

Persistência de Dados em Redes Móveis Ad-hoc Centradas em Interesses

Michael D. B. Silva, Heberte F. Moraes and Claudio L. Amorim
Laboratório de Computação Paralela e Sistemas Móveis - COPPE
Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil
michaeldouglas@poli.ufrj.br, heberte@ufrj.br, amorim@cos.ufrj.br

ABSTRACT

Redes móveis ad-hoc centradas em interesses, ou Radnets, propõem uma nova classe de MANETs (em Inglês, *Mobile Ad Hoc Network*) onde a comunicação é orientada ao conteúdo (interesses). Este tipo de comunicação vem em contraposição ao modelo Cliente/Servidor utilizado nas atuais propostas de MANETs, tal modelo herdada da arquitetura IP dominante na Internet. Nas Radnets, o modelo natural de comunicação das aplicações distribuídas é o Publicador/Subscriber, com suporte eficiente no nível de rede. A Radnet está em desenvolvimento no laboratório COMPASSO da COPPE/UFRJ e está implementada em Linux, disponível em versões para desktops, smartphones (Android +2.1) e Raspberry PI (Debian). Um problema importante em MANETs é o de persistência de dados devido aos períodos de inatividade das aplicações. Neste trabalho, propomos e avaliamos um sistema baseado em *message-oriented middleware* como uma solução de persistência em Radnets a nível de aplicação.

1. INTRODUÇÃO

Protocolos para redes móveis ad hoc ou MANETs (*Mobile Ad Hoc Network*) têm sido objeto de pesquisa há décadas e ainda representam um desafio para os pesquisadores da área. Em uma MANET cada nó opera de forma distribuída independente de roteadores ou pontos centrais de acesso, com pouca ou nenhuma infraestrutura. Na MANET, a rede é criada no momento em que um nó necessita trocar informações. A comunicação, muitas das vezes, ocorre através de múltiplos saltos de forma que cada nó atua como um roteador em potencial ao invés da necessidade de infraestrutura [6].

A maioria dos protocolos para MANETs fazem uso do endereçamento IP que é focado em comunicação direta entre duas máquinas. Recentemente, uma nova abordagem centrada em conteúdo vem sendo investigada, esta nova abordagem foi denominada Radnets (Rede centrada em Interesses) que são uma implementação distribuída do modelo Publicador/Subscriber (Pub/Sub) com o foco em MANETs. O

Pub/Sub baseia-se na interação de assinantes que definem seus interesses em um determinado evento, e de publicadores que são os geradores de eventos [4].

A Radnet faz uso do protocolo REPA (protocolo Endereçado por Prefixo Ativo) cuja comunicação é feita usando em estrutura chamada de Prefixo Ativo composto por dois conjuntos de campos denominados de Prefixo e Interesses, descrito em mais detalhes no decorrer do texto.

A Radnet atende parcialmente os níveis de desacoplamento propostos pelo modelo Publicador/Subscriber (Pub/Sub) [4]. O desacoplamento temporal é uma das bases do modelo no qual a Radnet ainda não implementa. Uma das soluções para prover este tipo de desacoplamento é através da persistência de dados que é importante nos períodos onde as aplicações nos nós não estão ativas, quando isto acontece informações são perdidas. Uma vez provido a persistência as aplicações serão capazes de receber informações independente de seu estado, ativo ou não.

O Message-oriented middleware (MOM) é uma infraestrutura de software que proporciona baixo acoplamento na comunicação em componentes de software distribuídos e trocas de mensagens assíncronas [5]. A solução de desacoplamento temporal proposta neste trabalho é um sistema MOM no qual as mensagens de interesse da aplicação são armazenadas em um base de dados para acesso posterior. Desenvolvemos um serviço para gerenciamento da base de dados chamado RepasMS e uma aplicação de exemplo para troca de SMSs (*Short Message Service*) que faz uso desta persistência implementada. Os dados que serão persistidos são providos pela Radnet e por sua API já desenvolvida e objeto de estudo [3] do mesmo grupo que compõe este trabalho.

2. O MODELO PUB/SUB

O modelo Pub/Sub idealiza como as aplicações distribuídas se comunicam em larga-escala reduzindo o acoplamento entre assinantes e publicadores de eventos. Um publicador é responsável por gerar eventos de um determinado interesse enquanto que subscritores ou assinantes consomem estes eventos gerados de acordo com o evento interessado. O modelo prevê um gerenciador de eventos responsável por notificar os assinantes de acordo com interesses registrados.

O modelo Pub/Sub idealiza ações realizadas entre os publicadores, os assinantes e o gerenciador de eventos. Uma das ações é a assinatura, nesta ação um assinante define os tipos

de eventos de seu interesse ao se registrar no gerenciador de eventos. Outra ação é a publicação de evento que consiste no envio de um evento gerado pelo publicador ao gerenciador de eventos. O gerenciador de eventos é responsável por notificar os assinantes da existência de um novo evento gerado por algum publicador de acordo com o interesse registrado anteriormente. Uma notificação pode ocorrer de maneira assíncrona, ou seja, não é necessário que o assinante esteja aguardando ativamente pelo evento no momento da notificação. A Figura 1 ilustra em alto nível o funcionamento do Pub/Sub.



Figura 1: Funcionamento do Modelo Pub/Sub

2.1 Desacoplamentos

Segundo o trabalho [4], o desacoplamento entre os publicadores e assinantes pode ser dividido em três tipos: espacial, sincronia e temporal. Dentro do ambiente de localização entre publicador e assinantes temos o desacoplamento espacial que provê a desobrigação de conhecer a localização entre publicador e assinante para comunicação. Na questão de processamento temos o desacoplamento de sincronismo entre publicadores e assinantes que não precisam estar bloqueados para gerar ou receber notificações sobre novos eventos, isto é, enquanto um evento está sendo produzido no publicador, o assinante podem estar executando outras tarefas e vice-versa. E, por último, temos o desacoplamento temporal que permite interação entre os nós de maneira independente do tempo, assim eventos ou notificações podem ser gerados, enviados para o gerenciador de eventos que os armazena até que os assinantes estejam disponíveis, por exemplo.

3. REDE AD HOC CENTRADA EM INTERESSES – RADNET

A Rede ad hoc centrada em interesses (Radnet) é uma rede ad hoc móvel (MANET) que usa comunicação orientada a interesse através de Prefixos Ativos [2] (PA). Um PA é composto de campos dividido em dois grupos: o prefixo (P) e interesse (I) (Figura 2). O prefixo é composto de campos numéricos gerados probabilisticamente. Cada dispositivo, ao entrar na rede, gera seu próprio prefixo seguindo uma distribuição probabilística e cada aplicação em execução define seu interesse [3].



Figura 2: Prefixo Ativo (PA)

3.1 Funcionalidade da Comunicação na Radnet

A Radnet possui funcionalidades de encaminhamento probabilístico de pacotes através de um filtro de casamento. Ao enviar um determinado pacote, os nós incluem o seu prefixo nele. Quando outro nó recebe este pacote, compara

cada campo do prefixo e verifica se há casamento de algum deles, se houver pelo menos um campo igual, o pacote é encaminhado. Outra funcionalidade é a formação de grupos, um nó ao receber um pacote contendo um interesse previamente registrado repassa-o à aplicação. Vários nós com o mesmo interesse poderão receber o mesmo pacote formando grupos do interesse. Além das funcionalidades citadas, temos o endereçamento baseado no prefixo ativo que identifica probabilisticamente um dispositivo unicamente na rede, entretanto pela própria questão de probabilidade há uma pequena chance de colisão de endereços, mas isto não é problema para a Radnet [2]. Para casos onde há a necessidade de pacote ponto-a-ponto o prefixo é usado, além da questão de encaminhamento.

A Radnet é uma implementação distribuída do modelo Pub/Sub para redes ad hoc móveis (MANET) [6]. Na Radnet (Figura 3), o gerenciador de eventos é constituído por toda a rede. As assinaturas de interesses são feitas na interface de rede do nó. As notificações de novos eventos ocorrem sempre que se recebe um pacote com um interesse que corresponde a algum pré-registrado. No trabalho [2] encontramos mais detalhes sobre o modelo utilizado para o desenvolvimento da Radnet.



Figura 3: Modelo Pub/Sub aplicado à Radnet

A Radnet também provê parcialmente os desacoplamentos previstos no modelo Pub/Sub. O desacoplamento espacial é atendido por padrão, dado que a rede é distribuída. O desacoplamento de sincronismo faz parte da implementação e da aplicação em uso, pois é possível o nó esperar informações de mais de um interesse enquanto executa outras ações, uma abordagem a nível de aplicação é o uso de *Callbacks* como proposto em [4]. O desacoplamento temporal não é implementado na Radnet, para adicionar o suporte a este tipo de desacoplamento, propomos neste trabalho a adição da persistência para que as aplicações possam receber as mensagens de seus interesse posteriormente, quando não estiverem em execução.

3.2 Exemplo de uma Radnet

A Figura 4 exemplifica a transmissão de pacotes em uma Radnet composta de quatro dispositivos, onde cada dispositivo tem um raio de alcance de transmissão delimitado pela circunferência tracejada. Esta rede usa um prefixo ativo composto de dois campos em P e um interesse I. A transmissão começa com um pacote do nó A com seu prefixo ativo: [1;5;Futebol], o nó B a recebe por estar no raio de alcance e a processa detectando que possui um campo do prefixo igual, logo a encaminha. O nó A, por sua vez, detecta que o pacote já foi processada e não a retransmite e não aceita. O nó C recebe o pacote do nó B por estar no raio de alcance e detecta que possui um campo do prefixo igual encaminhando o pacote. Quando o nó D recebe o pacote detecta que possui o mesmo interesse, e nenhum casamento

do prefixo, então a aceita entregando à aplicação registrada naquele interesse, e o nó B descarta o pacote.

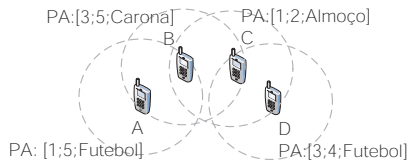


Figura 4: Exemplo de uma Radnet

3.3 Implementação

A Radnet foi implementada em sistemas Linux como um serviço a nível de usuário denominado de serviço REPD, sendo responsável pelo tratamento de pacotes recebidos pelos nós e por repassá-los as aplicações locais interessadas. Este serviço é executado em segundo plano e também é responsável por encapsular as mensagens originadas pelas aplicações e enviá-las a rede.

O serviço REPD apenas trata as questões de rede pertinentes a comunicação orientada a interesses. Com o foco na camada de aplicação, neste trabalho desenvolvemos o PMRep (*Protocol Messaging REP*). Este protocolo foi concebido para implementar a funcionalidade de SMS dentro de um pacote REPA. Seu desenvolvimento se baseou no protocolo HTML (*HyperText Markup Language*) onde existe um cabeçalho com as definições do PMRep separado do conteúdo determinado pela aplicação SMS.

As divisões no PMRep são chamadas de seções. A seção cabeçalho contém diversas informações separadas por linhas. A separação entre o cabeçalho e o conteúdo é feita por duas novas linhas em branco. Cada campo desta seção tem o formato “chave: valor”. Exemplos de uso dos campos, *nickname: bob*, *content-type: text/plain*. Outra seção é a de conteúdo que contém a própria mensagem com a formatação livre definida pela aplicação. Por enquanto não existe fragmentação de mensagens, logo o tamanho da mensagem está limitado ao tamanho máximo do pacote suportado pelo protocolo REPA.

Desenvolvemos o sistema RepaSMS que faz uso do protocolo PMRep definido anteriormente. Este sistema é composto de duas partes: o serviço RepaSMS e o gerenciador de mensagens, como ilustra a Figura 5.

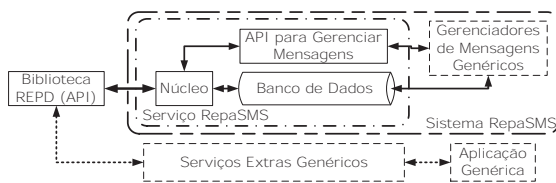


Figura 5: Sistema RepaSMS

O núcleo é parte do serviço RepaSMS que realiza o processamento das mensagens PMRep, o gerenciamento dos interesses, a notificação ao usuário sobre recebimento de novas mensagens e a inserção de novas mensagens e interesses na base de dados.

O componente importante deste trabalho é a base de dados do tipo SQLite para armazenamento das mensagens enviadas e recebidas, além dos interesses registrados. Com esta base de dados conseguimos adicionar a ideia de persistência ao protocolo REPA, pois permite o armazenamento das mensagens para visualização a qualquer tempo posterior. A constituição da base de dados é dada por duas tabelas: Interesses Registrados e Mensagens. Além disso, o acesso a esta base de dados é provida de forma compartilhada entre os gerenciadores de mensagens.

O gerenciador de mensagens é uma aplicação de exemplo do uso do serviço RepaSMS e tem como utilidade enviar, receber e gerenciar as mensagens através de uma interface para o usuário. Desenvolvedores podem criar aplicações para gerenciar as mensagens de acordo com sua necessidade utilizando o serviço RepaSMS e sua API para armazenar e recuperar mensagens.

4. AVALIAÇÃO

A avaliação prática do sistema proposto neste artigo foi feita entre o serviço REPD e o RepaSMS. Nesta avaliação usaram-se 5 celulares, 4 deles atuando como assinantes e 1 publicador de mensagens, utilizando de uma rede sem fio IEEE 802.11 em modo ad hoc. As especificações dos celulares que receberam as mensagens foram: Celular A - Motorola XT300, 256MB, CPU 528 Mhz, sistema Android 2.1; e Celular B - Motorola MB502, 512MB, CPU 600 MHz, sistema Android 2.1. Utilizamos como publicador de eventos o celular Samsung Galaxy S, 512MB, CPU 1Ghz, sistema Android 2.3.3.

A taxa de envio de mensagens fica limitada a velocidade do envio do serviço REPD. Coletamos as seguintes métricas: velocidade de transmissão contabilizando a quantidade total de *bytes* recebidos sobre o tempo de recebimento quando o dado é entregue pelo serviço REPD ao RepaSMS – denominado velocidade em baixo nível. E a velocidade de recebimento no gerenciador de mensagens, levando em conta o *overhead* do sistema operacional para repassar as mensagens do serviço RepaSMS para o gerenciador de mensagens – denominado velocidade em alto nível. Outra métrica coletada é a taxa de entrega de mensagens.

Estas métricas foram obtidas enviando 10, 100 e 1000 mensagens variando o seu tamanho em 200, 700 e 1450 *bytes*, salientando que os mensagens foram trocadas em apenas um salto, saindo do publicador diretamente para todos os assinantes na rede. Dos testes realizados consideramos apenas as métricas do celular B por ter obtido menores desvios padrão dos dados observados. Os outros nós foram utilizados para interferirem no meio de comunicação e tornar o experimento mais realista. Foram executados 20 vezes cada conjunto de experimentos, os resultados apresentados são valores médios com nível de confiança de 95%, com erros menores que 12% para velocidade em alto nível, devido a alta variabilidade do sistema operacional, e 8% para os demais casos.

4.1 Resultados e Análises

Na Figura 6 verificamos o aumento de velocidade conforme o tamanho dos pacotes crescem em baixo nível. Segundo [1], o meio físico da rede IEEE 802.11 proporciona maiores taxas de velocidade de transmissão conforme o tamanho do pacote

aumenta, sendo o limite de tamanho destes pacotes definido pelo MTU (*Maximum Transmission Unit*). Nota-se a pouca variação da velocidade conforme a variação da quantidade de mensagens enviadas em rajada, a explicação para isso é o uso de poucos nós e somente um salto na comunicação.

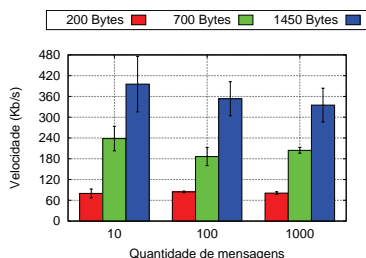


Figura 6: Velocidade em baixo nível

Na Figura 7 verificamos, também, o aumento da taxa de transmissão em relação ao tamanho das mensagens enviadas e a pouca variação da velocidade conforme variamos a quantidade de mensagens enviadas. A explicação para estes fatos é o mesmo que para velocidade em baixo nível.

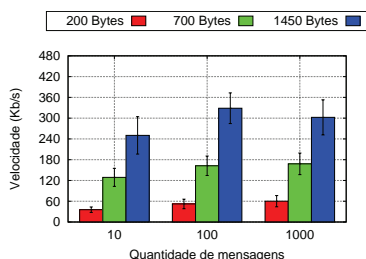


Figura 7: Velocidade em alto nível

Comparando as Figura 6 e 7 notamos significativa interferência do sistema operacional na velocidade do protocolo conforme subimos do nível de rede para aplicação. Verificamos as velocidades máximas de 395,6 Kb/s em baixo nível com 10 mensagens e 328,7 Kb/s em alto nível com 100 mensagens. Uma observação é que há *overhead* do sistema operacional tanto no envio quanto no recebimento.

Quanto a taxa de entrega, de acordo com a Figura 8, obteve-se pouca variação no intervalo de 91,0% a 98,0% indicando uma boa estabilidade da rede nas condições de um salto com cinco nós. A baixa taxa de perda é, também, dependente do ambiente de testes, se aumentarmos o número de nós e saltos teríamos maiores perdas de mensagens devido ao modelo probabilístico de encaminhamento da Radnet e a perdas por interferências meio de comunicação.

5. TRABALHOS RELACIONADOS

O trabalho [4] apresenta uma revisão de vários paradigmas de interação relacionando-os com o modelo Publicador/Subscriber (Pub/Sub). O trabalho [6] faz análises dos diversos paradigmas de Pub/Sub, igualmente, obtendo métricas como taxa de entrega de mensagens e consumo de energia dos nós. Outro trabalho [5] apresenta *benchmarks* para avaliar Message-oriented middleware (MOM) que implementam o modelo Pub/Sub e, como caso de estudo, simulação de cenários Pub/Sub (Fila, tópicos, barramento).

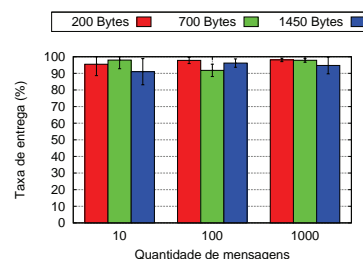


Figura 8: Taxa de entrega

6. CONCLUSÃO

Neste trabalho, introduzimos a persistência em redes ad hocs centradas em interesses, denominadas Radnets, de modo a armazenar as mensagens recebidas por um nó para serem acessíveis posteriormente pela aplicação, mesmo quando esta não está ativa no momento de sua recepção ou simplesmente o desacoplamento temporal. A abordagem usada foi a criação do sistema RepaSMS, sendo este um message-oriented middleware (MOM), atuando em um cenário de troca de mensagens SMS. Este sistema foi avaliado e revelou a velocidade máxima obtida pelo serviço REPA e RepaSMS usando os protocolos REPA e PMRep, a variação de velocidade entre os cenários apresentados, expondo o *overhead* do sistema operacional. Além disso, obteve-se bons resultados de entrega das mensagens. Trabalhos futuros compreendem avaliar a latência de transmissão, incluir novos cenários para métricas, proporcionar o suporte a busca de informação distribuída nas Radnets e o desenvolvimento de aplicações que se beneficiem desta distribuição.

Referências

- [1] B. Bing. Measured performance of the ieee 802.11 wireless lan. In *Local Computer Networks, 1999. LCN '99. Conference on*, pages 34–42, 1999.
- [2] R. C. Dutra. *Modelo de Comunicação Distribuída Orientada a Interesse*. PhD thesis, PESC/COPPE/Universidade Federal do Rio de Janeiro, 2012.
- [3] R. C. Dutra, H. F. Moraes, and C. L. Amorim. Interest-centric mobile ad hoc networks. *Network Computing and Applications, IEEE International Symposium on*, pages 130–138, 2012.
- [4] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. *ACM Comput. Surv.*, 35(2):114–131, June 2003.
- [5] K. Sachs, S. Appel, S. Kounev, and A. Buchmann. Benchmarking publish/subscribe-based messaging systems. In *Proceedings of the 15th international conference on Database systems for advanced applications, DASFAA'10*, pages 203–214, Berlin, Heidelberg, 2010. Springer-Verlag.
- [6] S. Yoo, J. H. Son, and M. H. Kim. A scalable publish/subscribe system for large mobile ad hoc networks. *Journal of Systems and Software*, 82(7):1152 – 1162, 2009.