**KARAMANOĞLU MEHMETBEY ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**TCP VE UDP**

**FİNAL ÖDEVİ**

**Hakan ÇELEBİ**

**Mühendislik Fakültesi**

**Bilgisayar Mühendisliği Bölümü**

**Ders Hocası: ÖĞR. GÖR. KEREM GENCER**

İÇİNDEKİLER

1. **TCP VE UDP**
   1. TCP
   2. UDP
2. **PORT FORWARDİNG**
   1. Windows Command Prompt
   2. Linux Terminal
   3. Router Interface
3. **SOCET PROGRAMING**
   1. C++
4. **TCP VE UDP**

TCP/IP Modelinde veya OSI Modelinde Ulaşım Katmanı (Trasports Layer) içerisinde

geliştirilmiş protokollerdir.

1. **TCP**

1974 yılında ‘Institute of Electrical and Electronics Engineers’(IEEE) tarafından

yayımlanan ve Vint Cerf ve Bob Khan tarafından yazılan “A Protocol Of Packet Network Intercomunication” isimli bir makale yayımlanmıştır. Bu makalede bir ağ üzerinde bulunan düğümler arasında veri alışverişi yapılabilmesi için “pocket-switching” yani paket anahtarlama yöntemini kullanan bir protokol yayımlanmıştır. İlk olarak “pocket-switching” yöntemi kullanılan ağ yapısı ‘ARPANET’dir.

“pocket-switching” yöntemi verileri anahtarlama (başlık, adres, hata denetimi, akış kontrolü vb. alanlar) ile kullanıcı verileri donanımdan donanıma farklılık göstermek üzere 500 ila 2000 byte boyutlarındaki paketlere ayrılır. Daha sonra paketler ağa gönderilir ve gönderilen paketler anahtarlanarak bir sonraki anahtarlama evresine verilir. Kontrol Algoritmaları ile de paketlerin doğru yolu bulması sağlanır. Paketler tek bir ağ yolu üzerinden iletilmez. Bu sayede veri kaybı önlenmiş/denetlenmiş olur.

TCP öncelikle local port üzerinden hedef porta bağlantı isteği gönderir ve eğer hedef port üzerinden olumlu cevap gelirse tekrar local port üzerinden bağlantı onaylandığına dair veri gönderilir ve veri iletişimi başlatılır. Veri iletişimi bittikten sonra hedef cihazda TCP başlıklarına göre veri birleştirilir ve bağlantı yine üç yönlü el sıkışma ile sonlandırılır. TCP de bazı stateler (durumlar) aşağıdaki gibidir.

LISTEN

(sunucu) İstemci tarafından bir TCP bağlantı isteğinin beklenildiği durum

SYN-SENT

(istemci) Karşı tarafa TCP bağlantısı isteği gönderildikten sonra karşı taraftan bağlantı isteğine cevap beklenilen durum

SYN-RECEIVED

(sunucu) İstemci tarafından SYN bayrağı ile yapılan bağlantı isteğine sunucunun SYN-ACK bayrağı ile cevap vermesinden sonraki bekleme durum

ESTABLISHED

(sunucu ve istemci) Bağlantı kurulduktan sonraki veri transferinin yapıldığı durum

FIN-WAIT-1

(sunucu ve istemci) karşı taraftan TCP sonlandırma isteği veya daha önce gönderilmiş olan bağlantı sonlandırma talebinin onaylanması bekleniyor.

FIN-WAIT-2

(sunucu ve istemci) karşı taraftan TCP bağlantısının bitirilme isteğinin beklendiği durum

CLOSE-WAIT

(sunucu ve istemci) Local kullanıcıdan bir bağlantı sonlandırma talebi bekleniyor.

CLOSING

(sunucu ve istemci) Karşı tarafa bağlantının bitirlmesine dair bir ACK bayrağı gönderildikten sonra bağlantının bitmesini bekleme durumu

LAST-ACK

(sunucu ve istemci) Daha önce uzak TCP'ye gönderilen bağlantı sonlandırma talebinin onayını bekliyor (bağlantı sonlandırma talebinin bir onayını içeren).

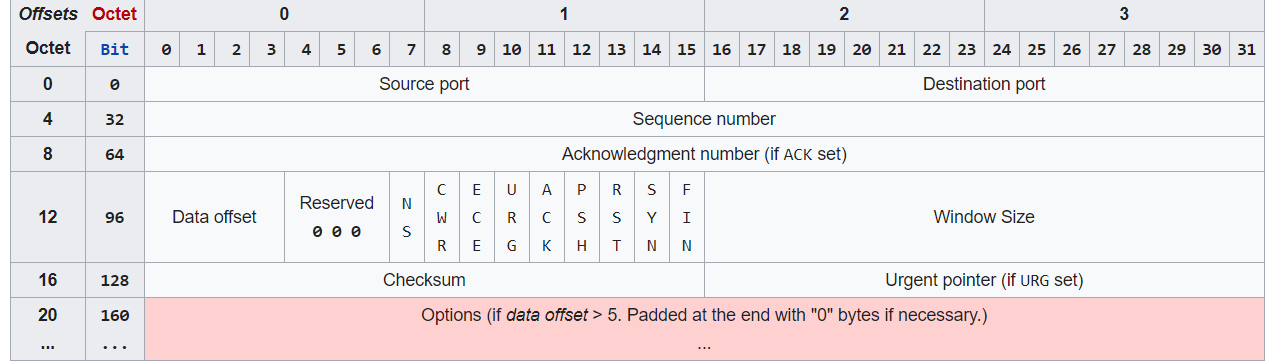
TIME-WAIT

(sunucu ve istemci) TCP'nin bağlantı sonlandırma talebinin onayını aldığından emin olmak için yeterince zaman geçmesi bekleniyor.

CLOSED

(sunucu ve istemci) TCP bağlantısının tamamen bittiği durum

TCP Header yapısı ve işlevleri aşağıdaki gibidir.



Kaynak Port (Source Port): Veriyi gönderen bilgisayarın kullandığı TCP portudur.

Hedef Port (Destination Port): Hedef bilgisayarın TCP portudur.

Sıra Numarası (Sequence Number): TCP'nin verinin böldüğü her bir segmentine verdiği numaradır.

Onay Numarası (ACK Number): Alınan bir SYN paketine karşılık olan onay mesajı ACK biti ile gönderilir.

Başlık Uzunluğu (Header Length/Data Offset): TCP başlığının uzunluğunu gösterir.

Rezerve Edilmiş (Reserved): İlerde kullanılmak üzere saklı tutulur.

Kod Bitleri ya da Bayraklar (Code Bits or Flags): Segment ile ilgili kontrol bilgilerini taşır.

Pencere (Window): Akış denetimi için kullanılır.

Hata Kontrol Bitleri (Checksum): Segmentin hatalı ulaşıp ulaşmadığını kontrol etmek için kullanılır.

Acil İşaretçisi (Urgent Pointer): Bir verinin acil olarak iletilmek istendiği durumlarda kullanılır.

Seçenek (Option): TCP segmentinin maksimum boyutunun bilgisini taşır.

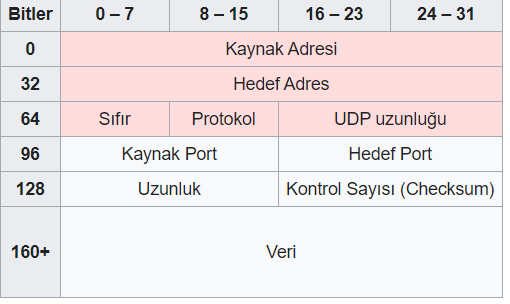
Veri (Data):Verinin bulunduğu kısım.

1. **UDP**

Bazı uygulamalar da (VoIP – Voice IP) verilerin daha az geçikme ile karşı tarafa

iletilmesi istenir. Bu yüzden TCP’de olduğu gibi kontrol yapıları gibi ekstra yapılar kullanılmadan UDP ile daha az geçikmeli olarak veriler iletilir. UDP Protokolünde verilerin iletim garantisi yoktur. Veriler ağ da kaybolabilir. (Verilerin ağ da trafik oluşturmaması için bir müddet sonra ağdan silinmesi gerekir.)

UDP Header yapısı ve işlevleri.



Bu header yapısının alanları ile TCP alanları aynı işlevleri görmektedir. Ayrıca kontrol sayısı test edilmesi için bazen udp paketinde sahte ipv4 kullanılır. Böylelikle checksum test edilir.

1. **PORT FORWARDİNG**

İnternete bağlanmak için kullandığımız routlerlarımızda güvenlik duvarı vardır. Bu

güvenlik duvarı dışardan gelecek zararlı verileri engellerken, bizim cihazımızdan çıkabilecek olan zararlı verilerin çıkmasını da engellemektedir. Aynı zamanda dışarıdan verilerin istediği gibi gelmesi engellenmiş olup sadece http – 80, ftp (file transfer protocol) – 21 gibi standart portlar açık bırakılmıştır. Kendi isteğimiz ile port yönlendirmek için “Port Forwarding” yani port yönlendirme yapmak zorundayız. Aynı zamanda windows işletim sistemlerinde “Gelişmiş Güvenlik Özellikli Windows Defender” üzerinden portun cihaza erişmesine izin verilmez. Buradan da gelen/giden kurallar ile uygulama için veya port numaraları için ağ portu kullanımına izin verilmesi gerekmektedir. Ayrıca sürekli olarak portlar TCP veya UDP bağlantı beklemez bağlantı isteği veya portlar “listening” durumda oldukları zaman aktif olmaktadırlar.

1. **Windows Command Prompt - Port Forwarding**

netsh interface portproxy add v4tov4 listenaddress=127.0.0.1 listenport=9000 connectaddress=192.168.0.10 connectport=80

Komut satırına yukarıdaki komut girilirse 127.0.0.1 adresi ve 9000 portu üzerinden

bizim 192.168.0.10 ip adresi ve 80 portumuza veri transfer işlemi başlatılmış olur Silmek/İptal etmek için ise;

netsh interface portproxy delete v4tov4 listenaddress=127.0.0.1 listenport=9000

Komutu kullanılır.

1. **Linux Terminal – Port Forwarding**

Linux da port yönlendirme yaparken SSH kulanıcaz. SSH güvenli olmayan ağda yani

denetimsiz ağda oluşturulan güvenli bir kanaldır. SSH üzerinde de güvenli kanallar yaratılabilir.

Masaüstündeki website adlı klasör içerisinde php websitemizi barındırdığımızı ve bu web sitesini local host da port yönlendirmeyle açalım.

Php -S ipaddress:80

Bu komutumuz ile local ağ da web sitemizi 80 portundan yayınlamış oluyoruz. Herhangi bir tarayıcı yardımı ile ipaddress:80 yazarak bu web sitesini görebiliriz.

Şimdi ise serveo hizmetini kullanarak SSH ile internetten erişime açacağız.

ssh -R 80:ipaddress:80 serveo.net

Ardından Forwarding http traffic from <https://xxxxx.serveo.net> yazısını göreceğiz ve web sitemize bu domain adı ile erişebileceğiz.

Eğer rastgele bir port numarası üzerinden web sitemize erişmek istersek

ssh -R 0:ipaddress:80 serveo.net

Komutu ile serveo.net:x şeklinde x random olmak üzere web sitemize erişim sağlayabiliriz.

serveo.net:our\_assigned\_port

Komutu ile web sitemizi tüm cihazlardan açılabilir hale getirdik.

Son olarak biz ücretsiz bir ssh hizmeti kullanıyoruz ve ücretsiz hizmetler sunucudaki aşırı yükleme sonucunda kaybolmaktadır bu yüzden

ssh -R 80:ipaddress:80 ssh.localhost.run

Komutu ile web sitemizdeki yenileyebiliriz.

1. **Router Interface – Port Forwarding**

Kullandığınız router markasına göre değişiklik gösteren varsayılan ip adresi ile

modem arayüzüne giriş yapabilir ve internet hizmetlerimizi kişiselleştirebiliriz. (wi-fi şifresi, wi-fi ssid vb.) aynı zamanda port forwarding işlemini de router üzerinden yapabiliriz. Fakat burda dikkat edilmesi gereken ip adresinizi CGNat ile almamanız gerektiği. Çünkü CGNat bir gerçek ip adresini birden fazla routera dağıtarak tüm kullanıcıların internete ip adresi yerine ip adresi:port numarası ile erişim sağlanmasıdır. Türkiye’de ki tüm mobil operatörler internet hizmetini CGNat ile sağlamaktadır. Bu sistemin uygulanma sebebi ise internet kullanıcıları arttıkça IPv4 adreslerinin yeterli gelmemesidir. Özetlemek gerekirse CGNat sisteminde siz internete zaten portlar ile çıkış yaptığınız için ip adresinize port yönlendirme yapamazsınız.

Gelelim modem arayüzünden port yönlendirme yapmaya. (Not: Windows işletim sistemi kullanıyorsanız yönlendireceğiniz porta Gelişmiş Güvenlik Özellikli Windows Defender üzerinden izin vermeniz gerekmektedir.) Ben netmaster marka modem kullandığım için 192.168.0.1 adresini tarayıcımın url kısmına yapıştırdıktan sonra modem arayüzüne girmek için bir kullanıcı adı şifre istiyor. İlk giriş için varsayılan şifre ve kullanıcı adı kullanım kılavuzunda yazılıdır. Şifreyi girdikten sonra bağlantı durumu cihaz durumu sayfası karşıma geliyor.



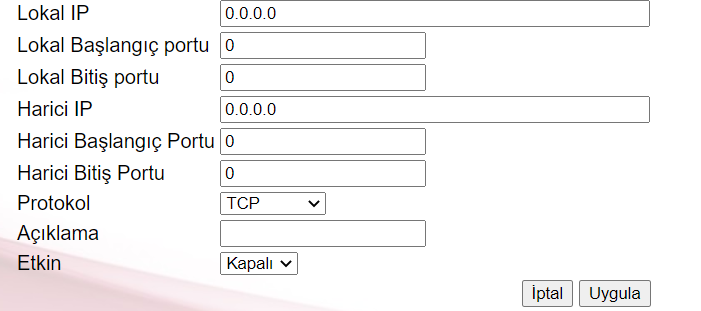
Bu sayfada gelişmiş kısmına bastıktan sonra



Karşımıza bu sayfa geliyor ve eğer DMZ Host ile cihazımızın local ip’sini belirtirsek tüm portlar üzerinden cihazımız açık oluyor. Tek veya birkaç port yönlendirmek için yönlendirme sekmesine tıklıyoruz.



Gelen ekranda daha önceden yönlendirme yaptığım local ip adresi ve portları görebiliyoruz ve gelen ekrandan IPv4 yarat butonuna tıklıyoruz.



Gerekli bilgileri doldurduktan sonra protokol seçerek TCP, UDP veya HER İKİSİ yönlendirmeyi tamamlıyoruz.

1. **SOCKET PROGRAMING**

İşletim sistemi kaynaklarına erişebilen C/C++ gibi işletim sistemi seviyesindeki diller

aracılığı ile programlanır. Çünkü programlar direkt olarak işletim sisteminin kaynaklarını yönlendirerek ağ kanalı oluşturulur.

1. C++

C++ ile socket programlama yapmak için bazı kütüphanelere ihtiyacımız vardır.

* Linux için

#include <sys/socket.h>

#include <arpa/inet.h>

#include <netinet/in.h>

#include <unistd.h>

* Windows için

#include <winsock2.h>

#include <ws2tcpip.h>

Kütüphaneleri kullanılır. İlk olarak socket için int geri dönüş tipli bir fonksiyon vardır. Parametre olarak int domain, int type ve int protocol alır.

Domain parametresi kullanılacak adresin tipini belirtmek için kullanılır.

* AF\_UNIX => Unix Dosyası
* AF\_INET => IPv4
* AF\_INET6 => IPv6

Sabitleri ile tanımlanır.

Type parametresi kullanılacak taşıma protokolünü belirtmek için kullanılır.

* SOCK\_STREAM => TCP Protokol
* SOCK\_DGRAM => UDP Protokol
* SOCK\_RAW => IP ve ICMP gibi protokollere arayüz sağlar. Sadece Ipv4 domaini ile çalışır

Sabitleri ile tanımlanır.

Protocol parametresi için varsayılan olarak TCP ve UDP için 0 değeri girilir. Getprotobyname(); ile belirtilen isimdeki protokolün int değeri de elde edilebilir. SOCK\_RAW için varsayılan bir protocol değeri yoktur. Protocol parametresi soket ile kullanılacak protokolü belirler.

Sunucu ve istemcide çalışacak komutlar farklıdır.

Sunucu oluşturmak için;

Sunucu bilgilerini tutmak için sockaddr\_in veya ipv6 için sockaddr\_in6 struct ı kullanılır. Bu struct’dan bir eleman türetildikten sonra string sınıfı içerisindeki memset ile structı temizleyip

* sin\_family => elemanına Domain parametresi
* sin\_addr nin s\_addr => elemanına htonl(parameter); fonksiyonuna parametre göndererek ip adresini girebilir veya INADDR\_ANY ile herhangi bir ip adresi için tanımlama yapılabilir.
* sin\_port => elemanına ip adresine bağlanacak/yönlendirilecek port girilirir.

Daha sonra geri dönüş tipi int olan ve başarılı ise pozitif, başarısız ise negatif bir değer döndüren ve bind(socket\_variable, socketaddr\_inVariable, sizeof(socketaddr\_in)) olmak üzere, ilk parametre socket fonksiyonu ile tanımlanan socket, diğeri belirtilen adres domain ve port numarası ve diğer parametresi de bu sınıfın boyutunu alan 3 parametresi vardır.

Şimdi ise gelen istek var mı kontrol etmek için istekleri dinlememiz gerekiyor. Bunu int geri dönüş tipli ve başarılı ise pozitif, başarısız ise negatif bir değer döndüren ve listen(socket\_variable, queue\_capacity); olmak üzere, yine socket fonksiyonundan türetilen socket değişkenimiz ve istekler de kaç kişinin kuyruğa aktarılıp diğerlerinin reddedileceğini belirttik.

Sırada gelen istekleri kabul etmek var. Gelen istekler parametre ve geri dönüş tipleri bind fonksiyonu ile aynı olan accept fonksiyonu ile yapılır.

Başarılı ile bağlantı kuruldu ve close ile accept ile kurduğumuz istemci’ye giden yol ile ana internete açılan yolu kapatabiliriz.

Ayrıca veri göndermek için bazı fonksiyonlar kullanılabilir send gibi.

İstemci oluşturmak için;

Bind kısmına kadar aynı yollar incelenir ve sonrasında connect ile veri tabanına bağlanma isteği gönderilir. Daha sonra kabul edilip mesaj alınmak isterse recv ile alınabilir.