

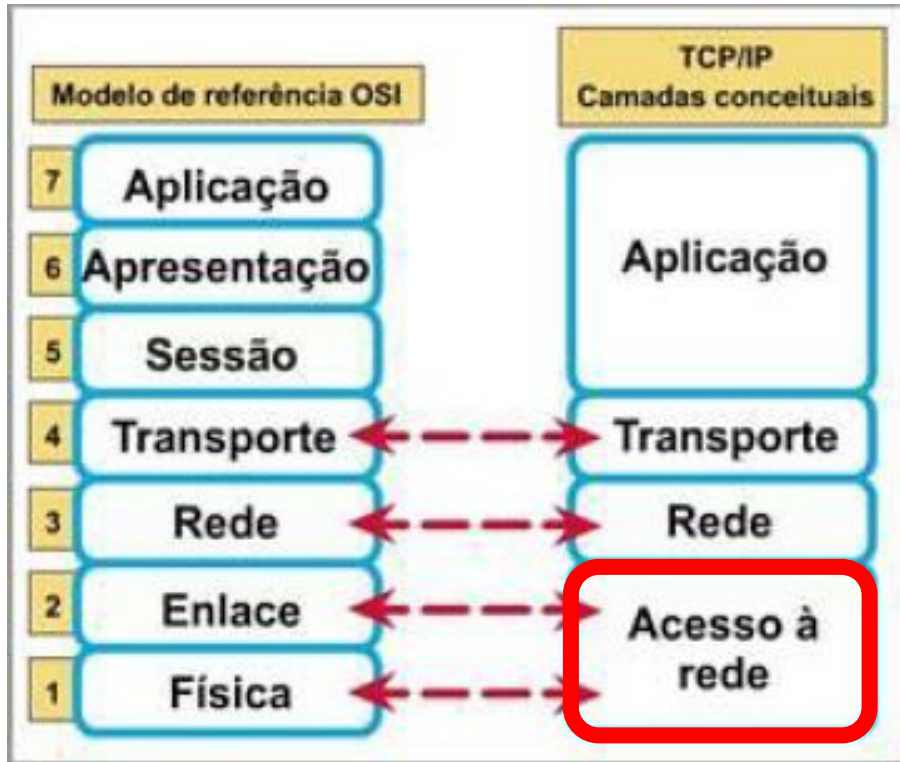
CAMADA FÍSICA

GRUPO 5: Eduardo Humberto
 Felipe Freitas
 Mariana Bulgarelli
 Yulli Dias

CAMADA FÍSICA



PHP



CAMADA FÍSICA

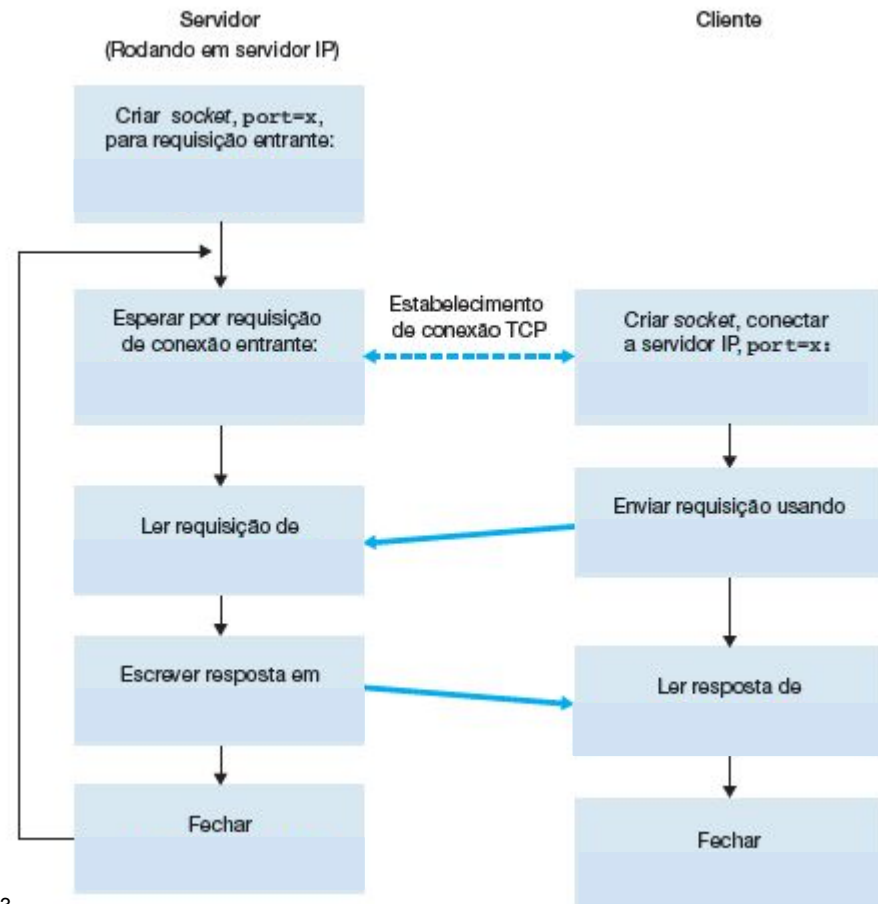
A camada **física** é responsável pela coordenação das funções para transportar de bits de um nó para o seguinte [2].

“A camada de **enlace** [...] transforma a camada física, um recurso de transmissão bruto, em um link responsável pela comunicação de dados nó-a-nó” [2].

CAMADA FÍSICA

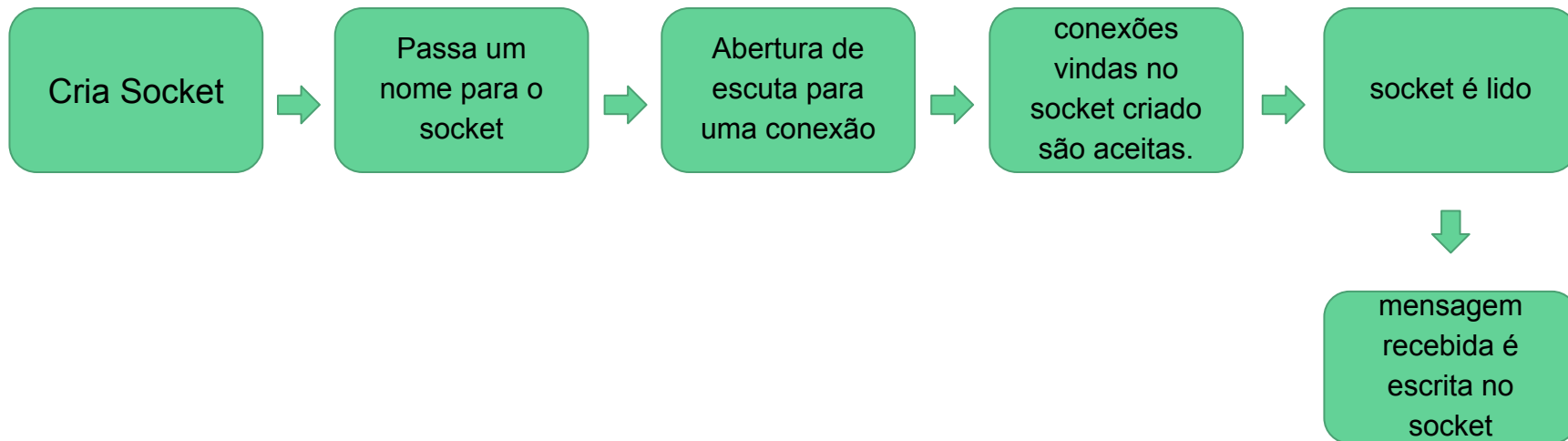
O TCP é um protocolo orientado a conexão. Uma ponta da conexão está ligada ao socket cliente e a outra está ligada a um socket servidor.

Ao criar a conexão TCP, associamos a ela o endereço de socket (endereço IP e número de porta) do cliente e do servidor. Com a conexão estabelecida, quando um lado quer enviar dados para o outro, basta deixá-los na conexão TCP por meio de seu socket (KUROSE e ROSS, 2013).



CAMADA FÍSICA - ESCOPO DE IMPLEMENTAÇÃO

SERVIDOR



CAMADA FÍSICA - ESCOPO DE IMPLEMENTAÇÃO

CLIENTE



CAMADA FÍSICA - CÓDIGO

SERVIDOR

```
1: Project
└─ Servidor.php
   log.txt

1  <?php
2  $ARQUIVO_LOG = "../log.txt";
3  $MEU_IP = "127.0.0.1";
4  $MINHA_PORTA = 8080;
5  $TAM_MAX_BYTES = '3000000';
6
7  function binarioParaString($sequenciaDeBits){
8      $string = '';
9      for($i=0; $i<(strlen($sequenciaDeBits)-1); $i+=8){
10         $hex = base_convert(substr($sequenciaDeBits, $i, length: 8), frombase: 2, tobase: 16);
11         while(strlen($hex)<2)
12             {
13                 $hex = '0'.$hex;
14             }
15         $caracter = pack( format: 'H*', $hex);
16         $string .= $caracter;
17     }
18     return $string;
19 }
20 function stringParaBinario($string){
21     $stringEmBinario = '';
22     $arrayDeCaracter = str_split($string);
23     foreach($arrayDeCaracter as $caracter){
24         $caracterEmHexadecimal = unpack( format: 'H*', $caracter);
25         $caracterEmBinario = base_convert($caracterEmHexadecimal[1], frombase: 16, tobase: 2);
26         while(strlen($caracterEmBinario)<8){ $caracterEmBinario = '0'.$caracterEmBinario; } //garante que tem 8 bits
27         $stringEmBinario .= $caracterEmBinario;
28     }
29     return $stringEmBinario;
30 }
```

CAMADA FÍSICA - CÓDIGO

SERVIDOR

```
31
32 function obterMensagemDoQuadro($quadro){
33     $preambulo = substr($quadro, start: 0, length: 4); //4 bits
34     $sfd = substr($quadro, start: 4, length: 8); //8 bits
35     $mac_org = substr($quadro, start: 12, length: 48);
36     $mac_dest = substr($quadro, start: 60, length: 48);
37     $tipo = substr($quadro, start: 108, length: 16); //16 bits
38     $tam_dado = strlen($quadro) - 156; //tamanho total - cabeçalho - crc
39     $data = substr($quadro, start: 124, $tam_dado);
40     $data = binarioParaString($data); //converte o pacote para string
41     $crc = substr($quadro, start: 124+$tam_dado, length: 32); //crc tem 32 bits
42     return $data;
43 }
44 function timestamp()
45 {
46     $now = getdate();
47     $data = $now['mday'] . ' ' . $now['month'] . ' ' . $now['year'] . ' ' . $now['hours'] . ':' . $now['minutes'] . ':' . $now['seconds'] . " ";
48     return $data;
49 }
50 function escreveNoLog($mensagem)
51 {
52     file_put_contents ( $GLOBALS['ARQUIVO_LOG'], data: timestamp() . "[Física: Servidor] " . $mensagem . ".\n", flags: FILE_APPEND | LOCK_EX); //lock_ex lock exclusivo
53 }
54 set_time_limit( seconds: 0); //sem timeout
55 $socket = socket_create( domain: AF_INET, type: SOCK_STREAM, protocol: 0);
56 if($socket === FALSE)
57 {
58     escreveNoLog( mensagem: "Socket com a camada física não criado");
59 }
60 else
61 {
62     escreveNoLog( mensagem: "Socket com a camada física criado");
63 }
64 }
65 }
```


CAMADA FÍSICA - CÓDIGO

SERVIDOR

```
65
66 if(socket_bind($socket, $GLOBALS['MEU_IP'], $GLOBALS['MINHA_PORTA']) === FALSE)
67 {
68     escreveNoLog( mensagem: "Erro ao vincular nome para o socket");
69 }
70 else
71 {
72     escreveNoLog( mensagem: "Vinculando um nome para o socket");
73 }
74
75 do
76 {
77     $result = socket_listen($socket);
78     if($result === false)
79     {
80         escreveNoLog( mensagem: "Erro ao ouvir conexão");
81     }
82     else
83     {
84         escreveNoLog( mensagem: "Ouvindo a conexão");
85     }
86     $spawn = socket_accept($socket);
87     if($spawn === false){
88         escreveNoLog( mensagem: "Conexão não aceita");
89     }
90     else{
91         escreveNoLog( mensagem: "Conexão aceita");
92     }
93     $quadro = socket_read($spawn, intval($TAM_MAX_BYTES));
94     if($quadro === FALSE)
95     {
96         escreveNoLog( mensagem: "Erro ao receber o quadro");
97     }
98     else
99     {
100         escreveNoLog( mensagem: "Quadro recebido");
101     }
102 }
```

```
function montaQuadro(&$macIp)
{
    $ipDestino = getIpPacote();
    $mensagem = getMensagemPacote();
    $preambulo = '0101';
    $sfd = '10101011'; // Delimitador de início de quadro
    $macOrigem = macParaBinario(getMAC($GLOBALS['IP_ORIGEM'], &$macIp));
    $macDestino = macParaBinario(getMAC($ipDestino, &$macIp));
    $tipo = '0100100101010000'; // IP
    $data = stringParaBinario($mensagem);
    $crc = '01000101010100100101001001001111'; // string ERRO
    return $preambulo.$sfd.$macOrigem.$macDestino.$tipo.$data.$crc;
}
```

CAMADA FÍSICA - CÓDIGO

Cliente

```
function enviarMensagemEObterRespostaDoServidor($mensagem, $limite)
{
    $socket = socket_create( domain: AF_INET, type: SOCK_STREAM, protocol: 0);
    if ($socket === FALSE){
        escreveNoLog( mensagem: "Socket com a camada física não criado");
    }
    else
    {
        escreveNoLog( mensagem: "Socket com a camada física criado");
    }
    $result = socket_connect($socket, $GLOBALS['IP_DESTINO'], $GLOBALS['PORTA_SERVIDOR_FISICA']);
    if($result === FALSE)
    {
        escreveNoLog( mensagem: "Conexão criada com a camada física");
    }
    enviarMensagemServidor($socket, $mensagem);
    $resposta = receberRespostaServidor($socket, $limite);
    socket_close($socket);
    return $resposta;
}
```

CAMADA FÍSICA - CÓDIGO

Cliente

```
$N_maxTentativas = 10;
$tentativa = 0;
while($tentativa < $N_maxTentativas)
{
    if(rand(0,100) > 30)
    {
        $tentativa += 1;
        escreveNoLog( mensagem: "Colisão! Tentativa " . $tentativa);
        sleep(rand(0,3));
    }
    else
    {
        $tentativa = 0;
        $mensagem = montaQuadro( &macIp: $MAC_from_IP);
        $resposta = enviarMensagemEObterRespostaDoServidor($mensagem, $GLOBALS['LIMITE_MAXIMO_MENSAGEM']);
        if(strcmp($resposta, $mensagem) == 0)
        {
            print "\n\nPacote recebido com sucesso!\n\n";
        }
        break;
    }
    sleep( seconds: 1);
}
```

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] SILVA, Adelson de Paula. Princípios de Comunicação de Dados - Arquitetura de Redes de Computadores. Centro Federal de Educação Tecnológica de Minas Gerais, Engenharia da Computação, 2018.
- [2] FOROUZAN, B. A. Comunicação de dados e redes de computadores. Tradução de Ariovaldo Griesi. Quarta Edição, McGraw Hill, Bookman, 2008.
- [3] PHP Manual. Disponível em: <<http://www.php.net/docs.php>>.
- [4] Geeksforgeeks. Computer Network | Ethernet Frame Format. Disponível em: <<https://www.geeksforgeeks.org/computer-network-ethernet-frame-format/>>.
- [5] KUROSE, James F.; ROSS, K. W.. Redes de computadores e a Internet: uma abordagem top-down. Tradução Daniel Vieira, 6. ed. – São Paulo: Pearson Education do Brasil, 2013.