Mobil Cihazlarda Farklı Ağ Bağlantılarını Kullanarak TCP Performansının Arttırılması

[*1] Necati Ersen ŞİŞECİ ve [2]Doç. Dr. Hacı Ali MANTAR
[1] Siber Güvenlik Enstitüsü, TÜBİTAK BİLGEM
[2] Bilgisayar Mühendisliği Bölümü, GYTE

Özet

Teknolojilerinin ilerlemesi ve mobil cihazların daha akıllı hale gelmesi ile her ortamda Internet'e bağlanma imkanı doğmuştur. Hücresel ağ bağlantısı ve kablosuz internet teknolojileri ile mobil cihazlar kesintisiz olarak Internet'e ulaşılabilmektedir. Bu bağlantı çeşitlerinin aynı anda kullanılabilmesi, Internet kullanım hızını ve performansını arttırmaktadır. Bu makalede önerilen yöntem her iki bağlantı çeşidinin aynı anda kullanılabilmesi durumunda TCP bağlantılarını izleyerek kullanılacak bağlantı arayüzünün belirli parametreler ve öncelik mekanizmasına bağlı olarak seçilmesine dayanmaktadır. Önerilen yöntemin test edilebilmesi için sanallaştırılmış FreeBSD işletim sistemi üzerinde bir çekirdek modülü geliştirilmiş ve çeşitli performans testleri yapılmıştır.

Anahtar kelimeler: multihome, connection-aware, mobile, TCP performance, FreeBSD

1. Giriş

Teknolojinin ilerlemesi ve mobil cihazların daha akıllı hale gelmesi ile her ortamda Internet'e bağlanma imkânı doğmuştur. 3G, 4G ve kablosuz internet gibi teknolojiler ile mobil cihazlar kesintisiz olarak Internet'e ulaşılabilmektedir. Taşınabilir bilgisayarlarda kablolu, kablosuz (IEEE 802.11) ve hücresel ağ bağlantıları (HSPA+,HSPA, 3G vb) kullanılabilmektedir[7]. Mobil cihazlarda ise kablosuz ve hücresel ağ bağlantıları kullanılabilmektedir. HSPA, GPRS gibi teknolojilerin, IEEE 802.11 tabanlı kablosuz ağlara göre kapsama alanı çok geniş olmasına rağmen bant genişlikleri düşüktür. Bu bağlantı teknolojileri ayrı ayrı kullanılabildikleri gibi aynı anda da kullanılabilmektedir. Kullanıcı için Internet erişiminde tek değişen şey fiziksel katmandır.

Aynı anda birden fazla ağ bağlantısı yapılabilen cihazlar aracılığı ile farklı Internet servis sağlayıcılar üzerinden Internet'e erişmek mümkündür. Bir cihaz, ağ bağlantısı yaptığı tüm arayüzleri için bağlandığı Internet servis sağlayıcısının verdiği ağ erişim bilgilerini (IP adresi, alt ağ maskesi, ön tanımlı ağ geçidi, DNS sunucusu vb.) alacaktır. Örneğin iş yeri, ev, kafe gibi ortamlarda bulunan kablosuz ağ bağlantıları kullanıldığında ortamda bulunan DHCP sunucusu ağ bağlantısı yapan cihazlara ağ erişim bilgilerini yollayacaktır. Aynı şekilde hücresel ağ bağlantısı kullanıldığı zaman ise, şebeke operatörü tarafından ağ erişim bilgileri cihaza yollanacaktır. Atanan IP adresleri, RFC1918'ye göre dağıtılabileceği gibi gerçek IP adresleri olarak da dağıtılabilmektedir [8]. RFC1918'e göre dağıtılan IP adresleri, Internet erişimi için RFC3022'de anlatıldığı gibi basit ağ adres dönüşümünden geçirilmelidir [9]. Bu işlem genellikle ağ geçidi

*İlgili kişi: Necati Ersen ŞİŞECİ Adres: TÜBİTAK BİLGEM Siber Güvenlik Enstitüsü, Gebze / KOCAELİ.
E-posta: necati.siseci@tubitak.gov.tr

görevi gören DSL modemler, kablosuz erişim noktaları veya ağ geçidi olan cihazlar tarafından yapılmaktadır.

Aynı cihaz üzerinde birden fazla ağ bağlantısına ihtiyaç duyulmasının en önemli sebeplerinden birisi bant genişliğinin arttırılmasıdır. Bunun yanı sıra, bağlantı yedekliği, hat güvenliği, erişilebilirlik gibi sebepler de birden fazla ağ arayüzü kullanım ihtiyacını doğurmuştur. Bazı kurumlar, ana yönlendirici cihazlarında birden fazla Internet Servis Sağlayıcı (ISS)'den hizmet alarak Internet'e bağlanmaktadırlar. Bağlantı hatlarından herhangi birisinde problem olması durumunda, kurumun Internet'i etkilenmeyeceği için hem kurum Internet kullanıcıları hem de kuruma Internet üzerinden erişen kullanıcılar bu kesintiden etkilenmeyecektir. Aynı şekilde mobil cihazlarda birden fazla ağ arayüzü kullanımında da benzer sebepler ek olarak ek olarak hücresel ağ bağlantı ücretlerinin kablosuz ağ bağlantılarına göre daha pahalı olması ve farklı ağ tiplerinin kapsama alanlarının farklı olması da eklenebilir.

Birden fazla farklı ağ bağlantısının aynı anda kullanılması ile ilgili yaptığımız bu çalışmadaki amaçlarımızdan birisi yapılacak değişikliklerin sadece kullanıcı tarafında yapılmasıdır. Kullanıcı cihazlarında güncelleştirme yapmak, ağ geçidi olan ve Internet üzerinde bulunan aktif cihazların TCP/IP yığınında değişiklik yapmaktan çok daha kolaydır. Diğer bir amaç ise kullanıcı tarafından belirlenecek politika ile ağ arayüzlerine ağırlık verilebilmesidir. Örneğin 802.11 tabanlı kablosuz ağdaki kayıp oranı arttığında, sistem otomatik olarak hücresel ağ bağlantısını daha fazla kullandırmalıdır.

Benzer çözümlere bakıldığında farklı ağ arayüzlerinin bant genişliklerinin birlikte kullanımı ile ilgili çalışmalar öne çıkmaktadır. Farklı katmanlarda önerilen çözümler mevcuttur. Ancak her çözümün kendine göre avantajları olduğu gibi dezavantajları da mevcuttur. Örneğin [1] de anlatılan yöntemde, bir TCP eklentisi geliştirilmiştir ancak geliştirilen TCP eklentisini tanımayan güvenlik duvarları ilgili paketleri düşürebilir, saldırı tespit sistemleri de bu paketler için çeşitli alarmlar üretebilir. Benzer şekilde MultiPath TCP[5]'nin getirdiği büyük avantajlar olmasına rağmen TCP Eklentisi olarak geliştirilmiş olması aynı problemlere sebep olabilir. Benzer şekilde *Stream Transmission Control Protocol* [2][3][4] geliştirilmiştir. Hem istemcinin hem de sunucunun birden fazla IP adresi olmasını destekler. SCTP'de dinamik olarak istemci ve sunucu IP adreslerinin değiştirilmesini sağlayan bir eklenti mevcuttur. Bu protokolün getirdiği bir çok avantaj olmasına rağmen henüz standartlaşmamış olması ve bazı işletim sistemleri tarafından tam olarak desteklememesi en büyük dezavantajlarındandır. Aynı şekilde SCTP protokolünün sağlıklı çalışabilmesi için güvenlik duvarı gibi cihazların da bu protokolü desteklemesi gerekmektedir. Ribeiro ve diğerlerinin önerdiği yöntem [6] ise gecikme toleransı düşük olan multimedya uygulamaları için gönderilme ve geri geliş yolunun farklı seçilip, en düşük RTT değerlerine sahip olan yolların tercih edilmesine dayanmaktadır. Bu makalede önerilen yöntemde ise bağlantının paketlerinde herhangi bir değişiklik yapılmamaktadır. Bu sebeple, ağ geçidi, güvenlik duvarı, saldırı tespit sistemi, yönlendirici gibi sistemler için herhangi bir değişiklik veya

güncelleme gerekmemektedir. TCP Eklentilerinde olduğu gibi hem istemcide hem de sunucuda değişiklik yapmaya gerek kalmamaktadır. Bu yöntemde sadece istemci tarafında değişiklik yapılması önerilmektedir.

2. Yöntem

Önerilen yöntem, yeni açılacak olan TCP bağlantılarının en uygun ağ arayüzü üzerinden gönderilmesine dayanmaktadır. Uygun ağ arayüzünün seçimi için daha önceden açılmış ve izlenen TCP bağlantılara ait *Round-trip time (RTT)* değerleri ve ağ arayüzü öncelikleri gibi parametreler kullanılmaktadır.

Herhangi bir cihaz, bir ağa bağlandığı zaman, o ağda olmayan cihazlara IP kullanarak ulaşabilmek için yönlendirme tablosuna ihtiyaç duymaktadır. Yönlendirme tablosunda, belirli ağ veya cihaz adresleri için özel olarak girilmiş yönlendirme girdileri bulunabileceği gibi tanımlı olmayan tüm ağ adresleri için kullanılacak olan ön tanımlı ağ geçidi de bulunabilmektedir. Cihaz üzerinde çalışan işletim sistemi, oluşturulan paketin hangi ağ arayüzü üzerinden ve hangi cihaza gönderileceğine, yönlendirme tablosuna bakarak karar vermektedir. Bir işletim sistemi üzerinde herhangi bir ek ayar yapılmadan sadece bir tane ön tanımlı ağ geçidi bulunabilmektedir. Örnek bir yönlendirme tablosu Şekil 1'de gösterilmektedir. Bazı işletim sistemlerinde birden fazla ön tanımlı ağ geçidi tanımlanabilmesine rağmen sadece bir tanesi kullanılabilmektedir.

```
Internet:
Destination        Gateway            Flags    Refs      Use  Netif
default            192.168.200.1      UGS         0     7715    em0
127.0.0.1          link#4             UH          0       56    lo0
192.168.200.0/24   link#1             U           0        0    em0
192.168.200.129    link#1             UHS         0        0    lo0
```

Şekil 1 Örnek yönlendirme tablosu

Birden fazla ağ bağlantısı olan cihazlar da aynı yönlendirme tablosu kullanılmaktadır ve tüm yönlendirme işlemleri bu tabloya göre yapılmaktadır. Bu durumda ön tanımlı ağ geçidi için bir öncelik veya bir politika kullanılması gerekmektedir. Bazı işletim sistemleri birden fazla yönlendirme tablosuna *(Forwarding Informatin Base - FIB)* izin vermektedir. Bu durumda her yönlendirme tablosuna bir adet ön tanımlı ağ geçidi eklenebilmektedir. *FIB*, genelde yönlendirici cihazlar üzerinde kullanılmaktadır.

FreeBSD [10] işletim sistemi 16 farklı yönlendirme tablosuna izin vermektedir [11]. FreeBSD'de bulunan *setfib* [12] komutu ile çalıştırılacak olan program için *FIB* tablosu önceden belirlenebilmektedir.

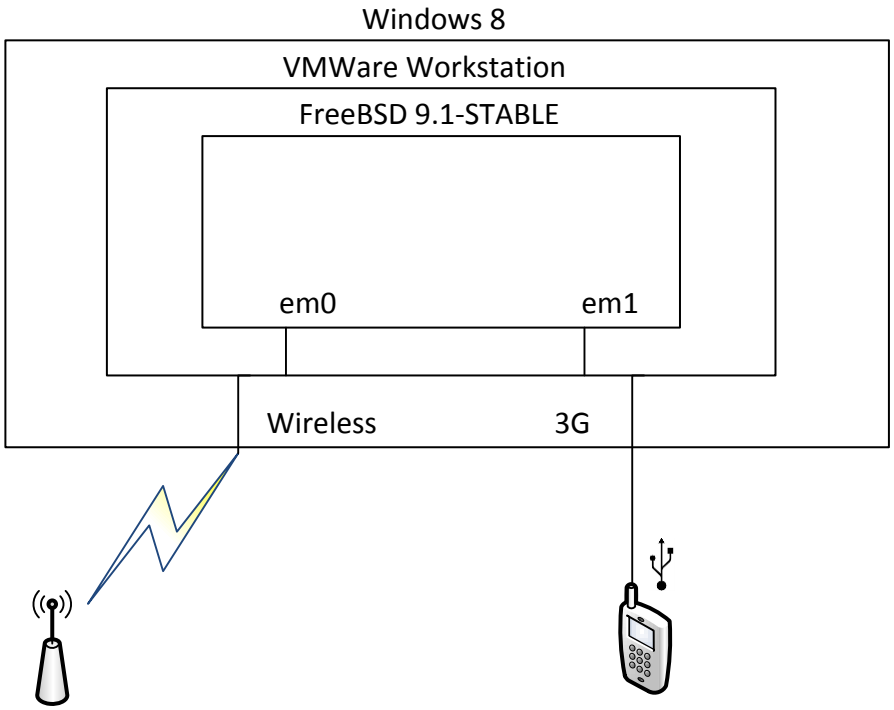
```
[root@tez ~]# setfib 1 netstat -nr
Routing tables

Internet:
Destination        Gateway            Flags    Refs      Use  Netif
default            192.168.42.129     UGS         0     5161    em1
10.100.0.0/16      link#1             U           0        0    em0
127.0.0.1          link#4             UH          0        0    lo0
192.168.42.0/24    link#2             U           0        0    em1
```

Şekil 2 FIB 1 tablosu örneği

RTT, bir paketin karşı cihaza gönderilip, karşı cihazdan gelen alındı bilgisi alınana kadar geçen zaman olarak nitelendirilir. Ping süresi olarak da bilinir ve *ping* komutu kullanılarak ölçülebilir. *RTT*'nin büyük olması karşı cihaz ile aramızdaki bağlantının yavaş olması anlamına gelir. Önerilen yöntemde, yapılan bağlantıların *RTT* değerlerinin sürekli takip edilmesi gerekmektedir.

NewTCP [13] araştırmaları sırasında, gecikme ve hız tabanlı tıkanıklık algoritmalarında kullanılmak üzere *ERTT* [14] isimli bir çekirdek modülü geliştirilmiştir. Bu modül FreeBSD 9.0 [15] ile gelen *hhook* [16] ve *khelp* [17] çekirdek programlama arayüzleri aracılığı ile çalışmaktadır. Bu modül sayesinde TCP bağlantılarına ait *RTT* değerleri sürekli alınabilmektedir. Önerilen yöntem için geliştirilen kodlar sanallaştırılmış ortamda, FreeBSD üzerinde C dili kullanılarak çekirdek modülü olarak geliştirilmiştir. Açılan bağlantıların izlenebilmesi ve yeni açılacak bağlantıların kullanacağı arayüzün değiştirilebilmesi için FreeBSD'nin *connect* [18] sistem çağrısı değiştirilmiştir. Test ve geliştirme ortamı Şekil 3'de gösterilmiştir.



Şekil 3 Test ve geliştirme ortamı

2.1. Önerilen Model

Önerilen yöntemde istemci tarafından yapılan tüm TCP bağlantılarının izlenmesi ve çeşitli parametrelere göre yeni açılacak olan bağlantıların kullanması gereken ağ arayüzünün seçimi hedeflenmektedir. Bu parametrelerden arayüz önceliği, ön tanımlı olarak verilebildiği gibi daha sonradan da *sysctl* [19] komutu yardımı ile değiştirilebilmektedir. Diğer tutulan parametreler ise *Time-To-Live (TTL), RTT*, gönderilen ve alınan paket sayıları gibi bağlantıya özgü parametrelerdir.

Önerilen yöntemde işletim sisteminde bulunan *connect* sistem çağrısı değiştirilmiştir ve yapılan her yeni bağlantı için, bağlantıya özel bilgiler bir tabloda tutulmuştur. Yapılan bağlantılar sürekli izlenmiş ve RTT değerleri tabloda güncellenmiştir.

İşletim sisteminin *connect* sistem çağrısı değiştirildiği için, herhangi bir yazılım *connect* sistem çağrısını kullanarak bir bağlantı açmak istediğinde ilk olarak geliştirilmiş olan kod çalışmaktadır. Daha önceden bağlanılmış bir hedefe tekrar bağlanılmak istendiğinde IP adresi ve hedef port numarasına göre tablodan en uygun *FIB* tablosu bulunmaktadır. *FIB* tablosu bulunurken, daha önceden aynı hedefe doğru yapılmış olan bağlantıların *RTT* değerleri, daha önceki bağlantıların paket sayıları ve ağ arayüzlerinin öncelik değerleri kullanılmaktadır. Daha uygun olduğuna karar verilen FIB tablosu yardımı ile bağlantı isteğinin gönderileceği ağ arayüzü seçilir ve soket parametreleri değiştirilir. Değiştirilmiş soket parametreleri kullanılarak işletim sisteminin gerçek *connect* sistem çağrısı çalıştırılır.

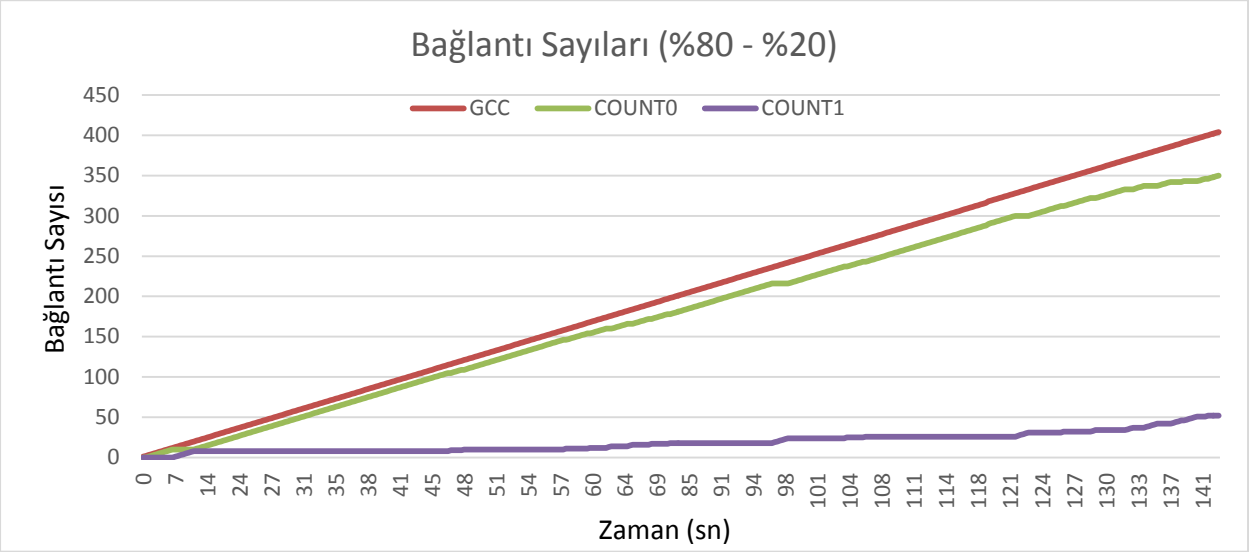
Bağlantı sırasında hedefe gidip gelen paketlerin *RTT* değerleri *h_ertt* yardımı ile hesaplanır ve daha sonra aynı hedefe açılacak olan yeni bağlantılara referans olması için tablo güncellenir. Herhangi bir hedefe yapılan ilk bağlantı ilk ağ arayüzü üzerinden gönderilir. Daha sonra aynı hedefe yapılan sonraki bağlantılar sırasıyla diğer ağ arayüzlerinden yapılarak, tüm ağ arayüzlerinin ilgili hedefe doğru olan bağlantısının parametreleri alınır. Tüm ağ arayüzlerinin *RTT* değerleri alındıktan sonra yeni yapılacak bağlantılar için kullanılacak *FIB* tablosu hesaplanarak ilgili ağ arayüzünden bağlanılması sağlanır.

3. Test Ortamı ve Sonuçları

Yapılan testler aynı hedefe doğru paralel 10 farklı bağlantı olacak şekilde ve farklı ağ arayüzü öncelikleri ve Internet ortamı kullanılarak yapılmıştır. Sanallaştırılmış FreeBSD işletim sistemine farklı iki ağ bağlantısı köprü (bridge) modda bağlanmıştır. Internet bağlantısı için kullanılan arayüzler bağlantı hızı 4Mbps olan kablolu ağ, diğeri ise hücresel ağ bağlantısıdır. İşletim sistemi güvenlik duvarından anlık olarak bağlantı sayıları alınmıştır. Paralel HTTP GET istekleri gönderilerek ölçüm yapılmıştır. Yapılan testlerde hedef sunucu olarak, yurt dışında bulunan bir kiralık sunucu seçilmiştir.

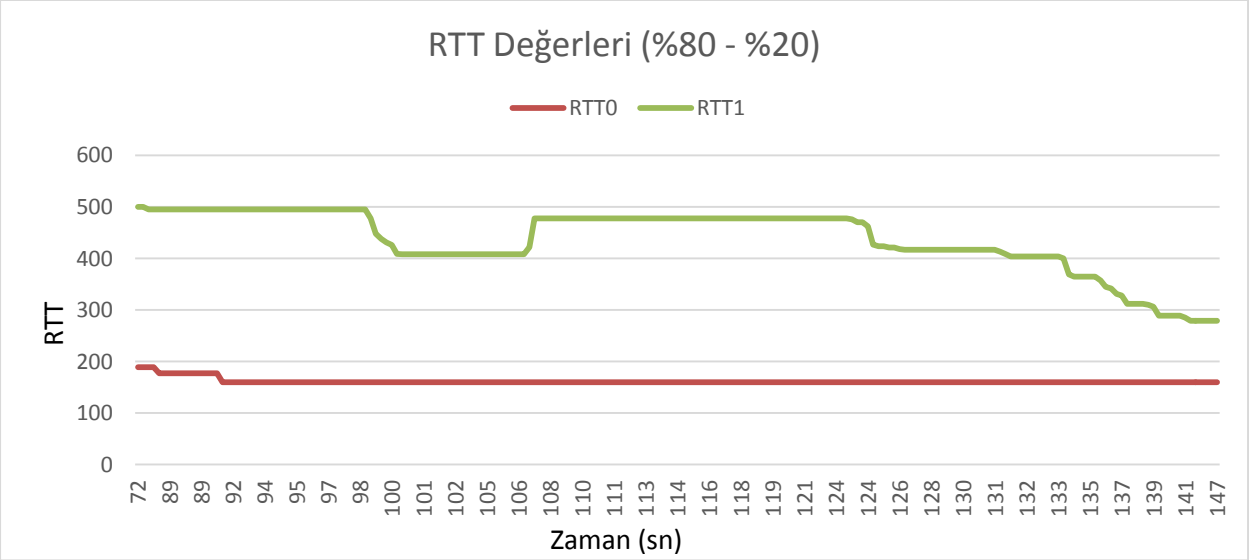
Tüm parametrelerin ve yapılan isteklerin aynı olduğu varsayılarak, kablolu ağ bağlantısı için öncelik %80 ve hücresel ağ bağlantısı için %20 verildiğinde her 100 bağlantının 20'sinin hücresel ağ bağlantısından gitmesi beklenir. Ancak *RTT* değerlerinin değişken olması bu sonucu etkiler.

Şekil 4,5 ve 6'da toplamda 400 adet HTTP GET isteği gönderilmiştir.

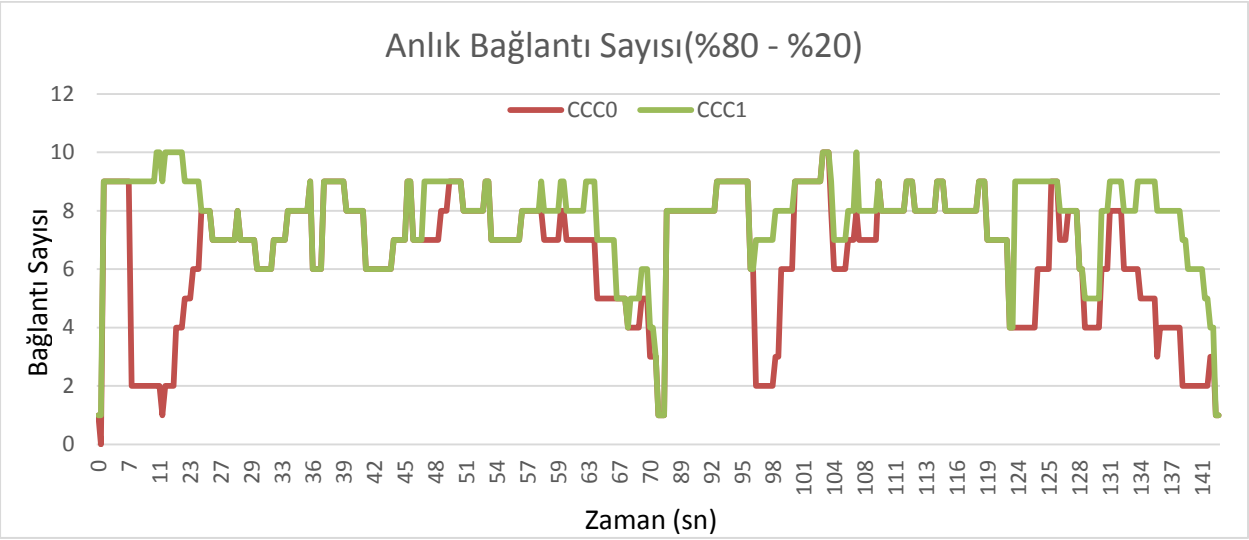


Şekil 4 80-20 öncelik testi için bağlantı sayıları

Şekil 4'de kullanılan *GCC* kısaltması, toplam bağlantı sayısı anlamına gelmektedir. *COUNT0* ve *COUNT1* kısaltmaları, sırasıyla ağ arayüzü başına yapılan toplam bağlantı sayısını göstermektedir.



Şekil 5 80-20 öncelik testi için RTT değerleri

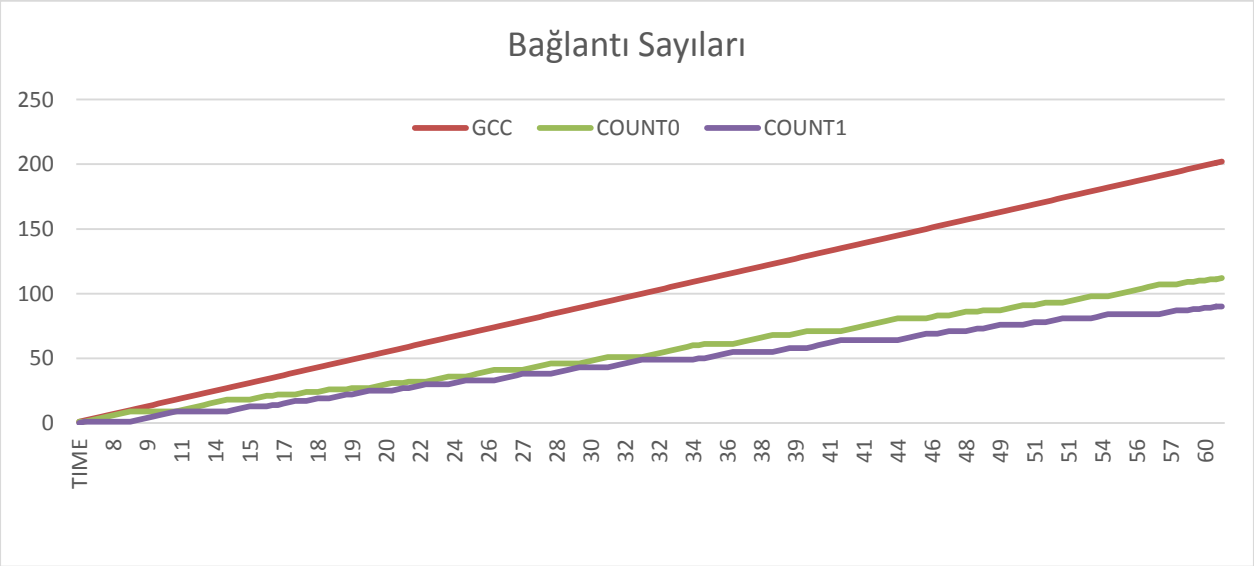


Şekil 6 80-20 öncelik testi için anlık bağlantı sayıları (Toplamalı)

Şekil 6'da her iki hattın anlık bağlantı sayıları toplamalı olarak görülmektedir. *CCC0* ve *CCC1* sırasıyla, ağ arayüzlerindeki anlık bağlantı sayılarıdır. Her iki bağlantı sayısının toplamı, sistemin anlık bağlantı sayısını vermektedir.

Yapılan testlerde her iki ağ bağlantısının *RTT* değerlerinin birbirine yakın olması durumunda bağlantı sayılarının seçilen oranlara yaklaştığı gözlemlenmiştir.

%50-%50 değerleri kullanılarak yapılan testte *RTT*'nin ağ arayüzü seçimine etkisi daha açık bir şekilde görülmektedir. Şekil 7'de toplam 200 adet HTTP GET isteği gönderilmiştir.



Şekil 7 50-50 öncelik testi bağlantı sayıları

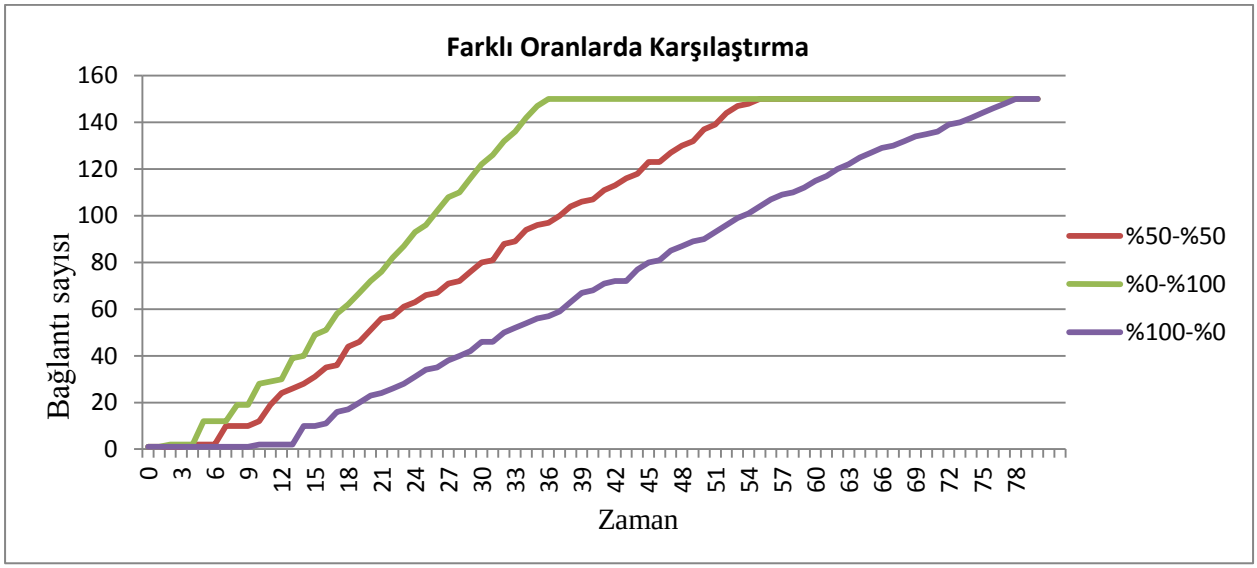
Şekil 7'de görüldüğü üzere, her iki hat içinde eşit öncelik verilmesine rağmen hatların bağlantı sayılarının farklı olduğu görülmektedir. Sadece hat önceliği parametre olarak kullanılsaydı, her iki hat da aynı miktarda bağlantı yapacağı için grafikte gösterimi aynı olacaktı. Ancak kullanılan parametreler sebebi ile bağlantı miktarının değiştiği grafikte görülebilmektedir.

Yapılan bir başka test setinde ise toplamda 150 HTTP GET isteği gönderilerek 3 farklı test yapılmıştır. Bu testlerdeki ağ bağlantılarının kullanım oranları Tablo 1'deki gibi seçilmiştir.

Tablo 1 Testte kullanılan oranlar

Test Numarası	**Kablosuz Ağ Bağlantısı kullanım oranı**	**Hücresel Ağ Bağlantısı kullanım oranı**
1	%100	%0
2	%0	%100
3	%50	%50

Tablo 1'e göre yapılan test sonuçları Şekil 8'de verilmiştir. Sonuçlardan da anlaşılacağı üzere, hücresel ağ bağlantısı ve kablosuz ağ bağlantısının %50 oranlarında kullanılması sonucu yapılan test, sadece kablosuz ağ bağlantısının kullanılması durumunda avantaj sağlamaktadır.



Şekil 8 Farklı oran testlerinde bağlantı sayıları

4. Sonuç ve Gelecekte Yapılacak Çalışmalar

Yapılan testler, aynı anda farklı ağ arayüzlerinin birlikte kullanılması sonucu istemcinin TCP performansının arttığını göstermiştir. Sadece bir ağ arayüzünün kullanılması durumunda alınan performans, diğer ağ bağlantılarının da kullanılmaya başlaması ile artmıştır. Ancak TCP performansını, hücresel ve kablosuz ağ bağlantılarının kayıp oranları, bant genişliği gibi parametreleri etkilemektedir. Hızlı bir kablosuz bağlantı ve yavaş bir hücresel ağ bağlantısının, %50-%50 oranlarda kullanılması durumunda performans olumsuz etkilenmektedir. Bu gibi durumlarda önceliğin değiştirilmesi performansı arttıracaktır.

Geliştirilen yöntemde anlık olarak ağ bağlantılarının kullanım öncelikleri değiştirilebildiği için, anlık olarak ağ arayüzlerinin çeşitli parametrelerinin izlenmesi ve öncelik parametrelerinin değiştirilebilmesi mümkündür. Örneğin kablosuz bağlantının kayıp oranın arttığının tespit edilmesi durumunda, performansın düşmemesi için ilgili ağ arayüzünün önceliğinin düşürülmesi mümkündür. Benzer şekilde önceden tanımlanmış bir politika ile ağ arayüzlerinin önceliklerinin, hedef port bazında tanımlanması mümkündür.

Referanslar

[1] Valdovinos, I. A., & Díaz, J. A. P. (2009). TCP Extension to Send Traffic Simultaneously through Multiple Heterogeneous Network Interfaces. *2009 Mexican International Conference on Computer Science*, *3*, 89–94. doi:10.1109/ENC.2009.28

[2] Dreibholz, T.; Rathgeb, E.P.; Rüngeler, I.; Seggelmann, R.; Tüxen, M.; Stewart, R.R., "Stream control transmission protocol: Past, current, and future standardization activities," *Communications Magazine, IEEE* , vol.49, no.4, pp.82,88, April 2011

[3] Natarajan, P.; Baker, F.; Amer, P.D.; Leighton, J.T., "SCTP: What, Why, and How," *Internet Computing, IEEE* , vol.13, no.5, pp.81,85, Sept.-Oct. 2009 doi: 10.1109/MIC.2009.114

[4] Atiquzzaman, M. (2003). SCTP: state of the art in research, products, and technical challenges. *2002 14th International Conference on Ion Implantation Technology Proceedings (IEEE Cat. No.02EX505)*, 85–91. doi:10.1109/CCW.2003.1240794

[5] Multipath TCP (mptcp) https://datatracker.ietf.org/wg/mptcp/charter/ (Şubat 2013)

[6] Ribeiro, E.P.; Leung, V. C M, "Minimum delay path selection in multi-homed systems with path asymmetry," *Communications Letters, IEEE* , vol.10, no.3, pp.135,137, Mar 2006 doi: 10.1109/LCOMM.2006.1603362

[7] Comparison of mobile Internet standards https://en.wikipedia.org/wiki/Template:Comparison_of_mobile_Internet_standards (Mart 2013)

[8] Address Allocation for Private Internets https://www.ietf.org/rfc/rfc1918.txt (Mart 2013)

[9] Traditional IP Network Address Translator (Traditional NAT) https://www.ietf.org/rfc/rfc3022.txt (Mart 2013)

[10] FreeBSD® is an advanced operating system. http://www.freebsd.org (Ocak 2013)

[11] FreeBSD 8.0-RELEASE Releases Notes http://www.freebsd.org/releases/8.0R/relnotes-detailed.html (Şubat 2013)

[12] FreeBSD setfib man sayfası http://www.freebsd.org/cgi/man.cgi?query=setfib&apropos=0&sektion=0&manpath=FreeBSD+8.0-RELEASE&arch=default&format=html (Ocak 2013)
[13] NewTCP, http://caia.swin.edu.au/urp/newtcp/ (Ocak 2013)
[14] NewTCP project tools http://caia.swin.edu.au/urp/newtcp/tools.html (Ocak 2013)
[15] FreeBSD 9.0-RELEASE Releases Notes http://www.freebsd.org/releases/9.0R/relnotes.html (Şubat 2013)
[16] HHOOK(9) man sayfası http://www.freebsd.org/cgi/man.cgi?query=hhook&sektion=9&manpath=FreeBSD+9.0-RELEASE (Ocak 2013)
[17] KHELP(9) man sayfası http://www.freebsd.org/cgi/man.cgi?query=khelp&sektion=9&manpath=FreeBSD+9.0-RELEASE (Ocak 2013)
[18] Connect(2) man page http://www.freebsd.org/cgi/man.cgi?query=connect&apropos=0&sektion=0&manpath=FreeBSD+9.1-RELEASE&arch=default&format=html (Ocak 2013)
[19] Sysctl(8) man page http://www.freebsd.org/cgi/man.cgi?query=sysctl&apropos=0&sektion=0&manpath=FreeBSD+9.1-RELEASE&arch=default&format=html (Ocak 2013)