# Um Serviço SDN para Detecção e Solução de Problemas em Redes Domésticas

**Alisson R. Alves, Henrique D. Moura, Luis H. C. Reis, Julio C. T. Guimarães**
**Jonas R. A. Borges, Philippe S. Silva, Daniel F. Macedo, Marcos A. M. Vieira**

[1]Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte – MG – Brazil

{alissonralves, henriquemoura, luiscantelli}@dcc.ufmg.br

{julio.guimaraes, jonasrafael, phss, damacedo, mmviera}@dcc.ufmg.br

***Abstract.*** *The number of smart devices in home networks is rapidly increasing, making it more complex to manage problems. In addition, the lack of customer knowledge and the lack of tools to automatically diagnose and fix problems aggravate the problem. In this paper, we propose a network service, based on software defined networks, for the detection and automatic solution of problems in home networks. We evaluate the service in a prototype, considering throughput, jitter, elapsed time to fix the problem, among other metrics. The results show that our service can provide increase throughput by 7%, reduction in wireless transmission delays, and reduction of human intervention in troubleshooting.*

***Resumo.*** *As redes domésticas têm recebido quantidades cada vez maiores de dispositivos inteligentes. O resultado disso é um aumento na complexidade de gerenciamento de problemas nessas redes. Além disso, contribuem também a falta de conhecimento dos clientes e a falta de ferramentas que diagnostiquem e corrijam automaticamente tais problemas. Neste artigo, é proposto um serviço de rede para a detecção e a solução automática de problemas em redes domésticas, baseado em redes definidas por software. O serviço foi avaliado em um protótipo, considerando vazão, jitter, tempo para solução da falha, entre outras métricas. Os resultados comprovam que o serviço é capaz de proporcionar aumento na vazão em 7%, redução em atrasos de transmissão sem fio e redução da intervenção humana ao solucionar problemas.*

## 1. Introdução

As redes domésticas têm se tornado cada vez maiores em função da quantidade de dispositivos inteligentes conectados [Cisco 2015, Perera et al. 2014]. Dentre os dispositivos pessoais inseridos nesse tipo de rede estão *smartphones, tablets*, *notebooks, smart TVs* e dispositivos de Internet das Coisas (IoT). Resultado disso, é a densificação dessas redes. Além disso, novos serviços como *streaming*, armazenamento de dados na nuvem, dentre outros, têm se tornado bastante populares entre seus usuários [Bouchet et al. 2014]. A partir disso, surge a necessidade de maior confiabilidade e qualidade de serviços (QoS), o que torna o gerenciamento de redes domésticas ainda mais complexo.

Outros fatores contribuem no aumento da complexidade de gerenciamento de redes domésticas. Como exemplo, tem-se o volume elevado de conexões móveis e a diversidade de padrões de comunicação entre dispositivos. O aumento no tráfego em redes,

cabeadas ou sem fio, bem como a diversidade de problemas ou falhas que podem ocorrer na rede, também contribuem para maior complexidade [Yiakoumis et al. 2011]. Em geral, quando ocorre um problema, o usuário de redes domésticas não é capaz de inspecioná-lo ou solucioná-lo. Muitas vezes tais usuários são os responsáveis por configurá-las [Dong and Dulay 2011]. Essa dificuldade é agravada pela falta de ferramental que apoie a identificação precisa dos motivos que geraram um problema, assim como a solução automática deste [Fratczak et al. 2013]. Neste trabalho, considera-se problemas como baixo desempenho, falhas, ataques à rede ou eventos que possam ser representados por uma condição que tenha uma ou mais métricas de rede associadas.

O diagnóstico e a correção automática de problemas em redes domésticas são benéficos tanto para os usuários quanto para as provedoras de acesso. Para os usuários, os benefícios são uma rede mais estável e confiável, aumentando a sua satisfação. Para as provedoras, a automatização permite uma redução no seu custo operacional, pois exige uma demanda menor de serviços de manutenção, que normalmente não são faturados aos clientes. Além disso, ferramentas que facilitam o diagnóstico da causa do problema permitem um menor tempo para a sua recuperação, aumentando a produtividade da equipe de operação. Ainda, do ponto de vista de regulação, uma rede mais robusta é importante para alcançar as metas de qualidade de rede e disponibilidade que são definidos pelos órgãos reguladores, que muitas vezes podem aplicar multas em caso de descumprimento dessas metas.

Apesar de diversos trabalhos na literatura tratarem de problemas e de falhas em redes domésticas [Gheorghe et al. 2015, Kim et al. 2014b, Biswas et al. 2015, DiCioccio et al. 2012], poucos utilizam o paradigma de redes definidas por software para lidar também com redes sem fio. Em complemento, poucos solucionam automaticamente problemas na rede ou permitem adicionar novas regras de monitoramento e diagnosticar novos tipos de dispositivos de rede. Além disso, alguns trabalhos apresentam limitações em termos de flexibilidade por dependerem de plataformas proprietárias em suas soluções.

Desta maneira, neste artigo é proposto o *HomeNetRescue* (HNR), um serviço para o gerenciamento autônomo de redes domésticas voltado para a detecção, o diagnóstico e a solução automática ou minimização de problemas, que também pode atuar em aspectos de desempenho, baseado em redes definidas por software (SDN) [Guedes et al. 2012, Macedo et al. 2015]. Uma contribuição do artigo encontra-se na modelagem do protótipo do serviço para o gerenciamento autônomo que utiliza o paradigma SDN em redes sem fio. Além disso, a utilização do *HNR* é capaz de proporcionar benefícios como redução da intervenção humana ao solucionar falhas, maior confiabilidade de rede, aumento na vazão em 7%, redução em atrasos de transmissão sem fio e melhor qualidade de experiência.

As seções deste artigo estão organizadas como segue. Na Seção 2, os trabalhos relacionados são discutidos. Na Seção 3, é realizada a descrição do *HomeNetRescue* juntamente com a arquitetura Ethanol. Na Seção 4, as avaliações e os resultados obtidos são descritos. Por fim, na Seção 5, são discutidas as conclusões e os trabalhos futuros.

## 2. Trabalhos Relacionados

A literatura apresenta diversas ferramentas e plataformas para o diagnóstico e a solução de falhas. Tais ferramentas se diferem nos tipos de redes e equipamentos que suportam, no escopo de problemas tratadas, nas abordagens (distribuídas ou centralizadas),

se somente detectam ou também solucionam falhas, entre outras características. Deste modo, na Tabela 1, apresentamos uma comparação entre nossa proposta e outras propostas da literatura. Posteriormente, descrevemos em síntese cada trabalho e ao final da seção, discutimos a respeito das principais diferenças. Na Tabela 1, as colunas identificam as seguintes propriedades: soluções focadas em redes domésticas (RD); uso de técnicas para solucionar falhas (*troubleshooting*) (TS); emprego do paradigma SDN (SDN); permitem o gerenciamento local de rede para a detecção de falhas (GL); fornecem recursos para a detecção automática de falhas (DA); suportam redes ethernet (ET); suportam redes wireless (WI); identificam falhas nas estações dos usuários (FE); e possibilitam a adição de novas regras de monitoramento e novos tipos de dispositivos, ou seja, extensibilidade (EX). A seguir apresentamos as propostas mais relevantes.

**Tabela 1. Comparação entre trabalhos na literatura.**

| Referência | RD | TS | SDN | GL | DA | ET | WI | FE | EX |
|---|---|---|---|---|---|---|---|---|---|
| [Kim et al. 2014b] | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ |
| [Biswas et al. 2015] | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ |
| [DiCioccio et al. 2012] | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ |
| [Gheorghe et al. 2015] | ✗ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ |
| [Sundaresan et al. 2013] | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ |
| [Kim et al. 2014a] | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ |
| **Serviço proposto** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

*DYSWIS* é *framework* P2P colaborativo, com uma arquitetura centralizada, para detecção e diagnóstico automático de falhas em redes de usuários finais, através do monitoramento de pacotes e relatórios de falhas de aplicações [Kim et al. 2014b]. As regras para a detecção de falhas são baseadas em consultas e sondagens distribuídas e recorrem, ainda, à colaboração de usuários para distinção de falhas na rede, bem como a identificação do ponto de origem da falha. Outra arquitetura, a *Meraki* [Biswas et al. 2015] da Cisco, realiza o gerenciamento de redes empregando um sistema em nuvem. A arquitetura fornece configuração centralizada, monitoramento e ferramentas de resolução de problemas em redes cabeadas e sem fio. É uma arquitetura *backend* composta de APs, comutadores e *firewalls* em residências ou empresas, que monitoram métricas de redes e as repassam a um sistema de gerenciamento central.

O *HomeNet Profiler* é uma aplicação cliente-servidor executada em estações clientes [DiCioccio et al. 2012]. Nesta proposta o usuário deve executar uma aplicação e, posteriormente, enviar os dados coletados a um servidor para análise. A aplicação monitora redes domésticas cabeadas e sem fio. As informações mensuradas incluem configurações de rede, desempenho, dispositivos ativos e serviços em execução, realizadas via protocolos ZeroConf [1] e UPnP [2]. Já *SDN-RADAR* consiste em uma abordagem distribuída para o monitoramento da infraestrutura de rede e a localização de problemas ou falhas de desempenho em redes de grande porte [Gheorghe et al. 2015]. Recursos SDN são utilizados no processo de identificação de enlaces de baixo desempenho mais prováveis em rede cabeadas. A ferramenta auxilia os administradores de rede a entenderem prováveis falhas de enlaces de rede, bem como a monitorar e depurar as falhas que afetam os serviços de entrega nas redes dos usuários.

[1] http://www.zeroconf.org/

[2] https://tools.ietf.org/html/rfc6970

*Where's The Fault?* é uma ferramenta executada em roteadores domésticos para detectar problemas de desempenho em redes domésticas cabeadas e sem fio [Sundaresan et al. 2013]. Informações de *timing* e *buffering* de rede são obtidas pelo monitoramento passivo do tráfego dos roteadores. Por fim, *Why is my Wi-Fi Slow?* (*Wi-Slow*) diagnostica o desempenho do sinal em redes Wi-Fi (abrangendo redes domésticas), utilizando sondagens em nível de usuário [Kim et al. 2014a]. Esta ferramenta auxilia na localização física e na identificação de causas que impactam o desempenho sem fio da rede via análise da perda de pacotes e do número de ACKs recebidos na rede.

Dentre os trabalhos citados, alguns propõem sistemas colaborativos para a solução de falhas, o que pode acarretar em problemas de privacidade em relação a informações de usuários da rede. Neste sentido, nossa proposta não armazena ou exibe dados a terceiros, respeitando a privacidade e a segurança dos dados dos usuários. Alguns dos trabalhos mencionados são pouco flexíveis, pois dependem de tecnologias proprietárias, consequentemente a adição de parâmetros de monitoramento e novos tipos de serviços ficam limitados aos fabricante dos componentes adotados nas redes. Por sua vez, em nossa solução, o paradigma SDN em conjunto com plataformas Linux embarcadas possibilita a implementação de recursos de controle da rede de forma mais flexível.

Nossa proposta também apresenta um diferencial ao analisar as falhas em todo o escopo de uma rede doméstica (APs, roteadores, estações). O *HomeNetRescue* pode solucionar ou minimizar automaticamente as falhas detectadas, diferentemente da maioria das abordagens apresentadas. Embora o *HomeNetRescue* apresente variadas funcionalidades, a solução herda problemas característicos de arquiteturas centralizadas. Todavia, devido a técnicas de alta disponibilidade para controladores serem um problema resolvido na literatura, o controlador não é o principal ponto de falha.

A visão global do controlador simplifica o gerenciamento da rede, possibilita obter soluções mais eficientes e controlar a rede com um grão mais fino, algo primordial para o processo de identificação e solução de falhas. Através dessa abordagem o serviço pode monitorar todos os componentes da rede em busca do seu funcionamento adequado. Ainda, tal visão possibilita às provedoras identificarem falhas nos dispositivos dos clientes, algo que atualmente depende de visita de técnicos ao domicílio. O resultado disto são reduções de custos para as provedoras (por não precisarem alocar profissionais em casos desnecessários) e para os usuários (dispensando a contratação de serviços adicionais de suporte). Outro benefício do serviço consiste em proporcionar maior índice de satisfação do cliente (QoE), pois com ele os problemas tornam-se passíveis de serem diagnosticados e resolvidos em menor tempo e automaticamente.

## 3. O Serviço *HomeNetRescue*

Nessa seção, é proposta e apresentada a arquitetura do *HomeNetRescue* (HNR), um serviço para o gerenciamento autônomo de redes domésticas voltado para a detecção, o diagnóstico e a solução automática ou minimização de falhas. Adicionalmente, o serviço também pode atuar em aspectos de desempenho. O HomeNetRescue baseia-se no paradigma SDN para o gerenciamento de redes sem fio e é composto por uma arquitetura modular e expansível, podendo ser empregado na resolução de diversos problemas desses ambientes. Destaca-se que as plataformas SDN são mais flexíveis do que as não SDN e isto permite ao HNR o controle com um grão mais fino no processo de gerenci-

amento. Além disso, SDN para o gerenciamento de redes sem fio é um tema em ampla investigação pela comunidade científica [Guedes et al. 2012].

O *HomeNetRescue* foi desenvolvido para ser executado sob demanda ou de forma agendada. Ainda, foi concebido para poder ser administrado por provedoras de acesso à internet, podendo ser fornecido aos seus clientes de forma gratuita ou paga. O *HomeNetRescue* pode ser executado a partir de *call centers*, quando necessário realizar diagnósticos ou soluções de problemas nas redes, mediante chamadas de suporte, ou ser executado programadamente (permanecendo em execução automática).

### 3.1. Arquitetura

A arquitetura do *HomeNetRescue* é composta por planos, camadas e módulos. Em seu planejamento foi considerado tratar eventos de falhas ou problemas a partir da camada de enlace até a camada de aplicação do modelo TCP/IP. Nesta, são apresentados os módulos de software juntamente com suas respectivas funcionalidades. Destaca-se também os componentes de hardware de uma rede doméstica. Considera-se uma rede doméstica composta por APs, roteadores IP, *switches* e estações clientes com e sem fio.

A arquitetura foi dividida em três planos, conforme apresentado na Figura 1. Estes planos são: $i)$ **plano de gerenciamento de problemas**: plano extensível onde são executados os algoritmos de detecção e gerenciamento de problemas. $ii)$ **plano de controle**: corresponde a um software (controlador) que pode ser executado de forma distribuída ou centralizada em um ou mais componentes da rede (ex: desktops, roteadores, entre outros); e $iii)$ **plano de dados**: responsável pelas funções de encaminhamento ou roteamento realizadas nos dispositivos de rede a serem gerenciados pelo serviço. A seguir são descritos mais detalhes sobre cada plano.

#### 3.1.1. Plano de Gerenciamento de Problemas

Este plano permite ao administrador da rede configurar as regras, as políticas e os parâmetros de gerenciamento do *HomeNetRescue*. É composto por uma ou mais aplicações. Estas aplicações são aplicações padrão do *HomeNetRescue* ou aplicações implementadas pelo gerenciador da rede. Elas fornecem as funcionalidades de detecção e solução de problemas. Cada aplicação possui dois módulos, diagnóstico e atuador.

O módulo **Diagnóstico** permite definir políticas de gerenciamento, registro de métricas a serem monitoradas, seu intervalo de monitoramento, prioridades de aplicações e alterações de configurações nos componentes de rede. Este módulo, assim como o módulo **Atuador**, é programável, porém não é vinculado ao módulo **Eventos**, com isso a implementação de inteligência das aplicações inseridas na arquitetura é flexibilizada. Este módulo comunica-se com o módulo monitor. Como demonstração dos módulos, uma aplicação para a detecção de problemas no sinal de transmissão sem fio é apresentada. Tal aplicação pode registrar o monitoramento, a cada segundo, das métricas *Signal-to-Noise-Ratio* (SNR) e porcentagem de pacotes perdidos em um roteador da rede. Com isso, a aplicação registra no plano de controle que, caso o SNR atinja o limiar de 10 $dbm$ ou a porcentagem de pacotes perdidos seja maior do que 10%, o plano de controle deve acionar o Atuador da aplicação com a prioridade predominante.

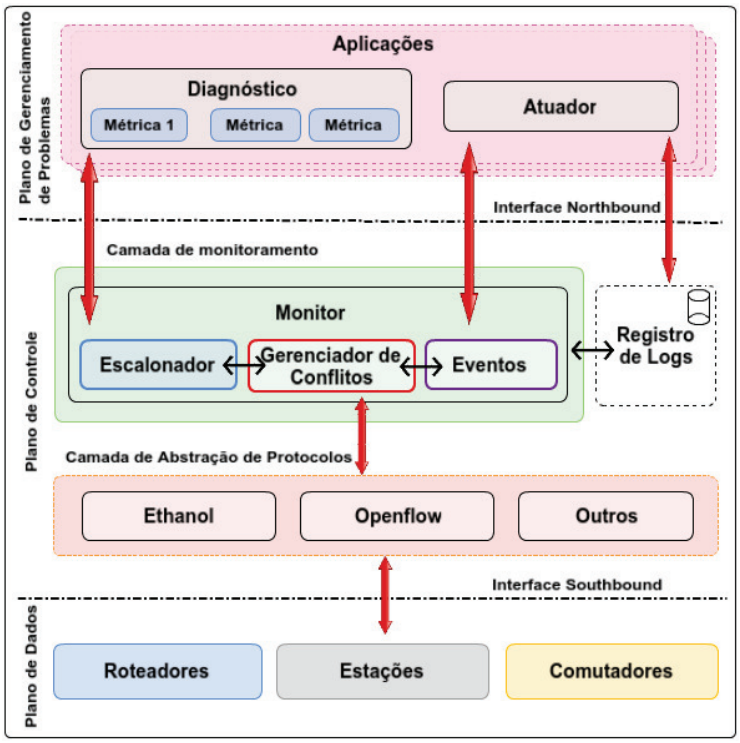


**Figura 1. Arquitetura do Serviço**

O módulo **Atuador** é invocado para atuar na rede quando limiares pré-definidos são atingidos. Assim, o **Atuador** pode executar funções que alterem parâmetros nos componentes de rede objetivando solucionar um problema detectado ou melhorar o desempenho. Nesse módulo, os parâmetros oriundos dos protocolos da camada de abstração são aqueles que podem ser configurados. Seguindo o exemplo anterior, o módulo **Atuador**, ao ser invocado, pode aumentar a potência de transmissão do AP afetado ou mudar o canal de transmissão, solicitar que as estações reassociem a outro AP, entre outros.

### 3.1.2. Plano de Controle

O plano de controle fornece uma visão global da rede e facilita a programação de aplicações de gerenciamento de problemas. O plano foi dividido em camada de monitoramento e camada de abstração de protocolos. A primeira camada realiza o monitoramento da rede, podendo escalonar aplicações conforme prioridades, gerenciar conflitos em ações de configuração, bem como disparar eventos de acordo com uma dada ocorrência. A segunda camada possibilita lidar com os protocolos de rede, a saber: Ethanol (descrito posteriormente), Openflow [3], Netconf [4], SNMP [5], entre outros. Mais informações sobre as camadas desse plano são apresentadas a seguir:

- **Monitor:** realiza o controle e o monitoramento da rede. Através dele são realizadas as coletas de dados da rede que permitem o serviço detectar e solucionar

[3]https://www.opennetworking.org/ja/sdn-resources-ja/onf-specifications/openflow
[4]https://tools.ietf.org/html/rfc6241
[5]https://www.ietf.org/rfc/rfc1157.txt

um problema. Nele, realizada-se a agregação dos dados, o processamento de métricas e a consolidação do parâmetro de tempo de monitoramento informado na aplicação. Contidos nesse módulo estão três sub-módulos:
`Escalonador:` escalona as aplicações do serviço. Dadas as aplicações, realiza solicitações periódicas às camadas inferiores (por exemplo, informações de SNR a cada minuto). Os dados coletados são repassados ao módulo de **eventos** e armazenados em **Registro de Logs**. O escalonador auxilia na otimização da coleta de dados originados de aplicações distintas. Por exemplo, se uma aplicação solicita o SNR a cada 100 $ms$ e outra a cada 200 $ms$, o escalonador evita o disparo duplo de solicitações de SNR oriundos das aplicações distintas.
`Gerenciador de Conflitos:` verifica se as ações enviadas por diferentes aplicações podem causar comportamento conflitante nos dispositivos ou nos protocolos utilizados na interface *Southbound*. Por exemplo, uma aplicação pode solicitar o aumento da potência de transmissão para reduzir a taxa de perda, enquanto outra solicita a redução da potência de transmissão para reduzir a interferência entre APs. O módulo de gerenciamento de conflitos resolve tais situações em função da prioridade de cada uma. Aplicações de maior prioridade sobrescrevem as ações das aplicações de menor prioridade.
`Eventos:` analisa as medições periodicamente e, em caso de evento de rede, dispara o evento para as aplicações (módulo **atuador**) com as regras previamente inseridas. Um exemplo de evento seria a taxa de perda ser superior a 10%.

- **Registro de Logs:** registra os logs das operações executadas pelo serviço e das leituras realizadas, permitindo auditorias aos administradores das redes.
- **Camada de Abstração de Protocolos**: representa os protocolos suportados pela interface *Southbound* do serviço. O serviço foi projetado para suportar múltiplos protocolos. Nesta camada podem ser utilizados protocolos como Openflow, NetConf, SNMP, Ethanol, entre outros.

O Ethanol consiste em uma arquitetura SDN para o gerenciamento de redes IEEE 802.11 (WLAN) empresariais e domésticas [Moura et al. 2015]. É definida uma interface *Southbound* que permite o controle de pontos de acesso IEEE 802.11 compatíveis, bem como estações sem fio que implementem alguns padrões IEEE 802.11. Através destas características, a arquitetura fornece recursos para controle de *handoff* de estações entre os roteadores, controle de autenticação de usuários, criação de redes virtuais, configuração de QoS, localização de usuários na rede, entre outras. Assim como no Openflow, são utilizadas conexões seguras (Socket SSL) para a comunicação do controlador com os clientes e APs.

### 3.1.3. Plano de Dados

O plano de dados é composto pelos dispositivos de encaminhamento ou roteamento da rede que suportam pelo menos um dos protocolos ativos no *HomeNetRescue*. Neste artigo, são empregados planos de dados compatíveis com os protocolos OpenFlow e Ethanol. Assim, o *HomeNetRescue* suporta switches SDN (via OpenFlow), bem como APs IEEE 802.11 (via Ethanol) e estações que implementam recursos de gerenciamento do protocolo IEEE 802.11/2012 (também via Ethanol). Nos **APs** IEEE 802.11, o *HomeNetRescue* pode realizar operações como mudar a frequência e canais de operação, mo-

dificar a potência de transmissão, os parâmetros do protocolo MAC (uso de RTS/CTM, DTIM, etc), solicitar varreduras do espectro, gerenciar parâmetros de QoS, controlar a associação de estações, entre outras operações.

Nas **estações**, devido ao suporte aos padrões IEEE 802.11 mencionados, é possível receber informações sobre a interface de rede sem fio, solicitar varreduras do espectro, ajustar parâmetros da camada física tais como a potência de transmissão, alterar o *bitrate*, realizar a troca para outra rede WiFi, entre outros ajustes. Vale ressaltar que na versão atual do Ethanol as estações devem executar um software que implementa os padrões de gerenciamento do IEEE 802.11, pois o kernel do Linux ainda não possui suporte para eles. Entretanto, tal software será desnecessário quando o suporte aos padrões for nativo.

### 3.2. Aplicabilidade da Solução

O *HNR* foi desenvolvido para permitir o gerenciamento autônomo de problemas e de falhas em redes domésticas. Conforme mencionado na Seção 3, o serviço pode ser executado pelas provedoras de acesso à internet, sendo esta uma alternativa de gerenciamento, à distância, das redes nas residências dos usuários. Diferentemente das redes empresariais, planejadas e configuradas por administradores de rede, as redes domésticas não apresentam *hardware* e *software* padronizados, assim como administradores dedicados para lidar com os possíveis problemas que essas redes são susceptíveis. Disso, a complexidade adicional e a demanda por uma solução extensível, customizável e modular para o gerenciamento desse tipo de rede, como também para o gerenciamento de problemas e falhas.

O *HomeNetRescue* provê extensibilidade e customização ao permitir que novas aplicações sejam facilmente adicionadas ao serviço, assim como novos dispositivos e parâmetros a serem monitorados. Além disso, o *HNR* possibilita a coordenação das redes sem fio ou cabeadas de uma mesma provedora. Deste modo, a provedora poderia alocar os canais dos roteadores de seus clientes em um edifício, evitando interferências e melhorando a qualidade do serviço oferecido. Outra possibilidade seria a detecção de áreas de sombra (desvanecimento) nas residências, que poderiam ser minimizadas ajustando a potência ou gerando sugestões para os usuários substituírem seus APs. O *HNR* também poderia gerenciar o ponto de conexão entre APs e estações, requisitando que as estações reassociem a outros APs visando obter melhor qualidade de conexão.

Além dessas aplicações, o *HomeNetRescue* também poderia ser utilizado para realizar a coordenação entre APs e estações, gerenciando: coordenadamente a intensidade de sinal dos APs de uma mesma rede de modo que não se interfiram; a alocação de canais de acordo com a vizinhança, realizando a leitura dos *beacons* dos outros APs; a alteração da taxa de *bitrate* ao detectar variações inadequadas na vazão; a associação de estações entre os APs da rede, de modo a balancear a carga nos roteadores; e a alteração da potência de transmissão das estações conforme características de SNR presenciadas nelas.

## 4. Avaliação

Nessa seção, o *HomeNetRescue* foi avaliado em um protótipo que simula problemas presenciados em redes domésticas. Embora o HomeNetRescue tenha sido modelado para lidar com problemas apresentados nas camadas 2-5 (modelo TCP/IP), nesse artigo avaliamos principalmente eventos na camada de enlace (camada 2). Desta maneira, o foco

da avaliação concentrou-se nas anormalidades que podem colaborar para a degradação da qualidade do sinal. Assim, foram apresentadas soluções visando solucionar problemas que provoquem variações em fluxos sem fio devido a interferências, localidade dos clientes (baixa cobertura do sinal), entre outras causas. O caso de uso apresentado é composto de uma breve descrição do problema, seguido de como ele é detectado e qual a abordagem adotada em sua solução ou minimização. Visando a confiabilidade dos resultados, exceto os gráficos de *jitter* e atraso, que foram realizadas 500 leituras, os demais experimentos foram repetidos 33 vezes. Os resultados correspondem a média das repetições.

### 4.1. Metodologia

Para a execução dos experimentos com o *HomeNetRescue* foram utilizados: $a)$ uma máquina virtual (controlador) com processador Intel Xeon E312xx 2.2 GHz e 4 GB de RAM, rodando Linux Ubuntu 14.04 (*host system*); $b)$ três computadores pessoais, com processadores: Intel Core 2 Duo 2.53 GHz (roteador com Hostapd[6]), Intel Pentium E2220 2.40 GHz (estação) e Intel Core 2 Duo 6300 1.86 GHz (estação), todos com 2 GB de RAM e Linux Ubuntu 14.04; $c)$ uma placa *Universal Software Radio Peripheral* (USRP) B210[7] (GNU Radio) com uma placa filha FE-TX2; e $d)$ um roteador Linksys WRT54G. Os dois últimos equipamentos ($c$ e $d$) foram utilizados para a geração de interferências sintéticas.

Os cenário da rede é descrito a seguir. A rede entre o $HNR_{AP1}$ (AP HNR) e a $HNR_{STA1}$ (estação HNR) é a rede controlada pelo o *HomeNetRescue*. Uma rede entre o $INT_{AP}$ (AP interferente) e a $INT_{STA}$ (estação interferente) foi montada nos cantos opostos de uma sala. Ao lado do $HNR_{AP1}$, a placa $USRP$ (GNU Radio) foi posicionada e configurada para gerar ruído gaussiano, com 100% de ganho, no canal de transmissão da rede controlada pelo *HomeNetRescue*. O mesmo canal é utilizado na rede do roteador Linksys WRT54G ($INT_{AP}$ ) e $INT_{STA}$. Sua potência de transmissão foi configurada em 18 $dbm$ gerando tráfego no mesmo canal de transmissão. O objetivo desta disposição consistiu em induzir a rede de testes ($HNR_{AP1}$ e $HNR_{STA1}$) a um alto nível de interferência, simulando assim ruídos de aparelhos domésticos (por exemplo, aparelhos micro-ondas, telefones sem fio e babás eletrônicas, etc), bem como a interferência entre APs devido a sobreposição de canais do IEEE 802.11

O tráfego de dados gerado nos experimentos foi obtido através da ferramenta cliente-servidor Bwping[8] (similar ao Iperf) . A ferramenta utiliza o protocolo de transporte UDP e é capaz de fornecer estatísticas de vazão, atraso, *jitter*, quantidade de pacotes enviados e recebidos, entre outras. Para os experimentos, a ferramenta foi configurada para gerar tráfego no limite da capacidade da transmissão sem fio.

### 4.2. Implementação

O *HomeNetRescue* e o controlador Ethanol foram desenvolvidos em Python 2.7 (mais de 5000 linhas de código[9]). Ambos compõem módulos executados no controlador POX Dart [10] em conjunto com o protocolo OpenFlow 1.3. Este controlador foi adotado

[6]http://linuxwireless.org/en/users/Documentation/hostapd/

[7]https://www.ettus.com/product/details/UB210-KIT

[8]https://github.com/h3dema/bwping-udp

[9]A implementação do serviço encontra-se no github, porém ainda em modo privado. Futuramente objetivamos disponibilizá-la.

[10]https://github.com/noxrepo/pox/tree/dart

por permitir rápida prototipação de aplicações, por ser implementado em Pyhton com código aberto e por ser de fácil configuração e execução em plataformas Linux. Embora o POX não seja o mais adequado para aplicações que demandem alto desempenho, para o HomeNetRescue não é um gargalo da rede, assim como não é o principal ponto de falha, pois técnicas de alta disponibilidade para controladores são consideradas um problema resolvido na literatura. Por adotar uma abordagem centralizada, tal controlador proporciona benefícios em termos de menor tempo de convergência em relação a uma solução distribuída. Adicionalmente, em um ambiente de produção do HomeNetRescue, os controladores de alta disponibilidade, ONIX e ONOS, poderiam ser utilizados.

Como mencionado, o controlador Ethanol pode usufruir dos recursos do protocolo OpenFlow. Em trabalhos futuros, outras versões do protocolo, com outras funcionalidades, podem ser utilizadas para outras aplicações. Por exemplo, uso de múltiplas tabelas no contexto de segurança, entre outras aplicações. Ainda, o controlador Ethanol também possui implementações na linguagem C (mais de 35000 linhas de código). Mensagens são trocadas entre o controlador e os agentes Ethanol. O controlador Ethanol permanece em execução enquanto as aplicações do *HomeNetRescue* requisitam informações (métricas) aos agentes em execução nos componentes da rede.

### 4.3. Caso de Uso: Combatendo Enlaces Sem Fio Ruins

Este caso de uso representa um cenário onde a interferência degrada a qualidade da transmissão sem fio nos APs, causando perda de pacotes, atraso na entrega de pacotes, retransmissões e diminuição da vazão de recebimento. Assim, nesse caso de uso é descrita a abordagem utilizada pelo *HomeNetRescue* para o tratamento de interferência detectada na transmissão sem fio dos APs. Neste sentido, O $HNR_{AP1}$ foi configurado com uma potência de transmissão de 1 $dbm$, enquanto que o $INT_{AP}$ foi configurado em 18 $dbm$, com uma antena acoplada com ganho de 2 $dbm$. Já a $USRP$, acoplada com uma antena de 8 $dbm$, foi configurada para emitir sinais gaussianos com 100% de ganho. Por limitações de espaço, o algoritmo do caso de uso foi substituído pela descrição a seguir.

O processo de verificação de interferência consiste do serviço requisitar ao $HNR_{AP1}$, a cada 10 $s$, através do controlador, as métricas qualidade do sinal e quantidade de pacotes perdidos de sua interface de rede. Isto feito, quando a quantidade de pacotes perdidos supera o limiar[11] especificado na aplicação, têm-se a necessidade de atuar na rede. Para solucionar este tipo de problema, primeiramente as métricas qualidade do sinal e quantidade de pacotes perdidos juntamente com o limiar de perda de pacotes aceitável são configuradas na aplicação do serviço. Após, o módulo monitor (Seção 3.1) requisita tais informações das camadas adjacentes verificando se a perda no $HNR_{AP1}$ encontra-se aceitável. Em caso negativo, o atuador repassa ao $HNR_{AP1}$ o valor da potência de transmissão a ser alterada, em $dbm$. Realizado o procedimento, em sequência, a aplicação registra o log da operação efetuada e a taxa de perda de pacotes, para que posteriormente, em uma nova consulta, o serviço realize e registre o cálculo de perdas de pacotes.

### 4.4. Resultados

Nas figuras 2, 4 e 6, diagramas de caixa foram utilizados para exibir os resultados. Neles, o eixo vertical representa a variável a ser analisada e o eixo horizontal os fatores

[11]Definido empiricamente para os experimentos, pois depende da quantidade de fluxos vigentes no $HNR_{AP1}$.

de interesse, indicado nas legendas de cada figura. Em cada um, a caixa é delimitada na parte superior pelo quartil $Q_3$ (distribuindo 25% dos dados acima) e na parte inferior pelo primeiro $Q_1$ (distribuindo 25% dos dados abaixo). O traço interno indica a mediana (distribuindo 50% dos dados abaixo e 50% acima). As duas linhas na horizontal que se estendem a partir da caixa são os bigodes. O intervalo interquartílico (II) é dado em função $II = Q_3 - Q_1$. O limite superior (LS) é dado em função de $LS = Q_3 + 1,5 * II$, enquanto que o limite inferior (LI), em função de $LI = Q_1 - 1,5 * II$. Por fim, os valores discrepantes ou *outliers* foram desconsiderados.

Na Figura 2, é exibido o impacto das interferências no ambiente de testes e como essas afetam as transmissões sem fio. Com isso, é demonstrada a relação entre o número de pacotes perdidos e a vazão percebida no $HNR_{AP1}$. Isto, para as situações onde o $HNR_{AP1}$ não esteve sob influência da interferência; quando a interferência foi inserida somente pelo tráfego do $INT_{AP}$ ou somente pela $USRP$; e, quando ambos geram interferências simultaneamente. Baseado nisso, conforme esperado, foi observado que à medida que se acrescenta novas fontes de interferência, a quantidade de pacotes perdidos por vazão aumenta, validando que as interferências comprometem o desempenho da rede.

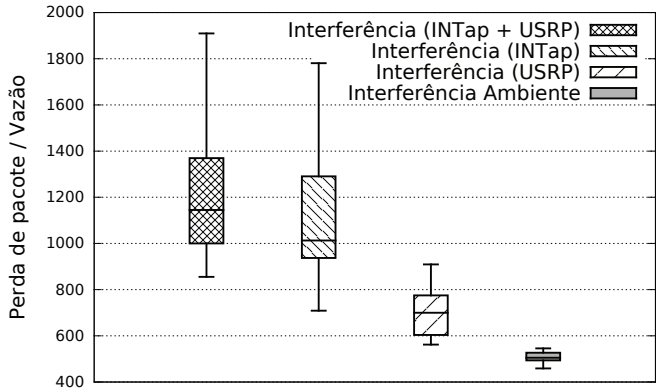


**Figura 2. Razão entre número de pacotes perdidos pela vazão percebida no AP**

Na Figura 3, é ilustrada a variação da vazão durante o experimento. Na fase inicial do mesmo (em amarelo entre 0 e 150 $s$), o $HNR_{AP1}$ esteve com uma potência de saída de 1 $dbm$. Em 150 $s$ do experimento, foi acrescentada a interferência do $USRP$. Aguardou-se alguns segundos para a vazão estabilizar. A figura mostra a vazão média depois da estabilização, no período de 170 a 250 $s$. Em 250 $s$, o controlador foi acionado, detectando um problema de vazão (através da quantidade de perdas) e aumentando a potência do $HNR_{AP1}$ para 15 $dbm$ (capacidade máxima). Com isso, o HNR conseguiu obter uma melhora de 7% na vazão percebida pelo cliente. O HNR também é capaz de reduzir dinamicamente tal potência, assim como utilizar outras métricas. Isso viabiliza o gerenciamento da rede com mais justiça, por exemplo, reduzir ruídos nos APs vizinhos uma vez que a vazão alcançada já atenda à necessidade da aplicação.

Vale ressaltar que ao executar o *HomeNetRescue*, o processamento realizado na CPU do $HNR_{AP1}$ alcançou no máximo 3% da capacidade total do dispositivo. A utilização de memória não ultrapassou 3% (cerca de 40 MB). Esses valores indicam que o serviço demanda poucos recursos dos dispositivos e pode, por exemplo pode ser utilizado com o TP-Link TL-WR2543ND que possui processador 400 MHz, 64 MB de memória

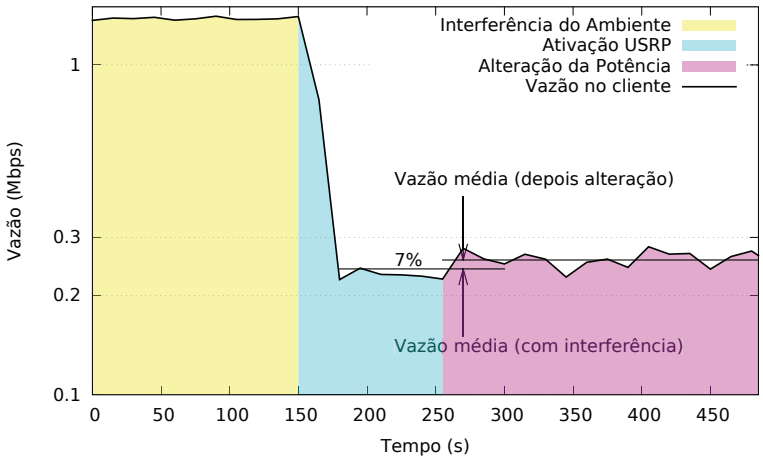


**Figura 3. Exemplo de execução do *HomeNetRescue* com ganho de 7% para um fluxo frente a interferência sintética**

RAM e 8 MB de memória flash. As alterações de potência de transmissão, quando realizadas, gastaram no máximo 29 $ms$ entre a detecção do problema pelo controlador e o envio da regra ao dispositivo.

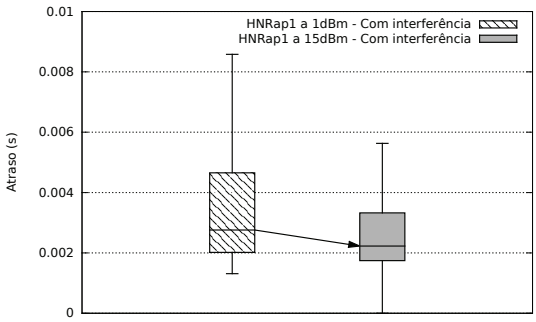


**Figura 4. Atraso medido pelo cliente**

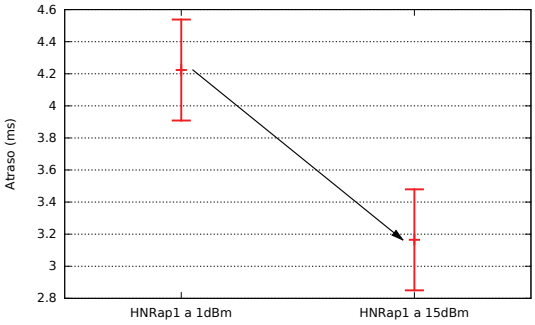


**Figura 5. Atraso médio com intervalo de confiança de 95% com interferência**

Na Figura 4, são apresentados os valores de atraso (em $s$) obtidos no experimento, medidos no cliente. Na barra da esquerda é mostrado o atraso observado pelo cliente com o $HNR_{AP1}$ configurado para a potência de 1 $dbm$ sob a influência do $INT_{AP}$. A barra à direita mostra a situação quando o controlador aumenta a potência do $HNR_{AP1}$ para 15 $dbm$. Pode-se notar na figura (seta) que há uma melhoria, com a redução do atraso resultado da ação do serviço. Na Figura 5, é representado o intervalo de confiança do atraso médio para a situação com o $HNR_{AP1}$ funcionando na potência de 1 $dbm$ e na situação depois da alteração para 15 $dbm$. Não existe sobreposição entre os intervalos de confiança da média, portanto as médias são independentes.

Na Figura 6, apresenta-se os valores de *jitter* (s) obtidos no experimento. Na barra à esquerda é exibido o *jitter* observado pelo cliente com o $HNR_{AP1}$ configurado para a potência de 1$dbm$ sob a influência do $INT_{AP}$ . A barra à direita mostra a situação quando o serviço aumenta a potência do $HNR_{AP1}$ para 15 $dbm$, mantidas as outras condições constantes. Verifica-se que há uma redução do *jitter*. O intervalo de confiança do *jitter*

médio é mostrado na Figura 7. As médias são independentes pois não existe sobreposição dos intervalos. Dessa forma, vemos que o controlador consegue, ao modificar a potência do ponto de acesso, melhorar o *jitter* dinamicamente quase obtendo uma situação próxima ao cenário sem interferência.

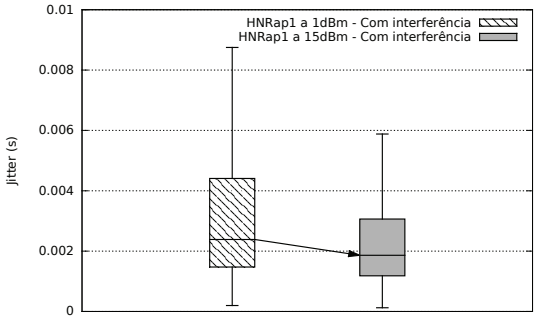


**Figura 6. Jitter medido no cliente**

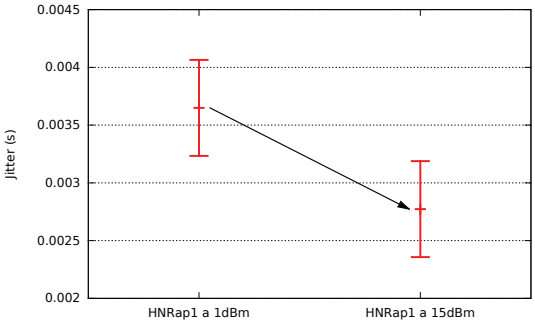


**Figura 7. Jitter médio com intervalo de confiança de 95% com interferência**

Finalmente, o *HomeNetRescue* melhorou o desempenho da rede. A perda de mensagens diminuiu, resolvendo o problema de perdas acima do limiar, o que impacta diretamente na vazão detectada. O objetivo do artigo concentrou-se na descrição do protótipo do HomeNetRescue em detrimento dos algoritmos de controle para tratar eventos específicos de falha ou como diferenciar tais eventos.

## 5. Conclusão e Trabalhos Futuros

Neste artigo, foi apresentado o *HomeNetRescue*, um serviço SDN para o gerenciamento autônomo de redes domésticas (inclusive redes sem fio) voltado para a detecção, o diagnóstico e a solução automática ou minimização de problemas que também pode atuar em aspectos de desempenho nessas redes. O *HomeNetRescue* apresenta uma arquitetura genérica e pode ser empregado pelas provedoras de acesso à Internet para que estas gerenciem problemas à distância nas redes domésticas de seus usuários. Tal serviço pode agregar a essas redes funcionalidades de detecção e solução automática de problemas de modo que estas se tornem mais estáveis e confiáveis. Isto pode proporcionar redução de custos para as provedoras e para os usuários, gerar menor demanda por serviços de suporte e menor tempo de recuperação em caso de falhas.

A partir da avaliação experimental,o *HomeNetRescue* demonstrou benefícios científicos proporcionando ganhos na vazão, redução em atrasos e no *jitter* de transmissões sem fio e redução da intervenção humana na solução de problemas. O serviço foi capaz de detectar um aumento na quantidade de perda de pacotes devido a colisões, atuando na rede automaticamente, aumentando a potência de transmissão, sendo capaz de combater enlaces sem fio ruins. Como trabalho futuros, pretende-se implementar casos de uso que demonstrem a capacidade do serviço em coordenar os dispositivos da rede, gerenciando APs em que as estações estão conectadas e atuando sobre os mecanismos de QoS.

## 6. Agradecimentos

Agradecimentos às instituições de amparo à pesquisa CNPq, CAPES e FAPEMIG pelo financiamento e suporte para o desenvolvimento dessa pesquisa.

## Referências

[Biswas et al. 2015] Biswas, S., Bicket, J., Wong, E., Musaloiu-E, R., Bhartia, A., and Aguayo, D. (2015). Large-scale measurements of wireless network behavior. In *ACM SIGCOMM*.

[Bouchet et al. 2014] Bouchet, Javaudin, Kortebi, Adbellaouy, E., Brzozowski, Katsianis, Mayer, Guan, Lebouc, Fontaine, Cochet, Jaffré, Mengi, Celeda, Aytekin, G., and Kurt (2014). Acemind: The smart integrated home network. In *2014 International Conference on Intelligent Environments (IE)*.

[Cisco 2015] Cisco (2015). Cisco Visual Networking Index: Forecast and Methodology, 2015–2020. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf. [Online; acessado 14-Novembro-2016].

[DiCioccio et al. 2012] DiCioccio, L., Teixeira, R., and Rosenberg, C. (2012). Measuring and characterizing home networks. *SIGMETRICS Perform. Eval. Rev.*

[Dong and Dulay 2011] Dong, C. and Dulay, N. (2011). Argumentation-based fault diagnosis for home networks. In *Proceedings of the 2nd ACM SIGCOMM (HomeNets)*.

[Fratczak et al. 2013] Fratczak, T., Broadbent, M., Georgopoulos, P., and Race, N. (2013). Homevisor: Adapting home network environments. In *2013 Second European Workshop on Software Defined Networks (EWSDN)*, pages 32–37.

[Gheorghe et al. 2015] Gheorghe, G., Avanesov, T., Palattella, M.-R., Engel, T., and Popoviciu, C. (2015). SDN-RADAR: Network troubleshooting combining user experience and SDN capabilities. In *IEEE NetSoft*, pages 1–5.

[Guedes et al. 2012] Guedes, D., Vieira, L. F. M., Vieira, M. M., Rodrigues, H., and Nunes, R. V. (2012). Redes definidas por software: uma abordagem sistêmica para o desenvolvimento de pesquisas em redes de computadores. *Simpósio Brasileiro de Redes de Computadores (SBRC)*, pages 160–210.

[Kim et al. 2014a] Kim, K.-H., Nam, H., and Schulzrinne, H. (2014a). WiSlow: A Wi-Fi network performance troubleshooting tool for end users. In *IEEE INFOCOM*, pages 862–870.

[Kim et al. 2014b] Kim, K.-H., Nam, H., Singh, V., Song, D., and Schulzrinne, H. (2014b). DYSWIS: crowdsourcing a home network diagnosis. In *International Conference on Computer Communication and Networks (ICCCN)*, pages 1–10.

[Macedo et al. 2015] Macedo, D. F., Guedes, D., Vieira, L. F. M., Vieira, M. A. M., and Nogueira, M. (2015). Programmable networks: From software-defined radio to software-defined networking. *IEEE Communications Surveys Tutorials*.

[Moura et al. 2015] Moura, H., Bessa, G. V., Vieira, M. A., and Macedo, D. F. (2015). Ethanol: Software Defined Networking for 802.11 Wireless Networks. *IFIP/IEEE International Symposium on Integrated Network Management (IM)*.

[Perera et al. 2014] Perera, C., Liu, C., Jayawardena, S., and Chen, M. (2014). A survey on internet of things from industrial market perspective. *IEEE Access*, pages 1660–1679.

[Sundaresan et al. 2013] Sundaresan, S., Grunenberger, Y., Feamster, N., Papagiannaki, D., Levin, D., and Teixeira, R. (2013). WTF? locating performance problems in home networks.

[Yiakoumis et al. 2011] Yiakoumis, Y., Yap, K.-K., Katti, S., Parulkar, G., and McKeown, N. (2011). Slicing home networks. In *Proceedings of the 2nd ACM SIGCOMM (HomeNets)*, pages 1–6.