

**UFPE – Centro de Informática – Infraestrutura de Comunicação**  
**Prova Final – 2020-3**

NOTA :

Nome do(a) aluno(a): Gabriel Nogueira Leite  
Número de Matrícula/CPF: 398.068.608-61

---

**1ª Questão – 4 Pontos**

Considere um datagrama IPv4 com 20 bytes de cabeçalho, tamanho total igual a 3000 bytes e com ID igual a 210. Esse datagrama deverá passar por um enlace cujo MTU é de 500 bytes.

a) Quantos fragmentos serão gerados? Justifique sua resposta.

**Resposta:** Para sabermos quantos fragmentos serão gerados precisamos olhar a relação entre o tamanho do datagrama (2980 bytes de dados + 20 bytes de cabeçalho) e o MTU (*Maximum Transmission Unit*), com isso iremos determinar quantos fragmentos serão gerados. Podemos enxergar melhor olhando a tabela abaixo.

Nº do fragmento	Bytes de dados	Bytes de cabeçalho	Flag	Offset	ID
1	480	20	1	0	210
2	480	20	1	$0 + (\text{Bytes de dados} / 8) = 60$	210
3	480	20	1	$60 + (\text{Bytes de dados} / 8) = 120$	210
4	480	20	1	$120 + (\text{Bytes de dados} / 8) = 180$	210
5	480	20	1	$180 + (\text{Bytes de dados} / 8) = 240$	210
6	480	20	1	$240 + (\text{Bytes de dados} / 8) = 300$	210
7	80	20	0	$300 + (\text{Bytes de dados} / 8) = 360$	210

Logo podemos observar que foram gerados **7 fragmentos**, sendo:

$6 * 480 \text{ (Bytes de dados)} = 2880$

$2980 \text{ (carga útil IP)} - 2880 = 100$

ou podemos recorrer a seguinte equação mostrada durante as aulas:

$$n^{\circ} \text{ de fragmentos} = \text{função teto} \left( \frac{\text{tamanho do datagrama} - \text{tamanho do cabeçalho}}{\text{MTU de saída} - \text{tamanho do cabeçalho}} \right)$$

$$n^{\circ} \text{ de fragmentos} = \text{função teto} \left( \frac{3000 \text{ bytes} - 20 \text{ bytes}}{500 \text{ bytes} - 20 \text{ bytes}} \right) = 7 \text{ fragmentos}$$

b) Qual a funcionalidade dos campos **length**, **ID**, **fragflag** e **offset** ?

**Resposta:** Esses campos são importantes no processo de fragmentação, suas funções são as seguintes:

**Length:** O campo length informa o tamanho em *bytes*, temos o *header length* que é o tamanho do cabeçalho, temos o *length* da carga útil IP e temos o *total length* que é igual o *length* da carga útil IP mais o *header length*.

**ID:** É o número de identificação do datagrama original e serve para identificar e ajudar no processo de remontagem do datagrama.

**Fragflag:** Indica se há mais fragmentos a serem recebidos. Quando o *fragflag* é 1, significa que há mais fragmentos a serem recebidos, quando é 0, significa que não há mais fragmentos a serem recebidos.

**Offset:** Traduzindo para o português, Deslocamento, esse campo indica a localização exata do fragmento no datagrama IP original.

c) Qual será o valor dos campos **length**, **ID**, **fragflag** e **offset** em cada um dos fragmentos gerados? Justifique suas respostas.

**Resposta:** Podemos aproveitar a tabela feita na letra a.

Com isso podemos ver que o valor dos campos em cada um dos fragmentos é o seguinte:

**Fragmento 1:**

- *Length* da carga útil IP = 480 bytes
- *Length* total = 500 bytes
- *Header length* = 20 bytes
- *ID* = 210
- *Fragflag* = 1
- *Offset* = 0

**Fragmento 2:**

- *Length* da carga útil IP = 480 bytes
- *Length* total = 500 bytes
- *Header length* = 20 bytes
- *ID* = 210
- *Fragflag* = 1
- *Offset* = 60

**Fragmento 3:**

- *Length* da carga útil IP = 480 bytes
- *Length* total = 500 bytes
- *Header length* = 20 bytes
- *ID* = 210
- *Fragflag* = 1
- *Offset* = 120

**Fragmento 4:**

- *Length* da carga útil IP = 480 bytes
- *Length* total = 500 bytes
- *Header length* = 20 bytes
- *ID* = 210
- *Fragflag* = 1
- *Offset* = 180

**Fragmento 5:**

- *Length* da carga útil IP = 480 bytes
- *Length* total = 500 bytes
- *Header length* = 20 bytes
- *ID* = 210
- *Fragflag* = 1
- *Offset* = 240

**Fragmento 6:**

- *Length* da carga útil IP = 480 bytes
- *Length* total = 500 bytes
- *Header length* = 20 bytes
- *ID* = 210
- *Fragflag* = 1
- *Offset* = 300

**Fragmento 7:**

- *Length* da carga útil IP = 80 bytes
- *Length* total = 100 bytes
- *Header length* = 20 bytes
- *ID* = 210
- *Fragflag* = 0
- *Offset* = 360

## 2ª Questão – 2 Pontos

Explique os mecanismos criados no nível de rede para a transição do IPv4 para o IPv6.

**Resposta:** Para enfrentar a transição do IPv4 para o IPv6 surgiram algumas ideias, como a abordagem de pilha dupla, implementação de túnel. A pilha dupla descreve a operação paralela de IPv4 e IPv6. Como é praticamente impossível mudar diretamente do IPv4 para o IPv6, a "estratégia de transição" estipula que todos os nós da rede podem suportar IPv4 e IPv6. Migrar para pilha dupla é mais fácil do que mudar todas as máquinas do mundo para o IPv6 de uma vez só, muitos sistemas operacionais podem lidar com pilhas duplas simultaneamente, ou seja, IPv4 e IPv6.

Primeiro veremos o tunelamento IPv4 para IPv6 e IPv6 para IPv4. Desejamos transmitir pacotes IPv4 em uma rede IPv6, para isso, precisamos de um cliente e servidor IPv4 e dois roteadores para operar com pilhas de protocolo duplo. Quando o cliente usa um roteador de pilha dupla, os pacotes IPv4 são empacotados em pacotes IPv6, transmitidos pela rede IPv6 e descompactados novamente no último roteador de pilha dupla. A partir daí, os pacotes IPv4 são encaminhados para o servidor IPv4.

Ao usar o procedimento inverso (IPv6 para IPv4 e IPv4 para IPv6) para encapsulamento, o cliente na rede local deve ter um endereço IPv6, e o servidor na Internet também deve estar acessível através do endereço IPv6. Deve haver dois roteadores ou *gateways* na operação de pilha dupla entre o cliente e o servidor. O primeiro roteador de pilha dupla gera pacotes IPv4 contendo pacotes IPv6. O pacote de dados IPv4 passa pela rede IPv4 para o último roteador de pilha de protocolo duplo, que extrai o pacote de dados IPv6 do pacote de dados IPv4 e o encaminha para um servidor compatível com IPv6.

### 3ª Questão – 2 Pontos

Explique como funciona e para o que serve o NAT (*Network Address Translation*).

**Resposta:** NAT (abreviação do nome inglês *Network Address Translation*) é o processo de conversão de um endereço de protocolo da Internet (*IP*) usado em uma rede “x” para outro endereço IP usado em outra rede “y”. Normalmente, a rede converterá seu endereço de rede local em um ou mais endereços IP globais e converterá o endereço IP global de volta no endereço IP local para pacotes de dados recebidos. Isso garante a segurança porque cada solicitação de saída e entrada deve passar por um processo de tradução, o que, por sua vez, oferece uma oportunidade de qualificar ou autenticar a solicitação ou compará-la com solicitações anteriores. O NAT também pode manter baixo o número de endereços IPv4 globais exigidos pela rede e permitir que a rede se comunique com o resto do mundo em um único endereço IP.

### 4ª Questão – 2 Pontos

Explique como funciona o endereçamento IP no CIDR (*Classless InterDomain Routing*).

**Resposta:** O CIDR é baseado na ideia de máscaras de sub-rede, uma máscara é colocada sobre um endereço IP e cria uma sub-rede (uma rede subordinada à Internet). A máscara de sub-rede sinaliza ao roteador qual parte do endereço *IP* é atribuída aos *hosts* e qual parte determina a rede. Em vez de adicionar uma máscara de sub-rede, podemos integrar uma especificação diretamente no endereço IP usando o roteamento entre domínios sem classes, mas também na forma de sufixos. Isso não apenas encurta a exibição, o CIDR também torna possível criar grandes redes, além de sub-redes. Isso significa que não podemos apenas subdividir uma rede com mais precisão, mas também combinar várias redes. Uma grande rede é importante, por exemplo, quando uma empresa possui vários locais, mas deseja gerenciar todos os computadores na mesma rede. Com uma grande rede, várias redes podem ser combinadas em uma rota, razão pela qual a tecnologia também é chamada de agregação de rota (ou seja, agrupamento de rotas). Isso significa que os pacotes de dados são enviados apenas para um destino, independentemente da localização dos hosts.

Boa Sorte !

Tabela para a questão 1 – cabeçalho do datagrama

OBS: Nesta tabela o campo “*Length*” se refere ao *Length* da carga útil IP. O *Header Length* é igual a 20 *bytes*.

Pacote N°	Length	ID	offset	fragflag
1	480	210	0	1
2	480	210	60	1
3	480	210	120	1
4	480	210	180	1
5	480	210	240	1
6	480	210	300	1
7	80	210	360	0

