



**T.C.
TRAKYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
Bilgisayar Mühendisliği Ana Bilim Dalı**

Mechanism For Quality-of-Service (QoS)

**Hazırlayan:
Temel YAVUZ**

Servis Kalitesi - Quality of Service - QoS

İnternetin yaygın kullanımında ve önemindeki artışın bir sonucu olarak, yeni nesil IP ağlardan sadece internet trafiğini değil aynı zamanda diğer arzu edilen servisleri de taşınması beklenmektedir. Buna rağmen, çoklu tahsis edilmiş ağların, tek bir ağın azaltılmış işlevsel maliyetinden gerçekten yararlanan bir operatörün yeni nesil ağların (Next Generation Networks), servis kalitesini sağladığından emin olması gerekir. QoS, tanımı oldukça belirsiz bir terim olmakla beraber iki ana kriterinin düşük gecikme değerlerine ihtiyaç duyması, düşük gecikme varyasyonu olduğunu gösterebilir.

QoS, bir ağın seçilen ağ trafiğine Frame Relay, ATM, Ethernet ve 802.1 ağlarını içeren çeşitli teknolojiler üzerinden en iyi servisi sağlayabilme kapasitesini ifade eder. Temel hedef, tahsis edilmiş bandwidth, kontrollü jitter ve gecikme ile iyileştirilmiş kayıp karakteristiklerini içeren önceliği sağlamaktır. Burada önemli olan şey bir ya da daha fazla trafik akışına öncelik verirken diğer akışları engellememektir.

Yüksek hızlı bir NGN, kabul edilebilir seviyedeki bir QoS'ı sağlamak için şu anahtar özelliklere sahiptir:

- Link kopmasından kaynaklanan herhangi bir kesinti kullanıcı tarafından ihmal edilebilecek kadar kısa olmalıdır.
- Ağdaki aşırı yüklenmeden kaynaklanan kalite bozulmalarından olabildiğince kaçınılmalıdır.

Bir ağda QoS'i sağlayabilmek için, bandwidth, gecikme süresi, jitter, paket kaybı ve paket gecikmesi gibi birçok parametre ele alınır.

Bir uygulama eğer tüm parametreler aynı öneme sahipse değerlendirilmeye alınamaz. Yeni nesil kablosuz uygulamalar, QoS ihtiyaçlarına göre 5'e ayrılır (Tablo 1).

Uygulama Sınıfları	Karakteristikler / Örnekler	QoS Ölçütleri
Gerçek Zamanlı	Sınırlı EndToEnd gecikme ve jitter'a ihtiyaç duyar. VoIP ve IP üzerinden Canlı video, video oyunları, PUsH-to-talk, tek yönlü ses ve video iletim uygulamaları gibi	End2End gecikme Gecikme varyasyonu (jitter) Paket kayıp seviyesi
Gerçek Zamanlı Olmayan	Yüksek bandwidthe ihtiyaç duyar. End2End gecikmeye daha az duyarlıdır ve jitter'a duyarlıdır. FTP, E-mail, WWW gibi.	Yük
İşlem Tabanlı	Yüksek güvenilirlikli kanallara ve daha düşük cevap verme zamanına ihtiyaç duyar. Çok bandwidth tüketmez. Tüm M-ticaret tabanlı uygulamalar.	Cevap zamanı Güvenlik Bozulma seviyesi
Mesaj Tabanlı	Başarılı teslim etmeye ve zaman zaman bu işlemin sınırlı bir sürede yapılmasına ihtiyaç duyar. Çok bandwidth tüketmez. MMS, haberlerin, hava, spor ve finansal bilgilerin güncelleme, anlık mesajlar.	Teslim başarı seviyesi Teslim zamanı
Konum Tabanlı	Bilgi transferi kullanıcının konumuna bağlıdır. Konum tabanlı tüm uygulamalar.	Konum Doğruluğu Cevap zamanı

Tablo1: Uygulamalar ve Qos Ölçütlerinin Sınıflandırılması

Bu sınıflandırmanın amacı End2End QoS metriklerini tanımlamak ve tanımlama işlemini basitleştirerek yeni uygulamalar için QoS ölçütlerini tanımlamaktır.

Yük			Cevap Verme Zamanı	
Gecikme	Paket Kaybı	Bandwidth H	Gecikme	Paket Kaybı
Gerçek Olmayan Zamanlı Uygulamalar (WWW, FTP)			İşlem Tabanlı Uygulamalar (M-Ticaret)	

Tablo 2: Anahtar QoS Ölçütlerinden Örnekler

Temel QoS Mimarisi

QoS uygulaması için temel mimarî 3 ana parçadan oluşmaktadır:

- Ağ elemanları arasında uçtan uca QoS düzenlemesi için QoS tanımlama ve işaretleme teknikleri.
- Tek bir ağ elemanı içindeki QoS
- Bir ağ boyunca end-to-end trafiği kontrol etmek ve yönetmek için QoS kural, yönetim ve hesaplama fonksiyonları.

Bir ağda QoS'i sağlamak için kullanılacak birçok QoS mekanizması vardır.

Bunlar:

- En iyi güç algoritmaları ile sağlam bir servis sağlayabilirler.
- En az yerleştirmeye, kaynağa gerekli olan bandwidthi maksimize edebilirler.
- Buffer dolduğunda routerlarda çakışma olursa ya da buffer işgali çok artarsa paketleri düşürebilirler.
- Çıkış bufferlarını N kuyruğa bölebilir ve bir zamanlayıcı yapabilirler.
- Farklı katmanlardaki IP akışlarını sınıflandırabilirler:

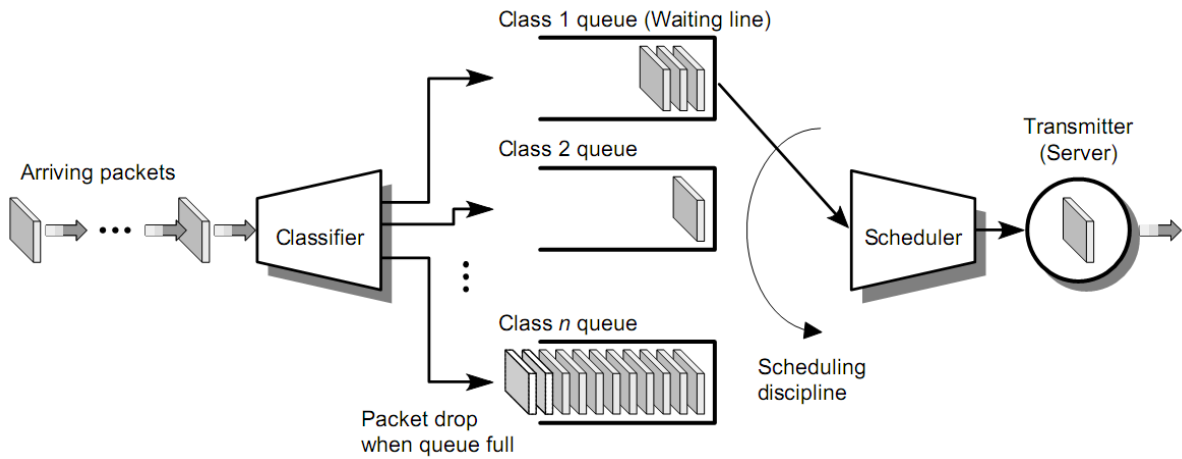
Her bir kuyruğa ağırlık tahsis edebilirler. QoS terimi kullanıcı perspektifinden belirli bir servis kalitesine erişmek için gerekli bağlantı parametrelerine kadar birçok anlama birlikte kullanılır.

Scheduling (Programlama)

1.a. Programlama Disiplini (Scheduling Disciplines)

Sıralama (queuing) modelleri, düşünölmüş gecikmeleri ve bloklama ihtimallerini, görevler/paketler içerisinde önce gelen önce gider ilkesinin (**FCFS-First Come First Serviced**) kullanıldığı varsayılarak değeriendirir. Ve bir görev dolu bir kuyruğa ulaşırsa bloklanır (Eğer bekleme kısmının kapasitesi sınırlı ise). Kuyruğun paket servislerinin sıralanmasına karar veren özelliğine **Programlama Disiplini** adı verilir (Ayrıca Servis Disiplini ya da Kuyruk Disiplini adını da alabilir). Kuyruğun sisteme gelen görevlerden hangisinin bloklanacağına ya da dolu bir kuyruk için hangi paketin düşürüleceğine karar veren özelliğine bloklama politikası, paket ayırma politikası veya **Düşürme Politikası** (Drop Policy) adı verilir. Son düşme işleminde temel kombinasyon FCFS 'dir. Daima kuyruğun başı servis edilir ve gerekli olduğunda da son ulaşan paket düşürölür.

Programlama paketin sıralama işlemlerindeki gecikmelerinde ve dolayısıyla da toplam gecikmeler üzerinde doğrudan etkiye sahiptir. Paketin hedefe hiçbir şekilde varıp varamayacağına **Düşürme** karar verir. FCFS paketler arasında hiçbir ayırım yapmaz. Paketler arasında ayırım yapmayan bir diğer yöntem ise FIRO (First-in-Random-out) yöntemidir. Ayrıca ihtiyaçlar, ağ tasarlamanları bazı paketler arasında ayırım yapmaya, daha karmaşık programlama disiplinleri ve **düşürme** (dropping) politikaları tasarlamaya zorlayabilir.



Şekil 1: Bir planlayıcının bileşenleri. Sınıflandırıcı (Classifier) gelen paketleri bir veya daha fazla kritere (öncelik, benzerlik) göre bekleyen farklı kuyrukların içine sıralar. Planlayıcı daha sonra paketleri planlama disiplinine göre servis içine yerleştirir. Bekleyen tüm kuyrukları tek bir sunucu servis eder.

Bazı ihtiyalar Őunları ierir:

- **Öncelik Verme**, farklı görev ve paketler iin farklı öncelikler tahsis edilebilir. Böylece gerekli paketler iin gecikme zamanı azaltılabilir.
- **Doğruluk**, böylece farklı akımlar iin (kaynak hedef iftine göre belirlenen) sistemin imkanlarına adil erişim sunulur.
- **Koruma**, öyle ki zarar verici (kendilerine ait olan orandan daha hızlı paket göndermek şeklinde) bazı akımlar diğerk akımlardan elde edilen performansı etkileyememelidir.

Öncelik verme ve doğruluk karşılıklı olarak bütünleyicidir. Doğruluk, eşit öncelikli adil hizmet alımına dayalı trafik akışı sağlar.

Doğruluk ve Koruma ilişkilidir, öyle ki doğruluğu sağlamak otomatik olarak korumayı sağlar. Onun hakkı olan alana yapılan zararlı akımları sonlandırır. Ancak tersinin doğru olmasına gerek yoktur. Örneğın akımlar bilgisayar ağına girişte kontrol altına alındığında, bir trafik şekli ilanını doğrulamaya zorlanırlar, ve birbirlerini korurlar, fakat onların kaynak paylaşımları doğru olmayabilir.

Paketler arasında ayırım yapmak isteyen sistemler iki bileşene ihtiyaç duyarlar (Şekil 1):

1.**Sınıflandırıcı**—Farklı paket tipleri iin bekleyen farklı kuyrukları düzenler. Farklı kuyruklara farklı paketleri sıralamanın kriteri Őunları ierir: öncelik, kaynak veya hedef ađ adresleri, uygulama port numarası.

2.**Planlayıcı**—Paketleri bekleyen kuyruklardan servis iin çağırır. Servise paketleri çağırmanın kuralları iin ayarlar Őunları ierir: (i) Öncelikle yüksek öncelikli kuyruktaki tüm paketlere servis hizmeti ver, daha sonra daha düşük öncelikliye git: (ii) kuyruklarda sıralı olarak bir kuyruktan bir veya daha fazla pakete hizmet ver (bekleyen tüm paketlere hizmet şart deđil), ve sonra bekleyen diğerk kuyruđa hizmete git, bunu bir döngü halinde devam ettir.

FCFS kuyrukta bulunan bütün paketleri ayırım yapmaksızın yerleřtirir. Öncelik fikri ile yüksek öncelikli paketler, sisteme gelenler üzerinde, kuyruğun baş kısmına yerleřtirilir. Böylece kuyrukta beklemekten kurtulurlar. Düşük öncelikli paket hala gönderilme aşamasındaysa paketler biraz bekleyebilir. Başkasına engel olmayan planlama devam eden düşük öncelikli paket iletiminin yüksek öncelikli paket iletiminde kesinti yapmamasıdır. Tersine engelli planlamada, yüksek öncelikli paket düşük öncelikli paket iletildiğinde ulaşırsa, düşük öncelikli paket servis dıřı kalır. Paket önceliđi temelde paket tiplerine göre belirlenebilir. Örneğın politikalar, yüksek öncelikli kullanıcı tipinin paket tipi iin günün belli bir zamanını ve düşük öncelikli iin ise başka bir zamanı belirleyebilir.

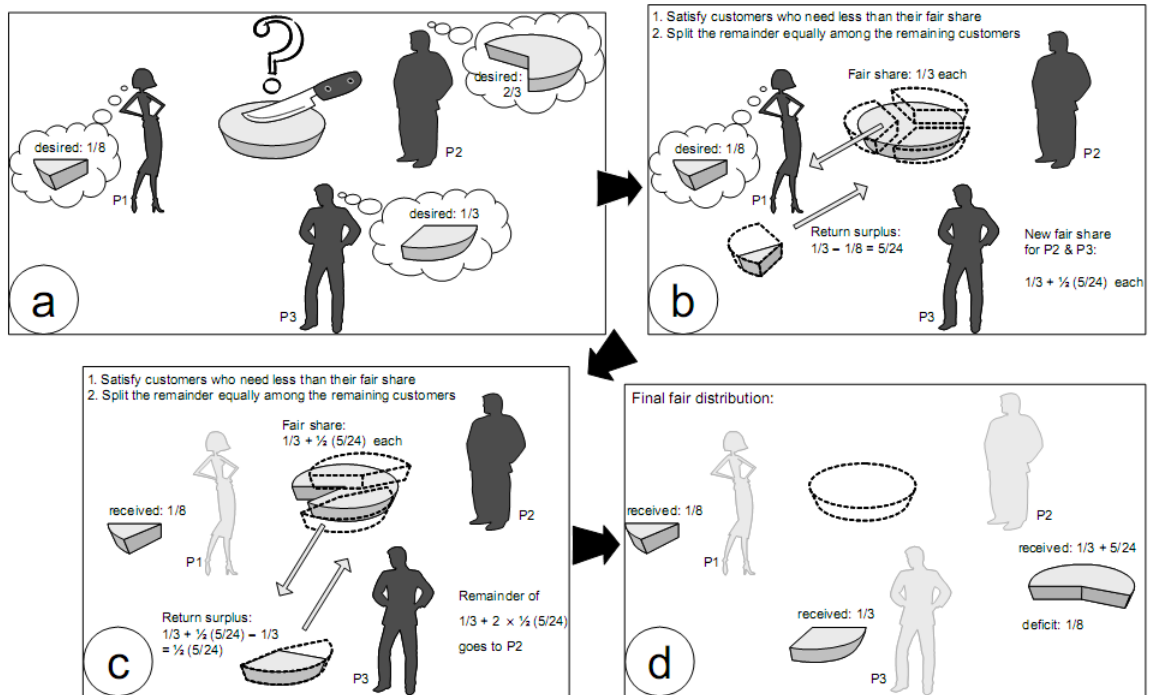
Öncelik planlayıcısı farklı sınıflara farklı performans karakterleri sağlarken hala eksiklikleri vardır. Örneğın doğruluk ve koruma ile ilgilenmez. Zararlı veya yanlış yüksek öncelikli kaynaklar iletişim hattını tutar ve diğerk kaynakları etkisizleřtirir. Sadece düşük öncelikli akışlar deđil ayrıca yanlış olan akışlarda da zarar görebilir. Farklı akışlar veya sınıf paketleri iindeki sıralı planlayıcılar servisi sıralar. En basit sıralama planlayıcısı servis iin her kuyruğun başında sırayla adlandırma yapar. Birinci paket sınıfı iletilir, ikinci paket sınıfı da bunu takip eder ve bir sınıftaki paketler iletilene dek bu işlem sürer. Bu işlem bütün paketler bitene dek durmadan devam eder. Eđer özel bir kuyruk ve bu türe ait belirli bir paket boş ise bu durumda iki tür seenek vardır:

1. Servisin kullanılmamış paraları ve bu belirli sınıf iin sunucuyu boş tutmak.
2. Farklı bir kuyruktan gelen paketlere izin vermek, eđer herhangi biri bu servisi kullanıyorsa.

Bir çalışma koruma planlayıcısı eğer paketler ileti için sıralanmış ise sunucunun boş kalmasına izin vermez. Bu tür planlayıcılar daha önce verilmiş bir paket arıyorlar fakat bulamıyorlar ise bir sonraki sınıfa geçerler. Kanalin durumunu kontrol etmenin bir yolu da zaman bölücü çoklamayı (TDM) veya frankans bölücü çoklamayı (FDM) görevlendirmektir. TDM/FDM her bir trafik akışı için ayrı kanal sağlar ve hiçbir paketi birbirine karıştırmaz, bundan dolayı birbirleri ile alakaları olamaz. TDM ve FDM çalışmayan koruyuculardır. İstatistiksel çoklayıcı çalışan koruyucudur.

1-b. Adil Sıralama (Fair Queuing)

Bir sistem düşünün, iletim ağı gibi, bütün kullanıcıların isteklerine cevap vermede yetersiz kaynağa sahip ve herkes kaynak üzerinde eşit haklara sahiptir. Fakat, aslında bazıları diğerlerinden daha az kaynak ister. Ancak kaynak bölünebilir mi? Max-Min adil paylaşma adı verilen bir paylaşma tekniği yaygın olarak kullanılır.



Şekil 2: Max-Min Adil Paylaşım Algoritmasının Çizimi

Sezgiyle, adil paylaşım ilk önce hak edenden daha az ihtiyaç duyan kullanıcıların taleplerini yerine getirir ve sonra eşit olarak kullanılmayan kaynakları “büyük” kullanıcılar arasında dağıtır (Şekil 1-2). Usulen, max-min adil paylaşım tahsisatı aşağıdaki gibi olur:

- Kaynaklar artan isteklere göre paylaşılırlar
- İhtiyacından daha geniş bir kaynak paylaşımı sağlayan kaynak yoktur.
- Kaynaklar yerine getirilmemiş istekler arasında eşit bir kaynak paylaşımı sağlar.

Bu resmi tanımlama sonraki işletimsel tanımlama ile eşleşir. s_1, s_2, \dots, s_n kaynak isteklerine sahip olan $1, 2, \dots, n$ 'e kadar kaynak dizisini göz önüne alalım. Genelleme yapmaksızın kaynak isteklerini sıralayalım, öyle ki $s_1 \leq s_2 \leq \dots \leq s_n$ olsun. Server C kapasitesine sahip olsun. Sonra, başlangıçta C/n kaynağını en küçük istek olan s_1 'e verelim. Bu **kaynak 1**'in istediğinden fazla olabilir, belki, işleme devam edelim. Kaynak istediğinden fazlasını alamayınca ve istek tatmin edici değilse, daha yüksek indisli bir başka kaynaktan daha fazla değilse, işlem sonlanır. Bu gibi bir tahsisat bir min-max adil tahsisat olarak adlandırılır. Çünkü bu talebi tamamen karşılanmamış bir kaynağın minimum paylaşımını maximize eder.

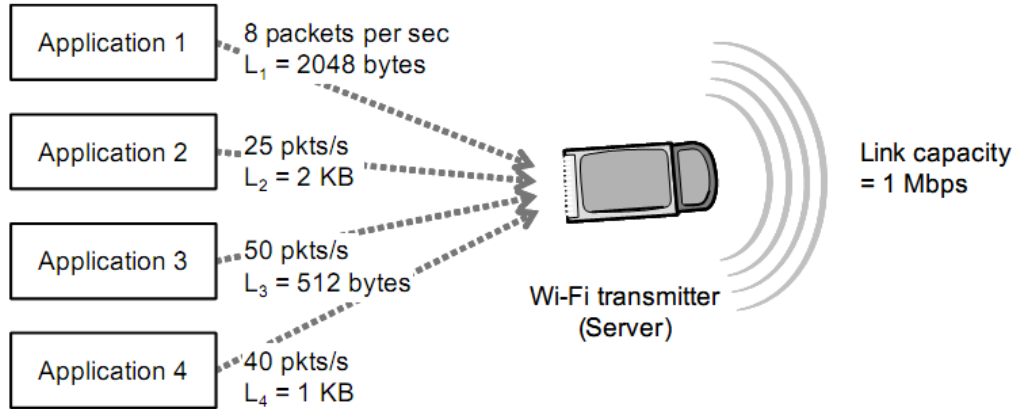
Örnek 1.1 Max Min Adil Paylaşım

Şekil 3 'teki sunucuyu (server) göz önüne alalım. Paketler eşit önceliğe sahip dört kaynak üzerinden ulaştırılıyor ve iletimde bir wireless bağlantısı kullanıyor. İhtiyaç duyulan toplam link kapasitesi:

$$\frac{8 \times 2048}{\text{Appl. A}} + \frac{25 \times 2048}{\text{Appl. B}} + \frac{50 \times 512}{\text{Appl. C}} + \frac{40 \times 1024}{\text{Appl. D}} = 134,144 \text{ bytes/sec} = 1,073,152 \text{ bits/sec}$$

Fakat ulaşılabilir link kapasitesi ise $C=1 \text{ Mbps}=1,000,000 \text{ bits/sec}$

Doğruluk kavramı sayesinde bütün kaynaklar eşit önemdedir. Her bir kaynak toplam kaynak kapasitesinin $\frac{1}{4}$ kadarını hak eder 250 Kbps .



Şekil 3: Wireless bağlantısı üzerinden dört kaynaktan (applications) paket iletimli bir sunucu örneği (Wi-Fi iletimi)

Bazı kaynaklar bu kadarına ihtiyaç duymaz ve artan kısım kalanlara arasında adil bir şekilde paylaşılır. Aşağıdaki tablo max-min adil tahsisat yöntemini göstermektedir.

Sources	Demands [bps]	Balances after 1 st round	Allocation #2 [bps]	Balances after 2 nd round	Allocation #3 (Final) [bps]	Final balances
Application 1	131,072 bps	+118,928 bps	131,072	0	131,072 bps	0
Application 2	409,600 bps	-159,600 bps	332,064	-77,536 bps	336,448 bps	-73,152 bps
Application 3	204,800 bps	+45,200 bps	204,800	0	204,800 bps	0
Application 4	327,680 bps	-77,680 bps	332,064	+4,384 bps	327,680 bps	0

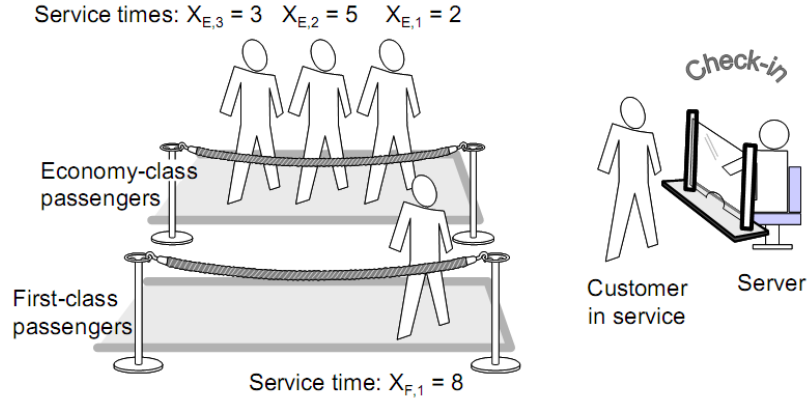
Şu ana kadar, kaynakların bütün kaynaklar arasında eşit olarak paylaşıldığını varsaymıştık. Bazen bazı kaynaklara diğerlerinden daha fazla pay ayırmak isteyebiliriz. Özellikle, , , ... , gibi ağırlıkları gibi kaynaklarla, kaynaklar onların üzerindeki yetkilerini yansıtmak için ilişkilendirebiliriz.

Max-min ağırlıklı adil paylaşım tahsisatını tanımlamada bu gibi ağırlıkların kullanılması görüşüne şu sebeplerden ulaştık.

- Kaynaklar artan istek sıralamasına göre tahsis edilir, ağırlıkları yoluyla normalleştirilir.
- İhtiyacından daha geniş bir kaynak paylaşımı sağlayan kaynak yoktur.
- Kaynaklar ile yerine getirilmemiş istekler arasında ağırlıklarına oranla bir kaynak paylaşımı gerçekleşir.



Waiting lines (queues)

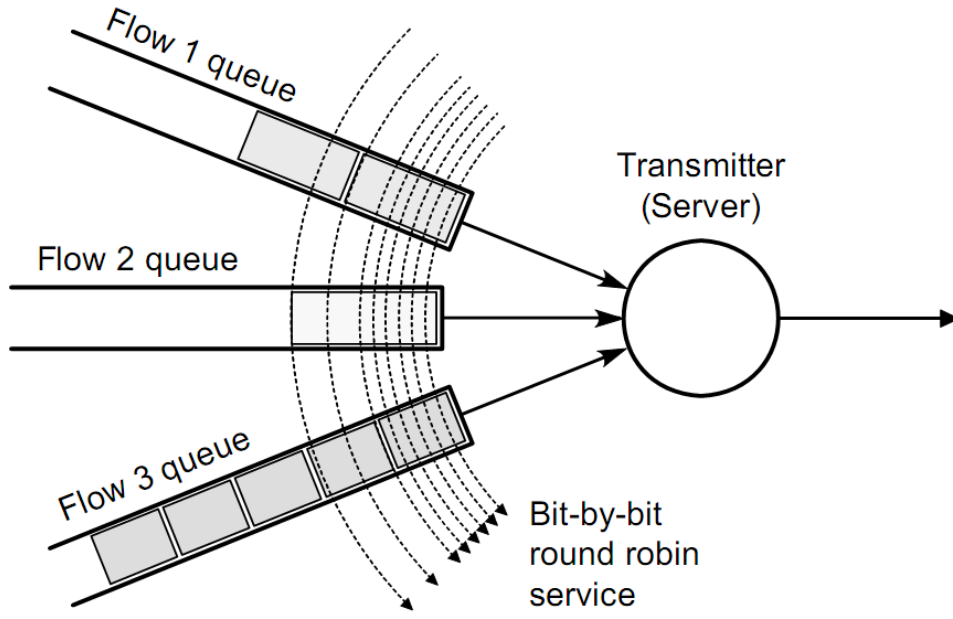


Şekil 4: Dinamik Adil Paylaşım Problemi: Mevcut bekleyen müşteriler hangi sırada servise çağrılacaklar

Min-max adil paylaşım yöntemi - Min-Max fair share (MMFS) bölüşülmüş az bir kaynağın, adil paylaşılmasının ideal bir tanımıdır. Verilen kaynak kapasitesi ve adet istemci için, MMFS yöntemi ile bir istemci en azından $\frac{1}{n}$ kadar kaynak elde eder. Eğer bazı istemciler hak ettiklerinden daha azına ihtiyaç duyarlarsa, diğer istemciler $\frac{1}{n}$ den daha fazlasına sahip olabilirler. Ağırlıklı MMFS yönteminde ise bir istemcisi en azından $\frac{1}{n}$ kadar bir kaynak paylaşımını garantiler. Ancak MMFS in zaman içinde kaynaklara olana talebin değiştiği dinamik bir sistemde bunu nasıl yapacağı açıkça belirtilmemiştir.

Problemi daha iyi anlamak için Şekil 1-4 'te gösterilen havalimanı kayıt işlemi senaryosunu ele alalım. Tek bir pencere (sunucu) olduğunu ve birinci sınıf ve ekonomi sınıfı yolcularına benzer ağırlığın verildiğini varsayalım. Bu durumda soru şu, bekleyen müşteriler hangi sıralamaya göre servise çağrılacaklar öyle ki iki sıranın da sunucu kaynağına eşit oranda erişim hakları var.

Belirtilen servis süresi içinde (Şekil 4), okuyucu ortalama üzerinde doğruluğu sağlayabilmeye dayalı şöyle bir sezgiye sahip olabilir. İki ekonomi sınıfı yolcusu çağırılmadan önce bir birinci sınıf yolcu çağrılır ve son olarak ta ekonomi sınıfı yolcusu çağrılır. Bu kısmın kalanları bunu başarmalarına göre yeniden gözden geçirilir, bu nedenle (Weighted) MMFS uzun vade de ortalama üzerinden kaynak tahsisatını garantiler.



Şekil 5: Bit-bit geliştirilmiş işlemci paylaşımı

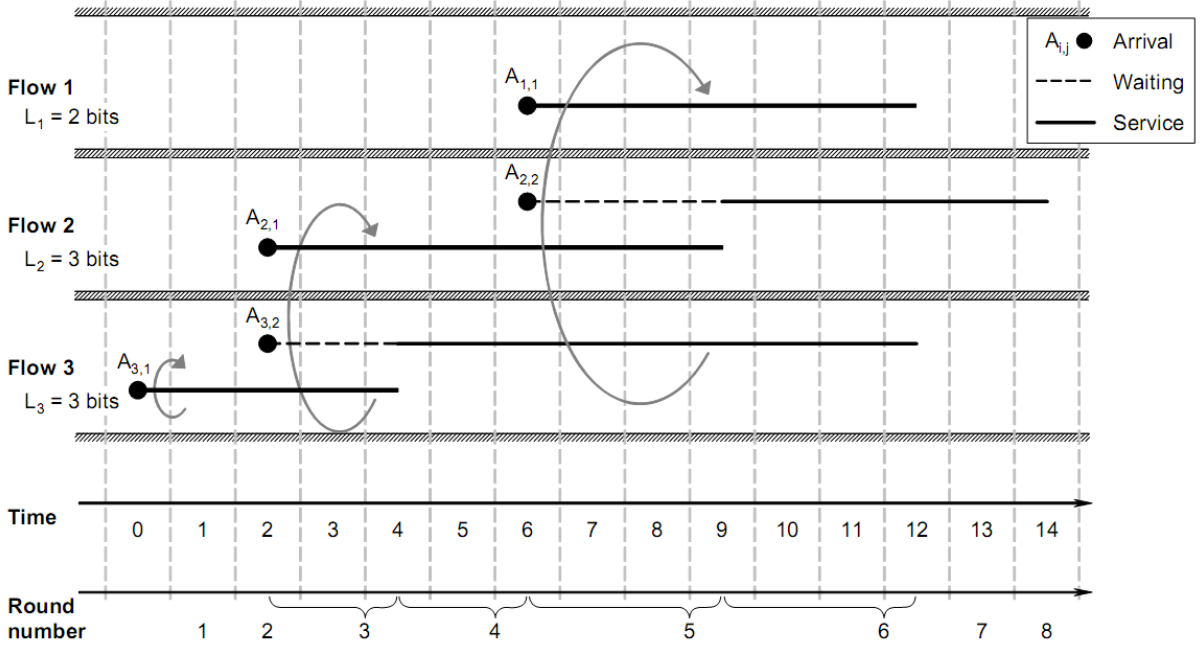
Genelleştirilmiş İşlemci Paylaşımı

MMFS doğrudan ağ senaryolarında uygulanamaz. Çünkü paketler atomik birimler olarak iletilir ve farkı uzunluklarda olabilirler ve bu yüzden farklı iletim zamanları gerektirirler. Pratik tekniğe ulaşmak için, ideal bir teknik olarak bilinen Genelleştirilmiş İşlemci Paylaşımı (GPS) tekniğinden başlayalım.

GPS farklı akımlara ait paketler için bekleyen farklı hatlar sağlar. Bu uygulama için iki tür kısıtlama mevcuttur.

- Bir paket bekleyen bir hattın üzerinden sıçrayamaz. Kuyruklar içinde FCFS yapısı kullanılır.
- Servis başkalarına engel olmaz, bunun anlamı servis içinde mevcut herhangi bir paket bir başkasına engel olmaz.(servis esnasında)

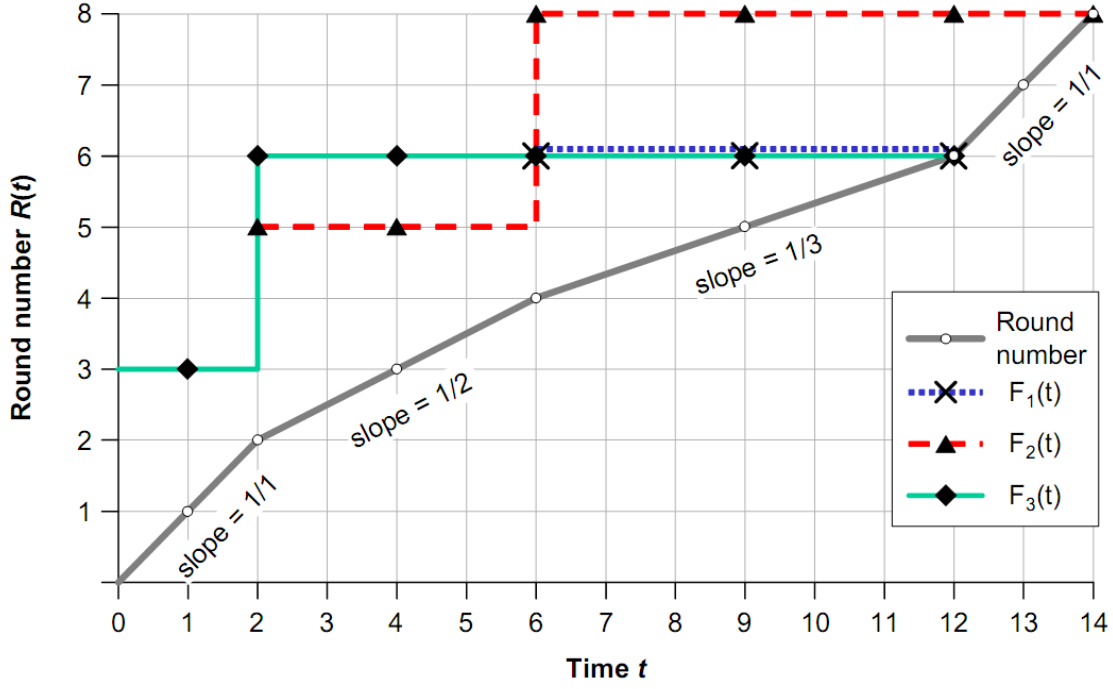
GPS birbirlerini bit-bit takip eden işlemlerin oluşturduğu bir yapıdır (Şekil 5). Burada, yönlendirici kuyruk 1'den bir bit iletir, sonra da kuyruk 2'den bir bit iletir. Ve bu şekilde devam eder. Bütün kuyruklar için iletme hazır paketler vardır.



Şekil 6: Bit-bit GPS örneği. Çıkış servis hattı oramı Bit/s

sunucudaki akımdan gelen paketi gösterebilir. Şekil 5-6 'da bu durum gösteriliyor. 1. akımdaki paketlerin uzunluğu 2 bit, 2. ve 3. akımlar 3 bit uzunluğundadır. Sıfırıncı anda, 3. akımdan paketi ulaşır ve boş bir hat bulur, böylece onun iletimi aniden başlamış olur, her döngüde bir bit işlem görür. anında gelenler iki adettir: ve . Çünkü 3. akımda nin önündedir. Bu yüzden 'nin iletimi sağlanıncaya kadar beklemek zorundadır. 'in iletimi aniden başlar çünkü 2. akımda ki tek mevcut pakettir.

Ancak bir iletim döngüsü şimdi iki zaman birimi alır. Çünkü her döngüde iki akım servis edilir. Bitler atomik olarak gönderilmelidir, her zaman biriminde her akımdan bir bit, şekilde gösterildiği gibi.



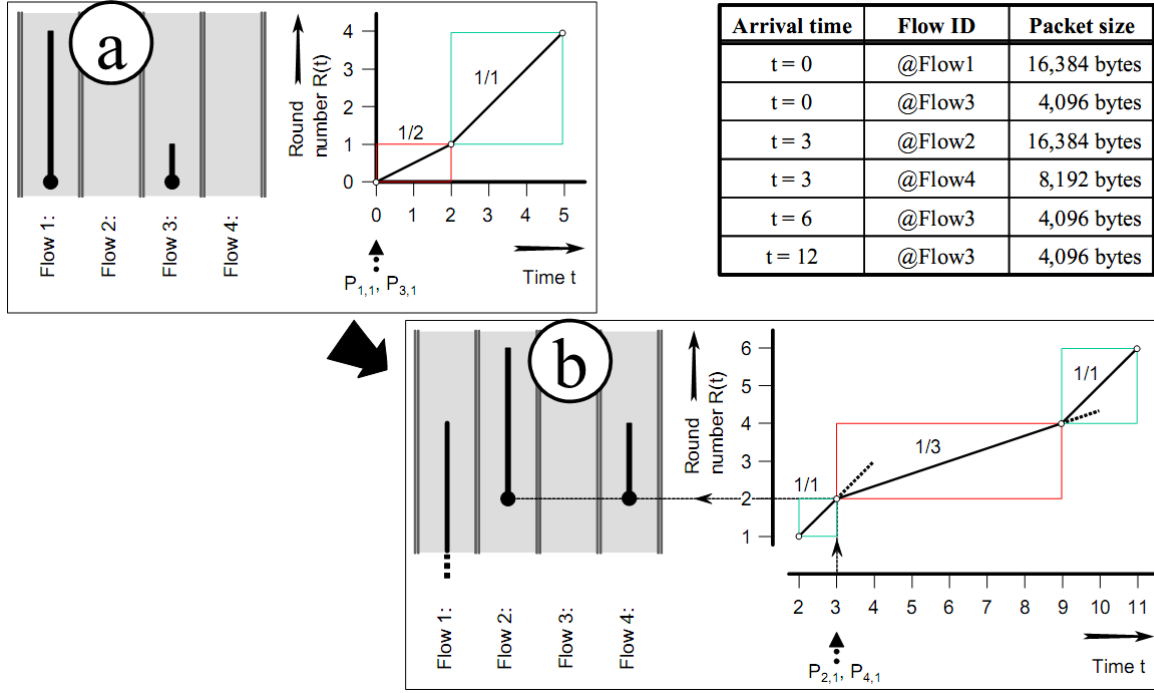
Şekil 7: Şekil 6 'daki döngü sayısı ile zaman arasındaki kısmi kısmi lineer ilişki. Ayrıca farklı akımlar için bitiş numaralarının gösterimi.

Adil Sıraya Alma (Fair Queuing)

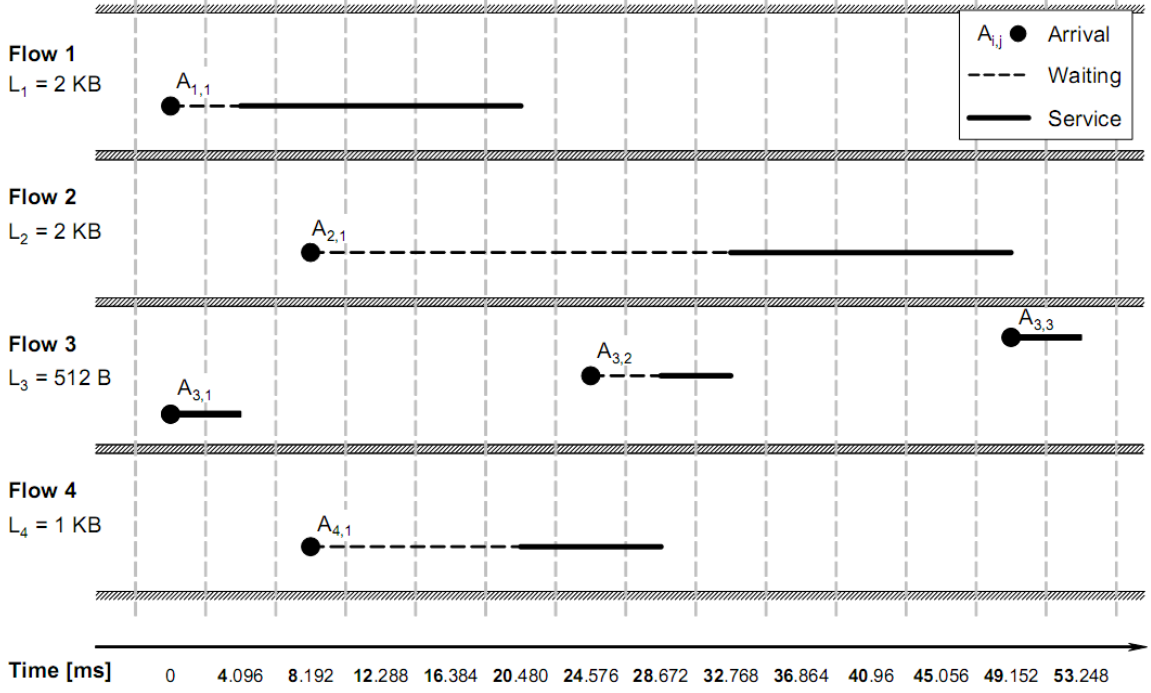
Gps 'e benzer olarak, FQ kullanan bir yönlendirici (router) her bir çıkış portunda farklı akımlara ait paketler için bekleyen farklı hatlar sağlar. Verilen bir paketin, eğer birbirini takip eden adımlarda bit-bit gönderiliyorsa (GPS), iletimi durdurulunca iletim için paketlerin sıralanıp etiketlenmesine FQ karar verir.

Bir servis döngüsünü bitiren paketi GPS içerisinde paketi bitme numarası olarak adlandırılır. Besbelli ki, paket bitme numarası, paket servisinin başlangıcındaki paket genişliğine ve döngü sayısına bağlıdır. Paket servisi edildiği anda genellikle gerçek zamandan farklı olan paket bitme numarasını geri çağırarak önemlidir.

GPS yönetiminde, paket, sadece servis alan veya servis kuyruğunda olan paketlerin her ikisi veya başka akımdan olan paketler bu akımdan ola mevcut paketleri servis için beklemelidir ve bu paketin servis başlangıcını etkileyemezler.



Şekil 8: Bit-Bit GPS yönetiminde döngü numaralarını kararlaştırma: (a) Başlangıç döngü numarası ve paketlerinin varışları için. (b) ve paketlerinin varışlarında döngü numaralarının yeniden hesaplanması.



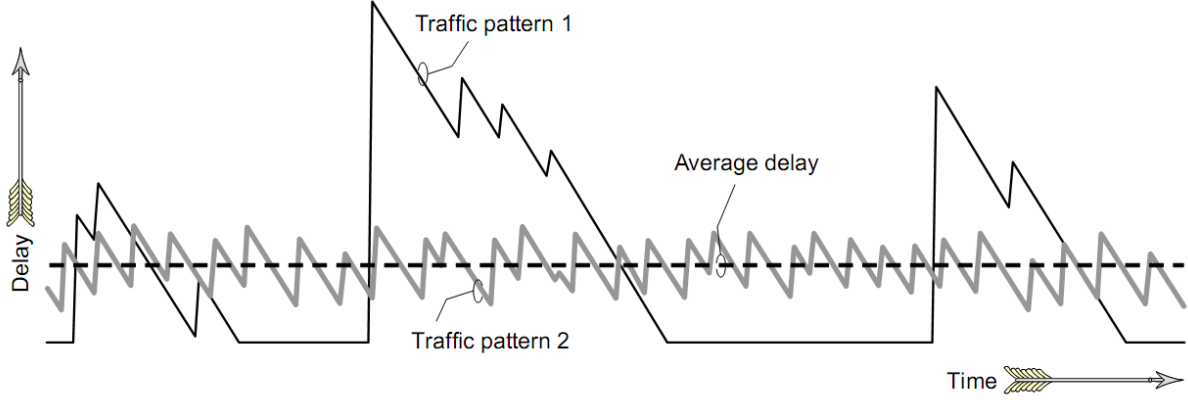
Şekil 9: Paket-paket FQ için zaman diyagramı

1-c. Ağırlıklı Adil Sıralama (Weighted Fair Queuing)

Şimdi $1, 2, \dots, n$ akım kaynakları ile birleşmiş w_1, w_2, \dots, w_n ağırlıklarının göreceli yetkilerine göre iletim yoluna sahip olduğunu varsayalım. Daha önce de söylenildiği gibi, bir kuyruk kendi akımını sağlamaya çalışır. Weighted-MMFS Ağırlıklı MMFS yönteminde, bir istemcisi en azından toplam bant genişliğinden $\frac{w_i}{\sum w_j}$ kadar bir bant genişliği elde etmeyi garantiler.

Ağırlıklı adil sıraya alma yönteminin (WFQ) her döngü için farklı numaralı bitleri kuyruklar arasında paylaştırdığı varsayılır. Her döngüdeki bit numaraları ağırlıkları ile orantılı olarak kuyruklara dağıtılır. Böylece iki kat ağırlıklı kuyruklar iki bit-döngüsü gerçekleştirirler.

Paket-paket WFQ, bit-bit WFQ ele alınarak aşağıdaki gibi genelleştirilebilir. uzunluklu anında erişen bir paket için bitiş numarası WFQ için — olarak hesaplanır.



Şekil 10: Farklı trafik örnekleri benzer gecikmeleri verebilir.

Bütün kuyruklar eşit bir büyüklükte düzenlenirler, bu yüzden en yüksek trafiğe sahip akımlardaki paket iletimi çoğunlukla yavaşlar, ve adil bir paylaşım kapasitesine sahip daha yavaş bir trafiğe izin verir.

Bu nedenle aç gözlü olmanın hiçbir avantajı yoktur. Aç gözlülük kuyruğun uzamsına neden olur. Çünkü belirli zaman periyotlarında ayrılması gereken kayıtların ayrılmamasına neden olur.

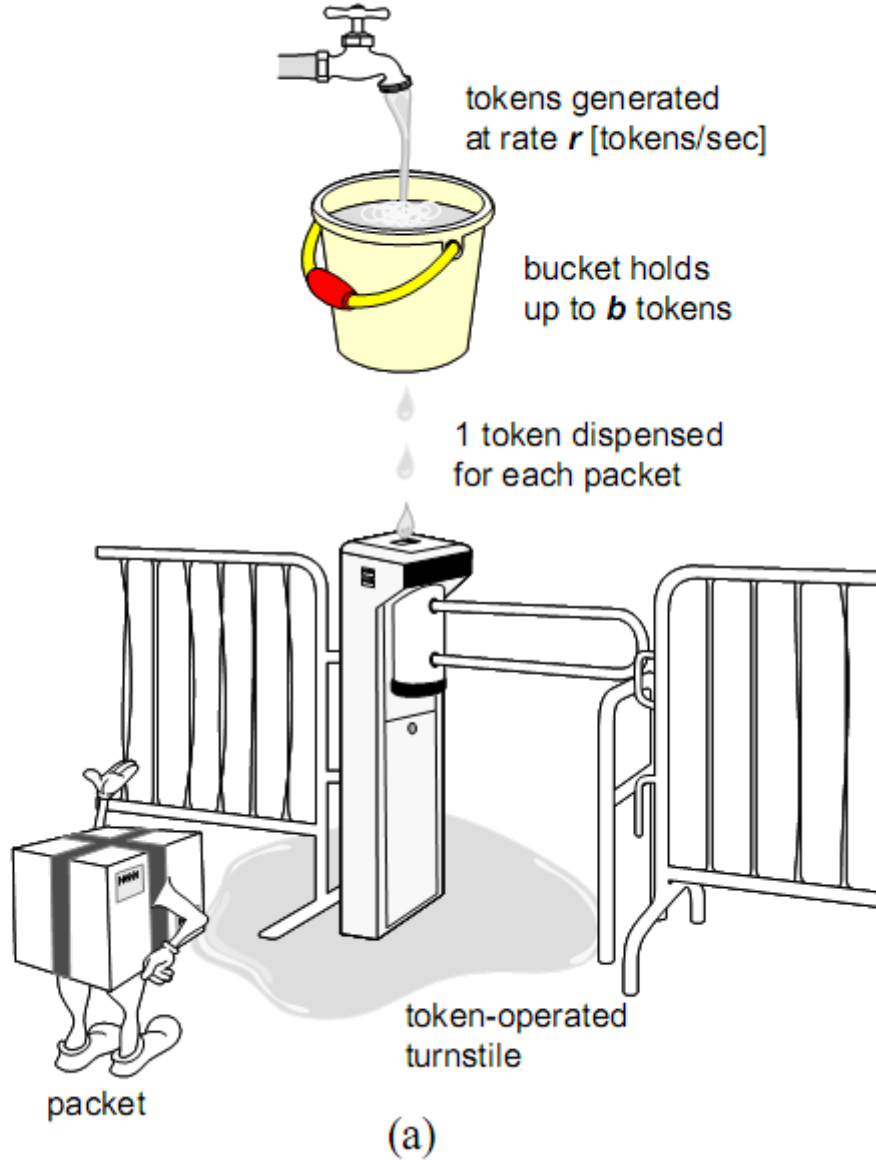
Problem sadece doyumsuz kaynaktan kaynaklanmaz fakat gecikmiş veya kaybolmuş paketler, ağ üzerindeki kaynakları zayıflatır. Bu nedenle bunların ağ erişimlerine izin verilmemelidir. Bu da Kontrol altında tutma (*policing*) aşamasının vazifesidir.

1. Kontrol Altında Tutma (Policing)

Şimdiye kadar, iletim bant genişliğinin veya başka ağ kaynaklarının WFQ kullanarak nasıl adil olarak paylaşılacağını gördük. Ancak bu trafik akışındaki gecikme sınırlamalarını ve düşük kayıpları sağlamayacaktır.

Bir paket-paket FQ planlayıcısı kaynakları eşit olarak paylaşımını garantiler. Ancak kabul edilebilir bir gecikme ortalaması ayrı paketler için mükemmel bir değişkenlik sağlar.

Bu durum Şekil 10'da gösterilmektedir. Multimedya uygulamaları gecikme değişkenliğine özellikle duyarlıdır. Paketlerin numaralarını düzenleyebilmenin bir yolu da delik kova düşüncesini kullanan yönlendirici içinden her zaman biriminde geçebilen özel bir akım kullanmaktır. (Şekil 11)



Şekil 11: Delik Kova (a)

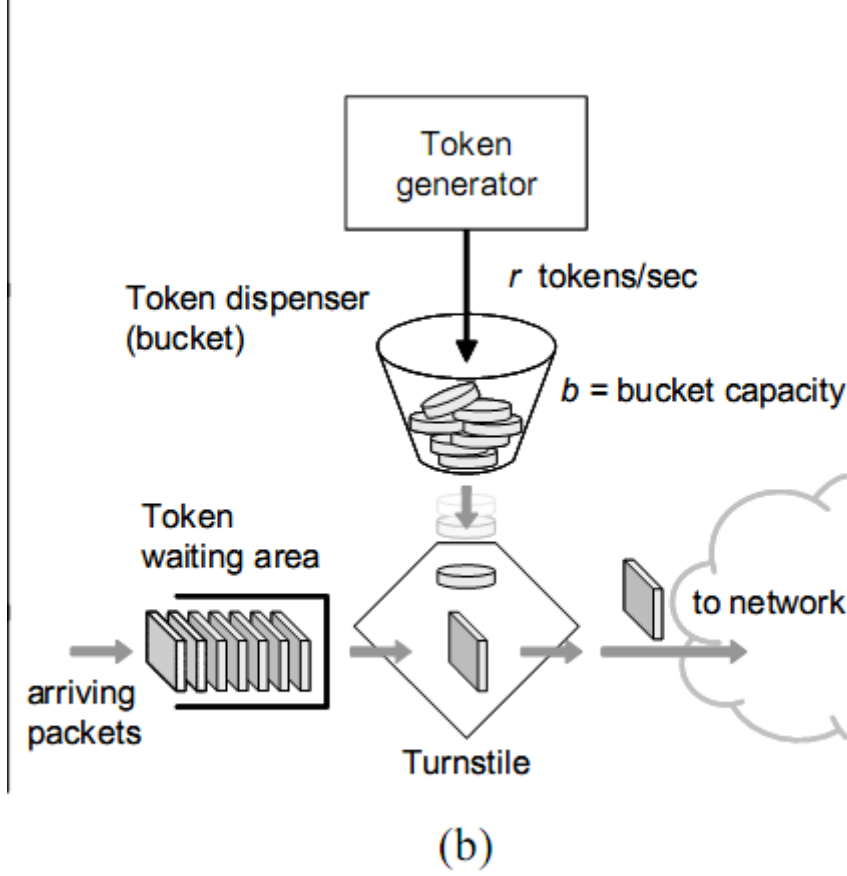
Tepe Oranı (Peak Rate): Çok kısa bir zaman periyodu içinde bir akımın gönderebildiği paketlerin numarasıdır.

Ortalama Oran (Average Rate) : Bir zaman penceresi üzerinden özel bir akımın gönderdiği paket sayılarının ortalamasını işaret eder.

Patlama Büyüklüğü (Burst Size): Ağ içerisindeki özel bir akım tarafından kısa bir zaman aralığında gönderilen toplam paketlerin numaralarını tutar.

Bir paket yönlendiriciye ulaştığında işleme başlamadan önce alandan bir simge çeker. Eğer alan boş ise paket işaret için beklemek zorundadır.

Eğer paketler benzer büyüklüğe sahip iseler, algoritma tarif edildiği gibi kullanılır. Ancak değişken uzunlukta paketler kullanılıyorsa, bu genellikle her bir işaret için sabit byte numaraları verilmesine sadece bir paket kullanılmasında daha iyi izin verir.



Şekil 11: Delik Kova (b)

2. Aktif Kuyruk Yönetimi (Active Queue Management AQM)

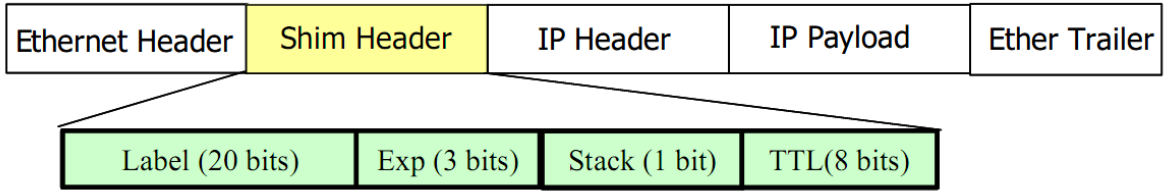
Paket düşürme stratejileri gelen paket için bellek olamayışı ile ilgilidir. Gelen paketlerin düşürülmesiyle ilgili temel politika (drop-tail) sondan düşüş politikasıdır.

AQM algoritmaları çok gelişmiş algoritmalar kullanır. En geniş biçimde tanınan ve üzerinde çalışılan algoritma Random Early Detection (RED) algoritmasıdır.

3.a Rastgele Erken Buluş-Random Early Detection (RED) Algorithm

4. Multiprotocol Label Switching (MPLS)

Multiprotocol Label Switching (MPLS) sanal devrelerin bazı özelliklerini datagramların dinçlik ve esnekliği ile birleştirir. Yolu düzenlemek için ip adreslerine ip yönlendirme protokollerine dayanır.



Şekil 12: MPLS Etiketli Çerçeve tabanlı paketler için

Neden MPLS:

- Aşırı hızlı gönderim
- Etiket eşleştirme çok daha hesaplı
- Routing yerine switching kullanımı
- İp trafik mühendislik hesapları
- Virtual Private Networks
- Kontrol edilebilir ilerleme mekanizması
- Koruma ve onarım