

KABLOSUZ AĞLARDA SEZGİSEL BİR YÖNLENDİRME PROTOKOLÜ VE TIKANIKLIK DENETİMİ

Mehmet ŞİMŞEK ve M. Ali AKCAYOL

Bilgisayar Mühendisliği Bölümü, Mühendislik Mimarlık Fakültesi, Gazi Üniversitesi, Maltepe, Ankara
mehmet.simsek@gazi.edu.tr, akcayol@gazi.edu.tr

(Geliş/Received: 07.02.2007; Kabul/Accepted: 04.08.2007)

ÖZET

Bu çalışmada, kablosuz ağlar için bir sezgisel yönlendirme protokolü geliştirilmiştir. Geliştirilen protokol, tıkanıklık oluşmadan önce yük dağılımını yeniden düzenleyerek tıkanıklığı önlemektedir. Geliştirilen protokol ns-2 (network simulator 2) benzetim aracı kullanılarak test edilmiş ve test sonuçları kablosuz ağlarda yaygın olarak kullanılan Ad Hoc On Demand Distance Vector (AODV) Protokolü ile karşılaştırılmıştır. Elde edilen deneysel sonuçlar geliştirilen protokolün kablosuz ağlarda tıkanıklık denetiminde başarılı olduğunu göstermiştir.

Anahtar Kelimeler: Tıkanıklık, yönlendirme, protokol, kablosuz ağlar.

A HEURISTIC ROUTING PROTOCOL AND CONGESTION CONTROL AT WIRELESS NETWORKS

ABSTRACT

In this study, a heuristic routing protocol has been developed for wireless networks. Developed protocol prevents congestion by compensating load balance before congestion occurs. The protocol has been tested using ns-2 (network simulator 2) simulation tool and experimental results have been compared with Ad Hoc On Demand Distance Vector (AODV) protocol used commonly in wireless networks. Obtained results showed that the developed protocol is succesful for congestion prevention in wireless networks.

Keywords: Congestion, routing, protocol, wireless networks.

1. GİRİŞ (INTRODUCTION)

Tıkanıklık, kablolu ağlarda olduğu gibi kablosuz ağlarda da ağın performansını etkileyen en önemli konulardan biridir. Kablolu ağlarda birçok tıkanıklık denetleme ve önleme yöntemi kullanılmaktadır [1,2]. Bu yöntemler genel olarak yönlendirme katmanında ve ulaştırma katmanında çalışmaktadır. Kablosuz ağlar fiziksel yapılarından dolayı kablolu ağlar gibi yüksek bant genişliğine sahip değildir ve bu ağlarda hareketli yapılarından dolayı sık sık bağlantı kopuklukları yaşanabilmektedir. Bu farklılıklardan dolayı kablolu ağlarda kullanılan yöntemler kablosuz ağlarda olduğu gibi kullanılamamaktadır [3]. Kablosuz ağlar için kullanılacak olan yönlendirme ve tıkanıklık giderme protokolü bu tür ağların özelliklerine uygun olmalıdır.

Kablosuz ağlar için gerçekleştirilmiş birçok tıkanıklık denetleme ve giderme yöntemi bulunmaktadır [4].

1988 Kasım ayında Jacobson ve Karels yaptıkları çalışmada, Berkeley Software Distribution (BSD) Transmission Control Protocol (TCP) 4.0'da yer alan yedi algoritmadan beşini tanımlamışlardır [2]. Bu algoritmalar, round-trip time variance estimation, exponential retransmit timer backoff, slow-start, more aggressive receiver ACK policy ve dynamic window sizing on congestion'dır. 1994 yılında Perkins ve Bhagwat, yaptıkları çalışmada hareketli kablosuz ağlarda kullanılan ilk yönlendirme protokolü olan Highly Dynamic Destination - Sequenced Distance Vector Routing'i (DSDV) tanımlamışlardır [5]. DSDV protokolünde her bir düğüm, ağdaki diğer düğümler için bir yol bulunduran tablo tutar. Her satırda hedef düğüm, sonraki düğüm, uzaklık ve sıra numarası kayıtlıdır. Sıra numarası o yolun güncelliğini gösterir. Johnson ve Maltz 1996 yılında, "Dynamic Source Routing Protocol in Ad-Hoc Wireless Networks" isimli kablosuz ağlarda kullanılan yönlendirmeli protokolünü

geliştirmişlerdir [6]. Bu protokol tablo kullanan protokollerde olan tablo güncellemelerinin ağ üzerinde taşınması ile meydana gelen yükün azaltılmasını sağlamaktadır. Bu protokolün en büyük farkı periyodik “merhaba” mesajlarının olmamasıdır.

Stevens 1997 yılında yaptığı çalışmada, tıkanıklıktan kaçınmak için TCP’de neler yapılabileceğini, tıkanıklık meydana gelmesi durumunda hızlı bir biçimde nasıl giderilebileceğini açıklamıştır [2]. 1999 yılında Allman, Paxson ve Stevens’in TCP’de tıkanıklık denetimi ve giderme ile ilgili yapmış oldukları çalışmaları Request For Comments (RFC) standartları arasına girmiştir [7, 8]. Kim, Toh ve Choi, 2001 yılında yaptıkları çalışmalarıyla, kablosuz ağlar için performansı artırılmış bir TCP sürümü olan TCP With Buffering Capability and Sequence Information’ı (TCP-BuS) geliştirmişlerdir [9]. TCP-BuS, dayanışma tabanlı yönlendirme protokolü üzerine kurulmuştur. Aktarımın olduğu her ara düğüm, aktarılan paketleri bir tamponda tutar. 2001 yılında Chandran, Raghunathan, Venkatesan ve Prakash yaptıkları çalışmalarıyla, kablosuz ağlar için değiştirilmiş olan bir TCP sürümü olan Feedback-Based TCP protokolünü (TCP-F) geliştirdiler [10]. Haas, Pearlman ve Samar, The Zone Routing Protocol (ZRP) for Ad Hoc Networks isimli yeni bir kablosuz ağ yönlendirme protokolünü 2002 yılında geliştirdiler. Bu protokol, ağda meydana gelen sel paketlerinin yayılma alanını önemli ölçüde azaltarak kaynakların verimli kullanılmasını sağlamaktadır [11].

Bu çalışmada, kablosuz ağlar için sezgisel olarak tıkanıklık denetimi gerçekleştiren bir yönlendirme protokolü geliştirilmiştir.

2. YÖNLENDİRME PROTOKOLLERİ VE TIKANIKLIK DENETİMİ (ROUTING PROTOCOLS AND CONGESTION CONTROL)

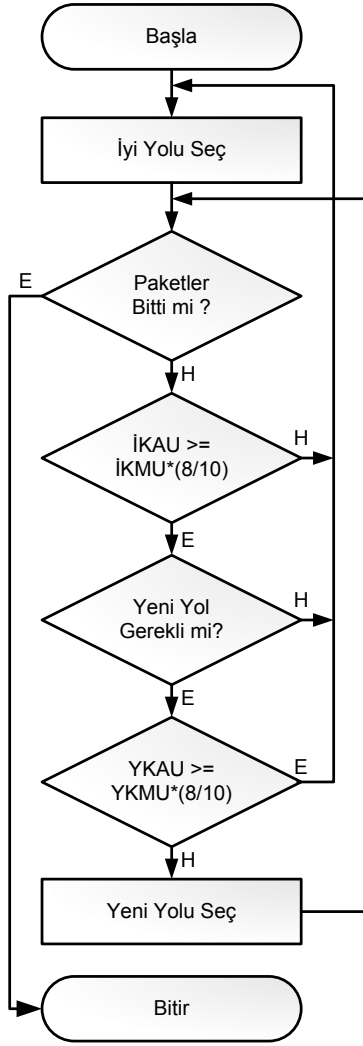
Bilgisayar ağlarında yönlendirme ve tıkanıklık denetimi ağın performansını etkileyen en önemli unsurlardır. Bir ağda fazla yükten dolayı paketlerin bekleme süreleri artarsa, paket kayıpları yaşanır ve ağın etkinliği azalır bu ağ tıkanmış demektir. Ağın etkinliğinin azalması, kaybolan paketlerin yeniden gönderilmesini gerektirmekte ve ağa fazladan yük getirmektedir [12,13]. Temel olarak iki çeşit tıkanıklık önleme yöntemi vardır. Bunlar, rezervasyon tabanlı ve isteğe bağlı metotlarıdır [14]. Bir kullanıcının ihtiyacı olan hizmet kalitesi için önceden ağ üzerinde kaynak ayrılır. Kullanıcı istediği zaman bu kaynakları kullanarak ağ üzerinden bilgi transfer eder. Bu yöntem rezervasyon tabanlı olarak adlandırılabilir. Alternatif olarak kullanıcı gerek duyduğu anda ağdaki kaynakları rezerve etmek isteyebilir. Fakat ağdaki mevcut yük dolayısı ile istediği kalitede hizmet alamayabilir. İkinci metot rezervasyonsuz ağlarda kullanılır. Bu durumda kullanıcılar kendilerini değişen ağ durumuna adapte etmek durumundadırlar. İsteğe bağlı planda ise ağ

kullanan kişiler sürekli olarak ağı gözlemlemeli ve tıkanıklığı önlemek için ağın durumunu değiştirmelidirler. Her iki yöntemin de avantajları ve dezavantajları vardır. Birinci yöntemde hizmet kalitesi garanti edilir fakat kullanıcı sayısı sınırlandırılmak zorundadır. İkinci yöntemde kullanıcı sayısı sınırlı değildir fakat her kullanıcı aldığı hizmetten ödün vermek durumunda kalabilir [14,15].

AODV hareketli kablosuz ağlarda yaygın olarak kullanılan ve bu çalışmada da karşılaştırma amacı ile kullanılmış yönlendirme protokolüdür. AODV yol bulmak için isteğe bağlı olarak hareket eder. Bir düğüm bir başka düğümle iletişime geçmek istediği zaman yol kurma isteği yayınlar. İsteğe karşılık gelen ilk cevabın geldiği yol kullanılarak haberleşmeye başlanır. Elde edilen yolların güncelliklerini belirlemek için her yola bir sıra numarası atanır. Sıra numarası en büyük olan yol en güncel yol anlamına gelmektedir. AODV’de paketlerin takip edeceği yol kaynaktan belirlenmez. Bunun yerine kaynak düğüm ve her ara düğümde yalnızca bir sonraki düğümün adresi tutulur [16]. AODV protokolünde herhangi bir yol, istenen hedef için kullanılabilir olmadığında kaynak düğüm ağa bir yol istek mesajı yayınlar [17]. AODV ile ilgili, literatürde yapılmış bir çok çalışma ve detaylı bilgi bulunabilir [16-19].

3. GELİŞTİRİLEN YÖNLENDİRME PROTOKOLÜ VE BENZETİM SENARYOLARI (DEVELOPED ROUTING PROTOCOL AND SIMULATION SCENARIOS)

Geliştirilen protokol aktarım yolu üzerindeki her düğümün arayüzünde oluşan kuyruk uzunluğunun tıkanıklık oluşturup oluşturmayacağını değerlendirilmesi, hedef düğüme giden ve kuyruk uzunluğu daha az olan bir başka düğümün belirlenmesi ve yük dağılımı yapılması olarak 3 kısımdan oluşmaktadır. Bu işlemler, yalnızca kaynaktan veya hedefte yapılmamakta, aktarım yolu üzerinde bulunan tüm düğümlerde yapılmaktadır. Kuyruk uzunluğunun sürekli artması ve kuyruk sınır değerine yaklaşması tıkanık oluşacağını göstermektedir. Hareketli kablosuz ağlarda düğümlerin hareketleri, veri alma-gönderme talepleri, düğümler arasındaki bağlantı durumlarının tahmin edilmesi son derece zor olduğu için tıkanıklığın nerede meydana geleceğini tahmin etmek mümkün olmamaktadır. Bu yüzden, kaynaktan veya hedefte yapılacak tıkanıklık önleme ve giderme yöntemleri her zaman en iyi sonucu sağlayamamaktadır. Tıkanıklığı önlemekte kullanılan çok yönlü yük dağılımı yöntemi, kaynak ve hedef düğüm için fazladan hesap yükü getirmekte, düğümlerin hareketli oluşu ve ağ yapısının sık sık değişiyor olması ilgili düğümlerin yapacakları hesapları daha da zorlaştırmaktadır. Geliştirilen yöntemde her düğüm kendi arayüzündeki kuyruk uzunluğuna göre tıkanıklık olup olmayacağına karar verip buna göre davranmaktadır. Kuyruğun anlık uzunluğu (KAU), kuyruk maksimum uzunluğu’nun (KMU) %80’ine



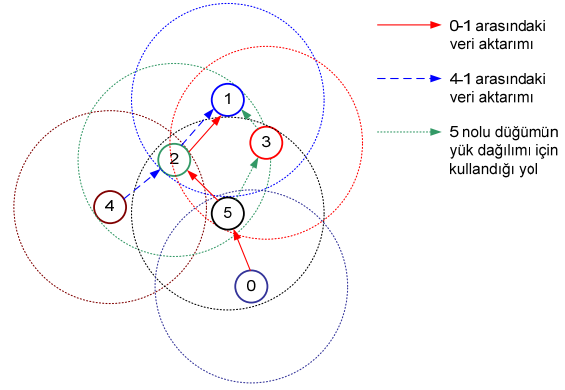
Şekil 1. Geliştirilen protokolün akış şeması
(Flow chart of the developed protocol)

ulaştığında tıkanıklık meydana geleceği kabul edilerek hedefe giden başka bir yol oluşturulur veya var olan yollar arasından başka bir tanesi seçilir.

Yol seçme ölçütü, ilgili yola bakan arayüzdeki kuyruk uzunluğudur. Şekil 1’de protokolün nasıl işlediği gösterilmektedir.

Düğüm, paketleri mevcut olan bağlantıdan yollamaya başlar. İletişimin herhangi bir anında mevcut bağlantının kullandığı arayüzün kuyruk uzunluğu, maksimum kuyruk uzunluğunun %80’ine ulaşmışsa yeni bir yol aranır. Yeni bir yol bulunmuşsa ve bu yolu kullanan arayüzün uzunluğu %80’den azsa bu yol kullanılmaya başlanır. Her paket gönderiminden sonra ilk ve en kısa olan yol kontrol edilerek kuyruk uzunluğunun azalıp azalmadığına bakılır. Kuyruk uzunluğu eşik değerden aşağı ise bu yol kullanılır. Yeni bir yol bulunamazsa iletişim ilk yoldan devam ettirilir.

Kurulan yeni yolun, paketler bitene kadar kullanılması, maliyetler açısından faydalı olarak gözükebilir. Çünkü her paket gönderiminden sonra ilk



Şekil 2. Tıkanıklık önleme işlemi (Congestion prevention) yolun kontrol edilmesi ve buna göre karar verilmesinin bir maliyeti vardır. Fakat ilk kurulan yol en kısa yol olduğu için bu yolun tamamen devre dışı bırakılarak yeni yol üzerinden aktarımın devam ettirilmesi uzun süren aktarımlarda fazladan maliyet demektir.

Şekil 2’de örnek bir topoloji üzerinde, önerilen tıkanıklık önleme yönteminin nasıl çalıştığı görülmektedir. 0 numaralı düğüm ile 1 numaralı düğüm veri aktarımı yaparken, 4 numaralı düğüm 1 numaralı düğüme veri göndermeye başlamaktadır. 5 numaralı düğümün 2 numaralı düğüme bakan arayüzündeki kuyruk uzunluğu, eşik değeri aştığı için 5 numaralı düğüm hedefe giden ikinci bir yol belirlemekte ve buradan aktarıma devam etmektedir.

Geliştirilen tıkanıklık önleme yöntemi test edilerek performansı AODV protokolü ile karşılaştırılmıştır. Esnek yapısından ve literatürdeki çoğu çalışmada kullanılmış olmasından dolayı benzetim aracı olarak ns-2 seçilmiştir. Ayrıca, AODV protokolünün ns-2 içerisinde tanımlanmış olması test süresinin kılmasını sağlamıştır.

Geliştirilen protokolü test etmek için altı adet senaryo oluşturulmuştur. Bu senaryolarda birbirleriyle eşdeğer bilgisayarların kablosuz ve altyapısız (ad hoc) haberleştikleri varsayılmıştır. İlk senaryo ile gerçekleştirilen benzetimde 6 tane eş değer düğüm kullanılmış ve bu düğümlerden dört tanesi haberleştirilmiştir.

İlk senaryodaki düğümleri oluşturan OTCL kodu aşağıdaki gibidir:

```

set val(chan) Channel/WirelessChannel
set val(prop) Propagation/TwoRayGround
set val(netif) Phy/WirelessPhy
set val(mac) Mac/802_11
set val(ifq) Queue/DropTail
set val(ll) LL
set val(ant) Antenna/OmniAntenna
set val(ifqlen) 15
set val(nn) 6
set val(rp) SezYonPro
  
```

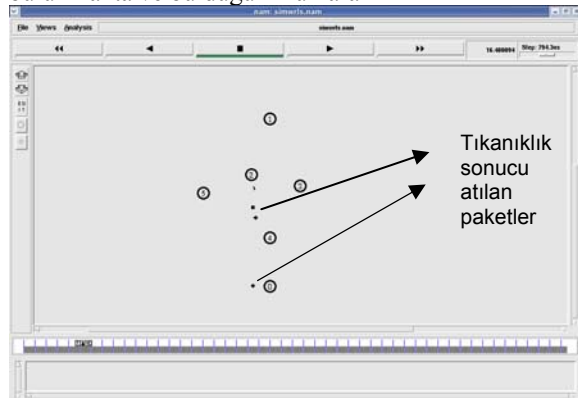
set val(x)	1000
set val(y)	1000
set val(stop)	150

Yukarıdaki parametrelerden ifqlen parametresi, tıkanıklık belirleme sırasında kullanılan arayüzlerdeki kuyruk maksimum uzunluğudur. Herhangi bir anda bu değer 12'den büyük olduğunda arayüzün bulunduğu düğüm, hedefe giden başka bir yol bularak buradan iletişime geçer. Bu değer eşik değerden aşağı düştüğü zaman da tekrar önceki yol kullanılmaya başlanır. Yukarıdaki işlemi yapan C++ kodu aşağıda verilmiştir.

```
if (ifqueue->length()>12
rt = rtable.rt_lookup_nc(ih->daddr(),rt->rt_nexthop);
if(rt == 0) rt = rtable.rt_add(ih->daddr());
```

Benzetimde ulaştırma katmanında TCP kullanılmıştır. TCP'de tıkanıklık önleme ve giderme için kullanılan yöntemler olmasından dolayı bu protokol tercih edilmiştir. TCP kullanılmasının bir diğer nedeni de karşılaştırma kriterlerinden biri olan tıkanıklık penceresi boyutunun yalnızca TCP bağlantılarında yer almasıdır. Veri trafiğini yaratmak için uygulama katmanında FTP (File Transfer Protocol) kullanılmıştır. İlk senaryoda benzetimin 10. saniyesinde 0 numaralı düğüm ile 1 numaralı düğüm arasında kurulan TCP bağlantısı üzerinden FTP trafiği başlamaktadır. İlk TCP bağlantısında veri aktarımı için belirlenen yol "0-4-2-1" şeklindedir. Benzetimin 15. saniyesinde 5 numaralı düğüm ile 1 numaralı düğüm arasında kurulan TCP bağlantısı üzerinden FTP trafiği başlamaktadır. İkinci TCP bağlantısında veri aktarımı için belirlenen yol "5-2-1" şeklindedir. AODV protokolü ilk senaryo ile denenerek sonuçlar elde edilmiştir. Benzetimde oluşturulan ikinci trafiğin başlaması ile birlikte 2 numaralı düğümün kuyruk uzunluğu artmış ve bir süre sonra tıkanıklık meydana gelmiştir. Tıkanıklık sonucu meydana gelen paket atılma durumu Şekil 3'te görülmektedir.

Daha sonra, geliştirilen protokol kullanılarak benzetim aynı şartlarda tekrar gerçekleştirilmiştir. Benzetimin 15. saniyesinde 5 numaralı düğüm yol kurma talebinde bulunmakta ve bulunduğu 2 numaralı



Şekil 3. Tıkanıklık sonucu paketlerin atılması (Packet drops due to congestion)

düğüm üzerinden kurduğu yolu kullanarak veri aktarımına başlamaktadır. Bu andan sonra 4 numaralı düğümün 2 numaralı düğüme bakan arayüzündeki kuyruk uzunluğu hızla artmakta ve belirlenen eşik değerini geçmektedir. Geliştirilen protokolde 4 numaralı düğüm, hedefe giden bir başka yol aramakta ve 3 numaralı düğümü bir sonraki düğüm olarak seçip paketleri hedefe göndermeye başlamaktadır. 4 numaralı düğüm ile 2 numaralı düğüm arasındaki kuyruk uzunluğunun değeri eşik değerinin altına düştüğü anda trafik tekrar bu arayüzden aktarılmaya devam etmektedir.

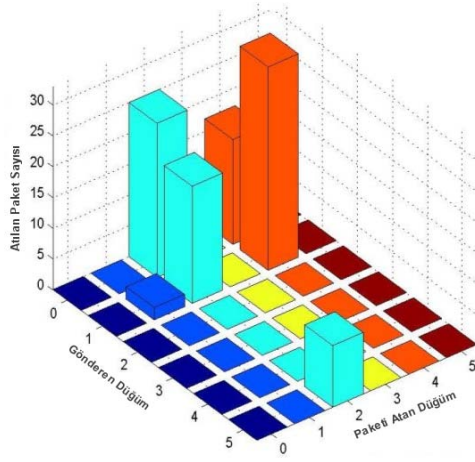
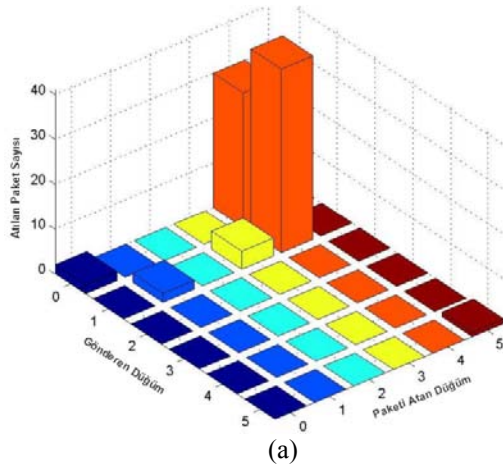
Yukarıda anlatılan senaryonun dışında 5 senaryoyla daha test işlemi gerçekleştirilmiştir. Bu senaryolarda sırası ile; 10, 20, 30, 40 ve 50 düğüm oluşturulmuştur. Düğüm sayıları ile orantılı olarak sırası ile 4, 8, 12, 16 ve 20 tane TCP bağlantısı oluşturulmuş ve bu bağlantılar üzerinde FTP ile iletişim sağlanmıştır. Testlerde çok sayıda bağlantı oluşturularak tıkanıklık için son derece müsait bir ortam hazırlanmıştır. Her bir senaryo 10 defa çalıştırılmış ve paket atılmalarının ortalama değerleri alınmıştır.

4. DENEYSEL SONUÇLAR (EXPERIMENTAL RESULTS)

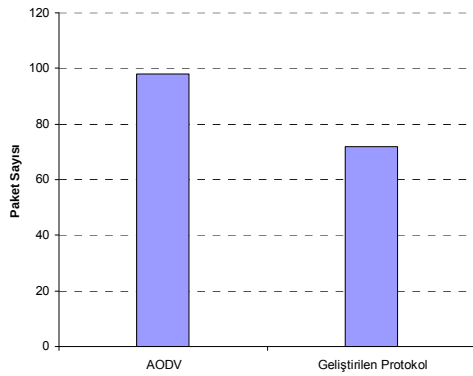
Geliştirilen protokol ile AODV protokolü altı senaryo için karşılaştırılarak sonuçlar elde edilmiştir. Geliştirilen protokolün tıkanıklık durumunda nasıl tepki verdiğini görmek için tüm ağda meydana gelen paket kaybı sayıları karşılaştırılmıştır. Paket kayıplarının büyük nedeni tıkanıklıktır. AODV protokolü ve geliştirilen protokol için ilk senaryo ile gerçekleştirilen benzetim sonucunda elde edilen atılan paketler grafiği Şekil 4'te görülmektedir.

Özellikle 0 ve 1 numaralı düğümlerin oluşturduğu paketlerin 2 ve 4 numaralı düğümlerde atıldığı görülmektedir. Kaynağı 0 düğümü olan paketler veri paketleri, kaynağı 1 olan paketler de Acknowledgement (ACK) paketleridir. 2 numaralı düğümde atılan paketlerin büyük kısmının atılma nedeni tıkanıklıktır. Atılan paketlerin çok küçük kısmı ise aktarım ortamını ele geçirme için yaşanan çekişme neticesinde atılmıştır. 4 numaralı düğümde atılan paketlerin büyük kısmı ACK paketidir. Tıkanıklıktan dolayı geciken ACK paketleri nedeniyle kaynak düğüm yeniden iletim yordamını çalıştırır. Yeniden gönderilen veri paketleri 4 numaralı düğüme geldiği zaman, 4 numaralı düğümde beklemekte olan ACK paketleri ve henüz gönderilmemiş olan veri paketleri atılır.

Geliştirilen protokolde önceden yük dağılımı yapılarak 2 numaralı düğümde paket atılmasının önlenmesi görülmüştür. 3 numaralı düğümde az sayıda ACK paket atılması olmuştur. 4 numaralı düğümde meydana gelen paket atılmaları da AODV protokolüne göre daha düşüktür. İki protokolün ilk senaryoya göre oluşturulmuş toplam atılan paket sayılarına ilişkin grafik Şekil 5'te verilmiştir.



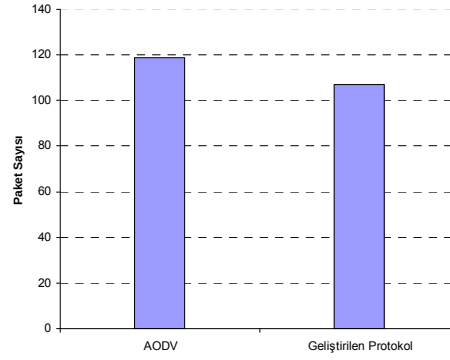
Şekil 4. (a) AODV ve (b) geliştirilen protokol için ilk senaryoda atılan paket sayıları (Number of dropped packet for the first scenario in (a) AODV and (b) developed protocol)



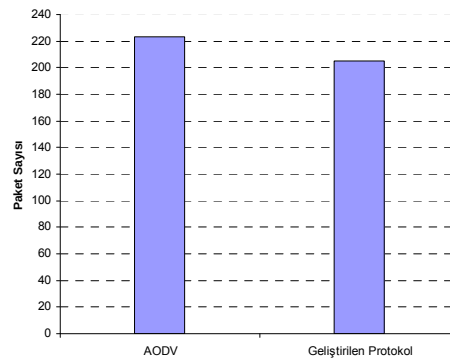
Şekil 5. İlk senaryoda toplam atılan paket sayıları (Total number of dropped packet for the first scenario)

AODV protokolünde toplam olarak 98 paket atılmış, geliştirilen protokolde ise 72 paket atılmıştır. Geliştirilen protokoldeki paket atılma oranının AODV protokolündeki paket atılma oranına göre yaklaşık olarak %18 daha az olduğu görülmektedir.

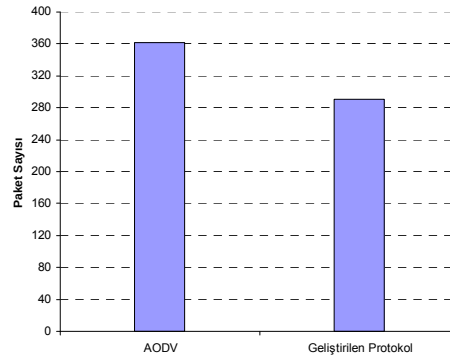
İki protokolün 2, 3, 4, 5 ve 6 numaralı senaryolara göre oluşturulmuş toplam atılan paket sayılarına ilişkin grafikler sırası ile Şekil 6, 7, 8, 9 ve 10'da verilmiştir.



Şekil 6. İkinci senaryoya göre toplam atılan paket sayıları (Total number of dropped packet for the second scenario)



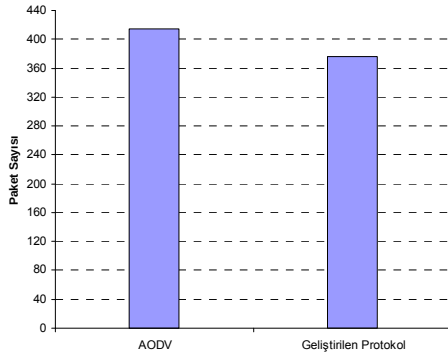
Şekil 7. Üçüncü senaryoya göre toplam atılan paket sayıları (Total number of dropped packet for the third scenario)



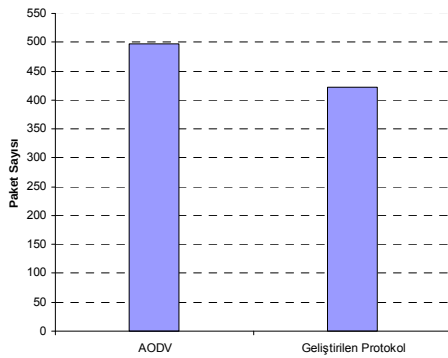
Şekil 8. Dördüncü senaryoya göre toplam atılan paket sayıları (Total number of dropped packet for the fourth scenario)

6 senaryo için elde edilmiş olan atılan paket sayıları grafikleri, her bir senaryonun 10 defa çalıştırılması neticesinden elde edilen değerlerin ortalamaları alınarak hazırlanmıştır.

Tablo 1'de AODV ile geliştirilen protokolün tüm senaryolarda atılan paket sayıları karşılaştırmalı olarak görülmektedir. Buradan da anlaşılacağı gibi, geliştirilen protokol AODV protokolüne göre bağlantı sayısından ve düğüm sayısından bağımsız olarak daha iyi sonuç vermektedir.



Şekil 9. Beşinci senaryoya göre toplam atılan paket sayıları (Total number of dropped packet for the fifth scenario)



Şekil 10. Altıncı senaryoya göre toplam atılan paket sayıları (Total number of dropped packet for the sixth scenario)

Yukarıdaki tablonun neticesinde geliştirilen protokolün paket atılmalarını 1. senaryoda %26, 2. senaryoda %10, 3. senaryoda %8, 4. senaryoda %19, 5. senaryoda %9 ve 6. senaryoda %15 oranlarında azalttığı görülmektedir.

5. SONUÇLAR (CONCLUSIONS)

Bu çalışmada kablosuz ağlarda tıkanıklık denetimi için sezgisel bir protokol geliştirilmiştir. Geliştirilen protokol ns-2 benzetim aracı kullanılarak test edilmiş ve kablosuz ağlarda yaygın olarak kullanılan AODV protokolü ile karşılaştırılmıştır. Geliştirilen protokol yükü birden fazla yola dağıtırken öncelikli olarak ilk kurulan yolu kullanmaktadır. Kurulan ilk yolun tıkanmaya başlaması ile birlikte ikinci yolun uygun olması durumunda trafik buraya kaydırılmakta ve ilk belirlenen yoldaki kuyruk uzunluğu eşik değerin altına ininceye kadar ikinci yol kullanılmaktadır. Bunun yerine aktarımın sonuna kadar iki yol birlikte kullanılabilir. Ancak ilk belirlenen yol en kısa yol olduğundan ikinci yolun kullanılması iletişim maliyetini artıracaktır. Maliyet etkin en iyi performansı elde etmek için kuyruk eşik değerinin kurulacak yolun maliyetine göre iyi belirlenmesi ve uyarlanabilir olması gerekmektedir. Deneysel çalışmaların sonucunda, geliştirilen protokolün en kısa yoldaki iletişim tamamen tıkanmadan, uzunluğu daha fazla olan başka bir yol kurup iletişimin bir kısmını bu yola aktarmasının toplam atılan paket sayısını önemli ölçüde azalttığı görülmüştür.

Tablo 1. AODV ile geliştirilen protokolün tüm senaryolarda atılan paket sayıları (Total number of dropped packet for the all scenarios with AODV and developed protocol)

Senaryo	AODV	Geliştirilen Protokol
1	98	72
2	119	107
3	223	205
4	362	291
5	414	376
6	498	422

Gerçekleştirilen benzetimler sonucunda elde edilen değerlerden, geliştirilen protokolün kablosuz ağlarda yaygın olarak kullanılan bir protokol olan AODV'den tıkanıklık durumunda daha iyi sonuçlar verdiği gözlemlenmiştir.

KAYNAKLAR (REFERENCES)

1. Jacobson V., "Congestion Avoidance and Control", **SIGCOMM '88**, 314-320, Stanford, ABD, 1988.
2. Stevens W.R., "TCP Slow-Start, Congestion Avoidance, Fast Retransmission, and Fast Recovery Algorithms", **IETF RFC 2001**, Virginia, 1997.
3. Keshav S., "Congestion Control in Computer Networks", PhD Thesis, Computer Science, **University of California**, Berkeley, Department of Electrical-Electronics and Computer Science, 1991.
4. Welzl M., "Network Congestion Control, Managing Internet Traffic", **Wiley**, New Jersey, ABD, 2005.
5. Perkins C. and Bhagwat P., "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", **ACM SIGCOMM'94**, 234-244, London, England, 1994.
6. Johnson D. and Maltz D., "Dynamic Source Routing in Ad Hoc Wireless Networks", **Mobile Computing**, 353(5), 153-181, 1996.
7. Allman M., Paxson V. and Stevens W., "TCP Congestion Control", **IETF RFC 2581**, Virginia, 1999.
8. Floyd S., "Congestion Control Principle", **IETF RFC 2914**, Virginia, 2000.
9. Kim D., Toh C., and Choi Y., "TCP-BuS: Improving TCP Performance in Wireless Ad Hoc Networks", **Journal of Communications and Networks**, 3(2), 175-186, 2001.
10. Chandran K., Raghunathan S., Venkatesan S., and Prakash R., "A Feedback-Based Scheme for Improving TCP Performans in Ad Hoc Wireless Networks", **IEEE Personal Communications Magazine**, 8(1), 2001.
11. Haas Z.J., "A New Routing Protocol for the Reconfigurable Wireless Networks", **IEEE International Conference on Universal Personal Communication (ICUPC)**, 562-566, San Diego, ABD, 1997.
12. Stallings W., "Wireless Communications and Networks", **Prentice Hall**, New Jersey, ABD, 2004.
13. Kurose J.F., "Computer Networking: A Top-Down Approach Featuring the Internet", **Addison Wesley**, Boston, ABD, 2004.

14. Basagni S., Conti M., Giordano S., Stojmenovi I., “Mobile Ad Hoc Networking”, **Wiley-IEEE Press**, New Jersey, ABD, 2004.
15. Holland G. and Vaidya N.H., “Analysis of TCP Performance Over Mobile Ad Hoc Networks”, **ACM/IEEE MOBICOM**, 219-230, Seattle, ABD, 1999.
16. Aggelou G., “Mobile Ad Hoc Networks: From Wireless LANs to 4G Networks”, **McGraw Hill**, New York, ABD, 2004.
17. Perkins C.E., Elizabeth M.B.R., and Das S., “Ad Hoc On Demand Distance Vector (AODV) Routing”, **IETF RFC 3561**, Virginia, 2003.
18. Murthy C.S.R, Manoj B.S., “Ad-Hoc Wireless Networks-Architectures and Protocols”, **Prentice Hall**, New Jersey, ABD, 2004.
19. Santi P., “Topology Control in Wireless Ad Hoc and Sensor Networks”, **Wiley**, New Jersey, ABD, 2005.