

A New CR-Chord Based P2P Protocol in the Presence of Malicious Nodes

Musa Milli
Ege University, Faculty of Engineering, Izmir, Turkey
E-mail: musamilli@gmail.com

Abstract

As is known, P2P networks work differently from traditional server-client architecture. Each node can act both a server and a client role. Traditional structural P2P systems are not architectures developed with security in the foreground. Therefore, there are many security vulnerabilities, and when malicious nodes are present in the system, P2P can adversely affect the functioning of the system. With the proposed method, secure nodes are kept in a specific table, taking into account the security gaps in structured peer-to-peer (P2P) networks, and subsequent queries are made over these secure nodes. In addition to the CR-Chord algorithm, half cycles were used to find more reliable cycles in dense malicious environments and increase lookup performance. This enabled P2P network to operate more robust. The lookup performance of the developed method was compared with Cyclic-Routing Chord algorithm in OverSim simulation environment. According to the obtained results, the developed Half Cyclic Routing Chord (HCR-Chord) algorithm was observed to give better results.

Keywords: P2P Networks, Chord, Malicious Node, HCR-Chord

DOI: 10.7176/JSTR/5-12-31

Kötü Niyetli Düğümlerin Varlığında CR-Chord Tabanlı Yeni Bir P2P Protokolü

Özet

Bilindiği üzere P2P ağlar geleneksel sunucu-istemci mimarisinden farklı çalışmaktadırlar. Her bir düğüm hem sunucu hem de istemci rolü alabilmektedir. Geleneksel yapısal P2P sistemler güvenlik ön planda tutularak geliştirilmiş mimariler değildir. Bu nedenle birçok güvenlik açıkları bulunmaktadır ve kötü niyetli düğümler sistemde yer aldıklarında P2P sistemin işleyişini olumsuz yönde etkileyebilmektedir. Önerilen yöntem sayesinde yapısal görevdeş (P2P) ağlarda güvenlik açıkları göz önünde bulundurularak güvenli düğümler belirli bir tabloda tutulacak ve daha sonraki sorgular bu güvenli düğümler üzerinden yapılacaktır. CR-Chord algoritmasının yanında yarım döngüler de bularak yoğun kötücül ortamlarda daha fazla güvenilir döngüler ile arama başarımları artırılmıştır. Böylece P2P ağın daha güvenilir bir şekilde çalışması sağlanmıştır. Geliştirilen yöntemin arama başarımları Cyclic-Routing Chord algoritması ile OverSim simülasyon ortamında karşılaştırılmıştır. Elde edilen sonuçlara göre geliştirilen Half Cyclic Routing Chord (HCR-Chord) algoritmasının daha iyi sonuçlar verdiği gözlenmiştir.

Anahtar Kelimeler: P2P Ağlar, Chord, Kötücül Düğüm, HCR-Chord

1. Giriş

Üstel hız ile artan verileri ekonomik ve hızlı bir şekilde hem işlemek hem de depolayıp gerektiğinde servis etmek geleneksel sunucu-istemci mimarisinin üstesinden gelmekte zorlandığı önemli bir sorundur. Son zamanlarda bu sorun ile baş etme konusunda başarılı uygulamalara imza atan P2P sistemler araştırmacıların ilgisini çekmeyi başararak sunucu-istemci mimarisine alternatif sistemler olmuşlardır. P2P sistemlerin merkezi otoriteden yoksun doğası gereği (Sedorf, 2006 ; Palomar et. al., 2006 ; Douceur, 2002 ; Moro et. al., 2003 ; Agushinta et. al., 2009 ; Ayyasamy and Sivanandam, 2010) çalışma biçimleri geleneksel sunucu-istemci mimarisinden farklıdır. P2P ağlar katılımcılarının bazen sunucu, bazen istemci

rolü üstlendiği dağıtık sistemlerin bir alt sınıfıdır. Dolayısı ile P2P katılımcıları hem kaynağı sağlarlar hem de tüketirler. Bu sebeple geleneksel sunucu-istemci modelinden farklı olarak P2P ağlarda her bir düğüm eş fonksiyoneliteye sahip olduğundan kaynaklardan da (bant genişliği, hesaplama gücü ve depolama alanı v.b.) faydalanma konusunda eşit haklara sahiptirler (Shen et. al., 2014). Bir düğüm ağdaki başka bir katılımcı düğüm ile iletişim kurarken bunu merkezi bir koordinasyona gerek duymadan doğrudan ilgili düğüme erişerek yapar. Çünkü ağ trafiği ve iş yükü bütün katılımcı düğümler arasında paylaştırılmıştır (Hamai et. al., 2009). P2P sistemler büyük verileri depolamak, işlemek ve bunları servis edip paylaşmanın ucuz, etkin ve ölçeklenebilir yolu olma konusunda sunucu-istemci sistemlere büyük üstünlük sağlamışlardır.

2. Literatür Özeti

İlk ortaya çıktıkları günlerden günümüze kadar birçok çeşit P2P mimarisi üzerinde durulmuştur. P2P sistemleri Şekil 1’de görüldüğü üzere yapısal olmayan (unstructured) ve yapısal (structured) olmak üzere iki ana sınıfa ayırmak mümkündür:

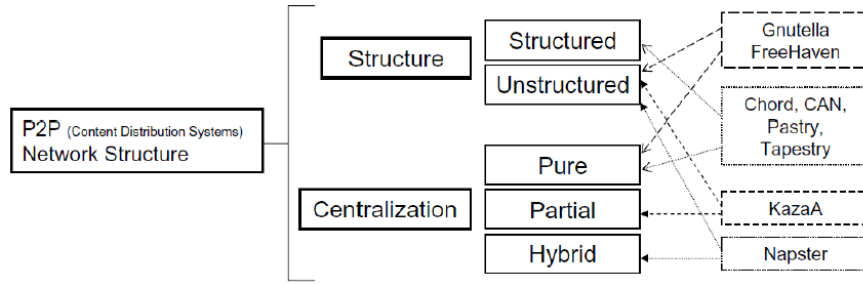
Yapısal olmayan P2P sistemler katılımcılarının hiçbir kurala bağlı olmadan keyfi olarak birbirlerine bağlandıkları bu sebeple ağa giriş çıkışın çok kolay olduğu P2P mimarileridir (Hamai et. al., 2009). Bu yöntem P2P ağlarda ağ yönetimini kolaylaştırır da arama etkinliğini düşürür (Forestiero et. al., 2010). Düğümlerin birbirlerine bağlanmaları belirli kurallar ile belirlenmediği için yönlendirme tabloları tutmazlar; dolayısı ile kaynak arama işlemi sel baskını (flooding) yöntemi kullanılarak yapılır. Bu sebeple bu tür ağlarda az popüler verilere erişim zor olmaktadır.

Bu tarz sistemler, genellikle P2P mimarisinin ilk ortaya atıldıkları dönemde sıkça rastlanılan ve Napster’ın (napster.com) öncülük ettiği sistemlerdir. Napster ilk ortaya atılan P2P uygulaması olmasının yanında türdeşlerinden farklı olarak merkezi bir sunucu barındırır. Bu sunucunun ana görevi dosyaları indeksleme işlemidir. Dosyaların hangi düğümlerde oldukları bu sunucu tarafından tutulur. Napster ile yaklaşık aynı zamanlarda geliştirilen bir başka yaklaşım Guntella (rfc-gnutella.sourceforge.net) olmuştur. Guntella, Napster gibi bir indeksleme sunucusu barındırmaz ve tamamen dağıtık bir yapıya sahiptir. Bu sebeple dosyaları ararken sel baskını yöntemini kullanır. Napster ve Guntella uygulamalarının başarılarından sonra bu alanda daha sonra KaZaA (kazaa.com), EMule (emule-project.net), Bittorrent (bittorrent.com) gibi uygulamalar görülmüştür.

Yapısal P2P sistemler de yapısal olmayanlardan farklı olarak, bütün düğümler ve kaynaklar belirli lokasyonlara atanırlar. Daha sonra bu lokasyon bilgileri kullanılarak kaynak ve düğüm arama işlemleri daha etkin bir şekilde gerçekleştirilir (Palomar et. al., 2006). Düğümleri ve kaynakları belirli lokasyonlara atama işlemi Distributed Hash Table (DHT) vasıtası ile yapılır (Forestiero et. al., 2010 ; Stoica et. al., 2001 ; Korzun and Gurtov, 2010 ; Korzun and Gurtov, 2013). Bu sebeple bu tür sistemlere DHT tabanlı P2P ağlar da denir. Bu yöntem ile bir düğüm ve/veya kaynak bilgisi spesifik bir hash fonksiyonuna tabi tutulur ve hash fonksiyonunun çıktısı düğümün ve/veya kaynağın atanacağı lokasyonu belirlemiş olur.

Düğümlerin ve kaynakların yerlerini belirlerken uniform hash fonksiyonlarının kullanılması ile ağdaki düğümler arasındaki iş dengesi en iyi şekilde ayarlanmış olacaktır (Stoica et. al., 2001 ; Korzun and Gurtov, 2013 ; Konstantinou et. al., 2009). Ayrıca düğümler bu hash fonksiyonu ile bağlantılı bazı düğüm bilgilerini tuttuklarından arama yönlendirmeleri efektif bir şekilde yapılmaktadır. Bu tür sistemler de istenilen verinin anahtar değeri kullanılarak direk konumlanma yapıldığı için az popüler veriler dahi hızlı bir şekilde bulunmaktadır.

Yapısal görevdeş ağlarda düğümler ve kaynaklar hash fonksiyonları yardımı ile deterministik olarak atandığından bu tür sistemlerin belirli yapısal bir formları vardır. Her düğüm komşularının yanı sıra bazı başka düğümlerin yerlerini de bilir. Dolayısı ile daha efektif bir yönlendirme gerçekleştirilir. Buna karşılık ağ altyapısını güncel tutmak için düğümler bazı kontrol mesajları ile komşu düğümlerin online olup olmadıklarını kontrol ederler. Kontrol mesajları ağa ek bir yük (overhead) getirir. Bu yükün fazla olmaması gerekir. Tasarlanacak sistemleri performanslarını artırmak, dezavantajlarını ortadan kaldırmak ve açıklarını kapatmanın yanında ayrıca kontrol mesajları yükü (control message overhead) yönünden de incelemek gerekir. Chord (Stoica et al., 2001), CAN (Ratnasamy et al., 2001), Pastry (Rowstron and Druschel, 2001) ve Tapestry (Zhao et al., 2004) en yaygın yapısal görevdeş ağlardır.

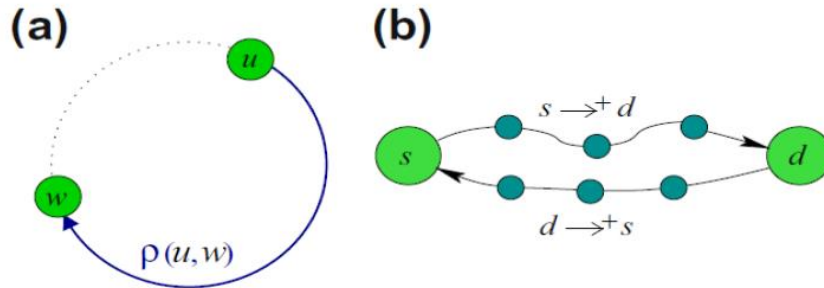


Şekil 1. P2P sistemlerin sınıflandırılması (Palomar et al., 2006).

P2P ağların merkezi kontrolden yoksun doğası gereği kimlik doğrulama (authentication) ve yetkilendirme (authorization) yapmak zordur. Bu sebepten dolayı P2P sistemler kötü niyetli düğümlerin (malicious node) saldırılarına daha açıktır. Bu açıkları kapatmak için literatürde birçok yöntem ve sistem önerilmiştir.

Chord: Chord n bitlik bir uzunluğa sahip tanımlayıcı uzay kullanan dairesel, DHT tabanlı bir P2P ağıdır (Stoica et. al., 2003). Düğümler ve kaynaklar deterministik hash fonksiyonu kullanılarak ID uzayında bu fonksiyonun çıktısına göre lokasyonlara atanırlar. Bir chord düğümü uzak ve yakın bazı komşularının bilgilerini parmak (finger) tablosunda tutar. Daha sonra herhangi bir arama esnasında bu tablodaki bilgileri kullanarak daha efektif (logN) bir kaynak araması gerçekleştirir.

CR-Chord: P2P networklerdeki karmaşıklık ve ölçeğin artması hiyerarşik yapıların kullanımı ve akıllı yönlendirmeleri zorunlu kılacaktır. DHT tabanlı ağlarda bir düğüm bir paket göndereceği zaman bu paketi aç gözlü (greddy) bir yönlendirme ile hedefe en yakın olan komşu düğümüne gönderir. Yönlendirmenin bundan sonrası ile ilgili fazla bir bilgiye sahip olmaz (Korzun et. al., 2010).



Şekil 2. Döngü oluşumu (Nechaev et al., 2011).

DHT tabanlı protokoller bir sorgunun cevabını belirli bir zamanda döndürmeyi garanti eden protokollerdir; ancak bu durum bütün düğümlerin güvenilir olduğu ağın en ideal halinde geçerlidir. P2P ağında belirli oranda kötücül düğüm olması durumunda bazı sorguların cevapları dönmeyebilir. Kötücül düğümler gelen sorguları bozabilir, yanlış yönlendirebilir veya silebilirler.

CR-Chord yapısal görevdeş ağlardan Chord'da kötü niyetli düğümlerin varlığında arama başarısını artırmak için geliştirilmiş ve arama geçmişlerinin döngü adı verilen yapılarda kaydedilmesine dayalı bir protokoldür (Nechaev et al., 2011). Döngü bir düğümün komşu düğümleri ile başlayıp bütün ağı dolaştıktan sonra tekrar sorgulayan düğümüne gelmesi ile son bulan bir yoldur (Şekil 2). Başarılı sorgulardan elde edilen bu döngü yapıları daha sonra kullanılmak üzere parmak tablosu benzeri bir tabloda tutulur. Bu tabloya döngü tablosu (cycle table) denir.

2.1 CR-Chord Döngü Oluşturma

CR-Chord algoritmasında döngüleri oluşturulmasında basit reaktif bir strateji izlenir. Eğer başarılı sorgular gerçekleştirilir ise ve bu gerçekleştirim bazı kriterlere (zaman, hop sayısı, vb.) göre etkin ise bu döngü sorguyu gönderen düğüm tarafından döngü tablosuna kaydedilir.

CR-Chord'da birincil arama (primary lookup) ve ikincil arama (secondary lookup) olmak üzere iki tip sorgu bulunmaktadır. Birincil aramalar döngülerden faydalanırlar. İkincil aramalar döngüleri oluşturmak için normal Chord alt yapısını kullanırlar. Her iki arama da kaynak aramak için kullanılabilirler. Bir

arama bir adet birincil ve md adet ikincil aramadan oluşabilir. İkincil aramalar düğümünden düğüme geçerken kesinlikle döngü tablosu kullanmazlar. Böylece söz konusu anahtara giden daha iyi bir döngü varsa ise o bulunmuş olur.

2.2 CR-Chord Arama Yönlendirme

Birincil aramalar döngüleri kullanarak hedef düğüme ulaşmaya çalışırlar. Her bir yönlendirme adımında u düğümü p paketi için bir sonraki adımı seçerken şu üç durum söz konusudur:

- Kendisine cp döngüsü üzerinden gelen p paketini aynı döngü üzerinden devam ettirir.
- Kendi tablosu Cu'da daha iyi bir döngü varsa onu kullanır.
- Kendi tablosu Cu'da bir cp yoksa normal Chord altyapısını kullanır.

3. Geliştirilen Yöntem

CR-Chord algoritmasında daha önceden de bahsedildiği gibi Chord'dun öncül (successor) listesi ve parmak tablolarına ek olarak bir adet de döngü tablosu bulunmaktadır. Döngü listesi her ne kadar diğer modüller gibi düğüm bilgileri tutuyor olsa da düğümleri bulma ve listeleme yöntemlerinin diğer modüllerinkinde farklı olması sebebi ile Şekil 3'te görüldüğü üzere ayrı bir modül olarak Chord bileşik (compound) modülüne eklenmiştir. Yani bu modül her ne kadar yaptıkları görevler benzer de olsa diğer iki listeden türememiş başlı başına ayrı bir modüldür. Bu sebeple kendi sınıf, metot ve veri yapıları yazılmıştır. Elde edilen döngüler Şekil 3'teki gibi bu modül içerisinde depolanmıştır. Omnet++'da bir modül eklenirse Omnet++ altyapısı o modülün simülasyon içerisinde nasıl bir davranış göstereceğini bilmek ister. Bu sebep ile cycleList modülüne bir sınıf yazıldı ve simülasyon içerisindeki davranışları belirlendi.



Şekil 3. Oversim Chord bileşik modülünün bileşenleri.

ChordCycleList bütün Omnet ++ modüllerinde varsayılan olarak bulunan initialize ve handleMessage metotlarının yanı sıra elde edilen döngünün uygun olup olmadığını tespit eden isIdenticalCycle, isMostNearestNeighbor, isValidRange metotları, bulunan döngüyü cycleList tablosuna ekleyen addVector metodu ve arama sırasında en uygun döngüyü bulan findClosestNode isimli metotlar yazılmıştır.

ChordCycleList sınıfı haricinde Chord altyapısı üzerinde şu değişiklikler yapılmıştır:

- Mesaj sınıfları arasına cycle, distance, messageType, vb. alanlar eklenmiştir.
- Her arama için 4 sorgu oluşturulmuştur. Bu sorgulardan bir tanesi birincil arama olup sadece döngü bilgisini kullanarak yönlendirme yaparken diğer 3 tanesi farklı yollardan gitmek ve yeni döngüler keşfetmek üzere Chord altyapısını kullanırlar.
- İkincil aramalar sorgunun cevabını bulduktan sonra döngüyü oluşturmak üzere yollarına devam ederler fakat sorgunun sonucunun başarılı olduğunu sorgulayıcı düğüme bildiren bir başarı (success) mesajı atmalıdırlar. Birincil aramalar sorgunun cevabını bulduktan sonra sorgulayıcı düğüme geri direkt cevap (response) mesajı gönderirler ve döngü oluşturmaya çalışmazlar.
- Ayrıca yüksek kötüçül değerlerinde sorguların döngü oluşturması zor olduğundan başarılı her sorguyu half-cycle tablosuna ekleyerek. Half-cycle tablosu oluşturulmuştur. Böylece tam döngünün olmadığı durumlarda en azından gidilecek yolun bir kısmına kadar güvenli bir şekilde gidilebilir.

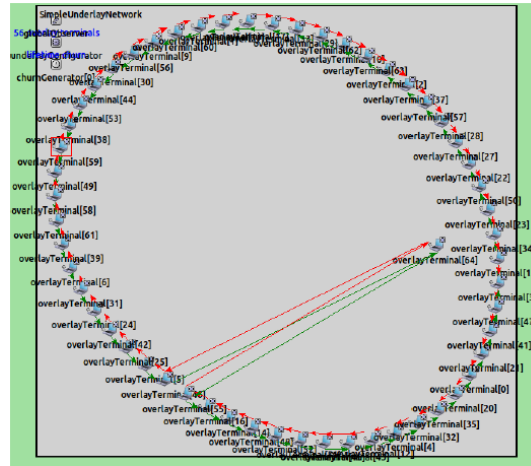
Bu değişikliklerin ilk üç tanesi literatürdeki CR-Chord algoritmasının gerçekleştirimi için yapılmışken dördüncü değişiklik makale kapsamında geliştirilen yöntemi tarif eder.

Kötücüllük oranı arttıkça düğümlerin döngü bulma ihtimalleri gittikçe azalmaktadır. Çünkü yüksek kötücüllük oranlarında bir sorgunun ağı tamamını dolaşıp sorgucu düğüme tekrar gelebilmesi çok düşük bir ihtimaldir. Bu sebeple %50 kötücüllükte birçok düğümün hiç döngüsünün olmadığı görülmektedir. Bu durum arama başarımını olumsuz etkilemektedir. Bu durumun yarattığı etkiyi en aza indirmek için düğümler sorguladıkları anahtardan sorumlu düğüme kadarki yol bilgilerini de sorgucu düğüme gönderebilirler. Sorgu anahtardan sorumlu düğüme ulaşınca anahtarın bulunduğu dair bir cevap mesajı döner. Söz konusu bu cevap mesajının içerisine o anki döngü bilgileri gömülebilir. Böylece sistemin üzerine ek bir yük getirmeden bazı bilgiler sorgucu düğüme göndermiş oluruz. Bu yol tam bir döngü olmaz; fakat yine de sorgucu düğüm güvenilir düğümlerden oluşan bazı komşularının bilgilerini elde etmiş olur.

4. Simülasyon Aracı

4.1 OMNeT++

Omnet++, C++ dili ile yazılmış nesne yönelimli ayırık olay simülasyonu (dicreate event simülatör)'dur. Omnet++'ın topoloji oluşturmak için kendine özgü Ned (network description language) dili vardır. Omnet++ programları modül (module), bileşik modül (compound module), kanal (channel) ve ağ (network) gibi bileşenlerden oluşur.



Şekil 4. OverSim üzerinde Chord'un çalışması sonucu oluşan bir kesit.

4.2 OverSim

OverSim, Omnet++ ile yazılmış yapısal (Chord, Kademia, Pastry vb.) veya yapısal olmayan (Gia) P2P ağları simüle eden açık kaynak kodlu bir simülasyon aracıdır.

Şekil 4'te OverSim'de Chord protokolünün 50 düğüm ile çalıştırılması sonucu oluşmuş bir yapı görülmektedir. Burada düğümler belirlenen yönlendirme protokolü (iterative, exahustive-iterative, semi-recursive, full-recursive, source-routing recursive) ile aramalarını yapmaktadırlar. Bu aramaları yapabilmek için ağı belirli bir düzene sokulması ve her düğümün bazı bilgileri tablolarına kaydetmesi gerekir. Dolayısı ile arama mesajlarının (FindNodeCall) yanında ağı yapısını korumak için kontrol mesajlarına (JoinCall, StabilizeCall, FixFingerCall, NotifyCall vb.) ihtiyaç vardır.

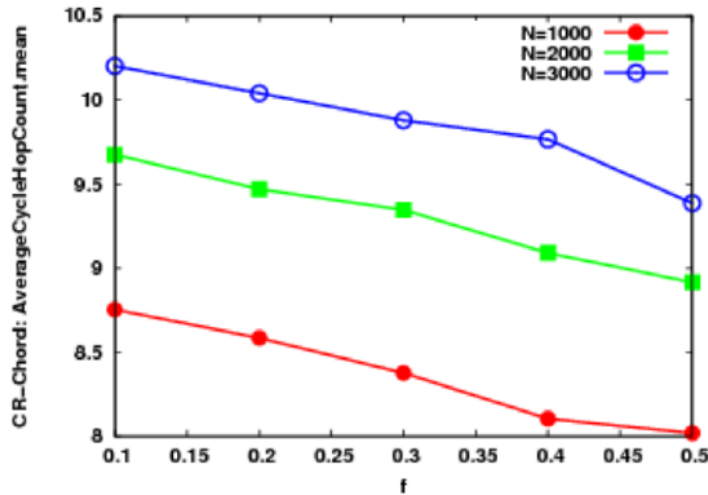
5. Deneysel Sonuçlar

Yapılan deneylerde Chord altyapısına belirli oranlarda kötücül düğümler eklenmiştir. Bu kötücül düğümler senaryo itibari ile sadece gelen arama sorgusunu düşürebilmektedirler (drop). Bu tarz kötücül düğümlerin varlığında sistemin arama başarımı ölçülmüştür. Kötücül düğümler Tablo 1'de de görüldüğü üzere 0.1, 0.2, 0.3, 0.4 ve 0.5 frekanslarında seçilmişlerdir. Eğer kötücül düğüm frekansı 0.1 ise bu sistemdeki düğümlerin 10%'nun kötücül 90%'nın da masum düğüm olduğu anlamına gelir.

Tablo 1: Gerçekleştirilen HCR-Chord ve literatürdeki CR-Chord algoritmasının karşılaştırılması (N düğüm sayısı, f kötücül düğüm frekansı).

N/f	HCR-Chord					CR-Chord (Nechaev et al., 2011)				
	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
1000	13.52	26.65	44.66	62.04	78.49	20.5	40.0	58.5	74.0	84.5
2000	16.45	32.73	55.53	76.03	86.67	21.0	44.0	64.0	79.0	90.0
3000	17.62	39.69	63.35	77.27	89.15	22.0	45.0	66.0	81.0	92.0

Tablo 1’de değişen network boyutlarına ve kötücüllük oranına göre sorguların başarısız olma durumları gösterilmiştir. Örneğin $f=0.1$ iken ve 1000 adet düğümlü bir networkte geliştirilen HCR-Chord algoritmasının sorgularının 13.52%’si başarısız olmuştur. Buna karşılık aynı simülasyon parametrelerin CR-Chord algoritmasının sorgularının 20.5%’i başarısız olmuştur. Tablo 1’den de görüleceği üzere kötücüllük oranı arttıkça sorguların başarısız olma oranları da artmıştır. Yine Tablo 1’den anlaşılacağı üzere sistemdeki düğüm sayısı arttıkça sorguların başarısız olduğu görülmektedir. Bu durumu en büyük nedeni düğüm sayısının artması ile arama sorgularının hop sayısının artmasıdır. Bu durum arama sorgularının bir kötücül düğümle karşılaşma ihtimalini artırmıştır. Bu durumu Şekil 5’ten de anlayabilmekteyiz.



Şekil 5. HCR-Chord ve döngü listesindeki döngülerin ortalama uzunluklarının karşılaştırması.

Şekil 5’te değişen network boyutuna ve kötücüllük oranına göre düğümlerde depolanan döngülerin ortalama uzunlukları gösterilmiştir. Bu sonuçlara göre kötücüllük oranının artması düğümlerde tutulan ortalama döngü uzunluklarını kısaltmıştır. Çünkü daha uzun döngüleri tamamlamak zorlaşmıştır. İnternet üzerindeki verilerin miktarının üstel olarak artması ile birlikte kaynaklara ulaşım zorlaşmıştır. Bu sebeple kaynak paylaşımı noktasında dağıtık sistemler özellikle de P2P sistemlerin kullanımı giderek artmıştır. Ancak bununla birlikte kaynakların daha efektif paylaşılabilmesi için akıllı yönlendirmelerin yapılması şarttır. Bu yayında HCR-Chord ile CR-Chord algoritmasına artı olarak yeni bir akıllı yönlendirme yöntemi geliştirilmiş ve arama başarımı olarak iyi sonuçlar elde edilmiştir.

Kaynakça

Agushinta, D., Suhendra, A., Febrinata, A. and Yuliyanti, A., 2009, Secure Mobile Agent System in Peer-to-Peer Networks: A Review of Security Mechanisms Based on Several Security Issues, Software Engineering, 2009. WCSE '09. WRI World Congress 1:212-216 pp.

- Ayyasamy, S. and Sivanandam, S.N., 2010, Trust Based Content Distribution for Peer-To-Peer Overlay Networks, *International Journal of Network Security & Its Applications (IJNSA)*, 2(2).
- Bittorrent, <http://www.bittorrent.com/>, <http://www.bittorrent.org>
- Douceur, J.R., 2002, The Sybil Attack, *Proceedings of 1st International Workshop on Peer-to-Peer Systems (IPTPS)*.
- Emule, <http://www.emule-project.net>
- Forestiero, A., Leonardi, E., Mastroianni, C. and Meo, M., 2010, Self-Chord: A Bio-Inspired P2P Framework for Self-Organizing Distributed Systems, *IEEE/ACM Transactions on Networking*, 18(5).
- Gnutella Developer Forum. "Gnutella protocol specification v0.4," rfc-gnutella.sourceforge.net
- Hamai, T., Fujii, M. and Watanabe, Y., 2009, ITU-T Recommendations on Peer-to-Peer (P2P) Network Security, Autonomous Decentralized Systems, ISADS '09. *International Symposium*, 1-6 pp.
- Kazaa, <http://www.kazaa.com>
- Konstantinou, I., Tsoumakos, D. and Koziris, N., 2009, Measuring the cost of online load-balancing in distributed rangequeriable systems. In: *IEEE P2P '09: proc. 9th int'l conf. on peer-to-peer computing*. IEEE, 135–138 pp.
- Korzun, D. and Gurtov, A., 2010, Survey on hierarchical routing schemes in "flat" distributed hash tables, *Peer-to-Peer Networking and Applications* 4(4):346-375 pp.
- Korzun, D. and Gurtov, A., 2013, Hierarchical architectures in structured peer-to-peer overlay networks, *Peer-to-Peer Networking and Applications*, 7(4):359-395 pp.
- Moro, G., Ouksel, A.M. and Sartori, C., 2003, Agents and Peer-to-Peer Computing A Promising Combination of Paradigms AP2PC'02 *Proceedings of the 1st international conference on Agents and peer-to-peer computing*, 1-14 pp.
- Napster, LLC. "Napster", <http://www.napster.com>
- Nechaev, B., Korzun, D. and Gurtov, A., 2011, CR-Chord: Improving lookup availability in the presence of malicious DHT nodes. *Computer Networks*, 55(13), 2914-2928.
- Palomar, E., Estevez-Tapiador, J.M., Hernandez-Castro, J.C. and Ribagorda, A., 2006, Security in P2P Networks: Survey and Research Directions, *Emerging Directions in Embedded and Ubiquitous Computing Lecture Notes in Computer Science*, 4097:183-192 pp.
- Ratnasamy, S., Francis, P., Handley, M., Karp, R. and Shenker, S., 2001, A Scalable Content Addressable Network, *SIGCOMM '01 Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, 161-172 pp.
- Rowstron, A. and Druschel, P., 2001, Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems ,*Middleware '01 Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, 329-350 pp.
- Seedorf, J., 2006, Security Challenges for Peer-to-Peer SIP, *IEEE Network*, 20(5) : 38-45 pp.

- Shen, X., Liu, L., Zha, Z., Gu, P., Jiang, Z., Chen, J. and Panneerselvam, 2014, Achieving dynamic load balancing through mobile agents in small world P2P networks, *Computer Networks*, 75:134-138 pp.
- Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R., Kaashoek, M.F., Dabek, F. and Balakrishnan, H., 2001, Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications, *SIGCOMM '01 Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, 149-160 pp.
- Stoica, I., Morris, R., Liben-Nowell, D., Karger, D. R., Kaashoek, M. F., Dabek, F. and Balakrishnan, H., 2003, Chord: a scalable peer-to-peer lookup protocol for internet applications. *Networking, IEEE/ACM Transactions on*, 11(1), 17-32.
- Zhao, B.Y., Huang, L., Stribling, J., Rhea, S.C., Joseph, A.D. and Kubiatoiwich, J.D., 2004, Tapestry: A Resilient Global-Scale Overlay for Service Deployment, *IEEE Journal on Selected Areas in Communications*, 22:41-53 pp.