

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 2009

1ª questão (2.0 pontos)

O MTU (*Maximum Transmission Unit*) é uma das características dos enlaces de rede. Responda às perguntas abaixo.

1. (0.5 pontos) Determine quando e por que é necessário fragmentar um datagrama para que o mesmo seja transmitido por um determinado enlace.

Resposta:

Um determinado datagrama precisa ser fragmentado pela camada de redes sempre que o tamanho do mesmo for maior do que o tamanho máximo de quadro permitido pela camada de enlace. É necessário fazer esta fragmentação por causa da limitação do tamanho máximo de bytes em um quadro da camada de enlace, que depende da tecnologia (tipo) do enlace.

2. (0.5 pontos) Por que o MTU de um enlace não deve ser muito pequeno?

Resposta:

Ao fragmentar o datagrama, a camada de rede precisa adicionar o seu cabeçalho em cada um dos fragmentos gerados. Se o MTU for pequeno, então uma proporção não desprezível do quadro será ocupado pelo cabeçalho e restando menos espaço para os dados a serem transmitidos. Isto pode aumentar muito o *overhead*, afetando o desempenho da transmissão.

3. (0.5 pontos) O padrão Ethernet utiliza um MTU de 1500 bytes. Considere a transmissão de um arquivo de 1MB (1MB = 10^6 bytes) por um enlace Ethernet. Considerando o cabeçalho do protocolo IP (que contém 20 bytes), determine quantos quadros Ethernet são necessários para transmissão do arquivo.

Resposta:

Como o cabeçalho do protocolo IP tem 20 bytes e o MTU do Ethernet é de 1500 bytes, sobram $1500 - 20 = 1480$ bytes para os dados em cada quadro. Como o arquivo tem 10^6 bytes, serão necessários $\lceil 10^6 / 1480 \rceil = 676$ quadros (ou fragmentos) para fazer a transmissão.

4. (0.5 pontos) Determine o *overhead* na transmissão do arquivo do item acima. Lembrando que o *overhead* é a razão entre o total de bytes de cabeçalho e total de bytes transmitidos. O *overhead* depende do tamanho do arquivo?

Resposta:

Total de bytes de cabeçalho transmitidos por quadro: 20. Total de bytes de dados transmitidos por quadro: 1480. Total de bytes transmitidos (cabeçalho + dados): 1500, que é o MTU. Logo, o overhead é dado por $20/1500 = 0.013 = 1.3\%$. Não, o overhead não depende do tamanho do arquivo, dado que o arquivo irá ocupar muitos quadros. Assim sendo, o overhead será o mesmo em cada quadro, e por isto será o mesmo para qualquer número de quadros.

2ª questão (2.0 pontos)

Considere os diferentes protocolos de acesso ao meio e responda às perguntas abaixo.

1. (0.5 pontos) Determine sob quais condições o protocolo Aloha é mais eficiente do que o protocolo TDMA. Explique sua resposta.

Resposta:

O protocolo Aloha permite que as estações acessem e transmitam pelo canal no instante em que a informação estiver disponível para transmissão. Isto pode levar a colisões, gerando retransmissões. Entretanto, se a carga (ou seja, a demanda das estações pelo canal) for muito baixa, então colisões serão muito raras. Neste caso, o protocolo Aloha terá um desempenho melhor do que o TDMA. Além disso, se apenas uma estação deseja transmitir, no protocolo Aloha esta estação poderá usar a capacidade máxima do canal, atingindo uma vazão muito maior do que no TDMA, onde o canal é dividido (no tempo) entre todas as estações.

2. (0.5 pontos) Considere os protocolos *slotted* Aloha e o protocolo Aloha puro. Qual dos dois protocolos é mais eficiente? Explique sua resposta.

Resposta:

O protocolo *slotted* Aloha é mais eficiente. Isto ocorre, pois no *slotted* Aloha quando uma colisão ocorre, esta dura exatamente 1 slot de tempo, pois todas as estações acessam o meio e transmitem no início de cada slot. Já no Aloha puro, uma colisão pode durar até dois slots de tempo, pois uma estação pode iniciar sua transmissão quando outra está quase terminando de transmitir, causando uma colisão. Esta colisão irá gerar um desperdício de tempo equivalente a dois slots, pois nenhuma das duas transmissões será aproveitada.

3. (0.5 pontos) Considere o protocolo CSMA. Neste protocolo, as estações escutam o meio antes de transmitir e apenas transmitem se o meio estiver livre de transmissões. É possível ter colisões neste protocolo? Explique sua resposta.

Resposta:

Sim, é possível ter colisões. Estas ocorrem por causa do tempo de propagação no meio. Considere uma estação A que detecta o meio livre e inicia sua transmissão. Os bits transmitidos pela estação A levarão um certo tempo até chegarem a estação B (tempo de propagação). Se a estação B escutar o meio durante este tempo, irá detectá-lo livre e poderá iniciar sua transmissão. Entretanto, logo após iniciar sua transmissão os bits de A irão chegar a B, causando assim uma colisão.

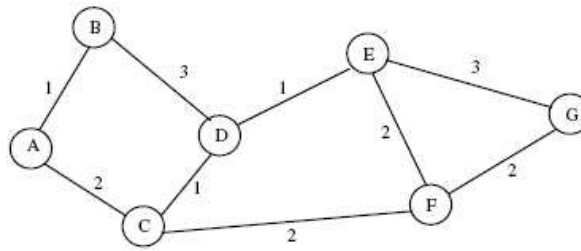
4. (0.5 pontos) Protocolos que detectam colisão em geral abortam a transmissão quando uma colisão é detectada. Por que este procedimento é vantajoso?

Resposta:

Este procedimento, empregado no CSMA-CD por exemplo, é utilizado para aumentar o desempenho do protocolo. Ao abortarem suas respectivas transmissões assim que uma colisão é detectada, as estações reduzem o tempo que o canal fica ocupado com transmissões que não podem ser aproveitadas (por causa da colisão). Assim sendo, o canal fica livre (desocupado) mais rapidamente para outras estações transmitirem, aumentando assim o desempenho do protocolo.

3ª questão (2.8 pontos)

Considere a rede de comunicação da figura abaixo:



- (1 ponto) Suponha que o algoritmo de roteamento implementado é o *Link State* onde a métrica utilizada para o cálculo do menor caminho é o retardo associado a cada link conforme indicado na figura. Construa a tabela de roteamento do nó A da rede indicando os passos usados na sua construção.

Resposta:

Cada nó armazena a topologia da rede e executa o algoritmo de Dijkstra para cálculo do menor caminho até todos os destinos da rede. 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) |
|-------|--------|-----------|-----------|-----------|-----------|-----------|-----------|
| 0 | A | 1,A | 2,A | ∞ | ∞ | ∞ | ∞ |
| 1 | AB | | 2,A | 4,B | ∞ | ∞ | ∞ |
| 2 | ABC | | | 3,C | ∞ | 4,C | ∞ |
| 3 | ABCD | | | | 4,D | 4,C | ∞ |
| 4 | ABCDE | | | | | 4,C | 7,E |
| 5 | ABCDEF | | | | | | 6,F |

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

Tabela de roteamento do nó A:

| Destino | Enlace de Saída |
|---------|-----------------|
| B | AB |
| C | AC |
| D | AC |
| E | AC |
| F | AC |
| G | AC |

2. Suponha agora que todos os nós já computaram as suas tabelas de roteamento baseados nos custos dos enlaces apresentados na figura e que todas as tabelas já convergiram para o menor caminho para todos os destinos na rede.

Durante a construção da tabela de roteamento, os nós trocam mensagens. Considere que os nós continuam a enviar essas mensagens periodicamente para os outros nós da rede, para informar que eles estão operacionais, mesmo que não haja nenhuma alteração no custo dos enlaces.

- (a) (0.8) Considerando que o algoritmo de roteamento implementado é o *Link State* que mensagens seriam recebidas pelo nó D ? Utilize a seguinte notação para as mensagens: M -> H (M-Z,10 ; M-Y,4), o nó M envia mensagem para H informando o custo dos seus enlaces até Z e até Y.

Resposta:

D recebe mensagens de todos os nós:

A -> D (A-B,1; A-C,2)

B -> D (B-A,1; B-D,3)

C -> D (C-A,2; C-D,1)

E -> D (E-D,1; E-F,2; E-G,3)

F -> D (F-C,2; F-E,2; F-G,2)

G -> D (G-E,3; G-F,2)

- (b) (1.0) Considere agora que o algoritmo de roteamento da rede é o *Distance Vector*. Assuma uma versão síncrona do algoritmo. Descreva que mensagens são recebidas pelo nó D. Utilize a seguinte notação para as mensagens: M -> H (Z,10 ; Y,4), o nó M envia mensagem para H informando o custo de 10 até o destino Z e o custo de 4 até o destino Y.

Resposta:

D recebe mensagens de seus vizinhos:

B -> D (A,1; C,3; D,3; E,4; F,5; G,7)

C -> D (A,2; B,3; D,1; E,2; F,2; G,4)

E -> D (A,4; B,4; C,2; D,1; F,2; G,3)

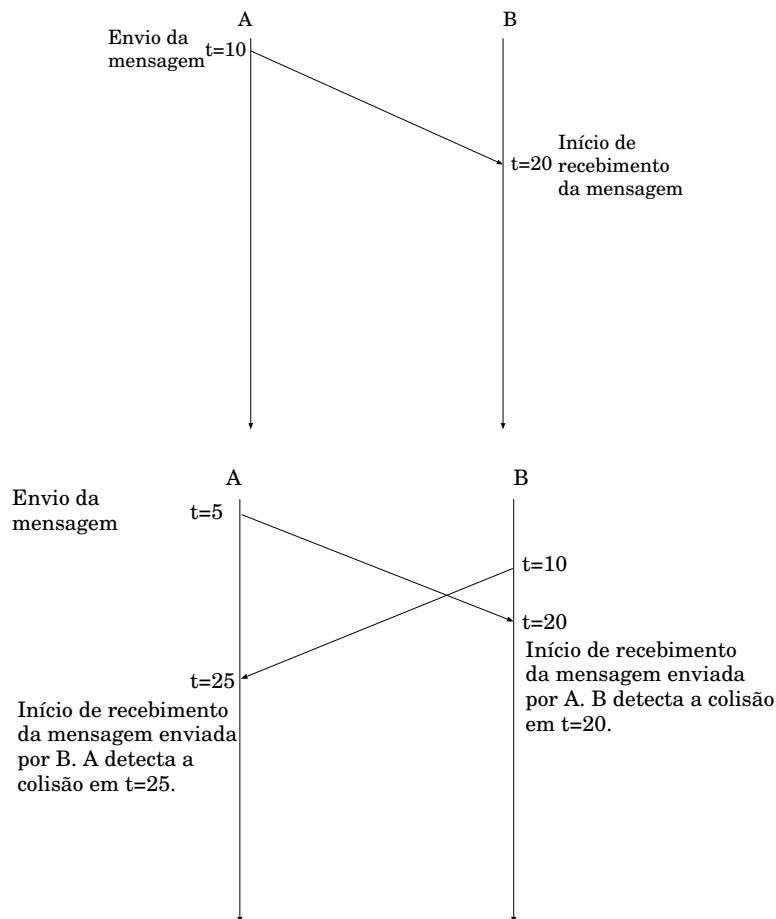
4ª questão (2.2 pontos)

Considere duas estações A e B que estão conectadas a uma rede Ethernet (que usa o protocolo CSMA/CD para controlar o acesso ao meio). Suponha que o tempo que o sinal leva para se propagar da estação A até B (e de B até A) é de 15 unidades de tempo.

Considere que o algoritmo usado para retransmissão após as colisões seja o *binary exponential backoff*, onde a estação espera $K \cdot 20$ unidades de tempo ($K=0,1,2,\dots$) após a colisão antes de escutar o meio de novo. Considere que cada vez que a estação detecta que o meio de transmissão está ocioso, ela inicia imediatamente sua transmissão.

Construa um diagrama temporal conforme o modelo abaixo para responder as perguntas a seguir. No diagrama estão ilustrados os eventos segundo a ordem cronológica de ocorrência. Por exemplo no diagrama abaixo, a estação A começa a transmitir sua mensagem em $t=10$ e B começa a recebê-la em $t=20$ pois o tempo que o sinal leva para se propagar da estação A até B é 10.

1. (0.6) Suponha que a estação A inicia sua transmissão em $t = 5$ e a estação B inicia sua transmissão em $t = 10$ e somente estas estações possuem quadros para transmitir. Em que instante de tempo a primeira colisão será detectada por A e por B ?



Resposta:

- (1.0) Ilustre o seguinte cenário no diagrama que você construiu para o item acima. Após a primeira colisão A obtém $K = 1$ e B obtém $K = 0$. Neste caso haverá outra colisão? Assinale no diagrama os instantes escalonados por A e B para retransmissão da mensagem.
- (0.6) Agora considere que a estação A inicia a sua transmissão em $t=5$ e B esteja pronto para transmitir em $t=25$. Neste caso ocorrerá uma colisão? Se A transmitir uma mensagem de duração igual a 20 unidades de tempo, quando B iniciará a transmissão de sua mensagem?

5ª questão (1.0 ponto)

Explique como funciona o mecanismo NAT e dê um exemplo onde um *host* dentro da sub-rede do NAT abre uma conexão com um *host* na Internet.

Resposta:

O NAT é um mecanismo que permite criar uma rede local privada, com endereços IPs que não são vistos diretamente por hosts na Internet pública. O NAT-box é um computador especial que permite que um host na rede local privada faça uma conexão com um host na rede pública da Internet, traduzindo os endereços. Ao abrir uma conexão TCP com um servidor Web na Internet, por exemplo, um host na rede privada atravessa o NAT-box de forma transparente para ambos (host e servidor Web). O NAT-box anota o endereço e porta de origem do pacote vindo da rede local privada, e muda a porta e o endereço do pacote antes

