Tinc VPN ile Güçlendirilmiş Merkeziyetsiz Dosya Depolama ve Paylaşım Uygulaması

Araştırma Makalesi/Research Article

Alihan ÖZEN[1], Derya YILTAŞ KAPLAN[1]*

[1]Bilgisayar Mühendisliği Bölümü, İstanbul Üniversitesi-Cerrahpaşa, İstanbul, Türkiye.
alihanozen76@gmail.com, dyiltas@iuc.edu.tr
(Geliş/Received:25.10.2024; Kabul/Accepted:28.01.2025)
DOI: 10.17671/gazibtd.1573426

Özet— Kuruluşlar ve özellikle şirketler için veri güvenliği kritik bir öneme sahiptir. Verilerin paylaşımı sırasında kullanılan ağın gizliliği, hata toleransı ve sistemin dayanıklılığı gibi faktörler, güvenlik ve gizliliği doğrudan etkilemektedir. Geleneksel istemci-sunucu modelinin bazı önemli eksiklikleri bulunmaktadır. Bu modelde genellikle bir veya iki sunucu bulunur ve istemciler bu sunuculara dosya paylaşımı veya erişimi talepleri gönderirler. Ancak, merkezi bir sunucuya bağlı olan bu sistemler, sunucunun arızalanması veya saldırıya uğraması durumunda işlevsiz hale gelebilir. Ayrıca, sistem bileşenlerinden birinin çökmesi, tüm sistemi devre dışı bırakabilir ve güvenlik açıklarına yol açabilir. Kullanılan ağ genellikle sanal, gizli veya özel değildir, bu da veri paylaşımını güvenlik açısından riskli hale getirmektedir. Bu çalışmada amaç, merkeziyetsiz bir özel ağ içerisinde eşten eşe dosya depolama ve paylaşım sisteminin oluşturulmasıdır. Çalışma, IPFS ve Tinc VPN kullanılarak güvenli ve dayanıklı bir veri paylaşım altyapısı sunmayı hedeflemektedir. Tasarlanan sistem, veri paylaşımı sırasında gizlilik, dayanıklılık ve güvenliği ön planda tutan bir yapı sunmaktadır.

Anahtar Kelimeler— depolama ve paylaşım, ipfs, merkeziyetsiz, sanal özel ağ, tinc, veri aktarımı

Decentralized File Storage and Sharing Implementation Powered by Tinc VPN

Abstract— Data security is of critical importance for organizations and especially companies. Factors such as the confidentiality of the network used during data sharing, fault tolerance and system durability directly affect security and privacy. The traditional client-server model has some important deficiencies. In this model, there are usually one or two servers and clients send file sharing or access requests to these servers. However, these systems, which are connected to a central server, can become dysfunctional if the server malfunctions or is attacked. In addition, a crash of one of the system components can disable the entire system and lead to security vulnerabilities. The network used is usually not virtual, private or private, which makes data sharing risky in terms of security. The aim of this study is to create a peer-to-peer file storage and sharing system within a decentralized private network. The study aims to provide a secure and durable data sharing infrastructure using IPFS and Tinc VPN. The designed system offers a structure that prioritizes confidentiality, durability and security during data sharing.

Keywords— storage and sharing, ipfs, decentralized, virtual private network, tinc, data transfer

1. GİRİŞ (INTRODUCTION)

Kurum ve kuruluşlar için dosya paylaşımının güvenli ve dayanıklı bir şekilde merkezi olmayan bir yapıda gerçekleştirilmesi, iş verimliliği ve iletişim açısından kritik bir rol oynamaktadır. Geleneksel merkezi dosya paylaşım sistemleri, etkin merkezi kontrol ve kaynak yönetimi ile kullanıcıya özerklik sağlasa da [1] çeşitli dezavantajlara sahiptir. Bu dezavantajlar, sistemin dayanıklılığı, hata toleransı ve güvenlik açığı ile ilgili bazı sorunları içermektedir [2]. Merkezi sistemlerde sıklıkla karşılaşılan sorunlar şunlardır:

- **Dayanıklılık Eksikliği:** Merkezi sistemler, tek bir veya sınırlı sayıda sunucuya bağlıdır. Sunuculardan birinin çökmesi durumunda sistemin işlerliği kesintiye uğrar. Ayrıca, hedef belirli olduğu için saldırılar karşısında savunmasız kalırlar. Bu tür sistemlerde tek bir sunucunun devre dışı kalması bile sistemin genel işleyişini olumsuz etkiler.
- **Düşük Hata Toleransı:** Merkezi sistemlerde sadece ana düğüme (node) talepler gönderilir. Ana düğümde herhangi bir hata oluştuğunda sistem yanıt vermez ve işleyiş kesintiye uğrar. Merkezi sistemlerin sınırlı hata toleransı nedeniyle güvenlik riski yüksektir.
- **Ağın Güvenlik Açığı:** Kullanılan ağ sanal, gizli veya özel olmadığı için güvenlik zayıftır. Bu durum, verilerin güvenli bir şekilde iletilmesini zorlaştırır ve saldırılara karşı açık hale getirir. Böylece veri bütünlüğü ve gizlilik riskleri de ortaya çıkmış olur [1].

Bu çalışmada, merkezi sistemlerdeki bu problemlerin üstesinden gelmek amacıyla, Gezegenler Arası Dosya Sistemi (InterPlanetary File System, IPFS) ve Tinc VPN (Virtual Private Network, Sanal Özel Ağ) kullanılarak merkeziyetsiz ve güvenli bir dosya paylaşım altyapısı oluşturulmuştur. IPFS, dosya paylaşımını merkezi olmayan bir ağ üzerinde gerçekleştirirken Tinc VPN, ağın güvenliğini ve gizliliğini sağlar. Önerilen sistemin temel avantajları şunlardır:

- **Yüksek Dayanıklılık:** Sistem, beklenen veya beklenmeyen değişikliklere karşı kendi işleyişini sürdürme yeteneğine sahiptir. Bu, dosya paylaşımının sürekli bir şekilde devam etmesini sağlar.
- **Güvenilir Veri Saklama:** Veriler, güvenli bir şekilde saklanır ve sadece yetkili kişilerle paylaşılır. Verilerin yedeklenme ihtiyacı ortadan kalkar, çünkü her dosya birçok farklı düğümde depolanır.
- **İstemci-Sunucu Modelinin Ötesinde:** Sistemdeki her düğüm hem istemci hem de sunucu olabilir. Bu, dosya paylaşımında esneklik sağlar.
- **Kesintisiz Güncellemeler:** Sistem güncellemeleri, işleyişi kesintiye uğratmadan yapılabilir.
- **Ölçeklenebilirlik:** IPFS ve Tinc VPN, sisteme daha fazla düğüm eklenmesine olanak tanır, böylece ağ genişledikçe performans da artar.
- **Güvenli ve Gizli İletişim:** Tinc VPN, düğümler arasındaki iletişimi güvenli hale getirir, böylece veri aktarımı sırasında gizlilik sağlanır.
- **Merkezi Denetimden Bağımsızlık:** Sistem, merkezi bir otoriteye ihtiyaç duymadan işleyebildiği için sansür ve izleme riskleri minimize edilmiş olur.

IPFS, içerik adreslemesi ile dosyalara ağ üzerinde hızlı ve güvenilir bir şekilde erişimi sağlar. Dosyalar, içeriklerine göre benzersiz kimlikler ile tanımlanır [2] ve her dosyanın bütünlüğü IPFS'nin oluşturduğu özet (hash) fonksiyonları ile korunur [3-4]. Bu sistem, veri güvenliği ve erişim hızını artırarak merkeziyetsiz dosya paylaşım altyapısını sağlamlaştırır.

Sonuç olarak, bu çalışmada önerilen sistem, veri güvenliği ve dayanıklılık açısından merkezi sistemlere göre üstünlük sağlar. Ağ üzerindeki her düğümün eşit derecede işlevsel olduğu bu modelde, sistemin herhangi bir parçası arızalansa dahi işleyiş kesintiye uğramaz. Ayrıca, sistemin hata toleransı yüksektir ve sistem saldırılara karşı dirençlidir. IPFS ve Tinc VPN teknolojileri ile desteklenen bu yapı, güvenli ve sürdürülebilir bir dosya paylaşım altyapısı sunmaktadır.

Bölüm 2'de sanal özel ağ yapısı gibi çalışmada geçen temel kavramlarla ilgili açıklamalar sunulmaktadır. Bölüm 3'te literatürdeki farklı çalışmalar incelenmektedir. Bölüm 4'te uygulamada kullanılan yazılım teknolojileri ile altyapılar ele alınmaktadır. Bölüm 5'te ağ testlerinin sonuçları ve uygulama arayüzü sunulmaktadır. Bölüm 6'da önerilen sistemin avantajları ve kısıtları tartışılmaktadır. Son olarak Bölüm 7'de gelecek çalışmalarla ilgili öneriler verilmektedir.

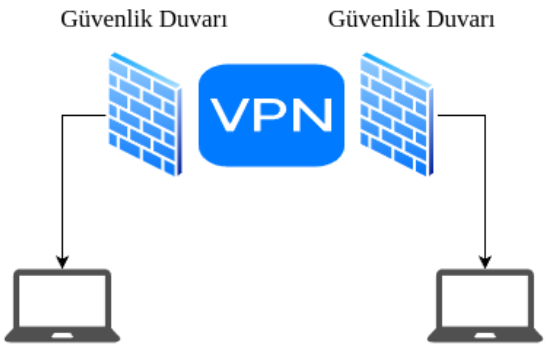
2. GENEL TANIMLAR (GENERAL DEFINITIONS)

Bu çalışmada ele alınan temel konular, sanal özel ağlar ve merkeziyetsiz depolama sistemleridir.

2.1. Sanal Özel Ağlar (Virtual Private Networks)

VPN, kullanıcıların internete açık bir ağ üzerinden güvenli ve gizli bağlantılar kurmalarını sağlayan bir ağ teknolojisidir. Bir VPN, kullanıcıyı fiziksel olarak o ağa bağlı olmadan, farklı bir ağa sanal bir şekilde bağlar. Bu bağlamda VPN, veri transferi sırasında kullanıcıların konum bilgilerini gizler ve güvenli veri aktarımı sağlar. Özellikle şirket içi ağlara uzaktan erişim sağlamak için kullanılan VPN teknolojisi, internet üzerinden güvenli bir tünel oluşturarak özel ağlara erişim imkânı sunar [5].

VPN'nin sağladığı en büyük avantajlardan biri, verilerin şifrelenerek aktarılması ve bu sayede yetkisiz erişimlerin önlenmesidir. VPN aynı zamanda esneklik, güvenlik ve yüksek üretkenlik gibi faydalara sahiptir [6]. Şekil 1, temel bir VPN yapısını şematik olarak göstermektedir.



Şekil 1. VPN şeması
(Figure 1. VPN schema)

Tablo 1'de VPN'nin güvenlik ve erişim avantajlarının yanı sıra yapılandırma hataları ve hız düşüşü gibi dezavantajları vurgulanmaktadır.

Tablo 1. VPN'nin avantajları ve dezavantajları
(Table 1. Advantages and disadvantages of VPN)

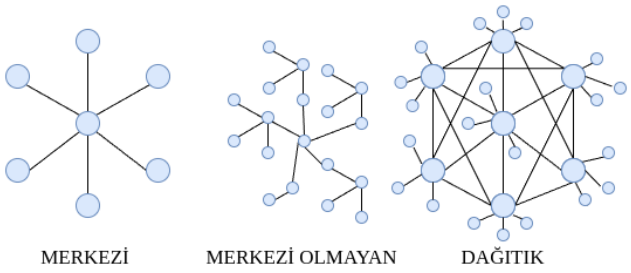
VPN'nin Avantajları	*VPN'nin Dezavantajları*
Veriler, şifrelenerek güvenli bir şekilde iletilir. Böylece internet tehditlerine karşı koruma sağlanır.	Yanlış yapılandırılması durumunda DNS ve IP sızıntıları gibi güvenlik açıkları oluşabilir.
Kablosuz ağlar üzerinden güvenli bağlantılar kurulmasına olanak tanır.	Kurulumu ve yönetimi zor olabilir. Özellikle büyük ağlarda yapılandırma hataları güvenlik sorunlarına yol açabilir.
Uzak cihazlar, şirket içi ağlara güvenli bir şekilde bağlanabilir.	İnternet hızında düşüş yaşanabilir; bu, şifreleme ve veri yönlendirme süreçlerinden kaynaklanabilir.

2.2. Merkeziyetsiz Depolama ve Paylaşım (Decentralized Storage and Sharing)

Merkeziyetsiz depolama, verilerin tek bir merkezi sunucu yerine çoklu otoriter düğümlerin bulunduğu bir ağ üzerinde saklandığı ve kullanıcılar arasında paylaşıldığı bir sistemdir. Bu sistemde veriler, merkezi olmayan bir ağ yapısı üzerinde birçok farklı düğümde saklanır ve bu düğümler, verilerin erişilebilir kalmasını sağlar. Merkeziyetsiz depolama, geleneksel bulut depolama sistemlerinin aksine, tek bir şirkete veya veri merkezine bağlı kalmadan veri depolama ve paylaşımını mümkün kılar [7]. Merkeziyetsiz sistemde her bir düğüm, servislerin bir alt kümesini veya belirli bir bölgeyi yönetebilir ve bunu yaparken diğer düğümlerle iletişimini devam ettirir. Bu durum sistem yükünün dengelenmesini, veri senkronizasyonunu ve ağ direncini getirir [1].

Merkez düğümü ortadan kaldırarak bir düğümün diğer birçok düğüme bağlantı sağladığı farklı bir yapı da dağıtık sistemlerdir [1]. Dağıtık sistemlerde veriler, birçok düğüm arasında eşit olarak dağıtılır. Merkezi sistemlerdeki gibi tek bir otoritenin kontrolü altında olmayan bu yapı, kullanıcıların veri depolama ihtiyaçlarını daha güvenli ve esnek bir şekilde karşılamalarını sağlar. Merkeziyetsiz depolama, aynı zamanda veri artıklığı ve yedekliliği sağlayarak veri kaybı riskini minimize eder [8]. Şekil 2, merkezi ve merkeziyetsiz ağ yapıları arasındaki farkları göstermektedir [1].

Merkeziyetsiz depolama teknolojileri, kullanıcıların veri kontrolünü artırırken merkezi sistemlerin yarattığı güvenlik ve gizlilik risklerini azaltır. Merkeziyetsiz bir uygulama, genellikle blok zinciri gibi dağıtık sistemler üzerine inşa edilir ve verilerin güvenli ve şeffaf bir şekilde işlenmesini sağlar [9].



Şekil 2. Ağ yapıları
(Figure 2. Network structures)

3. LİTERATÜR TARAMASI (LITERATURE REVIEW)

Merkeziyetsiz depolama ve veri paylaşım sistemleri ile ilgili literatürde birçok farklı çalışma mevcuttur. Bu bölümde özellikle son yıllarda yapılan çalışmalardan bazılarına yer verilmektedir. Çalışmaların çoğunun blok zinciri konusuna bağlı kaldıkları, bazılarının ise burada önerilen yapıdaki gibi IPFS teknolojisinden yararlandığı görülmektedir. Literatür taramasında Tinc VPN kullanılarak hazırlanan ve uygulama testleri için farklı senaryoların tasarlandığı bir çalışma ile karşılaşılmamıştır. Aşağıda literatürdeki çalışmaların her biri için ilgili açıklamalardan sonra bu çalışmada önerilen sistem ile karşılaştırmasına da yer verilmektedir.

Nevpurkar vd. (2020) IPFS ve Ethereum blok zinciri entegrasyonu ile merkeziyetsiz bir dosya depolama ve paylaşım sistemi geliştirmişlerdir [10]. Çalışma, IPFS'te depolanan dosyaların özet değerlerini Ethereum blok zincirinde saklayarak veri güvenliği ve bütünlüğünü sağlamayı amaçlamaktadır. Blok zinciri tabanlı bu yaklaşım, veri manipülasyonuna karşı güçlü bir koruma sağlarken, sistemdeki blok zinciri kullanımı ağ performansını ve ölçeklenebilirliği olumsuz etkileyebilir.

Önerilen sistemde ise Tinc VPN ile oluşturulan güvenli ağ sayesinde benzer bir güvenlik sağlanmakla birlikte daha düşük bir işlem maliyeti ve ağ gecikmesi avantajı sunulmaktadır.

Ghosh vd. (2023) oldukça büyük veri miktarları için erişim kontrolünü de yöneten merkeziyetsiz bir bulut depolama ve paylaşım sistemi tasarlamışlardır [11]. Çalışmada IPFS yapısı kullanılarak büyük boyuttaki multimedya dosyalarının parçaları ağ üzerindeki çeşitli düğümlerde depolanmaktadır. IPFS vekil sunucuları aracılığıyla kullanıcıların verilere erişim aşamaları için gerekli olan kimlik doğrulama adımlarında blok zinciri ağı çalıştırılmaktadır. Böylece güvenli bir paylaşım sistemi ile dosyaların kaç kişi tarafından görüntülendiği, dosyalar üzerinde bir değişiklik olup olmadığı kontrol edilebilmektedir. Önerilen sistem, Ghosh vd.nin (2023) çalışmalarındaki erişim kontrolüne dayalı güvenlik mekanizmalarına ek olarak, Tinc VPN ile sağlanan şifreli ağ yapısı sayesinde veri gizliliğini artırırken IPFS-Cluster entegrasyonu ile ölçeklenebilirliği ve ağ verimliliğini optimize etmektedir.

Peng vd.nin (2023) geliştirdikleri modelde, konsorsiyum blok zinciri tabanlı bir eşler arası dosya paylaşım sistemi önerilmiştir [12]. Bu sistem, kimlik doğrulama ve erişim kontrolü mekanizmalarıyla güvenli veri paylaşımını önceliklendirmektedir. Çalışma, özellikle çapraz organizasyonel veri paylaşımı için etkili çözümler sunsa da konsorsiyum blok zincirinin sınırlı katılımcı yapısı, ölçeklenebilirlik açısından bir dezavantaj oluşturabilir. Önerilen sistemde ise Tinc VPN ile oluşturulan şifreli ağ yapısı ve IPFS-Cluster entegrasyonu sayesinde daha geniş ölçekli ve dinamik bir cihaz ağı desteklenmektedir.

Shamdasani vd. (2023) IPFS ve blok zinciri teknolojilerini entegre ederek merkeziyetsiz bir dosya depolama sistemi oluşturmuşlardır [13]. Çalışma, dosya paylaşımında güvenliği artırmak için akıllı sözleşmelerle erişim kontrolünü entegre etmektedir. Bu sistemin avantajı, IPFS'nin dosya bölme ve kriptografik özetleme özelliklerini kullanarak yüksek güvenlik sunmasıdır. Ancak, merkezi bir yapıdan tamamen bağımsız olmaması ve IPFS'nin performans sorunları, özellikle büyük dosya işlemlerinde sınırlayıcıdır. Önerilen yöntemin Shamdasani vd.nin (2023) çalışmalarından farkı, Tinc VPN ile güvenli bir ağ kurulması ve IPFS-Cluster ile küme oluşturulmasının veri paylaşımında ek bir güvenlik ve performans avantajı sağlamasıdır.

Du vd. (2024) IPFS tabanlı sistemlerde maliyet ve güvenlik dengesini sağlamak için DWare isimli bir ara katman sunmuşlardır [14]. Çalışmada veri tekrarını önleme ve kontrol süreçleri iyileştirilmiş, ancak otomasyon çözümlerine yer verilmemiştir. Önerilen sistemde Tinc VPN ve IPFS-Cluster ile sağlanan güvenlik, DWare'in yalnızca veri tekrarını önlemeye odaklanan yapısından daha kapsamlıdır.

Lee vd. (2024) IoT cihazlarının depolama sorunlarını çözmek için IPFS ile merkezi bir veri yönetim sistemini birleştiren bir model sunmuşlardır [15]. Sistem, MQTT protokolü ve bir veritabanı ile veri organizasyonu sağlamaktadır. Öte yandan otomasyon çözümleri daha geniş kapsamda uygulanabilirlik sunmaktadır. Önerilen sistemde IPFS-Cluster altyapısı ile ağ etkileşimi optimize edilmiştir.

Zang vd.nin (2024) çalışmalarında merkeziyetsiz bir depolama sistemi tasarımı için blok zinciri ve bulut yerel teknolojiler birleştirilmiştir [16]. Bu sistem, kenar bilişim altyapılarında yüksek veri aktarım hızı ve esneklik sunar. Sistem, veri erişiminde ölçeklenebilirliği artırırken blok zinciri teknolojisini kullanarak veri bütünlüğü ve güvenilirlik sağlar. Bununla birlikte, IPFS'ye kıyasla veri aktarım performansında avantaj sağlasa da, sistemin kurulum ve işletim karmaşıklığı bir dezavantaj olarak değerlendirilebilir. Önerilen sistemde Tinc VPN kullanımı ile veri güvenliğinin artırılması, özellikle hassas verilerin korunması açısından üstünlük sağlamaktadır.

Han ve Son (2025), IPFS ve blok zinciri kullanarak belgelerin daha güvenli ve şeffaf bir şekilde yönetilmesini sağlayan bir sistem tasarlamışlardır [17]. Shamir'in Gizli Paylaşım Yöntemi ile belge yetkilendirme yapılmış, fakat otomasyon çözümlerine yer verilmemiştir. Önerilen sistem, ağ yapısı oluşturmada daha geniş bir esneklik ve kapsam sunmaktadır.

Samuel ve Kasturi (2025), bulut tabanlı bir blok zinciri ağı üzerinde derin öğrenme uygulamasıyla anahtar üretimi yaparak veri güvenliğini ve gizliliğini sağlamışlardır [18]. SpinalNet yapısının kullanımı, şifreleme süreçlerinde yüksek doğruluk ve güvenlik sağlarken sistemin karmaşık altyapısı ve yüksek işlem maliyetleri bir dezavantaj olarak öne çıkmaktadır. Önerilen sistemde ise ağ yapısının sadeliği ve düşük işlem maliyeti sayesinde daha pratik ve ölçeklenebilir bir çözüm sunulmaktadır.

Bu çalışmalar, merkeziyetsiz veri paylaşımı konusunda çeşitli yaklaşımlar sunmakta ve güvenlik, performans ile ölçeklenebilirlik gibi farklı parametreler üzerinde durmaktadır. Önerilen sistem, Tinc VPN ve IPFS-Cluster entegrasyonu ile düşük maliyetli, güvenli ve esnek bir altyapı sunarak mevcut çalışmalardan ayrılmaktadır.

4. MATERYAL VE METOT (MATERIALS AND METHODS)

Bu çalışma süresince kullanılan teknolojiler, altyapı bileşenleri ve sistemin gerçekleştirilme adımları bu bölümde detaylandırılmaktadır.

4.1. Kullanılan Teknolojiler (Technologies Used)

Bu çalışmada kullanılan yazılım ve altyapı teknolojileri aşağıda sıralanmıştır:

4.1.1. Ubuntu (Ubuntu)

Çalışmada kullanılan ana bilgisayarlar ve test için kullanılan sanal sunucuların işletim sistemi, özgür ve açık kaynaklı bir Linux dağıtımı olan Ubuntu'dur. Ubuntu, kullanıcıların ihtiyaçlarına göre özgürce geliştirilebilir ve ayarlanabilir bir işletim sistemidir. Linux çekirdeği üzerine inşa edilen Ubuntu, hem masaüstü hem de sunucu ortamlarında yaygın bir şekilde kullanılmaktadır. Altyapı, servis ve sistem çalışmalarında geniş bir uyarlanabilirlik sunar.

4.1.2. Tinc VPN (Tinc VPN)

Bu çalışmada kullanılan VPN teknolojisi, açık kaynaklı ve dağıtık ağları destekleyen Tinc VPN'dir. Tinc, 1998 yılında başlatılan ve örgü (mesh) ağ yapısını destekleyen, şifreleme ve sıkıştırma gibi özelliklere sahip bir VPN yazılımıdır. Tinc, trafiği fırsatçı bir şekilde yönlendiren ve ağın otomatik olarak yeniden yapılandırılmasına olanak tanıyan bir yapıya sahiptir [19].

Bu çalışmada Tinc'in hata toleransı ve kaynak tüketimi test edilmiştir. Bu testlerde, bir düğümün kapanması durumunda ağın çalışmaya devam ettiği gözlemlenmiştir. Ayrıca Tinc, CPU ve bellek kullanımında son derece verimli olup özellikle son kullanıcı cihazlarında düşük kaynak tüketimi ile çalışabilmektedir.

Tablo 2, farklı VPN teknolojilerinin gizlilik, veri bütünlüğü ve kimlik doğrulama açısından karşılaştırmasını göstermektedir [5].

Tablo 2. VPN teknolojilerinin kıyaslaması
(Table 2. Comparison of the VPN technologies)

VPN	Kısıtlar		
	Gizlilik	*Veri Bütünlüğü*	*Kimlik Doğrulama*
Tinc	Evet	Evet	Evet
Htun	Hayır	Hayır	Hayır
Cipe	Evet	Evet	Hayır
PPTP	Evet	Evet	Hayır
L2tpd	Hayır	Hayır	Hayır
OpenVPN	Evet	Evet	Evet
VTUN	Evet	Hayır	Hayır
Vpnd	Evet	Evet	Evet
Yavipin	Evet	Evet	Hayır

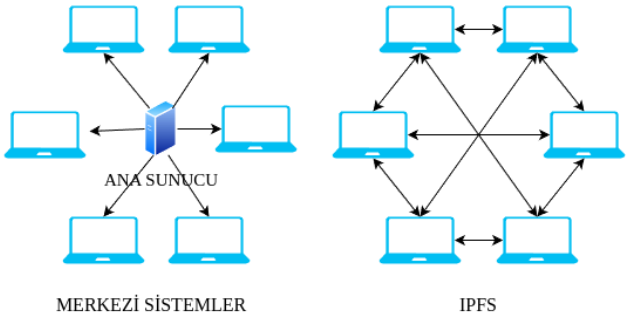
4.1.3. Gezegenler Arası Dosya Sistemi (InterPlanetary File System)

IPFS, merkeziyetsiz dosya depolama ve paylaşımı sağlayan, eşten eşe (peer-to-peer, P2P) bir ağ protokolü ve içerik adreslemeli bir dosya sistemi olarak tasarlanmıştır. 2015 yılında Juan Benet tarafından geliştirilen IPFS, Web'in mevcut HTTP tabanlı merkezi yapısından farklı olarak, daha güvenli, hızlı ve dağıtık bir yapı sunmayı hedeflemektedir [20]. IPFS, dosyaların saklanması ve paylaşılmasında merkezi sunuculara ihtiyaç duymayan, içerik adresleme yöntemini kullanan bir sistem olarak bilinir. Bu yöntem, dosyaların benzersiz bir İçerik Tanımlayıcı (CID) ile tanımlanmasını sağlar. CID, dosyanın içeriğine dayalı olarak oluşturulan kriptografik bir özet fonksiyonudur. Böylece her dosya, benzersiz bir adresle temsil edilir ve içeriğin güvenilirliğini sağlar. Uyarlanabilirliğini artırmak için çoklu aktarım/ağ protokollerini veya kriptografik karma işlevlerini destekler [21].

IPFS'nin temel çalışma prensibi, veriyi bir düğümde saklayıp, diğer düğümlerin bu veriye erişebilmesidir. Bir IPFS düğümü, başka bir düğümden dosya talep ettiğinde, dosyayı içeren blokları alır ve bu blokları birleştirerek dosyanın tam haline ulaşır. Bu yapı, geleneksel merkezi sunucuların aksine, verilerin birden fazla düğümde saklanmasına olanak tanır. Sonuç olarak, veri kaybı veya sunucu çökmesi gibi durumlar minimize edilir [22].

IPFS'nin sunduğu diğer önemli avantaj, sansüre dayanıklı olmasıdır. Geleneksel Web yapısında, içerik tek bir sunucu üzerinde barındırıldığında, bu sunucuya erişim engellendiğinde içeriğe erişim de engellenmiş olur. Ancak IPFS'de içerik, birden fazla düğümde depolandığı için bu düğümler aracılığıyla içerik erişilebilir hale gelir. Bu, özellikle internet sansürünün yoğun olduğu bölgelerde önemli bir avantaj sağlar [23].

Şekil 3'te görüldüğü gibi IPFS, diğer dosya paylaşım sistemlerinden farklı olarak merkeziyetsiz bir yapıda çalışır. Dosyaların depolanmasında ve paylaşımında, bir merkezi sunucu ya da herhangi bir hizmet sağlayıcısına ihtiyaç duymaz. Bu nedenle diğer dosya paylaşım sistemlerine göre daha güvenli ve daha özgür bir platform sunar.



Şekil 3. Veri saklama sistemlerinin karşılaştırması
(Figure 3. Comparison of the data storage systems)

IPFS aynı zamanda verilerin dağıtık bir şekilde depolanmasını sağladığı için, ölçeklenebilirlik açısından büyük avantajlar sunar. Merkezi sistemlerde verinin büyümesi, depolama kapasitesi veya sunucu maliyetlerini artırırken, IPFS'de bu maliyetler ağdaki düğümler arasında dağıtılır. Bu, IPFS'nin büyük veri setlerinin depolanması ve paylaşımı için ideal bir çözüm olmasını sağlar [24]. Tablo 3'te, IPFS'in içerik adresleme, P2P ağ yapısı, sansür direnci ve verimli veri dağıtımı gibi temel özellikleri açıklanmaktadır.

Tablo 3. IPFS'nin temel özellikleri
(Table 3. Basic features of IPFS)

Özellikler	*Açıklama*
İçerik Adresleme	IPFS, verileri sunucunun bulunduğu adresle değil, verinin kendisiyle adresler. Bu, verilerin her zaman benzersiz bir CID ile tanımlanmasını sağlar.
P2P Ağ	Veriler merkezi bir sunucuda tutulmaz, bunun yerine bir P2P ağda saklanır ve paylaşılır.
Yedeklilik ve Güvenlik	Veriler, birden fazla düğümde saklanarak yedeklenir. Bu yapı, veri kaybını önleyerek güvenliği artırır [25].
Sansür Direnci	Merkezi sunuculara bağlı olmadığı için, içerik erişimi engellenemez ve sansüre karşı dayanıklıdır.
Veri Dağıtımı	İçeriğe en yakın düğümden erişim sağlanır, bu da daha hızlı ve verimli veri dağıtımı sunar.

IPFS ile HTTP arasındaki farklılıklar Tablo 4'te yer almaktadır.

Sonuç olarak IPFS, dosya depolama ve paylaşım sistemleri için veri güvenliğini ve esnekliğini sağlayan, sansüre karşı dayanıklı, hızlı ve ölçeklenebilir bir çözüm sunar. Özellikle büyük veri setleri ve merkeziyetsizlik arayışı içinde olan projeler için önemli bir alternatif olarak öne çıkmaktadır.

Tablo 4. HTTP ve IPFS kıyaslaması
(Table 4. HTTP and IPFS comparison)

Özellikler	*HTTP*	*IPFS*
İstemci-Sunucu Modeli	Merkezi sunuculara dayanır.	Merkezi olmayan, P2P yapı.
Adresleme	Veriler sunucu adresiyle istenir.	Veriler içerik adresiyle (CID) istenir.
Erişim Güvenilirliği	Sunucu çöktüğünde verilere erişilemez.	Veriler birden çok düğümde saklanır.
Bant Genişliği Kullanımı	Sunuculara yüklenir, yavaş olabilir.	Yakındaki düğümlerden hızlı erişim.
Sunucu İhtiyacı	Barındırma sunucusu gerektirir.	Ana sunucuya gerek kalmaz.

4.1.4. Sanal Makine (Virtual Machine)

Sanal sunucu teknolojisi, bir fiziksel sunucu üzerinde birden fazla işletim sisteminin barındırılmasına olanak tanıyan bir yaklaşımdır. Bu teknoloji, sunucunun kaynaklarını etkin bir şekilde kullanarak birden fazla sanal sunucu oluşturulmasına imkân verir. Tüm sanal sunucular, tek bir fiziksel sunucu üzerinde yer almakta olup birbirlerinden bağımsız olarak çalışabilmektedir.

Bu çalışmada uygulama testlerinin gerçekleştirilmesi amacıyla, sanal sunucu merkezi olarak Linode platformu tercih edilmiştir.

4.1.5. Ansible (Ansible)

Ansible, sistemlerin kurulumu, yönetimi ve yapılandırması için kullanılan bir otomasyon dilidir. Genellikle makinelerin istenilen konfigürasyonla yapılandırılması ve yönetilmesi amacıyla kullanılır. Ansible, belirlenen görevlerin gerçekleştirilmesi ve makine konfigürasyon yönetimini birleştirerek kullanıcı dostu bir yapı sunar. Sistemler, çoğunlukla Güvenli Kabuk (SSH) protokolü üzerinden yönetilmektedir. Ansible'ın kullanımı için öncelikle bir Python ortamının kurulmuş olması gereklidir.

4.1.6. Terraform (Terraform)

Terraform, HashiCorp tarafından geliştirilen açık kaynaklı bir altyapı otomasyon aracıdır. Bu araç, kullanıcıların veri merkezi altyapısını tanımlayıp sağlamak için HashiCorp Yapılandırma Dili (HCL) veya isteğe bağlı olarak JavaScript Nesne Gösterimi (JSON) gibi bildirim temelli yapılandırma dillerini kullanmasına olanak tanır. Terraform, sanal sunucu merkezinde yeni makinelerin oluşturulmasında kullanılacak önemli bir araçtır.

4.1.7. Kabuk Betiği (Bash Script)

Uygulamadaki kodların çalıştırılması ve yapılandırılması, Kabuk Betiği ile gerçekleştirilmektedir. Kabuk Betiği, Unix tabanlı bir işletim sisteminde çalışan bir komut satırı yorumlayıcısıdır ve kullanıcıların çeşitli otomasyon görevlerini kolaylıkla gerçekleştirmesine imkân tanır.

4.1.8. JavaScript (JavaScript)

JavaScript, internet sayfalarına dinamiklik kazandırmak amacıyla yaygın olarak kullanılan bir programlama dilidir. Genellikle, Çoklu Metin İşaretleme Dili (HTML) ve Basamaklı Stil Şablonları (CSS) ile birlikte kullanılır. JavaScript, HTML ve CSS ile uyumlu çalışarak kullanıcı etkileşimini artırır ve internet sitelerinin daha dinamik ve etkileşimli hale gelmesine katkıda bulunur. Bu sayede, kullanıcı deneyimini zenginleştiren, etkileşimli ve kullanışlı web uygulamaları oluşturulmasına olanak sağlar.

4.2. Sistemin Gerçeklenmesi (Implementation of the System)

Bu çalışmada geliştirilen sistem, öncelikle makineleri özel bir ağ içine alarak, maskelenmiş IP adresleriyle IPFS'ye entegre edilmekte ve IPFS-Cluster aracılığıyla bir küme yapısı oluşturmaktadır. Altyapı, JavaScript kullanılarak bir web sayfasına aktarılmaktadır. Sistem kurulumu ve konfigürasyonu, Ansible, Terraform ve Kabuk Betikleri gibi otomasyon araçları aracılığıyla gerçekleştirilmektedir. Uygulamanın işleyişine dair detaylar Şekil 4'te sunulmaktadır.

4.2.1. Sunucu Makinelerinin Oluşturulması (Creating Server Machines)

Uygulamanın altyapısı, sanal sunucu merkezinde kiralanan makinelerde test edilmiştir. Bu testler için Linode adlı sanal sunucu sağlayıcısında Terraform kodu ve ilgili Kabuk Betikleri kullanılarak üç adet sanal makine oluşturulmuştur. Uygulamanın diğer bileşenleri Ansible ile geliştirilmiştir.

4.2.2. Envanter Dosyasının Oluşturulması (Creating the Inventory File)

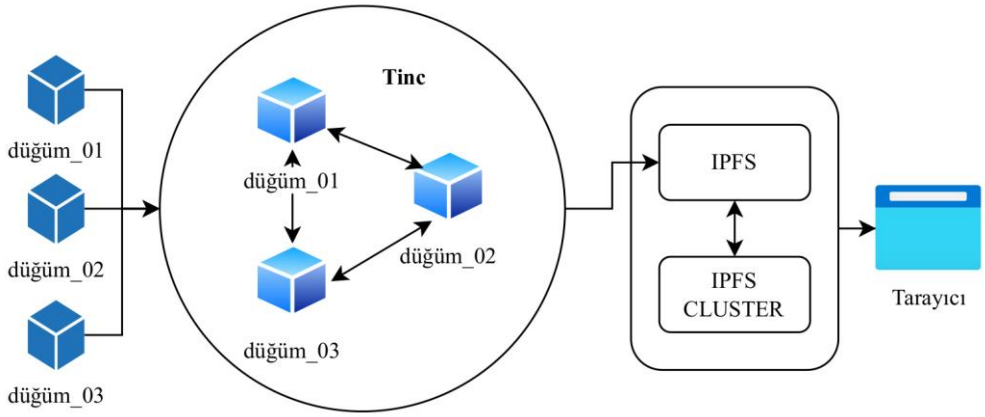
Ansible ile kurulacak makinelerin bilgilerini içeren envanter dosyası, sistemin yapılandırılması için kritik bir bileşendir. Bu dosyada makinelerin IP adresleri, kullanıcı adları ve bilgisayar adları gibi bilgiler yer almalıdır. Özel IP adresleri, yalnızca ağ içinde diğer cihazlarla veri transferinde kullanılırken, genel IP adresleri internete erişim için kullanılmaktadır. Özel IP adresleri, internete yönlendirilmediği için makinelerin iç IP adresleri olarak da adlandırılabilir.

4.2.3. Sanal Özel Ağın Oluşturulması (Creating the Virtual Private Network)

Bu aşamada Tinc VPN teknolojisi kullanılmaktadır. Tinc, tüm makinelerin birbirleriyle güvenli bir şekilde iletişim kurmasını sağlamak için bir rol aracılığıyla yapılandırılmaktadır. Güvenlik önlemleri için Kolay Ateş Duvarı (UFW) kullanılmaktadır. Böylece belirli portların açık olması sağlanarak güvenlik duvarı yönetimi basitleştirilmektedir. Ağın güvenliği için gerekli yapılandırmalar, Tinc ile sağlanmakta ve UFW üzerinden gerekli izinler verilmektedir.

4.2.4. IPFS Yapılandırması (IPFS Configuration)

Bu yapılandırmada dosya paylaşım sistemi olarak IPFS kurulumu gerçekleştirilmektedir. IPFS, Go programlama diline dayalı olarak çalıştığı için öncelikle Go'nun sistemde bulunması gerekmektedir. Daha sonra IPFS'nin yapılandırması, ilgili kullanıcı ve grup tanımlamaları ile devam etmektedir. IPFS, yüklenen dosyaların diğer makinelerde sabitlenmesini sağlamakta ve böylece merkeziyetsiz bir veri depolama çözümü sunmaktadır.



Şekil 4. Uygulama şeması
(Figure 4. Application schema

4.2.5. IPFS-Cluster Yapılandırması (IPFS-Cluster Configuration)

IPFS-Cluster, IPFS ile entegre çalışarak makineleri bir küme halinde organize eder. Bu yapı, yüklenen dosyaların ağdaki diğer tüm makinelerde sabitlenmesini sağlamaktadır. IPFS-Cluster kurulumunda, makinelerin aynı kümede olduğunu belirten anahtarlar ve diğer yapılandırma bilgileri bir değişken dosyasında tanımlanmaktadır. Bu şekilde, sistemin sürdürülebilirliği ve verimliliği artırılmaktadır.

4.2.6. IPFS-App Yapılandırması (IPFS-App Configuration)

Son olarak, oluşturulan altyapı JavaScript/Node.js ile bir web arayüzüne aktarılmaktadır. Node.js'nin açık kaynaklı yapısı, geliştirme sürecinde esneklik sağlamaktadır. Uygulama, belirli dizinlerin oluşturulması ve gerekli Node Package Manager (NPM) paketlerinin yüklenmesi ile başlatılmaktadır. Web uygulaması, kullanıcılara http://localhost:3000 adresinden erişim imkânı sunarak sistemin işleyişini kolaylaştırmaktadır.

Bu yapı, merkeziyetsiz ve özel bir sanal ağ içinde dosya paylaşım ve depolama sisteminin oluşturulmasını sağlamaktadır.

5. BULGULAR (FINDINGS)

Bu bölümde, oluşturulan VPN için gerçekleştirilen testlerin sonuçları ve sistemin işlevselliği ele alınmaktadır. Elde edilen bulgular, altyapının performansını ve kullanımını daha kapsamlı bir şekilde açıklamaktadır.

Testler için Tablo 5'teki özelliklere sahip 3 sanal makine ve 1 fiziksel makine kullanılmıştır.

Tablo 5. Test ortamındaki makinelerin teknik özellikleri
(Table 5. Technical properties of the machines in the test environment)

Özellik	Kategori	
	Fiziksel Makine	*Sanal Makine*
İşletim Sistemi	Ubuntu 20.04	Ubuntu 20.04
İşlemci (CPU)	4	1
Bellek	16 GB RAM	1 GB RAM
Depolama	512 GB SSD	25 GB
Makine Bölgesi	İstanbul, Türkiye	Frankfurt, Almanya (eu-central)
Bağlantı Hızı (Giriş/Çıkış)	33/25 Mbps	40/1 Gbps

5.1. Ağ Testleri (Network Tests)

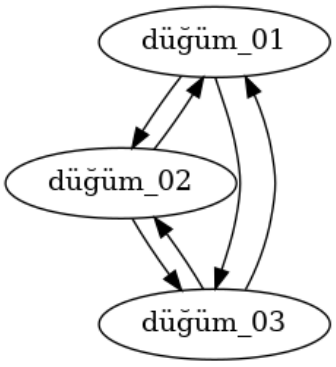
Kurulum tamamlandıktan sonra, makineler arası bağlantı ve iletişim testleri gerçekleştirilmiştir. Makinelerin portunun kapatılması, makinelerin tekrar başlatılması ya da kapatılması gibi durumlarda ağın durumu grafiklerle izlenmiştir. Bu grafiklerin oluşturulabilmesi için Tinc konfigürasyon dosyasında grafik özelliğinin aktif hale getirilmesi gerekmektedir. Tinc servisi, bu süreçte ağın topolojisini analiz ederek makineler arasındaki bağlantıları .dot formatında grafikler hâlinde üretmiştir. Bu sayede, ağın farklı durumlara nasıl tepki verdiği görsel olarak analiz edilebilmiştir.

Tablo 6'da makine port durumları ve örnek senaryolardaki koşullar belirtilmektedir. Daha sonra her bir örnek senaryo sonucunda elde edilen bulgular sunulmaktadır.

Tablo 6. Makine port durumları ve örnek senaryolar
(Table 6. Machine port states and sample scenarios)

Makineler	Örnek Senaryolar			
	Senaryo 1	*Senaryo 2*	*Senaryo 3*	*Senaryo 4*
düğüm_01	Port kapalı	Port kapalı	Port kapalı	Port açık
düğüm_02	Port açık	Port kapalı	Port açık	Port açık
düğüm_03	Port açık	Port açık	Port kapalı	Port açık

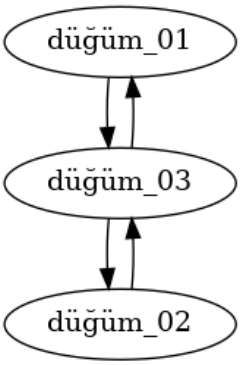
Örnek Senaryo 1: Tüm makineler yeniden başlatıldığında, ağ durumu Şekil 5'te gösterildiği gibi gözlemlenmiştir. Bu senaryo, ağın temel bağlantı durumunu ve iletişim işlevselliğini doğrulamaktadır.



Şekil 5. Örnek senaryo 1
(Figure 5. Sample scenario 1)

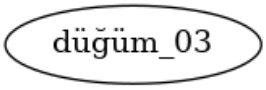
Örnek Senaryo 2: Tüm makineler yeniden başlatıldığında ağ durumu Şekil 6'da yer aldığı gibi kaydedilmiştir. Bu

durum, ağın iki düğümün kapalı olması durumundaki performansını incelemektedir.

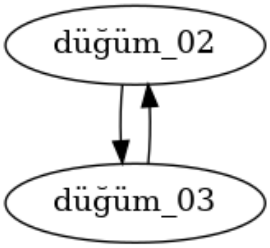


Şekil 6. Örnek senaryo 2
(Figure 6. Sample scenario 2)

Örnek Senaryo 3: düğüm_01 ve düğüm_02 kapatıldığında ağın durumu Şekil 7'deki gibi gözlemlenmiştir. Daha sonra düğüm_02 açıldığında ağ durumu Şekil 8'deki gibi kaydedilmiştir. Bu senaryo, ağın dinamik durum değiştirme kabiliyetini test etmektedir.

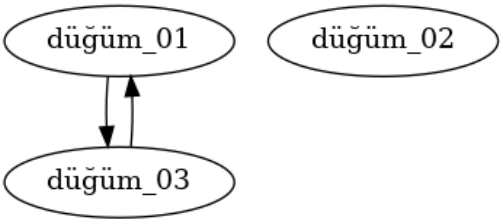


Şekil 7. Örnek senaryo 3 - 1. durum
(Figure 7. Sample scenario 3 - case 1)



Şekil 8. Örnek senaryo 3 - 2. durum
(Figure 8. Sample scenario 3 - case 2)

Örnek Senaryo 4: düğüm_02 kapalı olduğunda ve düğüm_01 ile düğüm_03 açık kaldığında ağın durumu Şekil 9'da gösterildiği gibi gözlemlenmiştir. Bu durum, tüm düğümlerin açık olduğu bir senaryoda ağın genel durumunu değerlendirmektedir.



Şekil 9. Örnek senaryo 4
(Figure 9. Sample scenario 4)

Sistemin kurulumu tamamlandıktan sonra, ağın işleyişini ve makineler arasındaki bağlantıyı doğrulamak amacıyla ağ testleri gerçekleştirilmiştir. Bu testlerde, düğüm_01 isimli makineden diğer ağ makineleri olan düğüm_02 ve düğüm_03'e ping denemesi yapılmıştır. Bu testler sayesinde makineler arasındaki bağlantının düzgün çalıştığı ve ağın işlerliğini koruduğu gözlemlenmiştir.

Şekil 10'da görüldüğü gibi düğüm_01 makinesinden düğüm_02 ve düğüm_03'e yapılan ping denemesi başarılı olmuştur. Bu sonuçlar, VPN üzerinden makinelerin birbirleriyle iletişim kurabildiğini ve verilerin sorunsuz şekilde iletilebildiğini doğrulamaktadır.

```
root@localhost:~# ping 172.16.1.2
PING 172.16.1.2 (172.16.1.2) 56(84) bytes of data.
64 bytes from 172.16.1.2: icmp_seq=1 ttl=64 time=1.23 ms
64 bytes from 172.16.1.2: icmp_seq=2 ttl=64 time=1.70 ms
64 bytes from 172.16.1.2: icmp_seq=3 ttl=64 time=9.62 ms
64 bytes from 172.16.1.2: icmp_seq=4 ttl=64 time=1.80 ms
^C
--- 172.16.1.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 1.227/3.585/9.615/3.488 ms

root@localhost:~# ping 172.16.1.3
PING 172.16.1.3 (172.16.1.3) 56(84) bytes of data.
64 bytes from 172.16.1.3: icmp_seq=1 ttl=64 time=1.38 ms
64 bytes from 172.16.1.3: icmp_seq=2 ttl=64 time=1.73 ms
64 bytes from 172.16.1.3: icmp_seq=3 ttl=64 time=1.77 ms
64 bytes from 172.16.1.3: icmp_seq=4 ttl=64 time=1.57 ms
^C
--- 172.16.1.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.384/1.614/1.768/0.152 ms
```

Şekil 10. Ağ testi - ping denemesi
(Figure 10. Network test - ping attempt)

Ping testleri sırasında elde edilen zaman değerleri, iki cihaz arasındaki veri iletiminin gecikme süresini yansıtmaktadır. Aşağıdaki analiz, düğüm_01'den düğüm_02 ve düğüm_03 makinelerine yapılan ping denemelerine dayanmaktadır.

Gecikme Süresi (Latency) İncelemesi:

Ping testleri esnasında kaydedilen gecikme süreleri şu şekildedir:

- **düğüm_02 için ping sonuçları**: 1.23 ms, 1.70 ms, 9.62 ms, 1.80 ms
- **düğüm_03 için ping sonuçları**: 1.38 ms, 1.73 ms, 1.77 ms, 1.57 ms

Bu değerler üzerinden analiz yapıldığında, düşük gecikme sürelerinin (örneğin 1-20 ms aralığı) ağın hızlı ve verimli çalıştığını, yüksek gecikme sürelerinin ise (örneğin 100 ms üzeri) ağda olası sorunlar veya yavaşlık olduğunu işaret ettiği bilinmektedir.

Gecikme Süresi Karşılaştırması:

düğüm_02'ye atılan pinglerde gecikme süresi, ortalama olarak 3 ms civarındadır. Bu sonuç, ağ bağlantısının sorunsuz ve hızlı olduğunu göstermektedir.

düğüm_03'e atılan pinglerde gecikme süresi, ortalama 1.5 ms olarak kaydedilmiştir. Bu değer, düğüm_03'ün ağ bağlantısının da istikrarlı olduğunu gösterir, ancak düğüm_02 ile kıyaslandığında biraz daha düşük bir gecikme süresi gözlemlenmiştir. Bu durum, düğüm_02'nin ağ topolojisi içinde daha uzakta yer alıyor olmasından veya daha yoğun bir yük altında olmasından kaynaklanabilmektedir.

Gecikme ve Performansın Önemi:

Gecikme sürelerinin düşük olması, dosya paylaşımı ve veri iletimi açısından sistemin performansını olumlu yönde etkiler. Özellikle IPFS gibi merkeziyetsiz dosya paylaşım sistemlerinde düşük gecikme süreleri, dosyaların daha hızlı bir şekilde erişilebilir olmasını sağlar. VPN ağı üzerinden çalışan bu sistemde, gecikmenin düşük olması güvenli veri iletimini hızlandırır ve sistem performansını artırır.

5.1.1 Yanlış Yapılandırılmış Ağ Testi (Misconfigured Network Testing)

Bu bölümde, bilerek yanlış yapılandırılmış bir düğüm eklenerek ağın güvenlik ve dayanıklılığı test edilmiştir. Temel amaç, bir cihazın ağ üzerindeki IP ve alt ağ (subnet) bilgilerinin yanlış yapılandırılması durumunda ağın bu hatalı düğüme nasıl tepki verdiğini ve veri iletişiminde nasıl bir kesinti yaşandığını incelemektir. Böylece sistemin hataya karşı dayanıklılığını ve ağın işleyişine etkisini değerlendirmek mümkün olacaktır.

Test Aşamaları:

- düğüm_02'den yapılandırma hataları içeren bir düğüme ping gönderme denemesi gerçekleştirilmiştir. Yanlış alt ağ tanımlaması ve /etc/hosts dosyasında gerekli girişlerin eksik olduğu bir senaryo oluşturulmuştur.
- Şekil 11'de test sırasında alınan çıktı verilmektedir. Bu çıktı, düğümün yanlış yapılandırılması nedeniyle veri iletişiminin gerçekleşmediğini ve "Destination Net Unknown" hatası aldığını göstermektedir.

```
root@localhost:/etc/tinc/vpn0/hosts# ping 172.16.1.5
PING 172.16.1.5 (172.16.1.5) 56(84) bytes of data.
From 172.16.1.5 icmp_seq=1 Destination Net Unknown
From 172.16.1.5 icmp_seq=2 Destination Net Unknown
From 172.16.1.5 icmp_seq=3 Destination Net Unknown
From 172.16.1.5 icmp_seq=4 Destination Net Unknown
From 172.16.1.5 icmp_seq=5 Destination Net Unknown
From 172.16.1.5 icmp_seq=6 Destination Net Unknown

--- 172.16.1.5 ping statistics ---
6 packets transmitted, 0 received, +6 errors, 100% packet loss, time 5104ms
```

Şekil 11. Test Çıktısı
(Figure 11. Test Output)

Bu test aracılığıyla yanlış yapılandırılmış bir düğümün ağda nasıl tespit edilebileceği ve veri iletişimine nasıl engel oluşturabileceği gösterilmiştir. Yanlış alt ağ veya eksik ana bilgisayarlar (hosts) dosyası gibi bilgiler nedeniyle düğümler arasında iletişim sağlanamamış ve paket kayıpları meydana gelmiştir. Bu durum, ağdaki hatalı yapılandırmaların hızla tespit edilebileceğini ve düzeltilmezse güvenlik ve veri bütünlüğü sorunlarına yol açabileceğini göstermektedir.

Bu testin güvenlik açısından önemi büyüktür. Yanlış yapılandırılmış bir düğüm, potansiyel bir güvenlik açığı oluşturabilir ve izinsiz veri erişimine veya bilgi sızıntısına neden olabilir. Örneğin, Tinc VPN ve IPFS tabanlı bir dosya sistemi üzerinde düğümlerde hatalı IP veya alt ağ bilgileri tanımlandığında, ağ üzerindeki trafiğin düzgün yönlendirilememesi, dosya transferlerinin kesintiye uğramasına veya yetkisiz düğümler tarafından ele geçirilmesine yol açabilmektedir. Ayrıca, kimlik doğrulama mekanizmalarının yanlış yapılandırılması, kötü niyetli aktörlerin ağa izinsiz erişim sağlamasına zemin hazırlayabilmektedir. Ancak ağın sağlam yapısı, bu tür hataları doğru bir şekilde tespit ederek sistemin bütünlüğünü koruma yeteneğini göstermektedir. Şifreleme protokollerinin etkin kullanımı ve ağ parametrelerinin doğru tanımlanması gibi önlemler, olası güvenlik risklerini azaltabilir. Bu deney, ağın hataya dayanıklı olduğunu, yanlış yapılandırmalardan kaynaklanan olumsuz etkilerin minimize edilebileceğini ve sistemin veri güvenliği ile operasyonel sürekliliği koruma kapasitesini kanıtlamaktadır.

5.2. Tekrarlanabilirlik (Repeatability)

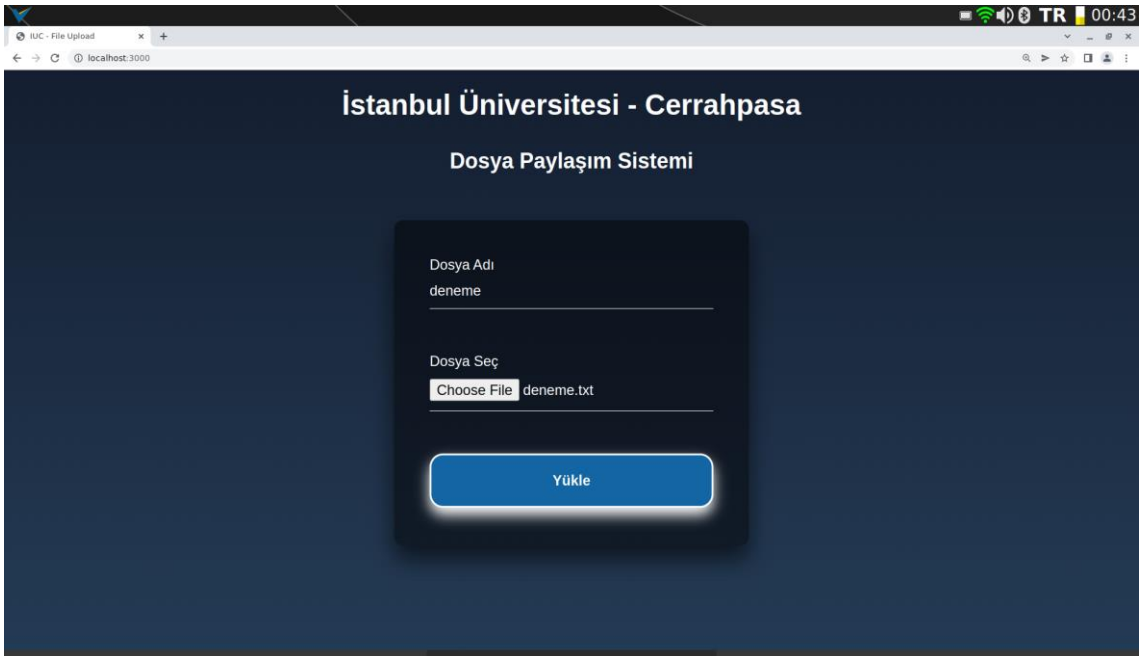
Ağ testlerinde elde edilen bulguların güvenilirliğini sağlamak için farklı senaryolar kullanılmış ve her bir senaryo birden fazla kez tekrarlanmıştır. Özellikle, Tinc VPN ve IPFS tabanlı sistemin dinamik koşullarda nasıl davrandığını anlamak amacıyla cihazların açılıp kapatılması, portların kapatılması gibi çeşitli durumlar uygulamaya dâhil edilmiştir. Bu testlerin her biri, benzer koşullarda tekrarlanarak sonuçların tutarlılığı doğrulanmıştır.

Tekrarlanabilirlik, ağın sadece bireysel testlerde değil, uzun vadede de kararlı ve güvenilir bir performans sergilediğini ortaya koymaktadır. Örneğin, iki düğümün portlarının kapatıldığı bir senaryoda, ağın fonksiyonelliği her testte aynı şekilde etkilenmiş ve yeniden yapılandırma süreçleri tutarlı sonuçlar vermiştir. Ayrıca, ağın dayanıklılığı ve otomatik toparlanma yetenekleri, farklı senaryoların tekrarıyla daha net bir şekilde gözlemlenmiştir. Bu yaklaşım, elde edilen bulguların sistemin genel davranışını temsil ettiğini ve güvenilirlik açısından güçlü bir temel oluşturduğunu göstermektedir.

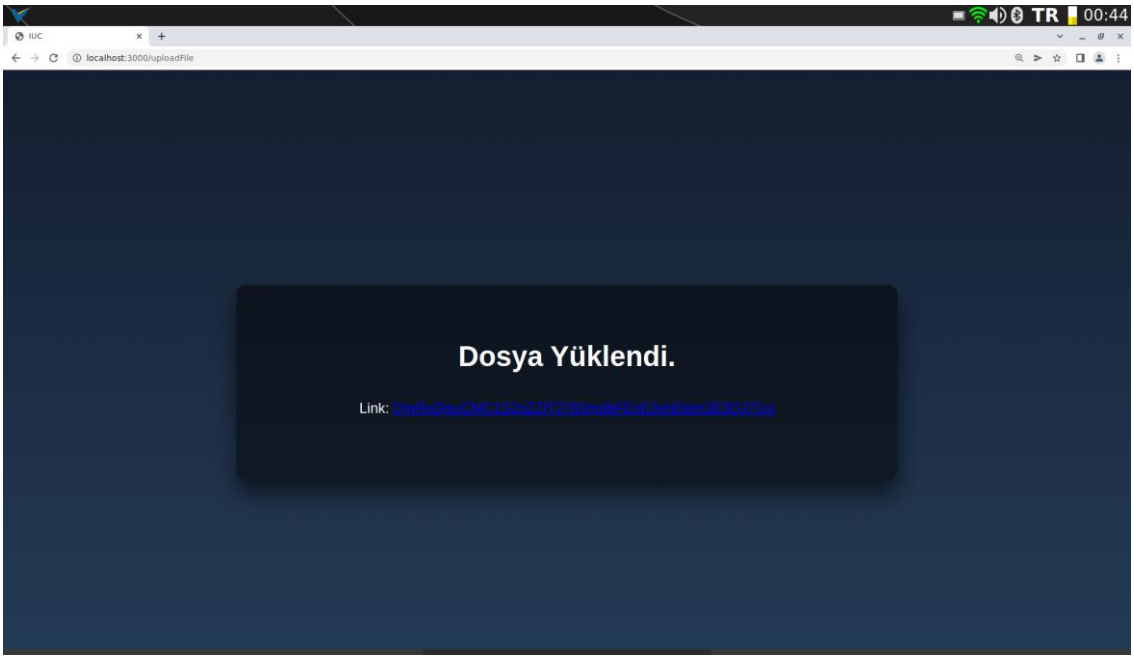
5.3. Uygulama Arayüzü (Application Interface)

Uygulama arayüzü, dosya yükleme sürecini kolaylaştırmak için kullanıcı dostu bir tasarım ile geliştirilmiştir. Şekil 12'de görülen ana sayfada kullanıcı, dosya adını belirleyip yüklemek üzere dosyasını sistemden seçebilir. "Yükle" butonuna tıklandığında, dosya güvenli ve gizli sisteme yüklenmiş olur.

Yüklenen dosyanın erişim linki, Şekil 13'te belirtilmektedir. Ağ içerisindeki diğer makineler bu bağlantıyı kullanarak yüklenen dosyaya erişebilir. Link açıldığında dosyanın içeriği kullanıcıya sunulmaktadır. Bu sistem ile farklı formatlardaki dosyaların da yüklenmesi mümkün olmaktadır. Ayrıca, dosya paylaşım sürecinin hızlandırılması ve erişimin kolaylaştırılması adına, sistemin kullanılabilirliği sürekli izlenmektedir.



Şekil 12. Uygulama arayüzü - dosya yükleme portalı
(Figure 12. Application interface - file upload portal)



Şekil 13. Uygulama arayüzü - dosya linkinin gösterilmesi
(Figure 13. Application interface - showing file link)

Şekil 14'te, yüklenen dosyanın içerik ekranı gösterilmektedir. Kullanıcı, bağlantıyı açarak dosyanın içeriğini doğrudan görüntüleyebilir. Bu ekran, dosyanın adını, türünü ve içeriğini açık bir şekilde sunarak kullanıcıların dosya hakkında hızlı bir değerlendirme yapmalarına olanak tanımaktadır. Uygulamanın bu özelliği, dosya paylaşım sürecini daha şeffaf ve kullanıcı dostu hale getirerek kullanıcıların dosyalarına kolayca erişimini sağlamaktadır.

Şekil 14. Uygulama arayüzü - dosyanın açılması
(Figure 14. Application interface - opening the file)

Yapılan testler sonucunda, dosya yükleme işlemi tamamlandığında, IPFS ağına bağlı her bir düğümün dosyayı başarıyla sabitlediği (pinned) gözlemlenmiştir. Yukarıdaki terminal çıktısında, üç farklı düğüm üzerinde dosyanın sabitlenme süreci ayrıntılı olarak verilmektedir (Şekil 15). Bu, verinin ağ üzerinde kalıcılığını ve erişilebilirliğini sağlayan önemli bir adımı temsil etmektedir.

```
root@localhost:~# ipfs-cluster-ctl add deneme.txt
added
QmRvDjsuCMC1S2oZZfT2783mdbFEnE3shEktm3E3QJ71gj
deneme.txt

root@localhost:~# ipfs-cluster-ctl status
QmRvDjsuCMC1S2oZZfT2783mdbFEnE3shEktm3E3QJ71gj
> düğüm_03      : PINNED | 2024-10-11T20:24:35Z | Attempts: 0 |
Priority: false
> düğüm_02      : PINNED | 2024-10-11T20:24:35Z | Attempts: 0 |
Priority: false
> düğüm_01      : PINNED | 2024-10-11T20:24:35Z | Attempts: 0 |
Priority: false

root@localhost:~# ipfs cat
QmU6iYxgJbZqQ8MAD5aqXwC1tbnoC8hGUDgsnbabcb5G8q
Bu bir deneme dosyasıdır.
```

Şekil 15. Terminal çıktısı - dosyanın paylaşımı
(Figure 15. Terminal output - sharing the file)

5.4. Ek Bulgular (Additional Findings)

Bu sistemin sağladığı güvenlik ve erişim kolaylıkları, kullanıcıların dosyalarını güvenli bir ortamda paylaşmalarını teşvik etmektedir.

Uygulamanın geliştirilmesi sırasında elde edilen kullanıcı geri bildirimleri, arayüzün kullanılabilirliğini artırmak için önemli bir kaynak olmuştur. Kullanıcılar tarafından sağlanan başlıca geri bildirimler şunlardır:

- **Arayüz sadeliği:** Kullanıcılar, arayüzün basit olduğunu belirtmiş, ancak dosya yükleme sırasında belirsizlik yaşadıklarını ifade etmişlerdir. Bu doğrultuda, arayüz sadeleştirilmiş ve geçişler iyileştirilerek kullanıcı deneyimi geliştirilmiştir.
- **Erişim linkinin görünürlüğü:** İlk sürümlerde erişim linkinin yerleşimi kullanıcılar tarafından yeterince belirgin bulunmamıştır. Bu nedenle, linkin daha görünür bir şekilde tasarlanması sağlanmıştır.

Test süreçleri sırasında, kullanıcıların sistemle olan etkileşimlerinin analizi, uygulamanın sürekli geliştirilmesine olanak sağlamıştır. Ayrıca, sistemin güvenlik özellikleri incelendiğinde, şifreleme ve veri bütünlüğü sağlama yöntemlerinin etkili olduğu gözlemlenmiştir. Kullanıcıların, dosyalarını güvenli bir şekilde paylaşarak merkezi sistemlerin risklerinden uzak durmaları teşvik edilmiştir.

Böylece bu çalışma kapsamında sunulan merkeziyetsiz dosya paylaşım sistemi, Tinc VPN ve IPFS altyapılarını kullanarak güvenli ve erişilebilir bir çözüm sunmaktadır.

6. TARTIŞMA (DISCUSSION)

Bu çalışma, kurum veya kuruluşlar arasında dosya paylaşımının güvenli ve gizli bir ağ üzerinden gerçekleştirilmesine olanak tanıyan bir sistemin tasarımını açıklamaktadır. Çeşitli teknolojilerin entegrasyonu ile oluşturulan bu uygulama, iş yönetimi ve akışı için güvenli ve merkeziyetsiz bir altyapı sunmaktadır. Böylece, sistemin işleyişi, farklı ülkelerde kullanılabilir hale getirilmiştir.

Önerilen altyapıda, öncelikle farklı ağlardaki tüm bilgisayarlar Tinc ile güvenli ve şifrelenmiş bir sanal ağa dâhil edilmiştir. Farklı ağlardaki makineler, Tinc ile yapılandırılmış ve ağlar maskelendikten sonra oluşturulan VPN ağına entegre edilmiştir. Daha sonra, IPFS konfigürasyonu ile bu VPN ağındaki cihazlar bir küme haline getirilmiş ve aralarında dosya paylaşımı gerçekleştirilmiştir. Ağda bulunan herhangi bir cihazın yüklediği dosya, şifrelenmiş biçimde diğer cihazlar tarafından özet bilgisiyle görüntülenip kaydedilebilmektedir. Sistemin testleri, sanal sunucu merkezinden sağlanan makinelerle gerçekleştirilmiştir. Uygulamanın sanal makinelerinin oluşturulmasında Terraform, sistemin kurulum kodu için Ansible, ortam ve sistemin yapılandırılmasında ise Kabuk Betikleri kullanılmıştır. Uygulamanın web arayüzüne aktarımı ise JavaScript ile gerçekleştirilmiştir. IPFS ve Tinc teknolojileriyle oluşturulan dosya paylaşım sisteminin birçok avantajı bulunmaktadır:

- **Merkezi Olmayan Yapı:** IPFS ve Tinc, merkeziyetsiz bir yapıda çalıştığından, dosyaların depolanması ve paylaşımında merkezi bir sunucuya ihtiyaç duyulmamaktadır. Bu durum, dosya paylaşım sisteminin güvenliğini artırmaktadır.

- **Güvenlik:** IPFS, dosyaların özet tabanlı adresleme sistemi kullanarak orijinalliğini korumakta ve dosyaların değiştirilmesini zorlaştırmaktadır. Tinc, güvenli ve şifreli bir VPN protokolü olarak iletilen verilerin güvenliğini sağlamaktadır.
- **Ölçeklenebilirlik:** Hem IPFS hem de Tinc, ölçeklenebilir yapıları sayesinde dosyaların depolanması ve paylaşımında genişleme imkânı sunmaktadır.
- **Hız:** IPFS, dosyaların bloklar halinde depolanmasını sağlayarak hızlı bir paylaşım imkânı sunmakta; Tinc ise hızlı bir VPN bağlantısı temin etmektedir.
- **Uygun Maliyet:** IPFS ve Tinc, açık kaynak kodlu ve ücretsiz olarak sunuldukları için dosya paylaşım sisteminin maliyet etkin bir şekilde kurulup yönetilmesine olanak tanımaktadır.

Ayrıca, sistemin öne çıkan özellikleri arasında dayanıklılık, yüksek hata toleransı, güvenli ve özel bir ağda çalışabilme yeteneği, güncellemelerin sistemin işleyişini aksatmadan yapılabilmesi, verilerin yedekleme ihtiyacının ortadan kalkması, arıza durumunda sistemin işlevselliğini yitirmemesi ve her makinenin hem tedarikçi hem de tüketici olabilmesi sayılabilir. Bununla birlikte, çalışmanın bazı kısıtları aşağıdaki gibi özetlenebilir:

- **Veri Erişimi Hızı:** IPFS, dosyaların yakın düğümlerden alınmasını teşvik etse de uzak düğümlerden veri alımında yavaşlık yaşanabilir. Özellikle büyük dosyaların paylaşımı ve indirilmesi durumunda hızlı erişim sağlamak zor olabilir.
- **Veri Bütünlüğü Sorunları:** IPFS, dosya bütünlüğünü sağlamak için kriptografik özet fonksiyonları kullanmasına rağmen, bu özet fonksiyonların kaybolmaması ve verilerin doğru bir şekilde saklanması gereklidir. Dosyaların kaybolması veya değiştirilmesi durumunda bütünlük bozulabilir [26].
- **Bağlantı ve Performans:** Tinc, bir VPN çözümü olarak bazı bağlantı ve performans sınırlamalarına sahip olabilir. Özellikle büyük veri miktarlarının iletimi performans sorunlarına yol açabilir.
- **Kullanım Zorlukları:** Tinc, diğer VPN çözümlerine kıyasla daha karmaşık bir yapıya sahip olabilir. Bu da ayarların yönetilmesi, kurulumu ve yapılandırması için daha fazla teknik bilgi gerektirebilir.

7. SONUÇ VE ÖNERİLER (CONCLUSION AND RECOMMENDATIONS)

Sonuç olarak, bu çalışma, merkeziyetsiz ve P2P mimarisiyle, makinelerin bir veya birkaçının devre dışı kalması durumunda bile işlevselliğini sürdüren bir dosya saklama ve paylaşım sistemi sunmakta ve verilerin paylaşımını daha güvenli ve verimli bir şekilde gerçekleştirmektedir.

Gelecek çalışmalar açısından sistemin daha geniş bir perspektifte değerlendirilebilmesi için şu alanlarda ileri araştırmalar yapılması önerilmektedir:

- **Performans Optimizasyonu:** Farklı ağ yükleri ve kullanıcı yoğunlukları altında sistemin yanıt süreleri ve işlem kapasiteleri detaylı olarak analiz edilmelidir. Özellikle yüksek trafiğe maruz kalan düğümlerin veri işleme hızlarının artırılması için yeni algoritmalar ve protokoller geliştirilebilir.
- **Mobil Entegrasyonu:** Sistem, mobil cihazlarda kullanılabilirlik açısından test edilmelidir. Bu entegrasyon, merkeziyetsiz dosya paylaşımının daha fazla kullanım alanına ulaşmasını sağlayabilir.
- **Çoklu Dil Desteği ve Kullanıcı Deneyimi:** Sistem arayüzü, farklı dilleri destekleyecek şekilde genişletilmelidir. Bu, uluslararası kullanıcıların sisteme erişimini kolaylaştıracaktır. Ayrıca, kullanıcı deneyimini artırmak için farklı yaş ve beceri seviyelerine uygun arayüz tasarımları oluşturulabilir.
- **Akademik ve Endüstriyel Testler:** Sistem, daha geniş bir kullanıcı kitlesi ile gerçek dünyadaki senaryolar altında test edilmelidir. Örneğin, akademik araştırmalarda büyük veri setlerinin paylaşımı veya endüstriyel projelerde hassas veri transferleri gibi senaryolar değerlendirilebilir.
- **Yapay Zekâ Destekli Veri Yönetimi:** Sistemin verimliliğini artırmak adına, yüklenen dosyaların sınıflandırılması ve erişim istatistiklerinin analiz edilmesi için yapay zeka algoritmaları entegre edilebilir. Bu, özellikle ağ üzerindeki yoğunluğun optimize edilmesine ve kullanıcı ihtiyaçlarının daha iyi karşılanmasına olanak tanıyacaktır.
- **Yasal ve Düzenleyici Uyum:** Merkeziyetsiz dosya paylaşımının yasal yönleri araştırılmalı ve farklı ülkelerdeki düzenlemelerle uyumlu bir sistem tasarımı yapılmalıdır. Böylece sistemin daha geniş bir kullanıcı tabanı tarafından benimsenmesi kolaylaşacaktır.

Bu öneriler, sistemin hem teknik hem de pratik açıdan daha işlevsel ve kapsamlı bir çözüm haline gelmesine katkı sağlayacaktır.

KAYNAKLAR (REFERENCES)

[1] M. M. Merlec, H. P. In, "Blockchain-Based Decentralized Storage Systems for Sustainable Data Self-Sovereignty: A Comparative Study", *Sustainability*, 16(17), 7671, 2024.

[2] C. Li, M. Xu, J. Zhang, H. Guo, X. Cheng, "SoK: Decentralized storage network", *High-Confidence Computing*, 4(3), 100239, 2024.

[3] H. R. Hasan, K. Salah, R. Jayaraman, M. Omar, I. Yaqoob, S. Pesic, T. Taylor, D. Boscovic, "A Blockchain-Based Approach for the Creation of Digital Twins", IEEE Access, 8, 34113–34126, 2020.

[4] Z. Fauziah, N. P. Anggraini, Y. P. A. Sanjaya, T. Ramadhan, "Enhancing Cybersecurity Information Sharing: A Secure and Decentralized Approach with Four-Node IPFS", *International Journal of Cyber and IT Service Management*, 3(2), 153–159, 2023.

[5] S. Khanvilkar, A. Khokhar, "Virtual Private Networks: An Overview with Performance Evaluation", IEEE Communications Magazine, 42(10), 146–154, 2004.

[6] L. Ibrahim, "Virtual Private Network (VPN) Management and IPSec Tunneling Technology", *Multi-Knowledge Electronic Comprehensive Journal for Education and Science Publications* (MECSJ), (1), 2017.

[7] T. Sabrina, A. Widjajarto, A. Budiyono, "Memory Analysis for IPFS Implementation on Ethereum Smart Contract", **International Conference on Rural Development and Entrepreneurship 2019: Enhancing Small Business and Rural Development Toward Industrial Revolution** 4.0, Yogyakarta, Endonezya, 1333–1343, 20-21 Ağustos, 2019.

[8] Internet: Cryptopedia Staff, An Overview of Decentralized Cloud Storage Services, https://www.gemini.com/cryptopedia/crypto-cloud-storage-decentralized-cloud-storage-providers, 18.12.2024.

[9] W. Cai, Z. Wang, J. B. Ernst, Z. Hong, C. Feng, V. C. M. Leung, "Decentralized Applications: The Blockchain-Empowered Software System", IEEE Access, 6, 53019–53033, 2018.

[10] M. Nevpurkar, C. Bandgar, R. Deshmukh, J. Thombre, R. Sadafule, S. Bhat, "Decentralized File Storing and Sharing System Using Blockchain and IPFS", *International Research Journal of Engineering and Technology* (IRJET), 7(5), 560-563, 2020.

[11] R. Ghosh, P. Yadav, Z. Khan, L. Panika, "Decentralized File Storing and Sharing System", *International Research Journal of Modernization in Engineering Technology and Science*, 5(5), 5847-5856, 2023.

[12] S. Peng, W. Bao, H. Liu, X. Xiao, J. Shang, L. Han, S. Wang, X. Xie, Y. Xu, "A Peer-to-Peer File Storage and Sharing System Based on Consortium Blockchain", Future Generation Computer Systems, 141, 197-204, 2023.

[13] N. Wadile, J. Shamdasani, S. Deshmukh, M. Sayyed, S. Khandare, "Decentralized File Storage (Interplanetary File System) Using Blockchain", *International Journal of Engineering Research & Technology* (IJERT), 12(3), 252-256, 2023.

[14] Y. Du, A. Zhou, C. Wang, "DWare: Cost-Efficient Decentralized Storage With Adaptive Middleware", IEEE Transactions on Information Forensics and Security, 19, 8529-8543, 2024.

[15] C. Lee, J. Kim, H. Ko, B. Yoo, "Addressing IoT Storage Constraints: A Hybrid Architecture for Decentralized Data Storage and Centralized Management", Internet of Things, 25, 101014, 2024.

[16] H. Zang, H. Kim, J. Kim, "Blockchain-Based Decentralized Storage Design for Data Confidence Over Cloud-Native Edge Infrastructure", IEEE Access, 12, 50083-50099, 2024.

[17] J. Han, Y. Son, "Design and Implementation of a Decentralized Document Management System", Expert Systems with Applications, 262, 125516, 2025.

[18] S. B, K. K, "A Novel Secure Privacy-Preserving Data Sharing Model With Deep-Based Key Generation on the Blockchain Network in the Cloud", Computer Standards & Interfaces, 92, 103932, 2025.

[19] A. A. Pardina, **Comparison of virtual networks solutions for community clouds, Lisans Tezi**, KTH Royal Institute of Technology, Stockholm, İsveç, 2014.

[20] Internet: J. Benet, IPFS - Content Addressed, Versioned, P2P File System, https://arxiv.org/pdf/1407.3561, 16.04.2025.

[21] N. Sangeeta, S. Y. Nam, "Blockchain and Interplanetary File System (IPFS)-Based Data Storage System for Vehicular Networks with Keyword Search Capability", Electronics, 12(7), 1545, 2023.

[22] E. Daniel, F. Tschorsch, "IPFS and Friends: A Qualitative Comparison of Next Generation Peer-to-Peer Data Networks", IEEE Communications Surveys & Tutorials, 24(1), 31–52, 2022.

[23] Internet: zK Capital, IPFS: A Complete Analysis of the Distributed Web, https://medium.com/hackernoon/ipfs-a-complete-analysis-of-the-distributed-web-6465ff029b9b, 18.12.2024.

[24] R. Kumar, R. Tripathi, "Implementation of Distributed File Storage and Access Framework using IPFS and Blockchain", **2019 Fifth International Conference on Image Information Processing** (ICIIP), Shimla, Hindistan, 246–251, 15–17 Kasım, 2019.

[25] R. R. Stewart, Q. Xie, Stream Control Transmission Protocol (SCTP): **A Reference Guide, Addison-Wesley,** A.B.D., 2001.

[26] B. Guidi, M. Conti, A. Passarella, L. Ricci, "Managing Social Contents in Decentralized Online Social Networks: A Survey", *Online Social Networks and Media*, 7, 12–29, 2018.