

Gömülü Sistem Üzerinde Uygulanan İkili Yedekleme Yöntemi ile Aksaklığa Dayanıklı Yazılım Mimarisi

Murat Cihan SORKUN^{1,2}, Salih BAYAR^{1,3}

¹ İdea Teknoloji Çözümleri A.Ş., Ar-Ge Merkezi, İstanbul

² Galatasaray Üniversitesi, Bilgisayar Mühendisliği Bölümü, İstanbul

³ Boğaziçi Üniversitesi, Bilgisayar Mühendisliği Bölümü, İstanbul

(Alınış / Received: 03.06.2016, Kabul / Accepted: 12.08.2016,
Online Yayınlanma / Published Online: 09.01.2017)

Anahtar Kelimeler Özet: Bu bildiride e-Fatura işlenmesinden sorumlu DIARIST Yazılım Mimarisi, Gömülü Sistem, İkili Yedekleme, Aksaklığa Dayanıklılık, E-Fatura

Özet: Bu bildiride e-Fatura işlenmesinden sorumlu DIARIST gömülü sistem cihazı üzerinde gerçekleştirilen, ikili yedekleme yöntemi (DMR) ile aksaklığa dayanıklılık uygulaması anlatılmaktadır. DIARIST cihazı kullanıcı tarafında bulunan, e-Faturaların işlenmesini, saklanması, yazıcıya aktarılmasını ve sunucu ile haberleşerek yedeklenmesini sağlayan özelleşmiş bir gömülü sistemdir. DIARIST sistemi vergi süreçlerini doğrudan etkilediği için kesintisiz ve hatasız hizmet vermesi gerekmektedir. Kesintisiz ve hatasız hizmet verebilmesi için, sistem üzerinde ikili yedekleme yazılım mimarisi ve uygulaması gerçekleştirilmiştir.

Fault Tolerant Software Architecture Applied on Embedded System Using Dual Modular Redundancy

Keywords Software Architecture, Embedded System, Dual Modular Redundancy, Fault Tolerance, E-Invoice

Abstract: In this paper, DIARIST embedded system, which is responsible for e-Invoice processing, is investigated by using dual modular redundancy (DMR) for fault tolerance. DIARIST device is a customized embedded platform that resides on the user side, is responsible for the processing of e-invoices, manages of the storage, communicates with the remote server, prints invoices on the local site. Since the DIARIST system affects the tax flow, it must give a continuous and flawless service. In order to provide uninterrupted and error-free service, DMR and corresponding software architecture are implemented on the system.

*Sorumlu yazar : murat.sorkun@ideateknoloji.com.tr

1. Giriş

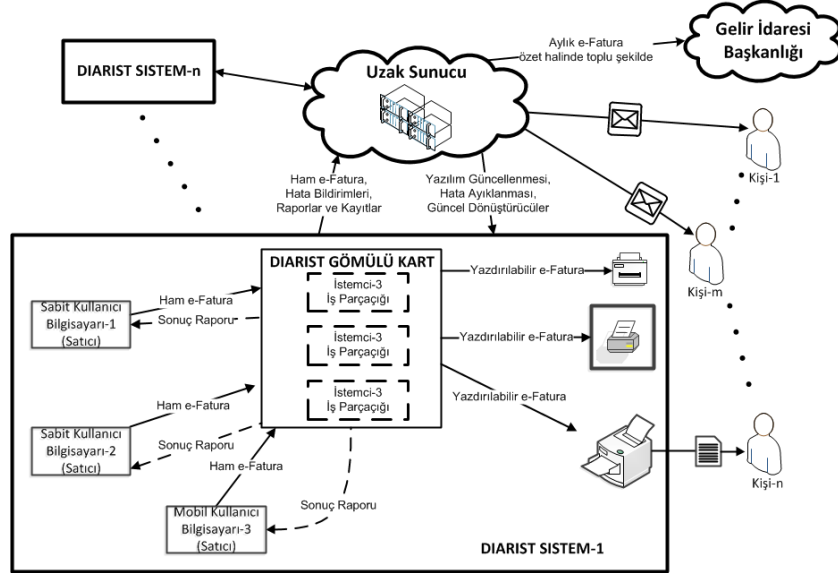
E-Fatura uygulaması, vergi mükellefleri tarafından kesilen faturaların kâğıt belgeden, elektronik ortama taşınmasını sağlamıştır. Türkiye ve birçok ülkede uygulanmakta olup, kullanım alanı sürekli genişlemektedir. E-Faturanın en temel özelliği, oluşturulmuş elektronik belgeleri taraflar arasında güvenli ve sağlıklı dolaşımını sağlamasıdır [1]. E-Fatura sürecinde faturaları kesen ve alan taraflar, Gelir İdaresi Başkanlığı tarafından kayıtlı e-Fatura mükellefi olmalıdır. Eğer kesen taraf e-Fatura mükellefi ve alan taraf e-Fatura mükellefi değil ise, kesilen faturalar e-Arşiv Fatura ismini almaktadır. E-Arşiv Fatura alıcıya, basılı olarak ya da basılı hali elektronik olarak teslim edilmeli ve bu fatura arşivlenmelidir. Faturaların yazdırılabilir hale gelmesi boyunca geçen süreç, uzak sunucu üzerinde gerçekleşmekte ve bu yüzden, hali hazırdaki E-Arşiv Fatura işleyişi yavaş ve maliyetli olmaktadır [1].

DIARIST, bu problem üzerine geliştirilmiş, süreci sunucu üzerinden kullanıcı lokaline taşıyan bir gömülü sistemdir. DIARIST sistemi ile kullanıcı tarafından oluşturulan ham haldeki fatura, tüm denetim ve dönüşüm işlemlerinden yine kullanıcı lokalinde geçmekte ve bastırılabilir hale dönüştürülmektedir. Aynı zamanda bu faturalar cihaz üzerinde saklanmakta ve arşivlenmek üzere, asenkron olarak uzak sunucuya gönderilmektedir. Şekil 1'de DIARIST sisteminin genel sistem mimarisi verilmiştir.

DIARIST sisteminin kesintisiz ve hatasız çalışması, kullanıcıların vergi denetimlerinde sorun yaşamamaları açısından oldukça önemlidir. Sistemin kesintisiz ve hatasız çalışması için ikili yedekleme sistemi uygulanmıştır. İkili yedekleme sistemi, sürecin birbirinin eşleniği iki sistem tarafından

gerçekleştirilmesi ve sonuçların hash değerlerinin oylanması ile karar verilmesine dayalı bir aksaklık tespiti yöntemidir [2]. Eşlenik cihazlardan gelen sonuçlar karşılaştırılarak sistemdeki aksaklıklar yakalanmaktadır. Karşılaşılabilecek aksaklıklar donanımsal veya yazılımsal olabilmektedir. Örnek olarak; cihazın arızalanması ve o anda yanıt alınamaması gibi donanımsal hatalar olabileceği gibi, cihazlardaki yazılımların güncellenmesi süresince bir cihazdaki yazılım eski sürümde kalması gibi farklı sonuçlar üretecek yazılımsal problemlerde oluşabilmektedir. İki cihazdan herhangi birinde hata oluşması durumunda sonuçlar farklı üretilmekte, karşılaştırma sonucunda eşleşme gerçekleşmediği için aksaklık tespit edilmektedir. İkili yedekleme sistemlerinde aksaklık olduğu tespit edilmekte ancak aksaklığın eşlenik cihazlardan hangisinde olduğu bilinmemektedir. Aksaklığı oluşturan cihazı tespit etmek için üçlü yedekleme sistemleri kullanılmakta, ancak bu iki yerine üç adet eşlenik sistemin oluşturulmasını gerektirmektedir. Bu yüzden sadece hatanın tespitinin yeterli olduğu birçok sistemde üçlü yerine ikili yedekleme sistemi kullanılmaktadır

Üçlü yedekleme sistemlerinde tüm süreç üç adet eşlenik cihaz için de ayrı olarak gerçekleştirilmektedir. Üçlü yedekleme ile aksaklığa dayanıklılıkta sağlanan avantaj, oy çokluğu ile hatanın hangi oylayıcıdan kaynaklandığının tespit edilebilmesidir [4]. Üçlü yedekleme tabanlı aksaklığa dayanıklılıkta temel bileşenlerinin iki adet kopyasına ek olarak oylayıcı modülünün de çoklu olduğu tasarımlar mevcuttur [5]. Eşlenik modüllerin sayılarının artması ile karmaşıklık düzey ve maliyet de yükselmektedir. Bu çalışmada sadece ikili yedekleyici ve tek oylayıcı içeren sistemin uygulaması incelenmiştir.



Şekil 1. DIARIST Sistem Mimarisi

2. DMR-Localvoter Sistemi (İkili Yedekleme- Yerel Oylayıcı)

İkili Yedekleme- Yerel Oylayıcı sistemi ile aksaklığa dayanıklılık senaryosu bileşenleri aşağıdaki gibidir:

- İkili Yedekleme Modülü (DMR): İkili yedekleme tabanlı aksaklığa dayanıklılıkta temel mantık sisteme ait bileşenlerinin bir kopyasının daha oluşturulması esasına dayanmaktadır. Bu yapıda her iki sistem de birbirinden tamamen bağımsız olarak çalışmakta olup, aynı anda aynı işlemleri yapmaktadır. İkili yedekleme modülünden çıkan çıktılar kontrol edilmek üzere oylayıcı modüle gönderilir.

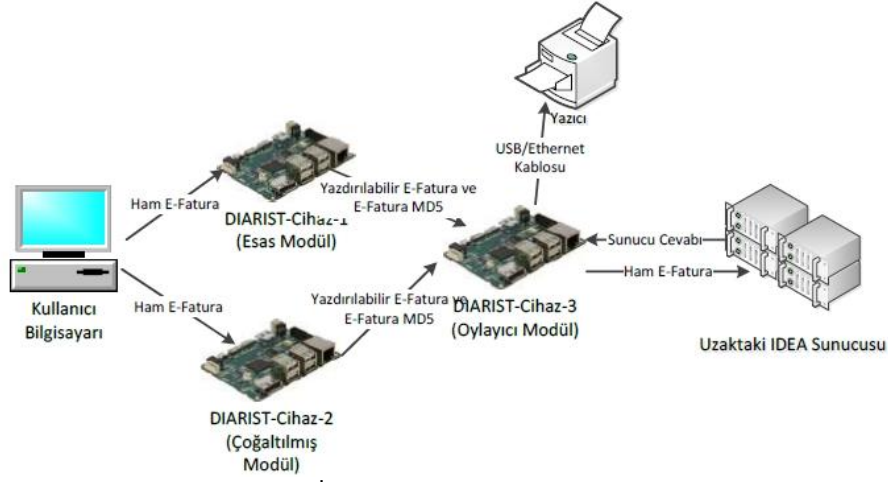
- Yerel Oylayıcı Modülü (LocalVoter): İkili yedekleme modülünden gelen çıktıların karşılaştırılması ile hata tespitinin yapıldığı modüldür. Oylayıcı modül, ikili yedekleme modülü ile beraber aynı yerel ağda bulunmaktadır. Hata tespiti bu modül üzerinde yerel olarak yapılmaktadır.

• Hata Tespiti:

İkili yedekleme tabanlı aksaklığa dayanıklı sistemlerde hata tespiti mümkün olurken, hatanın giderilmesi mümkün olmamaktadır. Çünkü hatanın hangi sistemden geldiği bilinmemektedir. Burada tespit edilebilen şey, var olan iki paralel sistemin aynı giriş verilerine aynı ya da farklı cevap vermesidir. Bu yüzden aynı giriş için iki sistemin çıktılarının karşılaştırılması gerekmektedir. Eğer çıktılar aynıysa sistemde herhangi bir hata olmadığı kanısına varılır. Eğer olur da bu çıktılar birbiri ile eşleşmezlerse, sistemlerin en az birisinde bir hata olduğu sonucuna varılır. Ancak hangi sistemin hatalı olduğu bilinmemektedir.

2.1. DIARIST DMR-Localvoter yapısı

DIARIST DMR-LocalVoter sisteminde, iki yedekleme modülü ve bir oylayıcı modül olmak üzere toplamda üç adet DIARIST cihazı kullanılmıştır. Yedekleme modülleri faturaları paralel olarak işleyip kontrol için oylayıcı modüle



Şekil 2. DIARIST İkili Yedekleme – Yerel Oylayıcı Yapısı

göndermektedir. Bu modüllerin görevi sadece faturaları işlemek ve oylayıcıya göndermektir. Bu modüller fatura işleme süresince veri tabanı, yazıcı ve sunucu ile hiçbir bağlantı kurmamaktadır. Oylayıcı modül ise gelen işlenmiş faturaların hash değerlerini karşılaştırarak hata tespiti yapmaktadır. Oylama sonucuna göre, bastırılabilir halde alınmış fatura yazıcıya gönderilip bastırılabilir. Sunucu ile haberleşme yine bu oylayıcı modül üzerinden gerçekleşmektedir. Şekil 2’de DIARIST ikili yedekleme, yerel oylayıcı yapısı verilmiştir.

2.2. Diarist Dmr-Localvoter aksaklığa dayanıklılık süreci

DIARIST DMR-LocalVoter sistemi ile aksaklığa dayanıklılık süreci aşağıdaki adımlarda gerçekleşmektedir.

• Ham Fatura Girişi:

Kullanıcı bilgisayar tarafından, ham faturanın işlenmek üzere ikili modüllere gönderilmesi ile süreç başlar. Kullanıcı tarafından iki cihaza da aynı fatura, tüm süreçte takip edilmesini takip edilmesini sağlayacak olan UUID (Eşsiz ID) de eklenerek gönderilir.

• Faturaların İşlenmesi:

Her bir işleyici DIARIST modülü ham haldeki faturayı kontrollerden geçirir ve işler. İşlenmiş olan bastırılabilir haldeki faturayı (pdf) , ham haldeki faturayı, karşılaştırmada kullanılacak olan işlenmiş faturanın hash değerini ve kendi cihaz numarasını oylayıcı modüle gönderir. Bu süreçte veri tabanı ile hiçbir bağlantı kurulmamaktadır. İşleme sonucunda alınan tüm sonuçlar oylayıcı modüle aktarılmaktadır.

• Oylama İşlemi:

İşleyici modülden gelen fatura için ilk olarak UUID değeri ile ilk oy mu, ikinci oy mu olduğu bilgisi veri tabanından kontrol edilir. Eğer ilk oysa tüm bilgiler VoteRequest (Tablo-1) ve Invoice (Tablo-2) tablolarına kaydedilir. Bu aşamada ikinci oy gelene kadar herhangi bir karşılaştırma yapılmaz.

İkinci oy geldiğinde (veri tabanında aynı UUID ile daha önce bir kayıt varsa) bu kaydın hangi cihazdan geldiği kontrol edilir. Eğer cihaz numaraları aynı değilse ikinci oy olarak kabul edilir, oy sadece VoteRequest tablosuna kaydedilir ve oylama süreci başlar.

Oylama sürecinde iki faturanın işleyici modüllerden veri tabanına kaydedilmiş olan hash değerleri (MD5) karşılaştırılır.

Eğer hash değerleri aynı ise süreç doğru olarak kabul edilir ve fatura yazıcıya

Tablo 1. VoteRequest

Kolon Adı	İçerik
RECORD_ID	Kayıt Numarası
INVOICE_UUID	Faturanın takip edildiği UUID değeri
ISCOMPLETED	İşleyici modül tarafından gönderilen başarı durumu
MESSAGE	İşleyici tarafından gönderilen mesaj
HASH	Oylamada kullanılacak Hash değeri
LOGTIME	Kayıt zamanı
CONNECTIONID	Cihaz Numarası
STATUS	Oylama durumu (Waiting, Equal, Unequal)

Tablo 2. Invoice

Kolon Adı	İçerik
RECORD_ID	Kayıt Numarası
INVOICE_UUID	Faturanın takip edildiği UUID değeri
INVOICE_NUMBER	Fatura Numarası
ZIPPED_SOURCEFILE	Zipli halde basılabilir Fatura
ZIPPED_UBLFILE	Zipli halde UBL Fatura
RECEIVED_TIME	Kayıt zamanı
ISPRINTED	Basılma bilgisi
PRINT_TIME	Basılma zamanı
ISPRINTCOMPLETED	Basılma kontrol bilgisi
ISSENT	Sunucuya gönderilme bilgisi
ISSENTCOMPLETED	Sunucuya gönderilme kontrol bilgisi
SENT_TIME	Sunucuya gönderilme zamanı
SEND_ERROR	Sunucudan gelen hata
STATUS	Fatura Durumu

gönderilir. Tablolardaki fatura ve oylama durumları tamamlandı olarak güncellenir. Hash değerlerinin eşleşmemesi sürecin hatalı olduğu anlamına gelmektedir.

Oylama işlemlerinde faturalar farklı olsa bile, teorik olarak hash değerlerinin aynı gelme olasılıkları bulunmaktadır. Ancak 128 bitlik hash değerleri için çakışma olma olasılığı yaklaşık olarak 2128 büyüklüğündedir. Bu yüzden sistemde çakışma olasılığı ihmal edilmektedir.

•Hata Tespiti:

Oylama sonucunda hash değerlerinin eşleşmemesi işleyici modüllerde hata olduğu anlamına gelir. Aynı faturayı işleyen modüller farklı sonuç üretmişlerdir. Hata durumu sunucuya gönderilmek üzere veri tabanına kaydedilir. Sistemde hata olduğu tespit

edilmiş olsa da modüllerden hangisinin hatalı sonuç ürettiği bilinemez. Bu yüzden iki fatura da hatalı kabul edilir ve yazıcıya gönderim yapılmaz.

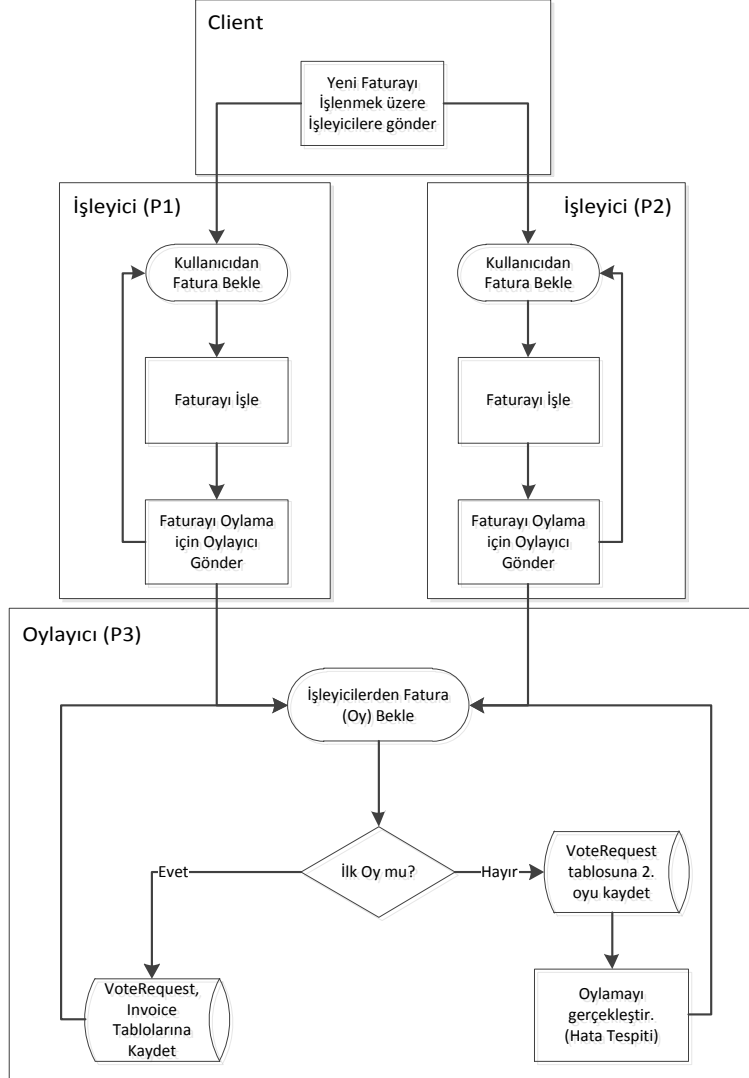
3. DMR-Localvoter Yazılım Mimarisi

DMR-LocalVoter aksaklığa dayanıklılık uygulaması için birbiri ile entegre olarak çalışacak 3 farklı yazılım geliştirmiştir.

İlk yazılım işleyici modüller için geliştirilmiş olup, temelde ham haldeki faturayı yazdırılabilir hale getirme işlemini yapmaktadır. Ham faturayı Client tarafından alıp işledikten sonra oylayıcı modüle gönderen bir servis olarak çalışmaktadır.

İkinci yazılım oylayıcı yazılımdır. İşleyicilerden gelen faturaları oylar ve hata olup olmadığını tespit eder. Bu faturaların bastırılması, veri tabanında

saklanması ve idea sunucularına yapılmaktadır. gönderilmesi bu aşamada



Şekil 3. DIARIST İkili Yedekleme – Yerel Oylayıcı Yazılım Mimarisi

Üçüncü yazılım Client (kullanıcı) yazılımıdır. Bu yazılım test amaçlı faturaları işleyicilere göndermektedir. Tüm modüllerin birbirleri ile entegre çalıştıkları mimari Şekil 3'te detaylı olarak gösterilmektedir.

4. DMR-Localvoter Aksaklığa Dayanıklılık Uygulaması

DMR-LocalVoter aksaklığa dayanıklılık uygulaması Odroid cihazlar üzerinde

geliştirilmiş olup İşleyici modül, Oylayıcı modül ve İstemci için ayrı yazılımlar geliştirilmiştir. Bu uygulamada kullanılan cihazlara ait detaylı bilgiler aşağıda verilmiştir.

Uygulamada Kullanılan Cihaz Bilgileri:

Uygulamada kullanılan tüm cihazlar TCP/IP ile soket üzerinden haberleşmektedir. Kullanılan 4 adet

cihaz ve işletim sistemleri aşağıda belirtilmiştir.

• P1, P2, P3: Odroid XU4 (Ubuntu 14.04.4 LTS)

• Client: (Windows)

Tablo 3. 10 Adet E-Fatura Verisi ile DMR-LocalVoter Test Sonuçları

Çalışma Adımları	Çalışma Süresi (ms)	Yüzde
1- Client Fatura Gönderme	44550	38,05
2- P1 Fatura İşleme	17166	14,66
3- P2 Fatura İşleme	17314	14,78
4- P3 Oylama-Yazdırma	38048	32,49
Toplam (Tüm İşlemler Toplamı):	117078	100
Toplam (Paralel Süreçler Toplamı):	86411	-

Tablo 4. Hatalı olarak gönderilen 5 faturanın oylama ile tespiti

Invoice Number	P1-Hash	P2-Hash	Voting Result
ABC0241	9fb6fee3f385ad3...	1a6d40e2e1973ff...	Unequal
ABC0256	1c634e4dec50e3b...	1c634e4dec50e3b...	Equal
ABC0257	80b7044a5137ad4...	4ed066cdaa08280...	Unequal
ABC0259	466b225538fa19e...	f90a0ee1c67961f...	Unequal
ABC0362	f322fa50e6346db...	f322fa50e6346db...	Equal
ABC0366	ef16f9062061abf...	ef16f9062061abf...	Equal
ABC0372	f8a77842004c2bb...	f8a77842004c2bb...	Equal
ABC0373	15c327923c198cf...	0018827e9946cf1...	Unequal
ABC0377	1de1a8c9e43c911...	cc20e60ec580186...	Unequal
ABC0378	50a3be8cd2685ef...	50a3be8cd2685ef...	Equal

4.1 10 Adet Hatasız Faturanın Oylama Senaryosu

Bu senaryoda 10 adet hatasız fatura DMR-Local Voter sisteminde test edilmiştir. Bu testin amacı çoklu fatura gönderiminde süreçlerde tıkanma olup olmadığını tespit etmek ve test süresince sistemin farklı adımlarında harcadıkları süreleri gözlemlemektir.

Test süresince Client uygulaması 10 adet hatasız faturayı işlenmek üzere iki işleyici modüle de göndermiştir. İki işleyici de paralel olarak aynı faturaları işlemiş ve oylayıcı modüle oylanmak üzere göndermiştir.

Tablo 3' de 10 adet e-Fatura'nın kullanıcı bilgisayarından DMR-LocalVoter sistemine gönderilip, paralel olarak işlenen bu fatura verisinin sistemin ana 4 farklı adımında her bir adım için ayrı olarak ne kadar süre geçirdiği milisaniye cinsinden

verilmiştir. Bu ayrı süreçlerin toplam olarak bağımsız çalışma süresi 'Tüm İşlemler Toplamı' satırında verilmiştir. Sürecin bir bütün olarak çalıştırılması ve paralel süreçlerin kazandırdığı zaman ile ölçülen çalışma süresi 'Paralel Süreçler Toplamı' satırında verilmiştir.

Test süresince tüm faturalar düzgün bir şekilde işlenmiştir. Oylama sürecinde yapılan karşılaştırmada faturaların hash değerleri aynı çıkmış ve başarılı bir şekilde süreç tamamlanmıştır.

4.2 5 Adet Hatalı 5 Adet Doğru İşlenmiş Faturanın Oylama Senaryosu

Bu senaryoda 10 adet hatasız faturanın 5'i işleyicilerden birinde kontrollü olarak bozulmuş ve farklı hash değeri gönderilmiştir. Bu testin amacı sistemin hatalı faturaları tespit edip, sistemdeki aksaklığın yakalanmasını gözlemlemektir.

Test süresince Client uygulaması 10 adet hatasız faturayı işlenmek üzere iki işleyici modüle de göndermiştir. İşleyici modüllerden ilki faturaları doğru bir şekilde işleyip oylayıcıya göndermiştir. İkinci modül ise fatura numarasının son karakteri çift sayı olan faturaların hash değerleri kontrollü olarak bozulmuş ve bu şekilde oylamaya gönderilmiştir.

Oylama sürecinde hatalı olarak gönderilen 5 adet fatura başarıyla fark edilmiş, sistemin aksaklığı tespit edilmiştir. Tablo 4'te hatalı olan faturaların 'unequal' statüsüne düşürüldüğünü gözlemlemekteyiz.

5. Sonuçlar

DIARIST sistemi üzerinde ikili yedekleme ile aksaklığa dayanıklılık testleri için iki farklı senaryo uygulanmıştır. İlk senaryoda 10 adet faturanın sistem üzerinde döngüsünü tamamlaması test edilmiş. Bu test ile sistemin çoklu faturalarda düzgün çalıştığı, aynı hash değerlerini üretebildiği gözlemlenmiştir. Bu senaryonun bir diğer amacı faturaların işleme ve oylanma sırasında modüllerde harcadığı sürelerin ve toplam sürenin tespit edilmesidir. Elde edilen süreler Tablo 3'te verilmişti, bu sürelerin aksaklığa dayanıklılık içermeyen, tek DIARIST cihazı içeren sistem ile 10 fatura için süre karşılaştırması Tablo 5'te verilmektedir. İkili yedekleme sistemi süre üzerinde %19'luk bir artışa sebep olmuştur.

Tablo 5. İkili yedekleme ve tekil sistem süre karşılaştırması

İkili Yedekleme (ms)	Tekil Sistem (ms)
86411	72586

İkinci senaryoda ise gönderilen çok sayıda fatura arasında aksaklığa uğrayanların yakalanması hedeflenmiştir. Kontrollü olarak bozulan 5 fatura da sistem tarafından

yakalanmıştır. Tablo 4'te görüldüğü üzere %100 başarı ile aksaklıklar tespit edilmiştir.

Test süresince faturaların hash değerleri bozularak sonuçlar karşılaştırılmıştır. Gelecek çalışmalarda testlerin hash değerleri yerine faturanın içerikleri değiştirilerek daha kapsayıcı testler yapılması amaçlanmaktadır.

Kaynakça

- [1] S. Bayar, M. G. Ülkar, A. Şen, "Kullanıcı Tarafında E-Bege Oluşturma ve Yazdırma Yazılım Deneyimleri", 9. Ulusal Yazılım Mühendisliği Sempozyumu (UYMS'15), İzmir, Türkiye, Eylül 2015.
- [2] V. Petrovic, G. Schoof, Z. Stamenkovic, "Fault-tolerant TMR and DMR Circuits with Latchup Protection Switches", Microelectronics Reliability, Ağustos 2014.
- [3] V. Petrovic, M. Ilic, G. Schoof, Z. Stamenkovic, "Design Methodology for Fault Tolerant ASICs", 15th IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems, Microelectronics Reliability Cilt 54, no. 8 Nisan 2012.
- [4] R.E. Lyons, W. Vanderkulk, "The Use of Triple-Modular Redundancy to Improve Computer Reliability", in IBM Journal of Research and Development, Cilt.6, no.2, ss.200-209, April 1962.
- [5] S. Habinc, "Functional Triple Modular Redundancy (FTMR). VHDL Design Methodology for Redundancy in Combinatorial and Sequential Logic.", Gaisler Research, Design and Assessment Report, 2002.