

Fabio Rafael Segundo

**ROTEAMENTO EM REDES TOLERANTES A ATRASOS
E INTERRUPÇÕES: UMA ABORDAGEM BASEADA EM
REDES NEURAIIS**

Tese submetida ao Programa de Pós-graduação em Engenharia de Automação e Sistemas para a obtenção do Grau de Doutor em Engenharia de Automação e Sistemas.

Orientador

Universidade Federal de Santa Catarina: Prof. Dr. Jean-Marie Farines

Coorientador

Instituto Federal de Santa Catarina:

Prof. Dr. Eraldo Silveira e Silva

Florianópolis

2015

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Segundo, Fabio Rafael

Roteamento em redes tolerantes a atrasos e interrupções
: Uma abordagem baseada em redes neurais / Fabio Rafael
Segundo ; orientador, Jean-Marie Farines ; coorientador,
Eraldo Silveira e Silva. - Florianópolis, SC, 2015.
216 p.

Tese (doutorado) - Universidade Federal de Santa
Catarina, Centro Tecnológico. Programa de Pós-Graduação
em Engenharia de Automação e Sistemas.

Inclui referências

1. Engenharia de Automação e Sistemas. 2. Redes
Tolerantes a Atrasos e Interrupções. 3. Roteamento. 4.
Redes Neurais Artificiais. I. Farines, Jean-Marie. II.
Silveira e Silva, Eraldo. III. Universidade Federal de
Santa Catarina. Programa de Pós-Graduação em Engenharia de
Automação e Sistemas. IV. Título.

Fabio Rafael Segundo

**ROTEAMENTO EM REDES TOLERANTES A ATRASOS
E INTERRUPÇÕES: UMA ABORDAGEM BASEADA EM
REDES NEURAIIS**

Esta Tese foi julgada aprovada para a obtenção do Título de “Doutor em Engenharia de Automação e Sistemas”, e aprovada em sua forma final pelo Programa de Pós-graduação em Engenharia de Automação e Sistemas.

Florianópolis, 25 de Maio 2015.

Prof. Dr. Rômulo da Silva
Coordenador
Universidade Federal de Santa Catarina

Prof. Dr. Jean-Marie Farines
Orientador
Universidade Federal de Santa Catarina

Prof. Dr. Eraldo Silveira e Silva
Coorientador
Instituto Federal de Santa Catarina

Banca Examinadora:

Prof. Dr. Jean-Marie Farines
Universidade Federal de Santa Catarina
Presidente

Prof. Dr. José Marcos Silva Nogueira
Universidade Federal de Minas Gerais

Prof. Dr. Igor Moraes
Universidade Federal Fluminense

Prof. Dr. Marcelo Maia Sobral
Instituto Federal de Santa Catarina

Prof. Dr. Roberto Willrich
Universidade Federal de Santa Catarina

Prof. Dr. Jomi Fred Hübner
Universidade Federal de Santa Catarina

Este trabalho é dedicado aos meus filhos,
esposa, pais e avós (*in memoriam*) queri-
dos.

AGRADECIMENTOS

Ao longo dos anos de trabalho que resultaram nesta tese, pessoas e instituições me ajudaram, ensinando e apoiando. Agora que alcanço meus objetivos não poderia deixar de reconhecê-las.

Começo por agradecer meus pais, como não poderia ser diferente. Sem seu esforço para minha formação pessoal e profissional e por incentivar a minha curiosidade, possivelmente não teria escolhido ser pesquisador. Acredito que os brinquedos eletrônicos abertos e dissecados na infância tem muito a ver com isso.

Agradeço aos meus irmãos, o mais velho por ter me orientado com sua experiência e pelo exemplo na docência e ao mais novo pela sua persistência e superação.

Minha família foi sempre uma fonte de alegria e descanso do trabalho. Agradeço especialmente minha esposa por me incentivar e dar o suporte suficiente para que pudesse me dedicar aos estudos e experimentos desse trabalho.

Aos meus colegas de disciplina e da sala dos doutorandos agradeço pelos momentos de discussão, aprendizado e também de descontração.

Agradeço aos meus orientadores Jean-Marie Farines e Eraldo Silveira e Silva por me guiar durante essa trajetória de pesquisa e escrita em que tive a oportunidade de aprender muito.

Meu reconhecimento aos professores, colegas e funcionários do Departamento de Automação e Sistemas (DAS) da UFSC, por tudo que me ensinaram em seus cursos, seminários e conversas, pelas oportunidades que me deram de participar de eventos nacionais e internacionais, e pela estrutura que vocês mantêm nesta instituição, propícia a formação de profissionais de nível internacional.

Agradeço à agência financiadora deste projeto CAPES (bolsa de doutorado REUNI). Agradeço também aos auxílios financeiros concedidos pelo DAS e PROPG para custear a apresentação de trabalhos em congressos internacionais.

Sobretudo, agradeço à Deus.

Se queres prever o futuro, estuda o passado.

(Confúcio, 551 a.C. - 479 a.C.)

RESUMO

As Redes Tolerantes a Atrasos e Interrupções (DTN) foram concebidas para operar considerando interrupções e grandes atrasos na comunicação. O roteamento se torna uma tarefa mais desafiadora em contextos com alta frequência de mudança da topologia e poucas informações a respeito da topologia futura. Em uma rede formada somente por ônibus do sistema de transporte público, os contatos entre os ônibus acontecem de forma quase-oportunista devido à regularidade não estritamente seguida nos itinerários. Uma forma de melhorar o roteamento nas DTNs é explorar informações históricas e do cenário para aumentar a taxa de entrega de mensagens e diminuir o atraso na entrega e o consumo de recursos. Com poucas informações a serem exploradas no cenário, o roteamento fica mais difícil de ser tratado. Neste contexto, esta tese propõe uma nova abordagem de roteamento baseado em Redes Neurais Artificiais (RNA) que apresenta vantagens em relação as outras estratégias aplicáveis nas mesmas condições, tal como a estratégia do caminho de probabilidade máxima (MaxProp). Um mecanismo de predição de contatos baseado em RNA foi desenvolvido para permitir a obtenção de contatos futuros que então são utilizados em um mecanismo de construção de jornadas, permitindo estimar a melhor jornada até o destino. Um procedimento para projetar as RNAs é apresentado. Um simulador de troca de mensagens foi desenvolvido para testar as estratégias avaliadas. Os resultados obtidos demonstram que a abordagem desenvolvida atinge um maior número de mensagens entregues, menor atraso e menor custo de uso da rede. Esses resultados foram obtidos nas versões com ou sem replicação de mensagens utilizando dados reais ou sintéticos. Uma modelagem para a implementação da estratégia de roteamento proposta projetada para funcionar na arquitetura da *Internet Research Task Force* (IRTF) é apresentada.

Palavras-chave: Redes Tolerantes a Atrasos e Interrupções. Roteamento. Redes Neurais Artificiais.

ABSTRACT

Delay/disruption Tolerant Networks (DTN) are designed to operate considering interruptions and high delays in communication. The routing becomes a more challenging task in a context of a high frequency of topology changes in which little information are available. In a network formed just by buses of a bus transportation system, the contacts are quasi-opportunistic due the regularity in the itineraries not strictly followed by the bus. A manner to improve the routing in DTN is to exploit historical information to increase the delivery rate and decrease the delivery delay and resources consumption. In this context, this thesis describes a new routing approach based on Artificial Neural Networks (ANN) presenting advantages over other applicable strategies in the same conditions, such as the maximum probability path (Max-Prop). A contact predictor based on ANN was developed to achieve future contacts. The predicted contacts are used in a journey predictor, aiming to obtain the best journey to the destination. A procedure for designing of ANNs is presented. A message forwarding simulator was developed to test the evaluated strategies. The obtained results demonstrate that the developed approach increases the number of delivered messages, decreases the delivery delay and decreases the delivery cost. These results are verified both in the version with message replication, as without message replication, using synthetic or real data. A modeling for the implementation of the proposed routing strategy designed to work in the Internet Research Task Force (IRTF) architecture is presented.

Keywords: Delay/disruption Tolerant Network. Routing. Artificial Neural Networks.

LISTA DE FIGURAS

Figura 1	Encaminhamento de mensagem na estratégia SCF.....	37
Figura 2	Exemplo de movimentações e contatos diários entre ônibus do STPU com encaminhamento de uma mensagem.....	48
Figura 3	Esquema para solução do problema de roteamento em DTN.....	64
Figura 4	Interação entre as fases do esquema de operação.....	67
Figura 5	Fase de tratamento do contato.....	68
Figura 6	Fase de atualização do conhecimento.....	69
Figura 7	Etapas do procedimento utilizado para definição das RNAs do preditor de contatos proposto.....	89
Figura 8	Exemplos de grafo de contato entre nós do DieselNet... ..	98
Figura 9	Tela do Weka com matriz de pontos geral.....	99
Figura 10	Matriz de pontos: contatos de 5 rotas com a Rota 31... ..	101
Figura 11	Gráfico de contatos da rota 30 em dias consecutivos... ..	102
Figura 12	Gráfico de contatos da rota 30 nas segundas-feiras.....	102
Figura 13	Topologia da RNA Global como proposta da função de predição de contato.....	105
Figura 14	Topologia das RNAs locais proposta com função de predição de contato.....	107
Figura 15	Gráfico de treinamento de uma RNA.....	110
Figura 16	Mensagens entregues sem cópia de mensagem.....	127
Figura 17	Atraso na entrega das mensagens em faixas de 10 minutos.....	129
Figura 18	Quantidade de decisões de encaminhamento.....	132
Figura 19	Número de mensagens entregues com variação do número de cópias da mensagem.....	134
Figura 20	Número de mensagens entregues por faixas de atraso na entrega.....	135
Figura 21	Ocupação da rede com variação do número de cópias da mensagem.....	136
Figura 22	Custo para cada mensagem entregue.....	137
Figura 23	Número de mensagens entregues variando o tamanhos variados da fila de mensagem.....	139

Figura 24 Número de mensagens entregues com duração real dos contatos.....	141
Figura 25 Comunicação na Arquitetura DTN proposta pelo DTNRG (WARTHMAN et al., 2013).....	153
Figura 26 Módulos da arquitetura DTN do DTNRG. Adaptado de (DEMMEER et al., 2004).	155
Figura 27 O módulo Roteador de Agregados para a estratégia RNA proposta.....	158
Figura 28 Diagrama de classes da arquitetura de roteamento proposta.....	161
Figura 29 Modelo de um neurônio artificial (MENDES et al., 2012).	186
Figura 30 Organização dos neurônios em uma rede neural multicamadas (KUMAR; VENKATARAM, 2002).....	187
Figura 31 Rotas de ônibus na área de serviço do UMass Transit Service (UMASSAMHERST, 2014).....	198
Figura 32 Tela do simulador de DTN ONE: Rotas e pontos de parada do cenário estão sublinhados.	202
Figura 33 Gráfico de contatos para o nó 0 em 10 casos.....	204

LISTA DE TABELAS

Tabela 1	Comparação dos cenários utilizados nas propostas de roteamento DTN no contexto do STPU.....	54
Tabela 2	Comparação dos métodos utilizados nas propostas de roteamento DTN no contexto do STPU.....	55
Tabela 3	Taxa de acerto (%) das funções de predição em cada dia.	112
Tabela 4	Número de mensagens entregues (RNA global).....	121
Tabela 5	Número de mensagens entregues (RNAs locais).....	122
Tabela 6	Tempo de execução (ms) das estratégias de roteamento (dados sintéticos).....	123
Tabela 7	Número de mensagens entregues em vários dias (contatos reais).....	126
Tabela 8	Tempo de execução (ms) (contatos reais).....	130
Tabela 9	Faixas de velocidade e tempo de permanência nos pontos de parada para 10 casos.....	203
Tabela 10	Velocidade e tempo de parada para 20 variações do cenário.....	205

LISTA DE ABREVIATURAS E SIGLAS

MANET	<i>Mobile Ad hoc Networks</i>	27
DTN	<i>Delay/disruption Tolerant Network</i>	27
SCF	<i>Store-Carry-and-Forward</i>	27
STPU	Sistema de Transporte Público Urbano	27
MAM	Método de Aprendizagem de Máquina	27
ANN	<i>Artificial Neural Network</i>	28
RNA	Rede Neural Artificial	28
DTNRG	<i>DTN Research Group</i>	28
IRTF	<i>Internet Research Task Force</i>	28
IETF	<i>Internet Engineering Task Force</i>	28
IPN	<i>InterPlanetary Networks</i>	35
VANET	<i>Vehicular Ad-hoc Networks</i>	35
PSN	<i>Pocket-Switched Network</i>	35
MAM	Método de Aprendizagem de Máquina	43
FF	<i>FeedForward</i>	58
MLP	<i>Multi-Layer Perceptron</i>	58
BSW	<i>Binary Spray and Wait</i>	80
TCA	<i>Tetherless Communication Architecture</i>	152
BL	<i>Bundle Layer</i>	153
BP	<i>Bundle Protocol</i>	153
IPC	<i>Inter-Process Communication</i>	156
RBF	<i>Radial Basis Function - RBF</i>	190

SUMÁRIO

1 INTRODUÇÃO	27
1.1 CONTEXTO DO TRABALHO	28
1.2 MOTIVAÇÕES	29
1.3 OBJETIVOS	30
1.4 ORGANIZAÇÃO DO TEXTO	31
2 ROTEAMENTO EM DTN	33
2.1 INTRODUÇÃO	33
2.2 ROTEAMENTO: DEFINIÇÕES E CONCEITOS	33
2.3 AS REDES DTN: CONCEITOS BÁSICOS	35
2.3.1 Processo SCF de Encaminhamento de Mensagens ..	36
2.3.2 Caracterização e classificação dos Contatos	38
2.3.3 Gerenciamento da Fila de Mensagens	40
2.4 ESTRATÉGIAS DE ROTEAMENTO	41
2.4.1 Abordagens básicas de roteamento DTN	41
2.4.2 O Roteamento no Contexto do Sistema de Transporte Público Urbano	43
2.5 O PROBLEMA DE ROTEAMENTO EM DTN	45
2.5.1 O problema de roteamento com contatos quase-oportunistas	47
2.6 PROPOSTAS PARA O ROTEAMENTO DTN	49
2.6.1 Propostas de Roteamento no Contexto do STPU ...	50
2.6.2 Comparação das propostas	54
2.6.3 As RNAs nas Propostas de Roteamento	57
2.6.4 Relações com o problema a ser tratado	59
2.7 CONCLUSÃO	60
3 ROTEAMENTO BASEADO EM PREDIÇÃO DE JORNADAS	63
3.1 INTRODUÇÃO	63
3.2 UMA PROPOSTA DE ROTEAMENTO BASEADO EM JORNADAS PREDITAS	64
3.2.1 Princípios do preditor de jornadas	65
3.2.2 Utilização do preditor de jornadas	66
3.3 MODELAGEM DO PREDITOR DE JORNADAS	70
3.3.1 Modelo de rede DTN	70
3.3.2 Modelo de jornadas previstas	71
3.3.3 Determinação de melhor jornada mediante um critério predefinido	73

3.3.4 Um exemplo de predição de jornadas	73
3.4 PROPOSTA DE UM PREDITOR DE JORNADAS	74
3.4.1 Invocação do preditor de jornadas.....	74
3.4.2 O algoritmo do construtor de jornadas	76
3.4.3 Critério para jornadas de menor custo	78
3.4.4 O ajuste da fatia de tempo na construção de jornadas: <i>timeSlice</i>	79
3.5 ESTRATÉGIA DE ROTEAMENTO COM REPLICAÇÃO DE MENSAGENS	80
3.6 CONCLUSÃO	82
4 UM PREDITOR DE CONTATOS BASEADO EM RNA	83
4.1 INTRODUÇÃO	83
4.2 FUNÇÃO DE PREDIÇÃO DE CONTATOS	83
4.2.1 Definição da Função de Predição de Contatos	84
4.2.2 Operação da Função de Predição de Contatos baseada em RNA	86
4.3 UM PROCEDIMENTO PARA OBTENÇÃO DE UM PREDITOR BASEADO EM RNA	88
4.3.1 Etapa de análise de dados	88
4.3.2 Etapa de definição da RNA	91
4.3.3 Etapa de testes e avaliação do preditor	94
4.4 APLICAÇÃO DO PROCEDIMENTO PARA CONSTRUÇÃO DE UM PREDITOR DE CONTATOS	95
4.4.1 Etapa de análise de dados	95
4.4.2 Etapa de definição das RNAs	103
4.4.3 Etapa de testes de avaliação do preditor de contatos	111
4.5 CONCLUSÃO	113
5 AVALIAÇÃO DO ROTEAMENTO DTN BASEADO EM RNA	115
5.1 MÉTRICAS E AVALIAÇÕES	115
5.2 CENÁRIOS, EXPERIMENTO E IMPLEMENTAÇÕES PARA A AVALIAÇÃO	117
5.2.1 Os cenários	117
5.2.2 As implementações e parametrizações dos algoritmos	119
5.3 AVALIAÇÃO DA ESTRATÉGIA DE ROTEAMENTO PROPOSTA USANDO CONTATOS SINTÉTICOS	120
5.3.1 Cenário para as simulações	120
5.3.2 Análise da entrega	120
5.3.3 Análise do tempo de execução	123
5.3.4 Conclusões	124

5.4	AVALIAÇÃO DA ESTRATÉGIA DE ROTEAMENTO PRO- POSTA USANDO CONTATOS REAIS	125
5.4.1	Cenário para as simulações	125
5.4.2	Avaliação da entrega de mensagens	126
5.4.3	Avaliação do atraso na entrega das mensagens	128
5.4.4	Avaliação do tempo de execução	129
5.4.5	Análise da influência das decisões de encaminha- mento	131
5.4.6	Conclusões	132
5.5	ANÁLISE DA INFLUÊNCIA DA REPLICAÇÃO	133
5.5.1	Avaliação da entrega de mensagens	133
5.5.2	Avaliação do atraso na entrega das mensagens	134
5.5.3	Avaliação do número de mensagens geradas	135
5.5.4	Avaliação do custo de entrega das mensagens	136
5.5.5	Conclusões	137
5.6	ANÁLISE DA INTRODUÇÃO DO CONTROLE DA FILA DE MENSAGENS E DURAÇÃO DA CONEXÃO	138
5.6.1	Avaliação da entrega de mensagens com tamanhos diferentes das filas de mensagens	139
5.6.2	Avaliação da entrega de mensagens com variação da duração dos contatos	140
5.6.3	Conclusões	140
5.7	ESTUDO DE AJUSTES NA RNA DO PREDITOR DE CONTATOS	142
5.7.1	Filtragem das informações para treinamento	142
5.7.2	Separação da RNA por informação de saída	144
5.7.3	Predição de jornadas com RNA global ou local	144
5.7.4	Relação com o nó de contato anterior	146
5.7.5	Treinamento e Codificação da Entrada	147
5.7.6	Fatia de tempo na construção de jornadas: <i>timeSlice</i>	147
5.7.7	Conclusão sobre os ajustes	148
5.8	CONCLUSÕES	149
6	EM DIREÇÃO A IMPLEMENTAÇÃO DA PROPOSTA DE ROTEAMENTO	151
6.1	INTRODUÇÃO	151
6.2	ARQUITETURA DTN DO IRTF	152
6.2.1	Arquitetura de Roteamento	154
6.3	PROPOSTA DE ARQUITETURA DE SUPORTE AO RO- TEAMENTO	157
6.4	MODELO PARA IMPLEMENTAÇÃO DA ARQUITETURA DE ROTEAMENTO PROPOSTA	160

6.4.1 Modelo de classes	160
6.4.2 Protocolo de comunicação	161
6.4.3 Discussão sobre a implementação	162
6.5 CONCLUSÃO	165
7 CONCLUSÃO	167
7.1 PREDITOR DE JORNADAS	169
7.2 PREDITOR DE CONTATOS	169
7.2.1 Estabelecimento de contatos nos cenários estudados	169
7.2.2 Procedimento em etapas para a definição das RNAs	170
7.2.3 Filtragem de dados	171
7.2.4 Perspectivas de melhorias no projeto das RNAs....	171
7.3 MÉTRICAS AVALIADAS NAS PESQUISAS	172
7.4 ARQUITETURA DE ROTEAMENTO	172
7.5 PUBLICAÇÕES PROVENIENTES DA PESQUISA	173
REFERÊNCIAS	175
APÊNDICE A – Redes Neurais Artificiais	185
APÊNDICE B – Dados de Contatos no STPU Utilizados	197
APÊNDICE C – Simulador de comunicação em DTN...	211

1 INTRODUÇÃO

O avanço das tecnologias das redes sem fio, aliado a redução dos dispositivos eletrônicos de comunicação, proporcionou uma grande difusão no uso de dispositivos móveis de comunicação. A maior parte dessas tecnologias se utiliza de infraestrutura fixa para facilitar a comunicação. É o caso das redes de telefonia celular e de redes com tecnologia WiFi baseadas em pontos de acesso.

Nas redes sem infraestrutura (redes *ad hoc*), tal como as redes *ad hoc* móveis (*Mobile Ad hoc NETWORKS* - MANET), foram desenvolvidos protocolos de roteamento específicos utilizando o reencaminhamento de mensagens através de outros nós (redes *ad hoc multihop*) (MISRA; WOUNGANG; MISRA, 2009).

Algumas aplicações assíncronas, tais como correio eletrônico e coleta de dados, não apresentam restrições rígidas de atraso e podem tolerar uma espera, eventualmente longa, pelo restabelecimento da comunicação. As redes que são chamadas de tolerante a atrasos e interrupções (*Delay/Disruption Tolerant Network* - DTN), aceitando inclusive a desconexão temporária de suas partes, devem permitir o funcionamento dessas aplicações. Nessas redes o roteamento torna-se um problema desafiador, pois deve-se manter a comunicação através de conexões intermitentes, normalmente causadas pela mobilidade dos nós (FALL, 2003; KHABBAZ; ASSI; FAWAZ, 2012).

As DTNs empregam um procedimento de armazenagem e transporte de mensagens (*Store-Carry-and-Forward* - SCF) para encaminhar a mensagem até o destino. No SCF a mensagem pode ficar armazenada em um nó até encontrar outro nó que tenha melhor oportunidade de entrega da mensagem ao destinatário da mensagem, sendo isso um elemento fundamental para o desempenho da rede.

O roteamento DTN é um processo dinâmico de tomada de decisão de repasse ou não da mensagem ao nó com o qual está em contato. Os contatos entre nós dessas redes são caracterizados de acordo com o conhecimento utilizado para determinar a sua ocorrência. As redes com contatos do tipo oportunista são mais difíceis de tratar e muitas pesquisas tentam trazer algum determinismo a esses contatos, normalmente apoiando-se na extração de informações do cenário. Um exemplo disso acontece no cenário no Sistema de Transporte Público Urbano (STPU).

Vários trabalhos realizam o tratamento dos contatos oportunistas a partir de Métodos de Aprendizagem de Máquina (MAM) tais como processos markovianos, redes bayesianas, lógica difusa e aprendi-

zagem por reforço com multi-agentes.

Neste trabalho é apresentada uma estratégia para o tratamento dos contatos oportunistas no cenário do STPU utilizando Redes Neurais Artificiais (RNA). Um modelo para implementação dessa abordagem é apresentado em conformidade com um padrão de referência.

1.1 CONTEXTO DO TRABALHO

No contexto dos ônibus do STPU muitos movimentos são repetidos diariamente. No STPU existem horários e locais estabelecidos para a parada dos ônibus, repetindo a execução das rotas normalmente várias vezes dentro do dia e a execução da linha de ônibus diariamente.

Os atrasos, adiantamentos e falhas nos nós fazem os contatos acontecerem de forma oportunista. Contudo, a regularidade dos horários e itinerários repetidos de forma próxima em dias diferentes, permite extrair informações históricas que podem ser usadas para diminuir o oportunismo nos contatos, tornando-os quase-oportunistas¹.

Por outro lado, no roteamento em DTN geralmente é empregada uma estratégia gulosa, tal como a inundação de mensagens, para aumentar as chances de uma comunicação alcançar seu destino, mas com mais consumo de recursos de rede e espaço para armazenamento de mensagens. Uma estratégia sem replicação minimiza o uso de recursos, mas é necessário haver muito conhecimento para prever a topologia futura da rede e aumentar as chances de entrega das mensagens.

Neste trabalho o cenário pesquisado é o do STPU que fornece informações do histórico de contatos. Pressupõe-se que a comunicação é omnidirecional com o raio de alcance limitado e o meio de acesso compartilhado. Assume-se ainda não haver inserção e remoção dinâmica de nós na rede.

Finalmente, a arquitetura de rede DTN definida pelo DTNRG (*DTN Research Group* - DTNRG)² é utilizada como referência para modelar a implementação da abordagem de roteamento proposta neste trabalho.

¹Entende-se por contatos quase-oportunistas os contatos oportunistas em contextos nos quais existem regularidades na movimentação, nos instantes de tempo, ou outra que ofereça alguma ajuda, mesmo que pequena, na determinação da ocorrência de contatos, mesmo que em um baixo grau.

²Grupo de pesquisa do IRTF afiliada a Força Tarefa de Engenharia da Internet (*Internet Engineering Task Force* - IETF)

1.2 MOTIVAÇÕES

A melhoria do roteamento nas DTNs com contatos oportunistas é importante para viabilizar o funcionamento de aplicações nessas redes. A ideia é que uma rede de comunicação formada por veículos (ônibus, táxis, trens, carros, entre outros) consiga obter desempenho suficiente para realizar a comunicação em locais onde não existe infraestrutura de comunicação, coletar e difundir informações de trânsito e da cidade, e diminuir a carga de dados de baixa prioridade de outras redes (p.e. celulares).

Enquanto uma estratégia gulosa consome muitos recursos, uma estratégia de entrega direta ou sem cópia de mensagens tende a oferecer baixas taxas de entrega de mensagens. Em ambos os casos, diminui-se a aplicabilidade da solução de roteamento diante das exigências de desempenho da aplicação e da coexistência de outras aplicações que geram demanda da rede. Para tratar esse problema foram desenvolvidas estratégias de roteamento DTN baseadas nas informações de nós da rede que possuam trajetórias conhecidas, cíclicas ou com alguma forma de agendamento. Contudo, apesar do determinismo apresentado, não existem as garantias suficientes de que o encaminhamento na maioria das vezes encontre o destino, por causa da expiração das mensagens e variações pequenas no movimento que podem mudar toda uma sequência de contatos.

Em cenários tais como o do STPU é possível encontrar um contexto de contatos oportunistas, no qual os elementos da rede (no caso ônibus) realizam movimentos repetitivos, em que seja possível explorar conhecimento útil sobre o estabelecimento dos contatos para favorecer a comunicação. Exemplos de aplicações suportadas por este sistema de comunicação são a coleta de dados, a verificação do estado de funcionamento dos ônibus, a previsão de chegada, o correio eletrônico e outras que não exijam interatividade e podem admitir atrasos elevados ou falhas na comunicação.

O desenvolvimento de novas abordagens de roteamento com aprendizagem de máquina, utilizando informações históricas de contato, vem apresentando resultados satisfatórios, porém ainda existem poucas pesquisas de aplicação de alguns desses métodos em DTNs, tal como RNAs. O emprego de RNAs pode ajudar a encontrar padrões pouco visíveis e não lineares nos dados históricos.

Nesse contexto de contatos, uma abordagem de roteamento com RNAs pode trazer desempenho suficiente para diminuir ou eliminar a necessidade de infraestrutura fixa de rede e permitir a comunicação de

outros dispositivos que estão fora das suas áreas de cobertura. Essa abordagem também pode ser implementada em dispositivos muito simples e com restrição quanto ao consumo de energia. E, além disso, pode permitir a comunicação em ambientes hostis, tais como em áreas sujeitas a vandalismos, guerras, em ambientes de comunicação sub-aquática e dentro de minas ou cavernas, entre outros.

1.3 OBJETIVOS

O objetivo geral desse trabalho é desenvolver uma nova abordagem para o roteamento em redes DTN com contatos oportunistas, aproveitando-se das RNAs alimentadas com informações do histórico de contatos para obter jornadas que permitam melhorar o encaminhamento das mensagens para os seus destinos. Essa abordagem deve se adequar à arquitetura de roteamento projetada e desenvolvida pelo DTNRG.

Para isso, alguns objetivos específicos precisam ser atingidos:

- a determinação de um procedimento que facilita o tratamento das informações disponíveis para o desenvolvimento de arquiteturas para as RNAs;
- a proposta de preditores de contato baseados em RNAs que modelam relações entre informações de toda a rede (global) e entre informações separadamente para cada nó (local);
- a proposta de um preditor de jornadas baseado no preditor de contatos desenvolvido que minimize o atraso na entrega das mensagens;
- a proposta de versões da estratégia de roteamento com preditores sem replicação de mensagens e com replicação controlada de mensagens.
- a avaliação do atraso na entrega das mensagens, do número de mensagens entregues, do custo de entrega, do tempo de execução e da influência do tamanho das filas de mensagens e da duração dos contatos no roteamento proposta e a comparação com outras abordagens conhecidas;
- a modelagem da implementação de acordo com uma arquitetura de referência;

1.4 ORGANIZAÇÃO DO TEXTO

O texto dessa tese está organizada em seis capítulos e três apêndices. O Capítulo 1 introduz uma visão geral do problema a ser tratado, dos métodos já utilizados para tratar o problema e da proposta desse trabalho. É apresentado o contexto em que o trabalho está inserido, as motivações e seus objetivos.

O Capítulo 2 fornece a fundamentação necessária sobre o roteamento em DTNs e a apresentação do problema a ser tratado na tese. Além disso, são apresentados os trabalhos de referência de roteamento em DTN.

No Capítulo 3 é apresentada uma proposta de construção de jornadas preditas a partir de um preditor do próximo nó e momento de contato. Duas versões da estratégia de roteamento DTN no contexto do STPU são discutidas: uma sem cópia de mensagens e outra que utiliza replicação controlada.

A descrição da proposta de um preditor de contatos para ser utilizado no preditor de jornadas é apresentada no Capítulo 4. Neste capítulo é descrito o procedimento para análise e extração de informações para esse preditor e o desenvolvimento das arquiteturas das RNAs utilizadas.

O Capítulo 5 trata da avaliação da abordagem proposta. São apresentadas avaliações dos preditores de contato baseados em informações globais e locais. A avaliação das duas versões, com e sem replicação, para a estratégia proposta de roteamento é apresentada.

Uma modelagem para implementação do roteamento proposto em uma arquitetura de DTN de referência é mostrada no Capítulo 6.

Como conclusão, no Capítulo 7 são apresentadas considerações organizadas por temas a cerca da pesquisa, dos resultados obtidos e dos trabalhos futuros.

2 ROTEAMENTO EM DTN

Neste capítulo são fundamentadas as redes DTN e o roteamento nessas redes. São apresentados conceitos básicos, as estratégias de roteamento, o problema a ser tratado e trabalhos sobre roteamento em DTN de interesse para o trabalho dessa tese. No final são apresentadas propostas para tratamento do roteamento em DTN em condições mais próximas a da proposta dessa tese.

2.1 INTRODUÇÃO

As redes DTN são redes que devem funcionar mediante longos atrasos e interrupções na comunicação, trazendo desafios para o roteamento. A intermitência nas conexões acontece por causa de falhas na comunicação causadas pela movimentação dos nós para fora da área de cobertura de comunicação, interferências de outros nós ou objetos na comunicação e falhas dos próprios nós da rede. Na DTN é complicado estabelecer uma rota, pois na maior parte do tempo não existe um caminho estabelecido de fim-a-fim entre os nós comunicantes.

Para uma mensagem ser encaminhada até seu destinatário, consome-se banda de comunicação, processamento e espaço de armazenamento dos nós no caminho dessa mensagem. Em DTN, a escolha de uma direção para o encaminhamento de mensagens é normalmente uma tarefa complexa que depende do tipo de contato que ocorre na rede. Esta complexidade aumenta quando é necessário controlar o consumo de recursos. Para tanto, são desenvolvidas abordagens mais elaboradas e dinâmicas do que em redes com nós fixos.

Um problema de fundamental interesse neste trabalho é o roteamento e encaminhamento de mensagens em DTN com contatos do tipo oportunista e com controle do consumo de recursos.

2.2 ROTEAMENTO: DEFINIÇÕES E CONCEITOS

Nesta seção são apresentados conceitos de roteamento, encaminhamento, jornada e caminhos.

Roteamento e encaminhamento podem ser vistos como partes de um processo, pois podem ser tratados como funções complementares. Contudo convém esclarecer e organizar suas funções para o entendi-

mento da arquitetura de roteamento e de detalhes da proposta. A diferença entre caminho e jornada é também apresentada neste contexto.

Neste trabalho não se faz distinção entre pacotes e mensagens.

Roteamento e Encaminhamento

O roteamento (*routing*) é a função que define as rotas que as mensagens devem seguir para atingir seu destino e o encaminhamento (*forwarding*) operacionaliza o envio da mensagem através de uma ou várias interfaces locais de comunicação, possivelmente baseado em definições de rotas.

Um planejamento com passos definidos e organizados para abordar o tratamento do roteamento determina uma estratégia de roteamento.

O roteamento deve ser definido de acordo com identificadores que permitam direcionar o fluxo de mensagens, seja endereço de nó, rede e sub-rede, informação de posição física ou temporal. Esses identificadores organizados em uma tabela de roteamento definem rotas que fornecem instruções para o encaminhamento das mensagens na direção de seus destinatários. O roteamento é reativo quando a tabela de roteamento é definida ou atualizada quando realizada a consulta por uma rota, caso contrário é proativo. Um protocolo de roteamento é definido e utilizado para a troca de informação entre nós (KUROSE; ROSS, 2012).

No caso das redes móveis sem infraestrutura, o roteamento é estabelecido salto-a-salto (*hop-by-hop*) pelos nós. Em cada nó, uma tabela de roteamento simples pode levar em consideração apenas a informação dos vizinhos de cada nó como identificadores de direção. A estratégia de roteamento DTN no contexto de contatos oportunistas (aleatórios) normalmente é dinâmica e reativa.

Caminho e Jornada

O termo caminho indica uma sequência ordenada de nós que uma mensagem segue de um determinado nó de origem até outro nó de destino. A cada mudança de nó que a mensagem sofre no caminho, considera-se um salto (*hop*). Além da identificação dos nós envolvidos na transferência da mensagem, pode-se associar ao salto informações tais como o momento ou intervalo no tempo e a capacidade de transmissão do canal de comunicação no momento da transferência.

Enquanto em um caminho se pressupõe a existência e disponibi-

lidade de todos os seus trechos em sua extensão, em uma jornada isso não acontece. A jornada deve considerar que o caminho pode estar desconectado, sendo que os trechos podem mudar e podem não estar mais disponíveis enquanto a jornada está sendo percorrida (XUAN; FERREIRA; JARRY, 2003; CASTEIGTS et al., 2010). Isso acontece por que uma mensagem até um destino deve passar por nós em movimento com conexões intermitentes. Em DTNs com contatos oportunistas (como será visto adiante) o tempo de permanência da mensagem no nó acaba sendo longo e indeterminado.

Existem informações que ajudam a determinar em que condições o salto foi ou poderá ser realizado, tais como largura de banda, instante no tempo e duração. Essas informações tem um papel fundamental nas estratégias de encaminhamento que selecionam o próximo salto de uma jornada para determinar um melhor encaminhamento ao destino, impactando no desempenho geral da comunicação na DTN.

2.3 AS REDES DTN: CONCEITOS BÁSICOS

A concepção das DTNs teve início com a idéia da criação de uma rede de comunicação interplanetária (*InterPlanetary Networks* - IPN). Nessa rede, os sistemas de comunicação localizados em planetas, cometas, satélites, naves e estações orbitais poderiam não obter comunicação se perdessem visada com o sistema localizado em outro local. Dessa forma, inspirado do sistema de correio eletrônico, foi idealizado um sistema de comunicação com a estratégia de armazenar as mensagens no próprio sistema ou em outro que tivesse melhores condições de restabelecimento da comunicação para alcançar o destinatário (SOCIEVOLE et al., 2011).

Essa concepção foi levada para outros cenários que incluem: as redes para monitoramento de animais, por exemplo o ZebraNet (JUANG et al., 2002 apud KHABBAZ; ASSI; FAWAZ, 2012); as redes de satélites, no qual a comunicação entre o satélite e pontos na terra sofre com longos atrasos e conectividade esporádica; as redes de sensores com limitação de energia, em que a energia da bateria é limitada e os sensores tem agendamentos para acordar e dormir periodicamente para enviar dados; as redes *ad hoc* militares, em que os nós podem se mover aleatoriamente e estão sujeitos a serem destruídos; as VANETs, em que os veículos se movimentam pelas estradas sujeitos a atrasos, adiamentos e falhas na comunicação; as redes de dispositivos de bolso (*Pocket-Switched Network* - PSN), cujos dispositivos de comunicação

pessoal representam os nós da rede e podem ser movimentados por ruas, edificações, meios de transporte e regiões de difícil acesso.

As redes DTN diferenciam-se das tradicionais, principalmente por apresentar algum ou todos os seguintes pressupostos (ZHANG, 2006):

- conectividade intermitente, podendo ter longos períodos de desconexão;
- taxa de perda de pacotes muito alta;
- atraso de propagação ou variação de atraso bastante alto;
- taxas assimétricas nos sentidos da transmissão, podendo ter disponibilidade de apenas um sentido.

Por isso as estratégias de roteamento das redes tradicionais não são apropriados para funcionar nas DTNs, exigindo novas abordagens.

Nas DTNs, não é necessário existir um caminho disponível entre a origem e o destino no momento que se queira transmitir uma mensagem, dispensando a existência de um caminho fim-a-fim conectado.

Os principais componentes em uma DTN são as mensagens, os nós, a fila de mensagens, o processo de roteamento, o processo de encaminhamento e o processo gerenciador da fila de mensagens (JAIN; FALL; PATRA, 2004; KHABBAZ; ASSI; FAWAZ, 2012).

Um contato acontece quando o dispositivo de comunicação de um nó está no alcance de comunicação de outro nó e ambos os nós se identificam para realizar uma comunicação. O tempo entre o fim de um contato e o início de outro contato de um mesmo nó é definido como tempo de intercontato. Uma transferência de mensagem ocorre quando um nó contatado envia e outro recebe uma mensagem para ser levada adiante. Quando um nó encontra o destinatário, a transferência é chamada de entrega. Em cada nó, as mensagens devem ficar armazenadas na fila de mensagens enquanto não encontram uma oportunidade de encaminhamento.

2.3.1 Processo SCF de Encaminhamento de Mensagens

Para que a DTN funcione, o encaminhamento de mensagens com armazenamento persistente de tempo indeterminado, baseado no método Armazena-Transporta-e-Encaminha (SCF - *Store-Carry-and-Forward*) é normalmente empregado. O encaminhamento por SCF é ilustrado na Figura 1, onde os nós *A*, *B*, *C* e *D* são representados por

círculos, as setas indicam suas direções (que não necessariamente precisam ser retilíneas) e o quadrado com a letra D significa uma mensagem endereçada para o nó D . Nesta figura estão representados quatro instantes distintos.

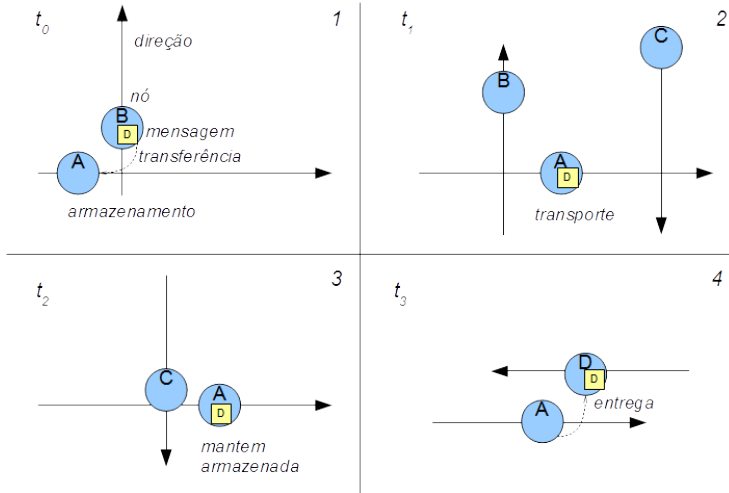


Figura 1: Encaminhamento de mensagem na estratégia SCF.

No SCF, quando um nó (B) possui uma mensagem para outro nó (D) e realiza um contato com outro nó (A), se for determinado que o nó em contato deve encaminhar esta mensagem, ela é transferida para este nó, tal como ilustrado no momento t_0 da Figura 1. A mensagem armazenada no *buffer* de mensagens do nó é transportada pelo nó (instante t_1) quando ele se movimentar. Pode acontecer de o nó em contato (C) não ser escolhido para encaminhar a mensagem para um determinado destino, tal como acontece no momento t_2 . Neste caso, a mensagem deve permanecer armazenada até que se tenha uma melhor oportunidade de encaminhamento ou se possa entregar a mensagem diretamente para seu destinatário. A mensagem pode ser transferida de nó em nó até a entrega para o nó de destino (instante t_3). O número de mensagens entregues aumenta quando a transferência das mensagens é feita para nós que tem melhores chances de entrega.

Para aumentar as chances de entrega da mensagem, ao invés da transferência da mensagem, pode-se utilizar a cópia ou replicação. Assim, o nó permanece com uma cópia da mensagem e pode tentar copiá-la novamente nos próximos contatos. Apesar do aumento da

entrega de mensagens, o consumo de banda de rede e espaço na fila de mensagem pode ser muito alto, limitando o funcionamento da rede e da própria estratégia de replicação.

Um problema dessa estratégia é fazer com que um nó tenha conhecimento para saber se é melhor manter a mensagem consigo, copiá-la ou transferi-la para um próximo nó. No exemplo ilustrado na Figura 1, no tempo t_2 a mensagem não é transferida nem copiada, pois o nó A tem algum conhecimento indicando que o nó C não tem chances de entrega da mensagem para o nó D . A situação fica mais difícil de tratar se C também tem chances de entregar a mensagem para D , pois A deve ter informações e algum método para avaliar quem tem mais chances de entregá-la. Dependendo das chances de C e A nessa situação, pode ser vantajoso replicar a mensagem.

Como percebido com o exemplo ilustrado, os contatos que estão ocorrendo vem determinando a topologia da rede. O conhecimento de alguma informação sobre a ocorrência dos contatos pode ajudar a determinar a topologia futura da rede. Informações do cenário e do contexto poderão ser usadas em uma estratégia de roteamento para que o processo de encaminhamento consiga decidir pela permanência, transferência ou cópia de cada mensagem para um nó em contato.

2.3.2 Caracterização e classificação dos Contatos

O termo contato empregado na DTN refere-se especificamente a situação em que um nó da rede detecta a presença de outro nó através do seu dispositivo de comunicação sem-fio e os dois ficam aptos a transmitir mensagens. Os contatos ficam sujeitos aos problemas de comunicação e outras eventualidades causadas pela movimentação dos nós e de outros objetos que venham a obstruir o sinal entre os nós em comunicação.

As rotas em uma DTN são compostas por uma sequência de contatos. Neste sentido, os contatos são oportunidades de comunicação dependentes do tempo em que as mensagens podem ser transferidas de uma origem até o destino. Os contatos são descritos em termos da direção, os dois nós em contato, um momento de início, um momento de fim, a banda de comunicação e o atraso.

Em uma rede com contatos oportunistas, a escolha adequada dos contatos a serem usados para a transferência de uma mensagem até seu destino é objeto de estudo em várias pesquisas. Assim, as decisões de encaminhamento devem prever o agendamento das transferências con-

siderando a disponibilidade temporal dos contatos além da sequência de saltos até o destinatário. A indeterminação da duração dos contatos aumenta a dificuldade em encontrar um agendamento adequado para transferência de maiores quantidades de dados (SHAO; LIU, 2009).

Dada a importância de entender como acontecem os contatos em DTN e os cenários em que acontecem (FALL, 2003; CERF et al., 2007), eles são objeto de vários estudos, principalmente para os trabalhos que realizam predição para melhorar a comunicação na rede.

Quanto ao conhecimento da sua ocorrência, os contatos podem ser classificados como:

- deterministas: são os contatos que tem sua ocorrência determinada antecipadamente;
- quase-deterministas: são contatos deterministas que podem não acontecer ou ter sua duração diminuída. O momento no tempo e/ou espaço para sua ocorrência é conhecido. Passando esse momento, os contatos só ocorrerão na sua próxima oportunidade determinada (DIANA et al., 2012);
- quase-oportunistas: aqueles contatos que ocorrem aleatoriamente, mas para os quais pode-se extrair informações que venham a determinar padrões, apesar destes poderem não ser seguidos de forma estrita. Neste caso, existe uma certa expectativa que contatos ocorram em determinada faixa de tempo e/ou espaço, podendo entretanto ocorrer antes, depois, ou não ocorrer. Essa classe de contato foi definida no trabalho de Segundo, Farines e Silva (2013);
- oportunistas: são os contatos que ocorrem aleatoriamente, sem informações sobre quando ou onde irão ocorrer.

No roteamento em DTN deve-se lidar com diferentes tipos de contatos. Assim, podem surgir estratégias de roteamento específicas para determinados contexto e cenários, influenciando os tipos de contatos.

Uma outra classificação mais específica quanto a forma de ocorrência dos contatos é a seguinte (CERF et al., 2007):

- Persistente: eles estão sempre disponíveis. Um exemplo seria um nó fixo ou sempre dentro de uma área de cobertura.
- Sob-demanda: eles ficam disponíveis quando solicitados. Exemplos seriam a realização de uma conexão discada ou se dentro da

área de cobertura é ativada a interface de rede WiFi/3/4G para comunicar.

- Intermitente agendado: quando é possível determinar o momento no qual o contato estará disponível. Por exemplo na comunicação com satélites de órbitas baixas em rede interplanetária (IPN). Também podem ser classificadas dessa forma as situações em que é possível determinar faixas de horários dentro de um ciclo de tempo para a ocorrência do contato e ainda quando houver precisão na localização e regularidade na movimentação.
- Intermitente oportunista: os contatos acontecem ao acaso, sendo difícil de determinar o tempo e o espaço exato em que podem ocorrer.
- Intermitente predito: eles não são baseados em horários e/ou locais fixos, mas são baseados em previsões de horário e/ou espaço e durações prováveis do contato com base em informações históricas de contato previamente observadas. Quando existe confiabilidade suficiente na previsão de contatos desse tipo é possível estabelecer uma estratégia de roteamento.

Os contatos Persistentes, Sob-demanda e Intermitente Agendado podem ser considerados contatos Deterministas. Contudo, diante da presença de falhas os contatos Sob-Demanda e Intermitente Agendado são considerados quase-deterministas.

Os contatos Intermitentes Oportunistas correspondem aos contatos Oportunistas, enquanto os contatos Intermitentes Preditos são os contatos quase-oportunistas.

No presente trabalho são concentrados esforços nos contatos quase-oportunistas que são propícios ao contexto do STPU para melhorar a eficiência na sua predição. Apesar de vários métodos de predição já terem sido utilizados para predição de localização do nó, da velocidade do nó, do tempo em que o nó permanecerá em determinado local e trajetos, este trabalho está focado na predição de contatos.

2.3.3 Gerenciamento da Fila de Mensagens

O gerenciamento da fila de mensagens ou simplesmente *buffer* de mensagens é fundamental para melhorar a entrega e diminuir o atraso de mensagens na DTN com limitações de recursos. Isso deve ser feito através de duas operações básicas (MASIP-BRUIN et al., 2011):

- **Priorização:** Uma vez que o espaço de armazenamento na fila de mensagens e a duração do contato é limitada, a priorização do envio de mensagens deve ser feito favorecendo o melhor uso desses recursos. Por exemplo, para aumentar o número de mensagens entregues, adota-se prioridade na transferência das mensagens que tem maiores chances de serem entregues ao seus destinos. Ou ainda, para diminuir o atraso das mensagens entregues, prioriza-se a transferência das mensagens que tem data de criação mais recente. Dessa forma, o processo de roteamento deverá fornecer instruções para o gerenciamento da fila de mensagens.
- **Descarte:** O descarte de mensagens deve ser aplicado para evitar que mensagens que não tem chances de alcançar o seu destinatário não fiquem ocupando espaço de armazenamento e não sejam transferidas, ocupando tempo de conexão. Assim, o desempenho do roteamento também pode ser melhorado, sendo que as outras mensagens com chances de entrega podem ocupar armazenamento e banda de rede. Um exemplo disso é o caso do descarte de mensagens que tem seu tempo de expiração alcançado ou do descarte de mensagens muito antigas, liberando espaço para mensagens mais recentes e diminuindo o atraso na entrega.

O gerenciamento de filas de mensagem tem um papel mais crítico em estratégias de roteamento baseadas em inundação por causa do consumo excessivo de recursos. Estratégias mais elaboradas de gerenciamento da fila de mensagens poderão permitir que a estratégia de roteamento obtenha melhores resultados.

2.4 ESTRATÉGIAS DE ROTEAMENTO

2.4.1 Abordagens básicas de roteamento DTN

As estratégias de roteamento em rede DTN estão associadas a forma com que os contatos são estabelecidos na rede e aos recursos disponíveis. De forma geral as abordagens de roteamento de mensagens em DTN podem adotar as seguintes estratégias:

- **inundação:** esta estratégia realiza cópia de todas as mensagens no *buffer* de um determinado nó para todos os nós que fazem contato com o mesmo até que a mensagem encontre seu destinatário. As mensagens são replicadas enquanto houver recursos de banda de

rede e de espaço para armazenamento nos *buffers* de mensagens dos nós. As mensagens que já alcançaram seu destinatário e aquelas sem chance de alcançar o destinatário no tempo necessário para diminuir o consumo de recursos são eliminadas. Neste sentido é importante adotar uma estratégia de gerenciamento das mensagens do *buffer*. Redes caracterizadas por contatos oportunistas normalmente conseguem extrair vantagens adotando essa estratégia. Um exemplo desta é implementada na estratégia de roteamento Epidemic (VAHDAT; BECKER, 2000).

- sem cópia: nesta estratégia somente a mensagem original é encaminhada pelos nós da rede até que encontre o destino ou que algum limite de tempo ou outra condição determine que a mensagem não será mais útil e será descartada. Ela procura determinar a topologia futura para realizar o encaminhamento das mensagens. Esta estratégia apresenta um baixo consumo de banda de rede e o desempenho de entrega de mensagens normalmente acaba sendo pior do que as baseadas em cópia de mensagens. Nas redes com contato determinista essa estratégia normalmente oferece melhores resultados. Alguns exemplos de estratégias que podem permitir o envio de mensagens sem cópia são o MaxProp (BURGESS et al., 2006), RAPID (BALASUBRAMANIAN; MAHAJAN; VENKATARAMANI, 2010), RUTS (DOERING, 2012) e o RCM (LIU; WU, 2011) quando o número de cópias é igual a zero.
- cópia limitada: se a estratégia principal for baseada em inundação, deve haver um controle da direção das cópias, não especificamente no sentido espacial, para diminuir o número de mensagens na rede, sem diminuir excessivamente as chances de entrega. Se for baseado em uma estratégia sem cópia, deve-se modificá-la para permitir um determinado número de cópias da mesma mensagem, copiando a mensagem somente para os nós que tem maiores chances de entregá-la ao seu destino. Dessa maneira, aumenta-se o consumo de banda de rede e *buffers*, mas com uma maior chance de entrega da mensagem. Nos dois casos a decisão de transferência da mensagem deve ser baseada em informações do cenário, tal como o tipo de contato (oportunistas, quase-oportunistas e quase-deterministas). Um limite para o consumo de recursos ou alguma negociação entre consumo de recursos e taxas de entrega e atraso mínimos e máximos desejados é geralmente adotado. Este é normalmente implementado pelo controle do número máximo de cópias de uma mesma mensagem na rede

e/ou por uma função utilidade. Exemplos dessas estratégias são: Spray & Wait (SPYROPOULOS; PSOUNIS; RAGHAVENDRA, 2005), Spray & Focus (SPYROPOULOS; PSOUNIS; RAGHAVENDRA, 2007), RAPID, MaxProp, RUTS (DOERING, 2012) e o RCM (LIU; WU, 2011) com número de cópia maior que zero.

Muitas pesquisas propõe o tratamento do roteamento na presença de contatos oportunistas. Algumas delas tentam tratar os contatos oportunistas usando algum método utilizado em aprendizagem de máquina (MAM), tal como processos markovianos (YUAN; CARDEI; WU, 2012), redes bayesianas (AHMED; KANHERE, 2010), lógica difusa (MAMOUN; BARRAK, 2013), aprendizagem por reforço com multi-agentes (ROLLA; CURADO, 2013) e Redes Neurais Artificiais (RNAs)(SEGUNDO; FARINES; SILVA, 2013).

Neste contexto de contatos, um dos cenários existentes é o do STPU. Neste cenário, por causa dos itinerários e agendamentos das paradas nos pontos de ônibus baseado em uma tabela de horários, os contatos ocorrem de forma quase-oportunista (SEGUNDO; FARINES; SILVA, 2013).

2.4.2 O Roteamento no Contexto do Sistema de Transporte Público Urbano

O cenário

Considera-se um serviço de transporte público onde os ônibus equipados com dispositivos de comunicação sem fio omnidirecionais de curto alcance são os nós de uma rede DTN. Cada linha de ônibus executa uma rota em determinada faixa de tempo e possui pontos de parada. Cada nó se move de acordo com um itinerário fixo em rotas circulares. Há um agendamento para os nós estarem nos pontos de parada e não há como determinar o tempo de permanência nesses pontos. Cada nó tem um identificador único, bem como cada linha e cada rota. Considera-se que a cada novo dia a movimentação dos nós se repete nas mesmas linhas de ônibus e em horário próximos ao do agendamento.

Dessa forma, é assumida a existência de ciclos diários de movimentação nos quais o nó inicia na mesma posição e repete o itinerário do dia anterior na mesma sequência. É esperado que a movimentação dos nós esteja sujeita a variações na velocidade e falhas ao se dirigir para os pontos de parada.

Algumas premissas são adicionais a esse cenário: não há inserção

ou remoção dinâmicas de nós no sistema; há uma baixa densidade de nós sobre uma grande área; não existe informação de posicionamento; e é assumido que cada nó consegue se comunicar em uma raio limitado de alcance.

Os contatos entre ônibus neste cenário acontecem com um grande grau de oportunismo. Apesar de existir uma tabela de horários e informações sobre as linhas de ônibus, a variação na movimentação e falhas fazem os contatos acontecerem ou não, antes ou depois de um determinado local ou momento no tempo. Assim, não é possível determinar precisamente em que tempo e espaço um contato irá acontecer, desde que as características destes são modificadas substancialmente a cada repetição do itinerário. Considera-se que seja possível obter informações de contatos em ciclos passados, sendo possível separar os contatos estabelecidos em uma determinada faixa de tempo do ciclo.

Devido à natureza do problema de transporte e apesar de possibilidade de uma grande variação nas oportunidades de contato observadas neste contexto, os nós apresentam uma tendência de organização temporal e espacial para cumprimento dos itinerários, mesmo que estes não sejam seguidos de forma estrita. Considera-se neste trabalho que tirar proveito dessa organização melhorará a comunicação na rede DTN.

Estabelecimento dos contatos no STPU

Um estudo sobre o estabelecimento dos contatos entre ônibus do STPU já havia sido conduzido por Doering (2012). Ele apresenta as formas básicas de encontro entre ônibus que podem resultar em contatos:

1. Encontro: dois nós passam um pelo outro em direções opostas. O contato ocorrido tem curta duração e eventualmente pode não acontecer dependendo da velocidade dos nós.
2. Interseção: dois nós passam um pelo outro em direções que se cruzam. O contato ocorrido tem curta duração e acontece quando coincide a passagem dos nós no mesmo tempo pelo cruzamento. Essa coincidência é difícil de prever, entretanto devido à redução de velocidade nos cruzamentos, o contato é possível.
3. Seguindo: um nó segue o outro por uma parte do caminho. Os contatos tem muitas chances de acontecer, normalmente repetidas vezes e totalizando uma grande duração.

4. Um nó parado: um nó passa pelo outro que não se move. Existe boas chances de acontecer um contato, pois a manutenção do nó em determinada região, seja parado ou em movimentação fraca favorece a oportunidade de contato.

O sentido, a velocidade e a permanência em pontos de parada, cruzamentos e congestionamento influenciam a realização dos contatos. Uma forma de utilizar esses fatores observados seria aplicá-los em equações que informariam as chances de realização de contato em cada itinerário realizado e utilizá-lo para favorecer o reconhecimento de contatos factíveis. Apesar disso, ainda é complexo identificar de forma automática os movimentos, trechos e quando os nós estão sujeitos aos fatores citados.

Outra observação pode ser feita baseado nos estudos de Zhang et al. (2007) e Pelusi, Passarella e Conti (2006). Conforme esses autores, o tempo dos intercontatos é caracterizado por uma função de distribuição acumulada de cauda longa, aproximando-se a uma distribuição exponencial com índice maior que 1. Nas observações feitas no cenário do STPU, isso acontece fortemente influenciado pela dispersão dos ônibus após deixarem uma área central de encontro (terminal ou garagem). Quando os nós vão se afastando de um ponto de partida compartilhado, o sincronismo com o itinerário preestabelecido vai diminuindo e o tempo de intercontato vai aumentando.

Devido às dificuldades de modelar o acontecimento dos contatos nesse contexto e por ser possível identificar certos padrões que ocorrem a partir, por exemplo, das formas de contato e distribuição dos intercontatos, pode-se dizer que esses contatos não ocorrem de forma totalmente oportunista. Tais padrões ajudam a perceber a existência de regularidades a serem exploradas no roteamento, mas que normalmente não são suficientes para se empregar métodos deterministas. Os contatos que ocorrem nessa circunstância, caracterizados como quase-oportunistas, se tratados adequadamente devem permitir melhorar o roteamento nas redes DTN.

2.5 O PROBLEMA DE ROTEAMENTO EM DTN

Existe uma série de questões a serem tratadas no roteamento em DTN, especialmente no contexto dos contatos oportunistas.

Zhang (2006) considera importante para o estudo do roteamento em redes DTNs os seguintes estudos:

1. determinar o número de nós para encaminhar a mensagem, quantas cópias manter na rede e qual métrica usar para um nó ser escolhido para encaminhar uma cópia;
2. definir a auto-aprendizagem e mecanismos de automação: tomada de decisão inteligente para o agendamento e encaminhamento de pacotes;
3. agendar transmissões (quando um certo grau de determinismo o permita);
4. utilizar informações de localização, perfil do usuário e estimativa de disponibilidade dos enlaces;
5. indicar o grau de confiabilidade da rede;
6. ajustar objetivos de roteamento (taxa de entrega, atraso médio de entrega, etc) de acordo com a capacidade do sistema.
7. balancear a utilização da banda com outros recursos não usados;
8. eliminar pacotes duplicados na rede;
9. definir o uso de antenas direcionadas quando for necessário para aumentar o alcance e probabilidade de entrega do pacote; e
10. definir o uso de métodos de DTN para sistemas de comunicação com antenas direcionais e/ou sistemas que economizam energia (com agendamento, encaminhamento, entre outros).

Dentre estas considerações, a primeira e a segunda são de especial interesse na pesquisa dessa tese.

Leguay (2007) relacionou considerações importantes quanto ao roteamento e ao encaminhamento. Diante dos problemas de conexão intermitente, um caminho completo confiável será muito difícil de existir. Assim, as principais questões no roteamento dizem respeito às técnicas e conhecimentos que poderão permitir abordagens com garantia de entrega e menor atraso possível na entrega de pacotes. Neste sentido, métodos que permitam a extração de informações do contexto e do cenário podem ajudar a definir estratégias que melhorem o encaminhamento das mensagens pela rede. E, dessa forma, torna-se importante identificar o tipo de informação em determinadas situações e configurações de rede que são relevantes para a melhoria do roteamento.

Sendo assim, o problema de roteamento DTN está muito relacionado com o conhecimento que caracteriza o tipo dos contatos estabelecidos na rede.

2.5.1 O problema de roteamento com contatos quase-oportunistas

O problema a ser tratado neste trabalho é o roteamento DTN em contextos de contatos quase-oportunistas, tal como no STPU.

Uma hipótese para propor um método de roteamento neste contexto consiste em encontrar relações nos dados de contatos que permitam prever com um certo percentual de acerto a realização dos próximos contatos a partir da observação de um histórico de longo prazo repetitivo, levando-se também em consideração o instante (ou momento) do contato.

No histórico de contatos existem informações importantes para a predição de contatos, tal como os nós envolvidos em um contato e o momento no tempo dentro de um ciclo em que esse contato ocorre. Neste trabalho assume-se que esse ciclo seja diário. Na observação de vários ciclos passados, pode ser determinado que um contato entre dois nós tem grandes chances de acontecer dentro de uma faixa de tempo no ciclo ou não. Pode-se ainda estabelecer que essas chances do contato acontecer estejam relacionadas com a ocorrência de contatos anteriores.

Determinar o acontecimento de contatos nesse contexto pode não ser simples, pois isso depende da movimentação dos dois nós que farão contato e que dependem de fatores externos muito difíceis de serem preditos. No cenário do STPU, esses fatores são o tempo de permanência nos pontos de parada embarcando e/ou desembarcando passageiros, congestionamentos devido ao tráfego, tempos de espera em cruzamentos, acidentes de trânsito, falhas nos ônibus, problemas de comunicação devido à objetos que venham a impedir a propagação do sinal, e outras manifestações difíceis de modelar. Um exemplo disso é ilustrado na Figura 2.

A Figura 2 mostra a variação nas oportunidades de contatos em dias diferentes. Isso prejudica a predição de contatos para o encaminhamento de uma mensagem entre os ônibus B e D. Nesta figura são apresentadas 3 linhas que representam os mesmos ônibus executando suas linhas em dias (ciclos) diferentes das 7 as 9 horas da manhã. Acima da linha estão representados os problemas que podem ocorrer para modificar a movimentação dos ônibus. Abaixo da linha os ônibus que estão sendo contactados ao longo do itinerário.

No primeiro dia (ciclo), o ônibus da linha A passa por congestionamento, espera no semáforo, espera no ponto de ônibus e então faz contato com o ônibus da linha B que transfere uma mensagem destinada a D. Passa por mais congestionamento e espera pela entrada e

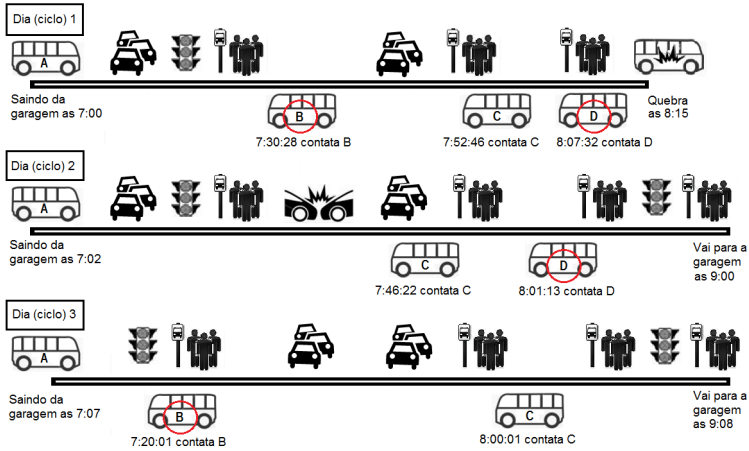


Figura 2: Exemplo de movimentações e contatos diários entre ônibus do STPU com encaminhamento de uma mensagem.

saída de passageiros em outro ponto de parada para então contatar o ônibus da linha C, depois contata D e entrega a mensagem de B. No segundo dia, A parte um pouco atrasado, perdendo o contato com B e sofre espera por causa de um acidente, antecipando contato com C e D em momentos não usuais. Como não contatou B, não tem a mensagem de B destinada a D. No terceiro dia há uma configuração diferente nos contatos entre os mesmo nós das respectivas linhas por que não sofre congestionamento no início do itinerário e o ônibus da linha B está adiantado. Contudo, B consegue entregar a mensagem destinada a D para A, mas A não contata D.

Nesse caso, o histórico de contatos é utilizado para melhorar o roteamento em DTN para cenários nos quais os nós apresentam movimentação cíclica.

Suponha que no exemplo ilustrado na Figura 2 existisse um número bem maior de dias constituindo um histórico de contatos de longo prazo, representando realizações de contatos parecidas e em determinadas situações. No histórico de longo prazo tem situações suficientes que caracterizam as principais movimentações dos nós da rede e conseqüentemente dos contatos que são estabelecidos. Todavia, extrair tais movimentações e identificar as possíveis relações entre elas e as informações de contato recentes são problemas complexos. Modelar os atrasos, adiantamentos e falhas a partir do histórico também é um

problema difícil de resolver.

Assim, um problema relevante a tratar consiste em determinar uma forma de explorar melhor as informações de contato com o objetivo de tirar proveito de regularidades na ocorrência dos contatos. Este método deve funcionar para o contexto de contatos quase-opportunistas e permitir a otimização de determinado critério, tal como o menor atraso na entrega das mensagens ou o aumento do número de mensagens entregues.

2.6 PROPOSTAS PARA O ROTEAMENTO DTN

Muitos trabalhos apresentam formas de estabelecer uma topologia futura para a rede através de diversos métodos para vários cenários, tal como o do STPU.

Realizar um roteamento eficiente em DTN no caso de contatos oportunistas é uma tarefa difícil devido à aleatoriedade, em particular por que informações são difíceis de serem extraídas do contexto e cenário de interesse.

Pesquisas mostram que métodos de predição apresentam melhorias para o roteamento DTN. Muitas destas utilizam o apoio de algum método probabilístico (LINDGREN; DORIA; SCHELÉN, 2003; BURGESS et al., 2006). Outras ainda incrementam a estratégia de roteamento com o uso de processo decisório de Markov (LIU; WU, 2011), redes bayesianas (AHMED; KANHERE, 2010), processos semi-markovianos (YUAN; CARDEI; WU, 2012), entre outros métodos utilizados para predição. Pesquisas que tratam o problema de alocação de recursos através do uso de funções de utilidade estão sendo também realizadas (BALASUBRAMANIAN; LEVINE; VENKATARAMANI, 2010; CAO; CRUICKSHANK; SUN, 2011).

Outra preocupação diz respeito a comparação dos resultados em pesquisas diferentes, dificultada pelas DTNs operarem em contextos e cenários diferentes e que são sensíveis as estratégias utilizadas para roteamento. De modo geral é verificada uma grande quantidade de trabalhos que trazem implementações de estratégias de roteamento com abordagens específicas para determinados cenários. Neste sentido,ENDERLE, PETZ e JULIEN (2009) confirmam a influência do cenário nos resultados e concluem sobre a necessidade de se desenvolver de um *framework* para testes de redes DTN que leve em consideração condições idênticas dos cenários e sistemas de simulação, sendo de grande validade para a realização avaliações mais adequadas.

Foram estudadas mais particularmente algumas propostas de interesse para este trabalho. Estas estão organizadas segundo dois pontos de interesse: a utilização do contexto do STPU e a utilização de algum método de redes neurais artificiais (RNA). Ao final de cada parte são apresentadas as relações de cada trabalho com essa pesquisa.

2.6.1 Propostas de Roteamento no Contexto do STPU

A proposta de Lindgren, Doria e Schelén (2003)

Lindgren, Doria e Schelén (2003) propuseram uma abordagem de roteamento probabilístico, o *Probabilistic ROuting Protocol using History of Encounters and Transitivity* (PRoPHET), no qual a probabilidade de entrega para os nós é calculada usando o histórico de contatos e também através da probabilidade de contato de outros nós (transitividade). A probabilidade de entrega de todos os outros nós para todos os destinos conhecidos é mantida por cada nó, sendo que a cada encontro as informações de contato são trocadas entre os nós. Apesar de não levar em consideração a informação do momento do contato, essa abordagem pode ser aplicada no cenário do STPU baseado na frequência de contatos entre os nós.

Nos experimentos originais do PRoPHET, a mensagem é copiada somente para outro nó que apresenta maior probabilidade de encontro com o destinatário da mensagem. A limitação para cópia de mensagens depende das filas de mensagens, todavia não apresenta sobrecarga pela característica seletiva da replicação. Nos experimentos realizados o PRoPHET diminuiu o consumo de banda de rede e filas de mensagem em relação ao Epidemic (VAHDAT; BECKER, 2000). Uma desvantagem apresentada é a sobrecarga de mensagens para a troca de informações.

A proposta de Burgess et al. (2006)

Uma estratégia de roteamento eficiente para DTN, originalmente com cópia ilimitada de mensagens, foi proposta por Burgess et al. (2006). A estratégia denominada MaxProp determina o caminho de máxima probabilidade para o destinatário da mensagem e envia uma cópia somente quando o nó em contato tiver maior probabilidade de contato com o destino. Também foi proposta uma forma de gerenciamento da fila de mensagens com priorização das mensagens a serem

transmitidas e as serem descartadas. Essa proposta foi avaliada com dados de um experimento real, o projeto DieselNet (BURGESS et al., 2008), que também é utilizado neste trabalho.

O MaxProp foi proposto e avaliado em um cenário de VANET formada por ônibus do STPU com limitações de duração dos contatos e do tamanho nas filas de mensagens. Essa abordagem não leva em consideração o instante no qual o contato é estabelecido, baseando-se na frequência dos encontros entre os nós, tal como no PRoPHET.

A estratégia de roteamento do MaxProp é baseada no encontro entre contatos. Para isso, cada nó mantém um vetor de probabilidade de entrega que é atualizado quando ocorre um contato usando a média incremental. O caminho de menor custo para o nó de destino da mensagem é escolhido na decisão de transferência da mensagem.

Nos experimentos de Li et al. (2008), Doering, Pögel e Wolf (2010) e Socievole et al. (2011) o MaxProp superou o PRoPHET na taxa de entrega, no atraso de entrega e em outras métricas avaliadas. Entretanto, ele foi inferior apenas na ocupação do processador e memória para execução, piorando quando aumenta o número de nós na rede.

A proposta de Liu e Wu (2011)

O trabalho de Liu e Wu (2011) apresenta a estratégia *Routing in Cyclic Mobispace* (RCM). Nesse trabalho é verificado que o STPU tem características de contatos que melhoram a eficiência do roteamento DTN por causa da previsibilidade temporal e espacial.

Liu e Wu (2011) utilizam técnicas de transformação de grafos para lidar com a complexidade inserida pelo tempo nos grafos da rede. A métrica de mínimo atraso esperado foi criada para diferenciar as ligações entre nós de acordo com o tempo que a mensagem irá levar para chegar ao nó ligado em um tempo futuro. Para considerar ligações através de outros nós (transitividade), essa métrica é determinada através de um processo de decisão de Markov usando informações do histórico de contatos.

Nas simulações realizadas, os resultados obtidos utilizam os registros de movimentação real de dois projetos: o DieselNet (BURGESS et al., 2008) e o NUS (*National University of Singapore Student Trace*) (KARAGIANNIS; BOUDECE; VOJNOVICI, 2010 apud LIU; WU, 2011). Apesar da estratégia de roteamento proposta levar a resultados promissores quanto ao uso de informações de agendamento em ciclos, inclusive superando o MaxProp, Hartog, Spyropoulos e Legendre (2010) apontam

que tais resultados são apenas obtidos para cenários de pequeno porte.

A proposta de Ahmed e Kanhere (2010)

As redes bayesianas são exploradas no trabalho de Ahmed e Kanhere (2010) para melhorar o encaminhamento de mensagens em uma rede DTN formada por nós do STPU, aproveitando-se das informações do instante e local de contato.

Os autores introduzem um *framework* estendível para decisões de roteamento baseado em classificação bayesiana que integra conhecimento das características da rede e dos padrões de mobilidade. O espaço de movimentação foi dividido em áreas quadradas e o tempo foi organizado em *time-slots*.

No exemplo de utilização do *framework* desenvolvido são usadas duas classes simples baseadas nas estatísticas de pacotes entregues previamente. As métricas de roteamento empregadas são a probabilidade de entrega posterior condicionadas ao atributo de tempo e a localização espacial.

A avaliação da eficiência nesta abordagem baseia-se em simulações sobre dados reais de mobilidade dos ônibus no sistema de transporte King County Metro em Seattle¹, sendo possível melhorar a taxa de entrega de mensagens em 25% comparado com o protocolo de roteamento MaxProp.

A proposta de Doering (2012)

Doering (2012) apresenta o *Routing in Urban Public Transport Systems* (RUTS), uma estratégia de roteamento DTN utilizando as informações de mapas e horário de paradas de ônibus que possibilita melhoras em uma rede veicular operando sobre o STPU. Foram realizadas comparações com o MaxProp e o PRoPHET. Essa abordagem explora propriedades da movimentação dos ônibus tais como o sentido e o trajeto.

Nas simulações são usadas informações do STPU de uma área urbana para montar um cenário sintético. Mapas baseados na cartografia real combinados com as definições de linhas, paradas e tabela de horários do sistema de transporte público real foram utilizados para aumentar o realismo. Um simulador de micromobilidade produziu registros de mobilidade em grande escala que foram introduzidos em um simulador DTN. O cenário simulado tem 5 horas com 54 paradas e 13

¹http://crowdad.cs.dartmouth.edu/rice/ad_hoc_city/

linhas, sendo que 28 nós ficam simultaneamente no cenário e a cada 2 minutos são geradas mensagens.

A estratégia proposta obtém as probabilidades de contato através de informações de contexto extraídas dos dados do projeto Dieselnet e, a partir dos dados do STPU avaliado, determina um grafo direcionado de roteamento e então calcula as probabilidades de contato e duração dos contatos entre todos ônibus. Para determinar os n melhores caminhos a serem utilizados para enviar as n cópias da mensagem é calculada uma pontuação através das informações desse grafo. Essa pontuação leva em consideração as informações da soma dos tempos de todos os caminhos do grafo menos o tempo do caminho escolhido, a frequência de passagem pelos nós e interseções, e o número de mensagens transmitidas no caminho. As chances de obter um contato considerando o acontecimento de distúrbios imprevisíveis na rede, é desta forma levada em consideração.

Nos experimentos a estratégia proposta aumentou a taxa de entrega em 20% comparado com o MaxProp.

A proposta de Yuan, Cardei e Wu (2012)

Em (YUAN; CARDEI; WU, 2012), os autores verificaram que o movimento dos nós de uma rede que tem comportamento social não movem-se totalmente ao acaso e utilizam a informação temporal dos contatos em um modelo de processo semi-markoviano de tempo homogêneo. Apoiado nesse modelo, uma estratégia de roteamento sem replicação de mensagens chamada *Predict and Relay* (PER) foi proposta.

Nesta estratégia, os nós determinam a distribuição de probabilidade de contato e escolhem o próximo salto da mensagem para aumentar a probabilidade de entrega de mensagens. Essa estratégia considera a observação de que um nó move-se ao redor de pontos mais visitados ao invés da aleatoriedade e que existe um tempo de permanência nesses pontos que pode ser previsto através do histórico de movimentações. Para determinar em qual ponto de parada o nó tem mais chances de estar, um modelo semi-markoviano foi adotado e estendido para considerar o tempo de deslocamento do nó entre dois pontos de parada.

Os resultados da estratégia proposta utilizando três funções diferentes foram apresentados em termos da taxa de entrega e atraso na entrega. Foi utilizado o resultado da estratégia epidêmica ilimitada e da entrega direta para mostrar os limites de desempenho desta estratégia, contudo não foram apresentadas comparações com o PROPHET ou o

MaxProp.

Apesar desse trabalho não corresponder especificamente ao cenário do STPU, o contexto da realização dos contatos caracteriza-se por apresentar contatos quase-oportunistas.

2.6.2 Comparação das propostas

Os trabalhos revisados nesta seção são de interesse para o tratamento do problema dessa pesquisa por apresentar estratégias de roteamento no contexto do STPU.

As Tabelas 1 e 2 apresentam as principais diferenças no cenário e nos métodos utilizados nesses trabalhos.

Os cenários

Nas diferentes propostas estudadas, pode-se verificar que há variações significativas quanto ao método de simulação, na forma de tratamento do conjunto de dados, na forma de geração das mensagens, no número de nós, tempo de simulação e abrangência do dispositivo de comunicação sem fio.

Tabela 1: Comparação dos cenários utilizados nas propostas de roteamento DTN no contexto do STPU.

Proposta	Dados: Mobilidade	Simulador: Simulação	Nós/ Qtd.histórico/ Área coberta/ Alcance rádio	Geração de mensagens
PRoPHET: Lindgren, Doria e Schelén (2003)	Dados sintéticos: RWP	Próprio (simples): movimentos, comunicação e troca de mensagens	50 nós / - / 1500 x 300m / 50 e 100m	todos para todos a cada segundo.
MaxProp: Burgess et al. (2006)	Dieselnet: sintético baseado no real; e reconstruído.	Não informado: movimentos, comunicação e troca de mensagens	30 nós / 60 dias / 2414Km ² / 15,24 até 122m	Cada nó de 2 a 18 msg/s/hora.
RCM: Liu e Wu (2011)	DieselNet: reconstruído	Não informado: troca de mensagens; e outras	28 nós / 55 dias / 2414Km ² / 35 a 140m	Todos para todos a cada 10 minutos.
Bayesian Routing: Ahmed e Kanhere (2010)	STPU: reconstruído	Próprio: mobilidade, comunicação, troca de mensagens	35 nós / 500 x 9hs / 58 Km x 88 Km / Não informado	50 pares aleatórios de (origem,destino) a cada 100 segundos.
RUTS: Doering (2012)	Simulação em área urbana: baseado no Dieselnet	SUMO e ONE: mobilidade, comunicação, troca de mensagens	28 nós / - / 2 x 2,5Km / 50 até 500m	Não encontrado.
PER: Yuan, Cardei e Wu (2012)	Simulação no campus	Próprio: troca de mensagens	30 nós / 2 semanas / 94 x 74m / 35 a 140m	Pares aleatórios geração de mensagens com Poison.

Apesar de as propostas tratarem do mesmo contexto de conta-

Tabela 2: Comparação dos métodos utilizados nas propostas de roteamento DTN no contexto do STPU.

Proposta	Modelo	Método	Resultados
PRoPHET: Lindgren, Doria e Schelén (2003)	Probabilidade	Determina a previsibilidade de entrega	Diminuiu fila de mensagens e atraso e aumentou entrega
MaxProp: Burgess et al. (2006)	Probabilidade	Determina o caminho de máxima probabilidade	Contatos reais, melhorou entrega e diminuiu fila de mensagens
RCM: Liu e Wu (2011)	Grafos variantes no tempo (TVG) e processo decisório de Markov	Transforma TVG em grafos direcionados e determina o mínimo atraso esperado	Contatos reais e sintéticos. Foi 40% melhor no desempenho geral
Bayesian Routing: Ahmed e Kanhere (2010)	Redes Bayesianas	Determina classes de acordo com regiões no mapa e do instante de contato, e calcula probabilidade de participar dessas classes.	Taxa de entrega de mensagens 25% maior do que com o MaxProp
RUTS: Doering (2012)	Grafo espacial-temporal baseado na probabilidade de encontros	Transforma em grafo unidimensional e determina maior pontuação de acordo com probabilidades e caminhos para chegar ao destino.	Menor utilização de recursos, atraso médio e maiores taxas de entrega
PER: Yuan, Cardei e Wu (2012)	Processo semi-markoviano de tempo homogêneo	Determina o mínimo atraso esperado	Mesma taxa de entrega do epidêmico com apenas uma cópia

tos, os experimentos são realizados de forma substancialmente diferentes, dificultando as comparações. Quando há experimento avaliando a mesma estratégia em trabalhos diferentes, fica difícil extrair análises sobre o desempenho devido ao processo de simulação, conjunto de dados e configurações da rede utilizados.

Os métodos

Cada trabalho apresenta uma forma de modelar o problema e trata-o com métodos diferentes, conforme apresentado na Tabela 2. Os métodos adotados pelo PRoPHET (LINDGREN; DORIA; SCHELÉN, 2003) e pelo MaxProp (BURGESS et al., 2006) são apresentadas por se tratarem de trabalhos de referência de roteamento no contexto de contatos que apresentam frequência na repetição. Em várias outras propostas estes trabalhos são utilizados para comparação. Além disso, apresentam implementações disponíveis publicamente, facilitando e tornando mais confiável a realização dos experimentos.

Nos trabalhos mais recentes pode-se perceber uma preferência pelo MaxProp por apresentar melhores taxas de entrega e diminuição do atraso em relação ao PRoPHET, apesar de ambos terem características muito parecidas. Dessa forma, o MaxProp foi escolhido para realizar comparações com a proposta deste trabalho.

O método baseado em processos markovianos utilizado por Yuan, Cardei e Wu (2012) foi um dos primeiros métodos a ser considerado para apoiar a solução do problema apresentado nesta tese. Contudo não foi levado adiante por causa da capacidade de modelar as variações do tempo no estabelecimento dos contatos oportunistas e implicações com o aumento do número de estados do sistema, tornando-se de difícil tratamento.

Yuan, Cardei e Wu (2012) realizou comparações com a estratégia epidêmica ilimitada e com a entrega direta para identificar limites de desempenho e isso foi adotado para avaliação da proposta neste trabalho.

Os trabalhos de Liu e Wu (2011), Ahmed e Kanhere (2010) e Doering (2012) apresentam abordagens de roteamento sem cópia de mensagens para tratamento dos contatos quase-oportunistas.

A proposta de Ahmed e Kanhere (2010) é relevante por apresentar-se no mesmo cenário e utilizar um método de aprendizagem de máquina baseado em redes bayesianas aproveitando-se de informações temporais e espaciais para melhorar o roteamento. Na proposta desta tese também é tirado proveito da informação do instante do contato, contudo não é utilizada informação de posicionamento. Nessa proposta a comunicação entre dispositivos é simulada a partir da reconstrução de dados de mobilidade devido ao intervalo de tempo para atualização, sendo que isso pode permitir perder contatos ou diminuir a duração deles. Por esse motivo, no trabalho desta pesquisa foi preferível o uso dos dados do DieselNet para se obter maior realismo.

Apesar do trabalho de Doering (2012) ser utilizado para o caso de contatos quase-oportunistas, a abordagem adotada para tratamento do problema apoia-se na maior parte em métodos deterministas. Esse trabalho utiliza também os dados do DieselNet, contudo foi necessário reconstruir as rotas por causa dos dados de GPS corrompidos e selecionar rotas específicas para os testes. Isso apontou problemas e soluções para o tratamento desse conjunto de dados. Além do trabalho com dados de contatos reais, aproveitou-se a ideia de utilizar grafos variantes no tempo (TGV) para modelar a DTN. Também foi aproveitado de Liu e Wu (2011) a modelagem com TVG e transformações para diminuir a complexidade da inserção do tempo nos grafos.

Uma observação é que se deve ter muita regularidade na movimentação dos ônibus para conseguir tirar proveito dos mapas e itinerários no contexto do STPU. Isso é devido ao grande número de possibilidades de contatos que acontecem com uma pequena variação na velocidade ou nos tempos de paradas nos pontos de ônibus. Além

disso, os dispositivos de comunicação sem fio baseados no IEEE 802.11b perdem muitos contatos com as interferências, o raio de alcance e a velocidade dos nós. Diante de contatos tão oportunistas encontrados no projeto DieselNet optou-se por um mecanismo que pudesse modelar melhor sua ocorrência.

Os trabalhos revisados apresentam alguns pontos em comum com o trabalho dessa proposta, principalmente por serem aplicáveis no mesmo contexto. Pode-se verificar que a maioria desses trabalhos procura se apoiar em dados reais de contato, porém poucos conseguem utilizá-los diretamente sem a necessidade de reconstrução, interpolação ou obtenção dos seus parâmetros para geração de dados sintéticos. O conjunto de dados do projeto DieselNet são diferenciados, pois apresentam as irregularidades nos contatos provenientes de um cenário real do STPU sem tratamentos, possuem informação do momento exato do contato, fornecem meios de recuperar informação de linha e rota e apresentam um histórico de longo prazo.

Observa-se que a maioria destes trabalhos utilizam métodos probabilísticos e processos markovianos na estratégia de roteamento desenvolvida, porém foram encontrados poucos trabalhos que exploram outros métodos utilizados em aprendizagem de máquina, tais como Redes Neurais Artificiais (RNA), motivando a exploração deste método.

Diferentemente de outros métodos citados, a utilização de RNAs pode trazer vantagens para a predição de contatos, permitindo-se acompanhar variações não lineares na sua ocorrência, como é o caso deste trabalho, devido à influência de várias situações no estabelecimento dos contatos. Dessa forma, a seguir serão revisados os trabalhos encontrados baseados em RNAs que são de interesse para o tratamento do problema de roteamento.

2.6.3 As RNAs nas Propostas de Roteamento

Poucos trabalhos abordam o roteamento em redes *ad hoc* baseado em RNAs. Alguns usam a RNA para auxiliar no controle do *broadcast* e na predição de mobilidade.

As RNAs podem ser vistas como métodos aproximadores de funções, normalmente utilizados em tarefas de classificação e predição. Inspirado nos neurônios biológicos, de forma geral, o modelo matemático de RNA recebe entrada (estímulos) através dos neurônios da camada de entrada que são propagados pelos neurônios das camadas intermediárias até atingir os neurônios da camada de saída. Cada neurônio tem várias

entradas que são ponderadas e somadas e essa soma submetida a sua função de ativação, determinando uma saída. Assim, as ponderações modelam os valores de saída de acordo com as entradas. Um processo de aprendizagem definirá pesos apropriados para que esses valores de entrada correspondam o melhor possível aos respectivos valores de saída, conforme as informações (exemplos) de treinamento utilizados. Pretende-se que as RNAs propostas para o preditor de contatos capturem padrões no estabelecimento dos contatos. Mais detalhes sobre o funcionamento, tipos e aprendizagem das RNAs são apresentados no Apêndice A.

A predição de contatos entre nós de uma rede utilizando RNAs é praticamente inexplorada, principalmente na predição de jornadas, tal como foi utilizada nesta tese. Entretanto, os trabalhos pesquisados trazem informações úteis para a modelagem das RNAs que foram propostas.

A proposta de Kumar e Venkataram (2002)

O trabalho de Kumar e Venkataram (2002), aborda o uso de redes neurais para predição de localização de nós em redes móveis. Esse autor propõe o gerenciamento da localização em redes móveis baseado em predição da localização futura a partir do histórico do padrão de movimento de um nó móvel. A predição foi realizada através de uma RNA alimentada para frente (*Feedforward-FF*) com múltiplas camadas (MLP) e o treinamento é feito com um algoritmo de retropropagação.

O padrão de movimentação é determinado como um conjunto de pares constituídos a partir do sentido e da frequência de movimentos neste sentido. Cada conjunto é referente a um intervalo de tempo determinado. Subconjuntos de cada padrão são utilizados para treinar a rede. Como resultado das simulações, com o padrão de movimentação uniforme a média de precisão da predição foi de 93% e respectivamente de 30% e 2% com os padrões de movimentação regular e aleatório. O método proposto ajudou a diminuir o custo de sinalização para gerenciamento da localização.

A proposta de Kotilainen e Kurhinen (2008)

Kotilainen e Kurhinen (2008) apresentam o NeuroRouter, uma estratégia baseada no conceito de roteamento com mobilidade assistida (MAR - *Mobility assisted routing*) em que os nós são usados como mulas de dados (*data mules*) para transportar as mensagens de um ponto até

o outro. O algoritmo decide o encaminhamento para um próximo nó ou a continuidade do transporte da mensagem de acordo com uma rede neural treinada. O algoritmo adapta seu comportamento baseado em variáveis ambientais, permitindo alcançar estratégias mais eficientes de roteamento.

A rede neural utilizada é uma MLP alimentada para frente. As informações de entrada da rede neural são a velocidade do nó em contato, a diferença da direção entre o nó encontrado e o nó de destino, a estabilidade da direção, a velocidade do nó encontrado, o número de cópias enviadas, o número de saltos desde a origem e a distância até o destino. É retornado como saída a decisão de encaminhar ou não a mensagem para o nó encontrado.

Devido à inviabilidade de recursos para o treinamento dessa RNA em tempo-real, essa abordagem somente é simulada. Apesar de obter resultados favoráveis sobre as decisões de roteamento baseadas na rede neural, não foram realizadas implementações e testes comparativos com outras abordagens de roteamento.

A proposta de Wang et al. (2014)

Um preditor de contatos também baseado em RNAs foi apresentado por Wang et al. (2014). Neste a RNA é usada para determinar a probabilidade de contato com um nó levando-se em consideração o instante em que esse contato é estabelecido. As RNAs são utilizadas para não ficar preso a um modelo que faça uma relação linear entre os dados do histórico de contatos, tal como em outras propostas, pois os contatos são influenciados pelas posições dos nós, trânsito, semáforos, demanda de passageiros velocidade e outros fatores.

Um conjunto de dados testado foi obtido no STPU de Beijing e outro conjunto foi o de Cambridge Hagggle². A predição de contatos futuros pelo preditor proposto superou a predição baseada em encontros realizada no PRoPHET, contudo não é apresentado resultado de experimentos com roteamento.

2.6.4 Relações com o problema a ser tratado

Alguns autores já estudaram o emprego de RNA para auxiliar na resolução de problemas de roteamento em redes sem fio (KUMAR; VENKATARAM, 2002; KOTILAINEN; KURHINEN, 2008). Os trabalhos pi-

²<http://crawdad.cs.dartmouth.edu/cambridge/hagggle/>

oneiros focaram na predição de mobilidade e alguns trabalhos mais recentes iniciaram estudos na predição de rotas, não focando diretamente a predição de contatos, exceto em Wang et al. (2014). Anteriormente a esta proposta, um preditor de contatos baseado em RNA foi proposto no contexto desse trabalho, conforme publicado em (SEGUNDO; FARNES; SILVA, 2013). Apesar de utilizarem o mesmo tipo de RNA, esses preditores de contatos se diferenciam por apresentarem objetivos de predição, estruturas de RNA e algoritmo de aprendizagem diferentes.

A utilização de RNAs do tipo alimentada para frente com MLP para a realização de predições aparece em todos os trabalhos relacionados. Nesta pesquisa, a proposta de predição de contato também se apoia nesse tipo de RNA. A experiência obtida na definição das entradas e das topologias empregadas nesses trabalhos são importantes para a pesquisa deste trabalho. Apesar disso, avalia-se a possibilidade de utilizar RNAs do tipo recorrente, contudo não foram encontrados trabalhos no sentido do problema a ser tratado.

Uma estratégia diferenciada pode utilizar um mecanismo de predição de contato baseado em RNA na construção de jornadas previstas para melhorar a decisão de encaminhamento das mensagens. Uma proposta neste sentido é apresentada nos Capítulos seguintes.

2.7 CONCLUSÃO

As características mais importantes nas DTNs dizem respeito à intermitência na conexão e ao longo atraso na comunicação, que devem ser tratadas de forma adequada para viabilizar o funcionamento das aplicações assíncronas que aceitam essa característica.

Os processos de encaminhamento e de gerenciamento da fila de mensagens influenciam no roteamento, sendo que os critérios utilizados para melhorar as métricas também devem considerar ambas as estratégias e para que os resultados não sejam interpretados de forma errada nas análises.

O roteamento nas DTNs pode necessitar de tratamentos dependentes da forma com que os contatos acontecem. Com mais determinismo nos contatos é possível diminuir incertezas e aplicar métodos de agendamento e otimização para favorecer métricas desejadas. No caso do contexto com contatos oportunistas é mais complicado obter essas melhorias, uma vez que dependem muito da capacidade de extração de conhecimento do contexto e do cenário. Assim, o desenvolvimento de novos métodos que permitam encontrar relações entre os dados dis-

poníveis para melhorar o roteamento em cenários desse contexto é um problema de interesse.

Um desafio interessante surge quando se verifica a possibilidade de utilizar características de movimentação baseada em agendamentos e itinerários para tentar extrair alguma previsibilidade nos contatos, como no caso do STPU. Neste sentido, já existem várias contribuições que utilizam a regularidade da movimentação dos ônibus do STPU. Entretanto estas são de difícil comparação, pois são aplicados em contextos diferentes.

Geralmente as pesquisas são conduzidas com o maior grau de realismo possível. Contudo, os dados de contato reais são difíceis de tratar, devendo passar por reconstruções e outros tratamentos para poderem ser utilizados.

As RNAs são uma opção de tratamento das relações entre contatos passados e próximos contatos, possibilitando algum tipo de vantagem sobre outros métodos não lineares. Contudo, ainda existe pouca pesquisa a este respeito em redes DTN, sendo que a predição de contatos é inexplorada.

Nos capítulos seguintes é detalhada a proposta dessa tese para a solução do problema de roteamento apresentado. É apresentado um preditor de jornadas que servirá como mecanismo de suporte para as tomadas de decisão de roteamento e que utiliza um preditor de contatos baseado em RNAs.

3 ROTEAMENTO BASEADO EM PREDIÇÃO DE JORNADAS

Neste capítulo é descrita uma nova proposta de roteamento baseada na construção de jornadas preditas. Primeiramente são explicados os princípios da solução proposta e descrito o funcionamento de um preditor de jornadas que é baseado em uma função de predição de contatos. Depois é apresentada a proposta de um preditor de jornadas e a descrição dos seus principais algoritmos.

3.1 INTRODUÇÃO

Uma estratégia de roteamento de mensagens em DTN depende da forma com que os contatos vão se estabelecendo na rede. Para contatos quase-oportunistas, uma solução consiste basear o roteamento em possíveis jornadas preditas que possam auxiliar na decisão de transferir ou não a mensagem para o nó em contato. Entretanto, prever uma jornada pode ser difícil à medida que se deve conhecer os nós e os instantes de contato futuros organizados em saltos até os nós de destino.

Uma nova estratégia de roteamento em DTN é proposta com a predição de jornadas a partir de predições de contatos. O preditor de contatos utilizado será detalhado no capítulo seguinte. Ele deve permitir relacionar os possíveis saltos das jornadas. As jornadas deverão ser computadas de acordo com um determinado critério.

Uma estratégia de roteamento DTN pode funcionar no modo sem cópia de mensagens (*single copy*) ou com replicação de mensagens (*multi-copy*). Considera-se o uso de cópias de mensagens para aumentar as suas chances de entrega, contudo há mais utilização de recursos da rede e dos nós. Uma versão dessa estratégia de roteamento considerará o envio sem cópia de mensagens e outra considerará a replicação controlada de mensagens.

Apoiando-se no modelo de grafos variantes no tempo (TVG), nas seções seguintes é descrita uma proposta para o preditor de jornadas. Antes disso, a seguir são apresentados os princípios de funcionamento do preditor proposto.

3.2 UMA PROPOSTA DE ROTEAMENTO BASEADO EM JORNADAS PREDITAS

Conforme ilustrado na Figura 3, uma possível solução para o problema de roteamento enunciado anteriormente consiste em obter um mecanismo que permita prever jornadas para auxiliar nas decisões de encaminhamento das mensagens que estão na fila (tratador da fila). A predição de jornadas deve ser realizada a partir do uso sucessivo de um preditor de contatos. Esse preditor de contatos deve utilizar adequadamente as informações do histórico de contatos para favorecer as predições.

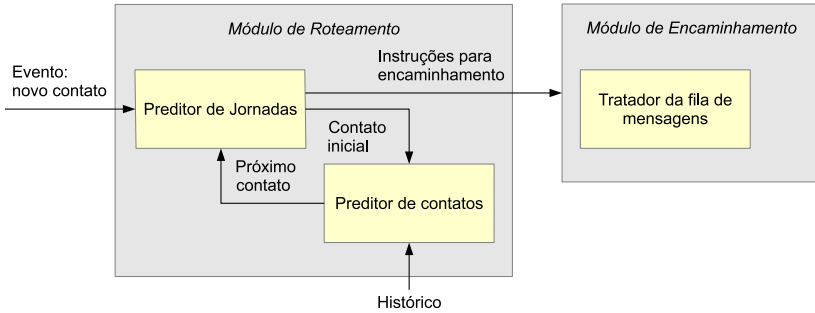


Figura 3: Esquema para solução do problema de roteamento em DTN.

No módulo de roteamento, a cada evento indicando um novo contato como sinal de entrada, as instruções de encaminhamento são determinadas como saída a serem utilizadas pelo módulo de encaminhamento. Para determinar instruções adequadas, o roteamento poderá ser estabelecido através de decisões de encaminhamento de acordo com as jornadas previstas construídas a partir de contatos previstos.

Considerando que cada nó da rede executa um sistema de roteamento DTN conforme a arquitetura da IRTF apresentada no Capítulo 6, a estratégia de roteamento proposta deverá ser implementada no módulo de roteamento dessa arquitetura (módulo Roteador de Agregados) e fornecer instruções de roteamento que permitam entregar mensagens no menor tempo possível ou outro critério a ser definido.

A estratégia de roteamento proposta se distingue do roteamento clássico por utilizar na atualização de rotas duas partes: a predição de contatos e a predição de jornadas. Por sua vez, a predição de jornadas é organizada em duas etapas: a construção de jornadas e a busca por

uma jornada que minimize um critério determinado.

Assim que um novo contato é sinalizado, o tratador da fila de mensagens solicita jornadas preditas informado os destinos desejados e informações do novo contato estabelecido. O preditor de jornadas inicia a construção de um grafo de jornadas começando pelo nó de referência (onde se encontra a mensagem) e pelo nó em contato até os nós de destino encontrados. Cada contato do grafo é obtido a partir das informações de um nó de referência e o instante do tempo desejado. O próximo contato predito deve ser utilizado para continuar a jornada. Ao final da busca por contatos preditos um critério a ser minimizado é aplicado para se determinar as melhores jornadas.

O preditor de contatos proposto deve prever o próximo nó e o momento de contato a partir de informações dos últimos contatos estabelecidos e de acordo com as informações extraídas do histórico de contatos. A função de previsão de contatos terá a capacidade de modelar a ocorrência dos contatos de acordo com padrões ocorridos nos contatos passados. O uso de RNAs foi a opção escolhida e avaliada neste trabalho.

O preditor de jornadas utiliza o preditor de contatos e organiza um grafo representando o conjunto de jornadas possíveis com seus custos para os destinos. Esses custos permitem que o módulo de roteamento forneça instruções para o encaminhamento das mensagens.

3.2.1 Princípios do preditor de jornadas

Na solução proposta cada nó executa um preditor de jornadas que a cada novo contato atualiza os custos de roteamento até os destino e que permitem definir as instruções de encaminhamento das mensagens. Para a determinação desses custos, um preditor de jornadas gera grafos de jornadas preditas de acordo com algum critério previamente estabelecido. O tratador da fila de mensagens pode então avaliar o destinatário da mensagem e decidir pela transferência ou não da mensagem para o nó em contato de acordo com os custos determinados pelo preditor de jornadas.

O preditor de jornadas proposto é estabelecido a partir de dois algoritmos:

- um deles constrói um grafo de jornadas a partir de contatos preditos;
- o outro seleciona a melhor jornada para um determinado nó de

destino.

No momento de um novo contato, as informações desse contato são o ponto de partida para iniciar a predição de contatos. A partir dos contatos obtidos nessa predição de contatos inicial, outras predições de contato são realizadas e assim sucessivamente, estabelecendo as jornadas preditas. O fim das predições de contato acontece quando se chega ao final do intervalo de tempo definido para a predição de contatos. Uma outra opção para finalização consiste em parar quando não for mais possível melhorar o critério determinado para relacionar as melhores jornadas.

O algoritmo construtor de jornadas deve estabelecer um conjunto de saltos representando as jornadas preditas, considerando duas situações para cada contato avaliado:

- a possibilidade da mensagem ser encaminhada para o contato predito, atualizando o nó de referência e o instante do último contato para a nova consulta, e também estabelecendo uma busca por mais contatos preditos;
- a possibilidade da mensagem continuar no nó de referência, mantendo-se o nó de referência e incrementando o instante do último contato para a próxima consulta pelo próximo contato, permitindo a busca em largura por mais contatos preditos.

Ao final da construção de jornadas, um algoritmo de busca pelo caminho de menor custo determinará as jornadas que levam aos destinos de acordo com um determinado critério. Nesse trabalho, o critério utilizado é o de menor atraso na entrega de mensagens. Ao terminar se obtém uma lista de custos para os destinos para se determinar as instruções de encaminhamento.

3.2.2 Utilização do preditor de jornadas

No contexto do STPU, o preditor de jornadas proposto deve ser executado a cada contato feito pelos ônibus equipados com equipamento de comunicação sem fio. Com os resultados das predições é possível decidir sobre a transferência das mensagens para o nó em contato. Para obter predições com informações atualizadas é previsto que as RNAs sejam atualizadas dinamicamente sem que a operação do roteamento seja interrompida. Isso é operacionalizado através de um esquema baseado em duas fases: o tratamento do contato e a atua-

lização do conhecimento. A interação entre essas duas fases é mostrado na Figura 4.

Uma fase fornece informações para a outra. A fase de tratamento do contato alimenta a fase de atualização do conhecimento com informações dos registros dos contatos e informações vindas de outros nós e a fase de atualização do conhecimento fornece novas topologias de RNAs e/ou novos conjuntos de pesos atualizados para a fase de tratamento do contato. A fase de atualização do conhecimento pode ser executada com os ônibus em serviço, mas deve ser interrompida para dar prioridade a fase de tratamento do contato e retomada após o seu término. O funcionamento dessas fases está ilustrado respectivamente nas figuras 5 e 6.

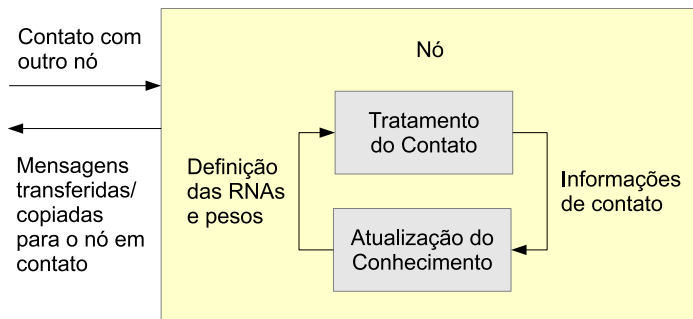


Figura 4: Interação entre as fases do esquema de operação.

Alguns parâmetros dos algoritmos do preditor de jornadas na fase de tratamento do contato podem ter atualização imediata, sem necessitar entrar na fase de atualização do conhecimento. Essas informações são recebidas nos contatos com outros nós e devem ser usadas para ajuste imediato dos parâmetros do preditor de jornadas. Exemplo disso são as informações heurísticas para determinar a validade de contatos preditos e a definição dinâmica do tempo de intercontato (tempo entre contatos) para reconstrução de jornadas.

Fase de tratamento do contato

A fase de tratamento do contato entra em funcionamento quando ocorre um contato. Neste caso a fase de atualização do conhecimento é suspensa ou perde prioridade para que se realize o registro do contato, a troca de informações entre nós, as predições e as transferências ou cópias de mensagens.

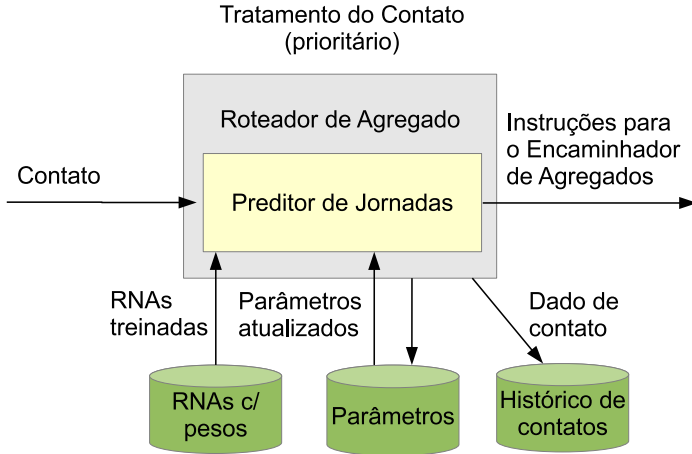


Figura 5: Fase de tratamento do contato.

Na Figura 5 é apresentada a fase de tratamento do contato, sendo que a operação do Preditor de Jornadas (conforme mostrada anteriormente na Figura 3) ocorre no módulo Roteador de Agregado (de acordo com a arquitetura DTN apresentada no Capítulo 6), iniciando com o recebimento do evento de contato. O tratamento do contato inicia com a troca de informações de roteamento com o nó em contato, depois o preditor de jornadas é invocado recebendo as informações do contato realizado e retorna instruções de encaminhamento das mensagens. O Preditor de Jornadas vai solicitando contatos ao Preditor de Contatos baseado no contato atual e nos contatos que vão sendo preditos (conforme explicado no Algoritmo 2). Dessa forma, o Preditor de Jornadas retorna ao Roteador de Agregados um grafo com as jornadas preditas de acordo com um critério determinado e este retorna os custos para alcançar os nós relacionados no grafo (conforme o Algoritmo 1) para que o módulo Encaminhador de Agregados, possa tomar as decisões de transferência ou cópia para cada mensagem (conforme o Algoritmo 3).

A obtenção de informações sobre RNAs de outros nós (no caso de RNAs locais) e heurísticas (parâmetros) deve acontecer quando os nós entram em contato. Um algoritmo pode determinar a prioridade para usar o tempo ou para transferir mensagens ou trocar informações. Essa troca de informações de histórico de contato, definições de novas RNAs e pesos, é realizada pelo protocolo de comunicação.

Fase de atualização do conhecimento

Na fase de atualização do conhecimento as RNAs são recriadas e treinadas utilizando as rotinas de Pré-processamento, Criação dinâmica de RNAs e Treinamento desenvolvidas para o preditor de contatos apresentado no Capítulo 4.

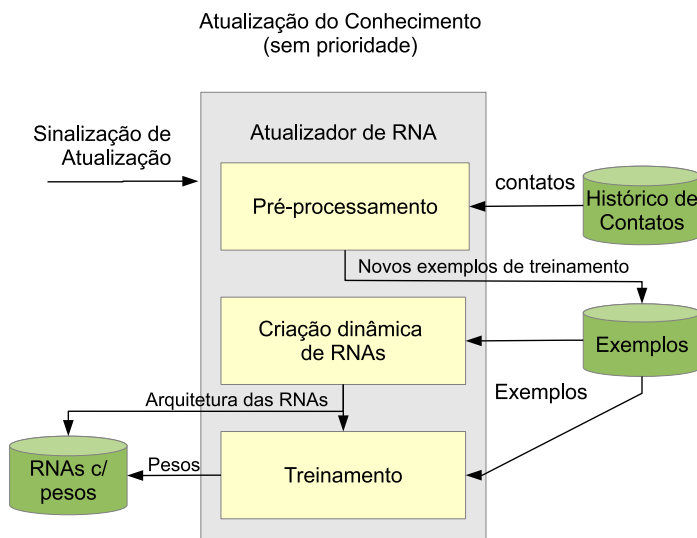


Figura 6: Fase de atualização do conhecimento.

A necessidade de atualização do conhecimento deve ser sinalizada por um processo que pode ser simples, gerando uma sinalização no final de cada dia (ciclo) de operação do ônibus, quando o ônibus estiver na garagem. Assim, o pré-processamento, a criação dinâmica das RNAs e o treinamento pode ser realizado quando os ônibus não está em serviço, sendo que todos os registros de contatos coletados nesse dia são utilizados em um novo treinamento. Um processo de sinalização mais elaborado poderia realizar a detecção de modificações relevantes no histórico de contatos, evitando atualizações desnecessárias.

A sinalização de atualização pode indicar ser necessário atualizar a codificação de entrada e a topologia da rede ou somente realizar um novo treinamento. Em ambos, o pré-processamento do histórico de contatos é realizado. No pré-processamento os dados de contatos são tratados para se obter os exemplos de treinamento, além de serem levantadas informações importantes para redefinição da codificação das

entradas da RNA e redefinição da sua topologia. Esse novo conhecimento somente estará disponível para a fase de tratamento do contato quando a fase de atualização do conhecimento for totalmente finalizada.

3.3 MODELAGEM DO PREDITOR DE JORNADAS

O preditor de jornadas foi modelado para gerar um conjunto de saltos com os contatos preditos, que permitirão reconstruir as jornadas preditas. Assim, na sequência é apresentada a modelagem de uma rede DTN que servirá de suporte para o modelo de jornadas preditas baseado no preditor de contatos proposto.

3.3.1 Modelo de rede DTN

Existem algumas formas de modelar uma DTN. Uma forma julgada adequada para este trabalho consiste em modelar a DTN como um *Time Varying Graph* (TVG) $G(t) = (V(t), E(t))$ em um instante $t \in \mathbb{R}$, onde $V(t)$ é o conjunto de todos os nós (vértices) na rede e $E(t) = \{(u, v) : u, v \in V(t)\}$ é o conjunto de todas as arestas correspondendo aos contatos conectando pares de nós no tempo t . Quando não há inserção ou remoção de nós $V(t)$ é considerado constante $V(t) = V$ (RAMANATHAN; BASU; KRISHNAN, 2007).

Quando existem mais informações sobre os contatos a serem modeladas, as arestas podem incluir outras informações. Como por exemplo, a taxa de transferência. O conjunto E é definido então por $E(t) = \{(u, v, b) : u, v \in V(t), b \in \mathbb{R}\}$, onde b é a quantidade de informação transferida por unidade de tempo no tempo t .

Na DTN, o caminho com trechos intermitentes que uma mensagem pode potencialmente seguir até alcançar o destino é chamada jornada. Na jornada, um salto ocorre quando uma mensagem é transferida entre um par de nós da rede. Um salto s_n , onde $n \in \mathbb{N}$ é o número de sequência do salto, pode ser apresentado pelo par (e_n, t_n) com $e_n \in E(t)$. Uma jornada consiste em uma sequência crescente de saltos ordenados por t . Uma jornada $j_{(u,v)}$ de um nó u até o nó v pode ser apresentada como uma sequência ordenada de saltos $((e_0, t_0), (e_1, t_1), \dots, (e_n, t_n))$, com $e_0 = (u, \cdot)$ e $e_n = (\cdot, v)$.

Dessa forma, o conjunto $J_{(u,v)}$ de jornadas do nó atual u até o nó de destino v , deve conter todos os saltos das jornadas $j_{(u,v)}$ entre u e v . Quando não houver jornadas entre dois nós o conjunto é vazio.

A partir desse conjunto de jornadas, pode-se definir um grafo e aplicar um algoritmo de menor custo para otimizar um critério a ser definido. Por exemplo, a jornada que minimiza o número de saltos de u até v pode ser definida como:

$$\min_s(j_{(u,v)}) = \arg_{j_{u,v}} \min\{|j_{u,v}| : j_{u,v} \in J_{(u,v)}\} \quad (3.1)$$

Ou seja, dentre todas $j_{(u,v)} \in J_{(u,v)}$, procura-se o $j_{(u,v)}$ com o menor número de saltos.

Uma variação proposta consiste em encontrar a jornada com menor tempo de entrega, ou seja, a jornada com o menor t_n obtido no último salto da jornada $((\cdot, v), t_n)$. Isso pode ser obtido com:

$$\min_t(j_{(u,v)}) = \arg_{j_{u,v}} \min\{t_n : ((\cdot, v), t_n) \in J_{(u,v)}\} \quad (3.2)$$

Outros critérios podem ainda ser definidos, sendo as vezes necessário adicionar informações nos saltos para isso.

3.3.2 Modelo de jornadas preditas

O preditor de contatos proposto no capítulo seguinte não diferencia a qual jornada o contato predito pertence. Uma vez que várias jornadas podem compartilhar partes de outras jornadas, basta que os contatos obtidos permitam estabelecer caminhos que possam ser seguidos por um algoritmo a fim de encontrar a jornada de menor custo. O preditor de jornadas deve garantir esta possibilidade.

O preditor de jornadas recebe como parâmetros: o nó de referência (n) de onde parte a mensagem; o nó em contato (n_{last}); e o momento desse contato (t_{last}). Ele retorna o conjunto de jornadas preditas $\hat{J}_{(n,\cdot)}$ com contatos no intervalo $[\tau_1, \tau_2]$.

Dessa forma, considerando que um nó de referência a está em contato com um nó b no instante t e que haja um preditor de jornadas que use um preditor de contatos global (Equação 4.3) ou local (Equações 4.4 e 4.5). O preditor de jornadas estabelece um conjunto de jornadas de interesse a partir de contatos (saltos) dentro do intervalo $[\tau_1, \tau_2]$ iniciando em a e terminando em qualquer nó, com $t \in [\tau_1, \tau_2]$. Ele pode ser definido como:

$$\begin{aligned} \hat{J}_{(a,\cdot)} = \{ & (P_{H_{[\tau_1, \tau_2]}}(a, n_{last} = b, t_{last}) = (n_{next}, t_{next}), \\ & (P_{H_{[\tau_1, \tau_2]}}(n_{next}, a, t_{next}) = \dots), (\dots = (n_{next_{n-1}}, t_{next_{n-1}}), \\ & (P_{H_{[\tau_1, \tau_2]}}(n_{next_{n-1}}, n_{next_{n-2}}, t_{next_{n-1}}), \dots) \} \quad (3.3) \end{aligned}$$

De forma similar, um outro conjunto de jornadas previstas de interesse é formado por saltos dentro do intervalo $[\tau_1, \tau_2]$ iniciando em b e terminando em qualquer nó, com $t \in [\tau_1, \tau_2]$. Ele pode ser definido como:

$$\begin{aligned} \hat{J}_{(b,\cdot)} = \{ & (P_{H_{[\tau_1, \tau_2]}}(b, n_{last} = a, t_{last}) = (n_{next}, t_{next}), \\ & (P_{H_{[\tau_1, \tau_2]}}(n_{next}, b, t_{next}) = \dots), (\dots = (n_{next_{n-1}}, t_{next_{n-1}}), \\ & (P_{H_{[\tau_1, \tau_2]}}(n_{next_{n-1}}, n_{next_{n-2}}, t_{next_{n-1}}), \dots) \} \quad (3.4) \end{aligned}$$

A partir de $\hat{J}_{(a,\cdot)}$ e $\hat{J}_{(b,\cdot)}$, para cada mensagem na fila de a é procurada uma jornada de menor atraso até seu destino c , conforme a Equação 3.2, através de um algoritmo de menor custo. Se $min_t(j_{(a,c)}) \in \hat{J}_{(b,\cdot)}$ então a mensagem é transferida para o nó b , caso contrário permanece na fila de a .

É possível não existir jornadas previstas a partir de a ou b ou ambos, então as mensagens permanecerão na fila.

Os valores de τ_1 e τ_2 podem indicar intervalos consecutivos, desde que seja previsto um algoritmo que controle a chamada ao preditor de contatos usando a RNA treinada para trabalhar em intervalos subsequentes. Tal estratégia de divisão em intervalo de tempos pode ser necessária para diminuir a complexidade das RNAs em sistemas muito grandes. Neste trabalho será utilizado um único intervalo de tempo para as predições.

É importante que o construtor de jornadas desconsidere os contatos previstos sem chances de acontecer, validando-os antes de juntá-los ao conjunto de saltos das jornadas. Isso melhora a escolha pelo melhores nós de contato e ajuda a diminuir o tempo de execução do algoritmo.

3.3.3 Determinação de melhor jornada mediante um critério predefinido

O problema da busca por uma jornada mais curta de u até v em termos do número de saltos, $\min.s(j_{(u,v)})$, conforme definido na Equação 3.1, consiste em encontrar uma jornada específica que seja composta pelo menor número de saltos. Isso pode ser feito através de um método que obtenha o primeiro salto de cada jornada, ou seja, o salto $((u, \cdot), t_0)$ com $(u, \cdot) \in E(t_0)$ na sua jornada, siga e conte os saltos até o salto $((\cdot, v), t_n)$ com $(\cdot, v) \in E(t_n)$.

Uma variação do problema, é a busca pela jornada com menor tempo de entrega, $\min.t(j_{(u,v)})$. Isso pode ser obtido de duas formas: uma através de um algoritmo de menor custo (p.e. Dijkstra) para buscar o caminho de menor custo; e outra extraindo diretamente de $\hat{J}_{(u,v)}$ a jornada que apresenta o menor tempo no vértice que liga v . Essa última forma é mais rápida em termos de tempo de processamento do que a primeira. Conforme a Equação 3.2, a jornada com o menor t_n obtido no último salto $((\cdot, v), t_n)$ de cada jornada pode ser encontrada com um método que selecione os saltos terminando no nó $(e_n(t_n), v)$ e escolha a jornada que apresenta $e_n(t_n)$ com o menor t_n .

Contudo, pode não existir jornadas levando ao destino, obtendo-se assim como resultado um conjunto vazio.

Dependendo dos atributos que forem adicionadas aos saltos, outras métricas podem ser usadas para a seleção de jornadas, tais como o melhor tempo de entrega até a expiração das mensagens, o menor consumo de energia e a maior largura de banda.

3.3.4 Um exemplo de predição de jornadas

A predição de jornadas é um mecanismo que deve ajudar a decidir qual é o melhor encaminhamento para uma mensagem chegar ao seu destinatário. Um exemplo de como o preditor de jornadas pode auxiliar nessa tarefa é apresentado para ajudar a entender seus princípios de funcionamento.

Para cada destino, é comparada a jornada predita partindo-se do nó de referência com a obtida partindo-se do nó contatado. Aquele nó que apresentar condições mais favoráveis de encaminhamento, de acordo com um determinado critério, será escolhido para encaminhar a mensagem. Neste trabalho o critério escolhido é do menor atraso.

Sejam dois nós, a e b , que entram em contato no instante t_0 ,

sendo que o nó a tem uma mensagem destinada ao nó c . O nó a executa o preditor de jornadas e obtém as jornadas previstas a partir dele no instante t_0 e também a partir do nó b no mesmo instante. A decisão de encaminhamento se dá avaliando as jornadas previstas a partir do nó a e do nó b de acordo com o critério escolhido. Neste exemplo, segundo o modelo de jornadas definido na seção 3.3.2, as jornadas previstas pelo nó a a partir dele próprio no instante t_0 até o nó c são as seguintes:

$$\hat{J}_{(a,c)_1} = \{((a, e), t_0), ((e, d), t_3), ((d, c), t_4)\}$$

$$\hat{J}_{(a,c)_2} = \{((a, b), t_1), ((b, c), t_6)\}$$

A partir do nó b , as jornadas previstas no instante t_0 até o nó c , são as seguintes:

$$\hat{J}_{(b,c)_1} = \{((b, d), t_2), ((d, c), t_5)\}$$

$$\hat{J}_{(b,c)_2} = \{((b, c), t_7)\}$$

Considerando que $t_0 < t_1 < \dots < t_7$, a jornada $\hat{J}_{(a,c)_1}$ a partir do nó b apresenta menor atraso de entrega (t_4). A mensagem é então mantida no nó a .

Entretanto, se a métrica fosse diminuir o número de saltos, a jornada $\hat{J}_{(b,c)_2}$ (um salto) seria escolhida. Neste caso, a mensagem seria transferida para b .

Quando não existirem jornadas previstas no nó a a partir do nó b para o destino ou existir jornada melhor a partir do nó a , a mensagem é mantida na fila.

3.4 PROPOSTA DE UM PREDITOR DE JORNADAS

O preditor de jornadas proposto é baseado em um algoritmo construtor de jornadas a partir de contatos preditos que organiza a sequência de saltos e realiza a minimização de um critério de menor custo para seus destinos. Essas duas partes e a invocação do preditor de jornadas são descritas a seguir.

3.4.1 Invocação do preditor de jornadas

Quando um nó contata outro, ele deve tomar uma decisão de encaminhamento para cada destinatário diferente em cada mensagem

do sua fila. Neste instante (t), o nó que mantém a mensagem (n) deve verificar as possibilidades que a mensagem tem de chegar ao seu destino pelo nó em contato (n_{last}) e comparar com as suas possibilidades de encaminhamento da mensagem. Essas verificações devem também considerar um critério escolhido, sendo que neste trabalho é o critério de menor atraso de entrega. Isso é feito no processamento da fila de mensagens, conforme apresentado no algoritmo 1.

Algoritmo 1: MESSAGEBUFFERPROCESSING

Input: n, n_{last}, t

- 1 *directDelivery*(n, n_{last}, t);
- 2 **if** $n.messageQueue \in \emptyset$ **then return**;
- 3 $n.costsTo \leftarrow highestCosts()$;
- 4 $n_{last}.costsTo \leftarrow highestCosts()$;
- 5 $\hat{G}_n \leftarrow BuildPredictedJourneys(depth, n, n_{last}, t)$;
- 6 $\hat{G}_{n_{last}} \leftarrow BuildPredictedJourneys(depth, n_{last}, n, t)$;
- 7 $n.costsTo \leftarrow minimizeDelay(\hat{G}_n)$;
- 8 $n_{last}.costsTo \leftarrow minimizeDelay(\hat{G}_{n_{last}})$;
- 9 **for each** m **from** $n.messageQueue$ **do**
- 10 $MessageForwarding(m, n, n_{last})$;
- 11 **return**;

A primeira tarefa executada por este algoritmo é a chamada do procedimento *directDelivery*(\cdot) para entregar as mensagens destinadas ao nó contatado (linha 1). Após isso, se ainda existem mensagens na fila a serem tratadas (linha 2), o algoritmo de roteamento dinâmico deve ser executado para atualizar informações sobre os destinatários.

Os vetores de custos para os destinos de n ($n.costsTo$) e de n_{last} ($n_{last}.costsTo$) são inicializados com o maior valor permitido (linhas 3-4). O procedimento *BuildPredictedJourneys*(\cdot) (Algoritmo 2 apresentado a seguir) é executado duas vezes: uma vez considerando n como o nó de referência (linha 5), resultando o grafo \hat{G}_n ; e outra vez considerando n_{last} e resultando $\hat{G}_{n_{last}}$ (linha 6). Para cada um desses grafos a função *minimizeDelay*(\cdot) atualiza o vetor de custos baseado no menor atraso (linhas 7-8). A partir de então, para cada mensagem da fila de mensagens (linha 9) a decisão de transferência da mensagem deve ser tomada pelo procedimento *MessageForwarding*(\cdot) (Algoritmo 3 apresentado posteriormente) (linha 10). A fila de mensagem deve ser processada até seu fim ou a ocorrência de um evento, tal como o de fim de conexão, interrompendo o processamento da fila.

3.4.2 O algoritmo do construtor de jornadas

Considerando que um contato tenha ocorrido entre os nós a e b no instante t_{last} dentro do intervalo de tempo $[\tau_1, \tau_2]$, o preditor de jornadas proposto inicia com um procedimento de construção de jornadas que recebe como entradas o nó de referência n , que pode ser a ou b , e t_{last} , e retorna um grafo G que representa o conjunto de jornadas preditas $\hat{J}_{(n, \cdot)}$.

O grafo G é construído pelo Algoritmo 2, *BuildPredictedJourneys*, que é inspirado no algoritmo de busca em profundidade e é apresentado a seguir. $G = (\hat{V}, \hat{E})$ é o grafo de jornadas preditas, onde $\hat{V} \in V$ é o conjunto de nós das jornadas preditas e $\hat{E}(t) = \{(u, v) : u, v \in \hat{V}, t \in \mathbb{R}\}$ é o conjunto de arestas que ligam os nós (u, v) no instante t .

Algoritmo 2: BuildPredictedJourneys

```

Input:  $depth, n, n_{last}, t_{last}$ 
Output:  $G = (\hat{V}, \hat{E})$ 
1 if  $n \notin \hat{V}$  then  $\hat{V} \leftarrow \hat{V} \cup \{n\}$ ;
2 while  $t_{last} < (maxTime - timeSlice)$  do
3    $(n_{next}, t_{next}) \leftarrow ANNpredictContact(n, n_{last}, t_{last})$ ;
4   if  $isValidOutput(n, n_{last}, t_{last}, n_{next}, t_{next})$  then
5     if  $n_{last} \neq n_{next}$  then
6       if  $depth > 1$  then
7          $(V', E') \leftarrow BuildPredictedJourneys$ 
8            $((depth - 1), n_{next}, n, t_{next})$ ;
9         if  $V' \neq \emptyset$  then
10           $\hat{V} \leftarrow \hat{V} \cup V'$ ;
11           $\hat{E} \leftarrow \hat{E} \cup E'$ ;
12       else
13          $t_{last} \leftarrow t_{next}$ ;
14    $t_{last} \leftarrow t_{last} + timeSlice$ ;
15 return  $G = (\hat{V}, \hat{E})$ ;

```

O parâmetro $depth$ controla a profundidade da pesquisa e $timeSlice$, o intervalo de tempo para a próxima busca por contatos. $maxTime$ é o tempo máximo permitido para busca. A função $ANNpredictContact(\cdot)$ implementa o preditor de contato baseado em RNA global

ou local que será apresentado no capítulo seguinte.

Os parâmetros de entrada desse algoritmo são: o nó de referência n ; o último nó contatado n_{last} ; o momento de contato t_{last} ; a profundidade máxima $depth$; e o tempo de intercontato $timeSlice$.

Este algoritmo deve realizar o controle de três situações na busca por nós preditos:

- controle da passagem do tempo: no laço principal, o controle da passagem do tempo é previsto iniciando no instante do último contato informado na entrada até $maxTime$ (linha 2). Toda vez que uma busca por contatos preditos no próximo nível de profundidade não retorna mais contatos, assume-se que o laço principal desloca-se no tempo de um $timeSlice$ (linha 13).
- controle de nós preditos válidos: quando se obtém um nó predito é importante validá-lo para saber se é um contato com chances de ocorrer. A função $isValidOutput(\cdot)$ (linha 4) faz essa verificação baseado em informações de faixas de valores e outras heurísticas que houverem. Nessa versão do algoritmo não é permitido voltar ao nó imediatamente anterior (linha 5). Controles para evitar a volta para nós já relacionados no grafo (laços) podem ser interessantes, mas devem ser especificados com cuidado para evitar a perda de contatos preditos com chances de ocorrer.
- controle da busca em profundidade: cada vez que se encontra um contato predito válido realiza-se uma busca por outros contatos a partir dele (em profundidade) até o limite definido por $depth$ (linha 6). Isso é feito através da chamada recursiva ao próprio algoritmo (linha 7), parametrizado com o nó de referência sendo o próximo nó predito e o momento do último contato sendo o momento predito do próximo contato.

O tempo de execução do Algoritmo 2 depende da: i) faixa de tempo para realizar a predição; ii) do tamanho da fatia de tempo para predições; iii) da profundidade da busca; e iv) do tempo de computação da RNA. Por sua vez, a computação da RNA depende da codificação dos valores de entrada e do somatório dos valores de entrada multiplicados pelos pesos e limitados de acordo com as respectivas funções de ativação de cada neurônio em cada camada até a camada de saída. O tempo de execução será determinado por $(maxDepth \times Computacao_da_RNA \times maxTime/timeSlice)$, onde:

- O maior tempo de execução acontece no caso em que $depth$ assume seu valor máximo (profundidade máxima), o tempo atual

está no início do ciclo (instante 0) e o *timeSlice* assume um valor mínimo (p.e. o menor tempo de intercontato do conjunto de treinamento).

- O menor tempo de execução ocorre no caso em que somente é realizada uma predição de contatos (*Computacao_da_RNA*) e o algoritmo não é executado em profundidade e o instante do próximo contato predito é próximo de *maxTime*, fazendo com que o algoritmo termine sua execução.

Uma simplificação útil desse grafo pode ser obtida se for necessário apenas representar o instante em que o contato foi estabelecido no conjunto de arestas. Neste caso o conjunto de arestas pode ser representado como $\hat{E} = \{(u, v), t) : u, v \in \hat{V}, t \in \mathbb{R}\}$. Assim, considerando $\hat{V} = \hat{V}(t)$, o grafo G é um multigrafo direcionado no qual é possível realizar uma busca pelo caminho de menor custo com algoritmos conhecidos.

3.4.3 Critério para jornadas de menor custo

Neste trabalho foi escolhido o critério de menor atraso para o destino, $\min.t(\hat{j}_{(u,v)})$, como critério de custo. Para determiná-lo pode ser aplicado uma variação do algoritmo de Dijkstra (MISA; FRANA, 2010) sobre o grafo G . Isso é implementado pela função *minimizeDelay*(\cdot) (invocada pelo Algoritmo 1 apresentado anteriormente).

Assumindo-se o instante de tempo do contato dentro do dia como peso das arestas, o algoritmo de Dijkstra não precisa somar o custo até o destino. As arestas para os destinos com o menor tempo absoluto podem ser diretamente obtidas do conjunto \hat{E} para determinar o menor custo ao destino e diminuir assim o processamento do algoritmo.

Dependendo das informações associadas as arestas, outras métricas podem ser usadas para a seleção de jornadas, tais como o menor número de saltos, o melhor tempo ajustado até a expiração da mensagem, o menor consumo de energia e a maior banda disponível.

O critério escolhido de menor atraso na entrega não significa necessariamente o aumento do número de mensagens entregues, pois ao privilegiar filas de mensagens e banda de rede para as mensagens chegarem em menor tempo, outras mensagens que poderiam ser entregues poderão ter que aguardar e perder a oportunidade de entrega. Entretanto, deve-se considerar que se uma mensagem chega mais rápido ao seu destino, ela deixa de consumir filas de mensagens e banda de

rede para ser encaminhada, favorecendo assim a entrega de mensagens. Neste sentido, quando as filas forem limitadas, a minimização do critério deve ser prevista na estratégia de gerenciamento das filas, além de prevista na estratégia de roteamento.

3.4.4 O ajuste da fatia de tempo na construção de jornadas: *timeSlice*

A variação do valor utilizado para a fatia de tempo *timeSlice* influencia no número de mensagens entregues e também no tempo de execução como se viu anteriormente. Para ajustar o *timeSlice* deve-se levar em consideração que:

- um valor pequeno de *timeSlice* implica mais interações na busca por contatos preditos; contudo, um valor muito pequeno favorecerá a repetição de predições que não são úteis na construção da jornada, aumentando o tempo de execução do algoritmo;
- um valor grande de *timeSlice* implica a perda de contatos preditos que podem ser úteis no estabelecimento das jornadas preditas, contudo permite obter um menor tempo de execução;
- um valor qualquer no *timeSlice* não favorece o acerto na predição de contatos, por fazer a RNA ser consultada com parâmetro de entrada de tempo atual desconhecido ou pouco conhecido no histórico de contatos, trazendo como consequência predições menos confiáveis.

Assim, a determinação do valor da fatia de tempo *timeSlice* é realizada da seguinte forma:

1. Obtém-se os valores de intercontato do histórico de contato (sem considerar o dia a ser testado);
2. Excluem-se os valores extremos que estão distantes das amostras consecutivas. Considera-se, por exemplo, até a terceira amostra que apresente uma diferença maior que $\frac{1}{6}$ para a próxima amostra no início e também no final.
3. Calcula-se a metade do desvio padrão e atribui-se ao *timeSlice* para favorecer a taxa de entrega; utiliza-se somente o valor do desvio padrão se o consumo de processador deve ser diminuído.

Na seção seguinte a estratégia de roteamento proposta é modificada com a adoção de uma estratégia de replicação de mensagens.

3.5 ESTRATÉGIA DE ROTEAMENTO COM REPLICAÇÃO DE MENSAGENS

Roteamento com replicação

O número de cópias das mensagens favorece o aumento do número de mensagens entregues, contudo aumenta consideravelmente o consumo de banda de rede e as filas de mensagens. O controle da replicação deve ser adotado para permitir aumentar a taxa de entrega e não comprometer os recursos da rede.

Uma abordagem de encaminhamento mais eficiente quanto ao consumo de recursos não faz cópia de mensagens. Entretanto, a não ser que exista grande conhecimento prévio sobre a ocorrência dos contatos ou uma rede com poucos recursos, essa abordagem não é geralmente suficiente para funcionar adequadamente as aplicações na DTN.

O roteamento nas redes DTN é feito através de um "encaminhador", sem cópia de mensagem ou através de um "replicador", no qual a mensagem pode ser copiada, normalmente adotando-se algum controle para evitar sobrecarga na utilização de recursos. Em alguns trabalhos, a própria expiração da mensagem e os limites das filas de mensagens são usadas para limitar a replicação. No caso da estratégia de roteamento apresentada neste trabalho, esse controle baseia-se no número máximo de cópias da mensagem na rede, tal como na estratégia de *Binary Spray and Wait* (BSW) (SPYROPOULOS; PSOUNIS; RAGHAVENDRA, 2005).

Uma abordagem com um "replicador" também deve ter junto um "encaminhador", a ser utilizado apenas quando não for possível realizar a replicar a mensagem.

Nesta versão da estratégia proposta, o emprego do replicador ou do encaminhador é definido a partir da avaliação da jornada predita pelo nó juntamente com o número de cópias permitido. Existem as seguintes condições a serem avaliadas para cada mensagem da fila:

- se o limite de cópia não foi atingido:
 - se o custo para o destino favorece o encaminhamento pelo nó em contato, a mensagem é replicada;
 - se o custo para o destino não favorece o encaminhamento pelo nó em contato, mas existe registro histórico de contato deste nó com o nó de destino da mensagem, a mensagem é replicada;

- se o limite de cópia foi atingido:
 - se o custo para o destino favorece o encaminhamento pelo nó em contato, a mensagem é encaminhada (sem cópia);
 - se o custo para o destino não favorece o encaminhamento pelo nó em contato, a mensagem é mantida na fila;

O algoritmo de encaminhamento e controle de cópias

As condições para cópia da mensagem são implementadas no procedimento *MessageForwarding*(\cdot) (Algoritmo 3). Esse procedimento é invocado conforme apresentado anteriormente no Algoritmo 1, o qual realiza primeiramente a chamada dos procedimentos do preditor de jornadas. Após a predição de jornadas, os vetores de custo atualizados serão utilizados pelo procedimento *MessageForwarding* que realizará o controle de cópia e o encaminhamento das mensagens.

O controle do número de cópias é realizado definindo-se primeiramente um número máximo de cópia *maxCopy* permitido para uma mesma mensagem na rede. A cada contato que for julgado ter melhores chances de se encaminhar a mensagem, a estratégia de controle do número de cópias dividirá o número de cópias com o nó em contato. *maxCopy* é então dividido por dois e configurado como novo limite máximo de cópias da mensagem copiada para o nó em contato. Em caso de divisão com resto, o nó de referência fica com uma mensagem a mais que o nó que recebe a cópia.

Algoritmo 3: MessageForwarding

```

Input:  $m, n, n_{last}$ 
1 if  $m.maxCopy > 1$  then
2   if  $m.destination \in n_{last}.listOfNodesContacted$  then
3      $newMaxCopy \leftarrow \lceil \frac{m.maxCopy}{2} \rceil$ ;
4      $messageCopy(m, n_{last}, maxCopy - newMaxCopy)$ ;
5      $m.maxCopy \leftarrow newMaxCopy$ ;
6 else
7   if  $(n_{last}.costsTo[m.destination] < n.costsTo[m.destination])$ 
   then
8      $messageTransfer(m, n_{last})$ ;
9 return;

```

Se $m.maxCopy$ é suficiente para copiar a mensagem (linha 1) e o nó de destino está na lista dos nós contatados do nó em contato (n_{last}) (linha 2), a cópia da mensagem é realizada (linhas 3-5). Caso contrário, o encaminhamento sem cópia é realizado.

A mensagem é transferida para o nó contatado n_{last} somente se o custo da jornada a partir de n_{last} para o destinatário da mensagem é menor do que o obtido pelo nó n (linhas 7-8). Se o custo for maior ou igual e não for possível replicar, a mensagem permanece no nó atual n .

3.6 CONCLUSÃO

Neste capítulo foi apresentada uma proposta para predição de jornadas que aplica sucessivamente um preditor de contatos para cenários com contatos quase-opportunistas.

Foram apresentados os princípios do preditor de jornadas e uma modelagem de redes DTN a partir de grafos variantes no tempo (TVG) e os principais algoritmos da estratégia de roteamento proposta.

O algoritmo de construção de jornadas permite obter as jornadas estabelecidas sobre contatos preditos.

A busca pela melhor jornada pode ser feita por um algoritmo de caminho de menor custo ou, no caso do número de saltos ou menor atraso de entrega, obtidas diretamente na construção das jornadas.

O algoritmo de processamento da fila de mensagens permite encaminhar as mensagens quando um nó é contatado. O algoritmo de encaminhamento de mensagens permite controlar o número de cópias da mensagem e o uso de um "encaminhador" ou de um "replicador" de mensagens quando possível.

Detalhes da proposta do preditor de contatos foram apresentados considerando a arquitetura DTN do *Internet Research Task Force* (IRTF).

O preditor de contatos que é invocado a partir do construtor de jornadas é descrito no capítulo que seque.

4 UM PREDITOR DE CONTATOS BASEADO EM RNA

Neste capítulo é apresentada uma proposta de preditor de contatos entre nós de uma DTN formada por ônibus do STPU. Este preditor é parte da estratégia proposta no capítulo anterior. Primeiramente é apresentada a função de predição de contatos almejada, sendo então apresentado um procedimento para obter a função de predição de contatos baseado em RNAs. Uma aplicação desse procedimento e as atividades relacionadas são apresentadas e na sequência as conclusões.

4.1 INTRODUÇÃO

A estratégia de roteamento proposta leva em consideração que a decisão de encaminhamento da mensagem seja baseada em jornadas previstas. Neste trabalho, essas jornadas previstas deverão ser definidas através do uso de um preditor de contatos. Por causa das características dos contatos oportunistas com algum tipo de regularidade (caso do STPU) e devido à capacidade das RNAs para modelar não linearidades, julgou-se interessante usá-las para prever a ocorrência de contatos. Uma revisão sobre as RNAs é apresentada no Apêndice A.

A definição da RNA adequada para a realização de predições é uma tarefa que exige muita análise e testes. A RNA é sensível à forma de representação dos valores de entrada e da apresentação desses valores no processo de aprendizagem. Para guiar a definição das RNAs desse trabalho foi desenvolvido um procedimento que é iniciado a partir da definição das entradas e saídas de uma função de predição.

4.2 FUNÇÃO DE PREDIÇÃO DE CONTATOS

Um pressuposto inicial para propor uma função de predição de contatos é que no contexto de movimentação repetitiva e com itinerários pré-definidos existem padrões de contatos entre os nós na repetição da movimentação. Utilizando o contexto dos ônibus de um transporte público urbano, pode-se pensar que existe uma grande regularidade nos contatos entre os ônibus, uma vez que seguem itinerários fixos com horários previamente estabelecidos.

Entretanto, mesmo que a movimentação dos ônibus do STPU esteja organizada por itinerários e tabelas de horários, mínimas va-

riações de velocidade e tempos de permanência nos pontos de parada tem uma influência muito grande na regularidade do estabelecimento dos contatos. Por exemplo, 5 segundos a mais ou a menos na espera pelo embarque de passageiros ou uma variação na velocidade de alguns quilômetros por hora, pode mudar substancialmente as oportunidades de contato. Isso torna necessário um estudo preliminar para avaliar se o cenário pode ser considerado totalmente oportunista ou se há algum tipo de regularidade que pode ser aproveitada. A partir disso é escolhido o método a ser utilizado para basear a função de predição de contatos.

Foram observadas formas básicas de movimentação dos ônibus no STPU que interferem na realização de contatos (DOERING, 2012). As condições que influenciam essas movimentações são o tráfego, a velocidade dos nós, os elementos controladores de tráfego (p.e. semáforos), as falhas de comunicação, as falhas nos nós e o tempo de permanência nas paradas de ônibus. Devido à imprevisibilidade dessas condições e dificuldade da identificação prática da sua ocorrência, torna-se difícil estabelecer uma relação de probabilidade de ocorrência desses movimentos e generalização para outros cenários. Diante disso, foi levantada a hipótese de obter um modelo utilizando o histórico de contatos de longo prazo e métodos de RNAs.

4.2.1 Definição da Função de Predição de Contatos

No contexto do STPU existem repetições na movimentação dos nós nas seguintes situações: (i) na execução de uma rota, podendo ser repetida algumas vezes no dia; (ii) diariamente, na execução da linha de ônibus. Para o preditor de contatos proposto é necessário ter o registro dos contatos entre dispositivos de comunicação instalados nos ônibus na forma de um histórico de contatos.

Identificando os ciclos de movimentação diária, o histórico de contatos pode ser definido como $H = \{H_1, H_2, \dots, H_l\}$, onde H_i representa os contatos do histórico H realizados no dia ou ciclo i , sendo l o último ciclo de contatos registrado. Cada contato de H_i é representado, com no mínimo, as seguintes informações básicas de contato:

- a : nó de referência;
- b : nó contatado; e
- t : momento de início do contato;

- t_{fim} : momento de fim do contato;

onde $a, b \in V$, o conjunto de nós da rede.

Quando H deve ser definido em relação a um determinado intervalo de tempo $[\tau_1, \tau_2]$ dentro do ciclo, se tem que $H_{[\tau_1, \tau_2]} = \{H_{1_{[\tau_1, \tau_2]}}, H_{2_{[\tau_1, \tau_2]}}, \dots, H_{l_{[\tau_1, \tau_2]}}\}$ onde $\tau_1 \leq t \leq \tau_2$, representando assim somente os contatos ocorridos neste intervalo de tempo.

Com essas informações, a função de predição de contato proposta $P_{H_{[\tau_1, \tau_2]}}$ deve ser projetada para operar com os dados históricos H sobre um intervalo de tempo definido $[\tau_1, \tau_2]$ recebendo como parâmetro o nó de referência $n = a$, o nó de contato anterior $n_{last} = b$ e o momento do contato anterior $t_{last} = t$ e então determinar n_{next} e t_{next} :

$$P_{H_{[\tau_1, \tau_2]}}(n, n_{last}, t_{last}) = (n_{next}, t_{next}) \quad (4.1)$$

onde $n_{next} \in V$ e $\tau_1 \leq t_{next} \leq \tau_2$.

Para que a função de predição forneça resultados satisfatórios é necessário modelar as relações entre as informações do histórico. Neste sentido, o processo de aprendizagem das RNAs descobre relações importantes por meio dos exemplos aprendidos (contendo os contatos anteriores) e permite que estas sejam modeladas na função de predição. Entretanto, frequentemente é necessário organizar as informações para facilitar a operação do método para tirar proveito dessas relações.

As informações que devem favorecer o relacionamento entre dados do histórico em dias diferentes devem ser obtidas a partir das informações básicas de contato. Por exemplo, para a proposta desse trabalho é determinado o nó n_{last} (b) como contato anterior do nó n (a) e b no próximo registro dos contatos de n é n_{next} e respectivo instante de contato t_{next} . Dessa forma, o conjunto de informações do histórico para cada contato é ampliada, ficando a seguinte tupla: $(n, n_{last}, t_{last}, n_{next}, t_{next})$.

É possível explorar um maior relacionamento entre os dados do que o obtido com dois contatos consecutivos, isto é, obtendo $n_{last_{last}}$, n_{last} , n_{next} . Entretanto, deve-se primeiramente avaliar se essa relação realmente existe para os contatos do cenário em questão.

Uma vez que as relações foram modeladas adequadamente, os contatos poderão ser preditos e usados para determinar os contatos estabelecidos nas jornadas preditas.

Para iniciar a construção do preditor de contatos baseado em RNA é necessário estudar e tratar as informações de histórico a serem utilizadas. Assim, a aplicação de uma série de análises e tratamentos dos dados é necessária a fim de se encontrar dados que apresentem

relações úteis para a definição de uma arquitetura de RNA que favoreça as predições desejadas.

4.2.2 Operação da Função de Predição de Contatos baseada em RNA

Para explicar a operação da função de predição de contatos, um exemplo é apresentado com informações de contato de um histórico de dois dias registrados a partir de um nó.

A função de predição de contatos deve relacionar uma sequência de contatos anteriores (horizonte de predição), com um tamanho que permita obter regularidade na predição do próximo contato em dias diferentes. Ao relacionar a sequência de dois contatos com um nó de referência, a camada de entrada será preparada para considerar a seguinte sequência de contatos: nó atual(referência), nó anterior, próximo nó.

Por exemplo, uma sequência de contatos S com o nó a ocorrida em dois dias e extraída do histórico de contatos poderia ser:

$$S_{a_{04/04/2014}} = ((g, 134), (e, 256), (b, 451), (d, 701), (f, 898))$$

$$S_{a_{05/04/2014}} = ((g, 213), (b, 330), (e, 342), (c, 680))$$

Onde b, c, d, e, f e g são nós e cada par (n, t) dos conjuntos $S_{a_{dia}}$ representa o nó e o instante de contato. Com essa sequência, podem ser extraídas várias informações para o conjunto de treinamento. Baseado nas informações dos conjuntos $S_{a_{dia}}$, para as informações de entrada da RNA (nó atual, último nó de contato e instante do último contato), deve existir uma saída (próximo nó de contato e instante do próximo contato). Baseado nessa sequência, alguns dos seguintes dados de treinamento (exemplos de treinamento) extraídos seriam:

- para a entrada $(a, g, 134)$ a saída é $(e, 256)$
- para a entrada $(a, e, 256)$ a saída é $(b, 451)$
- para a entrada $(a, b, 451)$ a saída é $(d, 701)$
- para a entrada $(a, d, 701)$ a saída é $(f, 898)$
- para a entrada $(a, g, 213)$ a saída é $(b, 330)$
- para a entrada $(a, b, 330)$ a saída é $(e, 342)$

- para a entrada $(a, e, 342)$ a saída é $(c, 680)$

Com uma RNA projetada e treinada para receber essas entradas e saídas, um exemplo da chamada da função de predição com o respectivo resultado pode ser o seguinte:

$$P_{H_{[50,1000]}}(a, b, 451) = (d, 698) \quad (4.2)$$

onde $[50, 1000]$ é o intervalo de tempo dos contatos extraídos do histórico para predição, a é o nó de referência, b é o nó de contato anterior, 451 é o momento de contato anterior, d é o próximo nó de contato predito e 698 é o instante predito do próximo contato. Neste caso, a diferença do valor de tempo treinado e o predito ($701 - 698$) é pequena e está dentro do intervalo para predição ($[50, 1000]$), tornando válido o contato predito.

A RNA deve ter a capacidade de generalização para fornecer predições adequadas aos valores de entrada que não foram treinados. Contudo, quando existe muita variação nos dados de treinamento, torna-se necessário obter diversos dados de sequência de contatos cobrindo a grande maioria das possibilidades de nós e instantes de contato. O treinamento deverá ser mais demorado, mas isso deve possibilitar que a RNA acompanhe as variações nos instantes de contato e nas sequências de nós que realizaram contato.

É importante entender que com o aumento do número de contatos a serem treinados aumenta-se também os contatos que podem ser interpretados como ruído, prejudicando as predições. Pode também acontecer a situação na qual as entradas tem valores muito próximos ou até mesmo iguais, mas devem predizer valores de saída muito diferentes, o que obviamente deve ser evitado.

Após a escolha das informações de entrada, pode-se iniciar a definição da codificação dessas informações para determinação do número de neurônios da camada de entrada da RNA. Além disso, outras decisões referentes a arquitetura da RNA devem ser informadas: o número de camadas intermediárias (ocultas) e respectivos neurônios; os neurônios que representam cada informação de saída; a função de ativação mais adequada para os neurônios; o método de treinamento; o algoritmo de aprendizagem; e os critérios parar a aprendizagem. Essas definições devem ser adequadas para resolver o problema a ser tratado e são dependentes das informações disponíveis.

Neste trabalho, devido à grande variação na ocorrência dos contatos, uma topologia de múltiplas camadas de neurônios deve ser usada, sendo que uma ou duas camadas intermediárias de neurônios na maioria

dos casos é suficientes para tratar problemas da classe não linearmente separáveis. Outro problema devido à essa grande variação é a realização adequada do treinamento da RNA para que seja possível obter boas predições.

Para organizar o processo de definição da RNA e para que elas funcionem do modo desejado foi definido um procedimento baseado em etapas com uma série de atividades. Na aplicação desse procedimento são propostas duas formas de organizar a topologia da RNA para estabelecer a função de predição da Equação 4.1: uma que deve tirar proveito de informações globais; e outra que utiliza apenas informações locais do nó consultado. Esse procedimento e essas topologias são apresentados na seção a seguir.

4.3 UM PROCEDIMENTO PARA OBTENÇÃO DE UM PREDITOR BASEADO EM RNA

Um procedimento inspirado no processo de mineração de dados (FAYYAD et al., 1996) foi criado para a implementação de RNAs objetivando a predição. O procedimento proposto envolve as etapas apresentadas na Figura 7.

Além das etapas, essa figura mostra a sua sequência de execução e as entradas e saídas envolvidas.

Na sequência, são apresentados os detalhes de cada etapa do procedimento apresentando as atividades envolvidas. Por sua vez, a aplicação desse procedimento no caso da proposta desse trabalho é apresentado na próxima subseção.

4.3.1 Etapa de análise de dados

A análise de dados é realizada na primeira etapa e é essencial para verificar a viabilidade de realizar a predição desejada com os dados disponíveis e de melhorar o desempenho da predição. Inicia com a formulação da predição, atividade em que é feita a escolha das informações de entrada e saída para atender o objetivo da predição. Depois é feita a obtenção de um conjunto de dados que contemple essas informações e objetivo definidos. Por fim, é realizada a obtenção das informações de entrada e saída para operacionalizar a função de predição.

Ao fim dessa etapa deve-se obter as informações para definição das RNAs, mais especificamente a definição das entradas e saídas, do

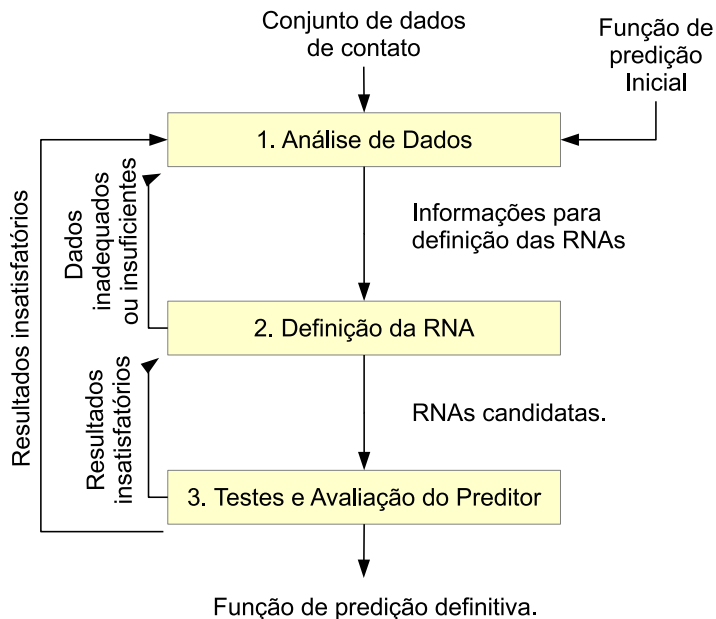


Figura 7: Etapas do procedimento utilizado para definição das RNAs do preditor de contatos proposto.

pré-processamento e filtros, dos exemplos de treinamento e das estatísticas para avaliações.

As atividades dessa etapa são detalhadas a seguir.

Formulação da predição

A primeira etapa é a definição do que se deseja prever. É necessário obter uma definição clara das informações a serem previstas (saídas) e das informações de entradas necessárias para se obter essas saídas, pois as outras etapas estarão sempre baseadas nessa definição. Essa definição influencia a escolha de um conjunto de dados na etapa seguinte.

Obtenção do conjunto de dados

Nesta atividade os conjuntos de dados disponíveis devem ser pesquisados para verificar se tem informações suficientes para alcançar o

objetivo da predição. Caso o conjunto de dados desejado não exista, deve-se considerar a possibilidade de implementar um projeto para obtenção dos dados reais ou de simulações para obtenção de dados sintéticos.

Obtenção das informações de entrada e saída

Essa atividade é realizada em conjunto com a atividade anterior. Ao analisar as informações mínimas de contato são verificadas possibilidades de extração de outras informações, agrupamentos e relacionamentos entre os dados.

As seguintes análises sobre o conjunto de dados são normalmente realizadas:

1. Verificação da frequência de valores nos atributos (variáveis): identificar valores que apresentam maiores frequência em determinadas condições, por exemplo, de acordo com uma faixa de tempo;
2. Verificação da regularidade das informações de entrada e objetivo da predição: determinar se existe ou não regularidade suficiente no conjunto de dados para estabelecer as informações de entrada e saída da função de predição;
3. Verificação da sequência entre atributos (variáveis) do conjunto de dados: identificar sequências que se repetem nos atributos.

Para apoiar essas análises pode-se utilizar grafos, matrizes de pontos, gráficos e programas específicos para isso. Os programas podem ser desenvolvidos para transformar os dados, agrupá-los, classificá-los, determinar o grau de influência do atributo no conjunto ou outro processamento que se julgar necessário.

Para favorecer bons resultados com o uso das RNAs, a relação estabelecida entre as entradas e saídas da RNA deve ser de uma ou mais entradas para uma saída, sendo três considerações importantes:

- a relação de entrada e saída não deverá ser de 1 para muitos, pois ao computar um mesmo conjunto de dados de entrada (estímulo) pela RNA, a saída deverá sempre ser igual. Dados de treinamento (exemplos) com entradas iguais e saídas diferentes não permitem alcançar um bom treinamento e boas predições. Nesse caso, é importante a RNA ter algum valor de entrada que permita diferenciar essas entradas;

- quando informada uma entrada treinada (conhecida), a RNA vai tentar aproximar os valores da saída com os valores desejados, contudo, normalmente não será igual por causa da interferência dos outros exemplos;
- se as entradas forem muito parecidas e tiverem valores de saída muito diferentes, os neurônios da RNA terão que conseguir modelar essa pequena variação da entrada em uma grande diferença na saída.

Essa etapa resulta na: (i) definição das informações de entrada para a função de predição e da confirmação ou modificação do objetivo da predição; (ii) definição do pré-processamento dos dados brutos para gerar os exemplos de treinamento a serem usados na RNA; (iii) obtenção de informações estatísticas sobre os dados a fim de auxiliar na operação do preditor de jornadas, na representação e transformações de dados de entrada da RNA e na avaliação de desempenho do preditor de contatos.

4.3.2 Etapa de definição da RNA

Na segunda etapa, recebe-se as informações das entradas e saídas da função de predição, determina-se a estrutura de RNA a ser utilizada e então, a partir disso cria-se a RNA.

A partir da codificação das entradas e dos exemplos para treinamento é necessário definir o número de camadas e ajustar o número de neurônios em cada camada. Isso é feito levando em consideração o desempenho no treinamento. O algoritmo de aprendizagem utilizado no treinamento e os critérios de parada são também definidos. Se o treinamento não apresentar um baixo erro de treinamento rapidamente na maioria das execuções do algoritmo de aprendizagem, deve-se voltar na primeira etapa para reanalisar os dados e redefinir a função de predição.

Ao final dessa etapa deve-se obter as RNAs candidatas, incluindo o algoritmo de aprendizagem e os critérios de parada de aprendizagem determinados.

Nesta etapa é prevista a realização das seguintes atividades: definição da topologia da RNA; separação dos exemplos para treinamento, validação e testes; e ajuste dos critérios de parada do treinamento. Detalhes são apresentados na sequência.

Definição da topologia da RNA

A partir dos dados pré-processados fornecendo exemplos para aprendizagem supervisionada das RNAs, pode-se então realizar a definição do tipo, topologia, neurônios da camada de entrada, camadas e neurônios intermediários e os neurônios da camada de saída.

Uma RNA necessita receber os dados de entrada adequados para melhorar a representação das informações que serão tratadas na RNA. Essa adequação também é realizada através da codificação dos valores de entrada. A codificação determinará a quantidade de neurônios da camada de entrada.

As codificações ocorridas nos dados de entrada serão as seguintes:

- definição da quantidade de neurônios por informação: o dado qualitativo ou que representa a classe ou identidade distinta, deve ter um neurônio para cada qualidade/identidade a ser assumida. O dado contínuo ou que representa uma escala numérica ordenada deve ser representados por um neurônio. Um problema da representação qualitativa é o aumento do número de neurônios. Uma forma de tratar esse problema é utilizar apenas um neurônio para representar as qualidades ordenadas em frequência;
- normalização e transformação de escala para 0 até 1 ou -1 até 1: garante que a grandeza tratada não seja modificada por causa da escala (esticada ou encolhida). A função de ativação do neurônio calcula a saída de acordo com sua definição, mas se o valor de entrada e a função de ativação funcionam em escalas diferentes, o limiar de ativação pode ocorrer em uma parte da escala que desfavoreça o trabalho da RNA.
- discretização: acontece quando os valores contínuos em uma escala são ajustados de acordo com valores que representam a faixa onde eles acontecem, favorecendo o trabalho da RNA. Uma desvantagem é a diminuição da capacidade de capturar detalhes (resolução).

Alguns cálculos baseados nas informações de entrada ajudam a definir o número de camadas intermediárias e neurônios. Métodos tais como o de poda também podem ser utilizados para melhorar os resultados e ajustar o desempenho.

Separação dos exemplos para treinamento, validação e testes

Ao tratar de testes de predição usando RNA, é normalmente importante isolar os exemplos que se deseja testar daqueles que se deseja utilizar no treinamento e na validação para que não haja interferência no resultado da predição. A predição será injusta se uma RNA é ensinada com os mesmos valores que deverão acontecer no futuro ou utilizando critérios de parada do treinamento baseados nesses valores. Apesar disso, alguns autores acreditam que quando a quantidade de dados é muito grande, essa informação futura se torna irrelevante diante de todas as outras informações do histórico. Para evitar favorecer as respostas da RNA, realizou-se uma separação dos exemplos usados para treinamento dos que seriam usados para testes.

Ajuste dos critérios de parada do treinamento

A etapa de treinamento depende da definição de critérios objetivos de parada do treinamento. Esta é uma etapa importante, pois um treinamento mal conduzido pode levar a RNA a fornecer predições frequentemente distantes das ideais, inviabilizando sua aplicação.

O treinamento da RNA deve ser suficiente para capturar características de padrões existentes nos dados, evitando-se as seguintes situações:

- o treinamento insuficiente, levando a RNA a ver padrões onde não existem;
- o treinamento excessivo, levando a RNA a não generalizar os dados de treinamento (exemplos de treinamento), respondendo apenas exatamente o que apareceu nos exemplos treinados;
- o excessivo consumo de tempo para treinamento, principalmente se a aplicação exige atualização frequente da RNA;
- o elevado erro de treinamento, indicando haver inconformidades da RNA modelada com os dados de treinamento;
- o elevado erro de validação ou testes, indicando haver inconformidades da RNA modelada e dos valores usados no treinamento para representar situações não conhecidas;

Essas situações são traduzidas em termos de critérios de parada do treinamento. Por exemplo, um critério de parada pode ser o número de ciclos (ou épocas) de treinamento. Contudo, o número de ciclos normalmente não será uma boa medida quando as entradas e saídas dos exemplos de treinamento apresentarem um relacionamento difícil de ser

representado. Isso normalmente acaba levando a adoção de um número alto de ciclos de treinamento e conseqüentemente a um treinamento excessivo.

Outro critério simples é que o treinamento continue até o erro de predição ser menor do que um determinado valor. Contudo, dependendo dos valores de entrada, o treinamento pode nunca atingir o critério de parada ou pode atingí-lo muito cedo, obtendo-se uma generalização grosseira, insuficiente para uma representação adequada do que deve ser aprendido. Torna-se necessário elaborar melhores critérios para treinar adequadamente a RNA.

Um critério bastante conhecido é quando o erro de predição e o erro de validação entram em regime permanente. Critérios de parada de treinamento mais elaborados levam em consideração a diminuição da taxa do erro de predição. A taxa de erro de predição indica a quantidade de aprendizado por ciclos de treinamento, ou seja, o quanto o erro de predição diminui com o passar dos ciclos de treinamento.

Algumas abordagens realizam múltiplos treinamentos dividindo os dados de treinamento em grupos e realizando a validação cruzando os grupos de exemplos.

4.3.3 Etapa de testes e avaliação do preditor

Nesta etapa, as RNAs já estão treinadas e prontas para realizar as predições. Deve-se então definir quais testes permitirão obter valores comparativos adequados para entender o funcionamento do método e suas limitações. Essas avaliações indicarão se as relações entre os dados foram modeladas pela RNA ou se é necessário voltar na segunda etapa para ajustes ou voltar para a primeira etapa e redefinir toda a RNA.

Na etapa de treinamento, normalmente os pesos são inicializados de forma aleatória e com distribuição uniforme. Isso fornece uma característica de aleatoriedade na resposta da RNA, pois a cada novo treinamento normalmente se tem pesos diferentes e resultados próximos, mas diferentes. Essa característica de aleatoriedade é perdida se os pesos forem iniciados sempre com os mesmos valores e o treinamento executado na mesma forma, contudo perde-se a oportunidade de encontrar melhores conjuntos de pesos, pois o processo de treinamento pode ficar preso em algum mínimo local.

A resposta da RNA também é caracterizada pela probabilidade, por causa da influência que exercem os dados de treinamento apresentados com maior frequência. Torna-se então necessário realizar testes

com mais de um treino, ou seja, é importante realizar testes com mais de um conjunto de pesos.

Na atividade de avaliação pode-se chegar a conclusão de que os dados ou a RNA criada não estão conseguindo apresentar resultados satisfatórios, exigindo voltar a etapas anteriores para se reanalisar e definir a topologia, a forma de aprendizagem e os critérios de parada do treinamento.

Pode-se também voltar a primeira etapa e avaliar que não será possível obter a predição desejada com os dados disponíveis, necessitando rever as informações de entrada, o objetivo da predição e o conjunto de dados utilizado.

Se os resultados foram satisfatórios nessa etapa, ao final do procedimento deve-se obter:

- a função de predição definitiva;
- a escolha definitiva do conjunto de dados a ser utilizado;
- a definição do pré-processamento;
- a definição das RNAs a serem utilizadas;
- a escolha do algoritmo de aprendizagem ; e
- a definição de critérios de parada do treinamento.

Na sequência é apresentada uma aplicação desse procedimento neste trabalho.

4.4 APLICAÇÃO DO PROCEDIMENTO PARA CONSTRUÇÃO DE UM PREDITOR DE CONTATOS

4.4.1 Etapa de análise de dados

Formulação da predição

Conforme definido na função de predição (Equação 4.1), o objetivo da predição é obter o próximo nó e o próximo instante de contato a partir de um nó de referência e do nó e instante de contato anterior.

Foi avaliado inicialmente que em um cenário com poucas informações, um mínimo de informações necessárias a serem disponibilizadas por um histórico de contatos seriam:

- o par de nós em contato;
- o dia e instante do dia que o contato iniciou;

Obtenção do conjunto de dados

A implementação de um projeto para obtenção de um conjunto de dados reais foi descartada dada a dificuldade de obtenção de equipamentos, trâmites com as empresas transporte público e tempo disponível para a realização do projeto. Dessa forma, foi optado por utilizar dados de projetos já realizados.

O conjunto de dados utilizado foi o do projeto DieselNet (detalhado no Apêndice B), no qual foram avaliados os contatos entre ônibus de um STPU com 40 ônibus em 10 rotas durante 55 dias. Esse conjunto de dados foi escolhido pois apresenta uma grande quantidade de dias de registro dos contatos com as informações básicas necessárias. No caso da predição de contato proposta é necessário ter informações de muitos dias para que a RNA consiga extrair padrões mais adequados de estabelecimento de contatos. Principalmente por isso e por desejar avaliar o comportamento com contatos reais, outros conjuntos de dados pesquisados não foram considerados adequados.

Obtenção das informações de entrada e saída

Diversos recursos, tais como grafos, gráficos, algoritmos e programas específicos foram utilizados para gerar informações para a análise de dados e obtenção de informações importantes para o preditor de contatos.

Neste trabalho foram utilizados as seguintes análises e recursos:

1. Verificação da frequência de nós em contato: identificar nós que apresentam maiores frequência de contatos no geral ou no mesmo dia da semana, além de identificar grupos de nós que fazem contatos com um determinado nó. Foram utilizados grafos diários e individuais para analisar isso, além de programas de ordenação, contadores e que geram estatísticas sobre o conjunto de dados;
2. Verificação de sequências de nós e instantes de contato: identificar a existência de sequências de nós fazendo contato com um determinado nó e intervalos de tempos regulares entre contatos ou de acordo com determinado intervalo do tempo. Foram uti-

lizadas matrizes de pontos, gráficos de contato e algoritmo de agrupamento;

3. Verificação da regularidade das informações de entrada e objetivo da predição: determinar se existe ou não regularidade suficiente no conjunto de dados para estabelecer que as informações do nó e seus contatos anterior e posterior a um determinado instante de contato podem ser utilizadas para estabelecer relações na função de predição. Foram utilizadas a matrizes de pontos e o algoritmo de agrupamento;

A verificação da frequência de nós em contato foi realizada através de grafos como os da Figura 8. Pode-se observar os nós que representam os ônibus do DieselNet e as linhas representando os contatos entre eles em quatro dias consecutivos. As linhas sólidas representam os contatos com duração maior que 100 segundos. Verifica-se, por exemplo, que o nó 3027, indicado pelas setas, teve uma frequência de contatos diferente em cada dia consecutivo. Além disso, os nós em contatos são substancialmente diferentes em dias diferentes. Essa mesma observação foi constatada avaliando-se o mesmo dia da semana. Isso pode significar que o conjunto de dados fornece informações com poucos relacionamentos entre si, não favorecendo as predições desejadas. Grafos representando apenas os contatos com um determinado nó também foram gerados, eliminando-se a interferência de outros nós para facilitar a visualização, mas não permitiram encontrar coincidências importantes para serem exploradas.

Não foi possível verificar nós que apresentaram regularmente maior frequência de contato na maioria dos dias. Também não foi possível verificar padrões nas sequências dos contatos nos grafos individuais por nó.

Foi realizada a verificação de regularidades nas informações de entrada e saída para a função de predição. Foram gerados gráficos da matriz de pontos e testes de agrupamento com os dados do DieselNet utilizando as ferramentas de mineração de dados do *software* Weka¹ (HALL et al., 2009).

Considerando cada coluna do arquivo de dados como um atributo, a Figura 9 mostra o gráfico da matriz de pontos disponibilizados pelo Weka para análise visual de relação entre os atributos. É possível verificar que os atributos: nó de referência (CurrentNode), nó de contato anterior (LastContactNode), momento do contato anterior (Last-

¹O Weka reúne diversos algoritmos para mineração de dados. Disponível em <http://www.cs.waikato.ac.nz/ml/weka/> (Universidade de Waykato - NZ)

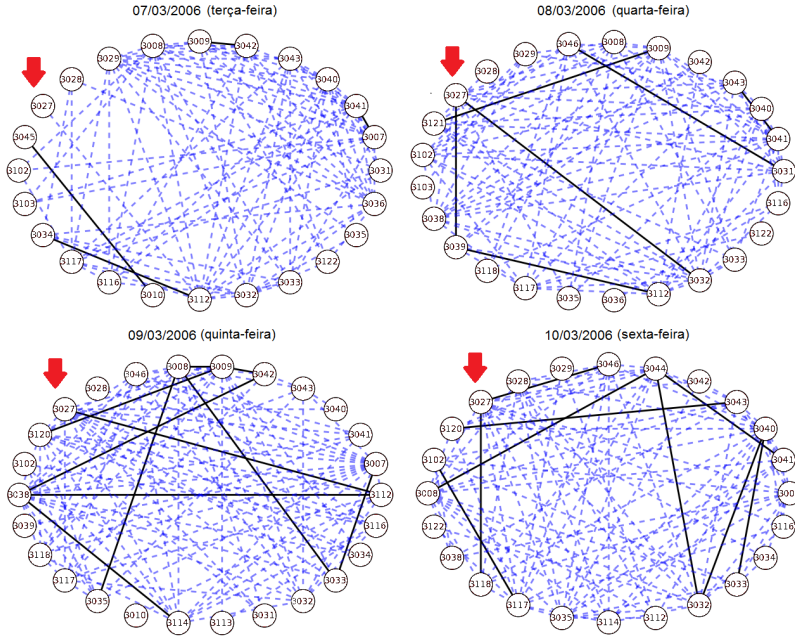


Figura 8: Exemplos de grafo de contato entre nós do DieselNet.

ContactedTime), nó contatado (ContactedNode), duração desse contato (Duration) e momento do próximo contato (StartingTime), são cruzados entre si e apresentados nos gráficos. Além disso, cada cor (variando do azul para o laranja - *Class Colour*) nos pontos do gráfico representa um identificador diferente do próximo nó de contato.

Verifica-se nesse gráfico nos pontos indicados pelas setas em vermelho, que existe uma dispersão de pontos quando avaliados os nós e instantes de contatos, levando-se a concluir que os mesmos nós se contataram em momentos muito variados no tempo. Baseando-se nos pressupostos de operação do STPU isso não é uma situação normal e poderia ser influenciada por um erro de interpretação dos dados.

Diante dos resultados dessas análises, foi necessário verificar se isso acontece por causa da complexidade dos dados reais ou se existe um tratamento mais adequado que permita obter relações mais úteis entre os dados.

Um conjunto de dados controlado foi gerado sinteticamente para verificar a irregularidade na ocorrência dos contatos do STPU. Após

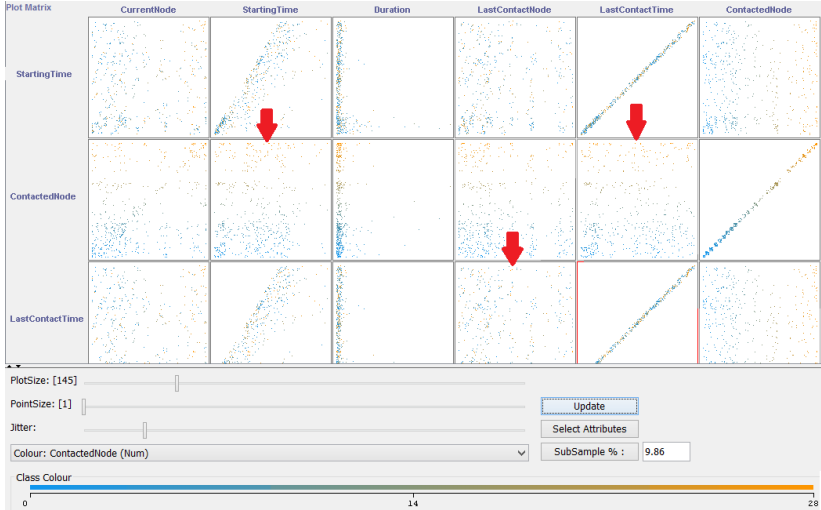


Figura 9: Tela do Weka com matriz de pontos geral.

estabelecer um cenário do STPU, testes e parâmetros para aplicação no simulador ONE (KERÄNEN; OTT; KÄRKKÄINEN, 2009), foram gerados dados de contatos com variação da velocidade e tempo de permanência nos pontos de parada para 10 dias, conforme descrito na segunda seção do Apêndice B.

Voltou-se ao início do procedimento, mas foi mantida a mesma função de predição inicialmente determinada. A verificação da frequência de nós em contato nos dados sintéticos mostrou muita regularidade no estabelecimento dos contatos em dias diferentes. Foi verificada uma certa regularidade dos contatos de um nó de referência com os mesmos nós durante o período avaliado.

Na verificação da sequência de contatos, os grafos de contato para os dados sintéticos mostraram pouca regularidade temporal na sequência de estabelecimento dos contatos. As variações de velocidade e permanência no ponto de parada promoveram momentos de contato substancialmente diferentes em cada um dos dias simulados, tal como no conjunto de dados reais.

A utilização dos dados reais são importantes pois levam em consideração uma série de condições no estabelecimento dos contatos possíveis de serem simuladas.

Em detalhes de outros trabalhos que utilizaram os dados do DieselNet foi encontrado ser possível encontrar padrões de contatos no

nível de rota (ZHANG et al., 2007). Uma vez que qualquer nó poderia ser usado para executar uma rota, não haveria regularidade nos contatos estabelecidos pelos nós, mas deveriam haver regularidades entre rotas.

Foi desenvolvido um programa para recuperar as informações das rotas realizadas pelos ônibus no DieselNet em cada um dos dias. Somente em um dos experimentos desse projeto (Primavera de 2006) foi possível recuperar essas informações de rota.

Com a substituição do identificador do nó pelo identificador da rota, o Weka foi usado para verificar o relacionamento entre cinco rotas que fazem contato com a rota 31² (uma rota que apresentou um maior número de contatos, além das rotas 30³ e 38⁴ - mais detalhes na Figura 31 no Apêndice B) e algumas regularidades começaram a ser encontradas. Na Figura 10, os eixos do gráfico relacionam o nó (rota) de referência e o momento do último contato. O último nó (rota) contatado é mostrado na variação de cor. O *jitter* foi de 33% enfatizando as diferenças ou tendências em faixas de valores representados no gráfico. Isso é mostrado na figura com os atributos de nó de referência na abscissa, o momento do último contato na ordenada e o último nó contatado representado pela coloração. Nessa figura é possível verificar a existência de padrões de intercontato do nó de referência com outros nós. A sensível variação de cores representando o próximo nó não ajudou a destacar visualmente a sua relação com o nó de referência. É possível observar a existência de um padrão nos momentos dos contatos em cada nó de referência avaliado separadamente. Também é possível ver que existe uma concentração de contatos no tempo inicial e que em grande parte acontece com os nós de número menor (mais azuis, rotas menores que 24).

Em outra análise foram verificadas regularidades em contatos realizados na mesma rota em dias consecutivos da semana e em segundas-feiras consecutivas. Conforme mostrado nos gráficos das Figuras 11 e 12, verificam-se algumas coincidências nos contatos em dias diferentes acontecendo em momentos próximos. Isso foi observado nas 3 três rotas avaliadas, 30, 31 e 38.

Imaginava-se obter padrões mais definidos ao avaliar diferentes dias no mesmo dia da semana em comparação a diferentes dias da semana, contudo isso não foi confirmado.

O uso das rotas fornecem regularidades que ajudam a melhorar

²Rota 31: Sunderland / UMass / South Amherst

³Rota 30: North Amherst / UMass / Old Belchertown Road

⁴Rota 38: Mt Holyoke College / Hampshire College / Amherst / UMass

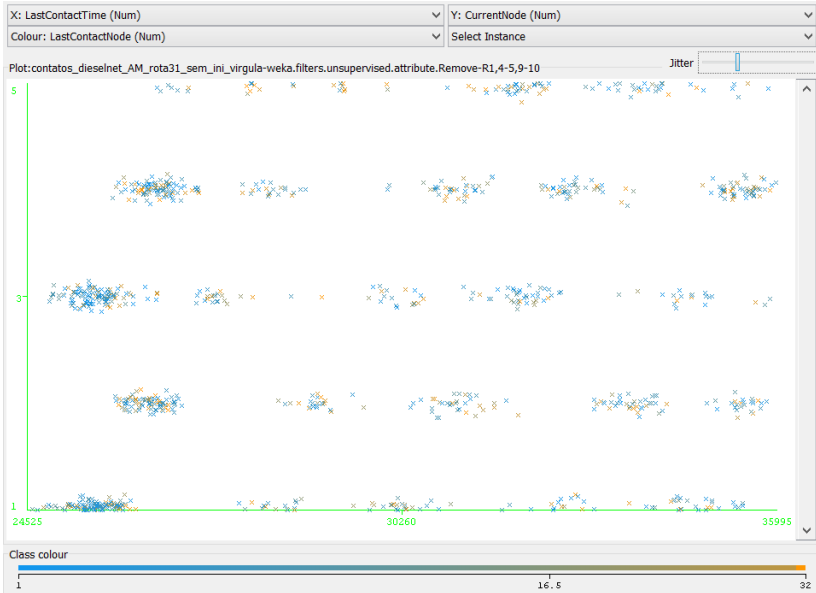


Figura 10: Matriz de pontos: contatos de 5 rotas com a Rota 31.

a predição de contatos, mas as avaliações utilizando as linhas de ônibus para identificação dos nós oferecem resultados melhores. Isso acontece por que uma rota pode ser executada algumas vezes no dia mas em condições de contato diferentes. Usar a identificação da rota ao invés da identificação da linha não favorece o treinamento da RNA, a não ser que se consiga extrair vantagens acrescentando a identificação da rota no projeto da RNA. Substituindo o identificador da rota pelo da linha esse trabalho é simplificado.

O uso do identificador da linha ou rota é necessário quando os ônibus não executam sempre as mesmas linhas. Observa-se nesse caso que a operação do sistema de roteamento utilizando o identificador da rota ou da linha deve pressupor a existência de um serviço de nomes ou outro mecanismo que permita que o nó assuma a identificação da linha e mantenha a identidade do ônibus se necessário.

Quanto a verificação das sequências de contatos entre contatos consecutivos, os primeiros testes foram conduzidos com janelas de dados (horizonte de predição) de dois contatos (saltos) consecutivos, associando-se os nós de contato anterior e o posterior com o nó de referência. As análises realizadas mostraram que as sequências de nós

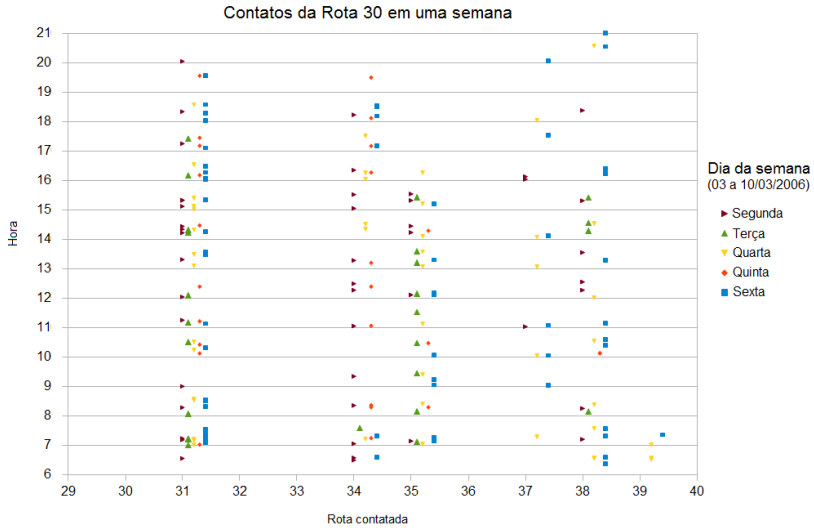


Figura 11: Gráfico de contatos da rota 30 em dias consecutivos.

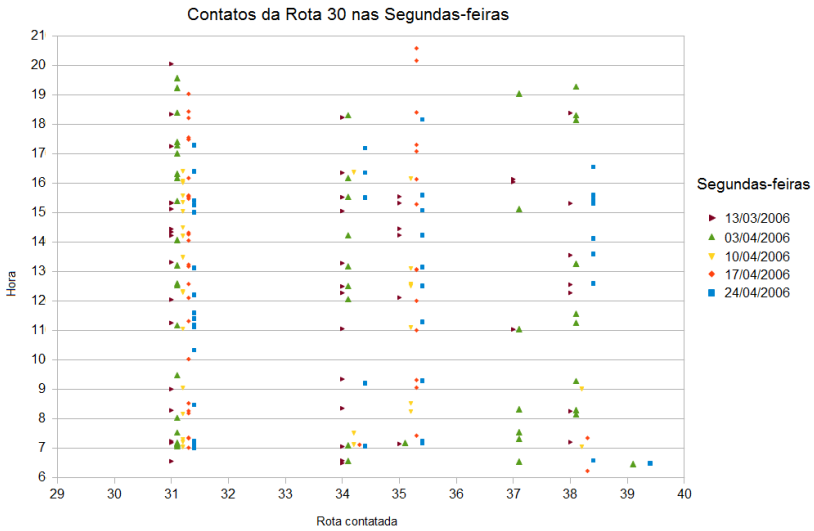


Figura 12: Gráfico de contatos da rota 30 nas segundas-feiras.

contatados estavam muito variadas, não permitindo encontrar padrões

em dias diferentes. Assim, para a representação da sequência de nós de contato, o horizonte de predição foi ajustado para apenas um salto, representando apenas a relação entre o nó de referência para o nó do próximo contato.

Já a relação do instante de contato entre os contatos consecutivos não foi descartada, mantendo-se a utilização do instante do contato anterior e o instante do próximo contato.

Após voltar a esta etapa algumas vezes para ajustes nas informações a serem usadas pela função de predição de contatos, as seguintes informações de entrada foram definidas:

- nó de referência;
- instante de contato anterior;

As informações objetivadas pela predição de contato mantiveram-se as mesmas desde o início:

- próximo nó de contato com o nó de referência; e
- próximo momento de contato com o nó de referência.

A relação entre as entradas e saídas se apresentaram adequadas para a utilização nas RNAs.

4.4.2 Etapa de definição das RNAs

Na sequência é apresentado o processo que definiu as duas topologias de RNAs utilizadas no trabalho: RNA global e RNAs locais.

Definição da topologia das RNAs

O tipo de RNA escolhido para realizar predição foi o *feedforward* com percéptrons (neurônios artificiais) em múltiplas camadas (MLP) e com aprendizagem supervisionada. Além dessas, as RNAs do tipo recorrentes também são indicadas para a tarefa de predição. Pequenos ensaios com as redes recorrentes (Apêndice A) foram conduzidos, mas não apresentaram melhorias em relação a *feedforward* MLP. Diante disso, sendo a MLP mais conhecida e apresentar mais informações disponíveis, optou-se por ela.

Inicialmente a definição das camadas da RNA e seus neurônios foi feita estatisticamente, mas com o avanço na implementação das

RNAs desse trabalho, essa definição foi feita em tempo de execução baseada nas informações de entrada.

Uma técnica utilizada inicialmente para o ajuste dos neurônios da RNA é a técnica de poda (*prunning*). Essa técnica é simples, porém gulosa e demorada. Ela consiste em remover neurônio por neurônio das camadas intermediárias, iniciando pela mais próxima da saída e testar seu acerto até se encontrar uma topologia satisfatória (i.e. uma que apresenta menor erro de treinamento e de validação).

Nas RNAs com camadas intermediárias definidas dinamicamente, a técnica empregada consiste em determinar o número de neurônios de acordo com um cálculo que utiliza o número de neurônios da camada de entradas, o número estimado de combinações possíveis dos valores de cada entrada e um valor que indica a precisão desejada na captura de detalhes.

Dois topologias de RNA foram utilizadas nessa proposta: a RNA global e as RNAs locais.

RNA global

A concepção de uma RNA treinada que fosse a mesma para todos os nós, permitindo-se explorar o conhecimento sobre a ocorrência de contatos considerando a situação de todo os nós em operação no ciclo, não apenas os contatos estabelecidos com um determinado nó, é interessante de ser avaliada. Ela foi denominada RNA global.

Para a RNA global foram testadas diferentes topologias nas camadas intermediárias e foi escolhida a que apresentou melhor aprendizagem⁵. Para os testes do preditor de contatos sobre os dados sintéticos de dez dias, utilizando um método de poda simples foi definida apenas uma camada intermediária com 6 neurônios. Com os dados do Diesel-Net e usando o método de poda, chegou-se na topologia de 15 neurônios na primeira camada intermediária e 9 na segunda.

A RNA global mostrada na Figura 13 apresenta duas camadas intermediárias com o número de neurônios indefinido, dependendo dos dados a serem utilizados e do método de poda. O número de neurônios das camadas de entrada e de saída dependem da codificação das informações de entrada e saída da função de predição. Além do mais, ele pode variar dependendo das informações do histórico de contatos.

Na Equação 4.3 é apresentado o modelo empregado para a RNA global. Nessa equação os valores de saída da RNA $((o_1, \dots, o_k))$ são a

⁵em termos do menor erro de treinamento e menor erro de validação em um menor número de ciclos de treinamento, conforme visto na atividade seguinte.

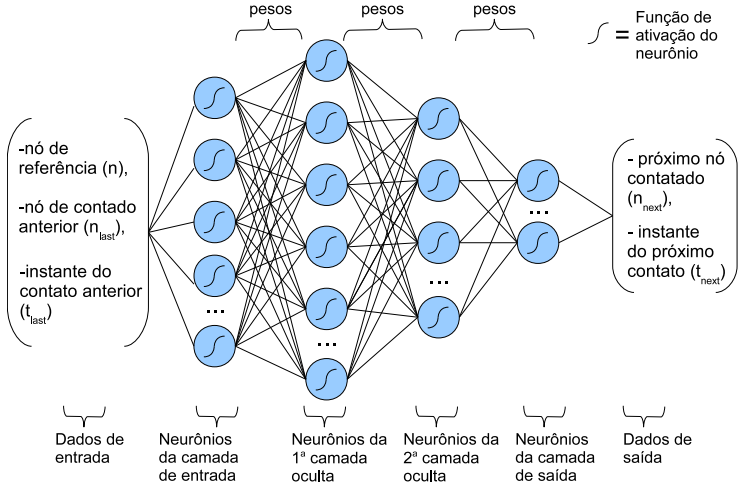


Figura 13: Topologia da RNA Global como proposta da função de previsão de contato.

codificação de n_{next} e t_{next} da Equação 4.1. De forma similar, os valores da entrada da RNA ((i_1, \dots, i_m)) são a codificação de n , n_{last} e t_{last} . Considerando que os conjuntos de pesos (w_{jk}, x_{lj}, y_{ml}) da RNA proposta é uma representação dos dados históricos de contato no intervalo de tempo $[\tau_1, \tau_2]$, a função de previsão de contato poderá ser descrita da seguinte forma:

$$o_k = \varphi_4 \left(\sum_j w_{jk} \varphi_3 \left(\sum_l x_{lj} \varphi_2 \left(\sum_m y_{ml} \varphi_1(i_m) \right) \right) \right) \quad (4.3)$$

onde:

- o_k são os valores de saída dos neurônios da camada de saída (camada 4). São valores codificados representando o próximo nó de contato e o próximo momento de contato preditos ($\hat{n}_{next}, \hat{t}_{next}$);
- m, l, j e k indexam os neurônios em cada camada (da entrada até a saída). É previsto que todas as camadas menos a de saída tenham um neurônio de bias.
- $\varphi_1, \dots, \varphi_4$ são as funções de ativação da camada de entrada (1),

das camadas ocultas (2 e 3) e da camada de saída (4). Nas camadas 1 a 3 é utilizada a função tangente-hiperbólica ou logística, dependendo da faixa dos valores de entrada e na camada de saída (4) é utilizada uma função linear.

- w_{jk} , x_{lj} e y_{ml} representam os conjuntos de pesos das ligações sinápticas entre os neurônios entre camadas adjacentes. Os dados do histórico no intervalo de $[\tau_1, \tau_2]$ são usados no treinamento para definir o valor desses pesos.
- i_m são os valores de entrada dos neurônios da camada de entrada (camada 1). São valores codificados representando n , n_{last} e t_{last} ;

Pretende-se com a RNA global capturar regras gerais no estabelecimento dos contatos levando-se em consideração os contatos com todos os nós da rede, mesmo que não estejam diretamente envolvidos. Além disso, pretende-se tirar proveito da facilidade de criação e reprodução da RNA nos nós, uma vez que é a mesma RNA para todos os nós. Contudo, deve-se prestar atenção no aumento do número de variações de contato a serem modeladas durante o treinamento, podendo aumentar muito o número de neurônios da RNA ou inviabilizar um aprendizado eficiente e fornecer previsões insatisfatórias.

RNAs locais

O estudo da criação dinâmica da RNA apontou a possibilidade de divisão da RNA global proposta em várias RNAs locais. Isso diminui a interferência causada por dados de nós não relacionados a previsão, diminuindo o número de combinações e, conseqüentemente, o número de neurônios das camadas intermediárias (ocultas). Baseada na diminuição da interferência, uma RNA diferente foi organizada para cada informação de saída.

As RNAs locais tem a topologia definida conforme mostrado na Figura 14. Da mesma forma que na RNA global, a topologia é influenciada pela codificação das informações de entrada e pelos métodos de definição do número de neurônios das camadas intermediárias (p.e. poda e cálculo baseado nas informações de entrada).

A equação da função de previsão para as RNAs locais segue a uma formulação parecida com a da Equação 4.3. A diferença é que para esta são apresentadas duas RNAs, cada uma retornando uma das informações de saída da função de previsão. Além disso, essas RNAs apresentam apenas uma camada intermediária e preveem informações

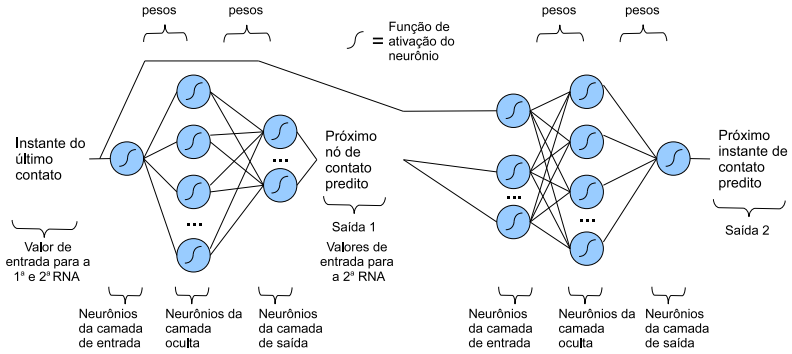


Figura 14: Topologia das RNAs locais proposta com função de predição de contato.

diferentes de entrada e saída. Uma deve prever o próximo nó de contato e a outra deve prever o instante em que esse contato ocorre. Da mesma forma que na RNA global as entradas e saídas são codificadas, sendo que pode haver mais de um neurônio para representar uma informação de entrada ou de saída. Essas RNAs são modeladas pelas equações 4.4 e 4.5.

$$o_k = \varphi_3 \left(\sum_j w_{jk} \varphi_2 \left(\sum_m x_{mj} \varphi_1(i_m) \right) \right) \quad (4.4)$$

onde:

- o_k são os valores de saída dos neurônios da camada de saída (camada 3). São valores codificados representando o próximo nó de contato predito (\hat{n}_{next});
- w_{jk} e x_{mj} representam os conjuntos de pesos das ligações sinápticas entre os neurônios entre camadas adjacentes. Os dados do histórico no intervalo de $[\tau_1, \tau_2]$ são usados no treinamento para definir o valor desses pesos.
- i_m é o valor da entrada codificada representando t_{last} ;

$$o = \varphi_3 \left(\sum_j w_j \varphi_2 \left(\sum_m x_{mj} \varphi_1(i_m) \right) \right) \quad (4.5)$$

onde:

- o é o valor de saída da RNA. É um valor codificado representando o próximo instante de contato predito (\hat{t}_{next});
- w_j e x_{mj} representam os pesos das ligações sinápticas entre os neurônios entre camadas adjacentes. Os dados do histórico no intervalo de $[\tau_1, \tau_2]$ são usados no treinamento para definir o valor desses pesos.
- i_m são os valores da entrada codificada representando o próximo nó predito (\hat{i}_{next}) pela Equação 4.4 e do último instante de contato t_{last} ;

Apesar de haver a execução de duas RNAs, deve-se considerar para efeitos de cálculo de utilização de processador que as RNAs locais somam juntas um número próximo de neurônios comparado com a RNA global. Apesar de ter um número parecido de neurônios, com a diminuição de uma camada intermediária haverá menos conexões, sendo mais rápido calcular o resultado.

Com as RNAs locais pretende-se melhorar as predições diminuindo a interferência de informações de contato que não estão diretamente envolvidos com o nó de referência e assim, reduzir a complexidade da RNA. Contudo, a manutenção separada de cada RNA incluindo o treinamento separado para cada RNA é mais trabalhosa e o espaço de armazenamento das RNAs também é aumentado, uma vez que cada nó deve manter duas RNAs para cada nó a ser usado como referência para consulta.

Implementação da RNA

As RNAs foram criadas primeiramente através de um software simulador de RNAs (JustNN⁶). Com a necessidade de programação dos primeiros modelos de RNA foi utilizada uma biblioteca em linguagem de programação Java (*Feed Forward Neural Network in Java* (FFNN)⁷). Com o avanço dos experimentos, devido às limitações dos dois softwares utilizados, as RNAs foram criadas utilizando o *framework* Encog (HEATON; CONTRIBUTORS, 2013).

O Encog apresenta um conjunto de bibliotecas bastante referenciado por desenvolvedores de RNAs. Apesar da complexidade devido

⁶ *Software* de simulação de RNA desenvolvido pela Neural Planner Software. Disponível em <http://www.justnn.com/application>

⁷ Criada por Aydin Gürel. Software obtido em <http://www.ncorpus.com/aydingurel/>

ao grande número de opções que esse software disponibiliza foi possível criar dinamicamente múltiplas RNAs, testar o desempenho obtido com outros algoritmos de aprendizagem, ter mais opções para a codificação dos dados de entrada e saída da RNA e ter mais opções para a definição do critério de parada da aprendizagem. Além das bibliotecas para implementação de programas, o Encog conta com uma ferramenta de *Workbench* que permite realizar de forma limitada a criação da RNA através de uma interface gráfica com menus, além de permitir visualizar gráficos de aprendizagem e validação.

Separação dos exemplos para treinamento, validação e testes

Do total de exemplos de treinamento disponíveis no conjunto de dados, foram separados para teste aqueles referentes a um determinado ciclo de movimentação dos nós (no caso dia) a ser testado. O número de ciclos e o erro de validação foram utilizados como critério de parada nas primeiras RNAs implementadas. A cada ciclo de treinamento, os exemplos de validação são submetidos a RNA e as saídas desses exemplos comparadas às saídas desejadas obtendo-se uma avaliação do erro. Após treinar a RNA, ao trocar o conjunto de exemplos utilizado para validação pelos exemplos que representam os dados do dia a ser testado é possível determinar a taxa de acerto na predição.

Uma desvantagem do uso das ferramentas do Encog foi que a validação realizada através da sua ferramenta de *Workbench* é feita comparando os valores codificados, interferindo na avaliação da validação dos valores reais. Programas foram criados para realizar a validação de valores reais.

Ajuste dos critérios de parada do treinamento

A aprendizagem realizada na RNA implementada é do tipo supervisionado, sendo inicialmente utilizado o algoritmo de retropropagação. Com o objetivo de diminuir o tempo de treinamento (*Backpropagation*) utilizou-se a seguir o algoritmo de retropropagação resiliente (*Resilient Backpropagation*).

Nos primeiros experimentos foram usados critérios simples de parada do treinamento, baseando-se no número de ciclos, determinado de acordo com as observações de vários treinamentos. Nos experimentos posteriores foi levada em consideração a diminuição da taxa do erro de treinamento ao passar dos ciclos.

A Figura 15 mostra o gráfico do aprendizado em um treina-

mento com dados sintéticos de 20 dias extraído do software JustNN. Pode ser observado no retângulo na parte final da figura a estrutura da RNA *feedforward* MLP (4 neurônios na camada de entrada, com 15 e 12 neurônios nas camadas intermediárias e 2 neurônios na camada de saída). No parte central da figura é apresentado o gráfico de treinamento, mostrando o erro de treinamento mínimo (azul), máximo (vermelho), médio (verde) e o erro de validação (laranja). Pode-se observar que o treinamento rapidamente alcançou um baixo erro de validação, mas ocorreram instabilidades nesse erro a partir do ciclo de aprendizagem 6700. No retângulo do lado direito da imagem são apresentados os critérios de parada para o treinamento. Observa-se que 80% dos exemplos de treinamento avaliados foram validados como corretos, ou seja, os valores preditos coincidiram com os valores desejados.

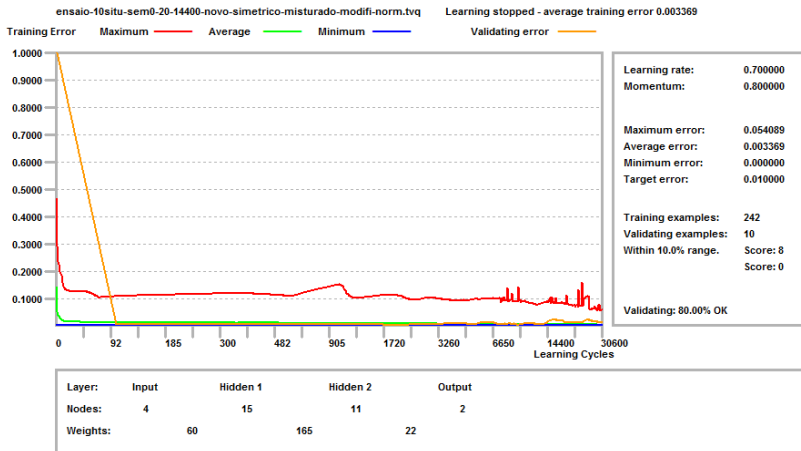


Figura 15: Gráfico de treinamento de uma RNA.

A partir deste gráfico, pelo número de ciclos de treinamento elevado (30600), pode-se perceber que com esse treinamento se obteve uma rede excessivamente treinada para os exemplos fornecidos, diminuindo-se a capacidade de generalização da RNA. Nesse treinamento, os exemplos de validação provêm de exemplos do conjunto de treinamento.

Nos treinamentos realizados nas RNAs criadas com o *framework* Encog, foram utilizadas as seguintes condições como possíveis critérios de parada:

1. parar o treinamento quando o erro de predição não variar mais do que 10^{-4} em 100 ciclos de treinamento;

2. parar o treinamento quando o erro de predição for menor do que 10^{-3} ;
3. parar o treinamento se atingir 4900 ciclos de treinamento. Esse valor foi baseado em diversos treinamentos com o conjunto de dados do DieselNet que apontaram não se ter ganhos na validação a partir dessa quantidade de ciclo. Para a predição do próximo nó de contato foi necessário um número de ciclos menor.

A condição 1 procura evitar um treinamento excessivo quando a rede não está mais aprendendo (o erro de treinamento não varia mais). Na condição 2 procura-se definir quando a diferença para o valor ideal deve ser adequado para parar o treinamento. Na condição 3 trata-se do caso em que a rede continua aprendendo, mas não consegue atingir o erro desejado e é forçada a terminar o treinamento para a rede não perder a capacidade de generalização.

Além dessas condições, foi interessante reiniciar automaticamente o treinamento que não consegue convergir rapidamente para menores erros de treinamento (não consegue sair de um mínimo local com erro de treinamento alto) ao observar que nessas condições ao final do treinamento dificilmente se obtém baixos erros de treinamento. Assim, o treinamento foi reiniciado (*reset*) até um determinado número de vezes com novos pesos (aleatórios, mas uniformemente distribuídos) se não for obtido um erro baixo até um determinado número de ciclos. Nos treinamentos realizados foi observado que 700 ciclos e 6 reinícios foram suficientes para oportunizar a obtenção de um erro menor que 15% para predição no próximo nó de contato e menor que 25% para o próximo momento de contato.

4.4.3 Etapa de testes de avaliação do preditor de contatos

Utilizando o conjunto de dados sintéticos de 10 dias foram realizados testes comparativos do percentual de acerto na predição. Essa comparação foi realizada entre a função de predição de contato baseada em RNAs e uma função de frequência do contato (que fundamenta as estratégias de escolha dos nós no PRoPHET e no MaxProp). O histórico de contato usado para determinar a frequência de contatos é o mesmo usado no preditor baseado na RNA. A função de frequência de contato fornece como resultado da predição o nó com maior frequência de contato com um nó informado e a duração média de contato desse nó. A taxa de acerto é determinada com o número de consultas que

Tabela 3: Taxa de acerto (%) das funções de predição em cada dia.

Dia	1	2	3	4	5
RNA	100	100	83	100	100
Frequência	75	50	50	100	100
Dia	6	7	8	9	10
RNA	75	70	100	71	75
Frequência	63	40	57	50	72

acertaram o contato futuro em relação ao número total de consultas realizadas. Para o instante futuro de contato foram considerados corretos os valores dentro de uma margem de $\pm 10\%$ do valor desejado.

O preditor de contato baseada em RNA global apresentou melhores resultados do que o preditor baseado na frequência dos contatos para o contexto do trabalho. A RNA alcançou 87,4% de taxa de acerto em comparação a 65,7% alcançados pela função de frequência. Esta vantagem também foi observada individualmente nos casos testados. Valores individuais são apresentados na Tabela 3. (SEGUNDO; FARINNES; SILVA, 2013). Em experimentos aumentando em 10% a faixa de valores de velocidade e tempo de permanência nas paradas, as mesmas taxas de acerto foram observadas.

Esses resultados foram obtidos parando o treinamento ao atingir o erro de predição de $\pm 10\%$, mas taxas de acerto similares foram observados quando o treinamento foi parado com erro de predição em $\pm 5\%$.

Praticamente as mesmas taxas de acerto foram verificadas modificando-se o critério de validação para parada de treinamento quando o instante do próximo contato está dentro de $\pm 10\%$ do valor desejado para todos os exemplos de validação.

Pode-se observar na Tabela 3 uma queda maior na taxa de acerto nos casos 6, 7, 9 e 10. Dois fatores podem ter influenciado esse resultado: i) um aumento da faixa de velocidade e tempo de permanência nas paradas sofrido pelos nós; ii) condições excepcionais existentes no histórico de contatos muito diferentes dos contatos que ocorreram no caso testado.

Além do mais, utilizando os dados do DieselNet e RNA global as taxas de acerto conjunto da predição do próximo nó e instante de contato variaram de 43% a 75%. Com RNAs locais as taxas de acerto foram de 68% até 86%.

Deve-se considerar que esses resultados foram obtidos na maioria

dos experimentos e que dependendo do caso avaliado e do treinamento realizado essas taxas podem ser maiores ou menores.

4.5 CONCLUSÃO

A RNA pode ser uma poderosa ferramenta para modelar não linearidades. Todavia, exige muita atenção para a preparação dos dados que devem ser usados no treinamento e na definição do tipo de RNA e topologia. Existe ainda muito campo de estudo para melhorar a aplicação das RNAs na predição, seja através do emprego de outros tipos de RNA, algoritmos de treinamento ou métodos para definição das suas entradas e topologia. Além disso, a própria área de RNA está evoluindo e oferecendo novas formas de aplicação.

Foi possível desenvolver a proposta de um preditor de contatos baseado em RNA oferecendo maior acerto do que o método convencional baseado na frequência. A codificação das entradas, as mudanças na topologia (global, local) e no treinamento das RNAs permitiram obter melhorias na predição. A principal modificação diz respeito ao aproveitamento adequado das informações para que elas não se tornem interferências e atrapalhem o trabalho da RNA.

O procedimento utilizado para desenvolver as RNAs permitiu melhorá-las, tal como em um processo evolutivo. O processo de desenvolvimento das RNAs ainda depende muito de um processo de tentativa e erro, razão pela qual o procedimento deve permitir a volta à etapas anteriores para aperfeiçoamentos.

A RNA global apresenta um grande potencial uma vez que os nós tentam se aproveitar de toda a informação da rede. As RNAs locais fornecem melhores resultados pois eliminaram a interferência das informações dos outros nós que não fazem contato com o nó de referência. Podem existir outras alternativas para construir e aplicar RNAs para o propósito em questão, mas dependem de outras análises e informações que possam ser obtidas, tais como o tempo de intercontato e a probabilidade de alcançar determinado nó.

O preditor de contatos apresentado com RNA global e RNAs locais é utilizado pelo preditor de jornadas descrito no capítulo anterior. Os resultados obtidos nos experimentos com o preditor de jornadas proposto, mediante diversas situações para comparação, são apresentados no próximo capítulo.

5 AVALIAÇÃO DO ROTEAMENTO DTN BASEADO EM RNA

Neste capítulo são apresentados os resultados obtidos na avaliação da estratégia de roteamento proposta e comparações com outras estratégias no cenário do STPU.

Primeiramente são apresentados os objetivos da avaliação, as métricas utilizadas e os cenários das avaliações. As análises dos experimentos são apresentadas na sequência e são discutidos os ajustes nos preditores que foram realizados em decorrência das observações. Por fim é apresentada a conclusão.

5.1 MÉTRICAS E AVALIAÇÕES

As métricas para avaliação utilizadas neste trabalho são:

- o número de mensagens entregues;
- o tempo de execução do algoritmo;
- o atraso na entrega das mensagens;
- o custo de entrega das mensagens; e
- o número de decisões de encaminhamento.

Os seguintes parâmetros serão variados para avaliar a estratégia de roteamento:

- o número de cópias da mensagem;
- o tamanho da fila de mensagens; e
- a duração dos contatos.

Dois tipos de avaliações são apresentadas neste trabalho: (i) utilizando dados sintéticos; e (ii) utilizando contatos reais. Detalhes desses conjuntos de dados estão descritos no Apêndice B. Os dados sintéticos permitem avaliar a solução de roteamento proposto em cenários controlados. Considera-se que existem variações aleatórias de velocidade dos nós e faixa do tempo de permanência nos pontos de parada. Entretanto, não são levados em conta os problemas de comunicação e as falhas nos nós.

Dois tipos de arquitetura de RNA foram utilizada e testadas para o preditor de contatos, conforme descrito na seção 4.4.2: (a) RNA global; e (b) RNAs locais. Com a RNA global foi possível verificar o desempenho da RNA relacionando contatos entre todos os nós da rede, inclusive aqueles indiretamente envolvidos com um determinado nó de referência. As RNAs locais apenas levam em consideração os contatos realizados pelo nó de referência.

A avaliação da estratégia de roteamento proposta é comparada com outras propostas, em particular as que:

- não exijam informações adicionais além das informações básicas do histórico de contatos;
- foram avaliadas com dados de contato reais; e
- apresentam reconhecidamente um bom desempenho.

Neste sentido, a abordagem do caminho de probabilidade máxima de entrega (MaxProp) e a probabilidade de contato (PRoPHET) são apropriados para realizar comparações. De acordo com alguns trabalhos, o MaxProp oferece melhor desempenho quanto a entrega das mensagens, comparando-se outras abordagens no contexto de contatos oportunistas, incluindo o PRoPHET. Assim, o MaxProp foi utilizado para comparações, além do Epidemic e a Entrega Direta.

Ao comparar os resultados entre as abordagens de roteamento em DTN é importante reconhecer os limites de referência. Dessa forma são utilizadas:

1. a abordagem de inundação (implementada pelo Epidemic) sem limitações de recursos para estabelecer o limite máximo para a entrega de mensagens, atraso na entrega de mensagens e o consumo de recursos; e
2. a abordagem de entrega direta (ou espera para entregar), para estabelecer uma referência de desempenho inferior para a entrega de mensagens, o atraso na entrega de mensagens e o consumo de recursos.

Para obter os dados utilizados nas avaliações, foi desenvolvido um simulador de comunicação DTN que simula a troca de mensagens a partir dos dados de contatos. Neste simulador foram implementadas as estratégias de roteamento avaliadas. Detalhes de implementação desse simulador e das estratégias simuladas foram descritos no Apêndice C.

São utilizados neste simulador os contatos obtidos no mundo real, os quais reproduzem problemas de comunicação e detalhes imprevisíveis de mobilidade, que acabam sendo refletidos nas oportunidades de contato. Testes também foram conduzidos com dados sintéticos obtidos em cenários com movimentação e comunicação (física e de enlace) simuladas a partir de itinerários com pontos de parada tal como no STPU.

5.2 CENÁRIOS, EXPERIMENTO E IMPLEMENTAÇÕES PARA A AVALIAÇÃO

5.2.1 Os cenários

A estratégia de roteamento proposta foi avaliada no contexto de contatos entre ônibus do Sistema de Transporte Público Urbano (STPU). Em cenários desse contexto existe um agendamento prévio dos locais e horários de parada dos ônibus, mas sujeito a imprecisões devido à movimentação dos ônibus e no tempo de permanência nas paradas. A movimentação é repetida dentro do dia e também no dia seguinte conforme já discutido anteriormente.

O conjunto de dados de contatos entre ônibus do STPU utilizado neste trabalho foi obtido a partir do projeto DieselNet (Apêndice B). Nesse projeto, 40 ônibus foram equipados com dispositivos de comunicação sem fio e foram registrados os dados de contato em 55 dias. Os motivos para essa escolha dentre outros conjuntos de dados são:

- a quantidade de dias registrados, maior do que os outros conjuntos de dados encontrados. Essa grande quantidade de dados é importante para o aprendizado da RNA uma vez que os contatos são oportunistas;
- o fornecimento do momento exato do contato evitando a necessidade de interpolações ou outros mecanismos para prever a ocorrência do contato.
- a representação de contatos reais, sujeitos as condições de mobilidade e comunicação que aconteceram de fato.

Devido ao fato de que os dados reais de contato apresentam condições imprevisíveis e que podem interferir na avaliação dos experimentos, torna-se necessário a utilização de um conjunto de dados controlado. O controle na geração dos dados ajuda a determinar se os

resultados obtidos na avaliação não estão sendo influenciados por essas condições.

Por isso, dois conjuntos de dados sintéticos foram criados para a avaliações dos preditores. O modelo das ruas segue o modelo em topologia de grade, tal como no Manhattan (BAI; HELMY, 2006) e o modelo de movimentação é uma variação do *Random Waypoint* (JOHNSON; MALTZ, 1996) considerando que a direção da movimentação é pré-estabelecida e em cada ponto de parada a velocidade até o próximo ponto de parada e o tempo de permanência nele são estabelecidos aleatoriamente.

O cenário para os dados sintéticos foi estabelecido com cinco nós e quatro rotas, somando 26 pontos de parada e duração total de 21600 segundos. No primeiro conjunto de dados foram geradas 10 variações do cenário e no segundo conjunto de dados 20 variações. Cada variação determina uma faixa de velocidade e uma faixa de tempo de permanência nos pontos de parada para utilização no modelo de movimentação. Detalhes estão disponíveis na segunda seção do Apêndice B. A diminuição do número de nós em relação ao DieselNet, a simplificação nas variações nos movimentos dos ônibus, a limitação da faixa de tempo de permanência nos pontos de parada e a eliminação de problemas na comunicação permitiram obter contatos com menos interferências.

Os experimentos realizados com dados sintéticos não consideraram cópia de mensagens, sendo realizadas as seguintes avaliações:

- entrega de mensagens usando RNA global e RNAs locais;
- tempo de execução usando RNAs locais.

Nos experimentos utilizando os dados reais de contato e RNAs locais, por apresentarem melhor predição, foram realizados as seguintes avaliações:

- número de mensagens entregues;
- atraso na entrega das mensagens;
- tempo de execução;
- influência das decisões de encaminhamento na entrega das mensagens;
- influência do número de cópias da mensagem na rede: número de mensagens entregues; atraso na entrega das mensagens; número de mensagens geradas; e custo de entrega;

- influência do controle da fila de mensagens e da duração da conexão: número de mensagens entregues variando o tamanho da fila de mensagens; e número de mensagens entregues com duração real dos contatos;

5.2.2 As implementações e parametrizações dos algoritmos

A implementação do MaxProp utilizada no simulador construído para este trabalho (Apêndice C) é o mesmo utilizado no simulador e redes oportunistas ONE(KERÄNEN; OTT; KÄRKKÄINEN, 2009). Nessa implementação, existem otimizações que reduzem significativamente o tempo de execução. O uso da mesma implementação pretende evitar comparações injustas que pudessem apresentar perda de desempenho.

Na implementação da estratégia baseada em RNA, apenas uma otimização foi realizada para simplificar a obtenção dos menores custos de tempo absoluto de entrega, dispensando o uso de um algoritmo de Dijkstra modificado.

No algoritmo de construção das jornadas previstas existe um parâmetro que controla o passo no tempo *timeSlice* para realizar uma próxima consulta à RNA. Esse parâmetro determina o intervalo de tempo entre predições, simulando o tempo de intercontato.

Usando as informações do histórico de contatos, foram determinados os valores do tempo de intercontato médio e o desvio padrão dos intercontatos para todo o conjunto de dados de treinamento. Para os dados do Dieselnet obteve-se 1.137 segundos de intercontato médio e 1.454 segundos de desvio padrão dos intercontatos. Nos dados sintéticos com 20 variações obteve-se 3.587 segundos de intercontato médio e 2.741 segundos no desvio padrão dos intercontatos. A metade do desvio padrão foi adotada nos experimento como valor do *timeSlice*.

O algoritmo de construção de jornadas também foi configurado para avançar a busca por contatos em profundidade. O uso de profundidade três no parâmetro *depth* do simulador mostrou-se adequado. Nessa profundidade o algoritmo considerou que pode haver até três saltos diferentes a partir do nó de referência fornecido na primeira chamada à função de predição.

5.3 AVALIAÇÃO DA ESTRATÉGIA DE ROTEAMENTO PROPOSTA USANDO CONTATOS SINTÉTICOS

Os experimentos e avaliações realizados com dados sintéticos e apresentados nessa seção tem por objetivo mostrar o desempenho da estratégia de roteamento proposta baseada em RNA quanto a entrega de mensagens e o tempo de execução. A RNA global se diferencia das locais por levar em consideração todas as informações de contato e não apenas aquelas relacionadas a partir de um nó (nó de referência) de cada vez, conforme explicado anteriormente na Seção 4.4.2.

Para evitar interferências de outros fatores nas avaliações, foi avaliado o roteamento sem cópia da mensagem, exceto na estratégia epidêmica ilimitada. Com o mesmo propósito, assume-se ter espaço suficiente nas filas de mensagens e banda suficiente para transmissão de todas as mensagens.

5.3.1 Cenário para as simulações

Através do simulador ONE, o cenário para os dados sintéticos foi estabelecido com cinco nós e quatro rotas com 26 pontos de parada somando os pontos de todas as rotas. Foram modeladas vinte variações do cenário e gerados dados sintéticos de contato com o dobro de dias em relação ao gerado anteriormente (Apêndice B). O histórico de contato gerado foi processado para aplicação no preditor de jornadas utilizando-se somente as primeiras 4 horas (14.400 segundos).

O simulador de troca de mensagens desenvolvido para este trabalho (Apêndice C) foi configurado para cada nó enviar uma mensagem para todo outro nó na rede a cada 900 segundos, o que é realizado da mesma forma em cada estratégia de roteamento avaliada.

5.3.2 Análise da entrega

O desempenho da estratégia de roteamento foi avaliado utilizando o conjunto de dados sintéticos de 20 dias, sendo isolados os dados do dia testado e as RNAs treinadas com os dados dos 19 dias restantes. Um critério simples de parada do treinamento foi utilizado, quando foram atingidos 850 ciclos de treinamento. Esse número de ciclos de treinamento foi obtido depois de avaliar o erro de treinamento e o erro de validação em diversos treinamentos. Os resultados apresentados

para a estratégia baseada em RNA são a média de cinco execuções com treinamentos diferentes consecutivos. Assim, procurou-se utilizar conjuntos de pesos diferentes para as RNAs sem escolhê-los. Os resultados foram obtidos para a estratégia proposta utilizando RNAs globais e depois RNAs locais.

Resultados com RNA Global

Os resultados obtidos utilizando os dados sintéticos com RNA global apresentam vantagem em apenas em alguns dias testados. Esses resultados apresentaram-se muito sensíveis ao treinamento realizado. A Tabela 4 mostra o número de mensagens entregues para cada estratégia em cada um dos nove dias avaliados. São apresentados os resultados mínimos (Min), máximos (Max) e na média da entrega de mensagens com a estratégia baseada nas RNAs.

Tabela 4: Número de mensagens entregues (RNA global).

Dia	EI	ED	MP	RNA		
				Min	Max	Média
1	81	68	75	70	81	79,3
2	84	70	80	75	78	77,4
3	114	74	114	77	82	79,4
4	100	80	94	95	97	96,0
5	54	42	54	48	54	49,5
6	132	96	96	91	107	102,2
7	65	50	65	61	65	63,0
8	99	74	71	67	84	77,6
9	141	108	95	104	114	108,6
Σ	870	662	744	688	762	733,0

(EI) Epidemico ilimitado, (ED) Entrega direta, (MP) MaxProp e Preditor de jornadas (RNA).

Verifica-se que em 5 de 9 dias a estratégia proposta com RNA global teve o número médio de mensagens entregues superior ao MaxProp, conforme mostrado em preto na tabela. Na média geral de mensagens entregues não foi possível superar o MaxProp, mas se considerar apenas os valores máximo obtidos, foi possível obter uma pequena vantagem com a RNA global.

Resultado com RNAs locais

Os resultados obtidos utilizando os dados sintéticos com RNAs locais apresentam vantagem na maioria dos dias testados. A Tabela 5 mostra o número de mensagens entregues para cada estratégia em cada um dos nove dias avaliados.

Tabela 5: Número de mensagens entregues (RNAs locais).

Dia	EI	ED	MP	RNA		
				Min	Max	Média
1	81	68	75	79	81	80,5
2	84	70	80	84	84	84,0
3	114	74	114	81	98	92,9
4	100	80	94	98	98	98,0
5	54	42	54	54	54	54,0
6	132	96	96	98	115	103,6
7	65	50	65	64	66	65,0
8	99	74	71	82	99	92,3
9	141	108	95	112	120	114,0
Σ	870	662	744	752	815	784,3

(EI) Epidêmico ilimitado, (ED) Entrega direta, (MP) MaxProp e Preditor de Jornadas (RNA).

Comparado o número de mensagens entregues com a estratégia usada no MaxProp, a estratégia proposta usando RNAs locais foi pior em apenas um dos dias, sendo melhor em 6 e equivalente em 2 dos 9 dias avaliados, conforme evidenciado em preto nessa tabela. A estratégia baseada em RNA entregou de 1,07% a 9,54% mais mensagens do que a estratégia do MaxProp, obtendo um aumento médio de 5,41% na entrega de mensagens.

Para explicar que em alguns dias não foi possível obter vantagem, suponha-se que nesses dias o histórico de contatos não apresentou oportunidades de contato similares as que ocorreram nos outros dias usados para o treinamento. Um exemplo disso acontece quando o treinamento é baseado somente em dados de dias com contatos típicos e as predições são realizadas em um dia com contatos atípicos.

5.3.3 Análise do tempo de execução

A avaliação do tempo de execução usando RNAs locais com dados sintéticos levou a verificar a diferença no tempo de execução entre estratégias diferentes. O tempo de execução avaliado é baseado no tempo de simulação.

O tempo de execução de cada estratégia em nove dias diferentes foi obtido em cinco experimentos¹. A soma desses tempos de execução está apresentada na Tabela 6. O tempo de execução necessário para o treinamento da RNA não foi considerado, assim como o tempo de preenchimento da matriz de probabilidades usada no MaxProp, pois trataram-se de processos *off-line*. O tempo de transferência das mensagens não está incluído nos valores apresentados.

Tabela 6: Tempo de execução (ms) das estratégias de roteamento (dados sintéticos).

Dia	EI	ED	MP	RNA
1	126	58	157	48
2	52	65	25	36
3	48	16	33	32
4	23	17	27	29
5	9	7	11	16
6	20	13	25	25
7	13	10	44	15
8	15	12	17	24
9	28	18	27	24
Tempo total	208	158	209	201
Msgs. filas	20327	10838	10612	10567

(EI) Epidemico ilimitado, (ED) Entrega direta, (MP) MaxProp e Preditor de jornadas (RNA).

Observa-se que no primeiro dia há uma discrepância no tempo de execução obtido no MaxProp em relação a outras estratégias e a outros dias. Essa discrepância foi provavelmente obtida devido à questões de inicialização do programa de simulação. Para evitar influência nas

¹Os experimentos foram executados em um computador com processador Intel Core i52410M CPU de 64 bit em 2,3 GHz com 6 GB de memória RAM com sistema operacional Windows 8.1 Pro, sem privilégio para aplicações em primeiro plano (*foreground*).

análises os resultados desse dia não serão considerados.

Pode ser observado que em 3 dos 8 dias o tempo de execução da estratégia baseada em RNAs foi menor do que para as estratégia do MaxProp e em um dos dias foi equivalente (em preto na Tabela 6). A soma do tempo de execução de todos os dias a RNA também foi um pouco melhor.

Foi observado que há uma relação do aumento do tempo de execução com o aumento do número de mensagens na fila de mensagens para entrega. Assim, também foi avaliado comparativamente o tempo de execução para cada mensagem da fila de entrega baseado na soma dos tempos de execução nos vários dias e no total de mensagens que ficaram na fila de mensagens. Neste sentido, a estratégia baseada em RNAs levou menos tempo de execução por mensagem na fila do que o MaxProp, apesar de ser uma diferença pequena (3,4%). Como esperado, em comparação à estratégia proposta, a estratégia epidêmica ilimitada e a entrega direta apresentaram os menores tempos de execução por mensagem da fila (85,92% e 30,45% menor do que a RNA).

Nos experimentos realizados a RNA global obteve maiores tempos de execução do que aqueles realizados com RNAs locais.

5.3.4 Conclusões

Nesta seção foi avaliado o desempenho da estratégia de roteamento proposta usando RNA global e RNAs locais com o uso de dados sintéticos.

A avaliação da estratégia de roteamento proposta com uso das RNAs locais apresentou melhores resultados do que a RNA global.

Esses resultados indicam haver coerência na proposta de divisão da RNA por nós com o objetivo de diminuir a interferência de dados de outros contatos. Todavia deve-se perceber que a arquitetura desenvolvida para a RNA global ainda pode ser investigada para melhorar seus resultados.

É possível explicar que não foi possível obter vantagem na entrega das mensagens por que o histórico de contatos não representou adequadamente as oportunidades de contato ocorridas no dia avaliado.

Deve-se ponderar que o tempo de execução da estratégia proposta é influenciado pela topologia da RNA, sendo que quanto mais conexões entre os neurônios, maior é o tempo de execução. A codificação das informações de entrada e saída também pode aumentar o tempo de execução. Além disso, os parâmetros de profundidade (*depth*), tempo

entre predições (*timeSlice*) e o tamanho do intervalo de tempo para operação do preditor, também influenciam no tempo de execução.

5.4 AVALIAÇÃO DA ESTRATÉGIA DE ROTEAMENTO PROPOSTA USANDO CONTATOS REAIS

Os experimentos apresentados nesta seção tem por objetivo avaliar o desempenho da estratégia de roteamento proposta utilizando dados de contatos reais e realizar comparações com outras estratégias de roteamento para DTN. Os dados de contatos reais foram obtidos do projeto DieselNet (primeira seção do Apêndice B).

Os experimentos foram realizados utilizando o preditor de contato baseado em RNAs locais, pois apresentou melhores resultados com utilização dos dados sintéticos conforme apresentado na seção anterior e também em testes preliminares com os dados reais.

5.4.1 Cenário para as simulações

Do total de informações disponíveis, foram selecionados para uso os dados históricos de 33 ônibus que executaram 10 rotas em 32 linhas de ônibus, registrados durante 55 dias no período das 7 até as 10 horas da manhã.

O histórico do dia testado não foi usado no treinamento e na validação da RNA, sendo apenas usado para gerar os eventos no simulador de troca de mensagens desenvolvido (Apêndice C). Os mesmos dados do histórico também foram usados para definir os valores da matriz de probabilidades do MaxProp. Os resultados obtidos nas estratégias de roteamento testadas, entrega direta, epidêmica, MaxProp e a proposta de jornada preditas baseada em RNA foram obtidas para cada um dos 9 dias testados.

Em cada caso ou dia testado, a cada 15 minutos (900 segundos) cada nó gerou uma mensagem de 1 Kbyte para cada outro nó da rede. Toda a simulação foi executada no intervalo de tempo entre 25.200 até o 36.000 segundos (totalizando 3 horas).

Para se obter uma comparação justa, o algoritmo de predição de jornadas foi parametrizado para usar *maxDepth* e *maxBreadth* igual a 4 e *timeSlice* de 1137 segundos, enquanto o MaxProp foi parametrizado para trabalhar com o tamanho máximo da matriz de probabilidade igual a 32.

Os resultados da estratégia baseada em RNA foram obtidos na média de cinco execuções com treinamentos diferentes consecutivos, ou seja, com conjuntos de pesos diferentes e sem escolha.

5.4.2 Avaliação da entrega de mensagens

O experimento a seguir tem por objetivo a realização de comparações do número total de mensagens entregues apresentado na execução das estratégias de roteamento epidêmica ilimitada, entrega direta, MaxProp e na proposta de jornadas preditas baseadas em RNAs locais.

Na realização desses experimentos não houve controle do limite de banda e de fila de mensagens para permitir a análise sem interferência de outros fatores que poderiam alterar o número de mensagens entregues.

Para os resultados da estratégia proposta (RNA), a média gerada foi dos valores obtidos em cinco novos treinamentos consecutivos, como mostrado na Tabela 7. Essa tabela mostra a quantidade de mensagens entregues por cada estratégia em cada um dos nove dias avaliados.

Tabela 7: Número de mensagens entregues em vários dias (contatos reais).

Dia	EI	ED	MP	RNA		
				Min	Max	Média
1	683	303	388	326	352	340,2
2	917	341	354	426	441	434,8
3	981	383	371	416	483	443,4
4	977	392	410	396	451	431,2
5	904	369	351	411	425	418,6
6	652	330	304	343	368	353,8
7	1084	442	434	499	522	509,8
8	701	328	379	340	367	357,8
9	1637	509	696	574	634	613,4
Σ	8536	3397	3787	3731	4043	3903,8

(EI) Epidêmico Ilimitado, (ED) Entrega Direta, (MP) MaxProp e preditor de jornadas (RNA).

A estratégia baseada na RNA entregou mais mensagens do que o MaxProp em 6 de 9 dias (em preto na tabela), na média (3,09%) e

no máximo (6,76%). Em 5 desses dias a estratégia proposta superou o MaxProp com os valores mínimos do número de mensagens entregues.

Em relação à estratégia de entrega direta (ED), usado como referência inferior de desempenho, o MaxProp entregou 390 mensagens e a estratégia baseada na RNA entregou 507 mensagens, representando 39,62% mais mensagens entregues do que o MaxProp.

Mesmo existindo uma relação entre um grande número de mensagens entregues e um baixo erro de treinamento, não necessariamente o treinamento que apresentou menor erro apresentou um grande número de mensagens entregues. Alguma diferença pode acontecer por que os contatos de um novo dia tem baixa relação com os contatos nos registros históricos. De fato, um treinamento excessivo irá fazer a rede aprender exatamente o que é apresentado nos exemplos (*overfitting*), perdendo-se a capacidade de generalização, o que não é desejável. Do mesmo modo um treinamento pequeno não conseguirá capturar os padrões desejados (*underfitting*) para uma correta representação geral dos exemplos.

A Figura 16 apresenta o gráfico do número de mensagens entregues ao passar do tempo nas quatro estratégias avaliadas. Nota-se que na estratégia epidêmica sem limitações há copia de mensagens, pois pretende-se mostrar o limite máximo de mensagens entregues. Na entrega direta se observa o que foi considerado o limite mínimo para entrega de mensagens.

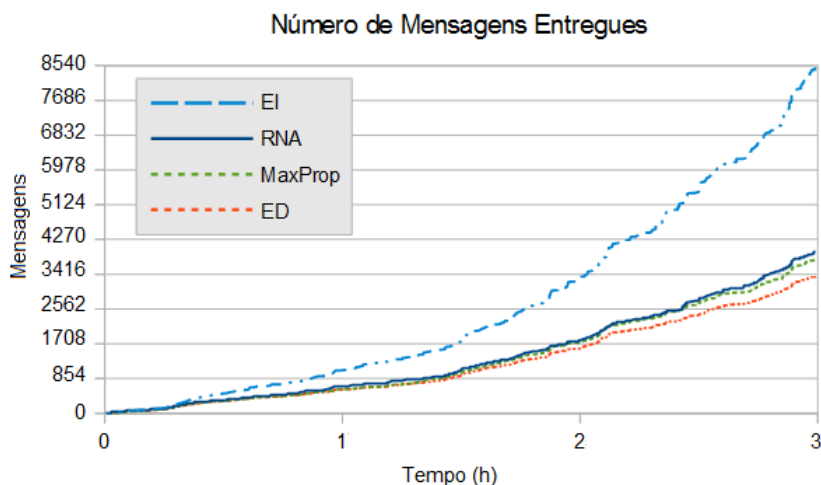


Figura 16: Mensagens entregues sem cópia de mensagem.

A diferença entre a estratégia proposta e o MaxProp parece muito pequena quando comparada com o limite máximo alcançado pelo epidêmico ilimitado. A grande diferença no número de mensagens entregues ocorre neste caso por haver muito recurso disponível na rede (tamanho das filas e largura de banda/duração de contato). De outra forma, para obter esse número de mensagens entregues, somente se tivesse conhecimento pleno sobre a topologia futura da rede, contudo em uma DTN com contatos oportunistas esta situação não é possível.

5.4.3 Avaliação do atraso na entrega das mensagens

O experimento a seguir tem por objetivo a realização de comparações do número de mensagens entregues por faixa de atraso na entrega das mensagens. São avaliados os casos das estratégias de roteamento epidêmica ilimitada, entrega direta, MaxProp e na proposta com RNAs locais. Este experimento permite verificar o quanto cada estratégia favoreceu a entrega da mensagem em menor tempo.

Para realizar esses experimentos, não houve controle de limitação de banda e de fila de mensagens para permitir a análise sem interferência de outros fatores que poderiam alterar o número de mensagens entregues.

Para a estratégia proposta, a média gerada foi dos valores obtidos em cinco novos treinamentos consecutivos.

O gráfico da Figura 17 mostra o número de mensagens entregues por faixa de atraso em intervalos de 10 minutos. Os valores representam o número total de mensagens entregues em nove dias avaliados.

É observado que a estratégia epidêmica ilimitada marca o limite de menor atraso na entrega de mensagens enquanto a entrega direta apresenta um limite de maior atraso, como seria esperado.

A estratégia proposta supera o MaxProp, contudo com pouca diferença. Nas faixas iniciais de atraso na entrega (até um pouco mais de 50 minutos), a estratégia proposta sempre apresenta maior número de mensagens entregues. Nas faixas intermediárias de atraso (até 150 minutos), a estratégia RNA proposta e o MaxProp alternam o maior número de mensagem entregues, mas com ligeira vantagem para a estratégia RNA proposta. No fim (acima de 150 minutos), a estratégia proposta também supera o MaxProp.

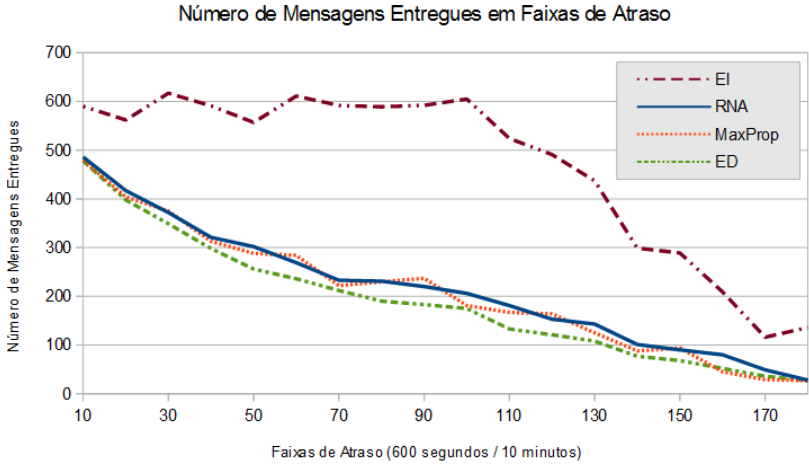


Figura 17: Atraso na entrega das mensagens em faixas de 10 minutos.

5.4.4 Avaliação do tempo de execução

A avaliação do tempo de execução foi realizada com o objetivo de comparar o desempenho das estratégias de roteamento avaliadas com os dados de contato reais.

A Tabela 8 apresenta o tempo (milissegundos) utilizado por cada estratégia em cada um dos nove dias avaliados. A soma dos tempos de execução obtido em cinco experimentos está apresentada na tabela. Ao final são apresentadas a soma de todos os dias para cada estratégia, a soma da quantidade de mensagens nas filas ao final do dia nos cinco experimentos e o tempo (microsegundos) gasto por cada mensagem da fila para se ter uma estimativa do tempo de execução por mensagem. Esse último valor é estimado, pois a fila de mensagens é constantemente atualizada durante a execução, tornando-se necessário determinar o tempo gasto em cada contato para processar a fila para se ter o tempo exato.

O tempo de execução para o treinamento não foi considerado pois trata-se de um processo *off-line*. Da mesma forma, o tempo para determinação dos valores da matriz de probabilidades do MaxProp também não foi considerado.

Observa-se na Tabela 8 que o MaxProp foi executado em menor tempo em 6 dos 9 dias avaliados. Esse resultado difere um pouco do observado com RNAs locais usando dados sintéticos, em que o Max-

Tabela 8: Tempo de execução (ms) (contatos reais).

Dia	EI	ED	MP	RNA
1	3075	350	775	750
2	1465	545	620	760
3	735	55	185	1065
4	2100	450	350	950
5	995	685	1090	1110
6	520	85	120	745
7	2115	440	975	995
8	1225	230	550	465
9	5320	1425	4000	2945
Tempo total	17550	4265	8665	9785
Msgs. filas	1642505	459175	457035	456390

(EI) Epidemico Ilimitado, (ED) Entrega Direta, (MP) MaxProp e Preditor de jornadas (RNA).

Prop havia sido superior em apenas 4 dias avaliados. Os três motivos principais para o aumento do tempo de execução são os seguintes:

- o aumento do número de neurônios: as RNAs locais são criadas dinamicamente de acordo com as informações do histórico. Devido ao maior número de nós e variações nas informações históricas, o número de neurônios das RNAs utilizadas nos experimentos com os dados do DieselNet é maior em relação aos das RNAs utilizadas nos experimentos com dados sintéticos, consequentemente aumentando o tempo de execução.
- o aumento das predições válidas: o número de consultas para as RNAs aumenta se as predições são válidas viabilizando a busca por nó em profundidade. Quando uma consulta é realizada com dados que não foram treinados, a possibilidade de retornar predições válidas diminui. Neste experimento as RNAs foram treinadas com contatos de 54 dias, cobrindo muitos instantes de tempo no intervalo de tempo definido para predição, aumentando as chances de obter contatos válidos.
- a diminuição do parâmetro *timeSlice* do algoritmo de construção de jornadas: o valor utilizado para a fatia de tempo foi ajustado para favorecer a obtenção de um próximo contato com um intercontato próximo (metade do o desvio padrão dos intercontatos

do histórico). Um valor maior de *timeSlice* diminui o número de predições de contato, conseqüentemente diminuindo o tempo de execução.

A análise do tempo de execução gasto pela estratégia de roteamento mostra a variação com o número de mensagens na fila. O MaxProp superou a estratégia proposta obtendo um menor tempo de execução por mensagem (13,08%).

5.4.5 Análise da influência das decisões de encaminhamento

A verificação das decisões de encaminhamento é realizada com o intuito de avaliar o número de decisões tomadas e o quanto estas estão de acordo com o encaminhamento ótimo, obtido através da execução da estratégia epidêmica ilimitada.

Uma decisão tomada é considerada correta se está de acordo com o encaminhamento ótimo obtido no caso do conhecimento prévio. Isso pode ser mais difícil de se obter dependendo do critério que deve ser otimizado na estratégia de encaminhamento. No caso desse trabalho o critério escolhido consiste em minimizar o atraso na entrega. Dessa forma, a comparação com o encaminhamento realizado pela estratégia epidêmica sem limitações pode ser realizado, pois nessa estratégia, a primeira mensagem entregue ao destino sempre será a de menor atraso.

O mesmo experimento usado para avaliar o número de mensagens entregues foi usado nesta avaliação.

Nesse experimento, foi avaliada a quantidade de decisões corretas e erradas de acordo com o observado na estratégia epidêmica ilimitada. As decisões foram classificadas como corretas se a decisão foi a mesma tomada pela estratégia epidêmica, caso contrário foram consideradas erradas. Uma decisão errada ainda pode ser classificada como errada sem transferência da mensagem ou classificada errada com transferência da mensagem.

A estratégia proposta sem cópia de mensagem foi avaliada e a média de cinco execuções mostra que ela realizou 25544 decisões, 13,75% menos decisões do que a estratégia do MaxProp (29561), tal como mostrado na Figura 18. Ou seja, a estratégia RNA proposta entregou um número maior de mensagens no mesmo tempo que tomou menos decisões de encaminhamento. Isso pode ser justificado avaliando-se que uma mensagem encaminhada baseada em uma decisão errada deve permanecer mais tempo na rede, ficando sujeita a mais decisões de encaminhamento.

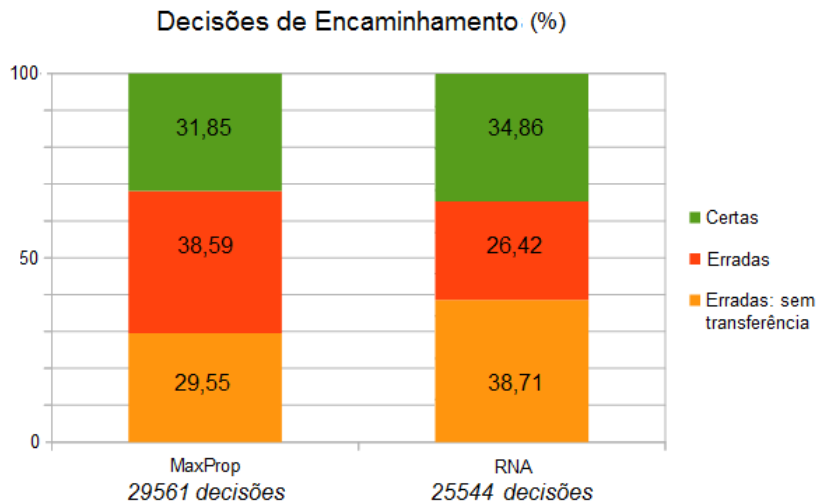


Figura 18: Quantidade de decisões de encaminhamento.

A estratégia baseada em RNA tomou mais decisões erradas (38,71%) que não promoveram a transferência da mensagem para outro nó, podendo-se dizer que tomou decisões mais conservadoras. As decisões mais conservadoras apresentam-se como uma opção mais interessante do que a transferência para um nó qualquer. Nesse contexto de contatos, a perda de uma oportunidade de transferência, mantendo a mensagem na fila de mensagem é melhor do que a transferência para um nó desconhecido (pois foi transferido erroneamente) sem chances conhecidas de encaminhamento da mensagem até o seu destinatário.

5.4.6 Conclusões

Nesta seção foram apresentados os resultados dos experimentos utilizando dados reais de contato sem realizar a replicação de mensagens. A estratégia de roteamento proposta utilizando RNAs locais foi avaliada, comparando com outras estratégias.

A estratégia proposta entregou um número maior de mensagens do que a estratégia do MaxProp e da entrega direta e ela também permitiu entregar mais mensagens em menor tempo. Ela gerou também um número menor de mensagens na fila de mensagens do que o MaxProp. Todavia, o tempo de execução do MaxProp foi menor na maioria

dos dias.

Os gráficos mostrando a entrega de mensagens e o atraso na entrega das mensagens pelas estratégias epidêmica ilimitada e entrega direta permitiram verificar os limites máximos e mínimos para comparações.

A estratégia de roteamento proposta tomou também mais decisões corretas proporcionalmente, evitando que a mensagem ficasse ocupando a rede e tomando tempo para outras decisões. A avaliação das decisões de encaminhamento ajudaram a mostrar que a decisão de manter a mensagem na fila no caso de pouca informação é melhor para o cenário avaliado com roteamento sem cópia de mensagens.

5.5 ANÁLISE DA INFLUÊNCIA DA REPLICAÇÃO

O objetivo dessa avaliação é determinar o desempenho das estratégias de roteamento mediante a variação do número de mensagens replicadas. São avaliados o número de mensagens entregues, atraso na entrega, mensagens nas filas e custo da entrega de mensagens para as quatro estratégias avaliadas.

Foram utilizadas versões da estratégia proposta e do MaxProp com controle de replicação de mensagens, sendo implementado o mesmo controle de replicação em ambas as estratégias. Foi também realizado o roteamento com replicação de mensagens até um limite determinado de cópias permitidas, sendo que a partir desse limite, o algoritmo de encaminhamento funciona como na versão sem cópia. A estratégia epidêmica ilimitada e de entrega direta continuam delimitando os limites superior e inferior das avaliações.

Os cenários para avaliação são os mesmos utilizados na seção anterior.

5.5.1 Avaliação da entrega de mensagens

Esta avaliação objetiva a comparação do número de mensagens entregues nas estratégias avaliadas utilizando dados de contatos reais. Avalia-se ainda neste estudo, a variação do número de cópias das mensagens.

A quantidade de mensagens entregues na estratégia do MaxProp é comparada com as quantidades mínimas e máximas de mensagens entregues obtidas em cinco execuções da estratégia proposta (RNA).

A Figura 19 mostra a quantidade de mensagens entregues para várias estratégias. A estratégia de entrega direta apresenta a pior entrega de mensagens e a estratégia epidêmica ilimitada apresenta o maior número de mensagens entregues.

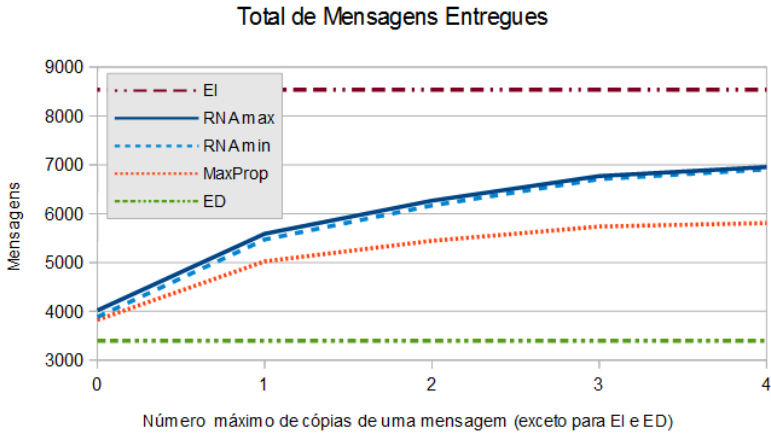


Figura 19: Número de mensagens entregues com variação do número de cópias da mensagem.

Na avaliação da estratégia sem cópia de mensagens, o número de mensagens entregues pela estratégia proposta é um pouco maior do que o obtido pelo MaxProp, conforme foi mostrado na seção anterior. A estratégia proposta com limite de cópias (de 1 a 4) acentua esta superioridade para o número de mensagens entregues. Observa-se ainda que há uma tendência das estratégias com valores mínimos (RNA min) e máximos (RNA max) convergirem para um mesmo valor com o aumento do número de cópias da mensagem.

Já quanto ao tempo de execução, foi observado um aumento de aproximadamente três vezes com a utilização de 4 cópias em comparação a versão sem replicação. Pouca variação foi encontrada em novos treinamentos.

5.5.2 Avaliação do atraso na entrega das mensagens

Esta avaliação objetiva comparar o desempenho das estratégias de roteamento em termos do tempo de entrega das mensagens.

O atraso de entrega das mensagens foi avaliado utilizando a versão com cópia de mensagem fixada em 4 cópias e também com a versão sem cópia.

Como pode ser observado na Figura 20, a estratégia baseada em RNA e o MaxProp apresentam duas curvas (min e max), uma para o caso sem cópia e outra para 4 cópias da mensagem, mostrando o aumento do número de mensagens entregues em todas as faixas de atraso em relação ao caso sem replicação.

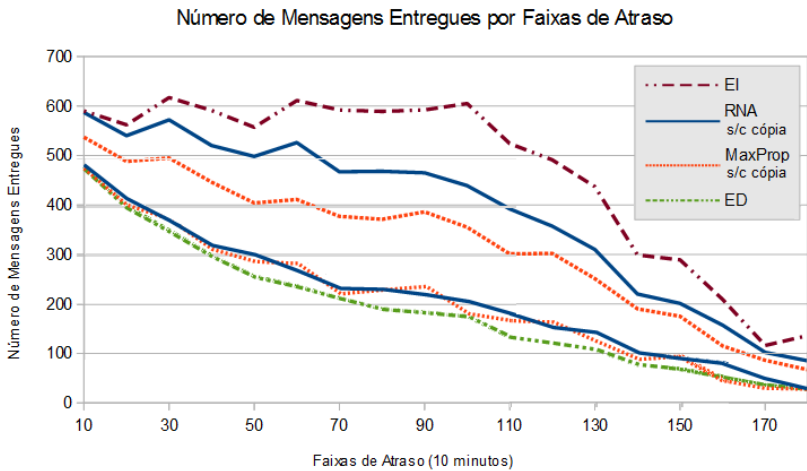


Figura 20: Número de mensagens entregues por faixas de atraso na entrega.

Verifica-se que tanto a versão sem replicação quanto a com replicação permitiram alcançar o objetivo de minimizar o atraso de entrega estabelecido no critério do preditor de jornadas.

5.5.3 Avaliação do número de mensagens geradas

Uma métrica importante de ser avaliada é o número de mensagens geradas na rede ou carga de ocupação da rede. O objetivo dessa avaliação consiste em verificar a ocupação das filas de mensagens e conseqüentemente a ocupação da rede variando-se o número de cópias permitidas para as mensagens.

Na Figura 21 observa-se que a estratégia epidêmica ilimitada gerou três vezes mais mensagens do que as outras estratégias e a entrega

direta é apresentada como uma referência de desempenho inferior. As curvas que representam os valores mínimos e máximo de mensagens na rede para a estratégia RNA proposta praticamente se sobrepõe para todo o número de cópias. Observa-se ainda que a quantidade de mensagens gerada na rede vai aumentando com o número de cópias.

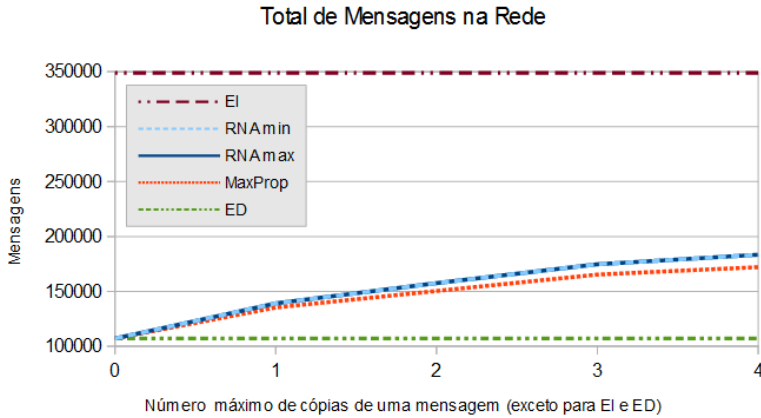


Figura 21: Ocupação da rede com variação do número de cópias da mensagem.

A estratégia baseada na RNA gerou uma quantidade um pouco maior de mensagens do que o MaxProp, o que poderia influenciar o resultado. Contudo, ao comparar a diferença entre as mensagens entregues e geradas para ambas as estratégias, foi verificado que a diferença das mensagens geradas é muito pequena para ter ocasionado a grande diferença nas mensagens entregues.

5.5.4 Avaliação do custo de entrega das mensagens

Esta avaliação tem por objetivo determinar o custo em termos de mensagens na rede para cada mensagem entregue. Assim, pode ser verificado se a estratégia de roteamento está consumindo uma grande banda de rede e espaço nas filas de mensagens em relação ao número de mensagens que está conseguindo entregar.

A Figura 22 apresenta curvas de custo de entrega, mostrando quantas mensagens estão presentes na rede para cada mensagem en-

tregue. São avaliados os diversos casos com uma quantidade máxima permitida de cópias da mensagem e o caso sem cópia de mensagem, no qual somente a mensagem original é encaminhada pela rede.

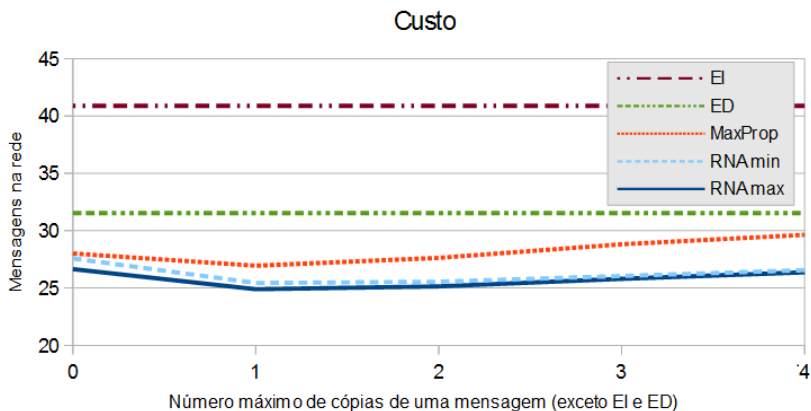


Figura 22: Custo para cada mensagem entregue.

Pode-se observar que a estratégia epidêmica ilimitada e a entrega direta apresentam os dois maiores custos. Todas as estratégias tiveram menor custo do que o epidêmico ilimitado do ponto de vista desta métrica, pois a estratégia epidêmica ilimitada consome muito recursos da rede para conseguir entregar mensagens no limite máximo (ótimo).

A estratégia proposta, tanto no seu limite inferior de entrega de mensagens (RNA min) e seu limite superior (RNA max), superam o MaxProp em termos do custo. Observa-se que RNA min apresenta maior custo do que RNA max sem cópia de mensagens e eles vão convergindo para um mesmo valor com o aumento do número de cópias. Tanto para a estratégia proposta quanto para o MaxProp, o custo de entrega sem cópia de mensagens é maior do que o custo utilizando uma cópia da mensagem, sendo este um limite de melhor desempenho. Na estratégia proposta, o limite de 4 cópias leva a um custo aproximado ao obtido sem cópia de mensagens.

5.5.5 Conclusões

Observa-se a partir dos resultados anteriores que o aumento do número de cópias da mensagem permite aumentar as chances de entrega da mensagem.

Os resultados confirmam que a implementação do preditor de jornadas com o critério de diminuição do atraso de entrega mostrou-se adequada, permitindo a estratégia RNA proposta entregar mais mensagens em menor tempo. Isso foi evidenciado com o aumento do número de cópias.

Com 4 cópias da mensagem, a estratégia proposta conseguiu entregar 15,94% mais mensagens do que o MaxProp, mas aumentou a carga da rede em 6,94%. A estratégia do MaxProp gerou um número um pouco menor de mensagens na rede, mesmo usando a mesma estratégia para replicação das mensagens utilizada na estratégia proposta. A estratégia RNA proposta diminuiu 1,12% o custo de mensagens na rede por mensagem entregue relação a sem cópia. Já com estratégia do MaxProp, esse custo aumentou 7,35%. Esses resultados apontam a superioridade do método proposto para o cenário avaliado.

Observa-se que ao utilizar um número maior de cópias, as RNAs com melhor (RNA max) ou pior (RNA min) o número de mensagens entregues tende a convergir o custo, tal como ocorrido com o número de mensagens entregues. Assim, os resultados apontam que o aumento do número de cópias da mensagem suprime a diferença de custo e taxa de entrega obtidas a partir de treinamentos diferentes para a estratégia de roteamento proposta. Isso pode eliminar a necessidade de identificar treinamentos de melhor desempenho para melhorar a estratégia de roteamento baseada em RNA.

O tempo de execução da estratégia proposta utilizando no máximo quatro cópias foi aproximadamente três vezes maior do que o sem cópia de mensagens.

5.6 ANÁLISE DA INTRODUÇÃO DO CONTROLE DA FILA DE MENSAGENS E DURAÇÃO DA CONEXÃO

Conforme apresentado anteriormente, a avaliação das estratégias sem influência de limitações impostas pelo tamanho da fila de mensagens e da duração dos contatos é importante para isolar fatores que interferem nos resultados e na análise do desempenho das estratégias avaliadas. Os resultados dos experimentos realizados nesta seção visaram comparar o comportamento das estratégias diante a limitação das filas de mensagens e duração das conexões.

Foram realizados experimentos com o controle do tamanho máximo permitido para a fila de mensagens e da duração do tempo de contato, pois essas condições modificam o desempenho da rede.

5.6.1 Avaliação da entrega de mensagens com tamanhos diferentes das filas de mensagens

Um experimento foi realizado para avaliar a entrega de mensagens variando-se o tamanho máximo das filas de mensagens dos nós e com limite de 4 cópias por mensagem.

A Figura 23 mostra o comportamento da entrega de mensagens diante dos tamanhos das filas de mensagens em: 100, 500, 1000, 2650 e 5300 KBytes. O tamanho das mensagens permaneceu em 1 KByte, conforme os experimentos anteriores. A estratégia epidêmica teve a fila de mensagem limitada por razões evidentes de implementação, mas os outros recursos foram mantidos ilimitados.

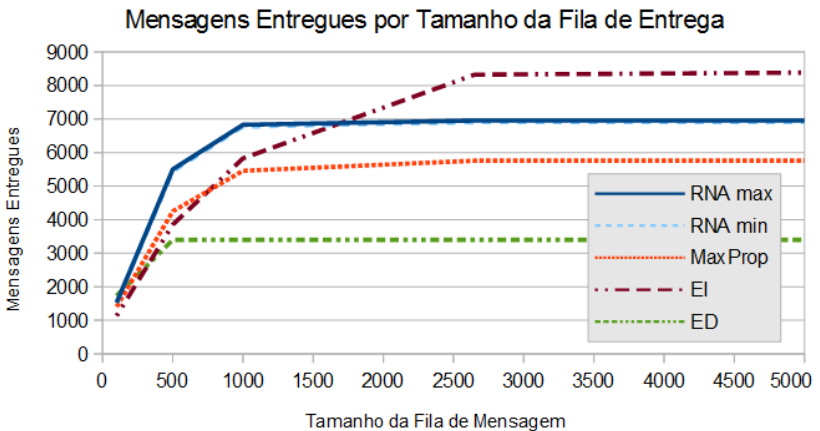


Figura 23: Número de mensagens entregues variando o tamanhos variados da fila de mensagem.

Observa-se que a estratégia proposta é superior na entrega de mensagens, inclusive superando a estratégia epidêmica até o tamanho da fila de mensagens de 1600 KBytes. Verifica-se que a estratégia epidêmica apenas limitada pelo controle do tamanho da fila somente começa a ser bem melhor que as outras estratégias quando possui espaço suficiente nas filas de mensagens para se aproveitar da replicação realizada.

A partir desses resultados é possível determinar um limite máximo para o tamanho das filas de mensagens dependendo da estratégia de roteamento, do número de mensagens geradas e do cenário. No cenário em questão, com uma mensagem gerada por nó para cada ou-

tro nó a cada 900 segundos, a estratégia RNA proposta pode entregar mais mensagens do que o MaxProp com a fila de mensagens de 2700 KBytes. Para a mesma quantidade de máxima de mensagens entregues pela estratégia do MaxProp, a estratégia RNA proposta precisaria em torno de 700 Kbytes de tamanho da fila. Isso mostra que há desempenho superior na entrega de mensagens com menos espaço para filas de mensagens com a estratégia proposta baseada em RNA.

5.6.2 Avaliação da entrega de mensagens com variação da duração dos contatos

Foi realizado um experimento para avaliar o desempenho das estratégias diante da variação da duração dos contatos.

Esse experimento adota a duração real do contato obtida dos dados do DieselNet e utiliza o número máximo de 4 cópias por mensagem. O tamanho da fila de mensagens é considerado ilimitado para não interferir na avaliação. O enlace é configurado para permitir no máximo a transferência de 1 Mbps em cada direção da transmissão em comunicação simultânea, devido ao compartilhamento do enlace.

A soma do número de mensagens entregues nos 9 dias avaliados para cada estratégia é mostrado na Figura 24.

Verifica-se que a estratégia proposta apresenta a maior quantidade de mensagens entregues, inclusive maior do que a obtida com a estratégia de epidemia limitada pela duração do contato. Comparando com o número de mensagens entregues no caso sem limitação da duração de contato e do tamanho da fila de mensagens (Figura 19), a estratégia epidêmica entregou 87,04% do total de mensagens geradas, enquanto a estratégia RNA proposta 83,91% e o MaxProp 81,11%. Isso é devido ao fato que muita banda é utilizada para tentar transmitir todas as mensagens da fila de mensagem para o nó em contato, sendo que com o limite na duração do contato, muitas mensagens acabam não sendo transferidas.

5.6.3 Conclusões

As condições de limitação do tamanho das filas de mensagens e de duração dos contatos mudam o desempenho da rede, pois:

- nas filas de mensagens cheias, mensagens são eliminadas: ou as mais antigas, ou novas mensagens que tentam entrar na fila, ou

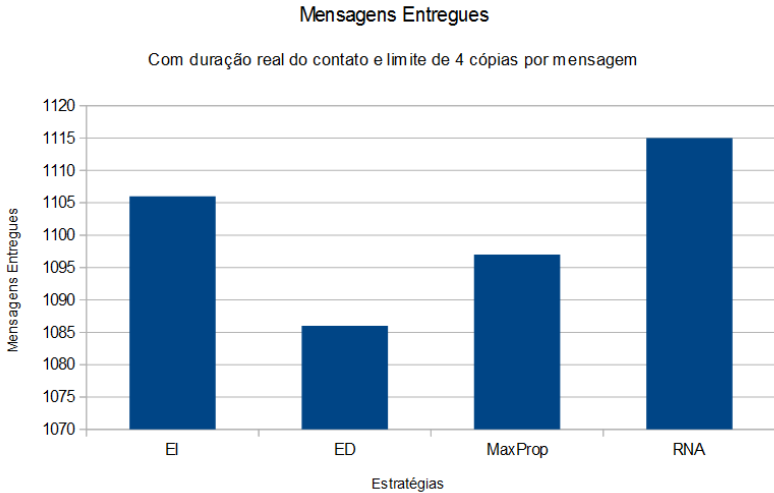


Figura 24: Número de mensagens entregues com duração real dos contatos.

mensagens com alguma priorização inferior;

- para contatos de duração limitada, nem todas as mensagens podem ser transferidas: a priorização das mensagens a serem enviadas também pode ser aplicada para a entrega, transferência ou cópia das mensagens para o nó em contato.

Nos experimentos realizados, para todas as estratégias avaliadas, foi priorizado o descarte das mensagens que estavam mais tempo na fila. Para a entrega e transferência das mensagens foram priorizadas as que estavam a menos tempo na fila.

As limitações do tamanho da fila e da duração do contato (ou banda de rede) fazem com que o aumento do número de mensagens entregues não obrigatoriamente diminua o atraso na entrega das mensagens. Isso pois, a estratégia de descarte da fila de mensagens e priorização para envio das mensagens influenciam no critério a ser melhorado no roteamento.

A avaliação dos experimentos apontam para a superioridade da versão com replicação da estratégia proposta utilizando RNAs locais em condições reais de operação do roteamento, havendo limitações das filas de mensagens e duração dos contatos comparando-se com as outras

estratégias avaliadas.

5.7 ESTUDO DE AJUSTES NA RNA DO PREDITOR DE CONTATOS

Para alcançar os resultados satisfatórios apresentados foram estudadas e aplicadas melhorias. Estas, aos poucos foram trazendo pequenos avanços até se alcançar o preditor de jornadas final e sua versão com replicação. Nesta seção são apresentadas essas melhorias.

5.7.1 Filtragem das informações para treinamento

A seleção de dados de entrada é uma forma de melhorar o desempenho do preditor. Assim, pretende-se remover a interferência de informações prejudiciais para a predição.

Uma forma de selecionar as informações para treinamento a serem aprendidas pela RNA é ter alguma informação antecipada e usá-la para filtrar as informações de treinamento. Por exemplo, um ônibus pode usar um processo de filtragem para manter apenas os contatos do histórico passado dos dias nos quais houver um grau elevado de coincidência com os contatos já realizados no ciclo atual. Assim, com menos interferências dos dias com contatos em situações diferentes é possível aumentar o acerto nas predições e também realizar treinamentos mais rápidos, pois há diminuição da quantidade de informações para o treinamento, além de favorecer a sua convergência para os valores desejados.

A teoria da informação é uma opção para modelar esse filtro, pois fornece meios de quantificar as informações e realizar comparações de duas fontes de dados. A distância de Kullback-Leibler (COVER; THOMAS, 1991) apresenta uma forma de medir a diferença entre estas.

A comparação de duas séries de eventos e o estabelecimento de uma distância de acordo com a diferença entre esses eventos é uma medida para estabelecer o filtro em questão. A distância de Kullback-Leibler, também chamada de entropia relativa, é uma medida da distância entre duas distribuições de probabilidade que expressa-se por:

$$D(p||q) = \sum p_i * \log\left(\frac{p_i}{q_i}\right) = \sum p_i * \log\left(\frac{1}{q_i}\right) - p_i * \log\left(\frac{1}{p_i}\right) \quad (5.1)$$

Ela leva em consideração as distribuições de probabilidades de p e q , nas quais p_i e q_i indicam as probabilidades do evento i de uma variável aleatória discreta das distribuições p e q . O valor indica quanto mais próximas forem p e q e quantifica a diferença na quantidade de informação.

Para a filtragem que se deseja estabelecer neste trabalho, p_i e q_i indicam respectivamente a probabilidade do nó de referência no ciclo atual e no ciclo passado se encontrar com o nó i . Para cada nó de referência, a soma do valor absoluto das distâncias de Kullback-Leibler $\sum |p_i * \log(\frac{p_i}{q_i})|$, onde i é o nó contatado, e p e q são as distribuições de probabilidade de contato em dois dias (aqui vistos como ciclos), é usado como a distância para determinar os dias com contatos mais parecidos.

Resultados

Para melhorar o desempenho da RNA global, um processo de filtragem de dados baseado na distância de Kullback-Leibler foi elaborado e aplicado para diminuir as interferências com dias que tiveram contatos muito diferentes, o número de dias de contatos a serem usados no treinamento e o tempo de treinamento.

Considerando-se a identificação de um nó de referência e conforme determinado pela Equação 5.1 foram desenvolvidos programas para realizar a classificação de todos os ciclos e foram separados aqueles parecidos com o dia a ser testado.

A RNA global foi treinada com as informações provenientes desse filtro, mas não foram obtidas vantagens consistentes, pois a taxa de acerto do preditor de contatos e o número de mensagens entregues pela estratégia de roteamento baseada em RNA foram parecidos com os obtidos em experimentos anteriormente realizados. Contudo, ao realizar testes com o MaxProp pode-se verificar uma pequena melhora no número de mensagens entregues. Todavia é interessante tratar essa estratégia de filtro já com as RNAs locais.

Entretanto, apesar de não ter melhoria do ponto de vista da entrega de mensagens, esta filtragem diminui a quantidade de informações necessárias para o processamento e o tempo de treinamento das RNAs. Neste sentido, o ensaio teve sucesso uma vez que se diminui o número de exemplos e se obtém resultados similares.

5.7.2 Separação da RNA por informação de saída

Em uma das modificações realizadas para melhorar o acerto das predições de contato foi realizada a divisão das saídas desejadas. Dessa forma o próximo nó de contato predito e o próximo momento de contato predito foram separados, criando-se uma RNA para prever cada uma dessas informações, isolando assim as predições.

Para a predição do próximo nó de contato as entradas da RNA permanecem as mesmas. Na RNA para predição do próximo momento de contato é adicionado o próximo nó de contato como entrada.

O custo de espaço de armazenamento da RNA e dos pesos é uma desvantagem, pois há informação repetida, apesar de as RNAs ficarem menores do que a global. A programação de duas RNAs com criação dinâmica também requer um maior trabalho e atenção, pois há informações repetidas para se tratar.

Resultados

A RNA foi dividida a partir das saídas, uma para cada saída desejada. Essa modificação não foi suficiente para obter resultados satisfatórios no desempenho do roteamento das mensagens e não foi confirmado o aumento ou a diminuição da taxa de acerto.

Como previsto, a replicação da estrutura resulta no aumento do espaço de armazenamento para os pesos das RNAs. Esse aumento não é o dobro, pois o número de neurônios nas camadas intermediárias e de saída diminuíram.

As vantagens encontradas foram um maior controle das predições e uma pequena diminuição do tempo de treinamento. Além disso, esta modificação facilita a criação automática e a avaliação das RNAs pela ferramenta *workbench* do Encog, permitindo realizar ensaios mais rápidos e menos sujeitos a erros, pois essa ferramenta não permite trabalhar automaticamente com RNAs que apresentam mais de uma saída.

A separação das saídas foi mantida para os experimentos posteriores.

5.7.3 Predição de jornadas com RNA global ou local

Uma questão relevante para melhorar a estratégia proposta consiste em determinar se é possível gerar uma RNA global que seja operacional em termos do tempo de treinamento e qual topologia possibilita

a extração de padrões globais localmente úteis.

O desafio neste caso é criar uma RNA que possa aproveitar uma grande massa de informações de entrada e encontrar relações entre os dados de todos os nós da rede (globais) que podem ser usadas localmente. Um problema a ser tratado é a interferência gerada por dados que não são úteis para o objetivo da predição.

Uma alternativa consiste em conceber uma versão do preditor com RNAs locais, onde cada nó mantém sua própria RNA e a obtenção das RNAs de outro nós é feita através de um protocolo de roteamento. Neste caso, se um nó deseja saber qual a decisão para outro nó, basta utilizar a RNA do nó em questão com os seus respectivos pesos.

Na abordagem com RNAs locais cada nó deve obter a RNA dos outros nós com os quais historicamente manteve contato, além de manter a RNA do próprio nó. Cada RNA é treinada pelo seu respectivo nó com informações de contato referentes a esse nó.

Resultados

Os experimentos realizados com RNA global mostraram resultados satisfatórios na entrega de mensagens com o conjunto de dados sintéticos, porém quando utilizado na estratégia de roteamento com os dados baseados no DieselNet não foi possível afirmar a superioridade da abordagem proposta, pois havia muita variação no resultado em treinamentos diferentes. Treinamentos que favoreciam a generalização apresentavam muita variação em ensaios diferentes e treinamentos mais prolongados apresentavam taxas de entrega insuficientes.

A arquitetura das RNAs locais utilizada no preditor de jornadas proposto apresentou melhores resultados do que a RNA global, aumentando o número de mensagens entregues em relação a RNA global. Conforme apresentado nos capítulos anteriores, ambas as arquiteturas de RNA (global e local) foram modeladas, implementadas no simulador e tiveram seus resultados apresentados. Todavia, ainda é possível explorar uma topologia de RNA global diferente (entradas, camadas e número de neurônios diferentes) para obter melhores resultados. A aplicação de filtragem nos dados de treinamento é uma opção interessante para lidar com as interferências de outros dados sem deixar de aproveitar informações globais que venham a ser úteis.

Outra forma de tentar reduzir as interferências foi através da modelagem de redes recorrentes, contudo os testes iniciais não apresentaram taxas de acerto satisfatórias.

5.7.4 Relação com o nó de contato anterior

Na concepção inicial da proposta desse trabalho é pressuposto existir e extrair relações entre o nó atual (nó de referência), o último nó que o contactou e próximo nó que o contactaria. Dessa forma, as primeiras arquiteturas de RNA propostas levam em consideração na sua entrada o nó de contato anterior, o próximo nó de contato e os respectivos instantes em que esses contatos ocorrem. O objetivo era obter a relação da sequência de nós em contato com um horizonte de predição de dois saltos.

Contudo, o estudo sobre o conjunto de dados do DieselNet mostra uma relação muito baixa entre esses nós. Isso foi avaliado ao verificar uma grande quantidade de grupos diferentes formados ao identificá-los pela sequência de nós e ao verificar os instantes de contato substancialmente diferentes em contatos do mesmo grupo. Além disso, foi realizada a verificação da significância das variáveis de entradas nas saídas da RNA (ambos com algoritmos do Weka). Por esta razão, o pressuposto inicial da existência de relação entre três nós em sequência (nó de referência com o nó anterior e próximo nó) foi trocado pelo pressuposto de relação entre apenas o nó de referência e o próximo nó de contato. Todavia, manteve-se a informação do instante do contato anterior e o próximo contato.

Com isto, a estrutura de RNA criada fica mais simples e o treinamento necessitará apenas modelar as relações entre o nó de referência, o próximo nó de contato e um momento no tempo.

Resultados

A suposição da existência de relação entre contatos consecutivos entre ônibus do STPU não estava se confirmando nos dados reais, pois a relação que estava sendo modelada entre as entrada e saída das RNAs avaliadas não estava fornecendo taxas de acerto que melhorassem o número de mensagens entregues.

Modificações na entrada das RNAs foram feitas removendo o nó de contato anterior, mas mantendo a informação do instante desse contato permitiram alcançar resultados satisfatórios na entrega de mensagens com os dados reais. A consistência desse resultado foi obtida observando que eles se repetiram em treinamentos diferentes com pouca variação.

5.7.5 Treinamento e Codificação da Entrada

A busca por melhores taxas de acerto para o preditor de contatos, além da utilização de outras topologias de RNA, levou a experimentar outros algoritmos de treinamento e avaliar outros critérios de parada do treinamento.

Testes buscando melhorar os resultados também foram realizados com a codificação das informações de entrada das RNAs utilizadas.

Resultados

Os primeiros experimentos utilizavam o algoritmo de aprendizagem de retropropagação. Os ensaios com outros algoritmos de treinamento apontam melhorias no tempo de treinamento com a utilização do algoritmo de retropropagação resiliente. Além disso, os critérios de treinamento foram aprimorados, conforme descrito na seção 4.4.2. Essas modificações permitiram obter ganho na estabilidade das respostas das predições a partir de treinamentos diferentes e também permitiram melhorar um pouco a taxa de acerto.

A codificação das informações de entrada da RNA também foi melhorada. Ficou evidente que a utilização de um neurônio para identificar cada classe ou identificador de uma informação de entrada permite melhorar o resultado da predição de contatos. O programa Encog oferece duas formas de codificar esse tipo de entrada: i) código "OneOf": cada valor diferente no conjunto de treinamento é mapeado em um neurônio diferente na entrada; ii) código "Equilateral": cada valor diferente da entrada é mapeada em um código binário de acordo com a distância de *Hamming*, ordenado conforme a frequência que esse valor se repete.

Alguns experimentos foram realizados discretizando os valores de momento do contato, contudo não foram continuados pois aumentaram muito o número de neurônios das RNA e não apresentaram vantagens consistentes.

5.7.6 Fatia de tempo na construção de jornadas: *timeSlice*

No algoritmo de construção das jornadas preditas existe um parâmetro que controla o passo no tempo *timeSlice* para realizar uma próxima consulta à RNA. O objetivo é permitir que se obtenha um contato predito que ocorra em um tempo à frente do momento atual.

O algoritmo de construção de jornadas pode então estabelecer uma seqüência temporal de contatos com um determinado nó e então adicioná-lo ao grafo de jornadas preditas como mais uma opção de encaaminhamento.

Nos primeiros testes o valor do *timeSlice* foi definido de acordo com um número fixo de interações considerando o tempo restante para o final do ciclo. Uma melhoria obtida no algoritmo de construção de jornadas foi determinada com um valor de *timeSlice* baseado nos intercontatos (tempo que vai do fim de um contato até o início do próximo contato em um nó) extraídos do histórico de contatos.

Resultados

Os experimentos realizados com valores ajustados de *timeSlice* permitiram que o número de mensagens entregues fosse aumentado até o limite obtido com o melhor conjunto de treinamento. Ou seja, os ensaios com treinamentos piores foram melhorados, mas não ultrapassaram o que foi obtido com o melhor treinamento. Esse resultado foi importante, pois permitiu dar mais estabilidade para a estratégia proposta e aumentou o número de mensagens entregues na média.

Verificou-se também que quando é permitida a replicação de mensagens na estratégia de roteamento, pode-se atribuir o valor do desvio padrão para o *timeSlice* sem dividi-lo por dois (conforme explicado na seção 3.4.2), diminuindo o tempo de execução e obtendo-se praticamente o mesmo número de mensagens entregues com limite máximo de cópias de mensagem igual a 4.

Na tentativa de melhorar o desempenho do roteamento foram feitos ensaios com um *timeSlice* dinâmico, supondo-se saber antecipadamente quais seriam os intercontatos realizados. Estes testes não mostram diferença significativa no número de mensagens entregues em comparação ao obtido através do cálculo do desvio padrão baseado em todo o histórico (desconsiderando o dia testado).

5.7.7 Conclusão sobre os ajustes

Foi imprescindível para a melhoria do desempenho do preditor de contatos o estudo e aplicação dos ajustes na estrutura das RNAs e no seu treinamento. Neste sentido, não relacionar o nó de contato anterior nos experimentos utilizando os dados de contato reais trouxe aumento significativo na quantidade de mensagens entregues.

A separação das RNAs por nós (locais) diminuiu a complexidade dos dados de contatos a serem tratados e isso contribuiu para aumentar o número de mensagens entregues. Já com a separação em saídas diferentes não foi possível determinar melhoria no desempenho das predições, contudo facilitou a realização de testes com a ferramenta de *workbench* do Encog permitindo automatizar algumas tarefas.

O treinamento realizado pelo algoritmo de retropropagação resiliente diminuiu o tempo de treinamento das RNAs. A melhoria da definição dos critérios de parada do treinamento permitiu automatizar o processo de treinamento, sem necessitar se apoiar em análises dos gráficos de evolução do treinamento.

É preciso acrescentar que existe espaço ainda para mais estudos que venham tirar proveito das informações globais, tal como pretendido com o uso da RNA global. Neste sentido, a filtragem das informações a serem utilizadas no treinamento baseada nos conceitos da teoria da informação é um estudo que pode ajudar.

A correta parametrização do construtor de jornadas e integração com o preditor de contatos também é importante para se obter melhores resultados.

5.8 CONCLUSÕES

A avaliação dos resultados dos experimentos apontam para a superioridade em determinadas métricas e condições da estratégia proposta de roteamento em DTN no contexto do STPU .

A estratégia proposta alcançou mais mensagens entregues do que o MaxProp utilizando RNA global com contatos sintéticos. Nesta situação apresentou um tempo de execução um pouco menor do que o MaxProp.

A RNA global não foi apresentada com experimentos usando dados de contatos reais, pois as RNAs locais apresentaram melhores resultados em testes preliminares.

As informações de entrada das RNAs foram modificadas para melhorar a representação dos dados de entrada a fim de melhor relacionar as informações a serem modelada pela rede neural. Dessa forma, utilizando-se RNAs locais foi possível usar os dados de contatos reais do DieselNet e obter melhoria no número de entrega de mensagem e atraso de entrega em relação ao MaxProp. O critério de diminuição do atraso de entrega foi favorável a proposta.

O tempo de execução não foi melhorado utilizando as RNAs

locais com os dados reais, ficando piores do que os apresentados pelo MaxProp. Além disso, a entrega de mensagens com replicação quase triplicou o tempo de execução em relação aos obtidos com o MaxProp.

A versão com replicação da estratégia proposta também permitiu aumentar o número de mensagens entregues e diminuir o atraso na entrega. Houve um pequeno aumento no número de mensagens geradas na rede em comparação ao MaxProp aumentando o número de cópias permitidas das mensagens.

Com limitação do tamanho da fila de mensagens e da duração dos contatos, a estratégia proposta se manteve superior, inclusive sendo superior a estratégia epidêmica.

6 EM DIREÇÃO A IMPLEMENTAÇÃO DA PROPOSTA DE ROTEAMENTO

No sentido de preparar a implementação da estratégia de roteamento proposta, este capítulo inicia com uma introdução aos conceitos gerais de que devem nortear uma arquitetura de DTN e apresenta as principais arquiteturas DTN existentes. Na seção seguinte são apresentados os detalhes da arquitetura DTN da *Internet Research Task Force* (IRTF) de maior interesse para esse trabalho, focando na arquitetura de roteamento.

Uma proposta de arquitetura de suporte focada nos preditores propostos neste trabalho (baseados em RNAs) e um modelo para implementação desta são apresentadas. A seguir é definido um protocolo de comunicação e discutido o que ainda é necessário para a implementação da estratégia proposta.

6.1 INTRODUÇÃO

Nas redes DTN, uma possível arquitetura pode ser vista como uma rede sobre outras redes (redes regionais) e uma sobreposição sobre a camada de transporte dessas redes, possibilitando a transmissão de informações mediante longos atrasos na comunicação. Assim, podem ser supridas necessidades de comunicação das redes clássicas, tais como manter a comunicação em dispositivos que podem falhar mediante falta de energia ou perda de sinal de radiofrequência (FALL, 2003).

Segundo Zhang (2006), uma arquitetura para redes DTN deve prover uma série de serviços essenciais tais como:

- roteamento sem conectividade contínua;
- armazenamento de pacotes e encaminhamento dentro da região;
- encaminhamento autenticado;
- esquema de nomes provendo interoperabilidade.

A ideia da manutenção da comunicação sem conectividade contínua está previsto no modelo RM-OSI de uma forma simples através da sincronização entre pontos de transferência da camada de sessão.

No modelo proposto pelo *DTN Research Group* (DTNRG) do IRTF é definida uma camada intermediária orientada a mensagem entre a camada de transporte e a camada de aplicação para tratar o

problema da comunicação em DTN (CERF et al., 2007) tal como apresentado no trabalho de Fall (2003). O DTNRG disponibiliza também uma implementação da sua arquitetura chamada DTN2, mas outras implementações também podem ser citadas, tais como a IBR-DTN e a DTNLite (DOERING et al., 2008).

Uma arquitetura estendida da proposta do DTNRG, chamada *Tetherless Communication Architecture* - (TCA) (LEGUAY, 2007, *apud Seth et al*) provê persistência através das conexões usando uma noção de estado de sessão entre origem e destino quando ocorre a interrupção da conexão. Além disso, provê suporte a mobilidade através de formas de localização de nós para que eles sejam alcançáveis a qualquer tempo.

Além desse, o projeto europeu Huggle¹ desenvolveu uma arquitetura que utiliza três métodos para superar o problema de intermitência na comunicação. Através dessa arquitetura é possível transmitir e receber dados das seguintes formas: uma infraestrutura existente; ou através de nós vizinhos no modo *ad hoc*; ou ainda ter os dados transportados através de dispositivos móveis de outros usuários, no modo "*messaging ferry*". Assim, tomando vantagem das conexões oportunistas, a ideia é que os dados possam ser transferidos através de objetos que as pessoas usam no seu dia a dia.

Dessas arquiteturas, a proposta pelo DTNRG é a mais completa e referenciada, possuindo muita documentação para consulta e implementação. Neste sentido a implementação da estratégia de roteamento proposta é baseada nesta arquitetura.

6.2 ARQUITETURA DTN DO IRTF

Na arquitetura proposta pelo DTNRG do IRTF, objetiva-se manter interoperabilidade com a arquitetura Internet baseado no TCP/IP. Nela é empregado um mecanismo de armazenamento salto-a-salto (*hop-by-hop*) e retransmissões como uma camada de transporte sobreposta, o SCF. Conforme explicado anteriormente no Capítulo 2, no SCF, um *host* intermediário deve manter o pacote em seu *buffer* e esperar uma oportunidade de encaminhá-lo, caso não seja possível fazê-lo logo que o recebe. Nessa oportunidade de encaminhamento deve-se selecionar criteriosamente o próximo *host* de forma que favoreça a entrega com o melhor desempenho possível para o destino, afastando a possibilidade

¹Projeto de pesquisa *An innovative Paradigm for Autonomic Opportunistic Communication* (<http://www.huggleproject.org/>) fundado pela *European Commission* (EC) no 6o *Framework* do programa *Information Society Technologies* (IST) (entre Janeiro de 2006 e Junho de 2010).

de atrasar infinitamente essa entrega.

Do ponto de vista geral, essa arquitetura consiste na comutação virtual de mensagens usando operações do SCF sobre a camada de transporte de qualquer modelo de intercomunicação. Nesta nova camada, chamada de Camada de Agregação (*Bundle Layer*, BL), é processado o Protocolo de Agregação (*Bundle Protocol*, BP) e a estratégia de roteamento de agregados (*Bundles*). As especificações do funcionamento dessa arquitetura estão descrito nas *Request For Comments* (RFC) 4838 e 5050 de 2007.

Nessa arquitetura, a unidade de dado da Camada de Agregação é definida como um conjunto de mensagens que devem ser entregues juntas e pode ser denominada mensagem, pacote ou agregado. O agregado não tem tamanho fixo e pode ser arbitrariamente grande (ZHANG, 2006).

A Figura 25 mostra as camadas e a atuação do *gateway* DTN com redes de tecnologias diferentes estabelecidas como regiões diferentes. Além disso, demonstra sua interação com as camadas de aplicação e de transporte e suas respectivas unidades de dados organizadas na hierarquia.

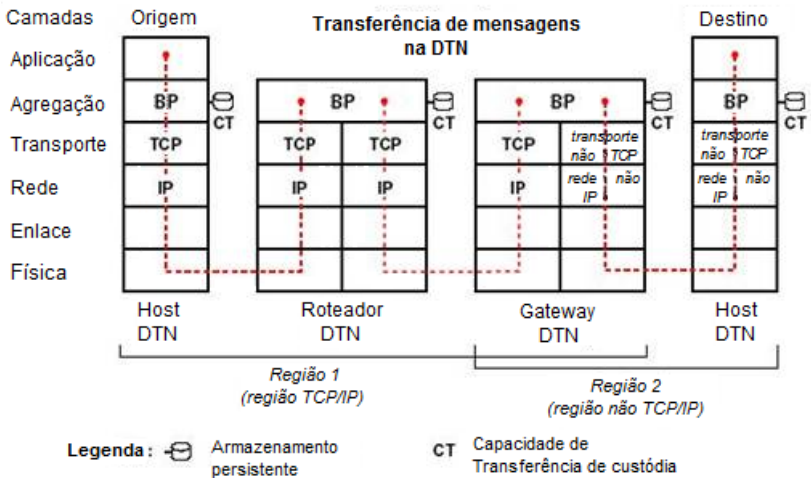


Figura 25: Comunicação na Arquitetura DTN proposta pelo DTNRC (WARTHMAN et al., 2013).

Essa organização também foi estabelecida para que os *hosts* na rede DTN possam retransmitir agregados perdidos ou corrompidos,

uma vez que a camada de transporte não é mais fim-a-fim. Dessa forma, os agregados devem ser armazenados no *host* até se ter a confirmação do *host* seguinte que deverá se responsabilizar pela transmissão, o que é conhecido como custódia da transmissão. Se o *host* para quem se está tentando repassar a custódia não responder dentro do tempo de reconhecimento de transmissão, o *host* que está repassando a custódia tenta retransmitir o agregado. Caso não consiga, o *host* pode tentar passar a custódia para outro *host*, ou ainda, continuar mantendo a custódia do agregado.

Na DTN os *hosts* podem ser de três tipos: 1) o *host* pode originar ou receber agregados, sendo que não pode encaminhá-los através da rede; 2) o roteador tem a função de rotear dentro de uma determinada região os agregados que chegam até ele; e 3) o *gateway* tem a função de encaminhamento de agregados na inter-região (entre regiões diferentes). A transferência da custódia pode ser realizada pelos três, sendo que nos *gateways* é obrigatória.

De forma geral e não somente na arquitetura apresentada pela DTNRG, a segurança ainda é um tema que necessita de mecanismos mais eficientes. Pode-se perceber que o desafio não é pequeno, pois uma grande parte dos métodos de segurança eficazes são conversacionais, o que não é viável realizar em DTN.

Além da segurança outras áreas também abrem margem para muitas pesquisas, tais como a área de endereçamento e de roteamento.

Essa arquitetura apresenta mais conceitos e considerações sobre outros aspectos de funcionamento de uma DTN não comentados aqui por não serem tão relevantes para esse trabalho. Contudo, os aspectos referentes ao roteamento são importantes nesta pesquisa e são apresentados a seguir.

6.2.1 Arquitetura de Roteamento

O DTNRG permite que sejam definidas diferentes abordagens de roteamento de acordo com os tipos de contatos e outras características da rede.

No desenvolvimento de uma abordagem de roteamento que esteja de acordo com a implementação da DTN2 ², é necessário levar em consideração os módulos, blocos funcionais e interação entre os procedimentos e eventos (DEMMEER et al., 2004).

A Figura 26 apresenta os principais módulos do sistema de rote-

²<http://www.dtnrg.org/docs/code/DTN2/>

amento dentro da arquitetura DTN do DTNRG. As setas indicam uma relação entre os módulos e o sentido indica o módulo que está sendo chamado. As cores indicam os blocos funcionais a que pertencem.

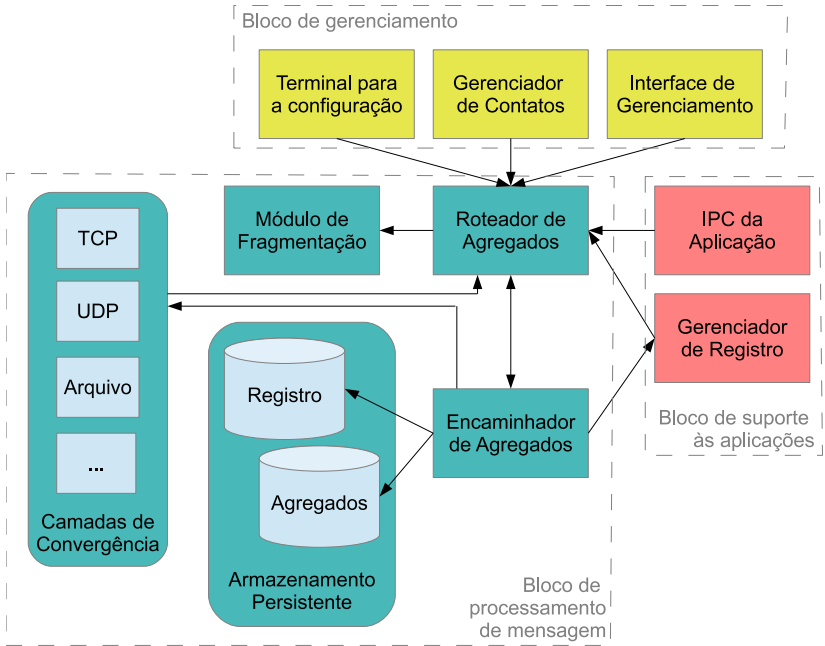


Figura 26: Módulos da arquitetura DTN do DTNRG. Adaptado de (DEMMER et al., 2004).

Estes módulos são organizados em três blocos:

Bloco de Processamento de Mensagem

- *Roteador de Agregados*: ele implementa toda a seleção de rota e tomada de decisão da política de agendamento. Ele recebe informações que podem afetar as decisões de roteamento e envia instruções para o módulo Encaminhador de Agregados. Além disso, ele implementa as políticas de roteamento. Esse módulo deve permitir a execução de diferentes algoritmos de roteamento.
- *Encaminhador de Agregados*: ele executa as instruções passadas pelo módulo Roteador de Agregados. Ele interage com as Camadas de Convergência, o Armazenamento Persistente e Registros para executar essas instruções.

- *Camadas de Convergência*: elas adaptam a unidade de dado do agregado para várias unidades de transporte e vice-versa. Elas também processam informações de sinalização requeridas pelo Roteador de Agregados em alguns casos.
- *Armazenamento Persistente*: ele mantém o conteúdo dos agregados durante a operação do SCF, além de manter os registros. A persistência pode ser implementada em um banco de dados, um arquivo ordinário ou outra forma de armazenamento.
- *Fragmentação*: ele é o módulo responsável pela fragmentação e remontagem dos fragmentos dos agregados. Quando o contato finaliza antes de transmitir todo o agregado, a fragmentação é feita reativamente. Quando todos os fragmentos de um agregado são recebidos, este módulo pode sinaliza-lo ao Roteador de Agregados.

Bloco de Suporte às Aplicações

- *Interprocessos de Comunicação (IPC) da Aplicação*: através do canal de comunicação de interprocessos, uma pequena biblioteca que comunica com o roteador é usada em uma aplicação DTN. Ela trata das associações da demultiplexação de mensagens e envia e recebe mensagens das aplicações.
- *Gerenciador de Registro*: quando uma aplicação solicita receber um conjunto de agregados, esta solicitação é encapsulada no Registro e armazenada de forma persistente para manter essa informação após falhas da aplicação ou do sistema. Mesmo que a aplicação não esteja disponível para receber o agregado, o último roteador concordará em armazenar o agregado para o qual uma aplicação foi registrada.

Bloco de Gerenciamento

- *Gerenciador de Contato*: ele identifica e mantém informações sobre o estado dos links, sua disponibilidade, informação histórica sobre conectividade, desempenho e agendamentos futuros de conexões. Ele converte informações de contato em descrições abstratas para serem usadas pelo módulo Roteador de Agregados.
- *Interface de Gerenciamento*: as restrições de políticas especiais ou preferências que podem mudar as decisões de roteamento são sinalizadas ao módulo Roteador de Agregados.

- *Terminal para a Configuração*: ele provê uma interface de linha de comando e um laço de evento para testes e depurações. Também, provê um método estruturado para informar as opções de configurações iniciais.

A abordagem de roteamento descrita nesta tese foi definida para operar baseado nessa arquitetura DTN, interoperando com as funções definidas para ele.

6.3 PROPOSTA DE ARQUITETURA DE SUPORTE AO ROTEAMENTO

Neste trabalho foi estudada a inclusão da abordagem de roteamento proposta na arquitetura apresentada pelo DTNRG e implementada na DTN2. Para essa arquitetura operar com contatos de tipos distintos, o módulo de roteamento deve ser projetado respeitando-se as interfaces já definidas na arquitetura, permitindo que módulos de roteamento diferentes tratem determinados tipos de contatos, conforme mostrado anteriormente na Figura 26.

A abordagem de roteamento baseada em RNAs proposta neste trabalho apresenta aspectos convergentes com a arquitetura proposta pelo DTNRG, apresentando partes em comum. Foram identificadas as seguintes partes:

- a base de conhecimento: com procedimentos para descrever e armazenar os informações, regras e filtros sobre eles;
- o armazenamento persistente das estruturas da RNA: normalmente é necessário recuperar as estruturas das RNAs e os pesos treinados para o funcionamento da RNA em caso de reinicializações, transferência de informação para outros nós ou liberação de memória principal para outras atividades.
- a avaliação de necessidade de aprendizado: se necessário, esta avaliação gera um evento para invocar a criação de novas RNAs e um novo aprendizado. Sua execução não deve interferir na estrutura que está rodando. Os procedimentos de avaliação podem ser simples, baseados em algum período de tempo ou mais complexos, baseados na análise da eficiência das predições.
- o recebimento de informações e atualizações em dois níveis: parametrizado com efeito imediato (ajuste de parâmetros do algoritmo

e para validação de contatos preditos); e aprendizado, somente disponível após o término do aprendizado (novas estruturas de RNA e/ou novos pesos).

A inclusão da proposta de roteamento na arquitetura do DTNRG implica na inclusão de um novo módulo de Roteamento de Agregados para a implementação da estratégia proposta. Algumas pequenas modificações são também previstas no módulo Armazenamento Persistente, para armazenar os exemplos de treinamento, as RNAs e os seus pesos. Além disso, é necessário prever o controle de cópias das mensagens no módulo de Encaminhador de Agregados.

Tomando por base a arquitetura DTN do DTNRG (Figura 26), a Figura 27 mostra a arquitetura proposta para o Módulo de Roteamento de Agregados e suas interações com os demais módulos da arquitetura do DTNRG. Esta figura mostra apenas as novas interações e as modificadas, sendo que as demais permanecem iguais.

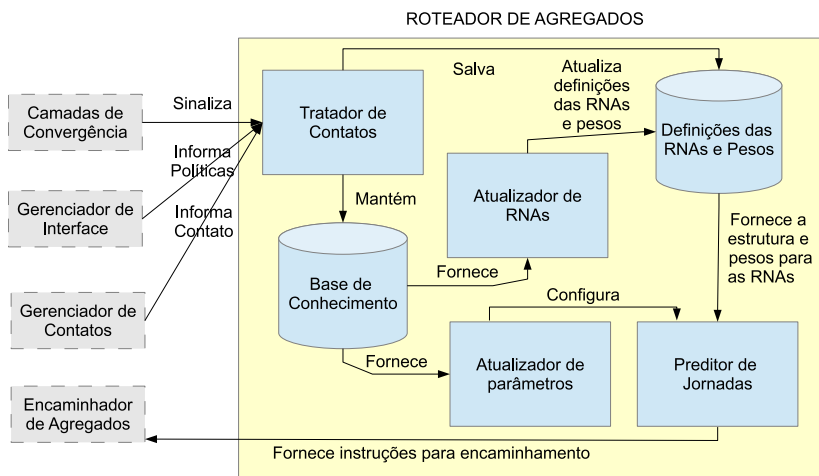


Figura 27: O módulo Roteador de Agregados para a estratégia RNA proposta.

O Preditor de Jornadas é executado no módulo Roteador de Agregados, sendo já apresentado em detalhes na Figura 3, no Capítulo 3. Ele é invocado a cada novo contato, fornecendo instruções para o módulo Encaminhador de Agregados.

A fase de operação de tratamento dos contatos é realizada pelo Tratador de Contatos, sendo mostrado na 5. A fase de atualização do

conhecimento é realizada pelo Atualizador de RNAs, conforme foi mostrado na Figura 6. Ambas as figuras foram apresentadas no Capítulo 3).

As partes funcionais propostas para o módulo roteador de agregado são as seguintes:

- **Tratador de Contatos:** ela recebe informações do protocolo de roteamento, tais como sinalização das camadas de convergência e dos gerenciadores de contato e da interface de gerenciamento, coleta informações dos contatos e mantém as informações processadas na Base de Conhecimento. Ela também é responsável por salvar dados pré-processados vindos de outros nós e que podem ser utilizados diretamente pela estratégia de roteamento. No caso dessa proposta que usa RNAs, é previsto que ao entrar em contato com outro nó sejam trocadas as estruturas e pesos atualizados das suas redes neurais;
- **Atualizador de RNAs:** ela recupera informações da base de conhecimento para realizar o pré-processamento considerando filtragens e transformações, define os dados para treinamento, redefine as RNAs utilizadas e realiza o treinamento.
- **Atualizador de Parâmetros:** ela redefine parâmetros dos algoritmos obtendo efeito imediato. Ela é empregada para permitir mudanças dinâmicas no comportamento da estratégia de roteamento, diferentemente da parte Atualizador de RNAs, em que é necessário tempo, processamento e memória para se encontrar novas estruturas e calcular valores para essas estruturas;
- **Preditor de Jornadas:** essa parte funcional deve implementar a estratégia de roteamento proposta, permitindo construir jornadas previstas a partir do uso do Preditor de Contatos baseado em RNAs (apresentado no Capítulo 4), de acordo com o critério de menor atraso na entrega.

A parte funcional Atualizador de Parâmetros é executada a cada contato obtendo informações vindas de outros nós, além do próprio contato, calculando e determinando rapidamente novos parâmetros para o algoritmo de predição de jornadas.

A parte funcional Atualizador de RNAs é acionada através de um evento que indica a necessidade de atualização nas RNAs utilizadas. Esse evento é acionado em um processo que determina se as RNAs devem ter um novo treinamento ou devem ser redefinidas. Esse

processo pode ser simples, baseado em algum horário do dia, ou mais elaborado, verificando erros de predição, taxas de acerto e outras informações. Essa parte funcional é suspensa quando um novo contato ocorre e retomada ao final do tratamento do contato.

Tanto o armazenamento das Definições e Pesos para as RNAs e da Base de Conhecimento são definidas no módulo de Armazenamento Persistente.

As RNAs locais treinadas e as RNAs vindas de outros nós são apenas salvas no módulo de Armazenamento Persistente. As RNAs locais são definidas e treinadas pela parte funcional Atualizador de Aprendizado e depois de salvas, são recuperadas do módulo de Armazenamento Persistente para serem usados na parte funcional do Preditor de Jornadas.

6.4 MODELO PARA IMPLEMENTAÇÃO DA ARQUITETURA DE ROTEAMENTO PROPOSTA

6.4.1 Modelo de classes

Para organizar a implementação da abordagem de roteamento proposta foi concebido um modelo abstrato de classes de acordo com a implementação da DTN2. Nesse modelo foi definida a classe Roteador RNA Proposto que é uma especialização do Roteador de Agregado definido nessa implementação. No diagrama de classes da Figura 28 é possível ver a estrutura de classes conforme determinado e implementado pelo DTNRG. As classes Roteador RNA Proposto, Base de Conhecimento e RNA são novas classes propostas neste trabalho. O *framework* é um pacote utilizado na implementação das RNAs. As demais classes são previstas na implementação da DTN2 do DTNRG.

Na DTN2, um roteador DTN (*BundleDaemon*) é iniciado instanciando uma classe do Gerenciador de Contatos e do Roteador de Agregados que então ficam a espera da ocorrências de eventos da classe Tratador de Eventos do Agregado. O início e o término de um contato são exemplos de eventos.

Os principais métodos para tratamento dos agregados e dos eventos são definidos nessas classes, sendo possível sobrescrevê-los na definição do próprio roteador para tratamento de alguma especificidade. As filas de mensagens entregues, a fila de mensagens pendentes (para entrega), custódias, ações de encaminhamento com ou sem cópia e outras já estão previstas nas classes generalizadas.

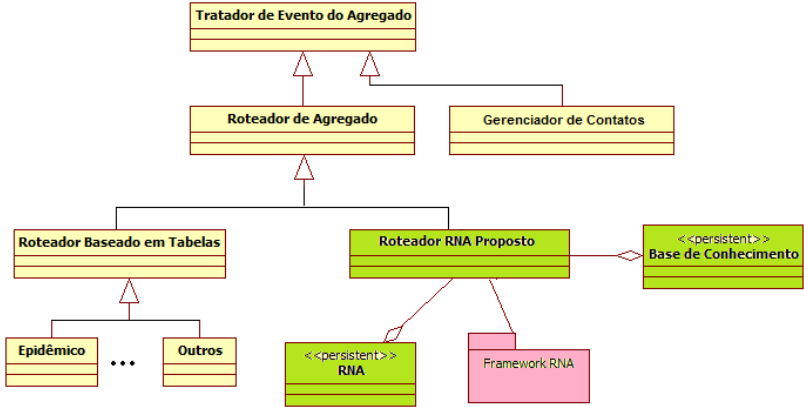


Figura 28: Diagrama de classes da arquitetura de roteamento proposta.

O roteador RNA proposto deve ter uma forma de armazenar e recuperar os dados de contato, os parâmetros, as definições das RNAs e os seus pesos. É prevista a busca desses dados mediante a aplicação de um filtro para realizar, por exemplo, a recuperação dos dados de contato de um intervalo de dias para serem utilizados no treinamento. Para tanto, esse dados devem ser armazenados em algum sistema de banco de dados.

Um método que determina a necessidade de aprendizado deverá ser definido para disparar um evento no Tratador de Eventos do Agregado. A interface desse evento é herdada e sobrescrita por uma especialização do Roteador de Agregado, tal como o Roteador RNA proposto.

O Roteador RNA proposto deve utilizar um pacote de métodos para construção das RNAs, algoritmos de treinamento e outros métodos necessários.

Uma instância da classe Gerenciador de Contatos deverá tratar da troca de informações e mensagens entre nós em contato a partir da definição de um protocolo de comunicação, entre outras tarefas.

6.4.2 Protocolo de comunicação

Um protocolo de comunicação deve permitir a troca de informações para atualizar as bases de conhecimento. No caso da proposta de roteamento deste trabalho é especificado o seguinte formato de campos para a troca de informações:

- identificador do nó: identifica que as informações transmitidas estão neste nó;
- número da parte do conteúdo do protocolo informado: identifica a parte que está sendo enviada, pois pode ser necessário dividir o conteúdo para não utilizar toda a duração do contato para envio de informações do protocolo. O conteúdo é enviado com o número da parte na sequência inteira, pois permite identificar quantas partes faltam. Quando for zero não há mais partes para receber;
- tipo de mensagem: é um código que identifica se o conteúdo trata da: indicação e solicitação das versões das informações disponíveis, parâmetros, pesos das RNAs, arquiteturas, exemplos de treinamento ou contatos;
- conteúdo: conteúdo de acordo com o tipo de mensagem transmitida.

Se o campo tipo de mensagem indicar uma solicitação da versão das informações disponíveis, no campo conteúdo deve ser informadas as versões das informações presentes no nó solicitante. Quando o nó de destino recebe essa solicitação ele responde com uma mensagem do tipo indicação das versões das informações disponíveis, levando no conteúdo da mensagem essas informações.

Essas definições básicas devem permitir a implementação da estratégia proposta de acordo com a arquitetura de roteamento proposta pelo DTNRG.

6.4.3 Discussão sobre a implementação

A implementação da estratégia de roteamento proposta deverá concentrar as principais alterações nas funcionalidades do módulo de Roteador de Agregados. Na implementação da DNT2 isso é feito através da herança da classe Roteador de Agregados (*BundleRouter*), alterando e incluindo novos métodos se necessário. O tratamento de eventos definidos na classe Tratador de Evento do Agregado (*BundleEventHandler*) também deve ser realizado. Existem vários métodos nessas classes, sendo que os mais importantes para este trabalho são:

- *BundleRouter*(): método construtor;
- *accept_bundle*(·): além de indicar se o Agregado deve ser aceito pelo sistema, permite indicar se o Agregado transporta informação

de roteamento e deve ser tratado pelo roteador através do novo método *getInfoFromContactedNode()*.

- *initialize()*: método para inicialização do roteador. Ele verifica se as RNAs estão criadas e estão funcionando, sendo que quando não estiverem funcionando deve ser invocado o método para recriar e retreinar a RNA com problema.
- *recompute_routes()*: método para forçar a reconstrução de rotas. Ele implementa ou invoca o preditor de jornadas.
- *should_fwd()*: método para verificar se o agregado deve ser encaminhado para o nó em contato. Nesse método deve ser implementado o algoritmo que realiza o controle de cópia das mensagens.
- *handle_event()*: método herdado da classe Tratador de Evento do Agregado (*BundleEventHandler*). Ele permite tratar vários eventos importantes:
 - estabelecimento de um contato: obtém informações do contato estabelecido para o preditor de jornadas e obtém informações de roteamento vindas do nó em contato invocando o método *getInfoFromContactedNode()*. Após isto chama o método *recompute_routes()* para redefinir as rotas.
 - perda/finalização de um contato: suspende a atividade que está sendo realizada. Se o próximo contato for com o mesmo nó em um intervalo de tempo pequeno (p.e. 300 segundos) a atividade suspensa continua, caso contrário o roteador é preparado para um novo contato.
 - necessidade de atualização das RNAs: é um novo evento que notifica o roteador da necessidade de recriar as definições das RNAs e/ou de realizar novo treinamento.
- *getInfoFromContactedNode()*: método para realizar a troca de informações com o nó em contato sobre RNAs, pesos e outras informações para redefinição de parâmetros dos algoritmos do preditor de jornadas. Essa troca de informações é regrada conforme a definição do protocolo de comunicação para roteamento especificada na seção anterior.
- *initializeRNA()*: método para realizar a criação e/ou treinamento das RNAs. Um parâmetro indica se a RNA deve ser recriada.

Além disto, é necessário criar um tratador de eventos baseado na classe *BundleEventHandler* para incluir o evento que indica a necessidade da recriação da RNA e de realizar um novo treinamento. A funcionalidade de verificação da necessidade de aprendizagem da RNA, a qual deve acionar esse evento, deve estar implementada como um novo método no próprio roteador, mas deve ser disparada por um evento de relógio ou pelo evento de fim do contato. A execução dessa verificação deve ser suspensa quando um evento de recebimento de contato é recebido. Também deve ter a execução suspensa nesse evento o método que atualiza as RNAs. Após o tratamento dos contatos a execução da verificação ou da atualização pode continuar.

O bloco de persistência de dados deve ser incrementado com as estruturas de dados para receber a base de conhecimento do roteador RNA proposto com as informações de:

- histórico de contatos;
- exemplos para treinamento;
- definições de RNAs para cada nó;
- conjuntos de pesos para as RNAs;

Em uma versão mais simples do roteador RNA proposto não é necessária a realização de buscas avançadas por informações, apenas consultas de acordo com um nó informado. Em outras versões poderá ser necessário executar filtros avançados para selecionar dados de contato, prevenendo-se com isso o uso de algum sistema de armazenamento e recuperação de dados com recursos para isso.

Deve-se ainda realizar a adequação das estruturas de dados para os nós, contato, rotas e agregados (que são as mensagens na implementação atual), bem como as mensagens de controle do roteador e eventos, tais como as mensagens de ação para o encaminhamento, o estado do agregado, erros e códigos de eventos.

Grande parte da implementação do método *recompute_routes()*, que contempla o preditor de jornadas incluindo a predição de jornadas com as RNAs já foi desenvolvida em linguagem Java, bastando convertê-la para linguagem C++, usado na implementação da DTN2 e adequá-la a estrutura de classes já existente nessa implementação com respectivos eventos e sinalizações. Da mesma forma está implementada a parte referente a inicialização, criação dinâmica das RNAs e treinamentos (método *initializeRNA()*). O controle da replicação deve ser implementado no método *should_fwd()*, sendo que um dos parâmetros

deve indicar a ação de realizar somente o encaminhamento ou copiar a mensagem.

Em *handle_event()*, um processo (mesmo que simples) deve ser implementado para ativar o evento de atualização das RNAs. Além disso, outro tratador de evento deve permitir a suspensão e continuação do processo de atualização das RNAs. O protocolo de comunicação definido na seção anterior também deve ser implementado, a fim de ser utilizado pelo método *getInfoFromContactedNode()*.

Será necessário ainda adequar o uso do sistema de registro de *logs* para se ter registro das operações realizadas para avaliar o funcionamento do roteador.

Por fim, é pressuposto que o tempo (data e hora) obtida nos nós esteja sincronizada com os demais nós, podendo apresentar um erro (*drift*) de até 1 minuto (ou um pouco mais).

6.5 CONCLUSÃO

Existem algumas arquiteturas propostas para a comunicação em redes tolerantes a interrupções e falhas (DTN). A mais completa e implementada é a DTN2 proposta pelo DTNRG do IRTF, que prevê extensões para tratamento de questões em aberto. Ela é mais indicada para a adição do suporte a arquitetura de roteamento proposta.

Foi definida neste trabalho uma arquitetura de roteamento organizada em partes e que se adequa a implementação da DTN2.

O diagrama de classes apresentado é adequado para modelar a implementação da estratégia de roteamento proposta para a arquitetura DTN do DTNRG. São descritos o funcionamento das partes principais, sua ligação com a proposta de roteamento desse trabalho e sua ligação com a arquitetura de roteamento DTN do DTNRG.

Uma definição para a troca de mensagens de roteamento através de um protocolo de comunicação foi também apresentada.

A implementação da DTN2 é modular e modelada com orientação a objetos, facilitando a reutilização de todas os blocos funcionais que interagem com o bloco funcional de Roteamento. A implementação de código principal envolve então a criação de um roteador baseado na classe Roteador de Agregados e na inclusão de um evento para indicar a necessidade de atualização das RNAs.

7 CONCLUSÃO

Nesta trabalho foram apresentados estudos e o desenvolvimento de uma nova estratégia de roteamento para Redes Tolerantes a Atrasos e Interrupções (DTN) baseada em Redes Neurais Artificiais (RNA). A estratégia proposta apresentou superioridade em vários aspectos em relação às estratégias avaliadas. A abordagem criada leva em consideração a aplicação sucessiva de um preditor de contatos e um processo de construção de jornadas preditas.

O tratamento do roteamento com contatos oportunistas é difícil, especialmente com poucas informações para ajudar no descobrimento de padrões e relações entre os dados históricos para diminuir a aleatoriedade dos contatos. Esse contexto de contatos foi chamado de quase-oportunista. O procedimento desenvolvido para projetar as RNAs nesse contexto de contatos e com ênfase no retorno à etapas anteriores foi fundamental. Dentre as etapas, a etapa de análise dos dados foi essencial para se obter informações importantes que permitiram alcançar um desempenho satisfatório no roteamento.

Foram usados dados de contato reais do projeto DieselNet, simulando-se apenas a troca das mensagens sobre as conexões reais. Assim, não houve interferência de muitas simplificações de simulação, trazendo resultados mais realistas sobre a aplicação das estratégias avaliadas.

A utilização de RNAs para a predição de contatos apresentou melhor taxa de acerto do que as taxas de acerto obtidas com a probabilidade baseada na frequência de contato. Dessa forma, pode-se entender que as RNAs conseguiram se aproveitar das informações temporais de contatos para modelar o estabelecimento de contatos entre nós da rede para predição do próximo nó e instante de contato. Contudo, para os dados reais não foi possível verificar relações úteis em dias diferentes na sequência de nós em contatos consecutivos maior que dois nós (nó de referência e próximo nó de contato) para melhorar o roteamento.

A abordagem criada superou a estratégia do caminho de maior probabilidade, desenvolvida no algoritmo de roteamento do MaxProp, nas métricas de taxa de entrega de mensagens, atraso na entrega de mensagens e custo de entrega baseado no número de mensagens na rede para cada mensagem entregue. Esses resultados foram obtidos em todos os cenários experimentados, tanto para o cenários com e sem cópia de mensagens usando dados reais ou sintéticos. A estratégia de Entrega Direta também foi superada em todas as essas métricas nos

cenários experimentados. A estratégia Epidêmica Ilimitada foi superada no custo de entrega baseado no número de mensagens na rede para cada mensagem entregue. Essa estratégia com limite do tamanho da fila de mensagens também foi superada até um determinado limite no tamanho da fila de mensagem. Essa superação da abordagem proposta também foi evidenciada quando a duração real dos contatos foi utilizada para limitar o tempo de contato.

A diferença da taxa de entrega em comparação a obtida com o MaxProp não foi considerada grande. Contudo, deve ser considerada a dificuldade de operar em um ambiente de contatos reais sujeitos a diversas imprevisibilidades.

Pode-se avaliar que o menor custo foi obtido ao encaminhar a mensagem original mais uma cópia desta. Também foi avaliado que a estratégia proposta utilizando o limite de 4 cópias da mensagem obteve custo de mensagem na rede por mensagem entregue ainda um pouco inferior ao sem cópia das mensagens.

O desempenho referente ao tempo de execução da simulação foi parecido na estratégia proposta e no MaxProp para o cenário com dados sintéticos. Quando aplicado aos dados reais, com maior número de nós e menor controle no estabelecimento dos contatos, o tempo de execução foi pior na estratégia proposta. O tempo de treinamento das RNAs na abordagem proposta e o tempo para popular a matriz de probabilidades no MaxProp não foram considerados nessa avaliação por tratar-se de processo *off-line*.

A proposta da estratégia RNA desenvolvida foi modelada para a arquitetura modular de roteamento DTN proposta pelo DTNRG de acordo com seus módulos e blocos funcionais. A implementação dessa estratégia de roteamento foi estudada e teve seus primeiros modelos apresentados em direção a implementação da DTN2 também proposta pelo DTNRG da IRTF.

Algumas limitações do trabalho são devido ao conjunto de dados que foi possível utilizar. Um conjunto de dados com outras informações poderia trazer melhores resultados. Além disso, novos estudos com a análise de dados podem ser realizados para se obter outras correlações entre as informações disponíveis.

O gerenciamento das filas de mensagens não foi uma preocupação desse trabalho, sendo mantido da mesma forma em todas as estratégias para não interferir nos resultados, contudo é importante se preocupar com isso, pois o desempenho do roteamento pode sofrer mudanças significativas.

Diante dos resultados obtidos são abertos espaços para avanços,

incluindo e integração com outros mecanismos que viabilizem a utilização da estratégia de roteamento proposta em aplicações reais.

Considerações finais, limitações e trabalhos futuros foram ainda organizados por assunto tratado na tese, conforme apresentado nas seções seguintes.

7.1 PREDITOR DE JORNADAS

O preditor de jornadas implementa um construtor de jornadas que aplica sucessivamente a predição de contatos para obter um conjunto de jornadas modeladas em um grafo. Assim, permite-se que as melhores jornadas previstas sejam encontradas a fim de se obter informações para a tomada de decisão de encaminhamento.

O ajuste do parâmetro *timeSlice* permitiu aumentar o número de mensagens entregues, considerando repetições do experimento com treinamentos diferentes, até valores próximos ao máximo obtido no experimento.

Existe espaço para a melhoria do algoritmo do construtor de jornadas. Uma melhoria a ser pesquisada é a partir de outras formas de determinação dinâmica do *timeSlice*, como por exemplo, adotando uma função exponencial que caracterize a frequência dos contatos de acordo com o tempo. Outras melhorias a serem avaliadas são a implementação da busca em largura por contatos previstos e de outros mecanismos para determinação de contatos válidos.

Outra modificação a ser investigada é a aplicação de um algoritmo inspirado na busca em largura, além da busca em profundidade implementada, para diminuir o tempo necessário na construção das jornadas previstas. Assim, os nós poderiam ser localizados até determinada largura e profundidade que fossem adequadas para manter o desempenho desejado.

7.2 PREDITOR DE CONTATOS

7.2.1 Estabelecimento de contatos nos cenários estudados

Diferente do que era o pressuposto inicial do trabalho, não foi observado um padrão no encadeamento do nó anterior com nó atual e com próximo nó nas sequências de contato observadas nos dados de contatos reais do STPU.

A tabela de horários dos ônibus acaba não sendo cumprida de forma estrita e por isso muitos contatos acabam ocorrendo de forma muito oportunista. Assim como no '*efeito borboleta*', uma pequena variação no tempo dos ônibus para se dirigirem para o próximo ponto ou no tempo em que o ônibus espera pela descida e subida de passageiros promove grandes mudanças nas oportunidades de contato.

Assim, apesar da considerável organização da mobilidade dos nós nos cenários estudados, essa organização contribui, mas não o suficiente para estabelecer uma organização clara no estabelecimento de contatos. Por isso, os contatos no contexto desse trabalho foram classificados como contatos quase-oportunistas.

Essa mesma análise sobre a tabela de horários levou a considerar que para os sistemas de transporte públicos urbanos que não apresentam o cumprimento muito próximo do definido nos itinerários, presentes em grande maioria, a execução de itinerários estritos muito raramente acontecerá. Por causa disso, dos problemas de comunicação e da sensibilidade de mudanças na movimentação no estabelecimento dos contatos, não foi encontrada uma forma de representar essa condição na RNA. Em um trabalho posterior deve-se verificar se existe um caso médio para o estabelecimento de contatos e se isso seria representado pelo itinerário.

O cenário simulado até a faixa de tempo estipulado apresentou o comportamento desejado para simular o início de um sistema de transporte de ônibus. Os resultados mostraram coerência entre os valores obtidos para as métricas com dados reais e com dados sintéticos.

Em um próximo trabalho seria interessante avaliar a aplicação da abordagem proposta em redes formadas por dispositivos pessoais de bolso (PSN), aproveitando-se da movimentação do dia útil (WDM). Neste cenário as pessoas e veículos descrevem itinerários parecidos diariamente.

7.2.2 Procedimento em etapas para a definição das RNAs

Um procedimento baseado em etapas foi desenvolvido para definir as redes neurais utilizadas na estratégia de roteamento apresentada. Esse procedimento pode auxiliar outros trabalho em que se deseje utilizar RNAs para realizar predição.

Um grande esforço foi necessário para análise, definição de estruturas, testes e ajustes das RNAs até se alcançar resultados satisfatórios. Neste sentido as ferramentas de análise de dados, muito

presentes em mineração de dados, são importantes. Principalmente, deve-se ter grande domínio sobre o dados e o seu significado dentro do contexto avaliado.

A consolidação do procedimento como um método ainda dependerá de mais testes que venham a aplicá-lo e aperfeiçoa-lo para atender a eventuais necessidades.

7.2.3 Filtragem de dados

A aplicação de um filtro baseado na distância de KullBack-Leibler foi realizada, mas não foi possível tirar proveito para a estratégia RNA proposta. De outra forma, testes preliminares apontaram a obtenção de pequenas melhoras do número de mensagens entregues com a aplicação desse filtro no MaxProp. Como o objetivo principal se concentrou em explorar as melhorias possíveis nas estratégia baseada em RNA, o melhoramento do MaxProp ficou para um segundo plano.

Todavia, ainda pode existir espaço para que essa estratégia de filtragem permita alcançar resultados melhores com as RNA, mesmo por que não foi testada com as RNAs locais.

7.2.4 Perspectivas de melhorias no projeto das RNAs

Existe uma grande faixa de aplicação para o emprego das RNAs. Uma vez que os dados podem ser analisados e interpretados de muitas maneiras, diferentes RNAs podem ser construídas para atender a um determinado objetivo e ainda compor vários mecanismos que poderão ser utilizados para operar conjuntamente.

Uma outra forma de projetar as RNAs para predição de contatos seria reformulando o objetivo da predição para informar a probabilidade de chegar ao destino a partir de um contato estabelecido com o nó atual. Com isso a predição de contato com o destino forneceria diretamente o custo para alcançar o destino e dispensaria o uso de um construtor de jornadas. Contudo, uma vez que não houvesse um grafo para se explorar outras jornadas, deveria ser avaliado como poderiam ser implementados outros critérios além de o menor atraso de entrega.

Outra forma de melhorar o desempenho das predições poderia ser a obtenção de RNAs baseadas em perfis de movimentação ou de características de contato. Essas RNAs poderiam ser aplicadas quando o nó observa que as características de um perfil estão acontecendo na

movimentação de um nó ou nos contatos que vem sendo realizado.

7.3 MÉTRICAS AVALIADAS NAS PESQUISAS

Ao realizar as pesquisas consultando outros trabalhos, verificou-se certa dificuldade para realizar a comparação entre abordagens diferentes em DTN.

Alguns trabalhos apresentam métodos de medição diferentes e realizam comparações difíceis de serem avaliadas com os resultados de outros trabalhos. Algumas pesquisas reconhecem a dificuldade de comparar cenários diferentes e seus conjuntos de dados com características de contatos diferentes. Neste sentido, torna-se importante existir um conjunto mínimo padronizado de avaliações a serem realizadas, pois uma vez caracterizada a forma que os contatos são estabelecidos, pode-se obter inferências importantes quando for possível se obter medidas comparáveis, mesmo com pequenas restrições.

Conforme é realizado nesse trabalho, sugere-se que sempre sejam apresentadas comparações com a estratégia epidêmica ilimitada como um limite superior de desempenho e com a estratégia de entrega direta como sendo uma referência de desempenho inferior.

A apresentação da escala da taxa de entrega deve mostrar adicionalmente o número de mensagens para facilitar o entendimento do volume de mensagens que se está tratando. Adicionalmente, a apresentação do custo de entrega relacionado o número de mensagens na rede para cada mensagem entregue fornece um entendimento mais claro e direto da eficiência da estratégia baseada em replicação das mensagens.

7.4 ARQUITETURA DE ROTEAMENTO

Os procedimentos idealizados para a arquitetura de roteamento estão de acordo com os requisitos dos procedimentos descritos nos blocos funcionais e diagramas de classes da DTN2.

A implementação da DTN2 favorece a modularização e reutilização de código por apresentar-se modelada com orientação a objetos e com blocos funcionais bem definidos. O modelo de roteador baseado em uma classe genérica foi estabelecido pelo DTNRG pela necessidade de contemplar o roteamento em contexto com contatos que apresentam características diferentes.

Deve-se avaliar a possibilidade de generalização da arquitetura e modelo apresentados para a definição e implementação de roteadores baseados em outros métodos de aprendizagem de máquina (MAM). Muitos MAMs apresentam requisitos parecidos com os presentes nas RNAs, como por exemplo a necessidade de uma base de conhecimento, um pré-processamento dos dados para serem aplicados na aprendizagem e um processo de aprendizagem ou otimização ou outro que tenha a função de modelar relações entre os dados.

7.5 PUBLICAÇÕES PROVENIENTES DA PESQUISA

No decorrer da pesquisa desse trabalho foram apresentados dois artigos em eventos e realizada uma submissão de artigo para revista.

No *Global Information Infrastructure Symposium* (GIIS), em outubro de 2013, foi apresentado o artigo intitulado *Quasi-opportunistic contact prediction in delay/disruption tolerant network*. Encontra-se disponível na biblioteca digital IEEE Xplore¹ (SEGUNDO; FARINES; SILVA, 2013).

Em outubro de 2014 o artigo intitulado *Predicting journeys for DTN routing in a public transportation system* foi apresentado na *IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications* (WiMob). Também está disponível na biblioteca digital IEEE Xplore (SEGUNDO; SILVA; FARINES, 2014).

Um artigo foi submetido para publicação no *Journal of Network and Computer Applications* da revista Elsevier em março de 2015. O artigo encaminhado, intitulado *A DTN Routing Strategy Based on Neural Networks*, encontra-se em fase de avaliação pelo editor.

¹Acessada pela Internet pelo endereço <http://ieeexplore.ieee.org/>.

REFERÊNCIAS

- AHMED, S.; KANHERE, S. A bayesian routing framework for delay tolerant networks. In: *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*. [S.l.: s.n.], 2010. p. 1–6. ISSN 1525-3511.
- BAI, F.; HELMY, A. A Survey Of Mobility Models in Wireless Adhoc Networks. *Kluwer Academic*, Chapter 1, p. 1–30, 2006.
- BALASUBRAMANIAN, A.; LEVINE, B. N.; VENKATARAMANI, A. Replication routing in dtns: A resource allocation approach. *IEEE/ACM Transactions on Networking*, IEEE Press, Piscataway, NJ, USA, v. 18, n. 2, p. 596–609, abr. 2010. ISSN 1063-6692. Disponível em: <<http://dx.doi.org/10.1109/TNET.2009.2036365>>.
- BALASUBRAMANIAN, A.; MAHAJAN, R.; VENKATARAMANI, A. Augmenting mobile 3G using WiFi. In: *Proceedings 8th ICMSAS*. New York, USA: ACM, 2010. p. 209–222. ISBN 9781605589855. Disponível em: <<http://portal.acm.org/citation.cfm?doid=1814433.1814456>
<http://dl.acm.org/citation.cfm?id=1814456>>.
- BURGESS, J. et al. Maxprop: Routing for vehicle-based disruption-tolerant networks. In: *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*. [S.l.: s.n.], 2006. p. 1–11. ISSN 0743-166X.
- BURGESS, J. et al. *CRAWDAD data set umass/diesel (v. 2008-10-21)*. out. 2008. Downloaded from <http://crawdad.org/umass/diesel/>.
- CAO, Y.; CRUICKSHANK, H.; SUN, Z. Asymmetric spray based routing for delay tolerant networks. In: *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*. [S.l.: s.n.], 2011. p. 1–5. ISSN 1550-2252.
- CASTEIGTS, A. et al. Time-Varying Graphs and Dynamic Networks. *Internation Journal of Parallel, Emergent and Distributed Systems*, nov. 2010.
- CERF, V. et al. *RFC 4838, Delay-Tolerant Networking Architecture*. 2007. Disponível em: <<ftp://ftp.internic.net/rfc/rfc4838.txt>,
<http://www.ietf.org/rfc/rfc4838.txt>>.

- COVER, T. M.; THOMAS, J. A. *Elements of information theory*. New York, NY, USA: Wiley-Interscience, 1991.
- DEMMER, M. et al. *Implementing Delay Tolerant Networking*. [S.l.], dec 2004. Disponível em: <<http://dtnrg.org/docs/docs/papers/demmer-irb-tr-04-020.pdf>>.
- DIANA, R. et al. A DTN routing scheme for quasi-deterministic networks with application to LEO. . . . *IEEE 76th VTC*, p. 3–6, 2012.
- DIESELNET, U. D. *UMass Dieselnet Dome Testbed*. Acesso em 25/04/2014: UMass Dieselnet Dome, 2014. Disponível em: <<https://dome.cs.umass.edu/umassdieselnet>>.
- DOERING, M. *Disruption Tolerant Vehicular Communication in Public Transportation Systems*. 141 p. Tese (Doutorado) — Technische Universität Carolo-Wilhelmina zu Braunschweig zur Erlangung des Grades, 2012.
- DOERING, M. et al. IBR-DTN : An Efficient Implementation for Embedded Systems. p. 117–119, 2008.
- DOERING, M.; PÖGEL, T.; WOLF, L. Dtn routing in urban public transport systems. In: *Proceedings of the 5th ACM Workshop on Challenged Networks*. New York, NY, USA: ACM, 2010. (CHANTS '10), p. 55–62. ISBN 978-1-4503-0139-8. Disponível em: <<http://doi.acm.org/10.1145/1859934.1859947>>.
- ENDERLE, J.; PETZ, A.; JULIEN, C. A Framework for Evaluating DTN Mobility Models. *TR-UTEDGE*, v. 001, 2009.
- FALL, K. A delay-tolerant network architecture for challenged internets. In: *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. New York, NY, USA: ACM, 2003. (SIG-COMM '03), p. 27–34. ISBN 1-58113-735-4. Disponível em: <<http://doi.acm.org/10.1145/863955.863960>>.
- FAUSETT, L. *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Prentice-Hall, 1994. (Prentice-Hall international editions). ISBN 9780133341867. Disponível em: <<https://books.google.com.br/books?id=ONylQgAACAAJ>>.
- FAYYAD, U. M. et al. (Ed.). *Advances in Knowledge Discovery and Data Mining*. Menlo Park, CA, USA: American Association for Artificial Intelligence, 1996. ISBN 0-262-56097-6.

FREITAS, P. J. F. *Introdução à Modelagem e Simulação de Sistemas*. 2. ed. [S.l.: s.n.], 2008.

HALL, M. et al. The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.*, ACM, New York, NY, USA, v. 11, n. 1, p. 10–18, nov. 2009. ISSN 1931-0145. Disponível em: <<http://doi.acm.org/10.1145/1656274.1656278>>.

HARTOG, L. D.; SPYROPOULOS, T.; LEGENDRE, F. Using Public Transportation as a DTN Backbone: Mobility Properties and Performance Analysis. *IEEE AOC'10*, Montreal, Canada, n. Section III, jun. 2010.

HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. NY: Macmillan, 1994.

HEATON, R. I.; CONTRIBUTORS. *Encog Machine Learning Framework*. Acesso em 09/12/2013: <http://www.heatonresearch.com/encog>, 2013. Disponível em: <<http://www.heatonresearch.com/encog>>.

HERTZ, J.; PALMER, R. G.; KROGH, A. S. *Introduction to the Theory of Neural Computation*. 1st. ed. [S.l.]: Perseus Publishing, 1991. ISBN 0201515601.

JAIN, S.; FALL, K.; PATRA, R. Routing in a delay tolerant network. *2004 Proceedings IEEE SIGCOMM*, ACM Press, New York, New York, USA, p. 145, 2004.

JOHNSON, D. B.; MALTZ, D. A. Dynamic source routing in ad hoc wireless networks. In: _____. *Mobile Computing*. SpringerLink, 1996. (The Kluwer International Series in Engineering and Computer Science, v. 353), p. 153–181. ISBN 978-0-7923-9697-0. Disponível em: <<http://www.cs.rice.edu/dbj/pubs/kluwer-dsr.pdf>>.

JUANG, P. et al. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. *SIGARCH Comput. Archit. News*, ACM, New York, NY, USA, v. 30, n. 5, p. 96–107, out. 2002. ISSN 0163-5964. Disponível em: <<http://doi.acm.org/10.1145/635506.605408>>.

KARAGIANNIS, T.; BOUDEC, J.-Y. L.; VOJNOVICI, M. Power law and exponential decay of intercontact times between mobile devices. *Mobile Computing, IEEE Transactions on*, v. 9, n. oct. 2010., p. 1377–1390, 2010.

KERÄNEN, A.; OTT, J.; KÄRKKÄINEN, T. The ONE Simulator for DTN Protocol Evaluation. In: *SIMUTools '09*. [S.l.]: ICST, 2009. ISBN 978-963-9799-45-5.

KHABBAZ, M.; ASSI, C.; FAWAZ, W. Disruption-tolerant networking: A comprehensive survey on recent developments and *Communications Surveys & . . .*, v. 14, n. 2, p. 607–640, 2012.

KOTILAINEN, N. P.; KURHINEN, J. A Genetic-Neural Approach for Mobility Assisted Routing in a Mobile Encounter. In: *The 5th International Conference on Information Technology and Applications (ICITA)*. [S.l.: s.n.], 2008. ISBN 9780980326727.

KUMAR, B. P. V.; VENKATARAM, P. Prediction-based location management using multilayer neural networks. *Journal of Indian institute of . . .*, 2002. Disponível em: <<http://journal.library.iisc.ernet.in/vol200201/paper1/vijay.pdf>>.

KUROSE, J. F.; ROSS, K. W. *Computer Networking: A Top-Down Approach*. 6th. ed. [S.l.]: Pearson Addison-Wesley, 2012. ISBN 978-0132856201.

LEGUAY, J. *Heterogeneity and Routing in Delay Tolerant Networks*. Tese (Doutorado) — Universit ´e Paris VI - Pierre et Marie Curie, 2007.

LI, X. et al. DTN Routing in Vehicular Sensor Networks. In: *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*. [S.l.: s.n.], 2008. p. 1–5. ISBN 9781424423248. ISSN 1930-529X.

LINDGREN, A.; DORIA, A.; SCHELÉN, O. Probabilistic routing in intermittently connected networks. In: *SIGMOBILE Mobile Computing and Communication Review*. New York, NY, USA: ACM, 2003. v. 7, n. 3, p. 19–20. ISSN 1559-1662. Disponível em: <<http://doi.acm.org/10.1145/961268.961272>>.

LIU, C.; WU, J. Practical routing in a cyclic mobispace. *IEEE/ACM Trans. Netw.*, IEEE Press, Piscataway, NJ, USA, v. 19, n. 2, p. 369–382, abr. 2011. ISSN 1063-6692. Disponível em: <<http://dx.doi.org/10.1109/TNET.2010.2079944>>.

MAMOUN, M. H.; BARRAK, S. E. Proposed Fuzzy based Routing for DTN. v. 66, n. 6, p. 19–22, 2013.

Queue-management architecture for delay tolerant networking. In: MASIP-BRUIN, X. et al. (Ed.). *Wired/Wireless Internet Communications*. [S.l.: s.n.], 2011, (Lecture Notes in Computer Science, v. 6649). ISBN 978-3-642-21559-9.

MENDES, D. Q. et al. *Tutorial de Redes Neurais: Aplicações em Bioinformática*. Laboratório de Bioinformática do Laboratório Nacional de Computação Científica, 2012. Disponível em: <<http://www.lncc.br/labinfo/tutorialRN/>>.

MISA, T. J.; FRANA, P. L. An interview with edsgar w. dijkstra. *Commun. ACM*, ACM, New York, NY, USA, v. 53, n. 8, p. 41–47, ago. 2010. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/1787234.1787249>>.

MISRA, S.; WOUNGANG, I.; MISRA, S. *Guide to Wireless Ad Hoc Networks*. Springer, 2009. (Computer Communications and Networks). ISBN 9781848003286. Disponível em: <http://books.google.com.br/books?id=bP_yvwiUPnUC>.

PELUSI, L.; PASSARELLA, A.; CONTI, M. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *Communications Magazine*, . . . , n. November, p. 134–141, 2006.

PÖGEL, T. Optimized DTN-Routing for Urban Public Transport Systems. In: *17th GI/ITG KiVS 2011*. [S.l.: s.n.], 2011. v. 17, p. 227–232. ISBN 978-3-939897-27-9. ISSN 2190-6807.

RAMANATHAN, R.; BASU, P.; KRISHNAN, R. Towards a formalism for routing in challenged networks. In: *Proceedings of the Second ACM Workshop on Challenged Networks*. New York, NY, USA: ACM, 2007. (CHANTS '07), p. 3–10. ISBN 978-1-59593-737-7. Disponível em: <<http://doi.acm.org/10.1145/1287791.1287794>>.

ROLLA, V. G.; CURADO, M. A reinforcement learning-based routing for delay tolerant networks. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 26, n. 10, p. 2243–2250, nov. 2013. ISSN 09521976. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0952197613001450>>.

SEGUNDO, F.; FARINES, J.-M.; SILVA, E. Silveira e. Quasi-opportunistic contact prediction in delay/disruption tolerant network. In: *Global Information Infrastructure Symposium, 2013*. [S.l.: s.n.], 2013. p. 1–6.

SEGUNDO, F.; SILVA, E. Silveira e; FARINES, J.-M. Predicting journeys for dtm routing in a public transportation system. In: *Wireless and Mobile Computing, Networking and Communications (WiMob), 2014 IEEE 10th International Conference on*. [S.l.: s.n.], 2014. p. 494–499.

SHAO, Y.; LIU, C. Delay-Tolerant Networks in VANETs. In: *Vehicular Networks: From Theory to Practice*. [s.n.], 2009. p. 1–36. Disponível em: <<http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Delay-Tolerant+Networks+in+VANETs#8>>.

SOCIEVOLE, A. et al. Routing approaches and performance evaluation in delay tolerant networks. In: *Wireless Telecommunications Symposium (WTS), 2011*. [s.n.], 2011. p. 1–6. ISBN 9781457701610. ISSN 1934-5070. Disponível em: <<http://ieeexplore.ieee.org/xpls/abs.all.jsp?arnumber=5960870>>.

SPYROPOULOS, T.; PSOUNIS, K.; RAGHAVENDRA, C. S. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In: *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*. New York, NY, USA: ACM, 2005. (WDTN '05), p. 252–259. ISBN 1-59593-026-4. Disponível em: <<http://doi.acm.org/10.1145/1080139.1080143>>.

SPYROPOULOS, T.; PSOUNIS, K.; RAGHAVENDRA, C. S. Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. In: *Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops*. Washington, DC, USA: IEEE Computer Society, 2007. (PERCOMW '07), p. 79–85. ISBN 0-7695-2788-4. Disponível em: <<http://dx.doi.org/10.1109/PERCOMW.2007.108>>.

UMASSAMHERST, T. S. *Maps: UMassAmherst Transit Services*. Acesso em 25/04/2014: UMassAmherst Transit Services, 2014. Disponível em: <<http://www.umass.edu/transit/maps.html>>.

VAHDAT, A.; BECKER, D. *Epidemic Routing for Partially-Connected Ad Hoc Networks*. [S.l.], abr. 2000. Disponível em: <<http://issg.cs.duke.edu/epidemic/epidemic.pdf>>.

WANG, H. et al. A Contact Prediction Method for DTN s Based on BP Artificial Neural Network. 2014.

WARTHMAN, F. et al. *Delay-Tolerant Networks (DTNs)*. [S.l.], 2013. Disponível em: <<http://www.dtnrg.org/docs/tutorials/warthman-1.1.pdf>>.

XUAN, B. B.; FERREIRA, A.; JARRY, A. Computing shortest, fastest, and foremost journeys in dynamic networks. *International Journal of Foundations of Computer Science*, v. 14, n. 2, p. 267–285, 2003. Disponível em: <<http://www-apr.lip6.fr/~buixuan/files/BFJ03.pdf>>.

YUAN, Q.; CARDEI, I.; WU, J. An efficient prediction-based routing in disruption-tolerant networks. *Parallel and Distributed Systems, IEEE Transactions on*, v. 23, n. 1, p. 19–31, Jan 2012. ISSN 1045-9219.

ZHANG, G. P.; BERARDI, V. L. Time series forecasting with neural network ensembles: an application for exchange rate prediction. *Journal of the Oper. Research Society*, v. 52, n. 6, p. 652–664, jun. 2001.

ZHANG, X. et al. Study of a bus-based disruption-tolerant network: Mobility modeling and impact on routing. In: *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*. New York, NY, USA: ACM, 2007. (MobiCom '07), p. 195–206. ISBN 978-1-59593-681-3. Disponível em: <<http://doi.acm.org/10.1145/1287853.1287876>>.

ZHANG, Z. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges. *Communications Surveys Tutorials, IEEE*, v. 8, n. 1, p. 24–37, 2006. ISSN 1553-877X.

APÊNDICE A – Redes Neurais Artificiais

Uma revisão sobre Redes Neurais Artificiais para apoiar o entendimento da proposta dessa tese é apresentada neste apêndice. É apresentada uma introdução com um breve histórico e principais definições. Depois são apresentados os tipos e como funciona o processo de aprendizagem sem aprofundamento nos métodos envolvidos.

A.1 INTRODUÇÃO

Redes Neurais Artificiais (RNAs) são definidas através de técnicas computacionais que tentam modelar matematicamente a estrutura neural de organismos inteligentes, adquirindo conhecimento através da experiência. Apesar da evolução da área de RNAs, mesmo com RNAs complexas e grandes (centenas ou milhares de unidades de processamento), ainda se está longe de chegar ao desempenho do cérebro de um mamífero, sendo que este pode ter muitos bilhões de neurônios (HERTZ; PALMER; KROGH, 1991; HAYKIN, 1994; ZHANG; BERARDI, 2001).

Se aproximar do funcionamento de um sistema nervoso não é uma tarefa simples, pois ele é formado por neurônios, constituindo um conjunto interligado extremamente complexo de células. O raciocínio e o comportamento do corpo humano dependem fundamentalmente do funcionamento dos neurônios. Os neurônios são formados pelos dendritos, que são um conjunto de terminais de entrada, pelo corpo central, e pelos axônios que são longos terminais de saída.

A primeira publicação a respeito da modelagem de um neurônio foi apresentada por McCulloch e Pitts em 1943. Hebb apresentou em 1949 um modelo básico de rede de auto-organização e Roseblatt, em 1958, apresentou o modelo Perceptron de aprendizado supervisionado. Após isso, depois de duas décadas com trabalhos menos conhecidos e em áreas próximas (visão, memória, controle e auto-organização), em 1982, Hopfield relatou em 1982 o emprego de RNAs simétricas para otimização e no mesmo ano, um método de aprendizagem supervisionada de retropropagação (*Backpropagation*) foi concebido por Rumelhart, Hinton e Williams.

A.2 FUNCIONAMENTO

Nas RNAs, as unidades que realizam o processamento são os neurônios. Assim, através da interligações dos neurônios uma rede pode ser capaz de armazenar o conhecimento derivado de experimentações e

torná-lo disponível para uso posterior (HAYKIN, 1994).

A exemplo do neurônio biológico, o neurônio modelado matematicamente também recebe sinais de entrada e retorna uma saída de acordo com uma função de ativação. A Figura 29 apresenta o modelo de neurônio definido por McCulloch-Pitts e que é utilizado na maioria das RNA atuais.

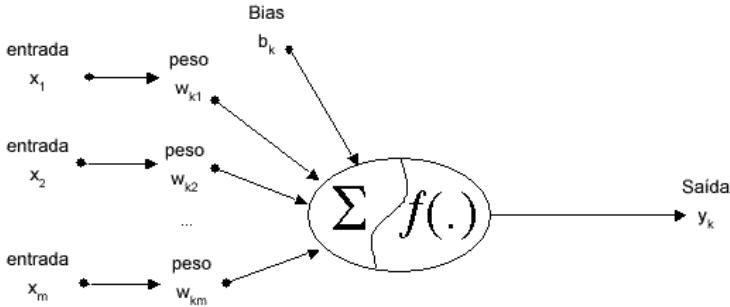


Figura 29: Modelo de um neurônio artificial (MENDES et al., 2012)

Seu funcionamento consiste em somar o **bias** e as entradas x_n já multiplicadas pelos seus pesos (w_{kn}) e submetê-la a função de ativação para obter como resultado a saída y_k . Conforme a ilustração, a saída do neurônio k seria: $y_k = f(\text{bias} + \sum_{n=1}^m x_n \cdot w_{kn})$.

Dessa forma, quando esse neurônio k for ligado ao neurônio (p.e. j) da próxima camada, y_k irá se tornar uma das entradas x_n do neurônio j e, por sua vez, a soma das entradas ponderadas pelos respectivos pesos sinápticos w do neurônio j irá ser submetida a função de ativação desse neurônio e será obtida uma nova saída $y_j = f(\text{bias} + \sum_{n=1}^i x_n \cdot w_{jn})$. Isso ocorre consecutivamente até o neurônio da camada de saída.

Uma função de ativação $f(\cdot)$ tipicamente utilizada é a função tangente hiperbólica que retorna um valor entre 0 e 1. Algumas outras funções também podem ser utilizadas, entre elas a função linear e a logística. Um exemplo de função logística é $f(x) = \frac{1}{1+e^{\beta x}}$. Outro exemplo é uma função tangente hiperbólica $f(x) = a \frac{e^{\beta x} - e^{-\beta x}}{e^{\beta x} + e^{-\beta x}}$, onde β é um parâmetro pré-definido de inclinação da curva. A função de ativação vai dependendo da faixa dos valores de entrada e saída desejados para o neurônio.

Quando vários neurônios são conectados e organizados eles for-

mam uma rede de neurônios artificiais (RNA). Para cada grupo de neurônios (i.e. camada, como mostrado na Figura 30) é normalmente definida uma função de ativação.

Na arquitetura de uma RNA é definida a topologia em que os neurônios serão interligados. Dependendo da estrutura do problema que se deseja resolver, a RNA pode assumir topologias diferentes. As principais topologias conhecidas são (HAYKIN, 1994):

- RNA alimentada para frente (*feedforward*): não há ciclos nas ligações entre os neurônios, ou seja, o sinal sempre vai da entrada em direção à saída. Quando os neurônios são ligados em camadas, a saída do neurônio é sempre conectada a entrada dos neurônios da próxima camada, nunca ligando-se a um neurônio de uma camada anterior.
- RNA recorrente: se distingue das anteriores se possuir pelo menos um laço de realimentação. Deve ter pelo menos, uma camada de entrada e uma de saída, com cada neurônio tendo seu sinal de saída alimentando a entrada de todos os outros neurônios. Isso permite que os resultados na saída possam referir-se a um determinado contexto representado por uma entrada.

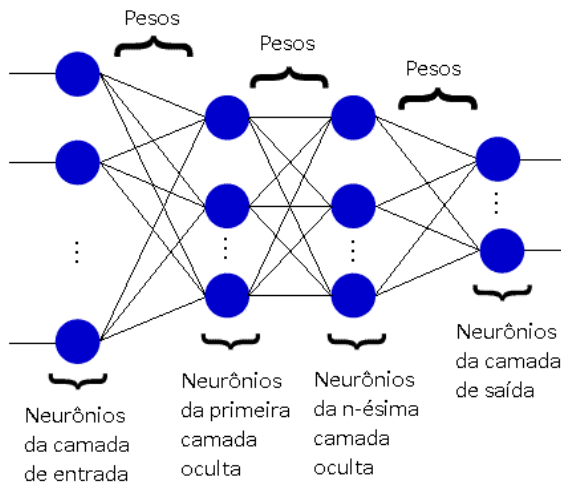


Figura 30: Organização dos neurônios em uma rede neural multicamadas (KUMAR; VENKATARAM, 2002)

A.3 CARACTERÍSTICAS

As características neurobiológicas que são imitadas na RNA são:

- Um processo de aprendizagem deve ser executado para a rede adquirir conhecimento;
- As ligações entre neurônios ganham ou perdem força ou peso no processo de aprendizagem. Esses pesos representam o conhecimento adquirido;
- Um determinado conhecimento é melhor fixado quanto maior for o número de vezes que ele for usado na aprendizagem. Os pesos se ajustam para representar esse conhecimento.

Contudo, pode acontecer de uma rede não conseguir representar o conhecimento desejado. As principais razões para isso são as seguintes (MENDES et al., 2012):

- mau aprendizado: os exemplos podem ser insuficientes; não significativos ou levar a memorização, mas não constituir generalizações;
- má definição da entrada: as informações de entrada não capturam o conhecimento que se quer extrair;
- má definição da arquitetura: a topologia não é adequada para o tipo de conhecimento que se deseja extrair.

As principais características da RNA são:

- Generalização: após informar um conjunto de exemplos, se consultada, a rede apresenta respostas coerentes sobre um valor não aprendido.
- Adaptabilidade: os pesos sinápticos são alterados quando novos exemplos são aprendidos, permitindo adaptar-se ao um novo ambiente. As RNAs podem ser projetadas para alterarem seus pesos em tempo real ou para operarem em ambientes que se modificam com o tempo.
- Informação contextual: uma vez que o processamento de um neurônio afeta outros por causa das suas ligações, a RNA processa as informações contextuais de forma natural.

- Uniformidade: tendo em vista que os neurônios estão presentes em todas as redes neurais conhecidas, a mesma estrutura básica é utilizada em variados domínios de aplicação. Assim, é possível utilizar os mesmos algoritmos de aprendizagem e teorias em diversas aplicações. Além disso é possível trabalhar com módulos e realizar integrações.

Essas características tornam as RNA satisfatoriamente aplicáveis a problemas não adequados à computação tradicional. De forma geral são aplicadas em tarefas de classificação, agrupamento, reconhecimento de padrões, regressão numérica e predição.

Uma RNA não será útil se não puder representar o conhecimento. Dessa forma, os algoritmos de aprendizagem são fundamentais para a sua aplicabilidade. O algoritmo de aprendizagem está muito relacionado com a arquitetura da RNA, pois ele deve permitir ajustar os pesos dos neurônios dentro da arquitetura definida.

Contudo, não é uma tarefa trivial estabelecer uma RNA para tratar um problema. São algumas dificuldades relacionadas a aplicação das RNAs:

- a modelagem depende da análise consistente de um sistema muitas vezes complexo, tornando difícil:
 - a escolha da arquitetura mais eficiente para o tratamento de determinado problema;
 - a escolha dos dados de entrada que sejam realmente relevantes também pode ser difícil;
 - a definição dos pesos sinápticos ideais, por causa da definição de uma forma apropriada de aprendizagem e critérios de parada do treinamento;
 - a definição dos níveis de bias.
- a dificuldade da extração de regras que justifiquem a decisão tomada pela rede;
- a dificuldade da definição da forma de aprendizagem ideal e os parâmetros de aprendizagem;

Neste sentido, os pesquisadores vem desenvolvendo algoritmos extratores de regras, tais como algoritmo KBANN (*Knowledge Base Neural Networks*) e sistemas híbridos que combinam técnicas complementares. Um exemplo de sistema híbrido consiste em combinar RNA com algoritmos genéticos para otimizar a taxa de aprendizado, pesos sinápticos e níveis de bias.

A.4 TIPOS DE RNA

As RNAs são baseadas em abordagens distintas que podem ser mais apropriadas para realizar determinadas funções. As principais abordagens são as seguintes (MENDES et al., 2012):

- Baseadas na topologia alimentada para frente:

- simples (*single layer*): existe apenas os neurônios da entrada (que recebem os estímulos externos) ligados na entrada do neurônio de saída, ligado a saída externa. São indicados para tratar problemas da classe linear.

- de múltiplas camadas (MLP): entre os neurônios da entrada e da saída deve existir uma ou mais camadas intermediários de neurônios, chamadas camadas ocultas. Com uma camada oculta é possível se obter um aproximador universal de funções e tratar alguns problemas da classe não-linear. Com duas camadas é possível tratar a grande maioria dos problemas da classe não linear. Quanto mais camadas existirem, maior é a dificuldade para se obter melhores generalizações. Quanto maior é o número de camadas ocultas e de neurônios nessas camadas, maior é a capacidade de classificação e armazenamento. Contudo, o número de operações no processamento aumenta. Uma forma de aprendizado realizado neste modelo de organização da arquitetura é o aprendizado supervisionado por correção de erro, implementado pelo algoritmo de retropropagação. Com esse algoritmo, o ajuste dos pesos tem maior influência nos neurônios mais próximos da camada de saída.

- de função de base radial (*Radial Basis Function - RBF*): essa ANN tem duas camadas de processamento (entrada e oculta) e uma camada de resumo (saída). Na camada de entrada, cada entrada é mapeadas em cada neurônio da camada oculta, em que os neurônios possuem como função de ativação uma função de base radial (normalmente gaussiana). Essa função tem embutido um critério de distância em relação a um centro. Apresenta a vantagem de não sofrer tanto com o problema de mínimos locais na solução e a desvantagem de necessitar cobrir todo o espaço das variáveis de entrada.

- Baseadas na topologia recorrente (*recurrent*):

- simples: uma pequena modificação nas ANN alimentadas para frente usando um neurônio para definir o contexto re-alimentado com sua entrada ligada na saída de um neurônio da camada intermediária ou de saída.
- de Hopfield: também conhecidas como Memória Associativa. A partir de estímulos de entrada, são recuperados padrões armazenados nessas redes. Esses padrões são adquiridos através de aprendizado Hebbiano. Nessas redes as saídas dos neurônios são ligadas nas entradas dos outros neurônios por um atraso de tempo, criando uma recorrência e, como consequência disso, a resposta da rede sempre depende de seu resultado anterior.
- de Kohonen: conhecidas como Mapas Auto Organizáveis, são baseadas nos mapas corticais. Eles utilizam o aprendizado competitivo, uma forma de aprendizagem não supervisionada no qual os neurônios competem entre si para responder a um estímulo que é apresentado. Neste processo de aprendizagem, os neurônios vão se agrupando organizados de forma topológica, sendo que cada grupo responde por uma classe de estímulos diferente. No aprendizado competitivo a única informação apresentada à rede são os padrões de entrada. As ligações sinápticas vão sendo estabelecidas de forma a recompensarem o neurônio comparável com os padrões desejados, chamado de neurônio vencedor. Isso confere a esse modelo de rede a capacidade de auto-organização.

Além dessas, existem outros tipos de RNA (KUMAR; VENKATARAM, 2002).

A.5 APRENDIZAGEM

A aprendizagem em RNA tem por objetivo a obtenção de um modelo implícito do sistema em estudo através do ajuste dos pesos sinápticos da rede (FAUSETT, 1994; HAYKIN, 1994).

Considerando $\mathbf{w}(\mathbf{k})$ como sendo o peso sináptico de um dado neurônio de uma RNA no instante de tempo \mathbf{k} , o aprendizado se dá através do ajuste $\Delta\mathbf{w}(\mathbf{k})$ aplicado a $\mathbf{w}(\mathbf{k})$, gerando $\mathbf{w}(\mathbf{k} + 1)$, estabelecendo a equação $\mathbf{w}(\mathbf{k} + 1) = \mathbf{w}(\mathbf{k}) + \Delta\mathbf{w}(\mathbf{k})$.

Existem várias formas de se obter $\Delta\mathbf{w}(\mathbf{k})$. A técnica utilizada no ajuste dos pesos da RNA determina o tipo de aprendizado. O algoritmo utilizado no aprendizado deve estabelecer um conjunto de regras

bem definido para obter os pesos. São exemplos os algoritmos de retropropagação, estratégias de competição, máquina de Boltzmann e lei de Hebb.

A abordagem de aprendizagem é definida pelo modo que o ambiente influencia a RNA em seu aprendizado. Exemplos dessas abordagens são:

- o aprendizado supervisionado: é apresentado à RNA um conjunto de treino constituído por exemplos, por sua vez formados pelas entradas e saídas correspondentes. Essa saída representa o comportamento esperado da RNA, sendo que a comparação da saída gerada pela RNA com a saída desejada permitirá obter informação para o ajuste dos pesos sinápticos.
- o aprendizado não-supervisionado: a rede atualiza seus pesos sem o uso de pares entrada-saídas desejadas e sem indicações sobre a adequação das saídas produzidas. Como resultado, as saídas representam o aprendizado de categorias de acordo com as entradas fornecidas;
- o aprendizado por reforço: algum critério numérico fornecido de tempos em tempos é usado para avaliar o comportamento da rede;

Uma representação do conhecimento é obtida ao alcançar o objetivo, independentemente de qual for o algoritmo ou abordagem utilizado. Essa representação obedece a uma ou mais regras:

- 1.mais neurônios e conexões devem ser empregados quanto maior for a importância de uma característica. Quanto mais complexa é a representação do conhecimento, maior deve ser o número de neurônios e conexões para representá-lo;
- 2.se informações conhecidas antecipadamente e constantes forem consideradas no projeto da RNA o processo de aprendizagem poderá ser simplificado.
- 3.representações similares dentro da RNA tendem a ser produzidos por dados de entrada provenientes de eventos ou fenômenos de classes parecidas;
- 4.exemplos que devem ser processados ou classificados de forma distinta devem provocar representações distintas dentro da RNA;

Os conceitos de distância ou similaridade utilizam as duas últimas regras. Esses conceitos podem ser expressos matematicamente a partir de uma definição formal de critério de medida.

Existem uma série de problemas que podem ser representados como um problema de aprendizado supervisionado :

- predição de séries temporais;
- classificação e reconhecimento de padrões;
- projeto de filtro em processamento de sinais;
- controle de processos;
- identificação de sistemas;

A aprendizagem supervisionada pode ser vista como um problema de otimização não-linear, em que o conjunto de pesos \mathbf{W} devem ser ajustados por uma função objetivo $\min_{\mathbf{W}} J(\mathbf{W})$.

Um exemplo de dado de treinamento no instante \mathbf{k} é formado por $\mathbf{x}(\mathbf{k})$ e $\mathbf{d}_j(\mathbf{k})$. Para o treino, considere $\mathbf{d}_j(\mathbf{k})$ a saída desejada do neurônio j no instante \mathbf{k} usando a entrada $\mathbf{x}(\mathbf{k})$ e $\mathbf{y}_j(\mathbf{k})$ a saída gerada pelo neurônio j no instante \mathbf{k} usando a entrada $\mathbf{z}(\mathbf{k})$. Quando $\mathbf{z}(\mathbf{k})$ é obtido em um ambiente ruidoso com distribuições de probabilidade desconhecidas, $\mathbf{e}_j(\mathbf{k}) = \mathbf{d}_j(\mathbf{k}) - \mathbf{y}_j(\mathbf{k})$ é o erro obtido comparando o valor desejado com o obtido na saída do neurônio j no instante \mathbf{k} . Em um ambiente ruidoso, $\mathbf{e}_j(\mathbf{k})$ é uma variável aleatória.

A correção desse erro é o objetivo do processo de aprendizagem supervisionada. Esse erro é observado em todos os neurônios, assim o ajuste deve ser feito para todos os j neurônios minimizando $\mathbf{e}_j(\mathbf{k})$ através de uma função objetivo, mantendo estatisticamente o mais próximo possível de $\mathbf{d}_j(\mathbf{k})$.

A função objetivo baseada no erro quadrático médio $\mathbf{J} = \mathbf{E}[\frac{1}{p} \sum_{j=1}^p \mathbf{e}_j^2(\mathbf{k})]$ é frequentemente utilizada, considerando que os processos aleatórios são estacionários. Para minimizar \mathbf{J} , ainda é necessário conhecer as características estatísticas do sistema. Assim, o erro quadrático médio instantâneo é utilizado para obter uma aproximação para o critério: $\mathbf{J} \cong \mathbf{J}(\mathbf{k})$.

O processo de aprendizagem ou treinamento da RNA em MLP pode ser entendido como um caso especial de aproximação de funções, em que não é considerado nenhum modelo explícito dos dados. Existem vários algoritmos de treinamento aplicando diferentes métodos para estabelecer a função objetivo. São exemplos de algoritmos de treinamento:

- retropropagação (*Backproppagation* - BP);

- método do gradiente (GRAD);
- método do gradiente conjugado (GC);
- método do gradiente conjugado escalonado (SCG);
- método de Newton (MN);
- Levenberg-Marquardt (LM);
- retropropagação rápida (Quickprop - QP);
- retropropagação resiliente (*Resilient Backpropagation* - RProp);

Os algoritmos de treinamento treinam a RNA em ciclos ou épocas. A cada ciclo uma determinada quantidade de exemplos tem seus estímulos submetidos a entrada da RNA e as saídas obtidas são comparadas com saídas desejadas dos exemplos. A quantidade de exemplos utilizada a cada ciclo pode ser de forma incremental: um exemplo por ciclo; em lotes pequenos: um determinado número de exemplos por ciclo; ou em um lote grande: apresentando todos os exemplos em um mesmo ciclo.

Nos ensaios dessa pesquisa foram utilizados os algoritmos de aprendizagem BP e RP e determinações mínimas e máximas do número de ciclos de treinamento, normalmente em lotes pequenos.

A.6 CONCLUSÃO

Apesar da concepção da RNA ser da década de 40 e 50, somente das últimas décadas foi possível desenvolver algoritmos complexos e se ter poder computacional para aproveitar melhor seu potencial e permitir mais avanços na área.

Tarefas de classificação, predição e aproximação de funções são exemplos de aplicação das RNAs, permitindo modelar funções não linearmente separáveis.

As RNAs FF MLP são bastante conhecidas e com ampla aplicação, sendo que já foram realizadas diversas pesquisas e aplicações apresentando resultados satisfatórios. Contudo a etapa de projeto ainda exige muito estudo e testes, sendo que não são encontrados métodos amplamente testados e conhecidos para facilitar essa etapa.

A aprendizagem não é uma etapa trivial, dependendo muito dos dados que se tem disponível, do algoritmo de treinamento, da inicialização dos pesos e de muitos testes para se conseguir critérios satisfatórios de parada do treinamento.

APÊNDICE B - Dados de Contatos no STPU Utilizados

Para este trabalho foi avaliada uma série de conjuntos de dados. Devido à necessidade de se obter a maior quantidade possível de dias de registro e registros no nível de contato, apenas um dos projetos pesquisados foi selecionado para os testes.

A grande quantidade de dias registrado favoreceu a obtenção de mais informações para serem aproveitadas pelo método a ser utilizado para predição. O registro dos contatos evitou a realização de simulações no nível de rede, o que diminuiria o realismo na realização dos experimentos.

Uma vez que foi difícil obter relações úteis entre as informações desse conjunto de dados, inicialmente foram gerados dados sintéticos baseado na movimentação dos ônibus do STPU para obter os primeiros resultados indicando em quais direções o trabalho poderia prosseguir.

Ambos os conjuntos de dados utilizados, real e sintético, são apresentados a seguir.

B.1 DADOS REAIS DO DIESELNET

Os dados de contato do projeto UMassDieselNet (DIESELNET, 2014) foram escolhidos devido à execução de movimentos repetitivos realizados pelos ônibus e o grande período de dias em que esses dados foram registrados. Além disso, são dados mais realistas pois já são o resultado da comunicação entre os dispositivos, não necessitando de simulação da comunicação. Essa quantidade de dados é fundamental para prover exemplos de treinamento para as RNAs.

Vários trabalhos sobre DTN já utilizaram esse conjunto de dados, entre estes (BURGESS et al., 2006; ZHANG et al., 2007; LIU; WU, 2011; PÖGEL, 2011; DOERING; PöGEL; WOLF, 2010; BALASUBRAMANIAN; LEVINE; VENKATARAMANI, 2010), tornando interessante a produção de pesquisas para realização de comparações entre estratégias e métodos testados sobre o mesmo conjunto de dados.

No sistema de ônibus avaliado no DieselNet existem 10 rotas seguidas por 40 ônibus servindo uma área de aproximadamente 300 quilômetros quadrados na localidade de Amherst, Massachusetts, Estados Unidos, conforme o mapa da Figura 31. Foram registrados contatos entre dispositivos de comunicação padrão Wireless-Fidelity (WiFi) instalados nos ônibus durante vários meses em quatro períodos diferentes entre 2005 e 2007. Os conjuntos de dados gerado podem ser encontrado na internet no arquivo de recursos CRAWDAD (BURGESS et al., 2008).

Os registros dos quatro períodos de coleta de dados foram os

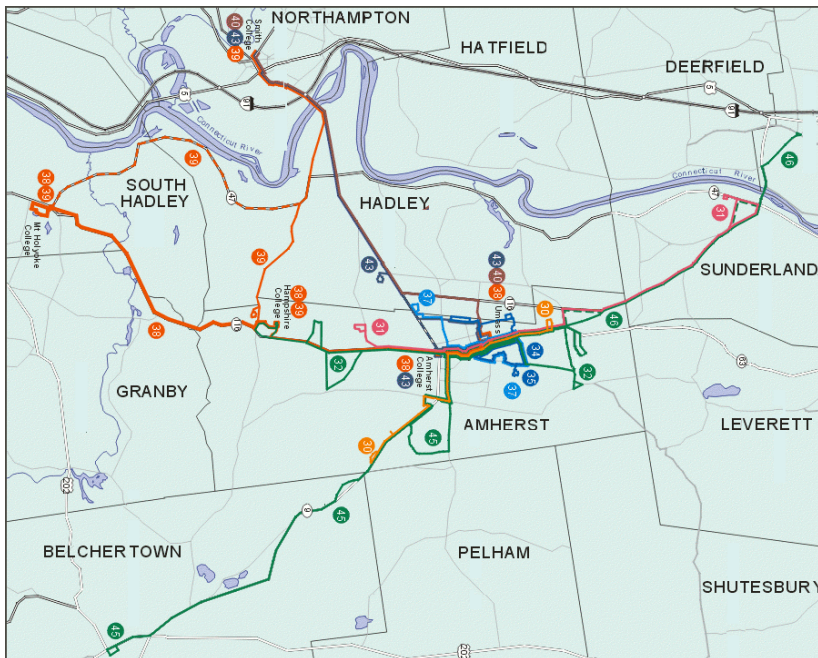


Figura 31: Rotas de ônibus na área de serviço do UMass Transit Service (UMASSAMHERST, 2014).

seguintes acompanhados das respectivas observações sobre os dados de contato:

- Primavera de 2005: o conjunto de dados foi obtido entre 25/01/2005 e 22/06/2005, não apresenta informação de rota e apresenta problemas na informação de GPS.
- Primavera de 2006: o conjunto de dados foi obtido entre 30/01/2006 e 22/06/2006. Ele apresenta informação de rota inválida no arquivo principal, mas rotas válidas foram digitadas em um arquivo separado e sujeitas a inconsistências. Ele apresenta problemas na informação de GPS.
- Primavera de 2007: o conjunto de dados foi obtido entre 06/02/2007 e 14/05/2007, não há informação de rota e apresenta menos problemas nas informações de GPS do que o conjuntos de dados anteriores.

- Outono de 2007: o conjunto de dados foi obtido entre 22/10/2007 e 16/11/2007, não há informação de rota e apresenta menos problemas nas informações de GPS do que o conjuntos de dados anteriores.

Os dados utilizados neste trabalho foram do conjunto obtido na primavera de 2006, pois a informação das rotas apresentaram-se importantes para estabelecer relações entre contatos em dias diferentes. Os registros de contato somente entre ônibus (sem considerar outros pontos de acesso) foram salvos em arquivos no formato MDDAAAA (Mês com um dígito, Dia com dois dígitos e Ano com quatro dígitos) e cada linha do registro indica uma conexão TCP. Conforme o exemplo de linha:

```
Bus 3112 at 72.532745 42.393852 on route 1 in contact with
bus 3114 at 72.532745 42.393852 on route 1 at time 418:27
for 45204688 bytes in 184792.0 ms
```

, temos o registro de uma conexão TCP do equipamento de comunicação instalado no ônibus 3112 executando a rota 1 nas coordenadas de GPS (72,532745, 42,393852) com o equipamento no ônibus 3114 executando a rota 1 nas coordenadas de GPS (72,532745, 42,393852) no tempo absoluto de 418 e 27 segundos após a meia-noite, transferindo 45204688 bytes em 184792 milissegundos.

Existem dois arquivos para registro de informação das partidas dos ônibus:

- DB_ sheet.txt: registro manual por linha dos horários e partidas de cada ônibus em cada ponto de parada, rotas e troca de motorista.
- DA_ all.txt: é um resumo do DB_ sheet.txt;

Alguns problemas com dados corrompidos, tal como os de GPS, falta de registro correto de informação em cada contato, tal como a informação de rota e assimetria no registro da conexão TCP entre dois ônibus foram verificadas. Em uma atualização recente das informações de descrições dos conjuntos de dados em (BURGESS et al., 2008) há uma explicação para o problema da assimetria nos contatos. É informando ser devido à problemas relativo a velocidade da conexão TCP quanto os ônibus estão se movimentando em direções opostas e/ou da reativação lenta do dispositivo de comunicação devido à configuração inadequada da pilha de protocolo nos *drivers* do Linux utilizado nos instalados nos ônibus.

Por esse motivo e para favorecer o aprendizado das RNAs, neste trabalho esse conjunto de dados necessitou ser pré-processado. De forma geral, o pré-processamento de dados é uma etapa importante nos trabalhos que realizam aprendizagem de máquina. Especialmente no caso de RNAs, essa etapa é fundamental para que se consiga obter vantagens de informações correlacionadas.

Técnicas convencionais de visualização em gráficos e grafos não facilitaram a busca por padrões nesse conjunto de dados. Contudo, muito estudo e o uso de ferramentas de mineração de dados permitiram encontrar indícios de relações entre variáveis que poderiam ser exploradas.

Uma parte do pré-processamento dos dados foi realizado no mesmo sentido que em (ZHANG et al., 2007), contudo o autor encontrou um melhor relacionamento entre contatos no nível de rotas enquanto neste trabalho foram encontradas no nível de linhas de ônibus. Uma rota pode estar presente em mais de uma linha de ônibus e uma linha de ônibus acontece somente uma vez no ciclo de movimentações, no caso no dia.

Nesse pré-processamento foi criada simetria nos contatos realizados, ou seja, se há um registro entre nós somente em um sentido o sentido oposto é adicionado no mesmo momento e duração no tempo. Então, contatos com um mesmo nó dentro de um curto intervalo de tempo foram agrupados em apenas um contato, ajustando-se o período de tempo inicial e final para englobar estes contatos. Cada registro do histórico do contato contém as informações: nó atual (ou de referência), nó contatado, momento do início do contato, momento do fim do contato, duração e bytes transferidos. O momento do último contato e o nó desse contato são incluídos para cada registro, objetivando auxiliar no relacionamento entre contatos subsequentes em dias diferentes, criando a relação: nó do contato anterior e nó do próximo contato com um nó de referência.

B.2 DADOS SINTÉTICOS BASEADO NA MOVIMENTAÇÃO DOS ÔNIBUS DO STPU

Com o objetivo de modelar os contatos ocorridos na movimentação dos ônibus do STPU foi criado um modelo sintético baseado no cenário de experimentação mostrado na Figura 32. Neste cenário, o modelo das ruas segue o modelo em topologia de grade, tal como no Manhattan (BAI; HELMY, 2006) e o modelo de movimentação é uma

variação do *Random Waypoint* (JOHNSON; MALTZ, 1996) considerando que a direção da movimentação é pré-estabelecida e em cada ponto de parada a velocidade até o próximo ponto de parada e o tempo de permanência nele são estabelecidos aleatoriamente. Foi estabelecido sobre uma área de 5,96 km² com ruas formando 32 x 32 quadras de tamanho regular com 180 m². Foram definidos 4 itinerários com percurso retangular que tem interseção entre si, com respectivamente 7.920, 9.360, 9.360 e 11.520 metros. Os itinerários apresentam pontos de paradas preestabelecidos com espaçamento de aproximadamente 1,3 a 1,8 km. Foram definidos 5 nós, sendo que dois deles seguem o mesmo itinerário e os demais tem seu itinerário exclusivo. Ao completar o itinerário, os nós iniciam novamente o mesmo itinerário. Cada nó possui um dispositivo de comunicação que tem alcance de 200 metros que é representado pelo círculo verde. Nesta figura as rotas foram sublinhadas por retângulos coloridos com pequenos traços que indicam os pontos de parada dos ônibus. A duração total da simulação foi de 12 horas (43200 segundos).

O software utilizado para simulação do cenário foi o *Opportunistic Network Environment* (ONE) (KERÄNEN; OTT; KÄRKKÄINEN, 2009). Para simular variações de velocidade e do tempos de permanência nos pontos de parada, foram inicialmente criadas 10 variações desse cenário. As variações consistem em alterar a faixa de velocidade que os nós podem escolher aleatoriamente para se deslocar de um ponto de parada até o próximo. Além disso, a faixa de tempo de permanência no ponto de parada também é variada, conforme descrito na Tabela 9. Cada uma das variações do cenário representa os contatos realizados em um novo ciclo, mas com condições de movimentação ligeiramente diferentes, tal como em dias com trânsito leve ou em dias com trânsito congestionado. O objetivo é que se permita simular a chegada e partida dos nós nos pontos de parada em horários próximos aos horários preestabelecidos em um itinerário. A utilização de somente os primeiros 14.400 segundos é adotada na falta de implementação de um mecanismo de sincronização da movimentação dos nós para o simulador. Com isso procurou-se reproduzir o início das atividades de um sistema de transporte urbano de ônibus com ônibus partindo de um mesmo local e tempo predefinidos. Alternativamente, cada caso pode ser visto como uma representação de movimentos de um diferente dia útil nos ônibus do sistema de transporte urbano.

Esta configuração de rotas foi escolhida para favorecer a ocorrência de contatos em interseções sem semáforos, denominados de encontros, pois são contatos mais difíceis de apresentar regularidades. Este

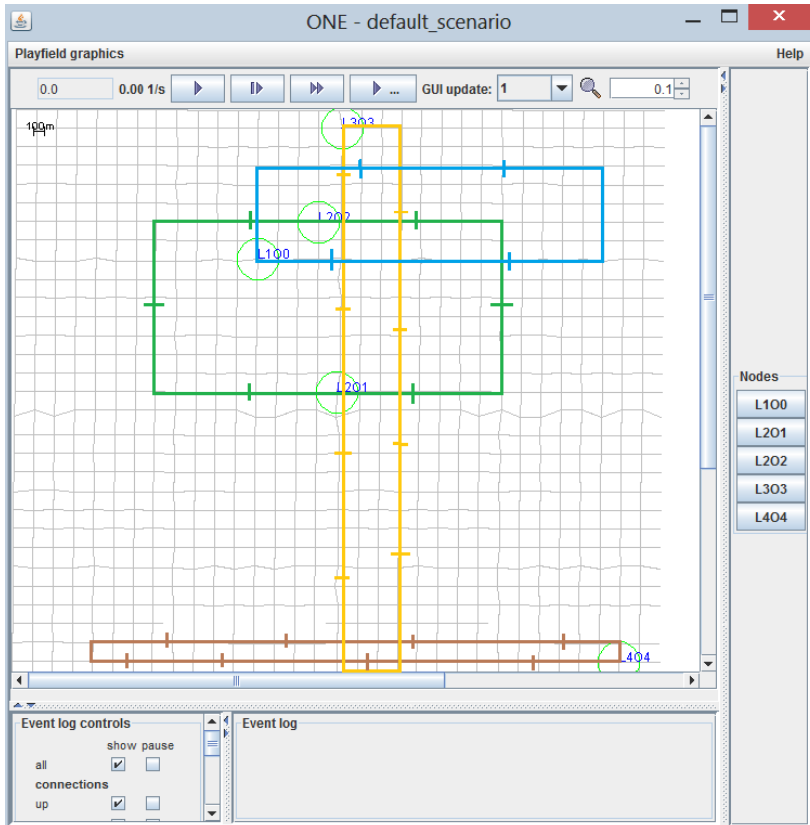


Figura 32: Tela do simulador de DTN ONE: Rotas e pontos de parada do cenário estão sublinhados.

tipo de contato é mais difícil de prever e é interessante avaliar como eles acontecem, mesmo considerando condições normais de operação da mobilidade dos nós e a regularidade das rotas, paradas e início de operação projetados para este cenário.

Os valores usados para as faixas de velocidade e tempo de permanência no ponto de parada apresentam pequenas variações que objetivam modelar dois aspectos importantes:

- pequenas variações na velocidade e no tempo de permanência nos pontos de parada modificam consideravelmente a oportunidade de contato, modificando o nó em contato e o momento do contato;

- um histórico de contato que contem muitas repetições do itinerário, assumindo faixas de velocidades e de permanência no tempo de parada próximas podem ser exploradas regularidades na modificações que trazem regularidades para o estabelecimento de contatos.

Tabela 9: Faixas de velocidade e tempo de permanência nos pontos de parada para 10 casos.

Caso	1	2	3	4	5
Vel.(m/s)	8	8	8	7 a 9	7 a 9
Parada(s)	20	18 a 22	15 a 25	18 a 22	20
Caso	6	7	8	9	10
Vel.(m/s)	7 a 8	7 a 8	8 a 9	8 a 9	7 a 9
Parada(s)	20	18 a 22	20	18 a 22	15 a 25

No simulador, as velocidades e tempos de permanência nas paradas são escolhidas aleatoriamente para cada trecho entre dois pontos de parada dentro de uma faixa predeterminada. A mesma semente aleatória, zero, foi usada em todos os casos para permitir a reprodução exata das mesmas condições no simulador e somente levar em conta a variação da faixa de velocidade e tempo de permanência nas paradas.

As simulações foram realizadas com estas configurações e os dados de contato foram obtidos. Através desses dados de contato objetivou-se entender o impacto da movimentação dos nós nas oportunidades de contato para cada variação do cenário.

A Figura 33 apresenta dois gráficos representando os contatos realizados pelo nó identificado por zero com outros nós através do tempo nos dez casos mostrados na Tabela 9. Cada caso é descrito na legenda e seus contatos são mostrados no gráfico em cores e formas diferentes. A linha azul que corta os dois gráficos identifica um período limite de 4 horas (14.400 segundos).

Esse limite de 4 horas é equivalente a um período de trabalho e foi estabelecido empiricamente para prevenir o acúmulo de variação de tempo que acaba prejudicando o cumprimento dos itinerários e não ocorre na realidade, por causa do sincronismo imposto pelos itinerários. Esse sincronismo, mesmo não sendo preciso na maioria dos STPU, ajudar a fornecer certa regularidade temporal. Depois de 4 horas, os movimentos gerados pela simulação tendem a não representar adequadamente os movimentos desejados para o contexto deste trabalho. Alternativamente, este controle poderia ser implementado em

um agendamento para parada dos nós, mas esse tipo de controle não é implementado na versão atual do ONE.

Considerando este limite, as análises de desempenho com esses contatos confirmaram os dois aspectos levantados. Por exemplo, na Figura 33 são mostrados pelo menos quatro conjuntos de linhas apresentando resultados similares em termos do tempo e da sequência de nós contactados. Estes conjuntos são: A) casos 1, 3, 4 e 7; B) casos 8 e 10; C) casos 7, 9 e 10, e; D) casos 2 e 6. Observa-se que um caso pode pertencer a mais do que um conjunto. Isso também é visto no conjunto D, o qual não tem contatos até alcançar o limite de 4 horas.

As considerações feitas são relativas ao nó zero e também foram observadas em todos os nós participantes do ensaio.

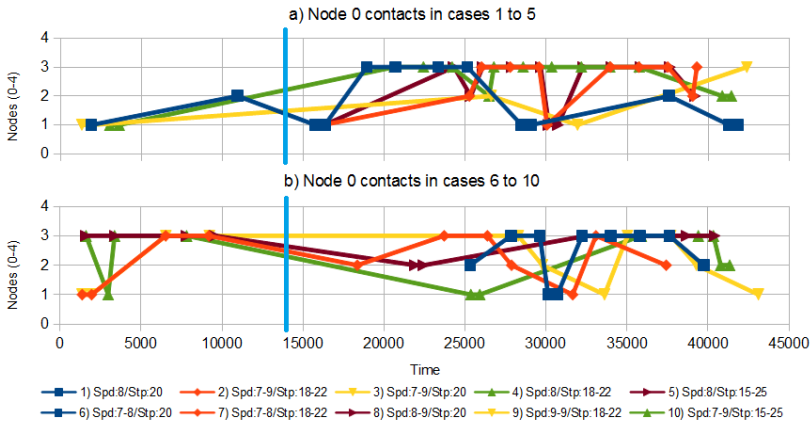


Figura 33: Gráfico de contatos para o nó 0 em 10 casos.

Modelo com 20 Casos

Para simular um número maior de casos de variações de velocidade e do tempos de permanência nos pontos de parada, foram criadas 20 variações do cenário simulado. O objetivo foi se ter mais casos para avaliar a variação nas oportunidades de contato de acordo com a variação da mobilidade dos nós.

Da mesma forma que no ensaio com 10 casos, as variações consistem em alterar a faixa de velocidade que os nós podem escolher aleatoriamente para se deslocar de um ponto de parada até o próximo. A faixa de tempo de permanência no ponto de parada também é variada,

Tabela 10: Velocidade e tempo de parada para 20 variações do cenário.

Variacão	1	2	3	4	5
Vel.(m/s)	7	8	9	7	8
Parada(s)	18	18	18	20	20
Variacão	6	7	8	9	10
Vel.(m/s)	9	7	8	9	7 a 9
Parada(s)	20	22	22	22	18 a 22
Variacão	11	12	13	14	15
Vel.(m/s)	7 a 8	8 a 9	7 a 9	7 a 8	8 a 9
Parada(s)	18	18	18	20	20
Variacão	16	17	18	19	20
Vel.(m/s)	7 a 9	7 a 8	8 a 9	7 a 9	7
Parada(s)	20	22	22	22	18 a 22

conforme descrito na Tabela 10.

Com mais casos, foi possível avaliar melhor o desempenho do preditor de contatos já aplicado ao construtor de jornadas.

B.3 OBSERVAÇÕES SOBRE OS CONTATOS NO STPU

No contexto de movimentação repetitiva e com itinerários pré-definidos existe uma grande probabilidade dos nós se encontrarem a cada repetição da movimentação. Utilizando o contexto dos ônibus de um transporte público urbano, imagina-se que exista grande regularidade nos contatos entre os ônibus, uma vez que seguem itinerários fixos.

Entretanto, ao contrário do que se imaginou sobre a movimentação dos ônibus do STPU, organizadas por itinerários e tabelas de horários, mínimas variações de velocidade e tempos de permanência nos pontos de parada tem muita influência na regularidade do estabelecimento dos contatos. De fato, por exemplo, 5 segundos a mais ou a menos na espera pelo embarque de passageiros ou uma variação na velocidade de poucos quilômetros por hora, normalmente muda substancialmente as oportunidades de contato.

O estudo das observações sobre o estabelecimento dos contatos nesse contexto realizado por Doering (2012) foi estendido para se tentar entender melhor como as situações que modificam a movimentação dos nós e influenciam na regularidade dos contatos.

Através da observação dos contatos reais e de simulações foram observados fatores que interferem isoladamente nas formas básicas de movimentação dos ônibus no STPU (DOERING, 2012). Esses fatores ajudam a entender melhor quais padrões de contatos podem ser estabelecidos no STPU. A seguir são apresentados quanto a forma de movimentação:

1. Encontro de nós:

- (a) O tamanho do trecho compartilhado da rota: quanto maior for o trecho compartilhado, maior é a expectativa de acontecer um contato.
- (b) Tempo de permanência no trecho compartilhado: quanto maior o tempo de permanência no trecho compartilhado mais é favorecido o encontro. O tráfego/congestionamento, o número de pontos de parada e a demanda pela entrada e saída de passageiros nesses pontos influenciam nesse fator;
- (c) Velocidade no trecho compartilhado: os contatos podem ser perdidos se a velocidade dos nós for alta.
- (d) Obstáculos para o sistema de comunicação: outros veículos, construções e desvios podem atrapalhar a realização de contatos e/ou diminuir a sua duração.

2. Cruzamentos:

- (a) nó em espera: o tamanho do trecho compartilhado é sempre pequeno não favorecendo o contato. Normalmente ocorrem filas próximas a interseção, o que aumenta as chances de os veículos ficarem próximos, diminui a velocidade e favorece a realização de contato e aumenta a sua duração;

3. Nó seguindo outro:

- (a) Distância entre os ônibus: na típica situação em que um ônibus vem logo após o outro, normalmente cada um deve esperar pela saída e entrada de passageiros, o que pode interromper o seguimento. Contudo, ainda não deixa de apresentar a maior duração e regularidade nos contatos.
- (b) Itinerários dos ônibus em seguimento: se o itinerário for o mesmo, pode-se considerar praticamente como sendo um único nó. Se os itinerários são diferentes, apenas haverá seguimento em um trecho comum. Além disso, a demanda

diferente de entrada e saída de passageiros dos ônibus irá favorecer que o seguimento não aconteça por muito tempo;

4.Nó parado:

- (a) Espera na garagem: normalmente há um agrupamento dos ônibus e a comunicação é facilmente estabelecida.
- (b) Espera nos terminais: nem todos os ônibus estão no terminal ao mesmo tempo, contudo nos terminais os ônibus ficam parados esperando pelo cumprimento do horário dos seus itinerários para serem liberado para serviço. Há uma sincronização do itinerário executado com o estabelecido na tabela de horários dos ônibus. Os contatos com outros ônibus que estão chegando ou esperando pelo horário de suas partidas são favorecidos.
- (c) Espera no ponto de ônibus: quanto maior o tempo de permanência na parada, maior é o favorecimento por contatos devido à parada e também do aumento da sua duração. Um ônibus que historicamente tem maior tempo de espera para embarques e desembarques terá mais chances de realizar contatos, desde que os outros ônibus tenham esse tempo inferior.
- (d) Espera no trânsito: em congestionamentos os veículos se concentram e favorece um aumento das chances de comunicação.

As condições que influenciam esses fatores são as condições de tráfego, velocidade dos nós, falhas de comunicação, falhas nos nós e tempo de permanência nas paradas de ônibus. Devido à imprevisibilidade dessas condições e dificuldade da identificação prática da sua ocorrência, torna-se difícil estabelecer uma relação de probabilidade de ocorrência desses fatores e generalização para outros cenários. Diante disso, a utilização de métodos que favoreçam tirar proveito de relacionamentos que possam existir nos dados de contato entre os nós são interessantes para modelar a ocorrências dos contatos.

APÊNDICE C – Simulador de comunicação em DTN

C.1 INTRODUÇÃO

Para avaliar o desempenho das estratégias de encaminhamento de mensagens neste trabalho foi desenvolvido um simulador de comunicação em DTN que funciona a partir das informações de contato: (i) nó atual; (ii) nó contatado; (iii) momento absoluto no tempo do contato; e (iv) duração do contato. Trata-se de um simulador baseado em eventos, no qual as informações de contato devem ser fornecidas organizadas no tempo e de forma assimétrica. Entende-se como registro assimétrico de contatos quando há dois registros da comunicação entre dois nós, invertendo-se o nó atual e o nó contatado. A linguagem de programação utilizada foi Java.

O simulador deve ser parametrizado com as informações de: (i) número máximo de nós da rede; (ii) momento absoluto no tempo do início da simulação e (iii) do final da simulação em segundos; (iv) tamanho da fatia de tempo em segundos do algoritmo para reconstrução das jornadas; (v) tamanho máximo da matriz de probabilidades do MaxProp; (vi) tempo de vida máximo das mensagens em segundos; (iv) número máximo permitido de cópias de uma mensagem na rede, um representa somente a mensagem original sendo transmitida na rede; (v) tamanho dos buffers de mensagens em bytes.

C.2 IMPLEMENTAÇÃO DAS ESTRATÉGIAS DE ROTEAMENTO

Foram implementadas cinco estratégias de roteamento no simulador: a entrega direta, a epidemia sem limitações ou com limitações, o caminho de máxima probabilidade (MaxProp) e a estratégia da jornada predita baseada em RNA.

A entrega direta (*direct delivery*), também conhecida como espera para entregar (*wait to deliver*) é simplesmente uma estratégia que mantém a mensagem na fila de mensagens do nó até que aconteça contato com o nó de destino e se entregue a mensagem. Nessa estratégia não há cópia, nem transferência de mensagens, produzindo baixa entrega de mensagens, baixo consumo de recursos da rede, baixo consumo de filas de mensagens e baixo processamento. Os valores de entrega de mensagens e consumo de recursos são usados como referência em testes comparativos.

Para obter valores máximos de entrega de mensagens e de consumo de recursos para comparações, pode-se utilizar a estratégia epidêmica sem limitações de recursos. Essa estratégia implementa a inunda-

ção (*flooding*) sem controle de replicação e sem limitações de banda de rede e espaço na fila de mensagens. Na implementação desse simulador não é previsto um método para eliminação de mensagens da rede, tal como a vacina. Quando acontece um contato, as mensagens da fila são todas copiadas para o nó em contato, exceto aquelas que já se encontram no nó contatado.

A estratégia epidêmica com limitações também foi implementada, sendo que a entrega e a cópia de mensagens ficam sujeitas às limitações de banda de rede e duração dos contatos.

A implementação da estratégia do caminho de máxima probabilidade, MaxProp, foi realizada da mesma forma que implementada no simulador de redes oportunistas ONE, exceto pela estratégia de replicação que foi modificada. Para o modo replicador, no MaxProp foi adotada a limitação do número máximo de cópias da mensagem implementada no *Binary Spray and Wait* (BSW), tal como é definido para a estratégia proposta. Isso foi feito para tornar mais justa a comparação do atraso de entrega e do número de mensagens entregues, pois o BSW permite aumentar o número de réplicas no início do encaminhamento, privilegiando a melhoria dessas métricas.

C.3 CARACTERÍSTICAS IMPLEMENTADAS

C.3.1 Filas de mensagens

Foram implementadas duas filas de mensagens para cada nó da rede. Uma é a fila de mensagens entregues, ou seja, as mensagens que atingiram seus destinatários. E a outra é a fila de mensagens a serem entregues ou *buffer* de mensagens.

No sentido de privilegiar a entrega de mensagem com menos atraso possível, a fila de mensagens foi implementada para operar no modo *Last In First Out* (LIFO) para todas as estratégias. Assim, as mensagens criadas mais recentes são privilegiadas. Contudo, mensagens mais antigas que venham ser transferidas ou copiadas, acabam sendo colocadas no topo da fila de mensagens do nó que as recebem e assim acabam sendo privilegiadas. Vendo-se utilidade e simplicidade nessa estratégia, ela foi preferida ao invés de uma ordenação na fila de mensagens.

C.3.2 Tempo de vida da mensagem

O controle de tempo de vida (*Time To Live* - TTL) da mensagem foi implementado. Diferente do que acontece no TTL do protocolo da internet (IP), esse TTL registra o tempo de vida em segundos. Se a mensagem tiver TTL menor do que o tempo atual menos a tempo da sua criação ela é descartada.

Esse controle pode ser habilitado ou desabilitado no simulador.

C.3.3 Tempo de processamento

O tempo de processamento foi considerado para fins de controle da duração do tempo de contato. Isso significa que é considerado o tempo levado para o algoritmo realizar uma computação e descontado da duração do contato.

C.3.4 Tempo de comunicação

A determinação do tempo de comunicação depende da duração do contato que está sendo tratado e da quantidade de bit transferidos nesse contato.

A duração do contato é informada no arquivo de contatos ou é determinada através da duração média dos contatos a ser parametrizada no simulador.

A determinação da taxa de transferências é obtida pelo número de bits transferidos em relação a duração do contato. Quando a quantidade de bits ou taxa de transferência não é registrada, utiliza-se um parâmetro de taxa de transferência configurado no simulador que será usado por todos os nós. Usando esse parâmetro, não é estabelecido um controle de uso do canal, nem conexão com velocidades variadas. Assim, dependendo de como é configurada a taxa de transferência, pode-se considerar o uso da taxa cheia, tal como usando canais independentes para cada nó ou o uso de meia taxa, considerando que cada nó terá sempre disponível metade da banda de transmissão.

O tempo necessário para transferência da mensagem é determinado de acordo com a taxa de transferência e o tamanho da mensagem. Esse tempo é descontado da duração do contato para cada mensagem transferida. Se a mensagem levar mais tempo para ser transferida do que a duração restante do contato, a mensagem não é transmitida, tal

como na comunicação real.

A sobrecarga referente aos campos de controle dos protocolos de comunicação foi considerada de forma estatística sobre os dados a serem enviados através de um parâmetro informado na inicialização. As perdas na comunicação e ajustes nas taxas de transferência foram consideradas já representadas nos dados dos contatos fornecidos.

Como o objetivo principal foi avaliar as estratégias de forma igual mediante contatos informados que já apresentavam características de oportunismo, os detalhes de compartilhamento de acesso ao meio foram simplificados sem causar prejuízo para avaliação dos resultados.

C.3.5 Considerações sobre o processamento

Por causa do tratamento assíncrono dos contatos, a transferência de mensagens entre dois nós é primeiro realizada em um sentido e depois realizada em outro sentido. Assim, ao tratar o outro sentido, considera-se se a mensagem acabou de ser transferida pelo nó em contato não realizando seu processamento.

O tempo de processamento levado pela estratégia de roteamento para computar rotas logo após reconhecer o nó de contato é descontado na duração do contato, tal como na realidade.

C.4 VERIFICAÇÃO E VALIDAÇÃO DO SIMULADOR PARA OS EXPERIMENTOS REALIZADOS

No desenvolvimento de um modelo, pressupostos e simplificações sobre o comportamento do sistema real são estabelecidos, por isso torna-se necessário avaliar a qualidade das respostas (FREITAS, 2008).

Para avaliar um sistema de simulação, este deve ser verificado e validado para oferecer certas garantias de que irá fornecer resultados próximos aos originados no sistema real. A verificação diz respeito a consistir o modelo computacional com seus pressupostos e simplificações para avaliar se sua implementação está correta. Na validação deseja-se saber se o modelo implementado comporta-se semelhante ao sistema real, considerando seus pressupostos e simplificações.

As implementações dos protocolos Epidemic e Entrega Direta são simples. A implementação do MaxProp é mais complexa e para evitar problemas relativos a diferenças na programação da mesma estratégia, ela foi realizada como no simulador ONE.

Extensas depurações foram realizadas durante o desenvolvimento do simulador. Arquivos de depuração passo-a-passo foram gerados para avaliar o funcionamento das estratégias. Esses arquivos permitiram verificar as decisões de encaminhamento e avaliar o roteamento das mensagens através dos nós, permitindo corrigir erros e comportamentos inadequados.

A verificação foi sendo realizada à medida que o simulador foi sendo desenvolvido. Os testes consistiram em inicializar parâmetros com valores que devem gerar saídas determinadas e avaliar se essas saídas foram alcançadas. Por exemplo, se as mensagens tiverem um tamanho muito maior do que a maior duração observada nos contatos, não poderá haver transferência de mensagens entre os nós. De forma parecida, se for desabilitado o controle da duração dos contatos, nenhuma mensagem deixaria de ser transferida para outro nó, mesmo que com um tamanho muito grande.

Além disso, verificações foram realizadas atribuindo-lhe valores estáticos a determinadas variáveis e avaliando o comportamento de outras variáveis de acordo com valores já conhecidos. Casos simplificados com poucas mensagens e nós de rede também foram executados e avaliados.

Uma forma de realizar a validação é a comparação dos resultados de um sistema real com o sistema simulado nas mesmas condições de operação. Considerando-se as simplificações de pressupostos, dado os mesmos dados de entrada e valores dos parâmetros, os resultados do sistema simulado devem ser próximos ao real.

A dificuldade de encontrar resultados reais na entrega de mensagens, mensagens nas filas e outras informações da execução de um sistema em DTN torna difícil validar o simulado implementado. Contudo, é possível realizar comparações com resultados obtidos com outros simuladores já conhecidos e de referência para se obter algum nível de validação.

Neste sentido, os resultados de entrega de mensagem e mensagens nas filas de entrega após a execução do protocolo Epidemic foram comparadas com os obtidos no simulador ONE (KERÄNEN; OTT; KÄRKKÄINEN, 2009). Por causa das simplificações assumidas no simulador criado, uma pequena diferença, na margem de até 2% foi encontrada nos resultados entre os dois simuladores. Nas avaliações realizadas, verificou-se que a simplificação no controle de conexões simultâneas e a desconsideração do protocolo de comunicação para a troca de informações de roteamento estavam gerando essa diferença. Contudo, isso não estava causando prejuízo nas comparações dos resul-

tados entre estratégias de roteamento visto que todas essas estratégias estão sujeitas ao mesmo controle.

É importante saber que o simulador apenas realizou a simulação da entrega e transferências de mensagens, sendo que as informações da comunicação provem de dados reais. Assim, não existiram problemas referentes às questões de simulação da comunicação até a camada de transporte.