Gazi Üniv. Müh. Mim. Fak. Der.
Cilt 27, No 1, 59-69, 2012

Journal of the Faculty of Engineering and Architecture of Gazi University
Vol 27, No 1, 59-69, 2012

FPGA TABANLI PROGRAMLANABİLİR GÖMÜLÜ SALDIRI TESPİT SİSTEMİNİN GERÇEKLEŞTİRİLMESİ

Taner TUNCER, Yetkin TATAR
Fırat Üniversitesi, Bilgisayar Mühendisliği, 23119, ELAZIĞ
ttuncer@firat.edu.tr, ytatar@firat.edu.tr

(Geliş/Received: 27.12.2010; Kabul/Accepted: 30.11.2011)

ÖZET

Bu makalede gerçek zamanda çalışan gömülü bir Saldırı Tespit Sistemi'nin (STS) tasarımı gerçekleştirilmiştir. Ağdan yakalanan paketlerin sınıflandırılabilmesi için kullanılan STS, programlanabilir (SOPC - System on Programmable Chip) bir yapıda oluşturulmuştur. STS'de kullanılan işlemcinin konfigüre ve kontrol ettiği 10/100 Ethernet IP MAC Core donanımsal modülü, IEEE 802.3 standardındaki çerçevelerin ağdan yakalanması için kullanılmıştır. Yakalanan her bir çerçeve gömülü STS'nin hafıza elemanlarına, çerçevelerden elde edilecek paketlerin sınıflarının belirlenmesi için depolanmıştır. İşlemci çerçevelerden paketlerin, paket özetlerinin elde edilmesi ve paket sınıfının belirlenmesi için programlanmıştır. Sistemin sınıflandırma sürecinde Yapay Sinir Ağları (YSA) kullanılmıştır. YSA'nın girişleri paketlerin başlık yapısından elde edilen IP numaraları, port numaraları gibi özelliklerdir. Gerçekleştirilen gömülü STS, eğitim verileri ile eğitilmiş daha sonra gerçek zamanda paketlerin sınıflandırılması yapılarak paket sınıflarının tespit süreleri elde edilmiştir. STS'nin gerek donanımsal gerekse yazılımsal olarak kolay ve esnek bir şekilde tasarlanıp modifiye edilebilme ve programlanabilme özelliğini taşıması için FPGA ortamı tercih edilmiştir. Bunun için üretici Altera firmasının Cyclone III EP3C40F484C7N FPGA (Field Programmable Gate Array-Alan Programlanabilir Kapı Dizileri) platformu kullanılmıştır.

Anahtar Kelimeler: Saldırı Tespit Sistemi, Chip üzerinde programlanabilir sistem, 10/100Ethernet MAC Modül

IMPLEMENTATION OF THE FPGA BASED PROGRAMMABLE EMBEDDED INTRUSION DETECTION SYSTEM

ABSTRACT

In this paper, an embedded Intrusion Detection System (IDS) running in the real time has been implemented. The implemented system is based on System On Programmable Chip (SOPC) to classify the captured packet from the network. IEEE 802.3 standard frames have been captured with 10/100 Ethernet MAC Core programming of the processor used in the system. Each captured frame has stored to the memory of the embedded IDS for the determination of packet classes from obtained frames. The processor has been programmed to determine the packet classification and to obtain the packet summaries and the packets from frames. Artifical Neural Network (ANN) has been used in the classification process of the system. The inputs of ANN are obtained from the features of packet headers, such as port number and IP number. The implemented embedded IDS has been first trained with training data. Then, packet classification has been performed in the real time and finally time of determining packet classes have been obtained. The system has been implemented on Altera's Cyclone III EPC3C40F484C7N platform and software environments.

Keywords: Intrusion Detection System, System On Programmable Chip, 10/100 Ethernet MAC Core

1.GİRİŞ (INTRODUCTION)

Son yıllarda bilgisayar sistemlerinin ve ağlarının hızlı gelişmesine paralel olarak, bunların saldırılara karşı güvenliğinin sağlanması da önem kazanmış olup bu konudaki araştırmalar ve geliştirmeler artarak devam etmektedir. Bilgisayar sistemlerine ve ağlarına yapılan saldırılar genel olarak 4 grupta incelenir.

Bunlar DoS (Denial of Service), Probe, R2L (Remote to Local) ve U2R (User to Root) saldırılarıdır. DoS saldırıları genelde TCP/IP protokol yapısındaki açıklardan faydalanarak veya bir sunucuya çok sayıda istek göndererek onu tıkamaya sebep olan saldırılardır. Probe saldırıları bir sunucunun ya da herhangi bir makinanın, geçerli IP adreslerini, aktif portlarını veya işletim sistemini öğrenmek için yapılan saldırılardır. R2L, kullanıcı haklarına sahip olunmadığı durumda misafir ya da başka bir kullanıcı olarak bilgisayar sistemine izinsiz erişim yapılmasıdır. U2R, bilgisayar sistemine girme izni olan fakat yönetici olmayan bir kullanıcının yönetici izni gerektirecek işler yapmaya çalışmasıdır. Bu tür saldırıları tespit etmek için kullanılan iki temel yöntem vardır. Bu yöntemler Kötüye kullanım ve Anormallik tespiti tabanlıdırlar. Kötüye kullanım tabanlı yöntemde, sisteme yapılmış olan saldırı örüntüleri kullanılarak bir kurallar kümesi oluşturulur. Bu kurallar kullanılarak saldırı tespiti yapılır. Bu yöntemde kurala uymayan saldırı tipleri tespit edilememesine rağmen yanlış alarm oranı çok düşüktür. Anormallik tabanlı yaklaşımda, korunacak sistemin normal davranışı modellenir. Bu davranıştan sapmalar saldırı olarak değerlendirilir. Bu yöntemde önceden bilinmeyen saldırılar tespit edilebilmesine rağmen, yanlış alarm sayısı yüksektir[1,2].

Saldırı Tespit Sistemi, İnternet veya yerel ağdan gelebilecek, sisteme zarar verebilecek çeşitli saldırı paketlerini belirlemek için, değişik kriterleri baz alarak değerlendirip yorumlayan yazılım veya yazılım-donanımdan oluşmuş bir yapıdır. Bir STS; paket yakalama, paket ön işleme, sınıflandırma ve karar mekanizmaları birimlerinden oluşur [3,4].

Literatürde saldırıların tespiti için makine öğrenmesi algoritması kullanan birçok STS geliştirilmiştir. Geliştirilen bu STS'ler Yapay Sinir Ağları, Genetik Algoritmalar, Bayes Ağları, Karar Ağaçları gibi makine öğrenmesi teknikleri yaklaşımlıdırlar [5-10]. Birden fazla makine öğrenmesi algoritmaları kullanılarak da karma STS'ler geliştirilmiştir [2,10,11]. STS performansını geliştirmek için araştırmacılar yeniden konfigüre edilebilir platformlarda STS tasarımlarının gerçekleştirilebileceğini vurgulamışlardır [12-14]. Bu tür bir sistemin yazılım tabanlı sistemlere göre daha iyi performanslı olacağı açıktır. Son yıllarda FPGA kullanılarak geçek zamanlı STS tasarımları araştırmacıların ilgi alanı haline gelmiştir [15 -18].

Bu makalede, bir bilgisayar ağından, gerçek zamanda yakalanan paketlerin saldırı özelliğinde olup olmadığını belirlemek için gerçek zamanlı bir STS'nin; gerçekleştirme, test etme ve sonuçlarını değerlendirme süreçleri açıklanacaktır. Kullanılan bilgisayar ağı IEEE 802.3 Ethernet teknolojisine göre yapılandırılmış olup TCP/IP protokoluna göre çalışmaktadır. Bu makalede kullanılan çerçeve yapısı, IEEE 802.3 standardına göre olup paket ve segment olarak adlandırılan veri bloklarının yapısı ise TCP/IP protokol kümesinde tanımlandığı şekildedir [19].

Gerçekleştirilen STS'de gerek yazılımsal gerekse donanımsal değişikliklerin kolayca yapılabilme ve denenebilme esnekliği ön planda tutulduğundan bir FPGA platformunun kullanılması tercih edilmiştir. FPGA'lar yeniden konfigüre edilebilir donanımlar olduğundan; uygulamalara özel olarak üzerlerinde farklı donanımsal sistemler oluşturulabilir. Hatta FPGA içerisinde donanımsal olarak oluşturulabilecek işlemci veya işlemciler, tasarımcıların uygulamalarına göre programlanabilirler. Bu şekilde oluşturulmuş işlemcilere Soft işlemci, soft işlemcilerin kullanıldığı sistemlere ise SOPC denir. FPGA içerisinde oluşturulan işlemciler, gerçek işlemcilerle (hard işlemci) karşılaştırıldığında; uygulamaya özel basit veya karmaşık bir veya daha fazla işlemcinin tek bir FPGA içerisinde gerçekleştirilebilmesi, güncellenebilmesi, FPGA içerisindeki diğer donanımsal birimler ile (çarpıcı, bölücü, gerçek zamanlı filitre hesaplayıcısı gibi donanımsal hızlandırıcılar) fiziksel devreler eklemeden iletişim kurabilme basitliği ve esnekliği avantajları olarak görülebilir. Ayrıca soft işlemcilerin gerçek işlemcilere göre daha düşük saat frekansı ile çalışmaları onların performanslarının düşük olacağı anlamına gelmez. Çünkü işe özel tasarlanabildikleri için gereken optimum donanımı kullanırlar.

Gerçekleştirilen STS'nin FPGA içerisinde oluşturulmuş donanımsal yapısı şu şekilde açıklanabilir. 1-10/100 Ethernet IP MAC Core modülü: Çerçevelerin ağ ortamından sisteme alınması / gönderilmesini sağlamak için gereklidir. Bu modül bilgisayar ağları için kullanılan 7 katmanlı OSI referans modelinin veri bağı katmanında tanımlanmış görevleri (çerçeve oluşturma, gönderme, alma, hata kontrol bitlerinin üretimi, doğrulanması v.b) yerine getirmek için kullanılacak bir donanımsal yapıdır. FPGA içerisinde gerçekleştirilecek bu donanımı konfigüre ve kontrol etmek için soft işlemci kullanılır. 2- Soft İşlemci: FPGA içerisinde gerçekleştirilir. Uygun şekilde programlanarak hem donanımsal 10/100 Ethernet IP MAC Core modülünü konfigüre ve kontrol eder hem de gerçek zamanlı STS için yazılımı yürütür. 3- DMA, hafıza birimi ve iç bağlantılar: STS sisteminde IP MAC modülü ve işlemcinin hafıza birimini performanslı bir şekilde kullanabilmesini ve sistemin iç ve dış etkileşimini sağlar.

STS'nin ağdan çerçeveleri alabilmesi için, 10/100 Ethernet IP MAC Core modülü işlemci tarafından konfigüre edilmelidir. Ayrıca alınan çerçevelerden paket ve segmentlerin başlık yapıları kullanılarak paket özetlerinin çıkartılması gerekir (paket ön işleme). Paket özetlerini kullanarak YSA algoritmasına göre paketlerin sınıflandırılması

yapılmalıdır (karar mekanizması). Sınıflandırma sonucuna göre paketin hafızadan silinmesi veya 10/100 Ethernet IP MAC Core modülü aracılığı ile ağ'a geri gönderilmesi gerekir. Bu işlemler sistemin yazılımsal yapısıyla oluşturulur. Bu yazılımsal yapı için soft işlemci; C dilinde yazılan bir program vasıtasıyla programlanmalıdır [20].

STS, üretici ALTERA firmasının Cyclone III EP3C40F484C7 FPGA tabanlı platformunda gerçekleştirilmiştir. Bu platform; Ethernet arabirimi (RJ45 portu), FPGA entegresi, bilgisayarla haberleşme birimi, hafıza birimleri, değişik giriş çıkış ünitelerinin olduğu bir kart ve Quartus 8.0 ve NIOS 8.0 yazılım araçlarından oluşur.

Yukarıda özetlenen STS'nin gerçekleştirme ve bilgisayar ağına bağlanarak, gerçek zamanda çerçeve alma, değerlendirme geri gönderme işlemleri ve test sonuçları ilerleyen bölümlerde detaylı olarak açıklanacaktır.

2. GELİŞTİRİLEN STS'NİN MİMARİSİ (ARCHITECTURE OF THE DEVELOPED IDS)

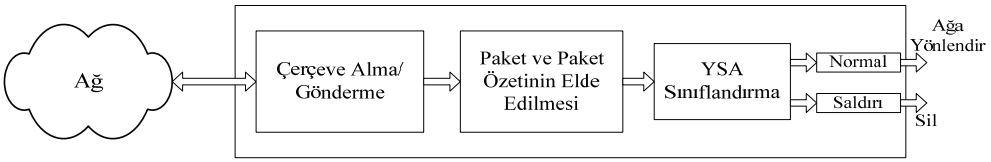
Ağdan akan paketlerin saldırı özelliğinde olup olmadığının tespiti için gerçekleştirilen gömülü STS Şekil 1' de gösterildiği gibi 4 birimden oluşmaktadır.

Bu birimler çerçeve yakalama, çerçevelerden paketlerin ve paket özetinin elde edilmesi, paketlerin sınıflandırılması ve karar mekanizmasıdır.
Sistemin ilk birimi olan çerçeve yakalama işlemi programlanabilir bir donanım yapısında gerçekleştirilmiştir. Bu donanım biriminin nasıl gerçekleştirildiği bir sonraki bölümde açıklanmıştır. Diğer üç birim ise bu programlanabilir donanım üzerinde çalışabilen bir yazılım ile gerçekleştirilmiştir.

Ağdan alınan bir çerçeveden, paketlerin ve sınıflandırma için uygun paket özetinin elde edilmesi sistemin ikinci birimini oluşturmaktadır. STS'nin sınıflandırma yapabilmesi için paket özetinin ve sınıflandırma yönteminin belirlenmesi ve gerçek zamanda uygulanabilir olması kritik bir konudur. Paketlerin değişik kriterlere göre değerlendirilip sınıflandırılması için tamamını değerlendirmek yerine paket özetini kullanmak hızlı çalışma açısından daha uygundur. Özet için paketlerin belirleyici birtakım özellikleri ve istatistiksel bilgileri kullanılabilir. Bu makalede paket özeti için başlık bilgileri protokol tipi, paket boyutu, kaynak ve hedef IP numaraları, kaynak ve hedef port numaraları alınmıştır. Paket özetinin sınıflandırma algoritmasında giriş verisi olarak kullanılabilmesi için Tablo.1'e göre düzenlenmesi ve normalize edilmesi gerekir[4]. Normalize değerler için kullanılan sözcük bit sayıları için bir sınırlama olmamakla beraber burada kullanılan bit sayıları konunun anlaşılması açısından küçük tutulmuştur. Örneğin gelen bir paketin hedef IP'si 10.47.2.0 ile 10.47.2.31 arasında ise bunun Tablo 1'e göre normalize değeri 000 olarak hesaplanır.

Alınan paketten normalize edilmiş paket özetinin oluşturulması için Algoritma.1'deki işlemler yapılmalıdır. Bu algoritma, 3. Bölümde anlatılan yazılımda kullanılarak gerçek zamanda gerçekleştirilecektir.

Algoritma 1. [4]:
1. Tespit sisteminde kullanılacak kodlama biçimini belirle.
2. Çerçeveden 2. katman başlık yapısını atarak paketi elde et.



Şekil 1. Gerçekleştirilen STS'nin prensip şeması (Principle schema of the implemented IDS)

Tablo 1. Paket özeti ve normalizasyon (Packet summary and normalization)

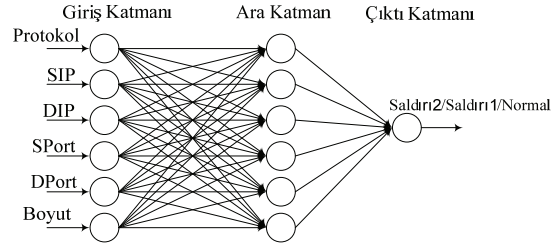
Özellik	Norm.değeri	Paket özellikleri ve özet değerler
Protokol Tipi	2 bits	**00**: TCP, **01**:UDP, **10**:ICMP, **11**; Diğer
Paket Boyutu	2 bits	64 ≤ **00** < 256, 256 ≤ **01** < 512, 512 ≤**10** <1024, **11** ≥ 1024
Kaynak IP	2 bits	**00**: A Sınıf, **01**:B Sınıf, **10**:C Sınıf, **11**: Diğer
Hedef IP	3 bits	**000**: 10.47.2.0 – 10.47.2.31 **001**: 10.47.2.32 – 10.47.2.63 **010**: 10.47.2.64 – 10.47.2.95 **011**: 10.47.2.96 – 10.47.2.127 **100**: 10.47.2.128 – 10.47.2.159 **101**: 10.47.2.160 – 10.47.2.191 **110**: 10.47.2.192 – 10.47.2.223 **111**: 10.47.2.224 – 10.47.2.255
Kaynak TSAP	8 bits	**00000000**: 0, **00000001**:1… **11111110**:254, **11111111**:Diğer
Hedef TSAP	8 bits	**00000000**: 0, **00000001**:1… **11111110**:254, **11111111**:Diğer

Tablo 2. Rastgele oluşturulmuş eğitim paketleri(Randomly generated training packet)

Protokol tipi	Paket boyutu	Kaynak IP(SIP)	Hedef IP (DIP)	Kaynak TSAP (SPort)	Hedef TSAP (DPort)	Sınıfı
00	11	00	010	00010001	10001001	Normal
00	00	00	000	01010000	11111111	Normal
00	01	01	101	00010111	11111111	Saldırı1
01	10	11	000	10101000	10101000	Normal
11	11	11	101	00000001	00010001	Saldırı2

3. Paketin IP başlık yapısından protokol tipini, IP numaralarını ve boyut bilgilerini kullanarak Tablo 1'e göre normalize değerleri oluştur.
4. Paketin başlığını soyarak ulaşım katmanında kullanılan segmenti elde et. Bunun başlık yapısını değerlendirerek ulaşım katmanı protokol tipini belirle, TSAP adreslerini kullanarak Tablo 1'e göre normalize değerleri oluştur.
5. Tablo 1' e göre elde edilen paket özetini adreslenmiş hafıza bölgesine yaz.

Elde edilen paket özetlerine göre sınıflandırmanın yapılıp paketin saldırı olup olmadığını belirlemek için YSA modeli kullanılmıştır. Kullanılan YSA yapısı Şekil 2' de verilmiştir [20]. Sistemde giriş katmanı, ara katman ve çıkış katmanı olmak üzere toplam 3 katman bulunmaktadır. Giriş katmanı önceden elde edilmiş paket özetini girdi olarak alır. Kullanılan sınıflandırma sistemi 3 sınıflıdır. YSA'nın çıkışı, "Saldırı1", "Saldırı2" saldırısı ve "Normal" etiketlerinden birini ifade eder. Genel olarak saldırılar paketin başlık yapısının analiz edilmesiyle belirlenebileceğinden dolayı bu makalede sınıflandırma için sadece paket başlık yapısı kullanılmıştır.



Şekil 2. Önerilen YSA'nın yapısı (The proposed ANN architecture)

6 giriş, 1 ara katman ve 1 çıkış katmanı bulunan YSA yapısının eğitilmesi için giriş ve ara katmanda Hiperbolik Tanjant Sigmoid, çıktı katmanında Lineer Transfer fonksiyonları kullanılmıştır. Giriş ve ara katmanda 6, çıktı katmanında 1 nöron kullanılmıştır. Ağın eğitilmesinde geriye yayılım algoritması kullanılarak ağırlıklar güncellenmiştir. Ağın eğitimi için hata değeri olarak 0,01 ve öğrenme oranı olarak da 0,005 değerleri alınmıştır. YSA'nın eğitilmesi bilgisayar ortamında C programlama dilinde yazılan bir programla off-line olarak yapılmıştır. YSA'nın eğitilmesi ile elde edilen katmanlar arasındaki ağırlık değerleri, bias değerleri, v.b değerler 3.Bölümde açıklanan gerçek zamanlı çalışma için geliştirilen yazılımda sabit değerler olarak kullanılacaktır.

YSA'nın eğitilmesi için 1000 adet eğitim paketinin özeti ve normalizasyonu kullanılmıştır. Tablo 2. eğitimde kullanılan paket özetlerinin ve sınıflarının bir bölümünü göstermektedir. Tablo 2'deki Protokol tipi, Paket boyutu, Kaynak ve Hedef IP, Kaynak ve hedef TSAP adreslerinin ikili sayısal değerleri Tablo 1'e göre verilmiştir.

Kullanılan eğitim paketlerinden 509 adedi Normal, 195 adedi Saldırı2 ve 296 adedi Saldırı1 saldırısı olacak şekilde oluşturulmuştur. Tablo 3 sınıflandırma başarımını belirlemek için kullanılan Confusion Matrisini, sınıflandırmanın Doğruluk, Duyarlılık ve Yanlış Alarm Oranının (FNR-False Negative Rate) tanımını, Tablo 4. ise sınıflandırılmış paketlerin dağılımını göstermektedir. "Saldırı1" ve "Saldırı2" saldırıları ile "Normal" paketlerin sınıflandırılması sonucunda elde edilen Doğruluk, Duyarlılık ve FNR değerleri Tablo 5' te verilmiştir. Örneğin; 509 normal paketin sadece 502 adedi normal olarak tespit edilmesine rağmen "Saldırı1" sınıfındaki 2 paket ile "Saldırı2" sınıfındaki 1 paket "Normal" olarak tespit edilmiştir. Dolayısı ile Normal sınıfı için TP değeri 502, FN değeri 3 alınarak duyarlılık değeri 0,994 olarak hesaplanır.

Tablo 3. Eğitim verileri için Confusion Matrix (Confusion Matrix for training data)

		Gerçek	
		Pozitif	**Negatif**
Tahmin	**Pozitif**	Doğru Pozitif(TP)	Yanlış Pozitif(FP)
	Negatif	Yanlış Negatif(FN)	Doğru Negatif(TN)
Başarım parametreleri			
Duyarlılık= $\frac{TP}{TP+FN}$, Doğruluk= $\frac{TP}{TP+TN}$, FNR= $\frac{FN}{TP+FN}$			

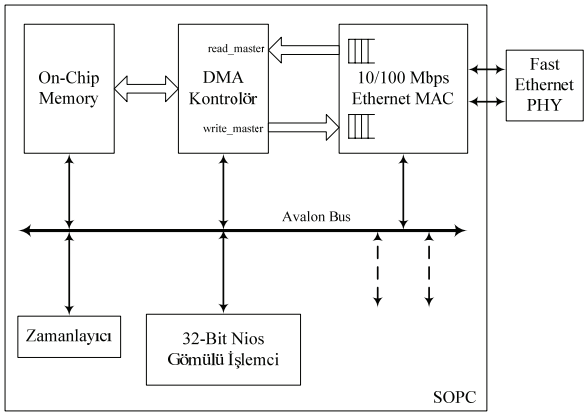
Tablo 4. YSA'ya göre sınıflandırma (Classification according to ANN)

	Saldırı1	**Saldırı2**	**Normal**
Saldırı1	292	0	5
Saldırı2	2	194	2
Normal	2	1	502

Tablo.5 Sınıflandırma başarımları (Classification performance)

	Saldırı1	**Saldırı2**	**Normal**
Doğruluk	%98,6	%99,5	%98,6
Duyarlılık	%98,3	%97,9	%99,4
FNR	%1,7	%2,1	%0,6

STS'nin son birimi ise karar mekanizmasıdır. Karar mekanizması YSA'nın çıkışından alınan sınıflandırma bilgisine göre çalışır ve paketleri eğer saldırı ise ağdan düşürür. Paketlerin normal olması durumunda çerçeve içine konularak, çerçeve yakalama/gönderme birimi aracılığı ile ağa yönlendirilir.



Şekil 3. Gerçekleştirilen STS'nin Mimarisi [20] (Architecture of the implemented IDS)

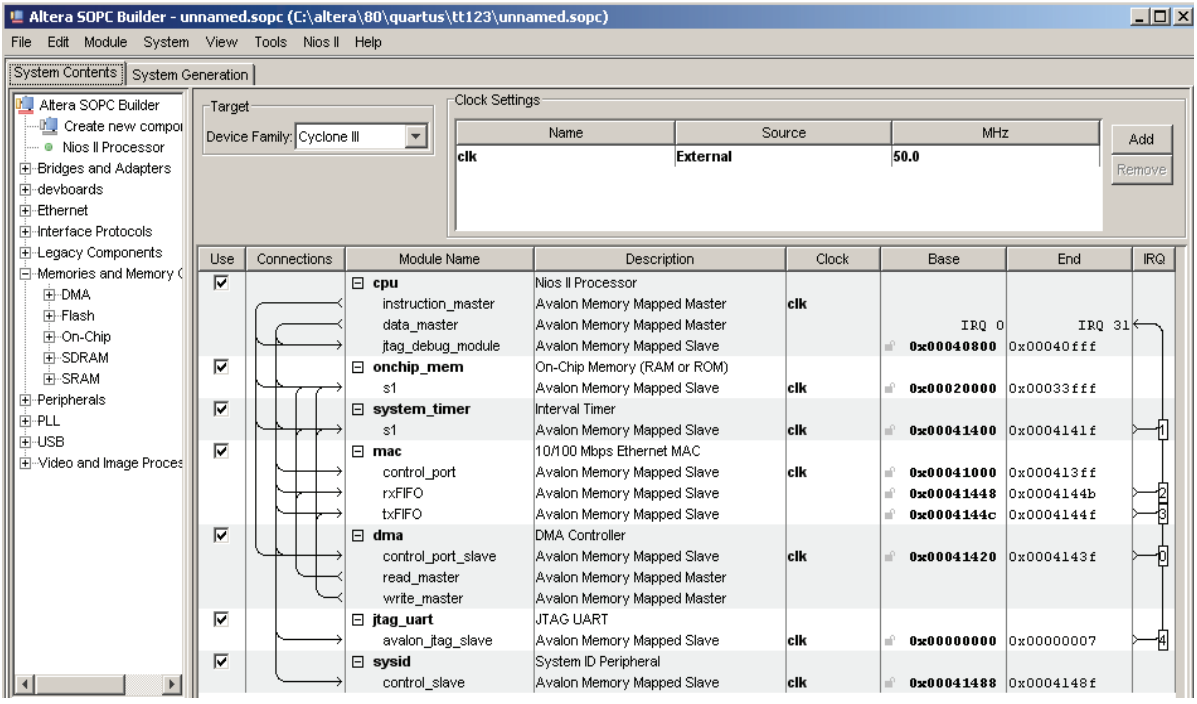
3. GÖMÜLÜ STS TASARIM SÜRECİ (EMBEDDED IDS DESIGN PROCESS)

İkinci bölümde temel yapısı verilen STS'nin gerçekleştirilmesi iki aşamadan oluşur. Birinci aşama; FPGA üzerinde Nios işlemci (Üretici Altera firmasının soft işlemci için kullandığı isim), 10/100 Ethernet MAC Core modülü, DMA, hafıza gibi donanım birimlerinin oluşturularak Avalon Bus sistemi ile birbirleriyle haberleşebilmesinin sağlanmasıdır. Bu durum Şekil 3'te verilmiştir. Buna donanım oluşturma süreci denir. İkinci aşama ise; oluşturulmuş bu donanımın (SOPC), programlanarak, 2.Bölümde bahsedildiği gibi STS'nin gerçekleştirilmesidir. Bu ise sistemin programlanma sürecidir.

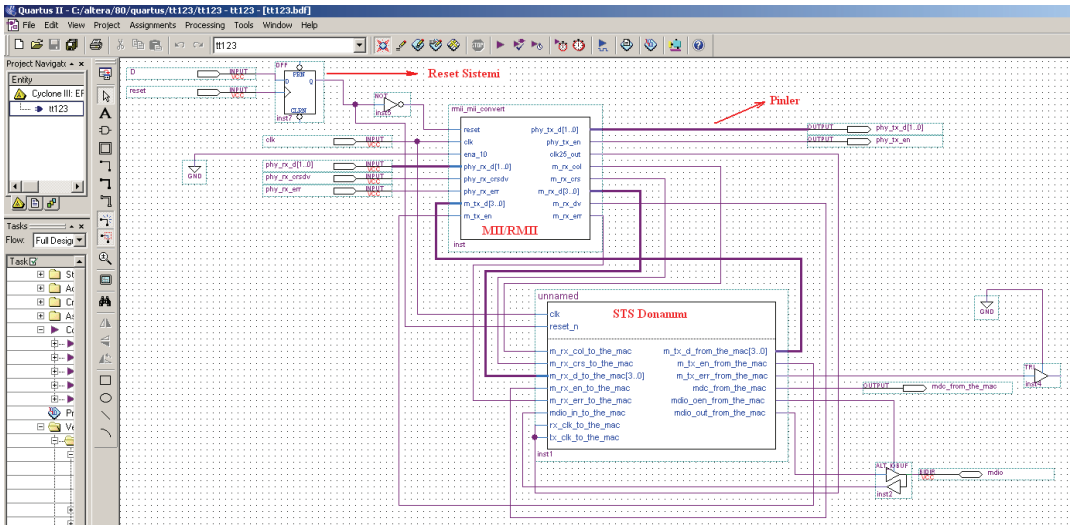
3.1 Donanım Oluşturma Süreci (Process of The Hardware Generation)

Şekil 3. teki STS donanımını elde etmek için tüm birimler SOPC Builder yazılım ortamında seçilip uygun şekilde birleştirilir. STS'nin kullanacağı kesmeler, hafıza haritası v.b belirlenir. Elde edilen bu sistem derlenir, hatalardan arındırılarak Şekil 4.'deki SOPC yapısı elde edilir.

SOPC Builder ile oluşturulmuş STS donanımı, Quartus 8.0 ortamına aktarılarak, varsa kullanıcı tanımlı çevre birimleriyle (Reset, Media Independent Interface/Reduced Media Independent Interface gibi) bağlantısı gerçekleştirilir. Son olarak STS gömülü sisteminin pin bağlantıları yapılır. Sistem derlenerek hatalar giderilip FPGA'ya gömülecek duruma getirilmiş olunur. Bu yapı Şekil 5'te verilmiştir. Bu ana kadarki olan işlemler Alteranın Nios ve Quartus yazılım araçları kullanılarak bilgisayar ortamında oluşturulmuştur. Bilgisayar ortamında oluşturulmuş bu yapının Cyclone III tabanlı karttaki FPGA ya gömülüp donanımsal devrenin oluşturulması, bilgisayar ve kart arasındaki iletişimi yapan JTAG arabirimi ile sağlanmıştır. Şekil 6'daki rapor STS donanımının FPGA içerisinde oluşturulması için FPGA'da harcanan donanımsal birimleri göstermektedir. Buna göre STS için FPGA' da toplam 7.875 adet Lojik Eleman, 4893 adet kayıtçı, 793.204 bit hafıza alanı, 4 adet çarpıcı devre kullanılmıştır.



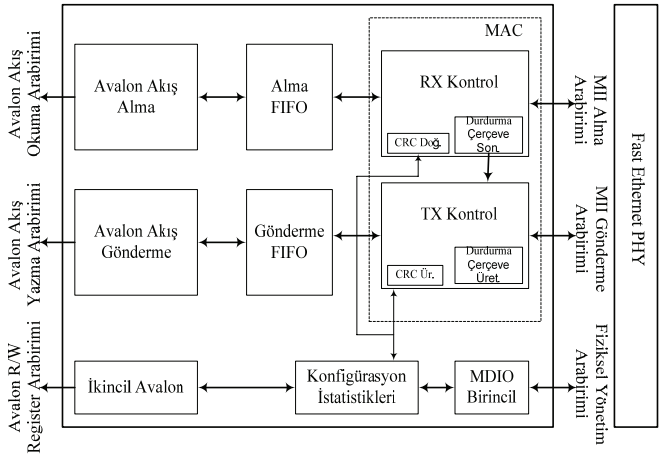
Şekil 4. STS donanımının bağlantı ve özellikleri (Connection and features of the IDS hardware)



Şekil 5. Quartus 8.0'da STS mimarisi [20] (The deployed IDS architecture on Quartus 8.0)

Flow Status	Successful - Sat Jan 30 14:04:32 2010
Quartus II Version	8.0 Build 215 05/29/2008 SJ Web Edition
Revision Name	tt123
Top-level Entity Name	tt123
Family	Cyclone III
Device	EP3C40F484C7
Timing Models	Final
Met timing requirements	N/A
Total logic elements	7,875 / 39,600 (20 %)
Total combinational functions	6,711 / 39,600 (17 %)
Dedicated logic registers	4,893 / 39,600 (12 %)
Total registers	4893
Total pins	17 / 332 (5 %)
Total virtual pins	0
Total memory bits	793,204 / 1,161,216 (68 %)
Embedded Multiplier 9-bit elements	4 / 252 (2 %)
Total PLLs	0 / 4 (0 %)

Şekil 6. STS' nin derlenmesi ve FPGA' ya gömülmesi ile elde edilen sonuç raporu (Compilation of the IDS and final report obtained by embedding to FPGA)



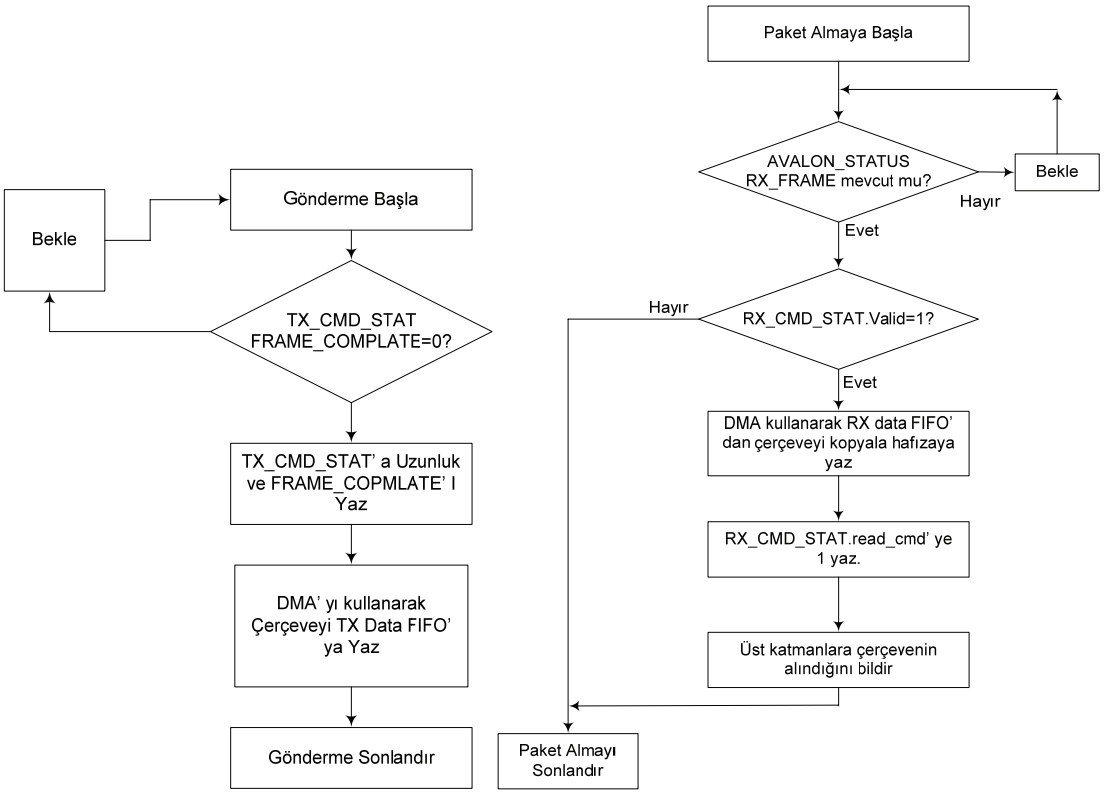
Şekil 7. MAC modülünün blok yapısı [21] (Block structure of the MAC Core)

FPGA üzerinde oluşturulmuş bu STS donanımsal yapının çalışabilmesi için sistemde kullanılan bazı birimlerinin doğru bir şekilde konfigüre edilmesi ve programlanması gerekir. Bu birimlerden en önemlisi, STS ile ağ arasında çerçeve alma/gönderme işlemini gerçekleştirecek 10/100 Ethernet MAC Core'dur. 10/100 Ethernet MAC Core'un blok yapısı Şekil 7'deki gibidir.

MAC Core modülü oluşturan birimler, STS mimarisine uygun olarak FPGA içerisinde donanımsal olarak gerçekleştirilmiş olup yazılımsal olarak konfigüre edilmiştir. Bu donanımın konfigüre edilmesindeki amaç çerçeve alma/gönderme işleminin doğru şekilde yapılmasıdır. Ağdan gelen veya giden çerçevelerin, OSI modelinin veri bağı katmanında kullanılacak biçime dönüşümü kart üzerindeki DP83640 fiziksel arabirim entegresi (Fast Ehernet PHY) ile gerçekleştirilir. Şekil 7'de görüldüğü gibi bu entegrenin çıkışları 10/100 Ethernet MAC Core'un MAC birimi ile ilişkilidir.

MAC Core modülünün, OSI veri bağı katmanındaki MAC alt katmanını tanımlayan IEEE 802.3'e göre çerçeve alma/gönderme işlemini yapabilmesi için konfigüre edilmesi gerekir. Bu işlem, Şekil 8' deki akış şemalarıyla ifade edilmiştir.

Çerçeve gönderme işlemi iki adımda gerçekleştirilir. Bu adımlardan ilki 10/100 Ethernet MAC Core'un TX_CMD_STAT kayıtçısının, çerçeve uzunluğunu gösteren bitleri ve çerçeve tamamlama (FRAME_COMPLETE) bitleri uygun değerlerle yüklenir. İkinci adımda ise DMA kontrolör yardımıyla hafıza uzayında (On Chip Memory) oluşturulmuş çerçeve, MAC modülünün içerisindeki Gönderme FIFO' ya kopyalanır. Bu süreç Şekil 8.(a)'da verilmiştir. Bu işlem, donanımsal gerçekleştirilme süreci yukarıda anlatılmış olan 10/100 Ethernet MAC Core modülünün, çerçeve gönderebilmesi için gerekli olan konfigürasyon işlemidir. Bundan sonraki süreçte, çerçeveye, Tx kontrol ünitesinde gerekli kontroller ve eklemeler (CRC bitleri v.b) yapılarak, çerçeve fiziksel arabirime gönderilir.



Şekil 8. (a) Çerçeve gönderme (b) Çerçeve alma ((a) Transmiting Frame, (b) Receiving Frame)

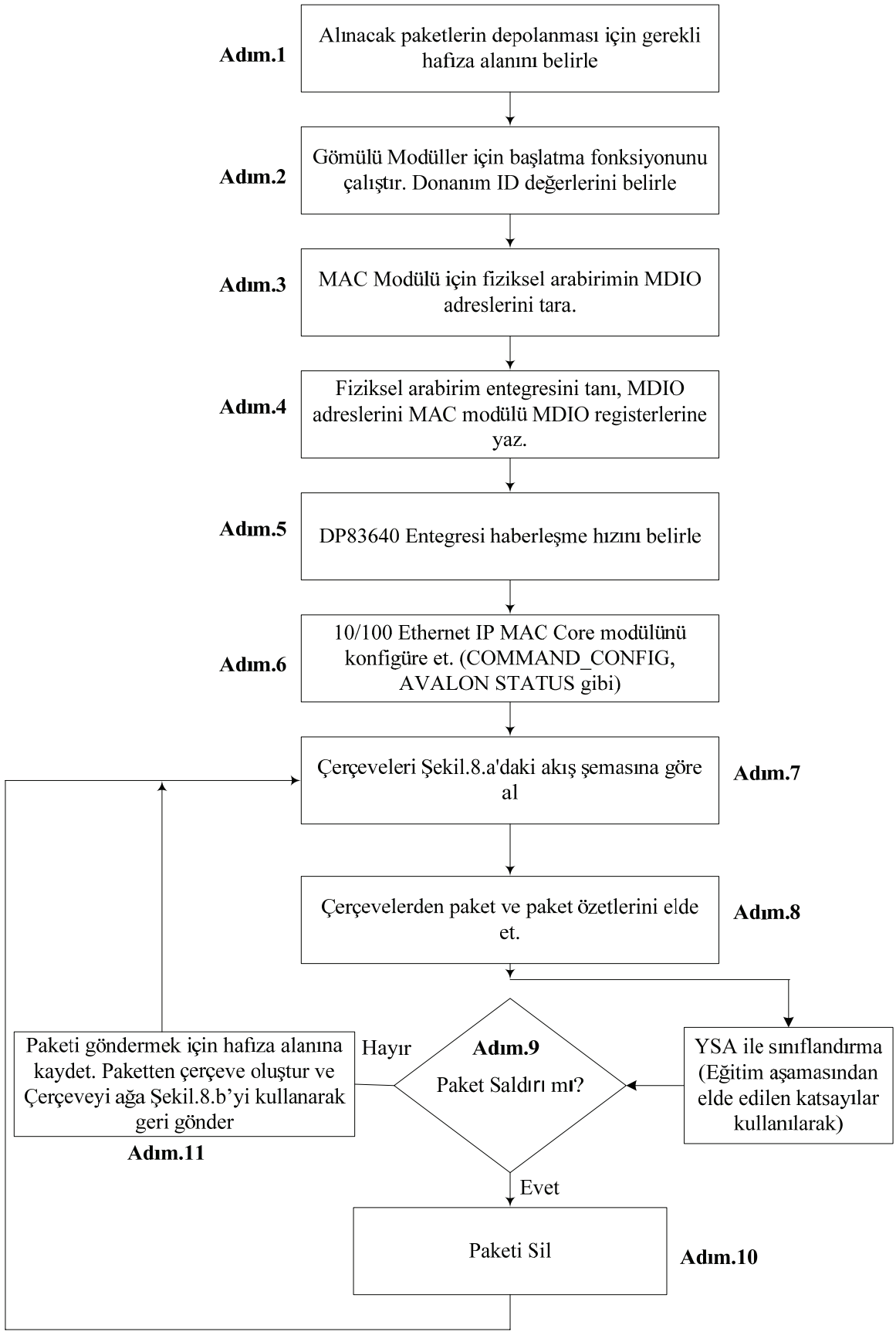
10/100 Ethernet MAC Core modülündeki çerçeve alma işlemi üç adımda gerçekleştirilir. Bu süreç Şekil 8.b'de verilmiştir. Bu adımlardan ilkinde AVALON_STATUS kayıtçısının RX_FRAME_AVAILABLE biti kontrol edilir. Çerçeve mevcutsa ikinci adıma geçilir. Bu adımda, Rx kontrol ünitesinden alınan çerçeve Alma FIFO'ya yazılır. DMA kullanılarak Alma FIFO'daki çerçeve Avalon veri yolu üzerinden hafızaya (SRAM, On-Chip Memory v.b) yazılır. Son adımda ise çerçevenin alındığını gösteren ilgili kayıtçılar (çerçevenin alınma işlemi bitene kadar), yeni bir çerçevenin alınmasını önlemek için uygun değerlerle yüklenir. Bu işlemler, donanımsal 10/100 Ethernet MAC Core modülünün, çerçeve alabilmesi için gerekli olan konfigürasyon işlemidir.

10/100 Ethernet MAC Core modülünün çerçeve alma ve göndermesi için gerekli olan konfigürasyon işlemleri, FPGA'da oluşturulmuş olan RISC mimarisine sahip Nios işlemcinin programlanmasıyla başarılır. Nios işlemci genel amaçlı, 512 dahili saklayıcıya sahip 32 bitlik bir işlemcidir [22]. Bu işlemci, 10/100 Ethernet MAC Core'un konfigürasyonun yanı sıra, çerçevelerin alınması/ gönderilmesi sürecinde önemli rol oynayan DMA'yı da konfigüre ve kontrol eder.

3.2. Yazılım Oluşturma Süreci (Process of The Software Generation)

Yukarıda bahsedilen konfigürasyon, paket özeti çıkarma, paket sınıflandırma ve karar verme işlemleri için gerekli programlar C/C++ programlama ortamında yazılarak derlenip Nios işlemcinin program hafızasına yüklenmiştir. Bu programın akış şeması Şekil 9'da verilmiştir.

Adım 1: Alınan ve gönderilen paketlerin, adresleri tanımlanmış bir hafıza alanında tutulması gereklidir. Bunun için 8 KByte'lık bir hafıza alanı belirlenmiştir.
Adım 2: Sisteme yüklenecek yazılımın çalışabilmesi için işlemci ve diğer birimlerin başlatılması gereklidir. Bu amaç için bir başlatma fonksiyonu yazılmıştır. Bu fonksiyon SOPC tarafından her bir donanım birimine verilen başlangıç adres değerini ID numarası olarak belirler. Bu ID numaraları kullanılarak donanım birimleri arasında veri aktarımı yapılır.
Adım 3: 10/100 Ethernet MAC Core modülünün fiziksel arabirim ile iletişim kurabilmesi için aktif fiziksel arabirim aranır.
Adım 4: Fiziksel katman görevlerini yerine getiren DP83640 entegresi tanınır. MDIO adresleri 10/100 Ethernet MAC Core' un MDIO_0 ve MDIO_1 kayıtçılarına yazılır.
Adım 5: DP83640 entegresinin 10/100 Mbps hızlarından birisi belirlenir.
Adım 6: 10/100 Ethernet MAC Core' un çerçeve alma ve gönderme yapabilmesi için ilgili kayıtçılar konfigüre edilir (COMMAND_CONFIG v.b.) [23].



Şekil 9. Gömülü STS'nin programlanması adımları[20] (Programming steps of the embedded IDS)

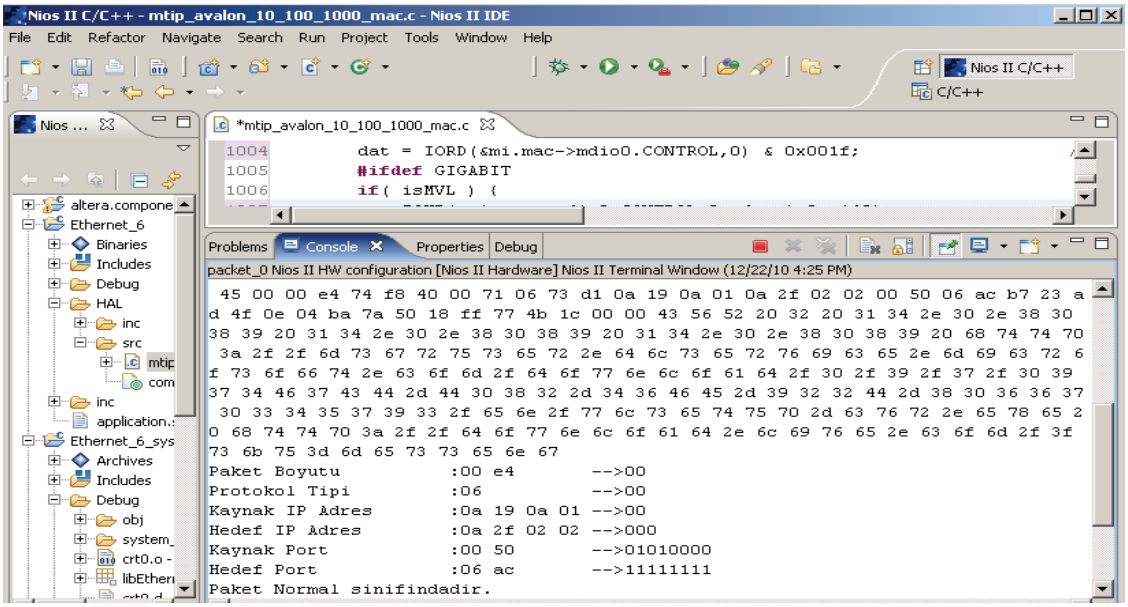
Adım 7: Ağdan alınacak çerçeveler Şekil 8.(a)' daki akış şemasına göre alınır.
Adım 8: Paketlerin özeti, 2. Bölümde açıklanan Algoritma 1'e göre elde edilir.
Adım 9: Elde edilen paket özeti kullanılarak, 2.Bölümde açıklanan YSA sınıflandırma algoritmasına göre paketlerin sınıfları belirlenir.
Adım 10: Eğer paketin sınıfı saldırı olarak belirlenmiş ise hafızadan silinir. 7. Adım'a dönülerek yeni çerçeve beklenir veya alınır.
Adım 11: Eğer paket normal sınıfına ait ise program hafızasındaki paketten çerçeve oluşturulur, çerçeve Şekil 8.(b)' deki akış şemasına göre tekrar ağa yönlendirilerek 7. adıma dönülür ve yeni çerçeve beklenir varsa alınır.

4. STS'NİN TEST EDİLMESİ (TESTING OF THE IDS)

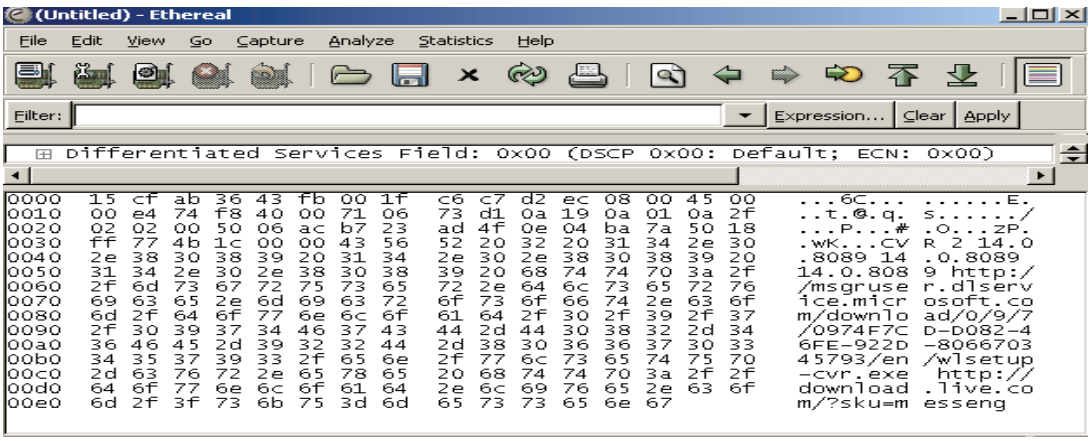
FPGA' da donanımsal olarak gerçekleştirilmiş olan STS; akış şeması yukarıda açıklanan yazılıma göre programlanarak, gerçek zamanda çalışır vaziyete getirilmiştir.

Gömülü STS' nin test aşamasında 2000 adet paket ColaSoft Packet Builder yardımıyla rastgele oluşturulmuştur. Oluşturulan paketler ardışıl olarak ağdan STS' ye gönderilmektedir. STS gerçek zamanda bu paketleri işleyip saldırı ise silmekte, normal ise ağa geri göndermektedir.

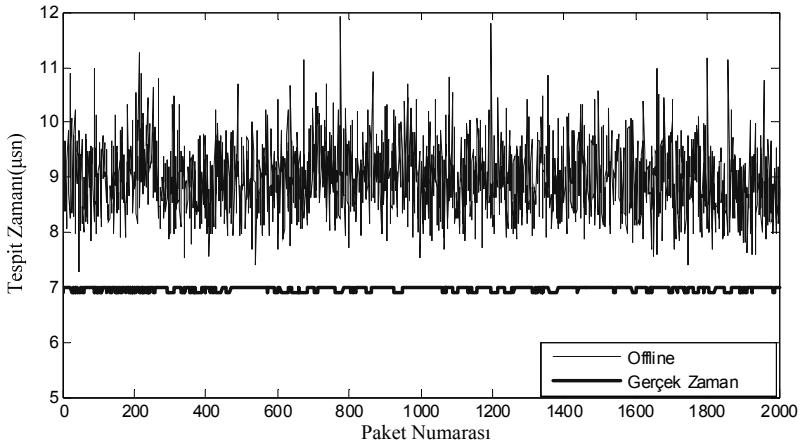
Şekil 10, ağdan gömülü sisteme gönderilen ve sistem tarafından yakalanan bir test paketinin başlık yapısı, verisi, elde edilen paket özeti ve YSA sınıflandırmasına göre pakete atanmış sınıf etiketini göstermektedir. Ağdan alınan paket normal olduğundan dolayı paket çerçeve yapısına dönüştürülerek tekrar ağa gönderilmiştir. Şekil 11 ise ağa gönderilen bu normal paketin Etheral programı ile bir çerçeve yapısında tekrar yakalandığını göstermektedir.



Şekil 10. Alınan bir paketin Nios 8.0' da özet ve sınıfının gösterimi (Summary of the a received packet and the class representation in Nios 8.0)



Şekil 11. Etheral ile ağdan yakalanan normal pakete ait çerçeve (The frame belonging to normal packet captured from network by Etheral)



Şekil 12. Paket sınıflarının tespit zamanı değişimleri (Changes of detection time of the packet classes)

Gerçek zamanda yapılan testlerde kullanılan 2000 paketin, STS tarafından sınıflarının tespit edilme sürelerinin değişimi Şekil 12'de verilmiştir. Gömülü sistem sadece sınıflandırma için programlandığından dolayı saldırı ve normal paketlerin tespit zamanları aynı zaman sürelerinde tespit edilmiştir. Ayrıca aynı test paketleri kullanılarak 3GB RAM, 3.0GHz işlemciye sahip bir PC'de yazılan C programı ile sistem offline olarak test edilmiştir. İşlemcinin sınıflandırma işlemlerinden başka yapması gereken işlerden dolayı, tespit süresi Şekil 12 de görüldüğü gibi gerçek zamanlı çalışmaya göre daha yüksek çıkmıştır. Tablo 6 her iki test içinde elde edilen sınıflandırma sonuçlarını göstermektedir.

Tablo 6. Gerçek zamanlı ve offline çalışma için sınıflandırma sonuçları (The results of real time and offline classification)

	Normal	Saldırı1	Saldırı2
Gerçek Zamanlı	1311	392	297
Offline	1311	392	297

5. SONUÇLAR (CONCLUSION)

Bu makalede gerçekleştirilen STS sistemi, ağla haberleşmeyi sağlayan 10/100 Ethernet MAC Core modülünün konfigürasyonu ile çerçevenin ağdan alınıp OSI modeline göre üst katmanlara yükseltme sürecinin her aşamasına hakim olma açısından çok önemlidir. Ayrıca FPGA içerisinde oluşturulan SOPC gerçek zamanlı birçok algoritmanın kolaylıkla denenmesine imkan verecek şekilde oluşturulmuştur. Sistemde her türlü yazılımsal ve donanımsal değişiklik kolaylıkla yapılabilir. Sistemin bir diğer avantajı da, gerçek zamanlı yeni STS uygulamalarının oluşturulmasına esnek bir altyapı oluşturmasıdır. STS' lerin güvenilir ve hızlı bir şekilde paket sınıflandırma yapabilmesi için hem uygun bir sınıflandırma algoritmasının seçilmesi ve algoritmanın hızlı bir şekilde çalışabilmesini sağlayacak yazılımın oluşturulması hem de ilgili donanımsal devrenin gerçek zamanlı çalışmayı desteklemesi gerekir. FPGA ortamında oluşturulan Nios işlemcisi ile STS sistemi denetlenebilir haldedir. Dolayısıyla oluşturulan bu soft işlemcinin ayrıca yazılımsal olarak da programlanabilmesi, bu tür sistemlerin geliştirilmesi aşamasında çok önemli bir esneklik sağlamıştır.

TEŞEKKÜR (ACKNOWLEDGEMENT)
Bu çalışma Türkiye Bilimsel ve Teknolojik Araştırma Kurumu tarafından 108E166 numaralı proje ile desteklenmiştir.

KAYNAKLAR (REFERENCES)

1. El-Semary, A., Edmonds, J., Gonzalez, J., Papa, M., "A Framework for Hybrid Fuzzy Logic Intrusion Detection Systems", The 2005 IEEE International Conference on Fuzzy Systems, Reno, Nevada, USA, 325-330,2005
2. Hwang, K., Cai, M., Chen Y., Qin, M., "Hybrid Intrusion Detection With Weighted Signature Generation Over Anomalous Internet Episodes", IEEE Transaction on Dependable and Secure Computing, Vol:4, No:1, 41-53, 2007.
3. T.Tuncer, Y.Tatar, "Programmable Embedded System Design:A Study on Sniffer " 3rd International Conference Application of Information and Communication Technologies, Baku,Azerbaijan, 1-5, 2009.
4. T. Tuncer, Y.Tatar, "FPGA Based Programmable Embedded Intrusion Detection System", 3rd International Conference on Security of Information and Networks, Taganrog, Russia, 245–248, 2010.
5. Bidgoli, B.M., Analoi, M., Rezvani, M.H., Shahhoseini H.S., "Performance Evaluation of Decision Tree For Intrusion Detection Using Reduce Feature Spaces", Trends in Intelligent System and Computer Engineering, LNCS 1876, 6, 273-284, 2008.
6. Ohta, S., Kurebayashi, R., Kobayashi, K., "Minimizing False Positives of a Decision Tree Classifier for Intrusion Detection on the Internet",

Journal of Network and System Management, Vol:16 ,No:4, 399-419, 2008

7. Owais, S., Snasel, V., Kromer, P., Abraham, A., "Survey: Using Genetic Algorithm Approach in Intrusion Detection System Techniques", 7th Computer Information System and Industrial Management Applications, Ostrava, The Czech Republic, 300-307, 2008.

8. Amor, N.B., Benferhat, S., Elouedi, Z., "Naïve Bayes vs Decision Trees in Intrusion Detection System", Proceedings of The ACM Symposium on Applied Computing, Nicosia, 420-424, 2004.

9. Shun, J., Malki, A.H., "Intrusion Detecting System Using Neural Networks", Fourth International Conference on Natural Computation, Jinan, China, 242-246, 2008.

10. Jahankhani, H., Hessami A.G., Hsu, F., Beqiri, E., "Neural Networks for Intrusion Detection System", Communications in Computer and Information Science, LNCS 45, 156-165, 2009.

11. Ali A.M., Zalim, H. A., Ceylan K.G., "A Hybrid Intrusion Detection system Design for Computer Network Security", Computer & Electrical Engineering, Vol:35, No:3, 517-526, 2009.

12. Kang, D-H., Kim, B-K., Oh, J-T., Nam, T-Y., Jang, J-S., "FPGA Based Intrusion Detection System Against Unknow and Know Attacks", LNCS 4048, 801-806, 2009.

13. Clark, C.R., Ulmer, C.D., Schimmel, D.E., "An FPGA-Based Network Intrusion Detection System on-chip Network Interface", International Conference Computer Engineering & System, 268-273, 2006.

14. Lee, J., Hwang, S.H., Park, N., Lee, S-W., Jun, S., Kim, Y.S., "A High Performance NIDS Using FPGA-Based Regular Expression Matching", Symposium on Applied Computing, Seoul, Korea, 1187-1191, 2007.

15. Weaver N., Paxson, V., Gonzalez, M.J., "The Shunt an FPGA-Based Accelerator for Network Intrusion Prevention", Proceedings of the 2007 ACM/SIGDA 15th International Symposium on Field Programmable Gate Arrays, Monterey, USA, 199-206, 2007.

16. Jorge, B., Carlos T. C., "A Low-Cost Embedded IDS to Monitor and Prevent MAN-in-the-Middle Attacks on Wired LAN Environments", International Conference on Emerging Security Information System and Technologies, Valencia, Spain,122-127,2007.

17. Das, A., Nuguyen, D., Zambreno, J., Memik, G., Choudhary, A., "An FPGA-Based Network Intrusion Detection Architecture", IEEE Transaction on Information Forensics and Security, Vol:3, No:1, 118-132, 2008.

18. Narayan, R., Hanbo, D., Memik, G., Choudhary, A., Zambreno, J., "An FPGA Implementation Decision Tree Classification", Proceedings of the Conference on Design Automation and test in Europa Nice, France, 189-194, 2007.

19. Tanenbaum, A.S., "Computer Networks", Prentice Hall, New Jersey, 2003.

20. Tuncer T., "Bilgisayar Ağları İçin Saldırı Tespit Sistemi Tasarımları ve FPGA Ortamında Gerçekleştirilmesi", Doktora Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, 2010.

21. Morethanip, "10/100 Ethernet MAC Core for Avalon Reference Guide", Germany, 2010.

22. Altera Corporation, "Nios II Hardware Development Tutorial", 2007.

23. Khan, S., "Basic Input/Output Functions Morethanip", Germany, 2005.