Kablosuz Ağlarda Çoğagönderim için Yeni bir Yol Atama Algoritması

A. Sinan Akyürek, Elif Uysal-Bıyıkoğlu
ODTÜ Elektrik ve Elektronik Müh. Bölümü, Ankara
alpersinanakyurek@gmail.com, elif@eee.metu.edu.tr

Öz. Kablosuz ağlarda verimli çoğagönderimin önemli bir problem haline gelmesi, ama öte yandan kablosuz ağ için optimal çoğagönderim ağacı probleminin NP-complete olması, hesaplanabilir iyi höristikleri gerekli kılmıştır. Bu makalede, kablosuz ağlarda verimli gönderim için yüksek başarımlı yeni bir dağıtık algoritma sunulmaktadır.En fazla $O(N^3)$ hesaplama karmaşıklığındaki bu algoritmanın kurduğu çoğagönderim ağacının uzunluğunun (daha doğrusu, yapraklara ulaşmak için yapılması gereken iletim sayısının), nod sayısı (N) ve çoğagönderim grubu büyüklüğü (M) ile artış hızının bilinen bir optimal altsınıra çok yakın olduğu benzetimler ile gözlenmiştir. Ayrıca, ağacın kurulumu için nodlar arası iletilecek mesaj miktarının da küçük olduğu gözlenmiştir. Topoloji değişikliklerine adaptasyon için de bir bakım ve tamir algoritması önerilmiştir.

Abstract As multicast transmission is gaining importance in the wireless network setting, the NP-complete nature of the optimum wireless multicast tree problem makes it necessary to develop efficient and computable heuristics. In this paper, we present a new distributed routing algorithm. The length of the multicast tree computed by the algorithm (more precisely, the number of transmissions needed to reach all multicast nodes) as a function of the number of nodes, N, and the size of the multicast group, M, is very close to a lower bound provided in the literature, while the computational complexity is at most $O(N^3)$. Moreover, the number of messages that need to be exchanged to establish the tree is reasonable. A repair and maintenance algorithm has also been suggested for adaptation to topology changes, which tend to occur in practical applications.

I. GİRİŞ

Kablosuz ve özellikle ad-hoc ağlarda verimli yol atama (routing) enerji ve dolayısıyla çevrimin ömrü açısından önemlidir. Bir kaynaktan tek bir terminale gönderim (unicast) için dağıtık ve polinom-zamanlı optimal yol atama algoritmaları iyi bilinmektedir (ör. Prim-Dijkstra, bkz. [1]). Yine, bir kaynaktan diğer tüm terminallere en verimli yolun bulunması (broadcast problemi), yani bir çizgedeki bütün düğümleri içeren en düşük ağırlıklı ağacın bulunması problemi (minimum-weight spanning tree- MST), polinom kompleksiteye sahiptir ve dağıtık çözumleri vardır (ör. Bellman-Ford.)

Ama, bir kaynaktan aynı verinin birden çok alıcı terminale iletilmesi, yani çoğagönderim (multicast) için polinom zamanlı optimal algoritmalar yoktur. Bunun nedeni, bütün düğümleri değil de *düğümlerin belli bir altkümesini* kapsayan en kısa (ya da en düşük ağırlıklı) ağacı bulma probleminin, yani Steiner Tree probleminin, NP-complete olmasıdır. Steiner Tree probleminin ve MST probleminin farkı 1-5 numaralı şekillerde bir örnek üzerinde açıklanmıştır. Temel olarak, Steiner Ağacı, toplam ağırlığı azaltmak için gerekirse kapsanması gerekmeyen ek nodlar da kullanma özgürlüğüne sahiptir.

Bu algoritma ASELSAN Haberleşme Cihazları Grubu, ve TÜBİTAK Kariyer proje no. 106E119 tarafından desteklenmiştir.

978-1-4244-1999-9/08/$25.00 ©2008 IEEE

Steiner Tree problemi ile ilişkili olan Kablosuz Çoğagönderim Ağacı (KCA) problemi de dolayısıyla karmaşıklığı yüksek bir problemdir. Öncelikle, KCA problemini tanımlayalım:

Kablosuz Çoğagönderim Ağacı (KCA) problemi: Verilmiş bir multicast kümesindeki bütün düğümlere ulaşan, toplam *gönderim sayısının* en küçük olduğu ağacı bul.

Yukarıda kullanılan *ulaşma* kavramını şöyle tanımlıyoruz: Nod kümesi V ve kenar kumesi E ile birlikte verilen $G(V, E)$ çizgesinde, bir $U \in V$ kümesinin, bir $M \in V$ kümesine ulaşması demek, her $m \in M$ için, $(v, m) \in E$ olacak şekilde bir $v \in V$ bulunabilmesi demektir.

Yukarıda, *gönderim sayısı*, bilginin kaynaktan, ağacın uçlarına dek iletilmesi için, ağaç üzerindeki terminallerin yapacakları toplam veri iletimi (transmission) sayısını ifade etmektedir. Ayrıca, Steiner ağacı problemindekinin aksine, KCA probleminde, kurulacak ağacın multicast kümesindeki nodları kapsaması değil, onlara ulaşması aranmaktadır. Bu, kablosuz iletimin doğal çoğagönderim özelliğinden kaynaklanmaktadır: Çoğagönderim ağacının ucundaki bir noda (bir yaprak noduna) komşu olan bütün nodlar, bilgi akışı üzerindedir. Bu nedenle, KC ağacının bütün multicast nodları içine alacak şekilde büyümesi gerekmez: bütün multicast nodlarına en az bir hop uzaklığa gelmesi yeterlidir.

KCA probleminin çözümünden, Steiner Ağacı probleminin çözümü de oluşturulabileceğinden, KCA probleminin de NP-complete olduğunu görüyoruz. Ayrıca, KCA probleminin özel bir durumu olan kablosuz broadcast tree probleminin de Çizge Kuramında NP-hard olarak bilinen "Minimal Connected Dominating Set" problemine eşdeğer olduğunu gözlemliyoruz. Bu durumda, KCA probleminin optimal çözümü pratikte kullanışlı olmayacaktır, ve optimale yakın, polinom zamanda hesaplanabilen çözümlere gereksinim vardır. Bu makalede bu özellikleri yüksek başarımla sağlayan bir algoritma sunuyoruz.

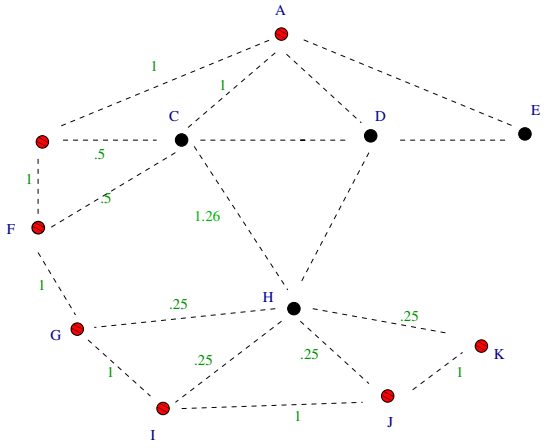

Fig. 1. Örnek bir çevrim: A düğümü kaynak, geri kalan kırmızı düğümler {B, F, G, I, J, K} çoğagönderim kümesidir. Kesikli siyah çizgiler olası bağlantıları gösteriyorlar, bağlantıların ağırlıkları yeşil sayılar ile gösteriliyor.

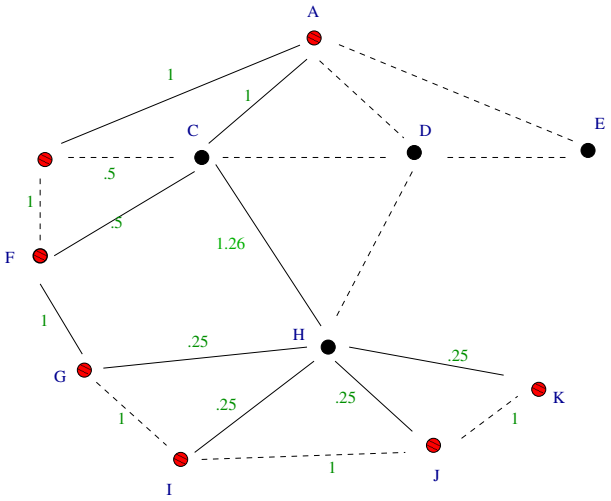

Fig. 2. Optimal unicast yollar birleştirildiğinde elde edilen yollar siyah çizgiler ile gösteriliyor. Eğer çoğagönderim bu yollar üzerinden yapılırsa (ve kablosuz çogagönderim avantajını kullanmak için mükemmel senkronizasyon da gerçekleştirilirse), veri toplam 4 gönderim ile multicast grubunun tamamına gönderilmiş olur. Bu durumda, A, C, F ve H nodları gönderici durumundadır.

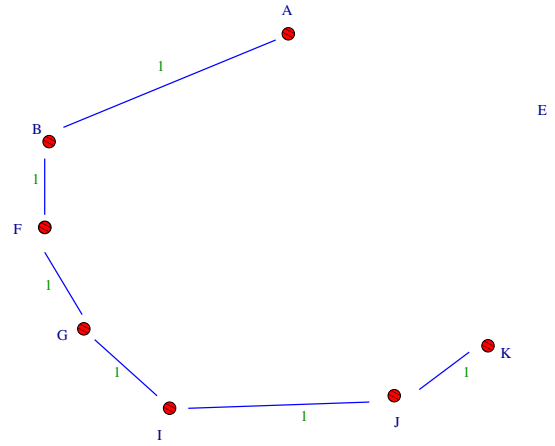

Fig. 3. Kırmızı nodları kapsayan MST: toplam ağırlığı 6.

Literatürde yayınlanmış kablosuz çoğagönderim höristikleri vardır, ör. [2], [3], [4], [5]: bunların hesaplanma bakımından basit olanlarının başarım açısından düşük olduğunu, ya da tam tersini görüyoruz.

II. KCA ALGORİTMASINDA İSTENEN ÖZELLİKLER

Algoritmanın başarımı ve verimliliğini ölçerken aşağıdaki üç kriter bakımından değerlendireceğiz:

1) KCA dağıtık ve verimli şekilde kurulabilmelidir. Yani, nodların KCA'yı kurmak için birbirlerine göndermesi gereken mesaj sayısı fazla olmamalıdır. Gereken mesaj sayısının nod sayısı ve multicast grubu büyüklüğü ile artışına bakılacaktır.
2) Nodlarda ağı kurmak için gereken hesaplama karmaşıklığı önemlidir. İdeal olarak $O(N^3)$ olmalıdır.
3) Ortaya çıkan ağaç (ya da altçizge)nin toplam ağırlığı (veya üzerindeki gönderici sayısı) düşük (optimale yakın) olmalıdır. Yine, bunun nod sayısı ve multicast grubu büyüklüğü ile artışına bakılacaktır.

Bunlara ek olarak, topoloji değişikliklerine çabuk yanıt veren bir modifikasyon algoritması olması, pratik uygulamalar için önemlidir.

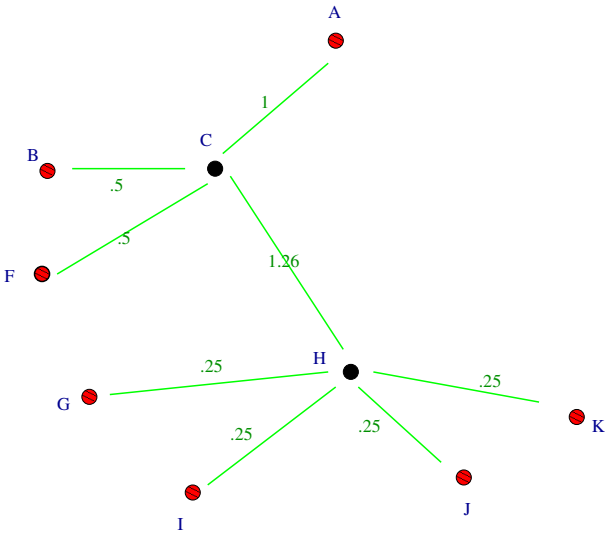

Fig. 4. Kırmızı nodları kapsayan Minimum-Ağırlıklı Steiner Ağacı. ağırlığı 4.26. C ve H nodlarını kullanarak MST'den daha düşük ağırlık elde ediyor.

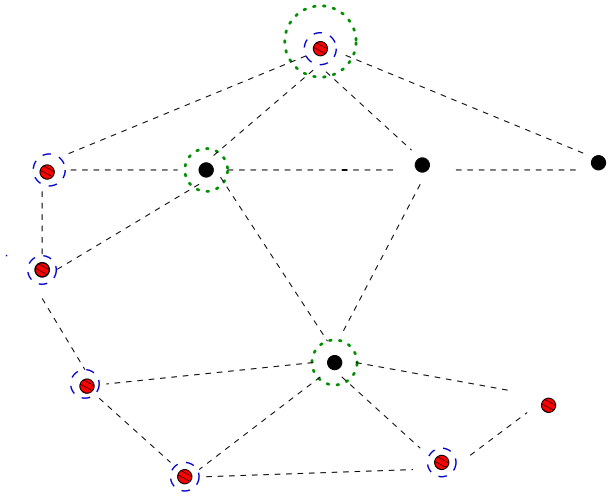

Fig. 5. MST'de gönderici görevi yapan düğümler mavi halka içine alınmış. Kablosuz Steiner Ağacı üzerindeki göndericiler ise yeşil halka içine alınmış. Görüldüğü gibi, Steiner Ağacı üzerinde iletim sadece 3 düğümün iletim yapmasını gerektiriyor: A, C and H.

III. ALGORİTMA

Algoritma anlatılırken kullanılacak olan bazı terimler aşağıda açıklanmıştır:

- M = multicast grubu (multicast verinin alıcısı olan nodlar kümesi)
- $h =$ Bir nodun hop sayısı cinsinden kaynağa uzaklığı
- $broadcast(bilgi)$ Bütün komşulara bilgiyi gönder
- $gönder(adres, bilgi)$ Adresi verilen komşuya, bilgiyi gönder

Bütün nod'lar için $h = NULL$ olarak tanımlanır.

A. İlk Bilgi Gönderimi

Kaynak, bilgi akışını başlatır.
$broadcast(h = 1)$ (komşulara hop=1 ata)
$broadcast(M)$ (multicast grubu M'yi gönder)
Geri kalan nod'lar kendilerine gelen hop sayısını bir artırıp komşularına gönderirler:
$ndeğişti = YANLIŞ$

$Eğer\ h == NULL$
$h = alınan$
$broadcast(h + 1)$
$broadcast(M)$
$ndeğişti = DOĞRU$
$Değilse$
$Eğer\ h > alınan$
(eğer alınan bilgi daha küçükse onu kullan)
$h = alınan$
$alınanbilgi(alınanadres) = alınan$
$broadcast(h + 1)$
$ndeğişti = DOĞRU$
$Bitir$
$Bitir$
$Eğer \quad ndeğişti$
$tabloyugüncelle(n, alınanadres)$
$Bitir$

B. *tabloyugüncelle fonksiyonu*

Her nod, komşuları ile ilgili gelen bilgileri tablolarında saklarlar. $tabloyugüncelle(h, alınanadres)$

$tüm\ i \in alınanadres$
$Eğer\ alınanbilgi(i) > h + 1$
$tablo(i) = üst$
$Değilse$
$Eğer\ alınanbilgi(i) == h + 1$
$tablo(i) = aynı$
$Değilse$
$tablo(i) = alt$
$Bitir$
$Bitir$
$Sonraki$ i

C. *"Alt, Üst, Aynı" kavramları*

Her nod'un komşuları, kaynağa göre bulundukları hop sayısına göre 3 gruptan birindedir:

1) Kaynağa daha yakın=Alt
2) Kaynağa aynı uzaklıkta=$Aynı$
3) Kaynağa daha uzak=$Üst$

Daha detaylı açıklama aşağıdaki grafiklerde yapılmıştır.

D. *Multicast Grubu Komşuları Bilgisi*

Kaynağa en uzak nod'lar, yani $yaprak$ nod'lar geri bilgi gönderimini başlatırlar:

$Eğer\ Tümkomşular == Alt \cup Aynı$
$broadcast(yaprak)$ (yaprak nod olduğunu açıkla)
$Bitir$

$yaprak$ komutunu alan her nod:
$Eğer\ alınanbilgi == yaprak$ ve $alınanadres == üst$
$multicasttablo(alınanadres) =$
$alınanmulticastbilgisi$
$Bitir$
$Eğer\ tümüstlerdenbilgialındıysa$
$broadcast(multicasttablo)$

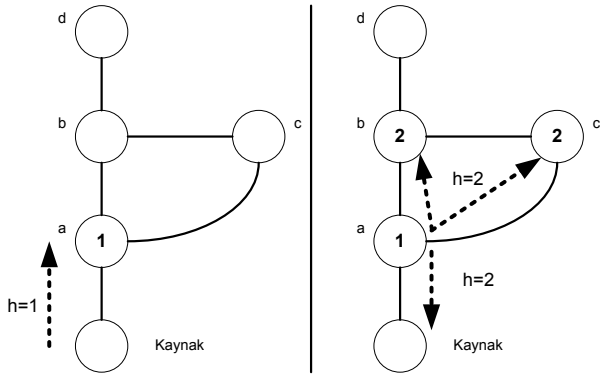

Fig. 6. Kaynak komşularına 1 hop uzaklıkta olduklarını belirtmek için $h = 1$ gönderir. Bu bilgiyi alan a komşularına 2 hop uzaklıkta olduklarını belirtmek için $h = 2$ bilgisini gönderir.

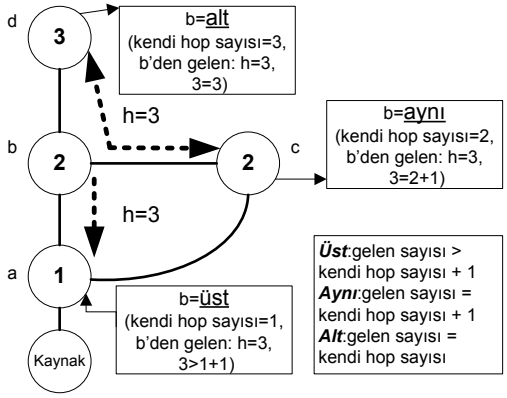

Fig. 7. b komşularına $h = 3$ gönderir. d aldığı bu bilgi ile kendi hop sayısını(3) karşılaştırır ve b'yi tablosunda alt olarak tanımlar. c, kendi hop sayısı 2 olduğu için b'yi tablosunda aynı olarak tanımlar. a, kendi hop sayısı 1 olduğu için b'yi tablosunda üst olarak tanımlar.

$broadcast(yaprak)$
$Bitir$

Ağdaki bütün nod'lar hangi $üst$ nod'u kullanarak hangi multicast grup üyelerine ulaşabileceklerini ve kaynağa olan hop sayısını biliyorlar.

E. *Seçme ve Budama*

Bütün nod'lar, veri iletiminde hangi üst komşularını kullanacaklarını belirlemeleri gerekiyor. Bunun için Kaynak, Seçme ve Budama işlemini başlatır.
$multicasttablo$ bilgisini alan her nod bu $seçbuda$ fonksiyonunu çalıştırır ve göndermesi gereken nod'ları belirler.

F. *Seç − Buda fonksiyonu*

$görülenmulticast = multicasttablo$
$görülenmulticast$ boş olmadığı sürece

$tüm$ i $\in multicasttablo$
$ağirlik(i) = büyüklük(i)$
$Eğer$ i $\cap seçilecekmulticast == i$
$ağırlık(i) = ağırlık(i) + 1$
$Bitir$
$görülenmulticast = görülenmulticast$
$-multicasttablo(maksimum(ağırlık))$
$seçilen = seçilen \cup maksimum(ağırlık)$
$gönder(maks.(ağırlık), multicasttablo(maks.(ağırlık)))$

Her seferinde tablodan en fazla kişiyi göreni seç ve bu seçilen kişiye, kişinin gönderim yapması gerekenlerin listesini gönder.

IV. ALGORİTMANIN DOĞRULUĞU

Bütün Multicast grubu elemanlarını, Kaynak'a götürebilecek bir yol olduğundan ve bu yolların Alt komşu dizileri şeklinde tanımlanabileceklerinden dolayı, Algoritma sonucu daima bütün Multicast grubunu içerecektir.

V. ALGORİTMANIN KARMAŞIKLIĞI

Algoritmada hesaplama karmaşıklığını belirleyen nokta, bir nod'un hangi nod'a göndereceğini seçme aşamasında gerçekleşmektedir (set-cover). Seçme algoritması seçebileceği bir kümeden her seferinde en büyük ağırlığa sahip olan komşusunu seçecek. Bu işlem için bütün kümeyi taraması gerekecektir. Bu işlemi tüm multicast grup elemanlarını kapsayana kadar yapacak. Yani tek bir nod'un seçme karmaşıklığı $O(N^2)$ olacaktır. N tane nod bu işlemi uyguladığından tüm algoritmanın karmaşıklığı en fazla $O(N^3)$ olacaktır.

VI. ALGORİTMANIN BAŞARIMI

Bu bölümde, algoritmanın başarımını, Giriş bölümünde belirtilen kriterler bakımından inceleyeceğiz: Başta gelen performans ölçütü, ortaya çıkan ağacın uzunluğu (yani her bir bilginin kaç iletim yaparak bütün alıcılara ulaştırılacağı). Bunun yanı sıra mesaj uzunluğu, hesaplama karmaşıklığı, ve son olarak, topoloji değişikliklerine uyum bakımından algoritmayı inceleyeceğiz.

Öncelikle, benzetim altyapısını tarif edelim. Algoritmayı diğer var olan algoritmalarla karşılaştırabilmek için aynı ağ yapısını kullandık. Bu ağ birim çember içine rastgele yerleştirilmiş nod'lardan oluşuyor. Eğer iki nod arası uzaklık 0.286'dan küçükse bu iki nod birbirlerini görebiliyorlar. Multicast grubu tamamen rastgele seçiliyor. Diğer algoritmalara göre farklı olarak ağ yapısı toplam 4 hop'ta sabitlendi. İki çeşit simülasyon yapıldı. Birincisinde toplam nod sayısı 50'de sabitken, multicast grubundaki eleman sayısı arttırılıyor. İkincisinde tüm nod'lar multicast grubu elemanı yapılıyor (broadcast) ve toplam nod sayısı değiştiriliyor. Her iki çeşit simülasyon için 15000 farklı rastgele dağıtılmış topoloji kullanılmış ve sonuçların ortalaması alınmıştır.

VII. TOPOLOJİ DEĞİŞİKLİKLERİNE UYUM

Dinamik ağlarda bazı bağlantılar kopabilir, ya da yeni bir nod ağa dahil olmak isteyebilir. Bu algoritma da bunu destekleyecek şekilde üretilmiştir. Eğer bir Nod'un Üst veya Alt komşusu değişirse tamir algoritması çalışır.

A. Tamir Algoritması

$broadcast(kendiMulticastgrubumuarıyorum)$
$broadcast(n = 1)$ (komşularına hop sayısını 1 atayarak en kısa yol aranacak)
Alıcı Multicast Grubu Üyesi ise:
$gönder(alınanadresi, n$ =(Kendi hop sayısı)+(Kendisine gönderilen tamir hop sayısı))
Değilse:
$n = n + 1$; $broadcast(kendiMulticastgrubumuarıyorum)$
$broadcast(n)$
Geri dönen cevaplardan en düşük hop sayısına sahip olan seçilir.

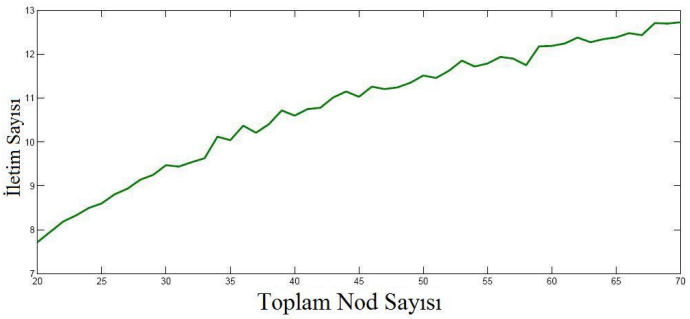

Fig. 8. Broadcast İletim Sayısı: Ağ'daki tüm noldara iletmek için gerekli toplam iletim sayısının, ağ'daki toplam nod sayısı ile bağlantısı gösterilmiştir.

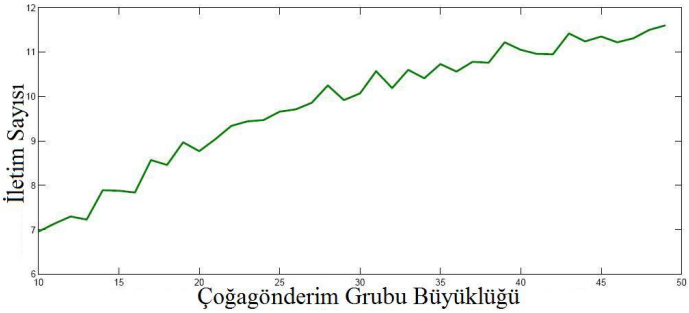

Fig. 9. Multicast İletim Sayısı: Belirli bir çoğagönderim grubuna iletmek için gerekli toplam iletim sayısının, çoğagönderim grubunun büyüklüğü ile bağlantısı gösterilmiştir. Ağdaki toplam nod sayısı 50'de sabitlenmiştir.

B. Yeni Nod Katılma Algoritması

Unicast(bire bir iletim) yolu sayesinde Kaynak ile görüş. Eğer kaynak yeni nod'un Multicast grubuna dahil olmasını isterse, yeni nod'a Multicast Grup numarasını gönderir ve yeni nod, Tamir Algoritması'nı çalıştırır.

VIII. SONUÇ

Simülasyon ve hesaplamalardan da görüldüğü gibi algoritma, yüksek performans ve düşük hesaplama karmaşıklığı ile çalışmaktadır. Kablosuz Çoğagönderim Ağacı problemine tam ve hızlı bir çözüm sağlamaktadır. Bu algoritmayla ilgili gelecekte yapılması gereken çalışma, algoritmanın MAC katmanıyla uyumu ve tamir algoritmasının performansının denenmesi olacaktır.

REFERENCES

[1] D. Bertsekas, R. Gallager, "Data Networks", Prentice Hall, 1996.
[2] B. Tavli and W. Heinzelman, "MH-TRACE" http://www.ece.rochester.edu/research/wcng/projects/trace.htm
[3] Aleksi Penttinen, "Minimum cost multicast trees in ad hoc networks", Alexi Penttinen, IEEE ICC 2006, pp. 3676-3681.
[4] Nguyen, Ephremides, Distibuted Algorithms for Energy-Efficient Broadcasting In Ad-Hoc Networks, pp. 820-825.
[5] Nguyen, Ephremides, On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks, IEEE INFOCOM 2000, pp. 585-594.