UNIVERSIDADE FEDERAL DE SANTA MARIA
COLÉGIO TÉCNICO INDUSTRIAL DE SANTA MARIA
CURSO SUPERIOR DE TECNOLOGIA EM REDES DE COMPUTADORES

COMUNICAÇÃO SEGURA EM UMA REDE MESH APLICADA A SMART GRIDS

TRABALHO DE CONCLUSÃO DE CURSO

Alexandre Silva Rodrigues

Santa Maria, RS, Brasil
2015

COMUNICAÇÃO SEGURA EM UMA REDE MESH APLICADA A SMART GRIDS

Alexandre Silva Rodrigues

Trabalho apresentado ao Curso de Graduação em Tecnologia em Redes de Computadores, Área de concentração em Segurança da Informação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Tecnólogo em Redes de Computadores.**

Orientador: Prof. Me. Tiago Antonio Rizzetti

Santa Maria, RS, Brasil

2015

Universidade Federal de Santa Maria
Colégio Técnico Industrial de Santa Maria
Curso Superior de Tecnologia em Redes de Computadores

A Comissão Examinadora, abaixo assinada,
aprova a Monografia

COMUNICAÇÃO SEGURA EM UMA REDE MESH ALICADA A SMART GRIDS

elaborada por
Alexandre Silva Rodrigues

como requisito parcial para obtenção do grau de
Tecnólogo em Redes de Computadores

COMISSÃO EXAMINADORA

Tiago Antonio Rizzetti, Me.
(Presidente/Orientador)

Alfredo Del Fabro Neto, Tecng.(UFSM)

Renato Preigschadt de Azevedo, Me.(UFSM)

Santa Maria, 03 de julho de 2015.

DEDICATÓRIA

Primeiramente, quero dedicar essa conquista ao meu pai (*in memoriam*). Sua ausência física jamais irá apagar todas as lembranças que tenho de quando estavas aqui. A cada dia que passa, a saudade aumenta e com ela, a certeza que sempre estarás presente em minha memória.

Quero dedicar a minha mãe, minhas irmãs e ao meu sobrinho. Essa vitória é de vocês também.

Dedico, também, a todas as pessoas que acreditaram em mim e de alguma forma, contribuíram para que eu chegasse até aqui.

AGRADECIMENTOS

Primeiramente, quero agradecer a minha mãe e minhas duas irmãs por estarem ao meu lado, dedicando-me muito amor, apoio e carinho em todos os momentos. Só tenho a agradecer por ter uma família tão maravilhosa. Vocês representam muito em minha vida.

Agradeço ao meu orientador Tiago Antonio Rizzetti, por todos os ensinamentos, pela disponibilidade que sempre tivestes para ajudar-me e pela oportunidade de fazer parte de projetos que contribuíram intensamente para minha formação acadêmica.

Agradeço aos colegas de IC (bolsistas da sala 302 do CTISM) pela amizade e disponibilidade para ajudar sempre que precisei.

Agradeço a todos professores do curso pelos ensinamentos.

Agradeço também a todos amigos e familiares que estiveram ao meu lado.

Muito obrigado a todos.

“The joy of life consists in the exercise of one's energies, continual growth, constant change, the enjoyment of every new experience. To stop means simply to die. The eternal mistake of mankind is to set up an attainable ideal.”

Aliester Crowley

RESUMO

Monografia

Curso Superior de Tecnologia em Redes de Computadores
Universidade Federal de Santa Maria

COMUNICAÇÃO SEGURA EM UMA REDE MESH APLICADA A SMART GRIDS

AUTOR: ALEXANDRE SILVA RODRIGUES
ORIENTADOR: TIAGO ANTONIO RIZZETTI
Data e Local da defesa: Santa Maria, 03 de julho de 2015.

A implementação de uma rede elétrica inteligente demanda uma rede de comunicação bidirecional para interação entre os diversos dispositivos do sistema e os sistemas de gerenciamento. Entretanto, a comunicação entre os dispositivos ativos no sistema é um grande desafio. Além de prover a comunicação entre um grande número de dispositivos, é necessário garantir a segurança das informações trafegadas. Nesse contexto, a utilização de redes mesh apresentam-se como uma alternativa promissora. A principal característica desse tipo de rede é a autoconfiguração da topologia e dispositivos ativos. Para isso, as tabelas de roteamento são trocadas entre os nós. Dessa forma, é necessário impedir que dispositivos não autorizados divulguem rotas ou consigam trafegar dados na rede. Nesse contexto, a autenticidade e a integridade das informações podem ser comprometidas por um falso nó. Entre as medidas de proteção que podem ser aplicadas, nesse caso, destaca-se a utilização de um mecanismo de autenticação de cada nó. Nesses termos, esse trabalho apresenta uma proposta para garantir a autenticidade dos nós participantes de uma rede mesh, através da assinatura das mensagens de controle do protocolo OLSR via middleware SECOM.

Palavras-chave: Redes Elétricas Inteligentes, Redes Mesh, Protocolo OLSR, Middleware SECOM, Segurança.

ABSTRACT

Monograph

Technology in Computer Networks Degree
Federal University of Santa Maria

SECURITY COMUNICATION IN A MESH NETWORK APLIED TO SMART GRIDS

AUTHOR: ALEXANDRE SILVA RODRIGUES
SUPERVISOR: TIAGO ANTONIO RIZZETTI
Defense Place and Date: Santa Maria, July 03, 2015.

The implementation of a smart grid demands a network of two-way communication for interaction between the several devices in the system and the management systems. However, communication between the active devices in the system is a big challenge. In addition to providing communication between a large numbers of devices, it is necessary ensure the security on transmitted information. In this context, the use of mesh networks is an interesting alternative. The main feature of this type of network is the auto configuration of the topology and the high dynamic on active devices. The routing infrastructure is implemented by these devices through exchange of routing tables between the nodes. Thus, it is necessary prevent unauthorized devices disclose routes or may transmit any data on the network. In this context, the authenticity and the integrity information's can be compromised by a false node. Among the protective measures that can be applied in this case highlights the use of authentication mechanism in each node. In these terms, this work presents a proposal to ensure the authenticity of the participating nodes of a mesh network, through signing the control messages of OLSR protocol via SECOM middleware.

Keyworks: Smart Grids, Mesh networks, Protocol OLSR, Middleware SECOM, Security.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

AODV	*Ad hoc On-Demand Distance Vector*
BATMAN	*Better Approach To Mobile Ad hoc Networking*
DoS	*Denial of Service*
ESSID	*Extended Service Set Identification*
HAN	*Home Area Network*
HNA	*Host and Network Association*
ID	Identificador
IEEE	*Institute of Electrical and Electronics Engineers*
IETF	*Internet Engineering Task Force*
INRIA	*Institute National de Recherche en Informatique et en Automatique*
IP	*Internet Protocol*
LAN	*Local Area Network*
MID	*Multiple Interface Declaration*
MPR	*Multipoint Relay*
OGM	*Originator Message*
OLSR	*Optimized Link State Routing Protocol*
RAN	*Regional Area Network*
RREP	*Route Reply*
RREQ	*Route Request*
SAODV	*Secure Ad hoc On-Demand Distance Vector*
SECOM	*Secure Communication Middleware*
SOLSR	*Secure Optimized Link State Routing Protocol*
TC	*Topology Control*
UDP	*User Datagram Protocol*
WAN	*Wide Area Network*

SUMÁRIO

1 INTRODUÇÃO

As redes elétricas inteligentes (*Smart Grid*) destacam-se por utilizarem tecnologias digitais para monitorar e controlar os dispositivos ativos do sistema elétrico. Para que isso seja possível, há necessidade de estabelecer uma rede de comunicação bidirecional entre os diversos dispositivos presentes na rede de energia com o sistema supervisório utilizado para gerenciá-la.

O sistema elétrico de potência apresenta uma diversidade de necessidades de rede. Em alguns sistemas, bem delimitados e geograficamente concisos, é possível utilizar comunicação através de meios cabeados confiáveis, por exemplo, fibra óptica. Sistemas de geração são um exemplo clássico onde isso é possível. No entanto, em sistemas de distribuição, em função da grande quantidade de dispositivos e de sua alta dispersão geográfica, a implementação dessa rede de comunicação torna-se uma tarefa mais complexa (GTREI, 2010).

Nesse contexto, diferentes tecnologias podem ser utilizadas para prover comunicação em determinados segmentos da rede de comunicação. Uma dessas tecnologias consiste nas redes mesh, em função da sua natureza dinâmica que permite a inserção, reconfiguração e saída de dispositivos da rede de forma frequente.

No entanto, a segurança da comunicação é um fator que pode trazer um grande impacto para uma rede de comunicação utilizada em uma *Smart Grid* (LOPES, 2012). Em função da criticidade do sistema, deve-se garantir que somente dispositivos autorizados efetuem as transmissões e recepção de dados nessa rede de comunicação (GTREI, 2010). Essa é uma prerrogativa muitas vezes negligenciada na concepção de protocolos para redes mesh (JUNIOR, 2003).

1.1 Objetivos

O presente trabalho tem como objetivo apresentar uma topologia, baseada em uma rede mesh, para realizar a comunicação segura em uma *Smart Grid*, entre concentradores e uma central de controle.

1.1.1 Objetivos específicos

Para possibilitar uma comunicação segura para as mensagens de controle de uma rede mesh, nesse trabalho, os seguintes objetivos específicos serão abordados:

- Implementar uma rede mesh, onde cada nó representa um elemento ativo no sistema elétrico;
- Analisar o protocolo e as tabelas de roteamento em cada nó;
- Simular a inserção de falsos nós na rede;
- Implementar uma forma de comunicação segura para a troca de mensagens de controle na rede mesh.

1.2 Motivação

Para que as *Smart Grids* possam vir a substituir as redes elétricas convencionais, existe ainda uma série de desafios. Nesse sentido, os principais aspectos a serem pesquisados e desenvolvidos são: a comunicação entre os equipamentos ativos na rede elétrica e a segurança das informações que serão trafegadas entre os consumidores e as concessionárias.

Nesse contexto, as informações de diversos clientes são coletadas por concentradores. Esses podem estar localizados em pontos geográficos distantes e de difícil acesso. Prover essa comunicação de forma segura é um grande desafio. Além disso, a rede utilizada nessa comunicação requer um elevado índice de disponibilidade, devido à necessidade de comunicação em tempo real entre os equipamentos.

1.3 Estruturação do trabalho

Este trabalho está estruturado da seguinte forma: o capítulo 2 apresenta uma visão geral sobre o conceito e as vantagens da substituição do sistema elétrico convencional por *Smart Grids*. O capítulo 3 aborda o conceito de redes mesh, suas vantagens, os principais

ataques e protocolos de roteamento para essa forma de comunicação. Ao final desse capítulo, será descrito o protocolo OLSR, o qual servirá de base para o desenvolvimento desse trabalho. O capítulo 4 apresenta uma proposta para tornar uma rede mesh mais segura através da assinatura das mensagens de controle do protocolo OLSR, por meio da utilização do *middleware* SECOM. No capítulo 5, será descrita a implementação dessa proposta. No capítulo 6, serão apresentados os testes e resultados obtidos para validar essa proposta. Além disso, serão apresentados testes realizados com o protocolo OLSR nativo e o SOLSR. No capítulo 7, serão analisados os resultados obtidos e apresentado um projeto a ser desenvolvido em um próximo trabalho.

2 SMART GRIDS

A energia elétrica é utilizada para os mais diversos fins, seja nas residências quanto nas indústrias. Em situações onde o seu fornecimento é interrompido, evidencia-se o quanto ela é importante e necessita de sistemas capazes de automatizar o processo de restabelecimento da mesma. Além disso, a relação entre as concessionárias de energia elétrica e seus clientes ainda é restrita. Para resolver essas questões, diversas tecnologias têm surgido para facilitar o processo de distribuição de energia elétrica.

Nesse contexto, destaca-se o conceito de redes elétricas inteligentes (*Smart Grids*), que apresentam uma série de vantagens em relação ao sistema convencional de energia elétrica, como por exemplo: interação entre dispositivos ativos no sistema elétrico de potência em tempo real, capacidade de autorrecuperação em casos de falhas no sistema, automatização e melhor gerenciamento dos processos de geração, distribuição e transporte da energia elétrica (WANG, 2011).

De acordo com RAMOS (2012), uma rede elétrica inteligente destaca-se por utilizar as tecnologias da informação para facilitar a administração e gerenciamento da rede elétrica convencional. As *Smart Grids* visam modernizar a rede elétrica, que ao longo dos anos, apresentou uma discreta evolução. A automação desse sistema possibilitará ações, em tempo real, em equipamentos presentes na geração até a distribuição da energia elétrica.

Uma das primeiras ações para tornar a rede elétrica inteligente é a utilização de *Smart Meters* (medidores inteligentes), que são capazes de comunicar-se com outros equipamentos, como por exemplo, enviar/receber dados para a concessionária de energia (LOPES, 2012). Dessa forma, além de registrarem o consumo de energia, esses medidores possibilitam ao consumidor final o controle sobre o seu consumo e obtenção de descontos na fatura em determinados horários. Com esses medidores, casos de falta de energia poderão ser detectados automaticamente pela concessionária, sem necessitar que o consumidor a notifique. Além disso, esses medidores são capazes de comunicar-se com outros equipamentos, como por exemplo: unidades de medição fasorial, relés inteligentes e dispositivos eletrônicos inteligentes instalados na rede elétrica (GTREI, 2010) (LOPES, 2012).

Outro aspecto importante, quando se trata de *Smart Grid*, é contingência em casos de falhas na rede de distribuição. Nesse contexto, aplica-se o conceito de *Self-Healing*, que pode ser visto como uma reconfiguração automática. Para isso, é realizado o monitoramento e

análise da rede, buscando possíveis falhas. Essa funcionalidade permite identificar a falha e o local da ocorrência e assim, facilitar o processo de restabelecimento de energia para os clientes.

Com relação à área de abrangência de uma *Smart Grid*, podemos dividir o sistema em camadas, conforme pode ser visualizado na Figura 1. (GTREI, 2010):

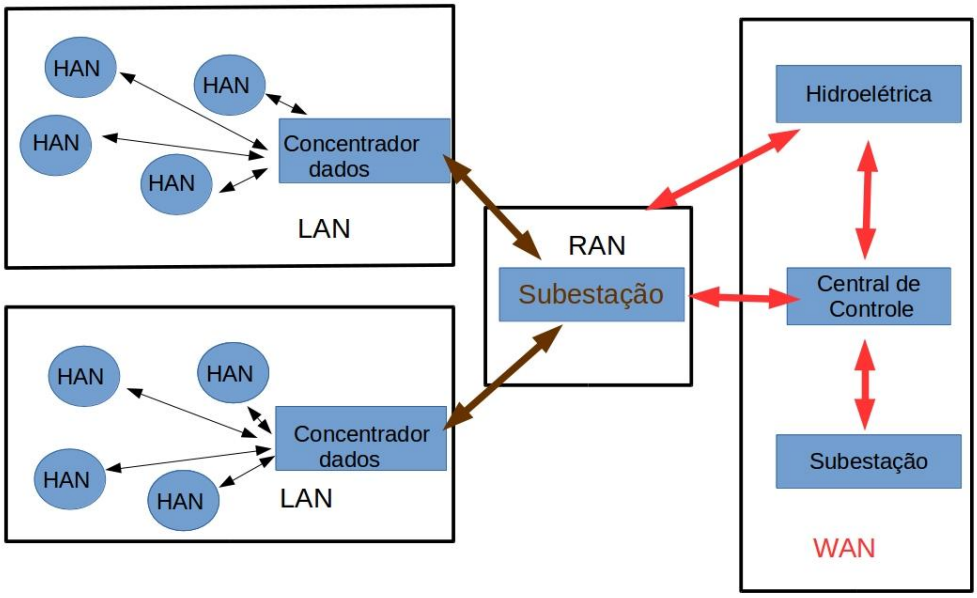

Figura 1: Camadas de uma Smart Grid, em relação a sua área de abrangência.
Fonte: Acervo Pessoal.

De acordo com a Figura 1, uma *Smart Grid* é dividida nas seguintes camadas (GTREI, 2010):

a) HAN (*Home Area Network*): compreende os dispositivos presentes na residência do cliente. Nessa camada destaca-se a utilização de medidores inteligentes. Esses são capazes de processar dados e enviar comandos para outros equipamentos. Com isso, é possível que o cliente possa ter um controle sobre o seu consumo de energia elétrica. Um exemplo disso consiste em desligar equipamentos em determinados horários em que a tarifa cobrada é mais cara. Nesse contexto, a concessionária poderá controlar a carga dentro de

cada residência, limitar a demanda e evitar sobrecargas na rede. Nessa camada, o medidor inteligente será capaz de interligar a residência ao restante da *Smart Grid*. As informações enviadas por ele serão recebidas por um concentrador de dados.

b) LAN (*Local Area Network*) e RAN (*Regional Area Network*): essas camadas são responsáveis por coletarem informações de diferentes concentradores. Nessas camadas existe uma maior preocupação com a segurança das informações e a disponibilidade da rede.

c) WAN (*Wide Area Network*): essa camada recebe informações de dispositivos espalhados em uma grande área geográfica. Por exemplo, vários concentradores de dados e dispositivos presentes em diversas subestações podem enviar informações para uma central de controle. A tecnologia a ser utilizada nessa camada deve ser altamente confiável, pois nela é trafegado um volume considerável de informações. A disponibilidade deve ser alta, pois, muitas operações de tempo real necessitam de um baixo tempo de latência.

2.1 Rede de comunicação de dados em uma Smart Grid

A rede a ser utilizada em uma *Smart Grid* deve permitir uma comunicação bidirecional entre os consumidores e as empresas que atuam na geração e distribuição da energia elétrica. Além disso, é importante ressaltar que essa comunicação necessita de um elevado índice de disponibilidade e confiabilidade, devido ao alto grau de criticidade das informações que nela podem trafegar. Nesse aspecto, podem ser utilizados diferentes tipos de tecnologias para levar informações de um ponto a outro (WANG, 2011) (EKANAYAKE, 2012).

Nesses termos, a alteração ou falsificação de uma informação pode causar grandes prejuízos ao sistema ou interrupção de importantes serviços oferecidos por ele. Dessa forma, em relação às informações trafegadas, os seguintes critérios necessitam ser observados (EKANAYAKE, 2012):

a) Confidencialidade: relaciona-se com a privacidade de uma informação. Dessa forma, apenas o emissor e o receptor têm acesso à mesma;

b) Autenticidade: refere-se à identidade do emissor de uma mensagem, ou seja, o receptor certifica-se que uma informação recebida não foi enviada por um impostor;

c) Integridade dos dados: garantia que uma informação não sofreu nenhuma modificação, ou seja, a informação que chegou ao receptor é exatamente igual a que foi enviada pelo receptor.

Para prover a comunicação entre os dispositivos ativos em uma *Smart Grid* é necessária a utilização de um padrão de rede que permita a interação entre os diversos dispositivos e protocolos de comunicação. Além disso, é necessário que todos os dispositivos sejam endereçados de forma única na rede. Para isso, a utilização de redes baseadas em IP destaca-se, em razão de sua consolidada utilização para as mais diversas aplicações.

Em razão do grande número de dispositivos e sua disposição geográfica, a preocupação com a topologia da rede é essencial. Esse aspecto pode influenciar diretamente em critérios críticos para este tipo de comunicação, tais como, latência, disponibilidade e segurança. Uma alternativa promissora para esse paradigma consiste em dividir esta rede de comunicação em sub-redes menores interligadas por concentradores, agindo como coletores de informações (GTREI, 2010). Devido à alta densidade de dispositivos e o constante crescimento de nós que a rede pode apresentar, as tecnologias de comunicação sem fios, através de redes mesh, são apresentados como uma alternativa interessante.

3 REDES MESH

Segundo ABELÉM (2007), uma rede mesh pode ser vista como uma rede com topologia dinâmica, variável e que pode ser facilmente expandida. Essa é constituída por diversos equipamentos (nós) interligados. A comunicação entre esses é realizada através do padrão IEEE 802.11s, formando uma grande rede sem fio (malha), utilizando múltiplos saltos para transmitir dados. Nesse caso, cada nó pode atuar como roteador e prover acesso a uma rede externa (nó funcionará como *gateway* para o demais nós) (CARDOSO, 2012).

Dessa forma, o sinal de um nó é replicado pelos demais nós pertencentes a uma rede, possibilitando a escolha do melhor caminho ou rotas alternativas para encaminhar um pacote (MACHADO, 2013). Assim, uma rede mesh pode ser utilizada em regiões geográficas extensas ou que apresentam determinada dificuldade de acesso. Um exemplo disso, é em regiões que apresentam montanhas ou prédios que dificultam a propagação do sinal sem fio de uma rede estruturada. Nesse caso, é possível criar rotas alternativas entre um nó emissor e o seu destino (ZHANG, 2006). A Figura 2 apresenta a topologia de uma rede mesh. Nela podemos visualizar a forma como os nós são interligados, possibilitando que diferentes rotas possam ser utilizadas para que um determinado nó possa acessar a rede externa (Internet), através de múltiplos *gateways*, que aumentam a disponibilidade da rede.

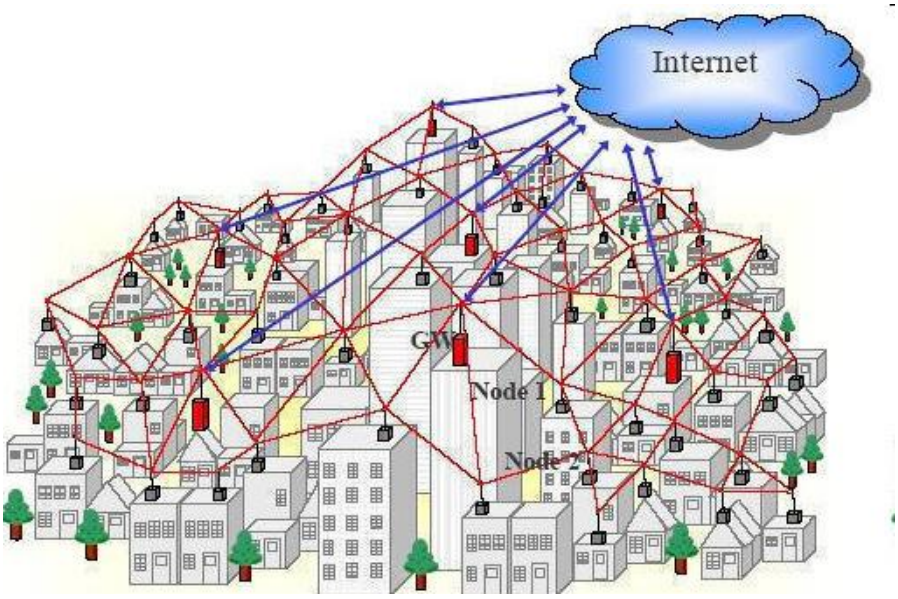

Figura 2: Topologia de uma rede mesh.
Fonte: (ZUCCHI, 2006).

Outro aspecto importante, em relação a esse tipo de rede, é autoconfiguração da topologia e descoberta dos nós ativos na rede. Dessa forma, um nó consegue ingressar na rede automaticamente, sem a necessidade de uma configuração manual. Além disso, se um nó apresentar uma falha e ficar indisponível, a rede cria rotas alternativas automaticamente sem afetar a disponibilidade da mesma, conforme pode ser visualizado na Figura 2 (MACHADO, 2013).

Nesses termos, uma rede mesh pode ser utilizada, com baixo custo e facilidade de implementação, em diversas aplicações, tais como (CARDOSO, 2012):

a) Conectar dispositivos localizados em regiões distantes e de difícil acesso, onde uma é inviável implementar uma rede que utilize meios físicos guiados, como por exemplo: cabo par trançado ou fibra óptica;
b) Desenvolvimento de cidades digitais, nas quais o sinal deve ser propagado em uma grande área geográfica, por exemplo: campus de uma universidade, hospital ou um bairro;
c) Comunicação entre os dispositivos ativos no sistema elétrico para implementação de uma *Smart Grid*.

3.1 Protocolos de roteamento para redes mesh e segurança nativa

Em razão do crescente interesse de estudo e implementação de redes mesh, diversos protocolos de roteamento surgiram. Esses protocolos podem ser divididos em três grupos (GRAAUD, 2011) (BOWITZ, 2011):

a) Protocolos pró-ativos: realizam atualização constante da rede, através da troca mensagens de controle. Dessa forma, esse protocolo reconhece uma possível modificação na topologia da rede, como por exemplo, a inclusão ou a indisponibilidade de um nó;
b) Protocolos reativos: criam rotas apenas quando um nó emissor precisar realizar a comunicação com um nó destino. Dessa forma, as mudanças na topologia da rede demoram um maior tempo para ser detectada;
c) Protocolos híbridos: mesclam as características dos protocolos descritos anteriormente.

Entre os protocolos mais utilizados, na implementação de redes mesh, destacam-se os seguintes:

a) AODV (*Ad hoc On-Demand Distance Vector*): é um protocolo de roteamento reativo, que utiliza um número de sequência para garantir que a rota utilizada está atualizada. Para realizar a descoberta da rota para um determinado nó da rede, o nó emissor envia uma mensagem RREQ em *broadcast*. Dessa forma, todos nós receberão a mensagem. Quando um nó recebe essa mensagem, ele estabelece uma rota reversa para o nó emissor e envia um pacote de RREP em *unicast* (apenas os nós que fazem parte da rota até chegar ao nó solicitante) (PERKINS, 1999).
b) BATMAN (*Better Approach To Mobile Ad hoc Networking*): é um protocolo pró-ativo, onde cada nó conhece apenas um vizinho (próximo salto). Para obter a rota ideal, um nó calcula o número de mensagens recebidas de cada nó e o último remetente. Cada nó envia uma mensagem (OGM) para notificar os vizinhos sobre sua presença na rede, os quais retransmitem, apenas uma vez, essa mensagem se o nó emissor for considerado o melhor salto (BOWITZ, 2011).
c) OLSR (*Optimized Link State Routing Protocol*): é um protocolo pró-ativo que utiliza o estado do enlace para selecionar suas rotas, onde é realizada a verificação dos nós que estão ativos para a construção da tabela de roteamento (PERKINS, 1999). Em razão dessa característica e sua larga utilização para implementar redes mesh, esse protocolo foi escolhido para o desenvolvimento desse trabalho.

Esses protocolos são muito utilizados na implementação de redes mesh comunitárias e não apresentam falhas significativas de desempenho e eficiência. Entretanto, não apresentam mecanismos para garantir uma comunicação segura. Nesses termos, o desenvolvimento de extensões e *plugins* de segurança para tais protocolos desperta grande interesse em pesquisadores e desenvolvedores.

Uma dessas iniciativas é uma extensão para o protocolo AODV chamada SAODV (*Secure* AODV) que adiciona campos de assinaturas digitais e *hash* nos pacotes do protocolo AODV. O objetivo dessas assinaturas é garantir a autenticidade do emissor de uma mensagem e aderir maior segurança na descoberta de rotas (ZAPATA, 2002).

Para o protocolo BATMAN, uma das soluções apresentadas é a utilização de certificados *proxy*, onde cada cliente da rede utiliza um certificado assinado por uma estação responsável pelo gerenciamento da rede. Essa proposta tem como premissa, a confiança entre os nós, ou seja, um nó encaminha suas próprias mensagens de controle ou dos nós que ele confia. Caso um nó, considerado como confiável na rede, apresente um comportamento

malicioso, ele poderá afetar toda a rede, visto que a integridade das mensagens não é verificada (BOWITZ, 2011).

O protocolo OLSR também apresenta um *plugin* de segurança denominado *Secure* OLSR (SOLSR), que utiliza uma chave simétrica de 128 bits para assinar as mensagens de controle do protocolo. Essa chave deve ser conhecida por todos os nós da rede. As mensagens são assinadas a cada salto, ou seja, não é possível estabelecer uma autenticação entre o emissor e destinatário. Dessa forma, é necessário que um nó confie em seus vizinhos (HAFSLUND, 2004).

Em termos de segurança contra ataques externos, essa abordagem contribui para manter a rede segura, visto que apenas os nós que conhecem a chave secreta podem enviar mensagens de controle do protocolo. Entretanto, se um nó participante da rede divulgar a chave utilizada a um nó malicioso, a segurança de toda rede está vulnerável. Além disso, se não existir um sistema eficiente de gerenciamento da chave utilizada que permita que ela seja atualizada frequentemente, através de um ataque de força bruta ela pode ser descoberta (FERNANDES, 2006).

3.2 Tipos de ataques em redes mesh

Em razão da utilização de meios de comunicação sem fio e da falta de mecanismos de segurança nos protocolos de roteamento, a segurança é um dos principais problemas para tornar viável a implementação de uma rede mesh. Essa pode ser exposta a uma série de ataques. Esses ataques podem ser praticados por atacantes externos, para vasculhar e modificar as informações trafegadas na rede. Além disso, um nó participante da rede pode agir de forma maliciosa, podendo ocasionar consequências mais graves (FERNANDES, 2006).

Esses ataques podem ser inativos, quando o objetivo é apenas vasculhar o trafego de dados de uma rede ou ativos, quando o objetivo é injetar, modificar ou descartar os dados trafegados na rede (SEN, 2013). Para ter acesso a esses dados, o atacante pode explorar vulnerabilidades nos protocolos de roteamento utilizados na rede. Em relação às tabelas de roteamento, o atacante pode utilizar a ausência de mecanismos de autenticação em uma rede para divulgar falsas informações sobre a topologia da rede.

Nesse contexto, a inserção de falsos pacotes, nos quais são inclusas mensagens de controle, na rede pode ocasionar um estouro na tabela de roteamento de um dispositivo ou alteração nas rotas verdadeiras, para um dispositivo intermediário, entre o emissor e o receptor de um dado. Dessa forma, o atacante pode afetar a confidencialidade da rede e obter informações privadas. Além disso, esses dados podem ser modificados antes de chegarem ao seu destino (FERNANDES, 2006).

Outro aspecto importante é a disponibilidade da rede que pode ser afetada, através da criação de buracos negros na rede. Nesse aspecto, uma falsa rota é divulgada na rede. A origem dessa rota pode ser um nó malicioso que descarta todos os pacotes recebidos (SEN, 2013).

Além disso, o atacante pode utilizar complexos sistemas computacionais para indisponibilizar os serviços ou recursos de um dispositivo ou rede. Nesse contexto, uma técnica denominada DoS é utilizada para realizar essa ação. Esse tipo de ataque pode ocasionar sérias consequências em uma rede que necessita de um nível alto de disponibilidade.

Dentre os protocolos para redes mesh, descritos anteriormente, neste trabalho utiliza-se como base o protocolo OLSR, sendo por isso descrito detalhadamente na seção a seguir.

3.3 Protocolo OLSR

De acordo com SILVA (2011), o protocolo OLSR foi desenvolvido pelo INRIA (*Institute National de Recherche en Informatique et en Automatique*) e padronizado pela IETF (*Internet Engineering Task Force)*, em caráter experimental, na RFC 3626. Esse protocolo foi criado para ser utilizado em grandes redes Ad Hoc, calculando e mantendo rotas para todos os dispositivos de uma rede construída sob uma topologia em malha.

Por ser um protocolo pró-ativo, o OLSR tem como característica principal, a atualização constante das tabelas de roteamento de todos os nós participantes da rede. Dessa forma, qualquer modificação na topologia da rede é detectada automaticamente. Nesse contexto, o protocolo OLSR utiliza mensagens de controle para realizar a descoberta de nós na rede e atualização da topologia (SILVA, 2011). Essas mensagens são enviadas em *broadcast* através da porta UDP 698 (CLAUSEN, 2003).

De acordo com CLAUSEN (2003) e FERNANDES (2006), as seguintes mensagens são essenciais para o funcionamento do protocolo OLSR:

a) *HELLO*: utilizada para realizar a escolha de um MPR e a descoberta de enlaces e vizinhos. Essa mensagem é enviada apenas para os vizinhos e não deve ser encaminhada para os vizinhos distantes a mais de um salto e apresenta os nós com os quais um nó possui uma comunicação. A Figura 3 apresenta o formato dessa mensagem, onde podemos visualizar o campo que define o intervalo de emissão das mensagens *HELLO* (*Htime*), o campo que específica o endereço IP de um nó (*Neighbor Interface Address*) e o campo que apresenta informações sobre o enlace entre a interface do remetente e os vizinhos especificados na mensagem;

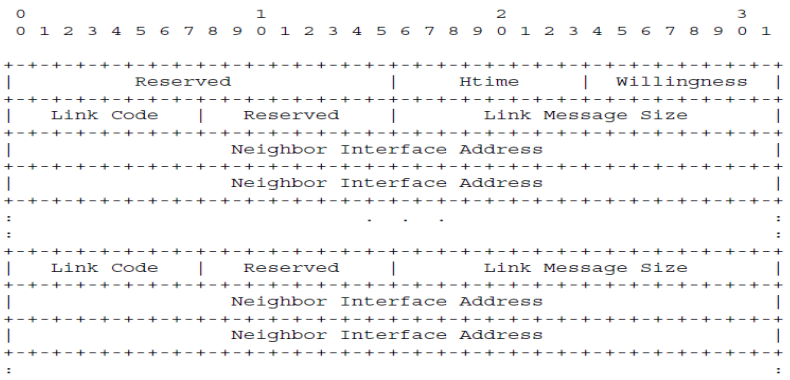

Figura 3: Mensagem HELLO.
Fonte: (CLAUSEN, 2003).

b) TC: é utilizada para realizar o controle da topologia da rede. Essa mensagem é composta por uma lista de nós que selecionaram o nó emissor dessa mensagem como MPR e enviada para toda a rede. Dessa forma, é possível que um nó não envie essa mensagem, em razão não ter sido escolhido como MPR. As informações recebidas através dessa mensagem são armazenadas na tabela de roteamento de cada nó, a qual é utilizada para realizar o cálculo de rota para enviar um pacote a um determinado destino. A Figura 4 apresenta o formato da mensagem TC, onde podemos visualizar os seguintes campos: endereço IP principal dos vizinhos do nó emissor da mensagem (*Advetised Neighbor Main Address*) e um número de sequência para comparar se uma mensagem recebida, com atualização sobre um vizinho, é mais recente que do que a informação que o nó possui;

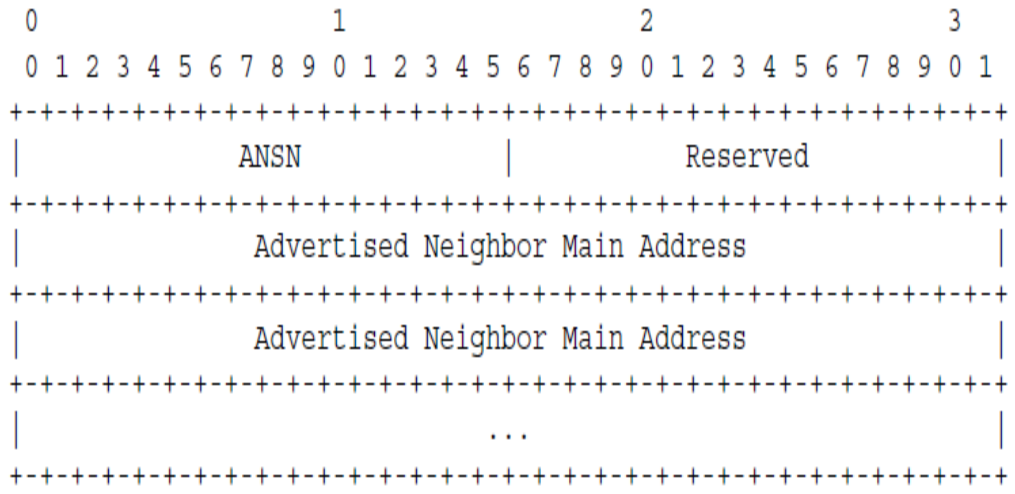

Figura 4: Mensagem TC.
Fonte: (CLAUSEN, 2003).

c) MID: essa mensagem é enviada para toda a rede e utilizada para declarar múltiplas interfaces em um nó. Através dessa mensagem, é possível associar diversas interfaces de um dispositivo no cálculo de rotas. A Figura 5 apresenta o formato da mensagem MID, onde podemos visualizar os campos que possuem os endereços IP de cada interface que o nó possui;

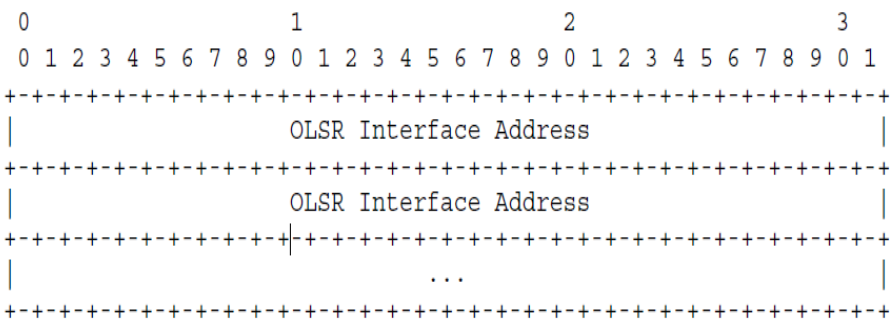

Figura 5: Mensagem MID.
Fonte: (CLAUSEN, 2003).

d) HNA: contém informações sobre os anúncios das redes de um nó, ou seja, um nó apresenta-se como *gateway* para o acesso de uma determinada rede. Na Figura 6,

podemos visualizar o formato da mensagem HNA, onde é inserido o endereço e a máscara de cada rede que um nó deseja anunciar.

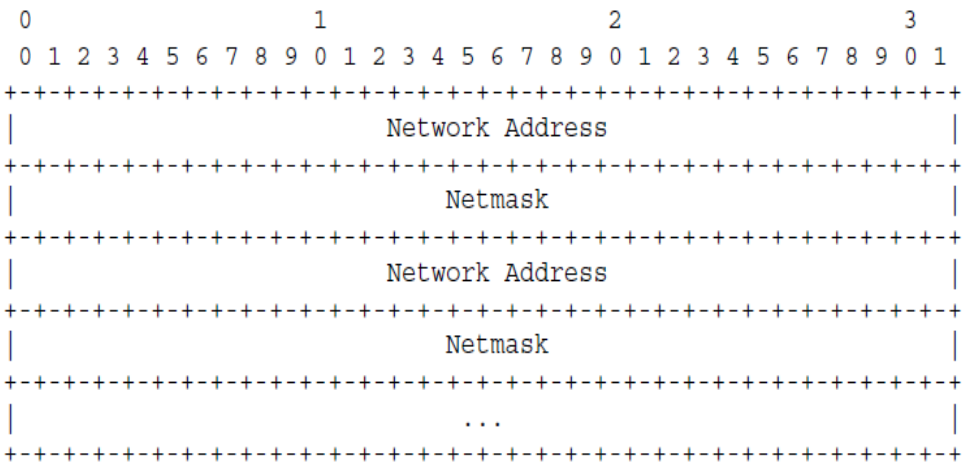

Figura 6: Mensagem HNA.
Fonte: (CLAUSEN, 2003).

Dessa forma, um grande número de mensagens de controle é trafegado na rede. Para evitar um *overhead* na rede, é aplicado o conceito de MPR, que permite controlar a inundação em cada dispositivo (SILVA, 2011). Com a utilização de MPR, cada nó seleciona um conjunto de nós para retransmitir suas mensagens. Para isso, esses nós precisam ser seus vizinhos a um salto de distância e que os nós distantes a dois saltos possam ser alcançados por esses. Cada MPR é responsável por enviar e controlar o tráfego das mensagens de controle, e assim, evitar a inundação da rede com informações redundantes (CLAUSEN, 2003).

3.3.1 Formato dos pacotes OLSR

Conforme CLAUSEN (2003), o protocolo OLSR utiliza um formato de pacote unificado para todas as informações relacionadas ao protocolo. Dessa forma, cada pode encapsular mais de uma mensagem em um mesmo pacote. Essas mensagens compartilham um formato de cabeçalho comum. A Figura 7 apresenta o formato de um pacote OLSR. Nela podemos visualizar o cabeçalho do pacote e o encapsulamento de mensagens.

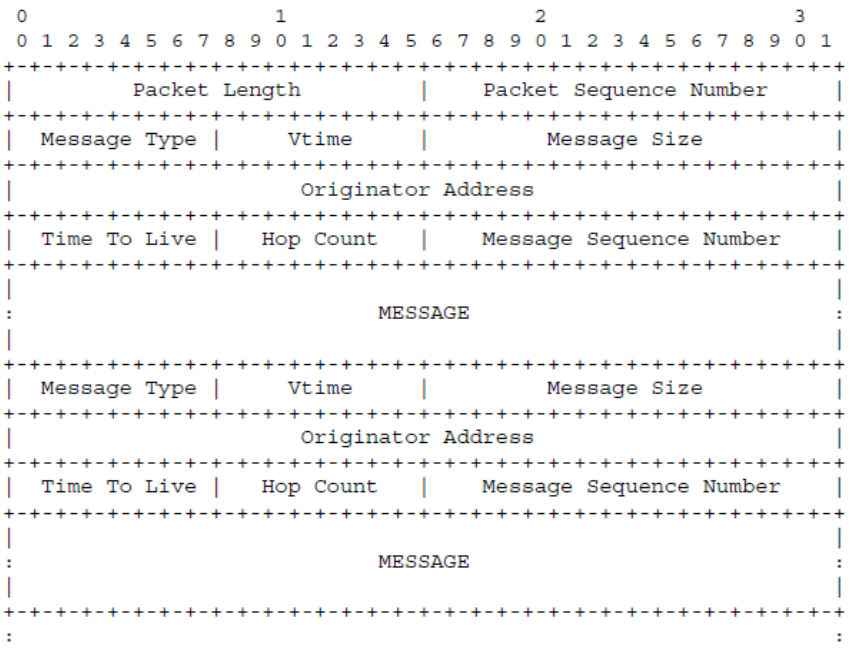

Figura 7: Pacote OLSR.
Fonte: (CLAUSEN, 2003).

Conforme pode ser visto na Figura 7, o cabeçalho do pacote OLSR possui as seguintes informações (CLAUSEN, 2003):

- *PacketLength*: informa o tamanho do pacote, em *bytes*;
- *PacketSequenceNumber*: esse número é incrementado a cada mensagem que um nó envia.

  Além do cabeçalho do pacote, cada mensagem encapsulada em um mesmo pacote OLSR, apresenta um cabeçalho, onde são inseridos os seguintes campos:
- *MessageType*: é adicionado um número para identificar o tipo de mensagem;
- *Vtime*: indica o tempo de validade de uma mensagem recebida;
- *MessageSize*: indica o tamanho da mensagem, incluindo o seu cabeçalho;
- *OriginatorAddress*: informar o endereço IP do emissor da mensagem;
- Time to Live: expressa o número de saltos que a mensagem pode ser encaminhada. A cada salto esse valor é decrementado;
- *Hop Count*: informa quantos saltos foram realizados até a mensagem chegar ao seu destino. Esse valor é incrementado a cada salto;
- *MessageSequenceNumber*: a cada mensagem enviada, esse valor é incrementado.

3.3.2 Descoberta de vizinhos e enlaces

Para realizar a descoberta de vizinhos, o protocolo OLSR utiliza a mensagem *HELLO*, onde cada nó envia essa mensagem para seus vizinhos distantes a um salto.

Através dessa mensagem, é possível verificar o estado de cada enlace, ou seja, se existe uma comunicação unidirecional ou bidirecional entre dois dispositivos. Nesses termos, se um nó A receber uma mensagem *HELLO* vazia do nó B, esse será inserido em sua mensagem HELLO e considerado como um enlace unidirecional, visto que o seu endereço (nó A) não foi encontrado na mensagem. Então o nó A envia essa mensagem para seus vizinhos. Se o nó B receber essa mensagem, esse insere o nó A em sua mensagem HELLO, que será enviada para seus vizinhos. Dessa forma, existe uma comunicação bidirecional entre ambos.

3.3.3 Cálculo de rotas

De acordo com CLAUSEN (2003), para realizar o cálculo de rotas, o protocolo OLSR utiliza as informações contidas na tabela da topologia da rede, a qual é atualizada após o recebimento de uma nova mensagem TC. Essa atualização permite que a topologia esteja sempre atualizada e que qualquer alteração na rede possa ser percebida rapidamente.

Com base nas informações sobre a topologia da rede, cada nó mantém rotas para todos os outros dispositivos participantes da rede. Quando for necessário enviar um pacote para determinado destino, o nó realiza uma busca em sua tabela de roteamento para verificar para qual nó o pacote deve ser repassado. Nesse caso, o próximo nó ao receber o pacote, irá repassar a um vizinho. Esse processo é realizado até que o destino seja alcançado. Nesses termos, cada nó sempre busca o menor caminho para encaminhar um pacote, visto que o protocolo OLSR utiliza o algoritmo Dijkstra, para calcular as rotas (FERNANDES, 2006).

3.3.4 Controle da topologia

Ao receber uma mensagem TC, um nó armazena as informações recebidas em sua tabela de topologia. Esse procedimento é realizado nas seguintes situações: se não existir informações na tabela de topologia para o endereço IP do remetente da mensagem ou se a informação possui um ANSN maior, ou seja, a mensagem TC que havia sido processada está desatualizada (ARAÚJO, 2010).

Na tabela de topologia são armazenados os endereços IP de todos os nós inseridos nas mensagens TC recebidas. Além disso, é armazenado o endereço IP do último nó que deve ser acessado para chegar ao dispositivo inserido na mensagem, ou seja, o nó que enviou ela.

3.3.5 Redes externas

Quando um nó deseja anunciar uma rede para seus vizinhos, ele utiliza a mensagem HNA para realizar essa ação. Quando um nó receber essa mensagem, ele acrescenta as seguintes informações à sua tabela de roteamento: endereço de rede e o *gateway*. Essas informações são obtidas através do campo *Neighbor Address Network* e o endereço IP do emissor da mensagem HNA, respectivamente. Dessa forma, quando um nó confia nas mensagens de seus vizinhos, qualquer anúncio de redes será aceito, o que pode ocasionar graves consequências na sua tabela de roteamento.

4 PROPOSTA DE UMA REDE MESH SEGURA

Com o objetivo de tornar possível a utilização de redes mesh em aplicações criticas (*Smart Grids*, por exemplo) e devido às vulnerabilidades inerentes a esta forma de comunicação, esse trabalho apresenta uma proposta que visa agregar segurança e confiabilidade a esse tipo de rede. Basicamente, essa proposta consiste em uma modificação no protocolo OLSR para possibilitar a autenticação dos nós que desejam participar de uma rede mesh.

Nesses termos, essa proposta aplica-se no controle e gerenciamento de dispositivos na rede, ou seja, apenas os pacotes que contenham mensagens de controle do protocolo OLSR serão submetidos à análise de autenticidade e integridade de seu conteúdo. Dessa forma, os dados trafegados, os quais não são originados pelo protocolo de roteamento, entre os dispositivos participantes da rede não serão abordados nesse trabalho.

Para isso, é utilizado o *middleware* SECOM, que será responsável por gerar uma assinatura para cada mensagem de controle, as quais são trocadas entre os dispositivos da rede. Além disso, esse é responsável por verificar se a assinatura adicionada a um pacote OLSR é válida, ou seja, se o pacote foi gerado por um determinado nó e não teve seu conteúdo original alterado. Dessa forma são utilizadas as seguintes funções disponibilizadas por esse *middleware*:

- Assinatura_assinar: recebe um pacote e gera uma assinatura pare ele, a qual é retornada para o protocolo OLSR. Essa função é utilizada antes de enviar um pacote com mensagens de controle na rede;
- Assinatura_verificar: essa função é utilizada cada vez que um nó receber um pacote com mensagens de controle do protocolo OLSR. Ao chamar essa função, são enviados os seguintes parâmetros: o conteúdo original do pacote, a assinatura gerada para ele e o endereço IP do emissor do pacote. Com essas informações, é verificada se a assinatura confere com o conteúdo do pacote. Em caso de sucesso, o *middleware* responde que o pacote recebido é autêntico e as informações contidas nele são integras. Caso contrário, a resposta informará ao protocolo, que o pacote recebido não deve ser aceito, ou seja, ele deve ser descartado.

4.1 Middleware SECOM

O *middleware* SECOM foi desenvolvido para ser utilizado nas mais diferentes aplicações, visando garantir os parâmetros de autenticidade, integridade e confidencialidade a uma comunicação. Dessa forma, ele implementa uma infraestrutura de chaves assimétricas, controladas por uma entidade central, onde para qualquer comunicação, a aplicação poderá utilizar funções do sistema que proveem os serviços de segurança necessários.

4.1.1 Estrutura do middleware SECOM

O middleware SECOM tem como base um servidor de chaves, o qual é responsável por conhecer e armazenar informações sobre todos os dispositivos autorizados a participar de uma rede. Além disso, ele é responsável por autenticar os dispositivos da rede. Dessa forma, apenas dispositivos devidamente autenticados pelo servidor podem se comunicar.

O servidor de chaves possui um par de chaves assimétricas para utilizá-las na comunicação com os dispositivos. Cada dispositivo cliente deve ter posse de um par de chaves provisórias, uma pública e outra privada, além de já possuir a chave pública do servidor. Para essa implementação, utilizou-se o algoritmo assimétrico mais popular atualmente, conhecido como RSA (MOLIN, 2013).

Além das chaves, os dispositivos devem possuir um identificador único de tamanho fixo, criado aleatoriamente, que servirá para identificar o dispositivo, no momento da autenticação com o servidor. Esse identificador (ID) gerado deve ser de conhecimento exclusivo do dispositivo e do servidor de chaves (SILVA, 2015).

A Figura 8 representa a arquitetura do sistema, onde se tem:

a) *Server*: serviço responsável por gerar, manter e distribuir as chaves assimétricas de cada dispositivo autorizado da rede. Cada novo dispositivo ao ingressar na rede, deverá ter um cadastro previamente realizado neste servidor, utilizando um identificador único gerado para cada dispositivo pelo *daemon* do sistema instalado nos nós.

b) *Daemon*: *software* que deverá ser executado em todos os dispositivos da rede. Ele será responsável por manter, localmente, as chaves públicas dos demais

dispositivos que irão se comunicar com o dispositivo onde ele está executando. O serviço *daemon* mantém uma estrutura de cache para minimizar as consultadas ao servidor de chaves. Desta forma, a busca da chave pública do dispositivo com que deseja se comunicar é realizada somente na primeira vez, após é utilizada a cópia já existente na cache.

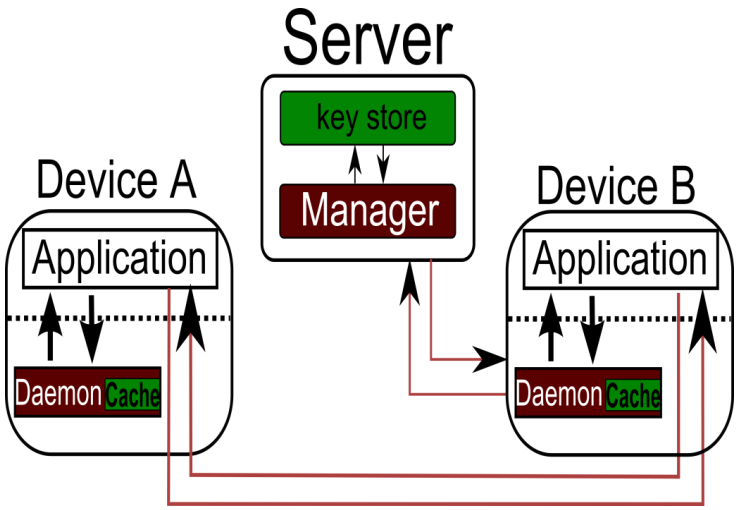

Figura 8: Estrutura do middleware SECOM.
Fonte: (SILVA, 2015).

4.1.2 Autenticação

A autenticação do dispositivo será feita utilizando criptografia assimétrica e assinatura digital. A primeira etapa inicia com o cliente enviando uma requisição de autenticação para o servidor de chaves. Essa requisição deve possuir o ID único do dispositivo e a chave pública do dispositivo, criptografados com a chave pública do servidor para garantir que apenas ele tomará posse desse ID.

Na segunda etapa, com o servidor já de posse do ID e de sua chave publica provisória do dispositivo, é realizada uma busca no banco de chaves do servidor, por alguma referência ao dispositivo solicitante. Caso o ID não se encontre entre os cadastrados, a conexão é encerrada e o dispositivo não é autenticado. Porém no caso do servidor encontrar o ID, o mesmo gerará um novo par de chaves para o dispositivo solicitante que será criptografado com a chave publica provisória deste, para que então possa ser enviado a ele. Após a

confirmação de recebimento, por parte do cliente, o processo de autenticação e a conexão são encerrados.

4.1.3 Comunicação entre os dispositivos

Sempre que um dispositivo deseja comunicar-se com outro dispositivo da rede, o mesmo deverá ter efetuado previamente o processo de autenticação com o servidor. Devidamente autenticado, o dispositivo que desejar se comunicar com outro dispositivo da rede, do qual ainda não possui a chave, deverá enviar uma requisição de chaves para o servidor. Esta requisição deve possuir o endereço IP do dispositivo desejado, juntamente com uma assinatura, provando ser um dispositivo válido da rede (SILVA, 2015).

O Servidor deverá enviar a chave pública do dispositivo de interesse para o dispositivo requisitante, devidamente assinada. Dessa forma o dispositivo emissor criptografará sua mensagem com a chave pública do dispositivo de destino para então enviá-la. O dispositivo de destino deverá efetuar o mesmo processo de pedido de chaves, caso ainda não possua a chave pública do emissor, para responder as mensagens originadas do mesmo. Para que esse processo não tenha que ser repetido com tanta frequência, será mantida uma tabela em cada dispositivo, com os dispositivos já contatados e suas respectivas chaves, durante um período de tempo.

4.2 Utilização do *middleware* SECOM para prover segurança a redes mesh

Em razão da importância das mensagens de controle utilizadas pelo protocolo OLSR, para realizar a descoberta de vizinhos e construção da topologia da rede, essas precisam ser protegidas contra ações de nós maliciosos. Entretanto, o protocolo OLSR não apresenta mecanismos eficientes para impedir que nós não autorizados possam realizar ações indevidas em uma determinada rede.

Nesses termos, um falso nó pode injetar informações na rede e obter livre acesso à mesma. Além disso, ele pode divulgar falsas redes, interceptar o tráfego de dados e afetar a integridades das informações trafegadas. Para resolver essas questões, a autenticação entre os

nós é imprescindível. Em razão da inexistência de um mecanismo eficiente de segurança nesse protocolo, será utilizado o *middleware* SECOM para realizar a assinatura de todos os pacotes do protocolo OLSR que contenham mensagens de controle.

Dessa forma, o pacote OLSR enviado por um nó é demonstrado na Figura 9. Nela podemos visualizar a inserção de um campo extra no final do pacote. Nesse campo é adicionada a assinatura, que contém 173 *bytes*.

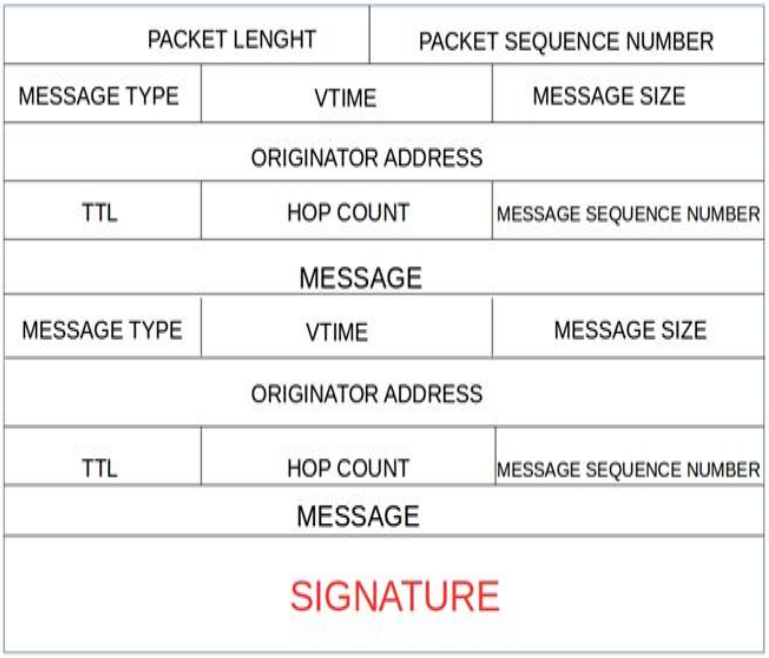

Figura 9: Pacote OLSR com adição do campo extra.
Fonte: Acervo Pessoal.

A Figura 10 apresenta de forma simplificada o processo de adição e verificação de uma assinatura em um pacote do protocolo OLSR. Nela, podemos visualizar o pacote OLSR modificado (pacote original e a assinatura correspondente ao seu conteúdo) sendo enviado por um nó A e recebido pelo nó B. Além disso, é possível os parâmetros passados para as funções disponíveis no *middleware* SECOM (*daemon*) e as respostas que retornam para o protocolo OLSR.

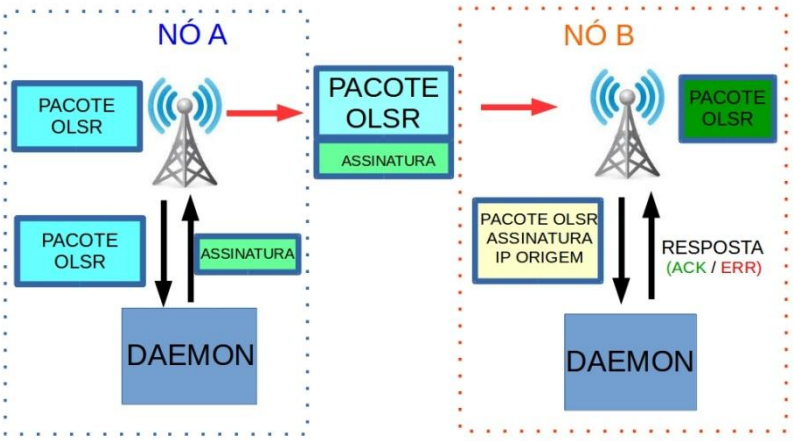

Figura 10: Processo de adição e verificação de uma assinatura nos pacotes OLSR.
Fonte: Acervo Pessoal.

5 IMPLEMENTAÇÃO DE UMA REDE MESH SEGURA

Para realizar a adição e verificação de uma assinatura em seus pacotes originais, o código-fonte do protocolo OLSR foi modificado. Dessa forma, foi inserida uma assinatura digital no *payload* do pacote OLSR. Essa modificação foi implementada para dispositivos que executam o protocolo OLSR no sistema operacional Linux. Para isso foi alterado o arquivo responsável por controlar a entrada e saída de dados das interfaces de rede do *kernel* do sistema operacional. Para realizar a comunicação com o *middleware* SECOM, é utilizado um *socket* que realiza uma comunicação local entre ambos, visto que o *middleware* é executado localmente em cada dispositivo.

Dessa forma, após o protocolo OLSR construir um pacote contendo mensagens de controle, esse é enviado para o *middleware* SECOM. A Figura 11 mostra a chamada à função do *middleware* SECOM que realiza a assinatura de pacote. Nela, o pacote OLSR foi codificado em Base64.

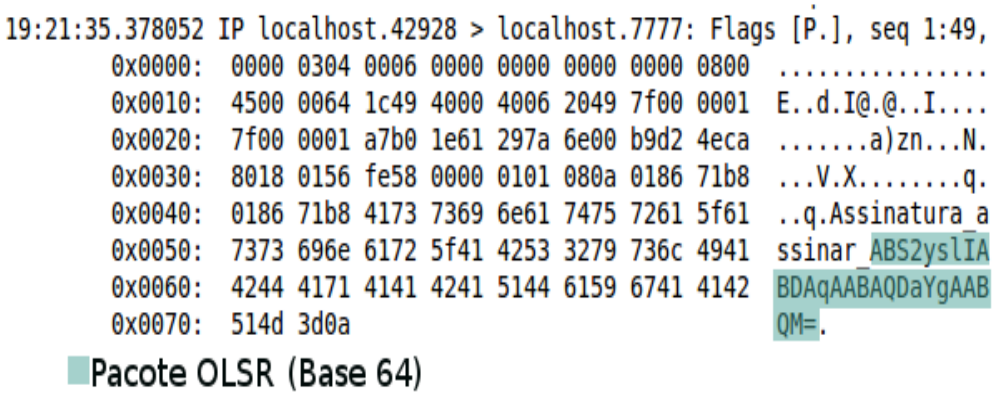

Figura 11: Solicitação de assinatura de um pacote OLSR.
Fonte: Acervo Pessoal.

Após gerar uma assinatura, essa é enviada como parâmetro de resposta para o protocolo OLSR. Essa assinatura contém um tamanho fixo de 173 *bytes*. A Figura 12 apresenta a assinatura gerada para o pacote mostrado na Figura 11.

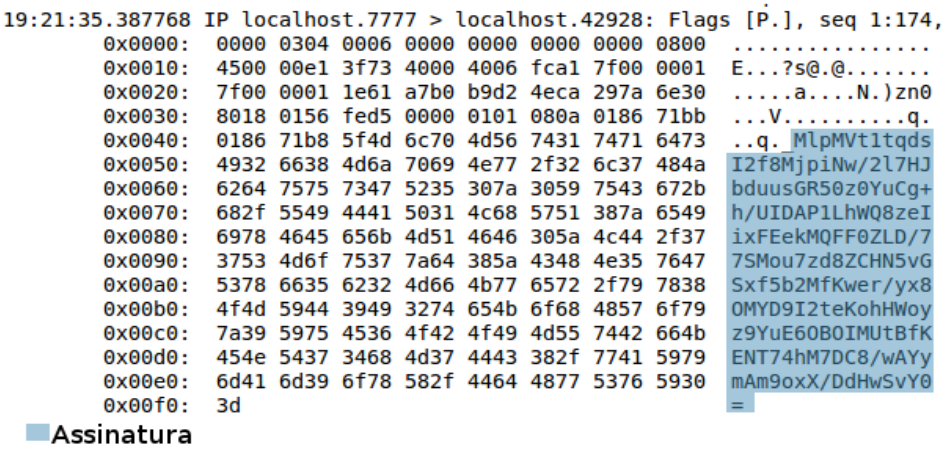

Figura 12: Assinatura de um pacote OLSR, gerada pelo middleware SECOM.
Fonte: Acervo Pessoal.

Ao recebê-la, o protocolo OLSR realiza a construção do pacote que será enviado em *broadcast*, ou seja, adiciona a assinatura ao final do pacote original. Então, o pacote é repassado para a interface responsável por enviá-lo, ou seja, o protocolo OLSR continua com o seu processo de inundação da rede com suas mensagens de controle. A Figura 13 possibilita visualizar o formato do pacote OLSR que será enviado para a rede e apresenta a origem do pacote (endereço IP 192.168.0.1) sendo enviado em *broadcast* (endereço IP 255.255.255.255) através da porta 698 (definida como padrão para o protocolo OLSR). Nela podemos visualizar a assinatura gerada pelo middleware SECOM sendo adicionada no final do pacote.

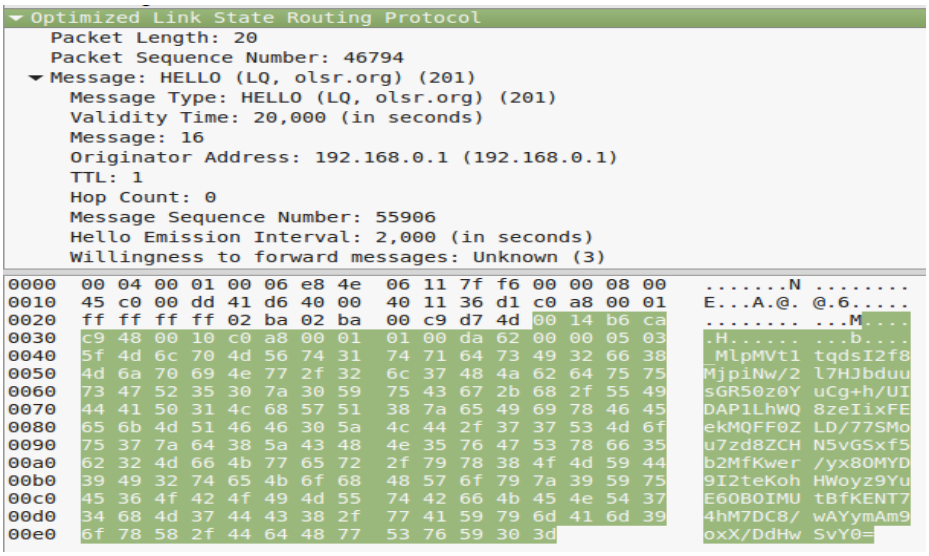

Figura 13: Pacote OLSR modificado e enviado em broadcast na rede.
Fonte: Acervo Pessoal.

Quando um nó, presente na rede, receber esse pacote, ele realiza o processo de verificação da autenticidade e a integridade do pacote recebido. Para isso, antes de realizar o processamento do pacote, é realizada uma chamada ao *middleware* SECOM para verificar se o pacote deve continuar sendo processado ou descartado.

Nesses termos, todos os pacotes recebidos pelo são verificados antes de serem repassados para o protocolo OLSR. Basicamente, é acionada a função do *middleware* responsável por verificar uma assinatura. Para chamar essa função são enviados os seguintes parâmetros: o pacote OLSR original (codificado em Base64), a assinatura adicionada no pacote e endereço IP do nó que enviou esse pacote, conforme pode ser visualizado na Figura 14.

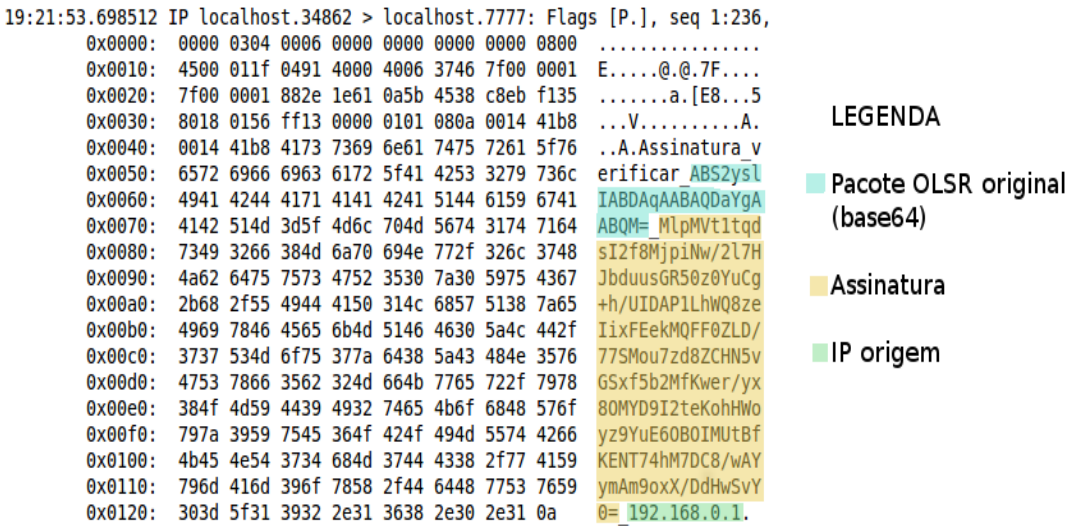

Figura 14: Solicitação ao middleware SECOM para verificar um pacote OLSR recebido.
Fonte: Acervo Pessoal.

Como resposta o *middleware* SECOM envia as seguintes respostas:

a) ACK: o pacote original teve sua autenticidade e integridade comprovada, ou seja, ele foi enviado por um nó autorizado a estar na rede e não sofreu nenhuma alteração durante a transmissão até ser recebido;

b) ERR: o pacote não deve ser processado pelo protocolo OLSR, pois não teve a autenticidade ou integridade comprovada.

Nesse caso, o *middleware* SECOM retornou a mensagem ACK como resposta, conforme pode ser visualizado na Figura 15.

```
19:21:53.700628 IP localhost.7777 > localhost.34862: Flags [P.], seq 1:5,
        0x0000:  0000 0304 0006 0000 0000 0000 0000 0800  ................
        0x0010:  4500 0038 b08e 4000 4006 8c2f 7f00 0001  E..8..@.@../....
        0x0020:  7f00 0001 1e61 882e c8eb f135 0a5b 4623  .....a.....5.[F#
        0x0030:  8018 015e fe2c 0000 0101 080a 0014 41b8  ...^.,........A.
        0x0040:  0014 41b8 4143 4b00                      ..A.ACK.
```

Resposta da verificação

Figura 15: Resposta da verificação enviada pelo middleware SECOM.
Fonte: Acervo Pessoal.

A Figura 16 apresenta um diagrama de estados, no qual é ilustrado o processo de adição de uma assinatura no *payload* de um pacote OLSR antes que ele seja enviado na rede. Além disso, é apresentado o processo realizado quando um nó recebe um pacote OLSR, ou seja, a verificação da assinatura e ação a ser realizada após esse processo.

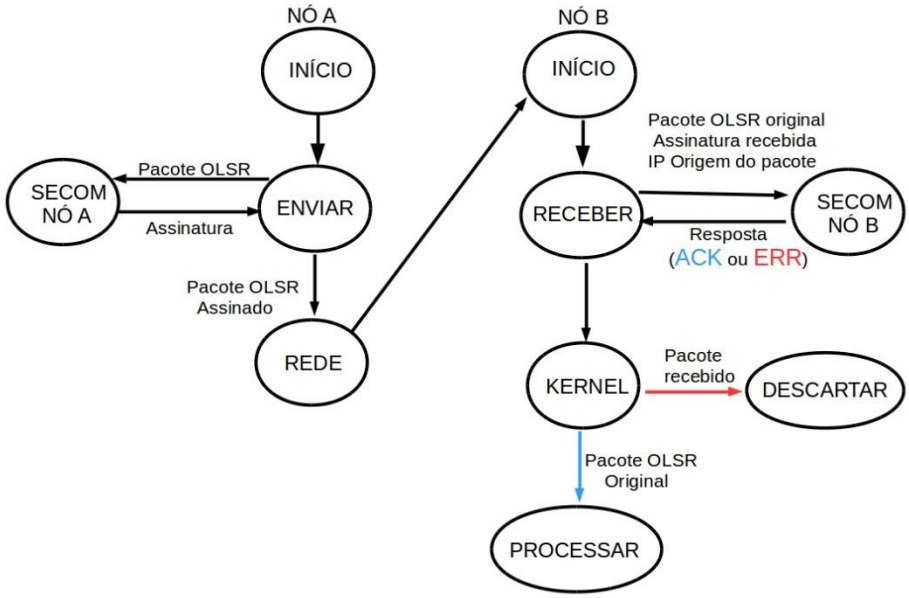

Figura 16: Diagrama de estados.
Fonte: Acervo Pessoal.

Conforme a Figura 16, após a verificação de uma assinatura, se a resposta recebida seja “ACK”, é retirada a assinatura adicionada no pacote e ele segue o fluxo normal de

processamento, realizado pelo protocolo OLSR. Se receber uma resposta “ERR”, o pacote é descartado. Essa resposta pode ser recebida nos seguintes casos:

a) Falha de autenticidade: o emissor do pacote não tem suas credencias cadastradas no servidor de chaves, ou seja, ele não possui autorização para ingressar na rede. Dessa forma, é possível detectar e impedir que falsos nós possam inundar a rede com mensagens de controle, divulgar falsas redes e inserir-se nas tabelas de roteamento dos dispositivos participantes da mesma.

b) Integridade comprometida: o pacote OLSR pode ter sido capturado por algum nó intermediário, o qual realizou a modificação de alguma informação contida nele. Nesse caso, após ser modificado o pacote foi repassado ao seu destino. Caso ele fosse aceito, um nó malicioso poderia modificar uma informação da topologia da rede para adicionar informações falsas, que poderiam comprometer toda a rede, como por exemplo: anúncio de uma falsa rede ou inserção de um falso nó na tabela de roteamento dos nós pertencentes a uma rede.

6 TESTES E RESULTADOS

Para verificar a segurança em uma rede mesh, foi implementado um cenário de testes, onde são representados os equipamentos ativos em uma *Smart Grid*. Para isso, utilizaram-se computadores com as configurações apresentadas na Tabela 1.

Tabela 1: Configuração dos dispositivos utilizados nos testes.

Nó	Sistema Operacional	RAM	Processador	Placa de rede sem fio
A	Linux Mint 64 bits	6 GB	Intel Core i5 2,5 GHz	802.11bgn
B	Linux Mint 64 bits	3 GB	AMD Phenon 2,0 GHz	802.11bgn
C	Linux Mint 64 bits	2 GB	AMD Phenon 2,5 GHz	802.11bgn
D	Linux Mint 64 bits	2 GB	AMD Phenon 2,0 GHz	802.11bgn
E	Linux Mint 64 bits	4 GB	Intel Core i5 2,5 GHz	802.11abgn

Fonte: Acervo Pessoal.

A Figura 17 ilustra a topologia utilizada e os endereços IP de cada dispositivo.

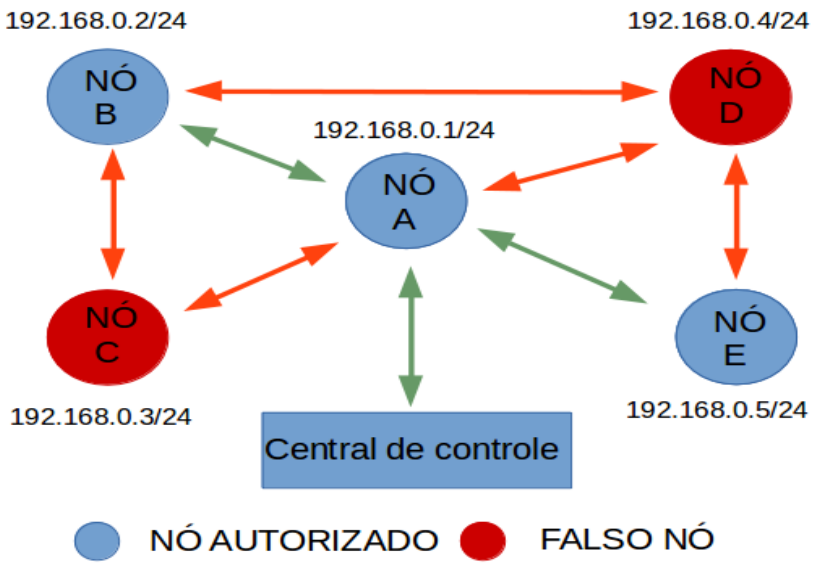

Figura 17: Cenário de testes.
Fonte: Acervo Pessoal.

De acordo com a Figura 17, para simplificar o cenário de testes, será utilizado um nó para representar um concentrador de dados (nó A), o qual será responsável por enviar os dados para a central de controle. Os demais nós (B e E) representarão os dispositivos instalados na residência do cliente, por exemplo, um medidor inteligente. Além disso, são adicionados falsos nós que tentarão obter acesso a rede (C e D).

Com base nesse cenário, os testes realizados foram divididos em três etapas, onde utilizou-se:

1. Protocolo OLSR nativo;
2. Protocolo OLSR e o *plugin secure* OLSR;
3. Protocolo OLSR modificado nesse trabalho e o *middleware* SECOM.

O primeiro passo para a realização dos testes, é a configuração dos dispositivos. Dessa forma, foram realizadas as seguintes configurações:

- Desativar o *daemon* do gerenciador de rede do sistema operacional;
- Desativar a interface da placa de rede sem fio;
- Placa de rede em modo *ad hoc;*
- Canal que a placa de rede sem fio irá operar: 7;
- ESSID da rede: mesh;
- Atribuir endereço IP para a interface da placa de rede sem fio;
- Ativar a interface da placa de rede sem fio.

Para facilitar esse processo, foi utilizado um *shell script*, conforme pode ser visualizado na Figura 18. Dessa forma, é necessário mudar apenas o endereço IP, antes de executá-lo em cada máquina.

```
alexandre@alexandre-sala302 ~/Área de Trabalho/TESTES $ cat comando_mesh_olsr.sh
#!/bin/bash
sudo su
service network-manager stop
ifconfig wlan0 down
iwconfig wlan0 mode ad-hoc
iwconfig wlan0 channel 7
iwconfig wlan0 essid mesh
ifconfig wlan0  192.168.0.1/24
ifconfig wlan0 up
```

Figura 18: *Script* para configurar rede mesh.
Fonte: Acervo Pessoal.

Após as devidas configurações nos dispositivos, é necessário realizar as configurações exigidas pelo protocolo OLSR. Para isso, foi editado o arquivo "/etc/olsr/olsrd.conf", o qual é criado durante a instalação do *daemon* "olsrd". Esse arquivo contém os seguintes parâmetros: porta utilizada pelo protocolo OLSR, endereço de rede e gateway que serão anunciados pela mensagem HNA, interface de rede a ser utilizada e os tempos de emissão e validade das mensagens de controle do protocolo OLSR. A Figura 19 apresenta as configurações realizadas nesse arquivo.

```
alexandre@alexandre-sala302 ~ $ cat  /etc/olsrd/olsrd.conf
DebugLevel 1
IpVersion 4
Hna4
{
0.0.0.0 0.0.0.0
}
OlsrPort 698
UseHysteresis no
TcRedundancy 2

Interface "wlan0"
{
        Mode "mesh"
        Ip4Broadcast 255.255.255.255
        HelloInterval 2.0
        HelloValidityTime 20.0
        TcInterval 5.0
        MidInterval 5.0
        MidValidityTime 300.0
        HnaInterval 5.0
        HnaValidityTime 300.0
        Weight 0
}
```

Figura 19: Arquivo "/etc/olsr/olsrd.conf".
Fonte: Acervo Pessoal.

6.1 Cenário utilizando o protocolo OLSR nativo

Com base na Figura 16, nesse cenário, será analisada a segurança nativa do protocolo OLSR. Ou seja, se o protocolo OLSR permite que falsos dispositivos, que possuam as mesmas configurações dos nós participantes da rede, insiram informações na rede.

Após a configuração de cada dispositivo (conforme a Figura 18 e Figura 19), o protocolo OLSR foi iniciado em ambos. Após as trocas de mensagens de controle entre os

dispositivos, podemos visualizar a tabela de roteamento do nó A, fornecida pelo *daemon* “olsrd” na Figura 20.

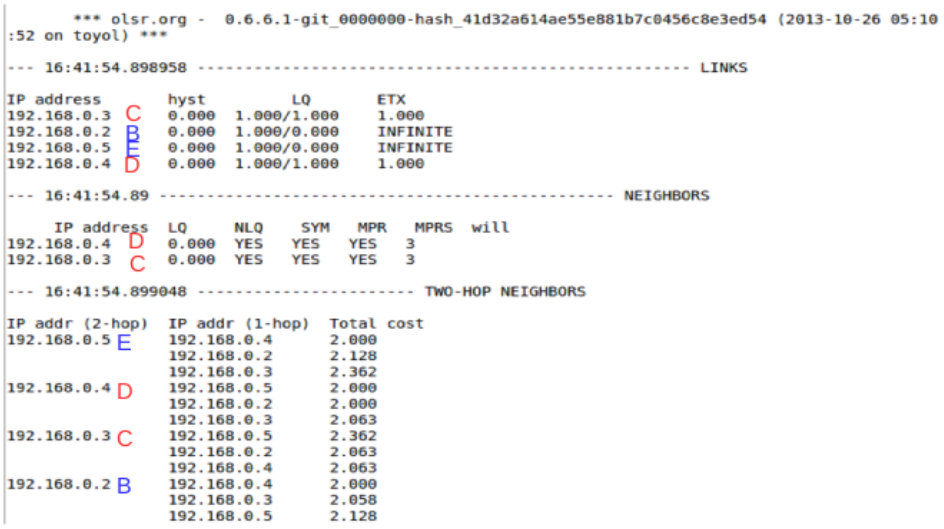

Figura 20: Tabela de roteamento do nó A.
Fonte: Acervo Pessoal.

Conforme a Figura 20, os dispositivos C e D conseguiram realizar alterações na topologia da rede, ou seja, eles tiveram suas mensagens de controle aceitas e constam nas rotas para a comunicação entre os nós verdadeiros da rede mesh implementada. Um exemplo disso é a comunicação entre os nós A e E. Além do *link* direto (apenas um salto) entre eles, ela pode ser realizada por *links* de dois saltos, passando pelos seguintes nós: B, C e D. Dessa forma, evidencia-se a necessidade de utilizar um mecanismo de autenticação entre os dispositivos, quando para prover uma rede segura.

6.2 Cenário utilizando o protocolo OLSR e o *plugin secure* OLSR

Para utilizar o *plugin secure* OLSR, é necessário editar o arquivo de configuração do *daemon* do protocolo OLSR (“olsrd”), apresentado na Figura 19, inserindo os seguintes parâmetros: versão do plugin e localização do arquivo que contém a chave simétrica a ser utilizada. Na Figura 21, podemos visualizar os parâmetros adicionados no referido arquivo, no

qual é apresentado os parâmetros necessários para que o *daemon* “olsrd” carregue o plugin Secure OLSR.

```
LoadPlugin "olsrd_secure.so.0.6"
{
        'PlParam     "Keyfile"    "/etc/olsr/.secure_key_olsr"
}
```

Figura 21: Parâmetros do plugin Secure OLSR.
Fonte: Acervo Pessoal.

Utilizando o cenário apresentado na Figura 17, os nós A, B e E utilizam a mesma chave para assinar as mensagens de controle do protocolo OLSR, o nó C utiliza uma chave diferente e o nó D, não acrescenta nenhuma assinatura em seus pacotes. Dessa forma, após a inicialização do *daemon* “olsrd” em todos os dispositivos, apenas os nós que utilizam a mesma chave, chamada de chave de grupo, podem fazer parte da topologia da rede.

Nesse caso, a Figura 22 apresenta a tabela de roteamento do nó A. Nela podemos visualizar os vizinhos descobertos por esse nó e as rotas que podem ser utilizadas para comunicação com outros nós da rede.

```
      *** olsr.org - 0.6.8-git_0000000-hash_f90f6d7f8b957fff3eea9cf8ba30a665 (2
015-06-17 17:49:32 on alexandre-note) ***

--- 18:27:54.438184 ------------------------------------------------- LINKS

IP address       hyst         LQ       ETX
192.168.0.5 E    0.000  1.000/0.854    1.169
192.168.0.2 B    0.000  1.000/1.000    1.000

--- 18:27:54.438205 ---------------------------------------------- NEIGHBORS

     IP address Hyst    LQ      ETX     SYM   MPR   MPRS  will
192.168.0.5 E  0.000   1.000/0.854     1.169   YES   NO    NO    3
192.168.0.2 B  0.000   1.000/1.000     1.000   YES   NO    NO    3

--- 18:27:54.438225 ---------------------- TWO-HOP NEIGHBORS

IP addr (2-hop)  IP addr (1-hop)  Total cost
192.168.0.5 E    192.168.0.2 B    2.128
192.168.0.2 B    192.168.0.5 E    2.528
```

Figura 22: tabela de roteamento do nó A, utilizando o *plugin Secure* OLSR.
Fonte: Acervo Pessoal.

De acordo com a Figura 22, apenas os nós autorizados (que utilizaram a chave de grupo) conseguiram acessar a rede, ou seja, tiveram suas mensagens de controle aceitas,

conforme pode ser visualizado na Figura 23, onde a assinatura (*hash*) adicionada em um pacote enviado pelo nó B teve sua autenticidade comprovada e foi aceita por A.

```
Recevied hash:
 187  83   6 219 190 137 175 161 152  60 120 253 110 191  68  66
Calculated hash:
 187  83   6 219 190 137 175 161 152  60 120 253 110 191  68  66
[ENC]Received timestamp 1434576463 diff: 12
[ENC]Packet from 192.168.0.2 OK size 60
```

Figura 23: Pacote enviado por B, utilizando o *plugin Secure* OLSR.
Fonte: Acervo Pessoal.

Dessa forma, o nó A possui um *link* de um salto com os nós B e E. Além disso, o nó A consegue se comunicar com o nó E, passando pelo nó B, e com o nó B, através do nó E.

As mensagens de controle enviadas pelos nós que tentaram obter acesso a rede foram descartas, conforme pode ser visualizado na Figura 24 (mensagem do nó C) e Figura 25 (mensagem do nó D).

```
Recevied hash:
 158  44  37 207 250  18  64 124 241  12 215  25  28 226  93  48
Calculated hash:
 244  81 147 243 221  39  93 217 254   0 159 166 230 171  91 164
[ENC]Signature missmatch
[ENC]Rejecting packet from 192.168.0.3
```

Figura 24: Pacote do nó C.
Fonte: Acervo Pessoal.

```
[ENC]Packet not sane!
[ENC]Rejecting packet from 192.168.0.4
```

Figura 25: Pacote do nó D.
Fonte: Acervo Pessoal.

Conforme pode ser visualizado nas Figuras 24 e 25, a utilização desse *plugin*, impossibilita que dispositivos não autorizados ingressem na rede. Entretanto, por utilizar uma chave compartilhada entre todos os dispositivos, caso essa seja descoberta ou compartilhada

com um nó malicioso, a segurança de toda rede pode ser comprometida. Dessa forma, um falso nó consegue ingressar na rede, enquanto essa chave for válida.

6.3 Cenário utilizando o Middleware SECOM e o protocolo OLSR modificado

Após o processo de configuração de cada dispositivo (conforme a Figura 18 e Figura 19), o protocolo OLSR foi iniciado em ambos. Além disso, foi inicializado o *daemon* do *middleware* SECOM nos dispositivos A, B, C e E. Dessa forma, esses nós adicionam um campo extra no pacote OLSR, contendo a assinatura gerada pelo *middleware* SECOM. No caso do nó D, essa assinatura não será adicionada e o nó C, assinará suas mensagens com uma chave não cadastrada no servidor de chaves do *middleware* SECOM.

Nesse contexto, os pacotes OLSR serão aceitos se tiverem sua autenticidade e integridade comprovada, ou seja, se foram enviados por um dispositivo autorizado na rede e não tiveram seu conteúdo modificado.

Após o envio das mensagens de controle, os nós A, B e E conseguiram interagir, isso é, cada nó conseguiu reconhecer seus vizinhos e estabelecer rotas para comunicar-se com eles. Com isso, podemos visualizar na Figura 26, a tabela de roteamento do nó A, com seus respectivos vizinhos e suas rotas.

```
       *** olsr.org - 0.6.8-git_0000000-hash_fc0159bf5ba1cc92b654571884d27b6c
TC:   found neighbor tc_entry 192.168.0.2
TC: add edge entry 192.168.0.1 > 192.168.0.2, cost (0.000/0.000) INFINITE

--- 18:07:48.754041 ------------------------------------------------- LINKS

IP address       hyst         LQ       ETX
192.168.0.2 B    0.000  0.313/0.113    28.027
192.168.0.5 E    0.000  0.113/0.059    INFINITE

--- 18:07:48.754070 ---------------------------------------------- NEIGHBORS

     IP address Hyst    LQ      ETX     SYM   MPR   MPRS  will
192.168.0.2 B   0.000   0.313/0.113     28.027  YES   NO    NO    3

--- 18:07:48.754091 ---------------------- TWO-HOP NEIGHBORS

IP addr (2-hop)  IP addr (1-hop)  Total cost
192.168.0.5 E    192.168.0.2      INFINITE
192.168.0.2 B    192.168.0.5      INFINITE
Processing TC from 192.168.0.2, seq 0x8ce6
TC:   found neighbor tc_entry 192.168.0.1
TC:   found inverse edge for 192.168.0.2
TC: add edge entry 192.168.0.2 > 192.168.0.1, cost (0.000/0.000) INFINITE
TC:   chg edge entry 192.168.0.2 > 192.168.0.5, cost (0.164/0.223) 27.161
```

Figura 26: Tabela de roteamento do nó A, utilizando a proposta desse trabalho.
Fonte: Acervo Pessoal.

Com base na Figura 26, o nó A possui um *link* de um salto com os nós B e E. Além disso, o nó A consegue se comunicar com o nó E, passando pelo nó B, e com o nó B, através do nó E.

Em razão de não utilizar o protocolo OLSR modificado nesse trabalho (não insere assinatura no *payload* de seus pacotes) e não estar cadastrado no servidor de chaves do *middleware* SECOM, o nó D não conseguiu obter informações da rede, visto que ele não esperava pacotes assinados. A Figura 27 mostra os pacotes recebidos sendo ignorados. Nela podemos visualizar o tamanho dos pacotes assinados pelos nós que utilizam o protocolo OLSR modificado.

```
       *** olsr.org -  0.6.6.1-git_0000000-hash_41d32a614ae55e881b7c0456c8e3ed54
 (2013-10-26 05:10:52 on toyol) ***

--- 13:24:47.129122 ------------------------------------------------- LINKS

IP address       hyst         LQ       ETX

--- 13:24:47.12 ----------------------------------------------- NEIGHBORS

     IP address  LQ     NLQ    SYM   MPR   MPRS  will

--- 13:24:47.129300 ---------------------- TWO-HOP NEIGHBORS

IP addr (2-hop)  IP addr (1-hop)  Total cost
Size error detected in received packet.
Received 213, in packet 40
Size error detected in received packet.
Received 257, in packet 84
Size error detected in received packet.
Received 213, in packet 40
Size error detected in received packet.
```

Figura 27: Pacotes recebidos pelo nó D.
Fonte: Acervo Pessoal.

A Figura 28 apresenta a tabela de roteamento do nó C.

```
Main address: 192.168.0.3

Scheduler started - polling every 50 ms

       *** olsr.org - 0.6.8-git_0000000-hash_fc0159bf5ba1cc92b654571884d27b6c (2015-05-23 19:56:13 o
n alexandre-note) ***

--- 18:07:49.379673 ------------------------------------------------- LINKS

IP address       hyst         LQ       ETX

--- 18:07:49.379771 ---------------------------------------------- NEIGHBORS

     IP address Hyst    LQ      ETX     SYM   MPR   MPRS  will

--- 18:07:49.379859 ---------------------- TWO-HOP NEIGHBORS

IP addr (2-hop)  IP addr (1-hop)  Total cost
REJECTED PACKET BECAUSE IT IS UNSIGNED
```

Figura 28: Tabela de roteamento do nó C, utilizando a proposta desse trabalho sem estar cadastrado no servidor de chaves.
Fonte: Acervo Pessoal.

Conforme podemos visualizar na Figura 28, o nó C não conseguiu obter nenhuma informação sobre a topologia da rede. Embora ele utilize o protocolo OLSR modificado nesse trabalho, em razão de não estar cadastrado no servidor de chaves, ele também não conseguiu realizar nenhuma alteração na rede. Ou seja, as mensagens de controle enviadas por ele, foram descartadas pelos dispositivos autorizados na rede (cadastrados no servidor de chaves).

6.4 Análise dos Resultados

De acordo com os resultados apresentados no primeiro cenário de testes (protocolo OLSR nativo), não existe nenhum mecanismo para restringir o acesso não autorizado a uma rede. Dessa forma, qualquer dispositivo consegue ingressar na rede, divulgar rotas e participar da tabela de roteamento dos dispositivos ativos na rede.

A utilização do *plugin secure* OLSR contribui para solucionar essa situação. Entretanto, ele necessita de um mecanismo eficiente para gerenciar a chave compartilhada na rede, visto que todos os dispositivos precisam conhecê-la para ingressar na rede. Se um nó malicioso obter acesso a essa chave, ele pode participar da rede como se fosse um nó

autorizado. Dessa forma, a chave precisa ser modificada e atualizada manualmente em todos dispositivos, para que uma chave descoberta não seja mais válida na rede.

Nesses termos, a proposta desse trabalho destaca-se por utilizador um servidor de chaves centralizado, ou seja, cada dispositivo utiliza suas próprias chaves para assinar as mensagens enviadas e verificar cada mensagem recebida. Com base nos resultados obtidos, esse trabalho contribui para implementar uma rede mesh segura, visto que para participar da rede, um nó precisa ter suas credenciais cadastradas no servidor de chaves. Dessa forma, é possível garantir a autenticidade e integridade das mensagens de controle trafegadas na rede. Em razão do descarte dos pacotes que não obtiveram sucesso na verificação da assinatura, um falso nó não pode realizar nenhuma alteração nas tabelas de roteamento dos dispositivos participantes da rede.

7 CONSIDERAÇÕES FINAIS

A solução apresentada nesse trabalho apresentou resultados satisfatórios, como por exemplo: cada dispositivo utiliza suas próprias chaves para assinar os pacotes OLSR, apenas dispositivos autorizados (cadastrados no servidor de chaves) conseguem ingressar na rede e o descarte de pacotes que não comprove sua autenticidade ou integridade. Assim, essa forma de comunicação pode ser aplicada na implementação de uma *Smart Grid*, atendendo aos requisitos necessários para garantir uma comunicação segura e confiável entre os dispositivos ativos no sistema elétrico de potência, como por exemplo, um concentrador de dados e os dispositivos atendidos por ele.

Portanto, a principal contribuição deste trabalho é provar que uma rede mesh pode fornecer uma comunicação segura e pode ser utilizada na implementação de um *Smart Grid*. A próxima etapa a ser realizada, em trabalhos futuros, é verificar o desempenho (tempo de processamento, consumo de banda) dos processos de adição e verificação de assinaturas nos pacotes OLSR. Além disso, pretende-se desenvolver um sistema operacional Linux, customizado para executar o protocolo OLSR e o *middleware* SECOM em dispositivos com baixos recursos computacionais.

7.1 Trabalhos relacionados aceitos para publicação

O presente trabalho é parte de um projeto de pesquisa que visa desenvolver soluções para a implementação de *Smart Grids*. A principal motivação desse grupo é prover a uma comunicação segura e confiável entre os dispositivos presentes no sistema elétrico de potência e sistemas supervisórios. Nesses termos, os artigos citados abaixo foram desenvolvidos pelos integrantes desse grupo e foram aceitos para publicação em um evento internacional.

- RIZZETTI, TIAGO ANTONIO; RODRIGUES, ALEXANDRE SILVA; SILVA, BOLÍVAR MENEZES; RIZZETTI, BRUNO AUGUSTO; WESSEL, PEDRO; CANHA, LUCIANE NEVES. Security of communications on a high availability mesh network applied in Smart Grids. In: 2015 50th International Universities Power

Engineering Conference (UPEC), 2015, Staffordshire University. 2015 50th International Universities Power Engineering Conference (UPEC), 2015.

- RIZZETTI, TIAGO ANTONIO; WESSEL, PEDRO; RODRIGUES, ALEXANDRE SILVA; SILVA, BOLÍVAR MENEZES; MILBRADT, RAFAEL; CANHA, LUCIANE NEVES. Cyber security and communications network on SCADA systems in the context of Smart Grids. In: 2015 50th International Universities Power Engineering Conference (UPEC), 2015, Staffordshire University. 2015 50th International Universities Power Engineering Conference (UPEC), 2015.

REFERÊNCIAS

ABELÉM, A. J. G. et al. **Redes mesh:** Mobilidade, qualidade de serviço e comunicação em grupo. 2007.

BOWITZ, A. G. et al. BatCave: Adding security to the BATMAN protocol. In: **Digital Information Management (ICDIM)**, 2011 Sixth International Conference on. IEEE, 2011. p. 199-204

CARDOSO, T. M.; MARQUES, P. C. F. FURLANETTO, P. C. Rede Mesh: topologia e aplicação. In: **Revista iTEC**. Vol. IV, n. 4, p. 16, 2012.

CLAUSEN, T.; JAQCQUET, P. Optimized link state routing (OLSR) RFC 3626. **IETF Networking Group.** 2003.

EKANAYAKE, J. et al. **Smart grid:** technology and applications. John Wiley & Sons, 2012.

FERNANDES, N. C. **Análise de Ataques e Mecanismos de Segurança em Redes Ad Hoc**. 2006. 117 f. Dissertação (Doutorado Engenharia Elétrica)-Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.

GTREI. Grupo de Trabalho de Redes Elétricas Inteligentes. **Smart Grid**. Relatório. 2010.

HAFSLUND, A. et al. Secure Extension to the OLSR protocol. In: **Proceedings of the OLSR Interop and Workshop**, San Diego, 2004.

JUNIOR, A. A.; DUARTE, O. C. M. B. **Segurança no roteamento em redes móveis ad hoc**. In: Seminário de Tópicos Especiais em Redes de Computadores, 2003. Rio de Janeiro: GTA-Universidade Federal do Rio de Janeiro, 2003. p. 1-16.

LOPES, Y. et al. **Smart Grid e IEC 61850:** Novos Desafios em Redes e Telecomunicações para o Sistema Elétrico. In: XXX Simpósio Brasileiro de Telecomunicações, 2012.Brasília: Universidade de Brasília ,2012.

MACHADO, A. L. **Autenticação centralizada com Freeradius em infraestrutura de redes mesh.** 2013. 55 f. Monografia (Graduação em Redes de Computadores)- Instituto Federal de Educação, Ciência e Tecnologia Catarinense, Sombrio, 2013.

MOLLIN, R. A. **RSA and public-key cryptography**. CRC Press, 2002.

PERKINS, C. E.; ROYER, E. M. Ad-hoc on-demand distance vector routing**.** In: **Mobile Computing Systems and Applications**, 1999. Proceedings. WMCSA'99. Second IEEE Workshop on. IEEE, 1999. p. 90-100.

RAMOS, M. L. S. **Proposta de um método de segurança da informação para sistemas de automação em redes elétricas inteligentes.** 2012. 108 f. Dissertação (Mestrado Desenvolvimento de Tecnologia)- Instituto de Engenharia do Paraná, Curitiba, 2012.

SEN, J. Security and privacy issues in wireless mesh networks: A survey. In: **Wireless networks and security**. Springer Berlin Heidelberg, 2013. p. 189-272.

SILVA, Z. S. **Construindo roteadores Wi-Mesh com GNU/Linux E OLSR**. 2015. 94 f. Monografia (Especialização Administração de Redes Linux)-Universidade Federal de Lavras, Lavras, 2011.

SILVA, B. M. **Um middleware para prover comunicação segura entre dispositivos.** 2015. Monografia (Graduação Redes de Computadores)-Universidade Federal de Santa Maria, Santa Maria, 2015.

WANG, W.; XU, Y.; KHANNA, M. **A survey on the communication architectures in smart grid.** Computer Networks, v. 55, n. 15, p. 3604-3629, 2011.

ZAPATA, M. G. **Secure ad hoc on-demand distance vector routing.** ACM SIGMOBILE Mobile Computing and Communications Review, v. 6, n. 3, p. 106-107, 2002.

ZHANG, Y.; LUO, J.; HU, H. **Wireless mesh networking**: architectures, protocols and standards. CRC Press, 2006.

ZUCCHI, W. L. O que é uma rede mesh e como o padrão IEEE 802.16 se aplica a esse tipo de topologia. In: **Revista RTI**, p. 104-107. 2006.