



**T.C.**

**BARTIN ÜNİVERSİTESİ**

**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**AKILLI SİSTEMLER MÜHENDİSLİĞİ ANABİLİM DALI**

**YÜKSEK LİSANS TEZİ**

**YAZILIM TANIMLI ÇOKLU AĞLARDA YÖNLENDİRİLMİŞ**

**DÖNGÜSÜZ GRAF MODELLİ DİJİTAL CÜZDAN**

**TEKNOLOJİLERİ İLE SERVİS KALİTESİ DESTEKLİ**

**YÖNLENDİRME MİMARİSİ**

**CEREN BUSE YILDIRIM**

**DANIŞMAN**

**DR. ÖĞR. ÜYESİ EVRİM GÜLER**

**BARTIN-2023**



**T.C.**

**BARTIN ÜNİVERSİTESİ**

**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**AKILLI SİSTEMLER MÜHENDİSLİĞİ ANABİLİM DALI**

**YAZILIM TANIMLI ÇOKLU AĞLARDA YÖNLENDİRİLMİŞ DÖNGÜSÜZ  
GRAF MODELLİ DİJİTAL CÜZDAN TEKNOLOJİLERİ İLE SERVİS KALİTESİ  
DESTEKLİ YÖNLENDİRME MİMARİSİ**

**YÜKSEK LİSANS TEZİ**

**Ceren Buse YILDIRIM**

**BARTIN-2023**

## BEYANNAME

Bartın Üniversitesi Lisansüstü Eğitim Enstitüsü tez yazım kılavuzuna göre Dr. Öğr. Üyesi Evrim GÜLER danışmanlığında hazırlamış olduğum “YAZILIM TANIMLI ÇOKLU AĞLARDA YÖNLENDİRİLMİŞ DÖNGÜSÜZ GRAF MODELLİ DİJİTAL CÜZDAN TEKNOLOJİLERİ İLE SERVİS KALİTESİ DESTEKLİ YÖNLENDİRME MİMARİSİ” başlıklı yüksek lisans tezimin bilimsel etik değerlere ve kurallara uygun, özgün bir çalışma olduğunu, aksinin tespit edilmesi halinde her türlü yasal yaptırımını kabul edeceğimi beyan ederim.

28.08.2023

Ceren Buse YILDIRIM



## ÖNSÖZ

Bu süreçte desteklerini ve yardımlarını hiç esirgemeyen danışman hocam Sayın Dr. Öğr. Üyesi Evrim GÜLER'e teşekkürlerimi sunarım.

Ceren Buse YILDIRIM

## ÖZET

**Yüksek Lisans Tezi**

# **YAZILIM TANIMLI ÇOKLU AĞLARDA YÖNLENDİRİLMİŞ DÖNGÜSÜZ GRAF MODELLİ DİJİTAL CÜZDAN TEKNOLOJİLERİ İLE SERVİS KALİTESİ DESTEKLİ YÖNLENDİRME MİMARİSİ**

**Ceren Buse YILDIRIM**

**Bartın Üniversitesi**

**Lisansüstü Eğitim Enstitüsü**

**Akıllı Sistemler Mühendisliği Anabilim Dalı**

**Tez Danışmanı: Dr. Öğr. Üyesi Evrim GÜLER**

**Bartın-2023, sayfa: 44**

Bilgisayar ağı, en az iki cihazın birbiri ile bağlanarak oluşturduğu iletişim bağıdır. Bu iletişim küçük bir alan içerisinde, kilometrelerce uzak cihazlar arasında kablolu-kablosuz olarak sağlanabilir. Küreselleşmenin artması ve günümüz teknolojisinin hızla gelişmesi birbiri ile iletişim halinde olan cihaz sayısını ve ağ topolojilerinin ölçeklerini oldukça arttırmıştır. Ağ topolojilerinin büyümesi sonucunda merkezi yönetimli ağlar yerine yazılım tanımlı bulut mimarilerinin de kullanılmasıyla dağıtık ağ modelleri yaygınlaşmaya başlamıştır. Çoklu ağ yapısı modellerinde, ağlar arasında kurulan iletişimde yaşanan gizlilik, güvenlik ve iletişim kalitesi ile ilgili olarak uçtan uca verilerin aktarımı sırasında bazı kalite sorunları oluşabilmektedir. Bununla ilgili olarak, verilerin bir kaynak noktadan hedef noktaya iletim aşamasında hizmet kalitesi standartları olarak kullanılan bant genişliği, gecikme, güvenilirlik gibi bazı önemli iletişim metriklerinin belirlenmesi ve kontrol edilebilmesi önem arz etmektedir. Bu nedenle, ağ topolojilerindeki verilerin dağıtık dijital cüzdan teknolojilerine dayalı bir şekilde erişilebilir ve güncellenebilir olması, sistem kalitesi, esnekliği, hızı ve güvenliği gibi konuları olumlu yönde etkilemektedir. Bu araştırmada, dağıtık dijital cüzdan teknolojilerinden bir tanesi olarak altyapısını oluşturduğumuz Yönlendirilmiş Döngüsüz Graf (YDG) modeli tabanlı yazılım tanımlı

çoklu ağ mimarilerinde uçtan uca servis kalitesi desteklenerek yol hesaplama mimarisinin oluşturulması planlanmaktadır. YDG'nin verileri tek yönde hareket eden graf formatında düğümlerde taşınması, geçmiş ve gelecekteki işlemleri doğrulayamayacak bir formata sahip olması ölçeklenebilirliği artırır. Önerilen mimari, YDG sayesinde, merkezi araçları ortadan kaldırarak, paralel ve ölçeklenebilir yol hesaplamasını sağlayacaktır. YDG mimarisinde bulunan düğümlerin her biri, işlem ve veri girişini, defter bütünlüğünü ve güvenliğini kriptografik teknikler kullanarak sağlayacaktır. Ayrıca yapının kontrol, veri, uygulama düzleminin birbirinden ayrı olduğu yazılım tanımlı ağ modeli ile desteklenmesi sistemin esnekliğini, ölçeklenebilirliğini olumlu yönde etkiler. Oluşturulacak YDG tabanlı mimari tedarik zinciri yönetimi, Nesnelerin İnterneti (IoT) ve merkezi olmayan finans platformları, vb. uygulamalarda eş zamanlı olarak işlemlerin yapılmasına olanak sağlayacaktır.

**Anahtar Kelimeler:** Dağıtık dijital cüzdan, yönlendirilmiş döngüsüz graf, yazılım tanımlı ağ, hizmet kalitesi

## **ABSTRACT**

**M. Sc. Thesis**

### **QUALITY OF SERVICE ASSISTED ROUTING ARCHITECTURE WITH DIGITAL WALLET TECHNOLOGIES USING A DIRECTED ACYCLIC GRAPH MODEL ROUTING IN SOFTWARE-DEFINED MULTI-DOMAIN NETWORKS**

**Ceren Buse YILDIRIM**

**Bartın University**

**Graduate School**

**Department of Intelligent Systems Engineering**

**Thesis Advisor: Asst. Prof. Dr. Evrim GÜLER**

**Bartın-2023, pp: 44**

A computer network is a communication link created by connecting at least two devices with each other. This communication can be provided in a small area, between kilometers away devices, as wired or wireless. The increase in globalization and the rapid development of today's technology have increased the number of devices in communication with each other and the scales of network topologies. As a result of the growth of network topologies, distributed network models have become widespread with the use of software-defined cloud architectures instead of centrally managed networks. In multi-network models, some quality problems may occur during the transfer of end-to-end data regarding the privacy, security, and communication quality experienced in communication between networks. In this regard, it is important to determine and control some important communication metrics, such as bandwidth, delay, and reliability, which are used as service quality standards during data transmission from a source to a destination point. Therefore, the fact that data in network topologies can be accessed and updated based on distributed digital wallet technologies positively affects issues such as system quality, flexibility, speed, and security. In this research, it is planned to create a

path computing architecture by supporting end-to-end service quality in software-defined multi-network architectures based on the Directed Acyclic Graph (DAG) model, which is one of the distributed digital wallet technologies. The fact that DAG carries the data in the nodes in a graph format that moves in one direction and has a format that cannot verify past and future transactions increases scalability. Thanks to DAG, the proposed architecture will enable parallel and scalable path computation by eliminating central intermediaries. Each node in the DAG architecture will provide transaction and data entry, ledger integrity, and security using cryptographic techniques. In addition, the support of the structure with the Software Description Network model, where the control, data, and application planes are separate from each other, positively affects the flexibility and scalability of the system. The DAG-based architecture to be created will provide supply chain management, Internet of Things (IoT) and decentralized finance platforms, etc. It will allow simultaneous transactions in applications.

**Keywords:** Distributed digital ledger, directed acyclic graph, software defined networking, quality of service

## İÇİNDEKİLER

|                                                                                                          |      |
|----------------------------------------------------------------------------------------------------------|------|
| KABUL VE ONAY.....                                                                                       | iii  |
| BEYANNAME .....                                                                                          | iv   |
| ÖNSÖZ .....                                                                                              | v    |
| ÖZET .....                                                                                               | vi   |
| ABSTRACT .....                                                                                           | vii  |
| İÇİNDEKİLER.....                                                                                         | x    |
| ŞEKİLLER DİZİNİ.....                                                                                     | xii  |
| SİMGELER VE KISALTMALAR DİZİNİ.....                                                                      | xiii |
| 1. GİRİŞ.....                                                                                            | 1    |
| 1.1. Mimari .....                                                                                        | 5    |
| 1.1.1. Yazılım Tanımlı Ağ.....                                                                           | 5    |
| 1.1.1.1. Kontrol Katmanı .....                                                                           | 10   |
| 1.1.1.2. Veri Katmanı .....                                                                              | 10   |
| 1.1.1.3.Uygulama Katmanı .....                                                                           | 10   |
| 1.1.2. Yönlendirilmiş Döngüsüz Graf.....                                                                 | 11   |
| 1.1.2.1.Yönlendirilmiş Döngüsüz Graf Uygulamaları.....                                                   | 11   |
| 1.1.2.2. Yönlendirilmiş Döngüsüz Grafları Diğer Dağıtık Dijital<br>Cüzdanlardan Ayıran Özellikleri ..... | 13   |
| 1.1.2.3. Yönlendirilmiş Döngüsüz Graflar'ın Kullanıldığı Alanlar .....                                   | 14   |
| 1.1.3. Tangle.....                                                                                       | 15   |
| 1.1.4. Ağ İletişimi Hizmet Kalitesi .....                                                                | 16   |
| 1.1.4.1. Bant Genişliği.....                                                                             | 16   |
| 1.1.4.2. Gecikme.....                                                                                    | 17   |
| 1.1.4.3. Dalgalanma .....                                                                                | 18   |
| 1.1.4.4. Kayıp.....                                                                                      | 18   |
| 2. LİTERATÜR ÖZETİ.....                                                                                  | 19   |
| 3. MATERYAL VE METOT .....                                                                               | 23   |
| 3.1. Sistem Modeli .....                                                                                 | 27   |
| 4. BULGULAR VE TARTIŞMA .....                                                                            | 32   |
| 4.1. Bulgular .....                                                                                      | 32   |
| 4.1.1.Hizmet Talebinde Bulunulması .....                                                                 | 32   |

|                            |                                  |
|----------------------------|----------------------------------|
| 4.1.2.Hizmet Yanıtı: ..... | 38                               |
| 4.2.Tartışma.....          | 40                               |
| 5. SONUÇ VE ÖNERİLER ..... | 41                               |
| KAYNAKLAR.....             | 42                               |
| ÖZGEÇMİŞ .....             | Hata! Yer işareti tanımlanmamış. |



## ŞEKİLLER DİZİNİ

| Şekil                                                                         | Sayfa |
|-------------------------------------------------------------------------------|-------|
| No                                                                            | No    |
| Şekil 1. YDG Tabanlı Dijital Defter Teknoloji Mimarisi .....                  | 4     |
| Şekil 2. Yazılım Tanımlı Ağ Mimarisi.....                                     | 5     |
| Şekil 3. Yazılım Tanımlı Ağ Topolojisi – Adım 1 .....                         | 7     |
| Şekil 4. Yazılım Tanımlı Ağ Topolojisi – Adım 2.....                          | 8     |
| Şekil 5. Yazılım Tanımlı Ağ Topolojisi – Adım 3.....                          | 9     |
| Şekil 6. Yazılım Tanımlı Ağ Topolojisi – Adım 4.....                          | 9     |
| Şekil 7. Yazılım Tanımlı Ağ Topolojisi – Adım 5.....                          | 10    |
| Şekil 8. YDG – Tangle Ağ Mimarisi .....                                       | 16    |
| Şekil 9. Tangle Alt Birimleri.....                                            | 27    |
| Şekil 10. Çoklu Alan Ağları Mimarisi .....                                    | 28    |
| Şekil 11. Her bir Düğüm İçinde Tutulacak Bilgiler .....                       | 29    |
| Şekil 12. YDG Tabanlı Çoklu Alan Ağlarında Uçtan Uca Yol Bulma Mimarisi ..... | 30    |
| Şekil 13. DAGNode mimarisi başlangıç yapısı .....                             | 32    |
| Şekil 14. DAGNode modülü fonksiyonları.....                                   | 33    |
| Şekil 15. DAGServiceRequestExample kod bloğu.....                             | 35    |
| Şekil 16. YDG üzerinde kısıtlamalı Dijkstra algoritmasının mimarisi .....     | 37    |
| Şekil 17. Servis Cevap Modülünün Oluşturulması .....                          | 39    |

## SİMGELER VE KISALTMALAR DİZİNİ

|      |                                                   |
|------|---------------------------------------------------|
| Gbps | : Ağ bağlantısından 1 saniyede geçen 1 milyar bit |
| kbps | : Ağ bağlantısından 1 saniyede geçen bin bit      |

## KISALTMALAR

|      |                                                                      |
|------|----------------------------------------------------------------------|
| DPoS | : Delegated Proof of Stake (Delege Edilmiş Hisse İspatı)             |
| ECC  | : Elliptic Curve Cryptography (Elleptik Eğri Kriptografisi)          |
| IoT  | : Internet of Thing (Nesnelerin İnterneti)                           |
| IOTA | : Nesnelerin İnterneti için oluşturulmuş para birimi                 |
| ISS  | : Internet Servis Sağlayıcısı                                        |
| PoS  | : Proof of Stake (Hisse ispatı)                                      |
| Pow  | : Proof of Work (Emek İspatı)                                        |
| PBFT | : Practical Byzantine Fault Tolerance (Pratik Bizans Hata Toleransı) |
| QoS  | : Quality Of Service (Ağ İletişimi Hizmet Kalitesi)                  |
| QC   | : QoSChain                                                           |
| RSA  | : Rivest-Shamir-Adleman' a ait bir algoritma                         |
| SMC  | : Secure Multi-party Computation (Güvenli Çok Taraflı Hesaplama)     |
| VAKS | : Veri Akış Kurulum Süresi                                           |
| YDG  | : Yönlendirilmiş Döngüsüz Graf                                       |
| YTA  | : Yazılım Tanımlı Ağ                                                 |

## 1. GİRİŞ

Klasik ağ yapılarında sistem için standart bir protokolün olmaması cihazı sağlayan firmaların kendi yazılımları ve iletişim protokolleri ile ağ yapısını oluşturması sonucu yaşanan sıkıntılar Yazılım Tanımlı Ağ (YTA) kavramını karşımıza çıkarmıştır (Cicioğlu ve Çalhan, 2017; Kreutz vd., 2015). YTA altyapısı ile ağ topolojisinde kontrol düzlemi ve yazılım düzlemi birbirinden ayrılarak ağ yapısı daha esnek bir hale geldiği için yönetmek ve yönlendirmek daha kolay bir hale gelmiştir (Cicioğlu ve Çalhan, 2017; Kreutz vd., 2015). Ağ yapılarında iletişim çok yönlüdür, iletişim muhatapları yapı ya da kişiler arasındaki iletim kuralları birbirinden farklılık gösterebilir. Bir ağ için belirleyeceğimiz öncelikler ve bu iletişim ağına tanımlayacağınız bant genişliği vb. ölçekler birbirinden farklıdır. Ağ iletişimi hizmet kalitesi ile bant genişliği, gecikme, dalgalanma, güvenlik gibi bazı önemli iletişim unsurlarının belirlenmesi ve kontrol edilebilmesi hedeflenir (Wibowo vd., 2017). Ayrıca verilerin bir otorite altında değil de merkeziyetsiz yapılar halinde saklanması fikir birliğine dayalı bir şekilde erişilmesi, güncellenmesi yani dağıtık dijital cüzdan teknolojilerini temel alan bir sistemin oluşturulması, sistem kalitesi, esnekliği, hızı ve güvenliği gibi konuları olumlu yönde etkiler. Yapılmak istenen araştırmada dağıtık dijital defter teknolojilerinde altyapı olarak kullanılmakta olan Yönlendirilmiş Döngüsüz Graf (YDG) modeli kullanılmak istenmektedir (Wibowo vd., 2017). YDG tabanlı sistem yaklaşımı kullanılması ile önemli ölçüde uçtan uca yol hesaplama süresinin azaltılması hedeflenirken aynı zamanda merkezi araçlar ortadan kaldırılarak, karşılıklı olarak güven sıkıntısı yaşayan katılımcılar ve yapılar arasında fikir birliğine dayalı bir çerçeve sunulmak istenmektedir.

Bankalar arası iletişimden, telefon operatörleri arasındaki iletişime, farklı ticari şirketler arası iletişimden ya da şirket içi bölümler arası iletişime yani birçok gerçek dünya uygulamasında büyük miktarda veri birden çok taraf ve kişi arasında dağıtılır. Taraflar arasında bu veri paylaşımları yapılırken taraflar bu hassas ve gizli bilgilerinin korunmasını ister yani sadece gerekli olan iletişim sağlanmalı, veri paylaşımı yapılmalı ve tarafların kendileri hakkında korumak istedikleri bilgiler karşı taraf haricinde ulaşılmamalı veya erişilebilir olmamalıdır. Ağlar arası veri alışverişinde sağlanmak istenen gerekli bilgilerin güvenli ve bütünleşik bir yapıda doğru olarak iletilmesiyle birlikte hassas bilgilerin doğrudan paylaşılmadan korunması sağlanmalıdır. Veri alışverişinde tarafların bünyesinde bulundurduğu hassas verilerin korunması ve bu durumun belirlenen yönlendirme

standartları dahilinde hızlı ve istenilen şekilde sağlanması için bir çözüm önerisi sunulması amaçlanmaktadır. Bu doğrultuda bir dağıtık defter teknolojisi olan YDG modelinin altyapı olarak kullanılması planlanmaktadır. Modelde YTA mimarisinin esnek yapısından yararlanılmak istenmekte ve bir önceliklendirme veya yönlendirme ağ servisi olan Ağ İletişimi Hizmet Kalitesi (Quality of Service – QoS) ile desteklenmesi hedeflenmektedir.

Geleneksel ağ mimarileri, yönlendirici, anahtarlayıcı, dağıtıcı, sinyal tekrarlayıcı gibi birçok aygıttan ve bu aygıtlara tanımlı birçok farklı ve karmaşık protokolden oluşur. Hali hazırdaki sistemlerde oluşturulan her protokol birtakım problemleri çözmek için oluşturulmuş olsa da birçok farklı cihaz için aynı işlem yapıldığından beraberinde daha karmaşık ve belirli bir zaman diliminde yönetilmesi daha zor olan bir ağ yapısının oluşmasına neden olur. Bunun yanı sıra ağ sistemi oluşturulurken, yine ağ sistemini yönetmek ve üzerinde farklı uygulamalar yapmak için de farklı farklı üreticilerden herhangi bir standartı olmayan cihazlar ve yazılımlar kullanılabilir. Bu nedenle sistemi birbirine entegre etmek ve bir bütün haline getirmek herhangi bir standartlaşma olmadığı için çok daha zor hale gelecektir. İlerleyen durumlarda bu ağ sistemine yeni bir cihaz entegre etmek ya da sistem üzerinde herhangi bir değişiklik yapma isteği oluştuğunda, durum çok daha karmaşık hale gelecektir ki günümüzde ağ sistemlerinin güvenliğini, performansını ve benzeri özelliklerini arttırmak için kullanılan ara cihazların sayısı, ağ iskelet yapısını oluşturan cihazların sayısını geçmiş durumdadır (Wibowo vd., 2017). Sistemin bu kadar karmaşık olması, birbirinden farklı birçok protokol ve yazılım ile oluşturulması, ağ sistem yapısında açıkların oluşmasına ve hantal bir yapı haline gelmesine neden olur.

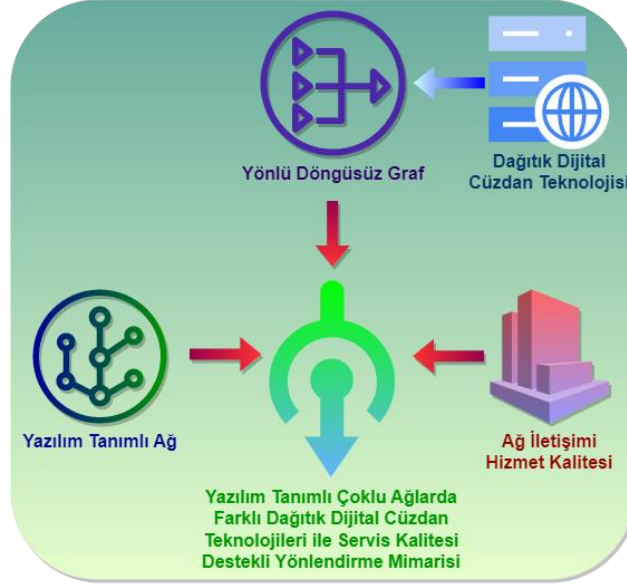
Yazılım tanımlı ağlar sınırları ve kısıtları ortadan kaldırırken klasik ağ sistemlerinde tümleşik olan veri ve kontrol düzlemini birbirinden ayırır. Böylece ağ yapısı daha basit ve otonom hale gelir. Standartlaştırılan protokollerle merkezden yönetilebilen, dinamik, yüksek performansa ve güvenlik yapısına sahip bir ağ modeli benimsenmiş olur. Bunun yanı sıra model hizmet kalitesi ile güçlendirilir. Hizmet kalitesi, ağ sistemindeki uygulamaların öncelik sıralamalarını belirtmek için kullanılır ve böylece zaman kaybı azaltılmış olur. Bu ağ servisi ile aynı ağ altyapısında devam eden farklı veri iletişimleri çeşitli tekniklerle düzenlenir ve öncelik durumuna göre sıralandırılır. Böylece ağlar üzerinde veri iletişiminde kesintiye, gecikmeye ve dalgalanmaya tahammülü olmayan paketler düzgün bir şekilde iletilmiş olur. Hizmet kalitesi güdümlü yönlendirme ile yazılım

tanımlı ağ mimarisi daha basit, ölçeklenebilir ve daha az zaman kaybı sağlayan bir yapı haline dönüşmektedir. Oluşturulan ağ mimarisi modelinde ise dijital cüzdan teknolojisi benimsenmektedir. Bir dijital cüzdan teknolojisinin avantajları arasında verilerin bir ağ üzerinde bulunan birden fazla cihazda ulaşılabilir olması değiştirilebilir veya güncellenebilir, onaylanabilir olmasına imkan sağlayan bir yapısının olması, dağıtık merkezi olmayan, teknolojik bir altyapıya sahip olması gösterilebilir.

Dağıtık dijital defter teknolojileri mantığı ile oluşturulan birçok farklı model vardır, bu araştırmada Yönlendirilmiş Döngüsüz Graf (YDG) modeli üzerine ağ modelimizi oluşturmayı hedefliyoruz. YDG modelinde düğümler arası iletişim tek taraflı ve tek yönlüdür. Örneğin A düğümünden B düğümüne bir bağlantının söz konusu olduğunu düşünelim, iletişim sadece tek yönlü olarak A düğümünden B düğümüne ve B düğümünden A düğümüne doğru bir direk bağlantı ilişkisinden söz edemeyiz. YDG'nin yapısı tek yönlüdür ve karşılıklı bağlantı söz konusu değildir. Bu yapısı sayesinde çoklu ağlarda taraflar arası hassas bilgilerin korunmasında ve ağı yönetmek için kullandığımız yönlendirme algoritmalarının neden olduğu yüksek yol hesaplama sürelerinin önüne geçmiş oluruz (Karakus vd., 2021).

Amaç çoklu ağ sistemlerinde taraflar arası veri alışverişini sağlarken hem karşılıklı olarak hassas bilgileri korumak hem de iletişim kalitesini üst düzeyde tutmak gerekmektedir. Bu doğrultuda ise;

- Dağıtık dijital cüzdan teknolojilerinden olan YDG Modeli temel alınacak,
- Oluşturulan model YTA mimarisi mantığı ile düzenlenecek,
- Ağ mimarisi YTA yapısının sağladığı esneklik ve ağ iletişimi hizmet kalitesi ile desteklenecektir.



Şekil 1. YDG Tabanlı Dijital Defter Teknoloji Mimarisi

Çoklu ağlarda klasik ağ yapılarının neden olacağı tümleşik, hantal, kontrol edilmesi ve yönlendirilmesi oldukça zor olan yapısından, YTA mimarisi modeli benimsenerek oluşturulmak istenen ağ topolojisi daha otonom hale getirilmektedir. Yazılım tanımlı ağlarda veri düzlemi ve kontrol düzlemi birbirinden ayrıldığı için ağ trafiğini yönetmek, veri akış tablolarına erişebilmek, ağ sisteminde yer alan cihazlara, anahtar yönlendiricilerine görev atamak gibi işleri tek ve standart bir yazılım ile sağlanabilir. Bunun yanı sıra YTA'lar ile veri bağı katmanında bağlantı sağlayarak katman yapılarının engelleri de ortadan kaldırılmış olur. Ayrıca ağın anlık durumunun takip edebilmesi ağ yapısını anlık olarak kontrol edilebilmesini sağlar.

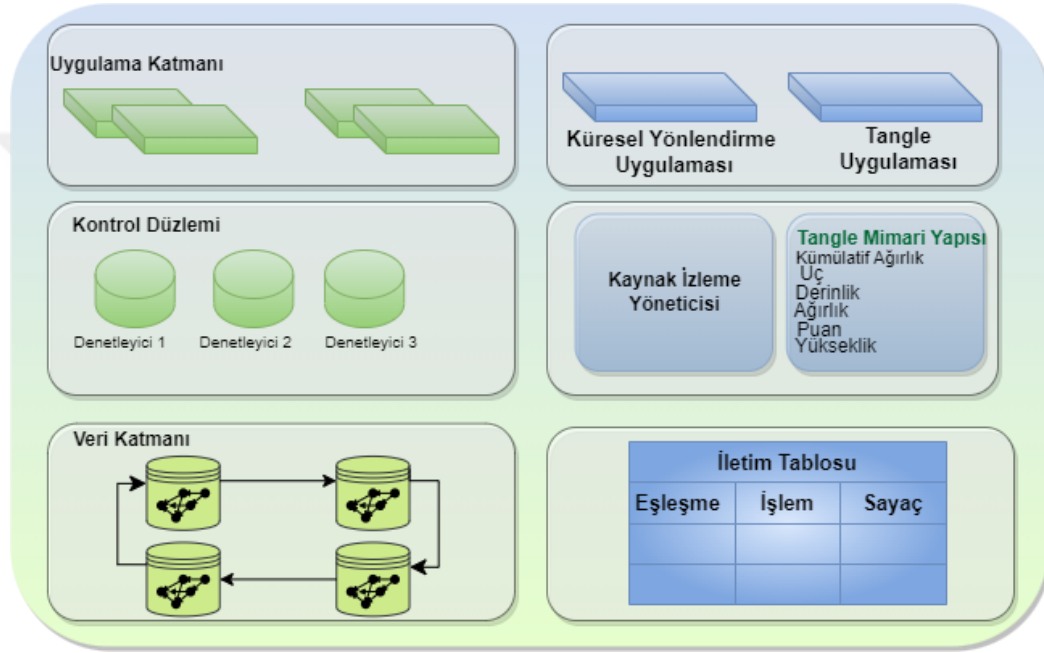
Ağ iletişimi hizmet kalitesi ile desteklenecek bu modelde izinler, yetkiler, veri iletim protokolleri yine verinin iletilmesinde belirlenecek olan tüm kurallar ve öncelik durumları merkezi olarak, kontrol altında olacak şekilde sağlanması beklenmektedir.

Çoklu ağlarda ağ iletişimi hizmet kalitesi sabitlerini hızlı bir şekilde karşılamaya çalışırken yönlendirme algoritmaları yüksek yol hesaplama süresi oluşturur. Bu nedenle hızlı bir şekilde değişen bağlantı, ağ iletişimi hizmet kalitesi sabitlerine uyum sağlayamaz. Bunun nedeni mevcut olan yönlendirme şemalarının ağ iletişimi hizmet kalitesi gereksinimlerine uyum sağlayabilmek için veri paketlerini iletme esnasında hem kaynak hem de hedef adresi kullanmasıdır. Bu sorunu ortadan kaldırmak için Şekil 1'de belirtilen altyapıyı destekleyecek YDG modeli kullanılır.

Burada hem kaynak hem hedef için ayrı ayrı yollar oluşturmak yerine, her hedef yönlendirici için tek bir yönlendirme modeli oluşturulur ve böylece önemli ölçüde düşük rota hesaplama sürelerine inmek hedeflenir.

### 1.1. Mimari

Yukarıda oluşturulmak istenen ağ topolojisi ile taraflar arası yürütülen iletişimin hem tarafların gizlilik beklentileri doğrultusunda gerçekleşmesi hem de iletişimin istenen kalitede sağlanması amaçlanmaktadır.



Şekil 2. Yazılım Tanımlı Ağ Mimarisi

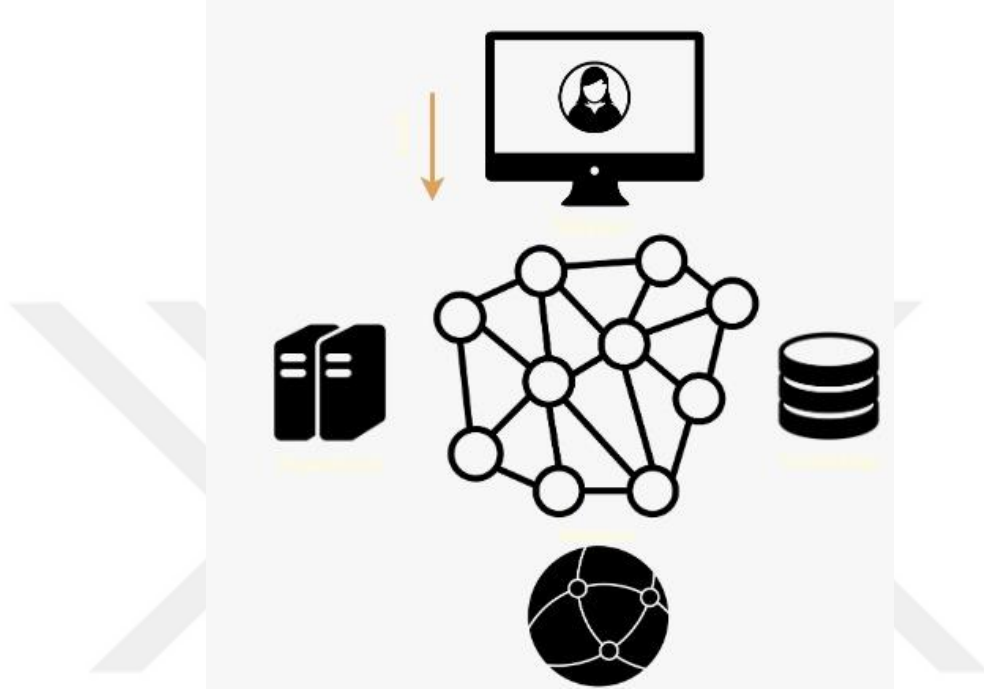
#### 1.1.1. Yazılım Tanımlı Ağ

İnternet kavramı dünyanın herhangi bir yerinden başka bir yerine iletişim kurabilen insanlar ve cihazları kapsayan dijital bir toplum oluşturmuştur. İnternet günden güne hızlı bir şekilde yaygınlaşan ve gelişen bir kavramdır. Teknoloji dünyasındaki gelişim ile birlikte kablolu ağ talep ve performansının artmasıyla 2000’li yıllardan günümüze 56kbps çevirmeli modemlerden 1Gbps bant genişliğine sahip fiber bağlantılara geçilmiştir. Günümüzde kullandığımız geleneksel ağ mimarisi bu hızla artan istekleri ve ihtiyaçları karşılayamamakla birlikte geleneksel ağ mimarisinin yönlendiricilerden, anahtarlardan, birçok aracı internet aygıtından ve bunlara tanımlanmış birçok karmaşık protokolden oluşması ağ yönetimini zorlayıcı ve karmaşık bir hale getirmektedir (Rischke ve Salah,

2020).



Ağı geliřtirmek ve gerekli ihtiyalar dahilinde daha kullanışlı hale getirmek iin herhangi bir standartı olmayan eklenecek ađ cihazları, ađ denetim yazılımları ve iřletim sistemleri ađ yapısını daha karmařık hale getirmektedir. Sisteme sonradan eklenen her aygıt ađ mimarisinin ynetimini daha zor hale getirmektedir. Yařanan bu sıkıntılar ve gereksinimler yazılım tanımlı ađ mimarisi kavramını ortaya ıkarmıřtır.

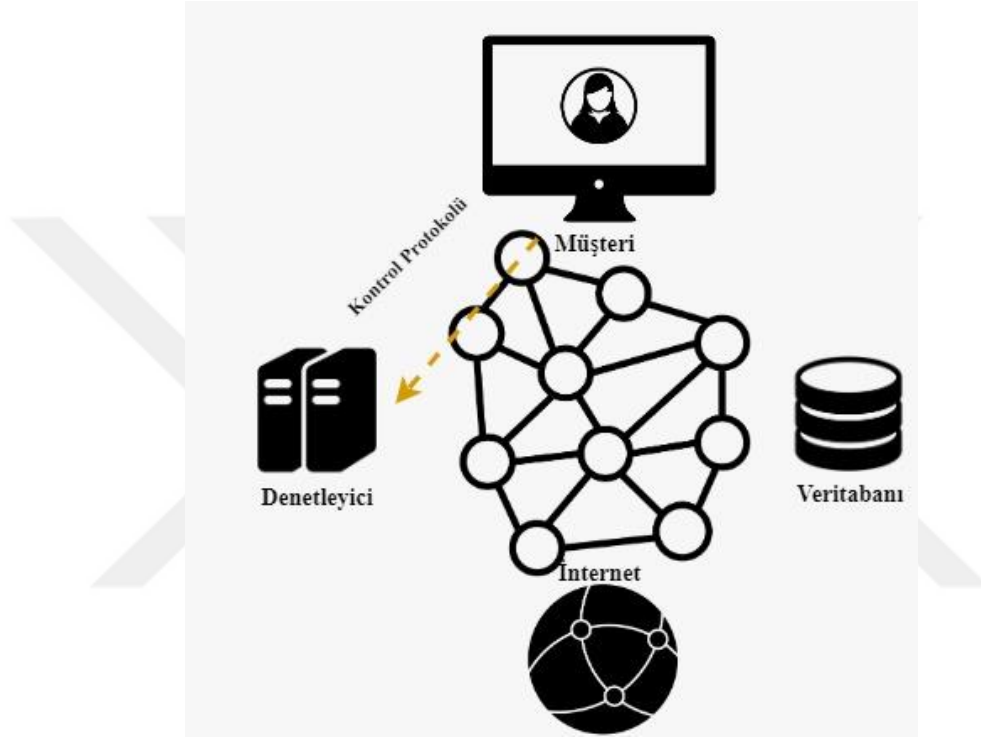


řekil 3. Yazılım Tanımlı Ađ Topolojisi – Adım 1

Yazılım tanımlı ađ mimarisinin, geleneksel ađ mimarisinin sahip olduđu dezavantajları ortadan kaldırması ve gnmzn hi durmadan byk bir hızla geliřen teknolojik ihtiyalarına daha etkili cevap vermesi onu popler hale getirmektedir. Yazılım tanımlı ađ mimarisinde, geleneksel ađ mimarisinden farklı olarak řekil 2’de gsterildiđi zere veri ve kontrol dzlemi birbirinden ayrıdır. Bu sayede tek bir merkezden ađdaki tm cihazları ynetebilen yksek performanslı, geliřtirmeye aık, hareketli ve olduka hızlı bir ađ mimarisi elde etmiř oluruz. Bylece servis bařarımı ve kaynakların kullanımını arttırmıř olur (Wibowo vd., 2017). Yazılım tanımlı bilgisayar ađlarında anahtarları ynetebilmek iin merkezi olmayan protokollere karřı merkezi bir denetleyici nerilmiřtir. Ama olarak, kresel bir grnme sahip merkezi bir ynetim sađlamak ve bu dođrultuda bir ađ modelindeki denetleyici ile anahtar/ynlendiriciler arasındaki iletiřim iin farklılık gstermeyen, standart hale getirilmiř bir protokol oluřturmaadır. Yazılım tanımlı ađlar, kontrol dzlemi ile veri dzleminin birbirinden ayrı olması ve kontrol dzlemine ihtiyaı

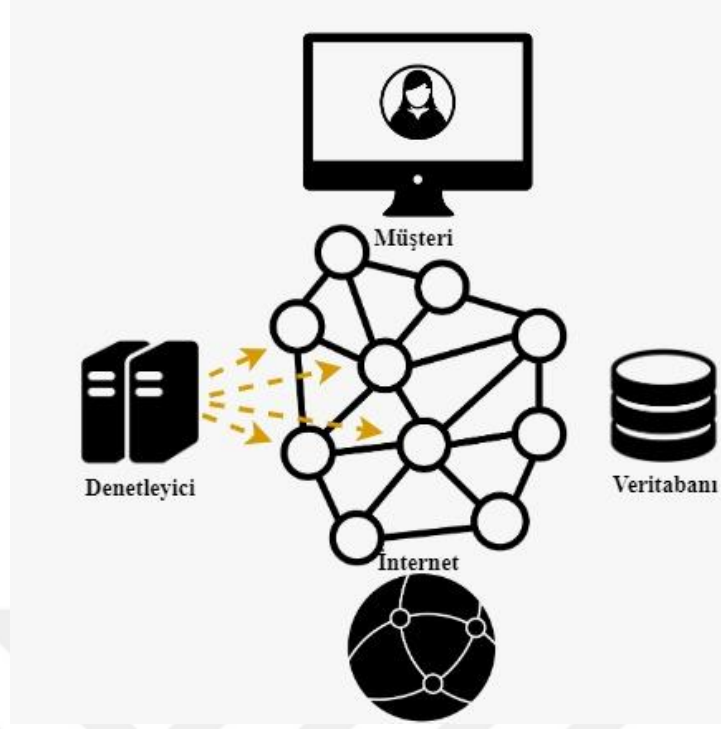
iletken ayrıca bir uygulama katmanına sahip olması ile diğer ağ modellerinden ayrılır. Yazılım tanımlı ağların bu yapısı herhangi bir değişiklik yapılacağı zaman ya da sisteme ek cihazlar ekleneceği durumlarda fiziksel yapının tümünden değiştirilmesi yerine, yeni protokoller ile kolayca istenilen işlemin yapılmasını sağlar. İşleyişi inceleyecek olursak;

İstemciler, bir ağ yapısı içinde iletişim kurmak ve bilgi almak için en yakın cihaz aracılığıyla istekte bulunur.



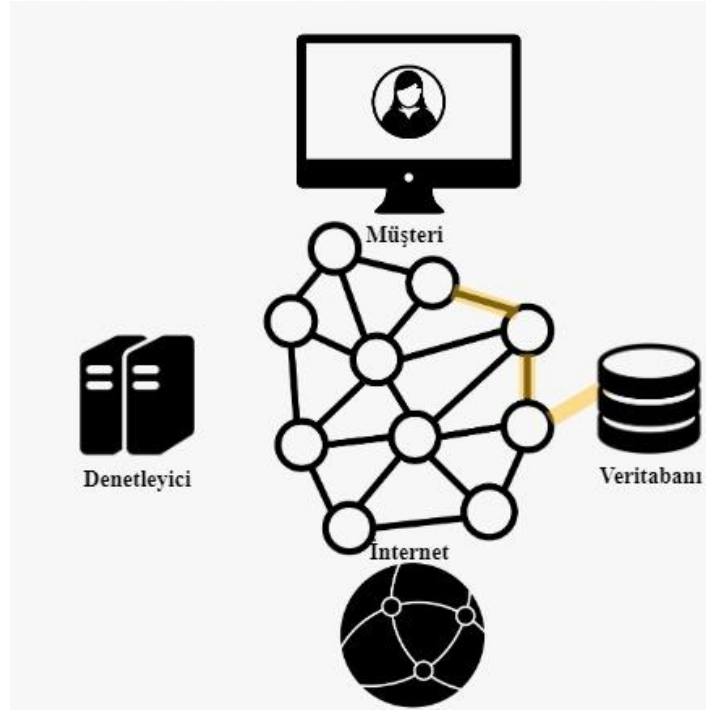
Şekil 4. Yazılım Tanımlı Ağ Topolojisi – Adım 2

Geçerli tabloda benzer bir sorgu olup olmadığı kontrol edilir. Aksi halde talebi alan cihaz merkezi denetleyici ile direkt olarak haberleşir ve denetleyici gelen trafik talebi ışığında nasıl hareket etmesi gerektiğini sorar. Denetleyici, veri katmanında herhangi bir kural olup olmadığını kontrol eder. Bu davranış, bir akış kuralı mevcut olduğunda gönderilir. Kural yoksa akış tablosuna tanımlanan kurallar dahilinde yeni bir kural eklenir ve davranışı sorgulayan cihaza yanıt olarak gönderilir.

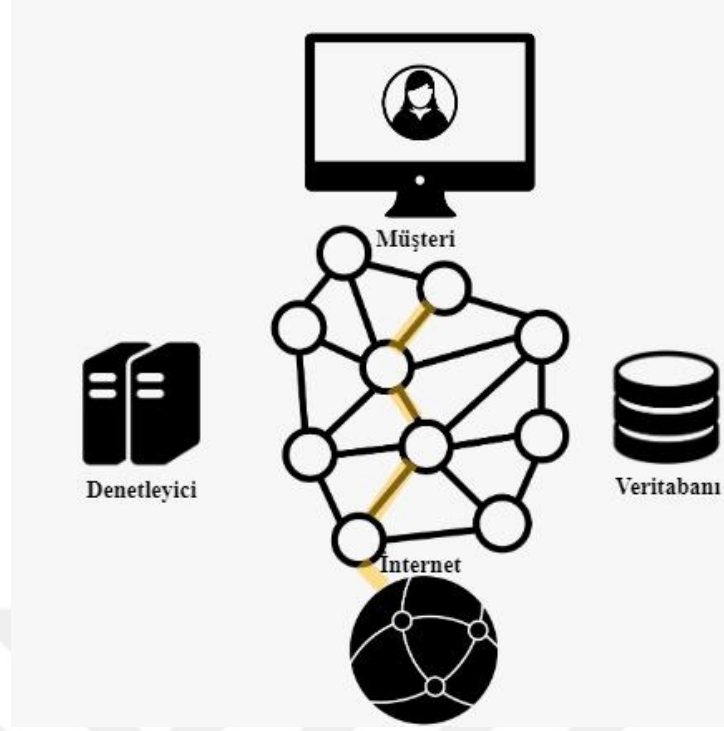


Şekil 5. Yazılım Tanımlı Ağ Topolojisi – Adım 3

Belirlenen kurallara bağlı olarak, gelen paket istekleri için kullanılan yol bilgisi denetleyici tarafından kullanılan cihaza gönderilir.



Şekil 6. Yazılım Tanımlı Ağ Topolojisi – Adım 4



Şekil 7. Yazılım Tanımlı Ağ Topolojisi – Adım 5

Müşteri, temsilci tarafından yönlendirildiği şekilde istediği veri tabanına ve internete erişebilir.

#### 1.1.1.1.Kontrol Katmanı

Tüm ağ operasyonları bu katmanda yönetilmektedir. Ağ yapısının mantıksal mekanizmasıdır ve cihazlardaki tanımlı protokollerin çalışmasını sağlar. Yönlendirme tabloları, adres tabloları ve yönlendirme protokolleri bu düzlemde bulunmaktadır. Uygulama katmanından gelen istekleri veri katmanına iletir.

#### 1.1.1.2. Veri Katmanı

Donanımsal yapının tam kontrolü bu katmanda yapılmaktadır ve cihazları kontrol katmanına bağlar. Temel amacı iletilen verileri ve işleme sürecini kontrol etmektir. Bir cihaza iletilecek ağ paketlerini ve buna ait çerçevelerini alır, işler ve iletir.

#### 1.1.1.3.Uygulama Katmanı

Uygulama katmanı saldırı tespit sistemi, yük dengeleme, güvenlik duvarları gibi kurum ve kuruluşların kullandığı yaygın ağ uygulamalarını ve görevlerini içerir. Bunları kontrol katmanına açık ve programlı bir şekilde iletmekle görevlidir.

### 1.1.2. Yönlendirilmiş Döngüsüz Graf

Yönlendirilmiş Döngüsüz Graf (YDG) belirli bir bağlamda bir dizi değişken ve bunların ölçümleri için varsayılan veri oluşturma sürecinin parametrik olmayan diyagramatik temsilleridir. Değişkenler ve ölçümleri, aralarında varsayılan ilişkileri gösteren tek yönlü yaylar (veya oklar; dolayısıyla 'yönlendirilmiş') ile bağlanan düğümler olarak tasvir edilir. İki düğüm arasındaki bir bağlantı, bir nedensel ilişkinin varsayılan varlığını ve yönünü belirtmektedir. Ancak işareti (yani pozitif veya negatif), büyüklüğü (yani büyük veya küçük), şekli (örneğin doğrusal veya doğrusal olmayan) veya biçimini belirtmez. Bu ilişki, (dolayısıyla 'parametrik olmayan') bir düğümün kendisinden kaynaklanamaz (dolayısıyla 'döngüsel' değildir). Yol, iki düğümü birbirine bağlayan bir veya daha fazla yayın topluluğudur. Yollar açık veya kapalı olabilir; açık yollar istatistiksel ilişkileri iletir, kapalı yollar ilişkisizlik anlamına gelmektedir. Nedensel yol, tüm kurucu yayların bir düğümden diğerine aynı yönde aktığı yoldur. Birlikte odak ilişkisini oluşturan belirli bir maruziyetin (yani nedenin) belirli bir sonuç üzerindeki toplam nedensel etkisi, maruziyeti sonuca bağlayan tüm nedensel yollardan iletilen ortak etkidir. Odak ilişkisi ile ilgili olarak, bir karıştırıcı hem maruziyetin hem de sonucun ortak bir nedenidir, bir arabulucuya maruz kalmaya neden olur ve sırayla sonuca neden olur (yani maruz kalma ile sonuç arasında nedensel bir yola düşer). Bu sebeple, bir rekabet halindeki maruziyet, maruziyetin ne sebep olduğu ne de sebep olduğu sonucun bir nedenidir.

#### 1.1.2.1. Yönlendirilmiş Döngüsüz Graf Uygulamaları

*Görev zamanlaması:*

YDG'ler, görevler arasındaki bağımlılıkları temsil eder ve iş akışlarını yönetmek için kullanılır. Özellikle proje yönetimi ve iş akışı otomasyonu için popülerdir. YDG yapıları, bir işi veya görevi tamamlamak için diğer işlerin tamamlanmasına dayanan karmaşık iş süreçlerini yönetmek için kullanılır. Her görev, diğer görevlerin tamamlanmasını bekler. Bu taahhüt, yönlendirilmiş bir kenar ile temsil edilir. Örneğin, bir yazılım geliştirme projesinin birden çok adımı ve yazılım sürümlerine bağımlılığı vardır. Önce belirli modülleri tamamlamanız ve ardından sınavı geçmeniz gerekebilir. Bu adımlar arasındaki bağımlılıkları eşlemek ve iş akışlarınızı verimli bir şekilde yönetebilmek için YDG'leri kullanmamıza olanak tanır.

### *Versiyon Kontrol Sistemleri:*

Sürüm kontrol sistemleri, yazılım geliştirme projelerinde kodu yönetmek ve izlemek için kullanılır. Her değişiklik bir düğümü temsil eder ve bu değişiklikler ana düğümlere yönlendirilmiş kenarlarla bağlanır. Proje geçmişi bu nedenle bir YDG olarak temsil edilir. Örneğin, Git sürüm kontrol sistemi, sözde taahhütteki her değişikliği bir düğüm olarak temsil eder ve her taahhüdün, yönlendirilmiş kenarlarla bağlanan ana taahhütlerde yapılan değişikliklerin bir ağacı vardır. Bu, proje geçmişini izlemek ve farklı sürümleri kontrol etmek için oldukça kullanışlıdır.

### *Blockchain Teknolojisi:*

Bazı blockchain projeleri, özellikle IOTA ve Nano gibi kripto para projeleri, yönlendirilmiş döngüsüz graf yapıları kullanır (Perazzo vd., 2020). Bu projelerde işlemler bloklar yerine doğrudan YDG yapısındaki düğümlerde saklanmaktadır. İşlemlerin işlenebilmesi için tüm ağ katılımcıları tarafından onaylanması ve önceki işlemlerle ilişkilendirilmesi gerekir. Bu yaklaşım, madenciler veya geleneksel blok tabanlı blokzincirler gibi belirli bir süreç gerektirmeyen bir fikir birliği mekanizması sağlar. Bu oluşturulan altyapı, işlemleri daha hızlı ve daha ölçeklenebilir hale getirmektedir. Ayrıca, çift harcama problemini ortadan kaldırmak için ek güvenlik önlemlerini sağlamaktadır.

### *Veri tabanı Yapıları ve Optimizasyonlar:*

YDG yapıları, bazı veri tabanı yönetim sistemlerinde ve derleyici optimizasyonlarında kullanılır (Ding ve Sato, 2020). Bir veri tabanı, bağımlılıkları daha iyi yönetmek için bir YDG yapısı kullanılabilir. Veriler arasındaki ilişkiler ve bağımlılıklar, işlemlerin sırasını ve verilerin nasıl düzenlendiğini belirlemek için kullanılır. Derleyici optimizasyonları, ortak alt ifadeleri tanımlamak ve gereksiz hesaplamalardan kaçınmak için genellikle YDG'leri kullanır. Özellikle kod optimizasyon aşaması, program yürütme süresini ve kaynak kullanımını optimize etmek için gerekli önlemleri belirler. YDG yapısı, birçok farklı alanda kullanılabilen çok yönlü bir veri yapısıdır. Görev zamanlaması, sürüm kontrol sistemi, blokzincir teknolojisi, veri tabanı yönetimi vb. için uygulanır. YDG'ler, bağımlılıkları ve akışları temsil etmek, verileri düzenlemek ve işlemleri kolaylaştırmak için güçlü araçlardır.

### 1.1.2.2. Yönlendirilmiş Döngüsüz Grafları Diğer Dağıtık Dijital Cüzdanlardan Ayıran Özellikleri

#### *Ölçeklenebilirlik:*

Yönlendirilmiş döngüsüz graflar, blokzincir tabanlı cüzdanlara kıyasla mükemmel ölçeklenebilirlik sunar. Blokzincir tabanlı cüzdanların bu blokları işlemesi, üzerlerine yeni işlemler eklendikçe uzun zaman alabilmektedir. Blokzincir cüzdan performansı, özellikle yüksek işlem trafiğinde düşebilmektedir. Ancak, yönlendirilmiş döngüsüz graflar tabanlı cüzdanlar, her işlemin kendi bloğunu oluşturmaya izin vererek işlemlerin paralel olarak işlenmesini sağlayarak sistemin daha yüksek işlem hızlarına ulaşmasını sağlar.

#### *Konsensüs Mekanizması:*

Yönlendirilmiş döngüsüz graflar defterleri daha basit bir mutabakat mekanizmasına sahiptir ve genellikle madencilik gibi işlemlere bağlı değildir. Bu, yönlendirilmiş döngüsüz graflar tabanlı bu defter doğrulama ve onaylama süresini azaltır ve enerji tüketimini azaltır.

#### *Ücretsiz İşlem:*

Bazı yönlendirilmiş döngüsüz graflar tabanlı cüzdanlar, işlem ücreti olmadan işlem gerçekleştirme olanağı sunar. Bu, kullanıcılar için daha düşük işlem maliyetleri anlamına gelir. Ödemeler ve sık mikro işlemler için kullanılabilir.

#### *Hafif Düğümler:*

Yönlendirilmiş döngüsüz graflar cüzdanları, tüm blokzincirini indirmek ve depolamak yerine, kullanıcıların yalnızca doğrudan kendi işlemleriyle bağlantılı işlemleri tutmalarına izin verir. Bu, düğümde daha az bellek ve işlem gücü gerektiği anlamına gelir ve cüzdanı daha hafif hale getirir.

### 1.1.2.3. Yönlendirilmiş Döngüsüz Graflar'ın Kullanıldığı Alanlar

#### *Kripto para birimleri:*

Bazı kripto para birimleri yönlendirilmiş döngüsüz graf tabanlı bir yapı kullanır. Örneğin, IOTA, Nano ve Hedera Hashgraph gibi kripto para birimleri, işlem onay sürelerini azaltmak ve işlem ücretlerini azaltmak için yönlendirilmiş döngüsüz graf teknolojisi ile çalışır (Perazzo vd., 2020).

#### *Nesnelerin İnterneti (IoT):*

Yönlendirilmiş döngüsüz graflar, Nesnelerin İnterneti (IoT) cihazlarından gelen verilerin güvenli ve hızlı bir şekilde işlenmesini sağlar. IoT cihazları çok büyük miktarda veri üretir ve bu verileri hızlı bir şekilde işlemek ve güvenli bir şekilde paylaşmak önemlidir (Perazzo vd., 2020).

#### *Dağıtılmış Dosya Sistemi:*

Yönlendirilmiş döngüsüz graf teknolojisi, verileri dağıtılmış bir dosya sisteminde yönetmek ve depolamak için kullanılabilir. Bu, büyük miktarda veriyi depolamak ve paylaşmak için kullanılan veri merkezlerinde ve bulut tabanlı hizmetlerde faydalıdır.

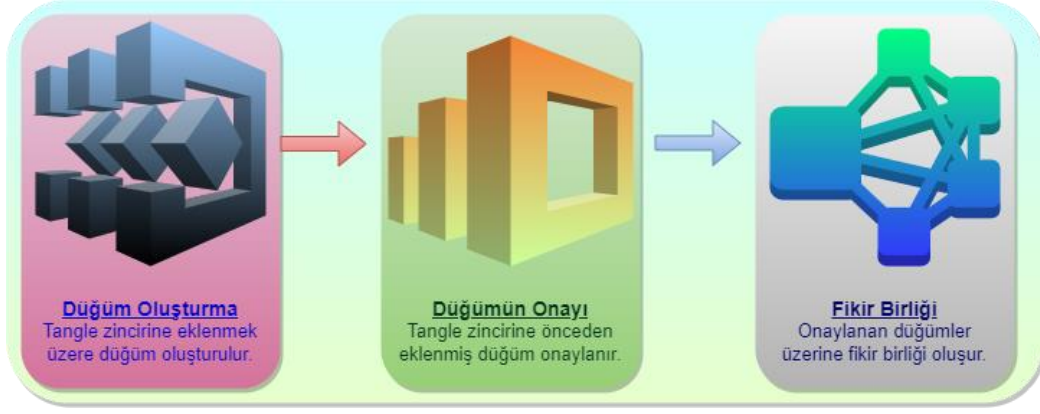
#### *Oylama ve Seçim Sistemleri:*

Oylama ve seçim sistemlerinde yönlendirilmiş döngüsüz graf teknolojisi kullanılabilir. Blockzincir tabanlı oylama sistemlerine alternatif olarak yönlendirilmiş döngüsüz graf yapıları, güvenli, hızlı ve uygun maliyetli bir oylama süreci sağlayabilir.

Bu nedenle yönlendirilmiş döngüsüz graflar, ölçeklenebilirlik, hızlı işlem onayı, düşük işlem ücretleri ve hafif düğümler gibi özelliklerle diğer merkezi olmayan dijital cüzdanlardan farklıdır. Sanal para birimi, nesnelerin interneti, dağıtık dosya sistemi, oylama sistemi gibi çeşitli alanlarda kullanılabilir.

### 1.1.3. Tangle

Tangle ağ modeli 2015'te ortaya çıkan, birbirine oklarla bağlı düğümlerden oluşan diğer dağıtık dijital defter teknolojileri gibi herkesin kullanımına açık bir modeldir. Her düğüm en az iki düğümle bağlantılı olmalıdır. Yeni bir düğümün ağa dahil olabilmesi için kendinden önce gelen herhangi iki düğümü onaylaması ve doğrulaması gerekir. Doğrulanmış bu düğümler ipuçları olarak adlandırılır. Bu ipuçları bünyesinde bazı verileri ve işlem belgesini içerir. Tangle yönlendirilmiş döngüsüz graf ağ yapısını kullanır. Yönlendirilmiş döngüsüz graf veri yapısı verileri, sadece tek bir yönde hareket eden bir graf formatında saklayarak geçmişteki ve gelecekteki işlemleri doğrulayamayacağı biçimde dairesel olmayan bir defter türüdür. Tangle modelinde fikir birliği mekanizmasında her işlem önceki iki işlemi onaylayıp doğrulaması gerekir. Bu doğrulama ve ağ yapısına dahil olma işleminde seçilen bir algoritmaya dayalı olarak kendinden önceki iki işlemi seçmeli ve kriptografik problem çözümlenerek tanımlanır. Bu durumda her düğüm zaten bir madenci olduğu için bu mimaride bir madenciye ihtiyaç duyulmamaktadır. Bu nedenle de yeni bir işlem oluşturmanın herhangi bir maliyeti bulunmamaktadır. Tangle ağlarında ölçeklenebilirliğin artışı tamamlanmış ya da güncellenmiş eski işlemlerin oluşturduğu karmaşık yapı sayesinde olmaktadır (Chafjiri ve Esfahani, 2019). Yeni oluşturulan işlemin onaylanma süreci konsensüs yapısına bağlı olarak iki rastgele seçilen önceki işlemlerin onaylanması ile yapılmaktadır. Bu sayede onaylanma işlemi için bir sonra oluşacak bloğu onaylama işleminde herhangi bir zaman kaybı oluşmayarak doğrulanma hızlıca yapılmış olacaktır. Tangle mimarisi sayesinde eş zamanlı olarak dağıtık dijital defterdeki kayıtların sayısının artmasıyla daha fazla işlem doğrulaması yapılabilmektedir. Bu şekilde sistemin çalışması tangle dağıtık dijital defter yapısının günümüz kuantam bilgisayarlara karşı daha güçlü olarak korunmasını sağlamaktadır (Bhandary vd., 2020; Popov, 2018).



Şekil 8. YDG – Tangle Ağ Mimarisi

#### 1.1.4. Ağ İletişimi Hizmet Kalitesi

Ağ İletişimi Hizmet Kalitesi (QoS), servis sağlayıcı ile son kullanıcılar arasındaki bir servis düzeyi sözleşmesini (diğer bir deyişle, müşterinin ne alacağını belirtir ve servis sağlayıcısından ne beklendiğini açıklığa kavuşturur) karşılamak için bant genişliği, gecikme, dalgalanma ve güvenilirlik gibi belirli ağ gereksinimlerini garanti etme durumu olarak adlandırılır. Hizmet kalitesi, çalışılan ağdaki trafik üzerinde kontrol sağlamak ayrıca kısıtlı ağ hacmine sahip uygulamaların çalışabilirliğini stabilize etmek için sistem üzerinde kullanılan teknolojilerdir. Ağ akışında belirlenmiş olan yüksek performanslı uygulamaları önceliklendirerek ağ trafiğini yönetmesini sağlar. Aşağıda Ağ İletişimi Hizmet Kalitesi standartları detaylı bir şekilde anlatılmıştır.

##### 1.1.4.1. Bant Genişliği

Bant genişliği, saniye başına bit (bps) cinsinden bağlantı hızıdır. Hizmet kalitesi gereksinimli ağ iletişimlerinde, yönlendiricinize bu bant genişliğini nasıl kullanacağını söylemenizi sağlar. İlk giren ilk çıkar (First In First Out ) ile paketler ilk gelene ilk hizmet esasına göre teslim edilir. Diğer şeylerin yanı sıra hizmet kalitesi gereksinimleri, farklı kuyruklar oluşturmanıza ve belirli trafik türlerini farklı kuyruklara koymanıza olanak tanımaktadır. Daha sonra bir yönlendirici, örnek olarak birinci katmandaki bant genişliğinin %50'sini, ikinci katmandaki bant genişliğinin %20'sini ve katman 3'teki bant genişliğinin kalan %30'unu alacak şekilde yapılandırılabilir.

#### 1.1.4.2. Gecikme

Bir veri iletiminde gönderilmek istenilen paketin kaynağından hedefine ulaşması için geçen süredir. Buna tek yönlü gecikme denir. Kaynaktan hedefe ve geri dönüş için geçen süreye gidiş-dönüş gecikmesi denir. Uçtan uca hizmet kalitesi destekli veri iletim modellerinde çeşitli gecikme türleri vardır.

##### *İşlem Gecikmesi:*

Cihazın bir paket göndermek için gereken tüm görevleri tamamlaması için geçen süredir. Örneğin, yönlendiriciler yönlendirme tablosu aramaları yapmalı, adres çözümleme protokolü tablolarını, giden erişim listelerini ve benzerlerini kontrol etmeleri gerekmektedir. Yönlendirici modeline, işlemci hızına ve anahtarlama yöntemine göre işlem gecikmesi değişiklik gösterebilmektedir.

##### *Kuyruk Gecikmesi:*

Bir paketin işleme alınıp hatta yazılma zamanına kadar kuyrukta beklediği süreye kuyruk gecikmesi denir. Arabirim tıkanmışsa, paketlerin gönderilmeden önce sırada beklemesi gerekebilmektedir.

##### *Serileştirme gecikmesi:*

Bir çerçevenin tüm bitlerini iletim için fiziksel ara yüze göndermek için geçen süreyi temsil etmektedir ve bekleme süresi fiziksel ara yüz hızına bağlılık göstererek artış veya azalış sağlayabilmektedir.

##### *Yayılma Gecikmesi:*

Bir bitin fiziksel bir alanı geçtiği süreye denir. Örneğin, bir bitin 10 millik bir fiber optik bağlantıyı kat etmesi, bitin bir uydu bağlantısını kullanmasına kıyasla çok daha az zaman alır. Yayılma gecikmesi gibi bu gecikmelerden bazıları değiştirilemez. Ancak hizmet kalitesi gereksinimlerine bağlı olarak, kuyruk gecikmelerini etkilemenize izin verir. Örneğin, her zaman diğer kuyruklardan önce hizmet verilen bir öncelik sırası oluşturabilirsiniz. Ses paketlerini öncelik sırasına eklemek, ses paketlerinin kuyrukta uzun süre beklemesini önleyerek sıradaki bekleme sürelerini azaltır.

#### **1.1.4.3. Dalgalanma**

Bir paket akışının tek yönlü gecikmesindeki değişiklidir. Örneğin, bir IP telefon sabit bir ses paketi akışı gönderir. Ağ tıkanıklığı nedeniyle bazı paketler gecikir. 1. ve 2. paketler arasındaki gecikme 20ms, ikinci ve üçüncü paketler arasındaki gecikme 40ms, üçüncü ve dördüncü paketler arasındaki gecikme 5ms olur. Böylece sabit bir gecikme yaşanır aksi halde ses kalitesi düşecektir.

#### **1.1.4.4. Kayıp**

Genellikle gönderilen kayıp paketlerin yüzdesi olarak ifade edilen, kaybedilen veri miktarıdır. 100 paket iletir ve sadece 95'i gitmesi istenen hedefe ulaşır, paketteki kaybınız %5 olacaktır ve bu kayıp olan paketlerin veri bütünlüğünün korunması amacıyla tekrar gönderilmesi gerekmektedir. Bu durum, paketlerin gönderim miktarlarını artırmakla birlikte ağ altyapılarındaki trafik yoğunluğunu da önemli ölçüde artırmaktadır. Paket kaybı çoğu zaman yaşanabilir. Örneğin, tıkanıklık oluştuğunda paketler kuyruğa alınacaktır ve ancak sıra gelene kadar kuyruk kapasitenin elverdiği ölçüde paket kuyrukta bekletilebilir. Aksi takdirde, devamlı fazla miktarda gelen verilerin bir kısmı kuyruk kapasitesi dolduğu için kaybolacaktır. Bu durumda hizmet kalitesi ile hangi paketlerin bırakılacağına karar vermemiz gerekebilmektedir.

## 2. LİTERATÜR ÖZETİ

Yapılan literatür taramalarında aşağıdaki gibi çalışmalar incelenmiştir. Makalede Yönlendirilmiş Döngüsüz Graf topolojilerine uygulanabilen parametreleştirilmiş bir mutabakattan bahsedilmiştir. Mutabakat her bir düğümdeki mirasçı listesinin topoloji sırası ve en kısa yol yönlendirme tablosu olmak üzere iki farklı protokol olarak tasarlanmış ve iki kere örneklendirilmiştir. Bu doğrultuda topolojinin daha genel topolojilere uygulanabileceği göz önüne alındığında yapılan iş yalnızca optimum yönlendirme için değil, aynı zamanda sağlamlık ve arızaların olduğu geniş bir dağıtım sınıfı için de uygun olduğu tespit edilmiştir (Das vd., 1999).

Bu makalede ise bir ağ bağlantısında bağlantıdan ve düğümden bağımsız Yönlü Döngüsüz Graf modelinden bahsedilmiş ve bu modeli hesaplamak için çok yollu yönlendirme sağlayan, tek bağlantı hatasından kurtarmayı garanti eden ve tüm olasılıkları kullanan polinom zaman algoritmaları geliştirmişlerdir. Araştırmalar, hedef adrese ve gelen bağlantıya dayalı yönlendirmenin ek yük olarak paket başına en fazla 1 bit çalıştığını göstermiştir (Cho vd., 2010).

Bu makalede düşük güç ve daha az kayıplı ağ modeli oluşturabilmek için bir yönlendirme protokolü tasarlanmıştır. Bu protokol ağda bulunan düğümleri var sayılan varış noktasına yerleşmiş bir ya da birden fazla Yönlendirilmiş Döngüsüz Graf'ta düzenleme hedeflenmiştir. Protokolde belirtilen noktadan noktaya (Eşler arası) yönlendirme mekanizması, Yönlendirilmiş Döngüsüz Graf boyunca paketin hedefine giden yolu bilen bir düğüm tarafından geri gönderilene kadar paketleri bir Yönlendirilmiş Döngüsüz Graf aracılığıyla yukarı doğru hareket ettirir. En kısa Eşler arası rotaları arasındaki yönlendirme maliyeti farkını ölçülmüş ve oluşturulan protokolde daha iyi bir Eşler arası yönlendirme çözümü aranmıştır (Xie vd., 2010).

Makalede çoklu kısıtlamalı Ağ İletişimi Hizmet Kalitesi yönlendirmesini karşılayan matematiksel algoritmalara dayalı bir model oluşturulur ve çoklu kısıtlamaları ceza fonksiyonu aracılığı ile uygunluk fonksiyonuna dönüştürmüştür. Önerilen algoritma ağ topoloji haritasını budar, çoklu kısıtlamaları karşılamaya yolları temizler ve arama kapsamını en aza indirir. Ayrıca optimuma düşme sorununu çözmek ve aramayı hızlandırmak için parçacık sürüsü optimizasyonunu kullanır (X. Hu vd., 2018).

Bu makalede Ağ İletişimi Hizmet Kalitesi 'ne dayalı ve Yönlendirilmiş Döngüsüz Graf tabanlı kısıtlı yönlendirme algoritmasından bahsedilmiştir. Bu algoritmanın iki işlevi vardır. Birincisi budama işlemidir bununlar birden fazla kısıtlamanın gereksinimlerini karşılayan tüm yollar bulur ve Yönlendirilmiş Döngüsüz Graf'a dönüştürür. Diğeri ise arama stratejisidir. Simülasyon algoritmanın yolu en az maliyetle hızlı bir şekilde bulabildiğini ve büyük ölçeğe sahip kısıtlı yönlendirme ağları için uygun olduğunu gösterir (Y. Hu vd., 2018).

Bu makalede blokzincirde doğrusal yapı ve fikir birliği algoritmalarından, sınırlı kaynak ve bant genişliği sorunlarından bahsedilmiş, YDG yapısının paralel veri işleme sağlayarak yüksek ölçeklenebilirliği vurgulanmıştır. Çalışmada blokzincir ve Tangle teknolojilerinin birlikte çalıştığı bir modelde her iki teknolojiye aracılık eden bir konektör kısmına odaklanılmış ve akıllı sözleşmelerin tangle düğümlerinde düzgün bir şekilde çalışmasını sağlayan bu yeni platformu güvenlik ve çevrim dışı çalışma özellikleri ile daha kullanışlı hale getiren bir mesaj kuyruklama protokolü ortaya çıkmıştır (Kotilevets vd., 2018).

Günümüzdeki büyük problemlerden biri olan veri güvenliğinden ve bu konuda yaşanan sorunlardan bahsedilmiştir. Blokzincir teknolojisinin enerji yönetimi kamu bilgi sistemi, emlak sektörü gibi birçok alanda kullanılarak üst düzey bir güvenlik sağlanabileceğinin altı çizilmiş aynı zamanda yine blokzincirin handikaplarından da bahsedilmiştir. Çalışmada blokzincir ve YDG birleştirilerek oluşturulacak yeni ağ modelinin blokzincirin sahip olduğu temel dezavantajları kaldıracağı, düşük güçlü cihazlarında kullanılabileceği belirtilmiştir (Schueffel, 2018).

Bu makalede ise kümeleme ve ağ kodlama tekniklerinin kullanıldığı Yönlendirilmiş Döngüsüz Graf tabanlı bir yayma yaklaşımı önerilmiştir. Burada ana amaç kayıp verilerin ve paketlerin kurtarılmasını sağlarken ağ güvenliğini arttırmak, enerji tüketimini fazlasıyla aza indirmek ve ağ geçitlerinde yükü dengelemeyi amaçlamaktır. Deneysel sonuçlar Yönlendirilmiş Döngüsüz Graf tabanlı bu mimarinin güçlü performanslar sergilediğini göstermektedir (Al-Hawri vd., 2020).

Makalede; Güvenli çok taraflı hesaplama (Secure multi-party computation- SMC) ile birleştirilmiş YDG ağ modelinden bahsedilmiştir. Güvenli çok taraflı hesaplama, tarafların kendi girdileri üzerinde bir işlevi ortaklaşa hesaplamaları ve bu girdileri gizli tutmaları için

kullanılan kriptografik bir modeldir. Fakat bu model belirli uygulamalar ile kullanılabilir ve kısıtlıdır. Bu sorunu çözümleyebilmek için temel işlemlerden oluşan ve YDG ile modellenmiş (+,-,\*,/) bir gizlilik modeli önermişlerdir ve modelin ardışık düzende çalışması dahilinde çeşitli ve karmaşık işlevleri uygulayabilir (Teo vd., 2020).

Bu makalede Yazılım Tanımlı Ağ modelinin küçük ve orta ölçekli yapılarda başarıyla kullanılırken geniş alan ağları kapsamında kullanımının handikaplarından (düğüm ve bağlantı arızaları, özel kontrol kanalının bulunmaması) bahsedilmiş ve handikapları giderebilmek için YDG tabanlı bir yönlendirme modeli üzerinde çalışılmıştır. Model bağlantı- düğüm hatalarını seri bir şekilde yüklenmesini sağlarken güncellemeler sırasında tutarsız iletme tablolarının oluşma durumunu önler ayrıca çevrimiçi yol hesaplama süresinin uzaması durumunun öne geçtiği çalışmalar ile gözlenmiştir (Avallone ve Ashraf, 2020).

Makalede yazılım tanımlı yer paylaşımli ağlardaki ana zorluklardan biri olan Hızlı değişen Ağ İletişimi Hizmet Kalitesi özelliklerinden bahsedilmiş, ayrıca çoklu Yönlendirilmiş Döngüsüz Graf kısıtlamalarını karşılamak için var olan yönlendirme algoritmaları yüksek yol hesaplama süreleri içerir ve bu algoritmalar çok çabuk değişen Ağ İletişimi Hizmet Kalitesi özelliklerine uyum sağlayamadığına değinilmiştir. Bu sorunu çözümleyebilmek için YDG tabanlı yönlendirme şeması olan Qroute önerilmiştir. Model, düşük rota hesaplama süresi ve veri düzlemindeki yönlendirme girişlerini azaltması ile başarı sağlamıştır (Varyani vd., 2020).

Makalede düzgün olmayan izleme yapılandırmalarında topoloji eşleme yönlendirmesi gerçekleştirmek için, veri yollarını belirli bir topolojide bağlamak amacıyla Yönlendirilmiş Döngüsüz Graf tabanlı bir algoritma önerilmiş ve benzer pin konumlarını paylaşan otobüsleri gruplandırarak yönlendirme karmaşıklığını azaltan bir veri yolu kümeleme tekniği uygulanmak istenmiştir. Ayrıca, rota sıkışıklığını azaltmak için bir sökme ve yeniden yönlendirme şeması uygulanmıştır. Son teknoloji topoloji eşleme veri yolu yönlendiricileriyle karşılaştırıldığında, önerilen modelin yönlendirme kalitesini önemli ölçüde arttırdığı ve karşılaştırılabilir yürütme süresiyle boşluk ihlallerinin sayısını azalttığı görülmüştür (Hsu vd., 2021).

Literatür taramasında, YTA oluşturma ve blockzincir teknolojilerinin avantajlarını harmanlayarak yeni bir blockzincir destekli Ağ İletişimi Hizmet Kalitesi özellikli Internet

Servis Sağlayıcılar (ISS) arası yönlendirme çerçevesi, QoSChain (QC) sunmaktadır. QC çerçevesi, YTA etkinleştirilmiş ISS'ler arasında Ağ İletişimi Hizmet Kalitesi sağlamada merkezi araçları ortadan kaldırarak, karşılıklı olarak güvensiz katılımcılarla bir koordinasyon çerçevesi sunar. Fakat burada da blok oluşturma süresi, mutabakat protokolü ölçeklendirilebilirliği ve blokzincir yol güncelleme ile bu duruma bağlı olarak ileri süreçte yaşanacak depolama sorunları gibi bazı handikaplara sahiptir (Karakus vd., 2021; Karakus ve Guler, 2020).

Yine başka bir araştırma, yüksek düzeyde iletişim ağının kullanımını daha efektif hale getirmek için büyük blokları küçük bloklara ayırma ve güvenliği sağlamak içinde içine Nakamoto zinciri yerleştirilmiş YDG mimarisi fikri üzerine çalışılmıştır. Amaç bir işlemin birden çok madenci tarafından işlenme olasılığını azaltarak işleme verimliliğini artırmaktır. Burada da yine çok miktarda verinin ortaya çıkması ve bu doğrultuda insanların depolama ve gerekli bant genişliği sağlamaya teşvik eden herhangi bir ekonomik modelin olmaması sorundur (Podili vd., 2022).

Makalede Yönlendirilmiş Döngüsüz Graf temelli blokzincir yapısının, blok zincilerde blok sırasının belirlemenin ve paralel bloklar oluşturma'nın güvenlik ve verimlilik açısından zor olduğundan bahsederek Yönlendirilmiş Döngüsüz Graf blokzincir mimarisi öne sürülmüştür. Bu modelde öncelikle blok ağırlığına dayalı yeni bir en ağır zincir kuralı, parent ve uncle blokların seçimine rehberlik etmesi için önerilmiştir. İkinci olarak mesaj fazlalığını azaltıp konsensüs verimliliğini arttırmak için ağaç tabanlı bir dedikodu protokolü önermişlerdir. Sonuç olarak rastgele dedikodu protokolü ile karşılaştırıldığında bu modelin iletişim fazlalığını azalttığı ve konsensüs verimliliğini arttırabileceği görülmüştür (Li vd., 2023).

### 3. MATERYAL VE METOT

Bu arařtırmada dađıtık dijital defter teknolojisinin ana hedeflerinden biri, güvenilir bir üçüncü tarafa ihtiyaç duymadan, birbirlerine mutlaka güvenmeleri gerekmeyen kullanıcıların güvenli bir altyapı ile etkileşime girmesine izin vermektir. Dađıtık dijital defter teknolojileri, güvenilmeyen taraflardan oluşan ortamlara bir düzeyde şeffaflık, izlenebilirlik ve güvenlik eklemektedir. Dađıtık dijital defter teknolojileri, işlemleri kaydetmek için veri yapıları ve bunları işlemek için bir dizi işlevden oluşmaktadır. Her dađıtık dijital defter, farklı veri modeli ve teknolojileri kullanarak kendini farklılaştırabilir. Bununla birlikte, genel olarak, tüm dađıtık dijital defter yapısı iyi bilinen üç teknolojiye dayanır:

1. Açık anahtar kriptografisi,
2. Dađıtılmış eşler arası ağlar,
3. Konsensüs mekanizmaları.

Her dađıtık dijital defter teknolojisi, farklı kullanım durumlarına ve/veya uygulama alanlarına uyan kendine özgü özelliklere ve avantajlara sahiptir.

*Açık Anahtar Kriptografisi:* Açık anahtarlı kriptografi (veya asimetrik kriptografi), güvenli iletişim ve mahremiyet sağlamak için kullanılan bir kriptografik yaklaşımdır. Asimetrik kriptografi, geleneksel simetrik kriptografiden farklı bir yaklaşım sunar. Temel olarak bu, iki farklı anahtarın kullanılması prensibine dayanmaktadır. Biri genel anahtar, diğeri ise özel anahtar olarak adlandırılır.

Bu bölüm, açık anahtar kriptografisinin temel ilkelerinin ve nasıl çalıştığının ayrıntılı bir açıklamasını sağlamaktadır.

*Açık Anahtar (Public Key) ve Özel Anahtar:* Her katılımcı bir kopya anahtar oluşturur. Genel anahtarlar herkese açık olarak paylaşılırken, özel anahtarlar yalnızca anahtarın sahibi tarafından bilinir ve saklanır.

*Şifreleme:* Alıcının ortak anahtarını kullanarak mesajı şifrelemesidir. Bu şifrelenmiş mesajın şifresi yalnızca alıcının özel anahtarı kullanılarak çözülebilir. Şifrelenmiş mesajınızın şifresini başka hiç kimse çözemez, çünkü yalnızca alıcının özel anahtarı vardır.

*Deşifreleme:* Alıcı, şifrelenmiş mesajın şifresini çözmek için özel anahtarı kullanır. Bu

adım, yalnızca alıcının özel anahtarınıza sahip olduğundan eminseniz gerçekleştirilebilir.

*Dijital İmza:* Bireyler, belgeleri veya mesajları imzalamak için özel anahtarlarını kullanabilir. İmza, bir belgenin veya mesajın size ait olduğunu ve içeriğinin değiştirilmediğini onaylar. İmzalı bir belge veya mesaj, herkesin erişebileceği bir genel anahtar kullanılarak doğrulanabilir.

*Anahtar dağıtımı:* Asimetrik kriptografi, anahtarların merkezi bir otorite olmadan güvenli bir şekilde dağıtılmasına olanak tanır. Herkes açık anahtarı alıcıya gönderebilir ve özel anahtarı gizli tutabilir.

*Hız ve güç:* Asimetrik şifreleme genellikle simetrik şifrelemeden daha yavaştır. Bu nedenle, genellikle küçük miktarlardaki verileri şifrelemek veya anahtarları güvenli bir şekilde paylaşmak için kullanılır. Simetrik şifreleme genellikle büyük miktarda veri için tercih edilir.

Açık anahtarlı şifreleme, güvenli İnternet iletişimi, dijital imzalar, kimlik doğrulama ve daha fazlası için kullanılır. En iyi bilinen açık anahtarlı şifreleme algoritmalarından biri, RSA (Rivest-Shamir-Adleman) algoritmasıdır. Diğer yaygın algoritmalar arasında Diffie-Hellman anahtar değişimi ve Eliptik Eğri Kriptografisi (ECC) yer alır.

Açık anahtarlı kriptografi, veri güvenliği ve veri koruması için temel bir araçtır ve modern iletişim sistemlerinin temelini oluşturur.

*Dağıtılmış Eşler Arası Ağlar:* Dağıtılmış eşler arası ağ, bilgisayarların veya cihazların merkezi bir sunucuya bağlanmadan birbirleriyle doğrudan iletişim kurduğu bir ağ yapısıdır. Bu tür ağlarda merkezi bir otorite veya sunucu bulunmamakta, veriler ve işlemler cihazlar arasında dağıtılmaktadır. Dağıtılmış eşler arası ağlar, dosya paylaşımı, içerik dağıtımı, veri tabanı yönetimi, iletişim ve diğer birçok uygulama için kullanılabilir.

Bu bölüm, dağıtılmış eşler arası ağları ayrıntılı olarak açıklamaktadır.

*Merkezi otorite eksikliği:* Geleneksel ağlardan farklı olarak, dağıtılmış eşler arası ağlarda merkezi bir sunucu veya merkezi otorite yoktur. Her cihaz aynıdır ve benzersiz bir işlevi yerine getirir.

*Kaynak paylaşımı:* Dağıtılmış eşler arası ağlar, örneğin dosya paylaşımı yoluyla kaynakları

etkili bir şekilde paylaşmak için kullanılır. Kullanıcılar, merkezi sunuculardaki yükü azaltarak dosyalarını başkalarıyla doğrudan paylaşabilir.

*İçerik Dağıtımı:* Kullanıcılara hızlı ve etkili teslimat için büyük dosyaları ve içeriği birçok cihaza dağıtmak gerekmektedir. Bu, özellikle video akışı gibi uygulamalar için kullanışlıdır.

*Veri tabanı Yönetimi:* Dağıtılmış veri tabanları, verileri birden fazla cihazda depolamak ve yönetmek için kullanılır. Bu, veri tabanı yedeklemesini, güvenliğini ve erişilebilirliğini geliştirir.

*Güvenlik ve ölçeklenebilirlik:* Dağıtılmış eşler arası ağlar, merkezi sunucu tabanlı ağlara kıyasla daha fazla ölçeklenebilirlik ve dayanıklılık sunabilir. Aynı zamanda, bazı merkezi olmayan ağ yapıları, veri merkezleri tek bir yerde yoğunlaşmadığı için veri güvenliğini artırabilir.

*Zorluklar:* Dağıtılmış eşler arası ağlar da bazı zorluklar oluşturabilir. Örneğin, ağdaki cihazlar arasında güveni sürdürmek zor olabilir ve veri bütünlüğünün korunması gerekebilir. Aynı zamanda, ağdaki cihazların dinamik davranışı ve hız farklılıkları yönetimi karmaşıktırabilir.

*Altyapı türleri:* Dağıtılmış uçtan uca ağların farklı altyapı türleri vardır. Tamamen merkezi olmayan yapılar, yarı merkezi yapılar ve açık eşler arası ağlar gibi farklı yapılar vardır. Yaygın merkezi olmayan eşler arası ağların örnekleri arasında BitTorrent (dosya paylaşımı), blokzincir ağları (kripto para ticareti ve akıllı sözleşmeler) ve çeşitli merkezi olmayan içerik dağıtım ağları bulunur.

*Konsensüs Mekanizmaları:* Mutabakat mekanizması, dağıtılmış bir sistemdeki farklı cihazların veya katılımcıların aynı veri veya durum üzerinde anlaşmasını sağlayan bir kurallar bütünü olarak protokol yapısıdır. Mutabakat mekanizmaları, özellikle blokzincir ve diğer birçok uygulama gibi dağıtık defter teknolojilerinde kullanılmaktadır (Antal vd., 2021). Aynı bilgi önemliyse, tüm taraflar aynı fikirde olmalıdır. Konsensüs mekanizmasının ayrıntılı bir açıklaması aşağıdadır.

*Merkezi olmayan karar verme:* Konsensüs mekanizmaları, merkezi bir otoritenin yokluğunda cihazlar veya katılımcı grupları arasında fikir birliğine varmayı amaçlar. Bu, katılımcıların ana veya merkezi sistemde bulunmadığı dağıtılmış sistemler için önemlidir.

*İşlem Onayı ve Blok Oluşturma:* Blokzincir benzeri bir sistemde, yeni işlemlerin onaylanmasını ve blokların eklenmesini bir mutabakat mekanizması yönetir.

Katılımcılar, hangi işlemlerin geçerli olduğu ve hangi işlemlerin bir sonraki bloğa dahil edileceği konusunda anlaşmalıdır.

*Güvenilirlik:* Tüm katılımcıların aynı bilgilere sahip olduğundan ve verilerin değiştirilmediğinden emin olmak için bir fikir birliği mekanizması kullanılması gerekmektedir. Bu, sistem güvenilirliğini sağlar.

*Farklı Mutabakat Algoritmaları:* Farklı mutabakat mekanizmaları vardır (Cao vd., 2020). Bunlar, İş İspatı - Proof of Work (PoW), Hisse İspatı - Proof of Stake (PoS), Pratik Bizans Hata Toleransı (PBFT), Raft ve diğer birçok algoritmayı içerir. Her algoritma fikir birliğine varmak için farklı bir yaklaşım kullanır.

*Proof of Work (PoW):* Bitcoin gibi kripto para birimlerinin temelini oluşturan bu algoritma, katılımcıların yüksek hesaplama gücü gerektiren matematik problemlerini çözmelerini gerektirir. İlk doğru çözücü yeni bloklar ekleyebilir. Daha güvenli olmasına rağmen, enerji tüketimi daha yüksektir.

*Proof of Stake (PoS):* PoS algoritmasında blok oluşturma gücü, katılımcıların elindeki kripto para miktarına göre belirlenir. Bu, enerji tüketimini azaltacak, ancak zengin katılımcılar üzerinde daha iyi kontrol sahibi olmanızı sağlayacaktır.

*Pratik Bizans Hata Toleransı (PBFT):* Bu algoritma hataya dayanıklı sistemlerde kullanılır. Katılımcılar, işlemi onaylamak için belirli bir süre içinde anlaşılır.

*Raft:* Raft, dağıtık sistemlerde fikir birliğine varmak için tasarlanmış bir algoritmadır. Daha basit yapı, lider seçimi, protokol senkronizasyonu ve düğüm değiştirme gibi işlemleri güvenli bir şekilde yönetir.

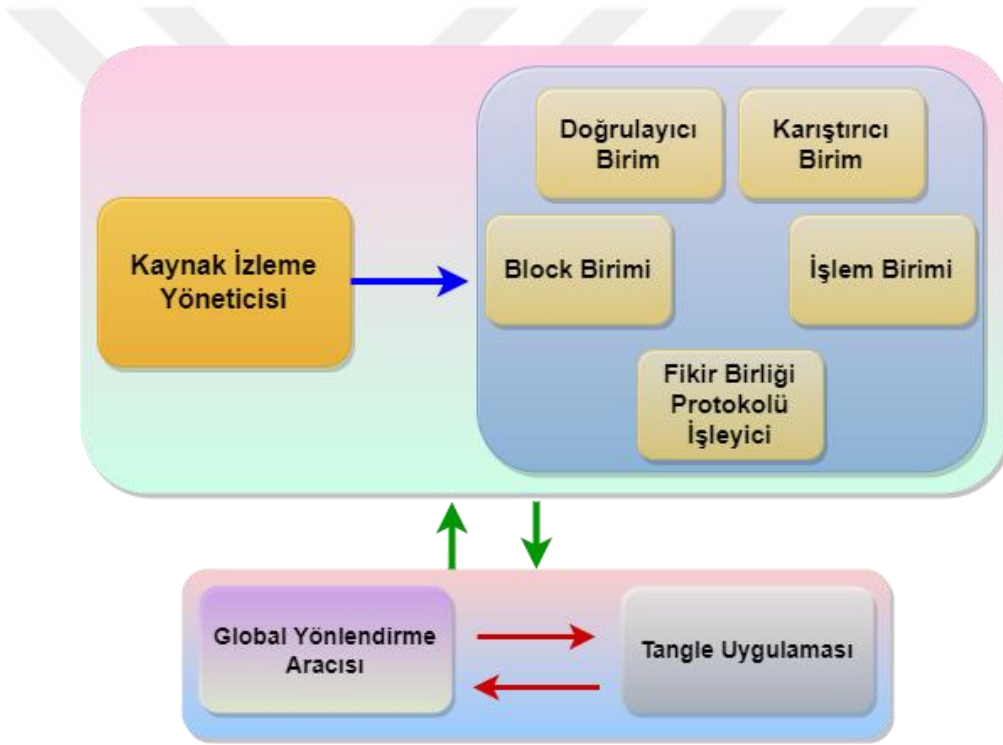
Mutabakat mekanizmaları, dağıtılmış sistemlerin güvenilirliğini, güvenliğini ve veri bütünlüğünü sağlamak için önemlidir. Hangi algoritmayı kullanacağınız, uygulama gereksinimlerinize ve hedeflerinize bağlıdır.

Çoklu alan ağlarında uçtan uca servis kalitesi destekli blokzincir tabanlı yol bulma çerçevesinde farklı dağıtık dijital defter teknolojilerinin uygulanması, veri akış kurulum

süresi, değiştirilen ve işlenen mesajlar gibi ölçeklenebilirlikle ilgili anahtar performans göstergeleri açısından blokzincir tabanlı çoklu ağlarda kullanılan yol bulma çerçevesinin dağıtık dijital defter tabanlı/yönelimli performanslarını karşılaştırma fırsatı verecektir. Böyle bir karşılaştırma, güvenilmeyen taraflarla hizmet kalitesi tabanlı akıllı sistemler arası yönlendirme koordinasyon çerçevesi için en verimli dağıtık dijital defter teknolojilerini belirlememize olanak sağlayabilir.

### 3.1. Sistem Modeli

Bireysel olarak tanımlı sistem denetleyicileri, tangle ağına katılan düğümleri temsil eder. Uygulanmak istenen sistemde Yazılım Tanımlı Ağ (YTA) denetleyicisi tangle'ın eşsiz yetenekleri ile uyarlanmış denetleyici modül ve ağ uygulamaları ile geliştirilmektedir.

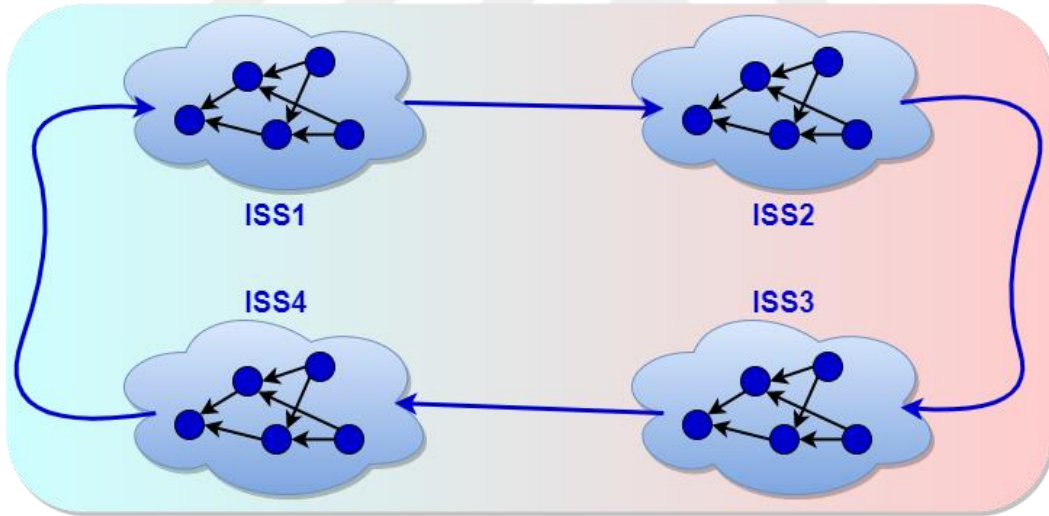


Şekil 9. Tangle Alt Birimleri

YTA denetleyicisinin modülleri Şekil 9'da gösterildiği gibi tangle yöneticisi ve alt bileşenlerdir. Oluşturulan bu modül özerk sistemde tangle ağı ile ilgili tüm işlevleri yürütmektedir. Doğrulayıcı birim, tangle ağındaki düğüm doğrulama kurallarına dayalı olarak diğer kontrolörlerden gelen düğümleri doğrulamaktan sorumludur. Karıştırıcı birim, tangle ağına gönderilecek işlemlerin ve düğümlerin şifrelenmesinden sorumlu olan birimdir. İşlem birimi, işlemleri oluştururken, düğüm birimi fikir birliği protokolüne bağlı olarak düğüm oluşturmaktan sorumludur. Mutabakat protokolü işleyicisi, sıradaki işlemler

ve düğümler üzerinde anlaşmaya varmak için fikir birliği algoritmalarını uygulamaktadır. Kaynak işleme yöneticisi, bant genişliği, gecikme, titreşim vb. gibi ağ kaynaklarını düzenli olarak izler ve bir değişiklik olması halinde tangle yöneticisi modülüne ilgili işlemleri düzenlemesi talimatını verir. Global yönlendirme aracı birimi, denetleyiciye bir özerk sistemler arası hizmet talebi geldiğinde özerk sistemler arası yönlendirme işlevini uygulamaktadır. Hizmet talebi geldiği zaman uçtan uca yolu bulmak amacıyla tangle ağındaki yeni işlemlerin giriş ve çıkış alanlarını giriş parametreleri olarak alır ve tangle uygulamasını çağırır. Tangle uygulaması ayrıca ağa düğüm göndermek ve almaktan, tangle yöneticisi ile iletişim halinde olarak hizmet-yol taleplerini işlemekten sorumludur.

Birbirleri ile veri alışverişinde bulunmak zorunda olan otonom sistemlerde ilgili verileri iletmek için yollar oluşturulması gerekmektedir. Otonom sistemlerdeki iletişim yolu tangle ağ yapısı ile sağlanabilir ve bilgiler tangle düğümlerinde tutulabilir. Bu düğümler sadece istenilen bilgileri iletmekle birlikte otonom sistem içinde bulunan diğer bilgilerin korunmasını da sağlamaktadır.



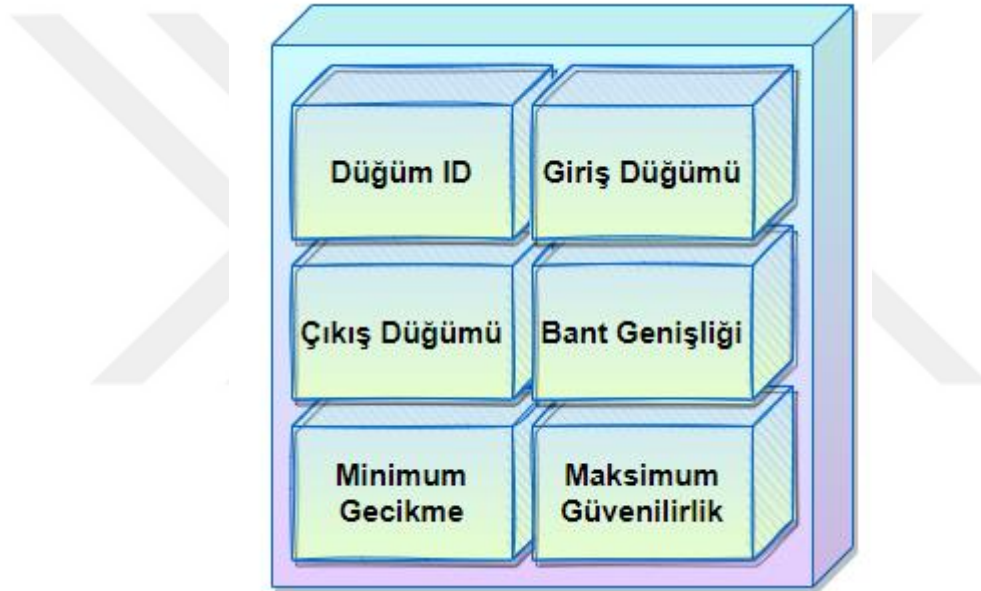
Şekil 10. Çoklu Alan Ağları Mimarisi

Şekil 10’da belirtilen alan ağlarında görüldüğü üzere Internet Servis Sağlayıcı (ISS) ISS1 ‘den ISS3’e veri aktarımı yapılmak isteniyor. Bu senaryoda ISS1 ‘den ISS3’e veri aktarımı ISS1 - ISS2 - ISS3 yönlü olabileceği gibi ISS1 - ISS4 - ISS3 yönlüde olabilir. Bu ters yönlü mimari eş zamanlı olarak denetleyicilerde bulunmaktadır. Burada iletişimi sağlayacak olan ISS2 ya da ISS3’lerdeki tangle ağında bilgiler korunacak ve gizliliği sağlanacaktır. Olası uçtan uca yollar ile ilgili ağ genişliği, gecikme süresi ve güvenilirlik bilgileri hesaplanabilir olacaktır ve yol uzunluğu gibi diğer bilgiler ISS’lerde gizli

tutulacaktır. Tangle düğümleri iletişim için belirli bilgileri içerisinde tutmalıdır.

Uçtan uca bulunması talep edilen hizmet kalitesi destekli yol bulunma işleminde ISS'lerin giriş düğümlerinden uç noktalara gidecek yolun hesaplanması ise özerk olarak sistemi yöneten sorumlu ISS denetleyici tarafından bulunarak diğer ISS'ler tarafından ağ içi yol bilgisi bilinmeyecektir.

Yönlendirilmiş Döngüsüz Graf modeli olarak oluşturulmuş tangle ağında, ISS'lerin yönetiminde tutulan her bir düğümde Şekil 11'da belirtilmekte olan bilgiler tutulmaktadır.



Şekil 11. Her bir Düğüm İçinde Tutulacak Bilgiler

Her bir düğüm için tutulacak bilgiler sırasıyla:

*Düğüm ID:* Her bir düğüm için ISS tarafından atanmış eşsiz bir değer olarak belirlenmektedir ve diğer ISS'ler tarafından bu değer ile düğüm adlandırılmaktadır.

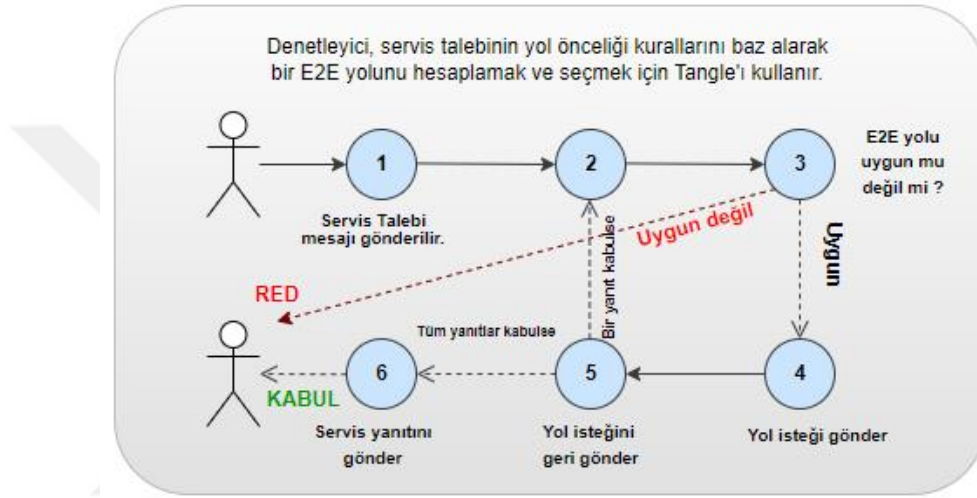
*Giriş Düğümü:* Her bir ISS için başka bir ISS'den özerk ISS'nin ağına bağlantı kurulmasını sağlayan düğümdür.

*Çıkış Düğümü:* Her bir ISS'den başka bir ISS'e bağlantıyı sağlayan düğüm olarak belirtilmektedir.

*Bant Genişliği:* Giriş ve çıkış düğümleri arasında sorumlu ISS tarafından belirlenen yoldaki bağlantıların sağlayabildiği minimum bant genişliğini ifade etmektedir.

*Minimum Gecikme:* Giriş ve çıkış düğümleri arasında sorumlu ISS tarafından belirlenen yoldaki bağlantıların sağlayabildiği toplam gecikme durumunu ifade etmektedir.

*Maksimum Güvenilirlik:* Giriş ve çıkış düğümleri arasında sorumlu ISS tarafından belirlenen yoldaki bağlantıların sağlayabildiği maksimum güvenilirlik değerini ifade etmektedir.



Şekil 12. YDG Tabanlı Çoklu Alan Ağlarında Uçtan Uca Yol Bulma Mimarisi

Bu çalışmada YDG ile geliştirilmiş hizmet kalitesi tabanlı uçtan uca yol seçimli modelin iş akışını Şekil 12’de temsil edilmektedir.

Adım 1’de kullanıcıdan hizmet talebi alındıktan sonra, denetleyici kendi ağının sınır düğümünden (çıkış düğümü) hedef ağın sınır düğümüne (giriş düğümü) uçtan uca giden yolu hesaplamak için tangle kullanır (Adım 2). Bunu yaparken hizmet talebinde belirtilen hizmet kalitesi gereksinimleri ve yol seçimi önceliklerini dikkate alırlar. Adım 3’te, eğer uygun bir hizmet kalitesini destekleyen yol bulunamıyorsa, kullanıcıya bir RET mesajı göndererek koşulları sağlayan uçtan uca herhangi bir yol mevcut olmadığı için hizmet talebini reddedilir. Hizmet talebinin gerekli ağ iletişimi hizmet kalitesi parametreleri sağlandığında ise kaynak ağ denetleyicisi yazılım tanımlı ağ alt yapısını kullanarak belirlenen uçtan uca yoldaki yol içeren her ağ denetleyicisine istek mesajı göndermeye başlar. Adım 4’te yol isteği mesajlarını aldıktan sonra tüm yollar hizmet kalitesi gereksinimlerini karşılayabiliyorsa Adım 5’te uçtan uca yol üzerinden kullanılacak uygun

yolun belirlenmesiyle birlikte, her bir ağ denetleyicisine tüm servis yanıtları gönderilmektedir. Adım 6’da ilgili ağ iletişimi hizmet kalitesi gereksinimlerine dayalı olarak hizmet talebine KABUL mesajı geri gönderilir. Uçtan uca yol üzerinde herhangi bir ağ denetleyicisi, kaynak ağ denetleyicisine bir RED yanıtı gönderirse, kaynak denetleyicisi aynı gereksinimlere sahip başka bir uçtan uca yol aramaya başlar.



## 4. BULGULAR VE TARTIŞMA

### 4.1. Bulgular

#### 4.1.1. Hizmet Talebinde Bulunulması

Bu noktada basit bir Yönlendirilmiş Döngüsüz Graf (YDG) düğümünü kodlandı ve bir hizmet talebinde bulunuldu. Çalışma sisteminin oluşturulması ve yazılım modüllerinin hazırlanması aşağıdaki kodlama yapılarında belirtilmektedir.

```
import java.util.ArrayList;
import java.util.List;

// Basit bir DAG service request kodu örneği

class DAGNode {
    private String fullPath;
    private List<DAGNode> children;
    private int nodeID;
    private DAGNode ingress;
    private DAGNode egress;
    private double maxBandwidth;
    private double minDelay;
    private double maxReliability;

    public DAGNode(int node) {
        this.data = "";
        this.children = new ArrayList<>();
        this.nodeID = node;
        this.ingress = new DAGNode(0);
        this.egress = new DAGNode(0);
        this.maxBandwidth = 0;
        this.minDelay = Double.Infinity;
        this.maxReliability = 0;
    }

    public void setNodeID(int node){
        this.nodeID = node;
    }

    public int getNodeID(){return this.nodeID;}

    public void setIngress(DAGNode ingress){
        this.ingress = ingress;
    }

    public DAGNode getIngress(){return this.ingress;}
```

Şekil 13. DAGNode mimarisi başlangıç yapısı

Bu kod Yönlendirilmiş Döngüsüz Graf modeli ile yapılan bir hizmet talebini temsil eder. Kodun işleyişinden bahsedecek olursak eğer, DAGNode Sınıfı içerisinde:

fullPath: Düğümün tam yolunu (ismini) tutan bir String değişkeni.

children: Düğümün çocuk (child) düğümlerini tutan bir List koleksiyonu.

nodeID: Düğümün benzersiz kimliğini tutan bir tamsayı değişkeni.

ingress: D ğ m n giriř d ğ m n  (parent d ğ m n ) tutan bir DAGNode nesnesi.

egress: D ğ m n  ıkıř d ğ m n  (child d ğ m n ) tutan bir DAGNode nesnesi.

maxBandwidth: D ğ m arasındaki maksimum bant geniřliđini tutan bir ondalık sayı deđiřkeni.

minDelay: D ğ m arasındaki minimum gecikme s resini tutan bir ondalık sayı deđiřkeni.

maxReliability: D ğ m arasındaki maksimum g venilirliđi (reliability) tutan bir ondalık sayı deđiřkeni.

T m bu tanımlanan deđiřkenler u tan uca yol bulma iřleminde internet servis sađlayıcıların giriř ve  ıkıř d ğ mlerinin bađlantılı olduđu yol bilgilerini tutmaktadır.

```
public void setEgress(DAGNode egress){
    this.egress = egress;
}

public DAGNode getEgress(){return this.egress;}

public void setMaxBandwidth(double bandwidth){
    this.maxBandwidth = bandwidth;
}

public double getMaxBandwidth(){return this.maxBandwidth;}

public void setminDelay(double minDelay){
    this.minDelay = minDelay;
}

public double getminDelay(){return this.minDelay;}

public void setmaxReliability(double minDelay){
    this.maxReliability = maxReliability;
}

public double getmaxReliability(){return this.maxReliability;}

public void addChild(DAGNode childNode) {
    this.children.add(childNode);
}

public String getData() {
    return this.data;
}
```

řekil 14. DAGNode mod l  fonksiyonları

**AddChild Metodu:** Bu metod, mevcut düğüme bir çocuk (child) düğüm yani yeni bir internet servis sağlayıcı tarafından eklenen servis sağlayıcının kendi ağ alanındaki yeni bir yolu eklemek için kullanılır. Yani, mevcut düğüm çocuk (child) düğümleri arasına yeni bir düğüm eklenir.

**Getter ve Setter Metodları:** Bu metodlar, DAGNode sınıfının özelliklerine erişmek ve onları güncellemek için kullanılır.

**DAGServiceRequestExample Sınıfı:** Bu sınıf, YDG yapısındaki düğümleri oluşturur ve service request işlemini başlatır.

**ProcessRequest Metodu:** Bu metod, service request işlemlerini simüle eder ve her düğümdeki işlemi gerçekleştirir.

**Main Metodu:** Bu metod, örnek bir YDG yapısı oluşturur ve processRequest metodunu çağırarak service request işlemini başlatır.

Kodun temel amacı, DAGNode sınıfını ve bu sınıftan oluşturulan düğümleri kullanarak basit Yönlendirilmiş Döngüsüz Graf (YDG) yapısı oluşturmak ve bu yapı üzerinde bir service request işlemi simüle etmektir.

Kod çalıştığında, program hizmet isteklerini yürütür ve her düğümün adını ekrana yazdırır. Tüm bu işleyişin yapısı ve sırası aşağıdaki gibidir.

İlk olarak, DAGNode sınıfı tanımlanır. Bu sınıf, YDG yapısındaki düğümleri temsil eder ve her düğümün üzerinde bazı özelliklerin saklanması sağlar. DAGNode sınıfında fullPath, children, nodeID, ingress, egress, maxBandwidth, minDelay, maxReliability gibi değişkenler ve bunlara erişmek için getter ve setter metotlar bulunur. “addChild” metodu, bir düğüme yeni bir çocuk (child) düğüm eklemek için kullanılır. Children listesine yeni bir düğüm eklemek için bu metod kullanılır. DAGServiceRequestExample sınıfı, ana sınıftır ve programın başlangıcını belirtir. ProcessRequest metodu, service request işlemi simüle eder. Ancak, şu anki haliyle sadece düğümün adını ekrana yazdıran sistem gerçek bir servis isteği işlemi bu metoda eklenerek düğümlerde gerçek işlemler yapılabilir. Şekil 16’da belirtilen “ShortestPathDAG” modülü ile birlikte bant genişliği ve uçtan uca gecikmeyi göz önünden bulunduran kısıtlama destekli Dijkstra en kısa yol bulma algoritması tanımlanmış ve main metodu ile programın çalıştırılmasını sağlar. Burada örnek bir YDG yapısı oluşturulur ve processRequest metodu ile service request işlemi

başlatılır.

DAGNode sınıfında oluşturulan düğümler ve aralarındaki bağlantılar şu şekildedir:

nodeA: "Node A" adında bir düğüm oluşturulur.

nodeB: "Node B" adında bir düğüm oluşturulur.

nodeC: "Node C" adında bir düğüm oluşturulur.

nodeD: "Node D" adında bir düğüm oluşturulur.

```
public List<DAGNode> getChildren() {  
    return this.children;  
}  
}  
  
public class DAGServiceRequestExample {  
    public static void processRequest(DAGNode node) {  
        // Service request işlemleri burada yapılır  
        System.out.println("Service request işlemi gerçekleştirildi: " + node.getData());  
    }  
  
    public static void main(String[] args) {  
        // DAG düğümlerini oluşturma  
        DAGNode nodeA = new DAGNode("Node A");  
        DAGNode nodeB = new DAGNode("Node B");  
        DAGNode nodeC = new DAGNode("Node C");  
        DAGNode nodeD = new DAGNode("Node D");  
  
        // DAG yapısını oluşturma  
        nodeA.addChild(nodeB);  
        nodeA.addChild(nodeC);  
        nodeB.addChild(nodeD);  
        nodeC.addChild(nodeD);  
  
        // Service request işlemi  
        processRequest(nodeA);  
    }  
}
```

Şekil 15. DAGServiceRequestExample kod bloğu

nodeA düğümlerine nodeB ve nodeC düğümleri eklenir. Bu şekilde nodeA, nodeB ve nodeC düğümlerine sahip olur. nodeB düğümlerine nodeD düğümleri eklenir. Bu şekilde nodeB, nodeD düğümlerine sahip olur. nodeC düğümlerine de nodeD düğümleri eklenir. Bu şekilde nodeC, nodeD düğümlerine sahip olur.

"ShortestPathDAG" başlıklı kod bloğunda, bahsedilen yöntem ShortestPathDAG yöntemidir. Mevcut yaklaşım, hem gecikme hem de bant genişliğini dikkate alarak, Yönlendirilmiş Döngüsüz Graf (YDG) bir başlangıç düğümünden bir hedef düğüme giden en verimli rotayı hesaplar. Aşağıda operasyonel sürecin sıralı bir analizi yer almaktadır:

Süreçteki ilk adım gerekli veri yapılarını başlatmaktır:

DelayMap ve bandwidthMap haritaları, çaprazlama süreci boyunca her düğüm için kümülatif gecikme ve minimum bant genişliği değerlerini tutmak için kullanılır.

ParentMap, en kısa yoldaki her düğüm için ana düğümün kaydını tutan bir veri yapısıdır.

MinDelayQueue, işlem için en düşük gecikmeye sahip düğümü seçmek amacıyla kullanılan bir öncelik kuyruğudur.

Gecikme ve bant genişliği haritaları başlatılır. Kaynak düğümün gecikmesi sıfır olarak ayarlanmalıdır, ancak bant genişliğine yüksek bir sayı atanmalıdır, bu da sınırsız bir bant genişliği kapasitesine işaret eder. Diğer tüm düğümlerin gecikme ve bant genişliği sonsuz olarak başlatılır. Kaynak düğüm öncelik kuyruğuna dahil edilmelidir.

Birincil iterasyon (Dijkstra Algoritması):

- Öncelik kuyruğundan `current` olarak etiketlenmiş bir düğüm alınır.
- Güncel olarak etiketlenmiş düğümden kaynaklanan her giden kenar için: yeni gecikme ve yeni bant genişliği, mevcut düğümün birikmiş gecikme ve bant genişliğinin yanı sıra kenarla ilişkili gecikme ve bant genişliği dikkate alınarak hesaplanabilir.

Yeni gecikmenin hedef düğümün mevcut gecikmesinden daha az olması veya gecikmelerin eşit ancak yeni bant genişliğinin daha fazla olması durumunda, gecikme ve bant genişliği haritalarının güncellenmesi, üst haritanın güncellenmesi ve hedef düğümün öncelik kuyruğuna dahil edilmesi gerekir.

En kısa yolu belirlenmesi, en kısa yoldaki düğümleri yeniden oluşturmak için, hedef düğümden başlanmalı ve `parentMap` kullanılarak yol geriye doğru izlenmelidir. Sonuçları görüntülemek için ise, en kısa yolda bulunan düğümlerin yanı sıra söz konusu yolla ilişkili toplam gecikme ve minimum bant genişliğinin yazdırılmasını içerir.

```

import java.util.*;

public class ShortestPathDAG {
    public static void shortestPathDAG(DAGNode source, DAGNode destination) {
        Map<Node, Integer> delayMap = new HashMap<>();
        Map<Node, Integer> bandwidthMap = new HashMap<>();
        Map<Node, Node> parentMap = new HashMap<>();
        PriorityQueue<Node> minDelayQueue = new PriorityQueue<>(Comparator.comparingInt(delayMap::get));

        // Initialize delay and bandwidth maps
        for (Node node : nodes) {
            delayMap.put(node, Integer.MAX_VALUE);
            bandwidthMap.put(node, Integer.MAX_VALUE);
        }
        delayMap.put(source, 0);
        bandwidthMap.put(source, Integer.MAX_VALUE);

        minDelayQueue.offer(source);

        while (!minDelayQueue.isEmpty()) {
            Node current = minDelayQueue.poll();
            for (Edge edge : current.edges) {
                int newDelay = delayMap.get(current) + edge.delay;
                int newBandwidth = Math.min(bandwidthMap.get(current), edge.bandwidth);
                if (newDelay < delayMap.get(edge.target) ||
                    (newDelay == delayMap.get(edge.target) && newBandwidth > bandwidthMap.get(edge.target))) {
                    delayMap.put(edge.target, newDelay);
                    bandwidthMap.put(edge.target, newBandwidth);
                    parentMap.put(edge.target, current);
                    minDelayQueue.offer(edge.target);
                }
            }
        }

        // Construct the path from destination to source
        List<Node> path = new ArrayList<>();
        Node current = destination;
        while (current != null) {
            path.add(current);
            current = parentMap.get(current);
        }
        Collections.reverse(path);

        // Print the shortest path and its metrics
        System.out.println("Shortest Path: " + path);
        System.out.println("Total Delay: " + delayMap.get(destination));
        System.out.println("Minimum Bandwidth: " + bandwidthMap.get(destination));
    }
}

```

Şekil 16. YDG üzerinde kısıtlamalı Dijkstra algoritmasının mimarisi

Bu oluşturulan modülün çalışmasının test edilebilmesi için, “Main” yönteminde, bir YDG oluşturmak için düğümler ve kenarlar örneklenir. Bu bölümün özelleştirilmesi, eldeki benzersiz grafik yapısına ve girdi verilerine göre uyarlanabilir. “ShortestPathDAG” fonksiyonuna, kaynak ve hedef düğümleri parametre olarak geçirilerek `shortestPathDAG`

yöntemi çağrılır. Mevcut yaklaşım, hem gecikme süresini hem de bant genişliğini dikkate alarak en verimli rotayı hesaplar ve çıktısını verir.

Uygulamanın, dikkate alınan YDG olduğunu göz önünde bulunduğunu belirtmek önemlidir. Grafiğin döngüler sergilemesi veya belirli kısıtlamalara tabi olması durumunda, kullanılan algoritma ve veri yapılarında uygun ayarlamaların yapılması gerekebilir. Ayrıca, uygulamanızın özel gereksinimlerine bağlı olarak hata işlemeyi dahil etmek ve kodu daha da optimize etmek de gerekli olabilir.

#### **4.1.2.Hizmet Yanıtı:**

Yukarıda yapılan bir servis talebine karşılık verilen servis cevabı aşağıdaki gibidir.

Detaylı bir şekilde anlatacak olursak, DAGNode Sınıfı, YDG yapısındaki düğümleri temsil eder. Her bir düğüm, bir tam sayı ID'si (nodeID), giriş düğümü (ingress), çıkış düğümü (egress), maksimum bant genişliği (maxBandwidth), minimum gecikme süresi (minDelay) ve maksimum güvenilirlik (maxReliability) gibi özelliklere sahiptir. Ayrıca, çocuk (child) düğümleri olan bir liste de bulunur. processRequest() Metodu, hizmet taleplerini işlemek için kullanılır. Main() Metodu programın başlangıç noktasıdır. Örnek bir YDG yapısı oluşturulur ve processRequest() metodu çağrılarak hizmet talepleri işlenir.

YDG yapısını oluşturulurken de main() metodu içinde örnek bir YDG yapısı oluşturulur. Dört düğüm (A, B, C, ve D) oluşturulur ve bu düğümler arasında bağlantılar kurulur. Örnek olarak, nodeA düğümü nodeB ve nodeC'ye bağlıdır; nodeB düğümü nodeD'ye bağlıdır ve nodeC düğümü de nodeD'ye bağlıdır. Bu, basit bir Yönlendirilmiş Döngüsüz Graf (YDG) oluşturur. processServiceRequests() Metodu, hizmet taleplerine göre hizmet yanıtlarını işlemek için kullanılır. BFS (Breadth-First Search) algoritmasını kullanarak, önce kök düğümden başlayarak her düğümün hizmet yanıtlarını işler ve ekrana yazdırır. processResponse() Metodu, her bir düğümün hizmet yanıtlarını işlemek için kullanılır.

Adım adım çalışma şeklini anlatacak olursak önce main() metodu çalıştırılır ve örnek bir YDG yapısı oluşturulur. Sonra processServiceRequests(nodeA) çağrısı yapılır ve işlemler kök düğümden başlayarak gerçekleşir. Daha sonra, nodeA düğümü işlenir ve verileri processResponse() metoduyla ekrana yazdırılır ve nodeB ve nodeC düğümleri işlenir (çocuk düğümleri olduğu için bu adım gerçekleşir). NodeB düğümü işlenir ve verileri ekrana yazdırılır. Ardından, nodeD düğümü işlenir (çocuk düğümü olduğu için bu adım

gerçekleşir). Son olarak, nodeC düğümü işlenir ve verileri ekrana yazdırılır. processServiceRequests() metodu düğümleri sıg öncelikli arama (Breadth-First Search – BFS) algoritmasına göre gezdiği için, çocuk (child) düğümler önce işlenir ve ardından torun düğümler işlenir.

```
import java.util.LinkedList;
import java.util.Queue;

public class DAGServiceResponseExample {

    public static void processResponse(DAGNode node) {
        // Hizmet yanıtı işlemleri burada yapılır
        // Örnek olarak, her bir düğümün verilerini ekrana yazdıralım:
        System.out.println("Node ID: " + node.getNodeID());
        System.out.println("Ingress Node ID: " + node.getIngress().getNodeID());
        System.out.println("Egress Node ID: " + node.getEgress().getNodeID());
        System.out.println("Max Bandwidth: " + node.getMaxBandwidth());
        System.out.println("Min Delay: " + node.getMinDelay());
        System.out.println("Max Reliability: " + node.getMaxReliability());
        System.out.println();
    }

    public static void processServiceRequests(DAGNode rootNode) {
        Queue<DAGNode> queue = new LinkedList<>();
        queue.add(rootNode);

        while (!queue.isEmpty()) {
            DAGNode node = queue.poll();
            processResponse(node);

            for (DAGNode child : node.getChildren()) {
                queue.add(child);
            }
        }
    }

    public static void main(String[] args) {
        // Örnek DAG düğümlerini oluşturma
        DAGNode nodeA = new DAGNode(1);
        DAGNode nodeB = new DAGNode(2);
        DAGNode nodeC = new DAGNode(3);
        DAGNode nodeD = new DAGNode(4);

        // DAG yapısını oluşturma
        nodeA.addChild(nodeB);
        nodeA.addChild(nodeC);
        nodeB.addChild(nodeD);
        nodeC.addChild(nodeD);

        // Hizmet taleplerine göre hizmet yanıtı işlemi
        processServiceRequests(nodeA);
    }
}
```

Şekil 17. Servis Cevap Modülünün Oluşturulması

## 4.2.Tartışma

Bu tez, yazılım tanımlı ağlar (YTA) tarafından desteklenen dağıtılmış dijital defter teknolojisi üzerine araştırmayı açıklamaktadır. Günümüzün hızla gelişen teknolojisi ve küreselleşmesi, iletişim ağlarının karmaşıklığını artırmış ve dağıtık ağ modellerinin kullanımını yaygınlaştırmıştır. Ağların büyümesi ve çeşitli alanlarda kullanılması veri alışverişinde gizlilik, güvenlik ve iletişim kalitesi gibi önemli konuları gündeme getirmiştir.

Bu tezde, yazılım tanımlı çoklu ağ mimarisinde uçtan uca hizmet kalitesi desteği ve merkezi olmayan dijital cüzdan teknolojilerinden biri olan Yönlendirilmiş Döngüsüz Grafik (YDG) modeline dayalı bir yol bilgi işlem mimarisinin oluşturulması anlatılmaktadır. YDG modeli, düğümlerdeki verileri tek yönde grafik biçiminde aktararak ölçeklenebilirliği artıran önemli bir yapıya sahiptir. Önerilen mimari, YDG sayesinde merkezi bir aracıyı ortadan kaldırarak paralel ve ölçeklenebilir yol hesaplaması sağlamayı amaçlamaktadır.

Birçok gerçek dünya uygulaması, çok sayıda taraf ve kişi arasında büyük miktarda veri yayar. Bu veri alışverişinde önemli olan verilerin doğrudan aktarılmaması ve sadece gerekli iletişimin sağlanmasıdır. YDG modeli, YTA'nın esnek yapısı ve Ağ İletişim Kalitesi (QoS) ile güçlü bir ağ modelidir. YTA, veri ve kontrol düzlemlerini ayırarak ağı daha basit ve daha otonom hale getirirken QoS, uygulama önceliklendirme nedeniyle kaybedilen zamanı azaltır. Bu yapı, paket kesintileri, gecikmeler veya dalgalanmalar olmadan sorunsuz paket aktarımı sağlar.

Geleneksel ağ mimarileri, karmaşık ağ yapıları oluşturmak için çeşitli cihazlar ve karmaşık protokoller kullanır. Bu, sistem entegrasyonunu ve yönetimini zorlaştırır. YTA, bu karmaşıklığı ortadan kaldırarak standartlaştırılmış protokoller kullanılarak merkezi olarak yönetilebilen dinamik, yüksek performanslı ve güvenli bir ağ modeli sağlar. YTA'nın esnek yapısı ve YDG modelinin tek yönlü olması, birden fazla ağ katılımcısı arasındaki hassas bilgilerin korunmasında ve hızlı yol hesaplama süresi elde edilmesinde büyük avantajlar sağlar.

Sonuç olarak, YTA tarafından desteklenen YDG modeli, dağıtık dijital defter teknolojisi ve çoklu ağ yapıları için önemli bir gelecek çözümü sunmaktadır. Bu yaklaşım, en üst düzeyde güvenlik ve iletişim kalitesini korurken, taraflar arasında veri alışverişini

sağlamak için etkili bir çözüm sunar. Özellikle tedarik zinciri yönetimi, Nesnelerin İnterneti (IoT) ve merkezi olmayan finansal platformlar gibi uygulamalarda paralel ve ölçeklenebilir yol bilgi işlem mimarilerinin kullanılması, ağların daha verimli ve güvenli yönetimine katkıda bulunur. Bu doğrultuda YTA destekli YDG modelinin pratik uygulamalarda yaygın olarak kullanılması beklenmektedir.

## 5. SONUÇ VE ÖNERİLER

Tasarlayacağımız YDG modelinde düğümler arası iletişim tek taraflı ve tek yönlüdür. Çoklu ağlarda ağ iletişimi hizmet kalitesi, kuralları hızlı bir şekilde karşılamaya çalışırken veri paketlerini iletme esnasında hem kaynak hem de hedef adresleri kullanır. YDG, hem hedef hem kaynak için ayrı ayrı yollar oluşturmak yerine tek bir yönlendirme modeli oluşturur. Böylece en büyük sorunlardan biri olan yüksek yol hesaplama sürelerinin önüne geçilmiş olur ve önemli ölçüde düşük rota hesaplama sürelerine inmek mümkün olur.

Bu modelde uygulanacak tangle ağ yapısının konsensüs mekanizması, ağa katılabilmek için kendinden önce gelen rastgele iki düğümü onaylama koşuluna sahiptir ve bu durum beklemek zorunda kalmadan birden fazla işlemin aynı zamanda doğrulanabilmesini mümkün kılar. Bu doğrultuda her düğüm bir madenci olduğundan bedel ödenmek zorunda kalınan madencilere ihtiyaç duyulmaz, yeni işlem eklemenin maliyeti sıfıra düşer. Bu sayede diğer dağıtık dijital defter teknolojilerinin zorluklarına karşı bir alternatif yaratır.

## KAYNAKLAR

- Al-Hawri, E., Correia, N. ve Barradas, A. (2020). DAG-Coder: Directed Acyclic Graph-Based Network Coding for Reliable Wireless Sensor Networks. *IEEE Access*, 8, 21886–21896. doi:10.1109/ACCESS.2020.2969029
- Antal, C., Cioara, T., Anghel, I., Antal, M. ve Salomie, I. (2021). Distributed ledger technology review and decentralized applications development guidelines. *Future Internet*, 13(3), 1–32. doi:10.3390/fi13030062
- Avallone, S. ve Ashraf, U. (2020). A DAG-Based Forwarding Paradigm for Large Scale Software Defined Networks. *IEEE Transactions on Network and Service Management*, 17(1), 577–591. doi:10.1109/TNSM.2019.2956678
- Bhandary., Mohan Manish Parmar., and D. A. (2020). Graph for IoT Data Security using IoTA Tangle. *IEEE Xplore*, (Icces), 827–832.
- Cao, B., Zhang, Z., Feng, D., Zhang, S., Zhang, L., Peng, M. ve Li, Y. (2020). Performance analysis and comparison of PoW, PoS and DAG based blockchains. *Digital Communications and Networks*, 6(4), 480–485. doi:10.1016/j.dcan.2019.12.001
- Chafjiri, F. S. ve Mehdi Esnaashari Esfahani, M. (2019). An Adaptive Random Walk Algorithm for Selecting Tips in the Tangle. *2019 5th International Conference on Web Research, ICWR 2019*, 161–166. doi:10.1109/ICWR.2019.8765264
- Cho, S., Elhourani, T. ve Ramasubramanian, S. (2010). Resilient multipath routing with independent directed acyclic graphs. *IEEE International Conference on Communications*, 0–4. doi:10.1109/ICC.2010.5502526
- Cicioğlu, M. ve Çalhan, A. (2017). Yazılım Tanımlı Ağlar - Software Defined Networks. *Karaelmas Fen ve Mühendislik Dergisi*, 7(2), 684–695. doi:10.72122Fzkufbd.v7i2.835
- Das, S. K., Datta, A. K. ve Tixeuil, S. (1999). Self-stabilizing algorithms in DAG structured networks. *Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks, I-SPAN*, 190–195. doi:10.1109/ispan.1999.778938
- Ding, Y. ve Sato, H. (2020). Dagbase: A Decentralized Database Platform Using DAG-Based Consensus. *Proceedings - 2020 IEEE 44th Annual Computers, Software, and Applications Conference, COMPSAC 2020*, 798–807. doi:10.1109/COMPSAC48688.2020.0-164
- Deshpande, A., Stewart, K., Lepetit, L. ve Gunashekar, S. (y.y.). Understanding the landscape of Distributed Ledger. [https://www.rand.org/content/dam/rand/pubs/research\\_reports/RR2200/RR2223/RAN](https://www.rand.org/content/dam/rand/pubs/research_reports/RR2200/RR2223/RAN)

D\_RR2223.pdf adresinden erişildi.

- El Ioini, N. ve Pahl, C. (2018). A review of distributed ledger technologies. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11230 LNCS, 277–288. doi:10.1007/978-3-030-02671-4\_16
- Gupta, H. ve Janakiram, D. (2019). CDAG: A Serialized blockDAG for Permissioned Blockchain. <http://arxiv.org/abs/1910.08547> adresinden erişildi.
- He, J., Wang, G., Zhang, G. ve Zhang, J. (2021). Consensus mechanism design based on structured directed acyclic graphs. *Blockchain: Research and Applications*, 2(1), 100011. doi:10.1016/j.bcra.2021.10001
- Hellani, H., Sliman, L., Samhat, A. E., Samhat, A., Hellani, H., Sliman, L., ... Tangle, E. E. (2022). Tangle the Blockchain : Towards Connecting Blockchain and DAG To cite this version : HAL Id : hal-03659390 Tangle the Blockchain : Towards Connecting Blockchain and DAG.
- Hilary, G. ve Xiaolei Liu, L. (2021). Blockchain and Other Distributed Ledger Technologies in Finance. *The Palgrave Handbook of Technological Finance*, (2017), 243–268. doi:10.1007/978-3-030-65117-6\_10
- Hsu, C. H., Hung, S. C., Chen, H., Sun, F. K. ve Chang, Y. W. (2021). A DAG-Based Algorithm for Obstacle-Aware Topology-Matching On-Track Bus Routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 40(3), 533–546. doi:10.1109/TCAD.2020.3002546
- Hu, X., Wang, K., Wang, K., Hu, Y., Wang, J. J. ve Wang, S. (2018). Multi-constrained routing optimization algorithm based on DAG. *Proceedings: IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 1, 5906–5910. doi:10.1109/IECON.2018.8591086
- Hu, Y., Hu, X., Wang, J., Wang, K. ve Wang, K. (2018). Multi-constrained routing algorithm based on PPSO-FWA in DAG. *2018 IEEE 4th International Conference on Computer and Communications, ICC 2018*, 275–279. doi:10.1109/CompComm.2018.8780608
- Karakus, M. ve Guler, E. (2020). RoutingChain: A Proof-of-Concept Model for a Blockchain-Enabled QoS-Based Inter-AS Routing in SDN. *2020 IEEE International Black Sea Conference on Communications and Networking, BlackSeaCom 2020*. doi:10.1109/BlackSeaCom48709.2020.9235021
- Karakus, M., Guler, E. ve Uludag, S. (2021). QoSChain: Provisioning Inter-AS QoS in Software-Defined Networks with Blockchain. *IEEE Transactions on Network and Service Management*, 18(2), 1706–1717. doi:10.1109/TNSM.2021.3060476
- Kotilevets, I. D., Ivanova, I. A., Romanov, I. O., Magomedov, S. G., Nikonov, V. V. ve

- Pavelev, S. A. (2018). Implementation of directed acyclic graph in blockchain network to improve security and speed of transactions. *IFAC-PapersOnLine*, 51(30), 693–696. doi:10.1016/j.ifacol.2018.11.213
- Kreutz, D., Ramos, F. M. V., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S. ve Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1), 14–76. doi:10.1109/JPROC.2014.2371999
- Lewenberg, Y., Sompolinsky, Y. ve Zohar, A. (2015). Inclusive block chain protocols. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8975, 528–547. doi:10.1007/978-3-662-47854-7\_33
- Li, L., Huang, D. ve Zhang, C. (2023). An Efficient DAG Blockchain Architecture for IoT. *IEEE Internet of Things Journal*, 10(2), 1286–1296. doi:10.1109/JIOT.2022.3206337
- Nunes, B. A. A., Mendonca, M., Nguyen, X. N., Obraczka, K. ve Turetti, T. (2014). A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys and Tutorials*, 16(3), 1617–1634. doi:10.1109/SURV.2014.012214.00180
- Perazzo, P., Arena, A. ve Dini, G. (2020). An Analysis of Routing Attacks against IOTA Cryptocurrency. *Proceedings - 2020 IEEE International Conference on Blockchain, Blockchain 2020*, 517–524. doi:10.1109/Blockchain50366.2020.00075
- Podili, P., Cherupally, S., Srinivas, B. ve Kataoka, K. (2022). Inter-Domain Prefix and Route Validation Using Fast and Scalable DAG Based Distributed Ledger for Secure BGP Routing. *Journal of Network and Systems Management*, 30. doi:10.1007/s10922-022-09668-2
- Popov, S. (2018). IOTA Whitepaper v1.4.3. *New Yorker*, 81(8), 1–28. [https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1\\_4\\_3.pdf](https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf) adresinden erişildi.
- Rischke, J. ve Salah, H. (2020). *Software-defined networks. Computing in Communication Networks: From Theory to Practice*. Elsevier Inc. doi:10.1016/B978-0-12-820488-7.00018-9
- Sarfraz, U., Alam, M., Zeadally, S. ve Khan, A. (2019). Privacy aware IOTA ledger: Decentralized mixing and unlinkable IOTA transactions. *Computer Networks*, 148, 361–372. doi:10.1016/j.comnet.2018.11.019
- Schueffel, P. (2018). Alternative Distributed Ledger Technologies Blockchain vs. Tangle vs. Hashgraph - A High-Level Overview and Comparison -. *SSRN Electronic Journal*, 1–8. doi:10.2139/ssrn.3144241

- Suciu, G., Nadrag, C., Istrate, C., Vulpe, A., Ditu, M. C. ve Subea, O. (2018). Comparative Analysis of Distributed Ledger Technologies. *6th Global Wireless Summit, GWS 2018*, 370–373. doi:10.1109/GWS.2018.8686563
- Teo, S. G., Cao, J. ve Lee, V. C. S. (2020). DAG: A general model for privacy-preserving data mining: eded Abstract). *Proceedings - International Conference on Data Engineering içinde* (C. 2020-April, ss. 2018–2019). IEEE Computer Society. doi:10.1109/ICDE48307.2020.00228
- Varyani, N., Zhang, Z. L. ve Dai, D. (2020). QROUTE: An Efficient Quality of Service (QoS) Routing Scheme for Software-Defined Overlay Networks. *IEEE Access*, 8, 104109–104126. doi:10.1109/ACCESS.2020.2995558
- Wibowo, F. X. A., Gregory, M. A., Ahmed, K. ve Gomez, K. M. (2017). Multi-domain Software Defined Networking: Research status and challenges. *Journal of Network and Computer Applications*, 87, 32–45. doi:10.1016/j.jnca.2017.03.004
- Xia, W., Wen, Y., Foh, C. H., Niyato, D. ve Xie, H. (2015). A Survey on Software-Defined Networking. *IEEE Communications Surveys and Tutorials*, 17(1), 27–51. doi:10.1109/COMST.2014.2330903
- Xie, W., Goyal, M., Hosseini, H., Martocci, J., Bashir, Y., Baccelli, E. ve Duresi, A. (2010). A performance analysis of Point-to-Point routing along a directed acyclic graph in low power and lossy networks. *Proceedings - 13th International Conference on Network-Based Information Systems, NBiS 2010*, 111–116. doi:10.1109/NBiS.2010.65
- Yang, S., Chen, Z., Cui, L., Xu, M., Ming, Z. ve Xu, K. (2019). CoDAG: An efficient and compacted DAG-Based blockchain protocol. *Proceedings - 2019 2nd IEEE International Conference on Blockchain, Blockchain 2019*, (61772345), 314–318. doi:10.1109/Blockchain.2019.00049

