

Kapasite Farkında Topoloji Kontrol Algoritmalarının Telsiz Duyurga Ağlarında Deneysel Analizi

Experimental Analysis of Capacity Aware Tree Based Topology Control Algorithms for Wireless Sensor Networks

Mustafa AŞÇI
Uluslararası Bilgisayar Enst.
Ege Üniversitesi, İzmir
mustafa.asci@ege.edu.tr
ORCID: 0000-0003-3669-1870

Can Umud İLERİ
Uluslararası Bilgisayar Enst.
Ege Üniversitesi, İzmir
can.umut.ileri@ege.edu.tr
ORCID: 0000-0003-4136-9421

Orhan DAĞDEVİREN
Uluslararası Bilgisayar Enst.
Ege Üniversitesi, İzmir
orhan.dagdeviren@ege.edu.tr
ORCID: 0000-0001-8789-5086

Öz

Haberleşme, telsiz duyurga ağları (TDA'lar) üzerinde çalışan duyurga düğümlerinin enerji tüketiminde en önemli etmendir. Haberleşmeyi en aza indirip, enerji etkinliği sağlamak amacıyla yoğun bağlı ağlar, seyrek bağlı bir ağa dönüştürülür. Bu dönüşüm için kullanılan yöntemlerden biri de topoloji kontrolüdür. Topoloji kontrolü yöntemiyle genelde TDA'lar için kapsayan ağaç oluşturulmaktadır. Kapsayan ağaçlardan kapasite kısıtını sağlayan, en düşük maliyetli ağacı bulmayı hedefleyen problem, kapasite kısıtlı en küçük ağaç (KEKA) problemidir. Alt ağaçların arasındaki yük dengesi, ağdaki mesaj sayısını ve enerji etkinliğini etkilemektedir. Bu çalışmada, TDA'lar üzerinde KEKA algoritmalarının performansı ve yük dengesi analiz edilmiştir. Esau-Williams algoritması referans alınarak geliştirilen merkezi CENTEW ve dağıtık MCO algoritmaları TOSSIM simülatöründe üzerinde yük dengesi, gönderilen ve alınan mesaj boyutu, harcanan enerji ve geçen zaman kapsamlarında karşılaştırılmıştır. 250 düğümlük ağlar üzerinde yapılan deneysel sonuçlara göre CENTEW daha az zaman harcamasına rağmen MCO, 3,98 kat daha az enerji kullanmaktadır. Dağıtık KEKA yaklaşımının enerji-etkin olduğu ve yük dengesini sağladığı görülmüştür.

Anahtar Sözcükler: Telsiz Duyurga Ağları, Kapasite Kısıtlı Kapsayan En Küçük Ağaç Problemi, Enerji Etkin ve Dengeli Ağlar, Ağ Tasarımı, Topoloji Kontrolü, Esau-Williams Sezgiseli

Abstract

Communication is the most important factor in the energy consumption of sensor nodes running on wireless sensor networks (WSNs). Densely connected networks are transformed into a sparsely connected network to minimize communication and provide energy efficiency. One of the methods used for the transformation is topology control. Topology control method generally provides a spanning tree for WSNs. The problem that aims to find the minimum spanning tree that provides capacity constraint is the capacitated minimum spanning tree (CMST) problem. Balancing the loads of subtrees affects the number of messages in the network and henceforth the energy-efficiency. In this study, we analyze the performance and load-balancing performance between subtrees of CMST algorithms on WSNs. The central CENTEW and distributed MCO algorithms developed based on the Esau-Williams algorithm are compared in terms of load balancing performance, sent and received message size, spent energy and elapsed time on TOSSIM simulator. According to the experimental results on 250-node networks, CENTEW uses less than 3.98 times less energy than MCO, although it consumes less time. The distributed CMST approach is energy-efficient and balances load more evenly.

Keywords: Wireless Sensor Networks, Capacitated Minimum Spanning Tree Problem, Energy Efficient and Load Balanced Networks, Network Design, Topology Control, Esau-Williams Heuristic

Gönderme ve kabul tarihi: 29.04.2020 - 30.11.2020

Makale türü: Araştırma

1. Giriş

Telsiz Duyurga Ağlar (TDA'lar – İng. wireless sensor networks), çevresel ortamlarda duyurgalar sayesinde bilgi toplayabilen, bilgileri telsiz iletişim ile birbirleriyle paylaşabilen ve birbirlerinden bağımsız olarak çalışabilen duyurga düğümlerin oluşturduğu telsiz ağlardır [1]. Genellikle bilgi bu ağlarda çıkış düğümünde toplanır. TDA'lar askeri, sağlık ve çevre gözlemlene uygulamalarında kullanılmaktadır. TDA'yı oluşturan düğümlerde genellikle güç kaynağı olarak pil kullanılmasından dolayı; TDA'larda yaşam ömrünü uzatmak, enerjiyi etkin kullanmak önemli bir problemdir. Düğümlerin enerji tüketimini etkileyen birincil etken ise haberleşmedir. Bu sebeple haberleşmenin tasarımı, enerji etkinliği için oldukça önemlidir.

Duyurga düğümler, birbirleriyle yoğun bağlı olduklarından mesaj çakışmaları fazla enerji kaybına neden olmaktadır. Topoloji kontrolü kullanılarak yoğun bağlı bir ağ, daha seyrek bağlı bir ağa dönüştürülerek enerji kayıpları engellenirken, düğümlerin iletim alanını azaltıp enerji etkinliği de sağlanmaktadır. Topoloji kontrolünün en temel yöntemlerinden biri kapsayan ağaç oluşturmaktır. Kapsayan ağaç yönlendirme altyapısı sağlamaktadır. V düğümlerin kümesi, E kenarların kümesi, ω ise $E \rightarrow R^+$ kenarlara ait ağırlık fonksiyonudur. $G(V, E, \omega)$ çizgesinde $T(V, E')$ bütün düğümleri içeren kapsayan ağaçtır. r kök düğümüne bağlı her bir alt ağaçtaki düğüm sayısı önceden tanımlanmış, c kapasitesini geçmeyen kapsayan ağaçlar arasından en düşük maliyetli ağacı bulma problemi *kapasite kısıtlı en küçük ağaç* (KEKA) olarak adlandırılır. Şekil 1'de KEKA örneği verilmiştir. Çizelge 1'de KEKA probleminin matematiksel formülü verilmiştir. Matematiksel formülde w ağırlık, C maksimum kapasite değeri, x iki düğüm arasındaki bağlantı, z düğümün kapasitesi, K alt ağaçların kümesi, S kök düğüme bağlı alt ağaçların kümesi, T komşu düğümlerin kümesi olarak kabul edilmiştir.

KEKA probleminin karmaşıklık sınıfı C kapasitesine göre değişmektedir. C kapasitesi $3 \leq C \leq \frac{n}{2}$, $n \in |V^+|$ olduğu durumlarda problemin karmaşıklık sınıfı NP-ZOR'dur [2], [3]. Özel C kapasite değerleri için probleme polinom zamanda çözüm üretmektedir [3]. Özel C kapasite değerleri aşağıdaki gibidir:

- $C = 1$ ise problemin çözümü, yıldız çözümdür (İng. star solution). Bütün düğümler kök düğüme direkt bağlıdır.
- $C = 2$ ise problemin çözümü, mükemmel eşleme probleminin çözümüdür [2].
- $C \geq n$ ise problemin çözümü, EKA probleminin çözümüdür.

KEKA oluşturularak yüksek iletim gücüne sahip olan düğümlerin, iletim güçleri ağaç yapısına göre azaltılarak enerji-etkin iletişim kanalları seçilir. Enerji etkinliğe ek olarak kapasite kısıtıyla KEKA'ya ait olan alt ağaçlar arasında yük dengesi sağlanır. KEKA probleminin çözümü için literatürde birçok çalışma olmasına rağmen bilgilerimiz dahilinde probleme, TDA üzerinde çözüm sunan birkaç çalışma mevcuttur [4].

Çizelge-1: KEKA problemi matematiksel formülü

Karar Değişkenleri	$x_{ij} = 1 \Leftrightarrow i \text{ ve } j \in \text{kapsayan ağaç}$ $z_i^k = 1 \Leftrightarrow i \in k' \text{inci alt kapsayan ağaç}$
Amaç	$\min \sum_{ij \in E} w_{ij} x_{ij}$
Kısıtlamalar	$\sum_{\forall ij \in E z_i^k = z_j^k = 1} x_{ij} =$ $\sum_{\forall i \in V} z_i^k - 1, \forall k \in \{1, \dots, K\}$ $\sum_{\forall ij \in E z_i^k = z_j^k = 1} x_{ij} =$ $\sum_{\forall i \in S} z_i^k - 1, \forall k \in \{i z_i^k = 1\}$ $\forall k \in \{1, \dots, K\}$ $\sum_{i \in V} z_i^k \leq C, \quad \forall k \in \{2, \dots, K\}$ $z_r^1 = 1 \text{ ve } \sum_{i \in V} z_i^1 = 1$ $\sum_{i \in V} z_i^k \geq 1, \quad \forall k \in \{1, \dots, K\}$ $\sum_{j \in T(i) i=r, z_j^k = 1} x_{ij} = 1, \quad \forall k \in \{2, \dots, K\}$ $x_{ij}, z_i^k \in \{0, 1\}, \quad \forall ij \in E$

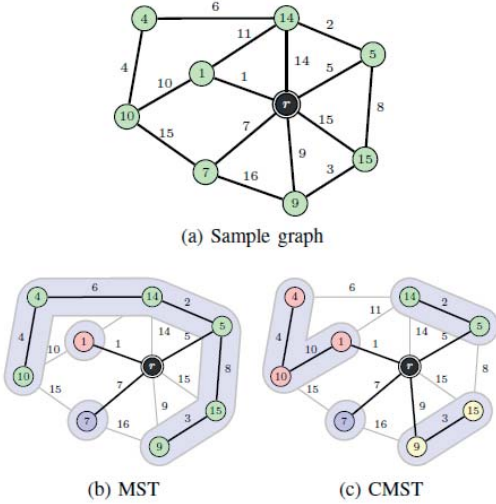
Bu çalışmada TDA'da KEKA algoritmalarının performansı değerlendirilmiştir. Bilgilerimiz dahilinde bu çalışma TDA'da KEKA algoritmalarının performans değerlendirilmesi konusunda yapılan ilk çalışmadır. Esau-Williams

algoritmasından türetilen merkezi (CENTEW) ve dağıtık (MCO [4]) algoritmalarla çalışılmıştır. [4]'te ek olarak kapasite kısıtı 5, 10, 15 ve 20'den 5, 10, 15, 20, 30, 40 ve 50 olarak çıkarılmış ve algoritmalar TOSSIM simülatörü üzerinden farklı ağ konfigürasyonlarıyla çalıştırılmıştır. Oluşan ağaçların dengeleri detaylı olarak incelenmiştir. KEKA problemini çözen merkezi ve dağıtık algoritmaların performansları, ağ metriklerine göre detaylı olarak karşılaştırılmıştır. Bölüm 2'de KEKA problemini çözen algoritmalarla ilgili yapılmış çalışmalar özetlenmiştir. Bölüm 3'de bu çalışmada uygulanan algoritmalar anlatılmıştır. Bölüm 4' de algoritmaların simülasyon sonuçlarına göre karşılaştırılmasından elde edilen performans değerlendirilmesi açıklanmıştır. Bölüm 5'de çalışmadan elde edilen sonuçlar verilmiştir.

2. İlgili Çalışmalar

Topoloji kontrolü, TDA literatüründe enerji tüketimi ve iletim gücü azaltılarak enerji etkinliği sağlamak açısından çalışılan önemli bir konudur [5], [6]. TDA üzerinde topoloji kontrol teknikleri ikiye ayrılmaktadır. Bunlar; homojen ve heterojen tekniklerdir [7], [8]. Düğümün iletim alanlarının homojen tekniklerde eşit olduğu varsayılmıştır. Düğümün iletim alanları heterojen tekniklerde farklı olabilmektedir [9], [10]. Bu çalışmada heterojen teknikte analiz edilen algoritmalar kullanılmıştır.

KEKA probleminin karmaşıklık sınıfı NP-ZOR olduğundan problemin çözümü üzerine araştırmacılar tarafından çok sayıda çalışma yapılmıştır. [11]'de Prim ve Kruskal algoritmalarında kapasite kısıt kontrolü eklenerek değişiklikler yapılmıştır. [12]'de KEKA problemini basit, hızlı ve etkin olarak çözen en önemli sezgisel algoritmalarından biri olan Esau-Williams (E-W) algoritması verilmiştir. Ağırlıklı yönsüz bir çizgede E-W algoritması, her düğümün, çıkış düğümüne direkt bağlı olarak oluşturulan kapsayan ağaç olduğunu varsayarak çözüme başlar. Bu ağaç "yıldız topoloji" olarak da tanımlanmaktadır. Her düğüm için E-W algoritması tarafından ödünleşme (İng. tradeoff) değeri hesaplanır. Ödünleşme değeri; düğümün ait olduğu alt ağacın o düğüm üzerinden başka bir alt ağaca bağlanması halinde, çıkış düğümüyle iletişim kurmasının toplam ağaç maliyetinde yani iletişim maliyetinde oluşacak artışı göstermektedir. Negatif ödünleşme değeri; iki alt ağacın, negatif ödünleşme değerine ait olan düğüm üzerinden birleştiklerinde ağacın toplam maliyetinin azalacağı anlamına gelmektedir. En düşük ödünleşme değerine sahip düğüm, algoritma tarafından seçilir. Seçilen düğüm üzerinden iki alt ağaç birleştiğinde oluşan yeni alt ağacın toplam düğüm sayısı, kapasite kısıtını (c) aşmıyorsa iki alt ağaç birleştirilir. Oluşan yeni alt ağaçtaki her düğüm için yeni ödünleşme değeri hesaplanır. Eğer kapasite kısıtı aşılmıyorsa, alt ağaçlar birleştirilmez ve seçilen düğüm için yeni bir ödünleşme değeri hesaplanır. Ödünleşme değer hesaplaması her düğüm için negatif ödünleşme değeri bulunmayana kadar devam eder. E-W algoritmasının sonucunda elde edilen kapsayan ağaç KEKA çözümüdür. [13]'de E-W algoritması kullanılarak KEKA problemi için çözüm oluşturan ve elde edilen çözüm üzerinde yerel arama yapmak amacıyla düğüm değiştirme yöntemi



Şekil-1: En küçük kapsayan ağaç ve kapasite kısıtlı en küçük kapsayan ağaç. (a)'da örnek çizge verilmiştir. Örnek çizgenin tam çizge olduğu ve görünmeyen kenarların ağırlıklarının çok büyük (örn. 1000) olduğu varsayılmaktadır. Koyu düğüm, kök düğümdür. Aynı alt ağaca ait olan düğümlerin renkleri aynıdır. Alt ağaçların arka planları daha koyu olarak gösterilmiştir. (b)'de en küçük kapsayan ağaç verilmiştir. 3 alt ağaç vardır. (c)'de kapasite kısıtlı en küçük kapsayan ağaç verilmiştir. Kapasite kısıtı $C=3$ 'dür. 4 alt ağaç vardır.

kullanılmaktadır. E-W algoritmasını geliştirmeyi amaçlayan veya E-W algoritmasını direkt kullanan birçok sezgisel algoritma geliştirilmiştir [14], [15]. KEKA probleminin çözümü için komşuluk arama, kenar değiştirme yöntemi, genetik algoritma, ağgözlü algoritma, karınca koloni algoritması ve bulanık mantık gibi başka sezgisel çözümlerde geliştirilmiştir. KEKA probleminin çok seviyeli versiyonu için melez (hibrit) bir algoritma geliştirilmiştir [16]. Sezgisel çözümlerden bazıları kenar değiştirme yöntemi [17], bulanık mantık [18] ve ağgözlü algoritmadır [19]. Komşuluk arama yöntemiyle KEKA problemine çözümler önerilmiştir [20], [21]. Karınca Koloni algoritması bütün düğümlerin talep ettiği kapasite değerleri 1 olduğu durum için önerilmiştir [22]. İkinci derece algoritma ve bellek özellikleri kullanılarak algoritma durma kriterine girse bile kapasite kısıtlı kapsayan ağaç aramaya devam algoritma geliştirilmiştir [23]. Genetik algoritmalar kullanılarak KEKA problemi için çözümler tasarlanmıştır [24], [25].

Kapasite kısıtlı en küçük kapsayan ağaç probleminin çözümü için literatürde birçok merkezi algoritma olmasına rağmen bilgilerimiz dahilinde TDA'da sadece 1 tane dağıtık algoritma (MCO algoritması) mevcuttur [4]. Bu çalışmada dağıtık algoritmanın, merkezi algoritmayla birlikte performansı değerlendirilmiştir ve algoritmaların etkinlikleri detaylı olarak karşılaştırılmıştır.

3. Algoritmalar ve Teorik Analiz

Bu bölümde CENTEW ve MCO algoritmaları tanıtılmıştır ve algoritmaların karmaşıklık analizleri verilmiştir.

3.1. CENTEW Algoritması

CENTEW algoritmasında her bir düğüm, komşuluk bilgilerini çıkış düğümüne gönderir. Çıkış düğümü, E-W algoritmasını merkezi olarak çalıştırır. Algoritma sonucunda elde edilen bilgiler, diğer düğümlerle paylaşılır. n düğüm sayısı için mesaj boyutu $O(n \log n)$ olmaktadır. Algoritmanın toplam bit karmaşıklığı $O(n^2 \log n)$ ve alan karmaşıklığı ise $O(n \log n)$ 'dir. Her bir düğüm, bütün komşuluk bilgilerini çıkış düğümüyle paylaşmak zorunda olduğundan bu çözüm yüksek seviyeli iletişim ve enerji maliyetine neden olmaktadır. Algoritma komşu düğümlerden bilgi alıp, düğümlere GÜNCELLE ya da BİTTİ mesajları gönderdiğinden

algoritmanın mesaj karmaşıklığı $O(m)$ 'dir. E-W algoritmasının zaman karmaşıklığı $O(n^2 \log n)$ [12]'dir. CENTEW algoritması, E-W algoritmasını kullandığı için CENTEW algoritmasının zaman karmaşıklığı da $O(n^2 \log n)$ 'dir.

3.2. MCO Algoritması

MCO algoritmasının ilk adımında sıradan düğümler, bütün komşuluk bilgilerini göndermek yerine sadece en düşük ağırlığa sahip olan komşusunun bilgilerini çıkış düğümüne göndermektedir. Çıkış düğümü ağaçtaki değişiklikleri kontrol eder ve alt ağaçlardaki değişiklikleri, kapasite kısıtına göre değerlendirip kapasitenin aşılmasını engeller. Çıkış düğümünde KEKA bilgileri bir tabloda tutulur. Düğümlerden alınan bilgilerle her düğüm için ödünleşme değeri hesaplanır ve birleştirilecek alt ağaçlara karar verilir. MCO algoritmasında kapasite kısıtından dolayı alt ağaçlar birleşmediyse $O(\log n)$ bit uzunluğundaki mesajlarla iletişim kurar. Algoritmanın toplam bit karmaşıklığı $O(n \log n)$ 'dir. Algoritmanın alan karmaşıklığı ise $O(\log n)$ 'dir. Çözüm bulunduktan sonra çıkış düğümü, diğer düğümleri bilgilendirir. Düğümlerin ebeveyn bilgileri değişmiş ise GÜNCELLE mesajı içerisinde ebeveyn id bilgisi gönderir. Çıkış düğümüne direkt bağlı düğümlere algoritmanın sonlandığını bildirmek amacıyla BİTTİ mesajı gönderir. MCO algoritması DEVAM, GÜNCELLE ya da BİTTİ mesajlarını alır ve EK-AĞIRLIK mesajlarını gönderir.

Mesaj tiplerinin detaylı açıklamaları aşağıda sunulmuştur:

- **EK-AĞIRLIK:** Sıradan düğümler tarafından çıkış düğümüne gönderilen mesaj tipidir. Mesaj içerisinde mesajı gönderen düğümün, komşuları arasındaki en küçük ağırlığa sahip düğümün numarası ve bağlantı ağırlığını gönderilmektedir.
- **DEVAM:** Çıkış düğümünden sıradan düğümlere, bir sonraki en küçük ağırlığa sahip düğüm bilgisini istemek için gönderilen mesaj tipidir.
- **GÜNCELLE:** Çıkış düğümünden sıradan düğümlere, düğümlerin ebeveyn bilgilerini güncellemek ve algoritmanın sonlandığını bildirmek için kullanılan mesaj tipidir. Mesaj içerisinde yeni ebeveyn düğüm bilgisi gönderilmektedir.

- **BİTTİ:** Çıkış düğümünden sıradan düğümlere, algoritmanın sonlandığını bildirmek için gönderilen mesaj tipidir. Bu mesajı alan düğümler çıkış düğümüne direk bağlıdır.

Algoritmanın mesaj karmaşıklığı $O(m)$ 'dir. Algoritmanın zaman karmaşıklığı, çıkış düğümü için $O(n)$ 'dir. Sıradan düğümlerin zaman karmaşıklığı ise kullanılan sıralama algoritmasıyla aynıdır. MCO algoritması, sıradan düğümler için hızlı sıralama algoritmasını kullanmaktadır.

Algoritmanın çıkış düğümü için sözde kodu Çizelge-2'de verilmiştir. Çizelge-3'de MCO algoritmasının sıradan düğümler için sözde kodu verilmiştir.

3.2.1 MCO Algoritmasının Çalışması

Şekil 2'deki örnek topoloji üzerinde maksimum kapasite 4, düğüm 0 çıkış düğümü olarak kabul edilerek MCO algoritması çalıştırılmıştır.

Çizelge-2: Çıkış düğümü için MCO algoritması

Adım 1: EK-AĞIRLIK mesajlarını topla. Gelen bilgilere göre KEKA tablosundaki ödünleşme değerlerini güncelle.

Adım 2: Maksimum kapasiteye ulaşan alt ağaçlardaki düğümlerin, KEKA tablosundaki ödünleşme değerlerini SONSUZ olarak belirle.

Adım 3: En küçük ödünleşme değerine sahip düğümü (SeçilenDüğüm) belirle. En küçük ödünleşme değeri negatif değil ise çocuklar listesinde olmayan düğümlere GÜNCELLE, diğer tüm düğümlere BİTTİ mesajı gönder ve algoritmayı sonlandır.

Adım 4: SeçilenDüğüm'ün bağlı olduğu alt ağaçtaki (SeçilenAltAğaç) düğümler ile bu düğümün en küçük ağırlığa sahip komşusunun bağlı olduğu alt ağaçtaki (BağlananAltAğaç) düğümlerin toplamı, kapasite kısıtını aşıyorsa veya döngü kısıtı sağlanmıyorsa 1. adıma geri dön.

Adım 5: SeçilenAltAğaç'taki tüm düğümleri çocuklar listesinden çıkar. SeçilenAltAğaç'taki düğümleri, BağlananAltAğaç'a ekle ve KEKA tablosunu yeni duruma göre güncelle. SeçilenDüğüm'e DEVAM mesajı gönder. 1. adıma geri dön.

Çizelge-3: Sıradan düğümler için MCO algoritması

Adım 1: Komşuları ağırlıklarına göre artan şekilde sırala.

Adım 2: En küçük ağırlığa sahip komşuyu (Ek-Komşu), EK-AĞIRLIK mesajı ile çıkış düğümüne gönder.

Adım 3: Çıkış düğümünden gelen her mesaj için, mesaj tipi:

DEVAM ise: Sıralı komşu listesinde sıradaki komşunun numarasını, çıkış düğümüne EK-AĞIRLIK mesajı ile gönder.

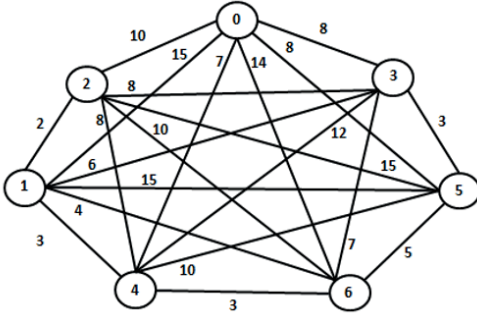
GÜNCELLE ise: Ebeveyn bilgisini güncelle ve algoritmayı sonlandır.

BİTTİ ise: Çıkış düğümünü ebeveyn olarak belirle ve algoritmayı sonlandır.

Çıkış düğümün kapasitesi 0 ve sıradan düğümlerin kapasitesi 1 olarak kabul edilmiştir. Senaryo kapsamında mesaj çıkışmaları ve mesaj kayıpları oluşmamaktadır. İlk RAUNT başladığında sıradan düğümler kendi ödünleşme değerlerini hesaplar. Ödünleşme değeri negatif olan düğümler, çıkış düğümüne EK-AĞIRLIK mesajı gönderir. Algoritma kapsamında zaman bölümlenmeli çoklu erişim (ZBÇE) protokolü kullanılmaktadır. Her düğüm, kendi sahibi olduğu zaman parçası (ZP) aralığında çıkış düğümüne, EK-AĞIRLIK mesajı gönderir. ZP sırası düğüm numarasına göre verilmektedir. Düğüm 1, ZPye sahip olan ilk düğümdür. Düğüm 1, çıkış düğümüne EK-AĞIRLIK(2,2) mesajını gönderir. Çıkış düğümü, EK-AĞIRLIK mesajını aldığı anda düğüm 1'in ödünleşme değerini (TradeOff(1)) Ek-Ağırlık(1) - Ağırlık(Komp(1), çıkış düğüm bağlantı ağırlığı(çıkış)) olarak hesaplar. Komp fonksiyonu düğümün bağlı olduğu alt ağacın çıkış düğümüne bağlı olduğu bağlantının ağırlığını vermektedir. Komp(1) değeri 1'dir. TradeOff(1) değeri $2 - 15 = -13$ 'dür. Düğüm 2 EK-AĞIRLIK(2,1), düğüm 3 EK-AĞIRLIK(3,5), düğüm 4 EK-AĞIRLIK(3,1), düğüm 5 EK-AĞIRLIK(3,3) ve düğüm 6 EK-AĞIRLIK(3,4) mesajlarını, düğümlerin sahip olduğu ZP aralığında çıkış düğümüne gönderirler.

İkinci RAUNT başladığında en küçük ödünleşme değerini veren düğüm 1, en küçük ödünleşme değerine sahip düğüm (SeçilmişDüğüm) değişkenine atanır. En küçük ödünleşme değeri hesaplandıktan sonra kapasite kısıt kontrolü

Yapılmaktadır. Düğüm 1'in kapasitesi ve en küçük ağırlığa sahip komşu (Ek-Komşusunun (düğüm 2)) düğümün kapasiteleri toplandığında oluşacak alt ağacın toplam kapasitesi 2 olmaktadır. Böylelikle kapasite kısıtı sağlanmaktadır. Kapasite kısıtı sağlandıktan sonra alt ağaç içerisindeki döngü kısıtı kontrol edilmektedir. Bağlantı yapılacak düğümün aynı alt ağaçta bulunmaması gerekmektedir. Düğüm 1 ve düğüm 2 hem kapasite hem de döngü kısıtlarını sağladıkları için düğüm 1, düğüm 2'ye bağlanır. Bağlanan AltAğaçlar listesi ve KEKA tablosu güncellenir.



Şekil-2: Topoloji örneği

Seçilen düğüm Çocuklar listesinden çıkartılır ve seçilen düğümün sahip olduğu ZP aralığında Çıkış düğümü tarafından DEVAM mesajı gönderilir. Düğüm 1, DEVAM mesajını aldığı anda sahip olduğu ZP aralığında çıkış düğümüne EK-AĞIRLIK(3,4) mesajı gönderir. Çıkış düğümü, EK-AĞIRLIK mesajını aldığı anda TradeOff(1) değerini Ek-Ağırlık(1) - Ağırlık(Komp(1), çıkış) olarak güncellenir. Komp(1) değeri artık 2'dir. TradeOff(1) değeri $3 - 10 = -7$ 'dir.

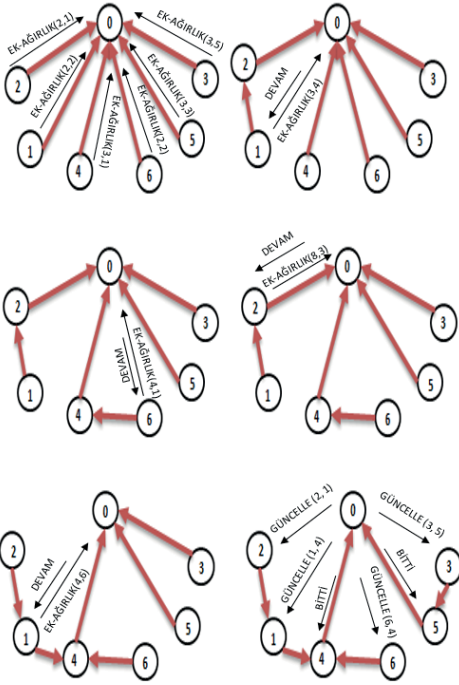
3.RAUNT başladığında düğüm 6, SeçilmişDüğüm olarak atanır. SeçilmişDüğüm ve Ek-Komşusunun kapasiteleri toplamı 2'dir. Kapasite kısıtı ve döngü kısıtı SeçilmişDüğüm için sağlanmaktadır. Düğüm 6, düğüm 4'e bağlanır. Bağlanan AltAğaçlar listesi ve KEKA tablosu güncellenir. Seçilen düğüm Çocuklar listesinden çıkartılır. Çıkış düğümü, düğüm 6'ya DEVAM mesajı gönderir. DEVAM mesajını alan düğüm 6, sahip olduğu ZP aralığında çıkış düğümüne EK-AĞIRLIK(4,1) mesajı gönderir. Çıkış düğümü EK-AĞIRLIK mesajını aldığı anda TradeOff(6) değeri Ek-Ağırlık(6) - Ağırlık(Komp(6), çıkış) olarak güncellenir.

4.RAUNT başladığında düğüm 2, en küçük ödünleşme değerini vermektedir. Düğüm 2, SeçilmişDüğüm olarak atanır. Düğüm 2 ve Ek-Komşusu aynı alt ağaçta olduğu için döngü kısıtı sağlanmamaktadır. Çıkış düğümü tarafından düğüm 2'ye DEVAM mesajı gönderilir. Düğüm 2, DEVAM mesajını aldığı anda sahip olduğu ZP aralığında çıkış düğümüne EK-AĞIRLIK(8,3) mesajı gönderir. Çıkış düğümü EK-AĞIRLIK mesajını aldığı anda TradeOff(2) değeri güncellenir.

5.RAUNT başladığında düğüm 1, en küçük ödünleşme değerini vermektedir. Düğüm 2, SeçilmişDüğüm olarak atanır. Düğüm 1'in Ek-Komşuna bağlanırsa oluşacak alt ağacın toplam kapasitesi 4'dür. Kapasite ve döngü kısıtları sağlandığı için düğüm 1'in düğüm 4'e bağlanması kabul edilir. Seçilen düğüm, Çocuklar listesinde olmadığından seçilen düğümün bulunduğu alt ağaçtaki diğer düğümlerin ebeveyn bilgileri güncellenir. Düğüm 2'nin ebeveyn bilgisi, düğüm 1 olarak güncellenir. Düğüm 1, düğüm 4 üzerinden alt ağaca bağlandığından TradeOff(2) güncellenir. Düğüm 2 Çocuklar listesinden çıkartılır. Düğüm 1'e Çıkış düğümü tarafından DEVAM mesajı gönderilir. DEVAM mesajını alan düğüm 1, sahip olduğu ZP aralığında çıkış düğümüne EK-AĞIRLIK(4,6) mesajı gönderir. Çıkış düğümü EK-AĞIRLIK mesajını aldığı anda TradeOff(1) değeri güncellenir.

6.RAUNT başladığında düğüm 4 üzerinden çıkış düğümüne bağlanan alt ağaç, maksimum kapasite kısıtını aşmaktadır. TradeOff(1), TradeOff(2), TradeOff(4) ve TradeOff(6) değerleri sonsuz(∞) olarak atanır. Düğüm 3 SeçilmişDüğüm olarak atanır. Düğüm 3'ün kapasitesi ve Ek-Komşusunun kapasiteleri toplamı 2'dir. SeçilmişDüğüm kapasite ve döngü kısıtları sağlandığı için düğüm 3, düğüm 5'e bağlanır. Bağlanan AltAğaçlar listesi ve KEKA tablosu güncellenir. Seçilen düğüm, Çocuklar listesinden çıkartılır. Seçilen düğümün çıkış düğümü tarafından DEVAM mesajı gönderilir. DEVAM mesajını alan düğüm 3, sahip olduğu ZP aralığında çıkış düğümüne EK-AĞIRLIK(6,1) mesajı gönderir. Çıkış düğümü EK-AĞIRLIK mesajını aldığı anda TradeOff(3) değeri güncellenir.

7.RAUNT başladığından en küçük ödünleşme değerini, düğüm 5 vermektedir. Seçilen düğüm olarak düğüm 5 atanır. Düğüm 5 ve Ek-Komşusu aynı alt ağaçta olduğu için döngü kısıtını sağlamamaktadır. Çıkış düğümü tarafından düğüm 5'e DEVAM mesajı gönderilir.



Şekil-3: MCO algoritmasıyla RAUNT sonlarında oluşan ağaçlar a) MCO algoritması ilk RAUNT sonunda oluşan ağaç b) MCO algoritması ikinci RAUNT sonunda oluşan ağaç c) MCO algoritması üçüncü RAUNT sonunda oluşan ağaç d) MCO algoritması dördüncü RAUNT sonunda oluşan ağaç e) MCO algoritması beşinci RAUNT sonucunda oluşan ağaç f) MCO algoritması sonucunda oluşan kapasite kısıtlı en küçük kapsayan ağaç

DEVAM mesajını alan düğüm 5, sahip olduğu ZP aralığında çıkış düğümüne EK-AĞIRLIK(5,6) mesajı gönderir. Çıkış düğüm EK-AĞIRLIK mesajını aldığı anda düğüm 5'in ödünleşme değerini (TradeOff(5)) -3 olarak günceller. 8.RAUNT başladığında en küçük ödünleşme değerini tekrardan düğüm 5 vermektedir. Seçilen düğüm olarak düğüm 5 atanır. Düğüm 5'in Ek-Komşuna (Düğüm 6) bağlandığında oluşacak olan alt ağacın toplam kapasitesi 6 olacaktır. Kapasite kısıtı sağlanmamaktadır. Çıkış düğümü tarafından düğüm 5'e DEVAM mesajı gönderilir. DEVAM mesajını alan düğüm 5, sahip olduğu ZP aralığında çıkış düğümüne EK-AĞIRLIK(8,0) mesajı gönderir. Çıkış düğümü EK-AĞIRLIK mesajını aldığı anda TradeOff(5) değeri 0 olarak güncellenir.

9.RAUNT'ta düğüm 3, en küçük ödünleşme değerini veren düğümdür. SeçilmişDüğüm olarak düğüm 3 atanır. Düğüm 3'ün Ek-Komşuna (Düğüm 1) bağlandığında oluşacak olan alt ağacın toplam kapasitesi 6'dır. Kapasite kısıtı sağlanmamaktadır. Çıkış düğümü tarafından düğüm 3'e DEVAM mesajı gönderilir. DEVAM mesajını alan düğüm 3, sahip olduğu ZP aralığında çıkış düğümüne EK-AĞIRLIK(7,6) mesajı gönderir.

10.RAUNT yani sonuncu RAUNT başladığında en küçük ödünleşme değeri 0 olduğu için çıkış düğümü tarafından Çocuklar listesinde olmayan düğümlere GÜNCELLE (1,4), GÜNCELLE (2,1), GÜNCELLE (3,5) ve GÜNCELLE (6,4) mesajları gönderilir. GÜNCELLE mesajını alan düğüm 1, ebeveyn bilgisini 4 olarak günceller ve algoritmadan çıkar. GÜNCELLE mesajını alan diğer düğümlerde ebeveyn bilgilerini güncelleyip algoritmadan çıkarlar. Çıkış düğümü, komşularına BİTTİ mesajı gönderir ve algoritmadan çıkar. BİTTİ mesajını alan düğüm 4 ve düğüm 5 algoritmadan çıkar. Şekil 3'de MCO algoritması sonucunda oluşan kapasite kısıtlı en küçük kapsayan ağaç verilmiştir.

4. Performans Değerlendirmesi

Bu bölümde CENTEW ve MCO algoritmalarının performans sonuçları verilmiştir. MCO algoritması TinyOS işletim sistemi tabanlı TOSSIM [26], [27] simülasyon ortamında uygulanmıştır. Fiziksel katmanda IEEE 202.15.4 standardı kullanılmıştır. Ortam erişim kontrol katmanında ise kenar çizelgeleme tabanlı zaman bölümlü çoklu erişim (İng. time division multiple Access - TDMA) protokolü kullanılmıştır. Çalışma kapsamında mesaj kayıpları oluşmaması için TDMA protokolü tercih edilmiştir. Her bir düğüm kendisine özel ayrılan zaman aralığında mesaj gönderip, yine kendisi için özel ayrılan zaman aralığında mesaj almaktadır. Bu senaryo kapsamında mesaj kayıpları yaşanmamaktadır.

MCO algoritması performansının, CENTEW algoritmasıyla karşılaştırılması için CENTEW algoritması da TOSSIM simülasyon ortamında uygulanmıştır. Algoritmalar, enerji etkinliği açısından karşılaştırılmıştır. 5, 10, 15, 20, 30, 40 ve 50 kapasite kısıtları için 10 farklı 50, 100, 150, 200 ve 250 düğümlük ağlar oluşturularak toplamda 350 ağ üzerinde çalışılmıştır. Maksimum kapasite kısıtı en fazla 50 olarak belirlenmiştir. KEKA problemi C

kapasite kısıtına bağlıdır ve kapasite kısıtı ağda bulunan düğüm sayısına eşit veya büyük olduğunda KEKA problemi EKA problem ile aynı yöntemlerle çözülmektedir. Rastgele oluşturulan en küçük ağ, 50 düğümlük ağ olduğundan kapasite kısıtı en fazla 50 olarak seçilmiştir. Rastgele oluşturulan her bir ağ aynı kapasite kısıtları ile değerlendirilmiştir. Kapasite kısıtları belirli oranlarla artırılarak kapasitenin ağ üzerindeki yük dengesi ve harcadığı enerji gözlemlenmiştir. Her düğüm, diğer düğümler ile direkt bağlıdır ve bağlantıların ağırlıkları rastgele üretilmiştir. Çizelge-4'de simülasyon ortamının parametreleri verilmiştir. Düğümün harcadığı enerji (E) gönderilen ve alınan mesaj boyutuna bağlıdır. Gönderilen ve alınan mesajlar Bayt, harcanan enerji J , harcanan toplam zaman Milisaniye (ms) cinsinden değerlendirilmiştir. v düğümü için enerji tüketimi aşağıdaki formülle hesaplanmaktadır [28]:

$$E(v) \approx \left(\begin{array}{l} 8 * T_{dinleme}(v) \\ + 0.008 * T_{uyku}(v) \\ + 17 * T_{gönderilen}(v) \\ + 16 * T_{alınan}(v) \end{array} \right) * 3300 \text{ mJ} \quad (1)$$

$T_{dinleme}$ dinleme için harcanan süre, T_{uyku} uyku için harcanan süre, $T_{gönderilen}$ mesaj gönderme için harcanan süre, $T_{alınan}$ mesaj alma için harcanan süre, S gönderilen mesaj boyutu, R alınan mesaj boyutu olarak kabul edilmiştir. Her bir IRIS duyargası 3.3 Volt (V) batarya ile çalışmakta ve uyku modunda 0.008 Mili amper (mA), dinleme modunda 8 mA, mesaj alırken 16 mA ve mesaj gönderirken 17 mA enerji tüketmektedir. Her bir düğüm tarafından tüketilen toplam enerji; düğümün dinleme, uyuma, mesaj gönderme ve mesaj alma işlemlerinde harcadığı enerjiye eşittir. TDMA protokolünde her bir düğüm kendisine ait zaman aralığında aktif hale gelip komşularını dinlemektedir. Çalışmada kullanılan MCO algoritması simülasyon ortamında her bir zaman aralığının uzunluğu 25, CENTEW algoritmasının zaman aralığının uzunluğu 100 ms'dir. v düğümü için dinleme işleminde harcanan süre aşağıdaki formülle hesaplanmaktadır. $Renk(\Gamma_v)$ fonksiyonu v düğümünün komşularına atanan toplam renk sayısını vermektedir. T toplam geçen zaman ve F TDMA protokolündeki çerçeve uzunluğudur. F değeri, ağdaki toplam düğümlere atanan renk sayısı ile her bir zaman aralığının uzunluğunun çarpımıdır. T/F simülasyonda kullanılan toplam çerçeve (İng. frame) sayısını vermektedir.

$$T_{dinleme}(v) \approx (100 * Renk(\Gamma_v)) * (T/F) \quad (2)$$

Her bir düğüm kendi zaman aralığında dinleme işlemini tamamladıktan sonra uyku durumuna geçmektedir. v düğümü için uyku işleminde harcanan süre aşağıdaki formülle hesaplanmaktadır.

$$T_{uyku}(v) \approx (F - (100 * (1 + Renk(\Gamma_v))))(T/F) \quad (3)$$

Veri iletim oranı her bir düğüm için 250 kbps (İng. Kilobit per Second) yani 31.25 kBps (İng. KiloBayt per Second)dir. Mesaj göndermek ve almak için harcanan süreler aşağıdaki formülle hesaplanmaktadır.

$$T_{sönderilen}(v) = (S/31.25) \quad (4)$$

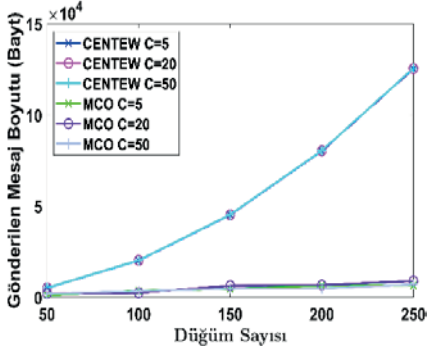
$$T_{alınan}(v) = (R/31.25) \quad (5)$$

Çizelge-4: Simülasyon ortam parametreleri

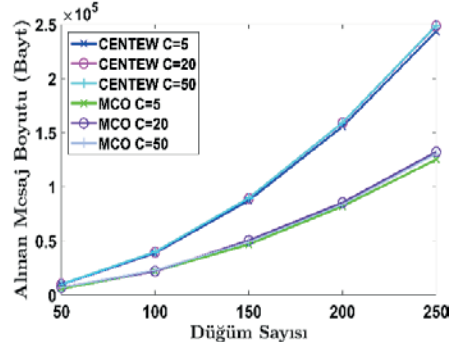
Simülasyon	TinyOS-TOSSIM
Duyarga Tipi	IRIS
Topoloji Sayısı	350
Düğüm Sayısı	50, 100, 150, 200 ve 250
MAC Protokolü	İleri TDMA
Kapasite Kısıtları	5, 10, 15, 20, 30, 40 ve 50
Düğüm Dağılımı	Rastgele dağılım
Ağ Modeli	Tam bağlı topoloji

Çizelge-5: CENTEW ve MCO algoritmalarının karşılaştırılması (Düğüm Sayısı:50)

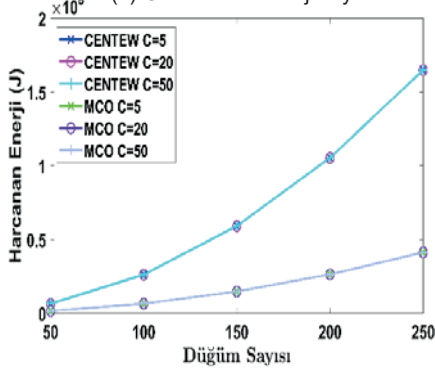
CENTEW				
Kapasite Kısıtı	Toplam Gönderilen Mesaj (Bayt)	Toplam Alınan Mesaj (Bayt)	Harcanan Enerji (J)	Geçen Zaman (ms)
5	5057	9590	6493413	8919
10	5077	9745	6493711	9423
15	5081	9764	6493750	9511
20	5085	9779	6493783	9622
30	5089	9790	6493808	9715
40	5089	9790	6493808	9727
50	5093	9797	6493827	9714
MCO				
Kapasite Kısıtı	Toplam Gönderilen Mesaj (Bayt)	Toplam Alınan Mesaj (Bayt)	Harcanan Enerji (J)	Geçen Zaman (ms)
5	1186	5719	1628824	282257
10	1276	5944	1629366	307382
15	1889	6572	1631528	524910
20	2173	6867	1632536	624932
30	2324	7025	1633074	677457
40	2142	6843	1632440	612464
50	1901	6605	1631605	524987



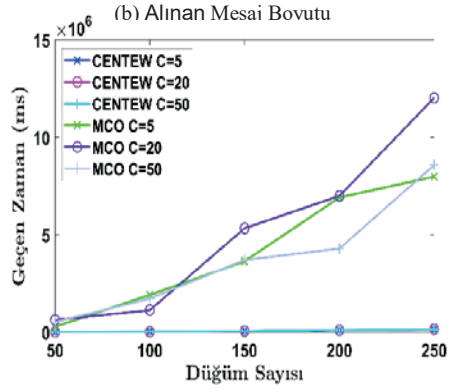
(a) Gönderilen Mesaj Boyutu



(b) Alınan Mesaj Boyutu



(c) Harcanan Enerji



(d) Geçen Zaman

Şekil-4: CENTEW ve MCO algoritmalarının kapasite kısıtı 5, 20 ve 50 bazında karşılaştırılması

Çizelge-6: CENTEW ve MCO algoritmalarının karşılaştırılması (Düğüm Sayısı:100)

CENTEW				
Kapasite Kısıtı	Toplam Gönderilen Mesaj (Bayt)	Toplam Alınan Mesaj (Bayt)	Harcanan Enerji (J)	Geçen Zaman (ms)
5	20109	38610	26237599	27723
10	20153	39347	26238923	28818
15	20169	39495	26239202	29220
20	20173	39522	26239254	29317
30	20181	39564	26239340	29524
40	20185	39579	26239372	29618
50	20185	39579	26239372	29616
MCO				
Kapasite Kısıtı	Toplam Gönderilen Mesaj (Bayt)	Toplam Alınan Mesaj (Bayt)	Harcanan Enerji (J)	Geçen Zaman (ms)
5	3466	21967	6577403	1909464
10	2586	21780	6575507	1249734
15	3652	22978	6579445	1999834

20	2438	21787	6575254	1129858
30	3909	23292	6580437	2174912
40	3276	22670	6578250	1719939
50	3276	22670	6578250	1719932

Şekil-4'te düğümlerin toplam alınan mesaj boyut değişimi a alt çizgesinde, düğümlerin toplam gönderilen mesaj boyut değişimi b alt çizgesinde, toplam enerji tüketimi (J) c alt çizgesinde ve bütün düğümler tarafından harcanan toplam zaman (ms) d alt çizgesinde gösterilmiştir.

Değişim Katsayısı, Toplam Gönderilen Mesaj, Toplam Alınan Mesaj ve Harcanan Enerji bölümlerinde üzerinde çalışılan metrikler hakkında değerlendirmeler yapılmıştır.

4.1. Değişim Katsayısı

Alt ağaçların birbirleri arasındaki denge, değişim katsayısı (İng. Coefficient of Variation - CV)

metoduyla ölçülmektedir. Değişim katsayısı metodu, standart sapma (İng. standard deviation - SD) ve ortalamanın (İng. Mean) birbirine oranıdır.

Çizelge-7’de MCO algoritmasının düğüm sayısı ve kapasite kısıtına göre değişim katsayısı sonuçları gösterilmektedir. Düğüm sayısının, kapasite kısıtına oranı az olduğunda alt ağaçlar daha dengeli olmaktadır. Çizelge-7, kapasite kısıt 5 ve 30 olduğunda daha dengeli ağaçların üretildiğini göstermektedir. Kapasite kısıtı, ağdaki toplam düğüm sayısına eşit ve daha fazla olduğunda sadece 1 alt ağaç üretilmektedir.

4.2. Toplam Gönderilen Mesaj

Çizelge-5, 6, 8, 9, 10, Şekil 4a, Şekil 5a ve Şekil 6a TDA’yı oluşturan düğüm sayısı arttığında toplam gönderilen mesaj boyutundaki değişim gösterilmiştir. Çizelgeler ve şekiller, düğüm sayısına göre her iki algoritmanın gönderdiği mesaj boyutunun arttığını göstermektedir.

Çizelge-7: MCO algoritmasının düğüm sayısına bağlı değişim katsayısı sonuçları

Düğüm Sayısı		50	100	150	200	250
C:5	MEAN	4.9	4.71	4.9	4.98	4.93
	SD	0.3	0.95	0.52	0.16	0.41
	CV(%)	6.12	20.17	10.61	3.21	8.3
C:10	MEAN	8.52	8.84	9.8	9.71	9.73
	SD	3.07	1.92	1.49	1.32	1.18
	CV(%)	36.03	21.72	15.20	13.59	12.13
C:15	MEAN	11.12	14.14	14.32	13.24	13.91
	SD	2.68	0.14	3.52	4.43	3.07
	CV(%)	24.1	0.99	24.58	33.46	22.07
C:20	MEAN	16.3	16.66	18.63	18.09	17.79
	SD	1.41	7.31	1.58	4.01	3.78
	CV(%)	8.65	43.87	8.48	22.16	21.25
C:30	MEAN	24.5	24.75	24.83	28.43	24.9
	SD	4.58	2.6	3.72	1.36	5.34
	CV(%)	18.69	10.51	14.98	4.78	21.45
C:40	MEAN	24.5	33	37.25	33.16	31.13
	SD	11.6	2.83	3.64	7.54	7.6
	CV(%)	47.35	8.56	9.77	22.74	24.41
C:50	MEAN	49	33	37.25	49.75	35.57
	SD	0	2.83	15.84	0.25	16.41
	CV(%)	0	8.56	42.52	0.5	46.13

CENTEW algoritmasında kapasite seviyesi, gönderilen mesaj boyutunu etkilememektedir. Bunun nedeni ilk rauntta her düğümün, komşuluk bilgilerini çıkış düğümüne göndermesidir. MCO algoritması, CENTEW algoritmasıyla karşılaştırıldığında MCO algoritması lineer davranış göstermektedir. Sonuçlara göre MCO algoritması, CENTEW algoritmasına göre çok daha az mesaj boyutuna ihtiyaç duymaktadır. MCO algoritmasında 250 düğümlük ağlarda gönderilen mesaj boyutu, CENTEW algoritmasına göre yaklaşık 15,56 kat daha azdır.

Çizelge-8: CENTEW ve MCO algoritmalarının karşılaştırılması (Düğüm Sayısı:150)

CENTEW				
Kapasite Kısıtı	Toplam Gönderilen Mesaj (Bayt)	Toplam Alınan Mesaj (Bayt)	Harcanan Enerji (J)	Geçen Zaman (ms)
5	45165	87189	59232790	56617
10	45229	88805	59235636	58217
15	45249	89100	59236170	58725
20	45261	89229	59236409	59019
30	45273	89322	59236588	59322
40	45281	89364	59236673	59513
50	45281	89364	59236673	59522
MCO				
Kapasite Kısıtı	Toplam Gönderilen Mesaj (Bayt)	Toplam Alınan Mesaj (Bayt)	Harcanan Enerji (J)	Geçen Zaman (ms)
5	4598	46622	14838125	3636689
10	7042	50618	14849264	6187085
15	5046	48897	14842773	4027209
20	6276	50244	14847257	5332288
30	4825	48874	14842338	3764863
40	5001	49084	14843009	3944908
50	4777	48860	14842228	3704913

MCO algoritmasının gönderdiği mesaj boyutu, CENTEW algoritmasına göre çizelge-5’de yaklaşık 2.93 kat, çizelge-6’da yaklaşık 6.41 kat, çizelge-8’de yaklaşık 10.31 kat ve çizelge-9’da yaklaşık 12.58 kat daha az olduğu gözlemlenmiştir. Kapasite kısıtının, gönderilen mesaj boyutunda etkisinin az olduğu görülmüştür. Ortalamada MCO algoritmasının gönderdiği mesaj boyutunun, CENTEW algoritmasına göre yaklaşık 9.22 kat daha az olduğu görülmüştür.

4.3. Toplam Alınan Mesaj

Çizelge-5, 6, 8, 9, 10, Şekil-4b, Şekil 5b ve Şekil 6b TDA’yı oluşturan düğüm sayısı arttığında toplam alınan mesaj boyutundaki değişim gösterilmiştir.

CENTEW algoritmasında kapasite seviyesi alınan mesaj boyutunu etkilememektedir. Bunun nedeni ilk rauntta her düğümün komşuluk bilgilerini çıkış düğüme göndermesi ve toplam alınan mesaj boyutundaki değişimin GÜNCELLE ve BİTTİ mesaj sayısına bağlı olmasıdır. Çizelgelere göre her iki algoritmanın toplam aldığı mesaj boyutu, kapasite kısıtına bağlı olarak artmakta ve MCO algoritması, CENTEW algoritmasına göre daha az mesaj boyutu kullanmaktadır. MCO algoritmasında 50 düğümlük ağlarda toplam alınan mesaj boyutu CENTEW algoritmasına göre yaklaşık 1,5 kat, 250 düğümlük ağlarda ise yaklaşık 1,9 kat daha azdır.

Çizelge-9: CENTEW ve MCO algoritmalarının karşılaştırılması (Düğüm Sayısı:200)

CENTEW				
Kapasite Kısıtı	Toplam Gönderilen Mesaj (Bayt)	Toplam Alınan Mesaj (Bayt)	Harcanan Enerji (J)	Geçen Zaman (ms)
5	80229	155630	105479507	95718
10	80301	158024	105483681	97515
15	80337	158735	105484947	98424
20	80353	158947	105485334	98821
30	80369	159095	105485613	99217
40	80373	159122	105485666	99319
50	80377	159145	105485712	99415
MCO				
Kapasite Kısıtı	Toplam Gönderilen Mesaj (Bayt)	Toplam Alınan Mesaj (Bayt)	Harcanan Enerji (J)	Geçen Zaman (ms)
5	6444	81845	26417985	6908962
10	5228	82951	26417671	5069405
15	8988	87386	26431914	10389637
20	6624	85218	26424007	6989735
30	8425	87151	26430506	9539836
40	6301	85050	26423144	6499864
50	4754	83503	26417753	4289860

Kapasite kısıtının toplam alınan mesaj boyutunu etkilemediği ve Çizelge-5, 6, 8, 9 ve 10 incelendiğinde toplam alınan mesaj boyut değerlerinin, birbirine çok yakın olduğu görülmüştür. CENTEW algoritmasının aldığı mesaj boyutu, kapasite kısıtına bağlı olarak MCO algoritmasına göre yaklaşık 1,77 kat daha fazladır.

4.4. Harcanan Enerji

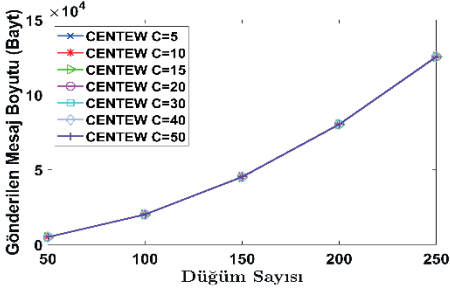
Çizelge-5, 6, 8, 9, 10, Şekil 4c, Şekil 5c ve Şekil 6c'de TDA'yı oluşturan düğümler tarafından harcanan toplam enerjinin sonuçları verilmiştir. Harcanan enerji, alınan ve gönderilen mesaj boyutuna bağlıdır. TDA'yı oluşturan düğüm sayısının artmasıyla her iki algoritmanın harcadığı

enerjinin arttığı görülmüştür. Sonuçlara göre MCO algoritması tarafından harcanan toplam enerji, CENTEW algoritması tarafından harcanan toplam enerjiye göre oldukça azdır. MCO algoritması, kapasite kısıtına bağlı olarak uygun bir çözüm bulunduğu, kapasite kısıtı artsa bile algoritma aynı çözümü üretebilmektedir. 100 düğümlük ağlarda kapasite kısıtı 40 ve 50 iken MCO algoritması aynı çözümü (KEKA) üretmiştir. 250 düğümlük ağlarda CENTEW algoritmasının harcadığı enerji, MCO algoritmasının harcadığı enerjiye göre yaklaşık 4 kat daha fazladır.

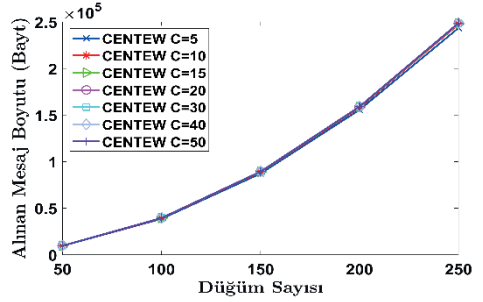
Çizelge-10: CENTEW ve MCO algoritmalarının karşılaştırılması (Düğüm Sayısı:250)

CENTEW				
Kapasite Kısıtı	Toplam Gönderilen Mesaj (Bayt)	Toplam Alınan Mesaj (Bayt)	Harcanan Enerji (J)	Geçen Zaman (ms)
5	125289	243540	164977064	144716
10	125389	247515	164983959	147218
15	125425	248334	164985408	148116
20	125441	248594	164985876	148519
30	125457	248790	164986236	148918
40	125461	248829	164986309	149026
50	125469	248895	164986434	149219
MCO				
Kapasite Kısıtı	Toplam Gönderilen Mesaj (Bayt)	Toplam Alınan Mesaj (Bayt)	Harcanan Enerji (J)	Geçen Zaman (ms)
5	6499	124953	41307952	7986237
10	9500	131626	41324614	13174331
15	9018	131927	41324257	12249562
20	8908	132061	41324286	12024656
30	9575	132908	41326915	13187261
40	7185	130553	41318645	8912283
50	7004	130430	41318113	8574834

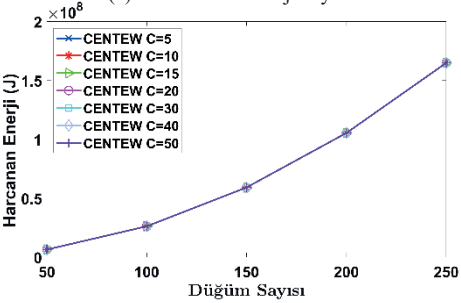
MCO algoritmasında 50 ve 100 düğümlük ağlarda harcadığı enerji, CENTEW algoritmasına göre yaklaşık 3,98 kat, 150 ve 200 düğümlük ağlarda ise yaklaşık 3,99 kat daha azdır. CENTEW algoritmasında paket boyutlarının büyük olması düğümlerin dinleme, uyuma, paket gönderme ve paket alma sürelerini doğrudan etkileyerek düğümlerin enerji tüketimini arttırmıştır. Çizelge-5, 6, 8, 9 ve 10 incelendiğinde kapasite kısıtının, harcanan enerjiye etkisi olmadığı görülmüştür. CENTEW algoritması, MCO algoritmasına göre kapasite kısıtına bağlı olarak yaklaşık 3,98 kat daha fazla enerji harcamaktadır.



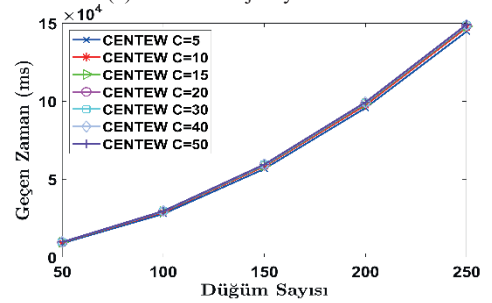
(a) Gönderilen Mesaj Boyutu



(b) Alınan Mesaj Boyutu

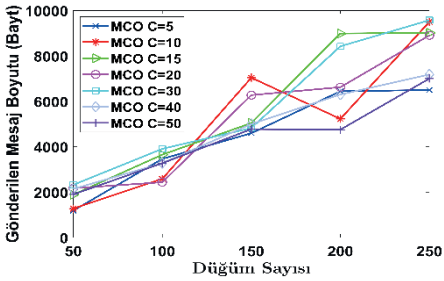


(c) Harcanan Enerji

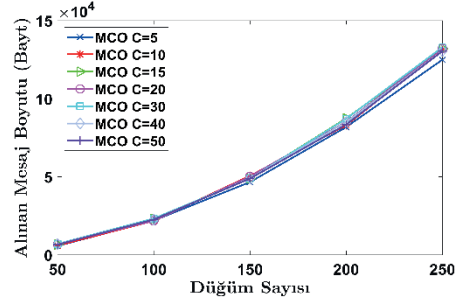


(d) Geçen Zaman

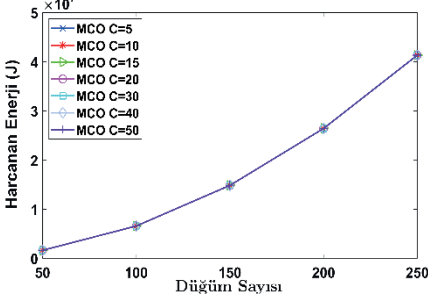
Şekil-5: CENTEW algoritmasının kapasite kısıtı 5, 10, 15, 20, 30, 40 ve 50 bazında sonuçları



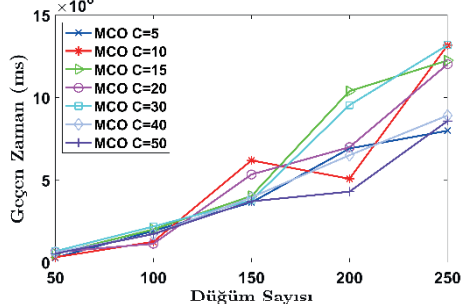
(a) Gönderilen Mesaj Boyutu



(b) Alınan Mesaj Boyutu



(c) Harcanan Enerji



(d) Geçen Zaman

Şekil-6: MCO algoritmasının kapasite kısıtı 5, 10, 15, 20, 30, 40 ve 50 bazında sonuçları

4.5. Geçen Zaman

Çizelge-5, 6, 8, 9, 10, Şekil 4d, Şekil 5d ve Şekil 6d'de algoritmaların TDA üzerinden sonlandırılması için geçen zamanın sonuçları verilmiştir. Kapasite kısıtının artmasıyla MCO algoritması tarafından harcanan zaman azalmaktadır. 150 ve 250 düğümlük ağlarda CENTEW algoritmasında geçen zaman, MCO algoritmasında geçen zamana göre yaklaşık 73.45 kat daha azdır. MCO algoritması, CENTEW algoritmasına göre çok daha fazla zaman harcamaktadır. Çünkü CENTEW algoritması, her düğümün bilgisini ilk rauntta toplamakta ve bu bilgiyi toplamak için oldukça az zamana ihtiyaç duymaktadır. CENTEW algoritması, E-W algoritmasını kullanarak KEKA problemini çözüp GÜNCELLE ve BİTTİ mesajlarıyla ağı bilgilendirir. MCO algoritması, çok sayıda mesaj göndererek veya alarak KEKA problemini çözer. Çizelge-5, 6, 8, 9 ve 10 incelendiğinde kapasite kısıtının, algoritmalar tarafından harcanan zamana etkisi olduğu görülmüştür. Düğüm sayısının artmasıyla, her iki algoritmanın harcadığı zamanında artış göstermesine rağmen kapasite kısıtının artmasıyla algoritmaların harcadığı zaman azalmıştır. Ortalamada CENTEW algoritmasında geçen zaman, MCO algoritmasında geçen zamana göre yaklaşık 66,32 kat daha azdır. Performans sonuçlarına göre MCO algoritması, CENTEW algoritmasına göre daha fazla zaman harcamaktadır.

5. Sonuç

TDA'yı oluşturan düğümlerin enerji kaynakları sınırlı olduğundan enerji etkin algoritmalarının tasarımı ve uygulanması oldukça önemlidir. TDA üzerinde KEKA bulunması, hem çıkış düğümüne bağlı düşük maliyetli bir ağaç hem de kapasite kısıtı sayesinde dengeli alt ağaçlar sağlamaktadır. KEKA tabanlı topoloji kontrol tekniği, enerji-etkin ve dengelidir. Bu özellikler ağı, yaşam ömrünü uzatmak için çok önemlidir.

Bu çalışmada TDA üzerinde KEKA algoritmalarının performansı değerlendirilmiştir. MCO ve CENTEW algoritmaları Esau-Williams algoritmasından türetilmiştir. Her iki algoritma da TinyOS tabanlı duyurğa ağ uygulamaları için kullanılan TOSSIM simülasyon ortamında uygulanmıştır. 5, 10, 15, 20, 30, 40 ve 50 kapasite kısıtları için 10 farklı 50, 100, 150, 200 ve 250 düğümlük ağlar oluşturulmuştur. Düğüm sayısının

artmasıyla algoritmaların gönderdikleri toplam mesaj boyutu, aldıkları toplam mesaj boyutu, harcadıkları enerji ve çalışma zamanları artmaktadır.

MCO algoritması; toplam gönderilen mesaj boyutu, toplam alınan mesaj boyutu ve harcanan enerji performansı açısından CENTEW algoritmasına göre daha iyi performans sağlamaktadır. MCO algoritması, CENTEW algoritmasına göre yaklaşık olarak 3,98 kat daha az enerji harcamaktadır. Performans değerlendirmesi sonucunda dağıtık KEKA yaklaşımının enerji-etkin, merkezi KEKA yaklaşımının zaman-etkin olduğu görülmüştür.

Gelecekteki çalışmalarda mesaj kayıplarının yaşandığı senaryolar, algoritma içerisine eklenerek TDMA tabanlı MAC, yarışma temelli MAC ve melez MAC ortamlarında MCO algoritması analiz edilecektir. Bağlantı kayıpları, düğüm hareketliliği ve güvenlik gibi gerçek parametreler dikkate alınarak kapasite kısıtları değerlendirilecektir.

Teşekkür

Bu çalışma, 215E115 nolu proje kapsamında TÜBİTAK tarafından desteklemiştir.

Kaynakça

- [1] Erciyes, K., Dagdeviren, O., Coskulu, D., *Modeling and simulation of wireless sensor and mobile ad hoc networks*, International conference on modelling and simulation, 2006.
- [2] Papadimitriou, C. H., *The complexity of the capacitated tree problem*, Networks, 8(3), pp. 217-230, 1978.
- [3] Ruiz y Ruiz, H. E., *The capacitated minimum spanning tree problem*, PhD Thesis, Universitat Politecnica de Catalunya, 2013.
- [4] Aşçı, M., İleri, C. U., Dağdeviren, O., *An energy-efficient capacitated minimum spanning tree algorithm for topology control in Wireless Sensor Networks*, 2017 25th Signal Processing and Communications Applications Conference (SIU), pp. 1-4, 2017.
- [5] Deif, D., Gadallah, Y., *Reliable wireless sensor networks topology control for critical internet of things applications*, 2018 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1-6, 2018.

- [6] Dubey, T. K., Mathur, R., Chouhan, D. N., *Localization independent aspects of topology control in wireless sensor networks*, Optical and Wireless Tech., Springer Singapore, pp. 1-6, 2018.
- [7] Santi, P., *Topology control in wireless ad hoc and sensor networks*, ACM Comput. Surv., 37(2), pp. 164-194, 2005.
- [8] Li, M., Li, Z., Vasilakos, A. V., *A survey on topology control in wireless sensor networks: Taxonomy, comparative, study, and open issues*, Proc of the IEEE, 101(12), pp. 2538-2557, 2013.
- [9] Yun, D., Qingjun, Z., Xiaohui, C., *Research on topology algorithm in heterogeneous wireless sensor networks based on the game theory*, Proceedings of the 3rd International Conference on Intelligent Information Processing, ACM, pp. 112-119, 2018.
- [10] Nayak, M. R., Tripathy, G., Rath, A. K., *A distributed transmission power efficient fault-tolerant topology management mechanism for nonhomogeneous wireless sensor network*, Progress in Advanced Comp. and Intelligent Eng., Springer Singapore, pp. 481-493, 2018.
- [11] Chandy, K. M., Lo, T., *The capacitated minimum spanning tree*, Networks, 3(2), pp. 173-181, 1973.
- [12] Esau, L. R., Williams K. C., *On teleprocessing system design, part ii: A method for approximating the optimal network*, IBM Systems Journal, 5(3), pp. 142-147, 1966.
- [13] Lee, Y. J., Atiquzzaman, M., *Least cost heuristic for the delay-constrained capacitated minimum spanning tree problem*, Computer Communications, 28(11), pp. 1371-1379, 2005.
- [14] Öncan, T., Altinel, İ. K., *Parametric enhancements of the esau-williams heuristic for the capacitated minimum spanning tree problem*, Journal of the Operational Research Society, 60(2), pp. 259-267, 2009.
- [15] Battarra, M., Öncan, T., Altinel, I., Golden, B., Vigo, D., Phillips, E., *An evolutionary approach for tuning parametric esau and williams heuristics*, Journal of the Operational Research Society, 63(3), pp. 368-378, 2012.
- [16] Campos, J., Martins, A., Souza, M., *A hybrid vns algorithm for solving the multi-level capacitated minimum spanning tree problem*, Electronic Notes in Discrete Mathematics, 66, pp. 159-166, 2018.
- [17] Kawatra, R., Bricker, D., *A multiperiod planning model for the capacitated minimal spanning tree problem*, European Journal of Operational Research, 121(2), pp. 412-419, 2000.
- [18] Öncan, T., *Design of capacitated minimum spanning tree with uncertain cost and demand parameters*, Information Sciences, 177(20), pp. 4354-4367, 2007.
- [19] de Souza, M. C., Duhamel, C., Ribeiro, C. C., *A grasp heuristic for the capacitated minimum spanning tree problem using a memory-based local search strategy*, Metaheuristics: Computer decision-making, Springer, pp. 627-657, 2003.
- [20] Ahuja, R. K., Orlin, J. B., Sharma, D., *Multi-exchange neighborhood structures for the capacitated minimum spanning tree problem*, Math. Programming, 91(1), pp. 71-97, 2001.
- [21] Ahuja, R. K., Orlin, J. B., Sharma, D., *New neighborhood search structures for the capacitated minimum spanning tree problem*, Technical Report by Massachusetts Inst. of Tech., Sloan School of Management, 1998.
- [22] Reimann, M., Laumanns, M., *A hybrid aco algorithm for the capacitated minimum spanning tree problem*, Hybrid Metaheuristics, pp. 1-10, 2004.
- [23] Martins, P., *Enhanced second order algorithm applied to the capacitated minimum spanning tree problem*, Computers & operations research, 34(8), pp. 2495-2519, 2007.
- [24] Gamvros, I., Raghavan, S., Golden, B., *An evolutionary approach to the multi-level capacitated minimum spanning tree problem*, Telecommunications Network Design and Management, Springer, pp. 99-124, 2003.
- [25] Ruiz, E., Albareda-Sambola, M., Fernandez, E., Resende, M. G., *A biased random-key genetic algorithm for the capacitated minimum spanning tree problem*, Computers & Operations Research, 57, pp. 95-108, 2015.
- [26] Levis, P., Lee, N., Welsh, M., Culler, D., *Tossim: Accurate and scalable simulation of entire tinyos applications*, Proceedings of the

1st Inter. Conference on Embedded Networked Sensor Systems, ACM, pp. 126-137, 2003.

[27] Erciyes, K., Dagdeviren, O., Cokuslu, D., Yilmaz, O., Gumus, H., *Mobile ad hoc networks: Current status and Future trends*, CRC Press, 2010.

[28] Akram, V. K., Dagdeviren, O., *DECK: A distributed, asynchronous and exact k-connectivity detection algorithm for Wireless Sensor Networks*, Computer Communications, 116, pp. 9-20, 2018.