

Kablosuz Algılayıcı Ağlarında Saat Eşzamanlaması İçin Yükte Hafif Bir Yöntem

A Lightweight Method for Time Synchronization in Wireless Sensor Networks

Kasım Sinan Yıldırım, Aylin Kantarcı

Bilgisayar Mühendisliği Bölümü
Ege Üniversitesi
{sinan.yildirim, aylin.kantarcı}@ege.edu.tr

ÖZETÇE

Algılayıcı düğümleri arasında ortak bir zaman kavramı oluşturmaya çalışan dağıtık saat eşzamanlama protokolleri, kablosuz algılayıcı ağlarındaki çoğu uygulama için önemli bir zorunluluktur. Bu bildiri, ana hedefi basit ve yükte hafif yöntemlerle ağ genelinde mikrosaniye hassasiyetinde eşzamanlama sağlamak olan Basit Saat Eşzamanlama (BSE)¹ protokolünü tanıtmaktadır. BSE bir kök düğümü tarafından periyodik olarak yayınlanan eşzamanlama mesajlarının ağa hızlı bir şekilde yayılmasına dayanır ve saat sapmasını telafi edebilmek için basit ortalama alma yöntemini kullanır. BSE Iris platformunda TinyOS işletim sistemi kullanılarak gerçekleştirilmiş ve 16 algılayıcı düğümü kullanılarak değerlendirilmiştir. DeneySEL sonuçlar, BSE protokolünün algılayıcı ağlarında pratikte kullanılan FTSP protokolünden daha üstün olduğunu göstermiştir.

ABSTRACT

A distributed time synchronization protocol whose objective is to provide a common notion of time between the sensor nodes is mandatory for many applications in wireless sensor networks. This paper presents *Simple Time Synchronization* (STS) protocol whose main objective is to achieve network-wide synchronization with microsecond precision using a simple and lightweight method. STS achieves synchronization by fast flooding synchronization messages which are periodically broadcasted by a reference node and uses a simple averaging strategy for drift compensation. STS is implemented on Iris platform using TinyOS and evaluated on a testbed setup including 16 sensor nodes. Experimental results show that the prototype implementation of STS outperforms Flooding Time Synchronization Protocol which is the de facto protocol for time synchronization in wireless sensor networks.

1. GİRİŞ

Kablosuz algılayıcı ağları çevreyi algılama yeteneğine sahip, ucuz ve enerji kısıtları olan küçük cihazlardan oluşur. Birbirlerinden fiziksel olarak ayrılmış olan algılayıcı düğümleri,

çevresel verileri algılayıcılarını ve kablosuz iletişim donanımlarını kullanarak işbirliği içerisinde toplar, işler ve yayarlar. Kablosuz algılayıcı ağlarının temel uygulamaları erişilmesi zor alanlardan veri toplama, çevrenin gözlemlenmesi ve hedef takipidir.

Her algılayıcı düğümü, farklı tıklama hızına sahip saat donanımına sahiptir. Bu yüzden, algılayıcı düğümlerinin saatleri ilk başta eşzamanlanmış olsa bile, zaman geçtikçe birbirinden uzaklaşır. Birçok uygulama, algılayıcı düğümleri okudukları çevresel verileri zaman bilgisi ile etiketliyerek bunları birleştiren ve yorumlayan merkez istasyonuna gönderirler. Birleştirme ve yorumlama işleminin tutarlı ve anlamlı olabilmesi için, verilerin zaman etiketlerinin tutarlı olması gerekmektedir. Diğer bir örnek, enerji kısıtına sahip algılayıcı düğümlerinin enerji tasarrufu sağlamak için eşgüdümlü olarak haberleşme donanımlarını açıp kapamalarıdır. Bu ve benzeri eşgüdümlü eylemlerin yapılabilmesi algılayıcı düğümlerinin saatlerinin sıkı bir şekilde eşzamanlanmasını gerektirir. Sonuç olarak, algılayıcı düğümleri arasında ortak zaman kavramı oluşturmaya hedefleyen dağıtık saat eşzamanlama protokolleri, kablosuz algılayıcı ağlarındaki çoğu protokol ve uygulamanın doğru ve verimli bir şekilde çalışabilmesi için olmazsa olmazdır.

Kablosuz algılayıcı düğümlerinin enerji kısıtları ve ağın devingenliği, saat eşzamanlaması için yeni yöntemlere ihtiyaç duyulmasına yol açmıştır. Dağıtık sistemler için tasarlanmış geleneksel saat eşzamanlama protokolleri, haberleşmenin yol açtığı enerji tüketimini önemsemezler. Çoğu geleneksel protokol, önceden tasarlanmış ve ağın yaşamı boyunca değişmemesi gereken bir haberleşme altyapısına ihtiyaç duyarlar. Buna karşın, algılayıcı ağlarında çevresel etkenlerden ya da biten pillerden ötürü, düğümlerin çalışması sonlanabilir. Yeni düğümler ağa eklenirken ağın topolojisi ve düğümler arasındaki bağlantılar sürekli değişmektedir. Kablosuz haberleşme donanımı ile haberleşme birçok paket çarpışmasına ve kayıp paketlere yol açmaktadır.

Bazı uygulamalar, algılayıcı düğümlerinin saatlerinin ağın dışındaki bir kaynaktan gelen saat değerine eşzamanlanmasına gerek duymaktadırlar. Buna *dışsal eşzamanlama* denmektedir. *İçsel eşzamanlamada* uygulamalar ağ içerisindeki düğümlerin saatlerinin kendi içinde tutarlı olmasını beklerler. Ek olarak, uygulamalar düğümlerin saatlerinin ağın yaşamı boyunca eşzamanlanmış olmasına (*sürekli*) ya da eşzamanlanmanın bir

¹ Kasım Sinan YILDIRIM Türkiye Bilimsel ve Teknolojik Araştırma Kurumu'na (TÜBİTAK) bu çalışmayı yurt içi doktora burs programı (BAYG-2211) ile desteklediği için teşekkür eder.

olay ile tetiklenmesine (*talep üzerine*) gerek duyabilirler [1].

Bu bildiride, kablosuz algılayıcı ağlarında sürekli saat eşzamanlaması sağlayan *Basit Saat Eşzamanlama* (BSE) protokolünü tanıtmaktayız. BSE bir kök düğümü tarafından periyodik olarak yayınlanan eşzamanlama mesajlarının ağa hızlı bir şekilde yayılmasına dayanır ve saat sapmasını telafi edebilmek için doğrusal bağlanım yerine basit ortalama alma yöntemini kullanır. Hem içsel hem dışsal eşzamanlama için kullanılabilen BSE, Iris platformunda TinyOS² işletim sistemi kullanılarak gerçekleştirilmiş ve 16 algılayıcı düğümü kullanılarak değerlendirilmiştir. BSE, algılayıcı ağlarında pratikte kullanılan FTSP protokolü ile kıyaslanmıştır.

Bu bildirinin geri kalan kısmı şu şekilde düzenlenmiştir. 2. Bölüm algılayıcı düğümlerindeki saat eşzamanlama konusunda ilgili çalışmaları özetlemektedir. BSE protokolü Bölüm 3'te detaylı bir şekilde açıklanmakta ve Bölüm 4'te değerlendirilmektedir. Son olarak Bölüm 5'te yorumlar ve gelecek çalışmalara ilişkin öneriler ortaya koyulmuştur.

2. İLGİLİ ÇALIŞMALAR

Algılayıcı ağlarına yönelik saat eşzamanlama protokolleri, uygulamanın gereksinimlerine göre düğümlerin saatlerinin olabildiğince birbirlerine yaklaştırılmasına dayanır. Genel yaklaşımlar iletişim sıklığını, verilerdeki bit sayısını, kaynak ve işlemci kullanımını azaltmayı içermektedir. Literatürde, ağ içerisindeki herhangi iki düğümün saatleri arasındaki farkı (*genel saat farkı*) en aza indirmeyi hedefleyen birçok protokol çalışması mevcuttur [2, 3, 4, 5, 6, 7].

Lightweight Tree Based Synchronization Protocol (LTS) [2] dışsal saat kaynağına sahip bir düğümü kök düğüm olarak kabul eden bir yayılım ağacını iletişim altyapısı kullanarak eşzamanlamayı sağlar. Bu ağacın derinliği ağdaki saat eşzamanlama hatasına doğrudan etkide bulunur. *Timing-Sync Protocol for Sensor Networks* [3] (TPSN) bir lider düğüm seçme safhasından sonra, bu düğümü kök düğüm olarak kabul eden bir yayılım ağacı oluşturur. Düğümlerdeki iş yükü, paketlerin ağ katmanındaki kuyrukta bekleme süreleri, paketin bir düğümden diğerine gitme süresi, paket verisinin gönderici düğümden kodlanması ve alıcı düğümden bu kodun çözülmesi belirsiz gecikmelere yol açar ve bu gecikmeler saat eşzamanlama hatasını doğrudan etkilemektedir[5]. TPSN bu belirsizlikleri büyük ölçüde gideren, paketlerin zaman etiketlerini MAC katmanında verme yöntemini tanıtmıştır. LTS ve TPSN'nin iletişim altyapısı olarak kullandığı yayılım ağacının devamlılığının sağlanması periyodik olarak mesajlaşmaya dayanmaktadır ve bu durum yüksek enerji tüketimine yol açar. Ek olarak, bu protokollerin hesaplama yükü ve bellek gereksinimleri fazladır.

Reference Broadcast Synchronization (RBS) [4] protokolünde, bir referans düğümü, kapsama alanında bulunan tüm düğümler için eşzamanlama paketi yayınlar. Bu paketi hemen hemen aynı zamanda alan tüm komşu düğümler, paketi alma zamanlarını kendi saatlerini kullanarak kaydederler. Daha sonraki adımda tüm alıcılar, bu saat değerini diğer alıcılarla değiş tokuş ederler ve saatlerini diğer alıcıların saat değerlerini gözeterek eşzamanlarlar. RBS'nin temel kusuru, bir refer-

ans düğümü ve onun kapsama alanındaki n düğümden oluşan küçük bir ağda bile $O(n^2)$ mesaj değiş tokuşuna gereksinim duymasıdır. *Flooding Time-Synchronization Protocol* (FTSP) [5] en küçük düğüm numarasına sahip düğümü lider olarak seçer. Lider düğüm, periyodik olarak saatini yayınlar. Ağdaki her düğüm, lider düğümünden veya diğer düğümlerden gelen eşzamanlama mesajlarını kullanarak, doğrusal bağlanım yöntemini kullanarak lider düğümün saatini tahminler. Ayrıca, bu tahmini saat değerini lider düğümden olduğu gibi periyodik olarak yayınlar. FTSP, MAC katmanında etiketleme yöntemini kullanarak mesaj iletim hattındaki belirsizlikleri büyük ölçüde giderir. FTSP'nin temel problemi, düğümlerin lider düğüme ilişkin bir saat tahmini aldığı anda bu tahmini anında diğer düğümlere iletmemeleridir. Her düğüm, kendi yayın zamanları gelince tahminleri yayınlamakta ve bu da FTSP'deki hataların ağırlık çapının üstsel fonksiyonu olmasına yol açmaktadır. *Rapid Time Synchronization* (RATS) [6] protokolü, FTSP gibi saat değerinin ağa kök düğümü tarafından periyodik olarak yayılmasına dayanır. Bu protokolde, her düğüm aldığı eşzamanlama mesajlarında yer alan ve gönderen düğümün kendi yerel zamanı cinsinden ifade edilmiş zaman etiketini kendi yerel saatleri cinsinden ifade edilen yerel zaman etiketine dönüştürürler. Düğümler sonraki adımda, aldığı mesajı yeniden yayınlarlar. Bu sayede her düğüm, eşzamanlama mesajını almış olur. Düğümler, doğrusal bağlanım yöntemini kullanarak saat sapmalarını hesaplarlar. Ancak RATS, kök düğümünün bozulması durumunda ağ genelinde eşzamanlama sağlayamamaktadır. PulseSync [7] kök düğümünden gelen saat değerlerinin ağa hızlı bir şekilde yayılmasına dayanır. Düğümler yine doğrusal bağlanım yöntemini kullanarak kök düğümünün saat değerini hesaplarlar.

BSE topolojik bir yapıya ihtiyaç duymadığı için, LTS ve TPSN protokolleri ile kıyaslandığında enerji tüketimi açısından daha verimlidir. PulseSync, RATS ve FTSP saat sapması için hesaplama yönünden yük getiren doğrusal bağlanım yöntemini kullanmaktadır. BSE ise doğrusal bağlanım yerine hesaplama açısından daha hafif olan ve az sayıda ondalık sayılarla işlem içeren ortalama alma yöntemini kullanmaktadır. RATS ile kıyaslandığı zaman, BSE kök düğümü hatasında eni bir kök düğümü seçip eşzamanlama işlemine devam etmektedir.

3. BSE PROTOKOLÜ

Bu bölümde *Basit Saat Eşzamanlama* (BSE) protokolü ayrıntılı olarak incelenecektir. Çok genel olarak, BSE protokolünde düğümler kök düğümünün ağa periyodik olarak yaydığı eşzamanlama mesajlarını kullanarak saatlerini eşzamanlarlar. Ağ genelinde eşzamanlama sağlayan bu yöntemin ayrıntıları izleyen alt bölümlerde sunulmuştur.

3.1. Ağ Genelinde Saat Eşzamanlaması

BSE'nin çalışma mekanizmasını açıklamadan önce, olaylara zaman etiketlerinin nasıl verildiğini ve eşzamanlama mesajlarını zaman etiketleri ile göndermeyi sağlayan *broadcast* çağrımını inceleyeceğiz. Kök düğümü her yeni eşzamanlama turuna girdiğinde, bu olay kök düğümünün saati ts ile etiketlenerek sistemde saklanır. Kök düğümü bu olayı mesaj verisi ve zaman etiketini parametre olarak alan *broadcast*(<

²<http://www.tinyos.net>

Algorithm 1 i Düzümündeki BSE Algoritması

```
1: İlkleme
2:  $msg_i.saat \leftarrow 0$ ;  $msg_i.olay \leftarrow 0$ ;
3:  $msg_i.kok \leftarrow i$ ;  $msg_i.sira \leftarrow 0$ ;
4: zamanlayıcıKur( $PERIOD$ )
5:
6:  $\square msg_i.kok = i$  olan düğümlerde yeni eşzamanlama turu
7:  $msg_i.olay \leftarrow yerelSaat()$ 
8:  $msg_i.saat \leftarrow msg_i.olay$ 
9:  $msg_i.sira \leftarrow msg_i.sira + 1$ 
10: broadcast( $< msg_i >$ ,  $msg_i.olay$ )
11:
12:  $\square msg_i.kok < msg_j.kok$  olan  $< msg_j >$  mesajı
alındığında
13:  $msg_i \leftarrow msg_j$ 
14:  $msg_i.olay \leftarrow zamanEtiketi(msg_j)$ 
15: tabloTemizle()
16: mesajEkle( $msg_i$ )
17:  $msg_i.saat \leftarrow tahminle()$ 
18: broadcast( $< msg_i >$ ,  $msg_i.olay$ )
19:
20:  $\square msg_i.sira < msg_j.sira$  ve  $msg_i.kok = msg_j.kok$ 
olan  $< msg_j >$  alındığında
21:  $msg_i \leftarrow msg_j$ 
22:  $msg_i.olay \leftarrow zamanEtiketi(msg_j)$ 
23: mesajEkle( $msg_i$ )
24:  $msg_i.saat \leftarrow tahminle()$ 
25: broadcast( $< msg_i >$ ,  $msg_i.olay$ )
```

.. $>$, ts) çağırmasını kullanarak yayınlar. broadcast, [6] çalışmasında olduğu gibi, mesaj iletim hattındaki belirsizlikleri otomatik olarak gideren ve gönderici taraftaki zaman etiketini alıcı tarafın saatindeki zaman etiketine otomatik olarak dönüştüren bir sistem servsidir.

BSE protokolüne ilişkin algoritma, Algoritma 1'de gösterilmiştir. BSE ağdaki düğümlerin her birinin ayrı kimlik numarasına sahip olmasına ihtiyaç duyar. Her i düğümü en güncel eşzamanlama verisini saklayan msg_i yapısını içermektedir. Bu yapının 4 sahası bulunmaktadır. kok o düğüm için kök düğümünün kimlik numarasını tutmaktadır. Kök düğümü, her eşzamanlama turu başlangıcında broadcast servisini kullanarak eşzamanlama mesajı yayınlamaktadır. $saat$ sahası kök düğümünün saatinin tahminini, ve $sira$ sahası ise kök düğümünden yayınlanan mesajın sıra numarasını içermektedir. $olay$ sahası ise kök düğümündeki güncel eşzamanlama turunun başlangıcının zaman etiketinin, i düğümünün yerel saatine dönüştürülmüş halini içermektedir. Daha önce belirtildiği gibi, bu dönüşüm işlemi iletişim katmanı tarafından otomatik olarak yapılmaktadır. msg_i yapısının dışında her düğüm kök düğümünün saatine ilişkin tahminleri içeren eşzamanlama mesajlarını biriktirmek için bir mesaj tablosu barındırmaktadır. Her düğüm, açıldığında $msg_i.kok$ sahasına kendi kimlik numarasını koyar. Böylelikle, her açılan düğüm algoritmayı kök düğümüymüş gibi çalıştırmaya başlar. Düğümler $PERIOD$ süre sonra ateşlenecek bir zamanlayıcı kurarak protokolü başlatırlar (1-4 Satırları). Eğer i düğümü kendi kimlik numarasından daha büyük bir kök

Algorithm 2 tahminle(): Kök Düğümünün Saatini Tahminler

```
1:  $toplam \leftarrow 0$ ;  $n \leftarrow 0$ ;  $saat \leftarrow 0$ ;  $hiz \leftarrow 0$ 
2: for  $msg_k$  ve  $msg_m$  ardışık mesajları
3:    $toplam \leftarrow toplam + (msg_m.saat - msg_k.saat) / (msg_m.olay - msg_k.olay)$ 
4:    $n \leftarrow n + 1$ 
5: end for
6:  $hiz \leftarrow toplam / n$ 
7:  $saat \leftarrow (yerelSaat() - msg_i.olay) * hiz + msg_i.saat$ 
8: return  $saat$ 
```

sahası içeren eşzamanlama mesajı almadıysa ve bir zamanlayıcı ateşlendiyse, yeni bir eşzamanlama turu başlamış olur. Zamanlayıcı ateşlenme olayı, düğümün saati ile etiketlenir ve bu değer $msg_i.olay$ ve $msg_i.saat$ sahaslarına yerleştirilir (Kök düğümleri için yerel zaman ve ağdaki genel zaman eşittir). $msg_i.sira$ sahası bir artırılarak $< msg_i >$ mesajı $msg_i.olay$ zaman etiketi ile yayınlanır(6-10 Satırları). i düğümü daha büyük bir kök düğümü kimlik numarası içeren mesaj aldığında (Satır 12), güncel eşzamanlama verisi msg_i yapısında saklanır (Satır 13). Böylece i düğümüne ait kök düğümü değişmiş ve kök düğümü seçimi eşzamanlama yapılırken beraberce yapılmış olunur. Eş zamanlama turunun başlangıcının i düğümünün yerel saatine dönüştürülmüş zaman etiketi, $msg_i.olay$ sahasında $zamanEtiketi()$ çağırımı ile yerleştirilir (Satır 14). Önceki kök düğümüne ilişkin mesajlar tablodan silinirken (Satır 15), yeni köke ilişkin veri mesaj tablosunda saklanmaktadır (Satır 16). Kök düğümünün saatinin ağdaki diğer düğümlere iletebilmesi için, eklenen yeni mesaj da kullanılarak yapılan saat tahmini (Satır 17) tekrardan yayınlanmaktadır (Satır 18). Eğer i düğümü kendi kök düğümünden daha güncel bir eşzamanlama mesajı alırsa (Satır 20), yeni kök seçtiğinde işlediği aynı adımları bu sefer mesaj tablosunu boşaltmadan yapmaktadır (21-25 Satırları).

3.2. Saat Sapmasının Hesaplanması

Her algılayıcı düğümünün donanımsal saati farklı tıklama hızına sahip olduğu için, eşzamanlama turunun başında düğümlerin saatleri iyi bir şekilde eşzamanlanmış olsa bile diğer tura kadar eşzamanlama hatası artmaktadır. Bu durumu engellemek için kullanılan yöntemlerden biri, yoğun enerji tüketimine yol açan eşzamanlama sıklığını arttırmaktır. Bunun yerine BSE'yi işleten her düğüm, kök düğümünün saatinin ilerleme hızını kendi donanım saatinin ilerleme hızı cinsinden hesaplar (göreceli hız).

Bir düğümün kök düğümün saatini tahminlediği algoritma Algoritma 2'de verilmiştir. BSE göreceli hızı ortalama karesel hatayı en aza indirerek yapmaktadır. Her düğüm N adet eşzamanlama mesajını mesaj tablosunda biriktirmektedir. Böylelikle, her düğüm eşzamanlama turlarının başlangıcının yerel saatleri cinsinden değerleri $\{H_0, \dots, H_{N-1}\}$ ve kök düğümünün saat değerleri $\{L_0, \dots, L_{N-1}\}$ verilerine erişebilmektedir. Kök düğümünün saatinin göreceli hızını hesaplamak için *fark kümeleri* $\{\partial H_0, \dots, \partial H_{N-2}\}$ ve $\{\partial L_0, \dots, \partial L_{N-2}\}$ hesaplanır. Son olarak, bu değerler kullanılarak göreceli hız kümesi $\{\partial h_0 = \partial H_0 / \partial L_0, \dots, \partial h_{N-2} = \partial H_{N-2} / \partial L_{N-2}\}$ hesa-

planır. Kök düğümünün saatinin tahmini göreceli hızı $\widehat{\partial h_i}$ olsun. Bu doğrultuda $\sum_{n=0}^{N-1} (\widehat{\partial h_i} - \partial h_n)^2$ değerini en aza indirmek için türevini alıp sıfıra eşitlesek bu fonksiyonun en küçük değerine tahminin $\partial h_i = \frac{1}{N-1} \sum_{n=0}^{N-1} \partial h_n$ değerini aldığında ulaştığını görürüz. Bu sonucu kullanarak, Algoritmanın 2. ve 6. satırları arasında, kök düğümünün göreceli hızının tahmini verimli bir şekilde yapılmaktadır. Göreceli hız kullanılarak anlık saat tahmini ise algoritmanın 7. satırında yapılmaktadır.

3.3. Kaynak Kullanımı ve Enerji Verimliliği

BSE doğrusal bağlanım yöntemi yerine, sadece bir adet ondalık sayılarla işleme gereksinim duyan basit ortalama alma yöntemini kullanmaktadır. Bu yöntem, ondalıklı işlemlerin özel yazılım kütüphaneleri ile yapıldığı kablosuz algılayıcı ağlarında işlemci yükü açısından daha hafif olmakla beraber, aynı zamanda daha da hızlıdır. BSE, kök düğümünden gelen saat değerlerinin tutulması için N adet eleman içeren bir tabloya ihtiyaç duymaktadır. Yayılım ağacı gibi topolojik bir altyapıya ihtiyaç duyulmadığı için, topolojik yapılara ilişkin veri tutulmasına gerek yoktur.

Algılayıcı düğümleri pil ile çalıştığından enerji kıt bir kaynaktır. İletişim sıklığının düşürülmesi enerji etkinliğini arttıracaktır. BSE saat sapmalarını hesapladığı için, iletişim sıklığı uzun tutulabilir. Ayrıca, yayılım ağacı gibi bir topolojik altyapıya ihtiyaç duyulmadığı için, bu ağacın kurulması ve eşzamanlama süresince yaşatılması için gereken ek mesajlaşma yükü ortadan kaldırılmıştır.

4. DEĞERLENDİRME

BSE protokolü TinyOS işletim sistemi kullanılarak gerçekleştirilmiştir. Protokolün değerlendirilmesi için ortak bir referans düğümünün kapsama alanı içerisinde yer alan 16 adet Iris³ algılayıcı düğümünden oluşan bir test ortamı hazırlanmıştır. Doğrusal bir topoloji oluşturmak için, yazılımsal olarak her düğümün kendi kimlik numarasının bir fazlası ya da bir eksikinden mesaj alması sağlanmıştır. Diğer düğümlerden gelen mesajlar bu sayede göz önüne alınmamaktadır. Düğümler tarafından gelen mesajların kaydedilmesi için PC'ye bağlı bir algılayıcı düğüm kullanılmıştır. Referans düğümü 20 ile 24 arasında düzenli bir dağılıma sahip her zaman diliminde bir mesaj yayınlamaktadır. Bu mesaj, diğer algılayıcı düğümleri tarafından yaklaşık olarak aynı zamanda alınır. Alınan mesaja düğümler tarafından bir zaman etiketi verilir ve bu andaki kök düğümünün saatinin tahminleri düğümler tarafından yayınlanır. PC'ye bağlı düğüm bu paketleri yakalayarak seri porta yönlendirmektedir. Seri portu dinleyen bir uygulama, bu mesajları kaydetmektedir.

Ölçümler sırasında BSE ve FTSP'nin eşzamanlama periyodları 30 saniye ve tablo boyutları 8 girdi olacak şekilde ayarlanmıştır. BSE ve FTSP protokolleri algılayıcı düğümlerine yükleyerek 4 saat boyunca düğümlerden saat tahminleri toplanmıştır. Değerlendirmede temel ölçüt olarak herhangi iki düğüm arasındaki (*genel fark*) ve iki komşu düğüm arasındaki (*yerel fark*) anlık eşzamanlama hataları gözlemlenmiştir.

Table 1: BSE ve FTSP için ölçüm özetleri

Saat farkı	BSE	FTSP
Max. Genel	15 μsec	167 μsec
Ortalama Genel	5 μsec	35 μsec
Max. Yerel	8 μsec	93 μsec
Ortalama Yerel	1 μsec	6 μsec

Tablo 1 ölçümleri özetlemektedir. Ölçüm sonuçları, FTSP ile kıyaslandığında BSE'in oldukça iyi bir başarıma sahip olduğunu göstermektedir.

5. GELECEK ÇALIŞMALAR

Bu çalışmada Basit Saat Eşzamanlama (BSE) protokolü tanıtılmıştır. Yükte hafif yöntemlerle ağ genelinde saat eşzamanlaması sağlayan protokol, algılayıcı ağlarında pratikte kullanılan FTSP protokolü ile kıyaslanmıştır. Deneysel ölçümler BSE'in FTSP'den daha iyi başarıma sahip olduğunu göstermiştir. Gelecekte BSE'in daha büyük ağlarda başarımını ölçmek için benzetimler yapmayı ve enerji tüketimini daha da azaltacak yöntemleri eklemeyi planlamaktayız.

6. KAYNAKÇA

- [1] K. Römer, P. Blum, and L. Meier, "Time synchronization and calibration in wireless sensor networks," in *Handbook of Sensor Networks: Algorithms and Architectures*, I. Stojmenovic, Ed. John Wiley & Sons, 2005, pp. 199–237.
- [2] J. van Greunen and J. Rabaey, "Lightweight time synchronization for sensor networks," in *WSNA '03: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*. New York, NY, USA: ACM, 2003, pp. 11–19.
- [3] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2003, pp. 138–149.
- [4] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 147–163, 2002.
- [5] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi, "The flooding time synchronization protocol," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2004, pp. 39–49.
- [6] B. Kusy, P. Dutta, P. Levis, M. Maroti, A. Ledeczi, and D. Culler, "Elapsed time on arrival: A simple and versatile primitive for canonical time synchronisation services," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 1, no. 4, pp. 239–251, 2006.
- [7] C. Lenzen, P. Sommer, and R. Wattenhofer, "Optimal Clock Synchronization in Networks," in *7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Berkeley, California, USA, November 2009.

³<http://www.xbow.com>