

Fundação CECIERJ - Vice Presidência de Educação Superior a Distância
Curso de Tecnologia em Sistemas de Computação
Disciplina: Redes de Computadores II
Gabarito da AP1 - 2º semestre de 2011

1ª questão (2.0 pontos)

Considere a seguinte faixa de endereços IP: 118.130.10.160/27. Responda às perguntas abaixo.

1. (0.5 pontos) Quais dos seguintes endereços IPs pertence a esta faixa de endereços IP: 118.130.10.140, 118.130.10.200, 118.130.10.180.

Resposta: A faixa de endereços IPs delimitada por 118.130.10.160/27 possui os três primeiros bits do último octeto, ou seja, os bits 25, 26 e 27, com os valores 1, 0, 1, respectivamente, dando origem ao valor 160 no último octeto. Desta forma, apenas endereços que tenham estes valores em seu último octeto pertencem a esta faixa de endereços. Destar forma, o endereço 118.130.10.140 não pertence a esta faixa (bits 25=1, 26=0, 27=0). O endereço 118.130.10.200 também não pertence (bits 25=1, 26=1, 27=0). Por fim, o endereço 118.130.10.180 pertence a esta faixa (bits 25=1, 26=0, 27=1).

2. (0.7 pontos) Determine a faixa de endereços válidos para o último octeto. Ou seja, qual é a faixa de endereço IPs válida para a rede acima.

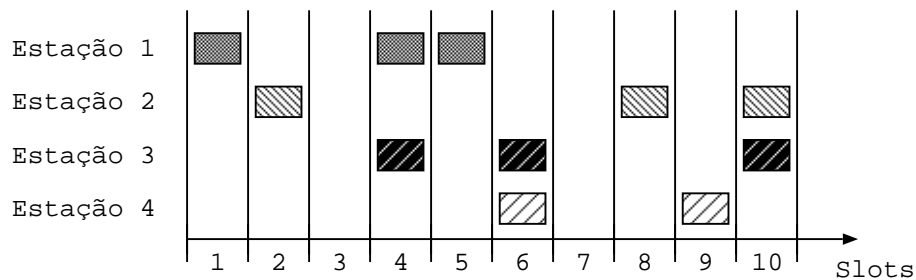
Resposta: A faixa de endereços válidos para o último octeto devem começar com os bits 25=1, 26=0 e 27=1. Os outros 5 bits podem variar a vontade, indo de zero (todos os bits em zero) ao seu valor mais alto (todos os bits em 1). Isto dá origem aos valores 160 (todos os 5 bits em zero) e o valor 160 + 31 (todos os 5 bits em 1) que é igual a 191. Ou seja, o último octeto desta faixa de endereço pode variar de 160 até 191.

3. (0.8 pontos) Suponha agora que quatro subredes serão criadas utilizando a faixa de endereços acima. Determine os endereços de cada uma das subredes. Explique sua resposta.

Resposta: Se quatro subredes serão criadas, então precisaremos de dois bits para endereçar cada uma destas subredes. Iremos usar os bits 28 e 29 para endereçar as subredes. Logo, a nova máscara de rede será /29, pois iremos precisar de 29 bits para endereçar as redes, sobrando apenas 3 bits para endereçar os hosts em cada uma das subredes. A primeira subrede terá os bits 28=0 e 29=0, dando origem ao valor 160 para o último octeto, ou seja, 118.130.10.160/29. A segunda subrede terá os bits 28=0 e 29=1, dando origem ao valor 168 para o último octeto, ou seja, 118.130.10.168/29. A terceira subrede terá os bits 28=1 e 29=0, dando origem ao valor 176 para o último octeto, ou seja, 118.130.10.176/29. A quarta subrede terá os bits 28=1 e 29=1, dando origem ao valor 184 para o último octeto, ou seja, 118.130.10.184/29.

2ª questão (2.0 pontos)

1. (0.5 pontos) Considere o protocolo de acesso aleatório Slotted-ALOHA. Explique porque colisões podem ocorrer e como o protocolo se recupera de uma colisão.



Resposta: Colisões podem ocorrer pois duas ou mais estações podem iniciar uma transmissão em um mesmo slot. Além disso, as estações não podem evitar as colisões pois elas não sabem se alguma outra estação irá ou não iniciar uma transmissão em um determinado slot. O protocolo se recupera de uma colisão retransmitindo o pacote que colidiu. Em particular, um pacote que colidiu será retransmitido por sua estação no próximo slot com probabilidade p . Ou seja, a estação “joga uma moeda” tendenciosa com probabilidade p que irá decidir se o pacote será retransmitido ou não no próximo slot. Este procedimento se repete até que o pacote seja transmitido com sucesso (ou seja, sem colisão).

2. (1.5 pontos) Considere o exemplo ilustrado na figura acima, onde 4 estações utilizam o protocolo Slotted-ALOHA para compartilhar o meio. Considerando os 10 slots ilustrados na figura, responda às perguntas abaixo.

- (a) (0.5 pontos) Para cada slot, indique se houve uma transmissão com sucesso, uma colisão, ou se o slot foi desperdiçado (ficou *idle*). Determine a vazão do canal (ou seja, a fração de slots com sucesso).

Resposta: Vamos usar S=Sucesso, C=Colisão e D=desperdiçado. Temos então a seguinte sequência para os 10 slots ilustrados na figura: S,S,D,C,S,C,D,S,S,C. A vazão do canal é dada pela fração de slots com sucesso, que neste caso é dado por $5/10 = 50\%$.

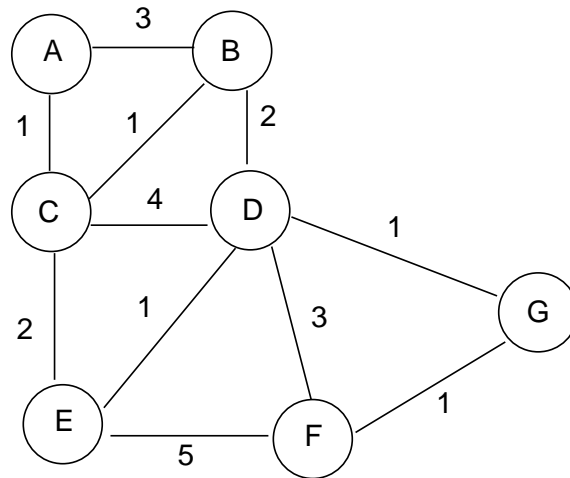
- (b) (0.5 pontos) Quantos pacotes cada uma das quatro estações transmitiu com sucesso? (ex. Estação 1 transmitiu x pacotes com sucesso, estação 2 transmitiu y pacotes com...).

Resposta: A transmissão de uma estação pode corresponder a um novo pacote ou a uma retransmissão de um pacote que colidiu. Assim sendo, a estação 1 transmitiu dois pacotes com sucesso, a estação 2 transmitiu 2 pacotes com sucesso, a estação 3 não transmitiu nenhum pacote com sucesso e a estação 4 transmitiu 1 pacote com sucesso.

- (c) (0.5 pontos) Considere o slot 7. Para cada estação, explique por que a estação não transmitiu neste slot.

Resposta: As estações 1 e 2 não transmitiram porque não tinham nada para transmitir, tendo em vista que suas últimas transmissões foram bem sucedidas (sem colisão). As estações 3 e 4 tinham pacotes para transmitir mas não transmitiram pois o algoritmo de recuperação de colisão aleatoriamente e de forma independente decidiu que a estação não iria tentar a retransmissão naquele slot. Por coincidência, isto ocorreu tanto para a estação 3 quanto para a estação 4.

3ª questão (3.5 pontos)



Suponha a rede da figura acima onde cada enlace está associado com o seu respectivo custo. Considere que o algoritmo usado para fazer o roteamento da rede é o *Link State Routing*.

- (1.0 ponto) Construa uma tabela igual a mostrada em aula que demonstra o funcionamento do algoritmo de Dijkstra (usado pelo *Link State Routing*), de forma iterativa.

Resposta:

O resultado do processo iterativo (algoritmo de Dijkstra) está ilustrado na tabela abaixo:

Passo	N'	d(B),p(B)	d(C),p(C)	d(D),p(D)	d(E),p(E)	d(F),p(F)	d(G),p(G)
1	A	3,A	1,A	∞	∞	∞	∞
2	AC	2,C		5,C	3,C	∞	∞
3	ACB			4,B	3,C	∞	∞
4	ACBE			4,B		8,E	∞
5	ACBED					7,D	5,D
6	ACBEDG					6,G	
7	ACBEDGF						

- (0.5) Construa a tabela de roteamento do nó A a partir da tabela acima.

Resposta:

A tabela de roteamento do nó A é facilmente contruída a partir da tabela acima.

Tabela de roteamento do nó A:

Destino	Enlace de Saída
B	AC
C	AC
D	AC
E	AC
F	AC
G	AC

- Responda as perguntas abaixo sobre o algoritmo *Link State Routing*, justificando cada uma de suas respostas.

- (a) (0.5) O algoritmo é distribuído ou centralizado ? Quais as características que permitiram você classificá-lo como distribuído ou centralizado ?

Resposta:

O algoritmo *Link State Routing* é distribuído pois cada nó calcula, independentemente dos outros, a rota de menor custo para todos os destinos na rede. Não existe uma entidade na rede que efetue este cálculo e distribua para todos os nós.

- (b) (0.5) Cite uma desvantagem desse algoritmo.

Resposta:

As principais desvantagens são: (i) cada nó deve armazenar a topologia da rede; (ii) o algoritmo pode ficar oscilando (sentido horário e sentido anti-horário) em alguns cenários (topologia em anel e tráfego semelhante no sentido horário e anti-horário); (iii) número grande de mensagens sendo transmitidas pela rede (usando flooding) no caso de muitas atualizações de custo em um curto período de tempo.

- (c) (0.5) Explique resumidamente qual o algoritmo usado pelos nós para enviarem as mensagens de atualização de topologia.

Resposta:

O algoritmo usado é do tipo *Flooding*. Considere como exemplo um nó A da rede. As mensagens geradas por A possuem um número de sequência que é incrementado a cada nova mensagem gerada por A. Quando um nó qualquer da rede recebe uma mensagem enviada por A, verifica o número de sequência da mensagem. O nó só a retransmite se a mensagem for nova, ou seja, se ele não tiver recebido anteriormente uma mensagem com o mesmo número de sequência. O nó retransmite a mensagem por todos os enlaces de saída exceto aquele pelo qual ele recebeu a mensagem.

- (d) (0.5) Qual o conteúdo das mensagens de atualização de topologia enviadas pelos nós ?

Resposta:

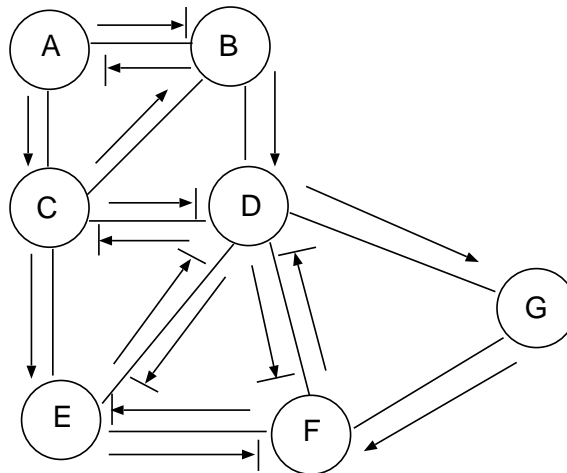
Suponha que o nó A está enviando a mensagem. A mensagem conterá o custo dos enlaces que ligam o nó A até cada um dos seus vizinhos.

4ª questão (2.5 pontos)

Suponha a mesma rede da questão acima onde o nó A envia uma mensagem em broadcast usando (a) o algoritmo de repasse pelo caminho reverso *Reverse Path Forwarding (RPF)* e (b) o algoritmo de *Flooding*. Cada enlace possui um custo associado conforme mostrado na figura.

- (1.0) Construa o diagrama com as mensagens enviadas usando o algoritmo RPF. Use a notação $\longrightarrow |$ para indicar que uma mensagem recebida por um nó não será retransmitida através de um link e a notação \longrightarrow para indicar que a mensagem será retransmitida (mesma notação usada no livro). Nota: Para facilitar a construção do diagrama, construa primeiro a árvore de menor custo para o nó A.

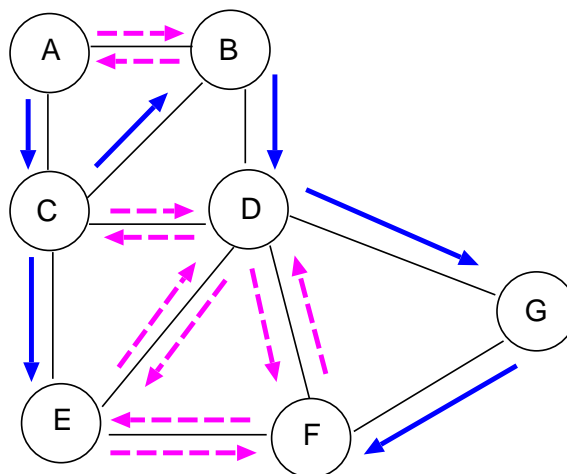
Resposta:



- (1.0) Construa o diagrama com as mensagens enviadas usando o algoritmo de *Flooding*. Use a notação \longrightarrow para indicar que uma mensagem recebida por um nó será retransmitida através de um link.

Resposta:

Na figura abaixo as mensagens com a notação \longrightarrow são aquelas que serão retransmitidas e as mensagens tracejadas são aquelas que não serão retransmitidas.



3. (0.5) Qual dos algoritmos é mais eficiente em termos do número de mensagens redundantes enviadas ? Explique sua resposta.

Resposta:

Os dois algoritmos tem o mesmo número de mensagens redundantes. O RPF só repassa cada mensagem recebida uma única vez. A mensagem só é repassada para os vizinhos caso seja recebida pelo enlace usado para alcançar o nó origem da mensagem. O Flooding também só repassa cada mensagem recebida uma única vez. A mensagem só é repassada para os vizinhos se tiver um número de sequência superior ao da última mensagem recebida do nó origem que gerou a mensagem.