

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**YÜKSEK HIZLI TCP TIKANIKLIK PROTOKOLLERİNİN
DEĞERLENDİRİLMESİ**

**Hazırlayan
Noor Faisal Mohammed MOHAMMED**

**Danışman
Dr. Öğr. Üyesi Serkan ÖZTÜRK**

Yüksek Lisans Tezi

**Temmuz 2018
KAYSERİ**

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**YÜKSEK HIZLI TCP TIKANIKLIK PROTOKOLLERİNİN
DEĞERLENDİRİLMESİ**

(Yüksek Lisans Tezi)

**Hazırlayan
Noor Faisal Mohammed MOHAMMED**

**Danışman
Dr. Öğr. Üyesi Serkan ÖZTÜRK**

**Temmuz 2018
KAYSERİ**

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

Adı-Soyadı: Noor Faisal Mohammed MOHAMMED

İmza:



YÖNERGEYE UYGUNLUK

Yüksek Hızlı TCP Tıkanıklık Protokollerinin Değerlendirilmesi adlı Yüksek Lisans Tezi, Erciyes Üniversitesi Lisans Üstü Tez Önerisi ve Tez Yazma Yönergesi'ne uygun olarak hazırlanmıştır.



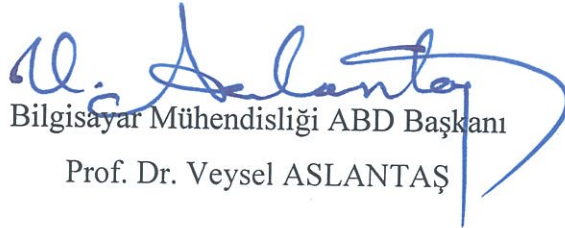
Tezi Hazırlayan

Noor Faisal Mohammed MOHAMMED



Danışman

Dr. Öğr. Üyesi Serkan ÖZTÜRK



Bilgisayar Mühendisliği ABD Başkanı
Prof. Dr. Veysel ASLANTAŞ

Dr. Öğr. Üyesi Serkan ÖZTÜRK danışmanlığında Noor Faisal Mohammed MOHAMMED tarafından hazırlanan “Yüksek Hızlı TCP Tıkanıklık Protokollerinin Değerlendirilmesi” adlı bu çalışma, jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü **Bilgisayar Mühendisliği** Anabilim Dalında **Yüksek Lisans** tezi olarak kabul edilmiştir.

09.07.2018

JÜRİ:

Danışman : Dr. Öğr. Üyesi Serkan ÖZTÜRK

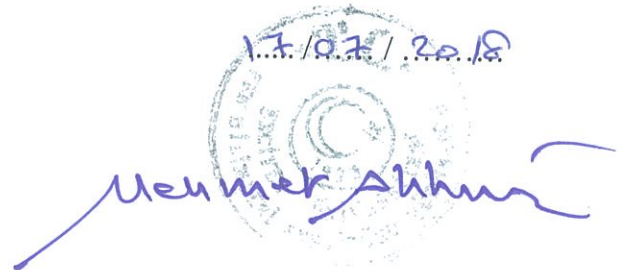
Üye : Dr. Öğr. Üyesi Mete ÇELİK

Üye : Dr. Öğr. Üyesi Ahmet Şakir DOKUZ



ONAY:

Bu tezin kabulü, Enstitü Yönetim Kurulunun 17/07/2018 tarih ve 2018/30-11 sayılı kararı ile onaylanmıştır.



Enstitü Müdürü
Prof. Dr. Mehmet AKKURT

TEŞEKKÜR

Bu çalışmanın gerçekleştirilmesinde, iki yıl boyunca değerli bilgilerini bizlerle paylaşan değerli hocam Dr. Öğr. Üyesi. Serkan ÖZTÜRK'e teşekkürlerimi sunarım. Ayrıca tüm hayatım boyunca bana destek olan değerli aileme sonsuz teşekkür ederim.

Noor Faisal Mohammed MOHAMMED

Kayseri, Temmuz 2018

YÜKSEK HIZLI TCP TIKANIKLIK PROTOKOLLERİNİN DEĞERLENDİRİLMESİ

Noor Faisal Mohammed MOHAMMED

Erciyes Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi, Temmuz 2018

Danışman: Dr. Öğr. Üyesi Serkan ÖZTÜRK

ÖZET

Paket anahtarlamalı ağlarda tıkanıklık önemli bir kavramdır. Bilgisayar ağlarında bir yönlendiricide veya bir alt ağda paket birikmesi ve ağın performansının düşmesi tıkanıklık olarak adlandırılır. Ağa gönderilen paketlerin sayısı ağın kapasitesinden fazla olması durumunda tıkanıklık meydana gelir. TCP tıkanıklık kontrol protokolleri, büyük tıkanıklık pencerelerine yavaş yanıt vermeleri nedeniyle, yüksek hızlı geniş alan ağlarında düşük performans gösterir. Bu TCP sınırlamasının üstesinden gelmek için, yüksek hızlı TCP protokolleri geliştirilmiştir.

Bu tez çalışmasında NS2 ağ benzetim programı kullanılarak, yüksek hızlı TCP tıkanıklık protokollerinden Bic-TCP, Cubic-TCP, Highspeed TCP (HS-TCP), Hamilton TCP (H-TCP) ve Scalable TCP (S-TCP) protokolleri değerlendirilmiştir. Kablolu bir ağda Random Early Detection (RED), Random Exponential Marking (REM) ve Stochastic Fairness Queueing (SFQ) kuyruk yönetim algoritmaları için yüksek hızlı TCP tıkanıklık protokolleri karşılaştırılmıştır. Düşük ve yüksek yayılım gecikmesi, bant genişliği ve paket pencere boyutu değişimine göre protokoller incelenmiştir. Düşen paketlerin oranı, throughput ve gecikme sonuçları değerlendirilmiştir.

Çalışmanın sonucunda Bic, Cubic, HS-TCP, H-TCP ve S-TCP protokollerinin SFQ kuyruk yönetim algoritması ile en iyi başarımları elde ettiği görülmüştür. Protokoller karşılaştırıldığında ise Bic-TCP protokolünün en iyi throughput, en düşük gecikme ve en az düşen paket oranı elde ettiği görülmüştür.

Anahtar Kelimeler: TCP, Bic-TCP, Cubic-TCP, HS-TCP, H-TCP, S-TCP, RED, REM, SFQ.

EVALUATION OF HIGH SPEED TCP CONGESTION CONTROL PROTOCOLS

Noor Faisal Mohammed MOHAMMED

Erciyes University
Institute of Science and Technology Department of Computer Engineering
M. Sc. Thesis, July 2018.

Supervisor: Dr. Serkan ÖZTÜRK

ABSTRACT

The congestion is an important concept in the networks. In another word, congestion occurs if the number of packets sent over the network exceeds the network capacity. TCP congestion control protocols perform poorly on high-speed wide area networks because they respond slowly to large congestion windows. In order to come up from this TCP limitation, high-speed TCP protocols have been developed.

In this thesis study, Bic-TCP, Cubic-TCP, HS-TCP, H-TCP and S-TCP high speed protocols are evaluated using NS2 network simulation program. In a wired network, high speed protocols are compared for Random Early Detection (RED), Random Exponential Marking (REM) and Stochastic Fairness Queuing (SFQ) queuing management algorithms. According to the variation of propagation delay, q and packet window size, the protocols were evaluated by the results of throughput, delay and number of dropped packets.

As a result of this study, Bic-TCP, Cubic-TCP, HS-TCP, H-TCP and S-TCP protocols achieved the best performance with SFQ queue management algorithm. When protocols are compared, it is seen that the Bic-TCP protocol achieves the best throughput, lowest delay and lowest dropped packet rate.

Keywords: TCP, Bic-TCP, Cubic-TCP, HS-TCP, H-TCP, S-TCP, RED, REM, SFQ.

İÇİNDEKİLER

YÜKSEK HIZLI TCP TIKANIKLIK PROTOKOLLERİNİN DEĞERLENDİRİLMESİ

ESayfa

BİLİMSEL ETİĞE UYGUNLUK.....	i
YÖNERGEYE UYGUNLUK.....	ii
TEŞEKKÜR.....	iv
ÖZET	v
ABSTRACT.....	vi
İÇİNDEKİLER	vii
KISALTMALAR	ix
TABLolar LİSTESİ.....	x
ŞEKİLLER LİSTESİ	xi
GİRİŞ	1

1. BÖLÜM

GENEL BİLGİLER.....	3
1.1. Yüksek Hızlı Ağlar İçin Tıkanıklık Kontrolü.....	3
1.1.1 Gecikme Tabanlı Tıkanıklık Kontrolü	3
1.1.2. Kayıp Tabanlı Tıkanıklık Kontrolü	4
1.2. High Speed TCP Protokolü	5
1.3. Scalable TCP Protokolü.....	5
1.4. Bic-TCP Protokolü	5
1.5. Cubic-TCP Protokolü	6
1.6. Hamilton TCP Protokolü.....	6

2. BÖLÜM

KUYRUK YÖNETİM ALGORİTMALARI.....	7
2.1. RED Algoritması	7
2.2. REM Algoritması	8
2.3. SFQ Algoritması.....	8

3. BÖLÜM

DENEYSEL ÇALIŞMALAR.....	9
3.1. Bic-TCP Protokolünün Değerlendirilmesi	9
3.2. Cubic-TCP Protokolünün Değerlendirilmesi	12
3.3. HS-TCP Protokolünün Değerlendirilmesi	16
3.4. H-TCP Protokolünün Değerlendirilmesi	19
3.5. S-TCP Protokolünün Değerlendirilmesi	22
3.6. Bic-TCP, Cubic-TCP, HS-TCP, H-TCP ve S-TCP Protokollerinin Karşılaştırılması	25

4. BÖLÜM

SONUÇLAR.....	47
KAYNAKLAR	48
ÖZGEÇMİŞ.....	53

KISALTMALAR

<u>Kısaltma</u>	<u>Açıklaması</u>
TCP	Taşıma Kontrol Protokolü
AIMD	Toplanarak Artan Çarpılarak Azalan
Cwnd	Tıkanıklı Penceresi
H-TCP	Hamilton TCP
HS-TCP	Highspeed TCP
S-TCP	Scalable TCP
RED	Random Early Detection
REM	Random Exponential Marking
SFQ	Stochastic Fairness Queuing

TABLÖLAR LİSTESİ

Tablo 3.1.	RED, REM ve SFQ için Bic-TCP protokolü kullanan ağlardaki gecikme ortalaması sonuçları	10
Tablo 3.2.	RED, REM ve SFQ için Cubic-TCP protokolü kullanan ağlardaki gecikme ortalaması sonuçları	14
Tablo 3.3.	RED, REM ve SFQ için HS-TCP protokolü kullanan ağlardaki gecikme ortalaması sonuçları	17
Tablo 3.4.	RED, REM ve SFQ için H-TCP protokolü kullanan ağlardaki gecikme ortalaması sonuçları	20
Tablo 3.5.	RED, REM ve SFQ için S-TCP protokolü kullanan ağlardaki gecikme ortalaması sonuçları	23
Tablo 3.6.	Düşük ve yüksek bant genişlikleri için ağlardaki gecikme ortalaması sonuçları.....	25
Tablo 3.7.	Düşük ve yüksek paket pencere boyutu değerleri için ağlardaki gecikme ortalaması sonuçları	32
Tablo 3.8.	Düşük ve yüksek yayılım gecikmesi için ağlardaki gecikme ortalaması sonuçları.....	46

ŞEKİLLER LİSTESİ

Şekil 1.1.	Gecikme Tabanlı Tıkanıklık Kontrolü	4
Şekil 1.2.	Doğrusal logaritmik artış grafiği	5
Şekil 3.1.	Benzetim için kullanılan ağ topolojisi	9
Şekil 3.2.	Bic-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için throughput sonuçları	10
Şekil 3.3.	Bic-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için gecikme sonuçları	11
Şekil 3.4.	Bic-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için düşen paketlerin oranı	12
Şekil 3.5.	Cubic-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için throughput sonuçları	13
Şekil 3.6.	Cubic-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için gecikme sonuçları	14
Şekil 3.7.	Cubic-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için düşen paketlerin oranı	15
Şekil 3.8.	HS-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için throughput sonuçları	16
Şekil 3.9.	HS-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için gecikme sonuçları	17
Şekil 3.10.	HS-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için düşen paketlerin oranı	18
Şekil 3.11.	H-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için throughput sonuçları	19
Şekil 3.12.	H-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için gecikme sonuçları	20
Şekil 3.13.	H-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için düşen paketlerin oranı	21
Şekil 3.14.	S-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için throughput sonuçları	22
Şekil 3.15.	S-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için gecikme sonuçları	23
Şekil 3.16.	S-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için düşen paketlerin oranı	24
Şekil 3.17.	Düşük bant genişliği için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki throughput sonuçları	26

Şekil 3.18. Yüksek bant genişliği için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki throughput sonuçları	27
Şekil 3.19. Düşük bant genişliği için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki gecikme sonuçları	28
Şekil 3.20. Yüksek bant genişliği için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki gecikme sonuçları	29
Şekil 3.21. Düşük bant genişliği için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki düşen paketlerin oranı ...	30
Şekil 3.22. Yüksek bant genişliği için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki düşen paketlerin oranı ...	31
Şekil 3.23. Düşük paket pencere boyutu için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki throughput sonuçları	33
Şekil 3.24. Yüksek paket pencere boyutu için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki throughput sonuçları	34
Şekil 3.25. Düşük paket pencere boyutu için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki gecikme sonuçları	35
Şekil 3.26. Yüksek paket pencere boyutu için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki gecikme sonuçları	36
Şekil 3.27. Düşük paket pencere boyutu için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki düşen paketlerin oranı	37
Şekil 3.28. Yüksek paket pencere boyutu için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki düşen paketlerin oranı	38
Şekil 3.29. Düşük yayılım gecikmesi topolojisi	39
Şekil 3.30. Yüksek yayılım gecikmesi topolojisi	39
Şekil 3.31. Düşük yayılım gecikmesi için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki throughput sonuçları .	40
Şekil 3.32. Yüksek yayılım gecikmesi için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki throughput sonuçları	41
Şekil 3.33. Düşük yayılım gecikmesi için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki gecikme sonuçları	42
Şekil 3.34. Yüksek yayılım gecikmesi için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki gecikme sonuçları	43

- Şekil 3.35. Düşük yayılım gecikmesi için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki düşen paketlerin oranı..... 44
- Şekil 3.36. Yüksek yayılım gecikmesi için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki düşen paketlerin oranı..... 45



GİRİŞ

Paket anahtarlamalı ağlarda tıkanıklık önemli bir kavramdır. Bilgisayar ağlarında bir yönlendiricide veya bir alt ağda paket birikmesi ve ağın performansının düşmesi tıkanıklık olarak adlandırılır. Ağa gönderilen paketlerin sayısı ağın kapasitesinden fazla olması durumunda tıkanıklık meydana gelir. TCP protokolü tıkanıklığı önlemek için Additive Increase/Multiplicative Decrease (AIMD) yöntemini kullanır. AIMD algoritması, düşük hızlı ağlar üzerinde oldukça iyi performans göstermektedir. Ancak günümüzde, büyük hacimli verilerin hızlı aktarımı için talepler artmaktadır ve bu talebi desteklemek için ağ altyapılarına ihtiyaç duyulmaktadır. AIMD algoritması büyük tıkanıklık pencerelerine yavaş yanıt vermekte ve yüksek hızlı geniş alan ağlarında düşük performans göstermektedir. Yüksek hızlı ağlar üzerinde bu TCP sınırlamasının üstesinden gelmek için, son yıllarda Bic-TCP, Cubic-TCP, Highspeed TCP (HS-TCP), Hamilton TCP (H-TCP) ve Scalable TCP (S-TCP) yüksek hızlı TCP protokolleri geliştirilmiştir. Bu protokollerin çoğu önümüzdeki yıllarda internet üzerinden standart TCP'nin yerini almak üzere tasarlanmıştır. Bu protokoller pencere büyüme politikaları açısından TCP ile farklılık göstermektedir ve farklı özelliklere sahiptir. Bu protokollerin amacı, ölçeklenebilirlik ve sürekli devam eden bant genişliği gereksinimine hizmet etmektir.

Bu tez çalışmasında NS2 benzetim programı kullanılarak, Bic-TCP, Cubic-TCP, HS-TCP, H-TCP ve S-TCP yüksek hızlı TCP tıkanıklık protokolleri değerlendirilmiştir. Yayılım gecikmesi, bant genişliği ve paket pencere boyutu değişimlerine göre yüksek hızlı TCP tıkanıklık protokolleri Random Early Detection (RED), Random Exponential Marking (REM) ve Stochastic Fairness Queueing (SFQ) kuyruk yönetim algoritmaları için karşılaştırılmıştır. Düşen paketlerin oranı, throughput ve gecikme sonuçları değerlendirilmiştir.

Tezin birinci bölümde yüksek hızlı TCP tıkanıklık protokolleri hakkında genel bilgiler verilmiştir. İkinci bölümde kuyruk yönetim algoritmaları anlatılmıştır. Üçüncü bölümde

ise NS2 ađ benzetim programını kullanarak RED, REM ve SFQ kuyruk ynetim algoritmaları iin yksek hızlı TCP tıkanıklık protokolleri deęerlendirme sonuları verilmiřtir. Son blmde ise, elde edilen sonular yorumlanarak nerilere yer verilmiřtir.



1. BÖLÜM

GENEL BİLGİLER

1.1. Yüksek Hızlı Ağlar İçin Tıkanıklık Kontrolü

Yüksek hızlı ağların birincil amacı, büyük miktarda verinin mümkün olan en kısa sürede taşınmasıdır [1-5]. Çok yüksek hızlı bağlantılarda tıkanıklığın az olması beklenir, bu nedenle hız birincil önem kazanır. Tıkanıklık protokollerinin hızlı ve tıkanıklığı önleme yeteneklerine sahip olması gerekir. Tipik bir ağda ortaya çıkabilecek tüm farklı durumları oluşturmak için ağ bağlantısı üzerindeki arka plan trafiğini kontrol etmek önemlidir [6].

Standart TCP, tıkanıklığı önleme ve mevcut düşük hızlı ağlarda ciddi sıkışıklığı önlemek için kontrolü gerçekleştirme konusunda son derece başarılı olmuştur. Bununla birlikte, standart TCP'nin yüksek hızlı ağlar için AIMD algoritması açısından uygun olmadığı bilinmektedir. Düşük performans problemini ortadan kaldırmak için, standart TCP ve AIMD algoritması, yüksek hızlı ağlarda geliştirilmiştir [7-12].

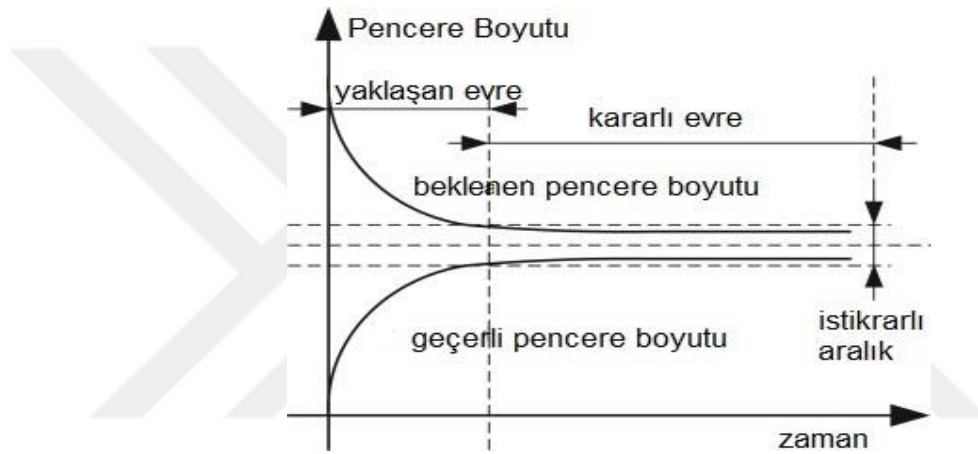
Yüksek hızlı ağlar için tıkanıklık kontrol algoritması, pencere boyutunun artırılmasının veya azaltılmasının gerekip gerekmediğini belirlemek için paket kayıp bilgilerini kullanır ve artırma veya azaltma miktarını belirlemek için kuyruklama gecikmesi bilgilerini kullanır [13]. Yüksek hızlı ve uzun mesafe ağ ortamına adaptasyon için gecikme tabanlı ve kayıp tabanlı tıkanıklık kontrol algoritmaları geliştirilmiştir [14-16].

1.1.1 Gecikme Tabanlı Tıkanıklık Kontrolü

Xu ve ark. 2010 yılında yüksek hızlı ağlarda ters trafik etkisini ortadan kaldırmak için Eefast tıkanıklık kontrol algoritmasını geliştirdi [17]. Gecikme tabanlı tıkanıklık kontrolünün tahmini bileşenlerinin tasarımı ve bir ağdaki tıkanıklığı hesaplamak için Eefast algoritmasının mekanizmaları kullanılmıştır. Bu algoritmaya dayanarak daha

fazla bir performans iyileştirmesi sağlamak ve pencere boyutunu ayarlamak için yeni kontrol stratejileri seti önerilmiştir [18].

Gecikme tabanlı tıkanıklık kontrol algoritmaları, bant genişliğinin tamamının kullanımı ve ortalama throughputu korumak için yönlendiricilerde sıralanan belirli sayıda paket gerektirir. Bu nedenle, yönlendiricilerin arabellek boyutu, gecikme tabanlı algoritmalarda belirtilen değerden daha büyük olmalıdır [12]. Bir ağ için belirtilen değer, kaynak sayılarının artışı kadar artmaktadır. Şekil 1.1'de Gecikme tabanlı tıkanıklık kontrolüne örnek gösterilmiştir [18].



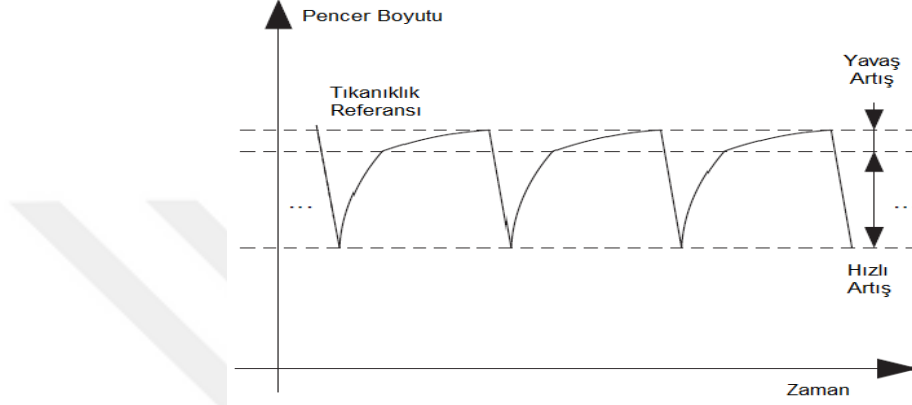
Şekil 1.1. Gecikme Tabanlı Tıkanıklık Kontrolü[18]

1.1.2. Kayıp Tabanlı Tıkanıklık Kontrolü

Yönlendiricilerin arabellek boyutu belirtilen değer için yeteri kadar büyük değilse, ağlarda paket kaybı olabilir. Bu problemin üstesinden gelmek için tıkanıklık kontrolü olarak paket kaybı kullanılır. Bunun için kayıp tabanlı tıkanıklık kontrolü geliştirilmiştir [19].

Ağ tıkanıklık durumuna yakın olduğu durumlarda pencere boyutunun hızlı bir şekilde artması tıkanıklık olayını daha kolay hale getirebilir. Kayıp tabanlı tıkanıklık kontrolü pencere boyutunda ağır salınımına neden olabilir, böylece her trafik kaynağı için throughput performansı azalır. Lineer ile logaritmik artış fonksiyonu, pencere boyutunun hızla artması nedeniyle oluşan yoğun tıkanıklığı önlemenin etkili bir yoludur. Başlangıç aşamasında pencere boyutu doğrusal olarak artar ve daha sonra bir tıkanıklık olayının meydana gelebileceği referans noktasına yaklaşmak için logaritmik olarak artış gösterir. Temel olarak, pencere boyutunun değişimi hızlıdan yavaş doğru

gider. Bu nedenle, bu gibi mekanizmaları kullanarak, trafik kaynağı mevcut bant genişliğini hızla yakalayabilir ve iki ardışık tıkanıklık olayı arasındaki zaman aralığını uzatabilir. Böylece daha iyi bir performans elde edilir. Şekil 1.2’de doğrusal logaritmik artış grafiği gösterilmiştir [20].



Şekil 1.2. Doğrusal logaritmik artış grafiği[20]

1.2. High Speed TCP Protokolü

High Speed TCP (HS-TCP), 2003 yılında Sally Floyd tarafından geliştirilen yüksek hızlı bir TCP tıkanıklık protokolüdür. Önerilen ilk büyük hızlı TCP protokollerinden biridir. Tıkanıklık penceresi minimum threshold (düşük pencere) değerinden düşükse, HS-TCP, standart TCP ile aynı davranır [20]. Tıkanıklık penceresi düşük pencere değerinin üzerinde olduğunda, HS-TCP, tıkanıklık penceresi boyutunu agresif bir şekilde artırarak bant genişliği elde etmede performans gösterir.

1.3. Scalable TCP Protokolü

Tom Kelly 2003 yılında HS-TCP protokolünü iyileştirerek Scalable TCP (S-TCP) protokolünü önermiştir. S-TCP protokolü tıkanıklık penceresini doğrusal olarak artırır ve tıkanıklık penceresini çarpımsal olarak azaltır.

1.4. Bic-TCP Protokolü

Bic-TCP protokolü, 2004 yılında Xu ve ark. tarafından önerilmiştir. Bu protokol, diğer AIMD protokollerine benzer özelliklere sahiptir, ancak bağlantı kapasitesine yaklaştığında davranışı farklılık gösterir [22]. Bic-TCP protokolü mevcut ağ bant genişliğini tahmin eder ve maksimum pencere olarak adlandırılan hedef pencere

boyutunu ayarlar. Minimum pencere adı verilen bir deęişkende mevcut tıkanıklık penceresinin boyutunu korur. Bic-TCP protokolü, yeni bir onay (ACK) alındığında minimum pencereyi minimum pencere ve maksimum pencere orta noktasına artırmak için ikili bir arama fonksiyonu kullanır. Başlangıçta minimum pencere ve maksimum pencere arasındaki fark büyük olduğunda, Bic-TCP protokolü katkı artışını kullanır. Fark eşik deęerinin altındaysa, Bic-TCP tıkanıklık penceresinin boyutunu ikili aramayla artırır [23].

1.5. Cubic-TCP Protokolü

Rhee ve ark. Bic-TCP protokolünü geliştirerek 2008 yılında Cubic-TCP protokolünü önermişlerdir. Bic-TCP protokolünün adaletli davranışı daha az girişken hale getirilerek iyileşme sağlanmıştır. Bu protokolde tıkanıklık penceresinin deęeri, önceki tıkanıklık penceresi deęerinden ziyade son tıkanıklık olayına baęlı hale getirilmiştir [24]. TCP uyumluluęuna ulaşma yaklaşımı da tamamen deęiştirilmiştir. Bu protokolde tıkanıklık penceresi deęeri için minimum eşik deęeri yoktur. Bunun yerine, benzer bir durumdaki bir TCP baęlantısının pencere boyutu bulunur ve Cubic-TCP'nin tıkanıklık penceresi deęerinden daha fazla olması durumunda bu deęer kullanılır [25].

1.6. Hamilton TCP Protokolü

Hamilton TCP (H-TCP) protokolü, 2004 yılında İrlanda Hamilton Enstitüsü'nde önerilmiştir. Yüksek ve düşük hızlı iletişim ortamlarında etkin bir şekilde çalışmak üzere tasarlanmıştır. Tıkanıklık penceresinin modellenmesi dięer tıkanıklık kontrol algoritmalarının çoęundan farklıdır. Bu protokol, tıkanıklık penceresinin artışından ziyade tıkanıklık olayları arasındaki zaman farklarındaki fonksiyonları artırmakta ve düşürmektedir [26-29]. H-TCP protokolü, tıkanıklık penceresini ayarlamak için p düşüşleri arasındaki süreyi kullanır. H-TCP protokolü tıkanıklığı tespit ettiğinde tıkanıklığı azaltmak için uyarlanabilir bir backoff stratejisi kullanır [30].

2. BÖLÜM

KUYRUK YÖNETİM ALGORİTMALARI

Kuyruk yönetimi, tıkanıklık oranını en aza indirmek için kullanılan bir yöntemdir. Bu yöntem sayesinde paketler göndericiden alıcıya başarılı bir şekilde iletilir [31]. Bir yönlendiriciye uygulanan kuyruk yönetim algoritması, hizmet kalitesinin (QoS) elde edilmesinde önemli bir rol oynar [32]. Kuyruk yönetim algoritmaları pasif, aktif ve proaktif olmak üzere 3 sınıfa ayrılmaktadır. Aktif kuyruk yönetimi, ağdaki tıkanıklığın erken tespiti için bir yönlendirici tabanlı mekanizma olarak önerilmiştir [33]. Pasif kuyruk yönetimi ise tıkanıklık meydana geldikten sonra tıkanıklığı önlemek için kullanılan yöntemdir [34-35]. Bu tezde Aktif kuyruk yönetimi algoritmalarından RED ve REM algoritmaları ve Pasif kuyruk yönetimi algoritmalarından SFQ algoritması üzerine çalışılmıştır.

2.1. RED Algoritması

RED (Random Early Detection) algoritması öncelikle paket kaybını ve kuyruklama gecikmesini azaltmak amacıyla önerilmiştir. Daha sonra, kaynakların küresel senkronizasyonunu önlemek ve yüksek bağlantı kullanımını sürdürmek için geliştirilmiştir. RED algoritması ağ tıkanıklığını üstel ağırlıklı ortalama kullanarak kuyruk uzunluğuna göre ölçmektedir [36]. Tıkanıklık önleminin seçimi tıkanıklığı yansıtacak şekilde nasıl güncellendiğini belirlemektedir. Dolayısıyla TCP tarafından kapalı olarak optimize edilen kullanıcı fonksiyonunu etkilenmektedir [37]. RED algoritması, TCP ile etkileşime girerek kaynak paket gönderim oranları arttıkça kuyruk uzunluğunu artırır. Daha fazla paket işaretlendiğinde kaynaklara paket gönderim oranlarını düşürmesi istenir ve döngü tekrarlanır. RED algoritması sayesinde erken tıkanıklık tespiti gerçekleşir. Fakat RED algoritması yoğun (bursty) trafığe duyarsız [38].

2.2. REM Algoritması

REM algoritması, tıkanıklık bilgilerini iletmek için geliştirilen aktif kuyruk yönetim algoritmasıdır. Bu algoritma hem yüksek kullanımı hem de ihmal edilebilir kayıp ve gecikmeyi basit ve ölçeklenebilir bir tarzda gerçekleştirmeyi amaçlamaktadır. REM algoritması, kullanıcı sayısına bakmadan arabellekleri temizlerken kullanıcı oranlarını ağ kapasitesiyle eşleştirmeye çalışır. Aynı zamanda bu algoritma, kullanıcı yolundaki tüm yönlendiricilere uçtan uca işaretleme (veya bırakma) ihtimalini toplamak suretiyle, tıkanıklık ölçümünü gerçekleştirir. Etkin akışların sayısı, arabellekdeki bir dizi farklı akışla lineer bir ilişki paylaşmaktadır. REM algoritması sayesinde ağda düşük hesaplama yükü, yüksek bağlantı kullanımı, ihmal edilebilir kayıp ve düşük gecikme sağlanır. REM algoritmasının dezavantajı ise, web trafiği için düşük throughput elde etmesidir [39-42].

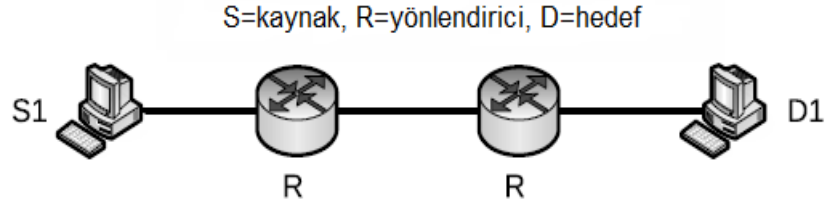
2.3. SFQ Algoritması

SFQ, stokastik algoritma olarak adlandırılan pasif kuyruk yönetim algoritmasıdır. Bu algoritma, her oturum için gerçekten bir sıra ayırmaz. Bir hash algoritma kullanarak sınırlı sayıda kuyruk üzerinde trafiği ayıran bir yöntem kullanır [43]. SFQ algoritması, adil kuyruklama algoritması prensibine dayanır. Bu durum çok adil davranışlara yol açar [44]. SFQ algoritması sayesinde kullanılabilir bant genişliğinin tüm alt-akış arasında adil dağılımı gerçekleşir. Bu algoritma, farklı kaynaktaki bağlantı kapasitesinin kullanımının eşit olduğu ağlar için kullanışlıdır ve uçtan-uca düşük bir gecikmeye sahiptir. Bu nedenle, kuyruk mekanizmaları gecikmeye duyarlı uygulamada kullanılabilir. SFQ algoritmasının dezavantajı ise, paketin kayıp oranının yüksek olması ve tıkanıklık penceresi dalgalanmasının daha fazla olmasıdır [45-47].

3. BÖLÜM

DENEYSEL ÇALIŞMALAR

Bu tez çalışmasında NS2 benzetim programı kullanılarak Bic-TCP, Cubic-TCP, H-TCP, HS-TCP ve S-TCP yüksek hızlı tıkanıklık kontrol protokolleri karşılaştırılmıştır. Kablolulu bir ağda aktif kuyruk yönetim algoritmalarından RED, REM ve pasif kuyruk yönetim algoritmalarından SFQ için tıkanıklık kontrol protokolleri değerlendirilmiştir. Tasarım bir gönderici, bir alıcı ve iki adet yönlendiriciden oluşturulmuştur. Şekil 3.1’de oluşturulan ağ topolojisi gösterilmiştir. Topolojide kullanılan bütün bant genişlikleri 2 Mbps olarak belirlenmiştir. Yönlendiricilerin kuyruk boyutu 10 paket olarak seçilmiştir. Tıkanıklık penceresi (cwnd) 1500 kbps olarak belirlenmiştir. Toplam benzetim süresi 150 sn olacak şekilde sonuçlar elde edilmiştir.



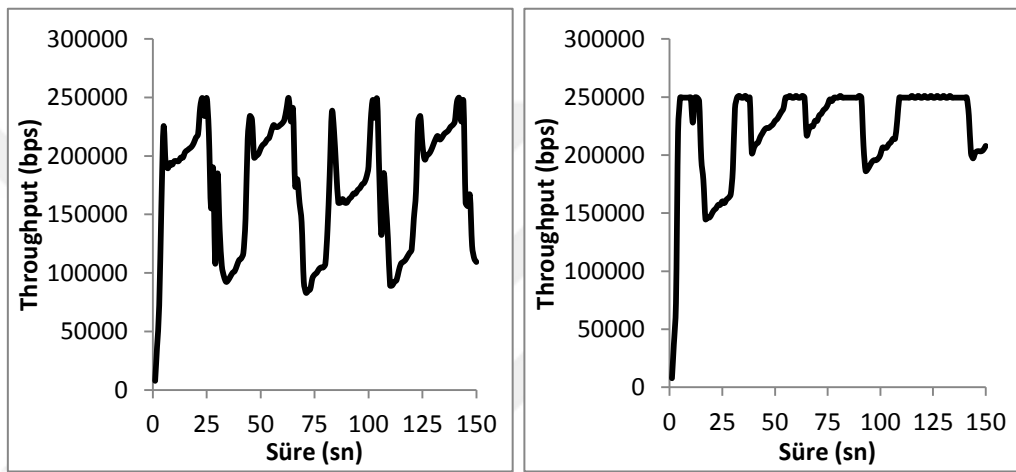
Şekil 3.1. Benzetim için kullanılan ağ topolojisi.

3.1. Bic-TCP Protokolünün Değerlendirilmesi

Bic-TCP protokolü kullanılan ağda RED, REM ve SFQ için throughput sonuçları Şekil 3.2’de gösterilmiştir. SFQ kuyruk yönetimi algoritmasının en iyi throughput değerine sahip olduğu şekilden anlaşılmaktadır. Bic-TCP protokolü kullanan ağda farklı kuyruk yönetim algoritmaları için gecikme sonuçları Şekil 3.3’de ve gecikme ortalamaları Tablo 3.1’de gösterilmiştir. Bic-TCP protokolü için RED kuyruk yönetim algoritması ile en düşük gecikme sonuçları elde edilmiştir.

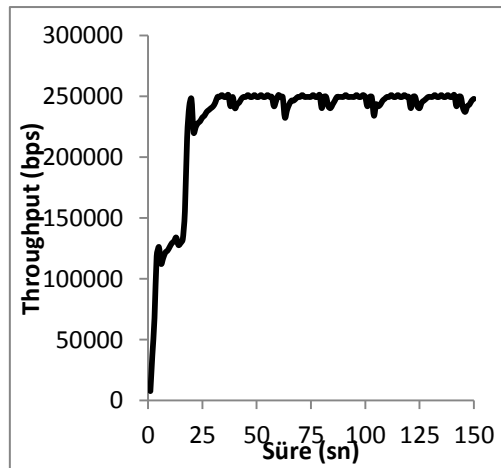
Tablo 3.1. RED, REM ve SFQ için Bic-TCP protokolü kullanan ağlardaki gecikme ortalaması sonuçları

Kuyruk Yönetim Algoritmaları	Gecikme Ortalaması (ms)
RED	276.801
REM	348.507
SFQ	285.399



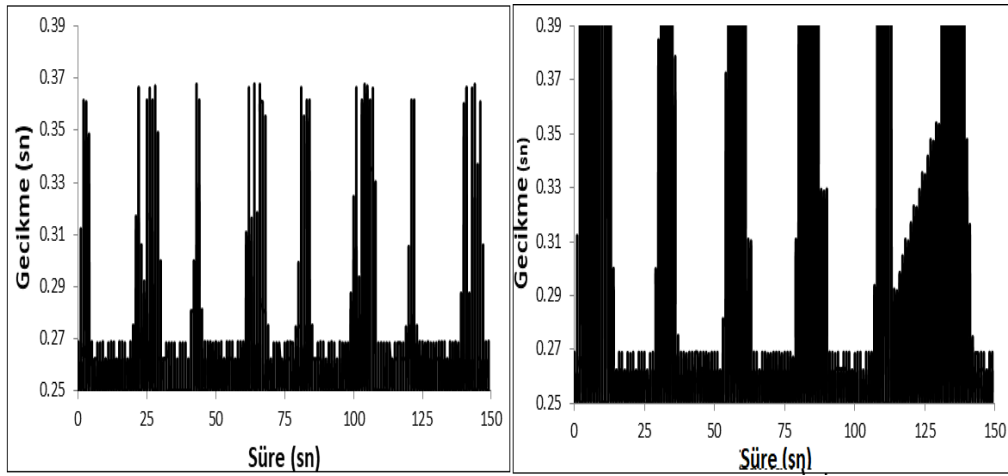
(a)

(b)



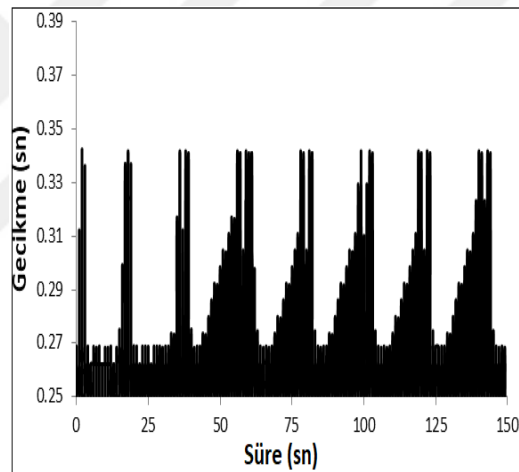
(c)

Şekil 3.2. Bic-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için throughput sonuçları



(a)

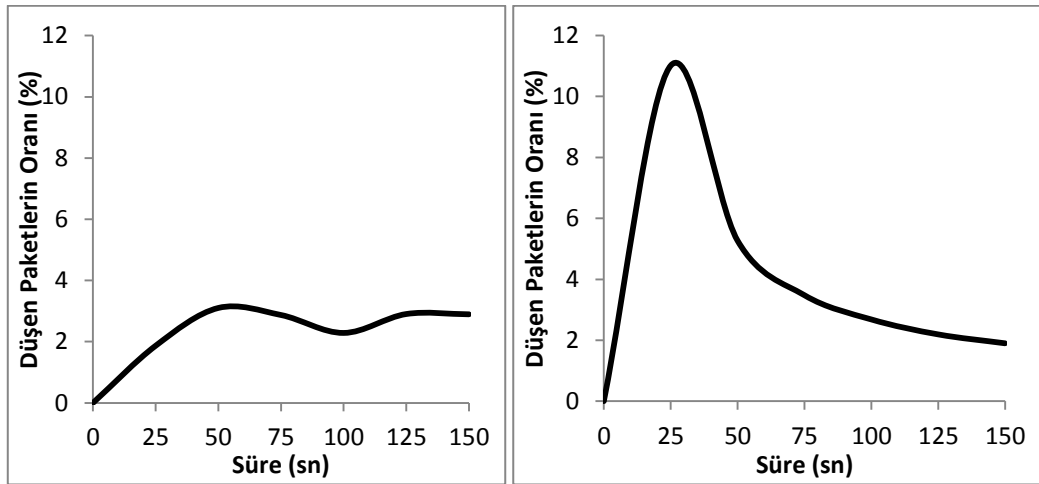
(b)



(c)

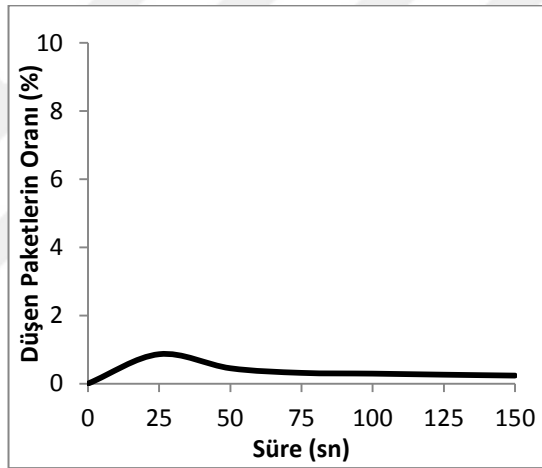
Şekil 3.3. Bic-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için gecikme sonuçları

Bic-TCP protokolü kullanılan ağda farklı kuyruk yönetim algoritmaları için düşen paketlerin oranını Şekil 3.4’de gösterilmiştir. REM kuyruk yönetim algoritmasının en çok düşen paket değerine sahip olduğu şekilden görülmektedir. SFQ algoritmasının ise en az düşen paket değerine sahip olduğu anlaşılmaktadır.



(a)

(b)

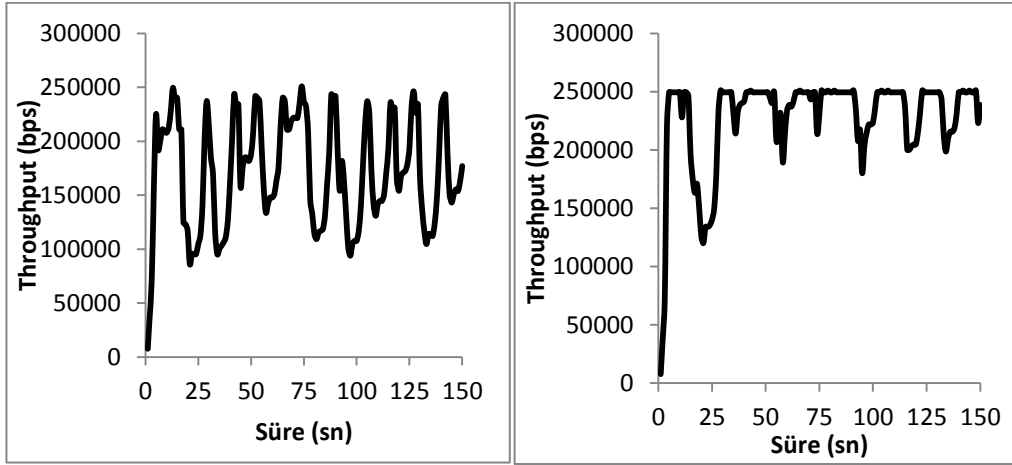


(c)

Şekil 3.4. Bic-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için düşen paketlerin oranı

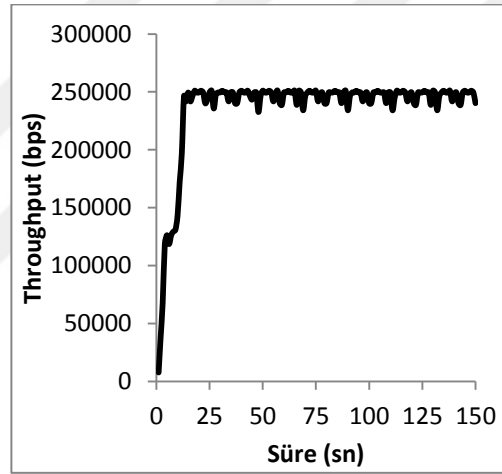
3.2. Cubic-TCP Protokolünün Değerlendirilmesi

Cubic-TCP protokolü kullanılan ağda RED, REM ve SFQ için throughput sonuçları Şekil 3.5'de gösterilmiştir. SFQ kuyruk yönetimi algoritmasının en iyi throughput değerine sahip olduğu şekilden anlaşılmaktadır. Bic-TCP protokolü kullanan ağda farklı kuyruk yönetim algoritmaları için gecikme sonuçları Şekil 3.6'da ve gecikme ortalamaları Tablo 3.2'de gösterilmiştir. Bic-TCP protokolü için RED kuyruk yönetim algoritması ile en düşük gecikme sonuçları elde edilmiştir.



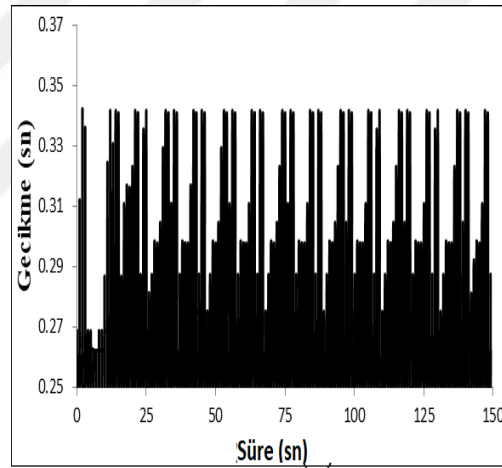
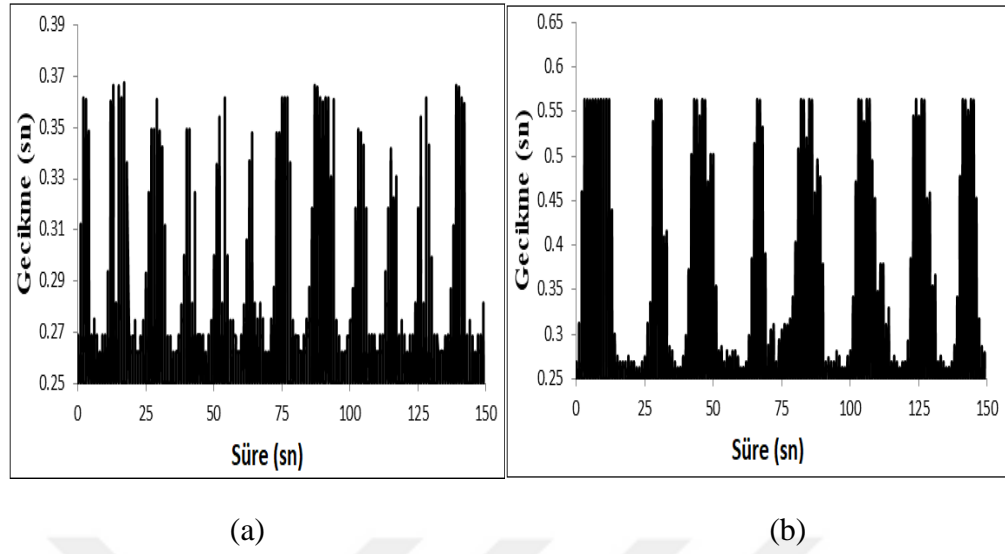
(a)

(b)



(c)

Şekil 3.5. Cubic-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için throughput sonuçları



(c)

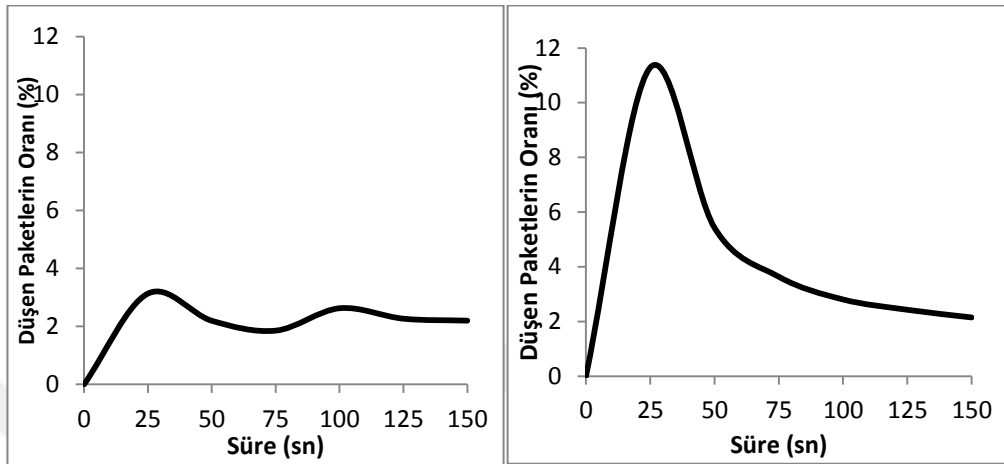
Şekil 3.6. Cubic-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için gecikme sonuçları

Tablo 3.2. RED, REM ve SFQ için Cubic-TCP protokolü kullanan ağlardaki gecikme ortalaması sonuçları

Kuyruk Yönetim Algoritmaları	Gecikme Ortalaması (ms)
RED	281.55
REM	375.504
SFQ	298.916

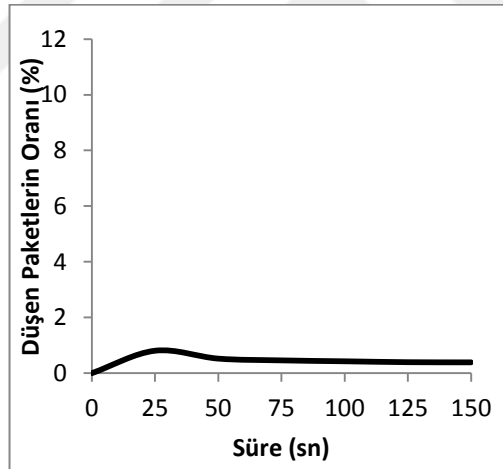
Cubic-TCP protokolü kullanılan ağda farklı kuyruk yönetim algoritmaları için düşen paketlerin oranını Şekil 3.7’de gösterilmiştir. REM kuyruk yönetim algoritmasının en

çok düşük paket değerine sahip olduğu şekilden görülmektedir. SFQ algoritmasının ise en az düşük paket değerine sahip olduğu anlaşılmaktadır.



(a)

(b)

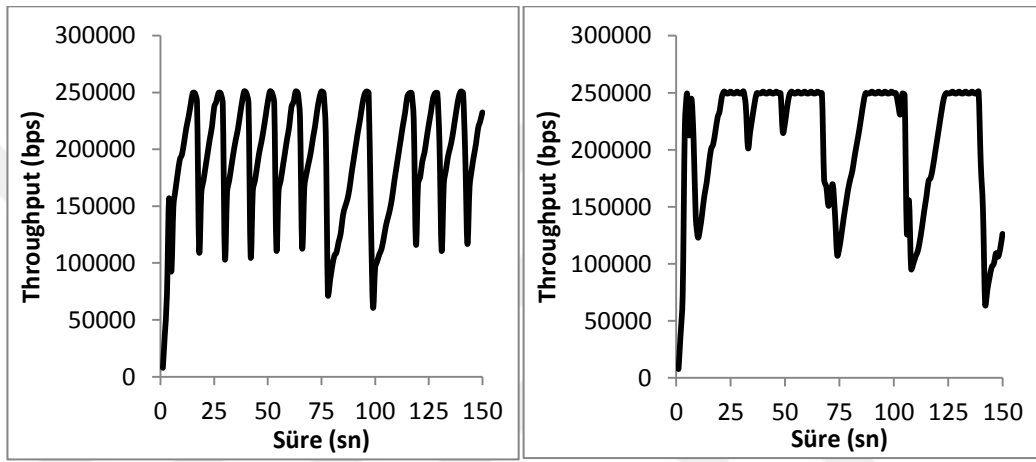


(c)

Şekil 3.7. Cubic-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için düşen paketlerin oranı

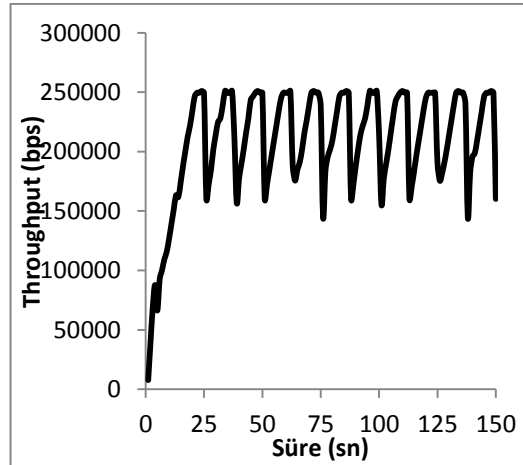
3.3. HS-TCP Protokolünün Değerlendirilmesi

HS-TCP protokolü kullanılan ağda RED, REM ve SFQ için throughput sonuçları Şekil 3.8’de gösterilmiştir. SFQ kuyruk yönetimi algoritmasının en iyi throughput değerine sahip olduğu şekilden anlaşılmaktadır. HS-TCP protokolü kullanan ağda farklı kuyruk yönetim algoritmaları için gecikme sonuçları Şekil 3.9’da ve gecikme ortalamaları Tablo 3.3’de gösterilmiştir. HS-TCP protokolü için RED kuyruk yönetim algoritması ile en düşük gecikme sonuçları elde edilmiştir.



(a)

(b)

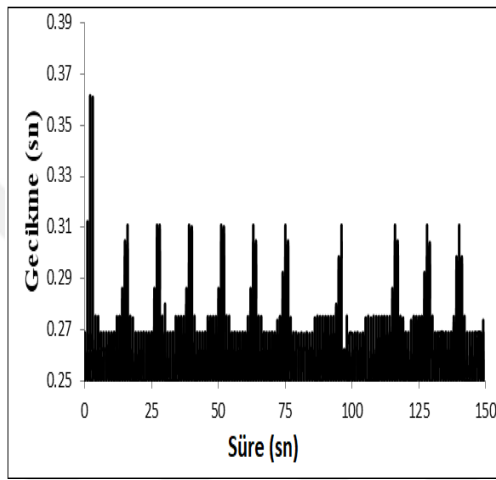


(c)

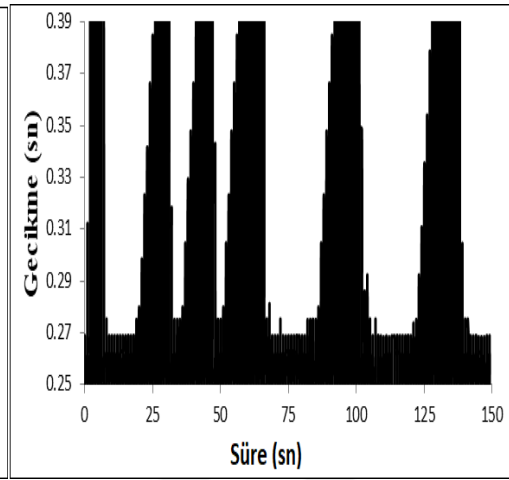
Şekil 3.8. HS-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için throughput sonuçları

Tablo 3.3. RED, REM ve SFQ için HS-TCP protokolü kullanan ağlardaki gecikme ortalaması sonuçları

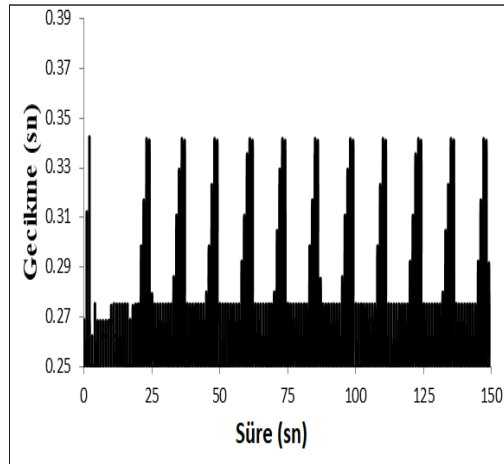
Kuyruk Yönetim Algoritmaları	Gecikme Ortalaması (ms)
RED	271.199
REM	350.804
SFQ	283.461



(a)



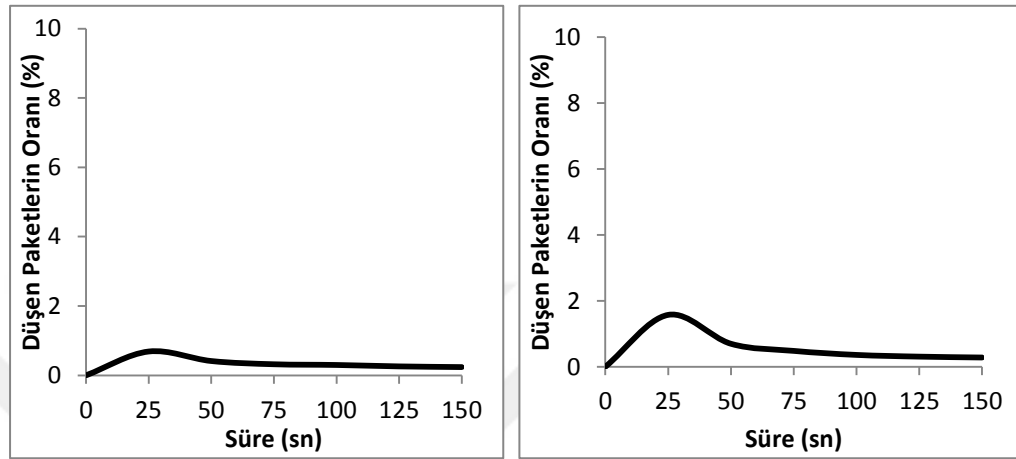
(b)



(c)

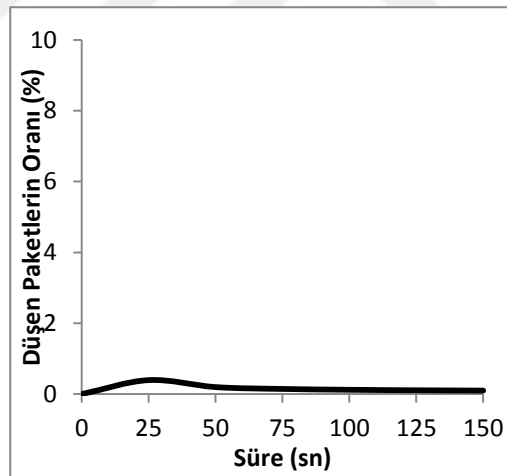
Şekil 3.9. HS-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için gecikme sonuçları

HS-TCP protokolü kullanılan ağda farklı kuyruk yönetim algoritmaları için düşen paketlerin oranını Şekil 3.10'da gösterilmiştir. REM kuyruk yönetim algoritmasının en çok düşen paket değerine sahip olduğu şekilden görülmektedir. SFQ algoritmasının ise en az düşen paket değerine sahip olduğu anlaşılmaktadır.



(a)

(b)

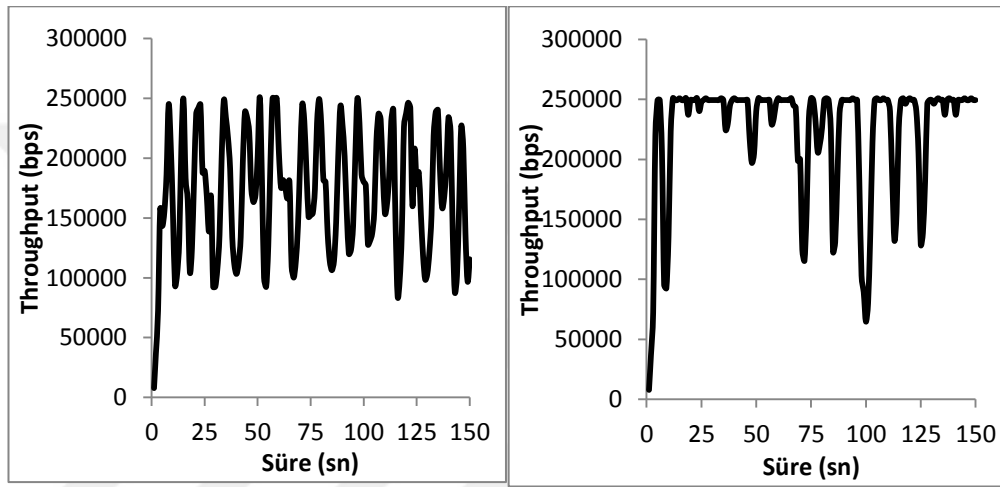


(c)

Şekil 3.10. HS-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için düşen paketlerin oranı

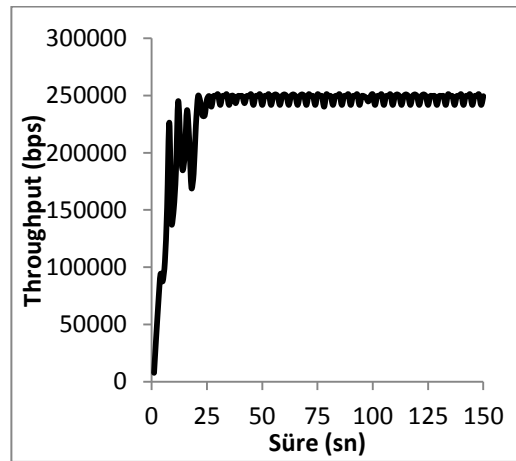
3.4. H-TCP Protokolünün Değerlendirilmesi

H-TCP protokolü kullanılan ağda RED, REM ve SFQ için throughput sonuçları Şekil 3.11’de gösterilmiştir. SFQ kuyruk yönetimi algoritmasının en iyi throughput değerine sahip olduğu şekilden anlaşılmaktadır. H-TCP protokolü kullanan ağda farklı kuyruk yönetim algoritmaları için gecikme sonuçları Şekil 3.12’de ve gecikme ortalamaları Tablo 3.4’de gösterilmiştir. HS-TCP protokolü için RED kuyruk yönetim algoritması ile en düşük gecikme sonuçları elde edilmiştir.



(a)

(b)

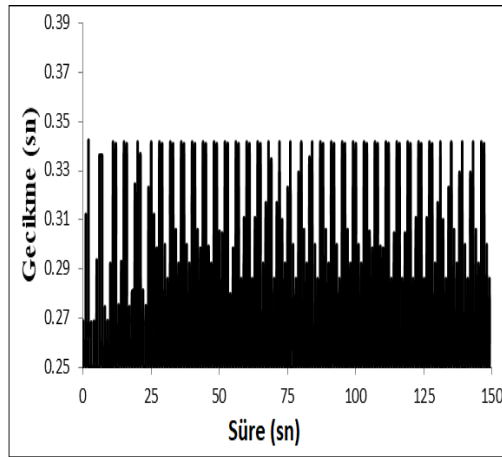
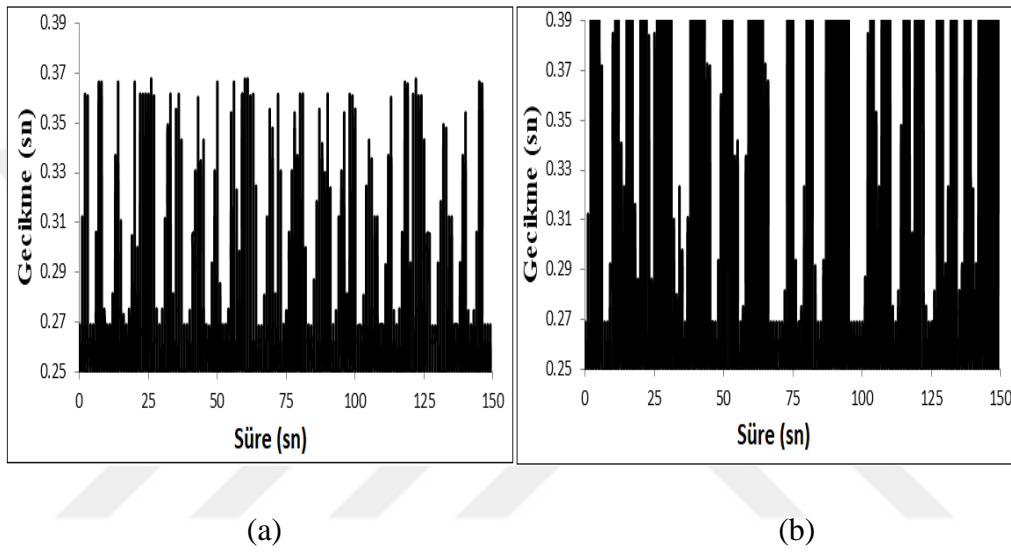


(c)

Şekil 3.11. H-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için throughput sonuçları

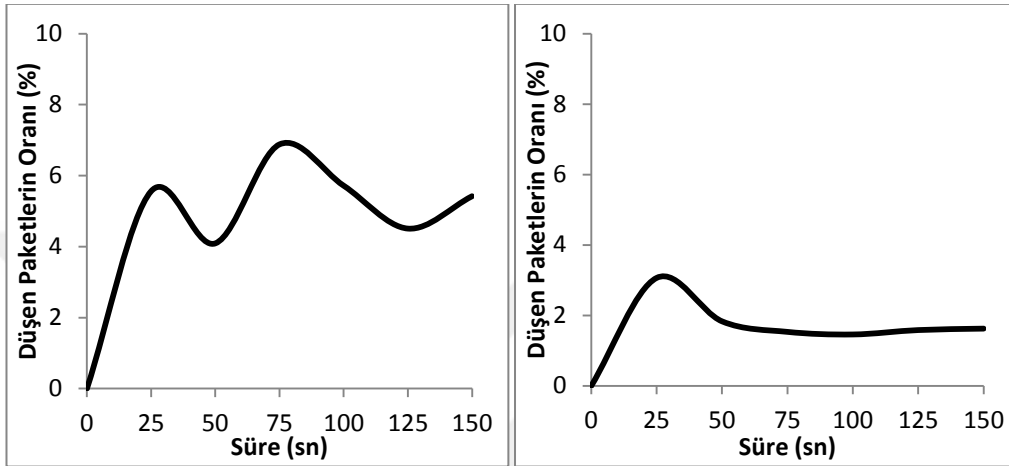
Tablo 3.4. RED, REM ve SFQ için H-TCP protokolü kullanan ağlardaki gecikme ortalaması sonuçları

Kuyruk Yönetim Algoritmaları	Gecikme Ortalaması (ms)
RED	288.834
REM	388.182
SFQ	297.792



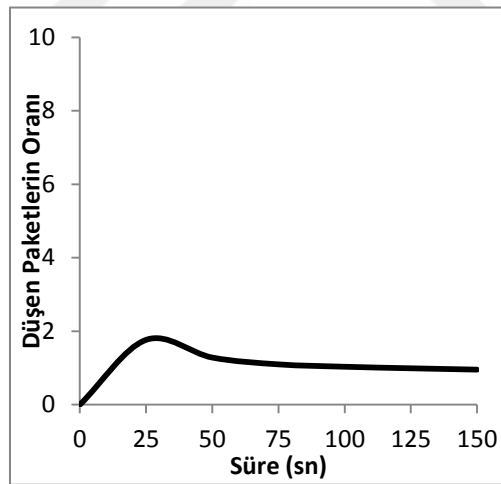
Şekil 3.12. H-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için gecikme sonuçları

H-TCP protokolü kullanılan ağda farklı kuyruk yönetim algoritmaları için düşen paketlerin oranını Şekil 3.13'de gösterilmiştir. RED kuyruk yönetim algoritmasının en çok düşen paket değerine sahip olduğu şekilde görülmektedir. SFQ algoritmasının ise en az düşen paket değerine sahip olduğu anlaşılmaktadır.



(a)

(b)

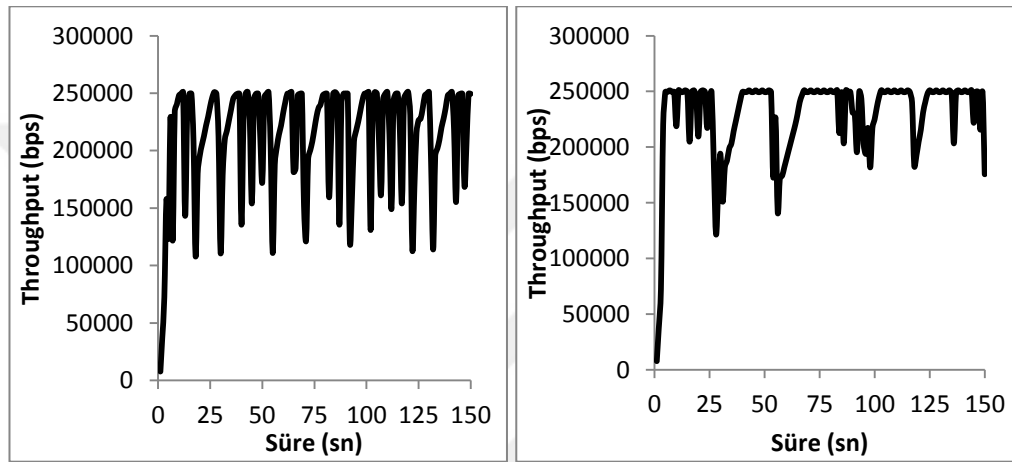


(c)

Şekil 3.13. H-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için düşen paketlerin oranı

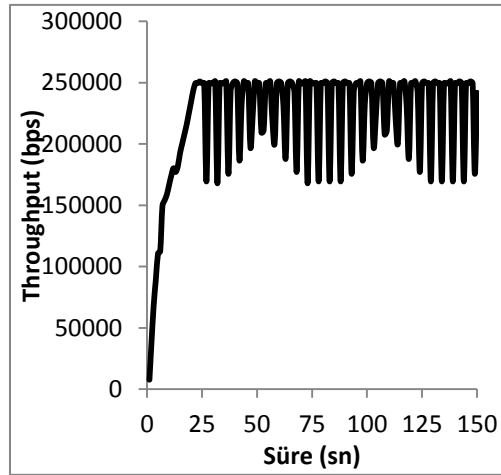
3.5. S-TCP Protokolünün Değerlendirilmesi

S-TCP protokolü kullanılan ağda RED, REM ve SFQ için throughput sonuçları Şekil 3.14’de gösterilmiştir. SFQ kuyruk yönetimi algoritmasının en iyi throughput değerine sahip olduğu şekilden anlaşılmaktadır. S-TCP protokolü kullanan ağda farklı kuyruk yönetim algoritmaları için gecikme sonuçları Şekil 3.15’de ve gecikme ortalamaları Tablo 3.5’de gösterilmiştir. HS-TCP protokolü için RED kuyruk yönetim algoritması ile en düşük gecikme sonuçları elde edilmiştir.



(a)

(b)

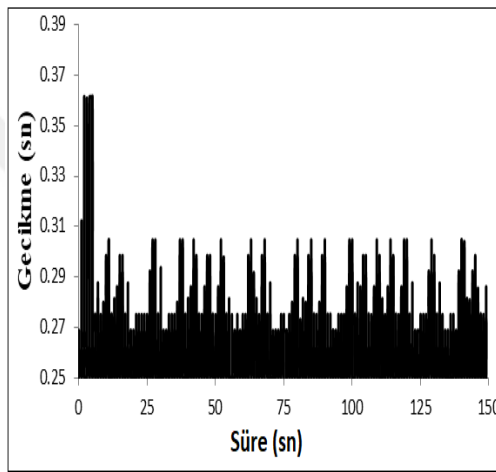


(c)

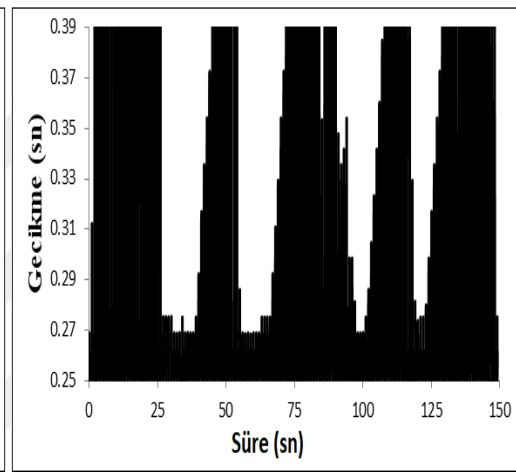
Şekil 3.14. S-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için throughput sonuçları

Tablo 3.5. RED, REM ve SFQ için S-TCP protokolü kullanan ağlardaki gecikme ortalaması sonuçları

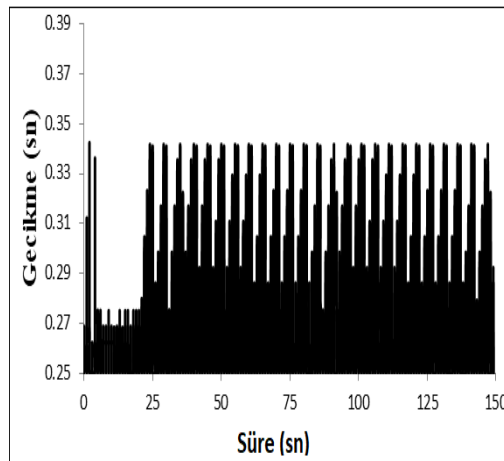
Kuyruk Yönetim Algoritmaları	Gecikme Ortalaması (ms)
RED	275.08
REM	374.505
SFQ	304.312



(a)



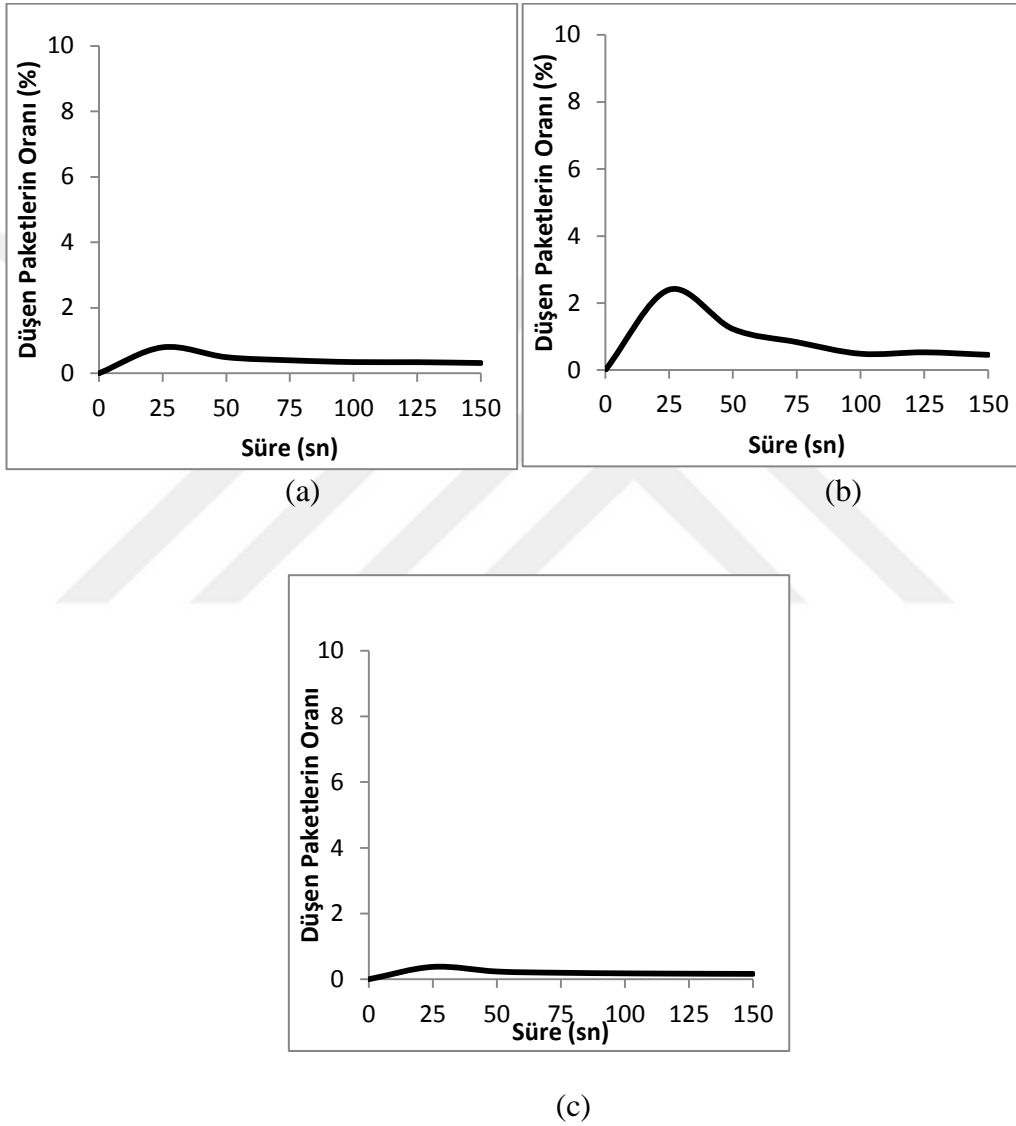
(b)



(c)

Şekil 3.15. S-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için gecikme sonuçları

S-TCP protokolü kullanılan ağda farklı kuyruk yönetim algoritmaları için düşen paketlerin oranını Şekil 3.16'da gösterilmiştir. REM kuyruk yönetim algoritmasının en çok düşen paket değerine sahip olduğu şekilden görülmektedir. SFQ algoritmasının ise en az düşen paket değerine sahip olduğu anlaşılmaktadır.



Şekil 3.16. S-TCP protokolü kullanılan ağlardaki (a) RED, (b) REM, (c) SFQ için düşen paketlerin oranı

3.6. Bic-TCP, Cubic-TCP, HS-TCP, H-TCP ve S-TCP Protokollerinin Karşılaştırılması

RED kuyruk yönetim algoritması kullanan Bic-TCP, Cubic-TCP, HS-TCP, H-TCP ve S-TCP tıkanıklık kontrol protokolleri düşük ve yüksek bant genişliği değişimine göre karşılaştırılmıştır. Topolojide kullanılan bütün bant genişlikleri 1 GB (düşük) ve 10 GB (yüksek) olarak belirlenmiş ve sonuçlar elde edilmiştir.

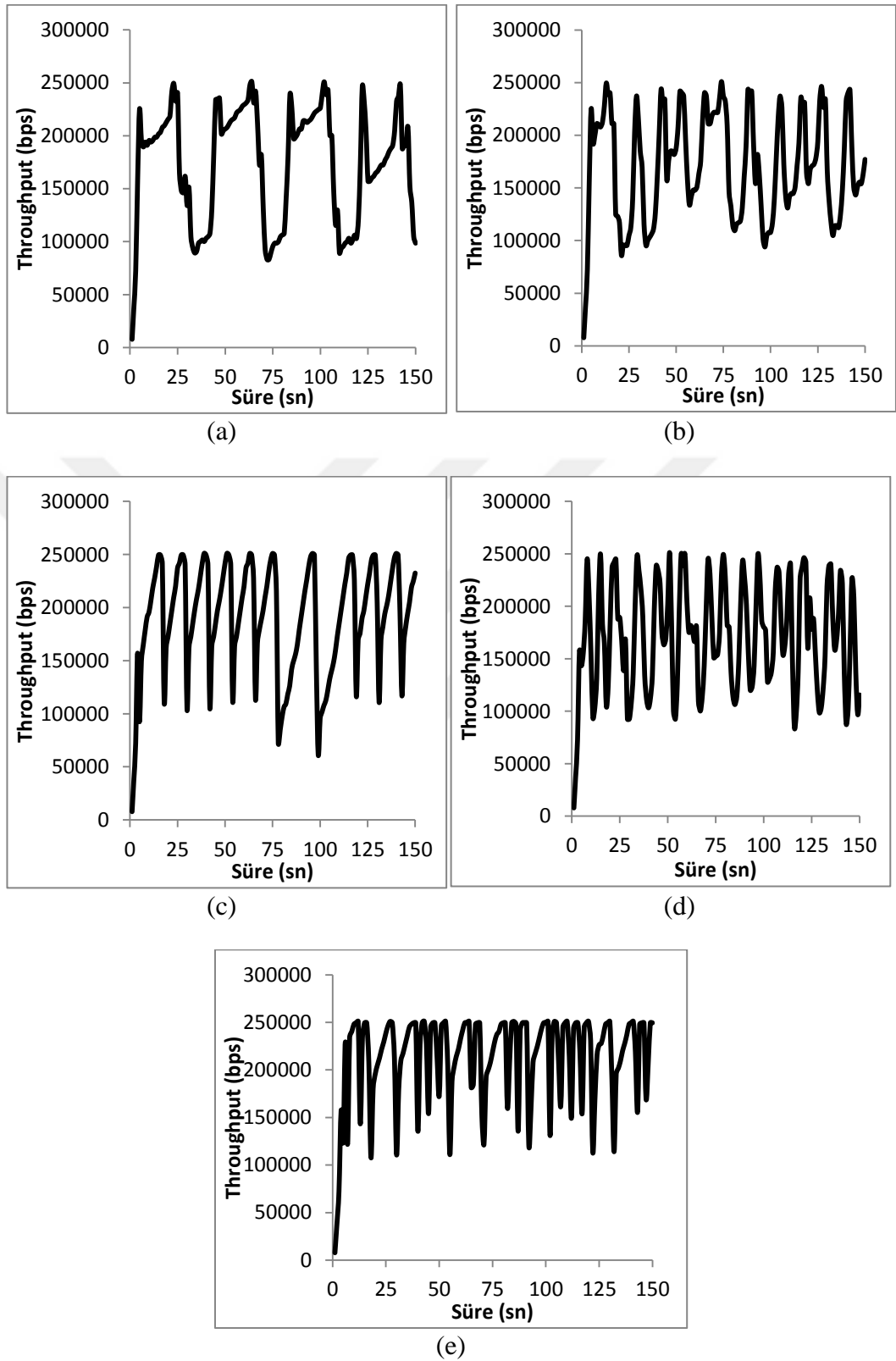
Düşük ve yüksek bant genişlikleri göre Bic-TCP, Cubic-TCP, HS-TCP, H-TCP ve S-TCP tıkanıklık kontrol protokolleri kullanılan ağlardaki throughput sonuçları Şekil 3.17 ve Şekil 3.18'de gösterilmiştir. Bic-TCP protokolünün düşük ve yüksek bant genişlikleri için en iyi başarımı elde ettiği görülmüştür.

TCP tıkanıklık kontrol protokollerinin düşük ve yüksek bant genişlikleri göre gecikme sonuçları Şekil 3.19 ve Şekil 3.20'de ve gecikme ortalamaları Tablo 3.6'da gösterilmiştir. Şekiller ve tablo incelendiğinde HS-TCP protokolünün düşük ve yüksek bant genişlikleri için en düşük gecikme değerleri elde ettiği görülmüştür.

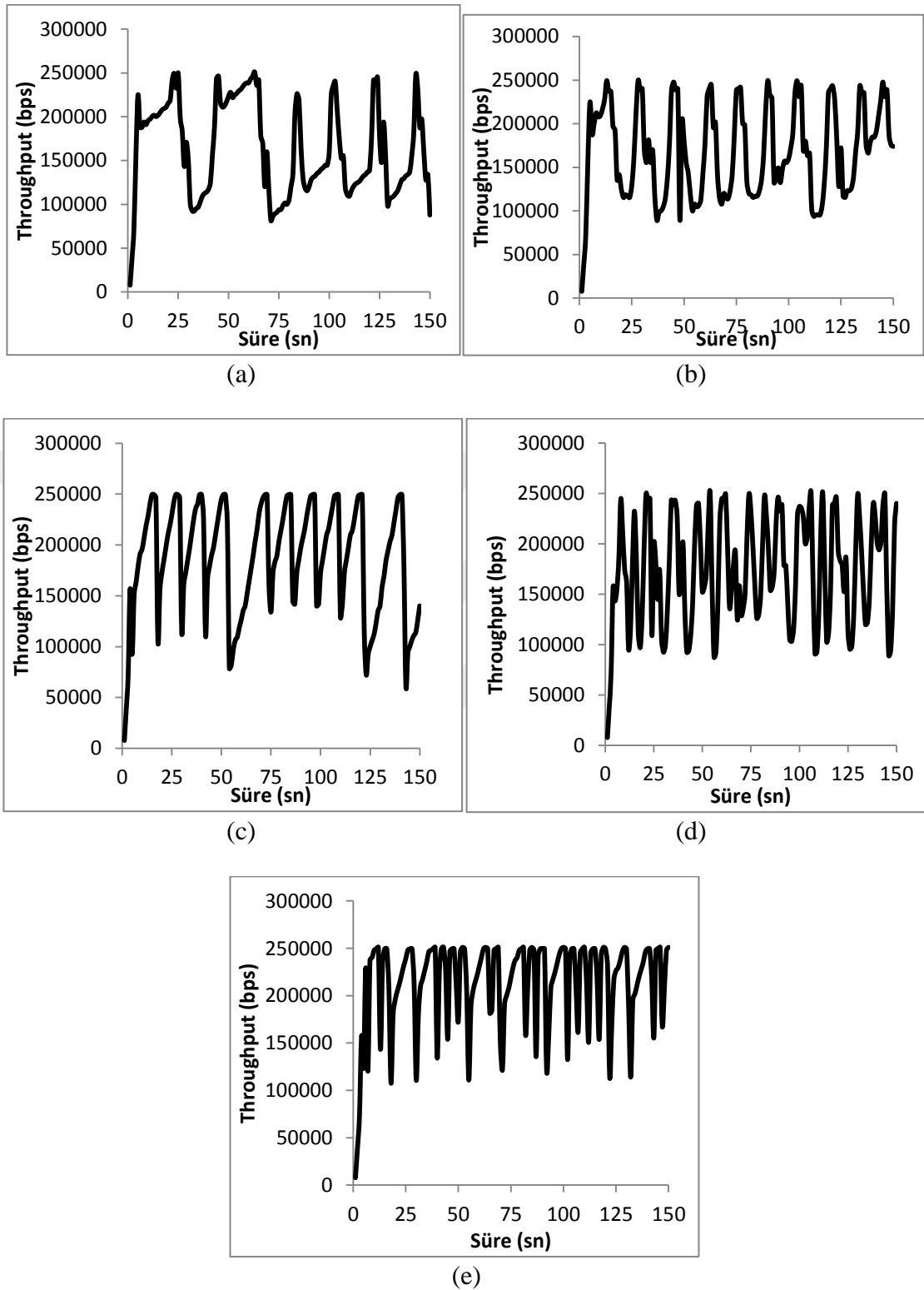
Şekil 3.21 ve Şekil 3.22'de ise TCP tıkanıklık kontrol protokollerinin düşük ve yüksek bant genişlikleri göre düşen paket oranları verilmiştir. S-TCP protokolünün düşük ve yüksek bant genişlikleri için en iyi başarımları elde ettiği görülmüştür.

Tablo 3.6. Düşük ve yüksek bant genişlikleri için ağlardaki gecikme ortalaması sonuçları

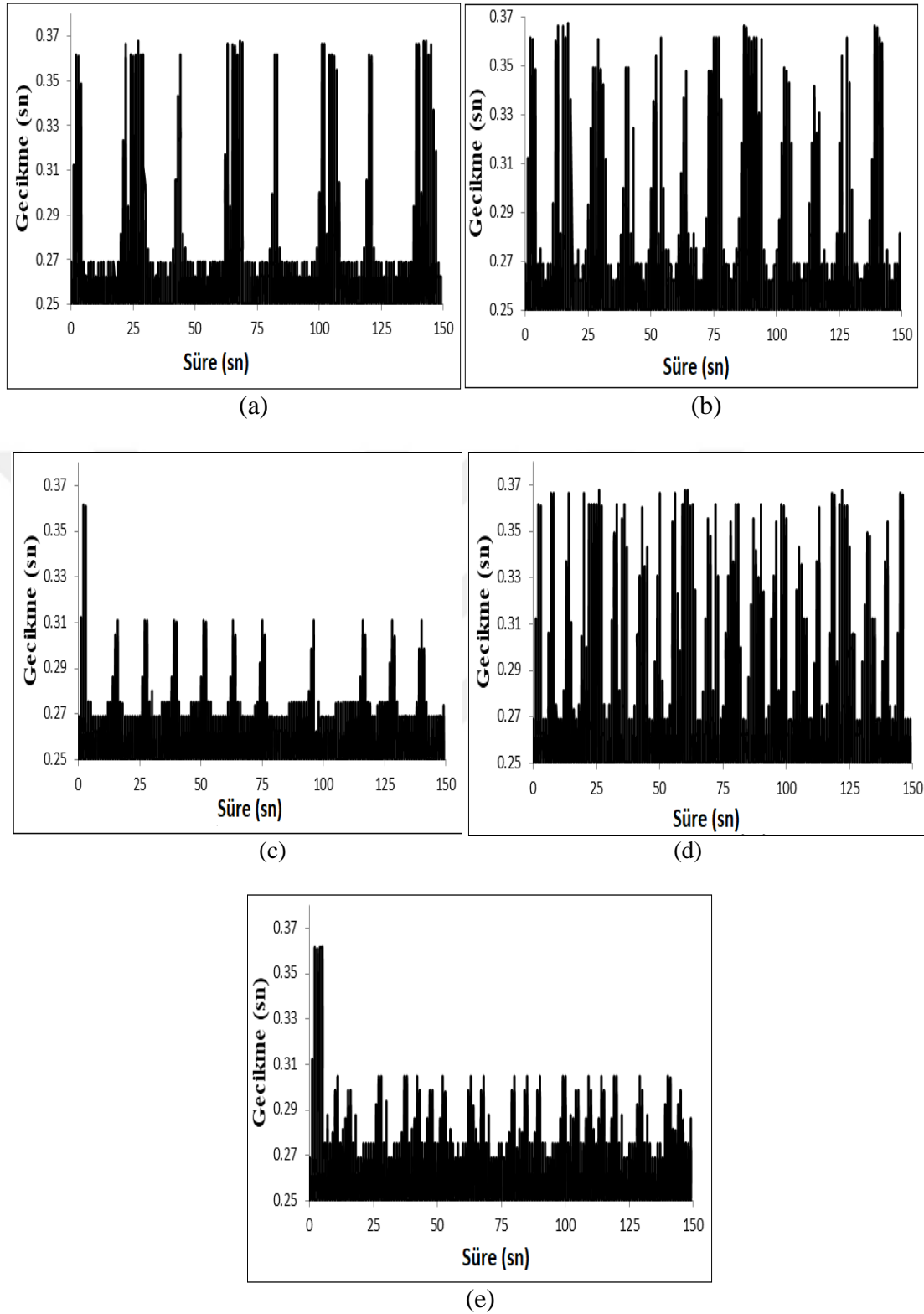
Protokoller	1 GB düşük bant genişliği için Gecikme ortalaması (ms)	10 GB yüksek bant genişliği için Gecikme ortalaması (sn)
Bic-TCP	276.801	279.589
Cubic-TCP	281.55	284.488
HS-TCP	271.199	271.219
H-TCP	288.834	288.998
S-TCP	275.08	275.066



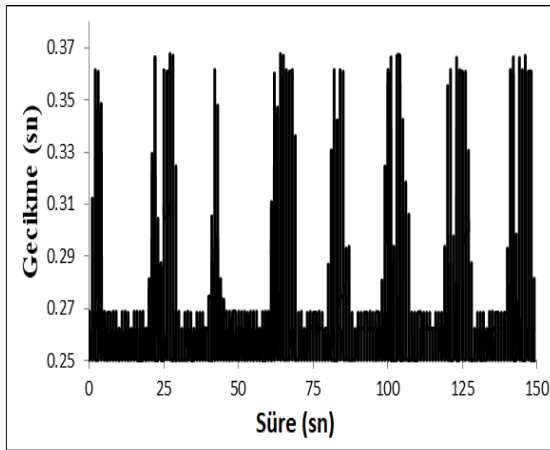
Şekil 3.17. Düşük bant genişliği için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki throughput sonuçları



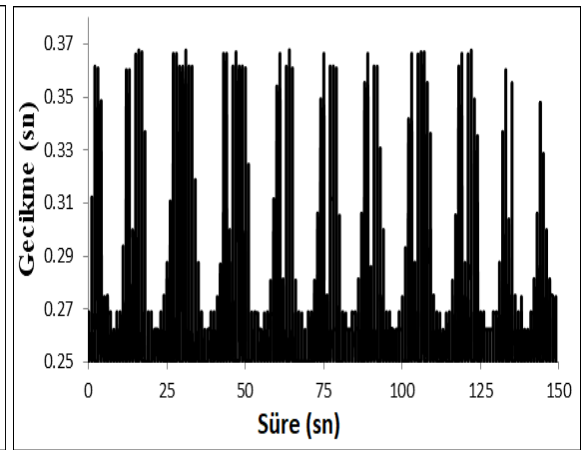
Şekil 3.18. Yüksek bant genişliği için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki throughput sonuçları



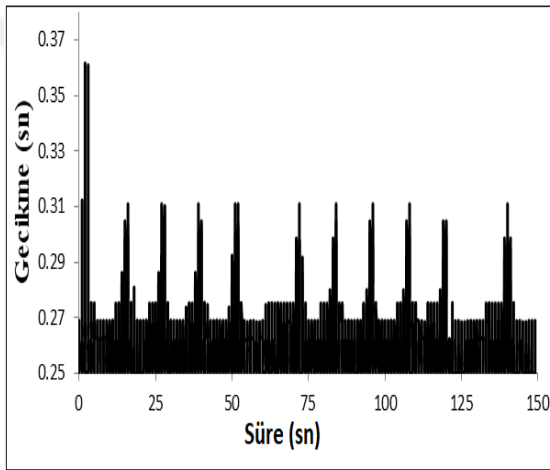
Şekil 3.19. Düşük bant genişliği için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki gecikme sonuçları



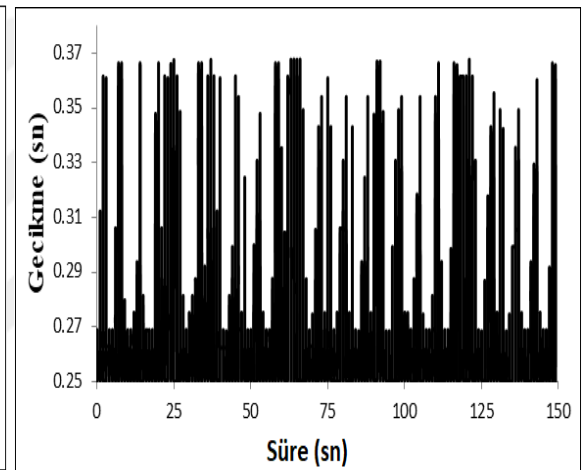
(a)



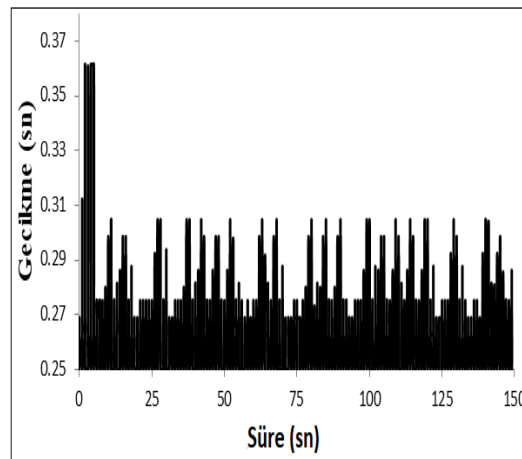
(b)



(c)

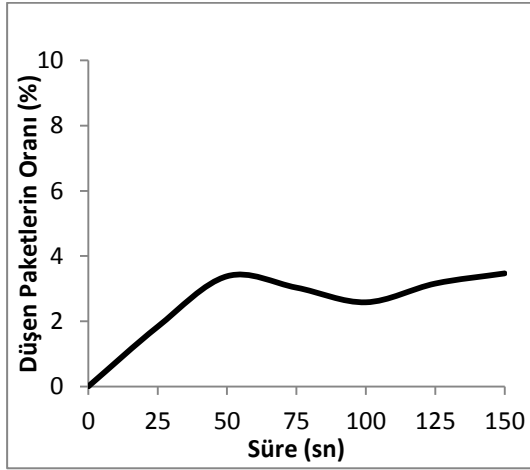


(d)

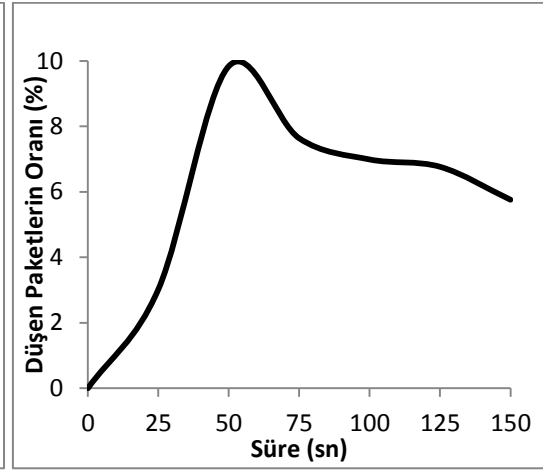


(e)

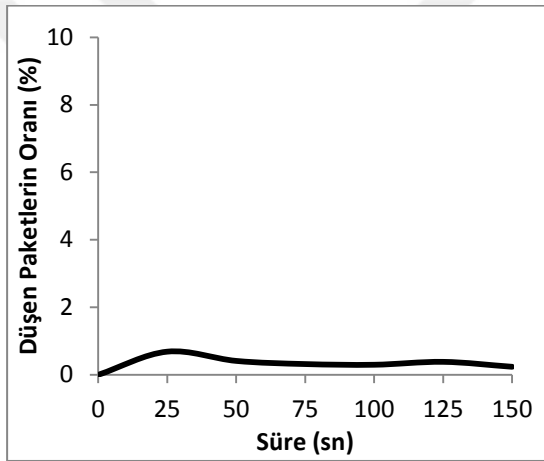
Şekil 3.20. Yüksek bant genişliği için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki gecikme sonuçları



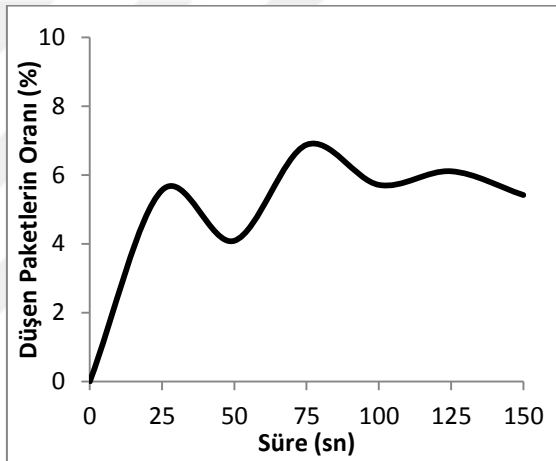
(a)



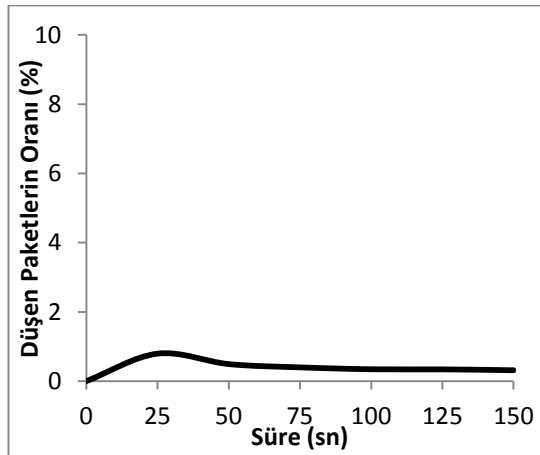
(b)



(c)

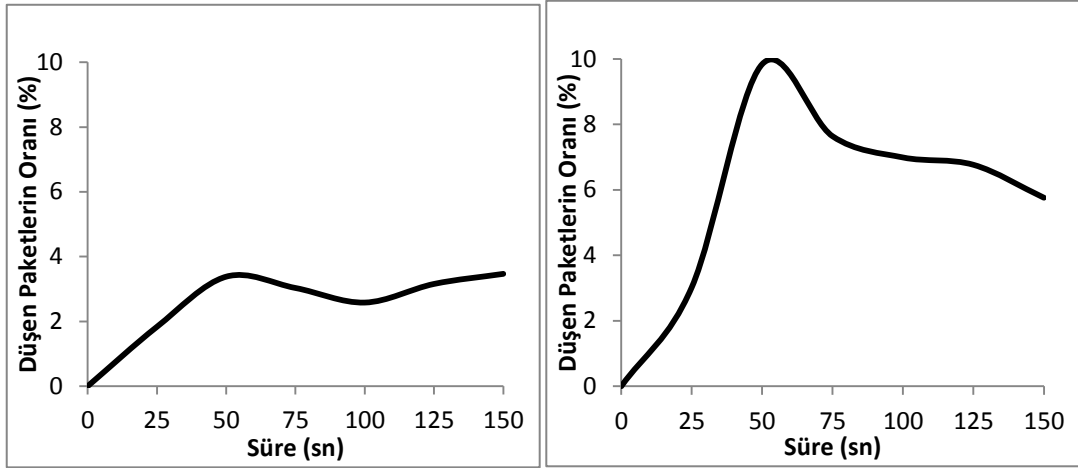


(d)



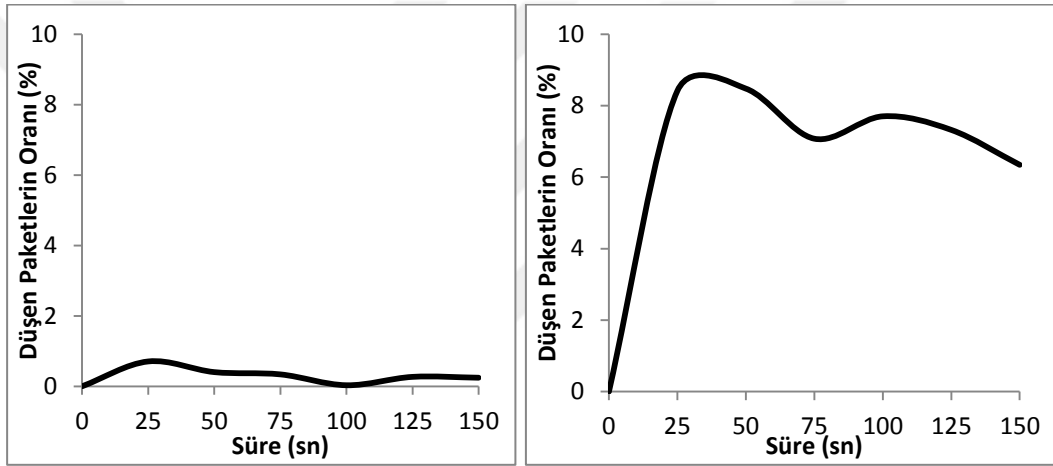
(e)

Şekil 3.21. Düşük bant genişliği için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki düşen paketlerin oranı



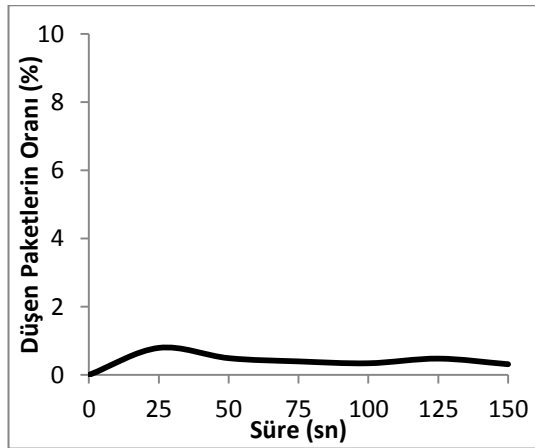
(a)

(b)



(c)

(d)



(e)

Şekil 3.22. Yüksek bant genişliği için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki düşen paketlerin oranı

RED kuyruk yönetim algoritması kullanan Bic-TCP, Cubic-TCP, HS-TCP, H-TCP ve S-TCP tıkanıklık kontrol protokolleri paket pencere boyutu değişimine göre karşılaştırılmıştır. Ağda iletilen paket boyutları 512 byte (düşük) ve 1024 byte (yüksek) olarak belirlenmiş ve sonuçlar elde edilmiştir.

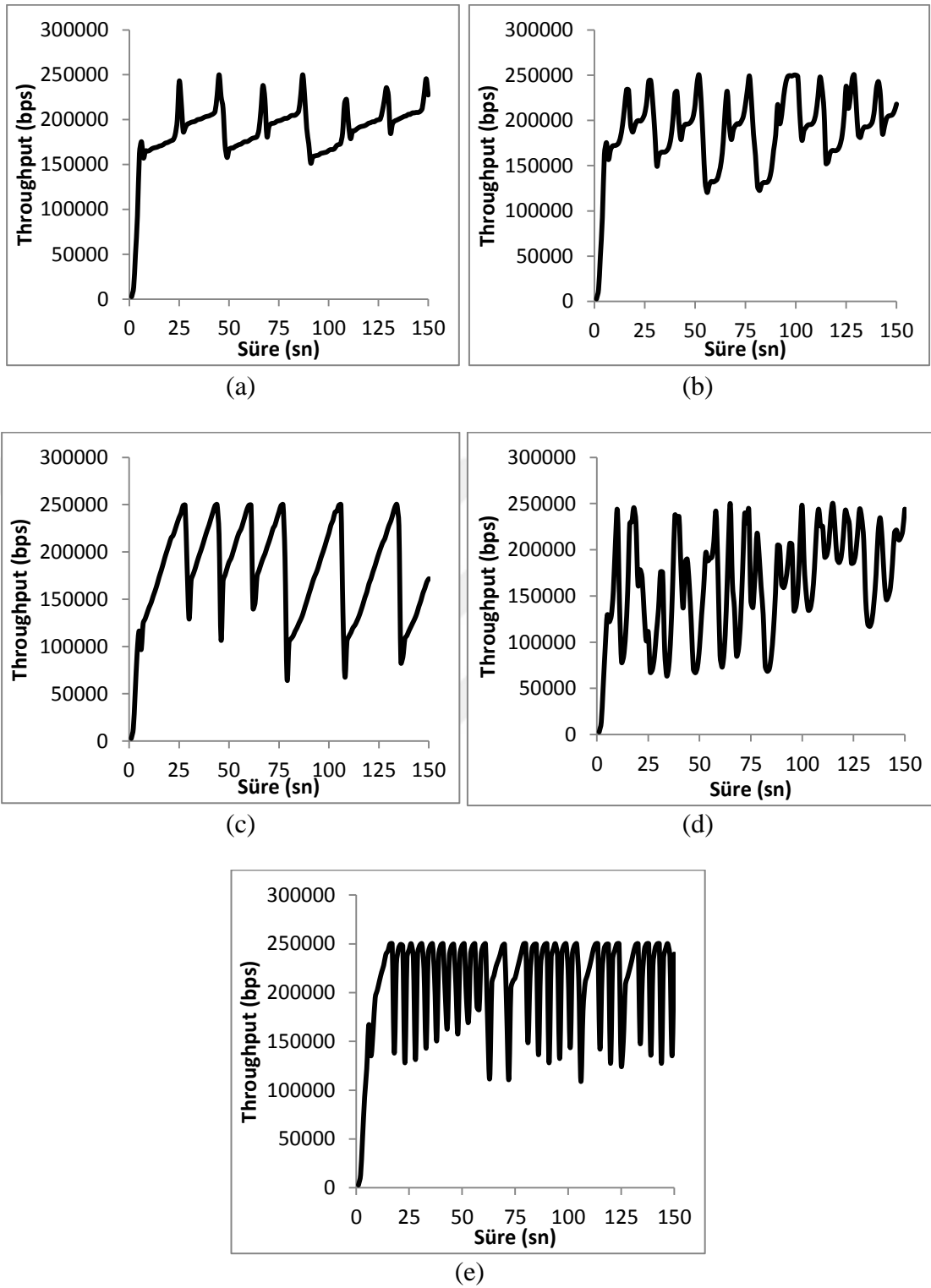
Düşük ve yüksek paket pencere boyutlarına göre Bic-TCP, Cubic-TCP, HS-TCP, H-TCP ve S-TCP tıkanıklık kontrol protokolleri kullanılan ağlardaki throughput sonuçları Şekil 3.23 ve Şekil 3.24’de gösterilmiştir. Bic-TCP protokolünün düşük ve yüksek paket pencere boyutu değerleri için en iyi başarımı elde ettiği görülmüştür.

TCP tıkanıklık kontrol protokollerinin düşük ve yüksek paket pencere boyutlarına göre gecikme sonuçları Şekil 3.25 ve Şekil 3.26’da ve gecikme ortalamaları Tablo 3.7’de gösterilmiştir. Şekiller ve tablo incelendiğinde HS-TCP protokolünün düşük ve yüksek paket pencere boyutu değerleri için en düşük gecikme değerleri elde ettiği görülmüştür.

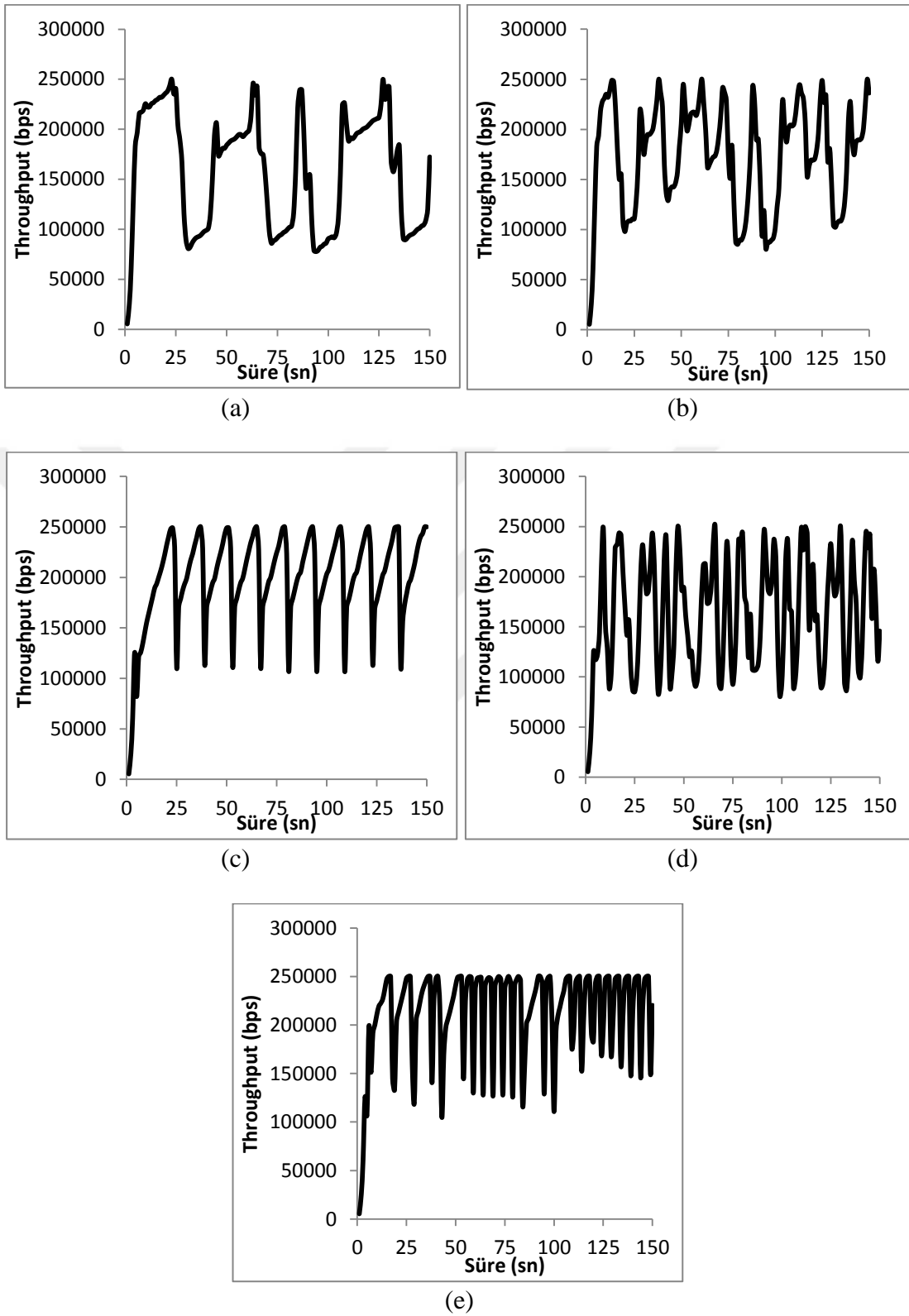
Şekil 3.27 ve Şekil 3.28’de ise TCP tıkanıklık kontrol protokollerinin düşük ve yüksek paket pencere boyutlarına göre düşen paket oranları verilmiştir. S-TCP protokolünün düşük ve yüksek bant genişlikleri için en iyi başarımları elde ettiği görülmüştür.

Tablo 3.7. Düşük ve yüksek paket pencere boyutu değerleri için ağlardaki gecikme ortalaması sonuçları

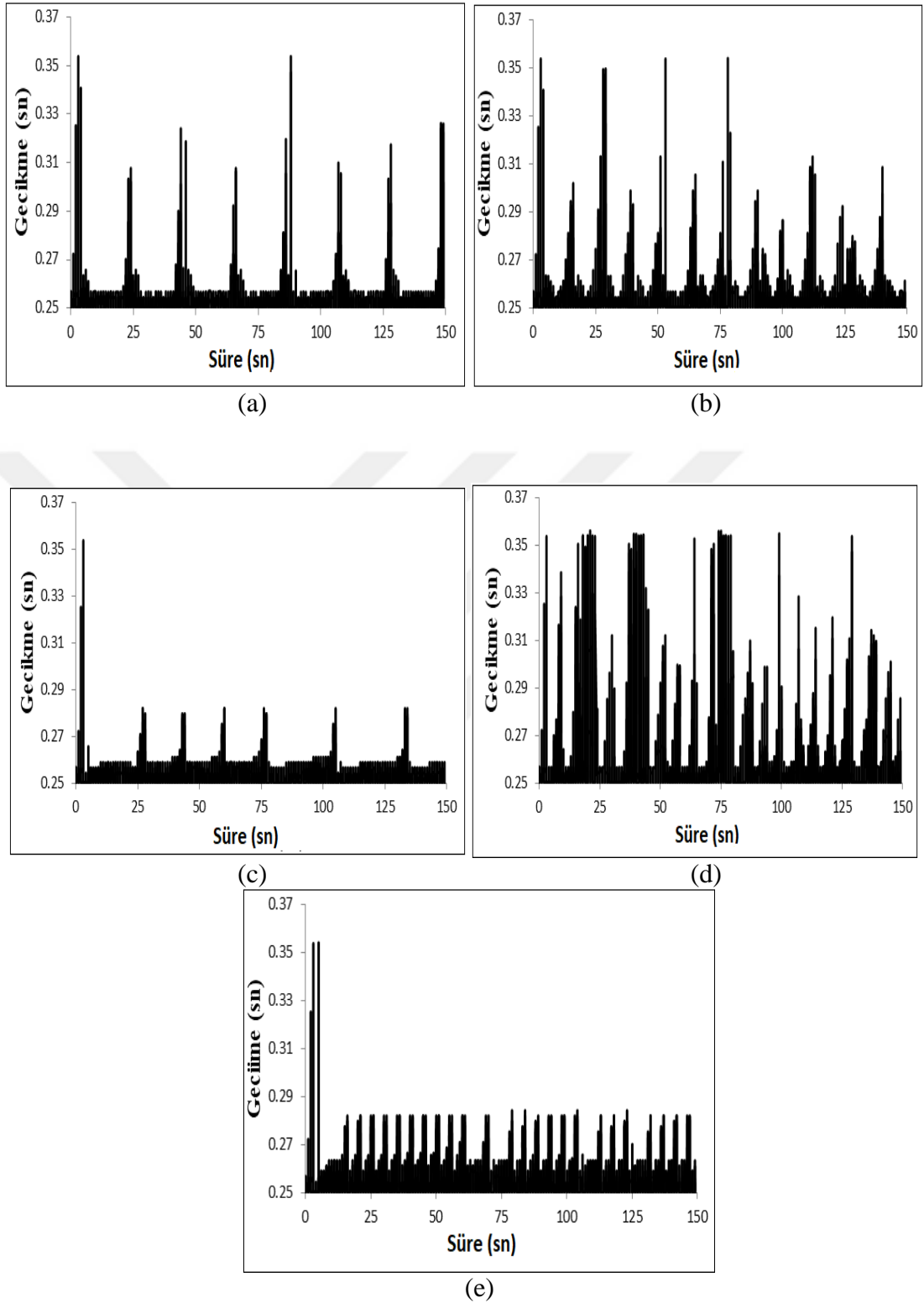
Protokoller	Paket pencere boyutu 512 byte için gecikme ortalaması (ms)	Paket pencere boyutu 1024 byte için gecikme ortalaması (sn)
Bic-TCP	262.531	270.945
Cubic-TCP	261.866	273.825
HS-TCP	258.637	264.871
H-TCP	271.076	282.931
S-TCP	263.397	269.106



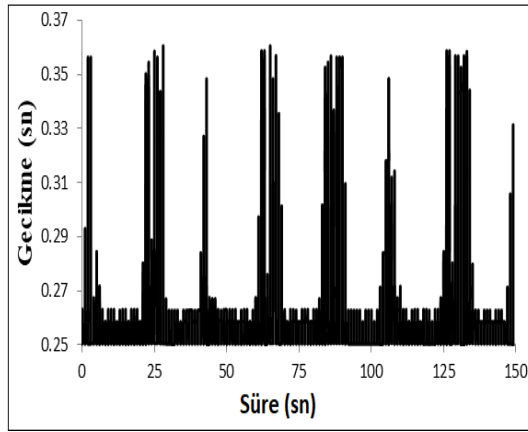
Şekil 3.23. Düşük paket pencere boyutu için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki throughput sonuçları



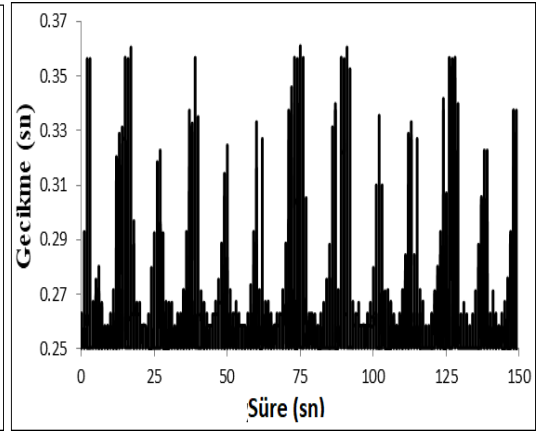
Şekil 3.24. Yüksek paket pencere boyutu için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki throughput sonuçları



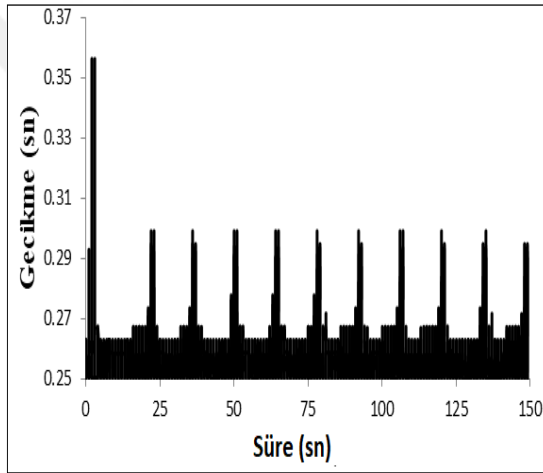
Şekil 3.25. Düşük paket pencere boyutu için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki gecikme sonuçları



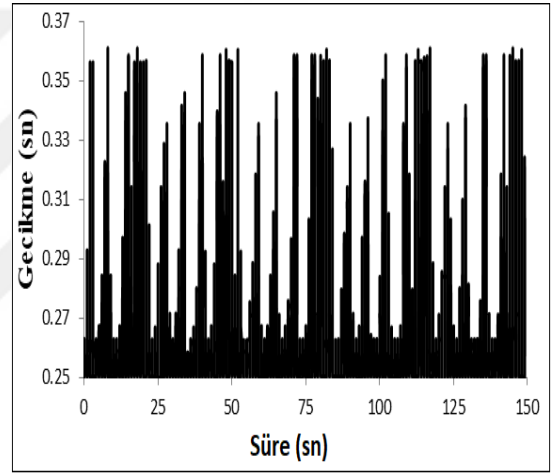
(a)



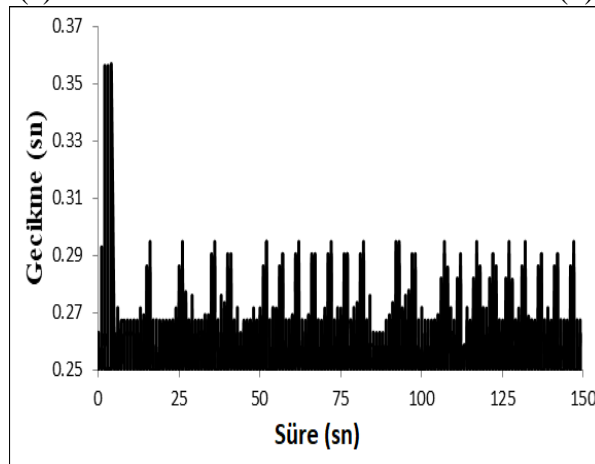
(b)



(c)

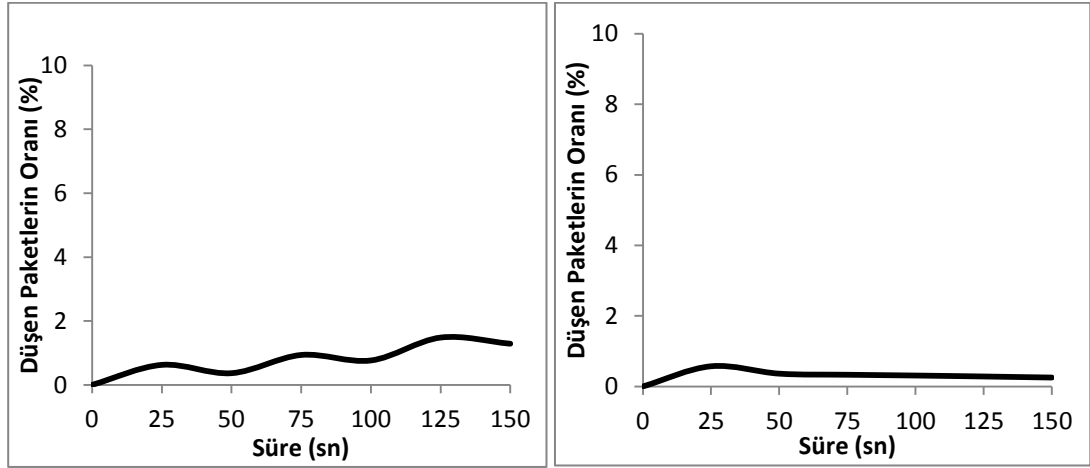


(d)



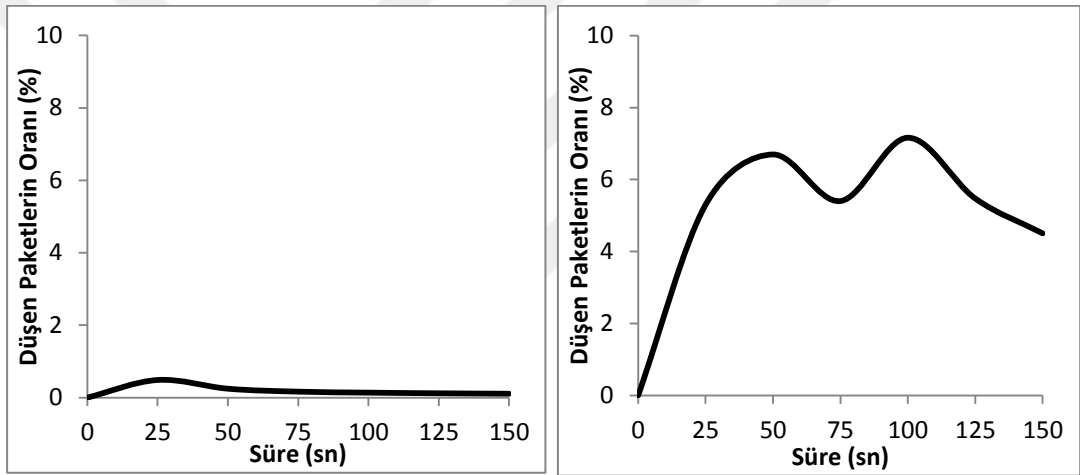
(e)

Şekil 3.26. Yüksek paket pencere boyutu için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki gecikme sonuçları



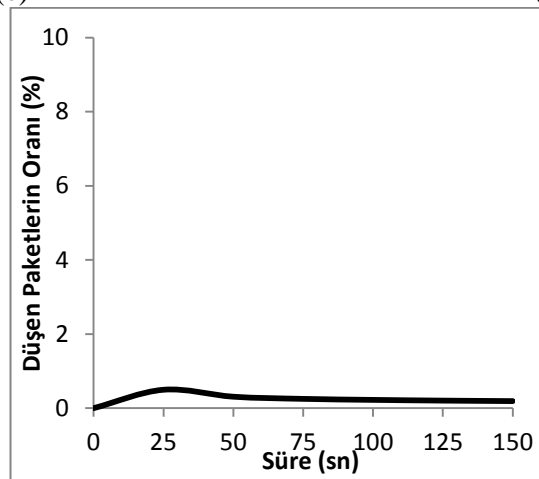
(a)

(b)



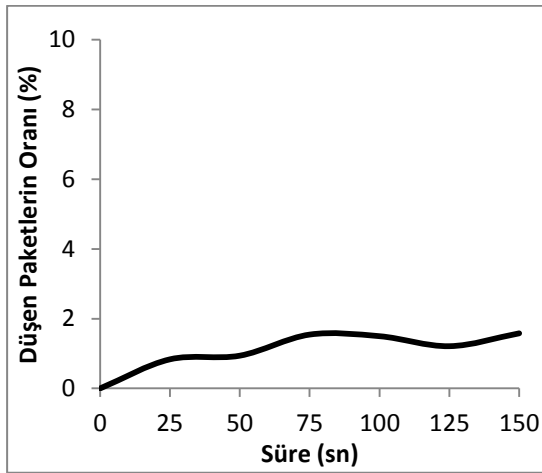
(c)

(d)

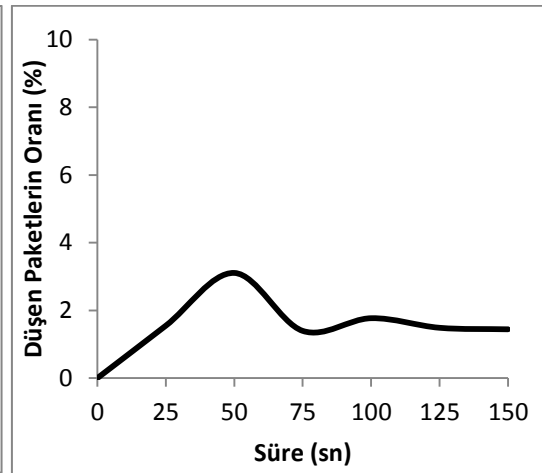


(e)

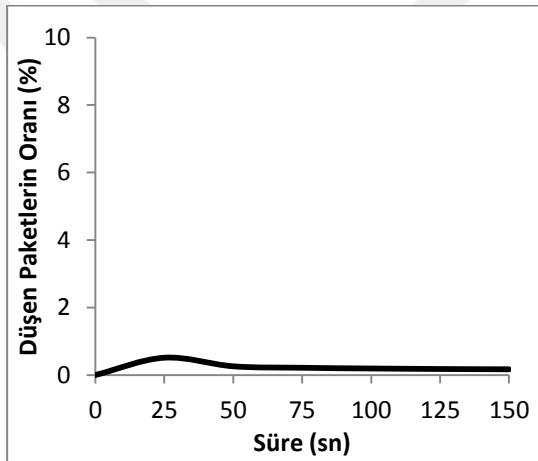
Şekil 3.27. Düşük paket pencere boyutu için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki düşen paketlerin oranı



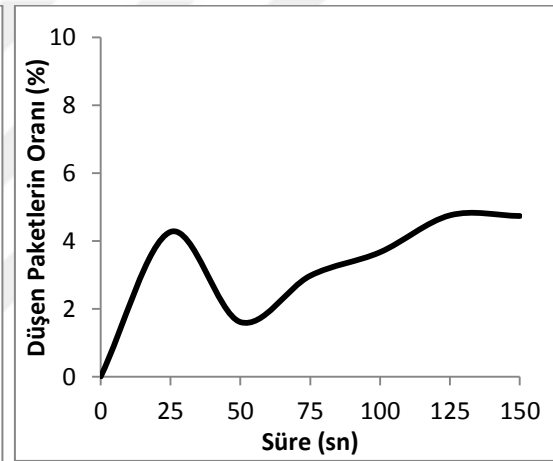
(a)



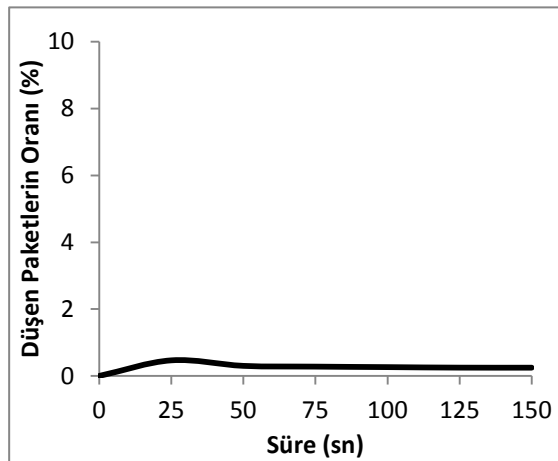
(b)



(c)



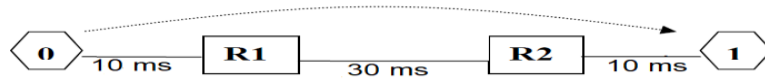
(d)



(e)

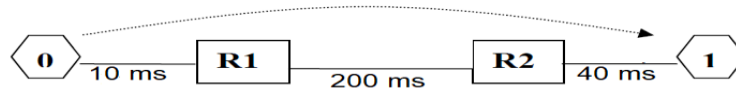
Şekil 3.28. Yüksek paket pencere boyutu için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki düşen paketlerin oranı

RED kuyruk yönetim algoritması kullanan Bic-TCP, Cubic-TCP, HS-TCP, H-TCP ve S-TCP tıkanıklık kontrol protokolleri düşük ve yüksek yayılım gecikmesi değişimine göre karşılaştırılmıştır. Düşük yayılım gecikmesi Şekil 3.29’da gösterildiği gibi belirlenmiştir. Kaynak ve hedef istasyonlar ile yönlendiriciler arasındaki yayılım gecikmesi 10 ms, yönlendiricilerin kendi aralarındaki yayılım gecikmesi 30 ms olarak seçilmiştir.



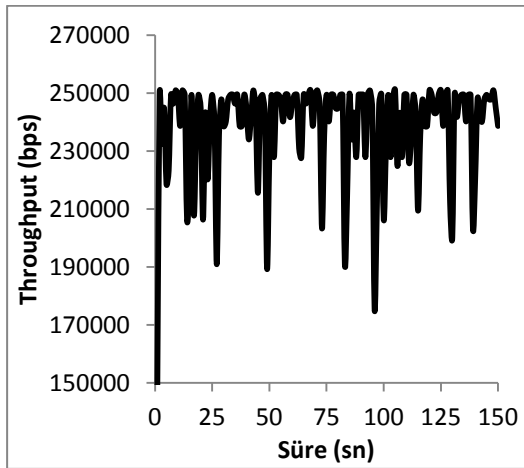
Şekil 3.29. Düşük yayılım gecikmesi topolojisi

Yüksek yayılım gecikmesi topolojisi Şekil 3.30’da gösterilmiştir. Kaynak istasyon ile R1 yönlendiricisi arasındaki yayılım gecikmek 10 ms, iki yönlendirici arasındaki yayılım gecikmek 200 ms ve hedef istasyon ile R2 yönlendiricisi arasındaki yayılım gecikmesi 40 ms olarak belirlenmiştir.

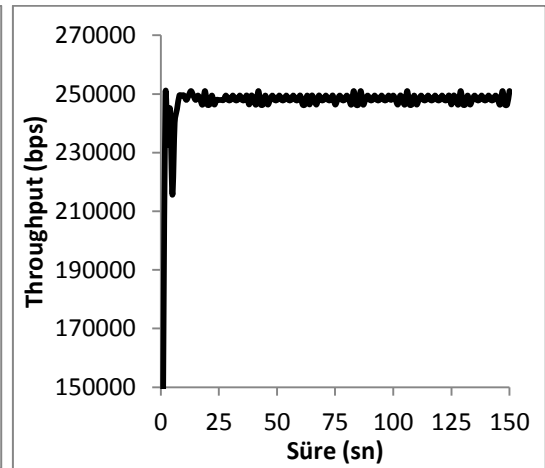


Şekil 3.30. Yüksek yayılım gecikmesi topolojisi

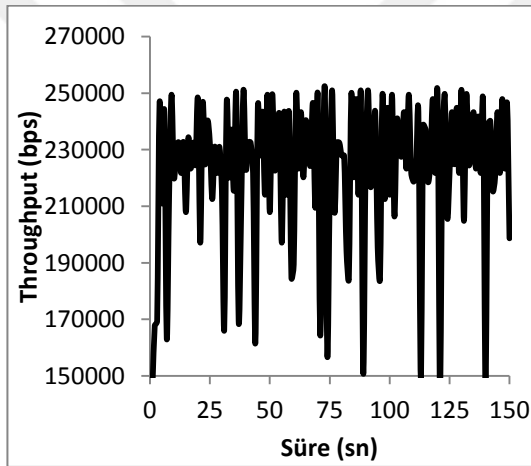
Düşük ve yüksek yayılım gecikmesine göre Bic-TCP, Cubic-TCP, HS-TCP, H-TCP ve S-TCP tıkanıklık kontrol protokolleri kullanılan ağlardaki throughput sonuçları Şekil 3.31 ve Şekil 3.32’de gösterilmiştir. Bic-TCP protokolünün düşük ve yüksek yayılım gecikmeleri için en iyi başarımı elde ettiği görülmüştür. Gecikme sonuçları Şekil 3.33 ve Şekil 3.34’de ve gecikme ortalamaları Tablo 3.8’de gösterilmiştir. Düşük yayılım gecikmesi için Cubic-TCP, yüksek yayılım gecikmesi için ise HS-TCP protokolünün en düşük gecikme değerleri elde ettiği görülmüştür. Şekil 3.35 ve Şekil 3.36’de ise düşen paket oranları verilmiştir. HS-TCP protokolünün düşük ve yüksek bant genişlikleri için en iyi başarım elde ettiği görülmüştür.



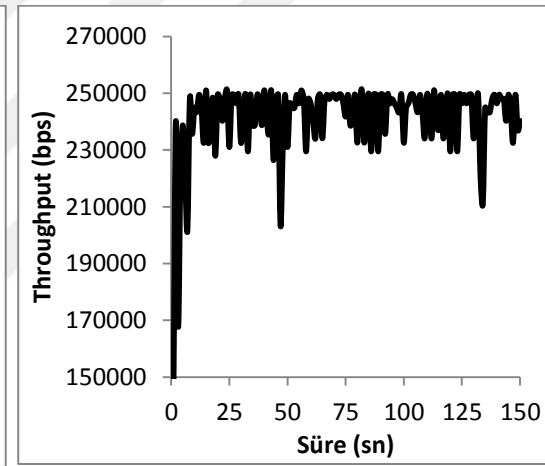
(a)



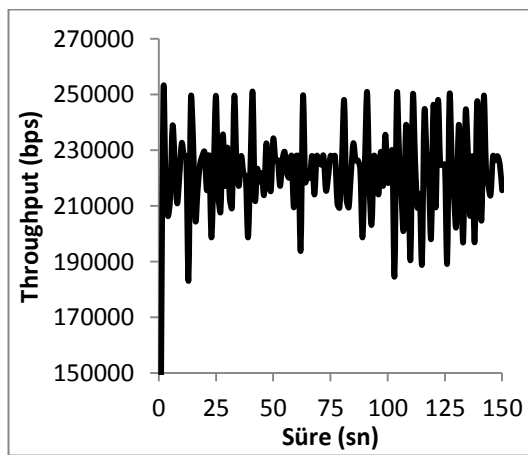
(b)



(c)

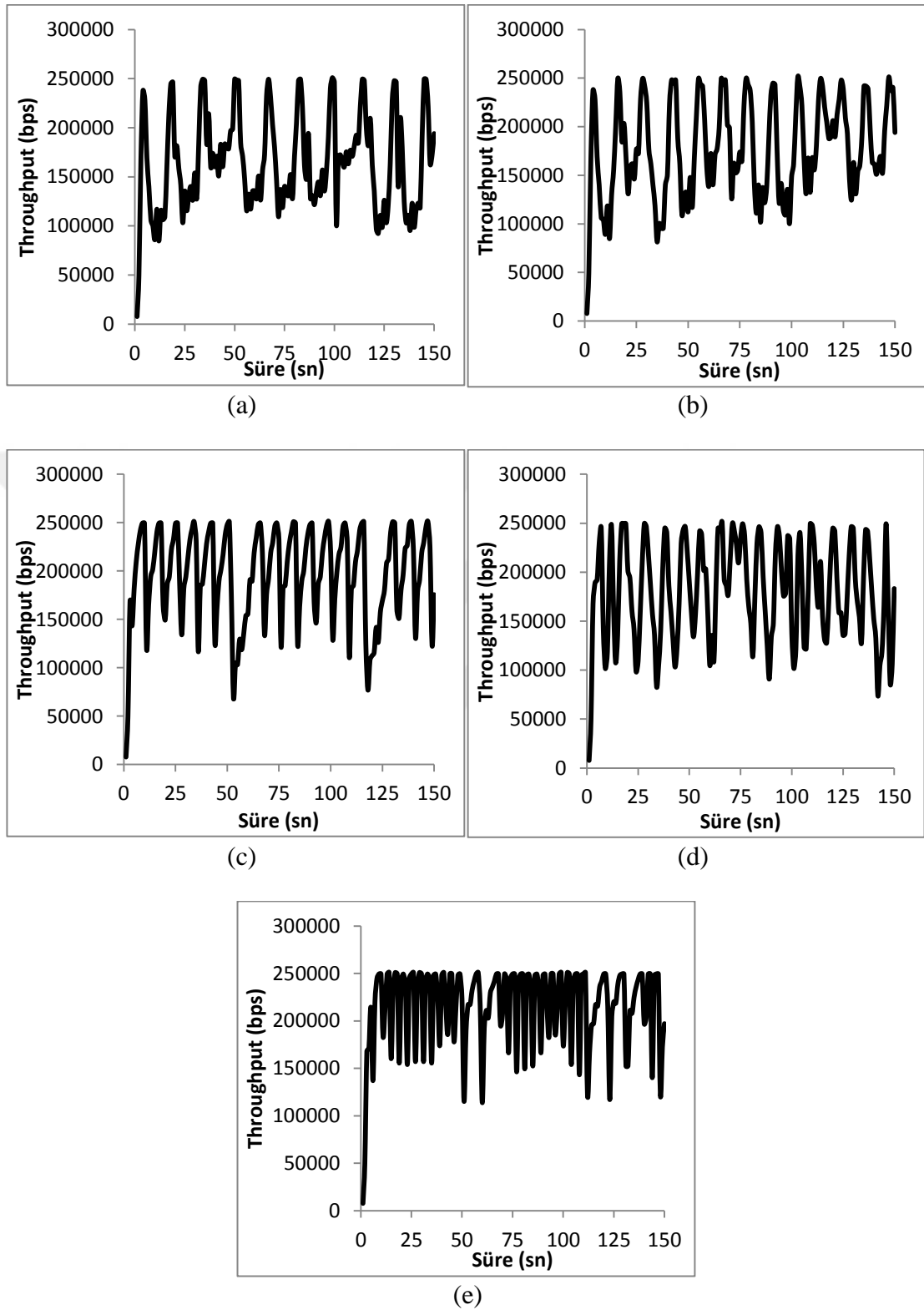


(d)

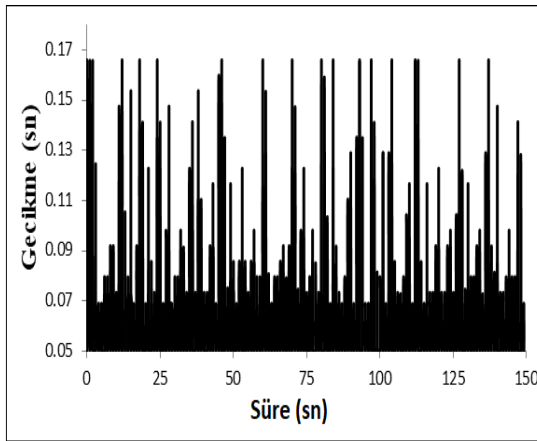


(e)

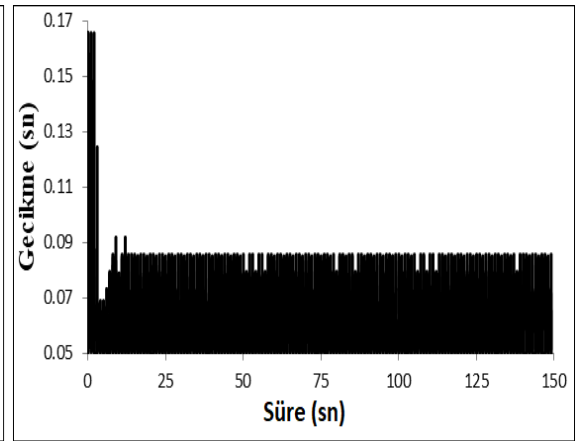
Şekil 3.31. Düşük yayılım gecikmesi için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki throughput sonuçları



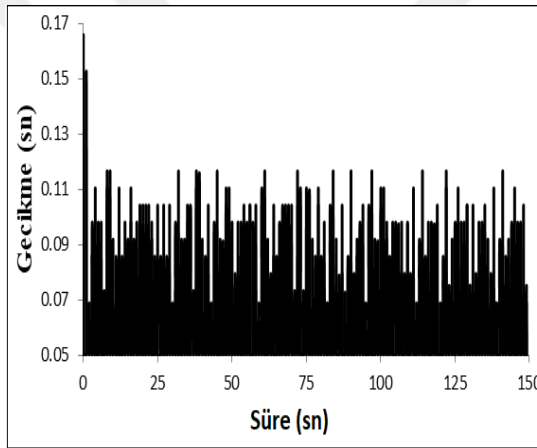
Şekil 3.32. Yüksek yayılım gecikmesi için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki throughput sonuçları



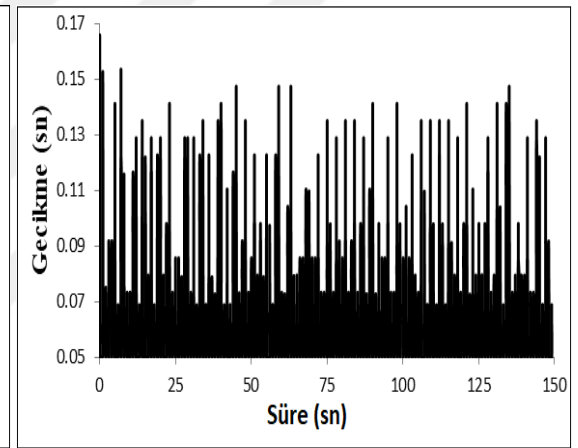
(a)



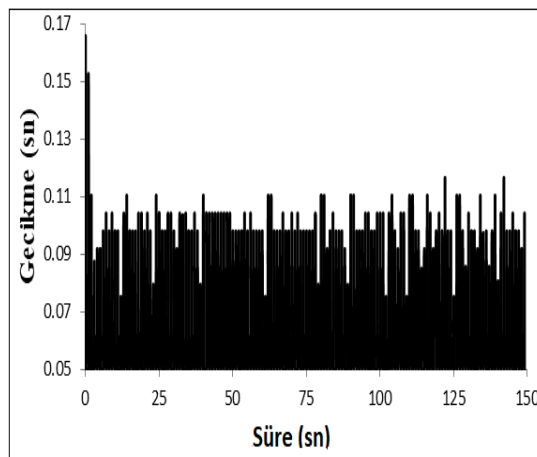
(b)



(c)

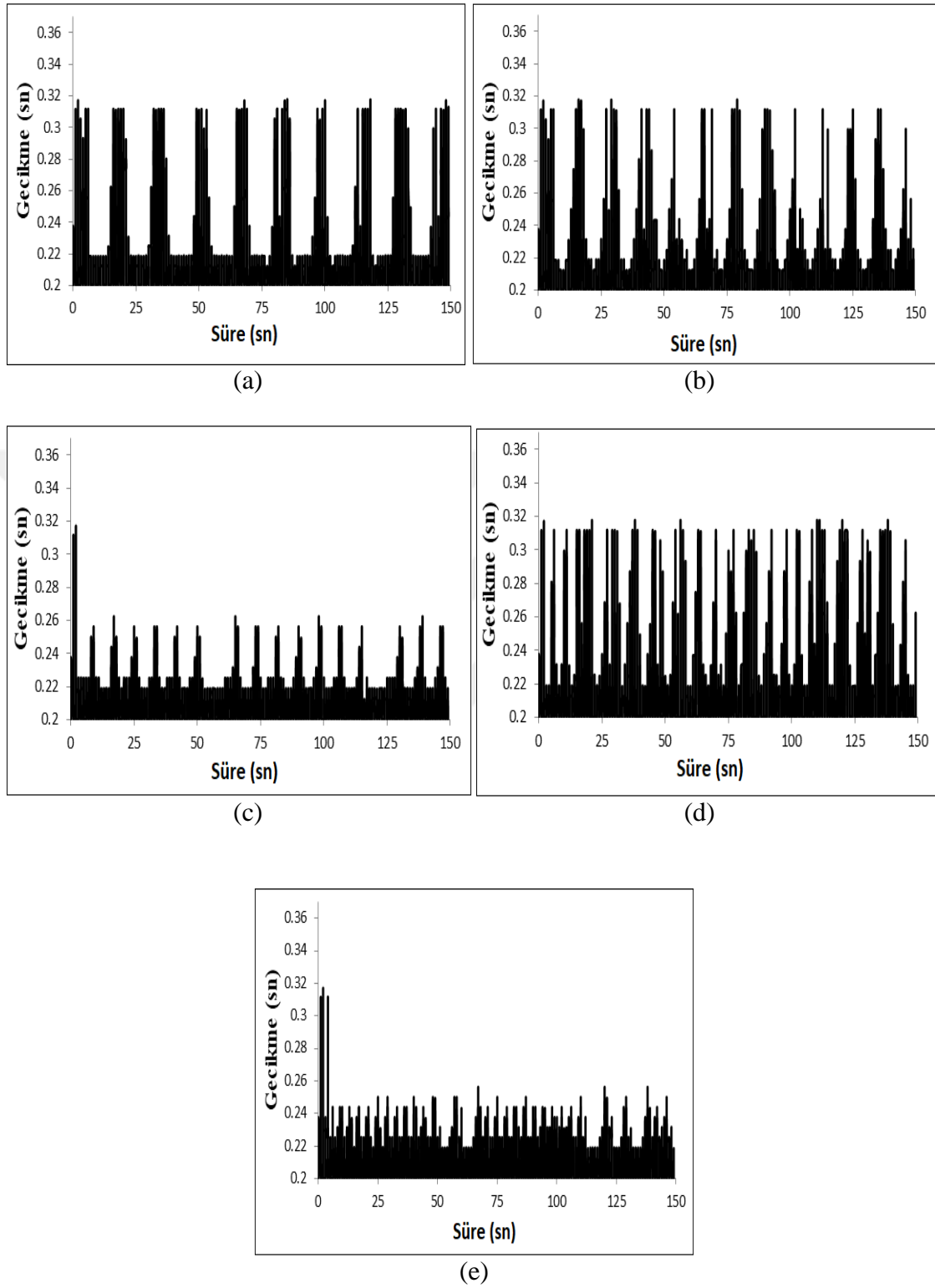


(d)

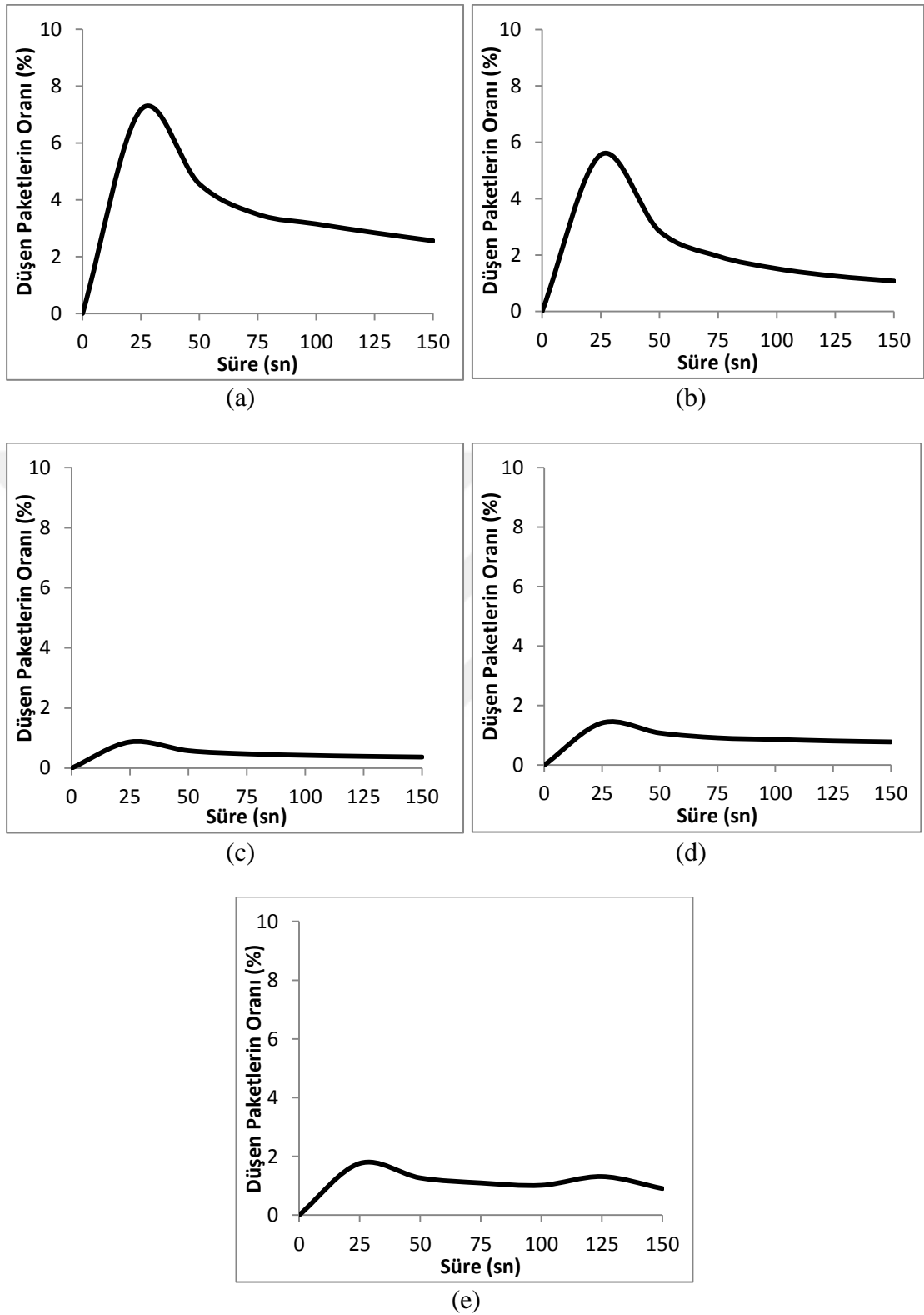


(e)

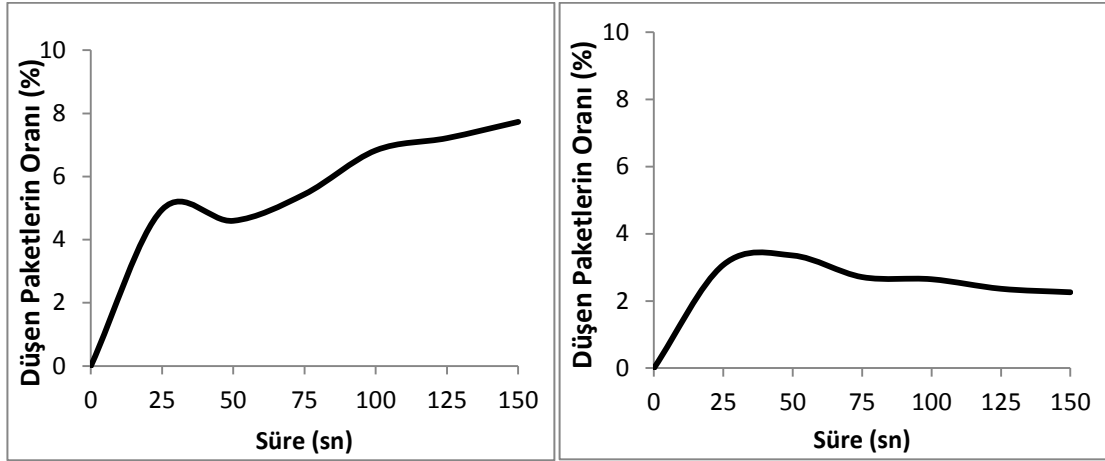
Şekil 3.33. Düşük yayılım gecikmesi için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki gecikme sonuçları



Şekil 3.34. Yüksek yayılım gecikmesi için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki gecikme sonuçları

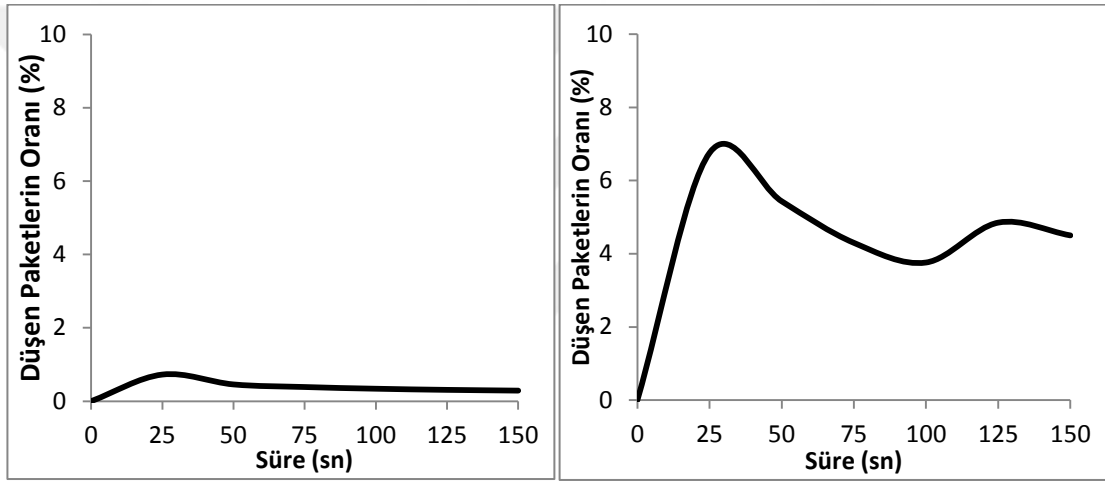


Şekil 3.35. Düşük yayılım gecikmesi için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki düşen paketlerin oranı



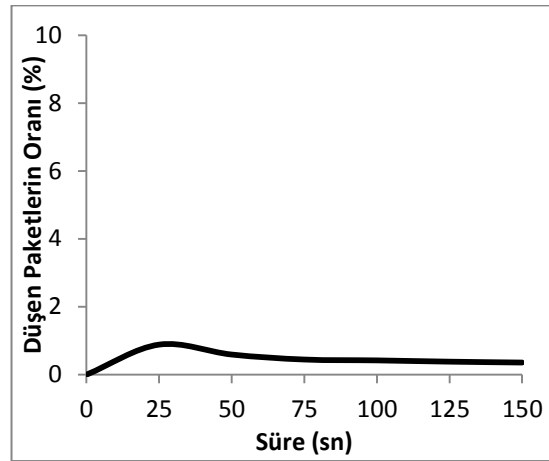
(a)

(b)



(c)

(d)



(e)

Şekil 3.36. Yüksek yayılım gecikmesi için (a) Bic-TCP, (b) Cubic-TCP, (c) HS-TCP, (d) H-TCP, (e) S-TCP protokolleri kullanılan ağlardaki düşen paketlerin oranı

Tablo 3.8. Düşük ve yüksek yayılım gecikmesi için ağlardaki gecikme ortalaması sonuçları

Protokoller	10-30-10 ms düşük yayılım gecikmesi için gecikme ortalaması (sn)	10-200-40 ms yüksek yayılım gecikmesi için gecikme ortalaması (sn)
Bic-TCP	80.4089	234.505
Cubic-TCP	77.0556	231.877
HS-TCP	78.8408	222.551
H-TCP	78.3276	238.259
S-TCP	81.9194	224.418

4. BÖLÜM

SONUÇLAR

Çalışmanın sonucunda Bic, Cubic, HS-TCP, H-TCP ve S-TCP protokollerinin SFQ kuyruk yönetim algoritması ile en iyi throughput ve düşen paketlerin oranı başarımları elde ettiği görülmüştür. RED kuyruk yönetim algoritması ile en düşük gecikme sonuçları elde edilmiştir. Protokoller karşılaştırıldığında ise Bic-TCP protokolünün en iyi başarımları elde ettiği görülmüştür. CUBIC, HS-TCP ve S-TCP protokollerin gecikme ve düşen paketlerin oranı en iyi protokoller olduğunu anlaşılmıştır.

KAYNAKLAR

- [1] Gelle.S., 2008. Performance of high-speed TCP Protocols over NS-2 TCP Linux. Old Dominion University, Department of Computer Science, Master' Thesis, Virginia, 120s.
- [2] Bateman, M., Bhatti, S., Bigwood, G., Rehunathan, D., Allison, C., Henderson, T., Miras, D., 2008. A comparison of TCP behaviour at high speeds using ns-2 and Linux, pp. 30-37. *ACM the 11th communications and networking simulation symposium*, April 14 – 17, Canada.
- [3] Sharma, P., 2006. Performance Analysis of High-Speed, Clemson University, Department of Computer Science, Master' Thesis, South Carolina, 100s.
- [4] Hatano, T., Shigeno, H., Okada, K., 2007. Tcp-friendly congestion control for highspeed network, pp. 10-10. *International SympAosium on pplications and the Internet*, SAINT 2007, Japan.
- [5] Dangi, R., Shukla, N., 2012. A new congestion control algorithm for high speed networks. **International Journal of Computer Technology and Electronics Engineering (IJCTEE)**, 2(1):218-221.
- [6] Weigle, M. C., Sharma, P., Freeman, R., 2006. Performance of competing high-speed TCP flows, pp. 476-487. *International Conference on Research in Networking*, Berlin, Heidelberg.
- [7] Reddy, K. S., Reddy, L., 2008. A survey on congestion control mechanisms in high speed networks. *International Journal of Computer Science and Network Security (IJCSNS)*, 8(1):187-195.
- [8] Tokuda, K., Hasegawa, G., Murata, M., 2003. Performance analysis of HighSpeed TCP and its improvement for high throughput and fairness against TCP Reno connections. In *High-Speed Networking Workshop (HSN)*, Osaka University.
- [9] Lukaseder, T., Bradatsch, L., Erb, B., Van Der Heijden, R. W., Kargl, F., 2016. A comparison of TCP congestion control algorithms in 10G networks, pp. 706-714. *41st Conference on Local Computer Networks (LCN)*, 7-10 November, Dubai, United Arab Emirates.

- [10] Wei, D. X., Jin, C., Low, S. H., Hegde, S. 2006. FAST TCP: motivation, architecture, algorithms, performance. **IEEE/ACM transactions on Networking**, 14(6):1246-1259.
- [11] Jagannathan, S., 2002. End to end congestion control in high-speed networks, pp.547-556. *27th Annual Conference on Local Computer Networks Proceedings LCN*, 6-8 Nov, Tampa, USA.
- [12] Katabi, E. D., Katabi, D., Handley, M., Rohrs, C., Tellabs, M. L., 2002. Internet Congestion Control for Future High Bandwidth-Delay Product, pp.89-102. *Proceedings of the 2002 SIGCOMM Conference*, August 19 - 23, New York, USA.
- [13] Barman, D., Smaragdakis, G., Matta, I., 2004. The effect of router buffer size on highspeed TCP performance, pp.1617-1621. *Global Telecommunications Conference*, 29 Nov.-3 Dec, Dallas, USA.
- [13] Hatano, T., Fukuhara, M., Shigeno, H., Okada, K., 2003. TCP-friendly Sqrt TCP for High Speed Networks. APSITT, pp. 455–460.
- [14] Khalil, E. A., 2012. A modified congestion control algorithm for evaluating high BDP networks. **International Journal of Computer Science and Network Security (IJCSNS)**, 12(11):84-93.
- [15] Tan, K., Song, J., Zhang, Q., Sridharan, M., 2006. A compound TCP Approach for High-Speed and Long Distance Networks. *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, 23-29 April, Barcelona, Spain.
- [16] Tokuda, K., 2003. On Congestion Control Mechanism of High-Speed TCP. Osaka University, Department of Informatics and Mathematical Science, Master's Thesis, Japan, 115s.
- [17] Edwan, T., 2010. Improved algorithms for TCP congestion control. Loughborough University, Institutional Repository, Doctora Thesis, UK, 85s.

- [18] Pedro, J., 2010 . End-to-end congestion control algorithms for the Internet. University of Lisbon, Computer Engineering and Computers Dept., Master's Thesis, Portugal, 100s.
- [19] Flo Li, Y. T., Leith, D., Shorten, N., 2007. Experimental evaluation of TCP protocols for high-speed networks. **Transactions on Networking IEEE (ToN)**, 15(5):1109-1122.
- [20] Abed, G. A., Ismail, M., Jumari, K., 2011. A comparison and analysis of congestion window for HS-TCP, Full-TCP, and TCP-Linux in long term evolution system model, pp.358-362. *IEEE Conference on Open Systems (ICOS)*, 25-28 Sept, Langkawi, Malaysia.
- [21] Xu, W., Zhou, Z., Pham, D. T., Ji, C., Yang, M., Liu, Q., 2011. Hybrid congestion control for high-speed networks. **Journal of Network and Computer Applications**, 34(4):1416-1428
- [22] Nabeshima, M., Yata, K., 2004. Improving the convergence time highspeed TCP, pp. 19-23. *12th International Conference Networks (ICON 2004)*, 19-19 November, Singapore, Singapore.
- [23] Nabeshima, M., Yata, K. 2006. Performance evaluation and comparison of transport protocols for fast long-distance networks. **IEICE transactions on communications**, 89(4):1273-1283.
- [24] Abed, J. B., Sinda, L., Mani, M. A., Mbarek, R., 2012. Comparison of high speed congestion control protocols. **International Journal of Network Security & Its Applications**, 4(5):15.
- [25] Song, K. T. J., Zhang, Q., Sridharan, M. 2006. Compound TCP: A scalable and TCP-friendly congestion control for high-speed networks. **Proceedings of PFLDnet**.
- [26] Mohammed, Y., 2006. Evaluation of TCP Based Congestion Control Algorithms Over High-Speed Networks. Osmania University, Department of Computer Science, Master's Thesis , Hyderabad, 90s.

- [27] Ahmad, S., Mustafa, A., Ahmad, B., Bano, A., Hosam, S., 2009. Comparative study of congestion control techniques in high speed networks, **International Journal of Computer Science and Information Security**, 8(35).
- [28] Li, Y. T., Leith, D., Shorten, R. N., 2007. Experimental evaluation of TCP protocols for high-speed networks. **IEEE/ACM Transactions on Networking (ToN)**, 15(5):1109-1122.
- [29] Kelly, T., 2003. Scalable TCP: Improving performance in highspeed wide area networks. **ACM SIGCOMM computer communication Review**, 33(2):83-91.
- [30] Hua, W., Jian, G., 2012. Analysis of TCP BIC Congestion Control Implementation, pp. 781-784. *International Conference on Computer Science & Service System (CSSS)*, 11-13 August, Nanjing, China.
- [31] Xu, L., Harfoush, K., Rhee, I., 2004. Binary increase congestion control (BIC) for fast long-distance networks, pp. 2514-2524. *In INFOCOM 2004*, 7-11 March, Hong Kong, China.
- [32] Katabi, D., Handley, M., Rohrs, C., 2002. Congestion control for high bandwidth-delay product networks. **ACM SIGCOMM computer communication review**, 32(4):89-102.
- [33] Petrov, I., Janevski, T., 2015. Evolution of TCP in High Speed Networks. **International Journal of Future Generation Communication and Networking**, 8(2):137-186.
- [34] [47] Pujeri, U. R., Palaniswamy, V., Ramanathan, P., Pujeri, R., 2015. Comparative analysis and comparison of various AQM algorithm for high speed, **Indian Journal of Science and Technology**, 8(35):1.
- [35] Matsuura, Y., 2006. Performance Evaluation of TCP Congestion Control Mechanism based on Inline Network Measurement. Osaka University, Department of Information Networking, Master's Thesis, Japan, 125s.
- [36] Bullot, H., Les Cottrell, R., Hughes-Jones, R. 2003. Evaluation of advanced TCP stacks on fast long-distance production networks. **Journal of Grid Computing**, 1(4):345-359.

- [37] Tahiliani, M. P., Shet, K. C., & Basavaraju, T. G., 2013. Comparative Study of High-Speed TCP Variants in Multi-Hop Wireless Networks. **International Journal of Computer Theory and Engineering**, 5(5):802.
- [38] Ha, S., Rhee, I., Xu, L., 2008, CUBIC: a new TCP-friendly high-speed TCP variant. **ACM SIGOPS operating systems review**, 42(5):64-74.
- [39] Nabeshima, M., Yata, K., 2006. Performance evaluation and comparison of transport protocols for fast long-distance networks. **IEICE transactions on communications**, 89(4):1273-1283.
- [40] Qureshi, B., Othman, M., Subramaniam, S., Wati, N. A., 2013. QTCP: improving throughput performance evaluation with high-speed networks. **Arabian Journal for Science and Engineering**, 38(10):2663-2691.
- [41] Tahiliani, M. P., Shet, K. C., Basavaraju, T. G., 2013. Comparative Study of High-Speed TCP Variants in Multi-Hop Wireless Networks. **International Journal of Computer Theory and Engineering**, 5(5):802.
- [42] Mo, J., Walrand, J., 2000. Fair end-to-end window-based congestion control. **IEEE/ACM Transactions on networking**, 8(5):556-567.
- [43] Abed, J. B., Sinda, L., Mani, M. A., Mbarek, R. 2012. Comparison of high speed congestion control protocols. **International Journal of Network Security & Its Applications**, 4(5):15.
- [44] Liu, S., Basar, T., & Srikant, R. (2005). Exponential-RED: a stabilizing AQM scheme for low-and high-speed TCP protocols. **IEEE/ACM transactions on networking**, 13(5):1068-1081.
- [45] Haider, A., Sirisena, H., Pawlikowski, K., Ferguson, M. J. 2001. Congestion control algorithms in high speed telecommunication networks, pp.88-97. Proceedings of 36th Annual ORSNZ, *Christchurch*, Christchurch, New Zealand,.
- [46] Singh, S., Balveer, S., 2013. Performance & Analysis of REM, RED, & GREEN AQM Algorithms in Congestion Control, **International Journal of Advanced Research in Computer Science and Software Engineering**, 3.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı, Soyadı: Noor Faisal Mohammed MOHAMMED

Uyruğu: İrak Cumhuriyeti (TC)

Doğum Tarihi ve Yeri: 29 Haziran 1988, Kerkük

Medeni Durumu: Bekâr

Tel: 05318326923

email: kir6kuk@gmail.com

Yazışma Adresi: Mevlana Mahallesi, Cemil baba cadesi, Sarmaşak apartmanı Blok 25/13, Daire 38 Talas, Kayseri

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Yüksek Lisans	EÜ Fen Bilimleri Enstitüsü	2018
Lisans	Kerkük Teknik Fakültesi, Bilgisayar Böl.	2010
Lise	Celal Riza Efendi	2006

YABANCI DİL

İngilizce

Arapça