

## Mobil Peer-To-Peer Ağlarda Bluetooth Üzerinden Anlık Mesajlaşma Protokolü: IMOB

Oğuz DÖNMEZ<sup>1</sup>

<sup>1</sup>Tire Kutsan Meslek Yüksekokulu, Ege Üniversitesi, 35900, Tire, İzmir, Türkiye  
(oguz.donmez@ege.edu.tr)

Gönderim Tarihi: 26.01.2018

Kabul Tarihi: 06.06.2018

**Özetçe---**Servis tabanlı yazılım geliştirmede servis sağlayıcılar servislerini servis istemcilerinin kullanımına sunarlar. Bu, servis istemcilerinin kaynakların yetmemesi durumunda bir avantaj sağlar. Bu çalışmada servis tabanlı yazılım geliştirme ve mobil peer-to-peer ağlar incelenip anlık mesajlaşma protokolü olan IMOB geliştirilmiştir. Bu protokol açıklanıp örnek olarak servis sağlayıcı rolü modellenmiştir.

**Keywords---**servis tabanlı yazılım geliştirme, mobil peer-to-peer, anlık mesajlaşma protokolü.

## Instant Messaging Protocol Over Bluetooth On Mobile Peer-to-Peer Networks: IMOB

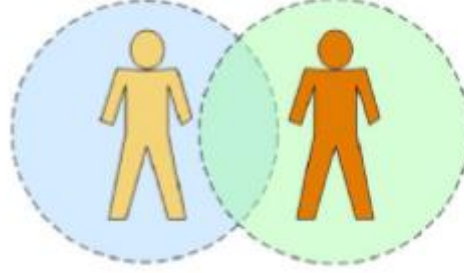
**Abstract---**In service oriented software development, service providers offer services to service clients. This is an advantage if the resources of the service clients are not sufficient. In this study, service-oriented software development and mobile peer-to-peer networks were examined and IMOB, an instant messaging protocol via bluetooth, was developed. This protocol was described and the role of the service provider was modeled.

**Keywords---**service oriented software development, mobile peer-to-peer, instant messaging protocol.

### 1. Giriş

Geleneksel olarak, dağıtık hesaplamada ana görüş istemci-sunucu yaklaşımıdır. Bununla birlikte, istemci-sunucu yaklaşımında servislerin ve bilginin merkezileştirilmesi performans darboğazı ve tek nokta hatası yüzünden azalmış uygunluk problemlerini doğurur. Peer-to-peer mimarisinde, ağdaki bütün bilgisayarlar eşit olarak davrandığı için bu problem ortadan kalkar (Singh, 2001). Eşit düğümlü bir P2P sistemi, sabit bir altyapı gerektirmez. Gelecekte, her cihaz hesaplama ve haberleşme yeteneklerine sahip olacak ve ortama kendi özel servislerini sunacaktır. Aynı mantıkla, bu cihazlar etkin bir şekilde çalışabilmek için diğer cihazların servislerini kullanmak isteyecektir (Gehlen ve Pham, 2005). Halen, çoğu peer-to-peer uygulamaları ve mimarileri İnternet gibi sabit ve kablolu altyapıda çalışmak üzere tasarlanmıştır. Kablosuz ağ teknolojilerinin geliştirilmesiyle, mobil cihazlar ve mobil cihazlar için programlama ortamı peer-to-peer hesaplamayı kablosuz ortama taşımaya mümkün kılmıştır (Kortuem, Schneider, Thaddeus, Thompson, Fickas ve Segall, 2001; Maibaum ve Mundt, 2002). Çoğu kablosuz cihazlar, kızılötesi ve Bluetooth gibi kişisel alan ağı teknolojilerinden bazılarını destekler (Miller ve Bisdikian, 2002). Mobil cihazlar ve bilgisayarlar ya da mobil cihazlar ve çevre birimleri (kulaklık ve klavye gibi) arasında veri transferi için genelde kişisel alan ağı kullanılır. Bir kişisel alan ağı, mobil cihazın etrafında yakın cihazlarla veri değişimine izin veren dijital bir küre olarak görülebilir. Bu dijital küre, mobil bilgisayar destekli birlikte

çalışmayı (mobile CSCW) desteklemek için kullanılabilir (Wiberg ve Grönlund, 2000). Şekil 1, mobil bilgisayar destekli çalışmayı; iki kullanıcı cihazında çalışan uygulamanın veriyi değişmesinde dijital küreleri çakıştırarak nasıl sağladığını gösterir. Böyle bir ortamda, mobil peer-to-peer desteği ve mobil geçici ağların (MANET'ler) kurulum desteği gereklidir (Wang, Björnsgård ve Saxlund, 2007). Kablosuz olarak bağlı, sürekli kendini yapılandıran, altyapıdan bağımsız mobil cihaz ağına MANET denir. (Wikipedia, 2018). MANET'ler, sabit bir altyapı sağlamazlar. MANET'ler, kalıcı olarak ortama giren ve çıkan düğümlerle karakterize edilirler. Bu yüzden servisler güvenilir değildir (Gehlen ve Pham, 2005).



Şekil 1. Kesişen Dijital Küreler

Dağıtık sistemler için yazılım geliştirmede servis tabanlı yazılım mimarilerinin (Service Oriented Architecture-SOA) rolü daha fazla popüler olmaya başladı (Gehlen ve Pham, 2005). SOA, bir ağ üzerinden bir iletişim protokolü aracılığıyla uygulama bileşenleri tarafından diğer bileşenlere hizmetlerin verildiği bir yazılım tasarımı tarzıdır (Wikipedia, 2018). Mobil dağıtık uygulamalar inşa etmede engellerden biri bütün dağıtık sistemin verimliliğini maksimumlaştırmadır. Kullanıcının cihazlarıyla etkileşimde bulunma gerekliliğinin yanında, cihazlar da birbirleri arasında işbirliği yapmak zorundadır. Dağıtık sistemlere yazılım geliştirmede makine-makine arayüzlerinin çokluğu engeliyle başa çıkabilmek için, SOA umut verici bir teknolojidir (Gehlen ve Pham, 2005).

Bu çalışmada, mobil cihazlarda servis tabanlı mimariyi kullanan mobil peer-to-peer ağlarda bluetooth üzerinden anlık mesajlaşma protokolü geliştirmek amaçlanmaktadır.

## 2. Servis Tabanlı Mimari

SOA, üç rol tanımlar; servis istemcisi, servis sağlayıcısı, ve servis kayıtçısı. Diğer yazılım etmenleriyle etkileşimde bulunan bir yazılım etmeni, bir ya da daha fazla role sahip olabilir. Sağlayıcılar, servislerini servis kayıtçısına yayımlarlar. Ortamda birden fazla servis kayıtçısı olduğu zaman, servis kayıtları kopyalanmalıdır. İstemciler, servisleri arama ve servis tanımlamasına erişerek onlarla bütünleşme için kayıtçiyi kullanırlar. Bu tanımlama, servise erişmek için gerekli bütün bilgiyi içerir (Gehlen ve Pham, 2005).

## 3. IMOB Protokolü

Bu bölümde kullanılacak kısaltmalar: SK, servis kayıtçısını; SS, servis sağlayıcısını; Sİ, servis istemcisini temsil eder.

Geliştirilen IMOB protokolü: *kayıtçı-sağlayıcı*, *kayıtçı-istemci*, *sağlayıcı-istemci* ve *istemci-istemci* altprotokollerinden oluşur.

SK ve SS'nin iki taraf olduğunu varsayalım. *Kayıtçı-sağlayıcı* altprotokolü aşağıdaki gibidir (Shmatikov ve Mitchell, 2000; Drielsma ve Mödersheim, 2003; Onieva, Zhou ve Lopez, 2005; Ogata ve Futatsugi, 2003; Carbone, Honda ve Yoshida, 2007; Kovan, 2005; Buttyan ve Hubaux; Rubin ve Honeyman, 1993; Islam ve

Zaid, 2008; Ratnasamy, Francis, Handley, Karp ve Shenker, 2001; Rowstron ve Druschel, 2001; Stoica, Morris, Karger, Kaashoek ve Balakrishnan, 2001; Zhao, Kubiatawicz ve Joseph, 2001):

1.  $SS \rightarrow SK$  : *konumbelirleyici, servisadı*
2.  $SK \rightarrow SS$  : *sonuç*

Altprotokolün birinci adımında:  $SS$ , servisin *konumbelirleyicisini* ve adını (*servisadı*) gönderir. İkinci adımda:  $SK$ , servisin *konumbelirleyicisi* ve adını kaydetme işleminin sonucuyla (*sonuç*) cevap verir.

$SK$  ve  $S\hat{I}$ 'nin iki taraf olduğunu varsayalım. *Kayıtçı-istemci* altprotokolü aşağıdaki gibidir (Shmatikov ve Mitchell, 2000; Drielsma ve Mödersheim, 2003; Onieva, Zhou ve Lopez, 2005; Ogata ve Futatsugi, 2003; Carbone, Honda ve Yoshida, 2007; Kovan, 2005; Buttyan ve Hubaux; Rubin ve Honeyman, 1993; Islam ve Zaid, 2008; Ratnasamy, Francis, Handley, Karp ve Shenker, 2001; Rowstron ve Druschel, 2001; Stoica, Morris, Karger, Kaashoek ve Balakrishnan, 2001; Zhao, Kubiatawicz ve Joseph, 2001):

1.  $S\hat{I} \rightarrow SK$  : *servisadı*
2.  $SK \rightarrow S\hat{I}$  : *konumbelirleyicisayısı, konumbelirleyici(ler)*

Altprotokolün birinci adımında:  $S\hat{I}$ , servisin adını (*servisadı*) gönderir. İkinci adımda:  $SK$ , eşleşen servis(ler)in *konumbelirleyicisayısı* ve *konumbelirleyicisi* (*konumbelirleyiciler*) ile cevap verir.

*Sağlayıcı-istemci* altprotokolü: *kayıt, sorgu ve kontrol* altprotokollerinden oluşur.  $SS$  ve  $S\hat{I}$ 'nin iki taraf olduğunu varsayalım. *Kayıt* altprotokolü aşağıdaki gibidir (Shmatikov ve Mitchell, 2000; Drielsma ve Mödersheim, 2003; Onieva, Zhou ve Lopez, 2005; Ogata ve Futatsugi, 2003; Carbone, Honda ve Yoshida, 2007; Kovan, 2005; Buttyan ve Hubaux; Rubin ve Honeyman, 1993; Islam ve Zaid, 2008; Ratnasamy, Francis, Handley, Karp ve Shenker, 2001; Rowstron ve Druschel, 2001; Stoica, Morris, Karger, Kaashoek ve Balakrishnan, 2001; Zhao, Kubiatawicz ve Joseph, 2001):

1.  $S\hat{I} \rightarrow SS$  : *cinsiyet, boy, mail, konumbelirleyici, testkonumbelirleyici*
2.  $SS \rightarrow S\hat{I}$  : *sonuç*

Altprotokolün birinci adımında:  $S\hat{I}$ , servis istemcisinin kullanıcılarının *cinsiyet, boy ve mailini* gönderir. Aynı zamanda sunucunun *konumbelirleyicisi* ve sunucu istemcisinin *testkonumbelirleyicisini* gönderir. İkinci adımda:  $SS$ , kaydetme işleminin sonucuyla (*sonuç*) cevap verir.

*Sorgu* altprotokolü aşağıdaki gibidir (Shmatikov ve Mitchell, 2000; Drielsma ve Mödersheim, 2003; Onieva, Zhou ve Lopez, 2005; Ogata ve Futatsugi, 2003; Carbone, Honda ve Yoshida, 2007; Kovan, 2005; Buttyan ve Hubaux; Rubin ve Honeyman, 1993; Islam ve Zaid, 2008; Ratnasamy, Francis, Handley, Karp ve Shenker, 2001; Rowstron ve Druschel, 2001; Stoica, Morris, Karger, Kaashoek ve Balakrishnan, 2001; Zhao, Kubiatawicz ve Joseph, 2001):

1.  $S\hat{I} \rightarrow SS$  : *cinsiyet, boy*
2.  $SS \rightarrow S\hat{I}$  : *mailsayısı, mail(ler), konumbelirleyici(ler)*

Altprotokolün birinci adımında:  $S\hat{I}$ , aranan kullanıcının *cinsiyet ve boyunu* gönderir. İkinci adımda:  $SS$ , eşleşen kullanıcıların maillerinin sayısı (*mailsayısı*), *mail(ler)* ve *konumbelirleyici(ler)* ile cevap verir.

*Kontrol* altprotokolü aşağıdaki gibidir (Shmatikov ve Mitchell, 2000; Drielsma ve Mödersheim, 2003; Onieva, Zhou ve Lopez, 2005; Ogata ve Futatsugi, 2003; Carbone, Honda ve Yoshida, 2007; Kovan, 2005; Buttyan ve Hubaux; Rubin ve Honeyman, 1993; Islam ve Zaid, 2008; Ratnasamy, Francis, Handley, Karp ve Shenker, 2001; Rowstron ve Druschel, 2001; Stoica, Morris, Karger, Kaashoek ve Balakrishnan, 2001; Zhao, Kubiatawicz ve Joseph, 2001):

- |  |
|--|
| 1. $S\dot{I} \rightarrow SS$ : <i>konumbelirleyici</i><br>2. $SS \rightarrow S\dot{I}$ : <i>mail</i> |
|--|

Altprotokolün birinci adımında:  $S\dot{I}$ , mesajlaşma sunucusunun *konumbelirleyicisini* gönderir. İkinci adımda: mail daha önceden kaydedildiyse;  $SS$ , servis istemcisi kullanıcısının *maili* ile cevap verir.

*Sağlayıcı-istemci* altprotokolünün; *kayıt*, *sorgu* ve *kontrol* altprotokolleri tek bir protokol gibi davranırlar. *Sağlayıcı-istemci* altprotokolünün çalışma sırası aşağıdaki gibidir (Shmatikov ve Mitchell, 2000; Drielsma ve Mödersheim, 2003; Onieva, Zhou ve Lopez, 2005; Ogata ve Futatsugi, 2003; Carbone, Honda ve Yoshida, 2007; Kovan, 2005; Buttyan ve Hubaux; Rubin ve Honeyman, 1993; Islam ve Zaid, 2008; Ratnasamy, Francis, Handley, Karp ve Shenker, 2001; Rowstron ve Druschel, 2001; Stoica, Morris, Karger, Kaashoek ve Balakrishnan, 2001; Zhao, Kubiatawicz ve Joseph, 2001):

*Kontrol*<sub>1</sub>.  $S\dot{I} \rightarrow SS$  : *konumbelirleyici*

*Kontrol*<sub>2</sub>.  $SS \rightarrow S\dot{I}$  : *mail*

---

IF *savebuttonpressed* THEN

*Kayıt*<sub>1</sub>.  $S\dot{I} \rightarrow SS$  : *cinsiyet, boy, mail, konumbelirleyici, testkonumbelirleyici*

*Kayıt*<sub>2</sub>.  $SS \rightarrow S\dot{I}$  : *sonuç*

---

IF *querybuttonpressed* THEN

*Sorgu*<sub>1</sub>.  $S\dot{I} \rightarrow SS$  : *cinsiyet, boy*

*Sorgu*<sub>2</sub>.  $SS \rightarrow S\dot{I}$  : *mailsayısı, mail(ler), konumbelirleyici(ler)*

---

$S\dot{I}_1$ ,  $S\dot{I}_2$  ve  $S\dot{I}_3$ 'ün 3 taraf olduğunu varsayalım.  $S\dot{I}_3$ ,  $S\dot{I}_1$  ile anlık mesajlaşmak istemektedir. Üç servis istemcisinin olduğu senaryoda;  $S\dot{I}_1$ ,  $S\dot{I}_2$  ile anlık mesajlaşmaktadır. *İstemci-istemci* altprotokolü aşağıdaki gibidir (Shmatikov ve Mitchell, 2000; Drielsma ve Mödersheim, 2003; Onieva, Zhou ve Lopez, 2005; Ogata ve Futatsugi, 2003; Carbone, Honda ve Yoshida, 2007; Kovan, 2005; Buttyan ve Hubaux; Rubin ve Honeyman, 1993; Islam ve Zaid, 2008; Ratnasamy, Francis, Handley, Karp ve Shenker, 2001; Rowstron ve Druschel, 2001; Stoica, Morris, Karger, Kaashoek ve Balakrishnan, 2001; Zhao, Kubiatawicz ve Joseph, 2001):

Ana Program	Thread
	1. $S\dot{I}_3 \rightarrow S\dot{I}_1$ : <i>konumbelirleyici</i> 2. $S\dot{I}_1 \rightarrow S\dot{I}_3$ : <i>istemcisayısı</i> 3. IF <i>participants=2</i> THEN 4.     IF <i>clientcount=2</i> THEN $S\dot{I}_1 \rightarrow S\dot{I}_3$ : <i>konumbelirleyiciler</i> 5.                                    ELSE $S\dot{I}_1 \rightarrow S\dot{I}_2$ : <i>konumbelirleyici</i> 6. $S\dot{I}_1 \rightarrow S\dot{I}_3$ : <i>Btadi</i> 7. $S\dot{I}_3 \rightarrow S\dot{I}_1$ : <i>Btadi</i>
1. IF <i>sendbuttonpressed</i> THEN 2. $S\dot{I}_1 \rightarrow S\dot{I}_3$ : <i>mesaj<sub>1</sub></i> 3. IF <i>participants=2</i> THEN $S\dot{I}_1 \rightarrow S\dot{I}_2$ : <i>mesaj<sub>1</sub></i>	8. WHILE <i>isRunning</i> DO 9. $S\dot{I}_3 \rightarrow S\dot{I}_1$ : <i>mesaj<sub>2</sub></i> 10. IF <i>participants=2</i> THEN $S\dot{I}_2 \rightarrow S\dot{I}_1$ : <i>mesaj<sub>3</sub></i>

Ana programın ilk adımında: *send* düğmesine basılıp basılmadığı kontrol edilir. Ana programın ikinci adımında: *send* düğmesine basıldıysa;  $S\dot{I}_1$ ,  $S\dot{I}_3$ 'e anlık mesaj (*mesaj<sub>1</sub>*) gönderir. Ana programın üçüncü adımında: iki katılımcı varsa;  $S\dot{I}_1$ ,  $S\dot{I}_2$ 'e anlık mesaj (*mesaj<sub>1</sub>*) gönderir.

*Thread*'in ilk adımında:  $S\dot{I}_3$ ,  $S\dot{I}_1$ 'e mesajlaşma sunucusunun *konumbelirleyicisini* gönderir. *Thread*'in ikinci adımında:  $S\dot{I}_1$ , istemci(lerin) sayısı (*istemicisayısı*) ile cevap verir. *Thread*'in üçüncü adımında: iki katılımcı olup olmadığını kontrol edilir. *Thread*'in dördüncü adımında: iki istemci varsa;  $S\dot{I}_1$ ,  $S\dot{I}_3$ 'e *konumbelirleyicileri* gönderir. *Thread*'in beşinci adımında: bir istemci varsa;  $S\dot{I}_1$ ,  $S\dot{I}_2$ 'e *konumbelirleyiciyi* gönderir. *Thread*'in altıncı adımında:  $S\dot{I}_1$ ,  $S\dot{I}_3$ 'e bluetooth ismini (*BTadı*) gönderir. *Thread*'in yedinci adımında:  $S\dot{I}_3$ , bluetooth ismiyle (*BTadı*) cevap verir. *Thread*'in sekizinci adımında: *Thread* çalıştığı sürece devam edecek bir döngü başlatılır. *Thread*'in dokuzuncu adımında:  $S\dot{I}_3$ ,  $S\dot{I}_1$ 'e anlık mesaj (*mesaj<sub>2</sub>*) gönderir. *Thread*'in onuncu adımında: iki katılımcı varsa;  $S\dot{I}_2$ ,  $S\dot{I}_1$ 'e anlık mesaj (*mesaj<sub>3</sub>*) gönderir.

Sağlayıcı-istemci ve istemci-istemci altprotokolleri tek bir protokol gibi davranır. Bu iki altprotokolün çalışma sırası aşağıdaki gibidir (Shmatikov ve Mitchell, 2000; Drielsma ve Mödersheim, 2003; Onieva, Zhou ve Lopez, 2005; Ogata ve Futatsugi, 2003; Carbone, Honda ve Yoshida, 2007; Kovan, 2005; Buttyan ve Hubaux; Rubin ve Honeyman, 1993; Islam ve Zaid, 2008; Ratnasamy, Francis, Handley, Karp ve Shenker, 2001; Rowstron ve Druschel, 2001; Stoica, Morris, Karger, Kaashoek ve Balakrishnan, 2001; Zhao, Kubiatoiwicz ve Joseph, 2001):

*Sağlayıcı-istemci.kontrol*

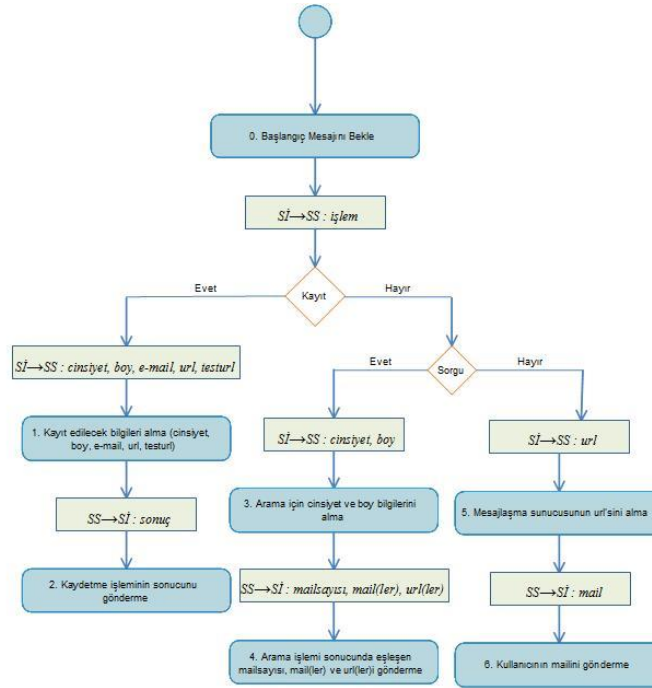
IF *savebuttonpressed* THEN *Sağlayıcı-istemci.kayıt*

IF *querybuttonpressed* THEN *Sağlayıcı-istemci.sorgu*

IF *connectbuttonpressed* THEN *İstemci-istemci*

#### 4. Örnek Modelleme: Servis Sağlayıcı Rolü

Katılımcı rollerinin modellenmesine örnek olarak servis sağlayıcı rolünün modellenmesi verilebilir. Servis sağlayıcısının (*SS*) durum diyagramı şekil 2'de gösterilmiştir.



Şekil 2. Servis Sağlayıcısının Durum Diyagramı

İlk olarak *SS*, *Sİ*’den mesaj gelene kadar bekler. Yapılacak işlem *Sİ*’den alındıktan sonra; işlem kayıt işlemiyse, kayıt edilecek bilgiler (*cinsiyet*, *boy*, *e-mail*, konumbelirleyici (*url*), testkonumbelirleyici (*testurl*)) *Sİ*’den alınır. *Sİ*’ye cevap olarak kaydetme işleminin sonucu (*sonuç*) gönderilir. Yapılacak işlem sorgu işlemiyse, *cinsiyet* ve *boy* bilgileri *Sİ*’den alınır. Daha sonra *Sİ*’ye arama işlemi sonucunda eşleşen *mailsayısı*, *mail(ler)* ve konumbelirleyici(ler) (*url(ler)*) gönderilir. Yapılacak işlem kontrol işlemiyse, mesajlaşma sunucusunun konumbelirleyicisi (*url*) *Sİ*’den alınır. Mail daha önceden kaydedildiyse, *Sİ*’ye servis istemcisi kullanıcısının *maili* gönderilir.

## 5. Tartışma ve Sonuçlar

Napster gibi merkezleştirilmiş mimariyi uyarlayan protokollerde meta-bilgiyi elde etme için merkezi sunucu kullanılmakta, diğer tüm haberleşme peer’ler arasında yapılmaktadır (Bustamante ve Qiao, 2004). Merkezleştirilmemiş gevşek yapıli protokollerde nesnelere yerinin kontrolü değışkenlik gösterir (Clarke, Sandberg, Wiley ve Hong, 2000) ya da peer’lere belirli derecede hiyerarşı yüklenebilir (Kazaa, 2001;Klingberg ve Manfredi, 2002). Yüksek yapılandırılmış protokollerde, ağ topolojisi ve kaynakların yeri kesin olarak belirlenmiştir (Ratsanamy, Francis, ;Handley, Karp ve Shenker, 2001; Rowstron ve Druschel, 2001; Stoica, Morris, Karger, Kaashoek ve Balakrishnan, 2001; Zhao, Kubiatoicz ve Joseph, 2001). Gnutella’nın eski versiyonları gibi merkezleştirilmemiş ve yapılandırılmamış protokollerde, merkezi dizinler ve ağ topolojisi ya da nesne yeri üzerinde kesin kontrol yoktur (Clip2, 2000).

Servis tabanlı yazılım geliştirme paradigmasında; servis sağlayıcılar, kaynaklarını ve yeteneklerini istemcilerin kullanımına sunarlar. İstemciler, kaynaklarının azaldığı veya yetmediği durumlarla karşılaşabilir. Kaynak azalması veya yetmemesi durumuyla başa çıkabilmek için servis tabanlı yazılım geliştirme kullanılabilir.

Bu çalışmada, mobil peer-to-peer ağlar ve servis tabanlı yazılım geliştirme incelenip; mobil peer-to-peer ağlarda bluetooth üzerinden anlık mesajlaşma protokolü olan IMOB geliştirilmiştir. IMOB protokolü, yüksek yapılandırılmış protokollerin kaynak yerinin bilinme özelliğini alıp; buna servis tabanlı yazılım geliştirmenin kayıtçı rolünü eklemiştir. Bu protokol tanımlanıp, örnek olarak servis sağlayıcı rolü modellenmiştir.

## 6. Kaynaklar

- M. P. Singh (2001) Pering at peer-to-peer computing. *IEEE Internet Computing*, 05(1):4–5.
- G. Gehlen, and L. Pham (2005) Mobile Web Services for Peer-to-Peer Applications. Consumer Communications and Networking Conference.
- G. Kortuem, J. Schneider, D. P. Thaddeus, G. C. Thompson, S. Fickas, and Z. Segall (2001) When Peer-toPeer comes Face-to-Face: Collaborative Peer-toPeer Computing in Mobile Ad hoc Networks. First International Conference on Peer-to-Peer Computing, Linköping, Sweden, 27-29 August 2001.
- N. Maibaum, and T. Mundt (2002) JXTA: A Technology Facilitating Mobile Peer-To-Peer Networks. International Mobility and Wireless Access Workshop (MobiWac’02), Fort Worth, Texas, USA, pp. 7-13, 12 October 2002.
- B. A. Miller, and C. Bisdikian (2002) Bluetooth Revealed, Addison-Wesley, 2 edition.
- M. Wiberg, and A. Grönlund (2000) Exploring Mobile CSCW: Five areas of questions for further research. Proceedings of IRIS23 (Information Research in Scandinavia), Trollhattan, Sweden.
- A. I. Wang, T. Björnsgård, and K. Saxlund (2007) Peer2Me - Rapid Application Framework for Mobile Peer-to-Peer Applications. 2007 International Symposium on Collaborative Technologies and Systems (CTS 2007), Orlando, Florida, USA, page 10, May 21-25, 2007.
- Wikipedia (2018). <https://en.wikipedia.org/>. Erişim tarihi 07 Haziran 2018

- V. Shmatikov and J. C. Mitchell (2000) Analysis of a Fair Exchange Protocol, <http://www.isoc.org/isoc/conferences/ndss/2000/proceedings/001.pdf>, Eriřim tarihi 07 Haziran 2018.
- P. H. Drielsma and S. Mödersheim (2003) The ASW Protocol Revisited: A Unified View, <http://www.avispa-project.org/papers/asw-arspa04-entcs05.pdf>, Eriřim tarihi 07 Haziran 2018.
- J. A. Onieva, J. Zhou and J. Lopez (2005) Analysis of an Asynchronous Multi-Party Contract Signing Protocol, <http://jerry.c-lab.de/ubisec/publications/indocrypt05.pdf>, Eriřim tarihi 15 Mayıs 2010.
- K. Ogata and K. Futatsugi (2003) Formal Analysis of the NetBill Electronic Commerce Protocol, <http://www.yonezaki.cs.titech.ac.jp/Workshop/iss2003/slides/Ogata.pdf>, Eriřim tarihi 15 Mayıs 2010.
- M. Carbone, K. Honda and N. Yoshida (2007) Structured Communication-Centred Programming for Web Services, <http://www.dcs.qmul.ac.uk/~carbonem/cdlpaper/submission.pdf>, Eriřim tarihi 15 Mayıs 2010.
- G. Kovan (2005) SPP Secure Payment Protocol: Protocol Analysis, Implementation and Extensions, <http://etd.uwaterloo.ca/etd/gkovan2005.pdf>, Eriřim tarihi 07 Haziran 2018.
- L. Buttyan and J. P. Hubaux, Rational Exchange, [www.hit.bme.hu/~buttyan/courses/BMEVIHI9367/June12\\_rex.ppt](http://www.hit.bme.hu/~buttyan/courses/BMEVIHI9367/June12_rex.ppt), Eriřim tarihi 07 Haziran 2018.
- A. D. Rubin and P. Honeyman (1993) Formal Methods for the Analysis of Authentication Protocols, <http://www.citi.umich.edu/techreports/reports/citi-tr-93-7.pdf>, Eriřim tarihi 07 Haziran 2018.
- S. Islam and M. A. Zaid (2008) Probabilistic Analysis and Verification of the ASW Protocol using PRISM, <http://ijns.femto.com.tw/contents/ijns-v7-n3/ijns-2008-v7-n3-p388-396.pdf>, Eriřim tarihi 07 Haziran 2018.
- S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker., (2001) A scalable content-addressable network. In Proc. of SIGCOMM, San Diego, CA, USA, pp. 161–172, August 2001.
- A. Rowstron and P. Druschel. (2001) Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In Proc. of the IFIP/ACM Middleware, Heidelberg, Germany, November 2001.
- I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan (2001) Chord: A scalable peer-to-peer lookup service for internet applications. In Proc. of SIGCOMM, San Diego, CA, USA, pp. 149–160, August 2001.
- B. Y. Zhao, J. Kubiatowicz, and A. D. Joseph (2001) Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCV/CSD-01-1141, Computer Science Division, University of California, Berkeley, CA, USA, April 2001.
- F. E. Bustamante and Y. Qiao (2004) Friendships that Last: Peer Lifespan and its Role in P2P Protocols, In: Douglis F., Davison B.D. (eds) Web Content Caching and Distribution. Springer, Dordrecht.
- I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong (2000) Freenet: A distributed anonymous information storage and retrieval system. In Proc. ICSI Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, USA, pp. 43–58, July 2000.
- Kazaa (2001). <http://www.kazaa.com>. Eriřim tarihi 15 Haziran 2012
- T. Klingberg and R. Manfredi (2002) Gnutella 0.6. RFC, The Gnutella RFC, June 2002.
- Clip2 (2000), The Gnutella protocol specification v0.4. RFC, The Gnutella RFC.