# INTRODUÇAO

# MODELO OSI

O modelo OSI foi desenvolvido pela *Internacional Standards Organization (ISO)* como proposta para padronização dos protocolos aplicados as diversas camadas em redes pois tem como objetivo a interconexão de sistemas que sejam abertos à comunicação com outros sistemas, estabelecendo um padrão bem definido de regras onde dispositivos de diferentes fabricantes possam independentemente de seu sistema operacional, tipo e sua finalidade estabelecer uma conexão e comunicar-se de forma igualitária, para isso sendo apenas necessária a existência de uma conexão de rede e o padrão devidamente implementado.

O modelo OSI propriamente dito não trata-se de uma arquitetura de rede na qual dita especificamente quais serviços e protocolos devem ser utilizados em cada uma das camadas, pelo contrário, o modelo trata apenas de como deve ser feita está comunicação entre cada uma das camadas existentes e o que cada uma delas deve fazer. No entanto, a *Internacional Standards Organization* produziu também padrões para cada uma das camadas do modelo OSI, cada padrão foi publicado como um padrão internacional distinto. (TANENBAUM, 2003).

O modelo OSI é tido como referencia em interconexão de dispositivos e é utilizado como base para criação de novas camadas nas quais são criadas quando há necessidade de outro grau de abstração, estas por exemplo, contendo regras específicas para a implementação de uma aplicação. O modelo de referencia OSI é constituído por cinto camadas, a sua ordem inicia-se pela camada inferior. A primeira camada é a física, seguida da camada de enlace, camada de rede, camada de transporte, camada de sessão e a quinta e última, camada de apresentação.

# PROTOCOLOS DE REDES DE COMPUTADORES

Uma rede de computadores é a conexão entre dispositivos que podem estar em diferentes espaços, lugares, mas que encontram-se interligados por um meio físico ~~trata-se~~ . Trata-se de uma combinação de hardware e software que estabelece uma comunicação e envia dados entre entidades em diferentes sistemas operacionais, uma entidade pode ser qualquer dispositivo capaz de enviar ou receber estes dados (FOROUZAN, 2007).

Entidades em uma rede não podem simplesmente enviar fluxos de dados entre elas e esperarem que sejam compreendidas, por este motivo é necessário que exista uma regra, um protocolo, no qual estas entidades devem concordar para que seja possível estabelecer uma comunicação de forma em que todas envolvidas nesta interconexão sejam capazes de compreenderem uma as outras. O protocolo de rede é o meio lógico para viabilizar a conexão, nele ~~contem~~ contém definições de como esta comunicação deve funcionar, especifica como as entidades devem tratar o fluxo de dados.

Uma comunicação entre dispositivos em uma rede de computadores só acontece quando tais dispositivos, entidades, possuem implementados os protocolos que possibilitam essa compreensão, nas quais determinam como a máquina ou programas devem tratar os dados, preparando-os para serem enviados para o estágio seguinte do processo de comunicação, também como devem ser comunicados. Existem três elementos-chave em um protocolo, são eles a sintaxe, semântica e *timing*.

Há diversos tipos de protocolos utilizados para comunicação em redes de computadores e equipamentos eletrônicos. Todos dispositivos com capacidade a conexão a uma rede utiliza-se de um determinado tipo de protocolo de comunicação para que possa se comunicar com o restante da rede a qual está conectado.

As implementações de protocolos costumam ser complicadas e por este motivo existem pesquisas com o objetivo de descobrir técnicas matemáticas formais para a especificação e verificação, tanto na camada de enlace de dados como para as demais camadas do conjunto. (TANENBAUM, 2003).

Um protocolo específico de é implementado para tratar dados entre uma camada e outra, criando uma comunicação entre as camadas possibilitando a transferência dos dados recebidos desde a camada mais inferior até a mais superior, assim podemos considerar que a concepção de um protocolo tem como base a resolução dos dados da sua camada específica para a próxima camada em que mantém comunicação. O modelo OSI por exemplo faz a utilização de protocolos específicos para cada uma de suas camadas, estes protocolos tem responsabilidades diferentes conforme a camada em que estão implementados, por exemplo, no modelo de referência OSI tem como camada inicial a camada física na qual é responsável pela movimentação de bits individuais de um *hop* para um seguinte, esta define

Um protocolo é implementado dentro de um conceito de camadas no qual é necessária a compreensão do processo que irá desempenhar. Para a verificação de protocolos e também a sua implementação um outro um conceito fundamental é utilizado, o conceito de máquina de estados finitos.

# FERRAMENTAS DE SIMULAÇAO DE REDES

Para o desenvolvimento do trabalho foram analisadas as ferramentas para simulação de ambientes de redes: GNS3 (Graphical Network Simulator, 2015), OPNET (Optimized Network Engineering Tools, 2011) e NS2 (Network Simulator 2, 2011). Essas ferramentas são utilizadas para a definição lógica de uma estrutura de redes de computadores e para simulação de tráfego de dados nessas redes. A partir da simulação, é possível realizar análises a respeito dos modelos de rede e seu funcionamento após a sua implementação (Isso daqui estará na justificativa).

A escolha para a utilização do software que mais se adeque ao trabalho foi realizada através de pesquisas com resultados e trabalhos correlacionados como (Pg 66) ou (Pg 72) (que trabalhos?) sobre verificação e validação de protocolos com a utilização de softwares de simulação de redes. Dessa forma foi levada em consideração características como tipo de licença, aplicabilidade, portabilidade e também se é gerado um arquivo de saída que possa ser utilizado para fins de analise das simulações realizadas.

O software GNS3 é um simulador *front-end* gráfico de licença gratuita que permite gerenciar simulações de redes complexas, para processadores MIPS, PowerPC e X86. A partir do processador selecionado é possível executar imagens de sistemas operacionais destinadas a roteadores, como a IOS (*Internetwork Operation System*) da fabricante Cisco, dos roteadores da série 1700, 2600, 3600, e 7200 e também possibilitando a execução de imagens destinadas ao *firewall* Cisco PIX e do JunOS destinadas aos roteadores Juniper.

A simulação pelo GNS3 possibilita a configuração dos dispositivos suportados, a ligação entre eles definindo os nós, seus enlaces, tráfego de conexão, os protocolos utilizados, isso sendo realizado de maneira idêntica aos dispositivos suportados pelo software.

Devido a possibilidade de simular roteadores e a execução de sua firmware a gama de opções para simulações de topologia de redes e protocolos proprietário e livre é ampla, o sistema é descomplicado tratando-se de uma interface simples e funcional, opção esta para simulações de redes onde não exista a necessidade de verificação de dados posteriores dos quais o GNS3 não gera, não possibilita a criação de um arquivo de saída no qual contenha informações da simulação realizada, informações estas necessárias para validação de protocolos de rede e possível análise através de outro software que receba o arquivo como entrada. Devido a ausência da opção de gerar um arquivo de saída que contenha os dados da simulação o uso do software GNS3 logo foi descartado.

O software OPNET é um simulador comercial, largamente utilizado em âmbito corporativo, devido a sua precisão nos resultados e também pela utilização fácil e descomplicada. Focada em uma interface gráfica simples que facilita o entendimento em tarefas de maior complexidade como a configuração de cenários de rede e a visualização de resultados, o software é de licença proprietária mas conta também com uma versão educacional.

O OPNET conta com a possibilidade de geração de um arquivo de saída, mas quando comparado ao NS2 algumas desvantagens são destacadas como sua licença, e a não portabilidade do software, pontos que definiram a escolha pelo software NS2, este trata-se de um software *open source* de código aberto, livre para modificações e dá suporte a múltiplas plataformas como Unix e sistemas baseados em Unix, FreeBSD, Solaris e Windows.

O NS2 é um simulador de eventos discreto que em como objetivo a criação de simulações baseadas nos protocolos de rede, foi construído através da utilização de duas linguagens, C++ em sua estrutura básica e Otcl para uso como *frontend.* O projeto no qual se deu origem ao software começou no ano de 1989 e surgiu inicialmente de um projeto concebido com título VINT (*Virtual InterNetwork Testbed*), seu código fonte é aberto e oferece suporte a um número significativo de tecnologias de rede, sua aplicação pode ser realizada em diferentes cenários baseados nos protocolos TCP e UDP, dentre outros compõem esse projeto a DARPA, USC/ISI, Xerox PARC, LBNL e a Universidade de Berkeley.A linguagem Otcl é interpretada, desenvolvida pelo MIT, é com esta linguagem que as simulações são escritas.

O NS2 possibilita ao usuário um maior controle do projeto a se desenvolver pois toda sua estrutura é documentada e de codificação aberta tornando possível a criação de frameworks, ferramentas que façam integração e complementem o sistema quando necessário para o projeto no qual irá se aplicar. A noção de tempo no NS2 é obtida através de unidades de simulação que podem ser associadas, para efeitos didáticos, a segundos. A rede é construída por nós nos quais se conectam através de enlaces, os eventos ocorridos são escalonados para que possam passar entre os nós através dos enlaces estabelecidos, os nós e enlaces podem ter propriedades associadas, agentes podem ser associados aos nós e estes são responsáveis pela geração de diferentes pacotes. A fonte de tráfego, a criadora dos dados a serem enviados pelo emissor, é uma aplicação a qual é associado um agente particular, os agentes necessitam de um receptor que receberá seus pacotes possibilitando a comunicação entre os nós, um agente TCP por exemplo, orientado a conexão, o receptor chama-se SINK e tem como tarefa a geração de pacotes de reconhecimento chamados de ACK – Acknowledge, no caso de uma agente UDP, não orientado a conexão, o receptor chama-se NULL.

A execução de uma simulação gera um arquivo de saída em formato texto chamado de *trace* que contem os dados capturados pela simulação. Este arquivo pode ser utilizado em outros softwares como dados de entrada para um detalhamento da mesma ou leitura para representação gráfica da simulação realizada. O processo de criação de uma simulação utilizando-se do NS2 é uma sequencia bem defina de processos escrita em um *script* de formato otcl, este arquivo em si é a simulação, arquitetura, na qual o software irá rodar e gerar seus respectivos dados.

No *script o*tcl a sequencia de processos definidos são a criação de um objeto simulador (escalonador de eventos), em seguida define-se a escrita para a abertura de arquivos para análise posterior chamados de arquivos *trace*, então inicia-se a criação da topologia, arquitetura, dos nós e seus enlaces, definindo o tipo de ligação que estes nós terão entre eles, a criação dos agentes da camada de transporte e conexão com os nós, após é necessário estruturar a topologia da rede, suas ligações e agentes definido os geradores de tráfego e conexão com os agentes da camada de transporte, a partir dai é realizada a programação dos eventos da simulação e finalizando com o fechamento da simulação e a possível geração de estatísticas. A sintaxe do *script* otcl para a geração de simulações no NS2 é especifica do software e requer conhecimento desta linguagem.

O NS2 conta com ferramentas de auxilio para análise dos dados extraídos com por exemplo a ferramenta Nam, na qual é responsável por fazer a leitura do arquivo Nam-trace e representar a topologia e comportamento de tráfego graficamente, sendo possível compreender melhor os nós, seus enlaces e como os dados estão sendo manipulados possibilitando a identificação de gargalos e perda de dados na rede. A ferramenta Nam é muito importante para uma análise detalhada e de fácil compreensão pois possibilita a visão geral de toda a estrutura planejada, visualizando o andamento da simulação, identificando além dos fluxos a formação de filas e o descarte de pacotes.

Há também a ferramenta gráfica Xgraph e Gnuplot nas quais, assim como a ferramenta Nam são voltadas para análise gráfica do arquivo *trace*. Estas ferramentas possibilitam a geração de gráficos para verificação de vazão de cada um dos fluxos utilizados na simulação, contribuem para análise das estatísticas do NS2, contam com uma opção de plotagem de gráficos e visualização interativa para gerar a animação. Trabalham com arquivos colunados, utilizando-se da estrutura de linhas e colunas x e y.

O Gnuplot é uma ferramenta de software de domínio público destinado a construção de gráficos e superfícies, tem como uma de suas características fortes a possibilidade de execução de um arquivo *script* em diferentes sistemas operacionais*.*

Diferentes protocolos podem ser implementados nas simulações utilizando o NS2, como protocolos da camada de rede que nos permite definir regras de roteamento, toda transferência de pacotes entre um nó emissor e um nó receptor faz uso da definição de qual o caminho deverá ser percorrido para que exista a comunicação entre os nós, o núcleo do protocolo de roteamento é o algoritmo de roteamento, este tem a função de determinar a rota adequada entre a origem e o destino dos datagramas. Diversos protocolos de roteamento podem ser implementados utilizando a simulação pelo NS2, como por exemplo o roteamento unicast, estático, dinâmico e vetor de distância.

Assim como apresentado pela pesquisa realizada por Blome (BLOME et al., 2013) existem diversos softwares que são utilizados para testes de penetração e testes de vulnerabilidades, mas ainda assim é possível se deparar com situações problemáticas nas quais o software para a realização destes testes não podem ajudar pois não contam com uma detectação de erros e validação de regras específicas.

# **MÁQUINAS DE ESTADOS FINITOS**

O Conceito de máquina de estados finitos, *Final State Machine* representado também pela siglas *FSM* ou em português MSF, trata-se de um conceito matemático utilizado na representação de programas ou circuitos lógicos possibilitando a visão geral dos estados que podem ser assumidos pelo protocolo, possibilitando uma visão do comportamento, dessa forma é possível criar um modelo formal deste protocolo.

Uma máquina de estados é uma técnica utilizada para facilitar o entendimento das definições que cada processo no sistema assume, os estados definem quais ações são permitidas aos processos e quais eventos são esperados pelo estado também definindo como cada estado responderá a estes eventos. (PASSOS. 2002)

Máquinas de estados são muito utilizadas em modelagens de diversos problemas como automação de design eletrônico, projeto de protocolos de comunicação, análise, entre outras situações em que uma máquina de estados poderá representar todo um comportamento.

Uma máquina de estados finitos possui uma quantidade finita de estados em que pode assumir, sendo que cada estado ocorrerá somente um por vez e quando ativo é chamado de estado atual. Para que os estados da máquina sejam alterados de um estado ativo para o próximo estado é necessário que ocorra alguma mudança, este acontecimento em uma máquina de estados é chamado de transição. Transições são as mudanças de estado que são compostas por eventos e ações. Um evento é correspondente a uma situação na qual permitirá que a transição ocorra e uma ação é a consequência desse evento. Para cada estado podem existir zero ou mais transições, para cada transição existirão eventos que acarretarão em uma ação dentro da máquina de estados finitos. (MASIN, XIVIER,TEOBALDO JÚNIOR. ####)

# **METODOLOGIA**

A partir das comparações realizadas entre as ferramentas de simulação de redes o software NS2 foi escolhido por contar com um arquivo de saída *trace*, assim como citado anteriormente no capítulo DADOS DA SIMULAÇÃO DE REDE, especificando também o significado de cada dado contido e seu significado .

Com o software NS2 definido, *scripts* de simulações distintas foram criados e executados pelo NS2. Através das simulações realizadas foram gerados os *traces* possibilitando a análise dos dados contidos no arquivo. Por meio da análise do *trace* foi possível identificar o conjunto de dados, o que estes representam na simulação, também como são ordenados no arquivo, gerando uma tabela para facilitar o entendimento do conteúdo do *trace*, como mostrado na figura a cima no capítulo DADOS DA SIMULAÇÃO DE REDE.

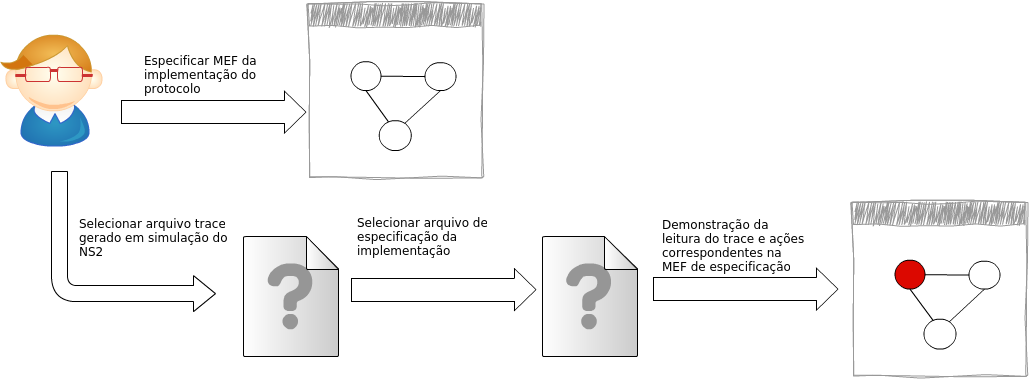
Para a realização de uma verificação de uma implementação de protocolo a metodologia aplicada será divida em duas partes, onde a primeira parte resulta em especificar a implementação do protocolo em uma máquina de estados finitos, definindo seus estados e quais eventos podem ocorrer em cada um deles. A modelagem utilizando MEF tem um papel importante na verificação de implementação pois é através dela que a segunda parte da metodologia se baseará.

A parte seguinte da metodologia será a construção de uma ferramenta de modelagem, especificação de implementação de um protocolo, esta terá como objetivo a modelagem de requisitos do protocolo, os quais sejam necessários para o funcionamento no projeto em questão no momento. Esta ferramenta verificará os eventos nos estados definidos a partir da modelagem do protocolo, realizada previamente com base no arquivo de *trace.* A ferramenta de verificação realizará uma leitura do arquivo *trace* comparando os eventos aguardados em cada estado na modelagem com os eventos encontrados nas tuplas do arquivo *trace*

*ferramenta de modelagem e especificação da implemetação do protoloco*

*ferramenta de verificação a a partir da modelagem e do trace*

*Escrever sobre a biblioteca que será utilizada.*



Identificados os eventos ocorridos na simulação será possível modelar uma possível verificação de implementação de um protocolo específico já simulado utilizando o conceito de máquinas de estados finitos. Com a modelagem através de MSF será possível especificar em cada um dos estados qual tipo de dados é esperado e juntamente com o arquivo *trace* da simulação compara-lo via software validando sua implementação (falar durante)