#PROCESSO SELETIVO – GRIS 2020

NOME: Leonardo Andrade

TAG REDES

1) Diga quais são as camadas do modelo OSI, em ordem, e dê uma breve descrição de cada uma.

R: O modelo OSI *(Open Systems Interconnection)* é um modelo em camada de funções referência da ISO *(International Organization for Standardization)* para protocolos de comunicação entre diferentes sistemas computacionais numa rede local (Ethernet), garantindo a comunicação *end-to-end*. Ele divide a rede em 7 camadas que garantiram a interconectividade, são elas: **Aplicação**, **Apresentação**, **Sessão**, **Transporte**, **Rede**, **Enlace** e **Física**.

1 - Física: A camada 1 é a camada Física, ela que estabelece a comunicação real no ambiente físico entre os dois dispositivos, responsável pelo cabeamento, características elétricas, eletromagnéticas, ópticas, ou qualquer que seja o tipo. Garante que seja entendido o que é o "1" e o que é "0". Faz o controle básico do meio, mas não trata erros de transmissão. Protocolos: Modem, USB, Bluetooth, 802.11 Wi-Fi, etc.

2 - Enlace (ou Link de dados): Essa camada detecta e corrige os erros acontecidos na camada anterior (Física), também faz controle de fluxo de dados entre os dispositivos. Trabalha com endereço MAC, estabelecendo a comunicação entre os sistemas diretamente conectados. Protocolos: Ethernet, IEEE 802.1Q, PPP, etc.

3 - Rede: Responsável pelo endereçamento dos pacotes, define o caminho, ou seja, a origem e o destino do pacote. Converte endereço IP em endereço MAC. Protocolos: IP (IPv4, IPv6), ARP, NAT, etc.

4 - Transporte: Detecta e corrige erros que ocorreram nas camadas inferiores. Empacota os dados vindos da camada de Sessão e os envia para a camada de Rede. Para os pacotes vindos da camada de Rede, ela remonta o dado e envia para a camada de Sessão. Faz o controle do fluxo e ordenação dos pacotes. Pode ser orientado ou não à conexão. Protocolos: TCP (orientado à conexão), UDP (não-orientado à conexão), etc.

5 - Sessão: Responsável pela comunicação entre dois processos que estão em máquinas diferentes, ela que controla a inicialização, término e reinício da aplicação. Define a forma de transmissão de dados, também os marca para que seja possível a correção de alguns erros. Protocolos: NetBIOS, etc.

6 - Apresentação (também chamada de camada de Tradução): Faz a conversão dos formatos de caractere para que eles possam ser usados na transmissão. Ela também é responsável pela compressão dos dados vindos da camada de Aplicação. É na camada de Apresentação que é possível criptografar os dados. Protocolos: XDR, TLS, etc.

7 - Aplicação: No topo do modelo OSI, identifica quais aplicações devem ser usadas, assim como seus protocolos. Protocolos: HTTP, FTP, SSH, DNS, Torrent, etc.

2) Defina domínio de colisão e domínio de broadcast.

R: Domínio de colisão é uma área lógica onde pacotes podem colidir uns contra os outros, em particular no protocolo Ethernet. Quanto mais colisões ocorrem, menor é a eficiência da rede. O domínio de broadcast serve para que uma máquina dentro de uma rede consiga mandar informação para todas as outras máquinas nessa mesma rede.

3) Determine os domínios de colisão e broadcast na figura abaixo (Segue apenas a resposta):

R:

Domínio broadcast:

>>> (R1+S1+PC1+PC2+PC3), nesse caso temos o roteador que define as rotas que os pacotes deverão seguir e o switch filtra esses pacotes (atua na camada 2), porém ele não conta como um dispositivo de roteamento, então os computadores se comunicam entre si sem necessidade de um equipamento desse tipo. O roteador se comunica com o PC3 sem nenhum dispositivo entre eles.

>>> (R2+H1+PC4+PC5), aqui temos o R2 que define o caminho do tráfego de dados, o H1 que atua na camada 1 replicando os sinais para todas as suas portas não atua como um dispositivo de roteamento, logo os computadores se comunicam sem a necessidade desses dispositivos.

>>> (R1+R2), dois dispositivos de roteamento se comunicam sem nenhum outro entre eles.

Domínio de colisão:

>>> (R1+R2), (R1+PC3), (R1+S1), (S1+PC1), (S1+PC2), (R2+H1), (H1+PC4), (H1+PC5). Considere esse sinal de ”+” como a linha entre eles que simboliza a área lógica de colisão. Temos que essas são os domínios lógicos de colisão. Vale salientar que o H1 concentra os dados enviados por PC4 e PC5 e os envia assim para o R2, o que deixa a rede mais lenta.

4) De acordo com a figura abaixo, suponha que A envia um pacote para B e recebe sua confirmação. Mostre os dados (IP e MAC) ao longo do caminho para que sua transmissão ocorra corretamente. (Segue apenas a resposta)

R:

IP -> Como não estão na mesma rede, o IP –A é levado até o roteador 2 para ser feito o reconhecimento do pacote e do endereço. Logo após isso o R2 envia o pacote e o remetente IP-A para o PCB, este por sua vez envia a confirmação com o IP-B para o R1, que envia para o PCA. Não alterações no IP.

MAC -> O MAC pula de dispositivo em dispositivo. **PCA** – from: MAC-A, to: MAC-S1. **S1** - from: MAC-S1, to: MAC-R1. **R1** – from: MAC-R1, to:MAC-R2. **R2** - from:MAC-R2, to: MAC-B. E na confirmação do pacote é a mesma coisa, porém ao contrário.

5) Resolva novamente a questão anterior supondo que A está por trás de um NAT implementado no roteador R1.

R:

IP => Dessa vez, por conta do NAT, teremos uma alteração no IP-A. Ele sairá com o pacote do PCA e chegará no roteador 1, nesse caso, o NAT transformará o IP-A Local num IP-A Global, digamos assim, um IP que só o PC-A tem e que é reconhecido na Internet mundial. Nesse caso, o R1 envia o IP-A Global para o R2, e este só o envia para o PC-B. No caminho de volta há também o processo feito pelo NAT, mas nesse caso o IP-B local se torna um IP-B global.

MAC => O MAC se propaga da mesma forma que na questão 4).

6) Explique os passos de um handshake TCP de acordo com o RFC793.

R: O handshake é a característica principal do protocolo TCP, é um processo de reconhecimento de início e fim de comunicação entre dois sistemas. Geralmente ele é retratado como se tivesse *three-ways* (três etapas, cumprimento em três etapas, melhor dizendo), basicamente o Cliente envia um pacote SYN (syncronization) para o Servidor, o Servidor por sua vez retorna um SYN e um ACK (acknowledge) e enfim o Cliente envia um ACK para o Servidor. Segundo o RFC793, não temos um *three-ways* handshake, e sim um *four-ways* handshake. Nesse caso, o documento afirma que antes dessas três etapas, há uma etapa inicial realizada pelo Servidor, que muda o seu estado de LISTEN para SYN SENT. Assim, essa mudança pode ser levada ao Cliente antes deste enviar o SYN para o Servidor.

7) Explique o que é MDI e MDI-X.

R: Uma MDI (*Medium-Dependent Interface*) é um dispositivo que descreve a interface (física e elétrica / óptica) em uma rede de computadores desde uma implementação da camada física até o meio físico usado para transportar a transmissão. É uma pequena chave que habilita o uso de cabo-crossover (cabo de rede par trançado que permite a ligação de 2 computadores pelas respectivas placas de rede sem a necessidade de um concentrador ou a ligação de modems). Basicamente MDI/MDIX são posições, o MDI é destinado aos roteadores, placas de redes, já o MDI-X está para os hubs e switchs.

8) Admitindo que temos apenas cabos UTP cat 5 para interligar os equipamentos da figura abaixo, diga quais configurações de cabos devem ser usados.

R: Quando houver um switch entre as placas de rede, utilizaremos conexão direta, caso contrário usaremos cross. Do PCA para o S1, do S1 para o S2 e do S2 para o R1, a configuração será direta. Do R1 para o R2 e do R2 para o PCB a configuração será cross.

9) Para os endereços abaixo, os classifique e diga a rede, host e broadcast.

R: Para resolvermos essas questões, precisaremos do conhecimento básico de binários.

1 BYTE = 1 1 1 1 1 1 1 1 =

2^7 | 2^6 | 2^5 | 2^4 | 2^3 | 2^2 | 2^1 | 2^0 =

= 128 | 64| 32 | 16 | 8 | 4 | 2 | 1

=> Faremos um AND lógico entre os binários IP e Máscara, o resultado será a REDE.

1. IP: 177.32.168.223 Masc: 255.255.255.248

IP: 10110001.00100000.10101000.11011111

Masc: 11111111.11111111.11111111.11111000

10110001.00100000.10101000.11011000 = **177.32.168.216 => REDE**

A máscara dessa rede é /29, isto é, são reservados 29 bits para redes e 3 para hosts. Para descobrirmos o broadcast, basta substituir os últimos 3 bits da rede por 1, vamos fazer isso...

10110001.00100000.10101000.11011111 = **177.32.168.223 => BROADCAST**

Note que o broadcast é o próprio IP dado!

Continuando, agora vamos encontrar os hosts. São 3 bits reservados para hosts, logo, teremos $2^3 = 8 – 2$(rede e broadcast) = 6 hosts!

177.32.168.217 ---- 177.32.168.222 => INTERVALO DE HOSTS

2. IP: 204.20.143.0 Masc: /18 (18 bits para redes e 14 para hosts)

IP: 11001100.00010100.10001111.00000000 Novamente o AND lógico !

Masc: 11111111.11111111.11000000.00000000

11001100.00010100.10000000.0000000 = **204.20.128.0 => REDE**

Substituindo por 1 os últimos 14 bits da rede...

11001100.00010100.10111111.11111111 = **204.20.191.254 => BROADCAST**

Percebe-se então que o IP fornecido é um **HOST**, segue a quantidade de hosts:

2^{14} = 16384 – 2 = 16382 hosts!

204.20.128.1 ---- 204.20.191.253 => INTERVALO DE HOSTS

3. IP: 36.72.109.24 Masc: 255.254.0.0 (/15, 15 bits para redes e 17 para hosts)

IP: 00100100.01001000.01101101.00011000 AND LÓGICO

Masc: 11111111.11111110.00000000.00000000

00100100.01001000.00000000.00000000 = **36.72.0.0 => REDE**

Substituindo os últimos 17 bits por 1 para encontrarmos o broadcast...

00100100.01001001.11111111.11111111 = **36.73.255.255 => BROADCAST**

Percebe-se então que o IP dado é um **HOST**, e a quantidade deles é dada por:

2^{17} = 131072 – 2 = 131000 hosts!

36.72.0.1 ---- 36.73.255.254 => INTERVALO DE HOSTS

4. IP: 7.26.0.64 Masc: /26 (26 bits para redes e 6 bits para hosts)

IP: 00000111.00011010.00000000.01000000 AND LÓGICO

Masc: 11111111.11111111.11111111.11000000

00000111.00011010.00000000.01000000 = **7.26.0.64 => REDE**

Substituindo os últimos 6 bits por 1 para encontrarmos o broadcast...

00000111.00011010.00000000.01111111 **= 7.26.0.137 => BROADCAST**

Percebe-se então que o IP dado é a **REDE**. Temos que o total de hosts são:

2^{6} = 64 – 2 = 62 hosts!

7.26.0.65 ---- 7.26.0.136 => INTERVALO DE HOSTS

5. IP: 200.201.173.187 Masc: 255.255.255.252 (30 bits para redes e 2 bits para hosts)

IP: 11001000.11001001.10101101.10111011 AND LÓGICO

Masc: 11111111.11111111.11111111.11111100

11001000.11001001.10101101.10111000 = **200.201.173.184 => REDE**

Substituindo os últimos 2 bits por 1 para encontrarmos o broadcast...

11001000.11001001.10101101.10111000 = **200.201.173.187 => BROADCAST**

Percebe-se então que o IP dado é o **BROADCAST**. Temos que o total de hosts são:

$2^2 = 4 – 2 = 2$ hosts!

200.201.173.185 ---- 200.201.173.186 => INTERVALO DE HOSTS

10) Diga se os endereços dados estão na mesma rede.

R: Aqui faremos algo semelhante a questão anterior, porém encontraremos os intervalos de hosts, a rede e o broadcast para os endereços IPs.

1. 240.128.192.154 e 240.128.192.158 com máscara 255.255.255.224

240.128.192.154 > 11110000.10000000.11000000.10011010 AND LÓGICO

11111111.11111111.11111111.11100000 /27

11110000.10000000.11000000.10000000 = **240.128.192.128 (rede)**

240.128.192.159(broadcast)

240.128.192.129-158(hosts)

240.128.192.158 > 11110000.10000000.11000000.10011110 AND LÓGICO

11111111.11111111.11111111.11100000 /27

11110000.10000000.11000000.10000000 = **240.128.192.128 (rede)**

240.128.192.159(broadcast)

240.128.192.129-158(hosts)

Logo, como os endereços possuem intervalos de hosts (e são hosts) semelhantes, eles pertencem a mesma rede.

2. 87.42.141.142 e 87.42.141.137 com máscara 255.255.255.248

87.42.141.142 > 01010111.00101010.10001101.10001110 AND LÓGICO

11111111.11111111.11111111.11111000 /29

01010111.00101010.10001101.10001000 = **87.42.141.136 (rede)**

87.42.141.143 (broadcast)

87.42.141.137-142(hosts)

87.42.141.137 > 01010111.00101010.10001101.10001001 AND LÓGICO

11111111.11111111.11111111.11111000 /27

11110000.10000000.11000000.10000000 = **87.42.141.136 (rede)**

87.42.141.143(broadcast)

87.42.141.137-142(hosts)

Logo, como os endereços possuem intervalos de hosts (e são hosts) semelhantes, eles pertencem a mesma rede.

3. 98.45.7.17 e 98.12.238.221 com máscara /10

98.45.7.17 > 01100010.00001100.11101110.11011101 AND LÓGICO

11111111.11000000.00000000.00000000 /10

01100010.00000000.00000000.00000000 = **98.0.0.0 (rede)**

98.63.255.255 (broadcast)

98.0.0.0 - 98.63.255.255(hosts)

98.12.238.221 > 01010111.00001100.11101110.11011101 AND LÓGICO

11111111.11000000.00000000.00000000 /10

01100010.00000000.00000000.00000000 = **98.0.0.0 (rede)**

98.63.255.255(broadcast)

98.0.0.0 - 98.63.255.255(hosts)

Logo, como os endereços possuem intervalos de hosts (e são hosts) semelhantes, eles pertencem a mesma rede.

11. De acordo com o diagrama de rede abaixo, faça o projeto de endereçamento de rede contemplando TODAS as redes descritas e suas capacidades. Todo o range 187.0.0.0/8 está à sua disposição.

R: (Não consegui resolver o exercício)

12. Classifique, quanto ao seu tipo, os protocolos RIP, OSPF e BGP.

R: O **RIP** (*Routing Information Protocol*) é um padrão para troca de informações entre os *gateways* e *hosts* de roteamento, ele é um protocolo de vetor de distâncias. É projetado para redes com tamanho moderado, que utilizam uma tecnologia razoavelmente homogênea. Não é destinado a uso em ambientes mais complexos, devido à suscetibilidade a falhas. O RIP2, é uma evolução do RIP, e destina-se a expandir a quantidade de informação útil carregada nos pacotes e também adicionar uma medida de segurança. RIP2 é um protocolo que utiliza UDP para transporte. Cada host que usa RIP2 tem um processo de roteamento que envia e recebe datagramas em UDP, na porta 520. RIP e RIP2 são para redes IPv4, enquanto o RIPng é projetado para a rede IPv6.

O **OSPF** (***Open Shortest Path First***) - é um **protocolo de roteamento dinâmico de interior de gateway** para redes que operam com protocolo IP, foi criado para substituir o protocolo RIP empregado no final da década de 1980 na então Arpanet (atual Internet) e que apresentava diversos problemas e limitações para operar satisfatoriamente em uma rede de porte.

O protocolo **Border Gateway Protocol (BGP)** foi projetado para atuar no roteamento de borda, transmitindo informações sobre toda sua estrutura de rede entre sistemas autônomos (AS). É o protocolo usado entre provedores de serviços de Internet, mas também pode ser usado entre uma empresa e um ISP. O BGP foi construído para confiabilidade, escalabilidade e controle.

Quando temos uma comunicação estabelecida entre vizinhos, utilizando o BGP, informações são enviadas descrevendo quais redes são encontradas atrás do roteador de borda. Essas informações são prefixos (classe IP atribuída a um AS), next-hop (próximo salto/caminho de como chegar até o prefixo destino), AS-Path (por quais ASes devo passar até chegar no prefixo destino) e métricas de roteamento.

13. Considere uma rede com 5 hosts onde 3 deles tem TCP window size de 64KB e 2 de 32KB. Calcule o throughput do link de borda sabendo que sua latência é de 15ms.

R: O throughput basicamente é a quantidade de dados transferidos de um lugar a outro.

TCP window size = 64KB*8 = 524288 bits, 524288*3 (3 hosts) = 1.572.864 bits

= 32KB*8 = 262144 bits, 262144*2 (2 hosts) = 524.288 bits

Somando o total de bits: 1.572.864 + 524.288 = **2.097.152 bits**

Latência: 15ms = **0,015 segundos**

2.097.152 bits/ 0,015 segundos = **139.810.133,133 bits/segundo**

Convertendo para megabits temos aproximadamente 140Mb/s de transferência entre o host e o gateway.

14. De acordo com o cabeçalho TCP, explique cada um dos campos sequence number, acknowledgement, window size e suas flags.

R: O *sequence number* (número de sequência) é o número de byte do primeiro byte de dados no pacote TCP enviado (também chamado de segmento TCP). O *acknowledgement*

number (número de confirmação) é o número de sequência do próximo byte que o receptor espera receber. O número de sequência é sempre válido. O *acknowledgement number* é válido apenas quando o sinalizador ACK é um. O único momento em que o sinalizador ACK não é definido, ou seja, o único momento em que não há um número de confirmação válido no cabeçalho TCP, é durante o primeiro pacote de configuração da conexão.

O *window size* é considerado um dos sinalizadores mais importantes no cabeçalho TCP. Este campo é usado pelo destinatário para indicar ao remetente a quantidade de dados que ele pode aceitar. Independentemente de quem seja o remetente ou o destinatário, o campo sempre existirá e será usado. Cada lado de uma conexão TCP anuncia um *window size* (tamanho de janela). Uma conexão pode ter esse número de bytes enviados, mas não reconhecidos. Cada pacote fornece uma confirmação, um *sequence number* confirmado e uma *window*. Se a confirmação for x e o tamanho da janela for w, bytes até x + w podem ser enviados. O objetivo disso é o controle de fluxo com base no consumo de dados dos aplicativos. Não é um dispositivo de controle de fluxo de rede, ou seja, um dispositivo de controle de congestionamento.

Tipos de Flags:

Synchronization (SYN) - É usada na primeira etapa da fase de estabelecimento da conexão ou no processo de handshake de três vias entre os dois hosts. Somente o primeiro pacote do remetente e do destinatário deve ter esse sinalizador definido. Isso é usado para sincronizar o número de sequência, ou seja, para dizer à outra extremidade qual número de sequência eles devem excluir.

Reconhecimento (ACK) - É usado para reconhecer pacotes recebidos com êxito pelo host. O sinalizador é definido se o campo do número de confirmação contiver um número de confirmação válido.No diagrama abaixo, o receptor envia um ACK = 1 e SYN = 1 na segunda etapa do estabelecimento da conexão para informar ao remetente que recebeu seu pacote inicial.

Finish (FIN) - É usado para solicitar o encerramento da conexão, ou seja, quando não há mais dados do remetente, ele solicita o encerramento da conexão. Este é o último pacote enviado pelo remetente. Ele libera os recursos reservados e finaliza normalmente a conexão.

Reset (RST) - É usado para encerrar a conexão se o remetente do RST achar que algo está errado com a conexão TCP ou se a conversa não deve existir. Ele pode ser enviado do lado do receptor quando o pacote é enviado para um host específico que não o esperava.

Urgent (URG) - Dados dentro de um segmento com sinalizador URG = 1 são encaminhados para a camada de aplicativo imediatamente, mesmo que haja mais dados a serem fornecidos à camada de aplicação. É usado para notificar o destinatário a processar os pacotes urgentes antes de processar todos os outros pacotes. O destinatário será notificado quando todos os dados urgentes conhecidos forem recebidos.

Push (PSH) - A camada de transporte, por padrão, espera por algum tempo que a camada de aplicação envie dados suficientes iguais ao tamanho máximo do segmento, para que o número de pacotes transmitidos na rede o minimize, o que não é desejável para alguns aplicativos, como aplicativos interativos (bate-papo). Da mesma forma, a camada de

transporte na extremidade do receptor armazena os pacotes em buffer e transmite para a camada de aplicação se ela atender a determinados critérios.Esse problema é resolvido usando o PSH. A camada de transporte define PSH = 1 e envia imediatamente o segmento para a camada de rede assim que recebe o sinal da camada de aplicação. Camada de transporte do receptor, ao ver PSH = 1, encaminha imediatamente os dados para a camada de aplicação.Em geral, ele diz ao receptor para processar esses pacotes à medida que são recebidos, em vez de armazená-los em buffer.

15) Explique de que maneira funciona o sequenciamento TCP padrão.

R: Os dados enviados são repetidos (ou ecoados) nas respostas ACK. Por exemplo, desejo enviar a informação "gris" a algum servidor, primeiramente envio *Seg=100, ack=300, dados='g'.* Após isso, o servidor "ecoa" a resposta no ACK, retornando para o cliente *Seg=300, ack=101, dados='g'.* Logo em seguida eu envio para o servidor *Seg=101, ack=301, dados='r'*, e ele me retorna *Seg=301, ack=102, dados='r',* e assim por diante. Perceba que um padrão é respeitado, a cada número ACK que eu envio, o servidor me retorna esse ACK como segmento e um novo ACK+1.

16) Explique de que maneira o TCP padrão se recupera de um timeout.

R: O timeout do usuário TCP controla quanto tempo os dados transmitidos podem permanecer não reconhecido antes de uma conexão ser fechada com força. É umparâmetro local por conexão.

Um timeout é iniciado toda vez que um segmento é transmitido. O timeout é cancelado quando o ACK correspondente é recebido. Se um pacote é perdido, mas os pacotes seguintes são recebidos, são enviados ACKs de mesmo valor (duplicados). O recebimento de três ACKs duplicados força a retransmissão do segmento perdido e cancela o timeout (fast retransmit).

17) Explique de que maneira funciona o fast retransmit TCP padrão.

R: A *Fast Retransmit* (retransmissão rápida) é um aprimoramento do TCP que reduz o tempo que o remetente espera antes de retransmitir um segmento perdido. Um remetente TCP normalmente usa um timer simples para reconhecer segmentos perdidos. Se uma confirmação não for recebida para um segmento específico dentro de um tempo especificado (uma função do tempo estimado de atraso de ida e volta), o remetente assumirá que o segmento foi perdido na rede e retransmitirá o segmento.

18) Explique de que maneira funciona o slow start e congestion avoidance TCP.

R: O *slow start* (início lento) faz parte da estratégia de controle de congestionamento usada pelo TCP em conjunto com outros algoritmos para evitar o envio de mais dados do que a rede é capaz de encaminhar, ou seja, para evitar causar congestionamento na rede.

O *slow start* começa inicialmente com um *congestion window size* (CWND) de 1, 2, 4 ou 10 MSS(*maximum segment size*). O valor do *congestion window size* será aumentado em um a cada reconhecimento recebido (ACK), dobrando efetivamente o tamanho da janela a cada tempo de ida e volta.

A taxa de transmissão será aumentada pelo algoritmo de *slow start* até que uma perda seja detectada ou a janela anunciada do destinatário (rwnd) seja o fator limitante ou a interrupção da transmissão. Se ocorrer um evento de perda, o TCP assume que é devido ao

congestionamento da rede e toma medidas para reduzir a carga oferecida na rede. Essas medidas dependem do algoritmo exato de prevenção de congestionamento de TCP usado.

19) Explique o comportamento "serrilhado" do TCP e por que ele é importante para seu funcionamento.

R: Pelo que entendi, o comportamento "serrilhado" se refere as flags trocadas entre o cliente e o servidor para que seja feito o reconhecimento e confirmação dos pacotes enviados e recebidos. Isso é importante porque há sempre uma verificação se de fato o rementente é quem ele diz realmente ser, e se o destinatário é quem ele realmente afirma ser, evitando assim que terceiros mal intencionados interceptem, leiam e alterem os dados transmitidos.

Garante também que sejam evitados congestinamentos de dados., pois a cada pico o *threshold* controla o tamanho da página.

20) Considere uma janela TCP de tamanho 8, e threshold 4. Assuma que são enviados ACK's sequenciais de 1 a 8 e os ACK's 2 e 3 se perderam. Mostre todo o sequenciamento TCP até que o envio dos ACK's seja normalizado (mostre uma janela completa correta).

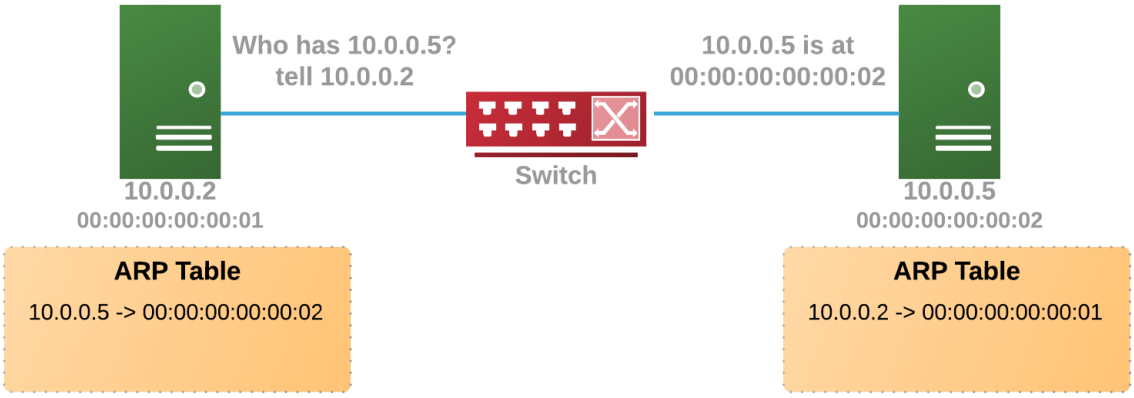
R: (Não consegui fazer a questão)

21) Defina AS (sistema autônomo).

R: Um sistema autônomo (AS) é um conjunto de redes, ou uma única rede, que além de estar sob uma gestão comum tem características e políticas de roteamento comuns. Um ASN único é atribuído a cada AS para uso em roteamento BGP. Números AS são importantes porque o ASN identifica cada rede na Internet.

22) Suponha que um host A envie uma requisição ARP para descobrir o endereço de um host B. Mostre o formato desta requisição e a resposta recebida.

R: Requisição ARP simplificada who has/tell para descobrir o endereço de host B.



23) O que é CSMA/CD? Explique brevemente.

R: é um protocolo de telecomunicações que organiza a forma como os dispositivos de rede compartilham o canal utilizando a tecnologia Ethernet. O **CS** (*Carrier Senser*) se refere a capacidade de identificar se está ocorrendo transmissão, ou seja, o primeiro passo na transmissão de dados numa rede Ethernet é verificar se o cabo está livre. O **MA** *(Multiple Access)* é a capacidade de múltiplos nós concorrerem pela utilização da mídia, ou seja, o protocolo CSMA/CD não gera nenhum tipo de prioridade (daí o nome de *Multiple Access*, acesso múltiplo). Como o CSMA/CD não gera prioridade pode ocorrer de duas placas tentarem transmitir dados ao mesmo tempo. Quando isso ocorre, há uma colisão e nenhuma das placas consegue transmitir dados. Por último, o **CD** (Collision Detection) é responsável por identificar colisões na rede.

24) O que é encapsulamento?

R: Para que duas aplicações se comuniquem em uma rede de 7 camadas como o modelo OSI, é necessário que haja um processo de **encapsulamento** e **desencapsulamento** na troca de mensagens. No encapsulamento as mensagens recebem um **cabeçalho** que contém informações de controle, o processo se repete a cada camada até que a mensagem chegue na camada física. A mensagem é então enviada de um dispositivo ao outro através de sinais pelo **meio** físico. Ao chegar no dispositivo de destino ocorre o processo inverso, onde o cabeçalho da mensagem é conferido e descartado a cada camada que passa até chegar na camada de aplicação. É importante observar que na camada de enlace é possível que a mensagem receba um **rodapé** (*trailer*), onde são adicionadas informações de controle de erros.

25) Defina o que é um protocolo, no âmbito de redes.

R: Um **protocolo** é uma convenção que controla e possibilita uma conexão, comunicação, transferência de dados entre dois sistemas computacionais.

De maneira simples, um protocolo pode ser definido como "as regras que governam" a sintaxe, semântica e sincronização da comunicação. Os protocolos podem ser implementados pelo *hardware*, *software* ou por uma combinação dos dois.

O uso do funcionamento de protocolos difundidos e a expansão dos protocolos de comunicação é ao mesmo tempo um pré-requisito e uma contribuição para o poder e sucesso da Internet. O par formado por IP e TCP é uma referência a uma coleção dos protocolos mais utilizados. A maioria dos protocolos para comunicação via Internet é descrita nos documentos RFC do IETF.