

Arquitetura de Compartilhamento Transparente de Conteúdos Entre Dispositivos Móveis em Redes Oportunistas

Transparent Sharing Architecture of Content Between Mobile Devices in Opportunistic Networks

Charles Tim Batista
Garrocho
ctgarrocho@gmail.com

Maurício José da Silva
badriciobq@gmail.com

Ricardo Augusto Rabelo
Oliveira
rrabelo@gmail.com

Laboratório IMóbilis
Departamento de Computação
Universidade Federal de Ouro Preto – MG, Brasil

RESUMO

Redes oportunistas são uma das evoluções mais interessantes de MANETs, onde permitem diversas aplicações, como em descarregar o tráfego celular, comunicações em situações de emergência e contorno a censura. O crescente aumento de dispositivos móveis deve, em tese, promover redes oportunistas. Porém, na prática, as tecnologias atuais para redes oportunistas, como Wi-Fi ad-hoc, Bluetooth e Wi-Fi Direct, ou não são disponíveis nos dispositivos atuais, ou exigem interação indesejada do usuário para estabelecer conectividade. Para superar essas deficiências, propomos uma arquitetura que utiliza o modo infraestrutura do Wi-Fi, de forma que promova a comunicação entre os dispositivos, permitindo a troca transparente de conteúdos sem a interação do usuário. Duas aplicações que empregam o uso dessa arquitetura são apresentadas. A primeira, de dispositivos pessoais, mostrou-se escalável em testes com até nove dispositivos. A segunda, de veículos, mostrou-se viável quando aplicada em cenários com baixa velocidade, gerando uma baixa perda de pacotes e altas taxas de transmissão.

Palavras-Chave

Arquitetura, Comunicações Oportunistas.

ABSTRACT

Opportunistic networks are one of the most interesting developments of MANETs, which allow various applications, such as downloading the mobile traffic, communications in emergency situations and contour censorship. The increasing number of mobile devices should, in theory, promote

opportunistic networks. However, in practice, current technologies for opportunistic networks, such as Wi-Fi ad-hoc, Bluetooth and Wi-Fi Direct, or are not available in current devices, or unwanted require user interaction to establish connectivity. To overcome these shortcomings, we propose an architecture that uses the Wi-Fi infrastructure mode in order to promote communication between devices, allowing the transparent exchange without user interaction content. Two applications that employ the use of this architecture are presented. The first, from personal devices, proved to be scalable in tests with up to nine devices. The second, vehicles, proved to be feasible when applied in scenarios with low speed, generating a low packet loss and high transmission rates.

Categories and Subject Descriptors

C.2.1 [Network Architecture and Design]: Wireless Communication

General Terms

Design

Keywords

Architecture, Opportunistic Communications.

1. INTRODUÇÃO

Hoje em dia, as redes de comunicação sem fio tornaram-se essenciais na sociedade da informação. As pessoas podem se conectar a redes de dados em qualquer lugar, por meio de diferentes dispositivos e tecnologias de comunicação. Dispositivos móveis como celulares, smartphones e tablets vêm se popularizando e evoluindo muito nos últimos anos, tornando a interação do usuário com o dispositivo uma experiência menos virtual e mais realista. A integração de diversos recursos de comunicação (Bluetooth, Wi-Fi Direct, 3G e LTE) abriu novas possibilidades de redes [1, 2].

Redes oportunistas são uma das evoluções mais interessantes de MANETs (*Mobile Ad hoc Networks*) [3]. Nas redes oportunistas, dispositivos móveis podem se comunicar

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SBSI '2015, May 26–29, 2015, Goiânia, Goiás, Brazil
Copyright SBC 2015.

uns com os outros, mesmo se uma rota para conectá-los não existe [4]. Redes oportunistas podem trazer benefícios para redes baseadas em infraestrutura sem fio (3G e LTE), como na utilização eficiente do espectro radio elétrico disponível [5] ou em descarregar o tráfego celular [6]. Além disso, ela é ótima para lidar com falhas de infraestrutura de comunicação, parciais ou totais causadas por desastres naturais [7], censura do governo [8] ou por até uma interrupção de Internet e telemóvel [9].

Redes oportunistas tem sido um tema de pesquisa há muitos anos, produzindo arquiteturas de comunicação [10, 11] e aplicações [12, 13]. No entanto, rede oportunista entre dispositivos móveis depende das capacidades de tais dispositivos para estabelecer comunicação ad-hoc entre si. Embora as tecnologias de interface sem fio, Wi-Fi ad-hoc, Wi-Fi Direct e Bluetooth oferecem tais capacidades em teoria, limitações da especificação do protocolo, chipsets e sistemas operacionais em dispositivos móveis tornam essas tecnologias, em grande parte inútil na prática.

Dispositivos móveis atuais não suportam Wi-Fi Ad-Hoc [14], a não ser com firmwares customizados que exigem acesso root ao sistema operacional do dispositivo, como foi feito em [15]. Bluetooth é limitado em termos de alcance de comunicação e largura de banda, bem como descoberta sem interação humana [17]. Além disso, o Bluetooth leva muito tempo para pareamento e muitas das tentativas de pareamento não são feitas com sucesso [16]. Comunicação via Wi-Fi Direct é outra opção, mas requer pareamento como a inserção de um PIN obrigatório, podendo demorar até 2 minutos para formação de grupo e ainda requer interação do usuário [18].

Essas características das tecnologias citadas, impedem na prática o desenvolvimento de aplicações que necessitam de comunicação transparente, isto é, a formação da rede de comunicação e a troca de conteúdos sem a necessidade de interação do usuário com o dispositivo. Para superar essas deficiências é proposto uma arquitetura baseada na tecnologia IEEE 802.11 WLAN modo infraestrutura, que possibilita que os dispositivos tornem-se pontos de acesso e os demais dispositivos próximos se conectem ao ponto de acesso, formando a rede de comunicação sem exigir pareamento. Essas características tornam a arquitetura uma abordagem prática para aplicações de redes oportunistas.

Como prova de conceito foram criadas duas aplicações: (i) *Crowd Wi-Fi* permite a troca de informações entre dispositivos móveis pessoais de forma transparente; (ii) *Black Box* é um protótipo em desenvolvimento que permite a entrega de vídeos tolerante a atrasos do veículo para um servidor.

Resultados mostraram que a aplicação *Crowd Wi-Fi* conseguiu compartilhar conteúdos de um dispositivo para 8 dispositivos simultaneamente de forma transparente, mostrando-se escalável com uma taxa média de transmissão de 17Mbps. E na aplicação *Black Box*, resultados mostraram que a mesma conseguiu em uma distância de até 30 metros entre veículos, ter uma taxa de transferência média de 500kbps e uma taxa de perda de pacotes de 25% e atraso médio de 30ms.

O restante do artigo está organizado da seguinte forma: na Seção 2 são apresentados os trabalhos relacionados. Na Seção 3 é apresentado a visão geral da arquitetura. Na Seção 4 é apresentado a aplicação *Crowd Wi-Fi*. Na Seção 5 é apresentado a aplicação *Black Box*. Na Seção 6 são apresentados os cenários e métricas avaliadas. Na Seção 7 são apresentados os resultados. Finalmente na Seção 8 são apresentadas as conclusões.

2. TRABALHOS RELACIONADOS

Atualmente são poucos os trabalhos que se concentram em uma estrutura de rede similar. Na maioria dos trabalhos publicados, houve comparação entre a forma de rede de infraestrutura com as redes ad hoc [19, 20]. Enquanto que outros trabalhos, concentram-se no lado do protocolo de roteamento da estrutura de uma rede de infraestrutura [21], ou em melhorar a precisão das decisões de um dispositivo estar em modo cliente ou ponto de acesso, tendo em conta também as frequências e tráfego de rede [22], ou em discutir uma forma de explorar os pontos de acesso fixos [23].

O trabalho MA-Fi [19] foca sobre considerações de mobilidade para os pontos de acesso e manutenção da conectividade multihop. Em contraste, Wi-Fi-Opp [20] visa comunicações oportunistas, onde é sugerido vários modos de operação: nós móveis analisar pontos de acesso no ambiente e se associar com um ponto por algum tempo, opcionalmente, se revezando entre pontos de acessos. Se nenhum ponto de acesso for encontrado, o dispositivo se torna em si um ponto de acesso por algum tempo para facilitar a comunicação para outros nós. Em seu estudo, os autores investigam o impacto de vários parâmetros (por exemplo, o tempo de baliza, intervalo de varredura, tempo de conexão) por meio de simulações e comparam o consumo de energia com Wi-Fi Ad Hoc, mas eles não analisam o tempo de formação de uma topologia de rede, nem mesmo como a rede se comporta com tráfego de conteúdos em aplicações distribuídas.

Enquanto que em Wi-Fi-Opp é apresentado uma aplicação e feito comparações com o Wi-Fi Ad Hoc, em nosso trabalho é feito uma extensão de Wi-Fi-Opp, onde é proposto uma arquitetura personalizada que serve de modelo para o desenvolvimento de aplicações oportunistas. A arquitetura proposta é baseada na funcionalidade infraestrutura WLAN, que estabelece a comunicação semelhante ao Wi-Fi Direct, porém sem necessitar de pareamento. Além disso, descrevemos aplicações em dois ambientes diferentes (smartphones e veículos) que emprega a arquitetura, e avaliamos o impacto do compartilhamento de conteúdos entre dispositivos nesse tipo de estrutura de rede.

3. VISÃO GERAL DA ARQUITETURA

A arquitetura incorpora uma coleção de dispositivos que, juntos, possibilitam a formação e gerenciamento da rede de compartilhamento de conteúdos. A Figura 1 ilustra os vários componentes da arquitetura. Ela é dividida em quatro módulos. Cada módulo tem sua particularidade e uma função especial. Nesta arquitetura, é definido que um dispositivo ponto de acesso é um *Líder*, e os dispositivos que se conectarem a ele são *Clientes*.

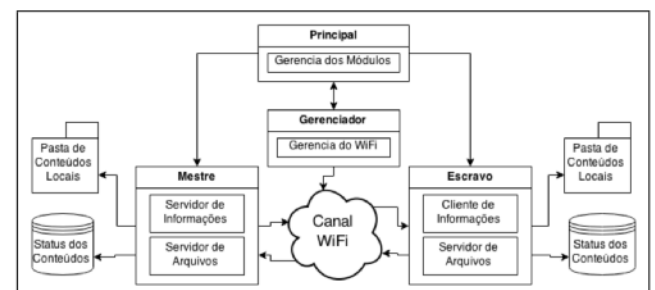


Figura 1: Modelo Esquemático da Arquitetura.

O módulo *Principal* é o primeiro módulo a ser inicializado na arquitetura. Ele é responsável por executar e gerenciar os três demais módulos. Sua primeira operação é executar o módulo *Gerenciador*.

O módulo *Gerenciador* pode executar dois tipos de operações diferentes. A primeira é o escaneamento de pontos de acesso sem fio e o estabelecimento de conexão aos pontos. Caso a rede seja estabelecida, o módulo *Escravo* é executado pelo módulo *Principal*. A segunda operação é a configuração do ponto de acesso sem fio no dispositivo. Com o ponto de acesso configurado, o módulo *Mestre* é executado pelo módulo *Principal*.

O módulo *Mestre* é executado apenas em um dispositivo *Líder*, e este tem informações sobre todos os dispositivos conectados a ele. Por este motivo, este módulo deve disponibilizar informações sobre o estado da rede através uma servidor de informações. Já o módulo *Escravo* é executado apenas em um dispositivo *Cliente* que está conectado ao *Líder*. O módulo *Escravo* requisita as informações da rede de comunicação ao módulo *Mestre* do dispositivo *Líder*.

Os conteúdos trocados entre os dispositivos conectados a rede de informações são transmitidos através de um servidor de conteúdo que fica nos módulos *Mestre* e *Escravo*. Este servidor deve ser multitarefa de forma a aceitar diversas requisições simultâneas de dispositivos *Clientes* ou *Líder*.

3.1 Estados de Execução

Para a rede de comunicação ser formada, um dos dispositivos deverá ser o ponto de acesso e os demais dispositivos deveram ser clientes conectados ao ponto de acesso. O módulo *Gerenciador* trata de definir se um dispositivo será um ponto de acesso ou um cliente. Para isso, o módulo *Gerenciador* alterna em dois modos de operações diferentes, sendo um de escaneamento de redes, e outro de se tornar um ponto de acesso (Figura 2).

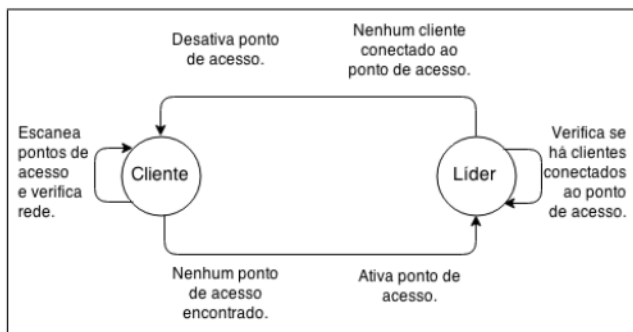


Figura 2: Estados de execução.

Dessa forma, um dispositivo *Líder* (ponto de acesso) tem os módulos *Principal*, *Gerenciador* e *Mestre* em execução. Já um dispositivo *Cliente* (cliente conectado ao ponto de acesso) tem os módulos *Principal*, *Gerenciador* e *Escravo*. Sendo que um dispositivo pode alternar sua execução conforme ocorra alterações na rede. Em caso de um dispositivo *Líder* inesperadamente terminar sua execução ou alguma falha ocorrer, outro dispositivo *Cliente* pode tomar seu posto.

3.2 Gerenciamento das Informações da Rede

Em uma infraestrutura de rede *WLAN* (*Wireless Local Area Network*), o dispositivo que é o ponto de acesso tem

todas as informações referentes a rede. Apesar de um cliente estar conectado a um ponto de acesso, ele desconhece quais os outros dispositivos atuais na rede, a não ser o dispositivo Ponto de Acesso que sempre fica com o endereço inicial da rede.

Para que todos os Clientes possam se contactar pela rede, o módulo Mestre do Líder deve fornecer aos Clientes uma lista de endereços *IP* (*Internet Protocol*) dos dispositivos conectados ao Mestre. Essa lista deve ser requisitada pelo Cliente de tempo em tempo, isso por que um dispositivo pode sair da rede e um endereço inválido pode ficar armazenado em um Cliente. Assim como uma lista de endereços *IP*, o dispositivo Ponto de Acesso pode também disponibilizar outras informações de acordo com a necessidade de cada implementação das aplicações.

Para que os arquivos compartilhados nessa rede sejam identificados por cada dispositivo, o nome do arquivo é identificado pela concatenação do endereço *MAC* (*Media Access Control*) mais o nome do arquivo. Dessa forma, além do arquivo ter uma identificação única, os dispositivos que receberem o nome do arquivo, saberão qual o dispositivo que disponibilizou o conteúdo na rede.

3.3 Segurança

Para o dispositivo se tornar um ponto de acesso, ele deve definir o identificador *SSID* (*Service Set Identifier*) e a senha *PSK* (*Pre-Shared Key*) da rede. Essas informações devem ser configuradas na aplicação de forma que todos os dispositivos possam automaticamente se conectarem a um ponto de acesso. Dessa forma, dispositivos clientes poderão escanear e se conectarem automaticamente as redes encontradas comparando com o *SSID* configurado na aplicação.

O propósito da arquitetura é possibilitar comunicação transparente entre os dispositivos. Deste forma, a arquitetura não tem uma forte ênfase na segurança da comunicação entre dispositivos, como é feito no Bluetooth e Wi-Fi Direct com a segurança de pareamento. Assim, para manter a comunicação transparente, é definido que a segurança das informações deve ser feita pelas aplicações, por exemplo, utilizando *TLS* (*Transport Layer Security*).

4. APLICAÇÃO CROWD WI-FI

Esta aplicação tem como objetivo compartilhar conteúdos de uma pasta do dispositivo do usuário entre vários smartphones de forma transparente e escalável ao usuário final. Os principais exemplos de utilização dessa aplicação, seria em eventos onde há uma pequena concentração de pessoas com smartphones, como em um restaurante, museu ou em uma sala de aula. Ela foi desenvolvida utilizando o sistema operacional Android 4.1.

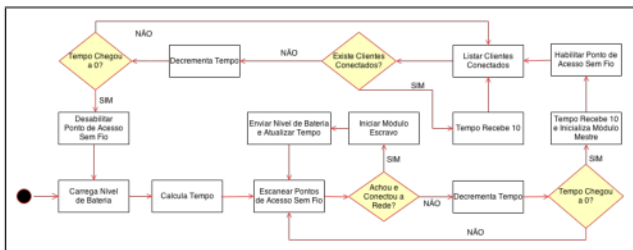
Para conseguir compartilhar os conteúdos de forma transparente, a aplicação incorpora os quatro módulos da Arquitetura definida neste trabalho. Os quatro módulos (ver Tabela 1) rodam em segundo plano, onde os mesmos permanecem em execução mesmo que outra tarefa entre em primeiro plano no smartphone.

O módulo *Principal* é a primeira e única *Activity* da aplicação, ela cuida da comunicação entre os módulos e o gerenciamento da interface gráfica. O módulo *Gerenciador* realiza o gerenciamento da interface Wi-Fi do dispositivo. Tanto para escanear as redes sem fio, quanto tornar o dispositivo um ponto de acesso sem fio, é utilizado a classe *WifiManager* do sistema Android.

Tabela 1: Módulos da aplicação *Crowd Wi-Fi*.

Módulo	Tipo	Operações
<i>Principal</i>	Activity	Executa e gerencia os demais módulos. Cuida da interação do usuário com a aplicação.
<i>Escravo</i>	Async-Task	Requisita ao <i>Mestre</i> informações referentes a rede. Disponibiliza e solicita arquivos.
<i>Mestre</i>	Async-Task	Disponibiliza informações de estado da rede de comunicação. Disponibiliza e solicita arquivos.
<i>Gerenciador</i>	Service	Gerencia a interface Wi-Fi e define se o dispositivo será <i>Líder</i> ou <i>Cliente</i> se comunicando com o <i>Principal</i> .

Nesta aplicação é utilizado o nível de bateria do dispositivo de forma a auxiliar o módulo *Gerenciador* a definir um tempo de pesquisa de redes sem fio. É definido que quanto menor o nível de bateria do dispositivo, maior é o seu tempo de escaneamento de redes sem fio, e quanto maior o nível de bateria, menor é o tempo de escaneamento e maior possibilidade de se tornar um ponto de acesso sem fio. O nível de bateria do dispositivo não só é utilizado para definir o tempo de escaneamento, mas também quando a rede de comunicação já está estabelecida. Quando a rede está formada, o módulo *Escravo* trata de enviar o nível atual de bateria do dispositivo para o módulo *Mestre* do dispositivo *Líder* de forma que o *Líder* auxilie os dispositivos, quem se tornará o próximo ponto de acesso sem fio, caso a rede seja destruída (Figura 3).


Figura 3: Diagrama de Atividades do módulo *Gerenciador* da Aplicação *Crowd Wi-Fi*.

Caso o dispositivo se torne um *Líder*, o módulo *Mestre* é executado. Este módulo é um *AsyncTask* que é disparado pelo módulo *Principal*. Ele é um servidor *multithreading* que possibilita conexões simultâneas dos módulos *Escravo* dos dispositivos *Clientes*. O módulo *Mestre* trata três tipos de requisições.

Quando um dispositivo *Cliente* disponibiliza um arquivo na rede, o módulo *Escravo* obtém o endereço do arquivo, e realiza uma requisição do tipo PUT mais o nome de identificação do arquivo e envia esses dados ao dispositivo *Líder*. O módulo *Mestre* do dispositivo *Líder* recebe a requisição do *Cliente* e atualiza sua lista local de arquivos disponibilizados na rede, e verifica se este novo arquivo disponibilizado na rede existe na sua pasta local, caso não exista, o módulo *Mestre* dispara uma tarefa que cuida de requisitar ao módulo

Escravo do *Cliente* o arquivo disponibilizado por ele.

Quando não há arquivos para serem disponibilizados, o módulo *Escravo* dos dispositivos *Clientes*, executa outras duas requisições ao dispositivo *Líder*. A primeira, requisição LIST, significa que o módulo *Mestre* do dispositivo *Líder* deverá retornar respectivamente, uma lista com todos os arquivos disponibilizados pelos dispositivos da rede, outra lista com todos os endereços IP dos dispositivos da rede, e por fim, uma lista os níveis de bateria de todos os *Clientes* conectados a rede. O módulo *Mestre* do dispositivo *Líder* responde essas informações ao módulo *Escravo* do dispositivo *Cliente* requisitante. O módulo *Escravo* do dispositivo *Cliente* valida a lista de arquivos disponibilizados na rede, e para cada arquivo não existente no dispositivo é disparado uma tarefa que cuida de solicitar o arquivo a todos os dispositivos conectados a rede utilizando a lista de IPs.

Após a requisição LIST, o módulo *Escravo* do dispositivo *Cliente*, realiza uma terceira requisição, a requisição AP. Nesta terceira requisição, o dispositivo *Cliente* envia uma mensagem contendo a identificação do dispositivo, seu endereço MAC, e o nível atual da bateria do dispositivo. O módulo *Mestre* do dispositivo *Líder* recebe esta requisição, e adiciona esta informação caso não exista em uma lista local com vários estados de bateria dos dispositivos *Clientes* conectados a rede.

Os módulos *Mestre* e *Escravo* também são compostos por um servidor de arquivos. O servidor aceita uma gama de conexões e para cada uma, verifica se o arquivo requisitado existe, caso exista, transmite uma mensagem ao dispositivo solicitante avisando que o arquivo existe. O dispositivo solicitante verifica a mensagem, se a mensagem é de arquivo existente, a transmissão do arquivo é inicializada, caso contrário, a conexão é fechada, e o dispositivo solicitante abre uma nova conexão com outro dispositivo, utilizando as informações da lista de Ips.

Ao iniciar a aplicação, a primeira tela exibida é a de seleção da pasta, conforme é ilustrado na Figura 4a, nesta tela o usuário pode selecionar a pasta de conteúdos que serão compartilhados. Após a seleção da pasta, a Figura 4b ilustra a tela de lista de dispositivos. Cada dispositivo tem uma pasta compartilhada, ao clicar em um conteúdo, este é verificado se já existe, caso não exista, ele é requisitado a outros dispositivos. Com o arquivo no dispositivo, o usuário deve escolher um serviço para sua execução (Figura 4c).


Figura 4: Imagens da Aplicação *Crowd Wi-Fi*.

5. APLICAÇÃO BLACK BOX

Esta aplicação foi desenvolvida para uma empresa de frota de caminhões, com o objetivo de realizar o gerenciamento de gravação de vídeos em veículos e a transmissão de vídeos entre veículos e infraestruturas com um atraso tolerante. Esta aplicação é um protótipo em desenvolvimento e os pontos de acessos e os clientes são estáticos, não seguindo completamente a estrutura de rede da arquitetura. Cada veículo é equipado de uma placa PandaBoardES e uma câmera que realiza o monitoramento do motorista. O principal objetivo desse sistema, é realizar a gravação de vídeos em partes e transmiti-los para um servidor da empresa de caminhões. Para conseguir compartilhar os conteúdos de forma transparente, a aplicação incorpora os quatro módulos da Arquitetura definida neste trabalho (Tabela 2).

Tabela 2: Módulos da Aplicação *Black Box*.

Módulo	Tipo	Operações
<i>Principal</i>	Shell-Script	É o primeiro a inicializar e gerenciar os demais módulos.
<i>Camera</i>	Python	Grava vídeos de acordo com um tempo definido e gerencia o espaço disponível em disco.
<i>Escravo</i>	Java	Requisita informações e transmite vídeos à infraestrutura.
<i>Mestre</i>	Java	Disponibiliza lista de vídeos relacionadas ao veículo requisitante. Recebe vídeos de veículos.
<i>Gerenciador</i>	Shell-Script	No veículo, ele escaneia e conecta a rede da infraestrutura. Na infraestrutura, configura o ponto de acesso sem fio.

O módulo *Principal* é o primeiro módulo a ser inicializado pela aplicação, ele é responsável por executar e gerenciar os cinco demais módulos da aplicação. Ele é executado de minuto em minuto no sistema, através da ferramenta *crontab*. Seu objetivo é monitorar se os demais módulos estão em execução. É utilizado a ferramenta *ps* para listar os processos do sistema e a ferramenta *grep* para cortar apenas as linhas referentes a um determinado módulo. Se não houver retorno quando pesquisado um processo de um módulo, o mesmo é inicializado.

5.1 Veículo

O veículo é apenas um dispositivo *Cliente*. Ele escaneia pontos de acesso sem fio e estabelece conexão aos pontos. O módulo *Principal* do veículo, executa e monitora os módulos *Camera*, *Escravo* e *Gerenciador*.

O módulo *Camera* realiza a gravação de vídeos da câmera instalada no veículo e gerencia o espaço disponível em disco. Ele é dividido em duas *Threads*, a primeira grava os vídeos de tempo em tempo em uma pasta. O tempo de intervalo dos vídeos é definido em um arquivo de configuração. O nome de identificação dos vídeos é definido através da data e hora atual do sistema. A segunda gerencia o espaço disponível em disco, para isso é utilizado a ferramenta *psutil*. O limite máximo de espaço utilizado em disco é definido em um arquivo de configuração. Caso o limite máximo exceda, o vídeo mais antigo é removido da pasta. Essa segunda etapa

é executada de minuto em minuto.

O módulo *Gerenciador* do veículo realiza uma sequência de operações, conforme ilustrado na Figura 5. Ele realiza o escaneamento de pontos de acesso sem fio utilizando a ferramenta *wpa supplicant*. Em um arquivo de configuração é definido o identificador (*SSID*), senha (*PSK*) e tipo de segurança *WPA (Wi-Fi Protected Access)* da rede. Todas as redes encontradas são comparadas com a carregada do arquivo de configuração. Caso uma rede seja encontrada, o módulo se conecta a rede e verifica se a conexão foi estabelecida com o *Líder*, para isso é executado um ping no Endereço *IP* da rede conectada. Caso o ping não tenha um retorno, nesse caso, o veículo não consegue se comunicar com o *Líder*, então é executado um outro comando, o *dhclient*. O *dhclient* é executado passando a interface Wi-Fi. O *dhclient* no momento de sua execução cuida de configurar a interface usando o protocolo DHCP e obter um endereço *IP* do *Líder*. Caso contrário, e o ping tenha um retorno, então o veículo verifica se o módulo *Escravo* já está executando, caso não esteja, o mesmo é executado e o *PID (Process Identifier)* do processo é armazenado, para que na próxima interação não seja executado novamente o módulo *Escravo*. Outra etapa desse módulo é o monitoramento do módulo *Escravo*, ele só é executado no momento em que se tem um retorno do ping, caso não tenha um retorno, e o *Escravo* estiver executando, o processo *Escravo* é finalizado através do seu *PID*.

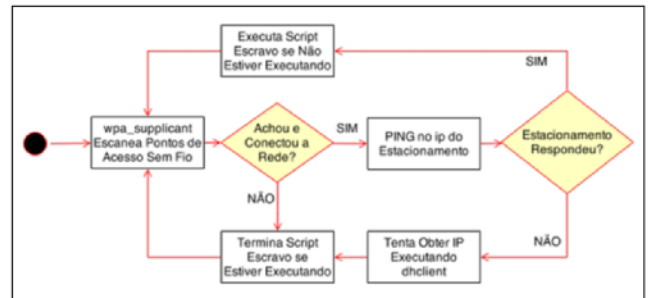


Figura 5: Diagrama de Atividades do módulo *Gerenciador* da Aplicação *Black Box*.

O módulo *Escravo* é um cliente implementado em Java, e este cuida da comunicação e gerenciamento dos vídeos locais dos veículos. Este módulo se comunica com o módulo *Mestre* da infraestrutura e requisita uma lista de vídeos referentes ao veículo. Ele recebe esta lista e transmite os vídeos existentes a infraestrutura.

5.2 Infraestrutura

A infraestrutura é apenas um dispositivo *Líder*, isto é, um ponto de acesso sem fio. Inicialmente é configurado na placa da infraestrutura um servidor *DHCP (Dynamic Host Configuration Protocol)*, o *isc-dhcp-server*. Este servidor gerencia os endereços *IP* dos veículos que estabelecem uma conexão com a placa da infraestrutura. Neste servidor é definido as faixas de endereços *IP* e a interface Wi-Fi utilizada.

O módulo *Principal* da infraestrutura, executa e monitora os módulos: *Mestre* e *Gerenciador*. Diferentemente do que ocorre no veículo, o módulo *Gerenciador* cuida apenas de configurar a o ponto de acesso sem fio na infraestrutura. Para isso ele utiliza a ferramenta *Hostapd* que carrega de um arquivo de configuração o identificador (*SSID*), senha (*PSK*) e tipo de segurança *WPA* da rede.



Figura 6: Comunicação entre os módulos *Escravo* e *Mestre*.

Uma sequência de trocas de mensagens é realizada entre os módulos *Mestre* e *Escravo* para a transmissão dos vídeos, conforme ilustrado na Figura 6. Na infraestrutura, o módulo *Mestre* é um servidor implementado na linguagem de programação Java, e este realiza o gerenciamento dos vídeos do veículo. Uma lista de vídeos é armazenada localmente e nela é armazenada informações sobre o status dos vídeos. A infraestrutura recebe a lista de vídeos de veículos a serem transmitidos do servidor da empresa de frotas. Quando um veículo estabelece conexão a rede Wi-Fi da infraestrutura, o módulo *Escravo* requisita ao módulo *Mestre* uma lista de vídeos a serem transmitidos. O módulo *Mestre* utiliza um identificador do veículo para filtrar os vídeos referentes ao mesmo. Então uma lista de vídeos é enviada ao veículo. O módulo *Escravo* recebe a lista de vídeos e transmite todos os vídeos existentes da lista requerida. O módulo *Mestre* da infraestrutura recebe os vídeos, e seus status são alterados, e estes vídeos recebidos são transmitidos ao servidor da empresa de frotas através da Internet.

6. CENÁRIOS E MÉTRICAS AVALIADAS

6.1 Aplicação Crowd Wi-Fi

A aplicação foi desenvolvida para um ambiente de restaurante ou museu, não considerando movimento dos dispositivos e interferências. Os experimentos foram realizados em laboratório, de maneira controlada, onde os dispositivos ficaram sobre uma mesa com uma média de 3 metros de distância entre os dispositivos. Foram utilizados 9 tablets, sendo que um tablet recebe as requisições, e os demais tablets requisitam. A aplicação foi alterada de modo que os dispositivos requisitassem um mesmo arquivo de 9MB. Os experimentos foram executados 30 vezes em cada dispositivo. O principal objetivo é avaliar como a rede se comporta quando um único dispositivo realiza a transmissão de um mesmo arquivo para vários outros dispositivos (quantidades diferentes) ao mesmo tempo. Essa avaliação é importante, pois através dela é possível dimensionar os limites e problemas relacionados a aplicação.

A avaliação foi realizada medindo o tempo para formação de uma topologia para a comunicação dos dispositivos, o tempo de atraso dos pacotes, a taxa de perda e taxa de transmissão. Em todos os experimentos as aplicações foram executadas todas ao mesmo tempo nos 9 tablets. No primeiro experimento, foi medido o tempo médio que todos os dispositivos gastaram para estabelecerem uma conexão ao *Líder* e formar a topologia de rede. Para o atraso da transmissão, foi medido o momento em que o pacote foi transmitido até o momento em que chegou ao receptor. Em relação

a taxa de perda de pacotes foi comparado o número de pacotes transmitidos com o número de pacotes efetivamente recebidos. Os dados foram obtidos realizando cálculos na própria aplicação desenvolvida, sendo considerado um intervalo de confiança de 95%. Foi utilizado quadros de controle *RTS/CTS* (*Request to Send / Clear to Send*) e potência de *TX* padrão do dispositivo de 20dBm.

6.2 Aplicação Black Box

Os experimentos foram realizados em uma avenida localizada na Universidade, composta de um trecho de 430 metros (Figura 7). Um veículo iniciou o deslocamento em uma das extremidades da avenida (ponto 2), mantendo as velocidades de 60 km/h, 50 km/h, 40 km/h, 30 km/h e 20 km/h. O veículo do ponto 1, atuou como a infraestrutura, permanecendo parado no meio da avenida. A distância entre os dois pontos é de 216 metros.



Figura 7: Vista aérea da região de experimentação. Ponto 1 é a infraestrutura fixa. Ponto 2 é o veículo móvel.

A avaliação da rede foi realizada medindo o tempo de atraso dos pacotes, a taxa de perda e taxa de transmissão. Para o atraso da transmissão, foi medido o momento em que o pacote foi transmitido até o momento em que chegou ao receptor. Em relação a taxa de perda de pacotes foi comparado o número de pacotes transmitidos com o número de pacotes efetivamente recebidos. Os dados foram obtidos utilizando o software *Wireshark*, que disparou pacotes com 512 bytes a uma taxa de transmissão de 2048 kbps. Quatro repetições foram realizadas para cada experimento. A posição geográfica do veículo foi registrada durante a realização dos experimentos. Foi utilizado quadros de controle *RTS/CTS* e potência de *TX* padrão da placa de 20dBm.

7. RESULTADOS

7.1 Aplicação Crowd Wi-Fi

No primeiro experimento, foi medido o tempo de associação entre os dispositivos. O tempo é obtido através do Log da aplicação Android. Como dito em seções anteriores, foram feitas 30 repetições deste experimento. Este experimento é importante para avaliar o impacto que a quantidade de dispositivo promove no tempo de associação dos dispositivos.

Na Figura 8a, pode se observar que quando há poucos dispositivos, o tempo de formação da topologia é consideravelmente grande, e sua faixa de erro também. Porém, quando a quantidade de dispositivos começa a aumentar o tempo de formação da topologia começa a diminuir e a taxa de erro também. Assim, o tempo de formação de topologia se comporta melhor em ambientes com maiores núme-

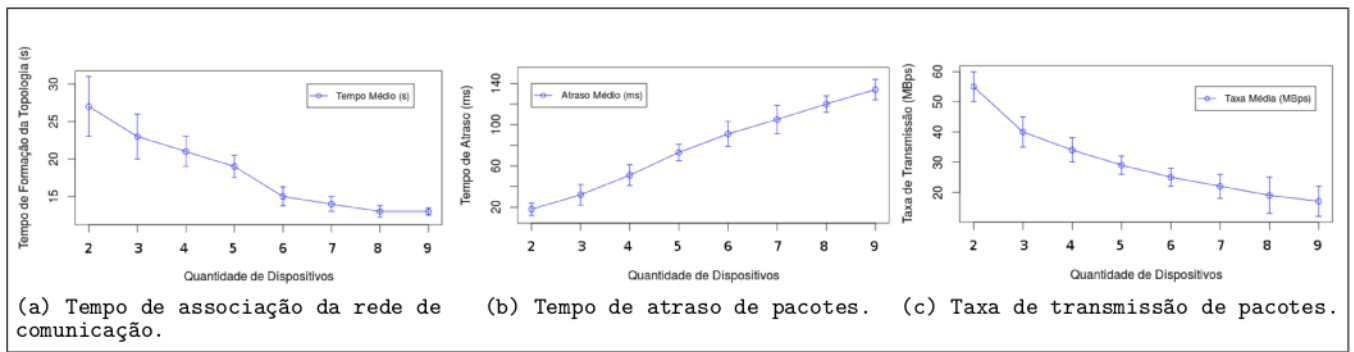


Figura 8: Resultados dos experimentos da aplicação *Crowd Wi-Fi*.

ros. Isso acontece por que quando há poucos dispositivos alguns dispositivos a probabilidade de dispositivos ficarem ao mesmo tempo escaneando ou se tornando um ponto de acesso é maior do que quando temos uma quantidade maior de dispositivos. Sendo assim, viável sua aplicação em ambientes com um número maior de dispositivos, como em um restaurante.

No segundo experimento, foi medido o tempo de atraso de pacotes entre os dispositivos da rede de comunicação. Todos os dispositivos armazenaram os tempos no momento de transmissão de pacotes e no recebimento, no final todos os tempos foram coletados e feito a média do atraso. Como dito em seções anteriores, foram feitas 30 repetições deste experimento. Este experimento é importante para avaliar o impacto que a quantidade de dispositivos promove no atraso da transmissão de pacotes.

Na Figura 8b, pode se observar que o tempo de atraso dos pacotes aumenta de acordo que a quantidade de dispositivos que recebem um arquivo aumenta. Isso ocorre por que o dispositivo que transmite o arquivo tem um trabalho maior com quantidades maiores de dispositivos, e assim o canal de comunicação do dispositivo fica mais ocupado com mais pacotes ao mesmo tempo para serem transmitidos e processados.

No terceiro experimento, foi medido a taxa de transmissão de pacotes entre os dispositivos da rede de comunicação. O mesmo processo do tempo de atraso de pacotes foi realizado, porém, neste caso, através dos tempos e do tamanho do arquivo foi possível calcular a taxa de transmissão dos pacotes.

Na Figura 8c, pode se observar que, assim como ocorre no tempo de atraso de pacotes, a quantidade de dispositivos

também influencia na taxa de transmissão. A taxa de transmissão de pacotes diminui de acordo que a quantidade de dispositivos que recebem um arquivo aumenta. Isso acontece por que o dispositivo servidor que está cuidando de transmitir o arquivo, tem um trabalho maior com quantidades maiores de dispositivos, e assim a banda de conexão desse dispositivo está mais ocupada com várias conexões simultâneas e mais pacotes para serem processados.

7.2 Aplicação Black Box

Os resultados foram extraídos de quatro repetições para cada experimento, no cenário apresentado na Figura 7. O intervalo de confiança considerado foi de 95%, mas não são representados no gráfico para facilitar a disposição das informações. Todos os três experimentos foram avaliados nas velocidades de 60 km/h, 50 km/h, 40 km/h, 30 km/h e 20 km/h. Nos gráficos apresentados, a distância negativa significa aproximação do veículo ao nó destino e a distância positiva representa seu afastamento.

No primeiro experimento foi avaliado a taxa de perda de pacotes. A Figura 9a apresenta a taxa de perdas. Os dados obtidos em diferentes velocidades mostram que a rede se comporta de maneira mais robusta em velocidades menores. Foi possível realizar a transmissão em um diâmetro de aproximadamente 85 metros. Quanto mais próximo o veículo está do nó receptor, menor é a perda de pacotes. Quando os nós estão a uma distância relativa de até 25 metros, a perda de pacotes ficou abaixo de 25%. A velocidade também influenciou na perda de pacotes, mas não tanto quanto a distância.

No segundo experimento foi avaliado o atraso na transmissão dos pacotes. A Figura 9b apresenta os atrasos. O

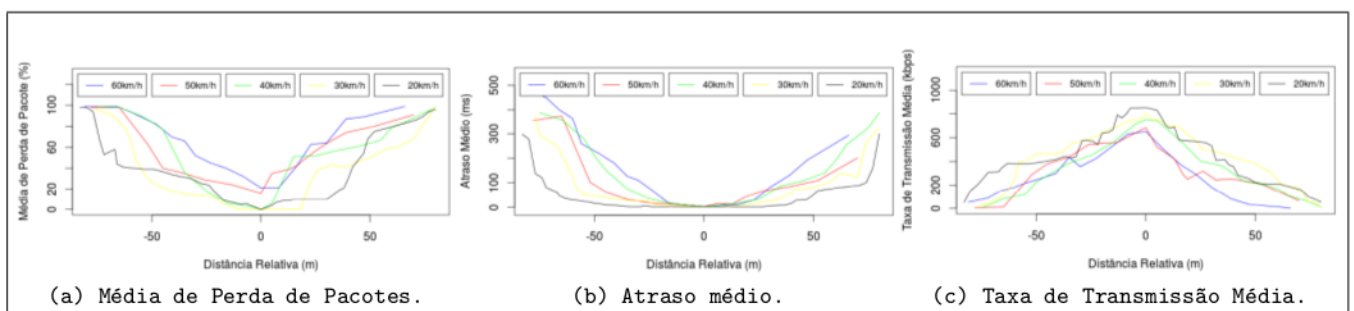


Figura 9: Resultados dos experimentos da aplicação *Black Box*.

atraso foi medido considerando apenas os pacotes efetivamente transmitidos. O atraso médio variou de forma significativa em relação a distância. Os valores obtidos quando os nós estavam em pontos distantes variaram bastante em relação ao atraso obtido quando os nós estavam próximos. Foi verificado que ao aumentar a velocidade, o atraso nas comunicações também sofre incremento.

No terceiro experimento foi avaliado a taxa de transmissão de dados. A Figura 9c apresenta as taxas obtidas nas comunicações. Os dados obtidos das cinco avaliações em diferentes velocidades mostram que nas comunicações, a taxa de transmissão média variou de forma significativa em relação a distância.

8. CONCLUSÃO

A arquitetura de compartilhamento de conteúdo proposta neste artigo, permitiu com sucesso a comunicação transparente nas duas aplicações desenvolvidas no Wi-Fi modo infraestrutura. A arquitetura possibilitou o compartilhamento de conteúdos entre dispositivos sem a necessidade de pareamento e intervenção do usuário final, mostrando ser uma solução viável para aplicações oportunistas de smartphones e veículos.

Os resultados mostraram que a aplicação *Crowd Wi-Fi* conseguiu ter um baixo tempo de associação dos dispositivos e ser escalável com até 9 dispositivos em uma rede de comunicação de dispositivos móveis pessoais. Sendo assim, viável sua aplicação em aplicações oportunistas de locais públicos ou eventos, como em um restaurante ou museu.

Para a aplicação *Black Box*, que ainda é um protótipo, os resultados mostraram que abaixo de 30 km/h e em uma distância máxima de até 30 metros da infraestrutura, os veículos conseguem se comunicar com uma alta taxa de transmissão e uma baixa perda de pacotes. Sendo assim, viável sua aplicação em empresas de frota de caminhões ou empresas de ônibus.

Como trabalhos futuros, além do aprimoramento das aplicações, principalmente da finalização do *Black Box* adicionando a alternância dos pontos de acessos e clientes, pretende-se estender os estudos, disponibilizando a aplicação *Crowd Wi-Fi* para um evento, e instalar a aplicação *Black Box* em uma empresa de frota de caminhões, com o propósito de avaliar mais profundamente o comportamento das aplicações nessa estrutura de rede criada pela arquitetura.

9. REFERÊNCIAS

- [1] N.D. Lane, E. Miluzzo, Hong Lu, T. Choudhury, e A. T. Campbell (2010). A survey of mobile phone sensing. In *Communications Magazine IEEE*, 48, 140–150.
- [2] R. K. Ganti, Fan Ye, e Hui Lei. (2011). Mobile crowdsensing: current state and future challenges. In *Communications Magazine IEEE*, 49, 32–39.
- [3] L. Pelusi, A. Passarella, M. Conti. (2006). Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. Em *Communications Magazine, IEEE*, 44, 134–141.
- [4] C. Boldrini, K. Lee, M. Onen, J. Ott, e E. Pagani. (2014). Opportunistic networks. Em *Computer Communications*, 48, 1–4.
- [5] V. Vukadinović, e G. Karlsson. (2010). Spectral efficiency of mobility-assisted podcasting in cellular networks. In *International Workshop on Mobile Opportunistic Networking*, 51–57.
- [6] B. Han, P. Hui, M. V. Marathe, e et al. (2010). Cellular traffic offloading through opportunistic communications: a case study. In *Chants*, 31–38.
- [7] M. Dekker, e C. Karsberg. (2013). Annual Incident Reports 2013. In *Technical Report October*, ENISA.
- [8] M. Helft, e D. Barboza. (2010). Google Shuts China Site in Dispute over Censorship. In *The New York Times*, March(22).
- [9] T. M. Chen. (2011). Governments and the executive 'internet kill switch'. In *IEEE Netw*, 25 (2), 2–3.
- [10] J. Ott, e J. Kangasharju. (2012). Opportunistic Content Sharing Applications. In *Proc. of the ACM MobiHoc NOM workshop*, 19–24.
- [11] M. Pitkänen, T. Kärkkäinen, J. Ott, e et al. (2012). SCAMPI: Service platform for soCial Aware Mobile and Pervasive computing. In *Proc. of the ACM workshop on Mobile Cloud Computing*, 503–508.
- [12] J. Ott, E. Hyttiä, P. Lassila, T. Vaegs, e J. Kangasharju, (2011). Floating Content: Information Sharing in Urban Areas. In *IEEE Percom*, 136–146.
- [13] A. Lindgren. (2011). Social networking in a disconnected network: fbDTN: facebook over DTN. In *Proc. of ACM MobiCom CHANTS workshop (demo)*, 69–70.
- [14] IEEE-SA (2007). IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.
- [15] Ó. R. Helgason, E. A. Yavuz, S. T. Kouyoumdjieva, e et al. (2010). A Mobile Peer-to-Peer System for Opportunistic Content-Centric Networking. In *Proc. of the ACM workshop on Networking*, 21–26.
- [16] A. K. Pietiläinen, e C. Diot. (2009). Experimenting with Opportunistic Networking. In *Proc. of the ACM MobiArch Workshop*, 28–36.
- [17] J.C. Haartsen. (2000). The Bluetooth radio system. In *Personal Communications, IEEE*, 28–36.
- [18] Wi-Fi Alliance, P2P Technical Group (2009). In *Wi-Fi Peer-to-Peer (P2P) Technical Specification v1.0*.
- [19] H. Wirtz, T. Heer, R. Backhaus, e K. Wehrle. (2011). Establishing mobile ad-hoc networks in 802.11 infrastructure mode. In *Proc. of the ACM workshop on Challenged networks*, 49–52.
- [20] S. Trifunovic, B. Distl, D. Schatzmann, e F. Legendre. (2011). Wi-Fi-Opp: ad-hoc-less opportunistic networking. In *Proc. of the ACM workshop on Challenged networks*, 37–42.
- [21] M.H. Sarshar, P. K. Hoong, e I.A. Abdurrazzaq. (2013). NodesJoins: A Framework for Tree-Based MANET in IEEE 802.11 Infrastructure Mode. In *IEEE Symposium of the Computers and Informatics*, 190–195.
- [22] D.J. Dubois, Y. Bando, K. Watanabe, e H. Holtzman. (2013). Lightweight Self-organizing Reconfiguration of Opportunistic Infrastructure-mode WiFi Networks. In *International Conference on Self-Adaptive and Self-Organizing Systems*, 247–256.
- [23] T. Kärkkäinen, M. Pitkänen, e J. Ott. (2013). Enabling Ad-hoc-Style Communication in Public WLAN Hot-Spots. In *ACM Mobile Computing and Communications Review*, 17, 4–13.