Düşük Maliyetli 802.11 Kablosuz Ağ Dinleyici
Low Cost 802.11 Wireless Network Sniffer

Yusuf Türk[1], Sezer Gören[1], Onur Demir[1]

[1]Bilgisayar Mühendisliği Bölümü
Yeditepe Üniversitesi
yturk@cse.yeditepe.edu.tr, sgoren@cse.yeditepe.edu.tr, odemir@cse.yeditepe.edu.tr

Özet

Paket dinleyici (sniffer) erişim alanındaki tüm ağ paketlerini varış adresi kendisi olmamasına rağmen yakalayan yazılım ya da cihazdır. Paket dinleyiciler kablosuz ağların test ortamlarında yoğunlukla kullanılmaktadır. Bu çalışmada Raspberry Pi üzerinden paket yakalama sistemi ucuz, hızlı ve genişletilebilir bir çözüm olarak sunulmaktadır. Dinleme (Monitor) modunda çalışan donanım istatiksel analiz yazılımı ile desteklenmiştir. Geliştirilen yazılım paketlerin tipleri ve alt-tipleri hakkında bilgi, en çok yayın yapan erişim noktalarının isimleri (SSID), ve yönetim paketlerini SSID ve MAC adresine göre filtrelemeyi sağlamaktadır. Sonuçlar aynı anda sistemden bağımsız olarak paket dinleyici görevi gören bir bilgisayarın verileri ile karşılaştırılarak doğrulanmıştır.

Abstract

Packet capture software or hardware which captures all the packets over the air even though the destination address is not itself is called a sniffer. Packet sniffers are heavily used in test environments of wireless networks. In this work, Raspberry Pi packet capture system is presented as a low cost, fast, and expandable solution. The hardware working on Monitor mode is supported with a statistical analysis software. Developed software gives information about packet types and subtypes, names (SSIDs) of the most broadcasting access points, and allows filtering of management packets by SSID and MAC address. Results are compared with the data gathered simultaneously by a computer acting as a sniffer outside of the test system.

1. Giriş

Kablosuz ağ kullanımı son yıllarda geleneksel yerel alan ağlara göre hızla artmış ve hız açısından Gigabit Ethernet hızlarıyla yarışır hale gelmiştir. Bu gelişmeler mobil cihazların kullanımın artması ve akıllı ev cihazları gibi daha önceki zamanlarda bir ağa bağlı olma gereksiniminin olmadığı cihazların da artmasıyla kablosuz ağlar varsayılan ağ türü olarak görülmektedir.

Şirketler, araştırma grupları ve profesyoneller bir ağa bağlı olması gereken ve fonksiyonlarının buna bağlı olduğu cihazlarını geliştirirken bir test ortamına ihtiyaç duymaktadırlar. Test ortamında cihazın ağa bağlanabilmesi, ağ üzerinden veri alışverişinin sorunsuz sağlanması, bağlantı kopması durumunda yapılacakların benzetiminin yapılması ve diğer durumlar canlandırılır. Testlerin başarılı bir şekilde yapılabilmesi ve ağ trafiğinin incelenmesi için ağ içerisinde ağa bağlı ya da bağlı olmayan dinleyici (sniffer) kullanılır. Kablosuz ağlarda paket yakalama ağa bağlı olarak rastgele (promiscuous) ya da izleyici (monitor) modu kullanılarak yapılır. Bu çalışmada izleyici modu kullanılacaktır.

Sektörde paket yakalama için açık kaynak kodlu ya da ticari yazılımlar kullanılmaktadır. Açık yazılımlardan en çok kullanılanlar arasında tcpdump [1] ve Wireshark [2] yer alır. İki yazılım da detaylı filtreleme özellikleri sunmasına rağmen görsel ve istatistiksel analiz bakımından eksiklikler içermektedir. Pahalı, ticari bir yazılım olan OmniPeek [3] ise daha detaylı analiz özellikleri sunmaktadır. Donanım açısından paket yakalama için desteklenen erişim noktaları üzerinde Linux tabanlı OpenWRT [4] yazılımı kullanan alternatifler de mevcuttur. Ancak düşük maliyetli erişim noktaları bu çalışmada kullanılan Raspberry Pi'a [5] kıyasla daha düşük işlemci hızına ve belleğe sahiptir. Bunlara ek olarak AirPcap [6] gibi USB adaptörler ve Eye P.A. [7] gibi yazılım ve donanım setleri de mevcuttur. Çizelge 1 2014 yılında sektörde kullanılan paket yakalama için kullanılan yazılım ve cihazlarının fiyat bakımından karşılaştırılmasını göstermektedir.

Çizelge 1: Paket yakalama araçlarının karşılaştırılması

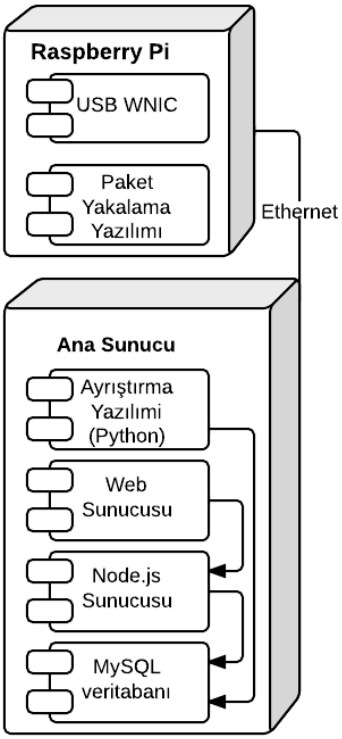
Ürün Adı	Fiyat (USD)
OmniPeek Yazılımı	1000+
AirPcapNX3 + Yazılım	700
Eye P.A. Yazılım	500
Eye P.A. + Donanım	1150
Erişim Noktası	60-100
Raspberry Pi	35-50

Bu çalışmada önerilen sistem güncel kablosuz test ortamlarında kullanılan paket yakalama cihazlarına ucuz, kolay taşınabilir ve aynı anda daha fazla işlem yapabilecek bir alternatif olarak tasarlanmıştır. Sistem kolay kurulumuyla her tür kablosuz ağ test ortamında kullanılabilecek ve yakaladığı paketler ile beklenen sonuçlar karşılaştırılarak kablosuz cihazların paketleri doğru gönderip göndermediği veya çıkan iş oranının yeterliliği hakkında bilgi sahibi olunabilecektir. Sistemin geliştirilmesinde Raspberry Pi (RPi) ve USB kablosuz ağ adaptörü kullanılmıştır. Buna ek olarak, izleyiciden alınan verileri analiz etmek için geliştirilen yazılım ile gerçek zamanlı istatistikler sunulmaktadır. Ayrıca çevredeki erişim noktaları hakkında detaylı veriler de

sunulmaktadır. Bu veriler incelenerek erişim noktalarının izleyici sisteme olan uzaklıkları da tahmin edilmektedir.

2. Raspberry Pi ile Paket Yakalayıcı Sistem

Önerilen sistem paket yakalama, ayrıştırma ve gözlemleme evrelerinden oluşmaktadır. Şekil 1'de sistem bileşenleri ve aralarındaki bağlantı gösterilmiştir. Paketler RPi'a USB veri yolu ile bağlı kablosuz ağ kartı ile yakalanmaktadır. RPi içerisinde paket yakalama kütüphanesi libpcap [8] kullanılarak hazırlanmış C programı çalışmaktadır. Bu programın çıktısı ana sunucuya yollanmaktadır. Burada ayrıştırma programı paketlerin tipleri, alt-tipleri, gönderildikleri ve gidecekleri MAC adresler gibi bilgileri işlemektedir. Elde edilen veriler paket veri tabanına yazılmaktadır. Bunlara ek olarak, mobil cihazlardaki bildirim mesajlarına benzer bir bildirim sunucusu da kullanıcı arayüzü ile iletişimi sağlamaktadır. Sistemin kullanıcıları arayüz üzerinden gerçek zamanlı bilgilere ulaşabilecektir.



Şekil 1: Sistem Bileşenleri

Sistem kablosuz ağ kullanan test ortamlarında kullanılacaktır. Test sırasında internet sunucusu üzerinden sunulan web sayfası ile her çalışan test ortamını gözlemleyip elde edilen istatiksel sonuçlar ve paket verilerine ulaşabilecektir. Bu şekilde test mühendisi cihazlarla dinlediği paket verisini zaman kaybetmeden ilgili yazılım geliştiricisine ulaştırıp cihazdaki sorunun çözüm sürecini hızlandıracaktır. Aynı zamanda küçük ölçekli ve bütçe konusunda sıkıntı yaşayan kuruluşlara da örnek teşkil etmektedir. Ağ dinleyici sistemden elde edilen sonuçlar kablosuz ağ güvenliği konusunda yapılabilecek akademik ve endüstriyel çalışmalar için bir başlangıç noktası olarak kullanılabilir. Örnek olarak kablosuz ağ içerisinde yapılan ağ güvenliği için tehlike unsuru oluşturabilecek saldırı tipleri de bu sistem kullanılarak tespit edilebilir.

2.1. İzleyici Mod ile Paket Yakalama

Paket yakalama ya da dinleme, kablolu ya da kablosuz ağlarda ağ paketlerinin işletim sistemi tarafından işlenmeden önce bir kopyasının yakalanmasıdır. Kullanılan ağ arayüzü bir ağa bağlı ise, tüm çerçeveler (frame) şifrelenmemiş durumda ve ağdaki tüm cihazlara görünür durumdadır. Eğer izleyici mod kullanılıyorsa çerçeveler yakalanır ancak şifrelenmiştir.

Kablosuz 802.11 ağlarında kullanılan altı işletim modu vardır. Bunlar sahip (master), yönetimli (managed), ad-hoc, örgü (mesh), tekrarlayıcı (repeater) ve izleyici (monitor) modlarıdır. İzleyici modu rastgele dinleme moduna benzerdir, ancak sadece kablosuz ağlar için geçerlidir. İzleyici modunda ağı dinleyen cihazlar ağa bağlı olmak zorunda olmadan kablosuz ağ kartının görebildiği tüm paketleri yakayabilir. İzleyici modu kablosuz ağ kartının sürücülerine, cihazın yerleşik yazılımına ve yonga kümesine (chipset) bağlıdır. Limitler de göz önünde bulundurulduğunda tüm ağ kartları bu modu desteklememektedir. 802.11 ağlarda ağ dinleme ve paket yakalama bir bilgisayara USB veri yolu ile bağlı kablosuz ağ kartı ile ya da özel yazılım kullanan erişim noktaları ile yapılmaktadır. İki yöntem de önceki çalışmamızda geliştirilmiş ve denenmiştir [9].

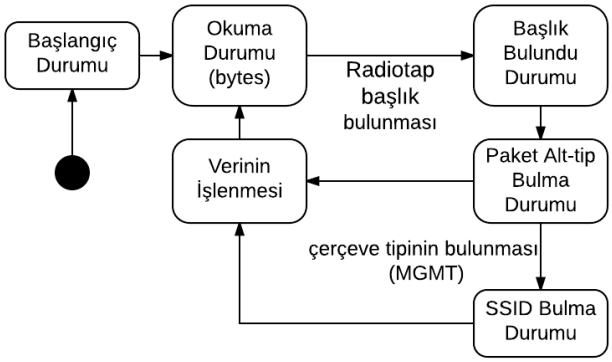
Raspberry Pi (RPi) düşük maliyetli, az yer kaplayan ve Fedora, Debian, Arch Linux gibi Linux türevleri ile çalışabilen bir bilgisayardır. İşlemci ve bellek özelliklerine göre birçok düşük maliyetli erişim noktalarından güçlü olmakla birlikte, içerisinde bir kablosuz ağ kartı bulunmamaktadır. Ancak USB veri yolu ile kolayca bir kablosuz ağ kartı bağlanabilir ve 802.11 paketleri yakalanabilir. Bu çalışma için ağ kartının modu izleyici olarak seçilmiştir. Bu sayede çevredeki ağlara bağlanmadan paketler yakalanabilmiştir. RPi üzerinde paket yakalamak için libpcap kütüphanesi kullanılarak geliştirilmiş bir C programı çalışmaktadır. Geliştirilen başlangıç komut dosyaları ile RPi bağlangıçta kendisini izleyici moda alır ve çevredeki paketleri yakalamaya başlamaktadır. Paket yakalanan kablosuz ağ kanalı otomatik olarak seçilmektedir. Programın çıktıları cihazın Ethernet bağlantı noktasına yönlendirilir. RPi üzerindeki yazılımların çalışma yapısı aşağıda verilmiştir.

1. RPi sistem başlangıcı
2. Başlangıç komut dosyalarının çalışması (rc.S)
3. Özelleştirilmiş komut dosyalarının çağırılması
4. Izleyici mod arayüzünün oluşturulması
5. Paketin yakalanması
6. Radiotap başlık [10] eklenmesi
7. Paketin Etnernet bağlantı noktasına yönlendirilmesi
8. Beşinci adıma geri dön

2.2. Paketlerin Ayrıştırılması

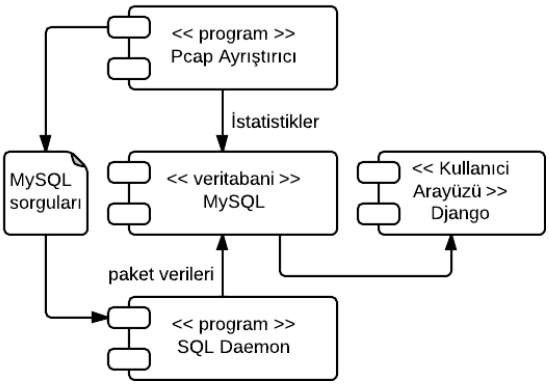
RPi tarafından yollanan veriler ana sunucuda .pcap uzantılı bir dosyanın sonuna eklenmektedir. Pcap ayrıştırıcı program çıktıların gerçek zamanlı olarak analiz edilmesi ve istatiksel verilerin elde edilmesini sağlamaktadır. Bu program Python programlama dilinde geliştirilmiştir ve bir durum makinası gibi çalışır. Şekil 2'de gösterildiği gibi programın başlangıç,

okuma, başlık bulma, alt-tip bulma, SSID bulma ve verinin işlendiği durum olarak altı adet durumu vardır.



Şekil 2: Ayrıştıcı Programın Durum Çizeneği

Başlangıç durumunda günlük dosyaları, veritabanı tabloları ve değişkenler ilk kullanıma hazırlanır. Amaç gerçek zamanlı takip sistemi oluşturmak olduğu için önceki günlük dosyaları boşaltılır. MySQL veritabanı bağlantısı ve imleci bu aşamada oluşturulur. Okuma durumunda paket dosyası işlenmeye başlar ve bir Radiotap başlık bulunana kadar bu durumda kalınır. Eğer başlık bulunduysa program devam eder ve paket alt tipi bulunur. Eğer paket yönetim paketi ise paketin gönderildiği erişim noktasının isim (SSID) ve adresinin (MAC) bulunması için SSID bulma durumuna geçilir.

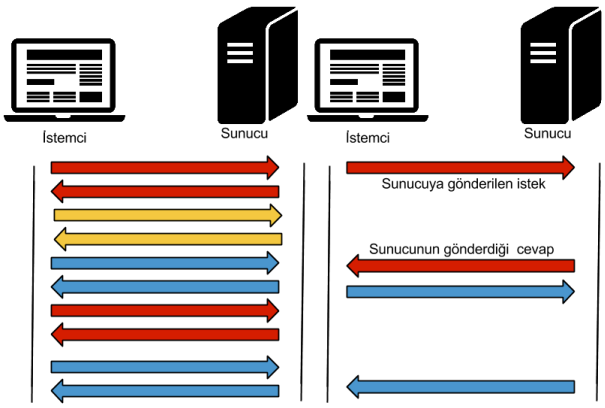


Şekil 3: Veritabanı kullanımı

Pcap ayrıştırıcı yazılım MySQL sorgularını hazırlar ancak tamamını yürütmez. Yazılım ve veritabanı ilişkisi Şekil 3'de görülebilir. Performansı düşürebilecek büyüklükteki sorgular aynı anda çalışan bir bekletici program (daemon) tarafından yürütülür. Bu program veritabanına yazılacak veri olmadığında performans açısından olumsuz sonuçlar olmaması için uyku moduna gireceği zamanın duruma göre optimize edildiği bir algoritma kullanır.

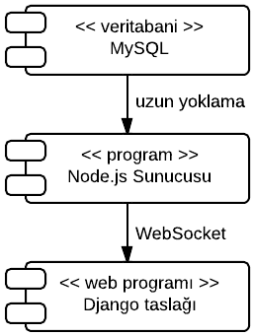
2.3. Kullanıcı Arayüzü

Bu çalışmada paket yakalama sistemine ek olarak kullanıcıların gerçek zamanlı olarak gelişmeleri takip edebilecekleri bir arayüz de geliştirilmiştir. Bu arayüzde sunulan veriler paket istatistikleri, paket özet verileri ve erişim noktası istatistikleridir. Arayüz web sunucusu üzerinden servis edilmektedir. Bu sayede ana sunucuya bağlı olan her kullanıcı arayüze ulaşabilecektir. Bunun yanında, arayüzün web üzerinden sunulması ağ sorunlarının çözülmesi aşamasında uzaktan bağlantı gibi çözümler de sağlamaktadır.



Şekil 4: Sunucudan veri yoklama işlemi

Toplanan verilerin gerçek zamanlı sunulabilmesi için kullanılan veritabanındaki verilerin yenilenip yenilenmediği kısa aralıklarla kontrol edilmelidir. Eğer yeni veri yok ise arayüz yenilenmemeli eski değerlerini korumalıdır. Arayüzü sadece yeni veri olması halinde yenilemek performansın düşmemesi açısından önemli bir etkendir. Bu işlem Node.js [11] ve Websocket [12] kullanılarak yapılmıştır. Uzun süreli yoklama (long polling) ana sunucuya yapılan istemlere sunucunun belirli bir süre sonunda ya da yeni veri olduğunda cevap verdiği yöntemdir. Bu yöntem bildiri mesaj teknolojisinin emulasyon edilmiş versiyonudur [13]. Şekil 4 yoklama ve uzun yoklama yöntemlerinin karşılaştırılmasını görselleştirmiştir. Her bir renk sunucuya yapılan yeni bir isteği ve sunucudan dönen cevabı göstermektedir. Web arayüzü olarak kullanılan Python tabanlı Django arayüzü [14] ve MySQL veritabanı ile iletişimi de Node.js sunucusu sağlamaktadır. Aralarındaki bağlantı Şekil 5'de görülebilir.

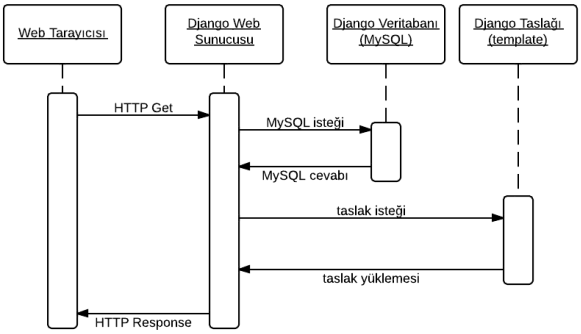


Şekil 5: Veritabanı ve web arayüzü bağlantısı

Django Python tabanlı bir web arayüzüdür. Sayfa içerisindeki öğeler veritabanında saklanır ve sayfalar istemci tarafından çağırılana kadar taslak olarak görülür. Sayfa yüklenirken veritabanındaki verilere göre dinamik olarak yaratılır. URL yapıları da düzenli ifadeler (regex) kullanılarak ayarlanmıştır. Şekil 6 Django sayfa yapısını göstermektedir.

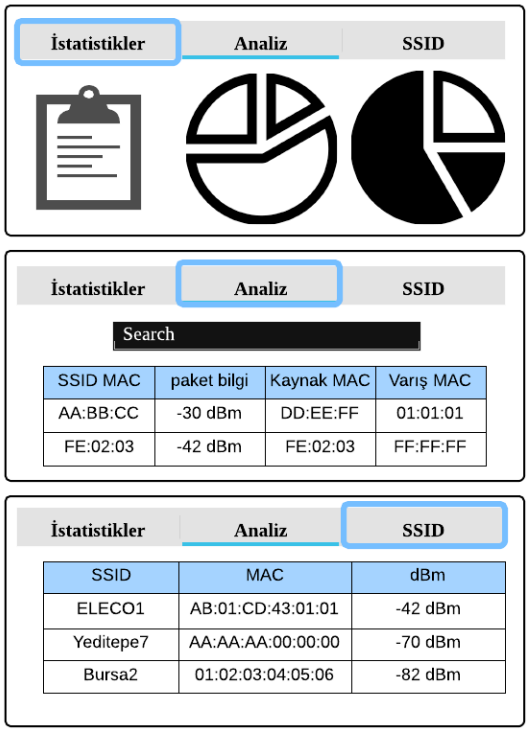
Arayüz önceki bölümlerde tanımlanan istatistikleri kullanıcıya sunmaktadır. Arayüzün modeli Şekil 7'de verilmiştir. Bu modelde ilk ekran istatistik sayfasını, ikinci ekran analiz ve arama sayfasını ve son ekran ise erişim noktası bilgilerini

içermektedir. Arayüzün ilk kısmı istatistik sayfasında yakalanan paket sayısı, paket türlerine göre dağılımı ve ayrıştırıcının işlediği paket sayısı gerçek zamanlı olarak sunulmaktadır.



Şekil 6: Django sayfa isteği sırasındaki işlemler

Analiz tarafında sunulan arama özelliği ile istenen paket detayları MAC adresine göre veritabanında aranıp kullanıcının kolay bir şekilde bulması sağlanmıştır. Burada paket bilgileri özet olarak sunulmuş olup arama sonucunda verilen paket numarası referans alınıp detaylı inceleme imkanı sunulmaktadır. Son arayüz ekranı SSID'de ise çevredeki erişim noktaları ve yolladıkları tüm paketler baz alınarak ortalama sinyal güçleri verilmektedir. Bu sayfa da gerçek zamanlı olarak kendisini yenilemektedir ve erişim noktası uzaklaştıkça sinyal gücü de azalmaktadır.



Şekil 7: Arayüz modeli.

3. Testler ve Performans

Sistemi test etmek için birkaç test ortamı hazırlanmıştır. Sistem ayrıca birçok erişim noktasının bulunduğu ortamlarda da test edilmiştir. İlk test ortamında bir erişim noktası ve iki Ubuntu Linux işletim sistemi kullanan bilgisayar bulunmaktadır. Bilgisayarlar arasında iperf [15] programı ile periyodik olarak UDP paket transferi yapılmıştır. Buna ek olarak ilk ağdan bağımsız bir ağda bir erişim noktası, bilgisayar ve iki mobil cihaz ile kesintisiz video yayın ortamı kurulmuştur. Sistem yayınlanan paketleri herhangi bir ağa bağlanmadan başarı ile yakalamıştır. Test ortamındaki cihazlar erişim noktasına önceden bağlı oldukları için çoğunlukla kontrol ve veri paketleri yakalanmıştır.

Başka bir testte ayrıştırıcı yazılım önceden kaydedilmiş büyük boyutlu pcap uzantılı paket yakalama dosyasını ayrıştırmış ve hızı incelenmiştir. Bu test Ubuntu 12.04 işletim sistemi kullanan, 1.7 GHz hızda iki çekirdeğe sahip ve 1GB belleğe sahip bir bilgisayarda yapılmıştır. Örnek paket dosyası birkaç saatte yakalanmış iki milyondan fazla paket bulundurmaktadır. Bu testin amacı yazılımın yoğun trafiği gerçek zamanlı olarak analiz edip edemeyeceğini görmektir. Yazılım bellekte paket verilerini saklamamaktadır. Yazılım çalışırken bellek kullanımı işletim sistemi görevleri dahil yüzde 30'u geçmemiştir ve 500.000 paket 24.7 dakikada ayrıştırılmıştır. Buna göre yazılım test sunucusunda saniyede 20 bin pakete kadar gerçek zamanlı çıktı sağlayabilmektedir. Ayrıştırma hızının orta seviyede bir ağ için makul bir hız olduğu gözlemlenmiştir. Bunlara ek olarak yapılan testlerde yakalanan 22 milyondan fazla paketin boyutları 20 ve 320 bayt arasında değişmiştir.

Bu çalışmada RPi geleneksel düşük maliyetli erişim noktalarına alternatif olarak kullanılmıştır. Geçmiş çalışmamızda [9] paket yakalamak için kullanılan erişim noktası ile RPi karşılaştırılması yapılmıştır (Çizelge 2). Buna göre ayrıştırma işlemi ve sonrası ana sunucuda yapıldığından dolayı paket yakalama amacıyla RPi kullanmak daha iyi bir çözümdür.

Çizelge 2: RPi ve erişim noktası karşılaştırılması

Özellik	RPi	Erişim noktası
1.CPU	700MHz ARM	300 MHz MIPS
2.Bellek (RAM)	512 MB	32 MB
3.Fiyat	35-50 USD	70-100 USD
4.Arttırılabilir Disk Alanı	USB ile sınırsız	Yok
5.Yazılım ekleme / değiştirme	gcc, python derleyicisi	Özel yerleşik yazılım / OpenWRT
6.Grafik kullanıcı arayüzü (GUI)	Var	Yok
7.SSH bağlantısı	Var	Var
8.Güç	3.5 W	10+ W
9.İşletim sistemi	Raspbian (Debian tabanlı)	Linux (Kernel 2.6)

4. Sonuçlar

Bu çalışmada düşük maliyetli, raftan alınabilir ürünlerle ve isteğe göre uyarlanabilir bir çözüm sunmaktadır. Kullanıcı arayüzünde paket istatistikleri, paket özet verileri ve erişim noktası istatistikleri verilmiştir. Paket içeriğinin tamamı kullanılmadığı için veritabanında sadece özet veriler depolanmaktadır. Bu sayede istenen paket verilerine daha pratik bir şekilde ulaşılmakta ve yoğun ağ trafiği olmasına rağmen istenen bilgiler ulaşılabilir durumdadır. Çalışmanın akademik projeler ve küçük çaplı şirketler için faydalı olması beklenmektedir.

Çizelge 3: Paket boyutları istatistikleri

Paket Boyutu (byte)	Toplam Paket Sayısı	Yüzde (%) Oranı
0 – 19	0	0
20 – 39	1010483	15.10
40 – 79	1391662	20.79
80 – 159	1504351	22.47
160 – 319	1288642	19.25
320 – 639	993486	14.84
640 – 1279	46810	0.70
1280 – 2559	458124	6.84
Toplam	6693558	100 %

Yapılan testlerde RPi kullanmanın yoğun kablosuz ağ trafiğine uygun olduğu görülmüştür. Önceki çalışmamızdaki [9] erişim noktası ile karşılaştırıldığında da avantajlar görülmüştür. *Çizelge* 3 yakalanan paketlerin boyut açısından istatistiklerini içermektedir.

Gelecek çalışmalarda paket yakalayıcının sunucuya veri gönderme performansı geliştirilebilir. Bunun yanında paket ayrıştırma işlemlerini RPi'a taşımak da hızı arttıracaktır.

5. Kaynaklar

[1] TCPDUMP/LIBPCAP public repository. Erişim tarihi: Haziran, 2014, www.tcpdump.org

[2] Wireshark Network Analyzer. Erişim tarihi: Haziran, 2014, www.wireshark.org

[3] Wildpackets Products. Erişim tarihi: Haziran, 2014, www.wildpackets.com/products/

[4] OpenWRT Table of Hardware. Erişim tarihi: Haziran, 2014, wiki.openwrt.org/toh/start

[5] Raspberry Pi, Erişim tarihi: Haziran, 2014, www.raspberrypi.org

[6] AirPcap Nx, Erişim tarihi: Haziran 2014, www.airpcap.nl/airpcap-nx.htm

[7] Eye P.A. Visual Packet Analysis for WLANs. Erişim tarihi: Haziran, 2014, www.metageek.net/products/eye-pa

[8] Carstens, T., Harris, G. , Programming with pcap. Erişim tarihi: Haziran, 2014, www.tcpdump.org/pcap.html

[9] Turk,Y. (2013). Wireless Sniffer System. Mezuniyet Projesi Raporu.Yeditepe Üniversitesi, Istanbul

[10] Radiotap Header, Erişim tarihi: Haziran, 2014 www.radiotap.org

[11] Node.js, Erişim tarihi: Haziran, 2014, www.nodejs.org

[12] WebSocket, Erişim tarihi: Haziran, 2014 en.wikipedia.org/wiki/WebSocket

[13] Push Technology, Erişim tarihi: Haziran, 2014 en.wikipedia.org/wiki/Push technology

[14] Django Project, Erişim tarihi: Haziran, 2014 www.djangoproject.com

[15] Iperf, Erişim tarihi: Haziran, 2014 en.wikipedia.org/wiki/Iperf