Um mecanismo para garantia de QoS na Internet das Coisas com RFID

Rafael Perazzo Barbosa Mota[1], **Daniel Macêdo Batista**[1]

[1]Departamento de Ciência da Computação – Universidade de São Paulo (USP)
Rua do Matão 1010 – 05508-090 – São Paulo – SP – Brasil

{perazzo,batista}@ime.usp.br

Abstract. *Despite advances in the development of new mechanisms for the Internet of Things (IoT), there are few studies aimed at ensuring QoS requirements and evaluated in scenarios with a large number of nodes. Mechanisms proposed to ensure IoT QoS requirements would improve the performance in environments sensitive to packet loss, and in scenarios of tracking and localization, especially in terms of scalability and network overhead. In this paper we present an algorithm that reduces the amount of messages in IoT scenarios. Simulations confirm the effectiveness of the algorithm. For example, in one scenario, the packet loss rate which was between 10% and 42% was reduced to less than 3%.*

Resumo. *Apesar dos avanços no desenvolvimento de novos mecanismos para a Internet das Coisas (IoT), existem poucos trabalhos voltados para a garantia de requisitos de QoS e que sejam avaliados em cenários com muitos nós. A proposta de mecanismos para garantir requisitos de QoS na IoT melhoraria o desempenho de ambientes sensíveis à perda de pacotes, e em cenários de rastreamento e localização, principalmente em termos de escalabilidade e sobrecarga da rede. Neste artigo é apresentado um mecanismo para garantia de QoS que reduz a quantidade de mensagens na rede em cenários da IoT. Experimentos de simulação confirmam a eficácia do algoritmo. Por exemplo, em um cenário, a taxa de perda de pacotes que era de 10% a 42% foi reduzida para menos de 3%.*

1. Introdução

Atualmente, a maior parte das interações na Internet é realizada entre seres humanos [Miorandi et al. 2012]. No entanto, em um futuro próximo, qualquer "coisa" (*thing*) poderá ser endereçada na grande rede. A Internet, então, tornar-se-á a Internet das coisas (*Internet of things - IoT*). As comunicações serão concebidas não apenas entre humanos mas também entre humanos e coisas e entre coisas sem a interação com seres humanos. Este novo paradigma vem rapidamente adquirindo espaço principalmente no moderno cenário das comunicações sem fio. Em resumo, a Internet das Coisas consiste na presença difusa de uma variedade de coisas ou objetos ao nosso redor, como, por exemplo etiquetas RFID - *Radio Frequency IDentification* (identificação por radiofrequência), telefones celulares inteligentes, redes de sensores sem fio - RSSF, entre outros, que se comunicam a fim de trocar muitas mensagens, além das poucas trocadas por simples sensores [Atzori et al. 2010].

Segundo previsão do NIC (*US National Intelligence Council*), até o ano de 2025, os nós da Internet poderão estar em todas as coisas e permitirão inúmeras

oportunidades para o desenvolvimento tanto econômico como tecnológico mundial [Evdokimov et al. 2011]. Nesta "nova" internet haverá um sem-número de objetos heterogêneos [Liu e Zhou 2012]. Dessa forma, conforme [Miorandi et al. 2012, Nef et al. 2012, Atzori et al. 2010], os diferentes tipos de objetos envolvidos tornam a IoT um paradigma diferente das atuais RSSF. Enquanto os protocolos e os nós em uma RSSF são voltados para cenários geralmente específicos para observação de fenômenos ambientais [Nef et al. 2012], na IoT espera-se expandir este cenário permitindo também aplicações onde os objetos possuam alguma conectividade sem necessariamente precisar lidar com fenômenos ambientais. Neste artigo são analisados cenários que oferecem serviços de localização e rastreamento de objetos, ou seja, cenários caracterizados pela sensibilidade à alta perda de pacotes.

O objetivo deste artigo é propor um mecanismo de QoS para cenários de IoT cujos nós sejam etiquetas RFID e que sejam sensíveis à perda de pacotes. O mecanismo proposto teve seu desempenho avaliado por meio de experimentos simulados no simulador ns-2[1], e os resultados confirmam a eficácia do mecanismo. Por exemplo, em um cenário, a perda de pacotes que era de 10% a 42% foi reduzida para menos de 3% quando o mecanismo foi utilizado. A escolha da tecnologia RFID deve-se ao fato da mesma ser uma das tecnologias chave da IoT assim como a mais adequada para aplicações de rastreamento e localização, devido sua própria característica implícita de identificação.

As contribuições deste artigo são:

- Proposta de um mecanismo para garantia de QoS em cenários IoT sensíveis a perda de pacotes e sua análise de desempenho.
- Desenvolvimento de uma extensão para simular cenários com etiquetas e leitores RFID no ns-2;

Este artigo difere-se dos encontrados na literatura porque realiza experimentos simulados de cenários IoT reais, com o ns-2, e quantidade de etiquetas variáveis, o que possibilita uma análise mais aprofundada do mecanismo de QoS proposto.

O artigo está organizado da seguinte forma: a Seção 2 aborda a contextualização da tecnologia RFID. A Seção 3 descreve os trabalhos relacionados. A Seção 4 detalha o mecanismo de QoS proposto, enquanto os cenários IoT propostos estão explicados na Seção 5. A análise de desempenho, a extensão RFID para ns-2 e os resultados obtidos nas simulações dos cenários, são expostos e discutidos na Seção 6. Finalmente, na Seção 7 são apresentados os trabalhos futuros e as conclusões da pesquisa.

2. Contextualização

A tecnologia RFID refere-se a um sistema de identificação sem fio que permite a comunicação entre etiquetas e leitores, por intermédio de ondas de radiofrequência. A principal vantagem da RFID é que os objetos identificados não precisam estar muito próximos dos leitores, ocasionando uma fácil automação do processo de leitura. A quantidade de etiquetas depende do contexto da aplicação. Elas podem ser empregadas em diversas situações como controle de acesso, ingressos eletrônicos, rastreamento animal, localização e rastreamento, passaportes, entre outras [Finkenzeller et al. 2010].

[1]http://www.isi.edu/nsnam/ns/

De acordo com [Finkenzeller et al. 2010], um sistema de RFID é formado por três componentes básicos:

- *Transponders* ou etiquetas RFID: localizadas nos objetos a serem identificados, armazenando o código de identificação;
- *Transceivers* ou leitores RFID: responsável pelas leituras/escritas nas etiquetas e
- *Middleware* ou aplicação: responsável pelo processamento da informação obtida pelo leitor.

Em geral, sistemas RFID passivos são baseados no fato de que "o leitor fala primeiro", conforme definido no padrão em [EPCglobal 2008]. Assim, o leitor tem a função de enviar requisições para etiquetas ao seu alcance que respondem com seus respectivos identificadores. Estes são recebidos pelo leitor que envia para a aplicação, onde os dados serão processados. Pode-se perceber que a aplicação exerce papel fundamental, pois esta utilizará os dados da forma que lhe for conveniente e pode criar, com uma única tecnologia, um ambiente de Internet das coisas. A título de exemplo, imaginemos um cenário em que diferentes tipos de objetos são etiquetados com etiquetas RFID, leitores são posicionados em pontos estratégicos e existe uma aplicação que em tempo real consegue disponibilizar a localização dos objetos etiquetados para um público específico. A partir deste ponto é possível visualizar uma gama de possibilidades de aplicações e cenários de IoT, adotando-se apenas a tecnologia RFID [Miorandi et al. 2012]. Em resumo, o funcionamento do sistema segue os passos a seguir: (i) O leitor faz a solicitação que é enviada a partir de sua antena de rádio usando um canal denominado de leitor-etiqueta; (ii) A etiqueta recebe a requisição também através de sua antena, prepara um pacote de resposta com seu identificador e (iii) envia de volta pelo canal etiqueta-leitor ; (iv) O leitor recebe o identificador e (v) repassa para a aplicação. Este procedimento constitui o protocolo básico de comunicação da tecnologia RFID.

É importante observar que o paradigma da IoT está fortemente relacionado à efetiva integração RSSF e RFID, pois enquanto no sistema de RFID, geralmente, os nós são passivos e respondem apenas quando solicitados pelos leitores; nas RSSFs, os nós são, na maioria, autônomos e podem, por iniciativa própria, estabelecer uma comunicação em um ambiente IoT [Miorandi et al. 2012]. Este aspecto confirma a expressiva importância dessas duas tecnologias ao tratarmos de Internet das Coisas. Entretanto, o foco deste artigo será na melhoria de cenários de RFID. A integração entre RFID e RSSF é um dos trabalhos futuros conforme apresentado na Seção 7.

3. Trabalhos relacionados

Em [Welbourne et al. 2009] é relatado que na Universidade de Washington foram utilizadas etiquetas RFID para identificar pessoas (voluntárias) e objetos para implementar a Internet das Coisas, em um projeto chamado de "*RFID Ecosystem*", que tem o objetivo de oferecer os serviços de localização e rastreamento das coisas. Vários leitores foram posicionados para abranger os sete andares dos oito mil metros quadrados de um prédio do campus. O projeto contou com quarenta e quatro leitores, cada um equipado com quatro antenas, distribuídos no local. Os leitores enviavam os dados coletados das etiquetas para um servidor central. O mecanismo de funcionamento do RFID Ecosystem baseia-se em consultas periódicas do *software* aos leitores, que buscam a detecção de novas etiquetas e geram um evento denominado TRE - *Tag-reader event*, ou seja, um evento

leitura-etiqueta por antena, por segundo. A partir dos experimentos, os resultados obtidos foram os dados relativos ao número de requisições/respostas realizadas, erros de leitura e confiabilidade das informações fornecidas pelo leitor às aplicações. Por conseguinte [Welbourne et al. 2009] sinalizaram a importância do desenvolvimento de mecanismos que lidem com a sobrecarga da rede, já que este foi um problema identificado, apesar de terem sido utilizadas apenas 324 etiquetas com 44 leitores distribuídos por toda a área de cobertura. A partir deste fato pode-se observar a fragilidade do modelo experimentado, pois foram utilizados muitos leitores, poucas etiquetas e mesmo assim muitas mensagens sobrecarregaram a rede, já que as consultas eram realizadas a cada segundo, gerando na maioria das vezes pacotes e comunicações desnecessárias.

Com relação à Qualidade de Serviço, [Nef et al. 2012] sinaliza a necessidade de se definir tipos de serviços na IoT, para que seja possível a determinação de parâmetros de QoS. O artigo utiliza as RSSF como uma das tecnologias chave, e a partir de suas características genéricas, compara os variados protocolos de acesso ao meio *(MAC - Media access control)* para expor os mecanismos já existentes para prover qualidade dos serviços. No entanto, uma RSSF, como parte de uma IoT, pode necessitar de parâmetros diferentes, já que outros tipos de dispositivos podem também estar inseridos na rede. Finalmente, a partir de uma classificação proposta para as mais diversas aplicações de IoT, propõe-se uma topologia IoT que melhor adequa-se à utilização de RSSF integradas nos cenários. Para cada tipo diferente de aplicação, são descritos os modelos de serviços das mesmas, para que seja possível a identificação e proposta dos parâmetros de QoS.

Segundo [Duan et al. 2011], cenários IoT, orientados a aplicação, consistem de uma integração de várias tecnologias, conforme já sinalizam outros autores, sendo necessária assim uma criteriosa investigação sobre arquiteturas de QoS, para possibilitar na prática, a oferta de serviços com qualidade. Posteriormente, requisitos de QoS são resumidos através da análise das características de aplicações para controle, consultas, monitoramento em tempo real e monitoramento genérico. Cada modelo de aplicação pode requerer parâmetros diferentes de QoS. Dessa forma os autores propõem a organização da IoT em três camadas: Aplicação, Rede e Percepção. Todas com capacidade de prover QoS. A arquitetura proposta visa ajudar pesquisadores da área, para tentar facilitar na busca de solução de problemas relacionados a QoS na IoT.

Diversas propostas de novos mecanismos anti-colisão são discutidos e analisados em [Wu et al. 2013, Han et al. 2012, Leonardo e Victor 2012, Zhong et al. 2012]. No entanto os objetivos das mesmas consiste em diminuir o tempo de singularização em uma única leitura, não sendo consideradas aplicações para a Internet das Coisas. Dessa forma, o mecanismo proposto neste artigo pode ser implementado em quaisquer protocolos previamente propostos na literatura.

Em termos de simulação de cenários de funcionamento de RFID, existem algumas soluções propostas na literatura. Podemos citar o *Rifidi*[2], ns-2[3] , ns-3[4] e OMNet++ [Pal 2012]. O *Rifidi* é um emulador específico para a tecnologia RFID que permite a montagem de cenários com leitores e etiquetas, em diversos tipos de funcionamento. Apesar de utilizado na literatura, o mesmo não dispõe de recursos necessários para a investigação

[2]http://www.transcends.co/

[3]http://www.isi.edu/nsnam/ns/

[4]http://www.nsnam.org/

de análise de desempenho, pois se trata de um *software* direcionado apenas para testes de cenários antes da compra dos equipamentos. O ns-3 é a nova geração do consagrado predecessor ns-2. Lançado em julho de 2008, foi completamente remodelado e reescrito em linguagem C++. Embora possua uma série de melhorias em relação ao anterior, ainda é discretamente aproveitado pela comunidade acadêmica relacionada às redes de computadores, além disso, tem limitada documentação e poucos novos módulos disponíveis. O OMNet++, assim como o ns-2 e ns-3, é também um simulador baseado em eventos discretos, tendo como finalidade preencher as lacunas deixadas entre aplicações livres como o ns e programas comerciais de alto custo. Finalmente, o popular ns-2 continua a ser o simulador mais utilizado, inclusive em publicações atuais, para as diferentes situações, amplamente aceito como ferramenta para validação e testes de arquiteturas, aplicações, tecnologias, protocolos, entre outros. Além disso, o mesmo possui implementações de tecnologias sem fio, como WiMax e RSSF, que poderão possibilitar trabalhos futuros para análises de cenários heterogêneos, característica esta presente na IoT. Por este motivo o ns-2 foi o escolhido para a realização dos experimentos deste trabalho. A fim de facilitar a realização das simulações, e também para auxiliar a comunidade que realiza pesquisas em RFID, nós implementamos e disponibilizamos uma extensão para o ns-2 suportar experimentos com leitores e etiquetas RFID. Nem o ns-3 nem o OMNet++ possuem módulos prontos que permitam simulações de cenários com dispositivos RFID.

4. Mecanismo proposto

Conforme [Welbourne et al. 2009], o excesso de pacotes trocados entre leitores e etiquetas RFID afeta significativamente a escalabilidade da rede e também as garantias de requisitos de QoS das aplicações. O nosso mecanismo visa reduzir essa quantidade de pacotes e pode ser aplicado a quaisquer mecanismos anti-colisão propostos na literatura.

O mecanismo proposto tem como objetivo garantir a qualidade dos serviços oferecidos em cenários de localização e rastreamento, reduzindo as taxas de perdas de pacotes. Uma característica comum pode ser observada nestas situações: As etiquetas não precisam responder a todas requisições do leitor. O fato é justificado pois as etiquetas apenas precisam responder requisições vindas de leitores diferentes da requisição anterior, ou seja, o nó não precisa informar a localização, se a mesma não foi alterada. O mecanismo proposto está detalhado no Algoritmo 1. A Figura 1 ilustra a máquina de estados das etiquetas. A implementa

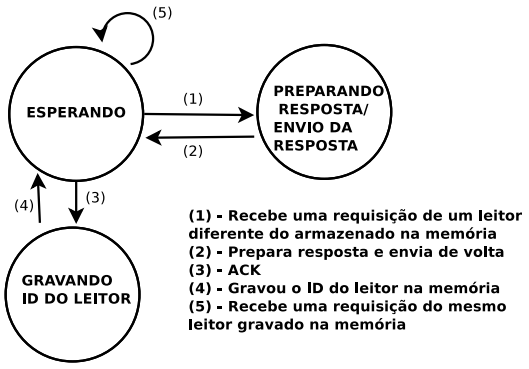

Figura 1. Máquina de estados das etiquetas para o mecanismo proposto

Algoritmo 1: Mecanismo implementado nas etiquetas para garantia de QoS em aplicações da IoT para rastreamento e localização

Entrada: Pacote de solicitação do leitor
Saída: Pacote contendo o ID da etiqueta, que é enviado após um tempo aleatório, conforme o Algoritmo 2 (Subseção 6.1), ou $\emptyset$

```
1  A cada novo pacote de requisição recebido pela etiqueta:
2  se (Primeira requisição recebida) então
3  |  Enviar o pacote de resposta com seu identificador
4  |  se (Receber pacote com ACK do leitor) então
5  |  |  Entrar em modo silencioso até receber uma requisição de um leitor
   |  |  diferente
6  |  fim
7  senão
8  |  se (A origem da requisição for a mesma já armazenada na etiqueta)
   |  então
9  |  |  Não responde a requisição
10 |  senão
11 |  |  Enviar pacote de resposta
12 |  |  se (Receber pacote com ACK do leitor) então
13 |  |  |  Gravar o identificador do leitor
14 |  |  |  Entrar em modo silencioso até receber uma requisição de um
   |  |  |  leitor diferente
15 |  |  fim
16 |  fim
17 fim
```

Observa-se que a etiqueta possui três estados: (i) "Esperando": A etiqueta está esperando uma requisição ou recebeu uma requisição do mesmo leitor que possui armazenado em sua memória. Caso a etiqueta receba uma requisição de um leitor diferente a mesma passa para o estado de (ii)"preparação e envio de resposta", e volta ao estado esperando. Caso o leitor receba a resposta corretamente, o mesmo envia um ACK de confirmação para etiqueta, que no estado "esperando" passa para o estado (iii) "gravando ID do leitor" onde é armazenado o ID do novo leitor, e posteriormente volta-se ao estado esperando. O Algoritmo 1, para implementação nas etiquetas realiza exatamente os passos descritos pelo diagrama de estados da Figura 1, ou seja, a etiqueta verifica a origem da requisição (linhas 2 e 7) e responde apenas se a origem for diferente do ID já armazenado na memória (linhas 3 e 11) . Ao responder, a etiqueta grava a nova ID do leitor apenas ao receber um ACK de confirmação (linhas 12 e 13). Como pode-se observar, a saída do Algoritmo 1 depende da execução do Algoritmo 2 que define o instante em que o pacote com o ID da etiqueta, caso seja gerado, será enviado. O Algoritmo 2 será apresentado na Subseção 6.1.

5. Cenários e experimentos

A fim de avaliar o desempenho do mecanismo proposto na Seção 4, dois cenários de IoT com dispositivos RFID foram simulados. A lista abaixo descreve os dois cenários.

- i) Um cenário que simula uma sala de aula, com turmas de 30 a 430 alunos variando de 50 em 50 que: Permanecem na sala durante a primeira aula (cinquenta minutos); Saem para o intervalo (20 minutos); Voltam para a segunda aula (cinquenta minutos); Saem do prédio onde se localiza a sala de aula. Foram utilizados três leitores, um na sala de aula, outro no pátio do intervalo e o último na saída do prédio. O cenário está ilustrado na Figura 2a;
- ii) Um cenário com cinco leitores, representando parte de uma feira de exposições, e quantidade de nós entre 50 e 1050, variando de 100 em 100, com posições e movimentos aleatórios. A Figura 2b ilustra o cenário.

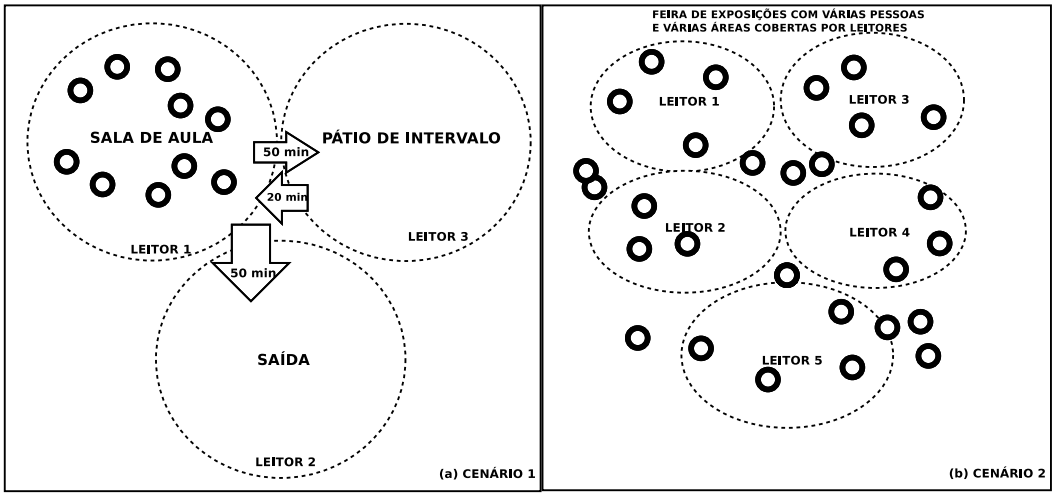

Figura 2. Cenários modelados

Os seguintes parâmetros foram levados em consideração:

- O tempo do experimento para o cenário (i) foi de cento e dez minutos, com leituras enviadas a cada três segundos. Para o cenário (ii) simulou-se trinta minutos, com solicitações a cada cinco segundos.
- O mecanismo RTS/CTS - *Request to Send / Clear to Send* - (Requisição para enviar / Livre para enviar) e o protocolo de roteamento foram desabilitados por não fazerem parte do padrão de comunicação RFID [Finkenzeller et al. 2010]. Estes parâmetros foram ajustados a partir do protocolo de acesso ao meio (*Medium Access Control - MAC* 802.11), simulando assim o MAC do sistema RFID.
- A potência de transmissão - *Power transmission - Pt* foi diminuída a fim de alcançar valores próximos dos reais, de um a dez metros, com os parâmetros *Pt_ 0.28 (mW) e RXThresh_ (potência mínima para que os pacotes sejam recebidos pelos receptores) 2.12249e-07 (W)* [Chen et al. 2007], calculados a partir de uma a ferramenta que encontra-se em `ns-2.35/indep-utils/propagation/threshold.cc` na árvore de diretório descompactada do ns-2 versão 2.35. Necessitou-se modificar os parâmetros de potência, pois o protocolo 802.11 possui a potência do sinal muito maior do que um leitor ou etiqueta RFID, possibilitando desta forma a simulação correta da potência do sinal.
- As taxas de transmissão dos canais leitor-etiqueta e etiqueta-leitor foram definidas como 9Kbps e 128Kbps [Chen et al. 2007], respeitando valores reais.

Cada cenário foi executado vinte vezes para cada quantidade de nós. O percentual de perda de pacotes para cada simulação foi calculado através da média aritmética entre as vinte medições, conforme a Equação 1. A Equação 2 mostra a fórmula de cálculo do tráfego gerado (quantidade de Kbytes transferidos) pelas respostas das etiquetas, no sentido etiqueta-leitor. As simulações foram realizadas em um servidor equipado com processador Intel Core i7-2700K 3.5Ghz, 16GB de memória RAM e 1TB de espaço em disco rodando o sistema operacional Debian GNU/Linux versão 6.0.

$$Tp = \frac{\sum_{i=1}^{20}(\frac{\sum_{j=1}^{ql}(\frac{\sum_{k=1}^{s}(\frac{d_k}{d_k + r_k})}{s})}{ql})}{20} \quad (1)$$

$$Qt = \frac{\sum_{i=1}^{20}(\frac{(d_i + r_i) * 8}{1000})}{20} \quad (2)$$

onde:

- Tp = Percentual de taxa de perda de pacotes (0-1);
- Qt = Quantidade, em *KBytes*, de tráfego gerado pelas etiquetas aos leitores;
- d_i = Quantidade de pacotes descartados simulação *i*;
- r_i = Quantidade de pacotes recebidos na simulação *i*;
- i = Número da simulação;
- ql = Quantidade de leitores;
- s = Quantidade de requisições;
- d_k = Quantidade de pacotes descartados na requisição *k*;
- r_k= Quantidade de pacotes recebidos na requisição *k* ;
- 8 - Tamanho em *bytes* do pacote RFID.

6. Análise de desempenho

A análise de desempenho dos cenários com e sem a implementação do mecanismo proposto foi realizada através de simulações com o *ns-2*, seguindo a Equação 1 e a Equação 2 para a obtenção dos valores que serão mostrados em gráficos nesta seção.

6.1. Extensão RFID para ns-2

O *software* escolhido, ns-2, não possui pacotes ou módulos próprios ou de terceiros que modelem os componentes de um sistema de RFID, como etiquetas e leitores e seu protocolo de comunicação. Na literatura também não existem publicações que ofereçam tal

funcionalidade. Por isso foi implementado uma extensão para o ns-2 que modela o funcionamento de componentes RFID. Os parâmetros da camada de enlace foram configurados a partir do padrão 802.11, ajustando-os para adequarem-se aos valores reais do padrão RFID.

A extensão desenvolvida considera as seguintes características existentes em sistemas RFID:

- As etiquetas não iniciam uma comunicação. Apenas respondem a uma requisição de um leitor, informando seu identificador.
- Os leitores enviam pacotes de difusão, solicitando que todas as etiquetas em seu alcance retornem seus identificadores. Estes, quando recebidos, são enviados a uma aplicação central;
- A aplicação central realiza todo processamento relacionado aos identificadores das etiquetas, e suas respectivas localizações a partir da informação recebida pelo leitor;
- O mecanismo anti-colisão das etiquetas é baseado em um algoritmo probabilístico (Algoritmo 2) em que a etiqueta espera um tempo aleatório entre zero e dois segundos, antes de responder ao leitor, minimizando, mas não eliminando, a possibilidade de colisão de pacotes na chegada ao leitor. Inseri esta frase: Este tipo de solução foi escolhido para generalizar as soluções probabilísticas propostas na comunidade científica.

Algoritmo 2: Algoritmo anti-colisão probabilístico implementado nas etiquetas

Entrada: Pacote de solicitação do leitor
Saída: Envio do pacote contendo o identificador da etiqueta
1 Desempacota a requisição recebida do leitor;
2 Sorteia número aleatório entre 0 e 2 segundos (t), baseado em uma distribuição uniforme;
3 Prepara o pacote de resposta;
4 Agenda o envio do pacote para *t* segundos.

A extensão[5] pode ser usada para modelar diversos cenários, com vários leitores e várias etiquetas espalhadas por toda uma área predefinida. Além disso, pode-se fazer com que as etiquetas movimentem-se, como se estivessem representando uma pessoa em movimento, ou um objeto pertencente a uma pessoa em movimento. Características como as apresentadas na seção 5 devem ser configuradas no arquivo *tcl* de modelagem do cenário, pois tratam-se de características da camada de acesso ao meio. Já os tipos de etiquetas suportadas seguem o padrão definido em [EPCglobal 2008].

As seguintes modificações foram realizadas no ns-2, versão 2.35:

- Criação dos agentes *RfidReader* (arquivos rfidReader.cc e rfidReader.h) e *RfidTag* (arquivos rfidTag.cc e rfidTag.h);

[5]A extensão desenvolvida está melhor descrita e validada no sítio eletrônico `http://www.ime.usp.br/~perazzo/rfid_module.php`

- Criação do tipo de pacote *RfidPacket* (rfidPacket.cc e rfidPacket.h), com os campos EPC (identificador da etiqueta) e ID (identificador do leitor que fez a solicitação);
- Inclusão do novo pacote *RfidPacket* no arquivo *common/packet.h*;
- Inclusão dos agentes *RfidReader* e *RfidTag* no arquivo *tcl/lib/ns-default.tcl* que é responsável pela disponibilização do acesso aos cabeçalhos do novo pacote;
- Alteração do arquivo *Makefile.in*, para inclusão dos agentes *rfidReader.o, rfidTag.o e rfidPacket.o*;

A Figura 3 ilustra o diagrama de classes criadas, com seus atributos e métodos. A classe *RfidReader* possui os atributos *id_* que representa o identificador do leitor, *tagEPC_* que armazena o código EPC recebido de uma etiqueta e *singularization_* que configura se o leitor requer ou não algum mecanismo anti-colisão, como o Algoritmo 2. A entidade etiqueta possui os atributos *id_* que armazena o identificador do leitor que enviou a requisição e *tagEPC_* que representa o próprio identificador da etiqueta. Já um pacote RFID é formado pelos seguintes campos:

- id_: Identificador do leitor que enviou a requisição;
- tagEPC_: Identificador da etiqueta que enviou a resposta;
- tipo_: Direção do fluxo: 0 para etiqueta-leitor e 1 para leitor-etiqueta;
- singularization_: 0 para nenhum mecanismo e 1 para mecanismo probabilístico;
- service_: 0 sem mecanismo de QoS e 1 com mecanismo de QoS.

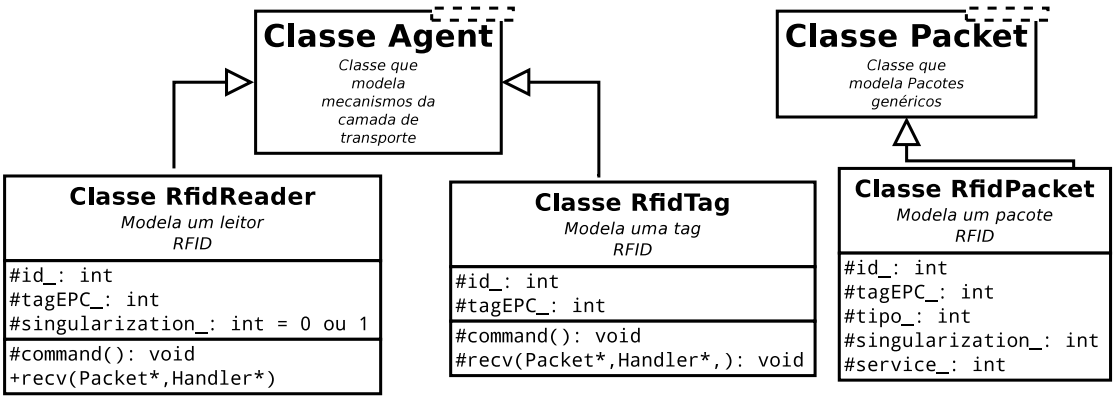

Figura 3. Diagrama de classes criadas

6.2. Resultados e discussão

As Figuras 4 e 5 apresentam gráficos que plotam o percentual de perda de pacotes em função da quantidade de nós para os cenários modelados. Duas curvas são apresentadas em cada gráfico. Uma delas, identificada como "Cenário 1 sem mecanismo" apresenta os resultados sem a implementação do mecanismo proposto no Algoritmo 1. A outra curva, identificada como "Cenário 1 com mecanismo" apresenta os resultados com a implementação do mecanismo proposto no Algoritmo 1. O Algoritmo 2 foi utilizado para ambas as curvas, ou seja, as etiquetas sempre esperam um tempo aleatório antes de responder, a fim de reduzir o número de colisões.

Observa-se no Cenário 1 (Figura 4) que o percentual de perdas de pacotes permanece constante quando a turma varia entre 30 e 180 alunos, mantendo uma taxa de perdas abaixo de 5%. Este fato é justificado pela característica do leitor que consegue lidar bem

com estas quantidades de nós. Percebe-se também que após os 230 alunos, a taxa de perda cresce significativamente, caracterizando que o leitor está no limite de sua capacidade de leituras, gerando perdas que podem inviabilizar a implantação do cenário. Ainda na Figura 4 pode-se observar que o mecanismo de QoS proposto levou os índices de perdas para um valor constante baixo, independente da quantidade de etiquetas, viabilizando a implementação do cenário sem que haja a necessidade de aquisição de novos leitores. Comparando o percentual de perdas no cenário com 430 nós observa-se que houve uma diminuição de aproximadamente 95% quando o mecanismo de QoS foi implementado.

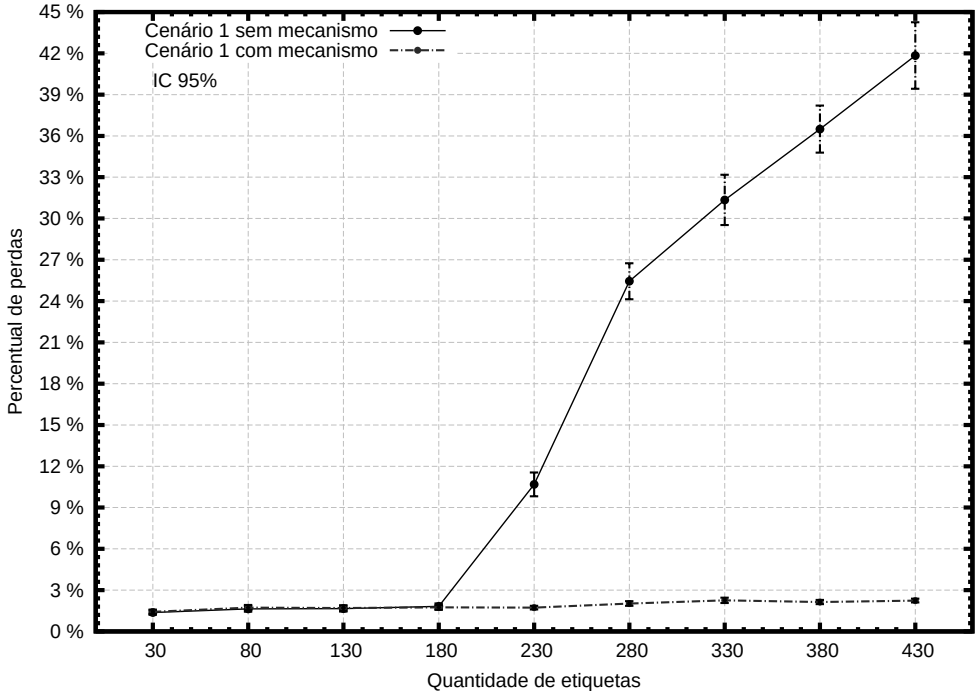

Figura 4. Percentual de perdas x Quantidade de etiquetas para o Cenário 1

O Cenário 2, que simula uma feira de exposições, onde diversas pessoas movimentam-se aleatoriamente por vários locais diferentes, revela pequenos percentuais de perdas até os 450 nós, conforme Figura 5. Quando o mecanismo proposto não foi utilizado, as perdas chegaram a quase 18% quando haviam até 1050 etiquetas. O mecanismo proposto mostrou-se bastante eficiente, pois assim como no Cenário 1, a curva permanece praticamente com valor constante pequeno. Comparando os resultados quando haviam 1050 etiquetas observa-se que a utilização do mecanismo levou a uma redução de 81% na porcentagem de perdas de pacotes.

Para cada um dos cenários, percebe-se que o a utilização do algoritmo anti-colisão não resolve o problema de falta de escalabilidade dos ambientes, ou seja, se aumentarmos muito a quantidade de nós, a taxa de perdas elevar-se-á significativamente. Este fato é resolvido com o mecanismo de QoS proposto.

As Figuras 6 e 7 apresentam a quantidade de Kbytes transferidos na rede nos Cenários 1 e 2 respectivamente. Como era de se esperar, a implementação do mecanismo de QoS reduz significativamente a quantidade de bytes na rede, o que trouxe como consequência a redução na taxa de perdas. Por exemplo, no Cenário 1, com 430 nós, a

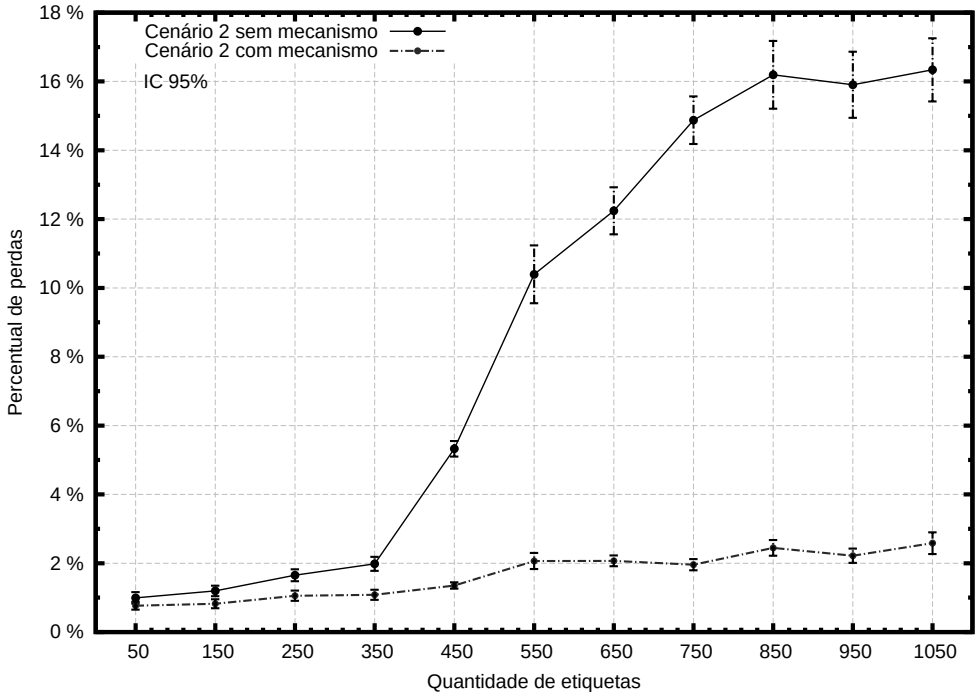

Figura 5. Percentual de perdas x Quantidade de etiquetas para o Cenário 2

quantidade de bytes foi reduzida em 84%. No cenário 2, a redução com a quantidade de 1050 etiquetas, foi de 63%.

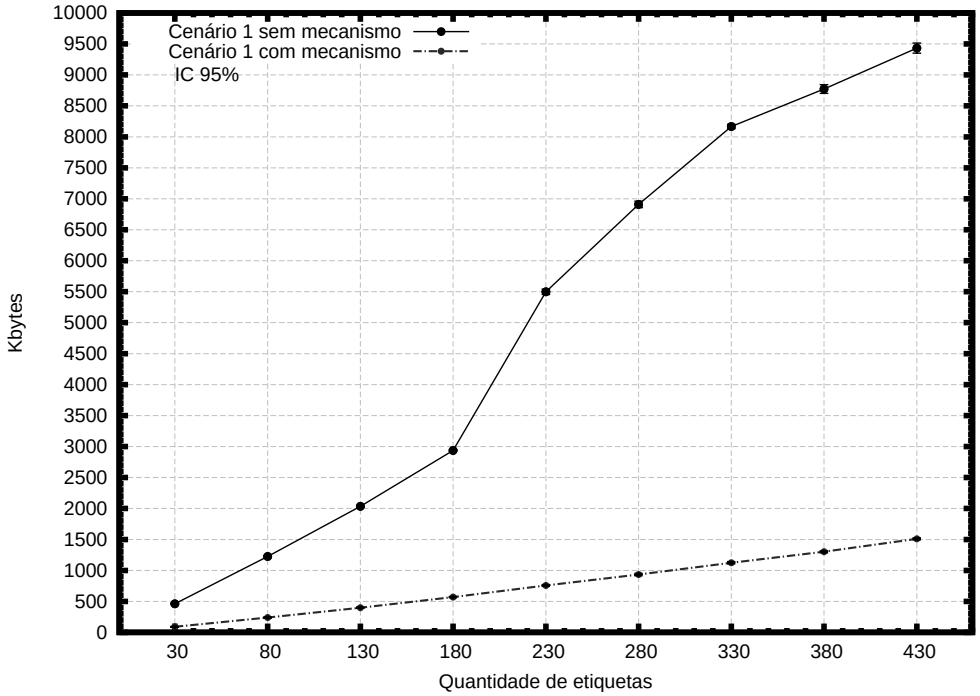

Figura 6. Quantidade média de Kbytes transferidos no Cenário 1

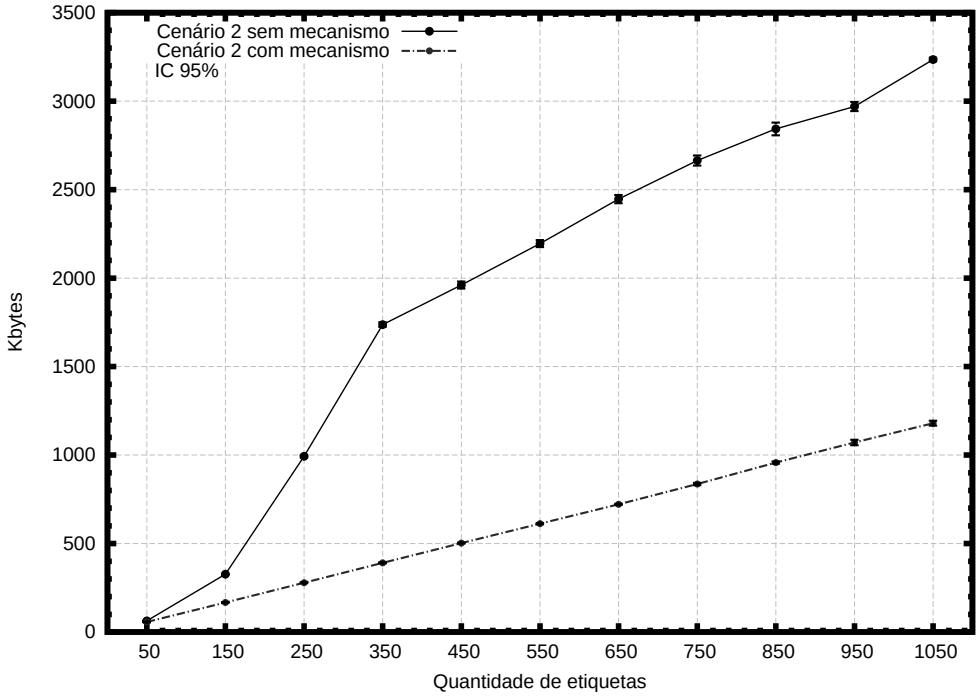

Figura 7. Quantidade média de Kbytes transferidos no Cenário 2

7. Conclusões

Este artigo analisou o desempenho de cenários de IoT com RFID, propôs e validou um mecanismo de QoS para cenários sensíveis à perda de pacotes e apresentou uma extensão à aplicação *ns* que modela a tecnologia RFID para utilização na IoT. Os resultados mostraram que o mecanismo implementado para a IoT consegue diminuir a taxa de perdas, assim como a quantidade de Kbytes transferidos na rede.

Dada a crescente utilização de RFID na prática, como por exemplo a utilização em passaportes e uniformes escolares[6] no Brasil, pode-se observar a importância da pesquisa realizada, visto que em um futuro não tão distante, as aplicações IoT, cada vez mais, farão parte da rotina das pessoas.

Como trabalhos futuros propomos ampliar o escopo da extensão desenvolvida, para que seja possível a realização de novos experimentos para simulação de novos mecanismos, assim como a integração com outras tecnologias.

Referências

Atzori, L., Iera, A., e Morabito, G. (2010). The Internet of Things: A Survey. *Computer Networks*, 54(15):2787–2805.

Chen, Q., Schmidt-Eisenlohr, F., Jiang, D., Torrent-Moreno, M., Delgrossi, L., e Hartenstein, H. (2007). Overhaul of IEEE 802.11 Modeling and Simulation in ns-2. In *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, MSWiM '07, páginas 159–168.

[6]http://www.dpf.gov.br/servicos/passaporte/passaporte-eletronico/
http://www.congressorfid.com.br/entrevista-edilson/

Duan, R., Chen, X., e Xing, T. (2011). A QoS Architecture for IOT. In *Internet of Things (iThings/CPSCom), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing*, páginas 717 –720.

EPCglobal, I. (2008). EPCTM Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860MHz-960MHz Version 1.2.0.

Evdokimov, S., Fabian, B., Günther, O., Ivantysynova, L., e Ziekow, H. (2011). *RFID and the Internet of Things: Technology, Applications, and Security Challenges*. Now Publishers Inc.

Finkenzeller, K. et al. (2010). *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-field Communication*. Wiley.

Han, H., Park, J., e Lee, T.-J. (2012). RFID Anti-Collision Protocol for Monitoring System of Tags in Motion. In *Ubiquitous and Future Networks (ICUFN), 2012 Fourth International Conference on*, páginas 318–321.

Leonardo, D. e Victor, M. (2012). Adding Randomness to the EPC Class1 Gen2 Standard for RFID Networks. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*, páginas 609–614.

Liu, Y. e Zhou, G. (2012). Key Technologies and Applications of Internet of Things. In *Intelligent Computation Technology and Automation (ICICTA), 2012 Fifth International Conference on*, páginas 197 –200.

Miorandi, D., Sicari, S., Pellegrini, F. D., e Chlamtac, I. (2012). Internet of Things: Vision, Applications and Research Challenges. *Ad Hoc Networks*, 10(7):1497 – 1516.

Nef, M., Perlepes, L., Karagiorgou, S., Stamoulis, G., e Kikiras, P. (2012). Enabling QoS in the Internet of Things. In *CTRQ 2012, The Fifth International Conference on Communication Theory, Reliability, and Quality of Service*, páginas 33–38.

Pal, D. (2012). A Comparative Analysis of Modern Day Network Simulators. In Wyld, D. C., Zizka, J., e Nagamalai, D., editors, *Advances in Computer Science, Engineering & Applications*, volume 167, páginas 489–498. Springer Berlin / Heidelberg.

Welbourne, E., Battle, L., Cole, G., Gould, K., Rector, K., Raymer, S., Balazinska, M., e Borriello, G. (2009). Building the Internet of Things Using RFID: The RFID Ecosystem Experience. *Internet Computing, IEEE*, 13(3):48 –55.

Wu, H., Zeng, Y., Feng, J., e Gu, Y. (2013). Binary Tree Slotted ALOHA for Passive RFID Tag Anticollision. *IEEE Transactions on Parallel and Distributed Systems*, 24(1):19–31.

Zhong, W., Chen, J., Wu, L., e Pan, M. (2012). The Application of ALOHA Algorithm to Anticollision of RFID Tags. In *Measurement, Information and Control (MIC), 2012 International Conference on*, volume 2, páginas 717–720.