

1. Introdução

Com o advento e a popularização dos dispositivos móveis (celulares, netbooks, tablets e outros) e conexão com a internet, a nossa sociedade está rodeada de atividades que se valem de um canal de comunicação para realizar trocas de informações. Dentro dessa perspectiva, as redes sem fio e outros tipos ainda mais as baseadas em mobilidade, tem ganhado espaço e recursos nos estudos científicos.

A segunda parte do curso visa tratar de Modelagem de Sistemas. Dessa maneira, através do modelo de mobilidade da UFAM, o Trabalho Prático II sugere implementar e avaliar (com v&v) duas políticas de gerência de buffer para redes tolerantes a atrasos e desconexões (ou simplesmente DTN), utilizando o simulador para redes oportunistas *One*, a saber: *Evict shortest life time first* e *N-Drop*. As métricas utilizadas na simulação para comparação são Probabilidade de Entrega, Tempo Médio no Buffer, Taxa de Overhead e Número Médio de Saltos. A seguir, apresenta-se a contextualização de cada área do problema, para que, diante do embasamento teórico fornecido, seja possível a fácil compreensão do experimento realizado e de todos os mecanismos utilizados para obtenção dos resultados.

Contextualização

V&V:

Validação: processo de determinar se um modelo de simulação é uma representação precisa do sistema real estudado. Para esta simulação foi utilizada a Técnica de Validação 01 definida: “membros da equipe de simulação, potenciais usuários do modelo, conhecedores do sistema em estudo, baseados em estimativas e intuições, comparam subjetivamente o modelo e o sistema para julgar se o modelo e seus resultados são razoáveis”.

Verificação: examinar os códigos criados com o intuito de verificar se o programa executa de maneira correta. Foram utilizadas aqui as Técnicas de Verificação 01, 02 e 03, a saber: 1) escrever e debugar em módulos e submódulos; 2) utilizar mais de uma pessoa para revisar o código, pois aquele que codificou um subprograma ou submódulo não é um bom crítico; e 3) executar a simulação adotando uma variedade de configurações dos parâmetros de entrada e verificar se a saída é razoável.

Redes Tolerantes a atrasos e desconexões: Delay Tolerant Networks - DTNs foram desenvolvidas pensando-se em se prover comunicação em ambientes onde a desconexão dos enlaces de comunicação são frequentes, através do emprego de mecanismos do tipo *store and forward* para o roteamento de mensagens.

As DTNs foram originadas a partir dos esforços voltados ao estabelecimento de uma rede interplanetária. Contudo, é possível se fazer uso de diversas características desse tipo de rede

em aplicações terrestres, especialmente em cenários onde as aplicações são assíncronas e não sensíveis a grandes variações nas condições de entrega.

Simulador para redes oportunistas One: O *Opportunistic Network Environment*, ou One, é uma ferramenta *open source*, desenvolvida em Java que simula uma DTN. Implementando todos os mecanismos necessários ao funcionamento de uma Rede Tolerante a Atrasos, fornece uma ferramenta poderosa para simular situações específicas de movimentações de nós em caminhos pré-determinados ou aleatórios, sendo possível customizar a execução conforme seja necessário.

Os eventos a serem gerados na simulação nada mais são do que mensagens ou pacotes com origem e destino determinados. Eles podem ser configuráveis em tamanho e intervalo de criação, além de poderem restringir quais nós podem gerá-los ou recebê-los.

Os relatórios são os agrupamentos de dados gerados a partir da simulação. Eles fornecem estatísticas essenciais para a interpretação e análise dos diferentes cenários construídos, sendo os principais:

Mensagens Criadas: dados de mensagens que foram geradas no intervalo de simulação;

Mensagens Entregues: dados de mensagens que alcançaram com sucesso o seu destinatário; e

Conectividade na rede: instantes de ocorrência de conexão e desconexão entre nós.

Os mapas representam o conjunto de pontos e retas nos quais os nós podem se deslocar. No *One* é utilizado um formato específico de representação de mapas: o WKT.

A interface do *One* é de fácil compreensão, composta por quatro áreas principais: área da simulação, controle da simulação, lista dos nós e log de eventos. A simulação pode ser pausada, adiantada ou acelerada e é possível obter zoom na área de simulação.

A área principal exibe os nós, nomeados distintamente, os caminhos em que podem trafegar e o raio de transmissão de cada um deles. O log de eventos contém informações sobre conexões e mensagens, como o instante de início da conexão, o processo de custódia de uma mensagem e o término da conexão entre os dois. É possível filtrar as mensagens do log e ainda acessar informações mais detalhadas sobre algum nó diretamente pela área de eventos.

Nas listas dos nós do cenário é possível acessar informações mais detalhadas sobre cada um deles clicando em sua identificação; com isso visualiza-se quantas mensagens possui em seu buffer, com quais nós está conectado e o status das suas mensagens geradas ou em custódia.

Modelo de Mobilidade: é a representação simplificada da realidade, através da qual é possível simular padrões conhecidos de mobilidade de entidades. Estes padrões são extraídos habitualmente da realidade e permitem a geração de novos cenários. A simulação executada com os modelos pretendidos é cada vez mais importante na verificação de protocolos de encaminhamento antes do uso dos seus benefícios por todos os utilizadores da Internet. Para verificar e melhorar o desempenho dos mesmos é necessário recolher dados de diversas métricas que permitam a comparação e análise dos modelos, comportamentos e capacidades.

Modelo de Mobilidade da UFAM: Utilizando smartphones equipados do aplicativo *MyTracks*, que registra trilhas traçadas a partir da geolocalização do dispositivo conforme o deslocamento do usuário no espaço, um conjunto de dados foram extraídos para gerar um modelo real de mobilidade na UFAM, com amostras localizadas no Campus Norte. Com o *Bonnmotion*, foi criado um modelo de mobilidade a ser interpretado pelo *One*, agregando os

dados de 5 trilhas traçadas, tratadas como 5 nós a se comunicarem. Os tempos mínimo e máximo da simulação e as posições mínima e máxima nas coordenadas X e Y do conjunto de trilhas são informados, e os dados reais são interpretados no simulador quanto ao envio de pacotes de um nó a outro em determinado tempo e espaço.

Evict Shortest life time first: Cada uma das mensagens é associada a um valor de probabilidade de entrega que, inicialmente, é zero. Esse valor é incrementado a cada envio da mensagem até chegar a um valor limite de probabilidade de entrega, valor este que é definido a critério do grupo. A mensagem que chegar a este valor ou estiver com o valor de probabilidade de entrega maior é descartada.

N-Drop: A mensagem que foi encaminhada N vezes é descartada. N é um valor definido a critério do grupo.

Métricas: Os modelos de mobilidade incluem diversas métricas para analisar, descrever e comparar modelos e características destes, permitindo escolher o modelo que melhor se adequa a cada objetivo ou aplicação. As métricas estão agrupadas em protocolos e modelos dado que são usadas para avaliar parâmetros nestas áreas. Métricas a serem avaliadas na simulação para comparação:

- *Probabilidade de entrega: essa função mede as mensagens que foram recebidas corretamente pelo destinatário final em determinado período de tempo;
- *Tempo médio no buffer: indica o tempo médio em que uma mensagem passa no buffer;
- *Taxa de overhead: mede a quantidade de mensagens não entregues ao destinatário; e
- *Número médio de saltos: indica a quantidade média de saltos que uma mensagem faz até chegar ao seu destino.

Implementação

Criamos um modelo baseado na mobilidade dos alunos do grupo dentro do *campus* da universidade e, vale ressaltar que, este *campus* escolhido é representado pela Faculdade de Tecnologia (FT), Instituto de Computação (ICOMP) e pela Faculdade de Direito (FD), visto que os integrantes da equipe frequentam quase sempre essas localidades.

A partir do site *uTracks*[1], foi possível verificar o trajeto realizado pelos alunos no *Google Maps*. E nas imagens geradas pelo site, foi possível notar que as mobilidades gravadas pelo celular condizem com a nossa própria movimentação cotidiana, visto que, visitamos sempre as salas de aula, almoçamos na FT ou lanchamos na FD. Isso pode ser comprovado pelas imagens.

V&V

Validação: O grupo observou, dado o problema real como sendo a mobilidade dos alunos da UFAM, se o modelo de simulação seria uma representação precisa do sistema real. Hipótese para Validação: se gerarmos um modelo de mobilidade baseado nos *Tracks* (trajetos) reais realizados pelos alunos e coletados pelo celular, a modelagem tende a seguir esses padrões,

como é possível verificar no Anexo I deste documento. Esses trajetos coletados são considerados aceitáveis de acordo com a nossa estimativa e intuição, visto que as imagens condizem com a realidade. Logo, o sistema modelado tende a seguir o padrão coletado e também tende a ser um sistema capaz de simular uma mobilidade bem próxima da realidade.

Verificação: Cada membro da equipe verificou os códigos criados a fim de procurar erros e verificar se os programas criados executavam de maneira correta. Isto ocorreu tanto na implementação das políticas quanto na implementação dos submódulos que auxiliaram a ordenação e junção dos arquivos coletados pelo *MyTracks*. A verificação foi realizada com pequenos testes e verificação intuitiva dos resultados, em que cada aluno da equipe realizou a etapa de verificação ao menos para confirmar e confrontar os resultados encontrados pelos outros usuários.

Execução do Tutorial

Foi utilizado o *Track Maker* para traçar o limite superior e mínimo do mapa de coleta. Foram gerados os arquivos de configuração, utilizando o *GPXImport*, conforme tutorial. Para gerar o arquivo unificado com os traces de coleta, utilizou-se um pequeno programa feito em Perl.

Foram adicionadas no arquivo de configuração do *ONE* as linhas:

```
Group.movementModel = ExternalMovement  
  
ExternalMovement.file = data/caminhos.gpx.one  
Group.nrofHosts = 5 <-- ----- NUMERO DE NOS
```

Ao rodar no *ONE*, ocorreu o seguinte erro:

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: -2147483647

```
at interfaces.ConnectivityGrid.cellFromCoord(ConnectivityGrid.java:191)  
at interfaces.ConnectivityGrid.updateLocation(ConnectivityGrid.java:143)  
at interfaces.SimpleBroadcastInterface.update(SimpleBroadcastInterface.java:70)  
at core.DTNHost.update(DTNHost.java:327)  
at core.World.updateHosts(World.java:217)  
at core.World.update(World.java:186)  
at gui.DTNSimGUI.runSim(DTNSimGUI.java:115)  
at ui.DTNSimUI.start(DTNSimUI.java:77)  
at core.DTNSim.main(DTNSim.java:92)
```

Implementação das políticas:

NDrop: recebem-se como parâmetros um valor booleano, para saber se a mensagem deve ser excluída, e um valor limite de encaminhamentos de mensagem, referente a 'N'. O valor limite referente a 'N' é definido por uma constante de valor 5, de acordo com decisão tomada pelo grupo durante implementação.

Para contar quantas vezes a mensagem é encaminhada, um *HashMap* de nome 'countN', referente a 'NDrop', foi instanciado. Este *HashMap* relaciona o ID da mensagem a um valor inteiro que é a contagem de encaminhamentos.

Toda vez que o método *makeRoomForMessage* for executado, o método que aplica a política *NDrop* será chamado e para cada mensagem os seguintes passos serão tomados:

- Verificar se está sendo enviada. Caso esteja, incrementa a contagem de envios;
- Se a contagem de envios da mensagem chegar ao limite ('N') e não houver mensagem marcada para exclusão, então ela será marcada para exclusão;
- Mensagem marcada para exclusão é retornada.

De acordo com decisão de implementação tomada pelo grupo, caso mais de uma mensagem atinja o limite de envios, somente a primeira será marcada para exclusão.

Evict Shortest LifeTime First: recebe-se como parâmetro o limite que a contagem da probabilidade de entrega pode atingir. O valor limite foi definido para 5, de acordo com decisão de implementação tomada pelo grupo.

Para contar a probabilidade e envio da mensagem foi instanciado um *HashMap* de nome 'countProb'. Este *HashMap* relaciona o ID da mensagem com a sua probabilidade de envio.

Toda vez que o método *makeRoomForMessage* for executado, o método que aplica a política *Evict Shortest LifeTime First* é chamado e para cada mensagem os seguintes passos serão tomados:

- Verifica se está sendo enviada. Caso esteja, a contagem da probabilidade é incrementada em um valor aleatório maior que 0;
- Se a contagem da probabilidade de envios da mensagem atingir, ou ultrapassar, o limite e não houver mensagem marcada para exclusão, então ela será marcada para exclusão;
- Mensagem marcada para exclusão é retornada.

De acordo com decisão de implementação tomada pelo grupo, caso mais de uma mensagem atinja o limite de probabilidade de envios, somente a primeira será marcada para exclusão.

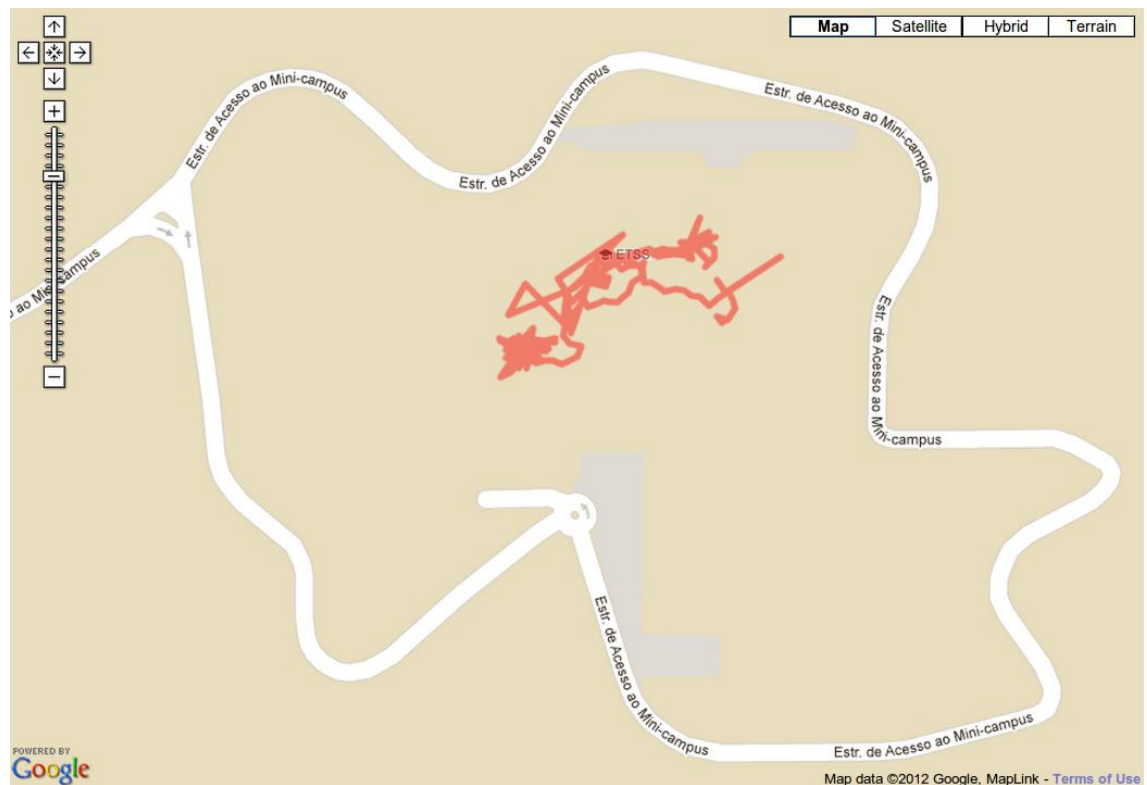
Resultados obtidos

Ao seguir todo o processo de implementação e geração do modelo, conseguimos simular a comunicação dos nós criados a partir do modelo real, ainda que com dificuldades na manipulação de determinados arquivos.

Conclusão

Diante das etapas de modelagem de sistemas, a simulação demonstrou que, se um modelo de simulação é válido, ele pode ser usado para tomar decisões sobre o sistema real. Dessa maneira, o trabalho prático proporcionou ao grupo o aprendizado de método para geração de modelos de simulação, de etapas de modelagem de uma simulação, de extração de dados a partir de um modelo real e de uso de ferramentas de simulação e de formatação de dados em modelo de simulação.

Anexo I







Referências

[1] *TrackMaker*: www.gpstm.com

Salvador, E., Macedo, D., Nogueira, J. (2011) “Gerencia Autônômica de Redes DTN”, In: I Workshop on Autonomic Distributed Systems, http://sbrc2011.facom.ufms.br/files/workshops/wosida/ST02_3.pdf, Maio.

Chilro, R. (2009) “Estudo do impacto do rádio em métricas de modelos de mobilidade DTN”, Dissertação de Mestrado em Redes e Serviços de Comunicação, Faculdade de Engenharia da Universidade do Porto, <http://repositorio-aberto.up.pt/bitstream/10216/58733/1/000136388.pdf>, Maio.

Tabarelli, A., Silva C. (2009) “Redes tolerantes a atrasos, protocolos de disseminação e aplicações”, Trabalho de Formatura Supervisionado, São Paulo, <http://pt.scribd.com/doc/89815246/monografia>, Maio.