

***nix' lerde Multicast Kullanımı**

/******

* Bâkır EMRE

* emre [at] enderunix [dot] org

* EnderUNIX Yazılım Geliştirme Takımı

* <http://www.enderunix.org>

*

* Sürüm : 1.1

* Tarih : 1.02.2007

* Makalenin en yeni versiyonu : <http://www.enderunix.org/docs/Multicast.pdf>

* adresinden elde edilebilir.

*****/

İçindekiler

Tanımlar.....	3
Multicast Adres Yapısı.....	4
TTL değeri.....	5
Freebsd de Multicast yönlendiriciyi etkinleştirmek.....	6
Linux ve FreeBSD için Bazı İpuçları.....	8
Multicast ve PF kullanımı.....	9
Multicast Programlama.....	10
Yapılacaklar	14
Kaynaklar.....	14

Tanımlar

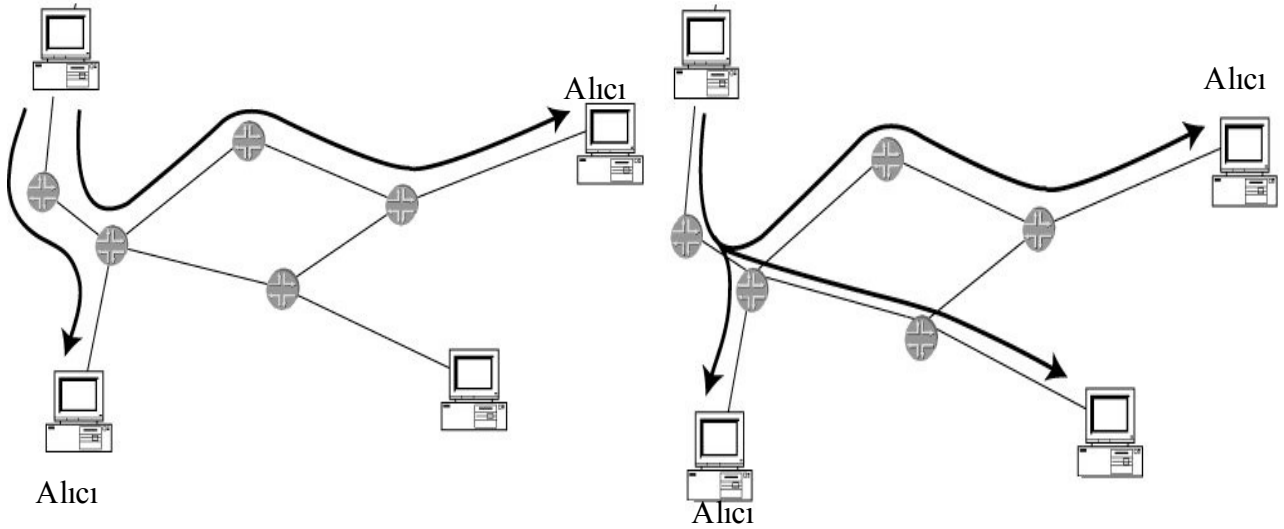
IPv4 de ağ içerisinde paketler bir uçtan başka bir uca veya uclara 3 farklı şekilde gönderilir. Bunlar broadcast,unicast ve multicast dir.

Broadcast paketleri ağ içerisindeki bütün uclara gönderilir.

unicast paketleri sadece belirlenen uc-uclara gönderilir.

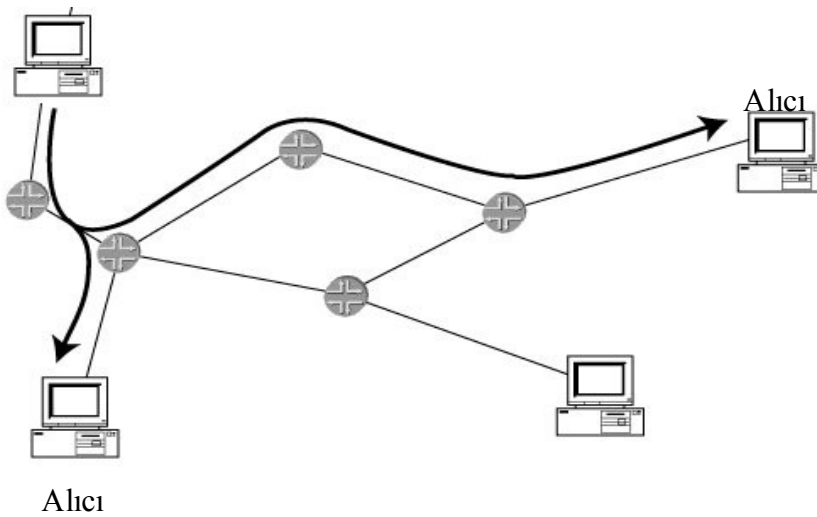
Multicast ise bir uçtan , üyelik sistem ile çalışan bir gruptaki alıcılara gönderilmesi sağlayan gönderim şeklidir

Multicast' i diğer yöntemleri şekilsel olarak karşılaştırmak gerekirse:



Unicast

Broadcast



Multicast

Görüldüğü gibi unicast te erişilecek herbir host için bir bağlantı oluşturulur ve herbiri için ayrı ayrı paket gönderilir. Her bir bağlantı için yük getirmektedir.

Broadcast te ise çıkan paket ağdaki bütün uc noktalara gönderilir dedik. Burada yayın yapan için ek yük getirmese bile ağ içerisinde bu trafik ile alakalı olmayan hostlar için sıkıntıdır.

Unicast, broadcast ve multicast iletim şekilleri karşılaştırıldığında, multicast aşağıdaki avantajları sağlamaktadır:

- Çok daha az bant genişliği kullanır,
- Çok daha az işletim gücü kullanır,
- Kullanıcı adedi artsa bile tıkanıklığa neden olmaz,

multicast ile bilgisayarın trafiği özel bir IP ye doğrudur. Diğer bilgisayarlar ise bu trafiği alabilmek için bu IP ye kaydolurlar. Bu şekilde sunucu veri akışını bir IP için yapar. Bir çok istemci ise bu IP ye kaydolduğu takdirde veriyi alabilir. Böylece aynı veriyi ağda birden fazla göndermenin önüne geçilmiş olur. Ayrıca ağın tamamı bu trafikle meşgul olmasının önüne geçilmiş olur.

Bu faydalarına karşın Multicast her zaman elverişli olmayabilir. işletim sisteminiz desteklemeyebilir. veyahut uygun ağ araçlarına (router,switch vb.) sahip olmayabilirsiniz.

Eğer Multicast yayının internet üzerinden almak yada göndermek isterseniz ISP nizin MBONE (Multicast Backbone) bağlı olması gerekmektedir. İnternet üzerinden sınırlı sayıda kullanıcı multicast yayını alabilmektedir. kı bunların çoğu üniversite veya araştırma merkezleridir..

İnternet tarafı böyle olmasına karşın yerel alan ağları uygun anahtarlama ve yönlendirme cihazları elde edildiği müddetçe Multicasti kolaylıkla destekler....

Multicast Adres Yapısı

Multicast adres yapısı ise IP adreslemeden kullanılan A,B,C,D,E sınıflarından D sınıfına girer bildiğimiz gibi IP adresleme sınıfları şu şekildedir.

Sınıf	Ensoldaki Bitleri	Başlangıç Adr.	Bitiş Adresi
A	0xxx	0.0.0.0	127.255.255.255
B	10xx	128.0.0.0	191.255.255.255
C	110x	192.0.0.0	223.255.255.255
D	1110	224.0.0.0	239.255.255.255
E	1111	240.0.0.0	255.255.255.255

Genel olarak multicast adresleri 224.0.0.0 ile 239.255.255.255 aralığındaki bir adres kullanılabilir. CIDR 224.0.0.0/4 olarak ifade edilir.

224.0.0.0 ile 224.0.0.255 arasındaki adres aralığı sadece yerel alan ağlarında multicast adresi olarak kullanılabilir ve bu şekilde atanmıştır.

Multicast Backbone (MBONE) 224.2.0.0 adresini multimedia (video – audio) konferansları için kullanır

Ayrıca multicast adres dağılımında tanımlı olan bir kaç adres aşağıdaki gibidir.

224.0.0.0	Temel adres (saklı)
224.0.0.1	Aynı ağ segmentindeki sistemleri kapsayan multicast grubunun bütün üyeleri
224.0.0.2	Aynı ağ segmentindeki yönlendiricileri kapsayan multicast grubundaki bütün yönlendiriciler
224.0.0.5	Open Shortest Path First (OSPF) kullanan yönlendiricilerin adresleri. OSPF yöndendirme bilgilerini network segmentindeki OSPF kullanan bütün yönlendiricilere göndermek için kullanılır
224.0.0.6	The OSPF Designated Routers adresi. OSPF yöndendirme bilgilerini network segmentindeki OSPF kullanan designated yönlendiricilere göndermek için kullanılır
224.0.0.9	RIP (Routing information protocol) v2 grup adresi. , RIP yöndendirme bilgilerini network segmentindeki RIP v2 kullanan bütün yönlendiricilere göndermek için kullanılır
224.0.0.10	EIGRP grup adresi. EIGRP kullanan yönlendiricilerin adresleri. EIGRP yöndendirme bilgilerini network segmentindeki EIGRP kullanan bütün yönlendiricilere göndermek için kullanılır
224.0.0.18	Virtual Router Redundancy Protocol
224.0.0.102	Hot Standby Router Protocol Version 2
224.0.1.41	H.323 Gatekeeper discovery address

TTL değeri

IP paketleri için TTL değerinin kullanılmasının ilk amacı, ağ içerisinde paketin yaşam süresini sınırlayan ve sirkülasyona girmesini engellemek için kullanılan değeri oluşturmaktır.

İkinci amacı ise yönetim sınırları ağ tarafında belirlenen alanları tanımlamak için paketlerin içerisinde yer alır Örneğin “..Site” “..Şirket” ... gibi. Bu alan multicast paketinin kapsadığı alanı “scope” unu oluşturur.

aşağıda TTL değerlerine karşı Kapsadığı alanları belirtilen tablo bulunmaktadır.

TTL	Kapsadığı alan
0	Aynı host ile sınırlı. Hiçbir arayüzden dışarı çıkamaz
1	Aynı altağ ile sınırlı. Yönlendiriciler arasından geçemez.
16	Aynı site, organizasyon veya bölüm ile sınırlı.
64	Aynı alan ile sınırlı.
128	Aynı kıta ile sınırlı.
192	Dünya geneli, sınırlı bantgenişliği için kullanılır.
255	Küresel herhangi bir kısta ugramamış.

Freebsd de Multicast yönlendiriciyi etkinleştirmek

Freebsd hostlar için varsayılan olarak multicasti desteklemektedir. Eğer bilgisayarınızı multicast router olarak kullanmak isterseniz işletim sisteminde multicasti etkinleştirmek için kernel e şu seçenekleri eklemeniz gerekmektedir. İzlememiz gereken adımlar şu şekildedir.

Öncelikle işletim sistemi çekirdeğine **MROUTING** ve **PIM** seçeneklerini ekleyelim

```
cd /usr/src/sys/i386/  
  
vi KernelYapilandirmaDosyaniz
```

daha sonra

```
options      MROUTING  
options      PIM
```

kaydedip çıktıktan sonra sırasıyla

```
config KernelYapilandirmaDosyaniz  
  
cd ../compile/KernelYapilandirmaDosyaniz  
make depend  
make  
make install
```

sistem yeniden başlatılır.

Multicast routing etkin olabilmesi için `/etc/rc.conf` dosyası içerisine şu satırlar eklenmeli

```
mrouting_enable="YES"
```

Servisi yeniden çalıştırmak için ise

```
/etc/rc.d/mrouted start
```

komutu kullanılmalıdır.

mrouted IP multicast routing daemon olarak bilinen aşadı kullanılan parametrelerin açıklamaları verilmiştir.

- c *config_file***
yapılandırma dosyasını belirtir (varsayılan olarak **/etc/mrouted.conf**).
- d [*debug_level*]**
debug seviyesini (varsayılan değeri 0).
- p**
mrouted ı non-pruning modda çalıştırır.

Multicast routing yapılandırma dosyası **/etc** klasoru altında koyabilirsiniz.

mrouted.conf örnek dosyasını **/etc** altına kopyalayın. Ve programı

```
mrouted -c /etc/mrouted.conf
```

komutuyla çalıştırabilirsiniz. Program bilgisayara bağlı tüm arayüzlere igmp mesajı gönderecektir.

mrouted programını -d parametresi ve devamında debug seviyesini kullanarak, komşu yönlendiricileri ve gönderilen igmp mesajlarını izleyebilirsiniz.

Örnek **mrouted.conf** ise aşağıdaki gibidir.

```
phyint xl0 allow_nonpruners threshold 1 metric 1  
phyint xl1 allow_nonpruners threshold 1 metric 1  
phyint fxp0 allow_nonpruners threshold 1 metric 1
```

phyint :

Tek başına kullanılmaz arayüz ile kullanılır. Arayüzle alakalı komutlar ile devam eder. Arayüz adresi veya ismi ile beraber kullanılır.

threshold :

IP yaşam süresinin (ttl) en az değerini belirtir ki bu da verilen arayüzden multicast datagramının gönderilmesi için gereklidir.

metric :

metric ise verilen arayüzden alınan datagramlar için bir maliyet olarak tanımlanır. Yönlendirme seçimlerinde etkilidir.

diğer seçenekler için :

```
man mrouted
```

Linux ve FreeBSD için Bazı İpuçları

- Linux altında hostun hangi multicast grubuna dahil olduğunu görmek isterseniz proc dosya sistemini kullanabiliriz.

```
cat /proc/net/igmp
```

ile ile hangi gruba dahil olduğunu görebiliriz

- tcpdump kullanarak multicast paketlerini dinlemek istersek

```
tcpdump igmp
```

dememiz yeterli olacaktır.

ayrıca bunu

```
tcpdump ip[9:1]=1
```

şeklinde de yapabiliriz. burada ki kullanımda `ip[9:1]` IP başlığındaki Protokol alanını verir 2 igmp nin protokol numarasıdır dolayısıyla 9:1 şeklindeki kullanım ilede igmp paketlerini dinleyebilirsiniz.

- `netstat -g` komutunu kullanarak multicast grup üyeliklerini de görebiliriz (Linux için aşağıdaki çıktıyı alırsınız)

```
netstat -g
```

```
IPv6/IPv4 Group Memberships
Interface          RefCnt Group
-----
lo                  1      ALL-SYSTEMS.MCAST.NET
eth0                2      224.0.0.251
eth0                1      ALL-SYSTEMS.MCAST.NET
lo                  1      ip6-allnodes
eth0                1      ff02::1:ffa9:638a%3218689112
eth0                1      ip6-allnodes
```

Freebsd için aşağıdaki şekilde bir çıktı vermektedir.

```
Virtual Interface Table
Vif  Thresh  Rate  Local-Address  Remote-Address  Pkts-In
Pkts-Out
  0      1      0    192.168.9.125  0              0

Multicast Routing Table is empty

IPv6 Multicast Interface Table is empty

IPv6 Multicast Routing Table is empty
```


IPv4/IPv6 Multicast Group Memberships		
Group	Gateway	Netif
224.5.5.5	01:00:5e:05:05:05	x10

- Linux'te multicast grubunun max eleman sayısı görmek için

```
cat /proc/sys/net/ipv4/igmp_max_memberships
```

- multicast grubunun max eleman sayısı değiştirmek için

```
sysctl de /etc/sysctl.conf dosyasına
```

```
net.ipv4.igmp_max_memberships = yeni_deger
```

yazarak değiştirmelisiniz ve sistemi yeniden başlattığımızda bu yeni değer sayısınınca grup elemanı multicast grubunun sayısı en fazla olacaktır

- FreeBSD de multicast uygulamaları

```
/usr/ports/mbone
```

altında bulunmaktadır

Multicast ve PF kullanımı

Güvenlik duvarı olarak Openbsd Packet filter kullanıyorsanız varsayılan olarak IGMP ve Multicast paketlerini bloklayacağını unutmayın. Çünkü PF varsayılan olarak IP options ayarlanmış paketleri durdurmaktadır. Bunu engellemek için **allow-opts** parametresini kuralların akabinde kullanmak gerekir,

Örnek kullanımı

```
pass quick on $dmz_if inet from any to 224.0.0.0/4 allow-opts keep state
```

```
pass out on $dmz_if proto igmp from any to any allow-opts
```

gibi...

Multicast Programlama

Multicast programlama için socket arayüzünü kullanılır. Multicast ile alakalı yapıları ve tanımlamaları

linux için `/usr/include/linux/in.h`

dosyasında

freebsd için ise `/usr/include/netinet/in.h`

içerisinde bulabilirsiniz.

multicast ile alakalı işlemlerimizi aşağıdaki iki fonksiyon ile gerçekleştirirsiniz.

```
int getsockopt(int s, int level, int optname, void* optval, int* optlen);
```

```
int setsockopt(int s, int level, int optname, const void* optval, int optlen);
```

bu iki fonksiyona gerekli parametreleri göndererek işlemlerimizi yapabiliriz.

ilk parametre `s` sistem çağrısının uygulanacağı socket. Multicasting için bu socket `AF_INET` ailesinden olması gerekmektedir Ayrıca socket tipi `SOCK_DGRAM` veya `SOCK_RAW` olmalıdır.

ikinci parametre `level` katmanı belirtir.

`SOL_SOCKET` socket katmanı için,
`IPPROTO_IP` ise IP katmanı için kullanılır.

`optname` ise ayarlanacak yada alınacak seçenekleri belirtir. aşağıda bu seçenekler gözükmektedir.

```
IP_MULTICAST_IF          /* u_char; set/get IP multicast arayüzü */
IP_MULTICAST_TTL        /* u_char; set/get IP multicast time-to-
live süresi*/
IP_MULTICAST_LOOP       /* u_char; set/get IP multicast loopback */
IP_ADD_MEMBERSHIP       /* ip_mreq; grup üyeliği eklemek */
IP_DROP_MEMBERSHIP      /* ip_mreq; grup üyeliğini çıkarmak */

struct ip_mreq {
    struct in_addr imr_multiaddr; /* IP multicast grup adresi */
    struct in_addr imr_interface; /* arayüz adresi*/
};
```

`ip_mreq` yapısı kullanılarak `IP_ADD_MEMBERSHIP` ve `IP_DROP_MEMBERSHIP` gerçekleştirilir.

son olarak `optlen` ise `optval` işaret ettiği struct' un uzunluğunu gösterir.

örnek bir multicast için oluşturulmuş sunucu istemci yapıları gösterilmektedir. Kısaca yaptıkları işe gelince sunucu multicast grup adresi olarak "225.1.1.1" yayın yapmaktadır. Gruba dahil olan istemcilere mesaj gönderen bir uygulamadır

istemci kodu aşağıdaki gibidir.

```
#define PORT_NUM          1234
#define GROUP_ADDR "225.1.1.1"

void main (void)
{
    unsigned int multi_server_sock;          // Multicast socket tanımlayıcısı
    struct ip_mreq mreq;                    // Multicast grup yapısı
    struct sockaddr_in client_addr;         // istemci adresi
    unsigned int addr_len;                  // adres uzunluğu
    unsigned char buffer[256];
    int retcode;

    /*
    socketi oluşturup ilgili alanlar ile dolduralım
    Burada multicast serverin adresi 225.1.1.1 olarak tanımlandı      */

    multi_server_sock = socket (AF_INET, SOCK_DGRAM, 0);
    mreq.imr_multiaddr.s_addr = inet_addr (GROUP_ADDR);
    mreq.imr_interface.s_addr = INADDR_ANY;

    //istemci adresin bilgileri multicast server ile bağlanıyor.

    client_addr.sin_family = AF_INET;
    client_addr.sin_addr.s_addr = INADDR_ANY;
    client_addr.sin_port = htons (PORT_NUM);
    retcode = bind (multi_server_sock, (struct sockaddr *) &client_addr,
                    sizeof (struct sockaddr));
    if (retcode < 0)
    {
        printf ("bind() hata kodu = %d \n", retcode);
        return;
    }

    /* socketi multicast gruba dahil
    * int setsockopt(int s, int level, int optname, const void* optval, int\
    *                  optlen)

    * yapısına uygun bir şekilde dolduruluyor ve dönüş koduna atanıyor. */

    retcode = setsockopt (multi_server_sock, IPPROTO_IP,
                          IP_ADD_MEMBERSHIP, (char *) &mreq, sizeof (mreq));
    if (retcode < 0)
    {
        printf ("setsockopt() hata kodu = %d \n", retcode);
        return;
    }
}
```

```

    }

    addr_len = sizeof (client_addr);

    while (1)
    {
        // multicast server datagram ları al
        if ((retcode = recvfrom (multi_server_sock, buffer, sizeof
(buffer), 0,
                                (struct sockaddr *) &client_addr,
                                &addr_len)) < 0)
        {
            printf (" recvfrom() hatası\n");
            exit (1);
        }

        //Gelen datayı ekrana yazdır
        printf ("%s\n", buffer);
    }

    //soket tanımlayıcısını kapat
    close (multi_server_sock);
}

```

Sunucu kodu da aşağıdaki gibidir.

```

void main(void)
{
    unsigned int      server_s;                // Sunucu soket tanımlayıcısı
    unsigned int      multi_server_sock;      // Multicast soket
                                                tanımlayıcısı
    struct sockaddr_in  addr_dest;            // Multicast grup adresi
    struct ip_mreq      mreq;                // Multicast grup
                                                tanımlayıcısı
    unsigned char      TTL;                  // multicast paketlerin
                                                TTL değeri
    struct in_addr      recv_ip_addr;        // IP address
    unsigned int        addr_len;
    unsigned char      buffer[256];
    int                 count;
    int                 retcode;

    // multicast soketi oluştur.
    multi_server_sock=socket(AF_INET, SOCK_DGRAM,0);

    // multicast grup adres bilgisi
    addr_dest.sin_family = AF_INET;
    addr_dest.sin_addr.s_addr = inet_addr(GROUP_ADDR);
}

```

```
addr_dest.sin_port = htons(PORT_NUM);

// TTL deęerini ayarla
TTL = 1;
retcode = setsockopt(multi_server_sock, IPPROTO_IP, IP_MULTICAST_TTL,
                    (char *)&TTL, sizeof(TTL));
if (retcode < 0)
{
    printf("setsockopt() hata kodu = %d \n", retcode);
    return;
}

addr_len = sizeof(addr_dest);

count = 0;
printf("*** multicast datagramı'%s' (port = %d) gönderiliyor\n",
        GROUP_ADDR, PORT_NUM);
while(1)
{
    count++;

    // Mesajı buffer a yaz
    sprintf(buffer, "Multicast Grup - mesaj sayısı %d", count);

    // buffer datagram multicast gruba gönder
    sendto(multi_server_sock, buffer, sizeof(buffer), 0,
           (struct sockaddr*)&addr_dest, addr_len);
    sleep(1);
}

close(multi_server_sock);
}
```

Yapılacaklar

- IGMP ve Multicast trafiklerinin analizleri
- FreeBSD mbone portlarının incelenmesi
- Multicast trafiği oluşturma
- Dilbilgisi ve yazım hatalarını düzeltilmesi (:

Kaynaklar

- <http://www.freebsd.org/ports/mbone.html>
- <http://www.ulakbim.gov.tr/ulaknet/servisler/multicast/unix.uhtml>
- <http://www.linuxfocus.org/English/January2001/article144.shtml>
- `man mroute`