

# Um mecanismo baseado em sensibilidade à situação para otimizar a transmissão de dados de aplicações de sensoriamento urbano

Carlos O. Rolim<sup>1</sup>, Anubis G. Rossetto<sup>1</sup>, Valderi R. Q. Leithardt<sup>1</sup>  
Guilherme A. Borges<sup>1</sup>, Tatiana F. M. dos Santos<sup>2</sup>  
Adriano M. Souza<sup>2</sup>, Claudio F. R. Geyer<sup>1</sup>

<sup>1</sup>Institute of Informatics – Federal University of Rio Grande do Sul (UFRGS)  
91.501-970 – Porto Alegre – RS – Brazil

<sup>2</sup>Postgraduate Program in Production Engineering  
Federal University of Santa Maria (UFSM)  
Santa Maria, RS, Brazil

carlos.oberdan@inf.ufrgs.br, taty.nanda@gmail.com

**Abstract.** *One challenge in urban sensing applications is related to the transmission of data when network infrastructure is intermittent or unavailable. For such task, the Opportunistic Networks emerged as a promising alternative. The main problem in this context is how to improve messages routing between network nodes. Thus, this work presents a engine that employs situation awareness in conjunction with uncertainty treatment to overcome such limitation. The experiments demonstrate that the proposed engine was an acceptable performance and can be used as a basis for opportunistic data transmission in a urban sensing architecture called UrboSenti.*

## 1. INTRODUÇÃO

O conceito de Cidades Inteligentes foi utilizado pela primeira vez por [Gibson et al. 1992] para expressar como o desenvolvimento urbano estava passando a utilizar a tecnologia, inovação e globalização em assuntos cotidianos. Para autores como [Hernández-Muñoz et al. 2011], objetos, serviços e pessoas em uma Cidade Inteligente estão interconectados por uma infraestrutura de rede. Como resultado, pesquisadores estão procurando alternativas para proporcionar aos habitantes da cidade novos serviços e aplicações que melhorem a qualidade de vida e ao mesmo tempo atendam os critérios de eficiência energética e sustentabilidade.

Uma das alternativas mais promissoras e que já possui aplicações reais é o uso de pessoas, dispositivos móveis e sensores para a coleta de informações a respeito de diferentes aspectos da cidade em uma nova área da computação ubíqua chamada de Sensoriamento Urbano. Uma limitação para aplicações de sensoriamento urbano que utilizam protocolos Internet para transmissão está relacionada a como transmitir os dados de sensoriamento quando a infraestrutura de rede for intermitente ou indisponível (devido aos protocolos tradicionais de roteamento necessitarem de um caminho fim-a-fim entre a origem e o destino [Cao and Sun 2013]). As Redes Oportunistas surgem então como uma alternativa para suprir essas limitações [Boldrini et al. 2008]. Entretanto, no contexto de Redes Oportunistas, um problema que pesquisadores da área ainda buscam resolver está

relacionado a como otimizar o roteamento e a disseminação de mensagens (aumentando sua taxa de entrega e diminuindo sua latência e overhead) em cenários heterogêneos com alta dinamicidade, como no caso de Cidades Inteligentes.

Assim, o presente trabalho tem por objetivo apresentar um mecanismo que emprega sensibilidade à situação juntamente com tratamento de incertezas para otimizar o roteamento de mensagens em Redes Oportunistas. Parte-se da hipótese de que derivar os dados de contexto dos nodos de uma Rede Oportunista para uma visão de mais alto nível proporcionada pela sensibilidade à situação melhora as tomadas de decisões de roteamento. Como contribuições desse trabalho destaca-se: (i) é o primeiro trabalho que demonstra resultados de aplicação de sensibilidade à situação em Redes Oportunistas; (ii) apresentação da modelagem conceitual de um mecanismo de sensibilidade à situação capaz de lidar com incertezas. Além disso, os conceitos desse mecanismo serão aplicados no módulo de sensoriamento de uma arquitetura ubíqua orientada a serviços para o sensoriamento urbano chamada UrboSenti [Rolim et al. 2014]. Desde já ressalta-se que esse artigo apresenta uma evolução do trabalho apresentado anteriormente em [Rolim et al. 2015]. Assim, os maiores diferenciais em relação ao trabalho anterior estão relacionados aos modelos e técnicas empregadas para o tratamento de situações e o tratamento de incertezas.

O trabalho está organizado da seguinte forma: na próxima seção é apresentado o cenário problema que serve de motivação para o desenvolvimento desse trabalho; a seção 3 descreve a modelagem do cenário-problema e também a arquitetura do mecanismo de situação proposto; a seção 4 detalha as configurações empregadas nas simulações e também os resultados experimentais e, por fim, a seção 5 apresenta as conclusões e trabalhos futuros.

## 2. CENÁRIO PROBLEMA

Esse artigo faz o uso de um cenário-problema de uma Cidade Inteligente que utiliza nodos fixos e móveis para efetuar o sensoriamento urbano. Veículos e pessoas carregam seus dispositivos para obter dados do trânsito, da qualidade do ar, níveis de ruído, concentração de  $CO_2$ , etc. Em locais estratégicos da cidade existem Pontos de Acesso sem fios (Access Points – APs) que são responsáveis por receberem os dados de sensoriamento coletados e os encaminharem para serem processados por um sistema central. Além de dados de sensores, redes sociais são usadas para que seja possível compreender o comportamento da cidade e assim, através de uma visão holística, fornecer melhores serviços aos seus habitantes.

Todo o processo de coleta, processamento, análise e retorno de informações aos moradores da cidade é efetuado por uma arquitetura ubíqua orientada a serviços para o sensoriamento urbano chamada UrboSenti. Um dos módulos da UrboSenti é chamado de *Sensing module* e é responsável pela coleta de dados de sensoriamento de forma intencional ou não-intencional. Ele é executado em dispositivos móveis (como por exemplo, smartphones, veículos, etc) ou até mesmo em sensores fixos que são dispersos ao longo da cidade. Uma das características desse módulo é o suporte a diferentes métodos para transmissão dos dados coletados. É possível usar o padrão IEEE 802.11b/g/n (em modo estruturado ou ad hoc), GPRS/EDGE/3G ou mesmo rede cabeada como base para a comunicação utilizando TCP/UDP. Em locais onde a infraestrutura física não estiver

presente ou for intermitente, o *Sensing module* proporciona que a comunicação ocorra através de formas alternativas, não baseadas em caminhos fim-a-fim, como acontece na pilha TCP/IP.

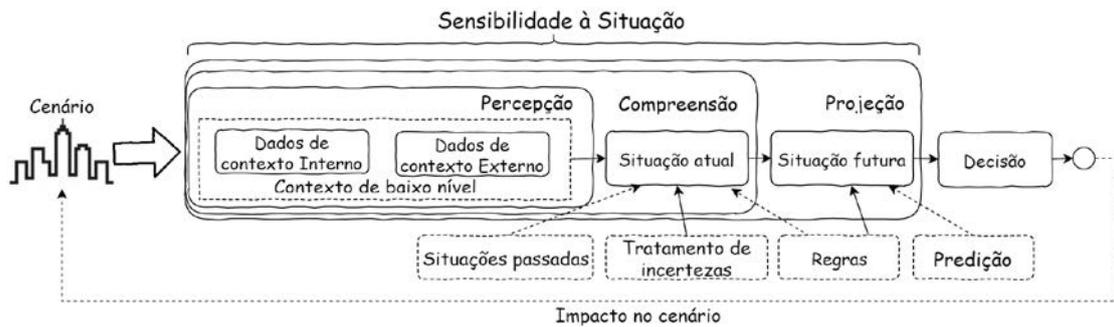
Essa forma de comunicação alternativa emprega o paradigma de Redes Oportunistas para permitir que os dados de sensoriamento sejam roteados entre os nodos até serem entregues aos APs que os enviarão para o processamento. As Redes Oportunistas (ou OppNets, do inglês *Opportunistic Networks*) são consideradas uma evolução das MANETs e um subcategoria de Redes Tolerantes a Atrasos e Desconexões. As OppNets herdaram todas as características das DTNs, porém elas exploram a mobilidade e contatos dos nodos como forma de proporcionar a entrega de dados. Dessa forma, mobilidade, desconexões, atrasos, particionamentos e demais aspectos considerados desafios em redes móveis são vistos como características das Redes Oportunistas. Pelo fato de não necessitar da existência de um caminho direto entre a origem e o destino, é empregado o conceito de armazena-carrega-encaminha (*Store-Carry and Forward*) para troca de mensagens. A cada contato entre os nodos é decidido quais mensagens devem sair do buffer e serem encaminhadas ao nodo encontrado e quais devem permanecer. Se o nodo encontrado recebeu mensagens, ele irá armazenar localmente e em um novo contato com outro nodo deverá decidir o que deve ser encaminhado ou permanecer consigo. Esse processo se repete em todos os encontros entre nodos da rede até que a mensagem seja entregue ao seu destino.

O desafio existente nessa forma de comunicação está relacionado à estratégia de seleção de boas “mulas de dados” (do inglês *data mule* – um termo utilizado em Redes Oportunistas para designar nodos com boa capacidade de carregar uma mensagem ao seu destino ou o mais próximo deste). Diferentes propostas foram feitas por pesquisadores para superar tal desafio. Como exemplos, podem ser citados: Epidemic, Prophet, BubbleRap, MobySpace, AFRON, CAR, HiBOp, Propicman, DRAFT entre outros. Não está no escopo do presente artigo analisar a cada uma das propostas (maiores informações podem ser obtidas em [Boldrini et al. 2008]). Entretanto, acredita-se que essas propostas existentes não capazes de lidar de forma adequada com os requisitos do cenário-problema apresentado nesse trabalho. Assim, busca-se verificar a validade da hipótese de que a aplicação de sensibilidade à situação é uma forma promissora de superar o desafio de escolha de boas mulas dados em cenários de sensoriamento urbano.

### 3. SOLUÇÃO PROPOSTA

#### 3.1. Modelagem do cenário

A modelagem conceitual do cenário-problema apresentado anteriormente é representada na Figura 1 e assim descrita: cada nodo do cenário possui dados de contexto chamados de **contexto de baixo nível**. Ele é definido pelo conjunto de dados de contexto interno e de contexto externo. O **contexto interno** envolve os dados que caracterizam o nodo em um determinado instante corrente ou passado. Como exemplos de atributos de contexto interno podem ser considerados: nome, posição, velocidade, energia da bateria, nodos próximos, distância percorrida total, distância percorrida desde o último encontro, capacidade de processamento, número de mensagens no buffer, espaço disponível no buffer para armazenamento, tráfego na rede e também outros atributos suportados pelo dispositivo. O contexto interno também é formado por valores que provavelmente irão caracterizar o



**Figura 1. Modelo do cenário-problema empregando sensibilidade à situação**

nodo em um momento futuro. Esses são chamados de **valores preditos** e são gerados a partir da aplicação de uma técnica de predição.

Além do contexto interno, cada nodo é capaz de obter informações dos nodos vizinhos e também do ambiente que o circunda. Esses dados são chamados de **contexto externo**. Como contexto externo são considerados os atributos dos nodos vizinhos como: velocidade, capacidade de processamento, espaço em buffer, energia disponível, etc. O ambiente possui atributos que podem ser coletados pelos sensores disponíveis no nodo, como por exemplo: ruído, luminosidade, qualidade do ar e também outros atributos que possam ser coletados pelos dispositivos.

O contexto de baixo nível do nodo é utilizado para auxiliar na tomada de decisão de encaminhamento de mensagens e também nas adaptações das funcionalidades internas do componente. Porém, conforme apontado por [Ye et al. 2011], utilizar partes isoladas de dados de contexto pode não apresentar significado útil para a tomada de decisões. Assim, com a finalidade de agrupar os dados de contexto de baixo nível e então derivar uma nova visão de alto-nível que facilite a tomada de decisão é empregado o conceito de *sensibilidade à situação* que originalmente foi proposto por [Endsley 1995].

Os 3 níveis desse modelo de sensibilidade à situação foram adaptados para o mecanismo proposto da seguinte forma: **Nível 1 - Percepção:** é considerado o nível mais baixo da sensibilidade à situação. Aqui, ele tem por finalidade perceber o status e valores dos atributos dos elementos e interações do contexto de baixo nível e então agrupa-los para serem utilizados pelas camadas superiores. É importante notar que alterações nos valores do contexto de baixo nível poderão caracterizar uma situação atual diferente da anterior, tornando assim o sistema dinâmico; **Nível 2 - Compreensão:** utiliza os dados de contexto de baixo nível advindos do Nível 1 e deriva a visão de alto nível chamada de situação. Em outras palavras, a principal função do nível 2 é de fornecer a situação dos nodos. Um repositório de *situações passadas* é usado juntamente com um conjunto de *regras* para caracterizar situações. Um fato importante que deve ser considerado é que os dados advindos do Nível 1 nem sempre serão dados precisos. Dessa forma, o Nível 2 deve ser capaz de lidar com a imprecisão dos dados, preferencialmente, através de uma abordagem baseada em Inteligência Computacional. **Nível 3 - Projeção:** tem por finalidade projetar ações futuras dos nodos. Os valores de contexto interno preditos são utilizados pelo nível 2 para caracterizar uma provável situação futura. O conjunto de situações passadas, corrente e futura são utilizados por este nível para estimar uma ação que deverá

ser executada em um futuro próximo. O acerto ou erro na estimativa da projeção futura influencia na decisão tomada. O impacto disso no cenário é mensurado para melhorar as novas projeções que serão efetuadas.

É importante ressaltar que devido às restrições de energia e processamento dos nodos existentes no ambiente, as atividades dos níveis 2 e 3 devem ser desenvolvidas com custo computacional que não comprometa a precisão das ações e não deteriore a bateria dos nodos.

### 3.2. Arquitetura do mecanismo

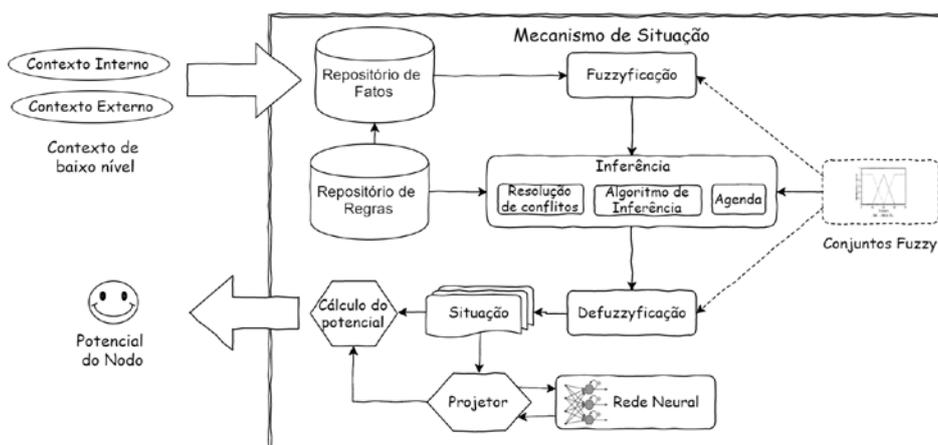


Figura 2. Arquitetura do Mecanismo de Situação

A arquitetura do Mecanismo de Situação é apresentada na Figura 2 e assim descrita: Os dados de *Contexto de baixo nível* (contexto interno e contexto externo) são fornecidos como entrada para o mecanismo. Esses dados, denotados por  $C_n$  onde  $n = \{power, buffer, distance\_travelled, node\_type, \#of\_messages, \dots\}$ , são estruturados de forma adequada e então armazenados no *Repositório de fatos*. Cada conjunto de dados de Contexto de baixo nível é considerado um fato. Dessa forma, no *Repositório de fatos* existem fatos atuais e também fatos históricos que são utilizados para proporcionar a percepção da situação (nível 1 do modelo de Endsley). Um fato passa então pelo processo de *fuzzyficação* onde seus valores são convertidos para valores fuzzy de acordo com funções (membership functions) definidas no *Conjunto Fuzzy*. Os valores convertidos representam o grau de pertinência daquele valor a determinado grupo (como por exemplo, carga da bateria cheia, pela metade, ou vazia). Tais valores fuzzy servem de entrada para o processo de *Inferência*. Nesse processo, são aplicadas regras armazenadas no *Repositório de Regras* juntamente com um algoritmo de inferência baseado em encadeamento progressivo ou regressivo (forward chaining ou backward chaining) para determinar quais regras são aplicadas aos dados fuzzy. Se forem verificadas regras conflitantes, é aplicado a *Resolução de conflitos* que determina quais regras devem ser disparadas. A *Agenda* então permite priorizar, dentre as regras quem combinam com os dados (considerado um match), a ordem de disparo de cada uma. A saída do processo de inferência é então *Defuzzyficada* e são obtidos valores  $V_{s_n}$  que caracterizam a situação de cada uma das variáveis de Contexto de baixo nível (nível 2 do modelo de Endsley). Os valores  $V_{s_n}$  são encaminhados para o *Projctor* que fornece os  $V_{s_n}$  atuais e anteriores (histórico) para

uma *Rede Neural*. Como saída são gerados valores  $P_{s_n}$  que representam a predição da caracterização de situações futuras (nível 3 do modelo de Endsley). Além de serem utilizados para a projeção de situação, os valores  $V_{s_n}$  são empregados para o *Cálculo do potencial* do nodo. Nesse processo é atribuído um peso  $W_n$  para cada  $C_n$  de acordo com sua importância para o nodo ser considerado uma boa “mula de dados”. Então é feita a normalização desses pesos utilizando uma função na forma  $W'_n = f(x) = w_{ni} / \sum_{i=1}^n w_{ni}$ . Uma vez que os pesos estão normalizados, é calculado o *Potencial do Nodo* com a equação  $P_{node} = V_{s_{n1}} * W'_{n1} + V_{s_{n2}} * W'_{n2} + \dots + V_{s_{nN}} * W'_{nN}$ . Todo esse processo é executado ciclicamente em períodos determinados de tempo.

Quando um nodo  $n_j$  encontra um nodo  $n_k$  é usado  $P_{node}$  para decidir se as mensagens que estão no buffer local devem ser encaminhadas para o nodo encontrado ou devem permanecer no buffer local. Para essa decisão  $n_j$  aplicada a equação:

$$forward = \begin{cases} true, & \text{if } P_{node}^{n_j} > P_{node}^{n_k} \\ false, & \text{otherwise} \end{cases}$$

Ressalta-se que ainda não foi incorporada à equação de definição do potencial do nodo  $P_{node}$  as situações preditas  $P_{s_n}$ . Tal aspecto encontra-se em análise e será publicado em artigos futuros.

## 4. SIMULAÇÕES E RESULTADOS EXPERIMENTAIS

### 4.1. Configurações dos cenários

Para verificar a funcionalidade e a performance do Mecanismo de Situação foram executadas simulações utilizando o ONE (Opportunistic Network Environment) Simulator [Keränen et al. 2009]. Sua escolha deve-se a ser de código aberto, extensível e amplamente utilizado por pesquisadores da área de Redes Oportunistas. As simulações foram executadas durante um período de 6 horas em um cenário da cidade de Porto Alegre/RS. A cidade teve as suas principais avenidas mapeadas e convertidas para o formato interno aceito pelo simulador. Foram então criados 2 grupos de nodos: pessoas e veículos.

Em cada uma das simulações foi adotada uma quantidade crescente de nodos (10 nodos de cada grupo na primeira simulação, 25 de cada grupo a segunda, 50, 75, 100, 500 e 1000 nodos de cada grupo na última simulação. Ambos os grupos se movimentavam no mapa utilizando o modelo de mobilidade *Shortest Path Map Based Movement* (nesse modelo, os nodos utilizam caminhos aleatórios disponíveis no mapa e utilizam o algoritmo de Dijkstra para encontrar o caminho mais curto entre os nodos e Pontos de Interesse). Além disso, aos dois grupos foram aplicadas as seguintes restrições de energia: todos os nodos iniciaram com uma carga completa de bateria de 19080 Joules (que corresponde a uma bateria comumente encontrada em celulares de 5.3Wh com 3.7V) e uma recarga sendo efetuada aleatoriamente em intervalos entre 4 horas e 4 horas e meia. A energia gasta para a busca de outros nodos foi definida em 0.092 mW/s e o gasto para enviar e receber mensagens definido em 0.08 mW/s. O grupo formado por pessoas se movimentavam com velocidade entre 0.5 e 1.5 Km/h e tinham um celular equipado com uma interface Bluetooth com alcance de 5 metros e velocidade de transmissão de 2Mbps e uma interface WiFi com alcance de 20 metros e 10Mbps de velocidade de transmissão. O grupo de carros se movimentava com velocidades entre 10-50 km/h e possuía uma interface WiFi com as mesmas características do grupo de pessoas. Tanto o grupo de carros

quanto o de pessoas possuía um buffer para armazenamento de mensagens com 50M de tamanho.

Em locais estratégicos da cidade foram distribuídos 7 APs responsáveis por receber os dados de sensoriamento coletados pelos nodos móveis e então os encaminhá-los via rede cabeada para serem processados por um sistema centralizado. Para isso, eles foram equipados com uma interface WiFi com alcance de 30 metros e 10Mbps de velocidade de transmissão e uma interface cabeada com 1Gb de velocidade. Por serem equipamentos dedicados, esses APs estavam conectados permanentemente na energia, não precisando de recarga de elétrica, e possuíam o buffer para armazenamento de mensagens com 512Mb de tamanho. Cada nodo do cenário foi equipado com uma aplicação responsável por simular o sensoriamento de eventos ao longo da cidade. Os eventos de sensoriamento ocorreram em intervalos de 100 a 200 segundos, sendo que as mensagens foram geradas com tamanho variando segundo uma distribuição uniforme de 100k a 2M. Cada mensagem foi definida com um TTL de 300 segundos (5 horas). Esse é o tempo máximo para que uma mensagem seja roteada no cenário e então entregue ao primeiro APs que o nodo que carrega tal mensagem tiver contato.

#### **4.2. Estratégia de implementação**

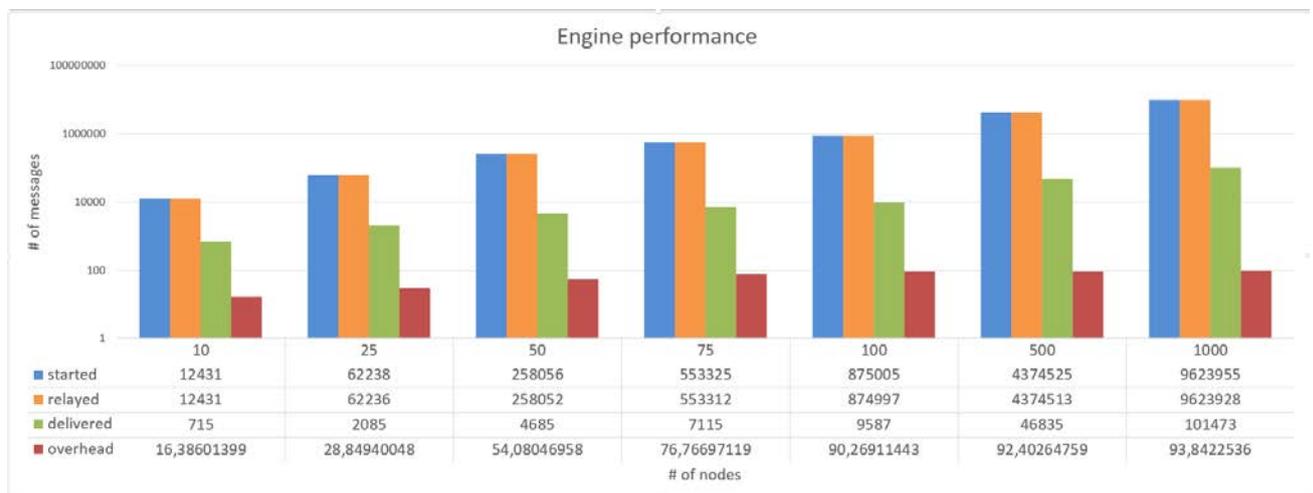
Para implementar o Mecanismo de Situação foi utilizado como base o JBoss Drools. O Drools é uma de sistema de inferência híbrido baseado em regras. Seu motor de inferência permite a construção de regras que serão avaliadas através de algoritmos baseados na combinação de encadeamento progressivo e regressivo, como RETE networks [Forgy 1982] (no Drools chamado de ReteOO para indicar suporte à orientação a objetos) e também uma evolução desse algoritmo chamado de PHREAK.

Para definição de regras, o Drools utiliza uma linguagem própria chamada DRL (Drools Rule Language). Um arquivo DRL, basicamente, contém um conjunto de regras do tipo “when LHS then RHS” onde LHS (Left Hand Side) ou antecedente corresponde aos elementos condicionais da regra e RHS (Right Hand Side), também chamado consequente, corresponde as ações que serão executadas em caso de disparo da regra. Tanto o LHS e o RHS podem ser formados por conjuntos de zero, um ou mais itens. Como o Drools nativamente somente é capaz de utilizar regras sem tratamento de incertezas, foi utilizado uma extensão do sistema chamada de Drools Chance [Sottara et al. 2008, Wulff and Sottara 2009]. O Chance permite que as regras sejam definidas e avaliadas utilizando Lógica Fuzzy. Dessa forma, a partir das definições de funções de pertinência e da aplicação dos algoritmos de inferência Fuzzy consegue-se efetuar o tratamento de incertezas conforme requerido pelo modelo de situação que está sendo proposto e foi apresentado na seção 3.1. O Chance segue a mesma lógica de declaração de variáveis e regras de um arquivo DRL normal. Entretanto, deve-se declarar a forma como os valores serão manipulados pela Lógica Fuzzy. Para isso, as variáveis linguísticas são definidas através de funções de pertinência e então podem ser utilizadas nas regras. No mecanismo de situação, cada uma dos atributos de contexto de baixo nível teve uma variável linguística vinculada com três funções triangulares com valores variando em escala de “pouco”, “médio” e “alto” de acordo com sua magnitude. Essas variáveis linguísticas foram empregadas em regras que fornecem como saída a situação de cada atributo de contexto de baixo nível. Por questão de espaço, não serão fornecidos exemplos de sintaxe das regras utilizadas.

Conforme apresentado na seção 3.2, uma vez definida a situação de cada atributo de contexto de baixo nível é calculado o potencial do nodo. Como parte da equação de cálculo do potencial, foi definida uma tabela contendo os pesos para cada um dos dados de contexto de baixo nível. Lembrando que esses pesos são estipulados de acordo com a importância do dado para o nodo ser considerado uma boa mula de dados. Foi definido então a energia da bateria e o tipo de nodo (se estático ou móvel) com peso 1.0, o espaço em buffer com peso 0.8, distância percorrida com peso 0.4 e a quantidade total de mensagens que o nodo já carregou com peso 0.2. Esses pesos então foram normalizados e aplicados na equação que como saída indicou o potencial momentâneo de cada um dos nodos e que define se uma mensagem deve permanecer no buffer local ou ser encaminhada ao nodo encontrado. Ressalta-se que o mecanismo de situação proposto no momento utiliza somente situações atuais, sem empregar situações projetadas (preditas). Dessa forma, não foi explicado nesse trabalho o funcionamento da Rede Neural que será utilizada (maiores detalhes da Rede podem ser encontrados em [Rolim et al. 2015]).

### 4.3. Resultados experimentais

Para verificar a performance do mecanismo proposto, foram realizados experimentos em diferentes cenários utilizando as configurações de simulação detalhadas acima. Os resultados são apresentados na Figura 3.



**Figura 3. Performance do mecanismo proposto**

Os resultados demonstraram que no geral há um incremento no número de mensagens entregues (delivered) conforme o aumento de nodos. Comparando os cenário de 10 nodos com o cenário de 50 nodos (crescimento de 5 vezes o número de nodos), ocorreu um aumento de mais de 500% na quantidade de mensagens entregues, porém o overhead foi de apenas 29%. No comparativo de 100 nodos com 500 nodos o aumento do overhead foi de apenas 2% e o aumento da quantidade de mensagens entregues foi de quase 400%. Ou seja, cada vez que o número de nodos duplica praticamente duplicam as mensagens entregues. O overhead não acompanha essa taxa de crescimento, mantendo-se baixo. Isso demonstra que o uso do potencial do nodo baseado na situação definida está sendo efetiva nas tomadas de decisões de encaminhamento.

Um fato que chamou a atenção nas simulações foi a quantidade de mensagens

criadas (started) e encaminhadas (relayed) entre os nodos. Isso ocorre devido ao aumento de contatos a medida que são inseridos mais nodos no cenário. Pode-se notar que algumas vezes uma mensagem saiu do buffer de um nodo e algum tempo depois, após outros contatos, acabou retornando ao mesmo nodo. Ao mesmo tempo que isso aumenta a probabilidade de entrega de uma mensagem ao destino, também aumenta o overhead para entrega da mensagem. Uma solução é fazer com que uma mensagem não possa retornar a um nodo pelo qual ela já passou. Nos testes isso reduziu em torno de 30% a quantidade de mensagens encaminhadas e também o overhead, mas também diminuiu a quantidade de entregas em torno de 20%. Essa solução precisa ser melhor analisada.

Outro aspecto interessante que pode-se observar nas simulações e que não está representado nos gráficos é o impacto do TTL das mensagens. Foram efetuados testes com TTL de 60 minutos e com 300 minutos (5 horas). Com poucos nodos do cenário a quantidade de mensagens expiradas foi maior com um TTL de 60 minutos. Isso também era esperado devido ao pouco contato entre nodos. Porém pode-se notar que a partir de 50 nodos no cenário o TTL de 60 minutos gerou um impacto menor no overhead e manteve praticamente a mesma taxa de mensagens entregues. Ou seja, pode-se pensar que o uso de um TTL dinâmico, definido de acordo com alguma variável de contexto (como por exemplo quantidade de contatos do nodo) pode ser uma boa alternativa. Essa opção também será analisada melhor no futuro.

Além disso, foram realizados comparativos de performance com algumas das principais iniciativas de Redes Oportunistas. Essas iniciativas foram escolhidas devido a representarem diferentes “classes de protocolos” estudadas por pesquisadores da área [Boldrini et al. 2008]. Foram efetuadas simulações com o Epidemic, Prophet, Bubble-Rap e Draft (devido a questões de espaço não serão fornecidas maiores informações das configurações usadas e o detalhamento dos resultados). Os experimentos demonstraram que o mecanismo proposto teve a quantidade de mensagens entregues apenas inferior ao Epidemic, porém com overhead menor em todos os cenários. Com isso, pode-se concluir que o uso de sensibilidade à situação está sendo uma boa estratégia para guiar as decisões de roteamento.

## 5. Conclusões e trabalhos futuros

Esse artigo apresentou um mecanismo que utiliza sensibilidade à situação para otimizar o roteamento de mensagens em Redes Oportunistas aplicadas ao sensoriamento urbano. Além do uso de sensibilidade à situação, um dos diferenciais desse mecanismo é o suporte ao tratamento de incertezas para definição da situação dos nodos. Os resultados experimentais demonstraram que a caracterização de situação proporciona melhorias na tomada de decisões de roteamento, validando assim a hipótese inicial.

O mecanismo proposto já implementa os níveis 1 e 2 do modelo de situação. Porém ainda não está fazendo projeções de situações futuras (nível 3). O suporte a essa característica está em fase de desenvolvimento com o uso de Redes Neurais e acredita-se que os resultados poderão ser melhorados quando a codificação estiver finalizada.

Como trabalhos futuros, pode-se apontar a finalização do suporte a projeção de situação, o desenvolvimento de novas simulações para verificação da performance do componente e também estudos do impacto da recusa de um nodo receber mensagens que já passou por ele anteriormente e o uso de TTL definidos dinamicamente de acordo

com alguma variável de contexto do nodo. Por fim, espera-se que com a conclusão do mecanismo o mesmo possa ser incorporado ao módulo Sensing da UrboSenti para então iniciarem pequenos testes de prova de conceito em ambientes reais.

## Referências

- Boldrini, C., Boudec, J. Y. L., Chaintreau, A., and Conti, M. (2008). Deliverable 2.2: Final specification of forwarding paradigms in Hagggle. Technical report.
- Cao, Y. and Sun, Z. (2013). Routing in Delay/Disruption Tolerant Networks: A Taxonomy, Survey and Challenges. *IEEE Communications Surveys & Tutorials*, 15(2):654–677.
- Endsley, M. R. (1995). Toward a Theory of Situation Awareness in Dynamic Systems.
- Forgy, C. L. (1982). Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, 19(1):17–37.
- Gibson, D. V., Kozmetsky, G., and Smilor, R. W., editors (1992). *Technopolis Phenomenon: Smart Cities, Fast Systems, Global Networks*. Rowman & Littlefield Publishers, Inc.
- Hernández-Muñoz, J. M., Vercher, J. B., Muñoz, L., Galache, J. A., Presser, M., Hernández Gómez, L. A., and Pettersson, J. (2011). Smart cities at the forefront of the future internet. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6656:447–462.
- Keränen, A., Ott, J., and Kärkkäinen, T. (2009). The ONE Simulator for DTN Protocol Evaluation. In *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, New York, NY, USA. ICST.
- Rolim, C. O., Rossetto, A. G., Leithardt, V. R., Borges, G. A., dos Santos, T. F., Souza, A. M., and Geyer, C. F. (2014). Towards Predictive Routing Agents in Opportunistic Networks. In *Fifth International Workshop on Collaborative Agents Research & Development, CARE for Intelligent Mobile Services - Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Paris, France. Springer-Verlag Berlin Heidelberg.
- Rolim, C. O., Rossetto, A. G., and Leithardt, V. R. Q. (2015). Putting Opportunistic , Situational and Smart Approaches to Underlie the Data Transmission of Social Urban Sensing Applications. In *7o Simpósio Brasileiro de Computação Ubíqua e Pervasiva, Anais do XXXV Congresso da Sociedade Brasileira de Computação*, pages 191–200, Recife, PE.
- Sottara, D., Mello, P., and Proctor, M. (2008). Adding uncertainty to a Rete-OO inference engine. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5321 LNCS:104–118.
- Wulff, N. and Sottara, D. (2009). Fuzzy reasoning with a rete-oo rule engine. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5858 LNCS:337–344.
- Ye, J., Dobson, S., and McKeever, S. (2011). A review of situation identification techniques in pervasive computing. *Pervasive and Mobile Computing*, In Press,(0).