



BİLGİSAYAR AĞLARINDA KULLANILAN KISICI PAKET TIKANIKLIK KONTROL ALGORİTMASI İÇİN SİMULATÖR GERÇEKLEŞTİRİMİ

B. BALCI*

Özet

Bilgisayar ağlarındaki tıkanıklık problemine çözüm olarak çeşitli tıkanıklık kontrol algoritmaları geliştirilmiştir. Bu çalışmada, kısıcı paket tıkanıklık kontrol algoritmasının eğitim amaçlı simulatörü tasarlanmıştır. Bilgisayar ağları dersinde ağ tıkanıklığı ile ilgili temel prensipler verilmekte, ancak laboratuvarlarda ağ tasarımı yapılamamakta ve öğrencilere uygulamalarla tıkanıklık kontrol algoritmaları anlatılamamaktadır. Bu simulatör sayesinde, algoritmanın işleyiş mantığının adım adım gözlenmesi mümkün olacaktır. Yönlendiricilerin kendilerine gelen paketleri alması ve göndermesi ile ilgili kurallar gerçektekine uygun olarak belirlenmiştir.

1. Giriş

Bilgisayar ağlarındaki kalabalık, ağ tıkanıklığı ya da kaynak paylaşım problemi olarak bilinmektedir. Bir başka ifadeyle, tıkanıklık, alt ağda (subnet) fazla paket bulunmasından dolayı oluşan performans düşmesidir. Alt ağda bir yönlendiriciye gelen ve yönlendiriciden gönderilen paket sayısı orantılı olmalıdır [1]. Kalabalığın nedenleri, gelen paket oranının çıkan paket oranından fazla olması [2], yavaş işlemler, paketlerin yeniden gönderimi [3] olarak özetlenebilir.

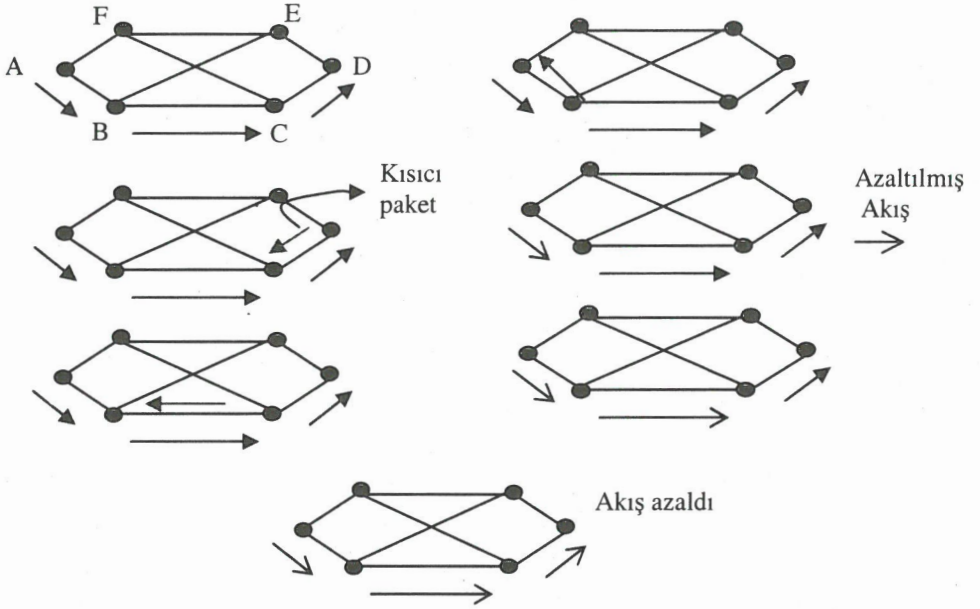
Bilgisayar ağları üzerinden bilginin iletim sırasında, tıkanıklık oluşması veri kaybı ya da geç iletimi, sistem kaynaklarının boşa kullanımı, verimde düşme, doğru sonuç alamama ve kilitleme (deadlock-lockup) gibi durumlar gözlenebilir[4]. Paket anahtarlama ağlarda gözlenen tıkanıklık problemleri, düğüm noktalarındaki paketlerin erişebileceği tampon alanını artırmakla çözümlenebilir[1]. Ancak çok fazla bellek alanı olması durumu, çok az bellek alanı olmasından daha zararlıdır[5]. Tıkanıklık sırasında oluşan problemler çözümlenmiş değil, sadece ertelenmiş olur[3]. Çok hızlı bağlantıya sahip olmak tıkanıklığı gidermede bir çözüm gibi görünse de yerel ağlar yavaş bağlantılarla birbirine bağlandığında, bağlantı noktalarında yine tıkanıklık problemi oluşacaktır [5]. Tüm bağlantılar ve işlemciler aynı hıza sahip olsa bile tıkanıklık oluşabilir[4].

2- KISICI (CHOKE) PAKET TIKANIKLIK KONTROL ALGORİTMASI

Tıkanıklık kontrolünde kullanılan Kısıcı Paket algoritması, global, karşılık veren, açık geribildirimli kapalı devre kontrolü grubuna dahil olarak gösterilmiştir. Hem

bağlantısı yönlendirilmiş (connection oriented- virtual circuits), hem de bağlantısız (connectionless- datagram subnet) ağlarda kullanılırlar.

Şekil 1'de, sadece kaynağı etkileyen kısıcıcı paket (uyarı paketi de denilebilir) algoritmasında A düğümünden D düğümüne paket gönderimi ve D noktasında tıkanıklık oluştuğunda, A düğümüne kısıcıcı paket (uyarı paketi) yollanması, kaynağın bunu aldıktan sonra hızını azaltması olayı basamaklar halinde gösterilmektedir. Tıkalı D düğümünün yolladığı uyarı paketini A kaynağının alıp, hızını azaltarak gönderdiği paketlerin tekrar D düğümüne ulaşması toplam yedi basamakta gerçekleşir .

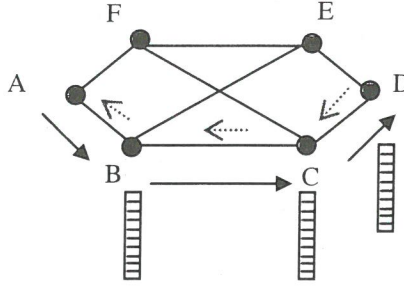


Şekil 1. Sadece kaynağı etkileyen uyarı paketi [1]

Kısıcıcı Paket algoritmasında, bir düğüm, kendi olası eşik değerine erişirse, kaynağa yükünü belli bir oranda azaltmasını bildiren bir azaltma mesajı yollar. Orjinal paket etiketlenir (başlık biti alır) ve yol boyunca başka kısıcıcı paket üretmez. Kaynak, kısıcıcı paket adı verilen bu uyarı mesajı içeren paketi aldığı anda, gönderdiği trafiği belli bir oranda azaltır ve belli bir süre, tekrarlanan uyarı paketlerini ihmal eder. Bu süre sonunda başka kısıcıcı paket gelmiyorsa trafiğini artırır. Kısıcıcı paket gönderimi belli bir eşik değerine bağlıdır, mesela %75 [6]. Genelde ilk kısıcıcı paket, veri akışının %50 azaltılmasına, bir sonraki ise %25 azaltılmasına neden olur. Akış hızındaki artırımlar ise tıkanıklığın çabuk oluşmasına izin vermemek için, daha küçük oranlarda olmaktadır [1].

3- SİMULATÖRÜN YAPISI

Bu çalışmada tasarlanan Kısıcıcı Paket Tıkanıklık Kontrol Algoritması simülâtörü [7], Nesneye Dayalı Programlama Dillerinden DELPHI 4.0 ile tasarlanmıştır. Tasarım için Şekil 2’de görülen altı tane yönlendiriciden oluşan sabit bir ağ modeli kullanılmıştır. Yönlendiricilerin altındaki kutular o yönlendiriciye ait bellek alanlarını göstermektedir.



Şekil 2. Ağ modeli

3.1 Simülâtör Arayüzü

Simülâtörün arayüz tasarımında, Şekil 3’te görüldüğü üzere farklı renkteki dairelerle gösterilen yönlendiricilere ve A...F adları verilmiştir. Tampon alanları kutucuklar halinde bu düğümlerinin altına sıralanmıştır.

Form üzerine *Başla*, *Dur* ve *Kapat* olmak üzere üç buton yerleştirilmiştir. Bunlardan *Başla* butonuna basılması, simülasyonun çalışmaya başlamasını; çalışırken basılması ise tüm değişken değerlerinin sıfırlanarak simülasyonun yeniden başlatılmasını sağlar. *Dur* butonuna basılması, simülasyonun çalışmasını aynı butona tekrar basılana kadar durdurmaya ve buton başlığının *Devam* haline dönüşmesini sağlar. *Kapat* butonu ise simülasyondan çıkmak için kullanılır.

Formun altına hız ayarlarının yapıldığı, A ve D düğüm noktaları ile ilgili bilgilerin yazıldığı tablolar eklenmiştir. Formun sağ üst köşesinde *çalışma prensipleri*, kullanılan değişkenlerin *değer analizleri* ve program hakkında *yardım* alınabilecek bir *Yardım* menüsü eklenmiştir. Kaynaktan düğüme gönderilen paketlerin dışında, tıkanma olan düğümlerden de kaynağa geribildirim olarak gönderilen kısıcıcı paketler kaynak düğümleriyle aynı renklerle belirtilmiştir.

3.2 Çalışma Kuralları

- 1- Veri paketlerinin iletiminde A-B-C-D yolu izlenmektedir. A yönlendiricisi kaynak düğümü olarak, D yönlendiricisi hedef düğümü olarak, B ve C ise paketlerin iletiminde görevli yönlendiriciler olarak çalışmaktadırlar.
- 2- Algoritmanın çalışma mantığının daha kolay anlaşılabilmesi için, kullanılan ağ modelinde paketlerin hareketi tek hat üzerinden sağlanmış, E ve F yönlendiricileri üzerinden veri iletimi yapılmamıştır.

- 3- Tıkanmanın oluştuğu yönlendirici, A kaynağına hızını azaltmasını bildiren uyarı paketini kesikli çizgilerle gösterilen D-C-B-A hattı üzerinden yollar.
- 4- Paket akışının kolay takip edilebilmesi için, kaynaktan yollanan paketler kaynak renginde, tıkalı düğümlerden yollanan kısıcıcı paketler ise o düğüm renginde belirtilmiştir.
- 5- B, C ve D yönlendiricilerinde RAM bellek için 10 paketlik tampon alanı ayrılmıştır.
- 6- Bir yönlendiriciye paket gelişi, yönlendiricinin paket yollamasından daha yavaşsa, yönlendirici paketlerin gelmesini bekler. Eğer bu süreler eşitse, gelen paket direk bir sonraki düğüme iletilir. Paket gelişi, gönderiminden daha hızlıysa, gelen paketler yönlendiricinin tampon alanına yığılır ve kalabalık oluşur.
- 7- Bir düğüme kalabalık oluşması, düğümün tampon alanının en az %60'ının dolu olması olarak kabul edilmiştir. Yani tampon alanında 5'ten fazla paket içeren yönlendirici, kaynağa geri bildirim olarak, hız azaltmasını bildiren uyarı paketi yollar.
- 8- Bir paketin, bir düğümden çıktıktan sonra diğer düğümün tampon alanına yerleşmesi, toplam 6 adımda gerçekleşir.
- 9- Geribildirim paketinin düğümler arasındaki hareketi tek adımda gerçekleşir.
- 10- Simulasyon toplam 50 paket için tasarlanmıştır.
- 11- A kaynağının hızı, başlangıçta 1sn olarak belirlenmiştir. A yönlendiricisi, ağdaki B yönlendiricisine her 1 sn'de bir 1 paket gönderir . Ancak, kısıcıcı paket durumuna göre hızını azaltma ya da artırması söz konusudur.
- 12- B ve C yönlendiricilerinin hızları sabittir. B yönlendiricisi kendisine gelen paketleri 1,5sn'lik aralıklarla C'ye iletirken, C de 2 sn'de bir D'ye paket gönderir.
- 13- D düğümü ise aldığı paketleri 1-4 sn arasında rasgele değişen hızlarda, sanki sanal bir düğüme paket yolladığı kabul edilerek, tampon alanını boşaltır.
- 14- A düğümü uyarı mesajı içeren ilk kısıcıcı paketi aldığı anda normal hızını (Durum0-1 sn), *Durum1* konumuna geçerek azaltır ve her 2 sn'de 1 paket göndermeye başlar. Ağa 4 paket çıkartana kadar bu hızda kalır ve gelebilecek diğer kısıcıcı paketleri ihmal eder. Bu süre sonunda gelen kısıcıcı paket yoksa, *Durum3* konumuna geçerek hızını artırır ve 1,5 sn'de 1 paket yollamaya başlar ve ağa 4 paket çıkartana kadar bu hızda kalır. Eğer geri bildirim geliyorsa, hızını 1 kademe daha azaltarak *Durum4* konumuna geçer ve her 3 sn'de 1 paket yollar. Ağa 2 paket çıkartana kadar, *Durum4* konumunda kalır. Bundan sonra hız azaltma yapılmaz. Hızını 2 paket için 3 sn olarak ayarlar, yani *Durum3* konumuna geçer.

Bu değerler Şekil 4 (b)'deki tablolarda görülmektedir.

3.3 Simulatör Uygulama Adımları

Kısıcıcı paket tıkanıklık kontrol algoritması simulatörü çalıştığından itibaren adım adım, gelen kısıcıcı paket sayısına göre A düğümünün hız değişiklikleri Şekil 3'te gösterilmektedir.

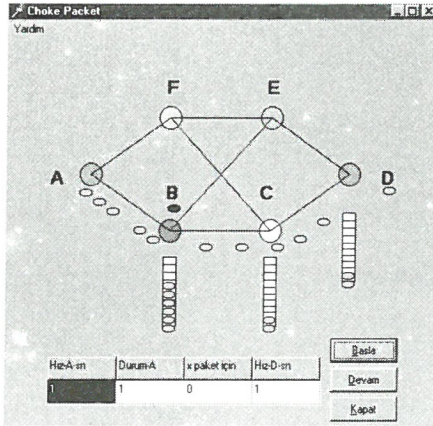
Şekil 3.(a)'da, tablodan da anlaşılacağı gibi, *DurumA* değişkeninin değeri önceden 0 iken yani, her 1 sn'de 1 paket yollarken, B düğümünün, tampon alanında 6 paket

olduğundan tıkanmış, ve B düğümü kısıcı paketini oluşturmuştur. Tabloda DurumA değişkeni 1 konumuna geçmiştir. B düğümü sabit 1,5 sn'de, C düğümünü sabit 2 sn'de bir paket göndermektedir. Tablodan da görüldüğü gibi D düğümü paket gönderme hızını rasgele 1 sn olarak seçmiştir. Şekil 3.(b)'de ise artık A düğümü B'nin yolladığı kısıcı paketi alarak hızını azaltmış, DurumA değişkeni 1 değerini aldığından, 2 sn'de 1 paket göndermeye başlamıştır. Bu hızda, 3 paket daha gönderecektir. Ancak B düğümünde tıkanma devam ettiğinden, B kısıcı paket yollamaya devam etmektedir. D ise tampon alanını boşaltma süresini 2 sn'de 1 paket olarak belirlemiştir.

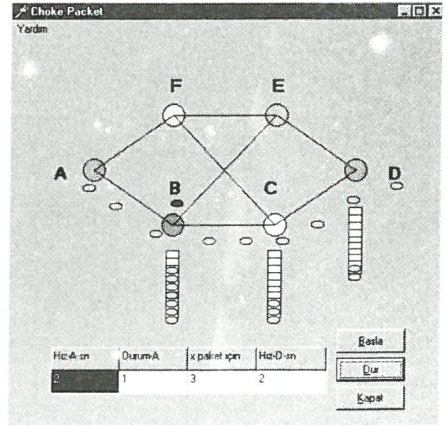
Şekil 3.(c)'de, henüz C ve D düğümlerinde tıkanma olmamıştır. A düğümü kısıcı paket almaya devam ettiğinden DurumA değişkeni 2 değerini almıştır. Bu da, A düğümünün hızını 4 sn'de 1 paket göndermeye ayarlayacağını belirtmektedir. Şekil 3.(d), artık B düğümünde tıkanıklık kalmadığını göstermektedir. Çünkü A düğümü 4 sn'de bir paket gönderirken, B düğümü sabit 1,5 sn hızında tampon alanını boşaltmaktadır. B'ye paket geliş hızı, gönderim hızından daha yavaştır ve tıkanıklık ortadan kalkmaktadır.

Şekil 3.(e), A düğümü hala 4 sn'de bir paket yollamaktadır. Bir sonraki harekette DurumA değişkeni 4 değerini alarak, hızı 3 sn'de 1 paket göndermek üzere ayarlayacaktır. Şekil, önce tıkanan D düğümü ve sonra tıkanan C düğümünün yolladığı kısıcı paketleri B düğümünün aldığı durumu göstermektedir. C düğümüne B'den 1.5 sn'de bir paket gelmekte, C ise 2 sn'de bir paket göndermekte olduğundan C düğümü tıkanmıştır. Şekil 3.(f)'de A düğümü B'ye 3 sn'de bir paket göndermektedir. B ve C düğümlerinde tıkanıklık yoktur. D düğümünün yolladığı kısıcı paketin D-C arasındaki hareketi görülmektedir.

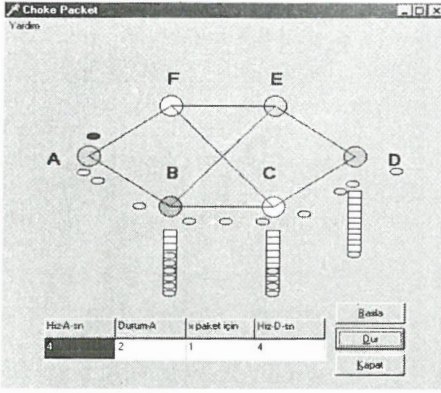
Bir sonraki adım ise Şekil 3.(g)'de görülmektedir. A düğümünde toplam 50 paket üretilmiş ve 50. paket B tampon alanı içine yerleşmiştir. B,C ve D düğümlerinde tıkanıklık yoktur. Paketler normal akışına devam ederek, D düğümünün dışına konumlanırlar ve program sonlanır.



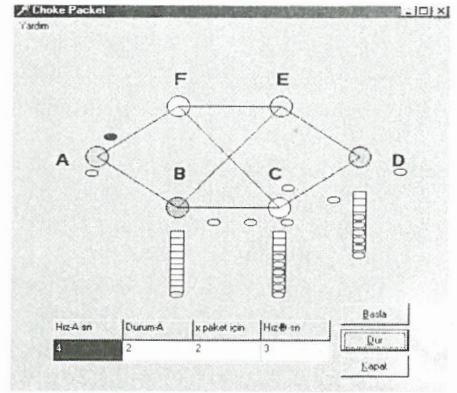
(a)



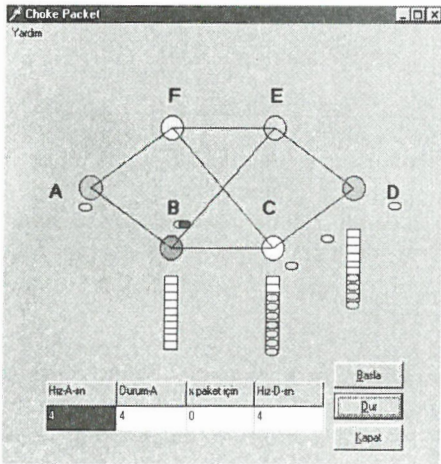
(b)



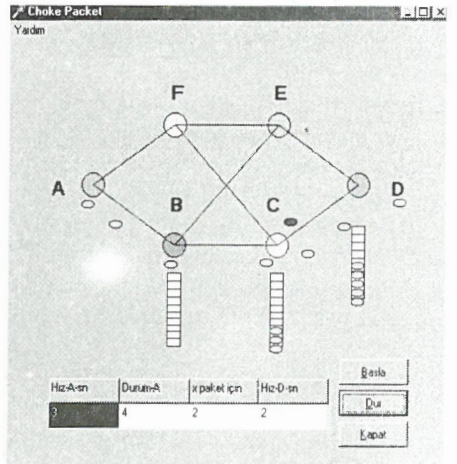
(c)



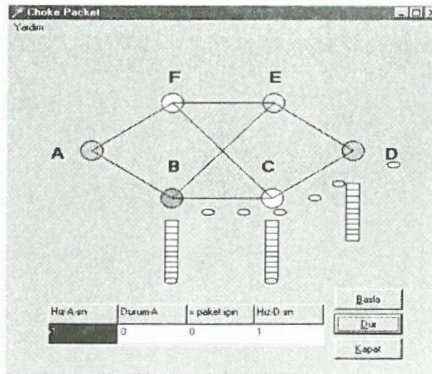
(d)



(e)



(f)

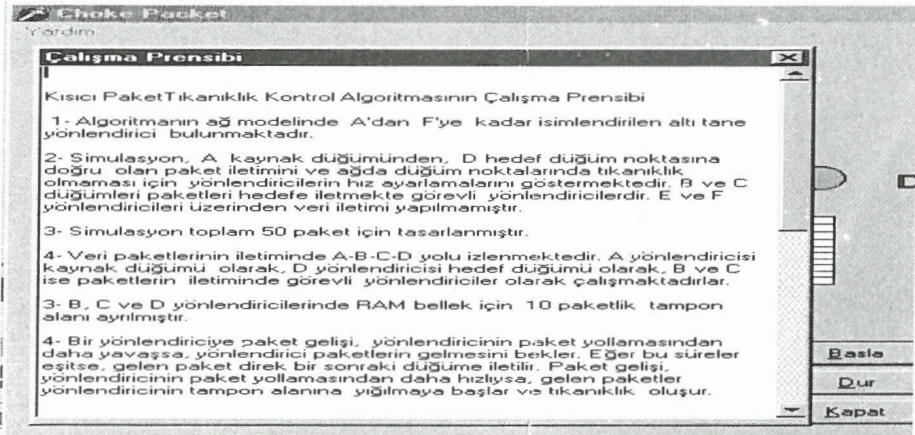


(g)

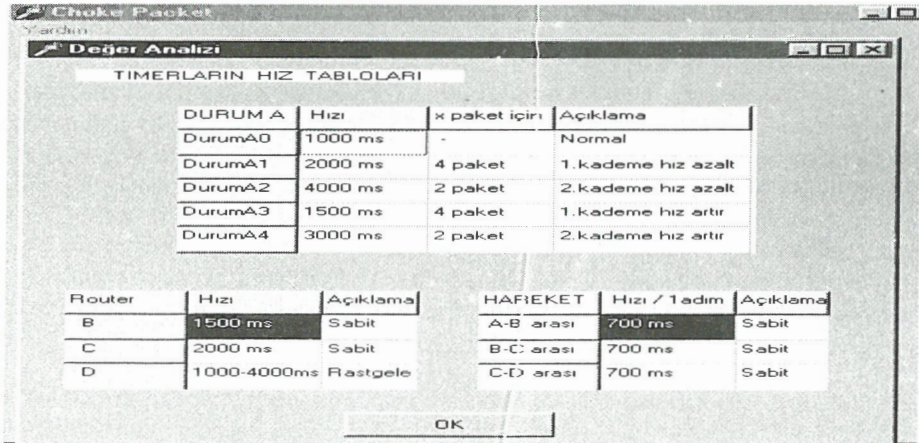
Şekil 3. a,b,c,d,e,f,g Simulasyon çalışma adımları

3.4 Yardım Menüsü Ekranı

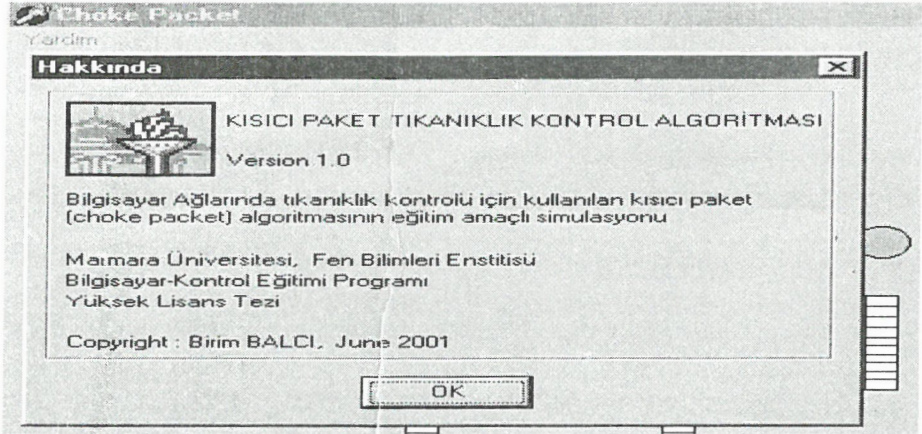
Simulatöre çalıştırıldığında sol üst köşede bulunan *Yardım* menüsünde Çalışma prensipleri, Değer Analizi, ve Hakkında olmak üzere üç başlık bulunmaktadır. Şekil 4'te Kısıcılı Paket arayüzü üzerine yerleştirilen yardım menüsü formları görülmektedir.



(a)



(b)



(c)

Şekil 4. Yardım menüsü formları .

(a) Çalışma Prensipleri (b) Değer Analizi (c) Hakkında

4- SONUÇ

Kısıcıcı paket algoritması, ağa kontrol mesajlarından dolayı ekstra yük getirirse de iyi işleyen bir algoritmadır[2]. Bu algoritmanın avantajı, kaynağın istediği kadar paket yollayabilmesidir. Gönderilen paket sayısı fazlaştığında, alıcı durumu bildirir. Dezavantajları ise şunlardır [8,9] : Düğümün ne kadar yavaşlayacağını bilmesi zordur. Bu yavaşlama oranı, düğümün ne kadar trafik yolladığına, tıkanıklığın ne kadarına sebep olduğuna, ve tıkalı bölgenin toplam kapasitesinin ne olduğuna bağlıdır. Bu bilgi pratikte gerçekten uygun değildir. Kısıcıcı paketi alan düğümün, mevcut paketler hedefe varana kadar kısıcıcı paket gönderimi devam edeceğinden, bir süre gelecek kısıcıcı paketleri ihmal etmesi gerekir ki bu süre için belirlenmiş kesin bir değer yoktur . Ayrıca kısıcıcı paketin kaynağa gitmesi çok zaman alır. Eğer kaynak hızlı bir makine ise, bu sürede ağa çok fazla miktarda paket göndermiş olacaktır .

Bu çalışmada tasarlanan simülör, laboratuvar ortamı olmadığından daha çok teorik olarak işlenen Bilgisayar Ağları dersinde öğrencilerin, tıkanıklık kontrolünde kullanılan Kısıcıcı Paket algoritmasının çalışma mantığını daha iyi anlaşılmasını sağlayacak eğitim amaçlı bir simülördür. Elde edilen sonuçlar, bu tıkanıklık kontrol algoritmalarının gerçek uygulama sonuçlarıyla aynı paralelliktedir. Açıklanan kurallar, bu çalışmada tasarlanan ağ modeli için kabul edilmiş kurallardır. Simülör her çalıştırıldığında farklı değerlerin elde edileceği ve değişik işlemlerin elde edileceği unutulmamalıdır. Ağ modelindeki paket iletiminde izlenen yol, yönlendiricilerin tampon alanı kapasiteleri, paket iletim hızı, düğümlerin hız ayarlamasında belirlenen kurallar tamamen birbirleriyle ilişkili olduğundan, bunlardan herhangi birinin farklı şekilde değerlendirilmesi sonucunda simülasyonlar beklenen sonuçları vermeyebilir.

KAYNAKÇA

- [1] Tanenbaum, A.S., *Computer Networks*, Prentice-Hall, 3th Ed.; United States of America, (1996) 374-391
- [2] Fortier, P.J., *Handbook of LAN Technolog*, McGrawHill, 2nd Ed.; New York, USA, (1992) 249-258
- [3] Filipiak, J.: *Real Time Network Management* ,Amsterdam, North-Holland, (1991) 27-37, 61-71
- [4] Balcı, B., *Bilgisayar Ağlarında Kullanılan Delik Kova Tıkanıklık Algoritması için Simulatrör*, Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, II.Bilgi Teknolojileri Kongresi Özel Sayısı, 2004
- [5] Jain R.: *Congestion Control in Computer Networks : Issues and Trends*, IEEE Network Magazine, (May 1990) 24-30
- [6] Yang, C., Reddy, A.V.S., *A Taxonomy for Congestion Control Algorithms in Packet Switching Networks*, IEEE Network Magazine, Vol.9 no.5 (July 1995)
<http://www.comsoc.org/pubs/surveys/yang/yang-orig.html>
Erişim Tarihi : 23.09.2000
- [7] Balcı, B., *Bilgisayar Ağlarında Tıkanıklık Kontrol Algoritmaları*

A SIMULATOR FOR CHOKE PACKET ALGORITHM TO CONGESTION CONTROL IN COMPUTER NETWORKS

B. BALCI

Abstract : As a solution to the congestion control problem in computer networks, various control algorithms are developed. In this study, a Simulator for education has been designed for the Choke Packet Congestion Control Algorithm. The basic principal of the network congestion are given in the computer network lessons, but a network design can not be done in the laboratories for practices. By the simulator designed in this study, it will be possible to see how the algorithm Works step by step. The rules about receiving and sending the packets of the routers are taken from real world.

Key Words : Congestion Control, Choke Packet Algorithm, Computer Networks, Simulator

* Maltepe Üniversitesi, Mühendislik-Mimarlık Fakültesi, Bilgisayar Mühendisliği
Bölümü, İstanbul, Türkiye
birimb@maltepe.edu.tr
birimb@yahoo.com