

**EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
(DOKTORA TEZİ)**

**ÖLÇEKLENEBİLİR VIDEO
KODLAYICI/ÇÖZÜMLEYİCİ İLE ÇOK KATMANLI
GÖREVDEŞ AĞLARDA VIDEO AKIŞLANDIRMA**

Müge SAYIT

Tez Danışmanı: Prof. Dr. E. Turhan TUNALI

Uluslararası Bilgisayar Anabilim Dalı

**Bilim Dalı Kodu: 619.02.04.
Sunuş Tarihi: 08.06.2011**

**Bornova-İZMİR
2011**

Müge SAYIT tarafından doktora tezi olarak sunulan “Ölçeklenebilir Video Kodlayıcı/Çözümleyici ile Çok Katmanlı Görevdeş Ağlarda Video Akışlandırma” 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 08.06.2011 tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

Jüri Üyeleri:

İmza

Jüri Başkanı	: Prof. Dr. E. Turhan TUNALI	
Raportör Üye	: Yard.Doç.Dr. Adil ALPKOÇAK	
Üye	: Prof. Dr. A. Murat TEKALP	
Üye	: Prof. Dr. Mehmet Emin DALKILIÇ	
Üye	: Prof. Dr. Bahar KARAOĞLAN	

ÖZET**ÖLÇEKLENEBİLİR VIDEO KODLAYICI/ÇÖZÜMLEYİCİ İLE****ÇOK KATMANLI GÖREVDEŞ AĞLARDA VIDEO****AKIŞLANDIRMA**

SAYIT, Müge

Doktora Tezi, Uluslararası Bilgisayar Enstitüsü

Tez Yöneticisi: Prof. Dr. E. Turhan TUNALI

Haziran 2011, 120 sayfa

İnternet üzerinden video akışlandırma uygulamalarına yönelik giderek artan ilgiyi karşılamak üzere görevdeş ağlar üzerinde video akışlandırma sistemleri hem ticari hem de akademik çevreler tarafından önerilmiş ve görevdeş ağlarda video akışlandırma önemli araştırma alanlarından biri haline gelmiştir.

Bu tezde, görevdeş ağ üzerinde ölçeklenebilir video kodlayıcı/çözümleyici kullanan hata dayanıklı bir video akışlandırma sistemi uygulaması için algoritma ve protokoller geliştirilmiştir. Altta yatan kaplama mimarisinde düğümler arası gecikme tabanlı gruptandırma yapılmış ve kaplama mimarisi üzerinde video akışlandırma sağlamak üzere çokluguönderim ağaçları oluşturulmuştur.

Önerilen algoritmalar benzetim ortamı üzerinde çeşitli deney parametreleri kullanılarak test edilmiş ve elde edilen sistem kazanımı raporlanmıştır. Video akışlandırma sistemlerinin önemli parametreleri incelendiğinde, önerilen mimarinin benzerlerinden daha iyi sonuçlar verdiği gözlemlenmiştir.

Anahtar sözcükler: Video akışlandırma, görevdeş ağlar, hata dayanıklı çokluguönderim ağaçları, ölçeklenebilir kodlayıcı/çözümleyici.

ABSTRACT**HIERARCHICAL PEER-TO-PEER VIDEO STREAMING WITH
SCALABLE VIDEO CODEC**

SAYIT, Müge

PhD. in International Computer Institute

Supervisor: Prof. Dr. E. Turhan TUNALI

June 2011, 120 pages

In order to meet growing demand of streaming video from the Internet, peer to peer video streaming systems are proposed by both academia and commercial industry and peer-to-peer streaming became one of the major research areas.

In this thesis, algorithms and protocols are proposed for resilient video streaming application with scalable video codec over a peer-to-peer network. Delay based clustering is implemented and multicast trees are constructed over the overlay network architecture in order to implement video streaming application.

Proposed algorithms are tested on a simulator by using various set of parameters and system performance is reported. It has been seen that, when essential parameters of video streaming applications are considered, the proposed architecture yielded better results than other similar studies in the literature.

Keywords: Video streaming, peer-to-peer networks, resilient multicast trees, scalable video codec.

TEŞEKKÜR

Doktora çalışmam boyunca, bilgi ve deneyimlerini benimle paylaşan, öğrenmeyi, açık fikirli olmayı, çalışma disiplinini ve bilimsel yaratıcılığı büyük bir özveri ve sabırla bana öğreten, her türlü fikir ve desteğini her zaman yanımda hissettiğim sayın danışmanım Prof. Dr. E. Turhan TUNALI'ya; değerli fikirleri, problemlere baktığı farklı açılar ve sağladığı bilimsel tartışma ortamı ile akademik yaşamıma tartışılmaz katkısı bulunan sayın jüri üyem A. Murat TEKALP'e; değerli görüşleri, fikirleri, paylaşımları ve çalışmaya sağladığı katkılar sebebi ile sayın jüri üyem Prof. Dr. Mehmet Emin DALKILIÇ'a teşekkürlerimi bir borç bilirim.

Tez çalışmamda jüri üyesi olarak görev almış ve fikirleriyle çalışmamı yönlendirmiş sayın hocalarım Prof. Dr. Bahar KARAOĞLAN ve Prof. Dr. Kayhan ERCİYES'e teşekkürlerimi sunarım.

Tez süresince fikir ve desteklerinden dolayı proje yürütücümüz olan Yrd. Doç. Dr. Hasan BULUT'a, projedeki çalışma arkadaşlarım araştırma görevlisi ve doktora öğrencileri Sercan DEMİRCİ ve Yağız KAYMAK'a; doktora eğitimim boyunca her türlü imkanı sunan Uluslararası Bilgisayar Enstitüsü'ne, Uluslararası Bilgisayar Enstitüsü'nde bana her zaman destek olmuş tüm mesai arkadaşlarıma, "108M411" numaralı "Görevdeş Ağlarda Ölçeklenebilir Gruplandırma ve Video Akışlandırma" isimli proje kapsamında doktora çalışmamı maddi olarak destekleyen Türkiye Bilimsel ve Teknolojik Araştırma Kurumu'na (TÜBİTAK), tez çalışmam boyunca benden manevi desteklerini esirgemeyen eşim, annem, babam, kardeşim, kayınvalidem, kayınpederim ve varlığıyla en büyük destekçilerimden biri olan canım oğluma teşekkürlerimi sunarım.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	V
ABSTRACT	Vii
TEŞEKKÜR	İx
İÇİNDEKİLER	Xi
ŞEKİLLER DİZİNİ	Xvi
ÇİZELGELER DİZİNİ	XX
KISALTMALAR DİZİNİ	XXi
1.GİRİŞ	1
2. LİTERATÜR ÖZETİ	4
2.1 Kaplama Mimarisi	4
2.1.1 Gecikme tabanlı kaplama mimarileri	4
2.1.2 İlgi tabanlı kaplama mimarileri	6
2.2 Akışlandırma Yöntemleri	6
2.2.1 Örgü tabanlı akışlandırma yöntemi	7
2.2.2 Ağaç tabanlı akışlandırma yöntemi	8
2.2.3 Karma tabanlı akışlandırma yöntemi	11
2.3 Çoklוגönderim Ağaç Oluşturma Algoritmaları	12

İÇİNDEKİLER (devam)

Sayfa

2.4 Hata Dayanıklılığı.....	13
2.4.1 Reaktif yöntemler.....	14
2.4.2 Proaktif yöntemler.....	15
2.5 Video Kodlama Teknikleri.....	17
2.5.1 Ölçeklenebilir kodlama	17
2.5.2 Çoklu tanımlama kodlama	17
2.6 Literatür Özet Tablo.....	18
3. ALTYAPI OLARAK KULLANILAN KAPLAMA MİMARİSİ	20
3.1 Hiyerarşik Katmanların Oluşturulması	21
3.2 Grup Bölünmesi	21
3.3 Lider Seçme Algoritması	22
3.4 Sonlu Durum Makinası	22
4. ÇOKLUGÖNDERİM AĞAÇLARININ OLUŞTURULMASI.....	30
4.1 Chang-Roberts Algoritması ile Elde Edilen Çoklugönderim Ağaçları.....	31
4.2 Kapasite-Farkındalıklı (Capacity-Aware) Çoklugönderim Ağaç Oluşturma Algoritması.....	33
4.2.1 Problem formülasyonu.....	33

İÇİNDEKİLER (devam)

	<u>Sayfa</u>
4.2.2 Maliyet tablosu	34
4.2.3 Algoritma	35
5. GÖREVDEŞ AĞ ÜZERİNDE HATA DAYANIKLILIĞINA SAHİP ÇOKLUGÖNDERİM AĞAÇLARININ OLUŞTURULMASI..	37
5.1 Videoların Talep Edilmesi ve Aranması	37
5.1.1 Liderin grup içindeki videoları belirlemesi	38
5.1.2 Düğümlerin video talebinde bulunması	38
5.1.3 Videoların aranması	39
5.2 Akışlandırma Liderleri	39
5.3 Çoklugoönderim Ağaçları Üzerinde Hata Dayanıklılığı	41
5.4 Hiyerarşik Katmanlar Üzerinde Çoklugoönderim Ağaçlarının Oluşturulması	46
5.5 Düğüm Kapasitelerinin Hiyerarşik Katmanlar Üzerinde Dağıtılması	48
6. GÖREVDEŞ AĞ ÜZERİNDE VIDEO AKIŞLANDIRMA OTURUMU	52
6.1 Reaktif Bileşen	52
6.2 Kalite Uyarlama	55
7. YEDEK EBEVEYN SEÇİMLERİNDE ANALİTİK YAKLAŞIM ...	57

İÇİNDEKİLER (devam)

Sayfa

8. BENZETİM SONUÇLARI.....	72
8.1 Chang Roberts Algoritması ile Oluşturulan Çoklugoönderim Ağaçları Test Sonuçları.....	72
8.2 Kapasite Farkındalıklı Algoritma ile Oluşturulan Çoklugoönderim Ağaçları Test Sonuçları.....	74
8.2.1 Kapasite farkındalıklı ağaç oluşturma algoritması karşılaştırmalı test sonuçları.....	74
8.2.2 Çoklugoönderim ağaçlarının hiyerarşik katmanlar üzerinde uygulanması.....	79
8.2.3 Çoklugoönderim ağaçları ile yapılan video akışlandırma çalışmaları.....	81
8.3 Yedek Ebeveynler ile Hata Dayanıklı Akışlandırma Test Sonuçları.....	85
8.4 Farklı Yedek Ebeveyn Seçim Yaklaşımı ile Ağaç Oluşturma ve Hata Dayanıklı Akışlandırma Test Sonuçları.....	93
8.4.1 En iyi yedek hücre – servis hücresi oranı	94
8.4.2 Karşılaştırmalı deney sonuçları.....	96
9. SONUÇLAR	102
KAYNAKLAR DİZİNİ	106
ÖZGEÇMİŞ	112

İÇİNDEKİLER (devam)

Sayfa

EKLER

Ek. 1 Türkçe – İngilizce Terimler Sözlüğü

Ek 2. Kaplama mimarisi sonlu durum makinası

Ek 3. Kaplama Mimarisinin Oluşumu ile İlgili Algoritmalar

Ek 4. Video Talep ve Arama Algoritmaları

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
4.1. Chang-Roberts algoritması sözde kodu.	32
4.3. Gecikme – Bant genişliği çiftine bağlı ağırlık değerleri ızgarası.....	34
4.2. Çoklugu gönderim ağaç oluşturma algoritması sözde kodu.	35
5.1. Yedek ebeveyn seçimi ile birlikte çoklugu gönderim ağaç oluşturma algoritması.....	42
5.2. Fırsatçı yöntemine göre yedek ebeveyn seçim algoritması.	44
5.3 Çevrimsel sıralı yöntemine göre yedek ebeveyn seçim algoritması.....	45
5.4. Kontrol ve akışlandırma liderleri ile hiyerarşik katmanlar.	47
5.5. Hiyerarşik katmanlar üzerinde çoklugu gönderim ağaç örneği.	49
6.1. Bağlantı kurulum algoritması.....	53
6.2. Örnek hata senaryosu.	54
7.1. Kökün yedek hücrelerini gösteren bir hata senaryosu.	58
7.2. Kökün çocuk sayısını belirleyen algoritma.....	59
7.3. Çoklugu gönderim ağaç oluşturma algoritması.	60
7.4. Uzlaşma algoritması.....	60
7.5. Hatalardan sonra ağaç tamiri a) Ağacın ilk görünümü, b) Tamirden sonra ağaç görünümü.	61
7.6. Farklı düğüm kapasiteleri ve hata oranları için karşılaştırmalı ağaç hata olasılıkları.....	65

ŞEKİLLER DİZİNİ (devam)

<u>Şekil</u>	<u>Sayfa</u>
7.7. Farklı düğüm ve çocuk sayılarına göre ağaçtan kopması beklenen düğüm sayısı.....	71
8.1. Düğüm sayısı – zıplama adedi histogramı.....	73
8.2. Düğüm sayısı – çocuk sayısı histogramı.	74
8.3. Kaynaktan hedefe zıplama sayısı cinsinden toplam uzaklık.	75
8.4. Alınan video bithızı OYF grafiği.....	77
8.5. Karşılaştırmalı deney sonuçları.	78
8.6. Düğüm sayısı – zıplama adedi histogramı.....	79
8.7.Düğüm Sayısı – zıplama Sayısına bağlı ortalama bantgenişliği değerleri	80
8.8. Düğüm sayısı – çocuk sayısı histogramı.	81
8.9. Ağaç üzerindeki düğümlerin gecikme cinsinden kaynağa uzaklıkları.	81
8.10. Düğüm derecesi.	83
8.11. Düğüm gecikme değerleri.	83
8.12. Sabit kanal kapasitesi için alınan video kalitesi.	84
8.13. Değişken kanal kapasitesi için alınan video kalitesi.	84
8.14. Kurulan ağaçlarda karşılaştırmalı gecikme KDF grafiği.....	86

ŞEKİLLER DİZİNİ (devam)

<u>Şekil</u>	<u>Sayfa</u>
8.15. Kurulan ağaçlarda karşılaştırmalı zıplama KDF grafiği.....	86
8.16. Ağaç sayısına bağlı ortalama zıplama sayısı.....	87
8.17. Ağaç sayısına bağlı ortalama toplam gecikme.....	87
8.18. Düşük seviye hata oranına ilişkin grafikler a) 1 ağaç için alınan ortalama kalite, b) Yeniden bağlanmak için geçen süre, c) Ağaç sayısına bağlı alınan ortalama kalite.....	89
8.19. Orta seviye hata oranına ilişkin grafikler a) 1 ağaç için alınan ortalama kalite, b) Yeniden bağlanmak için geçen süre, c) Ağaç sayısına bağlı alınan ortalama kalite.....	90
8.20. Yüksek seviye hata oranına ilişkin grafikler a) 1 ağaç için alınan ortalama kalite, b) Yeniden bağlanmak için geçen süre, c) Ağaç sayısına bağlı alınan ortalama kalite.....	92
8.21. İkili düğümlerin gecikme histogramı.....	94
8.22. Farklı yedek hücre ayırma yaklaşımlarının tamir süreleri a) Ortalama ağaç tamir süresi, b) En uzun ağaç tamir süresi.....	95
8.23. Kaynaktan hedefe bağlantı maliyetleri.....	97
8.24. Kaynaktan hedefe zıplama sayısı.....	97
8.25. Kaynaktan hedefe toplam gecikme.....	97
8.26. Düşük hata seviyesi için ağaç tamir süresi.....	99
8.27. Orta hata seviyesi için ağaç tamir süresi.....	99

ŞEKİLLER DİZİNİ (devam)

<u>Şekil</u>	<u>Sayfa</u>
8.28. Yüksek hata seviyesi için ağaç tamir süresi.	100
8.29. Ağaç tamir süresi a) Ortalama ağaç tamir süresi, b) Maksimum ağaç tamir süresi.....	101

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
2.1. Literatür özet tablo.....	18
3.1. NORMAL_DÜĞÜM durumundaki bir düğüm için durum ve çıktı tablosu. .	24
3.2. LİDER durumundaki bir düğüm için durum ve çıktı tablosu.	26
3.3. YEDEK_LİDER durumundaki bir düğüm için durum ve çıktı tablosu.....	27
3.4. LİDER_SEÇİMİ durumundaki bir düğüm için durum ve çıktı tablosu.....	27
7.1. Ağaçtan istemsiz kopma analizi notasyonu.	65
8.1. Karşılaştırmalı ortalama değerler.	78
8.2. Farklı türdeki mesajların karşılaştırmalı ortalama değerleri.	93
8.3. Farklı yedek hücre seçim yaklaşımlarına göre ağaç kazanımı.....	96
8.4. Karşılaştırmalı gecikme değerleri.	98

KISALTMALAR DİZİNİKısaltmalar

BDS	Bağlantı Deneme Sayısı.
Kbps	Saniyede Kilobit Hızı (Kilobit per Second).
KDF	Kümülatif Dağılım Fonksiyonu (Cumulative Distribution Function).
Mbps	Saniyede Megabit Hızı (Megabit per Second).
MDC	Çoklu Tanımlama Kodlama (Multiple Description Coding).
OYF	Olasılık Yoğunluk Fonksiyonu (Probability Density Function).
P2P	Peer to Peer (Görevdeş).
RTT	Round Trip Time (Paket Dolaşım Zamanı).
SDM	Sonlu Durum Makinası (Finite State Machine).
SVC	Scalable Video Codec (Ölçeklenebilir Video Kodlayıcı/Çözümleyici).

1.GİRİŞ

İnternet tarafından sağlanan olanakların günden güne gelişmesi ile birlikte, İnternet üzerinden video verisi gönderme uygulamaları giderek yaygınlaşmıştır. Klasik sunucu – istemci tabanlı mimarilerde gönderilecek olan videonun boyutu ya da istemci sayısı arttıkça, sunucu istenen talebi karşılayamamaktadır. Bu sebeple, video gönderme sürecinde, sunucunun üzerindeki yükün alınmasını sağlayan görevdeş ağlarda çokluguönderim sistemleri, hem ticari hem akademik anlamda etkin araştırma konularından biri haline gelmiştir (Liu et al., 2008).

IP protokolünün hata kontrol – kurtarma, ölçeklenebilirlik ve ağ yönetiminin yapılması gibi eksiklikleri sebebiyle IP üzerinden etkin bir çokluguönderim uygulamasının yapılması mümkün değildir. Dolayısıyla İnternet üzerinden çokluguönderim uygulamaları için genel yaklaşım, altta yatan fiziksel ağ topolojisinden yararlanarak bir kaplama mimarisi oluşturulmasıdır. Görevdeş ağlarda kaplama mimarisinin oluşturulmasına duyulan ihtiyaç sadece gerçek zamanlı video akışlandırma sistemleri için değil, dosya paylaşım sistemleri için de geçerliliğini korumaktadır. Ancak çokluguönderim amacını taşıyan dosya paylaşım sistemleri, yine aynı amaçlı video akışlandırma uygulamaları tarafından gerçek zaman kısıtlamaları dolayısıyla kullanılamamıştır (Hei et al., 2007). Dosya paylaşım sistemlerinde alıcıların aldığı veri paketleri ile ilgili bir zaman ve sıra sınırlaması bulunmaz iken, video akışlandırma sistemlerinde her video paketinin bir oynatım süresi bulunmakta ve oynatım süresinden önce alıcının eline geçmeyen video paketleri alıcı tarafından göz ardı edilmektedir.

Görevdeş ağlar üzerinden çalışan video akışlandırma sistemi tasarımında birçok zorluk öne çıkmaktadır. Görevdeş ağların boyutu, ağı oluşturan düğümlerin fikir değiştirerek ağdan çıkmaları nedeniyle sürekli olarak değişmektedir. Sisteme dahil olacak düğüm sayısında bir sınırlama olmadığı için, ölçeklenebilirliğin sağlanabileceği bir kaplama mimarisi kurulması gerekmektedir. Kaplama mimarisi bakım protokolü, sisteme giriş çıkışlara bağlı olarak dinamizm sağlamalıdır. Tez kapsamında ölçeklenebilirlik ve görevdeş ağların dinamizmine karşılık verebilecek hiyerarşik katmanlı bir kaplama mimarisi önerilmiştir. Sisteme giren düğümler coğrafi konumlarına göre gruplandırılmakta, oluşan grup sayısı arttıkça hiyerarşik katmanlar yapılandırılmaktadır. Sisteme yeni giren düğüm, hiyerarşik katmanlar arasında dolaşarak kendisine en uygun gruba yerleşmekte, sisteme giren ve çıkan düğümler nedeniyle gruplar sürekli yenilenmektedir.

Bir video akışlandırma oturumunun başarılı sayılabilmesi için video izlemede sürekliliğin sağlanması gerekmektedir. Görevdeş ağlar üzerinde çalışan video akışlandırma sistemlerinde yaşanan ve çözülmesi gereken en önemli sorunlardan birisi, video izleme sürekliliğinin sistemden çıkan düğümler sebebiyle kesintiye uğramasının engellenmesidir (Magharei et al., 2007). Bir düğüm, video verisi gönderdiği bir ya da daha fazla çocuk olduğu halde sistemden habersiz olarak ayrılabilir. Ebeveynini kaybeden bir düğüm ise mümkün olan en kısa zamanda yeni bir düğüme bağlanarak akışlandırma oturumunu sürdürmelidir.

Akışlandırılacak video dosyası sistem üzerinde bulunduktan sonra kaynak düğümden hedef düğüme ulaşırken paket taşımada görev alacak düğümlerin seçimi sistemin tüm performansını etkileyecek önemli bir parametredir. Örneğin uzakta bir düğümden birbirine yakın iki düğüme birbirinden bağımsız oturumlar ile akışlandırma yapmak sistem kaynaklarını etkin bir şekilde kullanılmamasına sebep olur. Bunun yanı sıra, düğümlerin video verisini alarak diğer düğümlere veri gönderirken topolojik konumlarına uygun bir şekilde gönderim ağacına yerleştirilmesi, video verisinin kaynaktan hedefe kısa zamanda iletilmesini sağlar; dolayısıyla bir diğer önemli parametre olan başlangıç bekleme zamanının düşürülmesine yardımcı olur.

Ölçeklenebilir video kodlayıcı/çözümleyici, video verisinin farklı kalitelere kodlanmasını sağlayarak değişen ağ koşullarına ya da farklı tipteki istemcilere uygun kalite gönderimine olanak sağlamaktadır. Görevdeş ağlarda bulunan düğümlerin yükleme ve indirme kapasiteleri çeşitlilik göstermektedir, dolayısıyla ölçeklenebilir video kodlayıcı/çözümleyiciye uygun bir mimari heterojen istemci altyapısına bir paralellik sağlayabilir. Geliştirilen Kapasite farkındalıklı çoklu gönderim ağaç oluşturma algoritması, düğümlerin ağaç üzerinde ölçeklenebilir video kodlayıcı/çözümleyicisine uygun bir şekilde yerleşmesini sağlamaktadır.

Tez kapsamında, video izlemek üzere görevdeş ağ sistemine giren düğümler arasında, herhangi bir videoyu izlemek isteyen düğümlerin bu talebi bildirmesi, videonun aranması, sistemde bulunan kaynak ya da kaynaklar ile istemciler arasında bir akışlandırma yolunun belirlenmesi ve akışlandırma sırasında düğümlerin sistemden ayrılması durumunda akışlandırmanın devam edebilmesine yönelik bir video akışlandırma sistemi geliştirilmiştir. Geliştirilen protokoller, benzetim ortamında test edilmiş ve başarımları raporlanmıştır. Elde edilen sonuçların literatürdeki yerini görebilmek amacı ile, literatürde yer alan benzer

alışmalar ile tasarlanan mimari karşılaştırılmış ve önerilen sistemin elde edilen video kalitesi, sisteme yeniden bağlanma süresi, başlangıç bekleme zamanı gibi parametrelerde daha başarılı sonuçlar elde ettiđi gözlemlenmiştir.

Tez kapsamında yapılan alışmaların katkısı genel olarak aşğıdaki şekilde maddelenebilir:

- Çok katmanlı görevdeş ađ üzerinde çoklu gönderim ađalarının oluşturulması sağlanmıştır. Çokluguönderim ađacının farklı kısımları farklı gruplar ve farklı hiyerarşik katmanlar üzerinde oluşturulmaktadır. Bu yaklaşım ađaçların hem dağıtık hem de ölçeklenebilir şekilde yapılandırılmasını sağlamaktadır.
- Çokluguönderim ađaçlarının kapasite farkındalıklı ve ölçeklenebilir video verisine uygun bir şekilde oluşturulması sağlanmıştır.
- Çokluguönderim ađaçlarında hata dayanıklılıđını sağlamak üzere hem proaktif hem de reaktif bileşenleri olması sebebiyle literatürdeki alışmalardan ayrılan bir yaklaşım öne sürölmüştür. Sunulan yaklaşım, dağıtık bir şekilde katmanlı mimari ile bütünleştirilmiştir.

Tezin bölümleri şu şekilde düzenlenmiştir. Tezin ikinci bölümünde konu ile ilgili kavramların açıklamaları ve ilgili alışmalar yer almaktadır. Tezin üçüncü bölümünde altyapı olarak kullanılan kaplama mimarisi, dördüncü bölümünde ise önerilen çokluguönderim ađaç oluşturma algoritması açıklanmaktadır. Beşinci bölümde, tasarlanan kaplama mimarisi üzerinde hata dayanıklı çokluguönderim ađaç oluşturma yaklaşımına yer verilmiştir. Altıncı bölümde ise, akışlandırma oturumu sürecinde alıştırılan protokolün ayrıntıları yer almaktadır. Yedinci bölümde hata dayanıklılıđı ile ilgili ek alışmalara yer verilmiş; sekizinci bölümde tasarlanan algoritma ve protokollerin benzetim ortamı üzerindeki başarımlar sonuçları raporlanmıştır. Sonuçların yer aldığı dokuzuncu bölümü kaynaklar ve ekler izlemektedir.

2. LİTERATÜR ÖZETİ

Tezin bu bölümünde, görevdeş ağlar üzerinde gerçekleştirilen video akışlandırma sistemleri ve bu çalışmaların altyapı mimarisi, karakteristik özellikleri ve konu ile ilgili problemlere yaklaşım yöntemlerinden söz edilecektir. Bu bölümde yer alan alt başlıklarda, konu ile ilgili kavramlara da açıklık getirilecektir.

2.1 Kaplama Mimarisi

Kaplama mimarisi, bir ağ üzerinde uygulama katmanında çalışan programların, amaçları doğrultusunda, ağ üzerinde yer alan bileşenleri organize ettikleri topoloji biçimidir. Video akışlandırma sistemlerinde, genellikle, kaplama mimarisi, altta yatan ağ topolojisine paralel olarak kurulur. Uygulama katmanında ağ topolojisi bilinemediği için, iki düğüm arasındaki gecikme ölçümleri, zıplama sayısı, bantgenişliği gibi değerler ile topoloji hakkında yeterli oranda bilgi edinilebilir. Video akışlandırma uygulamaları için kurulan kaplama mimarileri genel olarak gecikme tabanlı ve ilgi tabanlı olarak 2'ye ayrılabilir (Magharei et al., 2007).

2.1.1 Gecikme tabanlı kaplama mimarileri

Gecikme tabanlı kaplama mimarilerinin genel amacı, birbirine coğrafi olarak yakın olan düğümler arasında gruptandırma yaparak kaplama mimarisi oluşturmaktır. (Chu et al., 2002) , görevdeş ağlar üzerindeki uygulamalara yönelik genel amaçlı çalışmasında, sistemde bulunan bazı düğümler arasında aralarındaki gecikmeye bağlı olarak bağlantı kurar. Ağ yapısının değişken durumuna yönelik, düğümler arasında kurulan bu bağlantılar, eğer iki düğüm arası gecikme artmış ise koparılabilir, ya da sisteme giren düğümler ile birlikte yeni bağlantılar yapılabilir. Gecikmeye yönelik bu çalışmada Paket Dolaşım Zamanı (RTT) gözönüne alınmıştır. Sözü edilen çalışma, çoklugoönderim sistemleri için kaplama mimarisi kurulumunun, özellikle üyelik yönetimi ve paket iletimi açısından yararlı bir yaklaşım olduğunu göstermektedir. (Chu et al., 2002)'de önerilen topoloji kurulumu, küçük ve orta boyutta çoklugoönderim grupları için geliştirilmiştir; yüksek sayıda düğüm içeren sistemler için getirilecek çözümler açık bırakılmıştır.

(Zhang et al., 2004) tarafından yapılan çalışmada, görevdeş ağlar üzerinde gruptandırma yöntemi, düğümlerin sisteme girişi için farklı bir algoritma önerilmiştir. Kurulan yapıda, sisteme yeni giriş yapan düğüm, önceden bilinen bir

adrese başvurarak belirli grup liderlerinin adresini alır. Bu grup liderleri ile arasında RTT ölçümü yapar ve kendisine en yakın liderin grubuna eğer aralarındaki gecikme belli bir değerin altında ise dahil olur. Eğer ölçüm eşik değerini aşarsa, kendisine en yakın grup liderinin komşularının adreslerini alır ve aynı algoritmayı bu liderler arasında çalıştırır. Önerilen sistemde gecikmeye bağlı gruplar oluşur ancak, sistem belli düğüm sayısını aşan durumlar için ölçeklenebilir değildir.

NICE adı verilen çalışma (Banerjee et al., 2002), kaplama mimarisi olarak gecikme bakımından birbirine yakın olan düğümler ile birlikte, bu düğümlerin oluşturduğu gruplardan hiyerarşik katmanlar oluşturur. Sistemdeki her düğüm, en alt hiyerarşik katmanda bir gruba dahildir, en alt hiyerarşik katmandaki her grupta, grup üyelerinden biri lider seçilerek mantıksal olarak bir üst katmana çıkar. Bu şekilde üst katmana çıkan düğümler, gruplar oluşturur ve grup sayısı arttıkça hiyerarşik katman sayısı artar. Çalışmada, düğümler, sisteme giriş yaptıktan sonra gecikme açısından kendisine en yakın gruba giriş yapmak için hiyerarşik katmanlar arasında dolaşarak her katmanda kendisine en yakın gruba yönelir. Lider seçimlerinde coğrafi olarak merkezdeki düğümün lider olarak seçilmesi önerilmiştir. Bu yaklaşım, hiyerarşik katmanlar arasında dolaşarak kendisine en yakın gruba girmeyi hedefleyen sisteme yeni giren bir düğümün doğru gruplara yönlenmesine yardımcı olmaktadır. Sistemdeki düğümlerin gruplandırılarak bu gruplar yardımı ile hiyerarşik katmanlar oluşturma yaklaşımı, görevdeş ağlar üzerinde çalışan uygulamaların karşılaştığı en büyük problemlerden biri olan ölçeklenebilirlik konusunu çözmektedir.

NICE çalışmasında (Banerjee et al., 2002) önerilen gruplandırma ve hiyerarşik katmanlara yönelik bir başka çalışmaya örnek olarak ZIGZAG isimli çalışma (Tran et al., 2004) verilebilir. Sistemin kaplama mimarisi ile ilgili en zayıf noktası, en üst hiyerarşik katmandaki liderin aynı zamanda video akışlandıran sunucu görevinde bulunuyor olmasıdır. Hem sisteme giren düğümlerin bu düğüme başvurması hem de bu düğümün video akışlandırmada kök görevini görüyor olması, yoğun giriş çıkışların beklendiği görevdeş ağlarda bir darboğaz oluşturabilir.

Hiyerarşik katmanlar kullanan bir başka çalışma ise NEMO (Birrer and Bustamente, 2005) ismiyle adlandırılmıştır. Çalışmada, grupların ve hiyerarşik katmanların oluşması NICE çalışmasıyla paralellik göstermektedir. Grupların birleşme ve ayrılma gibi bakımları için olasılık tabanlı bir yaklaşım önerilmiştir.

Bu yaklaşıma göre, her düğüm ayrılmasında ya da sisteme her düğüm girişinde gruplar üzerinde bakım yapılmaz, onun yerine, belirli aralıklarla gruplar içerisindeki eleman sayısına bakılarak grup birleştirme ayrıştırma işlemleri yapılır. Böylece düğüm giriş çıkışının sıkça yaşandığı görevdeş ağlarda, her seferinde grup bakımı yapılması ekyükü hafifletilmiştir.

(Chakareski and Frossard, 2009) çalışmasında gecikmeye bağlı olarak gruplandırılan düğümlerin, video akışlandırma sırasında gösterim zamanı birbirine yakın video paketlerini alma açısından avantajlı olduğunu göstermiştir. Çalışmada, gecikmeye bağlı gruplar, sunucudan uzaklıklarına göre gruplandırılmıştır. Ancak bu çalışmada hiyerarşik katmanların kullanılmaması, büyük gruplar için ölçeklenebilirlik sorununu beraberinde getirebilir.

2.1.2 İlgi tabanlı kaplama mimarileri

İlgi tabanlı kaplama mimarilerinin amacı, birbirine yakın seçimlerde bulunabilecek düğümlerin aynı gruba konmasını sağlamaktır. Bu tip kaplama mimarilerinde grupları oluşturan düğümler, birbirine ilgi açısından yakın olan düğümlerdir.

Tribler adı verilen çalışma (Pouwelse et al., 2007), arkadaşlık ilişkisi içinde bulunan düğümleri benzer ilgi alanında değerlendirerek kaplama mimarisi üzerinde ilişkilendirmiştir. Yapılan deneylerde video akışlandırma hızının gecikmeden dolayı yavaşlayabildiği raporlanmıştır, bu durum kurulan altyapıda gecikmenin gözönünde bulundurulmamasından kaynaklanmaktadır.

(Bulut et al., 2010) çalışmasında bir görevdeş ağ üzerinde bulunan düğümler ilgilerine göre JSD (Jensen Shannon Divergence) metriğine bağlı olarak oluşturulan gruplandırılmış ve hiyerarşik katmanlar oluşturulmuştur. JSD metriği, düğümler ile gruplar arasında ilgi benzerliğini ölçmek üzere kullanılmıştır. Önerilen ilgi tabanlı gruplandırma mimarisi, video akışlandırma uygulamaları için bir altyapı oluşturmaktadır.

2.2 Akışlandırma Yöntemleri

Video verisinin iletimi için kurulan akışlandırma yolu, video paketlerinin kaynaktan herbir düğüme gidene kadar aldığı yoldur. Çokluguönderim sistemlerinde, sunucunun yükünü hafifletmek amacıyla, sunucudan paket alan

düğüm bir ya da daha fazla düğüme bu paketleri iletir. IP tarafından sağlanan çoklugoönderim ölçeklenebilirlik, ağ yönetimindeki eksiklikler, hata, akış ve sıkışıklık kontrolü gibi konular açısından yetersiz kalmaktadır (Liu et al., 2008). Bu durum, çoklugoönderim fonksiyonunun uygulama katmanında gerçekleşmesi ihtiyacını doğurmaktadır.

Akışlandırma yolunun kurulmasında düğümlerin birbirine olan uzaklıkları, başlangıç bekleme değerini etkileyeceği için önemli bir parametredir. Bunun yanısıra düğümler arası bantgenişliği değerleri, iletilen videonun alıcılar tarafındaki kalitesini etkileyen en önemli ölçüttür. Literatürdeki çalışmalar değerlendirildiğinde, temel olarak bu parametreler esas alınarak akışlandırma yolu oluşturulmuştur (Chu et al., 2002), (Tran et al., 2004), (Jeon et al., 2008), (Chakareski and Frossard, 2009), (Ding et al., 2010).

2.2.1 Örgü tabanlı akışlandırma yöntemi

Örgü tabanlı akışlandırma yönteminde, video verisi düğümler arasında parçalar halinde gönderilir. Ebeveyn çocuk ilişkilerinde bir sınırlama yoktur; bir düğümün birden fazla ebeveyni ve birden fazla çocuğu olabilir. Bunun yanısıra, iki düğüm birbirinin hem ebeveyni, hem çocuğu olabilir. Örgü tabanlı akışlandırma yönteminin altında yatan temel prensip, video dosyasının uygun parçalara bölünmesi ve bu parçaların düğümler arasındaki dağıtımının modellenmesidir.

(Xie et al., 2007), örgü tabanlı video akışlandırma sistemlerine örnek bir çalışma önermiştir. Önerilen ve Coolstreaming adı verilen sisteme göre, sisteme giren düğümler, sistemde bulunan düğümleri bir dedikodu protokolü aracılığı ile bulur ve aralarındaki bantgenişliğine bakarak bazı düğümleri ortak olarak listesine ekler. Ortak düğümlerin bulunmasında, sistemde yer alan sunucular düğümlere yardımcı olmaktadır. Düğümler, sisteme girdikten sonra da ortak arayışına sürekli olarak devam eder. (Zhang et al., 2005b) çalışmasında bir ortak seçim algoritması önerilmiştir. Sistem, ortakların birbirleri ile video parçalarını paylaşması ile video akışlandırma oturumunu gerçekler. İki ortak arasında uygun video parçalarının değişimi için, tampon haritası adı verilen mekanizma kullanılmaktadır. Tampon haritası üzerinde her düğüm, elinde bulunan ve bulunmayan video parçalarının bir desenini çıkararak ortaklarına gönderir, benzer şekilde ortaklarının tampon haritalarını alır; ortaklarında bulunan ve kendisinde bulunmayan video parçalarını talep eder. Bu sistemdeki en büyük sorun gecikme – ekyük ödünleşimidir (Hei et

al., 2008). Bir düğüm tampon haritalarını ne kadar sık değiştirirse gecikme azalır ancak sisteme binen ekyük artar.

(Magharei and Rejaie, 2009) çalışmasında PRIME adı verilen örgü tabanlı bir video akışlandırma sistemi; Coolstreaming sisteminin, örgü tabanlı video akışlandırma sistemlerinin temel zorlukları ile başa çıkma presipleri konusundaki eksikliğine bağlı olarak önerilmiştir. (Magharei and Rejaie, 2006) çalışmasında, örgü tabanlı yaklaşımın temel prensibi olarak, bir düğüm birden fazla düğümle bağlantı kurar ve her düğümün birden çok ebeveyni ve çocuğu olabilir. Sistemde, kendisine çocuk bağlı olan düğümler, yeni aldıkları her video parçasını tüm çocuklarına iletirler. Video parçaları, çoklu tanımlama kodlamaya (MDC), (Goyal, 2001) uygun bir şekilde üretilmiştir. Benzer şekilde, (Magnetto et al., 2010) çalışmasında da örgü tabanlı mimari üzerinde çoklu tanımlama ile kodlanmış video verisi gönderilmiştir. Çoklu tanımlama kodlama yöntemi hakkında bilgi ilerleyen bölümlerde verilecektir.

Akademik anlamda önerilen örgü tabanlı benzer çalışmalar ya da örgü tabanlı çalışmaları analiz eden yayınlar arasında (Vlavianos et al., 2006), (Sentinelli et al., 2007), (Tang et al., 2007), (Hei et al., 2008) ve (NextShare, 2011) sayılabilir. Örgü tabanlı ticari video akışlandırma çalışmalarına ise örnek olarak (PPLive, 2010) ve (PPStream, 2010) verilebilir. Örgü tabanlı yaklaşımlar gecikme – ekyük ödünleşimi yaşamaktadır. Eğer düğümler aldığı her veri parçası için mesaj gönderirse mesaj karmaşıklığı artmaktadır. Öte yandan, periyodik bilgilendirmeler mesaj karmaşıklığını düşürür ancak gecikmeyi artırır (Zhang et al., 2005a), (Venkataraman et al., 2006), (Liu et al., 2007). Bu dezavantaja ek olarak örgü tabanlı sistemlerin kullanıcıları, birkaç saniyeden birkaç dakikaya kadar uzayabilen uzun başlangıç bekleme zamanı ve ebeveyn – çocuk çiftleri arasında oluşan uzun oynatım bekleme zamanından negatif etkilenmektedir (Hei et al., 2007), (Chen et al., 2009).

2.2.2 Ağaç tabanlı akışlandırma yöntemi

Ağaç tabanlı akışlandırma yöntemi, bir düğüme bağlı olabilecek ebeveyn sayısını bir ile sınırlar; temel prensibi, her düğümün kendisine gelen video paketlerini eğer var ise, çocuk ya da çocuklarına iletmesidir. Tanımlı paket iletme yolları sebebiyle yapısal bir paket akışlandırma yöntemi olan ağaçların video akışlandırma sistemleri için kullanımı literatürdeki birçok çalışma tarafından önerilmiştir (Banerjee et al., 2002), (Tran et al., 2004), (Birrer and Bustamente,

2005), (Liu et al., 2007), (Jin et al., 2007), (Fei and Yang, 2007), (Jeon et al., 2008). Çokluguönderim ağaçları kullanan sistemlerin en büyük avantajı, düğümler arasında tampon haritaları gibi karmaşık yapıların kullanılmasını gerektirmemesidir. Düğümler sadece alınan paketi çocuklarına iletmekle yükümlüdür. Ağaçların en büyük dezavantajlarından biri, ağacın köküne yakın düğümlerin düşük yükleme kapasitesine sahip olması ile, ağacın tümünün düşük kalitede veri almasına sebep olabilmesidir. Bir diğer dezavantajı ise sistemden çıkan düğümler sebebiyle bu düğümlerin altağacında yer alan düğümlerde video akışlandırmanın kesintiye uğraması, dolayısıyla çokluguönderim ağaçlarında hata dayanıklılığının zayıf olmasıdır (Liu et al., 2007).

Ağaç tabanlı video akışlandırma sistemlerinde kurulacak ağaçlar için bazı çalışmalar, gecikme tabanlı kaplama mimarilerinden yararlanmıştır. Kurulan kaplama mimarisi yardımı ile altta yatan ağ yapısına yönelik bilgi edinen sistemler, bu bilgi yardımı ile çokluguönderim ağaçları oluşturur. (Chu et al., 2002) çalışmasında, altta yatan mimariden yararlanarak, düğümlerin birbirlerini bulmasını sağlar. Ağaç oluşumu için düğümler arasındaki ikili bantgenişliklerini değerlendirir ve yüksek bantgenişliğine sahip ikili düğümleri birbirine bağlayarak ağaç oluşturur. Eğer bantgenişliği değerleri bazı ikililer arasında eşit çıkarsa, düğümler arasındaki gecikme değerlendirilir ve en yakın düğümler arasında ebeveyn – çocuk ilişkisi kurulur. Önerilen sistem, görevdeş ağlar üzerinde çalışan genel amaçlı bir sistem olarak önerilmiş, video akışlandırma çalışmalarında da kullanılabileceği belirtilmiştir.

(Banerjee et al., 2002) çalışması (NICE), benzer şekilde, altta yatan gecikme mimarisine bağlı kalarak çokluguönderim ağacı oluşturur. Hiyerarşik katmanlar halinde gecikmeye bağlı gruplandırmaya sahip sistem, grup liderlerini kendi grubuna video verisi taşımakla görevlendirir. Önerilen sistem, ölçeklenebilirlik sorununu çözmüş olmasına rağmen, düğümlerin kapasitelerini dikkate almadığı için, ağaçlarda düşük kaliteli video iletimine sebep olabilir. Önerilen sistem mimarisinde grup içindeki eleman sayısı k ile $3k-1$ arasında değişmektedir. Grup liderine grup elemanlarının tümü çocuk olarak bağlabileceği için, çocuk sayısının k ile $3k-1$ arasında olduğu görülmektedir, bu durum lider ile çocuklar arasında içerik darboğazı oluşmasına neden olabilir.

(Birrer and Bustamente, 2005) çalışması, ağaç oluşumunda NICE çalışmasını temel almaktadır. Dolayısıyla bu çalışma ile benzer kapasite problemleri, bu sistemde de yaşanabilir.

ZIGZAG çalışması (Tran et al., 2004), gecikmeye bağlı oluşturulan gruplar ve grupların oluşturduğu hiyerarşik katmanlar üzerinde çokluguönderim ağaçları oluşturur. Bu sistemin, daha önce önerilen yöntemlerden temel farkı, grupları oluşturan ve bakımını yapan liderleri, gruplara video verisini ileten liderlerden ayırmasıdır. Sadece en üst katmanda yer alan lider, hem grup bakımından sorumlu hem de kaynak görevindedir. Başka bir deyişle, sistemdeki kaynak, aynı zamanda sisteme giriş yapan tüm düğümlerin başvuracağı en üst seviye grup lideridir. Grup bakımını yapan liderler ve video verisini ileten liderler ile çocukları arasındaki ilişkileri ve bağlantıları belirlemek üzere “bağlantı kuralları” adı verilen bir dizi kural belirlenmiştir. Bu kurallar ile, grup içindeki eleman sayısı k ile $3k$ arasında iken, ağaç üzerinde yer alan bir düğüme bağlanacak düğüm sayısının en fazla $6k-3$ olabileceği ispatlanmıştır. k değerinin 2’ye eşit olduğu durumda bile, bir düğüme bağlanabilecek düğüm sayısının 9 olabilmesi, sistemdeki düğümler ile ilgili kapasite problemleri yaratabilir.

(Padmanabhan et al., 2003) tarafından önerilen sistemde birden fazla çokluguönderim ağacı oluşturulur ve her ağaç üzerinden farklı video parçaları gönderilir. Çokluguönderim ağaçları üzerinde ebeveyn – çocuk ilişkileri altta yatan ağ mimarisine uygun bir şekilde oluşturulmaktadır. Bunun için düğümlerin birbirleri arasındaki gecikme zamanı gözönünde tutulur. Gönderilen video parçaları çoklu tanımlama kodlama yöntemi ile oluşturulmuştur.

Çoklu tanımlama kodlama yöntemi kullanan başka bir çalışmaya örnek olarak (Baccichet et al., 2007) verilebilir. Herbir tanım, bir önceki çalışmaya benzer şekilde farklı ağaçlar üzerinden gönderilir. Sisteme giren düğümler, farklı ağaçlar üzerinde bir ebeveyn bulmak üzere ağacın kökü görevindeki düğüme mesaj atarak aday ebeveyn listesi alır. Aday ebeveynler ile iletişime geçen düğüm, hem gecikmeyi gözönünde bulundurarak, hem de aday ebeveynlerin kapasitesini değerlendirerek kendisine en uygun adayı ebeveyn olarak seçer. Çoklu tanımlama kodlama yöntemi, özel bir kodlama gerektirdiği için her sistemde kullanılmayabilir; sözü edilen çoklu tanımlama kodlama kullanan sistemler, bu açıdan tercih edilmeyebilir.

Video akışlandırma sistemi için çokluguönderim ağacı kullanan yöntemler arasında (Yang and Fei, 2004) çalışması verilebilir. Bu çalışmada ağaçların oluşturulmasında temel olarak düğüm kapasiteleri ve düğümler arası gecikmeler dikkate alınmıştır. Video akışlandırma amaçlı ağaç kurulumu için önerilen algoritma, (Narula and Ho, 1980) çalışmasında verilen derece kısıtlı minimum

kapsayan ağaç algoritmasıdır. (Yang and Fei, 2004) çalışmasında oluşan çokluguönderim ağacının uzunluğu dikkate alınmamıştır, dolayısıyla oluşan ağaçlarda zıplama sayısı çok fazla olmakta ve bu da başlangıç bekleme zamanı ve paket kayıp oranını arttırmaktadır.

Video akışlandırma sistemleri için önerilen ağaç kurulumunda düğümlerin kapasitelerini gözönünde bulunduran çalışmalar arasında (Kusumoto et al., 2005) verilebilir. Çalışmada, sisteme giren düğüm kaynak düğüme başvurur, eğer kaynak düğümün yeterli kapasitesi var ise, düğümü çocuk olarak kabul eder; yoksa, kendi çocuklarına yönlendirir. Bu şekilde ağacın yapraklarına doğru hareket eden düğüm, uygun bir ebeveyn bulduğunda ağaca yerleşir. (Okada et al., 2005) ağaçları benzer şekilde oluşturur ancak sisteme yeni giriş yapan düğüm, kendi düğüm kapasitesini kaynak düğüme bildirir ve birden fazla sayıda ebeveyn aday adresi alır. (Jeon et al., 2008) çalışmasında da sözü edilen çalışmalara benzer şekilde, sisteme yeni giriş yapan düğüm kaynakta belli ebeveyn adresleri olarak derecesi uygun olan ve kaynak – aday ebeveyn arası gecikme ile aday ebeveyn – kendisi arasında gecikme toplamı minimum olan aday düğümü ebeveyn olarak seçer. Ancak çalışmada, düğümlerin kapasitesine bağlı şekilde bir yedek kapasite ayırma yaklaşımı bulunmamaktadır. Bu durum ağaç uzunluğunun kontrol edilememesine sebep olduğu gibi hata dayanıklılığı konusunda da dezavantaj sağlamaktadır.

2.2.3 Karma tabanlı akışlandırma yöntemi

Örgü tabanlı video akışlandırma sistemlerinin gecikme – ekyük ödünleşimi; ağaç tabanlı sistemleri bantgenişliğini etkin değerlendirme problemleri sebebiyle; her iki yöntemi de kullanan az sayıda çalışma ortaya atılmıştır. (Venkataraman et al., 2006) çalışmasında video verisi farklı bölümlere ayrılarak farklı şekilde iletilir ancak bu iletimin farklı ağaçlardan yapılması zorunluluğu yoktur. (Wang et al., 2010) çalışmasında önerilen yöntemde ise sistemde bulunan belli sayıdaki düğümler bir ağacın temel düğümlerini oluşturur ve bu düğümler arasında video paketlerini alma ve iletme işlemi klasik ağaç üzerinde video akışlandırma sistemlerine benzer biçimde yapılır. Bu düğümler ile bu düğümler dışında kalan düğümler arasında ise örgü tabanlı yaklaşım ile veri iletişimi yapılır.

(Liu et al., 2008) çalışmasında akışlandırma yöntemleri ile ilgili ayrıntılı bilgilere ve bu yöntemlerin farklı açılardan karşılaştırılmalarına ulaşılabilir. (Vénot and Yan, 2007) çalışmasında da video akışlandırma sistemleri, özellikleri,

yapılan testler ve yöntemlerinin karşılaştırılmasına yönelik bir araştırma yapılmıştır.

2.3 Çoklוגönderim Ağaç Oluşturma Algoritmaları

Bu bölümde, literatürde önerilen genel amaçlı çoklוגönderim ağaç oluşturma algoritmalarından kısaca söz edilecektir.

Chang-Roberts çoklוגönderim ağaç oluşturma algoritması (Chang, 1982), bu ağacı oluşturacak düğümler arasında gecikmeyi temel alarak bir ağaç yapısı oluşturulmasını sağlar. Chang-Roberts algoritmasının en önemli avantajı, düğümlerin ağaç oluşturmak için birbirlerine olan uzaklıklarını önceden bilmelerine gerek duymamalarıdır. Algoritma, önceden belirlenen bir düğüm tarafından çalıştırılmaya başlatılır ve bu düğüm, ağaçta yer alacak tüm düğümlere mesaj gönderir. Bir düğüm, ilk kez mesaj aldığı anda, mesajı gönderen kişiyi ebeveyni olarak belirler ve bu düğüm hariç diğer tüm düğümlere mesaj gönderir. Algoritmanın mesaj karmaşıklığı, n adet düğümden oluşan bir sistem için, mantıksal mimaride tüm düğümlerin birbirleri ile komşu olduğu göz önünde bulundurulduğunda $O(n^2)$ dir. Dağıtık ve minimum kapsayan ağaç oluşturan Chang-Roberts algoritması, düğümlerin derecelerini ve bantgenişliklerini dikkate almadığı için video akışlandırma çalışmalarında uygulanması tercih edilmemiştir.

Dağıtık çoklוגönderim oluşturma algoritmaları arasında en çok öne çıkmış algoritmalarından biri olan dağıtık BFS ve DFS algoritmaları (Ghosh, 2006; Kleinberg and Tardos, 2006), Chang Roberts'a benzer şekilde bir mesaj karmaşıklığına sahiptir, ancak benzer bir mantıkla ağaç yapılandırıldığı için Chang Roberts ile benzer dezavantajlara sahip olduğu öngörülmektedir.

Ağaç oluşturma aşamasında bantgenişliğini de göz önünde bulundurabilmek için bir çizge üzerindeki düğümler arasında bulunan ayrıtların ağırlıklarına bağlı olarak minimum kapsayan ağaç çıkaran GHS algoritması (Gallager et al., 1983) incelenmiştir. GHS algoritması, Prim's (Prim, 1957) ve Kruskal (Kruskal, 1956) algoritmalarına benzer bir şekilde çalışır; bu iki algoritmadan ayrılan en önemli özelliği dağıtık olarak çalıştırılmasıdır. Algoritmanın mesaj karmaşıklığı, mantıksal ağ mimarisinde her düğüm çifti arasında bir kenar bulunduğu düşünüldüğü için $O(n^2)$ dir. Ancak GHS algoritmasında bir düğüme bağlanacak düğümlerin sayısına müdahale edilememektedir, bu durumda diğer algoritmalara benzer şekilde çocuk sayısı kontrolü sağlanamaz. Ayrıca GHS algoritmasının

video akışlandırma amacı ile kullanımındaki bir başka sorun, bu algoritmanın birden fazla düğüm tarafından aynı düğüm grubu üzerinde çalıştırılmaya uygun olmamasıdır.

Düğümlerin derecelerini gözönünde bulundurarak minimum kapsama ağacı oluşturan bir algoritma (Narula and Ho, 1980) tarafından önerilmiştir. Algoritmanın temeli Prim's algoritmasına dayanmaktadır, dolayısıyla bu algoritmanın karmaşıklığı Prim's in karmaşıklığına benzer şekilde n adet düğüm için (mantıksal ağ mimarisinde her düğüm çifti arasında bir kenar bulunduğu düşünülür) $O(n^2 + n \log n)$ 'dir. Bu algoritma, düğümün kapasitelerini dikkate aldığı için, literatürde önerilen genel amaçlı çokluguönderim ağaçları arasında video akışlandırma çalışmasında kullanılan tek algoritmadır (Yang and Fei, 2004).

2.4 Hata Dayanıklılığı

Görevdeş ağlar üzerinde çalışan video akışlandırma sistemlerinin karşılaştığı en büyük sorunlardan biri sisteme giren düğümlerin habersiz olarak yüksek oranda çıkış hızıdır. Habersiz çıkışlarda, herhangi bir düğüm, eğer bir grup içerisinde ise grup liderine, ebeveyni var ise ebeveynine haber vermeden sistemden ayrılmakta; bu hatayı tespit etmek ve düzeltmek için gereken zaman bazı düğümlerin kesintisiz video izlemesini engellemektedir. Video akışlandırma sırasında kesinti olmasının sebebi, düğümlerin video paketlerini aldığı ebeveyninin sistemden çıkması ve tamponda bulunan veriyi tüketmeden yeni bir bağlantı kuramamasıdır. Bunun dışında, ebeveyni sistemden ayrılmadığı halde bağlantı kalitesinde sorun yaşayan düğümlerin izlediği video kalitesi düşebilmektedir.

Örgü tabanlı video akışlandırma sistemlerinde, sistemde bulunan düğümlerin birden fazla sayıda ortağı, dolayısıyla birden fazla sayıda ebeveyni vardır. Bu ebeveynlerden biri sistemden ayrıldığında, sürekli olarak yeni ortak arayışında bulunan düğümler yeni ebeveynler bulabilir. Arada kurulan bağlantıların kötüleşmesi durumunda ise bu bağlantılar koparılır. Dolayısıyla, örgü tabanlı akışlandırma yöntemlerinde hata dayanıklılığı oldukça yüksektir.

Ağaç tabanlı video akışlandırma sistemlerinin en büyük dezavantajlarından biri, hataya karşı daha dayanıksız olmasıdır (Magharei et al., 2007). Hataya karşı dayanıksızlık, tek bir ebeveyni bulunan düğümlerin ebeveyninin sistemden ayrıldığını farketmesi ve yeni bir ebeveyn bulması için gereken sürenin genellikle

yüksek olmasından kaynaklanmaktadır. Ağaç tabanlı akışlandırma yolu üzerindeki hata dayanıklılığını arttırmak için literatürde çeşitli çalışmalar yapılmıştır. Tezin bu bölümünde, ağaçlar üzerindeki hata dayanıklılığı üzerine geliştirilen yöntemler, özelliklerine göre sınıflandırılarak incelenecektir.

2.4.1 Reaktif yöntemler

Reaktif yöntemler, düğümlerin, ebeveyninde hata oluşuktan ve bunu tespit ettikten sonra yeni bir bağlantı kurma amacıyla hareket etmesini içeren yöntemlerdir. Ebeveyninde ya da ebeveyni ile arasındaki bağlantıda hata oluştuğunu tespit eden düğümler ebeveyninin ebeveynine ya da ağacın kökü konumundaki düğüme bağlanabilirler. Bu düğümlerin kapasitesinin dolu olması durumunda başvuruda bulunan düğüm, başka düğümlere yönlendirilebilir. Ancak bu yöntemlerin kazanım değerlerinin düşük olduğu (Fei and Yang, 2007) çalışmasında raporlanmıştır.

(Banerjee et al., 2002) çalışmasında hiyerarşik katmanları oluşturan gruplar içerisinde bulunan düğümler, grup liderlerinin video verisi taşıması ile video izleyebilmektedir. Eğer grup lideri sistemden ayrılırsa, bu durum tespit edildikten sonra yeni bir lider seçimi yapılır ve yeni seçilen lider, gruba video taşımaya devam eder. Ağacın yeniden yapılandırma zamanı, hata tespiti ve yeni lider seçimi için gereken sürenin toplamından oluşur.

Hiyerarşik katmanlar üzerinde çalışan (Tran et al., 2004) tarafından önerilen sistem, grup bakımı ve video taşınması işlevlerini farklı düğümlere verdiği için, gruba video taşıyan düğümün sistemden ayrılması grup yapısını bozmaz. Gruba akışlandırma ile görevli düğümün sistemden ayrıldığını tespit eden grup lideri, bu düğüm yerine yeni bir düğümü görevlendirir. Bu seçimin yeni bir grup lideri seçimine göre daha az ekyük getirdiği aynı çalışmada vurgulanmıştır. Ağacın yeniden yapılandırma zamanı, gruba video taşıyan düğümün ayrılmasının tespiti ile yeni düğümün seçilmesi için gereken sürenin toplamına eşittir.

Literatürdeki reaktif yöntemler incelendiğinde, video akışlandırmanın kesintisiz olarak devam edebilmesi için düğümlerdeki tampon değerlerinin yüksek olması gerekmektedir. Bu tip yöntemlerde ağacın yeniden oluşumu için gereken sürenin yüksek olduğu (Fei and Yang, 2007) çalışmasında belirtilmiştir.

2.4.2 Proaktif yöntemler

Proaktif yöntemler, düğümlerin ebeveyninde henüz hata oluşmadan aday bir ebeveyn belirlenmesini içeren yöntemlerdir. Bu durumda, hata oluştuğu zaman ağacın yeniden yapılandırılması için gereken süre, sadece sistemden ayrılan düğümlerin tespiti ve yeni bağlantının kurulması için gereken sürenin toplamına eşittir. Önceden belirli bir düğüm ile bağlantı kurma, yeni bir düğümün seçim zamanından daha kısa sürecektir.

(Padmanabhan et al., 2003) ve (Castro et al., 2003) çalışmasında oluşturulan birden fazla ağaç üzerinden videonun farklı parçaları akışlandırılmaktadır. Çoklu tanımlama kodlama ile parçalara ayrılan video verisinin her parçası farklı ağaç üzerinden düğümlere ulaşmaktadır. Sistemden bir düğüm ayrıldığında, bu düğümün çocukları, sadece bu ebeveyni ile bağlantı kurduğu ağaçtan veri almaz, diğer ağaçlar üzerinden veri alımına devam eder. Birden fazla ebeveynin olduğu bu sistemde ebeveynlerde oluşan hatalardan etkilenen düğümler video kalitesinde düşüş yaşar ancak bağlantı kesilmez.

Çoklu tanımlama yöntemi gibi özel bir kodlama yöntemi kullanmayan sistemler için geliştirilen proaktif yöntemlerden biri ağaç – içi fazlalık olarak adlandırılmıştır. (Birrer and Bustamente, 2005) tarafından önerilen yaklaşımda, hiyerarşik grup liderleri, kaynak üzerinden video verisini aldıktan sonra yardımcı liderler yardımı ile bu veriyi grup içindeki tüm düğümlere ve bir üst katmandaki liderlere gönderir. Yardımcı liderler henüz ağaç oluşum aşamasında belirlenmektedir. Eğer lider sistemden çıkarsa, yardımcı lider, liderin gönderim kümesine video verisini iletmeye devam eder. Ancak daha önce sözü edildiği gibi, bu yaklaşımda düğümlere, örneğin lidere bağlanabilecek çocuk sayısına bir sınırlama getirilmemiştir. Özellikle fazla sayıda hiyerarşik katmanın bulunduğu bir sistemde, lider görevini üstlenen düğümlere fazla sayıda çocuk bağlanabilir. Bu durumda bu yaklaşımın ölçeklenebilir olmasından söz edilemez.

Akışlandırma sırasında fazla sayıda paket gönderimine dayalı bir yaklaşım ise çapraz – bağlantı yaklaşımı olarak adlandırılmıştır (Banerjee et al., 2006). Bu yaklaşıma göre, ağaç üzerindeki düğümler belli bir olasılıkla yine aynı ağaç üzerinde bir başka düğüme paket gönderir. Eğer bir düğümün ebeveyni sistemden ayrılırsa, yeni bir ebeveyn bulana kadar yardımcı paketler aracılığı ile video verisi almaya devam eder. Bu yaklaşımda rastgele paket gönderme olasılığı yükseldikçe sistemin sorunsuz akışlandırma yapma olasılığı artar ancak fazlalık paketler henüz

hata olmadığı halde gönderildiği için sisteme bir ekyük getirir. Eğer bu ekyük azaltma yoluna gidilirse, fazladan paket alma olasılığı düşen düğümler, hata durumunda akışlandırma oturumuna devam edemez. (Birrer and Bustamante, 2007) çalışmasında ekyük açısından bu yaklaşımın, ağaç – içi fazlalık yöntemine göre dezavantajlı olduğu belirtilmiştir.

Proaktif yöntemeye dayalı hata dayanıklılığı sağlama kapsamında düğümler için yedek ebeveyn belirleme çalışmaları yapılmıştır. (Yang and Fei, 2004) çalışmasında ortaya atılan bu kavram, her düğüm için henüz ebeveyninde bir hata oluşmadan bir yedek belirleme esasına dayanır. Bu çalışmada, ağaç üzerinde bulunan her düğüm, kendisinde hata oluşması durumunda herbir çocuğu için yedek ebeveyn bulur. Burada kural, oluşacak yeni ağacın da minimum kapsayan ağaç olmasıdır. Algoritma sonucu belirlenen yedekler, çocuklara bildirilir. Ebeveynde hata oluşması durumunda, düğüm yedek ebeveynine bağlanır ve tekrar kendisine bir yedek ebeveyn belirlenir. Düğümler için belirlenen yedekler, düğümü ağaçta bulunduğu konumdan daha yukarıya ya da aşağıya taşıyabilir.

Yedek ebeveyn kullanan bir başka çalışma (Kusumoto et al., 2005) tarafından önerilen sistemdir. Bu çalışmada, ağaç üzerinde yer alan her düğümün bir hücresi yedek olarak ayrılarak bu hücreye çocuk bağlanmamaktadır. Ancak ağaç üzerinde hata oluşmaya başladıkça, ebeveyni ile bağlantısı kopan düğümler, diğer düğümler tarafından ayrılan bu boş hücrelere başvurur. Düğümlerin yedek ebeveynleri kök konumundaki düğüme uzaklıklarına göre ağacın kökü tarafından belirlenir.

(Jeon et al., 2008) tarafından önerilen çalışmada ağaç kurulum aşamasında her düğüm, ağacın köküne ayırdığı yedek hücre sayısını bildirir. Düğümler ağaca yerleştikten sonra, kök ile iletişime geçerek yedek hücre bilgilerini alır. Aldıkları düğüm listesi ile aralarında gecikmeyi ölçen düğümler, kendisine en yakın düğümün yedek hücrelerini ayırırlar. Bu şekilde yedek hücreleri için birer yedek çocuk belirlenen düğümler, yedek hücre sayısı dolduğunda, ağaç köküne durumu bildirerek yedek ebeveynler listesinden çıkar. Akışlandırma oturumu boyunca değişen ağaç üzerindeki düğümler, aynı algoritmayı, yedek ebeveynlerine bağlandıklarında yeniden çalıştırır. Bu çalışmada önerilen yeniden ağaç kurulumunun her düğümün derecesi (bir düğüme bağlanabilecek olan düğüm sayısı) eşit olduğu durumda (Yang and Fei, 2004) ve (Kusumoto et al., 2005) çalışmalarına kazanım açısından üstünlüğü belirtilmiştir.

2.5 Video Kodlama Teknikleri

Video akışlandırma sistemlerinde ölçeklenebilir, katmansız ya da çoklu tanımlama yöntemi ile video kodlayan sistemler bulunmaktadır. Katmansız video kodlayıcı kullanan sistemler daha çok tercih edilmekle birlikte, video verisinin anlamlı parçalara ayrılmaması nedeniyle ölçeklenebilir kodlama ve çoklu tanımlama kodlama yöntemlerinin düğümlere göre kalite uyarlama ya da hata dayanıklılığını artırma gibi birtakım avantajları bulunmaktadır. Bu bölümde, ölçeklenebilir kodlama ve çoklu tanımlama kodlama yöntemleri tanıtılacaktır.

2.5.1 Ölçeklenebilir kodlama

Ölçeklenebilir kodlayıcılarda bir taban ve bir ya da daha fazla sayıda genişleme katmanı bulunur. Genişleme katmanları zamansal, uzaysal ya da SNR ölçeklendirme ile elde edilebilir. Taban katman bilgisini elde eden düğüm, videoyu kodlanan en düşük kalitede izleyebilir, genişleme katmanlarına ait veriyi elde ettikçe izlediği kalite videosu artış gösterir. Ancak taban verisi bulunmadan genişleme katmanları verisi çözümlenemez, dolayısıyla özellikle taban katmanda oluşabilecek kayıplar kesintisiz video izlenmesini zorlaştırabilir. Ölçeklenebilir kodlayıcılar arasında geliştirilen son standard H.264/SVC video kodlayıcısıdır (Schwarz et al., 2007).

Ölçeklenebilir video kodlama sistemini kullanan çalışmalar arasından (Rejaie and Ortega, 2003) görevdeş ağlar üzerinde uyarlamalı bir sistem gerçekleştirmiş; (Ding et al., 2010) örgü tabanlı video akışlandırma sistemlerinde ölçeklenebilir kodlayıcı için tampon kullanım algoritması önermiştir.

2.5.2 Çoklu tanımlama kodlama

Çoklu tanımlama kodlama, video verisini $S > 1$ olmak üzere S adet tamamen kendi başına çözümlenebilen farklı parçaya ya da tanımlamaya ayıran kodlama tekniğidir. Alıcı düğüm, kodlanan video parçalarından ne kadar fazla sayıda alırsa, görüntü kalitesi o kadar artar. Ölçeklenebilir kodlamadan farkı, ölçeklenebilir kodlamada üst kalitede video izleyebilmek için taban ve uygun sırada genişleme katmanı alınmalıdır; oysa çoklu tanımlama kodlama ile birbiri ile arasında bağlantı bulunmayan video parçaları alındığında çözümlenebilir yapılabılır. Çoklu tanımlama kodlama bu esnekliğine rağmen ölçeklenebilir kodlamaya göre ekyük getirmektedir. Çoklu tanımlama kodlamaya örnek olarak bir video verisinin tek

numaralı ve çift numaralı çerçevelerinin 2 farklı gruba ayrılması verilebilir. 30 fps ile kodlanan bir video verisi için bu yöntem düşünüldüğünde, 15 fps'lik herhangi bir parçayı alan düğümler daha düşük zamansal çözünürlük ile videoyu izleyebilir. Farklı çoklu tanımlama kodlama yöntemleri (Goyal, 2001) çalışmasında bulunabilir.

Bu yöntem ile yapılan video akışlandırma çalışmalarında, birden fazla sayıda çoklugoönderim ağacı üzerinden farklı tanımlamalar gönderilmiş ve bir ya da daha fazla ağaç üzerindeki ebeveynini kaybeden düğümler, diğer ağaçlar üzerinden tanımlamaları alarak video izlemeye devam etmişlerdir (Padmanabhan et al., 2003), (Magharei and Rejaie, 2006), (Baccichet et al., 2007).

2.6 Literatür Özet Tablo

Literatür özeti bölümünde sözü edilen çalışmalar akışlandırma yöntemi, video kodlama tekniği, hata dayanıklılığını sağlama yolu ve kaplama mimarisi açısından incelenerek çalışmaların genel özellikleri çizelge 2.1'de listelenmiştir.

Çizelge 2.1. Literatür özet tablo.

Çalışma	Akışlandırma Yöntemi	Video Kodlama Tekniği	Hata Dayanıklılığı	Kaplama Mimarisi
(Chu et al., 2002)	Çoklugoönderim Ağacı	Genel	Reaktif	Gecikme
(Banerjee et al., 2002)	Çoklugoönderim Ağacı	Genel	Reaktif	Gecikme
(Padmanabhan et al., 2003)	Birden çok Çoklugoönderim Ağacı	MDC	Proaktif	-
(Rejaie and Ortega, 2003)	Çoklugoönderim Ağacı	Ölçeklenebilir Video	Reaktif	-

(Tran et al., 2004)	Çoklugönderim Ağacı	Genel	Reaktif	Gecikme
(Zhang et al., 2005b)	Örgü	Genel	Örgü tabanlı	Gecikme
(Okada et al., 2005)	Çoklugönderim Ağacı	Ölçeklenebilir Video	Proaktif	-
(Kusumoto et al., 2005)	Çoklugönderim Ağacı	Genel	Proaktif	-
(Venkatamaran et al., 2006)	Karma	Genel	Örgü tabanlı	-
(Pouwelse et al., 2007)	Örgü	Genel	Örgü tabanlı	İlgi
(Baccichet et al., 2007)	Birden çok Çoklugönderim Ağacı	MDC	Proaktif	-
(Fei and Yang, 2007)	Çoklugönderim Ağacı	Genel	Proaktif	-
(Jeon et al., 2008)	Çoklugönderim Ağacı	Genel	Proaktif	-
(Wang et al., 2010)	Karma	Genel	Örgü tabanlı	-
(Ding et al., 2010)	Örgü	Ölçeklenebilir Video	Örgü tabanlı	-
(PPLive, 2010), (PPStream, 2010), (NextShare, 2010)	Örgü	Genel	Örgü tabanlı	-

3. ALTYAPI OLARAK KULLANILAN KAPLAMA MİMARİSİ

Tezin bu bölümünde, video akışlandırma sisteminin altyapısını sağlayan kaplama mimarisinden söz edilmektedir. Kaplama mimarisi, görevdeş ağların ölçeklenebilirlik sorununa bir çözüm getirmektedir. Tezde kullanılan kaplama mimarisi (Banerjee et al., 2002), (Tran et al., 2004) ve (Birrer and Bustamente, 2005) çalışmalarına benzerlik göstermektedir. Video akışlandırma sisteminin oluşturulmasında 2 tip protokol kullanılmıştır; kontrol protokolü olarak isimlendirilen ilk kısımda gecikme tabanlı bir kaplama mimarisi geliştirilmiştir. Akışlandırma protokolü olarak isimlendirilen ikinci kısımda ise, kaplama mimarisi üzerinde, mimarinin hiyerarşik katmanlarından yararlanarak çoklugoönderim ağaçları oluşturulmakta, video paketleri oluşturulan bu ağaç üzerinden gönderilmektedir. Tezin bu bölümünde, tasarlanan kaplama mimarisinin, bir başka deyişle kontrol protokolünün ayrıntılarına yer verilmiştir. Tezin bundan sonraki bölümünde çoklugoönderim ağacı oluşturma algoritması, beşinci bölümünde ise önerilen çoklugoönderim ağaç oluşturma algoritmasının kaplama mimarisi üzerinde uygulamasının detayları verilmiştir.

Görevdeş ağ üzerinde çalışan bir sisteme bağlı bulunan düğüm sayısı, sisteme giriş ve çıkış sayısının yoğunluğu nedeniyle sıkça değişmektedir. Bu değişime uyum sağlayabilecek bir video akışlandırma sistemi için kurulması önerilen yapıda, sisteme bağlı düğümler ile gruplar oluşturulmakta, bu grupların arasından seçilecek olan lider düğümler ise üst hiyerarşik katmanları oluşturmaktadır. Tez kapsamında geliştirilen kaplama mimarisi gecikme metriği temel alınarak oluşturulmuştur. Ancak aynı algoritma, ilgi metriği ile de kullanılabilir, bu algoritmanın ilgi metriği kullanılarak elde edilen kazanım sonuçları (Bulut et al., 2010) çalışmasında bulunabilir.

Sisteme ilk olarak giriş yapacak olan düğüm, herkes tarafından adresi bilinen bir sunucuya başvurur ve sisteme katılmak üzere ilk mesaj atacağı düğümün adres bilgisini alır. Tasarlanan sistemde, düğümlerin, hiyerarşik sistem yapısına uygun olarak hareket edebilmesinden yararlanarak uygun gruba yerleşmeleri sağlanır. Düğümün ilk aldığı adres, sistemde bulunan en üst hiyerarşik katmanın liderinin adresidir. En üst katmanda kendisine en yakın grup liderini belirleyen düğüm, bir alt hiyerarşik katmana geçerek aynı algoritmayı tekrarlar. En alt hiyerarşik katmanda ise kendisine aynı şekilde en uygun gruba yerleşir. Katılım işlemi, n adet düğüm içeren bir sistem için her grubun m adet düğümden oluştuğu varsayılırsa, $\log_m n$ adım sürmektedir. Bir düğümün sisteme

girişinden itibaren bir gruba kayıtlanmasına kadar olan süreye önyükleme adı verilir.

3.1 Hiyerarşik Katmanların Oluşturulması

Sisteme ilk olarak giriş yapan düğüm, sunucudan bağlanmak üzere bir düğüm adresi talep ettiğinde, sunucu bu düğüme kendisinin sistemdeki ilk düğüm olduğunu, dolayısıyla kurulacak ilk grubun lideri olduğunu belirten bir mesaj gönderir. Bu düğüm, sistemde en alt katmanda kurulan ilk grubun tek düğümü, aynı zamanda lideridir. Sunucu, sisteme daha sonra giren düğümlere bu düğümün adresini gönderir. Sisteme ikinci olarak dahil olan düğüm, sistemdeki tek grubun aday lideri olarak görevlendirilir. Bundan sonra sisteme giren düğümlerin tümü oluşan bu ilk gruba yönlendirilecektir.

Grup içindeki düğüm sayısı belli bir eşik değerini geçtikten sonra grup bölünmesi gerçekleştirilir. En alt hiyerarşik katmanda grup bölünmesinin gerçekleştirilmesi ile birlikte bir üst hiyerarşik katman oluşturulur; en alt katmanda bölünmeyi başlatan lider üst hiyerarşik katmanda lider, diğer grup lideri ise bir üst katmanda aday lider olarak görevlendirilir. Bu aşamadan sonra herhangi bir katmanda grup bölünmesi gerçekleştirildiğinde, eğer bu bölünmeyi gerçekleştiren lider, bir üst katmanda da lider değil ise, bir üst katmanda kayıtlanmak üzere liderine mesaj gönderir.

3.2 Grup Bölünmesi

Grup bölünmesi için tasarlanan algoritmanın çalışması için grup liderinin grup eleman sayısının belli bir eşik değerini geçtiğini tespit etmesi ve kendisine mantıksal olarak en uzak olan düğümü belirlemesi gerekmektedir. Grup bölünmesi için az sayıda düğüm arasında çalışan ve düğümlerde bulunan grup elemanları ile ilgili veri yapılarını kullanabilecek hızlı bir algoritmaya ihtiyaç duyulmuştur. Literatürde var olan grup bölünme algoritmaları çok sayıda eleman içeren grupların etkin bölünmelerini sağlasa da, varolan algoritmaların dağıtık versiyonlarının mesaj karmaşıklığı, görevdeş ağlar gibi dinamik ortamlar için uygun değildir.

Lidere mantıksal olarak en uzak düğümü belirlemek için gözönüne alınan 3 ölçüt vardır. Bunlardan iki tanesi gecikme ve zıplama adedi, bir diğeri ise en uzaktaki düğümün komşularına olan uzaklığıdır. Lidere en uzak düğümü bulmak

için sadece gecikme değerini gözönüne almak aslında lidere uzak olmayan ama bağlantı kalitesinin kötü olmasından dolayı en uzak düğüm olarak seçilerek performansı iyi olmayan bir grup bölünmesi gerçekleştirilmesine sebep olmaktadır. Zıplama adedi de en uzak düğümün tespit edilmesinde gözönüne alındığında ise lidere uzaklığı belli bir değeri aşan düğümler belirlenebilir. Ancak yine gecikme değeri de kullanıldığı için, bağlantı kalitesi kötü bir düğümün en uzak düğüm olarak seçilme olasılığı vardır. Bu olasılığı en az indirmek için ise, zıplama adedi ve gecikme değeri yüksek olan düğümlerden komşu düğümleri ile arasındaki gecikme değerlerinin ortalaması istenmektedir. Bu durumda seçilen en uzak düğüm, aynı zamanda bağlantı kalitesi iyi olan bir düğüm olacaktır.

Bu aşamadan sonra lider ve lidere en uzak düğüm, grup listesindeki düğümleri oluşturulacak yeni gruba davet ederler. Bir düğüm, gruba katılma mesajını ilk hangi düğümden aldıysa onun grubuna dahil olur.

3.3 Lider Seçme Algoritması

Grup içinde lider seçme algoritması, belli zaman aralıklarında tekrarlanmaktadır. Lider, grup içi iletişimde en etkin olarak rol oynayacak düğüm olduğu için, lider seçiminde göz önüne alan ölçüt, grup içinde yer alan diğer elemanlara uzaklığı küçük olan bir düğüm seçmektir. Bu sebeple kendisine en uzak olan düğüm ile arasındaki uzaklık en küçük olan düğüm, lider seçilir.

3.4 Sonlu Durum Makinası

Görevdeş ağlar üzerinde çalışan kaplama mimarisini oluşturan ve bakımı yapan sistemin sonlu durum makinası (SDM) Ek 2’de verilmiştir.

Sonlu durum makinasında verilen süreç sisteme yeni bir düğümün katılması, ayrılması, grup bakımı ve lider seçimi algoritmasını içermektedir. Bu algoritma, sistemde bulunan tüm düğümler tarafından, tüm hiyerarşik katmanlar seviyesinde çalıştırılmaktadır.

Şekilde (Bkz. Ek 2) sonlu durum makinası verilen sistemde her düğüm BOŞ, NORMAL_DÜĞÜM, LİDER, YEDEK_LİDER ve LİDER_SEÇİM durumlarından birine sahiptir. Sistemde kontrol protokolü için çalışan 15 tip mesaj tanımlanmıştır. Bu mesajlar ve açıklamaları şu şekilde listelenebilir:

YENİ_DÜĞÜM: Sistemde var olan bir düğüm, herhangi bir hiyerarşik katmanda yer alan bir gruba katılacağı zaman bu mesajı o grubun liderine gönderir.

KATILAN_DÜĞÜM: Sisteme ilk kez giriş yapan düğüm tarafından, kendisine en uygun grubu bulmak için herhangi bir hiyerarşik katmanda bulunan grup liderlerine gönderilen mesaj tipidir.

YEDEK_GRUPLAR: Sisteme ilk kez giriş yapan bir düğüme grup lideri tarafından gönderilecek olan ve düğümün katılabileceği diğer grup liderlerinin adreslerini tutan mesaj tipidir.

YENİ_LİDER: Bu mesaj tipi, bir grup içinde yer alan lider değiştiği zaman ilgili düğümlere yeni lider adresini bildirmek üzere gönderilmektedir.

LİDER_GÖREVİ: Liderlik görevine getirilen düğüme gönderilecek olan mesaj tipidir.

GRUP_LİSTESİ: Yeni bir gruba dahil olan düğüme lider tarafından gönderilecek olan ve o grubun düğüm adresleri listesini içeren mesaj tipidir.

GECİKME_ÖLÇÜM_TALEP: Bir düğüm listesinde bulunan bir başka düğüm ile arasındaki gecikme değerini bilmiyor ise, bu mesaj tipinde bir mesajı göndererek gecikme hesaplama talebinde bulunur.

GECİKME_ÖLÇÜM: Bir önceki mesaj tipinde karşılık olarak gönderilen mesaj tipidir.

LİDER_SEÇİMİ: Lider seçimi algoritması sırasında lider ya da gruba üye düğümler tarafından gönderilen mesaj tipidir, yeni liderin seçimi için gerekli değerler bu mesaj içerisinde gönderilir.

KOMŞU_ORT_TALEP: Lider tarafından gönderilen bu mesaj, lidere en uzak düğümün belirlenmesi amacıyla gönderilir.

KOMŞU_ORT: Bir önceki mesaj tipinde karşılık olarak gönderilen mesaj tipidir.

AYRILMA: Bir düğümün bir gruptan haberli ayrılma sırasında gönderdiği mesaj tipidir.

GRUP_BÖLÜNMESİ: Lider tarafından lidere en uzak düğüme grup bölünmesi algoritmasını başlatmak üzere gönderilen mesajdır.

GRUBUMA_KATIL: Düğümlerin grup bölünmesi sırasında kendi grubuna almak üzere diğer düğümlere gönderdiği mesajdır.

YEDEK_LİDER: Grup içindeki bir düğüme, kendisinin yeni yedek lider olarak seçildiğini belirtmek amacıyla gönderilen mesajdır.

Sisteme ilk kez giriş yapan düğüm, BOŞ konumundadır, sunucudan aldığı lider adresi ile birlikte konumunu NORMAL_DÜĞÜM'e getirir. Bu konumdayken oluşabilecek olaylar, bunlar karşısında oluşacak çıktı ve düğümün geçeceği yeni durum Çizelge 3.1'de verilmiştir.

Çizelge 3.1. NORMAL_DÜĞÜM durumundaki bir düğüm için durum ve çıktı tablosu.

<i>Girdi</i>	<i>Çıktı</i>	<i>Bir Sonraki Durum</i>
YEDEK_LİDER	-	YEDEK_LİDER
LİDER_GÖREVİ	Üst katman varsa YENİ_DÜĞÜM	LİDER
GRUP_BÖLÜNMESİ	GRUBUMA_KATIL	LİDER
YENİ_DÜĞÜM	YENİ_LİDER	NORMAL_DÜĞÜM
GRUP_LİSTESİ	GECİKME_ÖLÇÜM_TALEP	NORMAL_DÜĞÜM
GRUBUMA_KATIL	GRUBUMA_KATIL	NORMAL_DÜĞÜM
KATILAN_DÜĞÜM	YENİ_LİDER	NORMAL_DÜĞÜM
YEDEK_GRUPLAR	KATILAN_DÜĞÜM	NORMAL_DÜĞÜM
GECİKME_ÖLÇÜM_TALEP	GECİKME_ÖLÇÜM	NORMAL_DÜĞÜM
LİDER_SEÇİMİ	LİDER_SEÇİMİ	NORMAL_DÜĞÜM
AYRILMA	-	NORMAL_DÜĞÜM
KOMŞU_ORT_TALEP	KOMŞU_ORT	NORMAL_DÜĞÜM

LİDER_GÖREVİ mesajını alan düğüm, artık bu grupta yeni liderdir ve eğer bir üst hiyerarşik katman oluşturulmuş ise, bu katmana kayıtlanmak zorundadır. Bu yüzden YENİ_DÜĞÜM tipinde bir mesajı üst katmanın grup liderine gönderir. GRUP_BÖLÜNMESİ mesajı aldığı anda, artık kendi oluşturacağı yeni grubun lideri olacağı için durumunu LİDER olarak değiştirir. YENİ_DÜĞÜM ve KATILAN_DÜĞÜM mesajları alması, bu mesajı atan düğümde bu düğümün lider olarak kayıtlı olmasından kaynaklanmaktadır. Bu durum daha önceden lider

olan ancak liderliđi son bulmuş düğümlere, henüz liderliđi devam ederken lider adresini almış düğümler tarafından gönderilebilir, bu durumda bu mesajı gönderen düğüme YENİ_LİDER mesajı ile yeni lider adresi bildirilir. Eğer düğüm, GRUP_LISTESİ mesajı almış ise, bir gruba yeni ilk kez kayıtlanıyor olabilir, grup listesindeki düğümler ile arasındaki gecikmeyi hesaplamak için listedeki düğümlere GECİKME_ÖLÇÜM_TALEP mesajı gönderir. Benzer şekilde GECİKME_ÖLÇÜM_TALEP mesajı aldığı anda ise, eğer mesajı gönderen düğüm ile arasındaki gecikmeyi henüz hesaplamamışsa, gelen mesajın gönderildiđi saate bakarak bu düğüm ile arasındaki gecikmeyi hesaplar. GRUBUMA_KATIL mesajı aldığı anda, eğer bu mesajı alan düğümün grubuna girmeyi kabul ederse, yeni liderine bunu belirtmek üzere GRUBUMA_KATIL mesajını geri gönderir. Eğer düğüm YEDEK_GRUPLAR mesajını aldıysa, sisteme ilk kez giriş yaparak aday liderlerin adreslerini almıştır. Bu aday liderlere, kendisine en yakın lideri, dolayısıyla en yakın grubu belirlemek üzere KATILAN_DÜĞÜM mesajını, mesaj aldığı katmanın bir alt hiyerarşik katmanında gönderir. Bu durumda KATILAN_DÜĞÜM mesajı, hiyerarşik katmanlar arasında dolaşarak, düğümün en alt katmanda en yakın gruba girmesini sağlar. LİDER_SEÇİMİ mesajını alan NORMAL_DÜĞÜM konumundaki bir düğüm, listesinde bulunan kendisine en uzak düğüm ile arasındaki gecikmeyi; liderinin grup bölünmesi başlatmadan önceki en uzak düğümü bulmasında kullanılan KOMŞU_ORT_TALEP mesajını aldığı anda komşuları ile kendisi arasındaki ortalama gecikmeyi liderine mesajlar. Bir düğümden AYRILMA mesajı aldığı anda ise, bu düğümü listesinden çıkarır.

LİDER konumunda bulunan bir düğümün alabileceđi mesajlar, gerçekleşebilecek girdiler, çıktılar ve liderin bir sonraki durumu Çizelge 3.2’de verilmiştir.

Grup lideri, YENİ_DÜĞÜM, GRUBUMA_KATIL ya da en alt hiyerarşik katmanda gelen KATILAN_DÜĞÜM mesajı alırsa bu mesajı atan düğümü grup listesine ekler ve grup listesini düğüme gönderir. KATILAN_DÜĞÜM mesajını alan lider eğer bu mesajı en alt hiyerarşik katmanda almamış ise, sisteme yeni giren düğümün en yakın grubu bulabilmesi için o katmanda bulunan düğümlerin listesini düğüme gönderir. AYRILMA mesajını alan lider, bu mesaj grubun yedek liderinden gelmiş ise, kendisine yeni bir yedek lider seçer ve bu seçimi YEDEK_LİDER mesajı ile bu düğüme bildirir. Grup eleman sayısı eşik değerine ulaştıđı zaman, grup bölünme algoritmasını çalıştırmaya başlamadan önce en uzak düğümü seçmek için KOMŞU_ORT_TALEP mesajlarını gönderir, aldığı son mesajdan sonra seçtiđi mantıksal olarak en uzak ve yeni grubun lideri olmaya

aday düğüme GRUP_BÖLÜNMESİ mesajını gönderir. Lider seçiminin yenilenmesi gerektiğini belirleyen zamanlayıcının tetiklemesi ile lider seçimi algoritması çalışır ve lider grup elemanlarına seçimin başladığını LİDER_SEÇİMİ mesajı göndererek haber verir.

Çizelge 3.2. LİDER durumundaki bir düğüm için durum ve çıktı tablosu.

<i>Girdi</i>	<i>Çıktı</i>	<i>Bir Sonraki Durum</i>
YENİ_DÜĞÜM	GRUP_LİSTESİ	LİDER
GRUBUMA_KATIL	GRUP_LİSTESİ	LİDER
KATILAN_DÜĞÜM	if en alt hiy. katman GRUP_LİSTESİ else YEDEK_GRUPLAR	LİDER
AYRILMA	Kaynak yedek lider ise YEDEK_LİDER	LİDER
KOMŞU_ORT	GRUP_BÖLÜNMESİ	LİDER
Grup eleman sayısı eşik değerine ulaştı	KOMŞU_ORT_TALEP	LİDER
ZAMAN_AŞIMI:Lider Seçimi	LİDER_SEÇİMİ	LİDER_SEÇİM

Yedek lider durumundaki bir düğümün girdi, çıktı ve sonraki durumu Çizelge 3.3'te listelenmiştir.

Yedek lider, gruptaki lider sistemden ayrılırsa lider olarak görev alacak olan düğümdür; ancak bunun dışındaki tüm davranışları NORMAL_DÜĞÜM durumundaki bir düğümle benzerlik taşımaktadır. Eğer yedek lider durumundaki bir düğüm, grup liderinden AYRILMA mesajı alırsa, kendisini lider ilan eder ve yeni bir yedek lider seçerek bu düğüme YEDEK_LİDER mesajı gönderir. Yedek liderin bunun dışındaki tüm davranışları çizelge 3.1 için yapılan açıklamalardan izlenebilir.

Çizelge 3.3. YEDEK_LİDER durumundaki bir düğüm için durum ve çıktı tablosu.

<i>Girdi</i>	<i>Çıktı</i>	<i>Bir Sonraki Durum</i>
LİDER_GÖREVİ	YENİ_LİDER, Üst hiy. Katman varsa YENİ_DÜĞÜM	LİDER
GRUP_BÖLÜNMESİ	GRUBUMA_KATIL	LİDER
AYRILMA	Kaynak lider ise YENİ_LİDER, YEDEK_LİDER	Kaynak lider ise LİDER
YENİ_DÜĞÜM	YENİ_LİDER	YEDEK_LİDER
GRUP_LİSTESİ	GECİKME_ÖLÇÜM_TALEP	YEDEK_LİDER
GRUBUMA_KATIL	GRUBUMA_KATIL	YEDEK_LİDER
KATILAN_DÜĞÜM	YENİ_LİDER	YEDEK_LİDER
GECİKME_ÖLÇÜM_TALEP	GECİKME_ÖLÇÜM	YEDEK_LİDER
LİDER_SEÇİMİ	LİDER_SEÇİMİ	YEDEK_LİDER
KOMŞU_ORT_TALEP	KOMŞU_ORT	YEDEK_LİDER

Çizelge 3.4’te LİDER_SEÇİMİ durumundaki bir düğüm için girdi, çıktı ve sonraki durum listesi yer almaktadır.

Çizelge 3.4. LİDER_SEÇİMİ durumundaki bir düğüm için durum ve çıktı tablosu

<i>Girdi</i>	<i>Çıktı</i>	<i>Bir Sonraki Durum</i>
YENİ_DÜĞÜM	GRUP_LİSTESİ	LİDER_SEÇİMİ
KATILAN_DÜĞÜM	if en alt hiy. katman GRUP_LİSTESİ else YEDEK_GRUPLAR	LİDER_SEÇİMİ
AYRILMA	-	LİDER_SEÇİMİ
LİDER_SEÇİMİ	if lider isem YEDEK_LİDER	LİDER
LİDER_SEÇİMİ	if yedek lider isem LİDER_GÖREVİ, AYRILMA	YEDEK_LİDER
LİDER_SEÇİMİ	if Lider ya da yedek lider değil isem LİDER_GÖREVİ, YEDEK_LİDER, AYRILMA	NORMAL_DÜĞÜM

LİDER_SEÇİM durumu, liderin lider seçme algoritmasını çalıştırırken geçtiği bir durum olduğu için, YENİ_DÜĞÜM ve KATILAN_DÜĞÜM mesajlarına lider durumunda verdiği tepkileri verir. Ancak AYRILMA mesajı

aldığında bu mesaj yedek liderden olsa bile yedek lider seçimini lider seçme algoritmasının sonuna kadar erteler. Lider seçimi algoritması, alınan son LİDER_SEÇİMİ mesajı ile sonlanır, seçim sonunda yine kendisi lider çıkarsa, lider durumuna geri döner; eğer yedek lider değişti ise yeni yedek lidere bu durumu mesajlar. Eğer kendisi yedek lider seçilmişse yedek lider konumuna geçer ve yeni lidere LİDER_GÖREVİ mesajı gönderir. LİDER_GÖREVİ mesajı içinde, eğer var ise üst katman grup liderinin adresi belirtilir. Lider seçimi sonunda kendisi ne lider, ne de yedek lider seçilmez ise, yeni lider ve yeni yedek lidere yeni görevlerini bildirir. Lider, YEDEK_LİDER ya da NORMAL_DÜĞÜM durumuna geçerse üye olduğu tüm üst hiyerarşik katmanlarda AYRILMA mesajı gönderir.

BOŞ durumunda olan bir düğüm, YENİ_LİDER mesajı alırsa, görevdeş sisteme giriş yapmak için başvurmuştur, ve sistemde bulunan en üst liderin adresi gelen mesaj içindedir. Bu mesajı alan düğüm, durumunu NORMAL_DÜĞÜM yaparak en üst katman liderine KATILAN_DÜĞÜM mesajı gönderir.

Eğer bir düğüm sistemden ya da herhangi bir hiyerarşik katmandan ayrılırsa, ayrıldığı katman/katmanlardaki durumunu BOŞ'a konumlandırır. Bu durum, Ek 2'de verilen SDM ve girdi/çıktı tablolarında gösterilmemiştir.

Sistemde sunucu görevini üstlenen düğüm, kendisine gelen iki tip mesajı kabul etmektedir. Eğer sunucu, YENİ_LİDER tipinde bir mesaj alırsa, sistemde bulunan en üst katman liderden gelen bu mesajın kaynağını, sisteme girecek düğümlere yönlendirmek üzere saklı tutar. KATILAN_DÜĞÜM mesajı tipinde bir mesaj aldığında, elindeki adresi, dolayısıyla o an sistemde bulunan en üst hiyerarşik katmanın lider adresini bu düğüme mesajlar. Sunucuda her zaman sadece bir düğüm adresi bulunmaktadır.

Kaplama mimarisi ile ilgili olarak çalıştırılan algoritmalar Ek.3'te verilmiştir. Mimaride oluşturulan gruplar için doğruluk %85, hassasiyet ($\gamma \leq 1$ için) %80 olarak belirlenmiştir. Doğruluk ve hassasiyet formülleri sırasıyla eşitlik 3.1 ve 3.2'de (Zhang et al., 2002) verilmiştir.

γ = Referans grup liderine olan RTT / En yakın grup liderine olan RTT (Eş. 3.1)

- N : Sistemdeki toplam düğüm sayısı
- K : Sistemdeki toplam grup sayısı
- G_i : i. grup

- $|G_i|$: G_i grubundaki düğüm sayısı
 L_i : G_i grubunun lideri
 X : G_i grubu içerisindeki bir düğüm
 Y : G_j grubu içerisindeki bir düğüm
 (x,y) : x ve y düğümleri arasındaki gecikme

$$1 - \frac{1}{N'} \sum_{\substack{i,j \in [1..k] \\ i < j}} \sum_{\substack{x \in G_i \\ y \in G_j}} \frac{|(x,y) - (L_i, L_j)|}{(x,y)}, N' = \sum_{\substack{i,j \in [1..k] \\ i < j}} |G_i| |G_j|. \quad (\text{Eş. 3.2})$$

4. ÇOKLUGÖNDERİM AĞAÇLARININ OLUŞTURULMASI

Tez kapsamında, görevdeş ağlar üzerinde çalışacak video akışlandırma yolunun kurulması için akışlandırma oturumunda sağladığı düşük mesaj karmaşıklığı ve ölçeklenebilir video kodlamaya uygun bir altyapı sağlaması sebebiyle çoklugönderim ağaçlarının kurulmasına karar verilmiştir. Önerilen çoklugönderim ağaç oluşturma algoritması, düğümlerin yükleme hızlarını ve gecikme cinsinden birbirlerine olan uzaklıklarını dikkate alarak bantgenişliğini etkin bir biçimde kullanmaktadır. Buna ek olarak, çoklugönderim ağaçlarının en büyük dezavantajlarından biri olan hata dayanıklılığını sağlamak üzere yeni bir yöntem geliştirilmiştir. Tezin altıncı bölümünde, video akışlandırma oturumunda kullanılması önerilen çoklugönderim ağaçlarında hata dayanıklılığı sağlamak üzere geliştirilen yöntemin ayrıntıları verilmiştir. Tezin bu bölümünde, bir ya da birden fazla kaynak ve bir ya da birden fazla istemci bulunması durumunda çoklugönderim ağaçlarını oluşturma yöntemi açıklanacaktır. Burada açıklanan algoritma, tek bir grup içerisinde çalışan bir çoklugönderim ağaç oluşturma algoritması olarak düşünülebilir. Tezin beşinci bölümünde ise, bu bölümde önerilen çoklugönderim ağaç oluşturma algoritmasının hiyerarşik katmanlar üzerinde uygulama yöntemleri verilmiştir. Çoklugönderim ağacı oluşturulurken tüm hiyerarşik katmanlarda bulunan tüm gruplar içerisinde ağacın bir kısmı oluşturulmaktadır, dolayısıyla önerilen sistem dağıtık ve ölçeklenebilir bir yapı sunmaktadır.

Oluşturulacak çoklugönderim ağaçlarının ölçeklenebilir P2P video akışlandırma sistemi ihtiyaçlarına karşılık vermesi beklenmektedir. Bu amaçla kurulmuş bir çoklugönderim ağacında bulunması beklenen temel özellikler aşağıdaki şekilde maddelenebilir:

- *Düğümler arasındaki ağ bağlantısının düğümler açısından en etkin şekilde kurulması:* Ağaç üzerinde birbirine bağlanacak düğümler arasında (ebeveyn ile çocukları arasında) gecikmenin en az düzeyde, bantgenişliğinin ise video akışlandırmaya uygun bir değerde olması gerekmektedir.
- *Ağacın yüksekliğinin çok fazla olmaması:* Ağaç yüksekliği arttıkça, yaprak düğümlere doğru video verisinin varma süresi uzayacak ve daha önemlisi, ağaçta bulunan ara düğümlerin sayısının artması, bu düğümlerde oluşabilecek hatalardan dolayı daha fazla hata olmasına, dolayısıyla ağacın

bakımının zorlaşmasına neden olacaktır. Bu yüzden ağaç yüksekliğinin belirli bir seviyeyi geçmemesi gerekmektedir.

- *Bir düğüme bağlı çocuk sayısının belirli bir sınırın altında tutulması:* Ağaç üzerindeki bir düğüm, çocuklarına yüksek bant aralığı sağlayabilecek bir konumda olabilir; ancak kendisine bağlanan çocuk sayısı arttıkça, bu düğümler ile ebeveyn arasındaki bağlantı kalitesinin düşmesine neden olur. Bu sebeple oluşturulacak ağaçta düğümlere bağlanan çocuk sayısının kontrolü sağlanmalıdır.
- *Birden fazla başlangıç düğümünün olduğu durumda çoklugoönderim ağaç oluşturma algoritmasının çalıştırılabilmesi:* Görevdeş bir sistemde aynı videoyu elinde bulunduran birden fazla kaynak, aynı düğüm listesi üzerinde ağaç oluşturmak durumunda kalabilir. Çoklugoönderim ağaç oluşturma algoritmasının, bu koşulda da sorunsuz çalışması beklenmektedir.

4.1 Chang-Roberts Algoritması ile Elde Edilen Çoklugoönderim Ağaçları

Chang-Roberts çoklugoönderim ağaç oluşturma algoritması, çoklugoönderim ağacını oluşturacak düğümler arasında gecikmeyi temel alarak bir ağaç yapısı oluşturulmasını sağlar (Chang, 1982). Chang-Roberts algoritmasının en önemli avantajı, düğümlerin ağaç oluşturmak için birbirlerine olan uzaklıklarını önceden bilmelerine gerek duymamalarıdır.

Algoritma, önceden belirlenen bir düğüm tarafından çalıştırılmaya başlatılır ve bu düğüm, ağaçta yer alacak tüm düğümlere mesaj gönderir. Bu özelliği ile ağaç oluşturma algoritmasının birden fazla düğüm tarafından başlatılabilmesine, dolayısıyla sistemde birden fazla kaynak olduğu durumda ağaç oluşturulabilmesine olanak sağlar. Bir düğüm, ilk kez mesaj aldığı anda, mesajı gönderen kişiyi ebeveyni olarak belirler ve bu düğüm hariç diğer tüm düğümlere mesaj gönderir. Algoritmanın mesaj karmaşıklığı, n adet düğümden oluşan bir sistem için, mantıksal mimaride tüm düğümlerin birbirleri ile komşu olduğu göz önünde bulundurulduğunda $O(n^2)$ dir.

```

integer: N (komşu sayısı), C, D
ebeveyn=i, C=0, D=0;
Kaynak_başlangıç()
//Bu rutin, algoritmayı başlatan düğümler tarafından çalıştırılır
tüm komşulara mesaj gönder;
D = komşu sayısı;
do{
    if (D != 0  $\wedge$  yankı_mesajı)
        D = D-1;
}while (D>0);
İstemci_başlangıç()
//Bu rutin, mesaj alan düğümler tarafından çalıştırılır
do {
    if (mesaj  $\wedge$  ebeveyn=i  $\wedge$  C=0) {
        C=1;
        ebeveyn=gönderici;
        if (i yaprak değil ise) {
            ebeveyn hariç tüm komşulara mesaj gönder;
            D=ebeveyn hariç komşu sayısı;}
        }
    if (yankı_mesajı) D = D-1;
    if (mesaj $\wedge$  gönderici!=ebeveyn) göndericiye yankı_mesajı gönder;
    if (C=1  $\wedge$  D=0) {
        ebeveyne yankı_mesajı gönder;
        C=0;
    }
}

```

Şekil 4.1. Chang-Roberts algoritması sözde kodu.

Chang-Roberts algoritması Şekil 4.1’de verilmiştir. Algoritma, görevdeş sistem mimarisi üzerinde test edildiğinde video akışlandırmaya uygun çokluguönderim ağaçlarının oluşmadığı gözlenmiştir. Chang-Roberts algoritması ile ilgili kazanım sonuçları, ağaç üzerinde bulunacak düğümlerin tespiti yöntemi açıklandıktan sonra, tezin daha sonraki bölümlerinde yer almaktadır. Literatürde bulunan diğer çokluguönderim ağaç oluşturma algoritmalarının da benzer dezavantajlarının bulunacağı gözönünde bulundurularak yeni bir çokluguönderim ağaç oluşturma algoritması tasarlanmıştır. Bir sonraki altbölümde tasarlanan çokluguönderim ağaç algoritması açıklanacaktır.

4.2 Kapasite-Farkındalıklı (Capacity-Aware) Çokluguönderim Ağaç Oluşturma Algoritması

4.2.1 Problem formülasyonu

Üzerinde çalışılan ağ yapısı yönsüz bir çizge olarak modellenenebilir. Buna göre, verilen bir $G(V, E)$ çizgesinde $e_{u,v}$ değeri, $u, v \in V$ için bu iki düğümü bağlayan kenarın maliyetine eşittir. Yukarıdaki bölümde sözü edildiği gibi, video akışlandırma uygulaması için kurulan ağaçlarda bulunması gereken özellikler bu model üzerinden aşağıdaki şekilde listelenebilir:

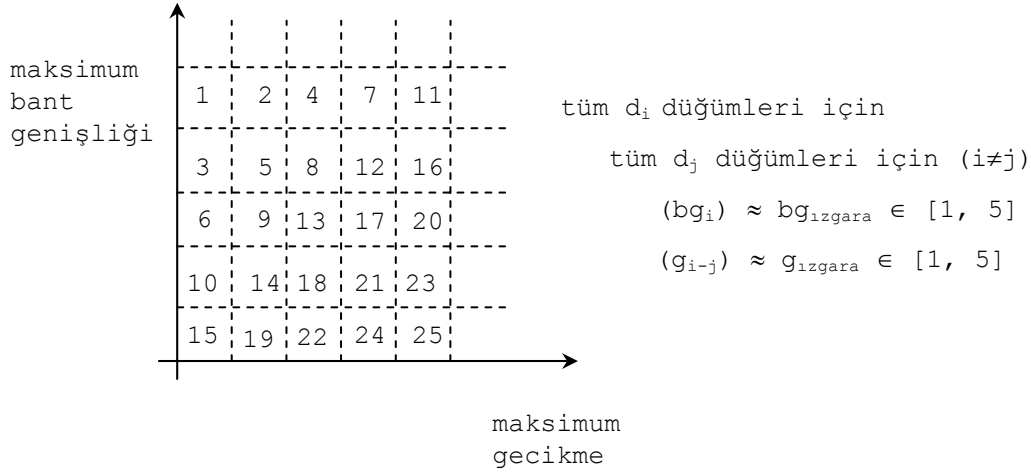
Kurulan $T(V, E')$ ağacında,

- Her $u \in V$ için, k_u değeri, u düğümünün kapasitesini, başka bir deyişle u düğümüne bağlanabilecek en yüksek sayıda düğümü ifade etsin. Kapasiteyi belirleyen değer, bir düğümün yükleme hızına (upload) bağlı bağlanabilecek düğüm sayısıdır, dolayısıyla düğümün ebeveyni, kapasitesinin azalmasına sebep olmaz. Ağaç üzerine yerleşen her düğüm için d_u ağaç üzerinde u düğümüne bağlı düğüm sayısını ifade ediyorsa, $d_u \leq k_u$ olmalıdır.
- Ağaç uzunluğunun mümkün olduğunca kısa tutulması için, henüz ağaca bağlanmamış düğümlerin halen var olduğu her durumda $d_u = k_u$; aksi durumda $d_u < k_u$ olmalıdır.

Ağaca yerleşecek düğümler gözönünde bulunduğunda, ilk olarak kaynak düğümlere bağlanacak düğümlerin seçimi gerçekleştirilebilir. Ağ üzerinde birden fazla sayıda kaynak bulunan durumlar için, her düğüm, ağaç uzunluğunun kısa tutulmasını sağlamak amacıyla, eğer kapasiteleri uygun ise, kaynaklardan birine bağlanmalıdır. Bu bağlanma işlemi, her düğüm için, en az maliyetli kaynak düğüme bağlanması ile gerçekleştirilir. Buna göre, minimum kenarlara sahip değerler ağaç için seçilecektir. Ancak, kenar seçimlerinde, bir kaynak düğüme bağlanacak olan düğümlerin sayısının o kaynak düğümün kapasitesini geçmemesi gerekmektedir. Dolayısıyla kaynaklara bağlanan düğümler dışında kanal ve video talep eden düğümler seçim sonrasında ağaca henüz yerleştirilmemiş durumdadır, ağaca yerleşen düğümler, kaynaklar ile arasında en az maliyete sahip olan düğümlerdir.

4.2.2 Maliyet tablosu

Verilen algoritma tasarımına göre, maliyet açısından her düğüme kaynağa hem maliyet hem de zıplama açısından uygun bir ağaç yapısı oluşturmalıdır. Ayrıca önerilen sistemde, maliyetlerin belirlenmesinde düğüm kapasiteleri ve düğümler arasındaki gecikme birlikte değerlendirilmiştir. Görevdeş ağ üzerinde yer alan düğümler, diğer düğümler ile aralarındaki gecikme ve düğümlerin yükleme bantgenişliği değerlerine bağlı olarak bu düğümler ile arasındaki bağlantı kalitesini belirleyen bir ağırlık değeri hesaplar. Bu değer hesaplanırken, tüm gecikme değerleri ve bantgenişliği değerleri 1 ile 5 arasında bir değer ile eşleştirilir ve 5x5 lik bir ızgara oluşturulur. Bu ızgara üzerinde gecikme ve bantgenişliği değerlerine uygun şekilde yerleştirilen düğüm adreslerine ızgara üzerindeki konumlarına göre Şekil 4.3'teki ağırlık değerleri verilir. Şeklin sağında yer alan i ve j değerleri, sistemde bulunan düğüm sayısına eşittir.



Şekil 4.3. Gecikme – Bantgenişliği çiftine bağlı ağırlık değerleri ızgarası.

Video akışlandırma uygulamalarında, akışlandırma yapılan iki düğüm arasındaki bantgenişliği değeri, aralarındaki sabit gecikme değeri ile karşılaştırıldığında akışlandırma kalitesini etkileyen daha önemli bir ölçüttür. Video akışlandırma amaçlı oluşturulan bir çokluguönderim ağacının, düşük yükleme kapasitesine sahip düğümlerin ağacın yapraklarında, yüksek yükleme kapasitesine sahip düğümlerin ise ağacın daha üst seviyelerinde olacak şekilde yapılandırılması, bu ağaç üzerinde gerçekleştirilen video akışlandırma kalitesinin arttırılmasını sağlar. Bu yüzden oluşturulan ızgara değerlerinde, bantgenişliği değerine öncelik verilerek ağırlık değerlerinin bu yönde atanması sağlanmıştır. Bir sonraki altbölümde verilen algorithmada kenarların maliyetlerinde bu bölümde verilen ızgaradaki değerler kullanılmaktadır.

4.2.3 Algoritma

Verilen bilgiler ve kısıtlar doğrultusunda tasarlanan ağaç oluşturma algoritması Şekil 4.2’de verilmiştir.

```

1   K = { k1, k2, ..., kn }
2   L = { l1, l2, ..., lm }
3   Ls = ∅

4   while L ≠ ∅ do
5       n = |K|;

6       while (  $\sum_{i=1}^n d(k_i) < \sum_{i=1}^n k(k_i)$  ) && (L ≠ ∅) do
7           a ∈ K ve b ∈ L olan ea,b optimum maliyete
              sahip kenarı seç;
8           Ls = Ls ∪ {b};
9           L = L - {b};
10          da = da + 1;
11          if (da == ka) then K = K - {a} end if
12      end while
13      K = Ls;
14      Ls = ∅;
15  end while

```

Şekil 4.2. Çokluguönderim ağaç oluşturma algoritması sözde kodu.

N adet kaynak düğüm ve m adet videoyu izlemek isteyen düğüm bulunduğu durumda, şekilde verilen algoritmaya göre, ilk 3 satırda kaynaklar K kümesine, video izlemek isteyen düğümler ise L kümesine atılır. Her katmanda ağaca yerleştirilmiş düğümler ise L_s kümesinde yer alacaktır ve bu küme boş olarak ilklenir. Algoritmanın 6. satırında yer alan *while* döngüsü, kaynak kümesinde bulunan düğümlerin dereceleri toplamının kapasitelerini aşp aşmadığını ve ağaca bağlanmamış düğümün kalıp kalmadığını kontrol eder. Bu satırda yer alan $d(k_i)$ terimi k_i düğümünün derecesini (o ana kadar bağlanmış olan çocuk sayısı), $k(k_i)$ terimi k_i düğümünün kapasitesini ifade etmektedir. Ağaca bağlanmamış düğüm kalmadıysa, algoritma sonlanır. Eğer tüm kaynakların kapasitesi dolarsa, ağacın bir sonraki katmanına geçilir ve yeni bağlantılar kurulur. Algoritmanın 7. satırında, minimum maliyete sahip kenarlar seçilerek ağaca bağlanan düğümler seçilir ve 8. satırda L_s kümesine eklenir. Seçilen düğümün bağlandığı kaynağın

derecesi 1 arttırılır (satır 10), ve derecesi kapasitesine ulaşmış ise, artık yeni bir bağlantı alamayacağı için kaynak kümesinden çıkarılır (satır 11). Algoritmanın 13. satırında, L_s kümesi yeni kaynak kümesi olarak atanır ve 14. satırda seçilen düğüm listesi silinerek algoritma ağacın bir alt seviyesindeki düğümleri belirlemek üzere tekrar başa döner.

5. GÖREVDEŞ AĞ ÜZERİNDE HATA DAYANIKLILIĞINA SAHİP ÇOKLUGÖNDERİM AĞAÇLARININ OLUŞTURULMASI

Tezin bir önceki bölümünde verilen çoklugoönderim ağaç oluřturma algoritması, önbilgi olarak bir grup düğüm üzerinde, düğümler arasındaki ikili gecikme deęerleri ve düğümlerin yükleme hızlarının bilinmesini gerektirmektedir. Ancak bu algoritmanın, tez kapsamında önerilen görevdeş hiyerarşik katmanlı sistemde çalıştırılabilmesi ve ölçeklenebilir olabilmesi için, dağıtık hale getirilerek bu sisteme uyarlanması gerekmektedir.

Tasarlanan görevdeş sistem mimarisi, kontrol ve akışlandırma protokolü olmak üzere iki kısma ayrılmıştır. Grup liderleri kontrol protokolü dahilinde grupların, hiyerarşik katmanların oluřturulması ve bakımı ile görevlidir. Çoklugoönderim ağacının oluřturulması ve ağaç üzerinde gerçeklenecek video akışlandırma ile ilgili kısımlar ise sistemde, akışlandırma protokolünün parçası olarak yer almaktadır. Akışlandırma protokolü dahilinde belli görevlerden sorumlu olan düğümler akışlandırma liderleri olarak adlandırılmaktadır.

Tezin bu bölümünde, çoklugoönderim ağacını oluřturacak olan kaynak ve istemci düğümlerin belirlenmesi, akışlandırma liderlerinin seçimi, ağacın oluřturulması sırasında düğümler arasında çocuk seçimlerinin yapılması, düğüm kapasitelerinin hiyerarşik katmanlar arasında dağılımı ve son olarak ağaç üzerinde hata dayanıklılığının sağlanmasından söz edilecektir.

5.1 Videoların Talep Edilmesi ve Aranması

P2P sistem üzerinde bulunan grup liderlerinin yürütmeleri gereken görevler arasında, grupta bulunan videoların kaydını tutmak, grup içersinden gelen video taleplerini toplamak, talep edilen videoları sistemde aramak ve kaynak düğümler ile video talebinde bulunan düğümler arasında bağlantı kurulmasını sağlamak bulunmaktadır.

Tasarlanan yapıda, tüm hiyerarşik katmanlar ve bir hiyerarşik katmanda bulunan tüm gruplar için aynı algoritmalar uygulanmaktadır. Dolayısıyla bu tip bir sisteme dahil olan her düğüm, bulunduğu hiyerarşik katmandan bağımsız olarak aynı yazılımı çalıştırmaktadır.

5.1.1 Liderin grup içindeki videoları belirlemesi

Bir düğüm, sisteme ilk kez giriş yaparak bir grup içine dahil olduğunda, elinde bulunan videoların listesini grup liderine bildirir. Grup liderleri, en alt hiyerarşik katmanda grup içerisinde bulunan videoların listesini ve bu videoların hangi grup elemanlarında bulunduğunu sürekli olarak günceller.

Grup video listeleri, grup bölünme ve birleşmeleri sırasında gruba yeni katılan ya da gruptan ayrılan düğümlerin elinde bulunan videolar gözönünde bulundurularak güncellenmektedir.

Tasarlanan yapıya göre, sistemdeki belli düğümler tarafından talep edilen bir video, sistemde kayıtlı bir ya da daha fazla kaynakta bulunabilir. Sisteme kayıtlı bazı düğümlerde hiçbir video bulunmayabilir.

5.1.2 Düğümlerin video talebinde bulunması

Sistemdeki düğümler, bir gruba dahil olduktan bir süre sonra, sahip olmadığı bir videoyu izleme talebinde bulunur ve bunu grup liderine bir mesajla bildirir. Grup lideri belirli bir zaman aralığında gelen talepleri alır.

Eğer talep edilen video grup içerisinde bulunuyorsa, kaynak düğümlerin adresleri grup liderinde bulunduğu için (en alt hiyerarşik katmanda) kaynak düğüm (ya da düğümler) ile bu videoyu talep eden düğümlerin listelerini birbirlerine gönderir ve bu aşamadan sonra akışlandırmanın yapılmasını sağlayacak gerekli akışlandırma yolunun kurulması için gerekli işlemler başlatılır. Akışlandırma yolunun belirlenmesi ile ilgili işlemlere bir sonraki bölümde değinilmiştir.

Talep edilen video, grup içerisinde bulunmuyorsa, lider video arama algoritmasını çalıştırmaya başlar. Eğer aranan video sistemde bulunursa; lider, grubunda bulunan diğer düğümlere bu videonun akışlandırılmaya başlayacağını haber veren bir mesaj gönderir. Bu mesajı alan düğümler, videoyu izlemek istiyorsa ve daha önce talepte bulunmamışsa, bu talebini lidere bildirir ve akışlandırma yolu üzerinde konumlandırılmayı bekler.

5.1.3 Videoların aranması

Lider, talep edilen bir videonun kendi grubu içerisinde olmadığını tespit ettiğinde, bir üst hiyerarşik katmana çıkarak buradaki düğümlere aradığı videoyu mesajla bildirir. Bir hiyerarşi katmanında video arama mesajı alan düğüm, lideri olduğu bir alt hiyerarşik katmandaki ($h-1$) grubuna bu videonun arandığını bildiren bir mesaj gönderir. Video arama mesajı bu şekilde katmanlar arasında grup liderleri arasında dolaşarak aranır. Bir grup içerisinde aranan video bulunduğu zaman, bu grubun lideri videonun bulunduğu dair mesajı bir üst hiyerarşik katmandaki grup liderine gönderir.

Videoyu talep eden grup lideri, videonun bulunduğu dair mesajı aldığı anda video arama algoritması sonlanmış olur. Bu aşamadan sonra videonun bulunduğu en üst hiyerarşik katmandan aşağıya doğru inilerek akışlandırma yolu oluşturma algoritması (çoklugoönderim ağaç oluşturma algoritması) çalıştırılır.

Video talebi ve arama algoritmasının sözkodları Ek.4'te verilmiştir.

5.2 Akışlandırma Liderleri

Akışlandırma liderleri, akışlandırma protokolünde lider olarak görev alan ve ana görevi bulunduğu gruba video taşımak olan düğümlerdir. Akışlandırma lideri elinde videoyu barındıran kaynak konumunda ya da istemci konumunda olan bir düğüm olabilir. Ancak akışlandırma liderleri ağacın kökünde ya da köküne yakın konumda bulunacağı için, yüksek yükleme kapasitesine sahip düğümlerdir. Akışlandırma liderleri, videonun talep edildiği en üst katmandan en alt hiyerarşik katmana kadar, içerisinde kaynak ya da istemci düğüm konumunda düğümlerin bulunduğu her grupta seçilir.

Çoklugoönderim ağaç oluşturma algoritması temel olarak belli sayıda düğümün bir grup düğüm arasından çocuk seçimine bağlı olarak çalışmaktadır. Aynı düğüm kümesi arasında çocuk seçimlerinin yapılması, bu düğümleri seçecek düğümler tarafından bir uzlaşma sağlanmasını gerektirmektedir; bu yüzden tasarlanan algoritma uzlaşma tabanlı çoklugoönderim ağaç oluşturma algoritması olarak adlandırılmıştır. Uzlaşma tabanlı çoklugoönderim ağaç oluşturma algoritmasının çalıştırılabilmesi için, ağaç üzerinde yer alacak düğümlerin diğer düğümler ile arasındaki gecikme ve diğer düğümlerin beyan ettikleri bant aralığı değerlerini elde etmesi gerekmektedir. Bunun için video talepleri grup liderleri

tarafından toplanarak istemci ve kaynak düğümlerin listeleri tamamlandıktan sonra, en üst hiyerarşik katmandaki grup lideri (videonun talep edildiği en üst hiyerarşik katman) aynı katmandaki video istemcisi ya da videoyu kaynağı konumundaki düğümlere gerekli ölçümlerin başlaması ile ilgili bir mesaj gönderir. Bu mesajı alan düğümler, bir alt hiyerarşik katmanda grup liderleridir, ve kendileri istemci ya da kaynak düğüm olmasa bile, kendi gruplarında bulunan istemci ya da kaynak düğümleri temsilen bir üst katmanda bu konumda değerlendirilmiştir. Eğer bir grup içerisinde kaynak düğüm bulunuyor ise, bu grubun lideri bir üst katmana kaynak düğüm olarak; eğer grup içerisinde istemci düğümler bulunuyor ve kaynak düğüm bulunmuyor ise bu grubun lideri bir üst katmana istemci düğüm olarak çıkar. Ölçümlerin başlaması ile ilgili mesajı alan grup liderleri, bu mesajı kendi alt grubunda bulunan istemci ya da kaynak düğümlere iletir. En alt hiyerarşik katmana ulaşıldığında, bu mesajı alan düğümler orijinal istemci ya da kaynak düğümlerdir; mesajı aldıktan sonra aynı grup içerisindeki istemci ve kaynak düğümler ile arasında gecikme değerlerini hesaplar ve kendi bantgenişliği değeri ile birlikte, hesapladığı gecikme değerlerinin en büyüğünü liderine gönderir.

Grup liderleri akışlandırma başlatılacak olan video için istemci ya da kaynak listesinde bulunmuyor ise ağaç üzerinde de bulunmayacaktır. Bu yüzden grup lideri, istemci ve kaynak düğümlerden en yüksek gecikme ve bant aralığı değerlerini aldıktan sonra, bu düğümler arasında en yüksek bant aralığı beyan eden ve diğer düğümlere olan maksimum gecikme değeri en düşük olan düğümü akışlandırma lideri olarak seçer. Maksimum gecikme değeri en düşük olan düğümün akışlandırma lideri olarak seçilmesinin sebebi, akışlandırma başladığı zaman bu düğümün bağlı olduğu grupta kaynak görevi görecektir; başka bir deyişle akışlandırma liderleri kendi gruplarına videoyu taşımak ile görevlidir. Bu yüzden gruptaki istemciler arasında, en yüksek yükleme hızına sahip düğümler arasından mantıksal olarak merkezde bulunan düğümün seçilmesi avantajlı olacaktır. Akışlandırma lideri seçilen düğüm, kendi grubunu temsilen bir üst hiyerarşik katmana çıkar. Bir hiyerarşik katmanda seçilen akışlandırma liderleri arasından aynı ölçüm kriteri ile yeni bir akışlandırma lideri seçilir ve bu seçimler, video talebinin yapıldığı en üst katmana gelindiğinde sonlanır.

Teorem 5.1: En kötü durum senaryosunda, grupların eleman sayısı k ve sistemdeki toplam düğüm sayısı N olduğu durumda, akışlandırma liderleri seçimi mesaj karmaşıklığı $O(kN)$ ile sınırlıdır.

İspat 5.1: En kötü durum senaryosuna göre, sistemdeki tüm düğümlerin aynı videoyu yakın zamanda izlemeyi talep ettiği varsayılabilir. Her grup içinde tüm düğümler çoklugoönderim ağacının bir parçası olacağı için düğümü oluşturan k düğümün aralarındaki gecikmeyi hesaplayabilmesi k^2 mesaj değişimi gerektirir.

Sistemdeki tüm videolar video talebinde bulunduğu için, akışlandırma liderleri seçimi $\log_k(N/k)$ adet hiyerarşik katmanda bulunan tüm gruplar içerisinde yapılır. Hiyerarşik katman 0'dan hiyerarşik katman $\log_k(N/k)$ 'ya kadar $k^{\log_k N/k}, \dots, k^2, k, 1$ adet grup bulunmaktadır. Her grupta $k*(k-1)$ mesajlaşma yapıldığı için tüm sistemde,

$$k^{\log_k N/k} * k*(k-1) + \dots + k * k*(k-1) + 1 * k*(k-1) \quad (\text{Eş.5.1})$$

adet mesajlaşma yapılır. Eşitlik 5.1 yeniden düzenlenirse,

$$\begin{aligned} k^2 * (k^{\log_k N/k} + \dots + k + 1) &= k*(k-1) \sum_{i=0}^{\log_k N/k} k^i \\ &= k*(k-1) \left(\frac{k^{\log_k N/k+1} - 1}{k - 1} \right) \end{aligned} \quad (\text{Eş.5.2})$$

elde edilir. Bu eşitlik, $O(k^{\log_k N/k+1}) = O(kN)$ karmaşıklığını verir.

5.3 Çoklugoönderim Ağaçları Üzerinde Hata Dayanıklılığı

Görevdeş sistemlerde, düğümlerin sistemden ayrılma oranı oldukça yüksektir ve düğüm hareketliliği bu tip sistemlerde çalışan gerçek zamanlı video akışlandırma uygulamaları için aşılması gereken en önemli problemlerden biridir. Kapasite farkındalıklı çoklugoönderim ağaçlarında hata dayanıklılığının sağlanması için, literatürde de bu amaçla önerilen yedek ebeveynler kullanılmıştır. Ancak önerilen yedek ebeveyn metodu, literatürdeki benzerlerinden farklılık göstermektedir. Bu farklılığın hata dayanıklılığına sağladığı olumlu kazanım, tezin test bölümünde gösterilecektir.

Hata dayanıklılığı için önerilen yöntem, proaktif ve reaktif bileşenler içermektedir, bu özellik tasarlanan yöntemi literatürdeki benzerlerinden ayırmaktadır. Önerilen yöntemin proaktif bileşeni kapsamında, hata henüz gerçekleşmeden hata olasılığına karşı önlem alınmaktadır. Çoklugoönderim ağaç oluşturulması sırasında seçilen yedek ebeveynler, düğümlerin ebeveyninde hata

```

1   K = { k1, k2, ..., kn }
2   L = { l1, l2, ..., lm }
3   Ls = ∅
4   katman = 0; //ağacın katmanı
5   while L ≠ ∅ do
6       n = |K|;
7       K kümesini kaynakların kapasitesine göre büyükten
        küçüğe sırala;
8       gerekli_yedek_hücre_sayısı = kmaks = k(k1);
9       for i=1 to n
10          uygun_hücre_sayısı=uygun_hücre_sayısı+k(ki)*0.5;
11      end for
12      if (uygun_hücre_sayısı<gerekli_yedek_hücre_sayısı)
13          gerekli_yedek_hücre_sayısı = uygun_hücre_sayısı;
14      end if
15      Y = Fırsatçı ya da Çevrimsel Sıralı yöntemine göre
        seçim*;
16      while (  $\sum_{i=1}^n d(k_i) < \sum_{i=1}^n k(k_i)$  ) && (L ≠ ∅) do
17          a ∈ K ve b ∈ L olan ea,b minimum maliyete
            sahip kenarı seç;
18          b'nin yedek ebeveynleri olarak
            Ykatman = Y ∪ Ykatman-1 ∪ {a'nın ebeveyni} ata;
19          Ls = Ls ∪ {b};
20          L = L - {b};
21          da = da + 1;
22          if (da==ka) then K = K - {a} end if
23      end while
24      K = Ls;
25      Ls = ∅;
26      katman = katman + 1;
27  end while

```

* Gerekli rutinler sonraki iki şekilde verilmiştir.

Şekil 5.1. Yedek ebeveyn seçimi ile birlikte çoklוגönderim ağaç oluşturma algoritması.

olması durumunda video akışlandırma oturumunu sürdürebilmek amacıyla bağlanacakları düğümlerdir. Dolayısıyla, ağaç üzerinde bulunan her düğüm için henüz ağaç oluşumu sırasında belirlenen bu düğümler, hata oluşmadan önce

belirlenmiş bir hata kurtarma politikası oluşturduğu için, tasarlanan hata dayanıklılığı yönteminin proaktif bileşenini temsil eder.

Yedek ebeveyn görevinde bulunan bir düğüm, kapasitesinin bir kısmını yedek hücreleri olarak, bir kısmını da çocuk bağlamak için servis hücreleri olarak ayırmaktadır. Yapılan testler sonucu, bir düğümün servis ve yedek hücreleri oranının %50 seçilmesi durumunda, bantgenişliği kullanımı, ağaç yüksekliği ve hata durumunda ağacın yeniden yapılandırılması açısından en iyi kazanımı sağladığı tespit edilmiştir. Dolayısıyla, bir düğümün kapasitesinin %50'sinden fazlası yedek olarak kullanılamaz.

Yedek ebeveynler, çokluguönderim ağacının her seviyesini oluşturan düğümler arasından seçilmekte ve ağacın bir alt seviyesindeki düğümler için yedek oluşturmaktadırlar. Yedek ebeveynlerin seçimi Fırsatçı (Greedy) ve Çevrimsel Sıralı (Round-Robin) olmak üzere 2 farklı yöntemle yapılmaktadır. Yedek ebeveyn seçim algoritması Şekil 5.1'de verilmiştir. Seçimler, ağaç oluşumu aşamasında yapıldığı için Şekil 5.1'de verilen algoritma, çokluguönderim ağaç oluşturma algoritmasına (Bkz. Şekil 4.2) yedek ebeveynlerin belirlenme yönteminin eklenmesiyle oluşturulmuştur.

Şekil 5.1'de verilen algoritmaya göre, kaynaklar K ve istemciler L kümesine atıldıktan sonra ağacın ilk katmanı (sıfırıncı katman) oluşturulmaya başlanır. Algoritmanın 7. satırında, kaynak kümesini oluşturan düğümler kapasitelerine göre büyükten küçüğe sıralanır. Ağacın $h+1$. katmanındaki düğümler için seçilecek yedek hücre sayısı, ağacın h . katmanında bulunan en yüksek kapasiteli düğümün alabileceği maksimum çocuk sayısı kadardır. Dolayısıyla, kapasiteye göre büyükten küçüğe sıralanmış olan K kümesinde ilk elemanın (ilk düğümün) kapasitesi, seçilmesi gerekli yedek hücre sayısını belirler. Algoritmanın 8. satırında bu işlem yapılmaktadır; $k(k_i)$, k_i . düğümün kapasitesini ifade etmektedir. Her düğüm, kapasitesinin en fazla %50'sini yedek hücre olarak ayırabileceği için her zaman seçilmesi gerekli yedek hücre sayısına ulaşamayabilir. Algoritmanın 9.-11. satırları arasında yer alan *for* döngüsünde K kümesinden elde edilebilecek maksimum yedek hücre sayısı, ağacın o katmanını oluşturan düğümlerin kapasitelerinin %50'sinin toplamı olarak hesaplanır. Algoritmanın 12. satırında, bu toplam gözden geçirilir ve eğer önceden belirlenen gerekli yedek hücre sayısı bulunamayacak ise, gerekli yedek hücre sayısı ağacın o katmanında elde edilebilecek yedek hücre sayısına eşitlenir. Yedek ebeveyn seçimleri Fırsatçı ya da Çevrimsel Sıralı yöntemlerinden biri ile seçilir ve Y kümesine atılır; bu

algoritmaların ayrıntıları Şekil 5.2 ve Şekil 5.3'te verilmiştir. Algoritmanın 16. ve 17. satırında çocuk seçimleri çokluguönderim ağaç oluşturma algoritmasında verilen ölçütlere göre yapılır. Çocuk olarak seçilen düğüm b ve bu düğümün ebeveyni a ile ifade edilirse, b 'nin yedek ebeveynleri

$$Y_{katman} = Y \cup Y_{katman-1} \cup \{a'nın\ ebeveyni\} \quad (Eş. 5.3)$$

kümesini oluşturan düğümlerden meydana gelir. Burada $Y_{katman-1}$ ağacın bir üst katmanı için seçilen yedek ebeveynleri belirtir. Dolayısıyla bir düğümün yedek ebeveyn listesi, ağacın o katmanı için seçilen yedek ebeveynler, daha üst katmanlar için seçilen yedek ebeveynler ve o düğümün ebeveyninin ebeveyninden oluşur. Yedek ebeveyn seçiminden sonra algoritma, çokluguönderim ağaç oluşturma esasları gözönüne alarak tamamlanır.

```
Girdi: GYHS(gerekli_yedek_hücre_sayısı)
      K= {k1, ..., kn} //Kapasiteye göre büyükten küçüğe
      sıralanmış ağacın o katmanını oluşturacak düğüm listesi
1      i = 1

2      while GYHS>0 do
3          if (GYHS > k(ki)*0.5)
4              GYHS = GYHS -k(ki)*0.5;
5              d(ki) = k(ki)*0.5;
6          else
7              GYHS = 0;
8              d(ki) = GYHS;
9          end if
10         Y = Y ∪ {ki};
11         i = i + 1;
12     end while
13     return Y;
```

Şekil 5.2. Fırsatçı yöntemine göre yedek ebeveyn seçim algoritması.

Şekil 5.2'de Fırsatçı algoritma verilmiştir. Algoritma parametre olarak çokluguönderim ağaç oluşturma algoritması içerisinde (Bkz. Şekil 5.1) belirlenen gerekli yedek hücre sayısını almaktadır. Algoritmanın 2. satırında başlayan *while* döngüsü gerekli yedek hücre sayısı kadar hücre seçilene kadar çalışır.

Algoritmanın 3. satırına göre, gerekli yedek hücre sayısı K sıralı kümesinde k_i düğümün ayırabileceği yedek hücre sayısından büyük ise, bu düğümün yedek hücreleri seçilerek gerekli yedek hücre sayısından çıkarılır (4. satır); k_i düğümünün derecesi kendisinden seçilen yedek hücre sayısına eşitlenir (8. satır). Eğer gerekli yedek hücre sayısı k_i düğümünün ayırabileceği yedek hücre sayısından küçükse, k_i düğümünden gereken yedek hücre sayısı alınır (7. ve 8. satır). 10. satırda k_i düğümü yedek ebeveyn listesine eklenir. Algoritma, çıktı olarak yedek ebeveyn listesini (Y) döndürür.

```

Girdi: GYHS(gerekli_yedek_hücre_sayısı)
        K= { $k_1, \dots, k_n$ } //Kapasiteye göre büyükten küçüğe
        sıralanmış ağacın o katmanını oluşturacak düğüm listesi
1      i = 0
2      n = |K|;

3      while GYHS>0 do
4          i = (i+1) mod n;
5          if ( $d(k_i) < k(k_i) * 0.5$ )
6              GYHS = GYHS - 1;
7               $d(k_i) = d(k_i) + 1$ ;
8               $Y = Y \cup \{k_i\}$ ;
9          end if
10     end while
11     return Y;

```

Şekil 5.3 Çevrimsel sıralı yöntemine göre yedek ebeveyn seçim algoritması

Şekil 5.3'te Çevrimsel Sıralı yöntemine göre yedek ebeveyn seçim algoritması verilmiştir. Algoritma Fırsatçı algoritmasına benzer şekilde (Bkz. Şekil 5.2) girdi olarak gerekli yedek hücre sayısını almaktadır ve algoritmanın 3. satırında başlayan *while* döngüsü gerekli yedek hücre sayısı kadar hücre seçilene kadar çalışmaktadır. Algoritmanın 4. satırında çevrimsel olarak ağacın o katmanını oluşturacak düğümler arasında bir sonraki k_i düğümüne geçilir. Eğer k_i düğümünden daha önce ayrılan yedek hücreler düğümün kapasitesinin %50'sini aşmadıysa (5. satır) k_i düğümünden 1 hücre daha yedek hücre olarak ayrılır ve yedek ebeveyn listesi güncellenir (6.- 8. satırlar). Gerekli yedek hücre sayısı seçildikten sonra çıktı olarak yedek ebeveyn listesi (Y) döndürülür.

Hem Fırsatçı hem de Çevrimsel Sıralı yöntemi ile seçilen yedek hücreler, verilen algoritmalarda görüldüğü üzere yedek ebeveyn listelerinin oluşturulmasını ve ağaç üzerindeki düğümlere gönderilmesini amaçlar. Bir düğüme birden fazla sayıda yedek ebeveyn ataması literatürde bulunan her düğüm için tek bir yedek belirleme çalışmalarından ayrılmaktadır ve önerilen yapının daha iyi hata dayanıklılığı sağladığı yapılan testler ile gösterilmiştir.

5.4 Hiyerarşik Katmanlar Üzerinde Çoklugönderim Ağaçlarının Oluşturulması

Tasarlanan çoklugönderim ağacı, daha önceki bölümlerde verilen algoritmalarda, merkezi bir biçimde oluşturulmaktadır. Tezin bu bölümünde, görevdeş ağ mimarisi üzerinde tasarlanan hiyerarşik katmanlar üzerinde çoklugönderim ağaç oluşturma algoritmasının nasıl oluşturulduğu anlatılacaktır.

Akışlandırma liderlerinin seçimi en üst hiyerarşik katman üzerinde sonlanır (Bkz. Bölüm 5.2). Akışlandırma liderleri en üst hiyerarşik katman üzerinde belirlendiğinde, en üst hiyerarşik katmandaki grup lideri, elinde videoyu bulunduran akışlandırma liderlerine çoklugönderim ağacını oluşturmaya başlamaları üzerine mesaj gönderir, mesajın içerisinde kaynak konumundaki akışlandırma liderlerinin adres listesi de bulunmaktadır. Bu mesajı alan akışlandırma liderleri, birbirlerine kendi yükleme hızı ve diğer düğümler ile aralarında gecikmeyi belirten ağırlık değerleri tablolarını gönderir. Bu tablolar yardımı ile tüm akışlandırma liderleri aynı çoklugönderim ağaç oluşturma algoritmasını çalıştırır (Bkz. Şekil 5.1) ve çocuk seçimleri en üst katmanda gerçekleştirilir. Seçilen çocuklar, ağacın bir alt katmanını oluşturur ve eğer bu grupta ağaca bağlanması gereken düğüm var ise, çocuk seçimleri devam eder. En üst hiyerarşik katmanda yer alan grupta çoklugönderim ağacı oluşturulduktan sonra akışlandırma liderleri bir alt hiyerarşik katmanda temsil ettikleri gruba geçer ve eğer burada kaynak durumunda bir düğüm bulunuyorsa onunla beraber, bulunmuyorsa tek başına çoklugönderim ağaç oluşturma algoritmasını çalıştırır. Bu şekilde en alt hiyerarşik katmana kadar inildikten sonra çoklugönderim ağaç oluşumu sonlandırılır. Algoritmanın yapısı gereği, ağacın her parçası farklı hiyerarşik katmanlarda oluşmaktadır.

Şekil 5.4'te 2 hiyerarşik katman üzerinde oluşmuş 5 grup içerisinde kontrol liderleri ile akışlandırma liderleri arasındaki ilişki ve oluşturulan örnek bir çoklugönderim ağacı gösterilmektedir. Birinci hiyerarşik katmanda yer alan her

gönderir. Ağacın 2. katmanında, x_1 düğüm toplam x_2 düğümü çocuk olarak seçer. Bu süreç, tüm düğümler ağaca yerleşene kadar devam eder. Bu durumda gönderilen toplam mesaj sayısı M ile ifade edilirse eşitlik 5.4 elde edilir.

$$M = x_1 + x_1(x_1 - 1) + x_2 + x_2(x_2 - 1) + \dots + x_{h-1}(x_{h-1} - 1) + x_h \quad (\text{Eş. 5.4})$$

Eşitlik 5.4 düzenlendiğinde,

$$M = \sum_{i=1}^{h-1} x_i^2 + x_h \quad (\text{Eşitlik 5.5})$$

elde edilir.

Tüm düğümlerin kapasitesinin 2 varsayıldığında, oluşan ikili ağacın her i . katmanında 2^i düğüm bulunur. Bu durumda eşitlik 5.5 aşağıdaki gibi yeniden yazılabilir.

$$M = \sum_{i=1}^{h-1} (2^i)^2 + 2^h$$

$$M = ((4^h - 1)/3) - 1 + 2^h \quad (\text{Eşitlik 5.6})$$

Her grupta M mesajlaşma yapılır:

$$\begin{aligned} M \cdot (1 + k + k^2 + \dots + \log_k(N/k)) &= M \cdot \sum_{i=0}^{\log_k N / k} k^i \\ &= M \cdot \frac{k^{\log_k N / k + 1} - 1}{k - 1} \quad (\text{Eşitlik 5.7}) \end{aligned}$$

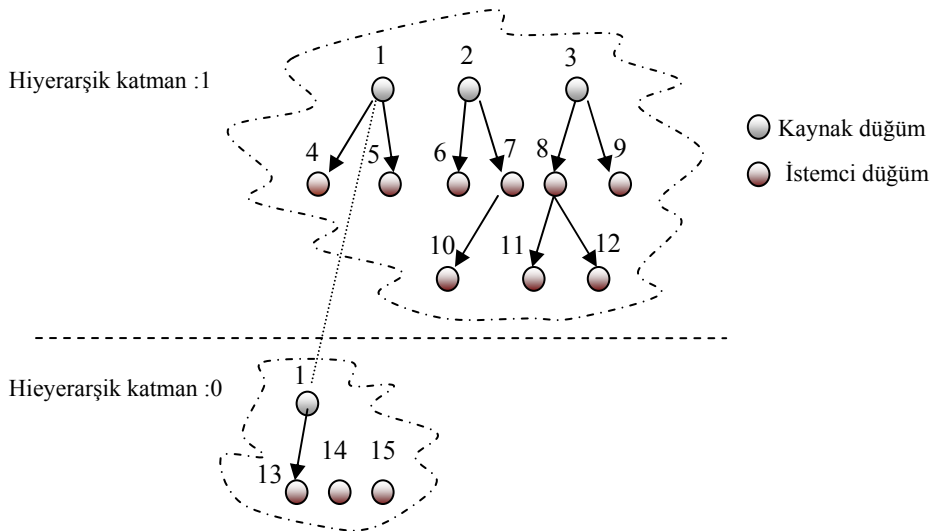
Bu durumda mesaj karmaşıklığı $O(4^{\log_2 k} k^{\log_k N / k}) = O(kN)$ ile sınırlanır.

5.5 Düğüm Kapasitelerinin Hiyerarşik Katmanlar Üzerinde Dağıtılması

Çokluguönderim ağaçlarının oluşturulmasında düğümler yükleme hızlarına göre bir başka deyişle kapasitelerine göre ağacın üst kısımlarında yer alırlar. Yüksek kapasiteye sahip düğümler, akışlandırma lideri seçilerek bir ya da birden fazla hiyerarşik katmanda yer alabilir. Bir akışlandırma liderleri, kapasitesini, bulunduğu tüm hiyerarşik katmanlardaki çocukları ve yedek hücreleri arasında

paylaştırmalıdır. Bu paylaşım, diğer akışlandırma liderleri ile uzlaşma içine girerken yapılır.

Bir akışlandırma lideri, çokluguönderim ağacının oluşması sırasında bir hiyerarşik katman üzerinde ağacın o katmana ait parçası tamamlandığında bir alt katmana geçerek burada temsil ettiği gruba video taşımak amacıyla kendisine en az 1 çocuk bağlamalıdır. Dolayısıyla kapasitesinden birer hücre alt hiyerarşik katmanlar için servis hücresi olarak ayrılmalıdır. Örnek olarak 2. hiyerarşik katmanda yer alan bir akışlandırma lideri, sıfırıncı ve birinci hiyerarşik katmanlar için kapasitesinden toplam 2 servis hücresi ayırır. Daha sonra çokluguönderim ağaç oluşturma algoritması çalıştırıldığında, kapasitesinin geri kalanına çocuk bağlanabilir ya da kapasitenin bir kısmı toplam kapasitesinin %50'sini geçmemek şartı ile yedek hücre olarak ayrılabilir. Akışlandırma lideri, bir alt hiyerarşik katmana geçerken, üst hiyerarşik katmanda kendisine bağlanan çocuk ve yedek hücre sayısını kapasitesinden ayırdıktan sonra geri kalan kapasiteyi bir alt hiyerarşik katmana taşır. Bir alt hiyerarşik katman için önceden ayırdığı 1 servis hücresini de ekleyerek çocuk seçimi yapar ve üst katmanda ayırdığı yedek hücre sayısı dahil olmak üzere kapasitesinin %50'sini geçmeden yedek hücre ayırabilir. Tüm akışlandırma liderleri benzer şekilde üst hiyerarşik katmanda kullanımdan sonra kalan kapasitelerini alt hiyerarşik katmana taşıyarak çokluguönderim ağaç oluşturma algoritmasını tamamlar. En üst hiyerarşik katmanda düğümlerin kapasitelerinin kullanıma tamamen açık olması, ağacın üst katmanlarında daha fazla sayıda yedek ebeveyn ve yedek hücre olmasına yardımcı olur; bu yaklaşım ağaç hata dayanıklılığını arttırmaktadır. Diğer yandan, üst katmanlarda olabilecek en fazla sayıda çocuk alınması ağaçların daha kısa olmasını sağlamaktadır.



Şekil 5.5. Hiyerarşik katmanlar üzerinde çokluguönderim ağaç örneği.

Şekil 5.5'te, hiyerarşik katmanlar üzerinde oluşan bir çokluguönderim ağacında düğüm kapasiteleri gözönüne alınarak oluşturulan bir ağaç örneği verilmiştir. Birinci. hiyerarşik katmanda yer alan 12 düğüm, alt katmanda yer alan 12 farklı grup içerisinde seçilen akışlandırma liderleridir. Uzlaşma tabanlı algoritmaya göre, kaynak düğümler kendilerine en yakın konumda bulunan düğümleri çocuk olarak seçerek çokluguönderim ağaç oluşturma algoritmasını başlatır. $k(i)$, i . düğümün kapasitesini temsil ediyorsa, $k(1) = k(2) = 5$ ve $k(3) = 4$ olduğu varsayılın. Kaynak düğümler sıfıncı hiyerarşik katmanda en az 1 çocuk almak zorunda oldukları için kapasitelerinden 1'er tane hücreyi alt katmana ayırır. Bundan sonraki adım, ağacın bir alt katmanı için yedek hücre sayısının (*GYHS*) belirlenmesidir. Bu sayı, 1, 2 ve 3 numaralı düğümler arasında en yüksek kapasiteli düğümün kapasitesine, 5'e eşittir. Yedek hücreler Çevrimsel Sıralı yaklaşıma göre seçildiğinde düğüm 1, 2 ve 3 sırasıyla 2, 2 ve 1 hücreyi yedek olarak ayırır. Bu durumda yedek ebeveyn listesi $Y_{katman=1} = \{1,2,3\}$ olarak belirlenir. Bu liste, düğümlerin seçtiği çocuklara yedek ebeveynler olarak gönderilir ancak her düğüm, yedek ebeveyn listesinden kendi adını çıkararak listeyi yeniden düzenler ve çocuklarına gönderir (Bir düğümün ebeveyni ile yedek ebeveyni aynı olamaz). 1, 2 ve 3 numaralı düğümlerin çocuk seçiminin Şekil 5.5'te verildiği şekilde yapıldığı varsayılın. Ağacın bir sonraki katmanında 4, 5, ... , 9 düğümleri yer almaktadır. Bu düğümlerde de benzer şekilde 1'er hücreleri bir alt katman için ayrıldıktan sonra, yedek hücre seçimleri yapılır. Ağacın bu katmanın uzlaşmaya giren tüm düğümlerin kapasitelerinin 4 olduğu varsayılın. Bu durumda ağacın bir alt katmanı için seçilecek yedek hücre sayısı $GYHS = 4$ olarak belirlenir ve yeni yedek ebeveyn listesi $Y_{katman=2} = \{4, 5, 6, 7\}$ olarak seçilebilir. $Y_{katman=2}$ yedek ebeveyn listesi, listeye gerekli ebeveyn ve üst katman yedek ebeveyn adresleri de eklenerek ağacın bir sonraki katmanındaki çocuklara (10, 11 ve 12 numaralı düğümler) gönderilir. Örneğin 10 numaralı düğümün alacağı yedek ebeveyn listesi $\{4, 5, 6, 1, 2, 3\}$ düğümlerinden oluşmaktadır. Çokluguönderim ağacı birinci hiyerarşik katmanda oluşturulduktan sonra, her düğüm birinci hiyerarşik katmanda kullandığı kapasite bilgisini de taşıyarak sıfıncı katmana geçer ve kendi grubu içerisinde çokluguönderim ağaç oluşturma algoritmasını tekrar çalıştırır. Şekilde verilen örneğe göre, $k(2) = 5$ kapasitesine sahip 1 numaralı düğüm bir üst katmanda 2 çocuk ve 2 yedek hücre seçimi yaptığı için en alt hiyerarşik katmanda kendisine 1 çocuk bağlar ve bu çocuğa $\{2, 3\}$ yedek ebeveyn listesini gönderir. $k(7) = 4$ kapasitesine sahip 7 numaralı düğüm, 1 yedek hücre ayırdığı ve 1 çocuk aldığı için 0. hiyerarşik katmana artan 1 hücrelik kapasitesini de aktarır ve bu katman için önceden ayrılan 1 hücreyi de ekleyerek kendisine 2 çocuk alabilir, ya da 1 çocuk bağlar ve 1 yedek hücre ayırabilir.

Önerilen yedek ebeveyn seçim algoritması ile verilen örnekte de görüldüğü üzere, düğümler başka kaynaklar tarafından oluşturulan ağaçlar üzerinde yedek ebeveyn seçebilmektedir. Böylece, sistemde birden fazla kaynak olduğu durumda kaynakların kök durumunda bulunduğu ağaç üzerindeki düğümler birbirleri arasında yedek ebeveyn bağlantıları oluşturur; bu durum kurulan sistemin hata dayanıklılığını arttırmaktadır.

6. GÖREVDEŞ AĞ ÜZERİNDE VIDEO AKIŞLANDIRMA OTURUMU

Bir önceki bölümde yer alan dayanıklı ağaç oluşturma algoritması sırasında yedek ebeveyn seçimi, tez kapsamında önerilen hata dayanıklılığı yönteminin proaktif kısmını oluşturmaktadır. Tezin bu bölümünde, akışlandırma oturumu sırasında yedek ebeveynlere bağlanma süreci, önerilen yapının reaktif kısmı anlatılmaktadır. Bu bölümdeki materyalin bir bölümü (Sayıt et al., 2009) da yayınlanmıştır, diğer bölümü ise (Sayıt et al., 2010b)'de verilmiştir.

6.1 Reaktif Bileşen

Her düğüm ağaç oluşumu sırasında ebeveyninden hata durumunda bağlanabileceği yedek ebeveyn listesini alır. Ancak hata durumunda hangi düğüme bağlanacağı belirli değildir. Akışlandırma oturumu başladıktan sonra ağaç üzerindeki bir düğüm belirli bir süre ebeveyninden paket gelmediğini tespit ettiğinde ebeveynine sistemde olup olmadığını sorgulayan bir mesaj gönderir ve zamanlayıcı çalıştırır. Eğer ebeveyninden yanıt alırsa, bu durumda atalarından birinin sistemden ayrıldığını anlar ve sorunu ebeveyninin çözmesini bekler. Eğer düğüm, paket kayıpları tespit ettikten sonra ebeveynini sinyallediğinde yanıt alamazsa yedek ebeveynlere bağlanma algoritmasını çalıştırır. Bu senaryonun dışında, sistemde çok fazla sıkışıklık olduğu durumlarda, düğümün ebeveyni sistemde olsa da video paketleri gelmeyebilir ya da akışlandırmayı devam ettirebilecek düzeyde olmayabilir. Bu durumda düğüm, tampon seviyesine bakarak yedek ebeveynlerden birine bağlanmayı tercih edebilir.

Yedek bir ebeveyne bağlanma kararı veren bir düğüm, yedek ebeveyn listesinde bulunan düğümlere sırası ile bağlanmayı dener. Bir düğümün yedek ebeveyn listesinde yer alan ilk düğüm, o düğümün ebeveyninin ebeveynidir. Ebeveyn hatası durumunda yedek ebeveyne bağlanmaya çalışan tüm kardeş düğümler ilk olarak ebeveyn ebeveynine bağlanmaya çalışır ve bu düğüme ilk kez ulaşan, en yakın düğüm olarak bağlanır. Bir düğüm, bağlanma talep ettiği yedek ebeveyn tarafından reddedilirse listede bir sonraki düğüme geçerek sıradaki yedek ebeveyne bağlanmayı dener. Yedek ebeveyn listesindeki düğümler, ağacın katmanlarına göre sıralıdır. Düğüm, ilk olarak kendisinin bulunduğu katmandan bir üst katmanda bulunan yedek ebeveynlere bağlanmayı dener, bağlantıda başarılı olamazsa ağacın üst katmanlarına doğru ilerler. Bu şekilde, kapasitesi düşük olduğu için ağacın alt kısımlarına yerleştirilen düğümlerin, ağaç üzerinde daha üst katmanlara çıkması engellenir.

Her düğüm, video akışlandırma oturumu başladıktan sonra kısa bir süre tamponunu doldurmaktadır. Tampondaki veri, ebeveyn ile bağlantı kesildiğinde yeni bir ebeveyne bağlanana kadar video akışlandırmanın kesintisiz devamını sağlamaktadır. Ancak düğümlerin tamponunda saklanan veri ne kadar büyükse başlangıç bekleme zamanı o ölçüde artacaktır. Bu yüzden düğümlerin tamponunda saklanan video verisi belli bir sınırdan tutulmaktadır. Yedek ebeveyn arama sürecinde birden fazla ebeveyne başvuru sebebiyle geçen süre içerisinde tampondaki veri tükenebilir. Video akışlandırmanın kesilmesini engellemek için tampon verisi tüketilmeden yeni bir ebeveyne bağlanılmalıdır. Tampon için gerekli zaman kısıtına karşı, yedek ebeveyn arama sürecinde olan düğüm başvurduğu her yedek ebeveyn için bağlantı deneme sayısını (*bds*) günceller ve bu değeri her başvuruda başvurduğu düğüme gönderir. Yedek ebeveyn, başvuran düğüm ile ilgili kararını kendisine gönderilen bağlantı deneme sayısına bakarak verir. α ve β değerleri önceden belirlenmiş eşik değerleri olarak düşünülürse, yedek ebeveynin kendisine yapılan başvuruyu değerlendirme algoritması şekil 6.1'de verilmiştir. Algoritmaya girdi olarak bağlantı deneme sayısı (*bds*) değeri verilir; çıktı olarak ise KABUL ya da RED sonuçlarından biri alınır.

```

Girdi: bds(bağlantı_deneme_sayısı)

1  if (bds <=  $\alpha$ )
2    then yeni bir çocuk destekleyecek kapasite var ise:KABUL,
3        aksi durumda: RED
4  else if ( $\alpha$  < bds <=  $\beta$ )
5    then yeni bir çocuk destekleyecek kapasite var ise:KABUL,
6        (bu durumda hücre sayısı en düşük kaliteye göre hesaplanır)
7        aksi durumda: RED
8  else if (bds >  $\beta$ )
9    then yeni bir çocuk destekleyecek kapasite var ise:KABUL,
10       (bu durumda hücre sayısı en düşük kaliteye göre hesaplanır)
11       aksi durumda: RED (aşağı yönlendirme ile)
12 end if

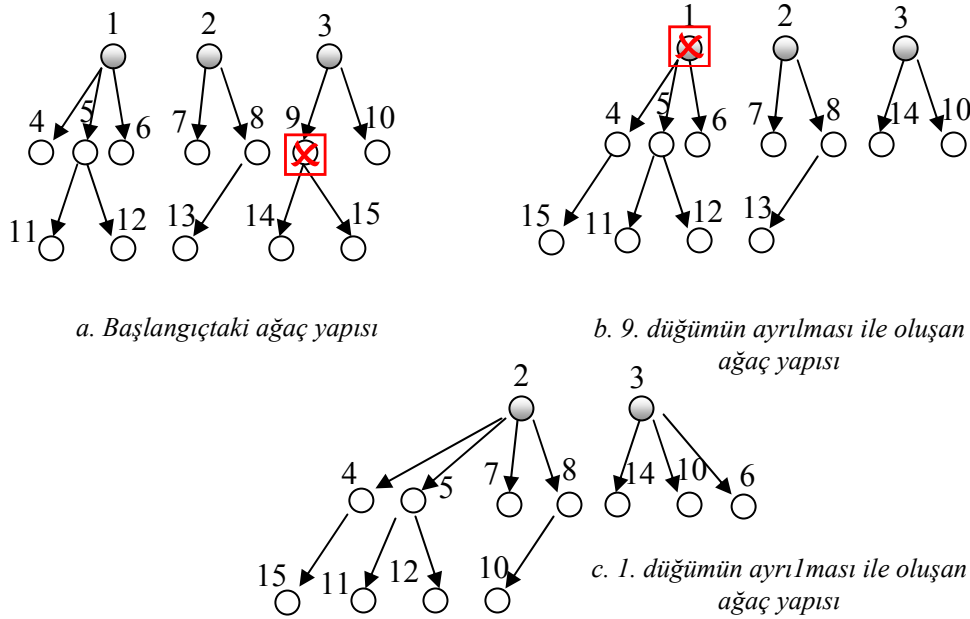
```

Şekil 6.1. Bağlantı kurulum algoritması.

Şekil 6.1’de verilen algoritmanın 9. satırında yer alan aşağı yönlendirme ile RED komutu, başvuruda bulunan düğümün ağacın alt kısımlarına doğru

yönlendirilmesini sağlar. Bu durum, başvuruda bulunan düğümün bir an önce yeni bir ebeveyne bağlanması gerektiğinde gerçekleşir, RED mesajı veren yedek ebeveyn bu mesaj ile kendisine bağlı çocuklardan en fazla kapasiteye sahip çocuğunun adresini verir. Bu durumda kendisine yeni bir ebeveyn arayan düğüm verilen yeni adrese yönelir. Aşağı yönlendirme yöntemi, ebeveyn arayan düğümlerin kısa sürede yeni bir ebeveyn bulma olasılığını artırarak akışlandırmanın kesintisiz devam etmesine olanak sağlar.

Bağlantı kurulum algoritmasına göre, düğümün mümkün olan en kısa sürede yeni bir ebeveyne bağlanabilmesi için bağlantı deneme sayısının β 'ya eşit olması gerekmektedir. Eğer düğümün tamponundaki veri belli bir seviyenin altına düşerse, düğüm bağlantı deneme sayısı değerini β 'ya eşitleyerek kısa sürede ağaca yeniden bağlanmayı garanti altına alır.



Şekil 6.2. Örnek hata senaryosu.

Şekil 6.2'de örnek bir hata senaryosu verilmiştir. Şekil 6.2.a'da ağ yapısının hata olmadan önceki hali verilmiştir. Burada yer alan yapıya göre, ağacın 3. katmanında yer alan 11, 12, ... , 15 numaralı düğümlerin yedek ebeveyn listesinde ebeveyn ebeveyn adresi ile ağacın ikinci katmanında yer alan düğümler arasından seçilen yedek ebeveynler bulunmaktadır. Ağaç üzerinde yer alan 9. düğüm sistemden ayrıldığında 14 ve 15 numaralı düğümler yedek ebeveyn arama algoritmasını çalıştırır. Bu düğümler için yedek ebeveyn listesi 3. düğümün adresi

ile başlamaktadır. Yedek ebeveyn listesinin {3, 4, 5} düğümlerinden oluştuğu varsayılır ise, 3 numaralı düğüme en yakın olan düğüm bu düğüme bağlanır. Verilen senaryoya göre 14. düğüm 3. düğüme ve 15. düğüm listedeki bir sonraki yedek ebeveyne, 4 numaralı düğüme bağlanmıştır. Şekil 6.2.b’de oluşan yeni ağ yapısı verilmektedir. Verilen ağ yapısında 1 numaralı düğüm sistemden ayrıldığında, bu düğümün çocukları olan 4, 5 ve 6 numaralı düğümler, yedek ebeveyn listesinde yer alan düğümlere bağlanır. Şekilde verilen senaryoya göre {2, 3} düğümleri, yedek ebeveyn listesi olarak verilmiştir. Ebeveyninin gitmesi ile ağ bağlantısı kesilen düğümlerden 4 ve 5 numaralı düğümler 2 numaralı düğüme, 6 numaralı düğüm 3 numaralı düğüme bağlanmıştır. Oluşan yeni ağ yapısı şekil 6.2.c’de verilmiştir.

6.2 Kalite Uyarlama

Video akışlandırma uygulamalarında karşılaşılan bir diğer büyük problem grubu, ağ üzerindeki sıkışıklıktan kaynaklanan sorunlardır. Tez kapsamında tasarımlanan sistemde, her düğüm sürekli olarak ebeveyninden aldığı video bithızına ve paket kayıp oranına bakarak kendisi ile ebeveyni arasındaki ağ üzerindeki sıkışıklık durumunu gözlemler. Eğer paket kayıp oranı belli bir seviyenin altına düşerse, ebeveynine bu durumu sinyaller ve ebeveyninin kalite uyarlaması yapması için bekler.

Ağ üzerindeki sıkışıklık durumunda çocuklarından kalite uyarlama üzerine sinyal alan ebeveyn konumundaki bir düğüm aşağıdaki senaryolardan birini uygular:

a. Video kalitesinin düşürülmesi: Düğümler tarafından gönderilen video, ölçeklenebilir video kodlayıcısı ile kodlanmıştır. Ölçeklenebilir video kodlayıcısının en önemli özelliği, farklı bithızlarında kodlanmış video verisi arasındaki geçişleri video katmanları ekleme/çıkarma yöntemi ile kolayca yapabilmesidir. Bir düğümün ölçeklenebilir video kodlayıcısı ile kodlanmış veriyi düzgün bir biçimde çözümleyebilmesi için taban katmana sahip olması yeterlidir. Alıcı düğümler tarafından taban katman üzerine alınan her genişleme katmanı, elde edilen görüntü kalitesini arttırmaktadır.

Video akışlandırma oturumu başladığında her düğüm eğer yükleme hızı uygunsa mümkün olan en üst kalitede video verisini çocuklarına gönderir. Video akışlandırma oturumu sırasında çocuk ya da çocuklarından sıkışıklık olduğuna

dair sinyal alan ebeveyn konumundaki bir düğüm, ağ üzerindeki sıkışıklığa önlem almak için ilk olarak gönderilen videonun kalitesi düşürür. Video kalitesini düşürme işlemi, genişleme katmanlarının çıkarılması ile sağlanır.

b. Bağlantı kesilmesi: Gönderilen videonun genişleme katmanlarının çıkarılması ağ üzerindeki sıkışıklık sorununu gidermiyor ise ebeveyn konumundaki düğüm çocuklarından birini kurban seçerek seçilen çocuk ile bağlantısını koparır. Kurban çocuğun seçiminde çocuklar tarafından gönderilen bağlantı kalitesi ve kayıp paket oranına bağlı olarak seçim yapılır. Bazı durumlarda sıkışıklık tespiti ebeveyn ile sadece bir çocuğu arasında bu iki düğüm arasındaki bağlantıdan kaynaklanıyor olabilir. Bu durumda düğümün diğer çocukları sıkışıklık algılamaz. Sıkışıklık algılayan çocuk konumundaki düğümün bağlantısının kesilmesi, bu düğümün daha iyi bağlantı kurabileceği yeni bir ebeveyne bağlanmasını sağlar. Bağlantısı koparılan çocuk konumundaki düğüm yedek ebeveyn arama algoritmasını çalıştırır. Ebeveyni tarafından haberli bir şekilde bağlantısı koparıldığı ve kardeş konumundaki düğümleri yedek ebeveyn aramadığı için, kısa sürede yeni bir ebeveyne bağlanabilir.

7. YEDEK EBEVEYN SEÇİMLERİNDE ANALİTİK YAKLAŞIM

Tezin beşinci bölümünde önerilen yöntemler ile yedek ebeveyn seçimlerinde hiyerarşik katmanlar üzerinde oluşturulan çoklugoönderim ağaçları için ağacın her katmanında belirli sayıda yedek ebeveyn seçilmektedir. Tezin bu bölümünde, tezin önceki bölümlerinde önerilen Fırsatçı ve Çevrimsel Sıralı yedek ebeveyn seçim yöntemlerinden farklı olarak geliştirilen yedek ebeveyn seçim yaklaşımından söz edilecektir.

Yedek ebeveyn kavramı literatürde çalışılmış olmasına rağmen, sistemde yer alan düğümler için en iyi “yedek hücre – servis hücresi oranı” çalışılmamıştır. Yedek hücre sayısının artırılması düğümlere bağlanacak çocuk sayısını kısıtlayarak ağaç yüksekliğini artırırken, servis hücre sayısının artırılması hata dayanıklılığını düşürmektedir. Dolayısıyla, verilen bir çizgede bulunan düğümler arasında oluşturulacak ağaç, düğümler arası gecikmenin en kısa, bantgenişliği kullanımı ve hata dayanıklılığın en yüksek olacağı şekilde tasarlanmalıdır.

Çoklugoönderim ağacının kurulması sırasında (Sayıt et al, 2009) çalışmasında önerilen, kapasitesi yüksek düğümlerin ağacın üst katmanlarına yerleştirilmesi yaklaşımı uygulanmıştır. Bu bölümde sözedilen hata dayanıklılık yöntemi için, ağacın her katmanında tüm düğümler yedek ebeveyn seçilir ve kapasitelerinin belli bir oranını yedek hücrelere ayırır. Ağacın kökünün sistemden çıkmayacağı farzedilmiştir. Bu bölümde verilen materyal (Tunalı ve Sayıt, 2011) da da verilmiştir.

Önerme 7.1: Sistemdeki düğümlerin kapasitesinin d olduğu farzedilsin. c , $0 < c < d$ olacak şekilde, ağacın kökü dışında tüm düğümlerin sahip olduğu çocuk sayısı olsun. Bu durumda, YH_i olarak ifade edilen, i . düğümün toplam yedek hücre sayısı $c=d/2$ seçilerek maksimuma çıkarılabilir.

İspat 7.1: $YH_i = c(d-c)$ ifadesi, c değişkenine göre türevi alınarak 0’a eşitlendiğinde $c=d/2$ değeri elde edilir.

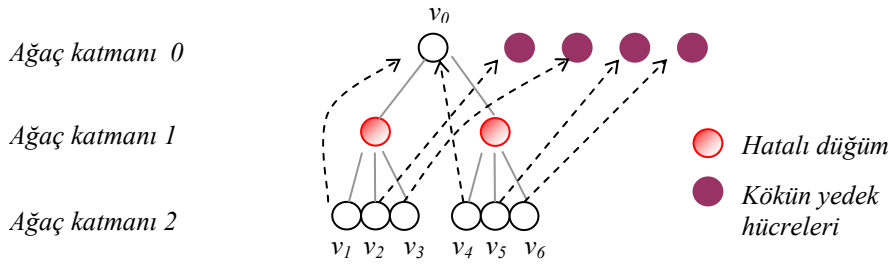
$(d_1 \geq d_2 \geq \dots \geq d_n)$ düğümlerin kapasitelerine göre sıralanmış bir dizi, d_0 ağaç kökünün kapasitesini temsil etsin ve düğüm kapasitelerinin çift sayı olduğu varsayalım. İlk olarak ağacın köküne d_1 kapasitesine sahip düğümden itibaren bağlanacak düğüm sayısı, c_0 , belirlenmektedir. Ağaca yerleşen her düğüm, önerme 7.1’de önerilen oran ile, $c_i=d_i/2$ olacak sayıda çocuk almaktadır. Verilen

parametrelere göre $f \leq c_0$ olmak üzere, en kötü durum senaryosu için ağaç kökünün f adet en yüksek kapasiteli çocuğunda hata olduğu varsayalım.

Önerme 7.2: Yedek ebeveyn kümesinde ağaç kökünün de bulunduğu varsayılırsa, eşitlik 7.1 durumu sağlandığında ağaç sorunsuz video akışlandırmaya devam edebilir.

$$\sum_{i=1}^f (c_i) \leq f + (d_0 - c_0) + \sum_{j=f+1}^{c_0} c_j \quad (\text{Eşitlik 7.1})$$

İspat 7.2: Eğer f adet kök çocuğunda hata olursa, bu çocukların çocukları (kökün torunları) yedek ebeveyn aramaya başlayacaktır; bu sayı eşitlik 7.1'in sol tarafında verilmiştir. Yedek ebeveyn arayan düğümler arasında f tanesi ebeveyninin ebeveynine bağlanabilir; bu sayı sağ taraftaki toplamın ilk terimini vermektedir. Sağdaki toplamın diğer terimleri ise kökün ve hata oluşan düğümlerin kardeşlerinin yedek hücrelerini ifade etmektedir.



Şekil 7.1. Kökün yedek hücrelerini gösteren bir hata senaryosu.

Şekil 7.1'de önerme 7.2'ye karşılık gelen bir senaryo verilmiştir. Şekle göre, sistemde $n=6$ adet düğüm vardır ve $i=0, 1, \dots, n$ olmak üzere $d_i = 6$ ve $c_0=2$ 'dir. Kökün her iki çocuğunda da hata olduğu varsayalım. Bu durumda ağacın yeniden yapılandırılmasında şu şekilde bir senaryo gerçekleşebilir: v_1 ve v_4 ağacın köküne hatalı düğümlerin yerine, v_2, v_3, v_5 ve v_6 kökün yedek hücrelerine bağlanır. Bu senaryoya göre eşitlik 7.1'in sağlandığı ve ağacın hatasız akışlandırmaya devam ettiği görülmektedir.

Şekil 7.2'de c_0 'ın bulunmasını sağlayan bir algoritma verilmiştir. Algoritmanın dıştaki döngüsü tüm çocuk sayısı olasılıklarına bakarken içte yer alan döngü değişen hatalı çocuk sayısı durumlarına göre ağacın sorunsuz video akışlandırmaya devam edip edemeyeceğini belirler. Algoritma daha sonra en fazla sayıda hatalı çocuğa rağmen akışlandırmanın devam edeceği çocuk sayısını tespit eder. Bu yaklaşım, ağacın üst bölgesinin hata dayanıklılığını arttırmaktadır.

```

1   for  $i = 1$  to  $d_0$            /*  $i$ : kökün çocuğu
2       for  $f = 1$  to  $i$            /* hatalı çocuk
3           if  $(\sum_{i=1}^f (c_i) > f + (d_0 - c_0) + \sum_{j=f+1}^{c_0} (c_j))$ 
4               then
5                    $f = f-1;$ 
6               break;
7           end if
8       end for
9        $f\_dizi(i) = f;$ 
10  end for
11   $c_0 = \text{ind}\{\text{en büyük } f\_array(i), i = 1, \dots, d_0\}$ 
    /* dizinin en büyük elemanının indeksi olan çocuk sayısı

```

Şekil 7.2. Kökün çocuk sayısını belirleyen algoritma.

Şekil 7.3'te, şekil 7.2'de verilen algoritmaya ve önerme 7.1'e dayanan çoklugönderim ağaç oluşturma algoritması verilmiştir. Şekil 7.3'te verilen çoklugönderim ağaç oluşturma algoritması ilk olarak ağaca konumlanacak düğümleri derecelerine göre sıralar, daha sonra köke bağlanacak çocuk sayısını belirlemek için şekil 7.2'deki algoritmayı çağırır. Belirlenen çocuk sayısı kadar istemci düğüm, sıralı listeden çekilerek ağacın köküne bağlanır. Önerme 7.1'de verilen şekilde, ağaca yerleşen her düğümün %50 kapasitesi yedek olarak ayrılmaktadır. Ağaca yerleşen düğümler şekil 7.4'te verilen uzlaşma algoritması ile istemci düğümler kümesinden ağaç içi gecikmeleri minimize eden çocuk seçimlerini yapar ve bu süreç istemci kümedeki tüm düğümler ağaca yerleşene kadar devam eder.

```

verilen: kök  $v_0$  ve derecesi  $d_0$ ,  $i=1, \dots, n$  ve dereceleri  $d_i$ 
            olmak üzere  $n$  adet istemci düğüm  $v_i$ 

sırala  $R = \{ v_i, i=1, \dots, n \}$  kümesini elde etmek için  $v_i$ ,
             $i=1, \dots, n$  düğümlerini  $d_i$  derecelerine göre büyükten
            küçüğe olacak şekilde

çağır  $c_0$ 'ı belirlemek için kökün çocuk sayısını belirleyen
            algoritma;

tanımla  $c_i = d_i/2$ ,  $i=1, \dots, n$  ;

bağla  $v_i \in R$ ,  $i=1, \dots, c_0$ ,  $\rightarrow$  ağaç köküne

 $k = 1$  ;

güncelle  $Q_k = \{ v_i \in R, i=1, \dots, c_0 \}$ ,  $R = R \setminus Q_k$ ,  $NL_k = c_0$ 

while (  $R \neq \emptyset$  )

    çağır  $v_i \in R$ ,  $i=1, \dots, \sum_{i=1}^{NL_k} c_i$  arasından  $v_i \in Q_k$ ,
             $i=1, \dots, NL_k$  düğümlerine çocuk bağlayan uzlaşma
            algoritması

     $k = k + 1$  ;

    güncelle  $Q_k = \{ v_i \in R, i=1, \dots, \sum_{i=1}^{NL_{k-1}} c_i \}$ ,  $R = R \setminus Q_k$ 
            ,  $NL_k = \sum_{i=1}^{NL_{k-1}} c_i$ 

end while

```

Şekil 7.3. Çokluguönderim ağaç oluşturma algoritması.

```

verilen  $C = \{ v_i, i=1, \dots, m \}$  çocuk kümesi,  $d_j$  dereceleri
            ile  $P = \{ v_j, j=1, \dots, n \}$  ebeveyn kümesi ve  $\{ gecikme_{ij},$ 
             $i=1, \dots, m, j=1, \dots, n \}$ 

sırala gecikme dizisindeki kolonları büyükten küçüğe
            olacak şekilde;

işaretle  $v_j$ 'e bağlamak üzere her  $j$ . kolonda üstten  $d_j/2$ 
            adet çocuk;

while (  $i \leq m$  )

    while (dizinin  $i$ . sırası 1'den çok işaret var ise)

        sil bu işaretler arasından üstten  $(d_j/2+1)$ .
            elemanlardan maksimum değeri içermeyen  $j$ 
            kolonundaki işaretler;

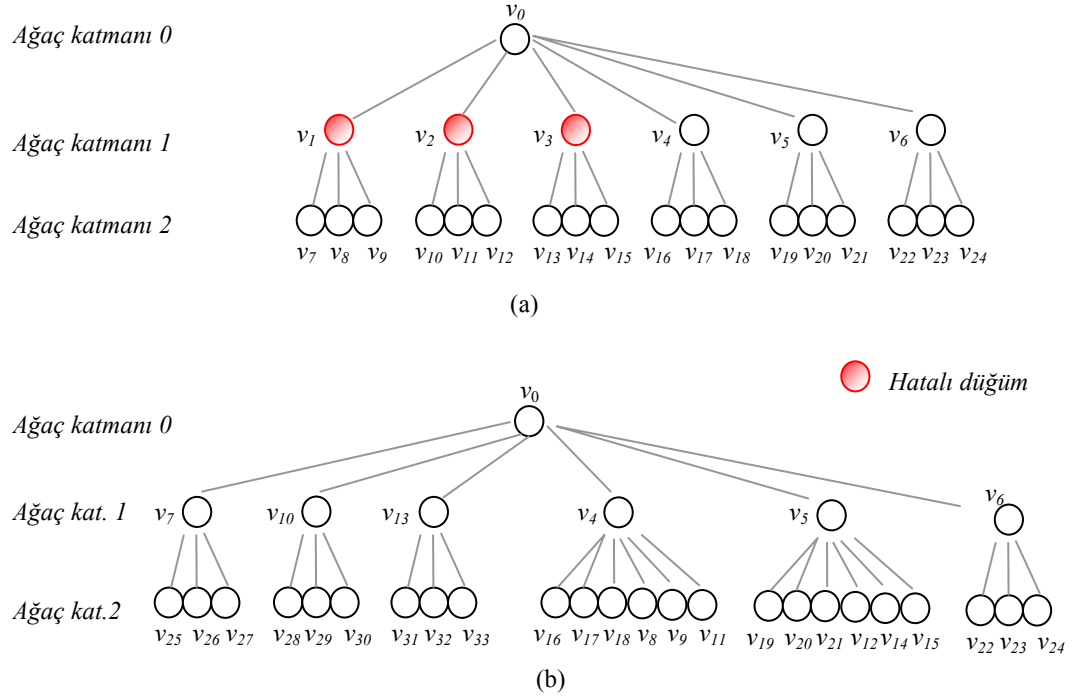
        işaretle işareti silinen  $j$ . kolonlardan,
            bir sonraki en düşük gecikmeye sahip
            çocuk;

    end while

end while

```

Şekil 7.4. Uzlaşma algoritması.



Şekil 7.5. Hatalardan sonra ağaç tamiri a) Ağacın ilk görünümü, b) Tamirden sonra ağaç görünümü.

Şekil 7.5, şekil 7.3'te verilen algoritmaya göre kurulan bir çokluguönderim ağacının hatalardan sonra tamirini göstermektedir. Şekle göre ağaçta yer alan her düğümün kapasitesi 6 olarak alınmıştır. Şekil 7.2'deki algoritma çalıştırıldığında, köke bağlanacak çocuk sayısı $f=3$ adet hatalı düğüm için 6'dır. Verilen koşullara göre şekil 7.5.a'da kurulan ağaçta, verilen şekle göre kökün 3 çocuğu v_1 , v_2 ve v_3 'te hata oluşmuştur. Daha sonra şekil 7.5.b'de görüldüğü üzere, kökün 3 çocuğu v_7 , v_{10} ve v_{13} ebeveynlerinin yerine köke bağlanmıştır. Hatalı düğümlerin diğer çocukları şekilde görüldüğü üzere hatalı düğümlerin kardeşlerine bağlanmış ve ağaç ayakta kalmıştır. Eğer 4 adet kök çocuğunda hata olursa ağaçta bozulma meydana geleceği açık olarak görülmektedir. Bu durumda yedek ebeveyn arayan ve açıkta kalan çocuklar, bir süre sonra hatalı düğümlerin yerine geçen düğümlerin yedek hücrelerine bağlanabilir ve ağaç yeniden yapılır. Şekil 7.1 ve 7.5'te verilen senaryolar eşit kapasiteli düğümlerden oluşan ağaçları temsil etmektedir ancak şekil 7.2, 7.3 ve 7.4'te verilen algoritmalar genel amaçlı olarak kapasiteleri eşit olmayan düğümlerden oluşan ağaçları oluşturabilmektedir.

Önerme 7.3: Tüm düğümlerin derecelerinin $d \geq 4$ ve sistemdeki düğüm sayısının N olduğu varsayılın. Bu durumda çoklugoönderim oluşturma algoritmasının karmaşıklığı $O(N^2)$ 'dir.

İspat 7.3: Kökün çocuk sayısını belirleyen algoritmanın karmaşıklığı $O(d^2)$ 'dir. *While* döngüsü içerisindeki uzlaşma algoritması çoklugoönderim ağaç oluşturma algoritmasının temel kısmını oluşturmaktadır; n_p ebeveyn ve n_c çocuk için uzlaşma algoritması karmaşıklığı $O(n_p n_c)$ 'dir. En kötü durum senaryosunda, $i = 1, \dots, h$ ve h ağacın uzunluğu olmak üzere ağacın i . katmanında $d(d/2)^{i-1}$ adet düğüm bulunmaktadır. Bu durumda, uzlaşma algoritması $n_p = d(d/2)^{i-1}$, $i = 1, \dots, h-1$ ve $n_c = n_p(d/2)$ değerleri için çalıştırılır.

$$N = d \sum_{i=1}^h \left(\frac{d}{2} \right)^{i-1} \quad (\text{Eşitlik 7.2})$$

ve $m = (h-1) + (h-2) - 1 = 2h-4$ olmak üzere uzlaşma algoritması *while* döngüsü içinde eşitlik 7.3'te verilen değer kadar çalışır.

$$M = \frac{d^3}{2} \sum_{\substack{i=0 \\ i \text{ çift}}}^m \left(\frac{d}{2} \right)^i \quad (\text{Eşitlik 7.3})$$

$$\sum_{i=1}^n q^{i-1} = \frac{q^n - 1}{q - 1}, q \neq 1 \quad (\text{Eşitlik 7.4})$$

$q = d/2$ için eşitlik 7.4, eşitlik 7.2'ye uygulandığında, eşitlik 7.5 elde edilir.

$$N = d \frac{\left(\frac{d}{2} \right)^h - 1}{\left(\frac{d}{2} \right) - 1} \quad (\text{Eşitlik 7.5})$$

$q = (d/2)^2$ için eşitlik 7.4, eşitlik 7.3'e uygulandığında, eşitlik 7.6 elde edilir.

$$M = \frac{d^3}{2} \frac{\left(\frac{d}{2} \right)^{2h-2} - 1}{\left(\frac{d}{2} - 1 \right) \left(\frac{d}{2} + 1 \right)} \leq \frac{d^3}{2} \frac{\left(\frac{d}{2} \right)^{2h-2}}{\left(\frac{d}{2} - 1 \right) \left(\frac{d}{2} + 1 \right)} \quad (\text{Eşitlik 7.6})$$

Eşitlik 7.6'dan eşitlik 7.7 yazılabilir.

$$M \leq \frac{d^3}{2} \cdot \left(\frac{d}{2}\right)^{-2} \frac{\left(\frac{d}{2}\right)^{2h}}{\left(\frac{d}{2}-1\right)\left(\frac{d}{2}+1\right)} = 2d \frac{\left(\frac{d}{2}\right)^{2h}}{\left(\frac{d}{2}-1\right)\left(\frac{d}{2}+1\right)} \quad (\text{Eşitlik 7.7})$$

Eşitlik 7.5 yardımı ile eşitlik 7.8 yazılabilir.

$$M \leq 2d \frac{\left(\frac{d}{2}\right)^{2h}}{\left(\frac{d}{2}-1\right)\left(\frac{d}{2}\right)} = 2d \left(\frac{2}{d}\right) \frac{\left(\frac{d}{2}\right)^{2h}}{\left(\frac{d}{2}-1\right)} = 4 \frac{\left(\frac{d}{2}\right)^{2h}}{\left(\frac{d}{2}-1\right)} \quad (\text{Eşitlik 7.8})$$

$d \geq 4$ olduğu için $\left(\frac{d}{2}-1\right) \geq 1$ 'tir ve eşitlik 7.9 elde edilir.

$$M \leq 4 \left(\frac{d}{2}\right)^{2h} \leq 4d^{2h} \quad (\text{Eşitlik 7.9})$$

$h = \log_d N$ olduğu için, eşitlik 7.10 elde edilir.

$$, \quad M \leq 4d^{2\log_d N} = 4d^{\log_d N^2} = 4N^2 \quad (\text{Eşitlik 7.10})$$

Sonuç olarak, eşitlik 7.11 geçerlidir ve ispat tamamlanır.

$$O(M) = O(N^2) \quad (\text{Eşitlik 7.11})$$

Önerme 7.4, sistemdeki düğümler eşit olduğunda ağacın tamiri için gerekli koşulu vermektedir.

Önerme 7.4: Sistemdeki tüm düğümlerinin kapasitesinin d 'ye eşit olduğu ve 0'dan daha altta yer alan bir l ağaç katmanında f adet hatalı düğüm olduğu varsayalım. Eğer yedek ebeveyn kümesi l 'ye eşit ve daha üst katmandaki tüm düğümleri içeriyor ise eşitlik 7.12 sağlandığı durumda ağaçta bozulma olmaz.

$$f \leq \frac{d}{d-1} \sum_{i=1}^l \left[\frac{d}{2} \right]^i \quad (\text{Eşitlik 7.12})$$

İspat 7.4: Ağacın bozulma olmadan devam edebilmesi için 0, 1, ..., l 'ye kadar olan tüm katmanlarda hatalı düğümlerin çocuklarının bağlanabileceği kadar sayıda yedek hücre olması gerekir. Ağaç üzerindeki düğümlerin kapasitesi d

olduğu için ağacın köküne d adet ve ağaç üzerindeki tüm çocuklara $d/2$ adet çocuk bağlanır. Bu durumda eşitlik 7.13 yazılabilir.

$$-f \frac{d}{2} + d \frac{d}{2} + d \left(\frac{d}{2} \right)^2 + \dots + d \left(\frac{d}{2} \right)^l + f \geq f \frac{d}{2} \quad (\text{Eşitlik 7.13})$$

Eşitliğin sol tarafındaki ilk terim hatalı düğümlerden dolayı kaybedilen yedek hücre sayısını, diğer terimler ilgili katmandaki yedek hücre sayısını ve son terim hatalı düğümlerden dolayı hatalı düğümlerin ebeveynlerinde oluşan boş hücreleri temsil etmektedir. Eşitliğin sağ tarafında ise yedek ebeveyn arayan çocuk sayısı hesaplanmaktadır. Eşitlik 7.13'ün düzenlenmesi ile eşitlik 7.12 elde edilir.

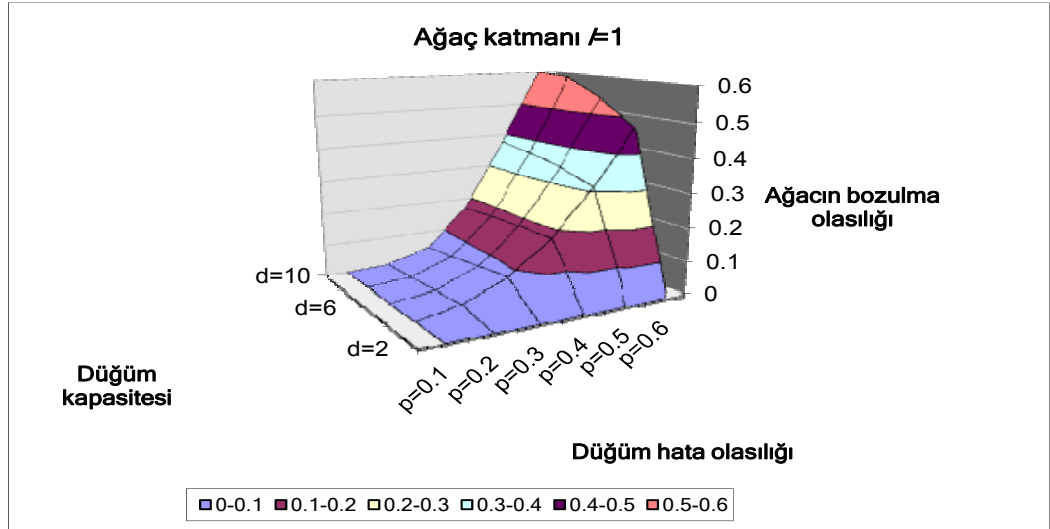
Yukarıda yer alan analizler düğüm hatalarından sonra yedek ebeveynlerin henüz güncellenmediği zaman içerisinde ağaçta oluşan yapılanmayı gözönünde bulundurmaktadır. Başka bir deyişle, önerme 7.4 ağacın bir katmanında yer alan düğümlerde ardarda hata oluştuğunu varsaymaktadır. Ardarda hataların yaşanmadığı durumlarda ise ağaç bozulmadan video akışlandırmaya devam edebilmektedir.

Belli bir zaman aralığındaki hata oranı p , ağaç uzunluğu h , düğümlerin kapasitesi d , ağacın l . katmanındaki düğüm sayısı a_l ve aynı katmanda ağacın bozulmasına sebep olan en az hatalı düğüm sayısı b_l ile ifade edilsin. Eşitlik 7.16, eşitlik 7.14 ve 7.15'te verilen değerleri kullanarak düğümlerin eşit kapasitede olduğu durumlar için ağacın l . katmanında hatalı düğümler bulunduğunda ağacın bozulma olasılığını vermektedir. Eşitlik 7.16'da verilen değer, a_l adet düğüm arasından b_l adet düğümde hata olma olasılıklarını toplayarak l . katman için ağacın bozulma olasılığını vermektedir.

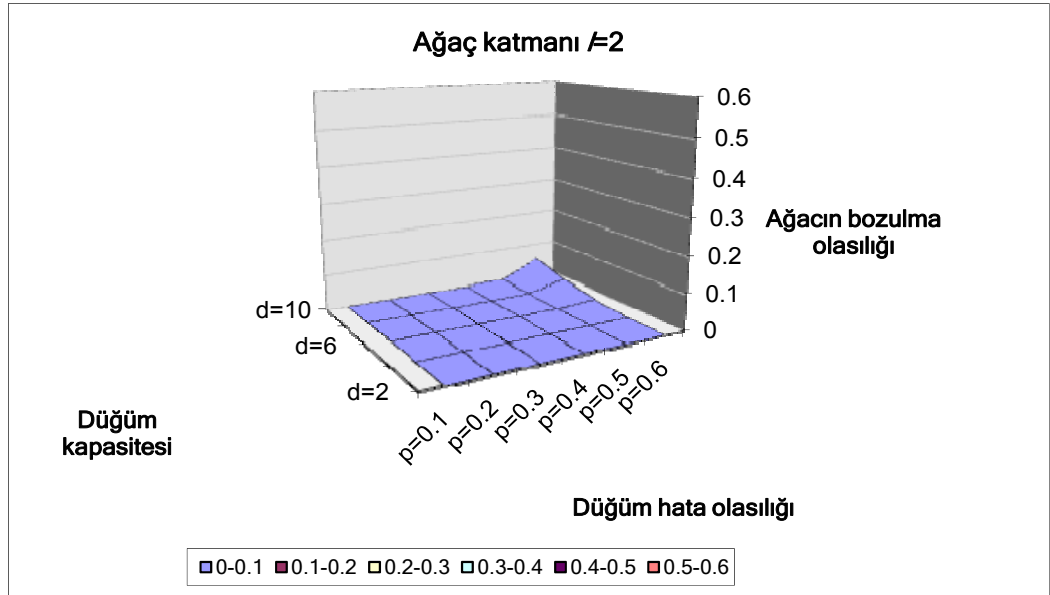
$$a_l = d \left(\frac{d}{2} \right)^{l-1} \quad (\text{Eşitlik 7.14})$$

$$b_l = \frac{d}{d-1} \sum_{j=1}^l \left(\frac{d}{2} \right)^j \quad (\text{Eşitlik 7.15})$$

$$\sum_{j=0}^{a_l-b_l} \binom{a_l}{b_l+j} p^{b_l+j} (1-p)^{a_l-b_l-j} \quad (\text{Eşitlik 7.16})$$



(a)



(b)

Şekil 7.6. Farklı düğüm kapasiteleri ve hata oranları için karşılaştırmalı ağaç hata olasılıkları.

Çizelge 7.1. Ağaçtan istemsiz kopma analizi notasyonu.

d	Düğümün kapasitesi
c	Bir düğümün kapasitesinden çocuklarına ayırdığı oran
N	Kök hariç ağaçtaki düğüm sayısı
nn_j	Ağacın j . katmanındaki düğüm sayısı (ağacın kökü 0. katmandır)
h	Ağacın yüksekliği
nns_j^k	$k=1, \dots, h-1$ olmak üzere kökü ağacın k . katmanında olan bir alt ağacın j . katmanındaki düğüm sayısı

nb_j	Ağacın j. katmanındaki yedek hücre sayısı
p	Belirli bir zaman aralığında düğüm hata olasılığı
nf_j	Ağacın j. katmanındaki hatalı düğüm sayısı
c_h	Ağacın son katmanındaki düğüm başına çocuk oranı, nn_h/nn_{h-1}
nnb_j	Ağacın j. katmanında yedek ebeveyn bulamayan düğüm sayısı
nnb_j^k	Ağacın k. katmanında oluşan hatalar sebebiyle j. katmanda yedek ebeveyn bulamayan düğüm sayısı
$nf_s_j^k$	$k=1,...,h-1$ olmak üzere kökü ağacın k. katmanında olan alt ağacın j. katmanında bulunan hatalı düğüm sayısı
nlb_j^k	Ağacın k. katmanındaki hatalardan dolayı yedek ebeveyn arayan ve j. katmanda bulunan düğüm sayısı
$ntlb_j$	Ağacın j. katmanında yedek ebeveyn arayan toplam düğüm sayısı

Eşitlik 7.16'da verilen formüle dayanarak ağaç katmanları $l=1$ ve $l=2$ için sırasıyla şekil 7.16.a ve 7.16.b'de farklı düğüm kapasitesi ve hata olasılığı değerleri için oluşan grafikler verilmiştir. Şekilde görüldüğü üzere, ağacın ilk katmanı hatalara karşı 2. katmanına göre daha dayanıksızdır. Düğümlerinin kapasitelerinin eşit olmadığı durumlarda ise ağacın birinci katmanının hataya karşı dayanıksızlığının daha da artacağı öngörülmektedir. Örneğin ağaç üzerinde sadece 1 tane yüksek kapasiteli düğüm bulunması durumunda, bu düğümün ağaçtan ayrılması ağacın tamamının bozulmasına neden olacaktır.

Eşit düğüm kapasitesine sahip durum için yapılan son analiz belirli bir zaman aralığında ağaç üzerinde oluşacak hatalarda belirli bir hata oranı için ağaçtan istemsiz kopacak ortalama düğüm sayısını vermektedir. Kullanılan notasyon çizelge 7.1'de verilmiştir. Yapılan analizde, ağacın yapraklarında düğümlerin bir üst katmandaki düğümlere eşit sayıda çocuk olark dağıtıldığı ve düğümler ebeveyn hatasıyla karşılaştığında sadece bulunduğu katmanın bir üst katmanındaki yedek ebevenlere bağlandığı varsayılmıştır.

Çizelge 7.1'de verilen notasyonların hesaplamaları eşitlik 7.17 – 7.20 arasında verilmiştir.

$$nn_j = \begin{cases} d^j c^{j-1} & j = 1, \dots, h-1 \\ N - \sum_{k=1}^{h-1} d^k c^{k-1} & j = h \end{cases} \quad (\text{Eşitlik 7.17})$$

$$nb_j = d^{j+1} c^{j-1} (1-c) - d(1-c)(E(nf_j) + E(nnb_j)) \quad j = 1, \dots, h-1.$$

(Eşitlik 7.18)

$$nns_j^k = \begin{cases} d^j c^j & j = 0, 1, \dots, h-k-1 \\ d^j c^j \frac{nn_h}{nn_{h-1}} & j = h-k \end{cases} \quad (\text{Eşitlik 7.19})$$

$$E(nfs_j^k) = \sum_{m=1}^{nns_j^k} m \binom{nns_j^k}{m} p^m (1-p)^{nns_j^k-m}, \quad j = 0, \dots, h-k, \quad k=1, \dots, h-1.$$

(Eşitlik 7.20)

Ağacın birinci katmanında yer alan bir düğüm için, ikinci katmanda cd adet çocuk bulunur. Bu düğümler arasında bir tanesi ebeveynde hata olması durumunda ebeveyninin ebeveynine bağlanır. Verilen hata oranına göre ağacın birinci katmanında eşitlik 7.21'den elde edilen değer kadar düğümde hata olacaktır. Ağacın birinci katmanında oluşan hatalardan dolayı ikinci katmanda yedek ebeveyn arayan düğüm sayısı ile ilgili denklemler eşitlik 7.22 ve 7.23'te verilmiştir.

$$E(nfs_1^1) = \sum_{k=1}^{cd} k \binom{cd}{k} p^k (1-p)^{cd-k} \quad (\text{Eşitlik 7.21})$$

$$E(nf_1) = \sum_{m=1}^{nn_1} m \binom{nn_1}{m} p^m (1-p)^{nn_1-m} \quad (\text{Eşitlik 7.22})$$

$$E(nlb_2^1) = (cd - 1 - E(nfs_1^1)) E(nf_1) = E(ntlb_2) \quad (\text{Eşitlik 7.23})$$

Eşitlik 7.24'te ağacın ikinci katmanında yedek ebeveyn bulamayan düğümlerin beklenen değeri verilmiştir.

$$E(nnb_2) = E(nnb_2^1) = E(ntlb_2) - nb_1 \quad (\text{Eşitlik 7.24})$$

Ağacın birinci katmanında oluşan hatalardan dolayı 3. katmanda eşitlik 7.25'ten elde edilen değer kadar düğüm yedek ebeveyn arayacaktır.

$$E(nlb_3^1) = cd E(nnb_2) \quad (\text{Eşitlik 7.25})$$

Ağacın ikinci katmanında yer alan bir düğüm için, üçüncü katmanda cd adet çocuk bulunur. Bu düğümler arasında bir tanesi ebeveynde hata olması durumunda ebeveyninin ebeveynine bağlanır. Verilen hata oranına göre ağacın ikinci katmanında eşitlik 7.26'dan elde edilen değer kadar düğümde hata

olacaktır. Ağacın ikinci katmanında oluşan hatalardan dolayı üçüncü katmanda yedek ebeveyn arayan düğüm sayısı ile ilgili denklemler eşitlik 7.27 ve 7.28’de verilmiştir.

$$E(nfs_1^2) = \sum_{k=1}^{cd} k \binom{cd}{k} p^k (1-p)^{cd-k} \quad (\text{Eşitlik 7.26})$$

$$E(nf_2) = \sum_{m=1}^{nn_2} m \binom{nn_2}{m} p^m (1-p)^{nn_2-m} \quad (\text{Eşitlik 7.27})$$

$$E(nlb_3^2) = (cd - 1 - E(nfs_1^2)) E(nf_2) \quad (\text{Eşitlik 7.28})$$

Böylece, eşitlik 7.29, 7.30, 7.31 ve 7.32 yazılabilir.

$$E(ntlb_3) = E(nlb_3^1) + E(nlb_3^2) \quad (\text{Eşitlik 7.29})$$

$$E(nnb_3) = E(ntlb_3) - nb_2 \quad (\text{Eşitlik 7.30})$$

$$E(nnb_3^1) = E(nlb_3^1) E(nnb_3) / E(ntlb_3) \quad (\text{Eşitlik 7.31})$$

$$E(nnb_3^2) = E(nlb_3^2) E(nnb_3) / E(ntlb_3) \quad (\text{Eşitlik 7.32})$$

Ağacın 3. katmanında $E(nnb_3^1)$ adet düğüm birinci katmanda oluşan hatalardan dolayı yedek ebeveyn bulamayacaktır. Bu düğümlere ait $E(nlb_4^1) = cd E(nnb_3^1)$ adet çocuk yedek ebeveyn aramaya başlar. Benzer şekilde, ağacın üçüncü katmanında $E(nnb_3^2)$ adet düğüm ikinci katmanda oluşan hatalardan dolayı yedek ebeveyn bulamayacak ve bu düğümlere ait $E(nlb_4^2) = cd E(nnb_3^2)$ adet çocuk yedek ebeveyn aramaya başlayacaktır.

Yapılan analiz geliştirilecek olursa, ağacın j . katmanında yer alan bir düğüm için, $j+1$. katmanda cd adet çocuk bulunur. Bu düğümler arasında bir tanesi ebeveynde hata olması durumunda ebeveyninin ebeveynine bağlanır. Verilen hata oranına göre ağacın j . katmanında eşitlik 7.33’ten elde edilen değer kadar düğümde hata olacaktır. Ağacın j . katmanında oluşan hatalardan dolayı $j+1$. katmanda yedek ebeveyn arayan düğüm sayısı ile ilgili denklemler eşitlik 7.34 ve 7.35’te verilmiştir.

$$E(nfs_1^j) = \sum_{k=1}^{cd} k \binom{cd}{k} p^k (1-p)^{cd-k} \quad (\text{Eşitlik 7.33})$$

$$E(nf_j) = \sum_{m=1}^{nn_j} m \binom{nn_j}{m} p^m (1-p)^{nn_j-m} \quad (\text{Eşitlik 7.34})$$

$$E(nlb_{j+1}) = (cd - 1 - E(nfs_1)) E(nf_j) \quad (\text{Eşitlik 7.35})$$

Böylece, eşitlik 7.36, 7.37 ve 7.38 yazılabilir.

$$E(ntlb_{j+1}) = \sum_{k=1}^j E(nlb_{j+1}^k) \quad (\text{Eşitlik 7.36})$$

$$E(nnb_{j+1}) = E(ntlb_{j+1}) - nb_j \quad (\text{Eşitlik 7.37})$$

$$E(nnb_{j+1}^k) = E(nlb_{j+1}^k) E(nnb_{j+1}) / E(ntlb_{j+1}) \quad k=1, \dots, j \quad (\text{Eşitlik 7.38})$$

Ağacın alttan bir üst katmanı $h-1$. katman için, toplam c_h adet çocuk bulunur ve bu çocuklardan biri ebeveyn hatası durumunda ebeveyninin ebeveynine bağlanır. Verilen hata oranına göre ağacın $h-1$. katmanında eşitlik 7.39'dan elde edilen değer kadar düğümde hata olacaktır. Böylece, $h-1$. katmandaki hatalardan dolayı h . katmanda yedek ebeveyn arayan düğüm sayısı ile ilgili denklemler eşitlik 7.40 ve 7.41'de verilmiştir.

$$E(nfs_1^{h-1}) = \sum_{k=1}^{c_h} k \binom{c_h}{k} p^k (1-p)^{c_h-k} \quad (\text{Eşitlik 7.39})$$

$$E(nf_{h-1}) = \sum_{m=1}^{nn_{h-1}} m \binom{nn_{h-1}}{m} p^m (1-p)^{nn_{h-1}-m} \quad (\text{Eşitlik 7.40})$$

$$E(nlb_h^{h-1}) = (c_h - 1 - E(nfs_1^{h-1})) E(nf_{h-1}) \quad (\text{Eşitlik 7.41})$$

Son olarak, eşitlik 7.42, 7.43 ve 7.44 yazılabilir.

$$E(ntlb_h) = \sum_{k=1}^{h-1} E(nlb_h^k) \quad (\text{Eşitlik 7.42})$$

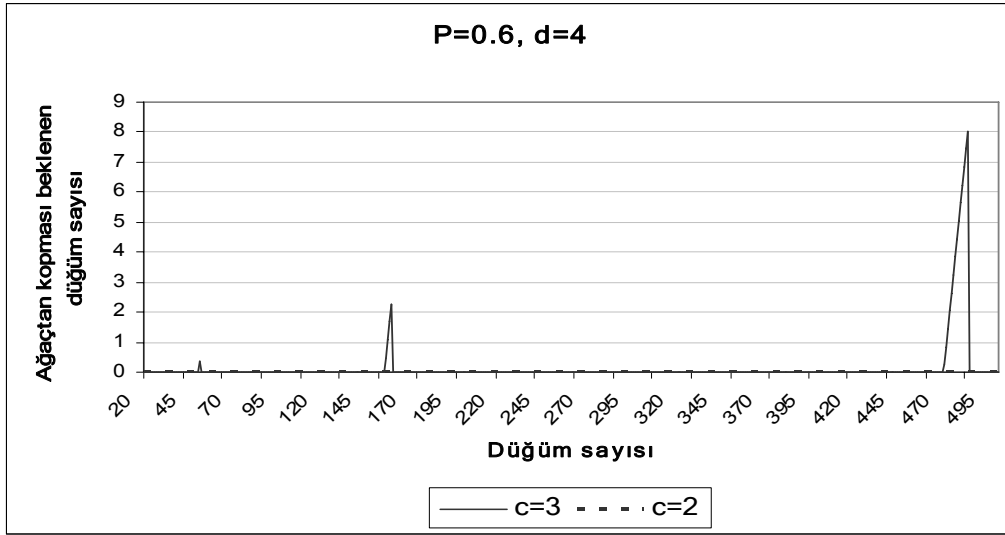
$$E(nnb_h) = E(ntlb_h) - nb_{h-1} \quad (\text{Eşitlik 7.43})$$

$$E(nnb_h^k) = E(nlb_h^k) E(nnb_h) / E(ntlb_h) \quad k=1, \dots, h-1 \quad (\text{Eşitlik 7.44})$$

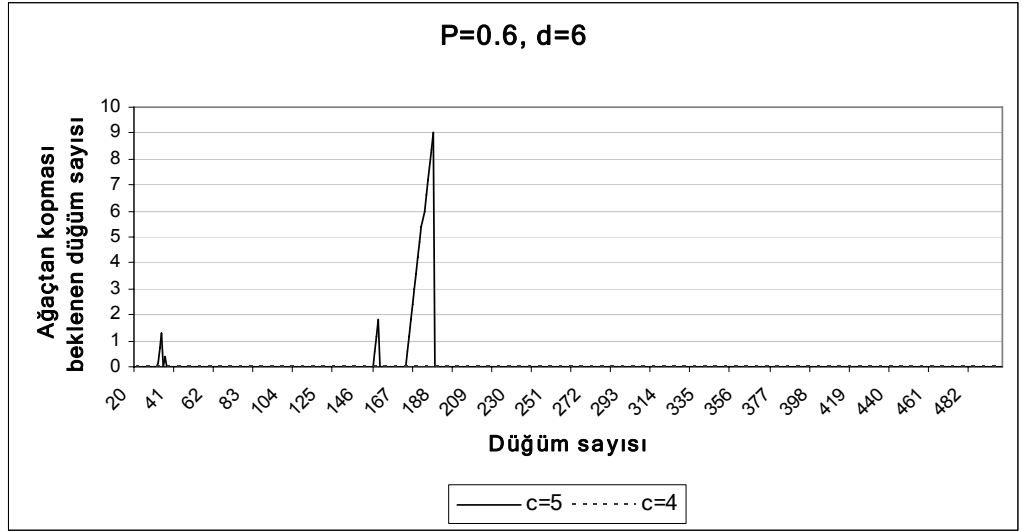
Verilen hata oranına göre ağaçtan istemsiz olarak kopması beklenen düğüm sayısı, N_k , eşitlik 7.45'te verilmiştir.

$$N_k = \sum_{j=2}^h E(nnb_j) \quad (\text{Eşitlik 7.45})$$

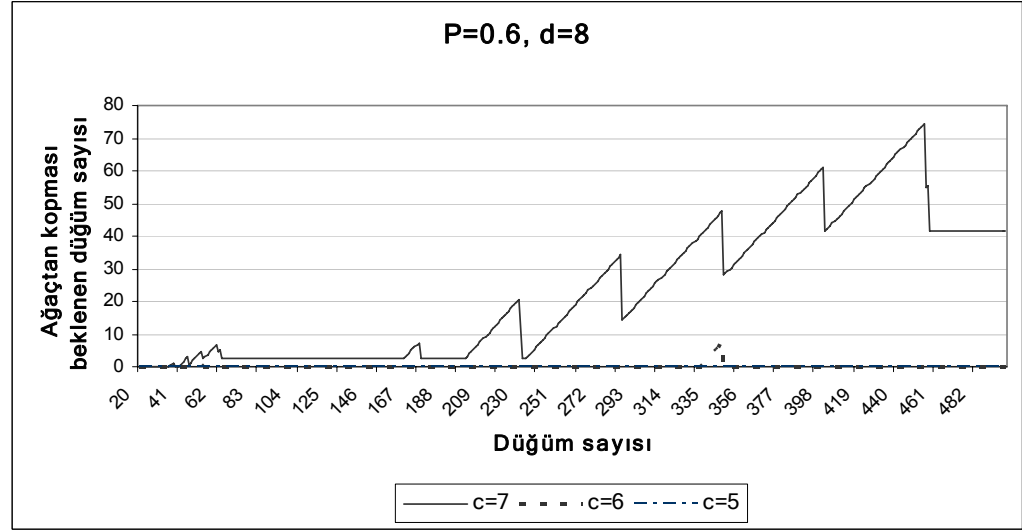
Eşitlik 7.45'te verilen formül farklı p , c , d ve N değerleri için denenerek ağaçtan istemsiz olarak kopması beklenen düğüm sayısı sonuçları incelenmiştir. Şekil 7.7.a, 7.7.b, 7.7.c ve 7.7.d'de $p=0,6$, $d=4$, 6, 8, 10 değerleri için elde edilen grafikler verilmiştir. Grafiklerde verilen c bir düğüme bağlanan çocuk sayısını temsil etmektedir. Grafiklerde verilmeyen daha düşük tüm c değerleri için çıkan N_k değerleri 0'dır. Buna benzer şekilde, denenen farklı parametrelerden de sonuç olarak 0 olarak alınmıştır. c değerinin $0,5d$ 'den daha büyük değerleri için ağaçtan istemsiz olarak kopması beklenen düğüm sayısının artmaya başladığı gözlemlenmiştir. Ancak, p değerleri arttıkça N_k değerleri beklendiği gibi artma göstermemiştir. Bunun sebebi fazla sayıda düğüm sistemden çıktığı için yedek ebeveyn arayan düğüm sayısında da bir düşüş yaşanmasıdır.



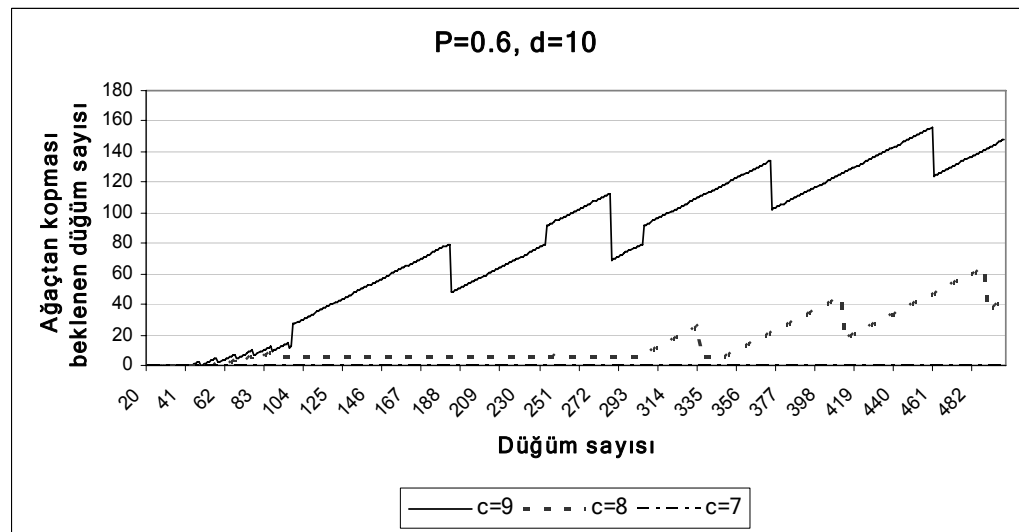
(a)



(b)



(c)



(d)

Şekil 7.7. Farklı düğüm ve çocuk sayılarına göre ağaçtan kopması beklenen düğüm sayısı.

8. BENZETİM SONUÇLARI

Tezin bu bölümünde, geliştirilen algoritma ve yöntemlerin test edilmesi ve benzetim ortamındaki kazanım sonuçları ilgili grafikler ve yorumlar verilecektir. Görevdeş ağlarda video akışlandırma çalışmasına yönelik geliştirilen algoritmalar için gerekli ağ yapısı topolojisi GT-ITM (GT-ITM, 2006) tarafından modellenmiştir. Oluşturulan ağ yapısı farklı deneyler için lokal ve NS2 (NS2, 2006) benzetim ortamları üzerine taşınarak algoritmalar test edilmiş ve ilgili kazanım sonuçlarına ulaşılmıştır.

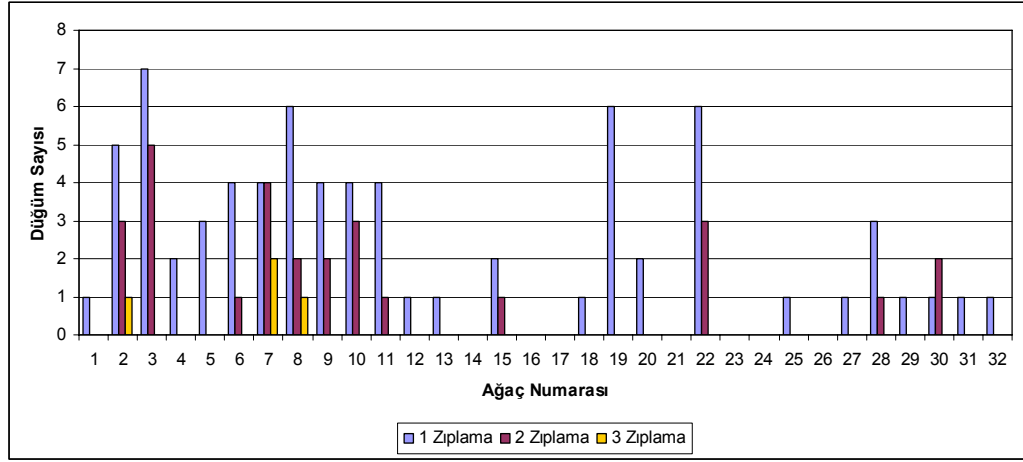
NS2 benzetim ortamı üzerinde yapılan gruplandırma ve hiyerarşik katmanların oluşturulması 500 düğüm üzerinde test edilmiştir. Çokluguönderim ağaçları ve video akışlandırma uygulamaları ile ilgili testler ise sayısı 250 – 500 arasında değişen düğüm üzerinde yapılmıştır. NS2 yığıt boyutunun sınırlı olması nedeniyle uygulanan protokol sayısı ve büyüklüğü arttıkça üzerinde çalışılan düğüm sayısı azalmaktadır. Lokal benzetim ortamı üzerinde yapılan testlerde, 500 düğüm ile çalışılmıştır. Lokal benzetim ortamı için gereken ağ altyapısı, NS2 deneylerine benzer şekilde GT-ITM tarafından modellenmiştir.

8.1 Chang Roberts Algoritması ile Oluşturulan Çokluguönderim Ağaçları Test Sonuçları

Video akışlandırma oturumu için altyapı oluşturacak çokluguönderim ağaçlarının oluşturulmasında ilk olarak Chang Roberts algoritmasından yararlanılmıştır. Bu algoritma ile hiyerarşik katmanlar üzerinde tüm gruplar içinde oluşturulan çokluguönderim ağaçları ile video izleme talebinde bulunan ve izlenmek istenen videoyu elinde bulunduran düğümlerin tümünü içeren çokluguönderim ağaçları oluşturulmuştur.

Şekil 8.1 ve 8.2’de Chang Roberts algoritmasının başarımı ile ilgili grafikler verilmiştir. NS2 benzetim ortamı deneylerinde, 500 düğüm içeren bir topolojide ilk olarak hiyerarşik katmanlar oluşturulmuştur. Grupların oluşumu tamamlandıktan sonra düğümlerin video talepleri liderler tarafından toplanmış ve sistemde arama işlemi yapılmıştır. Videoyu bulunduran kaynak düğümler ve isteyen düğüm listeleri tam olarak her hiyerarşik katman için oluşturulduktan sonra kaynak düğümler algoritmayı çalıştırmaya başlamıştır. Algoritmada ağ yapıslandırma, alınan ilk mesaja göre biçimlendirildiği için her kaynak aynı anda mesaj göndermeye başlamalıdır. Bu yüzden sistemde aynı düğüm grubuna mesaj

gönderecek olan kaynak düğümlerin aralarında mesajlaşarak zamanlayıcılarını ayarlaması ve algoritmayı aynı anda çalıştırmaya başlamaları sağlanmıştır.



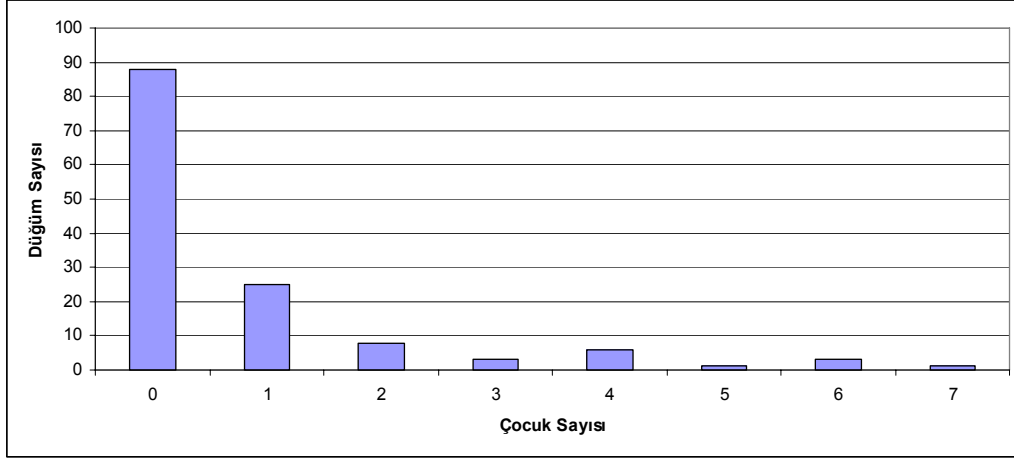
Şekil 8.1. Düğüm sayısı – zıplama adedi histogramı.

Şekil 8.1’de yer alan grafikte farklı hiyerarşik katmanlarda oluşan ağaçların üzerinde bulunan düğümlerin herbirine ulaşmak için gereken zıplama adedine bağlı düğüm sayısı frekansları verilmiştir. Örneğin 2 numaralı ağaçta 1 zıplamayla ulaşılan düğüm sayısı 5, 2 zıplamayla ulaşılan düğüm sayısı 3 ve 3 zıplamayla ulaşılan düğüm sayısı 1’dir. Hiçbir düğümün zıplama adedi bilgisinin görülmediği ağaçlar, o kaynağa hiçbir düğümün bağlanmadığını göstermektedir. Bu durumda bu kaynakların talepte bulunan düğüm listesinden hiçbir düğüme en yakın kaynak olmadığı sonucu çıkarılabilir.

Şekil 8.2’de verilen grafikte bir düğüme bağlı çocuk sayıları verilmiştir. Grafikte görüldüğü üzere, sistemde oluşturulan ağaçlar üzerindeki düğümlerin çok büyük bir kısmı yaprak olarak konumlandırılmış ve bu düğümlere hiçbir düğüm bağlanmamıştır. Bunun dışında kaynak düğümlere bağlı çocuk sayısının 7’ye kadar çıkabildiği gözlemlenmektedir.

Benzetim ortamında yapılan deneyler sonucu, Chang Roberts algoritmasının görevdeş ağlarda video akışlandırma sistem gereksinimlerinden çocuk sayısı kontrolü konusunda başarılı sonuçlar vermediği gözlemlenmiştir. Yapılan deneylerde, özellikle kaynak düğüm o videoyu talep eden düğümler arasında merkezi bir konumdaysa, her düğüm ilk olarak kaynaktan mesaj aldığı için tümü kaynağa bağlanmaktadır. Bir düğüm kendisine bağlı çocuk sayısını ancak mesaj gönderdikten ve gönderdiği mesajlara yanıt aldıktan sonra tespit edebildiği için gönderilen mesaj sayısı konusunda bir sınırlama

getirilememektedir. Video akışlandırma sistemlerinde gecikmeden daha önemli bir parametre olan bantgenişliği değerlerinin algoritmada ağaç yapılanma sırasında kullanılamaması Chang Roberts algoritmasının bir diğer dezavantajı olmuştur.



Şekil 8.2. Düğüm sayısı – çocuk sayısı histogramı.

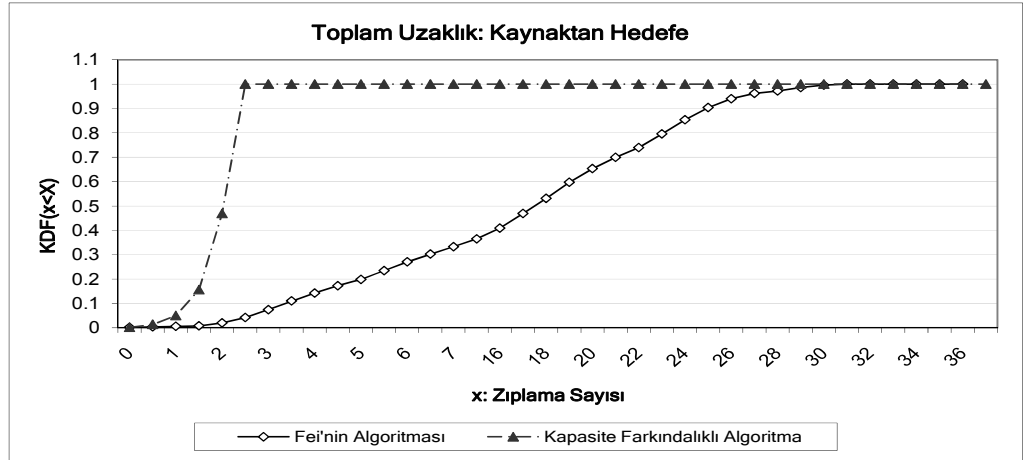
8.2 Kapasite Farkındalıklı Algoritma ile Oluşturulan Çoklugoönderim Ağaçları Test Sonuçları

Tezin dördüncü bölümünde ayrıntıları açıklanan ve video akışlandırma uygulamalarına özel olarak geliştirilen kapasite farkındalıklı çoklugoönderim ağaç oluşturma algoritmasının kazanımını ölçmek amacıyla farklı platformlarda testler yapılmış ve literatürde bulunan benzer çalışmalar ile karşılaştırma deneyleri raporlanmıştır.

8.2.1 Kapasite farkındalıklı ağaç oluşturma algoritması karşılaştırmalı test sonuçları

Kapasite farkındalıklı çoklugoönderim ağaç oluşturma algoritmasını fazla sayıda düğüm ile test edebilmek amacıyla GT-ITM topolojisini kullanmayı sağlayan bir simülatör yazılmış; GT-ITM ile oluşturulan ağ üzerinde 500 adet düğüm oluşturularak, yazılan simülatör ile bu düğümlerin tamamı çoklugoönderim ağacına dahil edilmiştir. Önerilen algoritma yoluyla oluşturulan ağaçlarda zıplama sayısı ve bantgenişliği etkin kullanımı ile ilgili sonuçlar, literatürde yer alan benzer bir çalışma ile karşılaştırılmıştır. Bu karşılaştırmada eşit koşullar yaratmak amacıyla kapasite farkındalıklı ağaç üzerinde yer alacak tüm düğümler bir grup içerisine yerleştirilmiş ve bu yaklaşım ile sistemdeki tüm düğümler düz (flat) bir yapıda değerlendirilmiştir. Bu bölümde verilen materyal (Sayıt ve Tunalı,

2010a)'da da verilmiştir. (Fei and Yang, 2007) tarafından video akışlandırma için kullanılması önerilen ağaç yapısında, ağaçlar temel olarak kapasite sınırlı çokluguönderim ağaç oluşturma algoritmasına (Fei and Yang, 2007) göre oluşturulmaktadır. İlk olarak yapılan deney setinde, her düğümün kapasitesi 3 olarak alınmıştır. Buna göre, her düğüme bir adet ebeveyn ve 3 adet çocuk bağlanabilmektedir. Şekil 8.3'te verilen grafiğe göre, yapılan istatistiki sonuçlar sonucu önerilen algoritma ile kaynağa en uzak düğümler en fazla 6 zıplama uzaklıkta yer almıştır; yapılan deneylerde düğümlerin %70'i ise 5 kaynağa 5 adet zıplama uzaklıktadır. Ancak (Fei and Yang, 2007) tarafından kullanılması önerilen algoritmada ağaç uzunluklarının 36 zıplamaya kadar çıkabildiği gözlemlenmiştir. Yapılan deneylerde, düğümlerin %4'ü kaynaktan 6 zıplama uzaklıkta yer almıştır. Daha önceki bölümlerde belirtildiği gibi, ağaç uzunluğu video akışlandırma uygulamalarında önemli bir parametredir, ağaç uzunluğu arttıkça ağacın üst katmanlarında bulunan düğümlerde oluşabilecek hatalardan ağacın altta kalan tüm kısmı etkilenmektedir. Dolayısıyla kısa ağaçlarda bulunan düğümler, video akışlandırma oturumu sırasında oluşan hatalardan en az seviyede etkilenmektedir.



Şekil 8.3. Kaynaktan hedefe zıplama sayısı cinsinden toplam uzaklık.

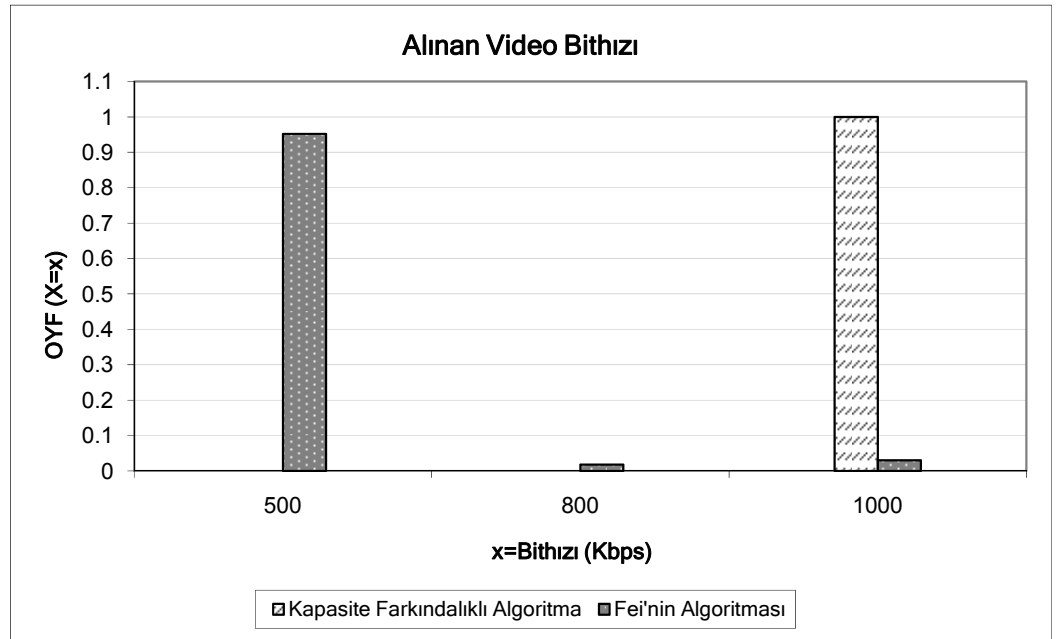
(Fei and Yang, 2007) tarafından kullanılması önerilen algoritma, ağaç kenar uzunluğunun en kısa olmasını sağlamaktadır. Bu algoritmaya göre yapılan deneylerde oluşan ağaçların üzerinde bulunan düğümlerin kaynağa ortalama uzaklığı 1,4 saniye olarak belirlenmiştir. Kapasite farkındalıklı algoritmada ise benzer şekilde düğümlerin kaynağa olan ortalama uzaklığı 2,46 saniyedir. Dolayısıyla, ağaçların gecikme açısından uzunluğu gözönüne alındığında, 1,06 saniyelik bir fark görülmektedir. Ancak akışlandırma başladıktan sonra kaynaktan hedefe giden yolda yer alan her düğüm için video paketlerinin alımı ve iletimi

sırasında geçen süre hesaba katılırsa ağaç üzerindeki paketlerinin video alımı uzun sürecektir.

(Fei and Yang, 2007) çalışmasında oluşturulan ağaçların gecikmeye bağlı olarak en kısa şekilde ancak zıplama sayısı açısından daha uzun olmasının sebebi, ağaç üzerinde bulunan her düğümün kapasitesinin tamamını kullanmaya yönelik çocuk seçiminin bulunmamasıdır. Başka bir deyişle, bu ağaçlar kapasiteyi bir üst sınır olarak kullanır, ancak alt sınır olarak kullanmaz, dolayısıyla bazı düğümlere hiç çocuk bağlanmayabilir. Bu yaklaşım da ağaçların zıplama sayısı bakımından uzamasına neden olur.

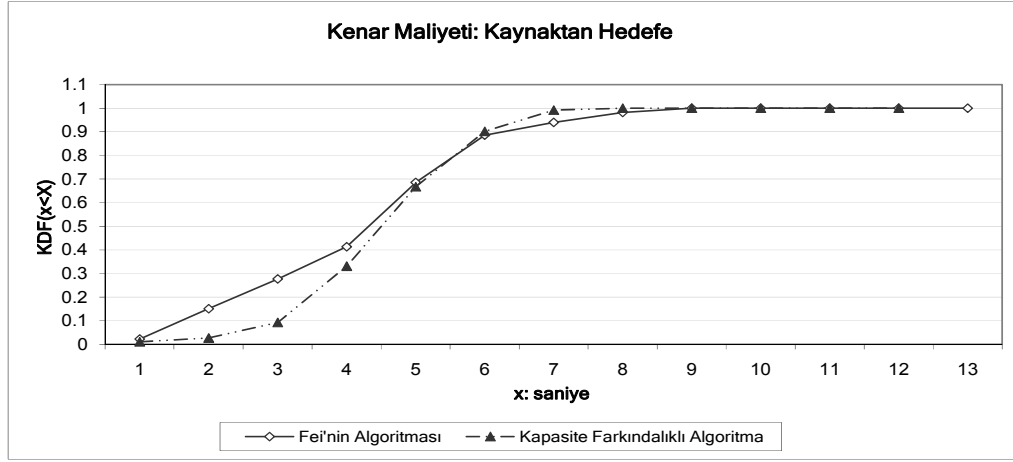
Kapasite farkındalıklı algoritma ile ilgili olarak gerçekleştirilen diğer deney setinde, farklı derecelerde ve farklı kapasitelerde düğümler 500 düğümden oluşan bir ağ topolojisi üzerinde yaratılmış ve oluşan ağaçların karakteristiği ile ilgili ölçümler yapılmıştır. Ağaç üzerinde yer alan düğümler, kendisine bağlı düğümlere toplamda maksimum kendi yükleme kapasitesi ya da aldığı kalite kadar oranda video iletebilir. Deney seti için seçilen düğümlerin kapasiteleri %5'ine 5, %1'ine 4, %15'ine 3, %20'sine 2 ve %50'sine 1 adet düğüm bağlanabilecek şekilde dağıtılmıştır. Her düğümün bir bağlantısı ebeveynine bağlanmak için kullanılmaktadır ancak bu bağlantı kullanılan kapasiteyi etkilemez. 1'den fazla sayıda düğüm bağlanabilecek düğümlerin yükleme bithızı bağlanabilecek çocuk sayısı başına 1000 Kbps olarak seçilmiştir, örneğin 3 adet kapasitesi olan bir düğüm, 3 adet çocuk alabilir ve yükleme hızı 3000 Kbps'dir. 1 adet kapasiteye sahip düğümlerin yükleme kapasiteleri %50'si 500 Kbps, %30'u 800 Kbps ve %20'si 1000 Kbps olmak üzere 3'e ayrılmıştır. Deney setinde, bir düğümün çocuklarına gönderebileceği maksimum bit hızı 1000 Kbps olacak şekilde ayarlanmıştır. Bu parametrelere göre, örneğin 800 Kbps'lık bir yükleme kapasitesine sahip bir düğüme bağlanan bir düğüm, 800 Kbps bithızında video alabilir. 5000 Kbps yükleme kapasitesine sahip bir düğüm ise, çocuk sayısına bağlı olmaksızın tüm çocuklarına 1000 Kbps video gönderebilir. Ancak, tüm çocuklarına 1000 Kbps video gönderebilen bir düğüm, eğer ebeveyninden aldığı bithızı daha düşükse, her çocuğuna bu bithızı kadar veri iletimi yapabilmektedir. Bu özellik, ağaç altyapısına sahip düğümlerin dezavantajı olarak belirtilmiştir (Magharei et al, 2007). Kapasite farkındalıklı algoritmada, ağacın köküne yakın düğümlerin yüksek kapasiteli olarak seçilmesi yaklaşımının, ağaç üzerinde bulunan tüm düğümlere en yüksek bithızında iletim yapılabilmesini sağladığı şekil 8.4'te verilen OYF (olasılık yoğunluk fonksiyonu) grafiğinde görülebilir. Elde edilen verilere göre, önerilen algoritma ile oluşan ağaçlardaki tüm düğümler

en yüksek bithızı olan 1000 Kbps bithızında video alabilmektedir. Ancak, aynı ağ yapısı ve aynı düğümler ile (Fei and Yang, 2007) çalışmasında önerilen yaklaşım ile kurulan ağaçlarda düğümlerin % 95'inin en düşük bithızı olan 500 Kbps bithızında video verisi aldıkları gözlemlenmiştir. Bunun sebebi, ağacın üst katmanlarında, köke yakın düğümler arasında düşük bithızına sahip düğümlerin bulunmasıdır. Bu durum, ağaçta, bu düğümler altında kalan tüm düğümlerin düşük kalitede bithızıyla veri almasına sebep olmaktadır. (Fei and Yang, 2007) yaklaşımıyla oluşan ağaçlarda, düğümlerin %2'si 800 Kbps ve %3'ü 1000 Kbps bithızında veri alabilmektedir.

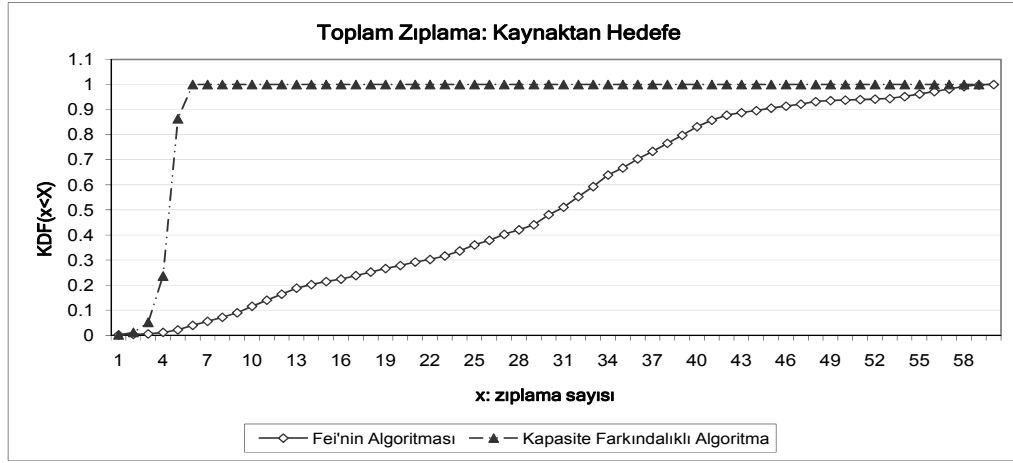


Şekil 8.4. Alınan video bithızı OYF grafiği.

Farklı düğüm kapasitelerinin dağıtıldığı deney setinde, 500 düğüm üzerinde oluşan ağaç karakteristiklerini gözlemlemek amacıyla her iki yöntemle de ağaçlar oluşturulmuş ve elde edilen karşılaştırmalı grafikler şekil 8.5'te verilmiştir. Önerilen algoritmada toplam uzaklık ile ilgili değerler, sistemdeki düğüm kapasiteleri farklılık gösterdiğinde saniye cinsinde benzer, zıplama sayısı cinsinden daha iyi kazanım sağlamıştır. Saniye cinsinden verilen değerler, düğümler arası kenar maliyetlerini belirtmektedir. Video akışlandırma sırasında kaynaktan hedefe giden düğümler üzerinde işlem zamanı hesaplandığında, ağaç üzerindeki düğümlerin (Fei and Yang, 2007) yaklaşımı ile daha uzun süre başlangıç bekleme zamanına sahip olacağı öngörülmektedir.



(a)



(b)

Şekil 8.5. Karşılaştırmalı deney sonuçları.

Deneyle ilgili ortama değerler çizelge 8.1’de verilmiştir.

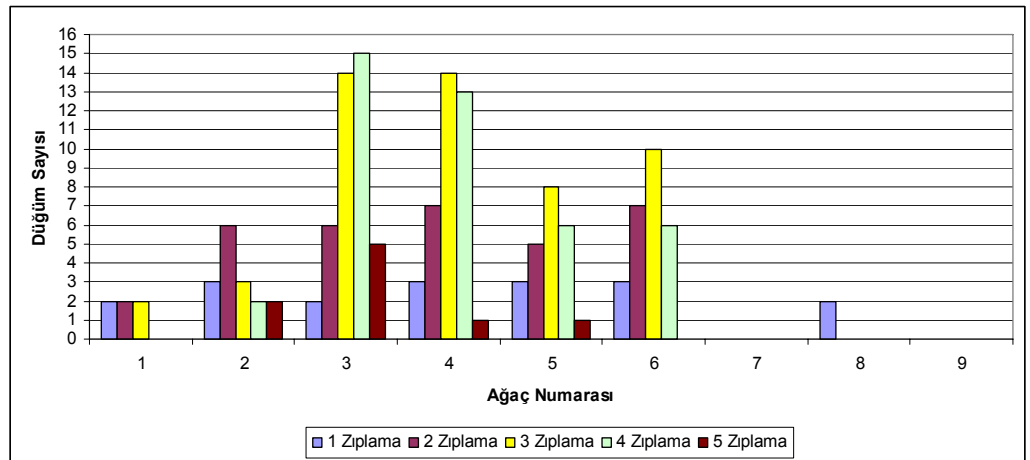
Çizelge 8.1. Karşılaştırmalı ortalama değerler.

	Fei'nin Algoritması	Kapasite Farkındalıklı Algoritma
Ortalama Uzaklık (sn)	2.21	2.22
Ortalama Zıplama Sayısı	30.2	3.8
Ortalama Alınan Kalite	540 Kbps	986 Kbps

8.2.2 Çoklugoönderim ağalarının hiyerarşik katmanlar üzerinde uygulanması

Kapasite farkındalıklı çoklugoönderim ağa oluřturma algoritmasının testleri Chang Roberts algoritmasına benzer řekilde, oluřturulan hiyerarşik katmanlar üzerinde gereklenmiřtir. Tasarlanan çoklugoönderim algoritmasında dögümlerin yükleme hızları kullanıldığđ için bu değler tanımlanmıřtır. Yapılan testlerde dögümlerin yükleme hızı değleri rastgele yaratılmıř; sistemde yer alan her dögüm için bantgeniřliğı değeri 3 farklı tipte kullanıcı grubunu temsilen 200, 400 ve 600 Kbps olarak seçilmiřtir.

Yapılan benzetimlerde ağa oluřturma başarımını test etmek için deney sonucu oluřan tüm ağaların her katmanındaki dögüm sayısı, ağaın her katmanında bulunan dögümlerin ortalama bantgeniřliğı, bir ağa üzerinde bulunan dögümlere baėlı çocuk sayısı ve kaynaktan hedef dögümlere video verisinin ağa üzerinde izleyeceėi yola göre toplam gecikme değleri değlendirilmiřtir. řekil 8.6, 8.7, 8.8 ve 8.9’da sözü edilen başarım ölçümleri ile ilgili grafikler verilmiřtir.

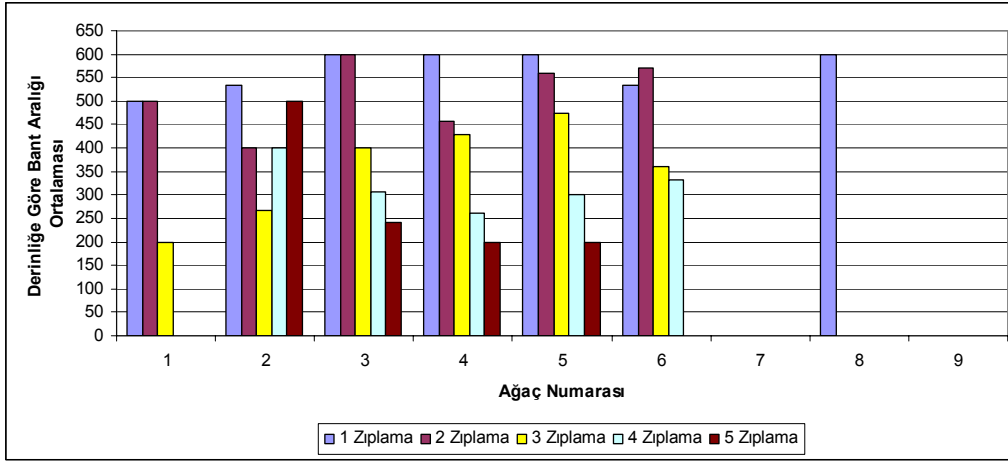


řekil 8.6. Dögüm sayısı – zıplama adedi histogramı.

Grafikler ile ölçüm sonuçları verilen deneyde, sistemde bulunan kaynak dögümlere baėlı olarak oluřan ağalar ile ilgili değler verilmiřtir. Yapılan deneyde 153 dögüm tarafından talep edilen video 9 dögümden bulunmaktadır. Kaynak dögümlerden iki tanesi istemcilere uzak konumda bulunduėu için hiçbir istemciyi çocuk olarak almamıřtır. Sistemde oluřan en büyük ağa, 42 dögümden oluřan ve řekillerde 4 sayısı ile temsil edilen ağatır. řekil 8.6’da verilen grafiėe göre, Chang Roberts algoritmasında karřılařılan en büyük sorun kaynak

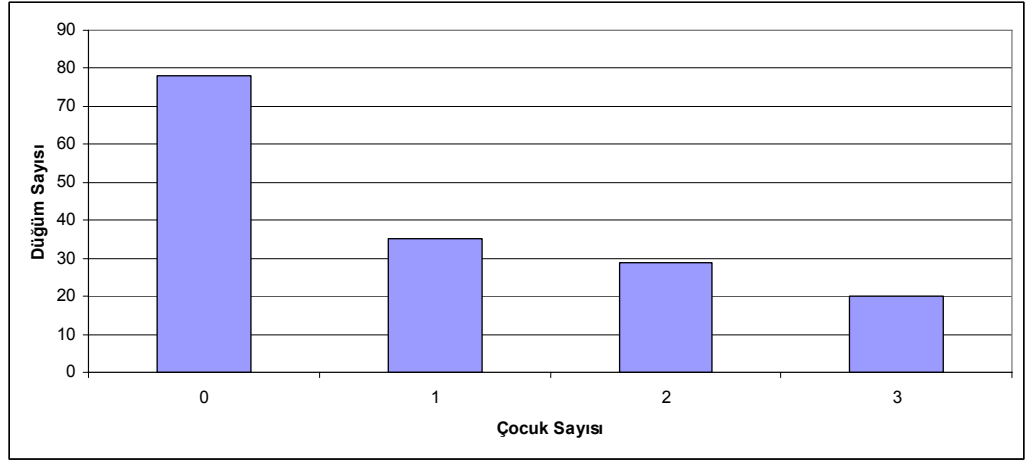
düğümlere bağlı çocuk sayısının kapasite farkındalıklı algoritmanın her düğüm için yükleme hızını gözönüne alması ile kontrol altında tutulduğu görülmektedir. Şekle göre, sistemde oluşan tüm ağaçlar için 1 zıplamayla ulaşılacak düğüm sayısı en fazla 3 olarak görülmektedir.

Şekil 8.7’de görüldüğü üzere, zıplama başına bantgenişliği ortalama değerleri giderek düşmektedir. Bu durum, hedeflendiği gibi düşük yükleme kapasitesine sahip düğümlerin ağacın yaprakları ya da yapraklara yakın konumlandığını göstermektedir.

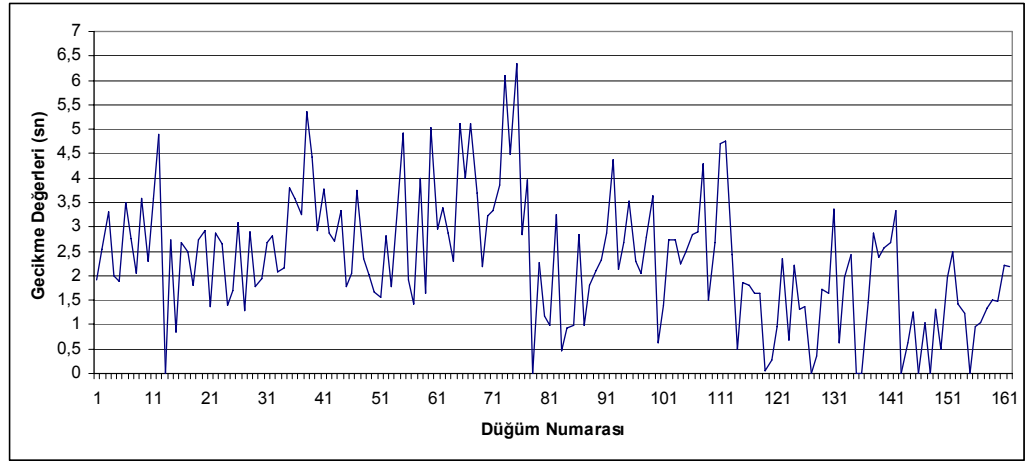


Şekil 8.7.Düğüm Sayısı – zıplama sayısına bağlı ortalama bantgenişliği değerleri

Şekil 8.8’de verilen grafikte görüldüğü üzere, sistemde en fazla çocuk sayısına sahip düğümlerin 3 adet çocuğu bulunmaktadır. Şekil 8.9’da verilen gecikme değerleri, düğümlerin üzerinde bulundukları ağacın kaynağına olan uzaklığıdır, bu uzaklık sadece akışlandırma başlangıç bekleme zamanını (initial waiting time) etkiler. Düğümlerin video verisini alacağı ebeveynlerine olan gecikme cinsinden uzaklığı, grafikte verilen değerlerden daha düşüktür.



Şekil 8.8. Düğüm sayısı – çocuk sayısı histogramı.



Şekil 8.9. Ağaç üzerindeki düğümlerin gecikme cinsinden kaynağa uzaklıkları.

8.2.3 Çoklugoönderim ağaçları ile yapılan video akışlandırma çalışmaları

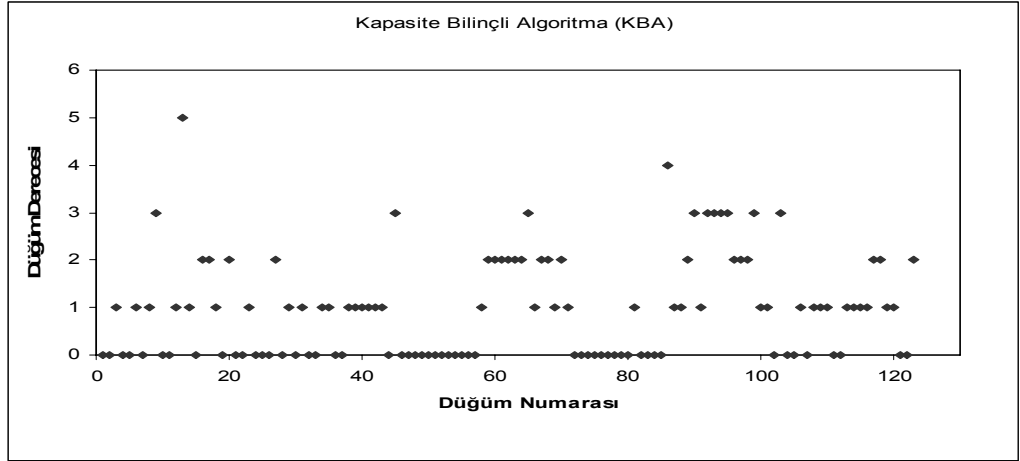
Tasarlanan algoritmanın video akışlandırma oturumunda olan başarımını görmek üzere, NS2 benzetim ortamında oluşturulan hiyerarşik katmanlar ve çoklugoönderim ağaç oluşturma algoritması üzerinde video akışlandırma çalışmaları yapılmıştır (Sayıt et al, 2009).

500 düğümden oluşan ağ topolojisinde grupların üye sayısı 20 ile sınırlı tutulmuş; yapılan deneylerde en fazla 3. hiyerarşik katmana kadar çıkıldığı görülmüştür. Video talep ve video arama süreçleri tamamlandıktan sonra kapasite farkındalıklı çoklugoönderim ağaçları oluşturulmuştur. Kaynak konumundaki düğümler en üst kalitede videoyu gönderebilecek yükleme hızına sahiptir. Kaynak

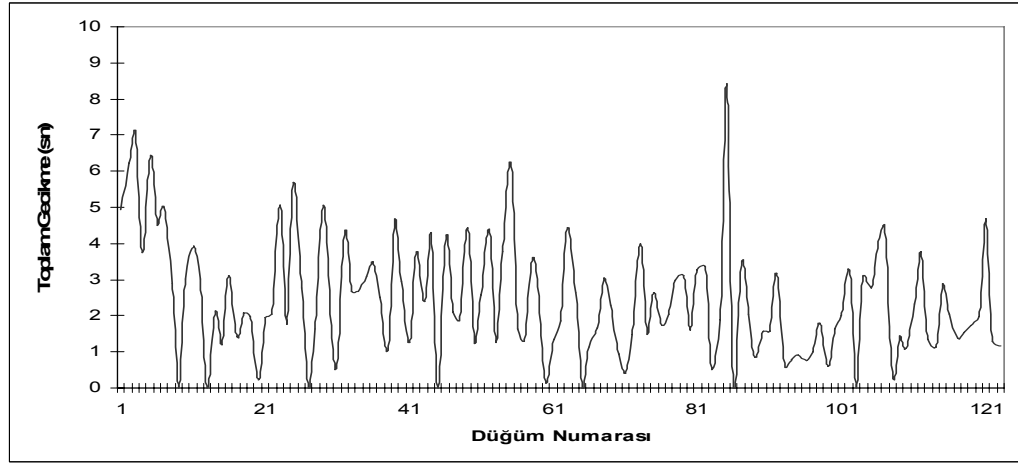
tarafından en yüksek kalitede video gönderimiyle başlayan video akışlandırma oturumu sırasında ara düğüm konumunda olan düğümler video paketlerini aldıktan hemen sonra tüm çocuklarına iletir. Eğer bir ara düğümün yükleme kapasitesi aldığı videoyu aynı kalitede gönderecek kadar yüksek değil ise bantgenişliğine uygun şekilde videonun bir ya da birden fazla genişleme katmanını atarak çocuklarına iletir. Düğümler aynı zamanda ağ üzerindeki sıkışıklığı gözönünde bulundurarak video kalite katmanlarını eklemekte ya da çıkarmaktadır.

Deneylerde yer alan düğümler yükleme kapasiteleri açısından 5 farklı grupta yer almaktadır: Düğümlerin %20'si 500 Kbps, %20'si 800 Kbps, %30'u 1 Mbps, %20'si 2 Mbps ve %10'u 5 Mbps yükleme hızına sahiptir. Video verisi 1 taban ve 2 genişleme katmanından oluşmaktadır ve bithızları sırasıyla 500 Kbps, 800 Kbps ve 1 Mbps'dir. Ağaç oluşturma aşamasında bir düğüme bağlanacak çocuk sayısının tespiti için videonun kalite katmanlarının kodlama hızları ve düğümün yükleme kapasitesi birlikte gözönüne alınmıştır. Örneğin 2 Mbps yükleme kapasitesine sahip bir düğüme bağlanacak çocukların en yüksek kalitede (1 Mbps) video alabilmesi için, çocuk sayısı 2 ile sınırlandırılmıştır.

Şekil 8.10'da verilen parametrelere bağlı olarak kurulan ağ topolojisi üzerinde oluşturulan ağaçlarda yer alan düğümlerin dereceleri, bir başka deyişle düğümlere bağlı çocuk sayıları verilmiştir. Şekilde görüldüğü üzere, ağaç üzerinde bir düğüme en fazla bağlı çocuk sayısı 5 ile sınırlıdır. Ağaçta yer alan ve 5 çocuğa sahip olan düğüm, sistemde bulunan tek kaynaktır. Grafikler, tek kaynaktan oluşan bir ağaçtan elde edilen verileri içermektedir. Şekil 8.11'de aynı ağacın üzerinde yer alan düğümlerin kaynağa uzaklıkları verilmiştir. İki grafik incelendiğinde oluşan ağacın dengeli ve gecikme değerlerinin belli sınırlar içinde kaldığı gözlenmektedir.

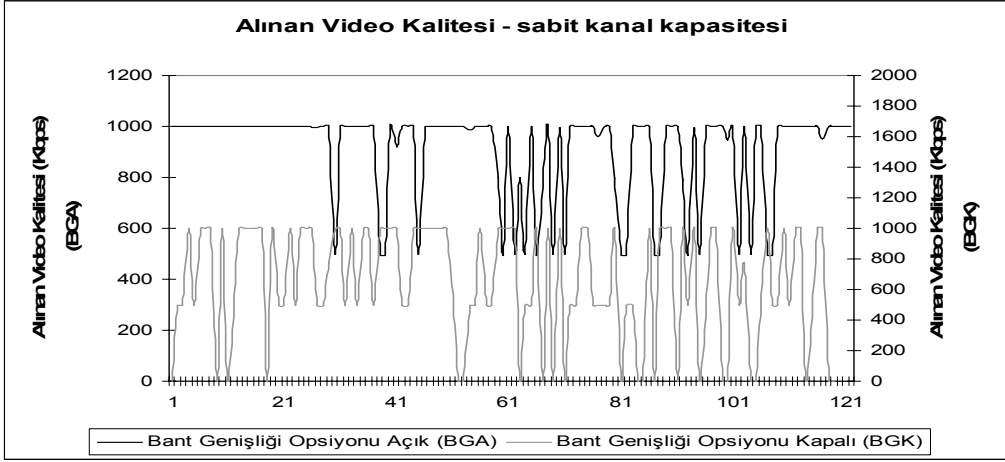


Şekil 8.10. Düğüm derecesi.

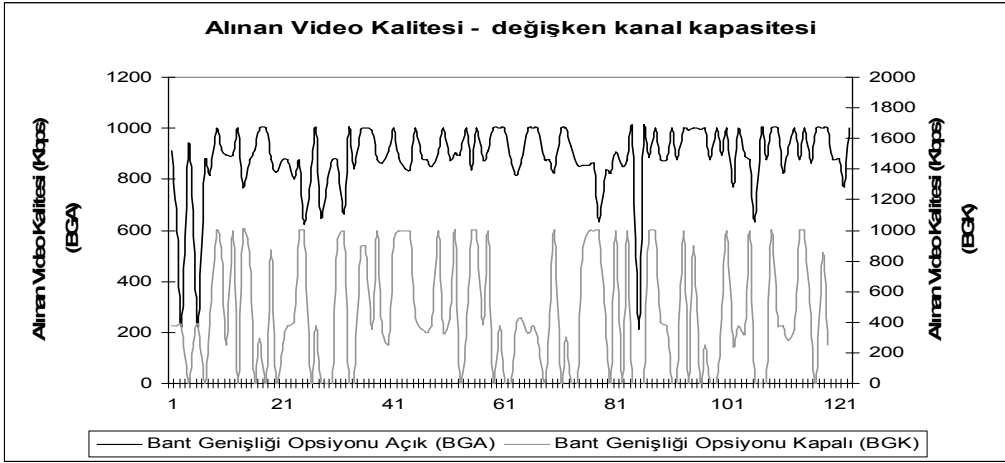


Şekil 8.11. Düğüm gecikme değerleri.

Verilen parametreler ve ağ topolojisi kullanılarak yapılan 2. deney setinde ise ağ üzerinde yer alan kanal kapasiteleri rastgele değiştirilmiş ve bu durumda kalite uyarlaması ile birlikte tasarlanan sistemde akışlandırma kazanım sonuçları incelenmiştir. Buna ek olarak, önerilen çoklugoönderim ağaç oluşturma algoritmasının kazanımını karşılaştırmalı olarak test edebilmek amacıyla, kapasite farkındalıklı ancak ağaç üzerinde yerleştirme yapılırken bantgenişliği değerlerini dikkate almayan bir algoritma kodlanmıştır. Her iki algoritmanın sabit ve değişken kanal koşullarında karşılaştırmalı kazanım sonuçları şekil 8.12 ve 8.13'te verilmiştir.



Şekil 8.12. Sabit kanal kapasitesi için alınan video kalitesi.



Şekil 8.13. Değişken kanal kapasitesi için alınan video kalitesi.

Sabit kanal koşulu için verilen grafikte alınan kalitede gözlemlenen dalgalanmaların sebebi hatta sıkışıklık olması nedeniyle gerçekleşen paket kayıplarıdır. Yüksek kapasiteli düğümlerin ağacın köküne yakın konumlandırılmasının getirdiği avantaj, karşılaştırmalı grafikte verilen, alınan video kalitesi değerlerinde görülmektedir. Bu opsiyonun kullanılmadığı algoritmada, en yüksek kalitede video izleyebilen düğüm sayısı, bantgenişliği opsiyonunun açık olduğu algoritmaya kıyasla çok daha az sayıdadır. Değişken kanal kapasitesi koşulunda, düğümlerin yükleme hızında oluşan dalgalanmalardan dolayı, düğümler tarafından alınan video kalitesi değişkenlik göstermektedir. Bantgenişliğinin gözönünde bulundurularak ağaca konumlandırma yapıldığı durumda elde edilen kalitenin genelde 850 – 1000 Kbps olduğu gözlemlenirken, bantgenişliğinin dikkate alınmadığı durumda alınan video kalitesinin 400 – 600 Kbps dolayında olduğu görülmektedir. Düğümleri ağaca konumlandırma süreci

bantgeniřlięi dikkate alınmadan ağaca yapıldığında ağaç uzunluęu artmakta, özellikle paket kayıplarının sık olduęu ağ kořullarında zıplama sayısı arttıkça paket kayıp olasılığı da yükselmektedir. Bu durum, alınan video kalitesinin düşmesine sebep olmaktadır.

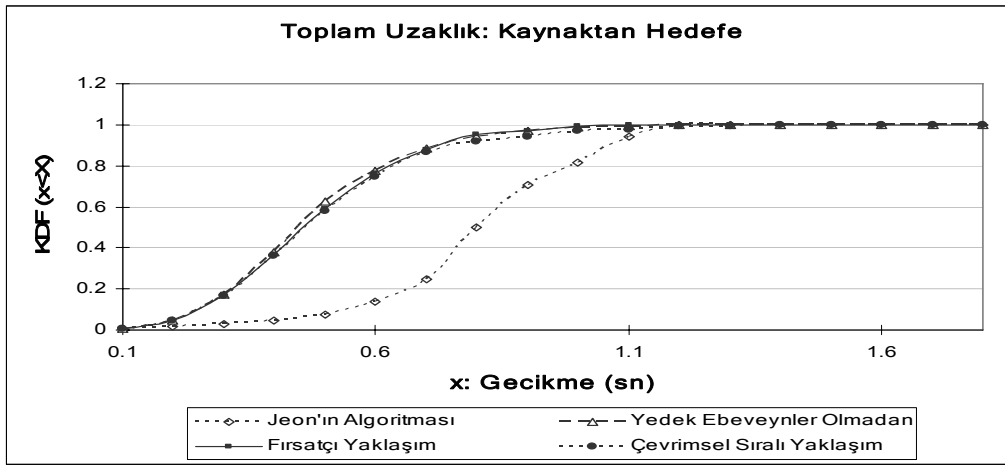
8.3 Yedek Ebeveynler ile Hata Dayanıklı Akışlandırma Test Sonuçları

Bu bölümde, tezin beřinci ve altıncı bölümünde açıklanan hata dayanıklılığına sahip ağaçlarda akışlandırma ile elde edilen test sonuçları verilecektir. NS2 benzetim ortamı üzerinde yapılan testlerde, GT-ITM tarafından üretilen 250 düęümden oluřan ağ kullanılmıştır. Düęüm dereceleri farklı olarak, 1 ile 5 arasında seçilmiştir. 1 kapasiteye ait düęümler için yükleme hızı farklı kalitede bitakışlarını temsil edecek řekilde 1000, 800 ya da 500 Kbps olarak deęiřmektedir. Her düęüm, 5 saniyelik video verisini tutacak řekilde bir bařlangıç bekleme süresine alınır. Böylece, ebeveynini kaybeden düęüm 5 saniye içerisinde yeni bir ebeveyne baęlanana kadar kesintisiz video izleyebilmektedir. Bu bölümde verilen materyal (Sayıt et al., 2010b)'de de verilmiştir.

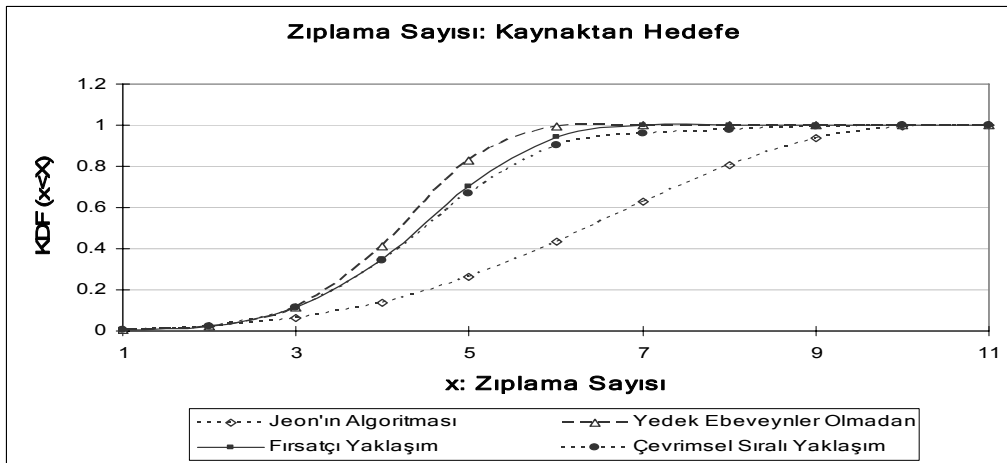
Yapılan testlerde, önerilen yöntemin kazanım sonuçlarını deęerlendirebilmek için literatürde yer alan benzer bir çalışma ile karřılařtırmalı deneyler yapılmıştır. Tezin ikinci bölümünde söz edilen (Jeon et al, 2008) çalışmasında her düęüm, sisteme girerken kendisine bir yedek ebeveyn seçmektedir. Ebeveyninde hata olan düęüm, önceden seçmiş olduęu yedek ebeveyne baęlanarak akışlandırmayı sürdürür. Yedek ebeveyne baęlandıktan sonra tekrar ağacın kökü ile iletiřime geçerek yedek hücresi bulunan düęümlerin listesini alır, yeniden paket dolařım zamanını ölçümler ve kendisine en yakın düęümü yedek ebeveyn seçerek bu seçimi yedek ebeveynine mesajla bildirir. Akışlandırma sırasında oluřan hataların ardından yedek ebeveyn seçiminde ağaç üzerinde döngülerin oluřmaması için belirli kurallar çalıştırılır. Her düęüm, alt ağacında bulunan düęümlerin listesini sürekli olarak günceller ve yedek ebeveynini bu listeyi kontrol ederek alt ağacı içinden seçmez. Bunun yanıřıra, ebeveyninin ebeveynine en yakın konumda bulunan düęüm, yedek ebeveyn olarak bu düęümü seçer. Bir düęüme en yakın torunun belirlenmesi için, torunlar arasında belirli aralıklarla mesajlařma yapılır ve en yakın torun bilgisi güncel tutulur. Karřılařtırma amacıyla yapılan testlerde, (Jeon et al, 2008) çalışmasında öne sürölen algoritmalar NS2 benzetim ortamı üzerinde kodlanmıştır. Yapılan deneylerde altta yatan ağ mimarisi, düęüm sayısı, düęümlerin kapasitesi,

düğümelerin tamponunda bulunan veri miktarı gibi tüm deney parametreleri önerilen algoritma ile yapılan testlerde kullanılan parametrelere eşittir.

Şekil 8.14 ve 8.15'te, verilen parametreler doğrultusunda oluşan ağaçlar ile ilgili olarak alınan toplam gecikme ve zıplama sayısı değerleri KDF (kümülatif dağılım fonksiyonu) grafiği ile verilmiştir. Şekilde önerilen sistemde yedek ebeveynler olmadan oluşturulan ağaçlar ile ilgili veriler, Fırsatçı ve Çevrimsel Sıralı yedek ebeveyn seçimlerinin getirdiği ekyükün oldukça kısıtlı olduğunu göstermektedir. Hem gecikme hem zıplama adedi açısından bakıldığında, önerilen algoritmanın (Jeon et al, 2008) tarafından önerilen yaklaşımdan daha iyi sonuçlar verdiği görülmektedir.



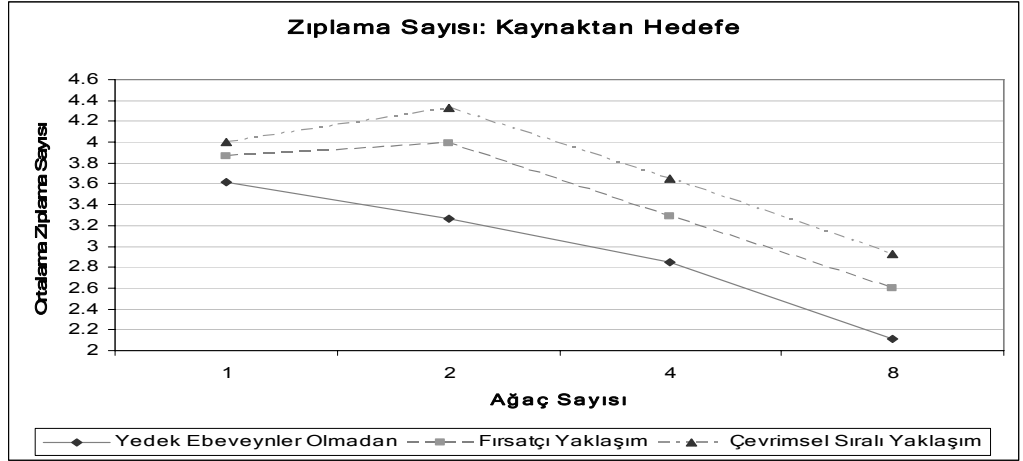
Şekil 8.14. Kurulan ağaçlarda karşılaştırmalı gecikme KDF grafiği.



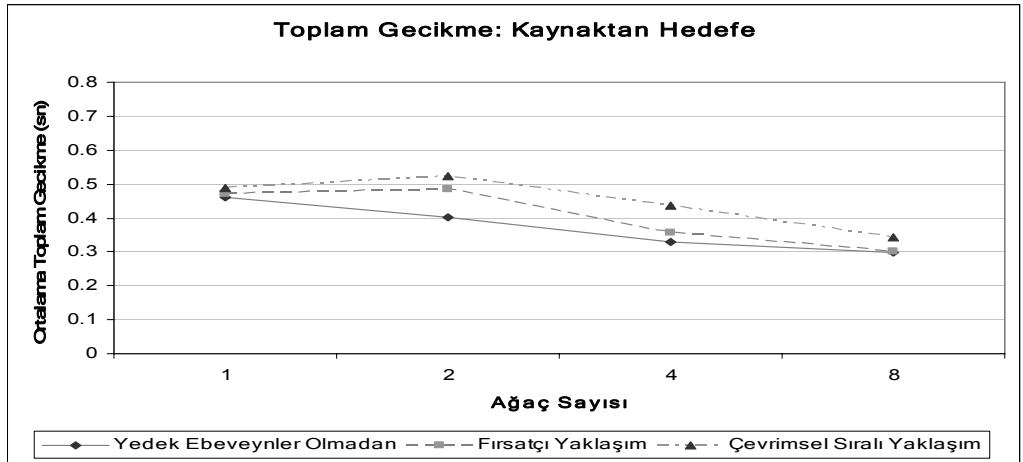
Şekil 8.15. Kurulan ağaçlarda karşılaştırmalı zıplama KDF grafiği.

Şekil 8.14 ve 8.15'te sistemde 1'den fazla sayıda kaynak bulunduğu durumda oluşan ağaçlar ile ilgili veriler verilmiştir. Önerilen algoritma fazla

sayıda kaynak bulunduğunda da çalışabilmekte ve birden fazla sayıda ağaç oluşturmaktadır. Yapılan deneylerde Çevrimsel Sıralı yaklaşımıyla oluşan ağaçların Fırsatçı yaklaşıma göre daha kısa olduğu gözlemlenmiştir; bunun sebebi, bir gruba video taşıyacak olan bir düğümün Fırsatçı yaklaşım ile büyük bir kısmını yedek olarak ayırması ve dolayısıyla grup içerisinde daha uzun ağaçlar oluşabilmesidir. Ancak bu etki, şekillerde görüldüğü üzere, oldukça sınırlıdır.



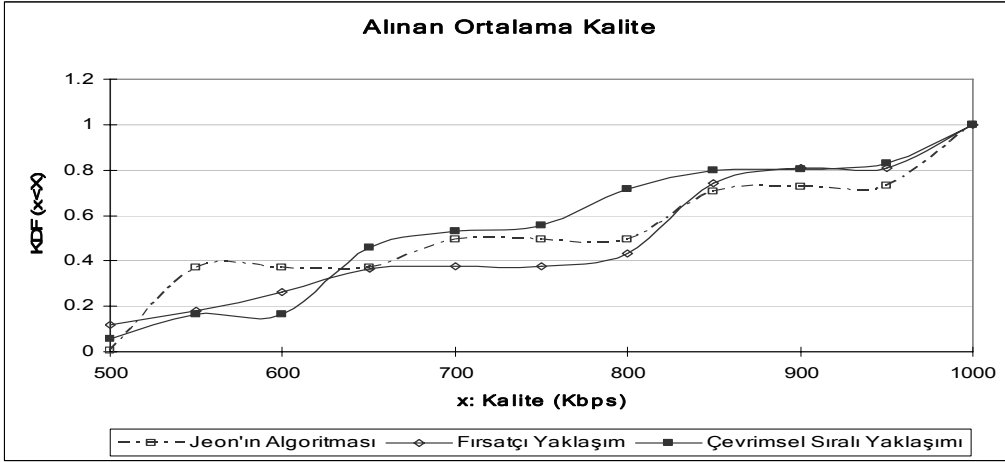
Şekil 8.16. Ağaç sayısına bağlı ortalama zıplama sayısı.



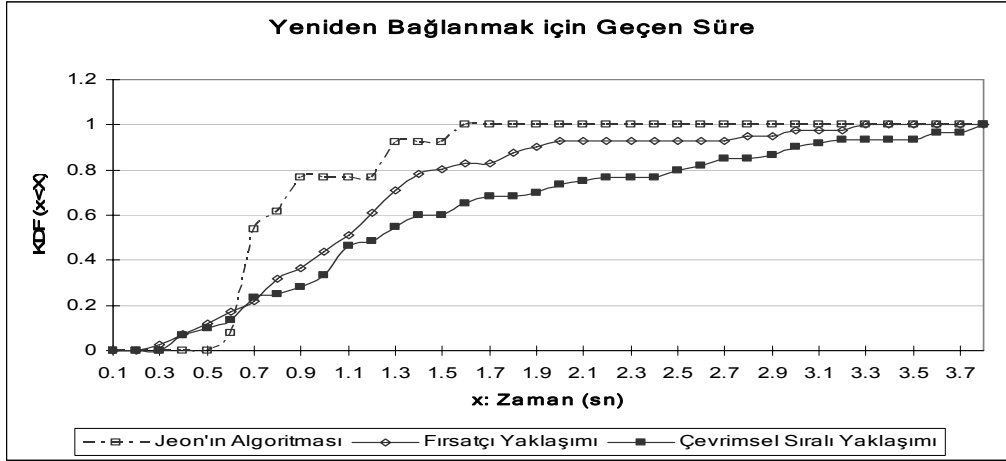
Şekil 8.17. Ağaç sayısına bağlı ortalama toplam gecikme.

Şekil 8.16'da verilen KDF çizgeleri, düğümler üzerinde düşük oranda (%10 ile %30 arası) gerçekleşen hata oranına ait grafiklerdir. Hata oranı düşük olduğunda, alınan kalite açısından en iyi sonuçları sırasıyla Fırsatçı yaklaşım, Çevrimsel Sıralı yaklaşım ve (Jeon et al, 2008) çalışmasının yaklaşımı vermiştir. (Jeon et al, 2008) tarafından önerilen algoritmada her düğüme kesin olarak belirlenen bir adet yedek ebeveyn olduğu için yeniden bağlantı sürelerinde en iyi

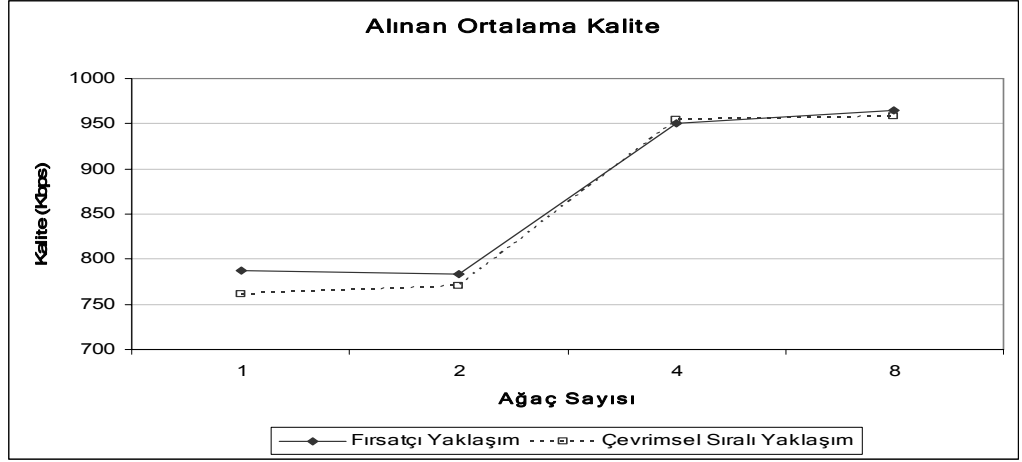
sonuçlar çıkmıştır. Fırsatçı yaklaşımın Çevrimsel Sıralı yaklaşıma göre daha iyi sonuçlar vermesinin sebebi, yedek olarak seçilen düğümlerde daha çok boş hücre olması ve kendisine bağlanmak isteyen düğümlerin çoğunu kabul etmesidir. Oysa Çevrimsel Sıralı yaklaşımında fazla sayıda düğüm, sadece ilk başvuran düğümü çocuk olarak kabul eder, dolayısıyla ebeveyninde hata olan düğümler genellikle birden fazla yedek ebeveyne başvuruda bulunur. Şekil 8.18.c’de yer alan grafikte, sistemde birden fazla sayıda kaynak bulunduğu durumda düğümlerin aldığı kalitedeki yükselme görülmektedir.



(a)



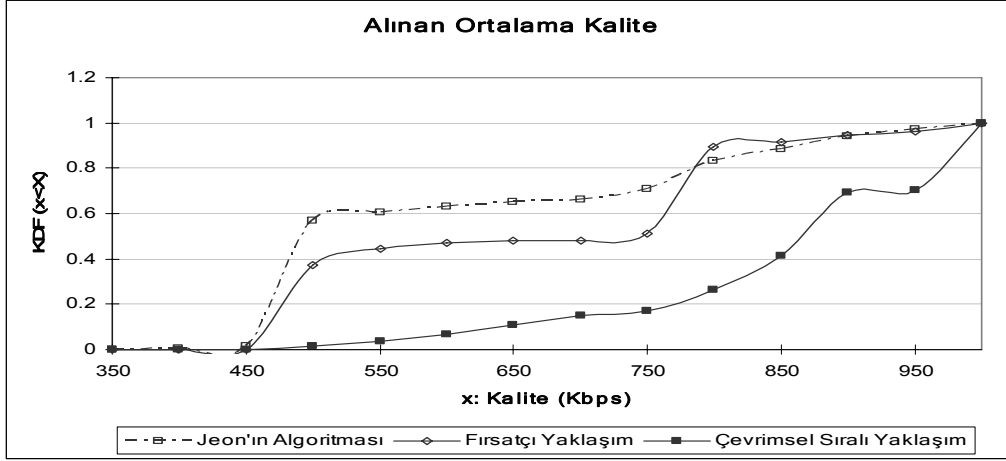
(b)



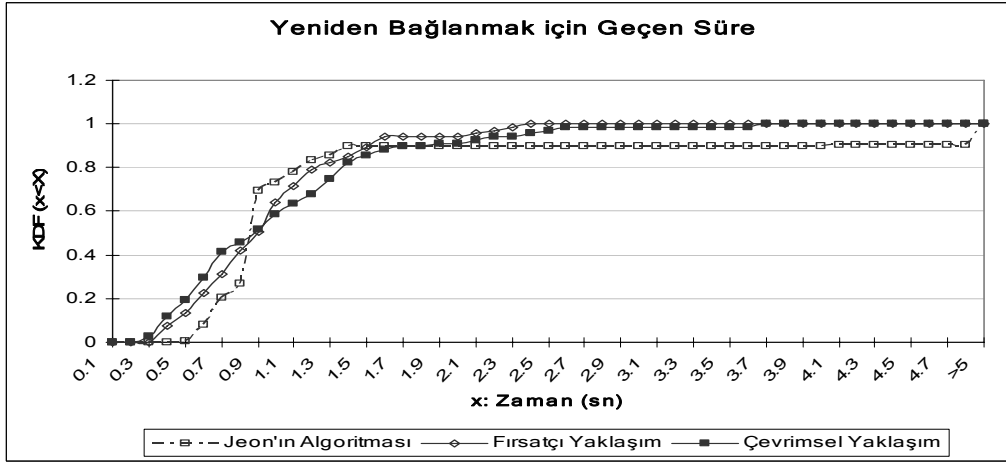
(c)

Şekil 8.18. Düşük seviye hata oranına ilişkin grafikler a) 1 ağaç için alınan ortalama kalite, b) Yeniden bağlanmak için geçen süre, c) Ağaç sayısına bağlı alınan ortalama kalite.

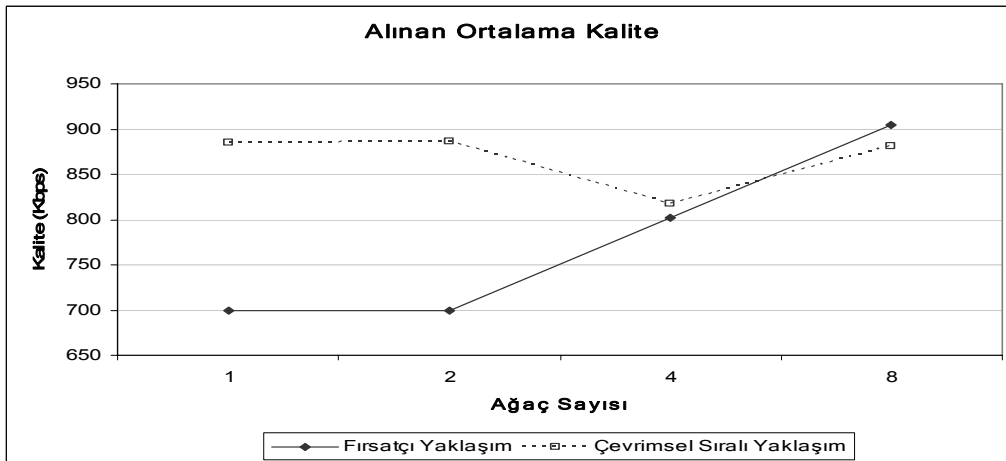
Düğümelerde orta seviye hata oranı (%30 - %50 arası) ile karşılaşıldığında ölçümlenen sonuçlar şekil 8.19’da KDF çizgesi halinde verilmiştir. Yapılan deneylerde çıkan istatistiki sonuçlara göre, düğümler tarafından alınan kalitede önerilen algoritmanın (Jeon et al, 2008) yaklaşımına göre daha yüksek olduğu, en iyi kalitenin ise Çevrimsel Sıralı yaklaşımı ile kurulan ağaçlardan alındığı gözlemlenmiştir. Bunun en önemli sebebi, Fırsatçı yaklaşım ile belli bir hata oranını geçen bir sistemde yedek ebeveynlerin de sistemden çıkmaya başlaması ile birlikte kalan yedek ebeveynlerin düşük kalitede bağlantı kabul etmek zorunda kalmasıdır. (Jeon et al, 2008) tarafından önerilen yaklaşım ile, bu orandaki hata ile karşılaşıldığında düğümlerin %10’unda video görüntüsü kesilmiştir. Bunun sebebi bir yedek ebeveyne bağlanıldığında yeni bir yedek ebeveyn bulamadan ebeveyninin sistemden çıkması ve yeniden bağlanmak için gereken sürenin uzun olmasıdır.



(a)



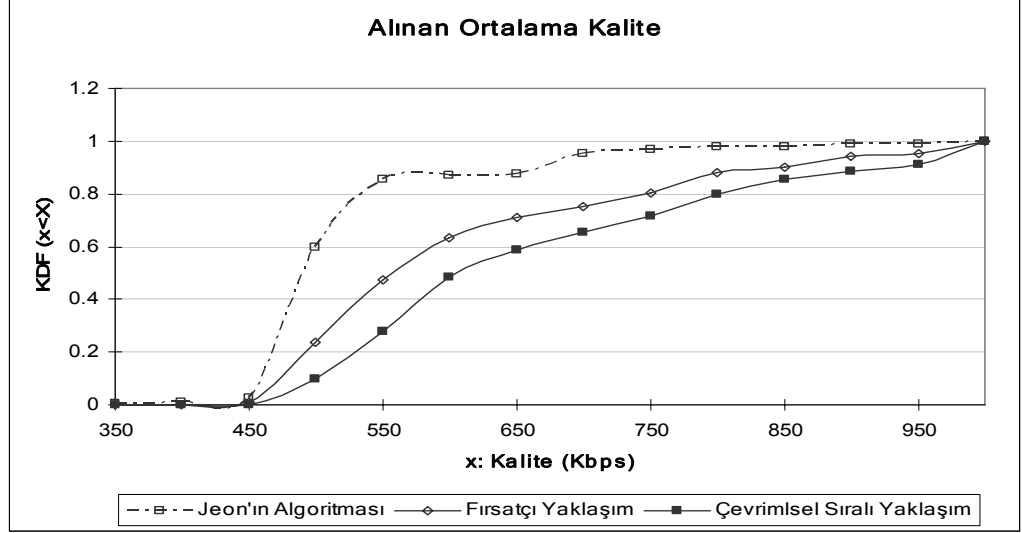
(b)



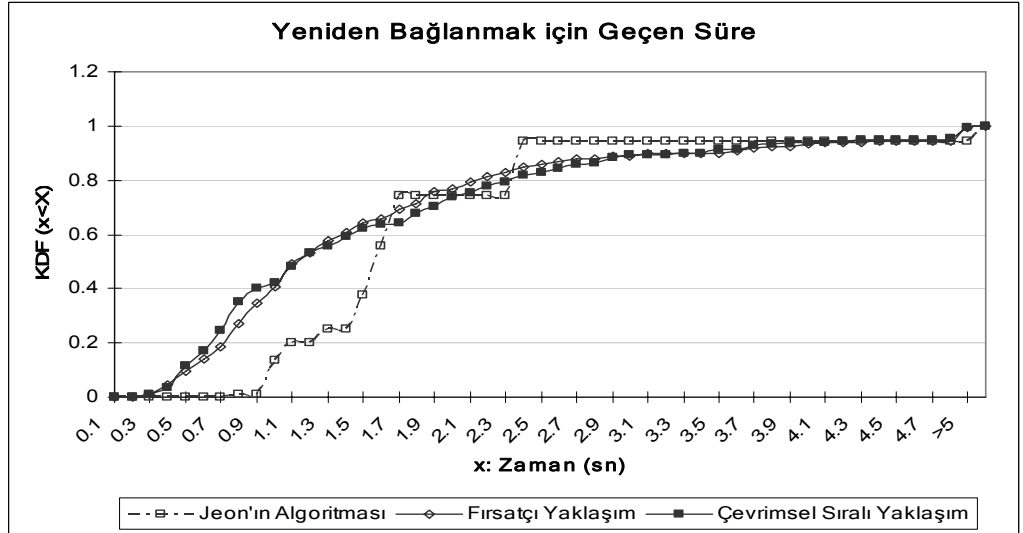
(c)

Şekil 8.19. Orta seviye hata oranına ilişkin grafikler a) 1 ağaç için alınan ortalama kalite, b) Yeniden bağlanmak için geçen süre, c) Ağaç sayısına bağlı alınan ortalama kalite.

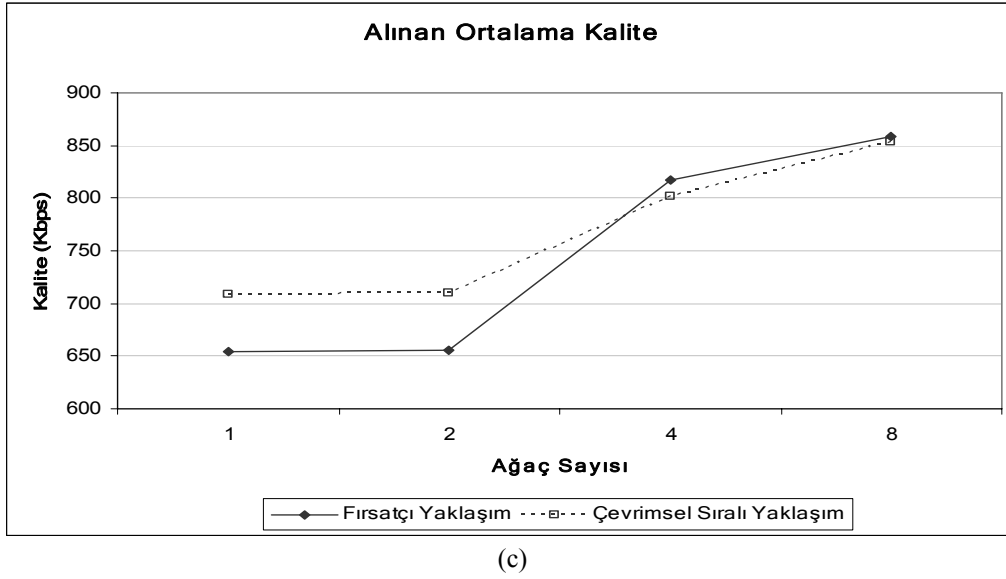
Yüksek seviye hata oranına (%50 ile %75 arası) sahip bir ortam için akışlandırma sonuçları şekil 8.20'de verilmiştir. Orta seviye hata oranına benzer şekilde, en iyi sonuçlar Çevrimsel Sıralı yaklaşımı ile elde edilmiştir.



(a)



(b)



Şekil 8.20. Yüksek seviye hata oranına ilişkin grafikler a) 1 ağaç için alınan ortalama kalite, b) Yeniden bağlanmak için geçen süre, c) Ağaç sayısına bağlı alınan ortalama kalite.

Yapılan deneyler ile birlikte önerilen sistemin sıkışık ağ üzerinde yüksek hata oranlarında bile ayakta kaldığı ve düğümlerin kesintisiz video izleyebildikleri raporlanmıştır. Çevrimsel Sıralı yaklaşım, özellikle orta ve yüksek oranda düğüm çıkışlarının yaşandığı görevdeş ağlar için daha iyi bir hata dayanıklılığı sunmaktadır. Fırsatçı yaklaşım ise, yüksek hata oranında az sayıda olan yedek düğümlerde tek noktada hata ile Çevrimsel Sıralı yaklaşıma göre daha düşük bir kazanım sağlamıştır. Her iki yaklaşım da, (Jeon et al, 2008) tarafından önerilen algoritmadan hem ağacın yeniden yapılandırılması, hem ağaç uzunluğu, hem de alınan kalite bakımından daha iyi sonuçlar vermiştir. Buna ek olarak, iki sistemin mesaj karmaşıklığı karşılaştırıldığında çizelge 8.2'deki değerler elde edilmiştir. Çizelgedeki değerler herbir düğümün 1 saniyede gönderdiği ortalama mesaj sayısıdır. Çizelge 8.2'de yer alan ortalama ağaç bakım mesajları sayısı (Jeon et al, 2008) çalışması için yedek ebeveyn olarak seçilmemesi gereken düğümlerin (düğümün alt ağacında yer alan düğümler) bildirimi, ebeveynin ebeveynine en yakında bulunan düğümün kardeşler arasında belirlenmesi, bağlantı talebi ve akışlandırma sırasında ölçülen RTT değerlerini içermektedir. Kapasite farkındalıklı algoritmada ise ağaç bakım mesajları sadece bağlantı talebi ve bağlantı taleplerinin reddedilmesini içeren mesajlardır. Yapılan deneylerde önerilen algoritmanın mesaj karmaşıklığının oldukça düşük olduğu görülmektedir. Bir düğümün sistemde olup olmadığını kontrol için gönderilen mesajların önerilen algoritmada daha fazla olmasının sebebi, her düğümün birden fazla sayıda yedek ebeveyninin sistemde olduğuna dair kontrol yapmasıdır. Bu kontrol, sistemden

çıkan herhangi bir yedek ebeveynin listeden çıkartılarak hata sırasında bağlanmaya çalışılmasını engellemek için yapılmaktadır.

Çizelge 8.2. Farklı türdeki mesajların karşılaştırmalı ortalama değerleri.

	Ortalama Ağaç Bakım Mesajları	Ortalama Kontrol Mesajları
Jeon'ın Algoritması	3.58	0.88
Önerilen Algoritma	0.01	1.69

8.4 Farklı Yedek Ebeveyn Seçim Yaklaşımı ile Ağaç Oluşturma ve Hata Dayanıklı Akışlandırma Test Sonuçları

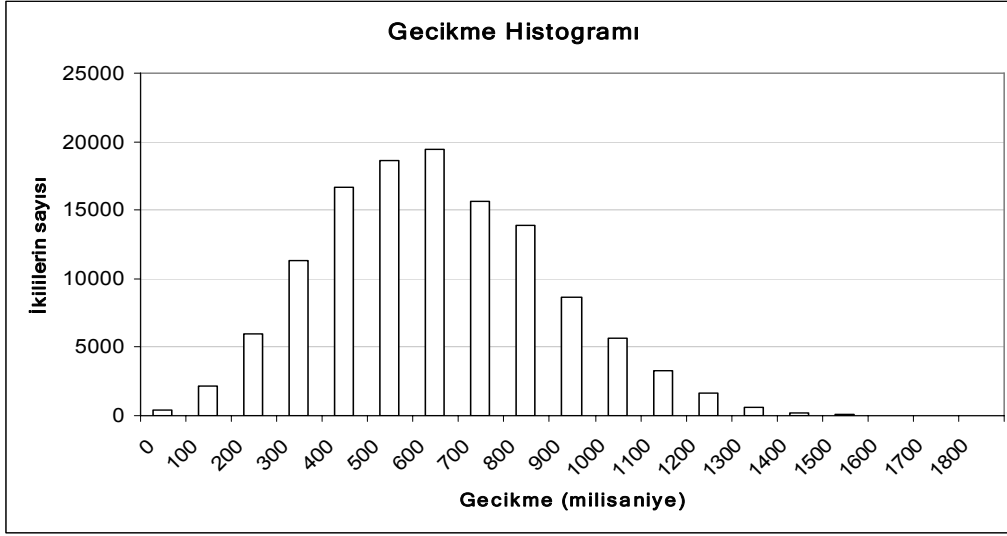
Bu bölümde, tezin yedinci bölümünde tasarlanan ve analizi yapılan yedek ebeveyn seçim yaklaşımı ile ilgili testler yer almaktadır. Bu yaklaşıma göre, ağaç üzerinde kök hariç her düğüm kapasitesinin yarısını yedek olarak ayırır. Ağaç kökü ise sistemdeki düğüm kapasitelerini gözönünde bulundurarak yedek hücre ayırmayabilir. Ayrıntılar tezin yedinci bölümünde yer almaktadır. Bu bölümde verilen materyal (Tunalı ve Sayıt, 2011)'de de verilmiştir.

Ağacın $l+1$. katmanında bulunan bir düğümün yedek ebeveyn listesinde 0'dan l 'ye kadar olan tüm katmanlarda bulunan düğümler yer alır. Her düğüm, kendi yedek ebeveyn listesindeki düğümleri yakınlığına göre sıralar, bu durum aynı düğüme başvuruda bulunan düğüm sayısını azalttığı için bağlantı deneme sayısını da azaltır ve ağaç kısa sürede yeniden yapılır. Bir düğüm yedek düğüm ile arasındaki mesafeyi akışlandırma sırasında gönderdiği yedek ebeveynin sistemde olup olmadığını kontrol eden mesajlar ile tespit edebilmektedir. Akışlandırma sırasında ebeveynde hata tespiti ve yedek ebeveyne bağlanma süreci tezin altıncı bölümünde anlatılan şekilde yapılmaktadır.

Deney setinde, GT-ITM modülü tarafından üretilen 500 düğüm ve ağ alt yapısı üzerinde testler yapılmıştır. Ağaç üzerinde birbirine bağlı 2 düğüm bu modül tarafından yaratılan fiziksel ağ üzerinde en kısa yol üzerinden iletim yapar. Şekil 8.21'de deneylerde yer alan düğümler arası gecikme değerleri histogramı verilmiştir.

Karşılaştırmalı deneyler için (Fei and Yang, 2007) çalışması ile benzer koşullarda testler yapılmış ve elde edilen sonuçlar istatistiki şekilde

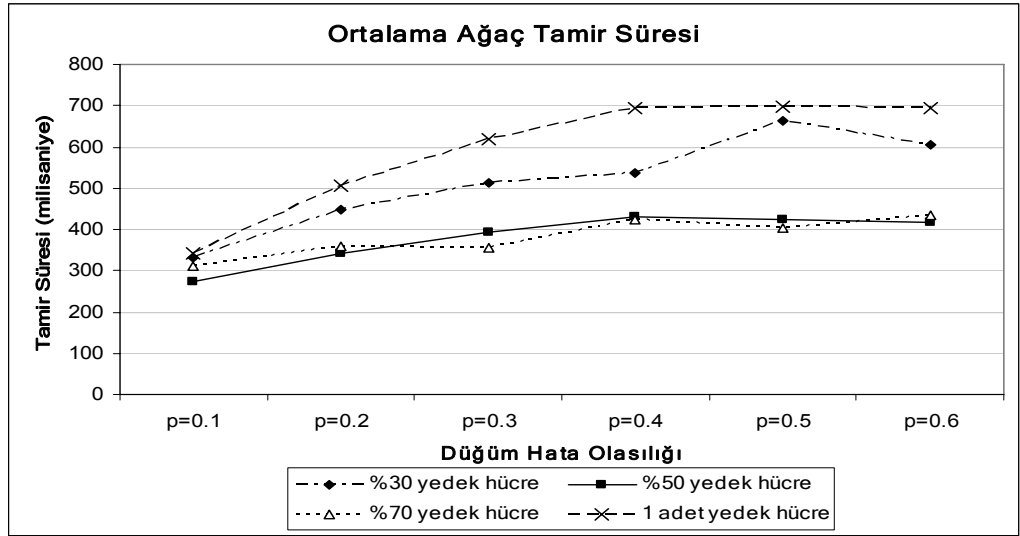
raporlanmıştır. Düğüm kapasiteleri (Fei and Yang, 2007) çalışmasında olduğu gibi 2, 4 ve 6 olarak uniform bir şekilde dağıtılmıştır. Düğümler üzerindeki iletim işlem zamanı 60 milisaniye ve üzerine çıktığı zaman tez kapsamında önerilen algoritmanın daha iyi sonuçlar verdiği gözlemlenmiştir. Deneylerde, akışlandırma sırasında bazı düğümlerde hata oluşturulmuş ve bu deneyler düğüm hata olasılıklarına göre sınıflandırılarak sonuç grafikleri oluşturulmuştur.



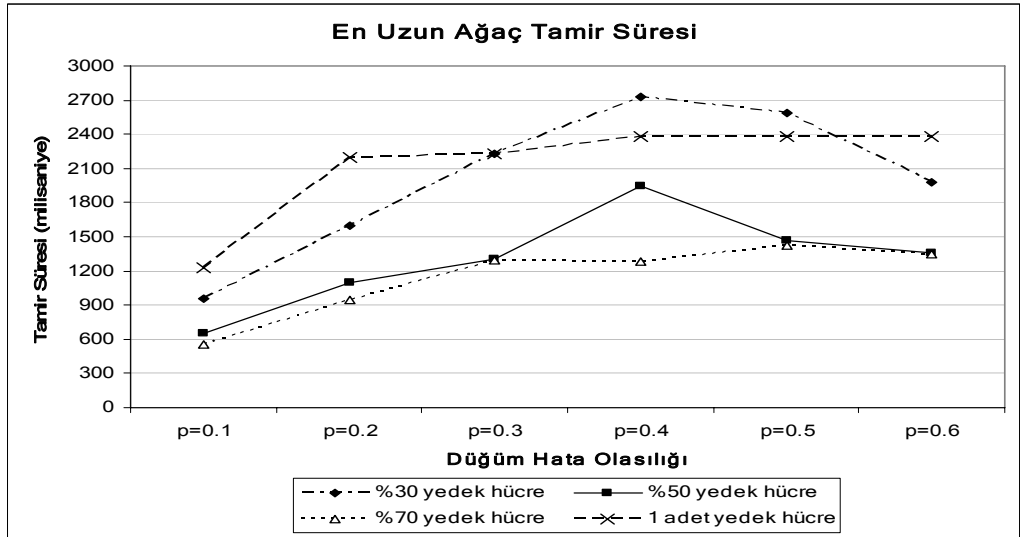
Şekil 8.21. İkili düğümlerin gecikme histogramı.

8.4.1 En iyi yedek hücre – servis hücresi oranı

İlk deney setinde, en iyi yedek hücre - servis hücresi oranını belirleyebilmek amacıyla sistemdeki düğümlerin farklı çocuk sayılarına sahip olduğu deneyler yapılmıştır. Şekil 8.22’de 4 farklı oranda yedek hücre ayırma yaklaşımı, ortalama ağaç tamir süresi ve en uzun ağaç tamir süresi açısından karşılaştırılmıştır. Düğümlerin hata oranı 0,1 ile 0,6 arasında değişmektedir. Ağaç tamir süresi ile ilgili elde edilen sonuçlarda kapasitenin %50’sini ve %70’ini yedek hücre ayırma yaklaşımlarının benzer kazanım sağladığı görülmektedir. Bu durum, bantgenişliği kullanımı açısından değerlendirildiğinde önerme 7.1’i desteklemektedir.



(a)



(b)

Şekil 8.22. Farklı yedek hücre ayırma yaklaşımlarının tamir süreleri a) Ortalama ağaç tamir süresi, b) En uzun ağaç tamir süresi.

Deneylerde kaynaktan hedefe toplam gecikme, akışlandırma yolu üzerindeki toplam bağlantı süresi ve bu yol üzerindeki düğümlerin paket işlem sürelerinin toplamı olarak hesaplanmıştır. Çizelge 8.3'te farklı yedek hücre seçim yaklaşımlarının sonuçları düğümlerin işlem süreci 500 milisaniye olarak alınarak hesaplanmıştır. Çizelgede değerleri verilen bekleme zamanı, en uzun ağaç tamir süresi ile en uzun toplam gecikmenin toplamına eşittir. Çizelgede verilen değerler için %50 yedek hücre ayırma yaklaşımlarının en iyi sonuçları verdiği görülmektedir. %30 yedek hücre ayırma yaklaşımında başlangıç bekleme zamanı %50 yedek hücre ayırma yaklaşımına göre 1 saniye kadar daha uzundur. %70

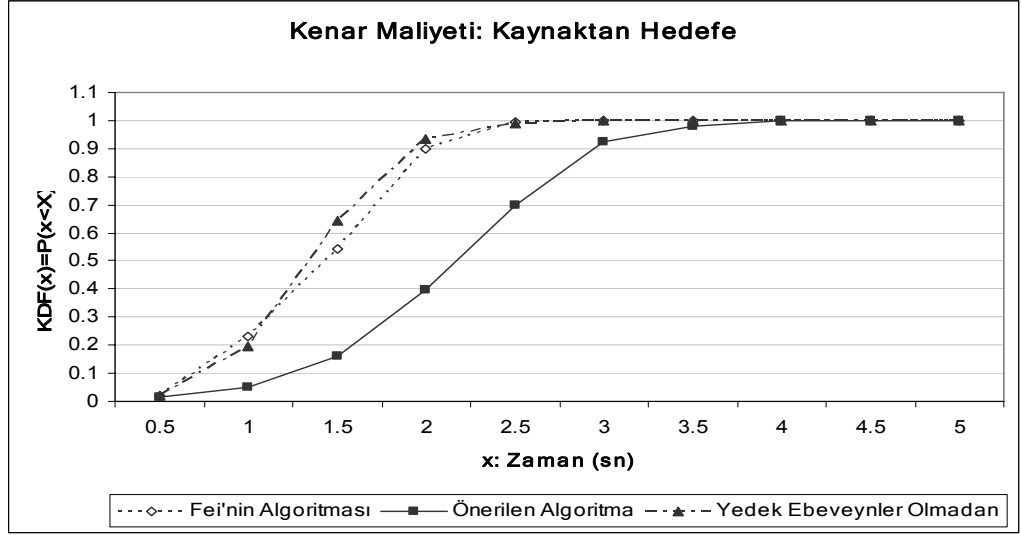
yedek hücre ayırmada ise uzun ağaçlar oluşmaktadır, paket kayıp olasılığından korunma açısından düşünüldüğünde, daha kısa ağaçlar daha avantajlıdır. Ayrıca %50 yedek hücre ayırma yaklaşımı, %70 yedek hücre ayırmaya göre daha iyi bantgenişliği kullanımı sağlamaktadır. Sabit 1 adet yedek hücre ayırmak %50 yedek hücre ayırma yaklaşımından 1 zıplama daha kısa ağaçlar oluşmasını sağlamaktadır; ancak tampon ihtiyacından dolayı 1 adet yedek hücre ayırma yaklaşımında bekleme zamanı artmaktadır. Tüm parametreler incelendiğinde, en iyi seçimin düğümlerin kapasitelerinden %50 yedek hücre ayırmak olduğu görülmektedir.

Çizelge 8.3. Farklı yedek hücre seçim yaklaşımlarına göre ağaç kazanımı.

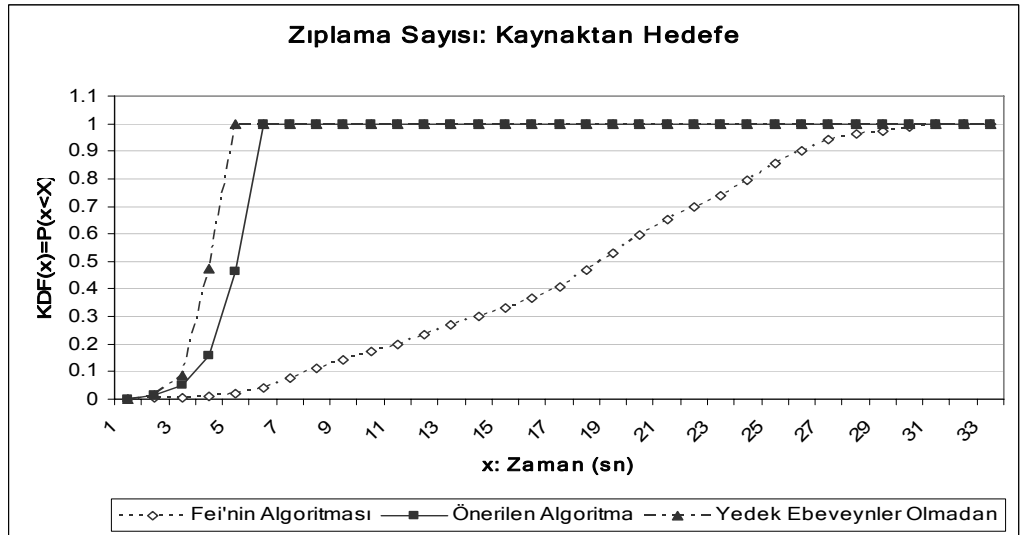
	Ortalama Zıplama Sayısı	Ortalama Toplam Gecikme (sn)	En Fazla Zıplama	En Fazla Toplam Gecikme (sn)	Ortalama Bekleme Zamanı (msn)
%30 yedek hücre	3.74	3.23	6	5.92	5206
%50 yedek hücre	3.82	3.12	5	6.285	4480
%70 yedek hücre	5.96	5.28	9	9.145	6626
1 adet yedek hücre	3.55	3.04	4	4.73	5420

8.4.2 Karşılaştırmalı deney sonuçları

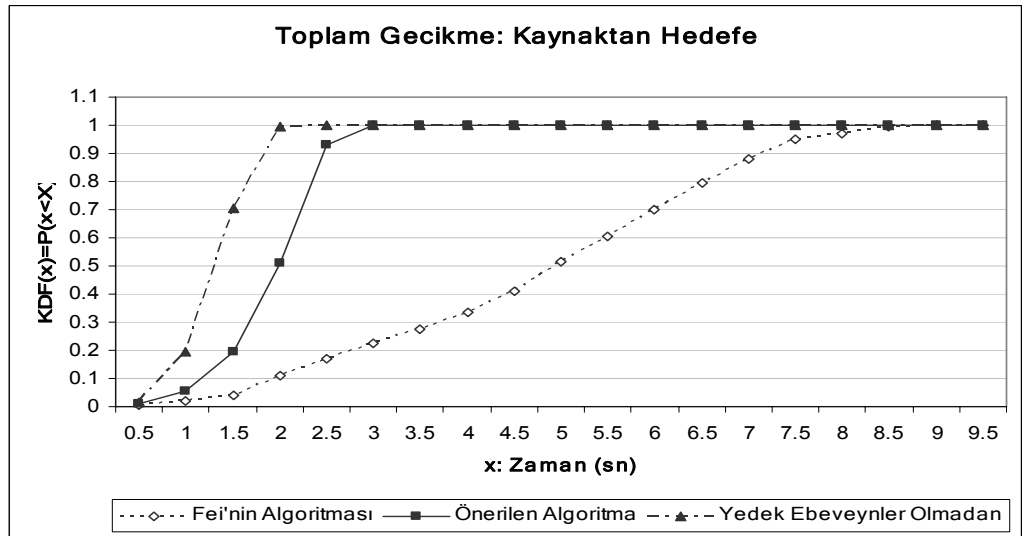
Önerilen hata dayanıklı ağaç yapısının dinamik yapılandırılmasını karşılaştırmalı olarak ölçümlemek amacıyla (Fei and Yang, 2007) çalışması ile aynı test ortamında deneyler yapılmıştır. Karşılaştırmalara ek olarak hiçbir yedek düğümün seçilmediği kapasite farkındalıklı ağaç oluşturma algoritması eklenmiş, böylece yedek hücre ayırmanın ağaçlara olan ekyükü gözlemlenmiştir. İlk olarak ağaç kenar maliyetini ölçmek için düğümlerin işlem süresi 0 olarak alınmıştır. İlgili grafik şekil 8.23'te verilmiştir. Kenar maliyetleri dikkate alındığında (Fei and Yang, 2007) çalışması en iyi sonucu vermiştir. Şekil 8.24'te verilen grafikte her 3 algoritmanın karşılaştırmalı olarak kaynaktan hedefe zıplama sayısı değerleri verilmiştir. Bu grafikte en uzun ağaçların (Fei and Yang, 2007) çalışması tarafından oluşturulduğu görülmektedir. Aynı zamanda tez kapsamında önerilen algoritmanın yedek ebeveyn kullanılmadan oluşturulan ağaçlara göre sınırlı bir ekyükü olduğu gözlemlenmiştir. Zıplama sayısının bekleme zamanına etkisi, şekil 8.25'te verilen grafik üzerinden görülebilir. Şekilde, düğümlerin işlem süresi 500 milisaniye olarak alınmış ve uzun ağaçlara sahip (Fei and Yang, 2007) çalışmasının toplam gecikme değerlerinin diğer algoritmalara göre yüksek olduğu gözlemlenmiştir.



Şekil 8.23. Kaynaktan hedefe bağlantı maliyetleri.



Şekil 8.24. Kaynaktan hedefe zıplama sayısı.



Şekil 8.25. Kaynaktan hedefe toplam gecikme.

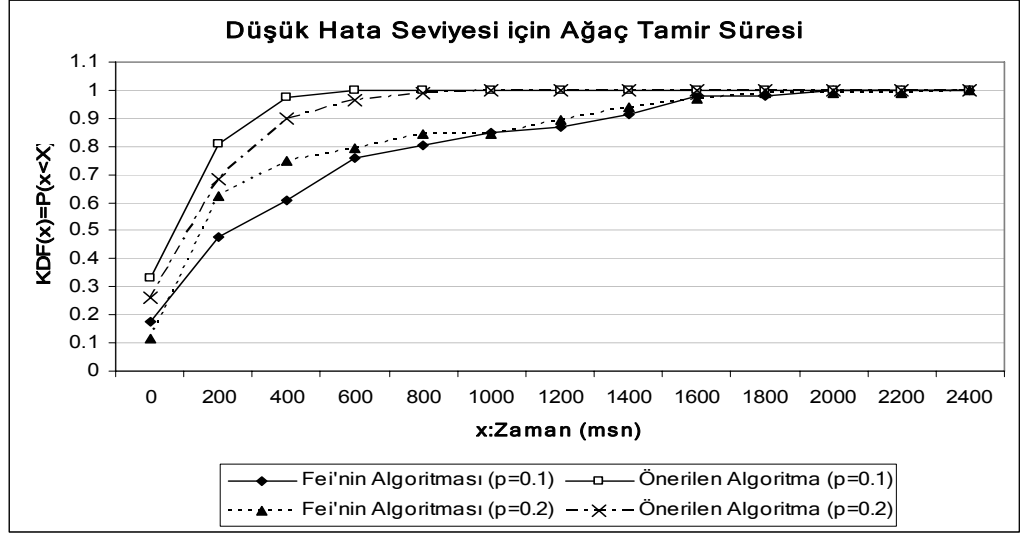
Şekil 8.23 ve 8.25 incelendiğinde, düğümlerin farklı işlem süresine sahip olması durumunda algoritmaların farklı kazanım sonuçlarına sahip olacağı görülmektedir. Tez kapsamında önerilen algoritmanın hangi düğüm işlem sürelerinde (Fei and Yang, 2007) çalışmasından daha iyi ağaçlar oluşturduğunu gözlemlemek amacıyla farklı işlem süreleri için testler yapılmış ve elde edilen veriler çizelge 8.4'te özetlenmiştir. Çizelgede, düğüm işlem süresinin 60 milisaniye ve daha uzun olduğu durumlarda tez kapsamında önerilen algoritmanın ortalama ve en fazla toplam gecikme bakımından daha iyi sonuçlar verdiği görülmektedir. Yapılan ek testlerde ağ üzerinde bulunan düğüm sayısı 100 ile 500 arasında değiştirilmiş ve aynı sonuca, değişen düğüm sayısı koşulunda da varılmıştır.

Çizelge 8.4. Karşılaştırmalı gecikme değerleri.

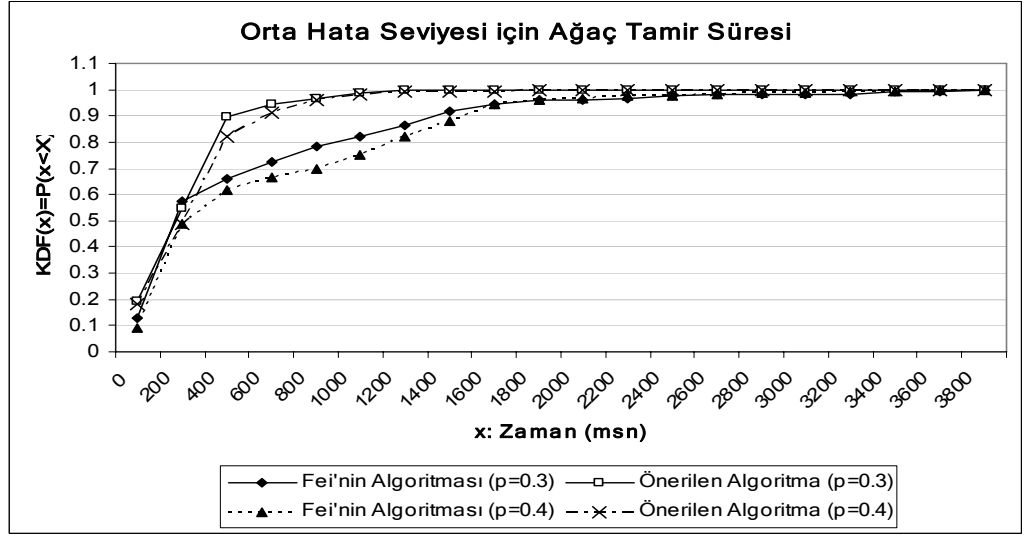
Düğüm işlem süresi (msn)	Önerilen Algoritma		Fei'nin Algoritması	
	Ortalama gecikme (sn)	Maksimum gecikme (sn)	Ortalama gecikme (sn)	Maksimum gecikme (sn)
0	2.14	4.2	1.40	2.59
50	2.31	4.48	2.21	4.06
60	2.34	4.52	2.37	4.32
70	2.38	4.5	2.53	4.66

Şekil 8.26'da düşük seviyede düğüm hata oranı $p=0,1$ ve $p=0,2$ için ağaç tamir süresinin karşılaştırmalı kümülatif dağılım fonksiyonu grafiği verilmiştir. Düşük seviyede hata oranında, önerilen sistem için ölçümlenen ağaç tamir süresi $p=0,1$ için 600 milisaniye, $p=0,2$ için 1000 milisaniye iken, bu değerler (Fei and Yang, 2007) yaklaşımında 1600 milisaniye civarındadır.

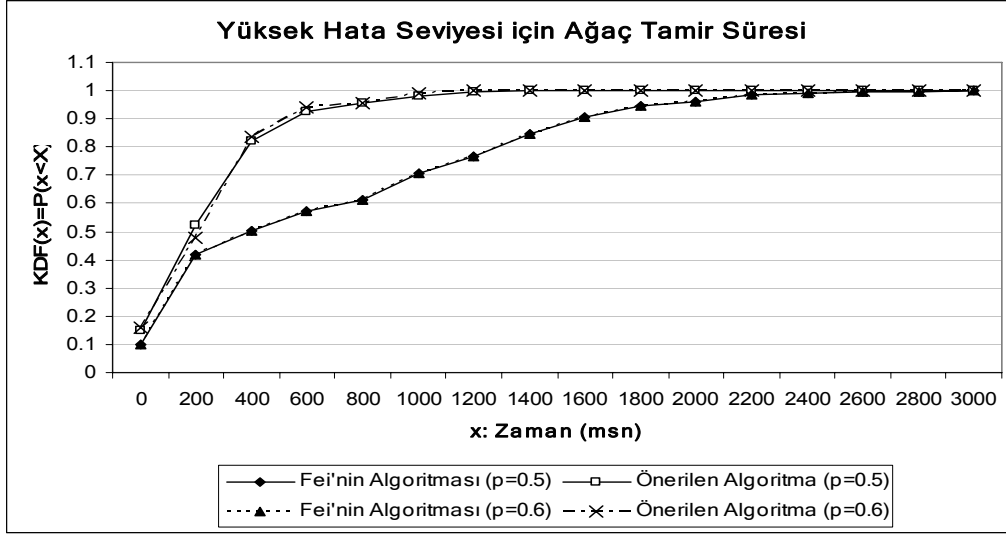
Düğüm hata oranı artarsa, ebeveyn arayan daha çok sayıda düğüm ortaya çıkacağı için ağaç tamir süresi de artacaktır. Orta seviye düğüm hata oranı $p=0,3$ ve $p=0,4$ için yapılan deneylerde her iki algoritmanın karşılaştırmalı deney sonuçları şekil 8.27'de verilmiştir. Önerilen algorithmada elde edilen ağaç tamir süresi %90 olasılıkla 600 milisaniyenin altındadır. Ancak (Fei and Yang, 2007) algoritmasında ağaç tamiri %90 olasılıkla 2000 milisaniyeye kadar uzamaktadır, bu değer, tez kapsamında önerilen algoritmanın yaklaşık 3 katına denk gelmektedir.



Şekil 8.26. Düşük hata seviyesi için ağaç tamir süresi.



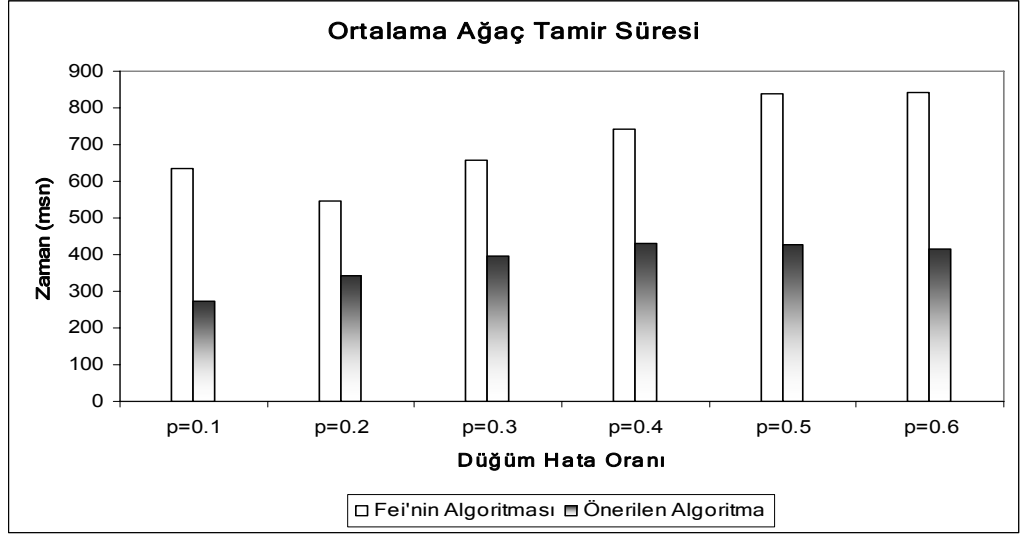
Şekil 8.27. Orta hata seviyesi için ağaç tamir süresi.



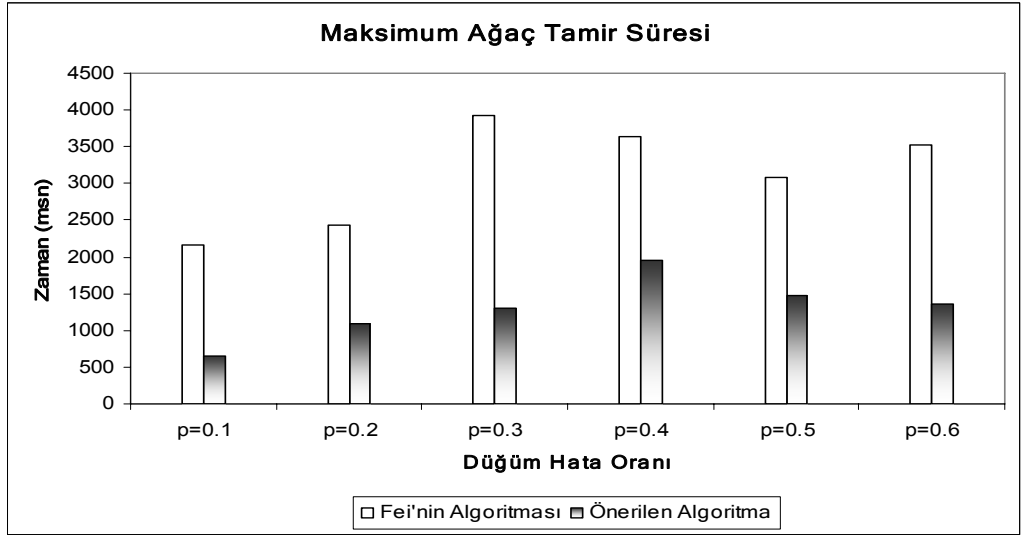
Şekil 8.28. Yüksek hata seviyesi için ağaç tamir süresi.

Şekil 8.28'de $p=0,5$ ve $p=0,6$ olmak üzere yüksek seviyede hata oranı için karşılaştırmalı kümülatif dağılım fonksiyonu grafiği verilmiştir. Önerilen algoritma ile ağaç tamir süresi %90 olasılıkla 1000 milisaniyenin altında iken (Fei and Yang, 2007) algoritması ile aynı olasılıkla bu değer 2000 milisaniye civarındadır.

Son olarak, şekil 8.29.a ve 8.29.b'de sırasıyla ortalama ve maksimum ağaç tamir süresi verilmiştir. Verilen grafiklerde, önerilen algoritmada ağaç tamir süresinin düğüm hata oranı $p=0,3$ 'ü geçtikten sonra sabit bir düzeyde kaldığı görülmektedir. Video akışlandırma uygulamalarında, başlangıç tamponlama zamanı, düğümlerin ebeveyn bağlantısı kopması durumunda yeni bir ebeveyn bulana kadar akışlandırmaya devam edebilmesi açısından önemlidir. Bu bağlamda, maksimum ağaç tamir süresi düğümlere kesintisiz video akışlandırma oturumu sağlayacak tampon süresi hakkında fikir vermektedir; kesintisiz video akışlandırma, tampon süresinin maksimum ağaç tamir süresinden uzun olması ile sağlanabilir. Şekil 8.29.b'de verilen grafik yardımı ile kesintisiz akışlandırma için gerekli tamponlama süresinin önerilen algoritma için 2 saniye, (Fei and Yang, 2007) algoritması için olması gerektiği görülmektedir.



(a)



(b)

Şekil 8.29. Ağaç tamir süresi a) Ortalama ağaç tamir süresi, b) Maksimum ağaç tamir süresi.

Yapılan geniş kapsamlı testlerde önerilen algoritma, farklı ağ koşullarında literatürde yer alan benzer bir çalışmaya göre başarılı sonuçlar vermiştir. Ayrıca, önerilen algoritma ile düğüm hata oranı artsa bile düğümlerin bu durumdan etkilenmeyerek belli sınırlar içinde ağaç tamirini tamamladığı gözlemlenmiştir.

9. SONUÇLAR

Bu tezde, görevdeş ağlar üzerinde ölçeklenebilir video kodlayıcı/çözümleyici kullanarak hata dayanıklı bir video akışlandırma sistemi geliştirilmiştir. Geliştirilen mimari, hiyerarşik katmanlı ve gecikmeye dayalı gruptandırılmaya dayalı olduğu için altta yatan fiziksel mimariden yararlanarak ölçeklendirilebilir bir altyapı oluşturmaktadır. Sistem tasarımında önerilen tüm algoritmalar görevdeş ağların doğasına uygun şekilde dağıtık olarak çalıştırılmaktadır.

Tasarlanan sistemde hata dayanıklılığı hem altta yatan kaplama mimarisinde hem de kurulan çokluguönderim ağaçlarında sağlanmıştır. Gruplarda liderlik görevini üstlenen düğümler sistemden ayrılırsa aynı bilgiyi elinde bulunduran yedek liderler liderlik görevini üstlenmektedir. Diğer yandan, akışlandırma sırasında ebeveynini kaybeden düğümler, yedek ebeveynlere bağlanarak kesintisiz video akışlandırma oturumunu sürdürmektedir.

Önerilen çokluguönderim ağaç algoritması, hem düğümlerin yükleme kapasitelerini hem de birbirlerine olan uzaklıklarını göz önünde bulunduran ve gecikme tabanlı gruplar üzerinde oluşturulan, dolayısıyla literatürde benzeri bulunmayan bir yaklaşım ile ağaç oluşumunu sağlamaktadır. Önerilen hata dayanıklılığı yaklaşımı, hem reaktif hem proaktif bileşenleri olan bir yaklaşım ile farklı hata oranlarında literatürdeki benzerlerinden daha kısa sürede ağaç yapılandırma ve daha kısa ağaçların oluşturulmasını sağlamaktadır. Düğümlerde oluşabilecek yüksek hata oranlarına rağmen sistemin video akışlandırmayı kesintisiz olarak sürdürebildiği, karşılaştırmalı deneyler ile gösterilmiştir. Yapılan analizler, ağaç yapısı kullanan akışlandırma sistemleri için kurulan sistemin en çok, ağacın birinci katmanındaki hatalara karşı dayanıksız olduğunu ortaya koymaktadır. Kök düğümde yedek hücre oranını belirleyen algoritma, ağacın birinci katmanındaki hata dayanıklılığını arttırmaktadır. Bu sayede, oluşan ağaçlarda hata riski azaltılmaktadır.

Kapasite farkındalıklı çokluguönderim ağaç oluşturma yaklaşımı, kapasitesi yüksek olan düğümleri ağacın kök kısımlarına yaklaştırarak yüksek kapasiteli düğümlere yüksek kalitede, bantgenişliğinin yeterli olmadığı durumlarda ise düşük kalitede video verisini düşük kapasitedeki düğümlere göndererek sistem üzerinde bulunan istemcilere adil bir yaklaşım sunmaktadır. Ölçeklenebilir video kodlayıcı/çözümleyici kalite katmanları çıkarma işlemini yalnızca düğümün yükleme hızına göre değil aynı zamanda ağın değişken durumuna göre

yapmaktadır. Yedek ebeveynlerin ayırdığı yedek hücreler henüz boş iken ağ üzerindeki sıkışıklıkta var olan çocuklara yüksek kalitede video akışlandırma yapılmasını sağlamaktadır, ağ üzerinde sıkışıklık olmadığı durumlarda ise var olan tüm çocuklar en üst kalitede video alabilmektedir. Yedek ebeveynlerin sadece ebeveynin sistemden çıkması durumunda değil, aynı zamanda ebeveyn – çocuk arasında bağlantı kötüleştiğinde de kullanılması istemcilerin akışlandırma boyunca ağ koşullarına uygun şekilde olabilecek en yüksek kalitede video almasını sağlamıştır. Literatürde önerilen benzer çalışmalar ile yapılan karşılaştırmalarda elde edilen bithızının daha iyi değerlerde olması, önerilen kalite uyarlama ve ağa düğüm yerleştirme yaklaşımlarından ileri gelmektedir.

Ticari ve akademik olarak öne sürülen örgü tabanlı video akışlandırma sistemleri, başlangıç bekleme zamanının çok uzun sürmesi (dakikalar bazında) ve iki düğüm arası oynatım zamanının uzun olması (ticari sistem PPLive iki komşu düğüm arası oynatım zamanını 140 saniye olarak raporlamıştır) sebebiyle canlı video akışlandırma uygulamalarında sorun yaşamaktadır. Önerilen ağaç tabanlı sistem, başlangıç bekleme zamanını düşürülmesini sağlaması ve iki komşu düğüm arasında oynatım zamanının milisaniyeler bazında olması sebebiyle canlı video akışlandırma uygulamalarına uygun bir altyapı sunmaktadır.

Çoklugoönderim ağaç oluřturma algoritması incelendiğinde, k bir gruptaki düğüm sayısı ve N sistemdeki tüm düğüm sayısı ise, en kötü durum senaryosunda ağaç oluřturma algoritmasının mesaj karmařıklığı $O(kN)$ ile, algoritma karmařıklığı ise $O(N^2)$ ile sınırlıdır.

Sonuç olarak, tasarlanan sistemde, görevdeř mimariler için geliştirilen algoritma ve protokoller ile bu tip sistemlerde karşılaşılan sorunlara farklı çözümler getirilmiştir.

- Hiyerarřık gruplandırma yapısı sayesinde ölçeklenebilir,
- Ekyükü ağaç yapısı ile örgü tabanlı mimariye sahip video akışlandırma sistemlerinden daha az (özel tampon kullanımı ve akışlandırma sırasında düğüm arama işlemlerini gerektirmez),
- Yedek ebeveyn yöntemleri ile ekyükü ve benzer (duplicate) paketleri olmayan, hem proaktif hem de reaktif bileřenleri olan bir hata dayanıklılığına sahip,

- Her düğümde çalışan ortak yazılım ile tüm çizge bilgisine ihtiyaç duymayan ve tamamen dağıtık

bir sistem tasarlanmıştır.

Tasarlanan sistemin pratikte kullanılması için belli sayıda yüksek sayıda yükleme hızına sahip düğüm gerekmektedir. Bunu sağlamak üzere belli sayıda CDN sunucusu görevdeş ağ üzerinde yerleştirilebilir. Sistemde birer düğüm olarak yerleştirilen CDN sunucuları, yükleme hızları yüksek olduğu için bulundukları gruplarda akışlandırma lideri olarak seçilecektir. Dolayısıyla sistem mimarisi gruplara video taşıyan sunucular ve grupların içerisinde birbirine veri gönderen düşük yükleme hızına sahip düğümler olarak şekillenerek pratikte uygulanabilir.

Görevdeş ağlarda video akışlandırma çalışmalarının başarımı düğümler tarafından alınan bithızı, PSNR değeri, başlangıç bekleme zamanı ve devamlılık indeksine bakılarak değerlendirilebilir. Tez kapsamında bithızı ve başlangıç bekleme zamanı parametleri raporlanmıştır. Tez konusu ile ilgili yapılabilecek gelecek çalışmalarda yapılan deneylerin ölçüm yöntemleri geliştirilerek PSNR ve devamlılık indeksi değerleri elde edilebilir ve önerilen algoritmaların istemciler tarafından elde edilen izlenen videonun kalitesi değerlendirilebilir.

Yapılabilecek gelecek çalışmalarda, ağdaki sıkışıklık düzeyinin sebep olduğu kayıp paket oranının düşürülmesi için sıkışıklık durumunda yedek ebeveyne bağlanması yanında, video paketlerine gönderme yönünde hata düzeltimi (FEC – Forward Error Correction) uygulanabilir. Ölçeklenebilir mimari için farklı katmanlara katmanın önem derecesine göre farklı oranda FEC uygulanabilir. Genişleme katmanları taban katman olmadan çözümlenemediği için, en yüksek oranda sadece taban katmana uygulanan FEC, beraberinde getirdiği ekyükün sınırlı kalmasını sağlayabilir. Bu yöntemin paket kayıplarını düşürerek daha kaliteli video izlenmesini sağlayacağı öngörülmektedir.

Tez kapsamında önerilen hiyerarşik gruplar ve ağaç tabanlı akışlandırma yolunun kurulması, sisteme yeni bir düğümün girişi için ölçeklenebilir bir altyapı oluşturmaktadır. Gelecekte yapılacak başka bir çalışma, akışlandırma başladıktan sonra sisteme giren bir düğümün videoyu o an itibariyle izlemeye başlayabilmesidir. Sonradan başvuran düğüm ilk olarak hiyerarşik katmanlar arasında ilerleyerek bir gruba dahil olabilir, daha sonra grup liderine başvuruda

bulunarak sistemde o anda akışlandırma yapılan videoyu izleme talebinde bulunabilir. Grup liderleri, akışlandırma liderlerinin adres bilgisine sahip oldukları için, kendilerine başvuruda bulunan düğümü ağacın o grup içerisinde bulunan parçasına dahil olmak üzere yönlendirebilir. Bu yaklaşım ile, ağaca dahil olma aynı grup içerisindeki, dolayısıyla birbirine yakın düğümler arasında mesajlaşma ile gerçekleşeceği için kısa süreli bir başlangıç bekleme zamanı elde edilmesi öngörülmektedir.

KAYNAKLAR DİZİNİ

- Baccichet, P., Schierl, T., Wiegand, T. and Girod, B.,** 2007, Low-delay Peer-to-Peer Streaming using Scalable Video Coding, IEEE Packet Video Workshop, İsviçre, 173-181pp.
- Banerjee, S., Bhattacharjee, B. and Kommareddy, C.,** 2002, Scalable Application Layer Multicast, ACM SIGCOMM, ABD, 205-217pp.
- Banerjee, S., Lee, S., Bhattacharjee, B. and Srinivasan, A.,** 2006, Resilient multicast Using Overlays, *IEEE/ACM Transactions on Networking*, 14(2):237-248pp.
- Birrer, S. and Bustamante, F.E.,** 2005, Resilient Peer-to-peer Multicast Without the Cost, MMCN, ABD.
- Birrer, S. and Bustamante, F.E.,** 2007, A Comparison of Resilient Overlay Multicast Approaches, *IEEE Journal on Selected Areas in Communications*, 25(9):1695-1705pp.
- Bulut, H., Yardımcı, A., Demirci, S., Kaymak, Y., Sayıt, M.F. and Tunali, E.T.,** 2010, An Efficient JSD-Based Search on Interest-Based Hierarchical Clustering of Overlay Networks, The Second International Conference on Advances in P2P Systems (AP2PS), İtalya.
- Castro, M., Druschel, P., Kermarrec, A-M., Nandi, A., Rowstron A. And Singh, A.,** 2003, SplitStream: High-bandwidth Multicast in a Cooperative Environment, SOSP, ABD.
- Chakareski, J. and Frossard, P.,** 2009, Delay-based Overlay Construction in P2P Video Broadcast, IEEE International Conference on Acoustics, Speech and Signal Processing, Tayvan, 1985-1988pp.
- Chang, E.J-H.,** 1982, Echo Algorithms: Depth Parallel Operations on General Graphs, *IEEE Transactions on Software Engineering*, 8(4):391-401pp.
- Chen, Z., Li, B., Keung, G., Yin, H., Lin, C. and Wang, Y.,** 2009, How Scalable Could P2P Live Media Streaming System Be with the Stringent Time Constraint?, IEEE International Conference on Communications, Almanya, 1-5pp.

KAYNAKLAR DİZİNİ (devam)

- Chu, Y., Rao, S.G., Seshan, S. and Zhang, H.,** 2002, A Case for End System Multicast, *IEEE Journal on Selected Areas in Communications*, 20(8):1456-1471pp.
- Ding, Y., Liu, J., Wang, D. and Jiang, H.,** 2010, Peer-to-Peer Video-on-Demand with Scalable Video Coding, *Computer Communications*, (in press).
- Fei, Z. and Yang, M.,** 2007, A proactive tree recovery mechanism for resilient overlay multicast, *IEEE/ACM Transactions on Networking*, 15(1):173-186pp.
- Gallager, R.G., Humblet, P.A. and Spira, P.M.,** 1983, A Distributed Algorithm for Minimum-weight Spanning Trees, *ACM Transactions on Programming Languages and Systems*, 5(1):66-77pp.
- Ghosh, S.,** 2006, Distributed Systems : An Algorithmic Approach, Chapman & Hall/CRC, 424p.
- Goyal, V.K.,** 2001, Multiple Description Coding: Compression Meets the Network, *IEEE Signal Processing Magazine*, 18(5):74-94pp.
- GT-ITM,** <http://www.cc.gatech.edu/projects/gtitm/>, (Erişim Tarihi:Eylül, 2006).
- Hei, X., Liang, C., Liang, J., Liu, Y. and Ross, K.W.,** 2007, A Measurement Study of a Large-scale P2P IPTV System, *IEEE Transactions on Multimedia*, 9(8):1-15pp.
- Hei, X, Liu, Y. and Ross, K.W.,** 2008, IPTV Over P2P Streaming Networks: The Mesh-Pull Approach, *IEEE Communications Magazine*, 46(2):86-92pp.
- Jeon, H., Son S.C. and Nam, J.S.,** 2008, Overlay multicast tree recovery scheme using a proactive approach, *Computer Communications*, 31:3163-3168pp.
- Jin, X., Yiu, W.P., Chan, S.H. and Wang, Y.,** 2007, On Maximizing Tree Bandwidth for Topology-Aware Peer-to-peer Streaming, *IEEE Transactions on Multimedia*, 9(8):1580-1592pp.
- Kleinberg, J. and Tardos, E.,** 2006, Algorithm Design, Addison Wesley, 864p.

KAYNAKLAR DİZİNİ (devam)

- Kruskal, J.B.**, 1956, On the Shortest Spanning Tree of a Graph and the Traveling Salesman Problem, Proceedings of the American Mathematical Society, 7:48-50pp.
- Kusumoto, T., Kunichika, Y., Katto, J. and Okubo, S.**, 2005, Tree-based Application Layer Multicast Using Proactive Route Maintenance And Its Implementation, ACM workshop on Advances in Peer-To-Peer Multimedia Streaming, Singapur, 49-58pp.
- Liu, J., Rao, S.G., Li, B. and Zhang, H.**, 2008, Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast, IEEE Special Issue on Recent Advances in Distributed Multimedia Communications, 96(1):11-24pp.
- Magharei, N. and Rejaie, R.**, 2006, Understanding Mesh-Based Peer-to-Peer Streaming, International Workshop on Network and Operating System Support for Digital Audio and Video, ABD.
- Magharei, N. and Rejaie, R.**, 2009, PRIME: Peer to Peer Resilient Mesh Based Streaming, *IEEE/ACM Transactions on Networking*, 17(4):1052-1065pp.
- Magharei, N. Rejaie, R. and Guo, Y.**, 2007, Mesh or Multiple-Tree: A Comparative Study of Live P2P Streaming Approaches, IEEE INFOCOM, ABD, 1424-1432pp.
- Magnetto, A., Gaeta, R., Grangetto, M. and Sereno, M.**, 2010, TURINStream: A Totally Push, Robust and Efficient P2P Video Streaming Architecture, *IEEE Transactions on Multimedia*, 12(8):901-914pp.
- Narula, S.C. and Ho, C.A.**, 1980, Degree Constrained Minimum Spanning Tree, *Computer and Operations Research*, 7:239-249pp.
- NextShare**, “P2P Next”, <http://www.livinglab.eu/index.html> (Erişim tarihi: Mayıs, 2011).
- NS2**, <http://www.isi.edu/nsnam/ns/>, (Erişim Tarihi:Eylül, 2006).
- Okada, Y., Oguro, M., Katto, J. and Okubo, S.**, 2005, A New Approach for the Construction of ALM Trees Using Layered Video Coding, ACM workshop on Advances in Peer-To-Peer Multimedia Streaming, Singapur, 59-68pp.

KAYNAKLAR DİZİNİ (devam)

- Padmanabhan, V.N., Wang, H.J. and Chou, P.A.**, 2003, Resilient Peer to Peer Streaming, IEEE International Conference on Network Protocols, ABD, 16-27pp.
- Pouwelse, J.A., Garbacki, P., Wang, J., Bakker, A., Yang, J., Losup, A., Epema, D.H.J., Reinders, M., Van Steen, M.R. and Sips, H.J.**, 2007, Tribler: A Social-based Peer-to-peer System, *Concurrency and Computation: Practice and Experience*, 20(2):127 – 138pp.
- PPLive**, <http://www.pptv.com/>, (Erişim tarihi: Haziran, 2010).
- PPStream**, <http://www.ppstream.com/>, (Erişim tarihi: Haziran, 2010).
- Prim, R.C.**, 1957, Shortest Connection Networks and Some Generalizations, *Bell Systems Technology Journal*, 36(6):1389-1401pp.
- Rejaie, R. and Ortega, A.**, 2003, PALS: Peer-to-Peer Adaptive Layered Streaming, NOSSDAV, ABD, 153–161pp.
- Sayit, M.F., Tunalı, E. . and Tekalp, A.M.**, 2009, Bandwidth-Aware Multiple Multicast Tree Formation Using Hierarchical Clusters In P2P Streaming, Proceedings of the 16th IEEE international conference on Image processing (ICIP), Mısır, 945-948pp.
- Sayit, M.F. ve Tunalı E.T.**, 2010a, Görevdeş Ağlar Üzerinde Video Akışlandırma Sistemleri için Çoklugoönderim Ağaç Oluşturma Algoritması, *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*’ne gönderildi.
- Sayit, M.F., Tunalı, E.T. and Tekalp, A.M.**, 2010b, Resilient Peer to Peer Multicast Tree Streaming with Backup Parents, *Elsevier Computer Communications* dergisine gönderildi.
- Schwarz, H., Marpe, D. and Wiegand, T.**, 2007, Overview of the Scalable Video Coding Extension of the H. 264/AVC Standard, *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1103–1120pp.
- Sentinelli, A., Marfia, G., Gerla, M., Kleinrock, L. And Tewari, S.**, 2007, Will IPTV Ride the Peer-to-Peer Stream?, *IEEE Communications Magazine*, 6:86-92pp.

KAYNAKLAR DİZİNİ (devam)

- Tang, Y., Luo, J.G., Zhang, Q., Zhang, M. and Yang, S.Q.,** 2007, Deploying P2P Networks for Large-scale Live Video Streaming Service, *IEEE Communications Magazine*, 6:100-106pp.
- Tran, D.A., Hua, K.A., and Do, T.T.,** 2004, A Peer-to-Peer Architecture for Media Streaming, *IEEE Journal on Selected Areas in Communications*, 22(1):121-133pp.
- Tunali, E.T. and Sayit, M.F.,** 2011, Optimal Backup Parent Pools for Resilient Multicast Trees on Peer to Peer Networks, *IEEE/ACM Transactions on Networking* dergisine gönderildi.
- Venkataraman, V., Yoshida, K. and Francis, P.,** 2006, Chunkyspread: Heterogeneous Unstructured Tree-Based Peer-to-Peer Multicast, *IEEE International Conference on Network Protocols, ABD*, 2-11pp.
- Vénot, S. and Yan, L.,** 2007, Peer-to-Peer Media Streaming Application Survey, *IEEE Ubiquitous Computing, Systems, Services and Technologies*, Taiti, 139-148pp.
- Vlavianos, A., Ilifotou, M. and Faloutsos, M.,** 2006, Bitos: Enhancing Bittorrent for Supporting Streaming Applications, *IEEE Global Internet Symposium, ABD*.
- Wang, F., Xiong, Y. and Liu, J.,** 2010, mTreeBone: A Collaborative Tree/Mesh Overlay Network Multicast Video Streaming, *IEEE Transactions on Parallel and Distributed Systems*, 21(3):379-392pp.
- Xie, S., Li, B., Keung, G.Y. and Zhang, X.,** 2007, CoolStreaming: Desing, Theory and Practise, *IEEE Transactions on Multimedia*, 9(8):1161-1671pp.
- Yang, M. and Fei, Z.,** 2004, A Proactive Approach to Reconstructing Overlay Multicast Trees, *IEEE INFOCOM, Hong Kong*, 2743-2753pp.
- Zhang M., Luo J-G., Zhao L. and Yang S-Q.,** 2005a, A Peer-to-peer Network for Live Media Streaming – using a Push Pull Approach, *ACM Multimedia, ABD*, 287-290pp.
- Zhang X., Liu J., Zhang Q. and Zhu W.,** 2002, gMeasure: A Group-based Network Performance Measurement Service for Peer-to-peer Applications, *IEEE GLOBECOM, ABD*, 2528-2532pp.

KAYNAKLAR DİZİNİ (devam)

- Zhang, X., Liu, J., Li, B. and Yum, Y.S.P., 2005b,** CoolStreaming/DONet: A Data-driven Overlay Network for Peer-to-Peer Live Media Streaming, IEEE INFOCOM, ABD, 2102-2111pp.
- Zhang, X.Y., Zhang, Q., Zhang, Z., Sond, G. And Zhu, W., 2004,** A Construction of Locality-Aware Overlay Network: mOverlay and Its Performance, *IEEE Journal on Selected Areas in Communications*, 22(1):18-28pp.

ÖZGEÇMİŞ

Müge SAYIT

Adres: Uluslararası Bilgisayar Enstitüsü 35100 Bornova/İZMİR

e-mail: muge.fesci@ege.edu.tr, mugefesci@gmail.com

Eğitim Durumu

Doktora : 2005 - , Ege Üniversitesi, Uluslararası Bilgisayar Enstitüsü
Yüksek : 2001 – 2005, Ege Üniversitesi, Uluslararası Bilgisayar
Lisans Enstitüsü
Lisans : 1995 – 1999, Ege Üniversitesi Matematik Bölümü

İş Deneyimi

Araştırma : 2002 –, Ege Üniversitesi, Uluslararası Bilgisayar Enstitüsü
Görevlisi
Yazılım : 1999 – 2000, Tansaş A.Ş.
Uzmanı

Yabancı Dil

Türkçe : Anadil
İngilizce : İyi derecede

Bilgisayar dilleri ve program deneyimi

- C/C++, Java, Assembly
- Microsoft Visual Studio, Matlab

Projeler

- Görevdeş Ağlarda Ölçeklenebilir Gruplandırma ve Video Akışlandırma – Tübitak 1001 (2008 - ,)

Yayınlar

- Müge Fesci Sayıt, Turhan Tunalı, “*H.264 ile Internet Üzerinden Veri Akışlandırma*”, IEEE SİU, 2006.
- Müge Sayıt, Gamze Seçkin, “*Scalable Video with Raptor for Wireless Multicast Networks*”, IEEE Packet Video Workshop, İsviçre, 2007.

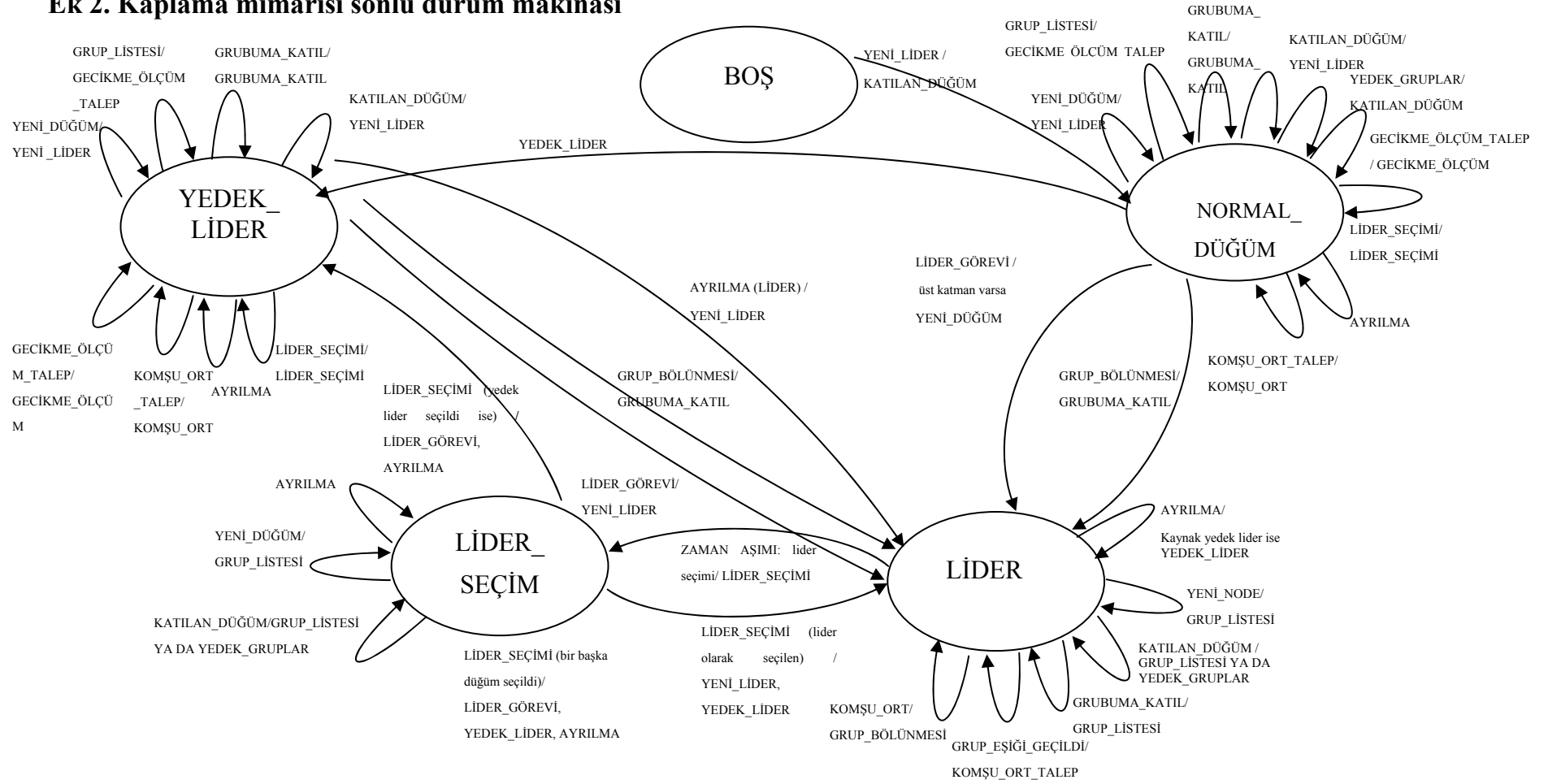
- Muge Fesci-Sayit, E. Turhan Tunalı, A. Murat Tekalp, “*Bandwidth-aware Multiple Multicast Tree Formation Using Hierarchical Clusters in P2P Streaming*”, IEEE ICIP (International Conference on Image Processing), Mısır, 2009.
- Hasan Bulut, Asil Yardımcı, Sercan Demirci, Yağız Kaymak, Müge Fesci Sayıt and E. Turhan Tunalı, "An Efficient JSD-Based Search on Interest-Based Hierarchical Clustering of Overlay Networks", The Second International Conference on Advances in P2P Systems (AP2PS 2010).
- Müge Sayıt, Sercan Demirci, Yağız Kaymak, Hasan Bulut, E. Turhan Tunalı, “*Planetlab Üzerinde Çoklugoönderim Ağaç Tabanlı Video Akışlandırma Uygulaması*”, IEEE SİU, 2011.

EKLER

Ek. 1 Türkçe – İngilizce Terimler Sözlüğü

Akışlandırma	Streaming
Akışlandırma yolu	Streaming path
Çevrimsel Sıralı	Round Robin
Çizge	Graph
Çokluguönderim	Multicast
Doğruluk	Correctness
Düğüm	Node
Ebeveyn	Parent
Fırsatçı	Greedy
Gecikme	Delay
Grup	Cluster
Görevdeş	Peer to peer
Hassasiyet	Accuracy
Hücre	Slot
Kapasite-farkındalıklı	Capacity-aware
Kaplama	Overlay
Kaynak	Source
Kenar	Edge
Kök	Root
Ölçeklenebilir	Scalable
Ölçeklenebilirlik	Scalability
Önyükleme	Bootstrapping
Örgü	Mesh
Sözdekod	Pseudocode
Tümegönderim	Broadcast
Yedek	Backup
Yığıt boyutu	Stack size
Yükleme hızı	Upload
Zıplama	Hop

Ek 2. Kaplama mimarisi sonlu durum makinası



Ek 3. Kaplama Mimarisinin Oluşumu ile İlgili Algoritmalar

Sisteme_giriş():

//Bu rutin, sisteme ilk kez giriş yapan düğüm tarafından çalıştırılır

mesaj_gönder(RN, GİRİŞ); //RN: Randevu Noktası

Lider₁ = mesaj_al(RN); //en üst hiyerarşik katmanın lider adresi

if (sistemdeki_ilk_düğüm)

 yeni_hiyerarşik_katman_olustur(Lider₁=ben, Grup=C₁);

else {

 for tüm H_i katmanları için, i=n:en üst katman to 0

 {

 mesaj_gönder(Lider_i);

 C_i_Düğüm_Listesi = mesaj_al(Lider_i);

 RTT_ölç(C_i_Düğüm_Listesi);

 Lider_i = min_RTT(C_i_Düğüm_Listesi);

 }

 gruba_katıl(C_i); //en alt katmandaki en yakın grup

 C_i_Düğüm_Listesi = mesaj_al(Lider_i);

 RTT_ölç(C_i_Düğüm_Listesi); //ölçüm değerleri tabl. oluşturulur

}

/******

Gruba_yeni_düğüm_ekle():

//Bu rutin, grup lideri tarafından, gruba yeni bir düğüm girdiğinde çalıştırılır

k: benim_grup_id;

mesaj_gönder(düğüm, C_k_Düğüm_Listesi);

//grup bölünmesi

if (grup_üyeleri_eşik_değerini_geçti){

 tümegönderim_mesajı(En_Uzak_Düğüm_Listesi); //zıplama sayısına göre en uzak düğümler

⁽¹⁾düğüm_i_için_ortalama_RTT=mesaj_al(En_Uzak_Düğüm_Listesi);

 en_uzak=en_uzak_düğüm_min_RTT();

 mesaj_at(en_uzak, GRUP_BÖLÜNMEİ);

 for tüm düğüm_i ∈ C_k_Düğüm_Listesi

 mesaj_gönder(düğüm_i, GRUBUMA_KATIL);

```

Ck_Düğüm_Listesi = Ø; //grup listesi sıfırlanır

yeni_düğüm = kaynak(mesaj_al(KABUL));
Ck_Düğüm_Listesi = Ck_Düğüm_Listesi ∪ yeni_düğüm;
mesaj_gönder(yeni_düğüm, Ck_Düğüm_Listesi);
}

if (üst_katman_yok){
    yeni_hiyerarşik_katman_olustur(Lidern=ben, Grup=Cn);
    Cn_Düğüm_Listesi = Cn_Düğüm_Listesi ∪ en_uzak;
}

```

⁽¹⁾ Bir düğüm, average_RTT değerini hesaplarken kendisine en yakın (zıplama sayısı olarak) belirli sayıda düğüme olan RTT değerlerinin ortalamasını alır. Burada amaç, yeni oluşacak grubun lideri olarak, gecikme cinsinden gruptaki diğer düğümlerden çok uzakta olan bir düğümün seçilmesini önlemektir.

Yeni_grup_olustur() :

//Bu rutin, GRUP_BÖLÜNMESİ mesajı alan düğüm tarafından çalıştırılır

k: benim_grup_id;

C_k_Düğüm_Listesi = Ø; //grup listesi sıfırlanır

for tüm düğüm_i ∈ C_k_Düğüm_Listesi

 mesaj_gönder(düğüm_i, GRUBUMA_KATIL);

yeni_düğüm = kaynak(mesaj_al(KABUL));

C_k_Düğüm_Listesi = C_k_Düğüm_Listesi ∪ yeni_düğüm;

mesaj_gönder(yeni_düğüm, C_k_Düğüm_Listesi);

if (üst_katman_var)

 gruba_katıl(C_{üst_katman});

 C_{üst_katman}_Düğüm_Listesi = mesaj_al(Lider_{üst_katman});

 RTT_ölç(C_{üst_katman}_Düğüm_Listesi);

/*****/

Yeni_gruba_katıl() :

//Bu rutin, GRUBUMA_KATIL mesajı alan düğüm tarafından çalıştırılır

```

k: benim_grup_id;

if (ilk_davet){
    mesaj_gönder(kaynak(mesaj_al(GRUBUMA_KATIL)), KABUL);
    Ck_Düğüm_Listesi = mesaj_al(Liderk);
    RTT_ölç(Ck_Düğüm_Listesi);
}

/*****/

Lider_seçimi() :
//Bu rutin grup lideri tarafından belirli aralıklarla ya da yedek
lider tarafından grup lideri sistemden çıktığında çalıştırılır

(2) max_RTT_değerimi_hesapla();
k: benim_grup_id;

for tüm düğümi ∈ Ck_Düğüm_Listesi
    mesaj_gönder(düğümi, LİDER_SEÇİMİ);
yeni_lider = min_max_RTT_ye_sahip_düğüm;
if (yeni_lider!=ben)
    mesaj_gönder(Liderüst_katman, AYRILMA); //üst katmandan çıkıyor
/*****/

Lider_seçimi_mesajı_alınması() :
//Bu rutin, LİDER_SEÇİMİ mesajı alan düğüm tarafından çalıştırılır

k: benim_grup_id;
max_RTT_değerimi_hesapla();
mesaj_gönder(Liderk, benim_max_RTT_değerim);
/*****/

Ayrılma_mesajı_alınması() :
//Bu rutin, AYRILMA mesajı alan lider tarafından çalıştırılır

k: benim_grup_id;
Ck_Düğüm_Listesi = Ck_Düğüm_Listesi - kaynak(mesaj_al(AYRILMA));
if (grup_eleman_sayısı_eşik_değeri_altında)
    komşu_gruba_katıl(); //en az elemana sahip komşu gruba katılır

```

Ek 4. Video Talep ve Arama Algoritmaları

```

Video_talep_mesajı_alınması():
//Bu rutin, VIDEO_TALEP mesajı alan grup lideri tarafından
çalıştırılır

h = hiyerarşik_katman;

if (grubumda video bulunuyor)
    çoklugoönderim_ağacı_oluşturma_algoritması();
else
    Videoyu_bir_üst_katmanda_ara():
/*****/
Videoyu_bir_üst_katmanda_ara ():
//Bu rutin, grup lideri tarafından bir üst hiyerarşik katmanda
video aranması durumunda çalıştırılır

k+1: (h+1) hiyerarşik katmanındaki grup numaram;

for tüm düğümi ∈ Ck+1_Düğüm_Listesi
    mesaj_gönder(düğümi ,VIDEO_ARAMA);

video_arama_zamanlayıcısını_kur();
h_talep = hiyerarşik_katman;
/*****/
Video_arama_mesajı_alınması():
//Bu rutin, VIDEO_ARAMA mesajı alan grup lideri tarafından
çalıştırılır

h = h_talep = hiyerarşik_katman;
istemci_düğüm = kaynak(mesaj_al(VIDEO_ARAMA));

if ((h-1) en alt hiyerarşik katman) {
    if (video grubumda bulunuyor ise)
        mesaj_gönder(istemci_düğüm, VIDEO_ARAMA_CEVAP);
}
else {
    k-1: (h-1) hiyerarşik katmanındaki grup numaram;

    for tüm düğümi ∈ Ck-1_Düğüm_Listesi

```

```

        mesaj_gönder(düğümi ,VIDEO_ARAMA);
    }

/*****/

Video_arama_cevap_mesajı_alınması():
//Bu rutin, VIDEO_ARAMA_CEVAP mesajı alan grup lideri tarafından
çalıştırılır

h = hiyerarşik_katman;

if (h == h_talep) {
    mesaj_gönder(istemci_düğüm, VIDEO_ARAMA_CEVAP);
}
else {
    mesaj_gönder(Liderkatman_katman, VIDEO_ARAMA_CEVAP);
}

/*****/

Video_arama_zaman_aşımı():
//Bu rutin, VIDEO_ARAMA mesajına yanıt alamayan liderin
zamanlayıcısı sonlandığında çalıştırılır

if (h_talep == en_üst_hiyerarşik_katman){
    "Aranan Video Sistemde Bulunmadı"
}
else {
    Videoyu_bir_üst_katmanda_ara():
}

```