

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

(DOKTORA TEZİ)

**KABLOSUZ ALGILAYICI AĞLARINDA
SAAT EŞZAMANLAMASI**

Kasım Sinan YILDIRIM

Tez Danışmanı: Doç. Dr. Aylin KANTARCI

Bilgisayar Mühendisliği Anabilim Dalı

Bilim Dalı Kodu: 619.01.00

Sunuş Tarihi: 4.Nisan.2012

Bornova-İZMİR

2012

Kasım Sinan YILDIRIM tarafından Doktora tezi olarak sunulan "**KABLOSUZ ALGILAYICI AĞLARINDA SAAT EŞZAMANLAMASI**" başlıklı bu çalışma E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliği ile E.Ü. Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi'nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve **12.04.2012** tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

Jüri Üyeleri:

İmza

Jüri Başkanı: Doç. Dr. Aylin KANTARCI

.....

Raportör Üye: Doç. Dr. Aybars UĞUR

.....

Üye: Prof. Dr. Efendi NASİBOV

.....

Üye: Prof. Dr. Oğuz DİKENELLİ

.....

Üye: Doç. Dr. Öznur ÖZKASAP

.....

ÖZET

KABLOSUZ ALGILAYICI AĞLARINDA SAAT EŞZAMANLAMASI

YILDIRIM, Kasım Sinan

Doktora Tezi, Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Aylin KANTARCI

12.04.2012, 130 sayfa

Kendi yerel zaman anlayışına sahip olan algılayıcı düğümleri arasında genel bir zaman kavramı oluşmasını sağlayan dağıtık saat eşzamanlaması, kablosuz algılayıcı ağlarındaki (KAA) çoğu uygulama ve protokolün doğru ve verimli çalışması için önemli bir gereksinimdir. Bu tezde, KAA'ndaki saat eşzamanlaması için çeşitli katkılar yapılmıştır.

Tezin ilk katkısı, düğümlerin bir dağıtık uzlaşma algoritması yürüterek ortak bir saat hızında anlaşmalarının yavaş sel yöntemiyle saat eşzamanlamasının başarımını ve ölçeklenebilirliğini büyük ölçüde arttırdığının gösterilmesidir. MICAz ortamındaki deneyler uygulanan uzlaşma algoritması sayesinde yavaş selin eşzamanlama başarımının yaklaşık 20 kat arttığını ortaya koymuştur.

Tezin ikinci katkısı, uygulamadaki çoğu eşzamanlama protokolü tarafından kullanılan en küçük kareler yönteminin düğümlerin saatlerinin geri alınmasına yol açabildiğinin gösterilmesi ve bu durumu engelleyecek bir yöntemin önerilmesidir.

Tezin üçüncü katkısı, birbirine komşu düğümler arasındaki eşzamanlama hatasını eniyilerken düğümlerin dışsal bir zaman kaynağına da eşzamanlanması problemini çözen bir protokolün geliştirilmesi ve bu protokolün başarımının değerlendirilmesidir.

Tezin son katkısı, literatürdeki algoritmalarından daha kolay gerçekleştirilebilir bir optimal meyilli saat eşzamanlama algoritmasının tasarlanmasıdır.

Anahtar Sözcükler: Kablosuz Algılayıcı Ağları, Saat Eşzamanlaması, Dağıtık Uzlaşma, En Küçük Kareler, Meyilli Saat Eşzamanlaması.

ABSTRACT**CLOCK SYNCHRONIZATION
IN WIRELESS SENSOR NETWORKS**

YILDIRIM, Kasım Sinan

PhD. in Computer Engineering

Supervisor: Assoc. Prof. Dr. Aylin KANTARCI

12.04.2012, 130 pages

The accurate and efficient operation of many applications and protocols in wireless sensor networks (WSNs) require distributed clock synchronization which provides global notion of time for the sensor nodes with local notion of time. This thesis presents several contributions to the clock synchronization in WSNs.

The first contribution of this thesis is to show that agreement on a common clock rate by employing a distributed agreement algorithm among the sensor nodes improves the performance and scalability of clock synchronization based on slow flooding considerably. The experiments on MICAz platform showed that with the agreement algorithm, the synchronization performance of slow flooding is improved by a factor of 20.

The second contribution of this thesis is to show that the method of least-squares, which is employed by most of the time synchronization protocols in WSNs, may entail the clocks of the sensor nodes being set back. We present a method to prevent this situation.

The third contribution of this thesis is to develop and evaluate a protocol which solves the problem of external synchronization while optimizing the clock skew between neighboring nodes.

The last contribution of this thesis is to design an optimal gradient clock synchronization algorithm which can be implemented more easily than the algorithms in the literature.

Keywords: Wireless Sensor Networks, Clock Synchronization, Distributed Agreement, Least-squares, Gradient Clock Synchronization.

TEŞEKKÜR

Öncelikle insanlığı aydınlatan bilim devlerine ve onların omuzlarına teşekkürü bir borç bilirim. Bana en büyük desteği sağlayan annem Sülbiye Yıldırım'a, babam Yasin Yıldırım'a, kardeşim Fikri Onur Yıldırım'a ve üzerimde çok büyük emeği olan anneannem Şengül Yılmaz'a sonsuz teşekkürlerimi sunuyorum.

Tez süresince yardımlarını ve desteklerini esirgemeyen başta değerli danışmanım Doç. Dr. Aylin Kantarcı olmak üzere herkese teşekkür ederim.

Son olarak, 2211-Yurt İçi Doktora Burs Programı kapsamında beni destekleyen TÜBİTAK'a teşekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	v
ABSTRACT	vii
TEŞEKKÜR	ix
ŞEKİLLER DİZİNİ.	xxii
ÇİZELGELER DİZİNİ	xxiii
ALGORİTMALAR DİZİNİ	xxv
SİMGELER VE KISALTMALAR DİZİNİ	xxvii
1 GİRİŞ.	1
1.1 Dağıtık Sistemlerde Saat Eşzamanlaması Problemi	1
1.2 Saat Eşzamanlamasının Kablosuz Algılayıcı Ağlarındaki Önemi.	2
1.3 Saat Eşzamanlamasının Kablosuz Algılayıcı Ağlarına Özgü Nitelikleri	3
1.4 Literatüre Katkıları	4
1.5 Teze Genel Bakış	6
2 KABLOSUZ ALGILAYICI AĞLARINDA SAAT EŞZAMANLAMASINA GİRİŞ.	9
2.1 Algılayıcı Düğümlerindeki Donanımsal Saatler	9
2.2 Saat Eşzamanlama Protokolleri ve Mantıksal Saat Kavramı.	10
2.3 Eşzamanlama Hatalarının Kaynakları	11

İÇİNDEKİLER (Devam)

	<u>Sayfa</u>
2.4 OEK Katmanında Zaman Etiketleme	12
2.5 Saat Eşzamanlama Yaklaşımları	13
2.6 Temel Başarım Ölçütleri	14
2.7 Kablosuz Algılayıcı Ağları İçin Tasarlanmış Literatürdeki Temel Protokoller .	15
2.7.1 LTS protokolü	15
2.7.2 RBS protokolü	16
2.7.3 TPSN protokolü	17
2.7.4 FTSP protokolü	17
2.7.5 PulseSync protokolü	18
2.7.6 RITS protokolü	19
2.7.7 RATS protokolü	19
2.7.8 TCTS protokolü	20
2.7.9 GTSP protokolü	20
2.7.10 RFA protokolü	21
2.8 Sonuçlar	21
3 KABLOSUZ ALGILAYICI AĞLARINDA SAAT EŞZAMANLAMASI PROBLEMİ İÇİN SİSTEM MODELİ	23
3.1 Algılayıcı Ağının Matematiksel Modeli	23

İÇİNDEKİLER (Devam)

	Sayfa
3.2 Algılayıcı Düğümlerinin Saatlerinin Matematiksel Modeli	23
3.2.1 Saat eşzamanlama algoritmaları ve eşzamanlama hatası	24
3.3 Algılayıcı Düğümleri Arasındaki İletişimin Matematiksel Modeli ve Analizi .	25
3.3.1 İki düğüm arasındaki eşzamanlama sürecinde basit bir iletişim senaryosu . .	25
3.3.2 Analiz.	26
4 ALGILAYICI DÜĞÜMLERİNİN SAATLERİ ARASINDA DOĞRUSAL BİR İLİŞKİ KURMA	29
4.1 İlgili Çalışmalar	29
4.2 En Küçük Kareler Yöntemi İle Doğrusal Bağlanımın Saatlerin Eşzamanlamasında Kullanılması	29
5 KABLOSUZ ALGILAYICI AĞLARINDA YAVAŞ SEL TABANLI SAAT EŞZAMANLAMASI.	33
5.1 İlgili Çalışmalar	34
5.2 Kablosuz Algılayıcı Ağlarında Zaman Bilgisinin Yavaş Yayılımı	35
5.2.1 En küçük kareler yöntemini kullanan yavaş selin eşzamanlama hatası. . . .	37
5.3 Eğim Geçmişi Kullanan Yavaş Sel Yöntemi	39
5.3.1 EG'deki medyan eğimi göz önüne almak (ME)	39
5.3.2 EG'deki ortalama eğimi göz önüne almak (OE)	40
5.3.3 EG'deki ortalama eğimle birlikte güncel en küçük kareler eğimini göz önüne almak (OGE)	41

İÇİNDEKİLER (Devam)

	<u>Sayfa</u>
5.4 En Küçük Varyansa Sahip İkili Eğimi (EKVSİE) Kullanan Yavaş Sel Yöntemi	42
5.4.1 En küçük kareler eğimi ve EKVSİE eğimlerinin karşılaştırılması	44
5.5 Hız Seli Eşzamanlama Protokolü (HSEP).	45
5.5.1 Analiz.	47
5.6 Saat Hızı Uzlaşması İle Sel Protokolü	49
5.6.1 Analiz.	51
5.7 Deneysel Sonuçlar	53
5.7.1 Göz önüne alınan başarımlar ölçütleri	54
5.7.2 Deneyler için kullanılan algılayıcı düğümleri	54
5.7.3 Deney düzeneği	54
5.7.4 Eğim geçmişi kullanan yöntemler için deneysel sonuçlar.	55
5.7.5 En küçük varyansa sahip ikili eğim yöntemi için deneysel sonuçlar.	58
5.7.6 HSEP ve SHUS protokolleri için deneysel sonuçlar	61
5.7.7 Deneysel sonuçların özeti	63
5.8 SHUS Protokolü İçin İyileştirme Önerileri	63
5.9 Sonuçlar ve Gelecek Çalışmalar	65

İÇİNDEKİLER (Devam)

	<u>Sayfa</u>
6 KABLOSUZ ALGILAYICI AĞLARINDA SAAT EŞZAMANLAMASI SÜRECİNDE SAATLERİN GERİ ALINMASINI ENGELLEYEN BİR YÖNTEM	67
6.1 İlgili Çalışmalar	67
6.2 En Küçük Kareler Yönteminin Temel Eksikliği	68
6.3 Mantıksal Saatlerin Geri Alınmasını Engelleyen Bir Yöntem	69
6.4 Deneysel Sonuçlar	70
6.5 Tartışma	73
7 KABLOSUZ ALGILAYICI AĞLARI İÇİN DIŞSAL MEYİLLİ SAAT EŞZAMANLAMASI.	75
7.1 İlgili Çalışmalar	76
7.2 GTSP Protokolüne Genel Bakış.	77
7.2.1 GTSP protokolünün temel eksikliği	79
7.3 Dışsal Meyilli Saat Eşzamanlaması Protokolü	80
7.3.1 Gerçek zamana eşzamanlanma	82
7.4 Deneysel Sonuçlar	84
7.4.1 DMSEP protokolü için benzetim sonuçları	91
7.5 Sonuçlar	92

İÇİNDEKİLER (Devam)

	<u>Sayfa</u>
8 OPTİMAL MEYİLLİ SAAT EŞZAMANLAMA ALGORİTMALARININ GERÇEKLEŞTİRİMLERİNİ KOLAYLAŞTIRMAK ÜZERİNE BİR ÇALIŞMA	95
8.1 İlgili Çalışmalar	96
8.2 Sistem Modeli İçin Değişiklikler ve Eklentiler	98
8.3 Literatürdeki Optimal Meyilli Saat Eşzamanlama Algoritmalarının Gerçekleştirim Zorlukları	99
8.4 Kolay Gerçekleştirilebilir Meyilli Saat Eşzamanlaması Sistemi	100
8.4.1 Mantıksal saat modülü	101
8.4.2 Tahminleme modülü	102
8.4.3 Eşzamanlama modülü	104
8.4.4 Literatürdeki diğer algoritmalarla karşılaştırma	106
8.5 Yazılımsal Gerçekleştirim	106
8.6 Donanımsal Gerçekleştirim İçin Taslak Tasarım	107
8.7 Sonuçlar ve Tartışma	108
8.8 Yardımcı Teorem 8.1'in İspatı	109
8.9 Teorem 8.2'nin İspatı	111
9 SONUÇLAR	119
9.1 Katkıların Özeti	119

İÇİNDEKİLER (Devam)

	<u>Sayfa</u>
9.2 Gelecek Çalışmalar	120
9.3 Sonuç	121
KAYNAKLAR DİZİNİ	123
ÖZGEÇMİŞ	127
TERİMLER SÖZLÜĞÜ.	129

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
1.1 MicaZ algılayıcı düğümü	2
1.2 Sel yöntemi ile eşzamanlama.	5
1.3 Hareketli bir nesnenin tespit edilmesi.	5
2.1 Mica2 düğümünün saat frekansının sıcaklık ile değişimi (Lenzen et al., 2009b).	10
2.2 İki düğüm arasındaki iletişim sürecinde meydana gelen mesaj gecikmesinin, belirlenimci ve belirlenimci olmayan bileşenleri. Bu bileşenlerin büyüklükleri (Maróti et al., 2004) çalışmasındaki Tablo 1'den alınmıştır.	11
5.1 FTSP protokolünde zaman bilgisinin sel yöntemi ile yavaş bir şekilde ağa yayılımı.	37
5.2 5 eşzamanlama noktası için ikili eğimler. 3. eğim, tahmini doğrusal bağlanım doğrusu için EKVSİE tahminleyicisidir.	43
5.3 FTSP protokolünü sırasıyla EKK, ME, OE ve OGE eğimlerini kullanarak yürüten düğümler için anlık genel (sol sütun) ve yerel (sağ sütun) saat farkı değerleri.	56
5.4 FTSP protokolünü sırasıyla EKK, ME, OE ve OGE eğimlerini kullanarak yürüten düğümlerden toplanan eğim değerleri (sol sütun) ve kimliği 1 olan dayanak düğümü ile diğer düğümler arasında görülen mantıksal saat farkının en büyük değerleri (sağ sütun). Eğim değerleri, gerçek değerlerden 1.0 çıkartılmış şekilde gösterilmektedir.	57

ŞEKİLLER DİZİNİ (Devam)

Şekil	Sayfa
5.5 EKVSİE yöntemini kullanan FTSP protokolünün, doğrusal ilingeye sahip 20 MICAz algılayıcı düğümü üzerinde yürütüldüğünde elde edilmiş olan sırası ile anlık en büyük genel saat farkı, en büyük ortalama genel saat farkı, en büyük yerel saat farkı, en büyük ortalama yerel saat farkı, eğim değerleri ve dayanak düğümü ile diğer düğümler arasında gözlemlenen en büyük saat farkı. Eğim değerleri, gerçek değerlerden 1.0 çıkartılmış şekilde gösterilmektedir.	59
5.6 FTSP protokolünün EKK ve EKVSİE yöntemleri için benzetim sonuçları.	60
5.7 HSEP ve SHUS protokolleri için ölçülen anlık genel ve yerel saat farkı değerleri.	61
5.8 HSEP ve SHUS protokolleri için toplanan eğim değerleri ve dayanak düğümü ile diğer düğümler arasında görülen en büyük saat farkı değerleri. Eğim değerleri, gerçek değerlerden 1.0 çıkartılmış şekilde gösterilmektedir.	62
5.9 SHUS protokolü için farklı ağ çapları için benzetim sonuçları. . . .	62
6.1 Yeni bir eşzamanlama mesajı alındığında hesaplanan en küçük kareler doğruları.	68
6.2 Güncel en küçük doğrusunun mantıksal saatlerin geri alınmaması için kaydırılması.	69
6.3 FTSP (sol sütun) ve değiştirilmiş FTSP (sağ sütun) için sırasıyla en küçük kareler doğrusunun eğimi, genel saat farkı değerleri, yerel saat farkı değerleri, dayanak düğümü ve diğer düğümler arasında gözlemlenen en büyük saat farkı değerleri. Eğim değerleri, gerçek değerlerden 1.0 çıkartılmış şekilde gösterilmektedir.	72

ŞEKİLLER DİZİNİ (Devam)

Şekil	Sayfa
7.1 Birbirine komşu düğümler, eşzamanlama bilgisini farklı yollardan aldıkları için, bu düğümler arasında büyük eşzamanlama hataları görülebilir.	77
7.2 GTSP (sol) and DMSEP (sağ) protokollerinin karşılaştırılması. GTSP protokolünde, düğümler komşularıyla bir uzlaşma protokolü yürüterek ortak bir saat hızında ve saat değerinde anlaşmaktadırlar. DMSEP protokolünde ise, özel bir düğüm ek olarak kendi hız çarpanını ve zaman öteleme değerini ağa sel ile yayarak diğer düğümlerin kendi saat hızında ve saat değerinde anlaşmalarını sağlamaktadır.	83
7.3 Algılayıcı düğümlerin halka ve doğrusal ilingelerde yerleşimi. . . .	84
7.4 FTSP, GTSP ve DMSEP protokolleri için doğrusal ilingede ölçülmüş genel ve ortalama genel (sol sütun), yerel ve ortalama yerel (sağ sütun) saat farkı değerleri.	85
7.5 FTSP (sol sütun) ve DMSEP (sağ sütun) protokolleri için doğrusal ilingede, dayanak düğümü (1 kimliğine sahip) ile diğer düğümler arasında gözlemlenen en büyük saat farkı değerleri.	86
7.6 Sırasıyla FTSP, GTSP ve DMSEP protokolleri için doğrusal ilingede, 19 ve 20 kimliğine sahip düğümler arasında gözlemlenen eşzamanlama hatası (sol sütun) ve düğüm başına gözlemlenen en büyük yerel saat farkı (sağ sütun).	87
7.7 FTSP, GTSP ve DMSEP protokolleri için halka ilingede ölçülmüş genel ve ortalama genel (sol sütun), yerel ve ortalama yerel (sağ sütun) saat farkı değerleri.	89
7.8 FTSP (sol sütun) ve DMSEP (sağ sütun) protokolleri için halka ilingede, dayanak düğümü (1 kimliğine sahip) ile diğer düğümler arasında gözlemlenen en büyük saat farkı değerleri.	89

ŞEKİLLER DİZİNİ (Devam)

Şekil	Sayfa
7.9 Sırasıyla FTSP, GTSP ve DMSEP protokoleri için halka ilingede, 9 ve 10 kimliğine sahip düğümler arasında gözlemlenen eşzamanlama hatası (sol sütun) ve düğüm başına gözlemlenen en büyük yerel saat farkı (sağ sütun).	90
7.10 DMSEP protokolü için farklı çaplara sahip doğrusal (sol) ve halka (sağ) ilingelerinde benzetim sonuçları.	92
8.1 Meyilli saat eşzamanlama sisteminin modüler yapısı. Şekildeki oklar sistem içerisindeki modüller arasındaki etkileşimi göstermektedir.	101
8.2 Eşzamanlama Modülü'ndeki mantıksal saatin hızının Java ile gerçekleştirimi.	107
8.3 Yazılım/donanım işbirliği ile gerçekleştirilen Saat Eşzamanlama Bileşeni (SEB).	107

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
2.1 Özetlenen eşzamanlama protokollerinin saat eşzamanlama yaklaşımlarına genel bakış.	21
5.1 Deneysel sonuçlarının özeti.	63
7.1 FTSP, GTSP ve DMSEP protokolleri için gözlemlenen deneysel sonuçların özeti.	91
8.1 SEB'in örnek işletimi için parametreler.	106

ALGORİTMALAR DİZİNİ

Algoritma	Sayfa
5.1 FTSP'nin v düğümü için sözde kodu.	35
5.2 v düğümü için HSEP sözde kodu.	46
5.3 v düğümü için, kimliği $KÖK$ olan sabit bir dayanak düğümüne sahip SHUS sözde kodu.	50
6.1 En küçük kareler yöntemi için önerilen değişiklik.	70
7.1 v düğümü için GTSP sözde kodu.	78
7.2 v düğümü için ref sabit dayanak düğümüne sahip DMSEP sözde kodu.	80
7.3 DMSEP protokolünü yürüten v düğümü için dayanak düğümünün saatini döndüren dayanakSaati arayüzü.	80
8.1 Lenzen et. al. (Lenzen et al., 2010) çalışmasında sunulan optimal meyilli saat eşzamanlama algoritmasının parçalarından Algoritma 3'ün 1. satırı.	99
8.2 Kuhn et. al. (Kuhn and Oshman, 2009) çalışmasında sunulan optimal meyilli saat eşzamanlama algoritması.	99
8.3 \mathcal{A}^{saat} algoritması: u düğümünün mantıksal saatini temsil eder. . . .	102
8.4 \mathcal{A}^{tahmin} algoritması: Komşu düğümlerin mantıksal saat değerlerini tahmin eder.	103
8.5 u düğümü için \mathcal{A}^{meyil} algoritması.	104

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltma Açılım

EEZ	Eşgüdümlü Evrensel Zaman (<i>Universal Coordinated Time</i> - UTC)
KAA	Kablosuz Algılayıcı Ağları (<i>Wireless Sensor Networks</i> - WSN)
KKS	Küresel Konumlanma Sistemi (<i>Global Positioning System</i> - GPS)
OEK	Ortama Erişim Kontrolü (<i>Medium Access Control</i> - MAC)
ZBÇE	Zaman Bölüşümlü Çoklu Erişim (<i>Time Division Multiple Access</i> - TDMA)

*“Bir saate sahip olan kiři zamanı bilir,
iki saate sahip olan kiři hiçbir zaman emin deęildir.”*

Segal Kanunu

1 GİRİŞ

Dağıtık sistemlerde saat eşzamanlaması, üzerinde derinlemesine çalışılmış bir konu olmasına rağmen halen birçok açık problemi barındıran özgün bir konudur. Kablosuz iletişim devrimi ve bunun devamındaki süreçte geliştirilen kablosuz iletişim yeteneğine sahip küçük algılayıcı cihazları, saat eşzamanlaması alanına birçok yeni problem kazandırmıştır. Bir başka deyişle, saat eşzamanlama problemi algılayıcı ağlarına özgü yeni gereksinimler ve parametreler ile adeta yeniden tanımlanmıştır.

1.1 Dağıtık Sistemlerde Saat Eşzamanlaması Problemi

Dağıtık sistemler birbirinden fiziksel olarak ayrılmış *düğüm* adı verilen birçok özerk bilgisayardan oluşmaktadırlar. Düğümler, birbirleri ile iletişim kurarak ortak bir görevi yerine getirmektedirler. Birçok dağıtık uygulama ve protokol, düğümler arasındaki eylemleri eşgüdümlemek ya da meydana gelen olaylar arasında önce/sonra şeklinde sıralamalar yapabilmek için ortak bir zaman kavramına gereksinim duymaktadır. Dağıtık sistemlerde, bütün düğümlerin ortak bir saat kaynağına doğrudan erişimleri mümkün değildir. Bunun yerine her düğüm içsel bir saat ile donatılmıştır. Düğümlerin saatleri farklı hızlarda tıklayabilmekte ve bunun sonucunda saatlerin gösterdiği değerler birbirinden uzaklaşabilmektedir. Bu durum her düğümün kendi yerel zaman kavramına sahip olmasına yol açmaktadır. Dağıtık sistemlerde, düğümlerin ortak bir zaman kavramına sahip olmalarını sağlayan sürece *dağıtık saat eşzamanlaması* denilmektedir. Bu süreçte, her düğüm bir *dağıtık saat eşzamanlama algoritması* çalıştırarak diğer düğümlerle ve zaman sunucuları ile haberleşir ve elde ettiği zaman bilgisini işleyerek bir ortak saat değerini hesaplar. Dağıtık saat eşzamanlama algoritmasının temel hedefi, herhangi bir anda düğümlerin hesapladığı ortak saat değerleri arasındaki farkın en aza indirilmesidir.



Şekil 1.1: MicaZ algılayıcı düğümü

1.2 Saat Eşzamanlamasının Kablosuz Algılayıcı Ağlarındaki Önemi

Günümüzde, dağıtık sistemlerin bir alt kolu olan *Kablosuz Algılayıcı Ağları* (KAA)¹ birçok araştırmacının ilgisini çekmektedir. KAA küçük, ucuz, düşük güç tüketimine sahip, çevreyi *algılayıcıları* ile algılayabilen ve kablosuz iletişim yeteneğine sahip *algılayıcı düğümlerinden* oluşmaktadır. Şekil 1.1’de MicaZ² platformuna ait bir algılayıcı düğümü gösterilmektedir. Birbirinden fiziksel olarak ayrı duran algılayıcı düğümleri, işbirliği içerisinde çevresel verileri toplar, işler ve ağa tümegönderim ile yayarlar. KAA’nın özgün uygulamaları hedef takibi, çevrenin gözetlenmesi ve insan erişiminin zor olduğu bölgelerden veri toplanması şeklinde örneklendirilebilir.

Saat eşzamanlaması, diğer dağıtık sistemlerde olduğu gibi, KAA’nda da önemli bir süreçtir. Ortak zaman kavramı, KAA’ndaki birçok uygulama ve protokolün verimli ve doğru yürütümü için vazgeçilmez bir gereksinimdir. Örneğin, Zaman Bölüşümlü Çoklu Erişim (ZBÇE)³ protokolünde, algılayıcı düğümlerine tümegönderim yapacakları zaman dilimleri atanmaktadır. Bu sayede birbirlerinin kapsama alanında olan iki düğümün aynı anda iletişime geçmesi ve bunun sonucunda bu düğümlerin paketlerinin çarpışması engellenmektedir. Düğümlerin kendilerine atanmış olan zamanlarda tümegönderim yapmaları ancak saatlerinin eşzamanlanmış olması ile mümkün olmaktadır.

Saat eşzamanlanmasının büyük önem taşıdığı bir başka örnek, algılayıcı düğümlerinin enerji tüketiminin eniyilenmesi için verilebilir. Pille çalışan algılayıcı düğümleri için enerji tüketimi hayati bir konudur. İletişim yongası, bir algılayıcı düğümünün donanımında yer alan birimler içinde en fazla enerji tüketen bileşendir. Düğümlerin enerji tüketimi, iletişim devrelerinin eşzamanlı açılıp kapatılması ile

¹*İngilizcesi: Wireless Sensor Networks - WSN.*

²www.memsic.com

³*İngilizcesi: Time Division Multiple Access - TDMA.*

oldukça düşmektedir. Bu tip eşgüdümlü eylemler, saatlerin eşzamanlanmadığı durumlarda doğru bir şekilde yapılamamaktadır.

Ortak bir zaman kavramına ihtiyaç duyan KAA uygulamalarına temel örnek olarak, veri toplama uygulamaları gösterilebilir. Bu tip uygulamalarda, algılayıcı düğümleri gözlemledikleri verileri, bu verilere bir zaman etiketi ekleyerek, bir merkez istasyonuna göndermektedirler. Merkez istasyonunda toplanan veriler üzerinde yapılan yorumların anlamlı olması için, zaman etiketlerinin tutarlı olması gerekmektedir.

1.3 Saat Eşzamanlamasının Kablosuz Algılayıcı Ağlarına Özgü Nitelikleri

Geleneksel dağıtık sistemler için tasarlanmış olan saat eşzamanlama protokolleri, hata dayanıklılığı, enerji tüketimi ve kaynak kullanımı noktalarındaki eksiklikleri nedeniyle KAA'na uygulanabilir değildirler. Internete bağlı bilgisayarların saatlerini eşzamanlayan *Network Time Protocol (NTP)* (Mills, 1991) protokolü örneğinde olduğu gibi, geleneksel protokoller yoğun bir iletişim yüküne sahiptir. Buna ek olarak, çoğu geleneksel protokol düğümlerin önceden belirlenmiş sıradüzensel bir şekilde yerleştirilmesine ve düğümlerin elle düzenlenmesine ihtiyaç duyarlar. Tüm bu özellikler ve gereksinimler, doğası gereği KAA için uygun değildir.

Geleneksel dağıtık sistemlerin aksine, enerji KAA için çok kısıtlı bir kaynaktır, çünkü algılayıcı düğümlerinin güç kaynağı pildir. Enerji verimliliği, algılayıcı düğümleri için tasarlanan donanım ve yazılımların ortak ve ana hedefidir. İletişim enerji tüketimi açısından oldukça maliyetli olduğu için, saat eşzamanlama protokolleri iletişimi en aza indirmek ve iletişimde gönderilecek veriyi mümkün olduğunca kısa tutmalıdır. Küresel Konumlandırma Sistemi (KKS)⁴ gibi donanımsal çözümler, maliyetleri, büyüklükleri ve getirdikleri enerji yükleri nedeniyle bütün düğümler tarafından eşzamanlanma için kullanılamaz.

KAA oldukça devingen bir yapıya sahiptir. Düğümlerin konumları çevresel etkenler yüzünden değişebilir, donanımsal hatalar yüzünden düğümler kapanabilir, ağa yeni düğümler eklenebilir. Ağın ilingesi tahmin edilemez bir şekilde sık değişebilir ve düğümler arasında bağlantıların durağanlığı garanti edilemez. Düğümler arası iletişimde yoğun paket kayıpları olabilir. Tüm bunlara ek olarak, algılayıcı düğümlerinin işleme yetenekleri ve veri depolama kapasiteleri

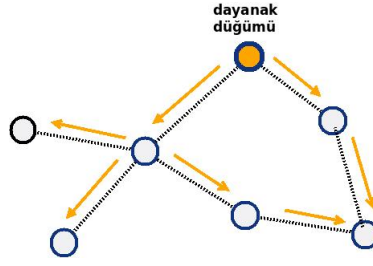
⁴İngilizcesi: Global Positioning System - GPS.

de sınırlıdır. Özetle, KAA için tasarlanacak saat eşzamanlama protokolleri, geleneksel protokollerin aksine ağır devingenliğini ve algılayıcı düğümlerinin sınırlı yeteneklerini gözetmek zorundadırlar.

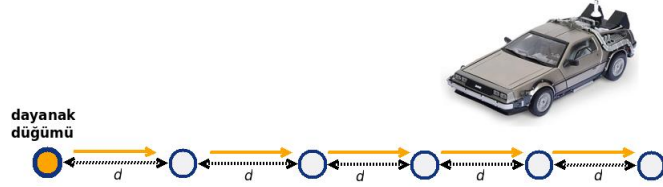
1.4 Literatüre Katkılar

Bu tez kapsamında, KAA'na yönelik saat eşzamanlaması konusuna üç temel katkı yapılmıştır. Bu katılardan ilki, ağ genelinde saat eşzamanlamasının bir dayanak düğümünün kendi saat bilgisini ağa yayması ile sağlandığı, Şekil 1.2'de gösterilen sel yöntemi ile saat eşzamanlamasına yöneliktir. Düğümlerin algılayıcılarından periyodik olarak okudukları sıcaklık değerlerini zaman etiketleri ile beraber bir merkezi istasyona gönderdiği *veri toplama* uygulaması, sel yöntemi ile eşzamanlamanın yeterli olduğu uygulamalara bir örnek olarak verilebilir. Literatürdeki genel geçer ve sel yöntemini kullanan Flooding Time Synchronization Protocol (FTSP) (Maróti et al., 2004) protokolünde selin yayılım hızı, düğümlerin kendilerine gelen zaman bilgisini alır almaz tümegöndermemeleri nedeniyle yavaştır. Selin yavaş yayılımı, ağa yayılan zaman bilgisinin her sekmedeki bekleme süreleri nedeniyle bozulmasına yol açar. Dolayısıyla, dayanak düğümüne daha uzakta olan düğümler zaman bilgisini büyük hatalarla aldıkları için büyük eşzamanlama hatalarına sahip olmaktadır. Bu durum FTSP protokolünün ölçeklenebilirliğini olumsuz etkilemektedir. Literatürde, selin yavaş olan hızının saat eşzamanlaması başarımı üzerindeki olumsuz etkisini ortadan kaldırmak için düğümlerin yeni bir güncel zaman bilgisini alır almaz hızlı bir şekilde ağa yayması, yani bekleme sürelerinin ortadan kaldırıldığı hızlı sel yöntemi önerilmiştir (Lenzen et al., 2009b). Ancak hızlı sel yöntemi, düğümlerin kendi komşuları ile iletişimlerini bitirmeden seli ilerletmeyeceklerinden dolayı KAA'nda oldukça yavaş olabilmektedir (Lu and Whitehouse, 2009; Schmid et al., 2010a). Bu bağlamda, tezin 5. bölümünde, yavaş selin saat eşzamanlaması üzerindeki olumsuz etkisini selin yayılım hızını değiştirmeden azaltacak yöntemler önerilmiştir. Bu bölümde yapılan ana katkı, yavaş sel yönteminin eşzamanlama hatasının ve ölçeklenebilirliğinin düğümler arasında bir saat hızı uzlaşmasıyla büyük ölçüde iyileştirilebileceğinin gösterilmesidir.

Tez kapsamında yapılan bir diğer katkı, KAA'na yönelik saat eşzamanlama protokolleri tarafından düğümlerin saatleri arasında doğrusal bir ilişki kurmak için yaygın bir şekilde kullanılan en küçük kareler yönteminin temel bir eksikliğinin ortaya koyulmasıdır. Bu yöntem, saat eşzamanlama protokollerinin yürütümü sırasında düğümlerin saatlerinin geriye alınabilmesine yol açmaktadır. Bu durum, teorik ve pratik çalışmalarda düğümlerin saatlerinin monoton artan fonksiyonlar



Şekil 1.2: Sel yöntemi ile eşzamanlama.



Şekil 1.3: Hareketli bir nesnenin tespit edilmesi.

olduğu yönündeki varsayımla çalışmaktadır (Lundelius and Lynch, 1984; Halpern et al., 1985; Biaz and Welch, 2001; Patt-Shamir and Rajsbaum, 1994; Srikanth and Toueg, 1987; Fan and Lynch, 2006; Locher and Wattenhofer, 2006; Lenzen et al., 2008; Pussente and Barbosa, 2009; Kuhn and Oshman, 2009; Lenzen et al., 2009a). Bu bağlamda, tezin 6. bölümünde en küçük kareler yöntemini kullanan KAA'ndaki saat eşzamanlama protokolleri için, düğümlerin saatlerinin geriye alınmasını engelleyecek bir yöntem önerilmektedir.

KAA'nda hedef takibi gibi uygulamalar için, birbirine komşu olan algılayıcı düğümleri arasındaki eşzamanlama hatasının düşük tutulması çok önemlidir. Hareketli bir nesnenin hızını metre/saniye cinsinden merkezi bir istasyona göndermek için, algılayıcı düğümleri Şekil 1.3'te görüldüğü gibi birbirlerine d metre aralıklarla yerleştirilmiş olsun. Herhangi bir algılayıcı düğümü nesneyi algıladığı anda, bu olayın tespit anını kaydet sin ve aynı zamanda kendisine komşu olan diğer düğümlerle de nesnenin tespit anını de ğiş tokuş etsin. Düğüm, nesnenin hızını tahmin edebilmek için kendi komşularının tespit etme zamanı ile kendi tespit etme zamanı arasındaki fark olan Δ de ğerini kullanarak $v = d/\Delta$ de ğerini hesaplamalıdır. Δ de ğerinin saniye cinsinden geçen ger çek zaman olarak hesaplanabilmesi için, düğümlerin Eşgüdümlü Evrensel Zaman (EEZ)⁵ gibi bir dışsal kayna ğa eşzamanlanmış olmaları gerekmektedir. Birbirine komşu düğümler arasında eşzamanlama hatası küçük olursa, Δ de ğeri ve bunun sonucunda nesnenin hızı çok daha do ğru tahminlenebilmektedir. Literatürde, bahsedilen uygulama gibi hem dışsal eşzamanlamaya hem de komşular arasındaki eşzamanlama hatasının eniyilemesine ihtiyaç duyan uygulamalar tarafından kullanılabilir bir saat eşzamanlama protokolü bulunmamaktadır. Gradient Time Synchronization

⁵İngilizcesi: *Coordinated Universal Time (UTC)*

Protocol (GTSP) (Sommer and Wattenhofer, 2009) protokolü, birbirine komşu düğümler arasında eşzamanlama hatalarını eniyilemeyi hedefleyen literatürdeki tek protokoldür. Ancak GTSP, düğümleri bir dayanak düğüme eşzamanlayamadığı için dışsal bir saat kaynağına eşzamanlanmayı da sağlayamaz. Bu bağlamda, tezin 7. bölümünde, algılayıcı düğümlerini bir dayanak düğüme eşzamanlarken birbirine komşu düğümler arasındaki saat farkını da eniyileyen yeni bir yöntem tanıtılmaktadır.

KAA'na yönelik saat eşzamanlaması dışında, tez kapsamında genel dağıtık sistemlerdeki saat eşzamanlaması konusuna da bir katkı yapılmıştır. Literatürde, düğümler arasındaki eşzamanlama hatasının düğümler arasındaki uzaklığın bir fonksiyonu olduğu *meyilli saat eşzamanlamasına* yönelik birçok çalışma vardır (Fan et al., 2005; Locher and Wattenhofer, 2006; Lenzen et al., 2008; Pussente and Barbosa, 2009; Kuhn and Oshman, 2009; Lenzen et al., 2010; Kuhn et al., 2009; Kuhn et al., 2010). Bu çalışmalar arasında, birbirine komşu düğümler arasında görülebilen eşzamanlama hatası açısından optimal olan algoritmalar öneren çalışmalar bulunmaktadır (Kuhn and Oshman, 2009; Lenzen et al., 2010). Ancak bu çalışmalarda önerilen optimal algoritmalar, uygulamada gerçekleştirilmeleri zor olan ifadeler içermektedirler. Bu bağlamda, tezin 8. bölümünde, literatürdeki algoritmalar göz önüne alınmış ve bu algoritmalarda yer alan uygulamada gerçekleştirimi zor olan ifadeler yerine, algoritmaların optimallik özelliklerini bozmayacak daha kolay gerçekleştirilebilir ifadeler sağlanmıştır. Buna ek olarak, bu algoritmaların yürüttüğü görevler ayrı ve gerçekleştirilebilir modüllere ayrılarak yeni bir meyilli saat eşzamanlama sistemi tanıtılmıştır.

1.5 Teze Genel Bakış

Bu tez sırasıyla şu bölümlerden oluşmaktadır: 2. bölümde KAA'nda saat eşzamanlamasının kendine özgü niteliklerinden bahsedilmekte ve kısa bir literatür taramasına yer verilmektedir. 3. bölümde KAA'nda saat eşzamanlamasının analizi için kullanılacak soyut sistem modeli sunulmaktadır. 4. bölümde algılayıcı düğümlerinin birbirlerinin saatleri arasında doğrusal bir ilişkiyi nasıl kurdukları ortaya koyulmakta ve analiz edilmektedir. 5. bölüm, öncelikle yavaş sel yöntemini kullanan FTSP protokolünün analizini içermektedir. Bu bölüm içerisinde yavaş sel yönteminin saat eşzamanlaması üzerindeki olumsuz etkisini azaltacak yöntemler sunulmaktadır. En küçük kareler yöntemini uygulayan protokollerde düğümlerin saatlerinin geri alınmasını engelleyecek bir yöntem 6. bölümde tanıtılmaktadır. 7. bölümde algılayıcı düğümlerini bir dayanak düğüme eşzamanlarken birbirine komşu düğümler arasındaki saat farkını da eniyileyen yeni

bir yöntem tanıtılmaktadır. 8. bölümde, uygulamada gerçekleştirimi daha kolay olacak yeni bir optimal meyilli saat eşzamanlaması sistemi tanıtılmaktadır. 9. bölüm ise genel sonuçlara, tartışmaya ve gelecek çalışmalara ayrılmıştır.

“Zaman, herşeyin bir anda olmasını engelleyen şeydir.”

John Archibald WHEELER

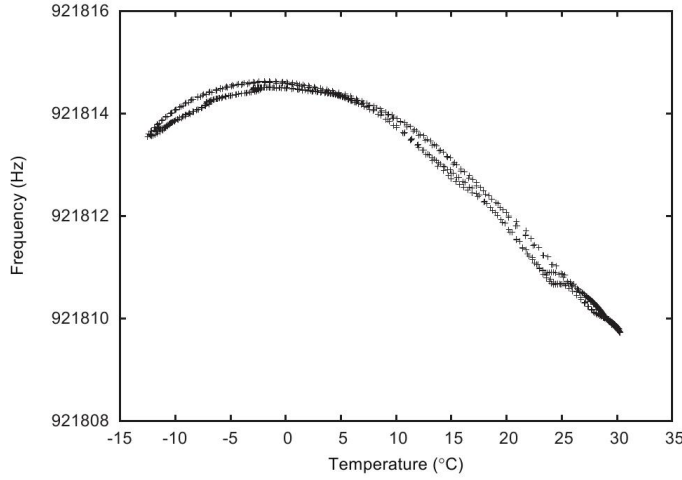
2 KABLOSUZ ALGILAYICI AĞLARINDA SAAT EŞZAMANLAMASINA GİRİŞ

KAA için geliştirilmiş eşzamanlama protokollerinin temel amacı, düğümlerin saatlerini uygulamaların istediği hassasiyette eşzamanlamaktır. Bu protokoller düğümlerin saatlerini eşzamanlarken iletişim sıklığını, iletişimde iletilen bit sayısını ve işleme süresini düşük tutarak eşzamanlama için harcanan enerji miktarını ve kaynak kullanımını eniyilemeye çalışırlar. Saat eşzamanlaması, saat bilgilerinin tümegönderim ile düğümler arasında değiş tokuşuna dayandığı için iletişim hattındaki belirsizlikler eşzamanlama hatasını doğrudan etkilemektedir. Tüm bunlara ek olarak, düğümlerin eşzamanlanmış saatleri saat sapmaları nedeniyle zaman içerisinde bozulmaktadır.

2.1 Algılayıcı Düğümlerindeki Donanımsal Saatler

Algılayıcı düğümlerinin donanımında yer alan mikrodenetleyiciler 8, 16 ya da 32 bitlik genişliğe sahip olabilen sayaç yazmaçlarına sahiptirler. Sayaç yazmaçları, yine algılayıcı düğümlerinin donanımında yer alan kristal salınıcıların her bir darbesinde 1 arttırılmaktadır. Bir sayaç yazmaç taşıdığı anda, sayacın değeri 0'a eşitlenerek her salınıcı darbesinde arttırma sürecine devam edilmektedir. Sayaç yazmaçının ardışık iki artımı arasında geçen süre, onu besleyen kristal salınıcının frekansına bağlıdır. Algılayıcı düğümlerindeki kristal salınıcılar 7.37 Mhz ya da 32 kHz gibi frekanslara sahiptirler.

Bir sayaç yazmaç, değeri salt okunur bir *donanım saati* olarak düşünülebilir. Donanım saatlerinden okunan değer, algılayıcı düğümlerinin *yerel zaman* anlayışlarını oluşturmaktadır. Düğümler, donanım saatlerini kullanarak zaman aralıklarını ölçme, zamanlayıcı kurma gibi işlemleri gerçekleştirebilirler. Donanım saatleri algılayıcı düğümleri için ortak bir zaman anlayışı sağlamaz. Bunun nedeni algılayıcı düğümlerinin kristal salınıcıların frekanslarının zaman içerisinde kararsızlık göstermesidir. Sıcaklık ve basınç gibi çevresel etkenler kısa dönemli frekans kararsızlıklarına yol açabilmekteyken, kristal salınıcının eskimesi uzun dönemli frekans kararsızlıklarına yol açmaktadır (Elson et al., 2002; Sivrikaya and Yener, 2004; Akyildiz and Vuran, 2010). Şekil 2.1'de Mica2 düğümünün donanım saatinin frekansının sıcaklık ile nasıl değiştiği gösterilmektedir.



Şekil 2.1: Mica2 düğümünün saat frekansının sıcaklık ile değişimi (Lenzen et al., 2009b).

Kristal salınıcılarının frekans kararsızlıkları sonucunda algılayıcı düğümlerinin donanımsal saatleri genellikle 30 ve 100 ppm aralığında bir *sapmaya* sahiptirler.⁶ Donanım saatleri salt okunur oldukları ve sık sık saptıkları için, düğümler arasında ortak zaman kavramı oluşturmak için saat eşzamanlama algoritmalarına ve mantıksal saat kavramına ihtiyaç vardır.

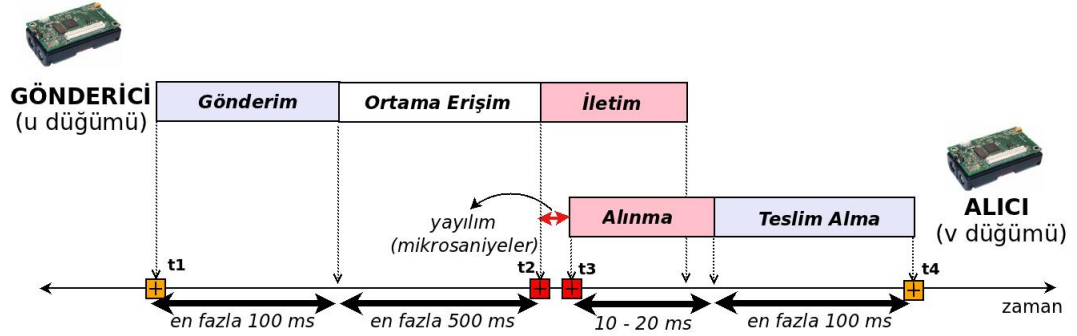
2.2 Saat Eşzamanlama Protokolleri ve Mantıksal Saat Kavramı

Algılayıcı ağlarındaki saat eşzamanlama protokolleri, düğümlerin birbirleri ile iletişim kurarak zaman bilgilerini değiş tokuş etmelerine ve bunun sonucunda ortak bir zaman kavramında anlaşmalarına dayanmaktadır. Saat eşzamanlama protokolleri, donanım saatlerinin değerini değiştiremedikleri için, ortak zaman anlayışını *mantıksal saatler* (yazılımsal saatler) üzerinden oluştururlar. Bir düğüm bir saat eşzamanlama protokolü işletirken, mantıksal saatini komşularından⁷ topladığı zaman bilgisini ve kendi donanımsal saatini kullanarak hesaplamaktadır.

Saat eşzamanlama protokollerinin ana amacı, herhangi bir anda düğümlerin mantıksal saatleri arasındaki farkı, yani *eşzamanlama hatasını* eniyilemektir. Herhangi bir anda ağdaki herhangi iki düğüm arasındaki en büyük mantıksal saat farkına *genel saat farkı* denilmektedir. Birbirine komşu herhangi iki düğüm arasında herhangi bir andaki en büyük mantıksal saat farkına ise *yerel saat farkı* denilmektedir. Saat eşzamanlama protokolleri genel saat farkının, yerel saat

⁶ppm: milyonda bir, 10^{-6} . Salınıcının idealde bir milyon salınım yapacağı sürede kaç ek ya da eksik salınım yaptığıdır.

⁷Bir düğümün komşuları, onun kapsama alanında olan ve doğrudan haberleşebildiği düğümler olarak tanımlanmaktadır.



Şekil 2.2: İki düğüm arasındaki iletişim sürecinde meydana gelen mesaj gecikmesinin, belirlenimci ve belirlenimci olmayan bileşenleri. Bu bileşenlerin büyüklükleri (Maróti et al., 2004) çalışmasındaki Tablo 1’den alınmıştır.

farkının ya da her ikisinin eniyilenmesini hedefleyebilirler.

2.3 Eşzamanlama Hatalarının Kaynakları

Eşzamanlama hataları, temel olarak düğümler arası iletişim süresince meydana gelen *mesaj gecikmelerinden* ve donanım saatlerinin *sapmalarından* etkilenmektedirler. Saat eşzamanlama sürecinde, bir düğüm kendi zaman bilgisini eşzamanlama paketleri aracılığıyla diğer düğümlere göndermekte ve eşzamanlama paketleri aracılığıyla diğer düğümlerin zaman bilgilerini almaktadır. Bir gönderici düğümün eşzamanlama paketini hazırlayıp göndermeye karar vermesinden alıcı düğümün bu paketi almasına kadar geçen süreye *mesaj gecikmesi* denilir. Mesaj gecikmesi aşağıdaki bileşenlerden oluşmaktadır (Maróti et al., 2004; Elson et al., 2002; Ganeriwal et al., 2003; Römer et al., 2005):

- *Gönderim Gecikmesi:* Eşzamanlama paketinin uygulama katmanından Ortama Erişim Katmanı’na (OEK)⁸ ulaşmasına kadar geçen süredir. Bu gecikme, süreçler arası geçiş gecikmesi ya da sistemin o anki yüküne bağlı olarak ortaya çıkan işletim sistemi kuyruklarındaki gecikmeler gibi yazılımsal gecikmelerden ötürü belirlenimci değildir.

- *Erişim Gecikmesi:* Fiziksel ortama erişilinceye kadar, OEK katmanında harcanan süredir. Algılayıcı düğümlerinin çalıştırdığı OEK protokolüne ve o andaki haberleşme yoğunluğuna bağlıdır. Bundan dolayı belirlenimci değildir.

- *İletim Gecikmesi:* Paketin kablosuz iletişimle bit bit fiziksel ortam üzerinden iletilmesi için geçen süredir. Bu gecikme, paket büyüklüğü ve iletim hızı değerleri

⁸İngilizcesi: Medium Access Layer.

kullanılarak tahmin edilebilir. Bundan dolayı belirlenimcidir.

- *Yayılım Gecikmesi:* Bir bitin gönderici taraftan çıkıp alıcı düğümün radyo kanalı tarafından yakalanması için geçen süredir. Birbirine sabit uzaklıkta duran düğümler için belirlenimcidir.

- *Alınma Gecikmesi:* Paketin bitlerinin alınma ve bunların OEK katmanına geçirilmesi için harcanan süredir. İletim zamanına benzer şekilde, belirlenimcidir.

- *Teslim Alma Gecikmesi:* Paketin uygulama katmanına iletilmesi için geçen süredir. Gönderim gecikmesine benzer şekilde, işletim sistemi katmanında değişkenlik gösteren gecikmeler nedeniyle belirlenimci değildir.

İki düğüm arasındaki iletişimde mesaj gecikmesinin bileşenleri ve bu bileşenlerin büyüklükleri Şekil 2.2’de gösterilmiştir. Görüldüğü gibi, bir düğüm kendi zaman bilgisini diğer düğümlere gönderirken, bu bilgi alıcı düğümler tarafından alınıncaya kadar güncelliğini yitirmektedir. Yani, *alıcı düğümler zaman bilgisini belirli bir hatayla almaktadırlar*. Mesaj gecikmesinin belirlenimci olmayan bileşenleri, algılayıcı ağlarındaki saat eşzamanlama protokollerinin eşzamanlama hatalarını doğrudan etkilemektedir.

Eşzamanlama hatasını doğrudan etkileyen bir diğer etken saat sapmalarıdır. Saat eşzamanlama protokolleri düğümlerin mantıksal saatlerini hesaplarken, onların donanım saatlerini de kullanılmaktadır. Donanım saatleri saptıkları için, *düğümlerin almış oldukları zaman bilgileri ve bunun sonucunda kendi mantıksal saatleri de zaman içerisinde sapar ve güncelliğini yitir*. Bu nedenle, genel ve yerel saat farklarının uygulamaların gereksinimleri doğrultusunda belirlenen sınırların dışına çıkmaması için, eşzamanlama işlemi periyodik olarak yapılmalıdır. Periyodun çok kısa olmaması yani iletişim yükünün ve dolayısıyla enerji tüketiminin artmaması için, saat eşzamanlama algoritmalarının saat sapmalarını telafi etmeleri gerekmektedir.

2.4 OEK Katmanında Zaman Etiketleme

Mesaj gecikmesinin bileşenlerinin etkileri, *OEK katmanındaki zaman etiketleme*⁹ yöntemi ile büyük ölçüde giderilebilmektedir (Maróti et al., 2004; Ganeriwal et al., 2003). Bu yöntemde her paketin içine, paket fiziksel ortamda iletmeye başlamadan hemen önce gönderici düğümünün o anki donanım saati

⁹İngilizcesi: *MAC layer timestamping*.

kullanılarak bir zaman etiketi yerleştirilmektedir. Benzer şekilde, alıcı düğüm paketin ön baytlarını aldığı anda, paketi kendi donanım saatini kullanarak etiketlemektedir. Bu sayede, gönderim, erişim, iletim, alınma ve teslim alma gecikmelerinin etkileri büyük ölçüde giderilmektedir. OEK katmanında zaman etiketleme yöntemini kullanarak zaman bilgisinin değiş tokuşunun detaylı analizi 3. Bölüm'de yapılacaktır.

2.5 Saat Eşzamanlama Yaklaşımları

Kablosuz algılayıcı ağlarındaki saat eşzamanlaması yaklaşımları en temel olarak aşağıdaki şekilde sınıflandırılabilir (Sundararaman et al., 2005; Römer et al., 2005):

- *İçsel veya dışsal eşzamanlama:* Dışsal eşzamanlamada düğümler Eşgüdümlü Evrensel Zaman (EEZ)¹⁰ gibi dışsal bir zaman kaynağına eşzamanlanırlar. İçsel eşzamanlamada ise düğümler dışsal bir zaman kaynağına ya da kendi saati diğer tüm düğümler için bir zaman kaynağı olacak özel bir düğüme ihtiyaç duymazlar.
- *Sürekli veya talep üzerine eşzamanlama:* Sürekli eşzamanlamada, düğümlerin saatleri ağın yaşantısı boyunca eşzamanlanmış olarak tutulmaktadır. Talep üzerine eşzamanlamada ise eşzamanlama bir istek üzerine sağlanmaktadır ve eşzamanlanmaya ihtiyaç duyulmadığı zaman dilimlerinde enerji tasarrufu sağlamaktadır. *Olay tetiklemeli* eşzamanlamada, eşzamanlama bir olay gerçekleştiği anda sağlanmaya çalışılmaktadır. *Zaman tetiklemeli* eşzamanlamada ise eşzamanlama, veri toplanması gereken sabit zaman aralıklarında yapılmaktadır.
- *Tam veya kısmi eşzamanlama:* Tam eşzamanlamada ağdaki tüm düğümler eşzamanlanırken, kısmi eşzamanlamada ağdaki düğümlerin bir alt kümesi eşzamanlanmaktadır.
- *Saatlerin eşzamanlanması veya bir zaman ölçeğine çevirilmesi:* Saatlerin eşzamanlanması tüm düğümlerin ortak bir saat değerinde anlaşmalarını sağlamaktadır. Zaman ölçeğine çevirme yönteminde ise, ağdaki tüm düğümler için ortak bir saat değeri hesaplanması yerine bir düğümün yerel zamanının diğer düğümün yerel zamanına çevirilmesi süreci gerçekleştirilir.
- *Bir dayanak düğüme eşzamanlama ya da eşten eşe eşzamanlama:* Dayanak düğüme eşzamanlama sürecinde, ağda bulunan bütün düğümlerin

¹⁰İngilizcesi: *Universal Coordinated Time (UTC)*.

saatleri önceden belirlenmiş ya da devingen olarak belirlenen bir dayanak düğümünün saatine eşzamanlanmaktadır. Eşten eşe eşzamanlamada ise, düğümler bir dayanak düğümüne ihtiyaç duymadan birbirleri ile eşzamanlanmaktadır.

- *Bir altyapı ile ya da altyapısız eşzamanlama:* Altyapıya ihtiyaç duyan saat eşzamanlamasında, öncelikle yayılım ağacı gibi bir ağaç tabanlı altyapı oluşturulmaktadır. Düğümler bu ağaç altyapısında yer alan üst seviyedeki düğümlerle eşzamanlanmaktadır. Düğüm arızaları bu altyapının yeniden oluşturulmasını gerektirmektedir. Altyapısız eşzamanlamada ise, düğümler arasındaki eşzamanlama için herhangi bir altyapıya ve dolayısıyla bu altyapının bakımına ihtiyaç yoktur.
- *Genel saat farkı ya da yerel saat farkının eniyilenmesi:* Eşzamanlama sürecinde, ağ içerisindeki herhangi iki düğümün arasında görülebilecek en büyük saat farkının eniyilenmesi hedeflenebilir. Bir başka hedef ise birbirine komşu olan düğümler arasında görülebilecek en büyük saat farkını eniyilemektir.

2.6 Temel Başarım Ölçütleri

KAA'ndaki saat eşzamanlama protokollerini değerlendirirken kullanılan temel başarım ölçütleri aşağıdaki gibi sıralanabilir (Karl and Willig, 2005):

- *Eşzamanlama Hassasiyeti:* Bu ölçüt eşzamanlama hatasını ifade eder. *Mutlak hassasiyet*, düğümün mantıksal saatinin dışsal standart bir zaman kaynağından, örneğin EEZ'den, ne kadar farklılık gösterdiğidir. *Göreceli hassasiyet*, düğümlerin mantıksal saatlerinin birbirinden ne kadar farklılık gösterdiğidir.
- *Enerji Verimliliği:* Bu ölçüt, eşzamanlama turlarında kaç adet paketin değiş tokuş edildiği, paketlerin işleme maliyeti ve paket değiş tokuş sıklığını ifade etmektedir. İletişim maliyeti işleme maliyetinden daha büyük olduğu için, enerji verimliliğinin temel ölçütü iletişim yüküdür.
- *Karmaşıklık:* İşlemci yükünü, bellek gereksinimini, algoritmik karmaşıklık ve mesaj karmaşıklığını bu ölçüt ifade etmektedir. Örneğin kayan noktalı sayılar üzerinde cebirsel işlem yapan algoritmalar daha yüksek işleme yüküne sahiptir. Mesaj karmaşıklığı aynı zamanda enerji verimliliğini de etkilemektedir.

- *Hata Duyarlılığı:* Protokolün düğüm ve iletim hatalarına dayanıklı olup olmadığını ifade eder. Kablosuz iletişim güvenli değildir ve iletilecek verinin alıcı düğümlere ulaşacağını garantiye yoktur. Çevresel etkenler düğümlerin çalışmasını ve iletişimi çok sık etkileyebilir. Eşzamanlama protokolleri hatalara karşı duyarlı olmalı ve devingen şartlara uyum sağlayabilmelidir.
- *Ölçeklenebilirlik:* Algılayıcı ağındaki düğüm sayısı arttıkça, eşzamanlama protokolünün başarımının nasıl değiştiğini ifade eder. Algılayıcı düğümleri ucuzladıkça, ağıdaki algılayıcı sayısı binleri bulabilir. Protokol, küçük ağlarda olduğu gibi büyük ağlarda da istenilen hassasiyette ve verimlilikte çalışabilmelidir.

2.7 Kablosuz Algılayıcı Ağları İçin Tasarlanmış Literatürdeki Temel Protokoller

Bu bölümde literatürdeki temel protokoller olan *Lightweight Tree Based Synchronization*, *Reference Broadcast Synchronization*, *Timing-Sync Protocol for Sensor Networks*, *Flooding Time-Synchronization Protocol*, *Routing Integrated Time Synchronization Protocol*, *Rapid Time Synchronization*, *Temperature Compensated Time Synchronization*, *Gradient Time Synchronization Protocol* ve *Reachback Firefly Algorithm* protokolleri özetlenecektir.

2.7.1 LTS protokolü

LTS (*Lightweight Tree Based Synchronization*) (van Greunen and Rabaey, 2003) çok sıkı eşzamanlamaya ihtiyaç duymayan uygulamalar için tasarlanmıştır. LTS, düğümlerin saatlerini önceden belirlenmiş dayanak düğümlerinin saatlerine eşzamanlamaktadır. Bu protokolün temel yapıtaşı, 2 adet mesaj değiş tokuşuna ihtiyaç duyan iki düğüm arasındaki *ikişerli eşzamanlama* adımlarıdır. İkişerli eşzamanlama sürecinde, bir düğüm eşzamanlanmak istediği diğer düğüme bir eşzamanlama paketi göndererek ve ondan bir eşzamanlama paketi alarak kendi mantıksal saati ile o düğümün mantıksal saati arasındaki farkı hesaplamaktadır.

Algılayıcı ağının çok sekmeli eşzamanlama sayesinde tamamen eşzamanlanması merkezi ya da dağıtık olabilir. Merkezi yaklaşımda ağ genelinde eşzamanlama, en az derinliğe sahip bir *yayılım ağacının* kenarları boyunca ikişerli eşzamanlama adımları uygulanarak sağlanır. Yayılım ağacının kökü olan ve KKS donanımına sahip bir düğümün tetikleme ile eşzamanlama başlamaktadır.

Öncelikle kök düğüm, çocukları ile eşzamanlanır ve bu süreç özyinelemeli olarak yaprak düğümlere kadar devam ettirilir. Dağıtık yaklaşımda ise yayılım ağacı kullanılmaz. Düğümler periyodik olarak, dayanak düğümüne en yakın komşu düğümleri ile ikişerli şekilde eşzamanlanırlar.

LTS'nin temel problemi eşzamanlama hatasının büyük olmasıdır. Diğer bir eksiklik, merkezi yaklaşım yayılım ağacı kullandığı için, kök düğümü bozulduğu anda ağ genelindeki eşzamanlama sürecinin bundan olumsuz yönde etkilenmesidir. LTS ile ilgili detaylı bilgi ve analizler (Karl and Willig, 2005; van Greunen and Rabaey, 2003) çalışmalarında bulunabilir.

2.7.2 RBS protokolü

RBS (*Reference Broadcast Synchronization*) (Elson et al., 2002) iki aşamadan oluşan bir protokoldür. İlk aşamada, bir dayanak tümegönderici düğümün kapsama alanı içinde olan düğümler birbirlerinin saatlerine eşzamanlanmaktadır. Bu süreci iki düğüm arasındaki bir senaryo ile açıklayacak olursak, bir dayanak düğümü kendi kapsama alanındaki iki düğüm için bir eşzamanlama paketini tümegönderir. Alıcı düğümler, bu paketi aldıkları anda kendi saat değerlerini kaydederler ve daha sonra kaydettikleri saat değerlerini birbirleri ile değiş tokuş ederler. Bir düğüm, her komşusu için bir tablo tutmakta ve bu tablo yardımı ile kendi saatini komşusunun saatine dönüştürebilmektedir. RBS'nin temel fikri, alıcı düğümlerin dayanak paketlerini aynı anda aldıklarını varsayması ve bu sayede mesaj yolundaki gecikmelerin yarattığı belirsizlikleri en aza indirmesidir. RBS'de eşzamanlama hatası sadece dayanak paketin yayılım gecikmesinden ve alıcı düğümlerdeki teslim alma gecikmelerinden etkilenmektedir. Düğümler, doğrusal bağlanım yöntemini kullanarak saat sapmalarını telafi etmektedirler.

RBS'nin ikinci bileşeni, ağ genelinde çok sekmeli eşzamanlama sağlama mekanizmasıdır. Bunun için birçok dayanak tümegönderici düğümün bulunması gerekmektedir. Farklı dayanak tümegönderici düğümlerin kapsama alanında olan düğümler arasında, saat dönüşümü yapılmaktadır. Dönüşüm işleminin verimli bir şekilde yapılabilmesi için, algılayıcı ağının kümelere ayrılmış olması gerekmektedir.

RBS çok sekmeli ağlar için tasarlanmamıştır ve oldukça yüksek bir mesaj karmaşıklığına sahiptir. Tek bir dayanak tümegönderici düğümün kapsama alanında bulunan n düğüm için $O(n^2)$ mesaj değişimi gerekmektedir. RBS için daha detaylı analiz ve açıklamalar (Elson et al., 2002; Karl and Willig, 2005) çalışmalarında

bulunabilir.

2.7.3 TPSN protokolü

TPSN (*Timing-Sync Protocol for Sensor Networks*) (Ganeriwal et al., 2003), LTS gibi sıradüzensel bir altyapıya ihtiyaç duyan bir protokoldür. LTS'ye benzer olarak ikili mesaj değiş tokuşuna dayanmaktadır. LTS'den farklı olarak TPSN eşzamanlama paketlerine OEK katmanında zaman etiketi vermekte ve mesaj yolundaki belirlenimci olmayan gecikmelerin etkilerini en aza indirmektedir.

TPSN'de bir seviye tespit protokolü işletilerek yayılım ağacı oluşturulmaktadır. Kök düğümüne 0 seviyesi atanmıştır. Kök düğümü kendi kimliğini ve seviyesini barındıran bir *seviye tespit* paketini tümegöndererek seviye tespit protokolünü çalıştırmaya başlar. Kök düğümünün kapsama alanında olan düğümler bu paketi alınca pakette taşınan seviye değerini bir artırarak kendi seviyelerini tespit etmiş olurlar. Kendi seviyelerini tespit etmiş olan düğümler de kök düğümü gibi seviye tespit paketlerini kendi çocukları için tümegöndermektedirler. Bu süreç ağdaki tüm düğümler kendi seviyelerini tespit edinceye kadar devam eder.

Sıradüzensel altyapı oluşturulduktan sonra, kök düğümü ağ genelinde eşzamanlamayı sağlamak için bir eşzamanlama paketini tümegönderir. 1. seviyedeki düğümler kök düğümüne eşzamanlanıp ondan onay aldıktan sonra, bu sefer kendileri 2. seviyedeki düğümlerin eşzamanlama sürecini tetiklerler. Bu süreç bütün düğümler eşzamanlanıncaya kadar devam eder. n seviyesindeki düğümler, $n - 1$ seviyesindeki düğümler $n - 2$ seviyesindeki düğümlere eşzamanlanmadan, $n - 1$ seviyesindeki düğümlere eşzamanlanamazlar.

TPSN'nin en temel eksikliği, sıradüzensel bir altyapıya gereksinim duyduğu için, düğümlerin bozulmasından olumsuz bir şekilde etkilenmesidir. TPSN için detaylı analiz ve açıklamalar (Karl and Willig, 2005; Ganeriwal et al., 2003) çalışmalarında bulunabilir.

2.7.4 FTSP protokolü

FTSP (*Flooding Time-Synchronization Protocol*) (Maróti et al., 2004) ağ genelinde eşzamanlama sağlayan ve paketlerin sel metodu ile yayılmasına dayanan bir protokoldür. FTSP'de yayılım ağacı gibi sıradüzensel bir altyapı yerine,

sel metodu ile tasarsız oluşan bir ağaç meydana gelmektedir. Bu sayede, diğer protokollerde olan yayılım ağacı oluşturma gibi aşamalara gerek duyulmaz. Böylelikle devingen ilinge değişikliklerine ve düğüm hatalarına karşı dayanıklılık sağlanmış olur.

FTSP’de en küçük kimlik numarasına sahip olan düğüm, kök düğümü olarak seçilir ve diğer düğümler için bir zaman dayanağı görevini yerine getirmektedir. Eğer kök düğümü bozulursa, ağ içindeki diğer düğümler arasından en küçük kimliğe sahip olanı yeni kök düğümü olarak seçilir ve eşzamanlama süreci kesintiye uğramadan devam eder. Kök düğümü periyodik olarak tüm düğümlerin üzerinde anlaşacağı ortak zamanı ağa sel metodu ile yayar. Kök düğümün kapsama alanında olan düğümler bu düğümünden her yeni paket aldıklarında, paket içerisindeki zaman bilgisini ve kendi donanım saatlerini kullanarak bir doğrusal bağlanım doğrusu hesaplamaktadırlar. Düğümler bu doğruyu kullanarak ortak zaman değerini tahmin edebilmektedirler. Düğümler kendi komşularının da eşzamanlanması için, kendi tahmini ortak zaman değerini tümegönderirler.

FTSP, TPSN gibi OEK katmanındaki zaman etiketleme yöntemi kullanmaktadır. FTSP, KAA’ndaki genelgeçer protokoldür ve literatürdeki birçok çalışma tarafından (Sommer and Wattenhofer, 2009; Lenzen et al., 2009b; Schmid et al., 2009; Schmid et al., 2010a; Schmid et al., 2010b) başarımlı karşılaştırılmasında kullanılmıştır. 5. Bölüm’de FTSP protokolünün problemleri gösterilecek ve analizi detaylı bir şekilde yapılacaktır.

2.7.5 PulseSync protokolü

PulseSync protokolü (Lenzen et al., 2009b), sel metodunu kullanarak zaman bilgisinin algılayıcı ağında hızlı bir şekilde yayılmasını amaçlayan bir protokoldür. Düğümler, FTSP’ye benzer şekilde kendi donanımsal saatleri ve kök düğümünün saati arasında doğrusal bağlanım ile doğrusal bir ilişki kurarlar. FTSP protokolünde düğümler yeni bir zaman bilgisi aldıklarında, bu bilgiyi kendi komşularına hemen göndermemekte ve bir süre beklemektedirler. Bekleme süreleri, doğrusal bağlanım yöntemi ile yapılan tahminlerin hatalarının büyümesine yol açmaktadır. Bunun sonucunda, düğümlerin komşularına gönderdikleri zaman bilgisi büyük hatalara sahip olabilmektedir. Her sekmede büyüyen hatalar, FTSP’nin eşzamanlama hatasının ağın büyümesinden oldukça olumsuz etkilenmesine yol açmaktadır. PulseSync protokolünde ise düğümler yeni bir zaman bilgisi alır almaz bu bilgiyi tümegönderim ile komşularına göndermektedirler. Bunun sonucunda, zaman bilgisi ağda çok daha düşük bir hatayla yayılmaktadır.

Deneyisel sonuçlar ve yapılan analizler doğrultusunda, FTSP ile kıyaslandığında PulseSync protokolünün eşzamanlama hatası açısından başarımının oldukça üstün olduğu gösterilmiştir. Ancak PulseSync protokolünün uyguladığı zaman bilgisinin hızlı yayılımı yöntemi, düğümlerin kendi komşuları iletilimlerini bitirmeden zaman bilgisini iletilemeyeceklerinden dolayı, KAA’nda oldukça yavaş olabilmektedir (Lu and Whitehouse, 2009; Schmid et al., 2010a). PulseSync için detaylı analizler (Lenzen et al., 2009b; Lenzen, 2011) çalışmalarında bulunabilir.

2.7.6 RITS protokolü

RITS (*Routing Integrated Time Synchronization Protocol*) (Sallai et al., 2006) protokolünde, zaman ölçeğine dönüşüm yaklaşımı uygulanmaktadır. Bir olayı tespit eden düğüm, merkezi bir düğüme ulaştırmak istediğı bir paketi, olayın olduğu andaki kendi donanım saatinin değeri ile zaman etiketler. Paketin zaman etiketinde yer alan zaman bilgisi, kaynak düğümden merkezi düğüme iletilirken her sekmede alıcı düğümün kendi donanım saat değeri dönüştürölmektedir. Her sekmedeki dönüşümler sonucunda merkezi düğümün eline geçen zaman etiketi merkezi düğümün donanım saati cinsinden olmaktadır. Doğrusal bağlanım, saat sapmalarını telafi etmek için kullanılmaktadır. RITS protokolü zaman ölçeğine dönüşüm yöntemini kullandığı için, düğümlerin her an erişebileceğı bir ortak zaman kavramı sağlayamaz. RITS için detaylı analizler (Sallai et al., 2006; Schmid et al., 2010a) çalışmalarında bulunabilir.

2.7.7 RATS protokolü

RATS (*Rapid Time Synchronization*) (Kusy et al., 2006) protokolünde, önceden belirlenmiş bir dayanak düğümü ağa kendi donanım saat değeri sel yöntemi ile yaymaktadır. Her düğüm eline geçen dayanak düğümünün saat bilgisini kendi donanım saat değeri dönüştüröürerek, dayanak düğümünün donanım saati ile kendi donanım saati arasındaki farkı hesaplar. Doğrusal bağlanım yöntemi saat sapmalarını telafi etmek için kullanılmaktadır. RATS protokolü, dayanak düğümü herhangi bir arızaya uğradığında ağ genelinde eşzamanlama sağlayamamaktadır. RATS için detaylı analizler (Kusy et al., 2006) çalışmasında bulunabilir.

2.7.8 TCTS protokolü

TCTS (*Temperature Compensated Time Synchronization*) (Schmid et al., 2009) algılayıcı düğümleri üzerindeki sıcaklık algılayıcısını kullanarak düğümler arasındaki eşzamanlama periyodunun devingen bir şekilde ayarlanmasını sağlamaktadır. Algılayıcı düğümler üzerinde yer alan kristal salıncıklar, sıcaklık değişimlerinden büyük oranda etkilendikleri için, bu yöntem herhangi bir sıcaklık değişimi olmadığı zaman dilimlerinde daha büyük eşzamanlama periyodları ile eşzamanlanmayı sağlayarak (iletişim sıklığını azaltarak) enerjiden tasarruf edilmesini sağlamaktadır. Düğümler, doğrusal bağlanım yöntemini uygulayarak bir dayanak düğümünün saati ile kendi donanım saatleri arasında doğrusal bir ilişki kurmaktadır. TCTS için detaylı analizler (Schmid et al., 2009) çalışmasında bulunabilir.

2.7.9 GTSP protokolü

Şu ana kadar değinilen protokoller ağdaki herhangi iki düğüm arasındaki eşzamanlama hatalarını küçültmeyi hedefleyen protokollerdi. Fan ve Lynch (Fan and Lynch, 2006) algılayıcı ağlarındaki bazı uygulamaların komşu düğümlerin daha iyi bir şekilde eşzamanlanmış olmasına ihtiyaç duyduğunu belirtmişlerdir. Ağdaki birbirine yakın düğümlerin eşzamanlama hatasını azaltmanın önemine değinirken birbirine uzak düğümlerin arasında daha büyük bir eşzamanlama hatasına izin verilebileceğini ortaya koymuşlardır. Ağdaki düğümler arasındaki eşzamanlama hatasının, düğümler arasındaki uzaklığın bir fonksiyonu olmasına gerek duyan *meyil özelliğini* tanıtmışlardır.

FTSP ve PulseSync gibi sel metodu ile tasarsız bir ağaç ilingesi meydana gelen protokollerde, bu ağacın farklı dallarında bulunan ama birbirine komşu olan düğümler birbirleri ile kötü bir şekilde eşzamanlanmaktadır. GTSP (*Gradient Time Synchronization Protocol*) (Sommer and Wattenhofer, 2009) protokolünün temel amacı birbirine komşu düğümler arasındaki eşzamanlama hatasının eniyilenmesidir. GTSP, kök düğümüne veya sıradüzensel bir altyapıya ihtiyaç duymamaktadır. GTSP’de her düğüm periyodik olarak kendi zaman bilgisini tümegönderir. Komşularından aldıkları zaman bilgilerini basit bir ortalama alma algoritması ile işleyen düğümler, ortak bir zaman değerinde anlaşılmaktadırlar. Başka bir deyişle, her düğüm bir kök düğüm yerine kendi komşularına eşzamanlanmaktadır. GTSP’nin problemleri ve analizi, 7. Bölüm’de ortaya koyulacaktır.

Çizelge 2.1: Özetlenen eşzamanlama protokollerinin saat eşzamanlama yaklaşımlarına genel bakış.

	LTS	RBS	TPSN	FTSP	PulseSync	RITS	RATS	TCTS	GTSP	RFA
Dışsal	•	•	•							
İçsel		•		•	•	•	•	•	•	•
Sürekli			•	•	•		•	•	•	•
Talep Üzerine	•	•				•				
Tam	•		•	•	•	•	•	•	•	•
Kısmi	•	•								
Saat	•		•	•	•		•	•	•	
Ölçek dönüşümü		•				•	•			
Dayanak Düzümü	•		•	•	•		•	•		
Eşten Eşe		•							•	•
Altyapılı	•	•	•							
Altyapısız				•	•	•	•	•	•	•
Genel Eniyileme	•		•	•	•	•	•	•		
Yerel Eniyileme		•							•	•

2.7.10 RFA protokolü

RFA (*Reachback Firefly Algorithm*) (Werner-Allen et al., 2005) protokolü, ateşböceklerinin anlık eşzamanlanmalarından esinlenilerek tasarlanmıştır. Bu protokolde her düğüm periyodik olarak bir mesaj tümegöndermekte ve komşu düğümlerden aldığı mesajları göz önüne alarak kendi mesaj gönderme anını ayarlamaktadır. Bu protokolde algılayıcı düğümler ortak bir saat değerinde değil de, ortak mesaj gönderme anında uzlaşmaktadırlar. Başka bir deyişle, algılayıcı düğümler ortak bir saat değeri hesaplamamaktadırlar. Bu durum, ağ genelinde tüm zamanlarda ortak bir saat değerine erişime gereksinim duyan uygulamaların RFA protokolünü kullanamamalarına yol açmaktadır. RFA için detaylı analizler (Werner-Allen et al., 2005) çalışmasında bulunabilir.

Özetlenen temel prokollerin, saat eşzamanlama yaklaşımlarına genel bakış, Çizelge 2.1’de gösterilmektedir.

2.8 Sonuçlar

KAA’nda saat eşzamanlaması, geleneksel dağıtık sistemlerden farklılıklar göstermektedir. Enerji tüketimi, hata duyarlılığı, kaynak kullanımı ve devingen ağ ilingeleri gibi geleneksel dağıtık sistemler tarafından önemsenmeyen kavramlar, KAA’ndaki saat eşzamanlaması probleminin kalbini oluşturmaktadırlar. Kısıtlı depolama ve enerji kaynaklarına sahip, işleme kapasitesi düşük, çevresel

řartların devingenliđinden dođrudan etkilenen küçük algılayıcı düđümleri için saat eşzamanlaması, hala birçok açık problemin olduđu özgün bir konudur.

“Durmuş saat bile günde iki defa doğruyu gösterir.”

Türk Atasözü

3 KABLOSUZ ALGILAYICI AĞLARINDA SAAT EŞZAMANLAMASI PROBLEMİ İÇİN SİSTEM MODELİ

Bu bölümde, ilerleyen bölümlerde sunulacak saat eşzamanlama protokollerinin analizi için kullanılacak soyut sistem modeli ortaya koyulmaktadır. Sistem modeli, bir algılayıcı ağının, ağda yer alan düğümlerin saatlerinin ve düğümler arası haberleşmenin 2. Bölüm’de belirtilen kısıtlar göz önüne alınarak matematiksel modelini temsil etmektedir.

3.1 Algılayıcı Ağının Matematiksel Modeli

Bir KAA, n adet algılayıcı düğümünün $V = \{1, \dots, n\}$ köşe kümesi tarafından temsil edildiği bir $G = (V, E)$ çizgesi olarak modellenmektedir. Düğümler \mathbb{R}^2 öklid düzleminde yer almakta ve kablosuz tümegönderim ile haberleşmektedirler. Düğümler arasındaki çift yönlü iletişim hatları $E \subseteq V \times V$ kenar kümesi ile temsil edilmektedir. Herhangi bir u düğümü sadece kendi kapsama alanındaki, yani G çizgesinde doğrudan bağlı olduğu, düğümlerle haberleşebilmektedir. Bu düğümler u düğümünün *komşuları* olarak adlandırılmakta ve $N_u = \{v \in V \mid \{u, v\} \in E\}$ ile temsil edilmektedirler.

Herhangi iki $u, v \in V$ düğümü arasındaki *uzaklık*, bu iki düğüm arasında G çizgesi üzerinde yer alan en kısa yoldaki toplam kenar sayısı olarak tanımlanmakta ve $d(u, v)$ ile gösterilmektedir. Ağın *çapı* herhangi iki düğüm arasındaki en büyük uzaklık olarak tanımlanmakta ve $\mathcal{D} = \max_{u, v \in V} d(u, v)$ ile temsil edilmektedir.

3.2 Algılayıcı Düğümlerinin Saatlerinin Matematiksel Modeli

Sistemdeki her u düğümü salt okunur bir $H_u()$ *donanım saatine* sahiptir ve bu saatin t gerçek zamanındaki değeri

$$H_u(t) = \int_0^t h_u(\tau) d\tau \quad (3.1)$$

ile tanımlanmaktadır. Burada $h_u(\tau)$, u düğümünün donanım saatinin τ gerçek zamanındaki hızını ifade etmektedir. Donanımsal saatlerin, 3.2 eşitsizliğinde gösterildiği gibi, sınırlı bir sapmaya sahip olduğu varsayılacaktır:

$$\forall t : 1 - \varepsilon \leq h_u(t) \leq 1 + \varepsilon, 0 < \varepsilon \ll 1 \quad (3.2)$$

Donanım saatleri saparak birbirlerinden uzaklaştıkları ve salt okunur oldukları için, düğümler eşzamanlanmış bir zaman kavramını donanım saatlerini kullanarak elde edemezler. Bu nedenle, her u düğümü donanım saatine ek olarak yazılımsal bir fonksiyon olan $L_u()$ *mantıksal saatine* de sahiptir. Mantıksal saatin t anındaki değeri, donanımsal saat kullanılarak, 3.3 eşitliğinde gösterildiği gibi hesaplanmaktadır:

$$L_u(t) = \int_0^t l_u(\tau) h_u(\tau) d\tau + \theta_u(t). \quad (3.3)$$

Yukarıdaki eşitlikte, $l_u(t)$ mantıksal saatin t anındaki *hız çarpanı* ve $\theta_u(t)$ mantıksal saatin t anındaki *öteleme değerini* temsil etmektedir. Hız çarpanı mantıksal saatin donanım saatine göre ilerleme hızının ayarlanması için kullanılmaktadır. Mantıksal ve donanımsal saatlerin monoton artan fonksiyonlar olduğu varsayılmaktadır. Yani, bir saat eşzamanlaması algoritmasının yürütümü sırasında, mantıksal saatler hiçbir zaman geriye alınamazlar.

3.2.1 Saat eşzamanlama algoritmaları ve eşzamanlama hatası

Bir \mathcal{A} eşzamanlama algoritmasının *bağlı* G çizgesinde *yürütümü*, ağ içerisindeki tüm düğümlerin donanım saatlerinin ve düğümler arasındaki iletişim süresince meydana gelen tüm mesaj gecikmelerinin zamanın her noktasındaki değerini belirtmektedir (Lenzen et al., 2010). Yürütüm boyunca algoritmanın amacı düğümlerin mantıksal saat değerleri arasındaki farkın en aza indirilmesidir. *Genel*

saat farkı, \mathcal{A} algoritması bağlı G çizgesi üzerinde yürütüldüğü zaman herhangi iki düğüm arasında görülen en büyük mantıksal saat farkı iken *yerel saat farkı* birbirine komşu düğümlerin mantıksal saatleri arasındaki farkın en büyük değeridir. Bu iki kavramın matematiksel tanımı aşağıda gösterilmiştir.

Tanım 3.1. Genel Saat Farkı: Bir \mathcal{A} saat eşzamanlama algoritması G bağlı çizgesi üzerinde yürütüldüğünde elde edilen $\sup_{G,u,v \in V,t} \{|L_u(t) - L_v(t)|\}$ değeridir.

Tanım 3.2. Yerel Saat Farkı : Bir \mathcal{A} saat eşzamanlama algoritması G bağlı çizgesi üzerinde yürütüldüğünde elde edilen $\sup_{G,u,v \in N_u,t} \{|L_u(t) - L_v(t)|\}$ değeridir.

3.3 Algılayıcı Düğümleri Arasındaki İletişimin Matematiksel Modeli ve Analizi

Bu bölümde, düğümler arasındaki iletişimin sonucunda alınan zaman bilgisinin hatasını analiz edebilmek için, TinyOS (Levis et al., 2004) işletim sistemindeki *paket seviyesinde saat eşzamanlaması*¹¹ (Maroti and Sallai, 2008) servisleri kullanılarak eşzamanlama bilgisinin nasıl değiş tokuş edildiği incelenecektir. Paket seviyesinde saat eşzamanlaması servisleri, OEK katmanında zaman etiketleme mekanizmasını gerçekleştirmektedir ve (Kusy et al., 2006) çalışmasında sunulan ETA (*Elapsed Time on Arrival*) servislerine mantıksal açıdan denktirler.

3.3.1 İki düğüm arasındaki eşzamanlama sürecinde basit bir iletişim senaryosu

Herhangi bir u düğümünün kendi mantıksal saatinin değerini t_1 anında v düğümüne göndermeye karar verdiği bir senaryo göz önüne alınsın. Zaman bilgilerinin *başlık* ve *yük* parçalarına sahip eşzamanlama paketleri ile gönderildiği ve paketlerin başlığında *olayZamanı* sahasının bulunduğu varsayılınsın. Öncelikle, u düğümü t_1 anındaki mantıksal saat değeri olan $L_u(t_1)$ 'yi bir eşzamanlama paketinin yük sahası içine yerleştirir ve paketi gönderme isteğini OEK katmanına bir servis çağırımı ile iletir. OEK katmanı paket gönderme isteğini aldığında, paketin başlığındaki *olayZamanı* sahasına paket gönderilme isteğinin yapıldığı t_1 anındaki donanımsal saat değeri olan $H_u(t_1)$ değerini yazar. OEK katmanı kablolu kanala eriştiği ve paketin ilk birkaç baytını göndermeye başladığı t_2 anında, düğümün o andaki donanımsal saati $H_u(t_2)$ değerini ve paketin başlığında

¹¹İngilizcesi: *packet-level time synchronization*.

yer alan $H_u(t_1)$ değerini kullanarak $H_u(t_2) - H_u(t_1)$ değerini hesaplar. Paketin hazırlanıp OEK katmanına iletilmesinden iletişim hattının ele geçirilip paketin gönderilmeye başlanılmasına kadar geçen süreyi gönderici düğümün yerel zamanı cinsinden gösteren bu değer, gönderilmekte olan paketin *olayZamanı* sahasına yeniden yazılır.

v düğümünün paketin ilk birkaç baytını almaya başladığı t_3 anındaki donanımsal saat değeri $H_v(t_3)$ sistemde kaydedilir. Paket tam olarak alındığı andan sonraki bir t_4 anında, v düğümü iletişim katmanı servislerini sorgulayarak kendi donanımsal saatinin t_1 anındaki bir tahminini elde edebilir. Bu sorgu sonucunda iletişim katmanı, alınan eşzamanlama paketinin başlığındaki $H_u(t_2) - H_u(t_1)$ değerini kullanarak 3.4 eşitliğinde gösterilen *olayZamanı* değerini döndürür:

$$\textit{olayZamanı} = H_v(t_3) - (H_u(t_2) - H_u(t_1)). \quad (3.4)$$

v düğümü eşzamanlama paketi içerisinde taşınan $L_u(t_1)$ mantıksal saat değerini de iletişim katmanı servislerini sorgulayarak okuyabilir.

3.3.2 Analiz

t' zamanı $H_v(t') = \textit{olayZamanı}$ eşitliğini sağlayan gerçek zaman olsun. Eğer u ve v düğümlerinin donanımsal saat hızlarının $[t_1, t_4]$ zaman aralığında sabit olduğu varsayılırsa¹², $H_v(t')$ aşağıdaki gibi yazılabilir:

$$\begin{aligned} H_v(t') &= H_v(t_3) - H_u(t_2) + H_u(t_1) \\ &= H_v(t_3) - (t_2 - t_1)h_u \\ &= H_v(t_1) + (t_3 - t_1)h_v - (t_2 - t_1)h_u \\ &= H_v(t_1) + (t_3 - t_2)h_v + (t_2 - t_1)(h_v - h_u). \end{aligned} \quad (3.5)$$

3.5 eşitliğinden 3.6 eşitliği elde edilebilir:

¹²Bu varsayım oldukça gerçekçidir. Çünkü düğümlerin donanımsal saatlerinin hızları, bir paketin gönderim ve alımı için geçen zaman aralığında oldukça yavaş değişirler (Sallai et al., 2006).

$$t_1 = t' - (t_3 - t_2) - (t_2 - t_1)(h_v - h_u) \frac{1}{h_v}. \quad (3.6)$$

Yukarıdaki eşitlik, $(t_3 - t_2)$ eşzamanlama paketinin ilk birkaç baytının iletim, yayılım ve alınma gecikmelerini içermektedir. Bölüm 2.3'te belirtildiği gibi, bu bileşenlerin belirlenimci olduğu varsayılabilir. Ancak, $(t_2 - t_1)$ mesaj gecikmesinin belirlenimci olan ve belirlenimci olmayan bileşenlerinden oluşmaktadır. (Ilae Noh et al., 2008; Wu et al., 2011) çalışmasında $(t_2 - t_1)$ 'nin değişken kısmı, birçok bağımsız rastgele süreçten oluştuğu için, merkezi limit teoremine göre normal dağılışa sahip bir şans değişkeni olarak modellenmiştir. (Elson et al., 2002) çalışmasında $(t_2 - t_1)$ 'nin değişken kısmı %99.8 güven aralığıyla bir normal şans değişkeni olarak modellenebilmiştir. Benzer şekilde, $(t_2 - t_1)$ 'nin değişken kısmını bu çalışmada da normal dağılışa sahip bir şans değişkeni olarak modelleneyecektir.

$(t_2 - t_1)$ 'nin belirlenimci kısmı γ ile ve değişken kısmı beklenen değeri μ_d ve varyansı σ_d^2 olan normal dağılışa sahip bir şans değişkeni ile temsil edilsin. μ_{r_v} ve $\sigma_{r_v}^2$ değerleri aşağıdaki gibi tanımlansın:

$$\mu_{r_v} = (t_3 - t_2) + (\gamma + \mu_d)(h_v - h_u) \frac{1}{h_v}, \quad (3.7)$$

$$\sigma_{r_v}^2 = \sigma_d^2 \left((h_v - h_u) \frac{1}{h_v} \right)^2. \quad (3.8)$$

Böylelikle, 3.6 eşitliği 3.9 eşitliğinde gösterildiği gibi yazılabilir:

$$t_1 = t' - r_v \quad (3.9)$$

Burada r_v bir şans değişkenidir ve $r_v \sim \mathcal{N}(\mu_{r_v}, \sigma_{r_v}^2)$ dağılışına sahiptir. Eğer düğümlerin mantıksal saat çarpanlarının ve öteleme değerlerinin de $[t_1, t_4]$ aralığında sabit olduğu varsayılırsa, v düğümünün u düğümünden aldığı eşzamanlama paketinin içinde taşınan $L_u(t_1)$ 'in $L_u(t')$ için bir tahmin olduğu Eşitlik 3.10'de gösterilmektedir:

$$\begin{aligned}
\hat{L}_u(t') = L_u(t_1) &= L_u(t') - l_u h_u r_v \\
&= L_u(t') - e_v
\end{aligned} \tag{3.10}$$

Burada e_v şans değişkeni $e_v \sim \mathcal{N}(\mu_{e_v} = \mu_{r_v} l_u h_u, \sigma_{e_v}^2 = \sigma_{r_v}^2 l_u^2 h_u^2)$ şeklinde bir dağılışa sahiptir. Bu eşitlik, $\hat{L}_u(t')$ tahmininin sistematik ve rastgele parçalardan oluşan bir hataya sahip olduğunu gösterir. Saat eşzamanlamasının ana hedeflerinden biri mantıksal saatlerin hızlarını eşzamanlamak olduğu için, tüm $u, v \in V$ için $l_u h_u \approx l_v h_v$ varsayımı yapılabilir ve dolayısıyla $\mu_{e_u} \approx \mu_{e_v} \approx \mu$ ve $\sigma_{e_u}^2 \approx \sigma_{e_v}^2 \approx \sigma^2$ elde edilir. Bu bilgiler doğrultusunda, 3.3 olgusunda belirtilen gerçeğe ulaşılabilir:

Olgu 3.3. *v düğümü u düğümünden bir eşzamanlama paketi aldığı anda, $(H_v(t'), \hat{L}_u(t'))$ eşzamanlama noktasını elde etmektedir. $\hat{L}_u(t')$ tahmini, ortalaması μ ve varyansı σ^2 olan normal dağılışa sahip bir hata barındırmaktadır.*

“Kim sorarsa saat kaç diye,cevabım hep aynı;

O’na doğru”

Cemal SÜREYA

4 ALGILAYICI DÜĞÜMLERİNİN SAATLERİ ARASINDA DOĞRUSAL BİR İLİŞKİ KURMA

KAA için tasarlanmış birçok saat eşzamanlaması protokolü, düğümlerin donanımsal ve mantıksal saatleri arasında doğrusal bir ilişki olduğunu varsaymaktadır. *Doğrusal bağlanım* ile kurulan bu ilişki sayesinde, algılayıcı düğümleri sık iletişim kurmadan birbirlerinin donanımsal ya da mantıksal saatlerini tahmin edebilmektedirler. *En küçük kareler yöntemi*, düğümlerin donanımsal saatleri ve ağ genelindeki ortak saat arasında doğrusal bağlanım ilişkisi kurmak için yaygın olarak kullanılmaktadır. Bu yöntem, algılayıcı düğümlerin işleme kapasiteleri göz önüne alındığında oldukça kabul edilebilir bir işlemci yükü getirmektedir. Bu bölümde, en küçük kareler yönteminin KAA’ndaki saat eşzamanlaması sürecinde nasıl kullanıldığı açıklanacak ve bu yöntem analiz edilecektir.

4.1 İlgili Çalışmalar

Literatürde düğümlerin saatleri arasında doğrusal bir ilişki kurmak için doğrusal bağlanımı kullanan birçok saat eşzamanlaması protokolü bulunmaktadır (Elson et al., 2002; Maróti et al., 2004; Kusy et al., 2006; Lenzen et al., 2009b; Schmid et al., 2009; Schmid et al., 2010a). *RBS* (Elson et al., 2002) protokolünde, düğümler saat sapmalarını telafi etmek için doğrusal bağlanım kullanılmaktadırlar. *FTSP* (Maróti et al., 2004), *RATS* (Kusy et al., 2006), *PulseSync* (Lenzen et al., 2009b), *TCTS* (Schmid et al., 2009) protokollerinde ve (Schmid et al., 2010a) çalışmasında, düğümler kendi donanım saatleri ile bir kök düğümünün ağa yaydığı zaman bilgisi arasında bir ilişki kurmak için doğrusal bağlanım kullanmakta ve bu sayede kök düğümünün gelecekte ağa yayacağı zaman bilgisini tahmin edebilmektedirler.

4.2 En Küçük Kareler Yöntemi İle Doğrusal Bağlanımın Saatlerin Eşzamanlamasında Kullanılması

3.3.1 Bölümü’nde sunulan $u, v \in V$ düğümleri arasındaki saat eşzamanlaması süreci göz önüne alınsın. 3.3 Olgu’suna göre v düğümü u düğümünden t

anına bir eşzamanlama paketi aldığında, $(H_v(t'), \hat{L}_u(t'))$ ikilisini elde etmektedir. v düğümü aldığı her eşzamanlama paketinden elde ettiği birbirinden bağımsız ikilileri, bir (x, Y) ikilisi olarak N adet ikili saklayabilme kapasitesi olan *zaman etiketi tablosunda* biriktirmektedir. Algılayıcı düğümlerinin depolama kısıtlarından dolayı, yeni bir (x, Y) ikilisi toplandığında zaman etiketi tablosunda yer alan en eski ikili atılmakta ve yeni alınan ikili tabloya eklenmektedir. Böylece en güncel N ikili saklanmış olmaktadır. v düğümü sakladığı her (x_i, Y_i) ikilisini oluşturan değerler arasında $Y_i = \alpha + \beta x_i - e_v$ şeklinde bir doğrusal ilişki olduğunu varsayarak, bu ilişkiyi en küçük kareler yöntemi ile tahmin etmektedir. Bu ilişkide e_v , ortalaması μ ve varyansı σ^2 olan bir rastgele hatadır.

En küçük kareler yönteminde zaman etiketi tablosunda yer alan (x_i, Y_i) ikilileri, $\bar{x} = \sum x_i/N$ ve $\bar{Y} = \sum Y_i/N$ eşitliklerinde kullanılarak, *tahmini doğrusal bağlanım doğrusu (en küçük kareler doğrusu)* \hat{Y}_{ekk} aşağıda gösterildiği gibi hesaplanmaktadır:

$$\begin{aligned}\hat{Y}_{ekk} &= \hat{\alpha}_{ekk} + \hat{\beta}_{ekk}x \\ &= \bar{Y} + \hat{\beta}_{ekk}(x - \bar{x}).\end{aligned}\tag{4.1}$$

Bu doğrunun $\hat{\beta}_{ekk}$ ve $\hat{\alpha}_{ekk}$ parametreleri, $S_{xx} = \sum (x_i - \bar{x})^2$ ve $S_{xY} = \sum (x_i - \bar{x})(Y_i - \bar{Y})$ eşitlikleri kullanılarak aşağıda gösterildiği gibi hesaplanmaktadır:

$$\hat{\beta}_{ekk} = S_{xY}/S_{xx},\tag{4.2}$$

$$\hat{\alpha}_{ekk} = \bar{Y} - \hat{\beta}_{ekk}\bar{x}.\tag{4.3}$$

$\hat{\beta}_{ekk}$ parametresi tahmini doğrusal bağlanım doğrusunun *eğimi*, $\hat{\alpha}_{ekk}$ parametresi ise bu doğrunun *y-kesimi* ya da kısaca *kesimi* olarak adlandırılmaktadır. Bu parametreler gerçek doğrusal bağlanım doğrusunun α ve β parametrelerinin bir tahminidir. Burada $\hat{\beta}_{ekk}$ parametresi, u düğümünün mantıksal saatinin v düğümünün donanım saatine göre *göreceli hızını* belirtmektedir ve *sapma tahmininde* kullanılmaktadır. $\hat{\beta}_{ekk}$ parametresinin olasılık dağılışı 4.4'te gösterilmiştir (Ross, 2004):

$$\hat{\beta}_{ekk} \sim \mathcal{N}\left(\beta, \frac{\sigma^2}{S_{xx}}\right).\tag{4.4}$$

v düğümü hesaplamış olduğu en güncel en küçük kareler doğrusunu kullanarak gelecekteki Y değerlerini tahmin edebilir. Gelecekteki herhangi bir x' noktası için tahmin edilen ve gerçekte alınan Y değerleri arasındaki fark olan $Y_{x'} - \hat{Y}_{x'}$ şans değişkeninin dağılışı, 4.5'te gösterilmiştir (Ross, 2004):

$$Y_{x'} - \hat{Y}_{x'} \sim \mathcal{N}\left(0, \sigma^2 \left(1 + \frac{1}{N} + \frac{(x' - \bar{x})^2}{S_{xx}}\right)\right). \quad (4.5)$$

Dolayısıyla $x' = H_v(t)$ ve \bar{x} arasındaki fark arttıkça tahmin hatası artmaktadır. Yeni bir eşzamanlama mesajının alınması sonucunda yeni bir (x, Y) ikilisi alınıp en küçük kareler doğrusu yeniden hesaplanıncaya kadar, $H_v(t) - \bar{x}$ değeri büyümektedir. Yeni bir (x, Y) ikilisi zaman etiketi tablosunda yer alan en eski ikili atılarak tabloya yerleştirildikten sonra, en küçük kareler doğrusu yeniden hesaplanmakta ve bunun sonucunda $H_v(t) - \bar{x}$ değeri küçülmektedir. Bu durum, tahmin hatasının varyansı göz önüne alındığında bizi aşağıdaki olguya ulaştırmaktadır:

Olgu 4.1. *v düğümünün en güncel en küçük kareler doğrusunu hesapladığı andan, yeni bir eşzamanlama mesajı alıp yeni bir en küçük kareler doğrusu hesaplayacağı ana kadar tahmin edilen saat değerlerinin varyansı artmaktadır. Bu nedenle, zaman geçtikçe düğümün tahmin ettikleri saat değerlerinin hatası büyümektedir.*

v düğümü, çevresel gürültüler ve çarpışmalar gibi etkenlerin yol açtığı sinyal bozulmaları yüzünden, üst üste birkaç eşzamanlama paketini alamayabilir. Aynı zamanda kendi gönderdiği paketler de alıcı düğümlere ulaşmayabilir. Bu durumlar gözetilerek, $(H_v(t) - \bar{x})/S_{xx}$ değeri için bir üst sınır olduğunu varsaymak kabul edilebilir bir durumdur. Bu varsayım, ilerleyen bölümlerde yapılacak analizleri basitleştirecektir.

Varsayım 1. *En küçük kareler yöntemi kullanarak saat tahmini yapan her $v \in V$ düğümü için*

$$1 + \frac{1}{N} + \frac{(H_v(t) - \bar{x})^2}{S_{xx}} \leq \mathcal{F}_v \quad (4.6)$$

eşitliğinin sağlandığı varsayılmaktadır.

“Zaman ve sel kimseyi beklemez.”

Avusturya Atasözü

5 KABLOSUZ ALGILAYICI AĞLARINDA YAVAŞ SEL TABANLI SAAT EŞZAMANLAMASI

Bir dayanak düğümünün güncel saat bilgisini ağa periyodik olarak sel ile yayması, KAA’nda ağ genelinde saat eşzamanlaması için yaygın olarak kullanılan yöntemlerden bir tanesidir. Örneğin KAA’nda genel geçer saat eşzamanlama protokolü olan ve sel yöntemini uygulayan FTSP (Maróti et al., 2004) protokolünde, düğümler dayanak düğümünün zamanı hakkında elde etmiş oldukları bilgiyi belirli bir süre bekleyerek ağa yaymaktadırlar. Bekleme sürelerinin neden olduğu yavaş yayılımın, yani selin yavaş ilerleme hızının, FTSP protokolünün eşzamanlama başarımını ve ölçeklenebilirliğini oldukça düşürdüğü gösterilmiştir (Lenzen et al., 2009b). Selin yavaş ilerlemesi, FTSP’de dayanak düğümüne uzak düğümlerin oldukça büyük eşzamanlama hatasına sahip olabilmelerine yol açmaktadır. Literatürde, yavaş sele alternatif olarak düğümlerin dayanak düğümü hakkında elde ettikleri güncel zaman bilgisini beklemeden hızlı bir şekilde yayması, yani hızlı sel yöntemi önerilmiştir (Lenzen et al., 2009b). Ancak hızlı sel yöntemi, düğümlerin kendi komşuları iletilişimlerini bitirmeden seli ilerletemeyeceklerinden dolayı, KAA’nda oldukça yavaş olabilmektedir (Lu and Whitehouse, 2009; Schmid et al., 2010a).

Bu bölümün amacı, yavaş sel yönteminin algılayıcı ağlarındaki saat eşzamanlaması üzerindeki olumsuz etkisini selin yayılım hızını değiştirmeden azaltmaktır. Bu amaçla öncelikle öncelikle FTSP protokolü göz önüne alınmakta ve bu protokolün doğrusal bağlanım mekanizmasına bir *eğim geçmişi* eklentisi yapılmaktadır. Eğim geçmişini kullanan ve bunun sonucunda eşzamanlama hatalarının bekleme süreleri yüzünden daha az şıştığı üç yeni yöntem sunulmaktadır. Bu yöntemlere ek olarak, eğim geçmişini kullanmayan ve tahmini doğrusal bağlanım doğrusunun eğimini zaman etiketi tablosundaki en eski ve en güncel eşzamanlama noktalarını kullanarak hesaplayan yeni bir yöntem de tanıtılmaktadır. Tüm bu yöntemler FTSP protokolünün eşzamanlama başarımını iyileştirse de, dayanak düğümüne uzakta bulunan düğümlerin eşzamanlama hatasının ağın çapının üstel bir fonksiyonu olmasını engelleyememekte ve FTSP protokolünün ölçeklenme konusundaki başarımını etkili bir şekilde iyileştirememektedirler.

FTSP protokolünü iyileştirme yöntemlerini sunduktan sonra FTSP’ye

alternatif olarak, dayanak düğümünün zaman bilgisi ile beraber saat hızını da ağa yaymayı amaçlayan Hız Seli Eşzamanlama Protokolü (HSEP) sunulmaktadır. HSEP, FTSP ile kıyaslandığında bekleme sürelerinin eşzamanlama üzerindeki etkisini oldukça azaltsa da, dayanak düğümüne uzakta olan düğümlerin eşzamanlama hatası halen ağın çapının üstel bir fonksiyonudur. Bu bölümde yapılan ana katkı, FTSP ve HSEP'in temel eksiklikleri gözetilerek yavaş sel yönteminin eşzamanlama hatasının ve ölçeklenebilirliğinin düğümler arasında bir saat hızı uzlaşmasıyla büyük ölçüde iyileştirildiğinin gösterilmesidir. Bu bağlamda, Saat Hızı Uzlaşması İle Sel (SHUS) protokolünü tanıtlmakta, FTSP ve HSEP'nin aksine bu protokolde eşzamanlama hatasının ağın çapının karekökünün bir fonksiyonu şeklinde arttığı gösterilmektedir.

5.1 İlgili Çalışmalar

Literatürde, KAA'nda eşzamanlama hatasını en aza indirmeyi hedefleyen birçok saat eşzamanlama protokolü bulunmaktadır (Elson et al., 2002; van Greunen and Rabaey, 2003; Ganeriwal et al., 2003; Dai and Han, 2004; Maróti et al., 2004; Sun et al., 2006; Kusy et al., 2006; Lenzen et al., 2009b; Schmid et al., 2009; Schmid et al., 2010a). Bu çalışmalar arasında, FTSP (Maróti et al., 2004) ağdaki genel saat farkını eniyilemek için tasarlanmıştır. FTSP'de yürütüm anında devingen olarak seçilen bir dayanak düğümü, kendi saat bilgisini periyodik olarak ağa yaymaktadır. Sel ile yayılan zaman bilgisini alan her düğüm, kendi donanım saati ile dayanak düğümünün saati arasında doğrusal bir ilişkiyi en küçük kareler yöntemi ile kurar. Bu ilişki sayesinde düğümler dayanak düğümünün gelecekteki saatini tahmin edebilmektedirler. Her düğümün dayanak düğümünün saati hakkındaki tahminini periyodik olarak komşularına tümegöndermesi ile ağ genelinde eşzamanlama sağlanmış olur. Ancak bir düğüm dayanak düğümünün tahmini saat değerini yeni bir eşzamanlama mesajını alır almaz değil kendi tümegönderim periyodu geldiğinde tümegöndermektedir. Selin her düğümdeki bu bekleme sürelerinden dolayı yavaş bir şekilde ilerlemesi, en küçük kareler yönteminin tahminleme hatası her sekmede artmaktadır (Lenzen et al., 2009b). Dolayısıyla, en küçük kareler yönteminin yavaş sel nedeniyle ortaya çıkan kötü başarımının iyileştirilmesi, açık bir problem olarak gözükmektedir.

Lenzen et al. (Lenzen et al., 2009b) hızlı sel yöntemini kullanarak bekleme sürelerinin yarattığı yan etkileri ortadan kaldıran PulseSync protokolünü önermişlerdir. Bu protokol FTSP'nin başarımını oldukça arttırsa da, hızlı sel yöntemi birbirine komşu düğümler arasındaki iletişim yarışından ötürü oldukça yavaş olabilmektedir. Çünkü bir düğüm kendi komşuları iletişimlerini bitirmeden

Algoritma 5.1 FTSP'nin v düğümü için sözde kodu.

```

1: İkleme
2:  $seq_v \leftarrow 0$ 
3:  $B$  periyodu ile periyodik bir zamanlayıcı başlat
4:
5:  $\square seq_v < seq_u$  olan bir  $< L_u, seq_u >$  alındığında
6:  $(H_v, L_u)$  ikilisini sakla ve en küçük kareler doğrusunu hesapla
7:  $seq_v \leftarrow seq_u$ 
8:
9:  $\square$  Zamanlayıcı zamanışımında
10: eğer  $v = KÖK$  ise  $seq_v \leftarrow seq_v + 1$  soneğer
11:  $tümegönder < L_v, seq_v >$ 

```

seli ilerletememektedir (Lu and Whitehouse, 2009; Schmid et al., 2010a). Dolayısıyla, yavaş sel yöntemini kullanan saat eşzamanlama protokollerinin başarımını iyileştirmek açık bir problemdir.

Tüm düğümlerin ortak bir saat hızında anlaşmalarını sağlayan bir yöntem *Average TimeSynch* (ATS) (Schenato and Fiorentin, 2011) protokolü tarafından tanıtılmıştır. ATS'ye ek olarak, komşu düğümlerin eşzamanlama hatasını eniyilemeyi hedefleyen Gradient Time Synchronization Protocol (GTSP) (Sommer and Wattenhofer, 2009) protokolü de bu yöntemi kullanmaktadır. ATS ve GTSP bir dayanak düğüme ihtiyaç duymazlar: Ağdaki tüm düğümler komşularına eşzamanlanmaktadırlar. Komşulardan toplanan saat değerleri ortak bir saat hızında ve saat değerinde anlaşmak için kullanılmaktadır. Ancak, bu çalışmalar, KKS alıcısı ile EEZ gibi tutarlı bir saat kaynağına eşzamanlanmayı sağlayacak bir dayanak düğüme eşzamanlanmaya odaklanmamaktadırlar (Schmid et al., 2010a). Dolayısıyla, bu çalışmalar sel yöntemini ya da yavaş sel yönteminin yan etkilerini gidermeyi hedeflememektedirler.

5.2 Kablosuz Algılayıcı Ağlarında Zaman Bilgisinin Yavaş Yayılımı

Bu bölümde, FTSP protokolünün yürütümü incelenerek bekleme sürelerinin eşzamanlama hatasını nasıl etkilediği gösterilecektir. FTSP, en küçük kareler yöntemini yavaş sel yöntemi ile birlikte uygulamaktadır. FTSP'nin *KÖK* kimliğine sahip sabit bir dayanak düğümü içeren sözde kodu Algoritma 5.1'de gösterilmiştir. Sözde kodda sadece genel yöntem sunulmuş ve ayrıntılar gösterilmemiştir. Tüm düğümlerin dayanak düğümünün kimliğini bildikleri varsayılmıştır.

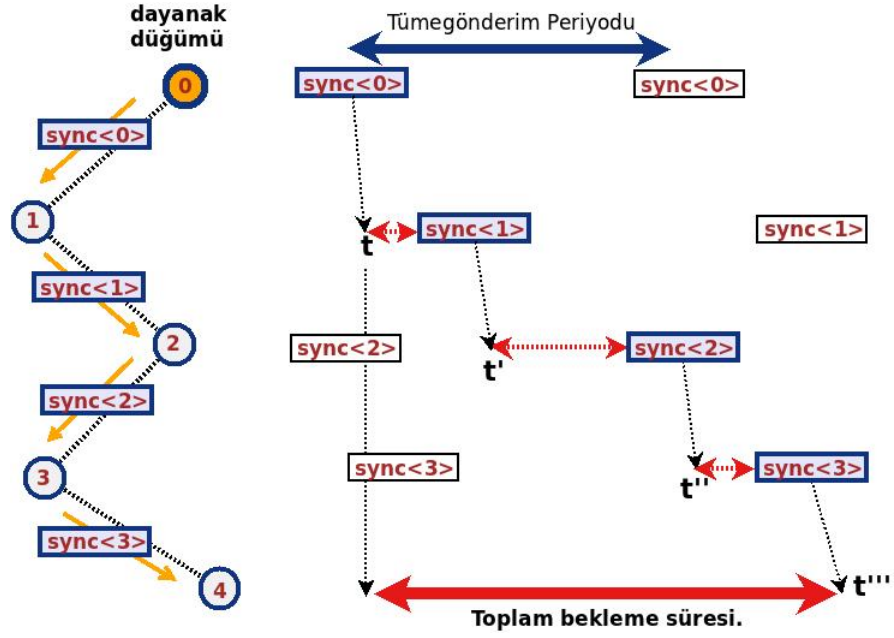
FTSP protokolünü yürüten her v düğümü, dayanak düğümünden en son aldığı

sıra numarasını saklamak için seq_v değişkenine sahiptir. İlk olarak her düğüm sıra numarasını sıfıra eşitlemekte ve B saniyede bir zamanaşımına uğrayacak bir zamanlayıcı başlatmaktadır (1-3 Satırları). En küçük kareler yöntemini uygulayabilmek için, v düğümü kendi donanımsal saati ve dayanak düğümünün saati arasında $L_{kök}(t) = \alpha + \beta H_v(t)$ şeklinde ifade edilen doğrusal bir ilişki olduğunu varsaymaktadır. Herhangi bir $u \in \mathcal{N}_v$ komşusundan t' anında kök düğümünün saatine ilişkin güncel bir tahmin alındığında (5. Satır), v düğümü elde ettiği $(H_v(t'), \hat{L}_u(t'))$ ikilisini (x, Y) şeklinde en güncel N adet zaman bilgisini depolayabilen *en küçük kareler tablosunda (EKKT)* saklamaktadır.¹³ u düğümünün tahmini mantıksal saat değeri olan $\hat{L}_u(t')$ 'nin aynı zamanda kök düğümünün mantıksal saatinin de bir tahmini olduğu göz önünde bulundurulmalıdır. v düğümü depolanan ikilileri kullanarak kendi mantıksal saatini temsil eden $L_v(t) = \hat{\alpha}_{ekk} + \hat{\beta}_{ekk} H_v(t)$ fonksiyonunu, en küçük kareler yöntemi ile hesaplamaktadır (6. Satır). Son olarak, alınan sıra numarası saklanmaktadır (7. Satır). Bir düğüm açılıp ilkeme işlemi yapıldıktan sonra, $\hat{\alpha}$ parametresinin değerinin 0 ve $\hat{\beta}$ parametresinin değerinin 1 olduğu unutulmamalıdır. Böylelikle, bu parametrelerin değeri yeni bir eşzamanlama mesajı alımı sonucunda değişmediği sürece $L_v(t) = H_v(t)$ eşitliği sağlanmaktadır.

Bir zamanlayıcı zamanaşımı gerçekleştiğinde (9. Satır), o düğüm dayanak düğümü ise sıra numarası bir arttırılmaktadır (10. Satır), değilse sıra numarası değişmeden kalmaktadır. Her düğüm kendi mantıksal saat değerini ve sıra numarasını içeren bir mesaj tümegöndermektedir (11. Satır). Sıra numarasının arttırılması ile beraber kök düğümünün yeni bir sel turunu başlattığına dikkat edilmelidir.

Şekil 5.1, FTSP protokolünün yürütümünde 0 kimliğine sahip dayanak düğümünün güncel zaman bilgisinin ağa nasıl yaydığını göstermektedir. Tüm düğümler aynı tümegönderim periyoduna sahip olmalarına rağmen farklı zamanlarda açıldıkları için, eşzamanlama paketlerini farklı anlarda tümegöndermektedirler. Bunun sonucunda, aldıkları güncel zaman bilgisini hemen gönderememekte, bir süre beklemeleri gerekmektedir. Örneğin 1 kimliğine sahip düğüm, dayanak düğümünden tümegönderilen $\langle sync\ 0 \rangle$ paketini t anında aldığı anda, bu paketi kullanarak tahmin edebildiği dayanak düğümünün güncel saat bilgisini 2 kimliğine sahip düğüme gönderebilmek için, kendi tümegönderim periyodunu beklemesi gerekmektedir. Güncel saat tahminini içeren $\langle sync\ 1 \rangle$ paketi, 2 düğüme ancak t' zamanında ulaşmaktadır. Benzer şekilde 2 düğümünün hazırlamış olduğu ve dayanak düğümünün güncel saatine ilişkin bir tahmin içeren $\langle sync\ 2 \rangle$ paketi, 3 düğüme ancak t'' anında ulaşmaktadır. Şekilde görüldüğü

¹³Bu ikilinin sistemden nasıl alındığı, 3. bölümde açıklanmıştır.



Şekil 5.1: FTSP protokolünde zaman bilgisinin sel yöntemi ile yavaş bir şekilde ağa yayılımı.

gibi, t''' anında 3 düğümü tarafından gönderilen $\langle sync3 \rangle$ paketini alan 4 düğümü, yaklaşık $t''' - t$ olan bir gecikmeyle dayanak düğümüne ilişkin saat tahminini almış olmaktadır.

5.2.1 En küçük kareler yöntemini kullanan yavaş selin eşzamanlama hatası

FTSP'nin v_0 düğümünün dayanak düğümü olduğu $v_0, v_1, \dots, v_{n-1} \in V$ doğrusal ilingesinde yürütüldüğü durum göz önüne alınsın. Yürütüm sırasında, v_1 düğümü v_0 düğümünden gelen birbirinden bağımsız $H_{v_0} + err_{v_0}$ saat değerlerini toplamaktadır. Toplanan saat değerleri, olgu 3.3 ile belirtildiği gibi, mesaj gecikmelerinin neden olduğu $err_{v_0} \sim (\mu, \sigma^2)$ dağılışına sahip bir hataya sahiptirler. v_1 düğümü kendi donanım saati ve dayanak düğümünün donanım saati arasında $H_{v_0} = \alpha_{v_1} + \beta_{v_1} H_{v_1} + err_{v_0}$ şeklinde bir doğrusal ilişki kurmaktadır. v_1 düğümü yeterli sayıda eşzamanlama noktası topladıktan sonra, kendi mantıksal saatini temsil eden en küçük kareler doğrusunu hesaplamaktadır. v_0 düğümünden gelecekte gelecek olan saat değerleri, v_1 düğümünün en küçük kareler doğrusunu hesaplamak için kullanılan noktalardan bağımsızdır. Dolayısıyla v_1 düğümünün mantıksal saati $L_{v_1} = H_{v_0} + e_{v_1}$, 4.5 dağılışı ve 4.6 eşitsizliği göz önüne alınacak olursa dağılışı aşağıda gösterilen e_{v_1} hatasına sahiptir:

$$e_{v_1} \sim \mathcal{N}(\mu, \sigma^2(\mathcal{F}_{v_1} + 1)). \quad (5.1)$$

Sonuç olarak alınan saat tahmininin varyansı $(\mathcal{F}_{v_1} + 1)$ katına çıkmıştır. Benzer şekilde, yürütüm esnasında v_2 düğümü v_1 düğümünden gelen eşzamanlama noktalarını toplamaktadır. Mesaj gecikmelerinden kaynaklanan hatalar tahminleme hatalarından bağımsız oldukları için, v_2 düğümünün topladığı $L_{v_1} = H_{v_0} + err_{v_1}$ saat tahminlerinin hataları $err_{v_1} \sim (2\mu, \sigma^2(\mathcal{F}_{v_1} + 1) + \sigma^2)$ dağılışına sahiptirler. Alınan her bir tahminin birbirinden bağımsız olduğu unutulmamalıdır, çünkü v_1 düğümünün birbirinden bağımsız donanım saat değerleri kullanılarak hesaplanmışlardır. v_2 düğümü topladığı tahminlerle kendi donanım saati arasında $L_{v_1} = \alpha_{v_2} + \beta_{v_2} H_{v_2} + err_{v_1}$ ile ifade edilen bir doğrusal ilişki kurmaktadır. Topladığı eşzamanlama noktalarını kullanarak en küçük kareler doğrusunu hesaplayan v_2 düğümü, v_0 düğümünün gelecekteki donanım saat değerini tahmin edebilmektedir. Gelecekte alınacak olan tahminler, en küçük kareler doğrusunu hesaplamak için kullanılmış olan tahminlerden bağımsız oldukları için, v_2 düğümünün mantıksal saati $L_{v_2} = H_{v_0} + e_{v_2}$ dağılışı aşağıda gösterilen e_2 tahminleme hatasına sahiptir:

$$e_{v_2} \sim \mathcal{N}(2\mu, (\sigma^2(\mathcal{F}_{v_1} + 1) + \sigma^2)(\mathcal{F}_{v_2} + 1)). \quad (5.2)$$

Bu yürütüm ile, v_{n-1} düğümünün herhangi bir t anındaki saat tahmininin hatası aşağıdaki gibi genelleştirilebilir:

$$e_{v_{n-1}} \sim \mathcal{N}\left(\sum_{i=0}^{n-2} \mu, \sum_{j=0}^{n-2} \left(\sigma^2 \prod_{k=j+1}^{n-1} (\mathcal{F}_{v_k} + 1)\right)\right). \quad (5.3)$$

Yukarıdaki olasılık dağılışı, bizi aşağıdaki gerçeğe götürmektedir.

Olgu 5.1. *Yavaş selin yürütümü sırasında, dayanak düğümünün tahminlenen saat değerinin varyansı her $v_i \in V$ sekmesinde $\mathcal{F}_{v_i} + 1$ katına çıkmaktadır.*

Olgu 5.2. *Dayanak düğümünden uzaklaştıkça, düğümler daha büyük hatalarla zaman bilgisini toplamaktadır. Dolayısıyla, dayanak düğümüne uzak düğümlerin tahminleme hatası yakın düğümlerden çok daha büyüktür.*

Eğer tüm $v_i \in V$ ve $v_j \in V$ düğümleri için $\mathcal{F}_{v_i} \approx \mathcal{F}_{v_j} \approx \mathcal{F}$ eşitliğinin

sağlandığı varsayılırsa, aşağıdaki sonuca ulaşılmaktadır.

Olgu 5.3. *Dayanak düğümü v_0 ile herhangi başka bir v_i düğümü arasındaki mantıksal saat farkı $\mathcal{O}(\mu d(v_0, v_i) + \sigma(\mathcal{F} + 1)^{d(v_0, v_i)/2})$ ile üstten sınırlanmıştır.*

5.3 Eğim Geçmişi Kullanan Yavaş Sel Yöntemi

4. bölümde, en küçük kareler yöntemi ile hesaplanmış olan tahminlerin hatasının olasılık dağılışı verilmişti. Bu dağılışı, dağılışı 4.4'te gösterilen en küçük kareler yöntemi ile hesaplanan eğimin olasılık dağılışına bağlıdır (Ross, 2004). Eğer bu dağılışın varyansı özellikle dayanak düğümüne uzak düğümler için düşürülürse, \mathcal{F} fonksiyonunun değeri ve dolayısıyla tahminleme hataları bu düğümler için düşecektir. Bu bölümde, en küçük kareler ile hesaplanan eğimin varyansını düşürecek üç yöntem önerilmektedir. Önerilen yöntemler her düğümün 5.1 Algoritmasını yürütürken en güncel N adet eğim değerini saklayacak bir *eğim geçmişi* (EG) sahip olmasını gerektirir. Başka bir deyişle her düğüm bu algoritmanın 6. satırını işlettiğinde ve en küçük kareler doğrusunun eğimini hesapladığında, EG'deki en eski eğim çıkartılır ve hesaplanan güncel eğim değeri EG'ye eklenir. Bu işlem FTSP protokolünün işleyişinde yapılan tek değişikliktir. Ancak önerilen yöntemler, düğümler üzerindeki uygulamalar için ortak zaman değerini, en küçük kareler yöntemi ile hesapladıkları mantıksal saat değeri olarak değil de, EG'deki eğim değerlerini farklı şekillerde kullanarak hesapladıkları *uygulama mantıksal saati* $L_v^{uygulama}$ ile sağlamaktadırlar. Burada, bir v düğümü Algoritma 5.1'in 11. satırını yürütürken, diğer düğümlere yine en küçük kareler yöntemi ile hesapladığı mantıksal saat değeri olan

$$L_v(t) = \hat{\alpha}_{ekk} + \hat{\beta}_{ekk} H_v(t) \quad (5.4)$$

değerini gönderdiğine dikkat edilmelidir.

5.3.1 EG'deki medyan eğimi göz önüne almak (ME)

En küçük kareler yönteminde, elde edilen eşzamanlama noktalarından biri bile büyük bir hataya sahip ise, tahmini doğrusal bağlanım doğrusunun eğimi bozulabilmektedir. Medyan tabanlı yöntemlerin bu durumun etkisini önemli ölçüde azalttıkları ve bu yöntemlerle hatalara karşı daha sağlam doğrusal bağlanım doğruları hesaplandığı bilinen bir gerçektir (Dietz, 1987; Nevitt and Tam, 1998;

Wilcox, 2010). Dolayısıyla, EG'deki medyan eğim (ME) hatalara karşı daha duyarlıdır ve varyansı en küçük kareler eğiminden daha düşüktür. EG'deki medyan eğimi kullanılarak, v düğümünün uygulamalara ileteceği t anındaki ortak saat değeri aşağıda gösterildiği gibi hesaplanmaktadır:

$$\begin{aligned}\hat{\beta}_{med} &= \text{medyan} \{ \beta_i \in EG, | i = 0..N - 1 \}, \\ \hat{\alpha}_{med} &= \bar{Y} - \hat{\beta}_{med} \bar{x}, \\ L_v^{uygulama}(t) &= \hat{\alpha}_{med} + \hat{\beta}_{med} H_v(t).\end{aligned}\tag{5.5}$$

İddia 5.4. $\hat{\beta}_{med}$ eğiminin varyansı en küçük kareler eğimi $\hat{\beta}_{ekk}$ 'nin varyansından küçüktür.

Bu iddia 5.7 bölümünde sunulacak deneysel sonuçlar ile desteklenecektir. Tahmini doğrusal bağlanım doğrusunun eğiminin varyansı azaldığı için \mathcal{F} değeri ve dolayısıyla saat tahminlerinin hataları düşecektir. ME yönteminin bir eksikliği EG'deki sadece medyan eğimi göz önüne alması ve diğerlerini ihmal etmesidir. Dolayısıyla, $\hat{\beta}_{med}$ 'in hesaplanma sürecinde diğer eğimlerin ağırlıkları 0'dır.

5.3.2 EG'deki ortalama eğimi göz önüne almak (OE)

Medyan tabanlı yöntemle benzer şekilde, tahmini doğrusal bağlanım doğrusunun eğiminin hatasını azaltacak bir diğer yöntem EG'deki eğimlerin ortalamasını almaktır. Bu yöntemde, v düğümünün uygulamalara ileteceği t anındaki ortak saat değeri, EG'deki ortalama eğim değeri kullanılarak aşağıda gösterildiği gibi hesaplanmaktadır:

$$\begin{aligned}\hat{\beta}_{ort} &= \sum_{\beta_i \in EG} \beta_i / N, \\ \hat{\alpha}_{ort} &= \bar{Y} - \hat{\beta}_{ort} \bar{x}, \\ L_v^{uygulama}(t) &= \hat{\alpha}_{ort} + \hat{\beta}_{ort} H_v(t).\end{aligned}\tag{5.6}$$

EG'deki her eğim gerçek doğrusal bağlanım doğrusunun eğimi için bağımsız bir tahmin olduğu ve her eğimin hatasının 4.4 dağılımına sahip olduğu için, bunların ortalamasını almak hatanın varyansını $1/N$ oranında azaltacaktır. Dolayısıyla \mathcal{F} ve bunun sonucunda tahmini saat değerlerinin hataları azalacaktır. Unutulmamalıdır

ki, $\hat{\beta}_{ort}$ eğiminin hesaplanması sürecinde EG'deki tüm eğimler aynı ağırlığa sahiptir.

Teorem 5.5. $\hat{\beta}_{ort}$ eğiminin varyansı en küçük kareler eğimi $\hat{\beta}_{ekk}$ 'nin varyansından $1/N$ oranında küçüktür.

5.3.3 EG'deki ortalama eğimle birlikte güncel en küçük kareler eğimini göz önüne almak (OGE)

EG'nde yer alan eğimler büyük hatalara sahipse, eğimlerin medyanını ya da ortalamalarını almak tahminleme hatalarını düşürebilir. Ancak, güncel en küçük kareler doğrusunun eğimi tahmini doğrusal bağlanım doğrusu için daha doğru bir tahmin ise, EG'de bulunan diğer eğimler bu daha doğru eğimin etkisini azaltabilir. Bu durumu engellemek için, EG'deki ortalama eğim ile güncel en küçük kareler doğrusunun eğiminin ortalamasının alınması önerilmektedir. Dolayısıyla, güncel en küçük kareler doğrusunun eğimine EG'deki diğer eğimlere göre daha büyük bir ağırlık verilmektedir. Bu yöntemde, v düğümünün uygulamalara ileteceği t anındaki ortak saat değeri aşağıda gösterildiği gibi hesaplanmaktadır:

$$\begin{aligned}\hat{\beta}_{oge} &= \left(\hat{\beta}_{ekk} + \sum_{\beta_i \in EG} \beta_i / N \right) / 2, \\ \hat{\alpha}_{oge} &= \bar{Y} - \hat{\beta}_{oge} \bar{x}, \\ L_v^{uygulama}(t) &= \hat{\alpha}_{oge} + \hat{\beta}_{oge} H_v(t).\end{aligned}\tag{5.7}$$

Görüldüğü gibi $\hat{\beta}_{oge}$ 'nin hesaplanması için güncel en küçük kareler doğrusunun eğimi $\hat{\beta}_{ekk}$ ve EG'deki ortalama eğim $\hat{\beta}_{ort}$ kullanılmaktadır. $\hat{\beta}_{ort}$ eğiminin $\hat{\beta}_{ekk}$ eğiminin varyansını $1/N$ oranında azalttığı gösterilmiştir. $\hat{\beta}_{ort}$ ve $\hat{\beta}_{ekk}$ eğimlerinin birbirinden bağımsız olmalarından ötürü, bu iki eğimin ortalamasının varyansı, $\hat{\beta}_{ekk}$ eğiminin varyansından $(N + 1)/4N$ oranında az olacak, \mathcal{F} ve bunun sonucunda tahminleme hataları düşecektir.

Teorem 5.6. $\hat{\beta}_{oge}$ eğiminin varyansı en küçük kareler eğimi $\hat{\beta}_{ekk}$ 'nin varyansından $\frac{N+1}{4N}$ oranında düşüktür.

5.4 En Küçük Varyansa Sahip İkili Eğimi (EKVSİE) Kullanan Yavaş Sel Yöntemi

Bu bölümde, özellikle dayanak düğümüne uzak düğümler için en küçük kareler yöntemi ile hesaplanan doğrunun eğiminin varyansını düşüren ve işlemci yükü açısından daha verimli olan bir başka yöntem önerilmektedir. En küçük kareler doğrusunu hesaplamak için 4.2 eşitliği dışında bir başka yöntem daha vardır. $\beta_{i,j}$, 5.8 eşitliğinde gösterildiği gibi, (x_i, Y_i) ve (x_j, Y_j) noktalarını birleştiren doğrunun eğimi olsun:

$$\beta_{i,j} = \frac{Y_i - Y_j}{x_i - x_j}, \quad 0 \leq i < j \leq N - 1. \quad (5.8)$$

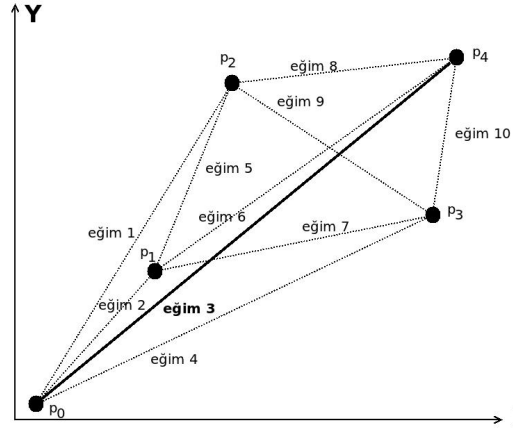
$w_{i,j}$ ise $\beta_{i,j}$ ikili eğiminin ağırlığı olsun ve aşağıdaki gibi tanımlansın:

$$w_{i,j} = \frac{(x_i - x_j)^2}{\sum_{k=0}^{N-2} \sum_{l=k+1}^{N-1} (x_k - x_l)^2}. \quad (5.9)$$

İkili eğimler ve bu eğimlerin ağırlıkları kullanılarak, tahmini doğrusal bağlanım doğrusunun eğimi ikili eğimlerin ağırlıklı bir ortalaması olarak 5.10 eşitliğinde gösterildiği gibi hesaplanabilmektedir (Gelman and Park, 2009):

$$\hat{\beta}_{ekk} = \sum_{i,j} w_{i,j} \beta_{i,j}. \quad (5.10)$$

Bu eğimlerin, noktaların hatalarından nasıl etkilendiğini gösterebilmek için, varyansları göz önüne alınmalıdır. Bir $v \in V$ düğümünün hesapladığı β_{ij} ikili eğiminin varyansı aşağıda gösterildiği gibi hesaplanabilir:



Şekil 5.2: 5 eşzamanlama noktası için ikili eğimler. 3. eğim, tahmini doğrusal bağlanım doğrusu için EKVSİE tahminleyicisidir.

$$\begin{aligned}
 Var(\beta_{i,j}) &= Var\left(\frac{Y_i - Y_j}{x_i - x_j}\right) \\
 &= \frac{Var(Y_i - Y_j)}{(x_i - x_j)^2} \\
 &= \frac{2\sigma_{e_v}^2}{(x_i - x_j)^2}.
 \end{aligned} \tag{5.11}$$

Görüldüğü gibi $x_i - x_j$ büyüdükçe, eğimin varyansı azalmaktadır.

Bu gerçek göz önüne alınarak, tahmini doğrusal bağlanım doğrusunun eğimi olarak N adet nokta içerisinde birbirine en uzak 2 nokta arasındaki, yani en düşük varyansa sahip, ikili eğimin alınması önerilmektedir. *En Küçük Varyansa Sahip İkili Eğim (EKVSİE)*, aşağıdaki gibi tanımlanmaktadır:

$$\hat{\beta}_{ekvsie} = \left\{ \frac{Y_i - Y_j}{x_i - x_j} \mid x_i - x_j = \max_{k,l} \{x_k - x_l\} \right\}. \tag{5.12}$$

Şekil 5.2, 5 adet nokta için hesaplanmış 10 adet ikili eğimi göstermektedir. Şekildeki 3. eğim, tahmini doğrusal bağlanım doğrusu için EKVSİE tahminleyicisidir. EKVSİE tahminleyicisi kullanılarak, tahmini doğrusal bağlanım doğrusunun kesimi aşağıda gösterildiği gibi hesaplanmaktadır:

$$\hat{\alpha}_{ekvsie} = \bar{Y} - \hat{\beta}_{ekvsie} \bar{x} \tag{5.13}$$

Yukarıdaki eşitliğin 4.3 eşitliğinden tek farkı $\hat{\beta}_{ekk}$ yerine $\hat{\beta}_{ekvsie}$ kullanılmasıdır. Dolayısıyla EKVSİE yönteminde mantıksal saat aşağıdaki şekilde hesaplanmaktadır:

$$L_v(t) = \hat{\alpha}_{ekvsie} + \hat{\beta}_{ekvsie} H_v(t). \quad (5.14)$$

Mantıksal saat değeri, aynı zamanda uygulamalar için de geçerli olacak ortak zamanı temsil etmektedir.

5.4.1 En küçük kareler eğimi ve EKVSİE eğimlerinin karşılaştırılması

EKVSİE eğimi diğer ikili eğimlerle kıyaslandığında, düşük varyansından dolayı büyük hatalardan daha az etkilenmektedir. $x_i < x_{i+1}$ eşitsizliğinin $i = 0..N - 1$ değerleri için sağlandığı en güncel N adet (x_i, Y_i) noktaları göz önüne alınsın. Eğer paket kayıplarının olmadığı varsayılırsa, B periyodu ile zaman bilgisi değiş tokuşundan dolayı aşağıdaki eşitlik sağlanmaktadır:

$$E[x_i] = x_0 + iB, \quad i = 1, \dots, N - 1. \quad (5.15)$$

Bu eşitlik kullanılarak, EKVSİE eğimi olan $\beta_{N-1,0}$ eğiminin en küçük kareler yöntemindeki beklenen ağırlığı, 5.10 eşitliği kullanılarak aşağıdaki gibi gösterilebilir:

$$\begin{aligned} E[w_{N-1,0}] &= E \left[\frac{(x_{N-1} - x_0)^2}{\sum_{i=0}^{N-2} \sum_{j=i+1}^{N-1} (x_i - x_j)^2} \right] \\ &= E \left[\frac{((N-1)B)^2}{\sum_{i=0}^{N-1} (N-i)(iB)^2} \right] \\ &= E \left[\frac{12(N-1)}{N^2(N+1)} \right]. \end{aligned} \quad (5.16)$$

Zaman etiketi tablosu büyüklüğü algılayıcı düğümlerinde genelde $N = 8$ olduğu için

$$E[w_{N-1,0}] \simeq 0.146 \quad (5.17)$$

elde edilir. Bu sonuç, daha düşük varyansa sahip ve hatalardan daha az etkilenen ikili eğimin en küçük kareler yöntemindeki ağırlığının yaklaşık %15 olduğunu gösterir. Bu eğimin doğrusal bağlanım doğrusunun eğiminin hesaplamasındaki katkısının düşük olmasından dolayı, özellikle dayanak düğüme uzak düğümler EKVSİE yöntemine göre daha büyük hatalar gösterebilirler.

İddia 5.7. EKVSİE eğimi dayanak düğüme uzak düğümlerin tahminleme hatalarını düşürmekte, büyük hatalara sahip noktalardan daha az etkilenmekte ve en küçük kareler eğimine göre daha kararlı olmaktadır.

İlerleyen bölümlerde bu iddiayı destekleyecek deneysel ve benzetim sonuçları sunulacaktır.

5.5 Hız Seli Eşzamanlama Protokolü (HSEP)

5.3 ve 5.4 bölümlerinde sunulan yöntemler, tahminleme hatalarının her sekmede artmasını engellememekte, sadece hataların en küçük kareler yöntemine göre her sekmede daha az artmasını sağlamaktadırlar. Dayanak düğüme uzakta bulunan düğümlerin tahminleme hatası hala ağın çapının üstel bir fonksiyonudur. Çünkü düğümler bekleme nedeniyle hataları her sekmede şişmiş eşzamanlama noktalarını kullanarak en küçük kareleri uygulamaktadırlar. Bu durum, bizi aşağıdaki gerçeğe götürmektedir.

Olgu 5.8. *Yavaş sel yönteminde, en küçük kareler uzak düğümlerden gelen zaman bilgisi üzerinde uygulanmamalıdır.*

Bu bölümde FTSP'ye alternatif olarak, sözde kodu Algoritma 5.2'de gösterilen *Hız Seli Eşzamanlama Protokolü (HSEP)* tanıtılmaktadır. Burada dayanak düğümünün *KÖK* kimliğine sahip olduğu ve tüm düğümlerin bu kimliği bildiği varsayılmaktadır. HSEP'te her düğümler dayanak düğümünün tahmini saat değeri yanında dayanak düğümünün saatinin kendi donanımsal saatine göreceli olarak hesapladığı hızını da ağa yaymaktadır. Düğümler, en küçük kareler yöntemini sadece bir sekme uzaklarında olan komşularından gelen veriler üzerinde, komşularının göreceli donanım saat hızlarını hesaplamak için uygulamaktadırlar.

Algoritma 5.2 v düğümü için HSEP sözde kodu.

```

1: İkleme
2: veri havuzunu temizle ve  $\forall u \in \mathcal{N}_v : h_v^u \leftarrow 1$ 
3:  $taban_v \leftarrow 0$ ;  $sonGüncellenme_v \leftarrow 0$ ;  $seq_v \leftarrow 0$ 
4:  $h_v^r \leftarrow 1$ 
5:  $B$  periyodu ile periyodik bir zamanlayıcı başlat
6:
7:  $\square < L_u, H_u, h_u^r, seq_u >$  alındığında
8:  $(H_v, H_u)$  'yi sakla ve  $h_v^u$  değerini en küçük kareler yöntemiyle tahmin et
9: eğer  $seq_v < seq_u$ 
10:    $taban_v \leftarrow L_u$ 
11:    $sonGüncellenme_v \leftarrow H_v$ 
12:    $seq_v \leftarrow seq_u$ 
13:    $h_v^r \leftarrow h_u^r \cdot h_v^u$ 
14: soneğer
15:
16:  $\square$  Zamanlayıcı zamanaşımında
17: eğer  $v = KÖK$  ise  $seq_v \leftarrow seq_v + 1$  soneğer
18: tümegönder  $< L_v, H_v, h_v^r, seq_v >$ 

```

Dayanak düğümünün tahmini saatini hesaplayabilmek için, her $v \in V$ düğümü kendi mantıksal saatine ait değişkenlere sahiptir. Mantıksal saatin herhangi bir t anındaki değeri aşağıdaki gibi hesaplanmaktadır:

$$L_v(t) = taban_v(t) + (H_v(t) - sonGüncelleme_v(t)) h_v^r(t). \quad (5.18)$$

Buradaki $taban_v$ alınmış en güncel eşzamanlama mesajı ile taşınan mantıksal saat değerini ve $sonGüncelleme_v$ ise $taban_v$ değişkeninin en son güncellendiği andaki donanım saati değerini saklamaktadır. h_v^r dayanak düğümünün donanım saatinin v düğümünün donanımsal saatine göre göreceli tahmini hızını ($\frac{h_r}{h_v}$ için bir tahmin) temsil etmektedir. Bu değer aynı zamanda mantıksal saatin hız çarpanı olarak da düşünülebilir. 5.18 eşitliğinden görüleceği üzere mantıksal saat, taban değerine, bu değer en son güncellendiği andan itibaren dayanak düğümünün mantıksal saatinin tahmini olarak ilerleme miktarı eklenerek hesaplanmaktadır. Her düğüm komşularının göreceli donanımsal saat hızlarını saklamak için bir veri havuzuna ve dayanak düğümünden alınan en son sıra numarasını saklamak için bir seq_v değişkenine sahiptir. Depolama kısıtları nedeniyle, düğümlerin takibini yapabilecekleri komşu düğüm sayısı sınırlıdır. v düğümü açıldığı anda komşular için tutulan veri havuzu temizlenir, her düğüm için saklanacak olan göreceli donanım saat hızları 1 ile ilklenir (2. Satır). seq_v , $taban_v$ ve $sonGüncelleme_v$

değişkenleri sıfırlanır ve h_v^r değeri 1 ile ilklenir (3-4 Satırları). Son olarak donanımsal saat her B ilerlediğinde zaman aşımına uğrayacak bir zamanlayıcı kurulur (5. Satır).

Bir $u \in \mathcal{N}_v$ düğümünden her eşzamanlama mesajı alındığında, elde edilen (H_v, H_u) ikilisi veri havuzunda o komşu düğüm için ayrılmış bölümde saklanır. Göreceli donanım saat ilerleme hızı, o komşu düğüm için saklanmış olan ikililer üzerinde en küçük kareler yöntemi uygulanarak tahmin edilir (7-8 Satırları). En küçük kareler yöntemi ile hesaplanan tahmini doğrusal bağlanım doğrusunun eğimi, h_u/h_v değeri için bir tahmindir.

Eğer alınan eşzamanlama paketi daha büyük bir sıra numarasına sahipse (9. Satır), bu durum dayanak düğümünün kendi mantıksal saat değerini ağa yaymaya başladığını gösterir. Dolayısıyla, v düğümü alınan mantıksal saat değerini taban değişkeninde ve bu andaki donanım saati değerini $sonGüncelleme_v$ değişkeninde saklamaktadır (10-11 Satırları). Ardından sıra numarası güncellenmektedir (12. Satır). v düğümü u düğümünün göreceli donanım saat hızını tahminleyebildiği için (h_v^u), dayanak düğümünün göreceli donanım saat hızı olan h_v^r 'yi, h_u^r değerini h_v^u ile çarparak tahmin edebilir (13. Satır).

Zamanlayıcı zamanaşımına uğradığı anda (16. Satır), eğer düğüm dayanak düğümü ise sıra numarası bir arttırılır (17. Satır). Değilse, sıra numarası değişmeden kalmaktadır. Sıra numarasının bir artması, dayanak düğümünün yeni bir sel turunu başlattığını işaret etmektedir. Her düğüm kendi mantıksal değerini, donanımsal saat değerini, dayanak düğümünün tahmini göreceli donanım saat hızını ve sıra numarasını tümegönderir (18. Satır).

5.5.1 Analiz

HSEP protokolünün v_0 düğümünün dayanak düğümü olduğu $v_0, v_1, \dots, v_{n-1} \in V$ doğrusal ilingesinde çalıştığı durum göz önüne alınsın. Bütün $v_i \in V$ düğümleri için $(H_{v_i}(t) - sonGüncelleme_{v_i}(t))^2 \leq \mathcal{U}$ ve $\frac{1}{S_{xx}} \leq \frac{1}{S}$ eşitsizliklerinin tüm düğümler için tüm t anlarında sağlandığı varsayılınsın. Yürütüm sırasında v_1 düğümü v_0 düğümünden $err_{v_0} \sim \mathcal{N}(\mu, \sigma^2)$ hatasına sahip bağımsız donanım saat değerleri toplamaktadır. $taban_{v_1}$ değişkeni en son alınan saat değeri ile güncellendiği için $taban_{v_1} \sim \mathcal{N}(H_{v_0} + \mu, \sigma^2)$ elde edilmektedir. Ek olarak 4.4 dağılışı göz önüne alındığında $h_{v_1}^{v_0} \sim \mathcal{N}(\frac{h_{v_0}}{h_{v_1}}, \kappa_{v_1} = \frac{\sigma^2}{S})$ sağlanmaktadır. $taban_{v_1}$ ve $h_{v_1}^{v_0}$ değişkenleri bağımsız oldukları için, v_1 düğümünün $L_{v_1} = H_{v_0} + e_{v_1}$ mantıksal saat değerinin e_{v_1} hatasının olasılık dağılışı aşağıdaki gibi ifade edilebilir:

$$e_{v_1} \sim \mathcal{N}(\mu, \sigma^2 + \mathcal{U}\kappa_{v_1}) \quad (5.19)$$

Mesaj gecikmelerinden kaynaklanan hatalar e_{v_1} hatasından bağımsız oldukları için, v_2 düğümünün eline geçen v_1 düğümünün saat tahminleri $err_{v_1} \sim \mathcal{N}(2\mu, 2\sigma^2 + \mathcal{U}\kappa_{v_1})$ hatasına sahiptir. $taban_{v_2}$ değişkeni en son alınan saat değeri ile güncellendiği için $taban_{v_2} \sim \mathcal{N}(H_{v_0} + 2\mu, 2\sigma^2 + \mathcal{U}\kappa_{v_1})$ elde edilmektedir. $h_{v_2}^{v_1} \sim \mathcal{N}(\frac{h_{v_1}}{h_{v_2}}, \frac{\sigma^2}{S})$ olduğu için, ve $h_{v_1}^{v_0}$, $h_{v_2}^{v_1}$ birbirinden bağımsız oldukları için, $h_{v_2}^{v_0} = h_{v_1}^{v_0} \cdot h_{v_2}^{v_1}$ için aşağıdaki dağılışı elde edilir:

$$h_{v_2}^{v_0} \sim \left(\frac{h_{v_0}}{h_{v_2}}, \kappa_{v_2} \right) \quad (5.20)$$

öyleki $\kappa_{v_2} = \left(\frac{h_{v_0}}{h_{v_1}} \right)^2 \frac{\sigma^2}{S} + \left(\left(\frac{h_{v_1}}{h_{v_2}} \right)^2 + \frac{\sigma^2}{S} \right) \kappa_{v_1}$.¹⁴ Dolayısıyla, v_2 düğümünün $L_{v_2} = H_{v_0} + e_{v_2}$ mantıksal saat değerinin e_{v_2} hatasının olasılık dağılışı aşağıdaki gibi ifade edilebilir:

$$e_{v_2} \sim \mathcal{N}(2\mu, 2\sigma^2 + \mathcal{U}\kappa_{v_1} + \mathcal{U}\kappa_{v_2}) \quad (5.21)$$

Aynı adımlar v_{n-1} düğümü için tekrarlandığında, $L_{v_{n-1}}$ mantıksal saat değerinin hatasının olasılık dağılışı aşağıdaki gibi formüleleştirilebilir:

$$e_{v_{n-1}} \sim \mathcal{N} \left(\sum_{i=1}^{n-1} \mu, \sum_{i=1}^{n-1} \sigma^2 + \mathcal{U} \sum_{m=1}^{n-1} \kappa_{v_m} \right). \quad (5.22)$$

Olgu 5.3, düğümlerdeki bekleme süreleri \mathcal{F} değerini oldukça büyüttüğü için, FTSP protokolünün eşzamanlama hatasının ağın çapının büyümesinden oldukça olumsuz etkilendiğini göstermişti. HSEP protokolünde ise düğümlerdeki bekleme zamanları \mathcal{U} değerini arttırmaktadır. Ancak, \mathcal{U} değeri eşitlik 5.22'de sadece bir katsayı olarak yer almaktadır ve bu değerin büyümesinin etkisi FTSP protokolüne kıyasla çok daha düşüktür. Tüm $v_i, v_j \in V$ düğümleri için

¹⁴Eğer X ve Y birbirlerinden bağımsız şans değişkenleri ise, $Var(X.Y) = (E[X])^2 Var(Y) + (E[Y])^2 Var(X) + Var(X)Var(Y)$.

$h_{v_i}/h_{v_j} \approx 1$ olduğu için $\kappa_{v_i} \approx (\frac{\sigma^2}{S} + 1)^i - 1$ ve dolayısıyla $\sum_{m=1}^{n-1} \kappa_{v_m} \approx \left((\frac{\sigma^2}{S} + 1)^n - 1 \right) \frac{S}{\sigma^2} - n$ elde edilir.¹⁵ Dolayısıyla, aşağıdaki teoreme ulaşılmaktadır.

Teorem 5.9. *HSEP protokolünde dayanak düğümü v_0 ile herhangi bir başka düğüm $v_i \in V$ arasındaki mantıksal saat farkı $\mathcal{O} \left(\mu d(v_0, v_i) + \sqrt{d(v_0, v_i) \sigma^2 + \mathcal{U} \kappa} \right)$ ile üstten sınırlanmıştır, öyle ki $\kappa = \left((\frac{\sigma^2}{S} + 1)^{d(v_0, v_i)+1} - 1 \right) \frac{S}{\sigma^2} - (d(v_0, v_i) + 1)$.*

Olgu 5.10. *HSEP protokolünde düğümlerdeki bekleme zamanlarının eşzamanlama üzerinde yarattığı olumsuz etki FTSP protokolüne kıyasla çok daha düşüktür.*

5.6 Saat Hızı Uzlaşması İle Sel Protokolü

Bir önceki bölümde HSEP protokolünün eşzamanlama hatasının ağın çapının üstel bir fonksiyonu olduğu gösterilmişti. Bu bölümde ağdaki tüm düğümlerin ortak bir saat hızında *saat hızı uzlaşması* ile anlaştığı Saat Hızı Uzlaşması ile Sel (SHUS) protokolü tanıtılmaktadır. Aynı saat hızında uzlaşma sonucunda, bekleme sürelerinin saat eşzamanlaması üzerindeki olumsuz etkileri ortadan kalkacağı ve eşzamanlama hatasının ağın çapının karekökünün bir fonksiyonu ile sınırlanacağı gösterilecektir. SHUS algoritmasının HSEP protokolüne oldukça benzeyen sözde kodu Algoritma 5.3'te gösterilmiştir.

SHUS protokolünü yürüten düğümler, HSEP protokolünde dayanak düğümünün göreceli donanım saat hızını saklamaya ihtiyaç duymazlar. Bunun yerine her $v \in V$ düğümü hız çarpanını temsil eden l_v değişkenine sahiptir. Bu değişken mantıksal saatin ilerleme hızını arttırmak ya da yavaşlatmak için kullanılmaktadır. SHUS protokolünde herhangi bir t anındaki mantıksal saat değeri aşağıdaki gibi hesaplanmaktadır:

$$L_v(t) = \text{taban}_v(t) + (H_v(t) - \text{sonGüncelleme}_v(t)) l_v(t). \quad (5.23)$$

Yukarıdaki eşitlik, $(H_v(t) - \text{sonGüncelleme}_v(t))$ değerinin h_v^u yerine l_v ile çarpılması dışında 5.18 eşitliği ile aynıdır. HSEP protokoine benzer şekilde her düğüm kendi komşularının takibini yapmak için bir veri havuzuna sahiptir. Farklı olarak bu veri havuzu herhangi bir $u \in \mathcal{N}_v$ düğümünden alınan (H_v, H_u) ikilileri dışında l_u değerlerini de saklamaktadır. HSEP protokolündeki ilklenme adımlarına

¹⁵ $\sum_{i=1}^{n-1} x^i = \frac{1-x^n}{1-x} - 1$.

Algoritma 5.3 v düğümü için, kimliği $KÖK$ olan sabit bir dayanak düğümüne sahip SHUS sözde kodu.

1: Initialization

2: veri havuzunu temizle ve $\forall u \in \mathcal{N}_v : l_u \leftarrow 1, h_v^u \leftarrow 1$
3: $taban_v \leftarrow 0$; $sonGüncelleme_v \leftarrow 0$; $seq_v \leftarrow 0$
4: $l_v \leftarrow 1$
5: B periyodu ile periyodik bir zamanlayıcı başlat
6:
7: $\square < L_u, H_u, l_u, seq_u >$ **alındığında**
8: (H_v, H_u) 'yi sakla ve h_v^u değerini en küçük kareler yöntemiyle tahmin et
9: (h_v^u, l_u) ikilisini sakla ve l_v 'yi güncelle
10: **eğer** $seq_v < seq_u$
11: $taban_v \leftarrow L_u$
12: $sonGüncelleme_v \leftarrow H_v$
13: $seq_v \leftarrow seq_u$
14: **soneğer**
15:
16: \square **Zamanlayıcı zamanaşımında**
17: **eğer** $v = KÖK$ **ise** $seq_v \leftarrow seq_v + 1$ **soneğer**
18: **tümegönder** $< L_v, H_v, l_v, seq_v >$

ek olarak, sistem açıldığında l_v değişkeni ve her $u \in \mathcal{N}_v$ komşu düğümü için tutulan l_u değerleri bir ile ilklenmektedir (2. ve 4. Satırlar).

HSEP ile kıyaslandığında SHUS protoköünün ana farkı Algoritma 5.3'ün yeni bir eşzamanlama mesajı alındığında işletilen 9. satırıdır. Her $u \in \mathcal{N}_v$ düğümü için saklanan h_v^u ve l_u değişkenleri kullanılarak, v düğümünün mantıksal saat çarpanı aşağıdaki gibi hesaplanmaktadır (9. Satır):

$$l_v(t^+) = \frac{l_v(t) + \sum_{u \in \mathcal{N}_v} h_v^u(t) \cdot l_u(t)}{|\mathcal{N}_v| + 1}. \quad (5.24)$$

Burada t^+ güncellemeden hemen sonraki bir zamanı ifade etmektedir. Bu yürütüm ile tüm düğümlerin ortak bir mantıksal saat hızında uzlaştıkları ispatlanabilir (Schenato and Fiorentin, 2011; Sommer and Wattenhofer, 2009).

Düğümler ortak bir mantıksal saat hızında anlaşmanın yanı sıra, eşzamanlama mesajları ile taşınan mantıksal saat değerlerini (dayanak düğümünün tahmini mantıksal saat değeri) mantıksal saatin taban değerini güncellemek için kullanmaktadır. Bu yürütüm ile, her eşzamanlama mesajı mantıksal saatin hız çarpanını güncellemek için kullanılmaktadır. Ancak, sadece yeni eşzamanlama turunu ait (daha büyük bir sıra numarasına sahip) mesajlar mantıksal saatin taban

değerini güncellemek için kullanılırlar.

5.6.1 Analiz

Bu bölümde SHUS protokolünün eşzamanlama hatasının ağın çapının karekökünün bir fonksiyonu ile üstten sınırlandığı gösterilmektedir. Öncelikle, eğer $\mu_{e_v} = 0$ ve $\sigma_{e_v}^2 = 0$ eşitlikleri 3.10 eşitliğindeki hata için sağlanırsa (yani iletişim gecikmeleri ihmal edilirse), ağdaki bütün algılayıcı düğümlerin uzlaşma mekanizması ile ortak bir saat hızında anlaştıklarını gösteren aşağıdaki teorem sunulmaktadır.

Teorem 5.11. *G çizgesi sıkı bağlı bir şekilde kaldığı sürece, tüm $v \in V$ algılayıcı düğümleri **hız** ortak saat hızında uzlaşırlar, öyleki $\lim_{t \rightarrow \infty} (h_v(t).l_v(t)) = \mathbf{hız}$.*

İspat. $x(t)$ her i girdisi $i \in V$ düğümünün t anındaki mantıksal saat hızı $h_i(t).l_i(t)$ değerini içeren $n \times 1$ 'lik matris ve M matrisi her bir a_{ij} girdisi aşağıdaki gibi tanımlanmış $n \times n$ 'lik satır-stokastik matris olsun:

$$a_{ij} = \begin{cases} \frac{1}{|\mathcal{N}_i|+1} & \text{eğer } \{i, j\} \in E \text{ ise} \\ 0 & \text{değilse} \end{cases}. \quad (5.25)$$

Bu matrisler kullanılarak, mantıksal saat hızını güncelleyen 5.24 eşitliği $x(t^+) = M.x(t)$ şeklinde yazılabilir. İspat, G çizgesi sıkı bağlı kaldığı sürece, satır-stokastik matrislerin çarpımlarının bir sayıya yakınsadığı gerçeğine dayanmaktadır. Detaylı ispat (Schenato and Fiorentin, 2011; Sommer and Wattenhofer, 2009) çalışmalarında bulunabilir. Bu sonuç algılayıcı ağlarındaki düğümlerin komşu sayısını sınırlandırmanın, çizgenin sıkı bağlılığını bozmadığı sürece uzlaşmayı engellemeyeceği gerçeğini de ortaya koymaktadır. \square

Ancak, $\mu = 0$ ve $\sigma^2 = 0$ değerlerinin sağlanması KAA için gerçekçi olmayacağı için, uzlaşma bir hatayla gerçekleşmektedir. Kolaylık olması için, eşzamanlama turlarının iki aşamadan oluştuğu varsayalım. İlk aşamada algılayıcı düğümleri komşularından hız çarpanlarını tolasınlar. Tüm düğümler bu aşamayı bitirdikten sonra, ikinci aşamada düğümler 5.24 eşitliğine göre hız çarpanlarını güncellesinler. Bütün $v_i \in V$ düğümleri için $(H_{v_i}(t) - sonGüncelleme_{v_i}(t))^2 \leq \mathcal{U}$ ve $\frac{1}{S_{xx}} \leq \frac{1}{S}$ eşitsizliklerinin tüm düğümler için tüm t anlarında sağlandığı varsayalım. İlk eşzamanlama turunda, $l_v \sim \mathcal{N}(1, \eta_1 = \frac{|\mathcal{N}_v|}{|\mathcal{N}_v|+1} \frac{\sigma^2}{S})$ dağılışı elde

edilmektedir çünkü hız çarpanlarının ilk değerleri 1'dir ve $\forall v_i, v_j \in: h_{v_i}/h_{v_j} \approx 1$ sağlanmaktadır. 2. turda gönderilen ve alınan tüm hız çarpanları bu dağılışa sahiptirler. Eşitlik 5.24'te yer alan l_v , h_v^u ve l_u değişkenleri birbirinden bağımsız oldukları için 2. tur sonunda $l_v \sim \mathcal{N}(1, \eta_2 = \frac{\eta_1}{|\mathcal{N}_v+1|} + \eta_1 + \eta_1^2)$ sağlanmaktadır.¹⁶ Dolayısıyla, i. eşzamanlama turu sonunda hız çarpanının dağılışı aşağıdaki gibi genelleştirilebilir:

$$l_v \sim \mathcal{N}\left(1, \eta_i = \frac{\eta_{i-1}}{|\mathcal{N}_v+1|} + \eta_1 + \eta_1 \eta_{i-1}\right). \quad (5.26)$$

Eğer $\xi = \frac{1}{|\mathcal{N}_v+1|} + \eta_1$ olursa, $n_i = \eta_1 + \eta_1 \sum_{j=1}^{i-1} \xi^j$ elde edilmektedir. $|\mathcal{N}_v| \geq 1^{17}$ ve $\eta_1 \ll 1$ olduğu için, $\xi < 1$ ve $\sum_{j=1}^{\infty} \xi^j = \xi/(1-\xi)$ sağlanmaktadır. Eğer $\eta = \frac{|\mathcal{N}_v|}{|\mathcal{N}_v+1|(1-\xi)} \frac{\sigma^2}{S}$ şeklinde ifade edilirse, hız anlaşması sağlandıktan sonra l_v değişkeninin dağılışı 5.27'de gösterildiği gibi olmaktadır:

$$l_v \sim (1, \eta). \quad (5.27)$$

SHUS protokolünün v_0 düğümünün dayanak düğümü olduğu $v_0, v_1, \dots, v_{n-1} \in V$ doğrusal ilingesinde çalıştığı ve uzlaşmanın sağlandığı andan sonraki durumlar göz önüne alınsın. Yürütüm sırasında v_1 düğümü v_0 düğümünden $err_{v_0} \sim \mathcal{N}(\mu, \sigma^2)$ hatasına sahip bağımsız donanım saat değerleri topladığı için, $taban_{v_1} \sim \mathcal{N}(H_{v_0} + \mu, \sigma^2)$ elde edilmektedir. $taban_{v_1}$ ve l_{v_1} değişkenleri birbirinden bağımsız oldukları için, v_1 düğümünün $L_{v_1} = H_{v_0} + e_{v_1}$ mantıksal saat değerinin e_{v_1} hatasının olasılık dağılışı aşağıdaki gibi ifade edilebilir:

$$e_{v_1} \sim \mathcal{N}(\mu, \sigma^2 + \mathcal{U}\eta) \quad (5.28)$$

Benzer şekilde, v_2 düğümü v_1 düğümünden $err_{v_1} \sim \mathcal{N}(2\mu, 2\sigma^2 + \mathcal{U}\eta)$ hatasına sahip saat tahminleri almaktadır. $taban_{v_2} \sim \mathcal{N}(2\mu, 2\sigma^2 + \mathcal{U}\eta)$ olduğu için, v_2 düğümünün $L_{v_2} = H_{v_0} + e_{v_2}$ mantıksal saat değerinin e_{v_2} hatasının olasılık dağılışı aşağıdaki gibi ifade edilebilir:

¹⁶ $Var(l_v^+) = \frac{Var(l_v) + \sum_{u \in \mathcal{N}_v} Var(h_v^u \cdot l_u)}{(|\mathcal{N}_v|+1)^2}$.

¹⁷İletişim çizgesi bağlı olmalıdır.

$$e_{v_2} \sim \mathcal{N}(2\mu, 2\sigma^2 + 2\mathcal{U}\eta) \quad (5.29)$$

Aynı adımlar v_{n-1} düğümü için tekrarlandığında, $L_{v_{n-1}}$ mantıksal saat değerinin hatasının olasılık dağılışı aşağıdaki gibi formülleştirilebilir:

$$e_{v_{n-1}} \sim \mathcal{N}\left(\sum_{i=0}^{n-2} \mu, \sum_{i=0}^{n-2} \sigma^2 + \mathcal{U} \sum_{j=0}^{n-2} \eta\right). \quad (5.30)$$

Bu gerçeği kullanan aşağıdaki teorem, SHUS protokolünde, dayanak düğümü ile bu düğümüne en uzak düğümün arasındaki mantıksal saat farkının ağın çapının karekökünün bir fonksiyonu olduğunu göstermektedir.

Teorem 5.12. *SHUS protokolünde dayanak düğümü v_0 ile herhangi bir başka düğüm $v_i \in V$ arasındaki mantıksal saat farkı $\mathcal{O}\left(\mu d(v_0, v_i) + \sqrt{d(v_0, v_i)(\sigma^2 + \mathcal{U}\eta)}\right)$ ile üstten sınırlanmıştır.*

Olgu 5.13. *SHUS protokolü yürütülürken düğümlerdeki bekleme sürelerinin eşzamanlama hatasına etkisi, FTSP ve HSEP protokolleri ile kıyaslandığında oldukça düşüktür.*

5.7 Deneysel Sonuçlar

Bu bölümde, teorik olarak gösterilen sonuçların ve iddiaların uygulamada da geçerli ve tutarlı olduğunu göstermek için, deneysel sonuçlar sunulacaktır. Önerilen yöntemler ve protokoller KAA için genel geçer işletim sistemi olan TinyOS 2.1.1¹⁸ ortamında gerçekleşmiştir. Yöntem ve protokollerin başarımlarının karşılaştırılması için, literatürdeki birçok çalışma tarafından kıyaslamada kullanılmış (Sommer and Wattenhofer, 2009; Lenzen et al., 2009b; Schmid et al., 2009; Schmid et al., 2010a) ve yavaş sel yöntemini uygulayan FTSP protokolü göz önüne alınmıştır. Deneylerde FTSP protokolünün TinyOS işletim sistemi ile beraber gelen açık kaynak kodlu gerçekleştirimi kullanılmıştır.

¹⁸<http://www.tinyos.net>

5.7.1 Göz önüne alınan başarıml ölçütleri

Önerilen yöntemlerin ve protokollerin değeriendirilmesi ve kıyaslanması için ağ içindeki herhangi iki düğüm arasındaki ve birbirlerine komşu düğümler arasındaki anlık en büyük eşzamanlama hatası (genel ve yerel saat farkları) göz önüne alınmıştır. Bunlara ek olarak, herhangi iki düğüm arasındaki ve birbirine komşu düğümler arasındaki anlık ortalama eşzamanlama hataları da (ortalama genel ve ortalama yerel saat farkları) gözetilmiştir.

5.7.2 Deneyler için kullanılan algılayıcı düğümleri

Deneylerde Memsic¹⁹ firması tarafından üretilen MICAz algılayıcı düğümleri kullanılmıştır. MICAz donanım ortamı düşük güç tüketimine, 8 bitlik, 4 kB RAM belleğe ve 128 kB kalıcı belleğe sahip Atmega128L mikrodeneleyicisini kullanmaktadır. MICAz kartında yer alan Chipcon CC2420 radyo yongası 2.4 GHz frekansında ve 250 kbps veri hızında çalışmaktadır. Zamanlayıcılar için, 7.37 MHz'lik kuvars salınıcı kristali saat kaynağı olarak kullanılmıştır. Zamanlayıcılar bu kristalin frekansının 1/8'inde, yani 921 kHz'de çalışmaktadırlar. Dolayısıyla düğümler yaklaşık 1 mikrosaniye hassasiyetinde zaman ölçümü yapabilmektedirler. MICAz ortamında OEK katmanında zaman etiketleme işlevselliği için, TinyOS tarafından sağlanan paket seviyesinde saat eşzamanlaması servisleri (Maroti and Sallai, 2008) kullanılabilmektedir.

5.7.3 Deney düzeneği

Deneylerde bir dayanak tümegönderici düğümün kapsama alanı içerisinde yer alan 20 adet algılayıcı düğümü kullanılmıştır. Düğümler kendi kimliklerinden bir fazla ya da bir az kimliğe sahip düğümlerden gelen mesajları kabul etmekte ve diğerlerini ihmal etmektedirler. Bu sayede yazılımsal olarak doğrusal bir ilinge oluşturulmuş olmaktadır. Doğrusal ilinge seçilmesinin nedeni, yavaş sel yönteminin başarımlının ağın çapının büyümesi ile düşmesi ve 20 algılayıcı düğümü ile en büyük çapın doğrusal ilinge ile elde edilmesidir (Lenzen et al., 2009b).

Deneylerde, eşzamanlama paketi gönderme periyodu 30 saniyedir ve protokol tablolarında (zaman etiketi tablosu, eğim geçmiş gibi) en fazla 8 girdinin saklanmasına izin verilmiştir. Çevresel etkenler oldukça kararlıdır ve ani sıcaklık değişimi yoktur. Algılayıcı düğümler rastgele bir şekilde ilk 3 dakika içinde

¹⁹<http://www.memsic.com>

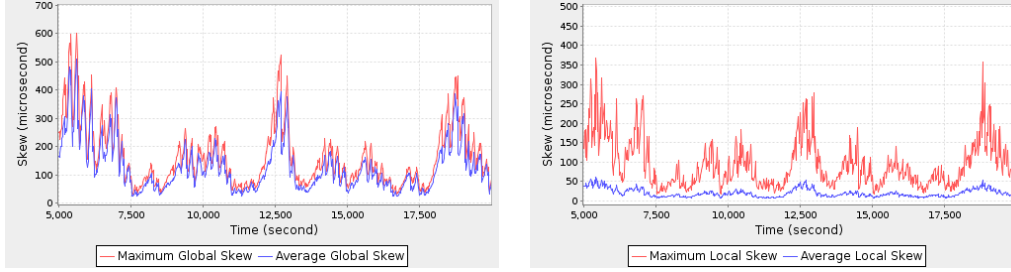
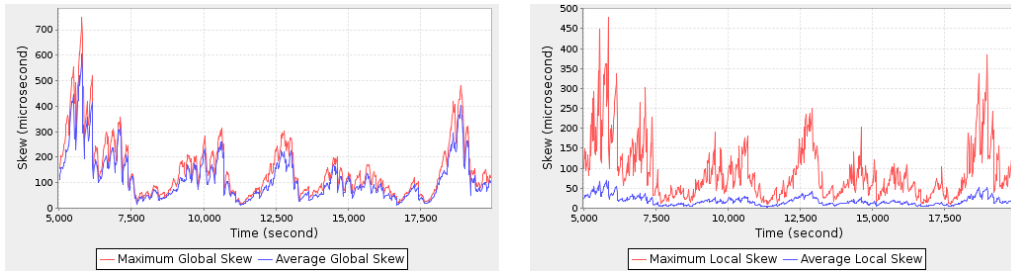
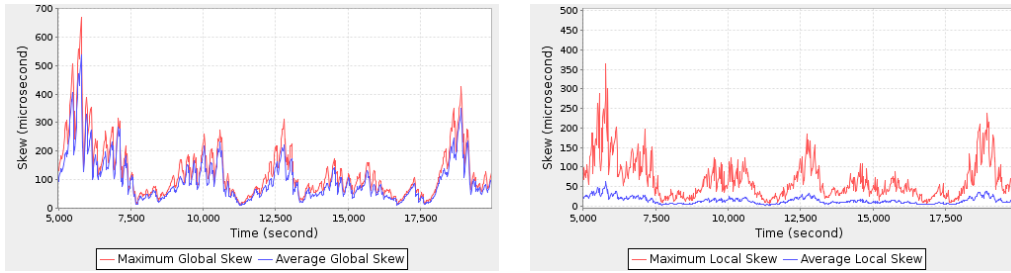
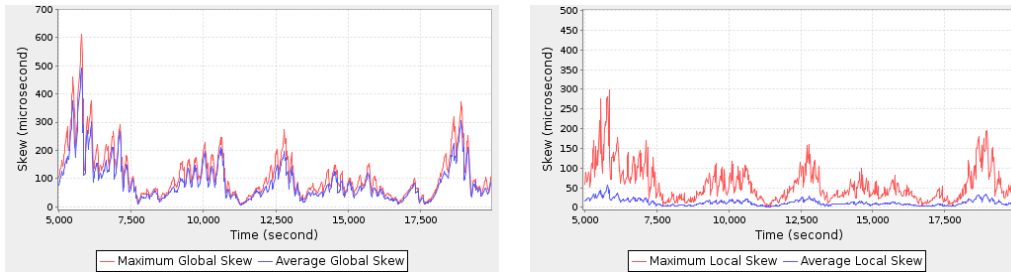
açılmaktadır. 20 ile 23 saniye aralığında uniform dağılışa sahip her bir zaman dilimi sonunda, dayanak tümegönderici düğüm bir dayanak paketi tümegöndermektedir. Dayanak paketi algılayıcı düğümlerin hepsi tarafından yaklaşık olarak aynı anda alınmakta ve düğümler paketi aldıkları andaki mantıksal saat değerlerini PC'ye bağlı bir merkezi düğüme göndermektedirler. Merkezi düğüm aldığı mantıksal saat değerlerini seri porttan, bu portu dinleyen bir uygulamaya gönderir. Uygulama kendisine gelen anlık saat değerlerini saklamaktadır. Deney sonunda, saklanan anlık saat değerleri kullanılarak anlık genel ve yerel saat farkları ve bunların ortalama değerleri hesaplanmaktadır.

5.7.4 Eğim geçmişi kullanan yöntemler için deneysel sonuçlar

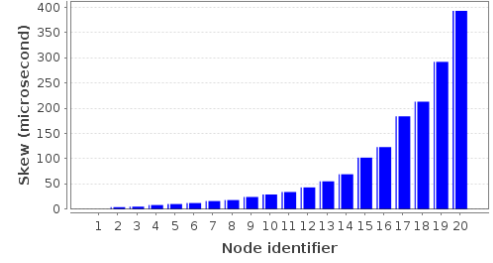
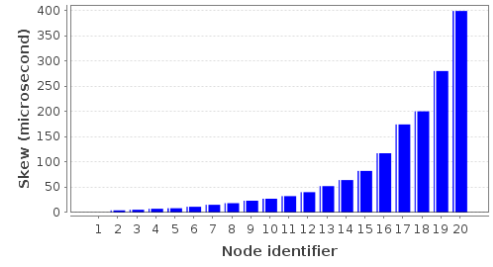
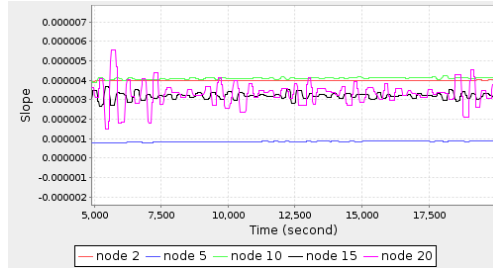
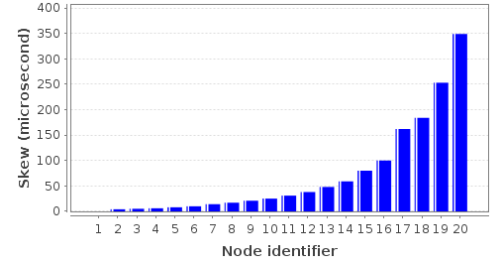
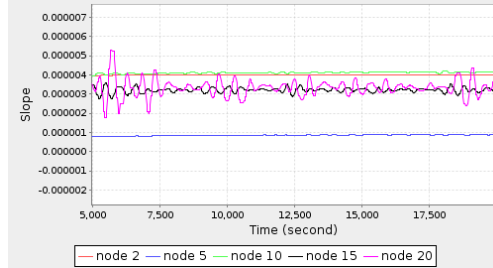
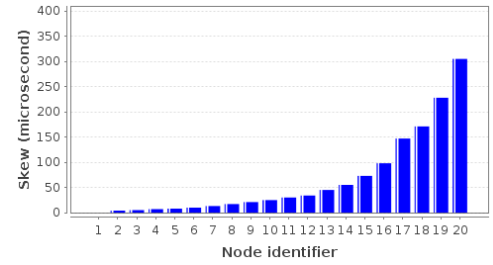
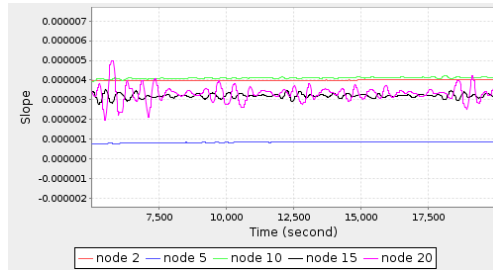
Bu deneyde FTSP protokolüne bir eğim geçmişi eklenerek dayanak düğümünün saatini 5.3 bölümünde tanıtılan eğim tahminleme yöntemlerini kullanarak da tahmin etmesi sağlanmıştır. Yeni bir arayüz daha eklenerek, uygulamaların bu yöntemlerle tahmin edilen saat değerlerine erişmesi sağlanmıştır. Bu eklentilerle, FTSP protokolünün farklı eğim tahminleme yöntemleri başarımını, aynı mesaj gecikmeleri, paket kayıp oranları ve çevresel koşullarda değerlendirme imkanı sağlanmıştır.

Şekil 5.3 FTSP protokolü için EKK, ME, OE ve OGE yöntemleriyle gözlemlenen genel ve yerel saat farklarını göstermektedir. Şekil 5.4 tahmini doğrusal bağlanım doğrusunun eğimini ve dayanak düğümü ile diğer düğümler arasındaki en büyük saat farkını göstermektedir. Başarım değerlendirmesi için, düğümlerin eğim geçmişleri tamamen dolduktan sonraki saat farkı değerleri gözetilmiştir (yaklaşık 7000 saniye sürenin geçmesi gerekmektedir). Deney süresince EKK yöntemi ile, en büyük genel ve en büyük ortalama genel saat farkı değerleri olarak sırası ile $526 \mu s$ ve $396 \mu s$ gözlemlenmiştir. Ek olarak EKK yöntemi ile $357 \mu s$ en büyük yerel saat farkı ve $54 \mu s$ en büyük ortalama yerel saat farkı gözlemlenmiştir. Dayanak düğüme uzakta olan düğümler zaman bilgisini büyük hatalara aldıkları için, eşzamanlama hataları ve tahmini doğrusal bağlanım doğrusunun eğiminin kararsızlığı, Şekil 5.4'te görüldüğü gibi fazladır.

FTSP protokolü ME yöntemi ile en büyük genel ve en büyük ortalama genel saat farkı olarak $479 \mu s$ ve $402 \mu s$ değerlerini ortaya koymuştur. En büyük yerel ve en büyük ortalama yerel saat farkı olarak $384 \mu s$ ve $52 \mu s$ değerleri gözlemlenmiştir. ME yöntemi ile toplanan eğim değerleri EKK'ye göre daha kararlılık gösterebilir, dayanak düğümü ile diğer düğümler arasındaki en büyük

EKK**ME****OE****OGE**

Şekil 5.3: FTSP protokolünü sırasıyla EKK, ME, OE ve OGE eğimlerini kullanarak yürüten düğümler için anlık genel (sol sütun) ve yerel (sağ sütun) saat farkı değerleri.

EKK**ME****OE****OGE**

Şekil 5.4: FTSP protokolünü sırasıyla EKK, ME, OE ve OGE eğimlerini kullanarak yürüten düğümlerden toplanan eğim değerleri (sol sütun) ve kimliği 1 olan dayanak düğümü ile diğer düğümler arasında görülen mantıksal saat farkının en büyük değerleri (sağ sütun). Eğim değerleri, gerçek değerlerden 1.0 çıkartılmış şekilde gösterilmektedir.

saat farkı EKK ile hemen hemen aynıdır. Bu durum Şekil 5.4'te görülmektedir. Bunun nedeninin ME yönteminin eğim geçmişindeki sadece medyan eğimi göz önüne alması ve diğerlerini ihmal etmesi olarak düşünmekteyiz. Dolayısıyla, ME yönteminde daha doğru bir eğim, eğim geçmişindeki medyan eğim olmadığı için, göz önüne alınmamaktadır.

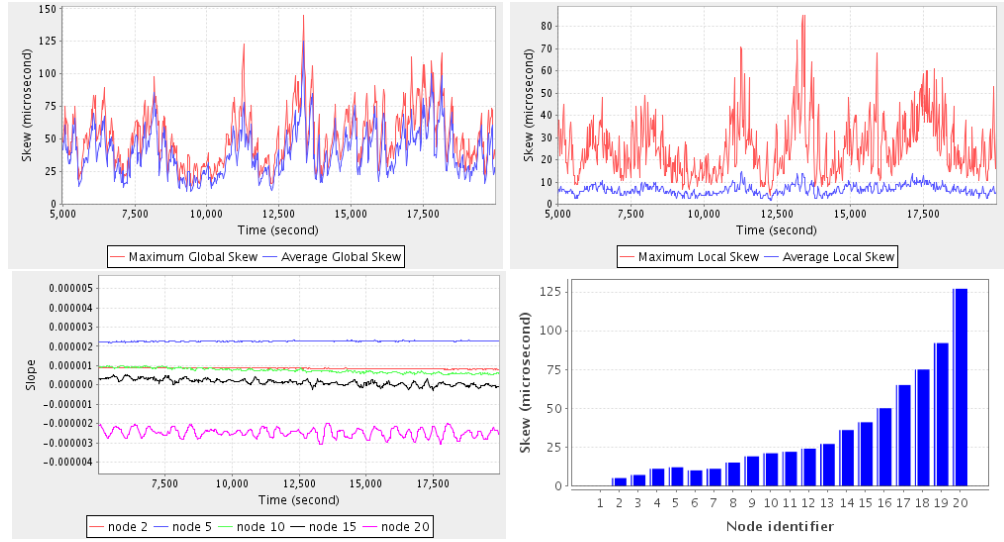
FTSP protokolü OE yöntemi ile $426 \mu s$ en büyük genel ve $349 \mu s$ en büyük ortalama genel saat farkı değerlerini ortaya koymuştur. Buna ek olarak, en büyük yerel ve en büyük ortalama yerel saat farkı değeri olarak sırası ile $239 \mu s$ ve $39 \mu s$ değerleri gözlemlenmiştir. EKK yöntemi ile hesaplanan tahmini doğrusal bağlanım doğrusunun eğiminin büyük değişkenlikler gösterdiği anlarda, OE yönteminin ME yönteminde olduğu gibi, bu değişkenliği oldukça azalttığı gözlemlenmiştir. Ancak OE ile uzak düğümler tarafından hesaplanan eğim değerleri, ME yöntemine göre zaman içinde daha yumuşak bir şekilde değişmektedir. OE yöntemi ile, dayanak düğüm ile bu düğüme uzak olan düğümler arasındaki saat farkının en büyük değeri düşmüştür. OE yönteminin ME yöntemine göre deneysel olarak daha iyi sonuç vermesinin nedeni, eğim geçmişindeki tüm eğimleri gözetiyor olmasıdır.

Deneylerde, OGE yöntemi OE yönteminden daha iyi başarımlar göstermiştir. En büyük genel ve en büyük ortalama saat farkı değerleri sırasıyla $371 \mu s$ ve $305 \mu s$ değerlerine inmiştir. Ek olarak en büyük yerel ve en büyük ortalama yerel saat farkı değerleri ise sırasıyla $196 \mu s$ ve $34 \mu s$ 'ye düşmüştür. OGE ile gözlemlenen eğim değerleri, Şekil 5.4'de görüleceği üzere daha karardır. OGE yönteminin OE yöntemine göre üstün olmasının nedeni OE yönteminin eğim geçmişindeki tüm eğimlere aynı ağırlığı vermesi ancak OGE yönteminin en güncel eğime daha büyük ağırlık vermesidir.

Deneysel sonuçlar, FTSP protokolünün tahmini doğrusal bağlanım doğrusunun eğimini hesaplama yönteminde ufak bir değişikliğin, 20 algılayıcı düğümden oluşan doğrusal bir ilingede hatırı sayılır bir başarımlar iyileşmesine yol açtığını göstermiştir. Bir eğim geçmişi eklenerek ve OGE eğim tahminleme yöntemi kullanılarak, selin hızı değiştirilmeden eşzamanlama kalitesi artmaktadır.

5.7.5 En küçük varyansa sahip ikili eğim yöntemi için deneysel sonuçlar

Bu deney için FTSP protokolüne en küçük varyansa sahip ikili eğim tahminleme yöntemini eklenerek düğümlerin bu yöntemle dayanak düğümünün saatini tahmin etmesi sağlanmıştır. Değiştirilmiş FTSP, bir önceki deneydeki aynı

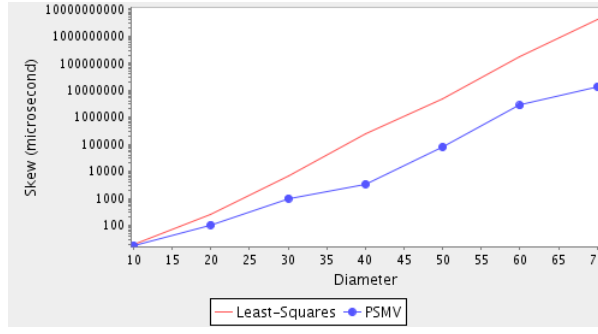


Şekil 5.5: EKVSİE yöntemini kullanan FTSP protokolünün, doğrusal ilingeye sahip 20 MICAz algılayıcı düğümü üzerinde yürütüldüğünde elde edilmiş olan sırası ile anlık en büyük genel saat farkı, en büyük ortalama genel saat farkı, en büyük yerel saat farkı, en büyük ortalama yerel saat farkı, eğim değerleri ve dayanak düğümü ile diğer düğümler arasında gözlemlenen en büyük saat farkı. Eğim değerleri, gerçek değerlerden 1.0 çıkartılmış şekilde gösterilmektedir.

düzenekte çalıştırılmıştır. Şekil 5.5’de EKVSİE yöntemi için hesaplanan anlık en büyük genel saat farkı, en büyük ortalama genel saat farkı, en büyük yerel saat farkı, en büyük ortalama yerel saat farkı, eğim değerleri ve dayanak düğümü ile diğer düğümler arasındaki en büyük saat farkı gösterilmektedir.

Şekil 5.5’deki eğim değerlerinden, EKVSİE yönteminin EKK ve eğim geçmişi kullanan diğer yöntemlere göre daha kararlı olduğu ve daha az değişkenlik gösterdiği görülebilmektedir. Dolayısıyla, önceki bölümlerde iddia edildiği gibi, EKVSİE yönteminde uzaktaki düğümler daha kararlı eğimlere sahip olmaktadır. Bu yöntem ile eşzamanlama kalitesi de belirgin bir biçimde artmaktadır. Sadece eğim tahminleme yöntemi değiştirilmiş olmakla beraber, FTSP protokolü EKVSİE yöntemi ile en büyük genel ve en büyük ortalama genel saat farkı olarak sırasıyla $145 \mu s$ ve $125 \mu s$ değerlerini ortaya koymuştur. En büyük yerel ve en büyük ortalama yerel saat farkı değerleride sırasıyla $85 \mu s$ ve $15 \mu s$ olarak gözlemlenmiştir. Dayanak düğümü ile diğer düğümler arasındaki en büyük saat farkı göz önüne alındığında, 14-20 arasındaki kimliklere sahip düğümler için gözlemlenen değerler EKK yöntemine göre oldukça iyileşmiştir. Tüm bu sonuçlar, EKVSİE yönteminin her sekmede artan hataların etkisini oldukça düşürdüğünü ortaya koymaktadır.

Bu yöntemin diğer bir avantajı ise, işlem yükünü EKK yöntemine göre oldukça düşürmesidir. $\hat{\beta}_{ekvsi}$ eğiminin hesaplanması 5.12 eşitliğinde gösterildiği

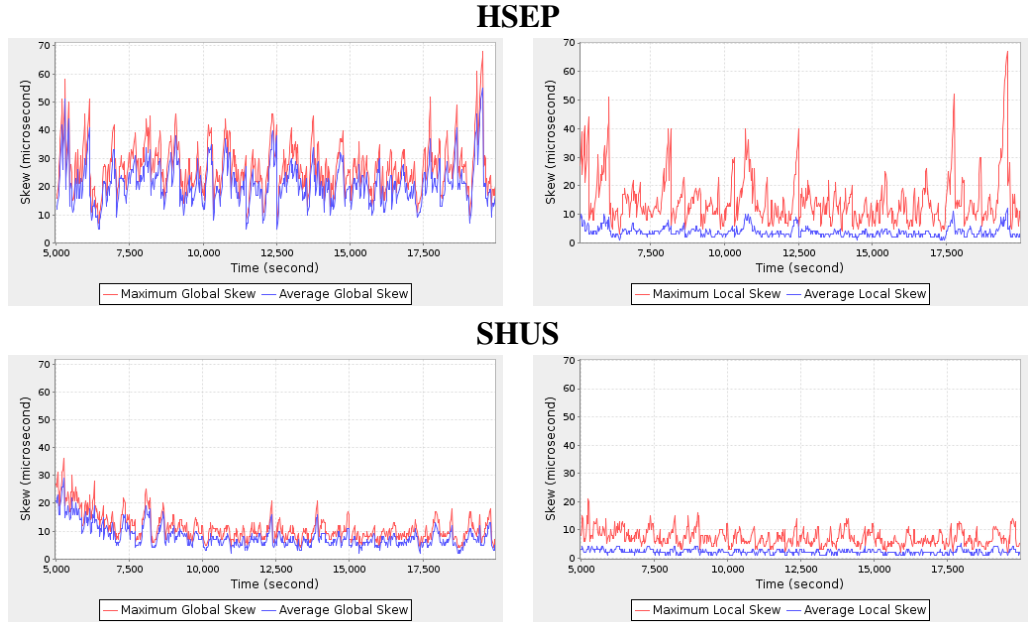


Şekil 5.6: FTSP protokolünün EKK ve EKVSİE yöntemleri için benzetim sonuçları.

gibi sadece 2 çıkartma ve 1 bölme işlemi gerektirmektedir. $\hat{\beta}_{ekk}$ eğiminin hesaplanması ise $6N + 1$ aritmetik işlem gerektirmektedir.²⁰ Algılayıcı düğümlerinde her x_i ve Y_i noktası 32 bitlik işaretli tamsayı ile ifade edilmektedir. $\hat{\beta}_{ekk}$ eğiminin hesaplanmasında $(x_i - \bar{x})(Y_i - \bar{Y})$ ve $(x_i - \bar{x})^2$ değerlerini saklamak için 32 bitten daha fazlasına ihtiyaç duyulmaktadır. Dolayısıyla, 4.2 eşitliğindeki bölen ve bölünen değerleri için 64 bitlik aritmetik işlemlere gerek vardır. $\hat{\beta}_{ekvsie}$ eğimi göz önüne alındığında ise, $(Y_{i+1} - Y_i)$ ve $(x_{i+1} - x_i)$ değerleri 32 bitlik tamsayılar ve 4.2 eşitliğindeki değeri hesaplamak için 64 bitlik işlemlere gerek yoktur. Algılayıcı düğümleri için 32 bitlik aritmetik işlemlerin getirdiği işlemci yükü 64 bitlik işlemlerin getirdiği yükten oldukça düşüktür. Deneyler esnasında, EKK eğiminin hesaplanması en fazla $5394 \mu s$ işlemci zamanı sürmüştür. Ancak, EKVSİE yöntemi ile eğim hesaplama yöntemi bu yükü $3145 \mu s$ işlemci zamanına düşürmüştür. EKVSİE yöntemi, eşzamanlama kalitesini yükseltmesinin yanı sıra, FTSP eşzamanlama protokolünün işlemci yükünü de yaklaşık %40 oranında düşürmüştür.

Deneylere ek olarak, FTSP protokolünün EKK ve EKVSİE yöntemlerini kullanan iki sürümü Java programlama dilinde geliştirilen benzetim aracı üzerinde benzetimi yapılarak kıyaslanmıştır. Bu sayede, EKVSİE yönteminin daha büyük çapa sahip ağlardaki başarımı hakkında ilk izlenimler elde edilmiştir. Benzetimler için algılayıcı düğümlerin donanım saatleri rastgele ± 50 ppm olacak şekilde modellenmiştir. Değişik çaptaki doğrusal ilingeye sahip ağlarda, her çap için 10 adet benzetim yapılmış ve bu benzetim sonuçlarında gözlemlenen en büyük genel saat farkı değerlerinin ortalaması alınmıştır. Şekil 5.6 benzetim sonuçlarını göstermektedir. Benzetim sonuçlarından, EKVSİE yönteminin eşzamanlama hatasının ağırlıklı olarak büyümesinden EKK yöntemine göre çok daha az etkilendiği ortaya çıkmaktadır.

²⁰ $2N$ çıkartma, $2N$ toplama, $2N$ çarpma ve 1 bölme.



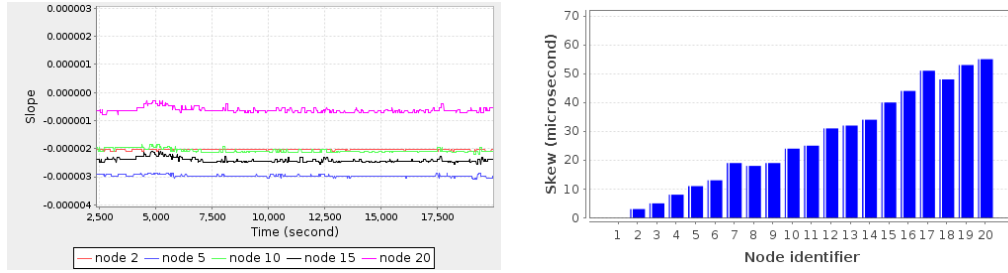
Şekil 5.7: HSEP ve SHUS protokolleri için ölçülen anlık genel ve yerel saat farkı değerleri.

5.7.6 HSEP ve SHUS protokolleri için deneysel sonuçlar

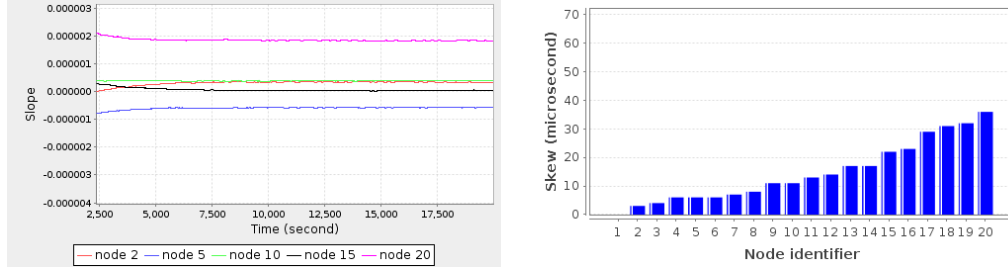
Bu bölümde HSEP ve SHUS protokollerinin daha önce belirtilen test düzeneğindeki başarımları kıyaslanacaktır. Şekil 5.7’de görüldüğü gibi, HSEP protokolü ile en büyük genel ve en büyük ortalama genel saat farkı değerleri sırasıyla $68 \mu s$ ve $55 \mu s$ olarak ölçülmüştür. En büyük yerel ve en büyük ortalama yerel saat farkı değerleri ise $67 \mu s$ ve $12 \mu s$ olarak ölçülmüştür. SHUS protokolü ise bu değerleri oldukça düşürmüştür. $25 \mu s$ en büyük genel, $19 \mu s$ en büyük ortalama genel, $16 \mu s$ en büyük yerel ve $5 \mu s$ en büyük ortalama yerel saat farkı SHUS protokolünü yürüten düğümlerde gözlenen eşzamanlama hatalarıdır. En büyük genel saat farkı göz önüne alındığında, HSEP protokolü FTSP protokolünü yaklaşık 8 kat, SHUS protokolü FTSP protokolünü ise yaklaşık 20 kat iyileştirmiştir. Bu protokoller FTSP ile aynı sel hızına, mesaj karmaşıklığına ve mesaj iletim desenine sahip olmakla beraber eşzamanlama kalitesi açısından FTSP protokolünden oldukça üstünlük göstermektedirler.

Şekil 5.8 HSEP ve SHUS protokollerinde gözlemlenen hız çarpanı değerlerini ve dayanak düğümü ile diğer düğümler arasında gözlemlenen en büyük saat farkını göstermektedir. HSEP protokolü tarafından uygulanan hızın sel yöntemi ile yayılımı ve SHUS protokolü tarafından uygulanan hız uzlaşımı yöntemleri sayesinde, en küçük kareler yönteminin FTSP protokolünün ölçeklenebilirliğine olan olumsuz etkisi büyük ölçüde ortadan kaldırılmıştır. Hız çarpanları (eğim değerleri) HSEP ve SHUS protokolleri için oldukça kararlı olduğu için, bu

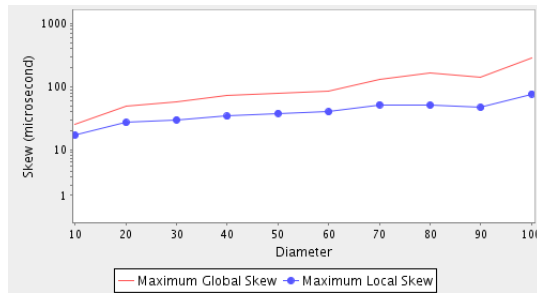
HSEP



SHUS



Şekil 5.8: HSEP ve SHUS protokolleri için toplanan eğim değerleri ve dayanak düğümü ile diğer düğümler arasında görülen en büyük saat farkı değerleri. Eğim değerleri, gerçek değerlerden 1.0 çıkartılmış şekilde gösterilmektedir.



Şekil 5.9: SHUS protokolü için farklı ağ çapları için benzetim sonuçları.

protokollerde dayanak düğümüne uzak düğümlerin eşzamanlama hataları FTSP protokolündeki uzak düğümlere göre oldukça azalmıştır.

Deneylere ek olarak, deneysel olarak çok iyi sonuç veren SHUS protokolünü kendi benzetim aracımızda çalıştırarak daha büyük çaplara sahip ağlarda başarımının nasıl etkilendiğini gözlemledik. Bunun için düğümlerin saatini yazılımla rastgele ± 50 sapmaya sahip olacak şekilde modellenmiştir. Doğrusal ilingeye sahip her çap için, 10 adet benzetim yapılmış ve bu benzetimlerdeki en büyük genel ve yerel saat farklarının ortalaması alınarak değerlendirilmiştir. Şekil 5.9'de gösterilen benzetim sonuçları, SHUS protokolünün eşzamanlama hatasının ağın çapının büyüdükçe çok daha yavaş arttığını göstermiştir.

Çizelge 5.1: Deneysel sonuçlarının özeti.

	FTSP					HSEP	SHUS
	EKK	ME	OE	OGE	EKVSİE		
En Büyük Genel Saat Farkı	526 μs	479 μs	426 μs	371 μs	145 μs	68 μs	25 μs
En Büyük Ortalama Genel Saat Farkı	396 μs	402 μs	349 μs	305 μs	125 μs	55 μs	19 μs
En Büyük Yerel Saat Farkı	357 μs	384 μs	239 μs	196 μs	85 μs	67 μs	16 μs
En Büyük Ortalama Yerel Saat Farkı	54 μs	52 μs	39 μs	34 μs	15 μs	12 μs	5 μs

5.7.7 Deneysel sonuçların özeti

FTSP protokolünün EKK, ME, OE, OGE ve EKVSİE yöntemleri ile, HSEP ve SHUS protokollerinin deney düzeneğinde yürütülmesi sonucu gözlemlenen eşzamanlama hatalarının özeti Çizelge 5.1’de göstermiştir. Sonuçlar, yavaş sel yöntemini kullanan SHUS protokolünün diğer tüm protokollerden eşzamanlama hatası açısından çok üstün olduğunu göstermektedir.

5.8 SHUS Protokolü İçin İyileştirme Önerileri

Önceki bölümlerde SHUS protokolünün yavaş sel yönteminin eşzamanlama üzerindeki etkisini FTSP ve HSEP protokolleri ile kıyaslandığında oldukça düşürdüğü gerçeği ortaya koyulmuştur. Bu bölümde, SHUS protokolü için bazı ek konular tartışılmaktadır.

Komşu Ekleme ve Silme: SHUS protokolü için yapılan gerçekleştirim basit bir komşu ekleme ve silme yöntemi uygulamaktadır. Bir düğüm herhangi bir komşu düğümden ilk defa mesaj aldığı anda, eğer komşular için kullandığı veri havuzu tamamen doluyorsa, o komşudan gelen verileri göz önüne almamaktadır. Eğer veri havuzunda boş bir yer varsa, o komşu için bir yer atanmakta ve o komşudan gelen veriler toplanmaya başlanmaktadır. Eğer veri havuzunda bilgisi takip edilen herhangi bir komşudan belirli bir zaman boyunca eşzamanlama mesajı alınmazsa, veri havuzunda o komşu için tutulan yer boşaltılarak o düğüm komşuluktan çıkartılmaktadır. Ağa yeni bir düğüm eklendiğinde, eklenen düğüm hemen düğümler arası hız uzlaşmasında yer almamaktadır. Bunun yerine öncelikle kendi komşularından birkaç eşzamanlama paketini dinleyerek, ağa bir ilk eşzamanlama

sağlamaktadır.

Dayanak Düğümü Seçimi: SHUS protokolünde dayanak düğümü ağ kurulmadan önce belirlenmiştir. Dolayısıyla SHUS protokolü, dayanak düğümü arızalandığında ya da ağın geri kalan kısmının dayanak düğümü ile bağlantısı kesildiğinde ağ genelinde eşzamanlama sağlayamamaktadır. Ancak FTSP protokolünde basit dayanak düğümü seçme adımları, SHUS protokolüne de kolayca eklenebilir.

Gerçek Zamana Eşzamanlanma: Eğer dayanak düğümü bir KKS ile donatıldıysa ve bu donanım üzerinden gerçek zamana erişime sahipse, tüm düğümlerin ortak saat hızında anlaştığı bir t' zamandan sonra $t' \leq t_0 < t_1$ olan iki gerçek zaman anında, kendi donanım saatinin değerini okuyarak, donanım saatinin hızını 5.31 eşitliğinde gösterildiği gibi tahmin edebilir:

$$\hat{h}_r = (H_r(t_1) - H_r(t_0)) / (t_1 - t_0). \quad (5.31)$$

Bu değeri kullanarak dayanak düğümü ağ genelindeki ortak saat hızını 5.32 eşitliğinde gösterildiği gibi tahmin edebilir:

$$\hat{h}z = \hat{h}_r \cdot l_r. \quad (5.32)$$

Eşzamanlama mesajlarının gerçek zaman değeri t ve uzlaşılan tahmini ortak saat hızı $\hat{h}z$ değerini de taşıyacak şekilde genişletilirse, her v düğümü $\hat{h}z = h_v \cdot l_v$ değerini ve l_v değerini bildiği için, kendi donanım saati hızı olan h_v 'yi tahmin edebilir. Bunları kullanarak, v düğümü bir gerçek zaman değerini aşağıdaki gibi tahmin edebilir:

$$gerçekzaman_v(t) = gztaban_v(t) + \frac{H_v(t) - sonGüncelleme_v(t)}{\hat{h}_v(t)}. \quad (5.33)$$

Burada $gztaban_v$ değişkeni en son alınan eşzamanlama mesajında taşınan gerçek zaman değeridir.

5.9 Sonular ve Gelecek alıřmalar

Bu alıřmada KAA’nda yavař sel yntemini kullanan saat eřzamanlama protokolleri iin, selin yavař ilerleme hızının eřzamanlama bařarımı zerindeki olumsuz etkisini gidermek amacıyla altı yeni yntem nerilmiřtir. nerilen yntemler iinde tm dęmlerin ortak bir saat hızında anlařtıkları SHUS protokolnn, teorik olarak algılayıcı aęının apının karekk cinsinden bir fonksiyon ile stten sınırlanan bir eřzamanlama hatasına sahip olduęu ve bu st sınırın dięer yntemlere gre olduka stn olduęu gsterilmiřtir. 20 MICAz dęm ieren bir deneysel dzenekte, SHUS protokolnn dięer yntemlerden uygulamada da olduka stn olduęu gzlemlenmiřtir. SHUS protokol, FTSP protokolnn eřzamanlama bařarımını en byk genel saat hatası cinsinden yaklaşık 20 kat arttırmıřtır. Benzetim sonuları da, SHUS protokolnn eřzamanlama bařarımının aęın apının bymesinden olduka az etkilendięi gereęini ortaya koymuřtur.

SHUS protokolnn en byk eksiklięi, ortak bir saat hızında uzlařmak iin geen zamanın uzun olmasıdır. Deneylerde, 20 algılayıcı dęmnn ortak bir saat hızında anlařabilmesi iin yaklaşık 5000 saniye gemiřtir. HSEP protokol ise, SHUS protokolne gre daha hızlı eřzamanlama saęlamaktadır. HSEP protokol yaklaşık 1200 saniyede aę genelinde sıkı bir eřzamanlama saęlamıřtır. Saat hızı uzlařma srecini hızlandırma, gelecek alıřmalar iin ucu aık bir problem olarak durmaktadır.

“Kimse geçmişini geri satın alabilecek kadar zengin değildir!”

Oscar WILDE

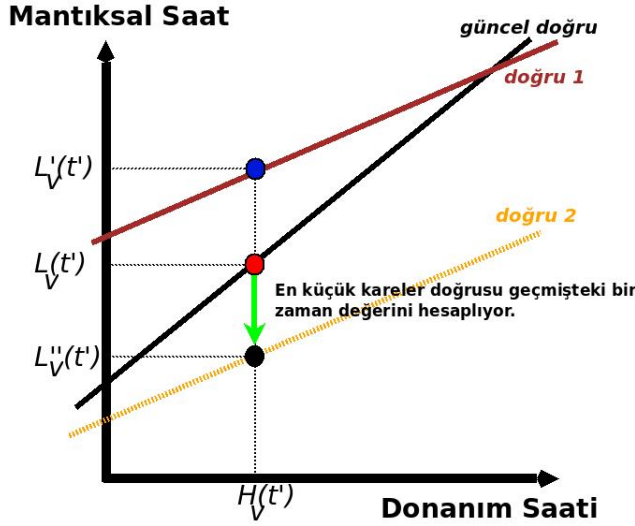
6 KABLOSUZ ALGILAYICI AĞLARINDA SAAT EŞZAMANLAMASI SÜRECİNDE SAATLERİN GERİ ALINMASINI ENGELLEYEN BİR YÖNTEM

Dağıtık sistemlere yönelik teorik saat eşzamanlama çalışmalarında, düğümlerin mantıksal saatleri *monoton artan* fonksiyonlar olarak varsayılmışlardır. Dağıtık sistemlerin bir alt kolu olan KAA’ndaki birçok saat eşzamanlama protokolü, algılayıcı düğümlerinin mantıksal saatleri arasında doğrusal bir ilişki olduğunu varsaymakta ve bu ilişkinin kurulması için en küçük kareler yöntemini kullanmaktadır. Ancak en küçük kareler yöntemi, saat eşzamanlama protokollerinin yürütümü sırasında mantıksal saatlerin geri alınmasına yol açabilmektedir. Bu durum, mantıksal saatlerin monoton artan fonksiyonlar olduğu yönündeki varsayımla çelişmektedir. Bu bölümde, en küçük kareler yöntemini kullanan KAA’ndaki saat eşzamanlama protokolleri için mantıksal saatlerin geri alınmasını engelleyecek bir yöntem önerilmektedir.

6.1 İlgili Çalışmalar

Saat eşzamanlaması problemi geleneksel dağıtık sistemler için derinlemesine çalışılmış bir konudur. Literatürde genel saat farkını üstten sınırlamaya yönelik birçok çalışma mevcuttur (Lundelius and Lynch, 1984; Halpern et al., 1985; Biaz and Welch, 2001; Patt-Shamir and Rajsbaum, 1994; Srikanth and Toueg, 1987). Fan ve Lynch (Fan and Lynch, 2006) birbirine komşu düğümlerin eşzamanlama hatasını gözetken ve düğümler arasındaki eşzamanlama hatalarının uzaklıklarının bir fonksiyonu olduğu *meyilli saat eşzamanlama* kavramını tanıtmıştır. Literatürde meyil özelliğine sahip algoritmaların geliştirilmesi ve analizine yönelik birçok çalışma bulunmaktadır (Fan et al., 2005; Locher and Wattenhofer, 2006; Lenzen et al., 2008; Pussente and Barbosa, 2009; Kuhn and Oshman, 2009; Lenzen et al., 2010; Kuhn et al., 2009; Kuhn et al., 2010). Bütün bu teorik çalışmalarda, mantıksal saatler monoton artan fonksiyonlar olarak tanımlanmıştır. Dolayısıyla, teorik çalışmalardaki genel varsayım mantıksal saatlerin sürekli ilerlediği ve hiçbir zaman geriye alınamayacağı yönündedir.

Literatürde KAA için geliştirilmiş ve düğümler arasındaki eşzamanlama hatasını en iyilemeyi hedefleyen birçok saat eşzamanlaması protokolü

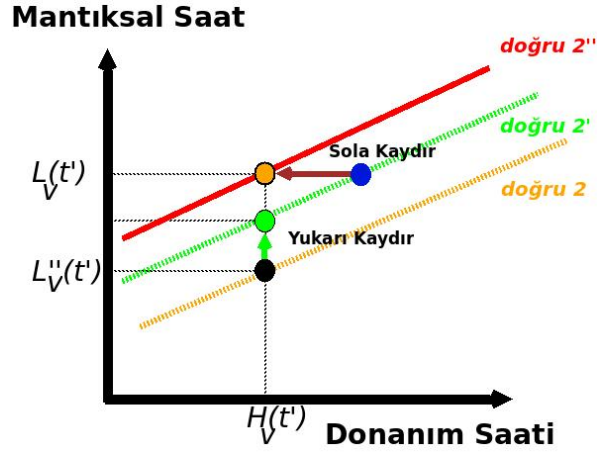


Şekil 6.1: Yeni bir eşzamanlama mesajı alındığında hesaplanan en küçük kareler doğruları.

bulunmaktadır (Elson et al., 2002; van Greunen and Rabaey, 2003; Ganeriwal et al., 2003; Dai and Han, 2004; Maróti et al., 2004; Sun et al., 2006; Kusy et al., 2006; Lenzen et al., 2009b; Schmid et al., 2009; Schmid et al., 2010a). Bu çalışmalar arasında yer alan (Elson et al., 2002; Maróti et al., 2004; Kusy et al., 2006; Lenzen et al., 2009b; Schmid et al., 2009; Schmid et al., 2010a) çalışmalarında, en küçük kareler yöntemi düğümlerin saatleri arasında doğrusal bir ilişki kurmak için kullanılmıştır. Ancak, bu protokollerin hiçbiri mantıksal saatlerin monoton artış gösteren bir fonksiyon olduğuna dikkat etmemekte ve yürütümleri sırasında mantıksal saatler geriye alınabilmektedir.

6.2 En Küçük Kareler Yönteminin Temel Eksikliği

Bu bölümde, KAA'nda genel geçer protokol olan ve en küçük kareler yöntemini kullanan FTSP protokolünün yürütümü sırasında mantıksal saatlerin nasıl geri alınabildiği gösterilecektir. t' anı, FTSP protokolünü yürüten v düğümünün yeni bir eşzamanlama mesajı aldığı an olsun. $L_v(t')$ mantıksal saat değeri, zaman etiketi tablosunda saklanmış olan N adet ikili kullanılarak hesaplanmış olan *güncel en küçük kareler doğrusu* kullanılarak hesaplanmaktadır. Mesaj alındığı anda, zaman etiketi tablosundaki en güncel ikili tablodan çıkarılmakta, yeni ikili tabloya eklenmekte ve *yeni en küçük kareler doğrusu* hesaplanmaktadır. Şekil 6.1'de gösterilen *doğru 1*'in hesaplanan yeni en küçük kareler doğrusu olduğu durum göz önüne alınsın. t' anında *doğru 1* kullanılarak hesaplanan $L'_v(t')$ değeri, $L_v(t')$ değerinden büyük olduğu için, bu durum bir soruna yol açmamaktadır.



Şekil 6.2: Güncel en küçük doğrusunun mantıksal saatlerin geri alınmaması için kaydırılması.

Şimdi, Şekil 6.1’de gösterilen *doğru 2*’nin hesaplanan yeni en küçük kareler doğrusu olduğu durum göz önüne alınsın. Bu durumda t' anı için hesaplanan mantıksal saat değeri $L_v''(t')$, $L_v(t')$ değerinden küçük olmaktadır. Bu durum mantıksal saatlerin monoton artan bir fonksiyon olduğu varsayımı ile çelişmektedir. Bu iki durum göz önüne alındığında, aşağıdaki olguya ulaşılabilir:

Olgu 6.1. *En küçük kareler yöntemi, kablosuz algılayıcı ağlarındaki saat eşzamanlama protokollerinin yürütümü sırasında mantıksal saatlerin geri alınmasına yol açabilir.*

6.3 Mantıksal Saatlerin Geri Alınmasını Engelleyen Bir Yöntem

Bu bölümde, mantıksal saatlerin geri alınmasını engellemek için yeni hesaplanan en küçük kareler doğrusunun önceki en küçük kareler doğrusu ile t' anında kesişmesi için kaydırılması önerilmektedir. Şekil 6.1’de gösterilen *doğru 2*’nin aşağıda gösterilen denklemi göz önüne alınsın:

$$\begin{aligned} L_v''(t) &= \alpha'' + \beta'' t \\ &= \bar{Y}'' + \beta'' (H_v(t) - \bar{x}''). \end{aligned} \quad (6.1)$$

$L_v''(t') < L_v(t')$ eşitsizliğinin sağlanması nedeniyle mantıksal saat t' anında geri alınmaktadır. Bunu engellemek için, öncelikle hesaplanan iki mantıksal saat değeri arasındaki fark aşağıdaki gibi hesaplanmaktadır:

$$\Delta_v = L_v(t') - L_v''(t'). \quad (6.2)$$

Bu değer hesaplandıktan sonra, yeni en küçük kareler doğrusu (*doğru 2*) öncelikle y ekseninde $\Delta_v/2$ kadar yukarı ötelenerek *doğru 2'* elde edilmektedir. Daha sonra *doğru 2'* x ekseninde $\Delta_v/2\beta''$ kadar sola ötelenerek *doğru 2''* elde edilmektedir. Bu adımlar Şekil 6.2'de gösterilmektedir. Dolayısıyla, mantıksal saatin hesaplanması için kullanılacak yeni denklem, *doğru 2''* 'nin denklemi olarak 6.3'te verilmiştir:

$$L_v(t) = \left(\bar{Y}'' + \frac{\Delta_v}{2} \right) + \beta'' \left(H_v(t) - \left(\bar{x}'' - \frac{\Delta_v}{2\beta''} \right) \right). \quad (6.3)$$

Yukarıdaki eşitlikten görüleceği gibi, tüm bu kaydırma işlemleri kaydırılmamış *doğru 2'* nin üzerindeki (\bar{x}'', \bar{Y}'') noktasını kaydırılmış *doğru 2''* üzerindeki $(\bar{x}'' - \Delta_v/2\beta'', \bar{Y}'' + \Delta_v/2)$ noktasına taşımakla denktir. Önerilen bu yöntem, yeni bir eşzamanlama mesajı geldiğinde en küçük kareler yöntemine yapılacak küçük bir değişikliktir ve işlem adımları Algoritma 6.1'de gösterilmiştir.

Algoritma 6.1 En küçük kareler yöntemi için önerilen değişiklik.

```

1 □ Yeni bir  $(x, y)$  eşzamanlama noktası alındığında
2    $eskiZaman \leftarrow L_v$ 
3    $(x, y)$  ikilisini zaman etiketi tablosuna ekle
4   en küçük kareler doğrusunu yeniden hesapla
5    $yeniZaman \leftarrow L_v$ 
6    $\Delta_v \leftarrow (eskiZaman - yeniZaman)$ 
7   eğer  $\Delta_v > 0$ 
8     en küçük kareler doğrusunu kaydır
9   soneğer

```

6.4 Deneysel Sonuçlar

Bu bölümde, bölüm 5.7'de daha önce tanıtılan deney düzeneği kullanılarak, doğrusal ilingiyeye sahip 20 adet MICAz düğümü üzerinde önerilen yöntem için gözlemlenen deneysel sonuçlar sunulmaktadır. Deneyler için, TinyOS 2.1.1 ile gelen FTSP protokolünün kodunda en küçük kareler yöntemini uygulandığı noktalar Algoritma 6.1 ile değiştirilmiş ve değiştirilmiş FTSP deneysel olarak sınanmıştır. Karşılaştırma için orjinal FTSP protokolü de deneysel düzeneğe çalıştırılmıştır. Tüm deneysel yürütümler yaklaşık 30000 saniye sürmüştür ve ağ genelinde ilk eşzamanlanmanın sağlanması için yaklaşık 5000 saniye geçmiştir.

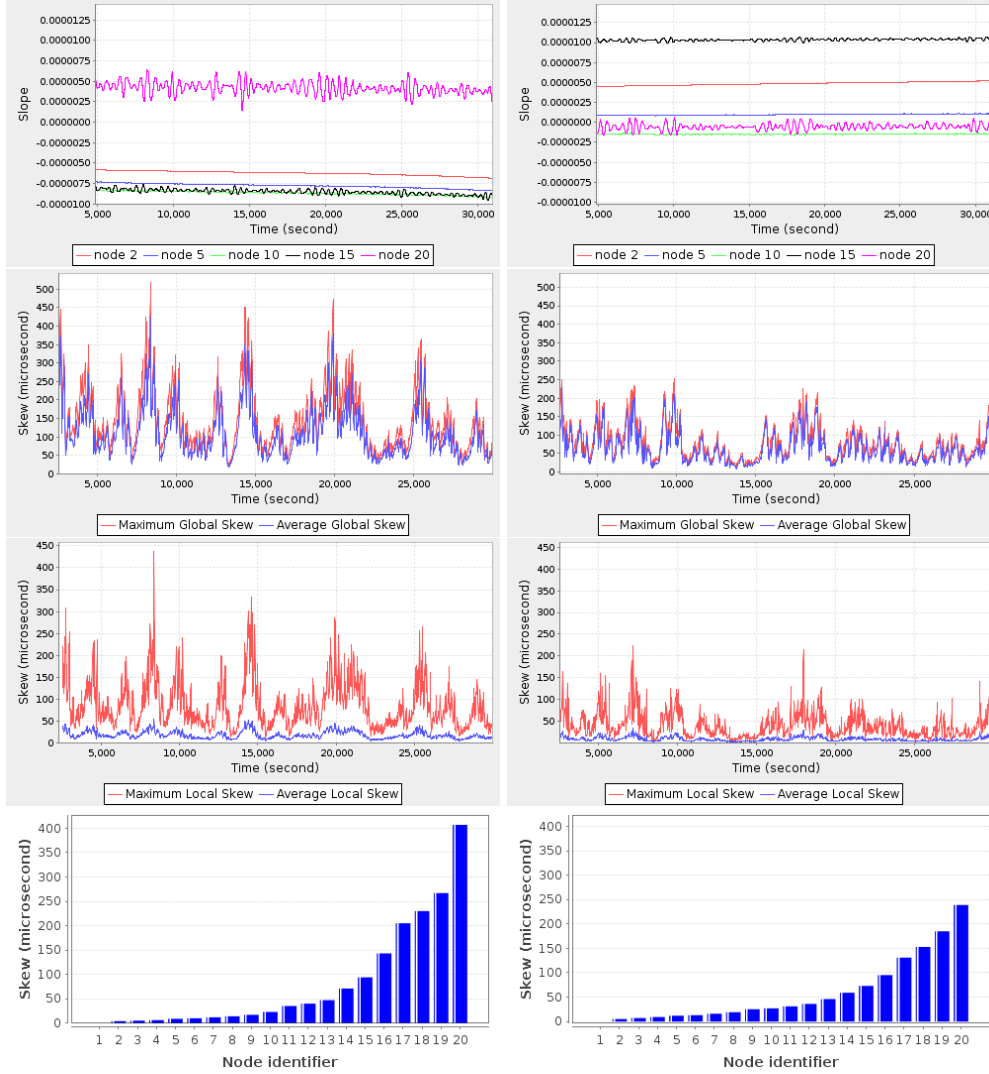
Bu nedenle, ilk 5000 saniye boyunca meydana gelen eşzamanlama hataları değerlendirilmede göz önüne alınmamıştır.

Deneyler süresince, FTSP protokolünün yeni bir eşzamanlama mesajı alındığında birçok kere mantıksal saatleri geriye aldığı gözlemlenmiştir. Önerilen değişikliklerin yapıldığı değiştirilmiş FTSP protokolünde ise hiçbir zaman mantıksal saatler geriye alınmamıştır. Buna ek olarak, değiştirilmiş FTSP orjinal FTSP protokolüyle kıyaslandığında başka üstünlükler de göstermiştir.

Şekil 6.3'de FTSP ve değiştirilmiş FTSP için gözlemlenen en küçük kareler doğrusu eğimlerini göstermektedir. Görüldüğü gibi dayanak düğümü olan 1. düğümden uzaklaştıkça, gözlemlenen eğim değerleri kararsızlıklar göstermektedir. Bu durum, her sekmede eşzamanlama bilgisinin hatasının büyümesinden ve uzak düğümlerin eşzamanlama bilgisini daha büyük hatayla almasından kaynaklanmaktadır (Lenzen et al., 2009b). Ancak, değiştirilmiş FTSP protokolünde uzak düğümlerin eğimlerinin kararsızlığı orjinal FTSP protokolünün eğiminin gösterdiği kararsızlıktan çok daha azdır. Bunun nedeni, saatlerin geri alınmasının saatlerin geriye zıplamasını engellemesi ve bunun sonucunda özellikle uzaktaki düğümlerin en küçük kareler doğrularının anlık büyük değişimler göstermesinin önüne geçilmesidir. Önerilen yöntemle, eski en küçük kareler doğrusundan yeni en küçük kareler doğrusuna geçiş yumuşak bir şekilde yapılmakta, bu da eğimlerin daha kararlı olmasını sağlamaktadır.

Önerilen yöntem saatlerin geri alınmasını engellemesi dışında genel ve yerel saat farklarını da hatırı sayılır bir şekilde düşürmektedir. 6.3 şekli orjinal ve değiştirilmiş FTSP için eşzamanlama hatalarını da göstermektedir. Orjinal FTSP ile en büyük genel ve en büyük ortalama genel saat farkı değerleri sırasıyla $518 \mu s$ ve $422 \mu s$ olarak gözlemlenmiştir. Değiştirilmiş FTSP protokolü ile bu değerler sırasıyla $252 \mu s$ ve $230 \mu s$ değerlerine inmiştir. Orjinal FTSP protokolü ile en büyük yerel ve en büyük ortalama yerel saat farkı olarak $437 \mu s$ ve $55 \mu s$ değerleri gözlemlenmiştir. Bu değerler değiştirilmiş FTSP protokolü için sırasıyla $224 \mu s$ ve $34 \mu s$ olarak gözlemlenmiştir. Önerilen yöntem ile kazanılan eğim kararlılığı, FTSP protokolünde genel ve yerel saat farklarında azalmaya yol açmıştır.

Şekil 6.3, dayanak düğümü ile diğer düğümler arasındaki en büyük saat farkı değerlerini de göstermektedir. Dayanak düğümü ile kimliği 2 ile 14 arasındaki düğümler arasında gözlemlenen en büyük saat farkı, orjinal ve değiştirilmiş FTSP protokolleri için oldukça benzerdir. Ancak, kimlikleri 15 ve 20 arasında olan düğümlerin eşzamanlama hatası, orjinal FTSP protokolünde değiştirilmiş FTSP protokolü ile kıyaslandığında oldukça büyüktür. Deneysel sonuçlar, önerilen



Şekil 6.3: FTSP (sol sütun) ve değiştirilmiş FTSP (sağ sütun) için sırasıyla en küçük kareler doğrusunun eğimi, genel saat farkı değerleri, yerel saat farkı değerleri, dayanak düğümü ve diğer düğümler arasında gözlemlenen en büyük saat farkı değerleri. Eğim değerleri, gerçek değerlerden 1.0 çıkartılmış şekilde gösterilmektedir.

yöntemin sağladığı eğim kararlılığı sayesinde, uzaktaki düğümlerle dayanak düğümü arasında görülen saat farkı değerlerinin de hatırı sayılır şekilde azaldığını göstermiştir.

6.5 Tartışma

Bu çalışmada, KAA’nda yaygın olarak uygulanan en küçük kareler yönteminin temel bir eksikliği ortaya koyulmuştur. FTSP protokolünün yürütümü sırasında mantıksal saatlerin en küçük kareler yöntemi nedeniyle geriye alınabildiği gösterilmiştir. Bu durumu ortadan kaldırmak için, önceki en küçük kareler doğrusundan güncel en küçük kareler doğrusuna yumuşak bir geçiş sağlayan öteleme yöntemi önerilmiştir. Önerilen yöntemin uygulamada üstün olduğunu göstermek için, üzerinde değişiklikler yapılan FTSP protokolü 20 adet algılayıcı düğümü içeren bir test düzeneğinde sınanmıştır. Deneysel sonuçlar, önerilen yöntemin mantıksal saatlerin geri alınmasını engellediği gibi genel ve yerel saat farkı değerlerini de düşürdüğünü göstermiştir.

Algılayıcı düğümlerindeki kristal salınıcılarının frekanslarının kararlı olmadığı ve zamanla değişebildiği göz önüne alınması gereken önemli bir ayrıntıdır. Sıcaklık değişimi gibi etkenler kısa dönemli frekans kararsızlıklarına, salınıcının zaman içinde aşınması ise uzun dönemli frekans kararsızlıklarına yol açmaktadır (Elson et al., 2002). Bir algılayıcı düğümünün donanım saatinin frekansının anlık bir sıcaklık değişimi nedeniyle değiştiği ve yeni bir eşzamanlama mesajı alınması sonucunda hesaplanan en küçük kareler doğrusunun, bir öncekinden daha küçük bir mantıksal saat değeri ortaya koyduğu bir durum göz önüne alınsın. Önerilen yönteme göre, güncel en küçük kareler doğrusu saatlerin geri alınmaması için ötelenektir. Ancak, düğümün donanım saatinin hızı değiştiği için, bu noktadan sonra yeni bir eşzamanlama mesajı alınmasıyla hesaplanan her bir yeni en küçük kareler doğrusu ötelenmek zorunda kalacaktır. Bu durum gözlemlenen saat farkını arttırabilir, çünkü mantıksal saatlerin geriye alınmasını engellemek için en küçük kareler doğrusu ötelenerek daha doğru bir mantıksal saat değerinden ödün verilmektedir. Mantıksal saat farkını düşürmek ve mantıksal saatlerin geriye alınmasını engellemek konusunda bir ortayol bulmak için, bir düğüm hesapladığı her yeni en küçük kareler doğrusu için üst üste kaç defa öteleme yaptığını sayabilir. Eğer sayılan değer önceden belirlenmiş bir değerden büyükse, düğüm mantıksal saatlerin geri alınmasına izin vererek saat farkını düşürebilir. Ancak bu durum, mantıksal saatlerin monoton artan fonksiyon oldukları gerçeğini yadsımak demektir.

Zaman süreksizliklerini ortadan kaldırmak bu çalışmanın ana hedefi olmasa da, mantıksal saatlerin geriye alındığı zamanlarda önerilen yöntemin zaman süreksizliklerini de ortadan kaldırdığı açık olarak görülebilir. Ancak mantıksal saatler ileri alındığında da zaman süreksizlikleri ortaya çıkabilir. Önerilen öteleme yöntemi yeni bir eşzamanlama mesajı alındığında hesaplanan en küçük kareler doğrusunun ortaya koyduğu mantıksal saat değeri, önceki en küçük kareler doğrusu ile hesaplanmış olan mantıksal saat değerinden bir eşik değerini geçecek şekilde büyük olduğu durumda da uygulanmış ve saat farkı açısından düşük bir başarıma sahip olduğu gözlemlenmiştir. Bunun nedeni ise, zaman süreksizliklerini engellemek için en küçük kareler doğrusu ötelenerek daha doğru bir mantıksal saat değerinden ödün verilmesidir. İleri yöndeki mantıksal saat zıplamalarını engelleyen yeni bir yöntemin geliştirilmesi, gelecek çalışmalar için ucu açık bir problemidir.

*“Çünkü kandırılmışlar arasında kavrayışı sağlam bir adam,
saat kulelerinin hepsi yanlış ayarlanmış bir kentte,
saati doğru işleyen birine benzer.
Tek o bilir doğru zamanı: Ama bunun ne yararı dokunur ona?
Herkes kentin yanlış saatine uyar,
doğru zamanı yalnız onun saatinin gösterdiğini bilenler bile.”*
Arthur SCHOPENHAUER

7 KABLOSUZ ALGILAYICI AĞLARI İÇİN DIŞSAL MEYİLLİ SAAT EŞZAMANLAMASI

KAA için tasarlanmış birçok saat eşzamanlama protokolü, ağ içerisindeki herhangi iki düğüm arasındaki eşzamanlama hatasını (genel saat farkı) eniyilemeye çalışmaktadır. Bu protokollerin yürütümünde, düğümler komşularının zaman bilgisini gözetmemektedirler. Bunun sonucunda düğümler komşuları ile kötü bir şekilde eşzamanlanmaktadır.

KAA’nda hedef takibi gibi uygulamalar için komşu düğümler arasındaki eşzamanlama hatasını (yerel saat farkı) eniyilemek çok önemli olabilmektedir. Birbirlerine d metre aralıklarla yerleştirilmiş düğümlerin, bir nesnenin hızını metre/saniye cinsinden merkezi bir istasyona göndermeleri gerektiği bir senaryo göz önüne alınsın. Herhangi bir algılayıcı düğümü hedef nesneyi algıladığı zaman, bu olayın tespit anını kaydetsin ve birbirine komşu düğümler kendi tespit anlarını birbirleri ile değiş tokuş etsinler. Düğümler nesnenin hızını tahmin edebilmek için $v = d/\Delta$ eşitliğini kullansınlar. Burada Δ nesnenin tespit edilme zamanları arasındaki farktır ve Δ değerinin saniye cinsinden geçen gerçek zaman olarak hesaplanabilmesi için düğümlerin Eşgüdümlü Evrensel Zaman (EEZ)²¹ gibi bir dışsal kaynağa eşzamanlanmaları gerekmektedir. Görüleceği gibi, birbirine komşu düğümler arasında eşzamanlama hatası küçük olursa, Δ değeri ve bunun sonucunda nesnenin hızı çok daha doğru tahminlenebilmektedir.

Gradient Time Synchronization Protocol (GTSP) (Sommer and Wattenhofer, 2009) protokolü, KAA’nda yerel saat farkını eniyilemeyi hedefleyen ilk ve tek protokoldür. GTSP protokolünde algılayıcı düğümleri ortak bir saat hızı ve saat değerinde bir uzlaşma algoritması yürüterek anlaşılmaktadır. GTSP protokolünü yürüten düğümler bir dayanak düğüme ihtiyaç duymazlar çünkü düğümler eşten eşe mantığıyla birbirlerine eşzamanlanmaktadır. Bu özellik GTSP için

²¹İngilizcesi: *Coordinated Universal Time (UTC)*

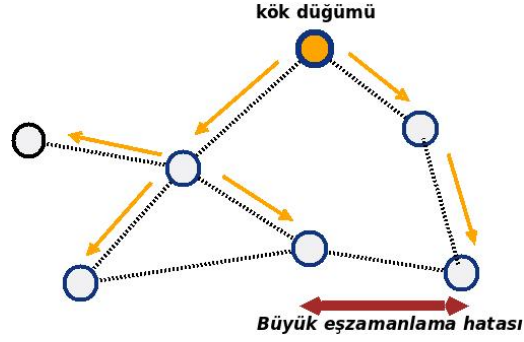
temel bir eksikliğe yol açmaktadır: GTSP dışsal bir saat kaynağına eşzamanlama (dışsal eşzamanlama) sağlayamamaktadır (Schmid et al., 2009). KAA'nda dışsal eşzamanlama sağlamak için uygulanan genel yöntem, bir dayanak düğümünü dışsal bir zaman kaynağına eşzamanlamak ve ağdaki geri kalan düğümleri içsel eşzamanlama ile dayanak düğümüne eşzamanlamaktır. GTSP, algılayıcı düğümlerini bir dayanak düğümüne eşzamanlayamadığı için, az önce bahsedilen uygulamada olduğu gibi hem dışsal eşzamanlamaya hem de yerel saat farkı eniyilemesine ihtiyaç duyan uygulamalar tarafından kullanılamamaktadır.

Bu bölümde, algılayıcı düğümlerini bir dayanak düğümüne eşzamanlarken aynı zamanda birbirine komşu düğümler arasındaki saat farkını da eniyileyen yeni bir yöntem tanıtılmaktadır. Bu bağlamda, bir dayanak düğümünden sel yöntemi ile ağa yayılan zaman bilgisini ve GTSP protokolündeki uzlaşma algoritmasını birlikte kullanarak, algılayıcı düğümlerinin dayanak düğümünün saat hızı ve saat değerinde uzlaşmalarını sağlayan Dışsal Meyilli Saat Eşzamanlaması Protokolü (DMSEP) sunulmaktadır. DMSEP protokolü, algılayıcı düğümlerini hem birbirleri ile hem de dayanak düğümü ile eşzamanladığı için GTSP protokolünden farklı olarak dışsal eşzamanlama için kullanılabilir.

7.1 İlgili Çalışmalar

Literatürde, dağıtık sistemlerde herhangi bir ağdaki genel saat farkını alttan ve üstten sınırlamaya ve optimal genel saat farkını sağlayan algoritmaların geliştirilmesine yönelik teorik çalışmalar mevcuttur (Lundelius and Lynch, 1984; Halpern et al., 1985; Biaz and Welch, 2001; Patt-Shamir and Rajsbaum, 1994; Srikanth and Toueg, 1987). Uygulamada, KAA için geliştirilmiş birçok protokol benzer şekilde genel saat farkını eniyilemeye çalışmaktadır (Elson et al., 2002; van Greunen and Rabaey, 2003; Ganeriwal et al., 2003; Dai and Han, 2004; Maróti et al., 2004; Sun et al., 2006; Kusy et al., 2006; Lenzen et al., 2009b; Schmid et al., 2009; Schmid et al., 2010a). Bu çalışmaların temel eksikliği, sunulan protokollerin yürütümünde birbirine komşu düğümler arasında büyük eşzamanlama hatalarının gözlemlenebilmesidir. Örneğin FTSP (Maróti et al., 2004) protokolünde, dayanak düğümünün kendi saat bilgisini ağa sel yöntemi ile yayması nedeniyle oluşan tasarsız ağacın farklı dallarında bulunan birbirine komşu düğümler, aldıkları eşzamanlama bilgisi kendilerine farklı yollardan farklı hatalarla geldiği için kötü bir biçimde eşzamanlanmaktadır. Bu durum Şekil 7.1'de gösterilmektedir.

Fan ve Lynch (Fan and Lynch, 2006) *meyilli saat eşzamanlaması* kavramını



Şekil 7.1: Birbirine komşu düğümler, eşzamanlama bilgisini farklı yollardan aldıkları için, bu düğümler arasında büyük eşzamanlama hataları görülebilir.

ortaya koymuşlar ve birbirine komşu düğümler arasındaki eşzamanlama hatasının eniyilenmesinin önemine dikkat çekmişlerdir. Meyilli saat eşzamanlamasında, düğümler arasındaki eşzamanlama hatası düğümlerin arasındaki uzaklığın bir fonksiyonudur. Dolayısıyla, birbirine yakın düğümler, uzaktakilere göre daha iyi eşzamanlanmaktadır. Literatürde optimal yerel saat farkını yakalayan teorik saat eşzamanlama algoritmaları geliştirme ve bu algoritmaları analiz etmeye yönelik birçok çalışma yer almaktadır (Fan et al., 2005; Locher and Wattenhofer, 2006; Lenzen et al., 2008; Pussente and Barbosa, 2009; Kuhn and Oshman, 2009; Lenzen et al., 2010; Kuhn et al., 2009; Kuhn et al., 2010). Ancak uygulamada, KAA için yerel saat farkını eniyileştirmeyi hedefleyen tek protokol Gradient Time Synchronization Protocol (GTSP) (Sommer and Wattenhofer, 2009) protokolüdür. GTSP protokolünde, düğümler komşularından aldıkları zaman bilgisinin ortalamasının alınmasına dayalı basit bir algoritmayı yürüterek²², ortak bir saat hızı ve saat değerinde anlaşmaktadırlar. Benzetimlerde ve deneysel gözlemlerde, GTSP protokolü FTSP protokolüne göre oldukça iyi bir yerel saat farkı değeri ortaya koysa da, algılayıcı düğümlerin bir dayanak düğümlerine eşzamanlanırken aynı anda komşularıyla olan saat farkını eniyileyip eniyilemeyeceği sorununun açık bırakılmaktadır.

7.2 GTSP Protokolüne Genel Bakış

Bu bölümde, sözde kodu Algoritma 7.1’de verilen GTSP protokolüne bir genel bakış sunulacaktır. GTSP protokolünü yürüten her $v \in V$ düğümü, komşularından gelen zaman bilgisinin takibini yapabilmek için bir komşu veri havuzuna sahiptir. v düğümü bu veri havuzunu kullanarak herhangi bir komşusunun göreceli donanım saati hızını ve mantıksal saat değerini tahmin edebilmektedir. Algılayıcı düğümlerinin depolama kısıtları nedeniyle, veri havuzunun büyüklüğü

²²Algoritma daha önce (Schenato and Gamba, 2007; Schenato and Fiorentin, 2011) çalışmasında tanıtılmıştır.

Algoritma 7.1 v düğümü için GTSP sözde kodu.

- 1: **İkleme**
 - 2: veri havuzunu temizle ve $\forall u \in \mathcal{N}_v : h_v^u \leftarrow 1, l_u \leftarrow 1$
 - 3: $l_v \leftarrow 1, \theta_v \leftarrow 0$
 - 4: B periyodu ile periyodik bir zamanlayıcı başlat
 - 5:
 - 6: $\square < H_u, L_u, l_u >$ **mesajı alındığında**
 - 7: (H_v, H_u) ikilisini sakla ve h_v^u 'yi tahmin et
 - 8: (h_v^u, l_u, L_u) üçlüsünü sakla
 - 9: l_v ve θ_v değerlerini hesapla
 - 10
 - 11: \square **Zamanlayıcı zamanaşımında**
 - 12: **tümegönder** $< H_v, L_v, l_v >$
-

ve bir algılayıcı düğümünün takibini yapabileceği komşu sayısı sınırlıdır. v düğümü açıldığı anda, veri havuzu temizlenir ve takip yapılacak her $\forall u \in \mathcal{N}_v$ komşusunun göreceli donanım saat hızı h_v^u ve mantıksal saatinin hız çarpanı l_u , 1 değeri ile ilklenir (2. Satır). Buna ek olarak v düğümü kendi mantıksal saatinin hız çarpanı l_v 'yi 1 ile ve öteleme değeri θ_v 'yi 0 ile iklemektedir (3. Satır). Dolayısıyla, v düğümünün mantıksal saat değeri, l_v ve θ_v parametreleri komşulardan alınan bir eşzamanlama mesajı sonucu değiştirilmediği sürece donanım saatinin değerine eşittir. Son olarak, donanım saati her B birim ilerlediğinde zaman aşımını uğrayacak bir zamanlayıcı kurulmaktadır (4. Satır).

Herhangi bir $u \in \mathcal{N}_v$ komşusundan $< H_u, L_u, l_u >$ mesajı alındığında (6. Satır), v düğümü alınan donanım saat değeri H_u 'yu ve kendi donanım saat değeri H_v 'yi (H_v, H_u) ikilisi şeklinde o komşu için veri havuzunda ayrılmış en güncel N ikiliyi saklama kapasitesine sahip tabloda saklamaktadır. v düğümü kendi donanım saati ile u düğümünün donanım saati arasında doğrusal bir ilişki olduğunu varsayarak tahmini doğrusal bağlanım doğrusunu hesaplayabilmek için, alınan ikililer üzerinde en küçük kareler yöntemini uygular. Hesaplanan en küçük kareler doğrusunun eğimi, h_u/h_v göreceli saat hızının h_v^u ile gösterilen bir tahminidir (7. Satır). Tahmini göreceli saat hızı h_v^u , alınan hız çarpanı l_u ve mantıksal saat değeri L_u , (h_v^u, l_u, L_u) üçlüsü şeklinde veri havuzunda o komşu için kaydedilir (8. Satır). Bu üçlüyü kullanarak, v düğümü u düğümünün herhangi bir t anındaki tahmini mantıksal saat değeri olan $L_v^u(t)$ 'yu, saklanan L_u değerini kendi mantıksal saatine göre $h_v^u \cdot l_u$ hızında ilerleterek hesaplayabilir.

v düğümü her $u \in \mathcal{N}_v$ komşusuna ait saklanan h_v^u ve l_u değerlerini kullanarak, 3.3 eşitliğindeki mantıksal saatinin hız çarpanını ve öteleme değerini aşağıda gösterildiği gibi hesaplamaktadır (9. Satır):

$$l_v(t^+) = \frac{l_v(t) + \sum_{u \in \mathcal{N}_v} h_v^u(t) \cdot l_u(t)}{|\mathcal{N}_v| + 1}, \quad (7.1)$$

$$\theta_v(t^+) = \theta_v(t) + \frac{\sum_{u \in \mathcal{N}_v} (L_v^u(t) - L_v(t))}{|\mathcal{N}_v| + 1}. \quad (7.2)$$

Yukarıdaki eşitliklerdeki t^+ , güncelleme işleminden hemen sonraki gerçek zamanı temsil etmektedir. Eğer iletişim çizgesi G sıkı bağlı bir şekilde kalırsa, tüm düğümlerin bu hesaplama yöntemiyle ortak bir saat değerinde ve saat hızında anlaşacakları (Sommer and Wattenhofer, 2009) ve (Schenato and Fiorentin, 2011) çalışmalarında gösterilmiştir.

Son olarak, kurulan zamanlayıcı her zaman aşımına uğradığında, v düğümü kendi donanımsal saat değerini, mantıksal saat değerini ve hız çarpanını içeren bir eşzamanlama paketini tümegöndermektedir (11-12 Satırları).

7.2.1 GTSP protokolünün temel eksikliği

KAA'nda dışsal saat eşzamanlaması, genel olarak aşağıdaki adımları uygulayarak sağlanabilir. Öncelikle bir dayanak düğümü EEZ gibi bir dışsal zaman kaynağına KKS donanımı yardımı ile eşzamanlanmaktadır. İkinci adım olarak geri kalan algılayıcılar dayanak düğümüne bir içsel eşzamanlama algoritmasıyla eşzamanlanmaktadır.

GTSP protokolünün temel eksikliği, düğümler bir dayanak düğümüne eşzamanlanmadığı için dışsal saat eşzamanlaması için kullanılamamasıdır. Bu eksikliği bir örnekle açıklamak için, algılayıcı düğümlerinin ref dayanak düğümünün donanım saatine eşzamanlanması gerektiği bir durum göz önüne alınsın. Daha önce bahsedildiği gibi, GTSP protokolünde tüm $k \in V$ algılayıcı düğümleri aşağıda gösterilen ortak bir saat hızında anlaşmaktadırlar:

$$\lim_{t \rightarrow \infty} (h_k(t) \cdot l_k(t)) = hız. \quad (7.3)$$

Ancak üzerinde anlaşılan ortak saat hızı $hız$, dayanak düğümünün donanım saatinin hızı olan h_{ref} değerinden farklı olabilir. Bu durum düğümlerin üzerinde uzlaştıkları ortak saat değerinin de dayanak düğümünün donanım saati değeri olan H_{ref} değerinden farklı olması sonucunu doğurur. Sonuç olarak GTSP protokolü dışsal eşzamanlama için düğümleri bir dayanak düğümüne de eşzamanlayacak bir

Algoritma 7.2 v düğümü için ref sabit dayanak düğümüne sahip DMSEP sözde kodu.

```

1: İkleme
2: veri havuzunu temizle ve  $\forall u \in \mathcal{N}_v : h_v^u \leftarrow 1, l_u \leftarrow 1$ 
3:  $l_v \leftarrow 1, \theta_v \leftarrow 0$ 
4:  $l_v^{ref} \leftarrow 1, \Delta_v^{ref} \leftarrow 0, seq_v \leftarrow 0$ 
5:  $B$  periyodu ile periyodik bir zamanlayıcı başlat
6:
7:  $\square < H_u, L_u, l_u, l_u^{ref}, \Delta_u^{ref}, seq_u >$  mesajı alındığında
8:  $(H_v, H_u)$  ikilisini sakla ve  $h_v^u$ 'yi tahmin et
9:  $(h_v^u, l_u, L_u)$  üçlüsünü sakla
10:  $l_v$  ve  $\theta_v$  değerlerini hesapla
11: eğer  $seq_v < seq_u$ 
12:    $l_v^{ref} \leftarrow l_u^{ref}$ 
13:    $\Delta_v^{ref} \leftarrow \Delta_u^{ref}$ 
14:    $seq_v \leftarrow seq_u$ 
15: soneğer
16:
17:  $\square$  Zamanlayıcı zamanışımında
18: eğer  $(v = ref)$ 
19:    $l_v^{ref} \leftarrow l_v$ 
20:    $\Delta_v^{ref} \leftarrow H_v - L_v$ 
21:    $seq_v \leftarrow seq_v + 1$ 
22: soneğer
23: tümegönder  $< H_v, L_v, l_v, l_v^{ref}, \Delta_v^{ref}, seq_v >$ 

```

Algoritma 7.3 DMSEP protokolünü yürüten v düğümü için dayanak düğümünün saatini döndüren dayanakSaati arayüzü.

```

1: dayanakSaati()
2:  $L_v + \Delta_v^{ref}$  değerini döndür.

```

yönteme ihtiyaç duymaktadır.

7.3 Dışsal Meyilli Saat Eşzamanlaması Protokolü

Bu bölümde, bir dayanak düğümünden sel yöntemi ile ağa yayılan zaman bilgisini ve GTSP protokolündeki uzlaşma algoritmasını birlikte kullanarak algılayıcı düğümlerinin önceden belirlenmiş ref dayanak düğümünün donanım saat hızı ve donanım saat değerinde uzlaşmalarını sağlayan, Dışsal Meyilli Saat Eşzamanlaması Protokolü (DMSEP) sunulmaktadır. DMSEP protokolünün GTSP protokolüne oldukça benzer olan sözde kodu, Algoritma 7.2’de gösterilmektedir.

DMSEP protokolünü yürüten her algılayıcı düğümü, GTSP protokolündeki değişkenler dışında ref dayanak düğümünden en son alınan zaman bilgisinin

takibini yapabilmek için üç ek değişken daha barındırmaktadır. l_v^{ref} değişkeni dayanak düğümünden en son alınan hız çarpanını, Δ_v^{ref} değişkeni dayanak düğümünün donanım saati ile mantıksal saati arasındaki en son farkı (zaman öteleme değerini) ve seq_v ise dayanak düğümünden en son alınan sıra numarasını saklamaktadır. GTSP protokolündeki ilkleme adımlarına ek olarak, sistem açıldığında l_v^{ref} değişkeni 1'le, Δ_v^{ref} ve seq_v değişkenleri ise 0'la ilklenmektedir (4. Satır).

DMSEP protokolünde eşzamanlama mesajları dayanak düğümünün hız çarpanı, zaman öteleme değeri ve sıra numarasını da taşıyacak şekilde genişletilmiştir. Yeni bir eşzamanlama mesajı alındığında (7. Satır), öncelikle GTSP protokolündeki uzlaşma algoritması işletilir (8-10 Satırları). Daha önce bahsedildiği gibi, tüm düğümler ortak bir mantıksal saat hızı ve mantıksal saat değerinde bu algoritma sayesinde anlaşmaktadırlar. Dolayısıyla, uzlaşma sağlandıktan sonraki her t zamanı için, ref dayanak düğümü ve v düğümü için 7.4 eşitliği sağlanmaktadır:

$$h_{ref}(t).l_{ref}(t) = h_v(t).l_v(t). \quad (7.4)$$

Bu eşitlikten, 7.5 eşitliği elde edilmektedir:

$$h_{ref}(t) = h_v(t). \frac{l_v(t)}{l_{ref}(t)} \quad (7.5)$$

Bu gerçeği kullanarak, v düğümünün kendi mantıksal saatini dayanak düğümünün donanım saat hızında ilerletebilmesi için, kendi mantıksal saatini 3.3 eşitliği yerine 7.6 eşitliği kullanarak hesaplamaktadır:

$$L_v(t) = \int_0^t h_v(\tau) \frac{l_v(\tau)}{l_{ref}(\tau)} d\tau + \theta_v(t). \quad (7.6)$$

Sonuç olarak, DMSEP protokolü GTSP protokolündeki uzlaşma algoritmasını değiştirmeden uygulamakta, ancak mantıksal saatleri ref dayanak düğümünün donanım saatinin hızında ilerletmektedir.

Eğer alınan eşzamanlama mesajı daha büyük bir sıra numarası taşıyorsa (11. Satır), bu durum dayanak düğümünün kendi hız çarpanını ve zaman öteleme değerini yakın zamanda ağa yaymaya başladığını göstermektedir. Bu

durumda v düğümü l_v^{ref} , Δ_v^{ref} ve seq_v değişkenlerinin değerlerini alınan değerlerle güncellemektedir (12-15 Satırları).

Kurulan zamanlayıcı bir zaman aşımına uğradığında (17. Satır), eğer düğüm dayanak düğümü ise, l_v^{ref} değişkenine düğümün kendi hız çarpanı atanır, Δ_v^{ref} değişkeni düğümün donanım saatinden mantıksal saati çıkartılarak hesaplanır ve sıra numarası bir arttırılmaktadır (18-22 Satırları). Sıra numarasının bir arttırılması, dayanak düğümünün yeni bir sel turu başlattığının işaretidir. Eğer düğüm dayanak düğümü değilse, bu değişkenlerin değerleri değiştirilmez. Her düğüm kendi donanım saatini, mantıksal saatini, hız çarpanını, dayanak düğümünden en son aldığı hız çarpanını, zaman öteleme değerini ve sıra numarasını tümegöndermektedir (23. Satır).

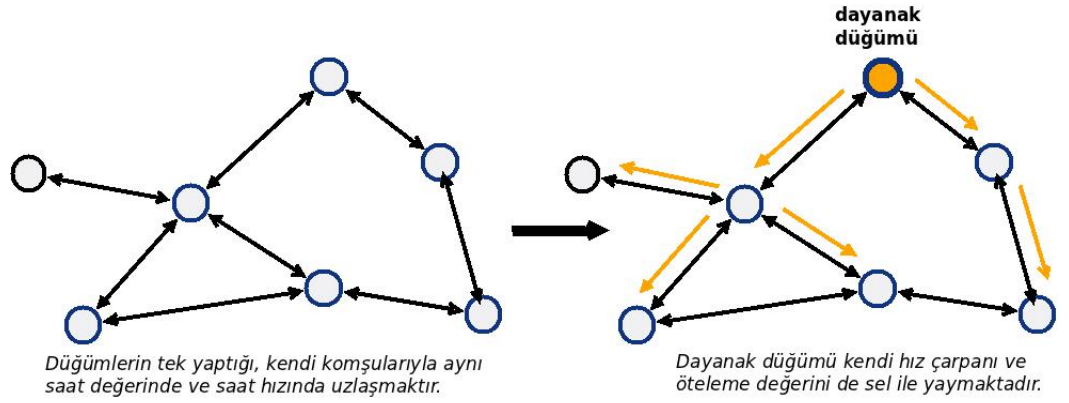
Bu yürütüm ile dayanak düğümü dahil tüm düğümler uzlaşma algoritmasına katılırken, bir yandan da dayanak düğümünün hız çarpanı ve zaman öteleme değerini ağa sel ile yayılmaktadır. Uzlaşma gerçekleştiği andan itibaren, tüm düğümlerin mantıksal saatleri dayanak düğümünün donanım saat hızında ilerlemekte ve aynı değeri göstermektedirler. Buradaki sorun, mantıksal saat değerlerinin birbirleri ile tutarlı olmasına karşın, dayanak düğümünün donanım saatine eşzamanlanmamış olmalarıdır. Ancak, eşzamanlanmış mantıksal saat değerini kullanarak, v düğümü dayanak düğümünün donanım saatini aşağıda gösterildiği gibi tahmin edebilmektedir:

$$H_v^{ref}(t) = L_v(t) + \Delta_v^{ref}(t). \quad (7.7)$$

Dolayısıyla, dayanak düğümünden alınan zaman öteleme değeri mantıksal saat değerine eklenerek dayanak düğümünün tahmini donanım saati hesaplanmaktadır. Algoritma 7.3, dayanak düğümünün tahmini donanım saat değeri H_v^{ref} 'i döndüren *dayanakSaati* arayüzünü göstermektedir. Sonuç olarak, *DMSEP* protokolü ağdaki tüm algılayıcı düğümlerini bir dayanak düğümünün donanım saatine eşzamanlamaktadır, çünkü düğümler bu düğümün donanım saati değerini tahmin edebilmektedirler.

7.3.1 Gerçek zamana eşzamanlanma

Eğer dayanak düğümü bir KKS donanımına sahipse ve bu sayede gerçek zamana erişebiliyorsa, tüm düğümler ortak saat hızında anlaştıktan sonra $t_0 < t_1$



Şekil 7.2: GTSP (sol) and DMSEP (sağ) protokollerinin karşılaştırılması. GTSP protokolünde, düğümler komşularıyla bir uzlaşma protokolü yürüterek ortak bir saat hızında ve saat değerinde anlaşmaktadırlar. DMSEP protokolünde ise, özel bir düğüm ek olarak kendi hız çarpanını ve zaman öteleme değerini ağa sel ile yayarak diğer düğümlerin kendi saat hızında ve saat değerinde anlaşmalarını sağlamaktadır.

olan t_0 ve t_1 anlarında iki gözlem yaparak, kendi donanım saat hızını 7.8 eşitliğinde gösterildiği gibi tahminleyebilir:

$$\hat{h}_{ref}(t) = \frac{H_{ref}(t_1) - H_{ref}(t_0)}{t_1 - t_0}. \quad (7.8)$$

Dayanak düğümü bu değeri kullanarak, ağ içerisindeki uzlaşma sonrası ortak mantıksal saat hızını, $l_{ref}(t)$ değerini bildiği için ve $h_{ref}(t)$ değerini 7.8 eşitliğini kullanarak tahmin edebildiği için, aşağıdaki gösterildiği gibi hesaplayabilir:

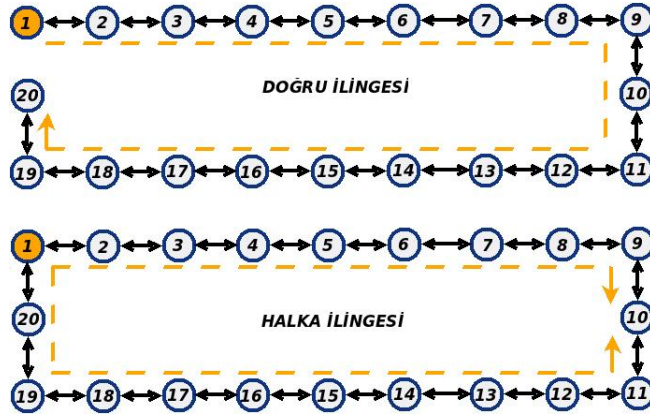
$$hiz = \hat{h}_{ref}(t) \cdot l_{ref}(t) \quad (7.9)$$

Bunlara ek olarak, dayanak düğümü, ağa $l_{ref}(t)$ yerine hiz değerini ve $\Delta_{ref}(t) = H_{ref}(t) - L_{ref}(t)$ yerine $\Delta_{ref}(t) = t - L_{ref}(t)$ değerini yaymalıdır.

Son bir değişiklik olarak, her v algılayıcı düğümü kendi mantıksal saatini 7.6 eşitliği yerine 7.10 eşitliğini kullanarak hesaplamalıdır:

$$L_v(t) = \int_0^t h_v(\tau) \frac{l_v(\tau)}{\hat{hiz}} d\tau + \theta_v(t). \quad (7.10)$$

Böylece, v düğümü kendi mantıksal saatini gerçek zamanın hızında (yani 1 hızında)



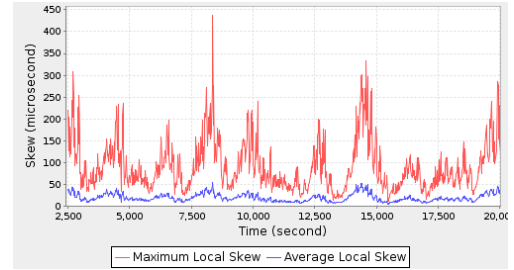
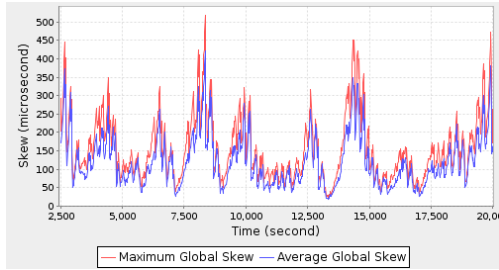
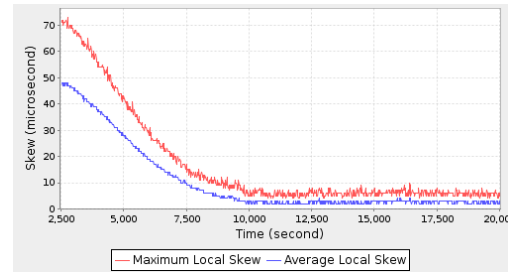
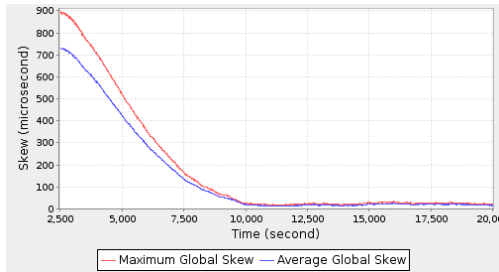
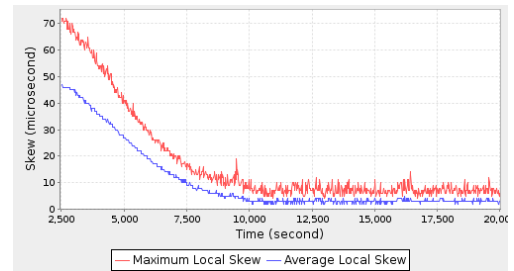
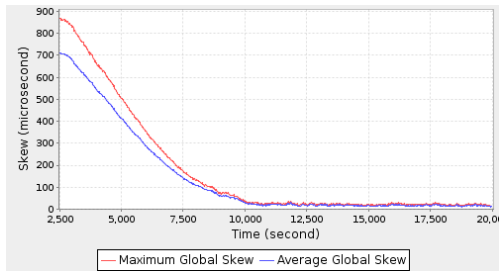
Şekil 7.3: Algilayıcı düğümlerin halka ve doğrusal ilingelerde yerleşimi.

ilerletmekte ve herhangi bir andaki gerçek zaman değerini, L_v değerine Δ_v^{ref} değerini ekleyerek hesaplayabilmektedir.

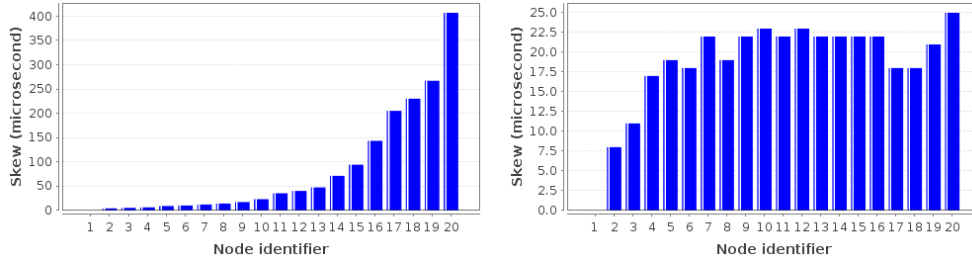
7.4 DeneySEL Sonuçlar

Bu bölümde DMSEP protokolü için, bölüm 5.7’te daha önce tanıtılan deney düzeneği kullanılarak gözlemlenen deneysel sonuçlar ve benzetim sonuçları sunulmaktadır. Başarım karşılaştırması için, DMSEP protokolü ve herkese açık bir gerçekleştirimi olmayan GTSP protokolü TinyOS 2.1.1 işletim sistemi kullanılarak geliştirilmiştir. FTSP protokolü ise bir dayanak düğüme ihtiyaç duyduğu için, yine başarım karşılaştırılmasında kullanılmıştır. Deneyler için, TinyOS 2.1.1 ile gelen en son FTSP sürümü sabit bir dayanak düğüme ile çalışacak şekilde değiştirilmiştir.

Deneyler için Şekil 7.3’de gösterilen halka ilingesi ve doğrusal ilinge oluşturulmuştur. Halka ilingesi, sel yöntemi ile tasarsız oluşan ağacın farklı dallarında olabilen birbirine komşu düğümlerin arasındaki eşzamanlama hatasını gözlemlemek için kullanılmıştır. Halka ilingesi ile kıyaslandığında daha büyük bir çapa sahip olan doğrusal ilinge ise, ağın çapı büyüdükçe eşzamanlama hatasının bundan nasıl etkilendiğini gözlemlemek için kullanılmıştır. Deneylerde eşzamanlama paketi gönderim periyodu 30 saniye ve herbir komşu için tutulan zaman etiketi tablosunun boyutu 8 olarak ayarlanmıştır. FTSP, GTSP and DMSEP için aynı uygulama derlenerek deney düzeneğinde yürütülmüştür. Deneylerde, algılayıcı düğümleri ilk 3 dakika içinde açılmıştır ve her bir deneysel süreç yaklaşık olarak 20000 saniye sürmüştür. Deneysel süreçlerin ilk 2500 saniyesi göz önüne alınmamıştır çünkü protokoller bu zamana kadar ağ genelinde eşzamanlanmayı henüz sağlayamamışlardır.

FTSP**GTSP****DMSEP**

Şekil 7.4: FTSP, GTSP ve DMSEP protokolleri için doğrusal ilingede ölçülmüş genel ve ortalama genel (sol sütun), yerel ve ortalama yerel (sağ sütun) saat farkı değerleri.

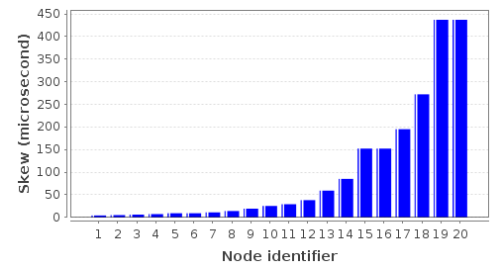
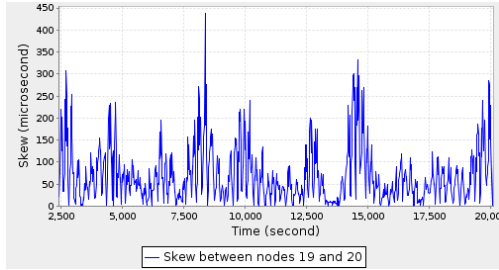
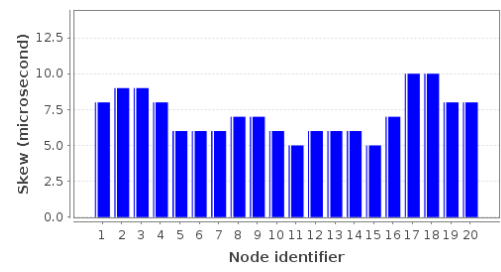
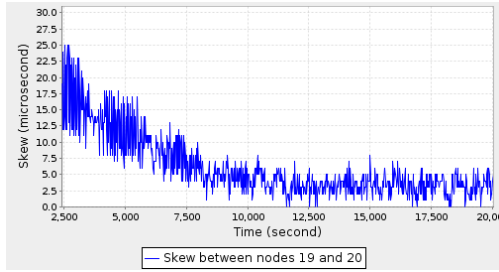
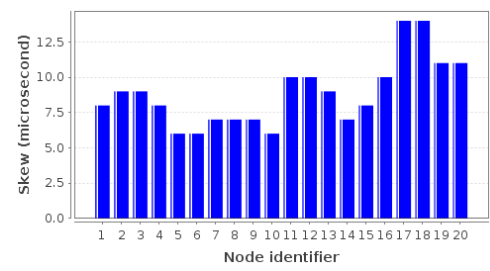
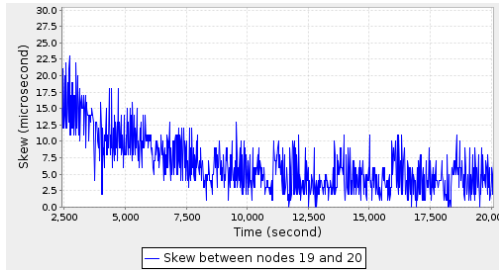


Şekil 7.5: FTSP (sol sütun) ve DMSEP (sağ sütun) protokolleri için doğrusal ilingede, dayanak düğümü (1 kimliğine sahip) ile diğer düğümler arasında gözlemlenen en büyük saat farkı değerleri.

Şekil 7.4 FTSP, GTSP ve DMSEP protokolleri için doğrusal ilingede ölçülmüş genel, ortalama genel, yerel ve ortalama yerel saat farkı değerlerini göstermektedir. FTSP ile en büyük genel ve en büyük ortalama genel saat farkı değeri olarak sırasıyla $518 \mu s$ ve $422 \mu s$ değerleri gözlemlenmiştir. FTSP için en büyük yerel ve ortalama yerel saat farkı değerleri ise sırasıyla $437 \mu s$ ve $55 \mu s$ olarak ölçülmüştür. GTSP ve DMSEP göz önüne alınırsa, uzlaşmanın sağlanması için yaklaşık 10000 saniye geçmesi gerekmiştir. Bu zamandan sonra, en büyük genel ve en büyük ortalama genel saat farkı değerleri olarak GTSP için sırasıyla $34 \mu s$ ve $26 \mu s$, DMSEP için ise sırasıyla $35 \mu s$ ve $29 \mu s$ değerleri gözlemlenmiştir. En büyük yerel ve en büyük ortalama yerel saat farkı değerleri ise GTSP için sırasıyla $10 \mu s$ ve $4 \mu sec$, DMSEP için ise $14 \mu s$ ve $5 \mu s$ olarak ölçülmüştür. Bu sonuçlar, GTSP ve DMSEP protokollerinin FTSP protokolünün doğrusal ilingedeki genel ve yerel saat farkı cinsinden başarımını oldukça düşürdüğünü ve bu iki protokolün başarımlarının birbirine oldukça benzer olduğunu göstermektedir.

Şekil 7.5 FTSP ve DMSEP protokolleri için dayanak düğümü ile diğer düğümler arasında gözlemlenen en büyük saat farkını göstermektedir. Görüleceği gibi, FTSP protokolünde dayanak düğümleri ile diğer düğümler arasındaki eşzamanlama hatası, bu düğümlerin aralarındaki uzaklık arttıkça üstel olarak artmaktadır. Bu durum daha önce (Lenzen et al., 2009b) çalışmasında gösterilmiş olup deneysel sonuçlar bu çalışmanın vurguladığı gerçeğe örtüşmektedir. Ancak DMSEP protokolünde tüm düğümler dayanak düğümünün saat hızı ve saat değeri üzerinde anlaştığı için, dayanak düğümüne yakın düğümler ile uzak düğümlerin eşzamanlama hataları birbirine oldukça yakındır. Dolayısıyla FTSP protokolünün sahip olduğu en ciddi sorun olan ölçeklendirilebilme sorunu DMSEP protokolü için geçerli değildir.

DMSEP protokolünün ana amacı, düğümleri kendi komşularıyla sıkı bir şekilde eşzamanlarken bu düğümleri bir dayanak düğümüne de eşzamanlamaktır. Şekil 7.6 doğrusal ilingenin en sonunda bulunan birbirine komşu 19 ve 20

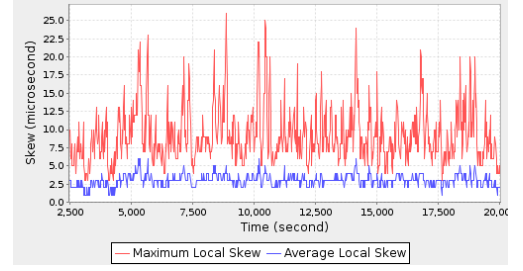
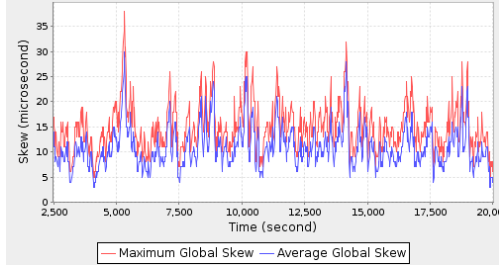
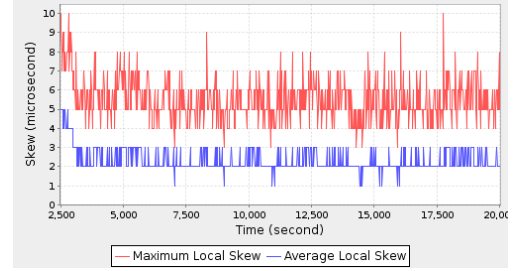
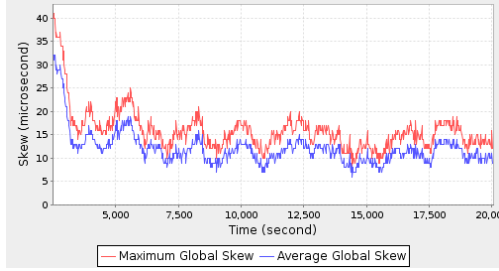
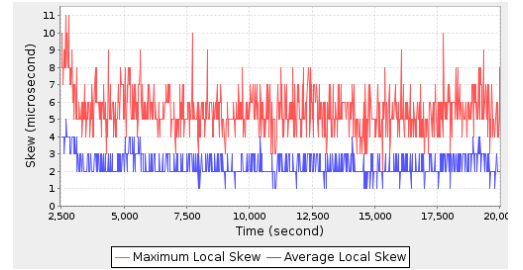
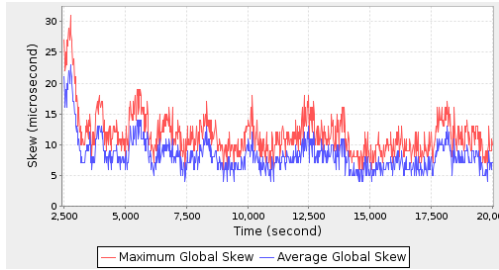
FTSP**GTSP****DMSEP**

Şekil 7.6: Sırasıyla FTSP, GTSP ve DMSEP protokoleri için doğrusal ilingede, 19 ve 20 kimliğine sahip düğümler arasında gözlemlenen eşzamanlama hatası (sol sütun) ve düğüm başına gözlemlenen en büyük yerel saat farkı (sağ sütun).

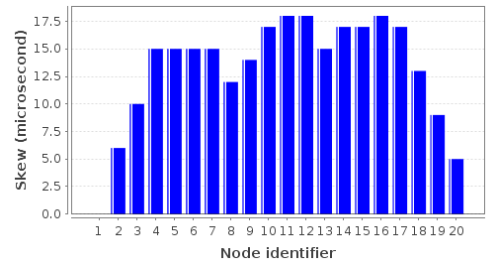
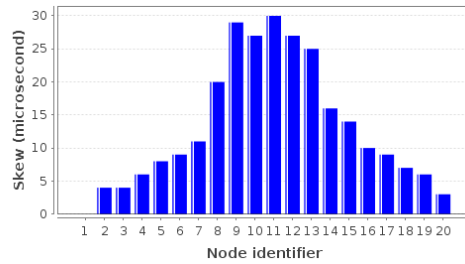
kimliklerine sahip düğümlerin arasındaki eşzamanlama hatasını göstermektedir. FTSP protokolünde, dayanak düğümünden uzaklaştıkça birbirine komşu olan düğümler arasındaki eşzamanlama hatası da artmaktadır. FTSP protokolü ile en büyük yerel saat farkı değeri olarak $400 \mu s$ 'den daha büyük bir değer gözlemlenmiştir. Ancak GTSP ve DMSEP protokolleri ile gözlemlenen en büyük yerel saat farkı değerleri, dayanak düğümüne olan uzaklık arttıkça aşırı derecede büyümemekte ve bu iki protokolde gözlemlenen değerler birbirine oldukça yakın olmaktadır. Bu durum 19 ve 20 düğümleri arasındaki saat farkı gözlemlenerek de görülebilir. FTSP protokolünde düğümler sadece dayanak düğümünün zaman bilgisini göz önüne alıp kendi komşularının zaman bilgisini kullanmadıkları için, 19 ve 20 düğümleri arasında gözlemlenen saat farkı değeri oldukça büyük olmaktadır. Bu iki düğüm arasında GTSP ve DMSEP protokolleri ile gözlemlenen saat farkı değeri FTSP protokolüne göre oldukça düşüktür. GTSP ve DMSEP protokollerinde düğümlerin kendi komşularıyla bir uzlaşma algoritması çalıştırmalarından dolayı bu iyileşme ortaya çıkmaktadır. GTSP protokolünden farklı olarak DMSEP protokolü düğümleri aynı zamanda bir dayanak düğümüne de eşzamanlamasına rağmen, birbirine komşu düğümler arasında gözlemlenen eşzamanlama hataları bu iki protokol için oldukça benzerdir.

Şekil 7.7 FTSP, GTSP ve DMSEP protokolleri için halka ilingede ölçülmüş genel, ortalama genel, yerel ve ortalama yerel saat farkı değerlerini göstermektedir. En büyük genel ve en büyük ortalama genel saat farkı değerleri FTSP için sırasıyla $38 \mu s$ ve $30 \mu s$ olarak gözlemlenmiştir. FTSP protokolünün ortaya koyduğu en büyük yerel ve en büyük ortalama yerel saat farkı değerleri ise $26 \mu s$ ve $6 \mu s$ 'dir. Gözlemlenen en büyük genel ve en büyük ortalama genel saat farkı değerleri GTSP için sırasıyla $20 \mu s$ ve $15 \mu s$, DMSEP için sırasıyla $19 \mu s$ ve $14 \mu s$ 'dir. GTSP protokolü en büyük yerel ve en büyük ortalama yerel saat farkı olarak sırasıyla $10 \mu s$ ve $3 \mu s$ değerlerini ortaya koymuşken, DMSEP protokolü için bu değerler sırasıyla $10 \mu s$ ve $4 \mu s$ olarak gözlemlenmiştir. GTSP ve DMSEP protokollerinde uygulanan uzlaşma algoritması 10 sekmelik küçük bir çapa sahip halka ilingesinde de üstünlüğünü göstermiştir.

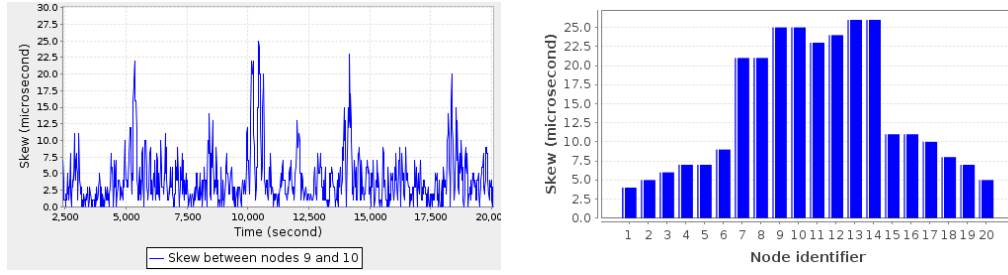
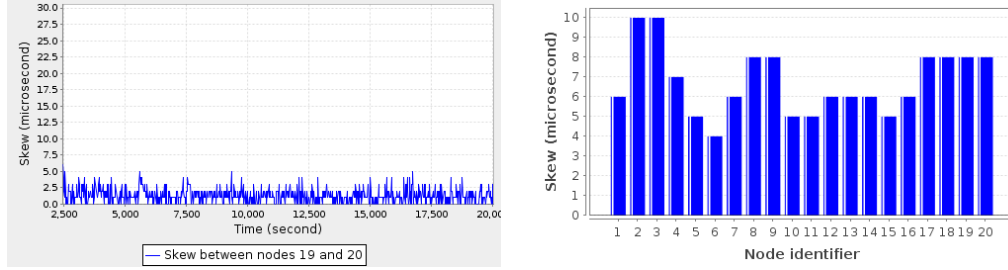
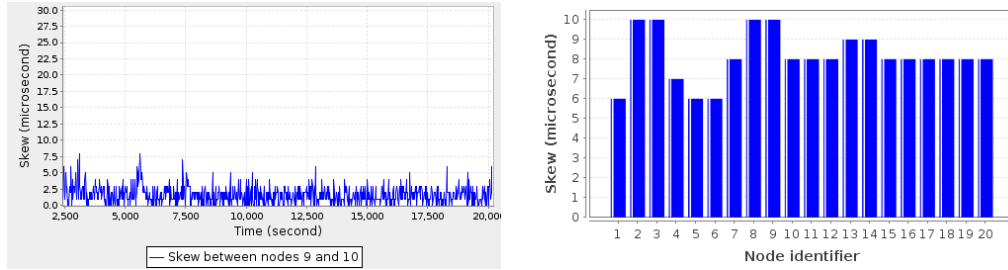
Şekil 7.8 FTSP ve DMSEP protokolleri için halka ilingede, dayanak düğümü ile diğer düğümler arasında gözlemlenen en büyük saat farkı değerlerini göstermektedir. Şekil 7.3'de gösterildiği gibi, sel yöntemi nedeniyle 1 düğümünün kök olduğu 2 alt ağaç oluşmakta ve 8 ve 12 düğümleri dayanak düğümüne en büyük uzaklığa sahip düğümler olmaktadır. FTSP protokolünde bu düğümler dayanak düğümü ile aralarında en büyük eşzamanlama hatasının gözlemlendiği düğümlerdir. Ancak, DMSEP protokolünde, dayanak düğümü ile diğer düğümler arasındaki saat farkı, doğru ilingesinde olduğu gibi halka ilingesinde de birbirlerine oldukça

FTSP**GTSP****DMSEP**

Şekil 7.7: FTSP, GTSP ve DMSEP protokolleri için halka ilingede ölçülmüş genel ve ortalama genel (sol sütun), yerel ve ortalama yerel (sağ sütun) saat farkı değerleri.



Şekil 7.8: FTSP (sol sütun) ve DMSEP (sağ sütun) protokolleri için halka ilingede, dayanak düğümü (1 kimliğine sahip) ile diğer düğümler arasında gözlemlenen en büyük saat farkı değerleri.

FTSP**GTSP****DMSEP**

Şekil 7.9: Sırasıyla FTSP, GTSP ve DMSEP protokoleri için halka ilingede, 9 ve 10 kimliğine sahip düğümler arasında gözlemlenen eşzamanlama hatası (sol sütun) ve düğüm başına gözlemlenen en büyük yerel saat farkı (sağ sütun).

yakındır.

Şekil 7.9 FTSP, GTSP ve DMSEP protokoleri için halka ilingesinde, 9 ve 10 kimliğine sahip düğümler arasında gözlemlenen eşzamanlama hatasını ve düğüm başına gözlemlenen en büyük yerel saat farkı değerlerini göstermektedir. Görülebileceği gibi FTSP protokolünde, dayanak düğümünden uzaklaştıkça gözlemlenen en büyük yerel saat farkı doğrusal ilingede olduğu gibi halka ilingesinde de artmaktadır. Ancak bu durum GTSP ve DMSEP protokollerinde için ortaya çıkmamaktadır. 9 ve 10 düğümleri göz önüne alındığında komşu olmalarına rağmen FTSP protokolünde bu düğümler arasında $25 \mu s$ yerel saat farkı gözlemlenmiştir. Ancak GTSP ve DMSEP protokollerinde bu iki düğüm arasında sırasıyla $5 \mu s$ ve $8 \mu s$ saat farkı gözlemlenmiştir. FTSP protokolünde gözlemlenen büyük eşzamanlama hatasının nedeni bu iki düğüme eşzamanlama bilgisinin sel yönteminin oluşturduğu tasarsız ağacın farklı dallarından ulaşması ve zaman bilgisindeki hataların bu iki farklı yol için farklı değerlere sahip olmasıdır.

	Doğrusal İlinge			Halka İlingesi		
	FTSP	GTSP	DMSEP	FTSP	GTSP	DMSEP
En Büyük Genel Saat Farkı	518 μs	34 μs	35 μs	38 μs	20 μs	19 μs
En Büyük Ortalama Genel Saat Farkı	422 μs	26 μs	29 μs	30 μs	15 μs	14 μs
En Büyük Yerel Saat Farkı	437 μs	10 μs	14 μs	26 μs	10 μs	10 μs
En Büyük Ortalama Yerel Saat Farkı	55 μs	4 μs	5 μs	6 μs	3 μs	4 μs

Çizelge 7.1: FTSP, GTSP ve DMSEP protokolleri için gözlemlenen deneysel sonuçların özeti.

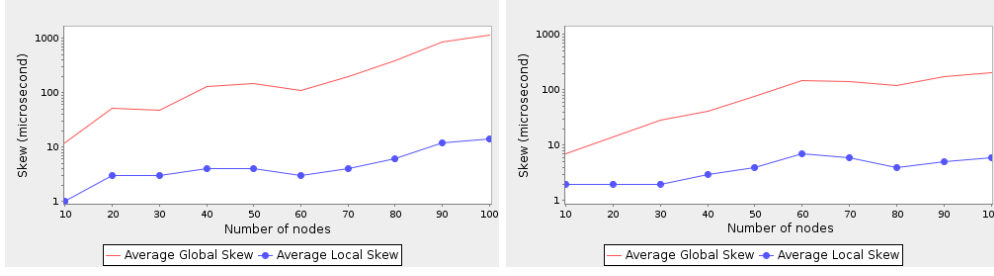
Gözlemler, GTSP ve DMSEP protokollerinin birbirine komşu düğümleri FTSP protokolüne göre daha iyi eşzamanladığını ve düğümlerin en büyük yerel saat farkı değerlerinin birbirlerine oldukça yakın olduğunu ortaya koymaktadır.

Çizelge 7.1 FTSP, GTSP ve DMSEP protokolleri için gözlemlenen doğrusal ve halka ilingelerindeki deneysel sonuçların bir özetini sunmaktadır. Deneysel sonuçlardan DMSEP protokolünün algılayıcı düğümleri bir dayanak düğüme eşzamanlarken GTSP protokolünün eşzamanlama başarımını koruduğunu göstermektedir. FTSP ve DMSEP protokolleri algılayıcı düğümleri bir dayanak düğüme eşzamanlamalarına rağmen, DMSEP protokolü FTSP protokolünü genel ve yerel saat farkı değerleri açısından oldukça geride bırakmaktadır.

7.4.1 DMSEP protokolü için benzetim sonuçları

Gerçek donanım ortamındaki deneyler dışında, DMSEP protokolünün başarımının ağıın çapının büyümesinden nasıl etkilendiği hakkında bir izlenim elde etmek için, Java programlama dilini kullanarak geliştirdiğimiz benzetim ortamında yapılan benzetim sonuçlarını bu bölümde sunmaktayız. Benzetimlerde düğümlerin saat sapmaları rastgele ± 50 ppm olacak şekilde yazılımla modellenmiştir. Değişik büyüklüklere sahip doğrusal ve halka ilingelerinde benzetimler yapılmıştır. Her bir ağ için 10 adet benzetim yapılmış ve bu 10 benzetimin ortalaması alınmıştır. Şekil 7.10'da gösterilen benzetim sonuçlarına dayanarak, DMSEP protokolünün eşzamanlama başarımının ağıın çapının büyümesinden oldukça yavaş bir şekilde etkilendiği söylenebilir.

Benzetimler DMSEP protokolünün, aynı uzlaşma algoritmalarını yürüttükleri



Şekil 7.10: DMSEP protokolü için farklı çaplara sahip doğrusal (sol) ve halka (sağ) ilingelerinde benzetim sonuçları.

için GTSP protokolü için de geçerli olan, önemli bir eksikliğini ortaya çıkarmıştır. Buradaki sorun ağın çapı büyüdükçe ağ genelindeki uzlaşmanın sağlanması için geçen zamanın oldukça artmasıdır. Şekil 7.7’de gösterilen 20 adet halka ilingesine sahip algılayıcı düğümü üzerinde (çapı 10 olan küçük bir ilinge) gerçekleştirilen deneyden görülebileceği üzere, algılayıcı düğümlerin arasında sıkı eşzamanlanmanın ve uzlaşmanın sağlanması için yaklaşık 4000 saniye (yaklaşık 1 saat) geçmesi gerekmektedir. 100 adet algılayıcı düğümü içeren halka ilingesindeki benzetimlerde ise uzlaşım için yaklaşık 90000 saniye (yaklaşık 25 saat) geçmesi gerekmiştir. Gerçek deneylerden ve benzetimlerden çıkarılan sonuç, GTSP ve DMSEP protokollerinin uyguladığı ortak uzlaşım algoritmasının özellikle çapı büyük olan ağları oldukça yavaş ama sıkı bir şekilde eşzamanladığıdır.

7.5 Sonuçlar

KAA için tasarlanmış birçok saat eşzamanlama protokolü herhangi iki düğüm arasında iyi eşzamanlama sağlayabilirken, birbirlerine komşu düğümler arasında büyük hataların oluşmasını engelleyememektedirler. Örneğin FTSP protokolünde, düğümler bir dayanak düğüme eşzamanlanırken komşularının zaman bilgisini göz önüne almadıkları için birbirine komşu düğümler kötü eşzamanlanmaktadır. GTSP protokolü, bu durumu düğümleri kendi komşularına eşzamanlayacak bir uzlaşım algoritması uygulayarak ortadan kaldırmaktadır. Ancak GTSP düğümleri bir dayanak düğüme eşzamanlayamadığı için, algılayıcı ağının dışsal bir zaman kaynağına eşzamanlanması ucu açık bir sorundur. Dışsal saat eşzamanlaması, birbirine komşu düğümlerin sıkı bir şekilde eşzamanlanmasına da ihtiyaç duyan hedef takibi gibi uygulamaların doğru işleyişi için hayati önem taşımaktadır.

Bu çalışmada, belirtilen açık sorun ele alınmış ve birbirine komşu düğümlerin arasında sıkı eşzamanlama sağlarken bu düğümleri aynı zamanda bir dayanak düğüme de eşzamanlayan Dışsal Meyilli Saat Eşzamanlama Protokolü (DMSEP) tanıtılmıştır. DMSEP protokolünde düğümler, bir dayanak düğümü

tarafından ağa sel yöntemi ile yayılan zaman bilgisini kullanarak ve GTSP protokolündeki komşu düğümlerden gelen verinin ortalamasını almaya dayalı uzlaşım algoritmasını yürüterek, dayanak düğümünün saat hızı ve saat değeri üzerinde eşzamanlanmaktadırlar. Deneysel sonuçlar DMSEP protokolünün GTSP protokolünün eşzamanlama başarımını koruduğunu ve FTSP protokolünü genel ve yerel saat farkı açısından oldukça geride bıraktığını göstermiştir. Benzetim sonuçları, DMSEP protokolünün eşzamanlama başarımının ağın çapının artmasından oldukça az bir şekilde etkilendiğini göstermiştir.

DMSEP protokolünün, aynı uzlaşım algoritmasını yürüttükleri için GTSP için de geçerli olan, temel eksikliği tüm algılayıcı düğümlerin aynı saat hızı ve saat değerinde uzlaşımı için geçen zamanın uzun olmasıdır. 20 algılayıcı düğümüne sahip doğrusal bir ilingede, uzlaşımın sağlanması için yaklaşık 10000 saniye (2.5 saatten fazla) zaman geçmesi gerekmiştir. DMSEP protokolünün uyguladığı uzlaşım algoritmasının uzlaşım zamanını hızlandırmayı bir gelecek çalışma olarak bırakıyoruz.

DMSEP protokolünün bir diğer eksikliği, algılayıcı ağı kurulmadan önce sabit bir dayanak düğümüne ihtiyaç olması ve algılayıcı ağının yaşamı boyunca bu düğümün değişmez oluşudur. DMSEP protokolü eğer dayanak düğümü arızalanırsa dışsal eşzamanlama sağlayamaz. Ancak uzlaşım algoritması çalışmaya devam edeceği için, DMSEP düğümlerin mantıksal saatlerini içsel olarak eşzamanlamaya devam eder. DMSEP protokolünü EEZ dışsal zaman kaynağına eşzamanlama için kullanılacaksa, bu zaman kaynağına erişebilen birden fazla dayanak düğümünün kullanılması protokolün dayanak düğümü arızalarına karşı sağlamlığını arttıracaktır.

“Ona sorarsanız: ‘Lafı bile edilemez, mikroskopik bi zaman. . .’

Bana sorarsanız: ‘On senesi ömrümün. . .’”

Nazım Hikmet RAN²³

8 OPTİMAL MEYİLLİ SAAT EŞZAMANLAMA ALGORİTMALARININ GERÇEKLEŞTİRİMLERİNİ KOLAYLAŞTIRMAK ÜZERİNE BİR ÇALIŞMA

Fan ve Lynch (Fan and Lynch, 2006), tanıtmış oldukları *meyilli saat eşzamanlaması* kavramı ile dağıtık sistemlerdeki saat eşzamanlamasına başka bir boyut getirmişlerdir. Herhangi bir ağda birbirine komşu iki düğüm arasında görülebilecek saat farkını alttan sınırlandırmışlardır. Gösterdikleri sınırı alttan ve üstten yakalayan, başka bir deyişle en kötü yerel saat farkı yönünden optimal olan saat eşzamanlama algoritması geliştirimi konusunu açık bırakmışlardır. Literatürde, meyilli eşzamanlama algoritmaları geliştirmeye yönelik birçok teorik çalışma bulunmaktadır. Bu çalışmalar içinden Kuhn et al. (Kuhn and Oshman, 2009) ve Lenzen et al. (Lenzen et al., 2010), birbirine komşu düğümler arasında en kötü durumdaki saat farkının Fan ve Lynch’in ispatladığı değerle ile alttan ve üstten sınırlandırıldığı, optimal meyilli saat eşzamanlama algoritmaları sunmuşlardır. Ancak bu iki optimal algoritmada düğümlerin saatlerinin hızlarını hesaplayan bölümler, uygulamada gerçekleştirilmeleri zor olan ifadeler içermektedirler. Lenzen et al. (Lenzen et al., 2010) geliştirdikleri algoritmanın, düğümlerin saat hızlarını değiştirdiği kısımlarının gerçekleştirimi için ne yazılımsal ne de donanımsal bir yöntem sunmuşlardır. Kuhn et al. (Kuhn and Oshman, 2009) kendi algoritmalarında düğümlerin saatleri için hızlı ve yavaş olmak üzere iki durum tanımlamışlardır ancak bu durumların yazılım ya da donanımla nasıl tespit edileceğine dair bir yöntem ortaya koymamışlardır.

Bu bölümün amacı, uygulamada gerçekleştirimi daha kolay olacak bir optimal meyilli saat eşzamanlaması algoritması ortaya koymaktır. Bu bağlamda (Kuhn and Oshman, 2009) ve (Lenzen et al., 2010) çalışmalarındaki optimal algoritmalar göz önüne alınmakta ve bu algoritmalarda yer alan uygulamada gerçekleştirimi zor olan ifadeler yerine, algoritmaların optimallik özelliklerini bozmayacak daha kolay gerçekleştirilebilir ifadeler sağlanmaktadır. Buna ek olarak, bu algoritmaların yürüttüğü görevler ayrı ve gerçekleştirilebilir modüllere ayrılarak, yeni bir meyilli saat eşzamanlama sistemi tanıtılmaktadır. Bu sistemin

²³Tezin büyük bir kısmını literatürdeki teorik meyilli saat eşzamanlaması algoritmalarını incelemeye ayırdık. Bu bölümün ortaya konulabilmesi için bize göre ömrümüzün on senesi geçmiştir, ancak okuyucu için mikroskopik bir zaman geçecektir.

uygulamada kolay bir şekilde gerçekleştirilebileceğini göstermek için yazılımsal bir gerçekleştirimi sunulmaktadır. Sistemin sadece yazılımla gerçekleştirimi uygulamada yoğun bir yük getireceği ve kullanışsız olacağı gerekçesiyle, sistemi yazılım işbirliği ile gerçekleştirecek özel bir donanım tasarımı taslağı da sunulmaktadır.

8.1 İlgili Çalışmalar

Literatürde, herhangi bir ağdaki herhangi iki düğüm arasındaki saat farkının olabilecek en küçük değerini (alt sınırını) ortaya koymaya teorik yönelik çalışmalar bulunmaktadır (Lundelius and Lynch, 1984; Halpern et al., 1985; Biaz and Welch, 2001; Patt-Shamir and Rajsbaum, 1994; Srikanth and Toueg, 1987). Bu çalışmalar arasında, Biaz ve Welch (Biaz and Welch, 2001) herhangi bir iletişim ağında genel saat farkının \mathcal{D} ağın çapı olmak üzere $\mathcal{D}/2$ değerinden küçük olamayacağını göstermişlerdir. Srikanth ve Toueg (Srikanth and Toueg, 1987) herhangi iki düğüm arasındaki saat farkının $\mathcal{O}(\mathcal{D})$ ile üstten sınırlandırıldığı optimal bir teorik algoritma sunmuşlardır. Ancak bu algoritmanın bazı yürütümleri, birbirine komşu düğümler arasında da $\mathcal{O}(\mathcal{D})$ saat farkının gözlemlenmesine neden olabilmektedir. Fan ve Lynch (Fan and Lynch, 2006) yerel saat farkını önemseyerek, düğümlerin arasında gözlemlenen saat farkının düğümlerin arasındaki uzaklığının bir fonksiyonu olduğu *meyilli saat eşzamanlaması* kavramını tanıtmışlardır. Herhangi bir ağda birbirine komşu düğümler arasında gözlemlenebilecek saat farkının $\Omega(\frac{\log \mathcal{D}}{\log \log \mathcal{D}})$ ile alttan sınırlandırıldığını ispatlamışlardır ve hiçbir saat eşzamanlama algoritmasının komşu düğümler arasında bu alt sınırdan daha düşük bir eşzamanlama hatası ortaya koyamayacağını göstermişlerdir. Meier ve Thiele (Meier and Thiele, 2005) iletişimde mesaj gecikmelerinin olmadığı bir modelde Fan ve Lynch'in gösterdiği alt sınıra benzer bir alt sınırı ispatlamışlardır.

Literatürde, yerel saat farkı için ispatlanan alt sınırı yakalamaya çalışan teorik meyilli saat eşzamanlama algoritması tasarımına yönelik birçok çalışma bulunmaktadır (Fan et al., 2005; Locher and Wattenhofer, 2006; Lenzen et al., 2008; Pussente and Barbosa, 2009; Kuhn and Oshman, 2009; Lenzen et al., 2010; Kuhn et al., 2009; Kuhn et al., 2010). Fan et al. (Fan et al., 2005) literatürdeki ilk gevşek meyilli saat eşzamanlama algoritmasını sunmuşlardır. Bu algoritmanın yürütümü süresince düğümler arasındaki saat farkı, birçok zaman düğümler arasındaki uzaklığın doğrusal bir fonksiyonudur. Locher ve Wattenhofer (Locher and Wattenhofer, 2006) birbirine komşu düğümler arasındaki saat farkını azaltmak için birkaç yöntem ortaya koymuşlardır. Bunlar saati en hızlı ilerleyen komşu düğüme eşzamanlanmak, tüm komşuların saatine eşzamanlanmak ve saati en yavaş

ilerleyen komşu düğüme eşzamanlanmak olarak sunulmuştur. En yavaş komşu düğüme eşzamanlanma yöntemini kullanan ve yürütümü sırasında komşu düğümler arasındaki saat farkının en kötü durumda $\mathcal{O}(\sqrt{\mathcal{D}})$ ile üstten sınırlandığı bir algoritma ortaya koymuşlardır. Bu algoritmanın eksikliği, düğümlerin ağın çapını bilmeye ihtiyaç duymalarıdır. Lenzen et al. (Lenzen et al., 2008) yerel saat farkının $\mathcal{O}(\log \mathcal{D})$ ile üstten sınırlandığı bir başka algoritma sunmuşlardır. Bu algoritmanın yürütümünde, tüm düğümler komşularının saatleri kendi saatlerinden belirli bir değer kadar geriye düşmedikleri sürece, kendi saatlerini o ana kadar aldıkları en büyük saat değerine ayarlayabilmektedirler. Düğümler komşularına emirler göndererek bu düğümlerin saatlerini arttırmalarını sağlayabilmektedirler. Ancak bu algoritma oldukça karmaşıktır ve uygulanabilirliği zordur. Pussente ve Barbosa (Pussente and Barbosa, 2009) yerel saat farkının $\mathcal{O}(1)$ ile üstten sınırlandığı bir algoritma sunmuşlardır. (Locher and Wattenhofer, 2006) çalışmasında sunulan algoritmaya benzer şekilde, bu algoritmanın yürütümünde düğümlerin ağın çapını bilmeleri gerekmektedir.

Lenzen et al. (Lenzen et al., 2010), Fan ve Lynch'in (Fan and Lynch, 2006) $\Omega(\frac{\log \mathcal{D}}{\log \log \mathcal{D}})$ olarak ortaya koyduğu yerel saat farkının alt sınırı geliştirerek, ε en büyük saat sapması olacak şekilde $\Omega(\log_{1/\varepsilon} \mathcal{D})$ değeri olarak ortaya koymuşlardır. Bu alt sınırın sıkı bir alt sınır olduğunu göstermek için, yerel saat farkının $\mathcal{O}(\log_{1/\varepsilon} \mathcal{D})$ ile üstten sınırlandığı bir optimal bir meyilli saat eşzamanlama algoritması sunmuşlardır. Kuhn ve Oshman (Kuhn and Oshman, 2009) komşu düğümlerin saatlerini tahmin etme görevini, bu tahminleri kullanarak saatlerin değerini düzeltme görevinden ayırmışlardır. Komşuların saatlerini tahmin etmek için *dayanak tümegönderim eşzamanlama*²⁴ yöntemini (Elson et al., 2002) kullanan optimal bir meyilli saat eşzamanlaması algoritması sunmuşlardır. (Kuhn and Oshman, 2009) ve (Lenzen et al., 2010) çalışmaları ile ortaya koyulmuş olan optimal meyilli saat eşzamanlama algoritmaları, uygulamada gerçekleştirilmeleri zor olan ifadeler içermektedirler. Bu ifadelerin yazılımla ya da donanımla nasıl gerçekleştirilebileceklerine dair bir yöntem önerilmemiştir.

Meyilli saat eşzamanlaması problemi düğümler ve kenarların yok olup tekrar ortaya çıkabildiği devingen ağlar için de çalışılmıştır (Kuhn et al., 2009; Kuhn et al., 2010). Devingen ağları göz önüne almak bu çalışmanın kapsamı dışındadır.

²⁴İngilizcesi: *Reference Broadcast Synchronization*.

8.2 Sistem Modeli İçin Değişiklikler ve Eklentiler

Bu bölümde, Bölüm 3'te sunulan sistem modelinde bazı değişiklikler ve eklemeler yapılacaktır. Düğümler arasındaki iletişimde ortaya çıkan mesaj gecikmelerinin belirlenimci olmayan bileşenleri, *mesaj gecikmesi belirsizliği* olarak adlandırılacaktır. Mesaj gecikmelerinin belirlenimci bileşenlerinin 0 olduğu ve mesaj gecikmesi belirsizliğinin $[0, \mathcal{T}]$ aralığında olduğu varsayılacaktır. Buna ek olarak düğümler arasındaki iletişimde mesaj kayıplarının olmadığı ve düğümlerin hiçbir zaman arızalanmadığı varsayılacaktır.

Özel bir düğümün ağa bir ilkleme mesajını sel ile yaydığı ve her $u \in V$ düğümü için bu ilkleme mesajını t_{init} anında alıncaya kadarki $t \leq t_{init}$ zamanlarında $H_u(t) = 0$ ve $L_u(t) = 0$ eşitliğinin sağlandığı varsayılacaktır. Buna ek olarak mantıksal saatin öteleme değeri olan $\theta_u(t)$ tüm t zamanları için 0 değerine sahip olacaktır. Bir saat eşzamanlaması algoritması için, mantıksal saatlerin hızını belirli aralıklarda tutmak tercih edilen bir durumdur. Matematiksel olarak u düğümünün mantıksal saatinin hız çarpanı için $1 < \beta$ sağlayan bir sabit olmak üzere aşağıdaki eşitsizliğin sağlandığı varsayılmaktadır:

$$1 \leq l_u(t) \leq \beta. \quad (8.1)$$

Düğümlerdeki işlemcilerin işlem yapmak için 0 süre harcadığı, yani işleme için bir zaman kaybı yaşanmadığı varsayılacaktır. Gerçek sistemlerde, herhangi bir anda okunan donanım saati bir tamsayı değerine sahip olduğu için, işlemcilerin mesajları işleme ve mesaj gönderme işlemlerini donanım saatlerinin bir tamsayı değere sahip olduğu anlarda yaptıkları varsayılacaktır. Bir mesaj alındığında, o mesajı işlemek için donanım saatinin bir tamsayı değere ulaşması beklenmektedir. Dolayısıyla,

$$\mathcal{T} \geq \frac{1}{1 - \varepsilon} \quad (8.2)$$

eşitliği sağlanmalıdır. Böylelikle, gerçek sistemlerde ki donanım saatinin kesikli ilerlemesi modellenmiş olmaktadır (Lenzen et al., 2009a).

8.3 Literatürdeki Optimal Meyilli Saat Eşzamanlama Algoritmalarının Gerçekleştirim Zorlukları

Algoritma 8.1 Lenzen et. al. (Lenzen et al., 2010) çalışmasında sunulan optimal meyilli saat eşzamanlama algoritmasının parçalarından Algoritma 3'ün 1. satırı.

$$R_v := \sup \left\{ R \in \mathbb{R} \mid \left\lfloor \frac{\Lambda_v^\uparrow - R}{\kappa} \right\rfloor \geq \left\lfloor \frac{\Lambda_v^\downarrow + R}{\kappa} \right\rfloor \right\}$$

Algoritma 8.1, Lenzen et. al. (Lenzen et al., 2010) çalışmasında sunulan optimal meyilli saat eşzamanlama algoritmasının, düğümlerin mantıksal saatlerinin ilerleme hızını belirleyen parçalarından Algoritma 3'ün 1. satırını göstermektedir. Bir v düğümü herhangi bir komşusundan yeni bir eşzamanlama mesajı aldığı zaman, burada gösterilmeyen satırlara ek olarak algoritmanın bu satırını da yürütmekte ve mantıksal saatinin donanım saatinin kaç tiki kadar hızlı çalışacağını temsil eden R_v değişkeninin değerini hesaplamaktadır. Bu algoritmada, Λ_v^\uparrow değişkeni mantıksal saat değeri en büyük olan komşuya olan saat farkını ve Λ_v^\downarrow değişkeni mantıksal saat değeri en küçük olan komşuya olan saat farkını temsil etmektedir. κ ise kabaca saat farkını ölçmek için algoritmanın kullandığı bir parametredir. Eğer bir $s \in \mathbb{N}$ tamsayı değeri için $\Lambda_v^\uparrow \leq s\kappa$ ve $\Lambda_v^\downarrow \geq s\kappa$ koşulu sağlanıyorsa, $R_v = 0$ sonucuna ulaşılır. Eğer böyle bir koşul sağlanmıyorsa, $R_v > 0$ olmaktadır. Buradaki temel sorun, R_v değişkeninin bir reel sayı olması nedeniyle, bu değişkeninin gerçek değerinin nasıl hesaplanacağıdır. $R_v > 0$ olduğu durumda, bu değişkenin değeri için bir üst sınır $\left\lfloor \frac{\Lambda_v^\uparrow - R}{\kappa} \right\rfloor \geq \left\lfloor \frac{\Lambda_v^\downarrow + R}{\kappa} \right\rfloor$ ifadesi kullanılarak

$$R_v \leq \frac{\kappa + \Lambda_v^\uparrow - \Lambda_v^\downarrow}{2} \quad (8.3)$$

olarak bulunabilir. Dolayısıyla R_v değişkeninin gerçek değerinin belirli bir ξ hassasiyeti ile hesaplanabilmesi için, $(0, \frac{\kappa + \Lambda_v^\uparrow - \Lambda_v^\downarrow}{2}]$ reel sayı aralığında bir *ikili arama* yapılabilir. İstenen ξ hassasiyeti arttırıldıkça, ikili aramanın üst sınırı olan $\mathcal{O}(\log \frac{\kappa + \Lambda_v^\uparrow - \Lambda_v^\downarrow}{2\xi})$ değeri de oldukça artmaktadır.

Algoritma 8.2 Kuhn et. al. (Kuhn and Oshman, 2009) çalışmasında sunulan optimal meyilli saat eşzamanlama algoritması.

(FC1) $\exists v \in N_u : L_u^v(t) - L_u(t) \geq (s - 1 - \lambda)\kappa$

(FC2) $\forall v \in N_u : L_u^v(t) - L_u(t) \geq (s - 1 + \lambda)\kappa$

(SC1) $\exists v \in N_u : L_u(t) - L_u^v(t) \geq (s - \frac{1}{2} - \lambda)\kappa$

(SC2) $\forall v \in N_u : L_u(t) - L_u^v(t) \leq (s - \frac{1}{2} + \lambda)\kappa$

Algoritma 8.2, (Kuhn and Oshman, 2009) çalışmasında sunulan optimal meyilli saat algoritmasının, düğümlerin mantıksal saatlerinin ilerleme hızını

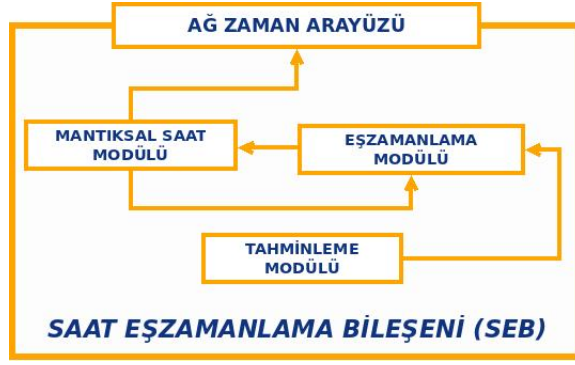
belirleyen koşullarını göstermektedir. Burada $L_u^v(t)$ değeri t anında u düğümünün v düğümünün mantıksal saati için tahminini ifade etmektedir. $\kappa > \frac{1}{\lambda}$ olmak üzere algoritmanın parametresi ve $\lambda < \frac{1}{4}$ ise bir sabittir. Eğer (FC1) ve (FC2) koşullarını (*hızlı durum*) aynı anda sağlayan bir $s \in \mathbb{N}$ tamsayısı bulunabilirse, u düğümünün mantıksal saati donanım saatinden daha hızlı ilerletilmektedir. Aksi taktirde, (SC1) ve (SC1) koşulları sağlanmakta ve u düğümünün mantıksal saati donanım saati ile aynı hızda ilerletilmektedir. Buradaki temel sorun, donanım saatinin her bir tıklamasında, hızlı ya da yavaş koşulu sağlayacak bir $s \in \mathbb{N}$ tamsayısının nasıl bulunacağıdır. (FC1) ve (FC2) koşullarını sağlayan $s \in \mathbb{N}$ için bir üst sınır aşağıdaki gibi elde edilebilir:

$$s \leq \left\lfloor \frac{\max_{v \in N_u} \{L_u^v(t) - L_u(t)\} + \max_{v \in N_u} \{L_u(t) - L_u^v(t)\}}{2\kappa} \right\rfloor + 1. \quad (8.4)$$

Burada $\left[0, \left\lfloor \frac{\max_{v \in N_u} \{L_u^v(t) - L_u(t)\} + \max_{v \in N_u} \{L_u(t) - L_u^v(t)\}}{2\kappa} \right\rfloor + 1\right]$ aralığında yer alan tamsayıların tek tek hızlı koşulu sağlayıp sağlamadıkları kontrol edilebilir ve hızlı koşulu sağlayan bir tamsayı bulunduğu anda mantıksal saat donanım saatinden daha hızlı ilerletilmektedir. $\mathcal{O}\left(\log\left(\left\lfloor \frac{\max_{v \in N_u} \{L_u^v(t) - L_u(t)\} + \max_{v \in N_u} \{L_u(t) - L_u^v(t)\}}{2\kappa} \right\rfloor + 1\right)\right)$ karmaşıklığına sahip bu arama, donanım saatinin her tıklamasında yapılmalıdır.

8.4 Kolay Gerçekleştirilebilir Meyilli Saat Eşzamanlaması Sistemi

Bu bölümde, genel mimarisi Şekil 8.1’de verilmiş yeni bir meyilli saat eşzamanlama sistemi tanıtılmaktadır. Bu sistem, (Kuhn and Oshman, 2009) ve (Lenzen et al., 2010) çalışmalarında sunulan optimal $\mathcal{O}(\log(\mathcal{D}))$ yerel saat farkını yakalayan algoritmalarla aynı işlevselliğe sahip bir *Saat Eşzamanlama Bileşeni (SEB)* içermektedir. Önceki bölümde, optimal meyilli algoritmalarının düğümlerin mantıksal saatlerinin ne zaman hızlı ne zaman normal hızda çalışacaklarına karar verebilmek için bir arama işlemine gereksinim duyduğu ortaya koyulmuştur. Özellikle (Kuhn and Oshman, 2009) çalışmasındaki algoritma için, bu arama her donanım saati tığında yapılmalıdır. Ancak SEB, düğümlerin mantıksal saatlerinin hız çarpanlarına herhangi bir arama yapmadan karar veren, daha kolay gerçekleştirilebilir bir algoritma içermektedir. Ek olarak, SEB optimal algoritmaların içerdiği görevleri ayrı ayrı gerçekleştirilebilir modüllere ayırmaktadır. Bu değişimler, diğer bölümlerde gösterileceği gibi,



Şekil 8.1: Meyilli saat eşzamanlama sisteminin modüler yapısı. Şekildeki oklar sistem içerisindeki modüller arasındaki etkileşimi göstermektedir.

SEB'in birbirinden ayrılmış yazılım ve donanım modülleri olarak gerçekleştirimini kolaylaştırmaktadır.

SEB üç farklı modülden oluşmaktadır. *Mantıksal Saat Modülü* düğümün mantıksal saatini temsil etmektedir. *Tahminleme Modülü* bir düğümün topladığı saat bilgilerini kullanarak kendi komşularının mantıksal saatlerini tahmin etmesini sağlamaktadır. Bu modül Kuhn et al. (Kuhn and Oshman, 2009) çalışmasında sunulan tahminleme modülüne benzerdir. *Mantıksal Saat Modülü* ve *Tahminleme Modülü*, diğer düğümlerle iletişimin ve bilgi değiş tokuşunun yapıldığı tek yerdir. *Eşzamanlama Modülü*, *Mantıksal Saat Modülü*'nün temsil ettiği mantıksal saatin ilerleme hızını *Tahminleme Modülü* ve *Mantıksal Saat Modülü*'nün sunduğu bilgiyi kullanarak ayarlamaktadır. Bu modüldeki algoritma, (Kuhn and Oshman, 2009) ve (Lenzen et al., 2010) çalışmalarındaki algoritmaların aksine, mantıksal saatin ilerleme hızını arama işlemine gerek duymadan daha kolay gerçekleştirilebilir bir şekilde değiştirmektedir. Dağıtık uygulamalar ise *Ağ Zaman Arayüzü*'nü sorgulayarak ağ genelindeki eşzamanlanmış zaman bilgisine erişebilmektedirler.

8.4.1 Mantıksal saat modülü

Mantıksal Saat Modülü düğümlerin mantıksal saatlerini barındırmakta ve mantıksal saatlerin ilerleme hızlarının değiştirilebilmesi için bir arayüz sunmaktadır. Bu modül kapsamında, her düğüm Algoritma 8.3'de gösterilen \mathcal{A}^{saat} algoritmasının adımlarını yürütmektedir.

\mathcal{A}^{saat} algoritmasını yürüten her u düğümü, dağıtık sistem genelindeki eşzamanlanmış saat değerini gösteren bir mantıksal saat değişkeni L_u 'ya sahiptir. Mantıksal saatin hız çarpanı l_u da bu modül içerisinde yer almaktadır. Mantıksal

Algoritma 8.3 \mathcal{A}^{saat} algoritması: u düğümünün mantıksal saatini temsil eder.

```

1 İkleme
2  $L_u \leftarrow 0$ ;
3  $l_u \leftarrow 1$ ;
4  $seq_u \leftarrow 0$ ;
5
6  $\square H_u$  her  $\partial\mathcal{H}$  kadar ilerlediğinde
7    $seq_u \leftarrow seq_u + 1$ 
8   Bütün  $v \in \mathcal{N}_u$  düğümlerine gönder  $\langle SYNC, L_u, seq_u \rangle$ 

```

saat değerini değiştirmenin tek yolu, mantıksal saatin hız çarpanını değiştirmektir. u düğümünün gönderdiği eşzamanlama mesajları için bir sıra numarası da bu modül içerisinde yer almaktadır. Sıra numarası, *Tahminleme Modülü*'nün komşuların mantıksal saat tahminlerini, yine komşulardan gelen mantıksal saat değerlerini kullanarak doğru bir şekilde güncelleyebilmesi için gereklidir.

Düğümler açıldığı anda (ikleme mesajını aldıkları t_{init} anında), mantıksal saat değeri 0 ile, hız çarpanı 1 ile ve sıra numarası değeri 0 ile ilklenmektedir (1-4 Satırları). Dolayısıyla, mantıksal saat *Eşzamanlama Modülü* tarafından hızı değiştirilmediği sürece donanım saati ile aynı hızda ilerlemektedir. Her düğüm periyodik olarak kendi mantıksal saat değerini taşıyan eşzamanlama mesajlarını komşularının *Tahminleme Modülü* için komşularına göndermektedir. u düğümünün donanım saati her $\partial\mathcal{H}$ değeri kadar ilerlediğinde (6. Satır), sıra numarası seq_u bir artırılarak gönderilecek mesajın en güncel mantıksal saat değerini taşıdığı belirtilmiş olmaktadır (7. Satır). Sonraki adımda $\langle SYNC, L_u, seq_u \rangle$ biçiminde bir mesaj tüm komşulara gönderilir (8. Satır). \mathcal{A}^{saat} algoritmasının kullandığı $\partial\mathcal{H}$ parametresi, eşzamanlama mesajlarının sıklığını azaltmak veya arttırmak için ayarlanabilmektedir.

8.4.2 Tahminleme modülü

Komşu düğümlerden gelen zaman bilgilerini toplayarak bu düğümlerin mantıksal saat değerlerini tahmin etmek *Tahminleme Modülü*'nün görevidir. Bu görevi yerine getirebilmek için her düğüm Algoritma 8.4'de gösterilen \mathcal{A}^{tahmin} algoritmasının adımlarını yürütmektedir.

Her u düğümü, her $v \in \mathcal{N}_u$ komşusu için bu komşusunun mantıksal saat değerinin bir tahminini temsil eden L_u^v değişkenine sahiptir. Bu komşu için tutulan mantıksal saat tahminini doğru bir şekilde güncelleyebilmek için, bu komşudan gelen mesajlardaki en büyük sıra numarası değeri olan seq_u^v de saklanmaktadır.

Algoritma 8.4 \mathcal{A}^{tahmin} algoritması: Komşu düğümlerin mantıksal saat değerlerini tahmin eder.

```

1 İkleme
2  $\forall v \in \mathcal{N}_u : L_u^v \leftarrow 0, seq_u^v \leftarrow 0$ 
3  $\Lambda_u^\uparrow \leftarrow 0; \Lambda_u^\downarrow \leftarrow 0$ 
4
5  $\square seq_u^v < seq_v$  olan bir  $\langle SYNC, L_v, seq_v \rangle$  mesajı alındığında
6    $seq_u^v \leftarrow seq_v$ 
7    $L_u^v \leftarrow L_v$ 
8    $\Lambda_u^\uparrow \leftarrow \max_{v \in \mathcal{N}_u} \{L_u^v(t) - L_u(t)\}$ 
9    $\Lambda_u^\downarrow \leftarrow \max_{v \in \mathcal{N}_u} \{L_u(t) - L_u^v(t)\}$ 

```

\mathcal{A}^{tahmin} algoritmasının yürütümü süresince, her $v \in \mathcal{N}_u$ komşusu için tutulan mantıksal saat tahmini L_u^v , u düğümünün donanım saatinin hızında ilerletilmektedir. Düğümler açıldığı zaman, tüm mantıksal saat tahminleri ve sıra numaraları 0 ile ilklenebilir (1-2 Satırları). (Lenzen et al., 2010) çalışmasındaki algorithmada olduğu gibi, her u düğümü en büyük mantıksal saat değerine sahip komşu düğüme olan saat farkı ve en küçük mantıksal saat değerine sahip olan komşu düğüme olan saat farkını temsil eden Λ_u^\uparrow ve Λ_u^\downarrow değişkenlerine de sahiptir. Bu değişkenler 0 ile ilklenebilirler (3. Satır). Λ_u^\uparrow ve Λ_u^\downarrow değerleri mantıksal saat donanım saati ile aynı hızda ilerlediği ve komşu düğümlerden yeni bir eşzamanlama mesajı gelmediği sürece değişmeden kalmaktadırlar. Ancak, mantıksal saatin hız çarpanı 1 değerinden büyük olduğu durumlarda Λ_u^\uparrow azalmakta ve Λ_u^\downarrow artmaktadır.

Herhangi bir $v \in \mathcal{N}_u$ komşusundan $\langle SYNC, L_v, seq_v \rangle$ biçiminde bir mesaj alındığı zaman, öncelikle sıra numarası kontrol edilmektedir. Eğer bu mesaj güncel bir mesaj ise (sıra numarası o an saklanan sıra numarasından büyük bir değere sahipse), o düğümün tahmini mantıksal saat değeri ve sıra numarası güncellenmektedir (5-7 Satırları). Tahmini mantıksal saat değeri güncellendikten sonra, Λ_u^\uparrow ve Λ_u^\downarrow değişkenleri yeniden hesaplanmaktadır (8-9 Satırları).

Aşağıdaki yardımcı teorem ile, düğümlerin komşuları için tahminledikleri mantıksal saat değerleri alttan ve üstten sınırlandırılarak, bu tahminlerin hatalarını ortaya koyulmaktadır.

Lemma 8.1. Her $u \in V$, $v \in \mathcal{N}_u$ düğümleri ve tüm t zamanları için,

$$-\epsilon < L_v(t) - L_u^v(t) < \epsilon \quad (8.5)$$

eşitsizliği, $\partial \bar{\mathcal{H}} = (\beta - 1 + 2\epsilon)\partial \mathcal{H}$ ve

$$\epsilon = \partial \bar{\mathcal{H}} + \frac{\beta + 2\epsilon}{1 - \epsilon} \mathcal{T} \quad (8.6)$$

için sağlanmaktadır.

İspat. Bu yardımcı teoremin ispatı bölüm 8.8’de verilmiştir. \square

8.4.3 Eşzamanlama modülü

Eşzamanlama Modülü mantıksal saatini hız çarpanını ayarlama görevini yerine getirmektedir. Bu modül içerisinde çalışan ve Algoritma 8.5’de gösterilen \mathcal{A}^{meyil} algoritması, (Kuhn and Oshman, 2009) ve (Lenzen et al., 2010) çalışmalarında sunulan algoritmaların üzerinde yapılan değişikliklere dayanmaktadır. Değişiklikler, gerçekleştirim için uygulanması zor olan ifadelerin elenmesini içermektedir. Bu değişikliklerin sonucunda, \mathcal{A}^{meyil} algoritması ilerleyen bölümlerde gösterileceği gibi yazılım ve donanım işbirliği ile daha kolay gerçekleştirilebilir olmaktadır. Düğümlerin \mathcal{A}^{meyil} algoritmasını t_{init} anında yürütmeye başladıkları unutulmamalıdır.

$$\mathcal{F}_u(t) = \left\lfloor \Lambda_v^\uparrow / \kappa - \frac{1}{4} \right\rfloor - \left\lfloor \Lambda_v^\downarrow / \kappa + \frac{1}{4} \right\rfloor. \quad (8.7)$$

Algoritma 8.5 u düğümü için \mathcal{A}^{meyil} algoritması.

- 1 $\square \mathcal{F}_u \geq 0$ sağlandığında
 - 2 $l_u \leftarrow \beta$
 - 3
 - 4 $\square \mathcal{F}_u < 0$ sağlandığında
 - 5 $l_u \leftarrow 1$
-

\mathcal{A}^{meyil} algoritmasının temel noktası, 8.7 eşitliğinde gösterilen \mathcal{F}_u mod fonksiyonudur. Mod fonksiyonu *Tahminleme Modülü*’nde tutulan Λ_v^\uparrow ve Λ_v^\downarrow değişkenlerini ve κ parametresini kullanmaktadır. κ parametresi saat farkını ölçmek için kullanılan bir birim olarak düşünülebilir. κ parametresi için aşağıdaki eşitsizliğin sağlanması gerekmektedir:

$$\kappa > 4\epsilon. \quad (8.8)$$

u düğümünün, mantıksal saati kendisinin mantıksal saatinden en ilerde olan komşusuna olan saat farkı, en geride olan komşusuna olan saat farkından çok büyükse, $\mathcal{F}_u \geq 0$ sağlanmaktadır. Dolayısıyla Algoritma 8.5'in 1-2 satırlarına göre u düğümünün mantıksal saatinin hız çarpanı β olmakta ve mantıksal saati donanım saatinden daha hızlı bir şekilde ilerletilmektedir. Aksi durumda ise $\mathcal{F}_u < 0$ sağlanmakta, u düğümünün mantıksal saatinin hız çarpanı 1 olmakta ve mantıksal saati donanım saati ile aynı hızda ilerletilmektedir (4-5 Satır).

Aşağıdaki teorem ile \mathcal{A}^{meyil} algoritmasının tüm işleyişlerinin tüm zamanlarında, düğümler arasında görülen yerel saat farkının $\mathcal{O}(\log(\mathcal{D}))$ olan optimal değerle üstten sınırlandığı ispatlanmaktadır.

Teorem 8.2. *İletişim ağı tüm t zamanları ve tüm $P = u, \dots, v$ yolları için aşağıda tanımlandığı şekilde **uygun** durumdadır:*

$$L_u(t) - L_v(t) \leq \kappa d(u, v) \left(\log_{\sigma} \frac{2\mathcal{D}}{d(u, v)} + \frac{1}{2} \right) \quad (8.9)$$

öyle ki

$$2 \leq \sigma \leq \sqrt{\frac{(\beta - 1)(1 - \varepsilon)}{4\varepsilon}}. \quad (8.10)$$

İspat. \mathcal{A}^{meyil} algoritmasının yürütümündeki $\mathcal{F}_u \geq 0$ ve $\mathcal{F}_u < 0$ koşullarının (Kuhn and Oshman, 2009) çalışmasında gösterilen hızlı ve yavaş koşullarına denk olduğu gösterilebilir. Detaylı ispat 8.9 bölümünde yer almaktadır. \square

1024 çapına sahip doğrusal bir ilingede ($\mathcal{D} = 1024$), her düğümün 32 KHz salıncı frekansına sahip bir donanım saatine sahip olduğu bir iletişim ağı göz önüne alınsın. Dolayısıyla, donanım saatinin her bir tıki 2^5 mikrosaniyeye denk düşmektedir. Donanım saatleri en fazla 32 ppm sapmaya ($\varepsilon = 2^{-15}$) sahip olsun. Düğümlerin birbirleri ile yaklaşık her dört saniyede bir iletişim kurdukları ($\partial\mathcal{H} = 2^{17}$) varsayalım. β ve κ parametrelerine, 8.6 eşitliği, 8.8 ve 8.10 eşitsizlikleri göz önüne alınarak, Çizelge 8.1'deki değerler verilmiş olsun. \mathcal{A}^{meyil} algoritması için yukarıda sunulan teorem ile ortaya konulan yerel saat farkı için üst sınır, uygulamada geçerli olabilecek parametre değerleri yerine konulursa, 3220 bulunur. Her bir tiki 2^5 mikrosaniye denk geldiği için birbirine komşu düğümler arasındaki eşzamanlama hatası, en fazla yaklaşık 103 milisaniye olabilmektedir.

Çizelge 8.1: SEB'in örnek işletimi için parametreler.

Parametre	Değer
κ	280
β	1.0005
ε	2^{-15}
$\partial\mathcal{H}$	2^{17}
σ	2
\mathcal{T}	0
\mathcal{D}	1024

8.4.4 Literatürdeki diğer algoritmalarla karşılaştırma

SEB'in literatürdeki diğer optimal algoritmalarla **temel farkı**, mantıksal saatlerin hız çarpanına karar verme sürecinin herhangi bir arama işlemi yapılmadan çok temel aritmetik işlemlerle yapılmasıdır. Daha önce belirtilmiş olduğu gibi, (Kuhn and Oshman, 2009) ve (Lenzen et al., 2010) çalışmalarındaki optimal algoritmalar bu süreç için karmaşıklığı büyük olabilecek arama işlemlerine gereksinim duymaktadırlar. Sonraki bölümlerde gösterileceği gibi, hız çarpanına karar verme işlemlerinin daha basit olması SEB bileşeninin yazılımla daha kolay bir şekilde gerçekleştirilebilmesini sağlamaktadır.

8.5 Yazılımsal Gerçekleştirim

SEB'in yazılım ile kolay bir şekilde gerçekleştirilebilir olduğunun ispatlanabilmesi için, içerdiği modüller Java programlama dili ile yazılımsal olarak gerçekleştirilmiş ve çalışması gözlemlenmiştir. Bir önceki bölümde sunulan optimal algoritmanın düğümlerin mantıksal saatlerinin ne zaman hızlı ne zaman normal hızda ilerleyeceğini belirlemek için kullandığı mod fonksiyonunun yazılımsal gerçekleştiriminin ana kısmı, Şekil 8.2'de gösterilmiştir. Donanım saatinin her bir tıki **tık()** metodu ile gerçekleştirilmiştir. Bu fonksiyon içerisinde, mod fonksiyonunun hesaplanabilmesi için öncelikle *Tahminleme Modülü* tarafından sağlanan Λ_v^\uparrow ve Λ_v^\downarrow değerleri alınmaktadır (4-6 Satırları). Bu değerler kullanılarak, görüldüğü gibi mod fonksiyonunun değeri basit matematiksel işlemlerle hesaplanmaktadır (8-10 Satırları). Mantıksal saatin yeni hızına ise, yine çok basit şekilde *if-then-else* yapısı ile karar verilebilmektedir (11-14 Satırları).

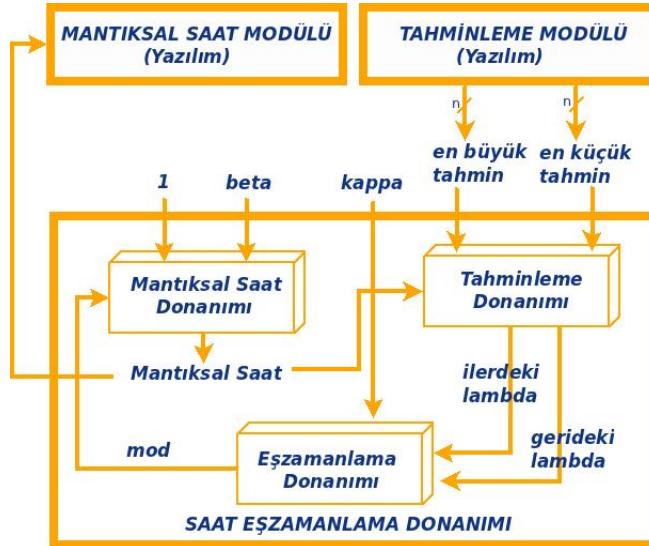
```

1: public void tick(){
2:   ...
3:
4:   /* Get lambda ahead and lambda behind. */
5:   double max=estimateModule.getLambdaAhead();
6:   double min=estimateModule.getLambdaBehind();
7:
8:   /* Compute the mode function. */
9:   double a=(max-logicalClock.getValue())/kappa-0.25;
10:  double b=(logicalClock.getValue()-min)/kappa+0.25;
11:  if(Math.floor(a)>=Math.floor(b))
12:    logicalClock.setMult(beta);
13:  else
14:    logicalClock.setMult(1);
15:  ...
16:
17:}

```

Şekil 8.2: Eşzamanlama Modülü'ndeki mantıksal saatin hızının Java ile gerçekleştirimi.

8.6 Donanımsal Gerçekleştirim İçin Taslak Tasarım



Şekil 8.3: Yazılım/donanım işbirliği ile gerçekleştirilen Saat Eşzamanlama Bileşeni (SEB).

SEB bileşeninin bir önceki bölümde gösterilen yazılımsal gerçekleştirimi, her donanım saat tıkında hesaplama ve düzenlemelere gereksinim duymaktadır. Mantıksal saatin değeri, Λ_v^\uparrow ve Λ_v^\downarrow değişkenlerinin değerleri her donanım saati tıkladığında ilerletilmeli, mod fonksiyonu da her donanım saati tıkında hesaplanmalı ve mantıksal saatin hız çarpanı ayarlanmalıdır. Bu durum göz önüne alındığında, SEB bileşenini donanımın da yardımıyla gerçekleştirmek, yazılımın getirdiği işlemci yükünü hafifletecektir. Bu bölümde, SEB bileşenini yazılımla işbirliği içerisinde gerçekleştirecek bir adanmış donanımın taslak tasarımı

sunulmaktadır. Şekil 8.3, donanım tasarımının ana bloklarını göstermektedir.

Saat eşzamanlama donanımında yer alan *Mantıksal Saat Devresi*'nin girdileri, 1 ve β olan mantıksal saatin hız çarpanları ve *mod* seçicisidir. Bu devre, mantıksal saatin değerini tutan bir yazmaça sahiptir ve her donanım saat tığında yazmaçın değerini seçilen hız çarpanı kadar arttırmaktadır. Mantıksal saat için β ya da 1 olabilen hız çarpanını, *Eşzamanlama Donanımı*'nın çıktısı olan mod seçicisi tarafından seçilmektedir. Yazılım ile gerçekleştirilmiş *Mantıksal Saat Modülü*, bu bileşenin sakladığı mantıksal saat değerini okuyabilir ve donanım saati her $\partial\mathcal{H}$ ilerlediğinde komşu düğümlere mantıksal saatin değerini göndermektedir.

Herhangi bir komşudan yeni bir mantıksal saat tahmini alındığında, yazılım ile gerçekleştirilmiş olan *Tahminleme Modülü*, donanım ile gerçekleştirilmiş olan *Tahminleme Devresi*'ne girdi olan, *en büyük tahmin* $\max_{v \in \mathcal{N}_u} \{L_u^v(t)\}$ ve *en küçük tahmin* $\min_{v \in \mathcal{N}_u} \{L_u^v(t)\}$ değerlerini güncellemektedir. Bu tahminler, *Tahminleme Devresi* tarafından donanım saati ile aynı hızda ilerletilmektedirler. *Gerideki lambda* ($\Lambda_v^\uparrow = \max_{v \in \mathcal{N}_u} \{L_u^v(t) - L_u(t)\}$) ve *ilerdeki lambda* ($\Lambda_v^\downarrow = \max_{v \in \mathcal{N}_u} \{L_u(t) - L_u^v(t)\}$) değerlerinin hesaplanabilmesi için, *Mantıksal Saat Devresi*'nden gelen mantıksal saat değeri de *Tahminleme Devresi*'ne girdi olarak verilmelidir.

Algoritma 8.5, *Eşzamanlama Devresi* tarafından gerçekleştirilmektedir. κ değeri ve *Tahminleme Devresi*'nden gelen *ilerdeki lambda* ve *gerideki lambda* bu devrenin girdileridir. Bu değerler kullanılarak mod fonksiyonunu donanımsal olarak gerçekleştirilmekte ve mantıksal saatin hız çarpanını seçen mod seçicisi çıktı olarak verilmektedir.

Özetlenecek olursa, sunulan taslak donanım tasarımı, yazılımsal gerçekleştirimde her donanım saat tığında yazılım tarafından yapılan işlemleri, özel donanım devreleri ile yapmaktadır. Yazılım/donanım işbirliğine dayalı bu tasarım, SEB bileşeninin sadece yazılımla olan gerçekleştirimindeki işlemci yükünü azaltmakla beraber, sistemin donanım karmaşıklığını arttırmaktadır.

8.7 Sonuçlar ve Tartışma

Literatürdeki optimal meyilli saat eşzamanlama algoritmaları, uygulamada gerçekleştirilmeleri zor olan ifadeler içermektedirler. Bu bölümde, bu algoritmalarda yer alan uygulamada gerçekleştirimi zor olan ifadeler yerine, algoritmaların optimallik özelliklerini bozmayacak daha kolay gerçekleştirilebilir

ifadeler sağlanmıştır. Algoritmaların yürüttüğü görevler ayrı ve gerçekleştirilebilir modüllere ayrılarak, yeni ve daha kolay uygulanabilir bir meyilli saat eşzamanlama sistemi tanıtılmıştır. Sistemin hem yazılımla hem de yazılım/donanım işbirliğiyle nasıl gerçekleştirilebileceği tartışılmıştır. Bu çalışmanın dağıtık sistemlerde meyilli saat eşzamanlamasının daha iyi anlaşılmasını sağladığı ve ilerdeki yazılımsal ve donanımsal tasarımlara rehberlik edeceği düşünülmektedir. Gelecek çalışmalar, sistemin sunulan donanım tasarımının gerçekleştirimini ve başarımını ölçmeyi hedefleyebilir.

8.8 Yardımcı Teorem 8.1'in İspatı

t_s zamanı, $v \in \mathcal{N}_u$ düğümünün $\langle SYNC, L_v, seq_v \rangle$ mesajını u düğümüne gönderdiği zaman ve t_{rcv} zamanı ise u düğümünün bu mesajı aldığı zaman olsun. $t_{rcv} \leq t$ zamanı ise bu mesaj alındıktan sonra, u düğümünün v düğümünden yeni bir eşzamanlama mesajı almasından önceki en son an olsun. Dolayısıyla,

$$\partial \mathcal{H} \leq H_v(t) - H_v(t_s) \leq \partial \mathcal{H} + \frac{\mathcal{T}}{1 - \varepsilon} \quad (8.11)$$

eşitsizliği sağlanmalıdır. \mathcal{A}^{tahmin} algoritmasına göre $t_{rcv} \in [t_s, t_s + \mathcal{T}]$ zamanında L_u^v değeri alınan L_v değeri ile güncellenmektedir ve aynı düğümünden bir sonraki mesaj alımına kadarki $[t_{rcv}, t]$ zaman diliminde u düğümünün donanım saatinin hızında ilerletilmektedir. Mantıksal saatler ise en fazla $(1 + \varepsilon)\beta$ ve en az $(1 - \varepsilon)$ hızında ilerleyecekleri için, u düğümünün v düğümünün mantıksal saati hakkındaki tahmini aşağıda gösterildiği gibi alttan sınırlandırılabilir:

$$\begin{aligned}
L_u^v(t) &\geq L_u^v(t_{rcv}) + (1 - \varepsilon)(t - t_{rcv}) \\
&\geq L_u^v(t_{rcv}) + \frac{(1 - \varepsilon)(H_v(t) - H_v(t_{rcv}))}{1 + \varepsilon} \\
&= L_u^v(t_{rcv}) + \frac{(1 - \varepsilon)(H_v(t) - H_v(t_s))}{1 + \varepsilon} - \frac{(1 - \varepsilon)(H_v(t_{rcv}) - H_v(t_s))}{1 + \varepsilon} \\
&\geq L_u^v(t_{rcv}) + \frac{(1 - \varepsilon)(H_v(t) - H_v(t_s)) - \mathcal{T}}{1 + \varepsilon} \\
&= L_v(t_s) + \frac{(1 - \varepsilon)(H_v(t) - H_v(t_s)) - \mathcal{T}}{1 + \varepsilon} \\
&= L_v(t) - \beta(H_v(t) - H_v(t_s)) + \frac{(1 - \varepsilon)(H_v(t) - H_v(t_s)) - \mathcal{T}}{1 + \varepsilon} \\
&\geq L_v(t) - (\beta - 1 + 2\varepsilon)(H_v(t) - H_v(t_s)) - \frac{\mathcal{T}}{1 + \varepsilon} \\
&\geq L_v(t) - (\beta - 1 + 2\varepsilon) \left(\partial \mathcal{H} + \frac{\mathcal{T}}{1 - \varepsilon} \right) - \frac{\mathcal{T}}{1 + \varepsilon} \\
&\geq L_v(t) - \partial \bar{\mathcal{H}} - \frac{\beta + 2\varepsilon}{1 - \varepsilon} \mathcal{T} \\
&= L_v(t) - \epsilon.
\end{aligned} \tag{8.12}$$

Benzer şekilde, $L_u^v(t)$ tahmini için bir üst sınır aşağıdaki şekilde elde edilebilir:

$$\begin{aligned}
L_u^v(t) &\leq L_u^v(t_{rcv}) + (1 + \varepsilon)(t - t_{rcv}) \\
&\leq L_u^v(t_{rcv}) + \frac{(1 + \varepsilon)(H_v(t) - H_v(t_{rcv}))}{1 - \varepsilon} \\
&\leq L_u^v(t_{rcv}) + \frac{(1 + \varepsilon)(H_v(t) - H_v(t_s))}{1 - \varepsilon} \\
&= L_v(t_s) + \frac{(1 + \varepsilon)(H_v(t) - H_v(t_s))}{1 - \varepsilon} \\
&\leq L_v(t) - \frac{(1 - \varepsilon)(H_v(t) - H_v(t_s))}{1 + \varepsilon} + \frac{(1 + \varepsilon)(H_v(t) - H_v(t_s))}{1 - \varepsilon} \\
&\leq L_v(t) + \frac{4\varepsilon}{(1 + \varepsilon)(1 - \varepsilon)} (H_v(t) - H_v(t_s)) \\
&< L_v(t) + \partial \bar{\mathcal{H}} + \frac{\beta + 2\varepsilon}{1 - \varepsilon} \mathcal{T} \\
&= L_v(t) + \epsilon.
\end{aligned}$$

8.9 Teorem 8.2'nin İspatı

Teorem 8.2'nin ispatı için bazı ek tanımlara ihtiyaç duyulmaktadır. G çizgesindeki tüm yollar \mathcal{P} ile gösterilmektedir. Ek olarak $P(u) \in \mathcal{P}$, G çizgesinde u düğümü ile başlayan tüm yolları göstermektedir. $P = u, \dots, v$, yolu için ℓ_P yolun başındaki u düğümü ile yolun sonundaki v düğümü arasındaki uzaklığı yani $d(u, v)$ değerini göstermektedir. Verilen bir zaman diliminde mantıksal saatteki ilerleme \mathcal{I} ile gösterilip matematiksel olarak aşağıdaki şekilde tanımlanmaktadır.

Tanım 8.3. Bir u düğümü ve bir $[t_1, t_2]$ zaman dilimi verildiğinde, bu zaman dilimi içerisinde L_u mantıksal saatinin ilerleme miktarı aşağıda tanımlanmıştır:

$$\mathcal{I}_u(t_1, t_2) = L_u(t_2) - L_u(t_1). \quad (8.13)$$

Burada sunulacak ispat, (Kuhn and Oshman, 2009) çalışmasındaki ispat yöntemi temel alınarak yapılmıştır. Kuhn et al. (Kuhn and Oshman, 2009) kendi ispatlarının (Lenzen et al., 2010) çalışmasındaki ispatı temel aldığını ancak kendi ispatlarının daha temiz ve anlaşılır olduğunu belirtmişlerdir. Biz de onların ispatını daha da temiz ve anlaşılır bir hale getirmeye çalıştık.

Ortalama saat farkı, $P = u \dots v$ yolundaki bir kenar üzerindeki saat farkının yolun uzunluğu göz önüne alınarak hesaplanmasını sağlamaktadır. Bu kavram bir u düğümün mantıksal saatinin v düğümünden ne kadar ileride ya da ne kadar geride olduğunu ölçmeyi ve düğümlerin mantıksal saatlerinin hız çarpanlarının tespit edilebilmesini sağlamaktadır.

Tanım 8.4. $P = u, \dots, v$ yolu ve bir t zamanı verildiğinde, P yolunda bulunan her kenardaki **ortalama saat farkı** Δ_P ve **ters ortalama saat farkı** ∇_P aşağıda gösterildiği gibi tanımlanmaktadır:

$$\Delta_P(t) = \frac{L_u(t) - L_v(t)}{d(u, v)\kappa}, \quad (8.14)$$

$$\nabla_P(t) = \frac{L_v(t) - L_u(t)}{d(u, v)\kappa}. \quad (8.15)$$

Bir $\gamma \in \mathbb{R}$ verildiğinde, Δ_P^γ ve ∇_P^γ değerlerinin t anındaki değerleri aşağıda

gösterildiği gibi tanımlanmaktadır:

$$\Delta_P^\gamma(t) = \Delta_P(t) - \gamma, \quad (8.16)$$

$$\nabla_P^\gamma(t) = \nabla_P(t) - \gamma. \quad (8.17)$$

Bunlara ek olarak t anındaki \blacktriangle_u^γ ve $\blacktriangledown_u^\gamma$ değerleri ise aşağıda gösterildiği gibi tanımlanmaktadır:

$$\blacktriangle_u^\gamma(t) = \max_{P' \in P(u)} \left\{ \ell_{P'} \Delta_{P'}^\gamma(t) \right\}, \quad (8.18)$$

$$\blacktriangledown_u^\gamma(t) = \max_{P' \in P(u)} \left\{ \ell_{P'} \nabla_{P'}^\gamma(t) \right\}. \quad (8.19)$$

Temel tanımlar verildikten sonra, şimdi *en büyük ortalama saat farkına* sahip yolu göz önüne alan bir yardımcı teorem sunulmaktadır. Eğer bu yolun sonunda bulunan düğümün mantıksal saati bu yolun başındaki düğümün mantıksal saatinden çok gerideyse, bu düğümün mod fonksiyonu \mathcal{F} sıfırdan büyük ya da eşittir. Dolayısıyla, bu düğümün mantıksal saatinin hız çarpanı β değerine sahiptir.

Lemma 8.5. (*Hızlı İlerleme*) $P = u \dots v$, bir $s \in \mathbb{N}$ tamsayısı için aşağıdaki eşitliğin sağlandığı bir yol olsun:

$$\blacktriangle_u^{s-\frac{1}{2}}(t) = \ell_P \Delta_P^{s-\frac{1}{2}}(t) > 0. \quad (8.20)$$

Bu durumda v düğümünün mantıksal saatinin hız çarpanı β değerine sahiptir.

İspat. 8.20 eşitsizliği sağlandığı için, tüm $P_{all} \in P(u)$ yolları için 8.21 eşitsizliği de sağlanmaktadır:

$$\ell_P \Delta_P^{s-\frac{1}{2}}(t) \geq \ell_{P_{all}} \Delta_{P_{all}}^{s-\frac{1}{2}}(t). \quad (8.21)$$

v düğümünün komşuları, u düğümüne v düğümünün u düğümüne uzaklığından en çok bir sekme uzakta ya da en çok bir sekme yakında olabilirler. $\ell_P \Delta_P^{s-\frac{1}{2}}(t) > 0$ eşitsizliği sağlandığı için, $\ell_P > 0$ ve dolayısıyla $u \neq v$ eşitsizliğine ulaşılır. Yani v düğümünün, u düğümüne v düğümünden 1 sekme daha yakın en az bir komşusu vardır. $P' = u \dots v'$, $v' \in \mathcal{N}_v$ ve $d(u, v') = d(u, v) - 1$ olan bir yol olsun. Eğer 8.21 eşitsizliği P ve P' yolları için düzenlenirse, aşağıdaki eşitsizlik elde edilir:

$$\frac{L_{v'}(t) - L_v(t)}{\kappa} \geq s - \frac{1}{2}. \quad (8.22)$$

Yukarıdaki eşitsizliğe 8.1 yardımcı teoremi uygulanarak aşağıdaki eşitsizlik elde edilir:

$$\frac{L_v^{v'}(t) - L_v(t)}{\kappa} > s - \frac{1}{2} - \frac{\epsilon}{\kappa} \stackrel{\epsilon/\kappa < \frac{1}{4}}{>} s - \frac{3}{4}. \quad (8.23)$$

$\max_{i \in \mathcal{N}_v} \{L_v^i(t) - L_v(t)\}$ değeri her zaman $L_v^{v'}(t) - L_v(t)$ değerinden büyük ya da eşit olacağı için, aşağıdaki eşitsizliğe ulaşılmaktadır:

$$\left\lfloor \frac{\Lambda_v^\uparrow}{\kappa} - \frac{1}{4} \right\rfloor \geq s - 1. \quad (8.24)$$

Şimdi v düğümünün herhangi bir komşusunda biten $P'' = u \dots v''$ yolu göz önüne alınsın. Tüm $v'' \in \mathcal{N}_v$ için $d(u, v'') \leq d(u, v) + 1$ sağlanacağından, 8.21 eşitsizliği P ve P'' yolları için düzenlenecek olursa aşağıdaki eşitsizliğe ulaşılır:

$$\frac{L_v(t) - L_{v''}(t)}{\kappa} \leq s - \frac{1}{2}. \quad (8.25)$$

Yukarıdaki eşitliğe 8.1 yardımcı teoremi uygulanırsa, $\frac{L_v(t) - L_{v''}^{v''}(t)}{\kappa} < s - \frac{1}{4}$ eşitsizliği elde edilir. Bu üst sınır v düğümünün tüm komşuları için geçerli

olduğundan, aşağıdaki eşitsizliğe ulaşılır:

$$\left\lfloor \frac{\Lambda_v^\downarrow}{\kappa} + \frac{1}{4} \right\rfloor \leq s - 1.$$

Bu sonuç 8.24 eşitsizliği ile beraber kullanılırsa, $\mathcal{F}_v(t) \geq 0$ elde edilir ve Algoritma 8.5'in 1-2 satırlarına göre v düğümünün mantıksal saatinin hız çarpanı β değerine sahiptir. \square

Aşağıdaki yardımcı teorem en büyük *ters ortalama saat farkı* değerine sahip yolu göz önüne almaktadır. Eğer bu yolun sonundaki düğümün mantıksal saati başındaki düğümün mantıksal saatinden çok ilerideyse, bu düğümün mod fonksiyonu sıfırdan küçüktür. Dolayısıyla, bu düğümün mantıksal saatinin hız çarpanı 1 değerine sahiptir.

Lemma 8.6. (*Normal İlerleme*) $P = u, \dots, v$, bir $s \in \mathbb{N}$ tamsayısı için aşağıdaki eşitsizliğin sağlandığı bir yol olsun:

$$\nabla_u^s(t) = \ell_P \nabla_P^s(t) > 0. \quad (8.26)$$

Bu durumda v düğümünün mantıksal saatinin hız çarpanı 1 değerine sahiptir.

İspat. Bu yardımcı teorem, 8.5 yardımcı teoremine benzer şekilde ispatlanabilmektedir. \square

Önceki iki yardımcı teorem, düğümlerin mantıksal saatlerinin hız çarpanlarının ne zaman β ya da 1 olduğunu ayırt etmeyi sağlayacak koşulları ortaya koymuştur. Bu gerçekler kullanan aşağıdaki yardımcı teorem, *en büyük ters ortalama saat farkına* sahip yolun sonundaki düğümün mantıksal saati çok ileride olduğundaki durumu göz önüne almaktadır.

Lemma 8.7. $u \in V$ bir düğüm, $s \in \mathbb{N}$ bir tamsayı ve $[t', t'']$ bir zaman dilimi olsun. Eğer tüm $t \in [t', t'']$ için $\nabla_u^s(t) > 0$ ise, aşağıdaki eşitsizlik sağlanmaktadır:

$$\nabla_u^s(t'') \leq \nabla_u^s(t') - \frac{\mathcal{I}_u(t', t'')}{\kappa} + \frac{(1 + \varepsilon)(t'' - t')}{\kappa}. \quad (8.27)$$

İspat. $\mathcal{P}^{max} = P_0, P_1, \dots, P_{n-1}$, tüm $P_i \in \mathcal{P}^{max}$ için $P_i \in \mathcal{P}(u)$ olan n adet yolun kümesi olsun. $\mathcal{S}^{max} = S_0, S_1, \dots, S_{n-1}$, $\bigcup_{i=0}^{n-1} S_i = [t', t'']$ olan ve tüm $\forall t \in S_i = [t_i, t_{i+1}]$ için $\nabla_u^s(t) = \nabla_{P_i}^s(t)$ eşitliğinin sağlandığı n adet sürekli, ardışık ve kapalı gerçekte zaman aralıkları olsun. \mathcal{P}^{max} kümesinin i . elemanı $P_i = u..v$ olsun. Yardımcı teorem 8.6'e göre, v düğümünün mantıksal saatinin hızı en fazla $(1 + \varepsilon)$ 'dir. $\nabla_u^s(t_{i+1}) = \ell_{P_i} \nabla_{P_i}^s(t_{i+1})$ olduğu için, aşağıdaki eşitlik elde edilir:

$$\nabla_u^s(t_{i+1}) = \ell_{P_i} \nabla_{P_i}^s(t_i) - \frac{\mathcal{I}_u(t_i, t_{i+1}) - \mathcal{I}_v(t_i, t_{i+1})}{\kappa}. \quad (8.28)$$

Dolayısıyla, 8.29 eşitsizliğine ulaşılmaktadır:

$$\nabla_u^s(t_{i+1}) \leq \nabla_u^s(t_i) - \frac{\mathcal{I}_u(t_i, t_{i+1})}{\kappa} + \frac{(1 + \varepsilon)(t_{i+1} - t_i)}{\kappa}. \quad (8.29)$$

$t_0 = t'$ ve $t_{n-1} = t''$ olduğu için, 8.29 eşitsizliğini \mathcal{P}^{max} kümesinin tüm elemanları için yazarsak ve toplarsak istenen sonuç elde edilir. \square

Ana teoremi ispatlayabilmek için, ağ uygun durumdayken, her $s \in \mathbb{N}$ için $\blacktriangle_u^{s-\frac{1}{2}}$ değeri üstten sınırlandırılmalıdır. Aşağıdaki yardımcı teorem, bu üst sınırı ortaya koymaktadır.

Lemma 8.8. *Eğer ağ t anında uygun durumda ise, verilen bir $s \in \mathbb{N}$ için aşağıdaki eşitsizlik sağlanmaktadır:*

$$\blacktriangle_u^{s-\frac{1}{2}}(t) < \frac{2\mathcal{D}}{\sigma^{s-1}}. \quad (8.30)$$

İspat. $P = u..v$, $\blacktriangle_u^{s-\frac{1}{2}}(t) = \ell_P \Delta_P^{s-\frac{1}{2}}(t)$ eşitliğinin sağlandığı bir yol olsun. $m \in \mathbb{N}_0$, $m = \left\lfloor \log_{\sigma} \frac{2\mathcal{D}}{d(u,v)} \right\rfloor$ eşitliğini sağlayan bir tamsayı olsun. Dolayısıyla $d(u, v) \leq$

$\frac{2\mathcal{D}}{\sigma^m}$ eşitsizliği de sağlanmaktadır. t anında ağ uygun durumda olduğu için, Δ_P 'nin tanımında $\Delta_P(t) < m + \frac{3}{2}$ eşitsizliği elde edilir. $\blacktriangle_u^{s-\frac{1}{2}}(t) = \ell_P \Delta_P^{s-\frac{1}{2}}(t) = d(u, v) \Delta_P^{s-\frac{1}{2}}(t)$ eşitliğine ve $\Delta_P^{s-\frac{1}{2}}(t) < (m - s + 2)$ eşitliğine sahip olduğu için, aşağıdaki eşitsizlik de sağlanmaktadır:

$$\begin{aligned} \blacktriangle_u^{s-\frac{1}{2}}(t) &< d(u, v) (m - s + 2) \\ &\leq \frac{2\mathcal{D}}{\sigma^m} (m - s + 2) \\ &= \frac{2\mathcal{D}}{\sigma^{s-1}} \left(\frac{m - s + 2}{\sigma^{m-s+1}} \right) < \frac{2\mathcal{D}}{\sigma^{s-1}}. \end{aligned} \quad (8.31)$$

Son adım, eğer $m - s \geq 0$ ya da $m - s < 0$ ise, $\sigma \geq 2$ olduğu için $\frac{m-s+\frac{3}{2}}{\sigma^{m-s+1}} \leq 1$ olacağından sağlanmaktadır. \square

Aşağıdaki teorem yardımıyla, \mathcal{A}^{meyil} algoritmasının uygun durumun hiçbir zaman dışına çıkmadığını ve düğümler arasındaki uzaklığın logaritmasının bir fonksiyonu olan yerel saat farkına sahip olduğu ispatlanmaktadır.

Teorem 8.9. *Ağ her zaman uygun durumdadır.*

İspat. Bu teorem olmayana ergi yöntemini kullanacaktır. t_1 zamanı ağın uygun durumunun ihlal edildiği zamanların en küçüğü olsun. Bu anda, ihlalin $P = u \dots v$ yolu tarafından yapıldığı varsayalım. s , $s = \left\lfloor \log_{\sigma} \frac{2\mathcal{D}}{d(u, v)} \right\rfloor$ eşitliğini sağlayan bir tamsayı olsun. Dolayısıyla aşağıdaki eşitsizlik sağlanmaktadır:

$$\frac{2\mathcal{D}}{\sigma^s} \geq d(u, v) > \frac{2\mathcal{D}}{\sigma^{s+1}}. \quad (8.32)$$

$P_r = v, \dots, u, P$ yolunun ters yolu olsun. $L_u(t_1) - L_v(t_1) > d(u, v) \left(s + \frac{1}{2}\right) \kappa$ eşitsizliği sağlandığı için,

$$\blacktriangledown_v^s(t_1) \geq \ell_{P_r} \nabla_{P_r}^s(t_1) > \frac{d(u, v)}{2} \quad (8.33)$$

eşitsizliği de sağlanmış olmaktadır. t_0 anı, $t_0 < t_1$ olmak üzere, $\blacktriangledown_v^s(t_0) \leq 0$ olan ve

∇_v^s değerinin $[t_0, t_1)$ aralığında $\nabla_v^s(t_0)$ değerini değişmeden koruduğu zamanların en büyüğü olsun. 8.7 yardımcı teoremi kullanılarak, aşağıdaki eşitsizlik gösterilebilir:

$$\begin{aligned} \nabla_v^s(t_1) &\leq \nabla_v^s(t_0) - \frac{\mathcal{I}_u(t_0, t_1)}{\kappa} + \frac{(1 + \varepsilon)(t_1 - t_0)}{\kappa} \\ &\leq \nabla_v^s(t_0) - \frac{(1 - \varepsilon)(t_1 - t_0)}{\kappa} + \frac{(1 + \varepsilon)(t_1 - t_0)}{\kappa} \\ &\stackrel{\nabla_v^s(t_0) \leq 0}{\leq} \frac{2\varepsilon(t_1 - t_0)}{\kappa}. \end{aligned} \quad (8.34)$$

Yukarıdaki eşitsizlik ve 8.33 eşitsizliği birlikte kullanılarak,

$$d(u, v) < \frac{4\varepsilon(t_1 - t_0)}{\kappa} \quad (8.35)$$

sonucuna ulaşılmaktadır.

Şimdi $\mathcal{I}_v(t_0, t_1)$ için bir alt sınır bulunacaktır. $P' = v, \dots, k$, $\ell_{P'} \nabla_{P'}^s(t_0) = \nabla_v^s(t_0)$ eşitliğinin sağlandığı bir yol olsun. Bu bölümün başında verilen tanımlar kullanılarak, aşağıdaki eşitsizlik elde edilebilir:

$$\begin{aligned} \ell_{P'} \nabla_{P'}^s(t_1) &= \nabla_v^s(t_0) - \frac{\mathcal{I}_v(t_0, t_1)}{\kappa} + \frac{\mathcal{I}_k(t_0, t_1)}{\kappa} \\ &\leq \blacktriangle_k^{s-\frac{1}{2}}(t_1). \end{aligned} \quad (8.36)$$

Yukarıdaki eşitsizlik düzenlenirse, aşağıdaki eşitsizlik elde edilir:

$$\begin{aligned} \nabla_v^s(t_0) - \frac{\mathcal{I}_v(t_0, t_1)}{\kappa} &\leq \blacktriangle_k^{s-\frac{1}{2}}(t_1) - \frac{\mathcal{I}_k(t_0, t_1)}{\kappa} \\ &< \frac{2\mathcal{D}}{\sigma^{s-1}} - \frac{\mathcal{I}_k(t_0, t_1)}{\kappa} \\ &\leq \frac{2\mathcal{D}}{\sigma^{s-1}} - \frac{(1 - \varepsilon)\beta(t_1 - t_0)}{\kappa}. \end{aligned} \quad (8.37)$$

Bu üst sınır, 8.7 yardımcı teoreminde kullanılırsa, aşağıdaki eşitsizlik elde edilebilir:

$$\begin{aligned}
\nabla_v^s(t_1) &\leq \nabla_v^s(t_0) - \frac{\mathcal{I}_v(t_0, t_1)}{\kappa} + \frac{(1 + \varepsilon)(t_1 - t_0)}{\kappa} \\
&\leq \nabla_v^s(t_0) - \frac{\mathcal{I}_v(t_0, t_1)}{\kappa} + \frac{(1 + \varepsilon)(t_1 - t_0)}{\kappa} \\
&< \frac{2\mathcal{D}}{\sigma^{s-1}} + \frac{(1 + \varepsilon)(t_1 - t_0) - (1 - \varepsilon)\beta(t_1 - t_0)}{\kappa} \\
&\stackrel{\frac{2\mathcal{D}}{\sigma^{s-1}} \leq \sigma^2 d(u, v)}{\leq} \sigma^2 d(u, v) + \frac{(1 + \varepsilon - (1 - \varepsilon)\beta)(t_1 - t_0)}{\kappa} \\
&< \frac{4\varepsilon(t_1 - t_0)}{\kappa} + \frac{(1 + \varepsilon - (1 - \varepsilon)\beta)(t_1 - t_0)}{\kappa} \\
&\stackrel{\beta \geq 4\sigma^2 \frac{\varepsilon}{1-\varepsilon} + 1}{<} \frac{2\varepsilon(t_1 - t_0)}{\kappa} \leq \frac{d(u, v)}{2}.
\end{aligned} \tag{8.38}$$

Bu durum 8.33 eşitsizliği göz önüne alındığında bir çelişkidir. \square

“Zaman, sessiz bir testeredir.”

Immanuel KANT

9 SONUÇLAR

Mikro-elektromekanik (MEMS) teknolojisindeki ilerlemeler, kablosuz iletişim yeteneğine sahip küçük algılayıcı cihazlarının doğmasına yol açmıştır. Birçok algılayıcı düğümünün bir araya gelmesinden oluşan KAA’nda, çoğu uygulamanın ve protokolün doğru ve verimli bir şekilde çalışabilmesi, düğümlerin saatlerinin eşzamanlanmış olmasını gerektirmektedir. Geleneksel dağıtık sistemler için tasarlanmış saat eşzamanlama protokolleri, enerji verimliliği, devingenlik, hatalara duyarlılık gibi konulara fazla önem vermediği için KAA’nda uygulanamazlar. Dolayısıyla saat eşzamanlaması, KAA’nın getirdiği yeni gereksinimler nedeniyle birçok yeni açık problemi barındırmaktadır.

Bu tez kapsamında, KAA için üç açık problem göz önüne alınmıştır. Ek olarak, daha genel dağıtık sistemler kapsamında meyilli saat eşzamanlamasına yönelik bir çalışma da ortaya koyulmuştur.

9.1 Katkıların Özeti

Sunulan tezde, öncelikle KAA’nda genel geçer eşzamanlama protokolü olan FTSP incelenmiştir. Bu protokolde düğümlerin kendilerine gelen zaman bilgisini alır almaz tümegöndermemeleri nedeniyle, zaman bilgisi ağ genelinde yavaş bir şekilde yayılmaktadır. Bu durum incelenerek zaman bilgisinin yavaş yayılım hızının, başka bir deyişle yavaş selin, saat eşzamanlamasına olan olumsuz etkisi ortaya koyulmuştur. Literatürde, bu olumsuz etkinin selin yayılımını hızlandırarak giderilmesi önerilmiştir. Ancak tezde sunulan çalışma ile, selin yayılım hızını değiştirmeyerek ve düğümler arasında bir uzlaşma algoritması yürüterek, zaman bilgisinin ağ genelinde yavaş bir şekilde yayılması ile de oldukça iyi bir eşzamanlama başarımının elde edilebileceği gösterilmiştir. Bu çalışma ile KAA topluluğuna, yavaş sel yöntemi ile mikrosaniye mertebesinde hassasiyetle eşzamanlama sağlayan, uygulamada rahatlıkla kullanılabilir ve açık kaynak koda sahip bir protokol kazandırılmıştır.

Uygulamadaki saat eşzamanlama protokolleri incelenirken, en küçük kareler yönteminin çoğu protokolün yürütümünde düğümlerin saatlerinin geri alınmasına yol açtığı gözlemlenmiştir. Bu durumun uygulamada birçok probleme yol

açabileceği ve saatlerin monoton artan bir fonksiyon olduğu varsayımı ile çelişki olduğu ortaya koyulmuştur. Bu durumu engelleyecek bir yöntem önerilmiştir.

Tezin KAA'na yönelik saat eşzamanlaması için bir diğer katkısı, hem dışsal eşzamanlamaya hem de komşular arasındaki eşzamanlama hatasının eniyilemesine ihtiyaç duyan uygulamalar tarafından kullanılabilir bir saat eşzamanlama protokolü tanıtmasıdır. Bu çalışma ile KAA topluluğuna, birbirine komşu düğümler arasındaki eşzamanlama hatasının ağırlığının büyümesinden çok az etkilendiği, mikrosaniye mertebesinde hassasiyetle dışsal eşzamanlama sağlayan, uygulamada rahatlıkla kullanılabilir ve açık kaynak koda sahip bir protokol daha kazandırılmıştır.

Sunulan tez, genel dağıtık sistemler için, düğümler arasındaki eşzamanlama hatasının düğümler arasındaki uzaklığın bir fonksiyonu olduğu meyilli saat eşzamanlaması konusuna da bir katkı yapmıştır. Literatürde yer alan optimal meyilli algoritmaların içerdiği uygulamada gerçekleştirimi zor olan ifadeler elenmiş, daha kolay gerçekleştirilebilir ve birbirinden bağımsız modüllerden oluşan yeni bir meyilli saat eşzamanlama sistemi tanıtılmıştır. Bu sistemi gerçekleştirecek bir taslak donanım tasarımı da verilmiştir.

9.2 Gelecek Çalışmalar

Tez kapsamındaki çalışmalar göz önüne alındığında, üzerinde durulması gereken ve çözüldüğü takdirde literatüre ciddi bir katkı olarak görülen iki önemli konu bulunmaktadır.

Uzlaşım Algoritmasının Yakınsama Süresi: Selin yavaş yayılımının eşzamanlama üzerindeki olumsuz etkisini gidermeye yönelik çalışmada ve hem dışsal eşzamanlamayı hem de komşular arasındaki eşzamanlama hatasının eniyilemesini hedefleyen çalışmada, düğümler bir uzlaşma algoritması yürütmektedirler. Bu uzlaşma algoritmasının yakınsama süresi, ağ büyüdükçe oldukça büyümektedir. Yakınsama süresini küçültmek, önemli bir gelecek çalışma olarak durmaktadır.

Meyilli Saat Sisteminin Donanım Gerçekleştirimi: Önerilen meyilli saat eşzamanlama sisteminin, taslak donanım tasarımının ayrıntılı bir şekilde gerçek donanıma dönüştürülmesi ve başarımının ölçülmesi, çok önemli bir çalışma olarak görülmektedir. Literatürde, optimal meyilli saat eşzamanlama algoritmaları için gerçekleştirilmiş bir donanım mevcut değildir.

Bu iki konu dışında, literatürdeki saat eşzamanlaması çalışmalarının *enerji verimliliği* açısından daha detaylı analizi ve eniyilenmesi, üzerinde durulması gereken gelecek çalışmalarından biri olarak görülmektedir. Örneğin, düğümlerin komşu sayılarını (derecelelerini) gözeterek ya da zaman bilgisinin yayılımını en küçük enerjili tümegönderim ağacı²⁵ ile yaparak saat eşzamanlaması sağlamak, saat eşzamanlaması için ağ genelinde harcanan toplam enerjiyi azaltabilir. Benzer şekilde, diğer düğümler için bir zaman kaynağı olan dayanak düğümü ağın merkezindeki düğümlerden biri olarak seçilebilir. Bu yöntemin, zaman bilgisinin yayılımı için harcanan toplam enerjiyi azaltacağı düşünülmektedir. Bunlar ve benzeri yöntemlerin uygulanabilirliğinin sorgulanması, mümkünse uygulanması ve birbirleriyle karşılaştırılması, KAA'ndaki saat eşzamanlama konusuna ciddi bir katkı olarak düşünülmektedir.

9.3 Sonuç

Tezde sunulan çalışmaların askeri amaçlar için kullanılmaması ve insanlığa zarar getirmemesi, ilerdeki araştırmalar için faydalı ve yol gösterici olması umulmaktadır. Yaşamın teknoloji için değil, teknolojinin ve bilimin yaşam için olduğu bir dünya ümidiyle, insanın insanca yaşayabilmesine bir katkıda bulunabildiysek ve bilimsel anlamda filizlenebilecek bir tohum ekebildiysek, bundan büyük bir onur duyuyor olacağız.

²⁵İngilizcesi: *Minimum Energy Broadcast Tree*.

KAYNAKLAR DİZİNİ

- Akyildiz, I. F. and Vuran, M. C.:** 2010, *Wireless Sensor Networks*, John Wiley & Sons
- Biaz, S. and Welch, J. L.:** 2001, Closed form bounds for clock synchronization under simple uncertainty assumptions, *Elsevier Information Processing Letters* **80**, 151
- Dai, H. and Han, R.:** 2004, Tsync: A lightweight bidirectional time synchronization service for wireless sensor networks, *ACM SIGMOBILE Mobile Computing and Communications Review* **8**, 125
- Dietz, E. J.:** 1987, A comparison of robust estimators in simple linear regression, **16**, 1209
- Elson, J., Girod, L., and Estrin, D.:** 2002, Fine-grained network time synchronization using reference broadcasts, *SIGOPS Oper. Syst. Rev.* **36(SI)**, 147
- Fan, R., Chakraborty, I., and Lynch, N. A.:** 2005, Clock synchronization for wireless networks, *Lecture Notes in Computer Science* **3544**, 400
- Fan, R. and Lynch, N.:** 2006, Gradient clock synchronization, *Distrib. Comput.* **18(4)**, 255
- Ganeriwal, S., Kumar, R., and Srivastava, M. B.:** 2003, 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, 138–149 pp
- Gelman, A. and Park, D.:** 2009, Splitting a Predictor at the Upper Quarter or Third and the Lower Quarter or Third, *The American Statistician* **63(1)**, 1
- Halpern, J. Y., Megiddo, N., and Munshi, A. A.:** 1985, Optimal precision in the presence of uncertainty, in *STOC '85: Proceedings of the seventeenth annual ACM symposium on Theory of computing*, New York, NY, USA, ACM, 346–355 pp
- Karl, H. and Willig, A.:** 2005, *Protocols and Architectures for Wireless Sensor Networks*, John Wiley & Sons
- Kuhn, F., Lenzen, C., Locher, T., and Oshman, R.:** 2010, Optimal Gradient Clock Synchronization in Dynamic Networks, in *29th Symposium on Principles of Distributed Computing (PODC)*, Zurich, Switzerland
- Kuhn, F., Locher, T., and Oshman, R.:** 2009, Gradient Clock Synchronization in Dynamic Networks, in *21st ACM Symposium on Parallelism in Algorithms and*

Architectures (SPAA), Calgary, Canada

- Kuhn, F. and Oshman, R.:** 2009, Gradient clock synchronization using reference broadcasts, *CoRR* abs/0905.3454
- Kusy, B., Dutta, P., Levis, P., Maroti, M., Ledeczi, A., and Culler, D.:** 2006, Elapsed time on arrival: A simple and versatile primitive for canonical time synchronisation services, *Int. J. Ad Hoc Ubiquitous Comput.* **1(4)**, 239
- lae Noh, K., Serpedin, E., and Qaraqe, K.:** 2008, A new approach for time synchronization in wireless sensor networks: Pairwise broadcast synchronization, *Wireless Communications, IEEE Transactions on* **7(9)**, 3318
- Lenzen, C.:** 2011, *Synchronization and Symmetry Breaking in Distributed Systems, Ph.D. thesis*
- Lenzen, C., Locher, T., and Wattenhofer, R.:** 2008, Clock Synchronization with Bounded Global and Local Skew, in *49th Annual IEEE Symposium on Foundations of Computer Science (FOCS), Philadelphia, Pennsylvania, USA*
- Lenzen, C., Locher, T., and Wattenhofer, R.:** 2009a, Tight Bounds for Clock Synchronization, in *28th ACM Symposium on Principles of Distributed Computing (PODC), Calgary, Canada*
- Lenzen, C., Locher, T., and Wattenhofer, R.:** 2010, Tight Bounds for Clock Synchronization, in *Journal of the ACM, Volume 57, Number 2*
- Lenzen, C., Sommer, P., and Wattenhofer, R.:** 2009b, Optimal Clock Synchronization in Networks, in *7th ACM Conference on Embedded Networked Sensor Systems (SenSys), Berkeley, California, USA*
- Levis, P., Madden, S., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., Gay, D., Hill, J., Welsh, M., Brewer, E., and Culler, D.:** 2004, Tinyos: An operating system for sensor networks, in *Ambient Intelligence*, Springer Verlag
- Locher, T. and Wattenhofer, R.:** 2006, Oblivious Gradient Clock Synchronization, in *20th International Symposium on Distributed Computing (DISC), Stockholm, Sweden*
- Lu, J. and Whitehouse, K.:** 2009, Flash flooding: Exploiting the capture effect for rapid flooding in wireless sensor networks, in *INFOCOM 2009, IEEE*, 2491-2499 pp
- Lundelius, J. and Lynch, N. A.:** 1984, An upper and lower bound for clock synchronization, *Information and Control* **62(2/3)**, 190
- Maróti, M., Kusy, B., Simon, G., and Lédeczi, A.:** 2004, The flooding time synchronization protocol, in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, New York, NY, USA,

ACM, 39–49 pp

- Maroti, M. and Sallai, J.:** May 2008, *Packet-level time synchronization*, Technical report, TinyOS Core Working Group
- Meier, L. and Thiele, L.:** 2005, Brief announcement: Gradient clock synchronization in sensor networks, in *Twenty-Fourth Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC 2005)*, Las Vegas, Nevada, USA, 238–238 pp
- Mills, D.:** 1991, Internet time synchronization: the network time protocol, *Communications, IEEE Transactions on* **39(10)**, 1482
- Nevitt, J. and Tam, H.:** 1998, A comparison of robust and nonparametric estimators under the simple linear regression model, *Multiple Linear Regression Viewpoints* **25(1)**, 54
- Patt-Shamir, B. and Rajsbaum, S.:** 1994, A theory of clock synchronization (extended abstract), in *STOC '94: Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, New York, NY, USA, ACM, 810–819 pp
- Pussente, R. M. and Barbosa, V. C.:** 2009, An algorithm for clock synchronization with the gradient property in sensor networks, *J. Parallel Distrib. Comput.* **69(3)**, 261
- Römer, K., Blum, P., and Meier, L.:** 2005, Time synchronization and calibration in wireless sensor networks, in I. Stojmenovic (ed.), *Handbook of Sensor Networks: Algorithms and Architectures*, John Wiley & Sons, 199–237 pp
- Ross, S. M.:** 2004, *Introduction to Probability and Statistics for Engineers and Scientists, Third Edition*, Elsevier Academic Press
- Sallai, J., Kusy, B., Ledecz, A., and Dutta, P.:** 2006, On the scalability of routing integrated time synchronization, *3rd European Workshop on Wireless Sensor Networks (EWSN 2006)*
- Schenato, L. and Fiorentin, F.:** 2011, Average timesynch: a consensus-based protocol for time synchronization in wireless sensor networks, *Automatica* **47(9)**, 1878
- Schenato, L. and Gamba, G.:** 2007, A distributed consensus protocol for clock synchronization in wireless sensor network, in *IEEE Conference on Decision and Control (CDC 07)*
- Schmid, T., Charbiwala, Z., Anagnostopoulou, Z., Srivastava, M. B., and Dutta, P.:** 2010a, A case against routing-integrated time synchronization, in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, SenSys '10*, New York, NY, USA, ACM, 267–280 pp

- Schmid, T., Charbiwala, Z., Shea, R., and Srivastava, M.:** 2009, Temperature compensated time synchronization, *Embedded Systems Letters, IEEE* **1(2)**, 37
- Schmid, T., Shea, R., Charbiwala, Z., Friedman, J., Srivastava, M. B., and Cho, Y. H.:** 2010b, On the interaction of clocks, power, and synchronization in duty-cycled embedded sensor nodes, *ACM Trans. Sen. Netw.* **7**, 24:1
- Sivrikaya, F. and Yener, B.:** 2004, Time synchronization in sensor networks: a survey, *Network, IEEE* **18(4)**, 45
- Sommer, P. and Wattenhofer, R.:** 2009, Gradient Clock Synchronization in Wireless Sensor Networks, in *8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, San Francisco, USA
- Srikanth, T. K. and Toueg, S.:** 1987, Optimal clock synchronization, *J. ACM* **34(3)**, 626
- Sun, K., Ning, P., and Wang, C.:** 2006, Tinsync: secure and resilient time synchronization in wireless sensor networks, in *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security*, New York, NY, USA, ACM, 264–277 pp
- Sundararaman, B., Buy, U., and Kshemkalyani, A. D.:** 2005, Clock synchronization for wireless sensor networks: a survey, *Ad Hoc Networks* **3(3)**, 281
- van Greunen, J. and Rabaey, J.:** 2003, 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, 11–19 pp
- Werner-Allen, G., Tewari, G., Patel, A., Welsh, M., and Nagpal, R.:** 2005, Firefly-inspired sensor network synchronicity with realistic radio effects, in *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, New York, NY, USA, ACM, 142–153 pp
- Wilcox, R. R.:** 2010, *Fundamentals of Modern Statistical Methods: Substantially Improving Power and Accuracy*, Springer New York, 2nd edition
- Wu, Y.-C., Chaudhari, Q., and Serpedin, E.:** 2011, Clock synchronization of wireless sensor networks, *Signal Processing Magazine, IEEE* **28(1)**, 124

ÖZGEÇMİŞ

Adı Soyadı: Kasım Sinan YILDIRIM
Doğum Tarihi: 1981
Doğum Yeri: Mersin
Uyruğu: T.C.

EĞİTİM

Yüksek Lisans: Ege Üniversitesi
Bilgisayar Mühendisliği Bölümü, 2006
Lisans: Ege Üniversitesi
Bilgisayar Mühendisliği Bölümü, 2003
Lise: Bartın Anadolu Lisesi, 1998

İLGİ ALANLARI

İşletim Sistemleri, Gömülü Sistemler, Dağıtık Sistemler, Dağıtık Algoritmalar

TERİMLER SÖZLÜĞÜ

TÜRKÇE

algılayıcı
 arıza dayanıklılığı
 bağlı çizge
 belirlenimci olan
 benzetim
 çizge
 çift yönlü
 çok sekmeli
 dağıtık
 darbe
 dayanak
 devingen
 doğrusal bağlanım
 doğrusal ilinge
 durağan
 düğüm
 düzenleşim
 elle
 en küçük kareler yöntemi
 eniyileme
 eşgüdümlü
 eşik
 eşzamanlı
 eylem
 frekans kararsızlığı
 halka ilinge
 iletim
 ilinge
 kablosuz
 kalıcı bellek
 kararlı
 kesme
 kristal salınıcı
 kurulum
 mantıksal saat
 merkez istasyon
 mesaj gecikmesi

İNGİLİZCE

sensor
fault tolerance
connected graph
deterministic
simulation
graph
bidirectional
multi-hop
distributed
pulse
reference
dynamic
linear regression
line topology
static
node
configuration
manual
least-squares
optimize
coordinated
threshold
synchronous
action
frequency instability
ring topology
transmission
topology
wireless
flash memory
stable
interrupt
crystal oscillator
deployment
logical clock
base station
message delay

meyil	<i>gradient</i>
mikrodenetleyici	<i>micro-controller</i>
olmayana ergi	<i>proof by contradiction</i>
ortama erişim kontrolü	<i>medium access control</i>
öteleme	<i>offset</i>
özerk	<i>autonomous</i>
periyot	<i>period</i>
saat farkı	<i>clock skew</i>
salt-okunur	<i>read-only</i>
sapma	<i>drift</i>
sapma hızı	<i>drift rate</i>
sapma telafisi	<i>drift compensation</i>
servis çağırımı	<i>service call</i>
sıkı bağlı	<i>strongly connected</i>
sıradüzensel	<i>hierarchical</i>
sözde kod	<i>pseudo code</i>
süreçler arası geçiş	<i>context switch</i>
süreksizlik	<i>discontinuity</i>
tasarsız	<i>ad hoc</i>
taşma	<i>overflow</i>
tık	<i>tick</i>
tümegönderim	<i>broadcast</i>
uzlaşma	<i>agreement</i>
veri havuzu	<i>repository</i>
yayılım ağacı	<i>spanning tree</i>
yazmaç	<i>register</i>
yerel	<i>local</i>
yonga	<i>chip</i>
yük	<i>payload</i>
yürütüm	<i>execution</i>
zaman aralığı	<i>time interval</i>
zaman bölüşümlü çoklu erişim	<i>time division multiple access</i>
zaman dilimi	<i>time slot</i>
zaman etiketi	<i>timestamp</i>
zamanlayıcı	<i>timer</i>
zamanlayıcı zamanaşımı	<i>timer timeout</i>