüm düğümleri aynı başlangıç enerjisi ile başlar.
- Her bir düğüm herhangi bir zamanda geri kalan enerjisini bilir.
- Baz istasyonu (0,0) alanında yer alır.

- Her iletim veya alım , enerji azalması her bir düğüm için oluşur. Bu nedenle, sonraki hop aynı düğüm seçme olasılığını azaltacaktır.

Simülasyonda kullanılan parametreler Tablo 5.1 'de listelenmiştir:

Tablo 5.1: Simülasyon parametreleri

Parametreler	Miktar
Düğüm sayısı	100
Her düğümün başlangıç enerjisi(Joules), Eint(n)	100
Packet size(k) in bytes	100
Eelec in nano Joule per byte	70
Eamp	100
Baz istasyon kordinatı	(0,0)
Her bir düğüm İletim aralığı (metre).	20

## 5.5. PERFORMANS PARAMETRELERİ

### 5.5.1. Genel ağda enerji tasarrufu

Bu ölçü simülasyon süresi sonunda her düğümde artakalan enerjisini topladıktan sonra ağ düğümleri tarafından enerji tasarrufunu ölçer. Enerji hesaplaması ağdaki bütün düğümlerde bir başlangıç enerjisinin yüklenmesiyle ve alınan güç, iletilen güç, yedek güç gibi diğer enerji parametrelerinin ayarlanmasıyla yapılır.

Aşağıdaki formüller, enerji tüketimini hesaplamak için kullanılır.

Bir düğüm için kalan enerji: Başlangıç enerjisi –enerji tüketimi (Joule).

Ağların toplam enerji tasarrufu (Joule) : Bütün düğümlerdeki toplam kalan enerji.

### 5.5.2. Uçtan-uca gecikme ortalaması

Uçtan uca gecikme ortalaması; bir kaynaktan alıcı düğümüne aktarılan veri paketi (uygulama paketleri) için geçen süreçtir. Bu terim, çoklu-hop yönlendirme protokolü kullanıldığı için kablosuz algılayıcı ağlarda çok önemlidir.

Uçtan uca gecikme ortalaması; her uygulamada gönderilen paket ve alıcı düğümü tarafından doğru alınan ortalama gecikmesi alınarak hesaplanır.

Paketi gecikme (msec)= Alıcı tarafından alınan gecikme-kaynak düğüm tarafından gönderilen gecikme.

Gecikme paket ortalaması(msec)=Tüm paket gecikmelerin ortalaması-alınan paketlerin sayısı.

### 5.5.3. Kayıp paket sayısı

Gossiping protokoluda ana konu paket kaybı olduğu için, önerilen protokol diğer protokollar ile karşılaştırmak için paket kaybı kullanılan en önemli ölçüt.

Paket Kayıp ortalaması baz istasyonu tarafından alınan veri paket sayısı toplam gönderilen paket sayısından çıkarılarak hesaplanır.

Aşağıdaki kayıp paket sayısını hesaplamak için kullanılan formül:

Paket Kayıp sayısı= (gönderilen paket sayısı) - (baz istasyonu üzerinden alınan paket sayısı).

### 5.5.4. Atlama sayısının ortalaması

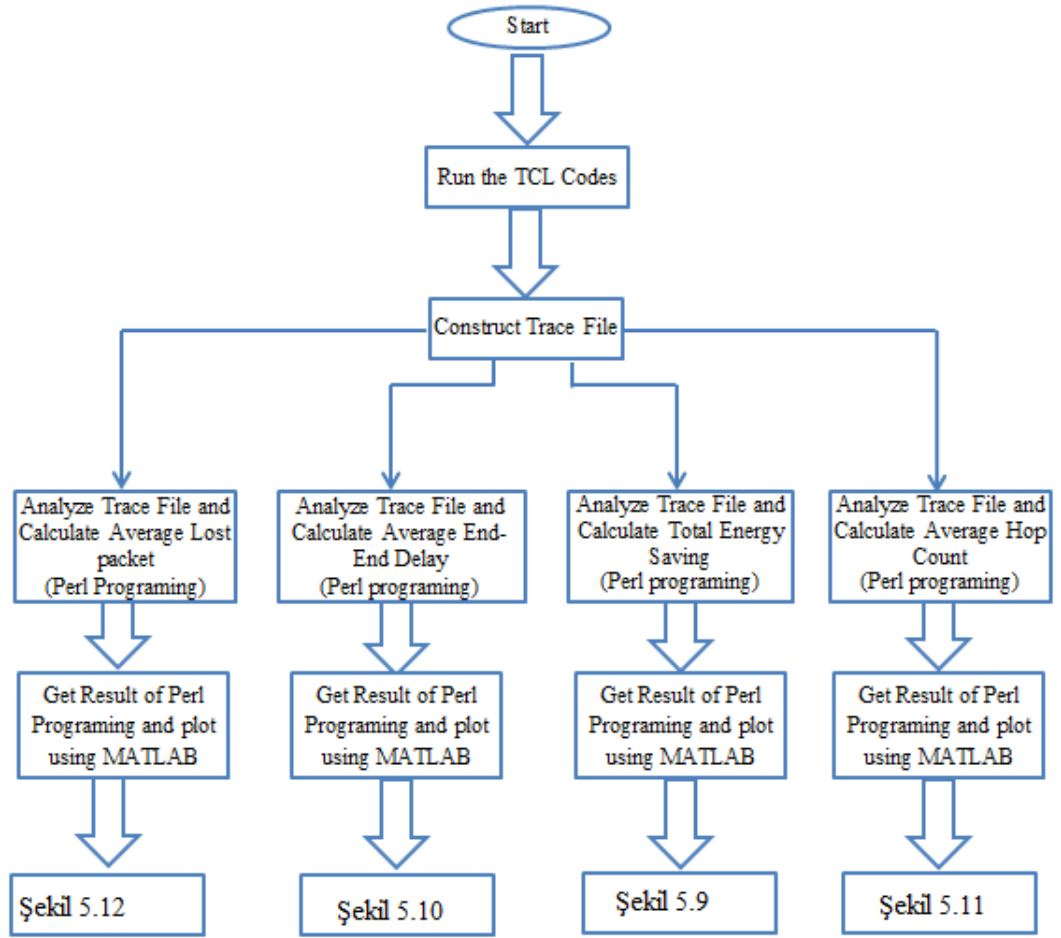
Bu ölçü, kaynak düğümden alıcı düğüme veri paketlerini iletmek için gerekli olan atlama sayısının ortalamasını ölçer. MAC katmanı paketlerinin sayısının, kaynak düğümlerden gönderilen uygulama paketlerinin sayısına bölünmesiyle hesaplama yapılır.

Ortalama atlama sayısı = (MAC katman paketleri) / (İletilen paketlerin sayısı)

## 5.6. SİMÜLASYON SONUÇLARI

Bu önerilen FELGossiping yönlendirme protokolü, diğer yönlendirme protokolleri ile kıyaslandığında daha iyi performans gerçekleştirdiğini gösterdi. Önerilen protokolde paket kaybı, uçtan uca gecikme ortalaması, atlama sayısı ve toplam enerji tasarrufu gibi performanslar kullandık. Simülasyon Aşağıdaki diyagramda (Şekil 5.8) açıklandı. TCL ve perl programlama kodu ekte eklendi.





Şekil 5.8 : Simülasyon akışı blok diyagramda açıklanmıştır.

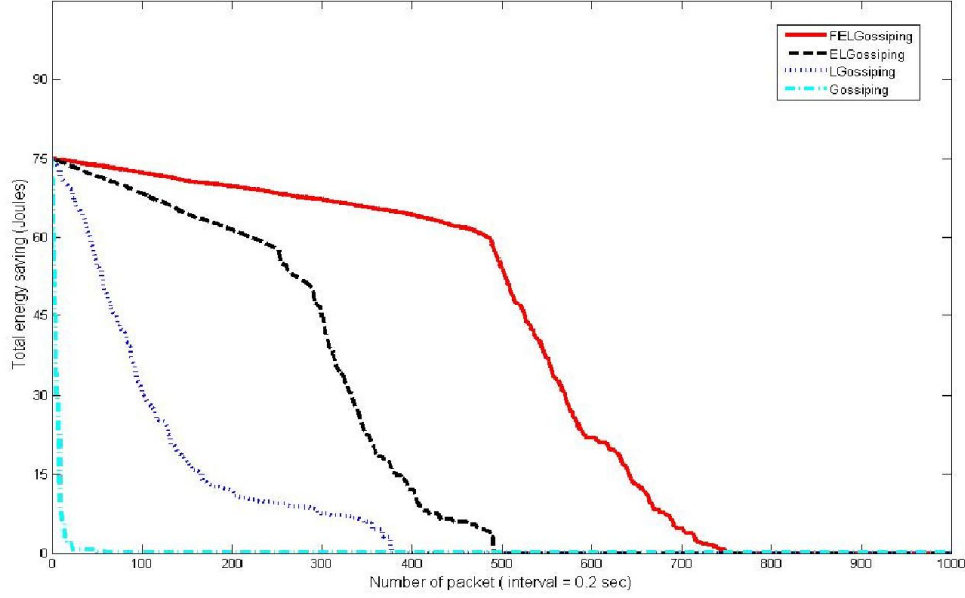
### 5.6.1. Genel ağda toplam enerji tasarrufu

Bu metrik simülasyon zamanı sırasında düğümler tarafından kaydedilen enerji miktarı hesaplar. Başlangıçta tüm düğümler aynı başlangıç enerjisiye sahiptir ve enerji tasarrufu hesaplamak için aşağıdaki verilen denklem kullanırız:

genel ağda toplam kaydedilen enerji (J) = tüm düğümlerin artakalan enerji toplamı

Önerilen protokol başka protokolar gibi seçilmiş değildir, önerilen protokolda geçiş düğümleri körü körüne (artakalan enerji ile ilgili bir fikir kalmadan) seçilmiş değildir. Ayrıca, her düğüm için iletimde veya alımda enerji azalması gerçekleşir.

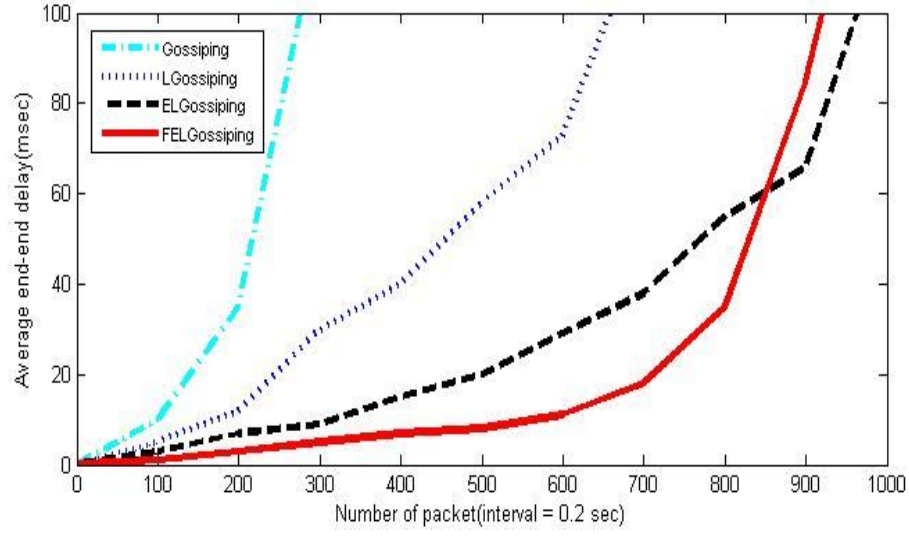
Bu nedenle, sonraki sekmeye aynı düğüm seçme olasılığını azaltacaktır. Böylece, enerji dengeli ve adil bir şekilde kullanılır. Şekil 5,9'da FELGossiping protokolu diğer protokol ile karşılaştırarak genel ağda enerji tasarrufunu ve ağ ömrünü uzadığını gösteriyor.



Şekil 5.9 : Toplam enerji tasarrufu

### 5.6.2. Uçtan uca gecikme ortalaması

Bu metrik kaynak düğümünden gelen orjinal packet ile baz istasyonunda alınan paket gecikmesini ölçer. Bu ölçüm, aynı zamanda diğer protokolleri ile karşılaştırıldığında daha kısa bir süre içinde veri paketleri sunmak için yönlendirme protokolü yeteneğini gösterir. Bu simülasyon çalışmasında, uçtan uca gecikme ortalaması milisaniye (msn) olarak bölüm 5.5.2 'de verilen denklem kullanılarak hesaplandı. Şekil 5.10, dört farklı yönlendirme protokolleri sonuçlar göstermektedir. Tüm grafikler eğilimini ağda paketlerinin sayısını artırarak artırır. Bu da ağ yükünü artırmaya neden olur. Şekil 7.5 FELGossiping protokol diğer protokoller ile karşılaştırdığında daha yüksek performans gösteriyor. Şekil 5.10 FELGossiping protokol diğer protokoller ile karşılaştırdığında daha yüksek performans gösteriyor. Baz istasyonuna en kısa yolu bulma yeteneğe sahiptir. Diğer protokollarda düğümler sonraki sekme için rastgele olarak seçilir, bu protokolda ise sonraki sekme için baz istasyona en yakın olan iki düğüm seçilir. Bu nedenle, gecikme FELGossiping protokoldında diğer protokollarla karşılaştırıldığında en az olur. Şekil 5.10 'da simülasyon sonucunu görebiliriz.

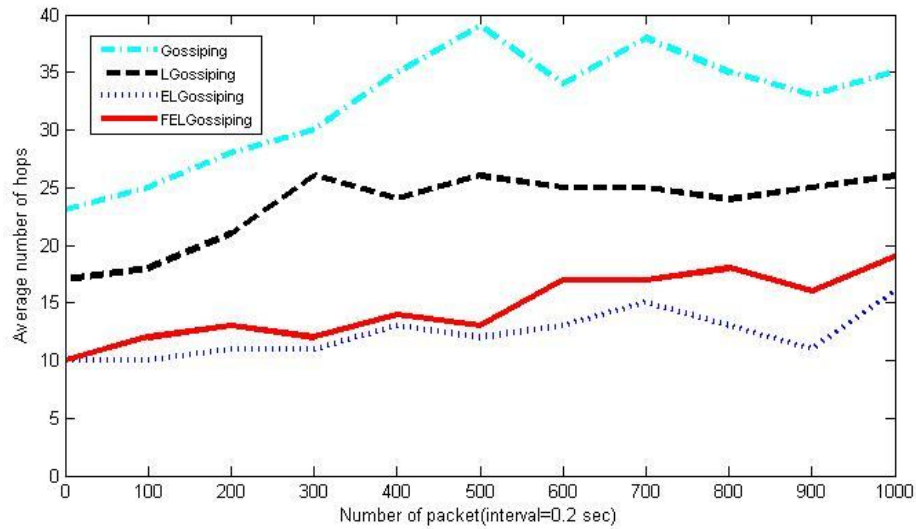


Şekil 5.10 : Uçtan uca gecikme ortalaması

### 5.6.3. Atlama sayısı ortalaması

Bu metrik çok-atlama sisteminde en az atlama sayısı kullanarak veri göndermek için yönlendirme protokolü performansini gösterir. Testimizde, baz istasyonu, (0, 0) X ve Y eksenlerinde yerleştirilmiş olur ve program tarafından kaynaklar rastgele seçilir. böylece kaynaktan gelen baz istasyona atlama ortalamasını ölçebiliriz. Atlama sayısının ortalaması, bölüm 7.2.4 'de verilen formül kullanılarak hesaplanır. Şekil 7.7 bu metrikin test sonuçlarını gösterir. FELGossiping baz istasyona az atlama sayısında ulaşır.

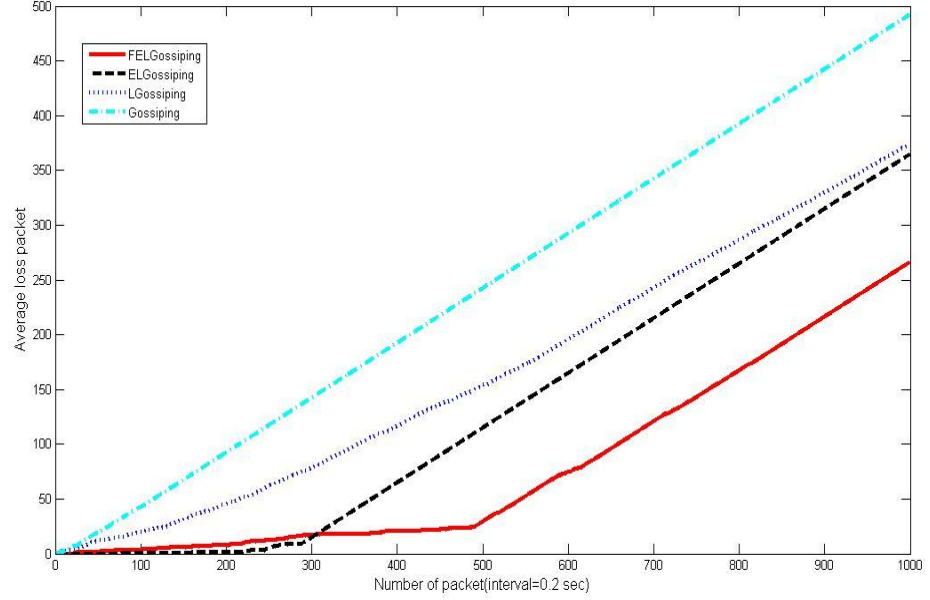
FELGossiping protokolün az atlama nedeni en az atlama sayıda baz istasyona doğru bir yol seçiminde daha etkin bir protokoldür.



Şekil 5.11: Atlama sayısı ortalaması

#### 5.6.4. Kayıp Paket sayısı

Şekil 5.12 'de görüldüğü gibi, FELgossiping protokolü Gossiping , LGossiping ve ELGossiping ile karşılaştırırken paket kaybı daha az olur . Ancak, 500 paket gönderdikten sonra paket kaybı çoğalıyor.



Şekil 5.12: Kayıp Paket sayısı

Son olarak, simülasyon sonuçlarından şu sonuca ulaşabiliriz: (FELGossiping) protokolümüzün enerji tasarrufu ve en az gecikme ile etkin bir şekilde veri sunarak ağ ömrünü çoğaltıyor.

## 6. TARTIŞMA VE SONUÇ

Bu tez, Kablosuz Sensor Ağlarda derin bir araştırmanın sonucudur. Bu çalışmada, sensor düğümlerin donanım mimarisi, WSNs uygulamaları ve kümeleme ve yönlendirme protokolleri, genel bir şekilde incelenmiştir. Normal pile çalışan kablosuz sensor düğümlerin enerjisi sınırlı olduğundan enerji- verimli yönlendirme protokolleri kullanılması gerekir. Yönlendirme protokollerin ana hedefi enerji tüketimini en aza indirmesidir yada indirmeye çalışmasıdır ve ağ yaşam süresini en üst düzeye çıkarmasıdır. Böylece, bazı enerji-verimli yönlendirme protokolleri geniş bir karşılaştırma yapılmıştır.

WSNs hakkında detaylı bir araştırma yaptıktan sonra, Konum-tabanlı yönlendirme protokollerini (LGossiping) geliştirerek FELGossiping (adil Enerji -Verimli konum-tabanlı Gossiping protokolü) önerdik. Önerilen protokol Gossiping protokolün gecikme, paket kaybı ve enerji tüketimi gibi önemli dezavantajlarını giderdiğini gördük, FELGossiping protokolü üç fazdan oluşur; ağ kurulumu, bilgi toplam fazı ve yönlendirme fazı. Kurulum fazında, düğümler ve baz istasyonu arasında bir gradyan oluşturarak her bir düğümün baz istasyonu olan uzaklık mesafesini bilir.

İkinci fazda ise kaynak düğümü iletim yarıçapında komşu düğümlerine bilgi mesajı göndererek, düğümlerden baz istasyona olan sekme sayısını ve düğümlerin artakalan enerjilerini ister. Her düğüm bilgi mesajı aldıktan sonra sekme sayılarını ve artakalan enerjilerini kaynak düğüme gönderir. Yönlendirme fazında, düğüm olayı yakaladıktan sonra, baz istasyona doğru olan en ve en az sekme sayısında olan iki düğüm seçer ,sonra iki düğümün arasındaki en çok artakalan enerji olan düğüm seçilir ve olay bu düğüm yoluyla baz istasyona doğru iletilir . Bu işlem olay baz istasyona ulaşana kadar yada TTL bitene kadar devam eder.

FEELGossiping yönlendirme protokol performansı, simülasyon platformunda üç yönlendirme protokolleri(Gossiping, LGossipig ve ELGossiping) ile karşılaştırılarak değerlendirilir. Ayrıca, önerilen protokolu diğer üç protokola karşılaştırmak için gecikme, paket kaybı, tur başına toplam enerji ve canlı düğüm sayıyı metrik performansını olarak kullanıldı. Bu çalışmanın simülasyon sonuçları FELGossiping

protokolü, gecikme, paket teslim oranı ve enerji tüketimi (ağ ömrünü uzatmak) açısından yönlendirme protokolleri muadillerine göre daha iyi performans gösteriyor. Sonuç olarak, düğümlerde dengeleme ve adil enerji kullanarak ağ ömrünü artırır, paket gönderirken daha fazla enerjisi olan düğüm seçilir. Sonuç olarak, (FELGossiping)protokol, enerji tasarrufu ve en az gecikme ile etkin bir şekilde veri sunmak ağ kullanım ömrü en üst düzeye çıkarır.

## KAYNAKLAR

1. CULLER, D., ESTRIN, D., and SRIVASTAVA M., 2004, *Overview of Sensor Networks, Guesteditors introduction*, Computer, 37 (8), 4.
2. AKYILDIZ, I.F., and VURAN, M., *Wireless Sensor Networks "IAN F. AKYILDIZ SERIES IN COMMUNICATIONS AND NETWORKING*, A John Wiley and Sons, Ltd, Publication, 3.
3. DARGIE, W., and POELLABAUER, C., 2010, *Fundamentals Of Wireless Sensor Networks Theory And Practice*, Wiley and Sons, United Kingdom, 978-0-470-99765-9.
4. SHINGHAL, K., DR. ARTI NOOR, DR. NEELAM SRIVASTAVA, DR. RAGHUVIR SINGH, 2010, *Wireless Sensor Networks In Agriculture: For Potato Farming*, Kshitij Shinghal et. al., International Journal of Engineering Science and Technology, 2(8), 3955-3963.
5. AKYILDIZ, I.F., SU, W., SANKARASUBRAMANIAM, Y., CAYIRCI, E., 2002, *A survey on Sensor Networks*, IEEE communication magazine.
6. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards
7. TinyOS. Available at <http://www.tinyos.net/>.
8. CAO, Q., ABDELZAHER, T., STANKOVIC, J., and TIAN HE, 2008, *The LiteOS Operating System: Towards Unix-like Abstractions for Wireless Sensor Networks*.
9. DUNKELS, A., GRÖNVALL, B., VOIGT, T., CONTIKI - *A Lightweight and Flexible Operating System for Tiny Networked Sensors*.
10. "Sun SPOT System " A SUN LABS RESEARCH PROJECT. Sun Microsystems, Inc.
11. KHAMFOROOSH, K., AND KHAMFOROUH, H., 2009, *A new routing Algorithm for Energy Reduction in Wireless Sensor Networks*, 1.
12. DECHENE, D. J., JARDALI, A.E., LUCCINI, M., AND SAUER, A., *A Survey of Clustering Algorithms for Wireless Sensor Networks*, The University Of Western Ontario London, Ontario, Canada.
13. AL-SAKIB, K. P., CHOONG, S. H., and HYUNG, W. L., 2006, *Smartening the Environment using Wireless Sensor Networks in a Developing Country*, ISBN 89-5519-129-4, 705-706.
14. STALLINGS, W., 2000, *Data & Computer Communications*, 6th ed., Prentice Hall, New Jersey, 453.

15. KARAYANNIS, G., 2003, *Emerging Wireless Standards: Understanding the Role of IEEE 802.15.4 & ZigBee in AMR & Submetering*.
16. S. DAI, X. JING and L. LI, 2005, *Research and Analysis on Routing Protocols for Wireless Sensor Networks*” in Proceedings International Conference on Communications, Circuits and Systems, 1, 407-411.
17. Q. LIANG, 2005, *Fault-Tolerant and Energy Efficient Wireless Sensor Networks: A Cross-Layer Approach*, in Proceedings of IEEE MILCOM 2005, 3, 1862-1868.
18. AMIYA NAYAK AND IVAN STOJMENOVIC, *Wireless Sensor and Actuator Networks Algorithms and Protocols for Scalable Coordination and Data Communication*, A John Wiley & Sons, 2-3- 6 .
19. FRANCOIS DEPIENNE, *Wireless Sensor Networks Application for Agricultural Environment Sensing in Developing Countries Francois Depienne*, Master Candidate in Communication Systems.
20. TATIANA BOKAREVA, WEN HU, SALIL KANHERE, BRANKO RISTIC, NEIL GORDON, TRAVIS BESSELL, MARK RUTTEN and SANJAY JHA, 2006, *Wireless Sensor Networks for Battlefield Surveillance*, Brisbane.
21. <http://www.cs.virginia.edu/wsn/vigilnet/index.html>
22. K .SOHRABY D. MINOLI T. ZNATI, 2007, *Wireless Sensor Networks Technology, Protocols, and Applications*, John Wiley & Sons, 56.
23. AMEER AHMED ABBASI A, MOHAMED YOUNIS, 2007, *A survey on clustering algorithms for wireless sensor networks*, Computer Communications, 30, 2826–2841.
24. DENNIS, J. BAKER, AND ANTHONY, EPHREMIDES, 1981, *The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm*, 0090-6778/81/1100-1694\$00.
25. A. AMIS, R. PRAKASH, T. VUONG, AND D. HUYNH, 2000, *Max-Min D-Cluster Formation in Wireless Ad Hoc Networks*, IEEE INFOCOM, March 2000.
26. R. KUMARI, J. SINGH LAMBA, 2010, *Performance Evaluation of LCA2, Highest-Connectivity, Max-Min D-Cluster Algorithms in Mobile Ad hoc Networks*, TECHNIA – International Journal of Computing Science and Communication Technologies, 3 (1), July 2010, 0974-3375.
27. C. R. LIN AND M. GERLA , 1997, *Adaptive clustering for mobile wireless networks*, IEEE Journal on Selected Areas in Communications , 15 (7), 1265 – 1275.
28. C. CHIANG , H. WU , W. LIU , AND M. GERLA ,1997, *Routing in clustered multihop, mobile wireless networks with fading channel* , in Proceeding of IEEE



- Singapore International Conference on Networks (SICON '07), Singapore, Apr. 1997, 197 – 211.
29. A. AMIS, R. PRAKASH, T. VUONG, AND D. HUYNH, 2000, *Max-Min D-Cluster Formation in Wireless Ad Hoc Networks*, IEEE INFOCOM, March 2000.
  30. MAINAK CHATTERJEE, SAJAL K. DAS and DAMLA TURGUT, 2002, *WCA: AWeighted Clustering Algorithm for Mobile Ad Hoc Networks*, Cluster Computing 5, 193–204.
  31. NAUMAN ISRAR and IRFAN AWAN, *Multihop Clustering Algorithm for Load Balancing in Wireless Sensor Networks*, I. J. of SIMULATION, 8 (1), 21.
  32. W. HEINZELMAN, A. CHANDRAKASAN, and H. BALAKRISHNAN, 2000, *Energy-Efficient Communication Protocols for Wireless Microsensor Networks*, In Proceedings of Hawaiian International Conference on Systems Science, January 2000.
  33. V. LOSCRÌ, G. MORABITO, S. MARANO, 2005, *A Two-Levels Hierarchy for Low-Energy Adaptive Clustering Hierarchy (TL-LEACH)*, IEEE, 0-7803-9152-7.
  34. MAO YE1, CHENGFA LI1, GUIHAI CHEN1 AND JIE WU, 2005, *EECS: An Energy Efficient Clustering Scheme in Wireless Sensor Networks*, Performance, Computing, and Communications Conference, IPCCC 2005. 24<sup>th</sup> IEEE International .
  35. OSSAMAYOUNIS AND SONIA FAHMY, 2004, *HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Adhoc Sensor Networks*, IEEE TRANSACTIONS ON MOBILE COMPUTING, 3 (4), OCTOBERDECEMBER 2004.
  36. STEPHANIE LMDSEY AND CAULIGI S. RAGHAVENDRA, 2001, *PEGASIS: Power-Efficient GATHERing in Sensor Information Systems*, IEEE, 0-7803-7231.
  37. KAIXIN XU, MARIO GERLA, *A Heterogeneous Routing Protocol Based on a New Stable Clustering Scheme*, IEEE, 0-7803-7625-010.
  38. P. DING, J. HOLLIDAY, A. CELIK, 2005, *Distributed Energy Efficient Hierarchical Clustering for Wireless Sensor Networks*, Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems(DCOSS'05), Marina Del Rey.
  39. E. Yoneki and J. Bacon, September, 2005, *A survey of Wireless Sensor Network technologies: research trends and middleware's role*, Computer Laboratory, University of Cambridge, Technical report UCAM-CL-TR-646.
  40. W. R. HEINZELMAN, J. KULIK, AND H. BALAKRISHNAN, 1999, *Adaptive protocols for information dissemination in wireless sensor networks*, In Proceedings of MobiCom '99, 174–185.

41. C. INTANAGONWIWAT, R. GOVINDAN AND D. ESTRIN, 2000, *Directed diffusion: A scalable and robust communication paradigm for sensor networks*, in the *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00)*, Boston, MA, August 2000.
42. HEINZELMAN, W., KULIK, J., BALAKRISHNAN, H., 1999, *Adaptive Protocols for Information Dissemination in Wireless Sensor Networks*, Proc. 5<sup>th</sup> ACM/IEEE Mobicom Conference (MobiCom '99), Seattle, USA, August, 174-85, (1999).
43. D. BRAGINSKY AND D. ESTRIN, *Rumor routing algorithm for sensor networks*, In WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, 22–31. ACM Press, 2002.
44. N. SADAGOPAN, B. KRISHNAMACHARI, AND A. HEMLY, 2003, *The ACQUIRE mechanism for efficient querying in sensor networks*, In Proceedings of the First International Workshop on Sensor Network Protocol and Applications, Anchorage, AK, USA, May 2003.
45. SCHURGERS , C. & SRIVASTAVA , M. B., 2001, *Energy-efficient routing in wireless sensor networks* , IEEE Military Communications Conference , 1, 357 – 361.
46. F. YE, A. CHEN, S. LU, AND L. ZHANG, 2001, *A scalable solution to minimum cost forwarding in large scale sensor networks*, In Proc. Intl. Conf. on Computer Communications and Networks (ICCCN), Scottsdale, AZ, 2001.
47. R. SHAH ET AL., 2002, *Energy aware routing for low energy ad - hoc sensor networks*, in Proceedings of IEEE Wireless Communications and Networking Conference (WCNC' 02) , Orlando, FL, Mar. 2002 , 350 – 355 .
48. A. MANJESHWAR AND D. P. AGRAWAL, 2001, TEEN: a protocol for enhanced efficiency in wireless sensor networks, In Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, USA, April 2001.
49. A. MANJESHWAR AND D. P. AGRAWAL, *APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks*, In Proceedings of the 2nd International Workshop on Parallel.
50. S. LINDSEY AND C. S. RAGHAVENDRA, 2002, *PEGASIS: power efficient gathering in sensor information systems*, In Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, March 2002.
51. Y. XU, J. HEIDEMANN, and D. ESTRIN, 2001, *Geography - informed energy conservation for ad - hoc routing*, *Proceedings ACM/IEEE MobiCom ' 01* , Rome, Italy, 70 – 84 .
52. Y. YU , R. GOVINDAN , and D. ESTRIN, 2001, *Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks*, Technical Report UCLA/CSD - TR - 01 - 0023 , UCLA Computer Science Department.

53. V. RODOPLU AND T. H. MENG , 1999, *Minimum energy mobile wireless networks*, IEEE Journal on Selected Areas in Communications, 17(8), 1333 – 1344 .
54. L. LI AND J. Y. HALPERN, 2001, *Minimum - energy mobile wireless networks revisited*, Proceedings IEEE ICC ' 01 , Helsinki, Finland, June 2001, 278 – 283 .
55. K. SOHRABI, J. GAO, V. AIL AWADHI, AND G.J. POTTIES, 2000, *Protocols for self-organization of a wireless sensor networks*, IEEE Personal Communications, 7(5), 16–27.
56. WAIRAGU G. RICHARD, 2009, *Extending Leach Routing Algorithm For Wireless Sensor Networks*, Master thesis.
57. Y. YAO AND J. GEHRKE, 2002, *The COUGAR approach to in-network query processing in sensor networks*, SIGMOD Record, 31(3), 9–18.
58. T. HE, J. A. STANKOVIC, C. LU, AND T. F. ABDELZAHER, 2003, *SPEED: A stateless protocol for real-time communication in sensor networks*, In *Proceedings of ICDCS*, 46–58.
59. C. KARLOF, Y. LI, AND J. POLASTRE, 2003, *ARRIVE: Algorithm for robust routing in volatile environments*. Technical Report UCB/CSD-03-1233, Computer Science Division, UC Berkley.
60. B. DEB, S. BHATNAGAR, AND B. NATH, 2003, *ReInForM: Reliable information forwarding using multiple paths in sensor networks*, In *Proceedings of LCN*, 406–415.
61. K. SOHRABI, J. POTTIE, 2000, *Protocols for self-organization of a wireless sensor network*, IEEE Personal Communications, 7(5), 16-27.
62. VERDONE, R. DARDARI, D. MAZZINI, G, AND CONT, A., 2007, *Wireless sensor and actuator networks technology, analysis and design (1 edition)*, Published by Elsevier, 978-0-12-172539-4.
63. WENDI BETH HEINZELMAN, 2000, *Application-Specific Protocol Architectures for Wireless Networks*, Wendi Beth Heinzelman, MM. All rights reserved. Submitted to the Department of Electrical Engineering and Computer Science on May 19, 2000, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.
64. W.B.HEINZELMAN, A. P. CHANDRAKASAN, H, BALAKRISHNAN, 2002, *An Application-Specific Protocol Architecture for Wireless Microsensor Networks*, IEEE Transactions on Wireless Communications, 1(4).
65. M. BANI YASSEIN, A. AL-ZOU'BI, Y. KHAMAYSEH, W. MARDINI, 2009, *Improvement on LEACH Protocol of Wireless Sensor Network (VLEACH)*, International Journal of Digital Content Technology and its Applications, 3(2), June 2009.

66. DISSERTATION, HANG ZHOU, ZHE JIANG AND MO XIAOYAN, 2006, *Study and Design on Cluster Routing Protocols of Wireless Sensor Networks*.
67. KAZEM SOHRABY, DANIEL MINOLI and TAIEB ZNATI, 2007, *WIRELESS SENSOR NETWORKS Technology, Protocols, and Applications*, Copyright \_ 2007 by John Wiley & Sons, Inc. All rights reserved. Published by John Wiley & Sons, Inc., Hoboken, New Jersey. ISBN 978-0-471-74300-2.
68. EDITED BY JUN ZHENG ABBAS JAMALIPOUR, 2009, *Wireless Sensor Networks a Networking Perspective*, Copyright © 2009 by Institute of Electrical and Electronics Engineers John Wiley & Sons, Inc., Hoboken, New Jersey, 978-0-470-16763-2.
69. SHUJIANG LI, XIN MA, XIANGDONG WANG, MINGHAO TAN, 2011, *Energy-efficient multipath routing in wireless sensor network considering wireless interference*, J Control Theory Appl 2011 9 (1) 127–132. DOI 10.1007/s11768-011-0263-4.
70. S. M. HEDETNIEMI, S. T. HEDETNIEMI, AND A. L. LIESTMAN, 1988, *A survey of gossiping and broadcasting in communication networks*. Networks, 18(4), 319–349, 1988.
71. YUECHENG ZHANG AND LIAUG CHENG, 2004, *Flossiping: A New Routing Protocol for Wireless Sensor Networks*, Proceedings of the 2004 IEEE. International Conference on Networking, Sensing & Control Taipei, Taiwan, March 21-23. 2004.
72. WEI YEN, CHING-WEI CHEN, CHENG-HSIANG YANG, 2008, *Single Gossiping with Directional Flooding Routing Protocol in Wireless Sensor Networks*, IEEE, 3(40), Jhongshan N. Rd. Jhongshan District, Taipei City, Taiwan. 978-1-4244-1718-6.
73. S. KHEIRI, MB. GHAZNAVI GOUSHCHI, M. RAFIEE AND B. SEYFE, 2009, *An improved Gossiping Data Distribution Technique with Emphasis on Reliability and Resource Constraints*, IEEE 2009 International Conference on Communications and Mobile Computing, 9780769535012
74. ALI NOROUZI, MEHDI DABBAGHIAN, ALIREZA HATAMIZADEH, AND BURAK BERK USTUNDAG, 2010, *An Improved ELGossiping Data Distribution Technique with Emphasis on Reliability and Resource Constraints in Wireless Sensor Network*, IEEE, 978-1-4244-7406-6/10/\$26.00\_c 2010 IEEE.
75. HAYOUNG OH, KIJOON CHAE, 2007, *An Energy-Efficient Sensor Routing with low latency, scalabilit in Wireless Sensor Networks*, 2007 International Conference on Multimedia and Ubiquitous Engineering (MUE'07), 0-7695-2777-9/07 \$20.00 © 2007 IEEE.
76. M. HASAN KHODASHAHI, ALI NOROUZI, FATEMEH AMIRI AND MEHDI DABBAGHIAN, 2010, *A novel optimal routing algorithm by creating concentrically sectors in wireless sensor networks*. 8th Annual Communication Networks and Services Research Conference, 978-0-7695-4041-2/10 \$26.00 © 2010 IEEE. DOI 10.1109/CNSR.2010.55.

**EKLER**

A - TCL file

```

# Routing FELGossiping over 802.15.4      #
=====
=====
# Define options
#
=====
=====
set val(chan)      Channel/WirelessChannel  ;# Channel Type
set val(prop)      Propagation/TwoRayGround ;# radio-propagation model
set val(netif)     Phy/WirelessPhy/802_15_4
set val(mac)       Mac/802_15_4
set val(ifq)       Queue/DropTail/PriQueue  ;# interface queue type
set val(ll)        LL                        ;# link layer type
set val(ant)       Antenna/OmniAntenna     ;# antenna model
set val(ifqlen)    100                      ;# max packet in ifq
set val(nn)        100                      ;# number of nodes
set val(rp)        fel                      ;# FELGossiping
set val(x)         100
set val(y)         100

set val(tr)        wpan_demo3.tr
##set val(nam)     wpan_demo3.nam
set val(nodeDown)  no
#set val(errRate)  0                      ;# %
set val(traffic)   cbr                    ;# mix/cbr/poisson/ftp
set val(trInterval) 0.2                   ;# in seconds

#read command line arguments
proc getCmdArgv {argc argv} {
    global val
    for {set i 0} {$i < $argc} {incr i} {
        set arg [lindex $argv $i]
        if {[string range $arg 0 0] != "-"} continue
        set name [string range $arg 1 end]
        set val($name) [lindex $argv [expr $i+1]]
    }
}

getCmdArgv $argc $argv

set appTime1      50.5 ;# in seconds
set appTime2      53.8 ;# in seconds
set appTime3      55.8 ;# in seconds
set appTime4      58.8 ;# in seconds
set appTime5      59.8 ;# in seconds

```

```

set stopTime      500      ;# in seconds

# Initialize Global Variables
set ns_           [new Simulator]
set tracefd      [open ./$val(tr) w]
$ns_ trace-all $tracefd
if { "$val(nam)" == "wpan_demo3.nam" } {
    set namtrace  [open ./$val(nam) w]
    $ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
}

$ns_ puts-nam-traceall {# nam4wpan #}           ;# inform nam that this is a trace file
for wpan (special handling needed)

Mac/802_15_4 wpanCmd verbose on
Mac/802_15_4 wpanNam namStatus on             ;# default = off (should be turned on
before other 'wpanNam' commands can work)
#Mac/802_15_4 wpanNam ColFlashClr gold        ;# default = gold
#Mac/802_15_4 wpanNam NodeFailClr grey       ;# default = grey

# For model 'TwoRayGround'
set dist(5m) 7.69113e-06
set dist(9m) 2.37381e-06
set dist(10m) 1.92278e-06
set dist(11m) 1.58908e-06
set dist(12m) 1.33527e-06
set dist(13m) 1.13774e-06
set dist(14m) 9.81011e-07
set dist(15m) 8.54570e-07
set dist(16m) 7.51087e-07
set dist(20m) 4.80696e-07
set dist(25m) 3.07645e-07
set dist(30m) 2.13643e-07
set dist(35m) 1.56962e-07
set dist(40m) 1.20174e-07
Phy/WirelessPhy set CStresh_ $dist(20m)
Phy/WirelessPhy set RXThresh_ $dist(20m)

# set up topography object
set topo [new Topography]
$stopo load_flatgrid $val(x) $val(y)

# Create God
set god_ [create-god $val(nn)]

set chan_1_ [new $val(chan)]

# configure node

```

```

$ns_ node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace ON \
    -movementTrace OFF \
    -channel $chan_1_ \
        -energyModel "EnergyModel" \
    -initialEnergy 100 \
    -rxPower 0.075 \
    -txPower 0.075 \
        -idlePower 0.0003 \
    -sleepPower 0.0 \

for {set i 0} {$i < $val(nn)} {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0           ;# disable random motion
}

BlockSize = $BlockSize, CSkip0 = $cskip0, CSkip1 = $cskip1, CSkip2 = $cskip2,
CSkip3 = $cskip3, CSkip4 = $cskip4, CSkip5 = $cskip5, CSkip6 = $cskip6""

source ./wpan_100.scn

$ns_ at 0.0    "$node_(0) NodeLabel \"base station\""

$ns_ at 0.0    "$node_(0) sscs startCTPANCoord 1 3 3 UseClusterTree 7 4 4" ;#
startCTPANCoord <txBeacon=1> <BO=3> <SO=3>

$ns_ at 0.5    "$node_(1) sscs startCTDevice" ;# startCTDevice <isFFD=1>
<assoPermit=1> <txBeacon=0> <BO=3> <SO=3>
$ns_ at 1.8    "$node_(2) sscs startCTDevice"
$ns_ at 2.3    "$node_(3) sscs startCTDevice"
$ns_ at 2.8    "$node_(4) sscs startCTDevice"

$ns_ at 3.5    "$node_(5) sscs startCTDevice 1 1 1"

```

```

$ns_ at 5.4 "$node_(6) sscs startCTDevice 1 1 1"
$ns_ at 5.5 "$node_(7) sscs startCTDevice 1 1 1"
$ns_ at 6.6 "$node_(8) sscs startCTDevice 1 1 1"

$ns_ at 7.3 "$node_(9) sscs startCTDevice"
$ns_ at 8.8 "$node_(10) sscs startCTDevice"
$ns_ at 9.3 "$node_(11) sscs startCTDevice"
$ns_ at 10.8 "$node_(12) sscs startCTDevice"

# ---

for {set i 13} {$i < 17} {incr i} {
    $ns_ at 11.3 "$node_($i) sscs startCTDevice"
}
for {set i 17} {$i < 21} {incr i} {
    $ns_ at 12.3 "$node_($i) sscs startCTDevice"
}
for {set i 21} {$i < 25} {incr i} {
    $ns_ at 13.3 "$node_($i) sscs startCTDevice"
}

# ---

for {set i 25} {$i < 29} {incr i} {
    $ns_ at 14.3 "$node_($i) sscs startCTDevice"
}
for {set i 29} {$i < 33} {incr i} {
    $ns_ at 15.3 "$node_($i) sscs startCTDevice"
}
for {set i 33} {$i < 41} {incr i} {
    $ns_ at 16.3 "$node_($i) sscs startCTDevice"
}

# ---

for {set i 41} {$i < 49} {incr i} {
    $ns_ at 17.3 "$node_($i) sscs startCTDevice"
}
for {set i 49} {$i < 57} {incr i} {
    $ns_ at 18.3 "$node_($i) sscs startCTDevice"
}
for {set i 57} {$i < 61} {incr i} {
    $ns_ at 19.3 "$node_($i) sscs startCTDevice"
}

# ---

for {set i 61} {$i < 65} {incr i} {
    $ns_ at 20.3 "$node_($i) sscs startCTDevice "
}

```



```

}
for {set i 65} {$i < 73} {incr i} {
    $ns_ at 21.3 "$node_($i) sscs startCTDevice"
}
for {set i 73} {$i < 77} {incr i} {
    $ns_ at 22.3 "$node_($i) sscs startCTDevice "
}
for {set i 77} {$i < 80} {incr i} {
    $ns_ at 23.3 "$node_($i) sscs startCTDevice"
}
for {set i 80} {$i < 85} {incr i} {
    $ns_ at 24.3 "$node_($i) sscs startCTDevice "
}
}

# ---

for {set i 85} {$i < 93} {incr i} {
    $ns_ at 26.3 "$node_($i) sscs startCTDevice"
}
for {set i 93} {$i < 101} {incr i} {
    $ns_ at 27.3 "$node_($i) sscs startCTDevice"
}
}

Mac/802_15_4 wpanNam PlaybackRate 12ms
$ns_ at $appTime1 "Mac/802_15_4 wpanNam PlaybackRate 1.0ms"
$ns_ at [expr $appTime1 + 0.5] "Mac/802_15_4 wpanNam PlaybackRate 2.0ms"

$ns_ at $appTime1 "puts \"\nTransmitting data ... \n\""

# Setup traffic flow between nodes

proc cbrtraffic { src dst interval starttime } {
    global ns_ node_
    set udp_($src) [new Agent/UDP]
    eval $ns_ attach-agent $node_($src) \ $udp_($src)
    set null_($dst) [new Agent/Null]
    eval $ns_ attach-agent $node_($dst) \ $null_($dst)
    set cbr_($src) [new Application/Traffic/CBR]
    eval \ $cbr_($src) set packetSize_ 80
    eval \ $cbr_($src) set interval_ $interval
    eval \ $cbr_($src) set random_ 0
    #eval \ $cbr_($src) set maxpkts_ 10000
    eval \ $cbr_($src) attach-agent \ $udp_($src)
    eval $ns_ connect \ $udp_($src) \ $null_($dst)
    $ns_ at $starttime "$cbr_($src) start"
}

proc poissontraffic { src dst interval starttime } {

```

```

global ns_ node_
set udp($src) [new Agent/UDP]
eval $ns_ attach-agent $node_($src) $udp($src)
set null($dst) [new Agent/Null]
eval $ns_ attach-agent $node_($dst) $null($dst)
set expl($src) [new Application/Traffic/Exponential]
eval $expl($src) set packetSize_ 70
eval $expl($src) set burst_time_ 0
eval $expl($src) set idle_time_ [expr $interval*1000.0-70.0*8/250]ms ;# idle_time +
pkt_tx_time = interval
eval $expl($src) set rate_ 250k
eval $expl($src) attach-agent $udp($src)
eval $ns_ connect $udp($src) $null($dst)
$ns_ at $starttime "$expl($src) start"
}

if {$val(rp) == "fel"} {
    Mac/802_15_4 wpanCmd callBack 2 ;# 0=none; 1=failure only (default);
    2=both failure and success
}

if {("$val(traffic)" == "mix") || ("val(traffic)" == "cbr") || ("val(traffic)" ==
"poisson")} {
    if {"$val(traffic)" == "mix"} {
        set trafficName "cbr + poisson"
        set traffic1 cbr
        set traffic2 poisson
    } else {
        set trafficName $val(traffic)
        set traffic1 $val(traffic)
        set traffic2 $val(traffic)
    }
    puts "\nTraffic: $trafficName"
    #Mac/802_15_4 wpanCmd ack4data on
    puts [format "Acknowledgement for data: %s" [Mac/802_15_4 wpanCmd ack4data]]

    set bb [expr {int (rand() * 100)}] ;# choose source randomly

    ${traffic1}traffic $bb 1 $val(trInterval) $appTime1

    if {"$val(nodeDown)" == "yes"} {
        $ns_ at [expr $appTime2 + 3.0] "$node_(0) node-down"
        set tmpTime [format "%.1f" [expr $appTime2 + 3.0]]
        $ns_ at [expr $appTime2 + 3.0] "$ns_ trace-annotate \"(at $tmpTime) node
down: 0\""
    }
}

```

```

    $ns_ at [expr $appTime2 + 10.0] "$node_(0) node-up"
    set tmpTime [format "%.1f" [expr $appTime2 + 10.0]]
    $ns_ at [expr $appTime2 + 10.0] "$ns_ trace-annotate \"(at $tmpTime) node up:
0\\""
    }
    Mac/802_15_4 wpanNam FlowClr -p AODV -c tomato
    Mac/802_15_4 wpanNam FlowClr -p ARP -c green
    Mac/802_15_4 wpanNam FlowClr -p MAC -c navy
    $ns_ at $appTime1 "$node_(67) add-mark m1 blue circle"
    $ns_ at $appTime1 "$node_(45) add-mark m2 blue circle"
    $ns_ at $appTime1 "$ns_ trace-annotate \"(at $appTime1) $traffic1 traffic from node
67 to node 45\\""
    $ns_ at $appTime2 "$node_(71) add-mark m3 green4 circle"
    $ns_ at $appTime2 "$node_(74) add-mark m4 green4 circle"
    $ns_ at $appTime2 "$ns_ trace-annotate \"(at $appTime2) $traffic2 traffic from node
71 to node 74\\""
    if { "$val(traffic)" == "cbr" } {
        set pktType cbr
        set pktType2 cbr
    } elseif { "$val(traffic)" == "poission" } {
        set pktType exp
        set pktType2 exp
    } else {
        set pktType cbr
        set pktType2 exp
    }
    Mac/802_15_4 wpanNam FlowClr -p $pktType -s 67 -d 45 -c blue
    Mac/802_15_4 wpanNam FlowClr -p $pktType2 -s 71 -d 74 -c green4
}

proc ftptraffic { src dst starttime } {
    global ns_ node_
    set tcp($src) [new Agent/TCP]
    eval \ $tcp($src) set packetSize_ 60
    set sink($dst) [new Agent/TCPSink]
    eval $ns_ attach-agent \ $node_($src) \ $tcp($src)
    eval $ns_ attach-agent \ $node_($dst) \ $sink($dst)
    eval $ns_ connect \ $tcp($src) \ $sink($dst)
    set ftp($src) [new Application/FTP]
    eval \ $ftp($src) attach-agent \ $tcp($src)
    $ns_ at $starttime "$ftp($src) start"
}

if { "$val(traffic)" == "ftp" } {
    puts "\nTraffic: ftp"
    #Mac/802_15_4 wpanCmd ack4data off
    puts [format "Acknowledgement for data: %s" [Mac/802_15_4 wpanCmd ack4data]]
    Mac/802_15_4 wpanNam FlowClr -p AODV -c tomato
    Mac/802_15_4 wpanNam FlowClr -p ARP -c green
}

```

```

Mac/802_15_4 wpanNam FlowClr -p MAC -c navy
$ns_ at $appTime1 "$node_(67) add-mark m1 blue circle"
$ns_ at $appTime1 "$node_(45) add-mark m2 blue circle"
$ns_ at $appTime1 "$ns_ trace-annotate \"(at $appTime1) ftp traffic from node 67 to
node 45\"""
Mac/802_15_4 wpanNam FlowClr -p tcp -s 67 -d 45 -c blue
Mac/802_15_4 wpanNam FlowClr -p ack -s 45 -d 67 -c blue
$ns_ at $appTime2 "$node_(71) add-mark m3 green4 circle"
$ns_ at $appTime2 "$node_(74) add-mark m4 green4 circle"
$ns_ at $appTime2 "$ns_ trace-annotate \"(at $appTime2) ftp traffic from node 71 to
node 74\"""
Mac/802_15_4 wpanNam FlowClr -p tcp -s 71 -d 74 -c green4
Mac/802_15_4 wpanNam FlowClr -p ack -s 74 -d 71 -c green4
}

# defines the node size in nam
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ initial_node_pos $node_($i) 2
}

# Tell nodes when the simulation ends
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ at $stopTime "$node_($i) reset";
}

$ns_ at $stopTime "stop"
$ns_ at $stopTime "puts \"\\nNS EXITING...\\n\"""
$ns_ at $stopTime "$ns_ halt"

proc stop {} {
    global ns_ tracefd appTime val env
    $ns_ flush-trace
    close $tracefd
    set hasDISPLAY 0
    foreach index [array names env] {
        #puts "$index: $env($index)"
        if { ("DISPLAY" == "DISPLAY") && ("env($index)" != "") } {
            set hasDISPLAY 1
        }
    }
    if { ("wpan_demo3" == "wpan_demo3") && ("hasDISPLAY" == "1") } {
        exec nam wpan_demo3.nam &
    }
}

puts "\\nStarting Simulation..."
$ns_ run

```

## B – Perl Programming

**1-Calculate energy**

Calculate-energy-2.pl

#!/usr/usc/bin/perl

```

#($#ARGV ne 0) || die "[USAGE] cbr-throughput.pl trace_file\n";

```

```

printf "%s\n", $ARGV[0];
open (INPUT, "<$ARGV[0]");
open (OUTPUT, ">sink_received");

```

```

@conn="";
$dest=-1;
$source=-1;
$sport = -1;
$dport = -1;
$seq=-1;

```

```

$num_received=0;
$num_dropped=0;
$num_send=0;

```

```

@srcid="";
@dstid="";
$num_conn_rcv=0;
$num_conn_send=0;

```

```

@energy="";
@nodeid=( 0_, 1_, 2_, 3_, 4_, 5_, 6_, 7_, 8_, 9_, 10_, 11_, 12_, 13_, 14_, 15_,
 16_, 17_, 18_, 19_, 20_, 21_, 22_, 23_, 24_, 25_, 26_, 27_, 28_);
#, 29_, 30_, 31_, 32_, 33_, 34_, 35_, 36_, 37_, 38_, 39_, 40_, 41_, 42_, 43_,
44_, 45_, 46_, 47_, 48_, 49_, 50_, 51_, 52_, 53_, 54_, 55_, 56_, 57_, 58_, 59_,
60_, 61_, 62_, 63_, 64_, 65_, 66_, 67_, 68_, 69_, 70_, 71_, 72_, 73_, 74_, 7
5_, 76_, 77_, 78_, 79_, 80_, 81_, 82_, 83_, 84_, 85_, 86_, 87_, 88_, 89_, 90_,
91_, 92_, 93_, 94_, 95_, 96_, 97_, 98_, 99_, 100_);

```

```

while (<INPUT>)
{
    next if !length($_);
    @things = split;

    #if( ($things[0] =~ /N/) && ($things[2] =~ /51.000/) )
#    {
        $source = $things[2] ;
        $energy[$source] = $things[13] ;
    }
}

```

```

#      }
}

#####
# Print Connection throughput      #
#####
$total_send=0;
$total_rcv=0;
$total_energy=100.0 * ($#nodeid+1);
##printf "total energy = %f\n", $total_energy;

#for ($i=0; $i< 101 ; ++$i)
#{
# $total_remain = $total_remain + $energy[$i];
#}
#printf "total remain energy =%f\n", $total_remain;

foreach $nodeid (@nodeid)
{
    $total_energy -= $energy[$nodeid];

    #printf "total energy remain = %f\n", $total_energy;
}

##printf "total energy = %f\n", $total_energy;

#printf "Aggregated send = %d\n", $total_send;
#printf "Aggregated rcv = %d\n", $total_rcv;

close INPUT;
close OUTPUT;

$out_file='./energy_consumption.gp';
open(result,">>$out_file");
print result "\n";
print result "$total_energy";

```

## 2- Claculate end-end delay and paket lost

```

anylizeHTR.pl

#!/usr/bin/perl

$infile='./test.tr';
print "$argc \n";
if ($#ARGV == 0)
{

```

```

$infile=$ARGV[0];
print "$infile \n"
}
$G=0;
$S=0;
$D=0;
$Gmac=0;
$forward=0;
$sendno=0;
$drop=0;
$Time_simulation=500;
my @PktNumber=();
my @PktTime=();

$Size=80;
$tmp=0;
open(DATA, "<$infile");
while(<DATA>)
{
    @x = split(' ');
    #for ($i=0; $i<5 ;$i++)

        #print "$x[6]-";

    #print "$tmp \n";
    #tmp++;
    if ($x[1] >=40.0)
    {
        if ($x[0] eq 'f')
        { $forward = $forward+1;
        }
        if($x[0] eq 'D')
        { $drop=$drop+1;
        }
        if (($x[0] eq 's') && ($x[3] ne 'AGT'))
        { $sendno=$sendno+1;
        }
        if ($x[0] eq 's')
        {
            #print "send paquet \n";
            if ($x[3] eq 'AGT')
            {

                $G=$G+$Size; # Calcul de G

                # Normalement ce ci est une fonction que je dois écrire
                $exist=1;
                $nombre=$#PktNumber+1;
            }
        }
    }
}

```

```

# print "nombre :$nombre \n";
for($i = 0; $i < $nombre ; $i++)
{
    if ($x[5] == $PktNumber[$i]) {$exist=0;}
    # print "test for existence \n";
}
#print "exist : $exist \n";
# vrai travail, pour le calcul de D
if ($exist == 1)
{
    push(@PktNumber, $x[5]);
    push(@PktTime, $x[1]);
    #print " Nombre de paquets après push : $#PktNumber \n" ;
}

}
else
{
    ###print "supposed mac frame \n";
    if ($x[6] eq 'cbr' ) { $Gmac=$Gmac+$Size;}
}
}
if ($x[0] eq 'r')
{
    #print "receive packet \n ";
    if (($x[3] eq 'AGT') && ($x[2] eq '_20_'))
    {
        #if (($x[2] eq '_15_') || ($x[2] eq '_8_') || ($x[2] eq '_36_') || ($x[2] eq '_29_') || ($x[2]
eq '_42_'))
# || ($x[2] eq '_42_') || ($x[2] eq '_32_') || ($x[2] eq '_35_') || ($x[2] eq '_68_') || ($x[2] eq
'_45_'))
# || ($x[2] eq '_25_') || ($x[2] eq '_77_') || ($x[2] eq '_66_') || ($x[2] eq '_27_') || ($x[2] eq
'_44_') || ($x[2] eq '_7_') || ($x[2] eq '_49_') || ($x[2] eq '_8_') || ($x[2] eq '_68_') || ($x[2]
eq '_24_') || ($x[2] eq '_57_') || ($x[2] eq '_42_') || ($x[2] eq '_12_') || ($x[2] eq '_58_') ||
($x[2] eq '_79_'))

        #print "AGT and 0 packet \n";
        $S=$S+$Size; #calcul de S
        #calcul de D
        $test=1;
        $i=0;
        #print " Nombre de paquets: $#PktNumber \n" ;
        $nombre=$#PktNumber+1;
        while(($i<=$nombre) && ($test == 1))
        {
            #print " $PktNumber[$i] - $x[5] \n";

            if ($PktNumber[$i] == $x[5])

```





```

$t=(SS/$G)*100;          #delivary ratio throughput/load
print result "$t ";
$out_file='./routing_overhead.gp';
open(result,">>$out_file");
print result "\n";
$t=($forward/$sendno); #routing overhead no of f packet /no of send packet
print result "$t ";
$out_file='./Average_throughput.gp';
open(result,">>$out_file");
print result "\n";
$t=(SS/500);            #Average throughput bit/sec
print result "$t ";
$out_file='./Average_load.gp';
open(result,">>$out_file");
print result "\n";
$t=($G/500);           #Average load bit/sec
print result "$t ";
$out_file='./number_Drop.gp';
open(result,">>$out_file");
print result "\n";
$t=$drop;              #number of drop packets
print result "$t ";
$out_file='./number_forward.gp';
open(result,">>$out_file");
print result "\n";
$t=$forward;          #number of forward packets
print result "$t ";
$out_file='./number_contol_packet.gp';
open(result,">>$out_file");
print result "\n";
$t=$sendno;           #number of contol packets
print result "$t ";

```

### 3- calculate Average hop count

```

Calculate-hop.pl
#!/usr/bin/perl

$infile='./test.tr';
print "$argc \n";
if ($#ARGV == 0)
{
    $infile=$ARGV[0];
    print "$infile \n"
}
$G=0;
$S=0;

```

```

$D=0;
$Gmac=0;
$shop=0;
$gg=0;
$count=0;
$forward=0;
$sendno=0;
$drop=0;
$bb=0;
$nbrbyte=0;
$Time_simulation=500;
$counter=0;
my @PktNumber=();
#my @PktTime=();
# Mostafa
my @NodID=();
#my @two= [[]];

$Size=80;
$tmp=0;
open(DATA, "<$infile");
while(<DATA>)
{
    @x = split(' ');

    if (($x[0] eq 's') && ($x[3] eq 'MAC') && ($x[6] eq 'cbr'))
    {
        $nbrbyte= $nbrbyte + $x[7];
    }
    if (($x[0] eq 's') && ($x[3] eq 'AGT'))
    {
        $counter=$counter + $Size;
    }
}
$out_file='./AveHop.gp';
open(result,">>$out_file");
print result "\n";
#print "tmp = $tmp \n";
$t=($nbrbyte/$counter);          # normalize load
print result "$t ";

```

## ÖZGEÇMİŞ

Ahmed Efendiođlu, 10 Temmuz 1982’de Irakta Musul’da doğdu. Ortaokul ve liseyi Telafer Lisesi’nde tamamladıktan sonra, Eylül 2002’de Musul Üniversitesi Bilgisayar Mühendisliđi lisans programına başladı. 2006 yılında bu bölümü başarıyla tamamladı, Ahmed, 2008 Eylül ayında İstanbul Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliđi Anabilim Dalı’nda yüksek lisans yapmaya hak kazandı. Ahmet mevcut durumda tez aşamasındadır ve tez konusu “Kablosuz Sensor Ağlarında Kümeleme Ve Yönlendirme Algoritma Analizi’dir. Arapça ve İngilizce yabancı dil biliyor.

Ahmed’in kullandığı programlama dilleri; C,C++, C#, Java, Omnet++,matlab, Web Geliştirme arayüzleri HTML, JavaScript, PHP, ASP. net’tir.