

# RCOM Network

up201606673-André Esteves

18 January 2019

## 1 Network

Network label - camada responsável pela transferência de pacotes

### 1.1 Overview

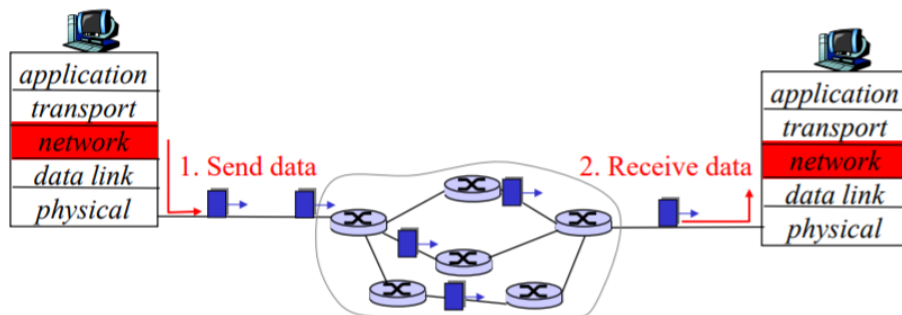
- Camada de Network (Network layer)
  - Transporta os pacotes(datagrams)
  - "from sending host to receiving host"
  - funções localizadas em todos os hosts e routers
- Transmissor(Sender):
  - Encapsula a informação em pacotes
  - Cria os pacotes
- Receptor(Receiver):
  - Recebe os pacotes
  - Envia a informação para o transport layer
- Router:
  - Recebe os pacotes pela linha de input
  - Examina o cabeçalho dos pacotes
  - Reencaminha os pacotes para o sítio certo
  - Tem de saber o caminho mais curto para determinar o caminho

## 1.2 Funções principais da camada de rede

- Forwarding
  - router trata de enviar o pacote desde a porta de entrada(input) até à porta de saída(output)
- Routing
  - determina a rota definida pelos packets
  - algoritmos, caminho mais curto

## 1.3 Rede de datagramas

- Serviço não orientado à ligação
- Não há o conceito de circuito
- Os pacotes são redirecionados de acordo com a fonte e o destino
- Pacotes com o mesmo par fonte-destino podem seguir caminhos diferentes



9

<u>Destination Address Range</u>	<u>Output Link Interface</u>
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

$2^{32}$  possible entries in IPv4

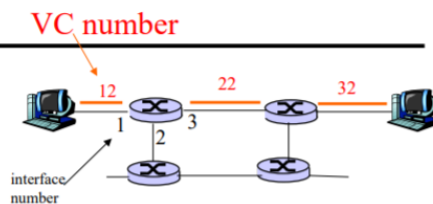
## 1.4 Circuitos Virtuais

- Serviço orientado à ligação
- Fases:
  1. Estabelecer o circuito
  2. Transferência de dados
  3. Terminação do circuito
- Cada pacote carrega um identificador do circuito virtual
- Caminho da fonte ao destino -  $i$  sequência de identificadores virtuais, um para cada ligação
- Estado de cada circuito mantido pelo router, que pode alocar recursos (bandwidth, buffers) por circuito virtual

### 1.4.1 Forwarding Table

Contém prefixos e a respetiva porta de saída [Endereço/Mask, port]

#### *VC - Forwarding Table*



Forwarding table in  
northwest router:

Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

**Routers maintain connection state information!**

#### 1.4.2 Ex: Maior correspondência de prefixo

<u>Prefix Match</u>	<u>Link Interface</u>
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
otherwise	3

#### Examples. Which Interface?

DA: 11001000 00010111 00010110 10100001 → 0

DA: 11001000 00010111 00011000 10101010 → 1,2 → 1

  
longest prefix

#### 1.5 Circuitos Virtuais versus Rede de Datagramas

Issue	Datagram subnet	Virtual-circuit subnet
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

#### 1.6 Arquitetura do router

- Funções principais:
  - Correr algoritmos de roteamento e protocolos (RIP, OSPF, BGP)
  - Reencaminhar pacotes

- Componentes principais:

- Input Port

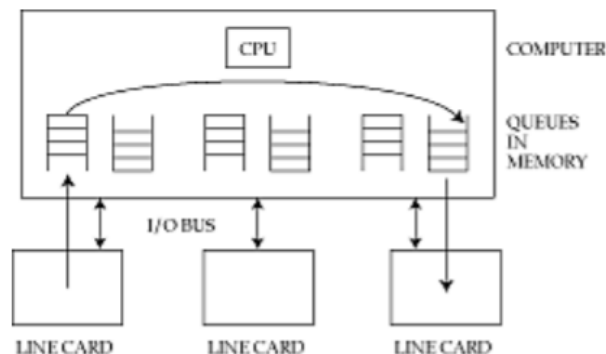
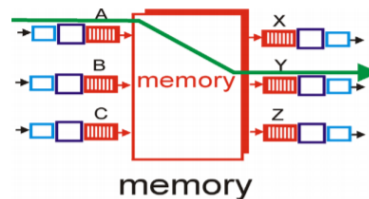
- \* Physical Layer (bit-level)
- \* Data Link Layer (e.g., Ethernet)
- \* Queuing (se os pacotes chegarem rápido demais)
- \* Lookup + Forwarding (faz algum reencaminhamento imediatamente)

- Output Port

- \* Buffering (quando é excedida a velocidade de saída)
- \* Queuing (com disciplina de agendamento)(Queuing perda e espera - devido ao overflow do buffer da porta de input)
- \* Data Link Layer (protocol, desencapsulação)
- \* Physical Layer (linha de terminação)

- Switching Fabric

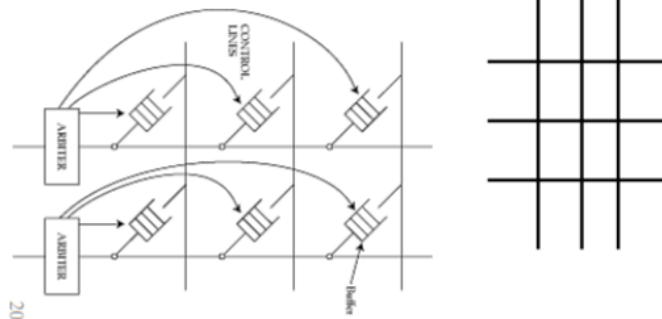
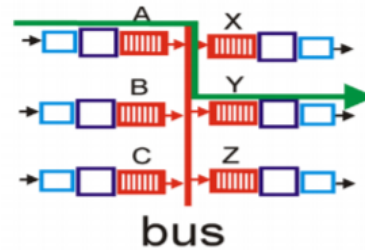
- \* Controla o reencaminhamento (fisicamente ou através dum CPU)
- \* Switching Via Memória do Computador
  - Router de primeira geração
  - Em computadores tradicionais, switching é controlado pelo CPU
  - Cada pacote é copiado para a memória do sistema e transferida duas vezes pelo bus



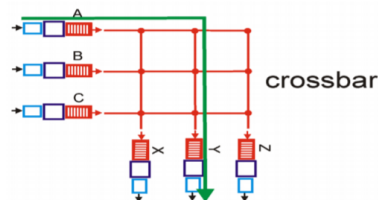
- \* Switching via a Bus

- Os pacotes são processados por um bus partilhado

- A transferência dos pacotes desde a linha de input e output é realizada de forma direta
- A taxa da conexão do bus é limitada pela bus bandwidth



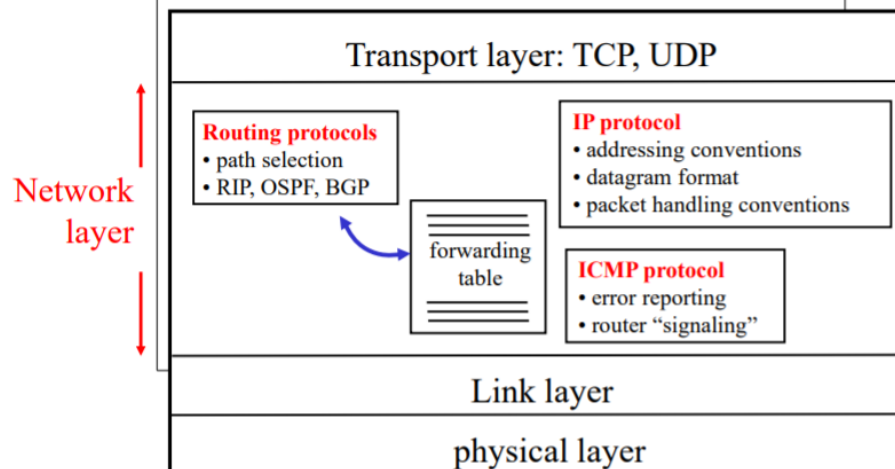
- \* Switching via a Crossbar
  - $2N$  buses
  - Possibilita transferências simultaneas de pacotes
  - a cross bar pode conter buffers internos
  - Ultrapassa os limites da bus bandwidth



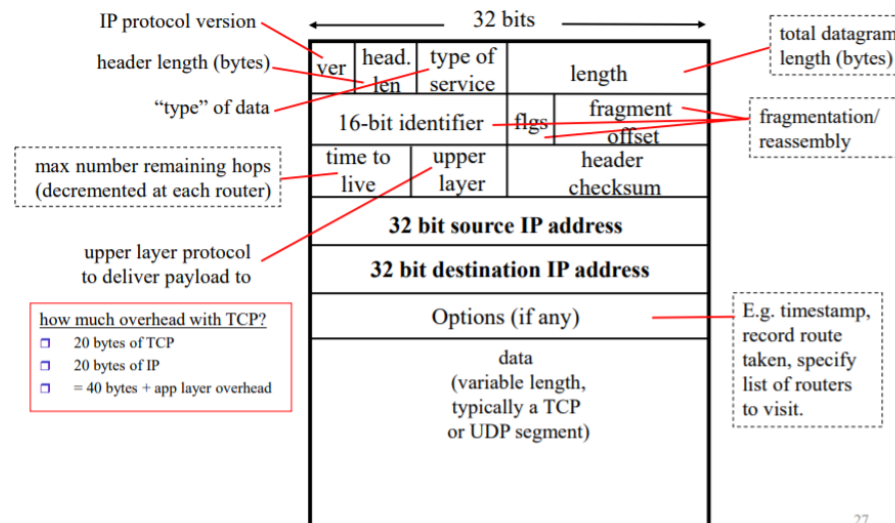
## 1.7 Protocolo Internet

1. Camada de rede Internet

## Host, router network layer functions



## 2. Formato datagrama IP



## 3. Internet Checksum

## Internet Checksum

- ♦ The Internet (**not layer 2**) uses a checksum
  - » easily implementable in software
  - » 1's complement sum of 16 bit words
  - » Performance: d=2

```
u_short
cksum(u_short *buf, int count)
{
    register u_long sum = 0;

    while (count--)
    {
        sum += *buf++;
        if (sum & 0xFFFF0000)
        {
            /* carry occurred,
             so wrap around */
            sum &= 0xFFFF;
            sum++;
        }
    }
    return ~(sum & 0xFFFF);
}
```

- ♦ One's complement sum
  - » Mod-2 addition **with carry-out**
  - » Carry-out in the most-significant-bit is added to the least-significant bit
  - » Get one's complement of "one's complement sum"

	<b>1010011</b>
	<b>0110110</b>
	<hr/>
carry-out ①	<b>0001001</b>
Carry wrap-around	<b>0000001</b>
	<hr/>
	<b>0001010</b>
	<hr/>
	<b>One's complement = 1110101</b>

### 1.7.1 Cada pacote contém:

- Versão do protocolo IP
- Tamanho do Header
- Tipo de serviço
- Tamanho da informação
- Identificador + Flags + Offset de Fragmento (Permite fragmentar mensagens em vários pacotes)
- Time To Live (para os pacotes não ficarem indefinidamente perdidos na rede)
- Upper Layer Protocol
- Checksum do Header
- IP de Origem
- IP de Destino
- Opções (opcional)
- Informação (Normalmente pacote TCP ou UDP)



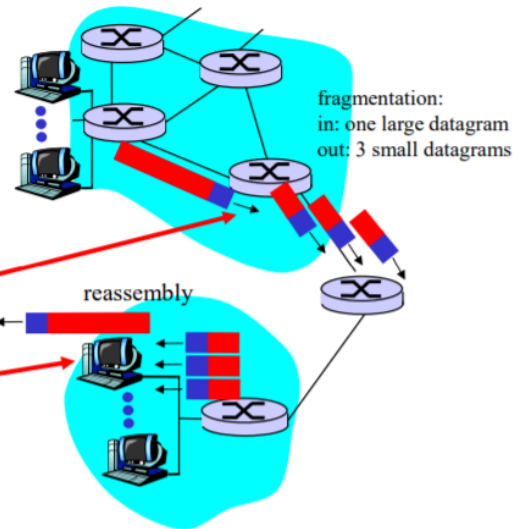
### 1.7.2 Fragmentação IP e Reassembly

- Identificador - Identifica o pacote
- fragflag - 1 se houver mais informação, 0 se for o último fragmento
- Offset - Offset do fragmento em bytes / 8

## *IP Fragmentation and Reassembly*

- ♦ Network links have MTU
  - » MTU - max. transfer size
  - » largest possible link-level frame
  - » different link types, different MTUs

- ♦ Large IP datagram is fragmented
  - » one datagram → n datagrams
  - » “reassembled” at final destination
  - » IP header bits used to identify, order related fragments



#### Example

- 4000 byte datagram
- 3980 bytes data + 20 bytes IP header
- MTU = 1500 bytes

length	ID	fragflag	offset
=4000	=x	=0	=0

One large datagram becomes several smaller datagrams

1480 bytes in data field

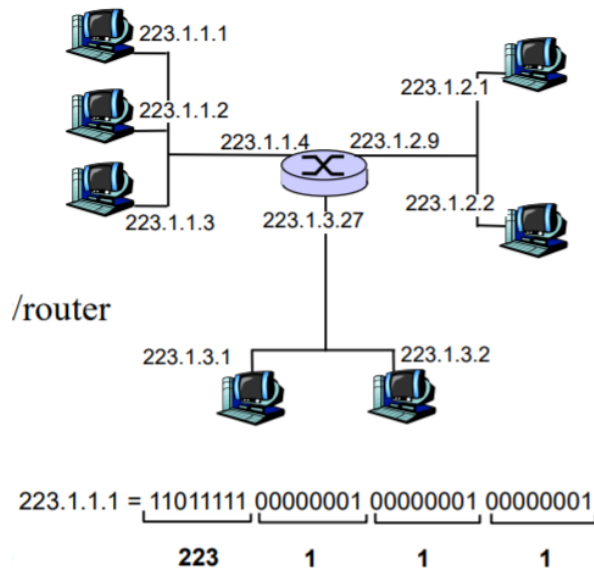
offset =  
1480/8

length	ID	fragflag	offset
=1500	=x	=1	=0
=1500	=x	=1	=185
=1040	=x	=0	=370

### 1.7.3 Endereço IP

Endereço IP - é formado por um identificador de 32-bit para uma interface host/router Interface possuem:

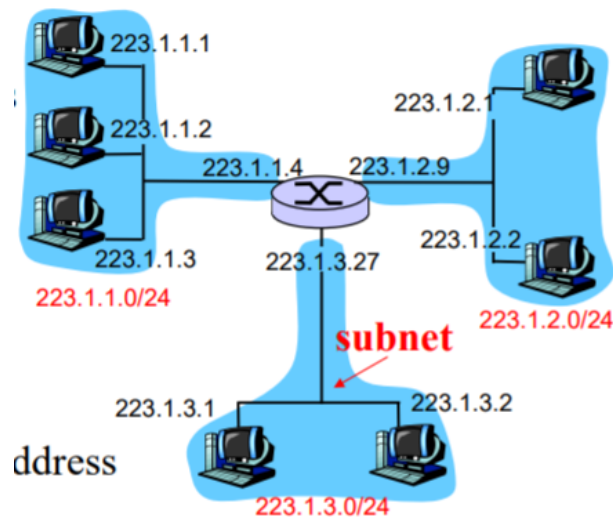
- conexão entre host/router e link físico(physical link)
- routers com multiplas interfaces
- endereços IP associados com as interfaces



### 1.7.4 Subnets

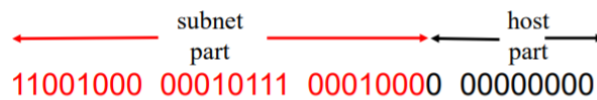
- Parte mais significativa do IP: Subnet parte
- Parte menos significativa: host(interface) parte
- Subnet é um set de interfaces
- cada um tem a subnet parte do IP igual para comunicação
- Cada computador consegue aceder a outro sem intervenção do router

### Network consisting of 3 subnets



CIDR - Classless InterDomain Routing

- a porção de bits do endereço subnet tem tamanho arbitrário
- formato -j a.b.c.d/x, em que x é o número de bits na porção do endereço subnet



200.23.16.0/23

### 1.7.5 Endereços especiais

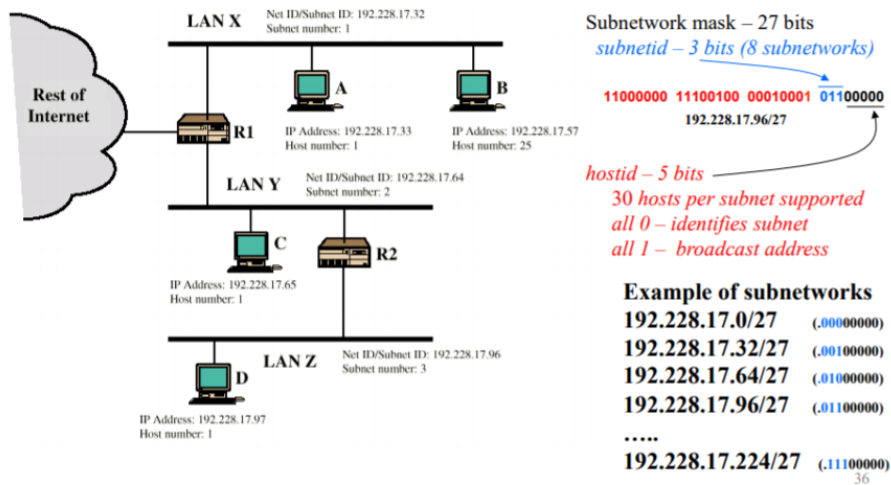
0 0																												This host
0 0				...				0 0				Host																A host on this network
1 1																												Broadcast on the local network
Network								1 1 1 1				...				1 1 1 1				Broadcast on a distant network								
127				(Anything)																								Loopback

- 0.0.0.0 - este host
- 127.0.0.0 - loopback
- 255.255.255.255 - broadcast
- x.x.255.255 - broadcast na subnet x.x.0.0/16
- x.x.0.0 - subnet x.x.0.0/16

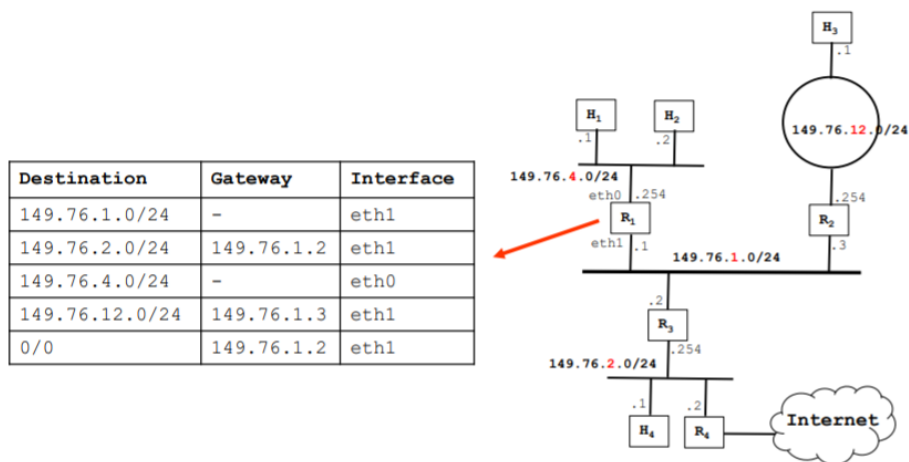
De Notar: - Uma subrede xx.xx.xx.0/24 suporta 255 endereços, no entanto, dois já estão reservados (xx.xx.xx.0 e xx.xx.xx.255), logo só suporta 253 máquinas.

### Forming Sub-Networks (importante)

Network **192.228.17.0/24** is divided in **8 subnetworks** → masks of 27 bits



### Criar table em R1 (importante)



função IP forwarding (importante)

- ◆ Forwarding table has entries in format

`<networkAddress/mask, port>`

- ◆ Forwarding function

- » When a datagram arrives with destination address **A**, then

- For each entry of the forwarding table

- ◆ `val = A & mask*` (e.g., `mask=8, mask*=255.0.0.0`)

- ◆ if ( `val == networkAddress & mask*` )

- add corresponding output port to the set of candidate ports

- Select the port with the largest mask → most specific route

- » Example

- `frdTbl = {<128.32.1.5/16, 1>, <128.32.225.0/18, 3>, <128.0.0.0/8, 5>}`

- Datagram with destination address `A=128.32.195.1`

- Set of candidate output ports → {1, 3, 5}.

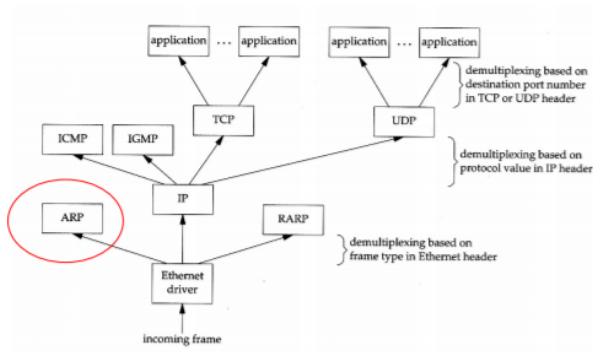
- Selected port → **3** ← largest mask, 18 bits

## 1.8 Address Resolution Protocol APR

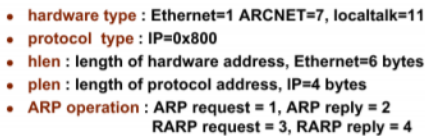
### *Demultiplexing*

---

- ♦ Ethernet header (type)
  - » IP - 0x0800
  - » ARP - 0x0806
  - » RARP - 0x8035
  - » IPX - 0x8037
  - » IPv6 - 0x86DD
  - » MPLS - 0x8847
- ♦ IP header (protocol)
  - » ICMP - 1
  - » IGMP - 2
  - » TCP - 6
  - » UDP - 17
- ♦ TCP/UDP header (port)
  - » FTP - 21
  - » Telnet - 23
  - » HTTP - 80
  - » SMTP - 25

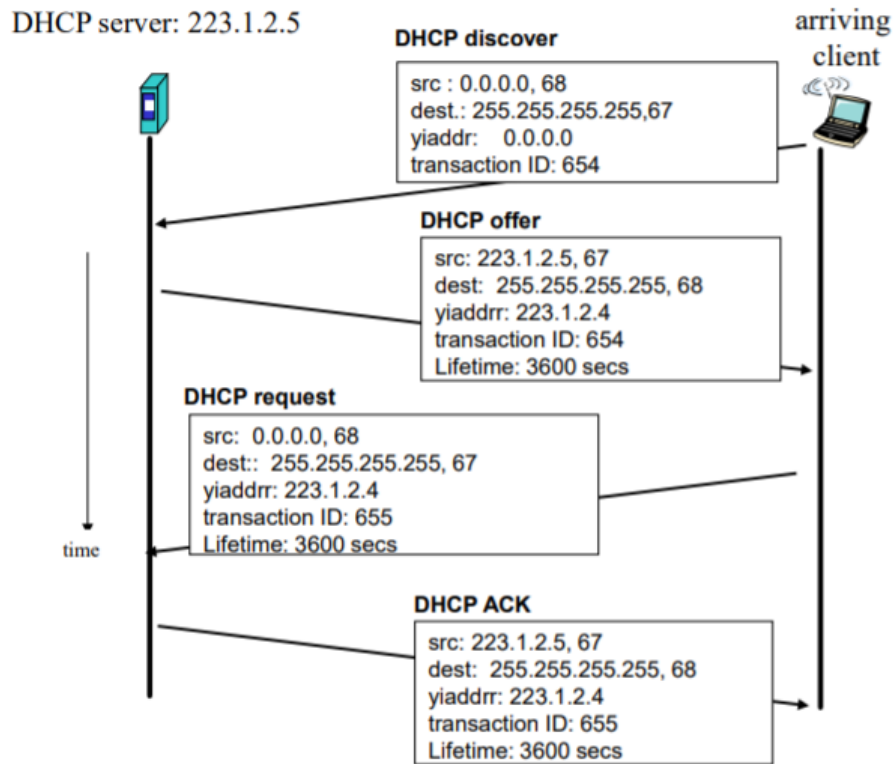


- Uma interface de rede tem 1 endereço MAC e 1 (ou mais) endereços IP
- ARP - protocolo usado para obter o endereço MAC associado a um endereço IP dado
- RARP - reverso de ARP - protocolo usado para obter o endereço IP associado ao endereço MAC



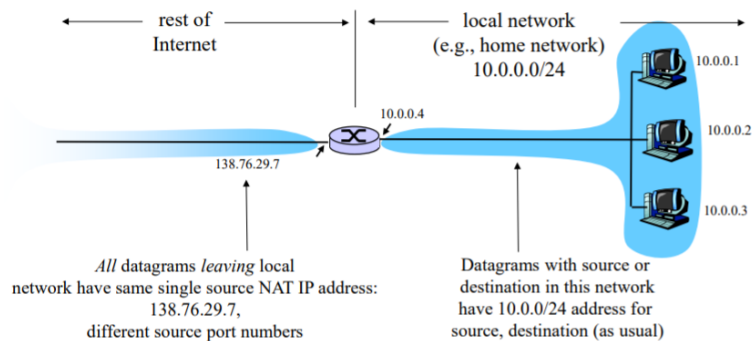
- controla o Domain Name Service (DNS)
  - associa os nomes do domínio
  - resolve conflitos
- o host obtém endereços IP de forma hard-coded pelo sistema admin num ficheiro ou pelo DHCP
- DHCP: Dynamic Host Configuration Protocol
  - Dinamicamente recebe endereços do servidor
  - "plug-and-play"
  - permite descobrir e obter endereços da rede do servidor
  - reusa os endereços
  - Overview:
    - \* O host faz broadcast de "DHCP discover" (msg)
    - \* O servidor DHCP oferece um endereço, enviando em broadcast "DHCP offer" (msg)
    - \* O host pede esse endereço enviando em broadcast "DHCP request" (msg)
    - \* Se tudo estiver em ordem, o DHCP responde em broadcast com um "DHCP ACK" (msg)
    - \* Todas as mensagens entre o host e o DHCP possuem um id de transação

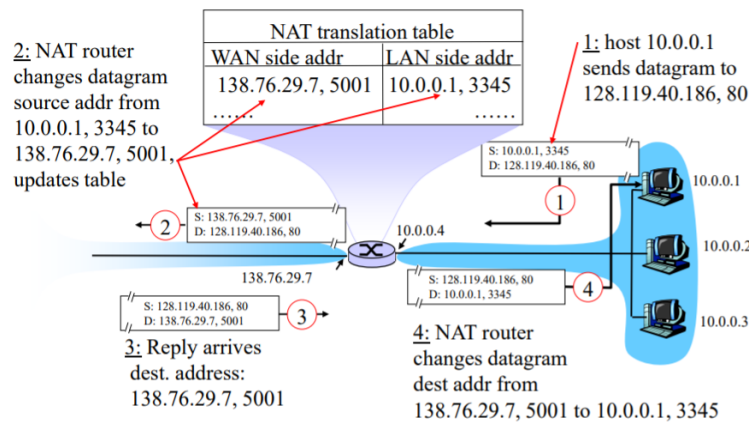




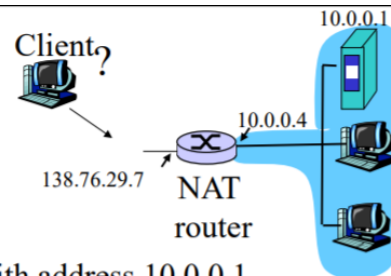
## 1.10 NAT - Network Address Translation

- Permite que cada computador tenha um IP interno numa rede, sendo o IP externo diferente
- Para isso, possui uma hash table a que associa um IP interno e uma porta a um número, que será a porta de saída
- Caso um cliente se queira ligar a um servidor dentro de uma rede com NAT, é necessário configurar o port forwarding





### 1.10.1 NAT Transversal



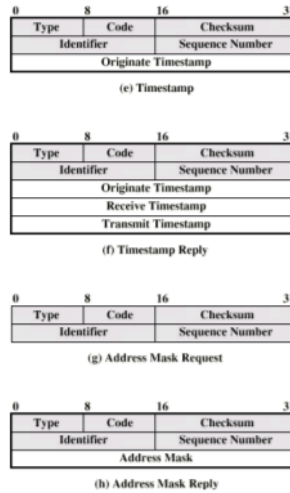
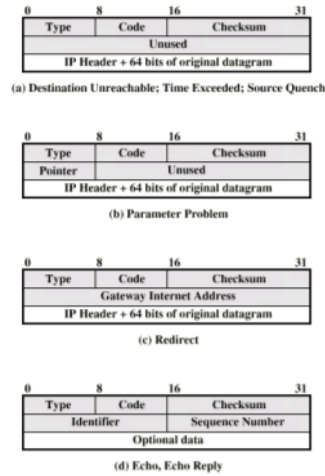
- ♦ Client wants to connect to server with address 10.0.0.1
  - » but server address 10.0.0.1 is private
  - » only one externally visible NATed address: 138.76.29.7
- ♦ Possible solution – **Port forwarding**
  - » **statically configure NAT**
    - to forward incoming connection requests at given port to server
  - » e.g., (138.76.29.7, port 2500) always forwarded to 10.0.0.1 port 25000

### 1.11 ICMP - Internet Message Control Protocol

- Usado pelo router ou host para mandar mensagens de erro ou de controlo (como o ping)

### 1.11.1 IP datagramas info:

#### ♦ Carried in IP datagrams



Type	Code	Description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
5		Redirect
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

62

### 1.11.2 Traceroute and ICMP

- Permite fazer traceroute enviando mensagens com TTL=1,2,3... e esperando respostas de erro "TTL expired" até receber um "Host unreachable"

#### ♦ Source sends series of UDP segments to destination

- » first segment has TTL =1
- » second segment has TTL=2, ...
- » unlikely port number

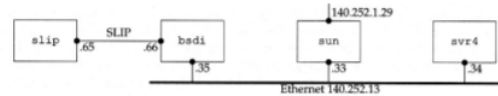
```
svr4% traceroute slip
traceroute to slip (140.252.13.65), 30 hops max, 40 byte packets
1 bsd1 (140.252.13.35) 20 ms 10 ms 10 ms
2 slip (140.252.13.65) 120 ms 120 ms 120 ms
```

#### ♦ When n<sup>th</sup> datagram arrives

to n<sup>th</sup> router

- » router discards datagram
- » sends to source:
  - ICMP TTL expired
  - message includes router name & IP address

```
slip% traceroute svr4
traceroute to svr4 (140.252.13.34), 30 hops max, 40 byte packets
1 bsd1 (140.252.13.66) 110 ms 110 ms 110 ms
2 svr4 (140.252.13.34) 110 ms 120 ms 110 ms
```



#### ♦ When ICMP message arrives, source calculates RTT

#### ♦ Traceroute does this 3 times for each TTL

#### ♦ Stop criterion

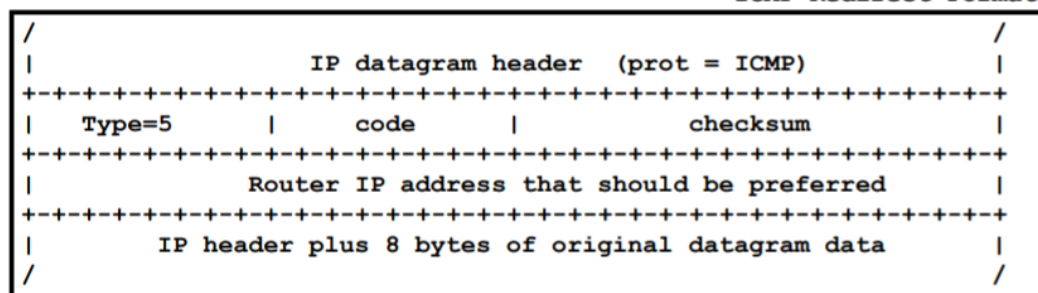
- » UDP segment eventually arrives at destination host
- » Destination returns ICMP "dest port unreachable" packet
- » source stops

### 1.11.3 ICMP Redirect

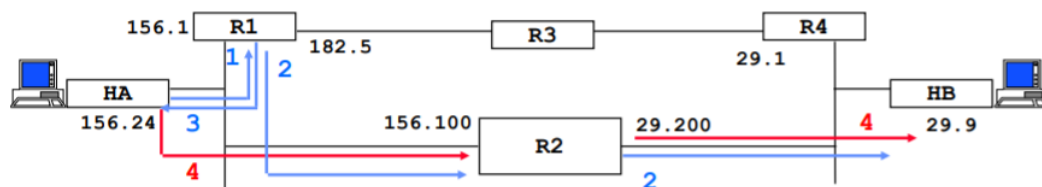
- ICMP Redirect - Permite informar outros hosts do caminho mais rápido para determinado destino

- ♦ General routing principle of the TCP/IP architecture
  - » routers have extensive knowledge of routes
  - » hosts have minimal routing information → learn routes also from ICMP redirects
- ♦ ICMP redirect message
  - » Sent by router R1 to source host A
    - when R1 receives a packet from A with destination = B, and R1
      - ♦ finds that the next hop is R2 and
      - ♦ A is on-link with R2
  - » R1 sends ICMP redirect to A saying next hop for destination B is R2
  - » A updates its forwarding table with a host route

ICMP Redirect Format



### ICMP Redirect Example



	dest IP addr	srce IP addr	prot	data part
1:	193.154.29.9	193.154.156.24	udp	xxxxxxx
2:	193.154.29.9	193.154.156.24	udp	xxxxxxx
3:	193.154.156.24	193.154.156.1	icmp	type=redir code=host cksum 193.154.156.100 xxxxxxx (28 bytes of 1)
4:	193.154.29.9	193.154.156.24	udp	.....

After 4

```
HA$ netstat -nr
```

Routing Table:

Destination	Gateway	Flags	Interface
127.0.0.1	127.0.0.1	UH	lo0
193.154.29.9	193.154.156.100	UGH	eth0
193.154.156.0	193.154.156.24	U	eth0
224.0.0.0	193.154.156.24	U	eth0
default	193.154.156.1	UG	eth0

Flags:

U - route Up

G - route to a Gateway (next hop router)

H - route to a Host

67

## 1.12 IPv6

- IPv4
  - espaço reduzido de endereçamento (32 bits)
  - uso não contínuo
  - o uso de algumas soluções como private networks (NAT) e classless networks (CDIR) superava os problemas acima
- IETF developed new IP version: IPv6
  - Uso dos mesmos princípios do IPv4
  - muitas melhorias
  - Header foi redefinido

### 1.12.1 IPv6 - Melhorias

- Endereços 128 bits (16 octets, 8 shorts). No classes
- Melhor QoS suporte (native flow level)
- funções nativas de segurança (autenticação, data encriptação)
- Autoconfiguração (Plug-n-play)
- Routing
- Multicast

### 1.12.2 Representação dos endereços

- 8 x 16 bit, hexadecimal, separados por:  
47CD : 1234 : 3200 : 0000 : 0000 : 4325 : B792 : 0428
- formato comprimido:  
FF01:0:0:0:0:43 -> FF01::43
- compatibilidade com IPv4:  
0:0:0:0:0:13.1.68.3 or ::13.1.68.3
- Loopback endereço:  
::1
- Prefixo de rede "/", igual ao IPv4:  
FEDC:BA98:7600::/40 -> network prefix = 40 bits

### 1.12.3 Endereços Reservados

Allocation	Prefix (binary)	Fraction of Address Space
-----	-----	-----
Unassigned	0000 0000	1/256
Unassigned	0000 0001	1/256
Reserved for NSAP Allocation	0000 001	1/128
Unassigned	0000 01	1/64
Unassigned	0000 1	1/32
Unassigned	0001	1/16
Global Unicast	001	1/8
Unassigned	010	1/8
Unassigned	011	1/8
Unassigned	100	1/8
Unassigned	101	1/8
Unassigned	110	1/8
Unassigned	1110	1/16
Unassigned	1111 0	1/32
Unassigned	1111 10	1/64
Unassigned	1111 110	1/128
Unassigned	1111 1110 0	1/512
Link-Local Unicast Addresses	1111 1110 10	1/1024
Site-Local Unicast Addresses	1111 1110 11	1/1024
Multicast Addresses	1111 1111	1/256

### 1.12.4 Tipo de Endereços

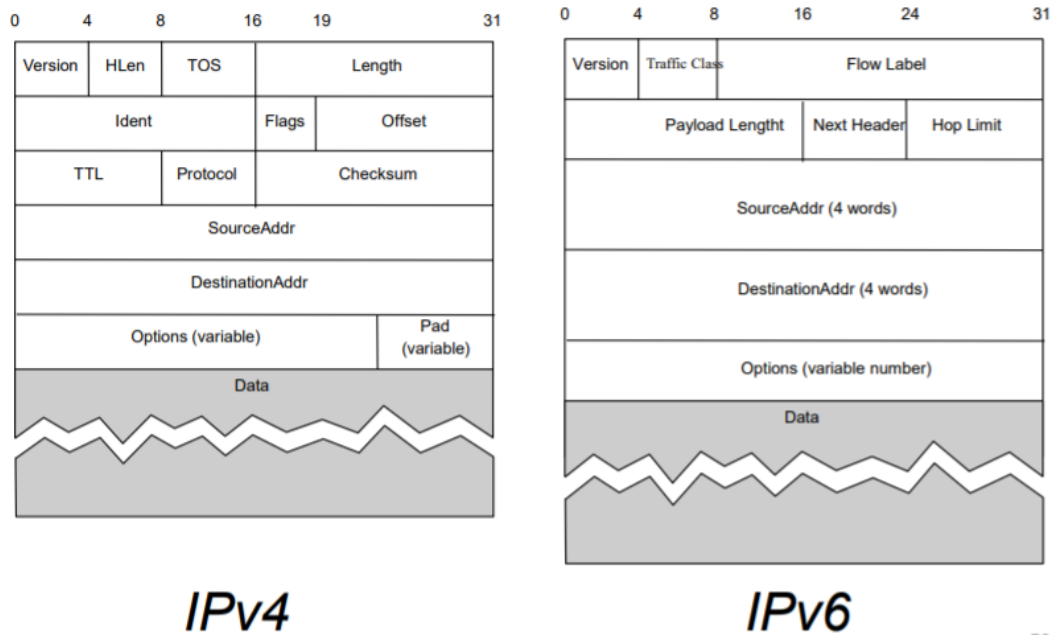
- Link-Local
  - Usado para a comunicação entre hosts na mesma LAN/link
  - Endereço criado pelo endereço MAC
  - Routers não enviam pacotes tendo endereços de destino Link-Local
- Global Unicast

- Endereços globais
- Endereços: prefixo de rede + identificador do computador
- Prefixos estruturados: Agregação de rede; menos entradas nas router forwarding tables
- Anycast
  - Endereços de grupo
  - Um pacote é recebido por um e um só membro do grupo
- Multicast
  - Endereços de grupo
  - Um pacote pode ser recebido por vários membros do grupo

#### 1.12.5 Formato dos Endereços

n bits             m bits             128-n-m bits			Global Unicast Address
+-----+-----+-----+			(2000::/3)
001 global rout prefix   subnet ID   interface ID			
+-----+-----+-----+			
10             54 bits             64 bits			Link-Local Unicast address
+-----+-----+-----+			(fe80::/10)
1111111010        0        interface ID			
+-----+-----+-----+			
10             54 bits             64 bits			Site-Local Unicast address
+-----+-----+-----+			(fec0::/10) (not used)
1111111011   subnet ID   interface ID			
+-----+-----+-----+			
n bits             128-n bits			Anycast address
+-----+-----+			
subnet prefix   0000000000000000			
+-----+-----+			
8             4             4             112 bits			Multicast address
+-----+-----+-----+			Scope - link, site, global, ...
11111111   flgs   scop        group ID			(ff::/8)
+-----+-----+-----+			

### 1.12.6 Headers IPv4 e IPv6

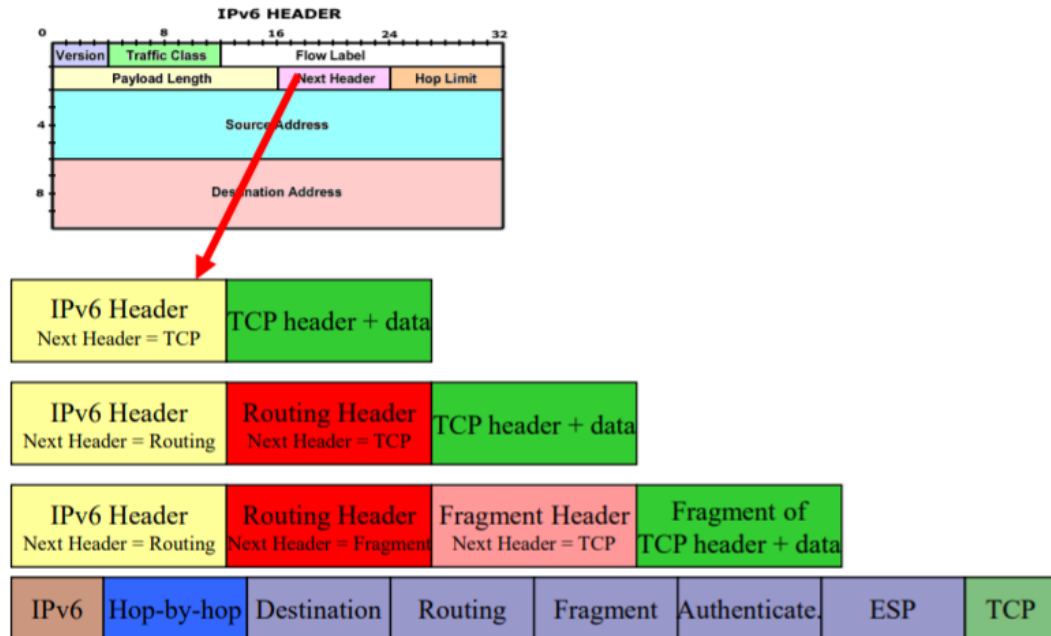


### 1.12.7 IPv6 Header

- Flow label - identifica o fluxo do pacote
  - QoS, ressalva de recursos
  - Pacotes recebem o mesmo serviço
- Payload length - Header não incluído
- Next header - identifica o próximo header/extensão
- Options - incluída nas extensões dos headers

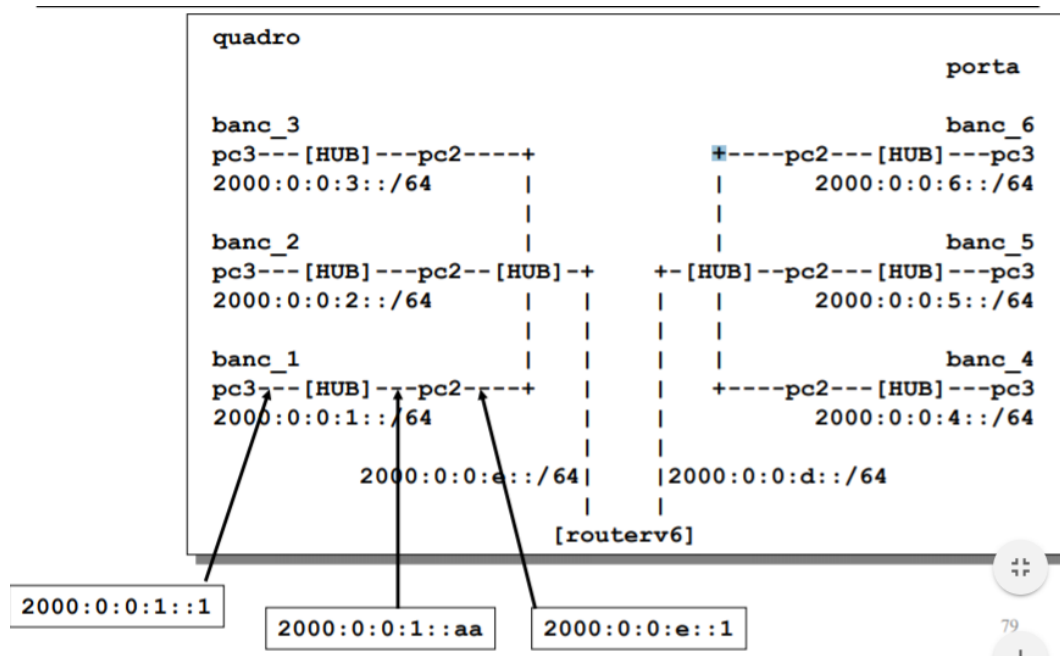


### 1.12.8 Extension Headers



- Hop-by-Hop: inspeciona todos os nodes atravessados pelo pacote
- Destination: informação do node de destino
- Routing: Lista dos nodes para serem visitados pelo pacote
- Fragmentation: feito pelo source, deve encontrar MPU
- Authentication: autenticação (assinatura) do header do pacote
- ESP: encriptação da informação(data)

### 1.12.9 Exemplo da Rede do Laboratório



### 1.12.10 Protocol Neighbor Discovery (ND)

IPv6 node usa ND para:

- Encontrar outros nodes no mesmo link/LAN
- Encontrar o node do endereço MAC (ND substitui ARP)
- Encontrar routers na sua rede
- Manter/Segurar a informação sobre os nodes vizinhos

ND similar às funções IPv4:

- ARP IPv4
- ICMP Router Discovery
- ICMP Redirect

### 1.12.11 ND Mensagens

- ICMP mensagens (over IP), Uso de endereços Link Local
- **Neighbor Solicitation:** Enviado pelo host para obter o endereço MAC de um vizinho/para verificar a sua presença

- **Neighbor Advertisement:** resposta ao pedido
- **Router Advertisement:** Informação sobre o prefixo da rede, periodica ou abaixo do pedido. Enviado pelo router para o endereço IP do Link Local multicast
- **Router Solicitation:** Hosts solicitam do router uma mensagem Router Advertisement
- **Redirect:** Usado pelo router para informar o host acerca da melhor rota para o destino