İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

WDM AĞLARINDA ÇOKLU AKTARIM İÇİN YENİ BİR YAKLAŞIM VE BAŞARIM ANALİZİ

YÜKSEK LİSANS TEZİ

Müh. M. Derin HARMANCI

Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ

Programı : BİLGİSAYAR MÜHENDİSLİĞİ

OCAK 2003

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

WDM AĞLARINDA ÇOKLU AKTARIM İÇİN YENİ BİR YAKLAŞIM VE BAŞARIM ANALİZİ

YÜKSEK LİSANS TEZİ

Müh. M. Derin HARMANCI
(504001551)

Tezin Enstitüye Verildiği Tarih : 24 Aralık 2002
Tezin Savunulduğu Tarih : 17 Ocak 2003

Tez Danışmanı :	**Prof.Dr. Bülent ÖRENCİK**
Diğer Jüri Üyeleri	**Doç.Dr. Sema OKTUĞ**
	Doç.Dr. Cem ERSOY (B.Ü.)

OCAK 2003

ÖNSÖZ

Tez çalışmam sırasında bilgisini ve tecrübesini benimle paylaşan, çalışmalarımda bana yol gösteren, hocam Prof. Dr. Bülent Örencik'e teşekkürlerimi sunarım.

Ocak 2003 M. Derin Harmancı

İÇİNDEKİLER

KISALTMALAR v
TABLO LİSTESİ vi
ŞEKİL LİSTESİ vii
SEMBOL LİSTESİ xii
ÖZET xiii
SUMMARY xiv

1. GİRİŞ 1
2. OPTİK AĞLAR 4
2.1 İletişim Ağlarında Üç Nesil 5
2.2 WDM ve Diğer Optik Çoklama Yöntemleri 6
2.2.1 OTDM 6
2.2.2 CDM 6
2.2.3 WDM 7
2.3 WDM Ağ Mimarileri 7
2.3.1 Pasif Optik Ağlar (PON) 8
2.3.2 Yayın Seçici Ağlar (Broadcast and Select Networks) 8
2.3.3 Dalgaboyu Atamalı Ağlar (Wavelength Routing Networks) 8
2.3.3.1 Dalgaboyu Yönlendiricilerinin Özellikleri 10
2.4 Dalgaboyu Atamalı Ağlarda Yol Atama 13
2.5 Dalgaboyu Atamalı Ağlarda Yeni Anahtarlama Yöntemleri 14
2.5.1 Optik Patlama Anahtarlaması (Optical Burst Switching - OBS) 14
2.5.2 Optik Etiket Anahtarlaması (Optical Label Switching -OLS) 15
2.6 IP- Optik Ağ Bütünleşmesi 16
3. ÇOKLU AKTARIM TEMELLERİ 19
3.1 Çoklu Gönderim ile Tekli İletişimin Farklılıkları 19
3.2 Çoklu Aktarım Grubu ve Çoğa Aktarım Adresi 20
3.3 Çoklu Aktarım Protokolleri 21
3.3.1 Internet Grup Yönetim Protokolü (Internet Group Management Protocol -IGMP) 22
3.3.2 Bölge-içi Çoklu Aktarım 23
3.3.2.1 Kaynak Tabanlı ÇAA Oluşturma Yöntemleri 25
3.3.2.2 Bölge-içi Çoklu Aktarım Protokolleri 29
3.3.3 Bölgelerarası Çoklu Aktarım 31
4. DALGABOYU ATAMALI WDM AĞLARDA Ç. AKTARIM YÖNTEMLERİ 32
4.1 WDM Çoklu Aktarım Yöntemlerinin Sınıflandırılması 33
4.2 Statik WDM ÇAA Oluşturma Algoritmaları 34
4.2.1 ÜyeYönelimli (Member-Only) 35
4.2.2 Üye Öncelikli (Member-First) 36
4.2.3 Evlat Edinmeli 39
4.3 Statik ÇAA oluşturma algoritmalarında Dalgaboyu Atama 41
4.4 Dinamik WDM ÇAA Oluşturma Algoritmaları 42
4.4.1 WDM Mesajlaşmalı Çoklu Aktarım Protokolü 42
4.4.1.1 MI Başlatmalı (Parent-Initiated LSP Setup) 43
4.4.1.2 Yetim Başlatmalı (Child-Initiated LSP Setup) 45
4.4.1.3 Komşu Destekli (Relative-assisted LSP Setup) 45
4.4.2 IP Mesajlaşmalı Çoklu Aktarım Protokolü 46

5. ÖNERİLEN ALGORİTMA – İKİ KATMANLI KAPSAYAN AĞAÇ YÖNTEMİ 48
5.1 Algoritmanın Kullandığı İki Katmanlı Yapı 48
5.2 Algoritmada Kullanılan Varsayımlar 50
5.3 Algoritmanın Aşamaları 51
5.3.1 Algoritmanın Birinci aşaması 51
5.3.1.1 Alt Ağların Oluşturulması 52
5.3.1.2 Alt Ağlar Arasındaki Bağlantıların Oluşturulması 54
5.3.2 Algoritmanın İkinci aşaması 59
5.3.2.1 Üst Katman ÇAA'sının Elde Edilmesi 59
5.3.2.2 Alt Katman ÇAA'sının Elde Edilmesi 60
5.3.3 Alt Ağ Elemanı Olmayan Düğümlere Erişim 61
6. SONUÇLAR VE TARTIŞMA 63
6.1 Sonuçlar ve Yorumlar 64
6.1.1 Ağdaki Link Kullanımı 65
6.1.2 Dalgaboyu Kullanımı 66
6.1.3 Ortalama ve Maksimum Gecikme 67
6.1.4 Yürütme Süreleri 68
6.2 Sonuç ve Öneriler 70
KAYNAKLAR 71
EK- A 73
ÖZGEÇMİŞ 108

KISALTMALAR

ATM : Asynchronous Transfer Mode
BGMP : Border Gateway Multicast Protocol
CBT : Core-Based Tree
CDM : Code Division Multiplexing
ÇAA : Çoklu aktarım Ağacı
DVMRP : Distance Vector Multicast Routing Protocol
GMPLS : Generalized Multi-protocol Label Switching
IGMP : Internet Group Managenemt Protocol
LSP : Label Switched Path
MI : Multicast Incapable
MC : Multicast Capable
MPLS : Multi-protocol Label Switching
MRT : Multicast Routing Table
MSDP : Multicast Source Discovery Protocol
OBS : Optical Burst Switching
OLS : Optical Label Switching
OSPF : Open Shortest Path First
OTDM : Optical Time Division Multiplexing
PIM : Protocol Independent Multicast
PON : Passive Optical Networks
QoS : Quality of Service
RIP : Routing Information Protocol
RPF : Reverse-Path Forwarding
SONET : Synchronous Optical Network
TTL : Time To Live
WDM : Wavelength Division Multiplexing

TABLO LİSTESİ

Sayfa No

Tablo 4.1 GRAFT ve PRUNE mesajlarının IP mesajlaşmalı Çoklu Aktarım Protokolündeki yorumları 47

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1 İletişim ağlarında üç nesil 5
Şekil 2.2 Örnek bir WDM Dalgaboyu atamalı ağ..... 9
Şekil 2.3 Ip –Optik ağ bütünleşmesine ilişkin mimari 17
Şekil 4.1 MI düğümlerin ÇAA sayısına etkisi 32
Şekil 4.2 Basit bir ÇAA oluşturma Algoritması..... 35
Şekil 4.3 Üye Yönlendirmeli ÇAA oluşturma Algoritması 35
Şekil 4.4 "Üye Öncelikli" algoritmadaki Güncelle_L() ve Yeni_düğüm_ekle() fonksiyonları 37
Şekil 4.5 Üye Öncelikli ÇAA oluşturma algoritması 38
Şekil 4.6 "Evlat edinmeli" ÇAA oluşturma algoritmasının sözde kodu 40
Şekil 4.7 "MI Başlatmalı" ÇAA algoritmasının sözde kodu**.**..... 44
Şekil 5.1 20 düğümlü bir ağ..... 49
Şekil 5.2 Ağ üzerinde gruplandırma yapıldıktan sonra elde edilen alt ağlar.... 49
Şekil 5.3 Alt Ağ Oluşturma algoritmasının sözde kodu..... 55
Şekil 5.4 Alt ağlar arası bağlantıları kurma algoritmasının sözde kodu..... 57
Şekil 5.5 Alt ağ merkezleri arası yolların bulunması..... 59
Şekil 5.6 Verilen bir çoklu aktarım grubu için oluşturulan çoklu aktarım ağacı.. 60
Şekil 6.1 Toplam link kullanımının düğüm sayısına göre değişimi..... 66
Şekil 6.2 Yürütme süresinin düğüm sayısına göre değişimi..... 69
Şekil A.1 Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 20).....73
Şekil A.2 Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 20).....73
Şekil A.3 Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 20).....73
Şekil A.4 Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 30).....74
Şekil A.5 Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 30).....74
Şekil A.6 Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 30).....74
Şekil A.7 Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 40).....75
Şekil A.8 Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 40).....75
Şekil A.9 Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 40).....75
Şekil A.10 Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 50).....76
Şekil A.11 Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 50).....76
Şekil A.12 Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 50).....76
Şekil A.13 Toplam Link Kullanımının düğüm sayısına göre değişimi (MC Oranı = %10 , Üye Düğüm Oranı=%10).....77

Şekil A.14 Toplam Link Kullanımının düğüm sayısına göre değişimi (MC Oranı = %10 , Üye Düğüm Oranı=%50)....77
Şekil A.15 Toplam Link Kullanımının düğüm sayısına göre değişimi (MC Oranı = %10 , Üye Düğüm Oranı=%90)....77
Şekil A.16 Toplam Link Kullanımının düğüm sayısına göre değişimi (MC Oranı = %30 , Üye Düğüm Oranı=%10)....78
Şekil A.17 Toplam Link Kullanımının düğüm sayısına göre değişimi (MC Oranı = %30 , Üye Düğüm Oranı=%50)....78
Şekil A.18 Toplam Link Kullanımının düğüm sayısına göre değişimi (MC Oranı = %30 , Üye Düğüm Oranı=%90)....78
Şekil A.19 Toplam Link Kullanımının düğüm sayısına göre değişimi (MC Oranı = %50 , Üye Düğüm Oranı=%10)....79
Şekil A.20 Toplam Link Kullanımının düğüm sayısına göre değişimi (MC Oranı = %50 , Üye Düğüm Oranı=%50)....79
Şekil A.21 Toplam Link Kullanımının düğüm sayısına göre değişimi (MC Oranı = %50 , Üye Düğüm Oranı=%90)....79
Şekil A.22 Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 20)....80
Şekil A.23 Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 20)....80
Şekil A.24 Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 20).... ..80
Şekil A.25 Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 30)....81
Şekil A.26 Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 30)....81
Şekil A.27 Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 30)....81
Şekil A.28 Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 40)....82
Şekil A.29 Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 40)....82
Şekil A.30 Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 40)....82
Şekil A.31 Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 50)....83
Şekil A.32 Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 50)....83
Şekil A.33 Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 50)....83
Şekil A.34 Dalgaboyu sayısının düğüm sayısına göre değişimi (MC Oranı = %10, Üye Düğüm Oranı=%10)....84
Şekil A.35 Dalgaboyu sayısının düğüm sayısına göre değişimi (MC Oranı = %10, Üye Düğüm Oranı=%50)....84
Şekil A.36 Dalgaboyu sayısının düğüm sayısına göre değişimi (MC Oranı = %10, Üye Düğüm Oranı=%90)....84
Şekil A.37 Dalgaboyu sayısının düğüm sayısına göre değişimi (MC Oranı = %30, Üye Düğüm Oranı=%10)....85
Şekil A.38 Dalgaboyu sayısının düğüm sayısına göre değişimi (MC Oranı = %30, Üye Düğüm Oranı=%50)....85
Şekil A.39 Dalgaboyu sayısının düğüm sayısına göre değişimi (MC Oranı = %30, Üye Düğüm Oranı=%90)....85
Şekil A.40 Dalgaboyu sayısının düğüm sayısına göre değişimi (MC Oranı = %50, Üye Düğüm Oranı=%10)....86

Şekil A.41 Dalgaboyu sayısının düğüm sayısına göre değişimi (MC Oranı = %50, Üye Düğüm Oranı=%50)....86
Şekil A.42 Dalgaboyu sayısının düğüm sayısına göre değişimi (MC Oranı = %50, Üye Düğüm Oranı=%90)....86
Şekil A.43 Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 20)....87
Şekil A.44 Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 20)....87
Şekil A.45 Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 20).... ..87
Şekil A.46 Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 30)....88
Şekil A.47 Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 30)....88
Şekil A.48 Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 30)....88
Şekil A.49 Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 40)....89
Şekil A.50 Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 40)....89
Şekil A.51 Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 40)....89
Şekil A.52 Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 50)....90
Şekil A.53 Ortalama Gecikmenin sayısının Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 50)....90
Şekil A.54 Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 50)....90
Şekil A.55 Ortalama Gecikmenin düğüm sayısına göre değişimi (MC Oranı = %10 , Üye Düğüm Oranı=%10)....91
Şekil A.56 Ortalama Gecikmenin düğüm sayısına göre değişimi (MC Oranı = %10 , Üye Düğüm Oranı=%50)....91
Şekil A.57 Ortalama Gecikmenin düğüm sayısına göre değişimi (MC Oranı = %10 , Üye Düğüm Oranı=%90)....91
Şekil A.58 Ortalama Gecikmenin düğüm sayısına göre değişimi (MC Oranı = %30 , Üye Düğüm Oranı=%10)....92
Şekil A.59 Ortalama Gecikmenin düğüm sayısına göre değişimi (MC Oranı = %30 , Üye Düğüm Oranı=%50)....92
Şekil A.60 Ortalama Gecikmenin düğüm sayısına göre değişimi (MC Oranı = %30 , Üye Düğüm Oranı=%90)....92
Şekil A.61 Ortalama Gecikmenin düğüm sayısına göre değişimi (MC Oranı = %50 , Üye Düğüm Oranı=%10)....93
Şekil A.62 Ortalama Gecikmenin düğüm sayısına göre değişimi (MC Oranı = %50 , Üye Düğüm Oranı=%50)....93
Şekil A.63 Ortalama Gecikmenin düğüm sayısına göre değişimi (MC Oranı = %50 , Üye Düğüm Oranı=%90)....93
Şekil A.64 Maksimum Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı= %10, Düğüm Sayısı = 20)....94
Şekil A.65 Maksimum Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı= %30, Düğüm Sayısı = 20)....94
Şekil A.66 Maksimum Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı= %50, Düğüm Sayısı = 20).... ..94
Şekil A.67 Maksimum Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı= %10, Düğüm Sayısı = 30)....95

Şekil A.68 Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 30).....95
Şekil A.69 Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 30).....95
Şekil A.70 Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 40).....96
Şekil A.71 Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 40).....96
Şekil A.72 Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 40).....96
Şekil A.73 Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 50).....97
Şekil A.74 Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 50).....97
Şekil A.75 Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 50).....97
Şekil A.76 Ortalama Gecikmenin düğüm sayısına göre değişimi (MC Oranı = %10 , Üye Düğüm Oranı=%10).....98
Şekil A.77 Ortalama Gecikmenin düğüm sayısına göre değişimi (MC Oranı = %10 , Üye Düğüm Oranı=%50).....98
Şekil A.78 Ortalama Gecikmenin düğüm sayısına göre değişimi (MC Oranı = %10 , Üye Düğüm Oranı=%90).....98
Şekil A.79 Ortalama Gecikmenin düğüm sayısına göre değişimi (MC Oranı = %30 , Üye Düğüm Oranı=%10).....99
Şekil A.80 Ortalama Gecikmenin düğüm sayısına göre değişimi (MC Oranı = %30 , Üye Düğüm Oranı=%50).....99
Şekil A.81 Ortalama Gecikmenin düğüm sayısına göre değişimi (MC Oranı = %30 , Üye Düğüm Oranı=%90).....99
Şekil A.82 Ortalama Gecikmenin düğüm sayısına göre değişimi (MC Oranı = %50 , Üye Düğüm Oranı=%10).....100
Şekil A.83 Ortalama Gecikmenin düğüm sayısına göre değişimi (MC Oranı = %50 , Üye Düğüm Oranı=%50).....100
Şekil A.84 Ortalama Gecikmenin düğüm sayısına göre değişimi (MC Oranı = %50 , Üye Düğüm Oranı=%90).....100
Şekil A.85 Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı= %10, Düğüm Sayısı = 20).....101
Şekil A.86 Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı= %30, Düğüm Sayısı = 20).....101
Şekil A.87 Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı= %50, Düğüm Sayısı = 20)..... 101
Şekil A.88 Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı= %10, Düğüm Sayısı = 30).....102
Şekil A.89 Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı= %30, Düğüm Sayısı = 30).....102
Şekil A.90 Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı= %50, Düğüm Sayısı = 30).....102
Şekil A.91 Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı= %10, Düğüm Sayısı = 40).....103
Şekil A.92 Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı= %30, Düğüm Sayısı = 40).....103
Şekil A.93 Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı= %50, Düğüm Sayısı = 40).....103
Şekil A.94 Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı= %10, Düğüm Sayısı = 50).....104

Şekil A.95 Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı= %30, Düğüm Sayısı = 50).....104
Şekil A.96 Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı= %50, Düğüm Sayısı = 50).....104
Şekil A.97 Yürütme Süresinin düğüm sayısına göre değişimi (MC Oranı = %10, Üye Düğüm Oranı=%10).....105
Şekil A.98 Yürütme Süresinin düğüm sayısına göre değişimi (MC Oranı = %10 , Üye Düğüm Oranı=%50).....105
Şekil A.99 Yürütme Süresinin düğüm sayısına göre değişimi (MC Oranı = %10 , Üye Düğüm Oranı=%90).....105
Şekil A.100 Yürütme Süresinin düğüm sayısına göre değişimi (MC Oranı = %30, Üye Düğüm Oranı=%10).....106
Şekil A.101 Yürütme Süresinin düğüm sayısına göre değişimi (MC Oranı = %30, Üye Düğüm Oranı=%50).....106
Şekil A.102 Yürütme Süresinin düğüm sayısına göre değişimi (MC Oranı = %30, Üye Düğüm Oranı=%90).....106
Şekil A.103 Yürütme Süresinin düğüm sayısına göre değişimi (MC Oranı = %50, Üye Düğüm Oranı=%10).....107
Şekil A.104 Yürütme Süresinin düğüm sayısına göre değişimi (MC Oranı = %50, Üye Düğüm Oranı=%50).....107
Şekil A.105 Yürütme Süresinin düğüm sayısına göre değişimi (MC Oranı = %50, Üye Düğüm Oranı=%90).....107

SEMBOL LİSTESİ

Ağ(i) : i . alt ağ.
Aday(i , j) : Ağ(i) içinde bir dış düğümün bağlanabileceği j. aday düğüm.
Alt_Ağ_Aday(i) : i. alt ağda dış düğüm ile arasındaki yolun en düşük maliyetli olduğu aday düğüm.
Ağ_Aday : Tüm ağ içinde, incelenen dış düğüm ile arasındaki yolun maliyeti en düşük olan aday düğüm.
Çocuk(*i*) : *i* düğümünün IP katmanında oluşturulan ÇAA 'sındaki çocuk sayısı
D : Tüm varış düğümleri kümesi.
D* : Henüz bir ağaca eklenmemiş varış düğümleri kümesi.
e(v,u) : *v* ile *u* düğümleri arasındaki link
h(i) : P(s,i) yolu üzerindeki sekme sayısı.
L : Öncelik listesi (Önceliklerine göre sıralanmış linkler kümesi.)
$L_ı$: Işık bölme yeteneği olan ağ düğümleri listesi.
L_d : Dış düğümlerin listesi.
M (x,y) : *x* ve *y* düğümleri arasındaki en kısa yolun maliyeti.
MC(*i*) : *i* düğümünün ışık bölme yeteneği (sayısal olarak) .
P(s,i) : *s* kaynak düğümü ile *i* düğümü arasındaki en kısa yol.
T : ÇAA üzerindeki ayrıtlar kümesi.
U_T : ÇAA üzerinde olmayan diğer düğümler
V : Graf üzerindeki tüm düğümlerin kümesi.
V_T : ÇAA üzerinde olan ve hala ışık bölme kapasitesi olan düğümler kümesi.
V'_T : ÇAA üzerinde olan ancak ışık bölme kapasitesi olmayan düğümler kümesi.

ÖZET

Bu çalışmada, gelecekte omurgalarda kullanılması öngörülen optik ağlarda çoklu aktarım yöntemlerini geliştirmek amaçlanmıştır. Çalışma optik ağlar içinden Dalgaboyu Atamalı WDM ağlar (Wavelength Routed WDM Networks) üzerine yoğunlaştırılmıştır.

WDM tekniği, elektronik ortamda bant genişliğinin daha verimli olarak kullanılmasını sağlayan FDM tekniğinin optik ağlara uyarlanmış biçimi olarak düşünülebilir. FDM'de her bir kanal için atanmış olan frekans, WDM'de her bir kanal için atanmış dalgaboyu olarak karşımıza çıkar. Dalgaboyu Atamalı WDM ağlar da, yönlendiricileri optik ortamda çalışan ve her bir yönlendiricisinde girişten gelen belli bir dalgaboyundaki işaretin hangi çıkıştan hangi dalgaboyu kullanılarak gönderileceğine karar verilen ağlardır.

Optik teknolojisi, henüz yeni ve olgunlaşmamış bir teknoloji olduğundan, optik ağların, dolayısıyla da Dalgaboyu Atamalı WDM ağların, bellek ve lojik işlemler yapabilen birimleri yoktur. Bu durumda, ağ üzerindeki yol atama işlemleri için gerekli kararların elektronik bir ortamda gerçeklenmesi gerekmektedir. Bunun yanında, dalgaboyu atamalı WDM ağlarda, iki düğüm arasında yol atama işlemi yapılırken, yalnız iki düğüm arasında en az maliyetli yolun seçiminin yapılması yeterli olmaz. Bunun yanında, bu yol üzerinde, iki düğümün bağlantısı için kullanılabilecek uygun dalgaboyları da belirlenmelidir.

Çoklu aktarım söz konusu olduğunda, yukarıda sözü edilen sorunlar yanında ayrıca optik ortamda paketlerin çoğullanması sorununun da aşılması gerekmektedir. Optik ortamda paketlerin çoğullanması, ışık bölme yöntemi kullanılarak yapılmaktadır ve bu işlem oldukça maliyetli olduğundan, her yönlendiricinin bu işlemi gerçekleştirmesi uygulanması mümkün olmayan bir çözüm olmaktadır. Halbuki, elektronik ağlarda bir düğüme gelen bir paket, düğümdeki bellek boyutuna bağlı olarak, istenildiği kadar çoğaltılabilir. Elektronik ortamdaki çoklu aktarım algoritmaları da bu varsayım üzerine kurulmuşlardır.

Bu çalışmada yapılan iş, yukarıda anlatılan zorluklar çerçevesinde optik ortamda çoklu aktarımın verimli bir şekilde gerçekleştirilmesi ve elde edilen çoklu aktarım yöntemlerinin elektronik ortamdaki çoklu aktarım yöntemleri ile uyumlu olması sağlamaktır. Bunun gerçekleştirilebilmesi için, günümüze kadar, bu konuda yapılmış çalışmalar incelenip, bu çalışmalardan yola çıkılarak dalgaboyu atamalı WDM ağlarda çoklu aktarım ağacı oluşturmak için kullanılabilecek yeni bir yaklaşım önerilmiş ve bu yaklaşım kullanılarak oluşturulan bir yöntem gerçeklenmiştir. Ayrıca, bu konuda daha önce geliştirilmiş olan algoritmalardan bir kısmı daha gerçeklenerek yeni yaklaşımın başarımı tespit edilmeye çalışılmıştır.

A NEW APPROACH FOR WDM MULTICAST ROUTING AND ITS PERFORMANCE ANAYSIS

SUMMARY

In this study, the objective is to improve the multicasting techniques and their performances in the optical networks expected to be the next generation infrasturcture of the backbones. The focus is on Wavelength Routed WDM Networks.

The WDM technique, can be considered as the optical equivalent of FDM which is used to utilize the bandwidth efficiently in the electronic networks. The frequencies allocated for the communication channels in FDM, become wavelengths allocated for the channels in WDM. On the other hand, Wavelength Routed WDM Networks can be described as networks consisting of wavelength routers and fibers connecting them, where each wavelength router decides which output gate and which wavelength should be allocated for each incoming signal wavelength.

Since optical switching technology is still an immature one, memory and logical operation requirements can not yet be met in the optical networks; thus in the wavelength routed WDM networks. As a result of this fact, the routing decision is still given in the electronic medium. Moreover, routing in WDM networks doesn't only include to find the minimum cost path between two network nodes, but also requires to allocate proper wavelengths along that path.

When it comes to multicast routing, a new problem arises besides the ones mentioned above, which is the problem of copying the multicast packets in the optical medium. The copying of the packets in the optical medium is realized by splitting the light (incoming signal) and the cost of a wavelength router which can manage to split light is really high. This makes it infeasable to construct a wavelength routed WDM network in which every wavelength router has the capability of splittilng light. On the other hand, in the electronic routers, an incoming packet can be copied as long as the memory has enough storage.Thus, the multicast routing algorithms implemented in the electronic medium, assumes that it is possible to duplicate packets freely in every node.

The efforts in this study, are to implement efficient multicast routing algorithms that would overcome the above mentioned problems while keeping in mind that these algorithms should be easily adoptable to electronic networks and the routing algorithms on them. To achieve this, the currently existing studies on the subject are examined. With the knowledge acquired from this analysis a new approach to construct multicast routing trees in wavelength routed WDM networks is presented and an example algorithm that uses this new approach is implemented. Also some of the already existing algorithms are implemented and their performance is compared to that of the new algorithm.

1. GİRİŞ

İnternet, haberleşme teknolojilerinin gelişmesi ile ortaya çıkmış ve farklı coğrafyalarda bulunan farklı bilgisayarların birbirleri ile haberleşmelerine olanak vermiştir. Bu sayede, insanlar birbirleri ile haberleşme, birbirlerine çeşitli nedenlerle veri aktarma imkanı bulmuşlardır. Dolayısıyla, dünyada internet üzerinden yapılmaya başlanan bir bilgi akışı oluşmuştur.

İnternetin kullanılmaya başlanmasından itibaren birçok ihtiyaca cevap verecek ve birçok kolaylık sağlayacak yazılımlar geliştirilmiştir. Bunlardan bazıları ve belki en çok kullanılanları e-posta programları, internet üzerinden sohbet programları ve internet tarayıcılarıdır. Bu gibi yazılımların artışı, internetin kullanımını arttırmış ve yaygınlaşmasına neden olmuştur. Sonuç olarak internet, günümüzde günlük hayatta bile sıkça kullanılan bir iletişim aracı olmuştur.

İnternet, yukarıda anlatıldığı gibi gelişirken genellikle yalnız iki bilgisayar arasında haberleşmeye yönelik yazılımlar geliştirilmiştir. Ancak, son yıllarda birden çok bilgisayarın birbirleri ile eş zamanlı olarak haberleşebileceği uygulamalara ihtiyaç duyulmaya başlanmıştır. Ortaya çıkan bu iletişim biçimi de *grup iletişimi* olarak adlandırılmıştır. Grup iletişiminin, bu iletişim tarzı için geliştirilecek özel yazılımlar olmadan gerçeklenmesi, internet üzerindeki veri trafiğinin önemli biçimde artmasına sebep olmaktadır [1].

Çoklu aktarım ise, grup iletişiminin daha verimli olarak gerçeklenmesini sağlamak üzere öne sürülmüş bir iletişim biçimidir. Bu iletişim yöntemi sayesinde, grup iletişimi nedeniyle internet üzerinde oluşan trafik yükünün kaydadeğer bir oranda düşürülebileceği öngörülmektedir. Günümüze kadar bu konuda yapılmış olan çalışmalar da bu yönde sonuçlar vermektedir.

Çoklu aktarım, hala üzerinde çalışılan ve gelişmekte olan bir iletişim biçimi olmakla beraber, internette çeşitli uygulamalarda kullanılmaya başlanmıştır. Bunlardan bazıları telekonferans, uzaktan eğitim, yazılım güncellemeleri, çağrılı video (ingilizce adıyla Video-On- Demand) gibi uygulamalardır. Bu uygulamalar varolan teknoloji ile gerçeklenirken, araştırma aşamasında bu iletişimin daha verimli bir şekilde

yapılması için çalışmalar sürmektedir. Bu çalışmalarda genellikle, çoklu aktarım yapılırken, mümkün olduğu kadar az ağ kaynağı kullanarak aktarım yapmak yanında, istenilen servis kalitesi (QoS) ölçütlerinin de sağlanması ve kullanılan çoklu aktarım yöntemlerinin ölçeklenebilir olması amaçlanmaktadır [2].

İnternette, çoklu aktarım ile beraber birçok başka uygulamanın da her geçen gün çok hızlı bir şekilde artması ve internet erişiminin gittikçe daha kolay ve ucuz bir hal alması ile [1], internet üzerinde akan veri trafiğinin hızla artmakta olduğu görülmektedir. Çoklu aktarımın, bir ağ üzerinde tekli iletişimlere göre çok daha fazla yük getireceği düşünüldüğünde ve çoklu aktarımın da yaygınlaştığı göz önüne alındığında, internet trafik yükünün ne hızla arttığı ve ne boyutlara vardığını anlamak daha kolay olmaktadır.

İnternet veri trafiğindeki bu artış, büyük miktarlarda veri taşıyan omurgalar üzerinde de olumsuz etkiler yaratmaktadır [3]. Omurgaların günümüzde kullandıkları teknolojiler, yakın gelecekte internet kullanıcılarının isteklerini yeterli derecede karşılayamayacaktır. Öte yandan, omurgaların altyapıları elektronik teknolojisinin limitlerini zorlamaya başlamıştır. Bu nedenle, artık omurgalarda elektronik teknolojisinin yerini alacak yeni bir teknolojiye ihtiyaç duyulmaya başlanmıştır [4]. Optik iletim teknolojisi, çok büyük bant genişliği ve oldukça yüksek hızlarda iletim yeteneği ile elektronik teknolojisinin yerini alabilecek çok önemli bir aday konumundadır [3-5].

Optik teknolojisi, avantajları nedeniyle son yıllarda yoğun olarak üstünde çalışılmaya başlanan bir alan olmaktadır. Bu konuda yapılan çalışmalardan da iyi sonuçlar alınmasına rağmen, bu teknolojinin halen yeterince olgunlaşmadığı ve daha alınacak çok yolun olduğu görülmektedir [6].

Optik ağların yüksek bantgenişliği kapasitesinin daha verimli olarak kullanılabilmesi için elektronik ağlarda olduğu gibi çoğullama yöntemleri geliştirilmiştir. Günümüzde, bu çoğullama yöntemlerinden pratik kullanıma en uygun olanı WDM (Dalgaboyu Bölmeli Çoğullama – Wavelength Division Multiplexing) olarak görülmektedir. Bu nedenle WDM konusundaki çalışmalar son yıllarda oldukça artmıştır.

Bu çalışmada, WDM tekniğinin kullanıldığı optik ağlara yönelik olarak geliştirilmiş çoklu aktarım yöntemleri üzerinde durulmuştur. Bu çoklu aktarım yöntemleri, çoklu aktarımı en iyi şekilde gerçeklemeye çalışırken, elektronik ağlardaki çoklu aktarım yöntemlerinden farklı olarak bazı ek kısıtlarları da göz önüne almak durumundadırlar.

WDM ağlarda çoklu aktarım ile ilgili en önemli sorun, düğümlerde paket çoğullama işleminin maliyetli oluşudur. Bu durum paket çoğullama işleminin WDM ağının her düğümünde gerçekleştirilebilmesini önlemektedir.

Bu çalışmada, WDM ağlarında çoklu aktarım için paket çoğullama problemini çözmek amacıyla gerçekleştirilen algoritmalar tek tek incelenmiş, gerçeklenmiş ve başarımları ölçülmüştür. Bunlara ilave olarak uygulamada daha verimli olacağı düşüncesiyle hiyerarşik yapıda yeni bir algoritma tasarlanmış, gerçeklenmiş ve başarımı ele alınan diğer algoritmalarla karşılaştırılmıştır.

Çalışmanın giriş bölümünü izleyen bölümleri şu şekilde özetlenebilir: İkinci bölümde, optik teknolojisi ve optik ağlar ile ilgili genel bilgiler bulunmaktadır. Üçüncü bölümde, çoklu aktarım ve çoklu aktarım protokollerine ilişkin bilgiler verilmiştir. Dördüncü bölümde, optik ağlarda gerçekleştirilmiş çoklu aktarım yöntemleri anlatılmıştır. Beşinci bölümde, bu çalışmada önerilen İki Katmanlı Kapsayan Ağaç Yöntemi'nin detaylı bir açıklamasına sunulmuştur. Altıncı ve son bölümde ise, önerilen yeni yöntem ile diğer yöntemlerin karşılaştırılması yapılmış ve elde edilen sonuçlar yorumlanmıştır.

2. OPTİK AĞLAR

İnternetin yaygınlaşması ve genişlemesi, kullanımının gittikçe artması, her geçen gün internet üzerinde taşınan veri trafiğinin artmasına neden olmaktadır.Trafiğin artışına karşılık, kullanıcıların istedikleri kalitede hizmet alabilmeleri için, ağlar üzerinde sağlanması gereken bant genişliği ve hız gibi özelliklerin sürekli daha ileriye götürülmesi gerekmektedir. Ancak, bakır tellerin bant genişliği ve elektronik ortamdaki işaretleşmenin hızı, artık bu artan yoğunluktaki trafiğe cevap vermekte yetersiz kalmaya başlamıştır [4,7] .

Optik ağlar ise, geniş bant genişlikleri, yüksek hızları ile gelecekte ağların altyapısını oluşturma yolunda önemli bir aday olarak ortaya çıkmaktadır [4]. Bant genişliği açısından artan gereksinimleri karşılayabilecek iki ortam, telsiz ortam (hava) ve optik fiberlerdir. Ancak, telsiz ortamın bant genişliği, sınırlı olduğu gibi, optik fiberlerle karşılaştırıldığında çok küçüktür.

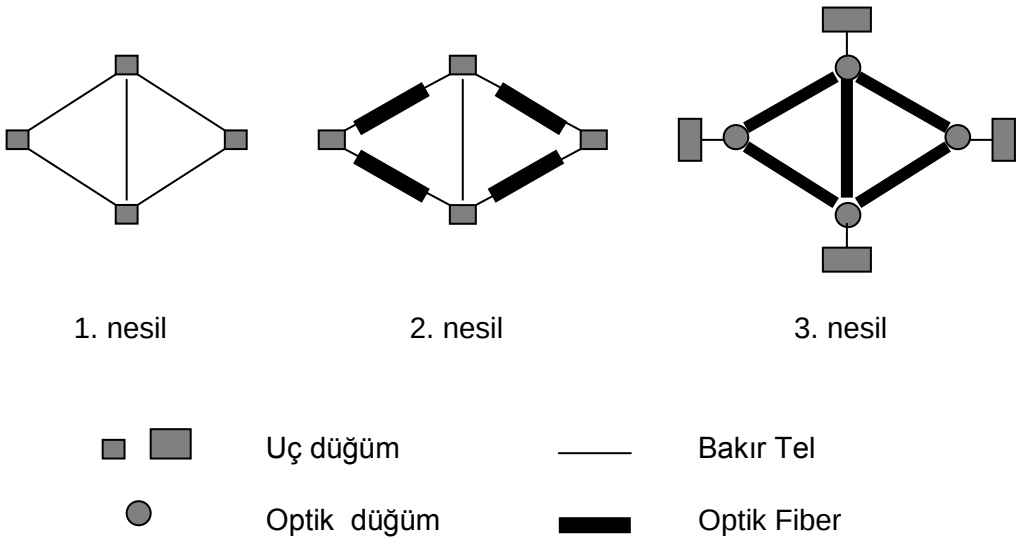
Optik fiberlerin başka bir özelliği de, iletim hata olasılıklarının çok düşük olmasıdır. Ayrıca fiberlerin bant genişliğinin büyük olması, veri sıkıştırması ve kodlaması gibi karmaşık işlemlerin yapılmasını bile gereksiz kılabilir. İletimin basit bir şekilde yapılması da, işaretlerin taşınması dışında, veri alışverişinde oluşabilecek hataların azalmasını sağlar [4].

Optik ağlarda kullanılan cihazlar da incelenirse, bu cihazların oldukça basit olduğu görülür [4]. Bu cihazların çoğu pasiftir ve veri saklama ile lojik işlem yapma gibi özellikleri yoktur. Bu özellikler, optik ağların daha dayanıklı olmasını, kullanımının ve bakımının da kolay olmasını sağlamaktadır.

Bütün bu özellikleri ile optik ağlar, gelecekte gereksinimleri karşılamak açısından en uygun çözüm olarak ortaya çıkmaktadır. Ancak, optik teknolojisi, elektronik teknolojisine göre çok yeni bir teknoloji olduğu için, tek başına optik teknolojisi, ağların yapılandırılmasında yeterli olamamaktadır (örneğin optik ortamda kullanılan bellekler çok yetersizdir.)

2.1 İletişim Ağlarında Üç Nesil

İletişim ağlarında, kullanılan teknoloji açısından bakıldığında 3 nesil gözümüze çarpmaktadır. Bunlardan ilki bakır tellerin kullanıldığı 1. nesil ağlardır. 2. nesil ağlarda ise, bu ağlar *opak ağlar* olarak da anılır, bakır tellerin yerini fiber lifler almıştır. Ancak, optik ortam, bu nesil ağlarda, yalnızca iletişim kuran iki varlık arasında işaretlerin taşınmasında kullanılmıştır; yani 1. nesil ağlardan 2. nesil ağlara geçişte değişen tek şey, işaretlerin taşınması için bakır teller yerine fiber liflerin kullanılması olmuştur. Tabii, iletişim kuran varlıklar değişmediği ve elektronik ortamda kaldığı için, fiber lifler üzerinden veri aktarılması ancak verinin alındığı ve verildiği noktalarda, optik ve elektronik işaretler arasında dönüşüm yapılmasını gerektirir. Bu durumda, iletim kuran iki nokta arasında ara düğümler olması halinde, iletişim hızını büyük ölçüde belirleyen şey, ara düğümlerdeki eletronik donanımın hızıdır. Halbuki, iletişim kuran iki uç arasında iletilen işaret, sürekli optik ortamda kalabilirse, yani iletilecek veri alıcıda elektronik/optik dönüşümü ve vericide de optik/elektronik dönüşümü geçirirken, ara düğümlerde iletimin hep optik olması sağlanabilirse, uçtan uca gecikme oldukça azalacaktır. İşte 3. nesil iletişim ağlarında bu yöntem kullanılmıştır ve bu ağlara *saf optik* (İngilizce karşılığı all-optical) ağlar adı verilmektedir. Şekil 2.1'de her üç nesil ağlara ilişkin farklılıklar gösterilmiştir [4].



Şekil 2.1: İletişim ağlarında üç nesil.

Saf optik ağlar, iletişimin iki uç arasında hep optik ortamda kalması nedeniyle saydamlık sağlarlar; yani farklı bit hızları, farklı veri biçimleri ve farklı protokolleri destekleyebilirler [6]. Opak ağlarda ise her düğümde işaretlerin elektronik ortama geçmesi nedeniyle iletişimin ara düğümlerdeki kısıtlamalara uyması zorunluluğu doğacaktır. Bu nedenle opak ağlar saydam değildir. Buna karşılık, opak ağlarda uçtan uca işaretin zayıflaması söz konusu değildir; çünkü her düğümde işaret elektronik ortama geçirilip kuvvetlendirilir. Saf optik ağlarda ise işaretin zayıflaması söz konusudur [6] ve bunu engellemek için optik kuvvetlendiriciler kullanmak gerekebilir.

Opak ağların bir başka avantajı da başarımın denetlenmesinin mümkün olmasıdır. Saf optik ortamda ise bu denetimin çok sınırlıdır; çünkü optik ortamda bellek gerçeklemek zordur [6].

2.2 WDM ve Diğer Optik Çoklama Yöntemleri :

Optik fiberlerin sağladığı yüksek bant genişliğinin verimli olarak kullanabilabilmesi için, aynı anda bir çok iletişimin beraberce yapılmış, yani çoklama yöntemlerinin geliştirilmiştir. Bunlardan bazıları OTDM (Optik Zaman Bölmeli Çoğullama - Optical Time Division Multiplexing) , CDM (Kod Bölmeli Çoğullama – Code Division Multiplexing) ve WDM (Dalgaboyu Bölmeli Çoğullama - Wavelenght Division Multiplexing) ‘dir [7].

2.2.1 OTDM

Çok sayıda düşük hızlı iletişim kanalı, çok kısa süreli darbeler halinde getirilip, her kanalın, birbirinin üzerine binmeyecek şekilde, zaman içinde paylaştırılması ve tek bir yüksek hızlı kanal üzerinden iletilmesi ile gerçekleştirilir [7]. Bu yöntem kullanılarak bir ağın veri taşıma kapasitesinin 100 Gigabit/s gibi bir değere çıkarılması mümkündür. Buna karşılık, iletişim yapan uçlardaki alıcı ve vericilerin çok kısa süreli darbeler (zaman dilimleri) oluşturabilecek ve bunları sezebilecek yeteneğe sahip olması ve bunu yapabilmek için, iletilen kanal içindeki zaman dilimleri ile eşzamanlı olarak çalışabilmesi gerekmektedir .

2.2.2 CDM

CDM [7], birden fazla sayıda düşük hızlı iletişim kanalının her birinin benzersiz (tek) bir kodlama dizisi ile kodlanarak, aynı iletişim kanalı üzerinden iletilmesi yöntemidir.

Burada, iletişim kanallarının birbirleri ile karışmaması için, her bir düşük hızlı iletim kanalında kullanılan kodlama dizilerinin karşılıklı ilişkilerinin (cross-correlation) düşük olması gerekir. Ayrıca, fiziksel bant genişliğinin büyük olması karşılıklı ilişkileri az olan çok sayıda kod kullanılabilmesine olanak tanır. Bu sayede, yine ağ kapasitesinin elektronik ortamdakilerden daha yukarılara çekmek mümkün olmaktadır.

2.2.3 WDM

Bir optik fiber içindeki dalgaboyu spektrumunda, kayıbı düşük olan dalgaboylarının herbiri, diğerlerinden bağımsız çalışan bir iletişim kanalı olarak düşünülebilir; yani her bir dalgaboyu üzerinden eşzamanlı olarak veri iletimi yapılabilir. WDM de, iletim kanallarının bu şekilde çoğullanması ile gerçekleştirlen çoğullama yöntemidir. WDM ile ağın veri taşıma kapasitesi bir fiberde kullanılabilecek dalgaboyu sayısı ile her bir dalgaboyunda yapılacak iletişimin hızının çarpımı ile orantılıdır. Kullanıcılar, isterlerse, iletişimi birden fazla dalgaboyunu eşzamanlı olarak kullanarak daha verimli hale getirebilirler [7].

Günümüzde, WDM ‘in OTDM ve CDM ’ye göre tercih edildiği görülmektedir [7]. Bunun nedenleri çeşitlidir. Öncelikle OTDM ve CDM için gerekli donanım, eşzamanlama sorunları nedeniyle oldukça karmaşık ve maliyetlidir. Bu nedenle OTDM ve CDM, henüz bu teknolojilerin yeterince gelişmemiş olmasının da etkisiyle, daha uzun vadede ele alınan çözümler olarak karşımıza çıkmaktadır. Buna karşılık, WDM’in hayata geçirilebilmesi için, gerekli teknoloji günümüzde mevcuttur. WDM ’in 2. bir avantajı da, iletişimde saydamlık sağlamasıdır. İletişim sırasında, her bir dalgaboyu diğerinden bağımsız bir iletişim kanalı olarak davrandığı için, bağlantı kurmuş olan kullanıcılar kendi aralarında anlaşarak, iletimin hızı, kullanacakları protokolleri vs. belirleyebilirler. Bu sayede, ağ üzerindeki iletişimin oldukça esnek olması sağlanır; çünkü WDM kullanan ağların günümüz iletişim protkolleri yanında gelecekte kullanılacak protokolleri de desteklemesi mümkün olacaktır.

2.3 WDM Ağ Mimarileri

WDM ağ mimarileri üç farklı şekilde sınıflandırılmaktadır. Aşağıda bu sınıflar ayrı ayrı incelenmiştir.

2.3.1 Pasif Optik Ağlar (PON)

Pasif optik ağ mimarisinde [7], iki temel birim vardır. Bu temel birimler, Merkezi Ofis (Central Office) ve Optik Ağ Birimleridir (ONU [Optical Network Units]) . Merkezi ofis ile ONU'lar arasında fiber linkler bulunmaktadır. Bu mimaride, ONU'lar ve merkezi ofis, aralarında iki yönlü bağlantılar kurdukları bir ağaç yapısı oluşturlar ve yol atama işlemi de merkezi ofiste yapılır. Bu mimari, bölgesel iletişim için uygun bir çözümdür. Bu mimarinin gerçekleştirilmesindeki en önemli teknolojik sorun, ucuz, basit ve dayanıklı ONU'ların tasarlanmasıdır.

2.3.2 Yayın Seçici Ağlar (Broadcast and Select Networks)

Yayın seçici ağlar [7], saf optik WDM mimarisinin en basit türlerinden biridir. Bu ağlarda, tüm altyapı cam malzemeden oluşmaktadır. Her uç düğümde alıcı ve vericiler vardır. Vericiler, birbirinden farklı dalgaboylarında işaretleri oluştururlar ve bütün vericilerin oluşturdukları bu işaretler bir araya getirilip bütün dalgaboylarındaki işaretler uç düğümlerin alıcılarına iletilir. Bir araya getirilen işaretler, tüm düğümlere aktarıldığı için, ağ içinde sürekli bir yayın söz konusudur. Uç düğümlerin istedikleri dalgaboyundaki işareti alabilmeleri için, alıcılarının ya ayarlanabilir olması ya da tek bir dalgaboyundan işaret alabilen birden fazla alıcısı olması gerekir. Alıcıların, bir dalgaboyundaki işareti alması, istenilen dalgaboyunu geçirip diğer dalgaboylarındaki işaretleri süzmesi ile gerçekleşmektedir. Vericilerin de alıcılar gibi ya ayarlanabilir olur ya da her uç düğüm farklı dalgaboylarında işaret üreten birden fazla verici kullanır.

Yayın seçici ağların ağ kapasitesi, doğrudan doğruya kullanılabilecek dalgaboyu sayısına bağlıdır. Dolayısıyla, ne kadar çok dalgaboyu sağlanabilirse o kadar büyük miktarlarda verinin ağ üzerinde dolaşması mümkündür [7].

Yayın seçici ağlar, çalışma ilkeleri gereği, yayın ve çoklu aktarım türündeki aktarımları doğrudan desteklerler. Yayın ve çoklu aktarım, internette çok kullanılan iletişim yöntemleri olduğundan bu ağların kullanım alanının çok geniş olabileceği öngörülebilir.

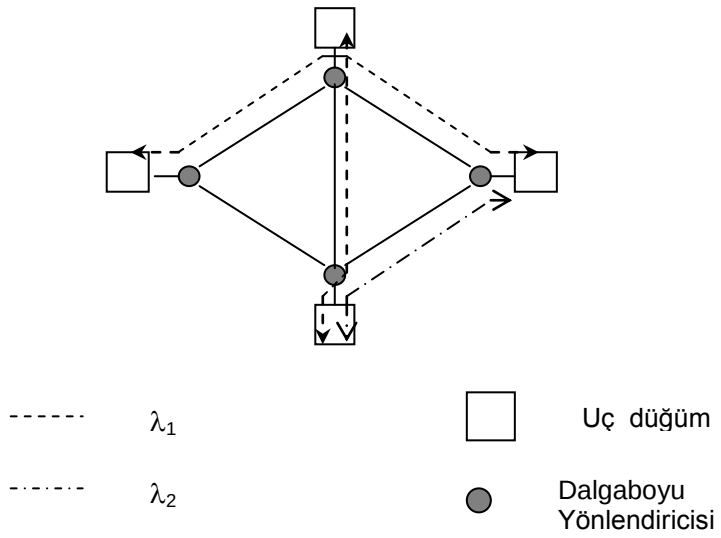
2.3.3 Dalgaboyu Atamalı Ağlar (Wavelength Routing Networks)

Yayın seçici ağlarda [7], ışığın bölünmesi ve dalgaboylarının verimli kullanılamaması bu ağların geniş alanlarda kullanılmasını engellemektedir. Dalgaboyu atamalı WDM ağlarda ise, iletilen ışığın bölünmeden başka bir yere yönlendirilmesi yoluyla iletilen

işaretin çok daha az zayıflaması sağlanmıştır. Ayrıca, bir dalgaboyundaki işaret birçok link üzerine gönderilmek yerine, belli başlı linklere gönderilerek, bu işaretin kullanılmayacağı linklerde dalgaboyunun boşuna işgal edilmesi engellenmiş olur.[7]

Dalgaboyu atamalı ağlar [7], bir veya daha çok dalgaboyu seçici öğe ile bu öğeleri birbirine bağlayan optik fiberlerden oluştur. Dalgaboyu seçici olan öğeler, tamamıyla cam kullanılarak yapılmış, optik-elektronik işaret dönüştürme yapmayan ağ öğeleridir ve dalgaboyu yönlendiricisi olarak adlandırılırlar.

Dalgaboyu yönlendiricileri [7], giriş kapılarındaki bir işareti herhangi bir çıkış kapısına yönlendirme yeteneğine sahiptir. Bir işaretin, hangi çıkış kapısına yönlendirileceği, işaretin geldiği giriş kapısı ile işaretin dalgaboyuna bakılarak belirlenir. Ayrıca, farklı dalgaboyunda olan işaretlerin aynı çıkış kapısına iletilmesi mümkündür; ancak aynı dalgaboyundaki işaretlerin aynı çıkış kapısına yönlendirilmesine izin verilmemelidir.



Şekil 2.2: Örnek bir WDM Dalgaboyu atamalı ağ.

Dalgaboyu atamalı ağlarda bulunan her dalgaboyu yönlendiricisi, ya başka bir dalgaboyu yönlendiricisine ya da bir uç düğüme (kullanıcıların iletişimine ilişkin alıcı ve/veya verici olan düğümler) bağlı olabilir. Örnek bir dalgaboyu atamalı ağ Şekil 2.2'de verilmiştir. İletişimde bulunmak isteyen uç düğümler, elektronik-optik işaret dönüşümü yapıldıktan sonra, verilerini bağlı oldukları dalgaboyu yönlendiricisine aktarırlar ve bundan sonra veri alıcıya ait uç düğümün bağlı olduğu

dalgaboyu yönlendiricisine varana kadar optik ortamda kalır. Optik ortamda verinin izlediği bu yola *ışıkyolu* adı verilir [7].

Bir dalgaboyu yönlendiricisinin aynı çıkış kapısına yönlendirilmiş iki faklı işaretin aynı dalgaboyundan olmasına izin verilmemesi gerektiğinden, en az bir fiber linki paylaşan iki ışıkyolunun paylaşılan fiber üzerinde aynı dalgaboyunu kullanmamaları gerekmektedir [7].

Uç düğümlerde, her bir ışıkyolu için bir tek alıcı/verici çifti bulunmaktadır. Bu yüzden bir uç düğümün ağ kaynaklarından verimli bir şekilde yararlanabilmesi için (örneğin eşzaman birden bağlantı kurabilmek için) birden fazla alıcı/verici çiftine sahip olması gerekmektedir.

2.3.3.1 Dalgaboyu Yönlendiricilerinin Özellikleri

Dalgaboyu Ekleme ve Ayırma

Bazı dalgaboyu yönlendiricileri, bir girişinden gelen optik işaret ile, başka bir girişinden gelen ve diğer girişteki işaretten farklı dalgaboyuna sahip bir optik işareti üstüste bindirip aynı fiber üzerine iletebilirler. Gelen iki işaretin dalgaboyları birbirleri ile çakışmadığından iki işaret de bozulmadan aynı fiber üzerine yönlendirilebilmektedir. Benzer şekilde yönlendiriciler girişlerinden gelen optik işarete ait bazı dalgaboylarını diğerlerinden ayırıp farklı bir çıkışa verebilirler . Ayrıca dalgaboyu yönlendirici düğümler kendilerine gelen bir işaretin bir kısmını kendileri kullanıp geri kalan kısmını da başka bir yönlendirici için kullanabilirler; böylece belli bir dalgaboyunda gönderilen paketler hem yönlendiricinin bulunduğu düğüm tarafından alınmış, hem de bir başka düğüme gönderilmiş olur [7].

Dalgaboyu Dönüştürme

Eğer bir dalgaboyu atamalı ağda, dalgaboyu dönüştürücülerinde dalgaboyu dönüşümü yapılamıyorsa, iki uç düğüm arasında kurulan bir ışıkyolunun, üzerinden geçtiği tüm fiber linklerde aynı dalgaboyu kullanılması gerekmektedir. Bu kısıtlamaya *dalgaboyu devamlılığı kısıtı* adı verilir [7].

Dalgaboyu devamlılığı kısıtı, dalgaboyu atamalı ağlardaki yol atama işleminin karmaşıklaşmasına neden olur; çünkü iki uç düğüm arasında en az bir dalgaboyunun serbest olduğu bir yol bulunsa bile, bu yol üstündeki tüm fiber linklerde aynı dalgaboyunun serbest olması gerekecektir. Yol üstündeki en az bir

fiberdeki serbest dalgaboyunun, kurulmak istenen yol üstündeki diğer fiberlerdeki dalgaboylarından farklı olması, o yolun kurulabilmesini engelleyecektir ve yeterli kaynak olmasına rağmen yol kurulamayacaktır. Bu durumda, bağlantının kurulması için farklı bir yol aranması gerekecektir. Halbuki yol üstündeki dalgaboyu yönlendiricilerinde dalgaboyu dönüştürme yeteneği bulunursa ve her fiberde kurulacak yeni yol için kullanılabilecek bir dalgaboyu mevcut ise ışıkyolu oluşturulup bağlantı kurulabilecektir [7].

Dalgaboyu dönüştürme işlemi, dalgaboyu atamalı bir ağın devre anahtarlamalı bir ağ gibi çalışmasını sağlar [7]. Bu sayede, ağ kaynaklarının kullanımındaki verim arttırlmış olur. Bunun yanında, dalgaboyu dönüştürme işlemi, ağ üzerinde farklı sayıda dalgaboyu kapasitesine sahip linklerin bulunması durumunda faydalı olabilir. Ayrıca, birbirinden bağımsız alt ağlar arasındaki geçiş bölgelerinde, ağların birbirinden bağımsız çalışmalarını engellememek için kullanılması faydalı olabilir.

Dalgaboyu dönüştürme işlemi iki farklı şekilde yapılabilir [7] : Optik-elektronik işaret dönüşümü yapılarak veya optik ortamda. Optik-elektronik dönüşüm yapılarak dalgaboyu dönüşümü yapılması, optik işaretin elektronik hale dönüştürüldükten sonra, elektronik ortamda kuvvetlendirilip başka bir dalgaboyunda bir optik işarete çevrilmesi ile yapılır. Bu yöntem karmaşıktır ve oldukça fazla güç harcanmasına neden olur. Ayrıca, işaret elektronik ortama geçtiği için saydamlık bozulur ve iletim yine elektronik hız sınırlamaları ile karşı karşıya kalır. Optik ortamda dalgaboyu dönüşümü ise bu sakıncaları taşımaz; çünkü dönüştürme işlemi için elektronik ortam geçiş yapılması gerekmemektedir.

Işık Bölme Yeteneği

Dalgaboyu yönlendiricileri, bir girişlerinden gelen ve belli bir dalgaboyunda olan bir işareti bölme yeteneğine sahip olabilirler [3,5,7,8] . Bu yetenek, optik ortamda işaretin çoğullanmasını, diğer bir değişle kopyalanmasını sağlar [3]. Elektronik ortama geçmeden bir işaretin çoğullanmasının sağlanabilmesi, çoklu aktarım ve yayın yapılması söz konusu olduğunda önem kazanır; çünkü bu durumda yine optik ortamın avantajlarından faydalanmaya devam edilirken, elektronik ortama ihtiyaç duyulmadan işaret çoğullanmış olur. Ayrıca, işaretin çoğullanması için elektronik ortamdakinden daha az zaman kaybedilir; çünkü elektronik ortamda verinin kopyalanabilmesi için öncelikle bir bellekte saklanması gerekmektedir. Halbuki, ışık bölme yöntemi ile işaret çoğullandığında böyle bir gecikme söz konusu olmaz.

Işık bölmenin yukarıda anlatılan avantajlarına karşılık, önemli dezavantajları da vardır. Bunlardan biri ışık bölme işleminin oldukça maliyetli olmasıdır. Işık bölmenin maliyetinin yüksek olması nedeniyle, dalgaboyu atamalı ağlarda bu yöntemin kısıtlı olarak kullanılması daha uygun gözükmektedir. İki farklı şekilde kısıtlanma yoluna gidebilir : i) Ağdaki tüm dalgaboyu yönlendiricilerinde ışık bölme yeteneği olur; ancak yönlendiriciler bir girişinden gelen ışığı tüm çıkışlarına bölemezler (Kısıtlı ışık bölme). ii) Ağdaki yönlendiricilerin yalnız bir kısmı ışık bölme yeteneğine sahiptir ve bu yönlendiriciler bir girişlerine gelen bir işareti tüm çıkışlarına aktarcak şekilde bölebilir (Seyrek ışık bölme) [3,5,8].

Işık bölmenin bir diğer dezavantajı da, bölünen ışık demetlerinin zayıflamasıdır. Işığın bölünme oranı da zayıflama miktarını etkileyecektir. Dolayısıyla optik ortamda bir işaretin çoğaltılma sayısı, elektronik ortamdaki gibi sınırsız olarak düşünülemez. Buna karşılık, işaretleri optik kuvvetlendiriciler yardımıyla kuvvetlendirmek mümkün olduğu için çoğaltma oranı istenilirse arttırılabileceği açıktır.

Bellek Kapasitesi

Optik ortamda, bilgi saklamak günümüz teknolojisi ile mümkün değildir [5]. Ancak, uzun optik fiberler kullanılarak yapılan fiber gecikme sarımları (fiber delay lines) ile bir miktar gecikme sağlamak mümkündür. Bu şekilde, optik ortamda eşzamanlı olarak yapılamayan işlerin, sıralı bir şekilde gerçekleştirilmesi mümkün olabilmektedir.

Genel olarak bellek, günümüz yönlendiricilerinin, yol atama sırasında kullanmak zorunda oldukları bir öğedir; çünkü bir verinin nereye yönlendirileceği belirlenirken, verinin bir yerde bekletilmesi gerekmektedir. Amaç verinin (ışık demetinin) beklemesi olduğundan, optik ortamda da yönlendiricilerde bellek görevini görmek üzere fiber gecikme sarımları kullanılabilmektedir.

Optik ortamda belleğin gerçekleştirilmesinin zor olması yanında, lojik işlemlerin gerçekleştirilmesi de mümkün değildir. Bu durumda, optik ortamda herhangi bir karar mekanizmasının oluşturulması söz konusu değildir. Yol atama işlemi, ağın içinde bulunduğu yerel veya global durum göz önüne alınarak verilmesi gereken bir karar olduğundan, doğası gereği, bir karar mekanizmasına ihtiyaç duyar. Bu ihtiyaç, optik ortamda karşılanamadığı için elektronik ortamın kullanılması zorunlu hale gelmektedir. Tabii bu durum, uçtan uca iletişimin tamamen optik ortamda olmasını engellemektedir.

2.4 Dalgaboyu Atamalı Ağlarda Yol Atama

Dalgaboyu atamalı ağlarda yol atama, elektronik ağlarda ağ katmanında yapılan yol atamadan farklılıklar gösterir. Öncelikle, elektronik ağlarda, yol atama işlemi her düğümde tekrarlanır. Dalgaboyu atamalı ağlarda ise, yol atama işlemi uçtan uca gerçekleştirilir [3]. Bunun en önemli nedeni, oluşturulacak olan bir ışıkyolu üzerindeki fiberlerde hangi dalgaboyunun seçileceğinin belirlenmesi gerekliliğidir. Daha önce de açıklandığı gibi, ışıkyolu oluşturulurken dalgaboyu devamlılığı kısıtına uyulması gerekiyorsa, kaynaklar uygun olan bir yol seçilmiş olsa da dalgaboyu devamlılığı kısıtı nedeniyle bu yoldan vaz geçilmesi ve yeni bir yol seçilmesi gerekebilir. Bu durumda, uçtan uca bağlantı kurulması aşamasının fazla uzun sürmemesi açısından yönlendirme işleminin uçtan uca yapılması uygun olmaktadır. Dalgaboyu dönüştürme yöntemi kullanarak, dalgaboyu seçiminin önemi ortadan kaldırılabilir ve yol atama işlemi her düğümde yapılabilir; ancak bu maliyetli bir çözümdür [3,5,8].

Elektronik ortamda kullanılan ağlarda, denetim işaretleri veri işaretleri ile beraber aynı iletişim kanalı üzerinden gönderilmektedir. Her bir paket ile ilgili yönlendirme bilgileri, paketlerin başlık bilgileri kısmında bulunur ve bu bilgiler paketin bir parçasıdır. Optik ortamda ise, herhangi bir karar mekanizması olmadığı için, gelen bir optik paketin içindeki hangi kısmının başlık, hangi kısmının veri olduğunun belirlenmesi mümkün değildir. Bu ayrımı yapabilmek için, yine elektronik bir karar mekanizmasının yardımına ihtiyaç vardır. Ancak başlık ve veriler farklı kanallar üzerinden gönderilerek, hangi işaretin veri hangi işaretin başlık (denetim bilgisi) olduğunu belirlemek mümkün olabilir. Bunun için, dalgaboyu atamalı ağlarda genelde denetim işaretlerinin iletişimin yapıldığı atanmış bir dalgaboyu bulunur. Geriye kalan dalgaboyları da veri iletişimi için kullanılır.

Optik ortamda uçta uca yol atama işlemi, dalgaboyu seçimi nedeniyle iki aşamalı bir süreç olarak görülür : i) Yol seçimi ii) Dalgaboyu atama [3,8]. Yol seçiminde, iki uç arasında kaynakları uygun ve belli servis kalitesi ölçütlerini sağlayan bir yolun seçilmesi işlemi gerçekleştirilir. İkinici aşama olan dalgaboyu atama işleminde ise, seçilen bu yol üzerinde ışıkyolunun seçilebilmesi için gereken dalgaboyu/dalgaboylarının bu ışıkyoluna tahsis edilmesi işlemidir.

Yol atama işleminin, dalgaboyu atama aşamasından bağımsız gerçekleştirmek için farklı çözümler bulunabilir. Bunlardan bir tanesinin dalgaboyu dönüşümü olduğu ve bu yöntemin maliyetli olduğu daha önce de açıklanmıştı. Başka bir yöntem de yol atamanın yapılacağı ağın, fiber linklerin taşıdığı dalgaboyu sayısı kadar

çoğullandığını düşünüp, her bir dalgaboyuna ilişkin ağda ayrı bir yol atama işlemi yapmaktır. Bu sayede sadece elektronik ortamda olduğu gibi bir yol atama işlemi yapılması söz konusu olacaktır. Ancak, bu yöntem de fiber linklerin taşıdığı dalgaboylarının sayısının artması ile kotarılması güç bir hal alacaktır.

Elektronik ortamdaki yol atama ile dalgaboyu atamalı ağlardaki (ve genelde bütün optik ağlarda) yol atama işlemlerinde, ayrıca yol atamaya karar aşamasında değerlendirilmesi gereken bilgi miktarları da farklılık göstermektedir. Dalgaboyu atamalı ağlarda yol atama yapılırken, elektronik ortamda yol atama yapmak için gerekli veriler yanında, dalgaboyu yönlendiricilerinin birçok özelliğinin bilinmesi, ağdaki bazı fiziksel verilerin de, örneğin iletilen işaretin zayıflama miktarı, değerlendirilmesi gerekmektedir [3]. Ayrıca istenilirse, elektronik ortamda olduğu gibi hata kotarma (fault-tolerance) için de gerekli bilgilerin değerlendirilmesi mümkündür. Bu durumda, ağdaki düğümlerin, elektronik ortamdaki gibi yol atama ile ilgili birbirlerine mesaj göndermeleri halinde, bu mesajların taşıdığı veri boyutunun oldukça büyük olması söz konusu olur ve işlemin başarımı bu nedenle düşebilir.

2.5 Dalgaboyu Atamalı Ağlarda Yeni Anahtarlama Yöntemleri

Dalgaboyu atamalı ağlarda yeni anahtarlama yöntemleri, optik patlama anahtarlaması (Optical Burst Switching - OBS) ve optik etiket anahtarlaması (Optical Label Switching - OLS) olarak iki bölümde incelenecektir.

2.5.1 Optik Patlama Anahtarlaması (Optical Burst Switching - OBS)

Optik Patlama Anahtarlaması, optik paketlerin karar aşamasında elektronik ortama geçirilmesi ihtiyacı ile optik paketlerin mümkün olduğunca optik ortamda kalması isteğini birleştiren bir yöntemdir. Bu yöntem elektronik ağlarda kullanılan kurtçuk deliği yönlendirmesine de benzetilebilir.

Optik paketlerin, karar aşamasında elektronik ortama geçirilmesi bir zorunluluktur. Ancak, aslında optik paketin tümünün elektronik ortama geçirilmesine ihtiyaç yoktur; çünkü paketlerin karar aşamasında kullanılan tek bölümü başlık kısımlarıdır. Bu durumda, yalnız başlık kısmı elektronik ortama geçirilip incelenirse, karar verme aşaması kısalacaktır; çünkü tüm paketin optik ortama aktarılması gerekmeyecektir.

Optik Patlama Anahtarlaması da [9,10], paketlerin başlık kısımlarının bir denetim kanalı (dalgaboyu) üzerinden (bu denetim kanalı, yalnız denetim işaretleri için kullanılan bir kanaldır), veri kısımlarının da, bir veri kanalı üzerinden (denetim kanalına ait dalgaboyu dışındaki herhangi bir dalgaboyu) belli bir zaman farkı ile gönderilmesine dayanır. Başlık kısmı veri kısmından belli bir süre önce yol üzerindeki bir sonraki düğüme gönderilir. Bu süre başlığın sonraki düğüme veriden önce varıp, o düğümde verinin nereye yönlendirileceğine karar verilmesi için belirlenmiş bir süredir. Böyle bir süreye ihtiyaç duyulmasının nedeni, optik ortamda bellek gerçekleştirilememesidir.

Optik Patlama Anahtarlaması'nda, başlık kısmında taşınması gereken veriler şöyledir : Başlığa ilişkin verinin geldiği dalgaboyu, verinin gönderilmek istendiği varış, başlık ile veri arasındaki zaman farkı, verinin boyu (uzunluğu). Başlık ile veri arasındaki zaman farkı ile verinin uzunluğu, verinin ne zaman ilgili dalgaboyunda okunmaya başlanacağını ve verinin ne zaman sonlanacağını belirlemek için kullanılır.

Bir başlık, belli bir düğüme ulaştığında, o düğümde o başlığa ilişkin verinin yönlendirilebileceği bir kaynak yoksa, gelen veri düşürülür. Eğer hata denetimi ulaşım katmanında yapılmıyorsa, optik paketi gönderen kaynağa paketin düşürüldüğüne dair bir olumsuz alındı mesajı gönderilmelidir. Ulaşım katmanında hata denetimi yapılıyorsa nasıl olsa, belli bir zaman aşımından sonra, paket tekrar gönderilmeye çalışılacağı için optik ortamda bir mesajlaşmaya ihtiyaç duyulmayacaktır.

Bir optik paketin veri kısımının tümünün yönlendirildiği fiber link üzerine çıkması ile optik paketin düğüme geldiği fiber linkte, paketin kullandığı dalgaboyu serbest bırakılır. Bu sayede bu kaynak başka bir paket için kullanılabilecektir.

2.5.2 Optik Etiket Anahtarlaması (Optical Label Switching - OLS)

Optik Etiket Anahtarlaması, OBS'e çok benzer. OLS'te de OBS gibi yalnız paketlerin başlık kısmının elektronik ortamda incelenmesi ve veri kısmının süreli optik ortamda ilerlemesi mantığı kullanılır. Asıl farklılık başlık kısmındadır. OLS' te, başlık kısmında OBS 'teki gibi bir çok bilgi olması yerine, etiket denilen çok kısa sayısal değerler konur. Bu etiketler ATM'deki gibi, iki düğüm arasındaki yerel bağlantı numarasını belirtir. İki düğüm arasında kurulan bu yerel bağlantıya LSP (Etiket Anahtarlamalı

Bağlantı Yolu) adı verilir. Bu şekilde bir başlık kullanılmasındaki amaç, yönlendirmenin daha hızlı bir şekilde gerçekleştirilebilmesidir.

OLS'te aslında bir başlıkta verilen her etiket değeri, bir dalgaboyuna karşı gelmektedir. Tabii, bir fiber üzerindeki dalgaboyu sayısı çok büyük değerlere ulaşamadığından, etiket değerleri de çok küçük olmaktadır. Etiketlerle yönlendirme de, bir giriş etiketine karşılık bir çıkış etiketinin seçilmesi şeklinde yapılır. Her bir giriş ve çıkış etiketinin düğümün hangi kapısına karşı geldiği de yönlendirmenin hangi iki kapı arasında yapılacağını belirlemek için kullanılır [11].

OLS'te başlığın kısa olması ve bu sayede yönlendirme sırasında karar verme işleminin kısa sürmesi nedeniyle başlık ve veri kısımları yine OBS'te olduğu gibi ayrı kanallardan ancak aynı anda gönderilir. Başlık ve veri bir sonraki düğüme ulaştıklarında, yönlendirme kararının verilmesi için geçen sürede, veri kısmı, bir fiber gecikme sarımı kullanılarak geciktirilir. Bu şekilde, başlık kısmına, başlık ve veri kısımları arasında beklenmesi gereken süre değerinin konulması gerekmez ve başlık bilgisi de kısalmış olur.

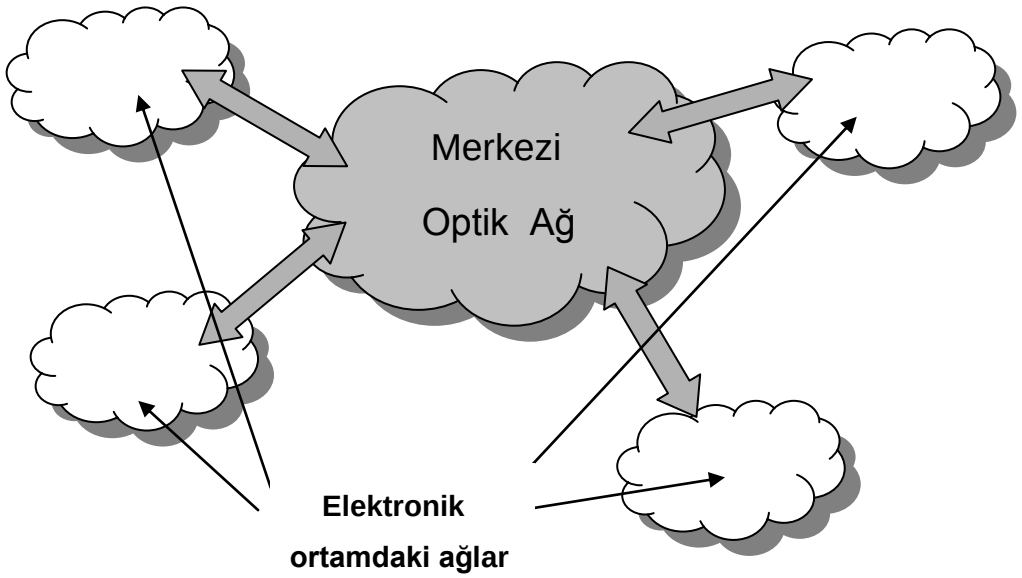
2.6 IP - Optik Ağ Bütünleşmesi

IP 'nin internette çok kullanılıyor olmasından dolayı, optik ağların doğrudan IP katmanı altında çalışabilecek bir arayüz sağlaması ve ATM, SONET gibi ara katmanlar ile ilişkili bir arayüzünün oluşturulmasına gerek olmadığı düşünülmektedir. Bunun nedeni, GMPLS (Genelleştirilmiş Çok Protokollü Etiket Anahtarlama - Generalized Multiprotocol Label Switching) çalışması çerçevesinde IP ile optik ağların bütünleştirilmesinin daha uygun olduğunun düşünülmesidir [6].

GMPLS çalışması, zaten çok sayıda protokolün internette beraberce yer alabilmesi için yapılan bir çalışmadır. Bu çalışma MPLS'i (Çok Protokollü Etiket Anahtarlama - Multiprotocol Label Switching) taban olarak alır. MPLS de, birçok protokolün beraberce IP kullanılarak internet üzerinde kullanılması için geliştirilmiştir. MPLS'de [11], her pakete bir başlık niteliğinde olan, ancak başlıktan çok daha kısa olan etiketler verilir. Yönlendirme, bu etiketler bakılarak yapılır. Her bir etiket iki düğüm arasında geçerlidir (ATM'deki sanal bağlantı numaraları gibi.). Bu nedenle yönlendirme tabloları da, gelen bir paketteki etikete karşılık (giriş etiketi) paketin hangi linke yönlendirileceğine dair başka bir etiket (çıkış etiketi) bulunacaktır. Bu yönlendirme işlemi çok basit ve oldukça hızlıdır. Etiketler, istenilen her tür paketin başına eklenerek, istenilen her protokolün internet üzerinde yönlendirilmesini

sağlayabilirler. Bu yüzden, MPLS oldukça esnek bir yapı sağlar. Ancak, MPLS'de etiketlerin düğümlere dağıtımının nasıl yapılacağı önem kazanmaktadır. MPLS'in genelleştirilebilmesi için geliştirilmesi gereken bölümü de etiket dağımı işlemidir. Bununla ilgili olarak değişik protokoller üzerinde çalışılmaktadır.

IP 'nin optik ağlar ile bütünleştirilmesi sırasında düşünülmesi gereken bir nokta da kullanılacak olan ağ mimarisidir [6]. Daha önce de söz edildiği gibi, optik ağların başarımının en iyi olması açısından, saf optik ağların kullanılması gerekmektedir. Bunun için, IP ile optik ağların bütünleşmesinde düşünülen mimari, optik ve elektronik ortamların birbirlerinden ayrı olduğu ve optik ortamda bulunan ağın bir omurga görevi gördüğü bir yapıdır. Buna göre, düşünülen mimarinin çok basit bir modeli Şekil 2.3'te verilmiştir. Şekilde gösterilen merkezi optik ağ, içinde optik alt ağların da bulunabileceği genel bir ağ olarak düşünülmelidir.



Şekil 2.3: IP – Optik ağ bütünleşmesine ilişkin mimari

Şekil 2.3'de görülen mimari iki farklı şekilde modellenebilir. Birinci model, merkezi optik ağın bir sunucu, dışındaki elektronik ortam ağların da birer istemci gibi davranmasıdır. Mimari, bu şekilde kullanılıdığında optik ortam ile elektronik ortam birbirinden yalıtılmış olur ve elektronik ortamdaki düğümler, optik ortamdaki düğümler hakkında herhangi bir bilgi elde edemezler. Elektronik ortam, yalnızca bilgilerinin iletilme isteğini merkezi optik ağa bildirir ve merkezi optik ağ da bu isteği karşılamak üzere yol atama işlemlerini kendi yapar [6].

Mimarinin kullanımı birinci modelde olduğu gibi ağlara ilişkin soyutlama yapılmadan da kullanılabilir (ikinci model). Bu yapıda, elektronik ve optik ortamdaki tüm düğümler birbiri ile haberleşebilir. Bu sayede bir düğümle haberleşmek isteyen bir düğüm, hangi yolları kullanarak haberleşeceği düğüme ulaşacağına kendi karar verebilir [6].

IP – Optik ağ bütünleşmesinde, bu modellerden yalnız bir tanesine destek verecek değişik protokoller olması yerine, her iki modele de destek verecek bir protokol kümesi oluşturulması düşünülmüştür [6]. Bu şekilde, istenilen model istenildiği zaman kullanılabilecektir. Her iki modele de destek veren protokol kümesinin geliştirilmesi yine GMPLS çalışması çerçevesinde gerçekleştirilmektedir.

3. ÇOKLU AKTARIM TEMELLERİ

İnternetin ve bilişim teknolojilerinin gelişmesi ile birlikte, internet üzerinde birden çok kullanıcının aynı anda birbirleri ile haberleşmelerini gerektiren birçok uygulama geliştirilmeye başlanmıştır [1]. Bu durum, yavaş yavaş, bir kaynak ile bir varış arasında gerçekleştirilen *tekli iletişim* dışında bir iletişim biçimine ihtiyaç duyulmasına neden olmuştur. Bu şekilde, çok kullanıcıyı içine alan veri alış-verişine *grup iletişimi* adı verilmektedir. Grup iletişimi, günümüzde kullanılmakla beraber henüz çok yaygınlaşmış değildir; ancak gelecekte çok önem kazanacağı düşünülmektedir [1].

Grup iletişiminin özel bir çeşidi olan *çoklu aktarım*, tek kaynaktan bir grup alıcıya yapılan veri aktarımıdır.(Genellikle çoklu aktarım n kaynak ile m varış arasında gerçekleştirilen iletişimi ifade etmek için de kullanılmaktadır. Bu çalışmada ise, çoklu aktarım ifadesi, tek kaynak ile birçok varış arasındaki iletişimi anlatmak üzere kullanılacaktır.)

Çoklu aktarım, üzerinde oldukça çok çalışılan konulardan biridir [1,3]. Çoklu aktarımın, özellikle diğer iletim biçimlerinden ayırdedilerek üzerinde durulması, bu yöntemin birçok kullanım alanının olmasından kaynaklanmaktadır. Yazılım güncellemeleri, uzaktan eğitim, telekonferans uygulamaları bu kullanım alanlarından birkaçı olarak gösterilebilir.

3.1 Çoklu Aktarım ile Tekli İletişimin Farklılıkları

Çoklu aktarımın, sadece kavramsal olarak tekli iletişimden farklı olduğu düşünülebilir; çünkü bu aktarım yöntemini bir kaynağın, tekli iletişim yöntemini kullanarak, varışların her birine aktaracağı paketleri ayrı ayrı iletmesi yoluyla gerçeklemek mümkündür. Bu gerçekleme, herhangi yeni bir algoritma geliştirilmesi gerekmediği için çok kolaydır; ancak çok verimsizdir [1,12-17]. Verimsiz olmasının iki sebebi vardır: Bunlardan birincisi ve en önemlisi, kaynak ile varış düğümleri arasında paketlerin izlediği yolların ortak linkleri olması halinde, aynı link üzerinden aynı paketin birden fazla kere gönderilmesi durumunun ortaya çıkmasıdır. Bu

durum, gereksiz yere bantgenişliği kullanması anlamına gelmektedir. İkinci neden ise [1,13], kaynağın aynı paketi her varış düğümü için tekrar göndermesi gerekliliğidir. Bu durum da kaynağın bir tek paket yollamak için gereğinden çok meşgul edilmesine neden olmaktadır. Bu nedenlerden ötürü, çoklu aktarımı farklı şekillerde gerçekleme ihtiyacı duyulmuştur.

Çoklu aktarım için önerilen alternatif yöntemler, tekli aktarımın kullanıldığı gerçeklemedeki olumsuzlukları ortadan kaldırmaya çalışırlar. Bunun için, ağ üzerinde, kaynağın kök, varış düğümlerinin ise yaprak olduğu bir ağaç yapısı oluşturulur. Bu ağaç, kaynaktan çıkan paketlerin izleyecekleri yolları belirler ve çoklu aktarım ağacı adını alır. (Bu noktadan itibaren çoklu aktarım ağacı ifadesi yerine ÇAA kısaltması kullanılacaktır.) ÇAA üzerinde, düğümler, kendilerine gelen paketleri çoğullayarak, ağaçtaki çocuklarının hepsine yollarlar. Bu şekilde, kaynaktan yollanılan bir paket, ağaç üzerindeki her düğüme (ağacın kökü yani kaynak hariç) bir tek kez gider. Bu sayede kaynaktan gönderilen bir paketin ağaç üzerindeki her link üzerinden bir tek kez iletilmesi sağlanır. Ayrıca kaynağın üyelere gidecek tüm paketleri oluşturması gerekmez, bu işlem ara düğümlere dağıtılmış olur.

Görüldüğü gibi, çoklu aktarımda iki önemli kavram ortaya çıkmaktadır. Bunlardan biri, kaynaktan üyelere ulaşan ÇAA, diğeri de, ÇAA üzerindeki düğümlerin kendilerine gelen paketleri çoğullamasıdır. Bu iki yeni kavramın gerçeklenebilmesi için ağ üzerindeki düğümlerin tekli iletişimdekinden farklı başka özelliklere de sahip olması gerekmektedir. Bunlardan biri, alınan bir paketin çoğullanabilmesi, diğeri ise ağaç üzerinde olan düğümün, ağaç üzerindeki çocuk düğümlerini (çoğulladığı paketleri göndereceği düğümler) bilmesi ve bunu bir yerde saklamasıdır. Ayrıca, bir düğüm birden fazla ÇAA'nın üzerinde ise, düğümün bir ağaç ile ilgili tuttuğu çocuk düğüm bilgilerinin hangi ağaca ait olduğunu da belirleyebilmesi gerekir.

3.2 Çoklu Aktarım Grubu ve Çoğa Aktarım Adresi

Çoklu aktarımda, kaynağın gönderdiği veriyi alacak olan kullanıcılara *çoklu aktarım üyesi*, üyelerinin oluşturduğu kullanıcı topluluğuna da *çoklu aktarım grubu* adı verilmektedir. Üyelerin hepsinin bir tek topluluk olarak düşünülmesinin temel sebebi, çoklu aktarımda kaynağın paketlerini üyelere yollarken bu üyelerin tamamına atanmış bir *çoklu aktarım adresi* kullanmasıdır [14]. Eğer kaynak düğüm paketleri bu şekilde tek bir adrese göndermiyor olsa idi, paketlerin üyelerin hepsine ayrı ayrı ulaşması için paket başlığına bütün üyelerin IP numaralarını yazmak durumunda

kalırdı. Çok fazla üye olması durumunda da paket başlığının paket yüküne oranla çok büyük olma durumu ortaya çıkardı. Bunun yanında, bu çözümün uygulanabilmesi için kaynağın üyelerin hepsinin IP adreslerinin kaynak tarafından bilinmesi gerekecekti .

İşte bu olumsuzlukları gidermek için çoklu aktarım üyelerinin oluşturduğu grup için tek bir IP adresi atanması yoluna gidilmiştir. Bu adres D sınıfından bir IP adresidir ve *çoklu aktarım adresi* adını alır. Kaynak, paketlerini üyelerine bu şekilde seçilmiş olan bir adrese gönderir. Kaynağın gönderdiği bu paketler, çoklu aktarım grubunu karakterize eden çoklu aktarım adresi sayesinde çoklu aktarım üyelerine iletilecek şekilde yönlendirilirler.

İnternette kullanılan çoklu aktarım modelinde kaynak ile üye düğümler birbirlerinden haberdar değildir [14,15,18]. Zaten böyle olması daha uygundur; çünkü her an gruba bir üyenin katılması veya gruptan bir üyenin çıkması mümkündür. Her katılan ve ayrılan üye için bütün gruptaki üyelerin ve kaynağın haberdar edilmesi gerekmesi ağ üzerindeki iletişim yükünü arttırıcı bir etken olurdu. Bu durum da, paketlerin üyelere çoklu aktarım adresi kullanılarak gönderilmesinin sebeplerinden biri sayılabilir.

3.3 Çoklu Aktarım Protokolleri

Çoklu aktarımın gerçeklendiği ilk zamanlarda, çoklu aktarım, iki tür protokolün beraber kullanılması ile gerçekleştirilmiştir [15,16]. Bu protokol türlerinden ilki, uç kullanıcıların çoklu aktarım gruplarına üye olması ve bu gruplardan ayrılması için, uç kullanıcılar (konak) ile yönlendiriciler arasındaki mesajlaşmayı içerir. Bu protokol türünün bir örneği IGMP (Internet Grup Yönetim Protokolü – Internet Group Management Protocol) protokolüdür [14-16]. İkinci protokol türü ise [16], çoklu aktarım paketlerin uç kullanıcılara eriştirilebilmesi için üzerinden geçecekleri yönlendiricilerin seçilmesi için, bir başka deyişle kaynak ile üye düğümler arasında yönlendiricilerden oluşan ağacın oluşturulması için kullanılır. Bu protokol türünün bazı örnekleri olarak DVMRP (Uzaklık Vektörlü Çoklu Aktarım Protokolü - Distance Vector Multicast Routing Protocol), MOSPF (OSPF için Çoklu Aktarım Eklentileri – Multicast Extensions to OSPF), CBT (Çekirdek Tabanlı Ağaçlar- Core-Based Trees) ve PIM (Protokolden Bağımsız Çoklu Aktarım – Protocok Independent Multicast) verilebilir.

Çoklu aktarımın yaygınlaşması ile, çoklu aktarımın gerçekleştirilebilmesi için iki protokol türünün yeterli olmadığı görülmeye başlanmıştır[15,16]. Bunun asıl sebebi,

ÇAA oluşturulmasını sağlayan protokollerin çoklu aktarımın destekleyen yönlendirici sayısı artışıyla birlikte veriminin çok düşmesidir. Bu protokoller, ilk uygulamaya konulduklarında zamanlarda, çoklu aktarım yeteneğine sahip yönlendiricilerin hepsinin aynı özelliklere sahip olduğu (yani aynı ÇAA oluşturma protokolünü kullandıklarını) ve birbirinden ayırdedilemediği varsayılmıştır. (Yani internet o zamanlarda, çoklu aktarım açısından, tek tip yönlendiricilerden oluşmuş bir ağ olarak modellenmiştir.) Ancak, zamanla, hem çoklu aktarım gerçekleştiren düğüm sayısının artması, hem de, çoklu aktarım protokollerinin sayısının artması ile birlikte, tüm yönlendiricilerin aynı ÇAA oluşturma algoritmalarını kullanmamaları nedeniyle bu yaklaşımın aslında çok geçerli olmadığı anlaşılmıştır. Bu nedenle, internet üzerindeki çoklu aktarım yeteneği olan yönlendiricilerin, tekli aktarımda olduğu gibi otonom sistemler şeklinde gruplandırılması uygun görülmüştür. Böylece, daha önce kullanılmakta olan ÇAA oluşturma protokollerinin bir otonom sistem içerisinde kullanılabilmesi mümkün hale gelmiş ve kullanımdan kaldırmasına gerek kalmamıştır. Tabii, bu durumda, çoklu aktarımın tüm internet üzerinde gerçeklenebilmesi için ek olarak otonom sistemler arasında çoklu aktarımın desteklenmesi gerekmektedir. Bu ihtiyaç nedeniyle de üçüncü bir protokol türü ortaya çıkmıştır. Bu protokol tiplerinin bir örneği de BGMP'dir (Sınır Köprüleri Çoklu Aktarım Protokolü - Border Gateway Multicast Protocol) [1,15,16].

Üçüncü bir protokol türünün ortaya çıkması ile birlikte eskiden kullanılmakta olan çoklu aktarım oluşturma protokolleri, bir tek otonom sistem içinde kullanıldıkları için, *bölge-içi* (ingilizce adıyla intra-domain) çoklu aktarım protokolleri, otonom sistemler arası protokoller ise *bölgelerarası* (ingilizce adıyla inter-domain) çoklu aktarım oluşturma protokolleri olarak adlandırılmışlardır.

3.3.1 Internet Grup Yönetim Protokolü (Internet Group Management Protocol- IGMP)

IGMP birinci tip çoklu aktarım protokollerindendir [14,16]. Bu protokol, yalnızca belli bir konak grubuna hizmet veren bir yönlendiricinin hangi çoklu aktarım gruplarına hizmet vermesi gerektiğini öğrenmesi için kullanılır. Bunun için IGMP'de üç adet mesaj kullanılmaktadır. Bunlar, *üyelik sorgusu*, *üyelik raporu* ve *gruptan ayrılma* mesajlarıdır. Üyelik Sorgusu mesajı, bir yönlendiricinin hizmet verdiği konaklara, hangi çoklu aktarım gruplarına dahil olduklarını sormak üzere gönderilir. Yönlendiriciler, bu mesajlarla, ayrıca, konakların belli bir çoklu aktarım grubuna dahil olup olmadığını da sorgulayabilirler. Üyelik Sorgusu mesajını alan konaklar, cevap olarak yönlendiriciye, üyesi bulundukları çoklu aktarım gruplarının çoklu

aktarım adreslerini Üyelik Raporu mesajları ile gönderirler. Gruptan Ayrılma mesajı ise, bir konak tarafından hizmet aldığı yönlendiriciye bir çoklu aktarım grubundan ayrılmak istediğini belirtmek için gönderilir. Ancak, Gruptan Ayrılma mesajının kullanılması zorunlu değildir; çünkü yönlendiriciler periyodik olarak Üyelik Sorgusu mesajları gönderirler ve konaklar daha önce üyesi oldukları çoklu aktarım grubundan ayrılmışlar ise yapılan sorguya yanıt vermeyecekleri için, bu konakların gruptan ayrıldığı anlaşılmış olur.

Bir yönlendiricinin belli bir çoklu aktarım grubuna hizmet vermesi için, hizmet verdiği konak grubu içinde en az bir konağın bu gruba üye olması gerekmektedir. Birden fazla konağın da üye olması durumunda da, yönlendirici açısından bir şey fark etmeyecek ve yine çoklu aktarım grubuna hizmet verilmesi gerekecektir. Buna göre, bir yönlendirici hizmet verdiği konaklara Üyelik Sorgusu mesajı gönderdiğinde, konaklar içinden aynı çoklu aktarım grubuna ait konakların yalnız bir tanesinden Üyelik Raporu mesajı alması yeterlidir. Bu nedenle konakların Üyelik Sorgusu mesajlarına şu şekilde yanıt vermesi sağlanmıştır [14] : Üyelik Sorgusu alan ve bir çoklu aktarım grubuna üye olan bir konak bir Üyelik Raporu göndermeden önce rastgele bir süre bekler. Bu süre içinde, ait olduğu gruba ilişkin bir Üyelik Raporu yönlendiriciye gönderilirse, kendi Üyelik Raporu mesajını göndermekten vazgeçer (yani bir çeşit üstel geri çekilme mekanizması kullanılır.). Bu şekilde, gereksiz Üyelik Raporu mesajı iletimleri önlenmiş olur ve bir yönlendirici, gönderdiği bir Üyelik Sorgusu mesajına karşılık, her bir çoklu aktarım grubu için bir tek Üyelik Raporu mesajı alır. Eğer daha önceden belli bir grup için Üyelik Raporu almasına rağmen, son sorgu sonrasında böyle bir rapor alamazsa, yönlendirici, bu grup için artık hizmet vermesine ihtiyacı olmadığını anlayacaktır.

3.3.2 Bölge-içi Çoklu Aktarım

Tekli aktarımda bir kaynak ve bir varış düğümü olduğundan bu iki düğüm arasında bir yol bulunmaya çalışılır. Çoklu aktarımda ise, birden çok varış olduğundan kaynak ile varış arasındaki bağlılık kaynak ve varışların (yani üyelerin) içinde bulunduğu bir ağaç ile sağlanır. Dolayısıyla, ağ üzerinde bir ağaç oluşturulmaya çalışılmalıdır. Ancak ağaç oluşturma işlemi karmaşıktır; çünkü çoklu aktarımda tüm üyeler tek bir çoklu aktarım adresi ile ifade edilirler ve üyelerin hepsinin adresleri ayrı ayrı bilinmez. Bu neden ile ağaç oluşturulmaya başlandığında kaynak hiçbir üye düğümün adresini bilmez, hiçbir üye düğüm de, diğer üye düğümlerin ve kaynağın adresini bilmez. Dolayısıyla ağaç hangi düğüm(ler)den başlanarak oluşturulmaya

başlanırsa başlansın ağaca bağlanması gereken düğümlerin bir şekilde aranması gerekir.

Arama işlemi de göz önünde bulundurulduğunda, ağaç oluşturma işlemi için üç farklı yaklaşım mevcuttur. Birinci yaklaşım, kaynağın göndereceği paketin tüm bölge-içi yönlendiricilere ulaştırılmasıdır. Yani arama yapmak için, taşma yöntemi kullanılır. Taşma yöntemi ile paketi alan yönlendiricilerden, gruba hizmet edecek olanlar paketleri almaya devam ederken üye olmayanların paket almaya devam edip etmeyecekleri kullanılan yöntemlere bağlı olarak belirlenir. Taşma yöntemi ile üyelerin yerinin tespit edilmesini sağlayan bu yaklaşımda, çoklu aktarım kaynağının kök olduğu bir ağaç oluşturulduğu için bu yaklaşıma *kaynak tabanlı ağaç oluşturma* adı da verilmektedir.

Kaynak tabanlı ağaç oluşturma yaklaşımı [1,13,14] , taşma yöntemi ile paketleri bölge-içi yönlendiricilerin birçoğuna ulaştırdığı için, çok sayıda yönlendiricinin meşgul edilmesine neden olan bir yaklaşımdır. Otonom sistem içinde, çoklu aktarım grubuna üye olan konak sayısı az olsa bile, yönlendiricilerin çoğu gereksiz şekilde meşgul edilmektedir. Bu nedenle, bu yaklaşım ancak otonom sistem içinde üye sayısının çok olması durumunda verimli hale gelmektedir. Bu yaklaşımı kullanan bölge-içi çoklu aktarım protokollerine *yoğun kip* protokolleri de denilmesinin nedeni budur.

Bölge-içi ÇAA oluşturma yaklaşımlarından ikincisi *paylaşılan ağaç oluşturma* yaklaşımıdır. Bu yaklaşımda [1,13,14], kaynağın ve üye düğümlerin birbirini araması işlemi ortadan kaldırılmıştır. Bunu gerçekleştirmek için ağacın kökü çoklu aktarım kaynağı değil, otonom sistem içinde başka bir düğüm olarak seçilir. Seçilen bu düğüme *merkez düğüm* ya da *buluşma noktası* gibi adlar verilmektedir. Bu düğümün çoklu aktarım grubu açısından özelliği, çoklu aktarıma dahil olan tüm düğümlerin (hem kaynak hem de üye düğümlerin) bu düğümü bilmesidir. Bu sayede, kaynak ve üye düğümler birbirlerinin yerini bulmak yerine, kendilerinin ağaca dahil olmaları için doğrudan merkez düğüm ile iletişim kurarlar. Bunun için, bütün üyeler merkez düğüme ağaca katılmak istediklerine dair KATILMA mesajları gönderirler. Eğer mesajlara olumlu yanıt gelirse KATILMA mesajlarının merkeze giderken üzerinden geçtikleri yollar, üyeleri merkeze bağlayan yollar olarak belirlenir. Bu şekilde bir ağaç oluşturulmuş olur. Bu ağaç üzerinde iletişim, genel olarak, kaynaklardan gelen verilerin merkeze iletilmesi, merkeze gelen verilerin de merkezden tüm üyelere dağıtılması biçiminde olur.

Paylaşılan ağaç oluşturma yöntemlerinde, ağ üzerinde çoklu aktarıma katılacak çok sayıda düğüm varsa, tüm trafiğin merkez düğüme doğru gidip oradan üye düğümlere dağıtılması, merkez düğüme gelen ve merkez düğümden çıkan trafiğin yoğunlaşmasına, yani tıkanıklıkların oluşmasına neden olur. Bu yüzden, bu yöntemlerin az sayıda üyenin katıldığı çoklu aktarım gruplarında kullanılması verimlidir ve paylaşılan grup ağaçları oluşturma yaklaşımını kullanan protokollere bu nedenle *seyrek kip* protokolleri adı da verilmektedir.

Üçüncü ve son yaklaşımda ise çoklu aktarım paketleri taşma yöntemi ile tüm ağa dağıtılmaz. Bunun yerine, yönlendiriciler arasında iletilen bağlantı durumu bilgileri kullanılır (tabii bu durumda, bu yaklaşım, ancak bağlantı durumlarının yayınlanabildiği ağlar üzerinde uygulanabilir). Bu nedenle, bu yaklaşımı kullanan yöntemlere *bağlantı durumlu* protokoller adı verilebilir [13]. Bu yaklaşımda, bağlantı durumu bilgileri içine çoklu aktarım üyelerinin ağın hangi yönlendiricilerinde olduğu bilgisi eklenir. (Bu şekilde, yönlendiricilerin yalnızca üyelik bilgileri taşma yöntemi ile ağa dağıtılmış olur.) Bütün yönlendiriciler diğer yönlendiricilerin bağlantı durumu bilgilerini aldıkları için, her yönlendirici ağın topolojisini bilir. Bu durumda, her yönlendirici çoklu aktarım kaynağı ile üyeleri arasındaki ağacı kendisi oluşturabilir. Tüm yönlendiriciler aynı bağlantı durumu bilgilerini aldıkları için, hepsinin aynı kaynak ve aynı çoklu aktarım üyeleri için oluşturacakları ağaçlar aynı olacaktır. Bu sayede, her yönlendirici bir çoklu aktarım grubu için oluşturduğu ağaca bakarak kendisine gelen bir çoklu aktarım paketinin hangi yönlendiricilere gönderileceğini belirler ve çoklu aktarım paketlerinin ağ üzerinde yönlendirilmesi bu şekilde sağlanmış olur.

Kaynak Tabanlı Ağaç yöntemlerinde, çoklu aktarım grubu içindeki her bir kaynak için farklı bir ağaç oluşturulur. Dolayısıyla bir çoklu aktarım grubu içinde N adet kaynak varsa N faklı ağaç oluşturulacaktır. Oluşturulan bu ağaçların hepsi de birer ÇAA 'dır ve bu ağaçlarda linkler tek yönde, yani kaynaktan üye düğümlere doğru, trafik taşırlar. (Bu tür ağaçlara tek yönlü ağaçlar denir.) *Paylaşılan Ağaç* yönteminde ise tüm grup üyelerinin kullanacağı tek bir ağaç oluşturulur [14]. Birden fazla kaynak olabileceği için bu ağacın linkleri çift yönlü trafik taşıyabilir, yani bu ağaç çift yönlü bir ağaçtır.

3.3.2.1 Kaynak Tabanlı ÇAA Oluşturma Yöntemleri

Daha önce de değinildiği gibi, kaynak tabanlı ÇAA oluşturma yöntemleri, ağaç oluşturabilmek için çoklu aktarım paketlerinin ağa dağıtılmasını sağlayan, taşma

veya taşma ile aynı ilkeyi kullanan bir yönteme ihtiyaç duyar. Bu nedenle aşağıda taşma yönteminden başlanarak, paketlerin ağ üzerine dağıtılmasını sağlayacak yöntemler incelenmiştir.

Taşma (Flooding)

Bu yöntemde [12], bir çoklu aktarım paketini alan düğüm, paketi aldığı komşusu hariç diğer bütün komşularına yollar; ancak bu işlemi yalnızca paketi ilk kez alıyorsa yapar. Bu şekilde, bir çoklu aktarım paketi ağ üzerindeki tüm düğümlere ulaştırılmış olur. Çoklu aktarım üyeleri paketi kendileri de kullanırken, diğer düğümler yalnızca paketin ağ üzerinde dağıtılmasını sağlarlar. Burada asıl sorun, paketin ilk kez gelip gelmediğini belirlemektir. Bunun için çeşitli yöntemler kullanılabilir. Bu yöntemlerden bir tanesi, çoklu aktarım paketinin beraberinde bağlantı durumu bilgisi taşımasıdır. Bir paket alan yönlendirici, paket üzerindeki bu bilginin kendi taşıdığı bilgiden taze olup olmadığına bakarak paketi ilk defa alıp almadığına karar verir. (Bu yöntemin uygulanabilmesi için düğümlerde OSPF (Open Shortest Path First) gibi bağlantı durumları kullanarak yönlendirme yapılıyor olması gerekir.) Başka bir yöntem de, paketin hangi düğümler tarafından alındığı bilgisini içinde taşımasıdır . Böylece yönlendirici, paket içindeki bu bilgiyi inceleyerek paketi ilk defa alıp almadığını anlar. Paketin içine yerleştirilen bu ek bilgi, ağın boyutuna bağlı olarak oldukça büyük boyutlara varabilir.

Paketlerin ilk defa alınıp alınmamasının kontrolü dışında ayrıca paketlerin ağ içinde aynı paketin sayısının çok artmaması için, içlerine bir TTL (yaşam süresi – Time To Live) alanı konulur ve TTL = 0 olduğunda da bu paket artık başka yönlendiriciye iletilmez.

Taşma yöntemi, paketlerin tüm ağ üzerine dağıtılmasını sağladığı için dayanıklı, sağlam ve güvenilir bir yöntemdir. Ayrıca uygulaması çok kolaydır ve yönlendirme tablosu gerektirmez. Buna karşılık, bir çoklu aktarım paketi ağ üzerindeki tüm linkler üzerinde birçok kez gönderileceği için gereksiz yere çok fazla ağ kaynağı harcanmasına neden olur. Ayrıca bu yöntemle gruba özel paket yollanamaz ve bu nedenle güvenli bir iletişim yapmaya uygun değildir. Bu önemli dezavantajına rağmen bu yöntem ağ üzerindeki iletim hatalarının belirlenmesi veya düğümler arasındaki en kısa yolların bulunması gibi test amacına yönelik durumlarda kullanılabilir.

Kapsayan Ağaç yöntemi (Spaning Tree)

Bu yöntem [12], ağ üzerinde tüm düğümleri içeren bir kapsayan ağaç oluşturularak, aktarımların yalnız bu ağaç üzerinde yapılmasına dayanır (Kapsayan ağaç, ağdaki her düğümden bir diğerine yolun bulunabildiği ve çevrim içermeyen bir graftır.). Bir çoklu aktarım paketi, bir düğüme geldiğinde, paket, geldiği komşu hariç, düğümün kapsayan ağaç üzerindeki tüm komşularına iletilir. Ağaç, tüm düğümleri kapsadığı için paketin hedefine varacağından emin olunur.

Kapsayan ağaç yöntemi, bir düğümün sadece kendisine bağlı linklerinin kapsayan ağaç üzerinde olup olmadığının belirlenmesini gerektirdiği için gerçeklenmesi oldukça kolay bir yöntemdir. Buna karşılık, belli bir kaynak ve üyeler grubu için, kullanılan kapsayan ağaç en hızlı şekilde iletimin gerçekleşmesini sağlayacakken, başka bir kaynak ve üyeler grubu için yavaş bir iletime neden olabilir [12]. Ayrıca, tüm kaynak ve üyeler grupları için hep aynı kapsayan ağaç üzerindeki linkler kullanılacağı için, bu linklerde trafik yoğunluğu oluşacaktır. Bunun haricinde, bu yöntemde taşma yönteminde olduğu gibi çoklu aktarım paketleri ağdaki tüm düğümlere ulaştırıldığı için grup içinde güvenli iletişim yapılması mümkün değildir.

Kaynak Yoluna Göre Sektirme (Reverse Path Forwarding - RPF)

Bu yöntemde kullanılan temel ilke şudur [12,14-16]: Bir düğüme bir çoklu aktarım paketi geldiğinde, düğüm paketin, kendisi ile çoklu aktarım kaynağı arasındaki en kısa yol üzerinden (daha doğrusu en kısa yol üzerindeki komşusundan) gelip gelmediğini inceler. Eğer paket en kısa yol üzerinden gelmişse, paket, alındığı komşu dışındaki bütün komşulara iletilir, aksi taktirde çöpe atılır.

Bu ilke kullanılarak taşma yönteminin sorunlarından biri olan paketin bir tek defa alınmasının sağlanması, basit ve fazla bellek gerektirmeyen bir şekilde çözülmüş olur. Ancak, kaynaktan bakıldığında, kaynak ile çoklu aktarım paketini alan yönlendirici arasındaki en kısa yol, yönlendiriciden bakıldığında, yönlendiriciden kaynağa olan en kısa yol ile aynı değilse, paketlerin varışlara giderken izleyecekleri yollar beklenildiği gibi en kısa yollar olmayacaktır [12].

Daha verimli olabilecek bir yöntemde de, bir çoklu aktarım paketi alan yönlendirici, kendisinin komşuları ile kaynak arasındaki en kısa yol üzerinde olup olmadığına bakar. Eğer en kısa yol üzerinde değilse paketi o komşularına yönlendirmez [12]. Bu yönteme geliştirilmiş RPF yöntemi adı vermek mümkündür.

RPF yönteminde, yönlendirme yapmak içim tekli iletişim için gerekli yönlendirme tabloları dışında bir bilginin saklanmasına gerek yoktur. Ayrıca RPF her çoklu aktarım kaynağı için farklı bir ÇAA oluşturur. Bu sayede, paketler, aynı ağaçlar üzerinden gönderilmeyeceğinden, ağ üzerinde farklı linkler üzerinden geçerek ağın daha verimli kullanılması sağlanmış olur. RPF yönteminin bir başka avantajı ise, üyelere en kısa süre varılması sağlanmasıdır; çünkü yöntemin çalışması gereği yönlendirme yalnızca en kısa yol üzerindeki düğümlerde yapılır [12]. Bu avantajlarının yanında RPF'nin en büyük dezavantajı, paketleri iletimi sırasında, yönlendiricilerin bağlı olduğu konakların gruba üye olup olmadığına bakılmamasıdır [8,14]; çünkü bu durum taşma yönteminde olduğu gibi çoklu aktarım paketlerinin ağ üzerindeki tüm düğümlere gönderilmesine neden olacaktır. RPF 'in başka bir dezavantajı da yönlendiricilerin hangi kapılarından hangi kaynak ve grup çiftleri ile ilgili paketleri göndermeleri gerekeceğini saklaması gerekliliğidir [12]. Bu durum, aynı grup içinde birden çok kaynak olduğunda, bir grup için oldukça çok bilgi saklanmasını gerektirebilir. Bu nedenle çok sayıda çoklu aktarım grubunun varlığı halinde yöntem oldukça çok bellek ihtiyacına neden olabilir.

RFP ve Budama

RFP'in dezavantajlarından birinin, çoklu aktarım paketlerinin tüm ağdaki yönlendiricilere gönderilmesi olduğu söylenmişti Bu durumun üstesinden gelmek üzere için *budama* mesajlarının kullanımı ortaya atılmıştır [12,14]. Budalama mesajlarının RPF ile beraber kullanımı şu şekildedir :

Bir çoklu aktarım paketi, RPF' in çalışması gereği kaynaktan tüm ağa gönderilir. Yaprak düğümlere kadar gelen paketi, çoklu aktarım grubuna üye düğümlere hizmet veren yönlendiriciler alır, grup üyelerine hizmet vermeyenler ise paketi aldıkları kapıdan, kendilerine çoklu aktarım mesajı gönderen yönlendiriciye, "Mesajın ait olduğu S kaynaklı G grubunun diğer mesajlarını bana bir daha gönderme" anlamını taşıyan budama mesajını gönderirler. Yönlendirici bu mesajı aldıktan sonra, bu yaprağa bir daha (S,G) çiftine ait bir mesajı göndermez. Eğer yönlendirici, paketi yönlendirdiği tüm düğümlerden budama mesajı almışsa ve kendiside herhangi bir grup üyesine hizmet vermiyorsa, bu durumda kendisine çoklu aktarım paketi gönderen düğüme budama mesajı gönderir.

Çoklu aktarım başladıktan sonra, grup üyelerinin zaman içinde değişmesi söz konusu olabileceğinden, daha önce budama mesajı göndermiş yönlendiricilerin çoklu aktarım ağacına dahil edilmesi gerekebilir. Bunun için, farklı yöntemler

kullanılabilir [14]. Bunlardan bir tanesi budama mesajı gönderen yönlendiricilerin AĞACA_BAĞLAN anlamında bir mesajı, kaynağa doğru en kısa yol üzerinden göndermesidir. İkinci yöntem ise budama mesajlarının belli bir zaman aşımı sonunda geçersiz hale gelmesidir. Böylece budanmış olan bir düğümler ağaca tekrar eklenmiş olur. Eğer ağaca yeniden eklenen düğüm, hala bir gruba hizmet vermiyorsa yine budama mesajı yollayarak ağaçtan çıkarılabilir.

RPF ile birlikte budama mesajlarının kullanılması sayesinde, yalnızca kaynaktan grup üyelerine aktarımın yapılacağı bağlantılar aktarım için kullanılır. Böylece, ağ kaynaklarının daha verimli kullanılması sağlanmış olur [12].

3.3.2.2 Bölge-içi Çoklu Aktarım Protokolleri

Uzaklık Vektörlü Çoklu Aktarım Protokolü (Distance Vector Multicast Routing Protocol - DVMRP)

DVMRP, internet üzerinde ilk kullanılmış olan çoklu aktarım protokolüdür . DVMRP basit bir uzaklık vektör yönlendirme protokolüdür ve RIP'e benzer. Çoklu aktarım yapabilen yönlendiriciler, RIP'te olduğu gibi birbirlerine uzaklık vektör bilgilerini gönderirler .Uzaklık vektör bilgilerinin içinde, varış ve kaynak düğümlerin ve bunlara ilişkin uzaklıkların (sekme sayısı cinsinden) bir listesi vardır [12].

DVMRP [14,15], ÇAA oluşturulmak üzere geliştirilmiş RPF yöntemini budama mesajları ile beraber kullanır. Budanmış linklerin tekrar ağaca eklenmesi için hem zaman aşımı yaklaşımı hem de AĞACA_BAĞLAN mesajları kullanır. AĞACA_BAĞLAN mesajları ağaca katılmak isteyen bir düğümün zaman aşımını beklemeden ağaca eklenmesini sağlar. Bunun dışında DVMRP, diğer çoklu aktarım protokolleri gibi, tünelleme yöntemini kullanarak çoklu aktarım paketi aktarma yeteneğine de sahiptir. Tünelleme [12,14], internet üzerindeki tüm yönlendiricilerin çoklu aktarımı desteklememelerinden dolayı çoklu aktarımı destekleyen düğümler arasında çoklu aktarım paketlerinin yollanabilmesine olanak sağlamak için kullanılır.

OSPF için Çoklu Aktarım Eklentileri (Multicast Extensions to OSPF – MOSPF)

MOSPF [1,14,15], OSPF protokolünün kullanıldığı otonom sistemlerde çoklu aktarımı desteklemek için geliştirmiştir. Bu protokol bağlantı durumlu bir protokoldür ve bu nedenle kaynak ile üyeler arasındaki ağaçlar her düğümde oluşturularak, bir düğüme gelen bir çoklu aktarım paketin iletileceği bir sonraki düğüm tarafından bulunan bu ağaç kullanılarak belirlenir.

Çekirdek Tabanlı Ağaçlar (Core-Based Trees - CBT)

CBT [1,14,15], paylaşılan ağaç yaklaşımı ile ÇAA oluşturan bir protokoldür. Protokolün oluşturduğu ağaç çift yönlüdür ve tek bir merkez (çekirdek) düğüm içerir. Her çoklu aktarım grubu için merkez düğümün seçimi yapıldıktan sonra, grup üyelerine hizmet vermesi gereken CBT yönlendiricileri BAĞLANTI_İSTEĞİ mesajlarını tekli iletişim yoluyla merkez düğüme doğru iletirler. Buna karşılık, merkez düğüm ya da BAĞLANTI_İSTEĞİ mesajını alan ve daha önceden ağaca eklenmiş olan bir yönlendirici BAĞLANTI_ONAYI mesajını BAĞLANTI_İSTEĞİ mesajını yollayan yönlendiriciye gönderir. Böylece BAĞLANTI_İSTEĞİ mesajını yollayan düğüm BAĞLANTI_ONAYI mesajını aldığı düğüme, bu mesajlaşmanın gerçekleştiği yol üzerinden bağlanmış olur. Ağacın bu şekilde oluşturulması ardından ağacın durumunun kontrolü için düğümler ağaç üzerindeki anne düğümlerine YANKI_İSTEK mesajları gönderirler. Bu mesajlar için annelerinden YANKI_YANIT mesajı aldıkları sürece ağaca bağlı kalırlar. Ancak YANKI_YANIT mesajı alamadıkları taktirde düğüm ağaçtan kopmuş olduğu kararını verir ve kendisi üzerinden ÇAA 'ya bağlı tüm düğümlere ağaçtan kopmalarını sağlayacak bir mesaj gönderilmesini sağlar.

Protokolden Bağımsız Çoklu Aktarım (Protocol Independent Multicast - PIM)

Bu protokolün [14,15] iki çalışma kipi mevcuttur. Bu çalışma kipleri yoğun kip ve seyrek kip olarak adlandırılmıştır. Protokol yoğun kipte, yoğun kip protokolleri gibi davranır, seyrek kipte de seyrek kip protokollerine benzer biçimde çalışır. Böylece ağdaki üyelerin yoğunluğuna göre en verimli çalışmayı gerçekleştirmeyi amaçlar.

PIM, adının da ifade ettiği üzere, tekli iletişim protokollerinden bağımsız olarak çalışma özelliğine sahiptir. Yani PIM, üzerinde çalıştığı düğümün tekli iletişim için kullandığı yönlendirme tablosu ne olursa olsun bu tablonun bilgilerini kullanabilir.

PIM yoğun kipinde, DVMRP ' ye benzer bir yöntem kullanır. Aralarındaki fark DVMRP'de ilerletilmiş RPF kullanılırken, PIM'de daha basit bir yöntem olan RPF kullanılmasıdır [15]. PIM seyrek kipinde ise paylaşılan ağaç tabanlı bir yöntemi kullanılır. Bu yöntem CBT 'ye oldukça benzer. Ancak PIM 'de ağaca katılmak için gönderilen KATILMA mesajlarına karşılık herhangi bir ONAY mesajı gönderilmez. Ağacın durumunu korumak ve için CBT 'de kullanılan YANKI_İSTEK mesajı da mevcut değildir, bunun yerine periyodik olarak KATILMA mesajlarının merkeze doğru gönderilmesi gerekmektedir [14]. PIM 'in bir başka özelliği de, belli bir üyenin kendisine ulaşan veri hızının çok düşmesi halinde, paylaşılan ağaçtan ayrılıp

doğrudan kaynaktan veri alabilmesinin sağlanabilmesidir. Böylece merkez düğüme giren ve çıkan trafik yoğunluğu azaltılabilmektedir. Ayrıca bir üyenin paylaşılan ağaç yönteminde kaynak ile arasında verinin gidip geldiği yol en kısa yol olmayabilir. Bu durumda da üyenin kaynaktan doğrudan veri alması daha iyi bir çözüm olabilir.

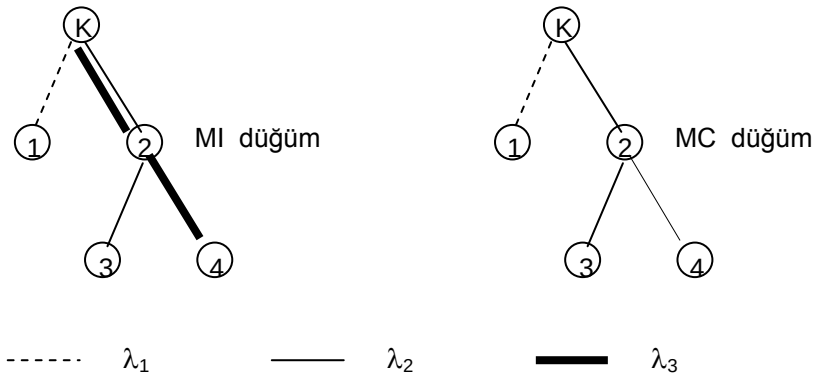
3.3.3 Bölgelerarası Çoklu Aktarım

Bölge-içi çoklu aktarımda bir otonom sistem içinde varolan ÇAA oluşturma algoritmalarını gerçekleştiren protokoller çalışmaktaydı. Ancak farklı otonom sistemlerin içinde kullanılan çoklu aktarım protokolleri farklılık gösterebilmektedir. Örneğin bir otonom sistem DVMRP kullanırken , bir diğeri de PIM protokolünü kullanabilir. Bu nedenle otonom sistemler arasında çoklu aktarımı destekleyen protokollere ihtiyaç vardır. Günümüzde bu tür protokoller henüz geliştirme aşamasındadır [15]. Bölgelerarası çoklu aktarım için kısa vadede kullanılmak üzere MSDP (Çoklu Aktarım Adres Bulma Protokolü - Multicast Source Discovery Protocol) adlı bir protokol geliştirilmiştir. Uzun vadede kullanılmak üzere ise BGMP protokolü önerilmektedir. Her iki yöntemde de seyrek-kip bölge-içi bir protokoldekine benzer bir çalışma söz konusudur. Bölgelerarası çoklu aktarımda her bir düğüm, bir otonom sistem olarak düşünülebileceğinden, bu protokollerin paketlerin bütün düğümlere gönderilememesine neden olan yoğun-kip bölge-içi protokollerine benzer bir çalışma yerine, yalnızca çoklu aktarıma katılacak olan otonom sistemleri içeren bir seyrek- kip bölge-içi çoklu aktarım protokolüne benzer çalışması daha uygun düşmektedir.

4. DALGABOYU ATAMALI WDM AĞLARDA ÇOKLU AKTARIM YÖNTEMLERİ

Optik ortamda çoklu aktarım söz konusu olduğunda, bellek ve yol atama ile ilgili sorunlar dışında ayrıca paketlerin çoğullanması da bir sorun olarak karşımıza çıkmaktadır. Optik ağlarda paketlerin çoğullanması, elektronik ya da optik ortamların herhangi birinde yapılabilir. Elektronik ortamda çoğullama, optik ve elektronik ortamlar arasında geçiş yapılmasını gerektirdiği için pek istenmeyen bir yöntemdir. Optik ortamda çoğullama ise dalgaboyu yönlendiricilerinin ışık bölme yeteneğinin olmasını gerektirir (Işık bölme ile ilgili daha ayrıntılı bilgi için 2.4.3.1 kısmındaki Işık Bölme Yeteneği alt başlığına bakınız.) Işık bölme işleminin de maliyeti oldukça yüksektir [3,5,8]. Bu nedenle, genellikle yalnız bazı düğümlerin ışık bölme yeteneğine sahip olduğu ağların kullanılması uygun görülmektedir. Bu ağlar üzerinde ışık bölme yeteneğine sahip olan düğümlerin de bir girişinden gelen ışığı diğer tüm çıkışlarına bölme yeteneğinde olması uygun görülmektedir [3,5,8].

Bu noktadan itibaren, yazım kolaylığı açısından yukarıda anlatılan anlamda ışık bölme yeteneğine sahip olan düğümlere MC (İngilizce “Multicast Capable” deyiminden) ve ışık bölme yeteneğine sahip olmayan düğümlere de MI (İngilizce “Multicast Incapable” deyiminden) düğümler olarak adlandırılacaktır.



Şekil 4.1: MI düğümlerin ÇAA sayısına etkisi.

Dalgaboyu Atamalı WDM ağlarda, tüm düğümler MC olmadığı için çoklu aktarım yapılırken, tüm düğümlere tek bir ağaç kullanılarak erişmek mümkün olmamaktadır. Bu durumda, tüm düğümlerin kapsanabilmesi için, birden fazla ağaçtan oluşan bir orman oluşturulması gerekebilir. Tabii, dalgaboyu dönüşümünün kullanılabilmesi, orman oluşturulurken dalgaboylarının daha verimli olarak kullanılmasını sağlayabilir. Şekil 4.1, düğümlerin MI olmasının neden çoklu aktarım ağaçlarının sayısının birden fazla olmasına ihtiyaç duyulduğunu göstermektedir [8].

4.1 WDM Çoklu Aktarım Yöntemlerinin Sınıflandırılması

IP 'nin, WDM ağlarla bütünleşmesi düşünüldüğünden çoklu aktarım için geliştirilen algoritmaların da bu bütünleşmeyi desteklemesi gerekmektedir. Buna göre, bir IP/WDM protokol yığını kullanılması gerekmektedir. Bu protokol yığının gerçeklenmesinde, ÇAA 'nın oluşturulması işleminin, hangi katmanda gerçekleştirileceği sorunu ortaya çıkmaktadır.

IP/WDM protokol yığınında, ÇAA 'nın üç farklı şekilde gerçeklenmesi mümkündür : i) ÇAA doğrudan IP katmanında, optik ağın fiziksel verileri kullanılarak oluşturulabilir. ii) IP katmanında var olan yöntemlerle oluşturulan ÇAA, WDM katmanında (2. katman) optik ağın fiziksel verileri değerlendirilerek değiştirilebilir. iii) ÇAA doğrudan WDM katmanında oluşturulabilir.

ÇAA'nın optik ağın fiziksel verileri kullanılarak gerçekleştirilmesi yöntemi, optik ağın fiziksel verilerinin IP katmanına aktarılmasını gerektirir. Bu bilgiler ışığında gerçekleştirilecek yol atama işlemi, günümüzde IP katmanında kullanılan yol atama işleminden daha karmaşık olacaktır. Ayrıca, bu şekilde, IP katmanındaki yol atama işlemi ile ilgili protokollerin değiştirilmesi, pratik olarak bu protokollerin internet ortamına uygulanabilmesini zorlaştırmaktadır; çünkü tüm yönlendiricilerde eski protokol yerine yenileri aynı anda koyulamamaktadır. Ayrıca, ÇAA'nın bu şekilde oluşturulması halinde, IP katmanında görülen topoloji ile WDM katmanında görülen topolojinin aynı olmaması nedeniyle tutarlılık sorunu ortaya çıkabilir [3,5].

IP katmanında var olan yöntemlerle oluşturulan ÇAA 'yı , WDM katmanına aktarıp, bu katmanda asıl ÇAA 'yı oluşturma yönteminin ilk bakışta görülen bir avantajı IP katmanındaki protokollerin değiştirilmemesidir. Bu sayede, yöntemini internet ortamında uygulanması oldukça kolaylaşacaktır. Buna karşılık, IP katmanında oluşturulan ÇAA, WDM katmanında değişikliğe uğramak zorunda kalacaktır; çünkü WDM katmanında tüm düğümlerin MC değildir. Ayrıca, IP katman topolojisi ile WDM

katman topolojisi de birbirinden farklı olabileceğinden dolayı, iki katmanda oluşturulan ağlar ÇAA ları arasında ciddi bir tutarlılık sorunu oluşacaktır [3,5].

ÇAA 'nın doğrudan WDM katmanında oluşturulması, yine IP katman protokollerinin değişmesine neden olacaktır; ancak internete uygulanırken ilk yöntem kadar sorun yaşanamayacaktır. Buna karşılık, yine, ÇAA oluşturulurken fiziksel bilgiler (dalgaboyu dönüşüm kapasitesi, düğümlerin ışık bölme yetenekleri vs.) değerlendirilmesi protokolü karmaşıklaştıracaktır. Bunun yanında, bu şekilde oluşturulacak ÇAA, doğrudan fiziksel bilgileri hesaba kattığından bant genişliğinin en verimli şekilde kullanılmasını sağlayacaktır. Tabii bu arada optik ortamda bellek olmaması nedeniyle protokolün verimli çalışması da gerekecektir [3,5].

WDM çoklu aktarım yöntemleri, ayrıca ÇAA 'nın statik veya dinamik olarak oluşturulmasına bağlı olarak da iki sınıfa ayrılabilir. Statik ÇAA oluşturulan yöntemlerde, ÇAA bir kere oluşturulduktan sonra, gruba yeni bir üyenin katılması veya gruptan bir üyenin çıkarılması söz konusu değildir. Buna karşılık, dinamik çoklu aktarım yöntemlerinde, çoklu aktarım grubuna üye olan düğümlerin değişmesine izin verilir.

Statik ÇAA oluşturma yöntemlerinde, ÇAA yalnız bir kere oluşturulduğu için bu yöntem doğrudan WDM katmanında gerçekleştirilmesi uygun bir çözüm olarak görünmektedir; çünkü bu şekilde ÇAA'nın oldukça verimli bir şekilde oluşturulması mümkündür. Buna karşılık, dinamik ÇAA oluşturma yöntemleri içinde IP katmanında ağaç oluşturulması daha avantajlıdır; çünkü grup üyelerinin sürekli değişmesi, sürekli olarak ağacın değişmesine, yani sürekli elektronik ortamda kararların verilmesine neden olacaktır. IP katmanında ağaç oluşturulurken, optik ortam bilgilerinin hesaba katılması, WDM katmanında yapılacak karar işlemleri miktarı azaltmakla beraber, IP katmanında oluşturulan ağacın WDM katmanına aktarılması ve WDM katmanında son ağacın oluşturulması, WDM katmanına çok ek yük getirmeyecek bir şekilde yapılabilirse, bu yöntem protokol yığınının internete uygulanabilirliği açısından en uygun çözümlerden biri olacaktır.

4.2 Statik WDM ÇAA Oluşturma Algoritmaları

Bu bölümde sırasıyla Üye Yönelimli (Member-Only), Üye Öncelikli (Member-First) ve Evlat Edinmeli statik WDM ÇAA oluşturma algoritmaları incelenmiştir.

4.2.1 Üye Yönelimli (Member-Only)

Bu yöntem [3], en basit statik ÇAA oluşturma yöntemlerinden biridir. Algoritma, basit bir statik ÇAA oluşturma algoritmasına dayandırılmıştır. Bu basit çoklu aktarım algoritması, ağaca eklenmiş düğümler (kaynak düğüm dahil) ve ağaca eklenmemiş grup üyelerinin oluşturduğu düğümler biçimde iki küme tanımlar. Her adımda, ağaca eklenmiş küme düğümlerinden henüz ağaca eklenmemiş olan grup üyelerine giden en kısa yollar hesaplanır. Bu kısa yollardan en kısa olanı seçilip bu yol ve yolun ulaştığı grup üyesi ağaca eklenir. Bu işlem ağaca eklenmemiş bir grup üyesi kalmayana kadar devam eder. Bu algoritma Şekil 4.2'de gösterilmiştir [3].

```
V = {kaynak & ağaca eklenmiş varış düğümleri }
UV = { Henüz ağaca eklenmemiş varış düğümleri}
while ( UV boş değil )
{
   Bir x ∈ UV ile y ∈ V arasında yol bulunan (x,y) çiftleri birbirlerine en yakın
  olanını seç;

   x ' i UV 'den çıkarıp V 'ye koy ;
}
```

Şekil 4.2: Basit bir ÇAA oluşturma Algoritması

```
V = { kaynak & ağaca eklenmiş varış düğümleri }
V' = { Tekrar kullanılamayacak düğümler. }
UV = { Henüz ağaca eklenmemiş varış düğümleri }
while ( UV boş değil )
{
   if ( Bir x ∈ UV ile y ∈ V arasında V' 'den bir düğüm bulunmayan yol yok )
            UV'de kalan düğümlerle yeni bir ağaç daha oluşturmak
           İçin algoritmayı tekrar yürüt (V ve V' 'nü sıfırla.) ;

   Bir x ∈ UV ile y ∈ V arasında yol bulunan (x,y) çiftleri içerisinden x ile y arasında
   V' 'den bir düğüm bulunmayanı ve birbirlerine en yakın olanını seç ;

   x ' i UV 'den çıkarıp V 'ye koy ;
   if ( y MI düğüm )
       y → V' ;

    for( x ile y arasındaki ara düğüm z 'ler )
         if( z MI düğüm ) z → V' ;
         else             z → V ;

}
```

Şekil 4.3: Üye Yönelimli ÇAA oluşturma Algoritması

Üye Yönelimli ÇAA oluşturma algoritmasında, ise basit algoritmadaki kümeler dışında, ayrıca bir de kullanılamayacak düğümlere ilişkin bir küme daha tanımlanmıştır. Bu kümedeki elemanlar, ağaca eklenmiş olan ve MI olan düğümler kümesidir. Bu düğümler, aynı ağaç üzerinde tekrar kullanılamayacağından kullanılmayacak düğümler kümesine eklenirler. Üye Yönelimli ÇAA oluşturma algoritmasının Şekil 4.2'de gösterilen basit ÇAA oluşturma algoritmasından farkı ise, ağaca eklenmiş düğümlerin bazılarının MI olmasından dolayı, ancak ağaç üzerindeki MC düğümler ile henüz ağaca eklenmemiş grup üyeleri arasında en kısa yolların aranacak olmasıdır. Ancak, kullanılamayacak MI düğümlerin sayısının artması ile oluşturulan bir ÇAA 'ya yeni düğüm eklenmesi mümkün olmayabilir. Bu durumda algoritma yeni bir ağaç oluşturmak üzere tekrar çalışmaya başlar. Tabii, yeni ağaç oluşturulmaya başlanmadan önce, ağaca eklenmiş düğümler kümesi ile kullanılamayacak düğümler kümesi boşaltılır. Üye Yönlendirmeli ÇAA oluşturma yöntemine ilişkin algoritma Şekil 4.3'te gösterilmiştir [3].

4.2.2 Üye Öncelikli (Member-First)

Bu algoritma [3], çalışması itibariyle minimum kapsayan ağaç oluşturma yöntemlerinden Prim Algoritmasına benzer. Prim Algoritmasında, her adımda o anki ağacın tüm düğümlerine çakışık olan linklerinden, her iki ucu birden ağaç düğümlerine çakışık olmayanları bir listeye konulur ve bu listedeki linkler maliyetlerine göre sıralanır. Bu liste içindeki en düşük maliyetli link ağaca eklenerek bir sonraki adım için aynı işlemler tekrarlanır. Üye Öncelikli algoritmada ise yine her adımda Prim Algoritması'ndaki olduğu gibi o anki ağacın düğümlerine çakışık linklerinden her iki ucu da ağaç düğümlerine çakışık olmayanları bir öncelik kriterine göre, ilgili öncelik listesine konulurlar. Daha sonra bu öncelik listesindeki en öncelikli link ağaca eklenir. Bu işleme tüm üye düğümlere ulaşılana kadar devam edilir. Bu noktaya kadar iki algoritma arasında tek farklılık Prim algoritmasında listeye konulan linklerin maliyetlerine göre sıralanmış olması, Üye Öncelikli algoritmada ise bu sıralamanın başka bir öncelik kriterine göre yapılmasıdır.

Üye Öncelikli algoritmada, öncelik listesine eklenmek üzere Prim algoritmasındaki ile aynı şekilde seçilen linklerin aslında öncelik listesine eklenmeyeceği durumlar da söz konusudur. Bu durumlardan bir tanesi öncelik listesine eklenmek istenen linkin daha önceden "kullanılamaz linkler listesi"ne konulmuş olmasıdır. Bir linkin tekrar "kullanılamaz linkler listesi"ne konulmuş olması, o linkin bir üye düğümden kaynağa giden yol üzerindeki bir MI düğüme çakışık olması demektir. Böyle bir linkin aynı

ağaçta tekrar kullanılamayacağı açıktır; çünkü bir MI düğümde ağacın dallanması mümkün değildir.

Bir linkin öncelik listesine eklenmesini engelleyen başka bir durum ise, o linkin öncelik listesinde daha önce bulunan bir link ile aynı bir düğüme çakışık olması durumudur. Bu durumda öncelik listesindeki link, öncelik kriterine göre daha öncelikli ise veya iki linkin önceliği eşitse, diğer link öncelik listesine eklenmez. Aksi durumda ise öncelik listesindeki link öncelik listesinden çıkarılıp yerine diğer link öncelik listesine yerleştirilir.

```
Güncelle_L ( v düğümü )
{
    for ( Budanmamış her e(v,u) )
    {
      if ( u ağacın elemanı değil
           VE [ ( e(v',u) ∉ L ) VEYA (e(v',u) ∈ L ama e(v,u)'dan düşük öncelikli ) ] )
        {
            e(v,u) 'yu L'ye ekle ;
            Varsa e(v',u) 'yu L'den sil;
        }
    }
}

Yeni_düğüm_ekle ( u1 düğümü )
{
    D* = D* - { u1 } ;
    u1 den kaynak düğüm s 'ye giden yoldaki ilk MC düğüm x 'i bul ;

    for ( x ile u1 arasındaki y düğümleri )
    {
        if ( Çocuk(y) > 1 ve MC(y) = 1 )
        {
            y 'nin komşularından x ile u1 arasındaki yol üzerinde olmayan z
            düğümleri için
                e(y,z)'leri buda ;

            y 'nin komşularından x ile u1 arasındaki yol üzerinde olan z
            düğümleri için
                z ile z 'nin T üzerindeki alt ağacını T' den kaldır ve bu alt
                ağaçtaki tüm linkleri L 'den kaldır ;
        }
    }
    Güncelle_L(u1) ;
    VT , V'T ve UT yi güncelle.
}
```

Şekil 4.4: "Üye Öncelikli" algoritmadaki Güncelle_L() ve yeni_düğüm_ekle() fonksiyonları.

Üye Öncelikli algoritma, her adımda öncelik listesinin en öncelikli elemanı ağaca eklerken, eklenilen linkin bir üye düğümü ağacın elemanı yapıp yapmadığını da kontrol eder. Eğer yeni bir üye düğüm ağaca eklenmişse bu üye düğümden kaynağa doğru giden yol üzerindeki düğümler incelenir ve bunlar arasında MI olanlar varsa bu düğümlere çakışık olan ve üye düğüme giden yol üzerinde olmayan tüm linkler "kullanılamaz linkler listesi"ne konulur. Eğer "kullanılamaz linkler listesi"ne konulan bir link, bu listeye konulmadan önce ağaca eklenmişse ve bu link ağaç üzerindeki bir alt ağaca gidilmesini sağlıyorsa, linkin "kullanılamaz linkler listesi"ne konulmasının ardından linkin ağaca bağladığı alt ağacın tamamı ağaçtan silinir.

Bu algoritmanın uygulanmasında, öncelik tanımı algoritmanın başarımını etkileyecektir. Diğer bir taraftan, öncelik tanımı değiştirilerek değişik ölçütlere uygun ÇAA 'lar oluşturmak da mümkündür [3].

Üye Öncelikli algoritma, optik ortam bilgilerini de değerlendirdiğinden, sadece maliyetlere bakmaz. Ayrıca, düğümlerin MC mi MI mı olduklarına da bakmak durumundadır. Bu nedenle "Üye Yönelimli" algoritmada olduğu gibi, bir ağacın oluşturulmasına devam edilemiyorsa, yeni bir ağaç oluşturulması işlemine geçilecektir. Bir ağacın oluşturulmasına devam edilip edilemeyeceğine, ağaca eklenebilecek linkleri gösteren öncelik listesinin boş olup olmadığına bakılarak karar verilir.

```
D* = D ; VT = {s} ; V'T = ∅ ;  UT = V – {s} ;  T = ∅ ;
Güncelle_L(s) ;

while ( D* ≠ ∅  ve  L ≠ ∅ )
{
        L'deki en yüksek öncelikli e(v1 ,u1 ) 'i  T ' ye ekle ;
        if ( u1 ∈ D* )
            Yeni_düğüm_ekle(u1 ) ;
}

if ( D* ≠ ∅  ve  L = ∅ )
   T ' yi ormanın bir elemanı yap ve yeni bir ağaç oluşturmak için başa dön.
else (D* = ∅ )
   Varış düğümlerinin hiç birine gitmeyen dalları buda ve algoritmayı sonlandır.
```

Şekil 4.5: Üye Öncelikli ÇAA oluşturma algoritması

Üye Öncelikli ÇAA oluşturma algoritmasında kullanılan bazı alt fonksiyonlar Şekil 4.4 'te, algoritmanın kendisi de Şekil 4.5'te gösterilmiştir. Bu algoritmada kullanılan sembollerin tanımları ve öncelik kriterinin bu semboller cinsinden tanımı da aşağıda verilmiştir [3] :

V : Graf üzerindeki tüm düğümlerin kümesi.

T : ÇAA üzerindeki ayrıtlar kümesi.

V_T : ÇAA üzerinde olan ve hala ışık bölme kapasitesi olan düğümler kümesi.

V'_T : ÇAA üzerinde olan ancak ışık bölme kapasitesi olmayan düğümler kümesi.

U_T : ÇAA üzerinde olmayan diğer düğümler

D^* : Henüz bir ağaca eklenmemiş varış düğümleri kümesi.

D : Tüm varış düğümleri kümesi.

P(s,i) : *s* kaynak düğümü ile *i* düğümü arasındaki en kısa yol.

e(v,u) : *v* ile *u* düğümleri arasındaki link

h(i) : P(s,i) yolu üzerindeki sekme sayısı.

L : Öncelik listesi (Önceliklerine göre sıralanmış linkler kümesi.)

Bu algoritmada önceliğin tanımı da aşağıdaki gibi yapılmıştır :

Öncelik: e(v_1, u_1) linki e(v_2, u_2) linkinden daha önceliklidir eğer ;

(1) $h(u_1) < h(u_2)$ ise veya

(2) $h(u_1) = h(u_2)$ ve ($u_1 \in D^*$ ama $u_2 \notin D^*$) ise

4.2.3 Evlat Edinmeli

Bu algoritma [3], IP katmanında oluşturulmuş bir ÇAA üzerinde değişiklikler yaparak, optik ortamda kurulacak olan ışıkyollarını belirler; dolayısıyla algoritma giriş parametresi olarak IP katmanında oluşturulmuş ÇAA'yı alır. Bu algoritmanın yaptığı tek işlem, IP katmanında oluşturulmuş olan ÇAA'nın her düğümünde, mevcut ağaç ile düğümün ışık bölme kapasitesinin aşılıp aşılmadığına bakıp, ışık kapasitesi

aşılmış olan düğümlerin çocukları olan düğümleri (*yetim düğümler*) ağacın ışık bölme kapasitesi aşılmamış düğümlerinden birine bağlayarak WDM katmanında çoklu aktarım ormanı oluşturmaktır. Bu nedenle de algoritmanın adı ”Evlat Edinmeli” konulmuştur .

Bu algoritma, yetim düğümlerin ağaç üzerindeki düğümlerden hangileri ile birleştirileceği konusunda değişik yöntemler izleyebilir. Burada, yetim düğümlerin ağaca birleştirilmesi yöntemlerinden yalnız ikisi gösterilecektir. *Kaynak-yönelimli* (ingilizce adı Re-Route-To-Source) adlı yaklaşımda, yetim olan düğümler, annelerini kaynağa bağlayan yoldaki en yakın MC düğüm üzerinde ormana dahil olurlar. *MC-yönelimli* (ingilizce adı Re-Route-To-Any) yaklaşımında ise, yetim olan düğümler kendilerine en yakın (maliyeti en düşük) MC orman düğümü ile birleşerek ormana dahil olurlar. “Evlat edinmeli” algoritmanın sözde kodu Şekil 4.6’da verilmiştir (Kod içinde Kaynak-yönelimli ve MC-yönelimli yaklaşımların ikisi de gösterilmiştir.). Kod içinde kullanılan bazı terimlerin tanımları da aşağıda gösterilmiştir [3]:

Çocuk(*i*) : *i* düğümünün IP katmanında oluşturulan ÇAA ’sındaki çocuk sayısı .

MC(*i*) : *i* düğümünün ışık bölme yeteneği (sayısal olarak) .

```
IP 'de kullanılan bir ÇAA oluşturma algoritması ile bir ÇAA oluştur.

for ( tüm ağaç düğümleri i ' ler ) ( ağaç yatay olarak taranır )
{
    if ( Çocuk( i ) > MC( i ) )
        if (Kaynak-yönelimli)
        {
            while ( i 'nin yetim çocuğu var )
            {
                for ( i 'den kaynak düğüme giden en kısa yoldaki j düğümleri )
                    if ( Çocuk( j ) < MC( j ) )
                        j ' ye taşıyabileceği tüm yetim çocukları bağla ;
            }
        }
        else if (MC-yönelimli)
        {
            for ( i'nin tüm yetim çocukları için )
            {
                Öksüz çocuğu (Maliyet olarak) en yakın ve MC 'si uygun
                olan ağaç düğümüne bağla ;
            }

        }

    i düğümünü bir orman düğümü olarak ata ;
}
```

Şekil 4.6: “Evlat edinmeli” ÇAA oluşturma algoritmasının sözde kodu.

4.3 Statik ÇAA oluşturma algoritmalarında Dalgaboyu Atama

Statik ÇAA oluşturma algoritmalarında, değişik yol seçimi şekilleri açıklanmıştı. Ancak, WDM dalgaboyu atamalı ağlarda yol atama işlemi içerisinde yalnız yol seçme değil, dalgaboyu atamasının da yapılması gerekir. Dalgaboyu atanması işleminin dalgaboyu dönüşümüne ne kadar bağlı olduğu, Bölüm 2.4'te açıklanmıştı. Dalgaboyu dönüşümü, çoklu aktarım için özellikle MC olan düğümlerde dalgaboyu dönüşümü yapılabilmesi durumunda, avantajlı olmaktadır. Bu durumda, MC düğüm, ışığı hem bölüp hem de dalgaboyunu değiştirebilir ve bu şekilde çoklu aktarım ormanı içindeki birden çok ağaç için bir çoklu aktarım kaynağı görevi görür. Böylece MI olan düğümün kurmak istediği ikinci yol, çoklu aktarımın gerçek kaynağına kadar uzanmak zorunda kalmaz ve dalgaboyu daha verimli kullanılmış olur [3,8].

MC düğümlerin dalgaboyu dönüştürmesi önemli olduğundan, yol seçme işleminin ardından belirlenmiş olan çoklu aktarım ormanı, *segman* olarak tanımlanan parçalara ayrılır [3,8]. Her segman, içinde bir tek dalgaboyu kullanılabilecek linkler dizisinden oluşur. Segmanların belirlenmesi için dalgaboyu yönlendiricilerinin dönüştürme özelliklerine bakılır. Segmanların başı ve sonu dalgaboyu dönüştürme özelliğinde olan düğümlerdir. Ayrıca segmanların başında çoklu aktarımın kaynak düğümü ve segmanların sonlarında da ÇAA'nın yaprak düğümleri de olabilir.

Çoklu aktarım ormanı içindeki segmanlar bulunduktan sonra, segman uçlarındaki düğümlerin dalgaboyu dönüştürme kapasitesine göre, her bir segmana bir dalgaboyu atanır. *Dalgaboyu dönüştürme kapasitesi*, dalgaboyu dönüştürme yeteneği olan düğümler varsa, bu düğümlerin bir dalgaboyunu var olan tüm diğer dalgaboylarına çevirip çeviremeyeceği ile ilgilidir. Bir dalgaboyu, sınırlı sayıda dalgaboyuna dönüştürülebiliyorsa buna *sınırlı dalgaboyu dönüştürme kapasitesi*, var olan tüm dalgaboylarına dönüştürülebiliyorsa, buna da *tam dalgaboyu dönüştürme kapasitesi* adı verilmektedir. Bu kapasite ölçütleri göz önüne alınarak, orman üzerindeki linklere birer dalgaboyu atanması ile dalgaboyu ataması işlemi tamamlanmış olur. Bu arada ağ üzerindeki hiçbir düğüm dalgaboyu dönüştürme kapasitesine sahip değilse ormanın segmanlara ayrılması, kaynak ile uç düğümler arasında yollar oluşmasını sağlayacaktır. Bu yollardan birbirleri ile hiçbir ortak linki olmayanları aynı dalgaboyuna atanarak aynı ÇAA 'ya dahil edilebilirler [3,8].

4.4 Dinamik WDM ÇAA Oluşturma Algoritmaları

Dinamik ÇAA oluşturulabilmesi için, WDM katmanında dinamik çoklu aktarım ağacını destekleyen bir yol atama yöntemi kullanılması gerekir. OBS ve OLS en az düzeyde bellek ve elektronik ortam desteği gereksinimleri ile dinamik çoklu aktarımın gerçekleştirilmesini sağlayabilirler.

Bu çalışmada, WDM katmanında OLS tarafından desteklenen iki dinamik çoklu aktarım protokolü anlatılacaktır. Bu protokollerden ilki, IP katmanda oluşturulmuş olan ÇAA 'yı kullanarak yol atamayı WDM katmanında gerçekleştirirken (WDM Mesajlaşmalı Çoklu Aktarım Protokolü), ikincisi dalgaboyu yönlendiricilerin ışık bölme kapasiteleri bilgisini de kullanarak çoklu aktarım yol atamasını doğrudan IP katmanında gerçekleştirir (IP Mesajlaşmalı Çoklu Aktarım Protokolü).

4.4.1 WDM Mesajlaşmalı Çoklu Aktarım Protokolü

Bu protokolün [5] uygulanabilmesi için gerekli ağın mimarisi şu şekilde olmalıdır : Her bir düğümde IP katmanı yönlendiricileri, WDM çoklu aktarım denetçileri ve optik anahtarlar bulunmalıdır. Burada, IP katman yönlendiricileri 3. katman, WDM çoklu aktarım denetçileri 2. katman (WDM katmanı) ve optik anahtarlar da 1. katman (fiziksel katman) varlıklarını temsil etmektedir. WDM çoklu aktarım denetçileri, sahip oldukları yönlendirme bilgilerine bağlı olarak optik anahtarların giriş ve çıkışları arasındaki anahtarlamasını denetlerler. IP katmanı yönlendiriciler, IP katmanında DVMRP protokolü uyarınca mesajlaşarak ÇAA 'yı oluştururlar. Her IP katmanı yönlendirici, ÇAA üzerinde yönlendirilecek paketlerle ilgili yol atama bilgilerini çoklu aktarım yönlendirme tablolarında (Multicast RoutingTable – MRT) tutar. WDM katmanı çoklu aktarım denetçileri, IP katmanı yönlendiricilerle yalnız MRT bilgilerini paylaşır; dolayısıyla çoklu aktarım denetçilerinin IP katmanında gerçekleşen mesajlaşmadan haberi yoktur. Aynı şekilde, IP katmanı yönlendiriciler de WDM katmanında gerçekleşen mesajlaşmadan habersizdir [5].

WDM katmanı çoklu aktarım denetçileri, IP katmanı yönlendiricilerin MRT tablolarından yararlanarak kendilerine ait bir yönlendirme tablosu oluştururlar. WDM katmanında OLS kullanıldığı için, bu tablolarda kaynak ve varışlara ilişkin adresler yerine etiket değerleri bulunacaktır. Buna göre, belli bir girişten gelen bir etiketin hangi çıkış etiketine atanacağı ve bu etiketin hangi çıkış kapısını gösterdiği verileri bu tablolarda yazacaktır [5].

Protokol yığınında, IP katmanında DVMRP protokolü aynen uygulanmış, WDM katmanında ise, düğümlerin ışık bölme yeteneklerine göre ÇAA 'nın yeniden belirlenmesi işlemi için üç adet yeni mesaj tanımlanmıştır : Bu mesajlar sırasıyla REQUEST, CUT ve ACTIVE adlı mesajlardır. REQUEST yeni bir LSP oluşturma isteğini, CUT ise var olan bir LSP 'yi kaldırma isteğini belirten mesajlardır. ACTIVE mesajı ise, yetim kalan bir düğümün dahil olduğu ÇAA hakkında bilgi edinmesi için periyodik olarak yayınlanan bir mesajdır [5].

WDM katmanındaki protokolün çalışma şekli şöyledir : IP katmanında MRT tabloları değiştiğinde bu değişiklik WDM katmanı çoklu aktarım denetçilerine ilişkin tablolarda değişikliğe neden olacaktır. Bu değişiklik, eğer yeni bir LSP kurulması ya da bir LSP 'nin kaldırılmasını gerektiriyorsa, bu durumda REQUEST ve CUT mesajları gönderilecektir. Aksi taktirde, hiçbir mesajlaşma yapılmayacaktır. Ayrıca protokol kullanılan türevine göre ACTIVE mesajlarını periyodik olarak WDM katmanında yayınlar [5].

Bu protokol yığını dinamik bir çoklu aktarımı desteklediğinden, üç farklı durumun ayrı ayrı incelenmesi gerekmektedir. Bu üç ayrı durum i) ÇAA'nın oluşturulma aşaması ii) Gruba üye Eklenmesi iii) Gruptan üye Çıkarılması durumlarıdır.

WDM katmanındaki protokolünde tanımlanmış olan mesajların, ne zaman ve nasıl göndereceği konusunda farklı yaklaşımlar söz konusu olabilir. Bu nedenle bu katmana ilişkin protokolün üç farklı türevi, protokol yığını içinde incelenecektir. Protokolün türevlerinin adları "MI Başlatmalı", "Yetim Başlatmalı" ve "Komşu Destekli" olarak adlandırılmıştır [5].

4.4.1.1 MI Başlatmalı (Parent-Initiated LSP Setup)

Bu protokolün özelliği, üyelerin ağaca eklenmesi aşamasında bir MI düğümün, birden fazla düğüme paket iletmesi gerekmesi halinde, yeni bir dalgaboyu atanması işlemini MI düğümün başlatmasıdır. Bunun için REQUEST mesajı kullanılır. Protokolün algoritması sözde kod olarak Şekil 4.7'de verilmiştir [5].

Protokolde, üye ekleme ve kaldırma aşamalarında gerçekleştirilen adımlar ayrıca aşağıda gösterilmiştir.

Üye Ekleme aşamasında gerçekleştirilen adımlar [5] :

1. Gruba üye olmak isteyen düğüm, IP katmanında GRAFT mesajı gönderir .

2. Bu GRAFT mesajını alan ağaç düğümünün MRT tablosu değişir .

3. Bu değişiklik 2. katmanda da anlaşılır .

4. Düğüm MI ise 2 . katman REQUEST mesajı oluşturulur ve bu mesaj düğümün annesine gönderilir, düğüm MC ise üye ağaca bağlanır .

```
if ( Bir MI düğüm birden çok çocuğuna paket iletecek ise )
{
     - MI düğüm, kendisinden kaynağa giden yoldaki komşu düğümüne
       REQUEST mesajı gönderir.

     - REQUEST mesajını alan düğüm
     if ( MC ise ve Dalgaboyu dönüştürme yeteneği varsa )
     {
        - Yeni yol için yeni bir dalgaboyu ata.
        - Gerekli_Değişiklikler() fonksiyonunu çağır.
     }
     else
     {
        - Yol atama tablosunda yeni yol için yer ayır.
        - REQUEST mesajı kaynağa doğru giden yoldaki komşu düğüme
          gönder.
     }
 }


Gerekli Değişikler()
{
   - Yol atama tablosunun güncelle: Yeni yol için tabloya yeni
     bir çıkış kapısı ve bu çıkış kapısına karşılık yeni bir etiket
     ata.

   - REQUEST mesajının alındığı kapıya REQ_ACK (alındı) mesajı
     gönder.
}
```

Şekil 4.7: "MI Başlatmalı" ÇAA algoritmasının sözde kodu.

Üye Kaldırma aşamasında gerçekleştirilen adımlar :

1. Gruptan ayrılmak isteyen üye, IP katmanında PRUNE mesajı gönderir .

2. Bu PRUNE mesajını alan ağaç düğümünün MRT tablosu değişir .

3. PRUNE mesajını alan ağaç üyesi, birden fazla çocuğa aktarım yapan bir MI düğüm ise 2. katmanda bu düğümden kaynağa giden yoldaki komşu düğüme CUT mesajı gönderilir. (Aynı durum CUT mesajını alan düğüm için de geçerli ise CUT kaynağa doğru üst düğümlere aktarılır.) Aksi taktirde MRT'de üye olan düğüme ilişkin satır silinerek üye ağaçtan kaldırılmış olur .

4.4.1.2 Yetim Başlatmalı (Child-Initiated LSP Setup)

Bu protokolün [5], "MI Başlatmalı" protokolden tek farkı, üyelerin ağaca eklenmesi aşamasında, bir MI düğümün birden fazla düğüme paket iletmesi gerekmesi halinde, yeni bir dalgaboyu atanması işlemini, paketlerin iletilemeyeceği yetim bir düğümün başlatmasıdır. Yani REQUEST mesajını yetim düğüm MI düğüme göndermelidir. Bundan sonra gerçekleşecek olaylar "MI başlatmalı" protokol ile aynı şekilde gerçekleşecektir [5].

Bu protokolde, "MI Başlatmalı" protokolde kullanılmayan ACTIVE mesajları da kullanılır. Bu mesajlar, ağaca bağlanamayan yetim düğümün, WDM düzeyindeki ÇAA ve bu ağacın kaynağı hakkında bilgiye sahip olmasını sağlamak amacı ile ağda periyodik olarak yayınlanır. Bu sayede yetim düğüm ağaca nasıl bağlanacağına karar verebilir ve REQUEST mesajını hangi komşusuna göndereceğini bu şekilde belirler [5].

Üye ekleme ve kaldırma durumlarında da protokolün çalışması "MI Başlatmalı" protokolün çalışması ile aynıdır [5].

4.4.1.3 Komşu Destekli (Relative-assisted LSP Setup)

"MI Başlatmalı" ve "Yetim Başlatmalı" protokollerin her ikisinde de IP katmanında oluşturulan ÇAA 'ya sadık kalınmaktadır; yani IP katmanında hangi düğümler birbirlerine bağlı ise, WDM katmanında da aynı düğümler birbirine bağlıdır. "Komşu Destekli" protokol ise [5], IP katmanındaki ağaca sadık kalmayıp WDM katmanında düğümlerin, IP katmanından farklı düğümler üzerinden birbirlerine bağlanmalarını sağlayabilir. Bu şekilde, IP katmanında var olan yollardan farklı olan yolları da değerlendirerek dalgaboyu kullanımını daha verimli olarak kullanmayı amaçlar [5].

Bu protokolde, bir düğüm, WDM katmanında ağaca bağlanmak için kaynağa doğru en kısa yol üstündeki komşusu (IP katmanındaki ÇAA 'daki, annesi) MI ise bu düğüm üzerinden bağlanmak yerine, yeni bir ağaç kurulmasını gerektirmeyen, ağaca bağlanabileceği başka bir komşusunu seçer ve bu komşusuna REQUEST mesajı gönderir. Ancak IP katmanında mesajın IP katmanındaki anneden geliyor gibi gözükmesi gerektiğinden, WDM katmanındaki çoklu aktarım denetçisi gerçekte iletimin yapıldığı kapıdan gelen veriyi IP katmanın beklediği kapıdan gelecek şekilde IP katmanına verir [5].

“Komşu Destekli” protokolde üye eklenmesi işlemi de ağaç oluşturma aşamasına benzer şekilde yapılır. Üye kaldırma aşamasında ise, ağaca IP katmanında ve WDM katmanında farklı yollar üzerinde bağlanmış olan düğüm, IP katmanındaki annesine IP katmanındaki PRUNE mesajını gönderirken, WDM katmanında gerçekte kurulu olan yol üzerindeki komşusuna da CUT mesajı gönderir. CUT mesajı gönderilen düğüm gruptan ayrılmışsa, protokolün nasıl davranacağı belirsizdir. Bunun nedeni, IP katmanındaki ağaç ile WDM katmanındaki ağacın birbiri ile tutarlı olmamasıdır [5].

4.4.2 IP Mesajlaşmalı Çoklu Aktarım Protokolü

Bu protokol yığınındaki çalışma [5], “WDM Mesajlaşmalı Çoklu aktarım Protokol” yığınındaki çalışmaya çok benzer; ancak bu protokol yığınında, WDM katmanındaki mesajlar kaldırılmış ve bu mesajlar IP katmanına taşınmıştır. Bu nedenle IP katmanındaki protokol değişmiş ve WDM katmanında herhangi bir karar mekanizması kalmamıştır. Tabii, daha önce WDM katmanında yapılan mesajlaşmaların IP katmanında yapılması, IP katmanında, düğümlerin ışık bölme ve dalgaboyu dönüştürme yeteneklerinin bilinmesini de gerektirmektedir [5].

“WDM Mesajlaşmalı Çoklu aktarım" protokol yığınında WDM katmanında kullanılan REQUEST ve CUT mesajlarının yaptıkları işleri, bu protokol yığınında sırasıyla GRAFT ve PRUNE mesajları yapacaktır. WDM katmanında kullanılan ACTIVE mesajı da ek bir mesaj olarak IP katman mesajları arasında yer alacaktır ve aynı işlevi görecektir [5].

GRAFT ve PRUNE mesajının IP katman protokolünde zaten belirli anlamları vardır ve bunların üzerine REQUEST ve CUT mesajlarının da işlevleri yüklendiği için GRAFT ve PRUNE’un ne zaman hangi anlama geleceğinin belirlenmesi gerekmektedir. Bu mesajların ne zaman nasıl yorumlandığı **Tablo 4.1**’de gösterilmiştir [5].

“WDM Mesajlaşmalı Çoklu Aktarım” protokol yığınında olduğu gibi bu protokol yığının da üç türevi gerçeklenebilir. Bu protokol türevleri, WDM Mesajlaşmalı yöntemdeki türevlerle çok benzerlik göstermektedir bu nedenle adları aynı kabul edilecektir [5].

“MI Başlatmalı” ve “Yetim Başlatmalı” protokollerde REQUEST ve CUT mesajlarının yerine sırasıyla GRAFT ve PRUNE mesajları kullanılarak aynı çalışma IP mesajlaşmalı protokol yığını için de gerçekleştirilebilir. “Komşu Destekli” protokolü

Tablo 4.1: GRAFT ve PRUNE mesajlarının IP mesajlaşmalı Çoklu Aktarım Protokolündeki yorumları

Mesaj / Düğümün durumu	GRAFT	PRUNE
Mesajın alındığı link üzerinde LSP yok.	Gruba katılma isteği	—
Mesajın alındığı link üzerinde LSP var.	Yeni bir LSP oluşturma isteği	—
Mesajın alındığı link üzerinde bir tek LSP var.	—	Gruptan ayrılma isteği
Mesajın alındığı link üzerinde birden fazla LSP var.	—	Belli bir LSP'yi yoketme isteği

protokolü için ise daha önce de söz edildiği gibi ACTIVE mesajları, IP katmanında yayınlanmalıdır. "Komşu Destekli" protokolün IP Mesajlaşmalı protokol yığınına adapte edilebilmesi için, çalışmanın şu şekilde değiştirilmesi gerekir : ÇAA oluşturulurken düğümler kendilerine gelen RPF mesajlarını hemen reddetmezler. Gelen RPF mesajı o düğümü, o an için ÇAA 'ya bağlayacak en iyi yol üzerinde ise, düğüm RPF mesajının geldiği yol üzerinden ağaca bağlanacaktır. Sonradan düğüme gelen başka bir RPF mesajı, düğümün o anki bağlantısından daha az maliyetli ise daha önce kurulmuş olan bağlantı PRUNE mesajı kullanılarak bozulurken, yeni RPF mesajının geldiği yol üstünden ağaca bağlanılır. Bu sayede, normal olarak kaynağa en kısa yol üstünde bulunan yol üstünden ÇAA 'ya bağlanması gereken bir düğüm, bu yol üzerindeki annesi MI olduğu için ağaca bağlanamayacağından (bu yol üzerinden çoklu aktarım yapmanın maliyeti oldukça yüksek olacaktır.), gelen diğer RPF paketleri sayesinde ağaca başka komşu düğümler üzerinden bağlanacaktır [5].

5. ÖNERİLEN ALGORİTMA – İKİ KATMANLI KAPSAYAN AĞAÇ YÖNTEMİ

Bu çalışmada, WDM ağlardaki ÇAA oluşturma algoritmalarına yöntemlerine alternatif olabilecek bir algoritma önerisinde bulunulmuştur. Bu algoritma, Bölüm 4'te anlatılan diğer algoritmalar gibi, paket çoğullama işleminin yalnızca MC düğümler üzerinde yapılmasını sağlayan bir ÇAA oluşturmaya çalışacaktır. Bunu sağlamak için, algoritmanın önerdiği çözüm, ÇAA oluşturma işleminin yalnızca MC düğümler üzerinde yapılmasıdır. Bu sayede, algoritma ÇAA oluşturma sürecinin elektronik ağlar için geliştirilmiş olan herhangi bir ÇAA oluşturma algoritması kullanılarak gerçekleştirilebilmesine imkan tanır.

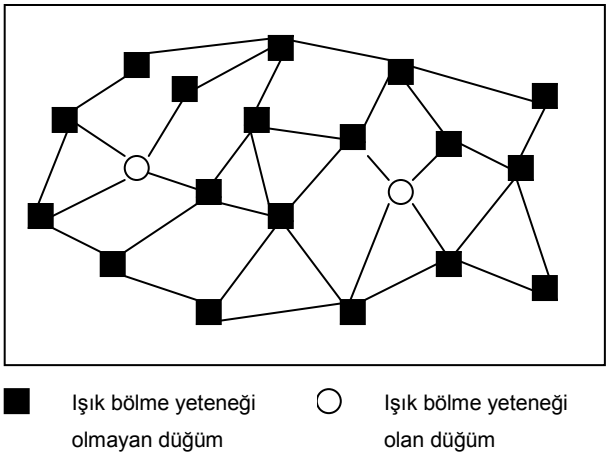
Önerilen algoritmanın, ÇAA oluşturma işlemini, yalnızca MC düğümler üzerinde gerçeklemesi, ÇAA'nın tüm ağ düğümleri üzerinde çalışmak yerine, ağ düğümlerinin bir alt kümesi üzerinde çalışmasına neden olacaktır. Ancak çoklu aktarım üyeleri ağın herhangi bir düğümünde olabilir ve algoritmanın olası tüm düğümlere erişebilen bir çözüm üretmesi gerekmektedir. Bunun için önerilen çözüm, iki katmanlı sıradüzensel bir yapı oluşturmaktır.

5.1 Algoritmanın Kullandığı İki Katmanlı Yapı

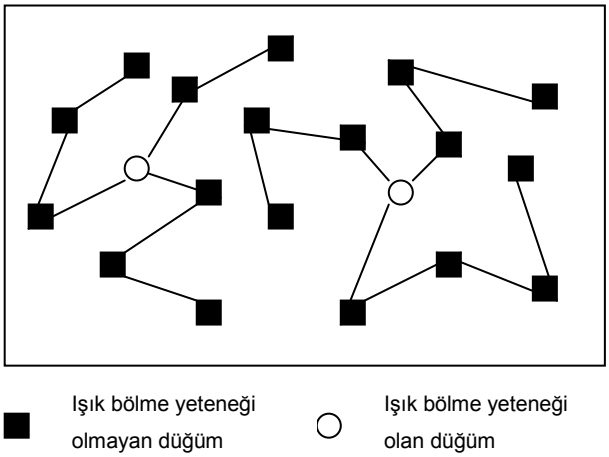
Optik WDM düğümlerinden oluşan ağı, algoritmanın kullanabileceği hale getirmek için iki katmanlı yapı şu şekilde oluşturulur : 1. katmanda (veya üst katmanda) MC düğümler, 2. katmanda (veya alt katmanda) ise MI düğümler bulunur. Her alt katman düğümü tek bir yol üzerinden tek bir üst katman düğümüne bağlıdır. Üst katman düğümleri ile alt katman düğümleri arasında bu şekilde bir bağlantı kurulduğunda, bir üst katman düğümü ve o düğüme bağlı olan tüm alt katman düğümleri, bu üst katman düğümü tarafından temsil edilebilecek bir düğüm grubu oluştururlar. Bu düğüm gruplarına *alt ağ* adı verilmiştir.

Alt ağlar, üst katmandaki tek bir üst katman düğümü ile temsil edilebildiğinden ve bütün alt katman düğümleri bir alt ağın içinde yer aldığından (alt ağın elemanı olduğundan), üst katman düğümleri ve bunlar arasında kurulacak bir üst katman ağı (grafı) yine tüm ağı temsil edecektir. Üst katman düğümleri MC özelliğinde

olduğundan, üst katman ağı, çoklu aktarım işlemi açısından elektronik bir ağ ile aynı özellikleri taşır; yani üst katman ağının tüm düğümleri paket çoğullayabilir. Bu nedenle üst katmanda ÇAA oluşturma işleminin elektronik ağlardakinden bir farkı yoktur ve bu katmanda istenilen herhangi bir ÇAA oluşturma algoritması yürütülebilir.



Şekil 5.1: 20 düğümlü bir ağ.



Şekil 5.2: Şekil 5.1'deki ağ üzerinde gruplandırma yapıldıktan sonra elde edilen alt ağlar.

Alt ağlar, üst katman açısından bir tek düğüm gibi görülürken, alt katmandan bakıldığında, MC bir düğüm ile bu düğümden çıkan ve en az bir düğüm içeren kollardan oluşan bir yapıdır. Bu durumda, bir alt ağda ikiden fazla düğüme bağlı olabilecek tek düğüm MC düğümdür. Bu özelliğinden ötürü, MC düğüme *alt ağın merkezi* adı verilmiştir. Merkez düğüm dışındaki tüm alt ağ düğümleri ise en fazla iki

alt ağ düğümüne bağlı olabilir. Bunun nedeni ise, düğümlerin MI olmasıdır. (Şekil 5.2'de Şekil 5.1'te verilen ağ için oluşturulmuş iki alt ağ görülebilir.)

Alt ağ merkezi dışındaki düğümlerinden, alt ağın tek bir düğümüne bağlı olan düğümlerine de *alt ağın uç düğümleri* adı verilir. Bir uç düğüm alt ağın merkezinden başlayan ve birbirine peş peşe bağlanmış olan alt ağ düğümlerinden oluşan bir yolun (bu yollara *alt ağın kolları* denilmiştir) son düğümüdür. Bu düğümün alt ağa bir tek bağlantısı olduğundan başka bir düğüme de bağlanabilmesi mümkündür. Bu nedenle de, bu düğümler bir alt ağın merkezinden diğerine giden yolların oluşturulmasında kullanılırlar (bkz. Alt Ağlar Arasındaki Bağlantıların Oluşturulması).

Alt ağ merkezi ile alt düğümleri arasındaki bağlılık alt ağ kolları ile sağlanırken, farklı iki alt ağın düğümlerinin birbirlerine bağlılığı, alt ağ merkezleri arasında yollar kurularak sağlanmıştır. Bu şekilde tüm ağ düğümlerini kapsayan bir ağaç oluşturulmuş olur. Algoritmanın adındaki kapsayan ağaç ifadesi bu durumu ifade etmek üzere konulmuştur.

Alt ağ merkezlerini birbirine bağlayan yollar, alt ağların aralarındaki bağlantıları oluşturduğu için, üst katman açısından bakıldığında, üst katman düğümleri arasındaki linklere karşı gelirler. Yani ağ üzerindeki fiziksel düğüm ve linklerden oluşan bir yol, üst katmanda bir tek link ile ifade edilir. Bu şekilde fiziksel düğümler üzerinde oluşturulmuş olan kapsayan ağaç mantıksal olarak bir üst katman grafı oluşturulmasını sağlar.

5.2 Algoritmada Kullanılan Varsayımlar

Algoritmanın çalışmasında yapılan varsayımlar aşağıdaki gibi sıralanabilir :

1) WDM düğümlerin kendilerine gelen işaretin bir kısmını kendileri için kullanıp geriye kalan kısmını da başka bir düğüme iletilmelerinin mümkün olduğu düşünülmüştür.

2) Diğer WDM ÇAA oluşturma algoritmalarında olduğu gibi algoritmanın yürütüleceği WDM ağ düğümlerinden yalnız bir kısmının ışık bölme yeteneğine sahip olduğu düşünülmüştür.

3) Ağ düğümleri arasındaki linkler çift yönlü trafik taşıyabilmektedir. Bu durum, zaten genelde linklerin sahip olduğu düşünülen bir özelliktir.

4) Yalnızca çoklu aktarım göz önüne alınmıştır. Yöntem bu hali ile çoklu aktarımı desteklemez.

5.3 Algoritmanın Aşamaları

İki katmanlı Kapsayan ağaç algoritması çoklu aktarım yöntemleri içinde anlatılan, kapsayan ağaç kullanılarak gerçekleştirilen ÇAA oluşturma yöntemine benzer. Algoritmada önce tüm ağ düğümlerini kapsayan bir ağaç oluşturulur. Daha sonra her çoklu aktarım grubu için bu ağacın yalnızca gerekli kısımları kullanılır, geriye kalan kısımları budanır. Algoritma bu işi birbirinden ayrı yürütülen iki aşamada yapar. Bu aşamalardan ilki, bir önceki kısımda (5.1) anlatıldığı gibi, ağ düğümleri üzerinde bir kapsayan ağaç oluşturur. Daha sonra bu kapsayan ağaçtan yola çıkarak, mantıksal bir yapı olan üst katman grafını oluşturur.

Algoritmanın ikinci aşaması ise, verilen bir "kaynak ve çoklu aktarım üyeleri" grubu için, bu düğümlerin hangi alt ağlarda olduklarını belirler ve kaynağın elemanı olduğu alt ağı üst katman grafındaki kaynak, üyelerin elemanı olduğu alt ağları üst katmandaki üye düğümler haline getirir. Böylece fiziksel ağdaki "kaynak ve çoklu aktarım üyeleri" grubuna karşı gelen üst katman grafındaki "kaynak ve çoklu aktarım üyeleri" grubunu oluşturur. Ardından, üst katmanda oluşturulan bu grup için, elektronik ortam ağlarında kullanılan herhangi bir çoklu aktarım algoritmasını kullanarak üst katmanda bir ÇAA oluşturur. Son olarak da üst katmanda oluşturulan ÇAA'nın, alt katmandaki kaynak ve üye düğümleri kapsayan bir eşdeğerini oluşturur.

Algoritmanın ilk aşaması belli bir ağ için iki katmanlı yapının oluşturulmasını sağlar. Bu yapı bir kere oluşturulduktan sonra grafın özellikleri değişmedikçe, yani graftaki düğüm ve linklerin sayısı ile graftaki MC düğümler değişmedikçe, kullanılabilir. Dolayısıyla sabit bir ağ için ilk aşama yalnız bir kere yürütülecektir.

Algoritmanın ikinci aşaması ise her farklı "kaynak ve çoklu aktarım üyeleri" grubu için ayrı ayrı çalıştırılmalıdır. Dolayısıyla ÇAA oluşturulması işlemi ikinci aşamada yapılır.

5.3.1 Algoritmanın Birinci Aşaması

Algoritmanın bu aşamasında bir kapsayan ağaç oluşturulduğu söylenmişti. Bu kapsayan ağacın oluşturulması işlemi iki adımda gerçekleştirilir. Bu adımlardan ilki

alt ağların oluşturulması diğeri ise alt ağlar arasındaki bağlantıları kurulması işlemidir.

5.3.1.1 Alt Ağların Oluşturulması

Algoritmanın bu adımında, tüm ağ düğümlerini alt ağlara ayrıştırma işlemi yapılır. Bu ayrıştırma işlemi sonunda alt ağlar içine yerleştirilebilen her düğüm bir ve yalnız bir alt ağın elemanı olacaktır. Ayrıca her alt ağda ışık bölme yeteneği olan bir tek düğüm olacaktır. Bu nedenle, oluşturulan tüm alt ağlar ayrık düğüm grupları olacaklardır. Bu sayede bir alt ağ, tek bir üst katman düğümü kullanılarak temsil edilebilmektedir.

Algoritmanın bu adımında alt katman düğümleri, bir üst katman düğümü ile beraber bir alt ağa dahil edilirken, düğümlerin yalnızca gruplandırılması değil, MI düğümlerin elemanı oldukları alt ağın merkezine doğru yollarının da belirlenmesi sağlanır. Bu şekilde algoritmanın bu aşamasının sonunda alt ağların kolları ortaya çıkar.

Alt ağların oluşturulması için kullanılan yöntem ise şu şekilde işlemektedir: Başlangıçta ağ üzerindeki MC düğüm sayısı kadar alt ağ oluşturulur ve bu alt ağların her biri yalnız bir ışık bölme yeteneği olan düğüm içerir. Bu başlangıç durumunda, ağın MC düğümleri dışında kalan tüm düğümler alt ağların dışındadır. Bu nedenle, bu düğümlere *dış düğümler* adı verilmiştir. Alt ağ oluşturma algoritmasının her adımında bir dış düğüm ve o dış düğümü alt ağa bağlayan yol üzerindeki tüm düğümler alt ağa eklenir ve o alt ağın elemanı olurlar; dolayısıyla dış düğümlerin sayısı her adımda azalır.

Bir alt ağın yalnızca uç düğümlerine yeni düğümler eklenebilir. Alt ağların uç düğümleri ise alt ağ kollarının sonundaki düğümlerdir. Ayrıca, eğer tüm komşuları alt ağın elemanı değilse merkez düğüm de bir uç düğüm olarak görülür. Bir başka deyişle, bir alt ağın uç düğümleri, henüz ışık bölme kapasitesi aşılmamış düğümleridir. Alt ağların uç düğümleri dışındaki tüm düğümlerinin (bu düğümler de *iç düğümler* olarak adlandırılmıştır) hepsi iki farklı düğüme bağlıdır ve ışık bölme kapasiteleri tamamıyla kullanılmaktadır. Bir iç düğümün ışık bölme kapasitesinin aşılmaması için, bir düğüm ne zaman iç düğüm haline gelirse, o düğüme çakışık olan linklerden, alt ağdaki komşularına da çakışık iki linki dışındaki tüm linkler ağdan koparılır. Bu şekilde, daha ileriki bir adımda bir dış düğümden alt ağın uç düğümlerine giden yollar bulunurken iç düğümler üzerinden geçen yolların kullanılması engellenmiş olur.

Algoritma her adımında dış düğümlerden birini seçip o dış düğümün en düşük maliyetle bağlanabileceği uç düğümü bulur. Bunun için tüm alt ağların uç düğümlerine olan en düşük maliyetli yollar bulunur. Her alt ağın uç düğümleri için bulunan en düşük maliyetli yollar içinden maliyeti en düşük olan uç düğüm, dış düğümün o alt ağa bağlanmak için kullanabileceği birer aday düğüm olarak belirlenir. Tüm alt ağlar için belirlenen aday düğümlerden dış düğüme en düşük maliyetli yolla ulaşılanı ise dış düğümün bağlanacağı uç düğümü belirler. Böylece dış düğüm ile seçilen uç düğüm arasındaki tüm düğümler uç düğümün alt ağına bağlanırlar ve alt ağa eklenen dış düğüm alt ağın bir uç düğümü haline gelir.

Algoritmanın belli bir adımda, bir alt ağa bağlamak için inceleme yapacağı dış düğümü rasgele olarak seçer. Ancak ağa tek bir linkle bağlı bir düğümün varlığı bulunursa bu linkin diğer linklerden önce alt ağlara bağlanması sağlanır. Bu şekilde olabildiğince çok düğümün bir alt ağın elemanı olması sağlanır.

Yukarıda anlatılan Alt Ağ Oluşturma algoritmasının sözde kodu Şekil 5.3'te gösterilmiştir. Bu sözde kod içinde kullanılan bazı simgeler ise şöyledir :

L_I : MC ağ düğümleri listesi.

L_d : Dış düğümlerin listesi. Başlangıçta bu liste L_I dışında kalan düğümlerden oluşur.

Ağ(i) : i . alt ağ.

Aday(i , j) : Ağ(i) içinde bir dış düğümün bağlanabileceği j. aday düğüm.

Alt_Ağ_Aday(i) : i. alt ağda dış düğüm ile arasındaki yolun en düşük maliyetli olduğu aday düğüm.

Ağ_Aday : Tüm ağ içinde, incelenen dış düğüm ile arasındaki yolun maliyeti en düşük olan aday düğüm.

M (x,y) : *x* ve *y* düğümleri arasındaki en kısa yolun maliyeti.

Alt Ağ Oluşturma algoritmasının oluşturduğu alt ağlara birer örnek Şekil 5.2'de görülebilir. Bu şekildeki alt ağlar Şekil 5.1'deki ağ için oluşturulmuştur. Elde edilen bu alt ağlar link maliyetlerine ve alt ağ düğümlerinin incelenme sırasına göre farklılık gösterebilir. Şekil 5.2'de alt ağ şekilleri, alt ağların biçimleri hakkında fikir vermek amacıyla gösterilmektedir.

5.3.1.2 Alt Ağlar Arasındaki Bağlantıların Oluşturulması

Alt ağların oluşturulması bittikten sonra sıra, alt ağlar arasındaki bağlantıların kurulmasına gelir. Alt ağlar arasındaki bağlantılar, alt ağların merkez düğümleri arasında oluşturulacak yollar biçiminde olacaktır. Bu yollar iki merkez düğüm arasındaki alt katman düğümlerini kullanacaktır. Bu durumda alt ağ merkezine bağlı alt ağ kollarından bazıları, alt ağın merkezinin başka bir alt ağın merkezi ile arasındaki yolu ile çakışacaktır. Çakışmanın sorun çıkarmaması için, alt ağ merkezleri arasında kurulan yolların rasgele seçilmeyip alt ağ kollarını izlemesi sağlanmıştır. Ayrıca ağ üzerindeki linklerin çift yönlü trafik taşıyabildiği varsayılmıştır. Alt ağ kolları ile, alt ağ merkezleri arasındaki yolların çakışması nedeni ile alt ağ merkezleri arasındaki yollar üç kısımdan oluşur: Başlangıç alt ağındaki yol, başlangıç alt ağı ile hedef alt ağ arasındaki bir link ve hedef alt ağdaki yol. Başlangıç alt ağındaki yol, üzerindeki trafiğin, merkezden çıkıp alt ağ düğümlerine doğru gitmesi nedeniyle, alt ağ kollarından birinin tümünün üzerinden geçmelidir; yani bir uç düğüm üzerinden geçmelidir. Eğer başlangıç alt ağındaki yolun bir alt ağ kolunun yalnızca bir kısmını kullanmasına izin verilirse, daha sonra bu alt ağ kolunda bulunan ve merkezler arası bağlantının kullanmadığı düğümlere erişilmek istenildiğinde bu işlemin aynı dalgaboyunda yapılması mümkün olmayacaktır; çünkü alt ağ kolunun ortasındaki bir düğüm başka bir alt ağdaki düğüme bağlı olacak ve ışık bölme kapasitesini tamamlamış olacaktır. Bu durumda da, diğer alt ağdaki düğüme bağlı olan düğümün, üzerinde bulunduğu alt ağ kolunda bulunan komşuna bağlanmaya çalışması bu düğümde dallanma olmasına ve düğümün ışık bölme kapasitesinin aşılmasına neden olacaktır.

Hedef alt ağdaki yolun ise, başlangıç alt ağında olduğu gibi, bütün bir alt ağ yolunu kat etmesi gerekmez; çünkü bu yolun taşıdığı trafik alt ağ merkezine doğrudur. Halbuki alt ağdaki herhangi bir düğüme erişmek için merkezden uç düğümlere doğru trafik akışı vardır. Ağ linklerinin çift yönlü olması nedeniyle hedef alt ağdaki yolun, bir alt yolunun herhangi bir düğümünden başlayıp bu kolu merkeze doğru kat etmesi yeterlidir. Alt ağlar arasındaki bağlantıyı kuracak linkin seçiminde ise bir kısıtlama yoktur.

İki alt ağ merkezi arasındaki bağlantının, başlangıç alt ağ kısmında bir alt ağ yolunun tamamının kullanılması gerekirken, hedef alt ağda bağlantı için bütün bir alt ağ kolunun kullanılmasına ihtiyaç olmaması, iki alt ağ merkezi arasındaki farklı yönlerdeki yolların farklılık göstermesine neden olmakta ve dolayısıyla alt ağlar

(1) Ağ içinde MC düğümleri L_1 listesine koy. Geriye kalan düğümleri L_d listesine koy.

(2) **for** (L_1 *içindeki k düğümleri*)

k düğümünü Ağ(k)' ya yerleştir.

(3) **while** (L_d *dolu*) **{**

(4) **if**(L_d*'de ağa tek linkle bağlı düğüm var*)

L_d' nin bu özellikteki ilk düğümü *d* 'yi seç.

(5) **else**

L_d' nin ilk elemanı *d* 'yi seç.

(6) **for** *(tüm Ağ (i)'ler*) **{**

(7) **for***(tüm Aday(i , j) 'ler*) **{**

if(*M(d, Aday(i , j))* < *M(d, Alt_Ağ_Aday(i))*)

Alt_Ağ_Aday(i) = Aday(i , j)

} *// **7 'nin sonu**.*

(8) **if**(*M(d, Alt_Ağ_Aday(i))* < *M(d, Ağ_Aday)*)

Ağ_Aday = Alt_Ağ_Aday(i)

} *// **6'nın sonu***

(9) Ağ_Aday(i) ile *d* arasındaki en kısa yolun tüm düğümlerini Ağ_Aday(i)'nin elemanı olduğu alt ağa ekle. Ayrıca bu düğümleri L_d 'den çıkart.

(10) Ağ_Aday(i)'yi elemanı olduğu alt ağın aday düğümleri arasından çıkar yerine *d* 'yi adaylar arasında koy. *d* 'yi L_d listesiden çıkart.

(11) Ağ_Aday(i) ile *d* arasındaki en düşük maliyetli yol üzerindeki tüm düğümlere (*d* hariç) çakışık tüm linkleri kopar.

} *//* **while sonu**

Şekil 5.3**:** Alt Ağ Oluşturma algoritmasının sözde kodu.

arasındaki bağlantılar bir yol çifti ile gösterilmektedir. İki alt ağ arasındaki farklı yönlerdeki bağlantıların simetrik olması, ancak her iki alt ağın, en az birer uç düğümünün birbirine doğrudan komşu olması ile mümkündür.

Alt ağ arasındaki bağlantılar bir yol çifti ile gösterilmelerine rağmen üst katmandan bakıldığında bu bağlantılar iki yönlü tek bir link olarak görülebilirler. Üst katmanda linkin hangi yönde kullanıldığına bağlı olarak alt katmanda bu linki gerçekleştirmek

için linkin yönüne ilişkin bağlantı yolu, alt ağlar arasındaki bağlantıyı oluşturmak için seçilir.

İki merkez arasındaki yolun başlangıç alt ağındaki kısmının hiçbir alt ağ kolunun tümünü kullanarak bir bağlantı kurması mümkün olmuyorsa, bu durumda alt ağın o anki uç düğümleri ayrı ayrı alt ağdan çıkarılarak alt ağın yeni uç düğümleri ile bağlantı kurulmaya çalışılır ve bir bağlantı kurulana kadar bu işlem devam eder. Bu şekilde bir bağlantı kurulduğunda yalnızca o bağlantı için alt ağdan çıkarılması gereken düğümler alt ağ dışına konulur. Ancak bu işlem düğümlerin alt ağ dışına çıkarılmasını gerektirdiği için zorunlu hallerde yapılır. Bu işlemin zorunlu olarak yapılmasını sağlamak üzere iki alt ağ arasındaki bağlantılara *komşuluk derecesi* adı verilen bir değer kullanılarak öncelikler atanmıştır. Komşuluk derecesinin değerini, iki alt ağ arasındaki bir bağlantı yolunun, bağlantının başladığı alt ağdan kaç tane düğümün çıkarılması gerektirdiği belirler. Dolayısıyla komşuluk derecesi derecesinin 0 olması iki alt ağ arasındaki bağlantı yolunun, bağlantının başladığı alt ağdaki bir alt ağ kolunun tamamını kullandığını ve diğer alt ağa bir uç düğüm üzerinden bağlandığını gösterir. Genelde istenilen 0 değerli komşuluk derecelerine sahip bağlantılar oluşturmaktır. Bu nedenle bu bağlantılar, her zaman komşuluk derecesi değeri daha yüksek olan bağlantılardan öncelikli tutulurlar. Ayrıca aynı komşuluk değerine sahip bağlantılar da bağlantıların maliyetine göre sıralanırlar ve seçim yapılırken düşük maliyetli bağlantılar tercih edilir.

Alt ağlar arasındaki bağlantıların kurulması için kullanılan algoritma yukarıda anlatılanları gerçekleştirmek için üç liste kullanır (Alt ağlar arasındaki bağlantıları kurma algoritmasının sözde kodu Şekil 5.4'te verilmiştir. Bu sözde kod içinde alt ağlar arasındaki bağlantıların kurulması işlemi *Alt_Ağları_Birbirine_Bağla()* fonksiyonu içinde yapılmaktadır. Bu fonksiyonun içinde kullanılan *Bağlı_Grafı_Guncelle()* fonksiyonunun iç yapısı da *Alt_Ağları_Birbirine_Bağla()* fonksiyonun hemen altında gösterilmiştir.). Bunlardan birincisi *bağlı alt ağlar* listesidir. Bu liste, algoritmanın belli bir adımında birbirlerine mümkün olan bütün yollar kullanılarak bağlanmış alt ağları içerir. Kullanılan ikinci liste ise *bağlantı listesi*dir. Bu liste bağlı alt ağlar listesindeki alt ağlardan, bu listede olmayan alt ağlara var olan iki yönlü bağlantıları içerir. Ancak bu listedeki bağlantılar, aralarında bir bağlantı bulunan bir alt ağ çifti için mümkün olan bağlantılardan en düşük komşuluk derecesi ve maliyet değerine sahip olanlarıdır. (Bir alt ağ çifti arasındaki iki yönlü bağlantının komşuluk derecesi, bu alt ağ çifti arasındaki tek yönlü iki bağlantının komşuluk derecesi değerleri toplamıdır. Benzer şekilde alt ağ çiftleri

Alt_Ağları_Birbirine_Bağla( ) {

(1) *Bağlı Alt Ağlar* listesine 0 no.lu alt ağı koy ;

(2) Bağlı_Grafı_Guncelle(0) ;

(3) **while** (*Bağlı alt ağ sayısı* < *Tüm alt ağ sayısı*) **{**

(4) **if**(*Acil_Bağlantılar_Listesi dolu*)

Acil_Bağlantılar_Listesi' nin ilk elemanını *Seçilen_Bağlantı* yap ;

(5) **else if** (*Bağlantı_Listesi dolu*)

Bağlantı_Listesi' nin ilk elemanını *Seçilen_Bağlantı* yap ;

else

Hatalı sonlanma *; // Tüm alt ağlar bağlanmadan yapılacak bağlantılar tükendi*

(6) *Seçilen_Bağlantı* 'yı kur ve bu bağlantı ile daha önce bağlı olan alt ağlara bağlanan *Seçilen_Alt_Ağ* 'ı, *Bağlı_Alt_Ağlar* listesine yerleştir ;

(7) Bağlı_Grafı_Guncelle(*Seçilen_Alt_Ağ*) ;

} *ll* **while sonu**

} *ll* **fonksiyon sonu**

Bağlı_Grafı_Guncelle(***Yeni_Bağlı_Alt_Ağ*** **) {**

(1) *Yeni_Bağlı_Alt_Ağ* 'ın komşusu olan alt ağları *Komşular Listesi*'ne koy ;

(2) **for** *(Komşular Listesi' nin tüm elemanları komşu 'lar*) **{**

(3) **if** (*komşu ! = Yeni_Bağlı_Alt_Ağ 'ın bağlantı kurduğu bağlı alt ağ*) **{**

(4) **if** (*komşu bağlı bir alt ağ)* **{**

Yeni_Bağlı_Alt_Ağ ile *komşu* arasında 0 komşuluk dereceli çift yönlü bağlantılar varsa bunlardan en düşük maliyetlisini kur ;

} ll (4) 'ün sonu

(5) **else {**

Yeni_Bağlı_Alt_Ağ ile *komşu* arasında en düşük komşuluk derecesi ve maliyete sahip olan çift yönlü bağlantıyı *Bağlantı_Listesi* 'ne koy;

} ll (5) 'in sonu

} ll (3) 'ün sonu

} ll (2) 'nin sonu

(6) *Yeni_Bağlı_Alt_Ağ* 'ın komşusu olan tüm bağlı alt ağların, *Bağlantı_Listesi* ve *Acil_Bağlantılar_Listesi* 'nde bulunan ve *Yeni_Bağlı_Alt_Ağ* ile olan bağlantılarını bu listelerden kaldır ;

} *ll* **fonksiyon sonu**

Şekil 5.4**:** Alt ağlar arası bağlantıları kurma algoritmasının sözde kodu.

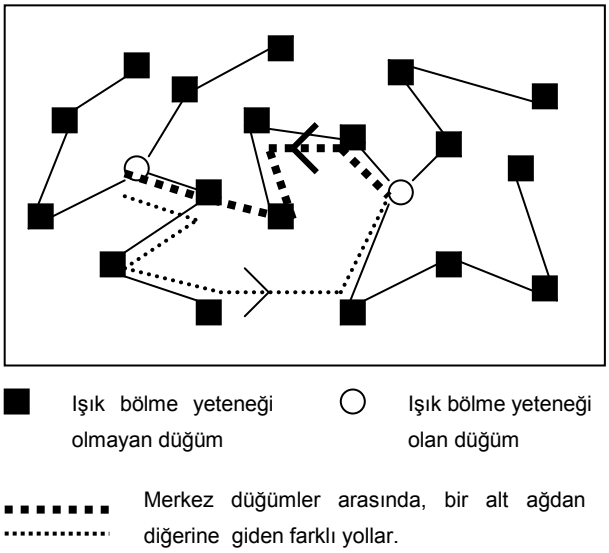
arasındaki çift yönlü bağlantıların maliyeti de bu alt ağlar arasındaki tek yönlü bağlantıların maliyetleri toplamdır. Bir tek yönlü bağlantının maliyeti ise bu bağlantıyı oluşturan link maliyetlerinin toplamı olarak alınmıştır.) Bağlantı listesindeki elemanlar da komşuluk derecelerine göre sıralı bir şekilde listede saklanır. Ayrıca aynı komşuluk derecesine sahip bağlantılar da kendi aralarında maliyetlerine göre sıralı bulunurlar.

Kullanılan listelerden sonuncusu *acil bağlantılar listesi*dir. Bu liste, bağlantı listesi ile aynı özellikte olan bağlantılar taşır; ancak bu listedeki her bir bağlantı, bağlantının birbirine bağladığı iki alt ağdan en az birinin, kendisi dışındaki bir alt ağ ile kurabildiği tek bağlantı olma özelliğini taşır. Bu özellikteki bağlantılar, bağlantı listesindeki bağlantılardan öncelikli olarak gerçekleştirilmesi gereken bağlantılardır; çünkü bir bağlantının kurulmasının ardından bazı alt ağ bağlantıları geçersiz hale gelebilir. Örneğin, bir alt ağın iki farklı alt ağa olan aynı yöndeki bağlantısı aynı alt ağ kolunu kullanıyorsa bu bağlantılardan birinin kurulması halinde diğer bağlantının kurulması mümkün olmayacaktır. Bu şekilde bağlantıların birbirlerini geçersiz kılması nedeniyle başka ağlarla tek bağlantısı olan alt ağların bağlantıları diğerlerinden önce kurulur.

Algoritmanın işleyişi ise şu şekilde olmaktadır: Algoritma her adımda, acil bağlantılar listesi boş değilse acil bağlantılar listesinin ilk elemanını, boş ise bağlantı listesindeki ilk elemanını seçer ve bu bağlantıyı kullanarak yeni bir alt ağı bağlı alt ağların birine bağlar. Bu işlemin sonunda yeni alt ağı da alt ağlar listesine ekler. Bağlı alt ağlar listesine eklenen bu yeni alt ağ, bu listedeki alt ağların yalnız birisi ile bağlantı kurmuştur. Bu nedenle yeni eklenen alt ağın bağlı ağlar kümesindeki diğer alt ağlar ile de bağlantısı olup olmadığına bakılır ve alt ağlar arasında her iki yönde de komşuluk derecesi 0 olan bağlantılar bulunursa bu durumda yeni alt ağ bu bağlantılar üzerinden diğer alt ağlara da bağlanır. Son olarak da bağlantı listesi güncelleştirilir. Bunun için, bağlantı listesinde yeni bağlanılan alt ağı içeren bağlantılar kaldırılır ve bunun yerine, bu alt ağın bağlı alt ağlar dışında kalan ağlar ile olan bağlantıları incelenir ve bu bağlantılardan uygun olanları bağlantı listesine eklenir.

Şekil 5.2'te gösterilen alt ağlar için alt ağlar arasında oluşturulacak bağlantılar Şekil 5.5'te gösterilmiştir. Görüldüğü gibi şeklin sağ tarafındaki alt ağ merkezinden sol tarafındakine kurulan bağlantı ile ters yönde kurulan bağlantı farklı iki yol izlemektedir. Sağ taraftaki alt ağdan sol taraftaki alt ağa, sağ alt ağların uç düğümleri üzerinden (yani 0 komşuluk derecesiyle) kurulabilecek tek yol bu iki alt ağın bağlanması için kullanılmıştır. Sol alt ağın ise bir uç düğümü üzerinden , sağ

alta bağlanmak için kullanabileceği iki farklı yol vardır (sol alt ağın şeklin tepesindeki uç düğümünden sağ alt ağa bağlantı kurması mümkündür); ancak şekil üzerinde gösterilen yolun seçimi bu yolun daha az maliyetli olduğunu gösterir.



Şekil 5.5: Alt ağ merkezleri arası yolların bulunması.

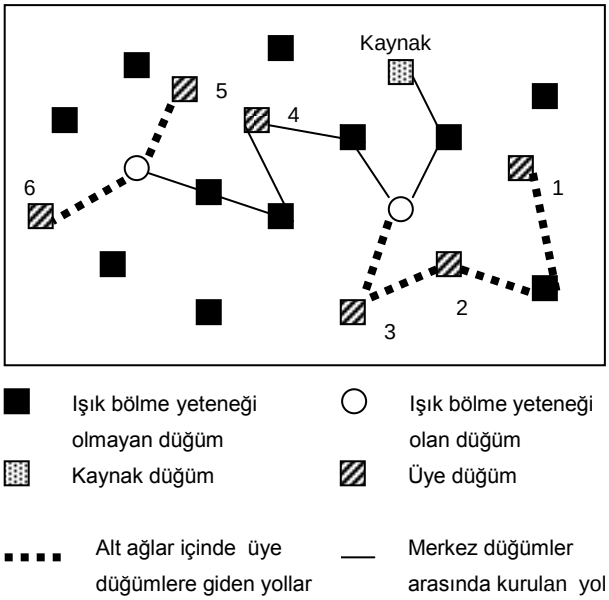
5.3.2 Algoritmanın İkinci Aşaması

Algoritmanın ikinci aşaması da birinci aşama gibi iki adımdan oluşur. Bu aşamada gerçekleştirilen iş, bir kaynak ve çoklu aktarım üyeleri grubu için ÇAA'nın oluşturulmasıdır. Bunun için bu aşamanın ilk adımında üst katmandaki ÇAA oluşturulur, ikinci adımda da bu ÇAA'nın alt katman düğümlerinden oluşan eşdeğeri elde edilir. Bu aşamanın ikinci adımı sonunda fiziksel ağ üzerinde kaynaktan çoklu aktarım üyelerine ulaşan bir ÇAA elde edilmiş olur.

5.3.2.1 Üst Katman ÇAA'sının Elde Edilmesi

Algoritmanın bu adımda önce fiziksel ağdaki kaynak ve üye düğümlerin hangi alt ağların elemanları oldukları belirlenir. Kaynağın elemanı olduğu alt ağ üst katmandaki kaynak alt ağ, üyelerin elemanı oldukları alt ağlar da üst katmandaki üye alt ağlar haline getirilir. Bir başka deyişle üst katman grafında bir kaynak ve çoklu aktarım üyeleri grubu oluşturulur. Bundan sonra üst katman grafı için elektronik ağlar için gerçeklenmiş herhangi bir ÇAA oluşturma algoritması yürütülerek, üst katmandaki kaynak ve çoklu aktarım üyeleri için bir ÇAA oluşturulur.

Şekil 5.2'deki alt ağları kullanarak, Şekil 5.6'da gösterilen kaynak ve üyeler için üst katman grafının elde edilişi yukarıdaki açıklamalara göre şu şekilde yapılır: Kaynak şekillerin sağ tarafında bulunan alt ağ üzerinde bulunduğu için üst katman grafında kaynak alt ağ şekillerin sağ tarafındaki alt ağdır. Bu örnekte, yalnız iki alt ağ olduğu için üst katman grafında yalnız iki düğüm vardır. Şekil 5.6'da her iki alt ağda da üye düğümler olduğu için, sol alt ağ da üst katman grafındaki üye alt ağdır (diğer alt ağ zaten kaynak alt ağdır.) Buna göre üst katmanda çoklu aktarım ağacı kökü sağ alt ağ, yaprağı sol alt ağ olan iki düğümlü bir ağaç olacaktır.



Şekil 5.6: Verilen bir çoklu aktarım grubu için oluşturulan çoklu aktarım ağacı.

5.3.2.2 Alt Katman ÇAA'sının Elde Edilmesi

Bu adımda algoritmanın ikinci aşamasının ilk adımında elde edilen ve üst katman için oluşturulmuş olan üst katman ÇAA kullanılır. İlk olarak üst katman ÇAA 'nın düğümleri arasındaki linklere karşı düşen alt ağlar arası yollar belirlenir ve bu yollar alt katman ÇAA 'ya konulur. Bu işlem yapılırken, üst katman linklerine karşılık hangi alt ağ bağlantı yollarının kullanılması gerektiği belirlenmelidir. Bunun için, daha önce Alt Ağlar Arasındaki Bağlantıların Oluşturulması kısmında anlatıldığı gibi üst katmanda linkin hangi yönde kullanıldığının anlaşılması yeterlidir. ÇAA 'da trafik akışı kaynaktan merkeze doğru olacağından linklerdeki trafik akışı da ağaç üzerindeki bir düğümden bu düğümün çocuğuna doğru olacaktır. Trafik akış yönü, üst katman ÇAA'daki düğümlerin anne-çocuk ilişkileri kullanılarak belirlendikten

sonra hangi üst katman linkleri için hangi alt ağlar arasındaki bağlantı yollarının hangilerinin kullanılacağı belirlenmiş olur.

Alt katmanda alt ağ merkezleri arasındaki yollar, alt katman ağacına konulduktan sonra, alt katmandaki ÇAA'nın tamamlanması için kaynak düğüm ve üye düğümlerin elemanı oldukları alt ağın merkezlerine bağlanmaları gerekir. Her alt ağ düğümünün alt ağın hangi kolu üzerinde olduğu belli olduğundan kaynak ve üye düğümlere erişmek için alt ağın merkezine bu düğümlerin üzerinde bulundukları kollar eklenir. Ancak kaynak ve üye düğümler üzerinde bulundukları alt ağ kolunun uç düğümleri değillerse, bu kolların yalnızca merkez ile ÇAA 'ya eklenecek düğüm arasındaki kısmının ÇAA'ya eklenmesi yeterlidir; kolun tamamının ağaca eklenmesine gerek yoktur. Kaynak düğümün üzerinde bulunduğu alt ağ kolu (veya bu kolun kaynak ile alt ağ merkezi arasındaki kısmı) alt katman ÇAA 'ya eklenirken, üye düğümlere ait alt ağ kollarından farklı olarak, kaynağın ağacın kökü olmasını sağlanır.

Şekil 5.6'da aynı şekil üzerinde gösterilen kaynak ve üye düğümler için elde edilen sonuç ÇAA gösterilmiştir. Algoritmanın Üst Katman ÇAA 'nın elde edilmesi kısmında bu örnek için, üst katman ÇAA'nın, sağ alt ağdan başlayıp sol alt ağda biten bir ağaç olduğu anlatılmıştı. Bu ağaçtan yola çıkarak yukarıdaki yöntem kullanılırsa, alt ağlar arasındaki bağlantı Şekil 5.6'da iki merkez arasında düz çizgilerle gösterilen yol olarak bulunur. Dikkat edilirse, bu yol sağ alt ağdan sol alt ağa gitmek için oluşturulmuş alt ağ bağlantısıdır. Üst katman ÇAA 'da da kök sağ alt ağ olduğu için alt ağları birleştirmek için bu yol seçilmiştir.Şekil 5.6'da alt katman ağacına kaynak düğümün eklendiği yol düz çizgilerle ve üye düğümlerin eklendiği yollar da kesikli çizgilerle gösterilmiştir. Burada da kaynak ve üyelerin elemanı oldukları alt ağ merkezinden çıkan alt ağ kolu üzerinden merkez düğüme birleştikleri görülmektedir.

5.3.3 Alt Ağ Elemanı Olmayan Düğümlere Erişim

Her ne kadar, iki katmanlı kapsayan ağaç yöntemi, birinci aşamasında tüm ağ düğümlerine erişecek biçimde bir kapsayan ağaç oluşturmaya çalışsa da bu her zaman mümkün olmayabilir ve bazı düğümler kapsayan ağacın dışında kalabilirler. Yalnızca ağın belli düğümlerine, mümkün olduğunca az dalgaboyu kullanarak erişmek için oluşturulmuş WDM çoklu aktarım algoritmalarının bile tek bir dalgaboyu kullanarak bütün üyelere erişemediği durumlar olduğundan, tüm düğümleri kapsayan bir ağacın da, her ağ için mümkün olmaması doğal görünmektedir. Bu nedenle iki katmanlı kapsayan ağaç yönteminin alt ağ elemanı olmayan düğümlere erişim için de önlemler alması gerekmektedir.

Hiçbir alt ağın elemanı olmayan üye düğümlere erişebilmek üzere yöntem, şu yolu izler : Öncelikle alt ağ elemanı olan tüm düğümler için, daha önce anlatılan iki aşamadan geçilerek, ÇAA oluşturulur. Ardından hiçbir alt ağın elemanı olmayan düğümlere erişebilmek üzere, oluşturulan ÇAA'nın MC olan tüm düğümlerinden (yani yaprak düğümler ve eğer kullanılmayan alt ağ kolları varsa alt ağ merkezlerinden), alt ağ elemanı olmayan düğümlere erişilip erişilemeyeceğini kontrol eder. Eğer hiçbir alt ağ elemanı olmayan düğümlere bu şeklide yol bulunabiliyorsa, bu düğümler ağaca bağlanabilecekleri en düşük maliyetli yollar üzerinden ÇAA 'ya bağlanırlar. Aksi taktirde ise ikinci bir dalgaboyunu kullanan bir ağaca ihtiyaç duyulacaktır. Bu ağacın oluşturulması için ise daha önce önerilmiş WDM çoklu aktarım algoritmalarından bir tanesinin kullanılması mümkündür. Sonuç olarak iki katmanlı kapsayan ağacın erişemediği düğümlerin sayısı oldukça az olacağından bu algoritmalar da hızlı bir şekilde bu ağacı oluşturabilirler. (Tabii gerektiğinde bu algoritmalar da birden fazla ağaç oluşturabilirler.)

Hiçbir alt ağın elemanı olmayan bir kaynak düğümün varlığı halinde ise yöntem kaynak alt ağı belirleyebilmek üzere öncelikle kaynak düğümü bir alt ağa bağlamak durumundadır. Bunun için izlenen yol ise kaynak düğümün, kendisine en düşük maliyet ile erişebilen alt ağ merkezine bağlanması ve kaynak düğümün bağlandığı bu alt ağın da kaynak alt ağ yapılmasıdır.

6. SONUÇLAR VE TARTIŞMA

Bu çalışmada, ortaya atılan İki Katmanlı Kapsayan Ağaç yönteminin başarım analizinin yapılması amaçlanmıştır. Bunun için, bu yöntem ve Bölüm 4'te anlatılan Üye öncelikli, Üye Yönelimli ve Evlat Edinmeli adlı üç adet statik ÇAA oluşturma algoritması, C++ programlama dili kullanılarak kodlanmış ve algoritmaların sonuçları karşılaştırılmıştır.

Karşılaştırma için, statik çoklu aktarım ağacı oluşturma algoritmalarının seçilmesinin sebebi, bu algoritmaların ağın tüm düğümlerinin komşuluklarını ve tüm linklerin maliyetlerini bilerek sonuç oluşturmalarıdır. Ağ hakkında bütün bu bilgileri bilerek oluşturulacak bu sonuçlar, en iyiye oldukça yakın olacaklardır. Burada önerilen yöntemin başarımı inceleneceğinden, yöntemin sonuçlarının, iyi olması beklenen diğer sonuçlarla karşılaştırılması daha uygun olmaktadır.

Algoritmaların sonuçlarının, karşılaştırma için daha anlamlı olması açısından mümkün olduğunca çok giriş parametresinin aynı yapılmasına çalışılmıştır. Örneğin, Üye Öncelikli ÇAA oluşturma algoritması, linklerin maliyetlerini eşit kabul ederek işlem yaptığı için, diğer bütün algoritmalar da link maliyetlerinin eşit olduğu düşünülerek yürütülmüştür. Ayrıca, Evlat Edinmeli çoklu aktarım algoritması giriş parametresi olarak aldığı IP ÇAA'nın, Üye Yönelimli çoklu aktarım ağacı oluşturma algoritması ile aynı prensibi kullanarak oluşturulması sağlanmıştır. Bunun için, IP çoklu aktarım ağacı oluşturma algoritması olarak Şekil 4.2 'de sözde kodu verilen algoritma kullanılmıştır.

Algoritmalara giriş olarak verilen parametreler ise şunlardır: Düğüm sayısı belli bir ağ topolojisi, bu ağ topolojisi üzerinde rasgele seçilmiş belli sayıda MC düğüm ve yine aynı ağ topolojisi üzerinde rasgele seçilmiş bir kaynak ve belli sayıda çoklu aktarım üyesi. Bu giriş parametrelerinden oluşan her farklı durum, adlandırılmada kolaylık olması açısından *senaryo* olarak isimlendirilmiştir.

Çalışmada, algoritmaların oluşturdukların sonuçların karşılaştırılması için şu değerler incelenmiştir: Elde edilen çoklu aktarım ağacının üzerindeki link sayısı, çoklu aktarım ağacının kaynağına en uzak üye düğümün kaynağa sekme sayısı

cinsinden uzaklığı (maksimum gecikme) , kaynağın tüm üye düğümlere olan uzaklıklarının sekme sayısı cinsinden ortalaması (ortalama gecikme) ve algoritmaların yürütme süreleri.

Elde edilen ağaçların toplam link sayıları, ağ kaynaklarının ne kadar elverişli kullanıldığını anlamak üzere kullanılırken, ağaç üzerindeki kaynak ve üye düğümlerin sekme sayısı cinsinden uzaklıkları ise ağaç üzerindeki iletişim gecikmesini belirlemek için kullanılmıştır. Algoritmaların yürütme süreleri de karmaşıklıklarını ölçmek üzere kaydedilmiştir.

Algoritmaların yürütme sürelerinin ölçüldüğü ortam, Windows XP Professional İşletim sistemini kullanan Pentium 4, 1.7 GHz işlemcili bir bilgisayardır.

6.1 Sonuçlar ve Yorumlar

Bu çalışmada algoritmaların başarımlarının denenmesi amacıyla 36 farklı senaryo denenmiştir. Senaryolarda, 4 farklı topoloji üzerinde kullanılmıştır ve bu topolojilerin düğüm sayıları sırasıyla 20, 30, 40 ve 50'dir. Her bir topoloji için MC düğümlerin sayısının tüm düğümlerin sayısına oranı (bundan sonra bu değer *MC oranı* olarak adlandırılacaktır.) %10, %30 ve %50 değerleri için incelenmiştir. MC oranı için %50 değeri üzerine çıkılmamıştır; çünkü %50 değeri ve üzerinde, bir düğümün MC olma olasılığı oldukça yüksektir. Bu durumda, algoritmalar dallanma yapılabilecek düğüm aramak için fazla zaman kaybetmeyeceklerdir. Ayrıca %50 değeri üzerinde algoritma başarımlarının fazla değişmeyeceği de gösterilmiştir [8].

Yapılan denemelerdeki senaryolar oluşturulurken, her topoloji için belli bir MC oranına sahip ağda, üye düğümlerin sayısının tüm düğümlerin sayısına oranı (bundan sonra bu değer *üye düğüm oranı* olarak adlandırılacaktır.) %10 , %50 ve %90 değerleri alacak şekilde rasgele çoklu aktarım grupları oluşturulmuştur.

Denemelerde kullanılan topolojiler şu modele göre oluşturulmuştur: Öncelikle düğümler sınırlı bir düzlem üzerinde rasgele konumlara yerleştirilir. Ardından her düğüm çifti arasına α olasılık değerine göre bir link yerleştirilir. Bu çalışmadaki denemelerde oluşturulan her bir topoloji için α = 0.1 olarak seçilmiştir.

Yapılan denemeler sonucunda elde edilen sonuçlar Ek.A'da grafikler halinde verilmiştir. Grafiklerde ağdaki düğüm sayının ve üye düğüm oranının çıkış değerlerine etkisi incelenmiştir.

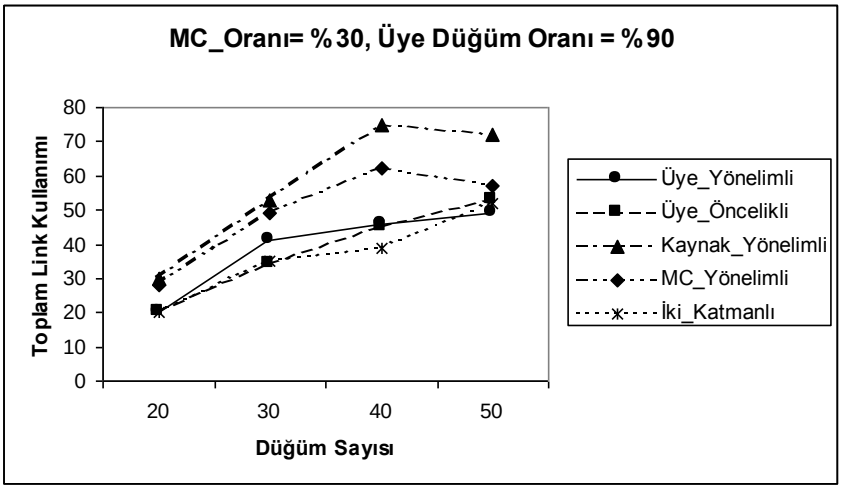
6.1.1 Ağdaki Link Kullanımı

Link kullanımı ile ilgili elde edilen sonuçlar Şekil A.1 ile Şekil A.21 arasında gösterilmiştir. Bu sonuçlarda, genellikle Üye Yönelimli algoritmanın en düşük link kullanımına sahip olduğu görülmektedir. Bu durum zaten beklenen bir sonuçtur; çünkü algoritmalar arasında en yoğun arama işlemi yapanı Üye Yönelimli algoritmadır.

Üye Öncelikli algoritma, Üye Yönelimli algoritma ile karşılaştırıldığında, genelde iki algoritmanın da link kullanımının birbirine yakın olduğu görülmektedir. Ancak, Üye Öncelikli algoritma çoğunlukla Üye Yönelimliye göre daha fazla link kullanmaktadır.

Evlat Edinmeli algoritmanın her iki yaklaşımı da, eğer elde ettikleri sonuç ağaç IP ÇAA ile aynı ağaç veya çok yakın bir ağaç ise, bu algoritmaların link kullanımı da Üye Yönelimli algoritmanın değerlerine yaklaşmaktadır. Ancak, IP ÇAA'dan sapma olursa, ki genel durum budur, link kullanımı artmaya başlamaktadır. Ayrıca Kaynak-Yönelimli ve MC-Yönelimli yaklaşımların her ikisi de ikinci bir ağaç oluşturduğunda ilk ağaç ile aynı linkleri kullanma olasılıkları yüksek oldukları için link kullanımı gittikçe kötüleşebilmektedir. Bunun yanında, Kaynak-Yönelimli yaklaşım genelde kaynak ile üyeler arasındaki en düşük maliyetli yolları seçmeyi tercih ettiğinden çok sayıda ağaç oluşmasına ve dolayısıyla link kullanımının oldukça yükselmesine neden olmaktadır. Buna karşın, fazla ağaç kullanmak yerine aynı ağaç üzerine mümkün olduğunca fazla üye eklemeye çalıştığı için MC-Yönelimli yaklaşımın Kaynak-Yönelimli yaklaşımdan daha iyi bir link kullanımına sahip olduğu görülmektedir.

İki Katmanlı Kapsayan Ağaç algoritmasının ise link kullanımının genellikle diğer tüm algoritmalardan fazla olduğu ancak üye düğüm oranının artması ile link kullanımının Üye Yönelimliye yaklaştığı görülmektedir (bkz. Şekil 6.1 ve Şekil A.16, Şekil A.17, Şekil A.18.). Bu algoritmanın link kullanımının fazla olması beklenen bir sonuçtur, çünkü üye düğümler ile kaynak arasındaki yollar, en kısa yollar olmasalar bile, merkez düğümleri birleştiren yollar üzerinden geçmek durumundadır. Ayrıca üye düğüm oranının artışı ile İki Katmanlı Kapsayan Ağaç algoritmasının link kullanımının Üye Yönelimli algoritmaya yaklaşması da beklenilebilecek bir sonuçtur; çünkü İki Katmanlı Kapsayan Ağaç algoritması, üyelere erişmek üzere mümkün olduğunca çok düğümü içeren bir kapsayan ağaç kullanır ve üye düğüm oranı arttığında bu kapsayan ağacın büyük kısmı üyelere erişmek için kullanılacaktır; yani



Şekil 6.1: Toplam link kullanımının düğüm sayısına göre değişimi

kapsayan ağaç verimli bir biçimde kullanılmış olacaktır. Bu sayede de, link kullanımı Üye Yönelimli algoritmanınkine yaklaşmış olacaktır.

6.1.2 Dalgaboyu Kullanımı

Algoritmaların dalgaboyu kullanımına ilişkin başarımlarını gösteren grafikler Şekil A.22 ile Şekil A.42 arasında yer almaktadır. Şekillerden, genellikle Kaynak-Yönelimli ve MC-Yönelimli yaklaşımların dalgaboyu kullanımının fazla olduğu ve diğer algoritmaların dalga boyu kullanımlarının birbirlerine yakın olduğu görülmektedir. Bu da beklenilen bir sonuçtur; çünkü Kaynak-Yönelimli ve MC-Yönelimli yaklaşımlar, MC düğümleri göz ardı ederek kurulan bir ağacı varolan ağa uyarlamaya çalışırlar ve bunu yaparken de kendilerine giriş olarak verilen ağacı mümkün olduğunca az değiştirmeye çalışırlar. Bu durum özellikle Kaynak-Yönelimli yaklaşımda daha da açık olarak ortaya çıkmaktadır. Kaynak-Yönelimli algoritma MC-Yönelimli yaklaşıma göre çok daha az arama yaparak başka bir ağaca geçme kararı verdiğinden genellikle dalgaboyu kullanımı en yüksek olan algoritma olarak karşımıza çıkmaktadır.

Önerilen algoritmanın dalgaboyu kullanımı genellikle Üye Yönelimli ve Üye Öncelikli algoritmalarınkine yakındır ve hepsinin kullandıkları dalgaboyu sayısı genellike 1 ile 3 değerleri arasında değişmektedir. İlk bakışta, önerilen algoritmanın aslında dalgaboyu kullanımının diğer algoritmalardan daha iyi olması beklenebilir; çünkü önerilen algoritma bir kapsayan ağaç oluşturmaya çalışmakta ve birçok üye düğüme bu ağaç üzerinden erişebilmektedir. Ancak her durumda tüm düğümleri kapsayan

bir ağaç oluşturulamadığı için iki katmanlı kapsayan ağaç tarafından kapsanamayan düğümlere başka bir dalgaboyunda erişilmesi gerekebilmektedir. Bunun için de diğer algoritmalar kullanıldığından dalgaboyu kullanımı Üye Yönelimli ve Üye Öncelikli algoritmalara yaklaşmaktadır.

6.1.3 Ortalama ve Maksimum Gecikme

Algoritmaların Ortalama Gecikme ile ilgili başarımlarını gösteren grafikler Şekil A.43 ile Şekil A.63 arasında, Maksimum Gecikme ile ilgili grafikler ise Şekil A.64 ile Şekil A.84 arasında yer almaktadır. Bu sonuçlardan, algoritmaların ortalama gecikme ve maksimum gecikme değerlerinin genellikle, en küçüğü Kaynak-Yönelimli yaklaşım, en büyüğü de İki Katmanlı Kapsayan Ağaç olmak üzere şu sırayı izlediği görülmektedir: Kaynak-Yönelimli yaklaşım, MC-Yönelimli yaklaşım, Üye Öncelikli , Üye Yönelimli ve İki Katmanlı Kapsayan Ağaç.

Kaynak-Yönelimli yaklaşımın en düşük ortalama gecikme ve en düşük maksimum gecikmeyi vermesi beklenen bir sonuçtur; çünkü bu algoritma genellikle kaynak ile üyeler arasındaki en kısa yolları tercih etmektedir. Bu algoritmanın bu yüzden çok fazla sayıda ağaç oluşturduğu daha önce açıklanmıştı. MC-Yönelimli yaklaşım için ise, ortalama gecikme ve maksimum gecikme değerlerinin Kaynak-Yönelimli yaklaşımınkine yakın çıkması her iki algoritmaya da giriş olarak verilen ağaçların aynı olması şeklinde yorumlanabilir; çünkü bu algoritmalar kendilerine giriş olarak verilen ve bir IP ÇAA algoritması ile oluşturulmuş olan ÇAA 'yı değiştirerek sonuca ulaşırlar. Kaynak-Yönelimli yaklaşım bu işlemi yaparken giriş ÇAA 'yı çok daha az değiştirirken, MC-Yönelimli yaklaşım daha çok arama yaparak giriş ÇAA üzerinde daha fazla değişiklik yapmaktadır.

Üye Yönelimli ve Üye Öncelikli algoritmalar, düğümlerin MC olup olmadıklarına bakarak ÇAA 'yı baştan oluşturdukları ve bir ağaca mümkün olduğunca çok üyeyi dahil ettikleri için Kaynak-Yönelimli ve MC-Yönelimli yaklaşımlara göre ortalama ve maksimum gecikmelerinin daha yüksek çıkması beklenebilecek bir sonuçtur.

İki Katmanlı Kapsayan Ağaç algoritmasının ise, ortalama ve maksimum gecikmesinin diğer bütün algoritmalardan fazla çıkması, yine beklenen bir sonuçtur. Bunun nedeni yine, kaynak ile üyeler arasındaki yolların, en kısa yollar olmasalar da, merkez düğümler arasındaki yollar üzerinden geçmesi gerekliliğidir.

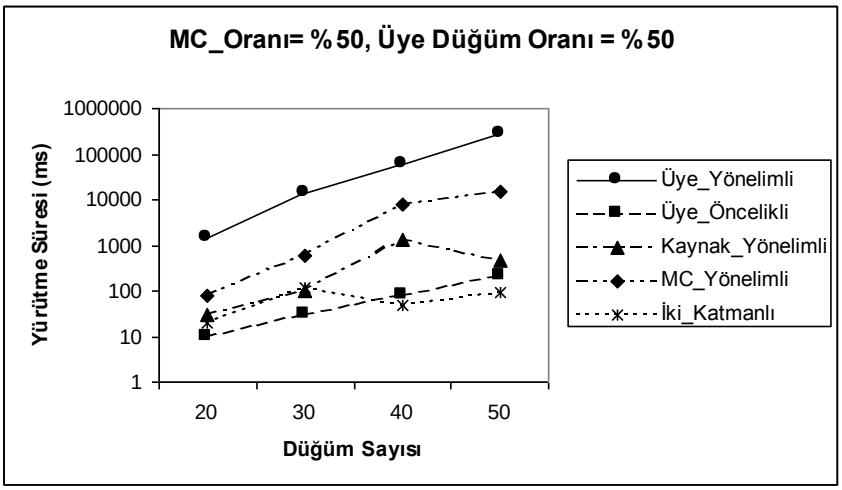
6.1.4 Yürütme Süreleri

Algoritmalar arasında yürütme sürelerine ilişkin başarımlarını gösteren grafikler Şekil A.85 ile Şekil A.105 arasında yer almaktadır. Bu sonuçlara bakıldığında ilk görülen, Üye Yönelimli algoritmanın yürütme süresinin, düğüm sayının ve üye düğüm oranının artışı ile üstele yakın biçimde artmasıdır. (Bu nedenle Şekil A.85 ile Şekil A.105 arasındaki şekiller için yürütme süresi logaritmik olarak ölçeklenmiştir.) Bu durum da beklenen bir sonuçtur; çünkü düğüm sayısının veya üye düğüm oranının artışı bu algoritmanın yapması gereken aramaları büyük oranda arttırmaktadır.

Üye Yönelimli algoritma dışındaki diğer algoritmaların yürütme zamanlarına bakıldığında ise bu algoritmalar ile Üye yönelimli algoritma arasında oldukça büyük farklar gözlenmektedir. Bu durumun nedeni, bu algoritmanın bir üye düğümü ağaca bağlarken birçok düğümden bu düğüme en düşük maliyetli yolların hesaplamasıdır. İki düğüm arasındaki en düşük maliyetli yolun hesabı, karmaşıklığı yüksek bir işlem olduğundan, bu hesabı çok sık kullanan bu algoritmanın yürütme süresi önemli ölçüde uzamaktadır. Diğer algoritmalardan ise yalnız Üye Öncelikli algoritması en düşük maliyetli yol hesabını hiç kullanmamaktadır, öte yandan geriye kalan algoritmalar da en düşük maliyetli yol hesabını Üye Yönelimli kadar sık kullanmamaktadır; bu yüzden de yürütme süreleri oldukça farklı bulunmuştur.

MC-Yönelimli yaklaşım, Üye Yönelimli algoritmaya göre daha kısa sürmek ile beraber geriye kalan algoritmaların hepsinden uzun bir yürütme süresine sahiptir ve zaman zaman yürütme süresi düğüm sayısının ve üye düğüm oranının artışı ile üstele yakın artış göstermektedir. Bunun nedeni ise, MC-Yönelimli yaklaşımın başlangıçta giriş olarak verilen IP ÇAA 'nın düğümleri kullanılamadığında en düşük maliyetli yol hesabı yapmasıdır. Kaynak Yönelimli yaklaşımda da başlangıçta giriş olarak verilen IP ÇAA 'nın düğümleri kullanılamadığında, en düşük maliyetli yol hesabı yapılması gerekmektedir; ancak bu hesap oluşturulan ağaç üzerindeki tüm MC düğümler için değil de, yalnızca kaynağa doğru en düşük maliyetli yol üzerindeki MC düğümler için yapılır, dolayısıyla hesap yükü MC-Yönelimli 'deki kadar yoğun olmaz.

Üye Öncelikli ve İki Katmanlı Kapsayan Ağaç algoritmaları diğer algoritmalara göre çok daha kısa sürelerde sonlanabilmektedir. Her iki algoritma da düğüm sayısının 50 olduğu durumda bile (deney ortamında) 500 ms altında sürelerde sonlanırken diğerlerinin yürütülmesi saniyeler almaktadır. Ayrıca düğüm sayısı arttığında İki Katmanlı Kapsayan Ağaç algoritmasının yürütme süresi, Üye öncelikli algoritmanın



Şekil 6.2: Yürütme süresinin düğüm sayısına göre değişimi.

altına düşmektedir (bkz. Şekil 6.2 ve Şekil A.103, Şekil A.104, Şekil A.105). Bu durum, beklenen bir sonuçtur; çünkü İki Katmanlı Kapsayan Ağaç algoritması için çalışacak ÇAA oluşturma algoritması Üye Öncelikli algoritmaya göre daha az düğüm üzerinde (MC düğüm sayısı kadar) çalışmaktadır.

İki Katmanlı Kapsayan Ağaç algoritması, kapsayan ağaç üzerinden tüm üyelere erişemediği durumlarda üyeleri ağaca bağlamak için ek olarak zaman harcandığından algoritmanın yürütme süresi uzayabilmektedir. Böyle bir durumda algoritmanın toplam yürütme süresi saniyeler mertebesine çıkabilmektedir. Bunun nedeni ise kapsayan ağaç ile erişilemeyecek üye düğümlerin varolması halinde, bu üye düğümlere, varolan ağacın hala MC olan düğümleri üzerinden en düşük maliyetle erişilmeye çalışılmasıdır. Ayrıca MC olan düğümler üzerinden erişilemeyen düğümler varsa o düğümlere de Üye Öncelikli algoritması kullanılarak ikinci (veya daha fazla) bir dalgaboyunda erişilmektedir. Ancak Üye Öncelikli algoritmanın yürütme süresi kısa olduğu ve erişilmesi gereken üye düğüm sayısı da az olduğundan farklı dalgaboylarındaki ağaçların oluşturulması için harcanan süre, ilk dalgaboyundaki MC düğümlerden erişilemeyen üye düğümlere en düşük maliyetle erişmek için yapılan hesabın süresi yanında küçük kalmaktadır. Bu çalışmada, iki Katmanlı Kapsayan Ağaç algoritmasının bu şekilde gerçeklenmesinin nedeni, mümkün olduğu kadar az sayıda dalgaboyu kullanmak istenilmesidir. Ancak algoritmanın yürütme süresi dalgaboyu sayısından daha öncelikli bir parametre ise kapsayan ağaç üzerinde erişilemeyen düğümler olduğunda, doğrudan Üye Öncelikli algoritmanın yürütülmesi daha uygun bir çözüm olur.

6.2 Sonuç ve Öneriler

Bu çalışmada, İki Katmanlı Kapsayan Ağaç algoritmasının başarımının yürütme zamanı açısından Üye Öncelikli algoritmanınkine oldukça yakın olduğu, hatta üye düğüm sayısının artması durumda daha iyi olduğu ortaya çıkmıştır. Buna karşın, algoritmanın link kullanımı diğer algoritmalara göre daha fazladır ve buna bağlı olarak üye düğümlere olan gecikmesi daha uzundur. Ancak, İki Katmanlı Kapsayan Ağaç algoritması, dinamik çoklu aktarım grupları (yani çoklu aktarım üyelerinin zaman içinde gruba eklenebildiği ve çıkarılabildiği aktarım grupları) için kullanılmaya daha elverişlidir. Diğer algoritmaların hepsi, çoklu aktarım grubunda bir değişiklik olduğunda baştan yürütülürken, İki Katmanlı Kapsayan Ağaç algoritmasının baştan yürütülmesi gerekmez; ÇAA 'nın güncellenmesi, elektronik ağlar için kullanılan dinamik çoklu aktarım yöntemlerinden biriyle kolayca gerçeklenebilir.

İki Katmanlı Kapsayan Ağaç algoritmasının, dinamik çoklu aktarım grupları ile çalışabilmesi için, algoritma şu şekilde geliştirilmesi öngörülmektedir:

i) Bir alt ağ düğümü, bir çoklu aktarım grubuna katılacağında, elemanı olduğu alt ağ kolu üzerinden alt ağın merkezine bağlanacak; düğümün gruptan ayrılması benzer şekilde gerçeklenecektir.

ii) Ağdaki tüm düğümlere sadece İki Katmanlı Kapsayan Ağaç algoritması kullanılarak erişilmesi hedeflenirse, bütün düğümleri kapsayacak şekilde gerekli sayıda kısmi kapsayan ağaç oluşturulup, bir çoklu aktarım grubuna erişmek için birden fazla kısmi kapsayan ağaç (her biri farklı bir dalgaboyunda olmak üzere) beraberce kullanılacaktır.

Sonuç olarak, bu çalışmada "İki Katmanlı Kapsayan Ağaç Algoritması" adıyla dalga boyu atamalı WDM ağları için yeni bir dinamik çoklu aktarım algoritması tasarlanmış, gerçeklenmiş ve başarıyla denenmiştir. Yapılan karşılaştırmalarda önerilen algoritmanın yürütme süresinin statik algoritmalar arasında başarımı en yüksek olduğu ölçülen Üye Öncelikli algoritmanın yürütme süresine yakın hızlarda olduğu da saptanmıştır.

KAYNAKLAR

[1] **Wittman R., Zitterbart M ,** 2001. Multicast Communication , Protocols and Applications, Morgan Kaufmann Pubslishers, San Francisco.

[2] **Wang. B., Hou J. C.,** 2000. Multicast Routing and its QoS Extension: Problems, Algorithms and Protocols, IEEE Network Vol 14, No. 1 p. 22-36.

[3] **Zhang, X., Wei, J. ve Qiao, C.,** 2000. Constrained multicast routing in WDM networks with sparse light splitting, *INFOCOM'2000* , 1781-1790.

[4] **Green, P.E.** , 1996 . Optical Networking Update, *IEEE Journal of Selected Areas in Commmunications*, **14**, No.5, 764 – 779.

[5] **Zhang, X. , Wei, J. ve Qiao, C. ,** 1999 . On fundamental issues in IP over WDM multicast, *IC3N'99*, 84-90.

[6] **Zhang, Z. ve Fu, J.**, 2001. Ligtpath Routing for intelligent Optical Networks , *IEEE Network*, **15**, No. 4, 28 - 35

[7] **Koçyiğit, A. , Gökışık, D. ve Bilgen, S.,** 2001. All – Optical Networking , *Turk J. Elec. Engin.*, **9**, No. 2, 69 – 121.

[8] **Malli, R., Zhang, X. ve Qiao, C.,** 1998. Benefit of Multicasting in All-Optical Networks, SPIE Proc. Of Conf. All-Optical Networking , **3531** , 209 - 220.

[9] **Qiao, C. ve Yoo, M.,** 2000. Choices, Features and Issues in Optical Burst Switching, Optical Networks Magazine , **1**, No. 2, 36 – 44.

[10] **Ramamirtham, J. ve Turner, J.**, 2002. Design of Wavelenght Converting Switches for Optical Burst Switching, Proceedings IEEE INFOCOM' 2002.

[11] **Rosen, E., Viswanathan, A., Callon R. ,** 2001. Multiprotocol Label Switching Architecture, *RFC 3031*.

[12] **Huitema, C. ,** 1995. Routing in the Internet , Prentice Hall PTR, New Jersey.

[13] **Williamson B.,** 2000. Developing IP Multicast Networks Volume 1, Cisco Press, Indianapolis.

[14] **Kurose J. F., Ross K. W. ,** 2001. Computer Networking, A Top-Down Approach Featuring the Internet, Addison Wesley.

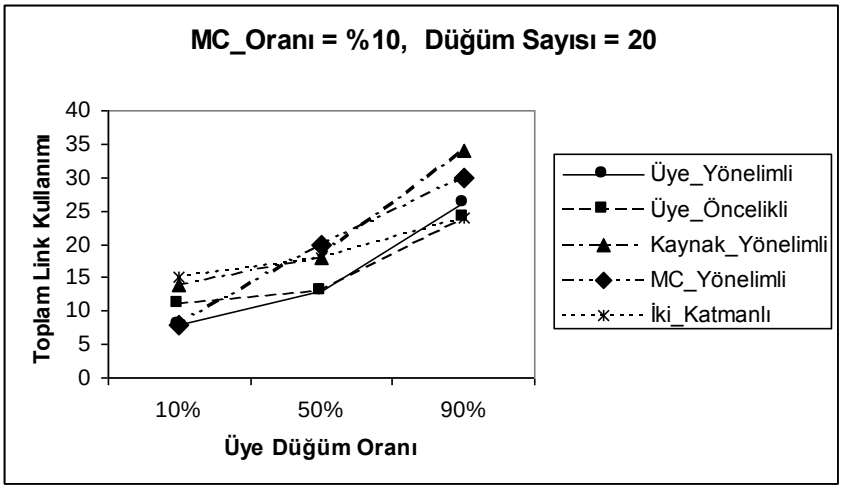
[15] **Almeroth K. C.,** 2000. The Evolution Of Multicast: From the MBone to InterDomain Multicast to Internet2 Deployment., IEEE Network Vol 14, No. 1, 10-20.

[16] **Sahasrabuddhe L. H., Mukherjee B.,** , 2000. Multicast Routing Algorithms and Protocols : A Tutorial. IEEE Network, Vol 14, No. 1, 90-102.

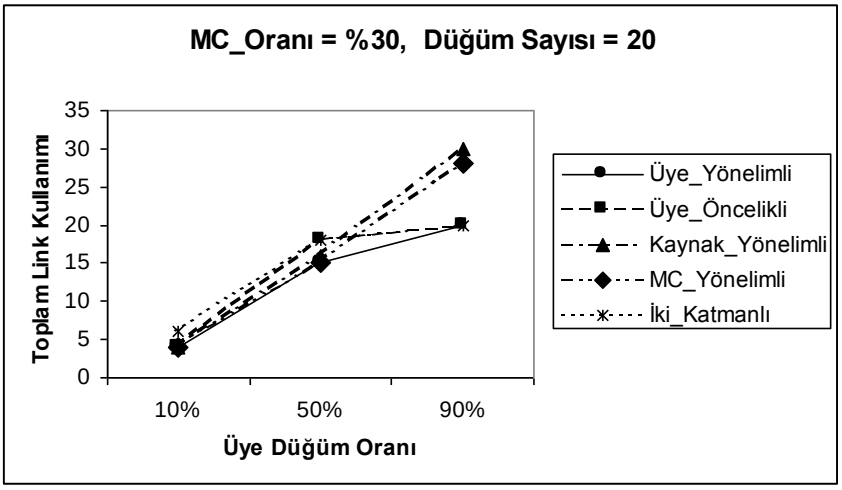
[17] **Ramalho M.,** 2000. Intra and Inter-Domain Multicast Routing Protocols: A Survey and Taxonomy, IEEE Communications Surveys & Tutorials. First Quarter 2000, **3**, No. 1, 2-25.

[18] **Deering S**. , 1989, Host Extensions for IP Multicasting, *RFC 1112*.

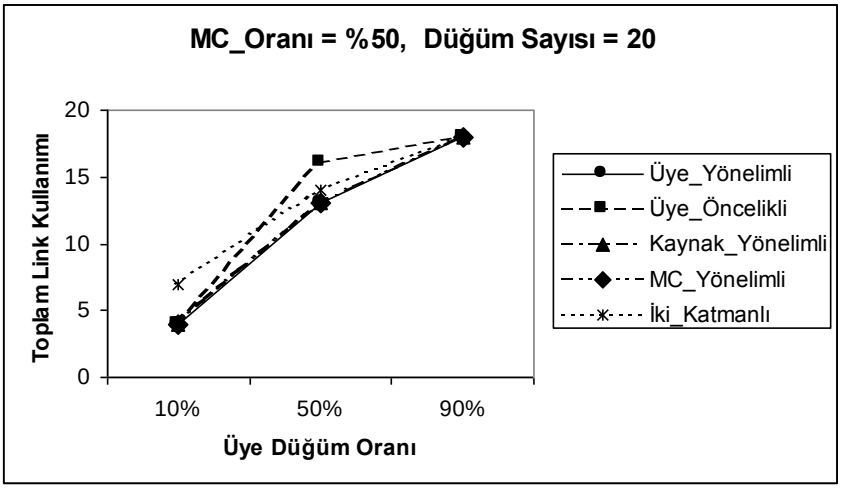
EK – A



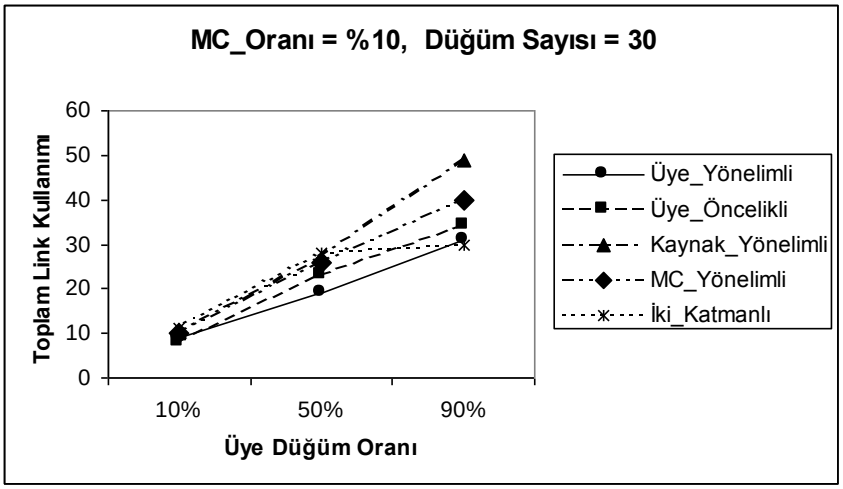
Şekil A.1: Toplam link kullanımının üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 20)



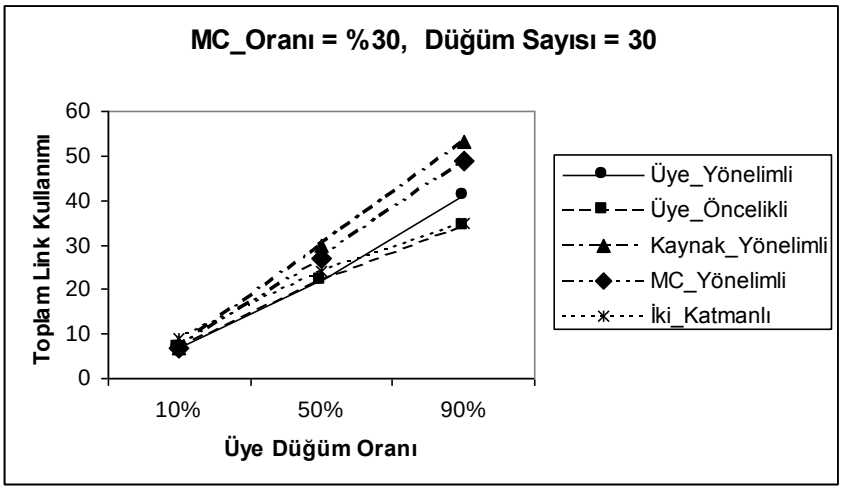
Şekil A.2: Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 20)



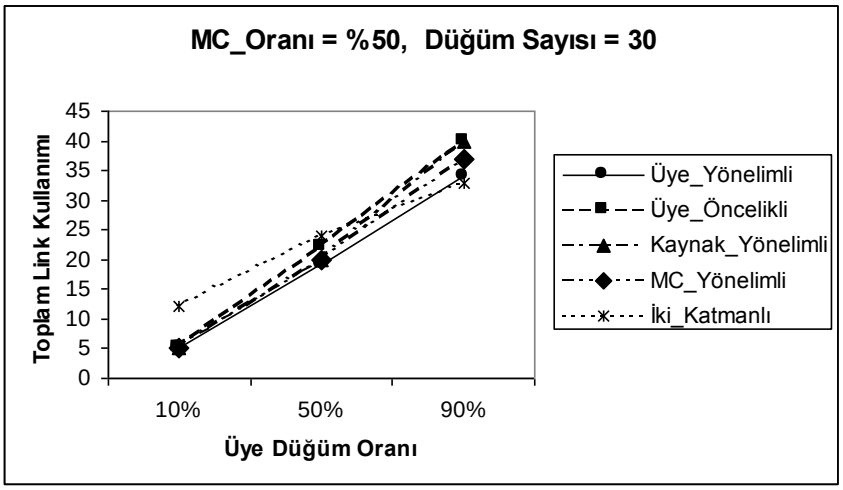
Şekil A.3: Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 20)



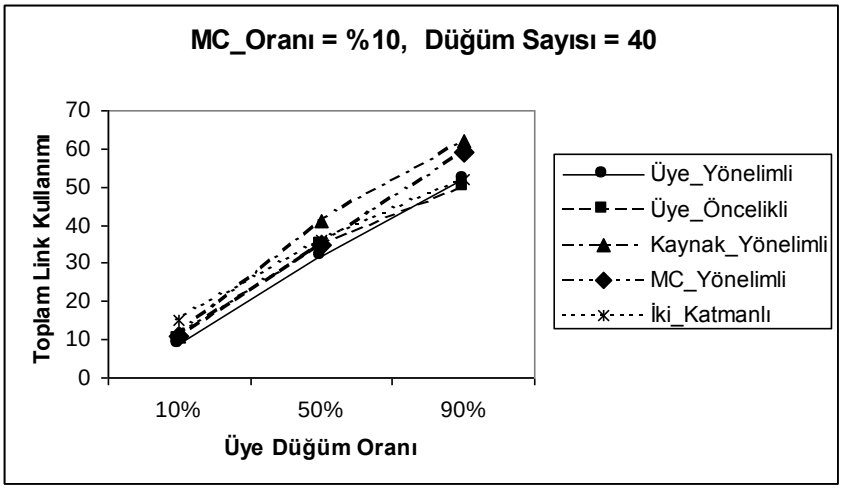
Şekil A.4: Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 30)



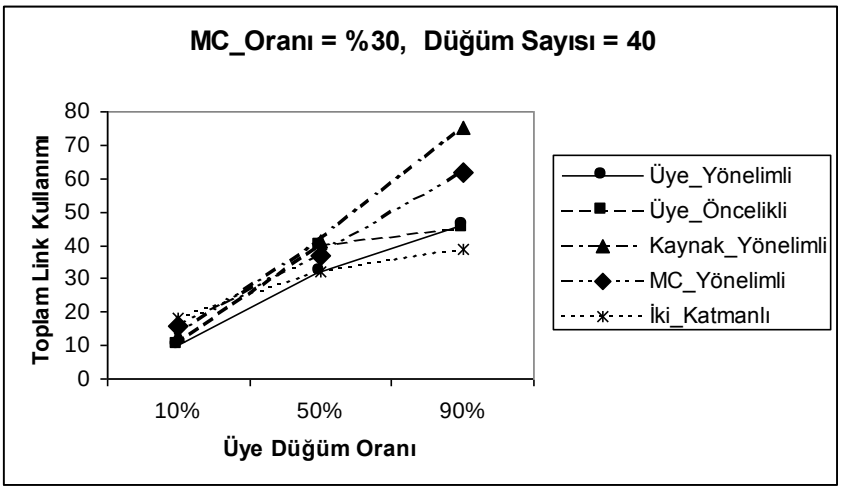
Şekil A.5: Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 30)



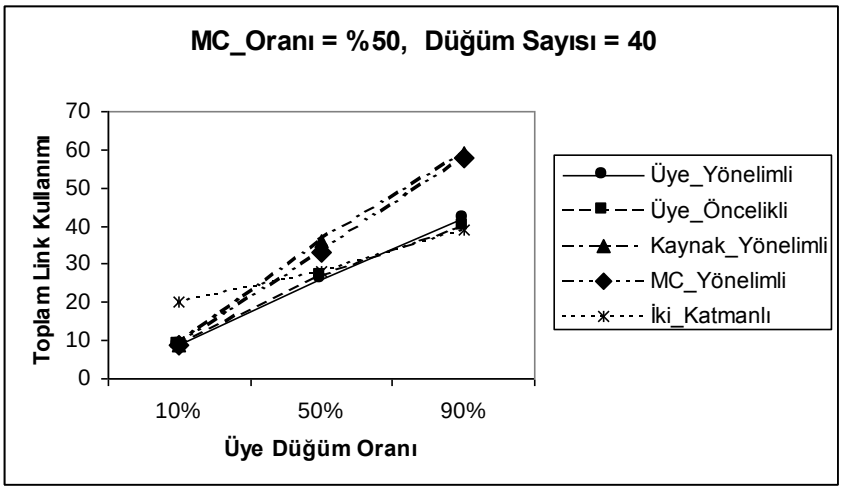
Şekil A.6: Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 30)



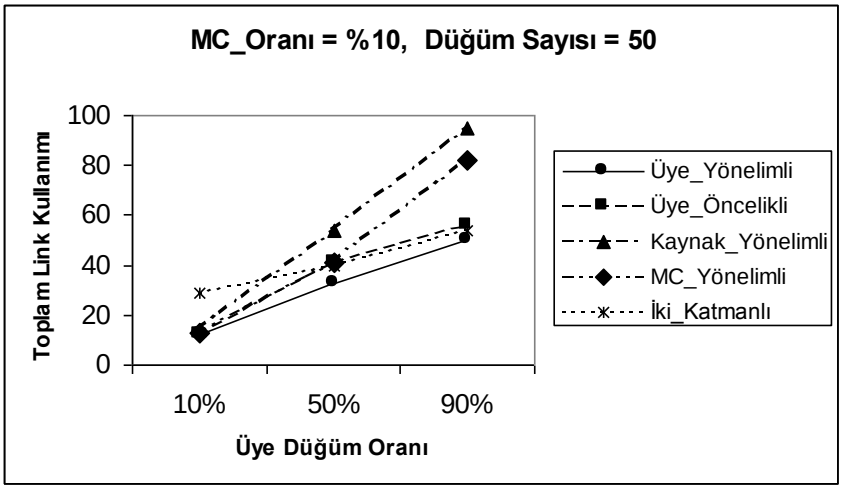
Şekil A.7: Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 40)



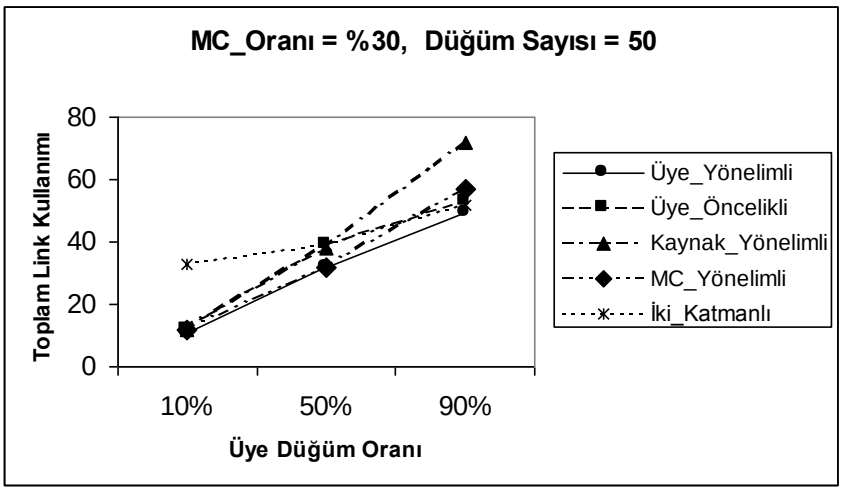
Şekil A.8: Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 40)



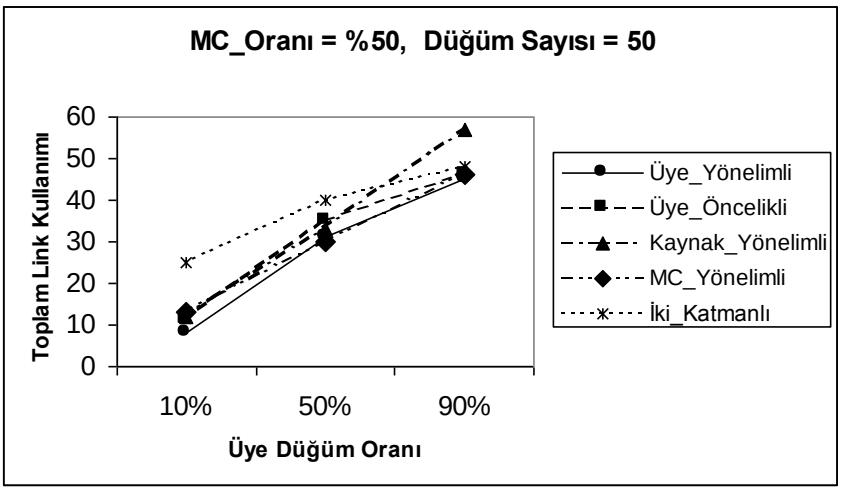
Şekil A.9: Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 40)



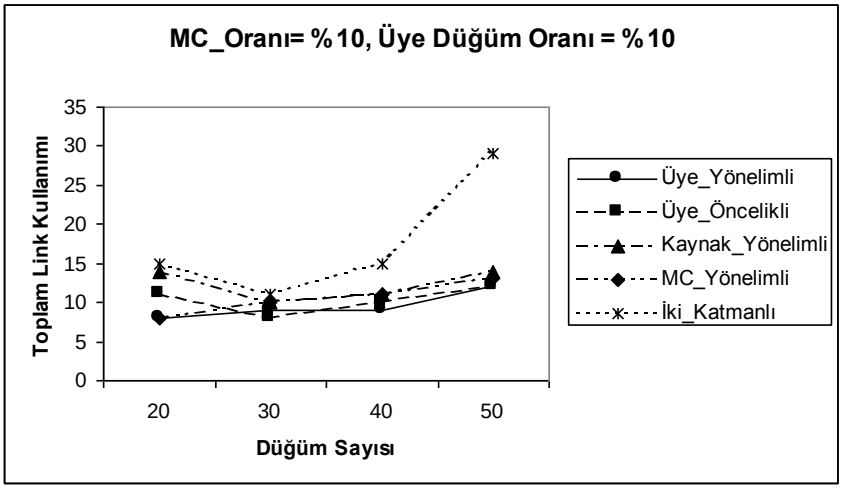
Şekil A.10: Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı= 50)



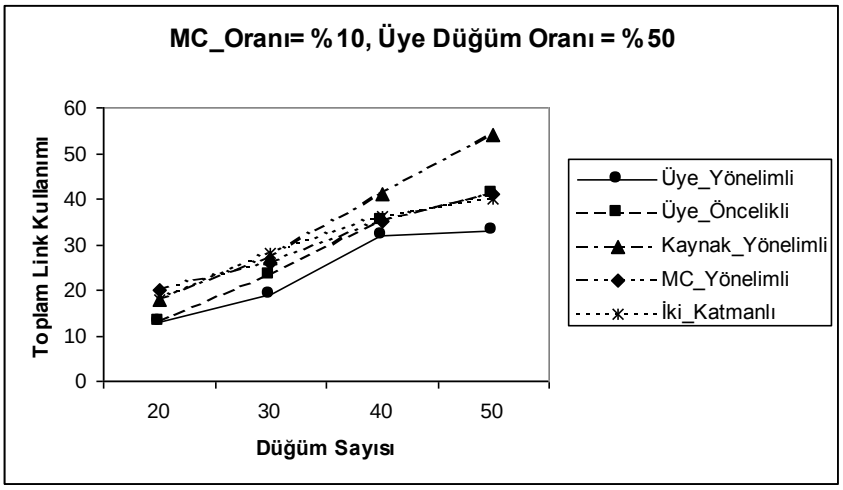
Şekil A.11: Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı= 50)



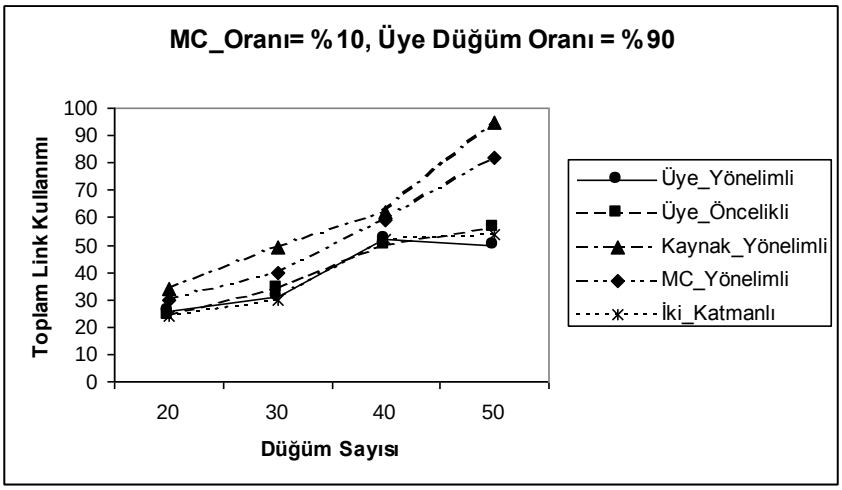
Şekil A.12: Toplam Link Kullanımının Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı= 50)



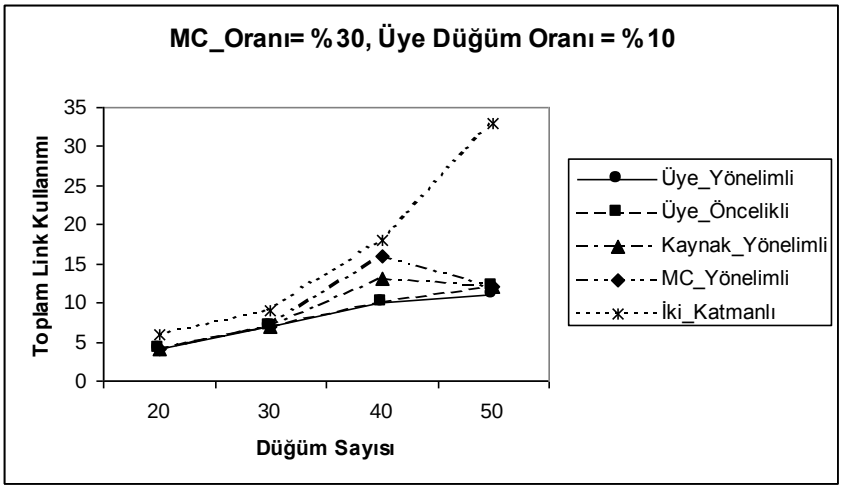
Şekil A.13: Toplam Link Kullanımının düğüm sayısına göre değişimi (MC Oranı=%10, Üye Düğüm Oranı=%10)



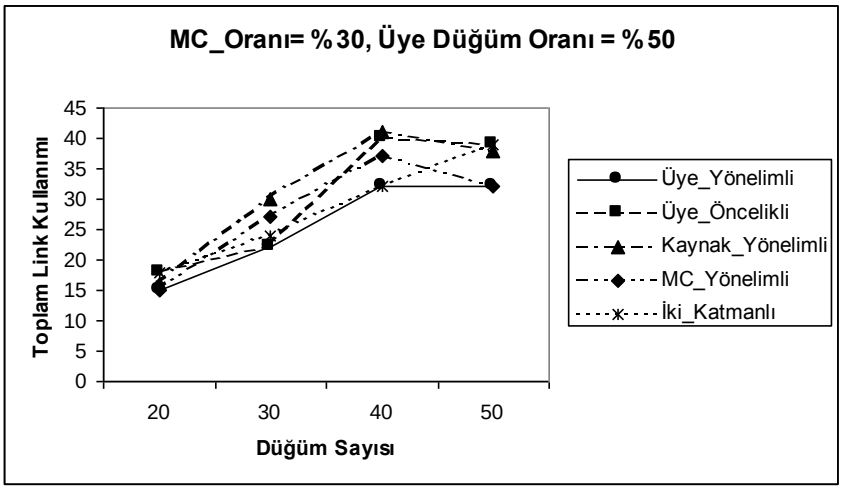
Şekil A.14: Toplam Link Kullanımının düğüm sayısına göre değişimi (MC Oranı=%10, Üye Düğüm Oranı=%50)



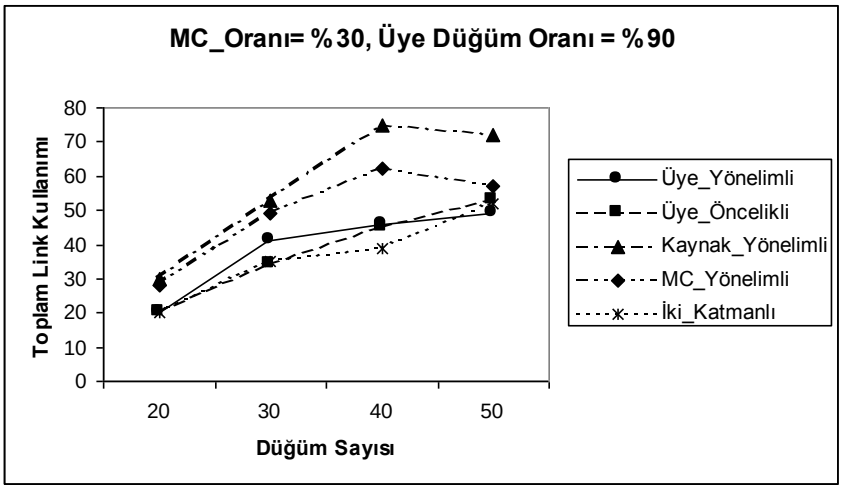
Şekil A.15: Toplam Link Kullanımının düğüm sayısına göre değişimi (MC Oranı=%10, Üye Düğüm Oranı=%90)



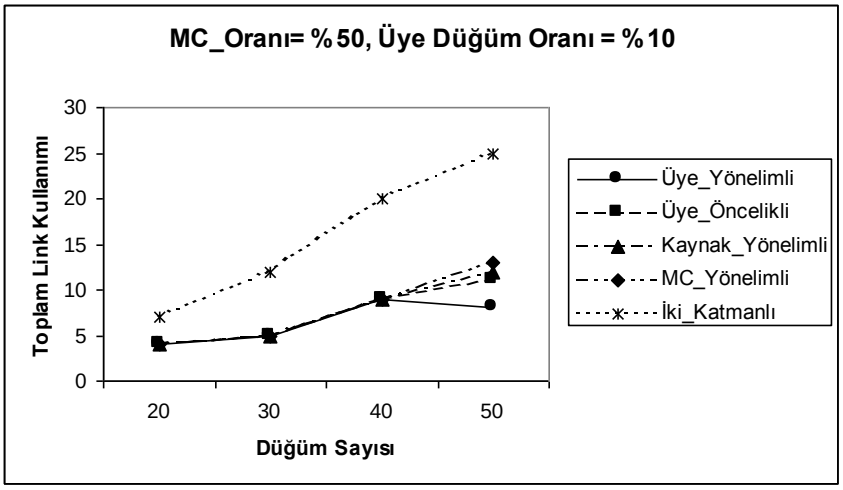
Şekil A.16: Toplam Link Kullanımının düğüm sayısına göre değişimi (MC Oranı=%30, Üye Düğüm Oranı=%10)



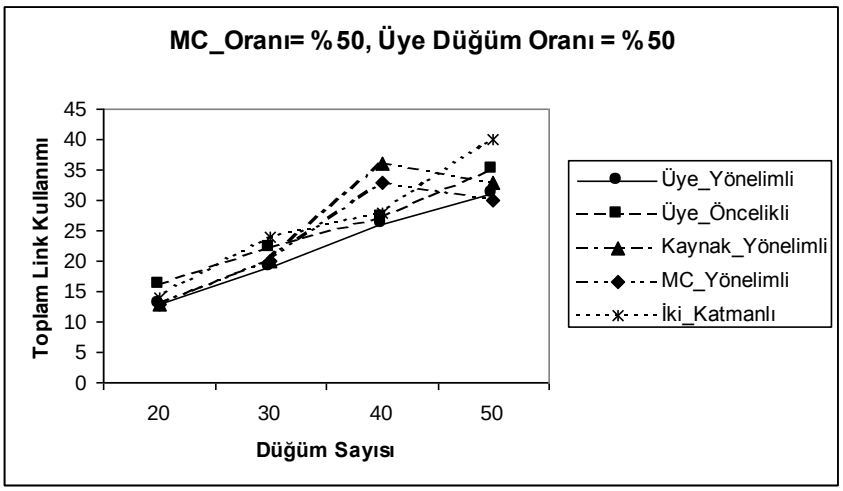
Şekil A.17: Toplam Link Kullanımının düğüm sayısına göre değişimi (MC Oranı=%30, Üye Düğüm Oranı=%50)



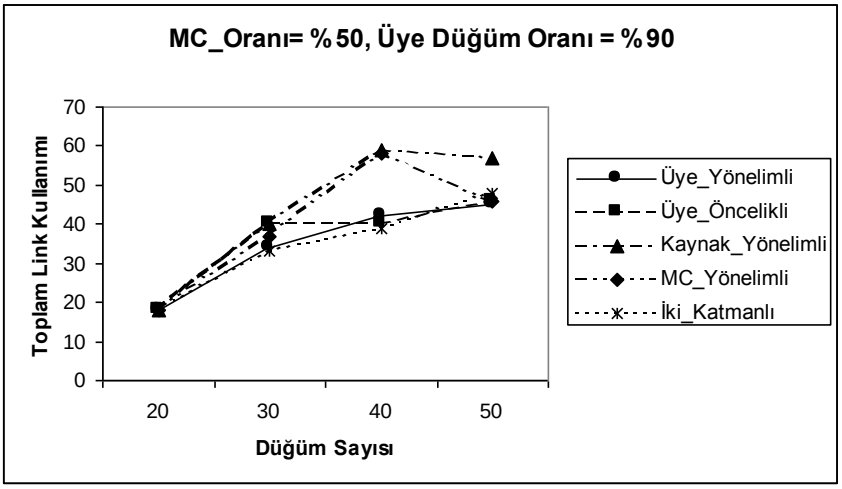
Şekil A.18: Toplam Link Kullanımının düğüm sayısına göre değişimi (MC Oranı=%30, Üye Düğüm Oranı=%90)



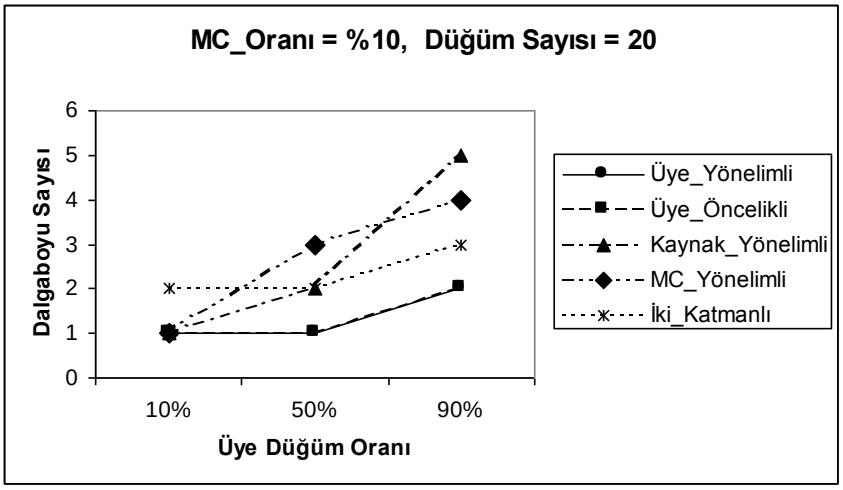
Şekil A.19: Toplam Link Kullanımının düğüm sayısına göre değişimi (MC Oranı=%50, Üye Düğüm Oranı=%10)



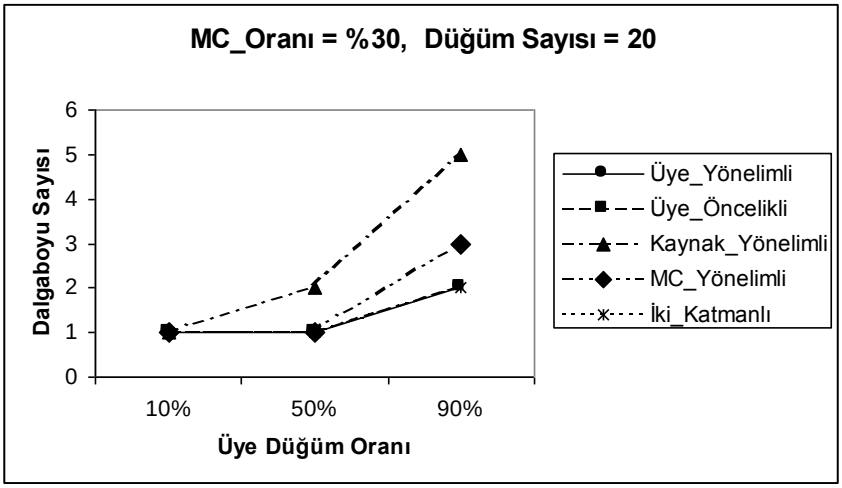
Şekil A.20: Toplam Link Kullanımının düğüm sayısına göre değişimi (MC Oranı=%50, Üye Düğüm Oranı=%50)



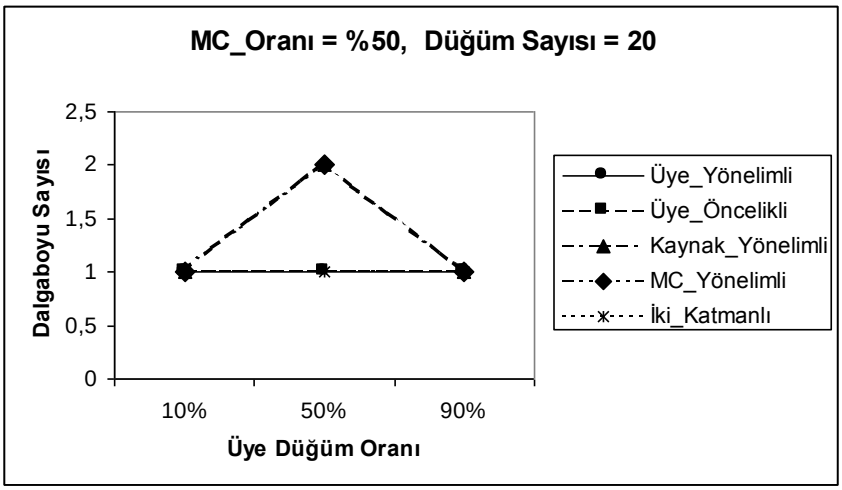
Şekil A.21: Toplam Link Kullanımının düğüm sayısına göre değişimi (MC Oranı=%50, Üye Düğüm Oranı=%90)



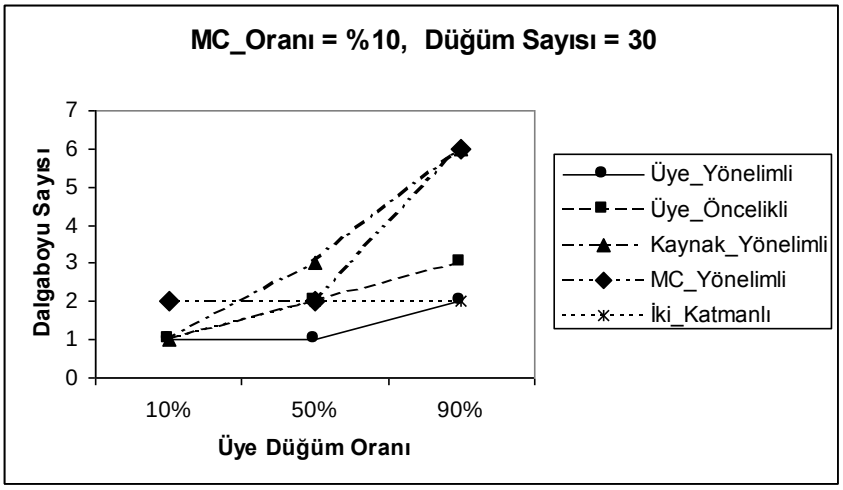
Şekil A.22: Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 20)



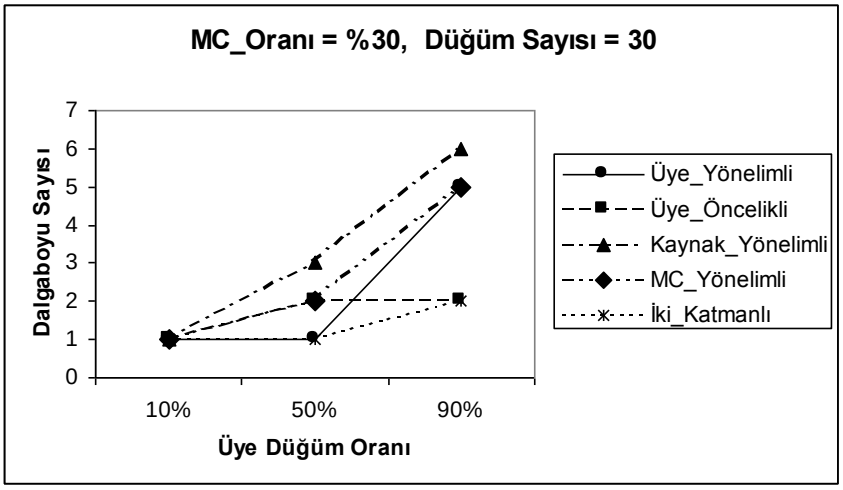
Şekil A.23: Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 20)



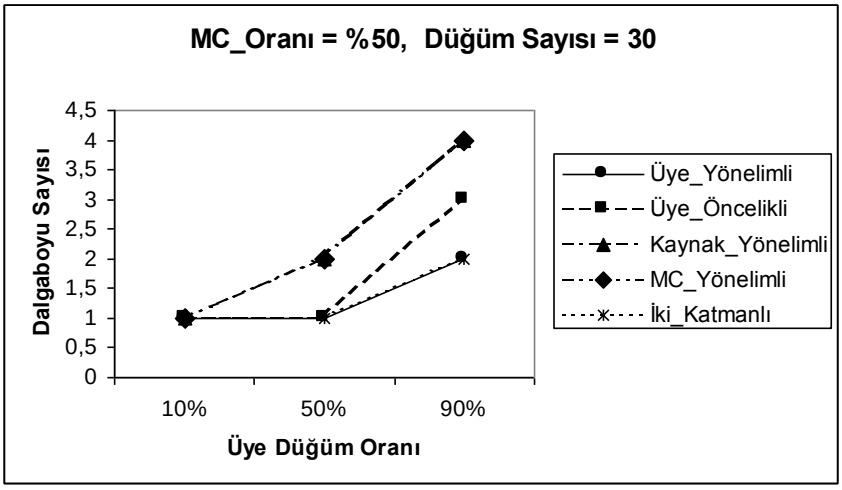
Şekil A.24: Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 20)



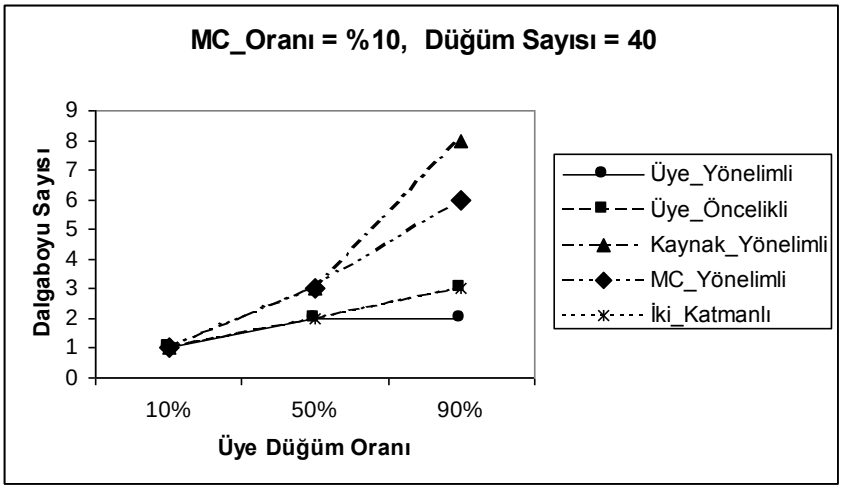
Şekil A.25**:** Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 30)



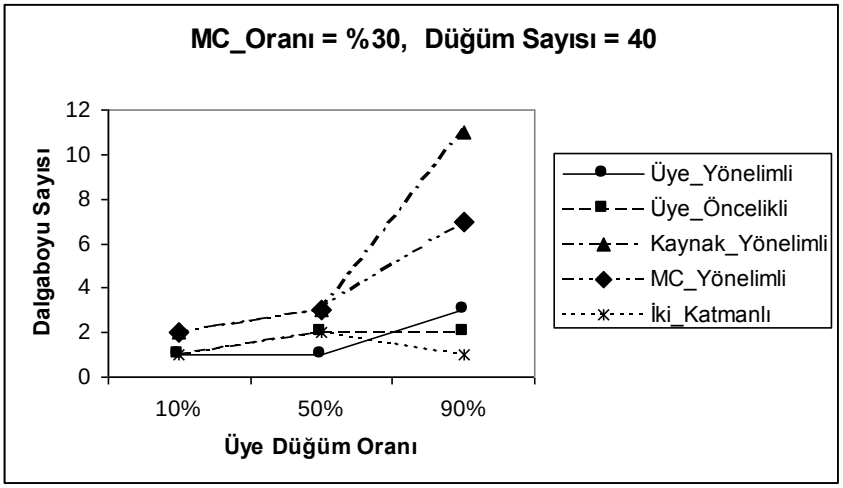
Şekil A.26: Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 30)



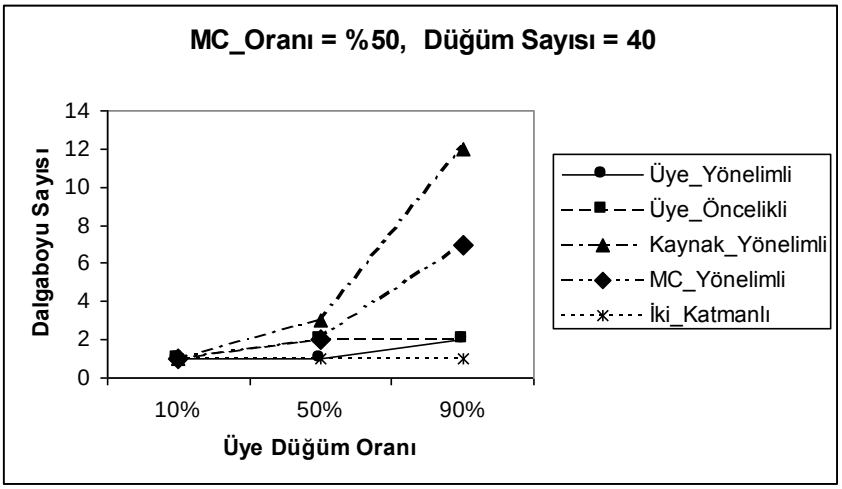
Şekil A.27: Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 30)



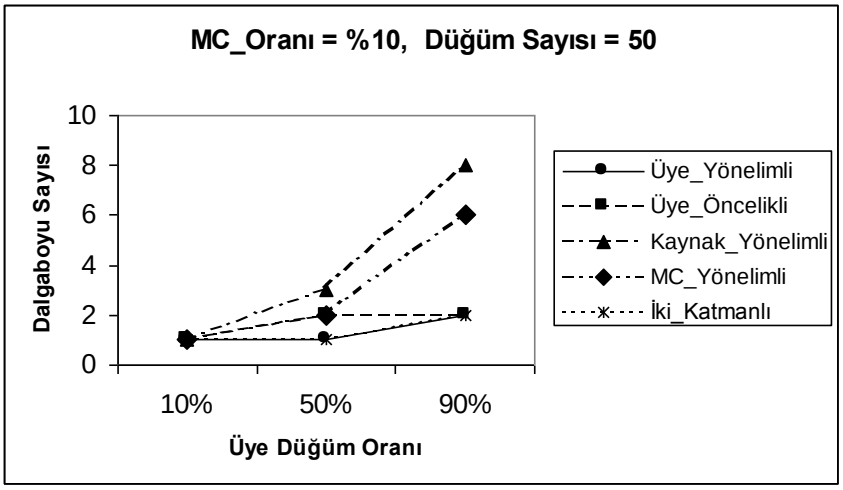
Şekil A.28: Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 40)



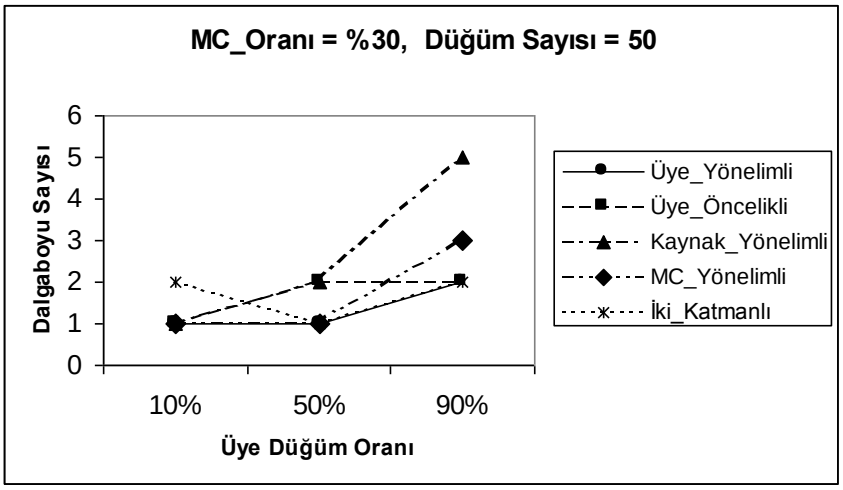
Şekil A.29: Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 40)



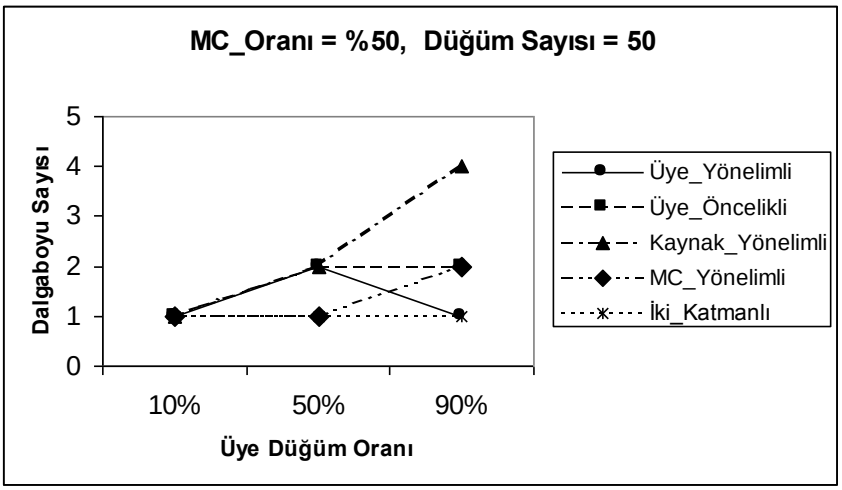
Şekil A.30: Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 40)



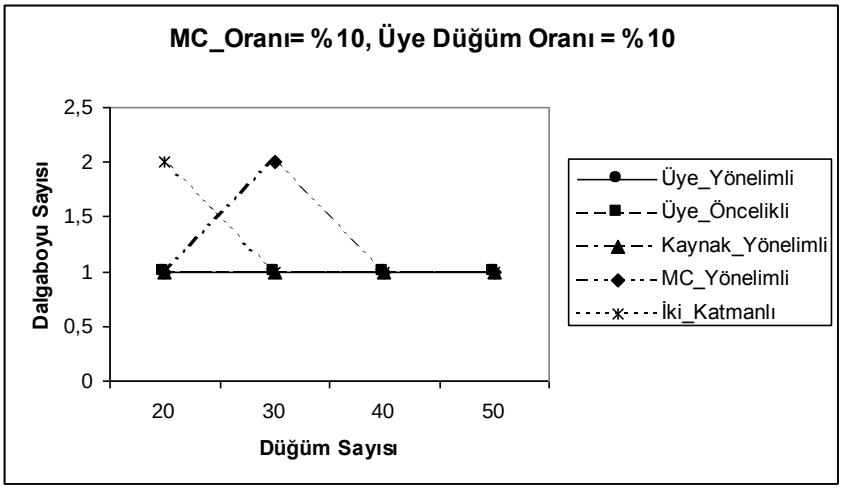
Şekil A.31: Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 50)



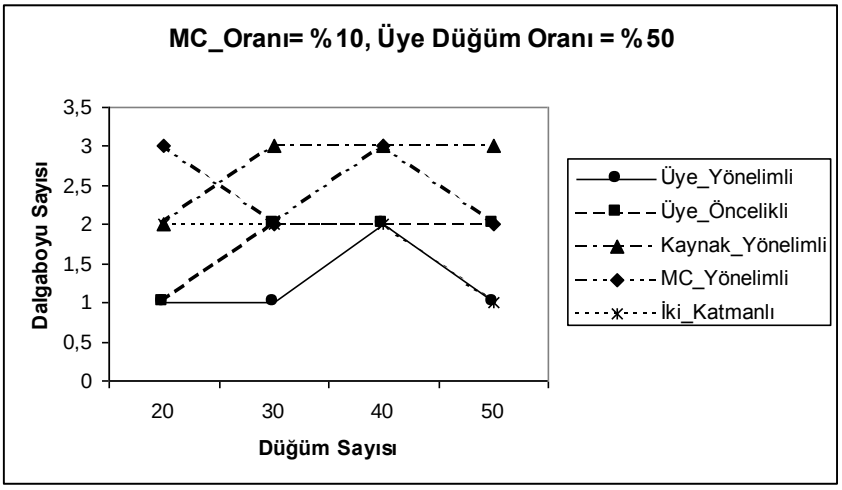
Şekil A.32: Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 50)



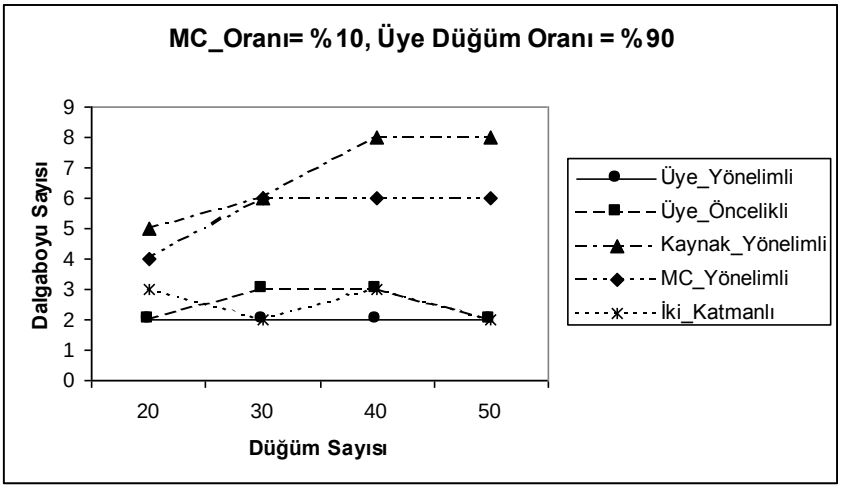
Şekil A.33: Dalgaboyu sayısının Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 50)



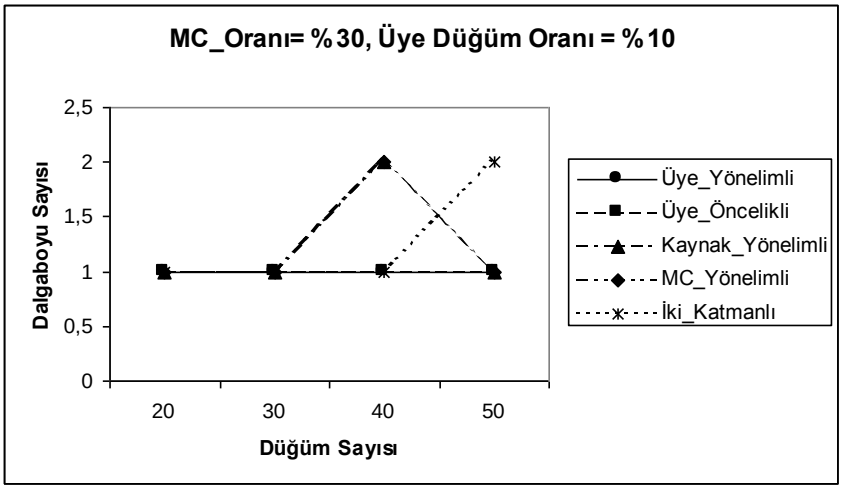
Şekil A.34: Dalgaboyu sayısının Düğüm sayısına göre değişimi (MC Oranı = %10, Üye Düğüm Oranı = %10)



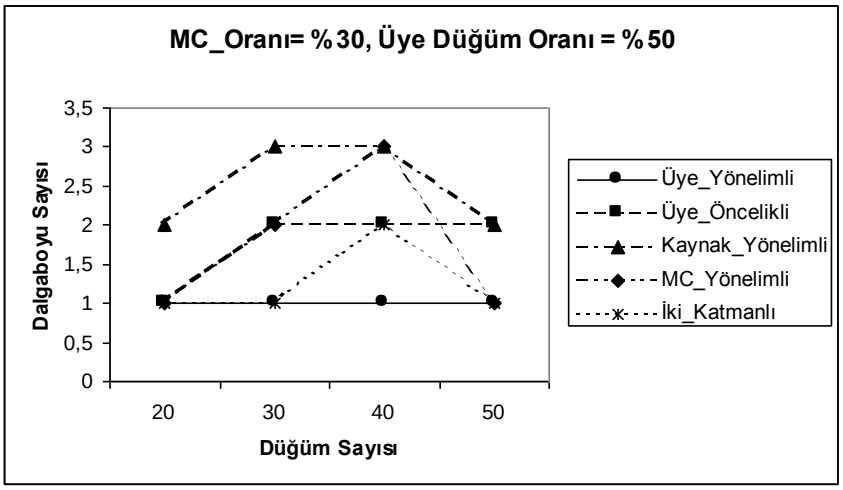
Şekil A.35: Dalgaboyu sayısının Düğüm sayısına göre değişimi (MC Oranı = %10, Üye Düğüm Oranı = %50)



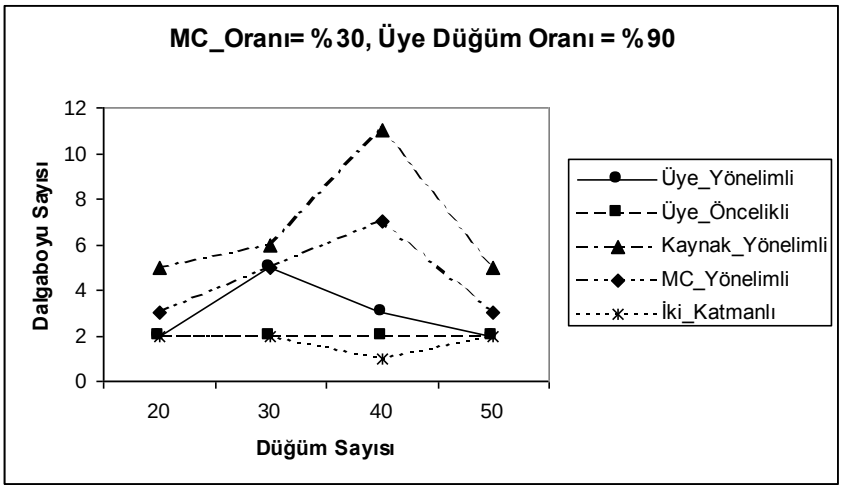
Şekil A.36: Dalgaboyu sayısının Düğüm sayısına göre değişimi (MC Oranı = %10, Üye Düğüm Oranı = %90)



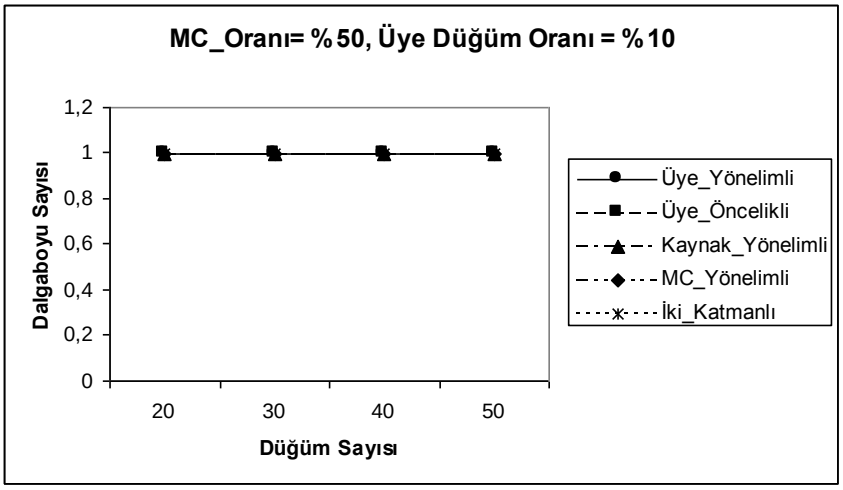
Şekil A.37: Dalgaboyu sayısının Düğüm sayısına göre değişimi (MC Oranı = %30, Üye Düğüm Oranı = %10)



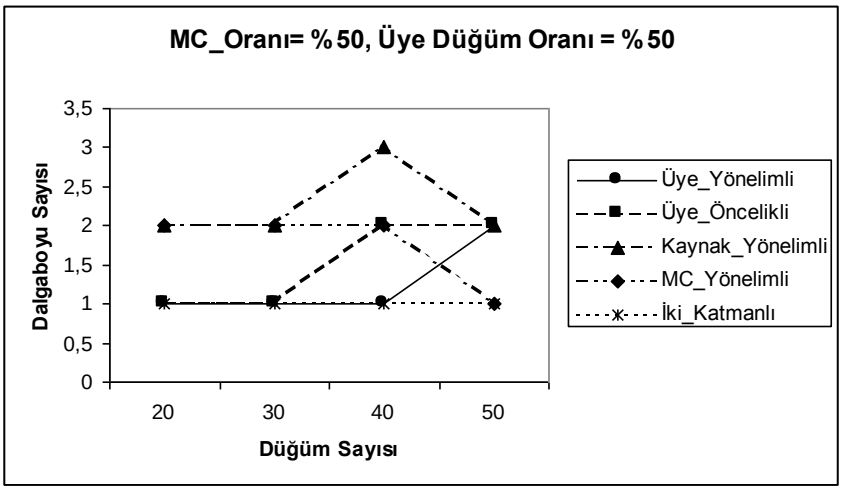
Şekil A.38: Dalgaboyu sayısının Düğüm sayısına göre değişimi (MC Oranı = %30, Üye Düğüm Oranı = %50)



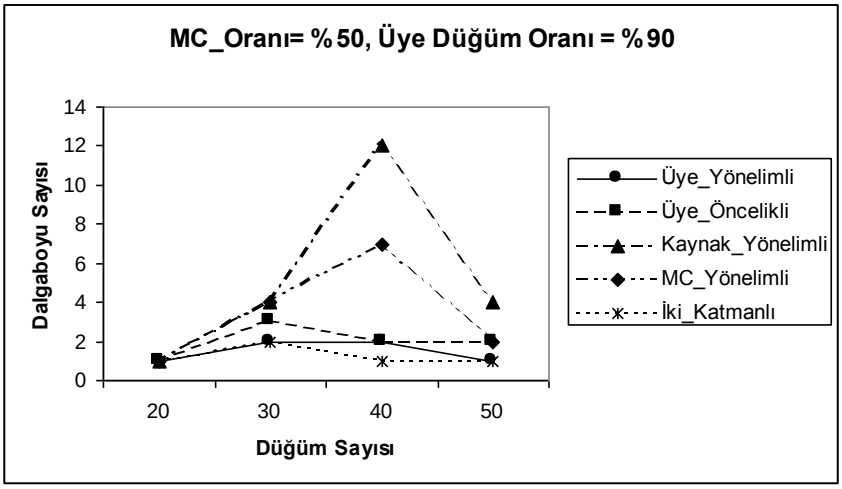
Şekil A.39: Dalgaboyu sayısının Düğüm sayısına göre değişimi (MC Oranı = %30, Üye Düğüm Oranı = %90)



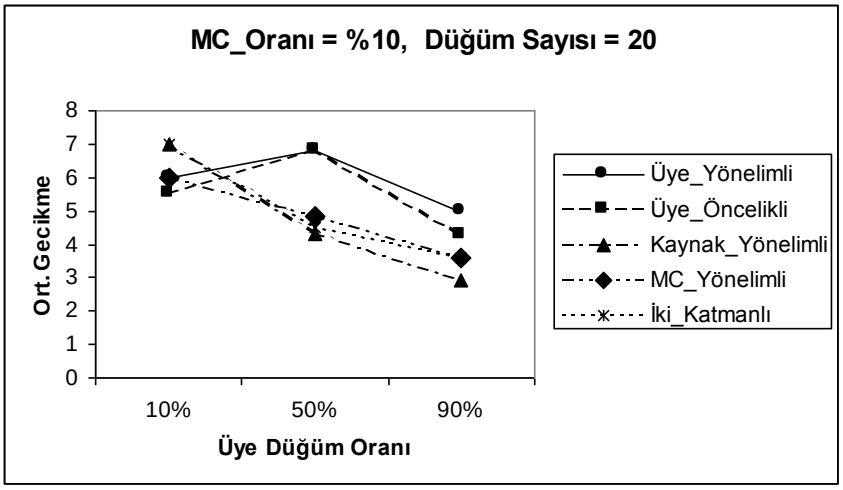
Şekil A.40: Dalgaboyu sayısının Düğüm sayısına göre değişimi (MC Oranı = %50, Üye Düğüm Oranı = %10)



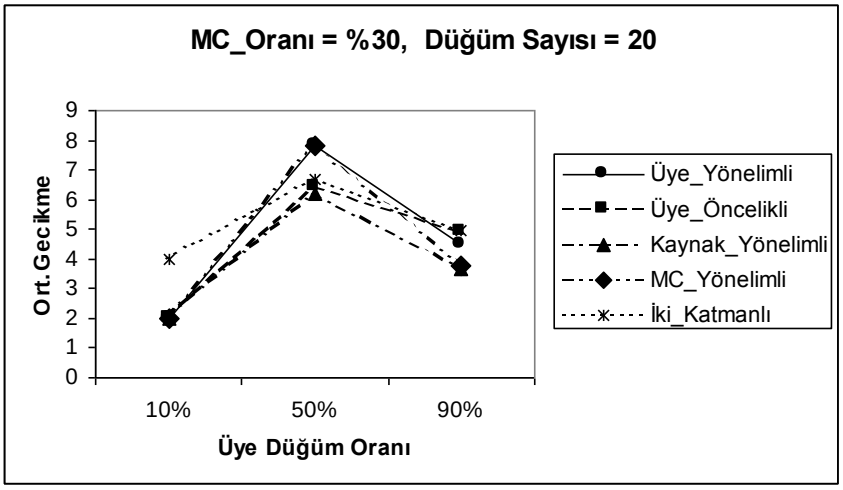
Şekil A.41: Dalgaboyu sayısının Düğüm sayısına göre değişimi (MC Oranı = %50, Üye Düğüm Oranı = %50)



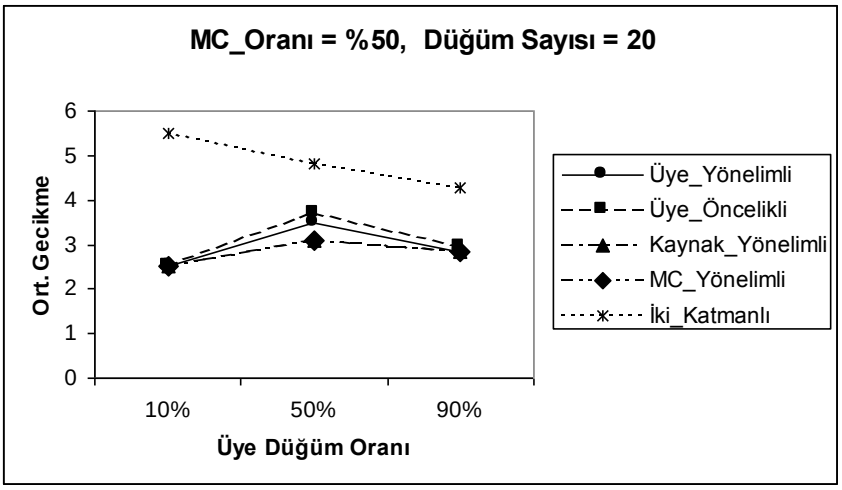
Şekil A.42: Dalgaboyu sayısının Düğüm sayısına göre değişimi (MC Oranı = %50, Üye Düğüm Oranı = %90)



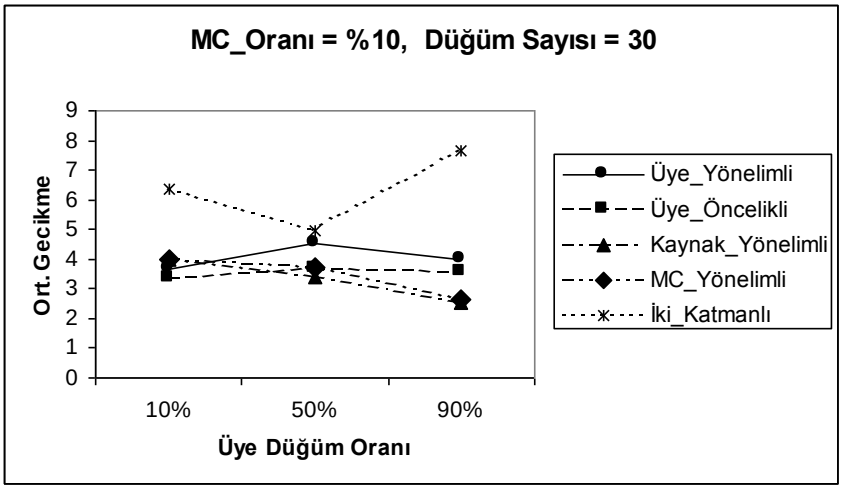
Şekil A.43: Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 20)



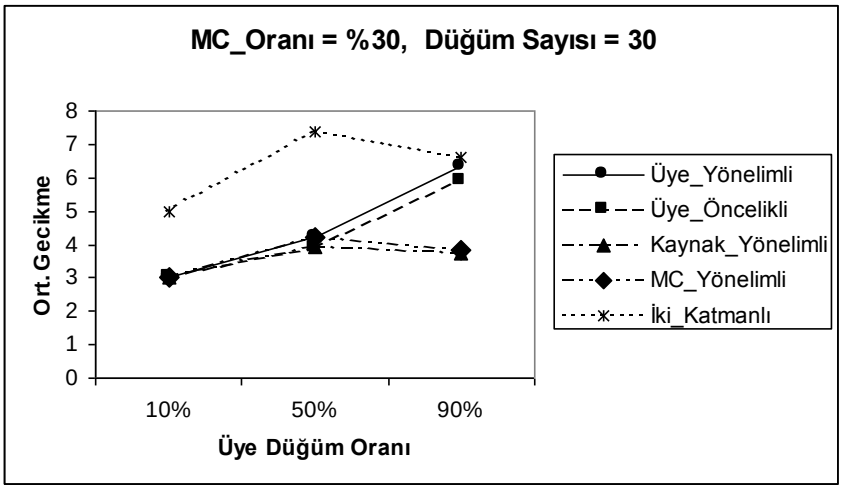
Şekil A.44: Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 20)



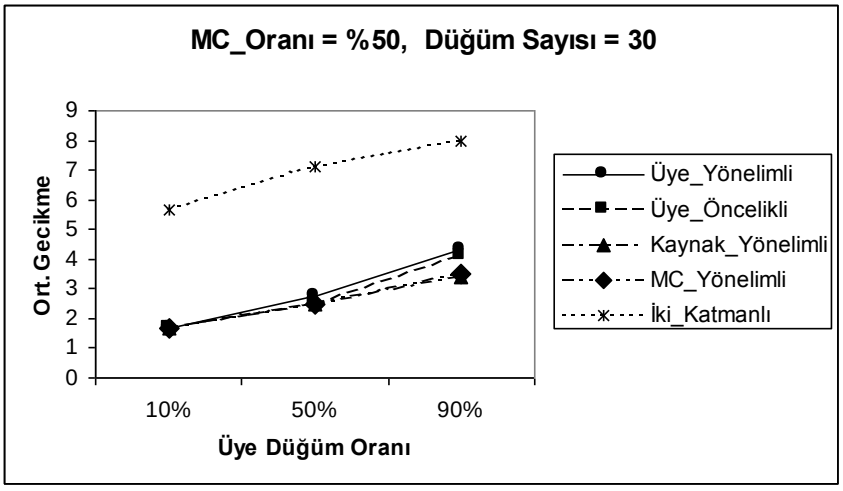
Şekil A.45: Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 20)



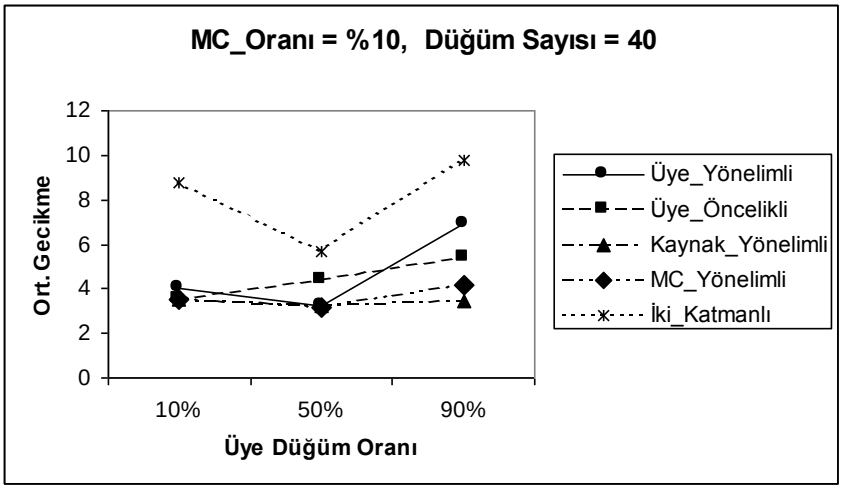
Şekil A.46: Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 30)



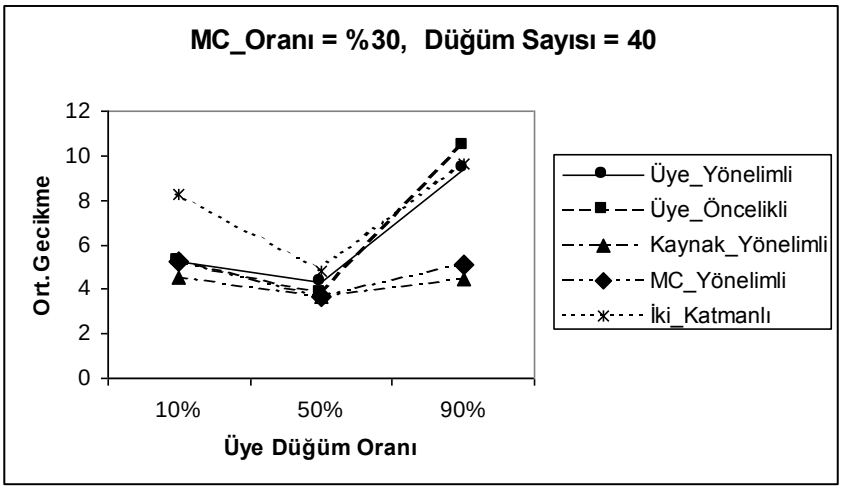
Şekil A.47: Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 30)



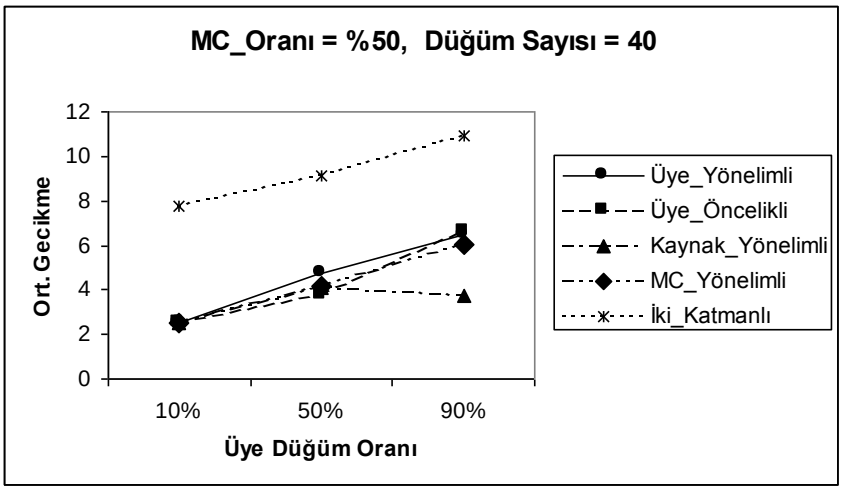
Şekil A.48: Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 30)



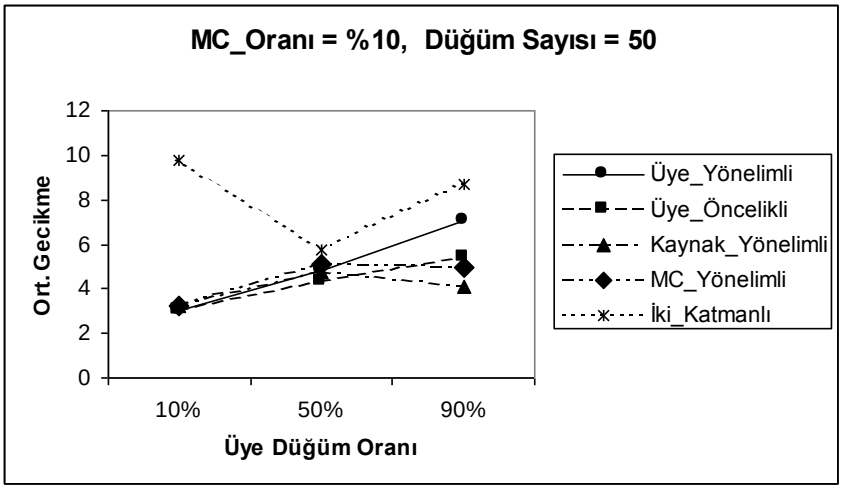
Şekil A.49: Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 40)



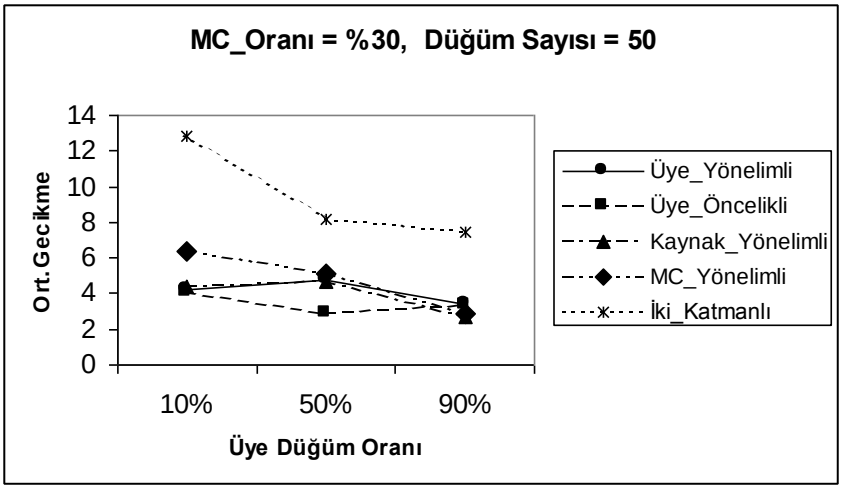
Şekil A.50: Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 40)



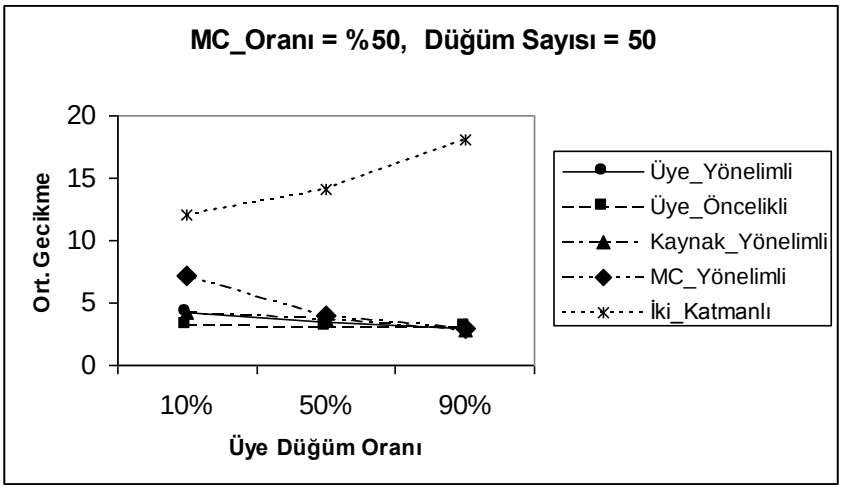
Şekil A.51: Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 40)



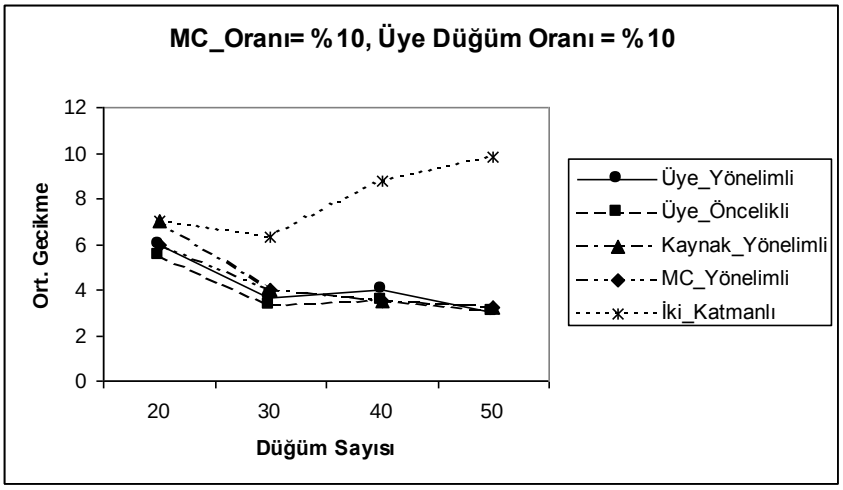
Şekil A.52: Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 50)



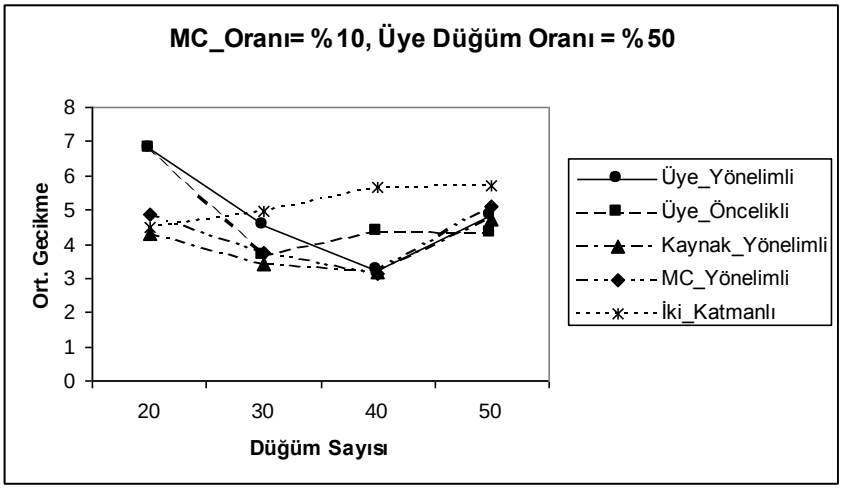
Şekil A.53: Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 50)



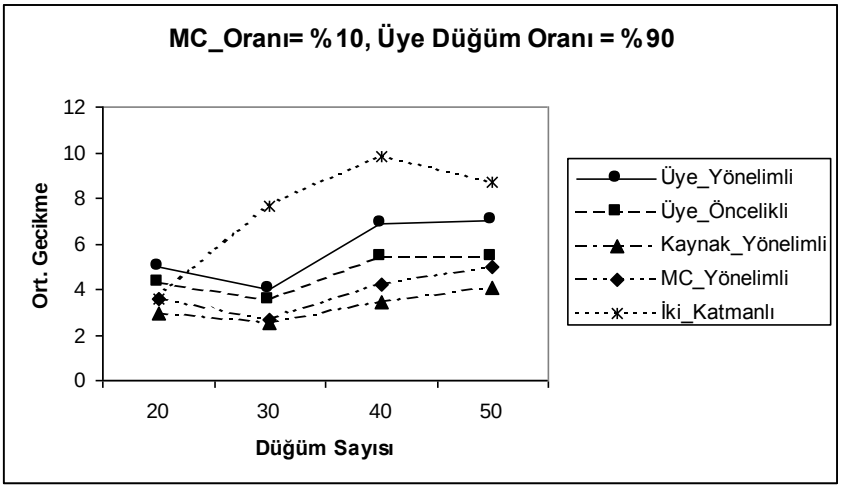
Şekil A.54: Ortalama Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 50)



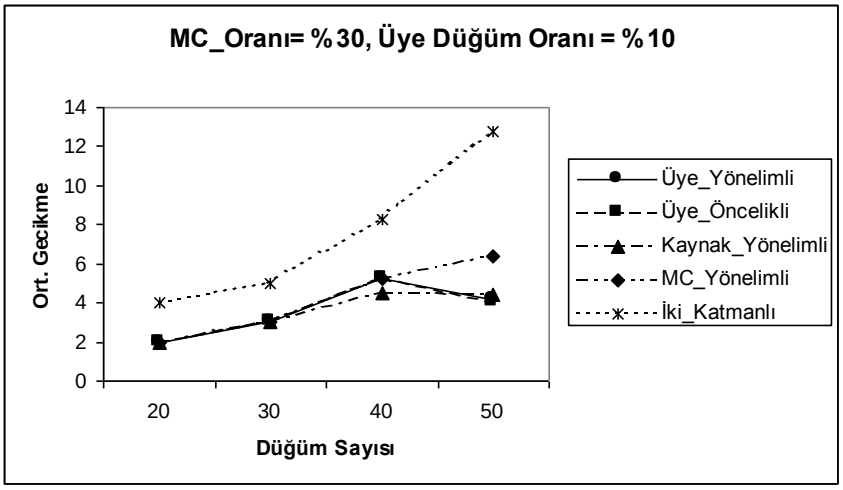
Şekil A.55: Ortalama Gecikmenin Düğüm sayısına göre değişimi (MC Oranı = %10, Üye Düğüm Oranı = %10)



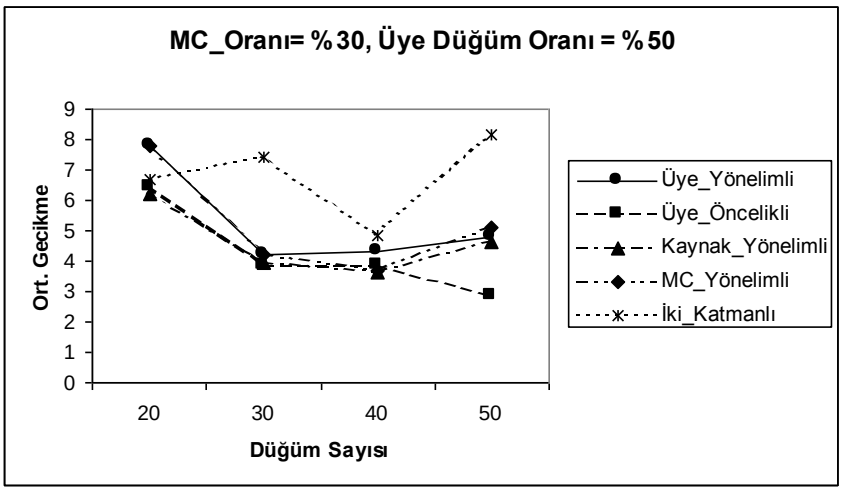
Şekil A.56: Ortalama Gecikmenin Düğüm sayısına göre değişimi (MC Oranı = %10, Üye Düğüm Oranı = %50)



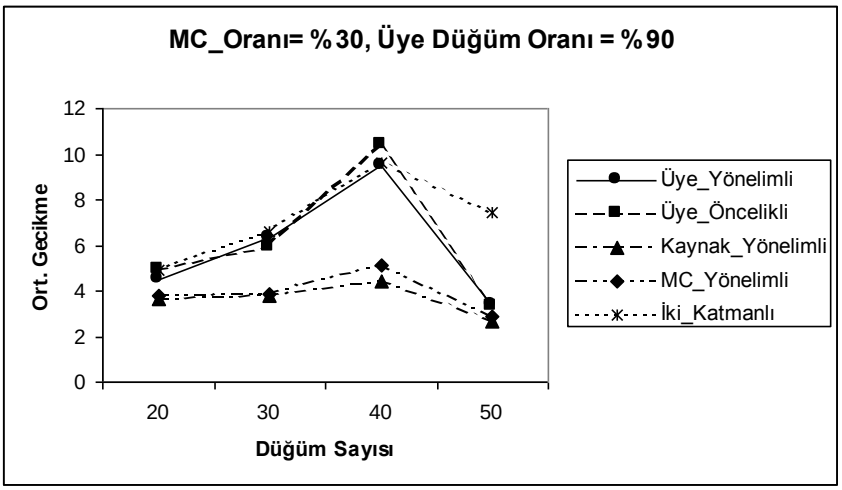
Şekil A.57: Ortalama Gecikmenin Düğüm sayısına göre değişimi (MC Oranı = %10, Üye Düğüm Oranı = %90)



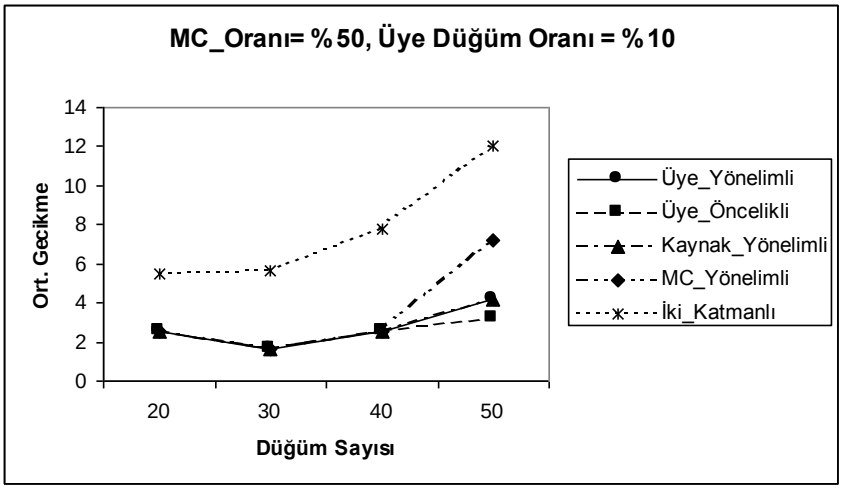
Şekil A.58: Ortalama Gecikmenin Düğüm sayısına göre değişimi (MC Oranı = %30, Üye Düğüm Oranı = %10)



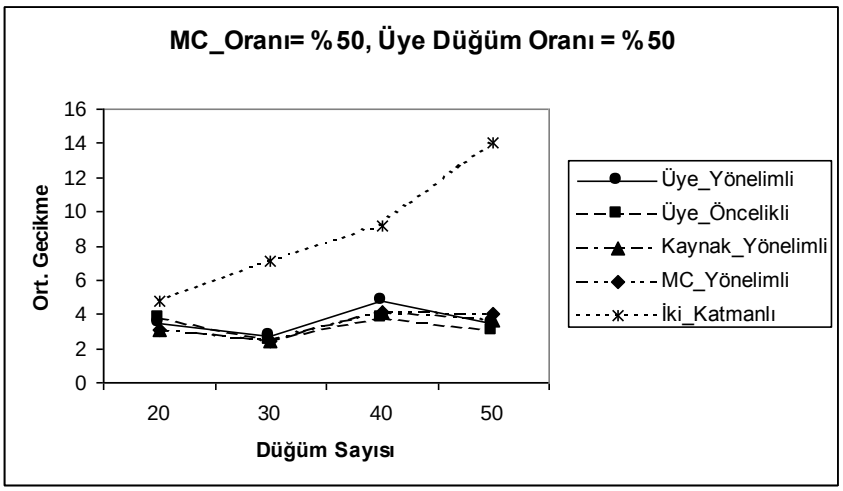
Şekil A.59: Ortalama Gecikmenin Düğüm sayısına göre değişimi (MC Oranı = %30, Üye Düğüm Oranı = %50)



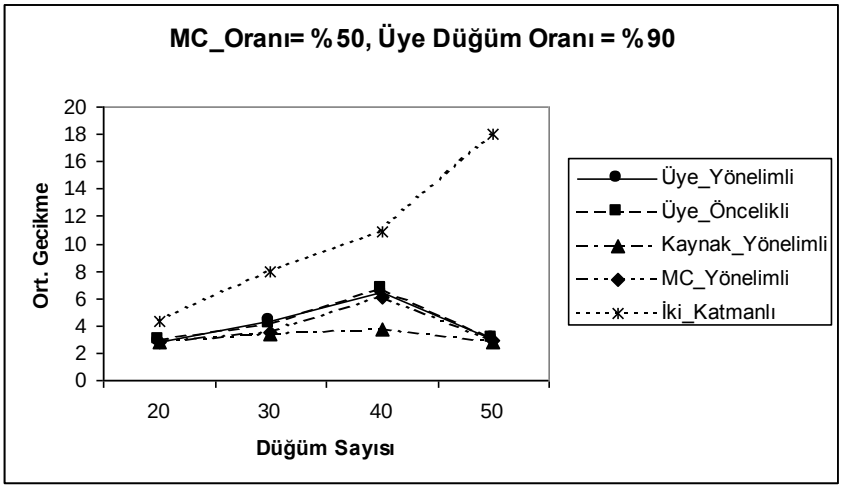
Şekil A.60: Ortalama Gecikmenin Düğüm sayısına göre değişimi (MC Oranı = %30, Üye Düğüm Oranı = %90)



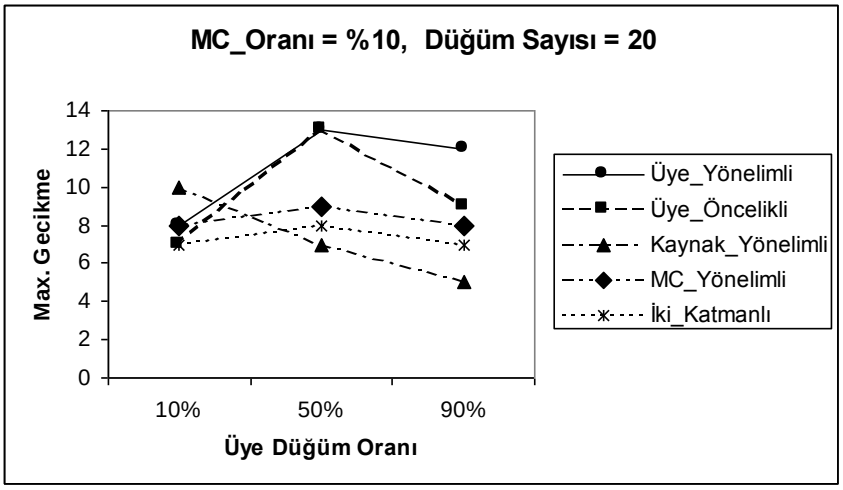
Şekil A.61: Ortalama Gecikmenin Düğüm sayısına göre değişimi (MC Oranı = %50, Üye Düğüm Oranı = %10)



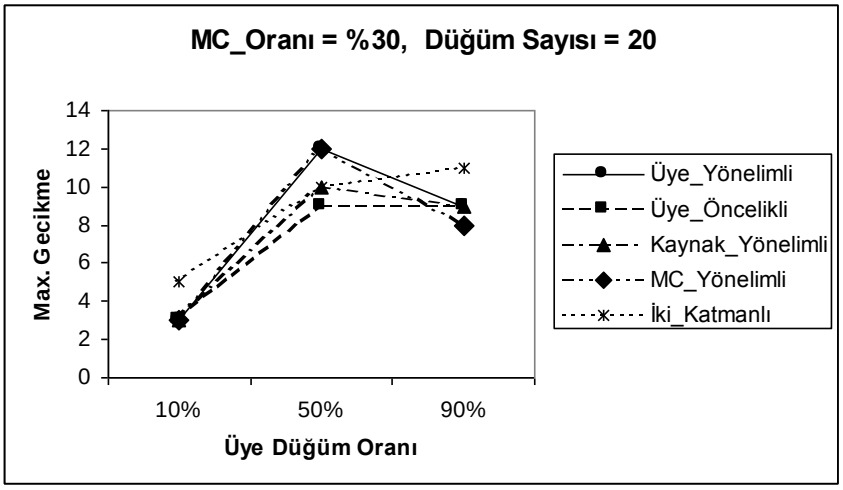
Şekil A.62: Ortalama Gecikmenin Düğüm sayısına göre değişimi (MC Oranı = %50, Üye Düğüm Oranı = %50)



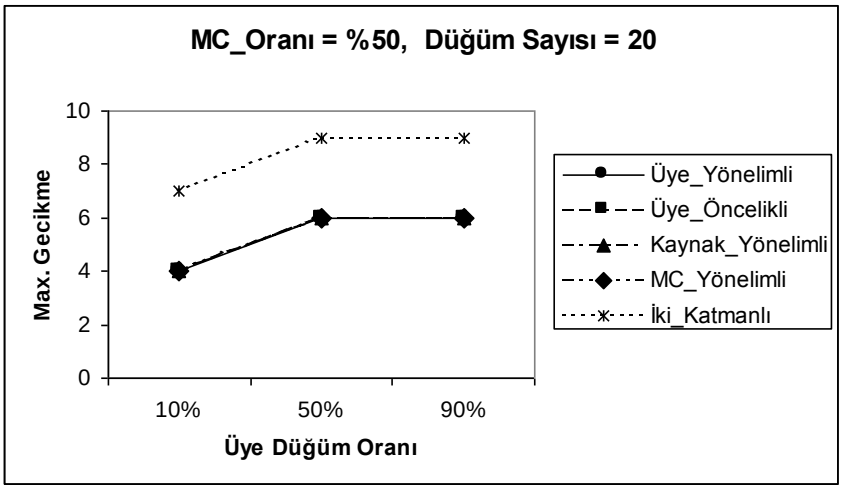
Şekil A.63: Ortalama Gecikmenin Düğüm sayısına göre değişimi (MC Oranı = %50, Üye Düğüm Oranı = %90)



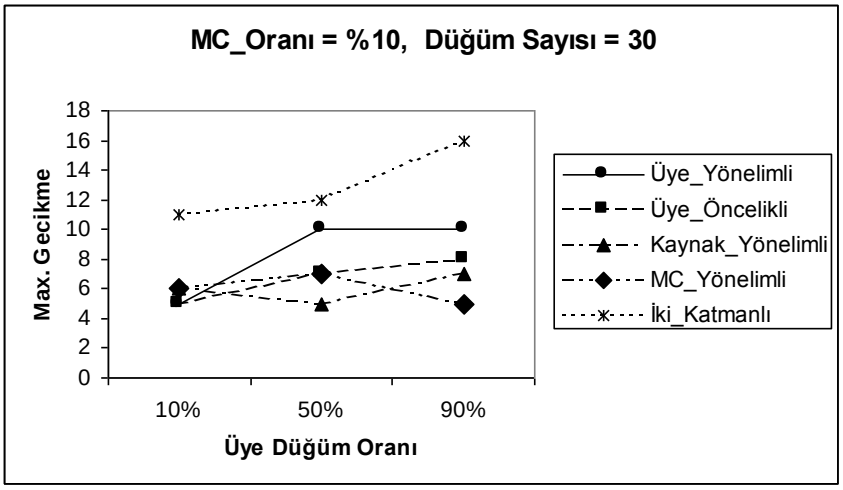
Şekil A.64: Maksimum Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 20)



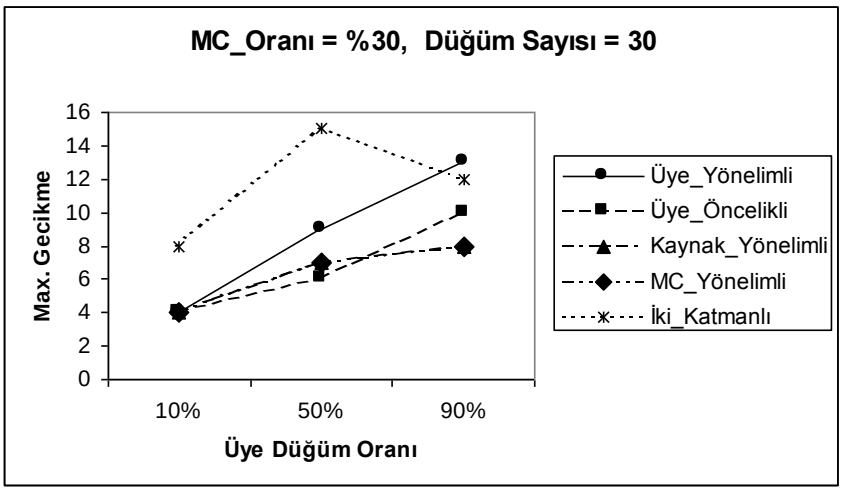
Şekil A.65: Maksimum Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 20)



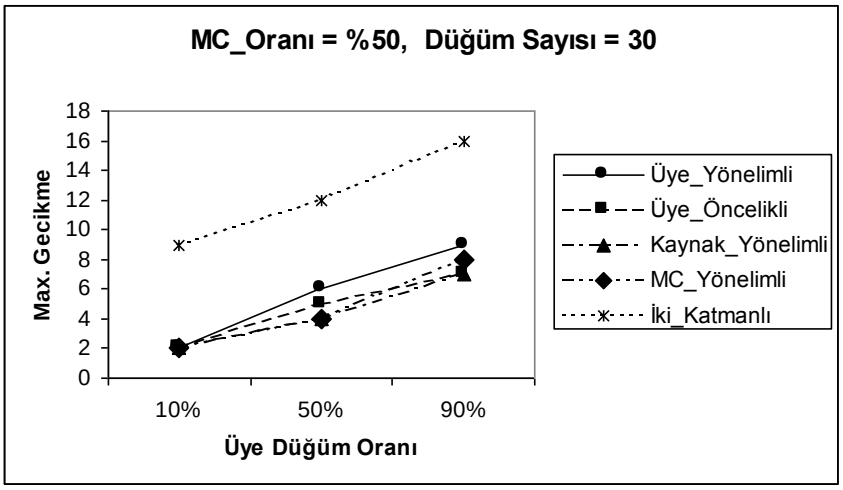
Şekil A.66: Maksimum Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 20)



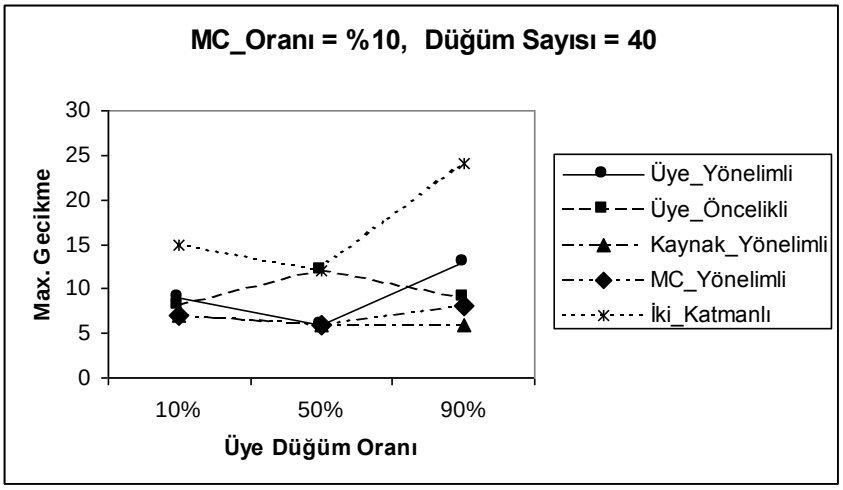
Şekil A.67: Maksimum Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 30)



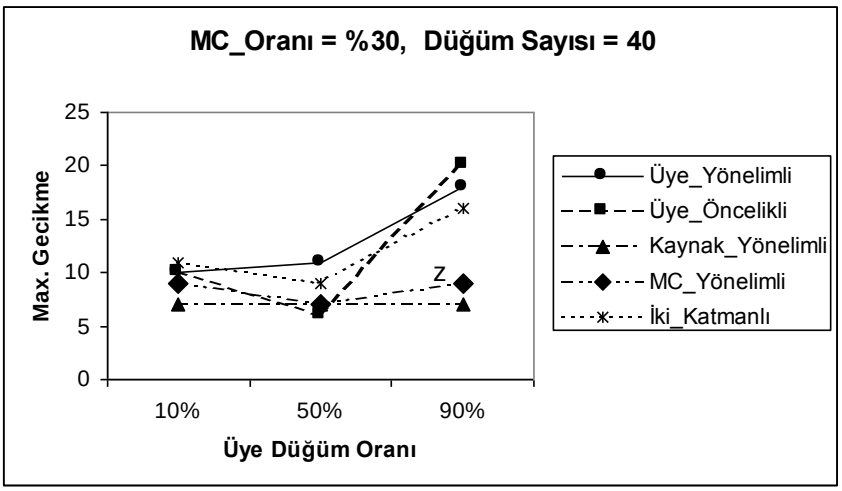
Şekil A.68: Maksimum Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 30)



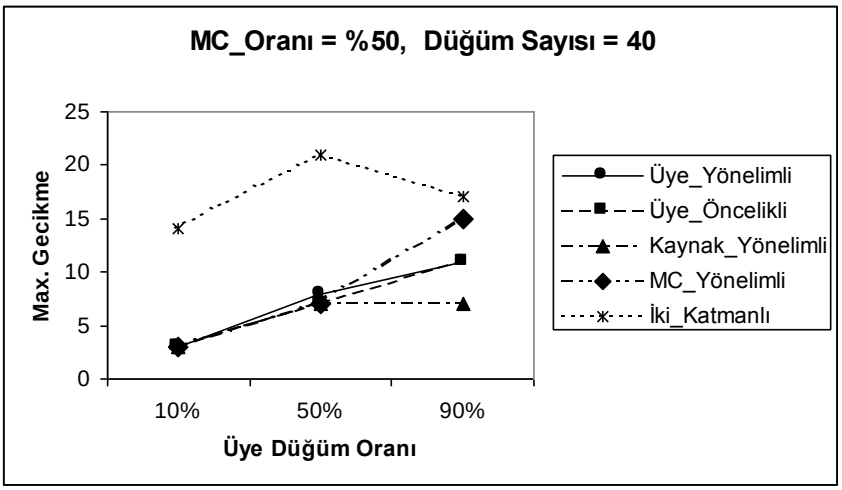
Şekil A.69: Maksimum Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 30)



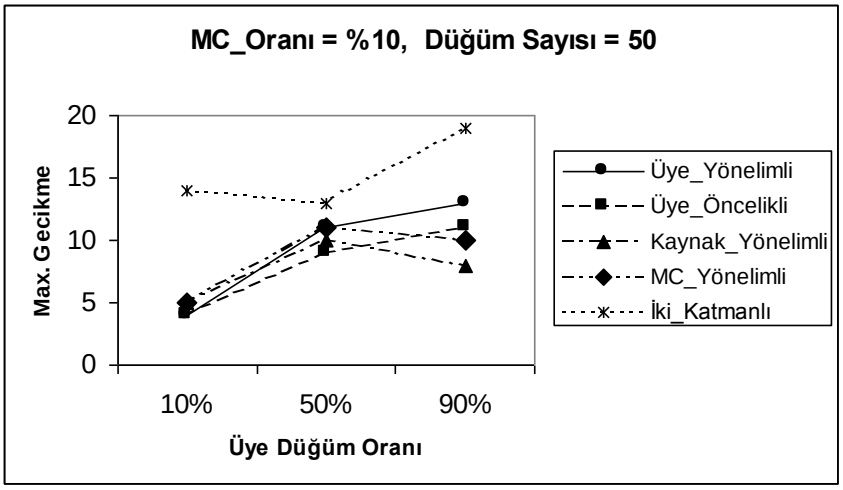
Şekil A.70: Maksimum Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 40)



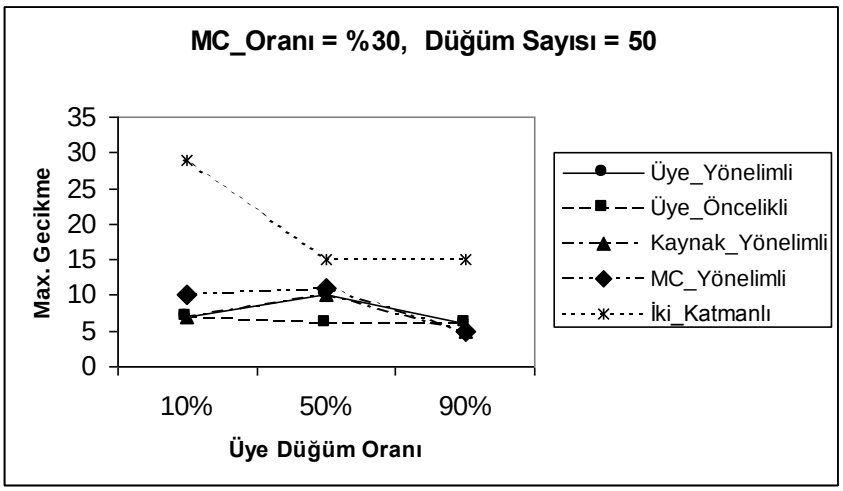
Şekil A.71: Maksimum Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 40)



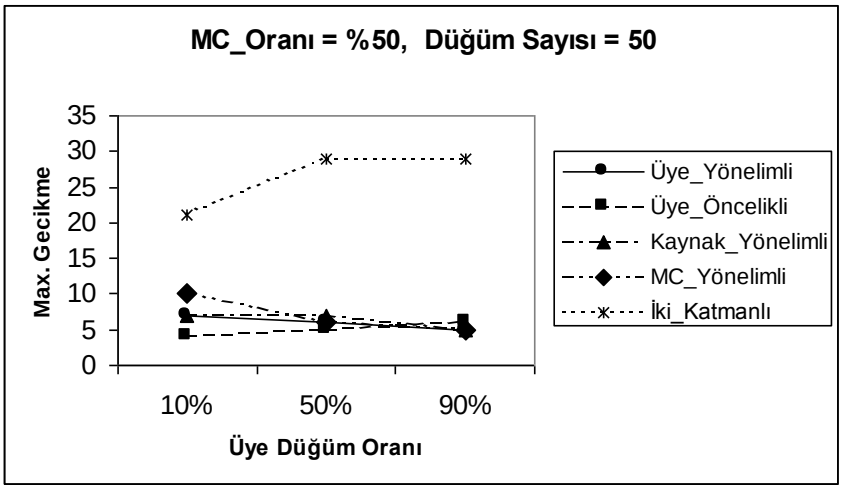
Şekil A.72: Maksimum Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 40)



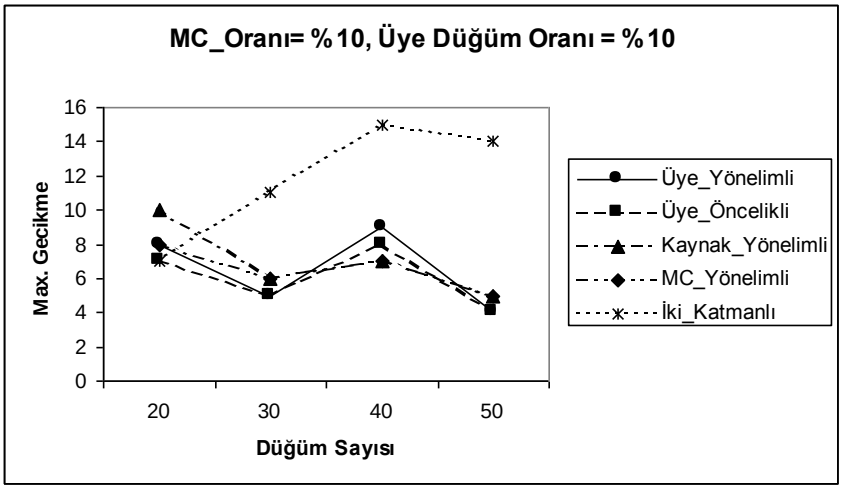
Şekil A.73: Maksimum Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 50)



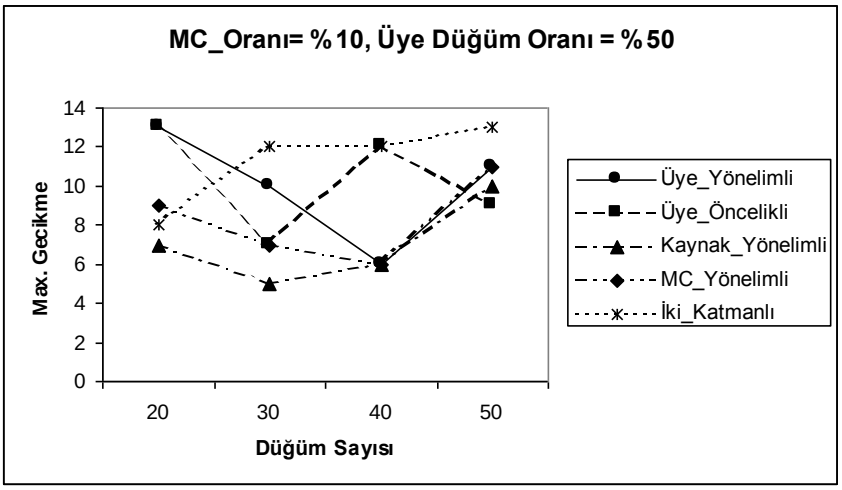
Şekil A.74: Maksimum Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 50)



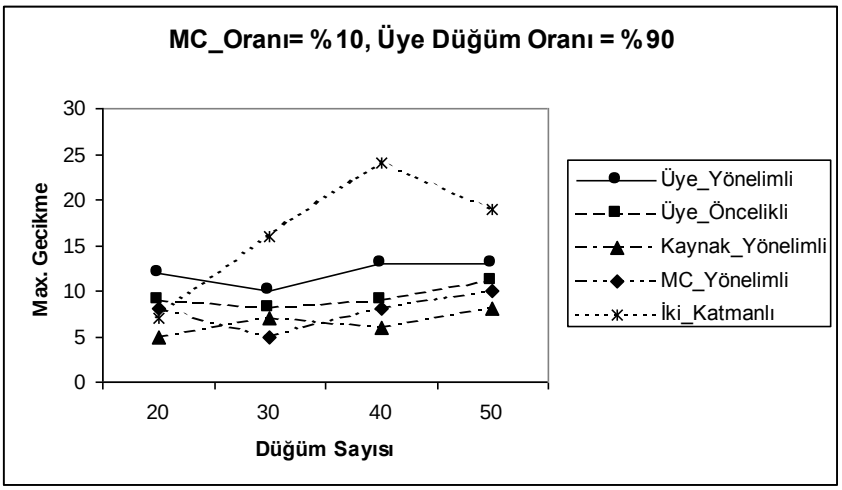
Şekil A.75: Maksimum Gecikmenin Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 50)



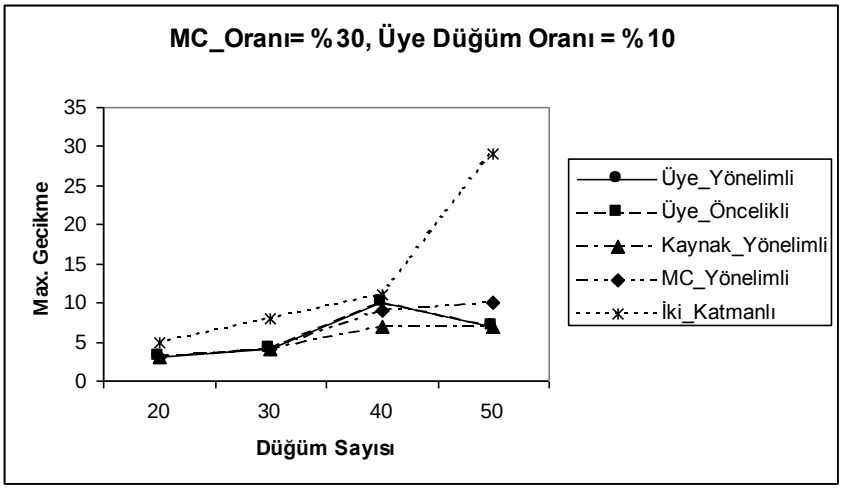
Şekil A.76: Maksimum Gecikmenin Düğüm sayısına göre değişimi (MC Oranı = %10, Üye Düğüm Oranı=%10)



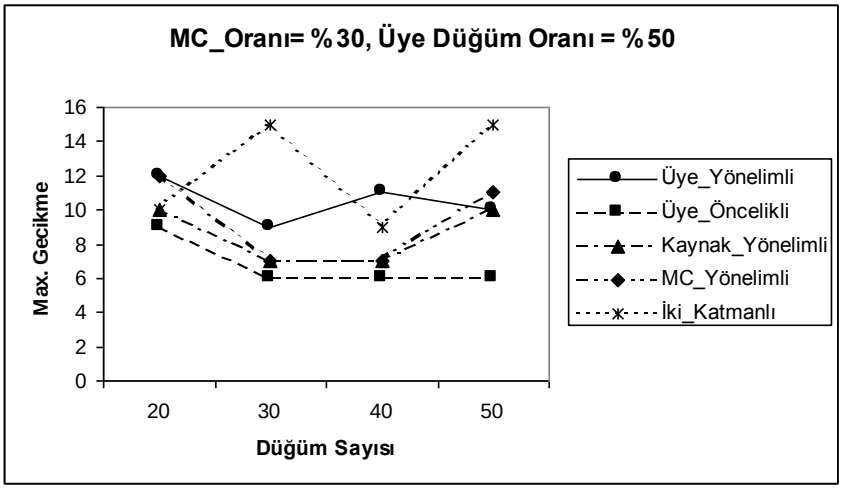
Şekil A.77: Maksimum Gecikmenin Düğüm sayısına göre değişimi (MC Oranı = %10, Üye Düğüm Oranı=%50)



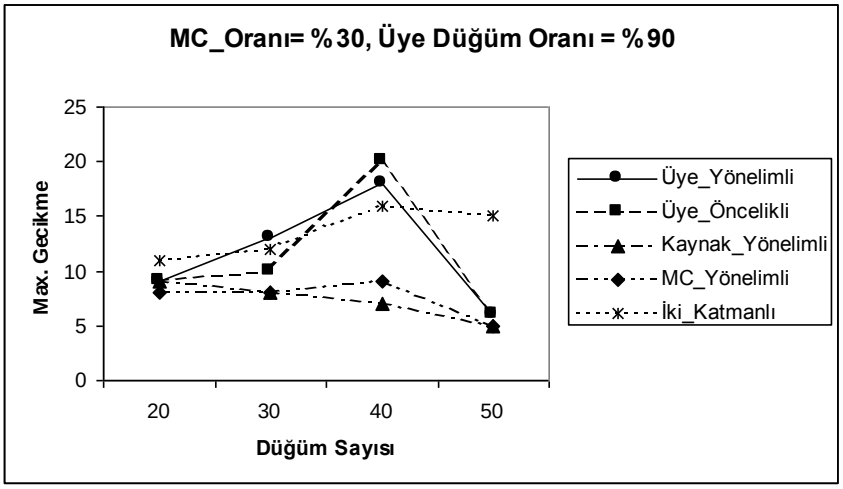
Şekil A.78: Maksimum Gecikmenin Düğüm sayısına göre değişimi (MC Oranı = %10, Üye Düğüm Oranı=%90)



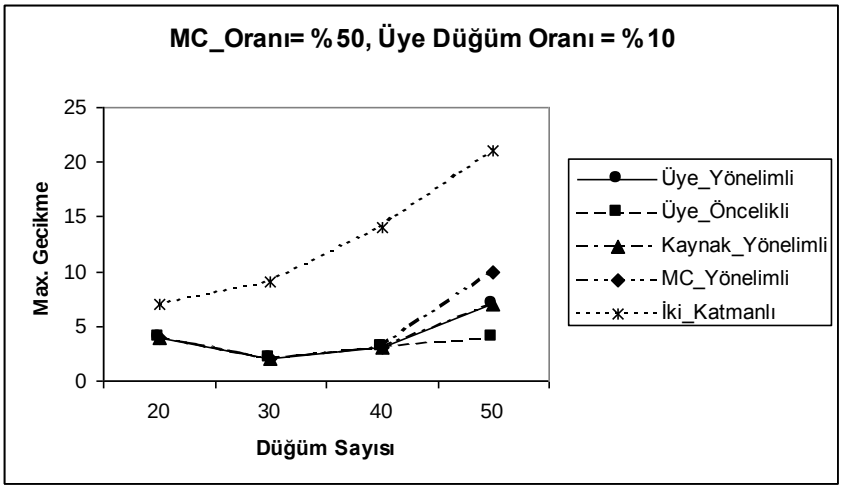
Şekil A.79: Maksimum Gecikmenin Düğüm sayısına göre değişimi (MC Oranı = %30, Üye Düğüm Oranı=%10)



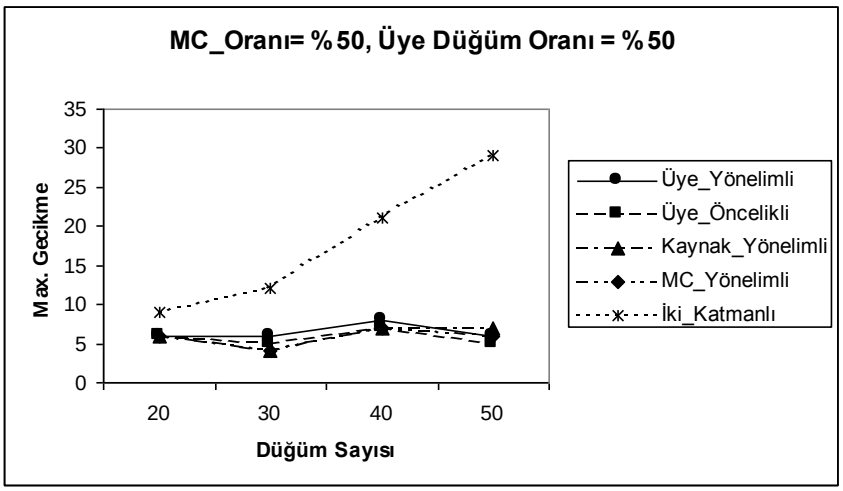
Şekil A.80: Maksimum Gecikmenin Düğüm sayısına göre değişimi (MC Oranı = %30, Üye Düğüm Oranı=%50)



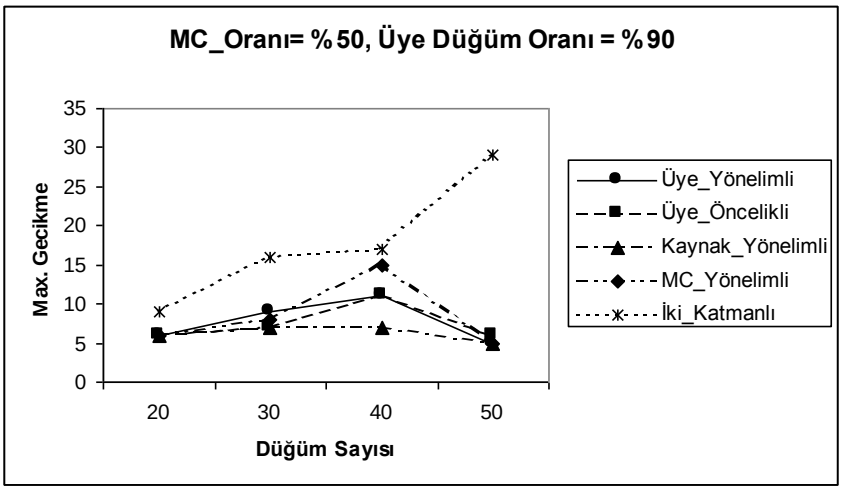
Şekil A.81: Maksimum Gecikmenin Düğüm sayısına göre değişimi (MC Oranı = %30, Üye Düğüm Oranı=%90)



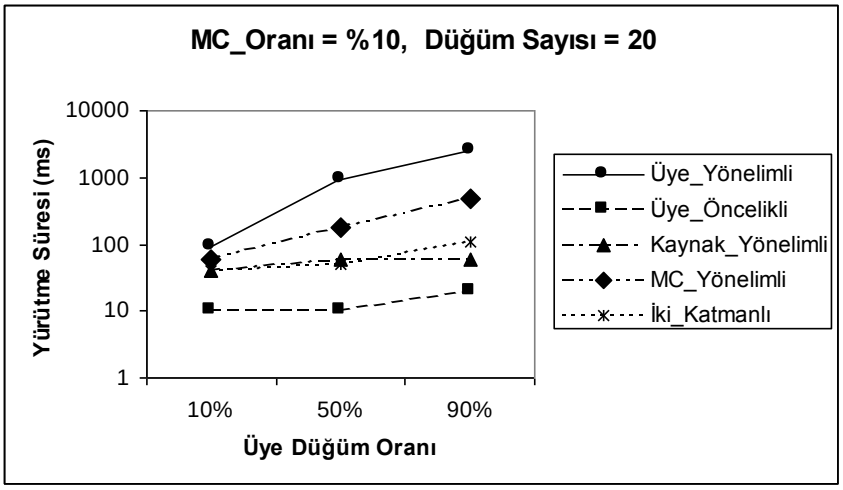
Şekil A.82: Maksimum Gecikmenin Düğüm sayısına göre değişimi (MC Oranı = %50, Üye Düğüm Oranı=%10)



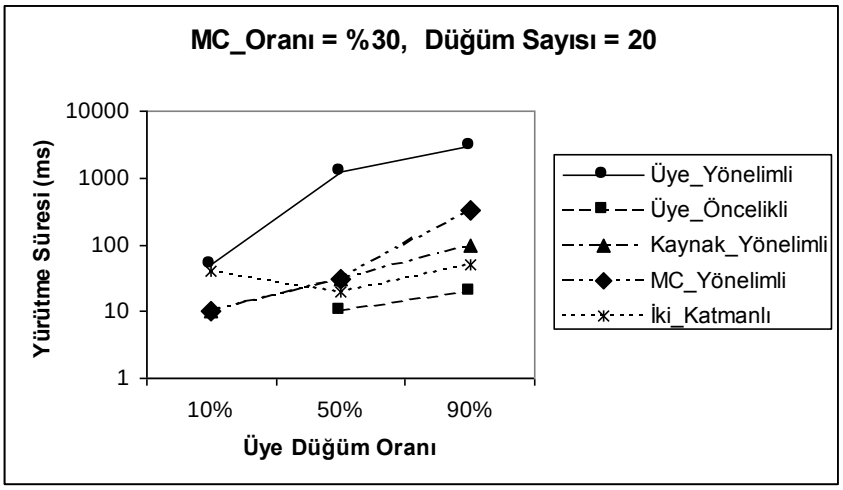
Şekil A.83: Maksimum Gecikmenin Düğüm sayısına göre değişimi (MC Oranı = %50, Üye Düğüm Oranı=%50)



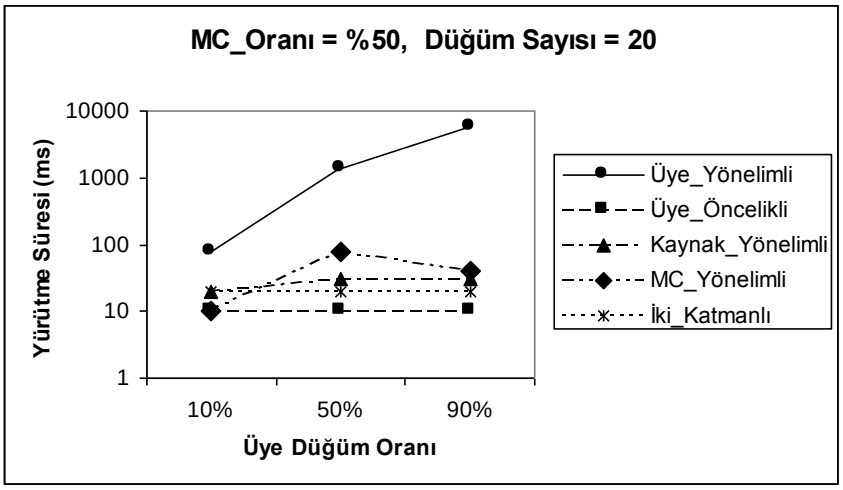
Şekil A.84: Maksimum Gecikmenin Düğüm sayısına göre değişimi (MC Oranı = %50, Üye Düğüm Oranı=%90)



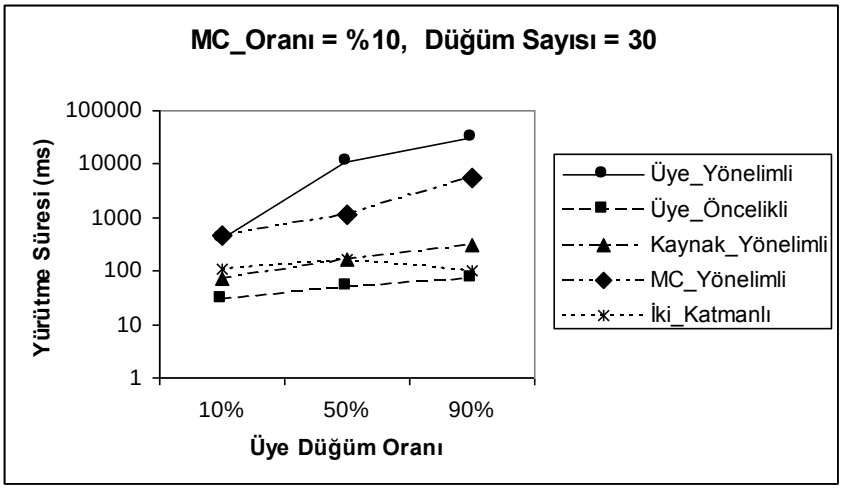
Şekil A.85: Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 20)



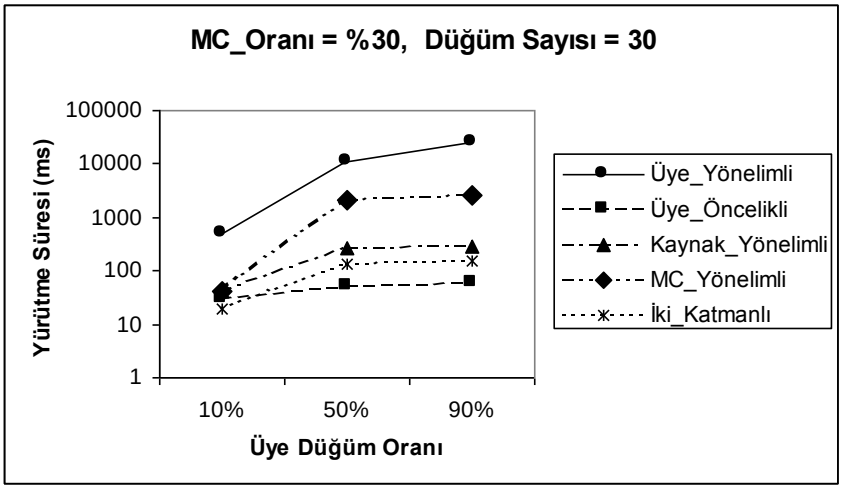
Şekil A.86: Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 20)



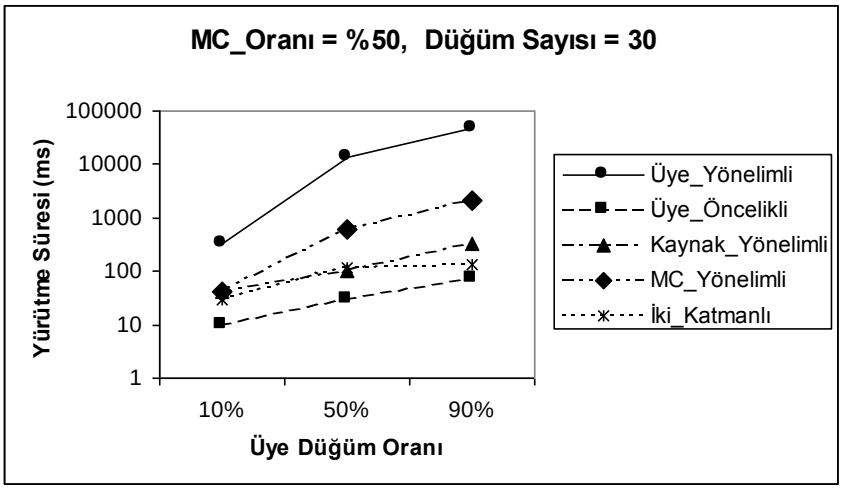
Şekil A.87: Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 20)



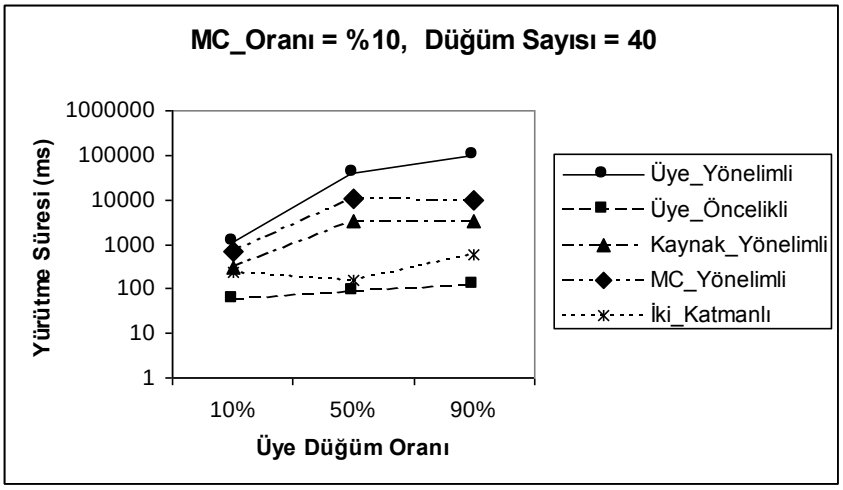
Şekil A.88: Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 30)



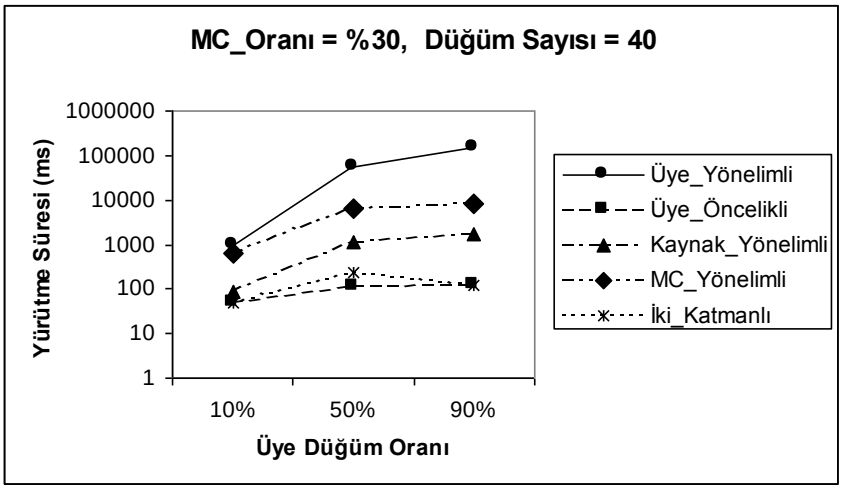
Şekil A.89: Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 30)



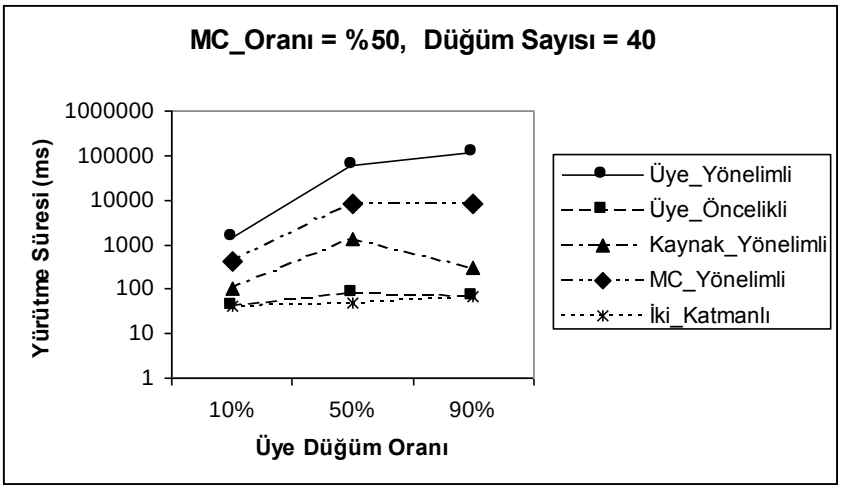
Şekil A.90: Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 30)



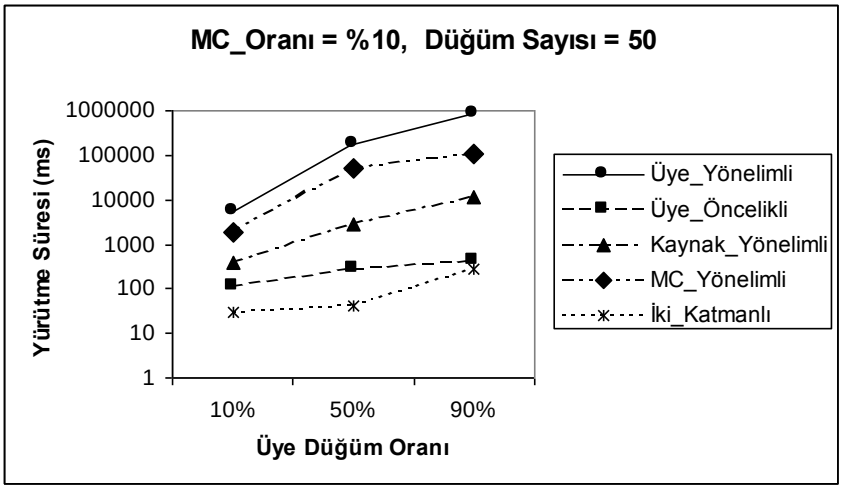
Şekil A.91: Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 40)



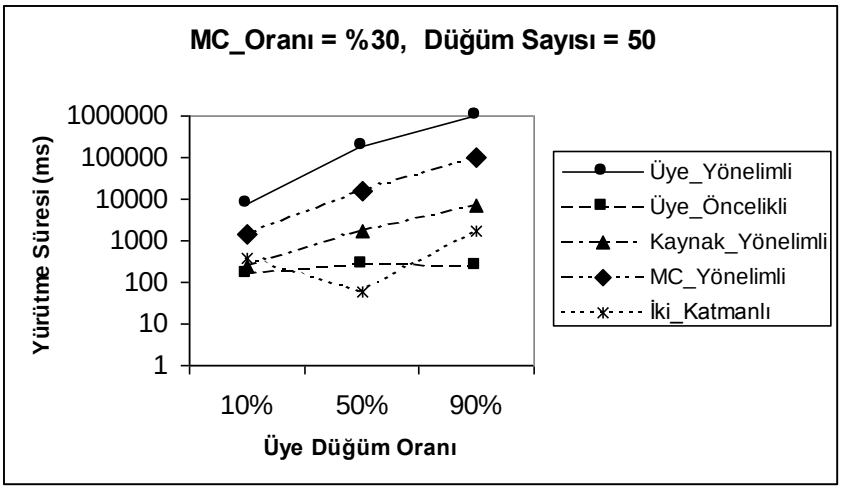
Şekil A.92: Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 40)



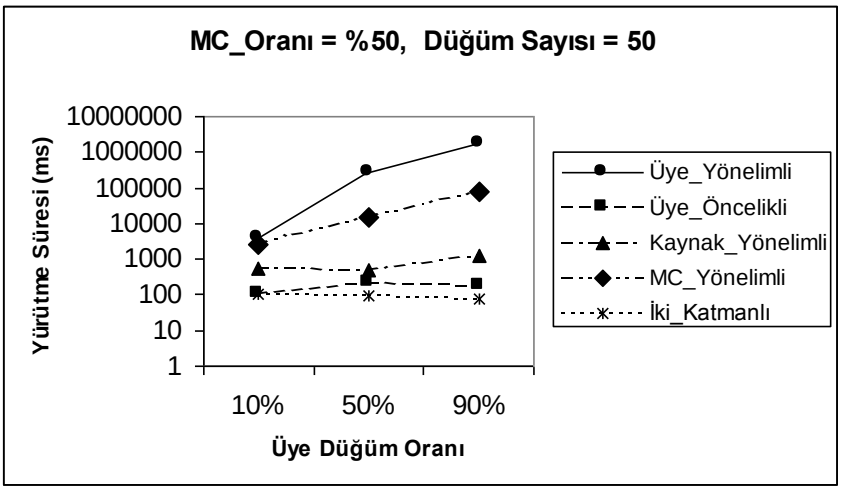
Şekil A.93: Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 40)



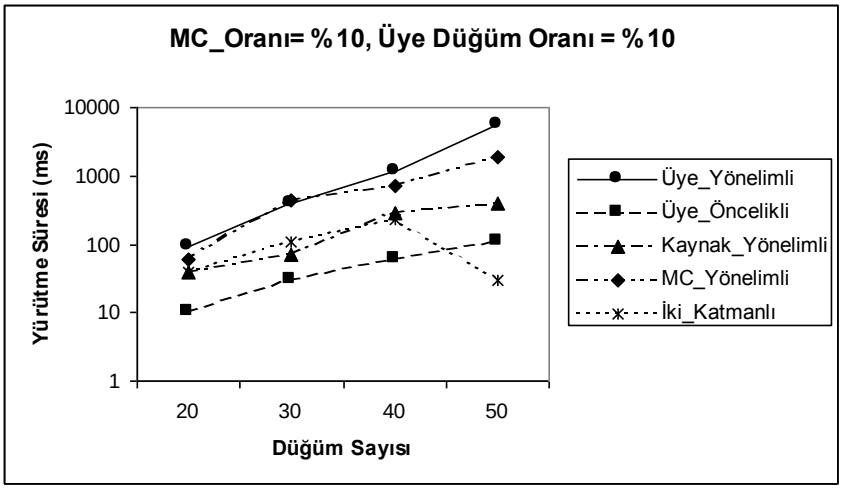
Şekil A.94: Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı = %10, Düğüm Sayısı = 50)



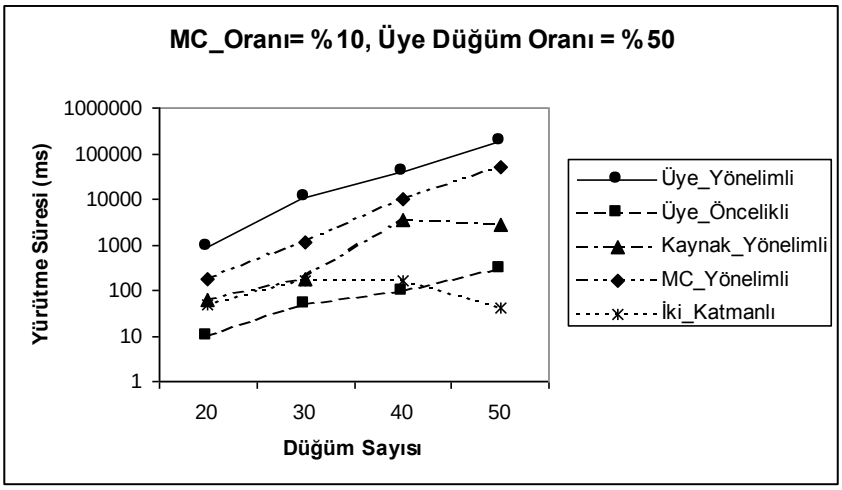
Şekil A.95: Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı = %30, Düğüm Sayısı = 50)



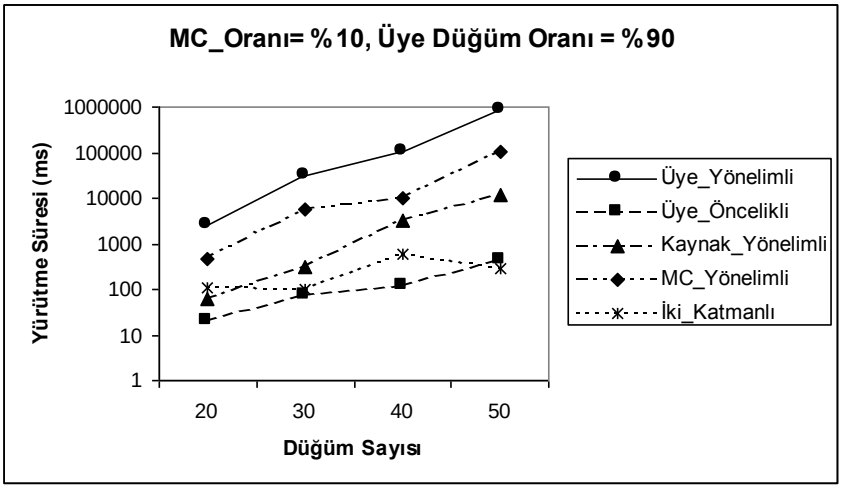
Şekil A.96: Yürütme Süresinin Üye düğüm oranına göre değişimi (MC Oranı = %50, Düğüm Sayısı = 50)



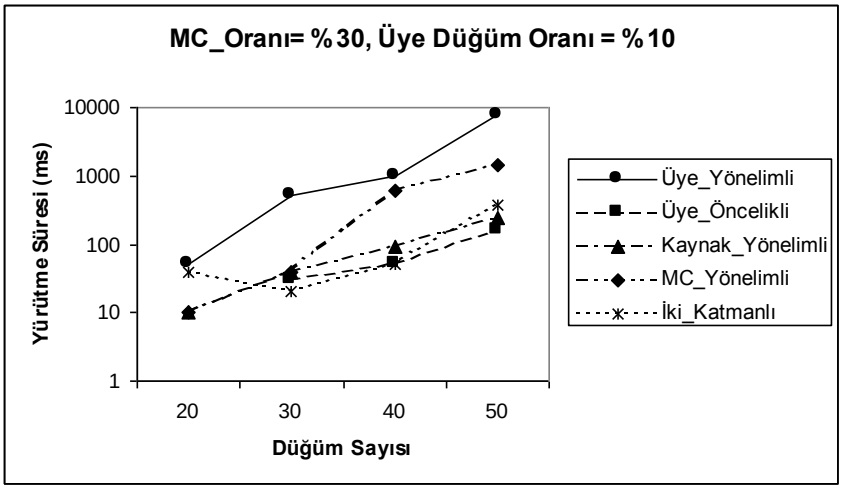
Şekil A.97: Yürütme Süresinin Düğüm sayısına göre değişimi (MC Oranı = %10, Üye Düğüm Oranı=%10)



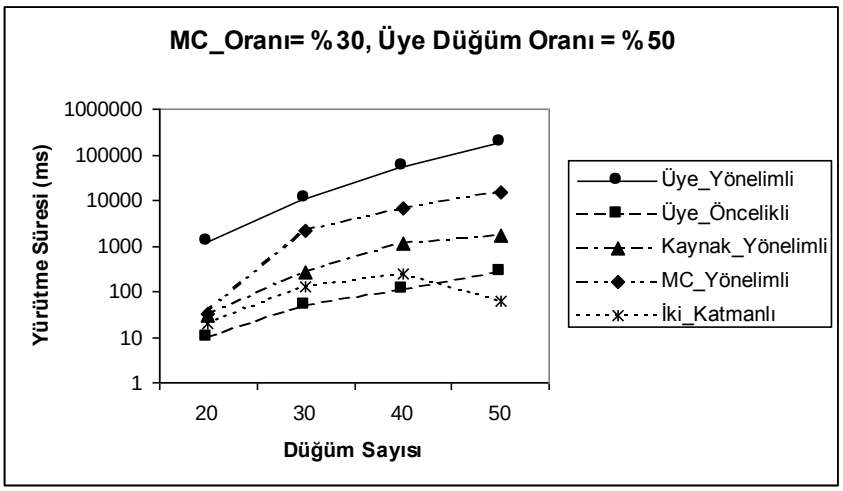
Şekil A.98: Yürütme Süresinin Düğüm sayısına göre değişimi (MC Oranı = %10, Üye Düğüm Oranı=%50)



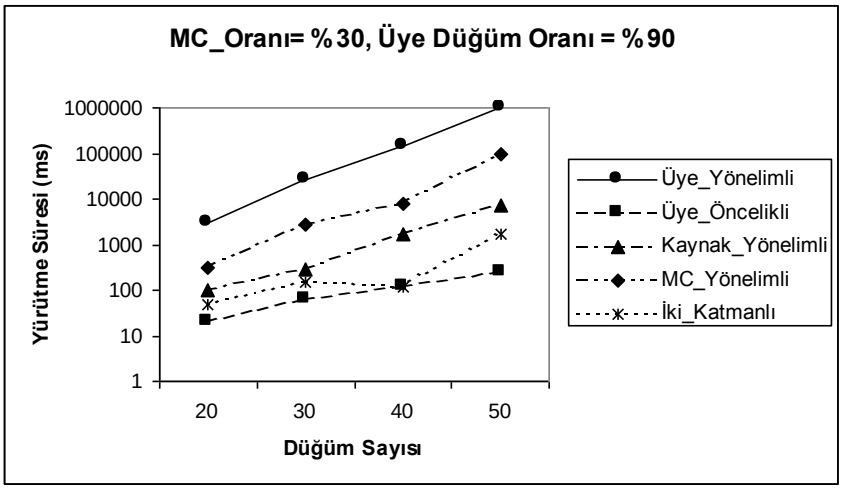
Şekil A.99: Yürütme Süresinin Düğüm sayısına göre değişimi (MC Oranı = %10, Üye Düğüm Oranı=%90)



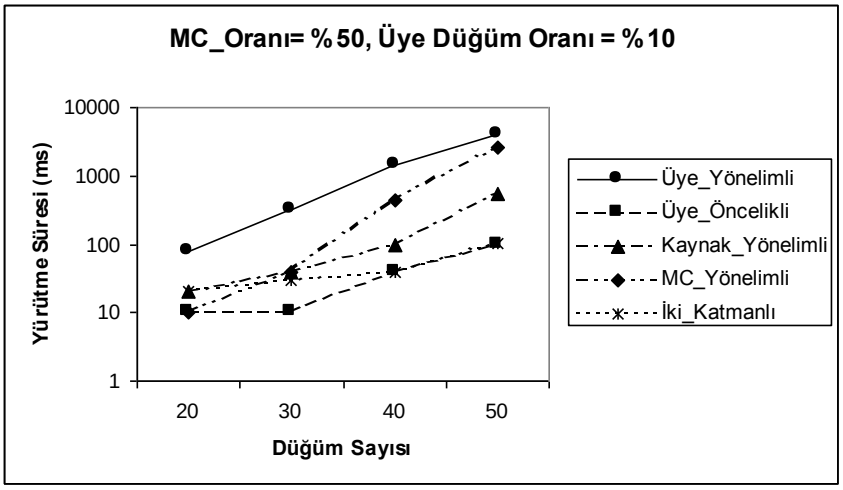
Şekil A.100: Yürütme Süresinin Düğüm sayısına göre değişimi (MC Oranı = %30, Üye Düğüm Oranı=%10)



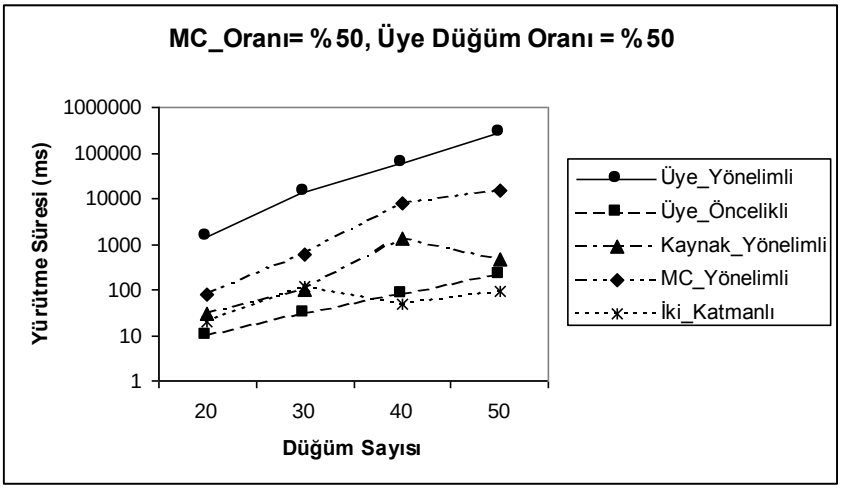
Şekil A.101: Yürütme Süresinin Düğüm sayısına göre değişimi (MC Oranı = %30, Üye Düğüm Oranı=%50)



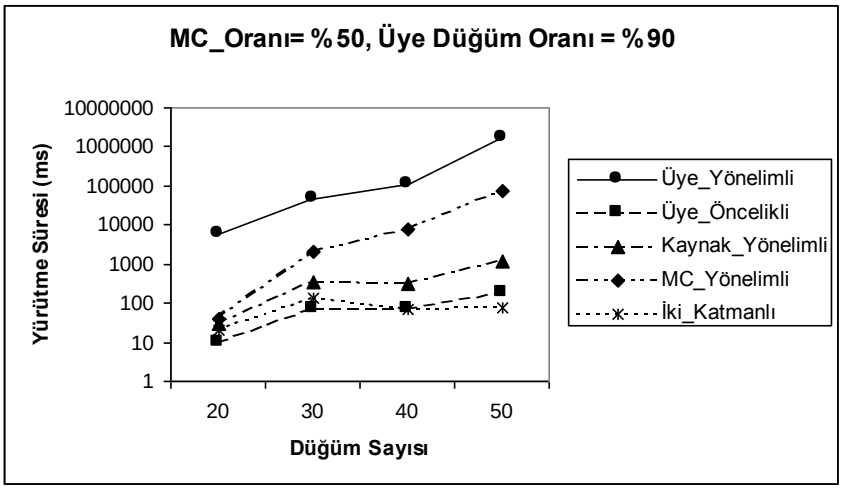
Şekil A.102: Yürütme Süresinin Düğüm sayısına göre değişimi (MC Oranı = %30, Üye Düğüm Oranı=%90)



Şekil A.103: Yürütme Süresinin Düğüm sayısına göre değişimi (MC Oranı = %50, Üye Düğüm Oranı=%10)



Şekil A.104: Yürütme Süresinin Düğüm sayısına göre değişimi (MC Oranı = %50, Üye Düğüm Oranı=%50)



Şekil A.105: Yürütme Süresinin Düğüm sayısına göre değişimi (MC Oranı = %50, Üye Düğüm Oranı=%90)

ÖZGEÇMİŞ

M. Derin HARMANCI, 1978 yılında İstanbul'da doğmuştur. Orta ve lise eğitimini Kadiköy Anadolu Lisesi'nde tamamladıktan sonra, 1996 yılında İstanbul Teknik Üniversitesi Elektronik ve Haberleşme Mühendisliği bölümünü kazanmıştır. 1997 yılında katıldığı ÇAP programı dahilinde Kontrol ve Bilgisayar Mühendisliği bölümünde de öğrenime başlamıştır. 2000 yılında Elektronik ve Haberleşme Mühendisliği , 2001 yılında da Kontrol ve Bilgisayar Mühendisliği bölümlerinden mezun olmuştur. 2001 yılından beri İTÜ Bilgisayar Mühendisliği bölümünde araştırma görevlisi olarak görev yapmaktadır.