

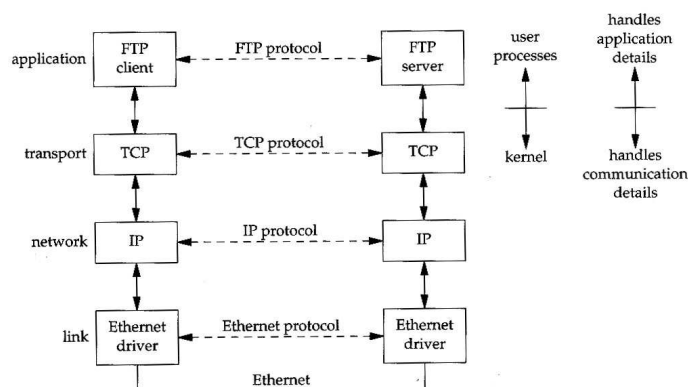
Interligação de Redes

Protocolos TCP/IP

FEUP/DEEC/CDRC I – 2002/03
MPR/JAR

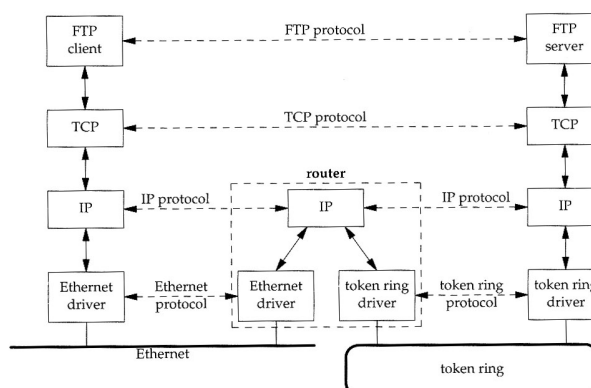
Pilha Protocolar TCP/IP

- ◆ Acesso à rede (subrede)
 - » Interface ao meio físico
 - » Carta de rede, *device driver*
- ◆ Rede (*internetworking*)
 - » Transferência de pacotes na rede (entre subredes)
 - » IP – Internet Protocol
 - » ICMP – Internet Control Message Protocol
 - » IGMP – Internet Group Management Protocol
- ◆ Transporte
 - » Multiplexagem de fluxos de dados entre duas máquinas
 - TCP – Transmission Control Protocol
 - UDP – User Datagram Protocol
- ◆ Aplicação
 - » Aplicações de rede
 - Telnet, FTP, SMTP, SNMP, etc.
 - » Modelo cliente-servidor



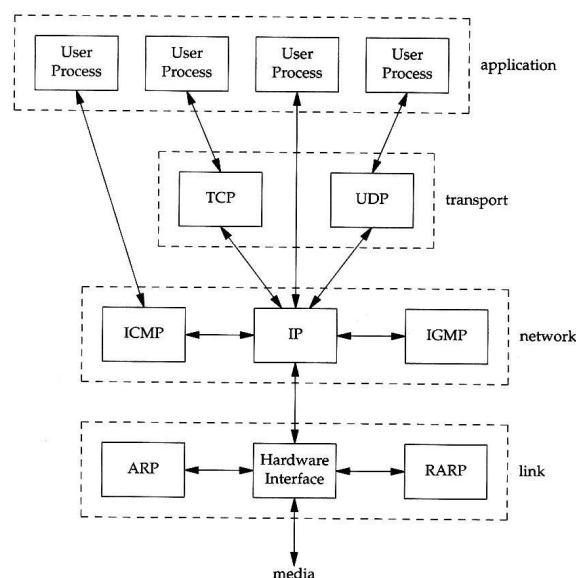
Interligação de Redes

- ◆ Redes de diferentes organizações e usando variadas tecnologias de acesso devem poder comunicar
- ◆ Internet → rede única, virtual
 - » *End-Systems* → computadores (*hosts*)
 - » *Intermediate-Systems* → *routers*
 - » Comunicação global
 - » Interligação de redes com *routers*
 - Um *router* tem uma interface por cada rede que interliga
- ◆ Comunicação protocolar
 - » Camadas de Aplicação e Transporte
 - Extremo-a-extremo (entre *hosts*)
 - » Camada de Rede
 - entre máquinas adjacentes (*routers* e *hosts*)

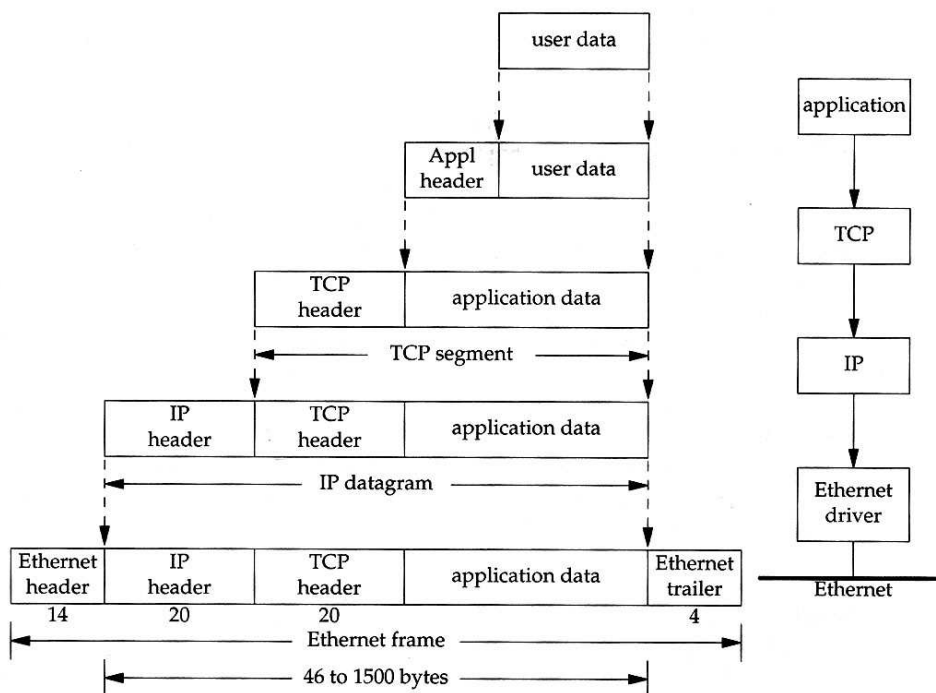


Arquitetura TCP/IP

- ◆ TCP
 - » Orientado às ligações
 - » Transporte fiável entre máquinas
- ◆ UDP
 - » Envia / recebe datagramas das aplicações
 - » Transporte não fiável
- ◆ IP
 - » Protocolo central da pilha
 - » Encaminha datagramas
- ◆ ICMP
 - » Auxiliar do IP
 - » Envia mensagens de erro
- ◆ IGMP
 - » Gere grupos de *multicasting*
- ◆ ARP, RARP
 - » Resolução de endereços IP em endereços físicos



Encapsulamento



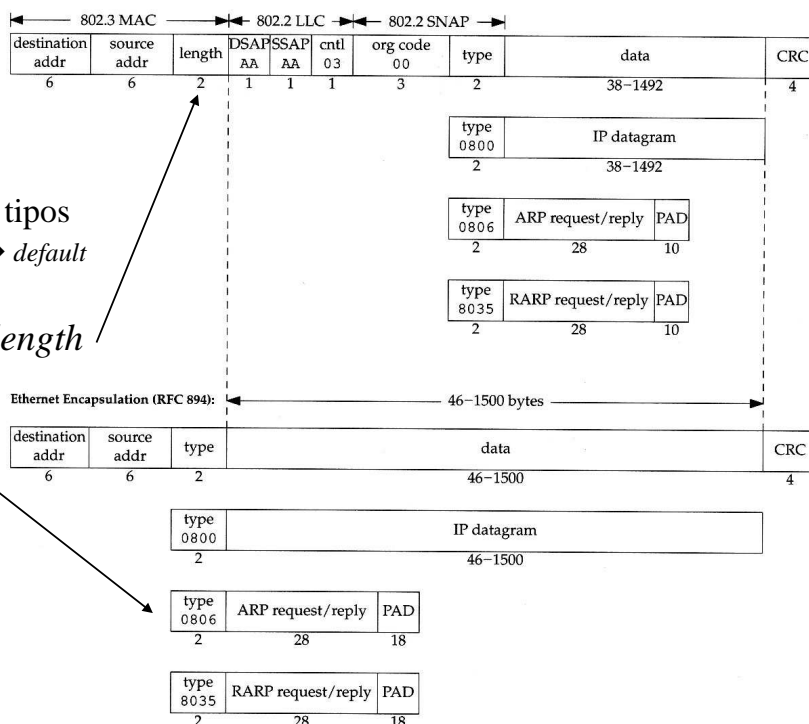
Encapsulamento Ethernet

Cartas Ethernet

- » Devem receber
 - encapsulamento IEEE 802
 - encapsulamento Ethernet
- » Se conseguem enviar os 2 tipos
 - encapsulamento Ethernet → *default*

Valores válidos IEEE 802 length

- » Diferentes de *type* válidos
 - Ex. 0x0800 = 2048



Desmultiplexagem

» Cabeçalho TCP/UDP (porta)

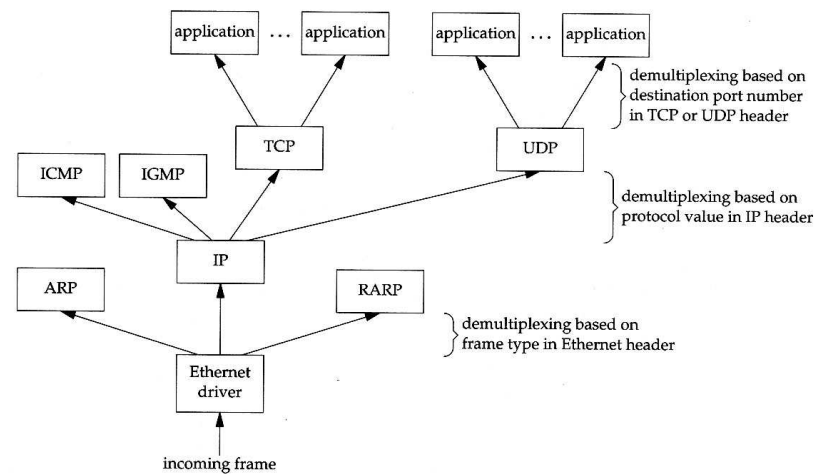
- FTP → 21
- Telnet → 23
- ...

» Cabeçalho IP (protocolo)

- ICMP → 1
- IGMP → 2
- TCP → 6
- UDP → 17

» Cabeçalho Ethernet (tipo)

- IP → 0x0800
- ARP → 0x0806
- RARP → 0x8035



Internet Protocol

♦ IP

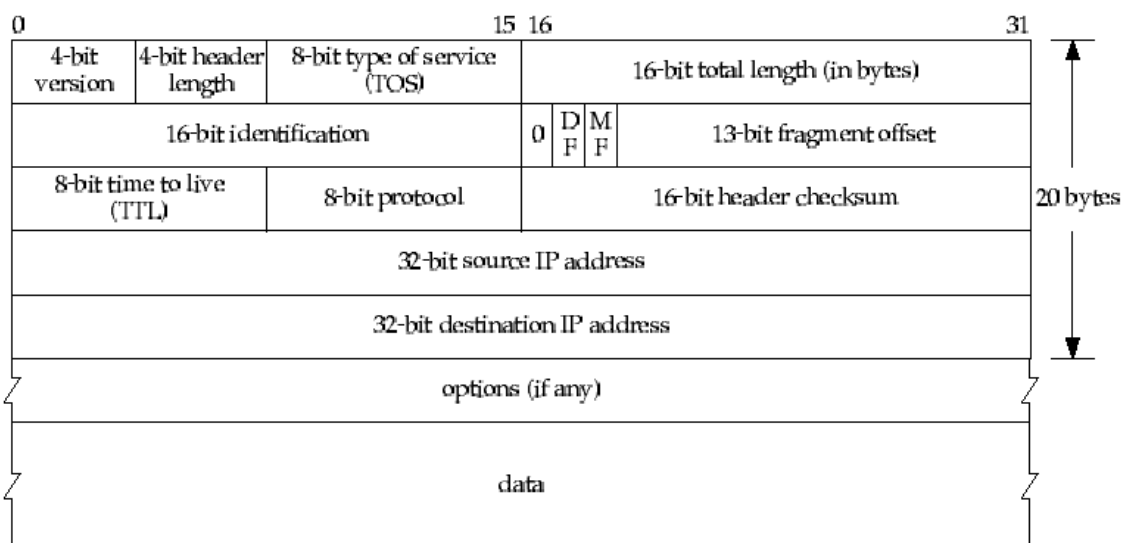
- » Comutação de datagramas
- » Serviço não fiável
 - Não há garantia de entrega no destino
 - Serviço *best effort*
- » Datagramas para um mesmo destino podem
 - Seguir caminhos diferentes
 - Ser entregues fora de ordem

♦ Cada *router*

- » Analisa o endereço de destino
- » Encaminha o datagrama recebido para uma das suas interfaces
- » Em situações de erro
 - Elimina datagrama
 - Envia mensagem de erro ICMP para a origem

Cabeçalho IP

IP Header



Cabeçalho IP

- » Versão – IPv4
- » Comprimento do cabeçalho
 - Número de palavras (de 32 bits) do cabeçalho
 - Limite do cabeçalho → 60 octetos
- » TOS (*type of service*)
 - 3 bits de precedência
 - 4 bits TOS
 - 1 bit não usado
 - Usado em novas redes (QoS, DiffServ)
- » Comprimento total
 - Comprimento total do datagrama
 - Máximo: 65535 octetos
- » Identificação
 - Identifica unicamente um datagrama
 - Incrementado de uma unidade por cada datagrama enviado
- » *Flags*
 - DF – *Don't Fragment*
 - MF – *More Fragments*
- » *Fragment offset*
 - Indica a posição dum fragmento em relação ao datagrama inicial
- » Tempo de vida
 - Número máximo de *routers* visitáveis por um datagrama
 - Inicializado pelo *host* (32 ou 64)
 - Decrementado de um quando visita *router*
- » Protocolo
 - Usado para desmultiplexagem
- » *Checksum*
 - Calculado sobre o cabeçalho
- » Endereço de origem (32 bits)
- » Endereço de destino (32 bits)
 - endereço do *host* final
- » Opções
 - Registo de rota
 - Registo de tempos
 - Encaminhamento definido pela origem
 - ◆ *Loose source routing*
 - ◆ *Strict source routing*

Type Of Service

Application	Minimize delay	Maximize throughput	Maximize reliability	Minimize monetary cost	Hex value
Telnet/Rlogin	1	0	0	0	0x10
FTP					
control	1	0	0	0	0x10
data	0	1	0	0	0x08
any bulk data	0	1	0	0	0x08
TFTP	1	0	0	0	0x10
SMTP					
command phase	1	0	0	0	0x10
data phase	0	1	0	0	0x08
DNS					
UDP query	1	0	0	0	0x10
TCP query	0	0	0	0	0x00
zone transfer	0	1	0	0	0x08
ICMP					
error	0	0	0	0	0x00
query	0	0	0	0	0x00
any IGP	0	0	1	0	0x04
SNMP	0	0	1	0	0x04
BOOTP	0	0	0	0	0x00
NNTP	0	0	0	1	0x02

MTU – Maximum Transmission Unit

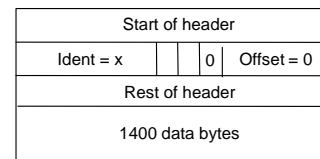
- ◆ Cada rede limita o tamanho do campo de dados nas respectivas tramas – este limite designa-se *Maximum Transmission Unit (MTU)*
- ◆ Se o tamanho de um datagrama IP for superior ao MTU de uma rede é necessário fragmentá-lo em datagramas mais pequenos, podendo ser necessário re-fragmentar
- ◆ Se um datagrama atravessar várias redes, o menor MTU condiciona o tamanho dos datagramas IP (fragmentos) entregues no destino

Network	MTU (bytes)
16 Mbits/sec token ring (IBM)	17914
4 Mbits/sec token ring (802.5)	4464
FDDI	4352
Ethernet	1500
IEEE 802.3/802.2	1492
X.25	576
Point-to-Point (low delay)	296

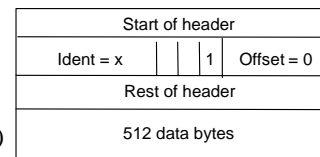
Fragmentação IP

- ◆ A fragmentação é realizada quando necessário em *routers* (a fragmentação na origem é evitada)
- ◆ A reconstrução de datagramas é feita no destino
- ◆ Informação usada na fragmentação (cabeçalho IP)
 - » *Identification*
 - Valor único atribuído a cada datagrama pelo emissor
 - Copiado em cada fragmento do datagrama original
 - » *Data length*
 - Comprimento dos dados (do fragmento) em octetos
 - » *Offset*
 - Posição do fragmento no datagrama ($n * 64$ bits)
 - » *More Fragments (MF flag)*
 - Indica que o fragmento não é o último
- ◆ Problema
 - » Perda de fragmentos
- ◆ Solução → tempo máximo de
 - » Reconstrução
 - » Vida de um datagrama

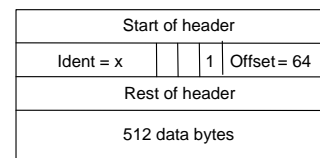
Rede 1
MTU = 1500



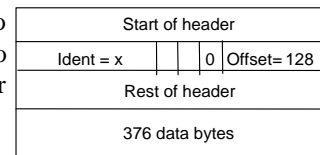
Rede 2
MTU = 532
(permite até 512 octetos de dados)



Valor do campo *offset* = *offset real* / 8

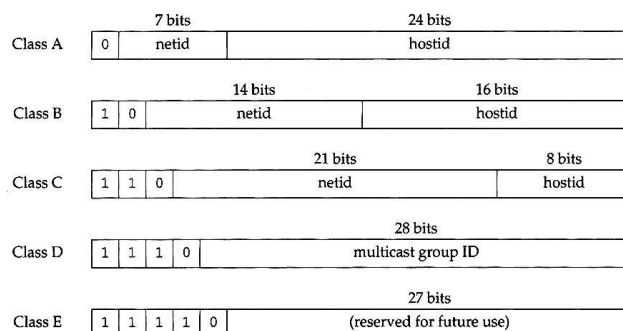


Com exceção do último fragmento, o tamanho do campo de dados deve ser múltiplo de 8 octetos



Endereçamento IP

- ◆ Endereços IP
 - » Um por interface, 32 bits (IPv4)
 - » Exemplo: 142.252.13.33
- ◆ Atribuição de endereços
 - » InterNic → atribui *netid*
 - » Administrador de rede → atribui *hostid*
 - All 0 → representa a subrede <*netid*>
 - All 1 → endereço de *broadcast* na subrede
- ◆ 5 classes de rede
 - » Classe A
 - 126 redes: 1.x.x.x até 126.x.x.x → ocupadas
 - ◆ All 0 → reservada
 - ◆ 01111111 (127) → loopback
 - » Classe B
 - $2^{14} - 2 = 16,382$ redes
 - 128.x.x.x até 191.x.x.x → todas ocupadas
 - » Classe C
 - $2^{21} - 2 = 2,097,150$ redes
 - 192.x.x.x até 223.x.x.x → quase todas ocupadas
- ◆ Tipos de endereço
 - » *Unicast*, *multicast*, *broadcast*



Classe	Valores
A	0.0.0.0 → 127.255.255.255
B	128.0.0.0 → 191.255.255.255
C	192.0.0.0 → 223.255.255.255
D	224.0.0.0 → 239.255.255.255
E	240.0.0.0 → 247.255.255.255

Interface de Loopback

- ◆ Interface de teste, num computador
 - » curto-circuito à rede
 - » *netid* → 127.0.0.0
 - » *hostid* → 127.0.0.1 (convenção)
 - » nome → *localhost*

- ◆ Datagrama enviado para *localhost* não é visto na rede

- ◆ Permite que um cliente e um servidor na mesma máquina comuniquem usando TCP/IP

IP – Subredes

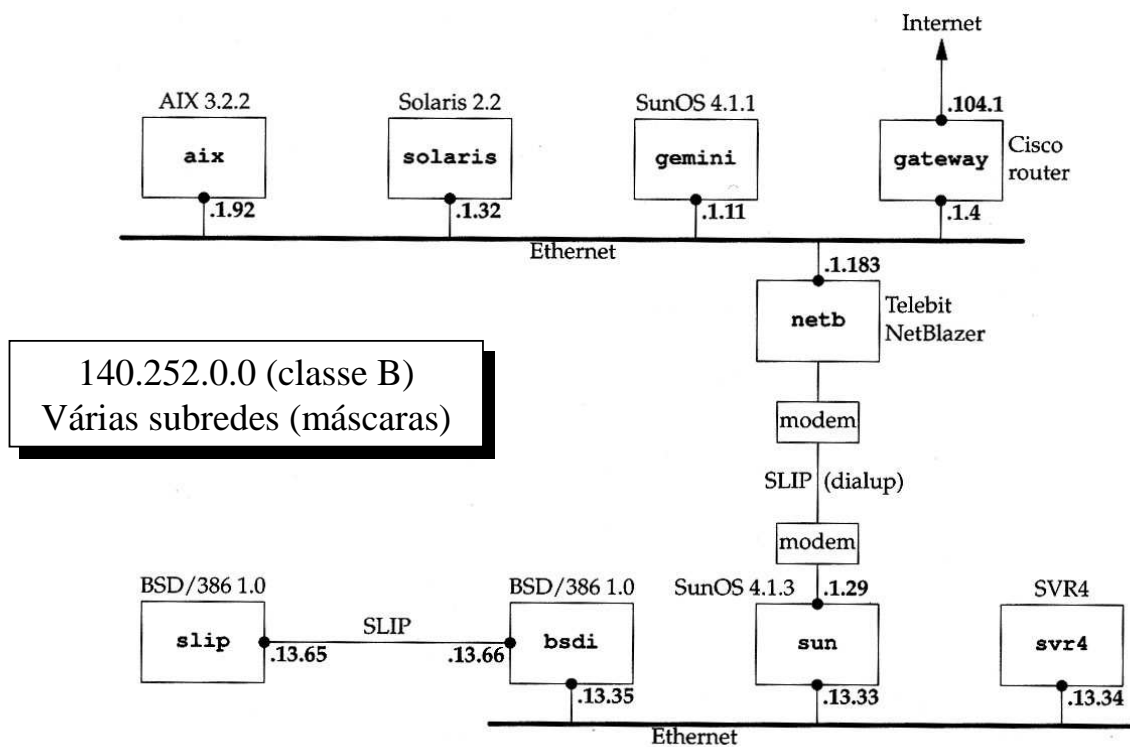
- ◆ Divisão de uma rede em subredes
 - » Endereço de um *host* → identificador de subrede + identificador de *host*
 - » Exemplo: 1 LAN → 1 subrede
 - » Todos os computadores de uma subrede têm o mesmo endereço de subrede

- ◆ Construção de um número arbitrário de redes numa empresa
 - » Esconde as redes da empresa do resto da Internet
 - » Do exterior as redes da empresa são vistas como uma rede única

- ◆ *Routers* da empresa encaminham usando informação da subrede

- ◆ Máscara de subrede
 - » Identifica bits que devem ser interpretados como bits de (sub)rede

Rede Exemplo



Encaminhamento IP

- ◆ Cada sistema IP (*host* ou *router*) deverá possuir uma tabela de encaminhamento; a tabela é relativamente simples no caso de um *host*, que pode ser configurado para funcionar, em certos casos, como *router*
 - » No caso de o sistema de destino estar directamente ligado ao *host* (e.g., uma ligação ponto a ponto) ou numa rede partilhada (e.g., uma LAN), os pacotes IP são enviados directamente para o destino
 - » Caso contrário, os pacotes são enviados pelo *host* para um *default router*
- ◆ Cada entrada de uma tabela de encaminhamento contém
 - » Um endereço IP de destino, que pode ser um *host address* (*netid* + *hostid*) ou um *network address* (*netid*, *hostid* = 0); estes casos são distinguidos por uma *flag H*
 - » A indicação de uma rota, que pode ser indirecta (endereço IP de um *next-hop router*) ou directa (endereço IP de uma interface local); estes casos são distinguidos por uma *flag G*
 - » Interface de rede à qual o datagrama deve ser passado

Encaminhamento IP

- ◆ Para encaminhar um pacote, a tabela é pesquisada com o objectivo de encontrar
 - » Uma entrada que seja igual ao endereço de destino (*net id + hostid*)
 - Ligações ponto a ponto são um exemplo deste caso
 - » Uma entrada que seja igual ao *netid* de destino, no caso de o passo anterior não ter sucesso
 - Todas as máquinas numa LAN são cobertas por este caso
 - » Uma entrada por omissão (*default*), no caso de não existir qualquer entrada que satisfaça os critérios anteriores
- ◆ Se for encontrada uma entrada que satisfaça uma das condições, o pacote é enviado de acordo com o especificado nessa entrada (*flag G*)
 - » para um *next hop router* – o endereço IP do pacote é o da máquina de destino, mas o endereço de nível 2 (trama) é o do *next hop router*
 - » para um destino directamente acessível – o endereço IP do pacote e o endereço de nível 2 (trama) são endereços da máquina de destino
- ◆ Se não for satisfeita qualquer das condições (o que significa que não existe uma entrada *default*), o pacote não é encaminhável, sendo por isso descartado

Exemplos de Tabelas de Encaminhamento

Host 140.252.13.34 (svr4) na rede 140.252.13.32/27

Destination	Gateway	Flags	Refcnt	Use	Interface
140.252.13.65	140.252.13.35	UGH	0	0	emd0
127.0.0.1	127.0.0.1	UH	1	0	lo0
default	140.252.13.33	UG	0	0	emd0
140.252.13.32	140.252.13.34	U	4	25043	emd0

Host / router 140.252.13.33 (sun) na rede 140.252.13.32/27

Destination	Gateway	Flags	Refcnt	Use	Interface
140.252.13.65	140.252.13.35	UGH	0	171	le0
127.0.0.1	127.0.0.1	UH	1	766	lo0
140.252.1.183	140.252.1.29	UH	0	0	sl0
default	140.252.13.183	UG	1	2955	sl0
140.252.13.32	140.252.13.33	U	8	99551	le0

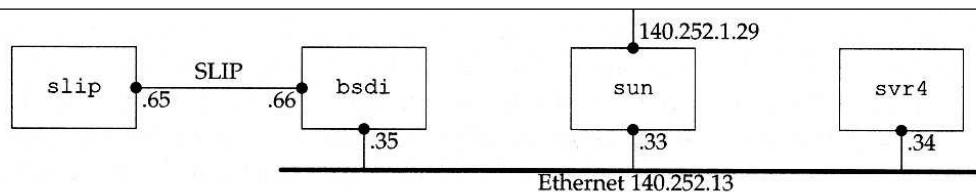


Tabela de Encaminhamento – interpretação

Destination	Gateway	Flags	Refcnt	Use	Interface
140.252.13.65	140.252.13.35	UGH	0	0	emd0
127.0.0.1	127.0.0.1	UH	1	0	lo0
default	140.252.13.33	UG	0	0	emd0
140.252.13.32	140.252.13.34	U	4	25043	emd0

Host 140.252.13.34 (svr4) na rede 140.252.13.32/27 (*subnetid* – 3 bits; *hostid* – 5 bits)

Gateways 140.252.13.33 e 140.252.13.35

Flags

U *route Up*

G *route to a Gateway (next hop router)*

se omitido: rota directa; o endereço da *Gateway* é o endereço IP da interface local do *host* svr4

H *route to a Host* – o endereço IP de destino na tabela é um *host address* completo

se omitido: *route to a network* – o endereço IP de destino na tabela é um *network address*

Refcnt

Número de utilizações activas da rota (por exemplo ligações TCP estabelecidas)

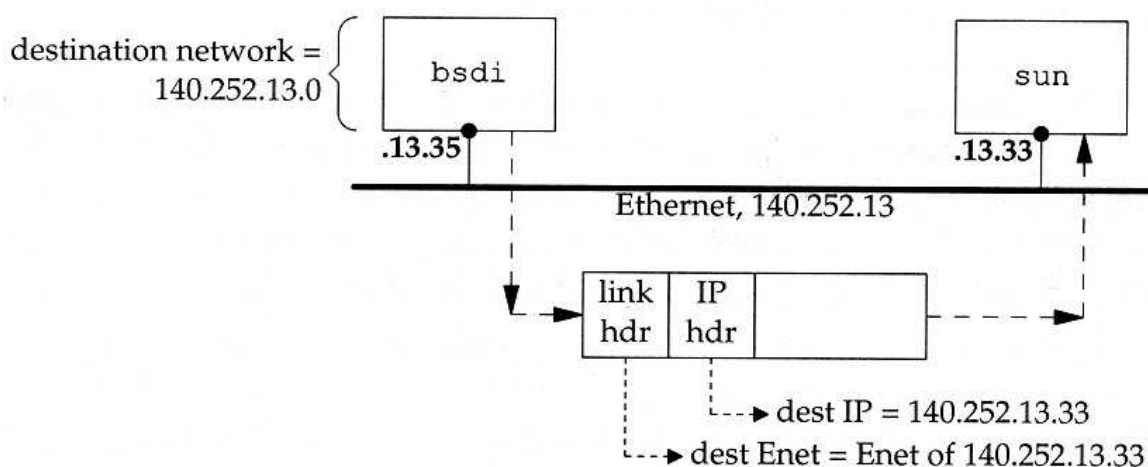
Use

Número de pacotes enviados pela rota

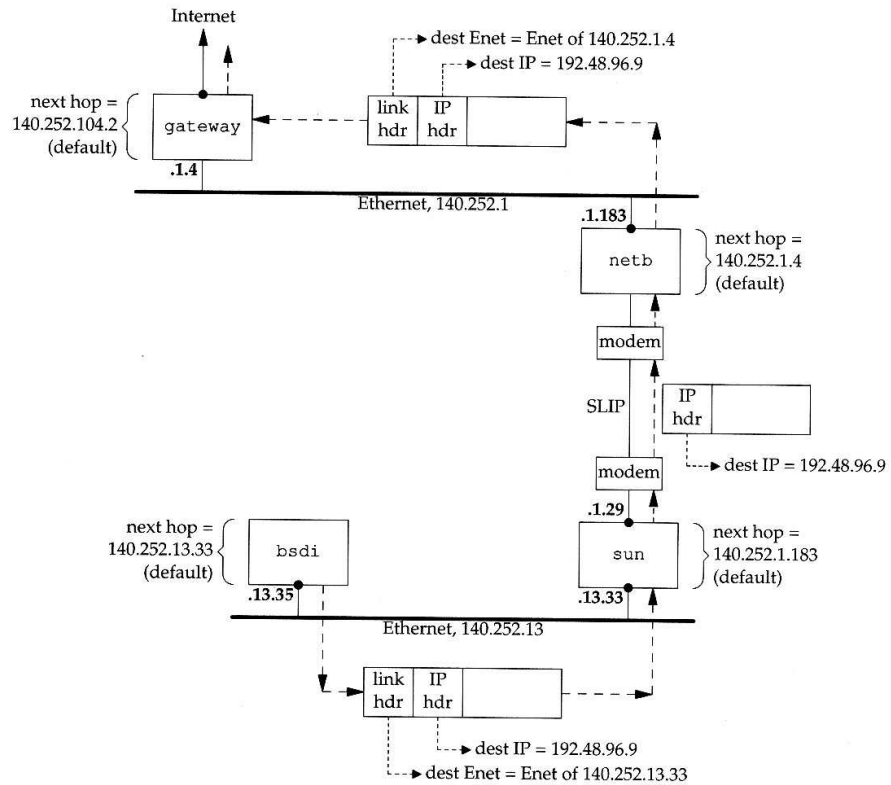
Interface

Nome da interface local

Exemplo 1

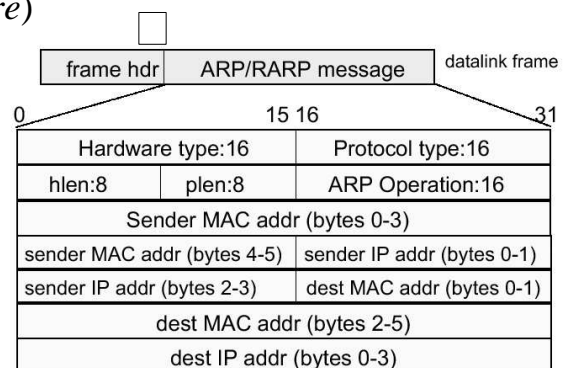


Exemplo 2



ARP – Address Resolution Protocol

- ◆ Host TCP/IP → endereço IP
- ◆ Carta de rede → endereço físico (*hardware*)
 - » Em LANs – endereço MAC
- ◆ ARP
 - » Mapeamento dinâmico
 - » Endereço IP ← → endereço de *hardware*



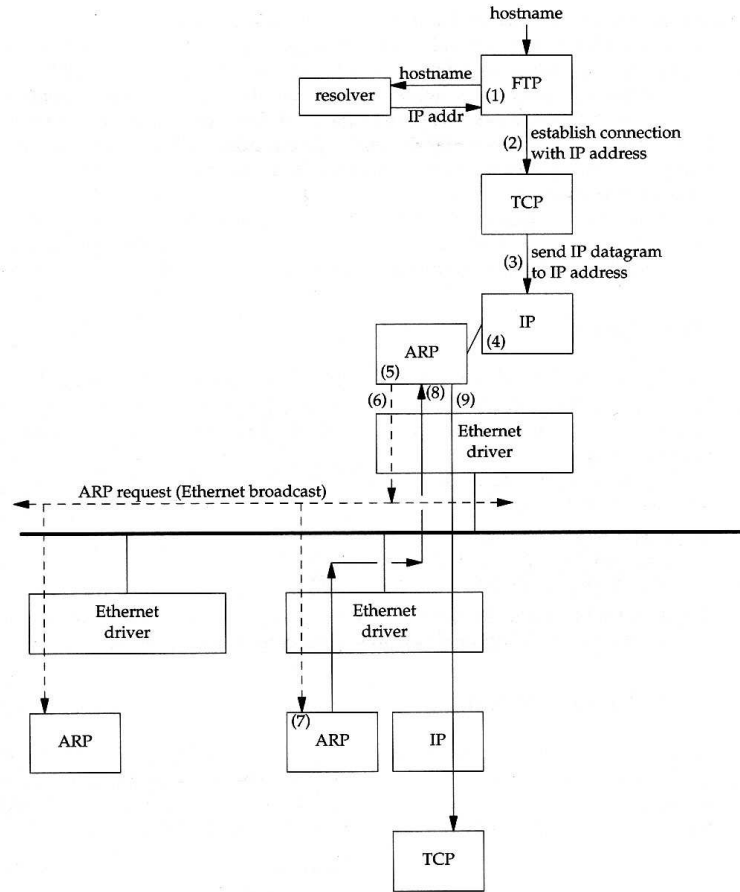
32-bit Internet address



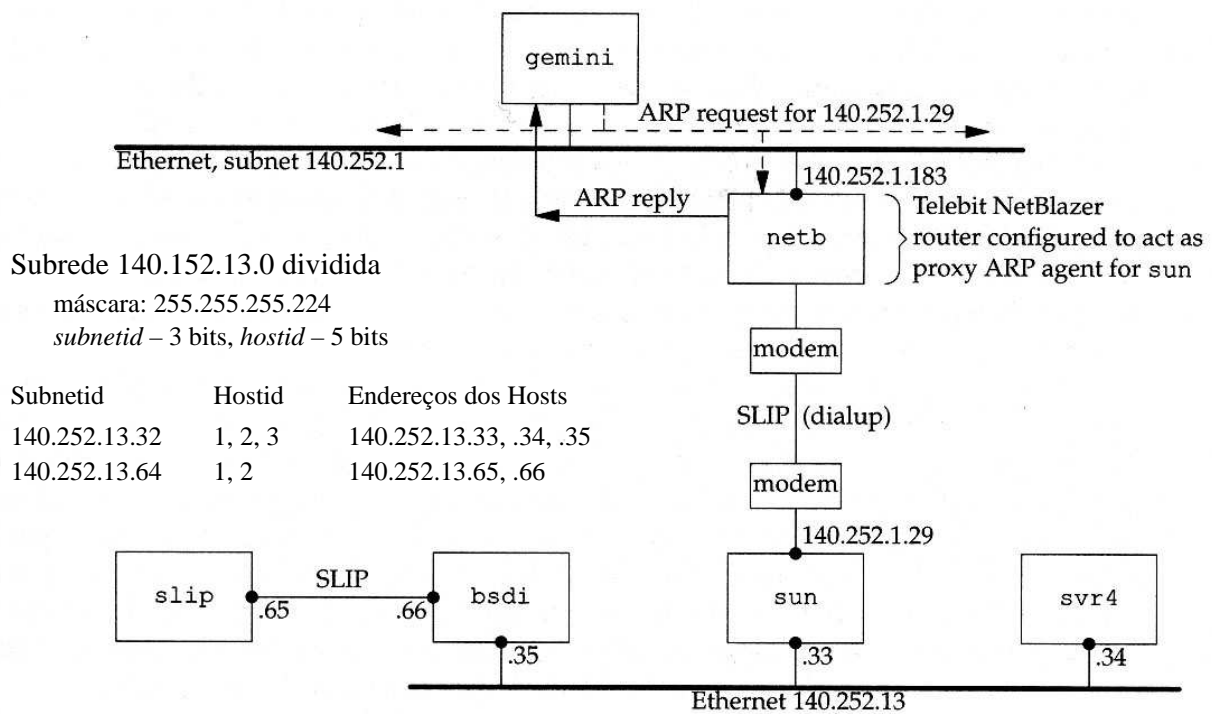
48-bit Ethernet address

- hardware type : Ethernet=1 ARCNET=7, localtalk=11
- protocol type : IP=0x800
- hlen : length of hardware address, Ethernet=6 bytes
- plen : length of protocol address, IP=4 bytes
- ARP operation : ARP request = 1, ARP reply = 2
RARP request = 3, RARP reply = 4

Exemplo

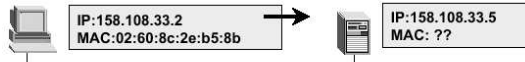


Proxy ARP



ARP Request e ARP Reply

ARP request packet



Sample ARP request Ethernet packet

FF:FF:FF:FF:FF:FF		dest MAC (broadcast)
02:60:8c:2e:b5:8b		source MAC
0x0806		ARP frame type
0x01	0x800	Ethernet / IP
0x06	0x04	MAC=6/ IP=4 /request
02:60:8c:2e:b5:8b		source MAC
158.108.33.2		source IP
00:00:00:00:00:00		dest MAC (unknown)
158.108.33.5		dest IP
checksum		Ethernet checksum

ARP reply packet



Sample ARP reply Ethernet packet

02:60:8c:2e:b5:8b		dest MAC (unicast)
00:00:e8:15:cc:0c		source MAC
0x0806		ARP frame type
0x01	0x800	Ethernet / IP
0x06	0x04	MAC=6/ IP=4 /reply
00:00:e8:15:cc:0c		source MAC
158.108.33.5		source IP
02:60:8c:2e:b5:8b		dest MAC
158.108.33.2		dest IP
checksum		Ethernet checksum

ifconfig

```
sun% /usr/etc/ifconfig -a
```

```
le0 : flags=63<UP, BROADCAST, NOTRAILERS, RUNNING>
inet 140.252.13.33 netmask fffffffe0 broadcast 140.252.13.63

slo : flags=105KUP, POINTOPOINT, RUNNING, LINKO>
inet 140.252.1.29 -> 140.252.1.183 netmask fffffff0

lo0: flags=49<UP, LOOPBACK, RUNNING>
inet 127.0.0.1 netmask ff000000
```

```
bsd% /sbin/ifconfig we0
```

```
we0: flags=863<UP, BROADCAST, NOTRAILERS, RUNNING, SIMPLEX>
inet 140.252.13.35 netmask fffffffe0 broadcast 140.252.13.63
```

```
bsd% /sbin/ifconfig slo
```

```
slo : flags=1011<UP, POINTOPOINT, LINKO
inet 140.252.13.66 -> 140.252.13.65 netmask fffffffe0
```

```
slip% /sbin/ifconfig slo
```

```
slo : flags=1011<UP, POINTOPOINT, LINKO
inet 140.252.13.65 --> 140.252.13.66 netmask fffffffe0
```

```
svr4% /usr/sbin/ifconfig emd0
```

```
emd0: flags=23<UP, BROADCAST, NOTRAILERS>
inet 140.252.13.34 netmask fffffffe0 broadcast 140.252.13.63
```


netstat

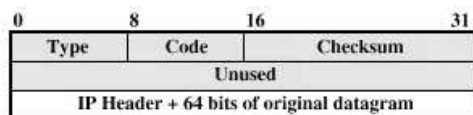
```
sun% netstat -in
```

Name	Mtu	Net/Dest	Address	lpkts	lerrs	Opkts	Oerrs	Collis	Queue
le0	1500	140.252.13.32	140.252.13.33	67719	0	92133	0	1	0
slo	552	140.252.1.183	140.252.1.29	48035	0	54963	0	0	0
lo0	1536	127.0.0.0	127.0.0.1	15548	0	15548	0	0	0

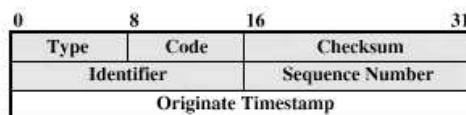
ICMP

- ◆ ICMP → Internet Control Message Protocol (RFC 792)
- ◆ Transferência de mensagens de controlo
 - » *Routers / host* ↔ *host*
 - » Indicam problemas
- ◆ Mensagens ICMP são encapsuladas em datagramas IP (não fiável)

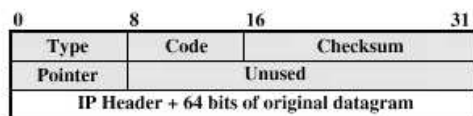
ICMP – Tipos de Mensagem



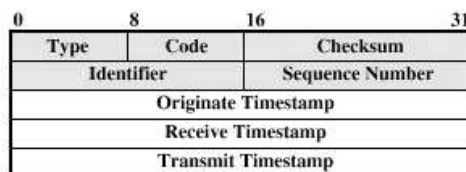
(a) Destination Unreachable; Time Exceeded; Source Quench



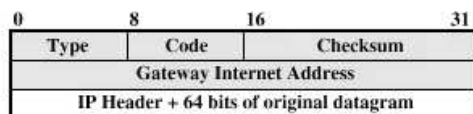
(e) Timestamp



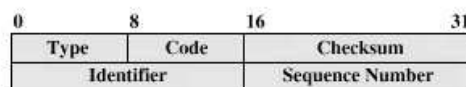
(b) Parameter Problem



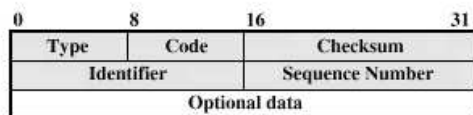
(f) Timestamp Reply



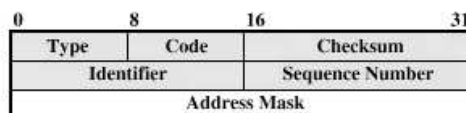
(c) Redirect



(g) Address Mask Request



(d) Echo, Echo Reply



(h) Address Mask Reply

Ping

◆ ICMP Echo / Echo Reply

◆ Echo

- » Identificador
 - Pid do cliente
- » Sequência
 - começa em 0
- » Dados
 - tempo de envio

◆ Echo Reply

- » Enviado por servidor
ping (kernel)
- » Ecoa tudo

```
sun% ping gemini
```

```

PING gemini: 56 data bytes
64 bytes from gemini (140.252.1.11): icmp_seq=0. time=373. ms
64 bytes from gemini (140.252.1.11): icmp_seq=1. time=360. ms
64 bytes from gemini (140.252.1.11): icmp_seq=2. time=340. ms
64 bytes from gemini (140.252.1.11): icmp_seq=3. time=320. ms
64 bytes from gemini (140.252.1.11): icmp_seq=4. time=330. ms
64 bytes from gemini (140.252.1.11): icmp seq=5. time=310. ms
64 bytes from gemini (140.252.1.11): icmp seq=6. time==290. ms
64 bytes from gemini (140.252.1.11): icmp_seq=7. time=300. ms
64 bytes from gemini (140.252.1.11): icmp_seq=8. time=280. ms
64 bytes from gemini (140.252.1.11): icmp_seq=9. time=290. ms
64 bytes from gemini (140.252.1.11): icmp_seq=10. time=300. ms
64 bytes from gemini (140.252.1.11): icmp_seq=11. time=280. ms
--gemini PING Statistics--
12 packets transmitted, 12 packets received, 0% packet loss
round-trip (ms) min/avg/max = 280/314/373

```

TraceRoute

- ◆ Permite detectar rotas entre máquinas
- ◆ Usa datagramas IP e mensagens de erro ICMP
 - » *Traceroute* começa por enviar datagrama para destino com
 - TTL = 1 e porta UDP inexistente
 - » O primeiro *router*
 - decrementa valor de TTL → elimina datagrama
 - envia mensagem de erro ICMP (*time exceed*) para origem
 - *Traceroute* obtém a sua identificação
 - » *Traceroute* envia novo datagrama para destino
 - com TTL = 2 e porta UDP inexistente
 - » O segundo *router* repete o processo e assim sucessivamente
 - » *Traceroute* envia datagrama com TTL até à máquina de destino
 - » A máquina de destino envia
 - mensagem de erro ICMP – *port unreachable*
 - *Traceroute* obtém a sua identificação

TraceRoute – Exemplo

```
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
```

```
slip% traceroute svr4
```

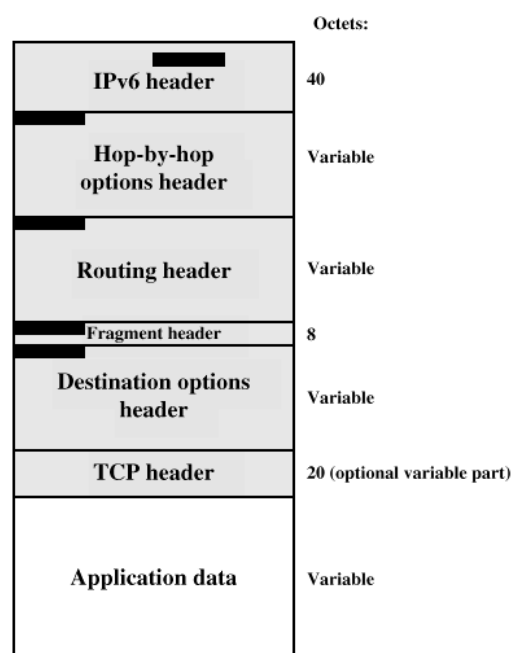
```
traceroute to svr4 (140.252.13.34), 30 hops max, 40 byte packets
1 bsd1 (140.252.13.66) 110 ms 110 ms 110 ms
2 svr4 (140.252.13.34) 110 ms 120 ms 110 ms
```

Versões IP – (IPv4 e IPv6)

- ◆ Versões
 - » IP v1-v3 → fora de uso
 - » IP v4 → versão em uso
 - » IP v5 → orientado às ligações
 - » IP v6 → substituto do IP v4
- ◆ Limitações do IPv4
 - » Endereçamento
 - Dois níveis de endereçamento (rede, *host*) desperdiça endereços
 - Endereços usados mesmo quando rede não ligada à Internet
 - Crescimento do número de redes e da Internet
 - Endereço único por computador
 - » Requisitos de novos tipos de serviços
- ◆ Melhorias do IPv6
 - » Flexibilidade de endereçamento
 - Espaço de endereçamento aumenta (128 bits)
 - *Multicast* - mais versátil e escalável
 - » Reserva de recursos
 - Substituição do *Type of Service*
 - Introdução do conceito de fluxo de pacotes
 - Fluxos com tratamento especial (QoS), e.g., Vídeo em tempo real

IPv6 – Protocolo

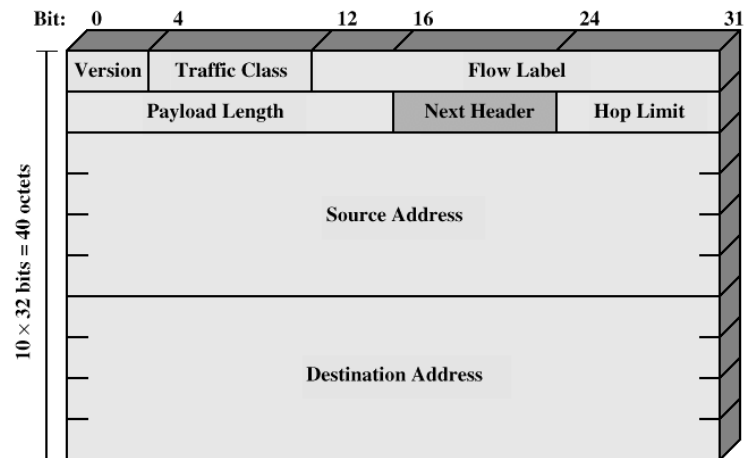
- ◆ Cabeçalho IPv6
- ◆ Cabeçalhos secundários (de extensão)
 - » *Hop-by-Hop Options*
 - Processado e alterado nos *routers*
 - » *Routing*
 - Semelhante ao *Source Routing* do IPv4
 - » *Fragment*
 - » *Destination options*
 - Para nó de destino
 - » TCP



■ =
Next Header field

Cabeçalho IPv6

- ◆ *Version*
 - » 6
- ◆ *Traffic Class*
 - » Classes / prioridades de pacote
 - » Em desenvolvimento
- ◆ *Flow Label*
 - » Suporte de QoS
- ◆ *Payload length*
 - » Cabeçalhos secundários + dados
- ◆ *Next Header*
 - » Tipo do próximo cabeçalho
(Extensão ou nível superior)
- ◆ *Source Address*
- ◆ *Destination Address*

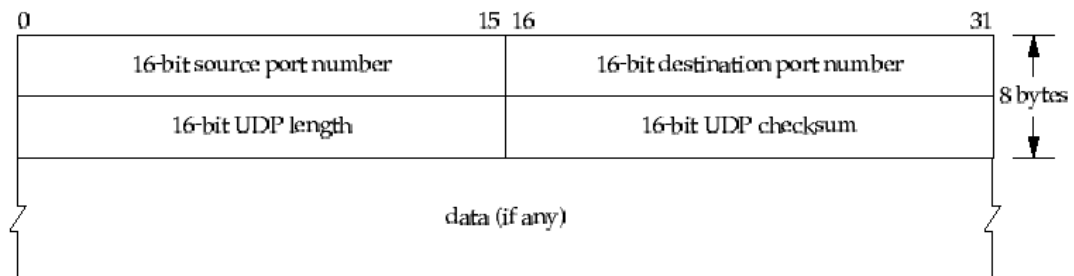


IPv6 – Endereços

- ◆ 128 bits
- ◆ Atribuídos a interfaces
- ◆ Uma interface pode ter múltiplos endereços
- ◆ Tipos de endereços
 - » *Unicast*
 - uma interface
 - » *Anycast*
 - Conjunto de interfaces
 - Pacote entregue a qualquer interface (a mais próxima)
 - » *Multicast*
 - Conjunto de interfaces
 - Pacote entregue a todas as interfaces

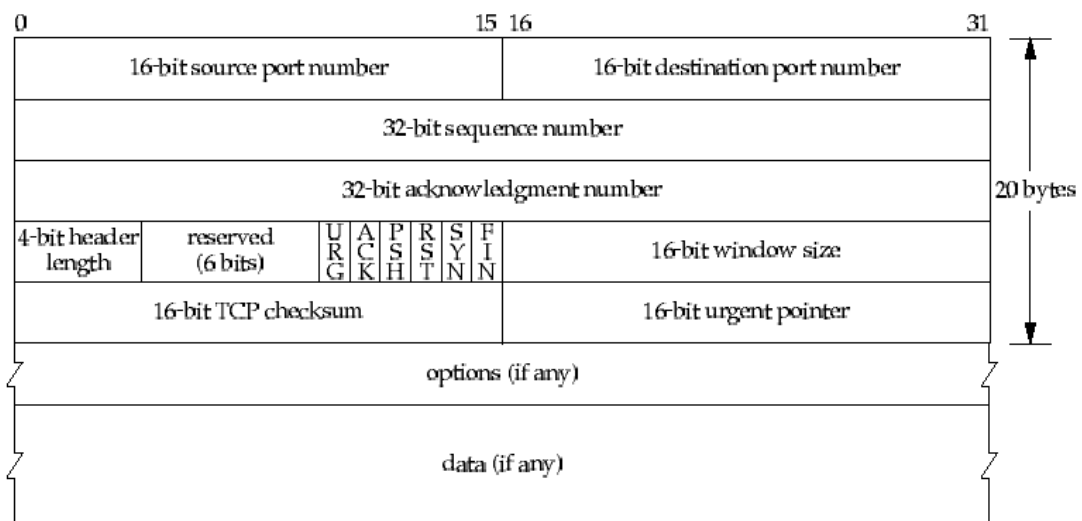
Cabeçalho UDP

UDP Header



Cabeçalho TCP

TCP Header



Bibliografía

» *TCP/IP Illustrated – The Protocols*, Richard W. Stevens

– Capítulos: 1, 2, 3, 4, 6, 7, 8