

1. Diga quais são as camadas do modelo TCP/IP, uma descrição e um protocolo.

1) quais são as camadas e a descrição do protocolo de cada uma delas:

R:

As camadas do modelo TCP/IP são:

**Aplicação:**

Essa camada atua apenas nos dispositivos finais na comunicação. As funções da camada de aplicação são:

- É utilizado para desenvolver aplicações baseadas na rede.
- Fornece serviços ao usuário como login de usuário, nomeação de dispositivos de rede, formatação de mensagens e e-mails, transferência de arquivos, etc.
- tratamento de erros e recuperação da mensagem.

HTTP --> protocolo de hipertexto, utilizado para trazer as paginas da web

SMTP protocolo utilizado para envio de emails

IMAP protocolo utilizado para envio de emails

FTP, protocolo utilizado para transferencia de arquivos

SIP, protocolo utilizado para transporte de voz (VoIP)

SSH, protocolo utilizado para estabelecer conexoes seguras entre 2 hosts

**transporte:**

na camada de transporte, ocorre a conexão entre hosts/portas, além de definirmos o protocolo utilizado, TCP ou UDP.

TCP --> É um protocolo orientado a confirmação, ou seja, a cada mensagem enviada, o remetente pede a confirmação para o destinatário para checar se os dados foram recebidos, se o host recebeu, envia a confirmação para o remetente, senão, pede o reenvio dos dados que não chegaram, tornando o protocolo mais seguro em relação ao UDP.

UDP --> não pede confirmação pro usuário, o que torna a conexão mais rápida

## internet:

camada responsável por entregar, endereçar e reconstruir os pacotes.

IPv4 - formado por 4 octetos (4 grupos de 8 bits), ou seja, abranje  $2^{32}$  endereços distintos

IPv6 – sucessor do IPv4, essencialmente, tem os mesmos objetivos, mas diferente do IPv4 que utiliza-se de 32 bits, o IPv6 utiliza-se de 128 bits, ou seja, o IPv6 possui  $2^{128}$  endereços diferentes.

Diferença detalhadas entre IPv4 e IPv6:

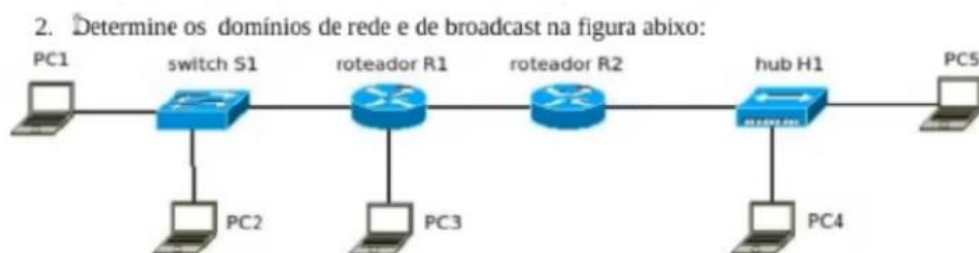
<i>IPv4</i>	<i>IPv6</i>
<i>Os endereços têm 32 bits (4 bytes) de tamanho.</i>	<i>Os endereços têm 128 bits (16 bytes) de tamanho.</i>
<i>Cada endereço corresponde a uma conexão, que muitas vezes é dividida entre vários computadores.</i>	<i>Com tantos endereços, cada computador terá o seu endereço real na internet.</i>
<i>Um endereço IP é binários números, mas podem ser armazenados como texto para leitores humanos. Por exemplo, um endereço de 32 bits numérico (IPv4) é escrito em decimal como quatro números separados por pontos. Cada número pode ser igual a zero a 255. Por exemplo, 1.160.10.240 poderia ser um endereço IP.</i>	<i>Os endereços IPv6 são 128-bit endereço IP escrito em hexadecimal e separados por dois pontos. Um exemplo de endereço IPv6 poderia ser escrito assim: 3ffe: 1900:4545:3:200: f8ff: fe21: 67cf</i>
<i>IPSec é opcional e deverá ser suportado externamente.</i>	<i>O suporte ao IPSec não é opcional.</i>
<i>Pode ser configurado manualmente ou por DHCP.</i>	<i>Não requer configuração manual ou DHCP.</i>
<i>Deve suportar um tamanho de pacote de 576-byte (possivelmente fragmentado).</i>	<i>Deve suportar um tamanho de pacote de 1280-byte (sem fragmentação).</i>

## acesso a rede:

interliga as camadas anteriores com a rede. As principais funções executadas dentro da camada de acesso a rede são encapsulamento (proteção dos dados na rede), mapeamento, e endereçamento de endereços IP aos endereços físico.

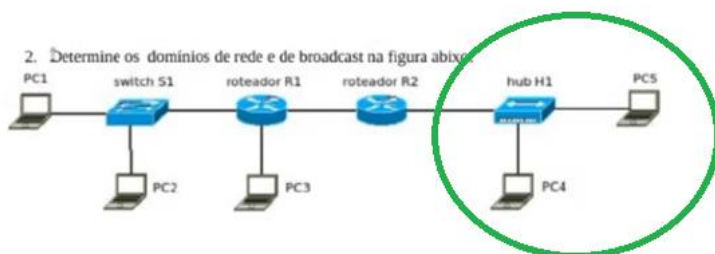
Ethernet – protocolo responsável por definir a velocidade e a maneira que os dados vão trafegar no meio físico.

## Ilustração do modelo TCP/IP:

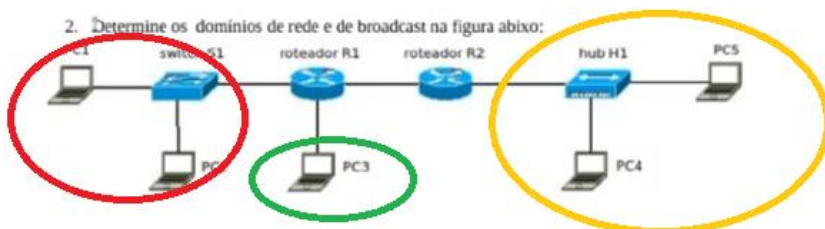


R:

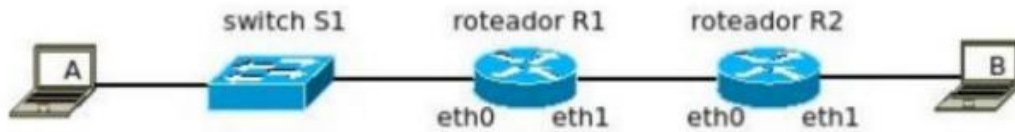
Temos 1 domínios de colisão:



Temos 3 domínios de broadcast, sendo eles listados abaixo:



- Suponha que A envia um pacote para B.  
Mostre o pacote ao longo do caminho, suas camadas e protocolos

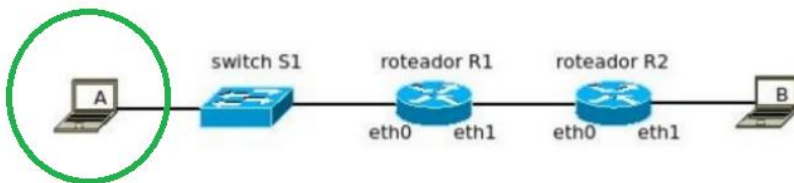


Aqui temos uma conexão do tipo peer-2-peer, onde os usuarios se conetam com o outro, sem um servidor no meio dessa conexão.

### Momento 1:

Camada de aplicação

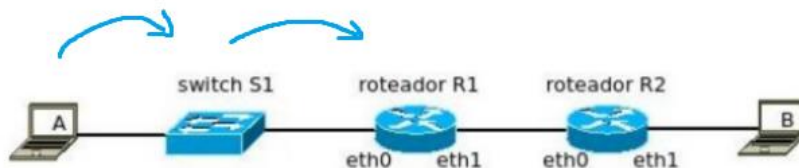
Chama uma syscall para escrever a mensagem através da placa de rede



O procolo definido na camada de aplicação depende do tipo de conteudo do pacote. Neste caso, vamos supor que seja enviado arquivos, via torrent, ou seja, o protocolo adotado será o FTP.

### Momento 2

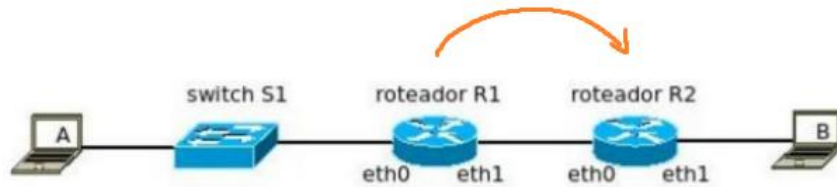
Camada de transporte



ocorre a conexão entre hosts/portas, além de definirmos o protocolo utilizado, TCP ou UDP. No caso, a conexão entre os dois usuários está ocorrendo por meio do torrent, onde a camada de transporte utiliza o TCP.

### Momento 3:

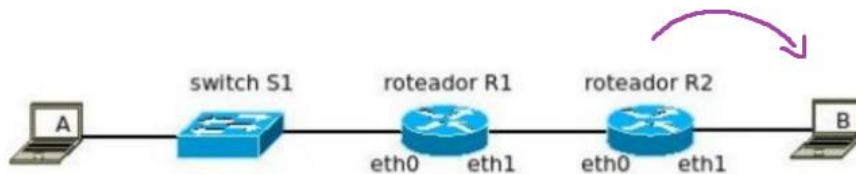
camada de internet



camada responsável por entregar, endereçar e reconstruir os pacotes.

### Momento 4:

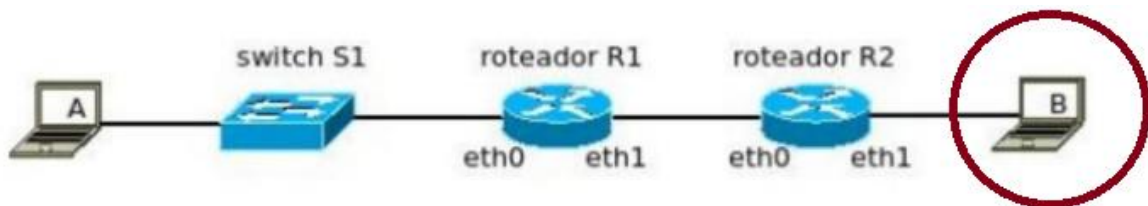
Transporte



Responsavel pela conexão entre hosts/portas

### Momento 5:

Camada de aplicação



quando o pacote chega no computador B, precisamos de uma syscall de write/read  
write para escrever no meu HD, e read para poder acessar o arquivo recebido.

- Para os endereços abaixo, diga a rede, host e broadcast:
  - IP:177.32.168.223 Máscara:255.255.255.248
  - IP:7.26.0.64 Máscara:/26

R:

#### Exemplo1:

IP: 177.32.168.223 máscara: 255.255.255.248 Nessa notação o 248 (no último conjunto de 8 bits) é o mesmo que  $248 = 255 - 1 - 2 - 4$ . Com isso, na mascara, temos tudo 1, exceto nos tres bits menos significativos. máscara: 255.255.255.248 = máscara: /29

Mascara:

11111111 . 11111111 . 11111111 . 11111 000

Para acharmos o endereço que serve para identificar a rede, fazemos um and (bit a bit) da mascara com o ip.

Rede = Resultado do and bit a bit: 10110001 . 00100000 . 10101000 . 11011 000

Decimal 177 . 32 . 168 . 216

broadcast : 10110001 . 00100000 . 10101000.11011 111

decimal 177 . 32 . 168 . 223

Todos os endereços entre a rede e o broadcast servem para os hosts, temos então  $(2^3-2)$  endereços

#### Exemplo2:

IP: 7.26.0.64 máscara: /26 --> esse 26 indica que os 26 primeiros bits são 1 na mascara

Máscara:

11111111 . 11111111 . 11111111 . 11 000000

Para acharmos o endereço que serve para identificar a rede, fazemos um and (bit a bit) da mascara com o ip.

Rede = 00000111 . 00011010 . 00000000 . 01 000000

Decimal 7 . 26 . 0 . 64

Broadcast: 00000111 . 00011010 . 00000000 . 01 111111

Decimal 7 . 26 . 0 . 127

Todos os endereços entre a rede e o broadcast servem para os hosts, temos então  $(2^6-2)$  endereços

- Explique o comportamento “serrilhado” do TCP

R:

**Controle de fluxo:**

Se resume em capacidade de envio x capacidade de recebimento

**Se capacidade de recebimento > capacidade de envio --> o envio acontece normalmente**

**Se capacidade de recebimento < capacidade de envio --> acontecerá perda de pacote.**

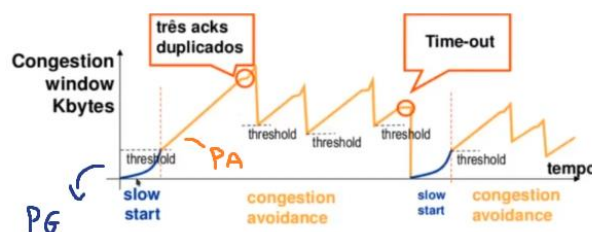
**Objetivo:** O controle de fluxo visa chegar no MTU(eixo y) ideal, sendo que, existem diversos fatores que podem influenciar o meu MTU ideal, ou seja, o meu MTU ideal muda dinamicamente e, o algoritmo abaixo, visa equilibrar a capacidade de envio x capacidade de recebimento, tentando sempre fazer que o pacote seja o maior possível e a perda de pacote a menor possível, de acordo com essas mudanças externas.

**Descrição do slow start:**

**Threshold Política de crescimento da capacidade de envio(cwnd):**

A variável responsável pelo controle da capacidade de envio é chamada de cwnd e tem seu valor inicial 1. A cada resposta positiva que diz que o pacote chegou ao seu destino(ack), o cwnd dobra o seu valor, ou seja, cresce em pg de razão 2.

O threshold controle o crescimento da janela, é ele que controla o crescimento do cwnd que, a principio a politica de crescimento é em PG, mas posteriormente passa a ser em PA. O threshold funciona como um checkpoint que armazena o estado(dinamico) que é a capacidade de recebimento por parte do meu receptor.

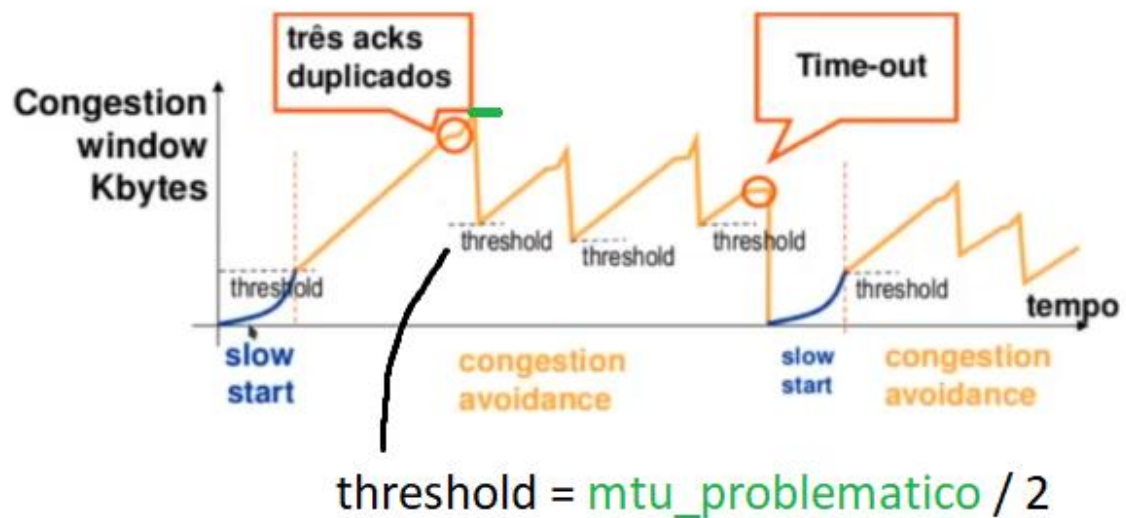


## Descrição do congestion avoidance:

### A fim de contornar a perda de pacote fazemos(3 acks duplicados):

Em um conexão do tipo TCP, eu envio o pacote eu deveria receber uma mensagem que diz se o pacote chegou(ack). Se eu recebo tres acks duplicados (3 acks referentes ao mesmo pacote), significa que eu aquele pacote não está chegando corretamente, com isso eu conluo que devo diminuir a capacidade de envio. No controle de fluxo, reduzimos essa velocidade pela metade (MTU). Vale destacar que a politica de crescimento neste contexto é a PA de razão 1.

Quanto ocorre essa divisão do meu MTU por 2, setamos o threshold como o resultado dessa operação, ou seja,  $\text{threshold} = \text{MTU\_problematico} / 2$



### Em caso de timeout:

Quando ocorre o timeout, voltamos a etapa de slow start mas o threshold continua com seu valor anterior, ou seja, utilizamos um crescimento em PG para atingir o threshold anterior, o que nos permite atingir o threshold rapidamente.

## Conclusão:

Esse comportamento “serrilhado” se deve ao fato de eu estar sempre regulando a minha capacidade de envio x capacidade de recebimento, com base nas regras descritas anteriormente.