Mühendislik Mimarlık Fakültesi Dergisi
Journal of The Faculty of Engineering and Architecture of Gazi University

# Kabul Edilmiş Makale/Accepted Manuscript


**Başlık:** Şehir içi kavşak yönetim sistemleri için SDN temelli bir VANET mimari önerisi

**Title:** A Proposal SDN based VANET architecture for urban intersection management systems

**Yazarlar/Authors:** Musa Balta, İbrahim Özçelik

ID: 5000215483
DOI: https://doi.or./10.17341/gazimmfd.460544

**Dergi İsmi:** *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*
**Journal Name:** *Journal of the Faculty of Engineering and Architecture of Gazi University*

**Geliş Tarihi/Received Date:** 02.04.2018
**Kabul Tarihi/Accepted Date:** 08.08.2018

**Makale Atıf Formatı/Manuscript Citation Format:**
Musa Balta*, İbrahim Özçelik, A Proposal SDN based VANET architecture for urban intersection management systems, *Journal of the Faculty of Engineering and Architecture of Gazi University* (2018), https://doi.or./10.17341/gazimmfd.460544

# Şehir içi kavşak yönetim sistemleri için SDN temelli bir VANET mimari önerisi

Musa Balta*, İbrahim Özçelik

Sakarya Üniversitesi, Bilgisayar ve Bilişim Bilimleri Fakültesi, Bilgisayar Mühendisliği Bölümü, Esentepe Kampüsü, Serdivan, 54050, Sakarya

**Öne Çıkanlar**

- Kavşak yönetim sistemi için SDN temelli VANET mimari önerisi
- Kavşak yönetim sistemlerinde veri çevrimi
- Simülasyon ve performans uygulaması

**Özet**

Akıllı şehirlerin en büyük sorunu araç trafiği ve bunun iyi bir şekilde yönetilememesidir. Şehir içi kavşak yapılarının fiziksel özellikleri ve plansız yol kesişmelerinden dolayı oluşan trafik akımları, zaman/nakit kaybı, stres, daha fazla yakıt tüketimi gibi birçok olumsuz etkiye sebep olmaktadır. Bu nedenle hem akademik hem de ticari çevrelerde bir akıllı şehir uygulaması olan kavşak yönetim sistemleri üzerine birçok çalışmalar yapılmaktadır. Son yıllarda yapılan bu çalışmalarda araçların birbirleri arasında veya saha kenarındaki cihazlar ile haberleşmelerini kolayca sağlayarak ilgili trafik verilerinin merkeze taşınmasını sağlayan VANET (Araçsal Ağlar-Vehicular ad hoc networks) mimarisinin çok sık kullanıldığı görülmektedir. Diğer taraftan günümüz geleneksel ağlarında karşılaşılan performans, programlanabilirlik, ölçeklenebilirlik, güvenlik ve yönetim zorluğu gibi sorunlara çözüm getiren yeni bir ağ mimarisi olan SDN (Yazılım tanımlı ağlar-Software defined networks) üzerine de birçok çalışma yapılmaktadır. Tüm bunlara bağlı olarak bu çalışmada, hem günümüz kavşak yönetim sistem modüllerini daha adaptif bir şekilde kullanabilmek hem de gelecekte eklenecek trafik tabanlı hizmetleri iletişim altyapısı değiştirilmeden istenilen servis kalitesinde sunabilmek için SDN ve VANET ağ paradigmalarının kavşak yönetim sistemlerinde nasıl birlikte kullanılacağı ile alakalı bir mimari önerisi sunulmuştur. Ayrıca önerilen SDN temelli VANET mimarinin, sinyalizasyon uygulamalarındaki işlevselliğini göstermek için SDN kontrolörün modüler yapısıyla entegre çalışan bir karınca koloni algoritması geliştirilmiş ve sonrasında geleneksel sistemlerde kullanılan sabit zamanlı sistemler, Webster metodu ile zeki hesaplama tekniklerinden karınca koloni algoritması, parçacık sürü algoritması ve bir bulanık model ile karşılaştırılmıştır. Elde edilen sonuçlar neticesinde önerilen algoritmanın kavşak içi ortalama gecikme ve kuyruklanma gibi performans kriterlerinde geleneksel sistemlere göre %15-22 arasında, hesaplamalı tekniklere göre ise %7-12 arasında iyileştime gösterdiği gözlemlenmiştir. Ayrıca önerilen sistem işlem süresi ve uçtan-uca performans kriterlerine göre de kıyaslanmış sonuçlar paylaşılmıştır.

**Anahtar Kelimeler:** Kavşak yönetim sistemleri, araçsal ağlar, yazılım tanımlı ağlar, karınca koloni algoritması

**A Proposal SDN based VANET architecture for urban intersection management systems**

**Highlights**

- SDN based VANET proposal for intersection management systems
- Data cycle in intersection management systems
- Simulation and performance application

**Abstract**

One of the main challenges for developed cities is vehicle traffic and its management. Because of physical structures and cross roads of urban intersections, traffic flows may cause time delay, congestion, traffic accidents and more fuel/time consumption. There are many studies about these challenges of urban intersection managements both in literature and in practice. In recent years, VANET (Vehicular ad hoc networks) architecture has been preferred in these studies, which enables vehicles to easily

communicate with each other or with devices on the edge of the road and also to transfer the related traffic data to the center. On the other hand, there are many studies on a new network paradigm called SDN (software defined networks) that solves problems such as performance, programmability, scalability, security and management difficulties in today's traditional networks. Depending on all this, in this study, we will present an architectural proposal about how to use SDN and VANET network paradigms together in intersection management systems in order to use both existing intersection management system modules more adaptively and future traffic based services to be added in desired service quality without changing the communication infrastructure. In addition, to show the functionality of the proposed SDN based VANET architecture in signaling system, we propose an ant colony algorithm based SDN controller's modules. The results obtained from ant colony algorithm based SDN modules was compared with fixed-time signaling and Webster equation which are used in traditional systems and also computational techniques (particle swarm optimization and fuzzy model) on a isolated intersection. It shows that the proposed algorithm and architecture provide an improvement about % 15-22 according to existing systems and %7-12 according to computational techniques in average delay and queueing.

**Key Words:** Intersection management systems, vehicular ad-hoc networks, software defined networks, ant colony algorithm

## 1. GİRİŞ (INTRODUCTION)

Büyük şehirlerde yaşayan insan sayısının 2030'larda yaklaşık 5 milyara ulaşacağı tahmin edilmektedir [1]. Şehirlerdeki bu nüfus artışı beraberinde enerji, su, sağlık, ulaşım, iletişim, barınma ve güvenlik gibi temel yaşam gereksinimlerini olumsuz etkileyeceğinden dolayı bu konular üzerine kaynakların daha efektif kullanılması için ulusal ve yerel otoriteler tarafından hem akademik hem de ticari çalışmalar yapılmaktadır [2,3]. Yapılan bu çalışmalardan biri de akıllı ulaşım sistem uygulaması olan kavşak yönetim sistemleridir. Kavşak yönetim sistemleri, büyük şehirlerde artan trafik yoğunluğunu azaltmak, olası kazaları engellemek, zaman/yakıt tüketimini azaltmak vb. hedeflere ulaşmak için trafik yönetim sistemlerinin sinyalizasyon ve yolcu bilgilendirme işlevlerini sağlayan en önemli alt modulüdür.

Günümüzde mevcut kavşak yönetim sistemleri, birbirine bağlı birçok ağ cihazından ve farklı veri trafiklerinden oluşan karmaşık ağ yapıları haline geldikleri için, bu sistemlerin iletişim altyapılarının özellikle değişken araç trafiği senaryoları altında (trafik yoğunluğunun değişmesi, trafik kazaları, ambulans geçişi vs.) ağ ölçeklenebilirliği, paket akış kontrolü, efektif bant genişliği kullanımı, kolay cihaz yönetimi ve güvenlik gibi konularda yetersiz kaldığı görülmüştür [4,5]. Ayrıca ağdaki veri akışında oluşabilecek her hangi bir problem (bant genişliğindeki darboğaz, cihaz bozulması, bağlantı kopması veya siber saldırı vb.) ise daha fazla trafik sıkışıklığı, trafik kazası, yakıt tüketimi ve çevre kirliliğine neden olabilir [6]. Kavşak yönetim sistemlerinde karşılaşılan bir başka problem ise sahadan veri çekmek için kullanılan dedektör ve kamera sistemlerinin fiziksel şartlardan (rüzgâr, yağış, bozulma vb.) kolay etkilenerek saha ile merkez arasında efektif bir veri akışının sağlanamamasıdır [7,8]. Bu problemlerin üstesinden gelebilmek adına sahadan veri çekme işlemleri için akademik çalışmalarda son zamanlarda VANET mimarisi tercih edilmektedir [9,10].

Kavşak yönetim sistemleri için bu çalışmada önerilen yapı içerisinde sahadan veri çekmek için kullanılacak olan VANET, son zamanlarda hem ticari hem de akademik çevrelerde çok ilgi gören bir konudur. Araçların mobil düğüm olarak modellendiği bu ağ paradigmasında araçlar kendi aralarında veya yol kenarındaki ünitelerle iletişim kurup trafik verimliliği, trafik emniyeti ve bilgi eğlence alanında çalışmalar yapılmaktadır. Her ne kadar VANET

ağ mimarisi günümüzün popüler bir çalışma konusu ve teknoloji olmasına rağmen, özellikle hızlı ağ topoloji değişikliğindeki bağlantı kopmaları, çoklu atlamalardaki dengesiz ağ trafiği, yetersiz ağ optimizasyonu ve güvenlik gibi konularda yetersiz kalmaktadır [12]. VANET ağlarda karşılaşılan bu problemlere çözüm getirebilmek için geleneksel ağlardaki yapılardan farklı olarak kontrol ve veri düzlemini birbirinden ayıran, ağ cihazlarının sadece iletim için işlem yapmasını sağlayan ve yapılan çalışmanın da iletim ve uygulama altyapısını oluşturan SDN temelli VANET mimari çözüm önerileri literatürde yer almaya başlamıştır. 2014 yılında Ku ve arkadaşları, VANET sistemlerinde karşılaşılan hızlı topoloji değişikliği ve yetersiz bantgenişliği problemlerine çözüm getirmek için SDN tabanlı bir VANET sistemi önermişler ve bu önerinin içerisinde araçları mobil SDN, RSU'ları ise sabit SDN olarak modellemişlerdir. Önerilen bu model için 3 farklı iletişim modu (merkezi, dağıtık ve hibrid) tanımlanmasına rağmen, bu modlarının akıllı ulaşım sistemleri alanında nasıl kullanılacağı ile alakalı bir bilgi verilmemiştir. Bir başka çalışmada ise Troung ve arkadaşları 2015 yılında diğer çalışmadan biraz daha farklı olarak SDN temelli bir VANET sistemini, merkezdeki işlem yükü fazlalığını azaltmak için yerel olarak hesaplanması mantığına dayalı fog computing (sis hesaplaması) ile önermişlerdir [13]. Ancak RSU'lar üzerinde çalışacak sis hesaplamasında uçtan-uca iletim mimarisi veya nasıl bir paket yapısının ele alındığından bahsedilmemiştir. Chun Liu ve arkadaşları ise 2015 yılında SDN temelli bir VANET sisteminde daha kolay yönetim sağlamak amacıyla konum-temelli yayın yapan bir yapı önermişlerdir [14]. Çalışmada, openflow tabanlı konum-temelli yayın paketleri (Packet_IN) sadece SDN anahtar cihazlarında oluşturulmakta, araçlar ise mobil bir SDN düğüm olarak tanımlanmamıştır.

Yapılan araştırmalar neticesinde, araçların ve RSU'ların (yol kenarı üniteleri-road side unit) birer SDN etmen düğüm olarak modellendiği ve karar verici bir merkezin olduğu SDN temelli VANET sistemlerinin, geleneksel VANET sistemlerine göre farklı trafik senaryoları ve farklı ağ topolojileri altında, araçların yeniden bağlantı kurulumu, yol hesaplaması veya bir yapılandırma yapmasına gerek kalmadan merkezi SDN kontrolörün göndereceği akış bilgileri sayesinde dinamik yol seçiminde, frekans/kanal seçiminde ve de kullanılan kablosuz arayüzün sinyal gücünün ayarlanmasında paket teslimi oranı, gecikme gibi konularda daha iyi sonuç verdiği görülmüştür [12-14]. Ayrıca SDN temelli bir VANET sisteminde geliştirilecek olan bir trafik güvenlik veya verimlilik uygulaması (öncelikli araç geçişi, iletişim aralığının belirlenmesi, rota hesabı, güvenlik vb.) neticesinde elde edilen sonuçlar araçlara kontrolör tarafından bir akış girdisi ile gönderilerek, araçların bu duruma kolayca reaksiyon almaları sağlanabilir.

Bu çalışmada, hem yukarıda VANET mimarileri için verilen yetersiz ağ optimizasyonu, bağlantı kopmaları ve dengesiz ağ trafiği gibi sınırlamalara bağlı olarak hem de kavşak yönetim sistemlerinin her bir veri çevrimi modülü içerisinde karşılaşılan iletim ve uygulama altyapılarındaki ölçeklenebilirlik, paket akış kontrolü ve dinamik programlamabilirlik gibi sınırlamalara çözüm getirmek adına uçtan uca her iki yönde (hem araçtan merkez kontrolöre hem de kontrolörden araca) çalışan SDN temelli VANET mimari önerisi ve sinyalizasyon hesabı için de SDN modüllerine dayalı karınca koloni algoritma (SDN temelli KKA) önerisi sunulmaktadır. Önerilen mimaride araçların trafik senaryolarındaki hareketleri ve VANET mimarisine göre birbirleri ile olan iletişimleri SUMO ve NS2 benzetim platformlarında gerçeklenmiş olup, araçlardan elde edilen trafik verilerinin SDN etmen RSU'dan, merkezdeki SDN kontrolöre kadar olan iletimi ise MiniNet sanallaştırma platformunda gerçeklenmiştir. SDN temelli KKA'da sahadan gelen trafik verilerini işlemek için SDN kontrolör üzerinde, "Hat Keşfi (Link

discovery)" ve "Topoloji Yöneticisi (Topology Manager)" modülleri kullanılmıştır. Bu modüller sayesinde kavşak içlerindeki SDN etmen araçların, kontrolör üzerinde net konum ve komşuluk bilgileri oluşturulmuş, her yöndeki kuyruk uzunlukları ve ortalama gecikme ağ topolojisine (kavşağın fiziksel yapısı) göre dinamik olarak hesaplanmıştır. Önerilen mimarinin kavşak yönetim sistemlerine uygunluğunu ölçmek için ise literatürdeki en önemli problem olan trafik sinyalizasyonu üzerine performans testleri yapılmıştır. Çalışmanın bilimsel katkıları aşağıda maddeler halinde belirtilmiştir:

- İncelenen akademik çalışmalar neticesinde, SDN ve VANET mimarilerinin birlikte ele alındığı çalışmaların olduğu görülmüş, fakat bu çalışmaların kavşak yönetim sistemi gibi bir akıllı ulaşım sistemi içerisinde kullanılmadığı ve her iki yönde uçtan-uca bir mimari önerisinin sunulmadığı belirlenmiştir. Bu sebeplerden ötürü, bu çalışmada özellikle değişken trafik şartları için geleneksel kavşak/trafik yönetim sistemlerinin her bir modülünde (veri toplama, veri ayrıştırma, veri işleme ve servis dağıtımı) karşılaşılan problemlere çözüm getirmesi için daha esnek, efektif, dinamik ve kolay programlanabilirlik sağlayan ve de uçtan-uca iletimi sağlayan SDN temelli bir VANET mimari önerisi sunulmuştur.
- Kavşak yönetim yazılımının istenilen fonksiyonlarını gerçekleştirmesi için hem araçtan hem de kavşaktan verilerin merkeze iletilmesi gerekmektedir. Araçlar mobil SDN düğüm, RSU'lar da sabit SDN düğüm olarak tanımlanacağı için uçtan uca SDN iletişimini sağlamak amacıyla hem araçtan hem de RSU'dan alınan verilerin SDN openflow protokol yapısı içerisinde taşınması gerekmektedir. Bu kapsamda, trafik yoğunluğu bilgilerini hesaplamak için sahadan merkeze iletim yönünde ve merkezdeki hesaplamalar sonucunda ise merkezden sahaya sahadaki trafik lambalarının nasıl bir reaksiyon alması ile ilgili aksiyon bilgilerini içeren paket yapıları tasarlanmıştır.
- SDN kontrolörün modüler yapısıyla bütünleşmiş çalışan bir karınca koloni algoritması (SDN temelli KKA) geliştirilmiş, geliştirilen bu optimizasyon yaklaşımı farklı trafik senaryoları altında mevcut geleneksel sinyalizasyon tekniklerinden sabit zamanlı sistemler, Webster eşitliği ve hesaplamalı tekniklerden parçacık sürü optimizasyonu ve bulanık mantık temelli yaklaşımlarla karşılaştırılmıştır. Elde edilen sonuçlara göre ortalama gecikme ve kuyruk uzunluğu gibi trafik performans karşılaştırmalarında önerilen SDN temelli KKA diğer optimizasyon tekniklerine göre %7-12 arasında, Webster ve sabit zamanlı sistemlere göre ise %16-20 arasında iyileşme sağladığı gözlemlenmiştir. Sunucu performansı ve uçtan-uca gecikme gibi ağ performans karşılaştırmalarında ise SDN temelli KKA'nın diğer optimizasyon tekniklerine göre %15-22 arasında iyileşme gösterdiği gözlemlenmiştir.

Çalışmanın bundan sonraki kısımlarında, önce Bölüm 2'de kavşak yönetim sistemi ve bileşenleri hakkında temel bilgiler verilerek, bu bileşenlerin günümüz trafik şartlarında karşılaştıkları problemler ele alınacaktır. Daha sonra Bölüm 3'de bu problemlerin çözümü için önerilen SDN temelli VANET mimarisinin genel yapısı ve çalışma şekli hakkında bilgiler verilecek, sonra çalışmada kullanılan sinyalizasyon tekniklerinden bahsedilecektir. Son olarak ise Bölüm 4'de, sistemin SUMO (Simulation Urban MObility) ve Mininet sanallaştırma platformları üzerinde nasıl gerçekleştirildiği ve diğer metotlarla yapılan performans karşılaştırılmaları analiz edilecektir.

## 2. KAVŞAK YÖNETİM SİSTEMLERİ VE BİLEŞENLERİ (INTERSECTION MANAGEMENT SYSTEMS AND ITS COMPONENTS)

Kavşak yönetim sistemleri, şehir içi kavşak içerisindeki ortalama araç gecikme sürelerini, ortalama dur/kalk

sayısını ve kuyruklanmayı azaltmak için, sinyalize kavşakların plan sürelerinin; oluşan trafik hacmi, kuyruklanma gibi parametrelere göre optimize edilerek yeni sürelerin gerçek zamanlı olarak uygulandığı çalışma sistemleridir [3]. Bu sistemler, saha ve merkez yapısı olmak üzere iki kısımdan oluşmaktadır. Saha modeli, trafik sinyalizasyon işlemleri için trafik lambalarından, yolcu bilgilendirme/yönlendirme için değişken mesaj sistemlerinden ve de trafik yoğunluk bilgilerini elde etmek için yol kenarına döşenmiş manyetik/loop dedektörler ile kamera cihazlarından oluşmaktadırlar. Merkez ile saha ekipmanları arasındaki verilerin iletimi ise, sahada kullanılan cihazların bir anahtar cihazına, anahtarın da bir geniş alan ağına bağlanmasıyla gerçekleşir. Kavşak yönetim sistemlerinin merkez yapıları ise atanmış (predefined) sinyal kontrol mekanizması, webster eşitliği veya bir hesaplamalı teknik (yapay zekâ, yapay sinir ağı vb) kullanılarak sahadan gelen verilere göre işlevsellik gösteren klasik ağ iletişim ve programlama yapısı mantığına dayalı uygulamalardan oluşmaktadırlar [4]. Çalışma yapısı ve veri çevrimi açısından ise kavşak yönetim sistemleri sahadan veri toplama, veri ayrıştırma, veri işleme ve hizmet dağıtımı olmak üzere 4 ana bölümden oluşmaktadır. Bu bölümde kavşak yönetim sistemlerinin, bu bileşenlerinin neler olduğu ve mevcut sistemlerde karşılaşılan temel problemlerden bahsedilecektir.

### *2.1. Veri toplama (Data gathering)*

Kavşak yönetim sistemlerini oluşturan ana bileşenlerden ilki sahada bulunan ekipmanlardan merkeze trafik verisi çekme işlemidir. Mevcut sistemlerde ve akademik çalışmalarda kullanılan değişik şekillerde sahadan veri çekme işlemleri vardır. İlk olarak yol kenarına yerleştirilen manyetik dedektörler ile başlayan veri çekme teknikleri daha sonraları loop dedektörler, kamera kullanımları ve de web uygulamaları (Yandex, Google vb.) ile gelişme göstermiştir.

Manyetik dedektörlerin düşük maliyetli ve kolay kurulum özelliklerine sahip olmasına rağmen yol kenarlarında açıkta bulunmalarından dolayı çevresel etkenlerden (rüzgâr, yön değiştirme vb.) çabuk etkilenmektedirler [1]. Manyetik dedektörlerin bu zaafiyetlerini ortadan kaldırmak amacıyla loop dedektörler geliştirilmiştir. Asfalt altına döşenen bakır kablolar üzerinden araç geçtikçe tetiklenen sistemlerden oluşan loop dedektörlerin ağır tonajlı araç geçişlerinde bozulmaları, onarılmaları için asfaltın kazılması gibi nedenlerden dolayı kullanımı zor olan bir sahadan veri çekme tekniğidir. Ayrıca loop dedektörler için döngü süreleri söz konusudur (min 6sn), fakat bu süre içerisinde bir kaç araç aynı anda geçerse sistem doğru bilgi üretememektedir. Diğer taraftan dedektörler sadece sezme işlemi yapmakta, trafik yoğunluğu ile alakalı herhangi bir bilgi sunamamaktadırlar [6].

Günümüz trafik/kavşak yönetim sistemlerinde sahadan trafik verisini çekmede daha fazla kamera sistemleri kullanılmaktadır. Anayollar, kavşaklar, köprüler gibi trafiğin yoğun olduğu alanlara yerleştirilen kameralar kendi görüş açılarında sanal noktalar oluşturup, teknik özelliklerine göre ya kendi üzerlerinde görüntü işleme yapıp üretilen trafik bilgisini merkeze gönderirler, ya da hiçbir görüntü işleme yapmadan görüntüyü direk olarak merkeze aktarırlar. Ağır hava şartları (yağmur, kar yağışı) sebebiyle görüntü açısının netliğinin kaybolması ve kameranın rüzgâr veya çevresel etkenler sebebiyle yön değiştirmesi gibi nedenlerden dolayı da kameralar günümüz trafik/kavşak yönetim sistemi uygulamalarını gerçeklemede tek başına yeterli olamamaktadırlar [3]. Son olarak da kullanıcıların trafik verisi girdiği (yandex gibi) web uygulamaları da internetin çekmediği veya kullanıcılar tarafından güvenilir veri girişi olmadığı durumlarda trafik/kavşak yönetim sistemlerini olumsuz etkilemektedir.

Literatürde, kavşak yönetim sistemlerinin karşılaşmış oldukları veri toplama problemlerine (cihaz bozulması, güvenilir olmayan veri ve çevresel etmenler vs.) çözüm bulmak adına araçsal ağlar kullanılarak yapılan çalışmalar da mevcuttur. Örneğin, Bai ve arkadaşları araçların hız, konum ve id bilgilerini alıp RSU'lar üzerinde tabloya kaydetme işlemi yapmış ve yoğunluğu hesaplamışlardır [10]. Bir başka çalışmada ise Barrachina ve arkadaşları araçların RSU'lara gönderdikleri beacon sinyali sayıları için eşik değeri belirleyerek trafik yoğunluğunu hesaplamışlardır [11]. Daha sonra bu sayı en yakın RSU'ya iletilmektedir. Araçsal ağlar kullanılarak yapılan bu çalışmalarda genel olarak periyodik olarak beacon sinyalleri temel alınmış ve hesaplamalar ona göre yapılmıştır. Yapılan çalışmada, veri toplama modülü için mevcut tekniklerin aksine fiziksel şartlardan (dedektörlerin bozulması, kameranın yönünün değişmesi, yoğun hava şartları, yanlış kullanıcı verisi gibi) iletim altyapısı anlamında etkilenmeyen VANET mimarisi tercih edilmiştir.

### *2.2 Veri ayrıştırma (Data fusion)*

Farklı kaynak ve metotlar ile toplanan trafik verileri birbirlerinden farklı paket yapıları ve içeriklere sahiptirler. Bu trafik verilerinin kavşak yönetim sistemleri içerisindeki veri işleme modüllerinde kullanılabilmeleri için filtreleme, anlamlandırma, analiz etme gibi standardize işlemlerden geçirilmesi gerekmektedir [3]. Yapılan çalışmada, veri ayrıştırma modülü için paketlerin her birinin formatlarının ayarlanması, filtrelenmesi ve standardize edilme işlem yüklerinin önüne geçmek amacıyla her bir araç için TCP tabanlı trafik paket yapısı oluşturulmuştur. Bu paket yapıları içerisinde aracın kimlik bilgileri (id), hız, konum, gecikme vb. bilgiler yer almaktadır. Aynı format ve boyutta oluşturulan bu TCP paketleri daha sonra SDN anahtar cihazı üzerinde akış tabloları ile kontrol edilerek Openflow mesaj tipleri içerisinde merkeze taşınmıştır.

### *2.3. Veri işleme (Data processing)*

Anlamlandırılan bu veriler, yeşil ışık süre optimizasyonu, trafik yoğunluğu hesaplama, rota ve ulaşım süresi hesaplaması gibi alt modüller ve belli algoritmalardan oluşan merkezi bir yazılım içerisinde işlenir ve yorumlanırlar. Önceleri sadece trafik sinyalizasyonunun sabit zamanlı olarak ayarlanması şeklinde tasarlanan bu sistem yazılımları, günümüz içerisinde sahadan gelen farklı trafik bilgilerine (anlık, periyodik, tanımlı, değişken vs.) göre dinamik ve adaptif olarak hizmet verebilen komplex yapılar haline gelmiştirler. Özellikle sinyalizasyon ve rota hesaplama gibi modüllerin sadece bir matematiksel modele oturtulamayacak kadar karmaşık ve doğrusal olmayan stokastik sistemler haline gelmeleri sebebiyle artık günümüz kavşak/trafik yönetim sistem yazılımlarının hesaplamalı teknik kullanımına açık olmaları gerekmektedir [4].

Literatürde, hesaplamalı teknik kullanılarak bu konular üzerinde yapılan birçok çalışma mevcuttur. Örneğin 2012 yılında Nietoa ve arkadaşları, trafik ışıklarının sinyalizasyon süreleri için efektif bir döngü kurmak için bir yapay zeka tekniği olan "sürü optimizasyonu" kullanmışlardır [15]. Kullandıkları bu tekniği Malaga ve Sevilia şehirlerini model alarak oluşturmuşlardır. Hea ve arkadaşları ise 2012 yılında, trafik sinyallerinin zamanlamasını ayarlamak için yapay zekada optimizasyon algoritması olarak kullanılan "karınca kolonisi" algoritmasını kullanmışlardır [16]. Trafik akışını etkin bir şekilde kullanmak için zaman gecikmesi, dur-kalk sayısı, trafik kapasitesi gibi performans parametreleri seçip, sistemi Webster eşitliği ve genetik algoritma ile karşılaştırmışlardır. Yapılan çalışmada, veri işleme modülünde kavşak yönetim sistemlerinin en önemli sorunu olan trafik

sinyalizasyonu ele alınarak bir sürü zekası tekniği olan karınca koloni algoritması SDN'nin modülleri ve dinamik programlaması yardımıyla gerçeklenmiş ve diğer sinyalizasyon hesaplama yöntemleriyle karşılaştırılmıştır.

### *2.4. Servis dağıtımı (Service Delivery)*

Kavşak yönetim sistem yazılımı tarafından işlenen veriler daha sonra trafik sinyalizasyon cihazları, elektronik bilgilendirme levhaları, mobil uygulama vs ile son kullanıcıların (sürücülere, yayalara, yetkili kurumlara, özel kuruluşlara) hizmetine sunulmaktadır [4,5]. Örneğin Kammouna ve arkadaşları 2014 yılında yaptıkları hiyerarşik karınca-bulanık model neticesinde elde ettikleri sonuçları kavşak içlerindeki ajan yazılımlarına iletmektedirler [17]. Bir başka çalışmada ise Mariagrazia ve arkadaşları, şehir-içi trafik kontrolü için dinamik bir sinyal planı oluşturup bu bilgiyi sahadaki bilgilendirme levhalarına da iletmişlerdir [18]. Çalışmanın servis dağıtım modülünde ise hesaplanan yeşil ışık süreleri sahadaki SDN destekli araçlara yine SDN iletim mimarisi üzerinden aktarılmaktadır.

## 3. ÖNERİLEN SİSTEMİN MİMARİSİ (THE ARCHİTECTURE OF PROPOSED SYSTEM)

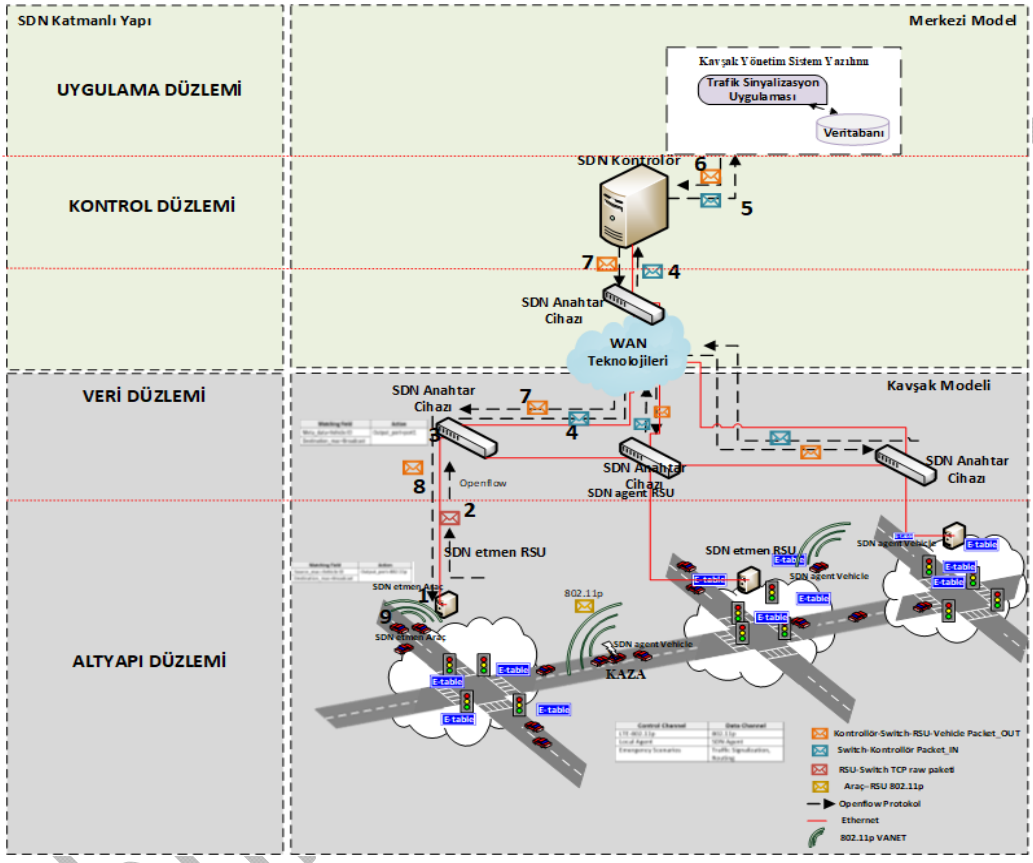
Çalışmanın bu bölümünde önce kavşak yönetim sistemleri için önerilen SDN temelli VANET mimarisi ile ilgili genel bilgiler verilecek, daha sonra ise bir üst bölümde anlatılan kavşak yönetim sistemi bileşenleriyle ilişkisi kurulacaktır.

### *3.1 Genel Yapı (Main Structure)*

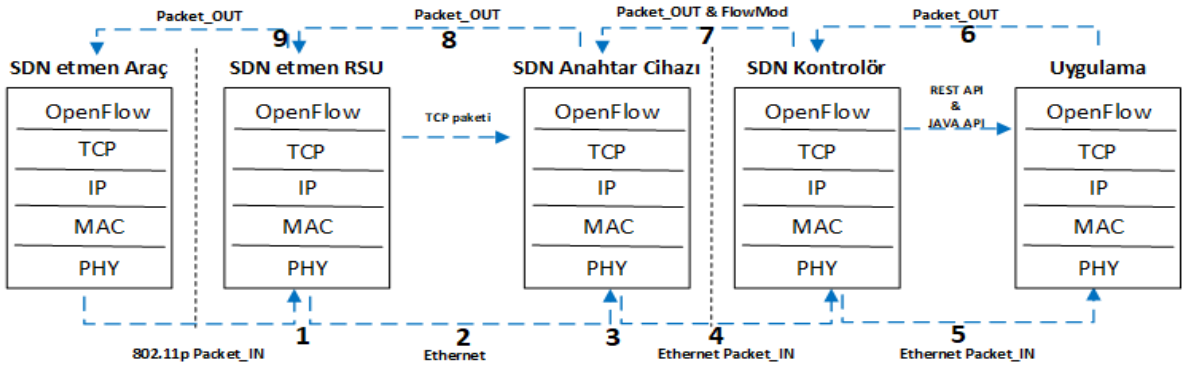
Yapılan çalışmada, önerilen SDN temelli VANET sistem mimarisi, mevcut sistemlerde olduğu gibi kavşak ve merkez modeli olmak üzere iki yapıdan oluşmaktadır. Şekil 1'de çalışma yapısı verilen mimari önerisine göre, kavşak modelinde SDN katmanlı mimarisinin altyapı ve veri düzlemleri bulunmakta iken, merkez modelinde ise kontrol ve uygulama düzlemleri bulunmaktadır. Önerilen bu yapı, ONF (Open Networking Foundation) tarafından önerilen SDN mimari hiyerarşisi dikkate alınarak tasarlanmıştır [19]. Altyapı katmanında SDN etmene sahip araçlar, SDN etmenli RSU cihazı, trafik lambaları ve yolcu yönlendirme/bilgilendirme de kullanılacak olan elektronik levhalar bulunurken, veri düzleminde ise bir adet SDN anahtar cihazı bulunmaktadır. SDN anahtar cihazlar birbirlerine karmaşık bir yapıda bağlanmıştır, kendi aralarındaki haberleşme ise bir WAN ağı (DSL, 3G, LTE, metroethernet vs.) üzerinden SDN kontrolör vasıtasıyla gerçekleştirilmektedir. Sistemin merkez modelinde ise SDN mimarisinin kontrol ve uygulama düzlemleri yer almaktadır. Kontrol düzleminde Floodlight kontrolörü bulunan sistemin uygulama düzleminde ise yapay zekaya dayalı bir kavşak yönetim yazılımı yer almaktadır.

Literatürdeki veri toplama yöntemlerine göre yapılan çalışmada, kavşak içindeki araçlar ile kavşak kenarındaki SDN etmenli RSU arasındaki iletişimde IEEE 802.11p VANET modülü tercih edilmiştir. Şehir-içi kavşak yönetim sistemleri için önerilen mimarinin Şekil 1'deki çalışma yapısına göre, önce SDN etmenli RSU sahadaki araçlardan gelen yayın paketlerini kendi üzerindeki akış tablosu ile karşılaştırır (1 numaralı işlem). Akış tablosundaki kontrol işlemlerinden sonra RSU, hem saha hem kavşak trafik bilgilerini içeren bir raw paket oluşturarak bağlı bulunduğu SDN anahtar cihazına gönderir (2 numaralı işlem). Anahtar cihaz da kendi akış tablosunda kontrol işlemi yaptıktan (3 numaralı işlem) sonra Openflow protokolü ile paketi merkezdeki SDN kontrolör yazılımı olan Floodlight'a gönderir (4 numaralı işlem). Floodlight kontrolörü kendisine gelen paketi ayrıştırarak ilgili trafik verisini trafik sinyalizasyonu, rota hesaplama gibi kavşak yönetim sisteminin modüllerine gönderir (5 numaralı işlem). Burada

işlenen veri neticesinde elde edilen sonuçlar önce kontrolör (6 numaralı işlem) ile oradan yine OpenFlow protokolü ile ağdaki SDN anahtar cihaza (7 numaralı işlem), SDN etmen RSU'ya (8 numaralı işlem) ve gezgin SDN düğümlere (araçlara-9 numaralı işlem) gönderilir. Önerilen sistemin uçtan-uca paket iletimi Şekil 2'de gösterilmiştir. Şehir-içi kavşak yönetim sistemlerinde SDN mimarisinin uygulanabilirliğini gösterebilmek adına bu çalışmada izole bir kavşak yapısı seçilmiş olup, diğer bölümde sistemin veri çevrimi tanımlanmıştır. SDN etmen özelliği, ağ yönetimi ve iletişimini daha kolay hale getirebilmek ve veri çevirminin son aşaması olan servis dağıtımı daha efektif olabilmesi için sahadaki tüm araç ve RSU'lara belli konfigürasyonlar neticesinde eklenmiştir.




Şekil 1. Önerilen SDN temelli VANET mimarisinin çalışma yapısı (General structure of proposed SDN based VANET architecture)




Şekil 2. Önerilen sistemin uçtan-uca iletim şeması (The end-to-end packet delivery scheme of the proposed system)

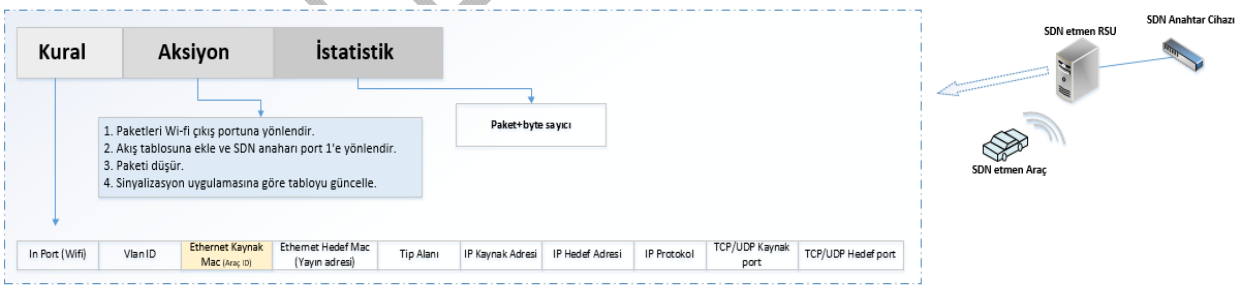
### 3.2 Önerilen Sistemin Veri Çevrimi (Data Cycle of the Proposed System)

Bu bölümde, önerilen SDN temelli VANET mimari yapısının, bir önceki bölümde anlatılan kavşak yönetim sistemlerinin her bir veri çevrim aşamasını gerçekleyen adımları anlatılacaktır. Mimarimiz içerisinde sahadaki araçlar ve yol kenarındaki RSU cihazlar, merkezdeki kontrolör ve anahtar cihazlarının üzerindeki yükü azaltmak ve SDN ağ paradigmasının avantajlarından yararlanabilmek için SDN etmen cihazlar olarak yapılandırılmıştır.

#### 3.2.1 Veri toplama ve ayrıştırma (Data gathering and fusion)

Sahadaki araçlardan kavşak içindeki RSU'ya kablosuz olarak gelen 802.11p paket ve veri tipleri merkeze iletilmeden önce üzerlerinde, anlamlandırma ve filtrelemeye gerek kalmadan yorumlanabilen, yönetimi kolay formatlara dönüştürmek ve de kablolu iletim ortamlarında kolayca aktarılabilmelerini sağlayacak işlemlerin yapılması gerekmektedir. Bu sebeple çalışmanın bu kısmında kavşak yönetim sistemlerinin veri toplama ve ayrıştırma bölümlerinin kendi mimarimiz içerisinde nasıl işlendiğiyle ilgili bilgiler verilecektir.

Mimarimizin veri toplama kısmında VANET ağ mimarisi tercih edilmiştir. Buna göre kavşak içlerindeki *araçlar id, konum, hız, kavşağa giriş ve çıkış zamanı, gecikme* gibi trafik bilgilerini IEEE 802.11p protokokolü ile yayınsal olarak SDN etmenli RSU cihazına aktarırlar. Sahadan gelen verilerin, RSU'lardan SDN anahtar cihazına, SDN anahtar cihazından da kontrolöre aktarılması işlemleri için ise 2 ayrı veri ayrıştırma modülü kullanılmıştır. İlk ayrıştırma modülünde, RSU cihazı sahadan gelen yayın trafik paketlerini kendi üzerindeki Şekil 3'de verilen akış tablosundaki Ethernet source mac (araç id) alanı ile karşılaştırır. Eğer daha önceden o aracın kaydı var ise, RSU gelen paketi 802.11p modülünden kavşak içerisindeki araçlara VANET yapısına göre geri gönderir. Eğer gelen paketin daha önceden kaydı yok ise (yani kavşağa yeni giren araçlar) RSU, kendi akış tablosuna aracın bilgilerini ekleyerek, kendi üzerlerindeki bilgileri de (*kavşak id, ışık durumu, zaman*) Şekil 4'deki TCP raw paketi içerisine ekler ve ikinci ayrıştırma modül işlemlerinin gerçekleneceği SDN anahtar cihazına gönderir.



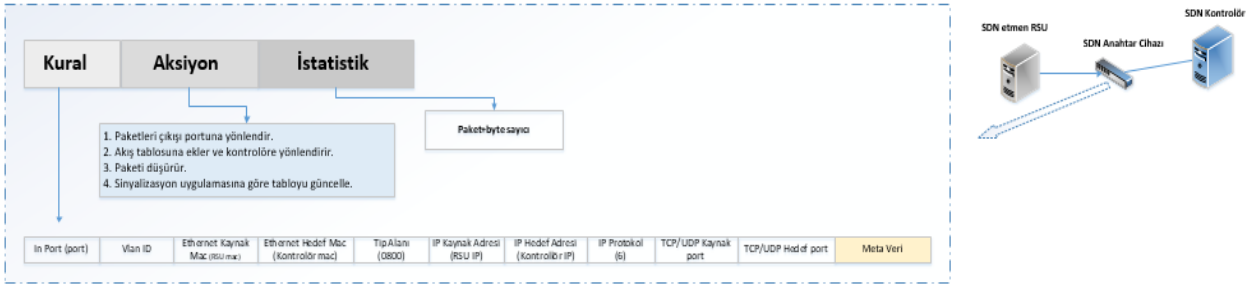

Şekil 3. RSU akış tablosu (RSU flow table)

| Ethernet Başlığı | IP Başlığı | TCP Başlığı | Araç ID | Kavşak ID | Ayrılma Konum | Varış Konum | Hız | Gecikme | Süre | Faz Durumu |
|---|---|---|---|---|---|---|---|---|---|---|

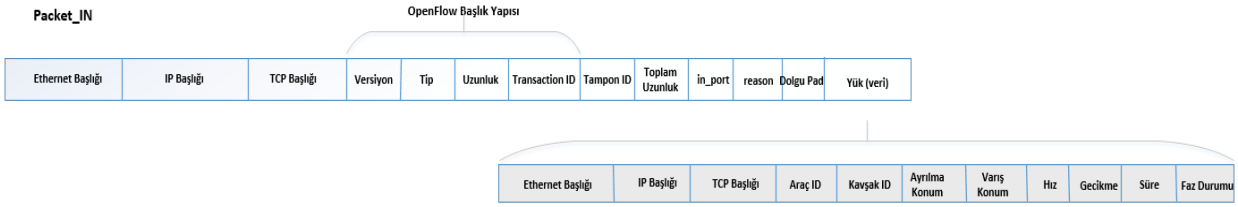
Şekil 4. RSU tarafından üretilen TCP trafik paketi (TCP raw packet generated by RSU)

2. veri ayrıştırma modülünde ise SDN anahtar cihazı kendisine bağlı kavşak içindeki RSU'dan gelen paketleri kendi akış tablosundaki Şekil 6'da gösterilen metaveriye göre değerlendirir. Metaveri, Şekil 4'deki RSU tarafından üretilen TCP paketinin Openflow packet_in mesajı içerinde kapsüllenerek tutulduğu kısımdır. SDN anahtar cihazı bu metaveri içerisindeki her aracın tanıtıcı numarasına (id) göre karar verir (Şekil 5). Eğer başka bir RSU'dan ilgili

araç ile ilgili bir girdi var ise, anahtar cihaz eşleşme olduğunu anlayarak gelen paketi elimine eder. Eşleşme olmaması durumunda ise daha önceden oluşturulan TCP paketi Şekil 6'daki Openflow protokolünün "packet_in" mesaj tipi içerisinde kapsüllenerek merkezdeki Floodlight kontrolör yazılımına iletilir. Böylelikle sahadan merkeze veri toplama ve ayrıştırma işlemleri tamamlanmış olur.




Şekil 5. SDN anahtar cihazı akış tablosu (SDN switch flow table)




Şekil 6. TCP trafik paketi ve packet_in mesajı (TCP raw traffic packet and Packet-in message)

### 3.2.2 *Veri işleme (Data processing)*

SDN kontrolör kendisine gelen packet_in mesajlarını pars ettikten sonra "tcp.payload" metodu ile ilgili verileri elde eder ve bir dahili (interior) uygulama olarak geliştirilen karınca algoritmasına aktarır. Bu sebepten dolayı bu bölümde önce sinyalizasyon ile ilgili genel bilgiler verilecek, daha sonra ise literatürde kullanılan hesaplama teknikleri ve karınca koloni algoritması anlatılacaktır. Sinyalizasyon sistemlerinin optimizasyon işlemlerinin yapılabilmesi için bazı trafik terimlerinin bilinmesi gerekmektedir. Bunlardan literatürdeki en önemlileri şu şekildedir [15,16];

- ***Trafik yoğunluğu:*** Bir yolda aynı yöne gitmekte olan taşıtların birim zaman içerisindeki sayısını tanımlar.
- ***Gecikme:*** Bir aracın, bir kavşak veya kontrol edilen bir noktada, diğer taşıtlar veya fiziksel etmenler (kavşak yapısı ve sinyalizasyon) nedeniyle kaybettiği süredir.
- ***Kapasite:*** Herhangi bir yol veya kavşaktan birim zaman içerisinde geçebilecek maksimum araç sayısıdır.
- ***Kuyruk uzunluğu:*** Yeşil fazda bir yol veya şeritten geçen araçların sayısıdır. Literatürde kuyruk uzunluğunun hesaplanması için kullanılan formül, sırasıyla Eş.1, Eş.2 ve Eş.3 de verilmiştir [16]:

$$q_{out}^{i}(t_1, t_2) = \min\left[q^{i}(t_1), 1 + Int\left(\frac{t_2 - t_1}{hw}\right)\right] \quad (1)$$

$$P(n) = \frac{(\lambda\Delta t)^n e^{-\lambda\Delta t}}{n!} \quad (2)$$

$$q(t) = [q^1(t), q^2(t), q^3(t), q^4(t)] \quad (3)$$

"$t_1$ ve $t_2$" zamanları arasında yeşil ışık süresince kavşaktan geçen araçların sayısının hesaplanmasında kullanılan Eş.1'deki formülde her yol için bulunan değerler bir diziye atanır. "q" kuyruktaki araç sayısını

ifade ederken, "hw" ise araçlar arasındaki boşluğu ifade eder. "Int" girdinin integer ifadesini tanımlarken, "$\lambda$" bir saatlik ortalama araç gelişini "$\Delta t$" ise süre değişimini ifade eder.

- ***Kavşak tipi:*** Kavşaktaki trafiğin sadece kendisini etkilemesi (izole) veya diğer kavşakları etkilemesi (koordineli) terimini tanımlar.

Literatürdeki trafik sinyalizasyonu konularında en çok bilinen ve bu çalışmada da performans karşılaştırması için tercih edilen geleneksel metotlar aşağıda verilmiştir [20,21]:

- ***Sabit zamanlı sinyalizasyon*** sistemlerinde trafik lambalarının yeşil ışık süre ve durumları daha önceden yönetici tarafından günün trafik bilgilerine (iş saatleri, haftasonları, özel günler vs.) ve kavşak yapılarına göre önceden ayarlanır. Günümüzde hala işlevselliğini koruyan bu sinyalizasyon sistemi özellikle orta ve yoğun trafik akışlarında çok fazla gecikmeye sebep olmaktadır.
- ***İngiliz (Webster) yöntemi*** ise ışıklı bir kavşağın kapasitesi sinyal kontrollü bir kavşaktan geçebilecek taşıt miktarı; "Bu trafiğin aldığı yeşil süreye ve yeşil süre boyunca dur çizgisini geçen maksimum taşıt sayısına bağlıdır." kavramını baz alır. Yeşil süre başladığı zaman taşıtların normal akış hızına ulaşana kadar bir miktar zamana ihtiyacı vardır. Bu birkaç saniyelik süreden sonra kuyruklanma sabit bir oranda boşalmaktadır ve bu sabit orana doygun akım denir. Literatürde özellikle doygun olmayan trafik şartlarında aşağıdaki Webster eşitliği (Eş.4) araçların kavşaklardaki gecikme sürelerinin hesaplanmasında kullanılmaktadır [22].

$$D_i = \frac{c(1-g_i)^2}{2(1-y_i)} + \frac{y^2}{2q_i g_i (g_i - y_i)} \tag{4}$$

"c" kavşaktaki yeşil ışıkların toplam süresi olan çevrim süresini ifade ederken, "q" birim zamanda geçen araç sayısı olan trafik akışını, "g" yeşil süresini, "y" ise sadece bir yeşil faz esnasındaki araç sayısını tanımlar.

Sabit zamanlı sinyalizasyon ve Webster eşitliğine dayalı sistemlerin özellikle doygun ve değişken trafik şartlarında yeterli performansı sağlayamamasından dolayı artık günümüzde hesaplamalı teknikler kullanılmaya başlanmıştır. Literatürde sinyalizasyon hesaplamalarında bulanık mantık, yapay sinir ağı, karınca koloni algoritması, reinforcement ve parçacık sürü optimizasyonu gibi bir çok yapay zeka tekniği kullanılarak yapılan çalışmalar vardır. Önerilen SDN temelli mimarinin işlevselliğini ölçmek için çalışmada SDN temelli KKA geliştirilmiştir. Performans karşılaştırmaları için ayrıca hesaplamalı tekniklerden parçacık sürü optimizasyonu ve bulanık mantık yaklaşımı da kullanılmıştır.

- ***Bulanık-mantık yaklaşımı,*** Bir sistemdeki ya da kavramdaki belirsizlik bulanıklık olarak adlandırılır. Belirsiz ve kesin olmayan bilgilere dayanarak etkili sonuçlar üreten bulanık mantık, kontrol modellerini kolaylıkla gerçekleştirme olanağı sağlar. Genel küme denklemi kısmi üyeliğe izin verirken ortaya bulanıklık çıkar. Geleneksel küme denklemi Eş. 5'te, bulanık küme teorisi ise Eş. 6'da verilmiştir [24,25]. Bu çalışmada performans karşılaştırması için ortalama gecikme, kuyruk uzunluğu ve yol önceliği sisteme girdi olarak verilmiş, yeşil ışık süresi de sistemin çıktısı olarak alınmıştır. Bulanık modelde yeşil ışık süresini belirleyebilmek için Mamdani modeline göre 27 adet kural (her girdi 3 üyelik fonksiyonuna sahiptir) kullanılmıştır.

$$\mu_A(x): x \rightarrow \{0,1\}$$
$$\mu_A(x) = \left\{ \begin{array}{ll} 1 & x \in X \\ 0 & x \notin X \end{array} \right\} \quad (5)$$

$$\tilde{A} = \{(x, \mu_A(x)) \, \forall \, x \in X\}$$
$$\mu_A(x): X \rightarrow [0,1] \quad (6)$$

- ***Parçacık sürü optimizasyonu,*** 1995 yılında Kennedy ve Eberhart tarafından balıkların ve böceklerin sürü şeklinde hareketlerinden esinlenerek tasarlanmış sezgisel bir optimizasyon tekniğidir. Sürüdeki her bir bireye parçacık denir. Parçacık sürü optimizasyonundaki ana amaç, sürüdeki her bir parçacığın kendi durum ve konumunu sürü içerisindeki en optimuma göre ayarlamasıdır [25]. Algoritmanın trafik sinyalizasyonuna uyarlanmasında yeşil ışık süresinin başlangıcından süre sonuna kadarki olan zaman aralıkları sürüdeki parçacıklar olarak atanır ve sahadaki trafik bilgilerine göre en iyi süre bulunana kadar algoritma sezgisel olarak çalışır. Algoritmanın sözde kodu şu şekildedir [26]:

  *Başlangıç pozisyonları ve hızları ile başlangıç sürüsü oluşturulur.*
  *Döngü:*
  *Her parçacık için;*
  *Uygunluk değerlerini hesapla.*
  *Mevcut durumdan en iyi (pbest) yerel durum bul.*
  *Yerel en iyiler içerisinden küresel en iyi (gbest)seç.*
  *Her parçacık için;*
  *Hızı Eş.7'ye, pozisyonu ise Eş.8'e göre güncelle.*
  *Eğer maksimum iterasyona ulaşıldıysa*
  *Döngüden çık*
  *Değilse devam et.*

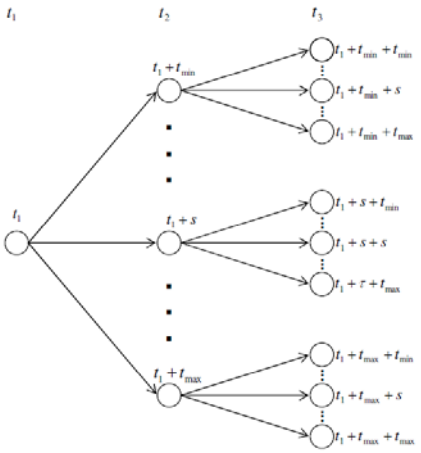
$$V_{id} = W * V_{id} + c_1 * rand_1 * (P_{id} - X_{id}) + c_2 * rand_2 * (P_{gd} - X_{gd}) \quad (7)$$

$$X_{id} = X_{id} + V_{id} \quad (8)$$

  Burada "$X_{id}$"pozisyon ve "$V_{id}$"hız değerlerini verirken, $rand_1$ ve $rand_2$ değerleri rasgele üretilmiş sayılardır. "W" atalet ağırlık değeri ve "$C_1$", "$C_2$"ölçeklendirme faktörleridir.

- ***Karınca koloni algoritması,*** karıncaların çalışma yapılarından esinlenerek geliştirilmiştir. Önlerine bir engel konulduğunda kendi salgıladıkları feromonları takip edemediklerinden, karıncalar gidebilecekleri iki yoldan birini öncelikle rastsal olarak seçmektedirler. Kısa olan yoldan birim zamandaki geçiş daha fazla olacağından bırakılan feromon miktarı da daha fazla olur. Buna bağlı olarak, zaman içerisinde kısa olan yolu tercih eden karıncaların sayısında artış olur. Belli bir süre sonra tüm karıncalar bu kısa yolu tercih ederler [27].

KKA'nın trafik sinyalizasyonuna uyarlanması işleminde ise bir faz süresi belli periyotlara ayrılır (1'er saniyelik dilimler) ve bu zaman dilimleri karıncaların hareket noktaları olarak atanır. Şekil 7'de karıncaların sinyalizasyon problemlerindeki hareketleri modellenmiştir [27].

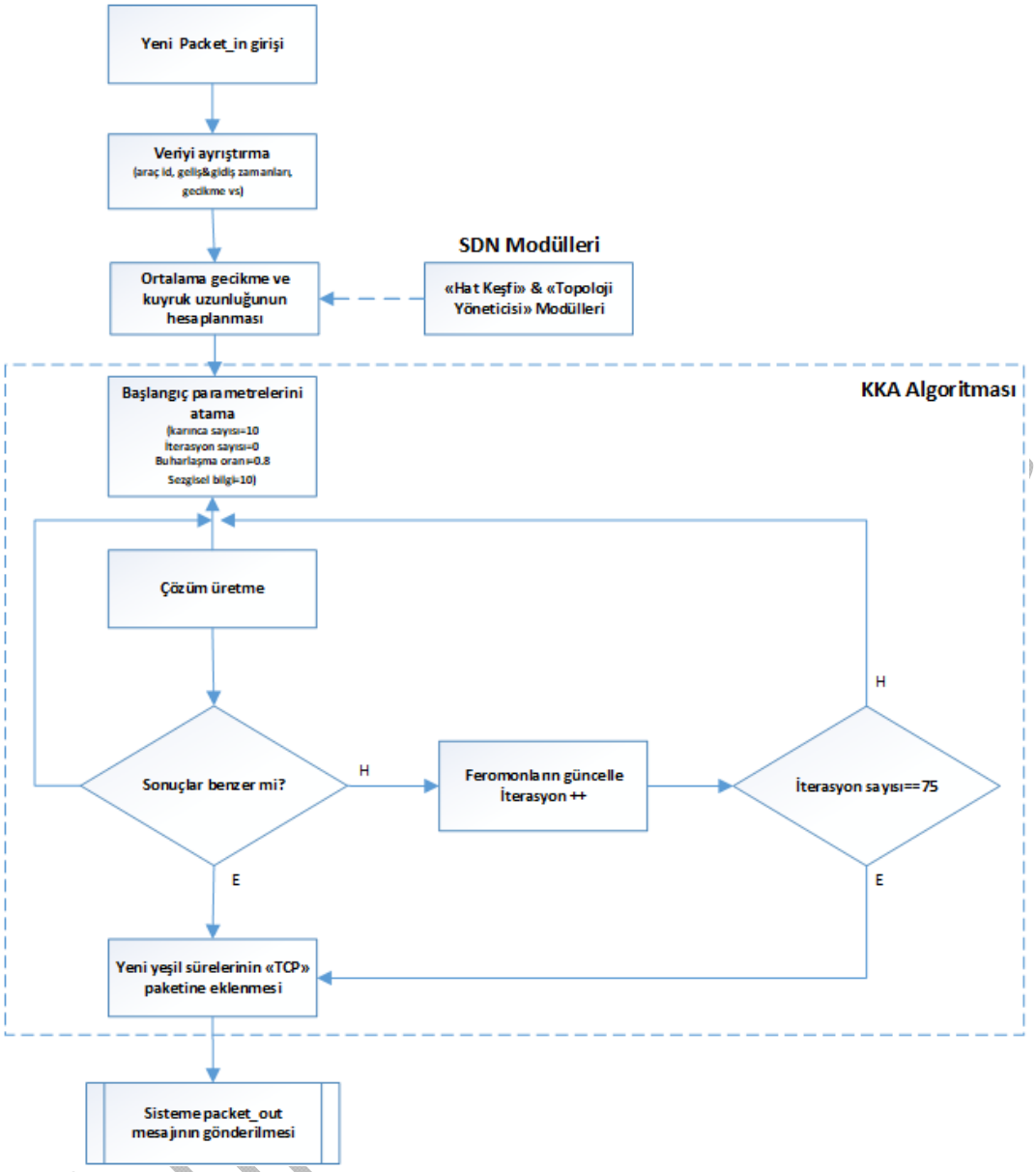


Şekil 7. Karıncaların sinyalizasyon optimizasyonundaki davranışları (Behaviours of ants in signaling problems)

Buna göre:

- Bir karınca t anında iken mümkün olan tüm düğümlere ($t + t_{min}$, $t + t_{min} + 1$….., $t + t_{max}$) gidebilir.
- Tüm karıncalar aynı anda harekete başlarlar ve sağındaki düğümlere doğru hareket ederler. Sağ taraftaki bu düğümler bir sonraki sinyal çevrim süresini ifade eder. Böylece full çevrim süresini bulabilirler.
- Herhangi bir anda veya belirlenen zamanda, $t_1$ sonraki çevrim zamanlarının ($t_2$ ve $t_3$) olası kombinasyonlarını içeren çözüm kümesi olmuş olur.

***KKA'nın Önerilen Sistemde Uygulanışı:*** Literatürdeki çalışmalar doğrultusunda, çalışmada da, kavşak içi ortalama araç gecikme sürelerinin düşürülmesi ve kavşaktan birim zaman içerisinde geçen araç sayısının arttırılması hedeflenmiştir. Bunun için Karınca kolonisi algoritmasına girdi olarak; mevcut trafik kuyruğu (iki faz için de) ve araç gecikme süreleri kullanılmıştır. Algoritma çıktı olarak ise ilgili faz için optimal yeşil ışık süresini ayarlamaktadır. Gerçek zamanlı veri işlemesi için KKA, SDN kontrolör içerisinde bir modül olarak geliştirilmiştir. Algoritmanın akış diyagramı Şekil 8'de gösterilmiştir.




Şekil 8. SDN temelli KKA'nın akış diagramı (Flow chart of the proposed ant colony algorithm with SDN modules)

Buna göre SDN anahtar cihazları ve SDN etmenli RSU'lardan gelen trafik bilgileri paket_in mesajları içerisinde Floodlight kontrolörüne geldikten sonra, paketler parçalanarak gerekli bilgiler elde edilir ve kontrolörün ağdaki düğümleri yönetmekle sorumlu olan iki modülü olan "Hat Keşfi (Link discovery)" ve "Topoloji Yöneticisi (Topology Manager)"a gönderilir. Bu modüller sayesinde bir çevrim süresi içerisinde bir yoldaki kuyruk uzunluğu ve ortalama gecikme tespit edilerek, karınca koloni algoritmasının girdileri olarak işleme alınır. Bu modüllerin KKA içerisinde nasıl kullanıldığı bölüm 4.1'de detaylı olarak anlatılmıştır. Girdiler elde edildikten sonra, o kavşak için yeni yeşil ışık sürelerinin hesaplanabilmesi için algoritmanın çözüm üretme adımında (Şekil 8'de) Eş.9'daki formül kullanılmıştır [22].

$$J_{1green}(t_1,t_2) = \frac{q(q-1)hw}{2} + \sum_{i=1}^{q}(t_1 - t_a^1) + \frac{\lambda((q-1)hw)^2}{2} + \frac{\lambda((q-1)hw)[\lambda((q-1)hw)-1]hw}{2} \quad (9)$$

"$t_1$" başlangıç zamanını ifade ederken, "$t_2$" o kavşaktaki herhangi bir çevrim zamanını ifade eder. Bu iki zaman arasında kavşaktan geçen araçların oluşturduğu kuyruk "q" ile ifade edilir. "hw" ise araçlar arasındaki boşluk

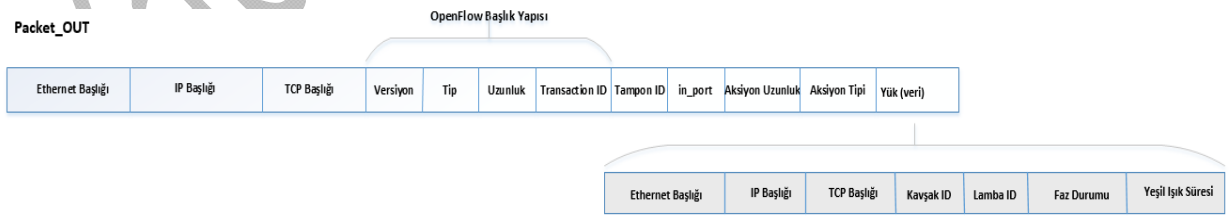
zamanını ifade eder. Bu formül karınca hareketlerine bağlı olarak her iterasyonda tekrar edilir. İterasyonlar neticesinde benzer değerlerin bulunmasıyla, algoritma sonlandırılır. Bu değerlere göre hesaplanan yeşil ışık süresi o kavşağın bir sonraki çevrim zamanı olarak atanır. Yeni yeşil ışık sürelerinin SDN ağı üzerinden kavşaktaki anahtar cihazına ve araçlara aktarımıyla ilgili bilgi bir sonraki bölümde verilecektir.

### 3.2.3 Servis dağıtımı (Service delivery)

KKA'nın çalışması neticesinde elde edilen yeni yeşil ışık durumu ve sürelerinin sistemdeki SDN anahtar, SDN etmenli RSU ve SDN etmen araçlara geri gönderilmesi gerekmektedir. SDN anahtar ve RSU cihazı trafik sinyalizasyon cihazlarını ayarlamak için, SDN etmen RSU ise SDN etmen araçlara VANET mimarisinin trafik verimliliği anlamında bilgi akışı sağlamak için bu sonuçlara ihtiyaç duyar. Kontrolör tarafından çalıştırılan KKA'nın üretmiş olduğu değerler için Şekil 9'daki gibi yeni bir TCP raw paketi oluşturulur ve Openflow protokolün "packet_out" mesajı ile içerisinde Şekil 10'daki gibi kapsüllenerek SDN anahtar cihazına gönderilir. Anahtar cihazı akış tablosuna yeni kayıt girişi yaptıktan sonra "packet_out"u hem port 1'den SDN etmen RSU cihaza gönderir hem de packet_out mesajını decapsule ederek içerisinde sistemin yeni yeşil durum ve süreleri olan TCP paketini elde eder ve de kendisine bağlı olan trafik lambalarına gönderir. RSU cihazı kendisine gelen packet_out mesajını hem 802.11p modülü vasıtasıyla sahadaki araçlara yollar, hem de kendi üzerindeki akış tablosunu güncellemiş olur.

Çalışmada önerilen SDN temelli VANET mimarisinde sahadaki araçlardan, yol kenarındaki RSU'ya, kavşak içindeki anahtar cihazdan merkezdeki kontrolör yazılımına kadar tüm düğümlere SDN ağ paradigmasının işlevselliği kazandırılmıştır. Günümüz trafik uygulamaların karmaşık sistemler olması, esnek iletişim altyapılarına ihtiyaç duymaları, SDN'nin de dinamik ölçeklenebilir, esnek ve dinamik programlanabilirlik özelliklerinden dolayı kavşak ve merkez modelleri SDN mimarisi üzerine oturtulmuştur. Sahadaki araçların ve RSU cihazların SDN etmen olarak ayarlanmasının en önemli sebebi ise, kontrolör tarafında gerçeklenen sinyalizasyon, yönlendirme, rota hesaplama, park yeri vs birçok trafik uygulamalarının sonuçlarının kolay bir şekilde yine bu ağ elemanlarına aktarılmalarının sağlanmasıdır.


Şekil 9. Kontrolör tarafından üretilen TCP paketi (TCP raw packet generated by controller)




Şekil 10. Packet-out mesajı (Packet-out message)

## 4 MODELLEME VE PERFORMANS DEĞERLENDİRMESİ (SIMULATİON AND PERFORMANCE EVALUATİON)

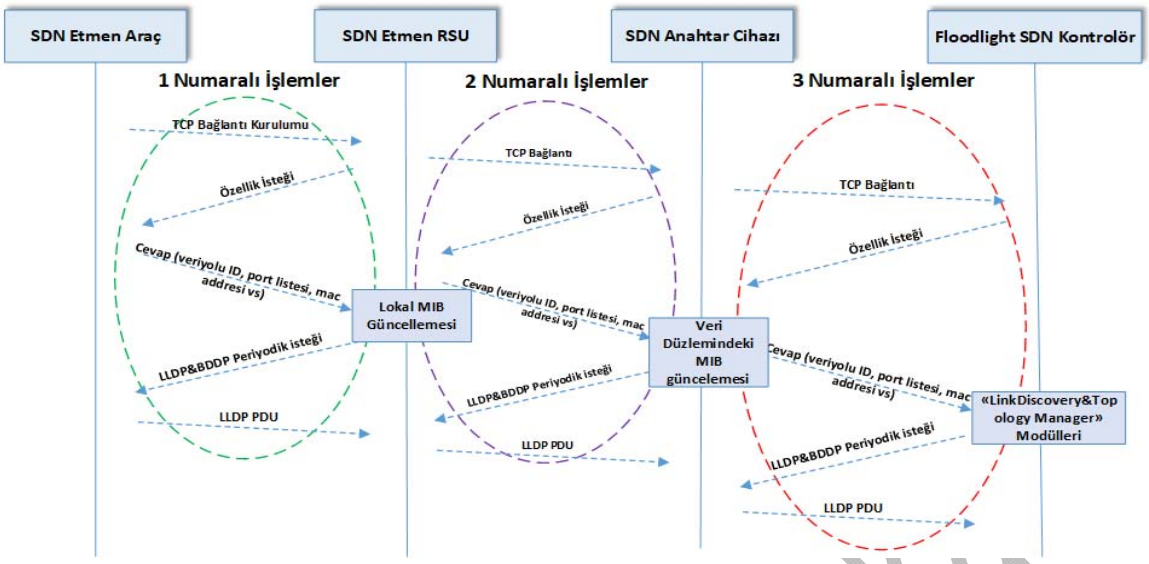
### *4.1 Modelleme (Modelling)*

Şehir-içi kavşak yönetim sistemleri için önerilen SDN temelli VANET mimarisi, SDN mimarisinin oluşturulması, kavşak modelinin oluşturulması ve trafik senaryolarının oluşturulması şeklinde 3 ana kısımda modellenmiştir.

#### *4.1.1 SDN mimarisinin oluşturulması (The creation of the SDN architecture)*

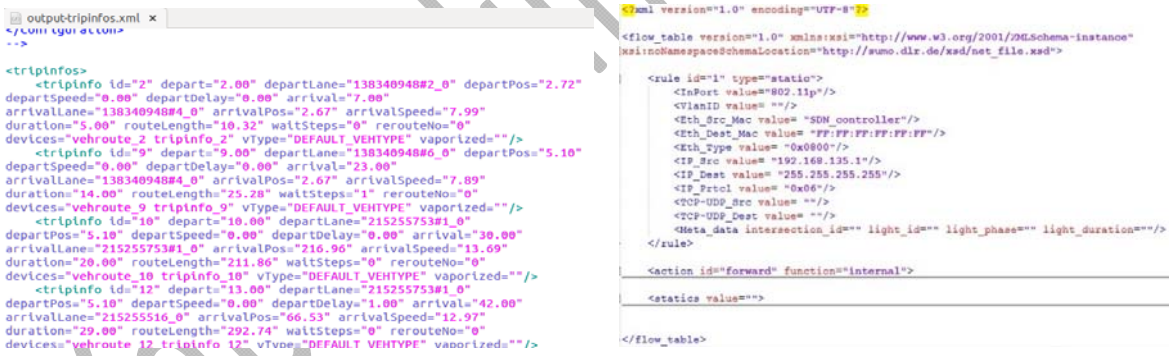
Bu kısım, araçlardan gelecek trafik verilerinin SDN etmen RSU vasıtası ile SDN anahtar cihaza ondan sonra da kontrolöre iletilmesi, kontrolör tarafından bu verilerin ilgili modüller aracılığıyla işlenmesi ve son olarak da elde edilen yeni sinyal sürelerinin sahadaki SDN etmen araçlara gönderilmesi işlemlerinden oluşur. Bu işlemler için gerçekleştirilen modelleme adımları şu şekildedir:

1. Sahadaki araçlardan gelen verilerin merkeze taşınması işlemi için MiniNet sanallaştırma platformu kullanılmıştır. Her senaryo için modellenen kavşaklar içerisindeki trafik lambaları (4 adet) ve SDN temelli RSU (1 adet), MiniNet sanallaştırma programında birer sanal host olarak atanmış ve bir SDN anahtar cihazına bağlanmıştır. SDN etmen olarak modellenen RSU sahadan topladığı bu trafik verilerini bir önceki bölümde anlatılan Şekil 4'deki TCP raw paket haline getirerek Şekil 6'daki OpenFlow packet_in mesajının içerisinde kapsüllenmesini sağlar ve Floodlight kontrolöre gönderir.
2. Sahadan gelen trafik verilerinin işlenmesi için literatürdeki çalışmaların aksine çalışmadaki karınca koloni algoritmasının girdileri olan kuyruk uzunluğu ve gecikme parametreleri, daha önceki bölümde ifade edilen matematiksel denklemler yerine Floodlight kontrolörün sağladığı "Hat Keşfi" ve "Topoloji Yöneticisi" modülleri kullanılarak hesaplanmıştır. "Hat Keşfi" modülü ile sahadaki SDN etmenli araçlara, SDN etmen RSU vasıtasıyla LLDP (Bağlantı Katmanı Keşif Protokolü - Link Layer Discovery Protocol) ve BDDP (Yayın Etki Alanı Keşif Protokolü - Broadcast Domain Discovery Protocol) ile mesajlar gönderilmektedir. Şekil 11'de LLDP mesajlarının mimari üzerindeki dizi (sequence) diyagramı verilmiştir. Buna göre SDN etmenli araç, SDN etmen RSU ile (Şekil 11-1 numaralı işlemler), veya SDN etmen RSU ,SDN anahtar cihazı ile (Şekil 11-2 numaralı işlemler) veya SDN anahtar cihazı merkezdeki kontrolör ile TCP bağlantısı kurmak istediği zaman (Şekil 11-3 numaralı işlemler), bağlantının kurulacağı cihaz istekte bulunan cihaza özelliklerini talep eden "özellik istek (feature request)" mesajını gönderir. Bağlantı kuracak olan cihazlar veri yolu tanıtıcı numarası (datapath ID), port listesi, kendisine bağlı olan host ve komşuluk bilgilerini kontrolöra "cevap (reply)" mesajı içerisinde gönderirler. Bağlantı kurulum aşamasından sonra ise kontrolör periyodik olarak (örneğin 3 sn) SDN anahtar cihazına ve onun üzerinden de SDN etmen RSU cihazına LLDP ve BDDP isteği gönderir. SDN etmen RSU da bu gelen istekleri yayınsal olarak sahadaki SDN etmen araçlara gönderir. "Topoloji Yöneticisi" modülü ise, "Hat Keşfi" modülünden elde edilen bilgilere istinaden ağın genel topolojisini çıkartır.




Şekil 11. LLDP'nin akış diyagramı (The sequence diagram of LLDP)

3. Uçtan uca SDN iletim mimarisini sağlamak için sahadaki araçların da SDN etmen mobil düğümler olarak ayarlanması gerekmektedir. Bunun için SUMO benzetim programının "additional file" özelliği kullanılarak içerisinde bir akış tablosu içeren "xml" dosyaları oluşturulmuştur. Bu akış tabloları, kontrolör tarafında araçlardan gelen verilere göre hesaplanan sinyal süre ve durumlarını araçlara iletmek ve araçların ona göre hareket etmelerini sağlamak için dizayn edilmiştir. Şekil 12.a'da çalıştırılan simülasyonlar neticesinde her bir aracın yolculuk bilgilerini içeren SUMO çıktı dosyası gösterilmiştir. Şekil 12.b'de ise araçlar için oluşturulan xml temelli akış tablosu örneği verilmiştir.

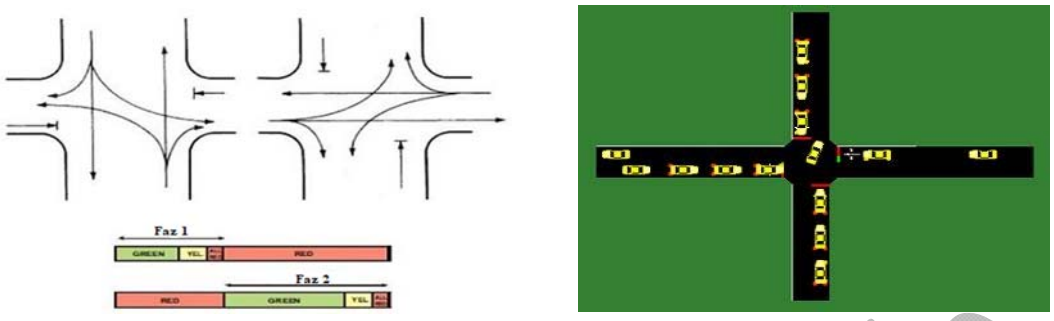



Şekil 12. a)SUMO program çıktısı (SUMO program output) b)Araçların akış tablosu xml kodu (Vehicle flow table xml codes)

### 4.1.2 *Kavşak modelinin oluşturulması (The creation of the intersection model)*

Şehir içlerindeki izole veya koordineli kavşak tiplerinde 1-8'e kadar kullanılan farklı faz çeşitleri mevcuttur. Faz sayısının artması o bölgedeki araç trafiğinin daha net yönetilebilmesi anlamına gelmekle birlikte her faz için optimal yeşil ışık süre ve sırası ayarlanacağından çok fazla bantgenişliği ve maliyet gerektirmektedir. Bunun için literatürde birçok senaryoyu kapsayabilecek 2 veya 4 fazlı sistemler tercih edilmektedir. Yapılan çalışmada Şekil

13 a'daki 2 fazlı izole kavşak sistemini ele alınmıştır. 2 farklı faz yönüne sahip olan izole kavşak altyapısının SUMO benzetim programında gösterimi ise Şekil 13 b'de gösterildiği gibidir.



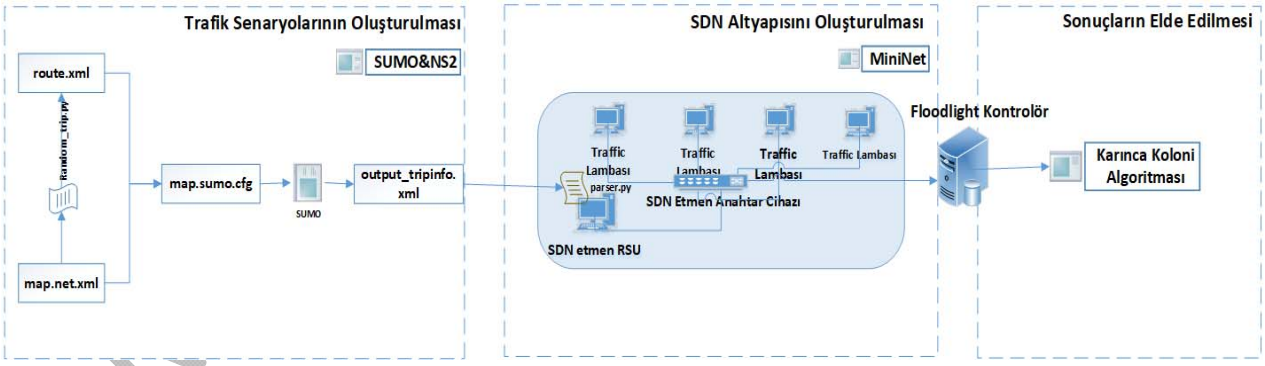

Şekil 13 a) 2 fazlı izole kavşak modeli (2 phase isolated intersection model) b) Kavşak yapısının SUMO gösterimi (The representation of intersection model in SUMO)

### *4.1.3 Trafik senaryolarinin ve simülasyonlarının oluşturulması (Construction and simulation of traffic scenarios)*

Bu adım için SUMO simülasyon programı kullanılmıştır:

1. Şekil 13'de gösterilen kavşak yapısının modellenebilmesi için gerekli harita koordinat bilgileri "OpenStreetMap" üzerinden elde edilerek "map.net.xml"dosyası oluşturulur.
2. Bu harita bilgileri, daha sonra uygulanacak her bir trafik senaryosuna göre "Random_trip.py" kodu üzerinde değişiklik yapılarak araçların hareket bilgilerini içeren "route.xml" dosyaları oluşturulur.
3. Araçların kendi aralarında veya RSU'lar ile olan iletişimlerinde 802.11p kullanılabilmesi için SUMO ile birlikte çalışan "ns2" kodu çalıştırılmıştır.
4. Son olarak ise "map.net.xml" ve "route.xml" dosyalarının refere ettiği "sumo.cfg" konfigürasyon dosyası simulator üzerinde çalıştırılır.

Şekil 14'de önerilen sistemin nasıl modellendiğinin şekli verilmiştir.




Şekil 14. Önerilen SDN temelli VANET sistem modeli (The schematized model of proposed SDN based VANET architecture)

### *4.2 Performans Değerlendirmesi (Performance Evaluation)*

Önerilen SDN temelli VANET kavşak yönetim sisteminin işlevselliğini ölçmek için de farklı trafik senaryoları üzerinde performans ölçümleri yapılmıştır. Şekil 16'daki 2 fazlı izole bir kavşak yapısı üzerinde 100 araçtan (az yoğun trafik) 500 araca (yoğun trafik) kadar olan senaryolar sırasıyla gerçeklenmiştir. Trafik yoğunlukları uniform dağılım ile oluşturulmuştur. Ayrıca 500 araçtan oluşan araçların yön ve zaman bilgilerinin binom dağılımla üretilerek simülasyon esnasında dinamik olarak değişkenlik gösteren senaryolar da oluşturulmuştur. Tüm senaryolar için sabit olarak atanan başlangıç trafik değerleri Tablo1'de gösterilmiştir.

Tablo 1. Trafik senaryolari için başlangıç parametreleri (Initial parameters for traffic scenarios)

| Trafik parametreleri | Değerler (sn) |
|---|---|
| minunum yeşil süresi | 5 |
| maximum yeşil süresi | 30 |
| kırmızı süresi | 2 |
| Araçlar arası boşluk süresi | 2 |
| periyodik LLDP ve BDDP mesajları | 3 |

Bu çalışmada, önerilen mimari üzerinde çalıştırılan SDN temelli KKA, geleneksel mimarilerde kullanılan sabit zamanlı sinyalizasyon, webster eşitliği, geleneksel karınca koloni algoritması, parçacık sürü optimizasyonu ve bulanık mantık yaklaşımı ile karşılaştırılmıştır. Karınca koloni algoritmalarında karınca sayısının artırılması daha kesin sonuçlar vereceği için çalışmada her bir trafik senaryosu için 10, 25 ve 50 adet karıncadan oluşan yapılar kullanılmıştır. Çalışmada performans karşılaştırması için 5 parametre belirlenmiştir;

- ***Ortalama araç gecikmesi;*** araçların kavşak içerisinde kaybettiği zamanı ifade eder.
- ***Kuyruk uzunluğu;*** bir yeşil ışık çevrim süresi içerisinde geçen araç sayısını ifade eder.
- ***İşlem süresi;*** Sinyalizasyon işleminin ne kadar sürede tamamlandığını ifade eder.
- ***Simulasyon süresi;*** Her bir senaryodaki tüm araçların yolculuklarının tamamladığı toplam zamanı ifade eder.
- ***Uçtan-uca gecikme;*** SDN etmen araçtan çıkan paketlerin merkezdeki sinyalizasyon hesabından sonra tekrar SDN etmen araca geri dönmesine kadar geçen süreyi ifade eder.

Şekil 15'de yapılan her bir farklı senaryoda kavşak içlerindeki araçların kavşağa giriş ve çıkışları arasındaki zamanda normal yolculuk süreleri haricinde kırmızı ışık ve trafik sıkışıklığından dolayı yaşamış oldukları ortalama gecikme süreleri (sn) hesaplanmıştır. Buna göre araç sayısının az olduğu senaryolarda yani kavşağın az yoğun (100-200 araç) olduğu durumlarda karınca koloni algoritması kullanılarak yapılan testlerde Webster ve sabit zamanlı hesaplamaya göre gecikme süreleri arasında %5-10 arasında iyileşme söz konusu iken özellikle doygun trafik senaryolarında (500 ve dinamik 500 araç) bu iyileşmenin %20-25 arasında olduğu gözlemlenmiştir. Parçacık sürü optimizasyonu ve bulanık mantık modeline göre bu iyileşme %8-11 arasında daha iyi sonuç vermektedir. Geleneksel karınca algoritması ile SDN temelli KKA arasında ise özellikle karınca sayısı arttığı zaman %5-12 aralığında bir iyileşme söz konusudur. Bu iyileşme, SDN'nin ağdaki tüm cihazlara ve düğümlere direk erişimi ve kontrolü, modüler yapısı ve de merkezi yönetim özellikleri sayesinde sağlanmıştır.

Şekil 15. Ortalama gecikme performans sonuçları (Performance results of scenario and algorithm-based average delay)

2 farklı yeşil fazın toplamı olan bir faz çevrim süresi içerisinde kavşaktan geçen araçların sayılarına oranın bulunduğu bu kuyruk uzunluğu performans testlerinin sonuçları şekil 16'da gösterilmiştir. Sabit zamanlı sistemlerde 80 sn olarak atanan bu çevrim süresi, diğer hesaplama tekniklerinde her senaryoya göre farklılık gösterdiğinden sonuçlar bir çevrim süresinde geçen araç sayısı oranı olarak hesaplanmıştır. Az yoğun senaryolarda oranlar birbirine yakın olsa da kavşaktaki araç yoğunluğu arttıkça SDN temelli KKA'nın bir çevrim süresi içerisinde kavşaktan geçen araç sayısında artış olduğu görülebilmektedir. Örneğin 100 araçlık senaryoda geleneksel karınca algoritması ile SDN temelli KKA arasında %0,02 oranında bir fark (bir çevrimde 3-4 araç) söz konusu iken, bu fark 500 araçlık ve dinamik senaryoda %1,5 oranına (bir çevrimde 7-8 araç) ulaşmaktadır.


Şekil 16. Kuyruk uzunluğu performans sonuçları (Performance results of scenario and algorithm-based queue length)

Şekil 17'de trafik sinyalizasyonu ile ilgili performans kriterlerinin haricinde kavşak yönetim sistemlerinin merkezlerindeki sunucuların performans karşılaştırılması için sinyalizasyon hesaplamaları için gereken süreler de hesaplanmıştır. Bu performans karşılaştırılmalarında geleneksel sistemler de SDN temelli sistemler de aynı fiziksel özelliklere sahip cihazlardan oluşmaktadırlar. Sunucu performans karşılaştırmaları neticesinde, sabit zamanlı ve Webster eşitliği gibi temel matematiksel denklemlere oturtulan metotların hesaplamalı tekniklere göre daha kısa sürede çözüm sunduğu görülmüştür. Fakat günümüz gelişen kavşak/trafik yönetim sistemlerinin çok bileşenli komplex yapılardan oluştuğu düşünülürse hesaplama tekniklerin sinyalizasyon sistemlerinde kullanımı artık zorunlu hale gelmiştir. Burada da özellikle karınca sayısı arttığı zaman SDN temelli KKA'nın diğer hesaplamalı tekniklere göre %16 ila %20 arasında avantajı söz konusudur.

Şekil 17. İşlem süresi performans sonuçları (Performance results of scenario and algorithm-based process time)

Şekil 18'de, kavşaktaki araçların üretmiş oldukları yayın paketlerinin önce SDN etmen RSU'ya daha sonra ise merkezdeki sunucuya gelmesi, sunucudaki sinyalizasyon hesaplaması neticesinde ise elde edilen yeni trafik bilgilerinin (yeşil ışık süresi, faz düzeni, akış bilgisi vb.) tekrar kavşak içlerindeki araçlara döndürülmesine kadar geçen süre olarak tanımlanan uçtan-uca gecikme performans sonuçları gösterilmiştir. Uçtan-uca gecikme performans karşılaştırması neticesinde, özellikle zeki optimizasyon algoritmalarında SDN iletim altyapısı sağlamış olduğu dinamiklik ve akış kontrolü ile geleneksel ağ yapılarına oranla %17-22 arasında bir iyileşme sunmaktadır. Uçtan-uca performans karşılaştırmasında Webster ve sabit zamanlı sistemlerin gecikme sürelerinin optimizasyon algoritmalarına göre daha az çıkmasının ana sebebi bir önceki şekilde gösterildiği üzere işlem sürelerinin kısa olmasından kaynaklıdır.


Şekil 18. Uçtan-uca gecikme performans sonuçları (Performance results of scenario and algorithm-based end-to-end delay)

Son olarak Şekil 19 'da ise her bir senaryodaki araçların yolculuk süreleri için geçen toplam SUMO simülasyon süreleri gözlemlenmiştir. Gerek araçların kavşak içlerindeki gecikme sürelerindeki azalma gerekse de faz süresindeki kuyruk sayılarındaki artış bu seyahat sürelerinin azalmasındaki (simülasyon zamanı) en önemli etkendir. Her bir senaryodaki araçların kavşağı terk etmelerine kadar geçen süreler dikkate alınmıştır. Düşük araç yoğunlukları için (örneğin 100 araç) simülasyon zamanında zeki sistemler, sabit zamanlı ve Webster eşitliğine göre olan uygulamalara göre %9-10 arasında bir iyileşme göstermekteyken, trafik yoğunluğunun arttığı senaryolarda bu değer %25-30 aralığına kadar çıkmaktadır. Ayrıca yoğun trafik ve değişken trafik senaryolarında SDN temelli KKA ve diğer hesaplamalı teknikler arasında da %8'lik bir iyileşme gözlenmektedir.


Şekil 19. Simülasyon süresi performans sonuçları (Performance results of scenario and algorithm-based simulation time)

## 5 SONUÇ VE GELECEK ÇALIŞMALAR (CONCLUSİON AND FUTURE WORKS)

Akıllı ulaşım sistemlerinin en önemli sorunu olan şehir-içi kavşak sinyalizasyon sistemleri karmaşık ve doğrusal olmayan bileşenlerden oluşan sistemler oldukları için bu sistemlerde iletişim altyapılarında esnek ve ölçeklenebilir yapılara, merkezde ise hesaplamalı teknikler kullanılması gerekliliği ortaya çıkmıştır. Literatürde kavşak yönetim sistemlerinin özellikle saha iletişim altyapıları için artık VANET mimarisinin efektif bir alternatif çözüm olduğu görülmüştür. Öte yandan büyük ve karmaşık ağ yapıları, veri ve kontrol düzlemlerini birbirinden ayırarak daha kolay ve dinamik yönetim sağlayan SDN ağ paradigmalarını kullanmaya başlamışlardır. Bu sebeplerden ötürü yapılan çalışmada, kavşak yönetim sistemlerinin günümüz ve gelecek trafik ihtiyaçlarını karşılamaları amacıyla, her iki yönde uçtan-uca iletim modeli sağlayan bir SDN temelli VANET mimari önerisi sunulmuştur.

Önerilen mimarinin işlevselliğini ölçmek için farklı senaryolar üzerinde yapılan performans testleri neticesinde, SDN'nin modüler yapısı ve dinamik programlama özelliği kullanılarak geliştirilen karınca algoritmasının, geleneksel sistemlerde ve hesaplamalı tekniklerde kullanılan mevcut hesaplama yöntemlerine göre (webster'a göre %25-30, zeki sistemlere göre %12-20) trafik sinyalizasyon, sunucu performans ve uçtan-uca gecikme kriterleri açısından daha iyi performans ortaya koyduğu görülmüştür. Bu başarının elde edilmesindeki en önemli faktör simülasyon ortamında kavşak içlerindeki araç ve RSU cihazlarına SDN etmen özelliği kazandırılarak, "Hat Keşfi" ve "Topolooji Yöneticisi" modülleri ile net konum ve komşuluk bilgilerinin elde edilmesidir.

Bu çalışmada, bir izole kavşak yönetim sistemi üzerinde SDN ve VANET mimarilerinin nasıl birbirlerine entegere edilebileceği ve önerilen bu sistem üzerinde veri çevrim işlemlerinin nasıl olacağı gösterilmek istenmiştir. Bundan sonraki çalışma olarak ise bu mimari üzerinde, farklı trafik senaryoları ve topolojiler (karmaşık yapıdaki koordineli kavşaklar) üzerinde VANET trafik emniyeti ve verimliliği açısından sinyalizasyon, rota ve yön hesaplama gibi konular üzerinde çalışmalar yapılacak, ayrıca bu trafik modüllerinin RSU'larda mı yoksa merkezde mi hesaplanmasının daha avantajlı olacağına bakılacaktır.

## KAYNAKLAR (REFERENCES)

[1] Zhang J., Wang F., Wang K., Lin W., Xu X., Chen C., Data-Driven Intelligent Transportation Systems: A Survey, IEEE Transactions on Intelligent Transportation Systems, pp 1624-1639, 2011

[2] Chen B., Cheng H., A Review of the Applications of Agent Technology in Traffic and Transportation Systems, IEEE Transactions on Intelligent Transportation Systems, pp 485-497, 2010

[3] Djahel S., Doolan R, Muntean G., Murphy J., A Communications-Oriented Perspective on Traffic Management Systems for Smart Cities:Challenges and Innovative Approaches, IEEE Communications Surveys & Tutorials (Volume:17 , Issue: 1), pp 125-151, 2014

[4] A. Ferreira J., Fonseca J., Introduction to Intelligent Transportation Systems, Studies in Systems, Decision and Control, vol 52, pp 1-17, 2016

[5] Qu F., Wang F., Yang L., Intelligent transportation spaces: vehicles, traffic, communications, and beyond, IEEE Communications Mag., 2010

[6] Dimitrakopoulos G., Demestichas P., Intelligent Transportation Systems, IEEE Vehicular Technology Magazine, pp 77-84, 2010

[7] Qureshi K., Abdullah A., A Survey on Intelligent Transportation Systems, Middle-East Journal of Scientific Research 15, pp 629-642, 2013

[8] Küçükmanisa A., Urhan O. "Gömülü bir platform üzerinde gerçek zamanlı şeritten ayrılma uyarı sistemi", Journal of the Faculty of Engineering and Architecture of Gazi University, vol 32, 2017

[9] Barba C., Mateos M., Soto P., Mezher A., Smart city for VANETs using warning messages, traffic statistics and intelligent traffic lights , IEEE Intelligent Vehicles Symposium (IV), pp 902-907, 2012

[10] Bai X., Ye X., Jiang H., Jun Li, A novel traffic information system for VANET based on location service, 16th ICON IEEE Networks, 2008

[11] BarrachinaJ. ,SanguesaJ. ; FogueM. ; Garrido P.; MartinezF. ; Cano J.; Calafate C.; Manzoni P., V2X-d: A vehicular-density estimation system that combines V2V and V2I communications, Wireless Days (WD), IFIP, 1-6, 2013

[12] Ku I., Lu Y., Gerla M., Ongaro F., Towards Software-Defined VANET: Architecture and Services, Ad Hoc Networking Workshop (MED-HOC-NET), 103-110, 2014

[13] Truong N., Lee G., Ghamri-Doudane Y., Software defined networking-based vehicular Adhoc Network with Fog Computing, Integrated Network Management (IM),IFIP/IEEE, 1202 - 1207, 2015

[14] Chun Liu Y., Chen C., Chakraborty S., A Software Defined Network architecture for GeoBroadcast in VANETs, Communications (ICC), IEEE, 6559 – 65, 2015

[15] Nietoa J., Albaa E., Oliverab A., Swarm intelligence for traffic light scheduling: Application to real urban areas, Engineering Applications of Artifical İntelligence, vol:25, 274-283, 2012

[16] Hea J., Houb Z., Ant colony algorithm for traffic signal timing optimization, Advances in Engineering Software, vol 43, 14-18, 2012

[17] Kammouna H., Kallela I., Casillasb J., Abrahamc A., Alimia A., Adapt-traf: An adaptive multiagent road traffic management system based on ant-hierarchical fuzzy model, Trasnportation Research Part C:Emerging Technologies, vol 42,147-167, 2014

[18] D. Mariagrazia, F. Maria Pia, Meloni Carlo. A signal timing plan formulation for urban traffic control. Control Eng Pract;14:1297–311., 2006

[19] https://www.opennetworking.org/sdn-resources/ sdn-definition

[20] P. Hunt, and D. Robertson, "The SCOOT on-line traffic signal optimization technique", Traffic engineering and control,1982.

[21] P. Lowrie, "The Sydney coordinated adaptive control systems –principles, methodology, algorithms", IEEE conf. publication, vol. 207, 1982.

[22] Renfew D., Yu Hua X., Traffic Signal Optimization Using Ant Colony Algorithm, IEEE World Congress on Computational Intelligence, 2012

[23] D'Acierno L., Gallo M., Montella B., An Ant Colony Optimisation algorithm for solving the asymmetric traffic assignment problem, European Journal of Operational Research, Volume 217, Issue 2, pp 459-469, 2012

[24] Zadeh L. A., Fuzzy Sets, Information and Control, 8, 338-353, 1965.

[25] Nieto J., Olivera A., Alba E., Optimal Cycle Program of Traffic Lights with Particle Swarm Optimization, IEEE Transaction on Evolutionary Computation, 2013

[26] Gökçe M.A, Öner E., Işık G., Traffic signal optimization with particle swarm optimization for signalized roundabouts, Sagejournal,2015

[27] M. Dorigo, and T. Stutzle, *Ant Colony optimization*, The MIT Press, 2004