

T.C.
GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

SCATTERNET OLUŞUMU VE
SCATTERNET ÜZERİNDE ADRESLEME
İÇİN UYGULANABİLİR BİR YAKLAŞIM

Sami ALTUNDAĞ
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

GEBZE
2006

**T.C.
GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ**

**SCATTERNET OLUŞUMU VE
SCATTERNET ÜZERİNDE ADRESLEME
İÇİN UYGULANABİLİR BİR YAKLAŞIM**

**SAMİ ALTUNDAĞ
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI**

**TEZ DANIŞMANI
Mehmet GÖKTÜRK**

**GEBZE
2006**

ÖZET

**TEZ BAŞLIĞI: Scatternet Oluşumu Ve Scatternet Üzerinde Adresleme
İçin Uygulanabilir Bir Yaklaşım**

YAZAR ADI: Sami ALTUNDAĞ

Bluetooth, veri, ses ve görüntü taşıyarak farklı türden uygulamalara hizmet vermeyi amaçlayan bir Mobil Tasarımsız Ağ (Mobile Ad-Hoc Network, MANET) türüdür. Uygulamalar ve çalışmaları için gerekli standartlar, profiller halinde belirtilmiştir (Bkz. Ek 1). Bluetooth, çalışmak için ortamda hiçbir altyapı gerektirmez. Bunun yanında bilgisayarlar ve telefonlar başta olmak üzere artık yaygın olarak kullanılmaktadır. Bu iki sebep, başlangıçta planlanan kullanım senaryolarının yanında, Bluetooth'u anlık ve altyapısız bağlantı gerektiren başka bağlantı ihtiyaçlarını karşılamaya aday göstermiştir. En küçük Bluetooth ağ birimi olan piconetleri birbirine bağlayarak scatternet oluşturmak ve böylece Bluetooth ağlarını büyütmek, bu tür ihtiyaçları karşılamaya yönelik çalışmaların bir bölümüdür. Bu tezde, yeni bir scatternet formasyon algoritmasını ve uygulamasını sunmayı amaçlamaktayız. Yeni bir algoritma ortaya konulmasının sebebi, varolan algoritmaların bahsedilecek senaryolarda kullanılamayacak kadar uzun süren bir formasyon sürecine sahip olmaları ve/veya Bluetooth standartlarında olmayan fonksiyonlar üzerinde çalışmalarıdır. Formasyon yanında, varolan tasarımsız ağlardaki adresleme yöntemlerinin, kurulan scatternet üzerinde çalışabilmesi için gerekli bileşenler de sunulacaktır.

Bu tez çalışması dört bölümden oluşmaktadır. İlk bölümde konu hakkında genel bilgi verilmektedir. İkinci bölümde, Bluetooth hakkında temel bilgiler verilmektedir. Üçüncü bölümde, sensör ve Tasarımsız kablosuz ağlar için öne sürülen protokoller anlatılmaktadır. Dördüncü bölümde scatternet formasyonu için yapılan çalışmalar özetlenmektedir. Beşinci bölümde, geliştirilen scatternet formasyon protokolü ve geliştirilen uygulaması anlatılmaktadır. Son bölümde ise yapılan çalışmanın değerlendirmesi ve gelecekte yapılması planlananlar anlatılmaktadır.

SUMMARY

TITLE OF THE THESIS: A Practical Approach to Scatternet Formation and Routing on Bluetooth

AUTHOR: Sami ALTUNDAĞ

Bluetooth is an emerging connectivity technology. Its profile based applications are evolving. However many of these applications are working on one hop scenarios. There are many types of applications where multi hop scenarios will be useful. To route messages between piconets, we need interval nodes running on more than one piconet. These nodes should be able to forward messages based on the destination which the packet contains. If more than one piconet are connected and messages are routed between them, this network is called as Scatternet.

In order to form scatternets, a number of propositions have been made [8, 11, 14, 17, 24] that form a graph of nodes to connect piconets. However, combined with bluetooth constraints, these protocols are slow and inefficient to apply to many application types. For example when an advertisement needs to be sent only once to the every node around, forming a graph will not be a good solution, since bluetooth inquiry and preoperation communication of algorithms together, will create a considerable overhead. By preoperation, we mean transmission of any kind of handshake information between nodes to calculate the best way of forming the graph. It is precommunication since data transmitted at this phase of communication is completely unrelated to the information that needs to be sent.

In this paper, we introduce a practical protocol inspired by existing Scatternet formation protocols, Sensor Network routing protocols, and Ad hoc routing protocols in order to establish a scatternet and route messages between piconets. We have built everything on existing Bluetooth 2.0 standards. In addition, an application is implemented using today's mobile phone Bluetooth features and user experiences of the developed application are given. The algorithm has a quick and efficient precommunication time when a message needs to be sent to every node on the network. If a message needs to be routed to a destination instead of the whole network, algorithms like AODV can be executed on top of the protocol easily. Our

algorithm can be useful for various types of applications. Sending out advertisements, multi-hop Bluetooth service discovery, and sending non-stream data packages are among them. If a stream needs to be sent, a tunnel can be formed by running AODV on top of the protocol again. When talking about streaming between piconets Quality of Service (QoS) considerations comes to mind, however not every applications has QoS requirements.

This paper is organized as follows: in section 1, an introduction will be given. In section 2, basics of Bluetooth are given. In section 3, we discuss the related work on Sensor and Ad-hoc Networks. In section 4, we discuss the related research on Bluetooth scatternet and challenges related with scatternet formation. We present the algorithm in section 5. A conclusion is presented in the last section.

TEŞEKKÜR

Danışmanım Yrd. Doç. Dr. Mehmet Göktürk’e desteği, akademik konulardaki rehberliği ve Bluetooth Chat Room fikri için teşekkür ederim.

Sami Altundağ

İÇİNDEKİLER DİZİNİ

	<u>Sayfa</u>
ÖZET	iv
SUMMARY	v
TEŞEKKÜR	vii
İÇİNDEKİLER DİZİNİ	viii
ŞEKİLLER DİZİNİ	x
TABLolar DİZİNİ	x
1. GİRİŞ	1
2. ÖNBİLGİ	2
2.1. Bluetooth ve Diğer Kablosuz Ağlar	2
2.2. Bluetooth Haberleşme Yığını	3
2.3. Bluetooth Haberleşme Kanalları	5
2.4. Bluetooth Paket Yapısı	6
2.5. Bluetooth Frekans Sıçrama Spektrumu	7
2.6. Servis Konumlandırma	8
2.7. Bluetooth Ve Servis Bulma	11
2.8. JSR 82	13
2.9. Bluetooth Scatternet Ve Adresleme	14
3. KONUS İLE İLGİLİ YAYINLAR VE ÇALIŞMALAR	17
3.1. Tasarımless Ağlar ve Adresleme	17
3.2. Sensör Ağlar ve Adresleme	20
3.2.1. Flooding	20
3.2.2. Gossiping	20
3.2.3. SPIN	20
3.2.4. LEACH	21
3.2.5. Directed Diffusion	21
3.2.6. Rumor Adresleme	22
3.2.7. Minimum Cost Forwarding	22
3.2.8. COUGAR	23
3.3. Scatternet Formasyon Algoritmaları	23

3.3.1. A New Bluetooth Scatternet Formation Protocol	23
3.3.2. BlueMesh	23
3.3.3. BlueStar	24
3.3.4. Sensör Bilgisi Toplama	25
3.3.5. Bluetrees	26
4. PROTOKOL	28
4.1. Algoritma	29
4.2. Kullanım Senaryosu 1	34
4.3. Kullanım Senaryosu 2	35
4.4. Bir Uygulama: BLUECHAT	36
4.4.1. Bileşenler	36
5. SONUÇ	42
KAYNAKLAR	44
ÖZGEÇMİŞ	47
Ek1 : Bluetooth Profilleri	
Ek 2: Bluechat için Kullanıcı Yorumları	

ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
2.1. Bluetooth haberleşme yığını	4
2.2. Bluetooth spesifikasyonu V 1.2	6
2.3. Bluetooth üye bağlantı durumları	8
4.1. Algoritma	34
4.2. Scatternet örneği	35
4.3. Uygulama mimarisi.	37
4.4. Scatternet Formasyon Örneği	39
4.5. Uygulama önyüzünden örnekler	41

TABLolar DİZİNİ

<u>Tablo</u>	<u>Sayfa</u>
2.1. Paket Bölümleri	6
2.2. Kullanılacak Bluetooth Service Özellikleri	12

1. GİRİŞ

Bluetooth veri, ses, görüntü, video uygulamaları gibi birçok türden uygulamaya hizmet vermeyi amaçlayan bir Mobil Tasarımsız Ağ (Mobile Ad-Hoc Network, MANET) türüdür. Kullanım türleri her gün artan Bluetooth, artık mobil cihazlar ve bilgisayarların standart bir parçası haline gelmektedir. Bluetooth bağlantıları kablo gibi ek bağlantı aparatları gerektirmiyor. Bu yönden Bluetooth'un kısa mesafeli ağsız bağlantı kurmada sağladığı avantajlar açıktır. Bu sebepler, başlangıçta planlanan kullanım senaryolarının yanında, Bluetooth'u anlık ve altyapısız bağlantı gerektiren başka bağlantı ihtiyaçlarını karşılamaya aday göstermiştir. Ancak Bluetooth, bağlantının kısa mesafeli olması, her piconet içinde maksimum 7 aygıt olması, çok sekmeli senaryoların spesifikasyonda belirtilmemesi gibi kısıtlara sahiptir. En küçük Bluetooth ağ birimi olan piconetleri birbirine bağlayarak scatternet oluşturmak ve böylece Bluetooth ağlarını büyültmek, bu tür ihtiyaçları karşılamaya yönelik çalışmaların bir bölümüdür. Scatternet oluşumu üzerine çalışmalar olmasına karşılık, geliştirilen scatternet algoritmaları ya standartlar dışında öğeler kullanmakta ya da birçok kullanım senaryosu için yavaş kalmaktadır. Örneğin Konferans odasında bilgi paylaşımı, çok sekmeli servis reklamları dağıtımı gibi konularda ileride detayları verilecek olan topolojilerin çoğunun kurulumu zaman, enerji ve frekans kaynaklarını boşa harcamaktadır. Bu çalışmada topoloji kurulumunu daha hızlı yapan, sensör ve tasarımsız ağlardaki veri dağıtımında kullanılan yöntemlerden avantaj sağlayan bir scatternet formasyon algoritmasını ve uygulamasını sunmayı amaçlamaktayız. Formasyon yanında, varolan tasarımsız ağlardaki adresleme yöntemlerinin, kurulan scatternet üzerinde çalışabilmesi için gerekli bileşenler de sunulacaktır.

2. ÖNBİLGİ

2.1 Bluetooth ve Diğer Kablosuz Ağlar

Günümüz kablosuz ağları üç kategoriye ayrılmıştır. Bunlar:

1. Hücresel Ağlar,
2. Tasarımsız Ağlar,
3. Sensör Ağlar adlarını alırlar.

Herbir ağ kategorisinin ihtiyaçları farklıdır. Bu farklılıklar OSI katmanlarından (Physical, MAC, Network, Transportation) herhangi birinde veya hepsinde olabilir. Örneğin sensör ağlar hızlı biçimde belki hiç ağ bağlantısı kurmadan, yayma yöntemi ile veriyi hedefine ulaştırmayı amaçlarken, hücresel ağlarda merkez istasyonlar ve haberleşme noktaları arasında sıkı bir ilişki vardır.

Bluetooth'un tasarımsız ağ oluşturma, servis bulabilme ve yaygın olma özelliklerinden dolayı yakın zamanda her yerde varolan ve özel bir altyapı gerektirmeden kullanılabilen bilişim servislerinde kullanılması olasıdır. Bluetooth dahilinde en küçük haberleşme birimi piconet adını alır. Piconetlerin bir araya gelmesi ile oluşan ağa scatternet adı verilir. Bluetooth kullanım senaryoları, profiller halinde Bluetooth spesifikasyonunda belirtilmiştir (Bkz. Ek 1). Profiller, Bluetooth aygıtlarının belirtilmiş servisleri birlikte verebilmeleri için gerekli temel gereksinimleri ve standart prosedürleri belirlerler. Profiller sadece önceden belirtilmiş kullanım senaryoları olup, Bluetooth kullanımı bu sayılan profiller sınırlı değildir. Bu profillerden SDAP ve GAP her Bluetooth aygıtının vermesi gereken zorunlu servisleri tanımlar. GAP (Generic Access Profile), aygıtların birbirini bulması, link yönetimi, güvenlik seviyesi prosedürleri gibi temel işlevleri tanımlar. SDAP (Service Discovery and Access Profile) ise GAP üzerinde çalışır ve bir Bluetooth aygıtının sahip olduğu servislerini bulmada kullanılır. Yalnız varolan bluetooth kullanımı tek sekme senaryolar yani bir piconet içinde çalışmak üzerine

kurulmuştur. Bu kadar yaygın bir teknolojiden gelen faydayı arttırmak için, Bluetooth'un birden çok sekme üzerinde çalışmasını sağlamak amacıyla yapılan birçok Scatternet oluşumu algoritmaları mevcuttur. Ancak bu algoritmalar genelde Bluetooth standartları içinde yeralmayan özellikler kullanmakta veya kurulum sırasında çoğu uygulama tipi için yavaş kalmaktadırlar. Bizim amacımız, sensör ve hücre ağlarda kullanılan birden çok sekme üzerinde adresleme senaryolarını ve şu ana kadar geliştirilen Scatternet formasyon tekniklerini temel alarak, tamamen standartlar üzerine kurulu ve hızlı bir formasyon geliştirmektir. Bu yöntem ile, Bluetooth noktaları uzak çevrelerindeki servisleri bulabileceklerdir veya uzak çevrelerdeki noktalar ile birlikte çalışabileceklerdir. Bütün bunlara ek olarak belirtmek gerekir ki, burada asıl yapılmak istenen tasarısız adresleme değildir. Ancak Tasarısız adresleme geliştirilen algoritma üzerine kurularak çalışabilir. Tüm bunlara ek olarak ağ içinde dağıtılan bilginin içeriği önemli değildir ancak iletilen verinin belli bir Quality of Service (QoS) gerektirmesi durumunda veya ağ kaynaklarını zorlayacak kadar büyük olması durumunda Bluetooth ağlarının scatternet performans kısıtları gözönünde tutulmalıdır. Scatternet ile ilgili performans ve QoS problemleri ise bölüm 3 de anlatılmaktadır.

2.2 Bluetooth Haberleşme Yığını

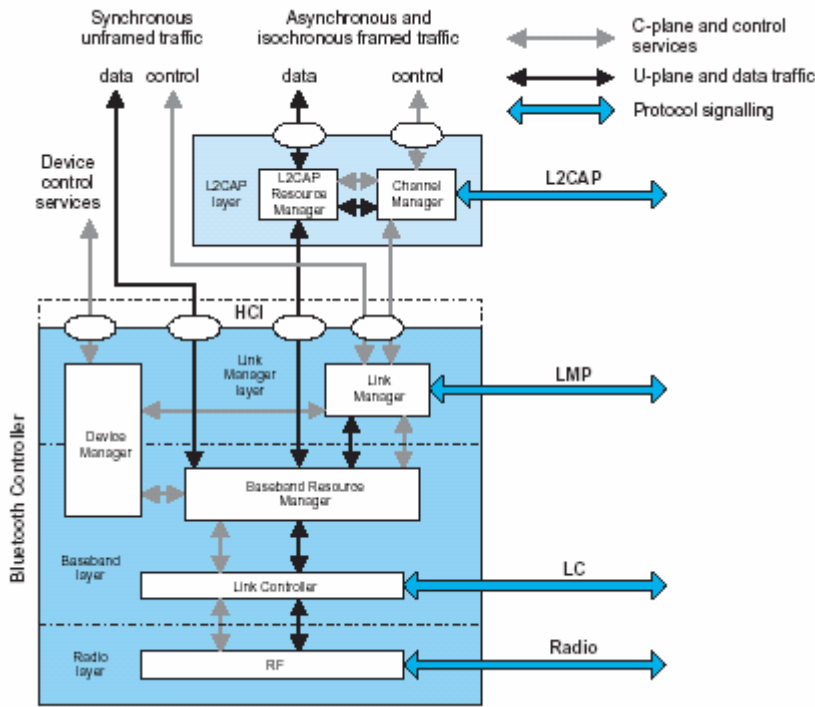
İncelencek konu Bluetooth scatternet oluşumu olduğu için, varolan bluetooth iletişim protokolünün scatternet oluşumu için kullanılacak kısımlarının temel niteliklerini ve yeteneklerini bilmek gerekli. Aşağıdaki bölüm, bu fikirden hareketle bluetooth protokolünün iletişim kanallarını ve bu kanallar üzerinden yapılabilen iletişim türlerini anlatmayı hedeflemektedir.

Bluetooth haberleşme yığını, OSI katmanlarına yakın bir yapıya sahiptir. Katmanların birbiri ile çalışabilirliğini arttırmak için, yığın Bluetooth Controller ve Bluetooth Host olmak üzere iki kısma ayrılmıştır. Bu ayrım yapılan çalışmalar için de önemlidir çünkü donanım seviyesinde gerçekleştirildiği için alt katman üzerinde değişiklik yapmak mümkün olmasa da, genellikle yazılım seviyesinde gerçekleştirildiği için üst katmanı değiştirmek çoğunlukla mümkündür. Üst

katmanlar için açık kaynak kodlu haberleşme yığınları bulmak da oldukça kolay. Bunların bazılarını listelemek gerekirse:

www.bluez.org (Resmi Linux Bluetooth haberleşme yığını)

www.JavaBluetooth.org (Java tabanlı Bluetooth haberleşme yığını)



Şekil 2.1. Bluetooth haberleşme yığını (Bluetooth Spesifikasyon Dökümanından alınmıştır)

Bluetooth ile birden çok sekme üzerinden haberleşme olanaklarını değerlendirebilmek için, Piconet çalışma prensiplerini, haberleşme için kullanılan kanalları, yayım yapma olanaklarını ve özellikle Bluetooth üzerinde adreslemenin nasıl yapıldığını bilmek gerekli.

Bluetooth, sensör ağların aksine, haberleşme için öncelikle bir bağlantı kurulmasını gerektirir. Bluetooth cihazları inquiry prosedürünü kullanarak önce etraflarında diğer aygıtları bulurlar. Haberleşmek isteyen aygıtlar ise page scan ile bağlantı kurabilirler.

Bağlantı kuran aygıtlar piconet oluşturur. Bir piconet'te sadece bir aygıt yönetici olur. Piconetin saati yöneticinin saatine senkronize olur. Bu yüzden de bir yönetici sadece bir piconet içinde bulunabilir. Üyeler ise zaman paylaşımı ile birden fazla piconet ile aynı anda haberleşebilir. Birden fazla piconet ile oluşan ağ türüne ise scatternet adı verilir.

Bir piconet oluşturulduğunda her üye 3 bitten oluşan LT_ADDR atanır. Yöneticinin LT_ADDR yoktur ve 000 yayın yapmak için kullanılan LT_ADDR'dir.

2.3 Bluetooth Haberleşme Kanalları

Bu bölümde ise bluetooth üzerindeki haberleşme kanallarının temelleri anlatılmaktadır.

Bluetooth üzerinde çoğu ağsız ağda olduğu gibi user ve kontrol verilerini taşımak için farklı sinyaller kullanılır. Bu sinyal çeşitleri aşağıdaki gibidir:

ACL-U: Frame yapısını kullanarak veri transferi yapmak içindir yani L2CAP sinyallerini taşır.

ACL-C: Link Management Protocol sinyallerini taşır.

PSB: Park etmiş üyelere yayın yapmak için kullanılır

ASB: Aktif üyelere yayın yapmak için kullanılır

SCO: Frame yapısını kullanmadan veri taşınması için kullanılır. Eğer gönderilen bilgi ham akım ise veya belli bir performans gerektiriyor ise veya frame kullanmıyor ise, uygulamalar ACL-U kullanmak yerine SCO kullanmayı tercih edebilir. Bu uygulamalara, ses taşıyan uygulamalar örnek gösterilebilir.

Logical Link Control and Adaptation Protocol (L2CAP), haberleşme için kullanılan temel veri transfer protokolüdür. L2CAP ACL linkleri üzerinde çalışır; yani veri iletişimi için frame kullanır. L2CAP bağlantıları, ses ve görüntü taşımak için kullanılamaz. RFCOMM ise, L2CAP üzerinde çalışan bir başka veri transfer

protokolüdür. RFCOMM, RS-232 seri port emulasyonu yapar ve kullanan sisteme akım olarak veri gönderildiği izlenimi verir.

Unicast Linkler

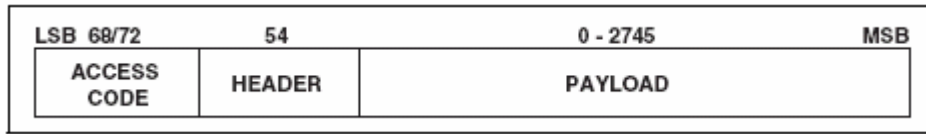
Bu tür linkler, veri transferinden önce bir bağlantı prosedürü gerektirir yani bağlantı tabanlıdır. Genellikle bağlantı tabanlı L2CAP kanalları kullanılır.

Yayın Yapılan Linkler

Bağlantısız linklerdir. Veri trafiği yöneticiden üyeye doğru tek yönlüdür ve güvenilir değildir. Aktif üyelere L2CAP verisini yaymak için ASB, park etmiş üyelere yayım yapmak için ise PSB kullanılır.

2.4 Bluetooth Paket Yapısı

Bluetooth genel paket formatı aşağıdaki gibi Access Code, Header, Payload olmak üzere üç bölümden oluşur.



Şekil.2.2. Bluetooth spesifikasyonu V 1.2 Sayfa 97:

Her bölümün uzunluğu aşağıdaki gibidir.

Tablo.2.1. Paket Bölümleri

Access code	72 veya 68 bit
Header	54 bit
Payload	0 - 2745 bit (343 Byte)

Çeşitli paket türlerinde aşağıda belirtilen bölümler olabilir.

- Access Code
- Access code ve Header
- Access code, Header ve Payload.

Her Bluetooth aygıtının bir bluetooth adresi vardır ve bu 48 bit uzunluğudur. Kullanıcı arayüzünde ise bu 12 tane hexadecimal karakter olarak gösterilir.

Örnek:

000C3E3A4B69

00:0C:3E:3A:4B:69

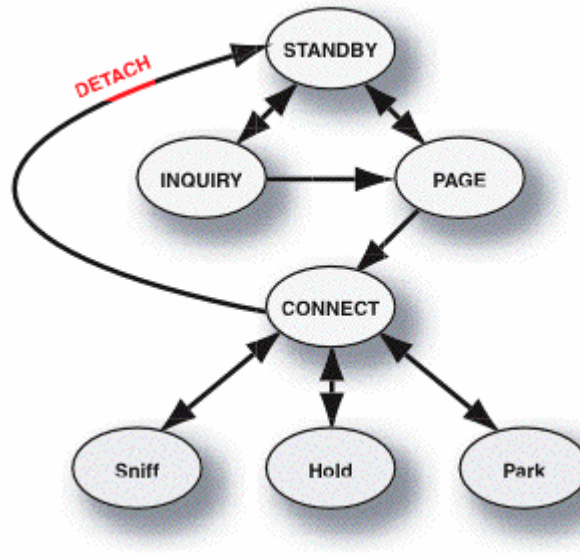
2.5 Bluetooth Frekans Sıçrama Spektrumu

Bluetooth'un, frekansların birbirine karışma ihtimalinin yüksek olacağı ortamlarda çalışacağı varsayıldığından, bir tek kanal üzerinden haberleşmek riskli görülmüştür. Bunun yerine Bluetooth, frekanslardaki karışımı engellemek için bir frekans sıçrama spektrumu kullanır; yani haberleşme için kullandığı kanalları bir sıralama çerçevesinde değiştirir. Bu değiştirme saniye 1600 defa tekrarlanır ve sıçrama parametreleri paging aşamasında yönetici tarafında üyelere iletilir. Bu parametreler:

- 1 Kullanılacak kanallar
- 2 Kullanılacak kanallar arasındaki sıçrama sırası
- 3 Senkronizasyon için saat
- 4 Tarafların gönderim/alım zamanlarıdır

Aralarında sıçrama yapılan 79 adet kanal vardır ve herbir kanal 1MHz bant genişliğine sahiptir. İlk kanal 2402 MHz'dedir.

Inquiry ve paging süresince ise sıçrama sırası daha farklıdır. Inquiry için kullanılan 32 adet kanal vardır. Inquiry yapan bir yönetici saniyede 3200 defa kanallar arasında frekans değiştirir. Inquiry scan yapan bir üye ise, dinlediği kanalı 1.28 saniyede bir defa değiştirir. Paging süresince ise 32 adet kanal üzerinde üyenin Bluetooth adresi ile belirlenen sıçrama sıralaması ile haberleşirler.



Şekil.2.3. Bluetooth üye bağlantı durumları (Bluetooth Spesifikasyon Dökümanı'ndan alınmıştır)

2.6 Servis Konumlandırma

Konumuz uzun zamandır üzerinde uğraşılan bir alan ile yani servis konumlandırma ile yakından ilgilidir. Heryeri kaplayan ve her yerde olan bilişim servislerinde, aygıtların kullanıcı müdahalesine gerek kalmadan çevredeki servisleri bulması ve bu servisleri kullanabilmesi gereklidir. Servis bulmak üzerine kurulmuş teknolojileri ise iki grup altında toplamak mümkündür. İlk grup merkezi yapı üzerine kurulanlar; ikinci grup ise tasarımısız servis arama yöntemleridir [Rats'mor et al, 2004]. JINI ve UDDI merkezi servis arama yöntemlerindendir. Bluetooth ise kendi içinde Tasarımısız bir servis bulma mekanizmasına sahiptir. Yalnız Bluetooth'un servis arama mekanizması tek sekmeli kullanım senaryolar üzerine kuruludur. Bu da kullanımı bluetooth sinyallarının ulaşabileceği kısıtlı bir alana (yaklaşık 10m.) ve bir bluetooth noktanın bağlantı içinde olabileceği maksimum çevre nokta sayısına (çoğunlukla 7) hapsetmektedir.

JINI, java teknolojisine dayanan ağ üzerinde servis bulma mekanizmalarından biridir. JINI içinde servis veren aygıtlar, sağlayıcı adını alır. Servisleri bulmak için kullanılan servisler ise arama servisi adını alır. Bir sağlayıcı ağa katılmak istediğinde, önce arama servisleri için yayın yapar. Bu yayın karşılığında, bulunan arama servisleri, kayıt olmak isteyen sağlayıcıya kayıt işlemini yapabileceği bir vekil gönderir. Vereceği servis için bir programcık ve özellikler listesi hazırlamış olan sağlayıcı, bulunan arama servisine bu programcığını ve özellikler listesini yollar. Bunun ardından, sağlayıcı ve arama servisi arasında bir canlı tutma protokolü çalışmaya başlar. Ağa bu servisi kullanmak isteyen bir istemci katıldığında ise şu adımlar gerçekleşir. İstemci, aynı sağlayıcı gibi önce yerel ağ içindeki arama servislerini yayın ile buluyor. İstemci, bunun ardından arama servisine istenen servis tipini ve istenen servis özelliklerini gönderir. Arama servisi ise bu isteğe uyan servis programcıklarını bir liste halinde istemciye iletir. İstemci ise bulunan listeyi kullanıcıya bir seçim yapması için sunar. Seçimin ardından, programcık, ayarlar için kullanıcıya bir dialog gösterebilir. Ardından çalıştırılan servis, tüm işini lokal yapabilir veya RMI kullanarak uzak bir makinede işlemi gerçekleştirebilir veya bunların karışımı olabilir. Örneğin servisi veren bir yazıcı ise, yazıcı önce ağ içinde arama servislerini bulur ve arama servislerine, yazıcı servisini kaydettirir. Bu aşama, servisi kullanmak isteyenlerin lokallerine indirecekleri programcığı da gönderir. Ağ üzerinde herhangi bir cihaz, yazıcı bulmak istediğinde, arama servisleri cihaza bir yazıcı listesi gönderir. Cihaz ise, bu listeyi kullanıcıya gösterir. Kullanıcı listeden bir yazıcı seçtikten sonra, programcık cihaza indirilir ve çalıştırılır. Yazdırma işlemi ise, programcığın yazıcı ile iletişim kurması ile gerçekleşir.

“A Ubiquitous Stable Storage for Mobile Computing Devices” [Burge et al, 2001] kablosuz ağ üzerinde merkezi servis bulma çalışmalarına örnek gösterilebilir. Veri yedekleme veya depolama ihtiyacı duyan cihazlar, ortamdaki erişim noktalarından faydalanarak kablosuz ağ ile LAN üzerindeki yerel servis rehberine ulaşır. Eğer LAN içinde herhangi bir depolama cihazı var ise, yerel servis rehberi, depolama cihazının adresini depolama ihtiyacı duyan cihaza geri gönderir. Ardından depo ile depolama ihtiyacı duyan cihaz arasında depolama işlemi başlar. Depolama cihazı ise, gönderilen veriyi Internet üzerinden asıl hedef olan cihaza gönderir. Bu şekilde veri depolama ihtiyacı duyan cihaz, kullanıcı müdahalesi olmadan depolama işlemini gerçekleştirir.

“Bluetooth and Sensor Network: A Reality Check” [Leopold et al, 2003] Bluetooth’un sensör ağlar uygulamaları için kullanabilirliğini araştıran bir çalışmadır. Çalışmada bluetoothun sensör ağlarını oluşturmadaki başarısı, Bluetoothun kompleks yığın yapısı, bağlantı tabanlı iletişimi ve piconetler gözönüne alınarak değerlendirilmiştir. Adı geçen çalışma bluetooth üzerinden uzun mesafeli veri diffüzyonu yapmaya çalışması yönünden bu çalışma ile benzerlik taşımaktadır. Çalışmanın scatternet oluşturma bölümü BlueTree [Záruba et al, 2001] çalışmasından esinlenilmiştir. Bu çalışma içinde her nokta üzerinde iki bluetooth aygıtı bulunmaktadır. Bu aygıtlardan biri tüm bağlantılarında yönetici, diğeri ise üye olmaktadır. Noktalardan biri ayağa kalktıktan hemen sonra bluetooth aygıtlarından birini açmakta ve çevredeki diğerk bağlanılabilir bluetooth aygıtlarını aramaya başlamaktadır. Bu işlemi yaparken diğerk noktalar tarafından görünebilir durumda olmamakta ve kendisini bağlanabilmek için bir ağ arayan tek başına bir aygıt olarak görmektedir. Eğer bir aygıt bulursa, bu aygıta üye olarak bağlanmayı denemektedir. Bağlantı gerçekleşir ise de, kendisini artık ağın bir parçası olarak kabul etmekte ve diğerk bluetooth aygıtını henüz ağın parçası olmayan aygıtlardan gelecek bağlantı isteklerini kabul edebilmek için açmaktadır. Eğer üye olarak yapılmak istenen bağlantı gerçekleşemedi ise, karşıdaki aygıt 7 tane olan üye sınırını aşmış demektir. Bu durumda aygıt, bulduğu diğerk aygıtlardan birine bağlanmayı denemektedir. Eğer tüm bağlantı denemeleri başarısız oldu ise, aygıt ilk bağlanmaya çalıştığı aygıta tekrar bağlanmayı dener. Eğer yönetici aygıtlardan biri, üç defa aynı aygıttan bağlantı isteği almış ise maksimum üye sayısı sınırında olduğu için redetmiş ise, bağlı olduğu üye bağlantılarından birini koparır ve yeni bağlantı isteğini kabul eder. Eğer bluetooth aygıtlarından biri yönetici ile bağlantısını kaybeder ise, yeniden kendi üyelerine bağlantı yapma riski olduğundan, yeni bir yönetici arayışına girmez. Bunun yerine önce yönetici olan bluetooth aygıtının tüm bağlantılarını koparır. Bu durumda birim, tüm ağ bağlantılarını kaybetmiş ve başlangıç durumuna dönmüş olur. Bir sonraki adım olarak birim en başa geri döner bağlantı sürecini en baştan işletir. Toplanan bilgi dış dünyaya taşımak için ise aygıtlardan biri aracı olur. Aracı, hem oluşturulan ağacın kökü olma hem de toplanan veriyi dış dünyaya taşıma görevlerini üstlenir. Kök olabilmek için, üye bağlantılarını kuran bluetooth aygıtı kapalı ve yönetici bağlantılarını yapan aygıt açık olarak çalışmaya başlar. Kökteki birimin üye aygıtını açmaması ve diğerk birimlerin önce üye birimlerini açarak arama yapması, bir

ağa bağlantıktan sonra ise yönetici aygıtlarını açması bir ağaç oluşumunu mümkün kılar.

Bu durumda ağda sadece bir aracı çalışır durumdadır ancak birden fazla aracı olması arzu edileceğinden, diğer araçlar ağacın yaprakları olarak görev sıralarını beklerler. Yaprak olmayı da ağaca bağlandıktan sonra yönetici aygıtlarını kapatarak sağlarlar. Herhangi bir sebeple ağacın kökünde bir iletişim problemi olursa, tüm ağ yeniden kurulur ve yeni bir aracı seçilir.

2.7 Bluetooth Ve Servis Bulma

Bluetooth, iki tür arama/sorgulama işlemine sahiptir. Bunlardan ilki, çevre aygıtları bulan aygıt arama işlemidir. İkincisi ise verilen bir spesifik aygıt üzerinde varolan servisleri sorgulamaya yarayan servistir ve Service Discovery Protocol (SDAP) adını alır

Service Discovery Protocol içinde bir aygıt üzerinde servislerin özelliklerinin nasıl tutulması gerektiği ve sorgulanması gerektiği belirtilmektedir. Service Discovery Protokolünün temel bileşenleri şunlardır:

Service Record Handle: Bir nokta üzerindeki servise o makine üzerinde tekil bir kimlik vermek için kullanılır. 32 bit uzunluğunda bir numaradır.

Service Attribute: Servisler, özelliklerini bu komponentler aracılığı ile belirtirler. Her özellik Attribute ID, Attribute Value, Attribute Type olmak üzere üç bölümden oluşur. Attribute ID 16 bit uzunluğunda bir sayıdır. Attribute Value ise farklı türde değerler alabilir. Değerin türünü ise Attribute Type belirler. Attribute ID için kullanılan standart ile önceden belirlenmiş ID'ler vardır. Bunlardan bu çalışma için önemli olanları şunlardır.

Tablo.2.2. Kullanılacak Bluetooth Service Özellikleri

Attribute ID	Açıklama
x101	Servis Açıklaması.
x100	Servis Adı.
	Uygulama Değeri (Bknz. 3. bölüm): BluePages
x4	ProtocolDescriptorList.
	Servisin üzerinde çalışabileceği prokollerin listesi. Örneğin RFCOMM, L2CAP.
	L2CAP için kullanılan değer ise $0x100 = 256$ 'dir.
x1	Service Class ID List
	Uygulama Değeri (Bknz. 3. bölüm): AFE3D433E06D4CDB9C8524B4A6DCDCEB
x0	Service Record Handle

Servis sorgulaması aşamasında, sorgulamayı yapan nokta, istediği kriterleri bir değer listesi şeklinde iletişimde olduğu aygıtlara yayar. Kriterler ve aygıtların üzerindeki servislerin uygunluğu, verilen değer listesindeki tüm değerlerin aranan aygıtların üzerinde herhangi bir servisin özelliklerinin bir alt kümesinde bulunabilmesi ile olur. Karşılaştırmada sadece Attribute Value karşılaştırılır. Attribute ID, karşılaştırma sırasında kullanılmaz.

Servis bulunduğunda, arayan aygıt gönderilen servis bilgileri içine şu özellikler eklenir: ServiceRecordHandle, ServiceClassIDList, ServiceRecordState, ServiceID, ve ProtocolDescriptorList.

Bluetooth spesifikasyonlarında aygıtlar iki ayrı modda başka aygıtlar tarafından bulunabilir. Bunlar General Inquiry Access Code (GIAC) ve Limited Inquiry Access Code (LIAC) 'dir. Eğer bir aygıt GIAC modunda ise bu aygıtın her zaman başka aygıtlar tarafından bulunmak isteğini belirtir. Sürekli ve genel kullanıma açık servisler bu kategori içinde girer. Yazıcılar ve fax cihazları bu servislere örnektir. LIAC modundaki aygıtlar ise kullanıcı istediği anda kısa ve blirli bir süre bulunabilir

modda kalırlar. İhtiyaçveya istek anında bağlantı kuran aygıtlar bu kategori içinde yer alır. Çok konsollu oyunlar, mobil modemler bu tür servislere örnektir.

2.8 JSR 82

Java uygulamalarını kolayca Bluetooth kullanabilir hale getirmek için ortaya çıkan JSR 82 standardı, Bluetooth'un temel fonksiyonlarını içerir. Bu standart ile sağlanan özellikler şunlardır:

- 1.L2CAP bağlantıları oluşturmak ve veri transfer yapabilmek
- 2.RFCOMM bağlantıları oluşturmak ve veri transferi yapmak
- 3.Service Discovery protocol
- 4.OBEX (Seçimlik)

Bluetooth için, aygıt ve servis iki önemli kavramdır. Aygıtlar bluetooth cihazını, servisler ise cihazlar üzerindeki sunucu bağlantılarını tanımlar. Bluetooth aygıtları ve servislerinin ise sahip oldukları özellikler vardır. Bu özelliklerin bir kısmı sadece okunabilirken, bazıları değiştirebilir. Servis özelliklerinin bir kısmı bağlantı URL'i içinde verilebilirken, bir kısmı servis kaydı üzerinden de değiştirilebilir. Bağlantı URL'i verilebilen özellikler şunlardır:

1. Service ID
2. Service Name
3. Authenticate, Authorize, ve Encrypt

Service ID, her servise tekil bir numara vermek için kullanılır. Service Name ile servise açıklayıcı bir isim verilebilir. Encrypt seçeneği ile bağlantıda şifreleme kullanılıp kullanılmayacağını belirler. Authenticate seçeneği açıldığında, bağlantı kurulan aygıtın kimliğinin doğrulanması gerekir. Bluetooth cihazları authentication işlemini pairing adı verilen bir süreçle gerçekleştirir. Pairing işlemi için, iki cihaz sıra ile kullanıcıya pairing şifresini sorarlar. Sürecin başarılı olması için, karşılıklı girilen şifrelerin aynı olması gerekir. Pair edilen cihazlar, tekrar pairing gerekmemesi

için kaydedilirler. Authorize seçeneği açıldığında ise, her bağlantıda kullanıcıdan bağlantı için onay istenir.

Bluetooth üzerindeki her servis, açılmış ve istek bekleyen bir sunucu bağlantısına karşılık gelir. Servis özellikleri ise sunucu bağlantısı açılırken değiştirilebilir. Bağlantı açılırken verilen servis özellikleri, servis arayanların istek bekleyen sunucu bağlantısını tanımasına yarar ve gerektiğinde yapılacak bağlantının bilgilerini içerir.

JSR-82 Discovery API sayesinde etraftaki bluetooth aygıtlarını bulmak ve bu aygıtların sahip oldukları servisleri sorgulamak mümkün. Sorgulama sonucunda yukarıda belirtilen aygıt ve servis bilgileri elde edilir. Böyle bir servis olduğunu sorgulama sonucunda elde eden istemciler, gerek gördükleri durumda sunucu ile iletişime geçiyorlar.

Gelen bağlantı isteklerini kabul eder hale gelmek için:

- 1.Servisi tanımlayan bir servis kaydı hazırlanmalı
- 2.Hazırlanan servis kaydı, arayan cihazların bulabilmesi için kaydedilmeli
- 3.Sunucu, gelecek bağlantılar için beklemeye başlamalıdır.

İstemci ise, etrafındaki sunucuları ve üzerlerindeki servisleri sorgulayarak bulmalıdır.

2.9 Bluetooth Scatternet Ve Adresleme

Bluetooth içindeki en küçük ağ birim Piconet adını alıyor. Bluetooth v1.2 spesifikasyonlarında bir piconet içinde nasıl haberleşme yapılacağı belirtiliyor. Bu belirtme göre, bir piconet içinde bir aygıt yönetici oluyor ve üyeler bu yöneticinin saatine senkronize oluyor. Bu yüzden yönetici birden fazla piconete katılamamasına rağmen üyeler birden fazla piconete üye olabilir. Birden fazla piconete üye olan üye aygıtlar ise Piconetleri birleştirerek Scatternet oluşturabiliyorlar. Ancak spesifikasyonun getirdiği zorunluluk burada bitiyor. Çünkü iki piconet arasındaki

üyenin adresleme yapmak gibi bir sorumluluğu yok. Bluetooth spesifikasyonu bir aygıtın birden fazla rol üstlenebileceğini belirtiyor. Yani aygıt bir piconet içinde yönetici iken, bağlı olduğu başka bir veya birden fazla piconet içinde üye olabiliyor. Buna ek olarak bir aygıt birden fazla piconet içinde sadece üye durumunda da olabiliyor. Birden fazla piconet ile iletişimin halinde olan aygıtlar, aracı olarak kullanılabilir. Böylece çok sekmeli tasarımsız bir ağ yani Scatternet oluşmasını sağlayabiliyorlar.

Scatternet, yukarda anlatılan kullanım biçimine ek olarak, piconet oluşturmada halihazırda kullanılan bir basamaktır. Şöyleki, varolan bir piconete katılma süreci iki farklı biçimde gerçekleşebilir. Bunlardan ilki üyenin piconete kendisi girmek istediğinde gerçekleşir. Bu işlem sırasında, Piconete katılacak aygıt, önce varolan Piconet yöneticisi ile kendi arasında, kendisinin yönetici olduğu geçici bir Piconet oluşturur. Piconetin oluşturulmasının hemen ardından ise, üye olmak isteyen aygıt, varolan Piconet yöneticisinden Yönetici-Üye değişimi için gereken bilgileri alır ve varolan Piconete üye olarak dâhil olur. Yönetici-Üye değişimi için alınan bilgiler arasında iletişim için gereken parametreler üzerinde anlaşılması, frekanslar arasında sıçrama sırası ve zamanlama vardır. İkincisinde ise, aynı prosedür bu kez Piconet yöneticisi ve üye arasında roller değiştirilerek gerçekleşir. Böylece oluşturulan Piconetin tekrar bozulmasına gerek kalmaz. Yönetici, paging ve inquiry yaparken, varolan Piconet üyelerini Hold veya Park durumuna alır.

Görüldüğü üzere, Scatternet çok temel bir işlem olan Piconet oluşturmada zaten işletilen bir bağlantı türü. Ancak Adresleme yapmak için Scatternet oluşturmak ve arkasından işler durumda tutmak, kaynakları zorlayan bir kullanım türüdür [Morrow , 2002, sayfa 237]. Aradaki üyenin kompleks TDM yapması gerekmektedir. Aradaki üyeler bunu yapsalar bile, paket trafiğinde, kolaylıkla trafiğin sıkıştığı nokta haline gelebilirler. Bu durumda, oluşturulan scatternetleri kısa süreli tutmak ve sadece zorunlu durumlarda oluşturmak yerinde olur [Morrow , 2002, sayfa 237]. Scatternet oluşturmaya iki alternatif te şunlardır:

1. Bir başka piconet ile haberleşmeden önce, varolan piconet bağlantılarını kesmek
2. Bir cihaz üzerine birden fazla bluetooth aygıtı koymak

Birinci seçenek, kolay olmasına karşın, her piconeti oluşturmak için geçen zaman bir kaç saniyeyi bulacaktır. Eğer piconet değiştiren aygıt, her iki piconette de üye ise, yukarıda bahsettiğimiz gibi üye ile yönetici arasında önce gecici bir piconet kurulacak, ardından Yönetici-Üye değişimi işlemi yapılacak ve kullanılacak piconete katılım olacaktır. Bu işlem ise yöneticinin 1.28 saniyede bir Page Scan yaptığı düşünüldüğünde bir kaç saniye alacaktır. Bu hesaplamaya göre, kurulan piconet ancak uzun süreli kullanılacak ise bu işlem uygulanabilir oluyor.

3. KONU İLE İLGİLİ YAYINLAR VE ÇALIŞMALAR

Önceden belirttiğimiz gibi kablosuz ağları üç temel gruba ayırmak mümkün. Bunlar:

1. Hücresel Ağlar
2. Tasarımsız Ağlar
3. Sensör Ağlar

Kablosuz ağlar üzerinde adresleme yapmak için kullanılan algoritmalar sahip oldukları özellikler ile birçok farklı biçimde sınıflandırılabilir. Adresleme bilgisinin hesaplanma yöntemine göre reactive, proactive ve hybrid sınıflarına ayrılabilirler. Reactive protokoller, hedefe ulaşmak için gereken yolu gerektiği zaman hesaplarlar. Proactive protokoller ise, yolları daha ihtiyaç olmadan hesaplayıp saklarlar. Hybrid protokoller ise, bu iki yöntemi birlikte kullanırlar. Haberleşen aygıtların yerleri sabit olduğunda, adresleme bilgisini tablolarda saklamak ve proactive yöntemler kullanmak daha avantajlı olurken, noktaların yerleri değişken olduğunda bundan bahsetmek mümkün değildir. Ayrıca ağ içindeki noktaların görevlerine göre de bir sınıflandırma yapmak da mümkündür. Haberleşen noktaların görevleri arasında farklılıklar var ise bu hiyerarşik, değil ise düz protokol adını verebiliriz. Bütün bunlara ek olarak, yol bilgisini bulmada noktaların konumlarını ve zamanı kullanan yöntemler de var.

3.1 Tasarımsız Ağlar ve Adresleme

Tasarımsız ve sensör ağların gereksinimleri farklı olduğu için, kullanılan adresleme algoritmaları da farklıdır. Başlıca tasarımsız ağ adresleme algoritmaları şunlardır:

1. AODV (Ad hoc On Demand Distance Vector) wireless [Perkins et al, 2001]

2. DSDV (Destination-sequenced distance vector) [Johnson and Maltz, 1999]
3. DSR (The Dynamic source routing) [Petrioli et al, 2004]
4. TORA (Temporally-ordered Routing algorithm) [Park and Corson, 2001]

AODV adresleme protokolü dahilinde, her uç kendi isteği ile ağı terk edebilir veya ağı girebilir. Hiçbir merkezi otoriteye gerek yoktur. Herhangi bir uç henüz nasıl ulaşacağını bilmediği bir uca ulaşmak istediğinde RREQ adı verilen uca olan yolu öğrenmek istediğini gösteren isteği tüm komşularına yayar. Bu isteği alan komşular, eğer hedefe olan yolu biliyorsa, RREP adı verilen ve hedefe yol bilgisini içeren yanıtı geri dönerler. Eğer bilmiyorlarsa, aynı şekilde RREQ isteğini tüm komşularına yayarlar. Bunun sonucunda, RREQ isteği yolu bilen bir uca veya hedefe ulaşınca kadar dolaşır. RREQ isteğini ağı yayan yani hedefe olan yolu bulmak isteyen uç cevabı aldığı anda, cevabı ona göndereni ve hedef adresi tablosuna ekler. Yani klasik adresleme tablolarında olduğu gibi, bir hedef için bir sonraki uç adresi tutar. Ayrıca, tablodaki her satır için, bu satırı kullananların listesi vardır. Hedefe ulaşmada bir sorun yaşandığında, RERR adı verilen hata paketleri, bu satırı kullananlara gönderilir. Böylece RERR paketini alanlar kendi tablolarını güncelleyebilirler. Buna ek olarak, belirli bir süre kullanılmayan satırlar geçerliliğini yitirirler.

DSR, AODV gibi hedefe olan yolu ihtiyaç anında bulmasına rağmen tuttuğu adresleme tablosu farklıdır. Her hedef için, tabloda o hedefe ulaşan tüm yollar, yol üzerindeki tüm noktalar ile birlikte tutulur. Eğer hedefe ulaşmada bir sorun var ise, geriye RERR cevabını alan nokta tablosunu günceller ve başka bir yoldan hedefe ulaşmaya çalışır. Eğer ulaşacak başka yol yok ise, tekrar RREQ isteğini çevresine yayar.

Servis Arama/Konumlandırma

Tasarımsız adresleme için kullanılan metotlar, ufak eklentiler ile servis aramak için kullanılır hale getirilebilir. Tek yapılması gereken, aranan noktanın adresi değil de istenen servisi açıklayan bilgileri içeren REQ paketlerini kullanmaktır. [Perkins and Koodli, 2002] içinde yapılan çalışma ise buna örnektir. Bu çalışmada, servisi veren birini bulmak için, ağı servisi tanımlayan istek paketleri yayılır. İstenen servise giden yolu bilenler ise cevap verirler. Service Request Packet (SREQ) adı verilen bu paket, RREQ isteğini ve bu isteğe uzantı olan Service Request

Extention (SRE) içerir. SRE, servisi tanımlayan bilgiler içerir. Benzer şekilde, SREP (Service Requeset Reply) paketi ise Service Reply Extention (SRE) adı verilen ve bulunan servis ile ilgili ek bilgileri içeren bir bölümü barındırır. SRE oluşturulduğunda, paket içindeki hedef adresi boş bırakılır. Servis isteği, ağ içinde birine ulaştığında, eğer ulaşılan nokta servisi vereni biliyorsa veya servisi kendi veriyor ise, paket içindeki boş olan hedef adresini doldurur. Servisi verenin adresi ile servis tipinin eşleştirilmesine Service Binding adı verilir. SREP içindeki hedef bilgisi, Service Binding bilgisi olarak paketin geçtiği noktalar tarafından tablolarda tutulur. Bu durumda bir servis tipi için birden fazla servis veren nokta olabilir. Tablolarda ayrıca her servis satırı için bir geçerlilik zamanı vardır. Bu metot içinde Servisi verene giden yolu bilmek ve Servis vereni bilmek ayrı bilgiler olarak kabul ediliyor. Servis vereni bilmek, ona giden yola bilmeyi gerektirmez. Dolayısıyla:

- a) Eğer bir uca gelen SREQ paketi hedef adres içermiyor ise:
 - i) Eğer uç servis için Service Binding'e sahip değil ise, isteği etrafına yayar.
 - ii) Eğer uç servis için Service Binding ve yol bilgisine sahip ise, SREP hazırlar ve isteği yaratana doğru geri gönderir. Eğer Service Binding var ama yol yok ise, hedef adresi doldurur, dolaşılacak sekme sayısını maksimuma alır ve SREQ isteğini etrafına yayar.
- b) Eğer uca gelen SREQ paketi, hedef adres içeriyor ise:
 - i. Eğer uç, hedefe giden yolu biliyorsa, SREP hazırlayıp isteği yaratana doğru gönderir.
 - ii. Eğer uç hedefe giden yolu bilmiyorsa alternatif servis sağlayıcısı bulmaya çalışır. Eğer var ise SREP gönderir. Aksi durumda SREQ isteğini etrafına yayar.

3.2 Sensör Ağlar ve Adresleme

3.2.1 Flooding

Sensör ağlarda bilgi taşımak için en basit yöntem, noktaların sahip olduğu bilgiyi çüm çevre noktalara yayın yapmasıdır. Yayını alan noktalar da aynı şekilde veriyi kendi çevrelerine yayarlar. Bu yöntemin başlıca sorunları ise şunlardır:

Implosion: Herhangi bir noktaya aynı datanın kopyaları farklı yollardan defalarca ulaşabilir.

Overlap: Birbiriri ile iletişimde olan iki nokta, ortak komşularına aynı verinin iki kopyasını gönderirler.

Resource blindness: Tüm bu işlemleri yapabilmek ise, enerji harcarlar.

3.2.2 Gossiping

Flooding methodunun geliştirilmiş bir versiyonu ise Gossiping'dir. Bu yöntem ile, veriyi yayacak olan nokta, veriyi tüm çevre noktalara göndermek yerine rastgele seçtiği bir çevre noktaya gönderir.

3.2.3 SPIN

Enerji ve hat kapasitesini kullanmaktaki verimliliği arttırmak için yine aynı temeller üzerine çalışan SPIN geliştirilmiştir [Heinzelman et al, 1999]. SPIN protokol dahilindeki kaynak noktalar gönderecek veriye sahip olduklarında, veriyi hemen ağa dağıtmazlar. Öncelikleri veriyi tanımlayan bir meta-data ağa gönderirler. Meta-data ile veriye ihtiyacı olduğuna karar veren karşı noktalar ise ağa veriyi istediklerini gösteren bir cevap gönderirler. Cevabı alan kaynak nokta ise tüm çevre noktalara gerçek veriyi iletir. Bu adımlar sırası ile ADV(Advertisement), REQ(Request), DATA adlarını alırlar. Çevre noktalar da aynı işlemi yapar ve tüm ağ verinin kopyasına sahip olur. Böylece isteyen olmadığında gerçek veriyi tüm ağ içinde taşımak için boşuna bant genişliği ve enerji harcanmaz.

3.2.4 LEACH

Verimli veri dağıtımını sağlayabilmek için geliştirilen bir başka yöntem ise LEACH (Low Energy Adaptive Clustering Hierarchy) protokolüdür [Heinzelman et al, 2000]. LEACH hiyerarşik bir protokol olup bazı ağ birimlerini küme lideri olarak seçer. Enerji kullanımı ağı eşit dağıtmak için ise bu rol zaman dilimlerinde değiştirilir. Küme lideri olan birim, küme içinden gelen bilgiyi derler, sıkıştırır ve istasyona gönderir. Bu gönderim periyodik aralıklarla tekrarlanır. Bu yüzden de LEACH periyodik aralıklarla veri toplamada kullanılması uygun bir algoritmadır. LEACH iki aşamadan oluşur. İlk aşama kurulum, ikincisi ise işletim aşamadır. İlk aşamada, her birim 0 ve 1 arasında rasgele bir sayı belirler. Bu sayı belirli bir eşiğin altında ise, birim lider olduğunu etrafına yayar. Yayını alan birimler, signal gücüne göre hangi birimin kendisinin lideri olması gerektiğine karar verir. Veri gönderimi için TDMA kullanılır. Küme içine katılacak birimler belirlendikten sonra, lider TDMA planını hazırlar ve her bir birime veri göndermesi için bir zaman dilimi verir. Bundan sonra işletim aşaması başlar. Belirli bir süre sonra ise, yeni liderleri belirlemek için ağ, tekrar kurulum aşamasına döner.

3.2.5 Directed Diffusion

Sensör ağlar için bir başka adresleme protokolü ise Directed Diffusion'dır [Intanagonwiwat et al, 2000]. Bu protokolde veriye ihtiyaç duyan her nokta, istenen veriyi tanımlayan bir sorgu hazırlayarak bu sorguyu ağı gönderir. Bu sorgulama istenen özellik ve özelliğin değerini içeren bir listedir. Sorgulamayı alan noktalar bu sorgulamayı kendi üzerinlerinde zula yapabilirler. Zula içindeki her bir sorgu için ayrıca sorgunun geldiği noktaların listesi bulunur. Listede gönderenin yanında isteğin geçerlilik zamanı, dolaştığı sekme sayısı gibi bilgiler de yer alabilir. Noktalar üzerinde varolan veya toplanan veri daha sonra zula ile karşılaştırılır. Sahip olunulan veri ile herhangi bir sorgu arasında eşleşme olduğunda ise, sorguyu gönderenlerinden listesinden biri seçilir ve veri ona gönderilir. Seçilen noktaya ulaşılamaması halinde ise diğer seçenekler denenebilir.

3.2.6 Rumor Adresleme

Rumor Adresleme [Braginsky and Estrin, 2002] ise Directed diffusion'a benzeyen ama Directed Diffusion'daki sorguları dağıtmak yerine oluşan olayları ağ içine dağıtmayı esas alan bir protokoldür. Eğer ağ içinde oluşan olay sayısı az ise, olayları ağa dağıtmanın, sorguları dağıtmaktan daha hesaplı ve az maliyetli olacağı esasına dayanır. Bu protokol, olayları ağ üzerinde dağıtabilmek için temsilci adı verilen olay bilgisi içeren paketleri kullanır. Herhangi bir nokta ağ içinde oluşan bir olay algıladığında, olayı kendi üzerindeki olay tablosuna kaydeder ve olay için bir temsilci oluşturur. Oluşturulan temsilci ise ağı dolaşarak olayı diğer noktalara iletir. Bu şekilde, herhangi bir sorunun cevabını almak için tüm ağı meşgul edecek bir sorgu dağıtımı yapılmaz. Bunun yerine sadece olay tablosuna bakmak yeterli olacaktır. Ayrıca olay oluşturulan tablolardan çıkarılacak yol ile, olayın kaynağına ulaşmak mümkün olacaktır. Directed Diffusion'dan ayrılan bir başka konu ise, bulunan bu yolun sadece bir tane olmasıdır.

3.2.7 Minimum Cost Forwarding

Eğer ağ içinde soruları yapan istasyonlar var ise, Minimum Cost Forwarding Algorithm [Chen et al, 2001] bu istasyonların veri toplamasını efektif hale getirmek üzere ortaya çıkmış bir algoritmadır. Her noktanın ayrı ayrı adresleme tablosu tutması yerine, noktalar gönderilen paket için, kendilerinin istenen istasyona en yakın yol üzerinde olup olmadıklarını bilirler. Eğer en yakın yol üzerinde iseler, gelen mesajı çevrelerine dağıtırlar. Aksi durumda hiç birşey yapmazlar. Her noktanın istasyona uzaklığını hesaplayabilmesi için ise şu işlem adımlar gerçekleştirilir. Ağ içindeki her nokta, istasyona uzaklığını gösteren değeri sonsuz yapar. Aynı değeri, istasyonun kendisi ise sıfır yapar. Ardından istasyon içindeki uzaklığın bulunduğu paketi ağa gönderir. Paketi alan noktalar eğer gelen değer ve geçilen son yolun değeri, kendi üzerlerinde olan değerden küçük ise, kendi üzerilerindeki değeri son yolun değeri ve gelen değer toplamına güncelleştirirler ve paketi tekrar dağıtırlar. Aksi durumda birşey yapmazlar. Bu şekilde, ağ üzerindeki her nokta istasyona uzaklığı gösteren değerlere sahip olmuş olur.

3.2.8 COUGAR

Bunlara ek olarak, sensör ağ üzerinde sorgulama yapmak için, ağı dağıtık bir veritabanı gibi gören algoritmalar da vardır. Bunlardan birisi olan COUGAR [Yao and Gehrke, 2002], istasyonlarının bir sorgu planı hazırlamasını gerektirir. Hazırlanan sorgu içinde, istenen veri, izlenmesi gereken yol ve noktaların veriyi derleme ile istasyona göndermek için kullanacakları lider noktaları nasıl seçecekleri yer alır.

3.3 Scatternet Formasyon Algoritmaları

3.3.1 A New Bluetooth Scatternet Formation Protocol

Bluetooth üzerinde Scatternet oluşturmak üzerine yapılmış çalışmalar mevcuttur. “A New Bluetooth Scatternet Formation Protocol” [Law et al, 2003], scatternet formasyonu ile ilgili yapılmış çalışmalardan biridir. Algoritmada, bir piconet, bir nokta veya bir scatternet komponent olabilmektedir. Her komponentin bir lideri vardır. Tek noktaların kendisi, piconetlerin yöneticisi, scatternet içinde ise piconet yöneticilerinden biri lider seçilmektedir. Başlangıçta her nokta, lider varsayılmakta ve her lider önceden belirlenen bir sürede biteceği varsayılan bir rutini çalıştırmaktadır. Bu rutin içinde ise, random seçilen olasılık değeri ile lider Seek veya Scan yapmaktadır. Seek yeni üyeler bulmak için, Scan ise üye olmak için kullanılan rutinlerdir. Seek içinde Inquiry ve Page yapılırken, Scan içinde Inquiry Scan ve Page Scan yapılır. Bu işlemler sonucunda oluşan match'lere göre Connected moduna geçiş yapılmaktadır. Bu modda ise, birbirini bulan noktaların o anki durumlarına göre, ağlar arasında farklı birleşmeler gerçekleşir.

3.3.2 BlueMesh

Bir başka scatternet oluşturma algoritması ise BlueMesh'dir [Petrioli et al, 2004]. Bu algoritma, diğer algoritmalarındaki gibi bir ağaç oluşturmak yerine, bazı aygıtların yönetici olmasına ve her yöneticinin en fazla 7 tane üye seçmesi esasına dayanır. Bir aygıtın yönetici olup olmaması ise, aygıt üzerinde hesaplanan bir ağırlığa göre belirlenir. Bu ağırlığın hesaplanmasında kullanılan faktörler ise, aygıt

yönetici olmaya uygun kılan batıri durumu gibi aygıt kaynaklarıdır. Yönetici seçildikten sonra, her yönetici çevresindeki en fazla 7 aygıtı üye olarak seçer. Bu seçimlerden ilk 5 tanesi, ağıın kalan kısmına ulaşabilecek şekilde yapılır. Bu seçim, bir noktanın kapsama alanında 5'den fazla nokta var ise, ilk 5 tanesinden sonrakilerin muhakkak birbirlerinin kapsama alanında olması esasına dayanır. Yani bir noktanın etrafına, birbirinin kapsama alanında olmayan 6 nokta yerleştirilemez, çünkü bazı noktalar aralarındaki mesafe, kapsama alanın yarıçapından az olur. Yönetici ve üye aygıtlar belirlendikten sonra, piconetler oluşturulur ve piconetler arasında geçisi sağlayacak araçlar seçilir. Bu işlemlerin gerçekleşmesi için ise, her aygıtın iki sekme uzaklığı kadar tüm aygıtları ve ağırlıklarını bilmesi gerekir. Bu bilgilerin toplandığı aşamaya ise, topology discovery adı verilir. Bu işlemin ilk aşamasında, tüm aygıtlar bir sekme uzaktaki yagıları bulur. İkinci aşamada ise, her aygıt bilgilerini diğer aygıtlar ile paylaşır. Böylece, her aygıt kendisinden iki sekme uzaklıktaki aygıtlara kadar olan bilgileri toplamış olur. Algoritmanın çalışması için aygıtların birbirlerini simetrik olarak bilmesi gerekir. Ancak Bluetooth ile çevre aygıtları aradığınızda sadece aramayı yapan aygıt çevre aygıtlardan haberdar olur. Arama sonucu bulunanlar ise, henüz bu bilgiye sahip olmaz. Bizim çalışmamızda da karşılaştığımız bu durum için önerilen en yararlı yöntem, aygıtların belirli sürelerde rol değiştirmesidir [Salonidis et al, 2001]. Bu yöntem dahilinde, her aygıt belirlenen maximum süre boyunca ardı ardına inquiry ve scan yapmaktadır. Süre bitimine kadar, her döngü başında ilk işlemin ne olacağı ise rastgele belirlenmektedir (1/2 olasılıkla). Simetrik bilgi değişimini sağlamak için ise, aygıtlardan biri inquiry sonucunda bir aygıt bulmuş ise ise o aygıt ile geçici bir piconet oluşturarak bilgileri değişmektedirler.

3.3.3 BlueStar

Scatternet oluşturmanın yanında oluşan Bluetooth cihazlarını daha büyük ağlara bağlamak için yapılan çalışmalar da vardır. BlueStar [Cordeiro et al, 2004], Bluetooth kablosuz kişisel ağları ile IEEE 802.11 kablosuz yerel ağları birbirine bağlamayı amaçlayan bir çalışmadır. Böylece Bluetooth cihazlar, yerel ağları kullanarak daha büyük ağlara örneğin Internet'e bağlanabilecektir. Bunu sağlamak için ise, daha büyük ağlara bağlantısı olan Bluetooth cihazları, aracı olarak kullanılacaklardır. IEEE 802.11 ve Bluetooth aynı frekans bandını kullandığı için,

bu çalışma IEEE 802.11'de varolan kanal kullanımını sezebilme özelliğinin Bluetooth'a eklenmesini önermektedir.

3.3.4 Sensör Bilgisi Toplama

Bluetooth altyapısını sensör bilgisini taşımak için yapılan çalışmalar da mevcuttur. Bunlardan biri olan “A new Bluetooth scatternet formation protocol” [Mehta and Zarki, 2004] adlı çalışma kapsamında, sensör bilgisi taşıma iki aşamada gerçekleşmektedir. Bu aşamalar, bilgi toplama aşaması ve bağlantı kurulumlarının yapılmasıdır. Sonuçta oluşan ağın en tepesinde Super Master (SM) adı verilen aygıt yer alır. Bu aygıtın görevi hem ağ orkestrasyonunu yapmak hem de toplanan tüm veriyi Internet'e aktarmaktır. İlk aşama süresince Super Master'dan başlayarak sıra ile her nokta inquiry yapar. Bir anda yalnızca bir aygıtın inquiry yapmasına izin verilir. Sırası gelen aygıt inquiry yaparken, diğer tüm aygıtlar inquiry scan yapmak durumundadır. Inquiry sırasında elde edilen veri, matrix oluşturması için üstündeki aygıta iletilir. Inquiry yapan aygıt, bulduğu aygıt ile SDAP yardımı ile ağ için gerekli parameterleri değiştirir. Inquiry yapan aygıtlar, daha sonra inquiry scan yapmazlar. Böylece aygıt matrixin simetri özelliğinden faydalanılarak verim arttırılmış olur. Inquiry aşamasında belli bir süre cevap alamayan aygıt sırayı kendi bulduğu aygıtlardan sırada olana verir. Eğer bulduğu aygıt yok ise veya bulduğu aygıtların hepsi Inquiry'lerinin bittiğine dair cevap dönmüş ise, bu aygıt da kendisine sırayı veren üst aygıta dönüş yapar. Dönüş sırasında topladığı verileri de üstüne iletir. Bu backtracking mekanizması ile tüm ağaç dolaşmış olur. Inquiry aşaması bittiğinde, Super Master'da bilgiler tüm ağ için toplanmış olur.

Bilgi toplama aşaması bittikten sonra, Super Master kendisi ile bir sekme uzaklıkta olan aygıtlar arasında k tanesini kendi üyesi olarak seçer. Bu seçim toplanan performans parametrelerine bağlıdır. K adet üye seçildikten sonra, bu işlem her bir üye için tekrar edilir. Super Master altındaki her üye, kendileri için seçilen üyelerin yöneticileri olurlar. Bu işlem tüm ağaç için devam ettirilir ve işlemi Super Master koordine eder.

3.3.5 Bluetrees

Bluetrees adlı algoritmada [Záruba et al, 2001] Scatternet, bir veya daha çok köklü, birbiri ile bağlantı kurmaya çalışan ağaçlardan oluşur. Toplu halde veya tek tek varolan ağaçlar ile yine toplu halde veya tek tek ayrılan ağaçlar döngü oluşturmadan ağın parçalarını birleştirmeye çalışırlar. Ayrıca algoritmada haberleşme yapan ve ağaçları birleştirmeye çalışan birimlerin görevleri rastgele seçilerek dengelenmeye çalışılır. Ayrıca iki birim arasında haberleşme yolu uzaklığını minimum tutmaya çalışır.

Herhangi bir zamanda Scatternet, birbirleri ile bağlantı kurmaya çalışan ağaçlardan oluşan orman görünümündedir. Bu ağaçlardan bazıları bir ağaca bağlanmayı amaçlayan tek bir birim içerir. Her ağaç içindeki bir birim diğer ağaçlar ile bağlantı kurabilmek için bir süre harcar. Scatternet oluşturmak için yapılan birleşmeler iki ağaç arasındaki birleşme ve bir ağaç ile tek bir birim arasındaki birleşme olarak ikiye ayrılır. Birinci birleşme bir döngü oluşturabilirken ikinci birleşmenin döngü oluşturma riski yoktur. Her ağaç içindeki bir birim diğer ağaçlar ile bağlantıyı sağlamak için koordinatör olarak seçilir. Ağaç içindeki diğer tüm birimler ise sadece çevreden gelebilecek Inquiry istekleri için kısa kısa zaman harcarlar. Birimlerin içinde bulunabilecekleri üç durum vardır. Bunlar Inquire, Scan ve Comm durumlarıdır. Her birim bu üç durum içinden iki tanesi arasında dolaşır. Inquire durumundaki bir birim Bluetooth inquiry işlemini yapar. Comm durumunda birim ya birşey yapmıyordur ya da haberleşiyordur. Scan durumundaki bir nokta ise Inquiry scan yapıyordur. Ağaçlar arasındaki döngüleri engellemek için ise sadece köklerin başka bir ağaç ile birleşmesine izin verilir. Kökler birleşme işini yaparken yapraklardan bir veya bir kaç Inquire işlemini yapar. Böylece tüm işlemi köklere yükleyerek kaynaklarını zorlamaları engellenmiş olur.

Her kök, diğer ağaçları araması için bir tane çocuğunu koordinatör yapar. Eğer sadece bir tane çocuğu var ise koordinatör kendisi olur. Eğer seçilen çocuk koordinatör olmak istemiyor ise kendi çocuklarından birini seçer. Bu işlem ağacın yapraklarına inene kadar veya bir koordinatör seçilene kadar devam eder. Daha önce de açıklandığı gibi koordinatörler Inquire ve Scan modlarında diğer koordinatörleri

ararlar. İki tane koordinatör birbirlerini bulduklarında içlerinde bulundukları ağaçların köklerini birbirlerini Page ve Page Scan edebilmeleri için gerekli bilgi ile uyarırlar ve aradan çekilirler. Diğer ağaç kökünü Page veya Page Scan etmek için bilgileri alan iki kök, Page ve Page Scan işlemlerini yaparak iletişime geçerler. Yönetici olan birim yeni ağacın kökü olur. Ağaç kökü bu arada yeni bir koordinatör seçer. Koordinatörler ansızın ağaçtan kopabilecekleri için ve çok fazla kaynak harcamamaları için ayrıca ağaç içindeki koordinatörler belli aralıklar değiştirilir.

Ayrıca ağaçlar herhangi bir zamanda herhangi bir kısımdan kopabileceğinden kendi kendini iyileştirme adımı uygulanmaktadır. Bu adımda, eğer kopan bir çocuk ise, yapılması gereken tek şey kopan çocuğun koordinatör olması durumunda tek başına kalıp kalmadığını kontrol etmesi ve ona göre tipini güncellemesidir. Eğer kopan ebeveyn ise bu defa yapılması gereken iç birimlerden birini kök olarak seçmektir.

4. PROTOKOL

Scatternet formasyonunda kurulum performansı, hızını ve etkinliğini etkileyen faktörleri dört başlık altında toplayabiliriz. Bunlar:

1. Birimlerin birbirini nasıl bulduğu
2. Oluşan grafiğin kopuk bir grafik olmamasının nasıl sağlanacağı
3. Yönetici ve aracı olan birimlerin nasıl seçildiği
4. Formasyonun merkezi olup olmadığıdır.

Birimlerin birbirini nasıl bulduğu scatternet formasyon performansını doğrudan etkiler. Bir birim bir anda sadece Inquiry ya da Inquiry Scan yapabilir. Bu durumda eğer Inquiry ve Inquiry Scan yapan birimlerin sayısını dengeleyemez isek, ya birimler birbirini hiç bulamazlar ya da bulmak için çok uzun zaman ve kaynak harcayabilirler. Bluetooth aygıt keşfinin uzun süre zaman alması da göz önüne alınırsa işlemin eğer düzgün yapılmaz ise iyi bir performans vermeyeceği açıktır. Bir takım algoritmalar, bu sorunu birimlerin rastgele belirledikleri aralıklarda yine rastgele olarak Inquiry ve Inquiry Scan yapması ise çözer [Law et al, 2003]. Bu durumda her birim bir sonraki adımda ne yapacağını rastgele bir sayı ile belirler. Rastgele seçimler ise Inquiry ve Inquiry Scan işlemleri arasında bir denge sağlar.

Birleştirilen piconetlerin büyüklüğü, scatternet performansını etkileyen bir diğer faktördür [Záruba et al, 2001]. Oluşan piconetlerin büyüklüğünün çok küçük olması durumunda, piconetleri bağlamak için kullanılan gayeway sayısı artacaktır. Bu durumda da araçların piconetleri bağlamak için kullanacakları kaynağın artması gerekecektir. Piconetlerin çok büyük olması durumunda ise, yönetici olan birimler üzerine çok fazla yük binecektir çünkü bildiğimiz gibi tüm piconet trafiği yönetici birimler üzerinden akar.

Oluşan piconet ağı gruplarının fiziksel engeller olmadığı sürece birbirine bağlı olmasını sağlamak da bir başka uğraşı konusudur. Grupları bağlı tutarken ise gruplar arasındaki araçları da optimumda tutmak gerekir. Grupları bağlı tutmam

maksadıyla, bir piconete başka bir piconetten birden fazla bağlantı yapılması da araçlar üzerindeki yükü arttıracaktır [Záruba et al, 2001]. Yani hem gruplar bağlı olmalı hem de aracı seçimleri ağ üzerindeki bağlantı yükünü minimuma indirecek şekilde yapılmalıdır.

Piconet içinde yönetici ve üye birimlerin seçilmesi de bir başka tartışma konusudur. BlueMesh [Petrioli et al, 2004] adlı algoritmada, birimlerin sahip oldukları kaynaklar, birimlerin rolünü belirlemede kullanır. Her birim sahip oldukları kaynaklara göre bir sayı hesaplar. Bu sayı, ağ oluşumu öncesinde çevre birimler paylaşılır. Karşılaştırılan değerlere göre, yönetici ve aracı olacak birimler yüksek değerlere ihtiyaç duydukları için, yüksek değere sahip olan birimler yönetici ve aracı olurlar.

Scatternet oluşturmanın orkestrasyonunun merkezi olup olmaması da algoritmaların başlıca ayrımlarındandır. Daha önce de belirtildiği gibi, Mehta ve Zarki'nin çalışmasında [Mehta And Zarki, 2004], Super Mode tüm scatternet oluşumunu orkestra eder ve sensörler tarafından algılanan değerleri toplar. Bluetrees [Tan et al, 2002] adlı çalışmada ise koordinatörler çevreyi ararken tüm birleşme işlemlerini ağaç kökleri yaparlar.

4.1 Algoritma

Algoritmayı oluştururken tüm adımları gözönüne almak gerekir. Bizim geliştirdiğimiz algoritma, hızlı bir ağ kurulumu yanında sürekli olarak bağlantıları iyileştirmeyi ve bunu yaparken yapılan işlemleri olabildiğince az kesintiye uğratmayı amaçlamaktadır.

Algoritma iki adımdan oluşmaktadır. Birinci aşamada sadece 1 sekme uzaklıktaki birimler incelenerek optimum scatternet bağlantıları kurulur. İkinci aşamada ise birimler ağı kullandıkça bağlantılar üzerinde değişiklikler yapılır. Bu değişiklikler ağı çalışır durumda tutulması ve topolojiyi daha iyi duruma getirilmesi arasındaki denge gözönüne alınarak yapılır. Bir başka önemli konu ise, ikinci aşamada eğer bir birim haberleşme yapmıyor ise inquiry scan yapıyor demektir. Yani

ağ içindeki birimler, haberleşmedikleri sürece ağa yeni katılacak olanlar ile bağlantı kurmaya hazır bekliyor olacaktırlar.

Aşama 1: Ağa bağlanmak isteyen her birimin ilk başlayacağı adım Aşama 1'dir. Ağ kurulumunun ilk aşamasında rastgele olarak inquiry ve inquiry scan yapmak zaten başarısı kanıtlamış bir method olduğundan, burada da bu işlem yapılmaktadır. Yani ağa yeni katılmak isteyen birimler an az bir piconet ile bağlantıya geçene kadar inquiry ve inquiry scan modları arasında rastgele olarak dolaşırlar. Eğer bir birim inquiry yapıyor ise, o birim yönetici olmaya çalışıyor demektir. Eğer birim inquiry scan yapıyor ise, birim üye olmaya çalışıyordur. Ancak daha sonra belirteceğimiz sebeplerden yönetici/üye yapılması da olasıdır. Inquiry yapan birim birkaç nokta bulur ise, rastgele modlar arasında dolaşmayı durdurur. Aşama 1 bu birim için artık bitmiştir. Bu aşamadan sonra bu birim için sadece inquiry scan yapılabilir. Yönetici birimler hemen buldukları üye birimleri komşu listelerine kaydederler ama üye birimler henüz yönetici birimden haberdar değildir. Bahsedildiği üzere, Bluetooth aygıt arama işlemi aşamasındaki bilgi paylaşımı simetrik değildir. Üye birimler, kendilerine ilk mesaj gelene kadar yönetici hakkında bilgi sahibi değildirler. Bu aynı zamanda ilk mesaja kadar, üye birimlerin Aşama 1'den çıkmayacağı anlamına gelir.

Inquiry yapan birim Aşama 1'den birkaç birim bulup çıktıktan hemen sonra, tüm üye birimlere özel bir mesaj gönderir. Bu mesaj ile, tüm üye birimlerden varsa hali hazırda bağlı oldukları piconetler hakkında bilgileri göndermeleri istenir. Bu ilk mesaj, üye birimlerin yönetici birimi bilmeleri açısından da önemlidir. Mesajı alan her üye kendi Bluetooth adresini ve katıldığı her piconet için piconet'teki birim sayısını ve yöneticinin Bluetooth adresini göndererek cevap verir. Yönetici tüm bilgiyi topladıktan sonra aşağıdaki işlemlerden birini yapar.

- 1- Eğer üyelerden biri kendi piconeti içinde yönetici ise sahip olduğu üye sayısı 7 'den az ise, yönetici ve üye arasında Yönetici-Üye değişimi yapılır.
- 2- Eğer üye birimlerden birinin gönderdiği yönetici listesi başka bir üyenin gönderdiği yönetici listesinin alt kümesi ise, ilk üye, yönetici birimin komşu listesinden silinir ve piconetten ayrılır.

İlk adım oluşan piconetlerin büyüklüğünü dengelerken ikinci adım birden fazla üye birimin aynı piconete aracı olmasını engeller. Bir başka deyiş ile ikinci adım iki komşu piconet arasında çember oluşmasını engeller. İkinci adım birinci adımdan önce yapılmalıdır.

Aşama 2: Algoritmanın ikinci aşaması, Sensör ve Tasarımız ağılarda zaten varolan tecrübeden faydalanır. Ancak bu tecrübeleri kullanabilmek için yayın yapmaya ihtiyacımız var. Bluetooth spesifikasyonu piconetler arasında yayın yapmak için bir yöntem belirtmiyor. Eğer bu boşluğu bir şekilde kapatabilsek bahsedilen tecrübelerden faydalanabiliriz. Flooding yayın yapabilmemizi sağlayacak bir yöntem gibi görünüyor. Aşağıda belirtilen iki bilgiyi kullanarak Flooding kullanılabilir.

1. Birinci aşamada toplanan simetrik komşu bilgisi Flooding için yeterli görünüyor. Flooding yapmak için komşuların birbirlerini simetrik bilmeleri gerekli. İlk aşamada ise üye birimlere gönderilen ilk mesaj, göndericinin adresini de içeriyor. Bu bilgiyi alan üye birimler, göndericiyi komşu listelerine ekliyorlar ve ikinci bir defa arama yapmaya gerek kalmıyor. Buradan sonra gönderilen paketlere temsilci ismi vereceğiz çünkü buradaki paketler ve Rumor Adresleme içinde geçem temsilciler aynı amaca hizmet ediyorlar. İkinci aşama boyunca birimler temsilcileri flood ederek uzak noktalar ile haberleşirler. Bir AODV RRE veya RREP isteği veya ağ içinde her birime ulaştırılması gereken bir yayın mesajı bir Temsilci olabilir. Alıcı uçundaki birim, göndericiyi bilmiyor ise komşu listesine hemen ekleme yapar. Bu liste, daha sonra bahsedilecek protokol yığımındaki birinci katman içinde kullanılmakta olup, yayın için temel teşkil eder.
2. İkinci konu ise Floodingi kullanmak ile ilgili. Flooding implosin, overlapping, ve resource blindness problemlerine [Heinzelman et all, 1999] sahiptir. Bu koşulları engellemek için, gönderilen her Temsilci içine tekil bir kimlik bilgisi konularak bu kimlik bilgisi ile temsilcinin daha önce birim üzerinden geçip geçmediği kontrol edilebilir. Eğer temsilci noktadan ikinci defa geçmeye çalışıyor ise görmezlikten gelinmelidir. Algoritmada, tekil

kimlik bilgisi kaynak birimin Bluetooth adresine yine kendi içinde artan bir sayacı eklemesi ile oluşturulur.

Tüm ağa gönderilen temsilciler iletişim için kullanılmakla beraber, ağ topolojisini iyileştirmek için de kullanılır. Eğer bir temsilci bir birime birden fazla yerden ulaşmış ise, ağ içinde bir yerde bir çember olduğu anlamına gelir. Bu durumda bazı aracı birimler gereksiz bilgi transferi yapıyor demektir. Eğer bir birime gelen kopya Temsilci sayısı sisteme girilen bir parametre değerini geçer ise, alıcı göndericilerden bir veya birkaçı ile arasındaki bağı koparma kararı alabilir. Alıcı gönderici ile arasındaki piconette üye rölünde olan sözkonusu piconetten ayrılmalıdır. Bu işlem ağ içindeki gereksiz aracı birimleri kaldırmasına karşılık, iki uca arasındaki yolu da uzatarak ağa ek yük getirebilir. Bu sebeple sisteme verilen parametre değeri sistemdeki birimlerin kapasitesine göre ayarlanmalıdır. Bu değer eğer birimlerin kapasitesi sınırlı ise 2 olmalıdır ve kapasite arttıkça 2'den yüksek değerler almalıdır.

Tüm bunların yanında ağın bağlılığını da maksimum da tutmak gereklidir. Bunun için ise Lunar over Bluetooth adlı [Rensfelt] çalışmada bahsedilen reqlue işlemi kullanılabilir. Bu çalışmada bir birim üzerinde reqlue durumu oluştuğunda, birim etrafındaki ağı genişletmek için tekrar inquiry moduna geçmektedir. Reqlue durumu ise, bir birimin sadece bir piconet üzerinde üye olması ve başka bir piconette yönetici olmaması biçiminde açıklanmaktadır. Biz birimlerin hareket halinde iken bağlı oldukları tüm komşuları yitirebiliceğini de işin içine katarak tanımı şu şekilde genişletebiliriz: Eğer bir birim sadece bir piconet içinde üye ise ve başka bir piconet içinde yönetici değilse veya hiçbir piconet ile bağlantısı kalmamış ise, reqlue durumu oluşur. Reqlue durumu oluştuğunda, birim reqlue yapabilmek için Aşama 1'e geri döner. Bu ise birime tekrar piconetlere katılmak ve piconetleri birleştirmek için bir şans tanır.

Uygun koşullar oluştuğunda son iki işlem piconet büyüklüklerini eşitlemek, aracı birimler üzerindeki yükü azaltmak ve ağ grafiğinin bağlı olmasını sağlamak için kullanılır.

Protokol Katmanları:

Kullanımı kolaylaştırmak için protokol iki katmana bölünmüştür. İlk katman scatternet oluşumu ve flooding ile uğraşırken ikinci katman adresleme algoritmalarını çalıştırarak unicasting yapar. Bazı uygulamaların sadece ilk katmanı kullanması yeterli olurken bazı uygulamalar ikinci katmana ihtiyaç duyabilir veya iki katmanı birlikte kullanabilir.

Algoritma ise aşağıdaki gibidir.

System parametreleri: k, l, m

```

ON_START_UP()
Loop While ( A master does not send an agent )
    RANDOM_VALUE=CALC_RANDOM_VALUE(0..1)
    If ( RANDOM_VALUE < k)
        M_POOL ← EMPTY
        NEIGHBOURS ← INQUIRY()
        If (Inquiry is successful)
            Collect Information about piconets around
            Join to some piconets
            Leave some slaves
            Quit Loop
        End If
    Else
        INQUIRY_SCAN()
    End If
End Loop

```

```

SEND_MESSAGE(MSG)
    MSG.U_ID=
        GENERATE_MESSAGE_UID(BT_ADDRESS)
    MSG.SENDER = BT_ADDRESS
    FOR EACH NGH in NEIGHBOURS
        SEND_VIA_BT(NGH, MSG)  (*)

```

(*) Eğer MSN temsilcisini NGH komşusuna gönderirken üst üste l tane hata meydana gelmiş ise, NGH komşu listesinden kaldırılır.

```

ON_RECEIVE_MESSAGE(NGH, MSG) (**)
    If (ON_START_UP is running)
        Quit ON_START_UP
    End If
    If MSG in M_POOL Then

```

```

If (# of instances > m)
    Ask permission to go to Phase 1
End If
Return;
End If
Add MSG to M_POOL;
If NGH NOT IN NEIGHBOURS Then
    Add NGH to NEIGHBOURS
End If
SEND_MESSAGE(MSG)

```

(**) Eğer aynı temsilcinin bir noktaya ulaşan kopya sayısı m'den fazla ise, söz konusu temsilcinin geldiği bağlantı(lar) kaldırılır.

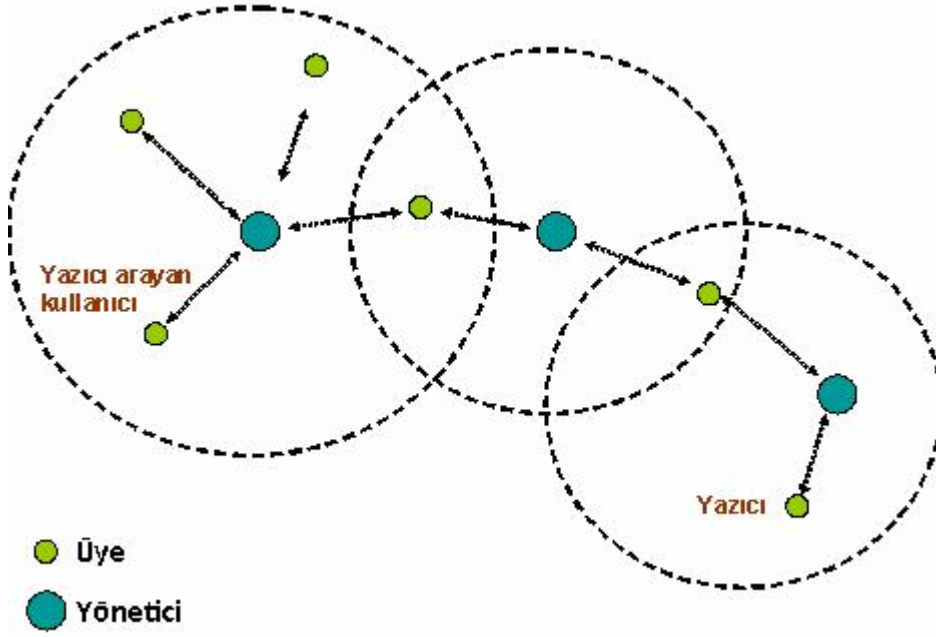
(**) Eğer reglue durumu oluşmuş ise, nokta tüm bağlantılarını kapatıp Aşama 1'e dönebilir.

Şekil 4.1. Algoritma

4.2 Kullanım Senaryosu 1

Büyük bir fuarda gittiniz ve standları dolaşıyorsunuz. Bu sırada dizüstü bilgisayarınızda bulunan bir belgeyi yazdırmanız gerekiyor. Dizüstü bilgisayarınızı açarak, çevrede bir yazıcı yani Basic Printing profiline ait bir Bluetooth cihazını aramaya başlıyorsunuz. Geliştirdiğimiz protokol sayesinde arama kriterleriniz arada üç sekme bulunan yazıcıya veya yazıcının konumunu bilen bir başka Bluetooth aygıtına ulaşıyor. Yazıcının konumunu veren bir mesaj size geri dönüyor. Siz yazıcıya doğru yöneliyorsunuz ve yazdırma işlemini gerçekleştiriyorsunuz.

Aranan servislerin türü yalnızca Bluetooth servisleri olmak zorunda değil. Servis kümesi en yakın taksi, benzin istasyonu, restoran gibi günlük servislere de kolaylıkla çevrilebilir.



Şekil.4.2. Scatternet örneği

Senaryo için bir scatternet örneği Şekil 4.2 üzerinde gösterilmektedir. Figürde, Üye 3 yazıcı arayan kullanıcıyı temsil ediyor. Üye 6 ise Basic Printing Profile'a sahip üzerinde Bluetooth cipi bulunan bir yazıcı. Algoritma sayesinde, Üye 3 çevredeki tüm yazıcıların konumunu öğrenebilmektedir. Örnek senaryoda ise Üye 6'nın konumu, Üye 3'e dönmelidir.

4.3 Kullanım Senaryosu 2

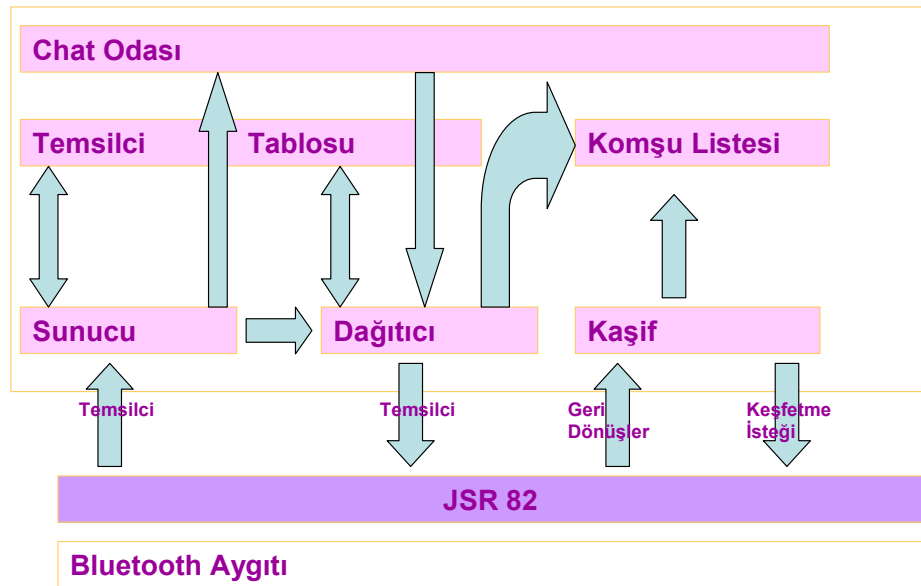
Oturduğunuz kafede cep telefonunuza yüklü chat programını açarak etraftaki kişiler chat yapıyorsunuz. Bu chat odasında sadece bluetooth dalgalarının ulaşabildiği sınırlı sayıda kişi değil tüm kafede oturanlar veya bir alışveriş merkezindeyseniz alışveriş merkezindeki herkes olabiliyor. Sizden metrelerce ötedeki biri ile bedava mesajlaşabiliyorsunuz.

Uygulama yaygın olabilmesi için cep telefonlarına yüklenebilen bir program olabilir. Bir kullanıcı chat yapmak istediğinde programı açmalıdır. Program açıldığında, algoritmada belirtildiği gibi, uygulama Aşama 1 içine girecek ve rastgele olarak inquiry ve inquiry scan yapacaktık. Aşama 2 ye geçildiğinde program ile etraftaki kişilere mesajlar gönderilebilecektir. Mesajlar tek bir kişiye olabileceği gibi tüm birimlere yayılabilmesi de mümkündür. Mesajlar tüm birimlere gönderildiğinde, bu işlem bulunan ortamda tek büyük bir chat odası yaratacaktır. Mesaj göndericinin ve alıcının biribiri ile komşu olmasını da gerek yoktur. İki noktanın birbirine mesaj gönderebilmesi için, arada bağlantıyı kuracak başka piconetlerin olması yeterli olacaktır.

4.4 Bir Uygulama: BLUECHAT

4.4.1 Bileşenler

Senaryo 2’de verilen genel chat odasının uygulamasının hazırlaması uygun görülmüştür. İlerleyen bölümlerde uygulamanın bileşenleri hakkında ayrıntılı bilgi verilecektir. Yazılım bileşenleri ve aralarında mesajlaşma aşağıda gösterilmiştir.



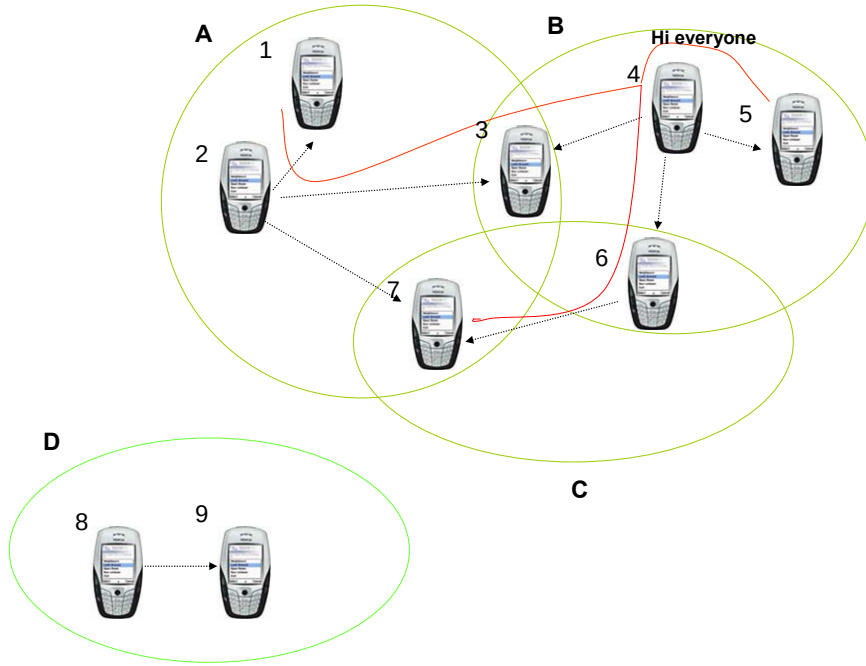
Sunucu: Gelen bağlantı isteklerini kabul eder. Her bağlantı sonucunda bir temsilci alınır. Temsilci içindeki bilgi ile sunucu, temsilcinin içeriğinin ne olduğuna ve ne yapması gerektiğine karar verir. Gelen temsilci içindeki mesaj chat mesajı ise, chat odasına gönderilerek ekrana yazdırılması sağlanır.

Dağıtıcı: Temsilcileri alarak asenkron olarak istenilen biçimde komşulara iletir. Hem chat odası hem de sunucu, dağıtıcıya ağa gönderilecek temsilcileri iletirler. Chat odasından gelen temsilciler, kullanıcı tarafından chat odasında yeni yazılan mesajlardır. Sunucudan gelen temsilciler ise, komşulardan alınan ve etrafa tekrar dağıtılması gereken temsilcilerdir.

Kaşif: Temel bluetooth aygıtı ve servis arama işlevlerini yerine getirir. Aygıt ve servis bulma süreçlerini başlatır. İstek başladıktan sonra, JSR-82 katmanından gelen geri dönüş çağrıları ile elde edilen servis ve aygıtlar komşu listesine eklenir.

Uygulama bu hali ile Bluetooth yığını dışında ve JSR-82 katmanının üstünde yer alıyor. JSR-82 ile ilgili genel bilgi aşağıda verilmektedir. Uygulamanın JSR-82 ve dolayısıyla Bluetooth yığını dışında olması da Bluetooth spesifikasyonunda önerilen biçimdir. Alternatif olarak uygulama, Bluetooth yığını içinde yer alabilirdi.

Bluechat içinde, chat odasını kontrol eden merkezi bir mekanizma yok. Tüm noktalar aynı yeteneklere sahipler. Yani her telefon bir sunucu ve istemci. Sunucu kısmı gelen mesajları kabul ederken, istemci kısmı telefona gelen mesajları ve telefon üzerindeki chat odasında yazılan mesajları komşularına iletiyor. Yalnız her telefon gelen mesajı daha önce üstünden geçmemiş olma şartı ile tekrar etrafındaki telefonlara yayıyor. Böylece herhangi bir telefon üzerinde yazılan mesaj, tüm ağı dolaşarak diğer telefonlara iletiliyor. Eğer mesaj daha önce gelmiş ise yok sayılıyor ve mesaj için herhangi bir işlem yapılmıyor.



Şekil .4.4. Scatternet Formasyon Örneği

Çalışma prensibini yukarıda oluşturulan chat oturumu üzerinden açıkça görebilmek mümkün. Ortamda toplam dokuz tane telefon var. Başlangıçta bu telefonların hiçbirinin chat odası oturumuna katılmadığını varsayalım.

Örnek senaryomuzda ilk önce, 2 nolu telefon etrafına bakıyor ve 1, 3, 7 nolu telefonları buluyor. Bu telefonların üzerinde uygun servisin varlığını da anlayan 2 numara, 1, 3, ve 7 numaralı telefonları iletişim kurulacak noktalar listesine ekliyor. Bu liste kullanılarak, 2 numara tarafından özel bir mesaj 1, 3, ve 7 numaralı telefonlara arka arkaya iletiliyor. 1, 3, ve 7 numaranın, bu ilk özel mesaj gelene kadar 2 numaradan haberi olmayacaktır. 2 numaradan gönderilen ilk mesaj, hem 1, 3, 7 numaralı telefonların 2 numaradan haberdar olmasını sağlayacak hem 2 numaraya geri dönüşü yapabilmek için gereken iletişim bilgilerini içerecek hem de 1, 3, 7 numaraların katıldıkları piconetler hakkında bilgi isteyen bir mesaj olacaktır. 1, 3 ve 7 numaranın önceden katıldıkları bir piconet henüz olmadığından bu mesaja gönderilen cevap, herhangi bir piconet bilgisi içermeyecektir. Ancak 1, 3, ve 7 numaralı telefonlar özel mesaj ile birlikte gelen iletişim bilgilerini alarak 2 numarayı kendi üzerlerindeki haberleşebilecekleri noktalar listesine eklerler. Özel mesajı

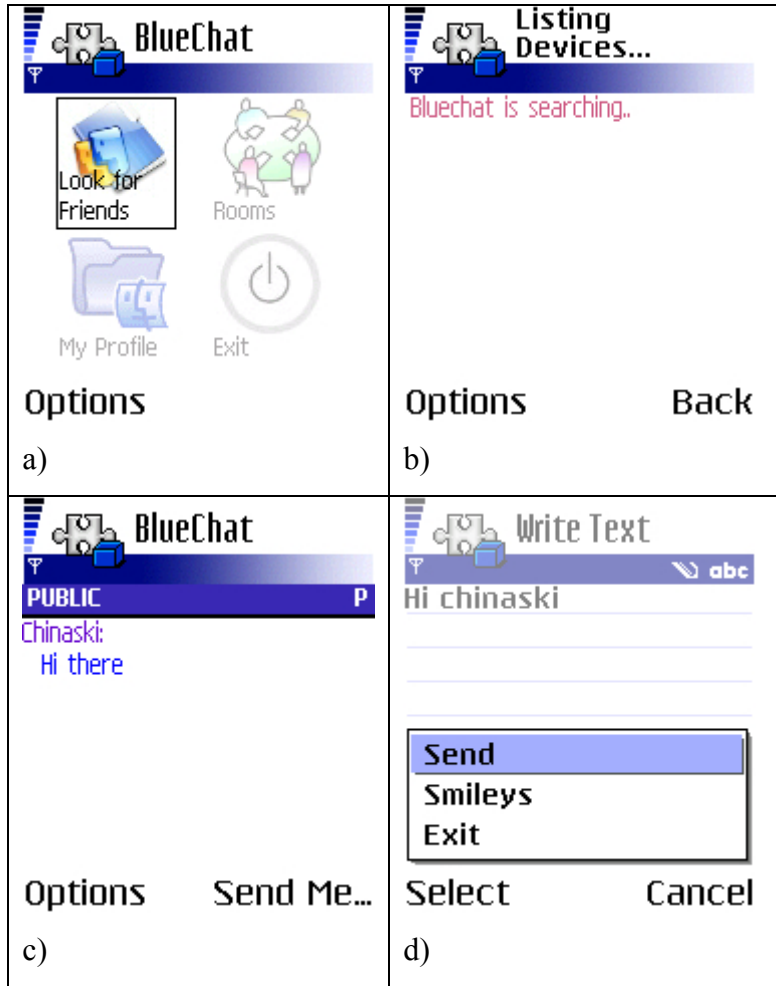
aldıktan sonra, 3 numarayı kullanan kullanıcı mesaj yazmaya karar verirse, yazılan mesaj önce 2 numaralı telefona, oradan da 1 ve 7 numaralı telefona iletiliyor.

Bu iletişim devam ederken, 4 numaralı telefon etrafına bakmayı başlatırsa, 3 ve 5 nolu telefonlarda uygun servisin olacağını görecektir. 3 ve 5 numarayı iletişim kurulabilecek noktalar listesine ekleyen 4 numaranın ilk mesajı 3 ve 5 numaralı telefonlardan piconet bilgisi isteyecektir. 3 numaranın katıldığı bir piconet olmasına rağmen algoritmada belirtilen durumlar oluşmadığından, yeni piconet değişiklik olmadan kurulacak ve 3 numara iki piconet arasında çalışmaya başlayacaktır. Eğer 4 numara, 3 numaranın yanında 7 numarayı da bulabiliyor olsaydı, 3 ve 7 tarafından gönderilen yönetici listeleri aynı olacaktı. Bu durumda 4 numara eleme yapacak ve sadece 3 ve 7 numaranın birisi ile aynı piconet içinde çalışmaya devam edecekti.

Bütün bu ağ yapısına ek olarak, 6 numara etrafına bakmayı seçmiş olsun ve 7, 4 numaraları bulmuş olsun. 4 numara kendi piconetinde yönetici olduğundan ve henüz dolu olmadığından, 6 numara, 4 numaranın yönetici olduğu piconete üye olarak katılma kararı verecektir ve 4 ile 6 arasında Yönetici-Üye değişimi yapılacaktır. 4 ve 7 numaralar ise farklı yönetici listeleri göndereceklerinden, 6 ve 7 arasında piconet oluşturulmasında bir problem olmayacaktır. Ancak ağ içinde gönderilen temsilcilerden birinin 6 numaraya 2 defa ulaşması durumunda, 6 numara ya 4 ya da 7 ile arasındaki bağı koparacaktır. Böylece piconetler arası döngüler kaldırılmış olacaktır.

Tüm bunlar ek olarak iletilen her mesaja verilen tekil numara sayesinde, hiçbir mesaj ağ üzerindeki noktalar üzerinden iki defa geçmeyecektir. Bir mesajı ikinci defa aldığını anlayan telefon, bu mesajı yok sayacak ve çemberi kaldırmak için yukarda anlatılan önlemleri uygulayacaktır.

8 ve 9 numaralı telefonlar ise, direkt bir bağlantı noktasına sahip olmadıkları için, oluşan scatternet dışında kalacaklar ve kendi aralarındaki chat oturumunu kullanacaklardır.



Şekil .4.5. Uygulama önyüzünden örnekler.

Şekil.4.5’de uygulamanın önyüzünden örnekler verilmektedir. İlk kısımda yeralan “Look for Friends” iki aşamalı bir işlem gerçekleştirmektedir. İlk aşamada çevredeki tüm Bluetooth aygıtları bulunmakta, ikinci aşamada ise aygıtlar üzerindeki servisler aranmaktadır. Bu kısım, yukarda belirtilen algoritmanın ilk aşamasını çalıştırmaktadır. İşlemin sonunda elde edilen komşu listesi kullanılarak, etrafa profil isteği gönderilmektedir. Profil isteğinin içinde, isteği yapan cihazın profil bilgileri de yer almaktadır. İsteği alan cihazlar kendi profillerini temsilci içine koyarak yanıt vermektedir. Tüm bu işlemlerin sonunda, ağdaki herkes birbirinin prdfil bilgilerine sahip olmaktadır ve “Look for Friends” bölümünde bu profiller listelenmektedir. İkinci kısımda protokolün ilk aşaması çalışırken görüntülenen önyüz görünmektedir. Üçüncü kısımda Chat Room’dan bir görüntü yeralmaktadır. Dördüncü kısım ise, yeni mesaj yazma bölümüdür.

5.SONUÇ

Bluetooth kolay bir bağlantı teknolojisi olmanın yanında gelen market baskısının da etkisi ile başka alanlarda kullanılmak üzere iyileştirmelere gebe kalacağını görüyoruz. Daha şimdiden Bluetooth'u kullanan pervasive ve ubiquitous uygulamalar mevcuttur. Bu kullanım senaryolarının artmasında Bluetooth'un varolan spesifikasyonu önemli bir engeldir çünkü amaç olarak bu tür kullanımlar gözönüne alınarak tasarlanmamıştır. Bu çalışma ise Bluetooth'un mevcut spesifikasyonlarını kullanarak bir Scatternet formasyonu tasarlamış ve uygulamaya almıştır.

Yapılan çalışmanın tamamen Bluetooth spesifikasyonlarını kullanması sebebi ile günümüz cihazları üzerinde uygulanabilir olduğu varsayımı olarak yola çıkılmıştır. Ancak ilk sonuçlarda, telefonlar üzerindeki Bluetooth modüllerinin sanıldığı kadar aksine zorunlu olmayan Bluetooth spesifikasyonlarını uygulamadığı bu sebep ile de algoritmanın çalışması için gereken temel gereksinimleri yerine getirmedikleri görülmüştür. Telefonlarda Bluetooth modüllerinin bu özelliklere sahip olmaması ise telefonlardaki güç kaynağının yeterli görülmemesine ve fiziksel yer sıkıntısına bağlanabilir. Telefonların uygulamadığı ve bizim ihtiyaç duyduğumuz temel özellikler ise:

1. Bir aygıtın piconetler arası çalışabilmesi,
2. Aygıtların Yönetici-Üye değişimi yapmasıdır.

Algoritmayı tam anlamıyla kullanamasa bile, uygulama tamamlanarak internete bedava indirilebilir biçimde konulmuştur. Bedava olarak dağıtılan uygulamaya gösterilen ilgiden görülmüştür ki, verdiğimiz senaryolara benzer uygulamaların ve çok hızlı kurulabilen Bluetooth topolojilerinin kullanım alanları oldukça fazladır.

Bundan sonraki çalışmalarımızda ise algoritmayı kullanılabilir bir platforma taşımayı veya simülatörler kullanarak algoritmanın çalışma prensiplerini daha derinden incelemeyi amaçlamaktayız.

KAYNAKLAR

- [1] A Bluetooth Based Sensor Network for Civil Infrastructure Health Monitoring
VIPIN MEHTA, MAGDA EL ZARKI, Wireless Networks 10, 401–412, 2004,
Kluwer Academic Publishers. Manufactured in The Netherlands.
- [2] A Ubiquitous Stable Storage for Mobile Computing Devices, Symposium on
Applied Computing, Proceedings of the 2001 ACM symposium on Applied
computing, Pages: 401 – 404, Las Vegas, Nevada, United States, 2001
- [3] BlueMesh: Degree-Constrained Multi-Hop Scatternet Formation for Bluetooth
Networks CHIARA PETRIOLI, STEFANO BASAGNI, IMRICH CHLAMTAC
Mobile Networks and Applications 9, 33–47, 2004 Kluwer Academic Publishers.
Manufactured in The Netherlands.
- [4] BlueStar: Enabling Efficient Integration between Bluetooth WPANs and IEEE
802.11 WLANs, CARLOS DE M. CORDEIRO, SACHIN ABHYANKAR,
RISHI TOSHIWAL And DHARMA P. AGRAWAL, Mobile Networks and
Applications 9, 409–422, 2004 Kluwer Academic Publishers. Manufactured in
The Netherlands.
- [5] Bluetooth Specification, <http://www.bluetooth.org>
- [6] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed di@usion: a scalable
and robust communication paradigm for sensor networks," Proceedings of ACM
MobiCom '00, Boston, MA, 2000, pp. 56-67.
- [7] C. Law, A.K. Mehta and K.-Y. Siu, A new Bluetooth scatternet formation
protocol, Mobile Networks and Applications (MONET), Special Issue on Mobile
Ad Hoc Networks 8(5) (2003).
- [8] C. Perkins and R. Koodli. Service discovery in on-demand ad hoc networks.
IETF Internet Draft, October 2002. Work in progress.
- [9] D. Braginsky and D. Estrin, Rumor Routing Algorithm for Sensor Networks," in
the Proceedings of the First Workshop on Sensor Networks and Applications
(WSNA), Atlanta, GA, October 2002.
- [10] David B. Johnson, Davis A. Maltz, "The Dynamic Source Routing Protocol
for Mobile Ad Hoc Networks" IETF Draft, Oct. 1999.
- [11] F. Ye, A. Chen, S. Liu, L. Zhang, A scalable solution to minimum cost
forwarding in large sensor networks", Proceedings of the tenth International

- Conference on Computer Communications and Networks (ICCCN), pp. 304-309, 2001.
- [12] G. Tan, A. Miu, J. Guttag and H. Balakrishnan, An efficient scatternet formation algorithm for dynamic environment, in: *Proceedings of the IASTED Communications and Computer Networks (CCN)*, Cambridge, MA (4–6 November 2002).
 - [13] G. Záruba, S. Basagni and I. Chlamtac, BlueTrees – Scatternet formation to enable Bluetooth-based personal area networks, in: *Proceedings of the IEEE International Conference on Communications, ICC 2001*, Helsinki, Finland, Vol. 1 (11–14 June 2001) pp. 273–277
 - [14] Gy. Miklós , A. Rácz , Z. Turányi , A. Valkó , P. Johansson, Performance aspects of Bluetooth scatternet formation, *Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*, November 20, 2000, Boston, Massachusetts
 - [15] Heinzelman, W., Kulik, J. & Balakrishnan, H. (1999), Adaptive Protocols for Information Dissemination in Wireless Sensor Networks, in ‘Proc. Fifth ACM/IEEE Int. Conf. on Mobile Computing and Networking’, pp. 174–185.
 - [16] Lunar over Bluetooth, Olof Rensfelt, Uppsala University Master Thesis, UPTEC IT04011, unpublished
 - [17] M. Leopold, M.B. Dydenborg, and P. Bonnet. Bluetooth and sensor networks: A reality check. In *Proc. 1st ACM Conf. Embedded Networked Sensor Systems (SenSys 2003)*, pages 103–113. ACM Press, New York, November 2003.
 - [18] Olga Rats’mor, Dipanjan Chakraborty, Anupam Joshi, T’mothy Finn And Yelena Yesha, Service Discovery in Agent-Based Pervasive Computing Environments, *Mobile Networks and Applications Volume 9 , Issue 6*, Pages: 679 - 692 I ACM Press, New York, December 2004.
 - [19] Park, V. and S. Corson, 2001. Temporally-ordered Routing algorithm (TORA). Internet Draft, draft-ietfmanet tora-spec-04 txt.
 - [20] Perkins, C.E., E.M. Royer and S.R. Das, 2001. Ad hoc on-demand distance vector (AODV) routing. draftietf-anet-aodv-08-txt, IETF Internet Draft.
 - [21] R. Morrow, Bluetooth Operation and Use, 2002

- [22] T. Salonidis, P. Bhagwat, L. Tassiulas and R. LaMaire, Distributed topology construction of Bluetooth personal area networks, in: Proceedings of the IEEE Infocom 2001, Anchorage, AK (April 22–26,2001) pp. 1577–1586.
- [23] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Mi-crosensor Networks," Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS '00), January 2000.
- [24] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks", in SIGMOD Record, September 2002.

ÖZGEÇMİŞ

Sami Altundağ 10.10.1976 doğumludur. Lisans öğrenimi, İstanbul Bilgi Üniversitesi - Bilgisayar Bilimleri Bölümü'nde tamamlamıştır. Yüksek Lisans eğitimine, Gebze Teknoloji Enstitüsü - Bilgisayar Mühendisliği Ana Bilim Dalı'nda devam etmektedir. 09.1999 – 10.2004 tarihleri arasında Türkiye Genel Sigorta A.Ş.'de yazılım uzmanı olarak çalışmıştır. 10.2004 – 05.2006 tarihleri arasında Persona İletişim A.Ş. çalışanı olarak Turkcell 'de proje liderliği yapmıştır. 05.2006 tarihinden itibaren ise HSBC Türkiye'de proje lideri olarak görev yapmaktadır.

Ekler:

Ek1 : Bluetooth Profilleri

Profiller

	Kodu	Vers.	Profil Adı
K1	<i>GAP</i>	1.1	Generic Access Profile
K2	<i>SDAP</i>	1.1	Service Discovery Application Profile
K3	<i>CTP</i>	1.1	Cordless Telephony Profile
K4	<i>IP</i>	1.1	Intercom Profile
K5	<i>SPP</i>	1.1	Serial Port Profile
K6	<i>HS</i>	1.1	Headset Profile
K7	<i>DNP</i>	1.1	Dial-up Networking Profile
K8	<i>FP</i>	1.1	Fax Profile
K9	<i>LAP</i>	1.1	LAN (Local Area Network) Access Profile
K10	<i>GOEP</i>	1.1	Generic Object Exchange Profile
K11	<i>OPP</i>	1.1	Object Push Profile
K12	<i>FTP</i>	1.1	File Transfer Profile
K13	<i>SP</i>	1.1	Synchronization Profile
Available Now	<i>ESDP</i>	0.95a	Extended Service Discovery Profile (for Universal Plug and Play)
Premium Access *	<i>A2DP</i>	0.95b	Advanced Audio Distribution Profile
Premium Access *	<i>AVRCP</i>	0.95b	Audio Video Remote Control Profile
Premium Access *	<i>BIP</i>	1.0	Basic Imaging Profile
Premium Access *	<i>BPP</i>	0.95a	Basic Printing Profile
Premium Access *	<i>CIP</i>	1.0	Common ISDN Access Profile
Premium Access *	<i>GAVDP</i>	0.95b	Generic Audio Video Distribution Profile

Premium Access *	HFR	0.96	Hands-Free Profile
Premium Access *	HCRP	0.95a	Hardcopy Cable Replacement Profile
Premium Access *	HID	0.95c	Human Interface Device Profile
Premium Access *	PAN	0.95a	PAN (Personal Area Networking) Profile
Premium Access *	SAP	0.95c	SIM Access Profile

Ek 2: Bluechat için Kullanıcı Yorumları

[10/10 on November 13, 2005] HAHAAAAHA (yoyoyouo@hotmail.com) from The BIN, using Sonyericsson w800i

Top you could find but no one else has it!!

[1/10 on November 12, 2005] Mohamad Charaf (sharaf_m@hotmail.com) from Alexandria, Egypt, using Nokia 6021

NullPointerException was thrown.

[1/10 on November 12, 2005] Ed (orellanacastillo@gmail.com) from San Salvador, El Salvador, using Sony ericsson T610

Claims to be compatible with all J2ME phones. I can download it to my pc and send it to my phone, but once the transfer is done the message INVALID APPLICATION shows up on the screen

[10/10 on November 10, 2005] koorosh Gh. from Iran, Shiraz, using nokia 6600

[3/10 on November 9, 2005] dez from philippines, using SE t630

*did not work on my fone.. *sigh* invalid application... though seems quite useful*

[4/10 on November 8, 2005] Mihai (logan_bad@yahoo.com) from Romania, using Nokia 6230i

i can` t start the application..it say that my name is error..what should i do?? please send me an e-mail if u know.

[10/10 on October 31, 2005] Smile (smile@runbox.com) from Istanbul, using imate PDA2K

Wonderful

[1/10 on October 31, 2005] EroWid from Helsinki, using Nokia N-Gage

In finland... Gets system error on install.

[6/10 on October 30, 2005] joso from portugal, using nokia 6230

[6/10 on October 28, 2005] rajesh (rajeshcameraman@rediffmail.com) from india,pune, using nokia 6630

i really fund it usefull. thanks

[1/10 on October 27, 2005] migo from scotland, using motorola v620

does not work on the v620 gives you an application error

[10/10 on October 26, 2005] Swirlymonkey (blitzerige@hotmail.com) from london, using k608i

NICE! very well made, i was sending everyone the jar file via bluetooth at school and i catted all the way through the lesson - THANKS FOR THIS MAN!

[9/10 on October 21, 2005] Sami from Turkei, using Nokia 6600

I succesfully downloaded and installed the app. I dont know if upper case letters might be problem for some phones. Also your midlet should be able to open bluetooth connection. some phones that have bluetooth dont let it.

[1/10 on October 20, 2005] Stown Nelani (tnelani@yahoo.co.uk) from South Africa,East London, using Sharp GX15

this application doesn` t work for anyone and i think it due to the fact that the file name is wrong "bluechat..jar"!did anyone notice the extention?well who ever uploaded it should just change the ext....:that`s my opinion from what i know about computers

[4/10 on October 19, 2005] Vagelis from Athens Greece, using Siemens s65

works fine.u ppl need a phone that allows java to access bluetooth non of your phones does that.sorry :P

[1/10 on October 19, 2005] David Lindman (dlindman831@gmail.com) from Germany, using Motorola E550

First time didn` t install. With updated jar file, would install, but get application error when I try to run. Guess it is not supported with the Motorola E550.

[1/10 on October 19, 2005] Alex from Italy, using sgh D500 E

Don` t Work on my d500....Error when i try to load the app.(UNEXPECTEDLY QUIT)

[1/10 on October 17, 2005] Richard Rodriguez from USA, Washington, using Razr V3

The application will not load due to an "Application Error" that appears after the program begins to load.

[1/10 on October 15, 2005] Razr_usr from USA, Cleveland, using Razr V3

The application will not load due to an "Application Error" that appears after the program begins to load.

[1/10 on October 15, 2005] David Lindman (dlindman831@yahoo.com) from Stuttgart Germany, using Motorola E550

Would not install on E550, so I gave it a 1 for that device. Perhaps other devices will do better. I hope they can get it working for my phone.

[8/10 on October 15, 2005] Norway from Norway, using Sony Ericsson K700i

Doesn` t work on Sony Ericsson K700i, get this message: "Program Error".

But this is indeed a great idea, due to the fact that it`s not working on my phone => great job, and keep up!

[1/10 on October 14, 2005] Alexander from Germany, using Sony Ericsson S700i
Doesn`t work for me. "Error in application" on start :-(

[1/10 on October 14, 2005] Alan (alanb@quantumconsultinggroup.com) from England, Dunstable, using Motorola Razr V3
Did not work. When installing it said "Invalid File"

[1/10 on March 16, 2006] Tim from BELGIUM, Zottegem, using SonyEricsson K700i
Application error.
It would not work on my mobile phone.

[10/10 on March 13, 2006] Bupahs from USA, Pennsylvania, using Motorola v360
Perfect application!!

[1/10 on March 7, 2006] Emanuel (emanuel.v26@gmail.com) from Philippines, Manila, using Sony Ericsson K700i
bluechat program did not work on my phone, the "APPLICTION ERROR"

[7/10 on February 28, 2006] Andro (andro_krezic@yahoo.co.uk) from Croatia, Zadar, using Motorola L6
This application is working great on motorola L6!!!

[9/10 on February 28, 2006] wisdom (wisdomka@pgzim.co.zw) from zimbabwe, using siemens cx65

[6/10 on February 22, 2006] edan (edantenan2003@gmail.com) from indonesia, using n-gage
After installing this program...My phone have a error message such Appliction : System Error.
So...anyone know what the program can run at nokia n-gage chat via bluetooth ?

[5/10 on February 20, 2006] James from uk, london, using Sony k608i and LG u880
Seems to work fine on the k608i, but fails to start up on the LG u880.

[10/10 on February 18, 2006] asim from uk, using Nokia 6230i
Brilliant app!!

All my friends use it at school all the time.

[1/10 on February 15, 2006] Caleb (sneakyтики64@gmail.com) from Morgantown IN USA,

using Motorola RAZR V3 Black

I had the same problem as everyone else but i wasnt sure if there was a fix for it that someone had found.. i dug deep into the bluetooth files on my phone and unlocked every locked feature there is and it still would not do it... i just want to message with my friend back and fourth in french class... if you can help e-mail me at sneakyтики64@gmail.com

[1/10 on February 9, 2006] Greg from San Francisco, CA, using Motorola v330
same experience as the rest of motorola users: immediate exit with `application error`

[5/10 on February 6, 2006] Fabio A from Brazil, Sao Paulo, using Motorola E398
Great idea. Unfortunately it doesn`t work for E398 (application error). Maybe something to do with Java API for bluetooth. Any suggestions on how to overcome that?

[5/10 on February 6, 2006] cedric from zinken, using k750i

[5/10 on February 6, 2006] Roney from Brazil, using V555
This program uses JSR-82. Motorola`s Phones seems to be not compatible with this. Look at: <http://en.wikipedia.org/wiki/JSR-82>

[8/10 on January 30, 2006] uzair (uzair_k_2000@hotmail.com) from India, using 6600
good.. i would like the source code for it.... i wish to develop it further.. starting from the beginning would take time... so...

[1/10 on January 23, 2006] Matt (doggyrunt@aol.com) from England, using Sony ericsson K700
Wasnt very good for the k700 never worked always came up with an applic error please can you sort this out.?

[7/10 on January 23, 2006] yosh (krepus@yahoo.com) from phils, using se k750i
the program was installed successfully, can send to a N6600, but cant received msgs. it displays an error in server something.

[6/10 on January 14, 2006] syamarnaz (syamarnaz@yahoo.com) from Malaysia, using E398
I was installed for two times on my mobile, but `application error` it`s does`nt work.

[9/10 on January 7, 2006] Timetraveler (hidder_mf@gmx.de) from Germany, Dresden, using W800i
I really love this programm, but the problem is, that it dosn`t work at the handys of my most friends :-(

e.g. the Motorola Razer v3 or similar and a nokia (sorry, no clue which, but not so old)

[7/10 on January 2, 2006] anthony (anthony24x@hotmail.com) from santo domingo, using motorola v635

application error

[10/10 on December 28, 2005] Steven from Scotland, using SE v600i

I`ve not actually tried the chat but the program has been installed and is running so far. Now to try the bluetooth chat xD

[9/10 on December 27, 2005] Chief (orderof2000@yahoo.com) from Greece, using Nokia 6600

Perfect 4 students

[1/10 on December 26, 2005] Andro (andro_krezic@yahoo.co.uk) from croatia zadar, using motorola v547

this would be great but always is APPLICATION ERROR

[5/10 on December 22, 2005] M MILES from UK, using Motorola Razr V3

This would be a fantastic product, but if you are useing a V3 forget it, application error.

If you are responsible for this product, please, please fix it!!

[1/10 on December 18, 2005] boran from turkiye, using sonyericsson K700i

application error.

doesnt work on k700i

[5/10 on December 18, 2005] jimmy (loudxandxqueer@gmail.com) from newport beach, ca, usa, using motorola razr V3

makes me sad.

application error after download and run.