

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Simulação de Algoritmos de Mitigação de
Broadcast Storm em Redes FANET

Douglas Amorim de Oliveira



Simulação de Algoritmos de Mitigação de Broadcast Storm em Redes FANET

Douglas Amorim de Oliveira

Orientador: Kalinka R. L. J. C. Branco

Monografia de conclusão de curso apresentada ao
Instituto de Ciências Matemáticas e de Computação –
ICMC-USP - para obtenção do título de Bacharel em
Ciências de Computação.

Área de Concentração: Redes de Computadores

USP – São Carlos
Junho de 2017

*Até mesmo o mais sábio
Sensei já foi um aprendiz.*

Agradecimentos

Inicialmente gostaria de agradecer a professora Kalinka pela oportunidade da realização do projeto e todo apoio fornecido, ao Instituto de Ciências Matemáticas e de Computação e a USP pelo curso, infraestrutura e desenvolvimento intelectual.

Agradecimento especial também ao Rayner de Melo Pires, doutorando do LSEC (Laboratório de Sistemas Embarcados Críticos). Sua ajuda e ensinamentos foram de extrema importância para a realização deste trabalho.

Resumo

Com a popularização e avanços das tecnologias envolvendo Veículos Aéreos Não Tripulados (os VANTs) é possível deparar-se cada vez mais com cenários onde uma rede destes se faz útil para melhorar a performance em missões de diversos segmentos, como agricultura, controle territorial e de logística. Estas redes são chamadas de *Flying Ad-Hoc Network*, ou simplesmente de FANET. Ocorre que por ter sua comunicação baseada em troca de mensagens wireless a mesma não é confiável, isto é, está suscetível a perda de pacotes. Como a comunicação entre estas máquinas é de extrema necessidade utilizar-se Broadcasts para comunicação e retransmissão das mensagens entre os nós da rede. Esta estratégia por sua vez acarreta em uma espécie de flooding - poluição da rede - chamado de Broadcast Storm que torna a rede lenta ou até mesmo incomunicável. O foco deste trabalho é fazer um aprofundamento sobre os conceitos que envolvem este tema, analisar implementações de códigos que buscam mitigar estes efeitos nocivos, e, propor formas de passar estes código para ambiente virtual para permitir maior flexibilidade em testes além de permitir escalar experimentos reais.

Sumário

LISTA DE FIGURAS	6
CAPÍTULO 1: INTRODUÇÃO	7
1.1. CONTEXTUALIZAÇÃO E MOTIVAÇÃO	7
1.2. OBJETIVOS	8
1.3. ORGANIZAÇÃO DA MONOGRAFIA	9
CAPÍTULO 2: REVISÃO BIBLIOGRÁFICA	10
2.1. CONSIDERAÇÕES INICIAIS	10
2.2. CONCEITOS RELEVANTES	10
2.2.1 FLYING AD-HOC NETWORK	10
2.2.2 BROADCAST STORM	10
2.2.3 OMNeT++ E INET	11
2.3. TRABALHOS RELACIONADOS	11
2.3.1 ALGORITMO COUNTER-BASED	11
2.3.1 ALGORITMO COUNTER-BASED COM PROBABILIDADE FIXA	12
2.3.1 ALGORITMO COUNTER-BASED COM PROBABILIDADE DINÂMICA	12
2.4. CONSIDERAÇÕES FINAIS	13
CAPÍTULO 3: DESENVOLVIMENTO DO TRABALHO	14
3.1. CONSIDERAÇÕES INICIAIS	14
3.2. DESCRIÇÃO DO PROBLEMA	14
3.3. DESCRIÇÃO DAS ATIVIDADES REALIZADAS	15
3.4. RESULTADOS OBTIDOS	19
3.4.1 MAPEAMENTO DE VARIÁVEIS E MÓDULOS	19
3.5. DIFICULDADES, LIMITAÇÕES E TRABALHOS FUTUROS	21
3.6. CONSIDERAÇÕES FINAIS	21
CAPÍTULO 4: CONCLUSÃO	22
4.1. CONTRIBUIÇÕES	22
4.2. CONSIDERAÇÕES SOBRE O CURSO DE GRADUAÇÃO	23
REFERÊNCIAS	25

Lista de Figuras

FIGURA 1 - CENÁRIO ONDE A FANET ESTENDE O ALCANCE DE UMA MISSÃO	7
FIGURA 2 - TOPOLOGIAS PROPOSTAS PARA ANÁLISE	16
FIGURA 3 - TUTORIAL WIRELESS INET STEP 1	17
FIGURA 4 - TUTORIAL WIRELESS INET STEP 14	17
FIGURA 5 - TOPOLOGIA 1 PROPOSTA POR SERGIO ZUMPARO REPRODUZIDA NO OMNeT++	18

CAPÍTULO 1: INTRODUÇÃO

1.1. Contextualização e Motivação

Com o crescente acesso ao hardware e o aumento da tecnologia, o uso de veículos autônomos ou controlados remotamente vem se tornando cada vez mais frequente. Dentro destes veículos tem-se os VANTs (Veículo Aéreo Não Tripulado), sobre os quais se dá o foco deste trabalho [1].

Mundialmente os VANTs vêm em uma evolução expressiva, tanto com pesquisas focadas em seu desenvolvimento e aperfeiçoamento quanto nos usos e aplicações para as quais eles são destinados, como transporte de cargas, monitoramento de regiões e auxílio na agricultura. Com essa expansão torna-se interessante em determinados cenários o uso de múltiplos VANTs. Estas redes são chamadas de FANETs[2] (Flying Ad-Hoc Network). Nestes cenários que utilizam múltiplos VANTs (conforme ilustrado na Figura 1) a comunicação entre eles é importante para a execução das missões, e, mesmo em cenários com VANTs únicos, eles ainda devem se comunicar com aeronaves próximas para evitar colisões.

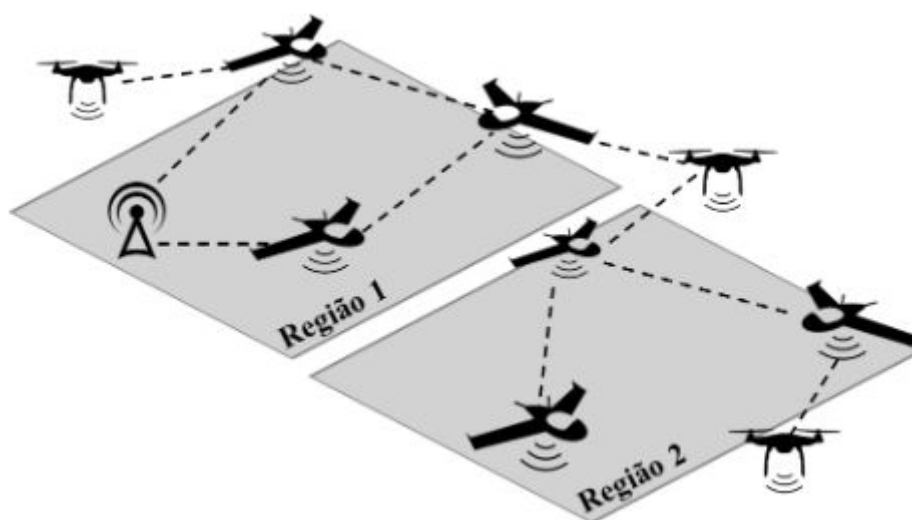


Figura 1 - Cenário onde a FANET estende o alcance de uma missão[3]

A difusão de dados em uma rede FANET é geralmente feita através de broadcasting. Essa técnica precisa ser ponderada a fim de evitar problemas que causam a contenção da rede[11]. Como uma solução, há algoritmos que mitigam mensagens a fim de evitar essa contenção.

Para que os estudos sobre estes algoritmos sejam feitos de forma mais prática, dinâmica e escalável faz-se necessário elaborar, em um ambiente simulado, condições iguais às encontradas em ambientes reais, permitindo assim o estudo do comportamento da comunicação em redes com N VANTs sem necessariamente ter que dispor destes equipamentos e de espaço suficiente.

1.2. Objetivos

Um trabalho anterior foi desenvolvido pelo estudante Sergio Zumpano Arnosti sobre a implementação de algoritmos com foco na mitigação de broadcast storm[3]. Foi desenvolvido para ser executado em ambiente real com 4 VANTs.

Neste trabalho é dada continuidade ao projeto, fazendo estudos sobre o trabalho que foi aplicado em ambiente real e como este pode ser replicado em um ambiente de simulação.

Para isto faz-se necessário aprender uma ferramenta que permita a simulação de uma rede de computadores. No caso foi escolhida a ferramenta OMNeT++[4], um simulador de eventos discretos que tem foco no uso acadêmico, educacional e de pesquisas, e que é frequentemente utilizado na literatura para este fim.

Em seguida, o objetivo é estudar como as variáveis se comportam nesta ferramenta e fazer comparações entre os parâmetros da simulação com os parâmetros do ambiente real para com isto manter a fidelidade necessária.

Portanto, os objetivos deste trabalho são:

- Aprendizado e/ou aperfeiçoamento das teorias envolvidas no desenvolvimento do projeto, como as teorias dos VANTs, FANETs e redes de computadores;

- Aprendizado do ambiente de simulação OMNeT++ e seu framework INET[5] de redes de computadores;
- Análise e estudo do código implementado em ambiente real para a mitigação de broadcast storm;
- Levantamento das variáveis em ambiente simulado que efetuam a mesma função dos códigos estudados no item anterior;
- Fazer um comparativo destes, com foco na fidelidade da simulação do ambiente real;
- Propor uma implementação no OMNeT++ que atenda aos requisitos citados no tópico anterior para tornar as simulações ideais e praticáveis.

1.3. Organização da Monografia

Após a introdução está disposto o capítulo sobre a revisão bibliográfica. Nele é apresentado um pouco sobre os conceitos relevantes para o projeto, trabalhos existentes relacionados com o mesmo, os materiais que foram utilizados como fonte de estudos, tutoriais de aprendizado da ferramenta de simulação (OMNeT++) e algumas considerações pertinentes.

Em seguida é apresentado o desenvolvimento do trabalho. O problema que foi abordado nesta monografia, as atividades realizadas durante o projeto, resultados alcançados, limitações encontradas, dificuldades durante a realização e trabalhos futuros.

E finalmente são apresentadas as conclusões e considerações sobre o curso de Bacharelado em Ciências de Computação no ICMC USP.

CAPÍTULO 2: REVISÃO BIBLIOGRÁFICA

2.1. Considerações Iniciais

Neste capítulo iremos abordar os conceitos utilizados no trabalho, como as terminologias da área, ferramentas, artigos e publicações de referência, tutoriais, códigos e trabalhos que serviram como base para a proposta deste projeto.

2.2. Conceitos Relevantes

2.2.1 *Flying Ad-Hoc Network*

As *Flying Ad-Hoc Networks*, conhecidas como FANETs, são especializações de redes de dispositivos móveis (MANETs - Mobile Ad-hoc Networks[12]) com adaptações de comunicação, mobilidade, implementação, missão e roteamento focadas em aeronaves.

Algumas de suas dificuldades são o fato de a FANET ter uma alta mobilidade dos nós, baixa densidade, 6 graus de liberdade no movimento e uma alta diversidade de sensores presentes nas aeronaves. Já como uma vantagem tem-se, geralmente, uma linha de visão sem obstáculos para interferir na troca de mensagens. Com estas características faz-se necessário buscar formas eficientes para a comunicação entre os nós [6][8].

2.2.2 *Broadcast Storm*

Em redes de comunicação que utilizam um meio sem fio para a sua comunicação podem ocorrer perdas de pacotes por muitos fatores (como alcance, colisões com frames de nós vizinhos, atraso pela contenção da rede). Como a FANET depende de uma comunicação confiável para a realização das missões utiliza-se a técnica de *Broadcast* para o envio de mensagens. Ocorre que a retransmissão excessiva e indiscriminada destas mensagens gera sobrecarga e inundação da rede, este problema é conhecido como *Broadcast Storm*[11].

2.2.3 OMNeT++ e INET

O OMNeT++ é uma ferramenta para a simulação de eventos discretos. Ele tem como um de seus frameworks o INET, que é responsável pela simulação de comunicação de redes wireless e mobiles. As ferramentas fornecem diversas informações sobre as simulações podendo acompanhar até mesmo, por exemplo, o consumo de energia por cada um dos componentes da rede.

2.3. Trabalhos Relacionados

Este trabalho busca dar continuidade nos estudos do trabalho “Estudo sobre a Difusão de Dados em FANETs” [3], utilizando-o como base e avançando para aproximar de uma futura implementação do mesmo em um ambiente totalmente simulado.

Diversos artigos e publicações foram estudadas como levantamento de conhecimento para o estudante[6][7][8].

Algumas soluções são propostas na literatura para a mitigação do *broadcast storm*, todas as abordagens focando em uma redução da taxa de retransmissão das mensagens recebidas.

2.3.1 Algoritmo *Counter-based*

Este algoritmo dispõe da implementação de um contador de ocorrências de uma mesma mensagem. É efetivo em ambientes onde há uma sobreposição de áreas cobertas pelos nós. A implementação é baseada na proposta em [13]. Nele os seguintes passos são executados:

- Quando uma mensagem é recebida pela primeira vez inicia-se um contador em 1 e a mensagem é armazenada em uma lista;
- Aguarda por um tempo aleatoriamente escolhido antes de tomar uma decisão. Neste tempo caso chegue uma mensagem redundante incrementa-se o contador;
- Ao chegar no tempo escolhido verifica se o número de ocorrências da mensagem no contador é maior do que um limite estabelecido. Caso

positivo, descarta a mensagem. Caso contrário retransmite e remove da lista.

2.3.1 Algoritmo *Counter-based* com probabilidade fixa

Em [14] os autores demonstram a combinação do algoritmo anterior com uma probabilidade fixa de retransmissão.

Em [14] e [15] o valor da probabilidade P é proposto como 0,65 para que o desempenho apresente ganhos comparado ao *counter-based* puro. Já em [16] os autores propõem o valor de 0,5 para resultados mais efetivos.

Pelas simulações realizadas em [15] obtivemos redução de 46% em redes esparsas e 56% em redes densas. Já as coberturas do pacote ficaram em 95% a 98%. Com o menor número de retransmissões temos também um menor atraso na rede por ocorrerem menos colisões dos pacotes.

2.3.1 Algoritmo *Counter-based* com probabilidade dinâmica

Segundo os autores de [17] (em sua maioria os mesmos de [16]) provavelmente o valor escolhido para a probabilidade fixa não será o ideal para todas as configurações de topologia de rede. Desta forma, foi proposta uma função que gera dinamicamente uma taxa de probabilidade ideal de acordo com o número médio de vizinhos de um nó na rede.

Em [17] foi proposta a seguinte função para o cálculo do número médio de vizinhos:

$$\lambda = (N - 1) \frac{\pi R^2}{A}$$

Onde temos:

N = Número de nós da rede;

R = Alcance do sinal de um Nó

A = Área total ocupada pela rede.

2.4. Considerações Finais

A etapa de revisão bibliográfica é importante para o desenvolvimento dos projetos acadêmicos, pois possibilita que os novos pesquisadores na área tenham um ganho muito significativo de conhecimento, além de mostrar qual o estado da arte na área, motivando e/ou possibilitando a buscar sempre avançar-lo.

CAPÍTULO 3: DESENVOLVIMENTO DO TRABALHO

3.1. Considerações Iniciais

Neste capítulo é apresentado todo o desenvolvimento do trabalho. O problema abordado, estudo da bibliografia, trabalho base em ambiente real realizado em trabalho prévio [3], aprendizado da ferramenta OMNeT++ e framework de simulações INET, análise de variáveis em ambiente real e simulado e o desenvolvimento dos algoritmos de mitigação de Broadcast Storm neste ambiente simulado que sejam coerentes com os apresentados pela implementação realizada em ambiente físico são agrupados neste capítulo.

Além dos tópicos citados acima serão também abordadas as dificuldades, as propostas de trabalhos futuros e as considerações finais sobre estes tópicos.

3.2. Descrição do Problema

O problema foco deste trabalho é a mitigação do problema de Broadcast Storm. Um dos problemas inerentes de redes wireless (e portanto das FANETs) é o fato de se tratar de uma rede não confiável, isto é, que está suscetível a perda de pacotes na transmissão, seja por fatores temporais, por fatores físicos ou mesmo por fatores técnicos (como uma nave tem 6 graus de liberdade em seu movimento, pode ocorrer, por exemplo, que a antena da mesma esteja posicionada de tal forma em um determinado momento t que ela não receba um pacote que foi-lhe transmitido).

Como a FANET depende de uma comunicação entre seus componentes para realizar sua missão (mantendo uma formação, trocando informações sobre eventos ou de varreduras em áreas) tem-se que fazer com que a comunicação seja mais segura. Para isto utilizou-se a técnica de Broadcast, onde um nó envia a mensagem para todos os demais nós da rede e os nós que recebem estas mensagens repassam novamente para os demais por

broadcast para aumentar a probabilidade de que todos recebam a mensagem.

Ocorre que, com a alta frequência dos broadcasts, têm-se o que se chama de Broadcast Storm, que nada mais é do que a sobrecarga da rede, causando lentidão e até mesmo indisponibilidade da comunicação, o que é altamente danoso para nosso cenário.

Para buscar mitigar os efeitos causados por esta forma de comunicação têm-se algumas implementações que utilizam técnicas para diminuir o número de ocorrências de replicação de pacotes e assim descongestionar, de forma significativa, a rede.

Algumas versões destes algoritmos, que são citadas posteriormente, foram implementadas e estudadas em ambiente real pelo estudante Sergio Zumpano, do ICMC USP [3]. Infelizmente o ambiente real traz limitações como números de aeronaves disponíveis e espaço físico. Neste trabalho é feita uma análise de como passar estes algoritmos para um ambiente simulado que mantenha fidelidade de resultados com o mundo real para então superar as limitações apresentadas.

3.3. Descrição das Atividades Realizadas

De início foi dado um foco no estudo da literatura do tema e da área. Como o estudante não tinha base profunda sobre FANET nem VANT estes foram os primeiros tópicos abordados.

Em seguida foi estudado o relatório "Estudo sobre a Difusão de Dados em FANETs"[3]. Neste foi aprofundado sobre os algoritmos de mitigação que o mesmo implementou durante seu projeto de pesquisa, que foram:

Algoritmo Counter-Based

Um algoritmo que usa como base principal para a decisão de rebroadcast ou não a redundância do pacote, utilizando um contador;

Algoritmo Counter-Based com Probabilidade Fixa

Faz uso da técnica implementada no Counter-Based mas combinada com uma probabilidade fixa para a retransmissão do pacote, que tem como valor

de p sugeridos 0,65 e 0,5;

Algoritmo Counter-Based com Probabilidade Dinâmica

Similar ao algoritmo anterior mas que, ao invés de simplesmente definir um valor para p de maneira constante, calcula o p dinamicamente com base nas informações do contador de pacotes redundantes e um número médio de vizinhos de um nó na rede.

Cada algoritmo trouxe alguns valores de otimização. Estes valores podem ser conferidos na íntegra em Zumpano[3], assim como a execução dos experimentos e como foi desenvolvido.

Cada um dos 3 algoritmos implementados foram executados em 3 topologias, conforme ilustrado na Figura 2:

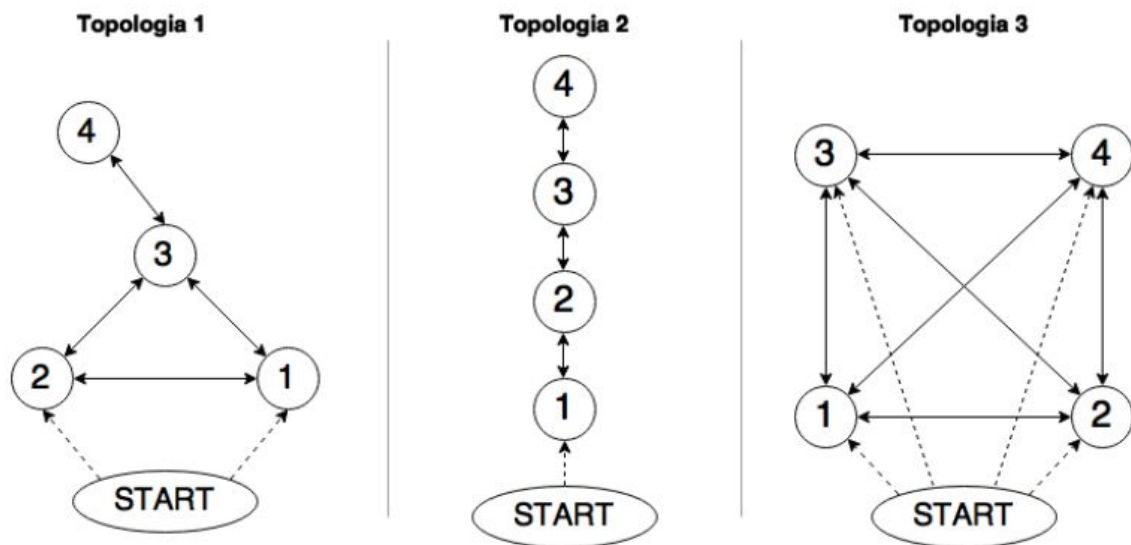


Figura 2 - Topologias propostas para análise[3]

Após estar familiarizado com os conceitos teóricos iniciou-se a buscar informações sobre a ferramenta OMNeT++. Trata-se de um simulador de eventos discretos que junto com o framework INET pode ser utilizado para a simulação de redes de computadores (com cabo, wireless e mobile).

Inicialmente foi estudado um tutorial chamado “TicToc”[9], do OMNeT++, para a

familiarização com a plataforma. Em seguida foi seguido um tutorial mais robusto para o aprendizado dos módulos e componentes disponíveis no framework INET, focado em wireless[10]. Este, dividido em 14 sub-tutoriais parte desde o básico, com 2 hosts se comunicando até uma rede com dispositivos mobile em movimento, com raio de distância e obstáculos interferindo na comunicação, como podemos ver nas Figuras 3 e 4:

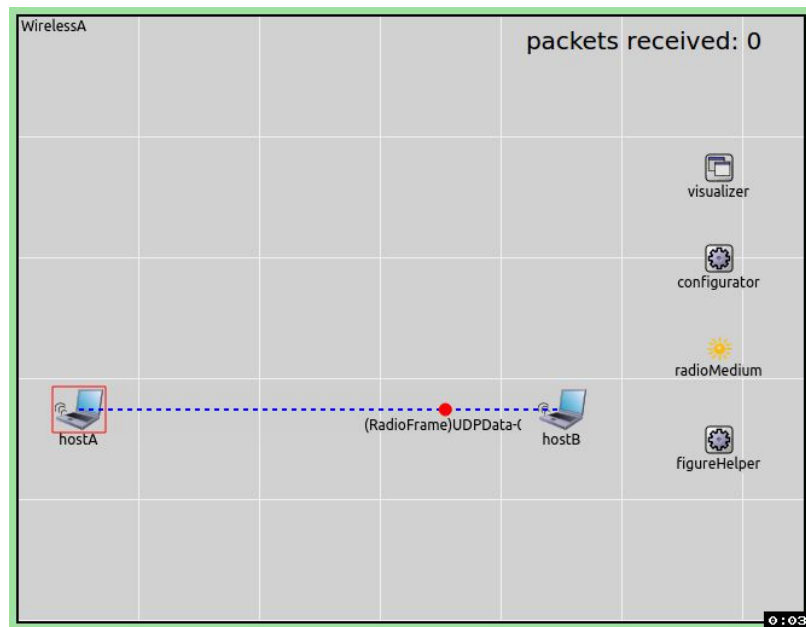


Figura 3 - Tutorial Wireless INET step 1[10]

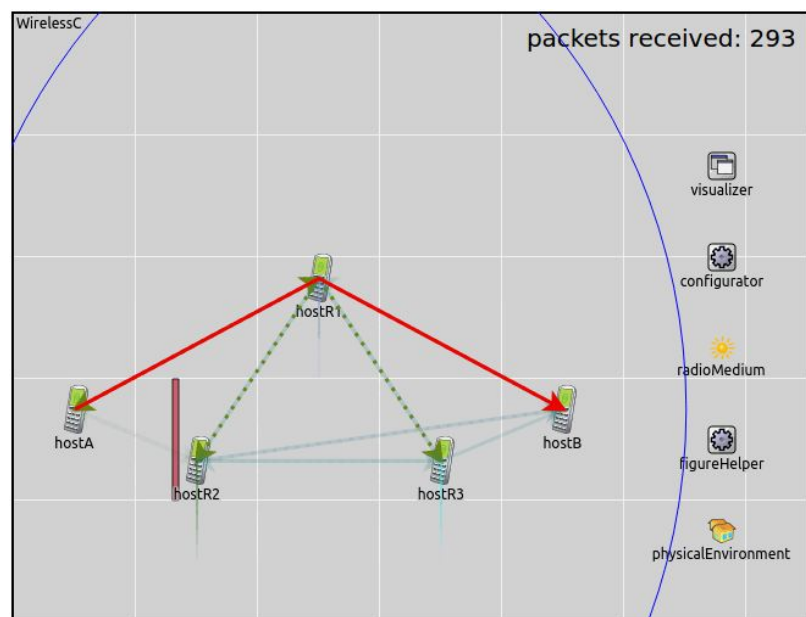


Figura 4 - Tutorial Wireless INET step 14[10]

A próxima etapa constituiu na fase de análise do código fonte aplicado em ambiente real. O protocolo utilizado para a comunicação é UDP (*User Datagram Protocol*), e ao analisar a forma de comunicação chegou-se a conclusão que um componente que era passível de utilização como base para os nós é o *UDPBasicApp*[19] e para o controle da potência e alcance escolheu-se o *Ieee80211DimensionalRadioMedium* [19] que apresentava características mais próximas das presentes em cenário real.

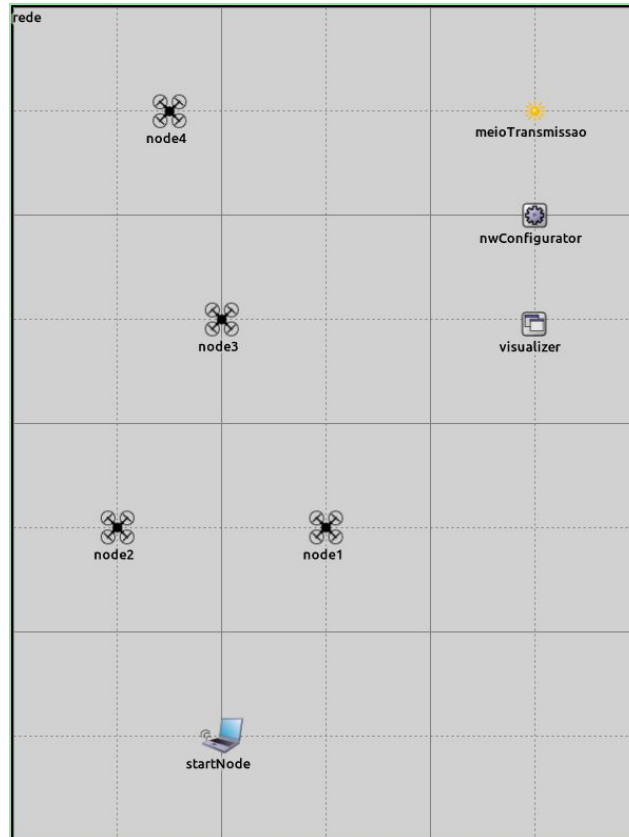


Figura 5 - Topologia 1 proposta por Sergio Zumpano reproduzida no OMNeT++

Em um primeiro momento foi projetada a topologia 1 (ilustrada na Figura 5) para a primeira versão de testes de migração e adaptação de código. Os códigos que devem ser adaptados são o código do sender e o códigos dos 3 algoritmos de mitigação. O sender no exemplo da Figura 5 é o startNode e todos os demais são os receivers. Como o código do sender é fixo este foi escolhido para ser migrado primeiro, visto que já seria possível testar seu funcionamento com nós comuns de receptores e com a experiência adquirida em sua adaptação os outros códigos são feitos de forma mais rápida e eficiente.

3.4. Resultados Obtidos

Como resultado tem-se um aprofundamento sensível do estudante nos temas abordados no trabalho como Veículos Aéreos Não Tripulados, redes móveis/sem fio de computadores e uma boa introdução a ferramenta robusta de simulação OMNeT++ com seu framework INET. Houve também um amadurecimento por parte do estudante no senso crítico para analisar os paradigmas e variáveis envolvidas quando se pensa em um cenário “Ambiente Real vs Ambiente Simulado”, dando uma noção para que o mesmo enfrente outros problemas do gênero no futuro com maior domínio. Tudo isto, aliado com o contato que o mesmo teve com pesquisadores da área, foi uma experiência que desenvolveu mais seu interesse para possível continuação na área de pesquisa acadêmica.

Além disso, os resultados das comparações das variáveis em Ambiente Real x Ambiente Simulado podem ser úteis para um trabalho de continuidade que conclua a migração dos algoritmos de mitigação de Broadcast Storm para o OMNeT++ que é o objetivo principal que este trabalho busca propiciar no futuro.

3.4.1 Mapeamento de Variáveis e Módulos

Ao migrar para ambiente simulado deve-se primeiro atentar sobre a simulação de comportamento do hardware e das interferências encontradas na execução real.

Primeiro foi mapeado então um módulo para a comunicação dos nós. Como citado anteriormente foi utilizado o *UDPBasicApp*[19] como base, pois a aplicação funciona com comunicação por protocolo UDP e tem seus requisitos de comportamento atendidos pelo mesmo.

Em seguida, no arquivo *rede.ned* o *Ieee80211DimensionalRadioMedium*[19] foi escolhido como simulador de frequência e alcance do raio visto que ele emula a queda da qualidade e potência da comunicação de acordo com a distância entre os nós, assim como ocorre com as execuções em ambiente real.

Para a configuração do IPv4 estático da rede e do *IPv4RoutingTable*[19] foi utilizado *IPv4NetworkConfigurator*[19].

Na etapa apresentada na Figura 5 os nós são criados a partir dos componentes *AdhocHost*. O *startNode* é configurado para não receber pacotes, ao contrários dos demais nós da rede.

Já no arquivo *sender.ned* foram implementadas informações referentes ao *StartNode*. Sendo as principais elas:

destinatario: uma string que deve conter o endereço *broadcast* da rede;

destPort: um inteiro que contém a porta de aplicação nos receptores;

TTL: o *TimeToLive*[18] configurado. Depende da topologia da rede simulada.

No arquivo *sender.h* são configuradas as informações de comportamento que este nó *startNode* deve seguir. Além de declarar o *socket* como sendo do tipo *UDPSocket*[19] o cabeçalho das funções do *sender.cc* são definidas.

No *sender.cc* as funções que foram implementadas são:

initialize(): função responsável por gerenciar o comportamento ao iniciar o módulo da aplicação. Quando um módulo ou componente é inicializado no OMNeT++ esta função é executada automaticamente;

handleNodeStart(): função chamada pelo *initialize()* quando o módulo é ativado. Inicia o *UDPSocket* e agenda uma mensagem com o tempo de início;

handleNodeShutdown(): função para gerenciar a finalização do módulo. Finaliza a mesma e dá um retorno de sucesso;

handleNodeCrash(): executada quando a aplicação *startNode* parar de funcionar;

buildMessage(): constrói a mensagem a ser enviada. Esta mensagem segue o seguinte formato: concatena o último octeto do IP do módulo, o contador da mensagem (qual mensagem é esta, controlada por um contador incremental), o endereço IP do nó; o TTL da mensagem e a cadeia “mensagem”. Cada uma destas informações é separada por um “;” (ponto e vírgula);

handleMessageWhenUp(): lida com as mensagens de acordo com sua característica. Caso seja recebida ignora-a (o *startNode* não trabalhará recebendo mensagens), caso seja

para enviar chama a função *sendPacket()* e caso seja de parada encerra sua execução;

sendPacket(): cria o pacote a ser enviado pela rede, utilizando a classe *ApplicationPacket*[19] que estende a classe *cPacket*[19] adicionando o atributo de número de sequência no pacote. Caso tenha mensagens na fila ele agenda o evento, caso contrário ele encerra sua execução.

3.5. Dificuldades, Limitações e Trabalhos Futuros

A principal dificuldade encontrada pelo estudante foi na etapa de análise dos sistema implementado para ser executado em ambiente real e buscar módulos e formas de migrá-lo para o OMNeT++. O software de simulação apresenta uma curva de aprendizado não muito favorável que, aliado com o tempo curto para a realização do Trabalho de Conclusão de Curso, de aproximadamente 15 semanas até a entrega da monografia, foi uma limitação para que o estudante atingisse todos os resultados desejados.

Como trabalho futuro existe a possibilidade da finalização da migração dos algoritmos, propostos pelo Sérgio Zumpano em seu trabalho, para a ferramenta OMNeT++.

Acredito também que, apesar de ter amadurecido durante o desenvolvimento deste trabalho, o estudante ainda tem alguns pontos a aprimorar para obter sua eficiência desejada.

3.6. Considerações Finais

O objetivo maior deste trabalho, como citado no capítulo, é servir de base para a migração total dos algoritmos de Mitigação de Broadcast Storm para ambiente virtual e permitir uma maior flexibilidade nas simulações desejadas.

CAPÍTULO 4: CONCLUSÃO

No trabalho foi possível desenvolver muitas atividades que foram de extremo enriquecimento para o estudante. Como dito no capítulo anterior ainda faltam alguns pontos para trabalhos futuros, mas pode-se afirmar que, com todas as dificuldades encontradas, o estudante se encontra satisfeito com o resultado obtido, onde estaria disposto a fazer tudo novamente, talvez apenas com um pouco mais de engajamento e proatividade nas etapas iniciais mas que irá levar como experiência para oportunidades futuras.

Para ter ido um pouco além nos resultados acredita-se que um período maior para a etapa de interpretação do ambiente físico e busca por modelagem no simulado faria-se necessário. De início o estudante ficou travado com o desenvolvimento desta fase, mas que junto com doutorando conseguiram um desenvolvimento maior no final.

4.1. Contribuições

As principais contribuições deste trabalho estão na exploração e busca de módulos e implementações no OMNeT++ que façam um bom papel de simulação fiel com os códigos de ambiente real físico. Acredita-se que em breve essa migração será completada.

Como profissional o projeto me proporcionou um amadurecimento e contato mais próximo com uma nova área de pesquisa e do conhecimento com a qual eu posso vir a utilizar em projetos futuros. A interação no laboratório com o doutorando Rayner foi-me de grande valia também para desenvolver a visão de que quando trabalhamos em conjunto com as pessoas tendemos a conseguir resultados maiores. Sempre me vi como uma pessoa mais individualista, que trabalha melhor sozinho, mas aprendi nesta experiência de vida que sempre devemos ter a humildade de reconhecer que muitas vezes não sabemos como agir e que perderemos muito tempo buscando soluções para problemas que o seu colega já pode ter encontrado e está disposto a compartilhar para gerar resultados mais significativos.

4.2. Considerações sobre o Curso de Graduação

Ingressei no curso de Bacharelado em Ciências de Computação no Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC-USP) no ano de 2014.

Nestes 7 semestres de curso houveram diversos eventos, como mudança da grade curricular do Curso, com a inclusão de algumas disciplinas e remoção de outras. Das remoções tivemos Sistemas Operacionais II, Laboratório de Bases de Dados e Tópicos Avançados em Redes de Comunicação. As remoções em meu ver foram acertadas, visto que as 3 são aprofundamentos/práticas em conteúdos e/ou áreas já estudadas em disciplinas anteriores. Quando penso em um Bacharel em Ciências de Computação eu espero que o mesmo tenha um conhecimento razoável sobre como uma rede, um banco de dados e um sistema operacional funcionem, sua teoria e conceitos. Já a prática depende do enfoque que o profissional planeja para sua carreira. Não vejo como necessidade que um Cientista, que trabalha com Otimização, por exemplo, saiba tecnologias de manipulação de Bases de Dados relacionais como SQL na prática. Obviamente é interessante o conhecimento de seu funcionamento e de seu background, mas os comandos em si não vejo como requisito.

Existem disciplinas que ainda hoje constam na grade que vejo que seguem a mesma linha e poderiam se tornar optativas. Um exemplo seria Desenvolvimento Web. Tecnicamente com as outras disciplinas já temos um background de programação, lógica, concorrência e otimização de recursos. Vejo a disciplina mais como algo focado em tecnologias, o que acredito que não seja necessário aos cientistas de modo geral. Mesmo tendo uma alta demanda do mercado por profissionais desta área, ela sendo optativa continuaria sendo cursada por alunos que planejam seguir na área mas não seriam necessárias para os demais.

Já sobre o curso como um todo classificaria-o como excelente. Vejo que temos uma base muito sólida de teoria, coisa que, infelizmente, não vemos em todas as instituições. Saímos com total capacidade de projetar um sistema de grande porte, fazer toda a sua engenharia, desenvolvimento e finalização. Seja em um sistema comercial ERP, na segurança de informação, ou mesmo em um sistema de alto desempenho que demande computação paralela. Vejo nos meus colegas também a capacidade de não apenas

aplicarem a teoria como também de seguirem no caminho da pesquisa e desenvolverem novos conceitos, teorias, tecnologias e assim avançarem o estado da arte.

Minha experiência em São Carlos foi maravilhosa. Primeiro que a Universidade tem políticas efetivas de permanência estudantil, permitindo que eu pudesse focar nos estudos sem ter grandes preocupações no âmbito financeiro que tornassem ainda mais difícil obter rendimento acadêmico. O Instituto de Ciências Matemáticas e de Computação dispõe de uma infraestrutura exemplar, tanto nas edificações como nos equipamentos e na biblioteca.

Gostaria de fazer alguns agradecimentos especiais para alguns professores do instituto que por suas aulas ou convívio me fizeram criar um grande carinho. Em ordem cronológica: Sueli Mieko Tanaka Aki , João Luís Garcia Rosa, Cynthia Lage Ferreira, Maria Cristina Ferreira de Oliveira, Kalinka Castelo Branco e Thiago Pardo.

De fato o ICMC é um instituto que, apesar de não ser perfeito, é memorável e com certeza deve deixar saudades a todos que dele saem.

REFERÊNCIAS

- [1] A. Saulnier, S. Thompson, *The “Rise” of Unmanned Aerial Vehicles (UAVs) in Canada: An Analysis of Special Flight Operation Certificates (SFOCs) from 2007 to 2012*
- [2] I. Bekmezci, O. K. Sahingoz e S. Temel, *Flying Ad - Hoc Networks (FANETs): A Survey*, *Ad Hoc Networks*, vol. 11, n. 3, pp. 1254 - 1270, 2013.
- [3] S. Z. Arnosti, *Estudo sobre a Difusão de Dados em FANETs*, São Carlos: Universidade de São Paulo - ICMC (Trabalho de Iniciação Científica), 2015.
- [4] “OMNeT++ Discrete Event Simulator” [Online]. Available: <https://omnetpp.org/> [Acesso em 12 de Junho de 2017]
- [5] “INET Framework An open-source OMNeT++ model suite for wired, wireless and mobile networks.” [Online]. Available: <https://inet.omnetpp.org/> [Acesso em 12 de Junho de 2017]
- [6] I. Bekmezci, O. K. Sahingoz e S. Temel, “Flying Ad - Hoc Networks (FANETs): A Survey”, *Ad Hoc Networks*, vol. 11, n. 3, pp. 1254 - 1270, 2013.
- [7] O. K. Sahingoz, “Networking models in flying ad hoc networks(FANETs): Concepts and challenges”, *Journal of Intelligent and Robotic Systems*, vol. 74, n. 1 - 2, pp. 513 - 527, 2014.
- [8] R. M. Pires, S. Z. Arnosti, A. S. R. Pinto e K. R. L. J. C. Branco, “Experimenting broadcast storm mitigation techniques in FANETs”, em *HICSS - 49 Hawaii International Conference on System Sciences*, Kauai, Hawaii, 2016.
- [9] “TicToc Tutorial for OMNeT++” [Online]. Available: <https://omnetpp.org/doc/omnetpp/tictoc-tutorial/> [Acesso em 12 de Junho de 2017]
- [10] “Wireless Tutorial for the INET Framework” [Online]. Available: <https://omnetpp.org/doc/inet/api-current/tutorials/wireless/> [Acesso em 12 de Junho de 2017]
- [11] “Definition - What does Broadcast Storm mean?” [Online]. Available: <https://www.techopedia.com/definition/6270/broadcast-storm> [Acesso em 12 de Junho de 2017]
- [12] “Roteamento em MANETs” [Online]. Available: https://www.gta.ufrj.br/seminarios/semin2002_1/Daniel/ [Acesso em 12 de Junho de 2017]
- [13] Y. - C. Tseng, S. - Y. Ni, Y. - S. Chen e J. - P. Sheu, “The broadcast storm problem in a mobile ad hoc network,” *MobiCom '99 Proceeding of the 5th annual ACM/IEEE international conference on mobile computing and networking*, pp. 151 - 162,

Aug. 1999.

[14] A. Mohammed, M. Ould - Khaoua e L. Mackenzie, “An efficient counter - based broadcast scheme for mobile ad hoc networks, ” em Formal Methods and Stochastic Models for Performance Evaluation , vol. 4748, Berlin, Springer Berlin Heidelberg, 2007, pp. 275 - 283.

[15] B. Williams e T. Camp, “Comparison of broadcast techniques for mobile ad hoc networks,” MobiHoc '02 Proceedings of the 3rd ACM international symposium on mobile ad hoc networking & computing, pp. 194 - 205, June 2002.

[16] A. Mohammed, M. Ould - Khaoua, L. Mackenzie e J. Abdulai, “Improving the performance of counter - based broadcast scheme for mobile ad hoc networks,” IEEE International Conference on Signal Processing and Communications, pp. 1403 - 1406, 27 Nov. 2007.

[17] A. Mohammed, M. Ould - Khaoua e L. Mackenzie, “ An improved rebroadcast probability function for an efficient counter - based broadcast scheme in MANETs,” Second UKSIM European Symposium on Computer Modeling and Simulation, pp. 536 - 541, Sept. 2008.

[18] “Time to Live” [Online]. Available: https://pt.wikipedia.org/wiki/Time_to_Live [Acesso em 12 de Junho de 2017]

[19] “INET Framework for OMNeT++” [Online]. Available: <https://omnetpp.org/doc/inet/api-current/inet-manual-draft.pdf> [Acesso em 12 de Junho de 2017]