

IP
{IP.doc}

1. O valor inicial de TTL (*Time To Live*) no cabeçalho pode ser inferior ao número de saltos (*hops*) na rede, entre originador e destinatário, de um datagrama-IP? Justifique.

R: Não. Se o fôsse, e como ele é decrementado por cada *hop*, ele chegaria a zero (e em consequência seria descartado) antes de atingir o nó destino.

2. [09E2.9] Suponha que um datagrama IPv4 contendo um pacote de transporte é fragmentado e que um dos fragmentos se perde na rede.

2.1. Na sequência desta perda, diga que acções (se as houver) são executadas ao nível de rede no destinatário do datagrama original.

2.2. Diga se o nível de transporte no destinatário será informado do facto (e de que forma).

2.3. Como será corrigida esta perda de fragmento?

R1: a recomposição do datagrama é abortada e fragmentos já recebidos são descartados, quando for excedido o tempo-máximo-de-recomposição;

R2: Não;

R3: No caso do TCP, é feita a retransmissão do pacote em causa, em outro datagrama IPv4. No caso de UDP, a aplicação poderá, ela mesma, provocar essa retransmissão.

(Curiosidade: na especificação original do IP, a RFC791, a recomposição do datagrama será abortada quando for excedido o maior entre 1) o tempo-máximo-de-recomposição (por omissão, 15 seg) após se ter recebido o primeiro fragmento, e 2) o Tempo de vida do segmento que apresentar um maior *tempo de vida* (TTL). Acontece que os nós da Internet *real* utilizam o TTL como número de *hops*, e não como “tempo” - pelo que a decisão de abortar a recomposição considera *apenas* o tempo-máximo-de-recomposição; um valor mínimo realístico para este será da ordem dos 60 segs)

3. [09E3.8] Considere que está envolvido no fabrico de um *router* para datagramas IPv4...

Recorde os campos de um datagrama-IPv4:

Versão	IHL	Tipo de Serviço	Comprimento Total	
Identificação			Flags	Offset
TTL (Tempo de Vida)	Protocolo		Checksum	
Endereço de Origem	Endereço de Destino		Opções	Dados

3.1. Quais os campos do cabeçalho de um datagrama que o *router* deve actualizar antes de o transmitir?

3.2. Eventualmente, em alguns datagramas o *router* necessita acrescentar uma opção de 4-bytes; quais os campos do cabeçalho que ele tem que actualizar?

3.3. Sucede que a primeira versão do *router* não processa o campo TTL (*TimeToLive*). Quais as consequências? Qual é o modo correcto de lidar com o campo TTL?

3.4. Em ordem a poupar nos gastos, decide ignorar a *checksum*: o *router* não a testa, nem a actualiza. E, no laboratório, verifica-se que *os testes até correm bem*. Daí a pergunta: será uma boa decisão?

3.5. Com o intuito de acelerar o processamento do datagrama, decide recolher o endereço de destino nos últimos 4 bytes do cabeçalho. Mas, de vez em quando, o *router* falha na determinação do endereço correcto. Porquê?

3.6. Qual o comprimento *mínimo* do cabeçalho? Se o nível superior passar ao nível IP, para serem enviados, 1 Mbyte de dados, mas cada datagrama não puder exceder 40 bytes de comprimento, quantos bytes, no total, terá o nível IP que enviar?

3.7. Admita que o *router* recebe 5 fragmentos de um datagrama de uma entrada cujo MTU é 60 bytes, e que ele decide fazê-los prosseguir por uma saída cujo MTU é 120. Quantos datagramas, no total, deverá transmitir?

R1: TTL e *checksum*

R2: TTL, *checksum*, comprimento do cabeçalho, comprimento total e opções

R3: Alguns dos datagramas poderão ser retransmitidos indefinidamente. O *router* deveria decrementar de 1 o TTL – e, se o resultado fosse 0, descartá-lo e devolver uma mensagem ICMP “*Time exceeded*”

R4: Não é uma boa ideia. A Internet *real* corrompe mesmos os pacotes! E *routers* de outros fabricantes descartariam os datagramas, pois a *checksum* estaria errada...

R5: Os datagramas podem ter opções no cabeçalho! Pelo que os tais últimos 4 bytes seriam Opções, não endereço! (A solução é utilizar como endereço de destino os bytes 17-20! Ou então somente utilizar os últimos 4 bytes do endereço se o comprimento do cabeçalho for 20...)

R6: 20 byte; 2 Mbyte.

R7: 5, pois o router não procede à recomposição (*re-assemble*) dos datagramas.

4. [04E1] Quais são os campos, do cabeçalho de um datagrama-IP, que são *alterados* pelo procedimento de *fragmentação*? Explique o que acontece com cada um deles.

Recorde os campos de um datagrama-IP:

Versão	IHL	Tipo de Serviço	Comprimento Total	
Identificação			Flags	Offset
TTL (Tempo de Vida)	Protocolo		Checksum	
Endereço de Origem	Endereço de Destino		Opções	Dados

R: *Offset*: reflectindo a posição do fragmento no datagrama *original*

Flags: excepto o último fragmento da segmentação dum datagrama com MF=0, será MF=1

Opções: algumas das opções podem ser copiadas para os fragmentos

IHL: tem que ficar de acordo com o comprimento do Header do fragmento

Comprimento total: tem que ficar de acordo com o comprimento do fragmento

Checksum: tem que ficar de acordo com o novo conteúdo do fragmento

Tempo de Vida: decrementado 1

5. Quais deverão ser os valores da flag **MF** (*More Fragments*) e do *Offset do Fragmento* num datagrama-IP completo, não-fragmentado?

R: MF=0, *Offset* do Fragmento=0

6. [08E3.6] Considere a fragmentação de datagramas no protocolo IP v4.


6.1. Teoricamente, qual o valor máximo de fragmentos que podem resultar de um único datagrama IP?

6.2. Qual o maior datagrama que não pode ser fragmentado?

R1: $\geq 65535 / 8 \Rightarrow 2^{13} = 8192$ fragmentos

(Recorde: são aceites datagramas com comprimento até, em teoria, 65535 octetos; e os dados são repartidos em porções terminando em fronteiras de 8 octetos)

R2: Datagrama com 60 octetos de cabeçalho e 8 octetos de dados (É aceitável a resposta 20 de cabeçalho + 8 de dados)

7.  Suponha que um segmento TCP tem 2048 bytes de dados e 20 bytes de cabeçalho. Este segmento tem que atravessar duas ligações para chegar ao destino. A primeira ligação tem um MTU de 1024 bytes e a segunda um MTU de 512 bytes. Assuma que o cabeçalho de qualquer diagrama IP tem 20 bytes.

7.1. Indique os comprimentos e *offsets* de todos os fragmentos entregues ao nó após a primeira ligação.

7.2. Indique os comprimentos e *offsets* de todos os fragmentos entregues à camada IP do destino

7.3. Como é que o destinatário sabe que já recebeu o último fragmento do datagrama?

Resolução:

Relativamente ao segmento TCP em causa, é passado à camada rede um total de 2068 (=20+2048) bytes.

Procedimentos relativos à *primeira* ligação: sendo MTU=1024 bytes, e havendo que reservar, em cada datagrama-IP, 20 bytes para o cabeçalho, sobram **1004** bytes (=1024-20) para os *dados* - o que é menos que os **2068** bytes do segmento a transmitir. Há, pois, que o fragmentar... As regras a observar são: *tentar encher cada datagrama o mais possível, e operar a fragmentação em fronteiras de 8 bytes*.

Quantos bytes de dados é que cada datagrama irá carregar? Repare-se que $1004/8=125,5$; por conseguinte, o maior múltiplo de 8 que não excede 1004 é $8*125=1000$. Pelo que as regras acima ficarão cumpridas vertendo, no campo *dados* de cada datagrama (exceptuando o último), 125 blocos de 8 bytes.

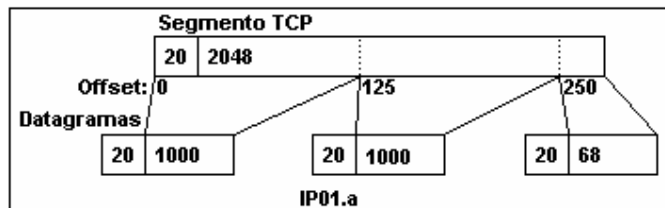
- numeram-se 0, 1, 2,... os blocos de 8 bytes dos dados do segmento; inicializa-se *offset*=0, cfr IP01.a;

- constroi-se um primeiro datagrama, com 125 blocos de 8 bytes do segmento-TCP, contados a partir de *offset*=0; *offset* incrementa de 125; ficam sobrando $2068-1000=1068$ bytes...

- constroi-se um segundo datagrama, com 125 blocos de 8 bytes do segmento-TCP, contados a partir de *offset*=125; *offset* incrementa de 125; ficam sobrando $1068-1000=68$ bytes...

- constroi-se o terceiro, e último datagrama, com os últimos 68 bytes do segmento-TCP, contados a partir de *offset*=250...

Sumarizando, as características dos datagramas despachados para a primeira ligação são:



Cabeçalho	Dados	Offset	MF-Flag	Comprimento do datagrama
20	1000	0	1	1020
20	1000	125	1	1020
20	68	250	0	88

Procedimentos relativos à *segunda* ligação: sendo MTU=512 bytes, e havendo que reservar, em cada datagrama-IP, 20 bytes para o cabeçalho, sobram 492 bytes para os *dados*.

O *primeiro* datagrama recebido da primeira ligação transporta 1000 bytes de dados - o que excede esses 492 bytes. Há, pois, que o fragmentar...

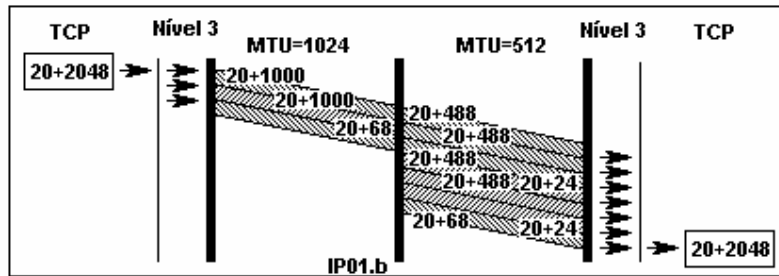
Quantos bytes de dados é que cada fragmento irá carregar? Repare-se que $492/8=61,5$. Pelo que em cada fragmento (exceptuando o último), deverão ser colocados 61 blocos de 8 bytes.

- constroi-se um primeiro fragmento, com 61 blocos de 8 bytes do datagrama, contados a partir de *offset=0*; *offset* incrementa de 61; ficam sobrando $1000-488=512$ bytes...

- constroi-se um segundo fragmento, com 61 blocos de 8 bytes do datagrama, contados a partir de *offset=61*; *offset* incrementa de 61; ficam sobrando $512-488=24$ bytes...

- constroi-se o terceiro fragmento, com os últimos 24 bytes do datagrama, contados a partir de *offset=122*...

Para o segundo datagrama proveniente da primeira ligação, o procedimento é análogo. Sumarizando, tem-se o seguinte, à chegada à camada rede do destino, cfr IP01.b:



Cabeçalho	Dados	Offset	MF-Flag	Comprimento do fragmento
20	488	0	1	508
20	488	61	1	508
20	24	122	1	44
20	488	125	1	508
20	488	186	1	508
20	24	247	1	44
20	68	250	0	88

Nota: em cada fragmentação, são também alterados os campos TTL e *checksum*.

8. © [2007/09] Considere que uma *gateway-IP* recebeu um datagrama-IP com os campos seguintes:

- Id: 314
- TL (*Total Length*)=132
- FO (*Offset do Fragmento*)=29
- MF (*More Fragments*)=0

e que ele deve ser re-enviado, para uma rede cujo $MTU_{Max}=128$.

Em quantos datagramas-IP deve aquele ser segmentado? Quais os valores que aqueles campos deverão passar a deter, em cada um deles?

Nota: Admita que o campo Opções do cabeçalho do datagrama-IP está vazio

R: $IHL=20 \Rightarrow [(128-20) / 8] * 8=104$ octetos por fragmento; $132 = 20+112$

$\Rightarrow 112 / 104 \approx 2$ datagramas-IP, com os seguintes campos:

Id	TL	FO	MF
314	20+104	29	1
314	20+8	42	0

9. [07T2] Considere que um computador *fonte* gerou um datagrama IPv4 em que, no respectivo cabeçalho, *Id*=345 e o campo *Opções* está vazio. No caminho até ao computador *destino*, o datagrama IP sofreu várias fragmentações. Suponha que os nós, quando segmentam os pacotes, agem por forma a *encher o mais possível* os fragmentos resultantes. Suponha também que, a certa altura, já chegaram ao destino três fragmentos, pela ordem seguinte e com os seguintes campos no cabeçalho:

	DG1:	DG2:	DG3:
<i>Id</i>	345	345	345
<i>Total Length</i>	140	50	140
<i>Fragment Offset</i>	15	60	30
Flag <i>More Fragments</i>	1	0	1

- 9.1. Com base na informação contida nos fragmentos já recebidos, qual o número de octetos de dados do datagrama original na fonte?
 9.2. Qual o valor mais elevado de MTU (*Maximum Transfer Unit*) que a Rede mais restritiva (visitada por aqueles três fragmentos) pode ter?
 9.3. Qual o *Fragment Offset* do fragmento que contém o 400º octeto do datagrama original?

R1: $(60 \cdot 8) + (50 - 20) = 510$ bytes (Nota: como o campo *Opções* está vazio, o *IP-Header* tem apenas 20 bytes)
 R2: $MTU_{Max} = 140 + (8 - 1) = 147$ bytes (Nota: a segmentação faz-se em fronteiras de 8 bytes)
 R3: $FO = \text{int} (400 / (15 \cdot 8)) \cdot 15 = 45$ (Nota: cada fragmento comporta $140 - 20 = 120 = 15 \cdot 8$ bytes)

10. [07E2.5] Um emissor envia um datagrama com 900 bytes de dados e com o campo *Opções* vazio. No caminho até ao destino, o datagrama original foi fragmentado. Suponha que todas as redes atravessadas pelos datagramas (fragmentos) possuem o mesmo MTU (*Maximum Transfer Unit*). Suponha também que, neste cenário, todos os *routers* que decidem fragmentar datagramas escolhem sempre o máximo tamanho possível para os fragmentos. Considere que chegaram ao receptor os seguintes datagramas, pela ordem indicada e com o seguinte conteúdo:

	DG1:	DG2:
<i>Id</i>	345	345
<i>Total Length (TL)</i>	180	180
<i>Fragment Offset (FO)</i>	80	20
Flag <i>More Fragments (MF)</i>	1	1

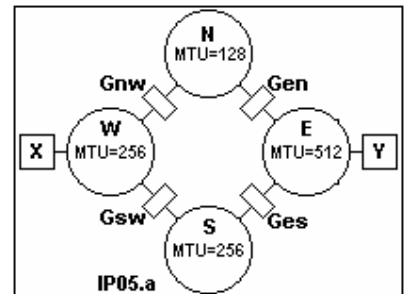
Quais os campos FO, MF e TL dos fragmentos que ainda não chegaram?

R: O número de bytes de dados em cada fragmento é de $180 - 20 = 160$. Então, a recomposição do datagrama será:

		DG2			DG1	
TL-20:	160	160	160	160	160	$100 = 900 - 5 \cdot 160$
FO:	0	20	40	60	80	100

→DG3: FO=0, MF=1, TL=180; DG4: {40,1,180}; DG5: {60,1,180}; DG6: {100,0,120}

11. [08T2.3] Considere que entre dois computadores, X e Y, existem várias Redes, com a topologia esquematizada na figura ao lado, onde se indicam os respectivos MTU (*Maximum Transfer Unit*). X gera um datagrama IPv4 com 512 bytes de Dados. Esse datagrama é sujeito a uma primeira fragmentação na rede W. O primeiro fragmento é encaminhado pela Gateway Gnw e os restantes são encaminhados para a Gateway Gsw. Assuma que se *enche o mais possível* os fragmentos resultantes. Especifique os campos *TotalLength*, *FragmentOffset* e *MoreFragment* dos fragmentos que chegarão a Y.

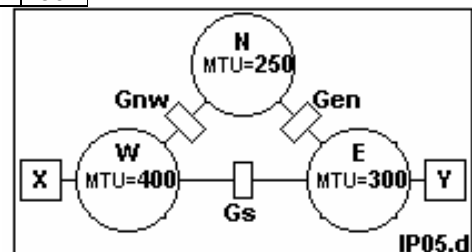


[Nota: Suponha que não haverá perdas nem duplicados, nem os datagramas sofrem *loops* no trânsito entre X e Y].

R: (Reparo: 256 cobre $20 + 29 \cdot 8$, 128 cobre $20 + 13 \cdot 8$)

<i>TotalLength</i>	124	124	44	252	68
<i>FragmentOffset</i>	0	13	26	29	58

12. [09T2.2] Considere que entre dois computadores, X e Y, existem várias Redes, com a topologia esquematizada na figura IP05.d, onde se indicam os respectivos MTU (*Maximum Transfer Unit*). X gera um datagrama IPv4 com 1024 bytes de Dados (e em cujo



cabeçalho o campo Opções está vazio). Os fragmentos resultantes de uma primeira fragmentação em X são encaminhados pelas Gateways **Gs** ou **Gnw** (e **Gen**). Ao destino Y, chegam no total cinco fragmentos. Na tabela abaixo, resumiram-se os dados (*TotalLength*, *FragmentOffset* e *MoreFragment*) já recolhidos desses datagramas. Preencha as quadrículas ainda vazias. Assuma que, no processo de fragmentação, se enche o mais possível os fragmentos resultantes.

[Nota: Suponha que não há perdas nem duplicados, nem os datagramas sofrem *loops* no trânsito entre X e Y].

<i>TotalLength</i>	300		292		172
<i>FragmentOffset</i>	0				
<i>MoreFragment</i>			0		

R: A fragmentação em X origina três datagramas, { D_{x1} , D_{x2} , D_{x3} }, com, sucessivamente, 376 ($= (400-20)/8*8$), 376 e 272 ($= 1024 - 2*376$) bytes de dados. A fragmentação de D_{x1} em **Gs** origina dois datagramas, com 280 ($= (300-20)/8*8$) e 96 bytes ($= 376-280$) de dados: são o primeiro e segundo (ou quarto) datagramas *recebidos*; os *FragmentOffset* são 0 e $280/8=35$. A fragmentação de D_{x2} em **Gn** origina dois datagramas, com 224 ($= (250-20)/8*8$) e 152 bytes ($= 376-224$) de dados são o quarto (ou segundo) e quinto datagramas *recebidos*. Os respectivos *FragmentOffset* são 47 ($= 376/8$) e 75 ($= 47+28$). D_{x3} detém *MoreFragment=0*; é o terceiro datagrama recebido, o seu *FragmentOffset* será $2*376/8=94$.

1ª Fragmentação	1024 = 376 + 376 + 272				
2ª Fragmentação	$D_{x1}: 376=280+96$		$D_{x3}: 272$	$D_{x2}: 376=224+152$	
	6			2	
<i>TotalLength</i>	300	116	292	244	172
<i>FragmentOffset</i>	0	35	94	47	75
<i>MoreFragment</i>	1	1	0	1	1

(Repare-se: O primeiro datagrama (cujo *FragmentOffset=0*) e o terceiro (cujo *MoreFragment=0*) contém respectivamente o primeiro e último bytes de dados do datagrama original. Pois que esses datagramas têm comprimentos que excedem o MTU de N ($=256$), deduz-se que *esses bytes não podem ter transitado por N*: o seu trajecto foi $X \rightarrow W \rightarrow E \rightarrow Y$. Isto é: D_{x1} e D_{x3} foram, ambos, encaminhados por **Gs**. E, quanto a D_{x2} , a sua fragmentação origina um datagrama com 152 bytes de dados – valor distinto daqueles {280 e 96} associados à fragmentação de D_{x1} ; i.e.: a Gateway por que transitou não foi aquela, **Gs**, por onde viajou D_{x1} , mas **Gn**.

O número de bytes de dados é o valor do *TotalLength* decrescido de 20, o comprimento do *header*)

13. Explique os conceitos, e compare as respectivas vantagens e desvantagens, de fragmentação *intranet* versus fragmentação *internet*.