

OK.

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**ATM (ASENKRON TRANSFER MODU) AĞLARDA
TİKANIKLIK DENETİMİ**

**YÜKSEK LİSANS TEZİ
Müh. Meltem GÖKKAYA
504970623011**

101031

101031

Tezin Enstitüye Verildiği Tarih : 17 Ocak 2000

Tezin Savunulduğu Tarih : 1 Şubat 2000

Tez Danışmanı : Prof.Dr. Bülent ÖRENCİK

Diğer Jüri Üyeleri Prof.Dr. Günsel DURUSOY

Y. Doç.Dr. Sema OKTUĞ

ŞUBAT 2000

ÖNSÖZ

Bu çalışmanın amacı, geniş bandlı haberleşme ağları için geleceğin teknolojisi gözüyle bakılan ATM (Asenkron Transfer Modu)' in çözüm bekleyen önemli problemlerinden biri olan tıkanıklık denetimi hakkında detaylı bilgi vermektir.

Bu çalışmanın oluşturulmasında bana destek veren danışman hocam Prof. Dr. Bülent ÖRENCİK' e teşekkür ederim. Ayrıca her türlü desteğini gördüğüm aileme ve arkadaşlarıma teşekkür etmeyi bir borç biliyorum.

Şubat 2000

Meltem GÖKKAYA

İÇİNDEKİLER

TABLO LİSTESİ	vi
ŞEKİL LİSTESİ	vii
ÖZET	viii
SUMMARY	ix
1. GİRİŞ	1
2. ASENKRON TRANSFER MODU (ATM)	3
2.1. ATM Kavramı	3
2.2. ATM Katman Yapısı	5
2.3. ATM Ağlarındaki Servisler	7
3. TIKANIKLIK DENETİMİ	10
3.1. Tıkanıklık Denetiminin Genel İlkeleri	10
3.2. Tıkanıklık ile İlgili Problemlerin ve Çözümlerin Sınıflandırılması	11
3.3. Ağlarda Kullanılan Tıkanıklık Denetim Yöntemleri	12
3.3.1. Tıkanıklığı Önleyen Yöntemler	12
3.3.2. Trafik Biçimlendirme	14
3.3.2.1. Delikli Kova Algoritması	14
3.3.2.2. Jetonlu Kova Algoritması	15
3.3.3. Akış Belirtme	16
3.3.4. Görüntü Devre Alt Ağlarda Tıkanıklık Denetimi	17
3.3.5. Uyarı Paketleri	17
3.3.5.1. Ağırlıklı Adil Kuyruklama	18
3.3.5.2. Düğümünden Düğüme Uyarı Paketi	19
3.3.6. Yük Atma	19
3.3.7. Seyirme Denetimi	20
3.3.8. Çoklu Göndermede Tıkanıklık Denetimi	20
3.3.8.1. Kaynak Ayırma Protokolü	21
4. ATM TEMELLİ AĞLARDA TIKANIKLIK DENETİMİ	22
4.1. Kabul Denetimi Yöntemleri	22
4.2. Kaynak Ayırma Yöntemi	23
4.3. Hız Temelli Tıkanıklık Denetimi	23

5. ÇAĞRI KABUL DENETİMİ YÖNTEMLERİ	27
5.1. Çağrı Kabul Denetimi	27
5.1.1. İstatistiksel Olmayan Çoğullamada Çağrı Kabul Denetimi	28
5.1.2. İstatistiksel Çoğullamada Çağrı Kabul Denetimi	28
5.2. Bir Varış Sürecinin Karakterize Edilmesi	29
5.3. Çağrı Kabul Denetimi Yöntemleri	32
5.3.1. Eşdeğer Band Genişliği Yaklaşımı	32
5.3.2. Hücre Kayıp Olasılığının Üst Sınırı	35
5.3.3. Hızlı Tampon / Band Genişliği Tahsisi	36
5.3.4. Zaman Pencereleeri	38
5.4. Eşdeğer Band Genişliği Temelli Kabul Kriterleri	39
5.4.1. Guerin-Ahmadi-Naghshineh (GAN) Kriteri	39
5.4.2. Elwalid-Mitra Kriteri	45
5.4.3. Geliştirilmiş Elwalid-Mitra Kriteri	47
5.4.4. Sohraby Kriteri (Yoğun Trafik Yaklaşımı)	48
5.5. Eşdeğer Band Genişliği Temelli Kabul Kriterlerinin Başarımlarının Karşılaştırılması	50
5.5.1. Girilen Kaynak Parametreleri, Çıkış Portunun Hızı, Tampon Boyutu ve ϵ Değeri İçin Eşdeğer Band Genişliği Temelli Kabul Kriterlerinin Kabul Bölgesini Hesaplayan PASCAL Programının Kaynak Kodu	50
5.5.2. Eşdeğer Band Genişliği Temelli Kabul Kriterlerinin Kabul Bölgesini Hesaplayan Programın Belirli Tampon Boyutlarında Yürütülmesi Sonucu Elde Edilen Değerler	56
6. "ON-OFF" KAYNAK MODELİNE DAYALI OLARAK KABUL EDİLEBİLECEK MAKSİMUM BAĞLANTI SAYISINI HESAPLAYAN BENZETİM PROGRAMLARI	62
6.1. Kaynakların Aynı Anda İletime Başladığı ve ON-OFF Sürelerinin Sabit Olduğu Durum İçin Yazılan Benzetim Programı	64
6.1.1. Kaynakların Aynı Anda İletime Başladığı ve ON-OFF Sürelerinin Sabit Olduğu Durum İçin Yazılan Benzetim Programının Kaynak Kodu	65
6.2. Kaynakların Belli Bir Aralıkta Rastgele İletime Başladığı ve ON-OFF Sürelerinin Sabit Olduğu Durum İçin Yazılan Benzetim Programı	68
6.2.1. Kaynakların Belli Bir Aralıkta Rastgele İletime Başladığı ve ON-OFF Sürelerinin Sabit Olduğu Durum İçin Yazılan Benzetim Programının Kaynak Kodu	70
6.3. Kaynakların Belli Bir Aralıkta Rastgele İletime Başladığı ve ON-OFF Sürelerinin Üssel Dağılıma Göre Rastgele Değiştiği Durum İçin Yazılan Benzetim Programı	74

6.3.1. Kaynakların Belli Bir Aralıkta Rastgele İletime Başladığı ve ON-OFF Sürelerinin Üssel Dağılıma Göre Rastgele Değiştiği Durum İçin Yazılan Benzetim Programının Kaynak Kodu	77
6.4. Benzetim Programlarının Belli Değerler İçin Yürütülmesi Sonucu Elde Edilen Sonuçlar	84
7. EŞDEĞER BAND GENİŞLİĞİ TEMELLİ KABUL KRİTERLERİNİN VE BENZETİM PROGRAMLARININ KARMAŞIKLIK ANALİZİ	95
7.1. Eşdeğer Band Genişliği Temelli Kabul Kriterlerinin Karmaşıklik Analizi	95
7.1.1. Guerin-Ahmadi-Naghshineh Kriterinin Karmaşıklik Analizi	95
7.1.2. Elwalid - Mitra Kriterinin Karmaşıklik Analizi	101
7.1.3. Geliştirilmiş Elwalid - Mitra Kriterinin Karmaşıklik Analizi	104
7.1.4. Sohraby Kriterinin Karmaşıklik Analizi	107
7.2. Benzetim Programlarının Karmaşıklik Analizi	110
7.2.1. Kaynakların Aynı Anda İletime Başladığı ve ON-OFF Sürelerinin Sabit Olduğu Durum İçin Yazılan Benzetim Programının Karmaşıklik Analizi	110
7.2.2. Kaynakların Belli Bir Aralıkta Rastgele İletime Başladığı ve ON-OFF Sürelerinin Sabit Olduğu Durum İçin Yazılan Benzetim Programının Karmaşıklik Analizi	113
7.2.3. Kaynakların Belli Bir Aralıkta Rastgele İletime Başladığı ve ON-OFF Sürelerinin Üssel Dağılıma Göre Rastgele Değiştiği Durum İçin Yazılan Benzetim Programının Karmaşıklik Analizi	116
8. SONUÇLAR VE TARTIŞMA	119
KAYNAKLAR	124
EKLER	126
ÖZGEÇMİŞ	146

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 2.1. ATM Ağlarındaki Servislerin Karakteristikleri	8
Tablo 2.2. Bazı Servis Kalitesi (QoS) Parametreleri	9
Tablo 5.1. A Tipi Kaynaklar İçin Kabul Bölgesinin Tampon Boyutuna ve Kullanılan Kabul Kriterine Göre Değişimi	58
Tablo 5.2. B Tipi Kaynaklar İçin Kabul Bölgesinin Tampon Boyutuna ve Kullanılan Kabul Kriterine Göre Değişimi	59
Tablo 5.3. A' Tipi Kaynaklar İçin Kabul Bölgesinin Tampon Boyutuna ve Kullanılan Kabul Kriterine Göre Değişimi	60
Tablo 5.4. A'' Tipi Kaynaklar İçin Kabul Bölgesinin Tampon Boyutuna ve Kullanılan Kabul Kriterine Göre Değişimi	61
Tablo 6.1. Kaynakların Aynı Anda İletime Başladığını Varsayan Birinci Benzetim Programının A Tipi Kaynak ve Sonsuz Yaşam Süresi Seçeneği İçin Çalıştırılması Sonucu Elde Edilen Değerler	85
Tablo 6.2. Kaynakların Aynı Anda İletime Başladığını Varsayan Birinci Benzetim Programının A Tipi Kaynak ve Belirli Yaşam Süresi Seçeneği İçin Çalıştırılması Sonucu Elde Edilen Değerler	86
Tablo 6.3. İkinci Benzetim Programının A Tipi Kaynak ve Kaynakların Bir Periyod Süresi İçerisinde Rastgele İletime Başlaması Durumu İçin Çalıştırılması Sonucu Elde Edilen Değerler	87
Tablo 6.4. İkinci Benzetim Programının A Tipi Kaynak ve Kaynakların Belirli Bir Yaşam Süresi İçerisinde Rastgele İletime Başlaması Durumu İçin Çalıştırılması Sonucu Elde Edilen Değerler	89
Tablo 6.5. Üçüncü Benzetim Programının A Tipi Kaynak ve Sonsuz Yaşam Süresi İçin Çalıştırılması Sonucu Elde Edilen Değerler...	90
Tablo 6.6. Üçüncü Benzetim Programının A Tipi Kaynak ve Kaynakların Belirli Bir Yaşam Süresi İçerisinde Rastgele İletime Başlaması Durumu İçin Çalıştırılması Sonucu Elde Edilen Değerler	91
Tablo 6.7. A Tipi Kaynaklar İçin Eşdeğer Band Genişliği Temelli Dört Kriter ve Benzetim Programları İle Bulunan Maksimum Bağlantı Sayısı Değerleri	92
Tablo 6.8. İkinci Benzetim Programının A Tipi Kaynak İçin Çalıştırılması Sonucu Elde Edilen Değerler	93
Tablo 6.9. Üçüncü Benzetim Programının A Tipi Kaynak İçin Çalıştırılması Sonucu Elde Edilen Değerler	94

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 2.1	ATM Hücre Yapısı 3
Şekil 2.2	BISDN ATM Protokol Referans Modeli 5
Şekil 3.1	Tıkanıklık Problemlerinin Sınıflandırılması 11
Şekil 3.2	Tıkanıklık Denetim Yöntemlerinin Sınıflandırılması 12
Şekil 3.3	Tıkanıklığı Etkileyen Stratejiler 13
Şekil 3.4	Delikli Kova Algoritması 14
Şekil 3.5	Örnek Akış Tarifnamesi 16
Şekil 4.1	RM Hücrelerinin İzlediği Yol 25
Şekil 5.1	Çıkış Tamponlu Bir ATM Anahtarı 27
Şekil 5.2	Bir ATM Çoğullayıcı 29
Şekil 6.1	Benzetim Programlarında Kullanılan ATM Kaynak Modeli 62
Şekil 6.2	Birinci benzetim programında kaynakların iletme başlama şekli 64
Şekil 6.3	İkinci benzetim programında kaynakların periyod süresi (Ton+Toff) içinde rastgele iletme başlaması durumu..... 69
Şekil 6.4	İkinci benzetim programında kaynakların girilen yaşam süresi içinde rastgele iletme başlaması durumu..... 69
Şekil 6.5	Üçüncü benzetim programında kaynakların periyod süresi içinde rastgele iletme başlaması durumu 75
Şekil 6.6	Üçüncü benzetim programında kaynakların girilen yaşam süresi içinde rastgele iletme başlaması durumu 75

ATM (ASENKRON TRANSFER MODU) AĞLARDA TIKANIKLIK DENETİMİ

ÖZET

“ATM (Asenkron Transfer Modu) Ağlarda Tıkanıklık Denetimi” isimli bu tez çalışmasında, ATM ağları ve tıkanıklık denetimi üzerine genel bir bakıştan sonra, ATM ağlarda çözülmesi gereken en önemli problemlerden biri olan tıkanıklık denetimi üzerinde detaylı bir inceleme yapılmıştır. Tez yazılımında izlenen plan şöyledir :

İkinci bölümde, ATM hücre yapısı, ATM ağların temel özellikleri, referans modeli, katman yapısı, katmanların görevleri ve ATM ağlarda kullanılan servis tipleri incelenmiştir.

Üçüncü bölümde genel olarak tıkanıklık denetimi açıklanmıştır. Tıkanıklık denetiminin genel ilkeleri, tıkanıklık denetimi problemlerinin ve çözümlerinin sınıflandırılması ve ağlarda kullanılan tıkanıklık denetim yöntemleri bu bölümün konularıdır.

ATM ağlarda kullanılan tıkanıklık denetim yöntemleri ise dördüncü bölümde incelenmiştir.

ATM temelli ağlarda tıkanıklığı denetlemek için kullanılan yöntemlerden biri olan çağrı kabul denetimi (call admission control-CAC), beşinci bölümün konusudur. Bu bölümde çağrı kabul denetimi kavramı ve ATM ağlarda varış sürecinin modellenmesi anlatılmıştır. ATM ağlarda kullanılan kabul denetimi algoritmaları genel olarak anlatıldıktan sonra, bu algoritmalarından eşdeğer band genişliği yaklaşımına dayalı dört kriter, Elwalid-Mitra, Guerin-Ahmadi-Naghshineh (GAN), Sohraby ve Geliştirilmiş Elwalid-Mitra kriterleri tanıtılmıştır. Ayrıca seçilen kriterin kabul bölgesini hesaplayan PASCAL programı bu bölümde verilmiştir.

Bu çalışmada, aynı tipteki kaynaklar için kabul edilebilecek maksimum bağlantı sayısını hesaplayan, PASCAL programlama dilinde üç farklı benzetim programı yazılmıştır. Benzetim programlarında ATM kaynağı, ON-OFF kaynak modeline dayalı olarak modellenmiştir. Yazılan benzetim programlarının açıklamaları ve kaynak kodları altıncı bölümde verilmiştir. Benzetim programlarının akış diyagramları ise eklerde verilmiştir.

Eşdeğer band genişliği temelli kabul kriterlerinin ve benzetim programlarının karmaşıklık analizi son bölümde verilmiştir.

CONGESTION CONTROL IN ATM (ASYNCHRONOUS TRANSFER MODE) NETWORKS

SUMMARY

In this M.S. thesis titled as “Congestion Control in ATM (Asynchronous Transfer Mode) Networks”, congestion control which is the most important problem for ATM networks are examined in detail after having given a general outlook on the ATM network and congestion control. The schedule of the thesis can be summarized as follows :

In the second chapter, the ATM cell structure, the main features of the ATM networks; reference model, layer structure, the tasks of the layers and service types used in ATM networks are examined.

In the third chapter, congestion control is explained generally. General principles of the congestion control, the classification of the congestion control problems and solutions and congestion control methods used in networks are the subjects included in this chapter.

Congestion control in ATM networks is examined in the fourth chapter.

Call Admission Control (CAC), which is a method used to control congestion in ATM networks, is the subject of the fifth chapter. Call admission control and the characterization of the arrival process in ATM networks are explained in this chapter. After explaining call admission control methods used in ATM networks, four equivalent bandwidth based criterions Elwalid-Mitra, Guerin-Ahmadi-Naghshineh (GAN), Sohraby and Refined Elwalid-Mitra criterions are introduced. The program coded in PASCAL to calculate the admission region of the selected criterion is also given in this chapter.

In this study, three different simulation programs are coded in PASCAL to calculate maximum number of the acceptable connections for the same type sources. In simulation programs, an ATM source is modelled as an ON-OFF type source. The explanations and source codes of the simulation programs are given in the sixth chapter. Flow charts of the simulation programs are given in the appendix.

The complexity analysis of the equivalent based admission criterions and simulation programs is given in the last section.

1. GİRİŞ

Asenkron Transfer Modu (ATM), çeşitli trafik karakteristiğine sahip uygulamalara yüksek band genişliği ve garantili servis kalitesi sağladığından son zamanlarda çok uygulama alanı bulan bir teknolojidir.

Diğer ağlarda olduğu gibi tıkanıklık (congestion), ATM ağlar için de başarılması gereken temel sorunlar arasında yer alır. Ancak farklı trafik karakteristiklerine hizmet veren ATM ağlarda tıkanıklık denetimi diğer ağlardan çok daha zordur ve bunun için çok sayıda yöntem önerilmiştir.

ATM ağlarda tıkanıklığı denetlemek için kullanılan yöntemlerden biri de, çağrı kabul denetimidir (call admission control-CAC). Çağrı kabul denetimi, bir çağrı isteğinin kabul edilip edilmeyeceğine karar verme işlemidir. Ağda yeni bir bağlantı isteği alındığı zaman, bu isteğin kabul edilip edilmeyeceğine karar vermek için bir kabul kriteri yürütülür. Ağ, kurulu bağlantıların servis kalitelerini etkilemeden yeni bağlantının gerektirdiği servis kalitesini sağlayabilecek durumda ise bağlantı isteği kabul edilir. Problem oldukça karmaşıktır. Çünkü bir kabul kriterinin sadece bütün bağlantıların servis kalitesi sağlanacak şekilde doğru karar vermesi yetmez, kabul kriteri aynı zamanda ağ kaynaklarının etkili bir biçimde kullanılmasını sağlamalıdır.

Bu problemi çözmek için çok sayıda kabul denetimi algoritması önerilmiştir. Önerilen bu yöntemler ATM ağlarda kullanılmasına rağmen, bu yöntemler problemin giderilmesinde ara çözümler olarak görülmelidir. Algoritmaların çoğu bağlantılara servis kalitesini sağlamak için gerekenden daha fazla band genişliği ayırırlar. Bu, ağ kaynaklarının etkili bir biçimde kullanılmamasına yol açar. Bazı algoritmalar ise, bağlantılara gerekenden daha az band genişliğinin ayrılmasına neden olabilirler. Bu da ağda ek güvenlik mekanizmalarının kullanılmasını gerektirir.

Bu çalışmada, ATM ağlarda tıkanıklığı denetlemek için kullanılan yöntemler incelenecek, bu yöntemlerden de özellikle eşdeğer band genişliği yaklaşımına dayalı kabul denetimi algoritmaları üzerinde durulacaktır.

Çalışmanın ikinci bölümünde genel olarak ATM hücre yapısı, ATM' in ana özellikleri, referans modeli, katman yapısı, katmanlarının görevleri ve ATM ağlarda kullanılan servis tipleri üzerinde durulacaktır. Ayrıca ATM ağların en önemli

özellikleri arasında yer alan kullanıcıya verilecek servis kalitesi kavramı açıklanacaktır.

Üçüncü bölümde ise genel olarak tıkanıklık denetimi konusu ele alınacaktır. Tıkanıklık denetiminin genel ilkeleri ve tıkanıklık problemlerinin ve bu problemi çözmek için öne sürülen yöntemlerin sınıflandırılması anlatıldıktan sonra, ağlarda kullanılan tıkanıklık denetim yöntemleri kısaca açıklanacaktır.

Son yıllarda popülerlik kazanan ve geleceğin ağ tipi olarak görülen ATM ağlarda kullanılan tıkanıklık denetim yöntemleri ise genel hatlarıyla dördüncü bölümde anlatılacaktır.

ATM temelli ağlarda tıkanıklığı denetlemek için kullanılan yöntemlerden biri olan çağrı kabul denetimi (call admission control-CAC), beşinci bölümün konusunu oluşturur. Bu bölümde öncelikle çağrı kabul denetimi kavramı açıklanacak, istatistiksel çoğullamada ve istatistiksel olmayan çoğullamada çağrı kabul denetiminin nasıl gerçekleştiği anlatılacaktır. Daha sonra kabul denetimi yöntemlerinin tasarlanmasında çok önemli bir yeri olan varış sürecinin modellenmesinden bahsedilecektir. ATM ağlarda kullanılan kabul denetimi algoritmaları genel olarak anlatıldıktan sonra, bu algoritmalarından eşdeğer band genişliği yaklaşımına dayalı dört kriter, Elwalid-Mitra, Guerin-Ahmadi-Naghshineh (GAN), Sohraby ve Geliştirilmiş Elwalid-Mitra kriterleri tanıtılacaktır. Ayrıca bu kriterlerin başarımlarını karşılaştırmak amacıyla yazılan ve girilen değerler için kriterlerin kabul bölgelerini hesaplayan PASCAL programının kaynak kodu ve bu programın belli değerlerde her bir kriter için ayrı ayrı yürütülmesi sonucu elde edilen sonuçlar bu bölümde verilecektir.

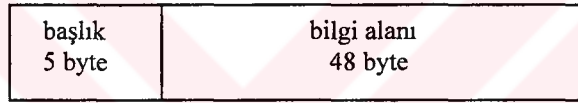
Bu çalışmada, kullanıcı tarafından girilen kaynak parametrelerine, tampon boyutuna, çıkış portunun hızına ve istenen servis kalitesi hücre kayıp olasılığına göre aynı tipteki kaynaklar için kabul edilebilecek maksimum bağlantı sayısını hesaplayan, PASCAL programlama dilinde üç farklı benzetim programı yazılmıştır. Benzetim programlarında bağlantı kurma isteğinde bulunan her bir kaynak, ON-OFF kaynak modeline dayalı olarak modellenmiştir. Yazılan benzetim programlarının açıklamaları, kaynak kodları ve akış diyagramları ise altıncı bölümde verilecektir.

Çalışmanın yedinci ve sonuncu bölümünde ise eşdeğer band genişliği temelli kabul kriterlerinden Elwalid-Mitra, Guerin-Ahmadi-Naghshineh (GAN), Sohraby ve Geliştirilmiş Elwalid-Mitra kriterlerinin ve yazılan üç ayrı benzetim programının karmaşıklık analizi yapılacaktır.

2. ASENKRON TRANSFER MODU (ATM)

2.1 ATM Kavramı

Asenkron Transfer Modu (ATM), geniş bantlı ISDN (B-ISDN) ağlarının transfer modu olarak adlandırılır. Buradaki transfer sözcüğü iletim ve anahtarlama işlemlerini içerir. ATM’ de veri iletimi “hücre” adı verilen sabit uzunluktaki paketler ile yapılır. Bu hücreler, 48’i veri, 5’i başlık bilgisi olmak üzere toplam 53 byte uzunluğundadır. Başlık kısmı ATM katmanının fonksiyonları için ayrılmıştır (Şekil 2.1).



Şekil 2.1 ATM Hücre Yapısı

ATM tabanlı ağlarda hücrelerin iletimi uygulamadan bağımsızdır. İsteğe göre değişken band genişliği sağlayabilir. Ancak veri iletiminde bu esneklik sağlanırken bazı önemli problemler de gündeme gelmektedir. Hücre kaybı, hücre iletimindeki gecikme ve bu gecikmelerdeki dağılım bozukluğu gibi problemler, servisler için gerekli olan servis kalitesini sağlamakta sorunlar çıkarmaktadır. Bu konular, ATM kavramında üzerinde en çok çalışılan konular olmuştur.

ATM’ in temeli, iki kavrama dayanır. Bunlar paket anahtarlama ve zaman bölmeli çoğullamadır (Time Division Multiplexing – TDM). Bu kavramların birleştirilmesiyle oluşturulan ATM’ in temel özellikleri şunlardır [1]:

- Veri alanı üzerinde herhangi bir hata kontrolü ve ATM ağı içinde bağlantılarda akış kontrolü yoktur.
- ATM bağlantı tabanlıdır (connection-oriented). Bilgi iletimine başlanmadan önce bir görüntü bağlantı kurulması gerekir. Bağlantı için gerekli kaynaklar mevcut ise bağlantı kurulur, aksi halde bağlantıya izin verilmez. Bağlantı sona erdikten sonra kaynaklar serbest bırakılır.

- Sabit ve küçük boyutlu paketler kullanılır. Bu paketlere hücre denir. Değişken uzunluklu paketler yerine sabit ve küçük boyutlu paketler kullanılması, yüksek hızda anahtarlama imkanı tanır ve servisler üzerinde herhangi bir kısıtlama getirmez. Ayrıca paket uzunluğunun kısa olması, tampon gecikmelerini azaltır.
- Hücrelerin başlık kısmındaki fonksiyonlar azaltılmıştır. Bunun nedeni, işlem hızını arttırmaktır. Başlık kısmının birinci fonksiyonu, ait olduğu görüntü devrenin tanıtılmasıdır. Ayrıca bu görüntü devre tanımlayıcısı üzerinde bazı hata kontrol ve düzeltme işlemleri de yapılır. Her hücrenin başlık kısmında hangi görüntü kanaldan iletileceğine dair bir tanımlayıcı (VCI) vardır. Başlık kısmında meydana gelecek bir hata, hücrenin yanlış yönlendirilmesine neden olabilir. Bu yüzden başlık bitleri üzerinde hata denetimi ve düzeltilmesi işlemleri yapılır. Böylece hatalar mümkün olduğunca aza indirilir.

ATM’ de başlık fonksiyonu kapsamında ele alınan görüntü kanal, görüntü bağlantı ve görüntü yol kavramları aşağıda kısaca açıklanmıştır.

Görüntü Bağlantı

Her görüntü bağlantı, başlıkta bulunan bir tanımlayıcı ile simgelenir. Başlık kısmında bu işlem için iki alan ayrılmıştır : Görüntü Kanal Tanımlayıcısı (Virtual Channel Identifier - VCI) ve Görüntü Yol Tanımlayıcısı (Virtual Path Identifier - VPI). VCI dinamik, VPI ise statik olarak tahsis edilen bağlantıları simgeler.

Görüntü Kanal

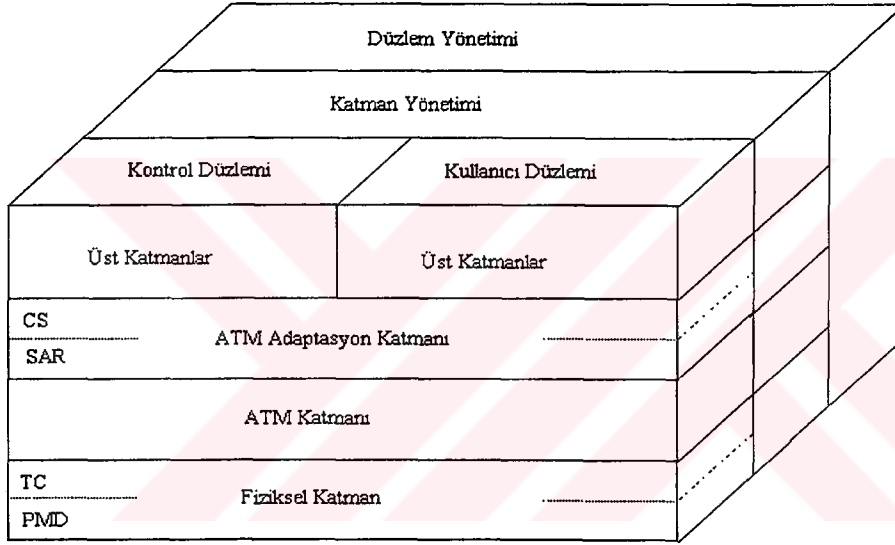
Her görüntü kanal bir VCI ile simgelenir. Bu VCI numarası bağlantı kurulurken atanır ve sadece yerel olarak anlamlıdır. Yani VCI, iki ATM düğümü arasında geçerlidir. Daha sonra anahtarlama ile VCI değeri değişebilir. Bağlantı koptuğunda hatlar üzerindeki VCI değerleri başka bağlantılar tarafından kullanılmak üzere serbest bırakılır.

Görüntü Yol

Uçlar arasında yarı özel bağlantılar kurulabilir. Bu bağlantılar pek çok eş zamanlı bağlantıya imkan tanır. Bu durum görüntü yol olarak tanımlanır. Ağ kaynakları yarı özel olarak tahsis edilir. Bu görüntü yolları tanımlamak için başlıkta görüntü yol tanımlayıcısı, VPI alanı bulunur.

2.2 ATM Katman Yapısı

ATM' in, BISDN protokol referans modeli Şekil 2.2' de gösterilmiştir [1]. Bu yapı ITU-T (International Telecommunications Union-Telecommunication) tarafından kabul edilmiştir. OSI (Open System Interconnection) modelinde olduğu gibi katmanlı bir yapı kullanılmıştır. Bu modelde kullanıcı ve kontrol düzlemleri vardır. Kullanıcı düzlemi kullanıcı bilgisinin, kontrol düzlemi ise işaretleşme bilgisinin iletiminde kullanılır. Katman yönetimi, ağın yönetilmesi ve temel işlemlerin yerine getirilmesinden sorumludur. Ek olarak referans modeline düzlem yönetimi adıyla üçüncü bir boyut eklenmiştir. Bunun görevi de düzlemlerin yönetimidir.



Şekil 2.2 BISDN ATM Protokol Referans Modeli

Bu yapıda üç tane katman tanımlıdır. Bunlar fiziksel katman, ATM katmanı ve ATM adaptasyon katmanı (AAL)' dır.

Fiziksel Katman

Fiziksel katmanın temel görevi bilginin iletilmesidir. Bu katman ikiye bölünmüştür : Fiziksel Ortama Bağımlı (Physical Medium Dependent-PMD) altkatmanı ve İletim Yakınsama (Transmission Convergence-TC) alt katmanı. PMD alt katmanı, fiziksel ortama bağımlı olarak bitlerin doğru iletilmesini ve alınmasını sağlar. TC alt katmanının görevi ise, ATM hücrelerinin iletim ortamına uygun olacak şekilde düzenlenmesidir. Böylece hücreler iletme hazır hale getirilir.

ATM Katmanı

ATM katmanı, verilerin iletiildiği fiziksel ortamdan tamamen bağımsızdır. ATM’deki VCI, VPI gibi kavramlar üzerindeki işlemler bu katmanın görevleri arasındadır. ATM katmanında aşağıdaki temel işlemler gerçekleştirir :

- Değişik bağlantılara ait farklı VCI/VPI numaralarına sahip hücrelerin tek bir fiziksel bağlantı üzerinden yönlendirilmesi.
- Hücrelerin adaptasyon katmanında iletilmeden önce başlık bilgilerinin çıkarılması ya da adaptasyon katmanından gelen hücrelere başlık bilgisinin eklenmesi.
- ATM anahtarlarında VCI/VPI bilgisinin dönüştürülmesi. Bu dönüştürme işlemi bir dönüşüm tablosu yardımı ile yapılır.
- Kullanıcı-ağ arabiriminde (User Network Interface-UNI) bir erişim akış kontrol mekanizmasının sağlanması. Bu işlem başlık kısmındaki bir bit yardımı ile yapılır.

ATM Adaptasyon Katmanı (AAL)

Bu katman, ATM katmanının sunduğu servisi daha da geliştirerek üst katmanlara sunar. Bazı uygulamalarda ATM katmanı tek başına yeterli olmaz. Bu durumda adaptasyon katmanı, ATM katmanına yardımcı olur. Bu katman kullanıcı, kontrol ve yönetim düzlemleri için gerekli fonksiyonları sağlar.

Adaptasyon katmanı iki alt katmana bölünmüştür. Bunlar : Ayrıştırma ve Bitiştirme alt katmanı (Segmentation and Reassembly-SAR) ve Yakınsama alt katmanı (Convergence Sublayer-CS) dir. SAR katmanının ana görevi üst katmandan gelen paketlerin hücrelere bölünmesi ya da ATM katmanından gelen hücrelerin birleştirilerek üst katmana sunulmasıdır. CS alt katmanının görevleri arasında mesaj tanımlama, zaman/saat belirleme gibi işlemler vardır. Genelde CS alt katmanı da kendi içinde ikiye bölünmüştür. Ortak alanlı yakınsama alt katmanı (Common Part Convergence Sublayer-CPCS) ve servise özel yakınsama alt katmanı (Service Specific Convergence Sublayer-SSCS).

Bazı uygulamalarda ATM katmanının sunduğu hizmet yeterli görülebilir. Bu durumda adaptasyon katmanı boş kalabilir yani protokol yapısında boş geçer. ATM adaptasyon katmanı (AAL) yukarıda da belirtildiği gibi, servislere göre farklılıklar

gösterir. Bu nedenle ITU-T tarafından değişik servis sınıflarına göre değişik AAL türleri tanımlanmıştır.

2.3 ATM Ağlarındaki Servisler

ATM ağlarında kullanılan servis tiplerini aşağıdaki biçimde gruptandırmak mümkündür [1]:

- CBR (Constant Bit Rate) Tipi Servisler
- RT-VBR (Variable Bit Rate - Real Time) Tipi Servisler
- NRT-VBR (Variable Bit Rate - Non-Real Time) Tipi Servisler
- ABR (Available Bit Rate) Tipi Servisler
- UBR (Unspecified Bit Rate) Tipi Servisler

1. CBR (Constant Bit Rate) Tipi Servisler : Bu sınıfta sabit bit hızıyla ve genelde gerçek zamanda gerçekleşen uygulamalar yer alır. Bu sınıftaki uygulamalar için hata ve akış kontrolü yapılmaz. En tipik örneği, telefon haberleşmesidir.

2. VBR (Variable Bit Rate) Tipi Servisler : Bu sınıfta bit hızı değişebilen uygulamalar yer alır. İki alt sınıfa ayrılır :

- Gerçek zamanda olan uygulamalar (RT-VBR)
- Gerçek zamanda olmayan uygulamalar (NRT-VBR)

Gerçek zamanda olan uygulamalar (RT-VBR) : Bu tip uygulamalarda hem ortalama hücre gecikmesi, hem de hücre gecikmesindeki varyasyon iyi bir biçimde denetlenmelidir. Aksi takdirde servis kalitesinde bozulmalar olacaktır. Öte yandan sıradan bir bitin veya hücrenin kaybı bu tip uygulamalar için çok önemli değildir. Bu sınıfa örnek; video konferans uygulamasıdır.

Gerçek zamanda olmayan uygulamalar (NRT-VBR) : Bu tip uygulamalarda da gecikme önemlidir, ancak gecikmeye karşı daha toleranslı davranılabilir. Bu sınıfa örnek olarak, multimedya öğeleri içeren elektronik posta uygulaması verilebilir.

3. ABR (Available Bit Rate) Tipi Servisler : Bu sınıftaki uygulamalar; minimum ve maksimum bit hızı değerleri kabaca bilinen, yığınsal(bursty) trafik oluşturması muhtemel uygulamalardır. Bu sınıfa örnek olarak, Web uygulamaları verilebilir.

Sadece ABR sınıfındaki uygulamalar için ağ, tıkanıklık oluştuğunda geri besleme yöntemlerini kullanabilir ve göndericiden yavaşlamasını isteyebilir.

4. UBR (Unspecified Bit Rate) Tipi Servisler : Bu sınıf, iletim hızının sınır değerleri üzerinde bile yorum yapılamayan ve iletim garantisinin bulunmadığı uygulamaları içerir. En tipik örneği, IP protokolünde IP paketlerinin iletilmesidir. Tıkanıklık oluştuğunda UBR hücreleri atılır. Hata ve akış kontrollerini kendileri yapmak isteyen uygulamalar için UBR bir seçenek olabilir. Dosya transferi, elektronik posta gibi gerçek zaman karakteristikleri bulunmayan uygulamalar için UBR düşünülebilir.

Yukarıda anlatılan ATM servis tiplerinin temel karakteristikleri Tablo 2.1’ de verilmiştir [1]. Servis kalitesi (Quality of Service – QoS) de ATM ağlar için önemli bir konudur ve aşağıda bu konuya kısaca değinilmiştir.

Tablo 2.1 ATM Ağlarındaki Servislerin Karakteristikleri

Servis Karakteristikleri	CBR	RT-VBR	NRT-VBR	ABR	UBR
Band Genişliği Garantisi	Evet	Evet	Evet	Seçimli	Hayır
Gerçek Zaman Trafikğine Uygunluk	Evet	Evet	Hayır	Hayır	Hayır
Yığınsal (Bursty) Trafikğe Uygunluk	Hayır	Hayır	Evet	Evet	Evet
Tıkanıklık için Geri Besleme	Hayır	Hayır	Hayır	Evet	Hayır

Servis Kalitesi (Quality of Service – QoS)

Görüntü devre kurulduğunda hem ulaşım katmanı, hem de ATM katmanı servisi tanımlayan sözleşmeye uymak zorundadır. Kaynak ve ağ arasında yapılan sözleşmede üç temel kısım bulunur [1]:

1. Verilecek trafiğin özellikleri
2. Üzerinde anlaşılan servisin özellikleri
3. Uyuma zorunlulukları (compliance requirements)

Ağ ve kaynak sözleşme şartları üzerinde anlaşmaya varamazlarsa, veya ağ istenen servisin özelliklerini sağlayamayacak durumda ise görüntü devre kurulmaz.

Trafik sözleşmesinin ilk kısmı, trafik tanımlayıcısıdır (traffic descriptor). Bu kısım, verilecek yükü karakterize eder. Sözleşmenin ikinci kısmı, istenen ve ağ tarafından kabul edilen servis kalitesini belirtir. Hem trafik yükü, hem de servisin özellikleri ölçülebilir nicelikler cinsinden ifade edilmelidir. Trafik sözleşmesinin üçüncü kısmı ise, ilk iki kısımda kesinlikle uyulması gereken şartların neler olduğunu gösterir.

ATM standartlarında bazı servis kalitesi parametreleri tanımlanmıştır. Her servis kalitesi parametresi için en kötü başarımları tanımlanmıştır ve ağın bunu karşılaması veya aşması gerekir. Parametre en kötü başarımları bazı durumlarda minimum değer için, bazı durumlarda ise maksimum değer için sergiler. Önemli parametrelerden bazıları Tablo 2.2’ de listelenmiştir ancak bu parametrelerin hepsi bütün servis tiplerine uygulanamaz [1].

Tablo 2.2 Bazı Servis Kalitesi (QoS) Parametreleri

QoS Parametresi	Sembolü	Anlamı
Tepe hücre hızı	PCR	Hücrelerin gönderileceği maksimum hız
Sürekli hücre hızı	SCR	Uzun dönemli ortalama hücre hızı
Minimum hücre hızı	MCR	Kabul edilebilir minimum hücre hızı
Hücre gecikmesi varyasyonu toleransı	CDVT	Maksimum kabul edilebilir hücre seyirmesi (jitter)
Hücre kayıp oranı	CLR	Kaybolan veya çok geç teslim edilen hücrelerin oranı
Hücre iletim gecikmesi	CTD	Teslimin ne kadar sürdüğü (ortalama ve maksimum)
Hücre gecikme varyasyonu	CDV	Hücre teslimi sürelerindeki varyasyon
Hücre hata oranı	CER	Hatasız teslim edilen hücrelerin oranı
Ağır hatalı hücre bloklarının oranı	SECBR	Belli bölümleri alınan blokların oranı
Yanlış yerleştirilen hücre oranı	CMR	Yanlış hedefe teslim edilen hücrelerin oranı

3. TIKANIKLIK DENETİMİ

Bu bölümde genel olarak tıkanıklık denetimi konusu ele alınacak ve tüm ağlarda kullanılan tıkanıklık denetim yöntemleri kısaca açıklanacaktır.

3.1 Tıkanıklık Denetiminin Genel İlkeleri

Tıkanıklık denetimi; ağda herhangi bir andaki toplam istek miktarı, ağın toplam taşıyabileceği kapasiteye eşit olduğunda veya bunu aştığında ağın bu durumdan minimum etkilenmesi için yapılması gereken işlerin tamamı olarak ifade edilebilir. Buradan da anlaşılacağı üzere ağlarda tıkanıklık, toplam istek mevcut kaynak kapasitesinden fazla ise meydana gelmektedir.

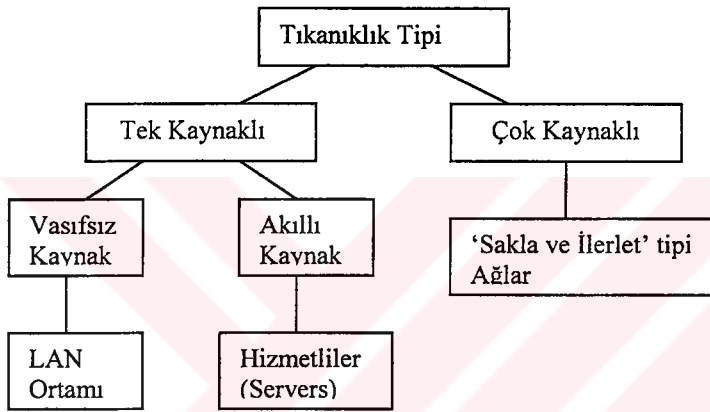
Araştırmalar sonucunda tıkanıklığın dinamik bir problem olduğu ve statik yöntemlerle bu problemin çözülmemeyeceği görülmüştür. Bu nedenle tıkanıklığı gerçek zamanlı bir şekilde, dinamik olarak çözmeye yönelik yöntemler bulunmalıdır.

Tıkanıklık probleminin çözümüne yönelik birçok yöntem önerilmesine rağmen, araştırmalar hala sürmektedir. Bunun temel nedeni ise, tıkanıklık denetiminin sağlanması gereken ve aşılması zor gözüken kriterlerin bulunmasıdır [2]. Bunların başlıcaları :

- Kullanılan tıkanıklık denetim yöntemi, sisteme minimum düzeyde yük getirmelidir.
- Kullanılan tıkanıklık denetim yöntemi kullanıcılar arasında adaleti sağlamalıdır.
- Kullanılan tıkanıklık denetim yöntemi, ağın mevcut kaynak kapasitesi ve trafik yükü arasında dinamik bir ilişki kurarak sistemde gerekli düzenlemeleri yapabilmelidir.
- Kullanılan tıkanıklık denetim yöntemi, sistemin içinde bulunduğu en kötü koşullarda bile verimli çalışabilmelidir.

3.2 Tıkanıklık İle İlgili Problemlerin ve Çözümlerin Sınıflandırılması

Tıkanıklıkla ilgili problemler, en genel olarak ağı oluşturan kaynakların sayısına göre sınıflandırılabilir (Şekil 3.1). Bu durumda sınıflandırma, tek kaynağa bağlı problemler ve dağıtılmış birden fazla kaynağa bağlı problemler olarak yapılabilir [2]. Tek kaynak kavramı da, vasıfsız kaynak ve akıllı kaynak olmak üzere ikiye ayrılabilir. Asıl tıkanma probleminin yaşandığı ağ tipleri, dağıtılmış kaynak yapısında olan ‘sakla ve ilerlet’ tipi ağlardır. Daha sonraki bölümlerde bu tip ağlardan olan ATM ağlarındaki tıkanıklık problemlerinden ve bunların çözüm yöntemlerinden bahsedilecektir.

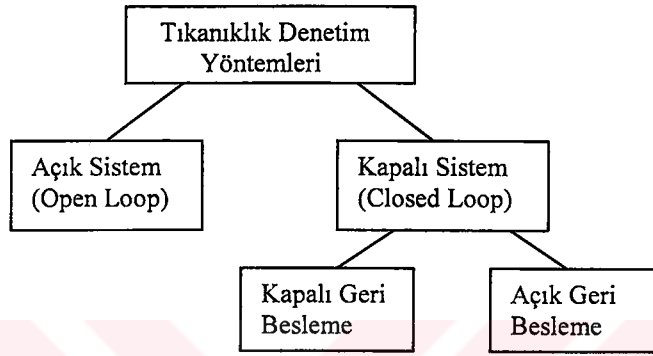


Şekil 3.1 Tıkanıklık Problemlerinin Sınıflandırılması

Tıkanıklık probleminin çözümü için ortaya konan denetim yöntemleri ise, açık sistem ve kapalı sistem denetim yöntemleri olmak üzere iki grupta toplanabilir (Şekil 3.2) [1]. Açık sistem denetim yöntemlerinde; ne zaman yeni trafik kabul edileceği, hangi paketlerin hangi öncelikle ve ne zaman ağdan atılacakları gibi tıkanıklığı önlemeye yönelik yöntemler, sistemin o anki mevcut durumundan bağımsız olarak, daha önceden belirlenmiş algoritmalara göre yürütülür. Kapalı sistem tıkanıklık denetim yöntemleri ise, geri besleme kavramı ile, sistemin o anki durumunun göz önüne alınmasının sağlandığı çözüm yöntemleridir. Bu tür yöntemlerde aşağıdaki adımlar sıra ile uygulanır :

1. Sistemin tıkanıklık olasılığına karşı sürekli gözlenmesi ve tıkanıklığın oluştuğu yerin tespiti.
2. Bu bilginin gerekli önlemlerin alınacağı düşünülen muhtemel bölgelere gönderilmesi.
3. Tıkanıklık problemini çözmek için uygun bir yöntem kullanarak, sistemin normal çalışma konumuna döndürülmesi.

Kapalı sistem tıkanıklık denetim yöntemleri de, açık geri beslemeli ve kapalı geri beslemeli olmak üzere ikiye ayrılır. Açık geri beslemeli yöntemlerde, tıkanıklık durumunda sorunlu bölgeden kullanıcılara uyarı paketi gönderilir. Kapalı geri beslemeli yöntemlerde ise, kullanıcı gözlemlerde bulunarak tıkanıklığı sezer. Örneğin onaylama paketlerinin zamanında gelmemesinden, tıkanıklık oluştuğunu anlar.



Şekil 3.2 Tıkanıklık Denetim Yöntemlerinin Sınıflandırılması

3.3 Ağlarda Kullanılan Tıkanıklık Denetim Yöntemleri

3.3.1 Tıkanıklığı Önleyen Yöntemler

Bu yöntemler, tıkanıklık oluşma olasılığını en aza indirmek amacıyla tasarlanmış yöntemlerdir. Bunu gerçekleştirmek amacıyla ağın çeşitli seviyelerinde uygun stratejiler (policies) kullanılır. Şekil 3.3' de veri bağı, ağ ve ulaşım katmanlarındaki çeşitli stratejiler gösterilmiştir [2].

Veri bağı katmanında yer alan tekrar iletim stratejisi, göndericinin ne kadar hızlı zaman aşımına (time out) uğrayabileceği, ve zaman aşımı durumunda ne ileteceği ile ilgilidir. Cep bellekte tutma stratejisi, tekrar iletim stratejisi ile ilgilidir. Alıcılar bütün sıra dışı paketleri atarsa, bu paketlerin daha sonra tekrar iletilmesi gerekir. Bu da ek yük getirir. Onaylama stratejisi de tıkanıklığı etkiler. Her paket için onaylama gönderilirse onaylama paketleri çok fazla trafik oluşturur.

Ağ katmanında bağlantı mekanizmasının görüntü devre veya datagram olması tıkanıklığı etkiler. Çünkü tıkanıklık denetim algoritmalarının çoğu sadece görüntü devre alt ağlarda çalışır. Paket kuyuklama ve servis stratejisi, yönlendiricinin giriş ve çıkış hatları için sahip olduğu kuyruk yapılarıyla ilgilidir. Ayrıca bu strateji,

işlenecek paketlerin sırasıyla da ilgilidir. Paket atma stratejisi, yer olmadığında hangi paketin atılacağını söyleyen kuraldır. Kötü paket atma stratejisi, tıkanıklığın daha kötü bir hal almasına neden olabilir.

Katman	Stratejiler
Ulaşım Katmanı	• Tekrar İletim Stratejisi • Sıra dışı paketlerin cep bellekte tutulması • Onaylama Stratejisi • Akış Denetimi Stratejisi • Zaman Aşımının Belirlenmesi
Ağ Katmanı	• Bağlantı Mekanizması (Görüntü Devre veya Datagram) • Paketlerin Kuyruklanması ve Servis Stratejisi • Paket Atma Stratejisi • Yönlendirme Algoritması • Paket Ömrü Kontrol Stratejisi
Veri Bağı Katmanı	• Tekrar İletim Stratejisi • Sıra dışı paketlerin cep bellekte tutulması • Onaylama Stratejisi • Akış Denetimi Stratejisi

Şekil 3.3 Tıkanıklığı Etkileyen Stratejiler

Yönlendirme algoritması trafiği bütün hatlara yayarak tıkanıklığın önlenmesine yardımcı olabilir. Bununla birlikte kötü bir yönlendirme algoritması tıkalı durumda olan hatlar üzerine daha fazla trafik yollayabilir. Paket ömrü kontrol stratejisi ise, bir paketin atılmadan önce ne kadar uzun yaşayabileceği ile ilgilidir. Bu süre çok uzun olursa kayıp paketler ağdaki çalışmalara engel olabilir. Bu süre çok kısa olursa bu kez paketler hedefe ulaşmadan zaman aşımına uğrayabilirler. Bu da, tekrar iletimlere neden olur.

Ulaşım katmanındaki sorunlar, veri bağı katmanındakiler ile aynıdır. Ancak bu katmanda zaman aşımı aralığının belirlenmesi daha zordur. Çünkü ağ katmanında iki düğüm arasındaki geçiş süresini tahmin etmek daha zordur.

3.3.2 Trafik Biçimlendirme

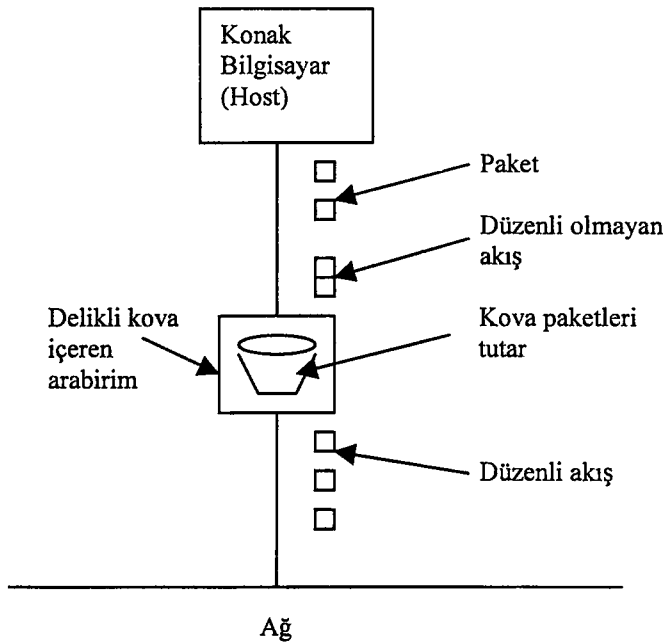
Bu yöntem, ortalama veri iletme hızının ve yığınsallığın belli bir algoritmaya dayalı olarak düzenlenmesi esasına dayanır. En yaygın kullanıma sahip algoritmalar ise “delikli kova”, “sanal delikli kova” [3] ve “jetonlu kova (token bucket)” dır.

Tüm bu algoritmaların ortak özelliği, veri iletiminin önceden belirlenmiş bir hızda seyretmesini sağlamaya yönelik olmalarıdır. Böylece ağda bir nevi akış denetimi sağlanmış olur.

3.3.2.1 Delikli Kova Algoritması

Bu algoritmada her konak bilgisayar (host) ile ağ arasında bir arabirim vardır. Bu arabirim, “delikli kova” olarak adlandırılan sonlu kapasiteki bir kuyruk içerir. Kuyruk dolu olduğunda gelen paketler atılacaktır. Bu yapı, donanımsal olarak bir arabirim ile gerçekleştirilebilir veya konak bilgisayarın işletim sistemi tarafından simule edilebilir.

Delikli kova algoritmasında konak bilgisayarın her saat darbesinde ağa bir paket yollamasına izin verilir. Bu işlem de, arabirim kartı ile veya işletim sistemi tarafından gerçekleştirilebilir. Bu mekanizmayla, konak bilgisayara düzensiz olarak gelen paketler ağa düzenli bir biçimde verilmiş olur. Böylece yığınsallıklar düzeltilmiş ve tıkanıklık olasılığı büyük ölçüde azaltılmış olur (Şekil 3.4).



Şekil 3.4 Delikli Kova Algoritması

Bütün paketlerin boyları aynı ise, algoritma yukarıda anlatılan biçimde yürütülür. Ancak değişken boyutlu paketler kullanıldığında her saat darbesinde bir paket yerine, sabit sayıda byte'ın ağı gönderilmesine izin vermek genelde daha iyi sonuçlar verir [1].

Asıl delikli kova algoritmasını uygulamak kolaydır. Delikli kova sonlu kapasiteli bir kuyruktan oluşur ve kuyrukta yer olmadığında paket atılır. Kuyruk boş olmadığı sürece her saat darbesinde bir paket iletilir.

Byte-sayan delikli kova algoritması, hemen hemen aynı biçimde uygulanır. Her saat darbesinde sayıcıya n değeri atanır. Kuyruktaki ilk paketin byte sayısı, sayıcının o anki değerinden daha az ise bu paket iletilir ve sayıcının değeri byte sayısı kadar azaltılır. Sayıcının değeri yeterli büyüklükte olduğu sürece ek paketler gönderilebilir. Sayıcı, kuyrukta bulunan bir sonraki paketin uzunluğundan daha küçük değer alırsa iletim bir sonraki saat darbesine kadar durur [1].

3.3.2.2 Jetonlu Kova Algoritması

Delikli kova algoritması, trafiğin yığınsallığını göz önüne almadan çıkış hızının ortalama bir değer almasını sağlar. Birçok uygulama için, büyük yığınlar geldiğinde çıkışın hızlanmasına olanak sağlamak daha iyidir. Bu yüzden daha esnek yöntemler gerekir. Jetonlu kova algoritması bu tür yöntemlerden biridir. Bu algoritmada delikli kova, saat tarafından her ΔT saniyede bir üretilen jetonları tutar.

Jetonlu kova algoritması, delikli kova algoritmasından daha farklı bir trafik biçimlendirmesi sağlar. Jetonlu kova algoritmasında n paketlik yığınlar bir kerede gönderilebilir. Böylece kovanın girişine gelen ani yığınlara daha hızlı cevap verilmiş olunur. Burada n , en fazla kovanın büyüklüğü kadar olabilir.

İki algoritma arasındaki diğer bir fark da şudur : Jetonlu kova algoritmasında kova dolunca jetonlar atılır, gelen paketler atılmaz. Buna karşın delikli kova algoritmasında kova dolunca gelen paketler atılır.

Jetonlu kova algoritması da, delikli kova algoritmasına benzer şekilde değiştirilebilir. Yapılan bu değişiklikle her jeton, gönderilebilecek bir paketi değil, k tane byte'ı temsil eder. Bir paket ancak, paketin uzunluğunu kapsayacak kadar jeton varsa iletilir.

Basit jetonlu kova algoritmasının uygulanması için sadece jetonları sayan bir değişken yeterlidir. Sayıcı her ΔT saniyede bir 1 artırılır ve bir paket gönderildiğinde 1 azaltılır. Jetonlu kovanın byte'ları sayacak şekilde değiştirilmiş şeklinde ise, sayıcı

her ΔT saniyede bir k byte kadar artırılır ve gönderilen her pakette paketin uzunluğu kadar azaltılır [1].

3.3.3 Akış Belirtme

Bu yöntemde, ağ ile kullanıcı arasında bağlantı kurulmadan veya datagramlar gönderilmeden önce, kaynak ağdan onay almak için ağa bir akış tarifnamesi (flow specification) yollar. Bu tarifname, ağa verilecek trafik ile ilgili bilgileri (maksimum paket büyüklüğü, maksimum iletim oranı v.b.) ve istenen servisin niteliği (minumum gecikme, maksimum gecikme değişimi, hücre kayıp duyarlılığı v.b.) ile ilgili bilgileri içerir. Ağ bunu kabul edebilir, red edebilir veya bazı parametreler için alternatif önerilerde bulunabilir. Gönderici ve ağ anlaşmaya varırsa, gönderici alıcıya bu şartları kabul edip etmediğini sorar.

Şekil 3.5' de Partridge (1992) tarafından tasarlanan örnek bir akış tarifnamesi gösterilmiştir [1]. Bu örnek akış tarifnamesinin ilk sütunundaki bilgiler, ağa verilecek trafik ile ilgilidir. Maksimum paket boyu, paketlerin ne kadar büyüklükte olabileceğini söyler. Daha sonraki iki parametreden, trafiğin jetonlu kova algoritması ile biçimlendirildiğinin varsayıldığı anlaşılmaktadır. Bu parametreler, saniyede kaç byte' ın jetonlu kovaya konulacağını ve kovanın büyüklüğünün ne kadar olduğunu gösterir. Hız r byte/s, kovanın boyutu b byte ise, rastgele bir Δt zaman aralığında gönderilebilecek maksimum byte sayısı $b + r \Delta t$ ' dir. Maksimum iletim hızı, konak bilgisayarın herhangi bir koşul altında üretebileceği en büyük hızdır.

Giriş Karakteristikleri	İstenen Servis
Maksimum paket boyu (byte)	Kayıp duyarlılığı (byte)
Jetonlu kovanın hızı (byte/s)	Kayıp aralığı (μs)
Jetonlu kovanın boyutu (byte)	Yığın kayıp duyarlılığı (paket)
Maksimum iletim hızı (byte/s)	Bildirilen minumum gecikme (μs)
	Maksimum gecikme varyasyonu (μs)
	Garanti kalitesi

Şekil 3.5 Örnek Akış Tarifnamesi

İkinci sütundaki bilgiler, uygulamanın ağdan istediklerini gösterir. Minimum gecikme, bir paketin ne kadar gecikebileceğini belirtir. Maksimum gecikme varyasyonu, gecikmeye çok duyarlı olmayan ancak seyirmeye çok duyarlı olan uygulamalar içindir. Seyirme, uçtan uca paket iletim süresi içindeki varyasyon miktarıdır. Garanti kalitesi ise, uygulamanın gerçekten ne isteyip istemediğini

belirtir. Örneğin bazı uygulamalar için gecikme veya kayıp karakteristikleri karşılanamazsa bunun çok fazla önemi olmayabilir. Öte yandan bazı uygulamalar için bunların karşılanması o kadar önemlidir ki, bu koşullar karşılanmazsa uygulama sona erebilir.

3.3.4 Görüntü Devre Alt Ağlarda Tıkanıklık Denetimi

Yukarıda anlatılan tıkanıklık denetim yöntemleri açık sistem temelli yöntemlerdir. Bu yöntemler, tıkanıklığı oluşmadan önlemeye çalışırlar. Bu bölümde, görüntü devre alt ağlardaki tıkanıklığı dinamik olarak denetleyen bazı yaklaşımlar anlatılacaktır.

Tıkanıklığı dinamik olarak denetleyen tekniklerden biri, yaygın olarak kullanılan kabul kontrolüdür [4]. Bu yöntemin dayandığı fikir oldukça basittir. Tıkanıklık fark edildiğinde, problem çözülünceye kadar yeni bir görüntü devrenin kurulmasına izin verilmez. Bu yöntem basit ve gerçekleştirmesi kolay bir yöntemdir.

Alternatif bir yaklaşım, yeni görüntü devrelerin kurulmasına izin vermek ancak problemlili bölgeleri göz önüne alarak bütün yeni görüntü devreler için yönlendirmeleri dikkatlice yapmaktır.

Görüntü devrelerle ilgili diğer bir yöntem ise, bir görüntü devre kurulacağı zaman konak bilgisayar (host) ile alt ağ arasında sözleşme yapılmasıdır. Bu sözleşme; trafiğin şeklini, istenen servis kalitesini ve diğer parametreleri belirtir. Alt ağ, devre kurulacağı zaman yol boyunca kaynakları ayırır. Böylece yeni görüntü devreler üzerinde tıkanıklık meydana gelmeyecektir. Çünkü bütün gerekli kaynaklar mevcuttur.

Bu tip kaynak ayırma her zaman yapılabilir, veya sadece alt ağda tıkanıklık oluştuğunda yapılabilir. Bu tip kaynak ayırmayı her zaman yapmakla, ağ kaynaklarından etkili bir biçimde yararlanmamış oluruz.

3.3.5 Uyarı Paketleri

Tıkanıklık durumunda ağ, kullanıcıya “uyarı paketi” (choke packet) denilen özel bir paket göndererek, kullanıcıdan trafik yükünü belli bir yüzde kadar düşürmesini isteyebilir.

Her yönlendirici, çıkış hatlarının ve diğer kaynaklarının kullanımını kolayca gözleyebilir. Bir çıkış hattının kullanımı belirli bir eşiği aştığında çıkış hattı “uyarı” durumuna girer. Yeni gelen her paket, çıkış hattının uyarı durumunda olup olmadığını görmek için denetlenir. Paketin çıkış hattı uyarı durumunda ise

yönlendirici kaynağa, içinde hedefin adresinin bulunduğu uyarı paketi yollar. Esas paket, yol boyunca daha fazla uyarı paketinin gönderilmesine yol açmayacak şekilde işaretlenir.

Kaynağın uyarı paketi aldığı anda, belirtilen hedefe yolladığı trafiği X oranında azaltması gerekir. Ayrıca kaynak sabit bir zaman aralığı için, aynı hedefi gösteren uyarı paketlerini ihmal etmelidir. Bu aralık sona erdikten sonra, kaynak aynı hattın hala tıkanık olduğuna dair uyarı paketi alırsa hızını biraz daha azaltır ve bu hedeften gelen uyarı paketlerini tekrar ihmal etmeye başlar. Aralık sona erdikten sonra aynı hedefe ilişkin uyarı paketi gelmezse kaynak tekrar hızını arttırabilir. Bu yöntemdeki geri besleme, tıkanıklığı önlemeye yardımcı olur [1].

Bu tıkanıklık denetim algoritmasının çeşitli değişik şekilleri önerilmiştir. Düğümde Düğüme Uyarı Paketi (hop by hop choke packet), adil kuyruklama (fair queueing), ve belli ağırlıkta adil kuyruklama (weighted fair queueing) bu yöntemlerden bazılarıdır.

3.3.5.1 Ağırlıklı Adil Kuyruklama (Weighted Fair Queueing)

Uyarı paketleri yöntemindeki temel problem, kaynakların bu paketleri aldıklarında istedikleri şekilde davranabilmeleridir. Örneğin bir düğüm dört kaynağa uyarı paketi yolladığında bunlardan birisi paket gönderimini azaltabilir, diğer üçü ise aynı oranda paket göndermeye devam edebilir. Bu durumda sadece düğümün uyarısına uyan kaynak band genişliği kullanımını azaltmış olacaktır. Bu problemin üstesinden gelmek için Nagle (1987), “adil kuyruklama” (fair queueing) algoritmasını önermiştir.

Adil kuyruklama algoritmasına göre düğümlerin her çıkış hattı ve her kaynak için birer tane kuyruğu vardır. Hat boşaldığında düğüm kuyruklarını tarar ve bir sonraki kuyruktaki ilk paketi alır. Örneğin bir çıkış hattı için n tane kaynağın yarışması durumunda, bu çıkış hattından iletilen n tane paketin her biri başka bir kaynağa ait olacaktır. Daha fazla paket yollamak bu oranı değiştirmez. Bazı ATM anahtarları bu algoritmayı kullanır [1].

Bu algoritmada büyük paket kullanan kaynaklara, küçük paket kullanan kaynaklardan daha fazla band genişliği ayrılır. Bu problemin üstesinden gelmek için Demers (1990) bir iyileştirme önermiştir. Bu algoritmada kuyruklar, her paketin sonu bulununcaya kadar sürekli taranır. Paketler daha sonra bitişlerine göre sıralanır ve her sanal saat işaretinde paketlerin birer byte' ı, bu sıra ile gönderilir [1].

Bu algoritmadaki sorunlardan biri, bütün kaynaklara aynı önceliğin verilmesidir. Birçok durumda dosya hizmetlisine veya diğer hizmetlilere istekçilerden daha fazla band genişliği verilmesi istenir. Her saat işaretinde bazı kaynakların paketlerinden bir byte'dan fazlası gönderilerek bu sağlanabilir. Bu algoritma “ağırlıklı adil kuyruklama” (weighted fair queueing) olarak adlandırılır ve yaygın olarak kullanılır. Ağırlık, görüntü devrelerin sayısına veya makineden gelen akışa eşit olabilir. Böylece her sürece aynı band genişliği ayrılmış olur.

3.3.5.2 Düğümünden Düğüme Uyarı Paketi (Hop-by-Hop Choke Packets)

Yüksek hızlarda ve uzak mesafelerde kaynağa uyarı paketi yollamak iyi sonuç vermeyebilir. Bu nedenle, uyarı paketinin geçtiği her düğümü etkilediği bir yöntem düşünülmüştür. Bu yöntemde uyarı paketi bir düğüme ulaştığında, bu düğümünden uyarı paketinin geldiği düğüme olan akış azaltılır. Bunu yapmak için uyarı paketinin alındığı düğümde daha fazla tampon kullanmak gerekir. Çünkü kaynak aynı hızla paket yollamaya devam etmektedir. Akışın azaltılması işlemi, uyarı paketi kaynağa doğru ilerlerken yol boyunca bütün düğümler arasında uygulanır. Uyarı paketi kaynağa ulaştığında ise kaynak paket gönderme hızını azaltır.

Düğümünden düğüme uyarı paketi yöntemi tıkanıklıktan daha çabuk kurtulmayı sağlar ancak daha fazla tampon kullanımını gerektirdiğinden maliyeti artırır [1].

3.3.6 Yük Atma

Bu yöntem genelde ağların en son başvurduğu çözüm şeklidir. Ağın tıkanıklık problemini yukarıdaki yöntemlerden biriyle çözemediği durumda, düğümlerin üzerlerine gelen yükü, diğer bir deyişle paket veya hücreleri atmasıdır.

Daha esnek bir çözüm, paket atma işinde öncelik uygulamasına gidilmesi ve önceliği yüksek paketlerin belli bir süreye kadar ağ üzerinde tutulmaya devam edilmesi şeklinde olabilir. Bu yöntem klasik, önceliksiz paket atma yöntemine göre daha iyi sonuç verir.

Öncelik sınıfları, trafik biçimlendirme ile birleştirilebilir. Örneğin, jetonlu kova algoritması için şu şekilde bir kural tanımlanabilir. Bir paket jeton yokken gelirse bu paket gönderilir, fakat bir sorun olduğunda atılmak üzere düşük öncelikli olarak işaretlenir. Diğer bir seçenek, görüntü devre kurulunca kaynağın sözleşmede

belirtilen sınırları aşmasına izin vermektir. Ancak sözleşmeyi aşan trafiğin tamamı düşük öncelikli olarak işaretlenir [1].

Paketleri düşük veya yüksek öncelikli olarak işaretlemek, bir veya daha fazla başlık bitinin kullanılmasını gerektirir. Bu amaç için ATM hücrelerinin başlık alanlarında 1 bit ayrılmıştır, dolayısıyla bütün ATM hücreleri düşük veya yüksek öncelikli olarak işaretlenir. ATM anahtarları atma kararlarında bu biti kullanırlar.

Bazı ağlarda paketler, tekrar iletmede kullanılmak üzere daha büyük birimler halinde gruplandırılır. Örneğin ATM ağlarda sabit boyutlu paketler hücre olarak adlandırılır ve hücreler mesajların parçalarını oluşturur. Bir hücre atıldığı zaman sadece kaybolan hücre değil, eninde sonunda tüm mesaj tekrar iletilecektir.

3.3.7 Seyirme Denetimi (Jitter Control)

Vidyo gibi uygulamalarda iletim süresi sabit olduğu sürece, paketlerin gecikme süresi çok önemli değildir. Ancak paket gecikmeleri arasında büyük farklar olması, ses veya görüntü kalitesinin bozulmasına neden olur. Bu yüzden sözleşmede paket gecikmelerinin değer alacağı aralık tanımlanabilir. Bunun için ortalama değer iyi seçilmelidir. Diğer bir deyişle paket gecikmelerinin ortalama değerinin içinde ortalama tıkanıklık miktarı da hesaba katılmalıdır.

Seyirme, yol boyunca her düğüm için beklenen geçiş süresi hesaplanarak sınırlandırılabilir. Yönlendiriciye bir paket ulaştığında yönlendirici, paketin kendi programının ne kadar ilerisinde veya gerisinde olduğunu kontrol eder. Bu bilgi pakette saklanır ve her düğümde güncellenir. Paket programın önündeyse, programa uymasını sağlamak amacıyla belli bir süre düğümde bekletilir. Paket programın gerisindeyse yönlendirici paketi mümkün olan en kısa sürede geçirir. Aslında aynı çıkış hattı için bekleyen paketlerden daima programın en gerisinde olan seçilecektir. Böylece programın gerisinde olan paketler hızlandırılır, programın ilerisinde olan paketler yavaşlatılır. Her iki durumda seyirmenin miktarı azalır [1].

3.3.8 Çoklu Göndermede Tıkanıklık Denetimi

Buraya kadar anlatılan tıkanıklık denetim yöntemlerinin tamamında tek bir kaynak ve tek bir alıcı vardır. Bu bölümde, çoklu kaynaklardan çoklu alıcılara olan akışın kontrolü anlatılacaktır. Bu akışa örnek olarak vidyo konferans uygulaması verilebilir.

Birçok çoklu gönderme uygulamasında gruplar üyeliklerini dinamik olarak değiştirirler. Bu koşullar altında göndericiler için band genişliği ayırmak, etkin bir yöntem değildir. Çünkü bu yöntemde her göndericinin bütün kayıtları taraması ve her değişiklikte ağacı yeniden oluşturması gerekir.

3.3.8.1 Kaynak Ayırma Protokolu

Kaynak ayırma protokolu, birden fazla göndericinin birden fazla alıcıya iletimde bulunmasına olanak sağlar. Aynı zamanda band genişliğinin optimum olarak kullanılmasını sağlar ve tıkanıklığı yok eder.

Bu protokolun en basit şeklinde ağaçlar kullanılarak çoklu yönlendirme yapılır. Her alıcı grubuna bir grup adresi atanır. Bu gruba paket göndermek isteyen gönderici, paketlerin içine grubun adresini koyar. Daha sonra standart çoklu yönlendirme algoritması tarafından, bütün grup üyelerini kapsayan bir ağaç oluşturulur. Yönlendirme algoritması kaynak ayırma protokolunun bir parçası değildir. Bu yöntemin normal çoklu gönderme yönteminden tek farkı, gruplara periyodik olarak ek bilgi gönderilmesidir. Bu bilgi, ağaç üzerindeki yönlendiricilerin saklamaları gereken veri yapılarını içerir.

Bu yöntemde tıkanıklığı önlemek için gruptaki alıcılardan herhangi biri göndericiye ağaç üzerinden bir ayırma mesajı gönderebilir. Bu mesaj ilerletme algoritmaları ile göndericiye kadar taşınır. Her düğümde ayırma kaydedilir ve gerekli band genişliği ayrılır. Yeterli band genişliği yoksa, bu durum başarısızlık şeklinde geriye rapor edilir. Ayırma isteği gerçekleştirildiğinde, göndericiden alıcıya giden yolun tamamında gerekli band genişliği ayrılmış olur [1].

Bir ayırma işlemi yapılırken alıcı, istediği bir veya daha fazla kaynağı belirtebilir. Ayrıca alıcı bu seçiminin ayırma süresi boyunca değişip değişmeyeceğini söyleyebilir. Yönlendiriciler bu bilgiyi band genişliği kullanımını optimize ederken kullanırlar. Örneğin iki alıcı için ortak yol ancak, bu alıcılar kaynaklarını daha sonra değiştirmeyeceklerse kurulabilir.

4. ATM TEMELLİ AĞLARDA TIKANIKLIK DENETİMİ

Bu bölümde, ATM temelli ağlarda tıkanıklığın denetlenmesi için kullanılan yaklaşımlardan bazıları tanıtılacaktır.

ATM ağlar, hem sistemin kotarabileceğinden daha fazla trafiğin gelmesinden kaynaklanan uzun dönemli tıkanıklıkla, hem de trafikteki yığınsallıktan kaynaklanan kısa dönemli tıkanıklıkla başa çıkabilmelidir. Bunun için de, farklı birçok yöntem birlikte kullanılır. Bu yöntemlerin en önemlileri üç grup altında toplanabilir [1]:

- Kabul Denetimi Yöntemleri
- Kaynak Ayırma Yöntemi
- Hız Temelli Tıkanıklık Denetimi

Bundan sonraki bölümlerde bu yöntemler incelenecektir.

4.1 Kabul Denetimi Yöntemleri

Düşük hızlı ağlarda tıkanıklığı denetlemek için tıkanıklığın gerçekleşmesini beklemek, tıkanıklık gerçekleşince de paketleri yollayan kaynağa yavaşlamasını söylemek genelde geçerli bir yöntemdir. Yüksek hızlı ağlarda ise bu yöntem yetersiz kalır. Çünkü kaynağa haber gönderilmesi ve haberin kaynağa ulaşması sırasında kaynaktan binlerce paket gelebilir.

Daha da önemlisi birçok ATM ağında gerçek zaman trafik kaynakları vardır. Böyle bir kaynağa yavaşlamasını söylemek işe yaramayabilir. Bu nedenlerden ötürü ATM ağlarda ilk aşamada, tıkanıklığın gerçekleşmeden önlenmesi düşünülmüştür.

Tıkanıklığın önlenmesi için kullanılan en önemli yöntemlerden biri de kabul denetimidir. Konak bilgisayar (host) yeni bir görüntü devre istediğinde, görüntü devre üzerinden yollayacağı trafiği ve beklediği servisi belirtmek zorundadır. Bu durumda ağ, varolan bağlantıları etkilemeden bu bağlantıyı kotarmasının mümkün olup olamayacağını kontrol eder, uygun bir yön bulmak için potansiyel yönler incelenir. Hiç yön bulunamazsa çağrı red edilir.

Red etme işlemi adil bir biçimde yapılmalıdır. Hiçbir denetim uygulanmazsa, az sayıdaki yüksek band genişliği kullanıcısı çok sayıdaki düşük band genişliği kullanıcısını ciddi bir biçimde etkileyebilir. Bunu önlemek için kullanıcılar, kullarımlarına dayalı olarak sınıflara ayrılmalıdır. Servisin red edilme olasılığı bütün sınıflar için aşağı yukarı aynı olmalıdır. Bu, her sınıfa ayrı bir kaynak havuzu ayrılarak sağlanabilir [1].

4.2 Kaynak Ayırma Yöntemi

Kabul denetimine benzer bir teknik de, kaynakların önceden, genelde çağrının kurulması aşamasında ayrılması tekniğidir. Trafik tanımlayıcısı tepe hücre hızını belirttiğinden ağ bu hızı kotarmak için gerekli band genişliğini yol boyunca ayırabilir. Band genişliğinin ayrılması, üzerinden geçtiği her hat boyunca band genişliği ayıran SETUP mesajı ile yapılabilir. Ancak bir hat boyunca ayrılan toplam band genişliği, hattın kapasitesinden daha az olmamalıdır. SETUP mesajı dolu bir hatta vurduğunda geri dönmeli ve alternatif başka bir yol aranmalıdır.

Trafik tanımlayıcısı sadece tepe band genişliğini içermez, aynı zamanda ortalama band genişliğini de içerir. Konak bilgisayar örneğın 100 000 hücre/s tepe band genişliği ve 20 000 hücre/s ortalama band genişliği için istekte bulunabilir. Bu durumda teorik olarak beş tane devre aynı fiziksel hat (trunk) üzerinden çoğullanabilir. Beş tane bağlantının hepsi belli bir süre boyunca boş olabilir, daha sonra hepsi tepe hızlarında ilettime başlayabilir. Bu da çok sayıda hücrenin kaybedilmesine neden olur. VBR trafiğı istatistiksel olarak çoğullanabileceğinden bu servis sınıfında sorunlar meydana gelir. Bu problemin çözümü için çalışmalar devam etmektedir [4].

4.3 Hız Temelli Tıkanıklık Denetimi

CBR ve VBR trafiklerinde bilgi kaynaklarının gerçek zaman veya yarı gerçek zaman doğasından ötürü, tıkanıklık durumunda bile kaynakların yavaşlaması mümkün değildir. UBR trafiğinde ağda çok sayıda hücre olması durumunda hücreler atılır.

ABR trafiğinde ise tıkanıklık durumunda ağın bir veya daha fazla sayıda göndericiye işaret göndermesi ve onlardan ağ kendini toparlayana kadar geçici bir süre yavaşlamalarını istemesi mümkündür.

ATM standartları hazırlanırken ABR trafiğı için tıkanıklığın nasıl belirleneceğı ve denetleneceğı, göndericiye nasıl işaret gönderileceğı hakkında çok sayıda yöntem önerilmiştir.

İlk önerilen yöntemde göre; kullanıcı bir veri yığını göndermek istediğinde önce gerekli band genişliğini ayıran özel bir hücre göndermek zorundadır. Onaylama (ACK) geldikten sonra yığın başlayabilir. Bu yöntemin avantajı, tıkanıklığın hiçbir zaman olmamasıdır. Çünkü veri göndermeye başlamadan önce gerekli band genişliği ayrılmıştır. ATM Forumu bu çözümü, konak bilgisayarın iletme geçmeden önce uzun süre bekleme ihtimali olduğu için red etmiştir.

İkinci önerilen yöntemde, tıkanıklık oluşmaya başlar başlamaz geriye doğru “uyarı” (choke) hücreleri gönderen anahtarlar vardır. Böyle bir hücreyi alan gönderici, hücre gönderme hızını yarıya indirir. Tıkanıklık ortadan kalktıktan sonra hızın tekrar artırılması için çeşitli yöntemler önerilmiştir. Bu yöntem de red edilmiştir. Çünkü tıkanıklık durumunda uyarı hücreleri kaybolabilir. Ayrıca bu yöntem küçük band genişliği kullanıcıları için adil değildir.

Üçüncü önerilen yöntemde, paket sınırlarının son hücredeki bir bit ile gösterilmesinden yararlanılmıştır. Buradaki düşünce tıkanıklığı gidermek için hücrelerin atılması, ancak bunun mümkün olduğunca seçici bir biçimde yapılmasıdır. Anahtar tıkanıklığı gidermek amacıyla gelen hücre katarını paket sonu için tarar ve bir sonraki paketteki hücrelerin hepsini atar. Atılan bu paket daha sonra tekrar iletilmelidir. Bu yöntemde aynı paketten k tane hücrenin atılması bir tane paketin tekrar iletilmesini gerektirir. Oysa rastgele k tane hücrenin atılması, k tane paketin tekrar iletilmesine yol açabilir. Bu yöntem adaleti sağlamaz. Çünkü atılan hücrelerin bulunduğu paket, anahtarı aşırı yüklemeye soğan göndericiye ait olmayabilir. Ayrıca bu yöntem standartlaştırılmaz. Çünkü tıkanıklık oluştuğunda hangi hücrelerin atılacağı anahtardan anahtara değişebilir.

Uzun tartışmalar sonunda ABR trafiğinde tıkanıklığın denetlenmesi için iki yöntem benimsenmiştir :

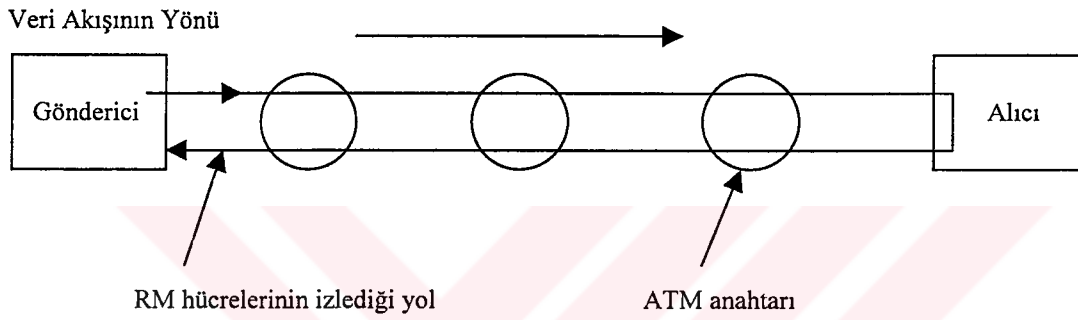
- Kredi Temelli Çözüm (Kung ve Morris, 1995)
- Hız Temelli Çözüm (Bononi ve Fendick, 1995)

Kredi temelli çözüm aslında dinamik kayan pencere protokoludur. Bu yöntem her anahtarın her bir görüntü devre için bir kredi (bu görüntü devre için ayrılan tamponların sayısı) tutmasını gerektirir. İletilen her hücre için bir tampon olduğu sürece tıkanıklık oluşmayacaktır [5].

Kredilerin hesabını tutmak için bütün hesaplama işlemlerinin anahtarda yapılmasını ve çok sayıda tampon ayrılmasını gerektirdiği için bu yöntem karşı çıkmıştır. Bu

yöntemde maliyetin çok fazla olacağı düşünülmüş ve alternatif olarak aşağıda çalışması anlatılan hız temelli tıkanıklık denetim yöntemi önerilmiştir [6, 7].

Hız temelli tıkanıklık denetim yönteminde her gönderici her k tane veri hücresinden sonra özel bir RM (Resource Management) hücresi yollar. Bu hücre, veri hücreleriyle aynı yol üzerinde ilerler, ancak yol üzerinde bulunan anahtarlar bu hücreye farklı biçimde davranırlar. RM hücresi hedefe doğru ilerlerken incelenir, güncellenir, daha sonra da göndericiye geri yollanır. RM hücrelerinin izlediği yol Şekil 4.1’ de gösterilmiştir.



Şekil 4.1 RM Hücrelerinin İzlediği Yol

Ayrıca bu yöntemle iki farklı tıkanıklık denetim mekanizması gerçekleştirilir. İlk olarak aşırı yüklü anahtarlar kendiliğinden RM hücrelerini üretebilirler ve onları göndericiye geri yollayabilirler. İkinci mekanizmaya göre de, aşırı yüklü anahtarlar göndericiden alıcıya giden veri hücrelerinin orta PTI bitini değiştirebilirler. Ancak bu yöntemlerin hiçbirisi tam olarak güvenilir değildir. Çünkü tıkanıklık durumunda hiç kimseye ulaşmadan bu hücreler kaybolabilir. Kayıp bir RM hücresi, belirlenen bir zaman aralığı içinde geri dönmeyince gönderici tarafından fark edilecektir. CLP biti ABR tıkanıklık denetimi için kullanılmaz.

ABR tıkanıklık denetimi şu şekilde gerçekleştirilir : Her kullanıcının o anki hızı ACR (Actual Cell Rate), MCR (Maximum Cell Rate) ve PCR (Peak Cell Rate) hızları arasındadır. Tıkanıklık oluştuğunda ACR, MCR dan daha küçük olmamak üzere azaltılır. Tıkanıklık yoksa ACR arttırılır ancak PCR ın üzerine çıkmasına izin verilmez. Gönderilen her RM hücresi, göndericinin iletimde bulunmak istediği hızı (PCR veya daha düşük bir hız) içerir. Bu hız ER (Explicit Rate) olarak adlandırılır. RM hücresi alıcıya doğru çeşitli anahtarlar üzerinden ilerlerken, tıkanık durumda olan anahtarlar ER nin değerini azaltabilirler. Hiçbir anahtar bu değeri arttıramaz. Azaltma işlemi ileri veya geri yönde gerçekleşebilir. Gönderici RM hücresini geri

aldığında yol üzerindeki bütün anahtarlar tarafından kabul edilebilecek maksimum hızı (ER) görür. Eğer gerekliyse, ACR değerini bu hıza eşit olacak biçimde değiştirir.

Orta PTI bitini kullanan tıkanıklık mekanizması RM hücreleri ile birleştirilmiştir. Bu biti son veri hücresinden alan alıcı, geri gönderilen her RM hücresiyle bu biti de gönderir. Bu bit RM hücresinden alınamaz. Çünkü bütün RM hücrelerinde bu bitin değeri her zaman aynı olacak biçimde ayarlanmıştır.

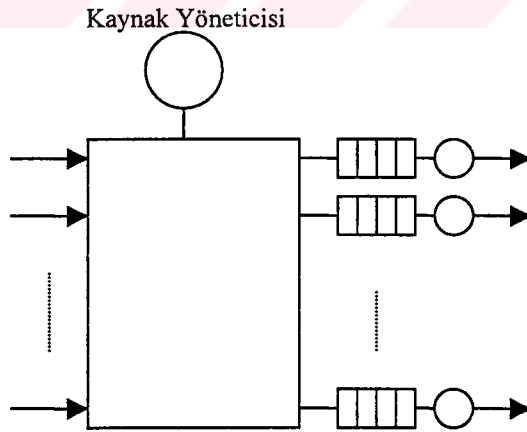


5. ÇAĞRI KABUL DENETİMİ YÖNTEMLERİ

5.1 Çağrı Kabul Denetimi (Call Admission Control-CAC)

ATM bağlantı yönelimli bir hizmettir ve kullanıcı ATM ağı üzerinden iletme başlamadan önce bağlantı kurulmak zorundadır. Bu işlem çağrı kurulma aşamasında yapılır. Bu işlemin temel amacı, gönderici ile alıcı arasında bir yol kurmaktır. Bu yol, bir veya daha fazla ATM anahtarını içerebilir. Yeni bir bağlantı için bu ATM anahtarlarının her birinde kaynak tahsis edilecektir.

Çağrı kurma işlemi kaynak yöneticisinde yürütülür. Kaynak yöneticisi, anahtara eklenen bir iş istasyonudur (Şekil 5.1) ve anahtarın çalışmasını denetler. Yeni bağlantıları kabul eder, eski bağlantıları düşürür ve diğer yönetim işlemlerini gerçekleştirir. Yeni bağlantı isteği kabul edilmişse, bu bağlantı için band genişliği ve/veya tampon tahsis edilir. Tahsis edilen kaynaklar, bağlantı sona erdiğinde bırakılır.



Şekil 5.1 Çıkış Tamponlu bir ATM Anahtarı.

Çağrı Kabul Denetimi, bir anahtarın yeni bir bağlantı isteğini kabul edip edemeyeceği ile ilgilidir. Yeni bir bağlantı isteğinin kabul veya red edilmesi kararı, aşağıdaki sorulara dayalı olarak verilir :

- Yeni bağlantı o an anahtarın taşıdığı bağlantıların servis kalitelerini (QoS) etkiler mi ?
- Yeni bağlantının gerektirdiği servis kalitesi sağlanabilir mi ?

Çağrı kabul denetimi, istatistiksel olmayan çoğullamada ve istatistiksel çoğullamada olmak üzere iki sınıfa ayrılır. İstatistiksel çoğullama için iyi çağrı kabul yöntemleri tasarlamak oldukça zordur [8].

5.1.1 İstatistiksel Olmayan Çoğullamada Çağrı Kabul Denetimi

İstatistiksel olmayan çoğullamada, kaynak için çıkış portunda daima tepe hızı kadar band genişliği ayrılır. İstatistiksel olmayan çoğullama, PCM-kodlu ses, çok düşük band genişliği uygulamaları, kodlanmamış video gibi CBR tipi servisler için kullanılır.

İstatistiksel olmayan çoğullamanın olumlu tarafı, yeni bir bağlantının kabul edilip edilmeyeceğine karar vermenin kolay olmasıdır. Bu kararı verebilmek için yeni bağlantının sadece tepe hızı bilgisi yeterlidir. Varolan bütün bağlantıların tepe hızları ile yeni bağlantının tepe hızı toplamı çıkış hattının (link) kapasitesinden daha az ise yeni bağlantı isteği kabul edilir.

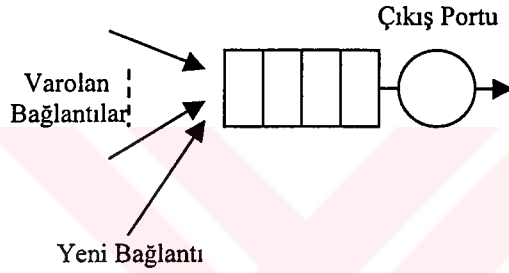
İstatistiksel olmayan çoğullamanın olumsuz tarafı ise, bağlantılar tepe hızlarında iletimde bulunmadıkları sürece çıkış hattından büyük ölçüde faydalanılmamış olacaktır [9].

5.1.2 İstatistiksel Çoğullamada Çağrı Kabul Denetimi

İstatistiksel çoğullamada yeni bağlantı için tahsis edilen band genişliği, kaynağın tepe hızından daha azdır. Bütün bağlantıların tepe hızlarının toplamı, çıkış hattının kapasitesinden daha büyük olabilir. İstatistiksel çoğullama yığınsal kaynaklarda ekonomik kazanç sağlar ancak istatistiksel çoğullamayı etkili bir biçimde gerçekleştirmek oldukça zordur. Bu zorluk, varış sürecinin karakterize edilmesindeki ve bu sürecin ATM ağının içerisinde nasıl şekil alacağını anlamadaki güçlükten kaynaklanır.

İstatistiksel çoğullama için çağrı kabul denetimi algoritması tasarlamadaki diğer bir güçlük, çağrının kabul edilip edilmeyeceğine karar verme işleminin hızlı yapılması zorunluluğudur. Bu zorunluluktan dolayı, çağrı kabul denetimi algoritmaları basit olmalıdır.

Yeni bir çağrının kabul edilip edilmeyeceğine karar verme işlemi, bir kuyruklama problemi olarak ele alınabilir. Şekil 5.1’ de gösterilen ATM anahtarı göz önüne alınırsa, çağrı kabul denetimi algoritması her çıkış portunun tamponuna uygulanmalıdır. Bir çıkış portu ve onun tamponu anahtarın geri kalanından ayrılırsa Şekil 5.2’ de gösterilen kuyruklama modeli elde edilir. Bu tip kuyruklama yapısı ATM çoğullayıcı olarak bilinir ve bir çıkış portunun hizmet verdiği, sonlu kapasiteli bir kuyruğu besleyen ATM kaynaklarını temsil eder. Hizmet süresi sabittir ve bir ATM hücresini iletmek için geçen süreye eşittir. Varolan bağlantılar için servis kalitesinin (QoS) sağlandığı varsayılırsa, çağrı kabul algoritmasında sorulması gereken soru, yeni bağlantı eklendiğinde servis kalitesinin devam edip etmeyeceğidir [9].



Şekil 5. 2 Bir ATM Çoğullayıcı

Bu soru, ATM çoğullayıcı varolan bağlantılar ve yeni bağlantı için çözülerek cevaplanabilir. Ancak bu problemin çözümü oldukça zordur ve fazla sayıda CPU işlemi gerektirir. Karmaşık varış süreçleri göz önünde bulundurulduğunda problemin çözümü daha da zorlaşır. Bu yüzden bu tip bir kuyruklama probleminin çözümünü gerektirmeyen, farklı yaklaşımlara dayalı çeşitli yöntemler önerilmiştir.

Çağrı kabul denetimi algoritmaları için aşılması gereken diğer bir sorun da video kaynaklarıdır. Video kaynağının modellenmesi konusundaki araştırmalar hala devam etmektedir.

5.2 Bir Varış Sürecinin Karakterize Edilmesi

ATM ağları için yapılan ilk başarımlı çalışmalarında, varış süreçlerinin Poisson veya Bernoulli dağılımı ile dağıldığı varsayılmıştır. ATM hücresi sabit uzunlukta olduğu için zaman eksenini slotlara bölerek hücre varışını Bernoulli dağılımına göre modellemek mümkündür. Bu modelde her slotun, bir hücrenin iletimini tamamlamasına yetecek kadar uzun olduğu varsayılmıştır. Slotlara bölünmüş zaman eksenine bakıldığında, her slot bir hücre içerebilir veya içermeyebilir. Bir slotun

hücre içirme olasılığının $p < 1$, boş olma olasılığının $(1 - p)$ olduğu varsayılırsa, ardışıl iki varış arasındaki zaman geometrik olarak dağılır ve her zaman birimindeki varışların sayısı Bernoulli dağılımına göre dağılır. Bernoulli süreci, Poisson sürecinin ayrık zaman eşdeğeridir.

Poisson veya Bernoulli trafik modellerine dayalı ilk başarımların çalışmalarından, bu trafik modellerinin veri dosyası taşınması, paketlenmiş kodlanmış video gibi uygulamalar sonucu trafikte oluşan yığılma yakalayamadığı anlaşılmıştır. Bu yüzden de IPP (Interrupted Poisson Process) veya onun ayrık zaman eşdeğeri IBP (Interrupted Bernoulli Process) gibi ON-OFF tipi modeller kullanılmaya başlanmıştır.

IPP modelinde, varışların Poisson dağılımına göre gerçekleştiği aktif bir periyodu, hiç varışın olmadığı boş bir periyod izler. Bu iki periyodun süreleri ise üssel dağılıma göre değişir. IBP modeli de IPP modeline benzer şekilde tanımlanmıştır. Ancak aktif periyod sırasındaki varışlar Bernoulli dağılımına göre gerçekleşir. Aktif ve boş periyodun süreleri ise geometrik dağılıma göre değişir. IPP veya IBP modelinde ardışıl varışlar arasındaki süreler birbirinden bağımsız olduğu için, bu modeller korelasyon notasyonunu yakalayamazlar. Bir ATM kaynağını modellemek için kullanılan diğer bir ON-OFF modeli de IFP (Interrupted Fluid Process) modelidir. Bu modele göre aktif periyod sırasındaki varışlar sürekli bir hızda meydana gelir [9].

ATM trafiği karakterize edilirken, belli bir kaynaktan gelen hücrelerin varışları arasındaki sürelerin çok bağlantılı olabileceği görülmüştür. Bunun sonucu olarak da, ATM trafiğini modellemek için daha karmaşık dağılımlar kullanılmıştır. Bu dağılımlar; MMPP (Markov Modulated Poisson Process), MMPP'nin ayrık zaman karşılığı olan MMBP (Markov Modulated Bernoulli Process) ve MMFP (Markov Modulated Fluid Process) dir. MMPP, farklı durumları olan bir Markov sürecidir. Her durumda varışlar, duruma bağlı bir hızda ve Poisson dağılımına göre gerçekleşir. MMBP ve MMFP modelleri benzer şekilde tanımlanmıştır. Ancak durumlardaki varışlar, MMBP modelinde Bernoulli dağılımına göre, MMFP modelinde ise sürekli bir hızda meydana gelir. IPP, IBP ve IFP ON-OFF modelleri, MMPP, MMBP ve MMFP Markov modellerinin özel halleridir [9].

MMPP, MMBP ve MMFP modellerinin altında yatan varsayımlardan biri, varış sürecinin her durumda harcadığı sürenin üssel (veya geometrik) olarak dağılmasıdır. Bu varsayım, matematiksel açıdan uygunluk sağlamak amacıyla yapılmıştır ve böyle bir varsayımın yapılmış olmasının fazla önemi yoktur. Çünkü bu modeller üssel dağılım yaklaşımından çok daha önemli olan iki faktörü, yığılma ve korelasyonu yakalayabilen modellerdir. Ancak bu varsayımın dosya transferi gibi uygulamalar

için gerçekçi bir varsayım olmadığı düşünülmektedir. Bu durumda yığınsal bir veri kaynağının, on ve off periyodlarının rastgele dağıldığı IBP gibi bir ON-OFF süreci ile karakterize edilmesi gerektiği sonucu ortaya çıkar [9]. VISTAnet' in ATM trafik çalışmasında, on periyodu sabit olan bir ON-OFF trafik modelinden bahsedilmiştir. On ve off periyodları rastgele dağıtılmış bir ATM çoğullayıcının davranışını analiz etmek oldukça zordur.

Vidyodan kaynaklanan trafiği karakterize etmek için de çeşitli modeller önerilmiştir. Video kaynağının modellenmesi konusundaki araştırmalar devam etmektedir.

ATM Forum, ATM trafiği için uygun bir model seçebilmek için aşağıdaki parametreleri standartlaştırmaya karar vermiştir :

- Tepe hızı
- Ortalama hız
- Tepe hızı için hücre gecikme varyasyonu
- Maksimum yığın uzunluğu

Tepe hızı ve hücre gecikme varyasyonu kullanılarak, kaynağın tepe hızı etkili bir biçimde kontrol altında tutulabilir. Ayrıca maksimum yığın uzunluğundan yararlanılarak hücre gecikme varyasyonu tahmin edilebilir. Band genişliği ayrılmasında bu parametreler son derece yetersiz kalır. Çünkü tepe hızı, ortalama hızı ve maksimum yığın uzunluğu aynı, fakat yığınsallığı ve varışlar arası korelasyonu farklı olan dağılımlar vardır. Yığınsallık ve korelasyon, hücre kayıp olasılığı gibi servis kalitesi ölçütlerini çok fazla etkileyen iki parametredir [9].

Varış sürecinin bir trafik modeli ile tam olarak karakterize edildiği varsayıldığında bir sonraki adımda sorulması gereken soru; çeşitli anahtarlar ve çoğullayıcılar aracılığı ile ağda ilerledikçe kaynağın yığınsallığının ve varışlar arasındaki sürenin korelasyonunun nasıl değişeceğidir. Ağ içerisinde ilerledikçe kaynağın yığınsallığı azalırsa, kaynak için ne kadar band genişliği ayrılacağına karar vermek kolaydır. Ancak ağ içerisinde kaynak daha yığınsal hale geliyorsa, bu kararı vermek çok daha zor olacaktır. Bu, henüz tam olarak çözülmemiş bir problemdir.

5.3 Çağrı Kabul Denetimi Yöntemleri

Bu bölümde, çok sayıdaki çağrı kabul denetimi yöntemlerinden bazıları tanıtılacaktır.

Bu bölümde tanıtılan yöntemler aşağıdaki biçimde gruplandırılabilir :

- Eşdeğer Band Genişliği Yaklaşımı
- Hücre Kayıp Olasılığının Üst Sınırı
- Hızlı Tampon / Band Genişliği Tahsisi
- Zaman Pencereleeri

Bu gruplandırma, yöntemi geliştirmekte kullanılan prensiplere dayalı olarak yapılmıştır. Sonraki bölümlerde her grubun temel özellikleri ve gruplarda yer alan çağrı kabul denetim yöntemlerinden bazıları anlatılacaktır.

5.3.1 Eşdeğer Band Genişliği Yaklaşımı

Tek bir kaynağın veya birden fazla kaynağın eşdeğer band genişliği, çağrı kabulünde çok yaygın bir notasyondur. Sonlu kapasiteli bir kuyruğu besleyen tek bir kaynak ele alındığında kaynağın eşdeğer kapasitesi, kuyruğun hücre kayıp olasılığı ε ' a karşı düşen servis hızıdır.

Tek bir kaynağın eşdeğer kapasitesi, Guerin, Ahmadi ve Naghshineh tarafından aşağıdaki biçimde türetilmiştir [10].

Her kaynağın IFP tipi kaynak olduğu varsayılmıştır. R kaynağın tepe hızı, r kaynağın aktif olduğu zaman bölümü, b aktif periyodun ortalama süresi olarak alınırsa; bir IFP kaynağı (R, r, b) vektörü ile tam olarak karakterize edilebilir. Kaynağın servis süresi sabit olan, sonlu kapasiteli bir kuyruğu beslediği varsayılmıştır. Kapasitesi x olan kuyruk için Anick, Mitra ve Sondhi' nin kullandığı teknik kullanılarak kuyruk uzunluğu dağılımı elde edilir. Bu dağılımdan, hücre kayıp olasılığı ε ' a karşı düşen servis hızı c' yi hesaplamak mümkündür. Sonuç olarak eşdeğer band genişliği, diğer bir deyişle eşdeğer kapasite c, aşağıdaki biçimde bulunmuştur [10] :

$$\varepsilon = \beta \exp \left\{ -\frac{x(c - rR)}{b(1 - r)(R - c)c} \right\}$$

Burada

$$\beta = \frac{(c - rR) + \epsilon r(R - c)}{(1 - r)c}$$

dir. Yukarıdaki eşitlik c için çözülerek eşdeğer kapasite c elde edilir. Ancak yukarıdaki denklemlerden çözüm, sayısal hesaplamalarla bulunabilir. β ' nın değeri 1 alınarak (genelde $\beta < 1$ dir) basitleştirme yapılırsa :

$$c = \frac{a - x + \sqrt{(a - x)^2 + 4xar}}{2a} R \quad (5.1)$$

dir. Burada $a = \ln(1/\epsilon)$ b $(1-r) R$ dir.

N tane kaynak olması durumunda, x kapasiteli tampon için eşdeğer band genişliği, yine hücre kayıp olasılığı ϵ ' u sağlayan servis hızı olacaktır. Ancak bu durumda eşdeğer band genişliğinin hesabı oldukça karmaşık bir hale gelir. Guerin, Ahmadi ve Naghshineh aşağıdaki yaklaşılmayı önermişlerdir [10]:

$$C = \min \left\{ m + \alpha \sigma, \sum_{i=1}^N c_i \right\} \quad (5.2)$$

Burada

c_i : i. kaynağın (5.1) 'e göre hesaplanan eşdeğer band genişliği, $\sum_{i=1}^N c_i$ ise bütün kaynakların eşdeğer band genişliklerinin toplamıdır.

m : Kaynakların ortalama bit hızlarının toplamıdır ($m = \sum_{i=1}^N m_i$). m_i ise i. kaynağın ortalama bit hızıdır.

σ : Kaynakların varyanslarının kareköklerinin toplamıdır ($\sigma = \sum_{i=1}^N \sigma_i$). σ_i^2 ise, i. kaynağın bit hızının varyansıdır. $\sigma_i^2 = m_i(R_i - m_i)$.

(5.2)' deki α değeri ise

$$\alpha = \sqrt{-2\ln(\epsilon) - \ln(2\pi)} \quad \text{şeklinde hesaplanır.}$$

Bu yaklaştırma iki gözleme dayalı olarak yapılmıştır. İlk olarak, çoğullanmış N tane kaynağın eşdeğer band genişliği, bütün kaynakların eşdeğer band genişlikleri toplamından daha azdır. İkincisi, N tane kaynağın stasyonier bit hızının yaklaşık olarak ortalaması m, varyansı σ^2 olan Normal dağılımı izlediği gözlenmiştir. Sonlu kapasiteli kuyruğun tamponu olmadığı varsayılırsa (yani $x = 0$), N tane kaynak için eşdeğer band genişliği, Normal dağılım $N(m, \sigma^2)$ içindeki bir nokta olacaktır. Bu nokta, standart sapmalarla $m + \alpha\sigma$ şeklinde ifade edilebilir. Buradaki α değeri ise, Normal dağılımın tanımlamalarından elde edilir. Böylece N tane kaynağın eşdeğer band genişliği, (5.2) ile verilen iki farklı eşdeğer band genişliğinin minimumuna eşit olacaktır. Bu ifade, gerçek band genişliği gereksinimleri için makul bir üst sınır olarak kabul edilir [10].

Elwalid ve Mitra, MMFP kaynağın eşdeğer band genişliğinin yaklaşık olarak, kaynağın parametreleri, çoğullayıcı kaynakları ve hücre kayıp olasılığından türetilen matrisin maksimum reel özdeğerine eşit olduğunu göstermişlerdir. Bir trafik kaynağının L tane durum ile modellendiği, Q matrisinin varış sürecinin durumları arasındaki geçişi yöneten Markov zincirinin üreticisi olduğu, ve Markov zincirinin bütün durumları için varış hızlarının $\vec{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_L)$ vektörü ile gösterildiği varsayılırsa bir kaynağın eşdeğer band genişliğinin

$$\Lambda - \frac{1}{\xi} Q$$

matrisinin maksimum reel özdeğeri olduğu gösterilmiştir [11]. Burada $\Lambda = \text{diag}(\vec{\lambda})$, ve $\xi = \ln(\epsilon)/x$ dir. Ayrıca N tane kaynak çoğullandığında, bunların eşdeğer band genişliğinin asimtotik olarak $c = \sum_{i=1}^N c_i$ ' e eşit olduğu gösterilmiştir. Burada c_i i. kaynağın eşdeğer band genişliğidir ve bu değer, kaynak sistemdeki tek kaynakmış gibi hesaplama yapılarak bulunur [11].

Yapılan bazı çalışmalarda, eşdeğer band genişliği yöntemlerinin belli durumlar için yanlışlıkları net olarak gösterilmiştir. Rege, eşdeğer band genişliği temelli çeşitli yöntemleri karşılaştırmış, ve bu yöntemlerin doğruluğunu arttırmak için bazı değişiklikler önermiştir [12]. Elwalid, eşdeğer band genişliği yönteminin eksiklerini gidermek amacıyla, eşdeğer band genişliği yaklaşımı ile Chernoff sınır yaklaşımını

birleştiren bir yöntem önerilmiştir [13]. Ancak bu yöntem, eşdeğer band genişliği yaklaşımının bütün eksiklerini gidermez.

Chang ve Thomas, çoğullayıcının çıkışında kaynağın eşdeğer band genişliğini değerlendirmek için bir yöntem tanıtmışlardır [14]. De Veciana, Kesidis ve Walrand, eşdeğer band genişliğinin “on-line” değerlendirilmesini önermişlerdir [15]. Duffield, trafik kaynaklarının ve onların eşdeğer band genişliklerinin karakterize edilmesi için “maksimum entropi” adlı bir yöntem önermiştir [16].

5.3.2 Hücres Kayıp Olasılığının Üst Sınırı

Hücre kayıp olasılığının üst sınırını temel alan birçok kabul yöntemi önerilmiştir. Saito, sabit bir aralıkta gelen ortalama hücre sayısına (ANA) ve aynı aralıkta gelen maksimum hücre sayısına (MNA) dayalı olarak bir üst sınır önermiştir [17]. Bu yöntemde sabit aralık, kabul edilebilir maksimum gecikme D' nin yarısına eşittir. Bu parametreler kullanılarak aşağıdaki üst limit türetilir. N tane bağlantıyı besleyen bir hat (link) için; $p_i(j)$ ($i=1,2,...,N$, ve $j=0,1,...$) $D/2$ periyodu sırasında i . bağlantıya ait j tane hücrenin gelme olasılığını göstermek üzere, hücre kayıp olasılığı CLP aşağıdaki biçimde sınırlandırılabilir :

$$CLP \leq B(p_1, \dots, p_N; D/2) = \frac{\sum_{k=0}^{\infty} [k - D/2]^+ p_1 * \dots * p_N(k)}{\sum_{k=0}^{\infty} k p_1 * \dots * p_N(k)}$$

Burada * konvolusyon işlemini göstermektedir. $\theta_i(j)$ fonksiyonu aşağıdaki biçimde tanımlanırsa :

$$\theta_i(j) = \begin{cases} ANA_i / MNA_i & j = MNA_i \\ 1 - ANA_i / MNA_i & j = 0 \\ 0 & \text{diğer} \end{cases}$$

CLP için :

$$CLP \leq B(p_1, \dots, p_N; D/2)$$

$$\leq B(\theta_1, \dots, \theta_N; D/2)$$

$$= \frac{\sum_{k=0}^{\infty} [k - D/2]^+ \theta_1 * \dots * \theta_N(k)}{\sum_{k=0}^{\infty} k \theta_1 * \dots * \theta_N(k)}$$

elde edilir.

Bulunan $B(\theta_1, \dots, \theta_{N+1}; D/2)$ değeri, kabul edilebilir hücre kayıp olasılığından daha az ise yeni bağlantı isteği kabul edilir. Saito, $\theta_1 * \theta_2 * \dots * \theta_N$ ' i hesaplamak için bir yöntem önermiştir. Ayrıca $D/2$ periyodu sırasında gelen hücrelerin sayısının ortalama ve varyansına dayalı olarak farklı bir üst limit elde etmiştir [17].

Ayrıca Rasmussen [18], ve Elwalid, Mitra ve Wentworth [19] tarafından farklı üst limit yöntemleri geliştirilmiştir.

5.3.3 Hızlı Tampon / Band Genişliği Tahsisi

Bu yöntem yığınsal kaynakların iletimi için tasarlanmıştır. Bu yöntemin arkasındaki temel düşünce şudur :

Bir görüntü devre kurulduğunda ağ aracılığı ile bir yol kurulur ve yönlendirme tabloları uygun biçimde güncellenir. Fakat görüntü devreye hiçbir kaynak tahsis edilmez. Kaynak, yığınsal bir veriyi iletmeye hazır olduğu anda, ağ yığın süresince gerekecek kaynakları tahsis etmeye çalışır. Aşağıda, bu biçimde çalışan iki yöntem incelenmiştir :

Tranchier, Boyer, Rouaud, ve Mazeas, tepe bit hızı hattın kapasitesinden %2 daha az olan VBR kaynaklar için hızlı band genişliği tahsisi protokolu önermişlerdir [9]. Bu yöntemde kaynak, “artış” ve “azalış” adımlarıyla band genişliği isteğinde bulunur. Her görüntü devre için istenen toplam band genişliği, sıfır ile tepe hızı arasında değişebilir. Görüntü devre artış adımıyla özel bir “ayırma isteği hücresi” kullanır. İstenen toplam trafik hat kapasitesini aşmıyorsa istenen artış düğüm tarafından kabul edilir. Artış adımı kabul etme veya red etme kararı, tepe band genişliğine dayalı olarak verilir. Artış adımı, görüntü devrenin yolu üzerindeki bir düğüm tarafından red edilirse artış adımıyla engel olunur.

Azalış adımları, bir “yönetim hücresi” aracılığı ile duyurulur ve azalış adımı daima kabul edilir. Tepe hücre hızı o an kabul edilen band genişliğine karşı düşecek biçimde, gelen hücreler hücre seviyesinde şekillendirilir. Bu işle ilgili yönetim

hücrelerini kotarmak için “hızlı ayırma protokol” (Fast Reservation Protocol – FRP) birimi kullanılır. Bu birim, kullanıcı-ağ arabirim (UNI) noktalarına yerleştirilir.

Protokol, güvenilir çalışmayı sağlamak için farklı tiplerdeki zamanlayıcılardan yararlanır. FRP birimi bir artış adımı isteği aldığı anda, bu isteği yol üzerindeki ilk düğüme yönlendirir. Daha sonra bu istek bir sonraki düğüme yönlendirilir, ve yönlendirme işlemi bu şekilde devam eder. İstek, yol üzerindeki bütün düğümler tarafından karşılanabilirse, en son düğüm FRP birimine bir onaylama (acknowledgment-ACK) gönderir. FRP birimi de, uç birimi isteğin kabul edildiğine dair bilgilendirir ve ayırmayı onaylamak için yol üzerindeki düğümlere onaylama hücreleri gönderir.

Bir düğüm isteği karşılayamazsa isteği red eder. Band genişliğini zaten ayırmış olan üst düzeydeki düğümler, sabit bir zaman periyodu içerisinde (örneğin zamanlayıcının ömrü bitene kadar) onaylama hücreleri alamazlarsa ayırmayı iptal ederler. Zamanlayıcı, FRP birimi ile en uzak düğüm arasındaki maksimum gidiş-geliş’e eşit olacak şekilde ayarlanır. İstek red edilirse FRP birimi, farklı bir zamanlayıcı tarafından belirlenen sürenin sonunda artış adımı isteğini tekrarlar. Tekrarların sayısı sınırlıdır.

Turner, band genişliği yerine tampon boşluğu ayıran bir hızlı ayırma yöntemi önermiştir [9]. Bu yöntemde kaynaklar, hat kapasitesine yakın büyük tepe hızlarına sahip olabilirler. Her düğüm her görüntü devre için iki durumlu bir durum makinesi tutar. Durum makinesinin durumları ise “aktif durum” ve “boş durum” olarak adlandırılır. Bir görüntü devre aktif durumda iken, hattın tamponunda önceden belirlenmiş sayıda yer ayrılır. Durum makinesinin geçişleri, başlangıç ve bitiş hücreleri alınınca gerçekleşir. Başlangıç hücreleri yığının başladığını, bitiş hücreleri ise yığının bittiğini gösterir. Yığındaki başlangıç hücreleri ile bitiş hücreleri arasındaki bütün hücreler orta hücreler olarak işaretlenir. Hücreler ayrıca CLP biti değiştirilerek işaretlenebilir. Tampon dolduğunda CLP biti değiştirilmiş olan işaretli hücre atılır.

Bu yöntemde her düğümün aşağıdaki bilgileri saklaması gerekir :

- Her görüntü devre i için, görüntü devrenin o anki durumu (boş veya aktif),
- Görüntü devre aktif hale geldiğinde ayrılacak olan, önceden tanımlanmış tampon slotu sayısı s_i ,
- Tampondaki i. görüntü devreye ait olan, işaretlenmemiş hücrelerin sayısı u_i ,
- Tamponda kullanılmayan slotların toplam sayısı K'

Bir önceki yöntemden farklı olarak bu yöntemde, kaynağın iletimi sırasında istek/onaylama prosedürü kullanılmaz. Kaynak, doğrudan başlangıç hücresi olarak işaretlenmiş hücreyi ve sonraki diğer hücreleri iletmeye başlar. Düğüm ise gelen hücrenin başlangıç hücresi olduğunu anladığı anda, hücreler için önceden belirlenmiş sayıda tampon slotunu ayırıp ayıramayacağına karar verir. Görüntü devre boş durumda ve $s_i > K'$ ise, yığındaki başlangıç hücresi ve sonraki diğer hücreler atılır. Öte yandan görüntü devre boş durumda ve $s_i \leq K'$ ise, düğüm yığını kabul eder. Görüntü devrenin durumu aktif olarak değiştirilir, bu görüntü devre için bir zamanlayıcı ayarlanır ve s_i değeri, K' değerinden çıkarılır. u_i değeri s_i' den küçük ise u_i 1 arttırılır. u_i değeri s_i' ye eşitse hücre işaretlenir (CLP biti değiştirilir) ve tamponda tutulur. Zamanlayıcı hücre gecikme varyasyonu ile belirlenir. Orta veya son hücre varmadan önce zamanlayıcının ömrü biterse, görüntü devrenin durumu boş olarak değiştirilir.

Hücreleri işaretlemek, i. görüntü devreden s_i den daha fazla sayıda hücre kabul etmeye olanak verir. Ancak i. görüntü devre için sadece s_i tane tampon slotu ayrılır (örneğin sadece s_i tane hücre işaretsiz olabilir). Geri kalan hücreler işaretlidir ve diğer görüntü devrelerden yeni yığınlar gelirse ve tampon dolarsa bu hücreler atılabilir. Bu da tamponun adil bir biçimde paylaşılmasını sağlar. Tampon ayırma mekanizması aynı biçimde CBR kaynaklara da uygulanabilir.

R bir görüntü devrenin tepe hızı, C hat kapasitesi, K tampon boyutu olmak üzere görüntü devre için ayrılan tampon slotları :

$$s_i = \lceil KR / C \rceil$$

ifadesiyle bulunabilir. Yeni bir görüntü devre içeren bir yön seçileceği zaman, yeni görüntü devrenin varolan görüntü devrelerle güvenli bir biçimde kullanılabileceğinden emin olmak gerekir.

Doshi ve Heffes de, uzun dosya iletimleri için bir hızlı tampon ayırma yöntemi önermişlerdir [20].

5.3.4 Zaman Pencereleeri

Sabit bir zaman periyodu içerisinde kaynakların ileteceği bit (veya hücre) sayısını sınırlayan çeşitli bağlantı kurma yöntemleri önerilmiştir. Bu zaman periyodu, çerçeve veya zaman penceresi olarak adlandırılır. Bu yöntemler, “kayan pencereler” yöntemine benzer.

Golestani, ağda her bağlantı için herhangi bir hat üzerinden iletilen hücre sayısının sınırlı olduğu bir yöntem önermiştir. Böylece ağ boyunca düzgün trafik akışı korunur. Bu ise, çerçeve olarak adlandırılan sabit zaman periyodu ile gerçekleştirilir. Her bağlantı için, bir çıkış hattı üzerinden her çerçevede iletilen hücre sayısının belli bir üst sınırı aşmadığı bir mekanizma önerilmiştir. Böylece bütün bağlantılar tarafından aynı hat üzerinden iletilen toplam hücre sayısı üstten sınırlanmış olur. Bu mekanizmada bir çerçeve içinde giriş portuna gelen bir hücrenin, komşu çerçevenin sonunda anahtardan iletileceği garanti edilir [21].

Vakil ve Singh, düğümden düğüme bir akış denetim mekanizması önermişlerdir. İletimde bulunan düğüm, her bağlantı için, her sabit zaman periyodunda sadece belli sayıda hücre iletebilir. İletilecek hücrelerin sayısı, alıcı düğüm tarafından krediler kullanılarak belirlenir. Alıcı düğüm ileten düğümü, sabit bir zaman periyodu içinde her bağlantı için kaç tane kredi kullanabileceği konusunda bilgilendirir. Belli bir bağlantı için ayrılan krediler zaman periyodu sona ermeden tüketilirse, zaman periyodunun geri kalan kısmında bu bağlantıya ait hücre iletilmez. Alıcı düğüm kredilerin sayısını dinamik olarak değiştirebilir. Bu yöntem tamponlama gerektirir [9].

5.4 Eşdeğer Band Genişliği Temelli Kabul Kriterleri

5.4.1 Guerin-Ahmadi-Naghshineh (GAN) Kriteri

Guerin, Ahmadi ve Naghshineh, kuyruk uzunluğu dağılımından yararlanarak, bir grup bağlantının eşdeğer kapasitesini veren tek bir ifade elde etmişlerdir [10]. Bir hat (link) üzerinden çoğullanan bir grup bağlantının eşdeğer kapasitesi, istenen servis kalitesini, örneğin tampon taşma olasılığını sağlamak için gerekli band genişliği miktarı olarak tanımlanır. Bu değer, her bir bağlantının karakteristiğinin ve tampon gibi ağ kaynaklarının bir fonksiyonudur.

Guerin, Ahmadi ve Naghshineh tarafından geliştirilen bu yöntemde, iki farklı yaklaşım birleştirilmiştir. İlk yaklaşım, “fluid-flow” kaynak modeline dayanır. İkinci yaklaşım ise, bir hat üzerindeki stasyoner bit hızı dağılımına dayanır. İlk yaklaşım, her bir bağlantının karakteristiğinin etkisinin önemli olduğu durumlarda iyi sonuçlar verir. İkinci yaklaşım ise istatistiksel çoğullamanın etkisinin büyük olduğu durumlar için düşünülmüştür. Her iki yaklaşım, bağlantıların farklı yönlerini ele aldığı ve birbirini tamamladığı için bu iki yaklaşım birleştirilmiştir.

A. “Fluid – Flow” Yaklaşımı :

Bu yaklaşım “Fluid-Flow” kaynak modeline dayanır. “Fluid-Flow” kaynak modelinde çoğullanan bağlantıların ürettiği bit hızı, yoğunluğu Markov zincirinin durumuna göre değişen, sürekli bit akışı ile temsil edilir. Bu Markov zinciri, kaynakların süper pozisyonundan elde edilir. Örneğin N tane özdeş iki durumlu Markov kaynağının çoğulllanması durumunda (her birinin tepe bit hızı R olmak üzere), j tane kaynak aktif ise toplam bit hızının yoğunluğu jR’ dir.

Bu yaklaşımda, C tampona yollanan toplam bit hızı olmak üzere, x boyutlu tampon için, taşma olasılığının ε ’ dan küçük olmasını sağlayan en küçük C değeri bulunur. Bu değer, çoğullanan bağlantıların eşdeğer kapasitesidir [10].

Eşdeğer kapasitenin belirlenmesi için önce tampon içeriği dağılımı, bağlantı karakteristikleri ve servis hızının bir fonksiyonu olarak elde edilmelidir. Bu ifade daha sonra, mevcut tampon boyutu x için, ε ’ dan daha küçük tampon taşma olasılığı sağlayan servis hızını belirleyecek şekle dönüştürülmelidir. Bu servis hızı değeri, bağlantılar için ayrılması gereken eşdeğer kapasitedir. Tampon içeriği dağılımı, özdeş kaynaklar için mevcut yöntemler kullanılarak türetilir [22]. Bu yöntemler eşdeğer kapasiteyi belirlemek için gerekli bilgiyi sağlamakla birlikte, bu yöntemlerdeki hesaplamaların karmaşıklığı gerçek zaman gereksinimleri ile uyumlu değildir. Eşdeğer kapasitenin değerini bulmak için iteratif sayısal yöntemler kullanılmak zorunda kalınır. Guerin, Ahmadi ve Naghshineh bu problemin üstesinden gelmek için, aşağıda anlatılan yaklaşımları önermişlerdir [10].

Tek bir Kaynak İçin Yapılan Yaklaşımlar :

İki durumlu bir Markov kaynağı; R kaynağın tepe hızını, r kaynağın aktif olduğu zaman bölümünü, b aktif periyodun ortalama süresini göstermek üzere, (R, r, b) parametreleri ile karakterize edilebilir. Servis hızı sabit olan bir tamponu besleyen böyle bir kaynak için tampon içeriği dağılımı, standart teknikler kullanılarak türetilir. Bu dağılımdan, belli servis kalitesini (taşma olasılığı ε) sağlamak için gerekli eşdeğer kapasite c’ yi belirlemek mümkündür. x boyutlu sonlu kapasiteli bir tampon ve eşdeğer kapasite c için, taşma olasılığı ε aşağıdaki biçimde elde edilmiştir [10]:

$$\varepsilon = \beta \times \exp\left(-\frac{x(c - rR)}{b(1 - r)(R - c)c}\right) \quad (5.3)$$

Burada

$$\beta = \frac{(c - rR) + \epsilon r(R - c)}{(1 - r)c}$$

dir. Yukarıdaki eşitlik c için çözülürse tek bir kaynağın eşdeğer kapasitesi elde edilir. Ancak eşdeğer kapasite c doğrudan bulunamaz, (5.3) denklemini sayısal olarak çözülmelidir. β 'nin değeri 1 alınır (genelde β 'nin değeri 1'den küçüktür) c için kesin bir ifade elde edilir. Bu ifade ile elde edilen değerlerin, gerçek değerlere yakın olduğu görülmüştür. Bu durumda tek bir kaynağın eşdeğer kapasitesi [10]:

$$c = \frac{\alpha b(1 - r)R - x + \sqrt{[\alpha b(1 - r)R - x]^2 + 4x\alpha br(1 - r)R}}{2\alpha b(1 - r)} \quad (5.4)$$

şeklindedir. Burada $\alpha = \ln(1/\epsilon)$ dur.

Birden Fazla Kaynak İçin Yapılan Yaklaşımlar :

Birden fazla kaynağın çoğullanması durumunda da servis hızı, tampon içerik dağılımından belirlenebilir. Bu yaklaşım, güçlü basitleştirme teknikleri kullanıldığında bile tek bir kaynağına göre çok daha fazla karmaşıktır. Giriş bit hızı, N durumlu Markov zinciri ile karakterize edildiğinde tampon içeriğinin dağılımı aşağıdaki biçimdedir [22] :

$$F(x) = \sum_{i=1}^N a_i \Phi_i e^{z_i x} \quad (5.5)$$

Burada z_i ve Φ_i sırasıyla, sistemin stasyoner olasılıklarının sağladığı diferensiyel denklemin çözümü ile ortak olan genelleştirilmiş özdeğerler ve özvektörlerdir. a_i ' ler ise sınır koşullardan bulunan katsayılarıdır.

$F(x)$ dağılımını belirleyen özdeğer, özvektör ve katsayıların hesaplanması için genelde kesin ifadeler yoktur (Sadece özdeş iki durumlu Markov kaynakları için kesin bir ifade elde edilmiştir [22]) ve bu değerler sayısal olarak hesaplanmak zorundadır. Bu yüzden, çoğullanmış bir grup kaynağa karşı düşen eşdeğer kapasite de ancak iteratif sayısal teknikler kullanılarak elde edilebilir ve her iterasyonda (5.5) için yeni bir çözüm hesaplanmak zorundadır. Böyle bir işlem doğru sonuçlar

vermekle birlikte, dinamik ve gerçek zamanlı ortamlar ile uyumlu olmadığı için yaklaşımlar yapılmak zorundadır.

İlk olarak asimtotik yaklaştırma yapılmıştır. Buna göre, sistemin sonlu kapasiteli bir tamponu olduğu ve tampon taşma olasılığının yaklaşık olarak, tampon içeriğinin gerçek tampon boyutunu geçme olasılığına eşit olduğu varsayılmıştır. Ayrıca büyük boyutlu tamponlar için (5.5) ifadesinde sadece en büyük negatif özdeğer hesaba katılarak bir yaklaştırma yapılmıştır (diğer terimler tampon boyutu arttıkça ihmal edilebilir hale gelir). Bu varsayım altında tampon taşma olasılığı $G(x)$ ' in aşağıdaki biçimde olduğu bilinmektedir [22, 23]:

$$G(x) = \beta e^{z_0 x} \quad (5.6)$$

Burada z_0 en büyük negatif özdeğerdir ve β , x ' den bağımsız sabit terimdir.

z_0 ' ın nispeten basit ifadelerden bulunabileceği, fakat β sabitinin belirlenmesinin oldukça güç olduğu Kosten tarafından ifade edilmiştir [23]. N tane özdeş iki durumlu Markov kaynağı için

$$G(x) \cong \left(\frac{N r R}{C} \right)^N \left[\prod_{i=1}^{N-[C/R]-1} \frac{z_i}{z_i + z_0} \right] \cdot \exp \left(- \frac{N(C - N r R)}{b(1-r)(NR - C)C} x \right) \quad (5.7)$$

yazılabilir [22]. Burada z_i bütün negatif özdeğerleri göstermektedir ve en büyük negatif özdeğer olan z_0 ' ın tam ifadesi üssel terim ile birlikte verilmiştir. C, N tane bağlantı için ayrılan kapasiteyi, $[C/R]$ ise, C/R ' den daha küçük olan en büyük tam sayıyı gösterir.

(5.7)' de görüldüğü gibi, β ' nın ifadesi oldukça karmaşıktır ve sadece en büyük negatif özdeğeri değil, diğer bütün negatif özdeğerleri de içerir. Tek kaynak için yapılan yaklaşımda olduğu gibi, N tane kaynak için de β ' nın değeri 1 kabul edilmiştir. Guerin, Ahmadi ve Naghshineh, bu varsayımın ancak bağlantı sayısının küçük olduğu veya gerçek toplam eşdeğer kapasitenin bütün bağlantıların ortalama hızına yakın olduğu durumlarda iyi sonuçlar verdiğini, diğer durumlarda ise kötü sonuçlar elde edildiğini göstermişlerdir [10]. Ancak yine de fluid-flow modelinde β ' nın değeri 1 olarak kabul edilmiş, bu varsayımın yetersiz kaldığı durumlar için stasyonier bit hızının dağılımına dayalı ikinci bir yaklaşım geliştirilmiştir.

(5.6) ifadesinde β' 'nın değeri 1 alınırsa tampon taşma olasılığı ε için

$$\varepsilon = e^{Z_0 X} \Rightarrow Z_0 = -\frac{\alpha}{X} \quad (5.8)$$

elde edilir. Burada $\alpha = \ln(1/\varepsilon)$ dur.

Kosten, iki durumlu Markov kaynakları için, en büyük negatif özdeğer z_0 ile, ayrılan kapasite C arasında bir bağıntı olduğunu göstermiştir. Bu bağıntı [23]:

$$C = \frac{\sum_{i=1}^N \left[(\lambda_i + \mu_i + R^{(i)} z_0) - \sqrt{(\lambda_i + \mu_i + R^{(i)} z_0)^2 - 4\lambda_i R^{(i)} z_0} \right]}{2z_0} \quad (5.9)$$

şeklindedir. $1/\lambda_i$ ve $1/\mu_i$; sırasıyla i . kaynağın boş ve aktif periyodlarının ortalama sürelerini göstermektedir. $R^{(i)}$ ise i . kaynağın tepe hızıdır.

Guerin, Ahmadi ve Naghshineh (5.8) ve (5.9) ifadelerini birleştirerek N tane kaynağın eşdeğer kapasitesi C için aşağıdaki ifadeyi elde etmişlerdir [10]:

$$C = \sum_{i=1}^N c_i \quad (5.10)$$

Buradaki c_i , (5.4)' den bulunur ve bu ifadeyi elde etmek için aşağıdaki bağıntılar kullanılmıştır :

$$\mu_i = \frac{1}{b_i} , \quad \lambda_i = \frac{r_i}{b_i(1-r_i)} , \quad \lambda_i + \mu_i = \frac{1}{b_i(1-r_i)}$$

(5.10) ifadesinin olumlu tarafı; hesaplanmasının kolay olması ve kaynak parametrelerine bağlı bir ifade olmasıdır. Ancak β' 'nın değerinin 1 alınmasıyla elde edilen bu ifadede, istatistiksel çoğullamanın etkisi göz ardı edilir. Özellikle her bir kaynağın eşdeğer kapasitesi ortalama bit hızına yakın değilse, bütün kaynakların eşdeğer kapasitesi için (5.10) ile elde edilen değer gerçek değerden uzak olduğu görülmüştür. Bu yüzden Guerin, Ahmadi ve Naghshineh istatistiksel çoğullamanın önemli olduğu durumlarda gerekli band genişliğini doğru bir biçimde belirlemek için stasyonier yaklaşımı önermişlerdir.

B. Stasyoner Yaklaşım :

“Fluid-flow” yaklaşımında da belirtildiği gibi, eşdeğer kapasiteleri ortalama bit hızlarından çok büyük olan bağlantılar çoğullandığında β değerinin 1’ den çok farklı değerler aldığı görülmüştür [10]. Bu durumda stasyoner bit hızı dağılımı ile, bağlantıların eşdeğer band genişliği için daha doğru tahminler yapılabilir.

Eşdeğer kapasite C' nin değeri özel olarak; toplam bit hızının C' yi aşma olasılığı, istenen tampon taşma olasılığından daha küçük olacak biçimde seçilebilir. Bu, ε ’ dan daha küçük tampon taşma olasılığı sağlar. Fakat tamponun etkisi ihmal edildiği için gerekli olan gerçek band genişliğinden daha büyük bir değer elde edilir. Bir grup bağlantının eşdeğer kapasitesi C , (5.11) ifadesinin en küçük değeri olarak tanımlanır.

$$\Pr (B > C) \leq \varepsilon \quad (5.11)$$

Burada B toplam stasyoner bit hızı, ε ise istenen tampon taşma olasılığıdır.

İstatistiksel çoğullamanın etkisinin büyük olduğu durumlarda stasyoner bit hızının, genelde Gaussian dağılımı ile dağıldığı kabul edilir.

Gaussian dağılımı varsayımı, bu dağılım için yapılan standart yaklaşımları kullanma olanağı sağlar. Eşdeğer kapasite C , Gaussian dağılımından elde edilen yaklaşımlardan yararlanarak aşağıdaki biçimde yazılabilir :

$$C = m + \alpha \sigma, \quad \alpha = \sqrt{-2 \ln(\varepsilon) - \ln(2\pi)} \quad (5.12)$$

Burada m , toplam bit hızının ortalaması ($m = \sum_{i=1}^N m_i$), σ ise toplam bit hızının standart sapmasıdır ($\sigma^2 = \sum_{i=1}^N \sigma_i^2$).

(5.12) ifadesinin hesaplanması kolaydır ve bu ifade sadece servis kalitesine ve her bir bağlantının bit hızının ortalamasına ve varyansına bağlıdır.

C. Eşdeğer Kapasitenin Belirlenmesi :

Bir grup bağlantının eşdeğer kapasitesini veren tek bir ifade elde etmek için, flow yaklaşımı ile elde edilen ifadeler [(5.4) ve (5.10)] ve stasyoner yaklaşım ile elde edilen ifade (5.12) birleştirilmelidir. Her iki yaklaşım ile eşdeğer kapasitenin gerçek değerinden büyük değerler elde edildiği için ve farklı gruptaki bağlantılarda bu yaklaşımlar iyi sonuçlar verdiği için eşdeğer kapasite C , her iki yaklaşım ile bulunan değerlerin küçüğüne eşit alınmalıdır [10] :

$$C = \min \left\{ m + \alpha \sigma, \sum_{i=1}^N c_i \right\} \quad (5.13)$$

Burada N, çoğullanan bağlantıların sayısıdır.

GAN kriterine göre, N tane kaynağın kabul edilebilir olup olmadığına (5.13) ifadesine dayalı olarak aşağıdaki biçimde karar verilir [12]:

W çıkış portunun band genişliği olmak üzere, bir grup trafik kaynağı ancak aşağıdaki koşul sağlandığı takdirde kabul edilebilir :

$$W \geq \min \left\{ m + \alpha \sigma, \sum_{i=1}^N c_i \right\}$$

5.4.2 Elwalid-Mitra Kriteri

Elwalid ve Mitra, bir MMFP (Markov Modulated Fluid Process) kaynağının eşdeğer band genişliğinin yaklaşık olarak; kaynağın parametreleri, çoğullayıcı kaynakları ve istenen hücre kayıp olasılığından türetilen matrisin maksimum reel özdeğerine eşit olduğunu göstermişlerdir [11].

Birçok eşdeğer band genişliği temelli kabul kriteri gibi, Elwalid-Mitra kriteri de tampon içeriği dağılımından yararlanılarak türetilmiştir [11].

Elwalid ve Mitra, MMFP modelini kullanarak trafik kaynakları için eşdeğer band genişliği türetmişlerdir. MMFP modeli aşağıdaki biçimde tanımlanabilir [12]:

1. N tane bağımsız “fluid” kaynağı vardır. Her kaynak, bir Markov zinciri ile yönetilir. Q matrisi, varış sürecinin durumları arasındaki geçişi yöneten Markov zincirinin üreticisidir ve

$$Q = \begin{bmatrix} -\alpha & \alpha \\ \beta & -\beta \end{bmatrix}$$

şeklinde tanımlanmıştır. α ve β değerleri, kaynağın ON periyodunun ortalama süresi T_{on} , ve kaynağın OFF periyodunun ortalama süresi T_{off} cinsinden aşağıdaki biçimde ifade edilir :

$$\alpha = 1 / T_{off}, \quad \beta = 1 / T_{on}$$

2. Kaynaklar W kapasiteli çıkış portu tarafından boşaltılan, sonlu kapasiteli, ortak bir tamponu kullanırlar.
3. Bir trafik kaynağının L tane durum ile modellendiği ve Markov zincirinin bütün durumları için varış hızlarının $\bar{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_L)$ vektörü ile gösterildiği varsayılırsa, Λ hız matrisi

$$\Lambda = \text{diag}(\bar{\lambda}) \quad \text{şeklinde tanımlanır.}$$

4. Bütün kaynaklar aynı önceliğe sahiptir.
5. Bir MMFP kaynağını $(Q, \bar{\lambda})$ parametreleri ile karakterize etmek mümkündür.

$G(x)$, tamponun içeriğinin x' i aşma olasılığını, W çıkış portunun band genişliğini göstermek üzere Elwalid ve Mitra çoklu durumlu MMFP kaynakları için aşağıdaki sonucu elde etmişlerdir [11]:

$\ln(\varepsilon / x) \rightarrow \xi \in (-\infty, 0]$ olmak üzere, $\varepsilon \rightarrow 0$ ve $x \rightarrow \infty$ için :

$$\sum_{i=1}^N c_i(\xi) < W \quad \text{ise} \quad G(x) \leq \varepsilon \quad (5.14)$$

$$\sum_{i=1}^N c_i(\xi) > W \quad \text{ise} \quad G(x) > \varepsilon \quad (5.15)$$

Buradaki $c_i(\xi)$ değeri, $[\Lambda - \frac{1}{\xi} Q]$ matrisinin maksimum reel özdeğerine eşittir.

$G(x)$ ' in x boyutlu tampon için taşma olasılığı olarak kullanılması çok yaygın bir varsayımdır. İstenen servis kalitesinin (ε) , tampon taşma olasılığının bir fonksiyonu olduğu varsayımı da yapılırsa; $c_i(\xi)$ değerinin ilgili trafik kaynağının eşdeğer band genişliğine karşı düştüğü söylenebilir [11]. Bu değer, kaynak sistemdeki tek kaynakmış gibi hesaplama yapılarak bulunur. Bu durumda N tane kaynağın eşdeğer band genişliği C , bütün kaynakların eşdeğer band genişliklerinin toplamına eşittir.

$$C = \sum_{i=1}^N c_i(\xi)$$

Yukarıdaki sonuç, asimtotik koşulların sağlandığı varsayımı altında elde edilmiştir [11]. Asimtotik koşullarda tampon boyutu x çok büyük, taşma olasılığı ε ise çok küçük değer alır. Bu koşullar altında, eşdeğer band genişlikleri toplamını çıkış portunun band genişliği ile karşılaştırarak elde edilen kabul bölgesi değerlerinin gerçeğe yakın olduğu görülmüştür [12]. Ancak bu yöntemin, asimtotik olmayan

koşullarda da kabul bölgesini belirlemek için kullanılması önerilmiştir. Bu durumda, tampon boyutu ve taşma olasılığı asimtotik koşulları sağlamasa bile maksimum reel özdeğer $c_i(\xi)$, ilgili kaynağın eşdeğer band genişliği olarak kabul edilir ve bu özdeğerlerin toplamı sistem band genişliği ile karşılaştırılarak kabul bölgesi belirlenir.

Elwalid ve Mitra ayrıca, $(Q, \bar{\lambda})$ ile tanımlanan 2 durumlu tek bir MMFP kaynağının eşdeğer band genişliğinin $(c(\xi))$ hesaplanması için aşağıdaki formülü önermişlerdir [11]:

$$c(\xi) = \frac{0.5}{\xi} \left[[(\lambda_1 + \lambda_2)\xi + \alpha + \beta] - \sqrt{\{(\lambda_1 + \lambda_2)\xi + \alpha + \beta\}^2 - 4(\lambda_1\lambda_2\xi^2 + \beta\lambda_1\xi + \alpha\lambda_2\xi)} \right] \quad (5.16)$$

5.4.3 Geliştirilmiş Elwalid-Mitra Kriteri

Elwalid-Mitra kriteri asimtotik olmayan koşullarda yetersiz kaldığı için, bu kritere Rege tarafından iyileştirme getirilmiştir. Bu yöntemde bir kaynağın eşdeğer band genişliği, yapılan daha sonraki hesaplamalarda kaynağın tepe hızı yerine kullanılır. Kabul edilebilir kaynak kümesini belirlemek amacıyla da, her bir band genişliğinin lineer olmayan bir fonksiyonu kullanılır.

Geliştirilmiş Elwalid-Mitra Kriteri' ne göre N tane kaynağın kabul edilebilirliğini belirlemek amacıyla aşağıdaki adımlar izlenir [12] :

1. Her kaynak için eşdeğer band genişliği c_i , Elwalid-Mitra kriterine göre hesaplanır. Ancak c_i hesaplanırken hücre kayıp olasılığı ε yerine ε' kullanılır. ε' şu şekilde tanımlanmıştır.:

$$\varepsilon' = \left[\frac{W}{(W - m)} \right] \varepsilon$$

Burada W çıkış portunun band genişliğini, m ise N tane kaynak için toplam ortalama hızı gösterir.

2. Bundan sonra her kaynak için tepe hızı yerine, 1. adımda hesaplanan eşdeğer band genişliği kullanılır.
3. N tane kaynak için varyans σ^2 , GAN kriterinde olduğu biçimde hesaplanır. Ancak bu formüllerde kaynağın tepe hızı yerine eşdeğer band genişliği kullanılır. Hesaplanan ortalama hız ve varyans değerleri kullanılarak, Gaussian yaklaşımı ile elde edilen aşağıdaki formüle göre $P(\text{loss})$ olasılığı hesaplanır.

$$P(\text{loss}) = \frac{\sigma}{m\sqrt{(2\pi)}} \exp\left[\frac{-1}{2}\left(\frac{W - m}{\sigma}\right)^2\right] \quad (5.17)$$

4. N tane kaynak için ancak aşağıdaki koşul sağlandığı takdirde, bu kaynaklar kabul edilebilir :

$$W \geq \sum_{i=1}^N c_i \quad \text{veya} \quad P(\text{loss}) \leq \varepsilon$$

Rege tarafından Elwalid-Mitra kriterine getirilen bu değişiklik, aşağıdaki varsayıma dayalı olarak yapılmıştır :

x boyutlu bir tamponu besleyen, tepe hızı R, ortalama hızı m_i , ve eşdeğer band genişliği c_i olan bir kaynak; sıfır boyutlu bir tamponu besleyen, tepe hızı c_i , ortalama hızı m_i olan bir ON-OFF kaynağına eşdeğerdir. Bu varsayım ise; her iki durumda benzer tampon taşma olasılıkları elde edildiği için yapılmıştır.

Rege ayrıca deneysel olarak bulunan aşağıdaki değişikliğin yapılması durumunda kabul bölgesi için daha doğru değerler elde edileceğini göstermiştir [12] :

3. adımda tepe hızı R, ortalama hızı m_i , eşdeğer band genişliği c_i olan kaynak için yapılan hesaplamalarda tepe hızı yerine eşdeğer band genişliği değil de, $m_i + \sqrt{(R - m_i)(c_i - m_i)}$ ifadesi kullanılırsa kabul bölgesi için daha iyi sonuç elde edilir.

5.4.4 Sohraby Kriteri (Yoğun Trafik Yaklaşımı)

Sohraby da, kuyruk-uzunluğu dağılımının asimtotik davranışına dayalı olarak bir yaklaşım önermiştir.

Sonlu kapasiteli ve servis süresi sabit bir tampon ve MMBP tipi varış süreçleri için, $Q = [Q_{ij}]$ Markov zincirinin olasılık geçiş matrisini, $\bar{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_L)$ farklı durumlardaki varış hızları vektörünü göstermek üzere, varış sürecinin olasılık üretme fonksiyonu $B(z)$, aşağıdaki gibi tanımlanabilir [9] :

$$b_{ij}(z) \stackrel{\Delta}{=} E[z^{A_{n+1}} \mathbf{1}(S_{n+1} = j \mid S_n = i)]$$

Burada S_n , Markov zincirinin n. durumunu temsil eder. $\mathbf{1}(V)$, V olayı doğru ise 1' e eşittir, aksi takdirde sıfırdır. Kararlı-hal kuyruk uzunluğu dağılımı, geometrik olarak dağıtılmış bir kuyruğu göstermektedir. Bu yüzden yeterince büyük i değerleri için :

$$\Pr(\text{kuyruk uzunluğu} > i) \approx \alpha (1/z^*)^i$$

yazılabilir [9]. Burada z^* , $|zI - B(z)|$ birim çemberinin dışında kalan en küçük köktür. α ise bilinmeyen bir sabittir. $\gamma_1, \gamma_2, \dots, \gamma_N, B(z)$ ' nin özdeğerleri olmak üzere $|zI - B(z)|$ determinanı aşağıdaki biçimde yazılabilir :

$$|zI - B(z)| = \prod_{i=1}^N (z - \gamma_i(z))$$

Buna göre, $B(z)$ ' nin özdeğerleri bulunursa yukarıdaki determinantın sıfırları kolayca elde edilebilir. Ancak n tane $z - \gamma_i(z) = 0$ denkleminin hangisinin z^* ı vereceği belli değildir. z^* kökünün $z - \gamma(z) = 0$ denklemini çözdüğü, $\gamma(z)$ ' nin de $B(z)$ ' nin Perron-Frobenius (PF) özdeğeri olduğu gösterilmiştir. Bağımsız varış süreçlerinin süper pozisyonu olan bir varış süreci için PF özdeğer, kaynakların PF özdeğerlerinin çarpımına eşittir. IBP kaynakların süper pozisyonu göz önüne alınırsa, z^* ın “sabit-nokta” (fixed-point) problemini çözerek elde edilebileceği gösterilmiştir. Ancak sabit-nokta problemi sayısal olarak çözülebilir. N tane IBP kaynağın süper pozisyonu durumunda z^* ın hesaplanmasını kolaylaştırmak amacıyla, R kaynağın tepe hızını, r kaynağın aktif olduğu zaman bölümünü, b aktif periyodun ortalama süresini göstermek üzere Sohraby aşağıdaki yaklaşımı önermiştir. Bu yaklaşım, b_i 's' nin çok büyük olduğu varsayımı altında geçerlidir [9].

$$z^* \approx 1 + \frac{1-r}{\sum_{i=1}^N r_i R_i (1-r_i)^2 b_i} \quad (r = \sum_{i=1}^N r_i R_i) \quad (5.18)$$

Sohraby, ON ve OFF periyodları keyfi bir dağılım ile karakterize edilen ON-OFF tipi kaynaklar için (5.19)' daki yaklaşımı önermiştir. Burada cv_{on} ve cv_{off} sırasıyla kaynağın ON ve OFF periyodunun uzunluğunun varyasyon katsayısını göstermektedir. ON periyodunun ortalama süresi b 'nin, ve tampon boyutu x ' in çok büyük olduğu durumlar için bu yaklaşım geçerlidir [9]:

$$z^* \approx 1 + \frac{2(1-r)}{\sum_{i=1}^N r_i R_i (1-r_i)^2 (cv_{on}^2 + cv_{off}^2) b_i} \quad (5.19)$$

Kuyruk uzunluğu dağılımından yararlanarak

$$Pr (\text{kuyruk uzunluğu} > i) \approx \gamma (1/z^*)^i$$

yazmak mümkündür. Burada γ , trafik yoğunluğunu göstermektedir. Sohraby bu yaklaşımın $0.8 < \gamma < 1$ aralığındaki trafik yoğunlukları için iyi sonuçlar verdiğini göstermiştir. x tampon boyutunu göstermek üzere, sistemdeki hücre kayıp olasılığı yaklaşık olarak:

$\gamma (1/z^*)^x$ şeklinde hesaplanabilir. z^* ise (5.18) veya (5.19)' dan bulunur. Bu durumda yeni bağlantının kabul edilip edilmeyeceğine karar vermek çok kolaydır :

$\gamma (1/z^*)^x < \varepsilon$ ise, yeni bağlantı isteği kabul edilir.

5.5 Eşdeğer Band Genişliği Temelli Kabul Kriterlerinin Başarımlarının Karşılaştırılması

Bu bölümde belirli kaynak modelleri kullanılarak, eşdeğer band genişliği temelli kabul kriterlerinden Elwalid–Mitra, GAN, Sohraby ve Geliştirilmiş Elwalid–Mitra Kriteri' nin belirli tampon boyutlarında ayrı ayrı kabul bölgesi değerleri hesaplanmıştır. Kabul bölgesi; istenen hücre kayıp olasılığının sağlanabildiği durumda kabul edilebilecek maksimum bağlantı sayısına karşı düşer. Bu hesaplamayı yapabilmek amacıyla, seçilen kabul kriterine ve girilen kaynak parametrelerine, tampon boyutuna, çıkış portunun band genişliğine ve hücre kayıp olasılığı ε değerine göre kabul bölgesini bulan bir PASCAL programı yazılmıştır. Programın açıklaması ve kaynak kodu Bölüm 5.5.1' de verilmiştir.

5.5.1 Girilen Kaynak Parametreleri, Çıkış Portunun Hızı, Tampon Boyutu ve ε Değeri İçin Eşdeğer Band Genişliği Temelli Kabul Kriterlerinin Kabul Bölgesini Hesaplayan PASCAL Programının Kaynak Kodu

Kaynak kodu aşağıda verilen bu program, kullanıcı tarafından girilen kaynağın parametrelerine, tampon boyutuna, çıkış portunun hızına, hücre kayıp olasılığına ve seçilen kabul kriterine göre istenen hücre kayıp olasılığını sağlayabilen maksimum bağlantı sayısını hesaplar.

Programın çalıştırılmasıyla ilk ekranda kullanıcıdan sırasıyla çıkış portunun band genişliği W (hücre/s), kaynağın tepe hızı R_{peak} (hücre/s), kaynağın aktif olduğu ortalama süre ton (s), kaynağın aktif olmadığı ortalama süre $toff$ (s), tampon boyutu x (hücre) ve kabul edilebilir hücre kayıp olasılığı Epsilon' u girmesi istenir. İkinci ekranda kullanıcıdan eşdeğer band genişliği temelli dört kabul kriterinden birisini seçmesi istenir. Girilen değerler için, seçilen kabul kriterinin Bölüm 5.4' de anlatılan algoritması yürütülür ve bu algoritmaya göre hesaplanan kabul edilebilecek maksimum bağlantı sayısı ekrana yazdırılır.

{BU PROGRAM SEÇİLEN KABUL KRİTERİNE GÖRE KABUL EDİLEBİLECEK
MAKSİMUM BAĞLANTI SAYISINI HESAPLAR.

PROGRAMDA KULLANICININ SEÇEBİLECEĞİ EŞDEĞER BAND GENİŞLİĞİ
TEMELLİ KABUL KRİTERLERİ :

1-) GAN KRİTERİ

2-) ELWALID-MITRA KRİTERİ

3-) GELİŞTİRİLMİŞ ELWALID-MITRA KRİTERİ

4-) SOHRABY KRİTERİ

}

program esdeger_band_genisligi_kriterleri;

uses crt;

var

N : integer;

W, Rpeak, x, Epsilon, ton, toff, ctop, ploss : real;

CLP : boolean;

ch :char;

{BU ALT PROGRAM GAN KRİTERİNE GÖRE TEK BİR KAYNAĞIN
EŞDEĞER BAND GENİŞLİĞİNİ HESAPLAR}

function cnew_gan:real;

var

a, temp_var, g:real;

begin

g:=ton/(ton+toff);

a:=Ln(1/Epsilon)*ton*(1-g)*Rpeak;

temp_var:=Sqr(a-x)+(4*x*a*g);

cnew_gan:= (((a-x) + (Sqrt(temp_var)))*Rpeak) / (2*a);

end;

{BU ALT PROGRAM ELWALID-MITRA KRİTERİNE GÖRE TEK
BİR KAYNAĞIN EŞDEĞER BAND GENİŞLİĞİNİ HESAPLAR}

function cnew_ebw(var alfa, beta, z:real):real;

var

temp_var1, temp_var2 : real;

begin

temp_var1:= (Rpeak*z) + (beta + alfa);

temp_var2:= (Sqr(temp_var1)) - (4*(alfa*Rpeak*z));

cnew_ebw:= (temp_var1 - (Sqrt(temp_var2))) / (2*z);

end;

{BU ALT PROGRAM GAN KRİTERİNE GÖRE N TANE KAYNAĞIN
EŞDEĞER BAND GENİŞLİĞİNİ HESAPLAR}

function call_gan:real;

var

c : array[1..1000] of real;

Exp1, Exp2, ctop, mtop, sigmat, alfa, c_real, m, g : real;

i : integer;

begin

g:=ton/(ton+toff);

ctop:=0;

if N=1 then

begin

call_gan:=cnew_gan;

end

else

begin

m:=g*Rpeak;

mtop:=m*N;

sigmat:=Sqrt((m*(Rpeak-m))*N);

alfa:=Sqrt((-2*Ln(Epsilon))-Ln(2*pi));

Exp1:= mtop + (alfa*sigmat);

for i:=1 to N do

begin

c[i]:=cnew_gan;

ctop:=ctop+c[i];

end;

Exp2:=ctop;

if Exp1<Exp2 then

c_real:=Exp1

else

c_real:=Exp2;

call_gan:=c_real;

end; {if N=1}

end;

{BU ALT PROGRAM ELWALİD-MİTRA KRİTERİNE GÖRE N TANE
KAYNAĞIN EŞDEĞER BAND GENİŞLİĞİNİ HESAPLAR}

function call_ebw:real;

var

c : array[1..1000] of real;

ctop, alfa, beta, z : real;

i : integer;

begin

alfa:=1/toff;

```

beta:=1/ton;

z:=(Ln(Epsilon)) / x;

ctop:=0;

if N=1 then
begin
  call_ebw:=cnew_ebw(alfa, beta, z);
end
else
begin
  for i:=1 to N do
  begin
    c[i]:=cnew_ebw(alfa, beta, z);
    ctop:=ctop+c[i];
  end;

  call_ebw:=ctop;
end; {if N=1}
end;

```

{BU ALT PROGRAM GELİŞTİRİLMİŞ ELWALİD-MİTRA KRİTERİNE
GÖRE N TANE KAYNAĞIN EŞDEĞER BAND GENİŞLİĞİNİ HESAPLAR}

```

procedure call_wr;
var
  c : array[1..1000] of real;
  g, m, mtop, sigmat, t1, t2, t, alfa, beta, z, Eps2 :real;
  i : integer;

```

```

begin
  alfa:=1/toff;
  beta:=1/ton;
  g:=ton/(ton+toff);
  m:=g*Rpeak;
  mtop:=m*N;
  Eps2:=(W/(W-mtop))*Epsilon;
  z:=(Ln(Eps2)) / x;
  ctop:=0;

  if N=1 then
    ctop:=cnew_ebw(alfa, beta, z)
  else
    begin
      for i:=1 to N do
        begin
          c[i]:=cnew_ebw(alfa, beta, z);
          ctop:=ctop+c[i];
        end;

      t:=m+Sqrt((Rpeak-m)*(c[1]-m));
      sigmat:=Sqrt((m*(t-m))*N);
      t1:=-(Sqr((W-mtop)/sigmat))/2;
    end;
  end;

```

```

t2:=exp(t1);

ploss:=(sigmat/(mtop*Sqrt(2*pi)))*t2;
end; {if N=1}
end;

```

{BU ALT PROGRAM SOHRABY KRİTERİNE GÖRE N TANE KAYNAĞIN EŞDEĞER BAND GENİŞLİĞİNİ HESAPLAR}

```
function sohraby:real;
```

```
var
  gtop, temp_var1, temp_var2, Rk, Tk, g : real;
```

```
begin
```

```

  g:=(ton/(ton+toff))*Rpeak;
  gtop:=g*N;
  Rk:=W/Rpeak;
  Tk:=Rpeak*ton;

```

```

  temp_var1:=((-Ln(Epsilon))/(1-gtop))-1;
  temp_var2:=N*(g*(Sqr(1-Rk*g)));

```

```

  sohraby:=temp_var1*temp_var2*Tk;
end;
```

{BU ALT PROGRAM, SEÇİLEN KRİTERE GÖRE, N TANE KAYNAK KABUL EDİLEBİLİR İSE FALSE, EDİLEMEZ İSE TRUE DEGERİNİ DÖNDÜRÜR}

```
function cell_loss:boolean;
```

```
var
  c_esit : real;
```

```
begin
```

```
  case ch of
```

```
    '1':begin
```

```

      c_esit:=call_gan;
      if c_esit <= W then
        cell_loss :=false
      else
        cell_loss:=true;
      end;

```

```
    '2':begin
```

```

      c_esit:=call_ebw;
      if c_esit <= W then
        cell_loss :=false
      else
        cell_loss:=true;
      end;

```

```
    '3':begin
```

```
      call_wr;
```

```

        if ((ploss<=Epsilon) or (ctop <= W)) then
            cell_loss :=false
        else
            cell_loss :=true;
        end;
    '4':begin
        c_esit:=sohraby;
        if c_esit < x then
            cell_loss :=false
        else
            cell_loss:=true;
        end;
    end; {case}
end;

```

{Ana Program}

```

begin
    clrscr;
    write('Trunk band genisligini girin (hucre/sn): ');
    readln(W);
    writeln;
    write('Kaynagin tepe hizini girin (hucre/sn) : ');
    readln(Rpeak);
    write('Kaynagin aktif oldugu sure (ton) : ');
    readln(ton);
    write('Kaynagin aktif olmadigi sure (toff) : ');
    readln(toff);writeln;
    write('Tampon boyutunu girin (hucre) : ');
    readln(x);
    write('Istenen Hucre kayip olasiligi Epsilon =');
    readln(Epsilon);
    clrscr;

    writeln('Lütfen aşağıdaki kriterlerden birini seçin '); writeln;
    writeln('1-) Guerin-Ahmadi-Naghsineh (GAN) Kriteri');
    writeln('2-) Elwalid-Mitra Kriteri');
    writeln('3-) Geliştirilmiş Elwalid-Mitra Kriteri');
    writeln('4-) Sohraby Kriteri'); writeln;
    write('Seçiminiz : ');
    readln(ch);

    clrscr;

    N:=1;

    CLP:=cell_loss;

    while (not(CLP)) do
        begin
            N:=N+1;
            CLP:=cell_loss;
        end;
    writeln('Maksimum kabul edilebilecek baglanti sayisi = ', N-1);
    readln;
end.

```

5.5.2 Eşdeğer Band Genişliği Temelli Kabul Kriterlerinin Kabul Bölgesini Hesaplayan Programın Belirli Tampon Boyutlarında Yürütülmesi Sonucu Elde Edilen Değerler

Bu bölümde Bölüm 5.5.1’ de kaynak kodu verilen programın, çeşitli kaynak modellerinde ve tampon boyutlarında dört kriter için ayrı ayrı yürütülmesi sonucu elde edilen kabul bölgesi değerleri tablolar halinde verilmiştir.

Aşağıdaki kaynak modelleri kullanılarak her bir kriter için kabul bölgesi hesaplanmıştır :

1. A Tipi Kaynak : ON-OFF tipi bir kaynaktır. ON periyodu sırasında kaynak, tepe hızı $0.0208*W$ ($1/48*W$) hücre/s. hızla ATM hücreleri üretir. Burada W, çıkış portunun band genişliğidir. OFF periyodu sırasında ise hiç hücre üretilmez. ON ve OFF periyodlarının ortalama süresi sırasıyla 9600 ve 54400 zaman birimidir. Bu kaynak tipinde ortalama yığın uzunluğu yaklaşık olarak 200 hücredir. ON periyodunun ortalama süresinin, ON ve OFF periyodlarının ortalama sürelerinin toplamına oranı ($Ton/(Ton + Toff)$) ise 0.15’ dir. Bu orana “iş çevrimi (duty cycle)” denir [12].
2. B Tipi Kaynak : ON-OFF tipi bir kaynaktır. ON periyodu sırasındaki hücre üretme hızı, yani tepe hızı $0.0625*W$ ($1/16*W$) hücre/s, ON ve OFF periyodlarının ortalama süresi ise sırasıyla 80 ve 3120 zaman birimidir. Bu kaynak tipinde ortalama yığın uzunluğu yaklaşık olarak 5 hücredir. Bu kaynağın iş çevrimi ($Ton/(Ton + Toff)$) ise 0.025’ dir.
3. A’ Tipi Kaynak : ON-OFF tipi bir kaynaktır. ON periyodu sırasındaki hücre üretme hızı $0.208*W$ hücre/s, ON ve OFF periyodlarının ortalama süresi ise sırasıyla 960 ve 5440 zaman birimidir. Ortalama yığın uzunluğu yaklaşık olarak 200 hücredir. İş çevrimi 0.15’ dir.
4. A’’ Tipi Kaynak : ON-OFF tipi bir kaynaktır. ON periyodu sırasındaki hücre üretme hızı $0.208*W$ hücre/s, ON ve OFF periyodlarının ortalama süresi ise sırasıyla 960 ve 63040 zaman birimidir. Ortalama yığın uzunluğu yaklaşık olarak 200 hücredir. İş çevrimi 0.015’ dir.

Bütün kaynak tipleri için zaman birimi, çıkış portu üzerinden hücre iletme süresidir ve çıkış portunun band genişliği W, 1 hücre/s’ ye eşit alınmıştır. Ayrıca yukarıdaki kaynak modelleri için yürütülen programda, sistemde sağlanması istenen hücre kayıp olasılığı (ϵ) 10^{-4} olarak alınmıştır.

Gerçek W değerinin 155 Mbit/s, hücre uzunluğunun da 53 byte yani 424 bit olduğu dikkate alınır, çıkış portu üzerinden hücre iletme süresi gerçekte yaklaşık olarak 2,7355 µs. dir. Bu durumda örneğin A tipi bir kaynak, gerçekte tepe hızı 3,224 Mbit/s, ON periyodunun ortalama süresi yaklaşık 26 ms, OFF periyodunun ortalama süresi yaklaşık 149 ms. olan bir kaynağa karşı düşer.

Kriterler için yukarıdaki kaynak modelleri kullanılarak elde edilen kabul bölgesi değerleri verilmeden önce, bu değerler karşılaştırılırken sıkça kullanılan “istatistiksel çoğullama kazancı” kavramı aşağıdaki biçimde açıklanabilir.

İstatistiksel Çoğullama Kazancı : Belli bir trafik sınıfı için istatistiksel çoğullama kazancı; kabul kriteri ile bulunan maksimum bağlantı sayısı N’ in, tepe hızında ayırma kullanıldığında bulunan maksimum bağlantı sayısı Nmin’ e oranı olarak tanımlanır. (N / Nmin). Tepe hızında ayırma kullanıldığında bulunan kabul edilen maksimum bağlantı sayısı Nmin, çıkış portunun kapasitesinin kaynağın tepe hızına oranına eşittir ($N_{min} = \lfloor W / R \rfloor$) [9].

A ve B Tipi Kaynaklar İçin Kabul Bölgesi Değerleri

A ve B tipi kaynaklar için eşdeğer band genişliği temelli dört kabul kriterinin tanımladığı kabul bölgesi değerleri sırasıyla Tablo 5.1’ de ve Tablo 5.2’ de verilmiştir. A ve B kaynak tipleri, iki farklı trafik tipini temsil eder. A tipi kaynağın yığın uzunluğu büyüktür ancak tepe hızının çıkış portunun band genişliğine oranı küçüktür. B tipi kaynağın yığın uzunluğu küçüktür ancak tepe hızının çıkış portunun band genişliğine oranı, A tipi kaynağa göre daha büyüktür.

Her iki kaynağın tepe hızının çıkış portunun band genişliğine oranı düşük olduğundan ve her iki kaynağın iş çevrimleri küçük olduğundan, bu kaynaklar küçük tampon boyutlarında büyük istatistiksel çoğullama kazancı sağlayabilirler.

A ve B tipi kaynaklar için büyük tampon boyutlarında (ortalama yığın uzunluğunun 40 katı veya daha büyük tampon boyutlarında) dört kriter ile belirlenen kabul bölgelerinin birbirine yakın olduğu görülmektedir (Tablo 5.1 ve Tablo 5.2). Tampon boyutu küçüldükçe, Elwalid-Mitra ve Sohraby kriterleri ile belirlenen kabul bölgesi değerlerinin diğer kriterler ile belirlenen kabul bölgesi değerlerinden daha küçük olduğu görülmektedir. Bu sonucun, Elwalid-Mitra ve Sohraby kriterlerinin istatistiksel çoğullamayı göz önünde bulundurmamalarından kaynaklandığı düşünülebilir.

Geliştirilmiş Elwalid-Mitra kriteri ile, orta tampon boyutlarında bile daha büyük kabul bölgesi değerleri elde edilir.

GAN kriteri ile elde edilen sonuçlar ilginçtir. Bu kriter, kabul bölgesini hesaplarken iki ifade kullanır ve bu ifadelerden küçük değer alanına göre kabul bölgesini belirler. Bu ifadelerden birisi tampon boyutuna bağlı olmadığı için belirli bir eşiğin altındaki bütün tampon boyutlarında GAN kriteri ile belirlenen kabul bölgesi değerleri aynıdır. Bu eşiğin üzerinde ise GAN kriteri ile belirlenen kabul bölgesi, Elwalid-Mitra kriteri ile belirlenen kabul bölgesi ile aynıdır.

Tampon boyutundan bağımsız bir şekilde istatistiksel kazancı hesaba katan GAN kriteri ile tanımlanan kabul bölgesi değerleri, A ve B tipi kaynaklarda olduğu gibi, kaynağın tepe hızı çıkış portunun band genişliğine göre küçükse veya kaynağın iş çevrimi küçükse, Elwalid-Mitra kriteri ile bulunan kabul bölgesi değerlerinden daha büyüktür. Küçük tampon boyutlarında GAN kriteri için istatistiksel çoğullama kazancı A tipi kaynaklarda yaklaşık olarak 4 (187/48), B tipi kaynaklarda ise yaklaşık olarak 15 (243/16)' dir.

Küçük tampon boyutlarında (ortalama yığın uzunluğu kadar veya daha küçük tampon boyutlarında) Elwalid-Mitra ile tanımlanan kabul bölgesi, alt sınırı, yani çıkış portunun band genişliğinin kaynağın tepe hızına oranına yaklaşıp. GAN kriteri ise, küçük tampon boyutlarında tampon boyutundan bağımsız değerler bulur. Geliştirilmiş Elwalid-Mitra kriterinin küçük tampon boyutlarında Elwalid-Mitra kriterinden daha büyük kabul bölgesi tahminlerinde bulunabildiği görülmüştür.

Tablo 5.1 A Tipi Kaynaklar İçin Kabul Bölgesinin Tampon Boyutuna ve Kullanılan Kabul Kriterine Göre Değişimi (con = 1, P (loss) = 10^{-4})

TAMPON BOYUTU x (Hücre)	KABUL EDİLEBİLECEK MAKSİMUM BAĞLANTI SAYISI			
	GAN Kriteri	Elwalid – Mitra Kriteri	Geliştirilmiş Elwalid – Mitra Kriteri	Sohraby Kriteri
50	187	49	217	12
75	187	50	218	18
100	187	50	219	24
350	187	58	224	71
1 000	187	89	235	142
3 500	210	210	244	234
5 000	240	240	269	254
7 500	265	265	276	272

Sohraby kriteri ile genelde Elwalid-Mitra kriterinden daha büyük kabul bölgesi değerleri bulunur. Ancak bu kriterin küçük tampon boyutlarında Elwalid-Mitra kriterinden daha küçük kabul bölgesi tahminlerinde bulunduğu görülmektedir.

Tablo 5.2 B Tipi Kaynaklar İçin Kabul Bölgesinin Tampon Boyutuna ve Kullanılan Kabul Kriterine Göre Değişimi (con = 1, P (loss) = 10^{-4})

TAMPON BOYUTU x (Hücre)	KABUL EDİLEBİLECEK MAKSİMUM BAĞLANTI SAYISI			
	GAN Kriteri	Elwalid – Mitra Kriteri	Geliştirilmiş Elwalid – Mitra Kriteri	Sohraby Kriteri
1	243	16	303	15
2	243	16	309	31
5	243	17	315	71
20	243	27	340	211
50	243	133	446	349
100	365	365	523	449
200	500	500	528	526
500	584	584	589	588

A' Tipi Kaynaklar İçin Kabul Bölgesi Değerleri

A' tipi trafik kaynağının tepe hızı, A tipi trafik kaynağının 10 katıdır. Dolayısıyla çıkış portunun hızının kaynağın tepe hızına oranı, A tipi kaynağa göre 10 kat azalmıştır. Ortalama yığın uzunluğu ve iş çevrimi A tipi kaynak ile aynıdır. A' tipi kaynaklar için eşdeğer band genişliği temelli dört kabul kriterinin tanımladığı kabul bölgesi değerleri Tablo 5.3' de verilmiştir.

Büyük tampon boyutları dışında Elwalid-Mitra kriteri ile elde edilen kabul bölgesi değerlerinin çok küçük olduğu Tablo 5.3' de görülmektedir. GAN kriterinin küçük tampon boyutlarında kabul bölgesi için daha büyük tahminlerde bulunduğu, ancak orta tampon boyutlarında Elwalid-Mitra kriteri ile aynı sonuçları verdiği görülmüştür.

Geliştirilmiş Elwalid-Mitra kriteri ile Elwalid-Mitra kriterinden daha büyük kabul bölgesi değerleri elde edilmiştir. Ancak A ve B tipi kaynaklarda olduğu gibi, Geliştirilmiş Elwalid-Mitra kriteri ile bulunan değerler ile diğer kriterler ile bulunan değerler arasında çok büyük fark yoktur.

Sohraby kriterinin ise, küçük tampon boyutları dışında bu kaynak tipi için büyük kabul bölgesi değerleri verdiği görülmektedir.

Tablo 5.3 A' Tipi Kaynaklar İçin Kabul Bölgesinin Tampon Boyutuna ve Kullanılan Kabul Kriterine Göre Değişimi (con = 1, P (loss) = 10^{-4})

TAMPON BOYUTU x (Hücre)	KABUL EDİLEBİLECEK MAKSİMUM BAĞLANTI SAYISI			
	GAN Kriteri	Elwalid – Mitra Kriteri	Geliştirilmiş Elwalid – Mitra Kriteri	Sohraby Kriteri
50	6	4	6	1
75	6	5	6	1
100	6	5	7	2
350	6	5	7	7
1 000	8	8	9	14
3 500	21	21	21	23
5 000	24	24	25	25
7 500	26	26	27	27

A'' Tipi Kaynaklar İçin Kabul Bölgesi Değerleri

A'' kaynak tipinde iş çevrimi, A veya A' kaynak tiplerine göre 10 kat daha küçüktür. Bu yüzden bu kaynak tipiyle küçük tampon boyutlarında bile büyük istatistiksel kazanç elde etmek mümkündür. A'' tipi kaynaklar için eşdeğer band genişliği temelli dört kabul kriterinin tanımladığı kabul bölgesi değerleri Tablo 5.4' de verilmiştir.

Elwalid-Mitra kriteri, temelde kaynakların kabul edilebilirliğini kaynakların tepe hızına dayalı olarak belirlediği için, küçük iş çevrimli kaynakların istatistiksel çoğullanmasındaki avantajı göz ardı eder. Bu yüzden Elwalid-Mitra kriteri ile, iş çevrimi çok küçük olan A'' tipi kaynaklar için küçük tampon boyutlarında küçük kabul bölgesi değerleri elde edilmiştir.

GAN kriteri ile bu kaynak tipi için küçük tampon boyutlarında daha büyük kabul bölgesi değerleri elde edilmiştir. Orta tampon boyutlarında ise kabul bölgesi değerleri, Elwalid-Mitra kriteri ile elde edilen değerler ile aynıdır.

Geliştirilmiş Elwalid-Mitra kriteri ile bu kaynak tipinde, A' tipi kaynağa göre daha büyük kabul bölgesi değerleri elde edilmiştir. Ancak A ve B tipi kaynaklarda olduğu gibi, Geliştirilmiş Elwalid-Mitra kriteri ile bulunan değerler ile diğer kriterler ile bulunan değerler arasındaki büyük fark bu kaynak tipinde de gözlenmemiştir.

Sohraby kriteri, A' tipi kaynak için sergilediği davranışı sergiler. Yani orta ve büyük tampon boyutlarında büyük kabul bölgesi değerleri bulmakla birlikte, küçük tampon boyutlarında çok küçük kabul bölgesi değerleri bulabilmektedir. Ancak A'' tipi kaynağın iş çevrimi A' tipi kaynağa göre daha küçük olduğu için, ve Sohraby kriteri Elwalid-Mitra kriteri gibi istatistiksel çoğullanmanın avantajlarını göz ardı ettiği için, bu kaynak tipinde Sohraby kriteri ile küçük tampon boyutlarında bulunan değerler,

diğer kriterlerin bulduğu değerlere ve büyük tampon boyutlarında bulunan değerlere göre oldukça küçük kalır.

Tablo 5.4 A'' Tipi Kaynaklar İçin Kabul Bölgesinin Tampon Boyutuna ve Kullanılan Kabul Kriterine Göre Değişimi (con = 1, P (loss) = 10^{-4})

TAMPON BOYUTU x (Hücre)	KABUL EDİLEBİLECEK MAKSİMUM BAĞLANTI SAYISI			
	GAN Kriteri	Elwalid – Mitra Kriteri	Geliştirilmiş Elwalid – Mitra Kriteri	Sohraby Kriteri
50	61	4	81	9
75	61	5	81	14
100	61	5	82	18
350	61	5	83	56
1 000	61	10	102	120
3 500	159	159	198	214
5 000	207	207	225	237
7 500	244	244	258	259

Yapılan çalışmalar sonucunda eşdeğer band genişliği temelli kabul kriterlerinin başarımları hakkında aşağıdaki sonuçlar elde edilmiştir [12]:

Elwalid-Mitra kriteri, büyük tampon boyutlarında iyi sonuçlar vermekle birlikte, küçük tampon boyutlarında kabul bölgesi için kötü tahminlerde bulunabilmektedir.

GAN (Guerin - Ahmadi - Naghshineh) kriteri, istatistiksel çoğullama kazancını hesaba katarak küçük tampon boyutlarında önemli bir iyileştirme sağlamıştır. Orta ve büyük tampon boyutlarında Elwalid-Mitra kriteri ile aynı davranışı sergiler.

Sohraby kriteri, orta ve büyük tampon boyutlarında, kabul bölgesi için Elwalid-Mitra kriterinden daha iyi tahminlerde bulunur. Trunk band genişliğinin kaynağın tepe hızına oranının küçük olduğu durumda, orta tampon boyutlarında Sohraby kriteri diğer üç kritere göre daha iyi sonuçlar verir. Küçük tampon boyutlarında ise Sohraby kriteri zayıf kalır.

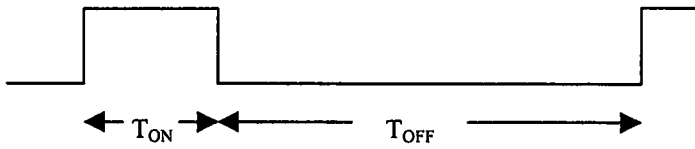
Geliştirilmiş Elwalid-Mitra kriteri, GAN kriterine benzer. Farklı olarak; GAN kriterinde tampon boyutunun hesaba katılmadığı durumlarda da tampon boyutunu hesaba katmak için, deneysel olarak bulunan iyileştirmeler kullanılmıştır. Geliştirilmiş Elwalid-Mitra kriteri, çıkış portunun hızının kaynağın tepe hızına oranı çok küçük değilse bütün tampon boyutlarında kabul bölgesi için iyi tahminlerde bulunur. Ancak bu oran küçük olduğunda, kabul bölgesi için yapılan tahminlerin %30-40 oranında yanlış olduğu görülmüştür [12].

Yukarıdaki tablolarda elde edilen değerler de, bu sonuçları doğrular niteliktedir.

6. “ON-OFF” KAYNAK MODELİNE DAYALI OLARAK KABUL EDİLEBİLECEK MAKSİMUM BAĞLANTI SAYISINI HESAPLAYAN BENZETİM PROGRAMLARI

Bu bölümde, PASCAL programlama dilinde yazılmış, kullanıcı tarafından girilen kaynağın parametrelerine, benzetim süresine, tampon boyutuna, çıkış portunun hızına, sistemde sağlanması istenen hücre kayıp olasılığına ve yaşam süresi seçeneğine göre, kabul edilebilecek maksimum bağlantı sayısını hesaplayan üç farklı benzetim programı anlatılacaktır. Bu benzetim programlarının akış diyagramları, kaynak kodları ve bu programların belli değerler için çalıştırılması sonucu elde edilen sonuçlar verilecektir.

Yazılan benzetim programlarında, bağlantı kurma isteğinde bulunan her bir kaynağın, “ON-OFF” kaynak modeline göre iletimde bulunacağı varsayılmıştır. Bu modele göre kaynağın, periyod süresi boyunca hücrelerini tepe hızıyla gönderdiği bir ON periyodu (aktif periyod), ve hiç hücre göndermediği bir OFF periyodu (boş periyod) vardır (Şekil 6.1)



Şekil 6.1 Benzetim Programlarında Kullanılan ATM Kaynak Modeli

Birinci benzetim programında, bağlantı kurma isteğinde bulunan ON-OFF tipi bütün kaynakların aynı anlarda iletime başladıkları varsayılmıştır (Şekil 6.2).

İkinci benzetim programında ise, kaynakların belli bir aralıkta rastgele bir biçimde iletime başladıkları varsayılmıştır (Şekil 6.3 ve Şekil 6.4).

İlk iki programda, kaynakların ON ve OFF periyodlarının süresi sabittir ve kaynak iletimde olduğu süre boyunca değişmez. Bu programlarda ON ve OFF periyodlarının süresi, kullanıcı tarafından girilen ortalama değerlerine (Ton ve Toff) eşittir.

Üçüncü benzetim programında ilk iki benzetim programından farklı olarak, her bir kaynağın ON ve OFF periyodlarının süreleri değişkendir. Bu programda da kaynakların ikinci benzetim programında olduğu gibi, belli bir aralıkta rastgele bir biçimde iletme başladıkları varsayılmıştır (Şekil 6.5 ve Şekil 6.6).

Ayrıca benzetim programlarında yaşam süresi kavramı da göz önüne alınmıştır. Yaşam süresi, bağlantının kurulduktan sonra ne kadar süre devam edeceğini gösteren değerdir. Yaşam süresinin sonsuz olması durumunda ise, sistemde kalıcı bağlantıların kurulduğu varsayılmıştır.

Benzetim programlarında bağlantı kurma isteğinde bulunan bütün kaynakların aynı tip kaynak olduğu varsayılmıştır.

İlk iki benzetim programı, “smooth” kaynaklar için yazılmıştır. Üçüncü benzetim programında ise “yığınsal” (bursty) kaynaklar göz önüne alınmıştır.

Benzetim programları yazılırken Şekil 5.2’ deki kuyruklama modeli göz önüne alınmıştır. Bu kuyruklama modeli Bölüm 5.1.2’ de belirtildiği gibi, bir çıkış portu ve tamponu ATM anahtarının geri kalan kısmından ayrılarak elde edilir. Bu modelde olduğu gibi benzetim programlarında da, bir çıkış portunun hizmet verdiği sonlu kapasiteli bir tamponun, ATM kaynakları tarafından beslendiği varsayılmıştır. Çıkış portunun servis süresi sabittir.

Yazılan benzetim programlarında ilk olarak, girilen kaynak parametrelerine ve programdan programa değişebilen kaynak modellerine (kaynakların iletime başlama şekilleri ve ON-OFF sürelerinin sabit veya değişken olması) göre, N tane kaynağın benzetim süresi boyunca yolladığı toplam hücre sayısı hesaplanır. Daha sonra tampondan atılan hücre sayısı hesaplanır.

Benzetim süresi boyunca tampona giren hücre sayısından, çıkış portunun işleme hızıyla benzetim süresince tampondan çıkan hücre sayısı çıkartılınca geriye kalan, tampondaki hücre sayısı ile atılan hücrelerin sayısının toplamıdır. Bu toplamdan tamponun alabileceği maksimum hücre sayısı (tampon boyutu) çıkarılarak atılan hücre sayısı doğrudan elde edilir.

Daha sonra da, aşağıdaki formülden yararlanarak hücre kayıp oranı bulunur :

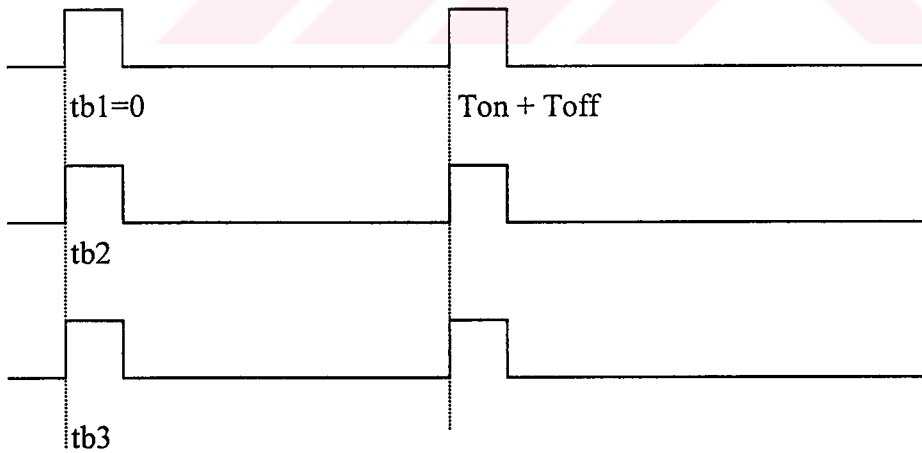
Hücre Kayıp Oranı = Atılan Hücre Sayısı / Yollanan Toplam Hücre Sayısı

Elde edilen hücre kayıp oranı, kullanıcı tarafından girilen kabul edilebilecek hücre kayıp olasılığı ile karşılaştırılır. Hesaplanan hücre kayıp oranı, istenen hücre kayıp olasılığından küçük olduğu sürece kaynak sayısı artırılır ve bu durumdaki hücre kayıp oranını elde etmek için aynı hesaplamalar tekrar yapılır. Hücre kayıp oranı, istenen hücre kayıp olasılığından büyük çıktığında kabul edilebilecek maksimum bağlantı sayısı, N tane kaynak için hesaplama yapıldığı varsayılırsa, N-1' e eşit olacaktır. Son olarak, kabul edilebilecek maksimum bağlantı sayısı için bulunan N-1 değeri ekrana yazdırılır.

Benzetim programları; Bölüm 6.1, Bölüm 6.2, ve Bölüm 6.3' de ayrıntılı olarak anlatılmıştır.

6.1 Kaynakların Aynı Anda İletime Başladığı ve ON-OFF Sürelerinin Sabit Olduğu Durum İçin Yazılan Benzetim Programı

Bu benzetim programında Şekil 6.2' de görüldüğü gibi, bütün kaynakların aynı anda ilettime başladıkları varsayılmıştır. Şekildeki tbi (i=1,2,3) değerleri i. kaynağın ilettime başladığı anı, Ton değeri kaynağın on periyodunun süresini, Toff değeri ise kaynağın off periyodunun süresini gösterir.



Şekil 6.2 Birinci benzetim programında kaynakların ilettime başlama şekli
(Bütün kaynakların aynı anda ilettime başladıkları varsayılıyor)

6.1.1 Kaynakların Aynı Anda İletime Başladığı ve ON-OFF Sürelerinin Sabit Olduğu Durum İçin Yazılan Benzetim Programının Kaynak Kodu

Kaynak kodu aşağıda verilen benzetim programı; kullanıcı tarafından girilen kaynağın parametrelerine, benzetim süresine, tampon boyutuna, çıkış portunun hızına, sistemde sağlanması istenen hücre kayıp olasılığına ve yaşam süresi seçeneğine göre kabul edilebilecek maksimum bağlantı sayısını hesaplar.

Bu benzetim programında kaynakların Şekil 6.2' deki modele göre ilettime başladıkları varsayılmıştır. Programın çalıştırılmasıyla ilk ekranda kullanıcıdan sırasıyla çıkış portunun hızı (trunk band genişliği) W (hücre/s), kaynağın tepe hızı Rpeak (hücre/s), kaynağın aktif olduğu ortalama süre ton (s), kaynağın aktif olmadığı ortalama süre toff (s), benzetim süresi tsim (s), tampon boyutu x (hücre) ve kabul edilebilir hücre kayıp olasılığı Epsilon' u girmesi istenir. İkinci ekranda kullanıcıya bağlantının yaşam süresi ile ilgili iki seçenek sunulur : Birinci seçenekte bağlantının yaşam süresi sonsuz kabul edilir, ikinci seçenekte ise bağlantının belirli bir yaşam süresi vardır.

Eğer kullanıcı birinci seçeneği, yani yaşam süresini sonsuz seçmişse, girilen değerlere göre hesaplanan maksimum kabul edilebilecek bağlantı sayısı ve periyodun sonunda tamponda kalan hücre sayısı ekrana yazdırılır.

Eğer kullanıcı ikinci seçeneği seçmişse, üçüncü bir ekranda kullanıcıya yaşam süresinin değeri sorulur. Girilen değerlere göre hesaplanan kabul edilebilecek maksimum bağlantı sayısı, periyodun sonunda tamponda kalan hücre sayısı ve aşağıdaki biçimde hesaplanan B değeri ekrana yazdırılır.

$$B = \frac{\text{Tampon Boyutu}}{\text{Yaşam Süresi} / (\text{ON Periyodun Süresi} + \text{OFF Periyodun Süresi})} \quad (6.1)$$

Bir periyodluk süre sonunda tamponda bulunan hücre sayısı ile B değerini karşılaştırılabilmek amacıyla B değeri hesaplanmıştır. Bu programda yaşam süresinin sonlu olması ile sonsuz olması arasındaki tek fark, yaşam süresi belirli olduğunda B değerinin hesaplanmasıdır.

{BU PROGRAM KAYNAKLARIN AYNI ANDA İLETİME GEÇMESİ DURUMU İÇİN KABUL EDİLEBİLECEK MAKSİMUM BAĞLANTI SAYISINI HESAPLAR.}

program simulasyon_durum_1;

uses crt;

var

N : integer;

W, Rpeak, ton, toff, tsim, x, Epsilon, yollanan, tyasam, per_tampon, B : real;

CLP : boolean;

ch : char;

{BU ALT PROGRAM, N TANE KAYNAĞIN AYNI ZAMANLARDA İLETİME BAŞLADIĞI DURUM İÇİN, SİMULASYON SÜRESİ BOYUNCA N TANE KAYNAĞIN YOLLADIĞI TOPLAM HÜCRE SAYISINI HESAPLAR}

function yol_hucre_say : real;

var

periyod_sayisi : integer;

Tp, artik_periyod, periyod_bit : real;

begin

 Tp := toff + ton;

 if tsim <= Tp then

 if tsim <= ton then

 yol_hucre_say := Rpeak * N * tsim

 else {t > ton}

 yol_hucre_say := Rpeak * N * ton

 else {tsim > Tp}

 begin

 periyod_sayisi := trunc(tsim / Tp);

 artik_periyod := tsim - (periyod_sayisi * Tp);

 periyod_bit := (Rpeak * N * ton) * periyod_sayisi;

 if artik_periyod >= ton then

 yol_hucre_say := periyod_bit + (Rpeak * N * ton)

 else {artik_periyod < ton}

 yol_hucre_say := periyod_bit + (Rpeak * N * artik_periyod);

 end; {tsim <= Tp}

end;

{BU ALT PROGRAM, N TANE KAYNAĞIN AYNI ZAMANLARDA İLETİME BAŞLADIĞI DURUM İÇİN, ATILAN HÜCRE SAYISINI HESAPLAR}

{W : ÇIKIŞ PORTUNUN HIZI}

function at_hucre_sayisi : real;

begin

 yollanan := yol_hucre_say ;

```

if ( ( yollanan - ( W * tsim ) ) > x ) then
    at_hucre_sayisi := ( yollanan - ( W * tsim ) ) - x
else
    at_hucre_sayisi := 0;
end;

```

{BU ALT PROGRAM, HESAPLANAN HÜCRE KAYIP ORANI, İSTENEN HÜCRE KAYIP OLASILIĞI EPSILON' DAN KÜÇÜK VEYA EŞİTSE FALSE, BÜYÜKSE TRUE DEĞERİNİ DÖNDÜRÜR.}

```

function cell_loss : boolean;

```

```

var
    clp2, atilan_say : real;

```

```

begin

```

```

    atilan_say := at_hucre_sayisi ;

```

```

    clp2 := atilan_say / yollanan;

```

```

    if clp2 <= Epsilon then
        cell_loss := false
    else
        cell_loss := true;
    end;

```

```

{Ana Program}
begin

```

```

    clrscr;
    write( ' Trunk band genisligini girin (hucre/s) : ' );
    readln(W);
    writeln;
    write( ' Kaynagin tepe hizini girin (hucre/s) : ' );
    readln(Rpeak);
    write( 'Kaynagin aktif oldugu ortalama sure (ton) : ' );
    readln(ton);
    write( 'Kaynagin aktif olmadigi ortalama sure (toff) : ' );
    readln(toff);
    write( ' Simulasyon suresini girin : ' );
    readln(tsim);
    write( ' Tampon boyutunu girin (hucre) : ' );
    readln(x);
    write( ' Istenen Hucre kayip olasiligi Epsilon : ');
    readln(Epsilon);

```

```

    clrscr;

```

```

    writeln( ' Lutfen Baglantinin Yasam Suresi Icin Asagidaki Seceneklerden Birini Girin ' );
    writeln;writeln;
    writeln( ' 1-) Baglantinin yasam suresi sonsuz kabul ediliyor ' );

```

```

writeln( ' 2-) Baglantinin belli bir yasam suresi var ' );
writeln;writeln;
write( ' Seciminiz : ' );
readln(ch);

if ch = '2' then
begin
write( ' Yasam suresini girin : ' );
readln(tyasam);
B := x / ( tyasam / ( ton + toff ) );
end;

N := 1;

CLP := cell_loss ;
while ( not ( CLP ) ) do
begin
N := N + 1;
CLP := cell_loss ;
end;

{Hesaplanan kabul edilebilecek max. bağlantı sayısı N-1 degeri için
periyod sonunda tampondaki hucre sayısı;}
per_tampon := ( Rpeak * ton * (N-1) ) - ( W * ( ton + toff ) );

if per_tampon < 0 then
per_tampon := 0;

writeln( ' Maksimum kabul edilebilecek baglanti sayisi = ', N-1 );
writeln;
writeln( 'Periyodun sonunda tampondaki hucre sayisi = ', per_tampon );
writeln;

if ch = '2' then
begin
writeln( 'B sayisi = ', B );
end;

readln;
end.

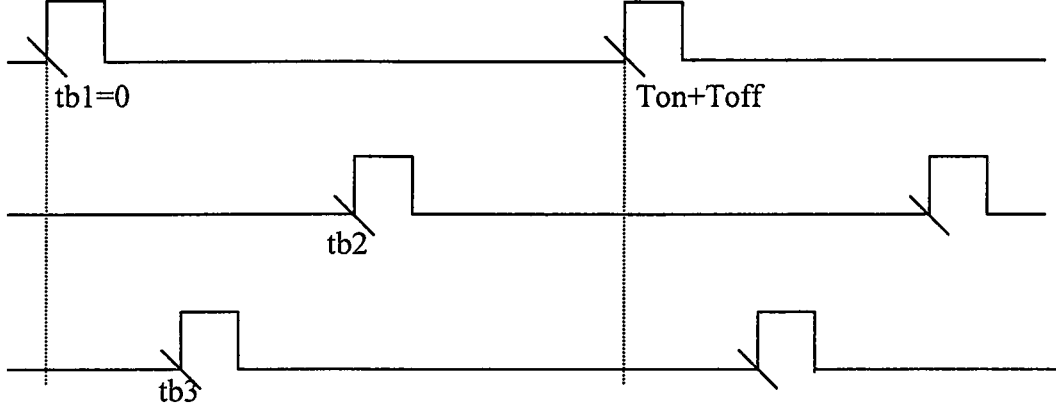
```

6.2 Kaynakların Belli Bir Aralıkta Rastgele İletime Başladığı ve ON-OFF Sürelerinin Sabit Olduğu Durum İçin Yazılan Benzetim Programı

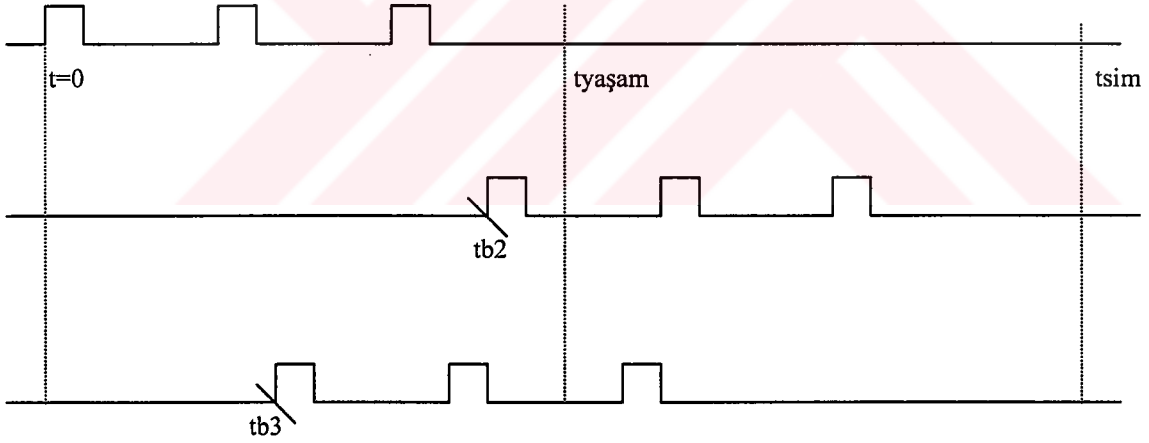
Bu benzetim programında iki durum sözkonusudur :

1. Bütün kaynaklar Şekil 6.3’de gösterildiği gibi bir periyodluk süre içerisinde rastgele iletme başlarlar. Bu durum, kullanıcı tarafından yaşam süresinin sonsuz seçilmesine karşı düşer.

2. Kaynaklar Şekil 6.4’de gösterildiği gibi yaşam süresinin içinde rastgele ilettime başlayacak şekilde dağılırlar ve herbir kaynağın benzetim süresi içerisinde iletimde kalma süresi, kullanıcı tarafından girilen yaşam süresine eşittir. Bu durumda benzetim süresi de yaşam süresinin iki katına eşittir.



Şekil 6.3 İkinci benzetim programında kaynakların periyod süresi ($Ton+Toff$) içinde rastgele ilettime başlaması durumu. (Yaşam süresinin sonsuz seçilmesi durumu).



Şekil 6.4 İkinci benzetim programında kaynakların girilen yaşam süresi içinde rastgele ilettime başlaması durumu. (Yaşam süresi burada üç periyodluk süre seçilmiş. Benzetim süresi yaşam süresinin iki katına eşit).

Şekil 6.3 ve Şekil 6.4’ deki tbi ($i=1,2,3$) değerleri, i. kaynağın ilettime başladığı anı, Ton değeri kaynağın on periyodunun süresini, $Toff$ değeri ise kaynağın off periyodunun süresini gösterir.

Şekil 6.4’ deki $tyaşam$, yaşam süresinin sona erdiği anı, $tsim$ ise benzetim süresinin sona erdiği anı gösterir.

6.2.1 Kaynakların Belli Bir Aralıkta Rastgele İletime Başladığı ve ON-OFF Sürelerinin Sabit Olduğu Durum İçin Yazılan Benzetim Programının Kaynak Kodu

Kaynak kodu aşağıda verilen benzetim programı, kullanıcı tarafından girilen kaynağın parametrelerine, benzetim süresine, tampon boyutuna, çıkış portunun hızına, sistemde sağlanması istenen hücre kayıp olasılığına ve yaşam süresi seçeneğine göre kabul edilebilecek maksimum bağlantı sayısını hesaplar.

Birinci benzetim programından farklı olarak bu benzetim programında kaynakların Şekil 6.3 veya Şekil 6.4’ deki modele göre ilettime başladıkları varsayılmıştır. Programın çalıştırılmasıyla ilk ekranda kullanıcıdan sırasıyla çıkış portu hızı (trunk band genişliği) W (hücre/s), kaynağın tepe hızı R_{peak} (hücre/s), kaynağın aktif olduğu ortalama süre t_{on} (s), kaynağın aktif olmadığı ortalama süre t_{off} (s), tampon boyutu x (hücre) ve kabul edilebilir hücre kayıp olasılığı $Epsilon$ ’ u girmesi istenir. İkinci ekranda kullanıcıya bağlantının yaşam süresi ile ilgili iki seçenek sunulur : Birinci seçenekte bağlantının yaşam süresi sonsuz kabul edilir, ikinci seçenekte ise bağlantının belirli bir yaşam süresi vardır.

Eğer kullanıcı birinci seçeneği, yani yaşam süresini sonsuz seçmişse, Şekil 6.3’ deki modele göre bütün kaynakların bir periyodluk süre içerisinde rastgele bir biçimde ilettime başladıkları varsayılır ve kullanıcıdan benzetim süresi t_{sim} (s)’ i girmesi istenir. Girilen değerlere göre hesaplanan kabul edilebilecek maksimum bağlantı sayısı ekrana yazdırılır.

Eğer kullanıcı ikinci seçeneği seçmişse, Şekil 6.4’ deki modele göre bütün kaynakların yaşam süresi içerisinde rastgele bir biçimde ilettime başladıkları ve bu süre boyunca iletimde kaldıkları varsayılır ve kullanıcıdan yaşam süresi t_{yasa} (s)’ ı girmesi istenir. Bu durumda benzetim süresi yaşam süresinin iki katına eşittir. Girilen değerlere göre hesaplanan kabul edilebilecek maksimum bağlantı sayısı ekrana yazdırılır.

{BU PROGRAM, KAYNAKLARIN BELLİ BİR ARALIKTA RASTGELE İLETİME BAŞLAMASI DURUMU İÇİN KABUL EDİLEBİLECEK MAKSİMUM BAĞLANTI SAYISINI HESAPLAR.}

{PROGRAMDA KULLANICIYA YAŞAM SÜRESİ İÇİN İKİ SEÇENEK SUNULUR:

1-) BAĞLANTININ YAŞAM SÜRESİ SONSUZ.

2-) BAĞLANTININ BELİRLİ BİR YAŞAM SÜRESİ VAR.

KULLANICI 1. SEÇENEĞİ SEÇERSE, KAYNAKLARIN BİR PERİYOD (Ton+Toff) SÜRESİ İÇİNDE RASTGELE İLETİME BAŞLADIKLARI VARSAYILIR. KULLANICIDAN SİMULASYON SÜRESİNİ GİRMESİ İSTENİR.

KULLANICI 2. SEÇENEĞİ SEÇERSE, KAYNAKLARIN YAŞAM SÜRESİ İÇİNDE RASTGELE İLETİME BAŞLADIKLARI VE BU SÜRE BOYUNCA İLETİMDE KALDIKLARI VARSAYILIR. KULLANICIDAN YAŞAM SÜRESİNİ GİRMESİ İSTENİR. SİMULASYON SÜRESİ İSE YAŞAM SÜRESİNİN İKİ KATI BÜYÜKLÜĞÜNDE ALINIR.

KAYNAK ÖRNEĞİN T=0 ANINDA İLETİME BAŞLAMIŞSA T=TYASAM ANINDA İLETİMİNE SON VERİR, T=1 ANINDA İLETİME BAŞLAMIŞSA T=TYASAM+1 ANINDA İLETİMİNE SON VERİR.}

```
program simulasyon_durum_2;
```

```
uses crt;
```

```
var
```

```
  N, periyod_sayisi, p_sayisi, i : integer;
```

```
  tper : longint;
```

```
  W, Rpeak, x, Epsilon, ton, toff, tsim, tyasam, artik_periyod, yollanan : real;
```

```
  CLP : boolean;
```

```
  tbas, tson : array [1..1000] of real;
```

```
  ch : char;
```

{BU ALT PROGRAM, N TANE KAYNAĞIN PERİYOD SÜRESİ (Ton+Toff) İÇERİSİNDE RASTGELE ZAMANLARDA İLETİME BAŞLADIĞI DURUM İÇİN, N TANE KAYNAĞIN YOLLADIĞI TOPLAM HÜCRE SAYISINI HESAPLAR}

```
function yol_hucre_say : real;
```

```
var
```

```
  Tp, yol_top : real;
```

```
  Ntemp : integer;
```

```
begin
```

```
  Tp := toff + ton;
```

```
  yol_top := 0;
```

```
  Ntemp := N;
```

```
repeat
```

```
begin
```

```
  if tsim <= Tp then
```

```
  begin
```

```
    if tbas[Ntemp] <= tsim then
```

```
    begin
```

```
      if tson[Ntemp] <= tsim then
```

```
        yol_top := yol_top + ( Rpeak * ton )
```

```
      else {tson[N] > t}
```

```
        yol_top := yol_top + ( Rpeak * ( tsim - tbas[Ntemp] ) );
```

```
    end;
```

```
  end
```

```
  else {t > Tp}
```

```
  begin
```

```
    if tbas[Ntemp] <= artik_periyod then
```

```

begin
  if tson[Ntemp] <= artik_periyod then
    yol_top := ( yol_top + ( Rpeak * ton ) ) + ( periyod_sayisi * ( Rpeak * ton ) )
  else {tson[N] > artik_periyod}
    yol_top := ( yol_top + ( Rpeak * ( artik_periyod - tbas[Ntemp] ) ) )
    + ( periyod_sayisi * ( Rpeak * ton ) );
  end
  else
  begin
    yol_top := yol_top + ( periyod_sayisi * ( Rpeak * ton ) );
  end;
end; {else} {t>Tp}
Ntemp := Ntemp - 1;
end; {repeat}
until Ntemp = 0;

yol_hucre_say := yol_top;
end;

```

{BU ALT PROGRAM, N TANE KAYNAĞIN YAŞAM SÜRESİ İÇERİSİNDE RASTGELE ZAMANLARDA İLETİME GEÇTİĞİ DURUM İÇİN, N TANE KAYNAĞIN YOLLADIĞI TOPLAM HÜCRE SAYISINI HESAPLAR}

```

function ys_yol_hucre_say : real;
begin
  {Bütün kaynaklar, simulasyon süresinin sonuna kadar iletimlerini
  tamamlamış olurlar.}
  {p_sayisi global olarak hesaplanır.}

  ys_yol_hucre_say := N * ( p_sayisi * ( Rpeak * ton ) );
end;

```

{BU ALT PROGRAM, N TANE KAYNAĞIN YAŞAM SÜRESİ VEYA BİR PERİYODLUK SÜRE İÇERİSİNDE RASTGELE ZAMANLARDA İLETİME GEÇTİĞİ DURUM İÇİN, ATILAN HÜCRE SAYISINI HESAPLAR}

{W : ÇIKIŞ PORTUNUN HIZI}

```

function at_hucre_sayisi : real;
begin
  if ch = '1' then
    yollanan := yol_hucre_say
  else
    yollanan := ys_yol_hucre_say ;

  if ( ( yollanan - ( W * tsim ) ) > x ) then
    at_hucre_sayisi := ( yollanan - ( W * tsim ) ) - x
  else
    at_hucre_sayisi := 0;
  end;
end;

```

{BU ALT PROGRAM, HESAPLANAN HÜCRE KAYIP ORANI İSTENEN HÜCRE KAYIP OLASILIĞI EPSILON' DAN KÜÇÜKSE FALSE, BÜYÜKSE TRUE DEĞERİNİ DÖNDÜRÜR}

function cell_loss : boolean;

var

clp2, atilan_say : real;

begin

atilan_say := at_hucre_sayisi ;

clp2 := atilan_say / yollanan;

if clp2 <= Epsilon then

cell_loss := false

else

cell_loss := true;

end;

{Ana Program}

begin

clrscr;

write(' Trunk band genisligini girin (hucre/s): ');

readln(W);

writeln;

write(' Kaynagin tepe hizini girin (hucre/s) : ');

readln(Rpeak);

write(' Kaynagin aktif oldugu ortalama sure (ton) : ');

readln(ton);

write(' Kaynagin aktif olmadigi ortalama sure (toff) : ');

readln(toff);writeln;

write(' Tampon boyutunu girin (hucre) : ');

readln(x);

write(' Istenen Hucre kayip olasiligi Epsilon : ');

readln(Epsilon);

clrscr;

writeln(' Lutfen Baglantinin Yasam Suresi Icin Asagidaki Seceneklerden Birini Girin');

writeln;writeln;

writeln(' 1-) Baglantinin yasam suresi sonsuz kabul ediliyor');

writeln(' 2-) Baglantinin belli bir yasam suresi var');

writeln;writeln;

write(' Seciminiz : ');

readln(ch);

if ch = '2' then

begin

writeln(' Simulasyon suresi yasam suresinin 2 kati alinacaktır.');

writeln;

```

    write(' Yasam suresini girin : ');
    readln(tyasam);
    tsim := 2 * tyasam;
end
else
begin
    write(' Simulasyon suresini girin : ');
    readln(tsim);
end;
writeln;writeln;

p_sayisi := trunc( tyasam / (ton + toff) );

N := 1;

tbas[1] := 0;
tson[1] := ton;
tper := round(ton+toff);

if tsim > (ton+toff) then
begin
    periyod_sayisi := trunc(tsim / (ton + toff));
    artik_periyod := tsim - (periyod_sayisi * (ton + toff));
end;

CLP := cell_loss ;

while ( not ( CLP ) ) do
begin
    N := N + 1;
    if ch = '1' then
    begin
        tbas[N] := random(tper);
        tson[N] := ton + tbas[N];
    end; {if ch='1'}

    CLP := cell_loss ;
end; {while}

writeln(' Maksimum kabul edilebilecek baglanti sayisi = ', N-1);

readln;
end.

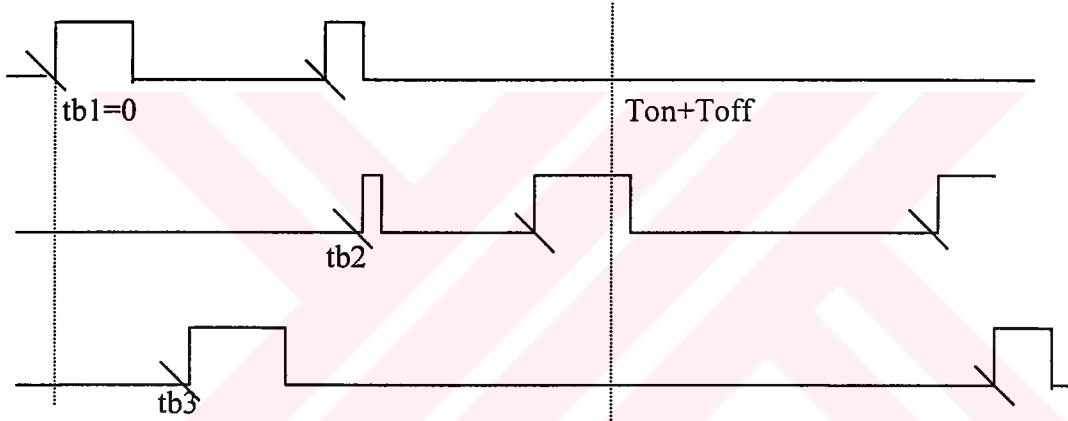
```

6.3 Kaynakların Belli Bir Aralıkta Rastgele İletime Başladığı ve ON-OFF Sürelerinin Üssel Dağılıma Göre Rastgele Değiştiği Durum İçin Yazılan Benzetim Programı

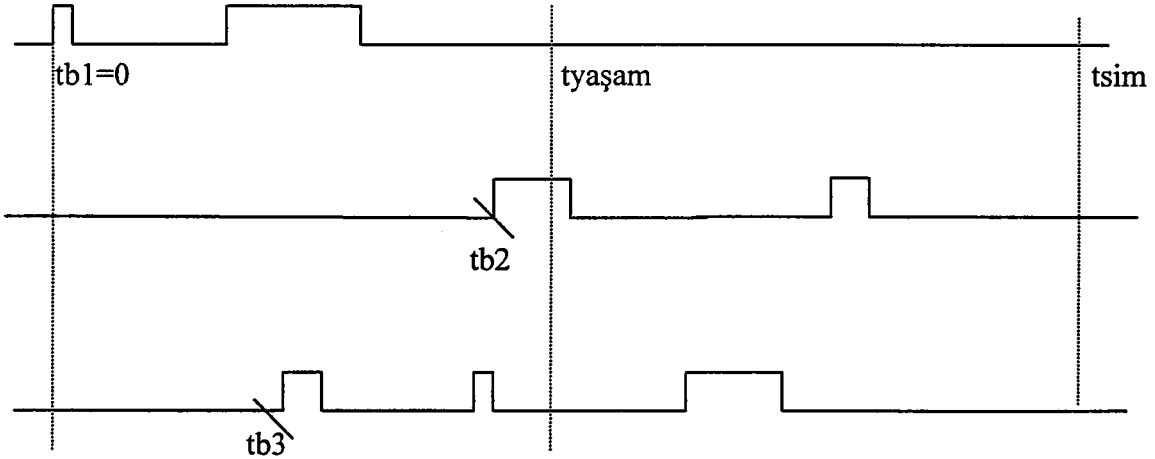
Bu benzetim programında da ikinci benzetim programında olduğu gibi, kaynakların iletime başlaması için iki durum söz konusudur :

1. Bütün kaynaklar Şekil 6.5’de gösterildiği gibi, bir periyodluk süre içerisinde rastgele iletme başlarlar. Bu durum, kullanıcı tarafından yaşam süresinin sonsuz seçilmesine karşı düşer.
2. Kaynaklar Şekil 6.6’da gösterildiği gibi, kullanıcı tarafından girilen yaşam süresinin içinde rastgele iletme başlayabilirler. Bu durumda herbir kaynağın benzetim süresi içerisinde iletimde kalma süresi, kullanıcı tarafından girilen yaşam süresine eşittir. Benzetim süresi de yaşam süresinin iki katına eşittir.

Bu benzetim programının ikinci benzetim programından farkı ise, bağlantı isteğinde bulunan ve yukarıda açıklanan iki durumdan birine göre iletme başlayan her bir kaynağın ON ve OFF periyodlarının sürelerinin değişken olmasıdır.



Şekil 6.5 Üçüncü benzetim programında kaynakların periyod süresi içinde rastgele iletme başlaması durumu. (Yaşam süresinin sonsuz seçilmesi durumu).



Şekil 6.6 Üçüncü benzetim programında kaynakların girilen yaşam süresi içinde rastgele iletme başlaması durumu.

IPP (Interrupted Poisson Process) ve IFP (Interrupted Fluid Process) tipi ON-OFF kaynak modellerinde, ON ve OFF periyodlarının süreleri, üssel dağılıma göre değişir [9]. Üssel dağılımın olasılık yoğunluğu fonksiyonu ise :

$$f(x) = b.e^{-bx} \quad (x > 0 \text{ için}) \quad (x < 0 \text{ için } f(x) = 0)$$

şeklinde tanımlanır [24]. Üssel dağılımın ortalama değeri $(1 / b)$ ' ye eşittir.

Benzetim programlarında kullanıcı tarafından girilen Ton değerinin ON periyodunun süresinin ortalama değeri, Toff değerinin OFF periyodunun süresinin ortalama değeri olduğu ve x raslantı değişkeninin ON veya OFF periyodlarının sürelerini gösterdiği dikkate alınır :

$$f(x) = \frac{1}{T_{on}} . e^{-x / T_{on}} \quad (\text{ON periyodu için})$$

$$f(x) = \frac{1}{T_{off}} . e^{-x / T_{off}} \quad (\text{OFF periyodu için})$$

elde edilir. $f(x)$ ise, ON veya OFF periyodunun süresinin x saniye olması olasılığına karşı düşer.

$f(x)$ fonksiyonunun, ON periyodunun süresi için, %95 olasılıkla değer alacağı aralık aşağıdaki biçimde bulunmuştur :

$$\int_0^y \frac{1}{T_{on}} e^{-\frac{1}{T_{on}}x} dx = 0.95$$

$$-e^{-\frac{1}{T_{on}}x} \Big|_{x=0}^y = 0.95$$

$$-e^{-y / T_{on}} + 1 = 0.95$$

$$y = -\ln(0.05) * T_{on}$$

$$y \cong 3 * T_{on}$$

Bu sonuç, ON periyodunun süresinin %95 olasılıkla $(0, 3 \cdot T_{on})$ aralığında bir değer alacağını gösterir.

Yukarıdaki hesaplamalara dayalı olarak, üçüncü benzetim programında her periyodun sonunda, ON periyodunun süresi için $(0, 3 \cdot T_{on})$ aralığında rastgele değerler üretilir. Aynı şekilde programda her ON periyodunun bitiminde, OFF periyodunun süresi için $(0, 3 \cdot T_{off})$ aralığında rastgele değerler üretilir.

6.3.1 Kaynakların Belli Bir Aralıkta Rastgele İletime Başladığı ve ON-OFF Sürelerinin Üssel Dağılıma Göre Rastgele Değiştiği Durum İçin Yazılan Benzetim Programının Kaynak Kodu

Kaynak kodu aşağıda verilen benzetim programı, kullanıcı tarafından girilen kaynağın parametrelerine, benzetim süresine, tampon boyutuna, çıkış portunun hızına, sistemde sağlanması istenen hücre kayıp olasılığına ve yaşam süresi seçeneğine göre kabul edilebilecek maksimum bağlantı sayısını hesaplar.

İkinci benzetim programından farklı olarak bu programda kaynakların ON ve OFF periyodlarının süresi değişkendir. Kaynakların ilettime başlama şekli ise, ikinci benzetim programında olduğu gibidir (Şekil 6.5 ve Şekil 6.6). Programın çalıştırılmasıyla ilk ekranda kullanıcıdan sırasıyla çıkış portu hızı (trunk band genişliği) W (hücre/s), kaynağın tepe hızı R_{peak} (hücre/s), kaynağın aktif olduğu ortalama süre ton (s), kaynağın aktif olmadığı ortalama süre t_{off} (s), tampon boyutu x (hücre) ve kabul edilebilir hücre kayıp olasılığı Epsilon' u girmesi istenir. İkinci ekranda kullanıcıya bağlantının yaşam süresi ile ilgili iki seçenek sunulur : Birinci seçenekte bağlantının yaşam süresi sonsuz kabul edilir, ikinci seçenekte ise bağlantının belirli bir yaşam süresi vardır.

Eğer kullanıcı birinci seçeneği, yani yaşam süresini sonsuz seçmişse, Şekil 6.5' deki modele göre bütün kaynakların bir periyodluk süre içerisinde rastgele bir biçimde ilettime başladıkları varsayılır ve kullanıcıdan benzetim süresi t_{sim} (s)' i girmesi istenir. Girilen değerlere göre hesaplanan kabul edilebilecek maksimum bağlantı sayısı ekrana yazdırılır.

Eğer kullanıcı ikinci seçeneği seçmişse, Şekil 6.6' daki modele göre bütün kaynakların yaşam süresi içerisinde rastgele bir biçimde ilettime başladıkları ve bu süre boyunca iletimde kaldıkları varsayılır ve kullanıcıdan yaşam süresi t_{yasa} (s)' i girmesi istenir. Bu durumda benzetim süresi yaşam süresinin iki katına eşittir. Girilen değerlere göre hesaplanan kabul edilebilecek maksimum bağlantı sayısı ekrana yazdırılır.

Eğer kullanıcı ikinci seçeneği seçmişse, Şekil 6.6' daki modele göre bütün kaynakların yaşam süresi içerisinde rastgele bir biçimde iletme başladıkları ve bu süre boyunca iletimde kaldıkları varsayılır ve kullanıcıdan yaşam süresi t_{yasam} (s)' ı girmesi istenir. Bu durumda benzetim süresi yaşam süresinin iki katına eşittir. Girilen değerlere göre hesaplanan kabul edilebilecek maksimum bağlantı sayısı ekrana yazdırılır.

{BU PROGRAM, KAYNAKLARIN BELLİ BİR ARALIKTA RASTGELE İLETİME BAŞLAMASI DURUMU İÇİN KABUL EDİLEBİLECEK MAKSİMUM BAĞLANTI SAYISINI HESAPLAR.

KAYNAKLARIN ON VE OFF PERİYODLARININ SÜRELERİ DEĞİŞKENDİR. PROGRAMDA KAYNAKLARIN ON VE OFF PERİYODLARININ SÜRELERİ İÇİN (0, 3* T_{on}) VE (0, 3* T_{off}) ARALIKLARINDA RASTGELE DEĞERLER ÜRETİLİR.

PROGRAMDA KULLANICIYA YAŞAM SÜRESİ İÇİN İKİ SEÇENEK SUNULUR:

1-) BAĞLANTININ YAŞAM SÜRESİ SONSUZ.

2-) BAĞLANTININ BELİRLİ BİR YAŞAM SÜRESİ VAR.

KULLANICI 1. SEÇENEĞİ SEÇERSE, KAYNAKLARIN BİR PERİYOD ($T_{on}+T_{off}$) SÜRESİ İÇİNDE RASTGELE İLETİME BAŞLADIKLARI VARSAYILIR. KULLANICIDAN SİMULASYON SÜRESİNİ GİRMESİ İSTENİR.

KULLANICI 2. SEÇENEĞİ SEÇERSE, KAYNAKLARIN YAŞAM SÜRESİ İÇİNDE RASTGELE İLETİME BAŞLADIKLARI VE BU SÜRE BOYUNCA İLETİMDE KALDIKLARI VARSAYILIR. KULLANICIDAN YAŞAM SÜRESİNİ GİRMESİ İSTENİR. SİMULASYON SÜRESİ İSE YAŞAM SÜRESİNİN İKİ KATI BÜYÜKLÜĞÜNDE ALINIR.}

program simulasyon_durum_3;

uses crt;

var

N : integer;

tper1, tper2, rton, rtoff : longint;

W, Rpeak, x, Epsilon, ton, toff, tsim, tyasam, yollanan : real;

CLP : boolean;

tbas, tond, toffd : array [1..1000] of real;

ch : char;

{BU ALT PROGRAM, N TANE KAYNAĞIN PERİYOD SÜRESİ ($T_{on}+T_{off}$) İÇERİSİNDE RASTGELE ZAMANLARDA İLETİME BAŞLADIĞI DURUM İÇİN, N TANE KAYNAĞIN YOLLADIĞI TOPLAM HÜCRE SAYISINI HESAPLAR }

function yol_hucre_say : real;

var

yol_top, on_topla, off_topla, ara_top, artik_periyod : real;

Ntemp, periyod_sayisi, i, j : integer;

```
dongu_cik : boolean;
```

```
begin
```

```
  yol_top := 0;
```

```
  Ntemp := N;
```

```
repeat
```

```
begin
```

```
  periyod_sayisi := 1;
```

```
  i := 1;
```

```
  on_topla := 0;
```

```
  off_topla := 0;
```

```
  dongu_cik := false;
```

```
  tond[i] := random(rton);
```

```
  toffd[i] := random(rtoff);
```

```
  ara_top := tbas[Ntemp] + tond[i] + toffd[i];
```

{Aşağıdaki döngüde iletme başlama anı farklı ve ON periyodunun süresi değişken olan her bir kaynak için simulasyon süresi içinde ON periyodlarının toplam süresi (on_topla) ve artık_periyod hesaplanıyor}

```
repeat
```

```
begin
```

```
  i := i + 1;
```

{ara_top degiskeni, on-off süreleri üretilirken simulasyon süresinin sonuna gelip gelmediğimizi anlamak için kullanılıyor.}

```
  if tsim > ara_top then
```

```
  begin
```

```
    tond[i] := random(rton);
```

```
    ara_top := ara_top + tond[i];
```

```
    if tsim > ara_top then
```

```
    begin
```

```
      toffd[i] := random(rtoff);
```

```
      ara_top := ara_top + toffd[i];
```

```
      if ara_top <= tsim then
```

```
        periyod_sayisi := periyod_sayisi + 1;
```

```
    end
```

```
  else {tsim > ara_top}
```

```
  begin
```

```
    for j := 1 to periyod_sayisi do
```

```
    begin
```

```
      on_topla := on_topla + tond[j];
```

```
      off_topla := off_topla + toffd[j];
```

```
    end;
```

```
    artik_periyod := tsim - (on_topla + off_topla + tbas[Ntemp]);
```

```
    dongu_cik := true;
```

```
  end;
```

```
end
```

```
else {tsim > ara_top}
```

```
begin
```

```
  for j := 1 to periyod_sayisi do
```

```
  begin
```

```
    on_topla := on_topla + tond[j];
```

```

        off_topla := off_topla + toffd[j];
    end;
    artik_periyod := tsim - (on_topla + off_topla + tbas[Ntemp]);
    dongu_cik := true;
end;
end; {repeat}
until dongu_cik;

if artik_periyod > 0 then
begin
    if tond[i] <= artik_periyod then
        yol_top := (yol_top + (Rpeak * tond[i-1])) + (on_topla * Rpeak)
    else { tond[i] > artik_periyod }
        yol_top := (yol_top + (Rpeak * artik_periyod)) + (on_topla * Rpeak);
    end
    else {artik periyod yok}
        yol_top := yol_top + (on_topla * Rpeak);

    Ntemp := Ntemp - 1;
end; {repeat}
until Ntemp = 0;

yol_hucre_say := yol_top;
end;

{BU ALT PROGRAM, N TANE KAYNAĞIN YAŞAM SÜRESİ İÇERİSİNDE
RASTGELE ZAMANLARDA İLETİME GEÇTİĞİ DURUM İÇİN, N TANE KAYNAĞIN
YOLLADIĞI TOPLAM HÜCRE SAYISINI HESAPLAR }

function ys_yol_hucre_say : real;
{Bütün kaynaklar, simulasyon süresinin sonuna kadar iletimlerini
tamamlamış oluyorlar.}

var
    yol_top, on_topla, off_topla, ara_top, artik_periyod : real;
    Ntemp, periyod_sayisi, i, j, on_periyod : integer;
    dongu_cik : boolean;

begin
    yol_top := 0;
    Ntemp := N;

    repeat
    begin
        periyod_sayisi := 1;
        on_periyod := 1;
        i := 1;
        on_topla := 0;
        off_topla := 0;
        dongu_cik := false;

        tond[i] := random(rton);
        toffd[i] := random(rtoff);

```

```

ara_top := tond[i] + toffd[i];

{Aşağıdaki döngüde iletme başlama anı farklı ve ON periyodunun
süresi değişken olan her bir kaynak için YAŞAM SÜRESİ
içinde ON periyodlarının toplam süresi (on_topla) hesaplanıyor}
repeat
begin
i := i + 1;
{ara_top degiskeni, on-off süreleri üretilirken YAŞAM süresinin
sonuna gelip gelmedigimizi anlamak için kullanılıyor.}
if tyasam > ara_top then
begin
tond[i] := random(rton);
on_periyod := periyod_sayisi + 1;

ara_top := ara_top + tond[i];
if tyasam > ara_top then
begin
toffd[i] := random(rtoff);

ara_top := ara_top + toffd[i];
if ara_top <= tyasam then
periyod_sayisi := periyod_sayisi + 1;
end
else {tyasam <= ara_top}
begin
for j := 1 to on_periyod do
on_topla := on_topla + tond[j];

dongu_cik := true;
end;
end
else {tsim > ara_top}
begin
for j := 1 to on_periyod do
on_topla := on_topla + tond[j];

dongu_cik := true;
end;
end; {repeat}
until dongu_cik;

yol_top := yol_top + (on_topla * Rpeak);
Ntemp := Ntemp - 1;
end; {repeat}
until Ntemp = 0;

ys_yol_hucre_say := yol_top;
end;

```

{BU ALT PROGRAM, N TANE KAYNAĞIN YAŞAM SÜRESİ VEYA BİR PERİYODLUK SÜRE İÇERİSİNDE RASTGELE ZAMANLARDA İLETİME GEÇTİĞİ DURUM İÇİN, TAMPONDAN ATILAN HÜCRE SAYISINI HESAPLAR}

{W :ÇIKIŞ PORTUNUN HIZI}

function at_hucre_sayisi : real;

```
begin
  if ch = '1' then
    yollanan := yol_hucre_say
  else
    yollanan := ys_yol_hucre_say ;

  if ((yollanan - (W * tsim)) > x) then
    at_hucre_sayisi := (yollanan - (W * tsim)) - x
  else
    at_hucre_sayisi := 0;
end;
```

{BU ALT PROGRAM, HESAPLANAN HÜCRE KAYIP ORANI İSTENEN HÜCRE KAYIP OLASILIĞI EPSILON' DAN KÜÇÜKSE FALSE, BÜYÜKSE TRUE DEĞERİNİ DÖNDÜRÜR }

function cell_loss : boolean;

```
var
  clp2, atilan_say : real;

begin

  atilan_say := at_hucre_sayisi ;

  clp2 := atilan_say / yollanan ;

  if clp2 <= Epsilon then
    cell_loss := false
  else
    cell_loss := true;
end;
```

{Ana Program}

```
begin

  clrscr;
  write('Trunk band genisligini girin (hucre/sn): ');
  readln(W);
  writeln;
  write('Kaynagin tepe hizini girin (hucre/sn) : ');
  readln(Rpeak);
  write('Kaynagin aktif oldugu ortalama sure (ton) : ');
  readln(ton);
  write('Kaynagin aktif olmadigi ortalama sure (toff) : ');
  readln(toff);writeln;
```

```

write('Tampon boyutunu girin (hucre) : ');
readln(x);
write('Istenen Hucre kayip olasiligi Epsilon = ');
readln(Epsilon);

clrscr;

writeln('Lutfen Baglantinin Yasam Suresi Icin Asagidaki Seceneklerden Birini Girin');
writeln;writeln;
writeln('1-) Baglantinin yasam suresi sonsuz kabul ediliyor');
writeln('2-) Baglantinin belli bir yasam suresi var');
writeln;writeln;
write('Seciminiz : ');
readln(ch);

if ch = '2' then
begin
    writeln('Simulasyon suresi yasam suresinin 2 kati alinacaktır. ');
    writeln;
    write('Yasam suresini girin : ');
    readln(tyasam);
    tsim:=2*tyasam;
end
else
begin
    write('Simulasyon suresini girin : ');
    readln(tsim);
end;

writeln;writeln;

N:=1;
tbas[1]:=0;
rton:=round(3*ton);
rtoff:=round(3*toff);
tper1:=round(ton+toff);
tper2:=round(tyasam);

CLP := cell_loss ;

while ( not ( CLP ) ) do
begin
    N := N + 1;
    if ch='1' then
        tbas[N]:=random(tper1)
    else
        tbas[N]:=random(tper2);

    CLP := cell_loss ;
end; {while}

writeln(' Maksimum kabul edilebilecek baglanti sayisi = ', N-1);

readln;
end.

```

6.4 Benzetim Programlarının Belli Değerler İçin Yürütülmesi Sonucu Elde Edilen Sonuçlar

Bu bölümde, yazılan benzetim programlarının çeşitli kaynak parametreleri, çıkış portu hızı, tampon boyutu, benzetim süresi ve yaşam süresi değerlerinde çalıştırılması sonucu elde edilen kabul edilebilecek maksimum bağlantı sayısı değerleri incelenmiştir. Benzetim programlarının yapısı gereği, bağlantı kurma isteğinde bulunan kaynakların hepsinin parametrelerinin aynı olduğu varsayılmıştır.

Şekil 6.2' deki modelde gösterildiği gibi, kaynakların hepsinin aynı anda iletme geçtiği varsayımı altında yazılan birinci benzetim programının A tipi kaynak için çalıştırılması sonucu elde edilen değerler Tablo 6.1 ve Tablo 6.2' de verilmiştir.

A tipi kaynak; ON periyodu sırasında tepe hızı 0.0208 hücre/s. hızla ATM hücreleri üreten ON-OFF tipi örnek bir kaynak modelidir. ON ve OFF periyodlarının ortalama süresi ise sırasıyla 9600 ve 54400 s. dir. A tipi kaynak için çeşitli tampon boyutu, benzetim süresi ve yaşam süresi değerlerinde çalıştırılan üç benzetim programında da, çıkış portunun band genişliği 1 hücre/s, kabul edilebilir hücre kayıp olasılığı ise 10^{-4} olarak alınmıştır.

Tablo 6.1'de aynı anda iletme geçen kaynakların sonsuz süreli iletimde kalması yani yaşam süresinin sonsuz olması durumu için elde edilen değerler verilmiştir. Bu durum için benzetim süresi, ON-OFF kaynak modeline göre modellenen kaynağın ON periyodunun süresine eşit seçilmiştir. Bunun nedeni ise, bu süre sonunda tüm kaynakların hücre göndermesinin sona ermesi ve bir sonraki periyoda kadar hiçbir kaynağın hücre göndermeyecek olmasıdır (Şekil 6.2). Tıkanıklığa yol açmayan ve istenen servis kalitesini sağlayan kabul edilebilecek maksimum bağlantı sayısının tampon boyutuna bağlı olarak değişimi bu tabloda gözlenmektedir. Maksimum bağlantı sayısı için elde edilen değerlerin doğruluğunu sınamak amacıyla, periyod sonunda tamponda kalan hücre sayısı hesaplanmıştır. Periyod sonunda tamponda hücre bulunmaması, hesaplanan maksimum bağlantı sayısı kadar bağlantının kabul edilmesi durumunda, ileriki periyodlarda sistemde tıkanıklık olabileceği düşüncesini ortadan kaldırır.

Tablo 6.2' de ise, aynı anda iletme geçen kaynakların belirli bir süre iletimde kalması, yani yaşam süresinin belirli olması durumu için elde edilen değerler verilmiştir. Burada yaşam süresi 3, 6, 12, 25, 50 ve 100 periyod süreleri seçilmiştir. Benzetim süresi ise, kullanıcı tarafından girilen kaynağın ortalama ON periyodu süresine (Ton) eşit alınmıştır. Tablo 6.2' den görüldüğü gibi, yaşam süresinin belirli seçilmesi durumunda bulunan maksimum bağlantı sayısı değerleri, yaşam süresinin

sonsuz seçilmesi durumunda bulunanlar ile aynıdır ve yaşam süresi değerlerine bağlı değildir. Bunun nedeni ise, yaşam süresi değerinin sadece B değeri hesaplanırken kullanılıyor olmasıdır. Burada, yaşam süresinin sonsuz olduğu durumdan farklı olarak, periyod sonunda tamponda bulunan hücre sayısının yaşam süresine bağlı B değeri ile karşılaştırılması düşünülmüştür. B değeri (6.1)'deki formüle göre hesaplanır. Bir periyodluk süre sonunda tamponda bulunan hücre sayısı B değerinden küçük ise, bulunan sayıda bağlantının kabul edilmesi durumunda ileriki periyodlarda sistemde tıkanıklık olabileceği kuşkusu ortadan kalkar.

Tablo 6.1 Kaynakların Aynı Anda İletime Başladığını Varsayan Birinci Benzetim Programının A tipi Kaynak ve Sonsuz Yaşam Süresi Seçeneği İçin Çalıştırılması Sonucu Elde Edilen Değerler

Tsim (s) BENZETİM SÜRESİ	X (Hücre) TAMPON BOYUTU	Kaynakların Hepsinin Aynı Anda İletime Başlaması Durumunda Kabul Edilebilecek Maksimum Bağlantı Sayısı (Yaşam süresi sonsuz)	PERİYOD SONUNDA TAMPONDAKİ HÜCRE SAYISI
9600 s. (Ton süresi)	100	48	0
	350	49	0
	1000	53	0
	3500	65	0
	5000	73	0
	7500	85	0

Tablo 6.2 Kaynakların Aynı Anda İletime Başladığını Varsayan Birinci Benzetim Programının A Tipi Kaynak ve Belirli Yaşam Süresi Seçeneği İçin Çalıştırılması Sonucu Elde Edilen Değerler

Tsim (s)	Tyasam (s)	x (Hücre)	Kaynakların Hepsinin Aynı Anda İletime Başlaması Durumunda Kabul Edilebilecek Maksimum Bağlantı Sayısı	PERİYOD SONUNDA TAMPONDAKİ HÜCRE SAYISI	B SAYISI
BENZETİM SÜRESİ	YAŞAM SÜRESİ	TAMPON BOYUTU			$B = x / A$ $A = \frac{tyaşam}{(ton + toff)}$
9600 s. (Ton süresi)	3*(ton+toff) =192 000 s.	100	48	0	33.333
		350	49	0	116.667
		1000	53	0	333.333
		3500	65	0	1166.667
		5000	73	0	1666.667
		7500	85	0	2500
9600 s. (Ton süresi)	6*(ton+toff) =384 000 s.	100	48	0	16.667
		350	49	0	58.333
		1000	53	0	166.667
		3500	65	0	583.333
		5000	73	0	833.333
		7500	85	0	1250
9600 s. (Ton süresi)	12*(ton+toff) =768 000 s.	100	48	0	8.333
		350	49	0	29.167
		1000	53	0	83.333
		3500	65	0	291.667
		5000	73	0	416.667
		7500	85	0	625
9600 s. (Ton süresi)	25*(ton+toff) =1 600 000 s.	100	48	0	4
		350	49	0	14
		1000	53	0	40
		3500	65	0	140
		5000	73	0	200
		7500	85	0	300
9600 s. (Ton süresi)	50*(ton+toff) =3 200 000 s.	100	48	0	2
		350	49	0	7
		1000	53	0	20
		3500	65	0	70
		5000	73	0	100
		7500	85	0	150
9600 s. (Ton süresi)	100*(ton+toff) =6 400 000 s.	100	48	0	1
		350	49	0	3.5
		1000	53	0	10
		3500	65	0	35
		5000	73	0	50
		7500	85	0	75

Kaynakların belirli bir aralıkta rastgele iletme başladığı varsayımı altında yazılan ikinci benzetim programının A tipi kaynak için çalıştırılması sonucu elde edilen değerler, Tablo 6.3 ve Tablo 6.4’ de verilmiştir.

Tablo 6.3 İkinci Benzetim Programının A Tipi Kaynak ve Kaynakların Bir Periyod Süresi İçerisinde Rastgele İletme Başlaması Durumu İçin Çalıştırılması Sonucu Elde Edilen Değerler (Yaşam Süresi Sonsuz Varsayılıyor)

x (Hücre) TAMPON BOYUTU	Tsim (s) BENZETİM SÜRESİ	Kaynakların Bir Periyod İçerisinde Rastgele İletme Başlaması Durumunda Kabul Edilebilecek Maksimum Bağlantı Sayısı
100	6 * (ton + toff) = 384 000 s.	320
1 000		321
10 000		328
100 000		404
350 000		612
500 000		737
750 000		946
100	12 * (ton + toff) = 768 000 s.	320
1 000		320
10 000		324
100 000		362
350 000		466
500 000		529
750 000		633
100	24 * (ton + toff) = 1 536 000 s.	320
1 000		320
10 000		322
100 000		341
350 000		393
500 000		424
750 000		477
100	50 * (ton + toff) = 3 200 000 s.	320
1 000		320
10 000		321
100 000		330
350 000		355
500 000		370
750 000		395
100	100 * (ton + toff) = 6 400 000 s.	320
1 000		320
10 000		321
100 000		325
350 000		338
500 000		345
750 000		358

100	200 * (ton + toff) = 12 800 000 s.	320
1 000		320
10 000		320
100 000		323
350 000		329
500 000		333
750 000		339

Tablo 6.3' de, Şekil 6.3' deki modele göre kaynakların bir periyodluk süre içerisinde rastgele ilettime başladığı durum için kabul edilebilecek maksimum bağlantı sayısının, çeşitli benzetim süresi ve tampon boyutu değerlerine göre değişimi verilmiştir. Bu durumda kaynakların iletimde kalma süresi olan yaşam süresi sonsuz kabul edilmiştir. Benzetim süresi ise 6, 12, 24, 50, 100 ve 200 periyod süreleri seçilmiştir.

Tablo 6.4' de, Şekil 6.4' deki modele göre kaynakların yaşam süresi içerisinde rastgele ilettime başladığı durum için kabul edilebilecek maksimum bağlantı sayısının, çeşitli yaşam süresi ve tampon boyutu değerlerine göre değişimi verilmiştir. Bu tabloda Tablo 6.2' deki yaşam süresi değerleri kullanılmıştır. Benzetim süresi ise programın yapısı gereği yaşam süresinin iki katına eşittir ve bu tablodaki benzetim süresi değerleri Tablo 6.3' deki değerlerin aynısıdır. Bu da, her iki durumun karşılaştırılmasında kolaylık sağlar.

Kaynakların belirli bir aralıkta rastgele ilettime başladığı ve ON-OFF sürelerinin değişken olduğu varsayımı altında yazılan üçüncü benzetim programının A tipi kaynak için çalıştırılması sonucu elde edilen değerler, Tablo 6.5 ve Tablo 6.6' da verilmiştir.

Tablo 6.5' de, Şekil 6.5' deki modele göre kaynakların bir periyodluk süre içerisinde rastgele ilettime başladığı durum için kabul edilebilecek maksimum bağlantı sayısının, çeşitli benzetim süresi ve tampon boyutu değerlerine göre değişimi verilmiştir. Şekil 6.5' deki modelin ikinci benzetim programında gözönüne alınan Şekil 6.3' deki modelden farkı; bu modelde her bir kaynağın ON ve OFF periyodlarının süresinin sürekli olarak değişmesidir. Kaynakların iletimde kalma süresi olan yaşam süresi ise, ikinci benzetim programında olduğu gibi sonsuz kabul edilmiştir. Benzetim süresi ise 6, 12, 24, 50, 100 ve 200 periyod süreleri seçilmiştir.

Tablo 6.4 İkinci Benzetim Programının A Tipi Kaynak ve Kaynakların Belirli Bir Yaşam Süresi İçerisinde Rastgele İletime Başlaması Durumu İçin Çalıştırılması Sonucu Elde Edilen Değerler

x (Hücre) TAMPON BOYUTU	tyaşam (s) YAŞAM SÜRESİ	tsim (s) BENZETİM SÜRESİ (Yaşam Süresinin 2 Katı) tsim = 2 * tyaşam	Kaynakların Yaşam Süresi İçerisinde Rastgele İletime Başlaması Durumunda Kabul Edilebilecek Maksimum Bağlantı Sayısı
100	3*(ton+toff) = 192 000 s.	6 * (ton + toff) = 384 000 s.	641
1 000			642
10 000			657
100 000			808
350 000			1225
500 000			1475
750 000			1893
100	6 * (ton + toff) = 384 000 s.	12 * (ton + toff) = 768 000 s.	641
1 000			641
10 000			649
100 000			724
350 000			933
500 000			1058
750 000			1267
100	12 * (ton + toff) = 768 000 s.	24 * (ton + toff) = 1 536 000 s.	641
1 000			641
10 000			645
100 000			682
350 000			787
500 000			849
750 000			954
100	25 * (ton + toff) = 1 600 000 s.	50 * (ton + toff) = 3 200 000 s.	641
1 000			641
10 000			643
100 000			661
350 000			711
500 000			741
750 000			791
100	50 * (ton + toff) = 3 200 000 s.	100 * (ton + toff) = 6 400 000 s.	641
1 000			641
10 000			642
100 000			651
350 000			676
500 000			691
750 000			716
100	100 * (ton + toff) = 6 400 000 s.	200 * (ton + toff) = 12 800 000 s.	641
1 000			641
10 000			641
100 000			646
350 000			658
500 000			666
750 000			678

Tablo 6.5 Üçüncü Benzetim Programının A Tipi Kaynak ve Sonsuz Yaşam Süresi Seçeneği İçin Çalıştırılması Sonucu Elde Edilen Değerler.

x (Hücre) TAMPON BOYUTU	Tsim (s) BENZETİM SÜRESİ	Kaynakların Bir Periyod İçerisinde Rastgele İletime Başlaması ve ON/OFF Sürelerinin Değişken Olması Durumunda Kabul Edilebilecek Maksimum Bağlantı Sayısı
100	6 * (ton + toff) = 384 000 s.	105
1 000		105
10 000		110
100 000		136
350 000		206
500 000		247
750 000		321
100	12 * (ton + toff) = 768 000 s.	105
1 000		106
10 000		106
100 000		118
350 000		150
500 000		172
750 000		207
100	24 * (ton + toff) = 1 536 000 s.	102
1 000		102
10 000		103
100 000		110
350 000		126
500 000		135
750 000		153
100	50 * (ton + toff) = 3 200 000 s.	102
1 000		102
10 000		103
100 000		105
350 000		113
500 000		119
750 000		127
100	100 * (ton + toff) = 6 400 000 s.	102
1 000		102
10 000		102
100 000		103
350 000		107
500 000		109
750 000		109
100	200 * (ton + toff) = 12 800 000 s.	101
1 000		101
10 000		101
100 000		102
350 000		104
500 000		106
750 000		107

Tablo 6.6’ da, Şekil 6.6’ daki modele göre kaynakların yaşam süresi içerisinde rastgele iletme başladığı durum için, kabul edilebilecek maksimum bağlantı sayısının çeşitli yaşam süresi ve tampon boyutu değerlerine göre değişimi verilmiştir. Şekil 6.6’ daki modelin, ikinci benzetim programında gözönüne alınan Şekil 6.4’ deki modelden tek farkı; bu modelde herbir kaynağın ON ve OFF periyodlarının süresinin değişken olmasıdır. Bu tabloda Tablo 6.2 ve Tablo 6.4’ deki yaşam süresi değerleri kullanılmıştır. Benzetim süresi ise yaşam süresinin iki katına eşittir.

Tablo 6.6 Üçüncü Benzetim Programının A Tipi Kaynak ve Kaynakların Belirli Bir Yaşam Süresi İçerisinde Rastgele İletime Başlaması Durumu İçin Çalıştırılması Sonucu Elde Edilen Değerler (Kaynakların ON ve OFF Periyodlarının Süreleri Değişken.)

x (Hücre)	tyaşam (s)	tsim (s)	Kaynakların Yaşam Süresi İçerisinde Rastgele İletime Başlaması ve ON/OFF Sürelerinin Değişken Olması Durumunda Kabul Edilebilecek Maksimum Bağlantı Sayısı
TAMPON BOYUTU	YAŞAM SÜRESİ	BENZETİM SÜRESİ (Yaşam Süresinin 2 Katı) $tsim = 2 * tyaşam$	
100	3*(ton+toff) = 192 000 s.	6 * (ton + toff) = 384 000 s.	184
1 000			187
10 000			190
100 000			234
350 000			355
500 000			427
750 000			551
100	6 * (ton + toff) = 384 000 s.	12 * (ton + toff) = 768 000 s.	190
1 000			190
10 000			195
100 000			219
350 000			281
500 000			320
750 000			383
100	12 * (ton + toff) = 768 000 s.	24 * (ton + toff) = 1 536 000 s.	198
1 000			198
10 000			200
100 000			210
350 000			245
500 000			264
750 000			292
100	25 * (ton + toff) = 1 600 000 s.	50 * (ton + toff) = 3 200 000 s.	201
1 000			201
10 000			202
100 000			207
350 000			224
500 000			231
750 000			248

100	50 * (ton + toff) = 3 200 000 s.	100 * (ton + toff) = 6 400 000 s.	203
1 000			203
10 000			203
100 000			205
350 000			212
500 000			218
750 000			226
100	100 * (ton + toff) = 6 400 000 s.	200 * (ton + toff) = 12 800 000 s.	202
1 000			202
10 000			202
100 000			204
350 000			208
500 000			210
750 000			215

Tablo 6.7’ de, A tipi kaynaklar için Bölüm 5.5.1’ de kaynak kodu verilen programın dört kriter için ayrı ayrı çalıştırılması sonucu ve yazılan benzetim programlarının belirli benzetim ve yaşam sürelerinde çalıştırılması sonucu bulunan kabul edilebilecek maksimum bağlantı sayısı değerleri verilmiştir. Bu tablodaki T değeri periyod süresini gösterir, yani kaynağın ON ve OFF periyodlarının ortalama sürelerinin toplamına eşittir.

Tablo 6.7 A Tipi Kaynaklar İçin Eşdeğer Band Genişliği Temelli Dört Kriter ve Benzetim Programları İle Bulunan Maksimum Bağlantı Sayısı Değerleri (con = 1, P (loss) = 10⁻⁴)

MPON YUTU (Hücre)	KABUL EDİLEBİLECEK MAKSİMUM BAĞLANTI SAYISI								
	GAN Kriteri	Elwalid Mitra Kriteri	Geliştirilmiş Elwalid Mitra Kriteri	Sohraby Kriteri	1.Benzetim Programı Yaşam süresi Sonsuz Simulasyon Süresi=ton (ton : ON periyod süresi)	2.Benzetim Programı Yaşam süresi Sonsuz Simulasyon Süresi=200*T (T : periyod süresi)	2.Benzetim Programı Yaşam süresi = 100* T Simulasyon Süresi=200*T (T : periyod süresi)	3.Benzetim Programı Yaşam süresi Sonsuz Simulasyon Süresi=200*T (T : periyod süresi)	3.Benzetim Programı Yaşam süresi = 100* T Simulasyon Süresi=200*T (T : periyod süresi)
0	187	50	219	24	48	320	641	101	202
100	187	89	235	142	53	320	641	101	202
1000	279	279	288	283	98	320	641	101	202
10000	316	316	318	316	548	323	646	102	204
100000	319	319	319	319	1801	329	658	104	208
1000000	319	319	319	319	2552	333	666	106	210
10000000	319	319	319	319	3804	339	678	107	215

Tablo 6.8’ de, ikinci benzetim programının A tipi kaynak için çalıştırılması sonucu elde edilen kabul edilebilecek maksimum bağlantı sayısının, çeşitli yaşam süresi ve tampon boyutu değerlerine göre değişimi verilmiştir. Benzetim süreleri, yaşam süresinin 4, 8, 10, 16 ve 100 katına eşit seçilmiştir. Burada tyaşam yaşam süresini, tsim benzetim süresini göstermek üzere, kaynakların (0, tsim-tyaşam) aralığında rastgele ilettime başladığı varsayılarak değerler elde edilmiştir.

Tablo 6.8 İkinci Benzetim Programının A Tipi Kaynak İçin Çalıştırılması Sonucu Elde Edilen Değerler. (Kaynaklar (0, tsim-tyaşam) aralığında rastgele ilettime başlıyor).

x (Hücre)	Tyaşam (s)	tsim (s)	Kaynakların Yaşam Süresi İçerisinde Rastgele İlettime Başlaması Durumunda Kabul Edilebilecek Maksimum Bağlantı Sayısı
TAMPON BOYUTU	YAŞAM SÜRESİ	BENZETİM SÜRESİ	
100			1282
1 000	3*(ton+toff)	12 * (ton + toff)	1283
10 000			1298
100 000	= 192 000 s.	= 768 000 s.	1449
350 000			1866
100			2564
1 000	3*(ton+toff)	24 * (ton + toff)	2566
10 000			2581
100 000	= 192 000 s.	= 1 536 000 s.	2731
350 000			3148
100			5128
1 000	3*(ton+toff)	48 * (ton + toff)	5130
10 000			5145
100 000	= 192 000 s.	= 3 072 000 s.	5295
350 000			5713
100			32 054
1 000	3*(ton+toff)	300 * (ton + toff)	32 056
10 000			32 071
100 000	= 192 000 s.	= 19 200 000 s.	32 221
350 000			32 638
100			1282
1 000	100*(ton+toff)	400 * (ton + toff)	1282
10 000			1282
100 000	= 6 400 000 s.	= 25 600 000 s.	1287
350 000			1299
100			2564
1 000	100*(ton+toff)	800 * (ton + toff)	2564
10 000			2564
100 000	= 6 400 000 s.	= 51 200 000 s.	2569
350 000			2581

100	100*(ton+toff) = 6 400 000 s.	1 000 * (ton + toff) = 64 000 000 s.	3205
1 000			3205
10 000			3205
100 000			3210
350 000			3222

Tablo 6.9' da, üçüncü benzetim programının A tipi kaynak için çalıştırılması sonucu elde edilen kabul edilebilecek maksimum bağlantı sayısının, çeşitli yaşam süresi ve tampon boyutu değerlerine göre değişimi verilmiştir. Benzetim süreleri, yaşam süresinin 4, 8, 16 ve 30 katına eşit seçilmiştir. Burada da tyaşam yaşam süresini, tsim benzetim süresini göstermek üzere, kaynakların (0, tsim-tyaşam) aralığında rastgele iletme başladığı varsayılarak değerler elde edilmiştir.

Tablo 6.9 Üçüncü Benzetim Programının A Tipi Kaynak İçin Çalıştırılması Sonucu Elde Edilen Değerler. (Kaynaklar (0, tsim-tyaşam) aralığında rastgele iletme başlıyor).

x (Hücre)	tyaşam (s)	tsim (s)	Kaynakların Yaşam Süresi İçerisinde Rastgele İletime Başlaması Durumunda Kabul Edilebilecek Maksimum Bağlantı Sayısı
TAMPON BOYUTU	YAŞAM SÜRESİ	BENZETİM SÜRESİ	
100	3*(ton+toff) = 192 000 s.	12 * (ton + toff) = 768 000 s.	369
1 000			369
10 000			372
100 000			420
350 000			537
100	3*(ton+toff) = 192 000 s.	24 * (ton + toff) = 1 536 000 s.	738
1 000			738
10 000			747
100 000			789
350 000			911
100	3*(ton+toff) = 192 000 s.	48 * (ton + toff) = 3 072 000 s.	1475
1 000			1475
10 000			1475
100 000			1528
350 000			1650
100	3*(ton+toff) = 192 000 s.	90 * (ton + toff) = 19 200 000 s.	2786
1 000			2786
10 000			2736
100 000			2832
350 000			2954
100	100*(ton+toff) = 6 400 000 s.	400 * (ton + toff) = 25 600 000 s.	406
1 000			406
10 000			406
100 000			406
350 000			411

7. EŞDEĞER BAND GENİŞLİĞİ TEMELLİ KABUL KRİTERLERİNİN VE BENZETİM PROGRAMLARININ KARMAŞIKLIK ANALİZİ

Bu bölümde, Bölüm 5.4’ de anlatılan eşdeğer band genişliği temelli kabul kriterlerinin ve Bölüm 6’ da açıklanan üç benzetim programının karmaşıklık analizi yapılmıştır.

Karmaşıklık hesaplamalarında 80486 işlemcisi temel alınarak; okuma ve yazma işlemlerinin karmaşıklığı 1, toplama, çıkarma, bölme ve VEYA (OR) işlemlerinin karmaşıklığı 8, çarpma işleminin karmaşıklığı 11, karşılaştırma işleminin karmaşıklığı 15, yuvarlama işleminin karmaşıklığı 21, karekök alma işleminin karmaşıklığı 83, logaritma alma ve üs alma işlemlerinin karmaşıklığı ise 196 alınmıştır [25].

7.1 Eşdeğer Band Genişliği Temelli Kabul Kriterlerinin Karmaşıklık Analizi

Bu bölümde eşdeğer band genişliği temelli kabul kriterlerinden Bölüm 5.4’ de ayrıntılı olarak anlatılan Guerin-Ahmadi-Naghshineh, Elwalid-Mitra, Geliştirilmiş Elwalid Mitra ve Sohraby kriterlerinin karmaşıklık analizleri yapılmıştır.

7.1.1 Guerin-Ahmadi-Naghshineh (GAN) Kriterinin Karmaşıklık Analizi

Çağrı kabul denetimi algoritmalarından biri olan bu kriterde, bir link üzerinde çoğullanan bir grup bağlantının eşdeğer kapasitesini veren tek bir ifade elde edilmiştir (7.1). Eşdeğer kapasite C , istenen servis kalitesini (QoS) sağlamak için gerekli band genişliği olarak tanımlanır [10].

$$C = \min \left\{ m + \alpha \sigma, \sum_{i=1}^N c_i \right\} \quad (7.1)$$

Burada m ve σ^2 , link üzerine yönlendirilen bütün bağlantıların toplam bit hızının ortalaması ve varyansdır. c_i , her bir bağlantının eşdeğer kapasitesidir. N , çoğullanan bağlantı sayısıdır. α değeri ise

$$\alpha = \sqrt{-2 \ln(\epsilon) - \ln(2\pi)} \quad (7.2)$$

olarak tanımlanmıştır. ϵ , sistemde sağlanması istenen hücre kayıp olasılığına karşı düşer.

Bu kriterde ayrıca (7.1)' deki ifadeye dayalı olarak, ağ linkleri üzerinde o anki kapasite tahsisini karakterize eden link metrikleri (L_j) tanımlanmıştır (7.3) [10]. Üç boyutlu bir vektör olan bu metrik, aşağıdaki biçimdedir :

$$L_j = (m = \sum_{i=1}^N m_i, \quad \sigma^2 = \sum_{i=1}^N \sigma_i^2, \quad C = \sum_{i=1}^N c_i) \quad (7.3)$$

L_j , j linkinin link metrik vektörüdür. m ve σ^2 ise sırasıyla j linki üzerine yönlendirilen bütün bağlantılara karşı düşen toplam bit hızının ortalama ve varyansdır. C ise, j linki üzerine yönlendirilen bağlantıların eşdeğer kapasitelerinin toplamıdır.

Yeni bir bağlantı isteğinin kabul edilip edilmeyeceğine karar verirken, j linki üzerindeki band genişliği tahsisini yani j linkinin eşdeğer kapasitesini elde etmek için, (7.3)' de verilen L_j link metrik vektöründe saklanan bilgiden yararlanılırsa gereken hesaplama işlemleri büyük ölçüde azalır. Aşağıda GAN kriterinin karmaşıklığı da, L_j link metrik vektörü göz önüne alınarak hesaplanmıştır.

Link metrik vektöründe saklanan bilgiden yararlanarak j linkinin eşdeğer band genişliğini hesaplamak için gerekli işlemler (7.1)' den görüleceği gibi; 1 karekök alma, 1 çarpma, 1 toplama ve 1 karşılaştırma işlemidir (σ^2 ' nin karekökü alınıyor, $(m + \alpha \sigma)$ ifadesi ile $(\sum_{i=1}^N c_i)$ karşılaştırılarak C hesaplanıyor).

Link metrik vektörünün güncellenmesi diğer önemli bir konudur. Güncellemeler sadece, link metrik vektöründeki bilgiye ve kurulacak veya bırakılacak bağlantıların

karakteristiklerine dayalı olarak yapılır. O an link üzerinde mevcut bağlantıların her birinin karakteristikleri için özel bir bilgiye ihtiyaç yoktur.

Link metrik vektörü $L_j = (m, \sigma^2, C)$ ' nin güncellenmesi şu şekildedir [10]:

1. Kaynak düğüm, bağlantı isteğinde bulunan kaynağın (R_i, r_i, b_i) şeklindeki bağlantı vektöründen yararlanarak $q_i = \pm (m_i, \sigma_i^2, c_i)$ şeklindeki istek vektörünü hesaplar. (+ işareti bağlantı kurma isteğini, - işareti ise bağlantı bırakma isteğini gösterir).

$$(R_i, r_i, b_i) \rightarrow q_i = \pm (m_i, \sigma_i^2, c_i)$$

Bu gösterimde

R_i : Bağlantı isteğinde bulunan i. kaynağın tepe hızını,

r_i : i. kaynağın aktif olduğu zaman bölümünü,

b_i : i. kaynak için aktif periyodun ortalama süresini,

m_i : i. kaynağın ortalama bit hızını,

σ_i^2 : i. kaynağın bit hızının varyansını,

c_i : i. kaynağın eşdeğer band genişliğini (kapasitesini)

temsil eder.

2. İstek vektörü q_i , bağlantının yönlendirileceği bütün düğümlere ve linklere gönderilir. İlgili link metrik vektörleri de, basit bir vektör ekleme veya çıkarma işlemi ile güncellenir.

GAN kriterine göre yeni bir çağrının kabul edilip edilmeyeceğine karar vermek için, yukarıda anlatılan link metrik vektöründe saklanan bilgiden yararlanıldığı varsayılırsa, GAN kriterinin karmaşıklığı aşağıdaki biçimde 4 adımda hesaplanabilir:

1. Bağlantı vektörü (R_i, r_i, b_i) ' den istek vektörü $q_i = (m_i, \sigma_i^2, c_i)$ hesaplanır.

$$(R_i, r_i, b_i) \rightarrow q_i = (m_i, \sigma_i^2, c_i).$$

Bu hesaplama sadece, yeni bağlantı kurma isteğinde bulunan kaynak için yapılır.

Bunun için aşağıdaki formüllerden yararlanılır [10]:

$$m_i = R_i r_i$$

$$\sigma_i^2 = m_i (R_i - m_i)$$

$$c_i = \frac{(a_i - x) + \sqrt{(a_i - x)^2 + 4 x a_i r_i}}{2 a_i} R_i \quad (7.4)$$

$$a_i = \ln(1 / \varepsilon) b_i (1 - r_i) R_i$$

Burada x, tampon boyutunu gösterir.

m_i 'nin hesaplanmasında :

R_i ve r_i değerleri bağlantı vektöründen okunur. Çarpma işlemi ile m_i hesaplanır ve bu değer istek vektörüne yazılır. Toplam işlem sayısı :

$$2 \text{ okuma} + 1 \text{ çarpma} + 1 \text{ yazma} = 4 \text{ işlem.}$$

Toplam karmaşıklık ise :

$$2 \times 1 + 1 \times 11 + 1 \times 1 = 14 \rightarrow O(1)' \text{ dir.}$$

σ_i^2 'nin hesaplanmasında :

R_i ve m_i değerleri okunur. Çarpma ve çıkarma işlemleri ile σ_i^2 hesaplanır ve bu değer istek vektörüne yazılır. Toplam işlem sayısı :

$$2 \text{ okuma} + 1 \text{ çarpma} + 1 \text{ çıkarma} + 1 \text{ yazma} = 5 \text{ işlem.}$$

Toplam karmaşıklık ise :

$$2 \times 1 + 1 \times 11 + 1 \times 8 + 1 \times 1 = 22 \rightarrow O(1)' \text{ dir.}$$

a_i 'nin hesaplanmasında :

ε , b_i , r_i , ve R_i değerleri okunur. Logaritma, çıkarma ve çarpma işlemleri ile a_i değeri hesaplanır ve bu değer yazılır. Toplam işlem sayısı :

$$4 \text{ okuma} + 1 \text{ logaritma işlemi} + 1 \text{ çıkarma} + 3 \text{ çarpma} + 1 \text{ yazma} = 10 \text{ işlem}$$

Toplam karmaşıklık ise :

$$4 \times 1 + 1 \times 196 + 1 \times 8 + 3 \times 11 + 1 \times 1 = 242 \rightarrow O(1)' \text{ dir.}$$

c_i ' nin hesaplanmasında :

1-) a_i hesaplanır.

İşlem sayısı = 10,

Karmaşıklık = 242

2-) a_i ' den yararlanarak c_i hesaplanır.

R_i , r_i , x ve a_i değerleri okur. (7.4)' deki formülünden yararlanarak c_i hesaplanır ve bu değer istek vektörüne yazılır.

Toplam işlem sayısı (a_i ' nin hesaplanması için gereken işlemler hariç):

$$4 \text{ okuma} + 2 \text{ çıkarma} + 2 \text{ toplama} + 1 \text{ bölme} + 1 \text{ karekök alma} \\ + 1 \text{ kare alma} + 5 \text{ çarpma} + 1 \text{ yazma} = 17 \text{ işlem}$$

c_i ' nin hesaplanmasında toplam $17 + 10 = 27$ işlem gerekir.

Toplam karmaşıklık ise :

$$242 + 4 \times 1 + 2 \times 8 + 2 \times 8 + 1 \times 8 + 1 \times 83 + 1 \times 11 + 5 \times 11 + 1 \times 1 = 436 \rightarrow O(1)' \text{ dir.}$$

Birinci adımda toplam :

$$4 + 5 + 27 = 36 \text{ işlem gerekir.}$$

Birinci adımın karmaşıklığı ise :

$$14 + 22 + 436 = 472 \rightarrow O(1)' \text{ dir.}$$

2. $q_i = (m_i, \sigma_i^2, c_i)$ istek vektörü ile link metrik vektörü $L_j = (m, \sigma^2, C)$ toplanır. (Yeni bağlantı isteği için).

Bu adımda her iki vektörün elemanları okunur (6 okuma işlemi). Vektör toplama işlemi ile yeni link metrik vektörünün elemanları hesaplanır ve link metrik vektörü L_j güncellenir (1 vektör toplama ve 3 yazma işlemi). Toplam işlem sayısı :

$$6 \text{ okuma} + 1 \text{ vektör toplama işlemi} + 3 \text{ yazma} = 10 \text{ işlem}$$

Toplam karmaşıklık ise :

$$6 \times 1 + 8 \times 3 + 3 \times 1 = 33 \rightarrow O(1)' \text{ dir.}$$

3. $C = \min \left\{ m + \alpha \sigma, \sum_{i=1}^N c_i \right\}$ formülünden yararlanarak C hesaplanır. C' nin hesaplanması için :

α değeri (7.2) formülünden hesaplanır. α 'nın hesaplanmasında toplam işlem sayısı:

$$1 \text{ okuma} + 1 \text{ karekök alma} + 1 \text{ çıkarma} + 1 \text{ logaritma işlemi} + 1 \text{ çarpma} = 5 \text{ işlem}$$

α 'nın hesaplanmasının karmaşıklığı ise :

$$1 \times 1 + 1 \times 83 + 1 \times 8 + 1 \times 196 + 1 \times 11 = 299 \rightarrow O(1)' \text{ dir.}$$

α hesaplandıktan sonra, güncellenmiş link metrik vektöründen m , σ^2 , ve $\sum_{i=1}^N c_i$ değerleri okunur. σ^2 'nin kare kökü hesaplanır. Son olarak; hesaplanan $(m + \alpha \sigma)$ ifadesi ile $(\sum_{i=1}^N c_i)$ ifadesi karşılaştırılarak C değeri elde edilir. C' nin hesaplanmasında toplam işlem sayısı :

$$3 \text{ okuma} + \alpha' \text{ 'nin hesaplanması için } 5 \text{ işlem} + 1 \text{ karekök alma} + 1 \text{ çarpma} \\ + 1 \text{ toplama} + 1 \text{ karşılaştırma} = 12 \text{ işlem}$$

Toplam karmaşıklık ise :

$$3 \times 1 + 299 + 1 \times 83 + 1 \times 11 + 1 \times 8 + 1 \times 15 = 419 \rightarrow O(1)' \text{ dir.}$$

4. Çağrının kabul edilip edilmeyeceğine karar vermek için çıkış portunun hızı W ile, 3. adımda hesaplanan N tane kaynağın eşdeğer band genişliği C karşılaştırılır.

$$W \geq C \text{ ise çağrı kabul edilir.}$$

Bu adımda W değeri okunur ve hesaplanan C değeri ile karşılaştırılır.

$$\text{Toplam işlem sayısı : } 1 \text{ okuma} + 1 \text{ karşılaştırma} = 2 \text{ işlem}$$

Karmaşıklık ise:

$$1 \times 1 + 1 \times 15 = 16 \rightarrow O(1)' \text{ dir.}$$

GAN Kriteri ile çağrının kabul edilip edilmeyeceğine karar vermek için 4 adım sonunda toplam :

$$36 + 10 + 12 + 2 = 60 \text{ işlem gerekir.}$$

Toplam karmaşıklık ise:

$$472 + 33 + 419 + 16 = 940 \rightarrow O(1)' \text{ dir.}$$

7.1.2 Elwalid - Mitra Kriterinin Karmaşıklık Analizi

Elwalid ve Mitra, MMFP kaynağın eşdeğer band genişliğinin yaklaşık olarak; kaynağın parametreleri, çoğullayıcı kaynakları ve istenen hücre kayıp olasılığından türetilen A matrisinin maksimum reel özdeğerine eşit olduğunu göstermişlerdir.

Bir trafik kaynağının L tane durum ile modellendiği, Q matrisinin varış sürecinin durumları arasındaki geçişi yöneten Markov zincirinin üreticisi olduğu, ve Markov zincirinin bütün durumları için varış hızlarının $\vec{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_L)$ vektörü ile gösterildiği varsayılırsa tek bir kaynağın eşdeğer band genişliğinin

$$A = \Lambda - \frac{1}{\xi} Q \text{ matrisinin maksimum reel özdeğeri olduğu gösterilmiştir [11].}$$

Burada $\Lambda = \text{diag}(\vec{\lambda})$, ve $\xi = \ln(\epsilon)/x$ olarak tanımlanmıştır. x tampon boyutunu, ϵ ise sistemde sağlanması istenen hücre kayıp olasılığını gösterir. Q matrisi ise :

$$Q = \begin{bmatrix} -\alpha & \alpha \\ \beta & -\beta \end{bmatrix}$$

şeklinde tanımlanmıştır. α ve β değerleri, kaynağın ON periyodunun ortalama süresi T_{on} , ve kaynağın OFF periyodunun ortalama süresi T_{off} cinsinden ifade edilebilir :

$$\alpha = 1 / T_{off}, \quad \beta = 1 / T_{on}$$

Ayrıca bu yöntemle göre N tane kaynağın eşdeğer band genişliği, bütün kaynakların eşdeğer band genişliklerinin toplamına eşittir [11].

$$c = \sum_{i=1}^N c_i$$

Burada c_i , i. kaynağın eşdeğer band genişliğidir ve bu değer, kaynak sistemdeki tek kaynakmış gibi hesaplama yapılarak bulunur.

W çıkış portunun band genişliği olmak üzere, bir grup trafik kaynağı ancak aşağıdaki koşul sağlandığı takdirde kabul edilebilir :

$$W \geq \sum_{i=1}^N c_i$$

Böylece, kaynakların eşdeğer band genişliklerinin toplamı C, çıkış portunun band genişliği W ile karşılaştırılarak kabul edilebilecek maksimum bağlantı sayısı, yani kabul bölgesi Elwalid-Mitra kriterine göre bulunmuş olur.

Elwalid ve Mitra, $(Q, \bar{\lambda})$ ile tanımlanan 2 durumlu tek bir MMFP kaynağın eşdeğer band genişliği $c(\xi)$ 'nin hesaplanması için aşağıdaki formülü önermişlerdir [11]:

$$c(\xi) = \frac{0.5}{\xi} \left[[(\lambda_1 + \lambda_2)\xi + \alpha + \beta] - \sqrt{\{(\lambda_1 + \lambda_2)\xi + \alpha + \beta\}^2 - 4(\lambda_1\lambda_2\xi^2 + \beta\lambda_1\xi + \alpha\lambda_2\xi)} \right] \quad (7.5)$$

Elwalid-Mitra kriterinin karmaşıklığı da bu formül gözönüne alınarak, kaynakların 2 durumlu MMFP kaynaklar olduğu varsayımı altında yapılmıştır.

Bu durumda Elwalid-Mitra kriterinin karmaşıklığı üç adımda hesaplanabilir.

1. 2 durumlu tek bir MMFP kaynağının eşdeğer band genişliği $c_i(\xi)$, (7.5) formülüne göre hesaplanır.

$$(\lambda_1, \lambda_2, \alpha, \beta, \xi) \rightarrow c_i$$

ξ 'nin hesaplanmasında :

$\xi = \ln(\varepsilon)/x$ formülüne göre hesaplanır.

ε ve x değerleri okunur. Logaritma ve bölme işlemleri ile ξ hesaplanır ve bu değer yazılır.

Toplam işlem sayısı : 2 okuma + 1 logaritma + 1 bölme + 1 yazma = 5 işlem

Toplam karmaşıklık ise : $2 \times 1 + 1 \times 196 + 1 \times 8 + 1 \times 1 = 207 \rightarrow O(1)$ ' dir.

ξ 'nin değeri kullanılarak (7.5) formülünden c_i hesaplanır :

$\lambda_1, \lambda_2, \alpha, \beta$ ve ξ değerleri okunur. (7.5) formülünden c_i hesaplanır ve yazılır.

İşlem sayısı : 5 okuma + 1 karekök alma + 2 kare alma + 2 çıkarma

$$+ 10 \text{ çarpma} + 8 \text{ toplama} + 1 \text{ bölme} + 1 \text{ yazma} = 30 \text{ işlem}$$

Karmaşıklık ise :

$$5 \times 1 + 1 \times 83 + 2 \times 11 + 2 \times 8 + 10 \times 11 + 8 \times 8 + 1 \times 8 + 1 \times 1 = 309 \rightarrow O(1)' \text{ dir.}$$

Birinci adımda toplam işlem sayısı :

$$5 \text{ işlem} + 30 \text{ işlem} = 35 \text{ işlem}$$

Toplam karmaşıklık ise :

$$207 + 309 = 516 \quad \text{dir.}$$

2. GAN kriterinde olduğu gibi, bu kriterde de link üzerindeki mevcut bağlantı bilgilerinin bir link metrik vektöründe saklandığı varsayılmıştır. Elwalid-Mitra kriterinin yapısı gereği, bu link metrik vektöründe linklerin üzerindeki toplam eşdeğer kapasite C' nin saklanması ve yeni bir bağlantı isteğinde bu değer güncellenmesi yeterlidir. Bunun için de 1. adımda hesaplanan c_i ' den oluşan $q_i = (c_i)$ istek vektörü ile tek elemanlı link metrik vektörü $L_j = (C = \sum_{i=1}^N c_i)$ toplanır. Elde edilen yeni C değeri ile link metrik vektörü L_j güncellenir.

Toplam işlem sayısı : 2 okuma + 1 toplama + 1 yazma = 4 işlem

Toplam karmaşıklık ise : $2 \times 1 + 1 \times 8 + 1 \times 1 = 11 \rightarrow O(1)' \text{ dir.}$

3. Çağrının kabul edilip edilmeyeceğine karar vermek için çıkış portunun hızı W ile, güncellenmiş link metrik vektöründen okunan N tane kaynağın eşdeğer band genişliğini gösteren $C = \sum_{i=1}^N c_i$ değeri karşılaştırılır.

$W \geq C$ ise çağrı kabul edilir.

Bu adımdaki işlem sayısı : 2 okuma + 1 karşılaştırma = 3 işlem

Karmaşıklık ise:

$$2 \times 1 + 1 \times 15 = 17 \rightarrow O(1)' \text{ dir.}$$

Elwalid-Mitra Kriteri ile çağrının kabul edilip edilmeyeceğine karar vermek için 3 adım sonunda toplam :

$$35 + 4 + 3 = 42 \text{ işlem gerekir.}$$

Toplam karmaşıklık ise:

$$516 + 11 + 17 = 544 \rightarrow O(1)' \text{ dir.}$$

7.1.3 Geliştirilmiş Elwalid-Mitra Kriterinin Karmaşıklık Analizi

Elwalid-Mitra kriteri asimtotik olmayan koşullarda yetersiz kaldığı için, bu kriteri Rege tarafından iyileştirme getirilmiştir. Bu yöntemde bir kaynağın eşdeğer band genişliği, yapılan daha sonraki hesaplamalarda kaynağın tepe hızı yerine kullanılır. Kabul edilebilir kaynak kümesini belirlemek amacıyla da, her bir band genişliğinin lineer olmayan bir fonksiyonu kullanılır.

QoS parametresi hücre kayıp olasılığı ε için, N tane kaynağın kabul edilebilirliğini belirlemek amacıyla şu adımlar izlenir [12]:

1. Her kaynak için eşdeğer band genişliği c_i hesaplanır. Ancak c_i hesaplanırken hücre kayıp olasılığı ε yerine ε' kullanılır. ε' şu şekilde tanımlanmıştır.:

$$\varepsilon' = \left[\frac{W}{(W - m)} \right] \varepsilon$$

2. Bundan sonra her kaynak için tepe hızı yerine, 1. adımda hesaplanan eşdeğer band genişliği kullanılır.
3. N tane kaynağın toplam hızının ortalaması m ve varyansı σ^2 , GAN kriterinde olduğu biçimde hesaplanır. Ancak bu formüllerde kaynağın tepe hızı yerine eşdeğer band genişliği kullanılır. Hesaplanan ortalama hız ve varyans değerleri kullanılarak $P(\text{loss})$ olasılığı aşağıdaki formüle göre hesaplanır.

$$P(\text{loss}) = \frac{\sigma}{m\sqrt{(2\pi)}} \exp \left[\frac{-1}{2} \left(\frac{W - m}{\sigma} \right)^2 \right] \quad (7.6)$$

4. N tane kaynak için ancak aşağıdaki koşul sağlandığı takdirde, bu kaynaklar kabul edilebilir :

$$W \geq \sum_{i=1}^N c_i \quad \text{veya} \quad P(\text{loss}) \leq \varepsilon$$

Bu kriterin karmaşıklığı hesaplanırken, GAN kriterinde olduğu gibi j linkine ait bir L_j vektörünün olduğu ve bu vektörün (7.3)' de tanımlı verilen 3 boyutlu vektör ile

aynı olduğu varsayılmıştır. Bu yöntemin karmaşıklığı, yukarıda anlatılan 4 adım ve L_j vektörünün varlığı gözönüne alınarak aşağıdaki biçimde hesaplanmıştır :

1. Yeni bağlantı kurma isteğinde bulunan kaynak için m_i , σ_i^2 ve c_i hesaplanır.

m_i 'nin hesaplanmasında :

$m_i = R_i r_i$ formülü kullanılır. Bunun için GAN kriterinde gösterildiği gibi 4 işlem gerekir. Bu işlemlerin karmaşıklığı ise 14 ' dür.

c_i 'nin hesaplanmasında :

(7.5)' deki formül kullanılır. Ancak bu formüldeki ξ değeri hesaplanırken hücre kayıp olasılığı ε yerine ε' kullanılır. ε' şu şekilde hesaplanır :

$$\varepsilon' = \left[\frac{W}{(W - m)} \right] \varepsilon$$

ε' hesaplanırken toplam işlem sayısı :

$$3 \text{ okuma} + 1 \text{ çıkarma} + 1 \text{ bölme} + 1 \text{ çarpma} = 6 \text{ işlem}$$

(ε') nün hesaplanmasındaki karmaşıklık :

$$3 \times 1 + 1 \times 8 + 1 \times 8 + 1 \times 11 = 30 \rightarrow O(1)' \text{ dir.}$$

c_i 'nin (7.5)' deki formüle göre hesaplanmasında Elwalid-Mitra kriterinde belirtildiği gibi 35 işlem gerekir. Karmaşıklık ise 516' dir. Yukarıda anlatılan değişikliklerle birlikte c_i hesaplanırken toplam işlem sayısı :

$$6 \text{ işlem} + 34 \text{ işlem} = 40 \text{ işlem}$$

c_i 'nin hesaplanmasındaki karmaşıklık :

$$30 + 515 = 545 \rightarrow O(1)' \text{ dir.}$$

(ξ değeri hesaplanırken ε için yapılan okuma işleminin yerini, ε' değerini hesaplamak için yapılan işlemler alıyor).

σ_i^2 nin hesaplanmasında :

$\sigma_i^2 = m_i (c_i - m_i)$ formülü kullanılır. m_i ve c_i değerleri okunur. Çarpma ve çıkarma işlemleri ile σ_i^2 değeri hesaplanır ve yazılır. Toplam işlem sayısı :

$$2 \text{ okuma} + 1 \text{ çıkarma} + 1 \text{ çarpma} + 1 \text{ yazma} = 5 \text{ işlem}$$

karmaşıklık :

$$2 \times 1 + 1 \times 8 + 1 \times 11 + 1 \times 1 = 22 \rightarrow O(1)' \text{ dir.}$$

Birinci adımda yeni bağlantı için (m_i, σ_i^2, c_i) değerlerinin hesaplanmasında :

$$4 + 40 + 5 = 49 \text{ işlem gerekir.}$$

Birinci adımın karmaşıklığı ise :

$$14 + 545 + 22 = 581 \rightarrow O(1)' \text{ dir.}$$

2. $q_1 = (m_i, \sigma_i^2, c_i)$ istek vektörü ile link metrik vektörü $L_j = (m, \sigma^2, C)$ toplanır. Bu adımda her iki vektörün elemanları okunur (6 okuma işlemi). Vektör toplama işlemi ile yeni link metrik vektörünün elemanları hesaplanır ve link metrik vektörü L_j güncellenir.

Toplam işlem sayısı :

$$6 \text{ okuma} + 1 \text{ vektör toplama işlemi} + 3 \text{ yazma} = 10 \text{ işlem}$$

Toplam karmaşıklık ise :

$$6 \times 1 + 8 \times 3 + 3 \times 1 = 33 \rightarrow O(1)' \text{ dir.}$$

3. $P(\text{loss})$ olasılığı, (7.6) formülüne göre hesaplanır. Bunun için önce link metrik vektöründen m ve σ^2 değerleri okunur. Ayrıca çıkış portunun hızı W okunur. σ^2 'nin kare kökü hesaplanır ve son olarak (7.6) formülünden $P(\text{loss})$ olasılığı hesaplanır. Toplam işlem sayısı :

$$3 \text{ okuma} + 1 \text{ karekök alma} + 1 \text{ çıkarma} + 2 \text{ bölme} + 1 \text{ kare alma} + 3 \text{ çarpma} \\ + 1 \text{ üs alma} = 12 \text{ işlem}$$

Toplam karmaşıklık ise :

$$3 \times 1 + 1 \times 83 + 1 \times 8 + 2 \times 8 + 1 \times 11 + 3 \times 11 + 1 \times 196 = 350 \rightarrow O(1)' \text{ dir.}$$

4. $C \leq W$ veya $P(\text{loss}) \leq \varepsilon$ ise yeni çağrı isteği kabul edilir. Bunun için C , W ve ε değerleri okunur. Üçüncü adımda hesaplanan $P(\text{loss})$ olasılığı da kullanılarak, iki karşılaştırma işlemi ile çağrının kabul edilip edilmeyeceğine karar verilir. Toplam işlem sayısı :

3 okuma + 2 karşılaştırma + 1 veya işlemi = 6 işlem

Karmaşıklık :

$$3 \times 1 + 2 \times 15 + 1 \times 8 = 41 \rightarrow O(1)' \text{ dir}$$

Geliştirilmiş Elwalid-Mitra Kriteri ile çağrının kabul edilip edilmeyeceğine karar vermek için 4 adım sonunda toplam :

$$49 + 10 + 12 + 6 = 77 \text{ işlem gerekir.}$$

Toplam karmaşıklık ise:

$$581 + 33 + 350 + 41 = 1005 \rightarrow O(1)' \text{ dir.}$$

7.1.4 Sohraby Kriterinin Karmaşıklık Analizi

Sohraby, x tampon boyutu olmak üzere sistemdeki hücre kayıp olasılığının yaklaşık olarak:

$\gamma (1/z^*)^x$ şeklinde hesaplanabileceğini göstermiştir [9].

Burada γ , trafik yoğunluğunu gösterir. R kaynağın tepe hızını, r kaynağın aktif olduğu zaman bölümünü, b aktif periyodun ortalama süresini göstermek üzere, N tane IBP tipi kaynak için z^* şu şekilde hesaplanır [9] :

$$z^* \approx 1 + \frac{1 - r}{\sum_{i=1}^N r_i R_i (1 - r_i)^2 b_i} \quad (r = \sum_{i=1}^N r_i R_i) \quad (7.7)$$

Bu durumda yeni bağlantının kabul edilip edilmeyeceğine karar vermek çok kolaydır:

$\gamma (1/z^*)^x < \varepsilon$ ise, yeni bağlantı isteği kabul edilir.

Sohraby tarafından önerilen bu kriterin karmaşıklığı 3 adımda hesaplanmıştır :

1. z^* hesaplanır.

Bu hesaplamada kullanılan $(r = \sum_{i=1}^N r_i R_i)$ ve $(A = \sum_{i=1}^N r_i R_i (1 - r_i)^2 b_i)$ ifadelerinin

mevcut bağlantılar için link metrik vektöründe tutulduğu varsayılmıştır. Bu durumda link metrik vektörü L_j aşağıdaki biçimde tanımlanır :

$$L_j = (r = \sum_{i=1}^N r_i R_i, \quad A = \sum_{i=1}^N r_i R_i (1-r_i)^2 b_i)$$

z^* hesaplanırken sadece yeni bağlantı isteğinde bulunan kaynak için r ve A değerleri hesaplanır. Hesaplanan değerler link metrik vektörüne eklenerek link metrik vektörü güncellenir.

r nin hesaplanmasında :

$r = r_i R_i$ formülü kullanılır. Yeni bağlantı isteğinde bulunan kaynağın r_i ve R_i değerleri bağlantı vektöründen okunur, çarpma işlemi ile r hesaplanır ve bu değer istek vektörüne yazılır. Toplam işlem sayısı :

$$2 \text{ okuma} + 1 \text{ çarpma} + 1 \text{ yazma} = 4 \text{ işlem}$$

Karmaşıklık :

$$2 \times 1 + 1 \times 11 + 1 \times 1 = 14 \rightarrow O(1)' \text{ dir.}$$

A nın hesaplanmasında :

$A = r_i R_i (1-r_i)^2 b_i$ formülü kullanılır. r_i , R_i ve b_i değerleri okunur. Formüle göre yeni bağlantı için A hesaplanır ve bu değer istek vektörüne yazılır. Toplam işlem sayısı :

$$3 \text{ okuma} + 1 \text{ kare alma} + 3 \text{ çarpma} + 1 \text{ çıkarma} + 1 \text{ yazma} = 9 \text{ işlem}$$

Karmaşıklık :

$$3 \times 1 + 1 \times 11 + 3 \times 11 + 1 \times 8 + 1 \times 1 = 56 \rightarrow O(1)' \text{ dir.}$$

Yeni bağlantı için hesaplanan r ve A değerlerinin bulunduğu istek vektörü, link metrik vektörüne eklenerek link metrik vektörü güncellenir. Bunun için toplam işlem sayısı :

$$4 \text{ okuma} + 1 \text{ vektör toplama} + 2 \text{ yazma} = 7 \text{ işlem}$$

Karmaşıklık:

$$4 \times 1 + 8 \times 2 + 2 \times 1 = 22 \rightarrow O(1)' \text{ dir.}$$

Güncellenmiş link metrik vektöründeki değerler kullanılıncaya, z^* ı (7.7)' den hesaplamak için 1 toplama, 1 bölme ve 1 çıkarma işlemi yeterli olacaktır. z^* ı hesaplamak için toplam işlem sayısı:

2 okuma + 1 toplama + 1 çıkarma + 1 bölme = 5 işlem

Karmaşıklık:

$$2 \times 1 + 1 \times 8 + 1 \times 8 + 1 \times 8 = 26 \rightarrow O(1)' \text{ dir.}$$

z^* değerini hesaplamak için birinci adımda toplam işlem sayısı :

$$4 + 9 + 7 + 5 = 25 \text{ işlem}$$

Birinci adımın karmaşıklığı :

$$14 + 56 + 22 + 26 = 118$$

2. $\gamma (1/z^*)^x$ hesaplanır. Bunun için toplam işlem sayısı :

2 okuma + 1 bölme + 1 üs alma + 1 çarpma = 5 işlem

Karmaşıklık:

$$2 \times 1 + 1 \times 8 + 1 \times 196 + 1 \times 11 = 217 \rightarrow O(1)' \text{ dir.}$$

3. İkinci adımda hesaplanan $\gamma (1/z^*)^x$ değeri ile ε karşılaştırılır. $\gamma (1/z^*)^x < \varepsilon$ ise, yeni bağlantı isteği kabul edilir. Bunun için toplam işlem sayısı :

$$1 \text{ okuma} + 1 \text{ karşılaştırma} = 2 \text{ işlem}$$

Karmaşıklık:

$$1 \times 1 + 1 \times 15 = 16 \rightarrow O(1)' \text{ dir.}$$

Sohraby Kriteri ile çağrının kabul edilip edilmeyeceğine karar vermek için 3 adım sonunda toplam :

$$25 + 5 + 2 = 32 \text{ işlem gerekir.}$$

Toplam karmaşıklık ise:

$$118 + 217 + 16 = 351 \rightarrow O(1)' \text{ dir.}$$

7.2. Benzetim Programlarının Karmaşıklık Analizi

Bu bölümde, Bölüm.6' da açıklanan üç farklı benzetim programının karmaşıklık analizi yapılmıştır.

Yazılan benzetim programlarında kabul edilebilecek maksimum bağlantı sayısı (N) hesaplanır. Maksimum bağlantı sayısını gösteren N sayacının başlangıç değeri 1' dir ve kabul edilebilecek her bağlantı için bu sayaç 1 arttırılır. Yeni bir bağlantının kabul edilip edilmeyeceğine karar verme işlemi CELL_LOSS alt programında gerçekleşir. CELL_LOSS alt programı false değerini döndürürse, bağlantı sayısını gösteren sayaç 1 arttırılır.

Benzetim programlarının karmaşıklığı incelenirken, o an mevcut bağlantı sayısı (N)' in bilindiği varsayılarak, (N+1) için yani yeni bağlantı isteği için CELL_LOSS alt programının karmaşıklığı hesaplanmıştır. Bu alt program, yeni bir bağlantı isteğinin kabul edilip edilmeyeceğine karar verme aşamasıdır.

7.2.1 Kaynakların Aynı Anda İletime Başladığı ve ON-OFF Sürelerinin Sabit Olduğu Durum İçin Yazılan Benzetim Programının Karmaşıklık Analizi

Kaynak kodu Bölüm 6.1.2' de verilen ve kaynakların aynı anda ilettime başladığı, ON-OFF sürelerinin sabit olduğu durum için yazılan benzetim programının karmaşıklığı 4 adımda incelenebilir :

1. Bütün kaynakların yolladığı toplam hücre sayısı hesaplanır. Bu hesaplama YOL_HUCRE_SAY alt programında yapılır. Bu alt programda öncelikle kaynağın ortalama ON periyodunun süresi ton ve kaynağın ortalama OFF periyodunun süresi toff değerleri okunur. Bu değerler toplanarak Tp değeri hesaplanır. Daha sonra benzetim süresi tsim okunur ve Tp ile karşılaştırılır. Bu aşamaya kadar yapılan toplam işlem sayısı :

$$4 \text{ okuma} + 1 \text{ toplama} + 1 \text{ yazma} + 1 \text{ karşılaştırma} = 7 \text{ işlem}$$

Karmaşıklık :

$$4 \times 1 + 1 \times 8 + 1 \times 1 + 1 \times 15 = 28 \rightarrow O(1)' \text{ dir.}$$

Benzetim süresi tsim' in, $T_p = t_{on} + t_{off}$ değerinden küçük veya eşit olması durumunda :

5 okuma + 1 karşılaştırma + 2 çarpma + 1 yazma = 9 işlem yapılır.

Karmaşıklık :

$$5 \times 1 + 1 \times 15 + 2 \times 11 + 1 \times 1 = 43 \rightarrow O(1)' \text{ dir.}$$

Benzetim süresi tsim' in, $T_p = t_{on} + t_{off}$ değerinden büyük olması durumunda :

9 okuma + 5 çarpma + 1 karşılaştırma + 1 toplama + 1 yazma = 17 işlem yapılır.

Karmaşıklık :

$$9 \times 1 + 5 \times 11 + 1 \times 15 + 1 \times 8 + 1 \times 1 = 88 \rightarrow O(1)' \text{ dir.}$$

Birinci adımda toplam (en kötü durumda):

$$7 + 17 = 24 \text{ işlem gerekir.}$$

Birinci adımın karmaşıklığı ise :

$$28 + 88 = 116 \rightarrow O(1)' \text{ dir.}$$

2. Tampondan atılan hücre sayısı hesaplanır. Bu hesaplama AT_HUCRE_SAYISI alt programında yapılır. Bu alt programda öncelikle YOL_HUCRE_SAY alt programı çağrılır ve bunun döndürdüğü değer, yollanan değişkenine yazılır. Daha sonra yollanan hücre sayısından çıkış portunun işlediği hücre sayısı çıkartılır (yollanan – (W*tsim)) ve bu değer tampon boyutu x ile karşılaştırılır. Bu aşamaya kadar yapılan işlem sayısı :

$$4 \text{ okuma} + 1 \text{ çarpma} + 1 \text{ çıkarma} + 1 \text{ karşılaştırma} = 7 \text{ işlem}$$

Karmaşıklık :

$$4 \times 1 + 1 \times 11 + 1 \times 8 + 1 \times 15 = 38 \rightarrow O(1)' \text{ dir.}$$

(yollanan – (W*tsim)) değerinin tampon boyutu x ' den büyük olması durumunda:

$$4 \text{ okuma} + 1 \text{ çarpma} + 2 \text{ çıkarma} + 1 \text{ yazma} = 8 \text{ işlem}$$

Karmaşıklık :

$$4 \times 1 + 1 \times 11 + 2 \times 8 + 1 \times 1 = 32 \rightarrow O(1)' \text{ dir.}$$

(yollanan – (W*tsim)) değerinin tampon boyutu x^6 den küçük veya eşit olması durumunda:

1 yazma = 1 işlem.

Karmaşıklık :

$1 \times 1 = 1 \rightarrow O(1)$ ' dir.

İkinci adımda toplam (en kötü durumda):

$7 + 8 = 15$ işlem gerekir.

İkinci adımın karmaşıklığı ise :

$38 + 32 = 70 \rightarrow O(1)$ ' dir.

3. CELL_LOSS alt programında hücre kayıp oranı (atılan hücre sayısı / yollanan hücre sayısı) hesaplanır. Bunun için önce AT_HUCRE_SAYISI alt programı çağrılır ve bunun döndürdüğü değer atılan_say değişkenine yazılır. Global olarak tanımlanmış ve bütün kaynakların yolladığı toplam hücre sayısını tutan yollanan değişkeni okunur. (atılan_say / yollanan) işlemi ile hücre kayıp oranı elde edilir. Bulunan bu değer clp2 değişkenine yazılır. Bu adımda toplam işlem sayısı :

2 okuma + 1 bölme + 1 yazma = 4 işlem

Karmaşıklık :

$2 \times 1 + 1 \times 8 + 1 \times 1 = 11 \rightarrow O(1)$ ' dir.

4. Yeni çağrı isteğinin kabul edilip edilmeyeceğine karar vermek için üçüncü adımda hesaplanan hücre kayıp oranı (clp2) ile kullanıcı tarafından girilen, sistemde sağlanması istenen hücre kayıp olasılığı ϵ karşılaştırılır. Hücre kayıp oranı, ϵ değerinden küçük veya eşit ise, CELL_LOSS alt programı false değerini döndürür. Ana programda da CELL_LOSS alt programının döndürdüğü değer okunarak, bu değer false olması durumunda kabul edilebilecek maksimum bağlantı sayısını gösteren N değişkeni 1 arttırılır. Bu adımda yapılan toplam işlem sayısı :

3 okuma + 2 karşılaştırma + 1 bir arttırma işlemi + 1 yazma = 7 işlem

Karmaşıklık :

$$3 \times 1 + 2 \times 15 + 1 \times 8 + 1 \times 1 = 42 \rightarrow O(1)' \text{ dir.}$$

Kaynakların aynı anda ilettime başladığı ve ON-OFF sürelerinin sabit olduğu durum için yazılan birinci benzetim programında, yeni bağlantı isteğinin kabul edilip edilmeyeceğine karar vermek için gereken toplam işlem sayısı :

$$24 + 15 + 4 + 7 = 50$$

Birinci benzetim programının karmaşıklığı ise :

$$116 + 70 + 11 + 42 = 239 \rightarrow O(1)' \text{ dir.}$$

7.2.2 Kaynakların Belli Bir Aralıkta Rastgele İletime Başladığı ve ON-OFF Sürelerinin Sabit Olduğu Durum İçin Yazılan Benzetim Programının Karmaşıklık Analizi

Kaynak kodu Bölüm 6.2.2' de verilen ve kaynakların belli bir aralıkta rastgele ilettime başladığı ve ON-OFF sürelerinin sabit olduğu durum için yazılan benzetim programının karmaşıklığı 5 adımda incelenmiştir. Bu benzetim programında yaşam süresi seçeneğine göre yollanan hücre sayısını hesaplayan iki ayrı alt program vardır.

1. Bütün kaynakların yolladığı toplam hücre sayısı, yaşam süresinin sonsuz seçilmesi durumunda YOL_HUCRE_SAY alt programı ile hesaplanır. Bu alt programda; kaynakların rastgele üretilen ilettime başlama anlarından ve kullanıcı tarafından girilen ortalama ON periyodunun süresinden (ton) yararlanarak bir döngü ile kaynakların yolladığı toplam hücre sayısı hesaplanır. Bu döngü, herbir kaynak için bir kez yürütülür. Sistemde N tane kaynak olduğu varsayılırsa bu döngü N kez yürütülecektir. Benzetim programlarının yeni bir çağrı isteğini kabul edip etmemeye karar verme aşamasındaki karmaşıklığı incelendiğinden; mevcut bağlantılar için bu bilginin (mevcut kaynakların yolladığı toplam hücre sayısı) bir yerde tutulduğu varsayılarak bu adımın karmaşıklık analizi yapılmıştır. Bu durumda döngüye gerek kalmaz, sadece yeni bağlantı isteğinde bulunan kaynağın yolladığı hücre sayısı hesaplanır. Bu varsayım altında bu alt programda öncelikle; kaynağın ortalama ON periyodunun süresi ton ve kaynağın ortalama OFF periyodunun süresi toff değerleri okunur. Bu değerler toplanarak Tp değeri hesaplanır. Daha sonra benzetim süresi tsim okunur ve Tp ile karşılaştırılır. Bu aşamaya kadar yapılan toplam işlem sayısı :

$$4 \text{ okuma} + 1 \text{ toplama} + 1 \text{ yazma} + 1 \text{ karşılaştırma} = 7 \text{ işlem}$$

Karmaşıklık :

$$4 \times 1 + 1 \times 8 + 1 \times 1 + 1 \times 15 = 28 \rightarrow O(1)' \text{ dir.}$$

Benzetim süresi tsim' in, $T_p = t_{on} + t_{off}$ değerinden küçük veya eşit olması durumunda :

$$8 \text{ okuma} + 2 \text{ karşılaştırma} + 1 \text{ çarpma} + 1 \text{ çıkarma} + 1 \text{ toplama} + 1 \text{ yazma} = 14 \text{ işlem}$$

Karmaşıklık :

$$8 \times 1 + 2 \times 15 + 1 \times 11 + 1 \times 8 + 1 \times 8 + 1 \times 1 = 66 \rightarrow O(1)' \text{ dir.}$$

Benzetim süresi tsim' in, $T_p = t_{on} + t_{off}$ değerinden büyük olması durumunda:

$$11 \text{ okuma} + 3 \text{ çarpma} + 1 \text{ çıkarma} + 2 \text{ karşılaştırma} + 2 \text{ toplama} + 1 \text{ yazma} = 20 \text{ işlem}$$

Karmaşıklık :

$$11 \times 1 + 3 \times 11 + 1 \times 8 + 2 \times 15 + 2 \times 8 + 1 \times 1 = 99 \rightarrow O(1)' \text{ dir.}$$

Bu adımda toplam (en kötü durumda):

$$7 + 20 + 1 \text{ okuma} + 1 \text{ yazma} = 29 \text{ işlem gerekir.}$$

Bu adımın karmaşıklığı ise :

$$28 + 99 + 1 \times 1 + 1 \times 1 = 129 \rightarrow O(1)' \text{ dir.}$$

2. Kaynakların yolladığı toplam hücre sayısı, yaşam süresinin belirli seçilmesi durumunda YS_YOL_HUCRE_SAY alt programı ile hesaplanır. Bu alt programda öncelikle, ana programda hesaplanan ve global olarak saklanan p_sayisi değişkeni, kaynağın tepe hızı Rpeak, kaynağın ortalama ON periyodunun süresi ton, kaynak sayısı N okunur. Daha sonra çarpma işlemleri ile yollanan hücre sayısı hesaplanır. Bu adımdaki toplam işlem sayısı :

$$4 \text{ okuma} + 3 \text{ çarpma} + 1 \text{ yazma} = 8 \text{ işlem}$$

Karmaşıklık ise :

$$4 \times 1 + 3 \times 11 + 1 \times 1 = 38 \rightarrow O(1)' \text{ dir.}$$

3. Tampondan atılan hücre sayısı hesaplanır. Bu hesaplama AT_HUCRE_SAYISI alt programında yapılır. Bu alt programda öncelikle yollanan hücre sayısını

hesaplamak için yaşam süresi seçeneğine göre hangi alt programın çağrılacağına karar verilir. Çağrılan alt programın döndürdüğü değer ise yollanan değişkenine yazılır. Bu adımda yapılan sonraki işlemler, kaynakların aynı anda iletme geçtiği varsayımı altında yazılan birinci benzetim programının 2. adımında yapılan işlemler ile aynıdır. Tampondan atılan hücre sayısının hesaplanmasında ikinci benzetim programının birincisinden tek farkı; fazladan yapılan bir karşılaştırma işlemidir. Bu durumda toplam işlem sayısı :

$$1 \text{ karşılaştırma} + 15 \text{ işlem} = 16 \text{ işlem}$$

Karmaşıklık :

$$70 + 1 \times 15 = 85 \rightarrow O(1)' \text{ dir.}$$

4. CELL_LOSS alt programında hücre kayıp oranı (atılan hücre sayısı / yollanan hücre sayısı) hesaplanır. Bu adımda yapılan işlemler, birinci benzetim programının 3. adımında yapılan işlemlerin aynısıdır. Bölüm 7.2.1' de gösterildiği gibi bu adımdaki işlem sayısı : 4 ,

Bu adımın karmaşıklığı ise : 11 $\rightarrow O(1)'$ dir.

5. Yeni çağrı isteğinin kabul edilip edilmeyeceğine karar vermek için bir önceki adımda hesaplanan hücre kayıp oranı (clp2) ile kullanıcı tarafından girilen, sistemde sağlanması istenen hücre kayıp olasılığı ε karşılaştırılır. Bu adımda yapılan işlemler, birinci benzetim programının 4. adımında yapılan işlemlerin aynısıdır. Bölüm 7.2.1' de gösterildiği gibi bu adımdaki işlem sayısı : 7 ,

Bu adımın karmaşıklığı ise : 42 $\rightarrow O(1)'$ dir.

Kaynakların belli bir aralıkta rastgele iletme başladığı ve ON-OFF sürelerinin sabit olduğu durum için yazılan ikinci benzetim programında, yeni bağlantı isteğinin kabul edilip edilmeyeceğine karar vermek için gereken toplam işlem sayısı (en kötü durum):

$$29 + 16 + 4 + 7 = 56$$

İkinci benzetim programının karmaşıklığı ise :

$$129 + 85 + 11 + 42 = 267 \rightarrow O(1)' \text{ dir.}$$

7.2.3 Kaynakların Belli Bir Aralıkta Rastgele İletime Başladığı ve ON-OFF Sürelerinin Üssel Dağılıma Göre Rastgele Değiştiği Durum İçin Yazılan Benzetim Programının Karmaşıklık Analizi

Kaynak kodu Bölüm 6.3.2' de verilen ve kaynakların belli bir aralıkta rastgele iletime başladığı ve ON-OFF sürelerinin değişken olduğu durum için yazılan üçüncü benzetim programında ikinci benzetim programında olduğu gibi, yaşam süresi seçeneğine göre yollanan hücre sayısını hesaplayan iki ayrı alt program vardır. Benzetim programının karmaşıklığı 5 adımda incelenmiştir :

1. Bütün kaynakların yolladığı toplam hücre sayısı, yaşam süresinin sonsuz seçilmesi durumunda YOL_HUCRE_SAY alt programı ile hesaplanır. Bu alt programda; kaynakların rastgele üretilen ilettime başlama anlarından yararlanılarak bir döngü ile kaynakların yolladığı toplam hücre sayısı hesaplanır. Bu döngü, herbir kaynak için bir kez yürütülür. Sistemde N tane kaynak olduğu varsayılırsa bu döngü N kez yürütülecektir. İkinci benzetim programından farklı olarak bu benzetim programında kaynakların ON periyodunun süresi değişken olduğundan; her bir kaynağın benzetim süresi içerisinde iletimde kalma süresi (ON periyodlarının toplamı) hesaplanmalıdır. Bunun için on_topla ve artik_periyod değişkenleri kullanılır. on_topla ve artik_periyod değişkenlerini hesaplamak için, her kaynak için yürütülen döngü içinde ikinci bir döngü yürütülür. İkinci benzetim programında olduğu gibi bu benzetim programında da, bağlantı halinde bulunan mevcut kaynakların yolladığı toplam hücre sayısının (yol_top) bir yerde tutulduğu varsayılmıştır. Ayrıca sadece bu programda bir döngü ile hesaplanması gereken ve her bir kaynak için farklı değerlere sahip olan on_topla ve artik_periyod değerlerinin de bir yerde tutulduğu varsayılmış ve bu değerler için sadece okuma işlemi yapılmıştır. Bu durumda her iki döngüye de gerek kalmaz. Yapılan varsayımlar altında bu adımın karmaşıklığı şu şekilde hesaplanabilir :

İlk olarak periyod bitiminde benzetim süresi içinde zaman kalıp kalmadığına (artik_periyod) bakılır. Eğer artik_periyod varsa bu değer, değişken olan kaynağın son ON periyodunun süresinden büyük olup olmadığına bakılır çeşitli işlemlerle kaynağın yolladığı toplam hücre sayısı hesaplanır. Bu adımda yapılan toplam işlem sayısı :

$$9 \text{ okuma} + 2 \text{ karşılaştırma} + 2 \text{ çarpma} + 2 \text{ toplama} + 2 \text{ yazma} = 17 \text{ işlem}$$

Bu adımın karmaşıklığı ise :

$$9 \times 1 + 2 \times 15 + 2 \times 11 + 2 \times 8 + 2 \times 1 = 79 \rightarrow O(1)' \text{ dir.}$$

2. Kaynakların yolladığı toplam hücre sayısı, yaşam süresinin belirli seçilmesi durumunda YS_YOL_HUCRE_SAY alt programı ile hesaplanır. Bu alt programda da YOL_HUCRE_SAY alt programında olduğu gibi, bir döngü ile kaynakların yolladığı toplam hücre sayısı hesaplanır. Herbir kaynak için bir kez yürütülen bu döngü içinde, her bir kaynağın yaşam süresi içinde iletimde bulunduğu süre (on_topla) hesaplanır. Yaşam süresinin belirli seçilmesi durumunda bu alt program yürütüldüğünden ve benzetim süresi daima kullanıcı tarafından girilen yaşam süresinin iki katına eşit olduğundan bütün kaynaklar benzetim süresi içinde iletimlerini tamamlarlar. Bu alt programda da, bağlantı halinde bulunan mevcut kaynakların yolladığı toplam hücre sayısının (yol_top) ve bir döngü ile hesaplanması gereken ve her bir kaynak için farklı değerlere sahip olan on_topla değerinin bir yerde tutulduğu varsayılmıştır ve bu değerler için sadece okuma işlemi yapılmıştır. Bu adımdaki işlem sayısı :

$$4 \text{ okuma} + 1 \text{ çarpma} + 1 \text{ toplama} + 2 \text{ yazma} = 8 \text{ işlem}$$

Bu adımın karmaşıklığı ise :

$$4 \times 1 + 1 \times 11 + 1 \times 8 + 2 \times 1 = 25 \rightarrow O(1)' \text{ dir.}$$

3. Tampondan atılan hücre sayısı hesaplanır. Bu hesaplama AT_HUCRE_SAYISI alt programında yapılır. Bu adımda yapılan işlemler, ikinci benzetim programının 3. adımında yapılan işlemlerin aynısıdır. Bölüm 7.2.2' de gösterildiği gibi bu adımdaki işlem sayısı : 16 ,

Bu adımın karmaşıklığı ise : 85 $\rightarrow O(1)$ ' dir.

4. CELL_LOSS alt programında hücre kayıp oranı (atılan hücre sayısı / yollanan hücre sayısı) hesaplanır. Bu adımda yapılan işlemler, birinci benzetim programının 3. adımında yapılan işlemlerin aynısıdır. Bölüm 7.2.1' de gösterildiği gibi bu adımdaki işlem sayısı : 4 ,

Bu adımın karmaşıklığı ise : 11 $\rightarrow O(1)$ ' dir.

5. Yeni çağrı isteğinin kabul edilip edilmeyeceğine karar vermek için bir önceki adımda hesaplanan hücre kayıp oranı (clp2) ile kullanıcı tarafından girilen, sistemde sağlanması istenen hücre kayıp olasılığı ϵ karşılaştırılır. Bu adımda yapılan işlemler, birinci benzetim programının 4. adımında yapılan işlemlerin aynısıdır. Bölüm 7.2.1' de gösterildiği gibi bu adımdaki işlem sayısı : 7 ,

Bu adımın karmaşıklığı ise : $42 \rightarrow O(1)$ ' dir.

Kaynakların belli bir aralıkta rastgele iletme başladığı ve ON-OFF sürelerinin değişken olduğu durum için yazılan üçüncü benzetim programında, yeni bağlantı isteğinin kabul edilip edilmeyeceğine karar vermek için gereken toplam işlem sayısı (en kötü durum) :

$$17 + 16 + 4 + 7 = 44$$

Üçüncü benzetim programının karmaşıklığı ise :

$$79 + 85 + 11 + 42 = 217 \rightarrow O(1)' \text{ dir.}$$



8. SONUÇLAR VE TARTIŞMA

Asenkron Transfer Modu (ATM), telefon ve veri ağlarının en iyi özelliklerini birleştirmeye yönelik bir girişimdir. Farklı trafik karakteristiğine sahip uygulamalara yüksek band genişliği ve garantili servis kalitesi sağladığından son zamanlarda ATM çok fazla uygulama alanı bulmaktadır. Ancak ATM ağlarda da çözülmesi gereken sorunlar vardır. Diğer ağlarda olduğu gibi tıkanıklık, ATM ağlar için de başarılması gereken temel sorunlar arasında yer alır. Her ne kadar ATM yüksek band genişliği sağlayan bir topoloji olsa da, gelişmeler sağlanan yüksek band genişliğinin de hızlı bir şekilde doldurulacağını göstermektedir. Bu yüzden ATM ağları için temel problemlerden biri olan bu konunun çözüme kavuşturulması gerekir. Tıkanıklık denetimi (congestion control), bu önemi ile şu an en popüler çalışma konularından birini oluşturur.

Tıkanıklık denetiminin temel hedefi; ağda herhangi bir andaki toplam istek miktarı, ağın taşıyabileceği kapasiteye eşit olduğunda veya bunu aştığında ağın ve kullanıcıların bu durumdan en az derecede etkilenmesini sağlamak ve ağ kaynaklarının kullanımını optimize etmektir. Farklı trafik karakteristiklerine hizmet veren ATM ağlarda tıkanıklık denetimi diğer ağlardan çok daha zordur ve bunun için çok sayıda yöntem önerilmiştir.

ATM ağlarda tıkanıklığı denetlemek için kullanılan yöntemlerden biri de, çağrı kabul denetimidir (call admission control-CAC). Çağrı kabul denetimi, bir çağrı isteğinin kabul edilip edilmeyeceğine karar verme işlemidir. Bu yöntemlerin ATM ağlarda gerçekleştirilmesi diğer ağlara göre daha zordur. Çünkü bu yöntemler ATM ağlarda çok sayıda yüksek hızlı uygulamayı aynı anda kotaracak şekilde tasarlanmalıdır. Üstelik ağdaki bütün düğümler çok kısa bir zaman aralığı içinde binlerce bağlantı isteği alabilirler.

Ağda yeni bir bağlantı isteği alındığı zaman, bu isteğin kabul edilip edilmeyeceğine karar vermek için bir kabul kriteri yürütülür. Ağ, kurulu bağlantıların servis kalitelerini etkilemeden yeni bağlantının gerektirdiği servis kalitesini sağlayabilecek durumda ise bağlantı isteği kabul edilir. Bu soruya cevap vermek için tasarlanan herhangi bir kabul kriteri gerçek zamanda yürütülür. Problem oldukça karmaşıktır. Çünkü bir kabul kriterinin sadece bütün bağlantıların servis kalitesi sağlanacak

şekilde doğru karar vermesi yetmez, kabul kriteri aynı zamanda ağ kaynaklarının etkili bir biçimde kullanılmasını sağlamalıdır.

Herhangi bir kabul denetimi algoritmasının temeli, bir grup kaynak için gereken band genişliğini tahmin etmeye dayanır. Eşdeğer band genişliği yaklaşımı da, çoğullanan kaynaklar için gerekli band genişliğini tahmin etmede kullanılan tekniklerden biridir.

Bu çalışmada ATM ağlarda tıkanıklığı denetlemek için kullanılan yöntemler incelenmiş ve eşdeğer band genişliği yaklaşımına dayalı kabul denetimi algoritmaları üzerinde yoğunlaşmıştır. Eşdeğer band genişliği yaklaşımına dayalı kabul denetim yöntemlerinden Guerin-Ahmadi-Naghshineh (GAN), Elwalid-Mitra, Geliştirilmiş Elwalid-Mitra ve Sohraby kriterleri incelenmiştir. Bu dört kabul kriterinin başarımlarını karşılaştırabilmek amacıyla, çeşitli kaynak modellerinde ve tampon boyutlarında bu kriterlerin ayrı ayrı kabul bölgesi değerleri hesaplanmıştır. Kabul bölgesi; istenen servis kalitesinin sağlanabildiği durumda kabul edilebilecek maksimum bağlantı sayısına karşı düşer. Bu hesaplamayı yapabilmek amacıyla, seçilen kabul kriterine ve girilen kaynak parametrelerine, tampon boyutuna, çıkış portunun hızına ve hücre kayıp olasılığına göre kabul bölgesini bulan bir PASCAL programı yazılmıştır.

Kriterlerle elde edilen kabul bölgesi değerlerine yorum getirebilmek amacıyla, kullanıcı tarafından girilen kaynak parametrelerine, tampon boyutuna, çıkış portunun hızına ve istenen servis kalitesi hücre kayıp olasılığına göre aynı tipteki kaynaklar için kabul edilebilecek maksimum bağlantı sayısını hesaplayan, PASCAL programlama dilinde üç farklı benzetim programı yazılmıştır. Yazılan benzetim programlarında bir çıkış portunun hizmet verdiği sonlu kapasiteli bir tamponun, ON-OFF modeliyle modellenen ATM kaynakları tarafından beslendiği varsayılmıştır. ON-OFF modelinde kaynağın hücreleri tepe hızıyla gönderdiği bir ON periyodu, ve hiç bir hücrenin üretilmediği bir OFF periyodu vardır. ON periyodu sırasında hücrelerin arka arkaya gönderildiği, iki hücre arasında hiç zaman boşluğu olmadığı varsayılmıştır.

Programlarda öncelikle, bütün kaynakların ortak olarak kullandığı varsayılan ve büyüklüğü kullanıcı tarafından girilen tamponun doluluk oranından yararlanılarak hücre kayıp oranı hesaplanır. Hesaplanan bu oran, istenen servis kalitesi olarak kabul edilen ve kullanıcı tarafından girilen hücre kayıp olasılığından küçük olduğu süreçte sistemde bulunan kaynakların hepsi kabul edilebilirdir. Bu özellikten yararlanarak benzetim programları ile sistemde kabul edilebilecek maksimum bağlantı sayısı

hesaplanır. Benzetim programlarıyla bu hesaplama yapılırken, bağlantı isteğinde bulunan bütün kaynakların aynı tip kaynak olduğu varsayılmıştır.

İlk iki benzetim programında, sistemde bulunan kaynakların ON ve OFF periyodlarının süresi sabit ve kullanıcı tarafından girilen ortalama sürelerle eşit alınmıştır. Üçüncü benzetim programında ise, kaynakların ON ve OFF periyodlarının süresi sürekli değişir. Üçüncü programda bu periyodların süreleri belli bir aralıkta rastgele olarak üretilmiştir. İlk iki benzetim programı, “smooth” kaynaklar için yazılmıştır. Üçüncü benzetim programında ise “yığınsal (bursty)” kaynaklar göz önüne alınmıştır.

İlk benzetim programı, kaynakların ilettime başlamasında olabilecek en kötü durumu ele almış, bütün kaynakların aynı anda ilettime başladığını varsaymıştır. İkinci ve üçüncü benzetim programlarında ise kaynaklar belli bir aralıkta rastgele ilettime başlarlar.

Ayrıca benzetim programlarında yaşam süresi kavramı da göz önüne alınmıştır. Yaşam süresi, bağlantının kurulduktan sonra ne kadar süre devam edeceğini gösteren değerdir. Programlarda kullanıcıya yaşam süresi için iki seçenek sunulur. Seçeneklerin birinde bağlantıların yaşam süresi sonsuzdur yani kaynaklar benzetim süresi boyunca sürekli iletimde bulunurlar. Diğer seçenekte ise bağlantıların belirli bir yaşam süresi vardır ve bu süre bitiminde kaynaklar iletimlerine son verirler.

İkinci ve üçüncü benzetim programlarında yaşam süresi seçeneği aynı zamanda, kaynakların ilettime başlama anlarının rastgele üretildiği aralığı da belirler. Eğer kullanıcı yaşam süresini sonsuz seçmişse, bütün kaynakların bir periyodluk süre içerisindeki rastgele anlarda ilettime başladıkları varsayılır ve kullanıcıdan benzetim süresini girmesi istenir. Girilen değerlere göre, kabul edilebilecek maksimum bağlantı sayısı hesaplanır. Eğer kullanıcı yaşam süresini belirli seçmişse, bütün kaynakların yaşam süresi içerisindeki rastgele anlarda ilettime başladıkları ve bu süre boyunca iletimde kaldıkları varsayılır ve kullanıcıdan yaşam süresini girmesi istenir. Bu durumda benzetim süresi daima yaşam süresinin iki katına eşittir. Girilen değerlere göre, kabul edilebilecek maksimum bağlantı sayısı hesaplanır.

Yazılan her üç benzetim programı da, eşdeğer band genişliği temelli dört kabul kriterinin kabul bölgesinin hesaplandığı örnek trafik kaynağı için, farklı tampon boyutlarında, benzetim ve yaşam sürelerinde çalıştırılarak kabul edilebilecek maksimum bağlantı sayısı için değerler bulunmuştur.

Birinci benzetim programında kaynakların iletimi açısından en kötü durum ele alınmış ve bütün kaynakların aynı anlarda iletimde bulunduğu varsayılmıştır. Yani

ON periyodu süresince bütün kaynaklar sürekli hücre gönderecekler, OFF periyodu sırasında ise hiçbir kaynak hücre göndermeyecektir. Bu yüzden birinci benzetim programı çalıştırılarak değerler elde edilirken, benzetim süresi ON periyodunun süresine eşit seçilmiştir. Dolayısıyla bütün kaynakların sürekli hücre gönderdiği bu benzetim ortamında kabul edilebilecek maksimum bağlantı sayısı için bulunan değerler, beklenildiği gibi küçüktür. Ancak aynı koşullarda küçük tampon boyutları için Sohraby kriteri ile daha küçük değerler bulunduğu görülmüştür. Küçük tampon boyutlarında Elwalid-Mitra kriteri ile elde edilen değerler, birinci benzetim programıyla bulunan değerlere yakındır. GAN ve Geliştirilmiş Elwalid-Mitra kriterleri ile bulunan değerler ise daha büyüktür.

İkinci benzetim programı hem yaşam süresinin sonsuz olması, hem de belirli olması seçeneği için çalıştırılmış ve ayrı ayrı değerler elde edilmiştir. Yaşam süresinin belirli seçilmesi durumunda maksimum bağlantı sayısı için elde edilen değerler, kaynakların yaşam süresinin sonsuz ve benzetim süresinin aynı seçilmesi durumunda elde edilen değerlerin yaklaşık iki katıdır. Bu beklenen bir sonuçtur. Çünkü yaşam süresinin belirli seçilmesi durumunda, benzetim süresi daima girilen yaşam süresinin iki katı alınmış ve kaynaklar benzetim süresinin yarısı kadar sürede, yani yaşam süresi boyunca iletimde bulunmuşlardır. İkinci benzetim programı ile elde edilen değerler, birinci benzetim programı ile elde edilen değerlere göre oldukça büyüktür. Ayrıca, bütün kriterler ile bütün tampon boyutlarında bulunan değerler, ikinci benzetim programı ile bulunan değerlerden küçüktür.

Yığınsal kaynakları modellemek için yazılan üçüncü benzetim programıyla, kabul edilebilecek maksimum bağlantı sayısı için ikinci benzetim programına göre daha küçük değerler bulunmuştur. Ancak bu değerler, birinci benzetim programı ile bulunan değerlerden büyüktür. İkinci benzetim programında olduğu gibi yaşam süresinin belirli seçilmesi durumunda elde edilen değerler, yaşam süresinin sonsuz seçilmesi ile bulunan değerlerin yaklaşık iki katıdır. Küçük ve orta tampon boyutlarında Sohraby ve Elwalid-Mitra kriterleri ile bulunan değerler, üçüncü benzetim programının çalıştırılması ile bulunan değerlerin altında kalır. GAN kriteri ile bulunan değerler bu benzetim programıyla bulunan değerlere yakındır. Geliştirilmiş Elwalid-Mitra kriteri ile bulunan değerler ise daha büyüktür.

Eşdeğer band genişliği yaklaşımına dayalı olarak birçok yöntem önerilmiştir. Guerin, Ahmadi ve Naghshineh tarafından ve Rege tarafından yapılan çalışmalarda, bir kaynak için bazı kriterlerle bulunan eşdeğer band genişliğinin, aynı koşullar altında deneysel olarak bulunandan çok daha yüksek olabileceği gösterilmiştir. Bu durum, özellikle küçük tampon boyutlarında geçerlidir. Bunun nedeni ise, bu yöntemlerin

çoğunun asimtotik koşullarda yani tampon boyutunun çok büyük ve hücre kayıp olasılığının çok küçük olduğu varsayımı altında türetilmiş olmalarıdır. Ayrıca bu yöntemlerin çoğu, istatistiksel çoğullamanın avantajlarını tam olarak hesaba katamazlar. İstatistiksel çoğullama, birden fazla sayıda bağlantının, tepe hızlarının toplamından daha az kapasiteye sahip olan bir linki paylaşabileceğini ifade eder. Guerin, Ahmadi, Naghshineh tarafından geliştirilen GAN kriterinde ve Rege tarafından geliştirilen ve Geliştirilmiş Elwalid Mitra kriteri olarak adlandırılan kriterde, istatistiksel çoğullamanın hesaba katılması için iyileştirme getirilmiş ve çeşitli simülasyonlarla bu kriterlerin birçok kriterden daha iyi sonuç verdiği gösterilmiştir. Rege ayrıca önerdiği iyileştirmenin, birçok kaynak tipi için birçok durumda Elwalid-Mitra ve Sohraby kriterlerinden hatta GAN kriterinden daha iyi sonuç verdiğini yaptığı simülasyon çalışmalarıyla göstermiştir. Rege tarafından geliştirilen Geliştirilmiş Elwalid Mitra kriteri, GAN kriterine benzemekle birlikte bu kriterde istatistiksel çoğullama kazancı hesaplamalarında tamponun etkisi de hesaba katılmıştır.

Bu çalışmada yazılan benzetim programları ile elde edilen sonuçlar da, Guerin, Ahmadi ve Naghshineh tarafından geliştirilen GAN kriteri ve Rege tarafından geliştirilen Geliştirilmiş Elwalid Mitra kriterinin incelenen diğer kriterlere göre daha iyi sonuçlar verdiğini doğrulamıştır. Benzetim programları ve kriterler için yazılan program, çeşitli tampon boyutlarında ve aynı örnek trafik kaynağı için çalıştırılarak maksimum bağlantı sayısı değerleri elde edilmiştir. Bu trafik kaynağı ON-OFF tipi bir kaynaktır ve tepe hızının çıkış portunun kapasitesine oranı küçüktür. Dolayısıyla bu kaynak için küçük tampon boyutlarında bile büyük istatistiksel kazanç sağlanabilir. Yazılan benzetim programlarının ve kriterler için yazılan programın böyle bir kaynak için çalıştırılması sonucu, istatistiksel çoğullama kazancını daha iyi hesaba katan yöntemlerin daha iyi sonuçlar verdiği gözlemlenmiştir.

Yazılan benzetim programlarında bütün kaynakların aynı tip olduğu varsayılmakla birlikte, bu programlar parametreleri farklı olan, farklı tipteki kaynaklar için hesaplama yapabilecek şekle kolayca dönüştürülebilir.

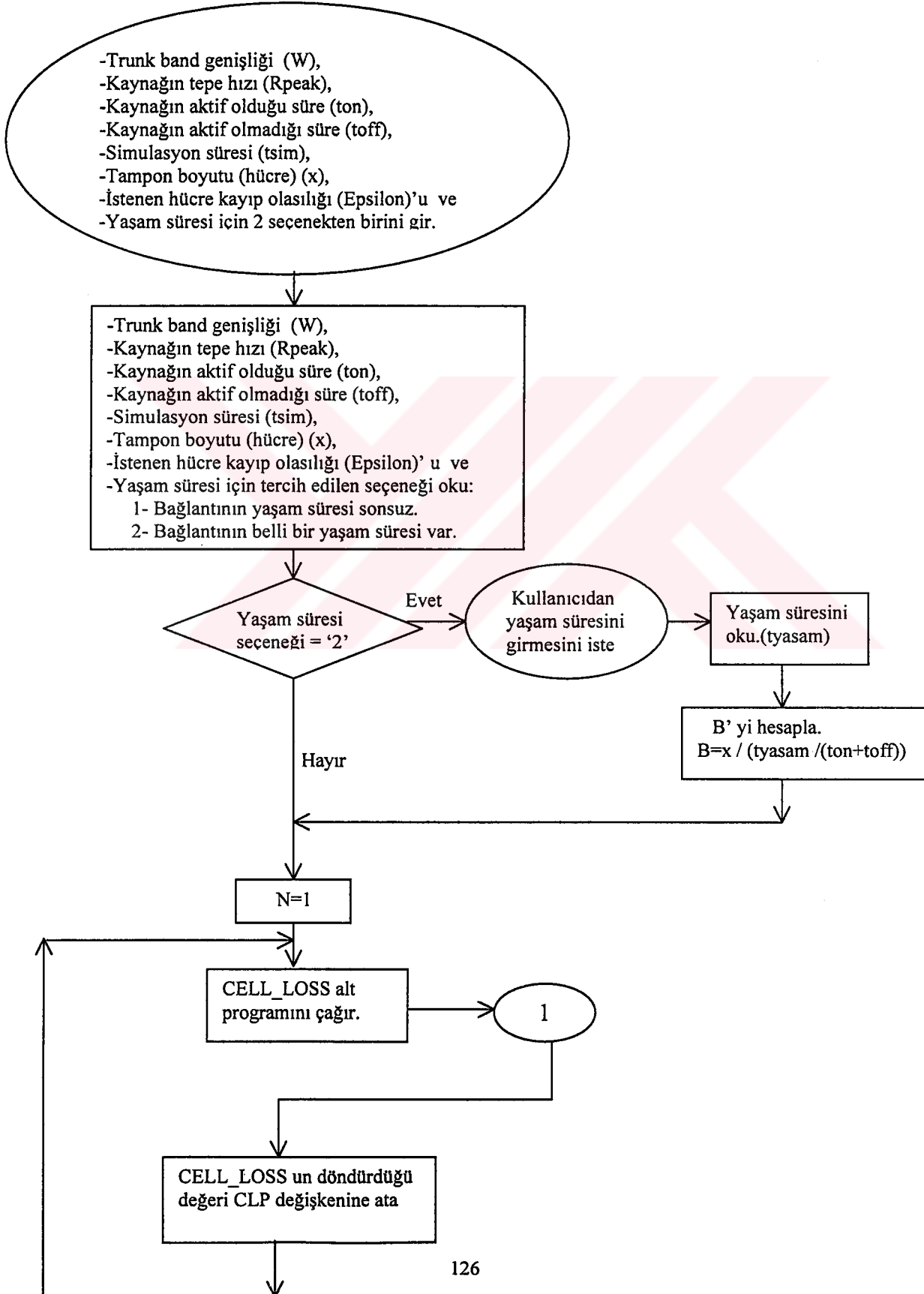
KAYNAKLAR

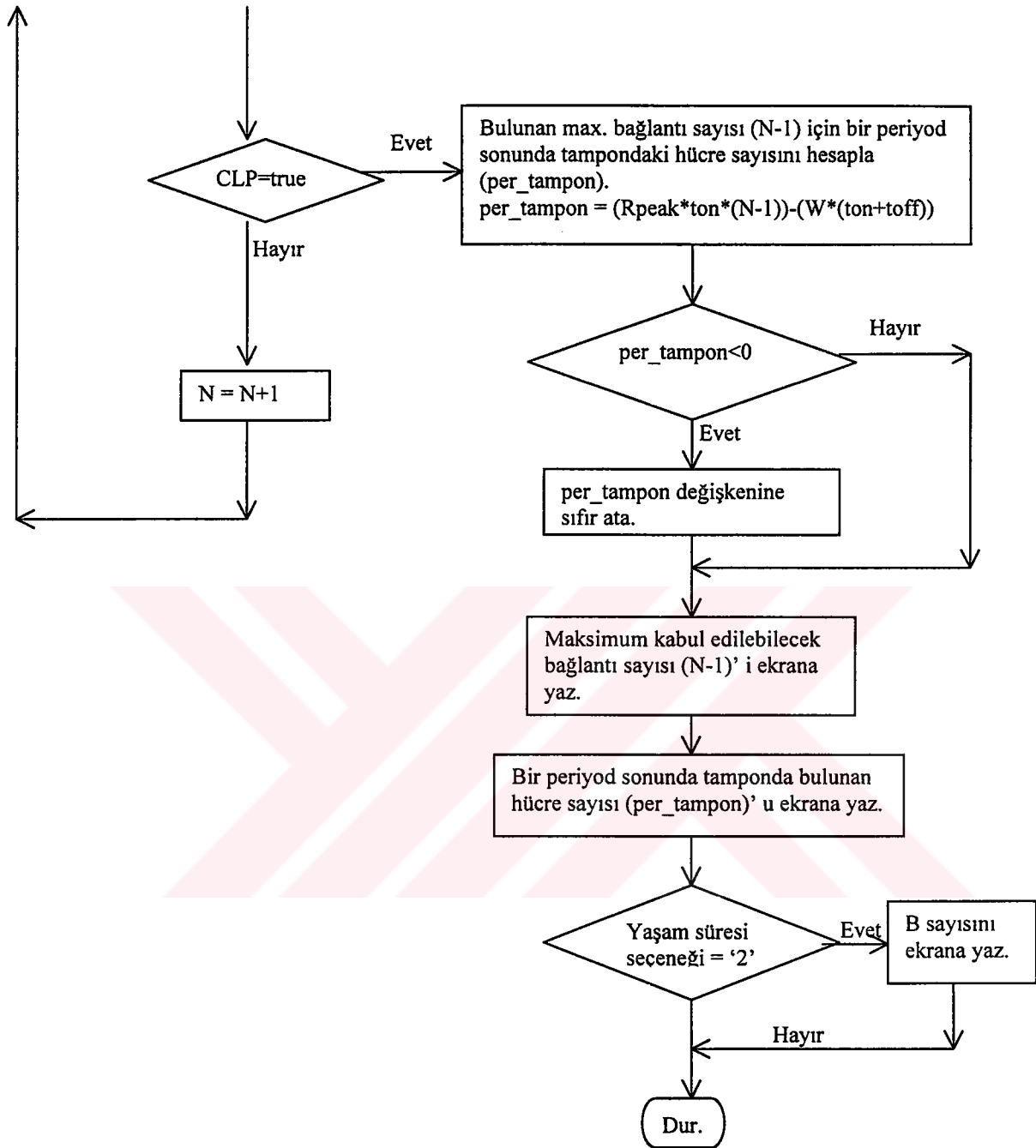
- [1] **Tanenbaum, Andrew S.**, 1996. Computer Networks, Prentice Hall, New Jersey.
- [2] **Jain, R.**, 1990. Congestion Control in Computer Networks : Issues and Trends, *IEEE Network Magazine*, vol.4, 24 – 30.
- [3] **Gallassi, G., Rigolio G. and Verri, L.**, 1990. Resource Management and Dimensioning in ATM Networks, *IEEE Network Magazine*, vol.4, 8-17.
- [4] **Cooper, C. A. and Park, K.I.**, 1990. Toward a Broadband Congestion Control Strategy, *IEEE Network Magazine*, vol.4, 18-23.
- [5] **Kung, H.T., and Morris, R.**, 1995. Credit Based Flow Control for ATM Networks, *IEEE Network Magazine*, vol.9, 40-48.
- [6] **Bonomi, F., and Fendick, K. W.**, 1995. The Rate Based Flow Control Framework for the Available Bit Rate ATM Service, *IEEE Network Magazine*, vol.9, 25-39.
- [7] **Ramakrishnan, K.K. and Newman, P.** 1995. Integration of Rate and Credit Schemes for ATM Flow Control, *IEEE Network Magazine*, vol.9, 49-56.
- [8] **Gelenbe, E., Mang, X., Önvural, R.**, 1997. Bandwith Allocation and Call Admission Control in High-Speed Networks, *IEEE Communications Magazine*, vol.35, 122-129.
- [9] **Perros, H. G., and Elsayed, K.M.**, 1996. Call Admission Control Schemes : A Review, *IEEE Communications Magazine*, vol.34, 82-91.
- [10] **Guerin, R., Ahmadi, H., and Naghshineh, M.**, 1991. Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks, *IEEE Journal on Selected Areas in Communications*, vol.9, 968-981.
- [11] **Elwalid, A. I. and Mitra, D.**, 1993. Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High Speed Networks, *IEEE/ACM Transactions on Networking*, vol.1, 329-343.
- [12] **Rege, K. M.**, 1994. Equivalent Bandwidth and Related Admission Criteria for ATM Systems- A Performance Study, *International Journal of Communication Systems*, vol.7, 181-197.

- [13] **Elwalid, A. I.**, 1995. Fundamental Bounds and Approximations for ATM Multiplexers with Applications to Video Teleconferencing, *IEEE Journal on Selected Areas in Communications*, vol.13, 1004-1016.
- [14] **Chang, S. and Thomas, J. A.**, 1995. Effective Bandwidth in High-Speed Networks, *IEEE Journal on Selected Areas in Communications*, vol.13, 1091-1100.
- [15] **Veciana, G., Kesidis, G., and Walrand, J.**, 1995. Resource Management in Wide Area ATM Networks Using Effective Bandwidth, *IEEE Journal on Selected Areas in Communications*, vol.13, 1081-1090.
- [16] **Duffield, N. G.**, 1995. Entropy of ATM Traffic Streams: A Tool for Estimating QoS Parameters, *IEEE Journal on Selected Areas in Communications*, vol.13, 981-990.
- [17] **Saito, H.**, 1992. Call Admission Control in an ATM Network Using UpperBound of Cell Loss Probability, *IEEE Transactions on Communications*, vol.40, 1512-1521.
- [18] **Rasmussen, C.** 1991. Source-Independent Call Acceptance Procedures in ATM Networks, *IEEE Journal on Selected Areas in Communications*, vol.9, 351-358.
- [19] **Elwalid, A., Mitra, D. and Wentworth, R.H.**, 1995. A New Approach for Allocating Buffers and Bandwidth to Heterogeneous Regulated Traffic in an ATM Node, *IEEE Journal on Selected Areas in Communications*, vol.13, 1115-1127.
- [20] **Doshi, B. T. and Heffes, H.**, 1991. Performance of an In-Call-Buffer-Window Reservation/Allocation Scheme for Long File Transfers, *IEEE Journal on Selected Areas in Communications*, vol.9, 1013-1023.
- [21] **Golestani, S. J.**, 1991. Congestion-free Communication in Broadband Packet Networks, *IEEE Transactions on Communications*, vol.39, 1802-1812.
- [22] **Anick, D., Mitra, D. and Sondhi, M. M.**, 1982. Stochastic Theory of A Data Handling System with Multiple Sources, *Bell System Technical Journals*, vol.61, 1871-1894.
- [23] **Kosten, L.**, 1986. Liquid Models for A Type of Information Storage Problems, *Mathematic and Information Engineering*, vol.11, 71-86.
- [24] **Önal, H.**, 1998. Olasılık Kuramı ve Raslantı Süreçleri, İ.T.Ü. Elektrik-Elektronik Fakültesi Matbaası, İstanbul.
- [25] **Brey, Barry B.**, 1994. The Intel Microprocessors 8086/8088, 80186, 80286, and 80486, Macmillan Publishing Company, New York.

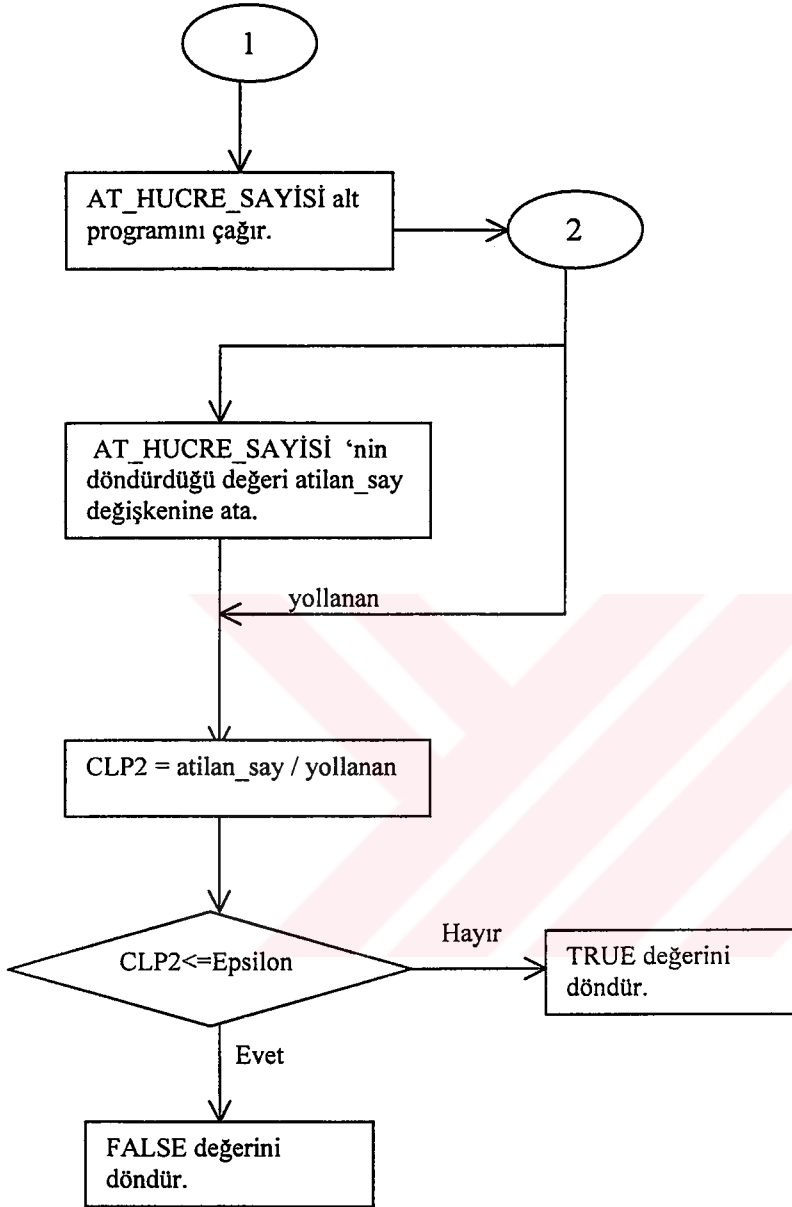
EK A : Kaynakların Aynı Anda İletime Başladığı ve ON-OFF Sürelerinin Sabit Olduğu Durum İçin Yazılan Benzetim Programının Akış Diyagramı

ANA PROGRAM :

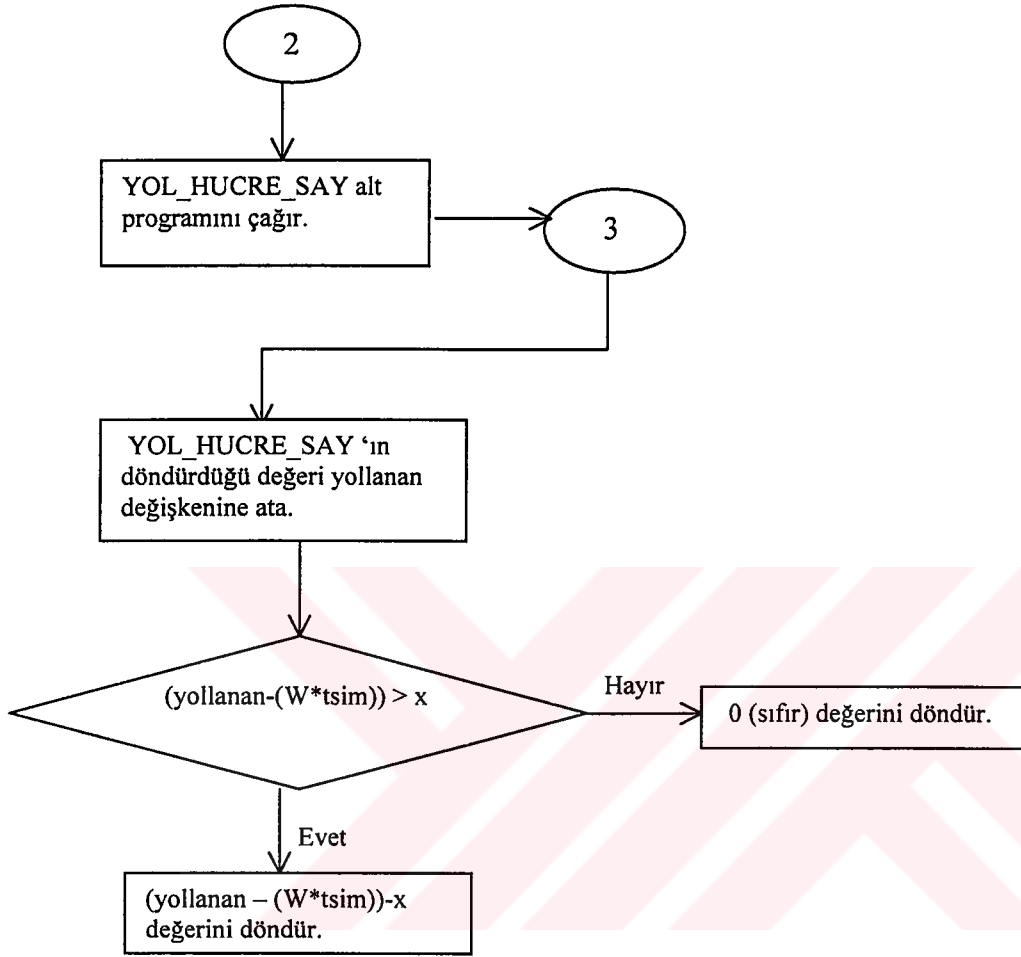




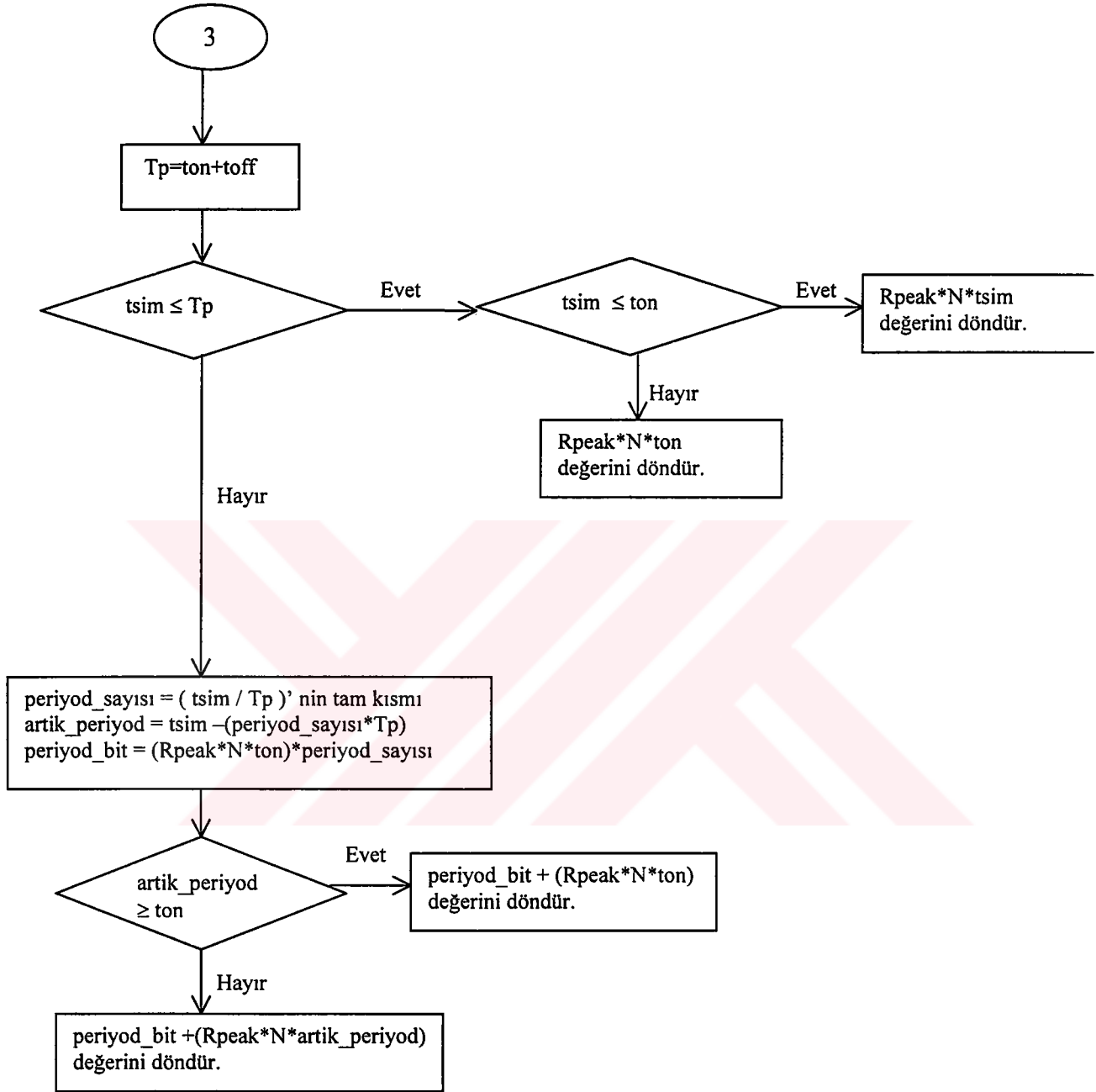
CELL_LOSS ALT PROGRAMI:



AT_HUCRE_SAYISY ALT PROGRAMI :

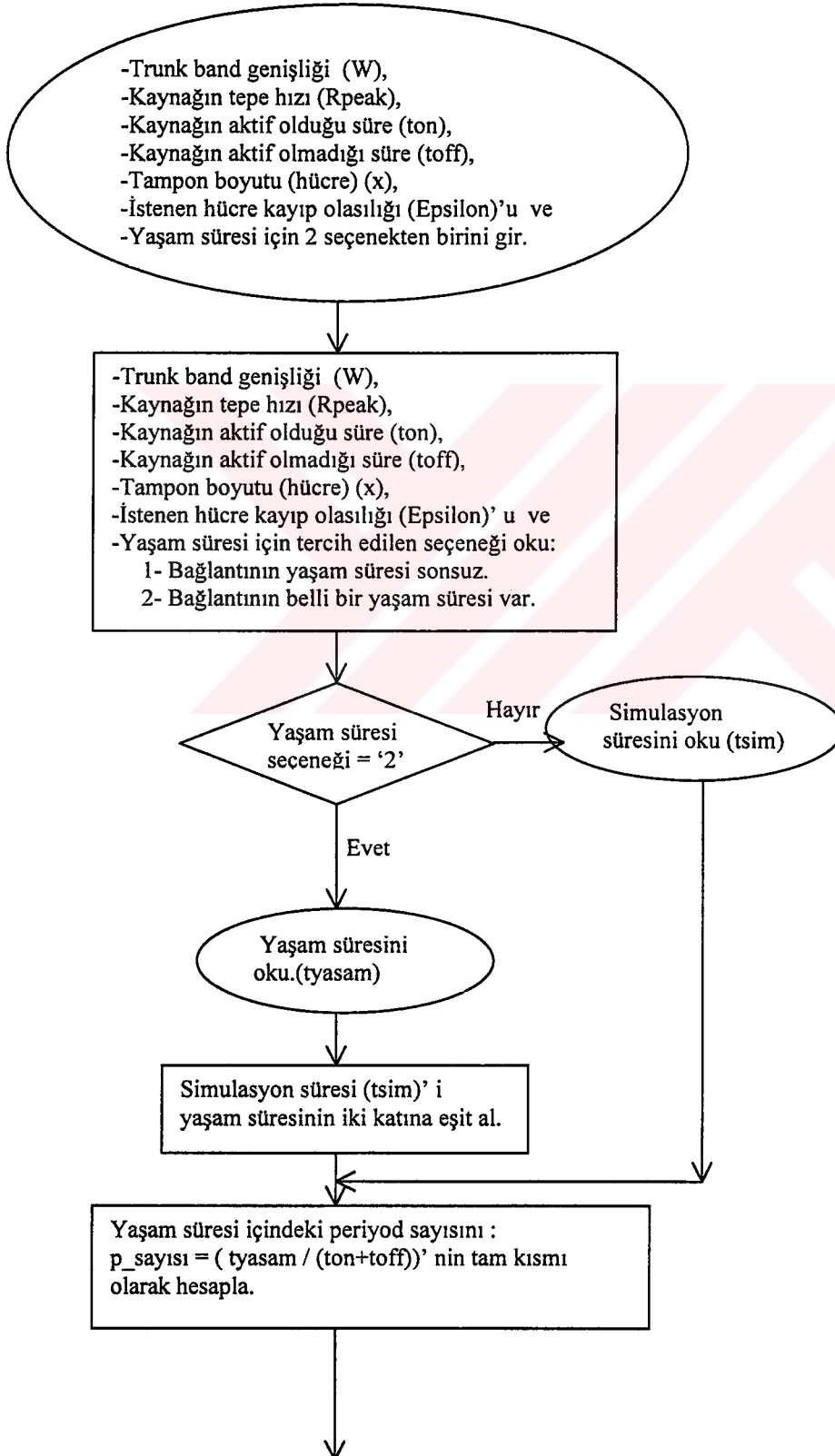


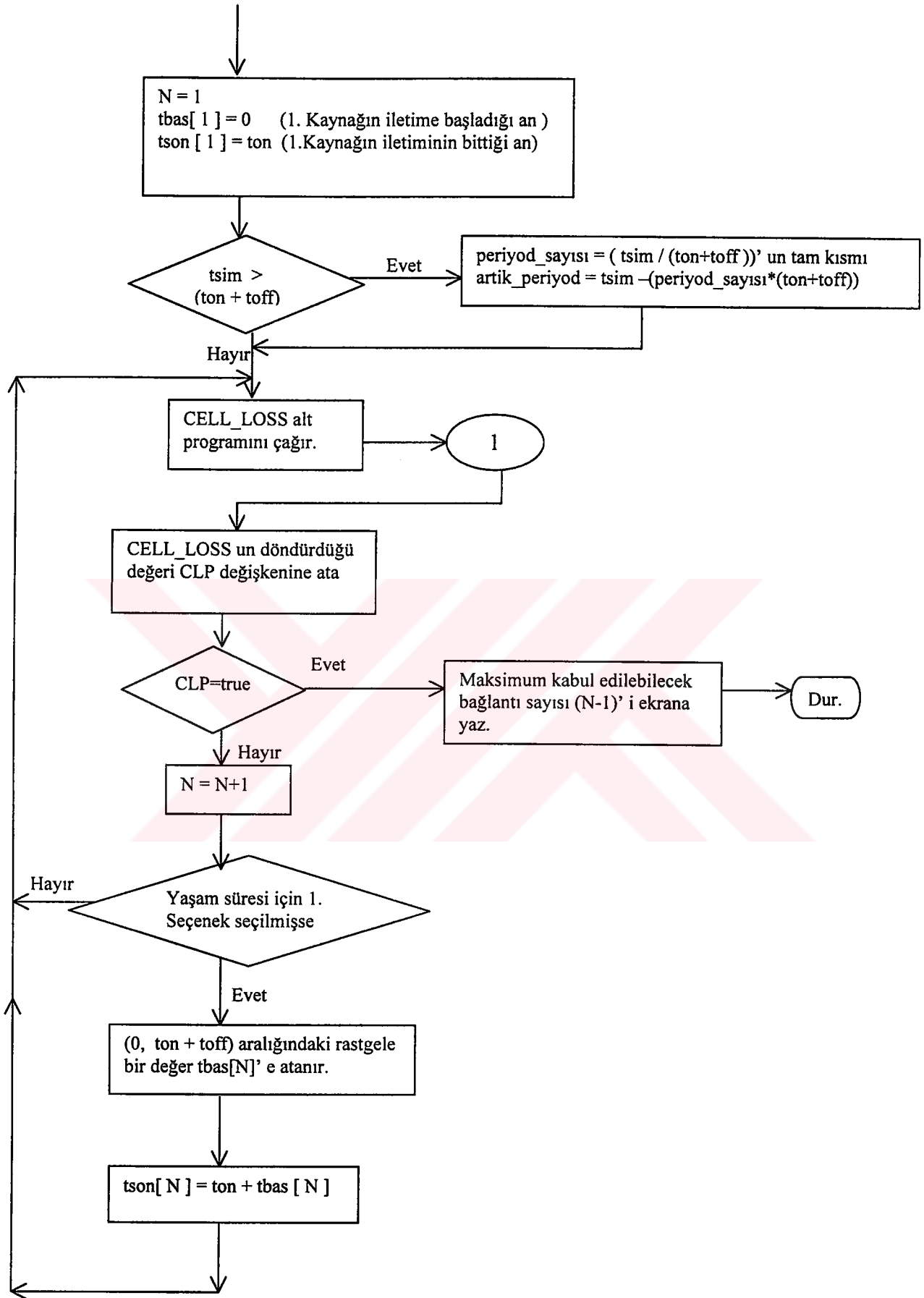
YOL_HUCRE_SAY ALT PROGRAMI :



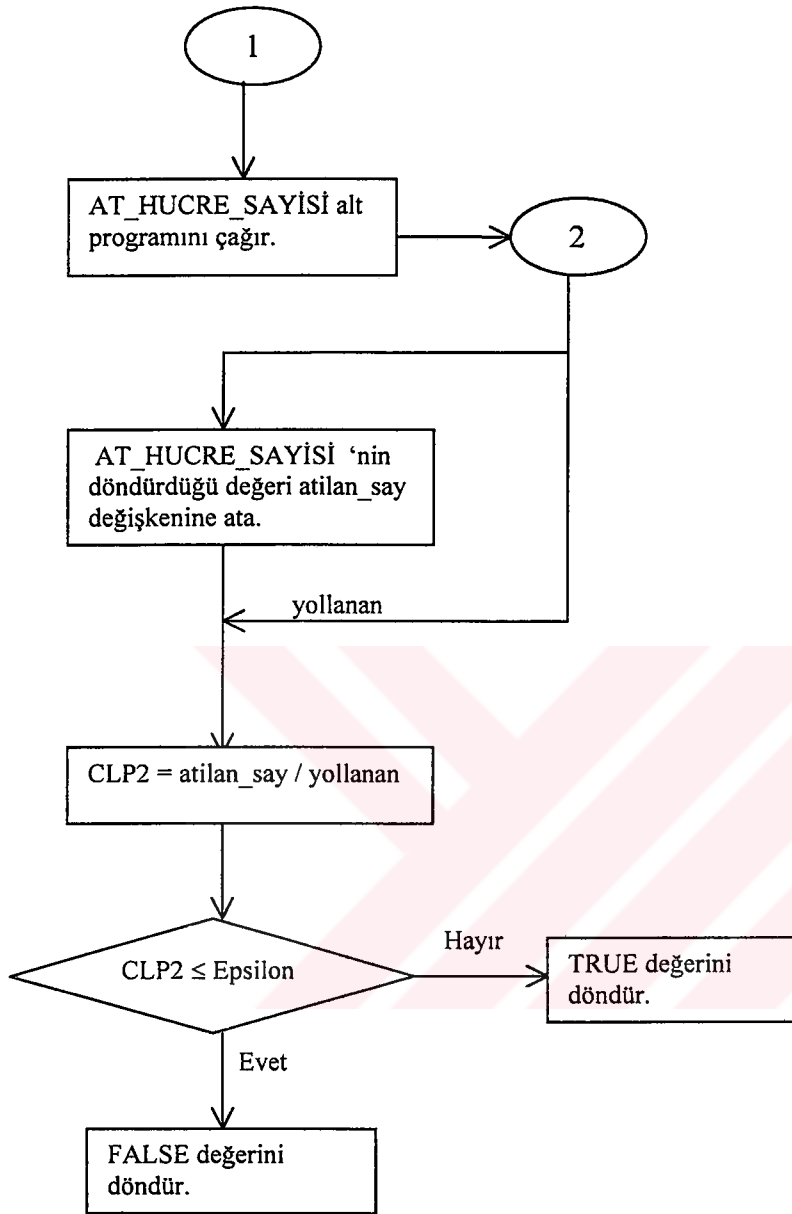
EK B : Kaynakların Belli Bir Aralıkta Rastgele İletime Başladığı ve ON-OFF Sürelerinin Sabit Olduğu Durum İçin Yazılan Benzetim Programının Akış Diyagramı

ANA PROGRAM :

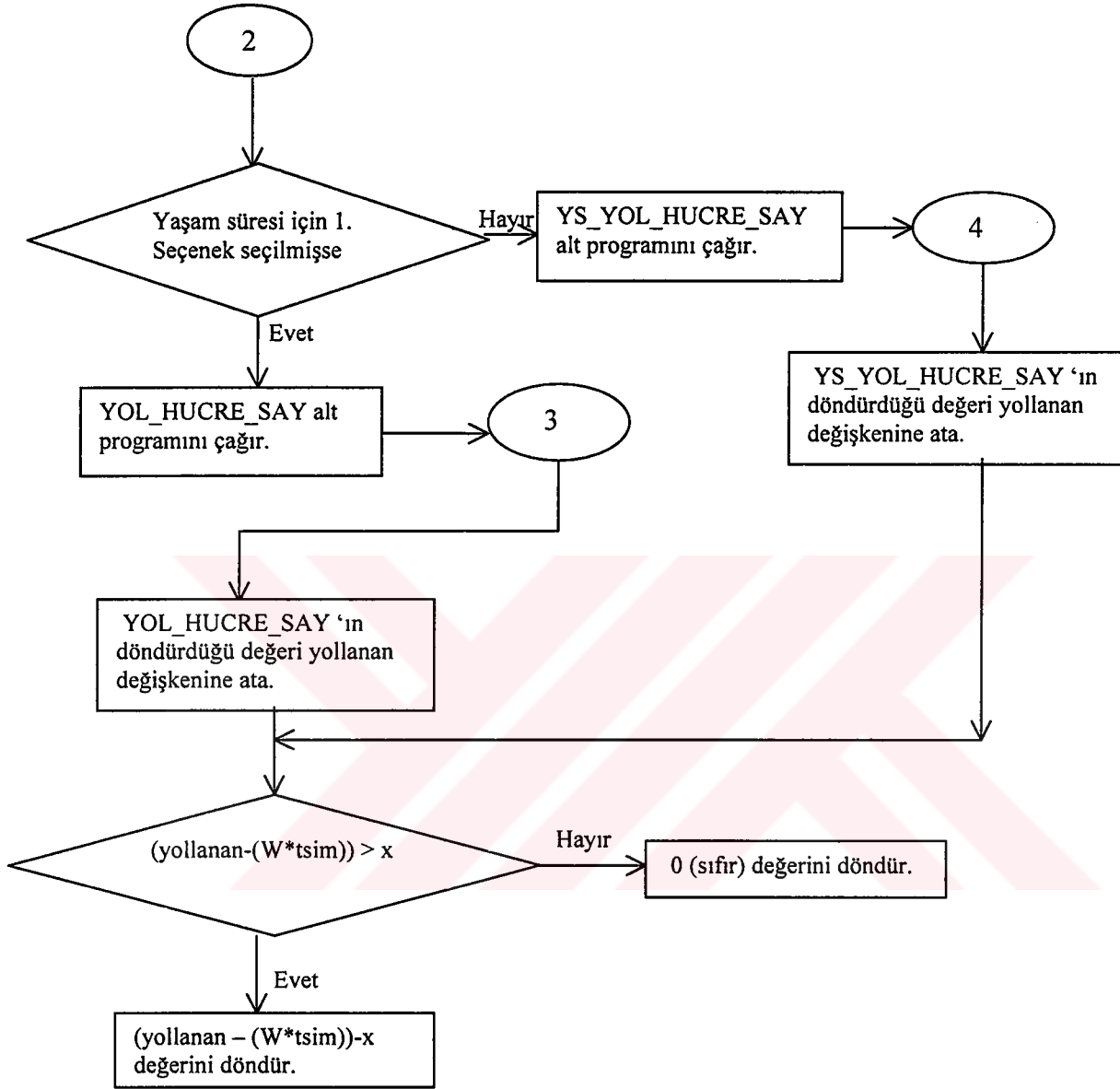




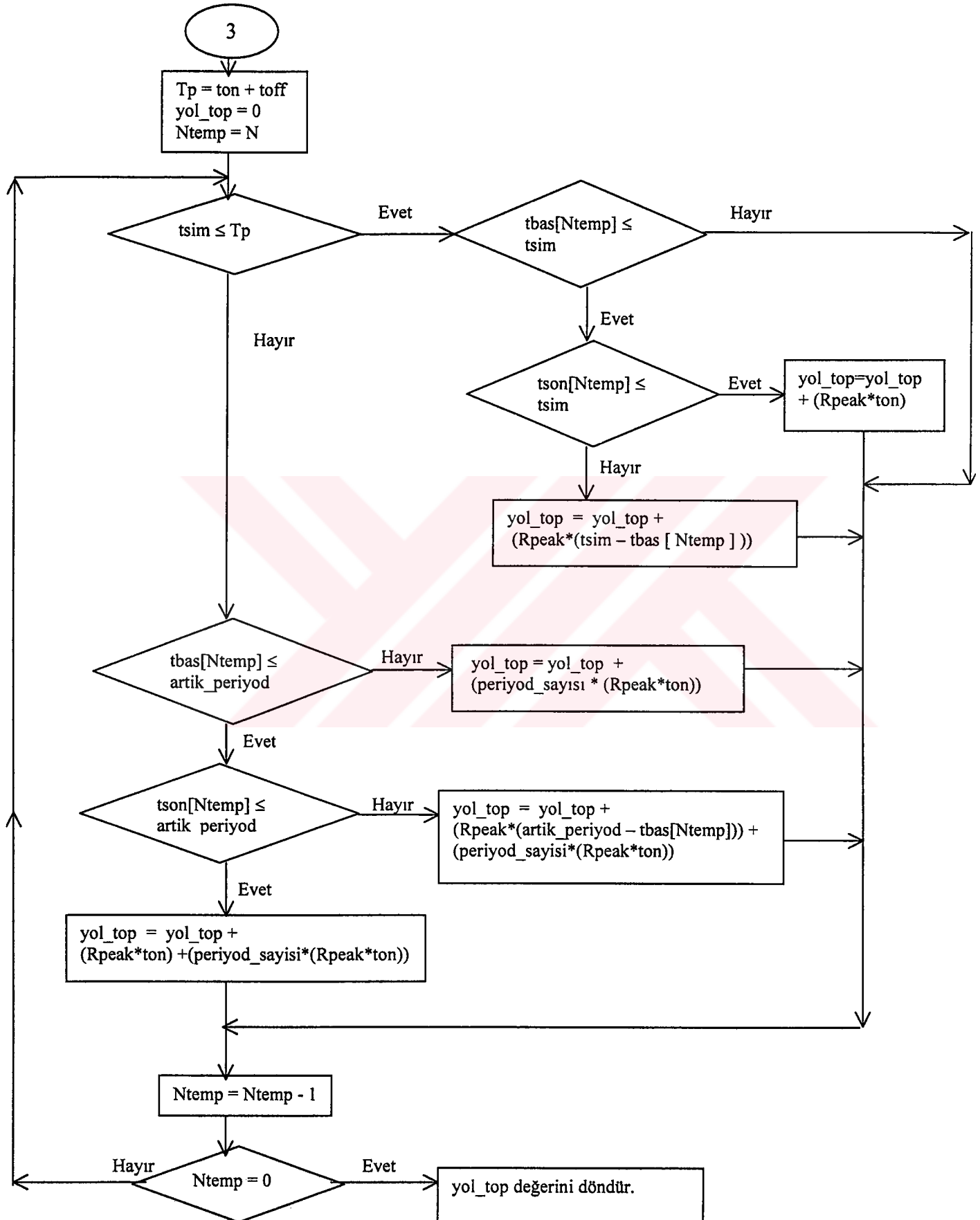
CELL_LOSS ALT PROGRAMI:



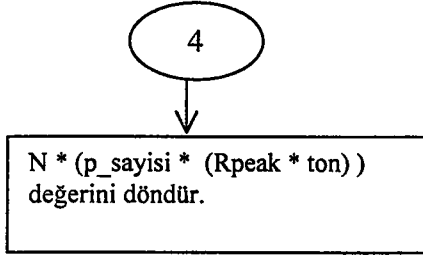
AT_HUCRE_SAYI ALT PROGRAMI :



YOL_HUCRE_SAY ALT PROGRAMI :

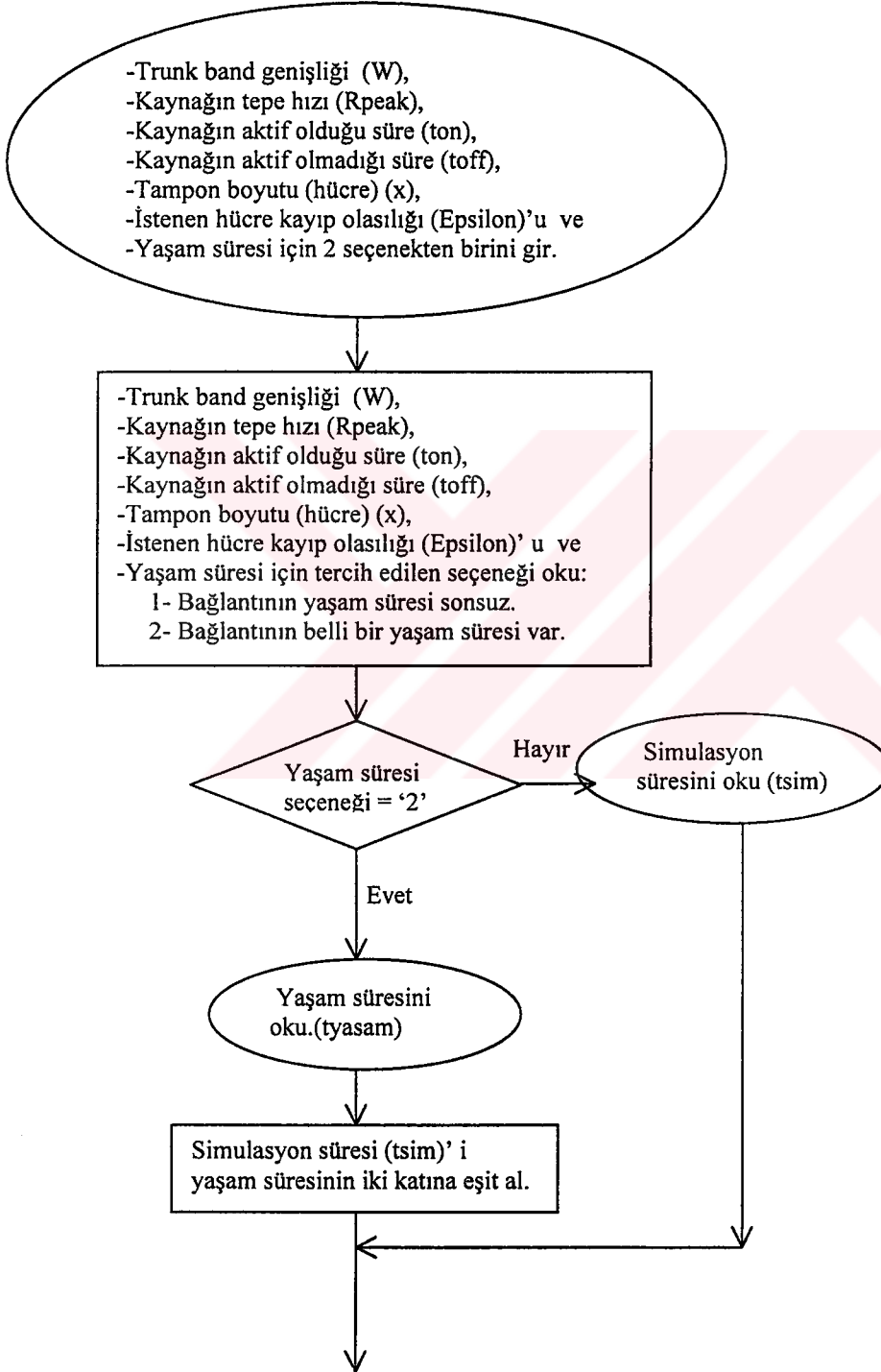


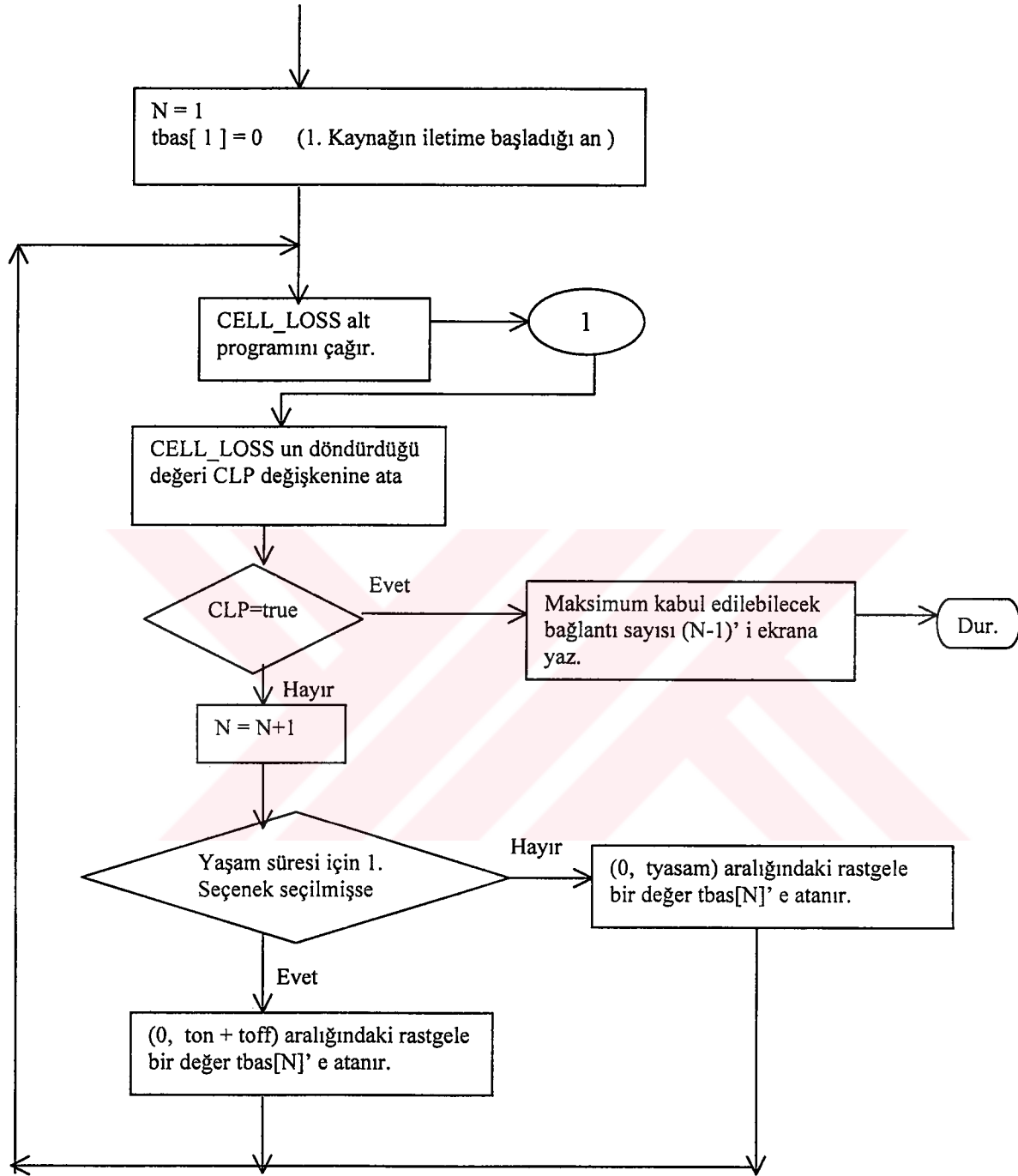
YS_YOL_HUCRE_SAY ALT PROGRAMI :



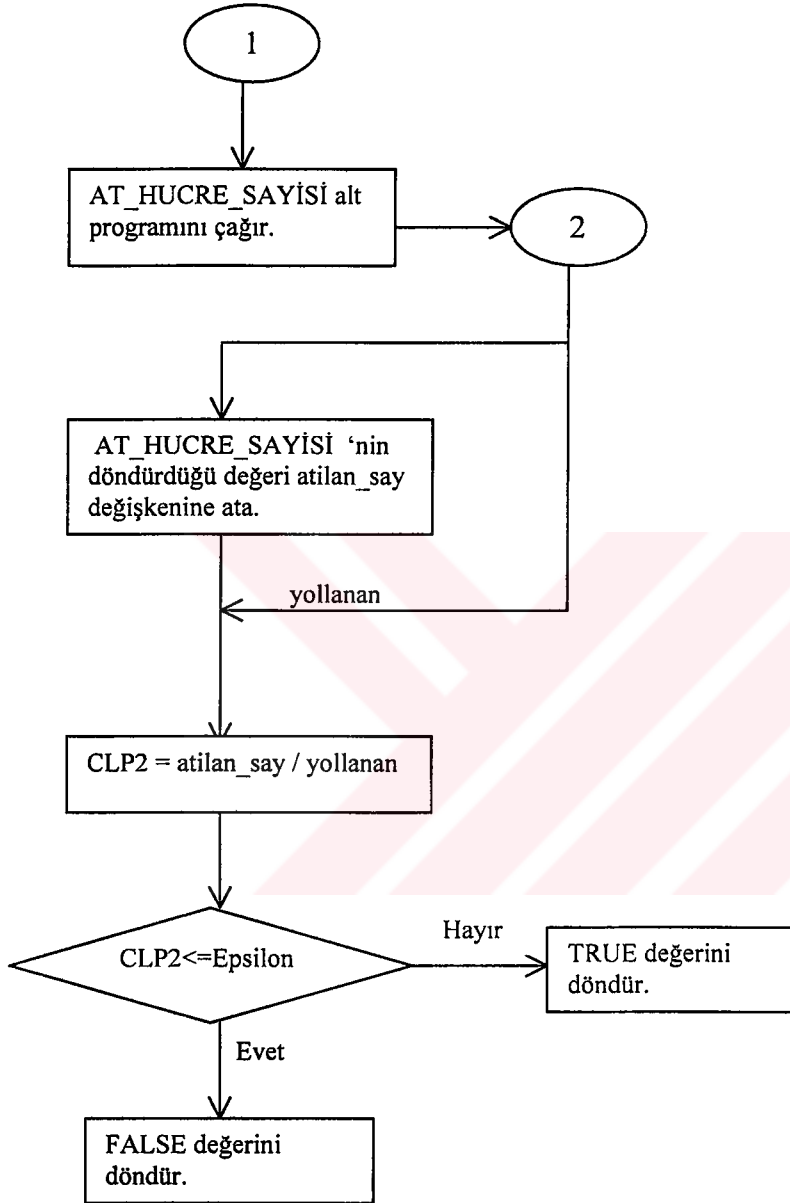
EK C : Kaynakların Belli Bir Aralıkta Rastgele İletime Başladığı ve ON-OFF Sürelerinin Üssel Dağılıma Göre Rastgele Değiştiği Durum İçin Yazılan Benzetim Programının Akış Diyagramı

ANA PROGRAM :

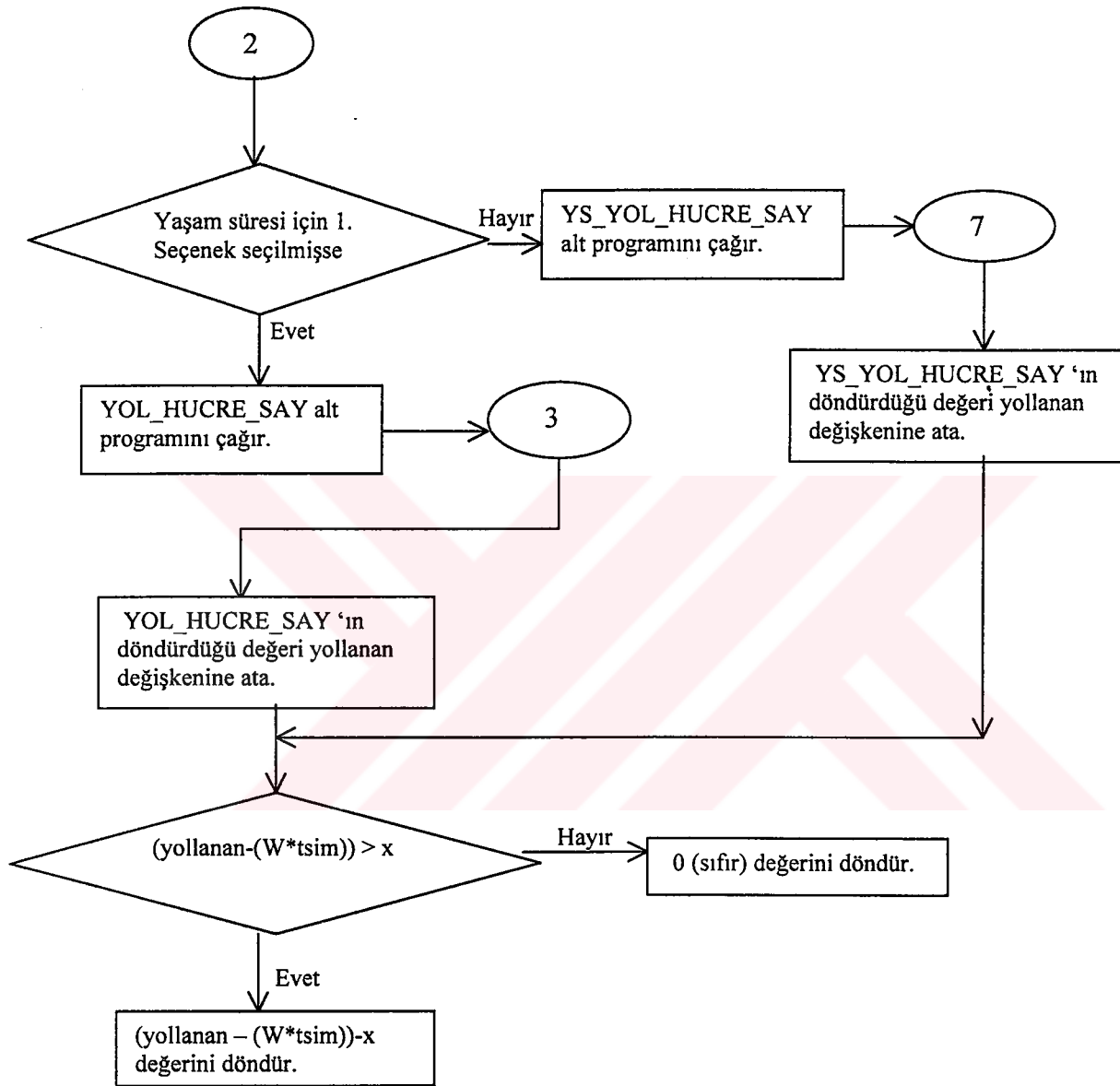




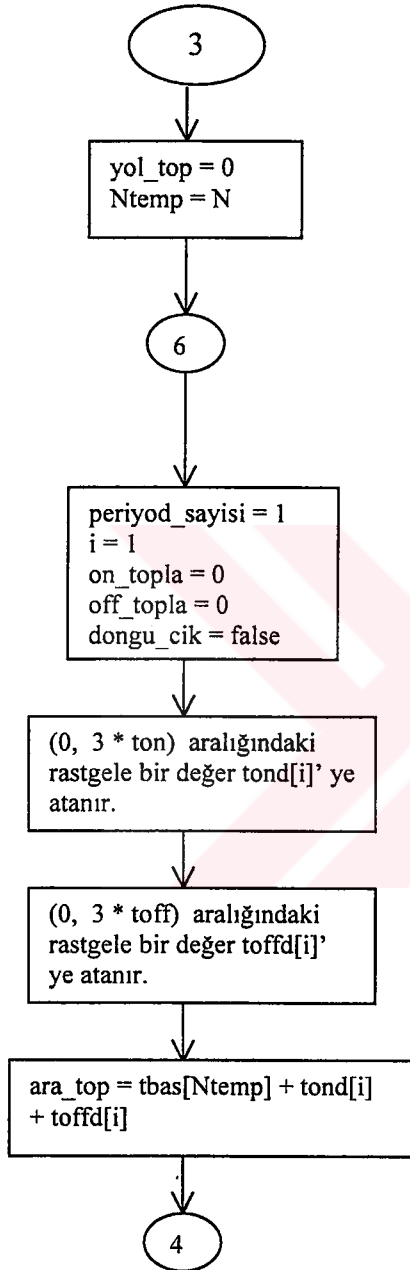
CELL_LOSS ALT PROGRAMI:

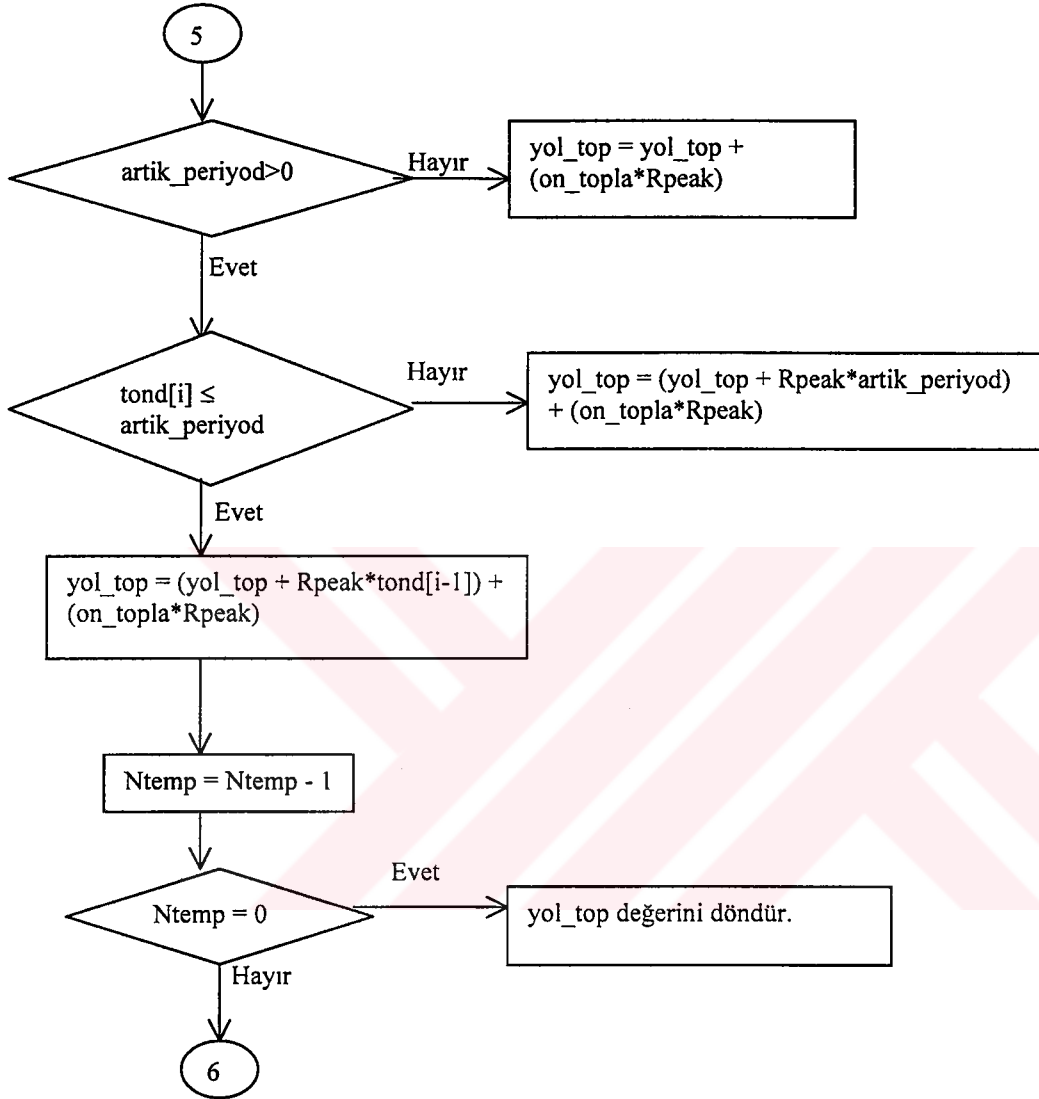


AT_HUCRE_SAYISI ALT PROGRAMI :

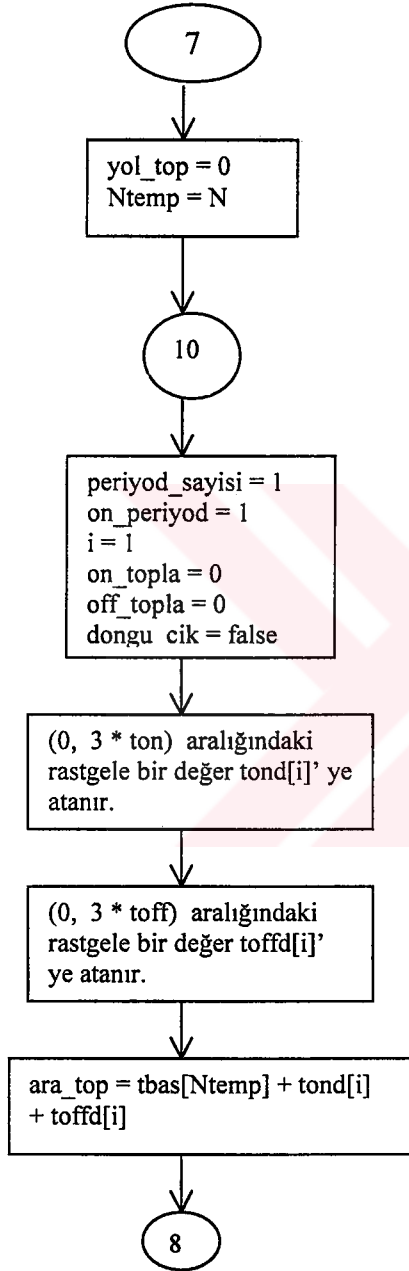


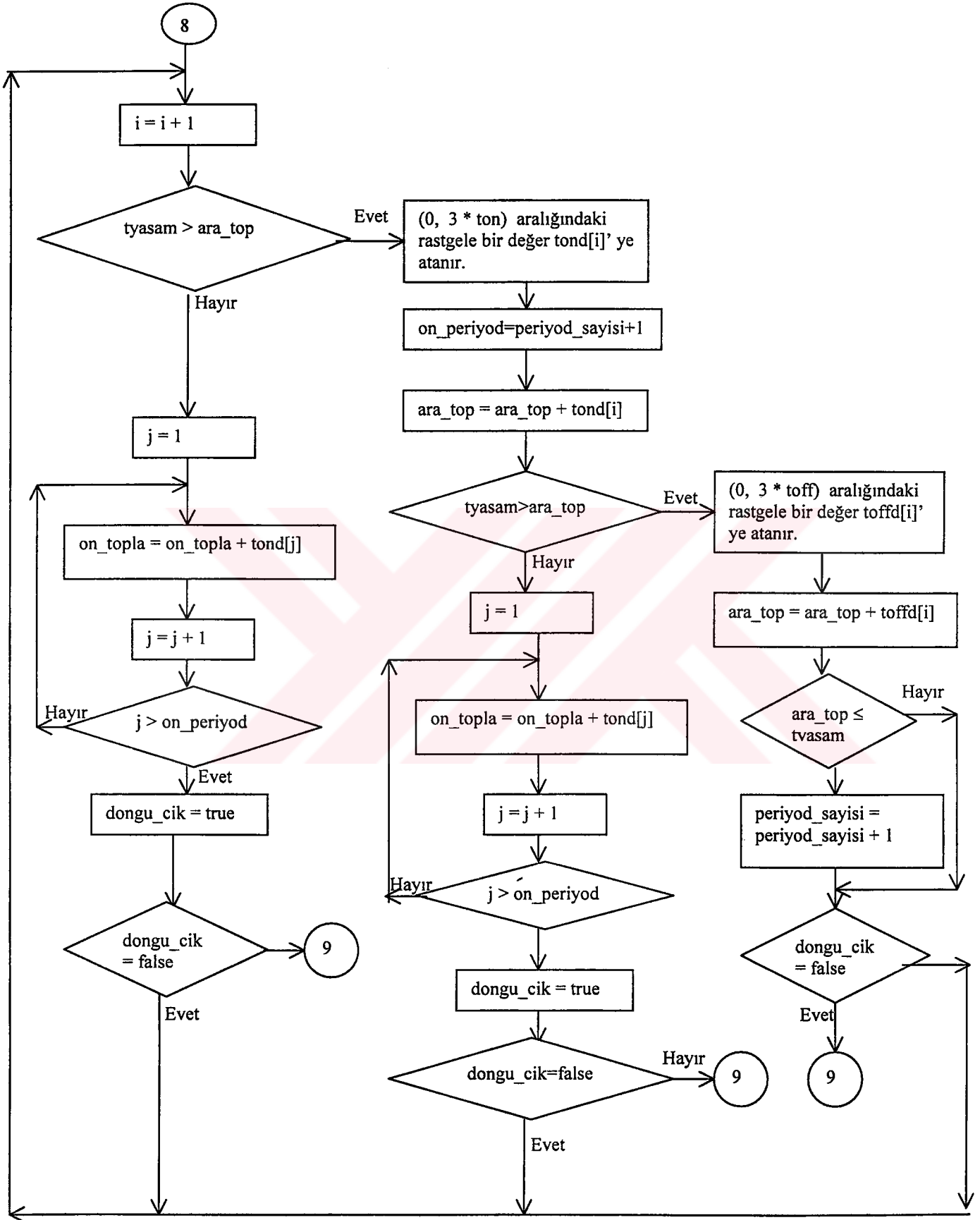
YOL_HUCRE_SAY ALT PROGRAMI :





YS_YOL_HUCRE_SAY ALT PROGRAMI :





Özgeçmiş

Meltem GÖKKAYA 1975 yılında Çanakkale’ de doğdu. 1992 yılında Çanakkale Lisesi’ nden ve 1997 yılında İstanbul Teknik Üniversitesi Kontrol ve Bilgisayar Mühendisliği Bölümünden mezun oldu. 1997 yılında İstanbul Teknik Üniversitesi Kontrol ve Bilgisayar mühendisliği dalında yüksek lisansa başladı. Halen Netaş’ ta Yazılım Tasarımcısı olarak çalışmaktadır.

