

Avaliando a Utilização do Bluetooth na Criação de Ambientes Computacionais Ubíquos

Anderson Luiz Menezes¹, Marcio Seiji Oyamada¹

¹UNIOESTE - Universidade Estadual do Oeste do Paraná
Rua Universitária, 2069. Jardim Universitário.
Caixa Postal 711 - CEP 85819-110 - Cascavel, PR

{andermenezes, msoyamada}@gmail.com

Resumo. Atualmente a busca por mobilidade e transparência tem se intensificado de forma realmente considerável. Esta realidade associada a velocidade da evolução tecnológica dos dispositivos portáteis e suas tecnologias, como o surgimento do Bluetooth, nos possibilita pensar na criação de ambientes computacionais ubíquos voltados para tais dispositivos. A tecnologia Bluetooth apresenta características que propiciam a formação de redes entre seus dispositivos, o que, com o auxílio de protocolos específicos e com a utilização do conceito de redes ad-hoc, permite a comunicação necessária na constante cooperação requisitada pela ubiquidade. O presente trabalho apresenta o processo de adaptação do protocolo de roteamento ad-hoc AODV (Ad-hoc on-Demand Distance Vector), descrito pela RFC-3561, para ser utilizado junto a dispositivos dotados da tecnologia Bluetooth. Serão discutidas as limitações do Bluetooth para o suporte ao AODV, assim como serão exibidos os resultados obtidos com os testes utilizando dispositivos reais.

1. Introdução

Atualmente, podemos perceber que a busca pela mobilidade tem se intensificado de forma surpreendente. A quantidade de dispositivos móveis existentes no mercado, assim como a quantidade e qualidade dos serviços disponíveis, vem crescendo e se solidificando de maneira visivelmente promissora. Não é difícil nos depararmos com pessoas utilizando seus PDAs (*Personal Digital Assistants*) e laptops em centros comerciais, restaurantes, aeroportos, dentre outros diversos locais comuns ao nosso dia-a-dia. Mais popular ainda é o uso dos telefones celulares. Há algum tempo esses aparelhos deixaram de ser simples telefones, passando a possuir características que despertam a curiosidade tanto das pessoas continuamente ligadas aos avanços tecnológicos quanto daquelas mais leigas que, mesmo assim, não abrem mão de sua utilização. Certamente devido a facilidade de acesso, os celulares vem ganhando, a cada dia mais, um número maior de usuários, acarretando no crescimento do interesse pelo estudo e desenvolvimento voltado para tais aparelhos.

Mediante este cenário, o foco do presente trabalho está voltado para os telefones celulares. Como a grande maioria dos aparelhos mais atuais dispõem da tecnologia Bluetooth, este trabalho visa o estudo da utilização desta tecnologia na criação de ambientes computacionais ubíquos. Mesmo após grandes avanços, o Bluetooth ainda pode ser

considerado subutilizado, uma vez que possui características e recursos muito relevantes no que diz respeito a criação de redes *ad-hoc*, “indispensáveis” devido sua propriedade de total descentralização. Assim como o Bluetooth, atualmente, grande parte dos aparelhos de telefonia celular suporta a instalação de aplicativos Java, facilitando o desenvolvimento de aplicações para celulares.

Sendo assim, o objetivo deste trabalho é estudar a viabilidade da utilização da tecnologia Bluetooth, apresentando as observações verificadas através da criação da ubiquidade em ambientes reais, não limitando-se apenas a simulação. É importante mencionar que questões de otimização, como economia no consumo de energia, ou segurança não estão como elementos primários, podendo ser postergadas para trabalhos futuros, ficando como principal ponto a criação do ambiente computacional ubíquo com a utilização dos dispositivos dotados da tecnologia Bluetooth.

2. Trabalhos Relacionados

Existem diversos trabalhos sendo desenvolvidos nesta área. Para comprovar este interesse dois deles são apresentados nesta seção.

O primeiro trata-se da adaptação do protocolo de roteamento *ad-hoc* LUNAR (*Lightweight Underlay Network Ad-Hoc Routing*) [1]. Para este trabalho o autor coloca três pontos cruciais observados na adaptação: a descoberta de novos vizinhos, a formação das Piconets e Scatternets e o roteamento entre e dentro das Piconets. Estas observações são importantes por pertencer exclusivamente a forma de funcionamento do Bluetooth, relevando suas características ligadas a formação de redes.

Já com relação a ambientes ubíquos, temos o Projeto UbiqMuseum [2]. Seu objetivo é a criação de um ambiente capaz de fornecer informações sobre as obras disponíveis no museu diretamente aos dispositivos móveis dos visitantes. Para tal, uma infraestrutura composta de servidores, redes cabeadas, redes sem-fio IEEE 802.11b e Bluetooth foi desenvolvida. Conforme o cliente passeia em meio as obras, seu dispositivo efetua buscas constantes pelos pontos de informação do museu; quando um é encontrado, inicia-se então a busca por novas informações; caso esta exista, o cliente é informado e questionado sobre o interesse em visualizá-la. Diferente de outros trabalhos, o UbiqMuseum possui uma implementação experimental, não limitando-se somente a simulação.

3. Computação Ubíqua

Se pararmos para analisar, a ubiquidade já é uma realidade em nossas vidas. Ao caminhar pelas ruas de uma grande cidade, nos deparamos facilmente com quantidades astronômicas de informações, no entanto, acabamos assimilando-as sem, nem ao menos, prestar atenção na forma em que ela está disponível.

A Computação Ubíqua, definida por Mark Weiser, propõe a existência de uma nova ideologia, onde a computação estaria presente em todo lugar e a qualquer momento. Segundo Weiser: “*As tecnologias mais brilhantes são aquelas que desaparecem. Elas se misturam com as estruturas presentes no nosso dia-a-dia até que não se possa perceber sua presença*” [3], ou seja, a idéia é que sejam desenvolvidos ambientes nos quais a tecnologia estaria totalmente transparente às necessidades diárias dos usuários.

Estender essa ubiquidade para computação não é algo tão simples, ao menos não da forma como Weiser propõe. Para alcançar o que podemos chamar de “ubiquidade ideal” na computação, os dispositivos teriam que sofrer mudanças extremamente significativas, principalmente com relação aos seus tamanhos e sua dinamicidade. Eles precisam ser capazes de reconhecer o ambiente onde se encontram e adaptar seus serviços de acordo com o contexto deste ambiente; eles precisam ser capazes de se auto-configurar para continuar suprindo as necessidades do usuário independente de toda e qualquer transição que possa ocorrer.

4. Tecnologia Bluetooth

Bluetooth é um padrão para comunicação sem fio que utiliza tecnologia de transmissão via rádio de curto alcance, fornecendo baixo custo e baixo consumo de energia. Os dispositivos desta tecnologia possuem uma velocidade de comunicação de 1 Mbps a 3 Mbps (representando, respectivamente, as versões 1.2 e 2.0 + *Enhanced Data Rate*) e são divididos em três classes que indicam a faixa de alcance conseguida, como mostra a Tabela 1 [4].

Tabela 1: Classes da Tecnologia Bluetooth

Classes	Alcances	Potência Máxima de Saída
Classe 3	1 - 3 metros	1 mW (0 dBm)
Classe 2	10 - 33 metros	2,5 mW (4 dBm)
Classe 1	100 - 300 metros	100 mW (20 dBm)

As redes Bluetooth podem ser de dois tipos: Piconet (formada por oito dispositivos) e Scatternets (associação de Piconets).

4.1. Piconets e Scatternets

A criação de uma rede Bluetooth baseia-se no conceito de canais. Para que os dispositivos estejam localizados em uma mesma rede, eles necessitam estar compartilhando um mesmo canal. Um ou mais dispositivos utilizando o mesmo canal físico forma o que chamamos de Piconet.

Essas Piconets são formadas por até oito nós ativos, onde um é eleito como mestre e os demais permanecem como escravos. Esta limitação é imposta apenas para os nós que se encontram ativos¹.

Existe, também, a possibilidade de um dispositivo Bluetooth servir como ponte unindo as Piconets. Essa interconexão de Piconets forma as chamadas Scatternets. As Scatternets vem suprir a limitação do número de dispositivos ativos imposta pelas Piconets, uma vez que podem existir n dispositivos permitindo esta junção.

5. Ad-Hoc On-demand Distance Vector (AODV)

A principal e fundamental diferença entre uma rede sem fio tradicional e uma *ad-hoc*, é a infra-estrutura. Quando se trata de redes *ad-hoc*, essa infra-estrutura deixa de existir, fazendo com que os próprios hospedeiros fiquem responsáveis pelos vários serviços necessários para o funcionamento de uma rede [5].

¹ Podem existir até 255 dispositivos não ativos (*parked state*)

No campo das redes *ad-hoc*, os protocolos são divididos em dois grupos: os **proativos** e os **reativos**. AODV, descrito na RFC-3561 [6], é um exemplo de protocolo reativo. Desenvolvido para suportar de dezenas a milhares de dispositivos, ele proporciona uma transmissão veloz de dados aos destinatários desejados, sem, para isso, precisar manter rotas armazenadas, com exceção das que estão atualmente em uso.

Se comparado a outros protocolos utilizados nas redes tradicionais, AODV apresenta três tipos especiais de mensagens: *Route Requests* (RREQs), *Route Replies* (RREPs) e *Route Errors* (RERRs). Para encontrar uma rota válida, mensagens do tipo RREQ são disparadas por *broadcast*, as quais se propagam até atingir o destino ou um nó intermediário que possui o restante da rota em sua tabela. Para ativar a rota encontrada, uma mensagem do tipo RREP é enviada de volta ao nó de origem, dessa vez por *unicast*.

Para manter a integridade do roteamento, todos os nós participantes possuem uma lista de precursores, contendo o endereço de todos os vizinhos que, eventualmente, poderiam utilizá-lo como parte de alguma rota. Caso ocorra uma quebra em determinada rota, ou seja, caso dado nó perceba que o próximo salto não está mais ativo, uma mensagem RERR é enviada a todos os presentes em sua lista de precursores, evitando, assim, que tentativas de utilização daquela rota inexistente sejam efetuadas.

6. AODV Sobre o Bluetooth

Para realização deste trabalho foi utilizado o conjunto de APIs JSR-82 [7] e uma implementação do protocolo AODV (denominada J-Adhoc [8]) como base para a adaptação. J-Adhoc é distribuída sob licença GPL (*GNU General Public License*), sendo escrita em Java e baseada na RFC-3561 [6].

Mediante ao foco do trabalho (utilização do Bluetooth como interface de rede), toda parte de comunicação utilizada pelo J-Adhoc necessitou ser alterada, gerando uma nova implementação chamada BtJAdhoc (Bluetooth Java Ad-Hoc), na qual as alterações principais ocorreram nas classes responsáveis pelo processo de envio e recebimento de pacotes, as quais estão implementadas no pacote Btpcap.

6.1. Adaptação

Buscando não modificar as características básicas do algoritmo do protocolo AODV, as alterações aplicadas foram direcionadas principalmente às questões de comunicação, deixando o gerenciamento interno praticamente inalterado.

J-Adhoc utiliza a biblioteca Java Jpcap [9] (também regida pela licença GPL) para o monitoramento do tráfego da rede, a qual permite às aplicações enviar e capturar pacotes através das interfaces de rede, manipulando-os posteriormente em Java. Essa foi a parte que mais sofreu alterações. Uma nova biblioteca com as funcionalidades do Jpcap foi desenvolvida para o Bluetooth. Esta nova biblioteca chamada de Btpcap manteve apenas algumas poucas características da implementação original. Para a implementação no Bluetooth, toda a variedade de pacotes suportada anteriormente pode ser descartada, sendo mantidas apenas algumas informações importantes para o algoritmo, como endereço de origem, endereço de destino, tamanho do pacote, dentre outras, com as quais foi possível criar dois novos tipos de pacotes, sendo eles:

- **BtAODVdgram**: para criação deste tipo de pacote foram utilizadas algumas informações existentes nos datagramas UDP. Através desses pacotes é que o BtJAdhoc envia as mensagens de controle do AODV;
- **BtPacket**: semelhante ao antigo IPPacket existente na implementação da classe Jpcap. Com o auxílio deste pacote é que o Btpcap manipula os dados que devem ser transmitidos entre os hospedeiros, ou seja, as informações trocadas entre os nós da rede.

Outro ponto importante alterado na biblioteca original foi a forma utilizada para captura dos pacotes. No Bluetooth serviços podem ser “registrados” e disponibilizados para que os demais dispositivos no alcance possam utilizá-los de alguma forma. Sendo assim, a classe responsável pela captura de pacotes (Btpcap) nada mais é do que um servidor. Quando o protocolo entra em funcionamento um serviço é disponibilizado (AODV-Protocol) e um canal de comunicação é aberto. Sempre que um dispositivo conecta-se ao serviço e envia algum pacote, os métodos da classe verificam qual o tipo de pacote recebido (pacote de controle AODV ou um BtPacket) e o repassa para a classe do protocolo responsável pela sua manipulação.

Com relação ao envio, a estratégia também teve que ser totalmente revista. Devido ao fato de que o Bluetooth não implementa *broadcast*, um grande gargalo é gerado no momento do envio das mensagens de controle ou, até mesmo, de outro tipo de pacote com necessidade de propagação *broadcast* ou *multicast*. A estratégia adotada foi realizar uma descoberta² para determinar quais são os dispositivos que estão no seu alcance. Procurando reduzir o *overhead* gerado por esse processo, optamos por manter armazenado durante um determinado período de tempo os dispositivos encontrados na última busca, ao invés de efetuar uma nova busca a cada envio. Após os dispositivos no alcance terem sido encontrados, uma verificação é efetuada em cada um para manter armazenado apenas os que possuem o serviço do protocolo AODV executando.

Em relação ao núcleo do protocolo J-Adhoc, algumas pequenas alterações também foram necessárias. Uma delas está relacionada ao endereçamento. A implementação original trabalha com endereços IP, o qual poderia ser mantido, no entanto, optou-se por trabalhar diretamente com o endereço do Bluetooth (com as mesmas características dos endereços MAC das interfaces de rede tradicionais). Esse endereço é único em cada dispositivo, anulando a necessidade de associar um dado endereço a cada dispositivo.

Existem também funcionalidades que foram totalmente extintas, como a atualização da tabela de roteamento do sistema operacional, por exemplo, ou a propagação das mensagens de *Hello*, utilizadas para avisar os nós vizinhos e para o controle das quebras de ligação. Com a existência do processo de busca necessário em todos os dispositivos que utilizam Bluetooth, o envio do *Hello* não é mais necessário.

6.2. Limitações e Dificuldades Encontradas

A tentativa de adaptar um robusto protocolo de roteamento impôs inúmeros complicadores relacionados tanto ao Bluetooth quanto ao desenvolvimento utilizando JavaME.

²O processo de descoberta de vizinhos do Bluetooth verificado durante os testes foi de cerca de dez segundos.

Primeiramente, algumas estruturas originalmente usadas na implementação do J-Adhoc, como as `LinkedLists` ou a interface `Map` (substituídas por `Vectors`) e a classe `InetAddress` da `java.net` (substituída por endereços representados em `Strings`), não puderam ser mantidas, visto que a JavaME não as suportava.

Ainda com relação a JavaME, o envio de mensagens de qualquer tipo também foi comprometido, uma vez que esta tecnologia não implementa serialização, não permitindo que instâncias de classes pudessem ser enviadas diretamente. Dessa forma, para transmissão de um pacote que é representado por um objeto Java, é necessário transformá-lo em um *array* de *bytes*. Ao ser recebido, esse *array* é então processado e os campos atribuídos a uma nova instância da classe no dispositivo de destino.

As limitações nativas dos dispositivos Bluetooth também impuseram algumas dificuldades na implementação. Como exemplo, temos os bloqueios efetuados a cada atividade do Bluetooth. Foi observado que apenas uma determinada operação pode ser efetuada por vez, ou seja, ele só pode buscar dispositivos, transmitir, ou receber informações em um determinado instante. Essa característica gera um grande atraso na execução das funcionalidades do protocolo e também no próprio gerenciamento das Piconets e Scatternets, tendo em vista que muitos conflitos podem acontecer durante as atividades.

Da mesma forma, o processo de busca do Bluetooth também gerou alguns problemas na implementação, principalmente em relação ao tempo necessário para realizá-la. Considerando que dispositivos móveis podem entrar e deixar a área de cobertura de outros dispositivos com extrema rapidez, seria interessante que os dispositivos mantivessem a tabela de vizinhos a mais atualizada possível. Na implementação proposta neste trabalho optou-se por um compromisso entre tempo de envio e precisão, visto que a tabela de vizinhos não é atualizada a cada envio, porém a cada 30 segundos. Desta forma, evita-se que a cada envio um tempo adicional de 10 segundos (referente ao processo de descoberta) seja consumido. Por outro lado, um dispositivo que acabou de entrar na área de cobertura e que poderia auxiliar no roteamento de pacotes não será detectado imediatamente.

7. Estudos de Caso

Afim de verificar a real viabilidade da aplicação da tecnologia Bluetooth em ambientes ubíquos, dois estudos de caso foram desenvolvidos.

O primeiro, chamado **Sistema Ubíquo de Propagação de Notícias**, consiste em divulgar determinadas notícias para o maior número de dispositivos possível. Para alcançar esse objetivo, um computador ficará responsável por coletar algumas notícias e iniciar a propagação, enviando-as para seus vizinhos na Piconet. Estes, por sua vez, distribuem para os demais em seu alcance. Essa interação continua até que todos os dispositivos recebam a notícia.

O segundo estudo de caso, denominado **Sistema Ubíquo de Controle de Frequência**, pretende prover um serviço de controle de frequência totalmente ubíquo, onde qualquer intervenção dos usuários é totalmente desnecessária. Após o aluno ter adentrado a sala de aula e seu aparelho celular ter encontrado um outro dispositivo no seu alcance, iniciará o processo de notificação da presença, definindo a rota até o *laptop* do professor e enviando seus dados.

8. Testes Efetuados e Resultados Obtidos

Para realização dos testes foram usados três dispositivos, sendo eles dois aparelhos celulares Nokia [10] (um da Série 40 e outro da Série 60) e um computador pessoal utilizando um adaptador Bluetooth USB da marca Encore [11].

Os testes basearam-se na verificação do tempo gasto e na distância máxima suportada pelos dispositivos na utilização do protocolo para o envio de mensagens. O foco dos testes se manteve sobre o protocolo de roteamento visando verificar sua eficiência quanto a criação dos ambientes ubíquos propostos nos estudos. Através de uma análise feita na atuação do protocolo BtJAdhoc é possível ter uma noção do funcionamento dos sistemas desenvolvidos, no entanto, como não foi possível efetuar testes com um número grande de dispositivos, esta noção baseia-se apenas em conclusões que podem ser tiradas do comportamento observado em baixa escala.

Para obter os valores exibidos na Tabela 2, foram enviados pacotes de 126 bytes contendo apenas uma mensagem de texto, tanto diretamente entre dois dispositivos quaisquer, quanto entre esses mesmos dois dispositivos, porém com a necessidade de um intermediário para formação da ponte de comunicação (vide Figura 1), devido a distância máxima de alcance ter sido ultrapassada. A Tabela 2 apresenta os resultados obtidos nos testes, onde os valores foram obtidos após dez execuções feitas para cada caso, na qual o tempo médio representa a média aritmética entre os tempos obtidos.

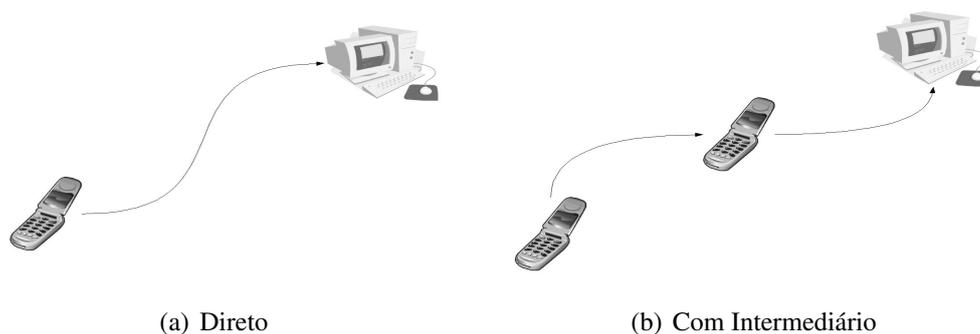


Figura 1: Tipos de envio utilizados

Tabela 2: Resultados Obtidos

Testes	Melhor Execução	Pior Execução	Tempo Médio
Envio Direto	23 seg.	8 min. e 30 seg.	6 min. e 52 seg.
Envio com Intermediário	3 min. 57 seg.	19 min. 23 seg.	14 min.

Essa enorme diferença obtida entre os melhores e piores tempos de execução são em decorrência das dificuldades que os dispositivos encontraram ao tentar transmitir seus pacotes de controle AODV para a descoberta da rota e, também, da dificuldade encontrada para a transmissão final. Os menores tempos de execução mostram que pode ocorrer comunicação sem muitas ocorrências de conflitos, no entanto, essas execuções ocorrem muito esporadicamente.

Como as tabelas de roteamento sempre iniciam vazias, todo processo de descoberta de rota teve que ser executado para os dois tipos de teste (com e sem necessidade de intermediário). É fácil observar o motivo pelo qual o tempo necessário para a

comunicação aumenta nos casos referentes a transmissão com existência de dispositivos intermediários. Tendo em vista que a quantidade de conflitos pode aumentar e que, nos testes efetuados, só poderia existir apenas um caminho para o hospedeiro de destino, todo processo de descoberta de rota e envio da mensagem final teve que ficar aguardando até que os dispositivos conseguissem efetivar suas atualizações.

Em alguns casos, tanto na comunicação direta quanto na comunicação com necessidade de saltos, a mensagem final simplesmente não foi entregue. Isso ocorreu devido a grande quantidade de conflitos não permitir que os pacotes de controle AODV ou o próprio pacote contendo a mensagem fossem transmitidos antes que seu tempo de vida expirasse, ou seja, a existência dos conflitos fez com que o processo fosse interrompido e os pacotes descartados. Este problema acontece com maior frequência quando trata-se da necessidade de formação de pontes (representando cerca de 90% das tentativas de execução, num total de 10 tentativas), pois o “dispositivo do meio” deve conciliar as mensagens de controle recebidas pelos dois outros dispositivos com o envio das suas, além de ocorrer travamentos constantes no adaptador USB utilizado. Nas execuções feitas através do envio direto, a situação é exatamente oposta ao envio com intermediário, onde pode-se observar a conclusão do envio em mais de 90% das tentativas.

Através dos testes foi possível observar, também, que a distância pode variar levando em consideração o dispositivo e o ambiente no qual se está realizando o experimento. Barreiras físicas, como paredes ou outros objetos, são fatores que auxiliam na redução desse alcance, fazendo com que a utilização dos saltos permitidos pelo protocolo de roteamento sejam realmente necessários. Com a utilização do protocolo desenvolvido conseguimos os alcances máximos apresentados na Tabela 3, sendo estes valores obtidos de forma empírica e em ambiente real. Não foi possível observar, porém, se esta distância influenciou consideravelmente nos tempos de execução, pois os conflitos ocorrem de maneira muito irregular, dificultando analisar o comportamento por outros pontos de vista.

Tabela 3: Distâncias com o Protocolo BtJAdhoc

Teste	Distância
Sem Intermediário	8.6 m
Com Intermediário	16,8 m

8.1. Considerações

Inserir o Bluetooth em ambientes reais de redes permitiu um contato direto com suas limitações e suas possibilidades de utilização.

Em nosso sistema de controle de presença, por exemplo, a possível existência de grande fluxo de informações é clara, pois, supondo a necessidade de formação de pontes para propagação dos dados da chegada do aluno, fica fácil verificar que o número de mensagens de controle trocadas entre os dispositivos presentes no ambiente ubíquo é consideravelmente alto. Através então dos resultados obtidos com o protocolo, podemos concluir que existe a possibilidade de um aluno simplesmente não ter sua presença confirmada, mesmo estando com o Bluetooth ativado e com o sistema em execução, o que geraria grandes transtornos. Os testes realizados com o sistema basearam-se apenas nos três dispositivos, sendo assim, a relação exata entre as dimensões de uma sala de aula

e a real necessidade da geração e pontes não foi verificada. No entanto, supondo que realmente exista essa necessidade, e baseando-se nos experimentos efetuados, podemos inferir que Bluetooth ainda deixa a desejar quando a quantidade de comunicação é muito intensa.

Para propagação de pacotes via *multicast* ou *broadcast*, a implementação do BtJAdhoc desempenhou suas funcionalidades de forma satisfatória. Tendo em vista que o Bluetooth não implementa tais tipos de envio de forma nativa, o protocolo desenvolvido teve que ser capaz de controlar com perfeição os dados transmitidos, para que estes não permanecessem “inundando” os dispositivos infinitamente. Para o *desktop* que gerencia o envio de notícias, o sistema desenvolvido atribui um número de sequência específico para cada nova notícia, permitindo que este número funcione como uma forma de controle para os demais dispositivos.

Algo que surgiu durante nossos testes e que não podemos afirmar ser um problema especificamente do *hardware* utilizado ou do *driver* de controle, foi o travamento dos serviços do adaptador Bluetooth USB quando a quantidade de atividade aumentava. Em grande parte das execuções de teste do protocolo e dos sistemas ubíquos (principalmente o de controle de frequência) o adaptador simplesmente parava de funcionar devido a quantidade de informação que estava sendo trocada no momento.

Outro grande complicador observado no aparelho da Nokia Série 40 utilizado nos testes, foi o fato do dispositivo não permitir que as comunicações via Bluetooth possam ocorrer de forma “silenciosa”. O aparelho solicita a intervenção do usuário antes de efetuar cada atividade de comunicação, seja ela de envio ou recepção de dados.

9. Conclusões

Este trabalho apresentou a adaptação de um protocolo de roteamento *ad-hoc* para o Bluetooth e sua utilização na criação de um ambiente computacional ubíquo real, mesmo considerando as inúmeras limitações existentes nas tecnologias utilizadas. Porém sua aplicação prática pode ser inviável devido, principalmente, aos problemas de atraso gerados pela busca de dispositivos e a baixa velocidade de comunicação.

É importante mencionar que nossas limitações tecnológicas impediram que a ubiquidade ideal proposta e defendida por Mark Weiser [3] em muitos de seus trabalhos fosse alcançada. Da mesma forma que uma implementação voltada para os princípios da tecnologia calma [12]. Para possibilitar a execução do protocolo e dos sistemas propostos pelo trabalho nos dispositivos móveis, foram necessárias efetuar suas instalações em cada dispositivo, necessitando ainda que uma interface com o usuário fosse disponibilizada, o que torna a intervenção do usuário muito grande para a proposta de Weiser.

Foi possível verificar uma discrepância significativa entre diferentes modelos de aparelhos celulares, o que dificulta consideravelmente o processo de desenvolvimento de uma aplicação genérica, principalmente se levarmos em conta a sofisticação de determinados tipos de aplicações, como é o caso do objeto-alvo deste trabalho.

Com base nos resultados, é possível perceber que se o ambiente ubíquo necessitar de grande dinamicidade, ou seja, caso ele necessite de constante troca de informações e grande agilidade nas mesmas, o Bluetooth ainda não é a alternativa mais indicada, principalmente no caso da necessidade de comunicação *ad-hoc*. Por outro lado, pode ser total-

mente aceito em algum tipo de atividade que não exija respostas imediatas e onde o custo-benefício seja um fator preponderante, lembrando que o Bluetooth fornece comunicação via rádio de baixo custo e baixo consumo de energia.

Outro fator que permite ao Bluetooth permanecer como uma boa opção é sua crescente adoção, podendo ser encontrado em telefones celulares, *notebooks*, PDAs, ou ser adquirido separadamente através dos adaptadores USB, permitindo que as aplicações que o utilizam possam atingir a um número muito grande de usuários.

Os trabalhos futuros compreendem no desenvolvimento de estratégias mais eficientes, as quais possibilitariam que as comunicações em meio a utilização do protocolo adaptado pudessem ocorrer com um menor número de conflitos de atividade do Bluetooth, tornando sua eficiência na comunicação e velocidade de troca de informações mais aceitáveis. Espera-se, também, a possibilidade de desenvolver testes com uma quantidade maior de celulares, colocando o protocolo adaptado e o sistemas frente a um ambiente mais complexo, possibilitando uma análise mais aprofundada.

Referências

- [1] of Basel, U., “LUNAR - Lightweight Underlay Network Ad hoc Routing,” Consultado na Internet em: 15/04/2008, Disponível em: <http://cn.cs.unibas.ch/projects/lunar/>.
- [2] Cano, J.-C., Manzoni, P., and Toh, C. K., “UbiqMuseum: A Bluetooth and Java Based Context-Aware System for Ubiquitous Computing,” *Wireless Personal Communications*, Vol. 38, No. 2, 2006, pp. 187–202.
- [3] Weiser, M., “The computer for the 21st century,” *Scientific American*, Vol. 265, No. 3, 1991, pp. 66–75.
- [4] Bluetooth, “Specs,” Consultado na Internet em: 11/02/2008, Disponível em: <http://www.bluetooth.com>.
- [5] Kurose, J. F. and Ross, K. W., *Redes de Computadores e a Internet: Uma abordagem top-down*, Addison Wesley, São Paulo, trad. 3 ed. ed., 2006.
- [6] Perkins, C., Belding-Royer, E., and Das, S., “Ad hoc On-Demand Distance Vector (AODV) Routing,” Consultado na Internet em: 11/04/2008, Disponível em: <http://www.ietf.org/rfc/rfc3561.txt>.
- [7] Motorola, Austin, *Java APIs for Bluetooth Wireless Technology (JSR 82)*, 2005.
- [8] University of Bremen, Bremen, *JAdhoc System Design*, 2003.
- [9] Fujii, K., “Jpcap - a Java library for capturing and sending networks packets,” Consultado na Internet em: 02/09/2008, Disponível em: <http://netresearch.ics.uci.edu/kfujii/jpcap/doc/>.
- [10] Nokia, “Nokia Connecting People,” Consultado na Internet em: 02/11/2008, Disponível em: <http://www.nokia.com>.
- [11] Encore, “Encore Electronics,” Consultado na Internet em: 02/11/2008, Disponível em: <http://www.encore-usa.com>.
- [12] Weiser, M. and Brown, J. S., “Designing Calm Technology,” Consultado na Internet em: 11/07/2008, December 1995.