T.C.

TRAKYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Mobil ve Kablosuz Dağıtık Sistemlerde Bilgi Erişilebilirliği
Yüksek Sistemlerin Geliştirilmesi ve Performans Araştırması

DOKTORA TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

Bu tez 02/07/2007 tarihinde Aşağıdaki Jüri Tarafından Kabul Edilmiştir.

Yrd.Doç.Dr. Erdem UÇAR
(Danışman)

Yrd.Doç.Dr.Yılmaz KILIÇASLAN
(Üye)

Yrd.Doç.Dr. Rafet AKDENİZ
(Üye)

Yrd.Doç.Dr. Özlem UÇAR
(Üye)

Yrd.Doç.Dr. Yılmaz ÇAN
(Üye)

# ÖZET

# MOBİL VE KABLOSUZ DAĞITIK SİSTEMLERDE BİLGİ ERİŞİLEBİLİRLİĞİ YÜKSEK SİSTEMLERİN GELİŞTİRİLMESİ VE PERFORMANS ARAŞTIRMASI

Cenk ATLIĞ
Bilgisayar Mühendisliği Doktora Tezi

İstemci/sunucu türü bilgi erişim mekanizması, bilgisayar ağlarında yıllardan beri yoğun olarak kullanılmakta olan bir yöntemdir. Eşdüzey ismi ile anılan yeni bir iletişim yönteminin keşfinden sonra bu yeni yöntemin kullanılması hızlı bir şekilde artmıştır. Popüler olan bu eşdüzey sistemler, birçok projede çeşitli ortamlar ve farklı amaçlar için uygulanmıştır ve halen geliştirilmeye devam etmektedir. Bilgisayar biliminde son yıllarda popüler olan diğer bir alan da mobil ve kablosuz sistemler olup bu alandaki gelişmeler hızlı bir şekilde devam etmektedir. Yaptığımız bu çalışmada, şu ana kadar var olan bilgi erişim teknikleri incelenmiştir. Ayrıca, mevcut olan mobil ve kablosuz sistemlerin özelliklerinden bahsedilip bu sistemler hakkında bilgi verilmiştir. Bahsedilen sistemler ajan tabanlı ve gevşek bağlı model kullanan eşdüzey sistemi ile daha verimli yapılabilmektedir. Bir çeşit JavaSpaces versiyonu olan GigaSpaces servisi bir Tuple Space uygulamasıdır. JavaSpaces servisinin dağıtık sistemlerdeki başarılı uygulama geliştirme özelliklerine ek nitelikler kazandıran GigaSpaces böyle bir sistemin tasarımı için çok uygundur.

Bu tez çalışmasında, bilgi erişilebilirliğinin dünyadaki yerini ve mobil ve kablosuz sistemlerin özelliklerini inceledik. Şu anda var olan bilgi erişimi ve mobil ve kablosuz sistemler üzerinde yaptığımız araştırma sonucunda, ajan tabanlı eşdüzey iletişim için gevşek bağlı sistemi önce tasarlayıp daha sonra bu sistemin uygulamasını kablosuz yerel bilgisayar ağında hayata geçirdik. Yaptığımız bu uygulamada GigaSpaces servisi kullanılmıştır.

Bu tezde yaptığımız çalışmalar, geliştirilen ajan tabanlı eşdüzey iletişim için gevşek bağlı sistemin kablosuz olarak eşdüzey servisinden yararlanmak isteyen cihazlara avantaj sağladığını göstermiştir. Tezin deneysel bölümünde, geliştirdiğimiz sistemdeki mobil ve kablosuz cihazlarda çalışmakta olan eşlerin mesaj trafiği ve enerji tüketimi üzerine testler yapılmıştır. Yapılan bu testler iki farklı kablosuz mimaride çalıştırılarak sistemlerin performansları gözlemlenmiş, çıkan sonuçlar üzerine değerlendirme yapılmıştır.

**Anahtar Kelimeler**: Bilgi erişimi, eşdüzey, kablosuz, ajan, Jini teknolojisi, JavaSpaces servisi, GigaSpaces servisi, Linda programlama dili, Tuple Space, mesaj trafiği testi, enerji tüketimi testi.

Temmuz-2007, 147 sayfa

# ABSTRACT

## DEVELOPMENT OF SYSTEMS WITH HIGH DATA ACCESSIBILITY IN MOBILE AND WIRELESS DISTRIBUTED SYSTEMS AND PERFORMANCE INVESTIGATION

Cenk ATLIĞ
Computer Engineering Ph.D. Thesis

The information access mechanism in the form of client/server has long been an intensively used method in computer networks. A new communication method called "peer-to-peer" has been increasingly used since its inception. These popular peer-to-peer systems have been applied to various environments in various projects and for various reasons and are still under development. There have been important developments in mobile and wireless systems, another popular area in computer science, in recent years. In this thesis, data access techniques are investigated. Additionally, information about mobile and wireless systems is given by stating their characteristics. These mentioned systems can be made more efficient via agent based peer-to-peer systems using loosely-coupled models. The GigaSpaces service is a kind of JavaSpaces version that is a Tuple Space application. GigaSpaces, which provides additional characteristics to successful application development characteristics in distributed systems of JavaSpaces service, is quiet suitable for the design of such systems.

In this thesis, the place of data access in world and characteristics of mobile and wireless systems are studied. After investigating data access and mobile and wireless systems of today, for agent based peer-to-peer communication, first, a loosely-coupled system was designed and the system was implemented on wireless local computer network. The GigaSpaces service was used in this implementation.

In this thesis, it was concluded that loosely-coupled systems provide devices which need to make use of a peer-to-peer service wirelessly with an advantage in agent based peer communication. In the experimental part of the thesis, tests were done on the message traffic and energy consumption of the peers operating on mobile and wireless devices we developed. These tests were operated in two different architectures and performances of the systems were observed, then results were evaluated.

**Key Words**: Data access, peer-to-peer, wireless, agent, Jini technology, JavaSpaces service, GigaSpaces service, Linda programming language, Tuple Space, message traffic test, energy consumption test.

July 2007, 147 pages

**TEŞEKKÜR**

Tez çalışmam süresince yaptığım çalışmalarda desteğini hiçbir zaman esirgemeyen danışmanım Yrd.Doç.Dr. Erdem UÇAR'a;

Bölümde konuşma ihtiyacı duyduğumda yaptığımız konuşmalar için Yrd.Doç.Dr. Özlem UÇAR'a hanıma;

Yaptığımız konuşmalar ile bana destek olan Yrd.Doç.Dr. Yılmaz KILIÇASLAN'a;

Başta merkez müdürü ve Rektör Yardımcısı Prof.Dr. Beyhan KARAMANLIOĞLU ve müdür yardımcısı Yrd.Doç.Dr. Nilgün TOSUN olmak üzere, Trakya Üniversitesi Uzaktan Eğitim Merkezindeki bütün çalışanlara;

Kendisini tanıdığım günden beri desteğinden dolayı Yrd.Doç.Dr. Kamil KAHVECİ'ye;

Bilimsel ve sosyal hayatıma büyük katkıları olan, Yüksek Lisans danışmanım Doç.Dr. Alexander E. KOSTIN'e;

Tanıdığım günden beri desteğini esirgemeyen büyüğüm Prof.Dr. Mehmet TOLUN'a;

Arkadaşlık desteğinden ötürü Arş.Gör. Erdinç UZUN ve Arş.Gör. Volkan AGUN'a;

Ofis arkadaşım Arş.Gör. Serkan GİLANLI'ya;

Hayatımın bu dönemine gelene kadar beraber çalıştığım ve aynı ortamda bulunduğum ancak şu anda burada ismini belirtmediğim bütün arkadaş ve dostlarıma;

Sevgileri ve destekleri ile her zaman yanımda olan, anneme, babama, kardeşime, halama ve anneanneme;

Sonsuz teşekkürlerimi sunarım.

# İÇİNDEKİLER

Sayfa

**ÖZET.....** i

**ABSTRACT……...** ii

**TEŞEKKÜR……...** iii

**İÇİNDEKİLER……..** iv

**TABLO LİSTESİ……...** ix

**ŞEKİL LİSTESİ……** x

**KISALTMA LİSTESİ……...** xiv

**1. GİRİŞ…….....** 1

**1.1** BİLGİ ERİŞİLEBİLİRLİĞİ YÜKSEK SİSTEMLER VE MOBİL VE KABLOSUZ SİSTEMLERDE UYGULANABİLİRLİĞİ..... 1

**1.2** TEZİN AMACI…… 3

**1.3** TEZİN KAPSAMI…… 3

**1.4** TEZİN TASLAĞI……..... 4

**2. BİLGİ ERİŞİLEBİLİRLİĞİ……** 5

Sayfa

**2.1** BİLGİ ERİŞİLEBİLİRLİĞİNİN BİLİŞİM DÜNYASINDAKİ YERİ..... 6

**2.2** İÇERİK PAYLAŞIMI..... 8

**2.3** BİLGİ ERİŞİMİ KONULARINDA GÜVENLİK VE MAHREMİYET..... 11

**2.4** BİLGİ ERİŞİLEBİLİRLİĞİNİN GENEL DURUM DEĞERLENDİRMESİ……. 12

**3. MOBİL VE KABLOSUZ SİSTEMLER…….** 16

**3.1** MOBİL VE KABLOSUZ SİSTEMLERİN GENEL ÖZELLİKLERİ..... 17

**3.2** MOBİL VE KABLOSUZ SİSTEMLERİN ÇEŞİTLERİ..... 21

**3.3** MOBİL VE KABLOSUZ DAĞITIK SİSTEMLERİN TEMEL SORUNLARI..... 23

**3.4** BİLGİ ERİŞİMİ YÜKSEK BİR SİSTEM UYGULAMASI İÇİN MOBİL VE KABLOSUZ SİSTEMLERİN KIYASLANMASI..... 25

**4. BİLGİ ERİŞİLEBİLİRLİĞİ YÜKSEK SİSTEMLERİN GELİŞTİRİLMESİNDE GEVŞEK BAĞLI MODEL KULLANAN AJAN TABANLI SİSTEM YAPISI…….** 27

**4.1** LINDA PROGRAMLAMA MODELİ…..……... 27

**4.2** JINI TEKNOLOJİSİ VE GENEL YAPISI…….. 31

**4.3** JAVASPACES SERVİSİ VE TEMEL ÖZELLİKLERİ…….. 34

**4.4** JAVASPACES SERVİSİNDE YAPILAN OPERASYONLARIN ANLAMBİLİMSEL YAPISI……..... 38

**4.5** AJAN TABANLI SİSTEMLER VE GEVŞEK BAĞLI MODEL İLE UYGULANMASI..... 43

Sayfa

**5. MOBİL VE KABLOSUZ SİSTEMLERDE AJAN TABANLI VE YÜKSEK PERFORMANSLI EŞDÜZEY İLETİŞİM İÇİN GEVŞEK BAĞLI MODEL KULLANAN SİSTEMİN DİZAYNI VE UYGULAMASI…….** 45

**5.1** VARSAYIMLAR..... 45

**5.2** MOBİL VE KABLOSUZ SİSTEMLERDE AJAN TABANLI VE YÜKSEK PERFORMANSLI EŞDÜZEY İLETİŞİM İÇİN GEVŞEK BAĞLI MODEL KULLANAN SİSTEMİN MODELİ……... 46

**5.3** ÖNERİLEN SİSTEMİN ANLAMBİLİMSEL YAPISI……... 49

**5.3.1** Sisteme Dahil Olma İşleminin Anlambilimsel Yapısı..……..... 49

**5.3.2** Sistemden Ayrılma İşleminin Anlambilimsel Yapısı..... 50

**5.3.3** Bilgi Arama İsteği İşleminin Anlambilimsel Yapısı..... 53

**5.3.4** Sisteme Bilgi Ekleme İşleminin Anlambilimsel Yapısı..... 57

**5.3.5** Sistemden Bilgi İndirme İşleminin Anlambilimsel Yapısı..... 60

**6. DENEYSEL ÇALIŞMALAR…….....** 64

**6.1** DENEYLERİN ORGANİZASYONU……. 64

**6.2** DENEYSEL SİSTEMİN ÖZELLİKLERİ..... 68

**6.2.1** GigaSpaces Servisinde Read İşlemi Yapılması Durumundaki Sistem Davranışı..... 68

**6.2.2** GigaSpaces Servisinde Write İşlemi Yapılması Durumundaki Sistem Davranışı..... 70

**6.2.3** GigaSpaces Servisinde Önce Write Arkasından Read İşlemlerinin Yapılması Durumundaki Sistem Davranışı..... 72

Sayfa

**6.3** DENEYSEL ÇALIŞMALARDA YAPILAN İNCELEMELER…. 74

**6.3.1** Aynı Salondaki Bilgisayar Sistemleri için Sistem Performansı….. 74

**6.3.1.1** Aynı Salondaki Bilgisayar Sistemlerinde Embeded Türü GigaSpaces Servisi Kullanan Sistem……. 74

**6.3.1.2** Aynı Salondaki Bilgisayar Sistemlerinde gsServer Türü GigaSpaces Servisi Kullanan Sistem……. 80

**6.4.1** Uzak Bilgisayar Sistemleri için Sistem Performansı……... 86

**6.4.1.1** Uzak Bilgisayar Sistemlerinde Embeded Türü GigaSpaces Servisi Kullanan Sistem……... 86

**6.4.1.2** Uzak Bilgisayar Sistemlerinde gsServer Türü GigaSpaces Servisi Kullanan Sistem……... 91

**6.5.1** Trafik İncelemesi……. 98

**6.5.1.1** Aynı Salondaki Bilgisayar Sistemleri için Trafik İncelemesi….. 99

**6.5.1.2** Uzak Bilgisayar Sistemleri için Trafik İncelemesi…….. 100

**6.5.2** Enerji İncelemesi…….. 103

**6.5.2.1** Aynı Salondaki Bilgisayar Sistemleri için Enerji İncelemesi…... 103

**6.5.2.2** Uzak Bilgisayar Sistemleri için Enerji İncelemesi…..... 105

**7. SONUÇLAR……...** 108

**7.1** YAPILANLARIN ÖZETİ…… 108

**7.2** ARAŞTIRMANIN KATKISI……... 109

**7.3** KISITLAR…… 110

**7.4** GELECEK ÇALIŞMALAR……. 110

**8.TARTIŞMA……** 112

**KAYNAKLAR……...** 116

**ÖZGEÇMİŞ……** 129

**EKLER……** 130

# TABLO LİSTESİ


| | | Sayfa |
|---|---|---|
| Tablo 2.1 | Bilgi erişilebilirliği yüksek sistemlerin potansiyel avantajları……. | 6 |
| Tablo 2.2 | Bilgi erişilebilirliği yüksek sistemlerin yaratabileceği problemler……. | 7 |
| Tablo 3.1 | Mobil ve kablosuz sistemlerin potansiyel avantajları…….. | 19 |
| Tablo 3.2 | Mobil ve kablosuz sistemlerin yaratabileceği problemler…… | 20 |
| Tablo 3.3 | Mobil ve kablosuz teknolojilerin çeşitleri ve özellikleri…….. | 23 |
| Tablo 4.1 | Linda programlama dilinde tuple space üzerinde yapılan işlemler..... | 29 |
| Tablo 4.2 | Linda programlama dilinin avantajları……. | 30 |
| Tablo 4.3 | Jini teknolojisinin ağ uygulamalarına getirdiği avantajlar……... | 32 |
| Tablo 6.1 | Deneylerde kullanılan bilgisayar sistemlerinin özellikleri……... | 64 |

# ŞEKİL LİSTESİ


| | | Sayfa |
|---|---|---|
| Şekil 2.1 | Bilginin tek noktada olması yerine (sol taraf), birden fazla noktada olması (sağ taraf), sisteme güç kazandırır…… | 5 |
| Şekil 2.2 | Bilgi merkezi bir noktada saklanmakta ve istemler buradan cevaplanmakta (İstemci/Sunucu {Client/Server} yöntemi bilgi erişimi)……... | 8 |
| Şekil 2.3 | Bilgilerin bulunduğu yerler ve özellikleri bir noktada saklanmakta, bilgi alışverişi eşdüzey kabul edilen işlemler arasında direkt yapılmakta……. | 9 |
| Şekil 2.4 | Bilgi alışverişi eşdüzey kabul edilen işlemler arasında direkt yapılmaktadır……. | 10 |
| Şekil 4.1 | Örnek üç tuple space ile içerisinde bulunan 7 tuple……..... | 28 |
| Şekil 4.2 | Jini teknolojisinin Java Platformundaki yeri ve JavaSpaces servisinin bu mimarideki konumu……. | 33 |
| Şekil 4.3 | Dört işlemin beş JavaSpaces servisini kullanımı. İşlemler farklı JavaSpaces servislerini uygun komutlar aracılığı ile kullanıyorlar…….. | 36 |
| Şekil 4.4 | JavaSpaces servisinde *write* komutunun çalışma şekli……..... | 39 |
| Şekil 4.5 | JavaSpaces servisinde *read* komutunun çalışma şekli……. | 40 |
| Şekil 4.6 | JavaSpaces servisinde *take* komutunun çalışma şekli…….. | 41 |
| Şekil 5.1 | Mobil ve kablosuz sistemler için önerilen ajan tabanlı ve yüksek performanslı eşdüzey iletişim için gevşek-bağlı model kullanan sistem mimarisi……..... | 48 |
| Şekil 5.2 | Sisteme dahil olma işleminin çalışma prensibi……..... | 50 |
| Şekil 5.3 | Sisteme dahil olma işlemi programının ana hatları……..... | 51 |
| Şekil 5.4 | Sistemden ayrılma işleminin çalışma prensibi | 52 |
| Şekil 5.5 | Sistemden ayrılma işlemi programının ana hatları……... | 53 |
| Şekil 5.6 | Bilgi arama isteğinin çalışma şekli……... | 55 |
| Şekil 5.7 | Bilgi arama isteği programının ana hatları……... | 57 |
| Şekil 5.8 | Sisteme bilgi ekleme işleminin çalışma şekli……... | 58 |
| Şekil 5.9 | Sisteme bilgi ekleme işlemi programının ana hatları……... | 59 |

Sayfa


| | | |
|---|---|---|
| Şekil 5.10 | Sistemden bilgi indirme işleminin çalışma şekli…….. | 61 |
| Şekil 5.11 | Sistemden bilgi indirme işlemi programının ana hatları…….. | 63 |
| Şekil 6.1 | Birinci tip deneylerde kullanılan bilgisayarların organizasyonu……... | 66 |
| Şekil 6.2 | İkinci tip deneylerde kullanılan bilgisayarların organizasyonu…….. | 67 |
| Şekil 6.3 | GigaSpaces servisinin üzerinde kurulu JavaSpaces servisinin deneylerde kullandığımız bilgisayar sistemlerinde tek thread kullanılarak yapılan *READ* işlemi için gösterdiği performans yapısı…… | 68 |
| Şekil 6.4 | GigaSpaces servisinin üzerinde kurulu JavaSpaces servisinin deneylerde kullandığımız bilgisayar sistemlerinde 10 thread kullanılarak yapılan *READ* işlemi için gösterdiği performans özellikleri……... | 69 |
| Şekil 6.5 | GigaSpaces servisinin üzerinde kurulu JavaSpaces servisinin deneylerde kullandığımız bilgisayar sistemlerinde tek thread kullanılarak yapılan *WRITE* işlemi için gösterdiği performans özellikleri……... | 70 |
| Şekil 6.6 | GigaSpaces servisinin üzerinde kurulu JavaSpaces servisinin deneylerde kullandığımız bilgisayar sistemlerinde 10 thread kullanılarak yapılan *WRITE* işlemi için gösterdiği performans özellikleri……... | 71 |
| Şekil 6.7 | GigaSpaces servisinin üzerinde kurulu JavaSpaces servisinin deneylerde kullandığımız bilgisayar sistemlerinde tek thread kullanılarak yapılan önce *WRITE* ve sonrasında *READ* işlemi için gösterdiği performans özellikleri……..... | 72 |
| Şekil 6.8 | GigaSpaces servisinin üzerinde kurulu JavaSpaces servisinin deneylerde kullandığımız bilgisayar sistemlerinde 10 thread kullanılarak yapılan önce *WRITE* ve sonrasında *READ* işlemi için gösterdiği performans özellikleri…….. | 73 |
| Şekil 6.9 | 1 eşli sistemde, "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman…….. | 75 |
| Şekil 6.10 | 1 eşli sistemde, "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman……. | 76 |
| Şekil 6.11 | 5 eşli sistemde, "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman……. | 77 |
| Şekil 6.12 | 5 eşli sistemde, "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman…… | 78 |

Sayfa


| | | |
|---|---|---|
| Şekil 6.13 | 10 eşli sistemde, "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman…….. | 79 |
| Şekil 6.14 | 10 eşli sistemde, "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman……. | 80 |
| Şekil 6.15 | 1 eşli sistemde "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman…….. | 81 |
| Şekil 6.16 | 1 eşli sistemde, "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman…… | 82 |
| Şekil 6.17 | 5 eşli sistemde "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman……. | 82 |
| Şekil 6.18 | 5 eşli sistemde, "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman…… | 83 |
| Şekil 6.19 | 10 eşli sistemde "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman……. | 84 |
| Şekil 6.20 | 10 eşli sistemde, "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman……..... | 85 |
| Şekil 6.21 | Uzak bilgisayar sistemleri için 1 eşli sistemde "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman……... | 87 |
| Şekil 6.22 | Uzak bilgisayar sistemleri için 1 eşli sistemde "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman…….. | 88 |
| Şekil 6.23 | Uzak bilgisayar sistemleri için 5 eşli sistemde "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman……... | 88 |
| Şekil 6.24 | Uzak bilgisayar sistemleri için 5 eşli sistemde "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman…….. | 89 |
| Şekil 6.25 | Uzak bilgisayar sistemleri için 10 eşli sistemde "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman……... | 90 |
| Şekil 6.26 | Uzak bilgisayar sistemleri için 10 eşli sistemde "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman…….. | 91 |
| Şekil 6.27 | Uzak bilgisayar sistemleri için 1 eşli sistemde "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey | 92 |

Sayfa

sistemden alınmak istenen bilgi için geçen zaman……...

Şekil 6.28 Uzak bilgisayar sistemleri için 1 eşli sistemde "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman…….. 93

Şekil 6.29 Uzak bilgisayar sistemleri için 5 eşli sistemde "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman……..... 94

Şekil 6.30 Uzak bilgisayar sistemleri için 5 eşli sistemde "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman…….. 95

Şekil 6.31 Uzak bilgisayar sistemleri için 10 eşli sistemde "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman……..... 96

Şekil 6.32 Uzak bilgisayar sistemleri için 10 eşli sistemde "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman…….. 97

Şekil 6.33 Aynı salondaki bilgisayar sistemleri için "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan mesaj trafiği.... 99

Şekil 6.34 Aynı salondaki bilgisayar sistemleri için "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan mesaj trafiği.... 100

Şekil 6.35 Uzak bilgisayar sistemleri için "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan mesaj trafiği……..... 101

Şekil 6.36 Uzak bilgisayar sistemleri için "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan mesaj trafiği……... 102

Şekil 6.37 Aynı salondaki bilgisayar sistemleri için "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan enerji tüketimi. 104

Şekil 6.38 Aynı salondaki bilgisayar sistemleri için "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan enerji tüketimi. 105

Şekil 6.39 Uzak bilgisayar sistemleri için "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan enerji tüketimi…… 106

Şekil 6.40 Uzak bilgisayar sistemleri için "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan enerji tüketimi……..... 107

**KISALTMA LİSTESİ**

| | | |
|---|---|---|
| ACM | : | Association for Computing Machinery |
| AMPS | : | Advanced Mobile Phone System |
| AP | : | Access Point |
| BAN | : | Body Area Network |
| BS | : | Base Station |
| DECT | : | Digital Enhanced Cordless Telephone |
| DHL | : | Distributed Hash Table |
| FCC | : | Federal Communication Commission |
| FIFO | : | First In First Out |
| GPRS | : | General Packet Radio Service |
| GPS | : | Global Positioning System |
| GSM | : | Global System for Mobile communications<br>ya da<br>Group Special Mobile |
| HIPERLAN | : | HIgh PERformance Local Area Network |
| IEEE | : | Institute of Electrical and Electronics Engineers |
| ISM | : | Industrial, Scientific, and Medical |
| ITU | : | International Telecommunication Union |
| MAC | : | Medium Access Control |
| MANET | : | Mobile Ad Hoc NETwork |
| MS | : | Mobile Station |
| NIC | : | Network Interface Card |
| OFDM | : | Orthogonal Frequency Division Multiplexing |
| P2P | : | Peer-to-Peer |
| PAN | : | Personal Area Network |
| PDA | : | Portable Digital Assistant<br>ya da<br>Personal Digital Assistant |
| QoS | : | Quality of Service |
| RF | : | Radio Frequency |
| SAR | : | Specific Absorption Rate |
| UWB | : | Ultra Wide Bandwidth system |
| VoIP | : | Voice over Internet Protocol |
| WEP | : | Wired Equivalency Privacy |
| WiBro | : | WIreless BROadband |
| WiFi | : | Wireless Fidelity |
| WiMAX | : | World interoperability for Microwave ACCess |
| WLAN | : | Wireless Local Area Network |
| WMAN | : | Wireless Metropolitan Area Networks |
| WMN | : | Wireless Mesh Networks |
| WPA | : | **W**i-Fi **P**rotected **A**ccess |
| WPA2 | : | **W**i-Fi **P**rotected **A**ccess 2 |
| WPAN | : | Wireless Personal Area Network |

# 1. GİRİŞ

Bilgisayar sistemleri, insan hayatına girmiş olduğu günden bu yana çok önemli mesafeler katetmiştir. Bu tezde bilgisayar sistemlerinde hayati bir yeri olan bilgi erişilebilirliği konusu, ve günümüzün popüler araştırma ve uygulama alanlarından mobil ve kablosuz sistemler üzerinde çalışılmıştır.

## 1.1 Bilgi Erişilebilirliği Yüksek Sistemlerin Mobil ve Kablosuz Sistemlerde Uygulanabilirliği

Bilgi erişilebilirliği son derece hayati önem arz eden bir konu haline gelmiştir. Bilginin değeri bilgisayar sistemlerinin kullanılması ile üstel bir şekilde artmaktadır. Bu bağlamda bilgiyi bulmak, korumak ve paylaşmak gibi konular ciddi önem kazanmıştır [Subramanian, R., v.d.].

Ortaya çıkan bu önem sonucunda çeşitli projeler ile bilginin erişilebilirliği, paylaşımı gibi konularda birçok proje ve çalışma yürütülmüştür. Bilgi erişilebilirliği, genel olarak üç ana sınıfta incelenmektedir [Kant, K., v.d.]. Bu yöntemlerden ilki istemci/sunucu (client/server) modelidir. İstemci/sunucu modeli bilgi erişiminde, bilgi merkezi bir noktada tutulmakta, gelen istekler merkezi nokta tarafından cevaplandırılmaktadır. Bu yöntem uzun süredir kullanılan bir yöntem olmakla birlikte, her uygulama için uygun değildir [JADE]. Örneğin, dosya paylaşımı için bu sistemin kullanılması sistemin işleyişinde problemler oluşturmaktadır.

İstemci/sunucu modelindeki bilgi erişiminin dezavantajlarını ortadan kaldırmak için değişik erişim yöntemleri geliştirilmiştir. Bu metotlardan ilki merkezi-indeks metodudur. Merkezi-indeks metodunda bilgiye erişmek isteyen bilgisayardan gelen istekler merkezi olarak kabul edilen sistemle iletişime geçer. Merkezi sistemdeki bu bilgiler incelendikten sonra istenen bilginin nerelerde ne şartlarla bulunduğu bilgiye erişmek istenen bilgisayar sistemine iletilir. Bu aşamadan sonra iletişim, bilgiye sahip olan bilgisayar ile bilgiye erişmek isteyen bilgisayar arasında gerçekleşir. Bu sistemin ismine melez-eşdüzey (Hybrid Peer-to-Peer) denilmektedir. Bu yöntemde ortada bir indeks bulunduğundan, Merkezi İndeks (Central Index) adıyla da anılmaktadır.

Merkezi-indeks modelinde sistemin tek bir nokta üzerinden geçiş yapması ve ancak ilk olarak bu merkezi indeksten gelen bilgilerden sonra bilgisayarlar arası direkt bağlantı kurulmasından dolayı daha başarılı performans özellikleri gösteren bir model bulmak üzere çalışmalar başlamıştır. Yapılan bu çalışmalar sonucunda, Tam Eşdüzey (Pure Peer-to-Peer) sistemi geliştirilmiştir. Tam Eşdüzey sisteminde bilgi alışverişi, bilgiye ulaşmak isteyen bilgisayar ile bilgiyi gönderecek bilgisayar(lar) arasında direkt olarak yapılmaktadır. İletişim için arada herhangi bir bilgisayar sistemine ihtiyaç duyulmaz.

Bilgisayar ağları kablolu sistemler üzerinde kurulmuşlardır. Bu sistemlerde iletişim kuracak bilgisayar veya diğer aygıtlar doğrudan veya dolaylı olarak fiziksel bir bağlantıları bulunmaktadır. Daha sonraki yıllardaki gelişmelerle birlikte artık cihazların iletişim kurması için fiziksel olarak bağlantıya ihtiyaçları kalmamıştır [Molisch, A.]. Mobil ve Kablosuz sistemler sayesinde sağlanan bu esneklik birçok avantajı beraberinde getirmiştir. Bu şekilde artık cihazlar sabit bir yere bağlı kalmaksızın iletişim kurmaya devam edebilmekte istenilen haberleşmeyi sağlayabilmektedir.

Kablolu ağların yüksek kablolama maliyetleri bulunmaktadır. Ayrıca, kablolu sistemlerde yol (bus), halka (ring), yıldız (star), ağaç (tree) gibi topolojiler kullanılmaktadır. Bu topolojilerin *güvenilirlik* ve *adaptasyon ve yük paylaşımı* problemleri bulunmaktadır. Güvenilirlik problemi, sistemde bir bağlantı kesilmesi durumunda tüm sistemin ya da bir grubun iletişiminin kesilmesine neden olmaktadır. Diğer taraftan adaptasyon ve yük paylaşımı probleminde ise kablolu sistemlerin başarılı bir adaptasyon ve yük paylaşımı yapabilmesi için yüksek maliyetli bir alt yapı yatırımı gerektirmektedir.

Kablosuz sistemlerde ise kablosuz cihazlar kendi aralarında kuracakları ağlar ile çok geniş ve de etkili yük paylaşımı yüksek bir ağ uygulaması gerçekleştirilebilmektedir. Bu tür sistemlerde trafiğin oluşturulup aktarıldığı düğümler veri ağının iletişimini sağlayan bağlantılar şeklinde bulunabilir. Bu yapıda oluşturulan sistemler sayesinde bilgiye başarılı şekilde ulaşım sağlanabilmektedir.

## 1.2 Tezin Amacı

Yapılan bilimsel çalışmaların ve güncel uygulamaların gösterdiği kadarıyla mobil ve kablosuz sistemlerde bilgi erişimi konularında daha fazla araştırma ve inceleme gerekmektedir. Birçok çalışma bilgiye erişimin etkin yöntemlerini bulmak amacıyla yapılmıştır. Bununla birlikte şu ana kadar mobil ve kablosuz sistemler için yapılan bilgiye erişim çalışmalarında, tüm konular incelenememiştir. Bu sebeple bu alanda yapılması gerekli çalışmalar bulunmaktadır.

Bu tezin esas amacı günümüzde popüler olarak kullanılan eşdüzey programlarının mobil ve kablosuz sistemlerde çalışan programlar tarafından da rahat bir şekilde kullanılabilir hale gelmesidir. Bu amaç ile eşdüzey sistemlerden kaynaklanan mesaj trafiğinin azaltılması hedeflenmekte, ayrıca tüketilen enerji miktarının da indirgenmesi planlanmaktadır. Bu tezde yapılan çalışmalar bu hedeflerin yakalanması doğrultusunda yapılan çalışmalardan oluşmaktadır.

## 1.3 Tezin Kapsamı

Bilgi erişim sistemlerinin kurulması ve değerlendirilmesi işlemleri önemli olarak değerlendirilmektedir. Bununla birlikte, bu adımların gerçekleştirilmesi zorluklar içermektedir ve mimari ve operasyonel birçok probleme sahiptir. Bu yüzden, bu alanda yapılacak araştırmalar çok yararlı olmaktadır. Bu noktalar göz önüne alınarak, kablosuz bilgi erişimi konusunda şu ana kadar ele alınmamış bir yaklaşım incelenmiştir.

Bu çalışmada ayrıca kablosuz bilgi erişim sistemlerinin teorik ve uygulamadaki eksikliklerine de değinilmektedir. Tezde karşılaşılan değerlendirme zorlukları ve kavramsal anlamda değerlendirmelere de değinilmiştir. Bu araştırmayı gerçekleştirebilmek izlenen yol açıklanmış ve değerlendirme metotları belirtilip bunlar arasındaki ilişkilerde belirtilerek incelemelerde bulunulmuştur.

Mobil ve Kablosuz sistemlerde bilgi erişilebilirliği çok güncel ve kapsamlı bir konu olduğu için, bu tezde incelenmeyen konularda araştırmalar yapılması tavsiye edilmektedir. Sürekli gelişmekte olan bu alanda yıllarca yeni konu bulunabilecektir.

## 1.4 Tezin Taslağı

Tezin geri kalan kısmı şu şekilde düzenlenmiştir. İkinci Bölümde bilgi erişilebilirliği konusu genel özellikleri ile incelenmiştir. Bu bölüm; *bilgi erişilebilirliğinin bilişim dünyasındaki yeri*, *içerik paylaşımını*, *bilgi erişimi konularında güvenlik ve mahremiyet*, ve *bilgi erişilebilirliğinin genel durum değerlendirmesi* alt konularından oluşmaktadır. Bu bölüm sayesinde daha önce bilgi erişimi konusunun literatürdeki genel görüntüsü verilmektedir.

Üçüncü bölümde mobil ve kablosuz sistemler ele alınmıştır. Bu amaçla üçüncü bölüm, *mobil ve kablosuz sistemlerin genel özellikleri*, *mobil ve kablosuz sistemlerin çeşitleri*, *mobil ve kablosuz dağıtık sistemlerin temel sorunları*, ve de *bilgi erişimi yüksek bir sistem uygulaması için mobil ve kablosuz sistemlerin kıyaslanması* alt bölümlerine ayrılarak incelenmiştir. Bu bölüm sonu itibarı ile mobil ve kablosuz mevcut sistemlerin iyi ve kötü yönleri tespit edilip seçilecek yöntem için bilgi zemini hazırlanmıştır.

Dördüncü bölüm, tezde önerilen bilgi erişilebilirliği yüksek sistem geliştirilmesinde ajan tabanlı ve gevşek bağlı model kullanan sistem yapısını anlatmaktadır.

Beşinci bölümde ise Mobil ve Kablosuz sistemler için ajan tabanlı ve yüksek performanslı eşdüzey iletişim için gevşek bağlı model kullanan, önerilen bilgi erişilebilirliği yüksek sistemin ne şekilde geliştirildiği, sistemin çalışma prensipleri, sistemin anlambilimsel yapısı ve sistemin entegrasyon yapısı anlatılmaktadır.

Altıncı bölümde bu tezde önerilen bilgi erişilebilirliği yüksek sistem üzerinde yapılan çalışmalar incelenmekte, çeşitli grafik ve tablolar ile sistem hakkında bilgiler verilmektedir.

Yedinci bölümde bu tezde yapılanların özeti, bu araştırmanın katkısı, sistemdeki kısıtlar ve gelecekte bu alanda yapılabilecek çalışmalardan bahsedilmekte ve çeşitli değerlendirmeler yapılmaktadır.

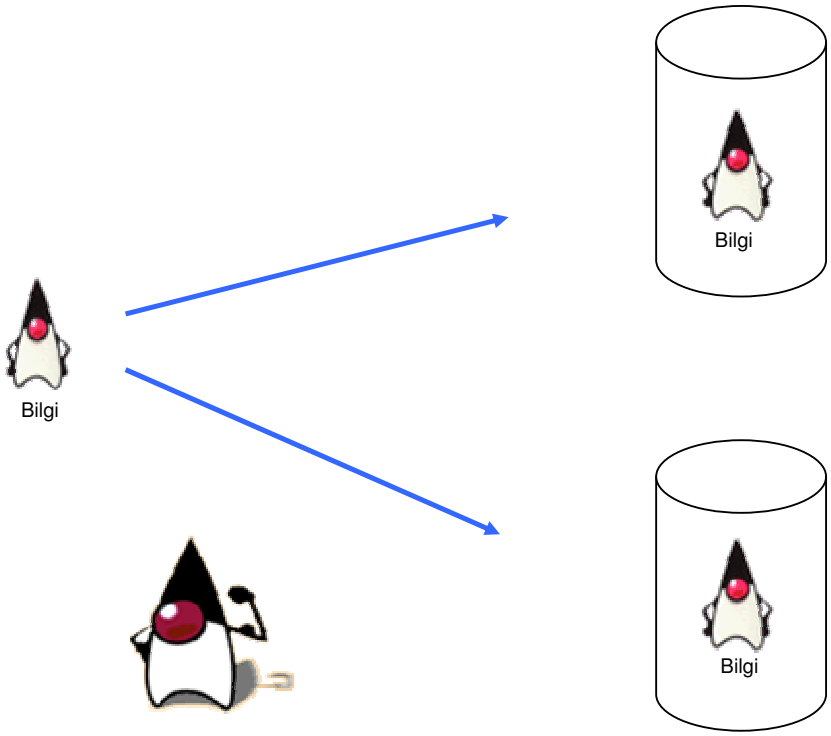
Son olarak sekizinci bölümde, bu tezdeki çalışmalar ile bu alandaki diğer çalışmalar kıyaslanmıştır.

## 2. BİLGİ ERİŞİLEBİLİRLİĞİ

Bilgi günümüzde son derece önem arz eden bir meta haline gelmiştir. Bilginin değeri bilgisayar sistemlerinin kullanılması ile üstsel bir şekilde artmaktadır. Bu bağlamda bilgiyi paylaşmak, korumak ve bilgiye erişmek gibi konular ciddi bir önem kazanmıştır.

Bu tezin amacı, işte bu çok kritik malzeme olan bilgi üzerine kuruludur. Amacı bir cümle ile özetlemek gerekirse: "*Bilgisayar dünyasının en temel servislerinden biri olan bilginin birden fazla noktada bulunması ve güvenliğinin arttırılması için yeni açılımlar getirme*." şeklinde özetlenebilmektedir.

Şekil 2.1 bu amacı açıklayan basit ama net bir örnek olarak yer almaktadır. Bu şeklin sol tarafında bilgi tek bir yerde saklanmaktadır. Bu sistemimizin tek bir kırılma noktası dezavantajı ile karşılaşmamıza sebep olmaktadır. Diğer taraftan, eğer elimizdeki bu bilgiyi birden fazla noktaya Şekil 2.1'in sağ tarafındaki gibi çoğaltırsak, sistemimiz birçok avantaj kazanmaktadır.




**Şekil 2.1** Bilginin tek noktada olması yerine (sol taraf),
birden fazla noktada olması (sağ taraf), sisteme güç kazandırır

## 2.1 Bilgi Erişilebilirliğinin Bilişim Dünyasındaki Yeri

Bilgi erişilebilirliği bilişim dünyasında kritik bir yere sahiptir. Bu yüzden bilgisayar sistemleri ilk kullanılmaya başlandığı dönemden bu yana en popüler araştırma konuları arasında yer almaktadır. Bununla birlikte yeni bulunan yöntemler ile daha önce kullanılan yöntemlerden daha başarılı uygun sistemler geliştirmek mümkün olmaktadır.

Bilgi erişilebilirliği yüksek bir sistemin kazanacağı birçok avantaj bulunmaktadır. Bunlara örnek olarak sistemdeki cihazların ve dolayısı ile bilgilerin daha fazla aktif olmasından kaynaklanan *yüksek erişilebilirlik oranı* sağlanmaktadır. Sistemin çok noktada bulunması sayesinde bazı noktalarda hata olsa bile diğer noktalar üzerinden sistemin işleyişinin sağlanarak *hataya karşı tolerans* sağlanması gerçekleştirilir. Sistemin doğru şekilde tasarlanması durumunda, sistemdeki bilginin bulunduğu konum sayısı genişlemesine rağmen (*ölçeklenebilirlik*) sistem başarılı bir şekilde çalışabilmektedir. Burada birkaçını örnekler ile açıkladığımız bu avantajların önemli bir kısmı Tablo 2.1'de listelenmiştir.

**Tablo 2.1** Bilgi erişilebilirliği yüksek bir sistemlerin potansiyel avantajları

| **Avantaj** | **Açıklama** |
|---|---|
| Yüksek Erişilebilirlik Oranı | Sistem daha fazla aktif olduğundan yüksek erişim oranı |
| Hataya Karşı Tolerans | Bir yada birkaç noktada hata olması durumunda, diğer noktalardan erişim olanağı |
| Ölçeklenebilirlik | Doğru sistem tasarımı yapılması durumunda, bilgiyi daha fazla noktada bulundurabilme |
| Yük Dengesi Sağlama | Veri alışverişi ve işleyişinin yoğun olduğu noktalardan, trafiğin az olduğu noktalara kaydırma olanağı |
| Yerel Ulaşılabilirlik Sağlama | Veri akışının daha hızlı yada daha güvenli olması için bulunulan noktaya daha yakın olan yerden bilgiye ulaşım |
| Güvenlik | Her kullanıcının istediği her noktaya ulaşımını engelleyerek, sistem güvenliklerinin arttırılması |

Hayatın her konusunda olduğu gibi bilgi erişilebilirliğinin de avantajları olduğu kadar kendine özel problemleri de bulunabilmektedir. Bu problemlerin bir kısmı şu şekilde karşımıza çıkmaktadır. Aynı bilgiyi birden fazla noktada aynı değerde tutmanın istenmesi durumunda sistemde *istikrarsızlık* durumu baş göstermektedir. Dağıtık sistemler dinamik bir yapıya sahiptirler. Bu sistemlerde daha önce çok hızlı erişilebilen bir yere diğer seferinde yavaş erişim gösterilebilir hatta bazen hiç erişilemeyebilir. Örneğin, hızlı bir bağlantının olduğu yerde hat kesilmesi durumunda, artık buraya ulaşmak mümkün olmaz, istenilen bilgiye ya bilginin saklandığı yerden ulaşılır yada sistemde yok ise ulaşılamaz. Bu tür *belirsiz erişim süresi* sistemlerin yaşadığı sorunlardan birisidir. Bir diğer sorun olarak bilginin çok noktada olmasından dolayı bilgiyi saklayan her sistemin potansiyel olarak kendine özgü sistem yapısı olması ile sistemler arası uyum sağlamak zordur. *Kullanım zorlukları* yaratan bu sistemlerin yapısal farklılıkları (heterogenous systems) sistemlere zorluk getirmektedir. Bir kısmı dağıtık sistemlerin yapısından gelen bilgi erişilebilirliği yüksek sistemlerin yaratabileceği problemler yukarıda birkaç örnekte dahil olmak üzere Tablo 2.2'de belirtilmiştir.

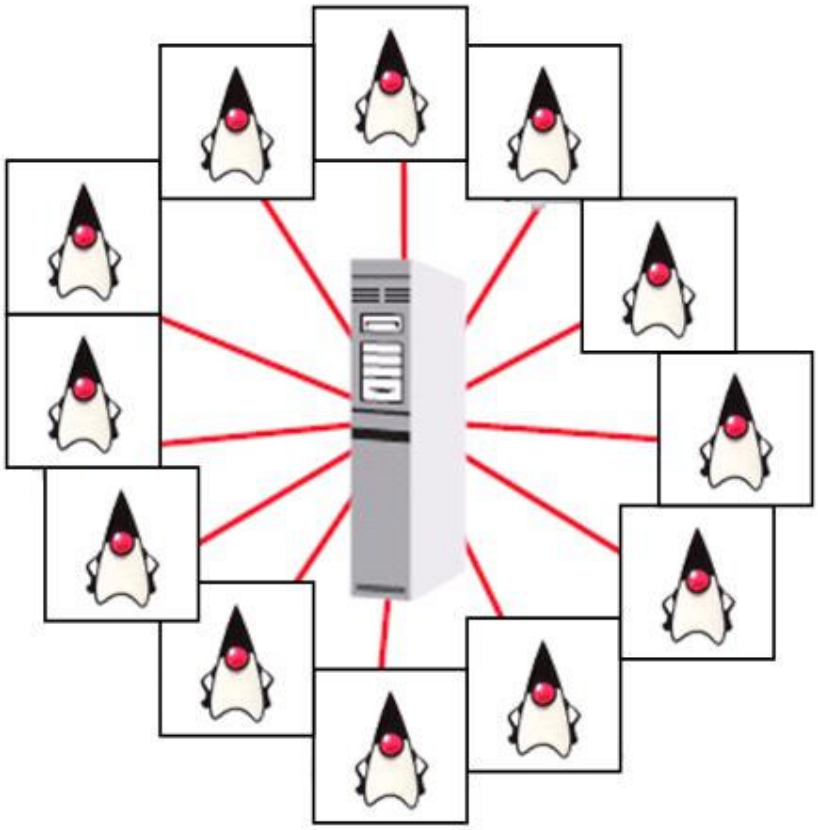
**Tablo 2.2** Bilgi erişilebilirliği yüksek bir sistemin yaratabileceği problemler

| **Problem** | **Açıklama** |
|---|---|
| İstikrarsızlık | Aynı bilginin farklı noktalarda farklı değerde olabilme sorunu |
| Belirsiz Erişim Süreleri | Bir noktadan bilgiye erişim süresinin sürekli farklılık gösterebilmesi |
| Kullanım Zorlukları | Bilgi birçok noktada olmasından dolayı sistemin kullanımının karmaşıklaşması |
| Performans | Bir bilginin çok noktada bulunmasından dolayı, bu değerin çok noktada aynı tutulması için oluşan kayıplar |
| Ölçeklenebilirlik | Seçilecek sistem metoduna göre sistem genişlemesinin belli bir noktanın ötesine geçememesi |

## 2.2 İçerik Paylaşımı

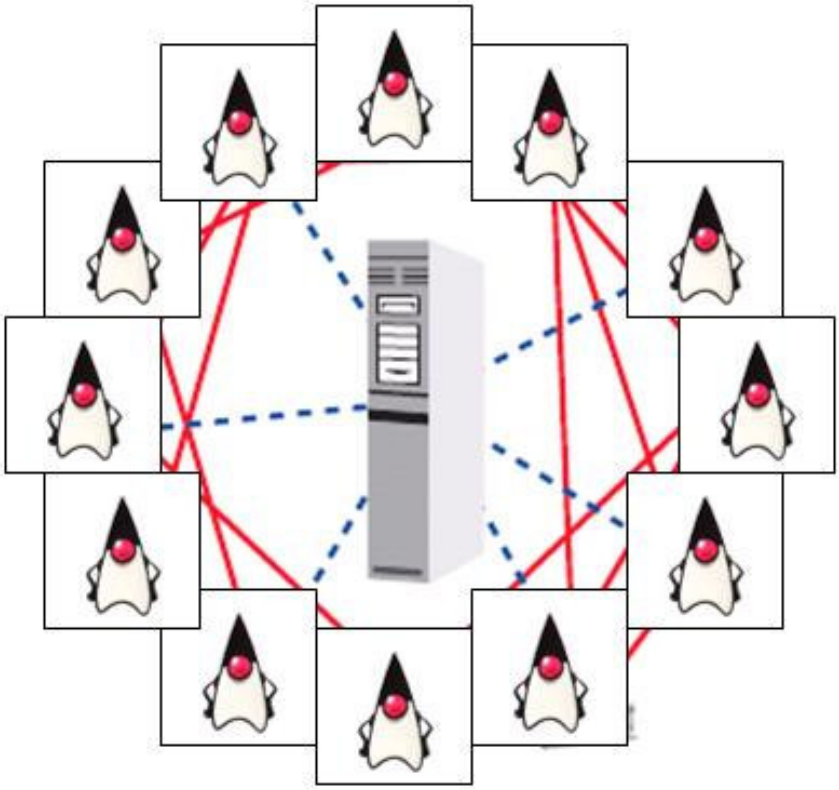
Bilgi erişiminde içerik paylaşımı önemli bir yer tutmaktadır. Bu sayede işlemler arası bilgi alışverişi yapılabilmektedir. [Dixit, S., v.d.], [Ganguly, S., v.d.]. Bilgi erişiminin üç temel yöntemi bulunmaktadır. Bunlardan ilki Şekil 2.2'de gösterilmektedir.

Bu yöntemde bilgi merkezi bir noktada tutulmaktadır. Gelen istekler merkezi nokta tarafından cevaplandırılmaktadır. Bu yöntemin ismi İstemci / Sunucu (Client / Server) olarak bilinmektedir [Bellavista, P., v.d.]. İstemci / Sunucu bilgi erişim yöntemi, çok bilinen ve sıklıkla kullanılan bir yöntemdir. Ancak bu yöntem her türlü uygulama için uygun değildir. Örneğin dosya paylaşımı için bu sistemin kullanılması sistemin işleyişinde problemler oluşturmaktadır.



**Şekil 2.2** Bilgi merkezi bir noktada saklanmakta ve istemler buradan cevaplanmakta (İstemci / Sunucu {Client / Server} yöntemi bilgi erişimi)

İstemci / Sunucu bilgi erişim yöntemine bir alternatif bilgi erişim metodu Şekil 2.3'te gösterilmektedir.

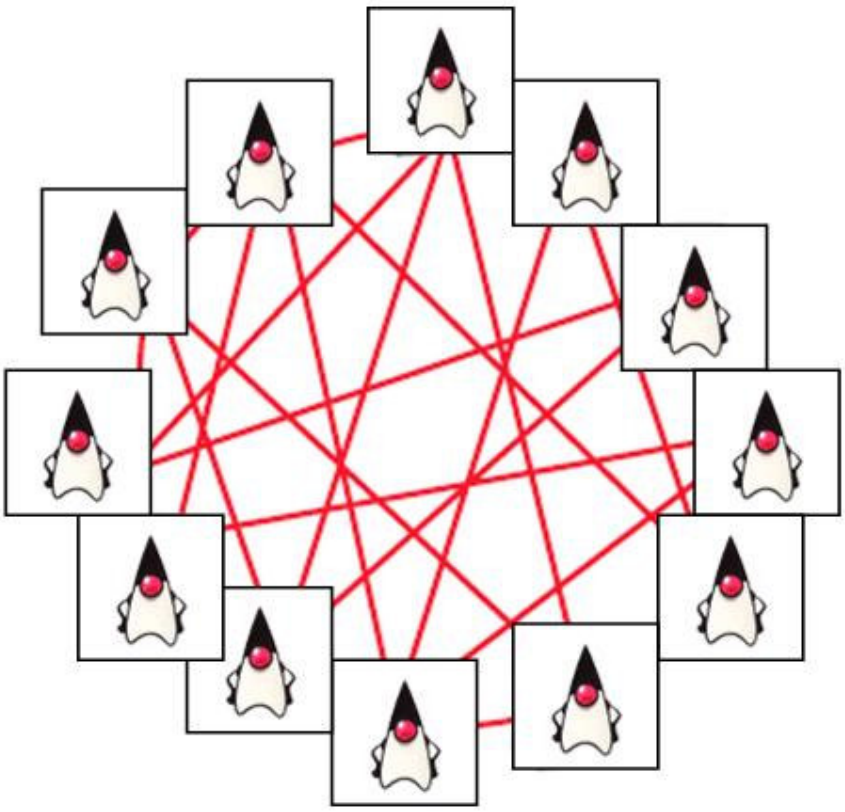


**Şekil 2.3** Bilgilerin bulunduğu yerler ve özellikleri bir noktada saklanmakta, bilgi alışverişi eşdüzey kabul edilen işlemler arasında direkt yapılmakta

Şekil 2.3'te gösterilen bilgi erişim metodunda, bilgiye erişmek isteyen bilgisayardan gelen istek merkezi olarak kabul edilen sistemle iletişime geçer [Chtcherbina, E., v.d.]. Merkezi sistemdeki bu bilgiler incelendikten sonra istenen bilginin nerelerde ne şartlarda bulunduğu bilgiye erişmek istenen bilgisayar sistemine iletilir. Bu aşamadan sonra bilgiye erişim bilgiyi almak isteyen ile bilgiye sahip olan bilgisayarlar arasında gerçekleşir. Bu sistemin ismine melez-eşdüzey (Hybrid Peer-to-

Peer) denilmektedir. Bu yöntemde ortada bir indeks bulunduğundan, Merkezi İndeks (Centeral Index) te denilmektedir.

İkinci metodun bir versiyonu olan bilgi erişim metodu ise Şekil 2.4'te gösterilmiştir.



**Şekil 2.4** Bilgi alışverişi eşdüzey kabul edilen işlemler arasında direkt yapılmaktadır.

Şekil 2.4'te gösterilen sistemde bilgi alışverişi, bilgiye ulaşmak isteyen işlem ile, gönderecek olan işlemler arasında direkt olarak yapılmaktadır. İletişim için arada herhangi bir bilgisayar sistemine ihtiyaç duyulmaz. Bu sistem, Tam Eş Düzey (Pure Peer-to-Peer) ismi ile anılmaktadır.

## 2.3 Bilgi Erişimi Konularında Güvenlik ve Mahremiyet

İletişimde bulunan bireyler arasında bilginin güvenli olarak iletilmesi ve bireylerin mahremiyetinin korunması, sistemlerin ciddi olarak kabul edilmeleri açısından çok önemlidir [Coulouris, G., v.d.]. Örnek olarak, evdeki bilgisayardan bir servis sağlayıcı aracılığı ile İnternet ortamına bağlanıldığında, bilgilerin başkası tarafından takip edilmesini istenmez. Yürütülen bu işlemlerin başkaları tarafından gözetlenmesi, kaydedilmesi veya değiştirilmesi kabul edilemez bir durum olarak karşımıza çıkmaktadır.

Bu örnekte bundan başka şöyle hoş olmayan durumlarda olabilmektedir. Örneğin, bahsedilen bu İnternet hattına başkası tarafından erişilerek, hattın izinsiz kullanılması sonucu hat sahibine ek mali yük getirme işlemi söz konusu olabilmektedir. Örneğin, son yıllarda kullanımı hızla artan IEEE 802.11 türü kablosuz alan ağı erişiminde normal hat kullanıcıları dışında hattın kullanılması şeklinde bu olay sıklıkla yaşanabilmektedir. Bu sorunu çeşitli yöntemleri kullanarak önlemek mümkündür [Ilyas, M., v.d.], [Katsaros, D., v.d.]. Bu sorunu engellemenin yöntemlerinden birisi, sistemin normal kullanıcılarının ağa erişim kartlarındaki (NIC), fiziksel adreslerini (MAC) kablosuz alan ağının yayılım listesine listeleyerek, yalnızca listede kayıtlı sınırlı sayıdaki kullanıcının ağdan yararlanması sağlanır.

Kablolu sistemlerde de kullanılan yöntemlerden birisi olan ağa erişim kartlarındaki fiziksel adreslerin listelenmesi şeklindeki güvenlik yönteminden [Tanenbaum, A.] başka türde yöntemler kullanılarak da bu örnekte bahsi geçen kablosuz alan ağlarının güvenliği sağlanabilir. Bu yöntemlerden birisi de, hattın yayılımında şifreleme yöntemleri kullanılarak dışarıdan erişim ve dinleme işlemlerinin önüne geçmektir. Kablosuz yerel ağlarda kullanılan şifreleme yöntemlerinden birisi WEP şifreleme standardıdır. Bu standart ile kablosuz sistemlerde de kablolu sistemlere eşit bir güvenlik seviyesi yakalanmak amaçlanmıştır.

Maalesef bu yöntemin keşfinden kısa süre sonra çeşitli metotlar kullanılarak bu güvenlik mekanizmasının aşılması mümkün olmuştur [Edney, J., v.d.]. Bu eksikliği ortadan kaldırmak için daha gelişmiş olan WPA şifreleme yöntemi kullanılmaya başlanmıştır. Bu yöntemde güçlü bir şifreleme hiyerarşisi kullanılmaktadır. Bu metot ile yeni bir şifreleme anahtarı mobil cihazın her bir iletişimi sırasında kullanılmaktadır

[Wi-Fi Alliance]. WPA yöntemi ile güvenli biçimde iletişimi sağlamak mümkün olmaktadır. WPA'nın bir ileri versiyonu olan WPA2 şifreleme metodu da bulunmakta olup, WPA metodunun IEEE 802.11i standardını desteklemesinde kalan eksiklikler WPA2 ile giderilmekte böylece WPA2, IEEE 802.11i standardını tam olarak desteklemiş olmaktadır.

WMN (Wireless Mesh Networks) yada kısaca *mesh ağlar* ile ilgili de çeşitli güvenlik açıkları bulunmaktadır. Bu konuda muhtelif çalışmalar ile bu problemler ortadan kaldırılmaya çalışılsa da, halen çözülmesi gereken riskler bulunmaktadır. Örneğin, mesh ağlarda merkezi bir yapı bulunmadığından, güvenilir bir merkez üzerinden *public key* dağıtımı yapılamamaktadır. [Akyildiz, I.F.,W., X., v.d.]

MANET (Mobile Ad Hoc NETwork) şeklinde adlandırılan sistemlerde de çeşitli problemler bulunmaktadır. Mesh networkler içinde önemli olan güven ve anahtar yönetimi ile kriptoloji yöntemleri kullanılarak güvenlik sağlanmaktadır. Ayrıca sisteme izinsiz girme ve güvenli paket yolu belirleme teknikleri kullanılmaktadır. Mesh sistemlerin karşılaştığı sorunlardan birisi de hizmeti engelleme (denial of service) ve selfish nodelar yüzünden yaşanan hizmet verememe durumudur [Buttyan, L., v.d.], [Guizani, M.].

## 2.4 Bilgi Erişilebilirliğinin Genel Durum Değerlendirmesi

Bilgi erişilebilirliği insanlık tarihi ile başlamış bir durumdur. Daha fazla bilgiye daha kısa sürede ulaşılması, bu yeteneğe sahip olanlar için her zaman önemli bir avantaj olmuştur. Bilgisayarın bulunması insanların yaşamını önemli ölçüde değiştirmiştir. Öncelikle tek bir bilgisayar üzerinde yapılan çalışmalar, zaman içinde bilgisayarların birbirine bağlanması fikri hayata geçirilerek çok daha kapsamlı sistemler kullanılmaya başlanmıştır.

Günümüzde bilgisayar sistemleri büyük ölçüde birbirleri ile iletişim halinde çalışmaktadır. Genellikle bilgisayar sistemleri denilince ilk akla gelen şey, sabit bir yere kurulmuş, büyük bir ekranı olan, sürekli bir enerji kaynağı bulunan cihazlar şeklinde düşünülmektedir. Son yıllarda hayata geçen değişiklikler sayesinde artık bilgisayarlar her yerde bulunabilen bir özelliğe kavuşmuşlardır. Örneğin, gelişmiş bir avuçiçi

bilgisayar ile seyahat eden bir kullanıcı gerek evinde, gerek işinde, gerekse turistik amaçlı gittiği yerlerde bu cihazını kullanabilmektedir.

Her geçen gün daha fazla kişi tarafından kullanılmaya başlanan ve kullanım amaçları hızla artan sabit bir noktaya bağlantı ihtiyacı olmayan bu sistemler birçok avantaj sağlamaktadır. Örneğin, kablosuz GPRS üzerinden Internete bağlanma özelliği olan gelişmiş bir cep telefonu ile hareket halindeyken e-postaları kontrol etmek mümkün hale gelmiştir. Ya da Wi-Fi özellikli bir Laptop veya avuçiçi bilgisayar ile Wi-Fi hizmetinin bulunduğu bir yerde Internete erişmek mümkün olmaktadır.

Bu tezin 2.2inci bölümünde bahsedildiği şekilde, içerik paylaşımı genel olarak üç şekilde yapılabilmektedir. Bunlardan ilki istemci/sunucu modelidir. Bu model uzun bir süredir kullanılmakta olan bir yöntemdir. Uzun süredir kullanılmakta olan bu modelin bir çok sıkıntısı bulunmaktadır. Bu sıkıntılardan birisi tek merkezli bilgi akışının olduğu bu sistemde iş yoğunluğunun tek noktada olmasıdır. Bu yüzden bilgi alışverişi potansiyel olarak yavaşlamaktadır. Diğer bir dezavantaj ise bilgilerin ulaşılmasını sağlayan sunucuda bir ulaşılamaz duruma gelmesinde bütün sistemin işleyişinin durmasıdır.

Yukarıda birkaç olumsuz özelliği belirtilen istemci/sunucu modeli yerine, ağ erişimi yeteneği ve kapasitesi her geçen gün artan cihazların bu yeteneklerini verimli kullanma isteğinden yola çıkılarak eşdüzey (P2P) yöntemleri kullanılması gündeme gelmiştir. Yalnızca bilgi erişiminde kullanılması gibi kısıtlı bir kullanım alanı olmayan eşdüzey modeli, çok değişik amaçlı kullanım alanlarına ayrılmaktadır. Ortak çalışma, dağıtık hesaplama ve içerik paylaşımı bu kullanım alanlarının ana gruplarını oluşturmaktadır.

Bu birçok kullanım alanı olan eşdüzey modelinin içerik paylaşımı açısından iki şekli vardır. Bunlardan birincisi, içeriği paylaşacak eşlerin (Peer) listesini tutan merkezi bir sunucu bulunma yöntemidir. Bu yöntemde erişilmek istenen içeriğe ulaşmak için öncelikle merkezi sunucuya ulaşılıp, bunun hangi eşlerden sağlanabileceği öğrenilmektedir. Hangi eşlerden içeriğe erişilebileceği öğrenildikten sonra gerekli bağlantı yapılıp içerik indirilir. Bu modeli kullanan Napster isimli popüler program dünyayı kısa süre içinde önemli ölçüde etkisi altına aldı [Ghosemajumder, S.]. Bu modelin diğer bir ismi de melez eşdüzey (hybrid peer-to-peer) olarak bilinmektedir. Bu adlandırmanın sebebi, bu modelin istemci/sunucu modelindekine benzer merkezi bir

sunucu olmasına karşın bilgi akışının eşdüzey bireyler arasında yürütülmesinden kaynaklanmaktadır.

İkinci tür eşdüzey modelinde ise içerik paylaşımında bulunan bireyler direkt olarak kendi aralarında iletişime geçmektedir. Burada merkezi bir sistem olmamasından dolayı, tek merkezliliğin yarattığı sıkıntılar ile karşılaşılmamaktadır. Bununla birlikte bilginin hangi bireylerde olduğunu bulma konusunda sıkıntılar yaşanmaktadır. Etkili bir şekilde arama işlemi yapılabilmesi için çeşitli yöntemler kullanılmaktadır. Bunlardan birisi *dağıtık hash tablosu (distributed hash table)* kullanma yöntemidir. Bu yöntem ile istenen şeyi bulmak sırasında gönderilen mesaj sayısı azaldığından, aranılana erişmek daha başarılı olarak yapılabilmektedir [Flocchini, P., v.d.].

Gnutella programı tam eşdüzey içerik paylaşım yöntemini kullanan bir popüler programdır [Aberer, K., v.d.]. Gnutella mimarisini kullanan LimeWire, BearShare, Morpheus, Mutella ve Swapper gibi birçok uygulama, şu anda yaygın bir şekilde kullanılmaktadır [Gnutella].

Gerek melez eşdüzey, gerekse tam eşdüzey modelleri genel anlamda gelişmiş özellikli sabit bilgisayar sistemleri için dizayn edilmiştir. Örneğin tam eşdüzey sistemlerde yoğun bir şekilde kullanılmakta olan dağıtık hash tablosu yöntemi sabit IP adresi olan eşit özellikli (homogenous) bilgisayar sistemleri için tasarlanmıştır. Oysaki daha öncede vurgulandığı gibi günümüzde bilgisayar sistemleri her alanda, her yerde ve her boyutta kullanılmaktadır. Bilgisayarların bu geniş spektrumlu kullanım ortamında birbirinden çok farklı bilişim cihazlarının birbiri ile iletişim kurması söz konusu olmuştur. Mimari ve işletim sistemi gibi farklı özellikteki sistemlerin (heterogeneous) bir araya geldiği bir ortamda, içerik erişimi ile ilgili konularda da farklı sistemler geliştirilmesi ihtiyacı ortaya çıkmıştır.

Örnek olarak bir mobil cihazı ele aldığımızda; mobil cihazın işlemci gücü sınırı, kısıtlı hafıza kapasitesi ve az miktardaki depolama özelliği problemler oluşturmaktadır. Diğer taraftan, mobil ve kablosuz sistemlerde ağa bağlanmanın genellikle yüksek ücrete bağlı olması, bu sistemlerin ağa bağlanma sürelerini yada gönderip aldıkları bilgi miktarını kısıtlama eğilimine girmelerine sebep olmaktadır. Ayrıca, mobil ve kablosuz sistemlerdeki iletişimin kesintiye uğraması yaygın bir şekilde karşılaşılan durumdur. Bazı yerlerde servis verilmemesi, bazı durumlarda ise sistemin çevresel etkenlerden dolayı düşük verimde çalışması yada devre dışı kalmasından dolayı bu sistemlerin

çalışma şekli kablolu bağlantısı olan sabit ve gelişmiş bir bilgisayar sisteminden oldukça farklı özellikler göstermektedir [Zoels, S., vd. 1].

Bilgi erişimi eşdüzey erişim modelleri kullanımı ile çok daha etkili şekilde gerçekleşme fırsatı bulmuştur. Bu fikirden yola çıkılarak çeşitli mimari tasarımlar hazırlanmıştır. Geliştirilecek bir sistemin hangi tür cihazlar ve servisler için kullanılacağı iyi analiz edilmeli, bu analiz sonucu gerekli tasarım gerçekleştirilmelidir.

# 3. MOBİL VE KABLOSUZ SİSTEMLER

Bilgisayar sistemlerinin geçmişten bugüne kadar oluşan durumuna baktığımızda çok ciddi bir değişiklik yaşandığını görmekteyiz. Bilgisayar sistemleri ilk dönemlerinde fiziksel olarak çok büyük yer kaplayan cihazlar olarak, büyük laboratuvarlarda yer almaya başlamışlardır. Bu ilk sistemlere girilen veriler kartlara ya da yazıcıdan kâğıtlara dökülmekteydi. Daha sonraları bireylerin bizzat kullanabilecekleri bilgisayarlar yavaş yavaş devreye girmeye başlamıştır. O dönemde bu cihazlar halen çok pahalı olmakta ve genellikle laboratuvarlarda kullanılmalarına rahmen, bilgisayarı kullanan kişiler monitör aracılığı ile bilgisayarlar ile nispeten daha yüksek etkileşim içine girebilmişlerdir.

Bu aşamaya kadar olan bilgisayar sistemlerinde işlemler yanlızca tek bir bilgisayar üzerinden çalışmıştır. Dolayısı ile, hesaplamada ya da haberleşmede bilgisayar sistemleri arasında bir iletişim söz konusu olmamıştır. Haberleşmenin önemli bir şey olacağı gerçeği göz önüne alındığında, bilgisayar sistemlerini kablolar ile birbirine bağlayarak etkinliklerinin arttırılması fikri ortaya çıkmıştır. Bu bağlamda, bilgisayarlar öncelikle yerel ağlar daha sonraları da orta ve uzak mesafeli kablolu ağlar ile birbirlerine bağlanarak bilgisayar sistemlerinin daha etkin olarak çalışmaları sağlanmıştır [Tanenbaum, A.].

Kurulan bu kablolu altyapı sayesinde bilgisayarlararası iletişimin etkin olarak kullanılmasının, birçok farklı amaca oldukça başarılı bir şekilde hizmet ettiği gözlemlenmiştir. Mesela bir bilgisayarın bir saatte bitirebildiği bir program, doğru bir şekilde düzenlenmesi durumunda, iletişim halindeki iki bilgisayar çok daha kısa bir sürede, örneğin otuz beş dakikada tamamlanabilir. Buna benzer performans gelişmeleri [Peterson, L.L., v.d.] çalışmasında da bahsedildiği gibi birçok farklı çalışmada irdelenmiştir.

Birden çok bilgisayar sisteminin iletişim halinde çalışması fikri çok çeşitli avantajlar sunmaktadır. Bununla birlikte elde edilen avantajlar bir noktada maksimum limitlerine ulaştı. Bu limit: "*Kabloların Yarattığı Limitler*" den başkası değildi. Bilgisayar sistemleri kendi aralarında ancak kablolu bir bağlantı olması durumunda iletişim kurabiliyorlardı. Bu olay özellikle bilgisayarların günlük hayatta daha fazla kullanılması ile ve bu sistemlerin küçülüp hafiflemeleri ile ciddi bir kısıtlama olarak bilişim dünyasının önüne engel olarak çıkmıştır.

İşte bu gerçeklerden yola çıkan araştırmacılar bilişim sistemlerinin iletişimini kablosuz ortama aktarmaya başlamışlardır [Molisch, A.]. 1990'lı yılların başında cep telefonu sistemlerinin dünyada hızla yaygınlaşması ile artan eğilim, 2000'li yılları yaşadığımız bu dönemde ciddi miktarda bilgisayar dünyasındaki diğer cihazlara da yayılmıştır. Bu bağlamda, artık birçok havaalanı, otel, otobüs terminali, okul da kablosuz Internet bulunabildiği gibi, evlerde de kullanımı hızla artmaktadır. Günümüzde bulunan tek kablosuz iletişim şekli burada kablosuz Internet olarak bahsettiğim IEEE 802.11x serisi teknolojiler ya da halk arasında cep telefonu şeklinde adlandırılan hücresel sistemler değildir. Bluetooth, Kızılötesi ve GPS gibi daha birçok kablosuz ve mobil teknoloji günümüzde bilişim dünyasının hizmetine sunulmuş ve yoğun bir şekilde kullanılmaktadır.

### 3.1 Mobil ve Kablosuz Sistemlerin Genel Özellikleri

Kablosuz sistemlere genel olarak bakıldığında, bir yere fiziksel bağlanmaksızın iletişimi sürdürebilme ve hareket özgürlüğünün teorik olarak sınırsız olduğu bir yapı karşımıza çıkmaktadır. Basit bir kablosuz sistem kurulduğunda ise gerçekte durum böyle değildir. Bunun sebebi, sistemin çalışabilirliği, bağlı olunan baz istasyonu yada karşı terminalin kapsama alanı gibi sınırların bulunmasıdır. Bu gerçekten yola çıkılarak mobil sistemler hayata geçirilmiştir. Bu bağlamda kapsama alanı biten bir kablosuz sistemin mobilitesini sağlamak için birçok baz istasyonu kurulmuş, bu şekilde iletişim kuracak kablosuz cihazın hareketlilik alanı arttırılarak, mobilitesi sağlanmıştır. Bu aşamada, her mobil sistemin kablosuz olması gerekmediğinide belirtmek gerekir [Coulouris, G., v.d.]. Buradaki önemli etken, mobil ve kablosuz sistemlerin sıklıkla birbirleriyle karıştırılmalarıdır. Mobil ve kablosuz özellikleri incelendiğinde sistemler dört ana gruba ayrılmaktadır.

Bunlardan birincisi, bir sistemin iletişim işlemlerini kablolu ortamda yaparak sürekli sabit bir yerde kullanılması durumunda karşımıza çıkmaktadır. Bu tür sistemlere *ne mobil, ne de kablosuz özelliği olan sistem* ya da diğer bir değişle *sabit sistem* denilir. Bu tür bir bilgisayar sistemine Ethernet kablolu erişim yöntemi ile iletişim işlemlerini yürüten bir masaüstü bilgisayarı örnek olarak verebiliriz. Bu örnek sistemin iletişim kurabilmesi için kablolu Ethernet bağlantısının yapılmış olması gerekmektedir. Sabit bir

yerde bulunan bu masaüstü sistemin, ne mobil ne de kablosuz özelliği bulunmamaktadır.

İkinci tür sistem ise kablosuz iletişim özelliği bulunan ancak bulunduğu çevreden uzaklaşıp bağlı olduğu baz istasyonu yada erişim noktası (AP) dışına çıktığında, iletişim kurmaya devam edemeyen sistemleri örnek verebiliriz. Bu sistemlere, *kablosuz ancak mobil olmayan sistemler* denilmektedir. Burada ikinci olarak belirtilen türdeki sistemlere örnek olarak, ev veya işyerlerinde kullanılan DECT [DECT Forum] telefon sistemleri verilebilir. Bilindiği üzere DECT telefonlar baz istasyonunun kapsama alanı içinde kablosuz olarak iletişim kurmaya olanak verse de, belli bir uzaklıktan sonra iletişim kurmak mümkün değildir.

Üçüncü grup olarak karşımıza, *mobil özelliği olan ancak kablosuz bağlantısı olmayan* sistemler gelmektedir. Örneğin, eski model bir Laptop ya da avuçiçi bilgisayarda kablosuz bağlantı özelliği bulunamayabilir. Böyle sistemlere bilgi yüklemek ya da göndermek istenildiği zaman kablolu bağlantı yapılarak gerekli iletişim sağlanabilir. Bu örnekte adı geçen gerek Laptop gerekse avuçiçi bilgisayarlar boyutları ve enerji yapıları sayesinde yer değiştirilmelerine müsait olmaları sayesinde gelen mobil özelliklerine rağmen, kablosuz bağlantıları bulunmamaktadır.

Mobil özelliği olan ancak kablosuz özelliği olmayan örneğin biraz değişik varyasyonu şu şekilde de gerçekleşebilmektedir. Bahsi geçen mobil cihazın kablosuz bağlantı özelliği olsa bile, eğer bu bağlantı için bir ödeme yapılması gerekiyor ise, ya da yapılacak kablosuz bağlantılar güvenlik v.b. gerekçeler ile engellenmesi söz konusu ise, bu sistemler bir tür mobil ama kablosuz olmayan sistem kategorisine girmektedir.

Son olarak, burada bahsedilen dördüncü grup sistemler ise, *hem mobilite özelliğine sahip, hem de kablosuz iletişim kurabilen* sistemlerdir. Bu grupta yer alan sistemler, sürekli hareket etmesine rağmen kablosuz iletişim kurmaya devam edebilme özelliğine sahiptirler. Örneğin, GSM, ya da i-mode gibi cep telefonu sistemleri hem mobil, hem de kablosuz olan sistemler sınıfına girmektedir. Bu bağlamda GSM telefonu olan bir kişi iletişime başladığında hareketli bile olsa iletişimini sürdürmeye devam eder.

Yukarıda genel anlamda bahsi geçen mobil ve kablosuz sistemlerinin pratik hayata yansıyan avantajları görüldükçe hızlı bir şekilde yararlı olan bu sistemlerin kurulması gerçekleşmiştir. Bu sistemlerden bazıları beklenen düzeyde başarılı olamasa

da, örneğin AMPS teknolojisi, mobil ve kablosuz sistemler hakkındaki genel görüş bu sistemlerin önemli yararlarının olduğu şeklindedir. Bu genel görüş ile kullanılmaları ve değişik yerlere uygulanmaları devam edilmekte ve ayrıca daha gelişmiş türevleri aralıksız bir şekilde geliştirilmeye devam etmektedirler.

Bu bağlamda mobil ve kablosuz sistemlerin bilişim dünyasına sağladığı *esnek erişilebilirlik olanakları*, *sisteme dahil olma kolaylığı*, *ölçeklenebilirlik imkanı*, *bulunulan ortama göre servis verebilme olanakları* şeklinde avantajları bulunmaktadır. Mobil ve kablosuz sistemlerin bilişim dünyasına getirmiş olduğu bu avantajlar Tablo 3.1'te gösterilmiştir.

**Tablo 3.1** Mobil ve kablosuz sistemlerin potansiyel avantajları

| **Avantaj** | **Açıklama** |
|---|---|
| Esnek Erişilebilirlik Olanakları | Yüksek bulunabilirlik sayesinde erişim kolaylığı |
| Sisteme Dahil Olma Kolaylığı | Fiziksel bir kurulum gerektirmeden dahil olma |
| Ölçeklenebilirlik | İyi tasarım yapılması durumunda kolay genişleme kapasitesi |
| Bulunulan ortama göre servis verebilme olanakları | Cihazlar bu sistemlerde sabit olmadığından, bulundukları ortamlar sürekli değişmektedir. Bulunulan ortama göre uygun hizmet sunulmaktadır |

Hemen hemen her sistemin olumlu yönleri olabileceği gibi olumsuz yönleri de bulunmaktadır. Çeşitli avantajları bulunan mobil ve kablosuz sistemlerinin incelenmesi sonucunda bu sistemlerin de çeşitli dezavantajları gözlenmiştir. Sürekli bir bağlantı garantisi olmaması ve yüksek hata olasılıkları sebebi ile *belirsiz erişim süreleri* bu sistemlerin yaşadığı ilk problem grubudur. Kablolu sabit bağlantı yöntemlerine göre sınırlı bağlantı hızları yüzünden *düşük performans* göstermektedirler. Çeşitli mobil ve kablosuz yöntem bulunduğundan, seçilen mobil ve/veya kablosuz sistemin uygun olmaması durumunda sistemin *ölçeklenebilirliği* sınırlanabilmektedir.

Kablosuz ortamda yapılan yayınlar, elektromanyetik spektrum üzerinden birçok farklı cihaza erişebilmektedir. Bu durum *güvenlik* açısından birçok risk taşımaktadır. Diğer bir problem ise mobil ve kablosuz sistemlerinin *canlıların sağlığına muhtemel etkileri* olarak karşımıza çıkmaktadır. Elektromanyetik spektrum üzerinden yayılan sinyaller canlılar tarafından emildiğinde muhtemelen olumsuz olarak etkilenebilmektedirler. Ayrıca, bu sistemlerin yakın temas ile kullanılması ile ısınma etkisinin de canlılar üzerinde olumsuz yönde rol oynadığı bilinmektedir.

Son olarak bir yere bağlı kalmaksızın hareket özgürlüğü fikri üzerine dizayn edilmiş bu sistemlerin *enerji ihtiyacı problemleri* bulunmaktadır. Burada değinilen mobil ve kablosuz sistemlerin problemleri Tablo 3.2'te gösterilmiştir.

**Tablo 3.2** Mobil ve kablosuz sistemlerin yaratabileceği problemler

| **Problem** | **Açıklama** |
|---|---|
| Belirsiz Erişim Süreleri | Cihazların bulundukları konum ve şartlara göre farklı erişim süreleri vermeleri |
| Performans | Kablosuz ortamda sınırlı olan bant genişliği ve yüksek olan gürültü gibi sebeplerden dolayı performans kaybı |
| Ölçeklenebilirlik | Kullanılan teknolojinin sistem genişlemesini kısıtlama olasılığı |
| Güvenlik | Herkese açık olan bir ortamda bilgi iletişimi sırasındaki güvenlik riski |
| Canlıların Sağlığına Etkileri | Elektromanyetik spektrum kullanılarak yayılan dalgaların canlılara muhtemel etkisi, cihazların ısınması ile etkilenme |
| Enerji ihtiyacı | Hareketli sistemlerin sabit kaynaktan enerji alamama durumu |

Sürekli gelişmekte olan mobil ve kablosuz sistemler alanına her geçen gün yeni teknolojiler ve gelişmiş versiyonlar ilave edilmektedir. Örneğin IEEE 802.11n [Aziz,

M.K.A., v.d.], IEEE 802.16 (WiMAX) [WiMAX Forum], WiBro [WiBro Org], HSDPA [Nokia HSDPA], ZigBee [ZigBee Alliance], Near Field Communication [Near Field Com.], Ultra Wideband (UWB) [Ultra Wideband] gibi teknolojiler, son dönemde öne çıkan mobil ve kablosuz yeni teknolojilerden yalnızca birkaçıdır.

### 3.2 Mobil ve Kablosuz Sistemlerin Çeşitleri

Günümüzde birçok çeşit mobil ve kablosuz sistem bulunmaktadır. Bunlardan ilk grup *hücresel* (cellular) sistemler olarak adlandırılan, kapsama alanının genişletilmesi için birden fazla baz istasyonu bulunan sistemlerdir. Bu teknolojin çalışma prensibinde sürekli bir kapsama alanı yaratılması fikri bulunmaktadır. Sunulan servisler taşınabilir telefon sistemleri ile ses, veri ve de son dönemdeki gelişmeler ile birlikte hızlı veri akışı gerektiren çoklu ortam (multimedia) hizmeti sunulmaktadır. Hücresel sistemlerin en büyük kısıtları, bu sistemlerin sınırlı bant genişlikleri yüzünden yavaş veri iletişimine olanak vermesi ile birlikte, az öncede değinildiği gibi, üçüncü nesil gibi son dönemde kullanılmaya başlanılan sistemler ile birlikte bu problemler kısmen aşılmaya başlanmıştır.

Diğer bir kablosuz teknoloji, *kablosuz yerel alan ağı*'dır (WLAN). Bu teknolojinin sağladığı servisler ile kablosuz arayüz ile geleneksel yerel alan ağına bağlanılmaktadır. IEEE 802.11x versiyonu teknolojiler ya da Motorola'nın ALTAIR sisteminin örnek verilebileceği WLAN'ların, şu an için yaşadıkları en büyük sınırlama kısıtlı alanlarda çalışabilme ve aynı bölgede bulunan WLAN servisleri arasında geçiş yapılamaması durumudur.

*Küresel konumlandırma sistemi* ya da diğer bir değişle GPS sistemi cihazın o anda dünya üzerinde bulunduğu yeri 3 boyutlu olarak belirtmektedir. Ayrıca bu teknoloji sayesinde cihazın hızını da ölçmek mümkündür. Dünyanın hemen-hemen her yerinde çalışabilen GPS sisteminin yaşadığı en büyük sıkıntı ise bu sistemin göreceli olarak pahalı olmasıdır.

Bir diğer mobil ve kablosuz teknoloji ise *uydu tabanlı kişisel iletişim servisleridir*. Şu ana kadar bahsi geçen mobil ve kablosuz teknolojilerden farklı olarak, bu sistemlerde iletişim kurulumu sırasında dünya üzerindeki bulunan cihazlar üzerinden değil fakat dünyanın yörüngesinde gezmekte olan uydulardan faydalanılarak iletişim

yürütülmektedir. Bu sistemlerin o şekildeki özelliklerinden ötürü hava şartlarına göre sistemin çalışma şekli değişebilmektedir. Bu teknolojiler ile çağrı cihazı hizmetleri, ses, görüntü gibi birçok çeşitte servis sunulabilmektedir.

Uydu tabanlı kişisel iletişim teknolojilerinin karşılaştığı en büyük sorun yüksek kullanım ücretleridir. Özellikle sistemden beklenen servis kalitesinin (QoS) yüksek tutulması amaçlandığı durumlarda çok pahalı sistemler ortaya çıkabilmektedir. Iridium ve Teledesic sistemleri uydu tabanlı kişisel iletişim servislerine iki örnektir. Yaklaşık olarak tüm dünyayı kapsayan bu iki örnekten başka, bazı sistemler ise dünyanın bir kısmını kapsayarak çalışan uydu tabanlı sistemler bulunmaktadır. Örneğin, Birleşik Arap Emirlikleri merkezli Thuraya firması [Thuraya] Orta Doğunun tamamı, Avrupa'nın tamamına yakını, Afrika'nın önemli bir kısmı ve Uzak Doğunun bir kısmını içine alan bir bölgede çalışmaktadır.

Diğer bir kablosuz teknoloji *Ad-Hoc ağlardır*. Bu teknoloji, her geçen gün sayısı artmakta olan kablosuz cihazlar kullanılarak, önceden herhangi bir alt yapı kurulumuna gerek kalmaksızın yürütülen bir kablosuz iletişim yöntemidir. Aynı mekânda bulunan bilgisayarların bilgi alışverişi yaptığı bu sistemde kapsama alanı, kablosuz yerel ağlara benzer şekilde olmaktadır. Bluetooth, Ultra-wide band ve ZigBee sistemleri Ad-Hoc iletişim yöntemine örnek sistemlerdir.

Yukarıdaki bahsedilen mobil ve kablosuz sistemlerden başka, *algılayıcı ağlar* da bir tür kablosuz teknolojidir. Bu sistemler kablosuz bağlantı özellikli küçük algılayıcılardan oluşmaktadır. Küçük bir bölgede çalışma özelliğine sahip olan bu cihazların en büyük sıkıntısı bu sınırlı çalışma alanları olmasıdır. Ayrıca, çevreye bilinçsizce atılması durumunda çevre kirliliğine de sebep olabilirler.

Algılayıcı ağlar teknolojisi savunma sistemlerinde ve sivil uygulamalarda kullanılabilmektedir. Örneğin, üzüm bağı olan bir çiftlikte bağın hava şartlarını sürekli kontrol ederek, gerekli müdahale yapılması sağlanabilmektedir. Bu bağlamda, sıcaklığın belli bir oranın üzerine yükselmesi durumunda sulama yapılması ya da gölgelendirme çalışması, donma tehlikesi yaşanması durumunda ise sıcak hava vererek bu durumun bertaraf edilmesi yoluna gidilebilmektedir. Bu sistemlerin askeri alanda kullanılabilecek bir örneği ise, belli bir bölgede hareket olup olmadığının kontrol edilmesi için kullanılmasıdır.

Bu bölümde genel olarak bahsettiğimiz mobil ve kablosuz teknolojilerin çeşitleri, özellikleri ve her biri için örnek sistemler, Tablo 3.3'te kategorize edilmektedir.

**Tablo 3.3** Mobil ve kablosuz teknolojilerin çeşitleri ve özellikleri

| **Teknoloji** | **Servisler/ Özellikler** | **Kapsama Alanı** | **Kısıtlamalar** | **Örnek Sistemler** |
|---|---|---|---|---|
| Hücresel | Taşınabilir telefonlar ile ses ve veri | Sürekli Kapsama | Düşük bant genişliği | Hücresel tel., avuçiçi bilgisayar |
| Kablosuz Yerel Alan Ağı (WLAN) | Kablosuz arayüz ile geleneksel yerel alan ağı | Yalnızca yerel yerde | Kısıtlı kapsama alanı | IEEE 802.11x, Motorola'nın ALTAIR |
| Küresel Konumlandırma Sistemi (GPS) | 3 boyutlu pozisyon ve hız | Dünyanın her yerinden | Pahalı | GNSS, NAVSTAR |
| Uydu-Tabanlı Kişisel İletişim Servisleri | Çağrı cihazı hizmetleri, ses | ~ Tüm dünya | Pahalı | Iridium, Teledesic, Globalstar, |
| Ad-Hoc Ağlar | Aynı mekândaki bilgisayarların bilgi alışverişi | Yerel Ağlara Benzer | Kısıtlı kapsama alanı | Bluetooth, Ultra-wide band, ZigBee |
| Algılayıcı Ağları | Kablosuz bağlantı özellikli küçük algılayıcılar | Küçük bir bölge | Kısıtlı kapsama alanı | Savunma ve sivil uygulamalar |

### 3.3 Mobil ve Kablosuz Dağıtık Sistemlerin Temel Sorunları

Mobil ve kablosuz sistemlerin sağladığı birçok faydası bulunmakla birlikte, bu sistemlerin kullanılması sırasında ortaya çıkan sıkıntılar, diğer bir deyişle dezavantajları da bulunmaktadır. Örneğin, mobil ve kablosuz sistemlerin canlıların üzerinde sağlık problemi yapıp yapmama durumu ciddi tartışmalar uyandırmaktadır [Moulder, J.E.].

900 MHz'te çalışan bir GSM telefon sisteminin sağlığa zarar verip vermediği konusunda çeşitli görüşler bulunmakla birlikte [Elektro Dalga ve Sağlık] genel olarak $10^{16}$ Hertz'in altında frekanslarında çalışan teknolojilerin elektro manyetik yayılımların canlılara zararsız olduğu kabul edilmektedir. Zarar olasılığını azalmak için yapılabilecek şeylerden birkaçı, baz istasyon sayısını arttırıp, her bir baz istasyonunun yayılım gücünü azaltmak, baz istasyonu verici cihazlarını canlılar ile aynı hizada değil de, daha yukarıda bulundurmak, canlılar ile yayılım yapan cihazlar arası mesafeyi uzun tutmak (resmi rakamlara göre en az 8 metre).

Elektro manyetik sinyallerin zararları dışında, mobil ve kablosuz sistemlerin ısınma yöntemi ile de canlılara zararları oluşabilmektedir. Buna örnek olarak cep telefonlarındaki ısınma sonucu başta küçük çocuklarda (kafataslarının ince olmasından dolayı) olmak üzere insanlar üzerinde ısınma sonucu da zararlı sonuçlar doğurabildiği görülmektedir.

Mobil ve kablosuz sistemlerin diğer bir problemi ise enerjiye olan ihtiyaçlarıdır. Bu sistemler bir yere sabit olarak kullanılan sistemlerde olduğu şekilde enerjileri prize takıp alamadıklarından dolayı, çalışmalarında enerjinin verimli olarak kullanılması gerekmektedir. Bahsi geçen bu enerji problemini aşmak için gerek Türkiye'de, örneğin Vestel'in geliştirmekte olduğu sistemler, gerekse de dünyanın birçok yerinde çeşitli projeler geliştirilmektedir.

Bundan başka, mobil ve kablosuz sistemlerin kullanımı sırasında herkese açık ortamda yapılan yayınların, istenmeyen kişiler tarafından dinlenmesi ya da yapılan iletişimin başkası tarafından yapılıyormuş gibi yapılması durumu gerçekleştirilebilir.

Bu sistemlerde görülen diğer bir problem ise bu teknolojileri kullanan cihazların bilgisayara daha fazla benzedikçe, bu sistemlerde artan bilgisayar virüsleridir. Öyle ki, günümüzde mobil ve kablosuz ortam üzerinden bulaşan virüs miktarı hızla artmaktadır.

Mobil ve kablosuz sistemlerdeki iletişim sınırlamaları da bu sistemlerin kablolulara göre daha yavaş bir şekilde iletişim kurma özellikleridir. Bu problemin gerçekleşme sebebi sınırlı olan elektro manyetik spektrumdan ancak belli hızda bilgi taşınabilmesidir. Ayrıca bu mobil ve kablosuz sistemlerde bit hata oranı yüksek olduğundan iletişim hızı bu açıdan da olumsuz olarak etkilenmektedir.

Mobil ve kablosuz sistemlerin karşılaştığı bir diğer problem ise bu sistemlerin sürekli hareketli olmalarından kaynaklanan ve ayrıca çevresel faktörlerden etkileri ile,

iletişim sırasında beklenmedik problemler yaşayabilmeleridir. Bu bağlamda bu sistemlerle iletişim sırasında beklenmedik yavaşlamalar ve de kesintiler gerçekleşebilmektedir. Örneğin kablosuz erişimin dışına çıkmış bir cihaz otomatik olarak iletişimden kopmuş olmaktadır.

## 3.4 Bilgi Erişimi Yüksek Bir Sistem Uygulaması İçin Mobil ve Kablosuz Sistemlerin Kıyaslanması

Bu bölümde bahsedildiği gibi birçok mobil ve kablosuz teknoloji bulunmaktadır. Bunlardan bazılarının kullanıcılara verdiği hizmetler sınırlıyken, bazıları ise kapsama alanı ya da düşük bant genişliği gibi problemler ile karşılaşmaktadırlar.

Örneğin hücresel sistemler sürekli kapsama yeteneği ile kullanıcılarına büyük kolaylıklar sağlaması ile birlikte, düşük bant genişliği yüzünden diğer kablosuz sistemlere göre genellikle daha yavaş hizmet vermektedir. Diğer taraftan kablosuz yerel alan ağları genellikle hızlı şekilde hizmet verebilmektedir. Örneğin IEEE 802.11b 11Mbit/s ve IEEE 802.11g ise 54 Mbit/s hızlarında hizmet vermektedirler. Bununla birlikte bu sistemlerin yalnızca dar bir alanda çalışmaları ve aynı teknolojiyi kullanan servislere geçiş sağlanamaması bu sistemlerin günümüzde yaşadığı problemlerdendir. Küresel konumlandırma sistemi diye adlandırılan GPS ise dünyanın her yerinde kullanılabilmesine rağmen diğer sistemlerle kıyaslandığında kullanılması pahalı bir yöntemdir. GPS sisteminin hızı da uydu servisleri kullanıldığı için pahalı olmaktadır. Uydulardan faydalanan diğer bir sistem ise uydu tabanlı kişisel iletişim servisleridir. Bu sistemlerde çağrı cihazı hizmetleri, ses ve de veri iletişimi yapılabilmektedir. GPS sistemine benzer şekilde dünyanın hemen hemen her noktasında çalışabilen versiyonları olduğu gibi dünyanın belli bir kısmına odaklanmış şirketler tarafından geliştirilen örnekleri de bulunmaktadır. Bu sistemler hem pahalı olması açısından, hem de bu sistemleri kullanacak kişilerin taşıması gereken cihazların büyük boyutlu olması açısından olumsuz özelliklere sahiptir.

Ad hoc ağlara bakıldığında ise aynı mekândaki bilgisayarların bilgi alışverişinde bulunması şeklinde olmaktadır. Kablosuz yerel alan ağları ile benzer kapsama alanları bulunmakla birlikte iletişim ortamında bulunan cihazların bir araya gelerek grup kurması sonucu iki cihazın normalde ulaşamayacağı daha uzak mesafelere ulaşılarak

iletişim kurulabilmesi mümkündür. Son olarak algılayıcı ağlar ucuz sistemler olmalarına rağmen küçük bir bölgede çalışmaları ve de enerjileri bittiğinde yeniden enerji yüklenememesi şeklinde problemleri bulunmaktadır.

Bütün bu gözlemler ışığında, bu tezde kullanılmak üzere kablosuz yerel alan ağı sistemi kullanılmasına karar verilmiştir. Bu seçimin yapılmasında aşağıdaki faktörler rol oynamıştır. Bu karardaki ilk etken, bu bölümde daha önce belirtildiği gibi kablosuz yerel alan ağları günümüzde sıklıkla kullanılan bir teknolojidir. Bu bağlamda bu teknolojinin kurulmuş olduğu birçok sistem bulmak kolaydır. İkinci olarak, bu teknolojinin veri iletişimine GPS ya da algılayıcı ağlar gibi sistemler ile kıyaslandığında daha uygun olmasıdır. Üçüncü olarak, Jini teknolojisinin bir servisi olan JavaSpaces, kablosuz yerel alan ağlarında kullanılmaya uygundur.

# 4. BİLGİ ERİŞİLEBİLİRLİĞİ YÜKSEK SİSTEMLERİN GELİŞTİRİLMESİNDE GEVŞEK BAĞLI MODEL KULLANAN AJAN TABANLI SİSTEM YAPISI

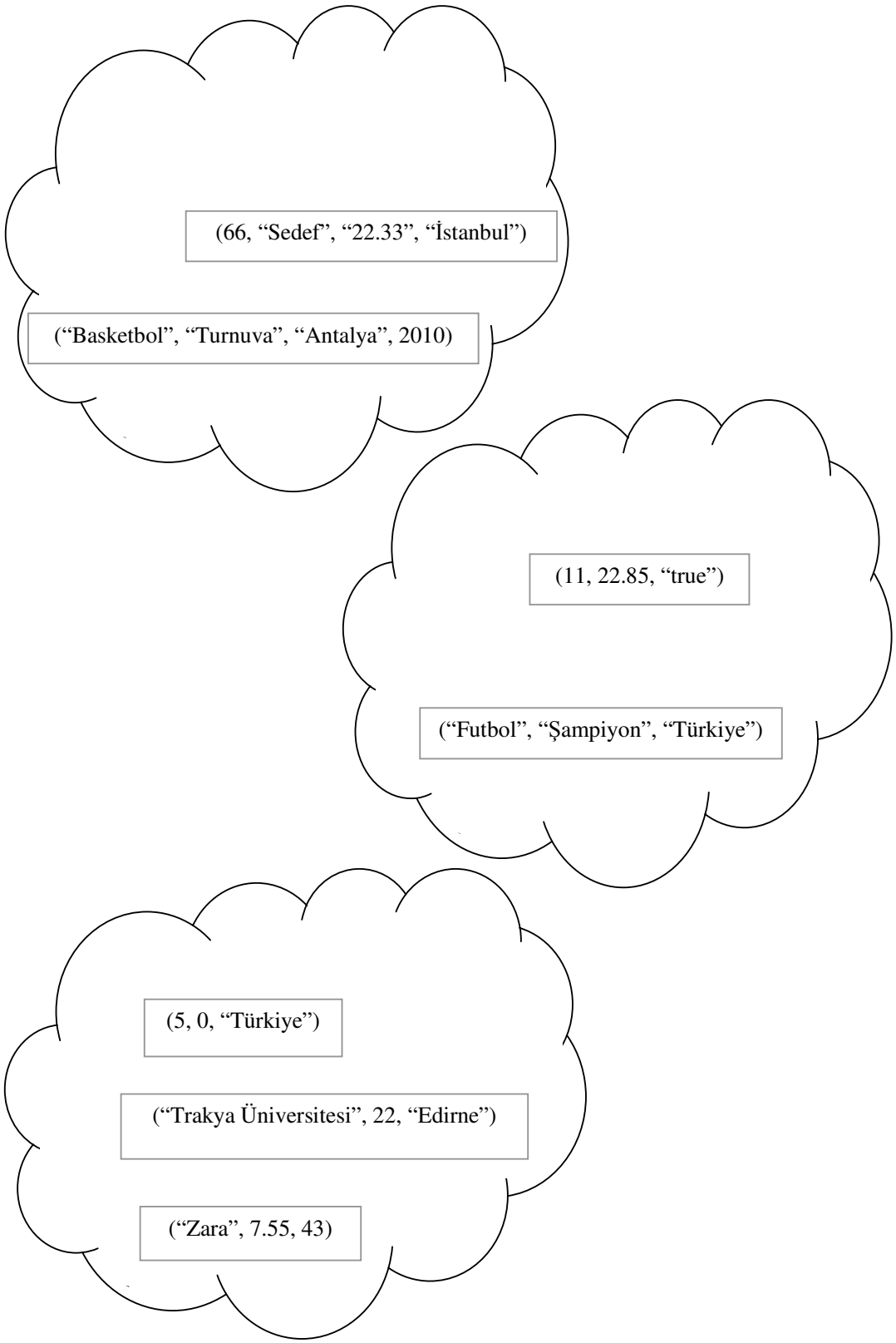
## 4.1 Linda Programlama Modeli

Gevşek bağlı (loosely-coupled) bilişim modeli ilk kez 1982 yılında David Gelernter ve arkadaşı Nicholas Carriero tarafından tanımlanmıştır [Gelernter, D., v.d.]. Bu tanımlamadan sonra çeşitli projeler ile bu yararlı model üzerinde değişik şekillerde çalışılmıştır.

Linda programlama dili olarak bilinen ve loosely-coupled modelinin gerçekleştirilmesinde kullanılan bu dil, paralel ve dağıtık bilişim dünyasına tamamen farklı bir bakış açısı getirmiştir. Linda dilinin kullanımı ile mesaj geçirme (message passing) ve paralel işlem uygulamaları, ortak bir alan üzerinden yürütülmeye başlanmıştır. Burada bahsedilen ortak alanın ismi *Tuple space* olarak adlandırılmaktadır. Tuple space'in özellikleri şu şekildedir. Tuple space bir tür dağıtık-ortak hafızadır. Tuple space'in dağıtık-ortak hafızadan farklarından ilki tuple space'te bulunan elemanlara çağrışımlı erişim (*associative lookup*) yöntemi ile erişilebilinmektedir. Tuple space'i dağıtık-ortak hafızadan ayıran diğer bir fark ise tuple space'te bulunan iki farklı okuma metodudur. Bu metotlardan biri tuple space'teki elemanı okuduğunda ortamdan silerken, diğer okuma ile içeriğini okunan elemanın ortamda bulunmasına devam etmesine izin verilmektedir.

Ortak bir alan olan tuple space, *tuple* ismindeki yapı taşlarından oluşmaktadır. Tuple space ismi de bu tuple ismindeki yapı taşlarının barındığı yer anlamına gelen tupleların alanından gelmektedir. Tuple kelimesinin okunuşu "*tu ' pul*" şeklindedir. Bir tuple'da bir veya birden fazla alan bulunup bu alanlardan her birinin bir türü bulunmaktadır. Örneğin bir tuple da bir alan olup o sayıda tamsayı türünde olabileceği gibi, başka bir tuple da yedi alan olup ilk üçü string, dördüncüsü tamsayı ve son üç alan ise gerçek sayı olabilmektedir.

Şekil 4.1'de üç farklı tuple space ile bunların içinde barındırdıkları yedi tane örnek tuple gösterilmektedir. Şekil 4.1'i incelendiğinde; en üstte yer alan tuple space'te iki tuple bulunmaktadır. Bunlardan üsteki tuple dört farklı bölümden oluşmaktadır. Bunlar sırası ile 66, Sedef, 22.33 ve İstanbul'dur.




**Şekil 4.1** Örnek üç tuple space ile içlerinde bulunan 7 tuple

Bir tuple'ın içinde barındırdığı bölümlere tip imzası (type signature) denilmektedir. Yukarıdaki örnekteki tuple'da ilk tip imzası tamsayı, ikincisi string, üçüncüsü ondalıklı sayı ve dördüncü ise string tir.

Linda programlama dilinde basit fakat etkili yalnızca birkaç işlem yapılabilmektedir. Tuple space üzerinde yapılan işlemler genel anlamda *yazma*, *okuma* ve *okuyup tuple space ten silme* olmak üzere üç gruptan oluşmaktadır. Tuple space üzerinde yapılan işlemlerin diğer önemli özelliklerinden biri, işlemlerin atomik olarak yapılması, diğeri ise tuple space'e işlemlerin eş zamanlı olarak erişerek istedikleri tupleları istedikleri zaman kullanabilmeleridir. Bu özellik sayesinde sistemdeki paralel çalışabilme özelliği arttırılmaktadır. Sistemde sıralı erişim kuralı yalnızca okuyup silme işlemi için kullanılması gerekmektedir.

Tablo 4.1'de Linda programlama dilinde kullanılan kısıtlı sayıda fakat etkili olarak sistem işleyişinin yürümesini sağlayan işlemler gösterilmektedir.

**Tablo 4.1** Linda programlama dilinde tuple space üzerinde yapılan işlemler

| **İşlem** | **Açıklama** |
|---|---|
| **out** | Tuple space içerisine tuple koyar |
| **in** | Tuple space ten bir tuple okur ve okuduğu bu tuple'ı tuple space ten siler, eğer aradığı tuple ı bulamaz ise uygun bir tuple gelene kadar bekler |
| **rd** | Tuple space ten bir tuple okur fakat okuduğu bu tuple'ı tuple space ten silmez, eğer aradığı tuple ı bulamaz ise uygun bir tuple gelene kadar bekler |
| **inp** | Tuple space ten bir tuple okur ve okuduğu bu tuple'ı tuple space ten siler, eğer aradığı tuple ı bulamaz ise bekleme yapmaz, programa geri döner |
| **rdp** | Tuple space ten bir tuple okur, eğer aradığı tuple'ı bulamaz ise bekleme yapmaz, programa geri döner |
| **eval** | Tuple space e aktif bir tuple (işlem) koyar |

Okuma (rd, rdp) ve okuyup silme (in, inp) işlemleri için çağrışımlı erişim kullanılmaktadır. Bunun gerçekleştirilebilmesi için bir şablon'a (template) ihtiyaç duyulmaktadır. Burada kullanılan şablonun belirlenmemiş alanların bulunmaktadır. Örneğin Şekil 4.1'de en üstteki tuple space te ikinci tuple'ın dört alanı bulunmaktadır. Bunlardan sadece sondakini belirtip diğerlerini genel olarak bıraktığımızda elimizde şu şekilde bir şablon oluşacaktır: (?, ?, ?, 2010). Diğer taraftan aynı tuple için ("Futbol", "Turnuva", ?, ?) şeklinde ya da tüm alanları dolu olarak ("Voleybol", "Eleme", "İstanbul", 2008) şeklinde de olabilir. Burada verilen örnek şablonların hepsi Şekil 4.1'deki ikinci tuple ile uyumlu yapıdadır. Diğer taraftan (43, "Turnuva", "Antalya", 2010) ve ("Basketbol", 13, 12, ?) şablonları ise örnek tuple'ımız ile uyumlu olmadığından onunla ilgili yapılacak işlemlerde kullanılamazlar.

**Tablo 4.2** Linda programlama dilinin avantajları

| **Özellik** | **Açıklama** |
|---|---|
| **Kullanım Kolaylığı** | Birkaç kolay ve güçlü işlem ile mantık olarak paylaşılan belleğe (tuple space) erişim sağlanır |
| **Taşınabilirlik** | Linda dağıtık hafıza ve paylaşılan bellek bilgisayar sistemleri ve ağlar gibi çeşitli paralel bilgisayar sistemlerinde kullanılabilir |
| **Gevşek-Bağlaşımlı** | Linda gevşek-bağlaşımlı iletişimi sağlayarak, kim, nerede ve ne zaman problemlerini ortadan kaldırır |
| **Türdeş-olmayan Erişim** | Linda özel bir saydamlık sağlayarak, farklı mimari ve diğer özelliklere sahip sistemlerin aynı tuple space üzerinden iletişim kurmasını sağlar |
| **Yük Dengesi** | Mantıki paylaşılan bellek ve görev çanta modelleri tasarımı yapılarak sistemde çalışan işlemlerin yük dengesi sağlanabilir |
| **Dengeli İletişim Karmaşıklığı** | İletişimin karmaşıklığı sistemde çalışan işlem ve işlemci sayısının artması ile artmaz |

Tuple'lar Noktadan-Noktaya yapılan mesaj iletişiminden daha güçlü ve etkili bir yüksek seviyeli metodtur. Ortak problem çözmede, bir tuple space ortak bülten tahtası olarak kullanılarak arama ve güncelleme işleminin iyi bir şekilde yapılmasına olanak sağlamaktadır. Bu işlemler yapılırken özel ve yerel mesaj değişimine gerek duyulmamaktadır. Sistem içinde değiştirilmesi gereken şey, çalışan sistem'in (runtime system) otomatik olarak özel bir programlama işlemi yapmaya gerek duymaksızın yapılması şeklindedir. Bu sayede işlemlerin birbiri hakkında kim, nerede, ne zaman gibi bilgiler olmaksızın çalışmasını sağlamaktadır. Bu sistem uygulama geliştirme için çok uygun bir ortam sağlamaktadır. Örneğin, bir ajanın davranışı sistemin genel durumu ile ilgiliyse, bu ilişki ajanın ortak tuple space'i incelemesi ile halledilebilir.

Tuple Space kullanmanın dezavantajlarından birisi, tek tuple space kullanılması durumunda sistem için bir dar boğaz oluşturmasıdır. Tuple space sayısı arttıkça ise sistemin tutarlılık sorunu ve performans kaybı yaşanmaktadır. Sistemin başarılı bir şekilde çalışması sağlanabilmesi için tuple space kullanıcıların tuplelara dağılımı dengeli yapılması gerekmektedir. Ayrıca, tuple space sayısı ve dağılımı da önemlidir. Böyle bir sistem daha önce geliştirdiğimiz [Atlig, C., v.d.] bir sistemde uygulanmış ve performans değerlendirmesi yapılmıştır.

### 4.2 Jini Teknolojisi ve Genel Yapısı

JavaSpaces servisi Jini teknolojisine bağlı olarak hizmet vermektedir. Bu servisin çalışabilmesi için sisteme öncelikle Jini teknolojisinin yüklü olması gerekmektedir.

Jini kelimesi kendi başına bir anlam taşımadığı gibi özel bir kısaltmada değildir. Sun Microsystems'e göre Jini, Jini Is Not Initials (Jini Baş harfler değildir) şeklinde özel bir kısalma ile özetlenebilir. Jini'nin okunuşu Türkçedeki gibi "ji-ni" olarak söylenmektedir.

Jini teknolojisinin geliştirilme amacı bilgisayar ağlarını kullanan cihazların kolay ve etkili bir şekilde çalışmasını sağlamaktır. Jini teknolojisinin sağladığı yararlarını Tablo 4.3'te görülmektedir.

Jini teknolojisinin dizayn fikri, Java programlama dilinin özelliklerinin incelenmesi sonucunda ortaya çıkmıştır. [Edwards, W.K.]. Java platformunun yapısı

sayesinde farklı bilgisayar mimarisine sahip sistemler özel bir değişiklik gerektirmeden aynı programı kullanılabilmektedir.

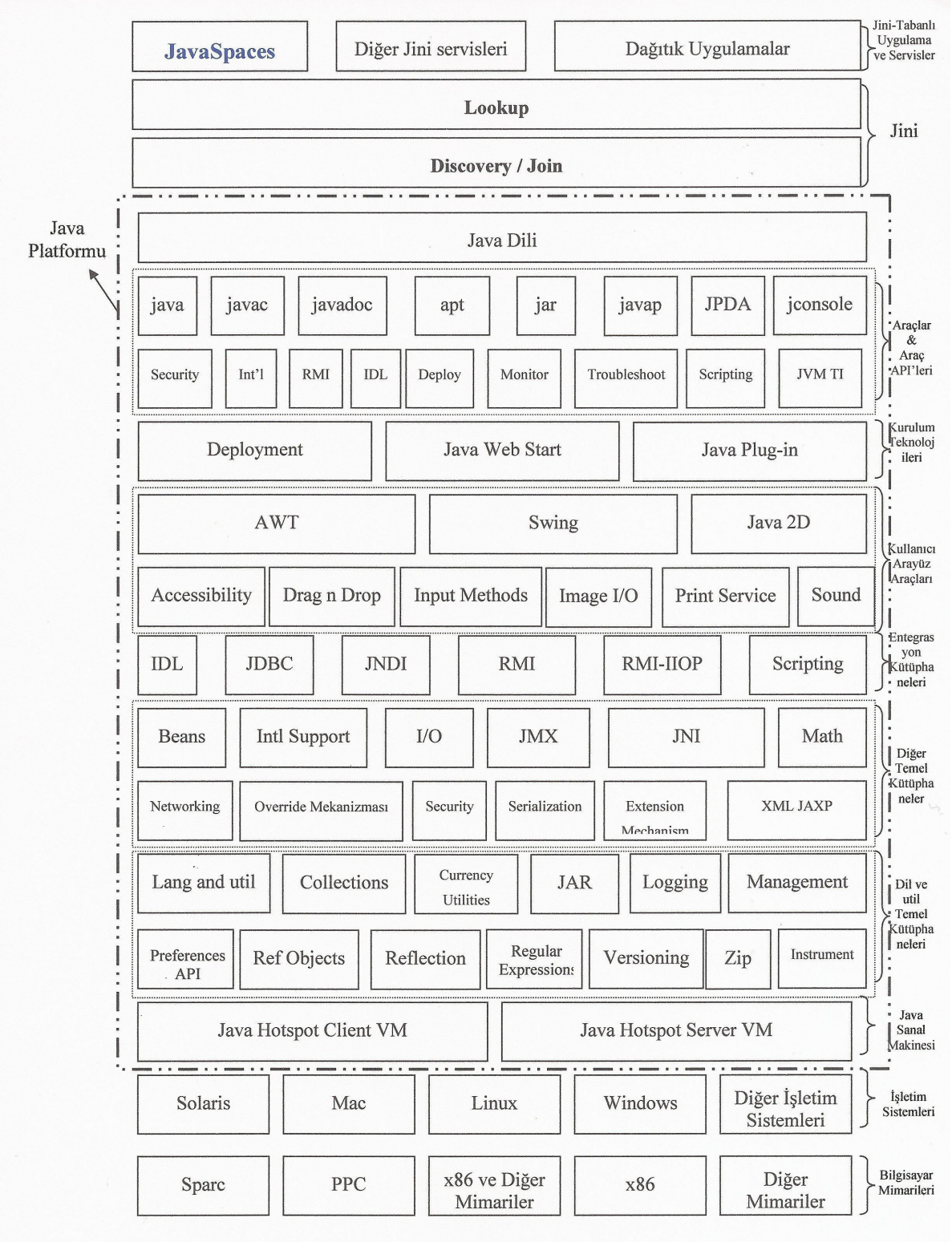
**Tablo 4.3** Jini teknolojisinin ağ uygulamalarına getirdiği avantajlar

| **Sistemin Özelliği** | **Açıklama** |
|---|---|
| **Ağdaki cihazın robust olması** | Ağa bağlı cihazların hataya karşı dayanıklıdır |
| **Tak ve Çalıştır** | Ağa bağlanan cihazların, ağa bağlanmasından ekstra başka bir ayar gerektirmeden çalışabilmelidir |
| **Ağdaki diğer cihazların kolay kullanımı** | Ağa bağlı cihazların, ağ üzerinden kullanımının en kolay bir şekilde yapılabilmesi gerekmektedir |
| **Anlık olaylara adaptasyon** | Ağa bağlı cihazlar için gelişen olaylarda oluşan uygulamalara anında adapte olabilmesi gerekmekteydi |

Java'nın bu özelliğine benzer bir şekilde, Jini teknolojisinde de farklı sistemler ve cihazlar bir ağa bağlanmaları için özel bir ayar gerekmemekte, direkt olarak ağa bağlanarak gerekli kurulum otomatik olarak gerçekleşmektedir.

Jini işleyişini sağlayan iki önemli yapı taşı vardır. Bunlardan ilki *discovery / join*, diğeri ise *lookup* tır. Bu yapı içinde SOAP, IIOP, ORPC gibi değişik protokolleri kullanabilen yapısı ile Jini protokol bağımsız bir özelliktedir. Jini teknolojisinin çeşitli servisleri bulunmaktadır. Bunlardan en önemlisi JavaSpaces servisidir. Jini teknolojisinin üzerinde çalışan diğer Jini servisleri bulunmakta, ayrıca dağıtık uygulamalar da Jini teknolojisi üzerinden faydalanmaktadır. Şekil 4.2'de Jini teknolojisinin Java platformundaki yerini ve JavaSpaces servisinin bu yapılanmada nerede yer aldığı görülmektedir.

Şekil 4.2 en alttan incelenmeye başlandığında piyasada bulunan birbirinden farklı bilgisayar mimarileri görülmektedir. Bu sistemlere örnek olarak SUN firmasının ürettiği Sparc işlemci sistemi ve PC'lerde kullanılan x86 işlemci sistemleri örnek olarak verilebilmektedir.




**Şekil 4.2** Jini teknolojisinin Java Platformundaki yeri ve JavaSpaces servisinin bu mimarideki konumu

Bir üst kademeye geçtiğimizde kullanılan işletim sistemleri karşımıza çıkmaktadır. Sparc mimarisini kullanan bir bilgisayar sistemi Solaris mimarisini kullanırken, x86 mimarisine sahip bir bilgisayar sistemi Windows ya da Linux işletim sistemini kullanmaktadır. Bunların üzerindeki toplam yedi katman Java platformunu oluşturmaktadır. Bunlar aşağıdan sırası ile Java Sanal Makinesi, Dil ve util Temel Kütüphaneleri, Diğer Temel Kütüphaneler, Entegrasyon Kütüphaneleri, Kullanıcı Arayüz Araçları, Kurulum Teknolojileri, ve Araç ve Araç API'leridir.

Yedi ana bölümden oluşan Java platformunun üzerinde Jini teknolojisi bulunmaktadır. Daha öncede bahsettiğimiz gibi Jini iki kademeli olup birinci katmanda Discovery / Join gelmekte, bunun üzerinde ise Lookup sistemi bulunmaktadır. İki katmanlı olan Jini'nin üzerinde ise Jininin servisleri ve dağıtık uygulamalar bulunmaktadır.

### 4.3 JavaSpaces Servisi ve Temel Özellikleri

JavaSpaces, Sun Microsystems tarafından geliştirilmiş olan Jini teknolojisinin bir servisi olarak hizmet vermektedir. Bu servis sayesinde dağıtık sistemlerde birçok işlem kolaylıkla yapılabilmektedir.

Genel olarak JavaSpaces servisi Şekil 4.3'te gösterilmiştir. Şekilde dört tane işlem, beş farklı JavaSpaces servisi üzerinde işlem yaparken görülmektedir. Örneğin İşlem 1, birinci JavaSpaces servisine write işlemini kullanarak nesne koymaktadır. Diğer taraftan İşlem 4, dördüncü JavaSpaces servisinden bir nesneyi take işlemi ile alırken, üçüncü JavaSpaces servisine ise write işlemi ile bir nesne koymaktadır.

JavaSpaces servisi içerisinde bulunan nesnelerin özel bir ismi vardır. "*Entry*" şeklinde adlandırılan bu nesneler, Linda dilindeki tuple larda olduğu gibi çeşitli alanlara sahiptir. Buradaki alanların her birine öznitelik (attribute) denmektedir. Bu öznitelikler Linda dilindeki tuplelardaki alanlardan daha fazla çeşit seçeneğine sahiptir. JavaSpaces'te kullanılan entryler program kodu transfer etme işinde de kullanılırlar. Ayrıca JavaSpaces'teki entrylerin her bir özniteliği kendi başına bir nesnedir ve bu şekilde her türlü veri nesne için kullanılabilmektedir.

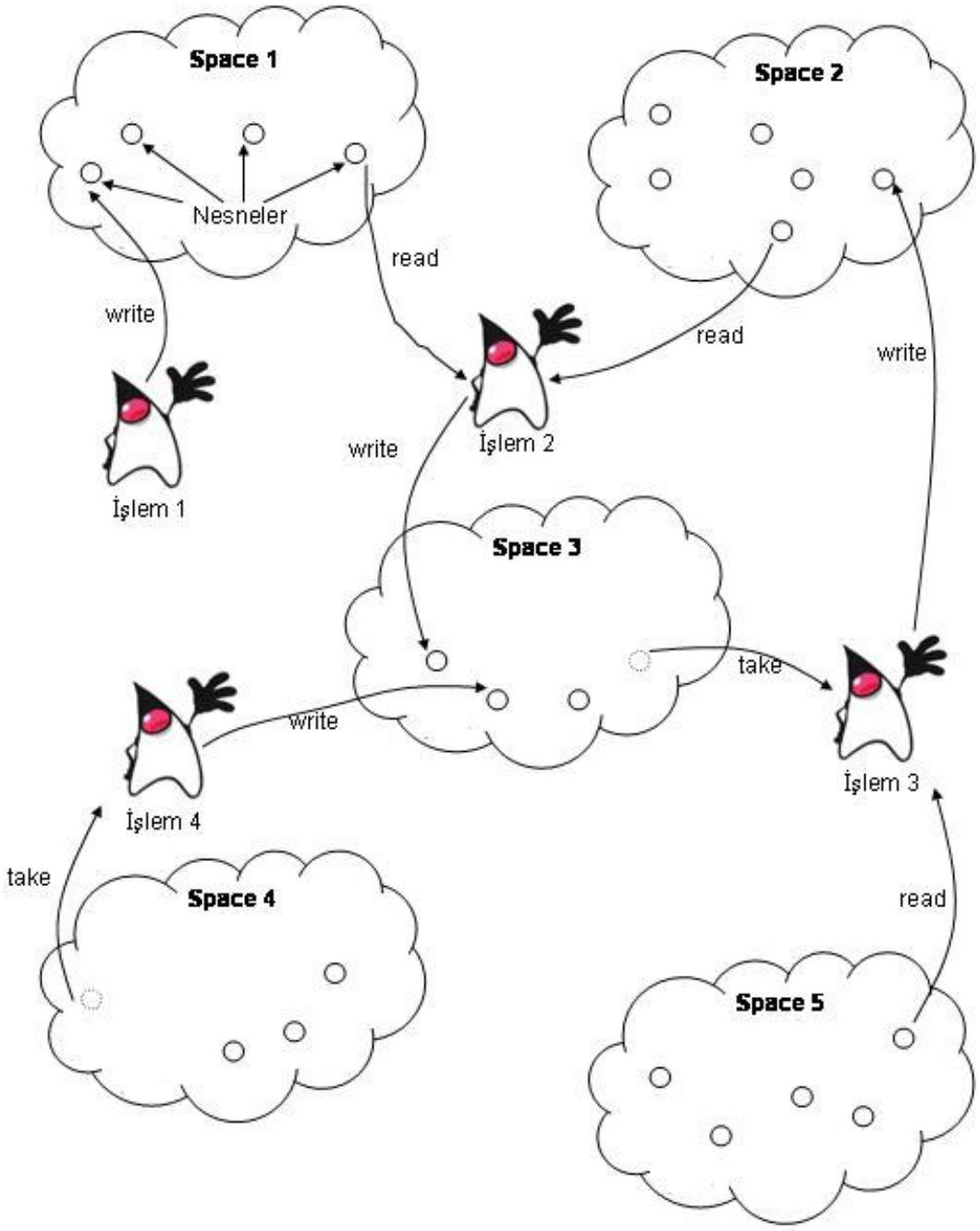
JavaSpaces servisini üzerinde yapılan ardı ardına yapılan işlemlerde, Jini teknolojisinin sağladığı transaction özelliğinden yararlanılabilir. Buradaki transaction

özelliği sayesinde arka arkaya olan birkaç farklı işlemin hepsinin yapılması ya da hiçbirinin yapılmaması garanti edilebilmektedir. Örneğin bir işlem, JavaSpaces servisinden take işlemi sonucu aldığı entry'yi, write işlemi kullanarak JavaSpaces servisine yazmasını garanti edebilir. Bu işlem sayesinde yapılan işlemlerin *atomic*'liyi garanti edilmiş olmaktadır.

JavaSpaces servisinde, Şekil 4.3'te görüldüğü gibi birden çok JavaSpaces bulunabilmektedir. Bu durumda geleneksel Linda sistemlerinden daha iyi bir özelliktir. Uygun bir yapılandırma ile bir veya daha fazla sayıda JavaSpaces servisi kullanılması sistemin esnekliğini ve yeterliliğini arttırmaktadır.

JavaSpaces servisi kullanan işlemlerin bu servisi kullanmaları sırasında birçok avantajları bulunmaktadır. Bunlardan biri karşılıklı olarak birbirlerinin kimlik ayrıntılarını bilme gereği duymamalarıdır. Geleneksel işlemler-arası iletişiminde (IPC) ise karşılıklı iletişim kuran iki işlem birbirlerinin tüm detaylarını bilmek ieltişim açısından zorunluluktur. Örneğin iki işlem birbirinin IP numaralarını ve Port numaralarını öğrenmek zorundadır. JavaSpaces servisinde böyle bir zorunluluk olmaması büyük bir avantajdır. JavaSpaces servisinde işlemlerin iletişimin sağlanabilmesi için, aynı JavaSpaces servisine bağlanmaları yeterlidir. Bunlara ilaveten, işlemlerin JavaSpaces servisine daha kolay bağlantı sağlayabilmeleri için yerel ağda bulunan JavaSpaces servisine kolay bağlantı sağlanabilmesi için SpaceAccessor sınıfının getSpace metodu aracılığı ile kolay bir şekilde JavaSpace servisine ulaşılabilmektedir.

JavaSpaces servisinin sağladığı diğer bir avantaj bu sistemin işlemler-arası veri paylaşımı için esnek bir ortam yaratmasıdır. JavaSpaces servisi kullanıldığında iletişim kuran işlemlerin aynı anda çalışmaları gerekmemekte, bilgi iletmek isteyen işlem bilgisini aktardıktan sonra sistemdeki etkinliğine ara verebilmektedir. Örneğin, saat beş itibarı ile çalışmaya başlayan bir işlem iletmek istediği bilgiyi hazırladıktan sonra JavaSpaces servisine yüklerse, ister kendisi çalışmaya devam etsin isterse etmesin gönderilen bu mesajın alıcısı çalışmaya başladığı zaman mesaj orada onu beklemektedir.




**Şekil 4.3** Dört işlemin beş JavaSpaces servisini kullanımı. İşlemler farklı JavaSpaces servislerini uygun komutlar aracılığı ile kullanıyorlar

İletişim sırasında JavaSpaces üzerinden bilgi alacak işlemler aradıkları bilgi JavaSpace servisi üzerinde bulunuyorsa hemen alabildikleri gibi, eğer o an için serviste yok ise gelene kadar bekleyebilmekte yada bekleme için belli bir süre ayırabilmektedirler.

JavaSpaces servisinin kullanılmasını popüler yapan üç avantaj, "*kim*", "*nerede*" ve "*ne zaman*" şekli ile bilinen ve dağıtık sistemdeki klasik yöntemler ile iletişim kurmak isteyen işlemlerin bilmesi gereken bilgilerin JavaSpaces servisinin kullanılması durumunda ihtiyaç olunmamasıdır [Freeman, E., v.d.].

Bu bağlamda gönderilen bir mesajın alıcısının bilinmesine yada gönderilmiş olan bir mesajın göndericisinin bilinmesine gerek olmaması sistemin "kim" açısından esnekliğe sahip olmasını sağlamaktadır. Bu şekilde, belli bir kişi yerine "*herhangi biri*" gibi esnek bir yöntem kullanılabilmektedir. Bu sayede birbiri hakkında bilgi sahibi olmayan işlemler bile iletişim kurabilmektedir. JavaSpaces servisini kullanacak işlemler iletişim sırasında eğer birbirinden haberdar olmak istiyorlar ise bunu da mesajlara ekleyecekleri bilgiler ile gerçekleştirebilmektedirler.

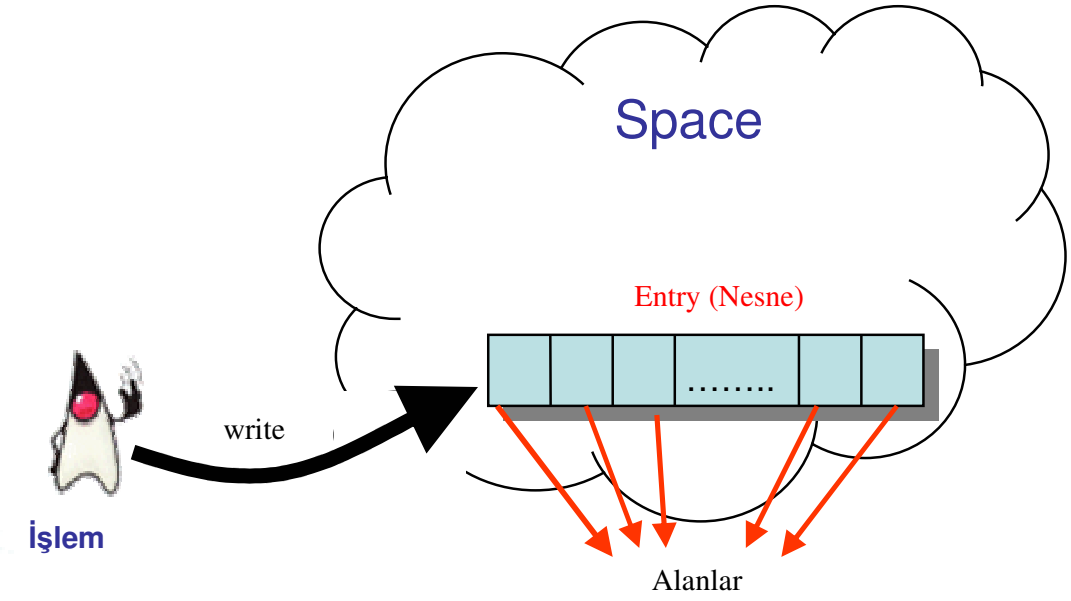
JavaSpaces servisinin sağladığı "nerede" konusundaki avantajı "*herhangi bir yerde*" sayesinde iletişim kuran işlemlerin o anda nerede olduğunun bilinmesine gerek yoktur. Bu durumda, iki işlem ister aynı makinede bulunsun, ister aynı yerel ağda yer alan farklı bilgisayarlarda çalışsınlar, isterlerse de geniş alan ağı üzerinden (WAN), birbirleri ile iletişim kurmak istesinler, bilmeleri gereken tek şey gerekli JavaSpaces servisine ulaşılmasıdır. Daha önce bahsettiğimiz gibi JavaSpaces servisine ulaşma sırasında da SpaceAccessor sınıfının getSpace metodu sayesinde JavaSpaces servisinin bulunması işlemi kolaylaştırılmıştır.

Klasik işlemler-arası iletişim (IPC) yöntemlerinde olmayan "Ne zaman" avantajı, JavaSpaces servisi ile "*herhangi bir zaman*" gerçekleştirilmektedir. Bu durumda, iletişim kuracak bir işlem karşı taraftaki işlemin sistemde olup olmamasına bakmaksızın iletmek istediği bilgiyi JavaSpaces servisine gönderir daha sonra bu mesajın alıcısı ister derhal isterse günler sonra bu mesajı JavaSpaces servisinden alabilmektedir.

### 4.4 JavaSpaces Servisinde Yapılan Operasyonların Anlambilimsel Yapısı

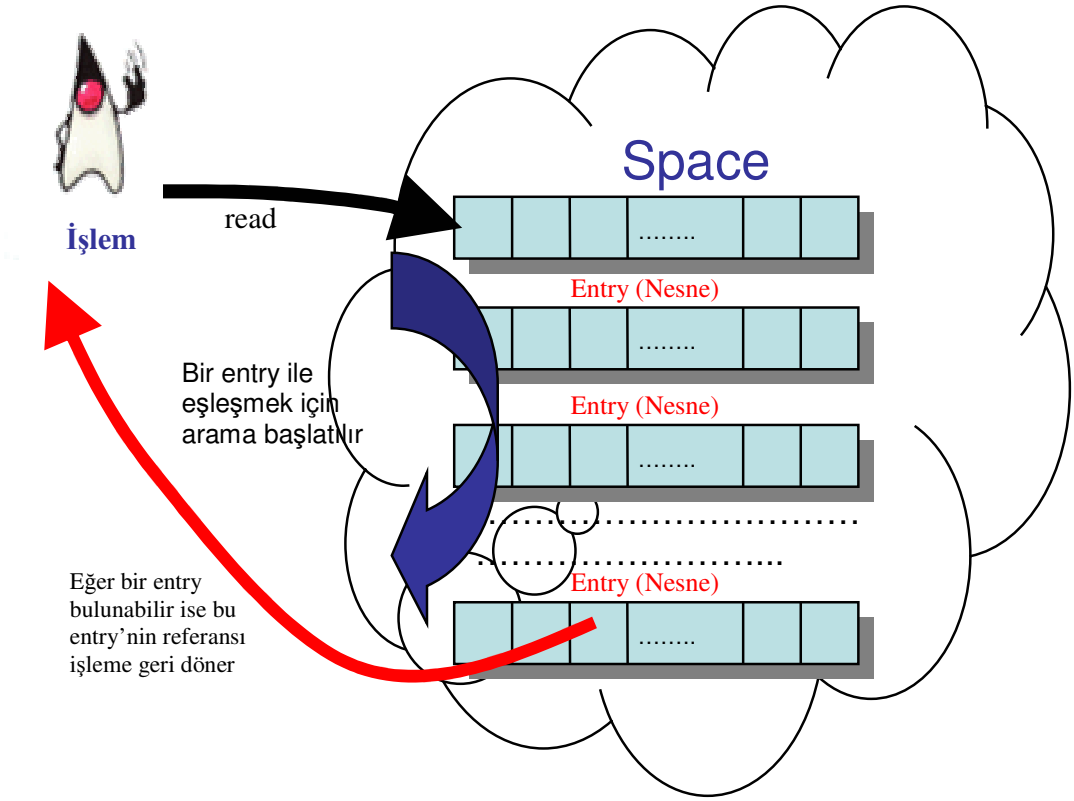
JavaSpaces servisi sınırlı sayıda işlem ile kullanılmaktadır ancak bu işlemler ile başarılı sonuçlar elde edilmektedir. Bu işlemlerden üç tanesinin ana işlerin yürümesinde kullanılmakta, toplamda yedi tane işlem bulunmaktadır. Bu işlemlere ilave olarak, JavaSpaces servisinin bu tez yazılırken son sürümü olan "Outrigger v2.1" de özellikle grup işlemlerde faydalanılmak üzere birkaç ek komut eklenmiştir.

Ana işlerin yürümesini sağlayan işlemlerden biri olan "write" işlemi JavaSpaces servisine bir şey eklemek için kullanılır. Bu işlemin işleyiş şekli şu şekildedir. JavaSpaces servisine yazılacak mesaj write işlemini kullanacak işlem tarafından hazırlanır. Daha önce bahsedildiği gibi JavaSpaces servisinde bulunan nesnelere *entry* denilmektedir. Her bir entry'nin *lease* isminde bir tür yaşam süresi vardır. Bu yaşam süresi entry'nin JavaSpaces servisinde yaşayacağı süreyi belirler. JavaSpaces servisinin bir diğer özelliği de *transaction* sistemi kullanılarak birkaç JavaSpaces işleminin gruplanabilmesidir. Write işleminde de eğer işlem gerek duyuyorsa transaction özelliğinden yararlanabilir. write işlemi tamamlandığında, bu işlemi çağıran işleme bir lease değeri döndürür. Bu değer yapılan write işleminde talep edilen lease değerine eşit veya daha küçük bir değerdir. write komutunun çalışma şekli, şekil 4.4'te gösterilmektedir.




**Şekil 4.4** JavaSpaces servisinde *write* komutunun çalışma şekli

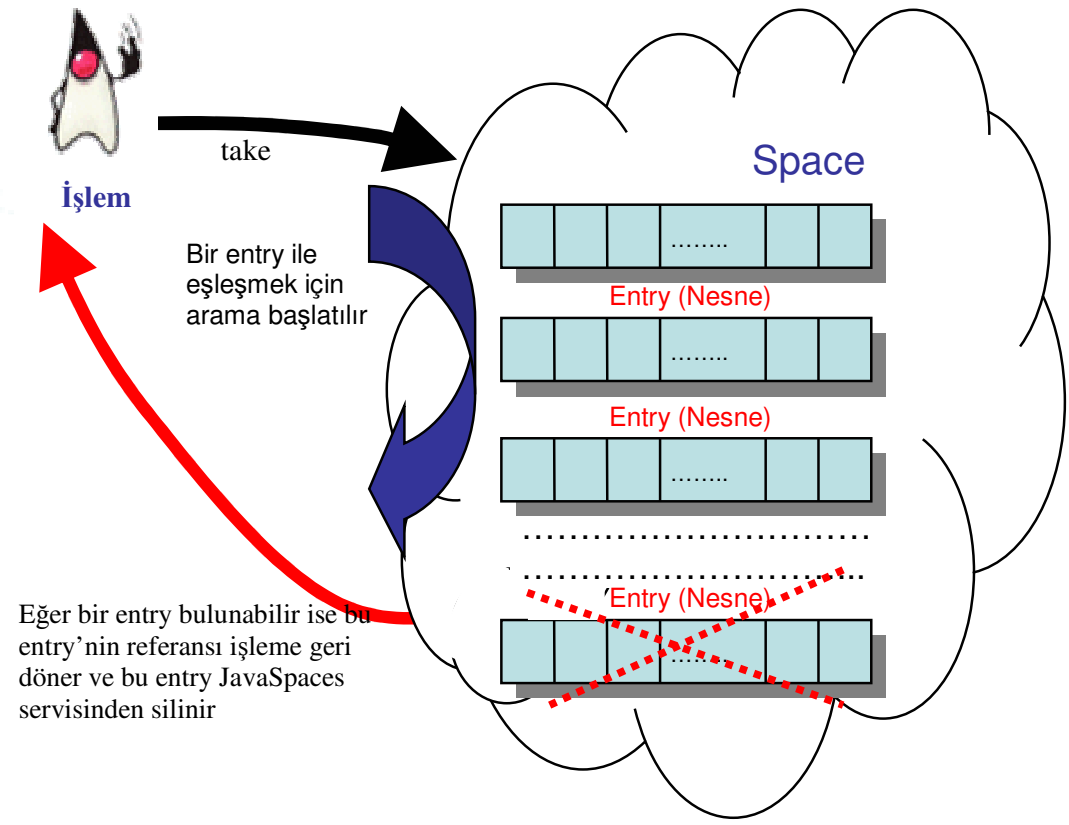
"read" işlemi JavaSpaces servisinde ana işlerin yürümesini sağlayan bir diğer komuttur. Bu komut ile belli bir entry'nin JavaSpaces servisinde okunması gerçekleştirilmektedir. JavaSpaces servisinden bir entry okunabilmesi için *template* isminde entry ile eşleşmede kullanılacak bir taslak hazırlanmalıdır. Burada template'in bütün alanları dolu olabileceği gibi, bazı alanları tam olarak belirtilmeden de bırakılabilir. Tüm alanları belirtilmiş bir template kullanılması durumunda JavaSpaces servisinden okunacak entry bellidir. Diğer taraftan sadece birkaç alanı belirtilmiş bir template kullanılması durumunda JavaSpaces servisinden bu template formatına uygun olup sadece belirtilen alanlar ile uyuşma gösteren herhangi bir entry okunmaktadır. read işleminde de transaction kullanılabilmektedir. Dolayısı ile read komutunu kullanacak işlemin transaction'a ihtiyacı varsa bunu belirtmesi gerekir. read işlemindeki diğer bir nokta ise eğer JavaSpaces servisinden okunmak istenen entry hazır değil ise maksimum bekleme miktarıdır. read komutu çalıştırıldığın da eğer uygun bir entry bulunamaz ise, bu süre boyunca bekleme yapılır. Şekil 4.5'te read komutunun çalışma şekli gösterilmektedir.




**Şekil 4.5** JavaSpaces servisinde *read* komutunun çalışma şekli

Ana JavaSpaces servis komutlarından üçüncüsü "take" komutudur. Bu komut ile JavaSpaces servisinden okuma işlemi yapılır ancak read işlemine benzer şekilde çalışan bu komut sonucunda okunan entry JavaSpaces servisinden silinir. Take komutunun çalışma modeli Şekil 4.6'da gösterilmektedir.

Yukarıda bahsedilen üç ana komuttan başka dört tane de daha az kullanılan diye adlandırılabilecek komut bulunmaktadır.




**Şekil 4.6** JavaSpaces servisinde *take* komutunun çalışma şekli

Daha az kullanılan komutlardan birincisi "readIfExists" komutudur. Bu komutun genel olarak işleyişi read komutuna benzemektedir. Dolayısı ile okuma için kullanılan bu komutun diğerinden farklı olarak eğer aranılan entry JavaSpaces servisinde bulunamaz ise işleme hemen *null* değeri geri gönderilir.

Diğer bir az kullanılan komut ise "takeIfExists" komutudur. Bu işlem de çalışma fikri ile readIfExists'e benzemektedir. Aranılan entry JavaSpaces servisinde bulunamaz ise hemen işleme null değeri geri gönderilir. Eğer herhangi bir entry bulunur ise take komutunun yaptığı şekilde bulunan bu entry alınır.

"notify" komutu da sıkça kullanılmayan bir JavaSpaces servisi komutudur. Bu komutu kullanan işlem gelecekte JavaSpaces servisine yazılacak bir entry'den haberdar

olmak için bu komutu kullanır. read komutundaki şekli ile aranan entry'nin uyuşması sağlanmaktadır. Uyuşan bir entry bulunduğu zaman işleme uyarı işlemi yapılır.

"snapshot" komutu JavaSpaces servisinde kullanılan yedinci komut olarak karşımıza çıkmaktadır. JavaSpaces servisinin işleyişinde Java Serialization sistemi kullanılmaktadır. snapshot komutu kullanılarak, aynı entrynin tekrar tekrar JavaSpaces servisinde kullanılması durumunda serialization işleminin yalnızca bir kere yapılarak zaman kazanılması sağlanmaktadır.

Yukarıda bahsedildiği gibi JavaSpaces servisinin 2.1'inci sürümünde birkaç yeni komut daha kullanıma sunulmuştur. Bu komutlardan JavaSpace sınıfını genişleterek (extends) kullanıma sunulan komutlar şu şekildedir.

Yeni "write" komutu tek bir çağrı ile çok entry kopyalarını space'e yüklenmesini sağlamaktadır. Bu komut sayesinde JavaSpaces servisine yazılan bir grup entrynin her biri için bir lease süre listesi geri gönderilir.

Yeni "take" komutu JavaSpaces servisinden bir veya daha fazla sayıda uyuşan entry alma işlemi gerçekleştirir. Burada JavaSpaces servisinden alınacak entrylerin uyuşma yapacağı template sayısı bir veya birden fazla olmaktadır. Bu templatelerden herhangi birine uyuşma sağlanması bu yeni take metodunun JavaSpaces servisinden entry alması için yeterlidir.

"contents" komutu da yeni eklenen komutlardan birisidir. Bu komut ile belli bir lease zamanı süresinde JavaSpaces'te uyuşma sağlanan bütün entrylerin okunması şeklinde çalışmaktadır.

Diğer bir yeni komut olan "registerForAvailabilityEvent" işleminin JavaSpaces servisinde şu şekilde çalışmaktadır. JavaSpaces servisi üzerinde registerForAvailabilityEvent komutu çalıştırıldığı anda bulunmayan, ancak daha sonraki gelişmeler sonucunda bulunur hale gelecek olan entrylerin haber verilmesine yaramaktadır. Örneğin transaction altında çalışmakta olan bir grup komutun bir tanesi ile bir veya birkaç entry take yani alınmaya çalışılıyorsa, bu transactionın başarı ile tamamlanamaması durumunda take olmaya çalışılan entry bulunmayan durumdan, bulunabilir duruma geçmektedir.

### 4.5 Ajan Tabanlı Sistemler ve Gevşek Bağlı Model ile Uygulanması

Bilgisayar sistemlerinde ajan kullanımı önemli avantajlar sağlamaktadır. Bu bağlamda gerek ticari gerekse araştırma projesi olarak ajan kullanan çeşitli sistemler geliştirilmiştir [Roth, V.], [Braun, P., v.d.], [Pinsdorf, P.,] ve [Peters, J., v.d.]. Örneğin, online alışverişlerde tavsiyede bulunmak üzere geliştirilen ajanlar yada ortamda bulunan çevresel faktörlere göre (sıcaklık, nem, güneş ışığı) sistemde gerekli ayarlamaları yapan özellikte ajanlar bulunmaktadır.

Ajan olarak bahsedilen yapılar yalnızca yazılım parçalarından oluşmamakta, aynı zamanda hayatta kendi başına hareket edebilen bilgisayar ve ağ özelliği olan cihazları da kapsamaktadır. Örneğin, bilişim özelliği olan buzdolabı, uçak, gerekli yerler ile iletişim halindeki uydu anten sistemi bunlara örnek bazı ajanlar olarak karşımıza çıkmaktadır.

Geleneksel olarak ajanlar kendi aralarında iletişimi mesaj geçirme (message passing) yöntemi ile yaparlar. Ajanlı programların performansı mesaj geçirme yöntemini kullanmalarından dolayı düşmektedir. Internode iletişim yerine intranode iletişim ile sistemin performansını arttırmak mümkündür. Örneğin, [Jang, M.-W., v.d.] çalışmasında büyük-ölçekli ajan tabanlı simülasyonlar için 2 çeşit dinamik ajan dağıtım modeli önerilmiştir. Bunlardan birincisi ajanların iletişim maliyetini azaltırken, diğeri ağır yüklü ajan platformundan daha hafif yüklü ajan platformuna aktarılmaktadır.

Coarse-grain iletişim ilişkisi kullanılan bu sistemde, bu iletişim yöntemi sayesinde ölçeklenebilirlik seviyesi yükseltilmektedir. Açık sistemlerde alıcının adresi gönderici tarafından bilinmediği için sıkı bağlı (tightly-coupled) sistemlerde problem yaşanmaktadır. Ayrıca, mesajın alıcısı, alıcının alacağı duruma veya özelliğe göre değişebilmektedir. Bu durumlarda gevşek bağlı model kullanımı ile sisteme esneklik sağlayan yapı kullanılmasında yarar vardır. Bu durumda yapılan işlem, göndericinin yolladığı mesajı aynı özellikteki bir grup potansiyel alıcıya göndermesidir.

Ajanlar arası koordinasyonun [Cabri, G., v.d. 2]'te belirtildiği şekilde dört çeşidi bulunmaktadır. Bunlar sırası ile zaman olarak aynı anda çalışan ve iletişim için birbirlerini tanımak zorunda olan sistemler, zaman olarak aynı anda çalışan fakat iletişim için birbirlerini tanımak zorunda olmayan sistemler, zaman olarak aynı anda çalışmasına gerek olmayan ancak birbirlerini tanımadan iletişim kurabilecek sistemler

ve son olarak iletişimde bulunmak için hem zaman olarak aynı anda çalışmasına hemde birbirlerini tanımalarına gerek olmayan sistemlerdir.

Mobil ajanlar ile tuple space koordinasyon modeli birçok projede uygulanmıştır. Bu projelerden bazıları [Ambit], [D'Agents], [MARS] ve [Cabri, G., v.d.], [PageSpace], [SOMA], ve [TuCSoN] ve [TuCSoN, www] dır.

# 5. MOBİL VE KABLOSUZ SİSTEMLERDE AJAN TABANLI VE YÜKSEK PERFORMANSLI EŞDÜZEY İLETİŞİM İÇİN GEVŞEK BAĞLI MODEL KULLANAN SİSTEMİN DİZAYNI VE UYGULAMASI

Bu bölüme kadar olan İkinci, Üçüncü ve Dördüncü bölümlerde bilgi erişilebilirliği özellikleri, mobil ve kablosuz sistemler ve bilgi erişilebilirliği yüksek sistemlerin geliştirilmesinde gevşek bağlı (loosely-coupled) modelinin kullanımı konularından bahsedilmiştir.

Bu bölümde yukarıda bahsi geçen bu konular ışığında deneylerde kullanılan ajan tabanlı eşdüzey iletişim için gevşek bağlı model kullanan sistemin tasarımı ve uygulaması gösterilmektedir.

## 5.1 Varsayımlar

Bu tezde bilgisayarlar ile ilgili şu varsayımlar bulunmaktadır. İlk varsayım olarak sistemde çalışan bütün processlerin hata ile karşılaşmaları durumunda durmaları, bu bağlamda eğer bir processin çalışmasında hata olursa, bu processin çalışmasının durması ve yanlış bilgi üretmemesi varsayılmaktadır. Diğer bir değiş ile Byzantine türü hatalar bu çalışmada kapsam dışı tutulmuştur.

İkinci varsayıma göre sistemde çalışan ajanların hataya uğraması ile o ajanın bağlı olduğu JavaSpaces servisinin hataya uğraması arasında bir ayrım yapılmamaktadır. Bu yüzden ajanın hataya uğraması yada ajan ile aynı bilgisayarda çalışan JavaSpaces'ın hataya uğraması durumunda o ajan hataya uğramış kabul edilmektedir.

Üçüncü olarak ağ ortamı ile ilgili varsayım bulunmaktadır. Bu varsayıma göre eğer sistemin bağlı olduğu ağda bölünme gerçekleşirse sistemin normal işleyişini sağlamak mümkün olmamaktadır. Bunun sebebi sistemin gevşek bağlı modeli kullanan JavaSpaces servisi üzerinden çalışmasıdır. Sistem yerel ağda çalışsada, ağın bölünme olasılığı bulunmaktadır. Bu varsayımda tasarlanan sistem üzerinde yapılan işlemler sırasındaki göz önünde bulundurulmalıdır.

## 5.2 Mobil ve Kablosuz Sistemlerde Ajan Tabanlı ve Yüksek Performanslı Eşdüzey İletişim için Gevşek Bağlı Model Kullanan Sistemin Modeli

Gerçekleştirilen sistem beş eleman grubundan oluşmaktadır. Bu elemanlar sistemde şu şekilde yer almaktadır. Sistemdeki ilk eleman grubu kablosuz cihazlarda çalışmakta olan peerlardır. Bu peerlar ellerinde bulundurdukları bilgileri diğer peerlar ile paylaşma, dolayısı ile gönderip alma özelliklerine sahiptirler. Bu sistemde bahsedilen peerlar, potansiyel olarak düşük band genişliği olan, ağ bağlantısı kurmaları sırasında sabit sistemlere göre daha fazla hata ile karşılaşan, işlemci özelliği daha kısıtlı, hafıza kapasiteleri daha sınırlı ve enerji kaynakları sınırlı miktardaki bilgisayar sistemlerinde çalışmaktadır. Bu kısıtlı özelliklere sahip kablosuz bağlantı kurma özelliği olan bilgisayar sistemleri, önerilen sistemde ikinci eleman grubu olarak bulunmaktadır.

Bu kısıtlı özelliklere sahip bilgisayar sistemlerinde bulunan eşdüzey (peer-to-peer) işlemlerin bu sınırlı özelliğe sahip cihazlarda çalışan processler tarafından yürütülmesi durumunda önemli dezavantajlar oluşacaktır. Örneğin, işlemci kapasitesi genel olarak daha zayıf olan bu sistemlerin, gelen yada giden isteklere cevap vermesi sistem için normal bir bilgisayar sisteminden daha fazla yüke sebep olmaktadır. Ayrıca, kısıtlı enerjisi bulunan mobil ve kablosuz cihazların yapacağı her ek işlem onlara çok ciddi enerji yükü getirecektir. Örneğin 10 ila 100 metrelik bir mesafeye 1 bitlik bir kablosuz yayın yapmak için mobil cihazın binlerce hatta milyonlarca aritmetik işlem yapabilecek miktarda enerjiyi harcaması gerekmektedir [Yoo, Y., v.d.].

Bu bağlamda kısıtlı özellikleri olan mobil ve kablosuz bilgisayarlarda çalışan peerların yürütecek oldukları eşdüzey işlemlerin yükünün azaltılması bu cihazlar için bir avantaj oluşturacaktır.

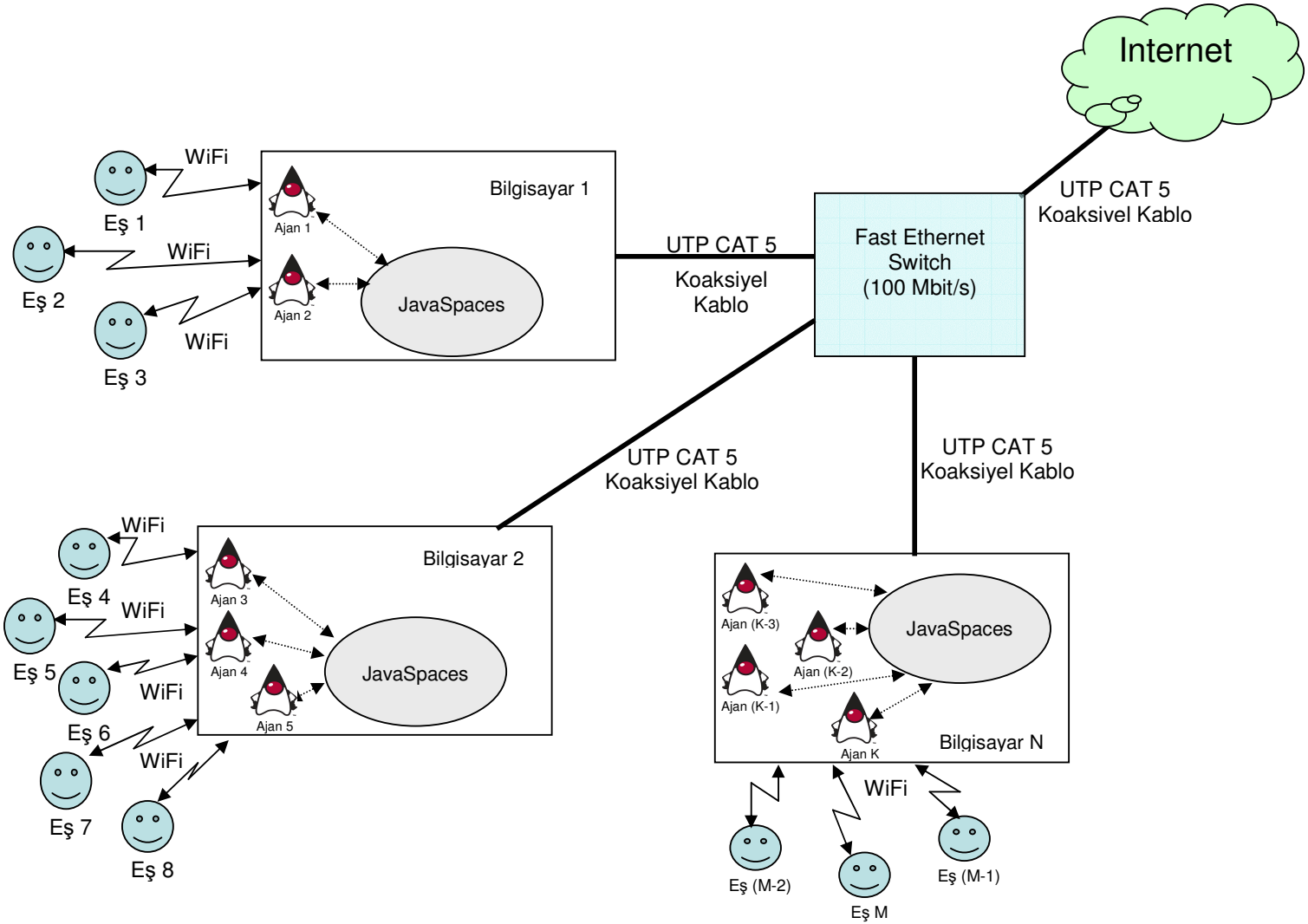
Önerilen bu sistemde bilgi paylaşımında bulunacak processlerin birçok işlemlerinin yürütülmesi ajanlara (agents) bırakılmıştır. Önerilen sistemin üçüncü elemanı olan ajanlar, mobil ve kablosuz cihazlarda çalışmakta olan peerlar ile sürekli iletişim içinde çalışmaktadır. Bilgi paylaşımında bulunacak olan eş belli bir ajan ile iletişime geçer, elinde paylaşmak istediği bilgilerin listesini gönderir. Ajanın görevi peerın eşdüzey sistemden hizmet almak istediği sürece işlemleri process adına yürütmesidir. Bu duruma göre ajanlar şu işlemleri yapmaktadırlar. İlk olarak peerlardan eşdüzey sistemine dahil olma mesajı alırlar. İkinci olarak, sistemden ayrılmak isteyen

peerlardan sistemden ayrılma mesajı alırlar. Ajanın üçüncü görevi bilgi aramak isteyen peerın bu talebini alıp yerine getirmektir. Dördüncü görev olarak ajan, bilgi yüklemek (upload) isteyen peerın yüklemek istediği dosyayı sisteme yükler. Ajanın beşinci görevi bilgi indirme (download) işlemi yapmak isteyen eş adına bu bilgilerin araştırmasını yürütmektir.

Önerilen sistemdeki dördüncü eleman JavaSpaces servisidir. Bu servis sayesinde ajanlar gevşek bağlı modeli kullanarak birbirleri ile iletişim kurabilirler. Sistemdeki kullanılan modele göre her sabit bilgisayarda bir tane JavaSpaces servisi yer almaktadır.

Önerilen sistemin beşinci ve son elemanı sabit ağ bağlantısına sahip ve peerların çalışmakta olduğu bilgisayar sistemlerine göre daha gelişmiş kapasiteye sahip merkezi bilgisayarlardır. Bu bağlamda bu bilgisayarlarda ajanlar ve JavaSpaces servisleri bulunmaktadır.

Bu bölümde anlatılan, önerilmiş ajan tabanlı ve yüksek performanslı eşdüzey iletişim için gevşek bağlı model kullanan sistem, Şekil 5.1'de gösterilmektedir.




**Şekil 5.1** Mobil ve kablosuz sistemler için önerilen ajan tabanlı ve yüksek performanslı eşdüzey iletişim için gevşek bağlı model kullanan sistem mimarisi

## 5.3 Önerilen Sistemin Anlambilimsel Yapısı

Bu bölümde tasarlanan ajan tabanlı eşdüzey sisteminin anlambilimsel yapısı açıklanmıştır. Önerilen sistemde birkaç temel işlem yer almaktadır. Bunlardan birincisi peerın *sisteme dahil olma* işlemidir. Bu işlem ile eş eşdüzey sisteme dahil olarak bilgi paylaşımı sistemine dahil olmak istediğini belirtir.

Sistemde yer alan ikinci işlem, peerın *sistemden ayrılma* işlemidir. Bu işlem ile eş eşdüzey sistemden ayrılarak bilgi paylaşımı hizmetlerinden artık yararlanmayacağını belli eder.

Sistemdeki üçüncü işlem *bilgi arama isteği* işlemidir. Bu işlem, ile bilgi aramak isteyen peerın ajana bu talebi iletmesi şekli ile gerçekleşir.

Dördüncü işlem olarak eşdüzey *sisteme bilgi ekleme* işlemi bulunmaktadır. Bu işlem sayesinde, bu işlemi gerçekleştiren peerın elinde bulundurduğu bilgi sistemde diğer peerlar tarafından erişime açık hale gelmiş olmaktadır.

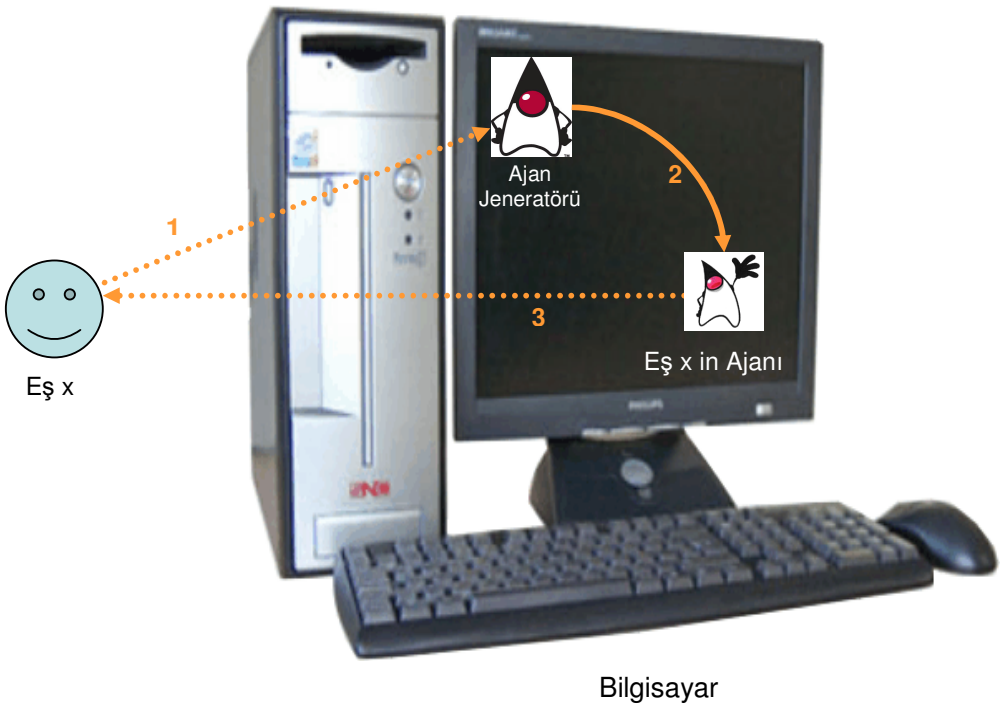
Beşinci işlem eşdüzey *sistemden bilgi indirme* işlemidir. Bu işlem sayesinde peerın almak istediği bilgi eşdüzey sisteminde bulmakta ise peera bulunan bu bilginin aktarılma işlemi yapılır.

### 5.3.1 Sisteme Dahil Olma İşleminin Anlambilimsel Yapısı

Eşdüzey sisteminden hizmet almak isteyen peerlar, sisteme dahil olma işlemi yapmaktadır. Bu işlem mobil veya kablosuz ortamda çalışan bir eş tarafından çalıştırıldığında, sistem otomatik olarak o eş ile birlikte çalışacak bir ajan yaratarak, yaratılan yeni ajanı o peera atar. Bu işlemden sonra eş sistemden ayrılma isteğinde bulunana kadar sürekli olarak yaratılan bu ajan o eş için çalışmaya devam eder.

Şekil 5.2 sisteme dahil olma işlemi çalışma prensibini göstermektedir. Şekil 5.2'nin solunda yer alan eş (eş x) sisteme dahil olma işlemi çalıştırdığında merkezi bilgisayarda bulunan ajan jeneratörü işlemine bu isteğini iletir (Şekil 5.2'deki 1 numaralı adım). Bu işlem sonucunda o eş için yeni bir ajan yaratılması gerekmektedir. Bu isteği alan ajan jeneratörü istekte bulunan eşin istekleri ile ilgilenecek ajan işlemini

yaratır (Şekil 5.2'deki 2 numaralı adım). Yeni yaratılan ajan hizmet edeceği eşe kendisi ile ilgili bilgileri iletir (Şekil 5.2'deki 3 numaralı adım).




**Şekil 5.2** Sisteme dahil olma işleminin çalışma prensibi

```
...
...
DatagramPacket paket = new DatagramPacket(tampon,
tampon.length);
datagram_Socket.receive(paket);
islem = new String(paket.getData(), 0, paket.getLength());
```

```
String peer_id = paket.getAddress() + "-" +
paket.getPort();
new YeniKanal(peer_id);
...
...

...
...
DatagramPacket paket = new DatagramPacket(tampon,
tampon.length, address, peer_port);
datagram_Socket = new DatagramSocket();
datagram_Socket.send(paket);
...
...
```

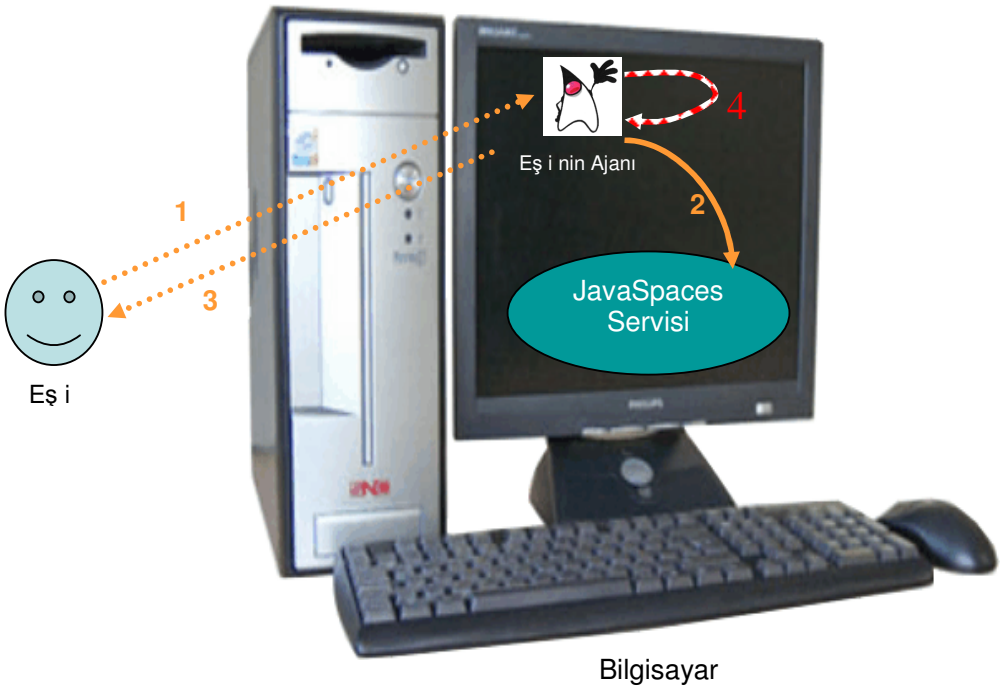
**Şekil 5.3** Sisteme dahil olma işlemi programının ana hatları

### 5.3.2 Sistemden Ayrılma İşleminin Anlambilimsel Yapısı

Eşdüzey sisteminden hizmet alma işlemine son vermek isteyen peerlar, sistemden ayrılma işlemini çalıştırmaktadır. Bu işlem mobil veya kablosuz ortamda çalışan bir eş tarafından çalıştırıldığında o eş ile birlikte çalışmakta olan ajan o peerın daha önce paylaşmak isteyip JavaSpaces servislerine konulmuş olan bilgiler var ise onları JavaSpaces servislerinden siler. Daha sonra ajan peera sistemden kaldırıldığına dair bir mesaj yollar. Son olarak ajan processi kendisini sonlandırır.

Şekil 5.4 sistemden ayrılma işleminin çalışma prensibini göstermektedir. Şekil 5.4'ün sol tarafında yer alan eş, sistemden ayrılmaya karar verdiği zaman kendine tahsis edilen ajana sistemden ayrılma mesajı gönderir (Şekil 5.4'teki 1 numaralı adım). Sistemden ayrılma mesajını alan ajan iletişim halinde olduğu peerın o ana kadar sistemdeki diğer peerların kullanımına sunduğu bilgiler var ise bu bilgileri JavaSpaces servisinden kaldırır (Şekil 5.4'teki ikinci adım). Daha sonra ajan peera sistemden başarı ile ayrıldığına dair onay mesajı gönderir ve bu sayede eş sistemden ayrılmış olur (Şekil

5.4'teki üçüncü adım). Sistemden ayrılma işleminin son adımı olarak sistemden ayrılan peera hizmet eden ajan processi kendisini sonlandırarak sistemden kendi kendisini kaldırır (Şekil 5.4'teki dördüncü adım).




**Şekil 5.4** Sistemden ayrılma işleminin çalışma prensibi

```
...
space = (IJSpace) SpaceFinder.find(args[0]);
if (space == null)
{
    System.exit(0);
}

space.setFifo(true);
```

```
Entry gigaSpacesteki_bilgiler[] = null;
Giga_Entry template = new Giga_Entry();
template.peerID = new Integer(peer_kimligi);
...


...
gigaSpacesteki_bilgiler   =   space.takeMultiple(template,
null, 100);
tampon = "-1".getBytes();
paket = new DatagramPacket(tampon, tampon.length, address,
peer_port);
datagram_Socket.send(paket);
...
System.exit(1);
```

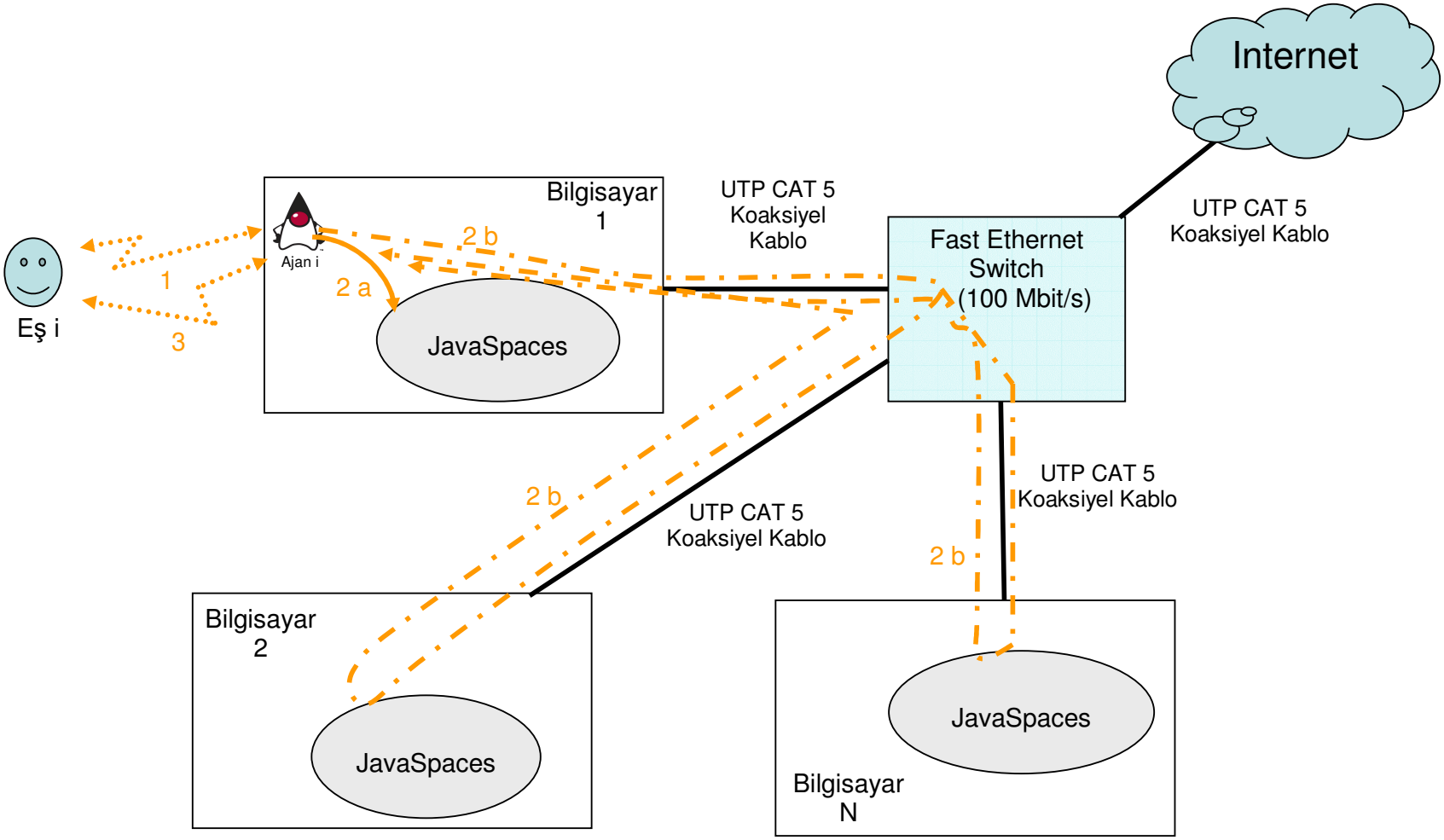
**Şekil 5.5** Sistemden ayrılma işlemi programının ana hatları

### 5.3.3 Bilgi Arama İsteği İşleminin Anlambilimsel Yapısı

Eşdüzey sisteminde bulunan bir eş bilgi aramak istediğinde, bilgi arama isteği işlemini çalıştırır. Bu işlem çalıştırıldığı zaman eş kendisine bağlı ajana bu talebi iletir. Ajan bu işlemi yerine getirebilme amacı ile bir arama işlemi gerçekleştirir. Bu arama işlemi sonucunda olumlu ise istenen bilgi peera iletilir. Eğer aranılan şey bulunamaz ise aranılan bilginin bulunamadığı peera bildirilir.

Şekil 5.6 bilgi arama isteği işleminin çalışma yapısını göstermektedir. Şekil 5.6'da bilgi arama isteğinde bulunan eş, bu talebini onun adına çalışan ajana iletir (Şekil 5.6'daki 1 nolu adım). Bilgi arama talebini alan ajan öncelikle kendi bulunduğu bilgisayar üzerinde bulunan JavaSpaces servisi üzerinde gerekli aramayı yapar (Şekil 5.6'daki 2a numaralı adım). Eğer istenilen bilgi yerel bilgisayarda çalışmakta olan JavaSpaces servisinde bulunur ise hemen peera bu bilgiyi aktarır (Şekil 5.6'daki 3 numaralı adım). Ajan eğer aynı bilgisayardaki JavaSpaces servisinde bu bilgiyi bulamaz

ise diğer bilgisayarlardaki JavaSpaces servislerinde bu bilgiyi sırası ile arar (Şekil 5.6'daki 2b numaralı adım). Eğer aranılan bilgi diğer bilgisayarlardaki JavaSpaces servislerinde bulunur ise peera aktarılır (Şekil 5.6'daki 3 numaralı adım). Eğer aranılan bilgi bulunamaz ise aranılan bilginin bulunamadığı peera iletilir (Şekil 5.6'daki 3 numaralı adım).




**Şekil 5.6** Bilgi arama isteğinin çalışma şekli

```
...
space = (IJSpace) SpaceFinder.find(args[0]);
if (space == null)
{
   System.exit(0);
}

Giga_Entry aranan_bilgi;
space.setFifo(true);
Giga_Entry template = new Giga_Entry();
template.paylasilan_bilgi = bilgi.toString();
...

...
aranan_bilgi  =  (Giga_Entry)  space.readIfExists(template,
null, Integer.MAX_VALUE);
if (aranan_bilgi != null)
{
   g_servis_numarasi = 1;
   aranan_bilgi_bulundu = "evet";
}

else if (aranan_bilgi == null)
{
   aranan_bilgi_bulundu = "hayir";
   space2 = (IJSpace) SpaceFinder.find(args[1]);
   if (space2 == null)
  {
...

...
```

```
if (aranan_bilgi_bulundu == "evet")
{
peer_a1 = aranan_bilgi_bulundu;
tampon = peer_a1.getBytes();
}
else if (aranan_bilgi_bulundu == "hayir")
{
peer_a1 = aranan_bilgi_bulundu;
tampon = peer_a1.getBytes();
}
...

DatagramPacket paket4 = new DatagramPacket(tampon,
tampon.length, address, peer_port);
datagram_Socket.send(paket4);
...
```

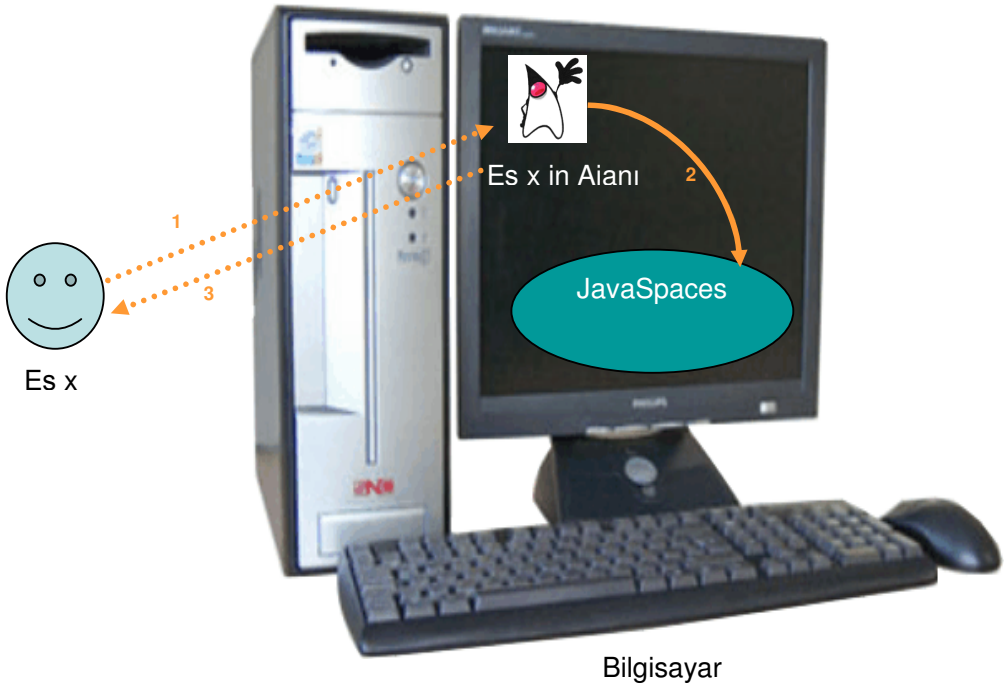
**Şekil 5.7** Bilgi arama isteği programının ana hatları

### 5.3.4 Sisteme Bilgi Ekleme İşleminin Anlambilimsel Yapısı

Eşdüzey sistemi üzerinden bilgi paylaşmaya karar veren peerlar sisteme bilgi ekleme işlemini gerçekleştirirler. Sisteme bilgi ekleme işlemi ile peerın elinde bulunan bilgilerin bir kısmı ya da bütün bilgileri eşdüzey sisteminden faydalanmakta olan peerların erişimine açılmış olmaktadır. Bu işlem çalıştırıldığında eş paylaşmak istediği bilgileri kendine bağlı ajana gönderir. Bu eş için çalışan ajan oradaki bilgilerin tamamını diğer ajanlar aracılığı ile eşdüzey sistemi kullanan diğer peerlar tarafından erişimine açılmasını sağlamış olmaktadır.

Şekil 5.8 sisteme bilgi ekleme işleminin çalışma prensibini göstermektedir. Eş bir ya da daha fazla miktar bilgiyi eşdüzey sistem aracılığı ile diğer peerlar ile paylaşmaya karar verdiğinde, sisteme bilgi ekleme işlemini çalıştırmaktadır. Bu işlemin ilk adımı olarak eş paylaşmak istediği bilgileri kendisi için çalışmakta olan ajanına

bildirir (Şekil 5.8'deki 1 numaralı adım). Paylaşılacak bilgileri alan ajan, bu bilgileri kendisi ile aynı bilgisayarda çalışmakta olan JavaSpaces servisine yükler (Şekil 5.8'deki 2 numaralı adım). Son aşama olarak ajan, yaptığı bilgileri yükleme işleminin başarılı ya da başarısız olduğunu kendi sorumlu olduğu peera bildirir (Şekil 5.8'deki 3 numaralı adım).




**Şekil 5.8** Sisteme bilgi ekleme işleminin çalışma şekli

```
...
address = paket.getAddress();
int port = paket.getPort();
```

```
space = (IJSpace) SpaceFinder.find(args[0]);
if (space == null)
{
System.exit(0);
}
...

space.setFifo(true);
Giga_Entry eklenecek_bilgi = new Giga_Entry();
eklenecek_bilgi.paylasilan_bilgi = bilgi.toString();
eklenecek_bilgi.peerID = new Integer(peer_kimligi);
leasing      =      space.write(eklenecek_bilgi,      null,
Integer.MAX_VALUE);

...

if (leasing != null)
{
tampon = "0".getBytes();
}
else if (leasing == null)
{
tampon = "-3".getBytes();
}

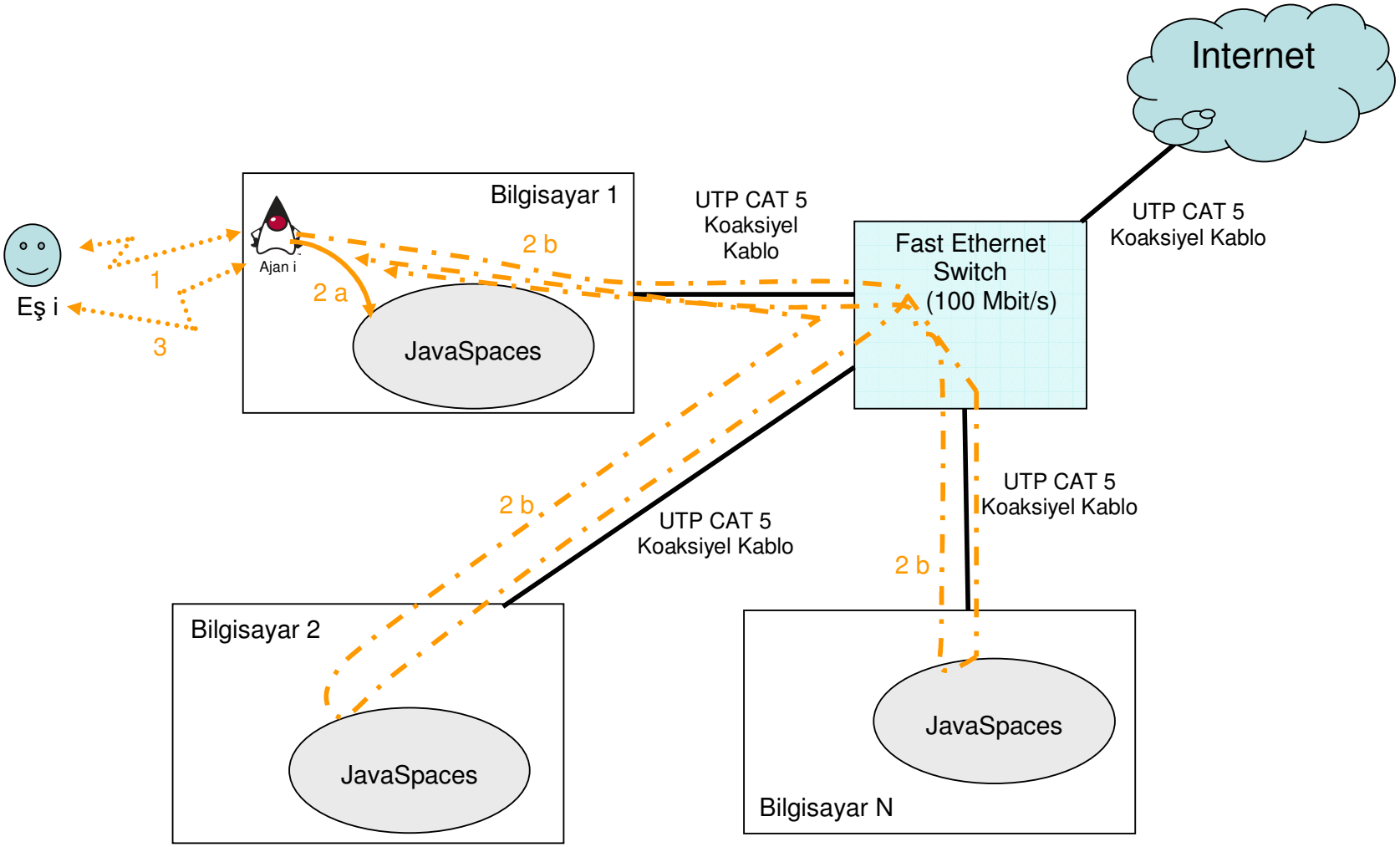
...
paket = new DatagramPacket(tampon, tampon.length, address,
port);
datagram_Socket.send(paket);
```

**Şekil 5.9** Sisteme bilgi ekleme işlemi programının ana hatları

### 5.3.5 Sistemden Bilgi İndirme İşleminin Anlambilimsel Yapısı

Eşdüzey sistemden bilgi indirmek isteyen peerlar, sistemden bilgi indirme işlemi yapmaktadır. Bu işlemin çalışması, önceki diğer dört işlemden farklı yapıya sahiptir, öyleki önceki dört işlemin çalışması diğer işlemler ile ilgili değil iken sistemden bilgi indirme işlemi için durum böyle değildir. Bu işlemin çalışmasından önce bilgi arama işleminin çalışması gerekir. Bilgi arama işleminin sonucuna göre eş bilgi indirme işlemini çalıştırıp çalıştırmamaya karar verebilir.

Şekil 5.6 bilgi indirme işleminin çalışma yapısını göstermektedir. Eş bilgi arama işlemini gerçekleştirdikten sonra ona dönen sonuçlar sonucunda eşdüzey sistemden bilgi indirme işlemini çalıştırmaya karar verebilir. Sistemden bilgi indirme işlemi eş tarafından çalıştırıldığı zaman o peerdan sorumlu olan ajana bu istek iletilir (Şekil 5.6'daki 1 numaralı adım). Ajan daha önce çalıştırmış olduğu bilgi arama sonucu yerini tespit ettiği bilgiyi ilgili JavaSpaces servisinden okur. Okunan bu bilgi ajan ile aynı bilgisayardaki JavaSpaces servisinde ise genellikle o tercih edilir (Şekil 5.6'daki 2a numaralı adım). Eğer bilgi diğer bilgisayarlardaki JavaSpaces servislerinde ise onlardan alınır (Şekil 5.6'daki 2b numaralı adım). Ajanın aldığı bu bilgi daha sonra bağlı olduğu peera iletilir (Şekil 5.6'daki 3 numaralı adım).




**Şekil 5.10** Sistemden bilgi indirme işleminin çalışma şekli

```
...
paket = new DatagramPacket(tampon, tampon.length);
datagram_Socket.receive(paket);
bilgi = new String(paket.getData(), 0, paket.getLength());

address = paket.getAddress();
int port = paket.getPort();

if (bilgi == "0")
{
space = (IJSpace) SpaceFinder.find(args[0]);
if (space == null)
{
System.exit(0);
}

Giga_Entry aranan_bilgi;
space.setFifo(true);
Giga_Entry template = new Giga_Entry();

template.paylasilan_bilgi = bilgi.toString();

aranan_bilgi = (Giga_Entry) space.readIfExists(template,
null, Integer.MIN_VALUE);

if (aranan_bilgi != null)
{
tampon = aranan_bilgi.paylasilan_bilgi.getBytes();
paket = new DatagramPacket(tampon, tampon.length, address,
port);
datagram_Socket.send(paket);
}
```

```
}
else if (bilgi == "1")
{
space2 = (IJSpace) SpaceFinder.find(args[1]);
if (space2 == null)
{
System.exit(0);
}

Giga_Entry aranan_bilgi;
space2.setFifo(true);
Giga_Entry template = new Giga_Entry();

template.paylasilan_bilgi = bilgi.toString();
aranan_bilgi = (Giga_Entry) space2.readIfExists(template,
null, Integer.MIN_VALUE);
if (aranan_bilgi != null)
{
tampon = aranan_bilgi.paylasilan_bilgi.getBytes();
paket = new DatagramPacket(tampon, tampon.length, address,
port);
datagram_Socket.send(paket);
}
}
else if (bilgi == "2")
{
...
```

**Şekil 5.11** Sistemden bilgi indirme işlemi programının ana hatları

# 6. DENEYSEL ÇALIŞMALAR

## 6.1 Deneylerin Organizasyonu

Bu bölümde deneylerde kullanılan sistemlerin organizasyonu anlatılmaktadır. Deneylerimiz iki tür kablosuz ağ kullanılarak gerçekleştirilmiştir. Bunlardan birinci sistemimiz kablosuz yerel alan ağında (WLAN) birbirine bağlı bilgisayar sisteminde gerçekleştirilmiştir. İkinci sistem ise daha uzak mesafeli ve binaların çatısına konumlandırılan antenler üzerinden bilgisayarların birbirleri ile haberleşmesi şeklinde gerçekleştirilmiştir.

Deney sonuçlarının daha gerçekçi olması için aynı testler binlerce kez tekrar edilmiştir. Deneylerde kullanılan bilgisayar sistemleri aynı özellikte olup (homogenous), bu bilgisayarlar aşağıdaki Tablo 6.1'de gösterilen özelliklere sahiptir.

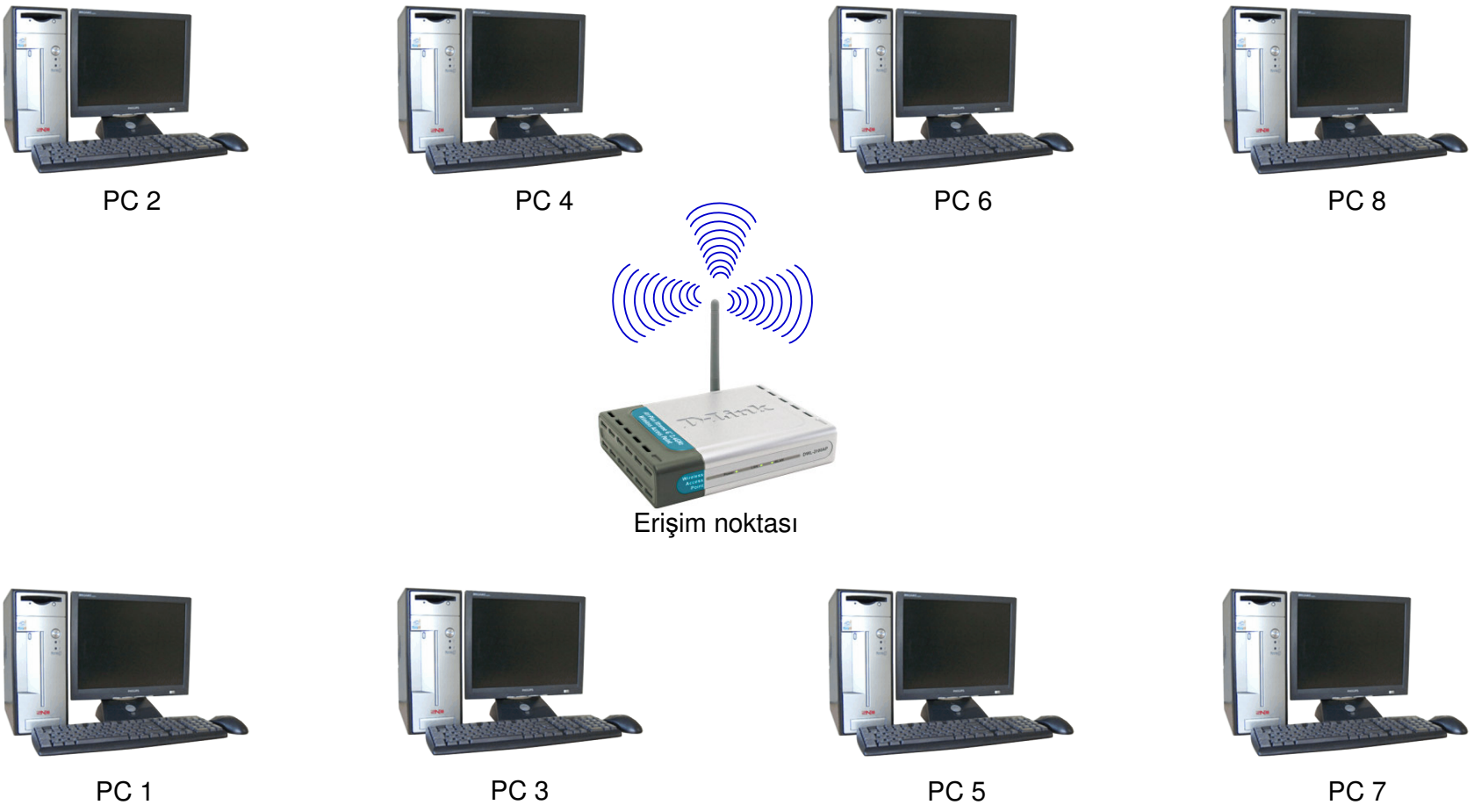
**Tablo 6.1** Deneylerde kullanılan bilgisayar sistemlerinin özellikleri

| **Parça** | **Türü** |
|---|---|
| İşlemci | Intel Pentium 4, 3.00 GHz |
| Cache | 2 MB |
| RAM | 512 MB |
| Hard disk | 40 GB, 7200 RPM |
| Kablosuz ağ kartı | D-Link DWL-G520, G 108 Mbit/s (Turbo mode), 2.4 GHz Wireless PCI Card |
| Kablolu ağ kartı | Broadcom NetXtreme Gigabit Ethernet |
| İşletim sistemi | Windows XP Professional Service Pack 2 |
| Java sanal makinesi (JVM) | Java version "1.6.0_01-b06" |
| Jini teknolojisi | Jini Technology Starter Kit v2.1 |
| JavaSpaces | Outrigger v2.2 |
| GigaSpaces | 5.2.2 |
| Kablosuz erişim noktası | D-Link DWL-2100AP, G 108 Mbit/s (Turbo mode) |

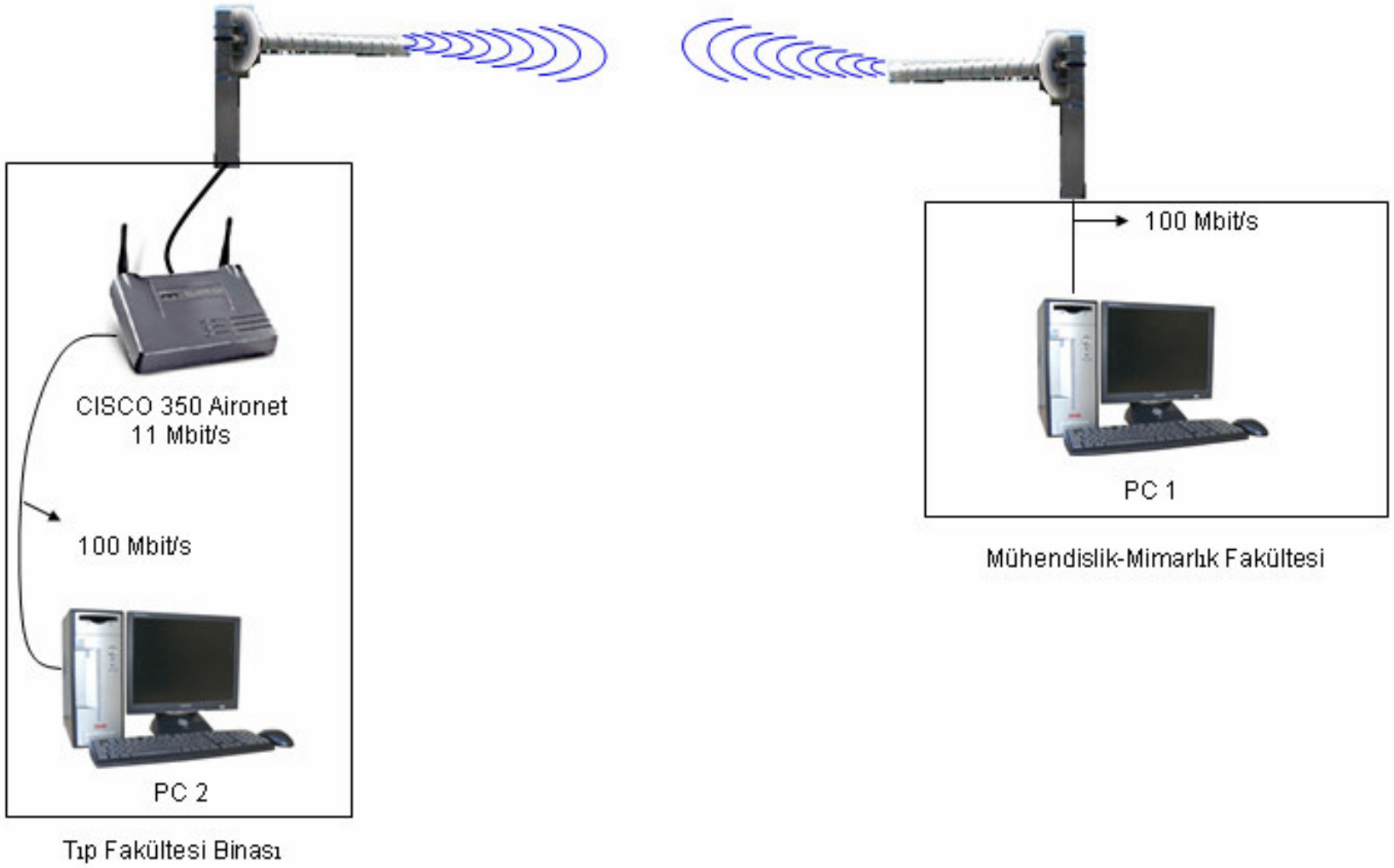
Birinci deneyler için kurulan bilgisayar sisteminde toplam sekiz adet Intel Pentium 4, 3.00 GHz, 512 MB RAM özellikli, IEEE 802.11g türünde çalışan bilgisayarlar kullanılmıştır. Bu sekiz bilgisayar aynı erişim noktası (AP) üzerinden birbirine bağlanmışlardır. Kablosuz iletişim sırasında verim düşmesini engellemek için bilgisayarların hepsi erişim noktasına yakın olarak Şekil 6.1'de gösterilen şekilde konumlandırılmışlardır.

İkinci deneyler için kurulan bilgisayar sisteminde ise toplam iki adet yine Intel Pentium 4, 3.00 GHz, 512 MB RAM özellikli bilgisayar kullanılmıştır. Bu iki bilgisayar öncelikle 100 Mbit/s hızında çalışan Ethernet hattı ile CISCO 350 Aironet cihazına bağlantı kurmakta, oradan da diğer bilgisayarın bulunduğu bina ile iletişim için binanın çatısına yerleştirilmiş olan anten aracılığı diğer bilgisayar ile kablosuz iletişim sağlamaktadır. İkinci tür deneyler sırasında kullanılan CISCO 350 Aironet kablosuz iletişim hızı 11 Mbit/s olarak kullanılmıştır. İkinci tip deneylerde kullanılan bilgisayar organizasyonu Şekil 6.2'de gösterilmiştir.

Jini teknolojisinin 2.1 sürümü iyi şekilde dökümantasyona sahip değildir. Tezin hazırlandığı sıradaki en güncel Jini sürümü olan bu paketin içinde yer alan JavaSpaces servisinin 2.2 versiyonu da benzer bir şekilde gerekli dökümantasyonu içermemektedir. Bu bağlamda birçok JavaSpaces uygulaması kullanıcısının yaptığı şekilde, başka bir JavaSpaces servisi uygulaması kullandık. Bu tezdeki çalışmalarda kullanmak üzere seçtiğimiz JavaSpaces servisi uygulaması GigaSpaces şirketinin ürünü olan GigaSpaces programının 5.2.2 sayılı sürümüdür. Diğer bölümlerde değinildiği şekilde GigaSpaces servisinin JavaSpaces servisinden birçok iyi özelliği bulunmaktadır. Bu bağlamda sistemde kullanılan bu servisin yararları görülmüştür.




**Şekil 6.1** Birinci tip deneylerde kullanılan bilgisayarların organizasyonu




**Şekil 6.2** İkinci tip deneylerde kullanılan bilgisayarların organizasyonu

## 6.2 Deneysel Sistemin Özellikleri

Bu bölümde sistemin deneysel çalışmalarında JavaSpaces servisinin yerine kullanılan GigaSpaces servisinin sistem davranışlarının anlaşılması amacı ile servisin performans özellikleri gösterilmektedir. Yapılan bu çalışmalar sırasında bilgisayarlarda yalnızca bu testler çalıştırılmıştır.

### 6.2.1 GigaSpaces Servisinde Read İşlemi Yapılması Durumundaki Sistem Davranışı

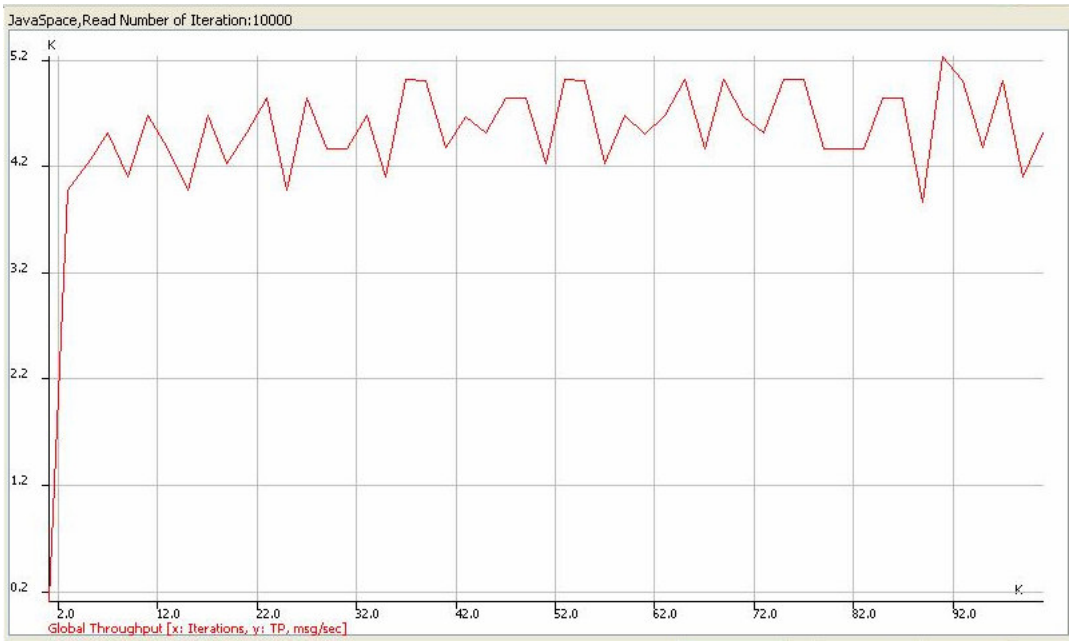
JavaSpaces servisinin temel işlemlerinden olan "*read*" işlemi GigaSpaces servisinde de bulunmaktadır. Deneylerde kullanılan bilgisayarlar üzerinde yapılan testlerden ilkinde sistemde tek thread kullanarak GigaSpaces servisinden okuma işlemi gerçekleştirilmiştir. Bu deney sonucunda elde edilen sonuç Şekil 6.3'te gösterildiği şekilde ortaya çıkmıştır.




**Şekil 6.3** GigaSpaces servisinin üzerinde kurulu JavaSpaces servisinin deneylerde kullandığımız bilgisayar sistemlerinde tek thread kullanılarak yapılan *READ* işlemi için gösterdiği performans yapısı

Şekil 6.3'te görülen grafikte bir saniyelik sürede GigaSpaces servisi üzerinde gerçekleştirilebilen read işlemi gösterilmektedir. 10 bin iterasyon gerçekleştirilerek yapılan testlerde saniyede 2.8 bin mesajın GigaSpaces servisinden okunabildiği görülebilmektedir.

GigaSpaces servisindeki read işlemi ile ilgili ikinci test olarak sistemde aynı anda 10 threadin çalışması ile yapılan test bulunmaktadır. Bu testde de yine 10 bin iterasyon yapılmıştır. Şekil 6.4'te bu test sonucunda ortaya çıkan grafik görülmektedir.




**Şekil 6.4** GigaSpaces servisinin üzerinde kurulu JavaSpaces servisinin deneylerde kullandığımız bilgisayar sistemlerinde 10 thread kullanılarak yapılan *READ* işlemi için gösterdiği performans özellikleri

Şekil 6.4'te gösterilen grafikte sistemin GigaSpaces servisinden bir saniyede toplam olarak 4 ila 5.2 milyon arasında okuma işlemi gerçekleştirebildiği görülmektedir. Şekil 6.3 ve 6.4 beraberce incelendiğinde ise GigaSpaces servisinden bir saniyede 5 milyon dolayında okuma işlemi gerçekleştirilebileceği sonucuna varılmaktadır.

### 6.2.2 GigaSpaces Servisinde Write İşlemi Yapılması Durumundaki Sistem Davranışı

JavaSpaces servisinin diğer bir temel işlemi olan "*write*" işlemi yine GigaSpaces servisinde bulunmaktadır. Deneylerde kullanılan bilgisayarlar üzerinde yapılan testlerden ikinci olarak sistemde tek thread kullanarak GigaSpaces servisine yazma işlemi gerçekleştirilmiştir. Bu deney sonucunda elde edilen sonuç Şekil 6.5'te gösterildiği şekilde ortaya çıkmıştır.

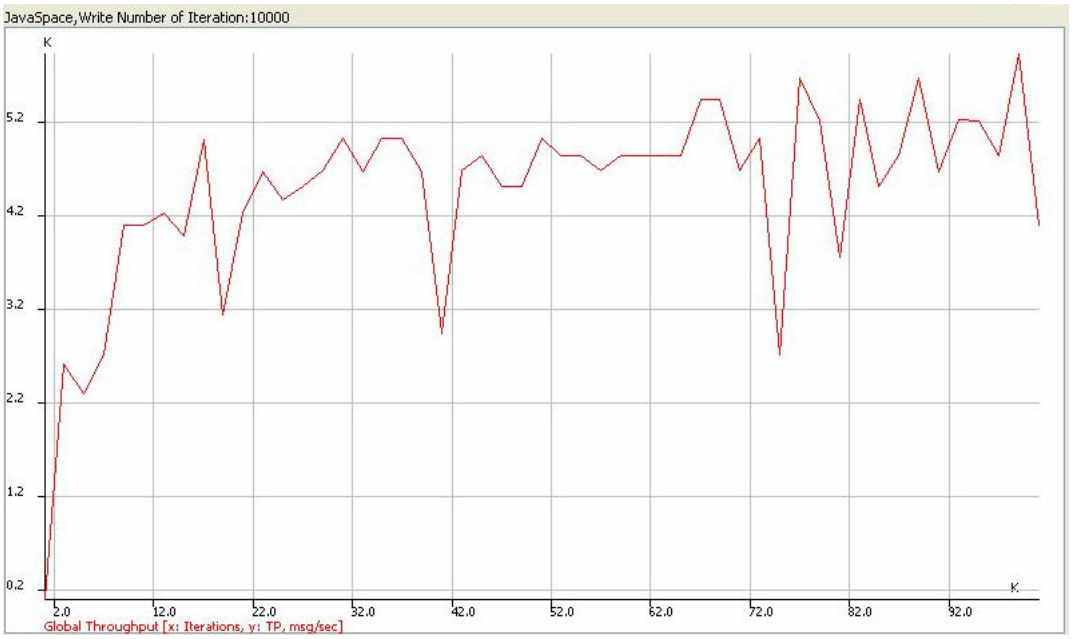



**Şekil 6.5** GigaSpaces servisinin üzerinde kurulu JavaSpaces servisinin deneylerde kullandığımız bilgisayar sistemlerinde tek thread kullanılarak yapılan *WRITE* işlemi için gösterdiği performans özellikleri

Şekil 6.5'te görülen grafikte bir saniyelik sürede GigaSpaces servisi üzerinde gerçekleştirilebilen write işlemi görülmektedir. read işlemi için gerçekleştirilen testte olduğu gibi yine on bin iterasyonluk yapılan testlerde saniyede 2 bin mesajın GigaSpaces servisine yazılabildiği görülebilmektedir.

GigaSpaces servisindeki Write işlemi ile ilgili ikinci test olarak sistemde aynı anda 10 threadin çalışması ile yapılan deney bulunmaktadır. Bu deneyde de yine on bin

iterasyon gerçekleştirilmiştir. Şekil 6.6'da bu test sonucunda ortaya çıkan grafik görülmektedir.

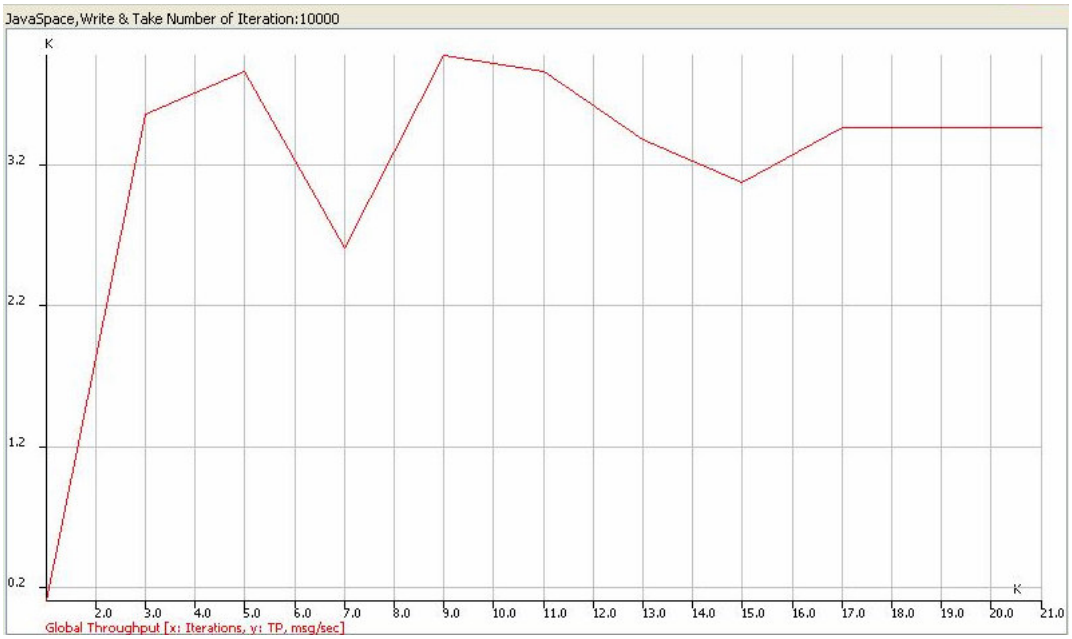



**Şekil 6.6** GigaSpaces servisinin üzerinde kurulu JavaSpaces servisinin deneylerde kullandığımız bilgisayar sistemlerinde 10 thread kullanılarak yapılan *WRITE* işlemi için gösterdiği performans özellikleri

Şekil 6.6'da gösterilen grafikte sistemin GigaSpaces servisinden bir saniyede toplam olarak 3 ila 5.5 milyon arasında yazma işlemi gerçekleştirebildiği görülmektedir. write işlemi ile read işlemi kıyaslandığı zaman, write işleminin çalışma performansı daha fazla değişiklik göstermektedir. Bu bağlamda sisteme bazen 3 milyon dolayında mesaj yazılırken, bu sayı aniden 5.5 milyon dolayına çıkabilmektedir.

### 6.2.3 GigaSpaces Servisinde Önce Write Arkasından Read İşlemlerinin Yapılması Durumundaki Sistem Davranışı

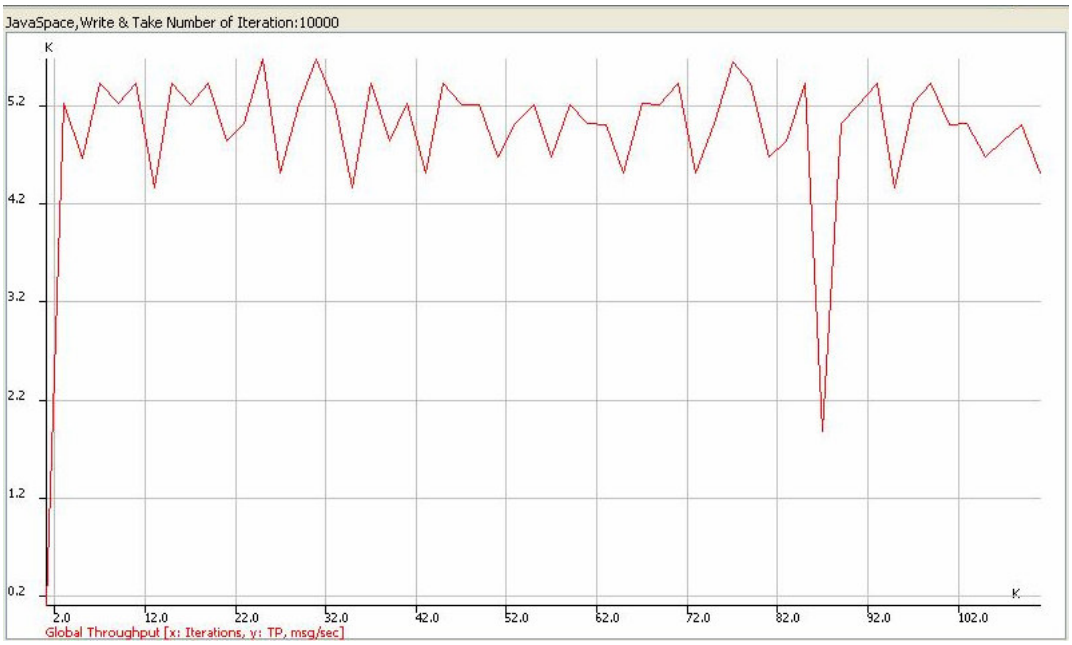
Deneylerde kullanılan bilgisayarlar üzerinde yapılan diğer bir test ise GigaSpaces servisine öncelikle Write işlemi ile entry yazdırmak daha sonra ise bu entry'yi okuma sonucunda oluşacak sistemin performansının gözlenmesidir. Bu amaçla yapılan ilk testte yine on bin iterasyonluk bir deney çalıştırılmış ve Şekil 6.7'deki şekilde sonuç elde edilmiştir.




**Şekil 6.7** GigaSpaces servisinin üzerinde kurulu JavaSpaces servisinin deneylerde kullandığımız bilgisayar sistemlerinde tek thread kullanılarak yapılan önce *WRITE* ve sonrasında *READ* işlemi için gösterdiği performans özellikleri

Şekil 6.7'te görülen grafikte bir saniyelik sürede GigaSpaces servisi üzerinde gerçekleştirilebilen önce write işlemi daha sonra ise yazılan entry'nin read işlemi üzerine oluşan sistem performansı görülmektedir. Diğer önceki testlerde olduğu gibi yine on bin iterasyonluk yapılan testlerde saniyede 2.6 bin dolayında mesajın GigaSpaces servisine yazma okuma yapılabildiği görülebilmektedir.

GigaSpaces servisindeki önce Write işlemi sonra Read işlemi ile ilgili ikinci test olarak sistemde aynı anda 10 threadin çalışması ile yapılan deney de gerçekleştirilmiştir. Bu deneyde de yine on bin iterasyon gerçekleştirilmiştir. Şekil 6.8'de bu test sonucunda ortaya çıkan grafik görülmektedir.




**Şekil 6.8** GigaSpaces servisinin üzerinde kurulu JavaSpaces servisinin deneylerde kullandığımız bilgisayar sistemlerinde 10 thread kullanılarak yapılan önce *WRITE* ve sonrasında *READ* işlemi için gösterdiği performans özellikleri

Şekil 6.8'de gösterilen on threadli sistemde saniyede 4.3 ila 5.6 bin arasında mesaj GigaSpaces servisine yazılıp okunabilmektedir. Bu bağlamda, Şekil 6.8'de gösterilen on threadli sistem ile Şekil 6.7'de gösterilen tek threadli sistem arasında önemli bir fark ortaya çıkmakta olup thread sayısının sistemin performansını olumlu etkilediği gözlemlenmektedir.

## 6.3 Deneysel Çalışmalarda Yapılan İncelemeler

Bu kısımda beşinci bölümde dizaynı ve uygulamasının gösterildiği ajan tabanlı eşdüzey sistemin göstermiş olduğu performans yapısı gösterilmektedir. Bu bağlamda GigaSpaces servisinin sağlamış olduğu iki farklı tür servis olan "*embeded*" ve "*gsServer*" servisi kullanılarak deneyler gerçekleştirilmiştir. Burada bahsedilen embeded servisinin anlamı GigaSpaces servisinin çalışması sırasında bilgisayarın yanlızca geçici hafızasında bulunması dolayısı ile çok daha hızlı çalışabilmesi ancak bilgisayar kapandığında elindeki bilgileri kaybetmesi şeklindedir. Diğer taraftan gsServer servisi çalıştığında sisteme birçok ek serviste yüklenir (örneğin lookup server, transaction manager, v.b.) ayrıca GigaSpaces servisinin içindeki bilgiler bilgisayar kapansada durmaya devam eder.

Bu kısımda yapılan deneylerin bir diğer özelliğide sistemde 1 peer, 5 peer ve 10 peer için farklı deneyler ile sistemin performansının gözlemlenmesidir. Bu bağlamda, hem embededli sistem için hemde gsServer kullanan sistemlerde 1, 5, 10 peer bulunan testler ayrı ayrı yapılmıştır.

Deneysel çalışmalardaki bir diğer özellik ise geliştirilen sistemin iki farklı mimari sistemde denenmesidir. Testlerde kullanılan sistemler sırası ile Şekil 6.1 ve Şekil 6.2'de gösterilmiştir.

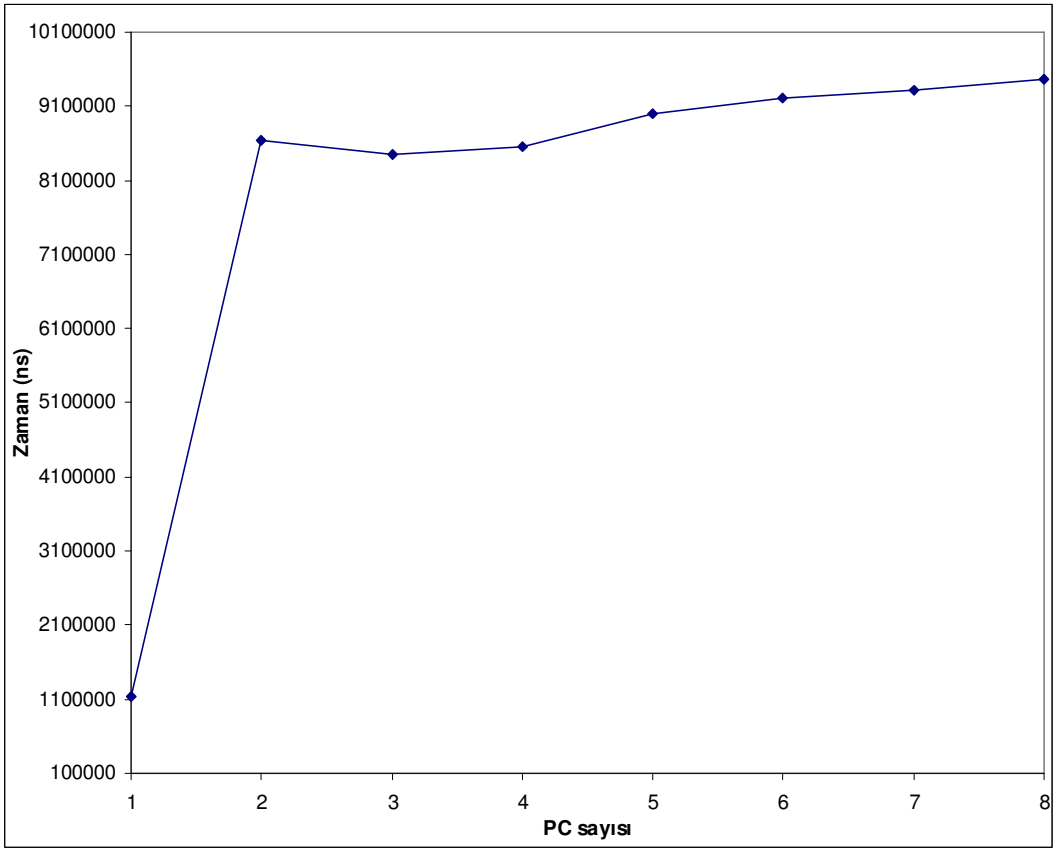
### 6.3.1 Aynı Salondaki Bilgisayar Sistemleri için Sistem Performansı

Bu kısımda gerçekleştirilen ilk deneyler aynı salondaki bilgisayar sistemleri için sistem performansının araştırılması şekli ile gerçekleştirilmiştir. Bu deneylerde kullanılan bilgisayarların organizasyonu Şekil 6.1'de gösterildiği şekildedir.

#### 6.3.1.1 Aynı Salondaki Bilgisayar Sistemlerinde Embeded Türü GigaSpaces Servisi Kullanan Sistem

Embeded türü GigaSpaces servisi kullanan sistem çalışması için önceden bir servis çalıştırılmasına gerek yoktur. GigaSpaces servisini kullanacak program çalıştığı zaman otomatik olarak Embeded space yaratılmaktadır.

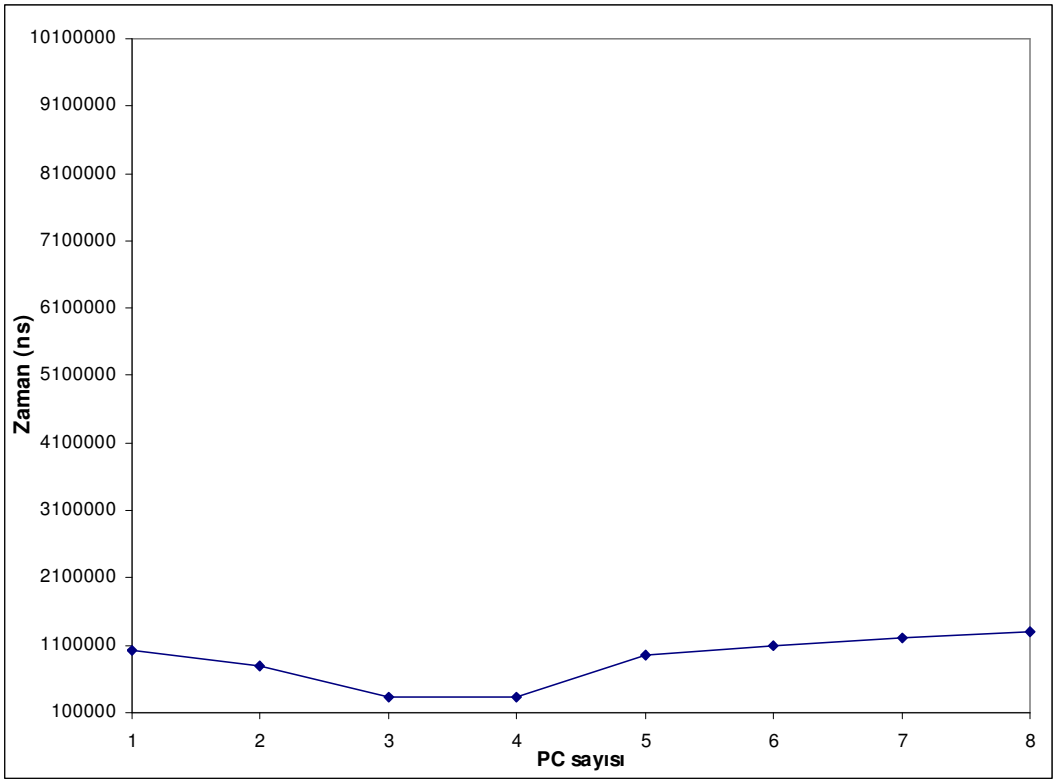
Yapılan deneylerde iki performans ölçümü yapılmaktadır. Bunlardan ilki eşdüzey sistemden alınmak istenen bilgi için geçen zamandır. Bu ilk performans kriteri; sistemden indirilmek istenen bilgi sırasında geçen zamanı göstermektedir. Şekil 6.9'da sistemde 1 peer olması durumunda embeded türü GigaSpaces servisi kullanılarak eşdüzey sistemden alınmak istenen bilgi için geçen zaman gösterilmektedir.




**Şekil 6.9** 1 eşli sistemde, "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman

Şekil 6.9'daki sistemin yapısı incelendiğinde yalnızca bir bilgisayar olduğunda beklendiği şekilde en kısa zamanda gerçekleşmiştir. Gerekli zaman iki bilgisayar ile birlikte bir anda yaklaşık sekiz kat artmış daha sonraki bilgisayar sayısının artışlarında küçük dalgalanmalar ile kısmen artış göstermektedir.

İkinci tür performans testi ise eşdüzey sistem üzerinde paylaşılmasına karar verilen bilgi için geçen zamandır. Şekil 6.10'da bu testin 1 peer kullanılarak yapılan versiyonu yer almaktadır.

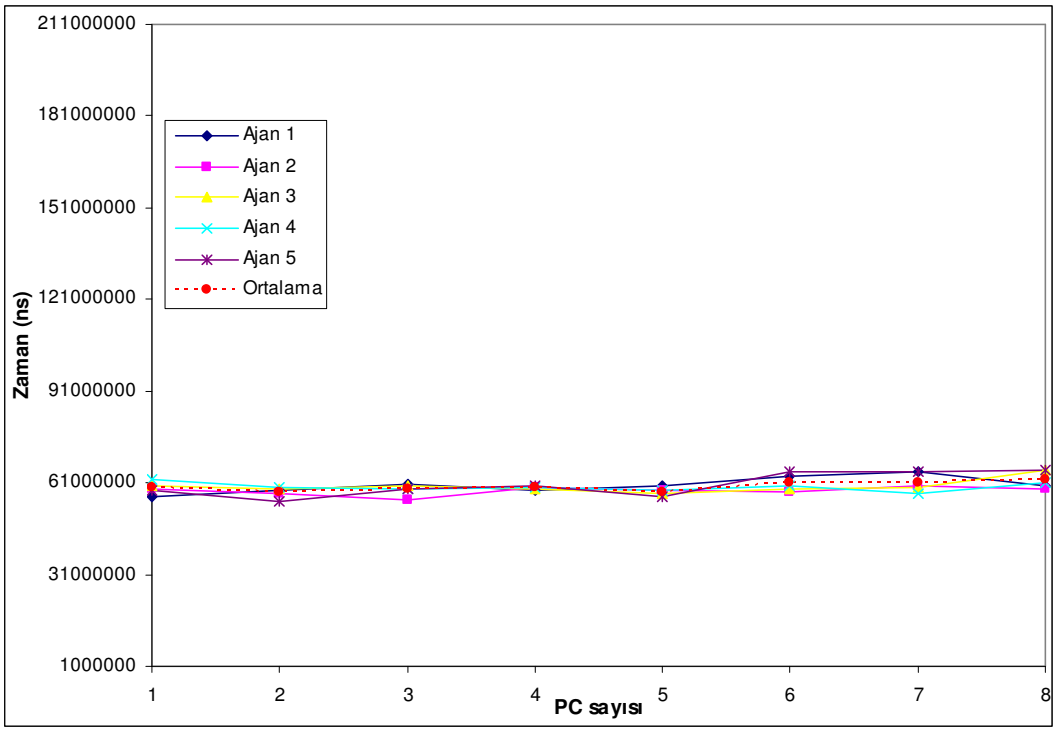



**Şekil 6.10** 1 eşli sistemde, "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman

Şekil 6.10'daki test sonucunda 1 ile 8 PClik sistemler arasındaki performans yapısının 2 ila 12 milyon arası gerçekleştiği gözlemlenmiştir. Bu testte sistemin yapısı gereği peerın sisteme birşey yüklemesi GigaSpaces servisinin yükü ile ilgili değildir. Bu bağlamda bu testteki sistemin embeded türü GigaSpaces servisi kullanmasına rağmen 1 PC'den 2 PC'ye geçtiğinde paylaşım için gerekli zamanın düştüğü, aynı şekilde 2'den 3'e geçildiğinde de bu düşüşün sürdüğü gözlemlenmiştir. Daha sonra 3 ve 4 PC için bu sürelerin birbirine yakın şekilde çıktığı görülmüş, mütakiben 4 PC ve sonrasında ise geçen zamanda bir artış gözlenmiştir.

Sistemdeki PC sayısı arttıkça gerekli zamanın önce düşüp sonra artmasının sebebi bu performans kriteri için embeded türü GigaSpaces olmasının bir önemi olmamasına rağmen PC sayısının artması peer ile beraber çalışacak ajanların sayısının artmasıdır. Dolayısı ile öncelikle peerlara hizmet kalitesi artarak paylaşılmasına karar verilen bilgi için geçen zaman düşmekte, ancak 4 PC'den sonra sistemin genel yükü artacağından bu performans kriteri içinde bir artış durumu ortaya çıkmaktadır.

5 peer için eşdüzey sistemden alınmak istenen bilgi için geçen zaman grafiği Şekil 6.11'de görülmektedir.

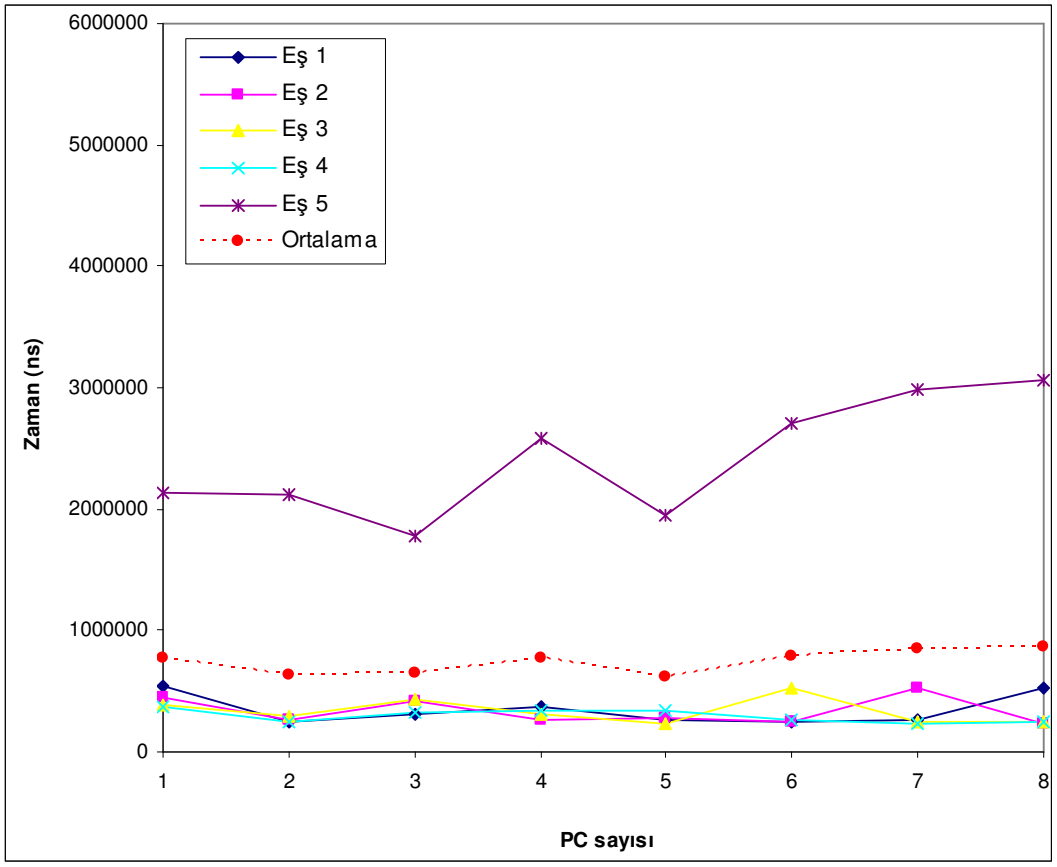



**Şekil 6.11** 5 eşli sistemde, "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman

Şekil 6.11'de gözlemlenen 5 peerlı sistem için eşdüzey sistemden alınmak istenen bilgi için geçen zaman yapısında 61 milyon nanosaniye dolayında gezinmektedir. Bu süreklilik gösteren özelliğin sebebi ajanın sayısının artmasından

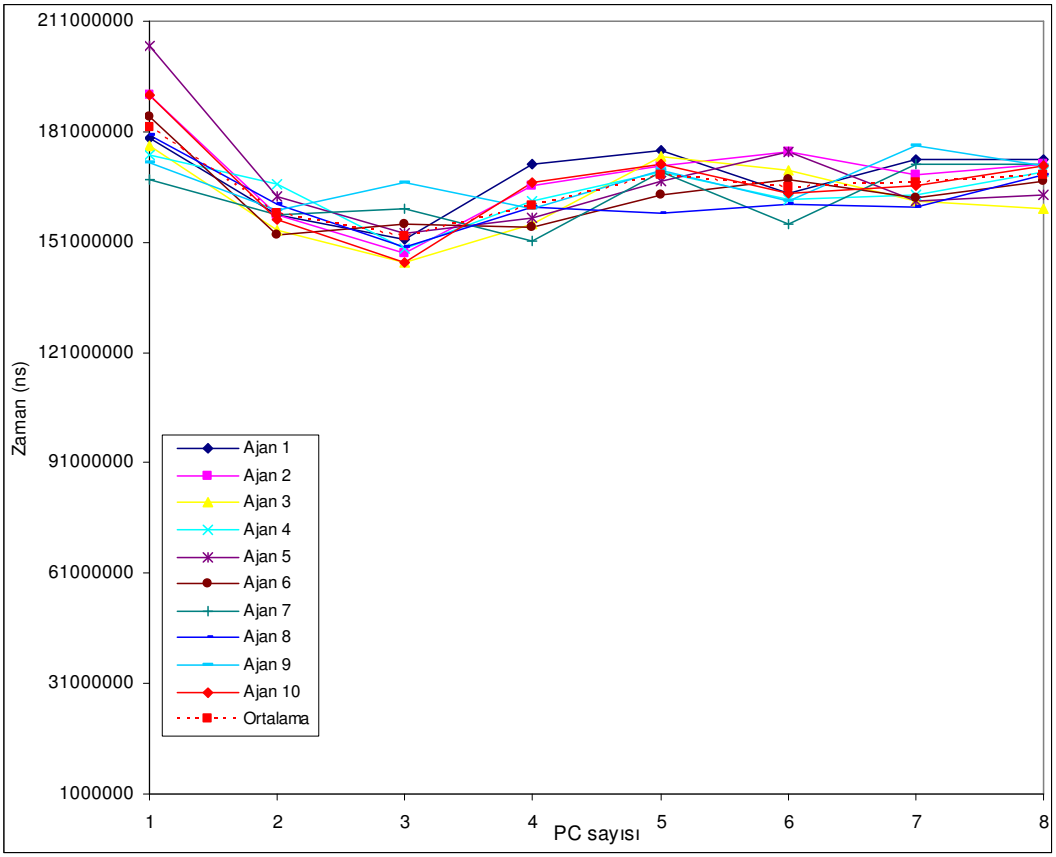
dolayı sistemin bilgisayar sayısı artmasına rağmen benzer süreli zamana ihtiyacı olduğu gözlenmektedir.

Şekil 6.12'de 5 peer çalışan sistemde paylaşılmasına karar verilen bilgi için geçen zaman görülmektedir. Bu sistemde sonuncu peer dışında kalan peerlar benzer özellikler göstermektedir. Diğer taraftan sonuncu peer ise diğer peerlardan dört katından fazla süreye ihtiyaç duymaktadır.




**Şekil 6.12** 5 eşli sistemde, "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman

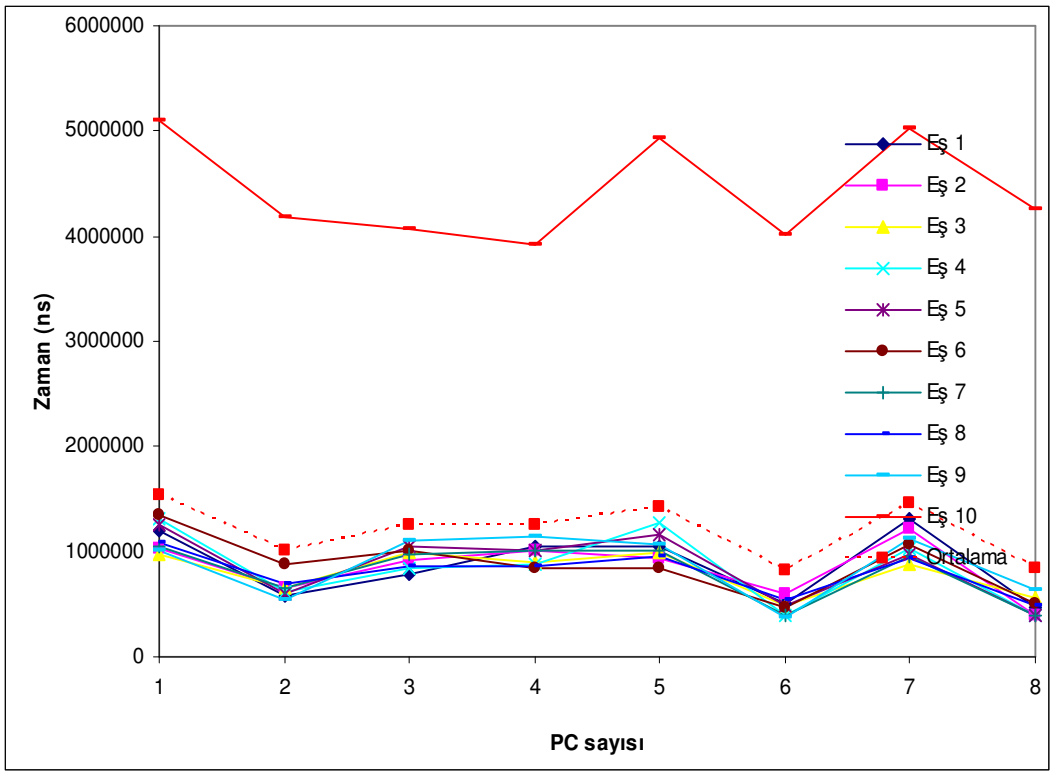
Şekil 6.13'te 10 peer bulunan sistemde eşdüzey sistemden alınmak istenen bilgi için geçen zaman gösterilmektedir.




**Şekil 6.13** 10 eşli sistemde, "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman

10 peerlı sistem için çalıştırılan sistem için olan Şekil 6.13'te görülen grafikte ajanların arasında oluşan varyans aynı performans kriteri için 5 peerlı sisteme göre daha fazladır. Bu bağlamda sistemde bulunan peer sayısının artması ile sistemden alınmak istenen bilgi için geçen zamanlar arasında oluşan zaman farkının arttığı gözlemlenmiştir.

Şekil 6.14'te 10 peer bulunan sistem için oluşan paylaşılmasına karar verilen bilgi için geçen zaman performansı için de 5 peerlı sistemde olduğu gibi sonuncu peer dışındaki peerlar birbirine nispeten yakın değerler üretmişlerdir. Ancak son olan onuncu peer ise diğerlerinden oldukça büyük bir zamana ihtiyaç duymuştur.

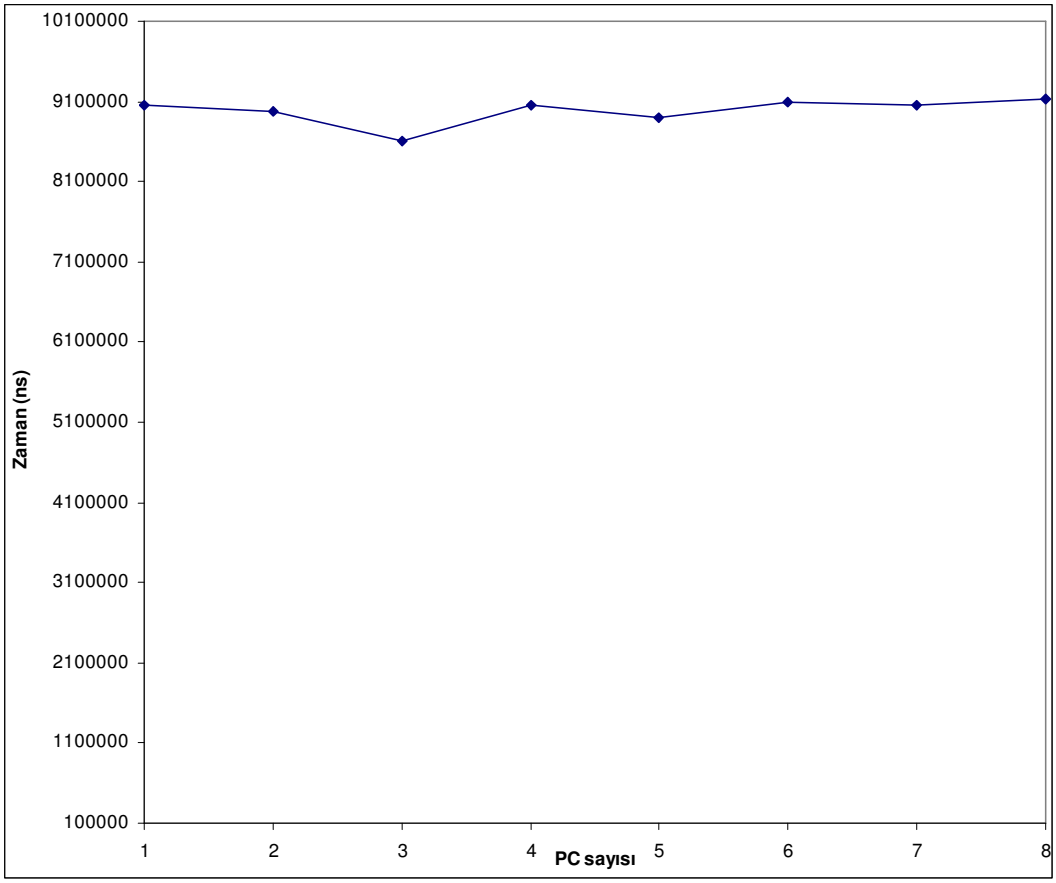



**Şekil 6.14** 10 eşli sistemde, "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman

Şekil 6.14 ile ilgili bir diğer özellik ise eşdüzey sistemden alınmak istenen bilgi için geçen zaman grafiğinde görüldüğü gibi peer sayısı büyüdükçe gerekli zaman miktarları arasındaki varyansın büyümesidir.

#### 6.3.1.2 Aynı Salondaki Bilgisayar Sistemlerinde gsServer Türü GigaSpaces Servisi Kullanan Sistem

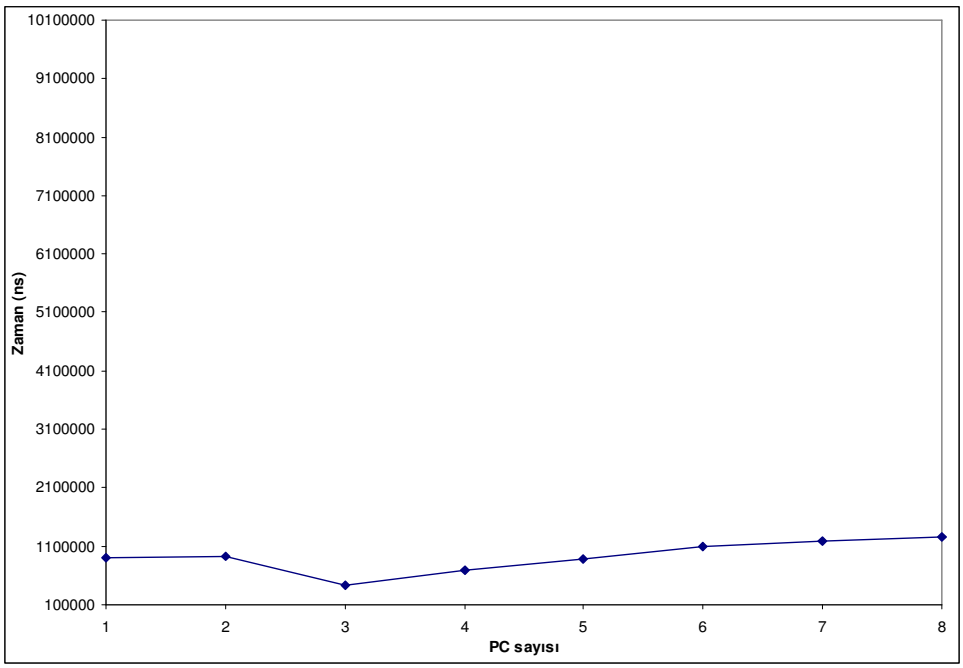
Tek peer kullanılması durumunda gsServer türü GigaSpaces servisi kullanan sistemde eşdüzey sistemden alınmak istenen bilgi için geçen zaman süresi PC sayısı artmasına rağmen benzer özellikler göstermektedir. Şekil 6.15'te gösterildiği şekilde 1 peer için sistemden alınmak istenen bilgi için 9.1 milyon nanosaniye civarında bir süre gerekmektedir.



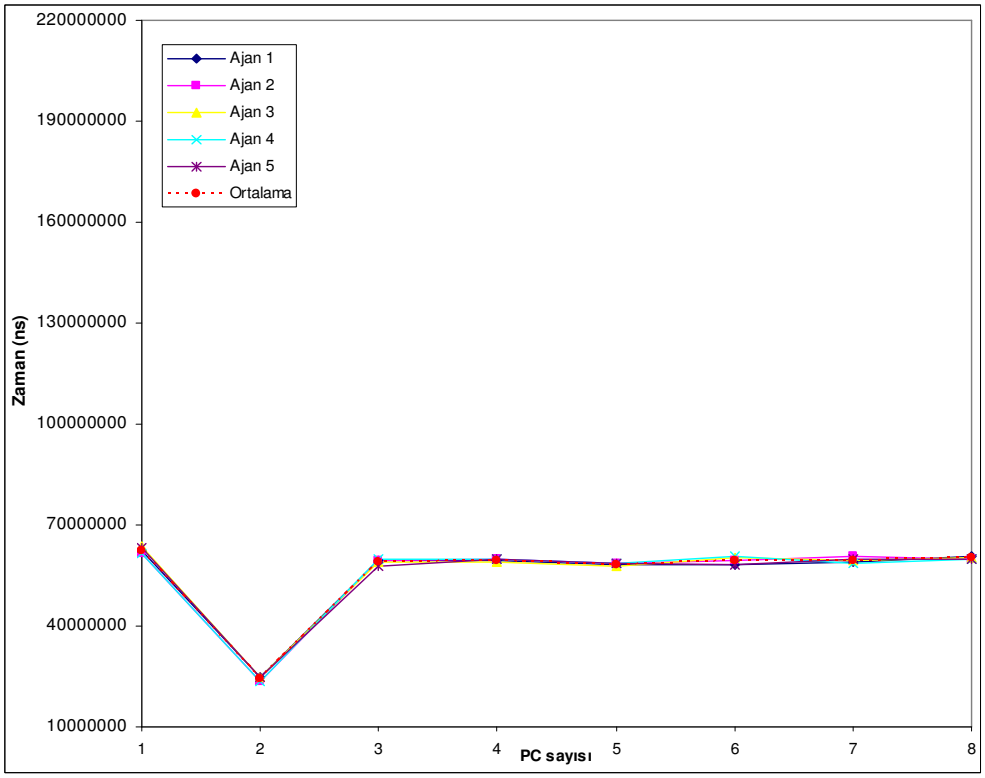

**Şekil 6.15** 1 eşli sistemde "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman

Şekil 6.16'da 1 peer için gsServer türü GigaSpaces servisi kullanıldığında, paylaşılmasına karar verilen bilgi için geçen zaman yapısı görülmektedir. Bu grafikte sistemde bulunan PC sayısının artmasına rağmen sistemin 1.1 milyon nanosaniye dolayında bir zamana ihtiyaç duyduğu gözlemlenmektedir.

Diğer bir taraftan, önceki kısımda performans yapısı incelenen embeded türü GigaSpaces servisi kullanan sistemde oluşana benzer paylaşılmasına karar verilen bilgi için geçen zaman özelliği gözlenmektedir.




**Şekil 6.16** 1 eşli sistemde, "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman

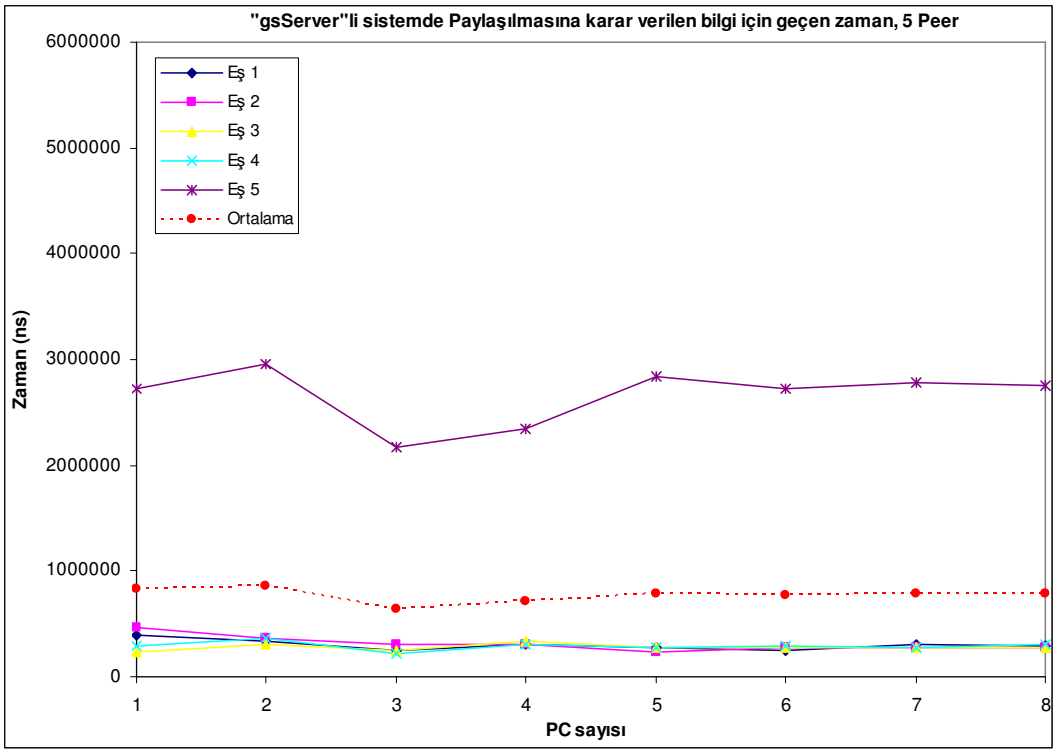



**Şekil 6.17** 5 eşli sistemde "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman

Şekil 6.17'de 5 ajan kullanılması durumunda sistemde eşdüzey sistemden alınmak istenen bilgi için geçen zaman gösterilmektedir. Bu şekilde çıkan yapıya göre sistemin 60 milyon nanosaniyenin üzerinde bir zaman gerektirdiği ortaya çıkmıştır. gsServerli sistem embededli ile kıyaslandığında yaklaşık olarak aynı karakteristiğe sahip oldukları gözlemlenmektedir.

Şekil 6.18'de 5 peerlı sistemde paylaşılmasına karar verilen bilgi için geçen zaman gösterilmektedir. Bu grafikte embeded türü GigaSpaces kullanılması durumundaki gibi ilk dört peer benzer özelikte davranış gösterirken, sonuncu peer ise daha fazla zamana ihtiyaç duymaktadır.

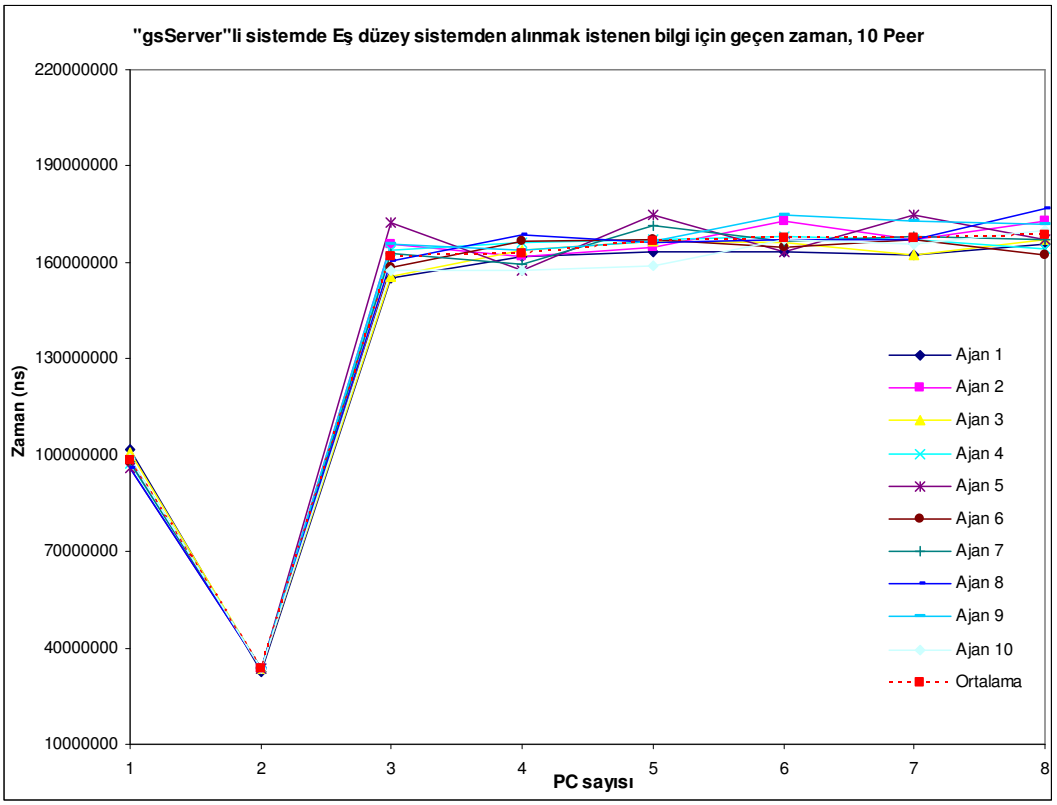
Şekil 6.18 ile aynı testin embededli versiyonu kıyaslandığınsa gsServerli sistemde peerlar arasındaki varyansın daha az olduğu görülmektedir.




**Şekil 6.18** 5 eşli sistemde, "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman

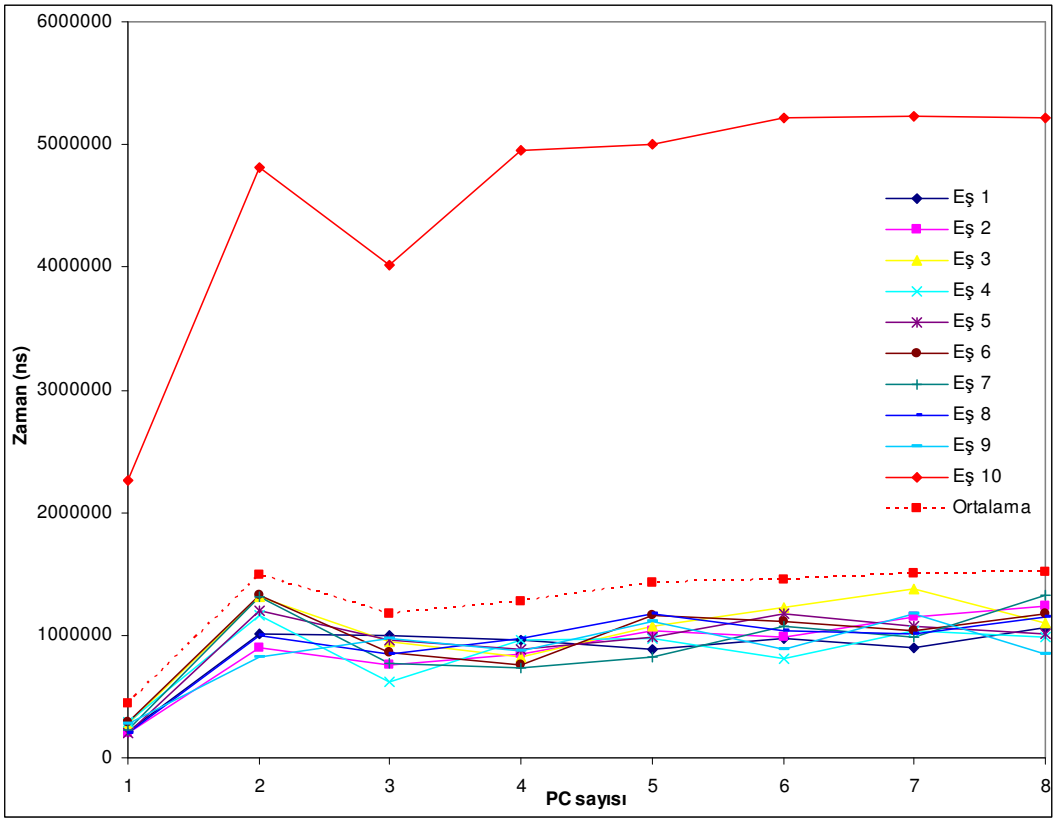
Şekil 6.19'da 10 peer çalışan sistemde gsServer türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman gösterilmektedir. 1 PC için gerekli olan süre 100 milyon nanosaniyeden başlayıp bir dalgalanmadan sonra 170 milyon nanosaniye dolayında bir performans gözlemlenmektedir.

Şekil 6.19 incelendiğinde gsServer'li sistemin ortaya koyduğu performans değerlerinde yine varyasyonun embededli sisteme göre düşük olduğu gözlemlenmiştir.




**Şekil 6.19** 10 eşli sistemde "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman

Şekil 6.20'de 10 peer bulunan sistemde paylaşılmasına karar verilen bilgi için geçen zaman karakteristiği gösterilmektedir. Aynı performans kriterinin birden fazla peer için olan önceki grafiklerinde olduğu gibi sonuncu dışındaki peerlar benzer zamanda işlemi tamamlarlarken sonuncu olan onuncu peer çok daha fazla zamana ihtiyaç duymaktadır. Bu grafikteki diğer bir özellik ise bu test ile aynı testin embededli sistem ile kıyaslandığında, 1 PC için gsServerli sistemin daha iyi performans göstermesi, daha sonra ise her iki sisteminde benzer performans yapısı göstermesidir.




**Şekil 6.20** 10 eşli sistemde, "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman

### 6.4.1 Uzak Bilgisayar Sistemleri için Sistem Performansı

Bu kısımda gerçekleştirilen ikinci deneyler uzak bilgisayar sistemleri için sistem performansının araştırılması şekli ile gerçekleştirilmiştir. Bu deneylerde kullanılan bilgisayarların organizasyonu Şekil 6.2'de gösterildiği şekildedir.

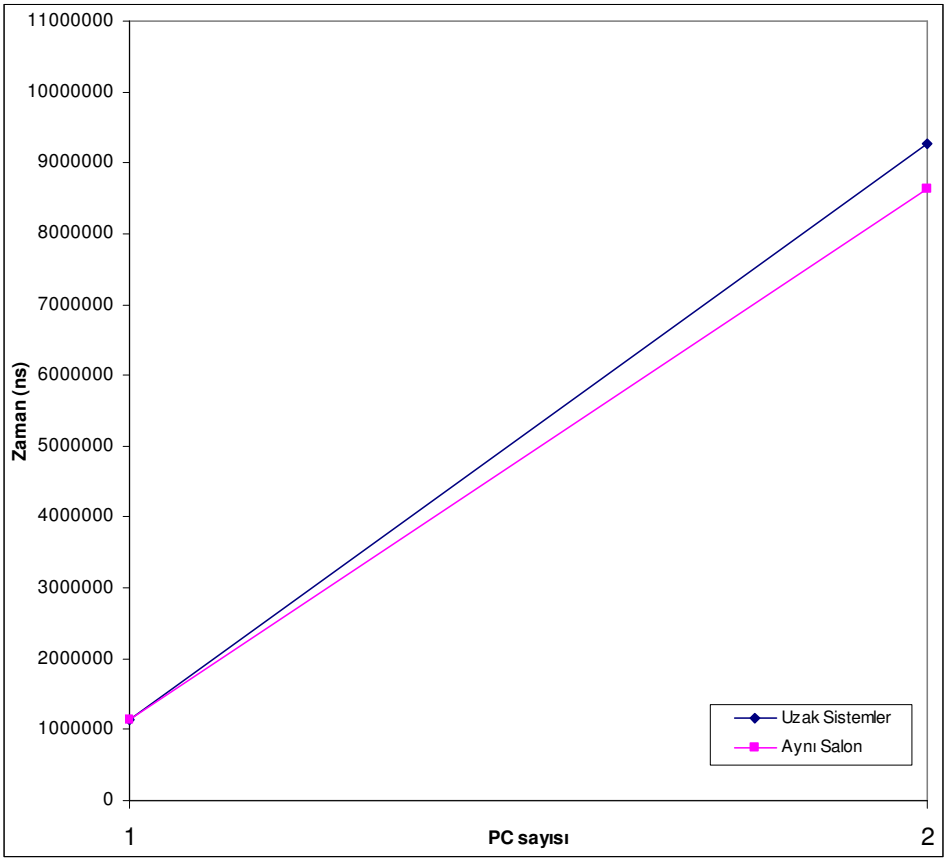
#### 6.4.1.1 Uzak Bilgisayar Sistemlerinde Embeded Türü GigaSpaces Servisi Kullanan Sistem

Embeded türü GigaSpaces servisi kullanan sistem çalışması için önceden bir servis çalıştırılmasına gerek yoktur. GigaSpaces servisini kullanacak program çalıştığı zaman otomatik olarak Embeded space yaratılmaktadır.

Yapılan deneylerde aynı salondaki bilgisayar sistemlerinde olduğu şekilde iki performans ölçümü yapılmaktadır. Bunlardan ilki eşdüzey sistemden alınmak istenen bilgi için geçen zamandır. Bu ilk performans kriteri; sistemden indirilmek istenen bilgi sırasında geçen zamanı göstermektedir. Şekil 6.21'de sistemde 1 peer olması durumunda embeded türü GigaSpaces servisi kullanılarak eşdüzey sistemden alınmak istenen bilgi için geçen zaman gösterilmektedir.

1 PC için gerekli süre, gerek aynı salondaki bilgisayar sistemleri, gerekse uzak sistemler için aynıdır. Bunun sebebi bilgi paylaşımında bulunan sistemlerin aynı bilgisayar üzerinden gerçekleşmesi dolayısı ile ağ üzerinden kaynaklanan ek zaman kaybına gerek duyulmamasıdır.

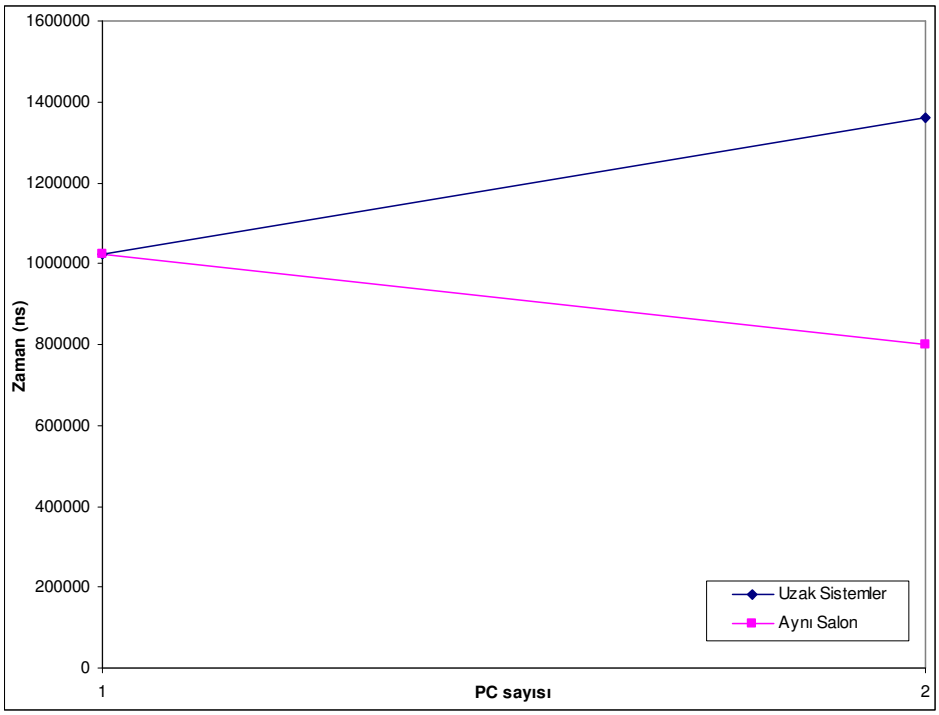
Aynı salondaki bilgisayar sistemleri çalışma sırasında ağın çalışma hızı uzak bilgisayar sistemlerine göre daha düşüktür. Bu bağlamda Şekil 6.21'de görülen yapıda aynı salondaki sistemler için daha düşük zaman geçmesi doğal olarak karşımıza çıkmaktadır.



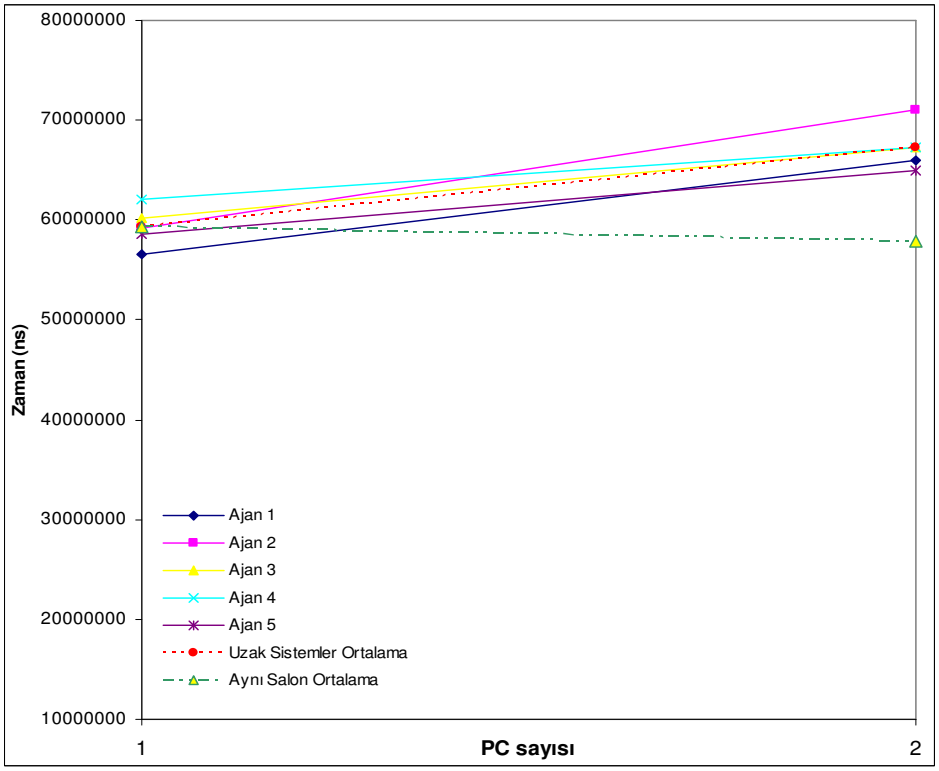

**Şekil 6.21** Uzak bilgisayar sistemleri için 1 eşli sistemde "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman

Şekil 6.22'de paylaşılmasına karar verilen bilgi için geçen zaman özelliği kıyaslanmıştır. Bu bağlamda şekilden de görüldüğü üzere, eşdüzey sistemden alınmak istenen bilgi için geçen zaman da olduğu gibi yine aynı salondaki bilgisayar sistemleri uzak sistemlerden daha az sürede bu işi gerçekleştirebilmişlerdir.

Şekil 6.23'te 5 tane peerın bulunduğu bir sistemde oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman grafiği görülmektedir. Uzak bilgisayar sistemleri için çalışan ajanların genel yapısı birbirine benzerken, aynı salon bilgisayarları ortalaması bu grafikte gösterilen uzak bilgisayar sistemlerinin hepsinden daha düşük olarak yer almaktadır.

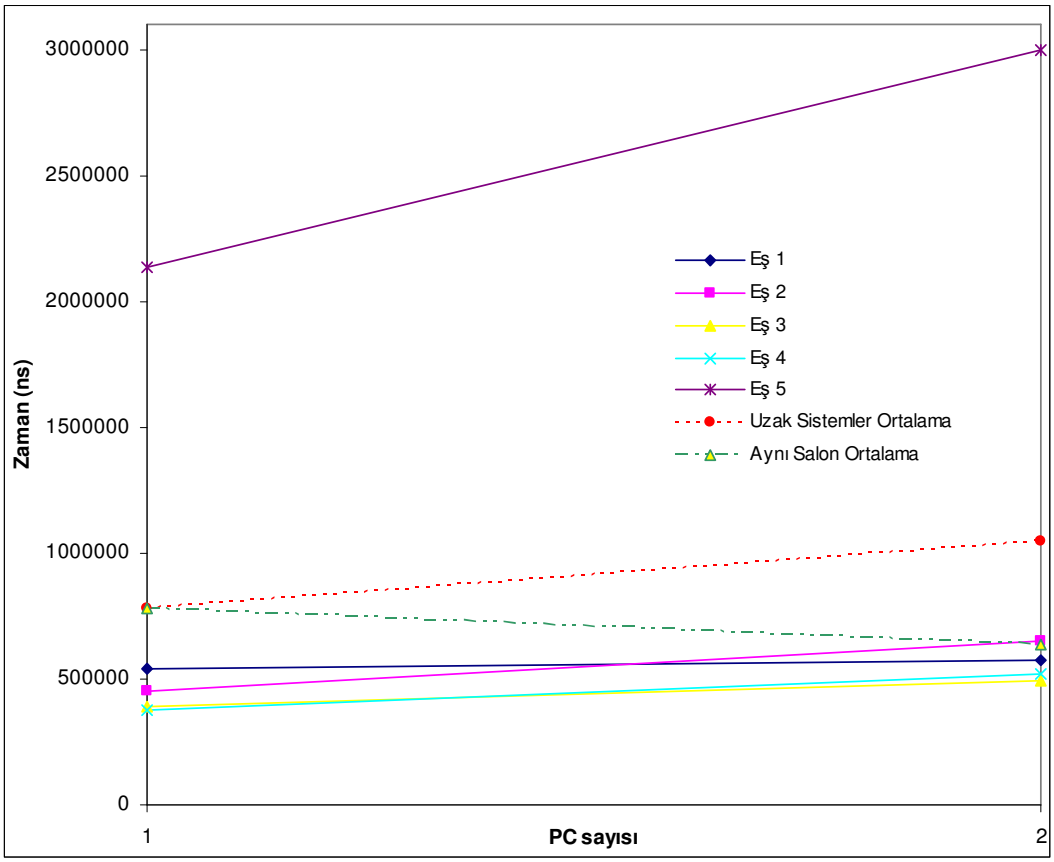



**Şekil 6.22** Uzak bilgisayar sistemleri için 1 eşli sistemde "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman



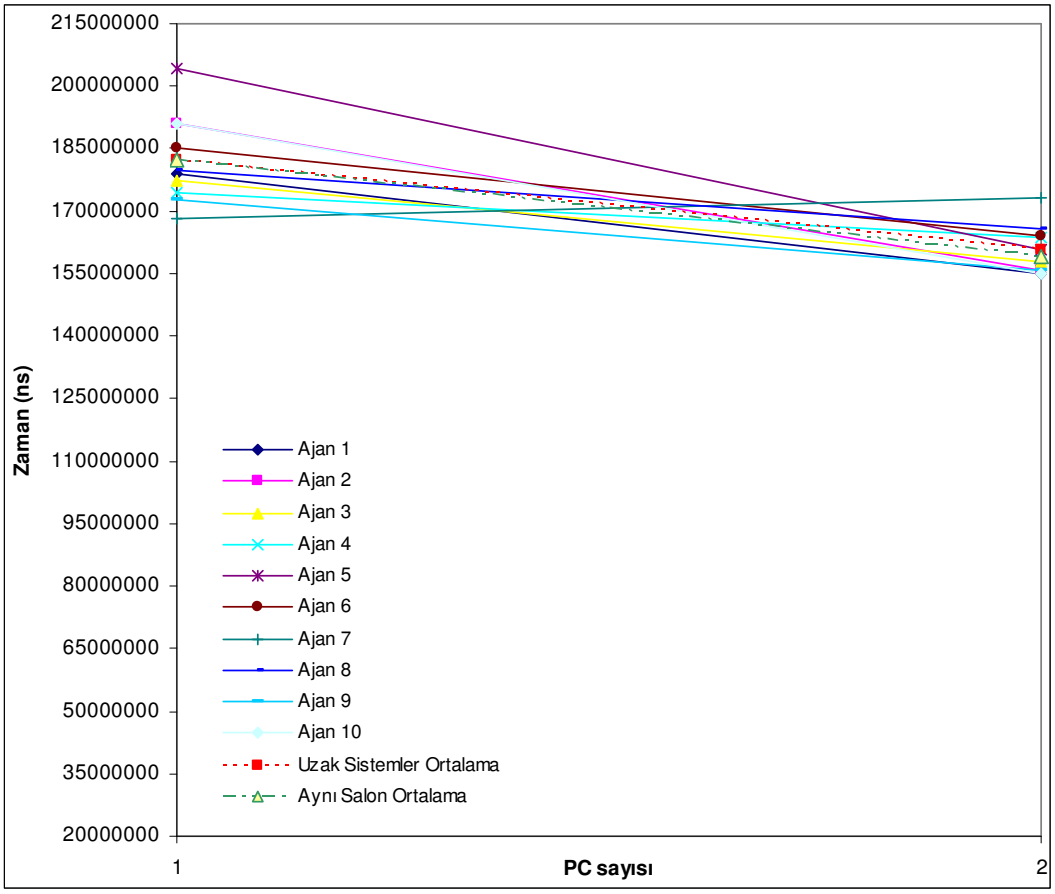

**Şekil 6.23** Uzak bilgisayar sistemleri için 5 eşli sistemde "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman

Şekil 6.24'te paylaşılmasına karar verilen bilgi için geçen zaman grafiği yer almaktadır. Bu grafikte bu kısımdaki diğer testlerde olduğu gibi aynı salondaki bilgisayar sistemleri daha az zamanda işlemlerini tamamlamaktadırlar. İlk dört peera bakıldığında aynı salon ortalamasının altında çıktığı görülse bile bu durumun sebebi ortalamalar alınırken sonuncu (beşinci) peerdan kaynaklanan uzun sürenin ortalamayı yükseltmesidir. Aynı salon ortalaması ile uzak bilgisayar sistemlerinin ortalaması kıyaslandığında, aynı salon için daha az zamana ihtiyaç olduğu görülmektedir.



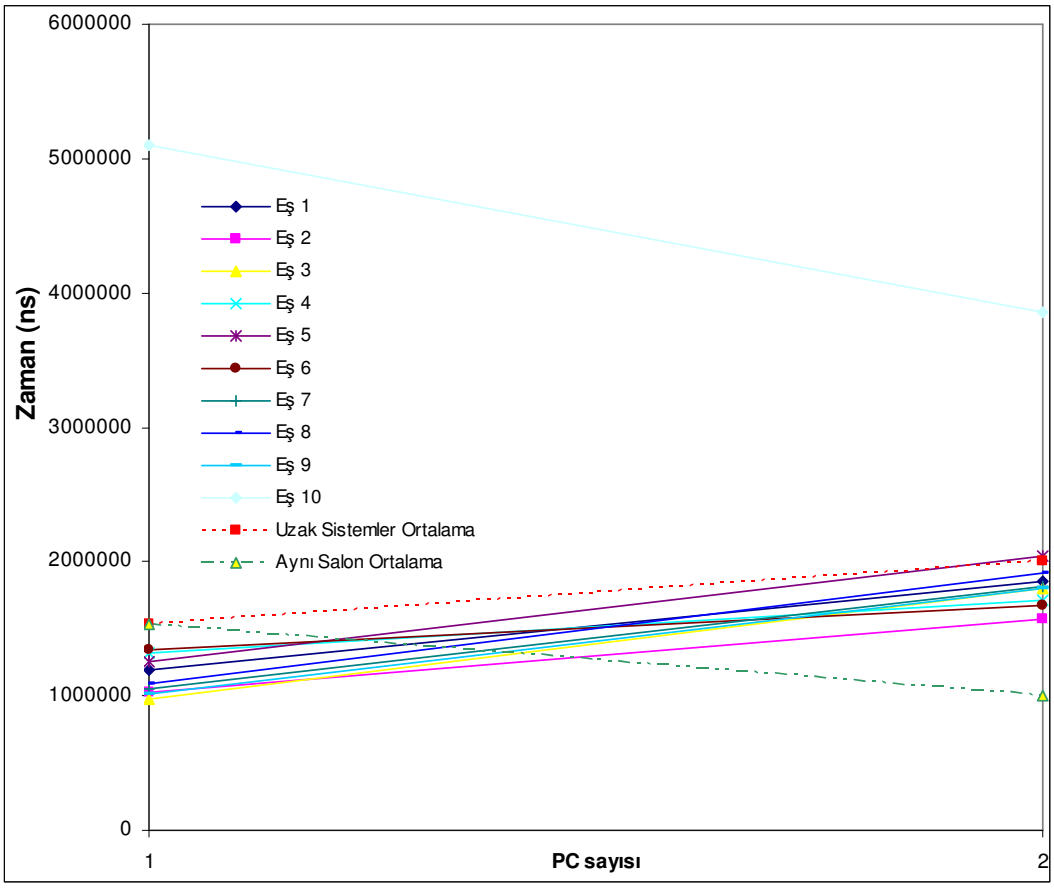

**Şekil 6.24** Uzak bilgisayar sistemleri için 5 eşli sistemde "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman

Şekil 6.25'te eşdüzey sistemden alınmak istenen bilgi için geçen zaman gösterilmektedir. Benzer şekilde aynı salondaki bilgisayarların ortalaması daha düşüktür.




**Şekil 6.25** Uzak bilgisayar sistemleri için 10 eşli sistemde "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman

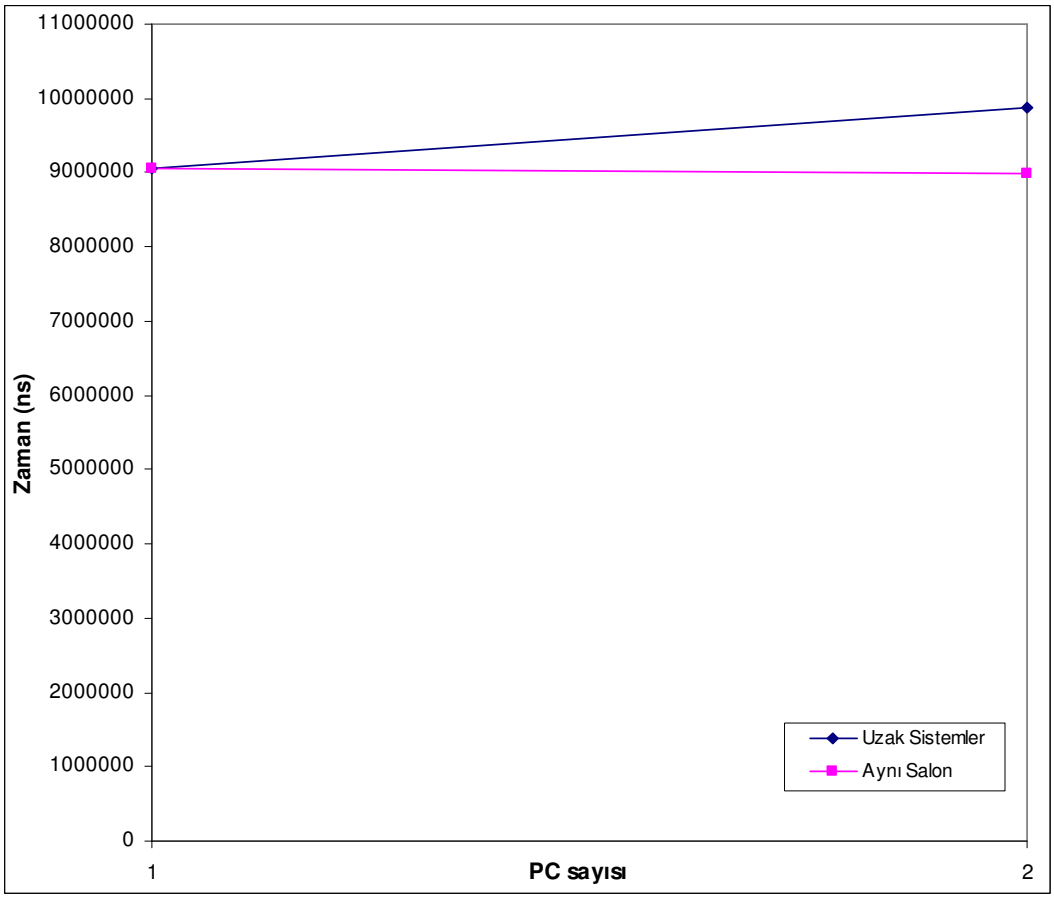
Şekil 6.26'da 10 peerın çalıştığı sistemde paylaşılmasına karar verilen bilgi için geçen zaman grafiği gösterilmektedir. Aynı salon ortalaması grafikte yer alan bütün değerlerden daha düşük yer alarak diğer testler ile paralellik sağlamaktadır.




**Şekil 6.26** Uzak bilgisayar sistemleri için 10 eşli sistemde "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman

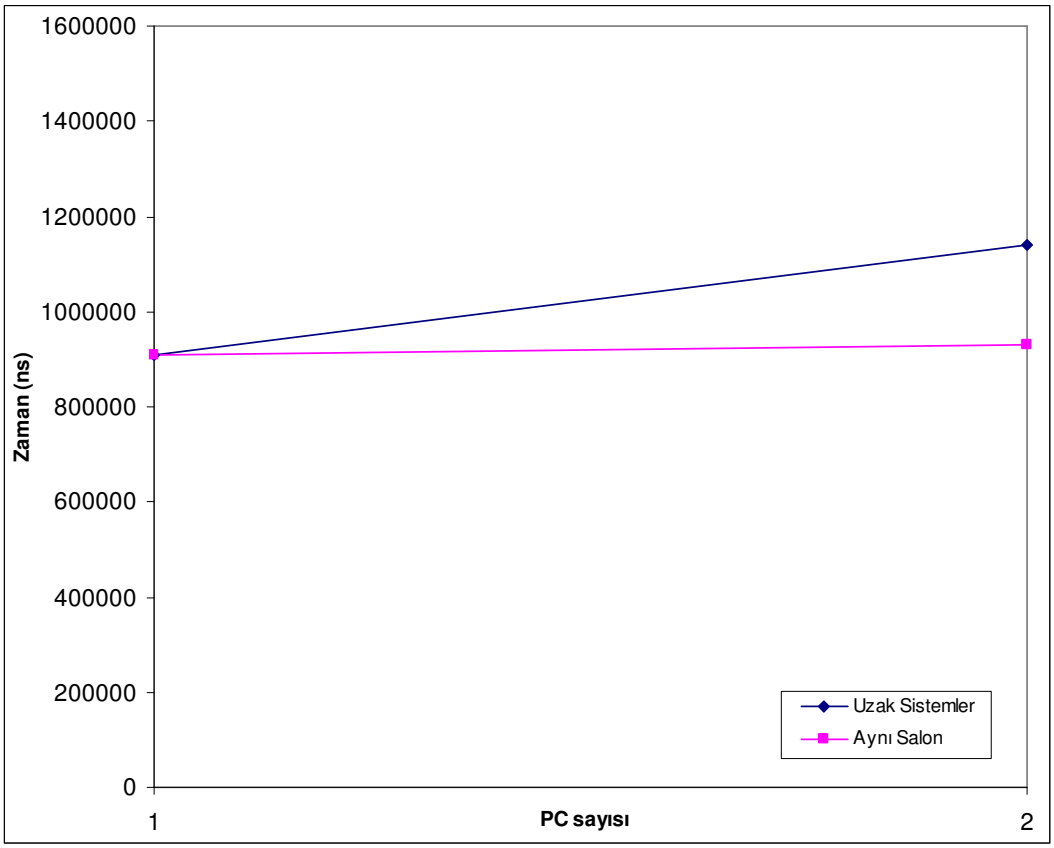
#### 6.4.1.2 Uzak Bilgisayar Sistemlerinde gsServer Türü GigaSpaces Servisi Kullanan Sistem

Bu kısımda gsServer kullanan sistem için uzak bilgisayar sistemleri ile aynı salon bilgisayar sistemlerinin kıyaslaması incelenmektedir. Şekil 6.27'de 1 peer bulunan sistemde eşdüzey sistemden alınmak istenen bilgi için geçen zaman gösterilmektedir. Beklenildiği şekilde aynı salondaki bilgisayar sistemleri daha kısa sürede işlemi tamamlamaktadır.



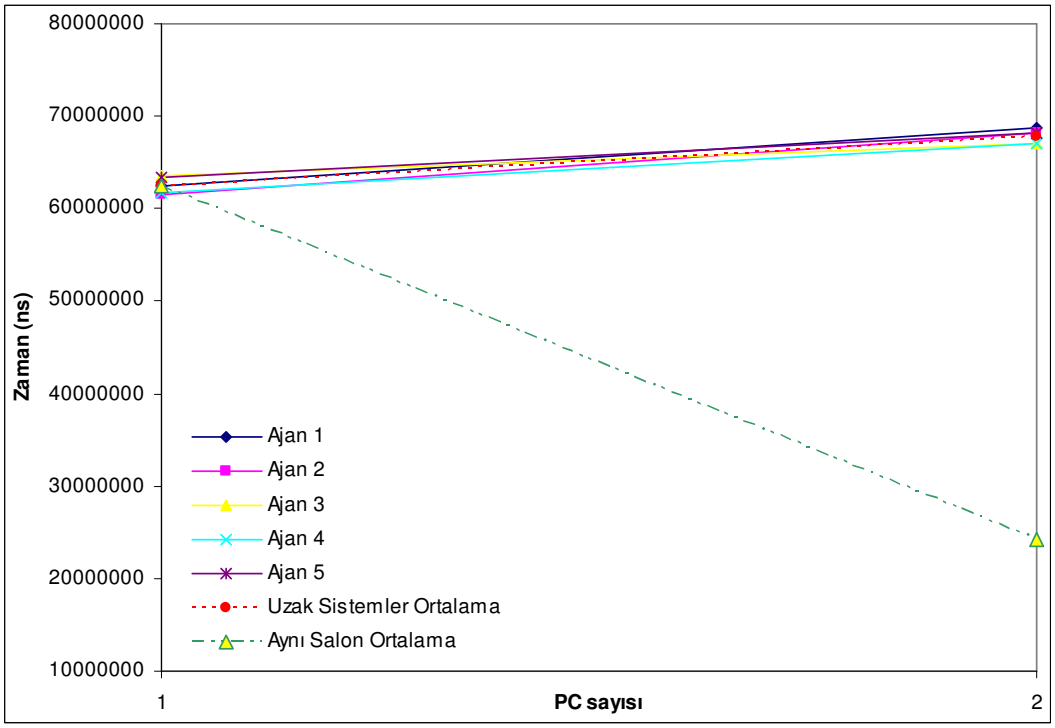

**Şekil 6.27** Uzak bilgisayar sistemleri için 1 eşli sistemde "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman

Şekil 6.28'de 1 peer kullanılarak paylaşılmasına karar verilen bilgi için geçen zaman grafiği gösterilmektedir. Bu grafikte de aynı salondaki bilgisayar sistemleri ağdaki geçen sürenin daha az olmasından kaynaklanan bir performans avantajına sahiptir. Bu bağlamda aynı salondaki bilgisayar sistemleri daha az zamanda işlemi tamamlayabilmektedir.



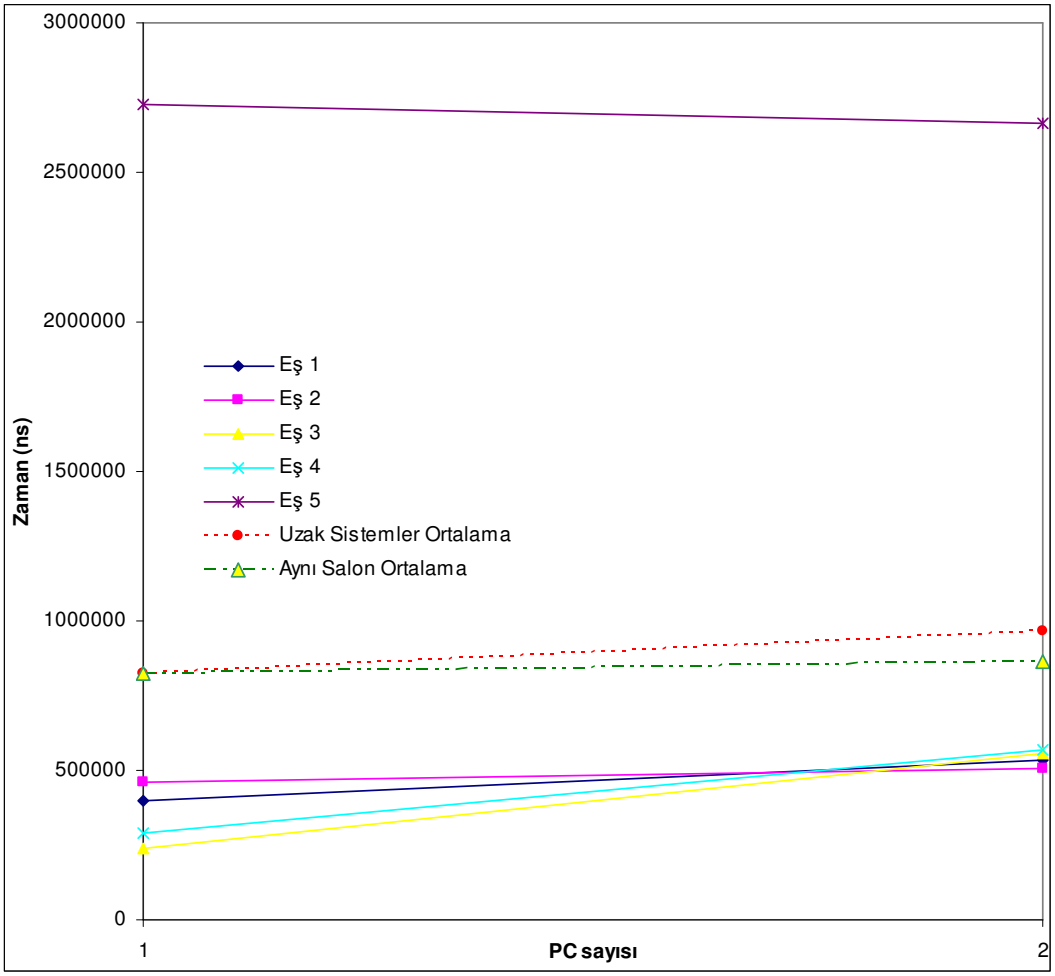

**Şekil 6.28** Uzak bilgisayar sistemleri için 1 eşli sistemde "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman

Şekil 6.29'da beş peer için eşdüzey sistemden alınmak istenen bilgi için geçen zaman grafiği gösterilmektedir. Bu testde de aynı salondaki bilgisayar sistemleri daha başarılı bir performans gösterip, daha az zamanda işlemin tamamlanması mümkün olmaktadır.



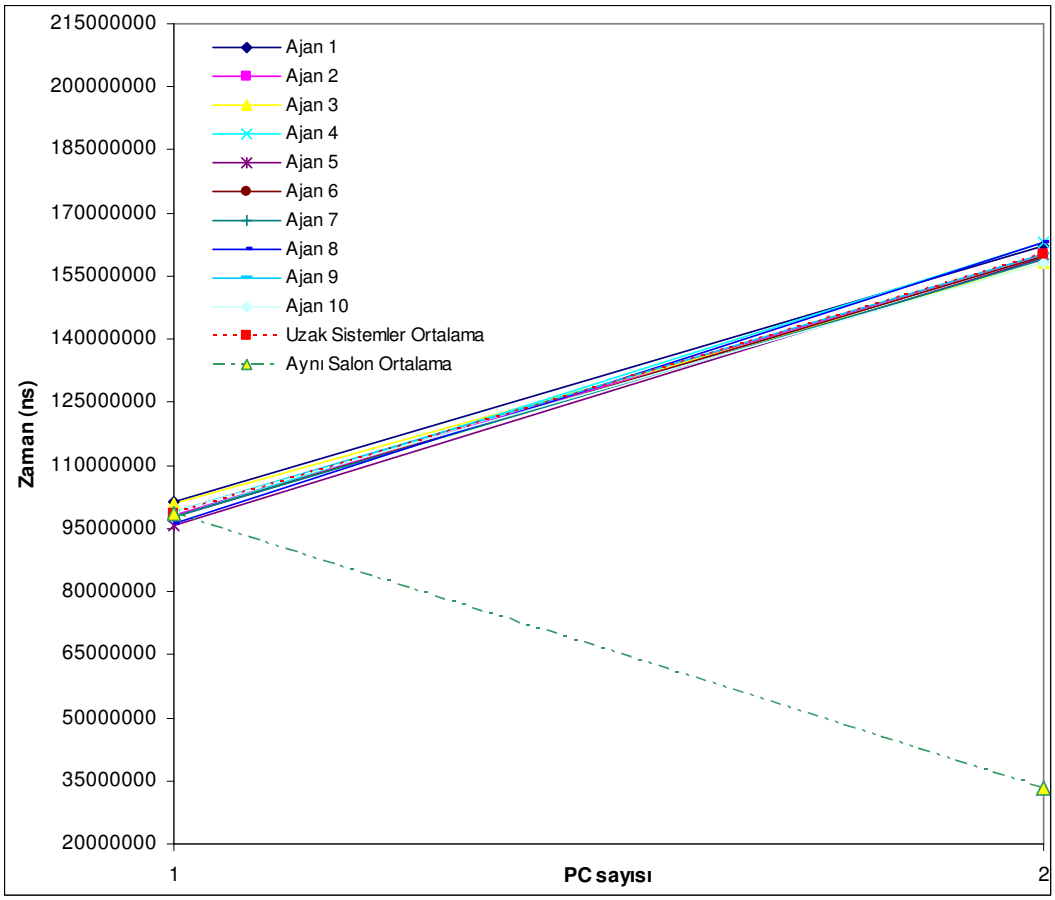

**Şekil 6.29** Uzak bilgisayar sistemleri için 5 eşli sistemde "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman

Şekil 6.30'da paylaşılmasına karar verilen bilgi için geçen zaman grafiği yer almaktadır. Aynı salondaki bilgisayar sistemleri ortalaması, uzak sistemler ortalamasından daha az süreye gereksinim duymaktadır. Aynı testin embeded'li versiyonunda gerçekleştiği şekilde peerlerın tek tek incelendiğinde düşük olmasına rağmen ortalama olarak daha yüksek çıkma sebebi son peer olan (beşinci) peerın yüksek zamana ihtiyaç duyup ortalamayı yükseltmesi söz konusudur.



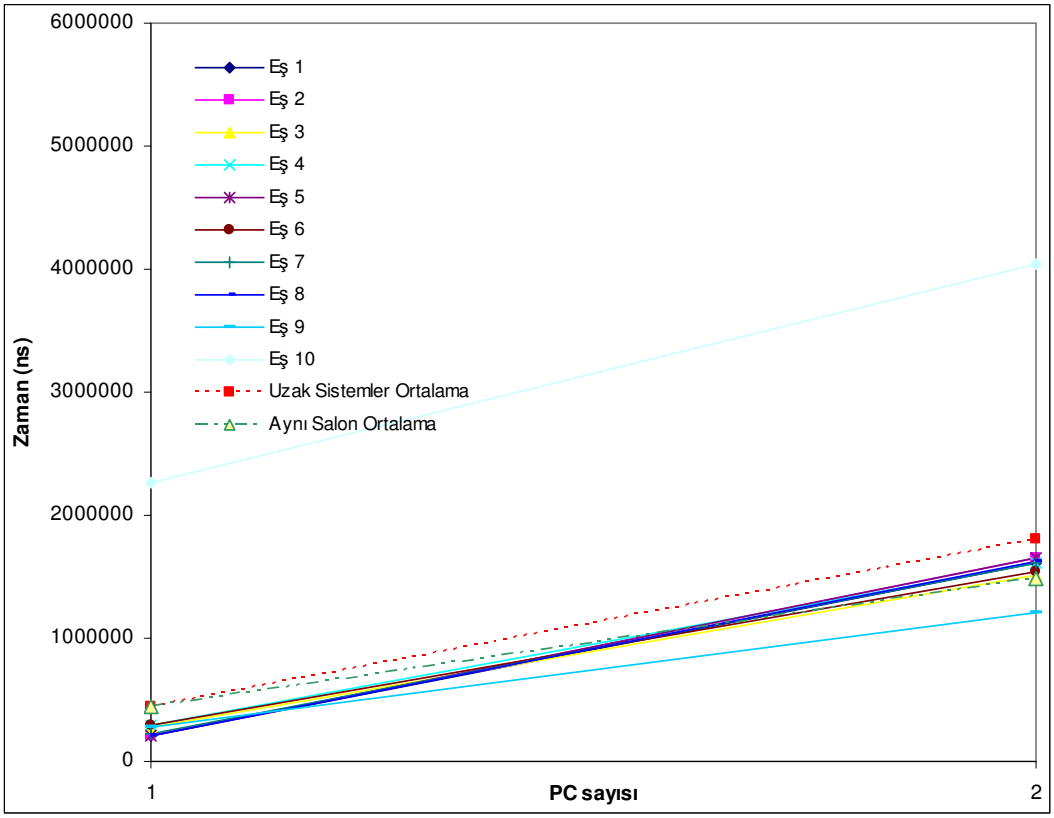

**Şekil 6.30** Uzak bilgisayar sistemleri için 5 eşli sistemde "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman

Şekil 6.31'de eşdüzey sistemden alınmak istenen bilgi için geçen zaman gösterilmektedir. Bu kısımdaki diğer testlerdeki şekilde aynı salondaki bilgisayar sistemlerinin ortalaması daha düşüktür.




**Şekil 6.31** Uzak bilgisayar sistemleri için 10 eşli sistemde "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan eşdüzey sistemden alınmak istenen bilgi için geçen zaman

Şekil 6.32'de 10 peer bulunan sistemde paylaşılmasına karar verilen bilgi için geçen zaman grafiği gösterilmiştir. Aynı salon bilgisayarları için gerçekleşen zaman diğer testlerde olduğu şekilde daha düşük çıkmıştır.




**Şekil 6.32** Uzak bilgisayar sistemleri için 10 eşli sistemde "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan paylaşılmasına karar verilen bilgi için geçen zaman

Bu kısımda yapılan testlerin tamamında aynı salondaki bilgisayar sistemleri için gerçekleşen performans uzak bilgisayar sistemlerinden daha iyi gerçekleşmiştir. Bunun anlamı bu kısımdaki bütün testlerde aynı salonda yapılan testlerde kablosuz ağın hızından kaynaklanan etkinin daha olumlu yani hızlı olmasıdır. Diğer taraftan geliştirilen sistemin her iki mimaride de (aynı salon ve uzak bilgisayarlar) benzer şekilde davranış gösterdiği gözlemlenmiştir.

### 6.5.1 Trafik İncelemesi

Normal bir eşdüzey uygulamasında, yaratılan trafiğin önemli bir bölümü istenen bilgileri arama işlemi sırasında geçmektedir [Hu, H.T.]. Yapılan bilgi transferinin içeriğine bakmaksızın bilgi transferindeki mesajları gelen ve giden olmak üzere iki gruba ayırmak mümkündür. Geliştirilen sistemimizde son derece az miktarda sisteme dahil olma bilgisi yollamak gerekmektedir. Bu bağlamda peer sadece yeni bir bilgi paylaşmak istediğinde bu yeni paketleri ajana göndermektedir. Oysaki standard bir eşdüzey programda her yeni bilgi isteme talebi geldiğinden peer tekrar tekrar aynı bilgileri göndermek zorundadır.

$$(Z_gön + Z_al) \leq St_z \qquad (1)$$

Z_gön: Paylaşılmasına karar verilen bilgi için geçen zaman.
Z_al: Eşdüzey sistemden alınmak istenen bilgi için geçen zaman.
St_z: Standart Eşdüzey Sistem için geçen süre

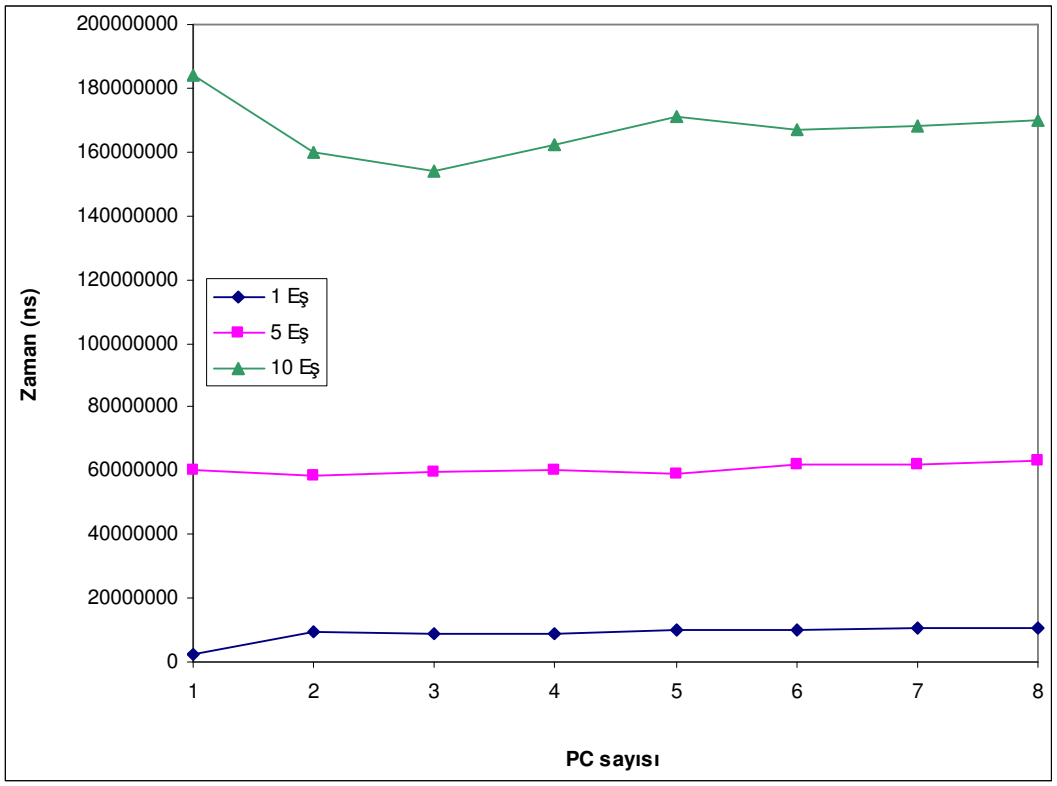
Yukarıdaki (1) numaralı eşitlikte Z_gön olarak gösterilen, "paylaşılmasına karar verilen bilgi için geçen zaman" peer ile ortak hareket edecek ajanın bulunduğu bilgisayara birkez yüklendiği için bu bilginin defalarca başka peerlar tarafından alınması bu bilgiyi paylaşan peerı etkilemeyecektir. Dolayısı ile paylaşılmasına karar verilen bilginin getireceği ek maliyet sadece birkez ajana o bilginin gönderilmesinden ibarettir. Bundan sonra o bilginin başka peerlar tarafından istenip istenmemesi, o peer için yeni bir trafik yaratmamaktadır.

Aynı eşitlikte Z_al şeklinde gösterilen, "eşdüzey sisteminden alınmak istenen bilgi için geçen zaman" sadece bir talep mesajı (ajana yapılmakta) ve bu talepten gelecek bir cevap mesajı olmak üzere yalnızca iki mesajdan ibarettir. Bu durumda, peer sistemden bilgi indirmek istediğinde de yalnızca bu isteği bir mesaj ile iletmekte, gelecek sonuçta yalnızca 1 mesaj olarak geri gelmektedir.

Dolayısı ile dizayn edilen sistem, mobil ve/veya kablosuz bir cihazda çalışmakta olan bir peer için, standart bir eşdüzey sistemine göre çok daha az bir mesaj trafiği yaratması söz konusudur.

### 6.5.1.1 Aynı Salondaki Bilgisayar Sistemleri için Trafik İncelemesi

Bu bilgiler ışığında deneysel sistemde yapılan çalışmalar sonucunda şu şekilde sonuçlar elde edilmiştir. Şekil 6.33'te aynı salondaki bilgisayar sistemleri için embeded türü GigaSpaces servisi kullanılması durumunda oluşan mesaj trafiği gösterilmektedir. Bu grafikte görüldüğü üzere üç farklı peer sayısı için inceleme yapılmıştır. Sırası ile 1 peer, 5 peer ve 10 peerlı sistem için oluşturulan 6.33'teki grafikte enerji tüketimleri sırası ile 1 peer, 5 peer ve 10 peer için artmaktadır.

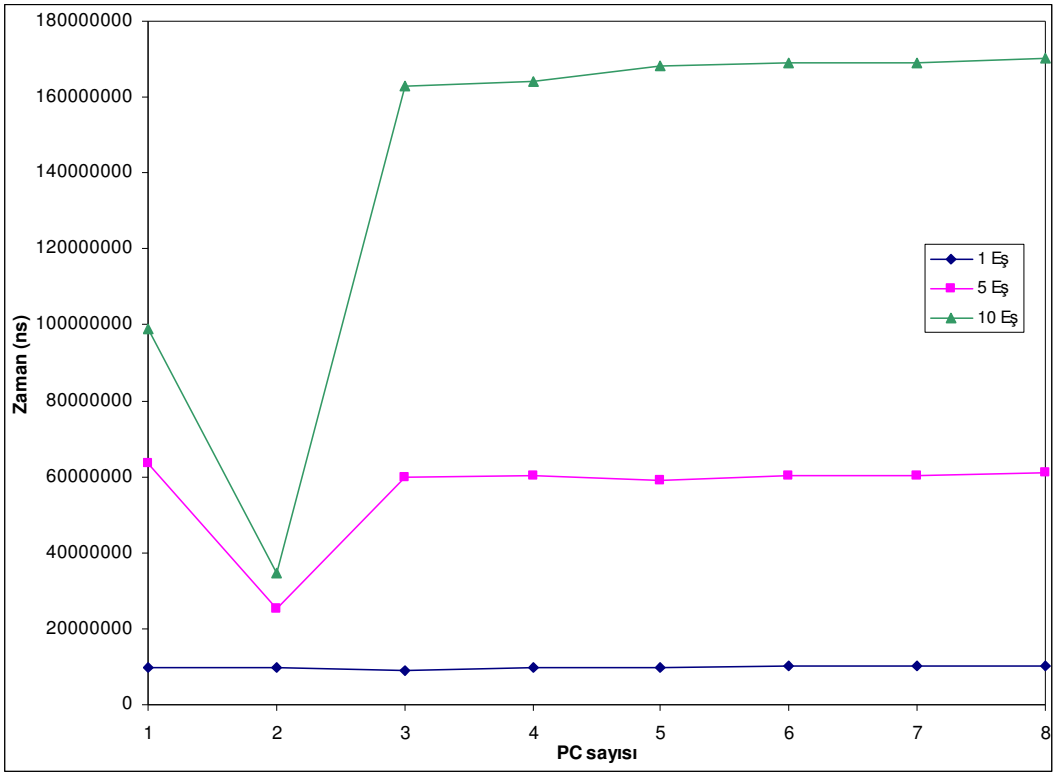



**Şekil 6.33** Aynı salondaki bilgisayar sistemleri için "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan mesaj trafiği

Şekil 6.34'te de Şekil 6.33 ile aynı testin gsServer türü GigaSpaces servisi ile denenmiş hali görülmektedir. Bu grafikten görüldüğü şekli ile ilk iki PC için oluşan mesaj trafiği nispeten azken, 3 PC den itibaren benzer miktarda mesaj trafiğinin

yaratıldığı gözlemlenmektedir.

Şekil 6.33 ile Şekil 6.34 karşılaştırıldığında gsServer kullanılan sistemin benzer mesaj trafiğine yol açmak ile birlikte kısmen daha yüksek değerlere ulaştığı gözlemlenmektedir.



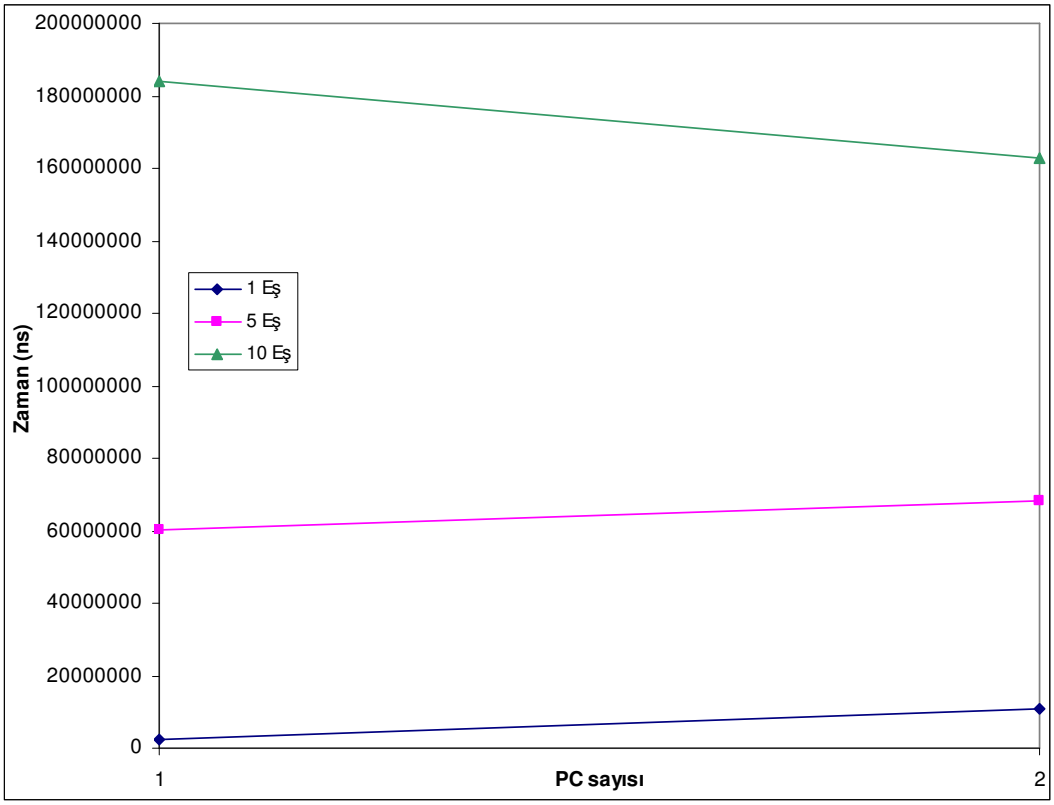

**Şekil 6.34** Aynı salondaki bilgisayar sistemleri için "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan mesaj trafiği

### 6.5.1.2 Uzak Bilgisayar Sistemleri için Trafik İncelemesi

Bir önceki bölümde incelenen aynı salondaki bilgisayar sistemleri için oluşan mesaj trafiğinden sonra, bu kısımda da uzak bilgisayar sistemleri için oluşan trafik incelemesi yapılmaktadır.

Şekil 6.35'te 1 peer ve 5 peerlı sistemler için bilgisayar sayısı arttıkça mesaj

trafiğinde de bir artış gözlenmektedir. Diğer taraftan 10 peer için bir azalma olmaktadır. Aynı salondaki bilgisayar sistemleri için yapılan aynı testler ile kıyaslanıldığında yine 10 peer için bir düşüş söz konusu olduğu diğerleri için ise 5 peer için sonucun aynı kaldığı, 1 peer için ise bir artış söz konusu olduğu gözlemlenmiştir. Bu bağlamda Şekil 6.33 ile Şekil 6.35'teki sonuçların 1 peer ve 10 peer için benzer, 5 peer için ise birbirlerine yakın özellikte olduğu gözlemlenmiştir.

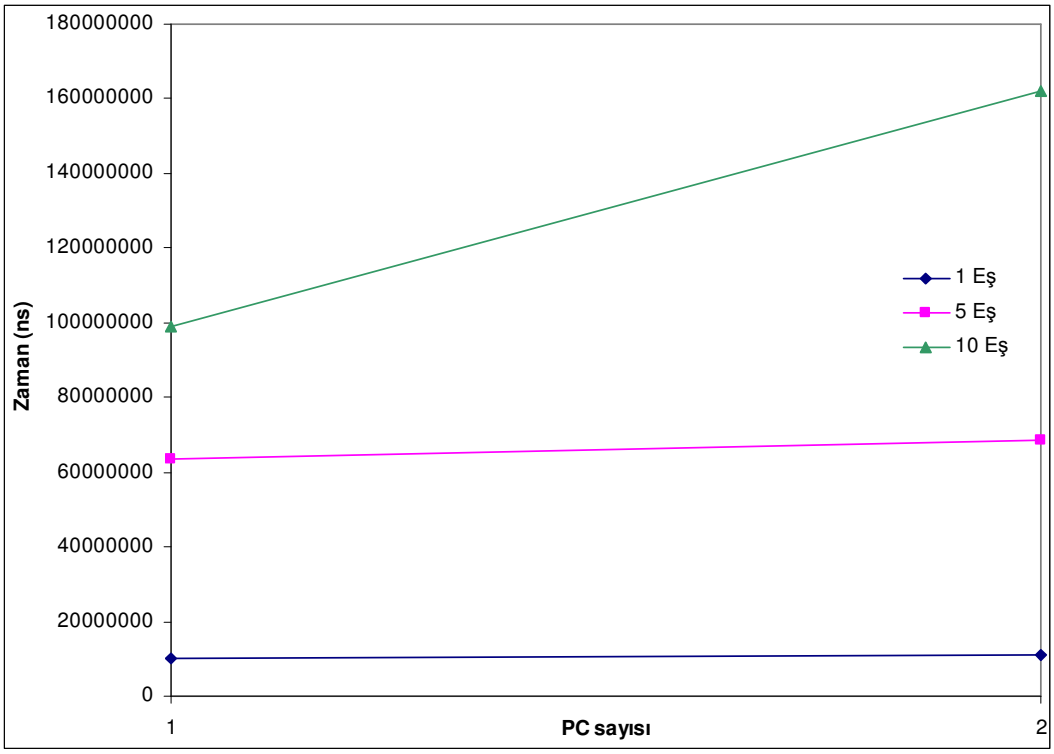



**Şekil 6.35** Uzak bilgisayar sistemleri için "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan mesaj trafiği

Şekil 6.36'da uzak bilgisayar sistemleri için gsServer türü GigaSpaces servisi kullanılması durumunda oluşan mesaj trafiği sırası ile 1, 5 ve 10 peerlık sistemler için gösterilmiştir. 1 peer ve 5 peer için oluşan mesaj trafiğinin çok kısıtlı artış gösterdiği

bölümde 10 peer için oluşan trafik büyük ölçüde arttığı gözlemlenmiştir.

Şekil 6.35 ile Şekil 6.36 karşılaştırıldığı zaman embeded türü GigaSpaces servisi kullanan sistemin mesaj trafiğinde düşüş gözlenebildiği görülmektedir.




**Şekil 6.36** Uzak bilgisayar sistemleri için "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan mesaj trafiği

Bu kısımda trafik ile ilgili yapılan gözlemler sonucunda, embeded türü GigaSpaces servisinin tercih edilmesi sistemde oluşan mesaj trafiğini kısmen düşürebileceği gözlemlenmiştir. Özellikle, sistemde yer alan peer miktarının yüksek olması durumunda embeded türü sistemin kullanılması oluşacak mesaj trafiğini kısmen düşürmektedir.

### 6.5.2 Enerji İncelemesi

Sistemimizin enerji incelemesi kısmı için yaygın olarak kullanılan bir enerji ölçme sistemi olan AIE (Actual Interface Energy) modelini kullanacağız. Bu modelde enerji bilgi gönderimi, bilgi alımı ve hazırda bekleme konumlarında harcanan enerji incelenmektedir. Model şu şekilde gösterilebilmektedir.

$$\begin{aligned} AIE &= E_gön + E_al + E_bekle \\ &= (Z_gön * P_gön) + (Z_al * P_al) + (Z_bekle * P_bekle) \end{aligned} \quad (2)$$

Z_gön = Paket gönderme süresi
Z_al = Paket alma süresi
Z_bekle = Paket gönderme ve alma yapılmayan süre (bekleme süresi)
P_gön = Paket gönderme için harcanan güç
P_al = Paket alma için harcanan güç
P_bekle = Paket gönderme ve alma yapılmayan (bekleme) süre içinde harcanan enerji

Yukarıda (1) ve (2) numaralı eşitlikler beraber incelendiğinde Geliştirilen sistemin enerji harcama eşitliği şu şekilde hesaplanmaktadır.

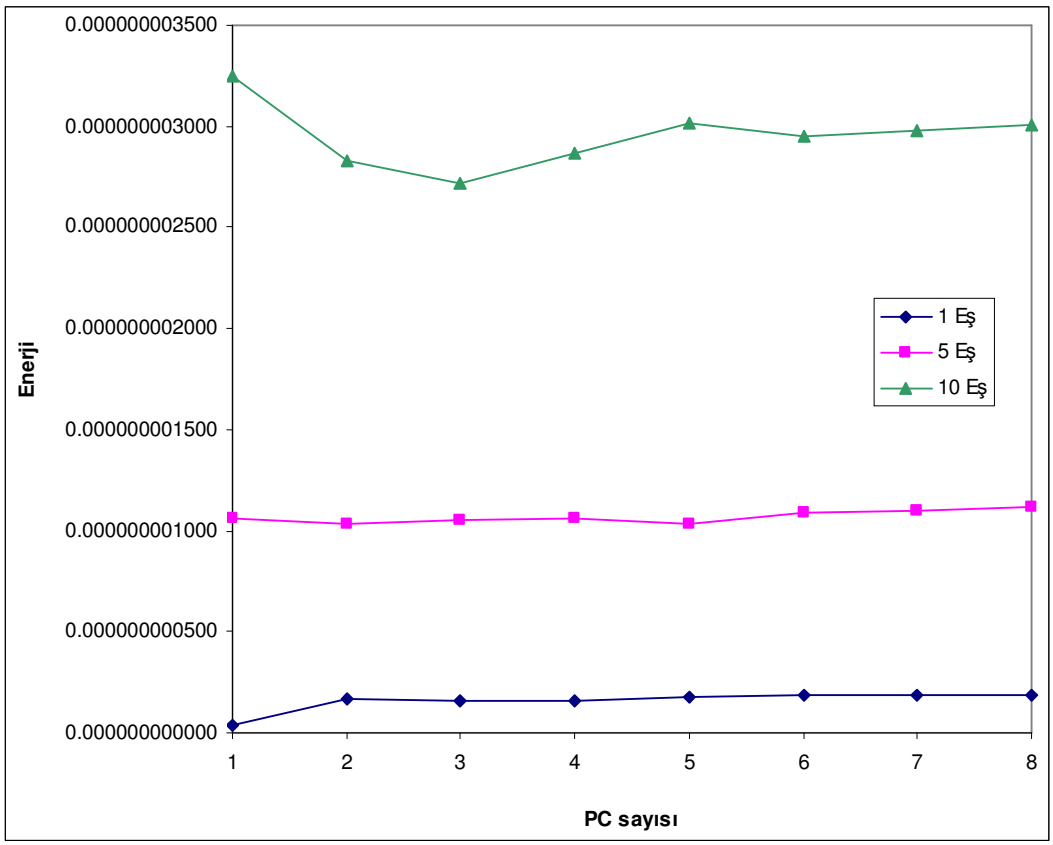
$$E_ATLIG = \left[ \left( \frac{Z_gön + Z_al}{Bantgenisligi} \right) * (P_gön) \right] + [Z_bekle * P_bekle] \quad (3)$$

#### 6.5.2.1 Aynı Salondaki Bilgisayar Sistemleri için Enerji İncelemesi

Bu kısımda aynı salondaki bilgisayar sistemleri için enerji incelemesi gösterilmektedir. Şekil 6.37'de aynı salondaki bilgisayar sistemleri için embeded türü

GigaSpaces servisi kullanılması durumunda oluşan enerji tüketimi gösterilmiştir. Bu grafik ve bu bölümdeki diğer grafiklerin hazırlanması sırasında yapılan enerji hesaplamaları için peerların çalıştığı bilgisayarlardan sürekli enerji yayılımı yapıldığı, bunun sonucunda da hiç bekleme süresi olmadığı kabul edilmiştir. Bu bağlamda (3) numaralı formülde eşitliğin beklemeden gelen tarafı olan $[Z_bekle * P_bekle] = 0$ dır.

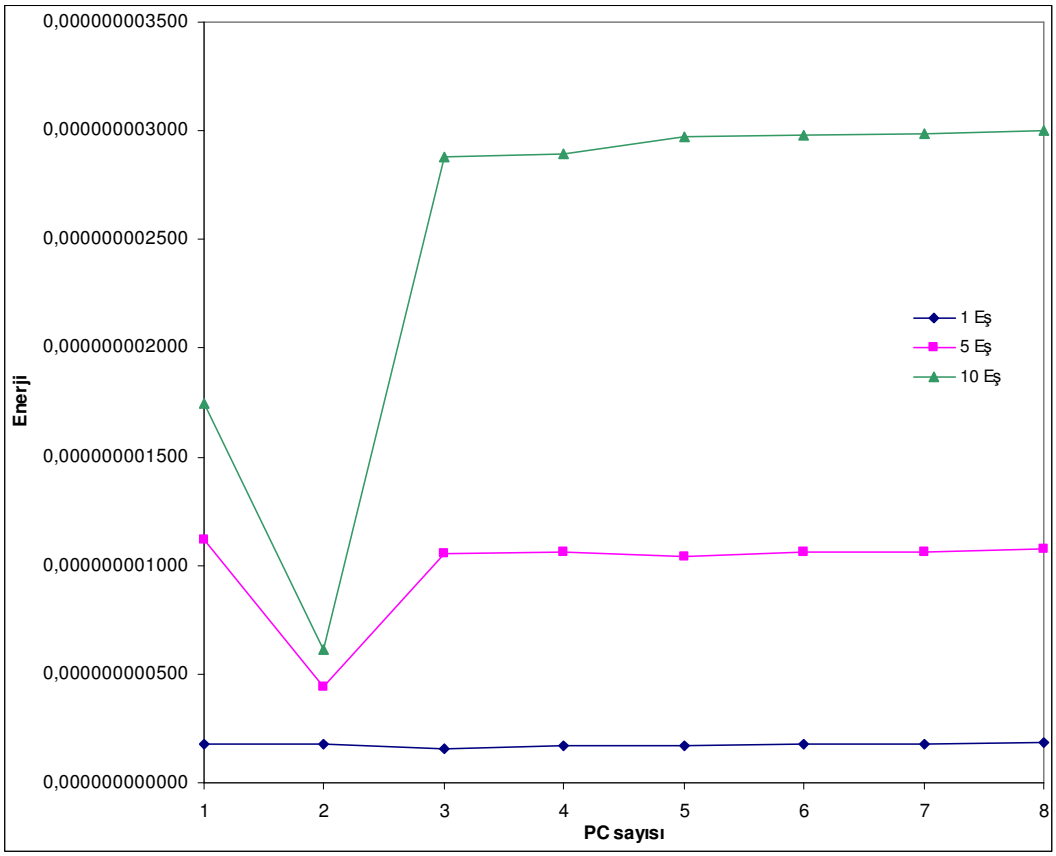
Şekil 6.37'de 1 peer için 5 x $10^{-11}$ civarından başlayan enerji tüketimi 10 peer için 35 x $10^{-10}$ e kadar ulaşmaktadır. Oluşan bu değerlerin gerçekten küçük olması sistemin son derece az miktar enerji tüketimi ile eşdüzey sistemden hizmet sağlayabilmesinin gerçekleştiğini göstermektedir.




**Şekil 6.37** Aynı salondaki bilgisayar sistemleri için "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan enerji tüketimi

Şekil 6.38'de aynı salondaki bilgisayar sistemleri için gsServer türü GigaSpaces servisi kullanılması durumunda oluşan enerji tüketimi görülmektedir. Embededli

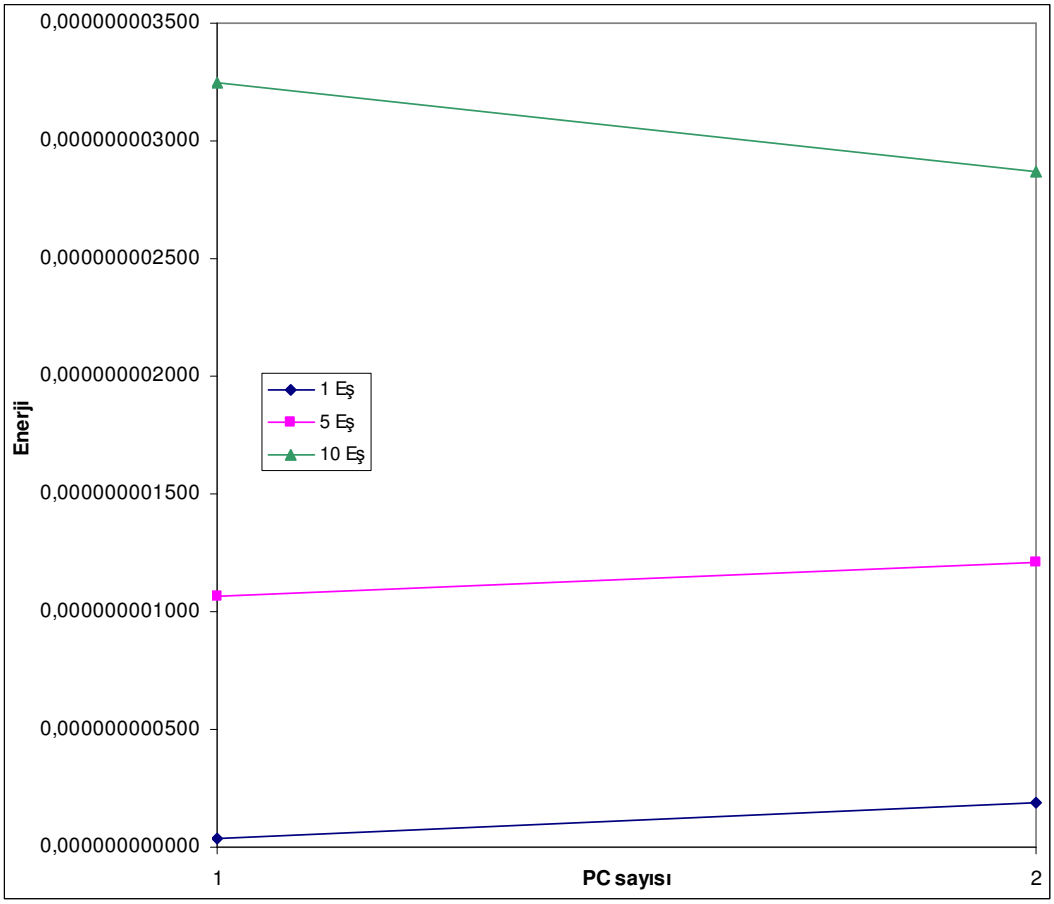
sisteme benzer şekilde son derece düşük enerji tüketimi gereksiniminden dolayı sistemin önemli bir avantajı bulunmaktadır. Üstelik gerek embededli sistem için gerekse gsServer'li sistemde büyük ölçüde stabil bir enerjiye ihtiyaç duyulmaktadır.




**Şekil 6.38** Aynı salondaki bilgisayar sistemleri için "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan enerji tüketimi

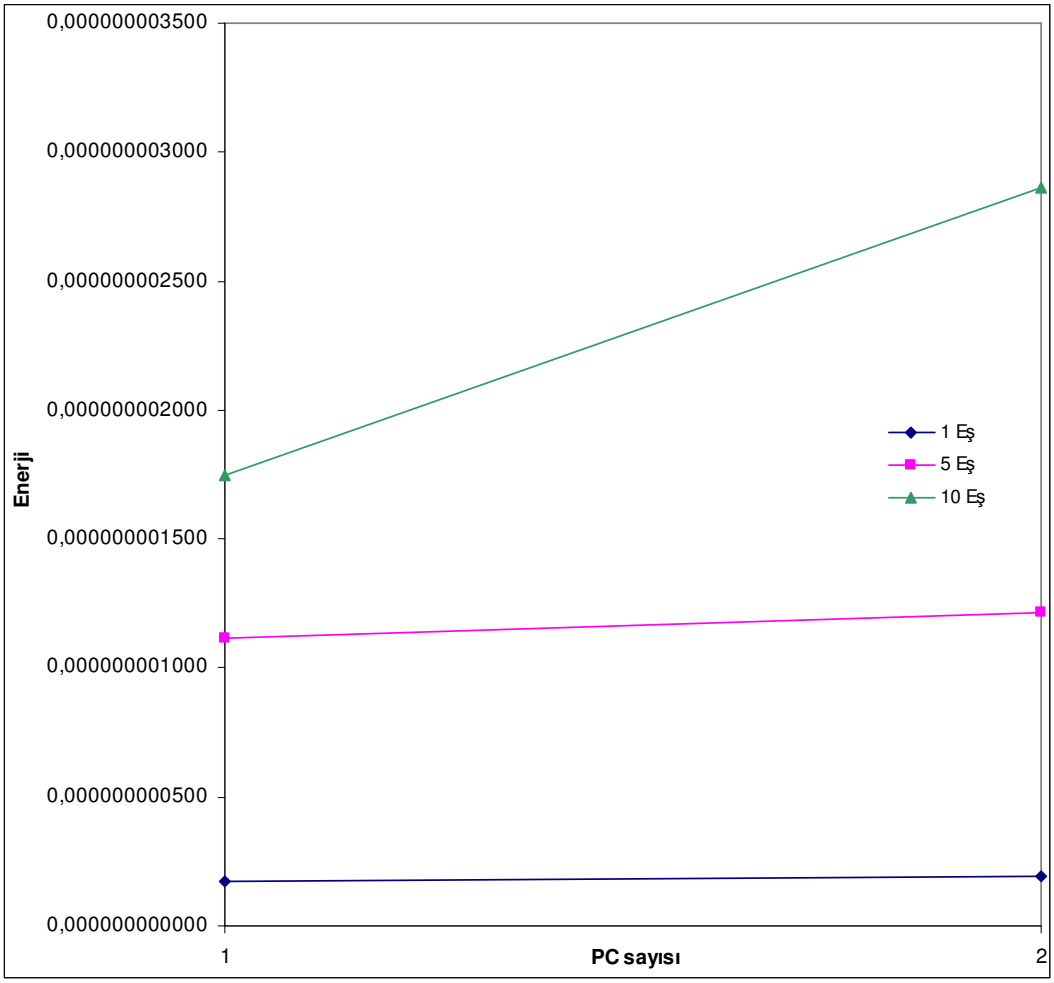
### 6.5.2.2 Uzak Bilgisayar Sistemleri için Enerji İncelemesi

Şekil 6.39'da uzak bilgisayar sistemleri için oluşan enerji tüketimi gösterilmektedir. Aynı salondaki bilgisayar sistemleri için gerçekleştiği şekilde son derece düşük miktarda enerji tüketimi gözlemlenmektedir. Bu bağlamda, $10^{-11}$ ila $10^{-10}$ arasında mertebesinde enerji tüketimi gerekmektedir.




**Şekil 6.39** Uzak bilgisayar sistemleri için "Embeded" türü GigaSpaces servisi kullanılması durumunda oluşan enerji tüketimi

Şekil 6.40'ta uzak bilgisayar sistemleri için gerekli gsServer türü GigaSpaces servisi kullanılması durumunda ne kadar enerji tüketildiği gösterilmektedir. Şekildeki bütün deneylerde çok düşük miktarda enerji gereksinimi ortaya çıkmakla birlikte 10 peer için gerçekleşen artışın diğerlerinden fazla olduğu gözlenmiştir.




**Şekil 6.40** Uzak bilgisayar sistemleri için "gsServer" türü GigaSpaces servisi kullanılması durumunda oluşan enerji tüketimi

Bu bölümde ele alınan bütün veriler göz önünde bulundurulduğu zaman geliştirilen sistemin enerji tüketimi açısından büyük ölçüde başarılı olduğu görülmektedir.

# 7. SONUÇLAR

Bu bölüm dört kısımdan oluşmaktadır. Bunlardan birinci kısmı tez çerçevesinde yapılanların bir özetidir. Bu kısmı takiben ikinci kısımda, yapılan araştırmanın katkısından bahsedilmektedir. Üçüncü kısım olarak geliştirilen sistemin kısıtlarına değinilmektedir. Bu bölümde dördüncü ve son kısım olarak gelecek çalışmalara değinilmiştir.

## 7.1 Yapılanların Özeti

Eş düzey sistemler son yıllarda çok popüler olmuş bilgisayar uygulamalarıdır. Bu popüler uygulamalar ile ilgili çeşitli çalışmalar yapılmaktadır. Ancak, şu ana kadar eş düzey sistemler ile ilgili yapılan araştırmaların büyük bir kısmı kablolu sistemlerde çalışan sistemler için yapılmıştır. Günümüzde ise eğilim hızlı bir şekilde bilgisayar dünyasının mobil ve kablosuz cihazlara kayması şeklinde gerçekleşmektedir.

Bilgisayar dünyasında yeni bir ortam olan mobil ve kablosuz sistemlerin sağlamış olduğu çeşitli avantajlara rağmen, bu sistemlerin normal sistemlere göre birçok kısıtlı özelliği de bulunmaktadır. Örneğin bu sistemlerin kısıtlı işlemci gücü, sınırlı hafıza kapasitesi, küçük ekranı, az ve ani kesintilere uğrayabilen bağlantı özelliği ve sınırlı enerji sorunları bulunmaktadır. Sınırlı özellikleri bulunan mobil ve kablosuz bilgisayar sistemlerinin günümüzün popüler uygulamalarından eş düzey sistemlerden yararlanabilmeleri için çeşitli iyileştirmelere ihtiyaç duyulmaktadır.

Yaptığımız bu çalışma ile eş düzey programların mobil ve kablosuz sistemlerde daha gerçekçi olarak kullanılabilmesi gerçekleştirilmiştir. Bu amaçla öncelikle eş düzey sistemi yeni baştan tasarladık. Tasarladığımız bu yeni eş düzey sistemdeki temel amaç mobil ve/veya kablosuz cihazda çalışmakta olan eş'in yükünü en aza indirmekti. Bu bağlamda eş düzey sistemde işlerin büyük bir kısmını ajanlara (agent) devrettik. Bu amaçla mobil ve/veya kablosuz cihazda çalışan her bir eş için çalışan bir ajan atanmaktadır. Bu ajan sistemde eş adına yapılan her işi takip etmekte, dolayısı ile eş sadece çok gerekli olması durumunda devreye sokulmaktadır. Eşler için çalışmakta olan ajanlar kablolu bilgisayarlarda çalışmakta olup, eş düzey programların makul performans ile çalışması için yeterli sistem kaynaklarına sahiptirler.

Dizayn edilen sistemin en önemli özelliklerinden birisi de ajanlar arası iletişimin gevşek-bağlı (loosely-coupled) model üzerinden gerçekleşmesidir. Bu modelin birçok avantajı bulunmaktadır. Bu bağlamda ajanların birbirini tanıma (kim), aynı anda çalışma (ne zaman) ve de nerede olduklarını bilme zorunlulukları ortadan kalkmaktadır. Bu bağlamda ajanlar birbirlerini ile iletişim kurarlarken hangi ajandan bilgi alınacağı, bu ajanın sistemdeki o anki durumu ve ajanın adresinin bir önemi bulunmamaktadır.

Geliştirilen sistem üzerinde çeşitli testler gerçekleştirilmiştir. Yapılan bu deneyler sonucunda sistemin mobil ve/veya kablosuz bilgisayarlardan da eş düzey programların kullanılmasının olanaklı hale geldiği gözlemlenmiştir. Yapılan testlerden birincisinde eşdüzey sistemdeki mesaj miktarı ölçülmüş, diğerinde ise bu işlemler için harcanan enerji miktarı incelenmiştir.

## 7.2 Araştırmanın Katkısı

Yapılan çalışma sayesinde mobil ve kablosuz cihazlarda eş düzey programların kullanılması daha olanaklı hale gelmesi hedeflenmiştir. Sistemin mobil ve kablosuz sistemlerde eşdüzey uygulamalar için daha olanaklı hale gelmesi için mesaj trafiğinin azaltılması ve harcanan enerjinin azaltılması hedeflenmiştir. Geliştirilen sistem için iki farklı mobil ve kablosuz mimaride çeşitli testler yapılmıştır.

Yapılan testlerde iki konunun kontrolü amacına yönelik çalışmalar uygulanmıştır. Eşdüzey sistemdeki mesaj trafiği ve bu sistemin çalışması sırasında harcanan enerji miktarına odaklanılmıştır. Bu testler sonucunda eş düzey sistemlerin mobil ve kablosuz bilgisayar sistemlerinde daha kolay bir şekilde kullanılması mümkün olduğu görülmüştür. Örneğin bu sistem kullanılması durumunda mobil ve kablosuz bilgisayarların çok daha az mesaj trafiğine maruz kalması söz konusudur. Bu durum ciddi bir işlemci dönüş kazancına ve de hafızanın verimli kullanılmasına söz konusu olmaktadır. Bu bağlamda mobil ve kablosuz cihazların yükü ciddi şekilde hafiflemektedir. Ayrıca, trafiğin azalması ile mobil ve kablosuz cihazlar için en önemli sıkıntılardan olan bu cihazların sınırlı enerji kapasitelerinin verimli kullanılması söz konusu olmaktadır. Yalnızca 1 bitlik bilginin kablosuz olarak gönderilmesi için binlerce hatta bazen milyonlarca aritmetik işlemin yapılmasına eşit enerjinin tüketilmesi

gerektiği gözönüne alındığında, mobil ve kablosuz cihazların enerji tasarrufuna hizmet eden şeylerin değeri daha da iyi anlaşılmaktadır.

Deneyler sonucunda elde edilen verilere göre geliştirilen sistemde mesaj trafiği 1 eşlik sistem için 10.000.000 ns, 5 eş için 60.000.000 ns ve 10 eş için ise 170.000.000 ns şeklinde olduğu görülmektedir. Bu değerler standart bir eşdüzey sistemi kullanılması durumunda trilyon ns mertebesine ulaştığından geliştirilen sistemin yararı ortaya çıkmaktadır.

Enerji ile ilgili yapılan deneyler sonucunda elde edilen sonuçlara göre ise 1 eşli sistemde $1.18 * 10^{-19}$ watt, 5 eşli sistemde $1.1 * 10^{-18}$ watt, $3 * 10^{-18}$ watt olarak değerler elde edilmiştir. Enerji tüketim miktarı da standart bir sistemden çok daha az bir değerde oluşmaktadır.

## 7.3 Kısıtlar

Önerilmiş sistemde bazı kısıtlamalar da bulunmaktadır. Bunlar sırası ile şu şekildedir.

Sistemin temel taşlarından birisi olan gevşek-bağlı modelden kaynaklanan, bilgisayar ağındaki muhtemel bölünmeler sistem için problem yaratabilmektedir. Eğer bir sistemin bağlı olduğu ağda bölünme gerçekleşirse sistemin normal işleyişi sürdürülememektedir. Geliştirilen sistemin yerel ağda kullanılması durumunda, bu gibi problemler nadiren yaşanmaktadır. Sistemin bölünmesi durumu ile karşılaşılması durumunda sistem tam fonksiyonla çalışamamaktadır.

## 7.4 Gelecek Çalışmalar

Yapılacak gelecek çalışmalar öncelikle sistemin kısıtlarını gidermek ve daha iyi performans değerlerine sahip sistem geliştirmek amacıyla yapılması planlanmaktadır. Yeni deneysel çalışmalar ile sistemin daha iyi anlaşılabilirliğini ve uygulanabilirliğinin sağlanması amaçlanmaktadır. Yapılan çalışmaların çeşitli alanlara uygulanması sırasında, geliştirilmiş olan sisteme yeni özellikler ve yetenekler eklenebilir.

Yeni özellikler eklenmesi sayesinde sistem hakkında daha kapsamlı araştırma yapma olanağıda bulunabilecektir. Bu bağlamda, gevşek-bağlı modelin getirdiği avantajlar yanında bu modelin bilgisayar ağlarının bölünmesine karşı etkisizliği çözülmeye çalışılabilir. Bundan başka, sistem yeniden dizayn edilerek bu kez eşler ile ajanların iletişimininde JavaSpaces servisi üzerinden yapılması denenebilir. Ayrıca sistemin daha hızlı çalışması için bazı düzenlemeler yapılabilir.

Geliştirilmiş sistem diğer mobil ve kablosuz sistemlerde test edilebilir. Örneğin sistemin GSM alt yapısını kullanarak ne şekilde bir performans göstereceğini incelenebilir. Benzer bir şekilde geliştirilen sistemin WiMAX alt yapısında nasıl davranacağını incelemek yararlı olacaktır.

Ayrıca bu tezde JavaSpaces servisi hizmeti için kullanılan GigaSpaces servisi yerine Blitz servisi kullanılarakta benzer testlerin yapılması yararlı olabilecektir. Bundan başka, sistemin orjinal JavaSpaces servisi kullanılarak çalıştırılması ve bu sistem ile diğer sistemlerin performans farklarının araştırılması da yararlı olacaktır. Jini teknolojisinin son sürümü özellikle de JavaSpaces servisi hakkında şu anda yetersiz dokümantasyon bulunmaktadır. Bu amaçla SUN Microsystems şirketinin bu teknoloji hakkındaki dokümantasyonu arttırmasında yarar vardır.

## 8. TARTIŞMA

Tuple fikri şu ana kadar birçok projede kullanılmıştır. Bunlardan bir kısmı normal tuple fikri üzerinde yoğunlaşırken, bir kısmı ise orijinal tuple fikrine ek özellikler eklemişlerdir. Genel olarak bakıldığında geliştirilmiş bu sistemler Tuple erişim sistemine ilave esneklik sağlamaktadırlar. Bir genişletilmiş okuma işlemi ile [Jang, M.-W., v.d.] Tuple erişimi geliştirilmiştir. Bu sayede, rasgele bir Tuple okumak yerine, istenen bir grup Tuple'a ulaşmak mümkün olmaktadır. Sistemin esnekliğini arttırmak için bekleme olmayan (asynchronous) Tuple erişimi, kullanılmakta ve bu sistemin performansını arttırabilmektedir. Fakat bu durum gelecekte geri dönecek Tuplelara erişim için iyi bir senkronizasyon sistemi kullanılmasını gerektirmektedir. Önemli diğer bir Tuple Space genişlemesi de, bu sistemlerde mantık operatörü kullanımıdır. "and", "or" gibi operatörler ile birleştirici ve ayırıcı işlemler ile birden fazla Tuple Space'e erişim şekillerine çeşitlilik sağlanmıştır. Örneğin "or" kullanılarak farklı Tuple Space'lerde istenen Tuple'ın bulunmasına olanak vermektedir. Bu durumu sağlayan sistem sağlanan esnekliğe karşılık düşük performans göstermektedir.

MARS [MARS] ve [Cabri, G., v.d.] projesinde Tuple Space'lerin reaktif olması üzerine çalışılmıştır. Bir olay olması durumunda, bir reaksiyon başlar. Reaktivite stimulus'a göre atomik olarak gerçekleşmektedir. JavaSpaces servisinde reaktivite "notify" işlemi ile gerçekleştirilmektedir. Buradaki notifikasyon işlemi asynchronous olmakta, bu yüzden, reaktivite olması durumunda ajan Tuple Space'te istediği Tuple'ı bulamama durumunu yaşayabilirler.

Bu tezde geliştirilmiş olan sistemde kullanılan GigaSpaces servisinde Tuplelar standard özelliklerinin yanında herbir Tuple'a özel bir numara verilmesi, ilk giren Tuple'ın ilk çıkması (FIFO) özelliği gibi normal bir Tuple'dan ek özellikleri bulunmaktadır.

Son yıllarda bilgisayar dünyasındaki önemli araştırma konularından olan ajan programlama (agent based programming) yeni bir programlama yöntemini içermektedir. Bu bağlamda ajan tabanlı programlama çeşitli projeler ile hayata geçirilmiştir. Bu projelere örnek olarak Ambit [Ambit], D'Agents [D'Agents], MARS [MARS] ve [Cabri, G., v.d.], PageSpace [PageSpace], TuCSoN [TuCSoN] ve [TuCSoN, www] çalışmaları örnek verilebilir. Ambit projesinde geliştirilen sistem zaman olarak ajanların

aynı anda çalışmalarını gerektirmezken, ajanların çalışma sırasında birbirini tanımaları gerekmektedir. D'Agents projesini bakıldığında ajanlar arasında iletişim sağlanabilmesi için mutlaka aynı anda çalışması gerekmektedir. Ayrıca, bu sistemdeki ajanlar Ambit projesindeki gibi birbirlerini tanımaları durumunda çalışabilmektedirler. Ajan kullanan diğer bir proje olan MARS, PageSpace ve TuCSoN projelerinde ajanların iletişiminde gerek aynı anda çalışma zorumluluğu gerekse ajanların birbirlerini tanıma zorumluluğu bulunmamaktadır. Bunun sebebi bu projelerin Linda programlama ana fikrini kullanmalarıdır.

Bu tezde geliştirilen sistemde daha önceki bölümlerde bahsedildiği gibi, Linda programlama modeli kullanılmıştır. Bu bağlamda MARS, PageSpace ve TuCSoN projelerine benzemektedir. Diğer taraftan bu tezde geliştirilen sistemde ajanlar birbirleri ile iletişim kurmaları gerekmemektedir. Sistemin çalışmasında kullanılan ve Linda programlama sisteminin uygulanmasını sağlayan JavaSpaces servisi ajanlar tarafından yerel bilgisayarlarda bilginin depolanması amacı ile kullanılmakta, diğer ajanlar bu bilgileri öncelikle yerel JavaSpaces servisinde, eğer bulamazlar ise uzaktaki JavaSpaces servislerinde aramaktadırlar.

Ajan olarak bahsedilen şeyler yalnızca yazılım parçalarından oluşmamakta, aynı zamanda hayatta kendi başına hareket edebilen bilgisayar ve ağ özelliği olan cihazları da kapsamaktadır. Örneğin, bilişim özelliği olan buzdolabı, uçak, gerekli yerler ile iletişim halindeki uydu anten sistemi bunlara örnek bazı sistemler olarak karşımıza çıkmaktadır. Burada bahsedilen sistemlerin robust olması olması gerekmekte ve sistemde ortaya çıkan beklenmeyen sorun durumunda kendiliğinden bu sorunu ortadan kaldırabilmelidir. Bu sistemlerin yapısını en fazla etkileyen faktörlerden biri, bu sistemlerin kendi aralarındaki koordinasyondur. Geliştirilen sistemde ajanların çalışması birbirinden bağımsız olup eğer herhangi bir ajanda problem çıkar ise sistemin akışı etkilenmemektedir.

Çeşitli projeler tuple space-tabanlı ara ajan servislerini optimize etmiştir. Mobile Co-ordination [Rowstron, A.] bu sistemde ajanların bir Tuple Space grubuna erişimde hataya karşı toleransı arttırılmıştır. JADA [PageSpace] projesinde bir işlemin birden çok template'e uyuşması sağlanmıştır. Bu sistemler sayesinde arama özellikleri geliştirilse de, bunların template'lerinin açıklayıcı gücü yeterli derecede gelişmiş değildir (sınırlıdır.)

MARS [MARS], [Cabri, G., v.d.] ve TuCSoN [TuCSoN], [TuCSoN, www] projelerinde Linda türünde Tuple spaces ile ajanlar için programlanabilir koordinasyon mekanizması ile Tuple space'in açıklayıcı gücü arttırılmıştır. Bununla birlikte, bu iki sistem reaktive Tuplelar ile sistemin açıklayıcı gücü arttırılmaktadır. Bir reaktive Tuple belli bir tür Tupleları kullanır ve çeşitli istemciyi etkiler. Diğer taraftan bir arama nesnesi çeşitli tupleları kullanır ve yalızca onu yaratan ajanı etkiler. MARS ve TuCSoN'da reaktive Tuple ile aktive edilen tupleların aynı olması gerekmez. Bir reaktive Tuple ilgili Tuple her insert edildiğinde aktive edilir. Bu insert edilme işlemi kimin gönderdiğine bakılmaksızın yapılmaktadır. Bununla birlikte, bir arama nesne tupleları gönderici ajan için işleme tabii tutar ve diğer ajanlar arama nesnesinden etkilenmezler.

Object Space [Polze, A.] sistemi dağıtık uygulamaların C++ dili kullanılarak template'e uyuşması mantığı ile çalışmaktadır. Buradaki eşleşme şekli eldeki Tuple template ile spaceteki nesneye uyması yöntemi ile olmaktadır. Object Space Tuple space'e bir eşleşme fonksiyonu gönderebilmektedir.

Geliştirilen sistemde GigaSpaces servisinin sağladığı birçok ileri özellikten yararlanılmaktadır. Örneğin, tupleların space'e yazılma sırası FIFO sıralama yöntemi ile hazırlanabilmektedir. Ayrıca, sistemde cluster yöntemi kullanılabileceği gibi sistemde replication yöntemi kullanılabilmektedir. Ayrıca GigaSpaces servisinin sağladığı toplu komut işlemlerinden de yararlanılabilmektedir.

MAP2P projesi geliştirdiğimiz projedeki sisteme benzemektedir. Örneğin, bir mobil cihaz eşdüzey sisteme sabit bilgisayarlarda bulunan ajanların yardımı ile erişebilmektedir. Diğer taraftan bu tezde geliştirilen sistemde GigaSpaces servisi özellikleri kullanılarak gevşek-bağlı model sisteminin sağladığı avantajlardan yararlanılmaktadır. Bu bağlamda ajanların aynı anda çalışmaları gerekmemektedir. Ayrıca, ajanların iletişimi sırasında birbirlerini tanımaları da gerekli değildir. Sistemimizin tasarım şekli sayesinde paylaşılmasına karar verilen bilgiler sisteme bir kez yüklendikten sonra paylaşımının mobil veya kablosuz cihazdaki peer tarafından kaldırılması talep edilinceye kadar sınırsız olarak, bütün eşdüzey kullanıcılara açık bir erişim ortamı yaratılmıştır.

MAP2P sistemine göre, geliştirilmiş olan sistemin mesaj trafiği azalmaktadır. Örneğin paylaşılmak istenen bilgi için yalnızca bir defa bağlı bulunduğu ajana bu bilgiyi

iletmesi yeterlidir. Diğer taraftan bilgi alımı sırasında da yalnızca sonuç ajan tarafından mobil veya kablosuz cihazda çalışmakta olan peera iletilmektedir.

**KAYNAKLAR**


[Aberer, K., v.d.] Aberer, K. et. al. "The Essence of P2P: A Reference Architecture for Overlay Networks," *Fifth IEEE International Conference on Peer-to-Peer Computing, (P2P'05)*, 2005, pp.11-20.

[Acharya, A., v.d.] Acharya, A., Misra, A., and Bansal, S., "High-Performance Architectures for IP-based Multihop 802.11 Networks," *IEEE Wireless Communications,* Vol.10, 2003, 5, pp.22–28.

[Adacal, M., v.d.] Adacal, M. and Bener, A. B., "Mobile Web Services: A New Agent-Based Framework," *IEEE Internet Computing*, Vol.10, No.3, 2006, pp.58-65.

[Adwankar, S., v.d.] Adwankar, S., Vasudevan, V., "Mobile Agent Based Pervasive Systems Manager for Enterprise Network," *ACM Symposium on Applied Computing 2003 (SAC 2003)*, Melbourne Florida, 9-12.March.2003, pp.886-892.

[Akyildiz, I.F.,K.,I.H., v.d.] Akyildiz, I.F., and Kasimoglu, I.H., "Wireless Sensor and Actor Networks: Research Challenges," *Ad Hoc Networks* 2, 2004, pp.351–367.

[Akyildiz,I.F.,L,W.-L.,v.d.] Akyildiz, I.F., Lee, W.-Y., Vuran, M.C., and Mohanty, S., "NeXt Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: A Survey," *Computer Networks*, Vol.50, No.13, September 2006, pp.2127-2159.

[Akyildiz,I.F.,W.,X.,v.d.] Akyildiz, I.F., Wang, X., and Wang, W., "Wireless Mesh Networks: A Survey," *Computer Networks*, Vol.47, No.4, March 2005, pp.445-487.

[Akyildiz, I.F.,Xie,J.,v.d.] Akyildiz, I.F., Xie, J., and Mohanty, S., "A Survey of Mobility Management in Next-Generation All-IP-Based Wireless Systems," *IEEE Wireless Communications* 11, 2004, 4, pp.16–28.

[Akyildiz, I.F., .....] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., and Cayirci, E., "Wireless Sensor Networks: A Survey," *Computer Networks*, Vol.38, 2002, pp.393-422.

[Ambit] Ambit, Mobile Computational Ambients ana sayfası, http://lucacardelli.name/Ambit/Ambit.html

[Arnold,K.,Gao,G.R.,v.d.] Arnold, K., Gao, G.R., Ghosh, S., *Java/Jini Technologies and High-Performance Pervasive Computing*, Society of Photo-Optical Instrumentation Engineers (SPIE), June 2002.

[Arnold, K., v.d.] Arnold, K., Gosling, J., Holmes, D., *Java Programming Language, 4th ed.*, Prentice Hall, Ağustos 2005.

[Atlig, C., v.d.] Atlig, C. and Kostin, A., "A Distributed Resource Replication System Based on JavaSpaces," *Sixteenth International Symposium on Computer and Information Sciences (ISCIS XVI)*, Antalya, November 5-7, 2001, pp.72-79.

[Aziz, M.K.A., v.d.] Aziz, M.K.A., Nix, A.R., Fletcher, P.N., "A Study of Performance and Complexity for IEEE 802.11n MIMO-OFDM GIS Solutions," *IEEE International Conference on Communications*, Vol.7, pp.3822-3826, June 2004.

[Basagni, S., v.d.] Basagni, S., Conti, M., Giordano, S., and Stojmenovic, I., *Mobile AD HOC Networking*, IEEE Press, 2004.

[Belding-Royer, E.M.] Belding-Royer, E.M., "Multi-level Hierarchies for Scalable Ad Hoc Routing," *ACM/Kluwer Wireless Networks* 9, 2003, 5, pp.461–478.

[Belding-Royer, E.M., v.d.] Belding-Royer, E.M., Melliar-Smith, P.M., and Moser, L.E., "An Analysis of the Optimum Node Density for Ad Hoc Mobile Networks," *IEEE International Conference on Communications (ICC)*, Vol.3, June 2001, pp.857–861.

[Bellavista, P., v.d.] Bellavista, P., (ed.), Corradi, A., (ed.), *The Handbook of Mobile Middleware*, Auerbach, October 2006.

[Blitz] Blitz ana sayfası, http://www.dancres.org/blitz/

[Bottigliengo, M., v.d.] Bottigliengo, M., Casetti C., Chiasserini, C.-F., and Meo, M., "Enhancing Fairness for Short-Lived TCP Flows in 802.11b WLANs," *IEEE Transactions on Vehicular Technology*, Vol.56, No.1, January 2007, pp.206-217.

[Boukerche, A.] Boukerche, A., (ed.), *Handbook of Algorithms for Wireless Networking and Mobile Computing*, Chapman & Hall/CRC Taylor & Francis Group, 2006.

[Braun, P., v.d.] Braun, P., Rossak, W., *Mobile Agents–Basic Concept, Mobility Models, and the Tracy Toolkit*, Morgan Kaufmann, 2005.

[Buttyan, L., v.d.] Buttyan, L., and Hubaux, J.-P., "Report on a Working Session on Security in Wireless Ad Hoc Networks," *ACM Mobile Computing and Communications Review* **7,** 2002, 1, pp.74–94.

[Cabri, G., v.d.] Cabri, G., Leonardi, L., and Zambonelli, F., "MARS: A Programmable Coordination Architecture for Mobile Agents," *IEEE Internet Computing*, July-August, 2000, pp.26-35.

[Cabri, G., v.d. 2] Cabri, G., Leonardi, L., and Zambonelli, F., "Mobile-Agent Coordination Models for Internet Applications," *IEEE Computer*, February, 2000, pp.82-89.

[Carriero, N., v.d.] Carriero, N., and Gelernter, D., "A Computational Model of Everything," *Communications of the ACM*, Vol.44, No.11, November 2001, pp.77-81.

[Chen, L., v.d.] Chen, L., and Skelton, G.W., *Mobile Commerce Application Development*, CyberTech, 2005.

[Chen, W.-T., v.d.] Chen, W.-T., Chen, P.-Y., "Group Mobility Management in Wireless Ad Hoc Networks," *IEEE Vehicular Technology Conference*, Fall 2003, pp.2202–2206.

[Chlamtac, I., v.d.] Chlamtac, I., Conti, M., and Liu, J., "Mobile Ad Hoc Networking: Imperatives and Challenges," *Ad Hoc Networks* 1**,** 2003, 1, pp.13–64.

[Chow, C.-Y., v.d.] Chow, C.-Y., Leong, H.V., and Chan, A.T.S., "GroCoca: Group-based Peer-to-Peer Cooperative Caching in Mobile Environment," *IEEE Journal on Selected Areas in Communications*, Vol.25, No.1, January 2007, pp.179-191.

[Chtcherbina, E., v.d.] Chtcherbina, E., Wieland, T., In the Beginning was the Peer-to-Peer, 2002. http://www.old.netobjectdays.org/pdf/02/papers/industry/1411.pdf

[Coulouris, G., v.d.] Coulouris, G., Dollimore, J., Kindberg, T., *Distributed Systems Concepts and Design, 4th ed.*, Addison-Wesley, June 2005.

[D'Agents] D'Agents ana sayfası, http://agent.cs.dartmouth.edu/

[DECT Forum] DECT Forum ana sayfası http://www.dect.org/

[Demirtas, T.] Demirtas, T., *Scheduling and Prototyping of Distributed Real-Time Systems (An Approach Using Jini/JavaSpaces)*, Storming Media, 2002.

[Dickman, J., v.d.] Dickman, J., Rath, K., Kotecha, L., "Proposal for 802.16 Connection Oriented Mesh," IEEE 802.16 Standard Proposal, March 2003. http://www.ieee802.org/16/tgd/contrib/S80216d-03_18.pdf

[Dixit, S., v.d.] Dixit, S. (ed.), and Wu, T., *Content Networking in the Mobile Internet*, John Wiley & Sons, 2004.

[Doudou, A., v.d.] Doudou, A., Garbinato, B., and Guerraoui, R., Tolerating Arbitrary Failures With State Machine Replication, *Dependable Computing Systems, Chapter 2*, Diab, H., and Zomaya, A. Y., (ed.), pp.27-56.

[Dutta, A., v.d.] Dutta, A., Wong, K.D., Burns, J., Jain, R., McAuley, A., Young, K., Schulzrinne, H., "Realization of Integrated Mobility Management Protocol for Ad Hoc Networks," *IEEE Military Communication Conference (MILCOM)*, 2002, pp.448–454.

[Edney, J., v.d.] Edney, J., and Arbaugh, W.A., *Real 802.11 Security: Wi-Fi Protected Access and 802.11i*, Addison-Wesley, July 2003.

[Edwards, W.K.] Edwards, W.K., *Core Jini, 2nd ed.*, Prentice-Hall, December 2000.

[Elektro Dalga ve Sağlık] Elektromanyetik Dalgalar ve İnsan Sağlığı Sıkça Sorulan Sorular ve Yanıtları, Tübitak-Bilten 2001.

[Eugster, P.Th., v.d.] Eugster, P.Th., Garbinato, B., and Holzer, A., "Location-based Publish/Subscribe," UNIL Technical Report, DOP-20050124, 17 pages.

[Flanagan, D.] Flanagan, D., *Java in a Nutshell, 5th ed.*, O'Reilly, March 2005.

[FCC Cognitive Radios] FCC Cognitive Radios, http://www.fcc.gov/oet/cognitiveradio

[FIPA] FIPA, Akıllı Fiziksel Ajanlar Kurumu ana sayfası, http://www.fipa.org/

[Flenner, R., v.d.] Flenner, R., Abbott, M., Boubez, T., Cohen, F., Krishnan, N., Moffet, A., Ramamurti, R., Siddiqui, B., Sommers, F.,

*Java P2P Unleashed, with JXTA, Web Services, XML, Jini, JavaSpaces, and J2EE*, Sams Publishing, August 2002.

[Flocchini, P., v.d.] Flocchini, P., Nayak, A., and Xie, M., "Enhancing Peer-to-Peer Systems Through Redundancy," *IEEE Journal on Selected Areas in Communications*, Vol.25, No.1, January 2007, pp.15-24.

[Freeman, E., v.d.] Freeman, E., Hupfer, S., and Arnold, K., *JavaSpaces Principles, Patterns and Practice*, Addison-Wesley, 1999.

[Ganguly, S., v.d.] Ganguly, S., Saxena, A., Bhatnagar, S., Izmailov, R., Banerjee, S., "Fast Replication in Content Distribution Overlays," *24th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2005*, 13-17 March 2005, Vol.4, pp.2246-2256.

[Ganjali, Y., v.d.] Ganjali, Y., and Keshavarzian, A., "Load Balancing in Ad Hoc Networks: Single-path Routing vs. Multi-path Routing," *IEEE Annual Conference on Computer Communications (INFOCOM)*, March 2004, pp.1120–1125.

[Gao, L., v.d.] Gao, L., Dahlin, M., Zheng J., Alvisi, L., and Iyengar, A., "Dual-Quorum Replication for Edge Services," *ACM/IFIP/USENIX 6th International Middleware Conference, Middleware 2005*, Grenoble, France, November 28-December 2 2005, pp.184-204.

[Gao, Y., v.d.] Gao, Y., Chiu, D.M., and Lui, J.C.S., "Determining the End-to-End Throughput Capacity in Multi-Hop Networks: Methodology and Applications," *Proceeding of the SIGMETRICS/Performance 2006*, Saint Malo, France, 26-30.June.2006, pp.39-50.

[Garg, V.K.] Garg, V.K., *Concurrent and Distributed Computing in Java*, IEEE Press, John Wiley & Sons, 2004.

[Gelernter, D.] Gelernter, D., "Generative Communication in Linda," *ACM Transactions on Programming Languages and Systems*, Vol.7, No.1, pp.80-112.

[Gelernter, D., v.d.] Gelernter, D., and BernStein, A., "Distributed Communications via Global Buffer," *ACM SIGACT-SIGOPS Symp. On Principles of Distributed Computing*, Ottawa, Canada, August 1982, pp.10-18.

[Ghosemajumder, S.] Ghosemajumder, S., "Advanced Peer-Based Technology Business Models," *MIT, Sloan School of Management*, 2002.

[GigaSpaces] GigaSpaces ana sayfası, http://www.gigaspaces.com/

[GigaSpaces Guide] GigaSpaces Programmer's Guide and Reference, for version 5.2, January 2007.

[Gnutella] Gnutella ağı ana sayfası, http://www.gnutella.com/

[Gosling, J., v.d.] Gosling, J., Joy, B., Steele, G., Bracha, G., *The Java Language Specification, 3rd ed.*, Addison-Wesley, 2005.

[GSM Association] GSM Association, ana sayfası, http://www.gsmworld.com

[Guizani, M.] Guizani, M. (ed.), *Wireless Communications Systems and Networks*, Kluwer Academic Publishers, 2004.

[Han, S.C., v.d.] Han, S.C., and Xia, Y., "Constructing an Optimal Server Set in Structured Peer-to-Peer Networks," *IEEE Journal on Selected Areas in Communications*, Vol.25, No.1, January 2007, pp.170-178.

[Hariri, S., v.d.] Hariri, S. (ed.), and Parashar, M. (ed.), *Tools and Environments for Parallel and Distributed Computing*, John Wiley & Sons, 2004.

[Harold, E., v.d.] Harold, E., and Harold, E.R., *Java Network Programming, 3rd ed.*, O'Reilly, October 2004.

[Hassan, M., v.d.] Hassan, M., and Jain, R., *High Performance TCP/IP Networking, Concepts, Issues, and Solutions*, Prentice Hall, October 2003.

[Holma, H., v.d.] Holma, H. (ed.), and Toksala, A. (ed.), *WCDMA for UMTS, Radio Access for Third Generation Mobile Communications, 3rd ed.*, John Wiley & Sons, 2004.

[Hu, T.H-t., v.d.] Hu, T.H-t., Thai, B., Lilakiatsakun, W., and Seneviratne, A., "MAP2P: An Architecture Supporting Gnutella for Mobile Devices," *IEICE Transactions Information and Systems*, Vol.E88-D, No.9, 2005, pp.2079-2088.

[Huang, C.-M., v.d.] Huang, C.-M., Hsu, T.-H., and Hsu M.-F., "Network-Aware P2P File Sharing over the Wireless Mobile Networks," *IEEE Journal on Selected Areas in Communications*, Vol.25, No.1, January 2007, pp.204-210.

[IBM Aglets] IBM Aglets ana sayfası, http://www.trl.ibm.com/aglets/

[IBM TSpaces 1998] IBM. TSpaces projesi ana sayfası, http://www.almaden.ibm.com/cs/tspaces/

[IEEE 802.11] IEEE 802.11 Standart Grup Sitesi, http://grouper.ieee.org/groups/802/11/

[IEEE 802.15] IEEE 802.15 Standart Grup Sitesi, http://grouper.ieee.org/groups/802/15/

[IEEE 802.16] IEEE 802.16 Standart Grup Sitesi, http://grouper.ieee.org/groups/802/16/

[Ilyas, M., v.d.] Ilyas, M., (ed.), Mahgoub, I., (ed.), *Mobile Computing Handbook*, CRC, June 2004.

[Intel Heterogeneous Sens.] Intel Exploratory Research, Heterogenous Sensor Networks, http://www.intel.com/research/exploratory/heterogeneous.htm

[Intel Mesh] Intel Inc., Multi-Hop Mesh Networks-Yeni bir tür WiFi network.

[JADE] JADE white paper, http://jade.tilab.com/papers/2003/WhitePaperJADEEXP.pdf

[Jain, K., v.d.] Jain, K., Padhye, J., Padmanabhan, V., and Qiu, L., "Impact of Interference on Multi-Hop Wireless Network Performance," *ACM International Conference on Mobile Computing and Networking (MOBICOM)*, September 2003, pp.66-80.

[JAJAH] JAJAH, mobil ve sabit sistemler için VoIP hizmeti veren program ana sayfası, http://www.jajah.com/

[Jang, M.-W., v.d.] Jang, M.-W., Momen, A.A., Agha, G., "ATSpace: A Middle Agent to Support Application Oriented Matchmaking And Brokering," *IEEE/WIC/ACM IAT (Intelligent Agent Technology)*, 2004.

[Kant, K., v.d.] Kant, K., Iyer, R., Tewari, V., "A Framework for Classifying Peer-to-Peer Technologies," *2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid'02)*, May 22-24, 2002, pp.338-345.

[Katsaros, D., v.d.] Katsaros, D., Nanopoulos, A., and Manolopoulos, Y., *Wireless Information Highways*, IRM Press, January 2005.

[Khosrow, M.] Khosrow, M. (ed.), *Cases on Telecommunications and Networking, Idea Group*, 2006.

[Kiyon Autonoumous Nets.] Kiyon Autonoumous Networks, http://www.kiyon.com

[Koivukoski, U., v.d.] Koivukoski, U., and Raisanen, V., *Managing Mobile Services, Technologies and Business Practices*, John Wiley & Sons, 2005.

[Kou, W., v.d.] Kou, W., (ed.), Yesha, Y., (ed.), *Enabling Technologies for Wireless E-Business*, Springer, May 2006.

[Kozat, U.C., v.d.] Kozat, U.C., and Tassiulas, L., "Throughput Capacity of Random Ad Hoc Networks with Infrastructure Support," *ACM International Conference on Mobile Computing and Networking (MOBICOM)*, San Diego, CA, 2003, pp.55-65.

[Kumar, R., v.d.] Kumar, R., Yao, D.D., Bagchi, A., Ross, K.W., and Rubenstein, D., "Fluid Modeling of Pollution Proliferation in P2P Networks," *SIGMETRICS/Performance 2006*, Saint Malo, France, 26-30.June.2006, pp.335-346.

[Lane, B.] Lane, B., Cognitive Radio Technologies in the Commercial Arena, in FCC Workshop on Cognitive Radios, May 2003.

[Larsen, J.E., v.d.] Larsen, J.E. and Spring, J.H., *GLOBE, A Dynamic Fault-Tolerant and Dynamically Scalable Distributed Tuplespace for Heterogeneous, Loosely Coupled Networks*. Master Thesis, Copenhagen University, 1999.

[Lee, J.S., v.d.] Lee, J.S., and Hsu, P.L., "Implementation of a Remote Hierarchical Supervision System Using Petri Nets and Agent Technology," *IEEE Transactions on Systems, Man. and Cybernetics-Part C: Applications and Reviews*, Vol.37, No.1, January 2007, pp.77-85.

[Liu, B., v.d.] Liu, B., Liu, Z., and Towsley, D., "On the capacity of hybrid wireless networks," *IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, CA, 2003, pp.1543–1552.

[Li, J., v.d.] Li, J., Blake, C., De Couto, D.S.J., Lee, H.I., and Morris, R., "Capacity of Ad Hoc Wireless Networks," *ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 2001, pp.61–69.

[Maltz, D.A., v.d.] D.A. Maltz, J. Broch and D.B. Johnson, "Lessons from a Full-Scale Multihop Wireless Ad Hoc Network Testbed," *IEEE Personal Communications*, 8, 2001, pp.8–15.

[MARS] Moon projesi, MARS ana sayfası, http://polaris.ing.unimo.it/MOON/MARS/

[Marshall, P] Marshall, P, "Beyond the Outer Limits: Next Generation Communications, FCC Workshop on Cognitive Radios, May 2003.

[MeshNetworking Inc.] MeshNetworks Inc. Scalable Routing Technology, http://www.meshnetworks.com/pages/technology/msr_atp_overview.htm

[Microsoft Mesh] Microsoft Mesh Networks, http://research.microsoft.com/mesh/

[Mock, M., v.d.] Mock, M., Couturier, S., "Middleware-Integration of Small Devices," *10th IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2005)*, 2005, pp.807-814.

[Modiano, E., v.d.] Modiano, E., Shah, D. and Zussman, G., "Maximizing Throughput in Wireless Networks via Gossiping," *SIGMETRICS/Performance 2006*, Saint Malo, France, 26-30.June.2006, pp.27-38.

[Molisch, A.] Molisch, A., *Wireless Communications*, IEEE Press, John Wiley & Sons, 2005.

[Moulder, J.E.] Mobile Phone (Cell Phone) Base Stations and Human Health, http://www.mcw.edu/gcrc/cop/cell-phone-health-FAQ/toc.html

[Napster] Napster, http://en.wikipedia.org/wiki/Napster

[Near Field Com.] Near Field Communication white paper, http://www.ecma-international.org/activities/Communications/2004tg19-001.pdf

[Newmarch, J.] Newmarch, J., *Foundations of Jini 2 Programming*, Apress, October 2006.

[Nokia HSDPA] Nokia HSDPA Solution white paper, http://www.nokia.com/NOKIA_COM_1/Operators/Technologies/Network_Enhancements/HSDPA/HSDPA_A4.pdf

[Nortel Mesh Net.] Nortel Mesh Networks, http://www.nortelnetworks.com/solutions/wrlsmesh/index.html

[PageSpace] PageSpace projesi ana sayfası, http://flp.cs.tu-berlin.de/pagespc/

[Peters, J., v.d.] Peters, J., Roth, V., "A Scalable and Secure Global Tracking Service for Mobile Agents," *Mobile Agents 2001, LNCS*, Springer-Verlag, 2001.

[Peterson, L.L., v.d.] Peterson, L.L., Davie, B.S., *Computer Networks: A System Approach, 3rd ed.*, Morgan Kaufmann, 2003.

[Polze, A.] Polze, A., "Using the Object Space: a Distributed Parallel Make," *4th IEEE Workshop on Future Trends of Distributed Computing Systems*, Lisbon, September 1993, pp.234-239.

[Pinsdorf, P.,] Pinsdorf, P., "A Formal Approach for Interoperability between Mobile Agent Systems and Component Based Architectures," *11th IEEE International Conference on the Engineering of Computer-Based Systems*, 2004.

[Pullum, L.] Pullum, L., *Software Fault Tolerance Techniques and Implementation*, Artech House, October 2001.

[Ramjee, P.] Ramjee, P., *OFDM for Wireless Communications Systems*, Artech House, 2004.

[Roth, V.] Roth, V., "Empowering Mobile Software Agents," *6th IEEE Mobile Agents Conference, LNCS*, Spinger-Verlag, 2002.

[Rowstron, A.] Rowstron, A., "Mobile Co-ordination: Providing Fault Tolerance in Tuple Space Based Co-ordination Languages," *3rd International Conference on Coordination Languages and Models*, 1999, pp.196-210.

[Saha, A.K., v.d.] Saha, A.K., and Johnson, D.B., "Self-Organizing Hierarchical Routing for Scalable Ad Hoc Networking, Technical Report, TR04-433, Department of Computer Science, Rice University.

[Sarikaya, B., v.d.] Sarikaya, B., "Home Agent Placement and IP Address Management for Integrating WLANs with Cellular Networks," *IEEE Wireless Communications*, December 2006, pp.77-86.

[SlackSpaces] SlackSpaces ana sayfası, http://slackspaces.geir.tv/ (Tezin hazırlandığı sırada web sitesi bakımdaydı.)

[SOMA] SOMA projesi ana sayfası, http://www-lia.deis.unibo.it/Software/MA/

[SPEC] Standard Performance Evaluation Corporation (SPEC), http://www.spec.org/

[Steinmetz, R., v.d.] Steinmetz, R., (ed.), Wehrle, K., (ed.), *Peer-to-Peer Systems and Applications*, LNCS 3485, Springer, October 2005.

[Subramanian, R., v.d.] Subramanian, R. (ed.), Goodman, B.D. (ed.), *Peer-to-Peer Computing: The Evolution of a Disruptive Technology*, Idea Group, February 2005.

[SUN Jini] SUN. Jini Network Technology, http://www.sun.com/software/jini/index.xml

[Symmy] Symmy, mobil ve sabit sistemler için VoIP hizmeti veren program ana sayfası, http://www.symmy.com/

[Tanenbaum, A.] Tanenbaum, A., *Computer Networks, 4th ed.*, Prentice Hall, 2003.

[Taylor, I. J.] Taylor, I. J., *From P2P to Web Services and Grids: Peers in a Client/Server World*, Springer, October 2004.

[Thuraya] Thuraya firması ana sayfası, http://www.thuraya.com/

[Tropos Networks] Tropos Networks, http://www.tropos.com/technology/whitepaper.shtml

[Tse, D.N.C., v.d.] Tse, D.N.C., and Grossglauser, M., "Mobility Increases the Capacity of Ad Hoc Wireless Networks," *IEEE/ACM Transactions on Networking*, 10**,** 2002, 4, pp.477–486.

[Tseng, Y.-C., v.d.] Tseng, Y.-C., Hsu, C.-S., and Hsieh, T.-Y., "Power-Saving Protocols for IEEE 802.11 based Multi-Hop Ad Hoc Networks," *IEEE Conference on Computer Communications (INFOCOM)*, 2002, pp.200–209.

[TuCSoN] TuCSoN Guide, TuCSoN version 1.4.5, apiCE Lab, Universita di Bologna, Cesena, Italy, and DEIS, Universita di Bologna, Italy, 04-11-2006.

[TuCSoN, www] TuCSoN projesi ana sayfası,

http://www.alice.unibo.it:16080/projects/tucson/

[UCSD Mesh Net.] UCSD mesh networks testbed, http://www.calit2.net/

[Ultra Wideband] Ultra Wideband Forum, http://www.uwbforum.org/

[Vassev, E., v.d.] Vassev, E. and Paquet, J., "A General Architecture for Demand Migration in a Demand-Driven Execution Engine in a Heterogeneous and Distributed Environment," *3rd Annual Communication Networks and Services Research Conference (CNSR'05)*, May 16-18, 2005, pp.176-182.

[Verma, D.C.] Verma, D.C., *Legitimate Applications of Peer-to-Peer Networks*, John Wiley & Sons, 2004.

[Wang, S.C., v.d.] Wang, S.-C., Chou, H.-Z., Wei, D.S.-L., and Kuo, S.-Y., "On the Fundamental Performance Limits of Peer-to-Peer Data Replication in Wireless Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, Vol.25, No.1, January 2007, pp.211-221.

[WiBro Org] WiBro organizasyonu ana sayfası, http://www.wibro.or.kr/

[Wi-Fi Alliance] Wi-Fi Alliance, http://www.wi-fi.org

[WiMAX Forum] WiMAX Forum, http://www.wimaxforum.org

[WiMedia Alliance] WiMedia Alliance, http://www.wimedia.org/

[Wyckoff, P., v.d.] Wyckoff, P., McLaughry, S.W., Lehman, T.J., and, Ford, D.A., "TSpaces," IBM Systems Journal, Vol.37, No.3, August 1998.

[Yoo, Y., v.d.] Yoo, Y., and Agrawal, D., "Why Does It Pay to be Selfish in a MANET," *IEEE Wireless Communications*, December 2006, pp.87-97.

[Zhang, F., v.d.] Zhang, F., Santos, E., and Luh, P.B., "A Performance Study on a Multiagent E-Scheduling and Coordination Framework for Maintenance Networks," *IEEE Transactions on Systems, Man. and Cybernetics-Part C: Applications and Reviews*, Vol.37, No.1, January 2007, pp.52-65.

[ZigBee Alliance] ZigBee Alliance, www.zigbee.org/

[Zoels, S., vd. 1] Zoels, S., Schubert, S., Kellerer, W., Despotovic, Z, "Hybrid DHT Design for Mobile Environments," *AP2P 2006 Workshop at AAMAS 2006*, Hakodate, Japan, May 09-12, 2006.

[Zoels, S., vd. 2] Zoels, S., Schollmeier, R., Kellerer, W., and Tarlano, A., "The Hybrid Chord Protocol: A Peer-to-Peer Lookup Service for Context-Aware Mobile Applications," *ICN 2005*, pp.781-792.

# ÖZGEÇMİŞ

Cenk Atlığ, Edirne'li bir ailenin ilk çocuğu olarak, 1976 yılında dünyaya gelmiştir. İlk, orta ve lise eğitimini 1993 yılı itibariyle Edirne'de tamamlamıştır.

Aynı yıl Doğu Akdeniz Üniversitesi (DAÜ) Bilgisayar Mühendisliği bölümünü kazanmıştır. İlk yıl İngilizce hazırlık okulunda okumuştur. Burslu olarak okuduğu bölümünde, 1998 yılında Bilgisayar Mühendisliği lisans programını bitirerek, Bilgisayar Mühendisi ünvanını kazanmıştır. Mezuniyetinin hemen arkasından, DAÜ Bilgisayar Mühendisliği bölümüne Araştırma Görevlisi olarak atanmıştır. 2000 yılında aynı bölümün yüksek lisans programından, Bilgisayar Yüksek Mühendisi olarak mezun olmuştur.

Yine 2000 yılında, merkezi ABD'de olan, Elektrik Elektronik Mühendisleri Enstitüsü (IEEE) Bilgisayar Derneği (Computer Society) tarafından, gösterdiği akademik başarı ve yürüttüğü sosyal faaliyetler ile dünyada gelecek vaad eden ve dünya çapında yalnızca dört kişiye verilen, Richard E. Merwin bursunu kazanmıştır. Atlığ, şu ana kadar bu bursu kazanma başarısı gösteren dünyadaki tek Türk olma özelliğini de taşımaktadır.

Atlığ, merkezi ABD'nin New York şehrinde olan, Bilgisayar Makineleri Kuruluşu (ACM) tarafından, ALTIN (GOLD) öğrenci olarak 2001 yılının Ekim ayında ABD'nin Florida eyaletinin, Tampa şehrinde yapılan, Nesne Tabanlı Programlama, Yazılım Dilleri, ve Uygulamaları (OOPSLA) bilgisayar konferansının organizasyon komitesi davetli üyesi olarak yer almıştır. Bir yıl sonra yine ACM tarafından, 2002 yılının Kasım ayında ABD'nin Washington eyaletinin, Seattle şehrinde düzenlenen, OOPSLA bilgisayar konferansına ALTIN (GOLD) öğrenci, davetli konferans organizyon komitesi üyesi olarak katılmıştır.

2002-2003 öğretim yılına kadar, DAÜ Bilgisayar Mühendisliği bölümünde Araştırma Görevlisi olarak görev yapan Atlığ, 2003-2004 öğretim yılı itibarı ile Trakya Üniversitesinde çalışmaya başlamıştır.

2006 yılında Fransa'nın Nantes şehrinde yine ACM tarafından düzenlenen, Avrupa Nesneye-Yönelik Programlama Konferansına (ECOOP), GÜMÜŞ (SILVER) öğrenci, davetli konferans organizyon komitesi üyesi olarak katılmıştır. Ayrıca, aynı yıl yine Fransa'nın Saint-Malo şehrinde düzenlenen SIGMETRICS/PERFORMANCE 2006 konferansına, konferans organizasyonunun verdiği seyahat bursu ile katılmıştır.

Atlığ 2004-2007 yılları arasında IEEE Türkiye Şubesinde GOLD kolu başkanı olarak görev yapmıştır. Ayrıca, 2006 yılından beri Avrupa Birliği (AB) Eğitim ve Gençlik Programları Merkezinde, proje değerlendirme uzmanı olarak görev yapmaktadır.

Anadili Türkçe olan Atlığ, iyi derece İngilizce bilmektedir. Halen Trakya Üniversitesi Bilgisayar Mühendisliği bölümünde Araştırma Görevlisi olarak görevini sürdürmektedir.

# EKLER

Aynı Salondaki Bilgisayarlar için

| Embeded | Tek PC | 2 PC |
|---|---|---|
| | | |
| **Ajan.java** | 1129062 | 8637677 |
| | | |
| **Peer.java** | 1024672 | 798826 |
| | | |
| Ajan+Peer | 2153734 | 9436503 |
| Enerji | 0,000000000038 | 0,000000000167 |
| | | |
| **Ajan.java** | 56607668 | 58633589 |
| | 59192893 | 57248755 |
| | 60138861 | 59186825 |
| | 62133846 | 59314606 |
| | 58653237 | 55134746 |
| Averaj | 59345301 | 57903704 |
| | | |
| **Peer.java** | 540765 | 246615 |
| | 453763 | 269020 |
| | 393143 | 295118 |
| | 378140 | 241460 |
| | 2135496 | 2116533 |
| Averaj | 780261 | 633749 |
| | | |
| Ajan+Peer | 60125562 | 58537453 |
| Enerji | 0,000000001062 | 0,000000001034 |
| | | |
| **Ajan.java** | 179163276 | 158333674 |
| | 191078618 | 158291348 |
| | 177339456 | 154027892 |
| | 174526148 | 166890819 |
| | 204356455 | 163209308 |
| | 185163619 | 153018277 |
| | 168126775 | 158333312 |
| | 179953977 | 161337470 |
| | 172647280 | 159732499 |
| | 191057751 | 156963986 |
| Averaj | 182341336 | 159013859 |
| | | |
| **Peer.java** | 1191407 | 582583 |
| | 1027664 | 661649 |
| | 975007 | 657296 |
| | 1318046 | 611704 |

| | | |
|---|---|---|
| | 1252747 | 602767 |
| | 1345105 | 872591 |
| | 1057438 | 662287 |
| | 1084873 | 691792 |
| | 1016680 | 550747 |
| | 5096285 | 4188281 |
| Averaj | 1536525 | 1008170 |
| | | |
| Ajan+Peer | 183877861 | 160022028 |
| Enerji | 0,000000003247 | 0,000000002826 |

| 3 PC | 4 PC | 5 PC |
|---|---|---|
| | | |
| 8449239 | 8566930 | 9005144 |
| | | |
| 320428 | 328904 | 949069 |
| | | |
| 8769667 | 8895834 | 9954213 |
| 0,000000000155 | 0,000000000157 | 0,000000000176 |
| | | |
| 60506310 | 58650168 | 59887592 |
| 55542854 | 59394402 | 58342068 |
| 59934356 | 58995478 | 57686569 |
| 59011568 | 59348160 | 58638310 |
| 59183785 | 60144277 | 56203695 |
| 58835775 | 59306497 | 58151647 |
| | | |
| 305390 | 375979 | 267916 |
| 415556 | 264300 | 283984 |
| 434966 | 305317 | 236831 |
| 331725 | 343638 | 336656 |
| 1776763 | 2579718 | 1941304 |
| 652880 | 773790 | 613338 |
| | | |
| 59488655 | 60080287 | 58764985 |
| 0,000000001051 | 0,000000001061 | 0,000000001038 |
| | | |
| 151686564 | 172349746 | 175803213 |
| 148069488 | 166454505 | 171959937 |
| 145355193 | 155914253 | 174107499 |
| 149188333 | 162172231 | 170111481 |
| 153502247 | 157643332 | 167579492 |
| 155807265 | 154957319 | 163641690 |
| 160126251 | 151114901 | 170252523 |
| 149617379 | 160492757 | 158612216 |
| 167165112 | 160202696 | 170468573 |
| 145353914 | 167053339 | 172002599 |
| 152587175 | 160835508 | 169453922 |
| | | |
| 790653 | 1052387 | 1044591 |
| 913829 | 1018025 | 939577 |
| 998800 | 891415 | 988248 |
| 851523 | 882777 | 1281678 |

| | | |
|---|---|---|
| 1040805 | 1004367 | 1166429 |
| 1005555 | 844558 | 851419 |
| 973270 | 1018476 | 1003367 |
| 862748 | 868937 | 958226 |
| 1105690 | 1135809 | 1073414 |
| 4071129 | 3914411 | 4931656 |
| 1261400 | 1263116 | 1423861 |
| | | |
| 153848575 | 162098624 | 170877783 |
| 0,000000002717 | 0,000000002863 | 0,000000003018 |

| 6 PC | 7 PC | 8 PC |
|---|---|---|
| | | |
| 9203416 | 9323416 | 9473445 |
| | | |
| 1086177 | 1214145 | 1304922 |
| | | |
| 10289593 | 10537561 | 10778367 |
| 0,000000000182 | 0,000000000186 | 0,000000000190 |
| | | |
| 63047562 | 64722202 | 60232277 |
| 57736683 | 60027739 | 58821788 |
| 59186786 | 59332286 | 65347634 |
| 59937740 | 57431788 | 61227345 |
| 64676203 | 64767771 | 65222453 |
| 60916995 | 61256357 | 62170299 |
| | | |
| 247776 | 255685 | 519801 |
| 244853 | 519801 | 239085 |
| 519801 | 244853 | 254432 |
| 255685 | 232876 | 247126 |
| 2709864 | 2982821 | 3054864 |
| 795596 | 847207 | 863062 |
| | | |
| 61712590 | 62103564 | 63033361 |
| 0,000000001090 | 0,000000001097 | 0,000000001113 |
| | | |
| 164368873 | 173473256 | 173402734 |
| 175502728 | 169322389 | 171968231 |
| 170463418 | 161991733 | 160212216 |
| 162694752 | 163677683 | 170252523 |
| 175511747 | 161944598 | 163641690 |
| 167902640 | 163127378 | 167579492 |
| 156091378 | 172247629 | 172311423 |
| 161447691 | 160344922 | 169107498 |
| 161944598 | 177232728 | 171959937 |
| 164447438 | 166348432 | 171903274 |
| 166037526 | 166971075 | 169233902 |
| | | |
| 514340 | 1321623 | 475627 |
| 604534 | 1226429 | 395889 |
| 486285 | 877478 | 557490 |
| 387378 | 1027667 | 399640 |

| | | |
|---|---|---|
| 482403 | 973426 | 401903 |
| 470682 | 1073414 | 501403 |
| 386889 | 953348 | 387378 |
| 550590 | 942365 | 486285 |
| 380640 | 1124544 | 632556 |
| 4017374 | 5021634 | 4247374 |
| 828112 | 1454193 | 848555 |
| | | |
| 166865638 | 168425268 | 170082456 |
| 0,000000002947 | 0,000000002974 | 0,000000003004 |

Aynı Salondaki Bilgisayarlar için

| gsServer | Tek PC | 2 PC |
|---|---|---|
| | | |
| **Ajan.java** | 9052825 | 8976856 |
| | | |
| **Peer.java** | 906959 | 931758 |
| | | |
| Ajan+Peer | 9959784 | 9908614 |
| Enerji | 0,000000000176 | 0,000000000175 |
| | | |
| **Ajan.java** | 62376100 | 24754265 |
| | 61598894 | 23494276 |
| | 63558857 | 24755074 |
| | 61685855 | 23493883 |
| | 63430626 | 24755398 |
| Averaj | 62530066 | 24250579 |
| | | |
| **Peer.java** | 399367 | 336236 |
| | 462314 | 360448 |
| | 239845 | 305226 |
| | 287137 | 369424 |
| | 2725871 | 2956658 |
| Averaj | 822907 | 865598 |
| | | |
| Ajan+Peer | 63352973 | 25116178 |
| Enerji | 0,000000001119 | 0,000000000444 |
| | | |
| **Ajan.java** | 101552275 | 32446625 |
| | 98202868 | 33397715 |
| | 101003150 | 33399470 |
| | 97588744 | 33400733 |
| | 95798587 | 33402216 |
| | 97871625 | 33403201 |
| | 97738864 | 33410996 |
| | 96064003 | 33430247 |
| | 98956127 | 33461448 |
| | 99049660 | 33516872 |
| Averaj | 98382590 | 33326952 |
| | | |
| **Peer.java** | 211278 | 1009944 |
| | 207075 | 894126 |
| | 284051 | 1318977 |
| | 296398 | 1163737 |

| | | |
|---|---|---|
| | 202201 | 1197124 |
| | 292685 | 1324786 |
| | 227625 | 1310618 |
| | 207804 | 992475 |
| | 274621 | 819781 |
| | 2259176 | 4813151 |
| Averaj | 446291 | 1484472 |
| | | |
| Ajan+Peer | 98828882 | 34811424 |
| Enerji | 0,000000001745 | 0,000000000615 |

| 3 PC | 4 PC | 5 PC |
|---|---|---|
| | | |
| 8606625 | 9048224 | 8898340 |
| | | |
| 441014 | 683212 | 886156 |
| | | |
| 9047639 | 9731436 | 9784496 |
| 0,000000000160 | 0,000000000172 | 0,000000000173 |
| | | |
| 59296824 | 59612694 | 58117664 |
| 59569486 | 59721090 | 58389021 |
| 58841929 | 58885490 | 57546054 |
| 59982578 | 59908029 | 58753141 |
| 57695278 | 59875258 | 58666378 |
| 59077219 | 59600513 | 58294452 |
| | | |
| 248788 | 301813 | 271341 |
| 303502 | 307809 | 229153 |
| 251419 | 337180 | 282286 |
| 225470 | 310982 | 278954 |
| 2170557 | 2340816 | 2844955 |
| 639947 | 719720 | 781338 |
| | | |
| 59717166 | 60320233 | 59075789 |
| 0,000000001055 | 0,000000001065 | 0,000000001043 |
| | | |
| 155259858 | 161762164 | 163273878 |
| 165863559 | 161752487 | 164848530 |
| 155544708 | 163631731 | 166887145 |
| 163579503 | 166130217 | 167021942 |
| 172503112 | 157298912 | 174734897 |
| 158252184 | 166819295 | 167032186 |
| 163046847 | 159236000 | 171589026 |
| 160370734 | 168699111 | 166264254 |
| 165825218 | 163761186 | 166878140 |
| 157384698 | 157656583 | 158964748 |
| 161763042 | 162674769 | 166749475 |
| | | |
| 992338 | 960931 | 881078 |
| 761260 | 850828 | 1031889 |
| 942601 | 823568 | 1077113 |
| 614625 | 963068 | 969597 |

| | | |
|---|---|---|
| 965160 | 887446 | 989546 |
| 863537 | 758992 | 1166254 |
| 772308 | 727322 | 819625 |
| 850257 | 967004 | 1180642 |
| 971105 | 871716 | 1111641 |
| 4022112 | 4949169 | 5000581 |
| 1175530 | 1276004 | 1422797 |
| | | |
| 162938572 | 163950773 | 168172271 |
| 0,000000002878 | 0,000000002895 | 0,000000002970 |

| 6 PC | 7 PC | 8 PC |
|---|---|---|
| | | |
| 9085224 | 9055834 | 9125877 |
| | | |
| 1086786 | 1186237 | 1253457 |
| | | |
| 10172010 | 10242071 | 10379334 |
| 0,000000000180 | 0,000000000181 | 0,000000000183 |
| | | |
| 58356052 | 58992052 | 60556308 |
| 59356111 | 60556308 | 59956322 |
| 60356052 | 59395432 | 60198308 |
| 60856367 | 58495432 | 59856674 |
| 58352252 | 59845773 | 60009898 |
| 59455367 | 59456999 | 60115502 |
| | | |
| 251991 | 300252 | 285636 |
| 278456 | 277834 | 275906 |
| 277264 | 276736 | 276344 |
| 297252 | 282486 | 305662 |
| 2724567 | 2774664 | 2755564 |
| 765906 | 782394 | 779822 |
| | | |
| 60221273 | 60239394 | 60895324 |
| 0,000000001064 | 0,000000001064 | 0,000000001075 |
| | | |
| 163364432 | 162343122 | 165544432 |
| 172878467 | 167328776 | 172873337 |
| 166264254 | 162447634 | 167234253 |
| 168089432 | 167021942 | 164443325 |
| 163273878 | 174734756 | 167328776 |
| 164848530 | 167032186 | 162447634 |
| 166887145 | 168089432 | 167021942 |
| 167021942 | 167234253 | 176544756 |
| 174734756 | 172878467 | 172032433 |
| 167032186 | 165544432 | 169959432 |
| 167439502 | 167465500 | 168543032 |
| | | |
| 969332 | 895078 | 1055872 |
| 986316 | 1150113 | 1237328 |
| 1224254 | 1376641 | 1100623 |
| 813455 | 1031889 | 983456 |

| | | |
|---|---|---|
| 1180642 | 1077113 | 1009641 |
| 1111641 | 1031889 | 1180642 |
| 1077113 | 986316 | 1323541 |
| 1031889 | 1009641 | 1150113 |
| 881078 | 1180642 | 843478 |
| 5221577 | 5235595 | 5215772 |
| 1449730 | 1497492 | 1510047 |
| | | |
| 168889232 | 168962992 | 170053079 |
| 0,000000002983 | 0,000000002984 | 0,000000003003 |

Uzak PC'ler için

| **Embeded** | **Tek PC** | **2 PC** |
|---|---|---|
| | | |
| **Ajan.java** | 1129062 | 9258108 |
| | | |
| **Peer.java** | 1024672 | 1362182 |
| | | |
| Ajan+Peer | 2153734 | 10620290 |
| Enerji | 0,000000000038 | 0,000000000188 |
| | | |
| **Ajan.java** | 56607668 | 65917572 |
| | 59192893 | 71037971 |
| | 60138861 | 67292484 |
| | 62133846 | 67341722 |
| | 58653237 | 64954685 |
| Averaj | 59345301 | 67308887 |
| | | |
| **Peer.java** | 540765 | 573192 |
| | 453763 | 651570 |
| | 393143 | 489575 |
| | 378140 | 521701 |
| | 2135496 | 2999898 |
| Averaj | 780261 | 1047187 |
| Ajan+Peer | 60125562 | 68356074 |
| Enerji | 0,000000001062 | 0,000000001207 |
| | | |
| **Ajan.java** | 179163276 | 154942610 |
| | 191078618 | 155685547 |
| | 177339456 | 157843130 |
| | 174526148 | 163461033 |
| | 204356455 | 160766120 |
| | 185163619 | 163950736 |
| | 168126775 | 173023366 |
| | 179953977 | 165784431 |
| | 172647280 | 155928749 |
| | 191057751 | 155123441 |
| Averaj | 182341336 | 160650916 |
| | | |
| **Peer.java** | 1191407 | 1851199 |
| | 1027664 | 1576313 |
| | 975007 | 1803737 |
| | 1318046 | 1706393 |
| | 1252747 | 2046773 |

| | | |
|---|---|---|
| | 1345105 | 1679286 |
| | 1057438 | 1809832 |
| | 1084873 | 1918194 |
| | 1016680 | 1800249 |
| | 5096285 | 3852844 |
| Averaj | 1536525 | 2004482 |
| Ajan+Peer | 183877861 | 162655398 |
| Enerji | 0,000000003247 | 0,000000002873 |

Uzak PC'ler için

| gsServer | Tek PC | 2 PC |
|---|---|---|
| | | |
| **Ajan.java** | 9052825 | 9870136 |
| | | |
| **Peer.java** | 906959 | 1141584 |
| | | |
| Ajan+Peer | 9959784 | 11011720 |
| Enerji | 0,000000000176 | 0,000000000194 |
| | | |
| **Ajan.java** | 62376100 | 68649497 |
| | 61598894 | 68143188 |
| | 63558857 | 67069476 |
| | 61685855 | 67073308 |
| | 63430626 | 68189374 |
| Averaj | 62530066 | 67824969 |
| | | |
| **Peer.java** | 399367 | 536803 |
| | 462314 | 508243 |
| | 239845 | 555783 |
| | 287137 | 570722 |
| | 2725871 | 2665760 |
| Averaj | 822907 | 967462 |
| Ajan+Peer | 63352973 | 68792431 |
| Enerji | 0,000000001119 | 0,000000001215 |
| | | |
| **Ajan.java** | 101552275 | 162156939 |
| | 98202868 | 160041954 |
| | 101003150 | 158082543 |
| | 97588744 | 163062588 |
| | 95798587 | 159343957 |
| | 97871625 | 159768186 |
| | 97738864 | 158929496 |
| | 96064003 | 163030117 |
| | 98956127 | 160141432 |
| | 99049660 | 158088122 |
| Averaj | 98382590 | 160264533 |
| | | |
| **Peer.java** | 211278 | 1617064 |
| | 207075 | 1653128 |
| | 284051 | 1513973 |
| | 296398 | 1613864 |
| | 202201 | 1651246 |

| | | |
|---|---|---|
| | 292685 | 1543243 |
| | 227625 | 1612754 |
| | 207804 | 1628151 |
| | 274621 | 1213952 |
| | 2259176 | 4039791 |
| Averaj | 446291 | 1808717 |
| | | |
| Ajan+Peer | 98828882 | 162073250 |
| Enerji | 0,000000001745 | 0,000000002862 |

## Bazı İstatistik Testlerin Kısa Tanımları

### Korelasyon Analizi

Bir gruptan alınmış iki veya daha fazla örnek grup alınarak, bu gruplar arasındaki ilişkiye bir oran ile incelenir. Korelasyon katsayısı olan bu oran *r* ile gösterilir. Değişken de denilen korelasyon analizinin yapılacağı gruplar arasında ilişki incelenirken, bağımlı değişken veya bağımsız değişken durumu önemli değildir. Korelasyon incelemesinde grup sayısına bakılmaksızın, ikili gruplar şeklinde incelenmektedir.

### Kruskal-Wallis Testi

Kruskal-Wallis testi en az üç değeri karşılaştırmak üzere kullanılan bir testtir. Test eşitliği parametrik olmayan gruptaki medianların sayısı üzerine yapılmaktadır. Mann-Whitney testinin testinin geliştirilmiş halidir. Parametrik olmaması sebebi ile Kruskal-Wallis testi normal bir dağılım kabul etmez.

### T-Test

T-testi istatistiksek olarak iki grup arasındaki farkı incelemektedir. Bu analiz iki grubun mean değerlerini karşılaştırmak istendiğinde uygun olmaktadır. Normal bir grubun mean değeri ile bir değişken üzerinden deney grubunun mean değerinin arasındaki fark incelenerek yapılmaktadır.