**Cuprins**

[**1)Primirea pachetului in router**](#_si739gam4weq) **3**

[**2) Calcularea Checksum-ului**](#_g7cmg4u1be64) **4**

[**3) Cautare in tabela de rutare**](#_axo3zx1dxf84) **4**

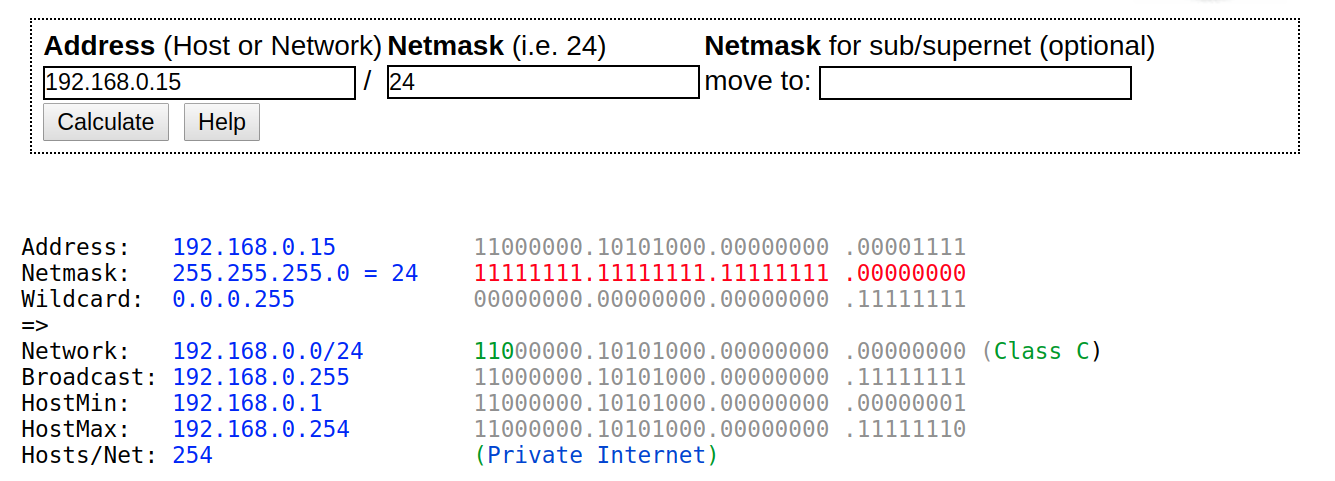
[**4) Modificare ttl**](#_prapk81f0sci) **5**

[**5) Modificare header Ethernet**](#_48awgzif6f71) **6**

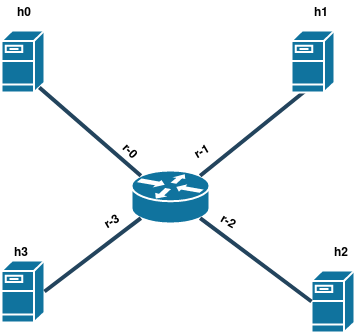
[**6) Trimiterea pachetului la next hop**](#_ekjv89qpn3f8) **7**

**Adrese IP, mask**

* In exemplul de mai jos am o adresa IP de statie 192.168.0.15 si o masca de retea /24 care se traduce in 255.255.255.0.
* Puteti observa in poza ca toate adresele de acolo au un corespondent in binar.
* Deci, o adresa IPv4 este formata din 4 Bytes, deci este un int. Notatia cu “.”, de exemplu “192.168.0.15” este ceva human readable. In memorie datele sunt stocate ca int-uri. De exemplu, adresa 192.168.0.15 va fi reprezentata in memorie ca 3232235535
* Poza de mai jos va spune ca 192.168.0.15 /24 apartine retelei 192.168.0.0/24, retea din care se pot asigna IP-uri pentru 254 calculatoare.



**Cum arata topologia pe care o folosim?**



Ce face un router in cazul nostru?

1. Primeste pachet de la o statie (get\_packet)
2. Calculeaza Checksum
3. Cauta cel mai bun entry din tabela de rutare in functie de care trimite pachetul mai departe
4. Updateaza campul TTL
5. Modifica headerul Ethernet
6. Trimite pachetul la next hop (send\_packet)

## 1)Primirea pachetului in router

**Pachet Ethernet si IP.**

In laborator aveti un header de Ethernet si un header de IP, ambele puse in payload-ul din msg (cand apelati get\_packet se populeaza structura de tip msg)

|  |
| --- |
| msg m*;* get\_packet(&m)*;* |

Cum arata structura msg pe care o primiti in get\_packet ?

|  |
| --- |
| struct msg {  Int len;  char payload[];  int interface; } |

In msg.payload voi aveti headerul de Ethernet urmat de headerul de IP:

|  |  |
| --- | --- |
| Ethernet Header | IP Header |

Cum arata headerul de Ethernet?

|  |
| --- |
| struct ether\_header {  u\_char ether\_dhost[6]*; // destination MAC (\*\*TREBUIE FOLOSIT\*\*)*  u\_char ether\_shost[6]*; // source MAC (\*\*TREBUIE FOLOSIT\*\*)*  u\_short ether\_type*;* }*;* |

Cum arata headerul de IP?

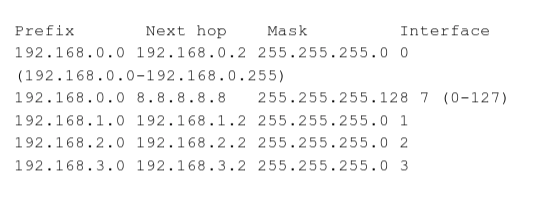
|  |
| --- |
| struct iphdr { #if defined(\_\_LITTLE\_ENDIAN\_BITFIELD)  \_\_u8 ihl:4,  version:4; #elif defined (\_\_BIG\_ENDIAN\_BITFIELD)  \_\_u8 version:4,  ihl:4; #else #error "Please fix <asm/byteorder.h>" #endif  \_\_u8 tos;  \_\_u16 tot\_len;  \_\_u16 id;  \_\_u16 frag\_off;  \_\_u8 ttl; *//Time To Live (\*\*TREBUIE FOLOSIT\*\*)*  \_\_u8 protocol;  \_\_u16 check; *//checksum (\*\*TREBUIE FOLOSIT\*\*)*  \_\_u32 saddr; *// source IP address*  \_\_u32 daddr; *//destination IP address (\*\*TREBUIE FOLOSIT\*\*)*   }; |

## 2) Calcularea Checksum-ului

|  |
| --- |
| uint16\_t checksumFromPacket = ipHeader->check ipHeader->check = 0 checksumComputerByRouter = ip\_checksum(ipHeader, sizeof(struct ipHeader))  If checksumFromPacket == checksumComputerByRouter  Packet is ok Else   Corrupt |

## 3) Cautare in tabela de rutare

**Routerul** este un echipament de retea care se ocupa cu rutarea pachetelor. Atunci cand un pachet ajunge in router, acesta extrage adresa destinatie din pachetul IP si in functie de tabela de rutare, trimite pachetul catre un next-hop.

Exemplu de tabela de rutare.

Next hop = adresa catre care se trimite pachetul

Mask = subnet mask. Identifica o retea

Interface = interfata pe care se trimite pachetul

Prefix = o adresa de retea

**Cum se citeste un entry din tabela de routare?** Urmatorul exemplu este pentru prima intrare din tabela:

* Ca sa ajung in reteaua 192.168.0.0/24(prefix+mask) trebuie sa trimit pachetul catre adresa 192.168.0.2(next-hop) pe interfata 0

**Cum se face match pe un entry din tabela de routare?**



|  |
| --- |
| *//daddr field in linux* dstIP = extract dstIP from IP header maxMask = -1 bestMatchEntry = null  for entry in routingTable  If dstIP & entry.mask == prefix  *//get the entry with the largest mask(the most specific)*  If entry.max > maxMask  maxMask = entry.mask   bestMatchEntry = entry *// you can now access bestMatch as the entry on which the routing is //based* |

## 4) Modificare ttl

TTL-ul se decrementeaza

|  |
| --- |
| ipHeader->ttl -= 1; |

## 5) Modificare header Ethernet

* De exeplu se da un ping din h0 in h1. Flow pachet: ho -> router -> h1 si invers
* Din moment ce h0 si h1 sunt in alta retea MAC(h1) nu are sens in reteaua lui h0 deci se foloseste de MAC-ul routerului in headerul ethernet

**Cum arata pachetul cand se trimite de la h0 -> router**

Primele 2 coloane apartin de headerul de Ethernet, ultimele 2 de headerul IP. OBS! Sunt si alte campuri in headere, dar cele de mai jos sunt cele mai importante

|  |  |  |  |
| --- | --- | --- | --- |
| Mac Destinatie | Mac Sursa | Ip Sursa | Ip Destinatie |
| Adresa MAC a interfetei routerului pe care se primeste pachetul | Adresa MAC a lui h0 | IP h0 | IP h1 |

**Cum trebuie construit pachetul in router pentru a putea fi trimis care h1?**

|  |  |  |  |
| --- | --- | --- | --- |
| Mac Destinatie | Mac Sursa | Ip Sursa | Ip Destinatie |
| Adresa MAC a lui h1 aflata din tabela ARP  **(\*)** | Adresa MAC a interfetei de iesire a routerului pe care se trimite pachetul **(\*\*)** | IP h0 | IP h1 |

**(\*)**

Tabela ARP mapeaza adresa IP la adresa MAC. Trebuie iterat prin aceasta tabela si identificata adresa MAC care corespunde adresei IP a next hop-ului (in cazul nostru h1)

Daca adresa next hop e 192.168.3.2 se intoarce de:ad:be:ef:00:03 si se completeaza in MacDestinatie

|  |
| --- |
| IP MAC  192.168.0.2 de:ad:be:ef:00:00 192.168.1.2 de:ad:be:ef:00:01 192.168.2.2 de:ad:be:ef:00:02 192.168.3.2 de:ad:be:ef:00:03 |

**(\*\*)**

Exsta get\_interface\_mac, care populeaza parametrul al doilea cu adresa MAC asociata interfetei routerului pe care se trimite pachetul(primul parametru) (din tabela de routare).

|  |
| --- |
| get\_interface\_mac(route->interface, &eth\_hdr->ether\_shost) |

## 6) Trimiterea pachetului la next hop

Trebuie folosita functia send\_packet care trimite pachetul pe o interfata specificata de voi. Interfata trebuie sa fie cea pe care routerul o va folosi pentru a trimite pachetul catre next hop. Interfata a fost aleasa din tabela de rutare (entry->interface).