



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS  
ENGENHARIA DA COMPUTAÇÃO

# **SIMULAÇÃO DO FUNCIONAMENTO DO MODELO DE PROTOCOLOS TCP/IP**

**THAIS DINIZ BRAZ**

Orientador: Sandro Renato Dias  
Centro Federal de Educação Tecnológica de Minas Gerais

BELO HORIZONTE  
JUNHO DE 2016

**THAIS DINIZ BRAZ**

# **SIMULAÇÃO DO FUNCIONAMENTO DO MODELO DE PROTOCOLOS TCP/IP**

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS  
ENGENHARIA DA COMPUTAÇÃO  
BELO HORIZONTE  
JUNHO DE 2016

## Lista de Figuras

Figura 1 – Funcionamento em camadas . . . . .	3
Figura 2 – Pilha do modelo de referência OSI . . . . .	4
Figura 3 – Pilha do modelo TCP/IP e o nome das PDU's transferidas entre as camadas	5
Figura 4 – Relação entre o modelo de referência OSI e os protocolos pertencentes ao modelo TCP/IP . . . . .	6
Figura 5 – Relação entre o modelo de referencia OSI, o modelo de protocolos TCP/IP e o modelo híbrido . . . . .	6
Figura 7 – Estrutura da PDU do protocolo UDP . . . . .	9
Figura 6 – Estrutura da PDU do protocolo TCP . . . . .	9
Figura 8 – Estrutura do datagrama IP . . . . .	10
Figura 9 – Funcionamento em camadas . . . . .	14
Figura 10 – Esquema da arquitetura desenvolvida . . . . .	15
Figura 11 – Three Way Handshake - Estabelecimento de conexão TCP . . . . .	18
Figura 12 – Resultados obtidos na execução do módulo de simulação referente a camada de Transporte . . . . .	19

## Lista de Tabelas

Tabela 1 – Métodos de requisição HTTP . . . . .	17
---	----

## Lista de Abreviaturas e Siglas

ZNB

HTTP      *HyperText Transfer Protocol*

UDP      *User Datagram Protocol*

TCP      *Transmission Control Protocol*

CWR      *Congestion window reduces - Flag pertencente ao cabeçalho TCP*

ECE      *Explicit Congestion Notification Echo - Flag pertencente ao cabeçalho TCP*

URG      *Urgent - Flag pertencente ao cabeçalho TCP*

ACK      *Acknowledgment - Flag percente ao cabeçalho TCP*

PSH      *Push - Flag percente ao cabeçalho TCP*

RST      *Reset - Flag percente ao cabeçalho TCP*

SYN      *Synchronize - Flag percente ao cabeçalho TCP*

FIN      *Finished - Flag percente ao cabeçalho TCP*

# Sumário

<b>1 – Introdução</b>	<b>1</b>
<b>2 – Fundamentação Teórica</b>	<b>3</b>
1    Modelo de Referencia ISO OSI	4
2    Modelo TCP/IP	4
2.1    Aplicação	7
2.2    Transporte	8
2.3    Rede	10
2.4    Enlace	11
3    Aplicações	11
<b>3 – Trabalhos Relacionados</b>	<b>13</b>
<b>4 – Metodologia</b>	<b>15</b>
<b>5 – Roteiro de funcionamento</b>	<b>17</b>
<b>Referências</b>	<b>20</b>

# 1 Introdução

A Internet está intrinsecamente presente na vida da sociedade contemporânea, de tal forma que as pessoas a consideram indispensável. Inseridos neste contexto, o modo de ter acesso ao mundo se adaptou: a obtenção de informações, comunicação, estudo, pesquisas, relacionamentos e etc.

[Kurose e Ross \(2013\)](#) levantam a possibilidade da Internet ser o "maior sistema de engenharia já criado pela humanidade". Porém, apesar da Internet fazer parte da vida cotidiana de forma tão categórica, o primeiro contato, do ponto de vista do usuário doméstico, ocorreu apenas nos anos 90, pouco mais de 20 anos após o início do seu desenvolvimento.

Em 1967 foi apresentada, pela ARPA (*Advanced Research Projects Agency*) do departamento de defesa dos Estados Unidos, a ARPANET. Idealizada como uma pequena rede de computadores conectados, no qual cada host (sistema final conectado à rede) fosse ligado a um computador especializado, denominado *Interface Message Processor* (IMP), sendo que estes seriam conectados a outros IMPs. Esta ideia foi concretizada 2 anos depois e em 1969 o sistema começou a operar em quatro localidades distintas. As Universidades da Califórnia de Los Angeles e Santa Barbara (UCLA e UCSB, respectivamente), em conjunto com o Instituto de Pesquisa de Stanford (SRI) e com a Universidade de Utah foram conectados através de seus IMPs. Um software, chamado NCP (*Network Control Protocol*), possibilitava a comunicação entre os institutos ([FOROUZAN; FEGAN, 2008](#)). Foi a primeira rede de comutação de pacotes operacional (os dados a serem transferidos através da rede são divididos em pequenas partes, multiplexadas em conexões entre máquinas) ([COMER, 2006](#)).

A partir do funcionamento bem sucedido da ARPANET, a mesma tecnologia de troca de pacotes foi empregado na comunicação por rádio tático e por satélite (SATNET). Porém, devido a particularidades de cada ambiente de comunicação, cada uma dessas redes utilizava diferentes parâmetros técnicos, foi necessário, então a criação de protocolos para integrá-las ([STALLINGS, 2007](#)). Vincent Cerf e Bob Khan idealizaram, em 1972, um projeto de interligação de rede, chamado *Internetting Project*, que pode ser considerado o predecessor da própria Internet. Para resolver os problemas de comunicação direta entre diferentes hosts, a solução sugerida foi a criação um hardware intermediário entre uma rede e outra, este dispositivo foi chamado *gateway*.

Em 1974 foi publicado um artigo ([CERF; KAHN, 1974](#)) no qual foi apresentado o protocolo TCP. A descrição presente no artigo original resume precisamente o conteúdo proposto: "Um protocolo que suporta o compartilhamento de recursos que existem em diferentes redes de comutação de pacotes. O protocolo prevê variação em tamanhos de

pacotes de rede individuais, falhas de transmissão, sequenciamento, controle de fluxo, verificação de erros fim-a-fim, bem como a criação e destruição de conexões lógicas processo-a-processo".

De acordo com [Forouzan e Fegan \(2008\)](#), pouco tempo depois as autoridades decidiram dividir o TCP em dois protocolos distintos, o TCP (*Transmission Control Protocol*) e o IP (*Internetworking Protocol*), e, em 1983, em detrimento dos protocolos originais da ARPANET, o TCP/IP tornou-se o modelo de conexão oficial da Internet. Ou seja, a partir de então, para se conectar à Internet, tornou-se necessário a execução deste modelo, o qual possibilita a intercomunicação entre redes heterogêneas, conectando, assim, milhares de dispositivos comunicantes de abrangência mundial.

O modelo de protocolo TCP/IP é definido oficialmente pelo IETF (*Internet Engineering Task Force*), instituição que especifica os padrões que serão implementados e utilizados em toda a Internet, no RFC<sup>1</sup> 1180 ([IETF, 1991](#)).

Pode-se perceber assim a importância do modelo TCP/IP, e, desta forma, surge a necessidade do seu estudo e entendimento por parte dos alunos de cursos que possuem em seu escopo o ensino de Redes de Computadores, como Engenharia da Computação e Elétrica, Sistemas de Informação, Ciência da Computação e etc. e consequentemente o conhecimento dos profissionais no mercado.

Dado tal contexto este trabalho se propôs, então, a partir da simulação do funcionamento do modelo de protocolos TCP/IP, formular um sistema a ser utilizado na aprendizagem, pretendendo compor a didática do ensino deste tópico na disciplina de Redes de Computadores, com o objetivo apoiar o docente no ensino e os alunos no aprendizado. Essa simulação é composta por quatro aplicações, cada uma simulando uma das camadas presentes na pilha de protocolos TCP/IP: aplicação, transporte, rede e física. Visando a exposição dos pacotes referentes a cada camada, e seu conteúdo.

Está incluso no escopo do trabalho apenas os principais protocolos utilizados em cada camada e seus funcionamentos ideais. Futuramente pretende-se a inclusão de erros e tratamento destes, para aprofundar o aprendizado da lógica por trás das ações de cada protocolo.

---

<sup>1</sup> *Request for Comments*: Documentos mantidos pela IETF que contêm conteúdo técnico e organizacional sobre a Internet.



## 2 Fundamentação Teórica

Os protocolos, anteriormente mencionados neste trabalho, são um conjunto de formalidades a serem seguidas para permitir a conversação e o entendimento entre dois dispositivos conectados à rede, chamados hosts. Nos modelos de conexão a serem apresentados (OSI e TCP/IP), estes protocolos são divididos em diferentes camadas para que assim os problemas possam ser tratados separadamente. Essa divisão se faz necessária devido à complexidade na comunicação. Além disso, este modelo provê a abstração de uma camada à sua respectiva camada superior.

Quando o dado é enviado de um host para outro, ele passa pelas várias camadas estruturadas ainda no remetente, as quais adicionam informações de forma incremental, as quais serão necessárias para seu envio ou entendimento por parte do host que irá receber este pacote, gerando assim uma nova PDU (*Protocol data unit*) a cada camada. Ao chegar no host destino este dado passa novamente pela pilha, porém em sentido inverso, e cada camada retira, interpreta os dados enviados pela camada correspondente e repassa pacote resultante é para camada superior. Este funcionamento é mostrado na Figura 1, onde as PDU's são representadas pelos elementos ao lado de cada camada, é possível perceber seu incremento e decremento nos hosts remetente e destinatário, respectivamente. Todo este processo é realizado por protocolos pertencentes à cada nível, assim cada camada conversa apenas com sua correspondente.

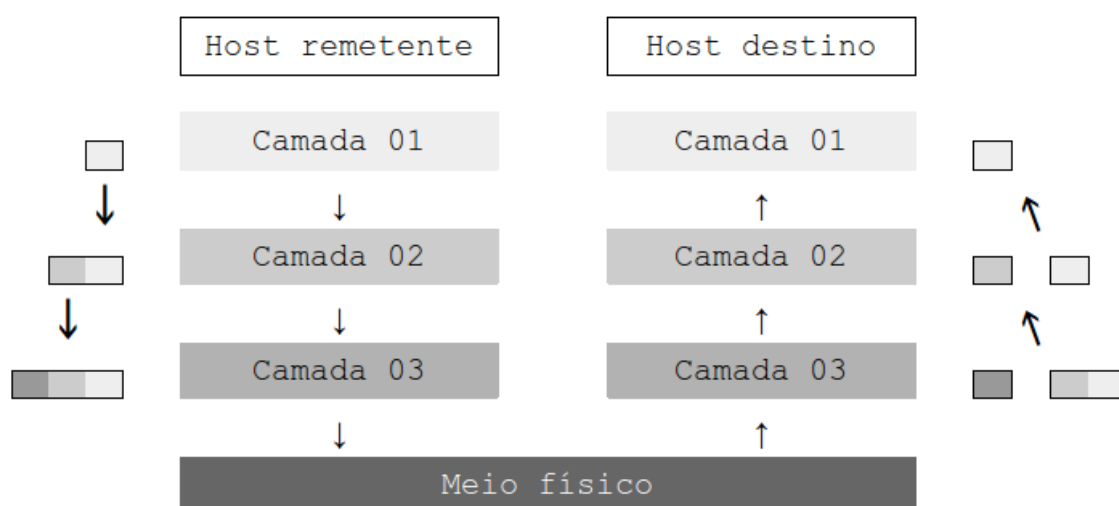


Figura 1 – Funcionamento em camadas

## 1 Modelo de Referencia ISO OSI

O modelo OSI (*Open System Interconnection*), Figura 2, criado no final dos anos 70 pela ISO (*International Standards Organization*), é um modelo conceitual estruturado em camadas, que de acordo com [Forouzan e Fegan \(2008\)](#) aborda todos os aspectos da comunicação de rede. Este mesmo autor define o termo "Sistema Aberto"(termo presente no nome dado ao modelo), como um conjunto de protocolos os quais permitem a comunicação entre dois sistemas distintos, inclusive em suas arquiteturas.



Figura 2 – Pilha do modelo de referência OSI

Fonte: ([COMER, 2006](#))

Para que isto seja possível são necessárias camadas bem delimitadas e com funções bem específicas: de acordo com [Stallings \(2007\)](#), limites bem definidos tornam possível que mudanças em uma camada não afetem os softwares existentes em outras camadas, já as restrições de suas funções provê o desenvolvimento independente de padrões.

## 2 Modelo TCP/IP

A arquitetura do modelo TCP/IP é apresentada no RFC 1122 ([IETF, 1989b](#)), o qual especifica os requerimentos exigidos para que um host seja capaz de se conectar à Internet. Este explana sobre as três últimas camadas do modelo: transporte, internet e enlace; já a primeira camada é apresentada separadamente pelo RFC 1123 ([IETF, 1989a](#)).

Diferentemente do modelo ISO o conjunto de protocolos TCP/IP não são necessariamente interdependentes. Segundo [Forouzan e Fegan \(2008\)](#), os protocolos presentes nesse modelo possuem uma certa autonomia e podem ser organizados de acordo com a necessidade da rede. O termo hierárquico apenas garante que um protocolo da camada superior seja suportado por outro(s) protocolo(s) da camada inferior.

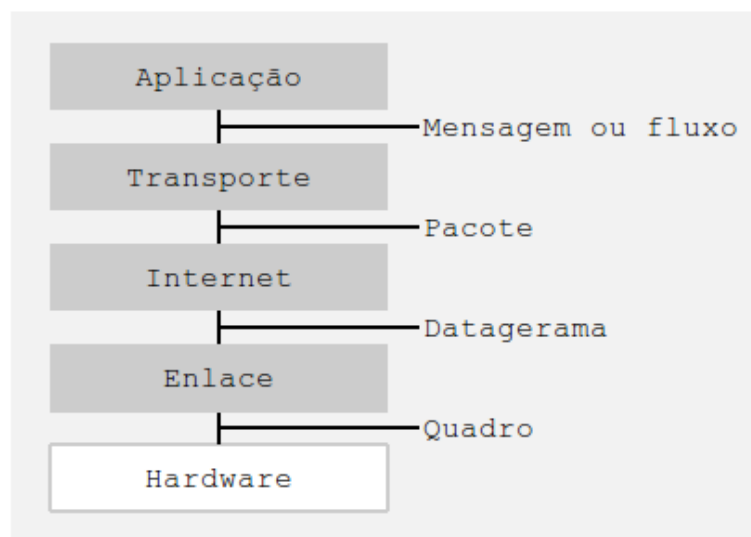


Figura 3 – Pilha do modelo TCP/IP e o nome das PDU's transferidas entre as camadas

Fonte: ([COMER, 2006](#))

A Figura 3 representa a pilha do modelo TCP/IP e a nomenclatura referente aos pacotes, PDU's, transferidos entre cada camada. Complementarmente na Figura 4, retirada do livro de [Forouzan e Fegan \(2008\)](#), está representada a relação do modelo de camadas ISO e os protocolos pertencentes ao modelo TCP/IP.

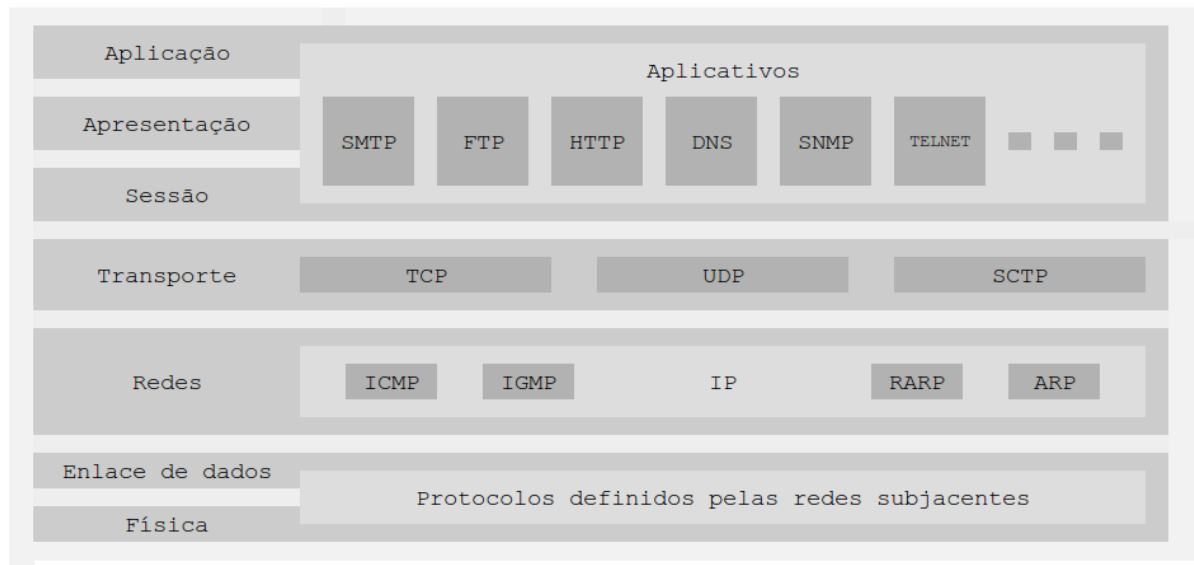


Figura 4 – Relação entre o modelo de referência OSI e os protocolos pertencentes ao modelo TCP/IP

Fonte: (FOROUZAN; FEGAN, 2008)

Ademais é importante ressaltar que a maioria das literaturas que tratam deste assunto utilizam um modelo híbrido, entre o OSI e com TCP/IP. Este modelo apresenta 5 camadas e sua relação com o modelo original TCP/IP e o modelo OSI está representada na Figura 5. De acordo com (TANENBAUM; WETHERALL, 2011) esse modelo visa incluir os benefícios de ambas referências. Enquanto a separação do modelo OSI e as definições características de suas camadas ainda o torna desejável como referência, o modelo TCP/IP contribui com seus protocolos amplamente aplicados.

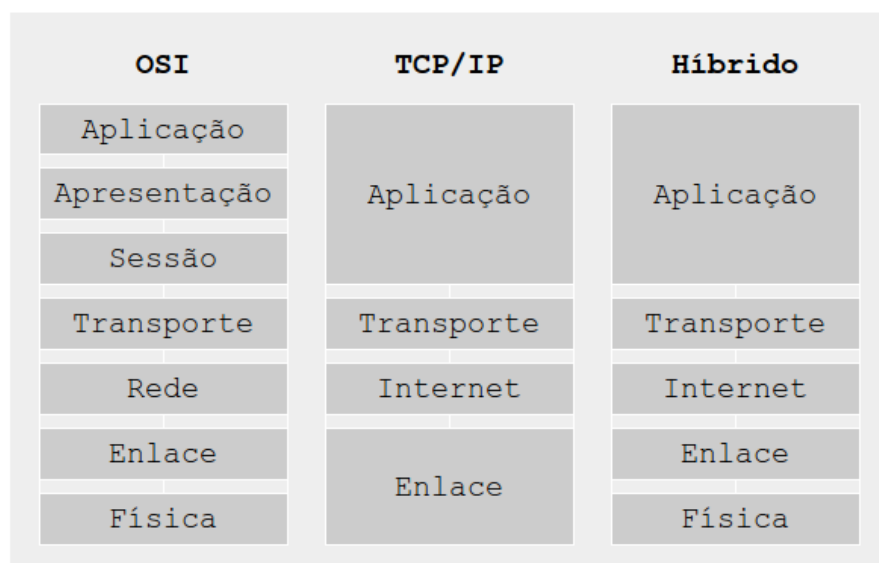


Figura 5 – Relação entre o modelo de referencia OSI, o modelo de protocolos TCP/IP e o modelo híbrido

Neste trabalho iremos focar no modelo TCP/IP oficial, de quatro camadas, e seus protocolos, os quais serão apresentados nas subseções seguintes: 2.2.1 Aplicação, 2.2.2 Transporte, 2.2.3 Internet e 2.2.4 Enlace.

## 2.1 Aplicação

Está é primeira camada da pilha TCP/IP, definida separadamente no RFC 1123 (IETF, 1989a). Ela mantém protocolos necessários e utilizados pelos usuários e serviços. Em suma, aplicações de redes executadas em um sistema final tem o objetivo de se comunicar com outras aplicações em outro sistema conectado pela rede (KUROSE; ROSS, 2013).

Os principais protocolos pertencentes à Aplicação, apresentados por (STALLINGS, 2007), são:

- Telnet: possibilita o usuário acessar uma máquina, através de outro sistema.
- FTP (*File Transfer Protocol*): utilizado para transferir arquivos entre sistemas.
- SMTP (*Simple Mail Transfer Protocol*): provê transporte de emails entre diferentes hosts.
- HTTP (*HyperText Transfer Protocol*): utilizado para a comunicação entre um sistema final e um servidor Web.
- DNS (*Domain Name System*): serviço de diretório que traduz nomes de hospedeiro para endereço IP.

Além de seus serviços é importante destacar as arquiteturas da camada de Aplicação. Kurose e Ross (2013) apresenta duas arquiteturas, as quais afirmam serem as mais comumente utilizadas hoje em dia, a cliente/servidor e a P2P (*Peer to Peer*).

O autor explica que a arquitetura cliente/servidor é configurada em um sistemas finais: primeiramente é necessário que um host, chamado servidor, ofereça serviço através de uma rede, o qual será acessado por um ou mais host(s), denominado(s) cliente(s), que enviará uma requisição ao servidor e espera deste uma resposta. Para que isto seja possível é necessário que o hospedeiro que atua como servidor esteja sempre disponível, além de possuir um IP conhecido, ou o seu nome (neste caso o protocolo DNS precisa fazer a tradução).

A arquitetura P2P, é um pouco diferente, o mesmo autor explica que nesta configuração dois sistemas finais utilizam comunicação direta, sem a definição estrita de quem é cliente e quem é o servidor, sendo assim os hosts podem desempenhar ambas funções, de forma intercalada. Adicionando à estas arquiteturas, temos uma configuração híbrida, que utiliza inicialmente o cliente/servidor para estabelecer uma nova conexão entre diferentes hosts, que passam a se comunicar utilizando a arquitetura P2P.

## 2.2 Transporte

De acordo com [Tanenbaum e Wetherall \(2011\)](#), o objetivo desta camada é prover serviço de comunicação entre processos de aplicações, em diferentes sistemas finais, ou seja ampliar o serviço de entrega IP ( que será apresentado na próxima subseção) para um serviço de entrega entre dois processos que rodam nestes sistemas. Este tipo de comunicação é chamada transferência fim-a-fim.

O pacote de dados, PDU, pertencente à esta camada é chamado "seguimento", o qual contém informações referentes à camada de aplicação, e após ter inserido informações adicionais, como aplicação origem e destino e checksum (código utilizado para a verificação da integridade do pacote), a PDU é passada para camada seguinte, a de Rede.

A Internet possui dois protocolos distintos responsáveis pelo serviço de transporte, o TCP (*Transmission Control Protocol*) e o UDP (*User Datagram Protocol*). A principal diferença entre os protocolos citados é em relação à sua confiabilidade. O protocolo UDP fornece à camada de aplicação um serviço não confiável e orientado à mensagem, ao contrário do TCP, o qual garante uma transmissão confiável dos dados, utilizando para isso um serviço orientado à conexão, ([COMER, 2006](#)).

As principais características do protocolo TCP são apresentadas por [Kurose e Ross \(2013\)](#), em seu texto eles explicam que para garantir uma entrega de dados correta e em ordem, entre processos remetente e destinatário, o protocolo TCP utiliza o controle de fluxo, números de sequência, reconhecimentos e temporizadores. Além de oferecer um serviço de controle de congestionamento, o qual limita a taxa de envio de tráfego do remetente para rede, permitindo que conexões TCP's compartilhem igualmente a largura de banda em um enlace de dados congestionado.

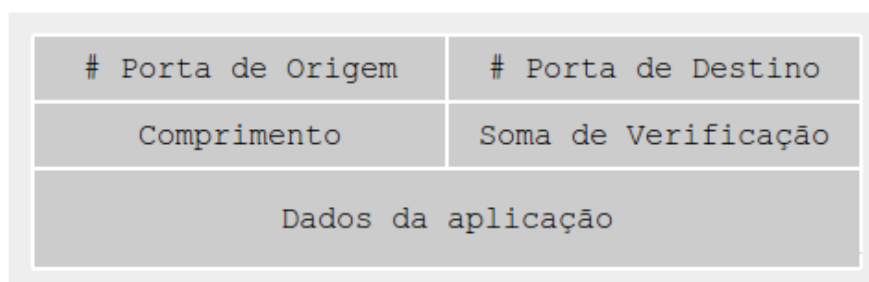


Figura 7 – Estrutura da PDU do protocolo UDP

Fonte: (KUROSE; ROSS, 2013)

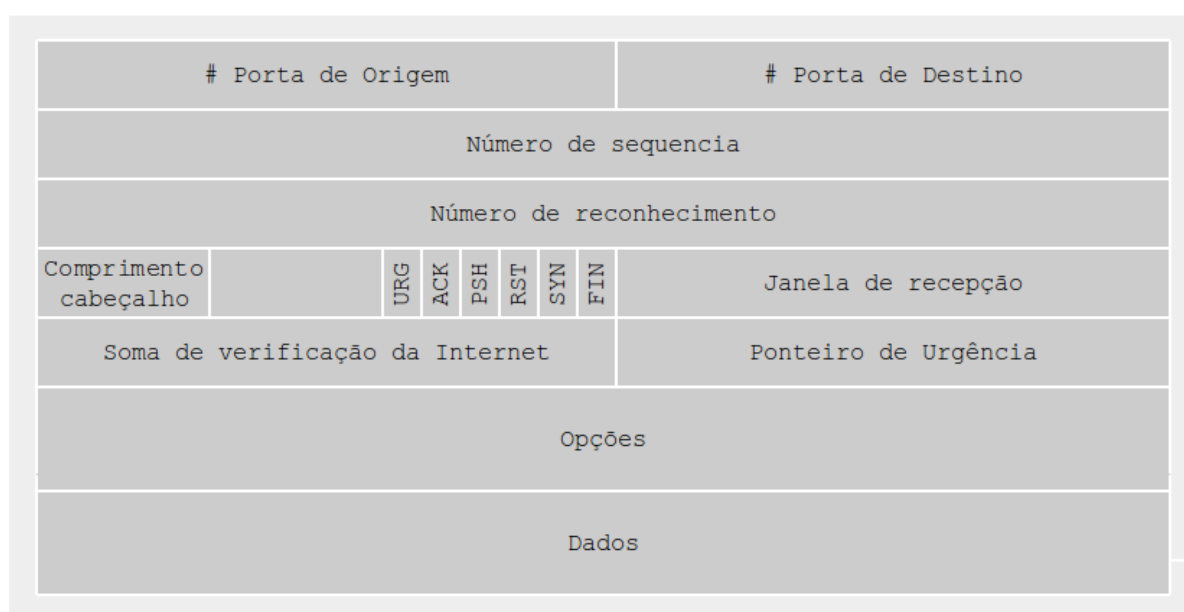


Figura 6 – Estrutura da PDU do protocolo TCP

Fonte: (KUROSE; ROSS, 2013)

O mesmo autor afirma que o protocolo UDP (IETF, 1980), entretanto, faz o mínimo necessário para a entrega de dados de uma aplicação à outra, este adiciona aos dados da camada de Aplicação apenas informações básicas para o encapsulamento da PDU em um segmento (chamado Datagrama). É possível reparar a diferença quanto a simplicidade da estrutura do segmento do protocolo UDP, apresentado na Figura 7, em comparação com a estrutura do segmento do protocolo TCP, mostrado na Figura 6.

As diferenças ressaltadas entre os protocolos TCP e UDP os tornam desejáveis para diferentes objetivos. Para conexões que são capazes de suportar perda de dados, como telefonia por Internet, ou necessitam de velocidade, como o protocolo DNS, é preferível a utilização do protocolo UDP. Já para os serviços que não suportam perdas e necessitam de confiabilidade, como HTTP, FTP e SMTP, o protocolo TCP é necessário.

Além dos protocolos citados, que são os mais conhecidos, [Forouzan e Fegan \(2008\)](#) citam outro protocolo pertencente a esta camada, o SCTP (*Stream Control Transmission Protocol*), definido no RFC 4960 ([IETF, 2007](#)), que combina os melhores recursos dos protocolos TCP e UDP, assim ele provê um serviço confiável e orientado à mensagem.

## 2.3 Rede

Esta camada, também é chamada de internet (minúsculo para referenciar uma conexão de redes genérica), recebe o segmento enviado pela camada superior, de Transporte, juntamente com uma identificação do host para qual deve ser enviado este pacote, que então é encapsulado em um pacote IP (Figura 8) antes de ser repassado para camada seguinte ([COMER, 2006](#)). Todos os pacotes transportados pela rede devem possuir o endereço de destino completo, contendo o número IP, pois estes serão independentemente transportados.

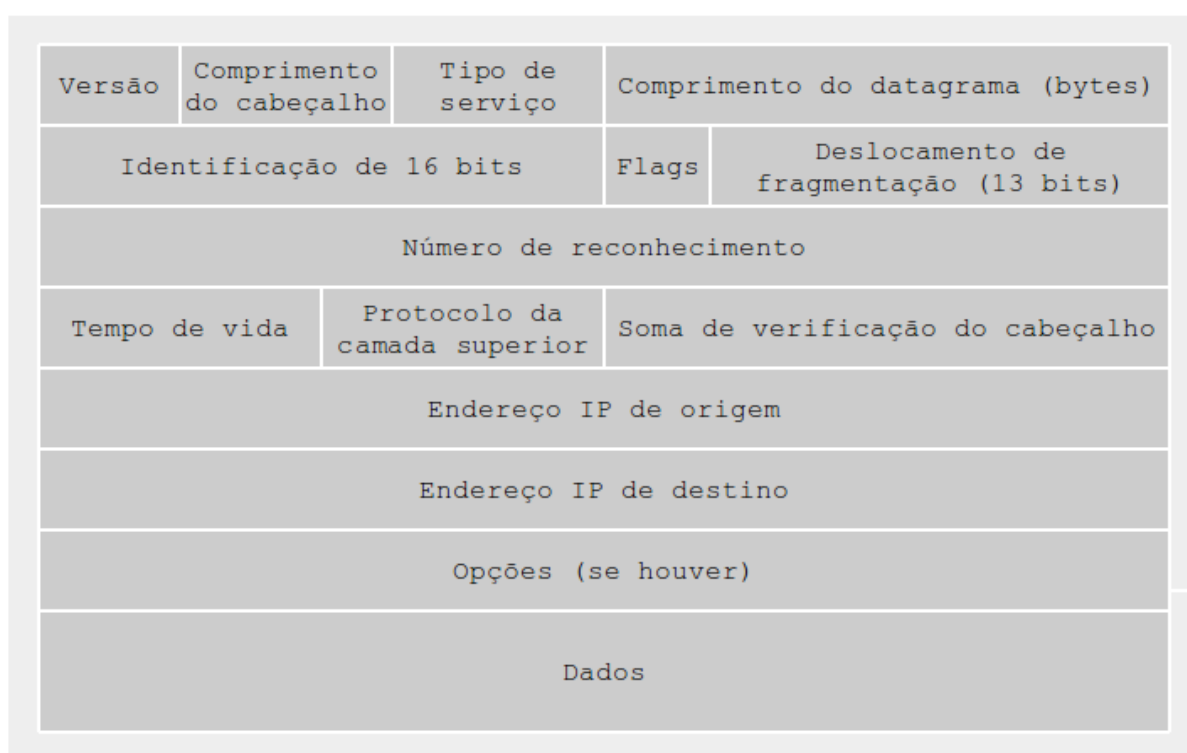


Figura 8 – Estrutura do datagrama IP

Fonte: ([KUROSE; ROSS, 2013](#))

O protocolo que define como a entrega desses pacotes deve ser feita é chamado de *Internet Protocol*, o IP. O tipo de serviço oferecido pelo IP é chamado de serviço de melhor esforço, o que de acordo com [Kurose e Ross \(2013\)](#) não garante temporização preservada entre pacotes, recebimento destes em ordem no destinatário e nem garantia da entrega final.



Sendo assim, a principal função desta camada, de acordo com [Tanenbaum e Wetherall \(2011\)](#), é o rotear pacotes da máquina origem para a máquina destino. Para isso são utilizados roteadores, que encaminham os pacotes recebido para o enlace de saída apropriado.

## 2.4 Enlace

De acordo com [Tanenbaum e Wetherall \(2011\)](#) esta camada é mais propriamente definida como uma interface entre os hosts e os enlaces de transmissão (canais de comunicação que conectam nós adjacentes, sendo que nós refere-se à qualquer tipo de dispositivo que rode um protocolo da camada de enlace). Ela define o que os enlaces devem fazer para cumprir os requisitos de interconexão com o serviço não orientado a conexão oferecido pela camada superior, o IP.

[Kurose e Ross \(2013\)](#) definem dois tipos de enlaces: no primeiro tipo, de difusão (*broadcast e multicast*) os quadros (nomenclatura referente à PDU desta camada) são enviados de um host para múltiplos receptores em um só canal de comunicação (para que esta lógica seja possível é necessário um protocolo de acesso ao meio), os segundo tipo é o enlace de comunicação ponto-a-ponto (também chamado *unicast*), um único remetente e um único receptor em cada extremidade do enlace.

Os quadros são enviados pela rede em um fluxo de bits correspondentes, o que faz necessário o tratamento de erros de transmissão e regulamento o fluxo de dados para minimizar perdas. O padrão de transmissão de dados mais comumente utilizado é o Ethernet para redes locais fisicamente conectadas, também conhecido com IEEE 802.3.

Outro importante protocolo que compõe esta camada é denominado ARP (*Address Resolution Protocol*), RFC 826 ([IETF, 1982](#)), responsável por obter o endereço físico, também chamado endereço MAC (*Media Access Control*), a partir do endereço IP passado pela camada de Rede acima, permitindo assim a transmissão do quadro para seu destino.

## 3 Aplicações

Na própria Internet estão disponíveis inúmeras aplicações que têm como foco redes e protocolos. Sobressaem-se simuladores de redes devido á sua funcionalidade tanto comercial quanto didática: softwares com este objetivo são de interesse para grandes empresas, pois possibilitam projetar e testar uma rede quanto ao seu desempenho, fluxo de dados, possíveis gargalos, perda de informações, segurança e etc., sem a necessidade de montar uma laboratório de testes, por exemplo, ou criar uma rede sem planejamento a qual pode vir a apresentar os problemas já citados, opções estas muito mais dispendiosas.

Três destes simuladores são mais conhecidos: o *Network Simulator*<sup>2</sup> (NS3) um software aberto, voltado especialmente para pesquisa e uso educacional; o *Cisco Packet Tracer*<sup>3</sup> desenvolvido pela Cisco®, que permite a simulação de uma rede utilizando diferentes equipamentos da marca (seu objetivo didático inclui o treinamento para certificação oferecidas pela própria empresa); e o GNS3<sup>4</sup> que além do software de simulação mantém uma comunidade online para discussões e troca de ideias para seus usuários.

Além dos simuladores existem outras aplicações interessantes, conhecidas como *sniffers*, que monitoram e analisam o tráfego de rede, ou seja, os pacotes enviados e recebidos em uma conexão. O mais conhecido destes é o *Wireshark*<sup>5</sup>, um analisador de protocolos utilizado para navegação interativa do tráfego de rede em tempo real. Uma ferramenta mais específica neste campo, utilizada para análise de tráfego da camada de Aplicação, é disponibilizada pelos browsers e conhecida como "Developers Tools". Essa ferramenta de depuração possibilita o acesso a informações internas do browser e sua aplicação web através do painel "Network"q

---

<sup>2</sup> <https://www.nsnam.org/>

<sup>3</sup> <http://www.packettracernetwork.com/>

<sup>4</sup> <https://www.gns3.com/>

<sup>5</sup> <https://www.wireshark.org/>

### 3 Trabalhos Relacionados

Em 2012 [Emiroglu e Sahin \(2013\)](#) apresentaram um estudo sobre a qualidade do aprendizado na disciplina de Rede de Computadores no curso de graduação da universidade de Baskent. A análise foi feita no decorrer de três anos, de 2007 a 2010, e constatou que a utilização de ferramentas como simuladores e aplicativos, além de animações, melhoraram resultados obtidos pelos alunos. Foram feitas análises semanais sobre a performance dos estudantes no laboratório do curso e estas constataram que os alunos aprendem melhor quando envolvidos na aplicação prática da teoria do estudo de Redes de Computadores. Além disso questionários aplicados ao final de cada do curso constataram que os alunos estavam felizes em serem envolvidos na aplicação prática dos tópicos abordados pela disciplina durante as aulas de laboratório.

Um simulador com este objetivo, educacional, foi desenvolvido por [Poletti \(2013\)](#), tendo como base um dos softwares citando anteriormente, o *Cisco Packet Tracer*, em seu trabalho de conclusão de curso. O simulador desenvolvido por ele facilita a visualização das estruturas dos principais protocolos do modelo TCP/IP, sem a necessidade, que havia na ferramenta Cisco, de configurar uma rede virtual para poder simular o envio destes. Além disso a implementação permite interação direta do usuário com a estrutura dos protocolos, possibilitando mudanças da maioria dos valores de seus campos, e desta forma visualizar o impacto que tais mudanças causam na comunicação (o que era muito limitado na ferramenta original onde o conteúdo dos pacotes são estáticos). O novo simulador ainda possui o diferencial de ser uma ferramenta distribuída, o que permite a interação entre vários simuladores distintos.

Também com objetivo de ser didático [Lee, Kim e Moon \(2014\)](#) desenvolveram um framework orientado à eventos chamado KENVSv2. Eles propõem a utilização deste à estudantes de Ciência da Computação, que devem implementar os drivers TCP e IP e protocolos de roteamento. A arquitetura proposta para o trabalho pode ser vista na Figura 9: o host virtual age como uma camada de aplicação para o TCP e como link de dados combinados e camada física para o IP. Além de possuir um par em funcionamento para a realização de testes de comunicação. De acordo com seu artigo, eles acreditam que para o uma experiência completa de aprendizado os alunos devem ser capaz de implementar e testar toda a pilha de protocolos.

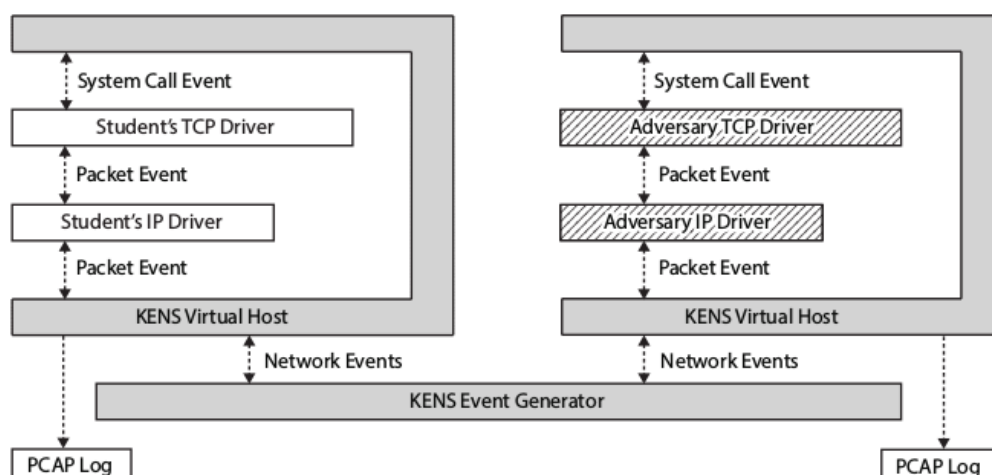


Figura 9 – Funcionamento em camadas

Fonte: (LEE; KIM; MOON, 2014)

## 4 Metodologia

Com o objetivo de criar aplicações que simulem cada uma das camadas do modelo TCP/IP, foram escolhidos cinco protocolos para serem implementados. Da camada de aplicação o protocolo HTTP, utilizado para comunicação de um sistema final e um servidor Web, permitindo assim navegação pela Internet, será implementado. Da camada seguinte, a de Transporte, devido a sua importância, os dois principais protocolos serão implementados, o protocolo de transporte confiável, TCP, e o não confiável, UDP.

Os seguimentos recebidos da camada de transporte deverão ser tratados na camada de internet pelo protocolo IP fazendo uso do roteamento o qual, por sua vez, deverá entregar o pacote resultante para camada de Enlace. Esta terá o protocolo Ethernet simulado, complementado do protocolo ARP para tradução do endereço IP para MAC, possibilitando assim o envio do quadro para o host destino.

A Figura 10 mostra a formulação implementada neste trabalho, representando o funcionamento da pilha de protocolos quando uma aplicação cliente, no caso o browser, faz uma requisição HTTP ao servidor. A arquitetura cliente/servidor foi utilizada para promover a interação entre as camadas, assim elas conversam entre si através de sockets, onde as portas são representadas pelo caminho aberto em cada camada e as mensagens são representadas pelas setas: vermelha ilustrando a mensagem de requisição, e azul a resposta mandada pelo servidor.

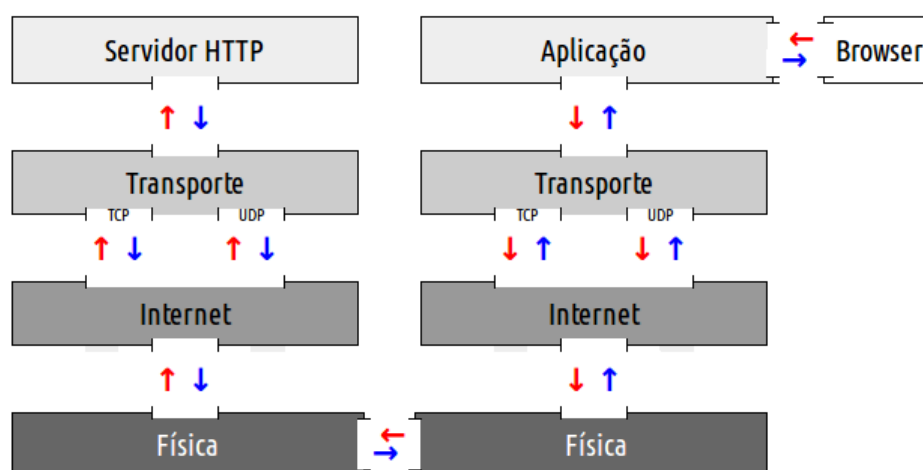


Figura 10 – Esquema da arquitetura desenvolvida

Cada camada foi implementada de forma independente, e a cada etapa os dados referentes à PDU são exibidos para, assim, manter a transparência objetivada para a utilização deste sistema como ferramenta de ensino. Estes dados são estruturados de

acordo com os respectivos padrões definidos pela RFC.

Objetivando a melhor forma de implementação, considerando a simplicidade e bibliotecas disponíveis para acesso baixo nível á rede, a linguagem escolhida foi o Python, versão 2.7. O planejamento teve como objetivo permitir a interação entre camadas e testes a cada etapa do desenvolvimento. Desta forma o primeiro módulo desenvolvido foi referente a camada Física para garantir a transmissão de mensagens entre hosts. Utilizando este módulo como base foram desenvolvidos, na seguinte ordem, os módulos simuladores referentes á camada de Aplicação, Transporte e Internet.

## 5 Roteiro de funcionamento

Com o objetivo principal de figurar uma ferramenta educacional, o acesso aos dados, o reconhecimento das etapas de funcionamento e troca de mensagens entre as camadas foram os principais focos do desenvolvimento. Tendo em mente tal premissa a execução da ferramenta segue exatamente as mesmas etapas da troca de mensagens que ocorre no modelo de protocolos TCP/IP.

A execução é iniciada com uma requisição http feita a partir de um browser, a camada responsável pela Aplicação recebe essa mensagem faz o reconhecimento de seu conteúdo e repassa para camada seguinte de Transporte. Os dados da referentes a uma requisição HTTP incluem o método, identificação do cliente, versão http, o host para o qual está se fazendo a requisição e características do cliente.

Os principais métodos de requisição HTTP são apresentado, sucintamente na Tabela 1, e como exemplo segue uma solicitação gerada em testes a qual é apresentada ao usuário.

Método	Descrição
GET	Lê uma pagina Web
HEAD	Lê um cabeçalho de página Web
POST	Acrescenta algo a uma página Web
PUT	Armazena uma página Web
DELETE	Remove a página Web
TRACE	Ecoa a solicitação recebida
CONNECT	Conecta através de um proxy
OPTIONS	Consulta opções para uma página

Tabela 1 – Métodos de requisição HTTP  
([TANENBAUM; WETHERALL, 2011](#))

GET /test.html HTTP/1.1

Host: localhost:1111

Connection: keep-alive

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko)

Chrome/54.0.2840.71 Safari/537.36

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8

Accept-Encoding: gzip, deflate, sdch, br

Accept-Language: en-US,en;q=0.8,pt;q=0.6,pt-BR;q=0.4

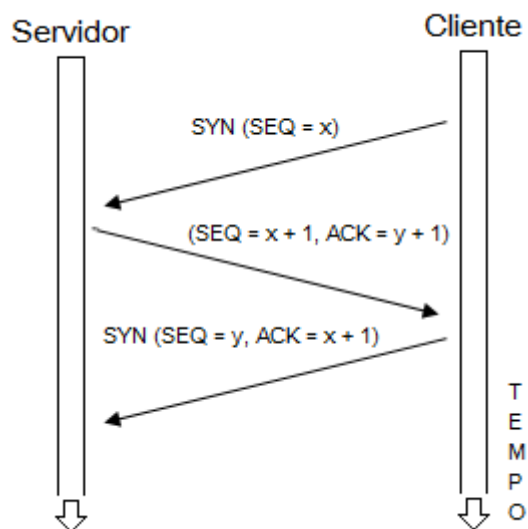


Figura 11 – Three Way Handshake - Estabelecimento de conexão TCP  
([TANENBAUM; WETHERALL, 2011](#))

Ao receber o pacote de requisição da camada superior, na camada de Transporte o usuário tem a opção entre dois tipos de protocolos o UDP e o TCP. Caso a escolha do usuário seja UDP é adicionado ao seguimento apenas a porta de origem e destino, o comprimento do UDP, incluindo cabeçalho e o *checksum* para verificação.

Caso a escolha do usuário seja utilizar o protocolo de transporte TCP, além dos dados reunidos pelo protocolo UDP, é adicionado ao cabeçalho o número de sequência, o número de confirmação, o tamanho da janela, ponteiro para urgente, opções e as flags. Ao todo são oito *flags*:

- CWR e ECE: utilizadas para sinalizar congestionamento.
- URG: indicador para o caso de uso do ponteiro de urgência.
- ACK: verifica se o número de confirmação é válido.
- PSH: declara o modo *push* solicitando ao que o receptor a entregar os dados à aplicação ao invés de armazena-los até que um *buffer* completo tenha sido recebido.
- RST: utilizado para reiniciar uma conexão.
- SYN: usado para estabelecer conexões
- FIN: encerra uma conexão

Duas dessas flags são de especial importância pois são responsáveis pelo estabelecimento da conexão entre o cliente e o servidor, o que garante a confiabilidade da conversação. As flags SYN e ACK são utilizadas pelo procedimento conhecido como *Tree Way Handshake*, representado na Figura 11.

Quando um cliente deseja estabelecer com um servidor, este envia um seguimento com a flag SYN igual a 1, o que será reconhecido pelo servidor como um pedido para o estabelecimento de uma conexão e se o cliente receber um seguimento como resposta



contendo as flags SYN e ACK iguais a 1 indica que a requisição foi aceita pelo servidor. O cliente envia, então, por último um seguimento com a flag ACK igual a 1, indicando o reconhecimento da resposta.

Em ambos os casos as PDU's são apresentadas ao usuário e, no caso do protocolo TCP, os passos realizados pelo *three way handshake* também são reportados ao serem executados.

```

*****
Porta de Origem = 2222          |          Porta de Destino = 9999
*****
Comprimento do UDP = 2222      |          Checksum do UDP = 9999
*****
                                |
                                |          Dados = Requisicao HTTP
                                |
*****

```

(a) Protocolo UDP

```

*****
Porta de Origem = 2222          |          Porta de Destino = 9999
*****
                                |
                                |          Numero de sequencia = 454
                                |
                                |          Numero de confirmacao = 0
                                |
Comprimento do cabeçalho      |          | C | E | U | A | P | R | S | F |          Tamanho
=                               |          | W | C | R | C | S | S | Y | I |          da
128                            |          | R | E | G | K | H | T | N | N |          Janela
                               |          | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
*****
Checksum = 0                   |          Ponteiro para urg.
*****
                                |
                                |          Opcoes
                                |
*****
                                |
                                |          Dados = Requisicao HTTP
                                |
*****

```

(b) Protocolo TCP

Figura 12 – Resultados obtidos na execução do módulo de simulação referente a camada de Transporte

## Referências

- CERF, V. G.; KAHN, R. E. A protocol for packet network intercommunication. **IEEE Transactions on Communication**, 1974. Disponível em: <<https://www.cs.princeton.edu/courses/archive/fall06/cos561/papers/cerf74.pdf>>. Citado na página 1.
- COMER, D. E. **A interligação de redes com TCP/IP**. 5. ed. Rio de Janeiro: Elsevier, 2006. Citado 5 vezes nas páginas 1, 4, 5, 8 e 10.
- EMIROGLU, B. G.; SAHIN, S. Analysis of students' performances during lab sessions of computer networks course. **Educational Technology & Society**, n. 16 (3), p. 329–346, 2013. Citado na página 13.
- FOROUZAN, B. A.; FEGAN, S. C. **Protocolo TCP/IP**. 3. ed. São Paulo: McGraw-Hill, 2008. Citado 6 vezes nas páginas 1, 2, 4, 5, 6 e 10.
- IETF. **User Datagram Protocol**. 1980. Disponível em: <<https://tools.ietf.org/html/rfc768>>. Acesso em: 23 de março de 2016. Citado na página 9.
- IETF. **An Ethernet Address Resolution Protocol**. 1982. Disponível em: <<https://tools.ietf.org/html/rfc826>>. Citado na página 11.
- IETF. **Requirements for Internet Hosts. Application and Support**. 1989. Disponível em: <<https://tools.ietf.org/html/rfc1123>>. Citado 2 vezes nas páginas 4 e 7.
- IETF. **Requirements for Internet Hosts. Communication Layers**. 1989. Disponível em: <<https://tools.ietf.org/html/rfc1122>>. Citado na página 4.
- IETF. **A TCP/IP Tutorial**. 1991. Disponível em: <<https://tools.ietf.org/html/rfc1180>>. Citado na página 2.
- IETF. **Stream Control Transmission Protocol**. 2007. Disponível em: <<https://tools.ietf.org/html/rfc4960>>. Citado na página 10.
- KUROSE, J. F.; ROSS, K. W. **Redes de Computadores e a Internet. Uma abordagem top-down**. 6. ed. São Paulo: Pearson, 2013. Citado 6 vezes nas páginas 1, 7, 8, 9, 10 e 11.
- LEE, K.; KIM, J.; MOON, S. An educational networking framework for full layer implementation and testing. 2014. Citado 2 vezes nas páginas 13 e 14.
- POLETTI, C. **Aprimoramento de um simulador didático de redes de computadores**. 2013. Disponível em: <<https://intranet.univates.br/bdu/bitstream/10737/381/1/CristianoPoletti.pdf>>. Citado na página 13.
- STALLINGS, W. **Data and Computer Communication**. 8. ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2007. Citado 3 vezes nas páginas 1, 4 e 7.
- TANENBAUM, A. S.; WETHERALL, D. **Redes de computadores**. 5. ed. São Paulo: Pearson Prentice Hall, 2011. Citado 5 vezes nas páginas 6, 8, 11, 17 e 18.