Projeto redes: implementação de uma MANET para a conexão de dispositivos IoT utilizando protocolo OLSR

Cristiano Michel Nunes[1], **João Mário Lopez Brezolin** [1]

[1]Instituto Federal Sul-rio-grandense (IFSUL) - Campus Passo Fundo
Estrada Perimetral Leste, 150, Bairro São José - Passo Fundo - RS - Brazil

cristianomn@gmail.com, joao.brezolin@passofundo.ifsul.edu.br

Abstract. *This paper presents a study about implementing a Mobile Ad Hoc Network using the OLSR routing protocol. This research aimed to demonstrate that through this protocol it is possible to implement a scalable, resilient and self-configuring network that allows the connection of IoT devices. The results achieved demonstrated that the OLSR protocol has successfully established the infrastructure required to interconnect these devices.*

Resumo. *O presente artigo apresenta o estudo acerca implementação de uma* Mobile Ad hoc Network *utilizando o protocolo de roteamento OLSR. Objetivou-se com essa pesquisa demonstrar que por meio desse protocolo é possível implementar uma rede para a conexão de dispositivos IoT escalável, resiliente e autoconfigurável. Os resultados alcançados demonstraram que o protocolo OLSR permitiu estabelecer com sucesso a infraestrutura necessária para realizar a interconexão desses.*

1. Introdução

Com o constante aumento do número de dispositivos IoT(*Internet of Things*) utilizados atualmente, cabe trazer a discussão a forma de como melhor utilizar esta tecnologia que já é uma realidade no cotidiano das pessoas. A difusão das tecnologias IoT demandam a implementação de mecanismos que permitam a agregação destes dispositivos ao contexto do cenário atual, no qual, se multiplicam dispositivos dessa natureza. Com o aumento de opções de dispositivos surge a necessidade de prover uma infraestrutura de conexão adequada.

Uma das formas comuns de viabilizar a comunicação entre esses dispositivos é fazer uso de redes sem fio. Essas redes caracterizam-se por estruturar-se em uma topologia estrela, na qual a comunicação entre todos os componentes da rede devem necessariamente passar por um centralizador/controlador que identifica os atores da comunicação (emissor e receptor), e estabelece um canal de comunicação entre eles.

Os dispositivos de comunicação sem fio, porém, estão sujeitos a interferências em suas faixas de frequência devido a outros equipamentos que operam na mesma faixa de frequência ou a mecanismos que geram ondas eletromagnéticas capazes de interferir no sinal transmitido ou recebido. Há ainda problemas com obstáculos físicos que podem atenuar a onda propagada para envio de dados ou na recepção de dados.

A distância entre o dispositivo e o *gateway* IoT também pode causar perda de comunicação, já que dispositivos IoT, geralmente, são projetados para baixo consumo

de energia, consequentemente possuem menor potência de transmissão de dados, assim como, velocidades de transmissão.

Nesse sentido, o presente trabalho procura avaliar o uso do protocolo OLSR (*Optmized Link State Routing Protocol*) para qualificar a comunicação de dispositivos IoT (*Internet of Things*) em redes *mesh*, objetivando oferecer mais rotas de acesso a estes dispositivos em uma infraestrutura escalável. A utilização de tabelas dinâmicas de rotas, gerenciadas pelo protocolo OLSR permitem instaurar uma comunicação mais eficiente, e também proporcionar uma infraestrutura sem fio escalável, autoconfigurável e multiplataforma.

Neste trabalho demostraremos que uma rede *mesh* gerenciada pelo protocolo OLSR, oferece uma infraestrutura de rede escalável, resiliente, autoconfigurável e multiplataforma para prover uma infraestrutura sem fio para conexões de dispositivos IoTs. Para tanto, foi implementada uma MANET (*Mobile Ad hoc Network*) na qual os testes de validação foram implementados.

2. Referencial teórico

A comunicação sem fio ocorre na mesma através de ondas de rádio nas quais transmissores/receptores operam na mesma faixa de frequência e utilizam um mesmo regramento (protocolo) de comunicação. A vantagem da comunicação *wireless* é a mobilidade e elimina a necessidade de cabeamento [Holt and Huang 2010]. A rede WiFi para computadores mais comumente utilizada, é estruturada na topologia estrela, na qual a comunicação entre todos os componentes da rede devem necessariamente passar por um centralizador/controlador que identifica os atores da comunicação, emissor e receptor, e estabelece um canal de comunicação entre eles.

Uma rede *mesh* se caracteriza por ser autoconfigurável, auto gerenciável, orgânica, resiliente e escalável [ZHANG et al. 2007]. As redes sem fios na topologia malha são chamadas de WMNs (*Wireless Mesh Networks*) e funcionam como uma rede *Ad hoc*, onde todos os *hosts* comunicam entre si. No entanto, uma rede *Ad hoc* somente consegue se comunicar com os nós ao alcance de sua placa *wireless*. Os nós não fazem o roteamento das informações. Ao implementar-se algoritimos de roteamento em nós *Ad hoc* cria-se uma rede *mesh*.

Segundo [ZHANG et al. 2007], WMNs possuem grande tolerância a falhas, principalmente em comparação com as WLANs (*Wireless Local Area Network*) no modo infraestrutura, pois requerem um AP (*Access Point*) para estabelecer *links* de comunicação entre *hosts* da rede. Nas WMNs todos os nós da rede que estão ao alcance de seu rádio transmissor são capazes de se comunicar entre si, portanto, a rede *mesh* também oferece uma melhor redundância, pois pode haver mais de uma rota para se comunicar com um nó da rede.

O protocolo OLSR é uma variação da versão do tradicional protocolo de estado de conexão [Boukerche 2009]. O OLSR é um protocolo de roteamento pró-ativo [OLSR page] que caracteriza-se por utilizar tabelas dinâmicas em cada nó da rede com a rota para cada nó da rede, assim quando a comunicação precisa ser estabelecida, o nó que envia a mensagem já sabe quais as rotas para se chegar ao seu destino. São protocolos conhecidos como *table-drive* [ZHANG et al. 2007].

Protocolos pró-ativos geram mais *overheads* na rede e mais sobrecarga de processamento nos roteadores, pois há um custo para manter uma tabela dinâmica de rotas atualizada. Protocolos pró-ativos geralmente utilizam o clássico algoritmo da métrica de contagem de saltos, *hop-count*, para estabelecer a localização dos nós na rede. Esta contagem é feita através de mensagens *broadcast* de *HELLOs*, em que cada nó que a recebe retransmite. Esta técnica gera um alto grau de *flooding*, pois cada nó retransmite o sinal de *HELLO* para todos os vizinhos que seu rádio alcança, assim como responde para o nó origem, causando *overheads* na rede e sobrecarga de processamento nos roteadores *Ad hoc*.

O OLSR otimiza os estados de conexão diminuindo o número de mensagens de *broadcast* retransmitidas, ou seja, diminui o número *HELLOs* na rede. Para isso, o protocolo OLSR implementa nós MPRs (*Multi Point Realy*) em que somente eles são responsáveis por retransmitir mensagens de *broadcast* para identificar a localização dos nós na rede. Nós MPRs, enviam mensagens de *broadcast* TC (*Topology Control*). Com isso são criadas as tabelas dinâmicas de roteamento, pois sabendo o número de MPRs existentes e que estão sempre a no máximo 2 saltos de distância dos nós, é possível calcular onde estão e montar uma tabela [ZHANG et al. 2007].

A implementação de nós MPRs é um aspecto central do protocolo OLSR. Sua implementação permite diminuir pela metade o *flooding* de mensagens de *broadcast* retransmitidos na rede para localizar os nós da rede. Isto causa uma diminuição significativa no número de *overheads* e sobrecarga de processamento de roteadores [Boukerche, 2009].

O protocolo ainda oferece mensagem de *broadcast* MID (*Multi Interface Declaration*), em que o nó declara ter mais de uma interface de comunicação, ou seja, mais de um rádio para transmissão. Isso faz com que um nó MID possa se comunicar com nós vizinhos com seus múltiplos rádios, o que torna a comunicação mais confiável, pois se um dos rádios do nó MID falhar, o nó ainda está ativo na rede utilizando o outro rádio. Outro serviço de mensagem de *broadcast* do OLSR é o HNA (*Host and Network Association*), em que o nó da rede que envia esta mensagem anuncia para os outros nós que ele é um *gateway* para à Internet ou para outra rede ou mesmo outro nó da rede, e ainda, o protocolo OLSR permite mais de um nó como *gateway* na rede. A próxima seção descreve os recursos utilizados para a implantação da MANET pelo protocolo OLSR.

3. Descrição da infraestrutura de redes implementada

Para compor os nós da rede *mesh* q foi utilizado um roteador *wireless* da marca TP-Link modelo N750 (TL-WDR4300) e dois computadores de placa única Raspberry Pi 3 modelo B. O Roteador *wireless* TP-Link N750 modelo TL-WDR4300, opera em duas frequências de rádio transmissão/recepção, em 2.4GHz a 300Mbps e em 5GHz a 450Mbps. Seu *firmware* proprietário é baseado em Linux, o que possibilita a substituição do seu *firmware* por outro de código aberto, neste caso, o *firmware* OpenWrt, estendendo suas funcionalidades permitindo adicionar suporte ao protocolo OLSR. Após a substituição pelo novo *firmware*, foi instalado os pacotes *olsrd* e *olsrd-plugins* ao *firmware* OpenWrt.

Utilizou-se apenas a placa de rede *wireless* que opera na frequência de 2.4GHz do roteador TP-Link, que foi configurada para operar em modo *Ad hoc*, e no canal 11. Foi necessário adicionar uma interface virtual nas configurações do OpenWrt para criar uma rede *wireless* com o ESSID (*Extend Service Set ID*) de nome OLSR, com endereço IP

estático 192.168.0.1. A placa rede *ethernet* WAN deste roteador TP-Link foi conectado por um cabo de rede *ethernet* a um outro roteador com a função de prover o acesso à Internet, fornecendo a rede *mesh* um *gateway* para Internet.

Para compor os outros nós da rede *mesh* foram utilizados dois computadores de placa única Raspberry Pi 3 modelo B com processador de 1GHz de 4 núcleos de arquitetura ARM, 1GB de memória RAM, 4 portas USB, 1 placa de rede WiFi com suporte as tecnologias 802.11b,g e n, tem suporte a Bluetooth 4.1 e uma interface *ethernet* com velocidade máxima de tráfego de dados de 100Mbps. Os minicomputadores Raspberry Pi são computadores de baixo custo e alto desempenho [Raspberry page].

Nos dois minicomputadores Raspberry Pi 3 modelo B, o sistema operacional instalado foi a distribuição Linux baseada no Debian o Raspbian. O SO foi instalado em dois cartões de memória micro SD com capacidade de armazenamento de 16GB. A configuração do protocolo OLSR nos Raspberrys, após a instalação do pacote *olsrd*, é necessário editar o arquivo *olsrd.conf* para definir a configuração operacional do protocolo e os *plugins* utilizados. Configurou-se as placas *Wi-Fi* dos Raspberry para operar em modo *Ad hoc*, no canal 11. Para as configuração da rede *wireless* atribuiu-se o ESSID de nome OLSR em ambos, e os endereços de IP estático, 192.168.0.10 para o Raspberry de *hostname* RASPI01 e 192.168.0.11 o RASPI02.

O protocolo OLSR criou rotas de rede no sistema operacional Raspbian(Raspberry Pi) e no OpenWrt(roteador TP-Link) para cada endereço IP dos nós da rede que estão ao alcance do rádio transmissor/receptor das placas de rede *wireless*, que no primeiro momento pós implementação da rede *mesh*, os 3 nós estão ao alcance um dos outros, portanto, cada nó de rede possui uma rota de rede para cada endereço IP dos *hosts*.

Após a implementação da infraestrutura da MANET deu-se início aos testes. O primeiro foi configurar um *laptop* com o sistema operacional Linux e configurado sua placa *wireless* em modo *Ad hoc*, canal 11 e ESSID igual a OLSR. Com estas configurações foi possível ingressar na rede *mesh*, e efetuar teste de comunicação com o comando *PING* do Linux, com todos os nós, o roteador TP-Link e os dois Raspberry Pi, todos estão ao alcance de seus rádios transmissores/receptores, o comando teve êxito no teste de comunicação com endereços IPs dos *hosts* da rede *mesh*, assim como, a recíproca foi verdadeira.

O próximo passo foi verificar se o protocolo OLSR está operacional, foi implementado uma infraestrutura de testes, em um prédio de 6 andares, onde o roteador TP-Link foi colocado no primeiro andar, o Raspberry Pi com endereço IP 192.168.0.11 no terceiro andar e o último Raspberry Pi no sexto andar. Neste cenário os nós da rede *mesh* não estão todos ao alcance se seus rádios transmissores/receptores, no entanto, acessando cada nó da rede resultou em sucesso o teste de comunicação com o comando PING com os demais *hosts*, constatando-se a operacionalidade do protocolo OSLR.

Para poder usufruir da infraestrutura sem fio disponibilizada pela rede *mesh*, a cada Raspberry Pi foi adicionado uma placa de rede *wireless* USB para servir de *Access Point* através da técnica de IP *masquerading*, também conhecida como NAT (*Network Address Translate*) para a comunicação com a MANET.

Este *Access Point* tem como SSID o nome HOSTAPD_OLSR e utilizou-se o software *HOSTAPD* para prover a conexão a dispositivos sem fio com suporte a tecnolo-

gia 802.11b, g e n, e dar suporte a criptografia WPA/WPA2 para as conexões de dispositivo sem fio. Ademais, nas placas de rede USB plugadas configurou-se endereços IPs estáticos, 192.168.2.1 para o *hostname* RASPI01 e 192.168.3.1 para o *hostname* RASPI02, também foi configura um endereço de *gateway* correspondente aos endereços IPs das placas *onboard*, 192.168.0.10 no RASPI01 e 192.168.0.11 no RASPI02, estas configurações efetuadas no arquivo */etc/network/interfaces* do SO.

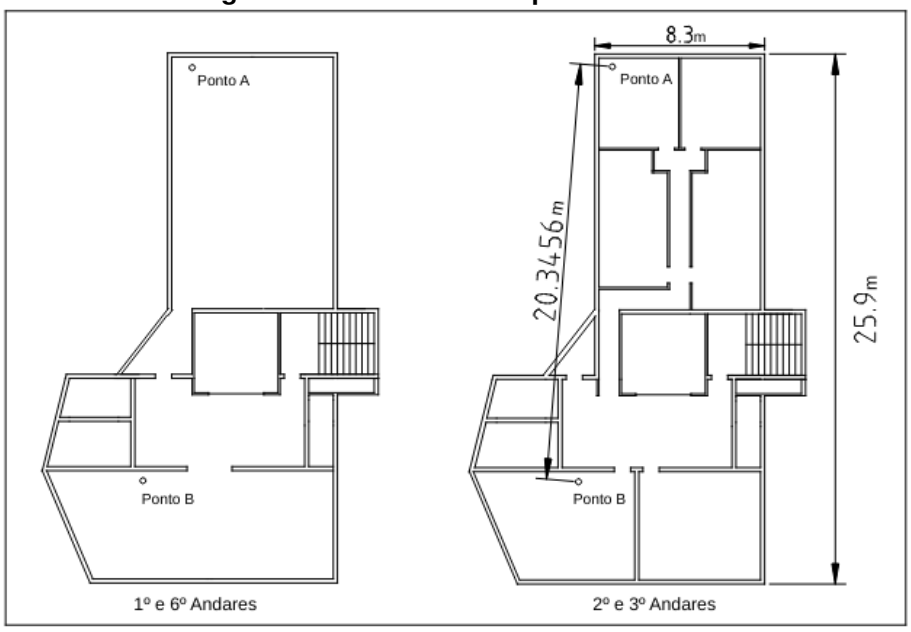
Com a impossibilidade de utilizar a rede *mesh* para a propagação de mensagens *broadcast dhcpdiscover*, e com isso, não permitindo a utilização de um único servidor DHCP para todos os *Access Point*, foi instalado em cada Raspberry Pi o software *DNSMASQ* para fornecer aos *hotspots* HOSTAPD_OLSR um servidor DHCP. A ferramenta de *software DNSMASQ* foi configurado para operar na placa de rede *wireless* USB de cada Raspberry Pi para prover endereçamento dinâmico aos dispositivos IoT que se conectarão aos *Access Point* HOSTAPD_OLSR em cada Raspberry Pi.

Com a conexão efetuada com sucesso ao *hotspot* HOSTAPD_OLSR, por um *laptop* simulando um dispositivo IoT, a rede *mesh* foi implementada fornecendo uma infraestrutura sem fio para ser acessada através dos *hotspots* HOSTAPD_OLSR aos dispositivos sem fio da Internet das Coisas. Os testes e validação são descritos na seção a seguir.

4. Validação e testes da rede mesh com protocolo OLSR

O protocolo OLSR permitiu configurar uma MANET na qual todos os nós se comunicam entre si, porém, nem todos os nós da rede estão ao alcance das suas antenas de rádios transmissores/receptores. No cenário de testes utilizado, orientando-se pela Figura 1, o roteador TP-Link está localizado no primeiro andar no Ponto B, o RASPI02 no terceiro andar no Ponto B e o RASPI01 no sexto andar Ponto B, em vista disso, o RASPI01 e o roteador TP-Link se comunicam através do RASPI02 pelo protocolo OLSR, deste modo, validando o OLSR para formar uma MANET.

Figura 1. Cenário de teste para a MANET



O protocolo OLSR foi configurado nos Raspberry Pi para enviar mensagens HNA informando que são *gateways* para as redes dos *hotspots* HOSTAPD_OLSR, ou seja, rotas para as redes 192.168.2.0 no RASPI01 e 192.168.3.0 no RASPI02. Isso permite que os nós da rede *mesh* mantenham em suas tabelas de rotas OLSR quais os nós da rede fornecem o acesso a uma rede específica, neste caso o RAPSPI01 e RASPI02, ou fornecem acesso a redes externas permitindo a conexão com à Internet, aqui o roteador TP-Link.

Figura 2. Teste de saída para Internet

```
cristiano@viking-laptop: ~
cristiano@viking-laptop:~$ tracepath www.terra.com.br
 1?: [LOCALHOST]                      pmtu 1500
 1:  192.168.2.1                                           2.651ms
 1:  192.168.2.1                                           2.604ms
 2:  192.168.0.11                                         16.240ms
 3:  192.168.0.1                                          87.026ms
 4:  10.72.73.129                                         37.889ms
 5:  10.72.73.1                                           42.837ms
 6:  10.72.21.54                                          89.960ms
 7:  10.2.208.3                                           42.031ms asymm
 9
 8:  10.2.208.6                                           41.471ms
^C
cristiano@viking-laptop:~$
```

A execução do comando *TRACEPATH* no *laptop* conectado ao *Access Point* HOSTAPD_OLSR, que resulta as rotas percorridas, verificou-se que a rede *mesh* implementada está fornecendo uma rede transporte para o acesso à Internet, como mostra a Figura 2.

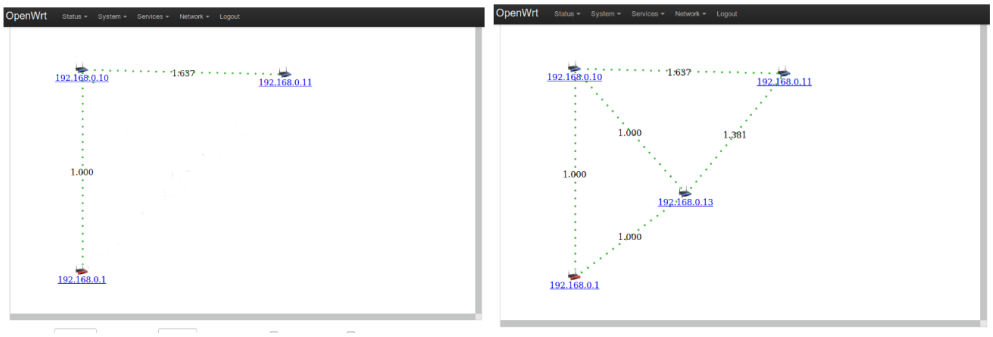
A requisição percorreu todos os nós da rede *mesh*, pois o nó a onde está conectado o *laptop*, o RASPI01(192.168.2.1), não possui acesso à Internet e está distante da comunicação com o roteador TP-Link, que envia mensagens HNA para rede como *gateway* de Internet, necessitando do seu "vizinho", o RASPI02(192.168.0.11), para repassar a requisição até o nó *gateway* da Internet.

Uma MANET, deve ser uma rede autoconfigurável. Para testar essa característica foi configurada uma mudança no cenário de testes. O Raspberry Pi de *hostname* RASPI01 ficou disposto no *Ponto A* e o RASPI02 no *Ponto B* do terceiro andar, já o roteador TP-Link ficou localizado no *Ponto A* do primeiro andar, e um novo cenário de testes foi montado.

A este novo cenário foi adicionado mais um Raspberry Pi de endereço IP 192.168.0.4. Este novo nó foi colocado no segundo andar no *Ponto A* do prédio utilizado como cenário de testes. Ao ligar o RASPI03, demorou pouco segundos para este nó ficar acessível na rede *mesh*. Este tempo refere-se ao período necessário para o protocolo enviar mensagem de HELLO na rede para que os outros nós saibam da sua existência, assim como, para ele descobrir os nós vizinhos para criar sua tabela de rotas para todos os endereços disponíveis. Através do *plugin olsrd_dot_draw* do protocolo OLSR verificasse a nova arquitetura da rede *mesh* pela imagem a direita da Figura 3.

Também foi possível comprovar a resiliência da rede *mesh* implementada, ao desligar o RASPI03, também em poucos segundos, o *backbone mesh* voltou ao seu estado inicial, as tabelas de rotas foram atualizadas, excluindo a rota para o nó desligado, bem

Figura 3. Adição de hosts a Rede *Mesh*



como, para a rede 192.168.4.0/20. A arquitetura *mesh* voltou a seu estado inicial apresentado pela imagem a esquerda da Figura 3 .

Ainda é possível destacar que ao adicionar mais um nó a rede *mesh*, que adicionou mais uma rota para cada nó, a rede torna-se mais tolerante a falhas e redundante, pois nesse novo cenário, no mínimo há duas rotas para acesso a cada nó da rede, se houvesse outro *gateway* para Internet, ter-se-ia uma redundância para a conexão com à Internet.

O protocolo OLSR permite configurar mais de um *gateway* na rede *mesh*, é possível dividir a sobrecarga de dados para à Internet, basta acrescentar um nó a rede como *gateway* à Internet, ou prover acesso a outro nó já existente na rede *mesh* e configurar o protocolo OLSR deste nó para tal. Como o protocolo OLSR sempre estabelece a rota com menor número de saltos para um *gateway* é possível dividir o fluxo de dados para à Internet.

O protocolo OLSR sempre escolhe a melhor rota através de seu algoritmo, e isso ocorre a cada resposta de mensagem HELLO, assim durante um período de tempo a melhor rota para um destino, quando há mais de uma, pode variar conforme o resultado do cálculo do algoritmo, as rotas não são estáticas quando há mais de uma opção para um destino, esse dinamismo tende a melhor o desempenho da rede.

No cenário de teste, foi adicionado mais um nó a rede *mesh* o RASPI03, e com isso, agora tem-se no mínimo duas rotas para acesso a cada nó nesta arquitetura, como já observado na imagem a direita da Figura 3 pelas linhas tracejadas em verde.

Figura 4. Melhor rota para o nó *mesh*

```
cristiano@viking-laptop: ~
cristiano@viking-laptop:~$ tracepath www.terra.com.br
 1?: [LOCALHOST]                      pmtu 1500
 1:  _gateway                                              3.069ms
 1:  _gateway                                              2.655ms
 2:  192.168.0.13                                         17.294ms
 3:  192.168.0.1                                          24.650ms
 4:  10.72.73.129                                         71.728ms
 5:  10.72.73.1                                          304.782ms
 6:  10.72.21.54                                        1021.590ms
 6:  10.72.21.54                                         359.501ms
 7:  10.2.208.3                                           89.239ms asymm
  9
 8:  ^C
cristiano@viking-laptop:~$
```

Quando utilizado o comando *TRACEROUTE* no RASPI02(192.168.0.11) para testar o acesso à Internet e visualizar o caminho percorrido pela requisição ao *web site* pela rede *mesh*, o verificou-se que o caminho "escolhido" foi pela rota do nó RASPI03(192.168.0.13) e não pelo nó RASPI01(192.168.0.10), como observa-se na Figura 4 rota esta, definida pelo algoritmo do protocolo OLSR, o resultado do algoritmo pode ser visualizado através do número presente meio das linhas tracejadas em verde nas imagens da Figura 3, a rota de menor resultado será a escolhida.

A rede *mesh* implementada tem o objetivo de ser uma rede de segunda camada, fornecendo uma infraestrutura sem fio para os dispositivos conectados, provendo acesso e transporte de dados através de seus nós até seu destino.

5. Conclusão

A implantação de uma da rede *mesh* com o protocolo OLSR para o gerenciamento de rotas entre os nós da rede para criar uma MANET e assim fornecer uma infraestrutura sem fio, em segunda camada, ou seja, uma camada superior a rede LAN, um *backbone* para dispositivos IoT, resultou em uma rede capaz fornecer um infraestrutura sem fio, um *backbone*, para fornecer conexão a dispositivos *wireless*. O protocolo OLSR atendeu as expectativas para a criação da MANET, se mostrou eficiente no gerenciamento de rotas para os nós da rede *mesh*, requer pouca configuração para escalar a rede, que se autoconfigura na adição ou remoção dos *hosts*.

A utilização de *plugins* do OLSR podem oferecer ferramentas de monitoração e gerenciamento da da rede implantada, os quais, forneceram muitos dos resultados para este trabalho. Um protocolo multi arquitetura e multiplataforma que pode ser instalado em vários dispositivos computacionais, para formar uma rede decentralizada e autoconfigurável para conexão de dispositivos com e sem fio.Como trabalho futuro objetiva-se estender o cenário de testes avaliando o comportamento dos serviços de rede nessa infraestrutura observando o seu comportamento em caso de falha de algum nó da rede.

Referências

Boukerche, A. (2009). *Algorithms And Protocols For Wireless And Mobile Ad Hoc Networks*. Wiley, New Jersey, EUA.

Holt, A. and Huang, C.-Y. (2010). *802.11 Wireless Networks: Security and Analysis*. Springer, New York, EUA.

OLSR (Webpage). Optimized link state protocol.

Raspberry (Webpage). A small and affordable computer.

ZHANG, Y., LUO, J., and HU, H. (2007). *Wireless Mesh Networking, Architectures, Protocols and Standards*. Auerbach Publications, Boca Raton, EUA.