

TRABALHO DE FORMATURA SUPERVISIONADO - MAC0499

REDES TOLERANTES A ATRASOS,  
PROTOCOLOS DE DISSEMINAÇÃO E  
APLICAÇÕES

ADRIANO TABARELLI  
CAIO CESTARI SILVA

(PROF. ORIENTADOR: ALFREDO GOLDMAN VEL LEJBMAN)

SÃO PAULO, 30 DE NOVEMBRO DE 2009

# Sumário

<b>Introdução</b>	<b>3</b>
<b>Observações Importantes</b>	<b>4</b>
<b>1 Redes Tolerantes a Atrasos[OMR<sup>+</sup>07]</b>	<b>5</b>
1.1 Premissas . . . . .	5
1.1.1 Representação Simplificada de Redes Estáticas . . . . .	5
1.1.2 Redes Móveis . . . . .	5
1.2 Origem . . . . .	6
1.3 Definição . . . . .	7
1.4 Representação . . . . .	7
1.5 Situações-Problema no contexto das <i>DTNs</i> . . . . .	8
<b>2 Protocolos de Disseminação</b>	<b>10</b>
2.1 Probabilísticos . . . . .	10
2.1.1 Motivação . . . . .	10
2.1.2 Dificuldades . . . . .	11
2.1.3 Algoritmos . . . . .	11
2.2 Não-Probabilísticos . . . . .	13
2.2.1 Motivação . . . . .	14
2.2.2 Dificuldades . . . . .	14
2.2.3 Algoritmos . . . . .	14
<b>3 O Simulador <i>ONE</i>[KOK09]</b>	<b>18</b>
3.1 Funcionamento e Características . . . . .	18
3.2 Detalhes da Implementação . . . . .	20
3.3 <i>Interface</i> . . . . .	21
<b>4 Aplicações na Comunicação interveicular</b>	<b>23</b>
4.1 Dispositivos . . . . .	23
4.2 Proposta . . . . .	24
4.3 Simulações . . . . .	25
4.3.1 Representando mapas com <i>OpenJUMP</i> [ope] . . . . .	25
4.3.2 Dificuldades . . . . .	27
4.3.3 Cenários de Simulação . . . . .	28

4.3.4	Resultados . . . . .	29
<b>5</b>	<b>Resumo das Atividades Realizadas</b>	<b>32</b>
<b>A</b>	<b>Parte Subjetiva - Caio Cestari Silva</b>	<b>34</b>
A.1	Motivação . . . . .	34
A.2	Disciplinas Relevantes . . . . .	34
A.3	Facilidades e Dificuldades . . . . .	35
A.4	Projetos Futuros . . . . .	36
<b>B</b>	<b>Parte Subjetiva - Adriano Tabarelli</b>	<b>37</b>
B.1	Motivação . . . . .	37
B.2	Disciplinas Relevantes . . . . .	37
B.3	Facilidades e Dificuldades . . . . .	38
B.4	Projetos Futuros . . . . .	39

## Introdução

Mobilidade: palavra-chave que rege uma crescente parcela das comunicações do mundo contemporâneo. As trocas de informações não apenas seguem a estrutura estática, fixa ao longo do tempo - tudo é dinâmico, se movimenta, tem novas regiões de alcance, perde conexão com certas áreas, troca dados com múltiplas origens e destinos, desconecta-se, é inserido em uma nova rede...

Comunicações interesaciais são um exemplo disso; nem sempre é possível trocar dados com um satélite que orbita um certo planeta, pois pode não haver sinal de transmissão e recepção para que a comunicação ocorra com sucesso. Porém, é possível determinar certos períodos em que dados podem ser recebidos e enviados sem interrupções, devido ao caráter cíclico e conhecido das órbitas e dos movimentos de rotação e translação dos astros. Nesse cenário, nota-se a intermitência que pode ocorrer com as conexões.

As antigas modelagens para as redes, com as conexões e os nós conhecidos, fixos, imutáveis, já são insuficientes para expressar o novo contexto que surge devido à mobilidade. Entra em cena um novo elemento, fator que passa a alterar inúmeras vezes a estrutura de uma mesma rede: o tempo. Dessa forma, somando conexões ao tempo e à instabilidade, obtemos uma representação que consegue aliar os três fatores e representá-los de forma concisa e teorizada: as Redes Tolerantes a Atrasos e Desconexões (*DTNs*), nas quais é possível visualizar todos os elementos de uma rede num determinado instante de tempo e determinar a dinâmica de conexões ali existente.

Caracterizada a nova representação para as redes, um outro cenário de mobilidade muito estudado e aplicado de forma realista é o da comunicação interveicular. Nele, o objetivo é transmitir a veículos dados sobre condições do trânsito em rodovias, ocorrência de acidentes, informações sobre condições adversas na pista, dentre outros; como o objetivo é atingir o maior número de veículos no menor tempo possível, temos um novo elemento neste cenário: o roteamento (encaminhamento) dos dados.

Como modelar a situação de uma rodovia utilizando-se os conceitos das Redes Tolerantes a Atrasos e Desconexões? Quais os elementos que dificultam essa modelagem? Qual o melhor método de rotear os dados para uma melhor abrangência? São necessários nós externos ao sistema para garantir maior confiabilidade no espalhamento dos dados? As tecnologias existentes conseguem garantir transmissão confiável e rápida de informações?

Todas essas questões, que emergem com o estudo aprofundado desse tema, procuraram ser respondidas de forma heurística, com base teórica, e os dados obtidos são apresentados nesse Trabalho de Conclusão de Curso.

## Observações Importantes

A origem do estudo do tema deste Trabalho de Conclusão de Curso é um grupo de Iniciação Científica, de mesmo professor orientador. O projeto é composto pelos dois integrantes que compuseram esta monografia, além de Cássia Garcia Ferreira, Paulo Henrique Floriano e César Gamboa Machado, que compuseram outras monografias com o mesmo tema-base, aprofundando-se em outros assuntos específicos.

Os dados técnicos reais que levaram aos resultados obtidos são fruto de uma parceria com o IPT (Instituto de Pesquisas Tecnológicas), vinculado à bolsa FIPT (Fundação de Apoio ao Instituto de Pesquisas Tecnológicas), por intermédio do pesquisador Alessandro Santiago.

Alguns resultados parciais obtidos com a pesquisa, além dos constantes nesta monografia, conceberam um artigo apresentado na Lapônia, em agosto de 2009, e tiveram apoio da professora doutora Luciana Arantes, da Universidade Paris 6, na França. Os resultados foram apresentados em um *Workshop* de Comunicação Extrema, o *ExtremeCom* 2009, somado a vários outros tópicos sobre Redes Tolerantes a Atrasos e Desconexões.

Antes do término de 2009, outros artigos pretendem ser enviados ao Simpósio Brasileiro de Computadores e Sistemas Distribuídos (SBRC); um destes é específico sobre o tema desta monografia e outro abrange algoritmos desenvolvidos para serem aplicados às *DTNs* e, possivelmente, uma análise do compromisso entre eles.

# 1 Redes Tolerantes a Atrasos[OMR<sup>+</sup>07]

## 1.1 Premissas

Alguns conceitos e idéias são necessários para que se compreenda o ambiente das Redes Tolerantes a Atrasos, e estas premissas têm o papel de preencher qualquer lacuna teórica existente nesse sentido.

### 1.1.1 Representação Simplificada de Redes Estáticas

Uma rede convencional (estática) pode ser representada por um simples grafo. Cada dispositivo é representado por um nó e a existência de conexão entre um par de nós é representada por uma aresta. Devido à perenidade da rede, um grafo que a representa é constante ao longo do tempo. Como exemplo, tomemos uma rede no modelo Cliente-Servidor, em que os clientes não trocam dados diretamente entre si; seja o conjunto  $\mathcal{C}$  formado pelos clientes  $a$ ,  $b$ , e  $c$ , e seja  $\mathcal{S}$  o conjunto unitário dos servidores, formado apenas pelo servidor  $s$ . O grafo  $\mathcal{G}$  formado pelos nós  $N = \mathcal{S} \cup \mathcal{C}$  e pelas arestas  $E = \{a \leftrightarrow s, b \leftrightarrow s, c \leftrightarrow s\}$ , ou seja,  $\mathcal{G} = (N, E)$ , é representado abaixo:

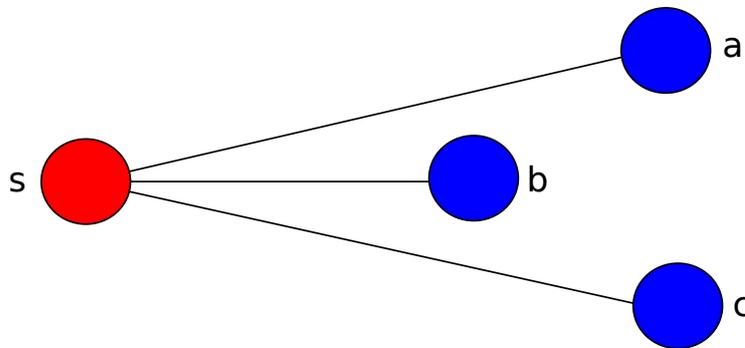


Figura 1: Grafo  $\mathcal{G}$  representando uma simples rede Cliente-Servidor

### 1.1.2 Redes Móveis

Há dois tipos de redes móveis existentes: as infra-estruturadas e as *ad-hoc*, diferenciando-se basicamente pelo nível de conhecimento da rede por cada nó.

Ao passo em que nas redes infra-estruturadas existe um servidor de dados que provê informações sobre a localização de todos os nós para algum elemento que as necessite, as redes *ad-hoc* são caracterizadas pelo conhecimento local de cada nó sobre seus vizinhos, podendo apenas comunicar-se com estes para transmitir dados.

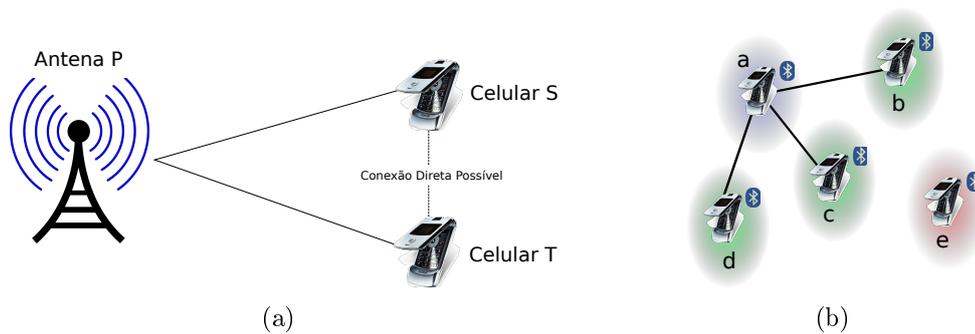


Figura 2: Celulares em Redes Infra-Estruturadas e *Ad-Hoc*

No exemplo acima, embora o celular  $S$  possa comunicar-se via voz (trocar dados) com o celular  $T$  diretamente, pois estão muito próximos, é necessário o intermédio da antena  $P$  para que  $S$  saiba a localização exata de  $T$  na rede em que estão inseridos. Já para a transmissão via *Bluetooth*, que usa ondas de rádio de curto alcance para enviar dados, isto não é necessário.

## 1.2 Origem

A evolução das redes de computadores nas últimas décadas foi tão importante quanto a própria evolução dos computadores. A crescente necessidade de comunicação entre máquinas impulsionou o desenvolvimento das redes, com a implementação de novos meios físicos de transmissão de informação e a utilização de protocolos de comunicação adequados.

Nas redes clássicas ou convencionais, cada computador (ou qualquer dispositivo eletrônico habilitado a trocar informações com outros dispositivos) depende de uma continuidade de conexão com outro computador para que ocorra a comunicação entre eles; a perenidade da conexão é possibilitada pela inexistente mobilidade dos dispositivos a ela conectados.

Contudo, os dispositivos que possuem alta mobilidade são muito comuns, e o modelo clássico de rede não se aplica a eles de forma adequada. Como os nós móveis podem sofrer conexões e desconexões bastante frequentes devido a diversos fatores (distância, queda de energia, defeitos, movimentação aleatória, etc.), é necessária a criação de um modelo de rede que suporte de forma satisfatória essas constantes conexões/desconexões entre os dispositivos. É nesse contexto que surgem as Redes Tolerantes a Atrasos.

### 1.3 Definição

Uma *DTN*, *Delay-And-Disruption Tolerant Network*, ou Rede Tolerante a Atrasos e Desconexões, é uma rede *ad-hoc*, ou *MANET* (*Mobile Ad-Hoc Network*), composta por conexões sem fio entre os nós, que se auto-configura de acordo com a inserção e remoção de novos nós em sua estrutura. Além disso, consegue comportar-se de forma adequada mesmo com desconexões repentinas ou atrasos na transmissão de dados entre dois nós.

Há três principais mecanismos responsáveis por tornar adequado o comportamento de uma DTN:

- **Custódia de mensagens:** se uma mensagem  $M$  precisa ser enviada de uma origem  $s$  até um destino  $t$ , que não está disponível no mesmo instante de tempo do envio, tal mensagem é custodiada (entregue como cópia) a nós intermediários que se encarregam de entregá-la ao nó de destino (ver abaixo);

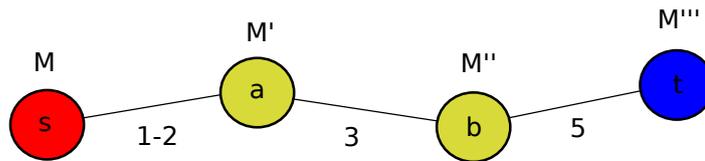


Figura 3: Custódias da mensagem  $M$  em  $M'$  e  $M''$ , nos nós  $a$  e  $b$ , até entrega no instante 5

- **Protocolos de roteamento específicos:** devido ao baixo ou nulo conhecimento da rede em cada instante, os protocolos de roteamento são diferenciados, assumindo estratégias de custódia fixas ou ainda analisando o comportamento da rede para prever as melhores custódias a cada instante, e;
- **Periodicidade:** pelo caráter aleatório da movimentação dos nós, é necessário assumir alguma periodicidade em seu comportamento para que haja alguma representação num intervalo de tempo finito da rede, isto é, para que não haja infinitas configurações em um mesmo sistema.

### 1.4 Representação

Os grafos tradicionais, que conseguem representar a situação das redes estáticas, já não mais contêm elementos que podem representar com fidelidade o cenário das *DTNs*. Como o fator principal de alteração da topologia da rede é o tempo, insere-se esse elemento na

estrutura de um grafo tradicional, resultando em um **grafo evolutivo**. Sua estrutura, então, é composta por:

- **Nós** - Elementos receptores, transmissores e armazenadores de informação;
- **Arestas** - Conexões entre os nós, vias de transmissão de informações, e;
- **Tempo** - Caracterizando os períodos de existência das arestas.

Em termos mais teóricos, um grafo evolutivo é uma coleção de subgrafos de um certo grafo, indexada, representando a situação das arestas e dos nós naquele dado índice de tempo. Denota-se um grafo evolutivo por  $\mathcal{G} = (G, S_G)$ , onde  $G = (V, E)$  é um grafo com  $V$  nós e  $E$  arestas,  $S_G = \{G_0, G_1, \dots, G_i\}$ , ( $i \in \mathbb{N}$ ), é um conjunto de subgrafos de  $G$ , onde o índice  $i$  representa o instante de tempo, e a união de todos os subgrafos resulta em  $G$ .

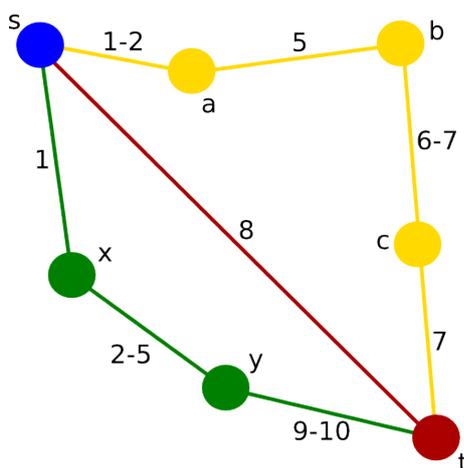


Figura 4: Representação de um grafo evolutivo

## 1.5 Situações-Problema no contexto das *DTNs*

Em diversas situações reais os conceitos de Redes Tolerantes a Atrasos podem ser empregados. Apesar da vasta abrangência da Internet atualmente, diversas regiões do mundo ainda permanecem relativamente isoladas quando se trata de conectividade com redes maiores ou com a própria Internet. Fatores econômicos e físicos podem contribuir para que uma região remota não disponha de acesso a rede de forma contínua; apesar disso, o acesso à informação e à comunicação com outras localidades é de natural interesse dos povos que vivem em regiões remotas. Uma alternativa encontrada para contornar tal problema de isolamento é a mula de dados, que nada mais é do que um meio de transporte

qualquer, capaz de carregar algum dispositivo de armazenamento de dados e deslocá-lo entre várias localidades, isoladas ou não. As informações são carregadas pelo dispositivo de armazenamento em uma localidade de origem e transportadas até uma localidade de destino.

Outra situação em que as Redes Tolerantes a Atrasos são de grande relevância é na comunicação interestelar. A conexão com satélites em órbita, estações, naves espaciais, sondas interplanetárias e telescópios depende da movimentação destes elementos e dos próprios movimentos dos corpos celestes. As rotações e translações de planetas, estrelas e satélites naturais podem eventualmente gerar conexões e desconexões com centros localizados na Terra. Este cenário de mobilidade e frequentes conexões/desconexões é típico das Redes Tolerantes a Atrasos.

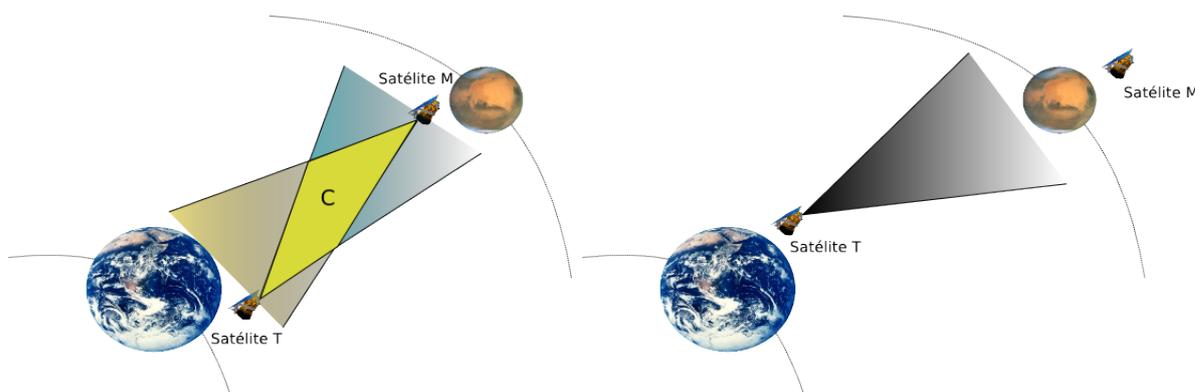


Figura 5: Diferentes situações de conexão no ambiente interestelar

O escopo do nosso trabalho, entretanto, encontra-se na comunicação interveicular. Já existem tecnologias disponíveis com a capacidade de transmitir informações de um meio externo para um veículo qualquer. No entanto, o desafio e alvo de diversos estudos atualmente é a comunicação entre os próprios veículos, atuando tanto como receptores quanto retransmissores e/ou geradores de informações. A utilidade de uma tecnologia deste tipo é dinamizar a transmissão de dados básicos e até mesmo críticos do tráfego rodoviário, tais como localização de acidentes, danos na pista, condições meteorológicas, etc. Como os veículos têm uma alta mobilidade, a comunicação entre eles também é passível de frequentes conexões/desconexões, caracterizando um ótimo cenário para a aplicação dos conceitos de Redes Tolerantes a Atrasos.

## 2 Protocolos de Disseminação

Em redes intermitentes, os protocolos clássicos de transmissão de dados são pouco eficientes, pois dependem de uma estabilidade da conexão fim-a-fim entre dois nós. No protocolo TCP/IP, por exemplo, a camada de transporte da rede depende da constante conexão entre transmissor e receptor, tanto para envio de pacotes quanto para confirmação de recebimento de pacotes. Sabemos que em redes intermitentes essa constância de conexão entre transmissor e receptor não é um evento muito frequente e, portanto, outros protocolos de transmissão devem ser considerados a fim de garantir taxas satisfatórias de entrega de pacotes.

Diversos protocolos de transmissão de dados foram desenvolvidos visando superar as dificuldades inerentes às redes intermitentes. Tais protocolos podem ser divididos basicamente em dois grupos: probabilísticos e não-probabilísticos; cada grupo possui características comuns, embora cada protocolo seja distinto em suas funcionalidades, como taxas de entrega, utilização de *buffer*, controle de quantidade de cópias da mesma mensagem, etc.

### 2.1 Probabilísticos

Os protocolos probabilísticos se baseiam em dados estatísticos calculados, principalmente, a partir de encontros entre nós, quantidade de retransmissões de uma mensagem, tempo de existência dos pacotes na rede e da quantidade de confirmações de recebimento emitidas por um destinatário. Os diferentes algoritmos deste grupo utilizam tais dados para aplicar critérios e funções particulares na comunicação entre nós da rede, optando pelo melhor deles para realizar a custódia da mensagem.

#### 2.1.1 Motivação

Uma constante preocupação dos protocolos probabilísticos é de utilizar os dados da rede ao seu favor, de maneira a otimizar taxas de entrega e criação de cópias de uma mesma mensagem. Por agirem dinamicamente, efetuando cálculos de probabilidade a cada instante, estes protocolos acabam por ter comportamentos satisfatórios em termos práticos. Comparados a algoritmos que não utilizam estatísticas da rede ou pouco utilizam, estes algoritmos apresentam melhores resultados empíricos no que se refere a taxas de entrega, quantidade de cópias e retransmissões de mensagens.

### 2.1.2 Dificuldades

Como mencionado anteriormente, os algoritmos probabilísticos dependem de dados estatísticos da rede para a aplicação de seus critérios particulares. Em uma rede recém implementada, estes dados não existem e então são aumentadas consideravelmente as possibilidades de erro e mensagens perdidas. O tempo de adaptação ou tempo de *warmup* da rede é crucial para a eficiência destes protocolos.

### 2.1.3 Algoritmos

Os algoritmos probabilísticos mais conhecidos e utilizados são:

#### ***Prophet***[LDS04]:

O algoritmo *Prophet* (***P*robabilistic **R**outing **P**rotocol using a **H**istory of **E**ncounters and **T**ransitivity**) utiliza-se do histórico de encontros entre nós para tomada de decisões no roteamento. Cada nó da rede armazena uma lista com todos os nós destinatários conhecidos, ou seja, todos os nós que já foram encontrados. A cada nó destinatário conhecido é então atribuída uma *delivery predictability* ou previsibilidade de entrega, que é inicialmente igual a zero e posteriormente recalculada durante um encontro. O cálculo da previsibilidade de entrega de um nó **A** para um nó **B** segue a seguinte fórmula:

$$P_{(A,B)} = P_{(A,B)_{old}} + (1 - P_{(A,B)_{old}}) * p_{ini}$$

em que  $0 \leq p_{ini} \leq 1$  é uma constante de inicialização.

Um mecanismo que também é utilizado no tratamento de previsibilidades de entrega é o envelhecimento. A previsibilidade de entrega entre um par de nós é decrementada de acordo com o tempo decorrido desde o último encontro. A seguinte equação de envelhecimento é aplicada sempre que um novo encontro ocorre:

$$P_{(A,B)} = P_{(A,B)_{old}} * \gamma^K$$

em que  $0 \leq \gamma \leq 1$  é a constante de envelhecimento e  $K$  é quantidade de unidades de tempo decorridas desde o último encontro entre **A** e **B**.

Em situações que um par de nós raramente se encontra na rede, mas possui um nó intermediário com que ambos se encontram frequentemente, é então aplicada a propriedade de **transitividade**. Desde que exista um nó intermediário que se encontre frequentemente com um determinado par de nós, este par assume uma maior relevância na previsibilidade de entrega de um para o outro. Segue a equação de transitividade:

$$P_{(A,C)} = P_{(A,C)_{old}} + (1 - P_{(A,C)_{old}}) * P_{(A,B)} * P_{(B,C)} * \beta$$

em que **B** é o nó intermediário entre um par de nós **A** e **C**, e  $0 \leq \beta \leq 1$  é a constante de relevância da propriedade de transitividade.

**MaxProp[BGJL06]:**

O algoritmo *MaxProp* baseia-se na priorização de pacotes para transferência durante as oportunidades de encontro. Para tanto, cada nó da rede mantém uma estrutura de fila ordenada para os pacotes, cuja ordenação funciona de acordo com a *delivery likelihood* (probabilidade de entrega) de cada pacote.

Além da própria fila, cada nó da rede mantém uma tabela de todos os outros nós destinatários, contendo o peso de cada um destes. Este peso representa a frequência de um encontro e é inicialmente igual para todos os nós destinatários; porém, na medida em que ocorrem os encontros, o peso de um nó pode ser incrementado. Supondo uma rede com apenas cinco nós, um determinado nó armazenará uma lista contendo o peso dos outros quatro nós da rede, sendo inicialmente cada peso igual a 0,25 (o somatório de todos os pesos deve ser igual a 1). Quando este determinado nó encontrar um dos outros nós, o peso do nó encontrado é incrementado em 1 e todos os pesos da tabela são normalizados de forma que a soma de todos volte a ser 1.

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
0,25	0,25	0,25	0,25

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
$\frac{0,25}{2}$	$\frac{0,25}{2}$	$\frac{0,25 + 1}{2}$	$\frac{0,25}{2}$

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
0,125	0,125	0,625	0,125

Figura 6: Normalização de pesos após um encontro com o nó **3**

Durante o encontro de um par de nós, cada nó têm acesso à tabela de pesos do outro, e estas são trocadas entre eles. Em suma, cada nó possui, além de sua própria tabela de pesos, uma tabela aproximadamente atualizada de todos os outros nós e a posse destas tabelas é que possibilita o cálculo das probabilidades de entrega. Para isto, é realizada uma busca em profundidade a partir do nó de origem, levando em conta todos os caminhos possíveis até o destinatário, sendo que o custo de um caminho entre um par de nós **A** e **B**

equivale a  $1 - \text{peso}_A(B)$ , com  $\text{peso}_A(B)$  sendo o peso que **B** possui na tabela de **A**. Após o cálculo do custo de todos os caminhos, toma-se o menor valor calculado para atribuir o valor de *delivery likelihood* para o pacote.

Além do mecanismo de priorização de pacotes, o *MaxProp* possui diversos outros mecanismos secundários:

- Durante um encontro, todas as mensagens destinadas ao nó vizinho são transferidas. Ocorre também a troca de *acknowledgements* (confirmações de recebimento emitidas por um destinatário) para uma eventual limpeza de *buffer*;
- Pacotes novos (que possuem poucos *hops*) têm prioridade nas transferências, e;
- Não há repetição de envio de um mesmo pacote para um mesmo nó e também não há recebimento de pacotes que já existem no *buffer*.

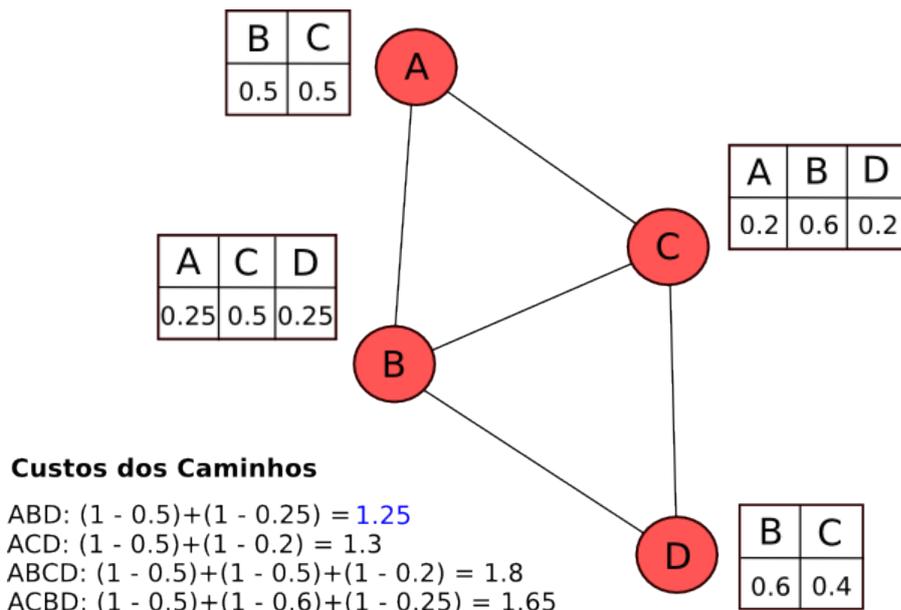


Figura 7: Representação de custo de caminhos no protocolo *MaxProp*

## 2.2 Não-Probabilísticos

Os protocolos não-probabilísticos têm como característica principal o espalhamento de pacotes. Tais protocolos supõem que quanto mais cópias de um mesmo pacote forem espalhadas na rede, maior será a probabilidade de entrega dos dados deste pacote, ou seja, aumentam-se as chances de um encontro com o destinatário quando aumenta-se a proporção de nós com os dados a serem entregues.

### 2.2.1 Motivação

Uma das primeiras idéias que surgem ao pensar no problema de entrega de pacotes em uma rede móvel é o espalhamento de cópias em diversos nós; confiando na mobilidade da rede, provavelmente encontra-se o destinatário. Os protocolos não-probabilísticos possuem políticas simples de roteamento, o que torna intuitiva a implementação dos mesmos.

### 2.2.2 Dificuldades

Quando se pensa em redes móveis reais, em que existe uma considerável limitação nos *buffers* dos nós e da taxa de transmissão dos mesmos, os protocolos não-probabilísticos começam a perder eficiência. Como são baseados quase em sua totalidade no espalhamento de mensagens, estes protocolos podem eventualmente gerar uma inundação de mensagens na rede, ocasionando estouros de *buffer* e um *overhead* de transmissão de pacotes (muitos pacotes a serem transmitidos num curto tempo de conexão entre pares de nós).

### 2.2.3 Algoritmos

Os algoritmos não-probabilísticos mais conhecidos e utilizados são:

#### **Epidêmico[VB00]:**

Antes do contexto das redes móveis, o algoritmo epidêmico podia ser resumido em três características: robusto, escalável e confiável; tais palavras se devem ao seu mecanismo de transmissão de mensagens, semelhante a uma doença em epidemia, contaminando todos os vizinhos a partir de um indivíduo (nó) “infectado”. Era certo que, em algum instante, todos os nós do sistema estariam contaminados (confiabilidade), independente do tamanho do mesmo (escalabilidade) e de onde a infecção houvesse iniciado (robustez).

O objetivo do algoritmo epidêmico é a entrega de uma mensagem a um destinatário específico - porém, para isso, torna a mensagem conhecida para uma grande parcela do sistema. Basicamente, a cada instante de tempo determinado, conhecido como *round*, ocorre a contaminação dos vizinhos, com uma certa probabilidade  $p$  e, dessa forma, caracterizamos um algoritmo epidêmico por quatro parâmetros:

- **Tamanho do Sistema ( $N$ ):** número de nós do sistema;
- **Número Máximo de Transmissões ( $T$ ):** quantidade máxima de custódias até o destino, ou ainda *TTL* (*time-to-live*) das mensagens;

- **Fanout** ( $F$ ): número máximo ou proporção de vizinhos que serão contaminados por um nó infectado a cada *round*,  $e$ ;
- **Buffer** ( $B$ ): tamanho do *buffer* de cada nó, responsável por armazenar as mensagens.

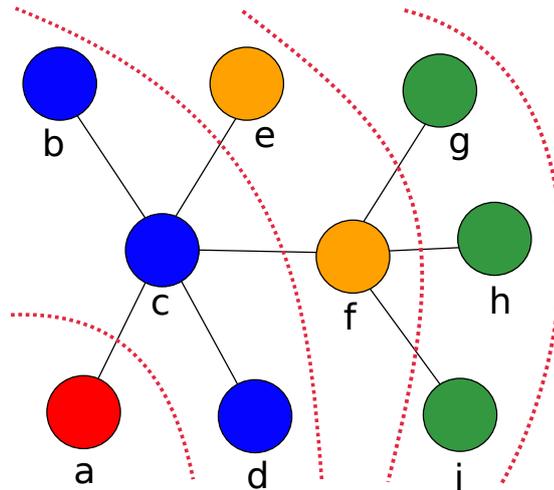


Figura 8: Contaminação epidêmica (com  $F = T = \infty$ ) de uma rede estática em 4 *rounds*

O desafio nas redes móveis é manter a escalabilidade, confiabilidade e robustez do algoritmo, pois situações específicas são grandes “gargalos” nesse processo. Se uma mensagem for gerada, por exemplo, por um nó que raramente se comunica com outros nós do sistema, prejudica-se bastante a confiabilidade do sistema - necessita-se de um maior intervalo de tempo para certeza de entrega da mensagem; porém, a mesma situação em um ambiente mais denso (com mais nós e conexões) tem mais chance de obter sucesso na entrega, o que compromete a escalabilidade e, num todo, a robustez do algoritmo.

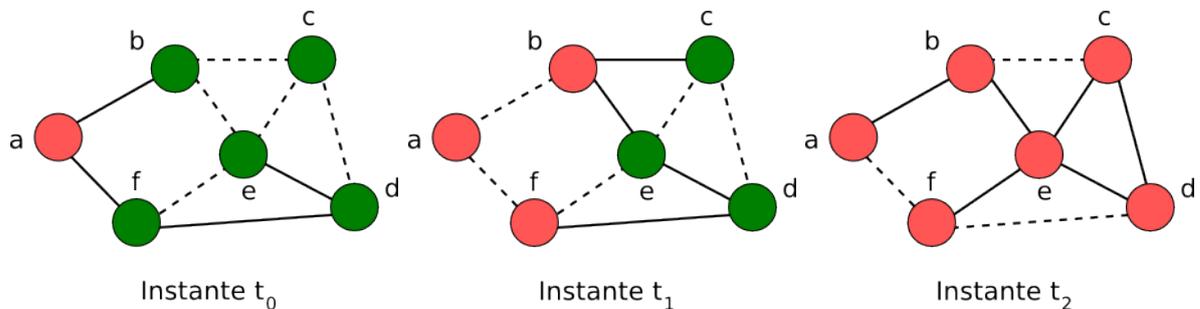


Figura 9: Contaminação epidêmica em uma *DTN*, com as desconexões tracejadas

Devido à alta troca de mensagens, o algoritmo tende a congestionar a rede, o que é resolvido nas DTNs com o uso de *beacons*: pequenos indicadores da presença da mensagem no buffer do nó de destino, para evitar retransmissões desnecessárias. Além disso, outra questão é o tamanho do *buffer* e como os nós lidam com seu estouro; algumas políticas priorizam mensagens mais antigas, outras priorizam mensagens mais novas, e outras ainda analisam parâmetros da própria mensagem, como o seu *TTL* (*time-to-live*), e decidem por aceitar ou rejeitar a nova mensagem a ser transmitida.

### ***Spray-and-Wait* [SPR05]:**

O algoritmo *Spray-and-Wait* foi desenvolvido com o intuito de limitar a quantidade de cópias de uma mesma mensagem na rede, diminuindo a probabilidade de ocorrência de estouros de *buffer* e congestionamento.

Seu funcionamento depende principalmente da determinação de um parâmetro  $L$ , que representa o número máximo de nós distintos a receberem a cópia de uma determinada mensagem. Exemplificando, para cada mensagem originada são atribuídos  $L$  *tokens*; o nó possuidor da mensagem original irá, nas próximas  $L$  oportunidades distintas de encontro, transmitir uma cópia da mensagem para outro nó, decrementando o valor de  $L$  a cada transmissão bem sucedida. Após o consumo desses  $L$  *tokens*,  $L + 1$  nós serão portadores de uma mesma mensagem, e estes aguardarão um encontro direto com o destinatário para enfim realizar a entrega.

O cálculo exato do valor de  $L$  depende de valores como o número de nós da rede e o tempo esperado de uma transmissão direta ao destinatário. Entretanto, valores satisfatórios de  $L$  estão em torno de 10% do total de nós da rede.

A maneira de se realizar a *spray phase*, ou seja, de se distribuir as  $L$  cópias, pode variar; existem dois principais métodos, sendo eles o *source spraying* e o *binary spraying*. No primeiro método, o nó de origem da mensagem fica responsável por distribuir todas as  $L$  cópias; já no segundo, o nó de origem repassa metade de seus *tokens* para o nó seguinte e decrementa esse valor da totalidade dos *tokens* que possui. O nó seguinte repete o mesmo procedimento, até que a quantidade de *tokens* da mensagem no nó seja igual a 1, entrando na *wait phase*.

É possível notar que o algoritmo depende de uma rede com mobilidade consideravelmente alta, pois em redes com nós pouco móveis aumentam-se as chances de um destinatário permanecer isolado por muito tempo.

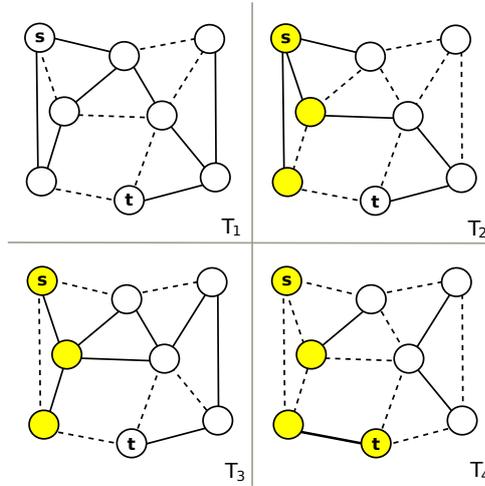


Figura 10: Representação do algoritmo *Spray-And-Wait* com  $L = 2$

***Spray-and-Focus*[SPR07]:**

O algoritmo *Spray-and-Focus* se assemelha ao algoritmo *Spray-And-Wait* em sua primeira fase, ou seja, na fase de distribuição de mensagens, ou *spray phase*. O protocolo também se baseia em um parâmetro  $L$  de quantidade de cópias a serem distribuídas e também possui as duas maneiras de fazer essa distribuição (*source spraying* e *binary spraying*).

A diferença se encontra na segunda fase (*focus phase*). Neste protocolo, cada nó armazena uma lista que contém o tempo decorrido desde o último encontro com os outros nós da rede. Após a *spray phase*, os nós que possuem as cópias de uma mensagem podem eventualmente reencaminhá-las a outro nó, caso este outro nó tenha tido um contato mais recente com o nó destinatário.

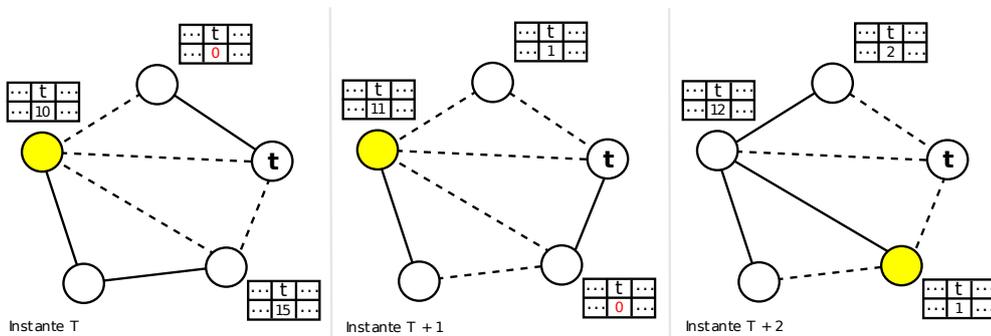


Figura 11: Exemplo de aplicação da *focus phase*

## 3 O Simulador *ONE*[KOK09]

O *Opportunistic Network Environment*, ou *ONE*, é uma ferramenta *open source*, desenvolvida em *Java*, que simula de forma exemplar uma *DTN*. Implementando todos os mecanismos necessários ao funcionamento de uma Rede Tolerante a Atrasos, como as custódias e os algoritmos de disseminação de mensagens, por exemplo, 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.

### 3.1 Funcionamento e Características

O funcionamento do *ONE* depende inicialmente da construção de um cenário de simulação por parte do usuário. Um cenário pode ser construído por meio da inserção de diversos elementos, dentre eles:

- Grupos de nós
- Eventos
- Protocolos de roteamento
- Modelos de movimentação
- Relatórios
- Mapas

Cada grupo de nós possui características próprias, tais como velocidade de deslocamento, taxa de transmissão, tamanho de *buffer*, quantidade de nós e identificador (*ID*). Diversos grupos podem ser inseridos no cenário, possibilitando uma variedade de nós presentes no ambiente de simulação.

Os eventos a serem gerados na simulação nada mais são 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 e recebê-los.

Os protocolos de roteamento podem ser aplicados de maneira global ou local no ambiente de simulação, ou seja, cada grupo distinto de nós pode possuir seu próprio protocolo de roteamento. No *ONE*, por padrão, já estão implementados os protocolos *Epidemic*, *Spray-And-Wait*, *MaxProp* e *PRoPHET*.

Os modelos de movimentação são padrões de deslocamento de nós e podem também ser aplicados de forma global ou local aos grupos de nós. Dentre os diversos modelos, merecem destaque:

- ***CarMovement***: modelo para carros, no qual o nó possui um comportamento de seguir para um destino a partir de uma origem;
- ***BusMovement***: modelo para ônibus, no qual o nó possui um comportamento que alterna origem e destino continuamente (comportamento *ping-pong*);
- ***ShortestPath***: modelo em que o nó possui dois pontos de origem e destino, e a menor rota entre tais pontos é calculada usando o algoritmo de *Dijkstra*, e;
- ***RandomWaypoint***: modelo em que são criados diversos caminhos aleatórios entre pares de pontos, com o nó se deslocando por estes caminhos.

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 seu destinatário, e;
- **Conectividade da 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 simulador *ONE*, é utilizado um formato específico de representação de mapas, o formato WKT. Basicamente, a estrutura de funcionamento do *ONE*, baseando-se no fluxo dos dados, é representada abaixo:

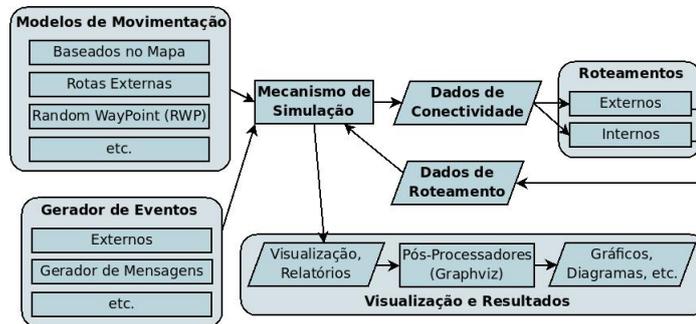


Figura 12: Estrutura de funcionamento do ONE

### 3.2 Detalhes da Implementação

Importando-se o projeto do *ONE* para o *Eclipse* (uma ferramenta gerenciadora de projetos em *Java*) e verificando-se sua estrutura, nota-se que há diversos pacotes que o compõem; dentre eles, os principais são:

- **core**: responsável pelos mecanismos essenciais do simulador, lidando com conexões (classe *Connection*), mensagens (classe *Message*), movimentos (classe *MovementListener*), configurações (classe *Settings*) e posições dos objetos (classe *World*);
- **data**: contém os dados dos mapas utilizados nas simulações;
- **doc**: documentação do projeto, completa e detalhada;
- **gui**: controle de toda a interface gráfica do simulador;
- **input**: toda a leitura de dados externos ao simulador, como os mapas, o arquivo de configuração, os roteamentos externos, etc. é tratada por este pacote;
- **movement**: abrange todos os modelos de movimentação, estendendo a classe abstrata *MovementModel* e implementando outra classe, caso necessário;
- **report**: geração de todos os relatórios que podem ser configurados no cenário, com base nas estatísticas de conectividade, mensagens, duração da simulação, dentre outras;
- **routing**: com as classes de roteamento, este pacote caracteriza-se por ter as classes estendendo *ActiveRouter*; um roteamento ativo faz com que os nós verifiquem a cada troca de mensagem se há espaço no nó de destino, se este é o destinatário final da mensagem, lida com buffer cheio e política de descarte de mensagens, etc.;

- *test*: testes unitários e funcionais para o simulador, e;
- *toolkit*: coleção de utilitários como escaladores de mapas, conversores de formatos de saída e outros *scripts* para *parsing* ou saídas mais amigáveis dos resultados gerados.

### 3.3 Interface

De fácil compreensão, a *interface* do *ONE* é composta por quatro áreas principais: área da simulação, controle da simulação, lista dos nós e *log* de eventos. Abaixo, observe-se a *interface* gráfica e estas quatro áreas:

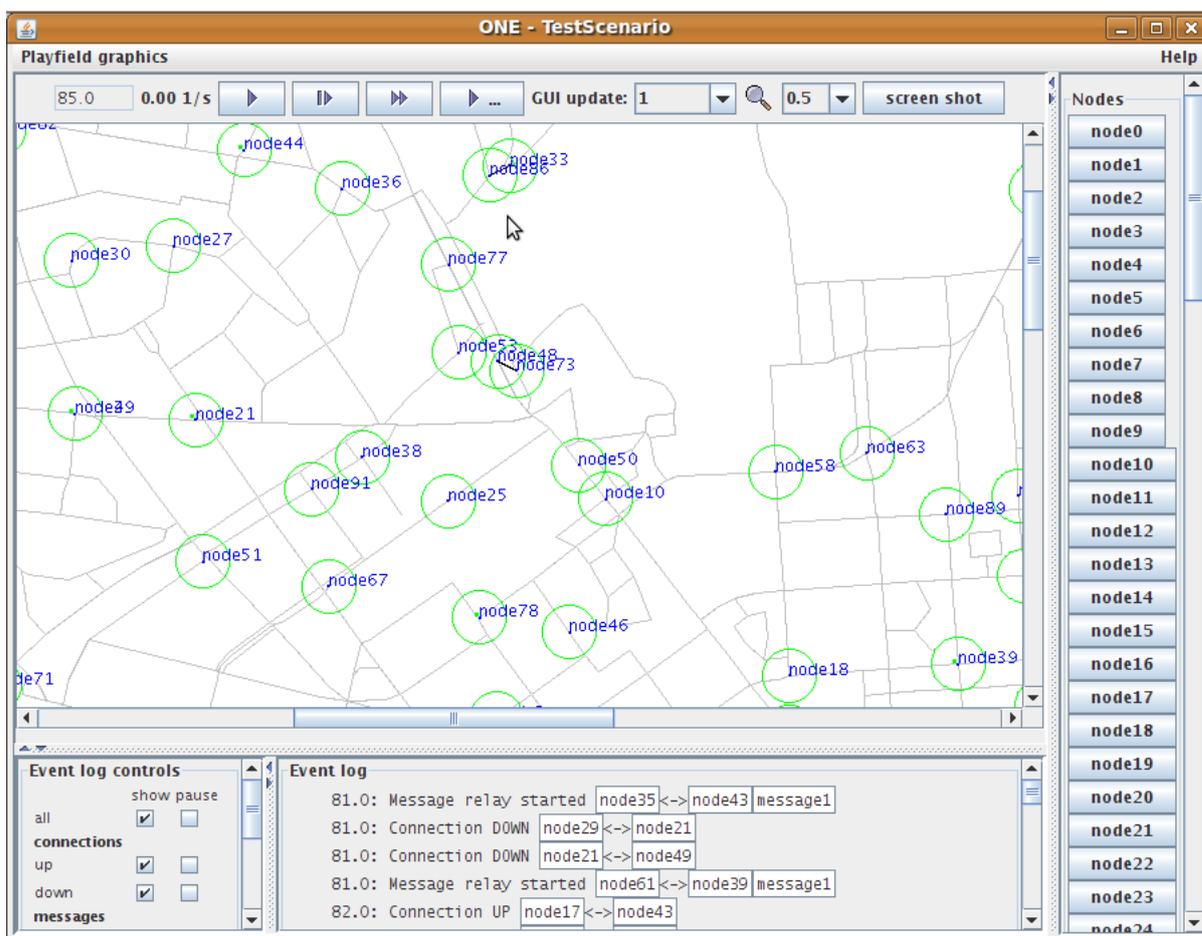


Figura 13: *Interface* gráfica do simulador *ONE*

É possível notar que a simulação pode ser adiantada, pausada ou acelerada, e há também possibilidade de ajustar a velocidade de atualização da *interface* gráfica e o nível de

*zoom* na área de simulação. Todas estas opções estão disponíveis na parte superior da *interface*.

O *log* de eventos, na parte inferior, 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 dois nós. É possível filtrar as mensagens do *log* e ainda acessar informações mais detalhadas sobre algum nó diretamente pela área de eventos.

A área principal exibe os nós, nomeados distintamente, os caminhos em que podem trafegar e o raio de transmissão de cada um deles; é possível notar que, durante a custódia de uma mensagem, os dois nós envolvidos permanecem ligados por um segmento de reta, indicando tal evento.

À direita, situa-se a lista 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. Além disso, a interface gráfica exibe a rota atual do nó em vermelho, como pode ser visualizado na imagem abaixo.

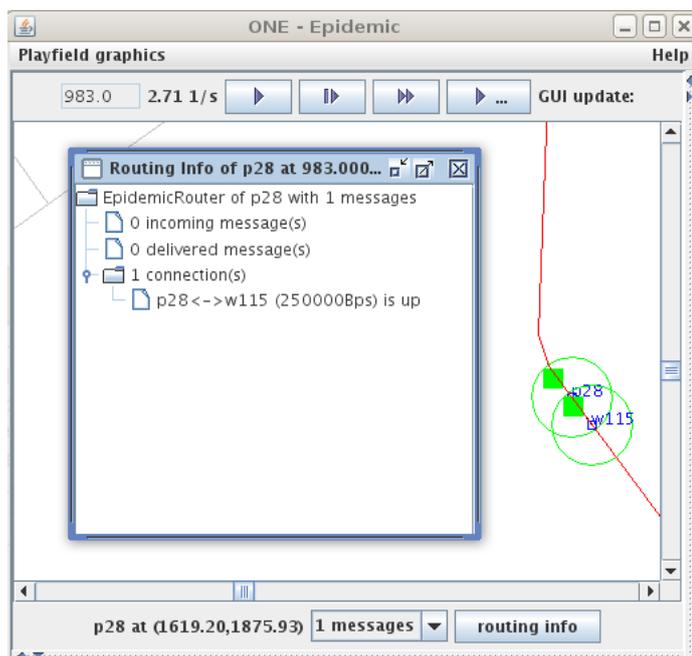


Figura 14: Detalhes de um nó específico no simulador *ONE*, com a rota de movimentação

## 4 Aplicações na Comunicação interveicular

Acompanhando a crescente evolução e acessibilidade dos computadores nas últimas décadas, os sistemas embarcados também vêm se tornando cada vez mais presentes no cotidiano. A aplicação desses sistemas atualmente é muito abrangente, e se estende desde aviões até simples aparelhos eletrodomésticos, o que engloba, obviamente, veículos terrestres. Sistemas embarcados em veículos terrestres já desempenham importantes funções, como trocas de marcha automáticas, controle de mistura de combustível em motores híbridos, navegação por *GPS (Global Positioning System)*, dentre outras. Entretanto, um fato pouco explorado mas que vem se tornando cada vez mais estudado é a comunicação entre veículos.

Diversas aplicações podem ser consideradas quando há a possibilidade de comunicação interveicular, tais como repasse de informações sobre o trânsito, danos na pista, acidentes, meteorologia, dentre outras. A comunicação entre veículos pode tornar o processo de transmissão de informações mais dinâmico, diminuindo a dependência de centros transmissores ao passo em que cada carro tem a possibilidade de transmitir e receber dados. Quando são consideradas características como alcance e velocidade de propagação de informações, a comunicação entre veículos deve ser considerada como uma forma de se otimizar tais características.

### 4.1 Dispositivos

Atualmente são conhecidos vários dispositivos transmissores de informação em uma rede *ad-hoc*; dentre eles, merecem destaque o *Bluetooth*, embarcados na maioria dos telefones celulares, e a *WLAN (Wireless Local Area Network)*, inserida no contexto dos roteadores presentes em muitos locais públicos. Além dessas duas tecnologias, está disponível também a *DSRC (Dedicated Short-Range Communication)*, sobre a *WLAN*, que embora tenha um custo mais elevado, atinge melhores resultados em desempenho e área de alcance do sinal de transmissão.

As características dos três dispositivos citados, em termos de taxa de transmissão de dados e alcance do sinal, são as seguintes:

- ***Bluetooth* de Classe 2** - Alcance médio de 22 metros e taxas de transmissão de até 128kB/s;
- ***Bluetooth* de Classe 3** - Alcance médio de 100 metros e taxa de transmissão de 384kB/s;

- **WLAN** - Alcance médio de 33 metros e taxas de transmissão de até 1,4MB/s na versão 802.11b, e;
- **DSRC** - Alcance médio de 1km e taxas de transmissão de até 2,5MB/s.

Embora haja um *Bluetooth* com classe de características superiores, o mais encontrado nos aparelhos celulares brasileiros é o de Classe 2. Com relação à *WLAN*, há uma maior necessidade de espaço, pois o aparelho que envia e recebe dados, juntamente com a antena, tem dimensões superiores ao pequeno *Bluetooth* embarcado. Para a *DSRC* tem-se uma mesma utilização de espaço, porém as antenas e o equipamento, por fornecerem maior desempenho, têm custo mais elevado.

## 4.2 Proposta

Dado um cenário real de rodovia brasileira, com distâncias reais, fluxos de veículos reais e modelos de movimentação próximos aos reais, estudar o melhor algoritmo de disseminação e o melhor dispositivo para que, gerada uma mensagem de alerta em um carro aleatório, esta atinja o maior número de veículos do sistema em menor tempo. Deseja-se avaliar, também, o impacto de *RSUs* (*RoadSide Units*, ou Unidades ao Longo da Via) na abrangência de veículos e no tempo para espalhamento da mensagem. Graficamente:

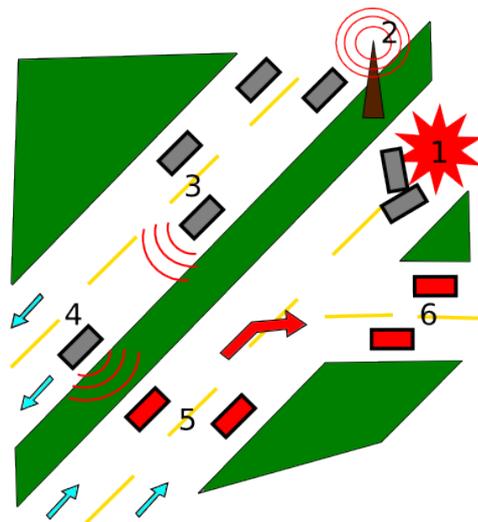


Figura 15: Simulação da transmissão de uma mensagem de alerta em um ambiente rodoviário

Foi escolhido o cenário da região de Ribeirão Preto, correspondendo as rodovias concessionadas Autovias, Intervias, Triângulo do Sol, TEBE e Vianorte como caminhos onde

os nós são movimentados. Estes nós representam fluxos oriundos dos pedágios de cada uma dessas rodovias, fornecidos pelo IPT, com classificação de veículos por classe (carros e caminhões com eixos) e horários da passagem do volume pelas praças de pedágios, onde algumas torres de transmissão foram eventualmente colocadas.

## 4.3 Simulações

Para a aplicação de dados e geração de resultados foi necessária a escolha de uma ferramenta confiável e compatível com o nosso contexto de Redes Tolerantes a Atrasos. O simulador *ONE*, primeiro a ser explorado, ganhou apreço por sua boa estrutura de código e documentação, o que nos permitiu modificações nas classes a nosso contento e de forma simples; em contrapartida, o *SUMO* (Simulation of Urban MObility) também realiza simulações de forma bem satisfatória, embora não tenha uma estrutura de implementação à altura do *ONE*, escolhido em definitivo para obtenção dos resultados.

### 4.3.1 Representando mapas com *OpenJUMP*[ope]

O simulador *ONE* suporta o formato *WKT* (*Well-Know Text*) de representação de mapas. A linguagem *WKT* permite de forma simples e direta descrever objetos geométricos como pontos, retas, polígonos e composições destes mesmos objetos, o que se encaixa de forma adequada na representação de mapas rodoviários em duas dimensões. Um exemplo simples da linguagem *WKT*:

```
POINT (202 598)
POINT (105 485)
POINT (306 484)
POINT (200 399)
LINESTRING (105 485, 200 399, 306 484, 202 598, 105 485)
LINESTRING (202 598, 200 399)
LINESTRING (105 485, 306 484)
```

O exemplo acima é um conjunto de 4 pontos distintos e diferentes retas que os interligam. Segue o mapa produzido a partir deste exemplo:



Como as rodovias da região possuem diferentes concessões de administração e diferentes dados relevantes, foi preciso que cada rodovia fosse representada de forma independente. No *OpenJUMP* é possível a separação do desenho em camadas e a interligação de pontos de camadas distintas, funcionalidade esta que permitiu a modularização do desenho final.

Uma dificuldade do *OpenJUMP* é trabalhar adequadamente com escalas. Porém, esta dificuldade foi contornada com a construção de um script em *Perl* que, a partir de um mapa representado em *WKT*, realiza uma transformação linear para aplicar uma desejada escala, que pode ser obtida a partir de uma relação da escala da imagem de fundo utilizada e a escala utilizada na *interface* do próprio *OpenJUMP*.

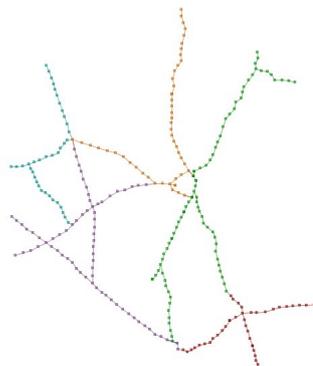


Figura 18: Mapa modularizado no *OpenJUMP*

### 4.3.2 Dificuldades

Ao simularmos os fluxos de todas as rodovias (em torno de 50 mil nós), notamos que a velocidade de simulação reduziu-se consideravelmente, o que nos impediu de utilizarmos todos os veículos e obtermos um resultado global para toda a região. Em vez disso, nos focamos na Rodovia Triângulo do Sol, analisando seus fluxos nos pedágios.

Outro problema foi organizar de forma realista os nós em cada trecho da estrada; por haver três trechos contíguos, com intersecções (a rodovia assemelha-se, de fato, a um triângulo), determinamos os fluxos em cada trecho contíguo, com alguma probabilidade de troca de trecho pelos veículos. Como modelo de movimentação, assumimos o *CarMovement* com parâmetro circular, ou seja, os carros seguem uma rota circular e retornam ao ponto inicial, indefinidamente, até que tomem uma nova direção e continuem a se mover nesta outra rota.

A geração aleatória da mensagem, caso a semente de aleatoriedade não seja alterada,

não confere uma boa gama de resultados, pois as simulações têm um grande viés ao produzirem a mesma sequência pseudo-aleatória de números que determinam o nó gerador do evento (mensagem). Para isso, é necessário alterar a semente de aleatoriedade e calcular a média dos resultados gerados, exibindo também a maior e a menor ocorrência de um resultado dadas diversas sementes aleatórias.

Uma particularidade da mensagem é que ela precisa obrigatoriamente de um destinatário; deste modo, construímos um destinatário *dummy*, em um ponto não alcançável, para todas as mensagens geradas. Como este nunca é atingido, a mensagem não para de se propagar e atingimos os objetivos esperados.

As *DTNs* assumem uma conexão entre dois nós como sendo a possibilidade de envio e recebimento por ambas as partes; portanto, um elemento precisa estar contido no raio de transmissão do outro. Isso dificulta a modelagem de estruturas como as antenas ao longo da via, que apenas repassariam as mensagens aos nós que alcançam, sem necessidade de receberem confirmações ou outros dados. É possível contornar a dificuldade inserindo nós com raios de transmissão infinitos, porém isto torna a simulação muito lenta desnecessariamente; esta foi a única dificuldade que não foi vencida na elaboração deste trabalho.

### 4.3.3 Cenários de Simulação

Finalizado o estudo da representação do ambiente de simulações, foram definidos os seguintes cenários de simulação:

- **Para avaliação dos protocolos:** cenário padrão, distribuído com o próprio *ONE*, com as configurações básicas e modificações apenas no número de nós e no algoritmo de roteamento, apenas, e;
- **Para avaliação das tecnologias:** cenário da Rodovia Triângulo do Sol, entre 200 e 600 veículos, com proporção de 30% de caminhões com velocidade média reduzida, 15 minutos de duração (tempo máximo de relevância de uma mensagem), geração em um veículo (carro ou caminhão) aleatório, protocolo epidêmico de propagação sem limite de transmissão de nós ou tempo de vida da mensagem.

Um exemplo da execução do *ONE* para avaliar uma certa tecnologia na Rodovia Triângulo do Sol pode ser observado abaixo:

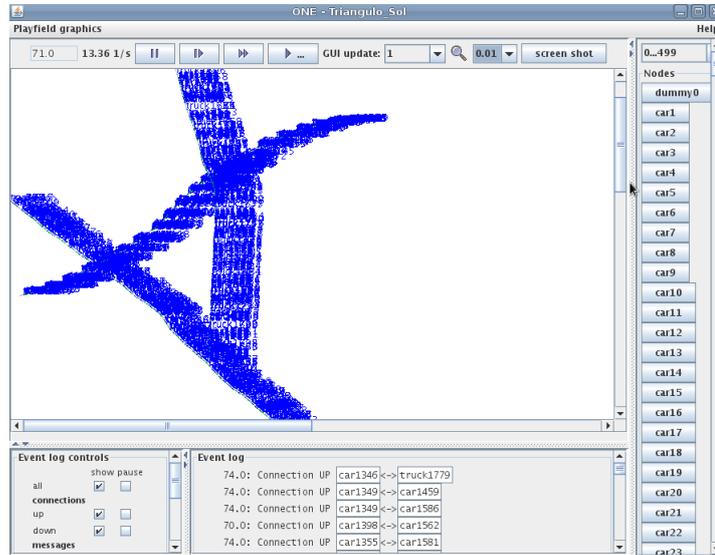


Figura 19: Simulação no cenário da Rodovia Triângulo do Sol

#### 4.3.4 Resultados

A primeira etapa de resultados compreendeu o estudo dos conceitos de Redes Tolerantes a Atrasos e a análise/comparação dos diversos protocolos de roteamento quando tratados pacotes com origem e destino especificados. Com todo o ferramental secundário, como *parsers* e *software* de plotagem de gráficos (*gnuplot*), e o simulador *ONE* foram coletados dados de diversas simulações.

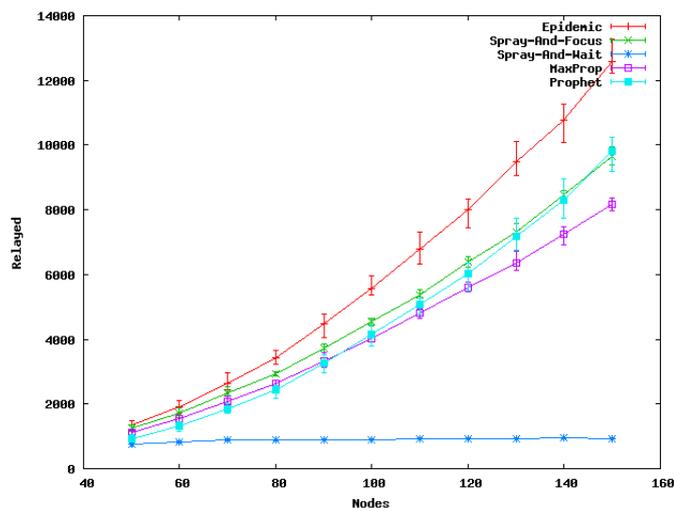


Figura 20: Gráfico representando o total de mensagens custodiadas em função do número de nós da rede

O gráfico exposto na figura 20 expressa o total de mensagens custodiadas (encaminhadas) em função do número de nós presentes na rede. É importante notar que, de fato, o protocolo *Epidemic* apresenta maiores quantidades de mensagens custodiadas em relação aos demais protocolos. Outra curva que se destaca no gráfico é a curva do protocolo *Spray-And-Wait*; devido à natureza do protocolo de dependência de encontros diretos com o destinatário para entrega de mensagem, a quantidade total de mensagens custodiadas fica restrita ao número de mensagens espalhadas na *spray phase*.

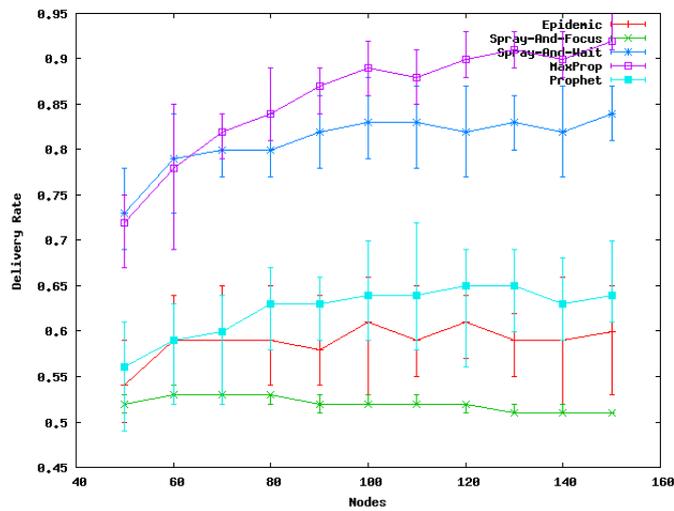


Figura 21: Gráfico representando a taxa de entrega em função do número de nós da rede

O gráfico exibido na figura 21 expressa a taxa total de mensagens entregues com sucesso (total de mensagens entregues dividido pelo total de mensagens criadas) em função do número de nós da rede. O protocolo que apresentou melhor desempenho foi o protocolo *MaxProp*, com taxas acima de 80% a partir de aproximadamente 70 nós presentes na rede. Outro protocolo que merece destaque é o protocolo *Spray-And-Wait*; nas condições da simulação o protocolo apresentou taxas de entrega superiores a protocolos mais complexos, como o *PRoPHET* e o *Spray-And-Focus*.

Já os protocolos *Epidemic* e *Spray-And-Focus* apresentaram taxas inferiores de entrega. O desempenho do protocolo *Epidemic* se deve principalmente pela limitação dos *buffers*, pois este assume um comportamento confiável apenas quando não há limitação dos mesmos. O desempenho do protocolo *Spray-And-Focus* é, de longe, o que causa maior estranheza: apesar de ser um protocolo mais refinado que o protocolo *Spray-And-Wait*, apresentou taxas de entrega inferiores nas condições da simulação.

Dos protocolos estudados, o *Epidemic* se mostrou mais adequado ao contexto de co-

municação interveicular, pois se deseja o maior espalhamento possível de uma mesma mensagem numa rede interveicular, isto é, que a maior quantidade possível de veículos seja notificada de um determinado evento em um menor espaço de tempo. Dentre outros resultados, obtivemos o seguinte gráfico ao aplicarmos o protocolo epidêmico às diferentes tecnologias de transmissão de dados:

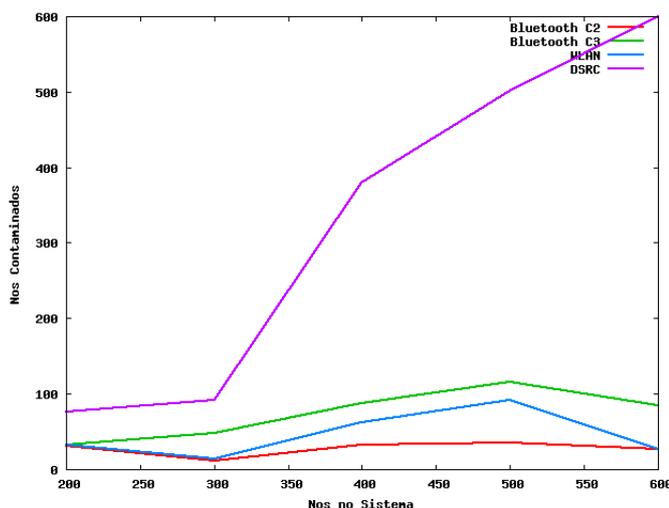


Figura 22: Gráfico representando o alcance de *broadcast* de um evento único gerado em um nó aleatório

O gráfico exibido na figura 22 representa o alcance de *broadcast* de um evento único em função da quantidade de nós presentes na rede. Em outras palavras, expressa quantos nós foram notificados de um evento gerado por algum nó aleatório no tempo de simulação.

Deste resultado, obtivemos a confirmação de que a tecnologia envolvida na implementação de uma rede de comunicação interveicular é de suma importância para a eficácia da mesma. Tecnologias como *Bluetooth* (Classe 2 e Classe 3) e *WLAN* apresentaram um alcance consideravelmente menor, no mesmo tempo de simulação, que o alcance obtido pela tecnologia *DSRC*. Esta, por exemplo, atinge um nível de aproximadamente 100% de alcance a partir da faixa de 400 nós presentes na rede, independentemente da posição do nó gerador do evento.

Também é importante ressaltar que as variações das curvas se devem principalmente à aleatoriedade dos eventos gerados, o que pode alterar significativamente o alcance de uma mensagem na rede. No entanto, é possível observar que as curvas raramente se cruzam e que então há consistência sobre a superioridade de uma tecnologia sobre outra.

## 5 Resumo das Atividades Realizadas

Em um ano de estudo, desenvolvimento, pesquisa e debates, muitas foram as atividades realizadas não só por nós, autores deste Trabalho de Conclusão de Curso, como por nosso grupo de Iniciação Científica. Dentre elas, de forma sucinta, vale citar a seguinte sequência:

- Estudo dos conceitos de Redes Tolerantes a Atrasos.
- Estudo dos possíveis contextos nos quais os fundamentos de Redes Tolerantes a Atrasos podem ser ou já estão sendo aplicados.
- Análise e comparação dos diferentes protocolos de roteamento aplicáveis em Redes Tolerantes a Atrasos.
- Participação no desenvolvimento e testes de novos algoritmos específicos para redes móveis com conhecimento prévio das conectividades futuras (Shortest, Foremost, Fastest).
- Estudo e aprendizado da utilização de um simulador adequado a redes móveis (ONE).
- Construção de cenários específicos e próximos da realidade para utilização no simulador ONE, com o auxílio da ferramenta OpenJUMP para representação de mapas rodoviários nacionais em formato WKT.
- Estudo das diversas tecnologias atuais com possibilidade de integrarem uma Rede Tolerante a Atrasos real.
- Análise das simulações obtidas com o cenário próximo da realidade construído e implementando as características dos dispositivos e tencologias atuais.

Do mais, nosso trabalho ainda está sendo continuado com o desenvolvimento de um artigo a ser submetido para o Simpósio Brasileiro de Redes de Computadores(SBRC), em dezembro de 2009. O tema deste artigo corresponde exatamente ao trabalho descrito nesta monografia.

## Referências

- [BGJL06] J. Burgess, B. Gallagher, D. Jensen, and B.N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *Proc. IEEE Infocom*, pages 1–11, 2006.
- [KOK09] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. The ONE Simulator for DTN Protocol Evaluation. In *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, New York, NY, USA, 2009. ICST.
- [LDS04] A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. *Lecture Notes in Computer Science*, pages 239–254, 2004.
- [OMR<sup>+</sup>07] C.T. Oliveira, M.D.D. Moreira, M.G. Rubinstein, L. Costa, and O. Duarte. Redes tolerantes a atrasos e desconexões. In *Minicursos do Simposio Brasileiro de Redes de Computadores (SBRC 2007)*, 2007.
- [ope] OpenJUMP - Open Java Unified Mapping Platform. <http://openjump.org>.
- [SPR05] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 252–259, New York, NY, USA, 2005. ACM.
- [SPR07] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. IEEE, 2007.
- [VB00] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical report, Citeseer, 2000.

## A Parte Subjetiva - Caio Cestari Silva

Todo trabalho desenvolvido possui uma base que sustentou sua elaboração. Essa base, que vem desde o início dos estudos, com uma motivação, até o aprofundamento em uma área específica passa por dificuldades e progressos, oscilando entre esses dois extremos à medida em que novas descobertas e discussões são realizadas. Nesta seção, serão descritos todos esses pontos que, com o trabalho, resultaram nesta monografia e nas conclusões nela apresentadas.

### A.1 Motivação

Com um convite informal em uma saída para o almoço, iniciamos nossos estudos sobre as Redes Tolerantes a Atrasos e Desconexões. Inicialmente, não sabíamos do que se tratava, então o início da Iniciação Científica baseou-se em leituras de vários artigos, pesquisas e apresentações sobre temas relacionados, aplicações e novidades na área, dentre outros assuntos.

Nosso orientador nos deixou bem à vontade com o tema para, então, propor três diferentes áreas de aprofundamento: algoritmos de jornadas, aplicações na comunicação interveicular e compromisso entre jornadas mínimas. Dos três temas, escolhemos o segundo, por seu caráter prático e atual, sobre o qual poderíamos nos aprofundar com maior naturalidade, utilizando teoria já estudada e desenvolvendo novas adaptações para o ambiente das *DTNs*.

Com o desenvolvimento de um artigo apresentado na Lapônia, em agosto, ficamos ainda mais motivados a pesquisar e, somando a tudo isso o auxílio do IPT, com a bolsa de Iniciação Científica da FIPT, tivemos o ambiente ideal para operarmos com dados reais, fornecidos pelo próprio Instituto.

### A.2 Disciplinas Relevantes

Toda a base para compreender a teoria das *DTNs* foi obtida da disciplina Algoritmos em Grafos, ministrada pelo professor Coelho. Dela, foram aplicados os conceitos de grafos, digrafos, nós, arestas, caminhos mínimos e fluxos ao simulador que, para ser compreendido, precisou da experiência adquirida ao cursarmos a disciplina de Estruturas de Dados e Laboratório de Programação Extrema.

Algumas noções de redes, denominações relacionadas aos roteamentos, fluxos, transmissão de dados, pacotes, mensagens de recebimento (*ACKs*), dentre outras foram obtidas

com o curso de Programação para Redes de Computadores, disciplina optativa lecionada pelo professor Marcílio. No auxílio ao desenvolvimento dos algoritmos na Iniciação Científica, utilizamos vários conceitos de Análise de Algoritmos para otimizar o desempenho dos mesmos. Para os algoritmos probabilísticos e análises de abrangência de algoritmos epidêmicos, muito de Noções de Probabilidade e Estatística foi aplicado e revisado, disciplina introdutória lecionada pelos professores Adilson e Elisabeti.

Para os desenvolvimentos realizados no *background*, ou seja, suportes aos resultados (*parsers*, escaladores de mapas, leitores e escritores de arquivos), utilizamos os conceitos aprendidos em Laboratório de Programação, com o professor Gubi. Foi claro observar que desde pequenos ensinamentos como *scripts* em *Perl* até grandes teoremas e algoritmos, como Dijkstra e Bellman-Ford, tudo se tornou útil no desenvolvimento do trabalho final.

### A.3 Facilidades e Dificuldades

Felizmente, há bastante material sendo publicado sobre *DTNs* atualmente; conseguimos ter uma base teórica bem sólida para começarmos a desenvolver nosso trabalho. Artigos sobre roteamentos, novos algoritmos, análises estatísticas, aplicações aos mais variados ambientes e resultados dessas aplicações consistiram em nossa principal fonte de pesquisa de desenvolvimento. Um estudo realizado pelo aluno Julian, do mestrado, consistiu na motivação teórica dos Grafos Evolutivos e seus resultados no desenvolvimento dos algoritmos de roteamento minimais foram continuados por nosso grupo.

A dificuldade apareceu quando exploramos a comunicação interveicular como exemplo de aplicação das *DTNs*. É difícil realizar uma simulação com carros, sensores e situações reais, pois é necessária uma verba considerável para a aquisição e/ou locação dos equipamentos e da infra-estrutura necessária para obter os resultados. Todos os dados obtidos, alguns até com certo sigilo, foram fornecidos pelo IPT e obtidos por intermédio de projetos realizados pelo Instituto com as principais rodovias concessionadas do Brasil, o que confere às simulações um caráter próximo do real, ainda que com algumas aproximações e normalizações de fluxos.

Outro problema foi enfrentado com o modelo de movimentação assumido para os veículos e até mesmo para as torres de comunicação das rodovias - não era claro qual modelo era mais adequado para representar a situação de forma mais fidedigna. Algumas tentativas e adaptações foram implementadas e conseguiu-se contornar o problema delimitando com certa clareza a rota dos veículos pelas rodovias, obedecendo os fluxos obtidos pelo IPT dos pedágios.

## A.4 Projetos Futuros

O artigo submetido para o *Workshop ExtremeCom*, na Lapônia, em agosto de 2009, foi de grande motivação para a escrita de novos artigos abrangendo as outras áreas de pesquisa e seus resultados. Desenvolvemos mais resultados desde o *Workshop* e estamos escrevendo um artigo para o Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), que será submetido até o final do ano.

Futuramente, pretendo também retornar ao IME-USP como pós-graduando e continuar as pesquisas na área de Sistemas, até mesmo com o tema específico desta monografia, dando ênfase ao estudo e pesquisa em outros tópicos de Redes Tolerantes a Atrasos e Desconexões.

## B Parte Subjetiva - Adriano Tabarelli

É indiscutível a importância do aprendizado adquirido durante os quatro anos do curso de Ciência da Computação. O conhecimento necessário para a realização deste trabalho de conclusão de curso abrange desde os conceitos básicos até o aprofundamento em tópicos específicos da área de computação. Dividirei nos próximos tópicos os pontos mais relevantes que possibilitaram a elaboração deste trabalho.

### B.1 Motivação

O início de nosso grupo de pesquisa na área de Redes Tolerantes a Atrasos deveu-se à nossa colega de curso Cássia Garcia Ferreira, que inicialmente apresentou seu interesse em ter o professor Alfredo Goldman como orientador do seu futuro trabalho de conclusão de curso. O professor Alfredo, posteriormente, manifestou interesse em criar um grupo para o estudo destas redes e então os convites para a inclusão no grupo foram sendo feitos. Em pouco menos de um mês, estaria formado o grupo composto pelos alunos Caio C. Silva, César G. Machado, Cássia G. Ferreira, Paulo H. Floriano e eu.

A atualidade da área de Redes Tolerantes a Atrasos e a grande quantidade de tópicos a serem estudados e desenvolvidos foram fatores decisivos na minha opção de entrar neste grupo de pesquisa. Dentre as possíveis derivações do tema de Redes Tolerantes a Atrasos, inicialmente foram propostas três: o estudo das redes intermitentes aplicadas à comunicação interveicular, algoritmos de jornadas e o compromisso entre estas jornadas. A primeira derivação mostrou-se mais atraente para o aprofundamento de nossa pesquisa, devido ao seu caráter atual e certamente mais prático.

### B.2 Disciplinas Relevantes

Ao longo do curso houve naturalmente um acúmulo de conhecimento e, posteriormente, tornou-se difícil separar quais disciplinas foram mais relevantes para o desenvolvimento de nosso trabalho. Diversas disciplinas se utilizam de conhecimento prévio adquirido em outras disciplinas, ou seja, muitas possuem uma base. Como bases essenciais, é possível citar as seguintes disciplinas:

- MAC0110 - Introdução à Computação
- MAC0122 - Princípios de Desenvolvimento de Algoritmos
- MAC0323 - Estruturas de Dados

Tais disciplinas foram as primeiras a apresentar os conceitos básicos de programação e estruturas de dados, o que foi de extrema importância para quase a totalidade das disciplinas posteriores.

Como disciplinas mais próximas dos conceitos que estudamos e aplicamos em nosso trabalho, é possível citar as seguintes:

- MAC0211 - Laboratório de Programação I
- MAC0328 - Algoritmos em Grafos
- MAC0242 - Laboratório de Programação II
- MAE0121/MAE0212 - Introdução à Probabilidade e Estatística I e II

A disciplina de Laboratório de Programação I foi a primeira a exigir um aprendizado sobre modularização de projetos e documentação dos mesmos. Como ponto principal, destaco o aprendizado de  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , ferramenta essencial para documentação de projetos e escrita de textos científicos e profissionais.

Já a disciplina de Algoritmos em Grafos ministrada pelo professor Coelho nos possibilitou um excelente conhecimento da estrutura de grafos e os algoritmos aplicados a eles. Desde a modelagem das redes intermitentes até a aplicação dos diversos protocolos de roteamento, os conceitos adquiridos em MAC0328 foram utilizados.

Em Laboratório de Programação II aprendemos sobre técnicas de desenvolvimento de projetos em *Java* e a utilização de linguagens de *script*, em especial o *Perl*. Para as ferramentas secundárias desenvolvidas, como *parsers* e escaladores de mapas, o conhecimento em *Perl* foi de grande importância.

Os conhecimentos adquirido em Introdução à Probabilidade e Estatística I e II foram importantes para a análise dos protocolos de roteamento, em especial os probabilísticos. Sem uma base de conhecimento de probabilidades, o entendimento destes protocolos não seria possível e o trabalho estaria restrito ao estudo de protocolos menos complexos.

### **B.3 Facilidades e Dificuldades**

O tema de Redes Tolerantes a Atrasos é um tema relativamente recente, entretanto, já existem diversos estudos publicados e em desenvolvimento. Ao longo de nossa pesquisa diversos artigos foram selecionados para leitura e usados como base para o desenvolvimento de nosso trabalho, ou seja, a quantidade de material teórico disponível não foi um fator negativo.

As maiores dificuldades se concentraram ao estudar a comunicação interveicular. Em diversos artigos, em especial os relacionados aos protocolos de roteamento, simulações em ambientes reais e com equipamentos reais foram realizadas. Encontramos grande dificuldades na tentativa de realizar experimentos reais, dado o enorme custo que a implementação de uma infra-estrutura real envolve. Logo, nosso estudo permaneceu restrito ao uso de simuladores em computador, como o ONE.

O uso apenas de simuladores não necessariamente facilitou nosso trabalho. A escolha de modelos de movimentação para cada elemento da simulação e a adaptação dos fluxos reais no simulador demonstraram ser nada triviais. Foram necessárias algumas adaptações e novas implementações para contornar tais problemas.

## **B.4 Projetos Futuros**

Ainda relacionado a este trabalho, estamos redigindo um artigo a ser submetido para o Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC). A motivação para esta submissão começou com nossa participação no desenvolvimento do artigo submetido para o *Workshop ExtremeCom*, realizado em agosto de 2009 na Lapônia, e o *feedback* positivo obtido deste trabalho.

Após a conclusão do meu bacharelado, tenho interesse em continuar os estudos e ingressar na pós-graduação do IME-USP na área de redes móveis. Quero aproveitar todo o conhecimento adquirido com nosso excelente grupo de pesquisa para a continuação dos estudos na área de Redes Tolerantes a Atrasos.