

**MOBİL KABLOSUZ AĞLARDA VERİ KAYIPLARININ SİMULASYON  
ORTAMINDA FARKEDİLEBİLİR HALE GETİRİLMESİ**

**UFUK TANYERİ**

**YÜKSEK LİSANS TEZİ  
BİLGİSAYAR EĞİTİMİ**

**GAZİ ÜNİVERSİTESİ  
BİLİŞİM ENSTİTÜSÜ**

**Nisan 2009  
ANKARA**

Ufuk TANYERİ tarafından hazırlanan **MOBİL KABLOSUZ AĞLARDA VERİ KAYIPLARININ SİMULASYON ORTAMINDA FARKEDİLEBİLİR HALE GETİRİLMESİ** adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Doç. Dr. Halil İbrahim BÜLBÜL

Tez Yöneticisi

Bu çalışma, jürimiz tarafından Bilgisayar Eğitimi Anabilim Dalında yüksek lisans tezi olarak oy birliği ile kabul edilmiştir.

Başkan : Prof. Dr. Ali Paşa AYDIN

Üye : Prof. Dr. Şeref SAĞIROĞLU

Üye : Doç. Dr. Halil İbrahim BÜLBÜL

Tarih : 20/04/2009

Bu tez, Gazi Üniversitesi Bilişim Enstitüsü tez yazım kurallarına uygundur.

## **TEZ BİLDİRİMİ**

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Ufuk TANYERİ

# MOBİL KABLOSUZ AĞLARDA VERİ KAYIPLARININ SİMULASYON ORTAMINDA FARKEDİLEBİLİR HALE GETİRİLMESİ

(Yüksek Lisans Tezi)

Ufuk TANYERİ

GAZİ ÜNİVERSİTESİ

BİLİŞİM ENSTİTÜSÜ

Nisan 2009

## ÖZET

Bu çalışmada, mobil plansız ağlarda (Mobile Ad-Hoc Network - MANET) paket iletimi esnasındaki veri kaybına neden olan çarpışmaların nedeni araştırılmış ve bu çarpışmaların gerçek ortama uyarlama amacıyla simülasyonu yapılırken tam olarak dikkate alınmadığı tespit edilmiş ve bu durum simülasyon ortamında yapılan değişikliklerle gerçek duruma uygun hale getirilmiştir. Bunun için oluşturulan topoloji, mobil bir düğüm, yönlendirme mekanizması ve ağ inşa etmek için gerekli olan ağ bileşenlerinden oluşmaktadır. Bu bileşenler özetle kanal, ağ arabirimi, radyo yayılım modeli, MAC iletişim kuralı, kuyruk arabirimi, bağ katmanı ve adres çözümleme iletişim kuralını kapsamaktadır. Bu modelin oluşturulması için ağ simülatör aracı NS2'den yararlanılmış ve kablosuz ağlardaki karışıklığa neden olduğu bilinen gizli düğüm (hidden node), örtülü düğüm (masked node) ve düğümler arasında ortaya çıkan birbirini engelleme (blocking) etkisi problemleri simülatör üzerinde irdelenmiştir. Sonuç olarak, simülatörde, bir düğümün veri iletiminde kullandığı eşik değerler; gönderme isteği eşiği (MAC\_RTSThreshold), taşıyıcı algılama eşiği (CSThresh\_) ve alıcı eşiği (RXThresh\_) ile tespit edilemeyen veri paketi çarpışması sorunu, yapılan değişikliklerden sonra elde edilen parametreler ile fark edilebilir hale getirilmiştir.

**Bilim Kodu : 702.3.006**

**Anahtar Kelimeler : Gizli Dügüm, Örtülü Dügüm, Bloklama Etkisi,  
IEEE802.11, MANET**

**Sayfa Adedi : 78**

**Tez Yöneticisi : Doç. Dr. Halil İbrahim BÜLBÜL**

**AWARE OF DATA LOSSES ON A MANET WIRELESS NETWORK BY  
SIMULATION**

**(M.Sc. Thesis)**

**Ufuk TANYERİ**

**GAZİ UNIVERSITY  
INFORMATICS INSTITUTE**

**April 2009**

**ABSTRACT**

**In this thesis, some packet collisions that are encountered in the realistic scenarios and result in packet losses investigated in mobile ad-hoc network (MANET). Our investigations reveal that these collisions cannot recognized by the network simulation tools for some specific realistic cases. In this thesis, we modify the well-known network simulator NS2 to allow the mobil ad hoc nodes to recognize these collisions to make NS2 more realistic. The topology which is created for solving this issue cover the internals of a mobilenode, routing mechanisms and network components that are used to construct the network stack for a mobile node. The communication paremeters considered in this study are Channel and Network interface, Radio propagation model, MAC protocols, Interface Queue, Link layer and Address resolution protocol model (ARP). NS2, which is a network simulator is used to create this model, and problems which is known as hidden node occuring a collisions over wireless network, masked node, and blocking effect among nodes are studied using simulator. As a result, the problems of data packet collisions is unable to be captured with threshold values, which is used in a data transmission by node, called as Request-to-send Threshold (MAC\_RTSThreshold), Carrier Sensing**

**Thresh (CSThresh\_), and Receiving Thresh (RXThresh\_) are able to be captured with parameters is obtained after changes.**

**Science Code : 702.3.006**

**Key Words : Hidden node, Masked Node, Blocking Effect, IEEE 802.11,  
MANET**

**Page Number: 78**

**Adviser : Assoc. Prof. Dr. Halil İbrahim BÜLBÜL**

## TEŐEKKÜR

Çalıőmalarım boyunca deęerli yardım ve katkılarıyla beni yönlendiren Danıőmanım Doç. Dr. Halil İbrahim BÜLBÜL'e, Prof. Dr. Ali Paőa AYDIN'a, Prof. Dr. őeref SAĐIROĐLU'na, Doç. Dr. M. Ali AKÇAYOL'a, Doç. Dr. O. Ayhan ERDEM'e, Öğr. Gör. Dr. Sabri ERDEM'e ve aileme teőekkürü borç bilirim.



## İÇİNDEKİLER

	<b>Sayfa</b>
ÖZET.....	iv
ABSTRACT.....	vi
TEŞEKKÜR.....	viii
İÇİNDEKİLER .....	ix
ŞEKİLLERİN LİSTESİ .....	xi
SİMGELER VE KISALTMALAR.....	xiii
1. GİRİŞ.....	1
2. MEVCUT ÇALIŞMALAR.....	3
3. MOBİL KABLOSUZ AĞLAR VE AĞ SİMÜLATÖRLERİ .....	6
3.1. Mobil Kablosuz Ağlar .....	6
3.1.1. Gizli Düğüm Problemi.....	7
3.1.2. RTS-CTS Mekanizması.....	10
3.1.3. Örtülü Düğüm Problemi .....	11
3.1.4. Örtülü Düğüm ve Gizli Düğüm Problemlerinin Simülatör Ortamında Karşılaştırması.....	14
3.1.5. Bloklama Etkisi Problemi.....	15
3.2. Ağ Simülatörleri .....	16
3.2.1. NS2 (Network Simulator).....	16
3.2.2. OPNET.....	24
3.2.3. QualNet.....	25
3.2.4. Ptolemy .....	25
3.2.5. LZ.....	25
3.2.6. SimEleven.....	25

3.2.7. REAL .....	26
3.2.8. WARP2 .....	26
3.2.9. GPSS .....	26
4. MATERYAL VE METOD .....	27
4.1. Fedora İşletim Sisteminin Tercih Edilmesi ile İlgili Araştırma .....	28
5. MOBİL KABLOSUZ AĞLARDA VERİ PAKETİ KAYIPLARINA YÖNELİK UYARLANAN SİMÜLASYONLAR .....	33
5.1. Gizli Düğüm Simülasyonu .....	33
5.2. NS’de RTS-CTS Mekanizması Simülasyonu .....	38
5.3. Örtülü Düğüm Simülasyonu .....	38
5.4. Bloklama Simülasyonu .....	39
5.5. Yeniden Yapılandırılan Kablosuz Ağ Modeli .....	42
5.5.1. Define-option.tcl .....	42
5.5.2. Confige-node.tcl .....	44
5.5.3. Ns-default.tcl’deki Düzenlemeler .....	45
5.5.4. Mac-802.11.h’deki Düzenlemeler .....	47
5.5.5. WirelessLan.tcl .....	47
5.6. TCP ve UDP Üstlencilerde Karşılaştırmalı Model .....	61
6. SONUÇ ve ÖNERİLER .....	68
6.1. Sonuçlar .....	68
6.2. Öneriler .....	71
KAYNAKLAR .....	72
ÖZGEÇMİŞ .....	77

## ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 3.1. Gizli düğüm problemi .....	8
Şekil 3.2. Fiziksel taşıyıcı algılama aralığının (Rcs) girişimi nasıl etkilediği.....	9
Şekil 3.3. Varsayılan değerlerle NS de 3 düğümün iletim aralıkları.....	9
Şekil 3.4. IEEE 802.11 MAC ile RTS-CTS tokalaşma.....	10
Şekil 3.5. Örtülü düğüm problemi .....	11
Şekil 3.6. Örtülü düğümün neden olduğu veri paket kaybı senaryosu.....	13
Şekil 3.7. Alındı damgasından kaynaklanan veri paket kaybı senaryosu.....	14
Şekil 3.8. Gizli ve örtülü düğüm için yüke karşılık gelen paket hata oranı .....	15
Şekil 3.9. Örtülü düğüm .....	16
Şekil 3.10. NS Yapısının kullanıcı görünümü.....	18
Şekil 3.11. NAM düzenleyici penceresi .....	20
Şekil 3.12. NAM simülasyon penceresi .....	20
Şekil 3.13. C++ ve OTel ikilisi.....	22
Şekil 3.14. NS Mimarisi .....	22
Şekil 3.15. Bir grafikte 3 izleme dosyasının XGraph'da karşılaştırılması .....	23
Şekil 4.1. Mesaj içeriğine göre işletim sistemlerinin posta listesinde geçme sıklığı .....	29
Şekil 4.2. Konu başlığına göre işletim sistemlerinin posta listesinde geçme sıklığı.....	29
Şekil 5.1. İzleme dosyasından alınan değerler .....	36
Şekil 5.2. NS varsayılan CStresh_ ve RXThresh_ ve RTS-CTS tokalaşmasız ...	37
Şekil 5.3. CStresh_ = RXThresh_ ve RTS-CTS tokalaşma var.....	37
Şekil 5.4. Dört düğümlü senaryonun izleme dosyasından alınan değerler.....	38
Şekil 5.5. Dört düğümlü senaryolu örtülü düğüm simülasyonundaki izleme dosyası değerleri.....	39

Şekil 5.6. Bloklama etkisi.....	39
Şekil 5.7. Bloklama etkisinin grafiksel gösterimi .....	40
Şekil 5.8. Beş düğüm için hazırlanan simülasyonun bant genişliği / yük sonuçları	41
Şekil 5.9. On düğüm için hazırlanan simülasyonun bant genişliği / yük sonuçları.	41
Şekil 5.10. MANET destekli bir ağ örneği.....	60
Şekil 5.11. CSTHresh_ = RXThresh_, RTSTHreshold_ 0 ile TCP Simülasyonu .....	62
Şekil 5.12. CSTHresh_ = RXThresh_, RTSTHreshold_ 3000 ile TCP Simülasyonu	63
Şekil 5.13. NS2'nin varsayılan durumu ile TCP Simülasyonu.....	64
Şekil 5.14. CSTHresh_ = RXThresh_, RTSTHreshold_ 0 ile UDP Simülasyonu .....	65
Şekil 5.15. CSTHresh_ = RXThresh_, RTSTHreshold_ 3000 ile UDP Simülasyonu	66
Şekil 5.16. NS2'nin varsayılan durumu ile UDP Simülasyonu .....	67

## SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

<b>Simgeler</b>	<b>Açıklama</b>
d	Mesafe
D	Düşen Paket
dBm	1 Milivat'a Karşılık Gelen Desibel Değeri
Gr	Verici Anten Kazancı
Gt	Alıcı Anten Kazancı
Hr	Verici Anten Yüksekliği
Ht	Gönderici Anten Yüksekliği
L	Sistem Kaybı
MHz	Megahertz. Bir Milyon Hertz
Pr	Alınan Sinyal Gücü
Pt	Gönderilen Sinyal Gücü
r	Alım İşlemi
Rtx	Göndericinin Alım Aralığının Çapı
Rcs	Taşıyıcı Algılama Aralığının Çapı
s	Gönderme İşlemi
$\rho$	Ağ İş Yüğü
<b>Kısaltmalar</b>	<b>Açıklama</b>
AODV	Ad hoc On Demand Distance Vector (Plansız İsteğe Bağlı Uzaklık Vektörü)
ARP	Address Resolution Protocol (Adres Çözümleme İletişim Kuralı)
CBK	DROP_RTR_MAC_CALLBACK
CBR	Constant Bitrate (Sabit Büyüklükte Kullanılan Trafik Deseni)
CSMA	Carrier Sense Multiple Access (Taşıyıcı Dinleyen Çoklu Erişim)
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance (Taşıyıcı Dinleyen Çoklu Erişim / Çarpışmadan Kaçınma)
CSThresh	Carrier Sensing Thresh (Taşıyıcı Algılama Aralığı)
CTS	Clear-to-send (Gönderime Açık)
DARPA	Defense Advanced Research Projects Agency (Savunma Bakanlığı İleri Araştırma Projeleri Ajansı)
DSDV	Destination-Sequenced Distance Vector routing (Hedef Sıralı Uzaklık Vektörü Yönlendirme İletişim Kuralı)
DSSS	Direct Sequence Spread Spectrum (Doğrudan Diziyle Yayma)

FTP	File Transfer Protocol (Dosya Transfer İletişim Kuralı)
GPSS	General Purpose Simulation System (Genel Amaçlı Simülasyon Sistemi)
IEEE	Institute of Electrical and Electronics Engineers (Elektrik ve Elektronik Mühendisliği Enstitüsü)
IEEE 802.11	Kablosuz Yerel Alan Ağ Standardı
IFQ	DROP_IFQ_QFULL
LBL	Lawrence Berkeley Laboratory (Lawrence Berkeley Laboratuvarı)
LLC	Logical Link Control (Mantıksal Bağ Kontrolü)
LOS	Line Of Sight (Görüş Hattı)
MAC	Media Access Control (Ortama Erişim Kontrolü)
MAC_RTSThreshold	MAC Request to Send Threshold (MAC Gönderme İsteği Eşiği)
MACAW	Multiple Access Collision Avoidance for Wireless (Kablosuz Çoklu Erişimli Çarpışmadan Kaçınma)
MANET	Mobile Ad-Hoc Network (Gezgin Plansız Ağ)
NAM	Network Animator (Ağ Animatörü)
NAV	Network Allocation Vector (Ağ Tahsis Vektörü)
NS	The Network Simulator (Ağ Simülasyonu)
NRTE	DROP_RTR_NO_ROUTE
OPNET	Optimized Network Evaluation Tool (En Uygun Ağ Değerlendirme Aracı)
PARC	Palo Alto Research Center (Palo Alto Araştırma Merkezi)
PHY	Physical (Fiziksel)
REAL	REAlistic And Large (Gerçekçi ve Büyük) Simülatör Aracı
RED	Random Early Detection (Rasgele Erken Algılama)
RET	DROP_MAC_RETRY_COUNT_EXCEEDED
RTS	Request-to-send (Gönderme İsteği)
RXThresh	Receiver Thresh (Alıcı Eşik)
TCL	Tool Command Language (Araç Komuta Dili)
TCP	Transmission Control Protocol (İletim Denetimi Kontrol İletişim Kuralı)
TORA	Temporally Ordered Routing Algorithm (Geçici Düzenlenmiş Yönlendirme Algoritması)
UCB	University of California, Berkeley (California Üniversitesi)
UDP	User Datagram Protocol (Kullanıcı Verikatarı İletişim Kuralı)
USC	University of South Carolina (Güney Carolina Üniversitesi)
VINT	The Virtual InterNetwork Testbed (Görsel Uluslar arası Ağ Ortamı)
WARP	Wireless Access Radio Protocol (Kablosuz Erişimli Radyo İletişim Kuralı)
WLAN	Wireless Local Area Network (Kablosuz Yerel Alan Ağı)

## 1. GİRİŞ

Mobil plansız ağ (Mobile Ad-Hoc Network), aralarında kablosuz bağlantı olan, düğümlerin (bağlantı noktaları, host) bağlı olduğu, ayarlanabilen mobil yönlendiricilerden (routers) oluşan kablosuz bir ağıdır. Bu tür ağların oluşturduğu topoloji değişken bir düzene sahiptir yani hareketlidir [1, 2].

Ağın mobil olmasından dolayı, yönlendirici ve düğümlerin oluşturduğu kablosuz topoloji her an hızlı ve beklenmedik bir şekilde değişebilmektedir. Diğer taraftan bu gibi ağlar kendi aralarında iletişim kurabildiği gibi daha geniş ağ olan Internet'e de bağlanabilirler.

Son yıllarda uluslar arası ve ulusal düzeyde yapılan çalışmalarda, dizüstü bilgisayarlarda olduğu gibi -60 dBm ile -80 dBm aralığında sinyal yayan 802.11/WiFi [3] standardını kullanan diğer mobil cihazların oluşturduğu kablosuz ağlarda da, farklı büyüklükte ve hızda veri desenlerini (CBR v.b.) taşıyarak ağ bant genişliğinin nasıl etkilendiği [1, 5, 11, 18, 19] bir düğümün bir diğerine en iyi hangi yoldan ulaşacağı [22, 23], düğümler arasında iletilen paketlerin ulaştırılma oranı ve bunların iyileştirilmesi [33, 36, 39], ağda yaşanan problemler [2, 8, 11, 16, 18, 20, 40] ve bunların simülatör ortamına ne kadar doğrulukla yansıtıldığı [9, 32] gibi konuların çokça irdelendiği görülmektedir.

Yukarıda bahsi geçen mobil plansız ağlar ile ilgili bu çalışmalarda, ağ iletişim kuralı (routing protocol, mac layer protocol v.b.) ve yönlendiriciler (routers), erişim noktası (AP), modem v.b. özel koşullar altındaki cihazların tasarımı ve değerlendirilmesi için LZ, QualNet, Ptolemy, NS, OPNET, SimEleven, REAL, WARP2, GPSS v.b. simülatörlerden yararlanılmaktadır [4-18]. Simülatörler iletişim kuralı ve cihaz uygulamalarının doğru analitik modellerinin inşa edilmesindeki maliyet, zaman v.b. zorluklardan dolayı tercih edilmektedirler. Çünkü simülasyonlar, değerlendirme ve analizin daha kolay ve ucuz maliyetle yapılmasını sağlarlar [19]. Ancak simülasyon modellerinde bile, ayrıntılı olarak sadece özel iletişim kurallarının tasarlandığı ve diğer katmanlardaki etkilerin yeterli bir şekilde hesaba katılmadığı durumlar da

olabilmektedir. Bu durum ise model kestirimlerinde önemli seviyede hatalar ortaya koyabilmektedir.

Bu çalışma plansız kablosuz ağlardaki bilinen gizli düğüm, örtülü düğüm ve düğümler arasında ortaya çıkan birbirini engelleme etkisi problemlerinden yola çıkılarak, gerçek ortamdaki değerler ile simülasyon ortamında kullanılan değerler arasındaki farklılıklara dikkat çekmeye dayanan bir çalışmadır [20]. Bahsi geçen bu problemlerin modellerinin simülatör ortamına doğru bir şekilde aktarılması gerçek ortamda yapılacak çalışmaların uygunluğu bakımından önemlidir. Bu uygunluğun sağlıklı bir şekilde yapılması için NS2 simülatöründe veri iletiminde önemli yer tutan taşıyıcı algılama eşiği (CSThresh), alıcı eşiği (RXThresh) ve gönderme isteği eşiği (MAC\_RTSThreshold) gibi bazı eşik değerler; değiştirilerek mobil plansız bir ağ tasarımı planlanmıştır.

Bu tasarım için oluşturulan mobil plansız ağ topolojisi, mobil bir düğüm, yönlendirme mekanizması ve ağ inşa etmek için gerekli olan ağ bileşenlerinden oluşmaktadır. Bu bileşenler Kanal, Ağ Arabirimi, Radyo Yayılım Modeli, Mac İletişim Kuralı, Kuyruk Arabirimi, Bağ Katmanı ve Adres Çözümleme İletişim Kuralı'nı kapsamaktadır [21].

Çalışmanın ikinci bölümünde kablosuz ağlarda yapılan mevcut yerli ve yabancı çalışmalara yer verilmiştir. Üçüncü bölümde ise mobil kablosuz ağlar, bu ağlardaki problemler ve RTS/CTS mekanizması ile ilgili modellerden bahsedilmiş, ayrıca mobil ağ tasarımının yapılabildiği ağ simülatörü olarak yaygın bir şekilde kullanılan NS2 [21] hakkında bilgi verilmiştir.

Çalışmanın dördüncü bölümünde mobil ağın tasarımı için gerekli materyal ve bu işlemlerin nasıl yapılacağını belirten metodoloji hakkında bilgi verilmiştir. Beşinci bölümde ise kablosuz ağdaki problemlerin ve RTS/CTS mekanizmasının NS2'de oluşturulan modelleri, oluşturulan mobil plansız ağ tasarımı ve bunlara ilişkin bulgular ortaya konmuştur. Çalışmanın son bölümünde ise elde edilen bulgulara ilişkin sonuç ve önerilere yer verilmiştir.



## 2. MEVCUT ÇALIŞMALAR

Plansız mobil ağlar ve bu ağlardaki gizli düğüm, örtülü düğüm ve bloklama problemlerinden kaynaklanan veri kaybına neden olan çarpışmalarla ilgili olarak Gazi Üniversitesi Kütüphanesi, Orta Doğu Teknik Üniversitesi Kütüphanesi, YÖK Dökümantasyon Merkezi, Ulakbim ve İnternet üzerinden yapılan araştırma sonucunda yurtiçi ve yurtdışında yapılmış çalışmalar incelenmiş ve bu çalışma ile ilgili yönleri özetlenmiştir.

2007 yılında Bay ve 2006 da Doğru tarafından yapılan iki ayrı çalışmada plansız ağlarda iletişim kuralı geliştirilmiş ve geliştirilen iletişim kuralları mevcut olanlar ile kıyaslanılmıştır. Bu çalışmalarda, NS2 kullanılarak enerji etkin yönlendirme iletişim kuralı ve hareketlilik yönetim iletişim kuralı geliştirmiştir. İlk çalışmada enerji etkin yönlendirme iletişim kuralı ile plansız ağlarda düğümlerin hareketliliğinden kaynaklanan sinyal zayıflaması ve bağlantı kopuklukları nedeniyle meydana gelen veri kaybını en aza indirmek planlanmıştır. İkincide ise geliştirilen diğer iletişim kuralı ile düğümlerin enerji seviyeleri kritik düzeye indiği zaman aktif hale gelerek yönlendirme için yeni yol oluşturma planlanmıştır [22-23].

Sözer, tarafından 2004 de yapılan çalışmada ise kablosuz plansız ağlar üzerinde eşler arası dosya paylaşım problemini çözen bir sistem önerilmiştir. Önerilen sistem, merkezi bir sunucuya ihtiyaç duymaksızın, eşler arası sistemlerin prensiplerine göre çalışmakta ve paylaşılan dosyaların konum bilgilerini ağın üyeleri arasında dağıtmaktadır [24].

Atsan ise plansız ağlar ile ilgili 2007 de yaptığı çalışmada, plansız kablosuz ağlarda ölçeklenebilirlik ile ilgili çalışmalar yapmış, daha büyük ağlarda ağ iletişim kuralları tasarımı yapılırken ölçeklemenin farklı olması gerektiğini, bunun dikkate alınmaması durumunda ise dosya paylaşımı gibi uygulamalarda sorunlar olabileceğinden bahsederek düşük maliyetli veri arama iletişim kuralı ile birleştirilmiş, ölçeklenebilir bir veri kopyalama sistemini (SCALAR) önermiştir [25].

Sokullu, ve Sayar ise 2005 de yaptıkları çalışmada bluetooth tabanlı plansız kablosuz ağlarda veri iletimi algoritmalarını incelemişler ve bir model önermişlerdir. Bu yapı; bluetooth için oldukça önemli olan harcanan enerjinin, minimize edilmesini ve ağ bant genişliğinin daha verimli bir şekilde kullanılmasını sağlayarak yönlendirme işleminin performansını artırmayı hedeflemektedir [26].

Lucio ve ark. ise CBR ve FTP [27] verileri kullanarak OPNET ve NS2 ağ simülatörleri üzerinde 6 farklı senaryo çalıştırmıştır. Elde edilen sonuçlar ile gerçek ağlardaki sonuçlara ne kadar doğrulukla yaklaşıldığını paket paket analiz yöntemi ile denemişler ve simülatörlerin parametrelerinde yapılan iyi bir ayarlama ile gerçek ağlara çok yakın davranacağı sonucuna varmışlardır. Buna göre CBR verileri gerçek ortamdakine daha doğrulukla yaklaşırken FTP trafiği için küçük ayarlamaların yapılması gerektiği sonucuna varılmıştır [28].

NS2 üzerinde yapılan bir diğer çalışma ise 2007 de Ivanov ve ark. tarafından yapılmıştır. Yapılan çalışmada, NS2'nin kablosuz ağlarda kullanılmasının % 0.3, % 10 ve % 57 ortalama hata ile ne kadar doğrulukla çalışabileceği gösterilmiştir [9].

Bu çalışmalara ek olarak aşağıda kablosuz ağlarda bilinen problemlerle ilgili çalışmalardan derlenen bilgilere yer verilmiştir.

Veri paketlerinin çarpışması ile ilgili yapılan çalışmalarda, bu sorunun kablosuz ağlardaki büyük problemlerden biri olduğuna değinilmektedir. Bu çarpışma, alıcının iletim aralığı içinde olup göndericinininkinde olmayan gizli düğüm'lerden kaynaklanmaktadır [10, 11, 18, 30, 31].

IEEE 802.11 ile ilgili yapılan çalışmada ise [13], kablosuz ağların gizli düğüm problemlerinden kaçınmak için RTS/CTS mekanizmasını kullanıldığı anlatılmaktadır [5, 6, 12, 39, 40].

Yapılan bir diğer çalışmalar da, RTS/CTS mekanizmasının gizli düğüm problemini çözmesine rağmen bütün veri paket çarpışmalarını koruyamadığı belirtilmektedir. Bu

çalışmalar, bazı komşu düğümlerin (neighboring node), diğer devam eden iletimlerinden dolayı RTS ve CTS'i doğru bir şekilde dinleyemediklerini ve bu durumdaki düğümlerin sağır veya örtülü düğüm olarak ifade edildiklerini göstermişlerdir. Aynı zamanda bu düğümlerin bir hayli veri paket çarpışmasına neden oldukları da bu çalışmalarda bahsedilmektedir [8, 18, 29].

İncelenen çalışmalarda tespit edilen bir diğer problem ise birbirini engelleme etkisidir. Bu etki ile ilgili bu çalışmada herhangi bir anda kablosuz ağlardaki iletimden çekilen pek çok düğüm, blokeli düğümler olarak gösterilmiştir [16].

Blokeli düğümler ile ilgili yapılan bir çalışma, herhangi bir RTS, blokeli düğüme gönderilirse, gönderici herhangi bir CTS geri almayabilir ve blokeli düğümün, birbirini engellemeye neden olan geri çekilme moduna (back-off mode) döneceği ifade edilmektedir. Ayrıca birbirini engelleme sebebiyle bu durumun ağ boyunca çoğaldığını ve ağda kilitlenmiş ya da tıkanıklık denilen durum yol açacağı bahsedilmektedir [2].

### 3. MOBİL KABLOSUZ AĞLAR VE AĞ SİMÜLATÖRLERİ

Bu bölümde mobil kablosuz ağlar ve bu ağlardaki gizli düğüm, örtülü düğüm, bloklama problemlerinden, RTS/CTS (Request-to-Send/Clear-to-Send) mekanizmasından ve ağ simülatörlerinden bahsedilmiştir.

#### 3.1. Mobil Kablosuz Ağlar

Mobil plansız ağ (Mobile Ad-Hoc Network), aralarında kablosuz bağlantı olan, düğümlerin (bağlantı noktaları, host) bağlı olduğu, ayarlanabilen mobil yönlendiricilerden (routers) oluşan kablosuz bir ağıdır. Bu tür ağların oluşturduğu topoloji değişken bir düzene sahiptir yani hareketlidir [22].

Ağın mobil olmasından dolayı, yönlendirici ve düğümlerin oluşturduğu kablosuz topoloji her an hızlı ve beklenmedik bir şekilde değişebilmektedir. Diğer taraftan bu gibi ağlar kendi aralarında iletişim kurabildiği gibi daha geniş ağ olan Internet'e de bağlanabilirler [2].

Kablosuz ağlarda, en yaygın kullanılan Ortama Erişim Kontrolü (MAC - Media Access Control) iletişim kuralı, basitliği ve ölçeklenebilirliği bakımından, Taşıyıcı Dinleyen Çoklu Erişim'dir (CSMA - Carrier Sense Multiple Access). CSMA, bir düğümün diğeri ile iletişime geçemediği, düğümün onun alım aralığının dışında olduğu, örtülü düğüm problemine duyarlıdır. Bu çoğunlukla veri paketlerinin çarpışması ile sonuçlandığı kablosuz ağlardaki temel problemlerden biridir. Bu gibi pahalı çarpışmalardan kaçınmak için, IEEE 802.11 kablosuz ağ standartları CSMA olarak adlandırılan RTS/CTS tokalaşma (handshake), Taşıyıcı Dinleyen Çoklu Erişim/Çarpışmadan Kaçınma iletişim kuralını (CSMA/CA-Collision Avoidance) önermektedir [20].

Halbuki RTS/CTS tokalaşma, gizli düğüm senaryosunu çözmesine rağmen ağı geçen verileri, anlamlı bir seviyede etkileyecek diğer meselelere yol açar. RTS/CTS tokalaşma varsayımında, kontrol paketlerini dinlemedeki komşu düğümler iletimlerini erteleyebilir ve bu yüzden her zaman veriyi doğrulukla taşımayabilir. Bu

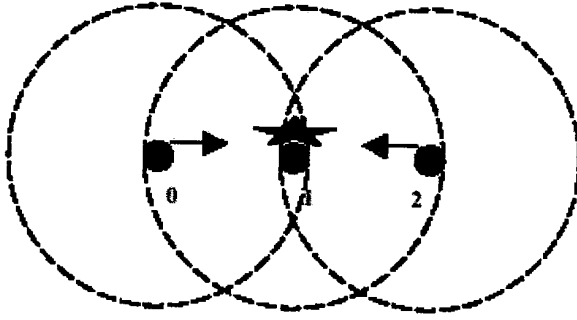
durum, komşu düğümlerin yakınlarında devam eden diğer iletimlerden dolayı kontrol paketlerini doğru bir şekilde yorumlayamamasından kaynaklanmaktadır. Bu da örtülü (ya da sağır) düğüm problemi olarak adlandırılan duruma yol açar ve ilgili düğüm örtülü (ya da sağır) düğüm olarak adlandırılmaktadır [8].

RTS/CTS tokalaşma'nın kablosuz ağlarda yol açtığı diğer bir ilintili problem birbirini engelleme etkisidir. Bu engelleme etkisi, herhangi bir anda, ağda bulunan çoğu düğümün, gereksiz bir şekilde iletimden çekildiği senaryolara yol açmaktadır. Bu gibi düğümler blokeli düğümler olarak adlandırılmaktadır [16].

### **3.1.1. Gizli Düğüm Problemi**

Kablosuz ağların performansı, başlıca, kullanılan ortama erişim kontrolü iletişim kuralına bağlıdır. Yaygın bir şekilde kullanılan MAC iletişim kuralı CSMA - Taşıyıcı Dinleyen Çoklu Erişim, gizli düğümlerden dolayı veri paketi çarpışmasını engelleyemez. Eğer bir düğüm göndericinin değil de alıcının iletim aralığına yerleştirilmişse, paket alışıverişini işitmeyecektir ve aynı alıcıya paket göndermeye başlayamayabilir. Bu çoğunlukla veri paketi çarpışması ile sonuçlanmaktadır. Bu gibi düğümler gizli düğüm olarak adlandırılır.

Kablosuz bir ağda, her bir düğüm alıcının aralığında ve taşıyıcıyı dinleyen aralıkta bir sinyale sahiptir. Bir düğüm kendi alıcı aralığından ötede herhangi bir paketi alamamakta ve taşıyıcı dinleyen aralığın ötesindeki herhangi bir komşu iletimini hissedememektedir. Bu iki aralığın eşit olmadığı varsayılırsa, aralığın dışında düğümlerin birbirinden gizlendiği bir senaryoya sahip olunmaktadır ve bu da çoğunlukla veri paketi çarpışmasına yol açmaktadır.



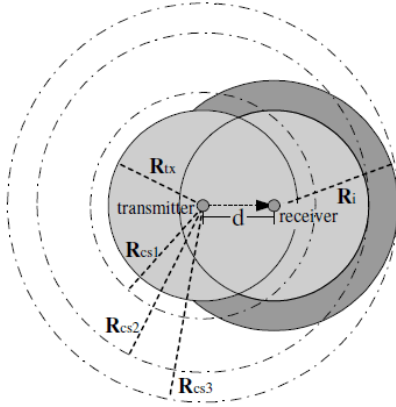
Şekil 3.1. Gizli düğüm problemi

Şekil 3.1.'de, düğüm 1'in düğüm 0 ve düğüm 2'nin alım aralığında olduğunu ama düğüm 0'ın düğüm 2'den gizlendiğini ve diğer durumları görülmektedir. Bu durumda, eğer düğüm 0 ve düğüm 2 paketlerini aynı anda düğüm 1'e gönderirse, düğüm 1'de veri paketi çarpışmasına neden olabilir.

Bu, verinin önemli derecede etkilendiği kablosuz ağlarda ciddi bir problemin olduğunu kanıttır ve bu durumlardan kaçınmak için, kablosuz MAC iletişim kuralı IEEE 802.11 standartları, iletme geçmeden önce RTS/CTS tokalaşmayı önermektedir [13].

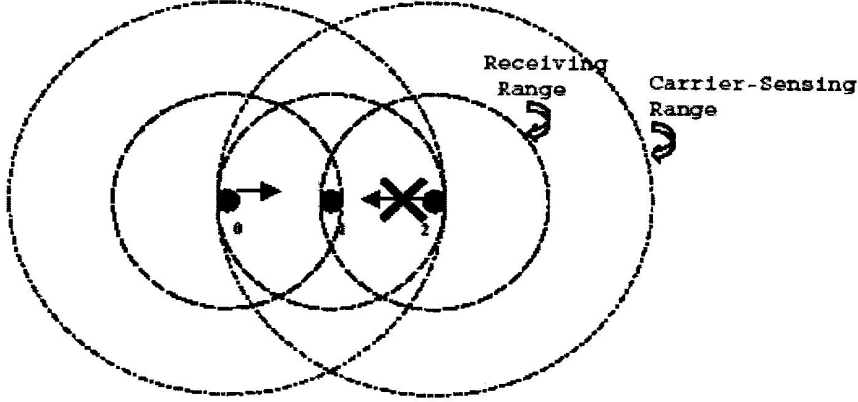
Ağ simülatörlerindeki duruma bakıldığında ise NS ağ simülatöründe, gizli düğüm senaryosu, varsayılan NS parametreleri ile asla gözlenilmemektedir. Çünkü NS'de bir düğümün varsayılan taşıyıcı algılama aralığı onun alım aralığının yaklaşık 2 katına çıkarılmıştır. Dolayısıyla varsayılan parametrelili herhangi NS simülasyonu, kablosuz ağların performansını anlamlı bir seviyeye indirmediği bilinen gizli düğümler olmadan, hatalı sonuçları göstermektedir.

Deng ve ark. da, IEEE 802.11'de taşıyıcı algılama aralığının iyi tanımlanmadığını, NS2'de bu değer 2.2 katı ayarlandığını ve iletim aralığının bu aralığa eşit olması gerektiğini belirtmiştir. Böyle olduğunda gizli düğüm probleminin daha belirgin olacağını da açıkça ifade etmiştir [32].



Şekil 3.2. Fiziksel taşıyıcı algılama aralığının ( $R_{cs}$ ) girişimi nasıl etkilediği

Xu ve ark. ise gizli düğüm probleminin tespiti için taşıyıcı algılama aralığını 3 farklı değerde alıp ağ performansını nasıl etkilediğine bakmışlar ve en büyük değerli taşıyıcı algılama aralığının ağ bant genişliğini önemli ölçüde azalttığını göstermişlerdir [12]. Şekil 3.2.'de bu durum açık olarak görülmektedir.



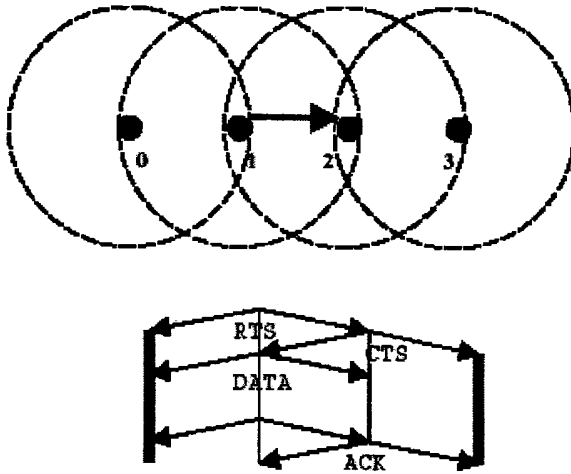
Şekil 3.3. Varsayılan değerlerle NS de 3 düğümün iletim aralıkları

Varsayılan NS parametreleri ile kurulan şekil 3.3.'deki gibi bir senaryoda, düğüm 0 düğüm 1'e veri paketi gönderdiği zaman, düğüm 2, düğüm 0'ın alım aralığında olmamasına rağmen onun algılama aralığında olduğu için, düğüm 0 ve düğüm 1 arasındaki paket değişimini hissetmektedir. Bu sebepten düğüm 2, veri paketi çarpışmasından kaçınmak için düğüm 1'e paket göndermekten sakınabilir. Paket çarpışmasını bu şekilde önlemek sadece algılama aralığı alım aralığının iki katına ayarlandığında mümkün olmaktadır. NS'deki alım ve algılama aralığını gösteren

parametreler arasındaki bu farklılık yüzünden, gizli düğümden kaynaklanan bütün veri paketi çarpışmaları gözlenmemektedir. Halbuki gerçek kablosuz ağlarda, gizli düğüm problemi eşit alım ve algılama aralığı ile epeyce ağ performansı etkisine sahip görünmektedir. NS simülatörü bu parametrelerini, 914MHz Lucent WaveLAN DSSS radio arabirimine göre ayarlamaktadır [21].

### 3.1.2. RTS-CTS Mekanizması

Gizli düğüm problemi, CSMA ile birlikte Kablosuz Çoklu Erişimli Çarpışmadan Kaçınma - MACAW (Multiple Access Collision Avoidance for Wireless) kullanan IEEE 802.11 tarafından desteklenmiş, RTS/CTS tokalaşma ile dikkate alınmıştır. Temel paket iletim yönteminde; eğer kanal boş ve veri paketini alan alıcı düğüm bir alındı damgası (Ack) ile geri dönmüş ise, düğüm, veri paketini iletmektedir. Daha önceki bölümde görüldüğü üzere bu, paket çarpışması ve paket kaybına neden olabilmektedir. Bu meseleyi çözmek için, RTS/CTS tokalaşma önerilmiştir [5, 6]. Bu mekanizmada, senaryo şekil 3.4.'deki gibi gösterilmektedir.



Şekil 3.4. IEEE 802.11 MAC ile RTS-CTS tokalaşma

Düğüm 1 düğüm 2'ye veri paketi göndermek istediğinde, gönderme isteği (RTS) olarak adlandırılan küçük bir kontrol paketi gönderir. RTS'yi alan Düğüm 2 gönderime açık (CTS) diye adlandırılan bir başka kontrol paketi ile yanıt vermektedir. RTS ve CTS'yi işiten komşu düğümler, düğüm 0 ve düğüm 3, bu

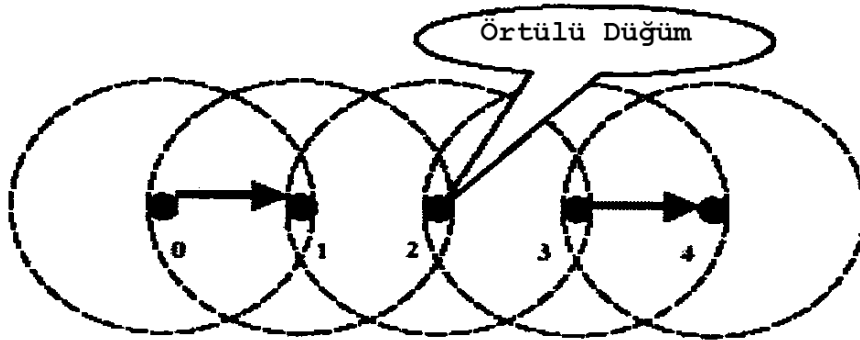


durumda, kendilerinin ağ yerleşim vektörlerini (Network Allocation Vector - NAV) ayarlamakta ve iletim bitine kadar kendi herhangi iletimlerini ertelemektedir. Erteleme süreci RTS ve CTS'ye karşılık olarak belirlenmiştir. CTS alındıktan sonra, düğüm 1, CTS'i alındı damgası (Ack) ile yanıtlayan düğüm 2'ye veri paketini göndermektedir. Eğer bir düğüm RTS'si ya da veri paketi için yanıt almazsa, yeniden gönderimden önce üstel geri çekilme (exponential back-off) moduna girmektedir.

### 3.1.3. Örtülü Düğüm Problemi

RTS-CTS tokalaşma, gizli düğüm problemini “iletim aralığındaki her düğüm RTS ya da CTS'yi işitebilir ve bu yüzden de iletimini erteler” varsayımı ile çözmeyi amaçlamaktadır. Halbuki bu varsayım her zaman doğru olmamaktadır. Göndericinin ya da alıcının aralığındaki bütün düğümler devam eden diğer iletimlerinden dolayı RTS ya da CTS'yi doğru bir şekilde çözmemektedir. Eğer bir düğüm aynı anda iki ya da daha fazla sinyal alırsa, genellikle bu paketleri çözemez. Bu gibi bir düğüm örtülü ya da sağır olarak adlandırılmaktadır [20].

Örtülü düğümler veri paketi çarpışmasına yol açmasa da, RTS ya da CTS'li veri paketi ya da alındı damgası paketi çarpışmasına yol açıyormuş gibi gösterilmektedirler. Bu gibi çarpışmaların olasılığı oldukça yüksektir, dolayısıyla ağın bant genişliğini oldukça etkilenmektedir. Bu bağlamda veri paketi çarpışması ve alındı damgası paketi çarpışması şeklinde iki senaryo açıklanmaktadır.



Şekil 3.5. Örtülü düğüm problemi

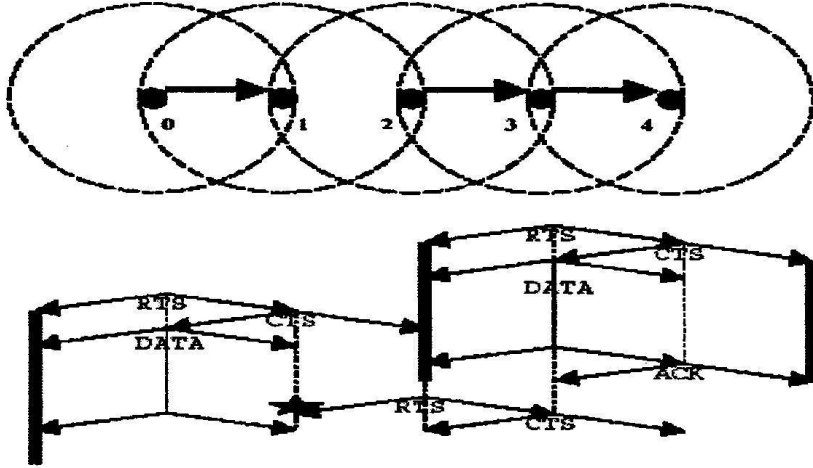
Örtülü düğüm problemleri ile ilgili olarak, ağ benzetim araçlarındaki duruma bakıldığında varsayılan parametresi ile NS, bu problemi gerçek ortamdaki gibi yansıtamamaktadır. NS veri iletiminde kullandığı parametrelerini eski 802.11 WaveLan kartlarına göre ayarladığından gerçekteki gibi örtülü düğümleri gösterememektedir. Bu yüzden günümüzde kullanılan kartlara göre NS'in yeniden yapılandırılması gerekmektedir.

### 3.1.3.1. Veri paketi Çarpışması

Başlangıçta bütün düğümlerin boşa olduğunu varsayalım, bu durumdayken düğüm 3 düğüm 4'e veri paketi göndersin. Veri paketinin iletiminden önce düğümlerin başarılı bir şekilde RTS-CTS tokalaşmanın karşılıklı olarak yapması gerekir. Bu esnada düğüm 2 de, düğüm 3'den gelen RTS'i alacak ve gereği olarak NAV'ını ayarlayacaktır. Tokalaşma bittikten sonra düğüm 3, düğüm 4'e veri paketini göndermeye başlar. Ancak düğüm 3'ün bu iletiminden haberi olmayan düğüm 0'ın da, düğüm 1'e veri paketi gönderdiğini varsayalım. O zaman, düğüm 1'e de RTS gelecek ve düğüm 1 de CTS ile geri yanıtlayacaktır. Düğüm 0, doğru bir şekilde bu CTS'i alır. Aynı zamanda düğüm 2'nin de, bu CTS'yi alması ve zaten düğüm 3 ve düğüm 4 arasındaki iletimden dolayı erteleme modunda olduğundan, doğru bir şekilde bu CTS'i çözemeyecek NAV'ını ayarlaması beklenir. Çünkü CTS paketi düğüm 1'in kapsama alanı içinde hem düğüm 0'a hem düğüm 2'ye gitmiştir. Bu durumda olan bir düğüm, yani düğüm 2, örtülü düğüm olarak adlandırılmaktadır. Düğüm 0, CTS'yi doğru bir şekilde aldığından, düğüm 1'e veri paketini göndermeye başlar. Düğüm 2, düğüm 3 ve düğüm 4 arasındaki iletim bittikten sonra erteleme süresi biteceğinden, diğer devam eden iletimlerden farkında olmaksızın her bir komşusuna RTS göndermekte serbesttir. Eğer böyle bir RTS paketi düğüm 1'e ulaşırsa, düğüm 1'in düğüm 0'dan aldığı veri paketi imha olacaktır.

Bir diğer ihtimal de, düğüm 3'ün düğüm 4'e iletimini bitirdikten sonra, düğüm 2'ye RTS göndermesi olsun. Bu durumda düğüm 2, CTS ile geri dönecek ve kapsama alanı içerisinde bu CTS aynı zamanda düğüm 1'e ulaşacaktır. Ancak düğüm 0'dan

düğüm 1'e giden veri paketi yine yok olacaktır. Şekil 3.6.'da bu durum gösterilmiştir.

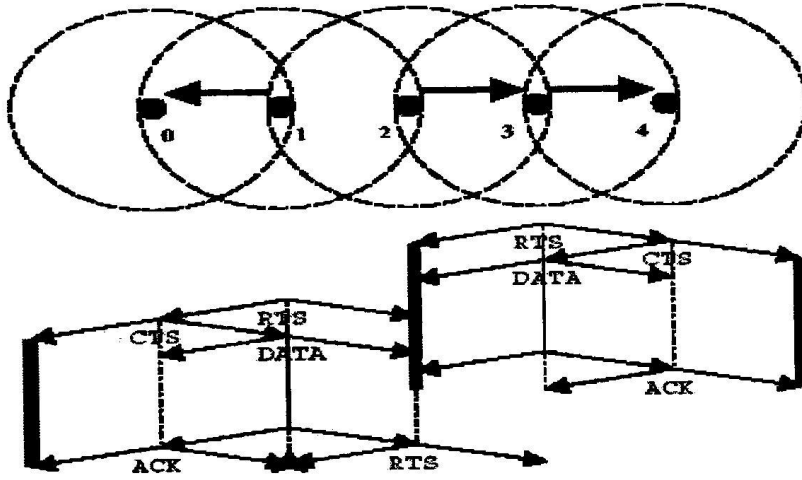


Şekil 3.6. Örtülü düğümün neden olduğu veri paket kaybı senaryosu

### 3.1.3.2. Alındı Damgası Paketi Çarpışması

Örtülü düğümün önemli derecede veri paketi çarpışmasına neden olduğu bir başka senaryo, alındı damgası paketinin yeniden veri paketi iletimine yol açan örtülü düğümden bir RTS ile çarpıştığı durumdur. Bu durumda, bütün düğümler başlangıçta boşta. Yukarıdaki duruma benzer şekilde, düğüm 3 düğüm 4 ile başarılı bir RTS-CTS değişimini başlatır. Düğüm 2 RTS alır ve gereği olarak NAV'ını ayarlar. Daha sonra düğüm 3, düğüm 4'e veri paketini göndermeye başlar. Aynı zamanda düğüm 3'ün bu iletişiminden farkında olmayan düğüm 1 de, düğüm 0'a veri paketi göndermek isterse, düğüm 0'a bir RTS gönderecek ve düğüm 0 CTS ile geri yanıtlayacaktır. Düğüm 1 bu CTS'i düzgün bir şekilde alacaktır, ve düğüm 2'nin de düğüm 1'den RTS'i alması ve çoktan erteleme sürecinde olduğundan bu RTS'yi düzgün bir şekilde çözemeyen NAV'ını ayarlaması beklenecektir. Bu durumdaki düğüm 2, örtülü düğüm olarak adlandırılır. Düğüm 1 daha sonra CTS'i düzgün bir şekilde aldığı için, düğüm 0'a veri paketini göndermeye başlar. Düğüm 2'nin, düğüm 3 ve düğüm 4 arasındaki iletişim bittikten sonra erteleme biteceğinden, diğer devam eden iletimlerin farkında olmaksızın komşu düğümlerin her birine RTS göndermekte serbesttir. Bu süreç içinde, düğüm 1 veri paketini göndermeyi bitirir ve düğüm 0 bir

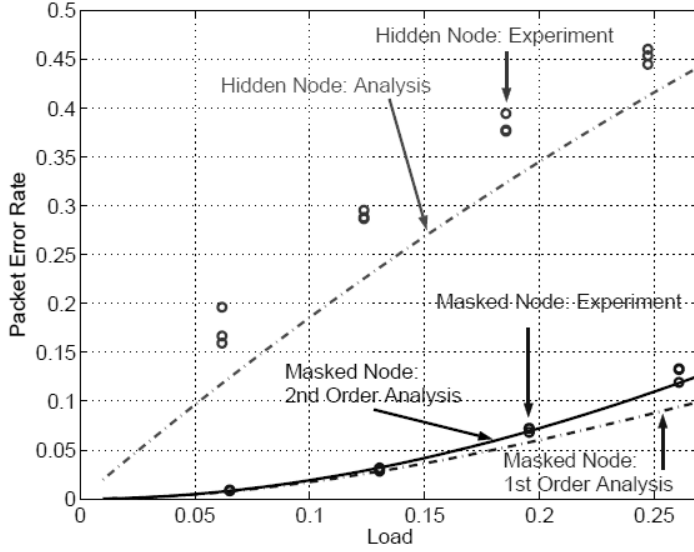
alındı damgasını gönderir. Bu alındı damgası, düğüm 2'den düğüm 1'e gelebilecek RTS ile çarpışabilir ve bu da zaten daha önceden başarılı bir şekilde gönderilmiş veri paketinin yeniden gönderilmesine neden olur. Şekil 3.7.'de bu durum gösterilmiştir.



Şekil 3.7. Alındı damgasından kaynaklanan veri paket kaybı senaryosu

### 3.1.4. Örtülü Düğüm ve Gizli Düğüm Problemlerinin Simülasyon Ortamında Karşılaştırması

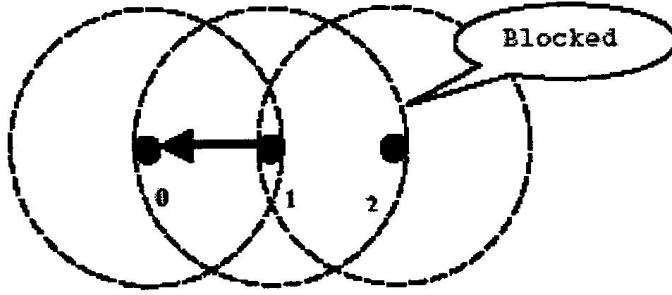
Örtülü düğümler gizli düğüm kadar kritik olmasa da, çok yüklü ağlar için anlamlı derecede yüksek hata olasılığına sahiptirler. Bu iki problem arasındaki karşılaştırma şekil 3.8.'de gösterilmiştir. Bu karşılaştırmanın yapıldığı simülasyon ortamında örtülü ve gizli düğüm simülasyonları için, paket büyüklüğü 1000 byte ve her bir düğümde Poisson paket üretiminin farklı oranlarında üretilmiş verinin hızı 1 Mbps olarak ayarlanmıştır. Şekildeki  $\rho$  (rho), ağ iş yükünü temsil etmektedir ki bu iş yükü Poisson paket üretimi ve toplam simülasyon zamanındaki üretilen hıza karşılık gelmektedir. Hata Olasılığı PER (Packet Error Rate), üretilen toplam paket için veri paketi kaybı olarak tanımlanmaktadır.



Şekil 3.8. Gizli ve örtülü düğüm için yüke karşılık gelen paket hata oranı

### 3.1.5. Bloklama Etkisi Problemi

Ağdaki bir düğüm eğer erteleme modunda ise bloke edilmiş olabilir, bunun anlamı iletimden yasaklanmıştır. Bloke edilmiş düğümün aralığı içindeki komşu düğümler bu düğümün bloke edildiğinin farkına varmayabilirler ve onlardan biri blokeli düğüm ile iletişime başlayabilir. Gönderici bloke edilmiş düğüm için planlanmış RTS gönderir ama alıcı bloke edildiği için CTS ile geri yanıtlamaz ve ağ performansının bozulması sebebiyle düğüm uzun süren geri çekilme moduna girer. Bu durum bloklama problemi olarak anılmaktadır. Halbuki bu RTS'i alan diğer komşu düğümler NAV'larını ayarlar ve aynı zamanda hatalı bloklamaya yol açan erteleme moduna girerler. Bu hatalı bloklama ağ boyunca yayılabilir ve ciddi bir şekilde ağ performansını etkileyebilir. Bir diğer ciddi problem eğer hatalı bloklama durumu ölü noktalara yol açarsa, nihayetinde çok sayıda paket düşmesi ile parçalara bölünme ve geri çekilme girişimleri oluşabilir [16].



Şekil 3.9. Örtülü düğüm

### 3.2. Ağ Simülatörleri

Ağ simülatörleri çeşitli ihtiyaçlara hizmet ederler. Bilgisayar, yönlendirici ve gönderici ve alıcı veri bağlantılarından oluşan çoklu ağ yapısındaki bütün test ortamları, zaman ve maliyet açısından karşılaştırıldıklarında, benzetim test ortamı olan ağ simülatörlerinin görece hızlı ve ucuz oldukları görülebilmektedir. Simülatörler, ağ hizmetlerine yapılan hizmeti engelleme saldırısı DoS (Denial-of-service) atak testleri ve veri trafiğindeki ani çoğalmanın etkisi gibi simülasyonlarda, gerçek donanımlar kullanılarak benzetilmeye çalışılan özellikle yapılması zor ve pahalı olan test senaryolarının hazırlanmasına yardım etmektedirler. Ağ simülatörleri kontrollü ve tekrar edilebilir bir ortamda özellikle yeni bir ağ iletişim kuralı tasarlanırken ya da mevcut iletişim kurallarının değiştirilmesinde yapılan testlerde kullanışlıdır.

#### 3.2.1. NS2 (Network Simulator)

NS, ağ arařtırmalarında hedeflenen ayrık olay benzetimidir. NS, İletim Denetimi Kontrol İletişim Kuralı (TCP - Transmission Control Protocol), yönlendirme (routing) ve kablolu ve kablosuz ağlar üzerinde (yerel ve uydu) çoğa gönderim iletişim kuralı (multicast protocols) için geçerli bir destek sağlamaktadır NS2, VINT projesi tarafından geliştirilen ağ simülatör aracının ikinci sürümüdür [21]. Ağ arařtırma topluluğunda oldukça popüler olan olay güdümlü bir ağ aracıdır. NS2, çoğa gönderim ve TCP seçici alınıldıma gibi daha yeni özellikteki iletişim kurallarını da içermektedir [44]. NS2 ile birlikte gelen Ağ animatörü, NAM [45], paket düzeyinde animasyonlar ve ağ iletişim kurallarının tasarım ve ayıklama işlemleri için özel

grafikler üretir. Ek olarak, NS2’de açık kaynaklı yapısından dolayı farklı seviyelerde yapılandırmalar da yapılabilmektedir. Özel uygulamalar yapabilme kapasitesi ve farklı katmanlarda çeşitli parametrelerde yapılan düzenlemelerle hazırlanan iletişim kuralları bunlara örnek olarak sayılmaktadır [42].

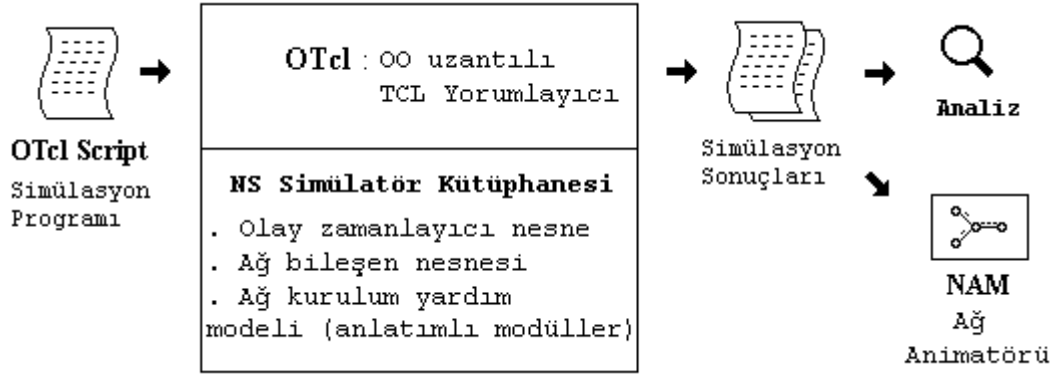
### **3.2.1.1. NS Hakkında**

NS, 1989’da REAL isimli ağ simülatörüne alternatif olarak başladı ve birkaç yıl içerisinde büyük ölçüde kendini geliştirdi. 1995’de LBL, Xerox, PARC, UCB ve USC/ISI’de yapılan VINT projesi aracılığıyla ve DARPA tarafından NS’in gelişimi desteklendi. Bugünlerde DARPA, SAMAN ve NSF CONSER ile ve her ikisinin ACIRI’yi kapsayan diğer araştırmalarıyla işbirliği içerisinde NS’in gelişimi desteklemektedir. NS her zaman kablosuz çalışmalar içeren UCB Daedalus ve CMU Monarch projesi ve Sun Microsystems ile sağlam bir işbirliği içerisinde bulunmuştur.

NS, TCP ve UDP gibi ağ iletişim kuralları ve FTP, Telnet, WEB, CBR ve VBR gibi trafik kaynakları, Drop Tail, RED ve CBQ gibi yönlendirici kuyruk yönetimi mekanizmaları, Dijkstra gibi yönlendirme algoritması ve daha fazlası ile uygulama yaptırabilmektedir. NS ayrıca LAN simülasyonları için MAC katmanı iletişim kurallarının bazıları ve çoğa gönderimli (multicasting) uygulamalar da yapabilir.

NS, nesne yönelimli TCL ve C++ ile yazılmıştır. Bu dil Otcl olarak adlandırılmaktadır. NS2’yi kullanmak için bu betik dili kullanmak gerekir. Bu noktada, C++ sıklıkla yürütülen kodlarda uygulanmaktadır. Otcl ise sistemi ayarlamaktadır. Ağ simülatörü aynı zamanda bir Otcl yorumlayıcıdır ve bir ağ simülasyonu için olay zamanlayıcıyı başlatan bir program, ağ nesnelere ile bir ağ topolojisinin ve nesnelere ve ağ düğümleri arasındaki yolun oluşturulması gibi programlar yapılabilmektedir. NS’den üretilen çıktılar, NAM, Xgraph, Tracegraph, Gnuplot ya da bazı betik diller kullanarak yürütülen işlemlerde gözlenebilmektedir. Kullanıcı, iletilecek paketlerin başlama ve bitiş zamanını vererek bir trafik sağlar. Program, birebir ve görsel olarak ağ boyunca geçen iletim paketlerini göstermektedir

ya da daha ileri seviyede analizler için ağ trafiğindeki paketleri gösterir bir grafik üretir.



Şekil 3.10. NS Yapısının kullanıcı görünümü

### 3.2.1.2. NS'in Amacı

Simülatorün kullanım amacını aşağıdaki gibi sıralayabiliriz:

- Ağ araştırma ve eğitimlerine destek
- İletişim kuralı tasarımı, trafik çalışmaları vb.
- İletişim kuralı karşılaştırmaları
- İşbirlikçi bir çerçeve sağlama
- Açık kaynak, serbest dağıtım
- Paylaşılan kodlar, iletişim kuralları, modeller vb.
- Benzer iletişim kurallarını kolaylıkla karşılaştırma
- Güvenli sonuçlar üretme

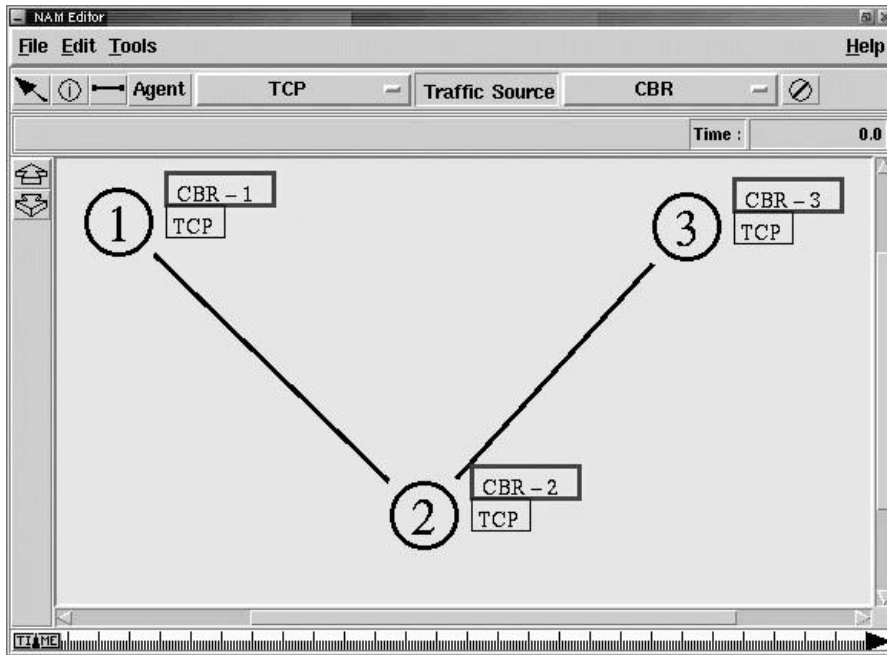


- Modellerde daha fazla durumun daha fazla insanla bakılması
- Uzmanların model geliřtirmesi
- Bir simülatörde çok düzeyli ayrıntı

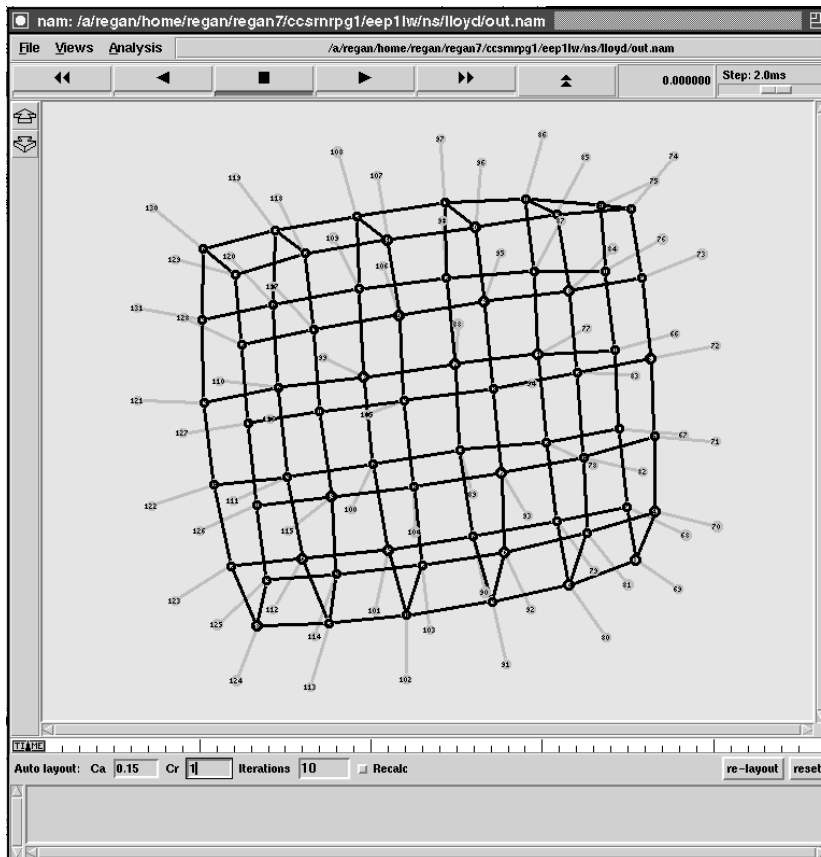
### **3.2.1.3. NS Araçları ve NS ile Kullanılan Bileşenler**

Ağ simülasyon sisteminin büyük bir bölümü NS programının kendisidir (ayrık olay benzetimi). NS programı simülasyon için bir ortamdır ve bazı diğeri bileşenleri onun işlevselliğini tamamlar. NS yerüstü, kablosuz ve uydu ağlarını içeren internet iletişim kurallarını zengin bir şekilde sunan kamuya açık bir simülasyondur.

İkinci önemli aracı NAM'dir (ağ animatör programı). NAM, gerçek zamanlı paket izlemlerini ve ağ simülasyon izlemlerini görüntüleyen Tcl/TK tabanlı animasyon aracıdır. Topoloji yerleşim, paket düzeyinde animasyon ve çeşitli veri denetim araçlarını destekler. NAM ağ kavramını öğrenmek için çok güçlü araçlar kümesidir. NAM düzenleyicisi ile animasyon yaratmak için Tcl kodu yazmaya gerek yoktur. Ağ topolojisi yaratmak ve fareyi sürükleyerek kullanılabilen trafik kaynakları yaratmak ve çeşitli iletişim kurallarını simüle etmek mümkündür.



Şekil 3.11. NAM düzenleyici penceresi



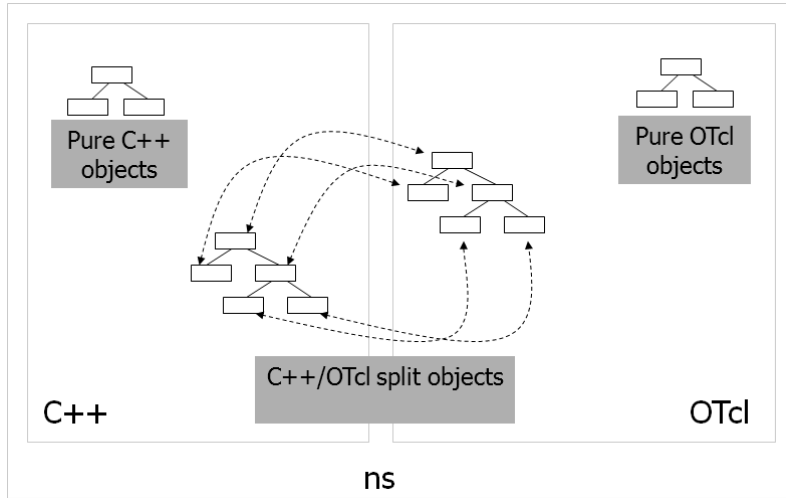
Şekil 3.12. NAM simülasyon penceresi

NAM'in faydalı özellikleri:

- Üretilen ağın görsel olarak kullanılmasını sağlar
- Tcl dili kullanılarak doğrudan yürütülebilir
- Çal, dur, ileri, geri, dondur, hız kontrol ve paket izleme işlevi olan kontroller
- Her bir bağlantıdaki Paket sayısı ve kullanılan bant genişliği gibi bilgiler verir
- Yapılan topolojiler için sürükle bırak arabirimleri sağlar

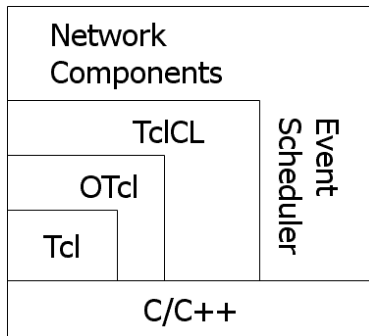
Kullanılan diğer bir araç ise TCL'dir. Tcl, Araç Komut Dili (The Tool Command Language), yorumlanmış betik bir dildir. Simülasyon senaryoları inşa etmek için NS2 ağ simülatörü tarafından kullanılmaktadır. TCL'in söz dizimi anlaşılacak düzeyde değildir.

NS sadece OTcl'de değil aynı zamanda C++'da yazılmıştır. Etkili bir sonuç için, NS veri yolu uygulamalarını kontrol yolu uygulamalarından ayırır. Olay süreç zamanını (simülasyon zamanı değil) ve paketleri azaltmak için de olay zamanlayıcı ve veri yolundaki temel ağ nesnesi bileşenleri, C++ kullanılarak yazılmış ve derlenmiştir. Bu derlenen nesnelere, C++ nesnelere her biri için OTcl nesnelere ile bir eşleşme yaratan OTcl bağlantısı boyunca OTcl yorumlayıcıyı mevcut kılmaktadır ve OTcl nesnelere karşılık gelen üye fonksiyon ve üye değişken olarak hareket eden C++ nesnelere tarafından belirlenen değişkenleri ayarlama ve kontrol fonksiyonları sağlar. Bu yolla, C++ nesnelere kontrolü OTcl'e verilmiştir. Ayrıca C++'ın bağladığı OTcl nesnelere üye fonksiyon ve değişken eklemek mümkündür.



Şekil 3.13. C++ ve OTcl ikilisi

C++’daki nesnelerin simülasyonda kontrol edilmesine ihtiyaç yoktur ya da içten OTcl’e bağlantı kurmasına ihtiyaç duyulmayan başka bir nesne tarafından kullanılmaktadır. Aynı şekilde, bir nesne (veri yolundaki değil) bütünüyle OTcl’de uygulanmış olabilir. Şekil 3.5. C++ ve OTcl’deki bir nesne hiyerarşi örneğini göstermektedir. Şekilde gösterilen şey, hiyerarşiyi düzenleyen OTcl bağlantılarına sahip C++ nesnelere, C++’inkine çok benzeyen OTcl nesne hiyerarşisi vardır.



Şekil 3.14. NS Mimarisi

Şekil 3.5. NS’in genel mimarisini gösterir. Bu şekilde genel bir kullanıcı (NS geliştiricilerden değil) sol alt köşede gösterilenin OTcl kütüphanesindeki simülasyon nesnelerini kullanan Tcl’de simülasyonu çalıştırmada ve tasarlamada olduğunu düşünebilir.

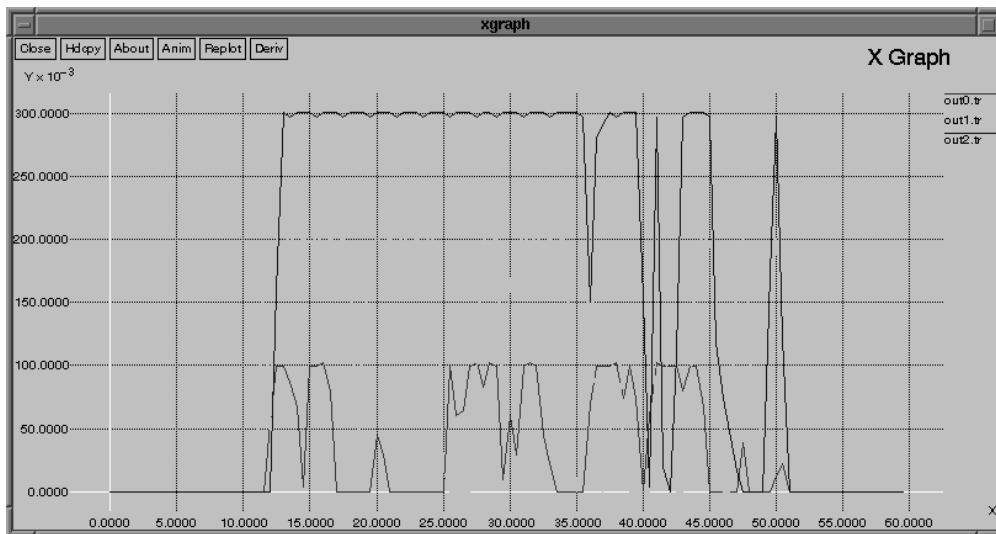
NS tarafından kullanılan diğer bir bileşen, Trafik ve Topoloji üreticilerdir. Topoloji Üretici, NS2 ile belirli bir ağ modelini simule etmek için ağ topolojisi sağlamada kullanılmaktadır. Her bir Topoloji Üretici grafiksel bir arabirim sağlar. Kullanıcı o zaman belli bir düğüm sayısındaki topoloji yapısını belirler. Bu tamamlandığında, üretici NS2 ile kullanılması için topolojiyi gösteren Tcl kodunu üretmek için çalıştırılır.

Veri izleme analizcisi NS ile kullanılan diğer gerekli araçtır. İzleme yapabilmenin iki şekli vardır: XGraph ve TraceGraph.

TraceGraph Windows sistemlerinde çalışan veri izleme analizcisidir ve MatLab 6.0 ya da daha yükseğini gerektirir. TraceGraph Kablolü, Uydu ve Kablosuz destek sağlar.

XGraph Unix sistemlerinde simülasyon sonuçlarının yeniden grafik olarak sunulması için kullanılan çizgisel bir programdır. XGraph;

- İnteraktif çizim ve grafik sağlama
- Animasyon ve türevsel davranış içerir.



Şekil 3.15. Bir grafikte 3 izleme dosyasının XGraph'da karşılaştırılması

### 1.1.1.1. NS'in Simülâtör Olarak Tercih Edilme Sebebi

Ivanov ve ark.'nın kablosuz ađlar üzerinde NS2 ile yaptıkları dođrulama testi sonuçları bu aracın kullanılması için etkenlerden biri olmuştur. Bu araştırmada, kablosuz ađ simülasyonları için kullanılan araçlar arasında, ađ simülâtörü NS2'den sık kullanılan araçlardan biri olarak bahsedilmektedir. Atıf alma incelemesi sonucu simülâtörler arasında NS2'nin %44'lük kullanımla ilk sırayı aldığı görülmektedir. Yine yapılan bu çalışmaya göre NS2 kullanılarak yapılan araştırmalar, IEEE/ACM gibi önemli yayın ve konferanslarda sunulmaktadır.

Uygulamalar ve ađ iletişim kuralları, kaynak ayırma örnekleri [33], gerçek zamanlı iletişim [34] ve plansız ađlarda enerji meseleleri [35], kablosuz sensör ađlarda taşıma iletişim kuralları [36] ve kablosuz robotlarda kontrol stratejileri [37] farklı tipte sistemler için ve farklı ađ katmanlarında geliştirilen çalışmalardır. Ivanov ve ark. bu çalışmalar için Şubat-Eylül 2006 zaman dilimi içerisinde ns-kullanıcı mail listesine gönderilen mesajları incelenmiş ve yaklaşık %30'unda "kablosuz", "mobil" ya da "plansız" anahtar kelimelerine rastlamışlardır. Bu gerçek de göstermektedir ki, kablosuz ađ alanında araştırma ve geliştirme için aktif olarak, NS2 simülâtör aracı kullanılmaktadır.

Bazı yazarlar [38], NS2'nin kablosuz ađın fiziksel katmanındaki yanlışlarından bahsetmektedir. Bu yazarlar ile aynı görüşte olunmasına rağmen hiçbir model %100 dođrulukla zaten çalıştırılmamaktadır.

### 3.2.2. OPNET

Bu modelleyici OPNET teknolojisinin ürettiđi pek çok araçtan birisidir. OPNET modelleyici modülü analitik modellerle entegre edilmiş sonlu durumlu bir modüldür. Modelleyici, iletişim kuralları, cihazlar ve yaklaşık 400 özel amaç için tasarlanmış fonksiyona sahip davranışlar modelleyebilmektedir [9]. Grafik kullanıcı arabirimi olan (GUI - Graphical User Interface) ve lisanslı sürümü, farkedilebilir miktarda doküman ve durum çalışması ile gelen bu modelleyici çarpıcı özellikler barındırmaktadır. İçindeki pek çok düzenleyici, ađ işlemcisi gerektiren farklı

seviyelerdeki modellerin basitleştirilmesini sağlamaktadır. Model parametreleri değiştirilebilen bu düzenleyici açık kaynaklı bir yazılım değildir ve bu modelleyici simülasyonun doğruluğu açısından önemli bir etkiye sahiptir [42].

### **3.2.3. QualNet**

QualNet v 3.1 ölçeklenebilir ağ teknolojilerindedir. QualNet kütüphanesinde çeşitli ayrıntıda kablosuz iletişim kurallarını barındıran ayırık olaylı paralel benzetim ortamıdır [5].

### **3.2.4. Ptolemy**

Ptolemy, farklı simülasyon tipleri için pek çok ayırık durumlu, eş zamanlı veri akışı olan ve koşut zamanlı işlemlere sahip simülasyon ortamlarına bağlanabilen nesneye yönelik genel amaçlı simülasyon aracıdır. Basit modellerin kurulabilmesini sağlayan bu araç, zengin bir kütüphaneye ve aralarında tecrübelerin paylaşılmasını sağlayan haber gurubu olan büyük bir kullanıcı kitlesine sahiptir [6].

### **3.2.5. LZ**

Kablolu ağlar ile ilgili pek çok çalışmada kullanılan paket düzeyli iletişim kuralları simülasyonlarının değerlendirilmesinde kullanılmıştır. Simülatör ayırık zamanlıdır ve şu bileşenlerden oluşmaktadır: çeşitli istatistik modellerine göre veri üretebilen trafik üretici, TCP, UDP, IP, X.25 paket iletişim iletim kuralında kullanılan bağlantı kurucu arabirimler ve telsiz erişim terminali. Simülatör uzayın küçük küplere bölünmesi ile oluşan bir benzetim ortamı kurar ve sinyalin kaynağından merkez küpe mesafesine göre her bir küpteki sinyalin dayanıklılığını ölçmektedir. Küp ile ilgili yaklaşımlardan doğan hatalar küpün boyutunun küçültülmesi ile gözlenebilmektedir [4].

### **3.2.6. SimEleven**

MATLAB'da geliştirilmiş ayırık zamanlı bir olay simülatörüdür. Bu simülatör, iki boyutlu bir ağ simülasyonu sağlamaktadır. Paket çarpışmalarına neden olan örtülü

düğümün etkisi, yayılım gecikmesi ve gezginlik durumunu inceleyen bir çalışmada kullanılmıştır [8].

### **3.2.7. REAL**

REAL, Kolombiya Üniversitesi tarafından NEST simülasyon araç kiti kullanarak inşa edilmiştir. REAL'in (REalistic And Large) inşa edilmesindeki temel motivasyon ilk girenin ilk hizmette kullanıldığı zamanlama şekli ile bilgisayar ağlarında, farklı trafik kaynaklarına iletim bandı olanaklarının adil dağıtılmasını sağlayan kuyruk yönetimini (fair queueing) karşılaştırmaktır [10].

### **3.2.8. WARP2**

Olay güdümlü stokastik simülasyon tabanlı Wireless Access Radio Protocol 2 (WARP2) simülasyon ortamı ComNets'de geliştirilmiştir. Telelogics TAU SDL yazılımı kullanarak özel ve tanımlayıcı bir dilde programlanmıştır. WARP2 kanal modelinde kablosuz ortamda radyo yayılımını doğrulukla simul etmeye çalışmaktadır [13].

### **3.2.9. GPSS**

General Purpose Simulation System (GPSS) ayrık zamanlı simülasyon dilidir. Sisteme giriş hareketi olarak bir sistem tasarlanmıştır ve bloklarla gösterilen bir hizmetten diğerine geçilebilmektedir. Özellikle fabrikadaki problemlere uygundur. 1960'ların sonunda 1970'lerden önce popüler olmuştur ve bugün çok az kullanılmaktadır. GPSS, Simula ve SIMSCRIPT II gibi simülasyon dillerinden daha az esnektir [15].



#### 4. MATERYAL VE METOD

Bu bölümde, mobil kablosuz ağlardaki 802.11 MAC katmanında, gerçek kablosuz ağ ortamındaki verileri simülatör ortamına aktararak, kablosuz ağlardaki gizli düğüm, örtülü düğüm ve bloklama etkisi gibi problemlerden kaynaklı paket kayıplarını gerçek ortamdaki gibi fark edebilmek amacıyla oluşturulmuş test ortamından ve bu test düzeneği için gerçekleştirilen faaliyetlerden bahsedilmektedir.

Bunun için bilgisayar ortamında aşağıdaki özelliklere sahip bir simülasyon ortamı oluşturulmuştur.

**İşletim Sistemi:** İşletim sistemi olarak Unix türevi olan Fedora tercih edilmiştir. Fedora'nın tercih edilmedeki en önemli nedeni, NS2'nin emülatöre gerek duyulmadan kurulabilmesidir. Bu işlem Windows işletim sisteminde yapıldığında cygwin isimli emülatör programının kurulması gerekmektedir. Fedora'nın tercih edilmesindeki diğer nedenler ise herhangi bir emülatöre ihtiyaç duymayan Fedora ile işletim sisteminden kaynaklanabilecek hataların en aza indirgeneceği ki NS2 Unix türevi işletim sistemleri için tasarlanmıştır; Fedora'nın ücretsiz olarak dağıtılması, Fedora'nın bu çalışma için kurulacak bilgisayara daha çok sürücü desteği vermesi; Fedora'da NS2'nin eğitsel amaçlı kullanımının yaygınlığı ve yazılım geliştirmede sadece bu işletim sisteminin tercih edilmesi olarak sayılabilmektedir. Bölüm 4.1 de bu konu ile ilgili daha ayrıntılı bir araştırma sonucuna yer verilmiştir.

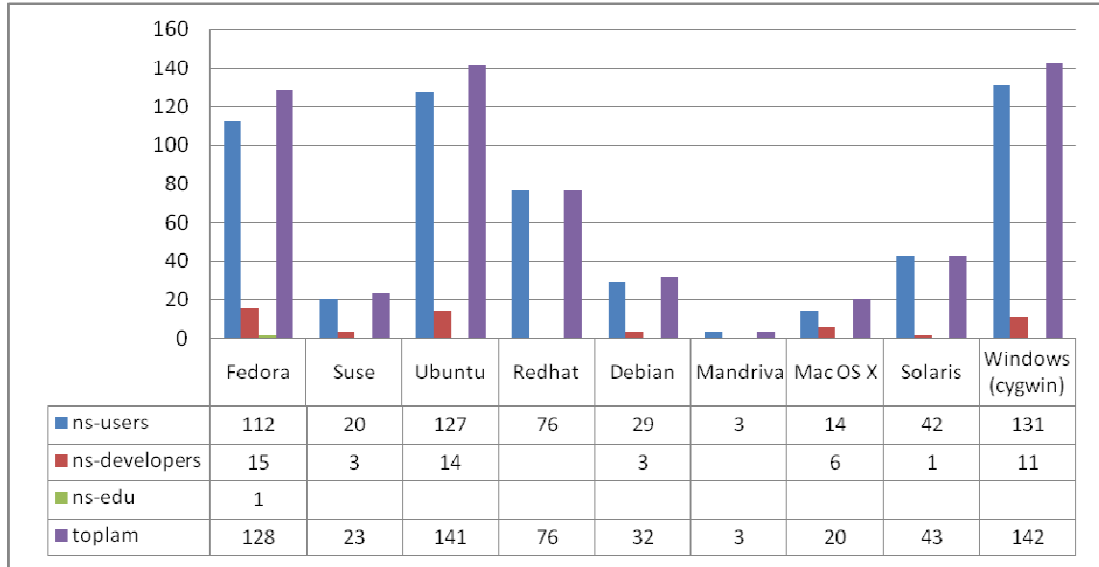
**Simülatör:** Bu çalışmada simülatör aracı olarak NS2 tercih edilmiştir. Tercih edilme sebepleri; diğer ağ simülatörleri içerisinde en çok kullanılan simülatör olarak seçilmiş olması; bilgisayarlar ve telekomünikasyon alanında adını sıkça duyuran IEEE ve ACM gibi önemli oluşumların yayın ve konferanslarında

sıklıkla rastlanması ve mobil kablosuz ağ simülasyonlarının NS'in kendi kullanıcı elektronik posta listesine bakıldığında çokça geçmesi olarak sayılabilir. Hangi simülatörü hangi simülasyonlarda kullanmak gerekir sorusuna yanıt arayan çalışmalara bakıldığında da, mobil kablosuz ağlar için tercih edilen simülatörlerin, NS ve OPNET olduğu bu iki simülatör arasında da NS'in gerçek ortama daha yakın sonuç ürettiği görülmüştür [42].

#### **4.1. Fedora İşletim Sisteminin Tercih Edilmesi ile İlgili Araştırma**

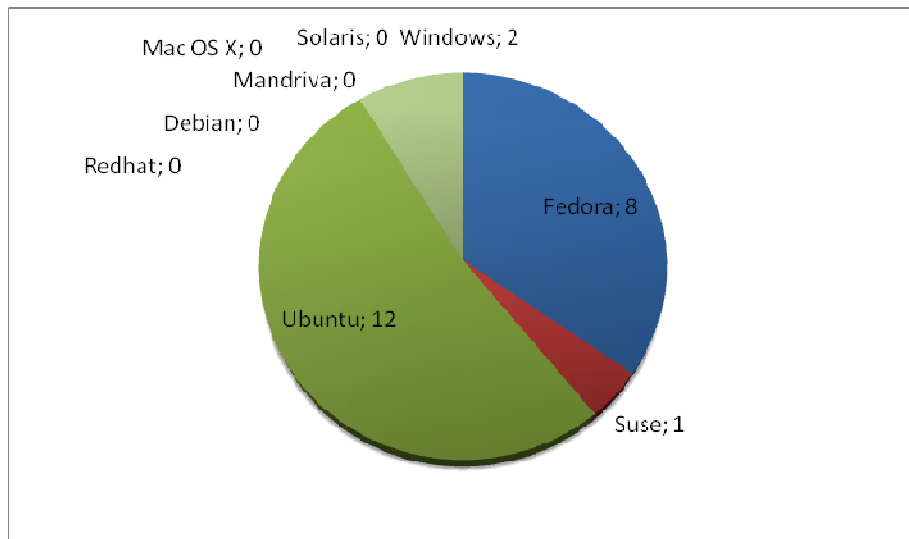
NS2'nin hangi işletim sistemi üzerine kurulması gerektiği ile ilgili kaynak taraması tamamlandıktan sonra bir de NS2'nin kullanıcı posta listesi üzerinde NS ve NAM ile ilgili genel sorular için açılan ns-users, sadece NS geliştiricilerinin duyuru için kullandığı ns-announce, NS ve NAM'ın eğitsel amaçlı kullanımı için açılan ns-edu, NS ve NAM kullanarak yazılım geliştiricilerin tartışabildiği ns-developers üzerinde yayınlanan posta listesinde, mesaj ve konu başlıklarına göre iki araştırma daha yapılmıştır.

Bunun için anahtar kelime olarak posta listesi dili olan İngilizcede “message:ns message:install message:<işletim sistemin adı>” ve “subject:ns subject:install subject:<işletim sistemin adı>” arama metinleri kullanılmıştır. Örneğin; message:ns message:install message:fedora anahtar değerleri ile arama yapılmıştır. Sonuçta şekil 4.1 ve şekil 4.2 deki gibi değerler elde edilmiştir.



Şekil 4.1. Mesaj içeriğine göre işletim sistemlerinin posta listesinde geçme sıklığı

Şekil 4.1.'e bakıldığında yazılım geliştirmede ve eğitsel amaçlı kullanımda toplamda Fedora'nın %30 kullanımla daha çok tercih edildiği görülmektedir. Diğer işletim sistemlerinde ise sırasıyla %26 Ubuntu, %20 Windows cygwin, %11 Mac OS X, %6 Debian, %5 Suse, %2 Solaris yer almış ve Mandriva ve Redhat ise hiç yer almamıştır. Genel toplama bakıldığında ise Fedora'nın 3. sırada yer aldığı görülmektedir.



Şekil 4.2. Konu başlığına göre işletim sistemlerinin posta listesinde geçme sıklığı

Şekil 4.2.'ye bakıldığında da Ubuntu'nun birinci sırada yer aldığı, ikinci sırada Fedora ve sonra sırasıyla Windows ve Suse'nin geldiği görülmektedir. Diğer işletim sistemleri ise mesajın konu başlığında hiç geçmemektedir.

Şekil 4.1. ve 4.2.'ye bakarak Fedora ve Ubuntu işletim sistemlerinin birbirine çok yakın sonuçlar ürettiğini ancak NS2 kurulacak bilgisayarda Fedora için daha çok sürücü bulunması bu çalışma için Fedora işletim sisteminin tercih edilmesine neden olmuştur. Ayrıca Fedora'nın eğitsel ve yazılım geliştiriciler tarafından daha çok tercih edilmesi ve bu çalışmanın da kablosuz ağlarda yazılım geliştirmek için kullanılması tercih edilme sebebini etkileyen diğer sebeptir. Bu çalışma yapılırken Fedora'nın 10. sürümü kararlı sürüm olarak sunulmakta ve 11. sürümü beta sürüm olarak devam etmekteydi.

İşletim sistemi ve Ağ Simülatörü NS2 tercihi yapıldıktan sonra NS2 de mobil kablosuz ağ topolojisinin oluşturulması için simülatör aşağıdaki gibi yapılandırılmıştır:

**Kanal:** Kanal türü olarak mobil ortam için kablosuz kanal tercih etmek gerekmektedir. Bu kanal ayarlama işlemi:

**“set val(chan) Channel/WirelessChannel ;#kanal tipi”**  
ayarı ile yapılabilmektedir.

**Anten:** Anten türü olarak plansız bir uygulama olduğu için her yöne sinyal yayan OmniAntenna türü tercih edilmiştir.

**Arabirim:** Yapılan çalışma kablosuz bir ortamda gerçekleşeceği için seçilen arabirim WirelessPHY ve buna bağlı değerlerde

CPTresh = 10.0 dB

CSTresh = 2.5e-10 W

RXThresh = 2.5e-10 W

Freq = 914 MHz olarak ayarlanmıştır.

**MAC:** Mevcut NS2’de mobil kablosuz ortamlarda kullanılan MAC iletişim kuralı için iki tip tanımlanmıştır, TDMA ve 802.11. TDMA MAC iletişim kuralında, tek sıçramaya destek verildiği için diğer MAC iletişim kuralı olan 802.11 MAC iletişim kuralı tercih edilmiştir ve bütün veri iletimlerinde RTS/CTS mekanizmasını aktif hale getirmek için de MAC RTS eşik değeri aşağıdaki gibi ayarlanmıştır.

**“MAC\_RTSThreshold 0 byte”**

**Kuyruk Tipi:** NS’de iki tip kuyruk vardır, Queue/DropTail, Queue/DropTail/PriQueue. NS’de kablosuz ağ modellemesinin sağlanabilmesi için düğüm yapılandırmasında kuyruk tipi olarak DropTail/PriQueue kullanıldığı için bu kuyruk tipi seçilmiştir.

**“set val(ifq) Queue/DropTail/PriQueue ;#kuyruk arabirim tipi”**

**Bağ Katmanı:** NS’de iki tip bağ katmanı vardır, LL ve LL/Sat. LL/Sat uydu iletişimlerinde kullanıldığı için bağ katmanı simülatörde aşağıdaki gibi ayarlanmıştır.

**“set val(ll) LL ;# bağ katmanı tipi”**

**Yayılm:** NS’de 3 tip radyo yayılım modeli bulunmaktadır, free space model, two-ray ground reflection model ve shadowing model. Bu çalışmada büyük ölçekli mesafelerde daha iyi olduğu bilinen TwoRayGround radyo yayılım modeli kullanılmıştır.

Simülatörde bu modeli aşağıdaki gibi ayarlamak gerekmektedir:

**“set val(prop) Propagation/TwoRayGround; #radyo yayılım modeli”**

- Hareket Şekli:** Simülasyonda, herhangi bir hareketi ya da rassal bir hareketi kapatmak için düğümler hareketliliği göstermemiştir. Düğümler başlangıç pozisyonunda verilmiştir ve simülasyon bant genişliğini korurlar.
- Üretilen Veri:** Veri deseni olarak ağ bant genişliğini sabit olarak kullanan CBR veri deseni kullanılmıştır.
- Yönlendirme:** Mobil kablosuz ağlarda en çok tercih edilen AODV ve DSDV yönlendirme iletişim kuralları arasında yapılan çalışmada, AODV yönlendirme protokolü ağ bant genişliğini daha verimli kullanmasına rağmen performansı düşünmeksizin yapılan daha karmaşık teknikler için DSDV yönlendirme iletişim kurallının tercih edildiği görülmüştür [43]. Bu çalışma sistem performansına yönelik hazırlanmadığı için yönlendirme iletişim kuralı olarak DSDV tercih edilmiştir.

Çalışmada yapılan bütün uygulamalarda düğümlerin düz bir zemin üzerinde hareket ettiği kabul edilerek mobil düğümlerin koordinatları (X, Y, Z=0) olarak kabul edilmiştir. X ve Y değerleri cbrgen.tcl kullanılarak üretilmiştir.

## 5. MOBİL KABLOSUZ AĞLARDA VERİ PAKETİ KAYIPLARINA YÖNELİK UYARLANAN SİMÜLASYONLAR

Bu bölümde daha öncede bahsedildiği gibi gerçek ortamda tercih edilen ağ arabirimlerine ait eşik değer parametreleri (taşıyıcı algılama aralığı, alıcı aralığı ve gönderici isteği eşik değeri) simülatör ortamlarına aktarılarak, kablosuz ağlardaki paket iletimindeki kayıplara neden olan problemler daha gerçekçi olarak sunulmaktadır. Bunun için ağ simülatörü NS2'de, kablosuz ağlarda veri paketi kayıplarına yönelik uyarlanan simülasyonlar yapılmıştır. Bu simülasyonlar boyunca simülatörün varsayılan parametreleri ve sonradan değiştirilen parametreleri (CSThresh\_, RXThresh\_, MAC\_RTSThreshold) kullanılmıştır. Daha sonra yeniden kablosuz ağ modellenmesi sağlanılarak gerçek ortamdaki gibi gizli düğüm, örtülü düğüm ve bloklama etkisi ve RTS/CTS mekanizması ile ilgili veriler elde edilmiştir.

### 5.1. Gizli Düğüm Simülasyonu

NS'de güç seviyeleri bakımından değerleri -78 dBm ve -64 dBm olan, Taşıyıcı Algılama Eşiği (Carrier Sensing Threshold - CSThresh\_) ve Alım Eşiği (Receiving Threshold - RXThresh\_) diye iki parametre tanımlanmıştır. Bu aralıklar, İki Işınlı Yer Yansımali (Two-Ray Ground Reflection) radyo yayılım modeli kullanılarak, mesafesi dikkate alınarak hesaplanmaktadır. Bu model, iki düğüm arasındaki yayılım için yansıma yolunu ve görüş hattını dikkate alır. Bu modelin serbest uzaydan (free space model) daha kesin bir mesafe tahmini verdiği NS tarafından belirtilmektedir.

Mesafeye göre tercih edilen modelde d mesafesinde alınan güç aşağıdaki eşitlikteki gibidir:

$$Pr(d) = PtGtGr(Ht)*(Ht)(Hr)*(Hr) / d*d*d*d*L \quad (1.1)$$

Pt : Gönderilen sinyal gücüdür. Bu değer hesaplanırken kullanılan radyo yayılım modeline bakılmaktadır.

Gt ve Gr : Verici (transmitter) ve alıcı (receiver) anten kazancını göstermektedir ve değer simülatör içerisinde varsayılan olarak 1 yapılmıştır.

Ht ve Hr : Verici ve alıcı anten yüksekliğini göstermektedir. Anten simülâtör içerisinde varsayılan olarak 1.5 m yüksekliğe ayarlanmıştır.

L : Sistem kaybını göstermektedir.

NS'in varsayılan parametreleri olan  $G_t = G_r = 1$  ve  $H_t = H_r = 1,5$  m'dir ve 914 MHz radyo frekansında yayın yapılmaktadır. 24.5 dBm gönderme gücünde olan sistemde, Taşıyıcı Algılama Aralığı 250 m ve Alıcı aralığı bunun yaklaşık iki katı 550 m olacak şekilde ayarlanmıştır. Bu bahsi geçen değerler, NS'in varsayılan parametreleri “../ns-allinone-2.26/ns-2.26/tcl/lib/ns-default.tcl” dosyasında aşağıdaki gibidir:

*# unity gain, omni-diractinal antennas*

*# set up the antennas to be centered in the nodes and 1.5 meters above  
itAntenna/OmniAntenna set X\_ 0*

*Antenna/OmniAntenna set Y\_ 0*

*Antenna/OmniAntenna set Z\_ 1.5*

*Antenna/OmniAntenna set Gt\_ 1.0*

*Antenna/OmniAntenna set Gr\_ 1.0*

*# Initialize the SharedMedia interface with parameters to make*

*# it work like the 914MHz Lucent WaveLAN DSSS radio interface*

*Phy/WirelessPhy set CPTresh\_ 10.0*

*Phy/WirelessPhy set CSTresh\_ 1.559e-11*

*Phy/WirelessPhy set RXThresh\_ 3.652e-10*

*Phy/WirelessPhy set bandwidth\_ 2e6*



*Phy/WirelessPhy set Pt\_ 0.28183815*

*Phy/WirelessPhy set freq\_ 914e+6*

*Phy/WirelessPhy set L\_ 1.0*

Bu varsayılan parametreler ile NS simülöründe, taşıyıcı algılama aralığı, alıcı aralığının yaklaşık iki katına getirildiğinden herhangi bir gizli düğüm gösterilememektedir. Bunun için CStresh\_ ve RXThresh\_ -66 dBm değerinde daha gerçekçi olarak ayarlanmış ve RTS-CTS mekanizması olmadan, basit üç düğümlü senaryo ile simule edilmiştir. Bu simülasyon sonucuna bakılarak CStresh\_ ve RXThresh eşitliğinin ayarlanması ile RTS-CTS mekanizması olmaksızın gizli düğüm problemi gösterilebilmektedir.

Bu problemi gösterebilmek için eşik değerler, tcl dosyasında aşağıdaki gibi ayarlanmıştır:

Varsayılan olarak tanımlı eşik değerler	Değiştirildikten sonra eşik değerler
<i># RX ve CS Eşik -78 ve -64 dBm</i>	<i># RX ve CS Eşik -66 dBm olarak ayarlı</i>
<i>Phy/WirelessPhy set CStresh_ 1.559e-11</i>	<i>Phy/WirelessPhy set CStresh_ 2.5e-10</i>
<i>Phy/WirelessPhy set RXThresh_ 3.652e-10</i>	<i>Phy/WirelessPhy set RXThresh_ 2.5e-10</i>

Normalde NS bütün iletimleri için RTS-CTS tokalaşma uygulamıştır. Halbuki tokalaşma simülörde mac-802\_11.h dosyasında (ns-allinone-2.26/ns-2.26/mac/mac-802\_11.h) MAC\_RTSThreshold'u 3000 byte gibi yüksek bir değere ayarlayarak ki bu 3000 byte'ın altındaki herhangi bir iletim için RTS gönderilmeyecek anlamına gelmektedir. NS varsayılan olarak eşik değerleri Lucent'in Orinoco kartına göre ayarlanmaktadır. RTS/CTS mekanizmasının aktif hale getirilmesi için MAC\_RTSThreshold'un değeri 0 byte'a getirilmiştir. Mac-802\_11.h

dosyasındaki MAC\_RTSThreshold parametresinin değiştirilmiş hali aşağıdaki gibidir:

```

/*

* IEEE 802.11 Spec, section 11.4.4.2

* - default values for the MAC Attributes

*/

#define MAC_RTSThreshold          3000      // bytes

#define MAC_RTSThreshold          0        // bytes

#define MAC_ShortRetryLimit       7        // retransmissions

#define MAC_LongRetryLimit        4        // retransmissions

#define MAC_FragmentationThreshold 2346    // bytes

#define MAC_MaxTransmitMSDULifetime 512    // time units

#define MAC_MaxReceiveLifetime    512     // time units

```

Yapılan değişiklikler hazırlanan üç düğümlü simülasyonda çalıştırılmış ve simülasyonunun ürettiği izleme dosyası şekil 5.1.'deki gibi bulunmuştur:

```

s 0.441355434 _2_ MAC --- 8 cbr 1088 [13a 1 2 800] ----- [2:0 1:0 32 1] [2]
s 0.441535434 _0_ MAC --- 7 cbr 1088 [13a 1 0 800] ----- [0:0 1:0 32 1] [1]
D 0.450059934 _1_ MAC --- 8 cbr 1088 [13a 1 2 800] ----- [2:0 1:0 32 1] [2]
D 0.450239934 _1_ MAC --- 7 cbr 1088 [13a 1 0 800] ----- [0:0 1:0 32 1] [1]
s 0.450907434 _2_ MAC --- 8 cbr 1088 [13a 1 2 800] ----- [2:0 1:0 32 1] [2]
s 0.451747434 _0_ MAC --- 7 cbr 1088 [13a 1 0 800] ----- [0:0 1:0 32 1] [1]

```

Şekil 5.1. İzleme dosyasından alınan değerler

Şekil 5.1.'den görülmektedir ki 7 ve 8 numaralı veri (CBR) paketi, gizli düğümün sonucu olarak düşürülmüştür. İlk olarak düğüm 2, 8 numaralı paketi düğüm 1'e göndermeye başlar ve o zaman düğüm 2'den gizli olan ve bu iletimin farkında olmayan düğüm 0 aynı zamanda 7 numaralı paketi düğüm 1'e gönderir. Paketlerin her ikisi düğüm 1'de çarpışır ve düşürülür. O zaman düğüm 0 ve düğüm 2'nin her ikisi düşen paketleri yeniden göndermeye başlar.

Hazırlanan bu senaryo bir de NS simülatörünün varsayılan CStresh\_ ve RXThresh\_ değerleri için çalıştırılmış ve şekil 5.2. deki gibi bir sonuç elde edilmiştir:

```
s 127.936679222 _0_ AGT --- 35 cbr 512 [0 0 0 0] ----- [0:0 2:0 32 0] [0] 0 3
r 127.954386897 _2_ AGT --- 35 cbr 532 [13a 2 1 800] ----- [0:0 2:0 31 2] [0] 2 3
s 130.171470892 _0_ AGT --- 37 cbr 512 [0 0 0 0] ----- [0:0 2:0 32 0] [1] 0 3
r 130.183831038 _2_ AGT --- 37 cbr 532 [13a 2 1 800] ----- [0:0 2:0 31 2] [1] 2 3
s 132.437411100 _0_ AGT --- 39 cbr 512 [0 0 0 0] ----- [0:0 2:0 32 0] [2] 0 3
r 132.449311246 _2_ AGT --- 39 cbr 532 [13a 2 1 800] ----- [0:0 2:0 31 2] [2] 2 3
s 137.138921309 _0_ AGT --- 40 cbr 512 [0 0 0 0] ----- [0:0 2:0 32 0] [3] 0 3
```

Şekil 5.2. NS varsayılan CStresh\_ ve RXThresh\_ ve RTS-CTS tokalaşmasız

Hazırlanan üç düğümlü senaryo, gerçek ortamda olması beklenen eşit CStresh\_ RXThresh\_ değerleri ve RTS/CTS tokalaşma aktif edilerek tekrarlanmış ve şekil 5.3. deki gibi sonuç elde edilmiştir:

```
s 148.128149424 _0_ AGT --- 47 cbr 512 [0 0 0 0] ----- [0:0 2:0 32 0] [6] 0 3
r 148.140249570 _2_ AGT --- 47 cbr 532 [13a 2 1 800] ----- [0:0 2:0 31 2] [6] 2 3
s 150.000000000 _1_ AGT --- 48 tcp 40 [0 0 0 0] ----- [1:0 2:1 32 0] [0 0] 0 2
r 150.001981727 _2_ AGT --- 48 tcp 60 [13a 2 1 800] ----- [1:0 2:1 32 2] [0 0] 1 2
s 150.001981727 _2_ AGT --- 49 ack 40 [0 0 0 0] ----- [2:1 1:0 32 0] [0 0] 0 2
r 150.004327455 _1_ AGT --- 49 ack 60 [13a 1 2 800] ----- [2:1 1:0 32 1] [0 0] 1 2
s 150.004327455 _1_ AGT --- 50 tcp 1500 [0 0 0 0] ----- [1:0 2:1 32 0] [1 0] 0 2
```

Şekil 5.3. CStresh\_ = RXThresh\_ ve RTS-CTS tokalaşma var

Şekil 5.2. ve şekil 5.3.'e bakılarak paket düşmesinin görülmediği anlaşılmaktadır. Ancak şekil 5.2. de hazırlanan senaryoda, varsayılan NS değerleri kullandığından ve veri paketi çarpışmasını önlemek için kullanılan RTS/CTS tokalaşma mekanizması kullanılmadığından, ideal çözümün şekil 5.3. de elde edilen senaryo ile olduğu anlaşılmaktadır. Aşağıda çalıştırılan bu üç senaryo için özet bir tablo gösterilmiştir:

Durumlar	Sonuçlar
NS varsayılan CStresh_ ve RXThresh_ ve RTS-CTS tokalaşmasız	Paket düşmesi yok
CStresh_ = RXThresh_ ve RTS-CTS'siz	Yukarıdaki izleme dosyasında gösterildiği gibi paket düşmesi
CStresh_ = RXThresh_ ve RTS-CTS tokalaşma var	Çözüm – Paket düşmesi yok

## 5.2. NS'de RTS-CTS Mekanizması Simülasyonu

RTS/CTS mekanizmasının değiştirilen değerler ile çalışıp çalışmadığını görmek adına dört düğümlü düzenek hazırlanmıştır. Yukarıda bahsedildiği gibi CStresh\_ ve RXThresh\_ - 66 dBm ve MAC\_RTSThreshold 0 değerine ayarlıdır. Bilindiği üzere bu şekilde RTS-CTS tasarısının, gizli düğüm probleminden kaynaklı paket düşmesi görülmemiştir.

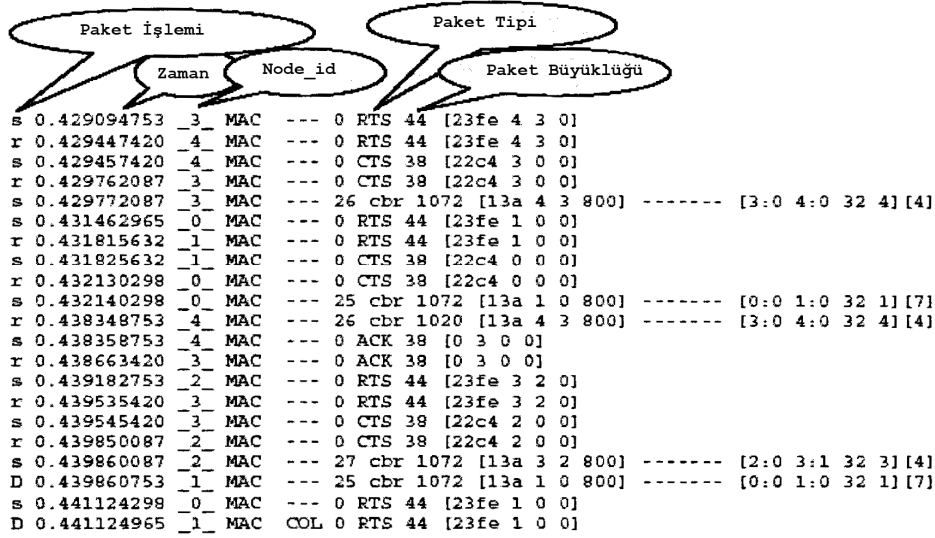
Dört düğümlü senaryonun NS simülasyon izleme dosyası RTS-CTS mekanizmasını şekil 5.4.'deki gibi göstermiştir. Birinci sütundaki s ve r, üçüncü sütundaki node\_id tarafından gönderilen (send) ve alınan (receive) anlamına gelmektedir. Yedinci sütun bu durumda cbr'ın veri paketi tipi olduğu paket tipini gösterir.

```
s 0.382842900 _1_ MAC --- 0 RTS 44 [497e 2 1 0]
r 0.383195400 _2_ MAC --- 0 RTS 44 [497e 2 1 0]
s 0.383205400 _2_ MAC --- 0 CTS 38 [4844 1 0 0]
r 0.383509900 _1_ MAC --- 0 CTS 38 [4844 1 0 0]
s 0.383519900 _1_ MAC --- 5 cbr 2272 [13a 2 1 800] ----- [1:0 2:1 32 2] [2]
r 0.401696400 _2_ MAC --- 5 cbr 2220 [13a 2 1 800] ----- [1:0 2:1 32 2] [2]
s 0.401706400 _2_ MAC --- 0 ACK 38 [0 1 0 0]
r 0.402010900 _1_ MAC --- 0 ACK 38 [0 1 0 0]
```

Şekil 5.4. Dört düğümlü senaryonun izleme dosyasından alınan değerler

## 5.3. Örtülü Düğüm Simülasyonu

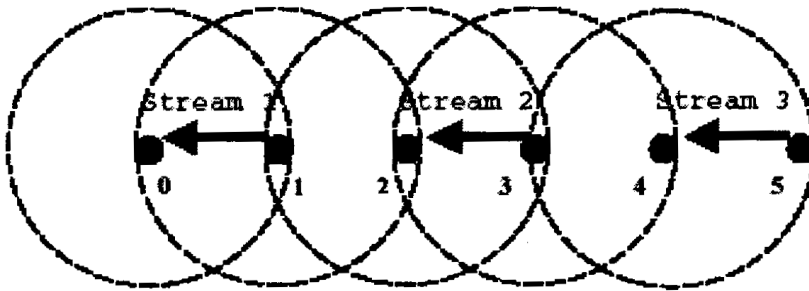
Değiştirilen parametrelerle örtülü düğümlerin gözlenip gözlenemediğini anlamak için dört düğümlü bir simülasyon hazırlanmıştır. Hazırlan simülasyon çalıştırıldıktan sonra, NS simülasyonu şekil 5.5.'deki izleme dosyasında gösterildiği gibi olayları yakalamıştır. 25 numaralı cbr veri paketinin bu problem yüzünden düşürüldüğü görülebilir.



Şekil 5.5. Dört düğümlü senaryolu örtülü düğüm simülasyonundaki izleme dosyası değerleri

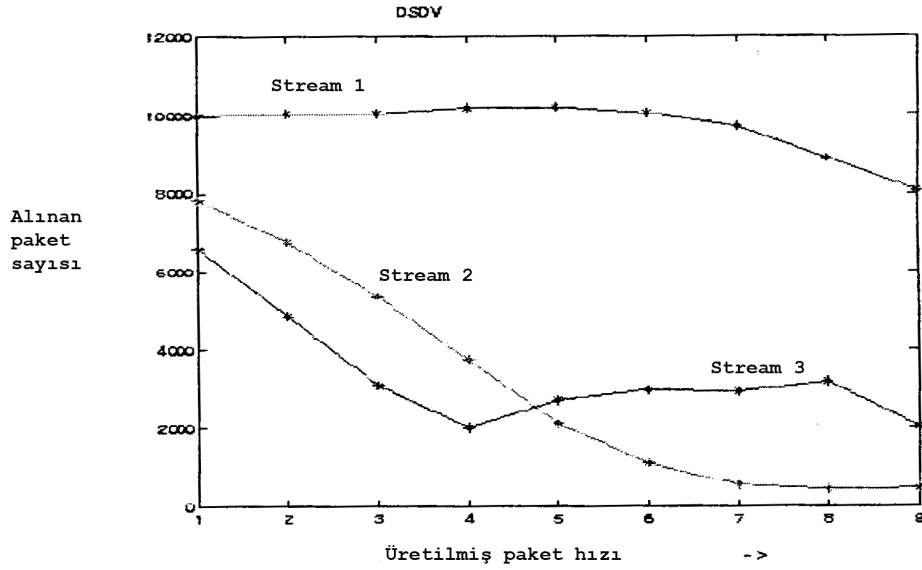
#### 5.4. Bloklama Simülasyonu

Bu simülasyonun yapılması için altı düğümlü bir senaryo oluşturulmuştur. Altı düğümlü bloklama senaryosu, NS simülasyon sonuçları ile doğrulanmış anlamlı seviyede hatalı bloklama göstermiştir. Altı düğüm senaryosunda üç bağımsız iletim simule edilmiş ve alınan veri (CBR) paketinin sayısı şekil 5.6.'da gösterilmiştir.



Şekil 5.6. Bloklama etkisi

Şekil 5.6.'daki gibi üç paket akımı (packet stream) için, her bir düğüm tarafından alınan paket sayısı ve paket üretimindeki farklı oranlar aşağıda verilmiştir. Simülasyon her bir düğüm için yaklaşık 10 000 paket üretmek için çalıştırılmıştır.

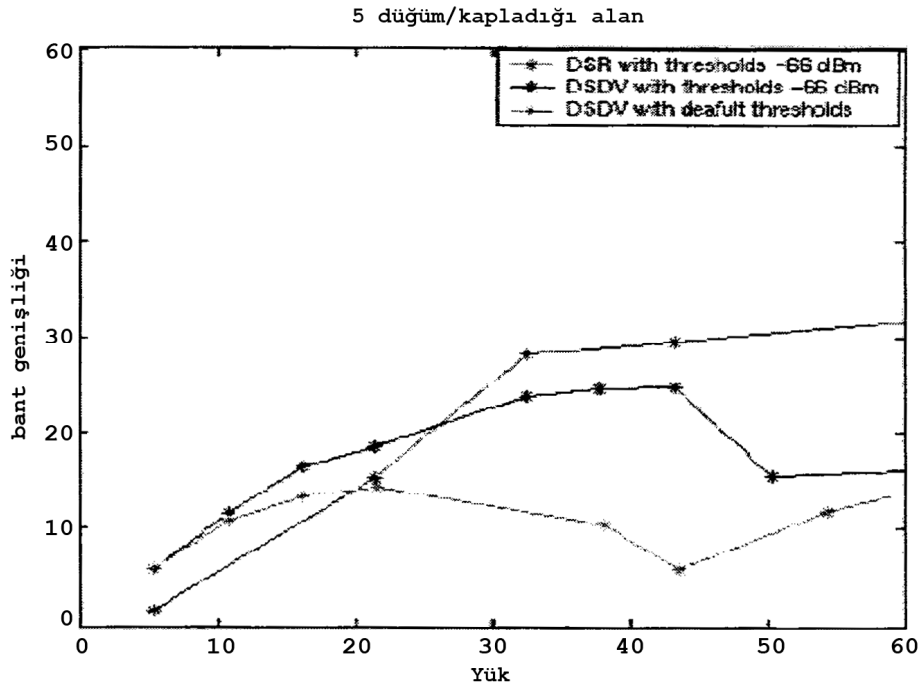


Şekil 5.7. Bloklama etkisinin grafiksel gösterimi

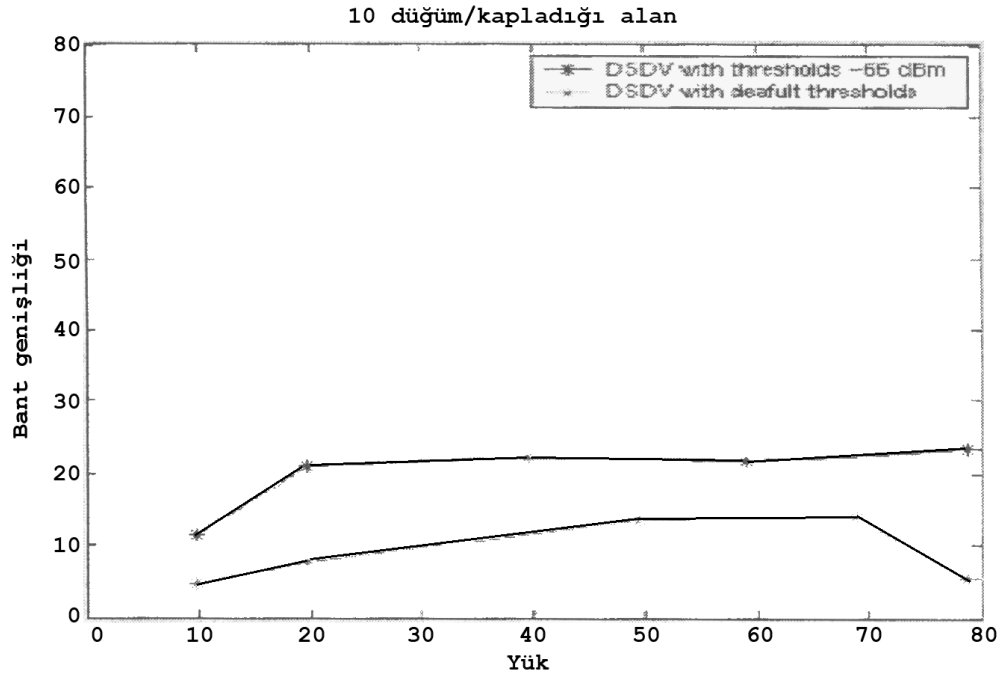
Yapılan simülasyon sonucunda düğüm 3'den düğüm 2'ye ve düğüm 5'den düğüm 4'e diğerlerine göre daha az bant genişliği ile paket akımı sağlandığı anlaşılmaktadır. Şekil 5.7.'ye bakarak daha yoğun kullanılan ağlar ile ağ bant genişliğinin kullanımının önemli bir ölçüde azaldığını söyleyebiliriz.

Bir de daha büyük ağlar için sabit dağılımlı düğüm topolojisi ile NS'de simülasyon yapılmıştır. Ağdaki her bir düğüm kapladığı alan olarak çağrılan (footprint) her bir düğümü saran çember sınır ve onun iletim aralığı içerisinde her yöne iletim yapar. RXThresh\_ ve CStresh\_ -66 dBm ile her bir kaplanılan alan 270m yarıçapındadır. Veri paketleri, her bir düğümde birbirinden bağımsız olarak üretilmiş Poisson varış sürecine göre 1 Mbps hızda iletilen 2300 byte'lık büyüklükte paketlerdir. Poisson paket üretim işlemi için NS'de cbrtraffic.cc dosyasından yararlanılmıştır. Ortalama paket üretim hızı, düğüm başına ağın bant genişliği kadardır.

Simülasyonlar toplam 200 düğüm ile kapladığı alan başına 5'er düğümlük ve toplam 400 düğümlük kapladığı alan başına 10 düğümlük ağ büyüklükleri için çalıştırılmıştır. Her bir simülasyon Poisson trafik üretimi ile yaklaşık 1000 veri paketi üretmek için belirli zaman periyotları ile çalıştırılmıştır. Şekil 5.8. ve Şekil 5.9 çalıştırılan simülasyon sonucu elde edilen grafiklerdir.



Şekil 5.8. Beş düğüm için hazırlanan simülasyonun bant genişliği / yük sonuçları



Şekil 5.9. On düğüm için hazırlanan simülasyonun bant genişliği / yük sonuçları

Yapılan simülasyonlar sonucunda, bu hatalı bloklamadan ötürü ağ bant genişliğinin fark edilebilir ağ yüküne neden olduğu görülmüştür. NS simülasyonunda, bu etkiyle RTS/CTS mekanizmasının ağı dengeleyebileceğinden daha fazla tıkanıklığa yol açabileceği de gösterilmiştir.

## 5.5. Yeniden Yapılandırılan Kablosuz Ağ Modeli

Yukarıda simülasyonların düzgün olarak çalıştırılmasından sonra simülatörde yeniden kablosuz bir ağ modeli tasarlanmıştır (WirelessLan.tcl). Bu ağ modeli tasarısının yapılmasında, NS'de bazı dosyalar düzenlenmiş (Ns-default.tcl, 5.5.4. Mac-802.11.h), bazı konfigürasyon dosyaları yapılmış (Define-option.tcl, Confige-node.tcl) ve birtakım eklemeler getirilmiştir. Bu işlemler aşağıdaki gibi bir hiyerarşiyi takip etmektedir.

Kablosuz bir ağ senaryosu simule edebilmek için, dikkate alınan pek çok faktör vardır. Bunun için hazırlanan ağ topolojisi'nde node\_(0), node\_(1)... node\_(n) gibi düğümler verilmiştir. Bu düğümler bir sınır içersinde tanımlanmıştır. Düğümler arasında UDP bağlantısı kurulmuştur ve paketler, biri diğerinin iletme aralığında olduğu için, kendi aralarında değış-tokuş edilmiştir. Düğümler iletişim aralığının dışına taşındığında, paketler düşmeye başlamaktadır.

### 5.5.1. Define-option.tcl

Tcl script'i kablosuz simulasyon için yazılmıştır. Gezgin bir düğüm, Bağ Katman (Link Layer - LL), Kuyruk Arayüzü (Interface Queue – IFQ), MAC katmanından oluşur. Kablosuz simülasyonun başında, bu ağ bileşenlerinin her biri için bir tip tanımlanmıştır. İlaveten, anten tipi, radyo yayılım modeli, mobil düğümler tarafından kullanılan plansız yönlendirme iletişim kuralı gibi diğere parametreleri tanımlamaya ihtiyaç duyulmuştur.

```
# =====
```

```
# Tanımlama seçenekleri
```

```
# =====
```



```
set val(chan) Channel/WirelessChannel ;# kanal tipi
```

```
set val(prop) Propagation/TwoRayGround;# radyo yayılım modeli
```

```
set val(netif) Phy/WirelessPhy ;# ağ arabirim tipi
```

```
set val(mac) Mac/802_11 ;# MAC tipi
```

```
set val(ifq) Queue/DropTail/PriQueue ;# kuyruk arabirim tipi
```

```
set val(ll) LL ;# bađ katmanı tipi
```

```
set val(ant) Antenna/OmniAntenna ;# anten modeli
```

```
set val(ifqlen) 50 ;# ifq'daki maksimum paket
```

```
set val(nn) 5 ;# düđüm sayısı
```

```
set val(rp) DSDV ;# yönlendirme iletişim kuralı
```

```
# =====
```

Bu alıřtırmada düđüm sayısı 5'dir ama daha büyük ađlar için kapladığı alan başına 5 düđüm, kapladığı alan başına 10 düđüm senaryolarında olduğu gibi daha yüksek sayılar ayarlayabiliriz:

```
set val(nn) 400 ;#düđüm sayısı
```

Bu parametreleri tanımladıktan sonra, örnek simülatör, izleme dosyası ve topoloji ařağıdaki gibi tanımlanmıştır:

```
set ns_ [new Simulator]
```

```
set tracefd [open wireless-masked.tr w]
```

```
$ns_ trace-all $tracefd
```

```
# =====
```

```
# Set up topography object
```

```
# =====
```

```
set topo [new Topography]
```

```
$topo load_flatgrid 3000 3000
```

### **5.5.2. Confige-node.tcl**

3000x3000 metrelik sınırlar içerisindeki düğümlerin dosya büyüklüğünü sınırlamak için yukarıdaki parametrelere göre sadece MacTrace ON ile yapılandırılmıştır. API yapılandırması aşağıda verilmiştir:

```
# =====
```

```
# Düğümlerin Yapılandırması
```

```
# =====
```

```
$ns_ node-config -adhocRouting $val(rp) \
```

```
    -llType $val(ll) \
```

```
    -macType $val(mac) \
```

```
    -ifqType $val(ifq) \
```

```
    -ifqLen $val(ifqlen) \
```

```
    -antType $val(ant) \
```

```
-propType $val(prop) \  
  
-phyType $val(netif) \  
  
-channelType $val(chan) \  
  
-topoInstance $topo \  
  
-agentTrace OFF \  
  
-routerTrace OFF \  
  
-macTrace ON \  
  
-movementTrace OFF
```

### 5.5.3. Ns-default.tcl'deki Düzenlemeler

NS varsayılan parametreleri "../ns-allinone-2.26/ns-2.26/tcl/lib/ns-default.tcl" de bulunmaktadır. Bu dosyadaki varsayılan NS parametreleri aşağıdaki gibidir:

```
# unity gain, omni-diractinal antennas
```

```
# set up the antennas to be centered in the nodes and 1.5 meters above
```

```
Antenna/OrnniAntenna set X_ 0
```

```
Antenna/OmniAntenna set Y_ 0
```

```
Antenna/OrnniAntenna set Z_ 1.5
```

```
Antenna/OmniAntenna set Gt_ 1.0
```

```
Antenna/OmniAntenna set Gr_ 1.0
```

```

# Initialize the SharedMedia interface with parameters to make
# it work like the 914MHz Lucent WaveLAN DSSS radio interface

Phy/WirelessPhy set CPTresh_ 10.0

PhyAVirelessPhy set CSTresh_ 1.559e-11

Phy/WirelessPhy set RXThresh_ 3.652e-10

Phy/WirelessPhy set bandwidth_ 2e6

PhyAVirelessPhy set Pt_ 0.28183815

PhyAVirelessPhy set freq_ 914e+6

Phy/WirelessPhy set L_ 1.0

```

Varsayılan parametresi ile NS simülasyonu, beklenildiği gibi, herhangi gizli bir düğümü göstermemiştir. Hâlbuki CSTresh\_ ve RXThresh\_ eşitlenmesi için düzenlendiğinde RTS-CTS mekanizması olmaksızın gizli düğüm problemi gösterilmiştir. CSTresh\_ = RXThresh\_ = -66 dBm olarak daha gerçekçi değerlerle ve RTS-CTS mekanizmasız ve yukarıdaki gibi basit üç düğümlü senaryo simule edilmiştir. Eşik değerler aşağıdaki tel dosyasında ayarlanmıştır:

```

# =====

# Set -66 dBm of both RX and CS Threshold

# =====

Phy/WirelessPhy set CSTresh_ 2.5e-10

```

*Phy/WirelessPhy set RXThresh\_ 2.5e-10*

#### **5.5.4. Mac-802.11.h'deki Düzenlemeler**

NS bütün iletişimler için RTS-CTS tokalaşma uygulamıştır. Halbuki tokalaşma "../ns-allinone-2.26/ns-2.26/mac/mac-802\_11.h" de MAC\_RTSThreshold 3 000 byte gibi yüksek bir değere ki bu 3 000 byte'ın altındaki herhangi veri paketi iletimi için RTS gönderilmeyecek anlamına gelir ayarlanarak kapatılmıştır ve o zaman daha düşük paket büyüklüğünde paket iletimi simule edilir.

```

##define MAC_RTSThreshold          3000      // bytes

#define MAC_RTSThreshold           0         // bytes

#define MAC_ShortRetryLimit        7         // retransmissions

#define MAC_LongRetryLimit         4         // retransmissions

#define MAC_FragmentationThreshold 2346     // bytes

#define MAC_MaxTransmitMSDULifetime 512     // time units

#define MAC_MaxReceiveLifetime     512     // time units

```

#### **5.5.5. WirelessLan.tcl**

Bu tezin ana tcl scriptidir ve bazı diğer kodları da barındırır. İlk olarak NS için tanımlama seçenekleri gereklidir.

```

# =====
# Tanımlama seçenekleri
# =====

set val(chan) Channel/WirelessChannel ;# kanal tipi

```

```
set val(prop) Propagation/TwoRayGround;# radyo yayılım modeli
```

```
set val(netif) Phy/WirelessPhy ;# ağ arabirim tipi
```

```
set val(mac) Mac/802_11 ;# MAC tipi
```

```
set val(ifq) Queue/DropTail/PriQueue ;# kuyruk arabirim tipi
```

```
set val(ll) LL ;# bađ katmanı tipi
```

```
set val(ant) Antenna/OmniAntenna ;# anten modeli
```

```
set val(ifqlen) 50 ;# ifq'daki maksimum paket
```

```
set val(nn) 5 ;# düđüm sayısı
```

```
set val(rp) DSDV ;# yönlendirme iletişim kuralı
```

```
# =====
```

Aşağıdaki kodda yeni bir NS simülatör örneđi yaratılmış ve bazı başlangıç kontrolleri yapılmıştır:

```
# rasgele besleme ve sınır parametreleri için kontrol
```

```
if { $opt(x) == 0 || $opt(y) == 0 } {
```

```
    puts "Kablosuz topoloji için x ve y degerleri verilmemistir\n"
```

```
}
```

```
if {$opt(seed) > 0} {
```

```
    puts "$opt(seed) ile rasgele sayı dizisi besleme\n"
```

```
ns-random $opt(seed)

}

# simülâtör örneđi yaratma

set ns_ [new Simulator]

# hiyerarşik yönlendirmeyi belirleme

$ns_ node-config -addressType hierarchical

AddrParams set domain_num_ 3           ;# alan (domains) sayısı

lappend cluster_num 2 1 1             ;# her bir alandaki grup sayısı

AddrParams set cluster_num_ $cluster_num

lappend eilastlevel 1 1 2 1          ;# her bir grupdaki düğüm sayısı

AddrParams set nodes_num_ $eilastlevel ;# her bir alanın

# topografî nesneleri yaratma

set topo [new Topography]

# topolojiyi tanımlama

$topo load_flatgrid $opt(x) $opt(y)

# God create

# 2 for HA and FA
```

```

create-god [expr $opt(nn) + 2]

# kablolu düğüm yaratma

set temp {0.0.0 0.1.0}      ;# hiyerarşik adresler

for {set i 0} {$i < $num_wired nodes} {incr i} {

    set W($i) [$ns_node [lindex $temp $i]]

}

# =====

# Düğümleri Yapılandırma

# =====

$ns_node-config -adHocRouting $val(rp) \

    -llType $val(ll) \

    -macType $val(mac) \

    -ifqType $val(ifq) \

    -ifqLen $val(ifqlen) \

    -antType $val(ant) \

    -propType $val(prop) \

    -phyType $val(netif) \

```



```
-channelType $val(chan) \
```

```
-topoInstance $topo \
```

```
-agentTrace OFF \
```

```
-routerTrace OFF \
```

```
-macTrace ON \
```

```
-movementTrace OFF
```

Simülasyonda, herhangi bir hareketi ya da rassal bir hareketi kapatmak için düğümler hareketliliği göstermemiştir. Düğümler başlangıç pozisyonunda verilmiştir ve simülasyon bant genişliğini korurlar.

Beş düğümlü senaryoda düğümler aşağıdaki gibi ayarlanmıştır:

```
for {set i 0} {$i < $val(nn) } {incr i} {

    set node_($i) [$ns_ node]

    $node_($i) random-motion 0 ;# rassal hareketi önleme

}

# =====

# Düğümler için başlangıç koordinatlarını üretme (X,Y, Z=0)

# =====

$node_(0) set X_ 50.0 ;# düğüm 0
```

*\$node\_(0) set Y\_ 200.0*

*\$node\_(0) set Z\_ 0.0*

*\$node\_(1) set X\_ 250.0 ;# düğüm 1*

*\$node\_(1) set Y\_ 200.0*

*\$node\_(1) set Z\_ 0.0*

*\$node\_(2) set X\_ 450.0 ;# düğüm 2*

*\$node\_(2) set Y\_ 200.0*

*\$node\_(2) set Z\_ 0.0*

*\$node\_(3) set X\_ 650.0 ;# düğüm 3*

*\$node\_(3) set Y\_ 200.0*

*\$node\_(3) set Z\_ 0.0*

*\$node\_(4) set X\_ 850.0 ;# düğüm 4*

*\$node\_(4) set Y\_ 200.0*

*\$node\_(4) set Z\_ 0.0*

*# =====*

*# Beş düğümün yerleştirilmesi*

*# =====*

```
$ns_ at 0.1 "$node_(0) setdest 50.0 200.0 0.0"
```

```
$ns_ at 0.1 "$node_(1) setdest 250.0 200.0 0.0"
```

```
$ns_ at 0.1 "$node_(2) setdest 450.0 200.0 0.0"
```

```
$ns_ at 0.1 "$node_(3) setdest 650.0 200.0 0.0"
```

```
$ns_ at 0.1 "$node_(4) setdest 850.0 200.0 0.0"
```

Halbuki büyük ağlar için, yukarıdaki yolla her bir düğümü ayarlama zahmetli bir iştir. Bu tezde, cbrgen.tcl dosyası kullanılarak sabit dağılımlı bir düğüm topolojisi üretildi ve çıkış düğüm koordinatlar dosyası ve NS simülasyonuna giriş olarak komşu düğüm dosyası (adjacency node) üretilmiştir. NS düğüm koordinatları ile dosyayı okur ve NS topolojisinde düğümleri ayarlamak için kullanılan x ve y dizilerini günceller.

```
set nodefd [open nodes.dat r]
```

```
set count 0
```

```
while {[gets $nodefd line] >= 0} {
```

```
    set x_value [lindex $line 0]
```

```
    set y_value [lindex $line 1]
```

```
    set x_array($count) [expr $x_value * 1000]
```

```
    set y_array($count) [expr $y_value * 1000]
```

```
    incr count
```

```

}

close $nodefd

for {set i 0} {$i < $val(nn)} {incr i} {

    $node_($i) set X_ $x_array($i) ;# X'i ayarlar

    $node_($i) set Y_ $y_array($i) ;# Y'yi ayarlar

    $node_($i) set Z_ 0.0 ;# Z'yi ayarlar

}

for {set i 0} {$i < $val(nn)} {incr i} {

    $ns_ at 0.1 "$node_($i) setdest $x_array($i) $y_array($i) 0.0"

}

Düğüm1er arasındaki trafik akışı, CBR trafikli (Poisson trafik kullanımı ile
değiştirildi) UDP üstlenici (agent) olan kaynaklar ile sağlanmıştır. Alıcı, Sink
üstlenici olarak ayarlanmıştır. Düğüm 0 ve düğüm 1 arasındaki trafik akışı aşağıdaki
gibidir:

# =====

# Düğüm1er arasında trafik akışı sağlama

# =====

set udp0 [new Agent/UDP]

$udp0 set class_ 1

```

*set udp2 [new Agent/UDP]*

*\$udp2 set class\_ 2*

*set udp3 [new Agent/UDP]*

*\$udp3 set class\_ 3*

*set cbr0 [new Application/Traffic/CBR]*

*\$cbr0 set packetSize\_ \$size\_*

*\$cbr0 set interval\_ \$interarrival\_*

*\$cbr0 attach-agent \$udp0*

*set cbr2 [new Application/Traffic/CBR]*

*\$cbr2 set packetSize\_ \$size\_*

*\$cbr2 set interval\_ \$interarrival\_*

*\$cbr2 attach-agent \$udp2*

*set cbr3 [new Application/Traffic/CBR]*

*\$cbr3 set packetSize\_ \$size\_*

*\$cbr3 set interval \$interarrival\_*

*\$cbr3 attach-agent \$udp3*

*set sink1 [new Agent/Null]*

*set sink3 [new Agent/Null]*

*set sink4 [new Agent/Null]*

*\$ns\_ attach-agent \$node\_(0) \$udp0*

*\$ns\_ attach-agent \$node\_(2) \$udp2*

*\$ns\_ attach-agent \$node\_(3) \$udp3*

*\$ns\_ attach-agent \$node\_(1) \$sink1*

*\$ns\_ attach-agent \$node\_(3) \$sink3*

*\$ns\_ attach-agent \$node\_(4) \$sink4*

*\$ns\_ connect \$udp0 \$sink1*

*\$ns\_ connect \$udp2 \$sink3*

*\$ns\_ connect \$udp3 \$sink4*

*\$ns\_ at 0.200 "\$cbr3 start"*

*\$ns\_ at 0.212 "\$cbr0 start"*

*\$ns\_ at 0.224 "\$cbr2 start"*

Sonda, düğümler kendilerini ilk duruma getirmek istediğinde, durma zamanı tanımlanmıştır. Stop{} prosedürü izlemeyi yazmak ve izleme dosyasını kapatmak olarak kullanılmıştır ve nihayetinde simülasyon çalıştırılmıştır.

*\$ns\_ at \$stoptime\_ "stop"*

```
proc stop {} {
```

```
    global ns_ tracefd
```

```
    $ns_ flush-trace
```

```
    close $tracefd
```

```
    exit 0
```

```
}
```

```
puts "Simülasyon başlatılıyor..."
```

```
$ns_ run
```

Bununla birlikte büyük ağlar için trafik akışı bir döngüde dizin kullanılarak ve komşu listesi referans verilerek ayarlanmıştır.

```
# =====
```

```
# Komşuluk listesi dosyasını okuma ve komşuluk listesi dizinini yükleme
```

```
# =====
```

```
set neighborfd [open adjfile.dat r]
```

```
set k 0
```

```
while {[gets $neighborfd line] >= 0} {
```

```
    set neighborlistl($k) [split $line {}]
```

```
    set neighborlistl($k) [join $neighborlistl($k) {}]
```

```

for {set m 0} {$m < [llength $neighborlist($k)]} {incr m} {

    set neighbor [lindex $neighborlist($k) $m]

    if {$neighbor == 0 && $m == 0} {

        lappend neighborlist($k) -1

    } elseif {$neighbor > 0} {

        lappend neighborlist($k) $neighbor

    } else {

        break

    }

}

set neighborlistlength($k) [llength $neighborlist($k)]

incr k

}

close $neighborfd

# =====
# D ğ mler arasındaki trafik akışını ayarlamak. Komşu d ğ m listesi kontrol ve
# sırayla cbr trafik akışını kurmak ve trafięi bařlatmak.
# =====

```



```

Agent/UDP set packetSize_ 3000

for {set i 0} {$i < $val(nn) } {incr i} {

    for {set j 0} {$j < $neighborlistlength($i) } {incr j} {

        set neighbor [lindex $neighborlist($i) $j]

        if {$neighbor != -1} {

            set neighbor [expr $neighbor -1]

            set interarrival [expr $neighborlistlength($i) / $lambda]

            set udp($i,$neighbor) [new Agent/UDP]

            $udp($i,$neighbor) set class_ 1

            $ns_ attach-agent $node_($i) $udp($i,$neighbor)

            set cbr($i,$neighbor) [new Application/Traffic/CBR]

            $cbr($i,$neighbor) set packetSize_ 2200

            $cbr($i,$neighbor) set interval_ $interarrival

            $cbr($i,$neighbor) attach-agent $udp($i,$neighbor)

            $ns_ connect $udp($i,$neighbor) $sink($neighbor)

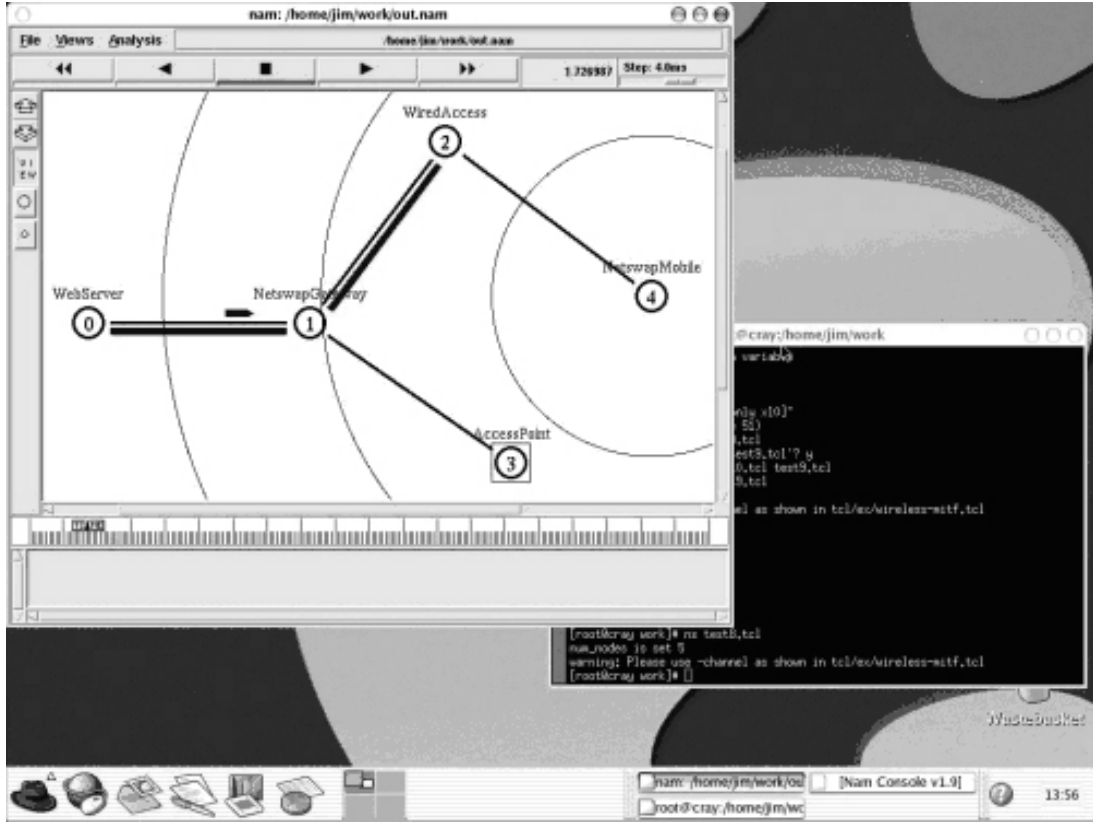
            $ns_ at 0.2 "$cbr($i,$neighbor) start"

        }

    }

}

```



Şekil 5.10. MANET destekli bir ağ örneği

Wirelesslan.tcl tasarlanırken öncelikle kanal tipi, radyo yayılım modeli, ağ arabirim tipi, bağ katmanı tipi, kuyruk arabirim tipi, anten modeli, maksimum paket sayısı, düğüm sayısı, yönlendirme iletişim kurallarını belirleyen tip tanımlamaları yapılarak, NS simülöründe yapılan değişikliklerin (mac-802.11.h, ns-default.tcl) simülörün çalışmasını değiştirip değiştirmediğini görmek hedeflenmiştir.

Düğümlemler arasında bağlantı kurulurken trafik deseni olarak CBR veri tipi kullanılmış, gönderici düğüm de UDP üstlenici, alıcı düğümde Sink üstlenici tanımlanarak iki düğüm arasında veri alış verişi sağlanmıştır. Simülasyon boyunca düğümlemlerin rassal hareketi önlenmiş ve simülasyon sonlandırılırken düğümlemler ilk konumlarına geri getirilmiştir.

Simülasyonun çalışmasının güvenilirliği açısından 3000x3000 metre gibi büyük bir topoloji sınırı belirlenmiştir. Bu topoloji sınırı farklı boyutlarda alınabilmektedir, ancak çalışmanın daha büyük alanlarda etkisinin görülmesi için bu değerler

kullanılmıştır. Simülatörde 5 ve 10 düğümlük olmak üzere iki senaryo uygulanmıştır. Değiştirilen parametreleri ile hazırlanan simülasyonlar, kablosuz ağlarda veri kaybına neden olan problemleri başarılı bir şekilde göstermiştir. Sonuç olarak simülasyon herhangi bir hata vermeden başarılı bir şekilde yürütülmüştür.

### **5.6. TCP ve UDP Üstlencilerde Karşılaştırmalı Model**

Bu kısımda, TCP üstlenici kullanılarak üzerinde ftp trafiği ve UDP üstlenici kullanılarak üzerinde cbr trafiği oluşturulmuş iki ayrı simülasyondan bahsedilmektedir. Bu simülasyonlar hazırlanarak gerçek ortamdaki gibi davranması istenilen NS2'nin UDP gibi basit ve TCP gibi daha güvenilir olan iki protokolde de denenmesi sağlanmıştır. Birincisinde paketleri gönderen TCP üstlenici için paketlerin ulaşabileceği TCPSink üstlenici, ikincisinde ise UDP üstlenicinin paketlerini gönderdiği Null üstlenici kurulmuştur. Oluşturulan bu üstleniciler için hazırlanan kablosuz arabirim, gerçek ortama çok yakın olarak ayarlanmış, NCR ve AT&T firmalarının 1990'larda ürettiği 914 Mhz'de frekans yayan 802.11 WaveLan kartlar gibi davranan donanımsal birimlere benzetim sağlanmıştır. Bu şekilde, her üstlenici üzerinde de 3 ayrı uygulama olacak şekilde kablosuz plansız bir ağ simülasyonu oluşturulmuş ve NS2'nin varsayılan hali ile gerçek ortamdaki gibi düzenlenilmiş hali karşılaştırılmıştır.

Bütün simülasyonlar 500x500m'lik topolijiyeye sahip bir alanda 150 saniye boyunca izlenmiş ve bu alanda düğümlerin iletişimleri sağlanırken kuyruğa maksimum 50 adet paket gönderilmiştir. Simülasyonlarda birinci düğüm (5, 2) konumunda iken 10. sn'de 1m/s hızla (20, 18) konumuna, diğer düğüm 50. sn'de (390, 385) konumunda iken (25, 20) konumuna 15m/s hızla hareket ederek birbirlerine yaklaştırılmıştır. Simülasyonun 100. sn'de ikinci düğüm (490, 480) konumuna 15m/s hızla geçerek diğer düğümden uzaklaştırılmıştır. Böylece düğümlerin her iki yöne de hareketi sağlanmıştır. En sonunda ise düğümler tekrar kendi konumlarına döndürülmüştür. Yapılan uygulamalar sonucunda aşağıdaki gibi bulgular elde edilmiştir.

a) TCP Simülasyonunda CStresh\_ ve RxThresh\_ aynı değerde ve RTSThreshold 0 iken;

Bu şekilde hazırlanan simülasyonda iki düğüm iletişim halindeyken; 11960 byte tcp paketi, 1020 byte ack paketi ve 88 byte RTS paketinin düştüğü gözlenmiştir. Tcp paketleri NRTE (DROP\_RTR\_NO\_ROUTE), CBK (DROP\_RTR\_MAC\_CALLBACK) ve RET (DROP\_MAC\_RETRY\_COUNT\_EXCEEDED) nedeni ile düşmüştür. NRTE ve CBK hataları birbirine yakın hatalar olup, paketin ulaşacağı yerde yönlendiricinin devreye girmemesinden ya da yönlendiricideki diğer nedenlerden kaynaklı paket düşmeleridir. Bu, paketin ulaştırılacağı hedefin bulunamadığı anlamına da gelmektedir. RET'den kaynaklı paket düşmesi ise RTS paketinin ya da veri paketinin karşı tarafa tekrar gönderilirken ulaşmamasından kaynaklı paket düşmesidir. Bu tekrar NS2'nin varsayılan halinde yedi kere ile sınırlıdır. Yediden sonra paket tekrar gönderilmemekte ve otomatikman düşmektedir. Ack paketi CBK ve RET nedeni ile düşmüştür. Son olarak RTS paketinin ise RET nedeni ile düştüğü görülmektedir. Şekil 5.11'de bu paket kayıpları ile ilgili izleme dosyasından bir kesit gösterilmiştir.

6	D	29_0	RTR	NRTE	0 tcp	60 [0
7	D	29_0	RTR	NRTE	1 tcp	60 [0
8	D	29_0	RTR	NRTE	2 tcp	60 [0
14	D	58_0	RTR	NRTE	3 tcp	60 [0
15	D	58_0	RTR	NRTE	4 tcp	60 [0
19523	D	118.05_1	MAC	RET	0 RTS	44 [62e
19524	D	118.05_1	RTR	CBK	2408 ack	60 [13a
19525	D	118.05_1	RTR	CBK	2413 ack	60 [0
19526	D	118.05_1	RTR	CBK	2414 ack	60 [0
19527	D	118.05_1	RTR	CBK	2415 ack	60 [0
19528	D	118.05_1	RTR	CBK	2417 ack	60 [0
19529	D	118.05_1	RTR	CBK	2418 ack	60 [0
19530	D	118.05_1	RTR	CBK	2420 ack	60 [0
19531	D	118.05_1	RTR	CBK	2421 ack	60 [0
19532	D	118.05_1	RTR	CBK	2425 ack	60 [0
19533	D	118.05_1	RTR	CBK	2429 ack	60 [0
19534	D	118.05_1	RTR	CBK	2432 ack	60 [0
19535	D	118.05_1	RTR	CBK	2434 ack	60 [0
19536	D	118.05_1	RTR	CBK	2435 ack	60 [0
19537	D	118.05_1	RTR	CBK	2436 ack	60 [0
19538	D	118.05_1	RTR	CBK	2438 ack	60 [0
19539	D	118.05_1	RTR	CBK	2440 ack	60 [0
19540	D	118.05_1	MAC	RET	2408 ack	60 [13a
19543	D	118.05_0	MAC	RET	0 RTS	44 [256e
19544	D	118.05_0	RTR	CBK	2430 tcp	1060 [13a
19545	D	118.05_0	RTR	CBK	2431 tcp	1060 [0
19546	D	118.05_0	RTR	CBK	2433 tcp	1060 [0
19547	D	118.05_0	RTR	CBK	2437 tcp	1060 [0
19548	D	118.05_0	RTR	CBK	2439 tcp	1060 [0

Şekil 5.11. CSTHresh\_ = RXThresh\_, RTSTHreshold\_ 0 ile TCP Simülasyonu

b) TCP Simülasyonunda CStresh\_ ve RxThresh\_ aynı değerde ve RTSThreshold 3000 iken;

Bu şekilde hazırlanan simülasyonda; 9840 byte tcp paketi ve 1140 byte ack paketinin düştüğü gözlenmiştir. Tcp paketleri NRTE, CBK ve RET nedeni ile düşmüştür. Ack paketi ise CBK ve RET nedeni ile düşmüştür. Şekil 5.12'de izleme dosyasından bir kesit gösterilmiştir.

7	D	29	0	RTR	NRTE	1 tcp	60	[0]
8	D	29	0	RTR	NRTE	2 tcp	60	[0]
14	D	58	0	RTR	NRTE	3 tcp	60	[0]
15	D	58	0	RTR	NRTE	4 tcp	60	[0]
10554	D	118.02	1	RTR	CBK	2578 ack	60	[13a]
10555	D	118.02	1	RTR	CBK	2581 ack	60	[0]
10556	D	118.02	1	RTR	CBK	2583 ack	60	[0]
10557	D	118.02	1	RTR	CBK	2586 ack	60	[0]
10558	D	118.02	1	RTR	CBK	2588 ack	60	[0]
10559	D	118.02	1	RTR	CBK	2589 ack	60	[0]
10560	D	118.02	1	RTR	CBK	2591 ack	60	[0]
10561	D	118.02	1	RTR	CBK	2592 ack	60	[0]
10562	D	118.02	1	RTR	CBK	2594 ack	60	[0]
10563	D	118.02	1	RTR	CBK	2599 ack	60	[0]
10564	D	118.02	1	RTR	CBK	2601 ack	60	[0]
10565	D	118.02	1	RTR	CBK	2602 ack	60	[0]
10566	D	118.02	1	RTR	CBK	2606 ack	60	[0]
10567	D	118.02	1	RTR	CBK	2608 ack	60	[0]
10568	D	118.02	1	RTR	CBK	2610 ack	60	[0]
10569	D	118.02	1	RTR	CBK	2611 ack	60	[0]
10570	D	118.02	1	RTR	CBK	2613 ack	60	[0]
10571	D	118.02	1	RTR	CBK	2614 ack	60	[0]
10572	D	118.02	1	MAC	RET	2578 ack	60	[13a]
10577	D	118.07	0	RTR	CBK	2607 tcp	1060	[13a]
10578	D	118.07	0	RTR	CBK	2609 tcp	1060	[0]
10579	D	118.07	0	RTR	CBK	2612 tcp	1060	[0]
10580	D	118.07	0	MAC	RET	2607 tcp	1060	[13a]
10589	D	123.22	0	RTR	NRTE	2615 tcp	1060	[0]

Şekil 5.12. CStresh\_ = RXThresh\_, RTSThreshold\_ 3000 ile TCP Simülasyonu

c) TCP için NS2'nin varsayılan durumunda yapılan denemede;

10900 byte tcp paketi ve 1080 byte ack paketi kaybı olmuştur. Tcp için NRTE, CBK ve RET; ack paketi için ise CBK ve RET'den kaynaklanmıştır. Şekil 5.13'de izleme dosyasından bir kesit gösterilmiştir.

6	D	29	0	RTR	NRTE	0	tcp	60	[0
7	D	29	0	RTR	NRTE	1	tcp	60	[0
8	D	29	0	RTR	NRTE	2	tcp	60	[0
14	D	58	0	RTR	NRTE	3	tcp	60	[0
15	D	58	0	RTR	NRTE	4	tcp	60	[0
9386	D	116.41	1	RTR	CBK	2295	ack	60	[13a
9387	D	116.41	1	RTR	CBK	2296	ack	60	[0
9388	D	116.41	1	RTR	CBK	2298	ack	60	[0
9389	D	116.41	1	RTR	CBK	2299	ack	60	[0
9390	D	116.41	1	RTR	CBK	2301	ack	60	[0
9391	D	116.41	1	RTR	CBK	2302	ack	60	[0
9392	D	116.41	1	RTR	CBK	2303	ack	60	[0
9393	D	116.41	1	RTR	CBK	2306	ack	60	[0
9394	D	116.41	1	RTR	CBK	2307	ack	60	[0
9395	D	116.41	1	RTR	CBK	2310	ack	60	[0
9396	D	116.41	1	RTR	CBK	2312	ack	60	[0
9397	D	116.41	1	RTR	CBK	2314	ack	60	[0
9398	D	116.41	1	RTR	CBK	2315	ack	60	[0
9399	D	116.41	1	RTR	CBK	2318	ack	60	[0
9400	D	116.41	1	RTR	CBK	2322	ack	60	[0
9401	D	116.41	1	RTR	CBK	2323	ack	60	[0
9402	D	116.41	1	RTR	CBK	2325	ack	60	[0
9403	D	116.41	1	MAC	RET	2295	ack	60	[13a
9406	D	116.44	0	RTR	CBK	2319	tcp	1060	[13a
9407	D	116.44	0	RTR	CBK	2320	tcp	1060	[0
9408	D	116.44	0	RTR	CBK	2321	tcp	1060	[0
9409	D	116.44	0	RTR	CBK	2324	tcp	1060	[0
9410	D	116.44	0	MAC	RET	2319	tcp	1060	[13a

Şekil 5.13. NS2'nin varsayılan durumu ile TCP Simülasyonu

d) UDP Simülasyonunda CStresh\_ ve RxThresh\_ aynı değerde ve RTSThreshold 0 iken;

Bu şekilde yapılan simülasyonda 5529420 byte'lık cbr paket düşmesi ve 44 byte'lık RTS paket kaybı olmuştur. Cbr paketi kaybı IFQ (DROP\_IFQ\_QFULL), CBK, NRTE ve RET'den kaynaklanmaktadır. IFQ hatası, kuyrukta kuyruğun alabileceğinden fazla paket birikimden kaynaklanan paket kaybı hatasıdır. RTS paket kaybına ise RET neden olmuştur. Şekil 5.14'de izleme dosyasından bir kesit gösterilmiştir.

3688	D	67.25	0	RTR	IFQ	3674	cbr	1020	[0
3689	D	67.27	0	RTR	IFQ	3675	cbr	1020	[0
3690	D	67.29	0	RTR	IFQ	3676	cbr	1020	[0
3691	D	67.3	0	RTR	IFQ	3677	cbr	1020	[0
3692	D	67.32	0	RTR	IFQ	3678	cbr	1020	[0
3693	D	67.34	0	RTR	IFQ	3679	cbr	1020	[0
3694	D	67.36	0	RTR	IFQ	3680	cbr	1020	[0
3695	D	67.38	0	RTR	IFQ	3681	cbr	1020	[0
3696	D	67.39	0	RTR	IFQ	3682	cbr	1020	[0
3697	D	67.41	0	RTR	IFQ	3683	cbr	1020	[0
26894	D	118.03	0	MAC	RET	0	RTS	44	[242e
26895	D	118.03	0	RTR	CBK	6580	cbr	1020	[13a
26896	D	118.03	0	RTR	CBK	6581	cbr	1020	[0
26897	D	118.03	0	MAC	RET	6580	cbr	1020	[13a
26904	D	119.18	0	RTR	IFQ	6582	cbr	1020	[0
26905	D	119.2	0	RTR	IFQ	6583	cbr	1020	[0
26906	D	119.21	0	RTR	IFQ	6584	cbr	1020	[0
26907	D	119.23	0	RTR	IFQ	6585	cbr	1020	[0
26908	D	119.25	0	RTR	IFQ	6586	cbr	1020	[0
26909	D	119.27	0	RTR	IFQ	6587	cbr	1020	[0
26910	D	119.29	0	RTR	IFQ	6588	cbr	1020	[0
26911	D	119.3	0	RTR	IFQ	6589	cbr	1020	[0
26913	D	119.32	0	RTR	IFQ	6590	cbr	1020	[0
26914	D	119.34	0	RTR	IFQ	6591	cbr	1020	[0
26915	D	119.36	0	RTR	IFQ	6592	cbr	1020	[0
26916	D	119.38	0	RTR	IFQ	6593	cbr	1020	[0
26917	D	119.39	0	RTR	IFQ	6594	cbr	1020	[0
26918	D	119.41	0	RTR	IFQ	6595	cbr	1020	[0
26919	D	119.43	0	RTR	IFQ	6596	cbr	1020	[0

Şekil 5.14. CSTHresh\_ = RXThresh\_, RTSTHreshold\_ 0 ile UDP Simülasyonu

e) UDP Simülasyonunda CSTHresh\_ ve RxThresh\_ aynı değerde ve RTSTHreshold 3000 iken;

Sadece 5522280 byte'lık cbr veri kaybı olmuştur ve bu paket kaybına IFQ, CBK, NRTE ve RET sebebiyet vermiştir. Şekil 5.15'de izleme dosyasından bir kesit gösterilmiştir.

3692	D	67.32	0	RTR	IFQ	3678	cbr	1020	[0
3693	D	67.34	0	RTR	IFQ	3679	cbr	1020	[0
3694	D	67.36	0	RTR	IFQ	3680	cbr	1020	[0
3695	D	67.38	0	RTR	IFQ	3681	cbr	1020	[0
3696	D	67.39	0	RTR	IFQ	3682	cbr	1020	[0
3697	D	67.41	0	RTR	IFQ	3683	cbr	1020	[0
15302	D	118.1	0	RTR	CBK	6580	cbr	1020	[13a
15303	D	118.1	0	RTR	CBK	6581	cbr	1020	[0
15304	D	118.1	0	RTR	CBK	6582	cbr	1020	[0
15305	D	118.1	0	RTR	CBK	6583	cbr	1020	[0
15306	D	118.1	0	RTR	CBK	6584	cbr	1020	[0
15307	D	118.1	0	RTR	CBK	6585	cbr	1020	[0
15308	D	118.1	0	MAC	RET	6580	cbr	1020	[13a
15316	D	119.25	0	RTR	IFQ	6586	cbr	1020	[0
15317	D	119.27	0	RTR	IFQ	6587	cbr	1020	[0
15318	D	119.29	0	RTR	IFQ	6588	cbr	1020	[0
15319	D	119.3	0	RTR	IFQ	6589	cbr	1020	[0
15320	D	119.32	0	RTR	IFQ	6590	cbr	1020	[0
15321	D	119.34	0	RTR	IFQ	6591	cbr	1020	[0
15322	D	119.36	0	RTR	IFQ	6592	cbr	1020	[0
15323	D	119.38	0	RTR	IFQ	6593	cbr	1020	[0
15324	D	119.39	0	RTR	IFQ	6594	cbr	1020	[0
15325	D	119.41	0	RTR	IFQ	6595	cbr	1020	[0
15326	D	119.43	0	RTR	IFQ	6596	cbr	1020	[0
15327	D	119.45	0	RTR	IFQ	6597	cbr	1020	[0
15328	D	119.46	0	RTR	IFQ	6598	cbr	1020	[0

Şekil 5.15. CSTHresh\_ = RXThresh\_, RTSTHreshold\_ 3000 ile UDP Simülasyonu

f) UDP için NS2'nin varsayılan durumunda yapılan denemede;

Yapılan bu simülasyonda 6601440 byte'lık cbr paketinin düştüğü gözlenmiştir ve bir önceki simülasyonda da olduğu gibi kayba IFQ, CBK, NRTE ve RET'in neden olarak gösterildiği gözlenmiştir. Şekil 5.16.'da izleme dosyasından bir kesit gösterilmiştir.



12036	D	116.42	0	RTR	CBK	6487	cbr	1020	13a
12037	D	116.42	0	RTR	CBK	6488	cbr	1020	0
12038	D	116.42	0	RTR	CBK	6489	cbr	1020	0
12039	D	116.42	0	RTR	CBK	6490	cbr	1020	0
12040	D	116.42	0	RTR	CBK	6491	cbr	1020	0
12041	D	116.42	0	MAC	RET	6487	cbr	1020	13a
12049	D	117.57	0	RTR	IFQ	6492	cbr	1020	0
12050	D	117.59	0	RTR	IFQ	6493	cbr	1020	0
12051	D	117.61	0	RTR	IFQ	6494	cbr	1020	0
12052	D	117.63	0	RTR	IFQ	6495	cbr	1020	0
12053	D	117.64	0	RTR	IFQ	6496	cbr	1020	0
12054	D	117.66	0	RTR	IFQ	6497	cbr	1020	0
12055	D	117.68	0	RTR	IFQ	6498	cbr	1020	0
12056	D	117.7	0	RTR	IFQ	6499	cbr	1020	0
12057	D	117.71	0	RTR	IFQ	6500	cbr	1020	0
12058	D	117.73	0	RTR	IFQ	6501	cbr	1020	0
12059	D	117.75	0	RTR	IFQ	6502	cbr	1020	0
12060	D	117.77	0	RTR	IFQ	6503	cbr	1020	0
12061	D	117.79	0	RTR	IFQ	6504	cbr	1020	0
12062	D	117.8	0	RTR	IFQ	6505	cbr	1020	0
12063	D	117.82	0	RTR	IFQ	6506	cbr	1020	0
12064	D	117.84	0	RTR	IFQ	6507	cbr	1020	0
12065	D	117.86	0	RTR	IFQ	6508	cbr	1020	0
12066	D	117.88	0	RTR	IFQ	6509	cbr	1020	0
12067	D	117.89	0	RTR	IFQ	6510	cbr	1020	0
12068	D	117.91	0	RTR	IFQ	6511	cbr	1020	0
12069	D	117.93	0	RTR	IFQ	6512	cbr	1020	0

Şekil 5.16. NS2'nin varsayılan durumu ile UDP Simülasyonu

Bu karşılaştırmalı modelde veri kayıplarına 3 durumda bakılmış i) CStresh\_ = RXThresh\_ ve RTSThreshold = 0 ii) CStresh\_ = RXThresh\_ ve RTSThreshold = 3000, iii) NS Simülatörünün varsayılan hali ve aşağıdaki gibi sonuç elde edilmiştir:

	I. Durum (byte)	II. Durum (byte)	III. Durum (byte)
<b>TCP (ftp)</b>	13068	10980	11980
<b>UDP (cbr)</b>	5529464	5522280	6601440

Her iki üstleniciye bakıldığında gerçek ortamdaki gibi TCP üstlenici, UDP üstleniciye göre paketleri daha kontrollü olarak taşıdığı için gerçek ortamda olması gerektiği gibi bu üstlenici de veri kaybının daha az olduğu görülmektedir.

## 6. SONUÇ ve ÖNERİLER

Bu bölümde, çalışma kapsamında yapılan simülasyonlar; Gizli Düğüm simülasyonu, Örtülü Düğüm simülasyonu, RTS/CTS mekanizma simülasyonu, Bloklama Etkisi simülasyonundan elde edilen sonuçlara, bu sonuçlara dayanarak hazırlanan kablosuz bir ağ modeli simülasyonunun gerçek bir ortamdaki gibi nasıl tasarlanması gerektiğine yer verilmiş ve bu alanda bilime olan katkısı aktarılmıştır.

### 6.1. Sonuçlar

Bu tezde kablosuz ağlarda NS simülasyonu boyunca simülatörün varsayılan parametrelerini kullanarak ve daha gerçekçi parametrelerle yeniden modellenmesi sağlanarak iki ayrı durum üzerinde gizli düğüm, örtülü düğüm ve bloklama etkisi çalışılmış ve simülasyon ortamında veri kayıplarının fark edilebilirliğine yönelik aşağıdaki sonuçlar elde edilmiştir:

- Gizli düğüm simülasyonu bölüm 5.1. de görüleceği üzere çalıştırılmış ve NS2’de çalıştırılan, varsayılan taşıyıcı algılama aralığı, alım aralığının 2.2 katı olan simülasyonda, gizli düğüm probleminin varsayılan bu değerlerle ile gerçekteki gibi gözlenilemeyeceği tespit edilmiştir.
- Kablosuz iletişimin yapıldığı gerçek bir ortamda, İki Işınlı Yer Yansımali Radyo Yayılım modelinin, düğümler arasındaki yayılım için yansıma yolunu ve görüş hattını dikkate aldığı ve Serbest Uzay modelinden daha kesin bir mesafe tahmini verdiği bilinmektedir. Bu çalışma kapsamında çalıştırılan gizli düğüm, örtülü düğüm, bloklama, RTS/CTS mekanizması simülasyonları’nda da İki Işınlı Yer Yansımali model kullanılmış ve uzun mesafelerde (670 x 670, 1500 x 1500 ve 3000 x 3000 metre) kablosuz iletim yapılması sağlanmıştır. Farklı boyutlarda çalışılan bu simülasyonlardan elde edilen sonuçlarda simülasyon süresinin mesafe ile doğru orantılı olarak arttığı tespit edilmiş ancak veri kaybına yönelik değişiklik gözlenmemiştir.

- Gizli düğüm problemini simülatör ortamında daha belirgin yapabilmek adına bölüm 5.1. de görüleceği üzere CStresh\_ ve RXThresh\_ eşik değerleri -66 dBm değerinde daha gerçekçi değerler ile eşitlenmiş ve gizli düğüm problemini daha doğrulukla gösterdiği çalıştırılan üç düğümlü üç çeşit senaryoda doğrulanmıştır.
- Bölüm 5.1. de görüleceği üzere Gizli düğüm simülasyonunda, NS2'nin varsayılan CStresh\_ ve RXThresh\_ değerleri ile RTS-CTS mekanizması kullanılmadan yapılan iletimde 3000 byte'ın altında paket düşmesi gözlenmemiştir. Ancak eşik değerlerin (CStresh\_ = -78 dBm ve RXThresh\_ = -64 dBm) gerçek ortamda böyle olmadığı bilindiğinden sonucun gerçek ortamdakini yansıtması beklenilmemektedir. Bu çeşit hazırlanan senaryo ile gerçek ortamdaki gibi iletişimin benzetilmediği tespit edilmiştir.
- Bölüm 5.1. de görüleceği üzere, CStresh\_ = RXThresh\_ ve RTS-CTS'siz çalıştırılan simülasyonda, gizli düğüm sonucu 7 ve 8 numaralı Veri (cbr) paketinin düştüğü görülmüştür. Ancak RTS-CTS kullanımının gerçek ortamda kullanıldığı bilindiğinden, bu çeşit hazırlanan senaryo ile de gerçek ortamdaki gibi iletişimin benzetilemediği tespit edilmiştir.
- Bölüm 5.1. de görüleceği üzere, CStresh\_, RXThresh\_ eşik değerleri eşit ve RTS-CTS tokalaşma olduğunda çalıştırılan simülasyon, kullanılan değerler gerçek ortamdaki gibi olduğundan geçerli sonuç ürettiği tespit edilmiş, ayrıca tokalaşma kontrolü en ufak bir veri için bile yapıldığından gizli düğümden kaynaklanacak paket düşmesinin şekil 5.3. de görüldüğü üzere olmadığı tespit edilmiştir. Bu çeşit hazırlanan senaryo ile gerçek ortamdaki gibi iletişimin benzetimi yapılmıştır. RTS/CTS mekanizması için hazırlanan bu üç çeşit senaryodan bu çeşidin en güvenilir olduğu sonucuna varılmıştır.

Çalıştırılan bütün simülasyonlarda Mac-802.11.h dosyasındaki, MAC\_RTSThreshold değeri 3000 byte'dan 0 byte'a

```
(#define MAC_RTSThreshold    0    // bytes)
```

gibi ayarlanarak bu deęerin altındaki verilerde bile RTS/CTS mekanizmasının alıřabildięi saęlanmıřtır.

Ayrıca 3000x3000 metrelik sınır ierisinde 5 ve 10 duęümlük hazırlanan kablosuz duęüm senaryosu alıřtırılmıř ve ařaęıdaki gibi sonular elde edilmiřtir:

- Plansız aęlarda gizli duęümlerden kaynaklı arpıřmadan kaınmak iin yaygın olarak RTS-CTS mekanizması kullanılmıřtır. Mevcut uygulamada, RTS ya da CTS alan her bir duęüm daha bařka bir bilgi kullanımı ile iletimini ertelemektedir. Halbuki varsayılan deęerler ile kullanılan NS'de, eřikler öyle tanımlanmıřtır ki, herhangi kablosuz aę, gizli duęümün etkisini görememektedir. Hâlbuki deęiřtirilen eřik deęerleri ile mevcut olanın aksine bu problem farkedilebilir hale getirilmiřtir.
- NS simlasyonu, aık bir řekilde göstermektedir ki RTS-CTS alıř-veriři, örtülü duęümden kaynaklanan arpıřmayı önemli bir ölçüde ortadan kaldırmakta başarısızdır. RTS-CTS mekanizması aynı zamanda, yakınında duęüm iletimi olmasa bile, iletimden bloke edilen bir duęüm olduęunda hatalı bloklama etkisine neden olmaktadır. Bu hatalı bloklamadan ötürü aę bant geniřlięi fark edilebilir aę yüküne neden olmaktadır. NS simlasyonunda, bu etkiyle RTS/CTS mekanizmasının aęı dengeleyebileceęinden daha fazla tıkanıklıęa yol aabileceęi de gösterilmiřtir.

Son olarak bu alıřmada gerek ortamdaki gibi düzeneęi hazırlanan NS simlatörünün TCP ve UDP üstlenicilerde de iki farklı veri tipi kullanılarak (TCP üstlenici iin ftp ve UDP üstlenici iin cbr veri tipi) karřılařtırmalı olarak denenmesi saęlanmış ve hazırlanan simlasyonlar başarılı bir řekilde alıřtırılıp sonlandırılmıřtır. TCP üstleniciyi kullanan simlasyon daha kontrollü bir iletim saęladığından, veri kaybı, gerek ortamdaki gibi daha az olarak gözlenmiřtir.

Bu çalışma sonucunda hazırlanan NS simülatörü ile gerçek ortamdakine daha yakın ve veri kaybının gerçekteki gibi gözlenebildiği bir simülatör düzeneği hazırlanmıştır. Ayrıca kablosuz plansız mobil ağlarda bu gibi bir çalışma hazırlamak isteyenler için de aşağıdaki gibi bir öneri hazırlanmıştır.

## 6.2. Öneriler

- Mobil kablosuz ağlarda NS2 ağ simülatörü tabanlı yapılan bu çalışma, diğer simülatör ortamlarında (OPNET, QualNet v.b.) da denenip, diğer ağ simülatörlerinin de doğruluğu hakkında sonuçlar üretilebilir. Böylece diğer simülatörlerin de gerçek ortamlar için ne kadar uygun olduğu karşılaştırmalı olarak sunulabilir.
- Mobil kablosuz ağlarda gizli düğüm, örtülü düğüm ve bloklama etkisinden kaynaklı paket kayıp oranı, hazırlanacak büyük ve küçük ölçekli ağlarda (ağ bant genişliğinin kullanımını etkileyecek derece, 5 düğümlü, 30 düğümlü v.b.) ve farklı trafik desenleri kullanılarak (CBR, FTP v.b.) yapılacak simülasyonlar ile değişik simülatör ortamlarında elde edilen sonuçlar karşılaştırılabilir.

## KAYNAKLAR

1. Corson, S., Macker, J., "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", RFC 2501, 1-12 (1999).
2. Chlamtac, I., Conti, M., Liu, J. J. N., "Mobile ad hoc networking: imperatives and challenges", *Ad Hoc Networks*, 1(1): 13, 17-19 (2003).
3. The editors of IEEE 802.11, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", ANSI/IEEE Std 802.11, ISBN 0-7381-1658-0, *The Institute of Electrical and Electronics Engineers*, USA, 195-222 (1999).
4. Bharghavan, V., Demers, A., Shenker, S., Zhang, L., "MACAW: A Media Access Protocol for Wireless LAN's", *Proceedings of the conference on Communications architectures, protocols and applications*, United Kingdom, 212-225 (1994).
5. Ju, H., Rubin, I., Kuan, Y., "An Adaptive RTS/CTS Control Mechanism for IEEE 802.11 MAC Protocol", *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*, USA, 1469-1473 (2003).
6. Weinmiller, J., Woesner, H., Ebert, J., Wolisz, A., "Analyzing the RTS/CTS Mechanism in the DFWMAC Media Access Protocol for Wireless LANs", *IFIP TC6 Workshop Personal Wireless Comm.*, Prague, 117-130 (1995).
7. Li, J., Blake, C., Douglas, S. J. De, Hu, C., Lee, I., Morris, R., "Capacity of Ad Hoc Wireless Networks", *Proceedings of the 7th annual international conference on Mobile computing and networking*, Italy, 61-69 (2001).
8. Ray, S., Carruthers, J. B., Starobinski, D., "Evaluation of the Masked Node Problem in Ad Hoc Wireless LANs", *IEEE Transactions on Mobile Computing*, 4(5): 439 (2005).
9. Ivanov, S., Herms, A., Lukas, G., "Experimental Validation of the ns-2 Wireless Model using Simulation, Emulation, and Real Network", *In Proceedings of the 4th Workshop on Mobile Ad-Hoc Networks (WMAN'07) in conjunction with the 15th ITG/GI - Fachtagung Kommunikation in Verteilten Systemen (KiVS'07)*, Switzerland, 433-444 (2007).
10. Cobb, J. A., Agrawal, P., "Congestion or corruption? a strategy for efficient wireless tcp sessions," *Proc. of IEEE Symposium on Computers and Communications*, USA, 262-268 (1995).

11. Li, F. Y., Kristensen, A., Engelstad, P., “Hidden Terminal Detection in 802.11-based Wireless Ad Hoc Networks”, *Proceedings of the 15th IST Mobile & Wireless Communication Summit (ISTsummit'06)*, Greece, 1-5 (2006).
12. Xu, K., Gerla, M., Bae, S., “How Effective is the IEEE 802.11 RTS/CTS Handshake in Ad Hoc Networks?”, *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, USA, 72-76 (2002).
13. Hiertz, G. R., Max, S., Zang, Y., Junge, T., Denteneert, D., “IEEE 802.11s MAC Fundamentals”, *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE International Conference*, Aachen, 1-8 (2007).
14. Çakıroğlu, M., Özcerit, A. T., “Kablosuz Algılayıcı Ağlarda Hizmet Engelleme Saldırılarına Dayanıklı Ortam Erişim Protokolü Tasarımı”, *Gazi Üniv. Müh. Mim. Fak. Der.*, 22(4): 699 (2007).
15. Lyakhov, A., Vishnevsky, V., Yakimov, M., “Multicast QoS Support in IEEE 802.11 WLANs”, *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE International Conference*, Moscow, 1-3 (2007).
16. Ray, S., Starobinski, D., “On false blocking in RTS/CTS-based multihop wireless Networks”, *IEEE transactions on vehicular technology*, 56(2): 850, 856, 857 (2007).
17. Vardakas, J. S., Papapanagiotou, I., Logothetis, M. D., Kotsopoulos, S. A., “On the End-to-End Delay Analysis of the IEEE 802.11 Distributed Coordination Function”, *Second International Conference on Internet Monitoring and Protection, 2007, ICIMP 2007*, Patras, 16-20 (2007).
18. Lindgren, A., “On Three Issues in Wireless Networking”, Doktora Tezi, *Luleå University of Technology*, Sweden, 6, 10 (2003).
19. Cavilla, A.L., Baron, G., Hart, T. E., Litty, L., de Lara, E., “Simplified simulation models for indoor MANET evaluation are not robust”, *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference*, Canada, 610-620 (2004).
20. Zhai, H., Wang, J., Chen, X., Fang, Y., “Medium Access Control Protocols in Mobile Ad Hoc Networks: Challenges and Solutions”, *Wireless Communications and Mobile Computing*, 6(2): 151-170 (2006).
21. İnternet : UC Berkeley, LBL, USC/ISI, and Xerox PARC “The ns Manual” <http://www.isi.edu/nsnam/ns/ns-documentation.html>, (2008).
22. Doğru, İ. A., “Ad-Hoc Ağlarda Hareketlilik Yönetimi Protokolü”, Yüksek Lisans Tezi, *Gazi Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 87-88 (2006).

23. Bay, İ., “Ad-Hoc Ağlarda Enerji Etkin Yönlendirme Protokolü”, Yüksek Lisans Tezi, *Gazi Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 1-2 (2007).
24. Sözer, H. “A Peer-To-Peer File Sharing System For Wireless Ad-Hoc Networks”, Yüksek Lisans Tezi, *Bilkent Üniversitesi Mühendislik ve Fen Bilimleri Enstitüsü*, Ankara, 38-40 (2004).
25. Atsan, E., “A Scalable and Reactive Replication Framework For Mobile Ad-hoc Networks”, Yüksek Lisans Tezi, *Koç Üniversitesi Mühendislik ve Fen Bilimleri Enstitüsü*, İstanbul, 78-79 (2007).
26. Sokullu, R., Sayar, N., “Recommended a routing algorithm for bluetooth based ad-hoc network”, *Signal Processing and Communications Applications Conference, 2005. Proceedings of the IEEE 13th*, Kayseri, 557-560 (2005).
27. Postel, J., Reynolds, J., “File Transfer Protocol”, RFC 0959, 1-2 1985.
28. Lucio, G. F., Farrera, M. P., Jammeh, E., Fleury, M. Reed, M. J., Ghanbari, M., “Packet by Packet Analysis in Contemporary Network Simulators”, *Latin America Transactions, IEEE (Revista IEEE America Latina)*, 4(4): 299-307 (2006).
29. Chane L. Fullmer, J.J. Garcia-Luna-Aceves, “Solutions to hidden terminal problems in wireless Networks”, *Proceedings of ACM SIGCOMM 97*, France, 14-18 (1997).
30. Fouad Tobagi and Leonard Kleinrock, “Packet switching in radio channels: Part 2 - the hidden node problem in carrier sense multiple access modes and the busy tone solution”, *IEEE Transactions on Communications*, 23(12): 1417-1433 (1975).
31. Karn, P., “MACA - a new channel access method for packet radio”, *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, London, 134-140 (1990).
32. Deng, J., Liang, B., Varshney, P. K., “Tuning the Carrier Sensing Range of IEEE 802.11 MAC”, *Global Telecommunications Conference, 2004. GLOBECOM '04*, USA, 2987-2991 (2004).
33. Xue, Y., Li, B., Nahrstedt, K., “Optimal resource allocation in wireless ad hoc networks: A price-based approach,” *IEEE Transactions on Mobile Computing*, 5(4): 347-364 (2006).
34. Tavli, B., Heinzelman, W. B., “Energy and spatial reuse efficient network-wide real-time data broadcasting in mobile ad hoc networks”, *IEEE Transactions on Mobile Computing*, 5(10): 1297-1312 (2006).



35. Zheng, R., Hou, J. C., Sha, L., "Asynchronous wakeup for ad hoc networks", *MobiHoc'03: Proceedings of the 4th ACM international symposium on Mobile adhoc networking & computing*, USA, 35-45 (2003).
36. Akan, Ö. B., Akyildiz, I. F., "Event-to-sink reliable transport in wireless sensor networks", *IEEE/ACM Trans. Netw.*, 13(5): 1003-1016 (2005).
37. Ye, W., Vaughyan, R. T., Sukhatme, G. S., Heidemann, J., Estrin, D., Mataric, M. J., "Evaluating control strategies for wireless-networked robots using an integrated robot and network simulation", *Proceedings of the IEEE International Conference on Robotics and Automation. IEEE*, Korea, 2941-2947 (2001).
38. Kotz, D., Newport, C., Gray, R. S., Liu, J., Yuan, Y., Elliott, C., "Experimental evaluation of wireless simulation assumptions", *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, USA, 78-82 (2004).
39. Bruno, R., Conti, M., Gregori, E., "IEEE 802.11 optimal performances: RTS/CTS mechanism vs. basic access", *Proc. The 13th IEEE International Symposium. Personal, Indoor and Mobile Radio Communications*, Italy, 1747-1751 (2002)
40. Shih, K. P., Chang, C. Y., Chen, H. C., Chang, C. W., "On Avoiding RTS Collisions for IEEE 802.11-Based Wireless Ad Hoc Networks", *Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference*, Taiwan, 747-752 (2006).
41. Breaslu, L., Estrin, K., Fall, D., Floyd, S., Heidemann, J., Helmy, A., Huang, P., McCanne, S., Varadhan, K., Ya, X., Yu, H., "Advances in network simulation", *IEEE Computer*, 33(5): 59-67 (2000).
42. Lucio, G. F., Paredes-Farrera M., Jammeh F., Reed M. J., "OPNET Modeler and Ns-2: Comparing the Accuracy Of Network Simulators for Packet-Level Analysis using a Network Testbed", *WSEAS Transactions on Computers*, 2(3): 700-707 (2003).
43. Khalaf, R., El-Haj-Mahmoud, A., Kayssi, A., "Performance Comparison of the AODV and DSDV Routing Protocols in Mobile Ad Hoc Networks", *Proceedings of the 1st IASTED International Conference on Communication Systems and Networks (CSN'02)*, Spain, 41-46 (2002).
44. Fall, K., "Network emulation in the VINT/ns Simulator", *Proc. Of the fourth IEEE Symposium on Computers and Communications (ISCC'99)*, Egypt, 59-67 (1999).

45. Estrin, D., Handley, M., Heidemann, J., McCanne, S., Xu, Y., Yu, H., “Network visualization with the VINT Network Animator Nam”, *IEEE Computer*, 33(11): 63-68 (2000).

## ÖZGEÇMİŞ

### Kişisel Bilgiler

Soyadı, adı : TANYERİ, Ufuk  
 Uyuğu : T.C.  
 Doğum tarihi ve yeri : 09.08.1978 / Ankara  
 Medeni hali : Bekâr  
 Telefon : 0 (312) 485 11 24  
 Faks : 0 (312) 485 31 23  
 e-mail : utanyeri@gazi.edu.tr

### Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	Gazi Üniversitesi/Bilgisayar Eğitimi	2001
Orta Öğrenim	Başkent Lisesi Fen Bilimleri	1995

### İş Deneyimi

Yıl	Yer	Görev
2005-	Gazi Üniversitesi	Araştırma Görevlisi
2001-2005	M.E.B.	Bilgisayar Öğretmeni
2000-2001	Yargıtay Ek Bina	Bilgi İşlem Personeli

### Yabancı Dil

İngilizce, Almanca

### Proje

TUFEKÇİ, A., ÇAKIR, H., İNCETAS, M.O., PEHLIVAN, S., TANYERİ, U., "Intranet Üzerinden Yarı On-Line Bir Sınav Sisteminin Hazırlanması ve Kullanılabilirliğinin Ölçülmesi (Preparing online examination over intranet and evaluating its usability)", code number: 25/2006-03, Gazi BAP 2006.

**Sunulu Konferans**

-Yasemin GENC, Selcen YUKSEL, Salih DEMIR, Ufuk TANYERI, Can ATES, Derya OZTUNA, "Kumelenmis Verilerde Mann-Whitney U Testinin Uygulanmasi", XI. Ulusal Biyoistatistik Kongresi, 27-30 Mayıs 2008, İnönü Üniversitesi Tıp Fakültesi - Malatya 2008.

-BULBUL, H.I., TANYERI, U., SAHIN, Y.G., "Internet Protocol v6 (IPv6) and the Transition Process", The Proceedings of 8th International Educational Technology Conference, May 6-9, 2008, Anadolu University – Eskisehir 2008.

-TANYERI, U., TUFEKCI, A., "How Accessible is The Internet Based Higher Education Programs For Visually Impaired People in Turkey?", The Proceedings of 2nd International Computer and Instructional Technologies Symposium, April 16-18, 2008, Ege University - Kusadasi 2008.

-TUFEKÇİ, A., ÇAKIR, H., İNCETAS, M.O., PEHLIVAN, S., TANYERI, U., "İntranet Tabanlı Bir Sınav Sistemi: Gereke ve Tasarım", The Proceedings of 7th International Educational Technology Conference, IETC 2007, May 3-5, 2007, Near East University - Cyprus 2007.

-TUFEKÇİ, A., ÇAKIR, H., İNCETAS, M.O., PEHLIVAN, S., TANYERI, U., "İnternet Tabanlı Yarı Çevrimiçi Bir Sınav Sistemi Örneği", 1. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu, May 16-18, 2007, S. 867- 875, Çanakkale Üniversitesi - Türkiye.

**Hobiler**

Yüzme, Bilgisayar teknolojileri, Basketbol