

# İZMİR YÜKSEK TEKNOLOJİ ENSTİTÜSÜ'NÜN BİLGİSAYAR AĞI İÇİN BİR SERVİS KALİTESİ UYGULAMASI

Gökhan Çağrı ve Tolga Ayav  
Bilgisayar Mühendisliği Bölümü  
İzmir Yüksek Teknoloji Enstitüsü, İzmir  
{gcagrici, tayav}@arf.iyte.edu.tr

## Özetçe

Günümüzün gelişen bilgisayar ağları ve bu ağlarda oluşan trafik tiplerindeki çeşitlilik, **Servis Kalitesi** için büyük gereksinim doğuracak nitelikte sıkışıklıklar meydana getirmektedir. Nitekim, Servis Kalitesi'nin ne olduğu ve sıkışıklık yaşayan bilgisayar ağlarına neler sunduğu konuları hala tam bir açıklık kazanmamıştır. Ağ sistemleri üreticileri ve üniversite gibi bilimsel kuruluşlar, Servis Kalitesi mekanizmaları konusunda çeşitli araştırmalar yapmaktadır. Bu bildiriye, Servis Kalitesi'nin anlamı genel hatlarıyla sunulacak ve İzmir Yüksek Teknoloji Enstitüsü'nün yerel bilgisayar ağı için geliştirilen bir Servis Kalitesi uygulaması açıklanacaktır.

## Giriş

Bilgisayar ağlarında oluşan trafik, büyük farklılıklar göstermekte ve her bir trafik tipi; bant genişliği, gecikme, gecikme sürelerindeki değişim ve bulunabilirlik faktörleri açısından kendisine özgü gereksinimlere sahip olmaktadır. İnternet'in kullanımının patlamasıyla, bu trafiklerin büyük bir kısmı IP tabanlı olmuştur. Tek bir transfer protokolünün kullanılması, hem gerekli olan donanımın daha az karmaşık olması hem de işletim bedellerinin düşük olması bakımından yararlı olmaktadır. Bu gereksinimi karşılamak için de, bağlantısız (transfer öncesi bağlantı kurmayan) bir protokol olan IP seçilmiştir ve IP paketleri hedeflerine varmak için belirli bir yol izlemek yerine her biri kendi yollarını diğerlerinden bağımsızca belirlemektedir. Bütün bunlar, ideal bir bilgisayar ağı için Servis Kalitesi açısından belirsizlikler getirmektedir. IP protokolü dizayn edildiği zamanlarda, bir paketin kaynak noktadan alınıp hedef noktaya ulaştırılmasında zaman faktörüne dikkat edilmemiştir; ama bugünün bilgisayar ağlarında IP, değişik uygulama tiplerinin paketlerini ulaştırma görevini üstlenmiştir. Bu uygulamalardan bir çoğu düşük gecikme miktarları talep etmektedir. Aksi takdirde son kullanıcıya verilen hizmetin kalitesi belirli ölçüde etkilenmekte ve hatta bazı durumlarda uygulamanın görevini yerine getirememesiyle sonuçlanmaktadır.

İnternet bağlantılarının sıkışmasını önlemek amacıyla mevcut bant genişliğinin artırılması, akla gelen ilk çözümdür. Ancak sorun, basit bir kapasite sorunundan daha ötedir. Durum, trafiğin hacimsel olarak artışının yanında yapısal olarak değişmesidir. IP tabanlı bir çok uygulamadan bir çok yeni tip trafik ve bu trafik tipleri için de oldukça değişken yapıda operasyon gereksinimleri vardır.

Temel olarak Servis Kalitesi, kullanıcılara belli veri akışları için daha iyi servis verebilmektedir. Bu işlem, hedef veri akışının önceliğinin artırılmasıyla veya diğer veri akışının önceliğinin kısıtlanmasıyla mümkün olmaktadır. Sıkışıklığın kontrol edilmesinde kullanılan araç-gereçler, bir veri akışının önceliğinin artırılmasında kuyrukla ve bu kuyruklara farklı şekillerde servis sağlama yolunu denemişlerdir. Sıkışıklığın

gidirilmesinde kullanılan kuyruk yönetim aracı, önceliği artırmak için daha düşük önceliği olan veri akışlarını durdurmuştur. Planlama ve şekillendirme tekniği, bir veri akışına öncelik sağlarken diğer veri akışlarının birim zamanda ürettikleri paket sayısını kısıtlamaktadır. Hat tesirli araç-gereçler, küçük veri akışlarına öncelik sağlamak için büyük veri akışlarına bir sınır getirirler. Servis Kalitesi, sıkışıklığı doğuran bir çok problemi çözmekte yardımcı olabilir fakat bir çok zaman, sağlanan bant genişliğini aşan büyük miktarda trafik oluşmaktadır. Bu durumlarda, Servis Kalitesi, bu yarayı iyileştirmekte kullanılan bir sargıdan öteye gidememektedir.

Bu bildirinın konusu olan İzmir Yüksek Teknoloji Enstitüsü'nün yerel ağı için geliştirilen Servis Kalitesi uygulaması, sınırlı bant genişliğinin farklı kullanıcılar tarafından farklı amaçlar için kullanılmasından dolayı oluşan sıkışıklıkları önlemek amacıyla gerçekleştirilmiştir. İlerleyen kısımlarda ilk olarak Servis Kalitesi hakkında genel bir bilgi verilecek ve sonra sözkonusu uygulama daha detaylı olarak açıklanacaktır.

## Bilgisayar Ağlarının Servis Kalitesi

Bilgisayar ağları makineleri birbirine ağ bağdaştırıcısı, router, switch, hub vs. gibi değişik bir çok aygıtla bağlanmaktadır. Bu aygıtların her biri ağ ara yüzlerine sahip olmaktadır. Her bir ara yüz, trafiği sınırlı bir oranda alabilmekte ve gönderebilmektedir. Herhangi bir ara yüze bağlı trafik, o ara yüzün desteklediği oranı aştığında sıkışma meydana getirir. Ağ aygıtları bu durumun üstesinden gelebilmek için, sıkışıklık ortadan kalkana değin, trafiği aygıtın içerisinde yer alan hafızalarında kuyruklamaktadırlar. Diğer durumlarda, sıkışıklığı giderebilmek için ağ aygıtı trafiği tümüyle kesebilir. Sonuç olarak görülmüştür ki, uygulamalar, değişen gecikmeler ve trafik kayıplarıyla yüz yüze kalmaktadırlar [1].

Sıkışıklık, yanıt zamanında büyük gecikmelere ya da bir uygulamanın kritik görevlerini ağ üzerinde yerine getirememesine neden olmaktadır. Servis Kalitesi, mevcut olan bant genişliğini öncelikle daha önemli uygulamaların kullanımına sunmakta, daha az önem taşıyanlara ise görevlerini yürütebilecekleri oranda daha küçük bant genişlikleri sunmaktadır.

Servis Kalitesi yönetimine ayrılacak emek ve paranın, bant genişliğinin artırılmasında gerekli olan emek ve paraya göre daha büyük miktarlarda olduğu iddia edilmektedir. Ancak, bant genişliği ihtiyacının, ayrılan bütçenin büyüme hızına göre daha çok arttığı da bir gerçektir. Servis Kalitesi ayrıca ağın çökmesi ihtimalini de minimuma indirmektedir.

Her ne kadar bant genişliğinin artırılması akla gelen ilk çözüm gibi görünse de, yüksek bant genişliklerinde dahi farklı bir çok kaynaktan gelen ve farklı bir çok hedefe yönlendirilmiş değişen hız ve miktarlardaki paketlerle ilgilenen cihazlar

(router, switch, vs.) sıkışıklık yaşamaktan kaçınmamaktadırlar. Bu sıkışıklığın tümüyle ortadan kaldırılabilmesi için, karşılıklı olarak giren paketlerin tümü, çıkanlardan daha az olacak şekilde bir tasarım gerekmektedir, ancak bütün veri akışlarının tahmin edilebilir veya statik olmasını gerektiren bu çözüm hiç pratik değildir. Açıkça görülüyorki, sunucu ve istemci modellerin kullanıldığı günümüz bilgisayar ağlarında çalışan uygulamalar için bu tür bir yaklaşım getirilememektedir.

Sıkışıklık, kendi içinde üçe ayrılabilir:

**1. Port Düzeyinde Sıkışıklık.** Giren çeşitli veri akışlarının aynı çıkış portu için mücadele verdikleri sıkışıklık tipidir. Giriş yapan veri akışlarının toplamı, çıkış portlarının kapasitelerini aşıyorsa sıkışıklık meydana gelir. Eğer sıkışıklık sıkça oluyorsa ve/veya uzun zaman sürüyorsa, uygulamanın yanıt verme süresi de etkilenecektir.

**2. Aradaki Cihazlarda Meydana Gelen Sıkışıklık.** Ara cihazın destekleyebildiği bant genişliği, tüm giriş yapan veri akışlarının toplamından düşükse meydana gelir. Eğer sıkışıklık sıkça oluyorsa ve/veya uzun zaman sürüyorsa, cihaz bu durumu düzeltmeye çalışırken uygulamanın yanıt verme süresi de etkilenecektir.

**3. Ağdaki Sıkışıklık.** Kaynak ve hedef makineler arasında görev alan bir ya da daha çok ağ aygıtında veya bağlantılarda (linklerde) meydana gelen sıkışıklık tipidir. Bu sıkışıklık, diğer iki tip sıkışıklığın birleşiminden meydana gelmekte ve uçtan uca performansı düşürmektedir.

Sıkışıklığın çözülmesi ve etkili bir şekilde kontrolünün sağlanması için, bütün ağda geçerli olacak şekilde sıkışıklığı belirleyebilen ve tepki verebilen bir yönetim planlaması yapılmalıdır. Bunu sağlayabilen kontrolün adı da **Trafik Kontrolü**'dür.

Daha önce de belirtildiği gibi, Servis Kalitesi, uygulamalara belirlenmiş düzeylerde servisler sağlayarak performans ihtiyaçlarını karşılayan bir bilgisayar ağı yeterlidir. Servis Kalitesi, kaynak paylaşılmasını yapabilmek için değişik trafik tiplerini birbirinden ayırabilme özelliğine sahiptir. Ölçüt olarak kullanılan parametreler ise: bant genişliği, gecikme, gecikmelerdeki değişim ve paket kayıplarıdır. Yararı ise; bir sıkışıklık ortamında, kullanılmadığı bilgisayar ağlarına oranla kullanıldıklarında trafiğin daha iyi kontrol edilebilmesidir.

Servis Kalitesi, bant genişliği yaratmaz. Olmayan bir şeyi yaratamayacağı görüldükten sonra, Servis Kalitesinin işlevini anlayabilmek için ilk başlangıç noktası bant genişliğinin bulunabilirliğidir. Servis Kalitesi, bant genişliğini uygulamaların beklentilerine ve ağ yönetim ayarlarına göre paylaşırken hiçbir şekilde kesinlik veremez. Diğer taraftan, garantili servis mekanizmalarının uygulanması için her bir veri akışına ayrı ayrı kaynak ayrılmalıdır.

İki temel Servis Kalitesi tipi mevcuttur:

**1. Kaynak Rezervasyonu (Bütünleşik Servisler).** Ağ kaynakları, bant genişliği kontrol politikasına bağlı olarak, uygulamaların Servis Kalitesi isteklerine göre ayrılmaktadır.

**2. Önceliklendirme (Farklaştırılmış Servisler).** Ağ trafiği, bant genişliği politikasına bağlı olarak sınıflandırılmaktadır. Daha çok gereksinimi olan uygulamalara, bu sınıflandırma metoduyla öncelikli haklar tanınmaktadır. *Diffserv* tekniği bu servisi sağlamaktadır.

## DIFFSERV

Diffserv, yakın zamana kadar standartlaşması için az çaba gösterilmiş olmasına rağmen, yıllarca kullanımda olan bir 3. katman Servis Kalitesi mekanizmasıdır. Diffserv, 'diffserv codepoint' (DSCP) olarak tanımlanan, IP paketlerinin 3. katman başlığında bir alan belirler. Bu tür bir diffserv ağında trafiği oluşturan her kullanıcı makinesi ve router, gönderilen paketleri uygun DSCP değeri ile işaretlemektedirler. Bizzat bu diffserv ağı içinde yer alan router'lar ise, bu DSCP değerlerinden yararlanarak paketleri sınıflandırmakta ve uygun kuyruklama ve zaman programlamasını gerçekleştirmektedir [1].

Diffserv yaklaşımının avantajı, bir çok davranışsal yaklaşımın, ağ üzerindeki elemanlarda konfigüre edilebilmesini sağlamasıdır. Bu yaklaşım seti istenildiği ölçüde karmaşık olabilir ayrıca bu şekilde ağ elemanları yerine, paketlerin ne şekilde servis alacağı bilgisi bizzat paketlerin kendilerinde saklanmış olur ve böylece ağ elemanlarında yer alan yol izleme tabloları küçültülmüş olduğundan yönetilmeleri ve takip edilmeleri de basitleştirilmiş olur. Diffserv yaklaşımının dezavantajı ise tüm diğer farklılaştırılmış servislerde olduğu gibi bunda da kesin ve hızlıca geçerliliği sağlanan garantilerin yerine istatistiksel tercihleri sunmasıdır [2].

## DIFFSERV Uygulaması: CBQ'nun Linux'a Uyarlanması

Sınıf Tabanlı Kuyruklama (CBQ), Diffserv mekanizmasını tanıyan bir ağ elemanının trafiği yönetmesini sağlamak için kullanılır. CBQ, IP bilgisayar ağlarında Servis Kalitesi ve bant genişliği yönetimini sağlayan kaynağı açık ve anonim bir trafik sınıflama ve kuyruklama teknolojisidir. CBQ, kullanıcı trafiğinin 3. ve 7. katmanları arasındaki bilgilere dayalı parametrelerden yararlanarak sınıflandırma yapabilmekte ve servis düzeylerini tanımlayabilmektedir. Örneğin, trafik; IP adres aralığına, makine adresine, kullanılan protokole, port'a, uygulamaya veya bu parametrelerin değişik kombinasyonlarına göre sınıflandırılabilir [3]. Kuyruklama kullanılarak, verinin gönderilme şekli belirlenir. Dikkat çekmek gerekirse, gelen trafiğin tam aksine sadece gönderilen trafik biçimlendirilebilmektedir. İnternet'in kullanım şekline bakıldığında, elbetteki, kullanıcıların belirli bir IP adresine gönderdikleri bilginin doğrudan kontrolü yapılamaz. Ancak, İnternet büyük oranda TCP/IP mimarisine dayalıdır ve TCP/IP çok yararlı olabilecek bazı özelliklere sahiptir. Aradaki ağ ya da ağların kapasitesiyle ilgili hiçbir bilgisi olmayan TCP/IP, iki bilgisayar arasındaki herhangi bir veri transferini gittikçe daha hızlı gerçekleştirmeyi denemekte ve paket kaybı başlayınca kadar bu tutumunu sürdürmektedir. Paket kaybı devam ettikçe de hızı gittikçe düşürmeye çalışarak optimum seviyeyi bulmayı deneyecektir. Sınıfsız kuyruklama mekanizmaları, gelen veriyi kabul ettikten sonra sıralamayı yeniden yaparlar, geciktirirler veya düşürürler. Bu mekanizmalar, bütün bir ara yüzün trafiğini alt ayrımlar yaratmaksızın biçimlendirmede kullanılır. Bu kuyruklama mekanizmaları şunlardır:

- Pfifo\_fast
- Token Bucket Filtler
- Stochastic Fairness Queueing

Bazı kuyruklama mekanizmaları ise 'sınıf' adı verilen diğer kuyruklama mekanizmalarını içerebilmektedir. Bir sınıfın,

diğer bir kuyruk mekanizmasının içinde yer almasından başka sade bir kuyruk mekanizmasından farkı yoktur. Sınıfsal kuyruk mekanizmaları, deęişik muamele gerektiren tipte bir çok trafik içeren ağlar için çok yararlıdır. Sınıf Tabanlı Kuyruklama bu tür mekanizmalardan sadece birisidir.

CBQ, gerçek bant genişliğini, konfigüre edilen orana düşürünceye dek hattı boş bir durumda tutarak çalışır. Bunu yapmak için de, ortalama paketlerin arasında geçmesi gereken zamanı ölçer [4]. 10 Mbit/s lik bir bağlantıyı 1 Mbit/s lik düzeye getirebilmek için, hattın, zamanın %90'ında boş tutulması gerekmektedir. Operasyon anında, efektif boş bırakılma süresi, geçmekte olan paketlerin daha önceliklere göre ekspanansiyel olarak daha çok önem taşıdığını varsayan ekspanansiyel ağırlıklı ilerleme ortalaması (Exponential Weighted Moving Average: EWMA) teknięi kullanılarak belirlenir. Hesaplanan boş kalma süresi EWMA deęerinden çıkarılarak ortalama boş bırakılma süresi (avgidle) bulunur. İdeal yüklü bir bağlantıda bu deęer sıfırdır, yani paketler hep hesaplanan aralıklarla varmaktadır. Bütün bu sınıflar başka diđer sınıfları içerirler (deęiştirilmedięi takdirde her biri için kuyruklama mekanizması pfi<sub>o</sub>\_fast seçilmiştir). Bir paketin kuyruğa sokulması esnasında, hangi sınıfa gireceğini belirlemek amacıyla ana noktadan (root) başlayarak deęişik metodlar uygulanır. Ağ aygıtından gönderilmesi esnasında kuyruktan çıkarılma sırasında ise hangi sınıfların gönderilmesine izin verileceğini CBQ, paketleri içeren sınıfların her bir dönüşte şans elde ettikleri Weighted Round Robin (WRR) işlevini kullanarak karar verir. WRR işlevi, en yüksek öncelięi taşıyan sınıflara gönderecekleri paket olup olmadığını sorduktan sonra eđer paketleri kalmamışsa daha düşük öncelik taşıyanlarla devam eder. Bu sayede, daha öncelikli olan sınıflar paketlerini göndermeden diđerlerine fırsat tanınmaz. Sınıflar 'sınırlanmış' ('bounded') olarak tanımlanmadığı sürece kendisiyle aynı hiyerarşik düzeyde olan diđer sınıflardan ödünç bant genişlięi alabilir. Herhangi bir sınıf, kullanmadığı zaman dahi sahip olduęu bant genişliğini ödünç olarak vermek istemiyorsa 'yalıtılmış' ('isolated') olarak tanımlanır [5].

## CBQ'ya Bir Alternatif: Hierarchical Token Bucket (HTB)

HTB, Linux'ta CBQ'nun yerini doldurabilecek nitelikte daha kolay anlaşılabilir ve sezgisel olarak kullanılabilir bir kuyruk bir mekanizmasıdır. CBQ gibi HTB de sadece dışarı giden trafięi biçimlendirmekte ve fiziksel bir hattı çok deęişik tiplerdeki trafięi taşıyabilecek daha yavaş hatlar halinde simüle edebilmektedir. Kullanıcı, hem bu fiziksel hattın ne şekilde simüle edilmiş hatlar arasında paylaşılacağını hem de bu hatlar üzerinde hangi trafiklerin iletileceğini bildirmekle yükümlüdür.

Martin Devera, gerek CBQ'nun gereęinden fazla karmaşık olması, gerekse belli bazı durumlarda optimum şekilde çalışmaması yüzünden HTB adını verdięi bu hiyerarşik yaklaşımı getirdi. HTB, deęişik amaçlar için bölünmesi gereken sabit miktardaki bir bant genişliğini garantili olarak bölerek, her birinin borç alınıp alınmayacağı bilgisini de eklemiştir. HTB nin CBQ dan önemli bir farkı, boş tutulan zamanlara göre biçimlendirme yapmamasıdır. Bunun yerine az sayıda kolay anlaşılır parametresi olan TBF sınıflandırma mekanizmasını içermektedir [4]. Bütün bu Servis Kalitesi protokolleri ve algoritmaları birbirlerinin yerine kullanılmasından ziyade beraber farklı amaçlar için bütünleşik olarak tercih edilebilirler [6].

## 'tc' Komutu

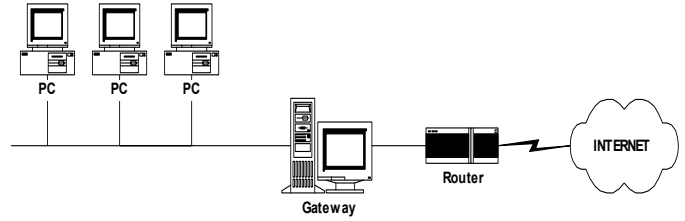
Son zamanlardaki Linux çekirdeklerinde, diffserv mekanizmasının uygulanmasını saęlayan komutun adı 'tc' (traffic control) dir.

'tc' komutu başlıca kuyruk mekanizmalarıyla (qdiscs), sınıflarla (classes) ve filtreleme mekanizmalarıyla (filters) uğraşmaktadır. Bütün bunlar sisteme sırayla **tc qdisc...**, **tc class...** ve **tc filter...** şeklinde bildirilmektedir.

## HTB Kullanılarak Servis Kalitesi Yönetim Aracı

İzmir Yüksek Teknoloji Enstitüsü'nde bir Servis Kalitesi uygulaması geliştirme fikri, bir çok kullanıcının İnternet bağlantılarının verimsizliğinden ötürü şikayetlerinden ortaya çıktı. Detaylı bir trafik analizi yapıldıktan sonra görüldüğü, mevcut olan bant genişliğinin bir kısmı, Napster, Morpheus vs. gibi dosya paylaşımı saęlayan uygulamalar tarafından kullanılmaktaydı. Bu yüzden bir bant genişlięi yönetim aracına ihtiyaç duyulmuştur.

Uygulamamız bu bölümde kısaca anlatılacaktır.



Şekil-1: Bilgisayar Ağının Genel Görünümü

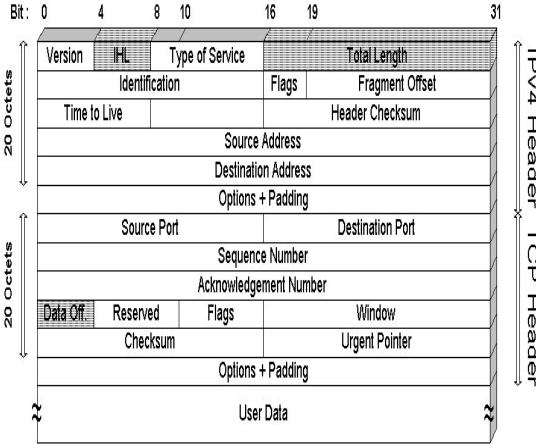
Kampüsün bilgisayar ağ mimarisi kaba hatlarıyla Şekil-1 de gösterilmiştir. Router ve kampüs yerel ağı arasında üzerinde Linux işletim sistemi yer alan ve Servis Kalitesi yönetim görevini üstlenen bir ağ geçidi bulunmaktadır. Yerel ağ üzerindeki her bir makineye uygulanacak politikanın belirlenmesinde sınıf ve filtre mekanizmasının belirtilmesi için makine başına iki komut satırı girilmelidir (tc class... ve tc filter...). Uygulanan kuyruk mekanizması içinse varsayılan deęer olan pfi<sub>o</sub>\_fast kullanıldığı için ayrıca bildirilmesine gerek görülmemiştir. Ancak yüzlerce makine için bu komutların elle girilmesinin çok elverişsiz olması yüzünden otomatik bir komut giriciye gereksinim doğmuştur. Ayrıca kullanıcı hızlarının hesaplanması, karar verme mekanizması (henüz uyarlanmadı) ve kullanıcı arayüzü görevlerini gerçekleştirmek amacıyla da tüm bunları içine alabilecek komple bir sistem geliştirilmesi düşünölmüştür.

Ağ geçidinin üzerine düşen görevler şu şekilde özetlenebilir:

- Sistem yöneticisine karar verme aşamasında yardımcı olacak kullanıcı download ve upload hızlarının hesaplanarak bildirilmesi ve 'tc' komutlarının çalıştırılmasının hemen ardından deęişimlerin gözler önüne serilmesi,
- Sınıfların yapısının ve onlara uygulanan politikaların etkili bir şekilde ve kalıcı olarak saklanabileceęi basit bir veritabanının barındırılması,
- Sisteme erişilebilirlięi, izlenebilirlięi ve yönetilebilirlięi saęlayan kolay anlaşılır bir ara yüzün saęlanması.

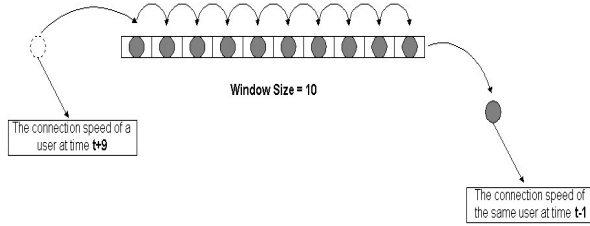
Uygulama sırasında, ağ geçidine uğrayan her paket incelenerek kaynak ve hedef IP adresleri elde edilir. Daha sonra, paketdeki verinin büyüklüğü bit cinsinden bulunarak LAN'daki her IP adresi için download ve upload hızları hesaplanır. Hesaplanan hız değerlerindeki ani değişimleri engellemek içinse bir alçak geçiren filtre kullanılmıştır.

Belirli bir IP adresinin upload ve download hızlarının hesaplanmasında IP paketinin Şekil-2'de gösterilen gölgelendirilmiş alanları kullanılır.



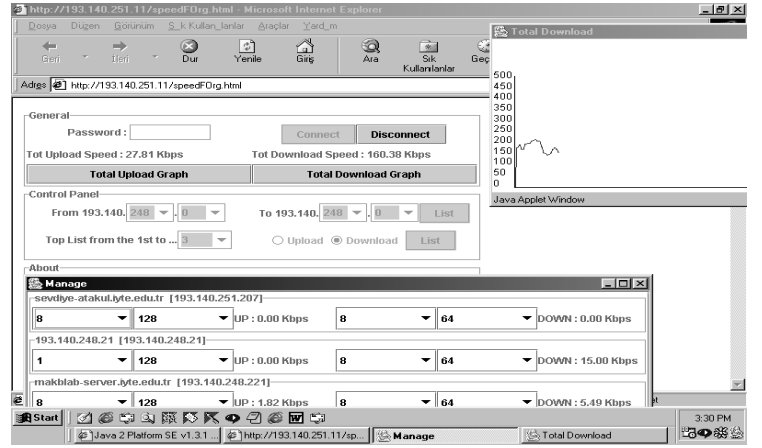
Şekil-2: Kullanıcı verisinin boyutunu hesaplamak amacıyla kullanılan alanları gölgelendirilmiş bir IP paketi

Yukarıda bahsedilen alçak geçiren filtrenin işlevi şu şekilde açıklanabilir. Belirli bir kullanıcının download ve upload hızları hesaplandıktan sonra, bu değer saniyede bir pencerelerden birine kaydedilir ve tekrar saniyede bir bu değerler toplanarak ortalaması alınır (pencere boyutu çok kritiktir). Uygulamamızda, pencere boyutunu 10 olarak seçtiğimiz bu filtrenin çalışması Şekil-3'te açıklanmıştır.

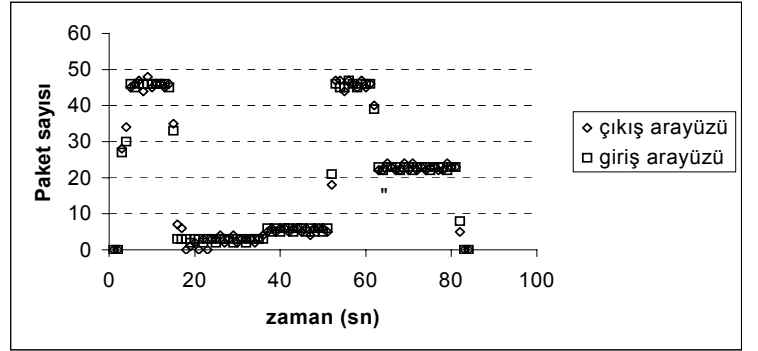


Şekil-3: Uygulamada kullanılan filtre

Her bir makine için bit/s cinsinden hesaplanan hız değerleri, sonucu bir program tarafından istemciye gönderilir. İstemci kısım JAVA applet olarak, sunucu kısım ise JAVA uygulaması olarak dizayn edilmiştir. Bu hız değerlerinin istemci tarafından alınması sırasında ise JAVA Remote Method Invocation (RMI) mimarisi kullanılmıştır. Bu appletin kullanıcı ara yüzü ise şekil-4'te gösterilmektedir.



Şekil-4: Uygulamanın kullanıcı arayüzü



Şekil-5: Bant genişliğinin değişimi sırasında giriş-çıkış paket sayıları

Bu şekilde sözkonusu mekanizma ağ yöneticisine toplam Internet upload ve download hızlarını ve yerel ağ üzerindeki her bilgisayarın ayrı ayrı upload ve download hızlarını takip edebilme ve bunun yanısıra her bilgisayarın kullanabileceği en az ve en fazla upload ve download bant genişliği sınırlarını belirleyebilme imkanı vermektedir. Kullanıcılara (Bilgisayarlara) bant genişliği atanması el ile ağ yöneticisi tarafından yapılmakla birlikte, kullanıcıların trafik tiplerinin incelenerek toplam bant genişliğini optimum kullanacak şekilde atamaların otomatik olarak yapılması yakın gelecekte çok önemli bir çalışma konusu olarak durmaktadır.

Projenin test kısmında ilk önce değişik bant genişliği konfigürasyonlarında çeşitli bağlantılar kurularak performans incelenmiş, kampüsteki kullanıcıların düşünceleri alınmıştır. Ağdaki her bilgisayarın kullanabileceği bant genişliğine belli bir sınırlama getirildiğinde genel performansta bir artış görülmüştür. Kullanıcıların da genel görüşü olumlu yönde olmuştur. Diğer bir deneme ise ağ geçidinin performansını ölçmek için yapılmıştır, öyle ki bant genişliğine bir sınırlama getirildiğinde ağ geçidinin bir ağ ara yüzünden giren paket sayısı ile diğer ara yüzünden çıkan paket sayıları arasındaki dengesizliğin büyüklüğü önemlidir ki bu da kuyruklarda birikmeye neden olan bir faktördür. Bunun için bilgisayarlardan birinden Internet üzerinde bir sunucuya ftp bağlantısı yapılmış ve bu bağlantıda yönlendiriciden ağ geçidine gelen ve ağ geçidinden bilgisayara giden paket sayıları ölçülmekteyken bu bilgisayarın kullanabileceği bant genişliği 17. saniyede 256 kbits/s'den 16 kbits/s'ye, 36. saniyede 32 kbits/s'ye, 52. saniyede 256 kbits/s'ye ve son olarak 63. saniyede 128 kbits/s'ye değiştirilmiştir. Ölçüm sonucu şekil-5'te gösterilmektedir. Görüldüğü gibi bant genişliğinin değişiminden sonra giriş ve çıkış paket sayıları

arasında ilk başta oluşan fark bir kaç saniye içinde ortadan kalkmaktadır ki bu TCP'nin pencere boyutunu dolduracak kadar paketi gönderdikten sonra onay bekleme özelliğinden kaynaklanmaktadır. Sonuç olarak, kuyruklarda birikmeye neden olacak ve sistemin performansını düşürecek nitelikte farklar oluşmadığı görülmüştür.

## Sonuç

Gün geçtikçe büyüyen bilgisayar ağları, masaüstü ve ağ elemanlarındaki teknolojik ilerlemeler ve multimedia ile diğer interaktif teknolojilerin kullanımındaki artış, LAN/WAN sınırlarındaki router'lara ve bu ağların çekirdek kısımlarında kullanılan diğer ağ elemanlarına yüklenmektedirler. Farklılaştırılmış servisler gibi Servis Kalite mekanizmaları kullanılmadığı takdirde tüm ağın performansı olumsuz ölçüde etkilenmektedir. Günümüzün yerel ağlarına Servis Kalite mekanizmalarını uygulamak genellikle pahalı ve zor olmakla beraber yukarıda uyarlaması anlatılan kolay ve basit yöntemler de mevcuttur. Bu farklılaştırılmış servis mekanizmaları sayesinde, ağın çeşitli yerlerinde oluşabilecek darboğazlar etkili bir şekilde engellenebilmektedir.

Bu çalışmada, farklılaştırılmış servis mekanizmalarından biri olan HTB, İzmir Yüksek Teknoloji Enstitüsü'nün yerel ağı için uygulanmış ve Internet bağlantısına ilişkin performansta iyileşmeler gözlenmiştir. Bu bilgi sisteminin yönetimi el ile yapıldığı için ağ yöneticisi, Internet kullanımını incelemeli ve gerekli bant genişliği sınırlamalarını getirmelidir. Gelecekte ise sistemin kendi kendini yönetmesi sağlanabilir. Çeşitli kontrol mekanizmaları ile toplam bant genişliğinin belirli bir değerin altında kalmasını sağlayarak, değişik bağlantıların hızları, belirtilmiş politikaların sınırları dahilinde dinamik olarak değiştirilebildiğinde bu bilgi sistemi çok daha iyi sonuçlar üretecektir.

## Kaynakça

[1] **Microsoft Corporation**, August 1999. Quality of Service Technical White Paper.

[2] **Geoff Constable and Dave Price**, 22 March 2001. Options for Quality of Service Mechanisms, University of Wales, Aberystwyth.

[3] **Ashley Stephenson**, December 1999. QoS: The IP Solution White Paper, Vice President and CTO, Lucent Technologies, WAN Systems Group;.

[4] **Bert Hubert, Gregory Maxwell, Remco van Mook, Martijn van Oosterhout, Paul B. Schroeder and Jasper Spaans**, 06 December 2001. Linux Advanced Routing & Traffic Control HOWTO.

[5] **Alexey N. Kuznetsov**, CBQ(8)

[6] **Philippe Levillain**, 7 February 2000. The OMNI family Quality of Service (QoS) Framework Technology White Paper.