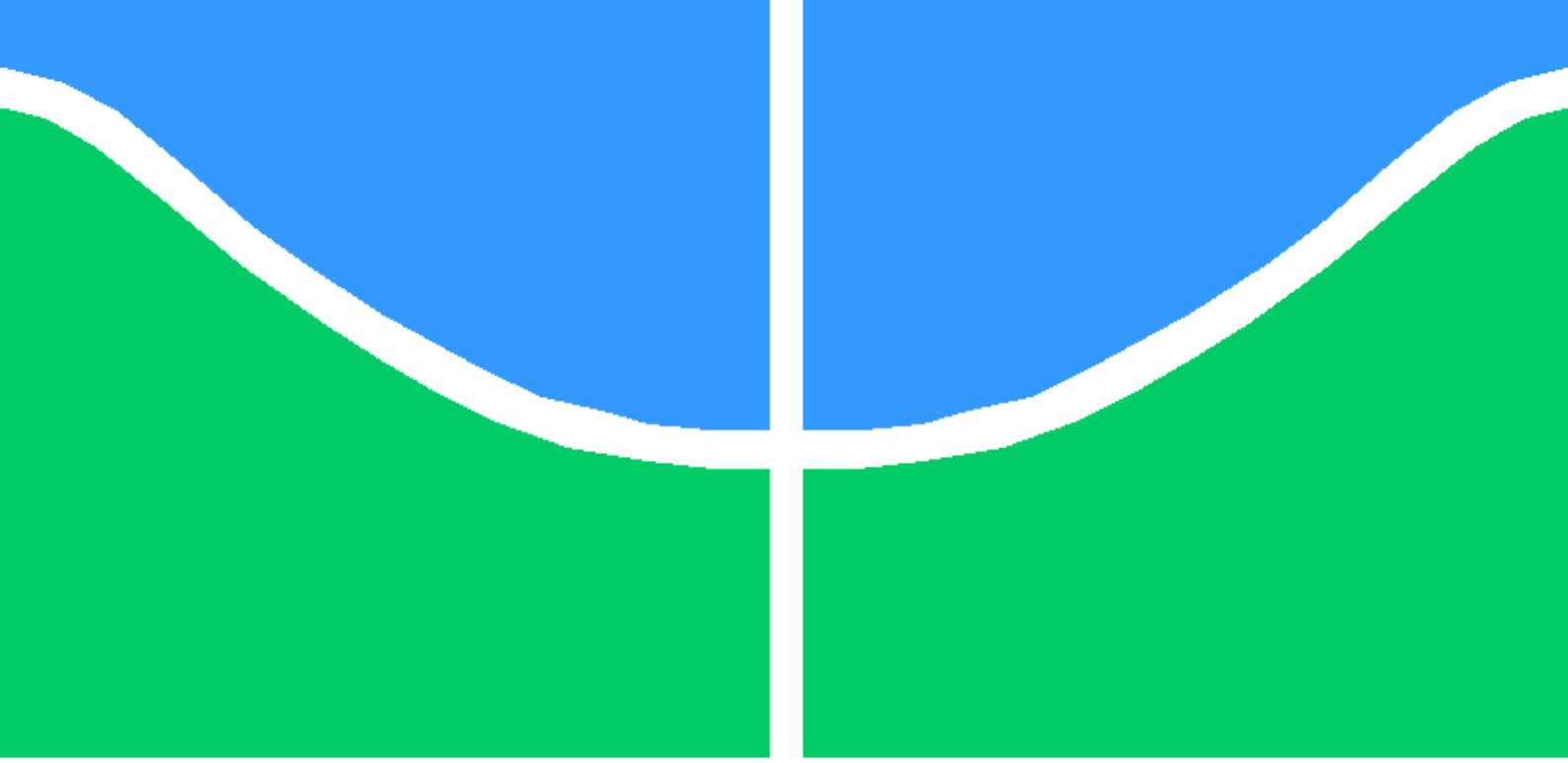

FOG COMPUTING COMO ARQUITETURA DE REDE DISTRIBUÍDA PARA INTERNET DAS COISAS

EDUARDO ANTÔNIO DE SENA

TCC EM ENGENHARIA DE REDES DE COMUNICAÇÃO

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

FOG COMPUTING COMO ARQUITETURA DE REDE DISTRIBUÍDA PARA INTERNET DAS COISAS

EDUARDO ANTÔNIO DE SENA

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO DE REDES DE COMUNICAÇÃO.

APROVADA POR:

_____

Prof. Doutor Edgard Costa Oliveira, FGA/UnB
(Orientador)

_____

Prof. Doutora Edna Dias Canedo, FGA/UnB
Examinador externo

_____

Prof. Doutor Rafael Timoteo de Sousa Jr, ENE/UnB
Examinador interno

BRASÍLIA, 25 DE JUNHO DE 2015.

À todos que me ajudaram a chegar até aqui

AGRADECIMENTOS

Agradeço, em primeiro lugar, a minha mãe, por ter batalhado toda a vida para que eu conseguisse chegar até aqui. Sou eternamente grato a você pela sua vida de dedicação, pelo seu carinho e afeto, por ser a melhor mãe que eu poderia ter.
Agradeço a minha amada Fiama Kétuli, por me apoiar e me acompanhar nessa jornada, por ter escolhido ser minha grande companheira e me acompanhar nos momentos felizes e nos não tão felizes durante todos esses anos.
Agradeço a minha irmã Brunna Hisla e a meu cunhado Roberto Arrial pelo carinho e pelas dicas dadas a respeito de assuntos acadêmicos e não acadêmicos.
A meus amigos Alejandro Barrios, Deijaval Pereira, Marco Paulo, Thiago Zanette e Letícia Costa pela amizade e companheirismo durante todos esses anos.
Ao professor Edgard Oliveira pela orientação dada para concluir esta etapa.

EDUARDO ANTÔNIO DE SENA

RESUMO

FOG COMPUTING COMO ARQUITETURA DE REDE DISTRIBUÍDA PARA INTERNET DAS COISAS

Autor: EDUARDO ANTÔNIO DE SENA

Orientador: Prof. Doutor Edgard Costa Oliveira, FGA/UnB

Projeto Final de Graduação

Brasília, 25 de junho de 2015

Fog Computing é uma extensão não trivial da Computação em Nuvem, que possibilita uma série de novos serviços e aplicações que não são completamente compatíveis com a arquitetura em nuvem. É apresentado o conceito de Fog Computing e como ele pode servir como arquitetura de rede para Internet das Coisas. Por meio de pesquisa bibliográfica, são mostrados os principios básicos para que o cliente final execute tarefas como: controle e configuração da rede; controle de HetNets; medidas e inferências de throughput para adaptação de taxa de transmissão; inferências sobre carga da rede por meios como active probing, correlação de performance e data mining; inferências no nível de aplicação usando DPI e análise de porta; pooling e caching de recursos com virtualização e cyber foraging. Também são apresentados aspectos de segurança e privacidade em Fog Computing e Internet das Coisas como segurança nas camadas de rede e em interfaces web.

Palavras-chave: Fog Computing. Arquitetura Distribuída. Computação Móvel

ABSTRACT

Authors: EDUARDO ANTÔNIO DE SENA

Supervisor: Prof. Doutor Edgard Costa Oliveira, FGA/UnB

Final Graduation Project

Through specific literature search , basic principles such as control and network configuration are shown; HetNets Control; throughput inferences and measure for the rate adaptation ; inferences about network load by means such as active probing, performance correlation and data mining; inferences on the application level using DPI and port analysis; pooling and caching capabilities with virtualization and cyber foraging. Security and privacy aspects of Fog Computing and Internet of Things are also presented.

Keywords: Fog Computing. Distributed Architecture. Mobile Computing

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

Siglas

6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
B2C	Business to Consumer
CoAP	Constrained Application Protocol
CPS	Cyber-Physical System
CSRF	Cross Site Request Forgery
DoS	Denial of Service
DPI	Deep packet inspection
DTLS	Datagram Transport Layer Security
HetNet	Heterogeneous Network
IoT	Internet of Things
IP	Internet Protocol
IPSec	IP Security Protocol
LTE	Long-Term Evolution
M2M	Machine-to-Machine
OWASP	Open Web Application Security Project
RAT	Radio Access Technologies
RF	Radio frequency
RRAA	Robust Rate Adaptation Algorithm
RSSI	Received Signal Strength Indicator
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
RPC	Remote Procedure Call
RTT	Round-Trip Time
S/MIME	Secure/Multipurpose Internet Mail Extensions
SSL	Secure Sockets Layer
SNR	Signal-to-Noise Ratio
SQLi	SQL Injection
SRTP	Secure Real-time Transport Protocol
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TLS	Transport Layer Security
UDP	User Datagram Protocol
VM	Virtual Machine
XSS	Cross-site scripting

1 INTRODUÇÃO

Fog Computing é uma extensão não trivial da Computação em Nuvem, que possibilita uma série de novos serviços e aplicações que não são completamente compatíveis com a arquitetura em nuvem. Pode ser definido como uma arquitetura de rede, que usa um ou uma multidão de colaboração de usuários finais, tais como celulares ou computadores, ou dispositivos de borda perto do usuário, como conversores digitais para TV e gateways domésticos, para entregar uma quantidade substancial de recursos como armazenamento, comunicação e medidas de controle, configuração e gestão da rede (BONOMI et al., 2014).

Computação em Nuvem possui uma arquitetura centralizada, que permite várias facilidades, mas vêm com um custo: latência. Caso o data center não esteja próximo dos utilizadores, aplicações sensíveis a atrasos perdem desempenho no modelo de computação em nuvem. Fog Computing é uma arquitetura de rede distribuída, mais perto dos clientes, na borda da rede. Isso possibilita uma baixa latência e evita que todo o tráfego dos dispositivos seja direcionado para o centro da rede, na nuvem. Tal arquitetura permite diversas novas aplicações que estão surgindo com a Internet das Coisas, que possuem requisitos que não podem ou são mal atendidos pela arquitetura centralizada da Computação em Nuvem. A distância entre a nuvem e os usuários pode ser entendida usando uma analogia: uma pessoa em solo (usuário final) se encontra longe de nuvens reais no céu. E agora, com Fog Computing, a nuvem está descendo para estar dentro, entre ou perto dos dispositivos e usuários finais, na borda da rede.

Sendo uma abordagem nova, Fog Computing não possui tecnologia padrão que possa ser usada para ser implementada. Porém, existem alguns princípios básicos para que tarefas sejam executadas por clientes finais. Estes princípios podem ser usados para a construção de uma arquitetura distribuída na borda, onde os clientes finais ou perto da borda participam ativamente de tarefas como controle e configuração da rede, medições e inferências sobre o estado da rede, bem como pooling de recursos e caching na borda. A segurança e a privacidade são assuntos que também são importantes nessa arquitetura. Novos protocolos e serviços devem ser desenvolvidos ou melhorados para atender a nova demanda da Internet das Coisas, trazendo comunicações seguras para todos os tipos de dispositivos, inclusive dispositivos muito limitados em recursos. Estes, em especial, pedem medidas de segurança que sejam leves para serem executadas com seu restrito poder computacional.

Fog Computing pode realizar tarefas da nuvem, como armazenamento, processamento e gerenciamento de rede mais perto dos clientes, na borda da rede, o que possibilita baixa latência para as aplicações e pode evitar que um grande tráfego seja direcionado exclusivamente para a nuvem, prevenindo um possível colapso da rede. Tal arquitetura distribuída abre diversos desafios, incluindo segurança e privacidade, que são preocupações cada vez maiores na internet.

1.1 MOTIVAÇÃO

Atualmente há uma grande quantidade de dispositivos conectados à internet, e a tendência é que esse número aumente exponencialmente ao longo dos próximos anos, em virtude da chegada da Internet das Coisas, da rápida inovação de dispositivos e protocolos e do contínuo barateamento de custos computacionais (Cisco VNI, 2015b). Diante disso, haverá mais dispositivos conectados que consumirão cada vez mais dados, necessitando de uma rede diferenciada para atendimento da demanda potencial. Uma possibilidade é utilizar a tecnologia de arquitetura distribuída na borda da rede, chamada de Fog Computing, pois ela habilita uma série de novos serviços e aplicações que não são completamente compatíveis com a arquitetura em Computação em Nuvem. Fog Computing é um assunto novo e não existe muito material em língua portuguesa que cubra o assunto.

1.2 OBJETIVOS

Este trabalho tem o objetivo apresentar o conceito de Fog Computing para uma arquitetura de rede distribuída para Internet das Coisas.

1.2.1 Objetivos Específicos

- Apresentar o conceito de Fog Computing;
- Apresentar princípios gerais para uma arquitetura distribuída na borda da rede;
- Analisar a segurança e privacidade ao utilizar Fog Computing e Internet das Coisas.

1.3 METODOLOGIA

Esta pesquisa caracteriza-se por ser do tipo exploratória-descritiva. A pesquisa exploratória é aquela que oferece uma primeira abordagem, elucidando ideias e uma visão panorâmica de um fenômeno pouco explorado, e a pesquisa descritiva preocupa-se em apresentar as características de um objeto de estudo (GONSALVES, 2001).

Será utilizada a pesquisa bibliográfica como método de pequisa. A pesquisa bibliográfica tem sido utilizada com grande frequência em estudos exploratórios ou descritivos, já que quando objeto de estudo é pouco estudado, é difícil a formulação de hipóteses, e é indicada em virtude da aproximação com o objeto se dar a partir de fontes bibliográficas (GIL, 2010). Para tanto, foram analisados artigos científicos nacionais e internacionais, livros e periódicos das áreas de Ciência da Computação e Engenharia Elétrica.

Para apresentar os conceitos básicos de Fog Computing, foram pesquisados artigos científicos nas bases de dados IEEE Xplorer e ACM e livros internacionais que abordam Fog Computing e Internet das Coisas, com autores de renome que apresentaram pela primeira vez o conceito de Fog Computing. Também são analisados relatórios com previsões sobre tráfego de internet e quantidade de dispositivos para os próximos anos.

Para apresentar os conceitos gerais de Fog Computing como uma arquitetura de rede distribuída na borda da rede, foram pesquisados arquivos científicos nas bases de dados IEEE Xplorer e ACM relacionados a Fog Computing, Mobile Computing, Edge Computing e Computação Pervasiva.

Para os conceitos de Segurança e Privacidade em Fog Computing, foram pesquisados artigos científicos nas bases de dados IEEE Xplorer e ACM, livros internacionais que abordam Fog Computing e Internet das Coisas, bem como artigos online criado por organizações influentes preocupadas com a segurança e privacidade na internet.

Serão feitas sínteses dos assuntos abordados, que estarão no final de cada capítulo.

2 FOG COMPUTING

Fog Computing é uma extensão não trivial da Computação em Nuvem, que possibilita uma série de novos serviços e aplicações que não são completamente compatíveis com a arquitetura em nuvem. Este capítulo apresenta uma visão geral da ideia de Fog Computing, também conhecido como Fog Networking ou redes Fog. Neste capítulo serão mostrados quais são os fatores responsáveis por trazer esse conceito de Fog Computing e quais os caminhos para que essa arquitetura alcance seus objetivos.

Fog traz a possibilidade de desenvolvimento de novas aplicações e serviços, que incluem (BONOMI et al., 2014):

- Serviços que necessitam de uma latência baixa e previsível;
- Aplicações geodistribuidas, como sensores de ambiente;
- Aplicações em dispositivos que se movem em alta velocidade, como carros ou trens;
- Sistemas de controle distribuídos em larga escala, como semáforos, medidores e rede elétrica inteligentes, cidades inteligentes em geral.

Essas categorias de serviços e aplicações estão presentes em diversos casos de uso da Internet das Coisas (IoT) e causam mudanças de paradigma em Big Data.

2.1 REVISÃO BIBLIOGRÁFICA

Esta seção visa revisar os conceitos básicos de Computação em Nuvem, que é a arquitetura que será estendida com o conceito de Fog Computing.

2.1.1 Computação em Nuvem

Cloud Computing, ou Computação em Nuvem, é um modelo que permite um acesso conveniente, ubíquo e sob demanda a um pool compartilhado de recursos computacionais configuráveis (como rede, servidores, armazenamento, aplicações, serviços) que podem ser rapidamente providenciados e lançados com um mínimo de esforço de gerência ou interação com o provedor de serviços (MELL; GRANCE, 2011). Esse modelo em nuvem é composto de cinco características essenciais, três modelos de serviços e quatro modelos de implantação.

2.1.1.1 Características

Computação em Nuvem possui as seguintes características (ABBASOV, 2014):

- Autoatendimento sob demanda: Um usuário pode usar recursos da nuvem, como armazenamento de rede e tempo de servidor, de um modo automático e sem necessidade de interação humana com os provedores de serviço;
- Amplo acesso a rede: Os recursos da nuvem estão disponíveis pela internet e são acessíveis através de diversos terminais, como computadores, telefones celulares, tablets, entre outros;
- Pool de Recursos: Os recursos computacionais são agrupados de maneira a servirem vários consumidores. Os recursos físicos e virtuais são atribuídos de maneira dinâmica, a medida que o cliente precise. Consumidores geralmente não possuem controle sobre a localização exata dos recursos computacionais providos. Mas consumidores podem ser capazes de especificar de qual país, cidade ou Data Center desejam ser providos com os recursos. Recursos computacionais incluem armazenamento, processamento, memória e largura de banda;
- Elasticidade rápida: Recursos computacionais podem ser providos aos consumidores de uma maneira elástica, ou seja, podem ser alocados apropriadamente mais ou menos recursos dependendo da demanda. Consumidores geralmente podem dimensionar recursos que parecem ser ilimitados e que podem ser providos a qualquer quantidade e a qualquer tempo;
- Serviços mensuráveis: Recursos computacionais podem ser automaticamente controlados e otimizados utilizando uma métrica apropriada ao serviço provido. Utilização de recursos pode ser monitorada, controlada e relatadas para proverem transparência entre o provedor e consumidor.

2.1.1.2 Serviços baseados em Computação em Nuvem

Computação em Nuvem oferece os seguintes serviços (HAMDI, 2012):

- Software as a Service (SaaS): provê aplicações rodando na nuvem, onde o usuário praticamente não pode controlar nem gerenciar a infraestrutura interna;
- Platform as a Service (PaaS): provê uma série de ferramentas que suportam certas tecnologias de desenvolvimento e todo o ambiente necessário para implantação de aplicativos criados pelos usuários. O cliente é habilitado a controlar e gerenciar suas aplicações;

- Infrastructure as a Service (IaaS): provê recursos computacionais básicos como processamento, armazenamento e largura de banda, onde o usuário älugaä infraestrutura e paga por utilização ou tempo de utilização, sendo o usuário responsável por todos os outros aspectos de desenvolvimento.

2.1.1.3 Modelos de Implementação

Computação em Nuvem oferece os seguintes modelos de implementação (MELL; GRANCE, 2011):

- Nuvem Privada: a infraestrutura da nuvem é provisionada para uso exclusivo de uma única organização. Podem ser possuídas, gerenciadas e operadas pela organização, por terceiros, ou por uma combinação de ambos. Nuvens Privadas podem ou não fazerem parte da estrutura local da organização;
- Nuvem Comunitária: a infraestrutura da nuvem é fornecida para uso exclusivo de uma comunidade específica de organizações que possuam preocupações em comum (por exemplo, estejam em uma missão, compartilhem requisitos de segurança). Podem ser possuídas, gerenciadas e operadas por uma ou mais organizações da comunidade, por terceiros, ou por uma combinação de ambos, e podem ou não fazerem parte da estrutura local das organizações;
- Nuvem Pública: A infraestrutura da nuvem é fornecida para o público em geral. Pode ser possuída, gerenciada e operada por organizações empresariais, acadêmicas, governamentais, ou por uma combinação de ambos. A infraestrutura existe nas instalações do provedor de nuvem;
- Nuvem Híbrida: A infraestrutura da nuvem é composta por duas ou mais infraestruturas distintas (pública, privada, comunitária) que permanecem entidades únicas, mas são unidas por tecnologia proprietária ou padronizada, que permite portabilidade de dados ou aplicações (por exemplo, balanceamento de carga entre nuvens).

2.2 MOTIVOS PARA IR DA NUVEM PARA FOG

Existem muitas definições para Internet das Coisas, onde todas falam sobre “coisas”, conectividade com a internet e diversas maneiras de como acessá-la (HALLER, 2009)(ACCENTURE, 2014)(CHUI; LOFFLER; ROBERTS, 2010). A definição de Gartner diz: “Internet das Coisas é a rede de objetos físicos que contém tecnologia embarcada para comunicar, sentir ou interagir com seus estados internos ou ambiente externo”(GARTNER, 2013). O que todas elas tem em comum em suas definições é que Internet das Coisas se trata de conectividade, novas “coisas”, interação entre máquinas (Machine-to-Machine, ou M2M), novas

experiências, operações eficientes e modelos de negócios, onde o valor é melhorado ao se fazer melhor utilização dos ativos físicos.

Internet das Coisas traz inovações em diversas direções: na indústria, com fábricas conectadas, energia conectada (Smart Grids), Business to Consumer (B2C), cidades inteligentes, casas conectadas, segurança e proteção. Inovações na indústria trazem melhoramentos de processos e operações de negócios (ZANELLA et al., 2014).

O aumento de sistemas IoT que precisam de requisitos que a nuvem não pode oferecer tende a crescer muitos nos próximos anos. Algumas previsões, como da Cisco, esperam um total de bilhões de dispositivos inteligentes até 2019 (Cisco VNI, 2015b). Também prevê um rápido crescimento de comunicações Machine-to-Machine (M2M) até 2019, com a consolidação do Smartphone como o equipamento móvel mais comum, ocupando 40% do mercado (Cisco VNI, 2015a).

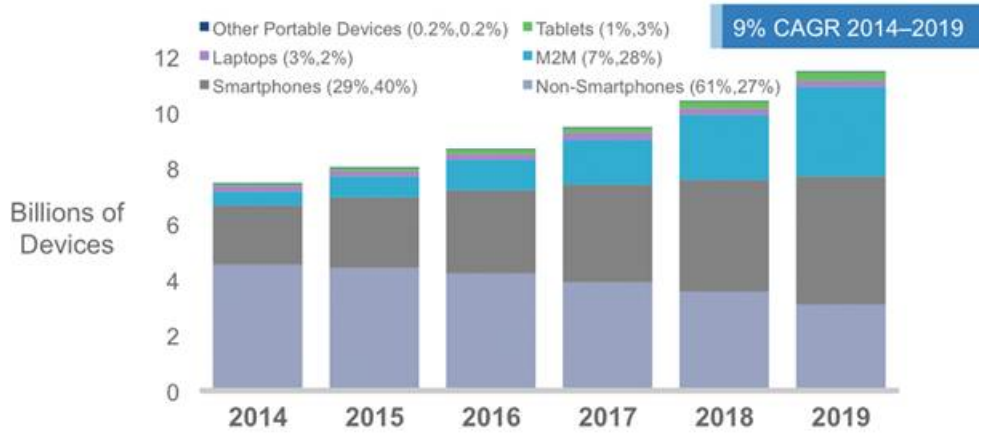

Figura 2.1: Previsão da quota de dispositivos móveis. (Cisco VNI, 2015a)

Tal quantidade de dispositivos leva a um grande problema de escalabilidade, caso nada mude na arquitetura de rede e nos modelos de comunicação. A Internet das Coisas traz à tona diversos cenários que obrigam a repensar o paradigma de Computação em Nuvem existente. A previsão da Cisco VNI sugere que, em grande parte, dispositivos finais vão ser organizados em sistemas sem a presença de humanos interferindo (comunicações M2M), e isso força mudança de visão de apenas indivíduos atrás de dispositivos para adoção de uma visão de um sistema com diversos dispositivos conectados. A rede está se estendendo não apenas com a tecnologia da informação em escritórios e residências, provendo recursos de computação para serviços cibernéticos. Agora passamos para a tecnologia operacional, com equipamentos que monitoram e controlam dispositivos físicos em um mundo ciberfísico, com crescimento do uso de dispositivos inteligentes e interligados em indústrias e fábricas.

Isto leva a algumas observações: a possibilidade de utilizar "coisas" conectadas fornece uma ampla variedade de casos de uso. Em alguns casos os dispositivos estarão geograficamente distribuídos, como em cidades inteligentes ou redes inteligentes. Suporte para alta mobilidade é essencial em alguns casos, como em tráfego inteligente de veículos conecta-

dos. Nesse mesmo caso, procedimentos de emergência precisam de uma latência pequena e previsível, o que também é um requisito em outros exemplos de uso da Internet das Coisas. É interessante notar que essas conclusões estão alinhadas com os atributos de Fog Computing já falados na introdução do capítulo.

Aparelhos na Internet das Coisas estão se tornando cada vez mais poderosos, com maior capacidade de armazenamento, processamento e de comunicação, o que está levando a questões interessantes: tarefas de armazenamento, de controle, de comunicação, de gestão de rede, podem ser feitas no cliente ou perto da borda da rede? É possível fazer isso em uma arquitetura distribuída, em contraste com a arquitetura centralizada da Computação em Nuvem, onde servidores estão possivelmente longe de onde os clientes estão?

Estas questões são interessantes pois muitas coisas não são muito eficientes com a atual arquitetura centralizada. Por exemplo, uma comunicação entre dois clientes de telefonia celular que estão apenas a algumas dezenas de metros de distância percorre um longo caminho de ida e volta entre switches e data centers, onde faria muito mais sentido se comunicarem localmente, diretamente. Até pouco tempo atrás não era viável esse tipo de abordagem pois os equipamentos finais não eram tão poderosos. Mas atualmente, com a Internet das Coisas, temos dispositivos com muito mais capacidade e potencial de realizar tarefas computacionalmente exigentes. Isso não quer dizer que tudo deve ser feito na borda da rede, existem tarefas mais naturais que devem residir no lado da nuvem. Na visão tradicional de Computação em Nuvem temos pequenos clientes usando os serviços prestados pelos grandes datacenters ou grandes gateways, que estão no centro da rede. A visão de Fog Computing, ao contrário, diz que os pequenos dispositivos fazem parte da gestão global da rede, como também participam de tarefas de armazenamento e dos sistemas de comunicação. Ou seja, eles fazem parte da infraestrutura da rede.

2.3 REQUISITOS HETEROGÊNEOS DE FOG COMPUTING E INTERNET DAS COISAS

Ao olhar para todas as diferentes aplicações da Internet das Coisas, algo fundamental que emerge é que temos necessidades heterogêneas dessas aplicações (BONOMI et al., 2014). Elas podem ter diferentes necessidades de energia para os dispositivos, em termos de energia para manter seu funcionamento ou energia para transmitir informações. Alguns destes são alimentados por bateria, podendo até mesmo estarem implantados dentro de corpos humanos. Nestes casos, é extremamente desejável operações energicamente eficientes. Já em outros, onde o dispositivo pode estar ligado em uma tomada, tal eficiência não é tão preocupante.

Outro exemplo é a tolerância a atraso. Algumas aplicações são criticamente dependentes do tempo, onde se queira fornecer uma comunicação com baixíssima latência e com pouco

jitter. como por exemplo um sistema de carros interconectados. Outras podem operar tolerando atrasos maiores, indo de horas até dias.

Algumas aplicações exigem alto throughput, como em uma operação envolvendo vídeochamada, e outras funcionam com baixo throughput, enviando e recebendo apenas alguns kilobytes de vez em quando, como por exemplo em termostatos registrando quais são as temperaturas lidas. Novamente, isto representa várias ordens de magnitude de throughput dentro de aplicações IoT.

Podem ter custos bastante variáveis. Podem custar muito caro, tendo grandes implicações financeiras em empresas, enquanto outros podem ser tão baratos que podem ser comprados aos milhares.

E, finalmente, algumas delas continuam a exigir interação humana. Mesmo falando de Internet das Coisas, ainda vão existir aplicações com humanos efetuando o controle e examinando dados, enquanto outros casos são totalmente automatizados. Como se percebe, aplicações IoT possuem necessidades heterogêneas e é um grande desafio interconectar diferentes dispositivos com funções e requisitos tão diferentes.

2.4 CARACTERIZAÇÃO DE FOG COMPUTING

Fog Computing pode ser definido como uma arquitetura de rede, que usa um ou uma multidão de colaboração de usuários finais, tais como celulares ou computadores, ou dispositivos de borda perto do usuário, como conversores digitais para TV, gateways domésticos ou femtocélulas, para entregar uma quantidade substancial recursos como armazenamento, processamento, comunicação e medidas de controle e gestão de rede (BONOMI et al., 2012).

Essa definição difere do conceito de Computação em Nuvem, onde o armazenamento de dados é feito principalmente em data centers na nuvem, onde as conexões e tráfego são encaminhados através de grandes áreas, como as redes de backbone para a comunicação, e onde o controle é feito principalmente por gateways de rede (ZHU et al., 2013).

Em Computação em Nuvem temos uma arquitetura centralizada, onde data centers concentram uma grande quantidade de servidores, switches e roteadores poderosos, que realizam as funções de armazenamento, processamento e gerenciamento de rede. Na borda da rede temos aparelhos como smartphones e outros aparelhos inteligentes, computadores de bordo de carros, femtocélulas, em geral equipamentos de pequeno porte. Alguns possuem mais mobilidade e as limitações são bastante variáveis, como por exemplo, autonomia da bateria. E esses aparelhos estão muito mais distribuídos e suborganizados em comparação com servidores e roteadores da nuvem.

O modelo de computação em nuvem é um meio eficiente para que consumidores tenham em mãos data centers privados, usando-os para processamento de dados ou para aplicações

relacionadas a Web. A nuvem livra empresas e consumidores de muitos detalhes e custos na implementação de serviços, cobrando-os a medida que os serviços são utilizados sob demanda, a relativamente baixo custo.

Computação em Nuvem permite várias facilidades, mas vêm com um custo: latência. Caso o data center não esteja próximo dos utilizadores, aplicações sensíveis a atrasos perdem desempenho no modelo de computação em nuvem. Essa distância entre a nuvem e os usuários pode ser entendida usando uma analogia: uma pessoa em solo (usuário final) se encontra longe de nuvens reais no céu. E agora, com Fog Computing, a nuvem está descendo para estar dentro, entre ou perto dos dispositivos e usuários finais, na borda da rede. A Computação em Nuvem funciona majoritariamente com recursos que são homogêneos e que funcionam com uma forma centralizada. Fog Computing possui uma arquitetura distribuída para atender os requisitos heterogêneos que estão surgindo com a Internet das Coisas. Fog estende e complementa a nuvem, funcionando na borda da rede e nos usuários finais.

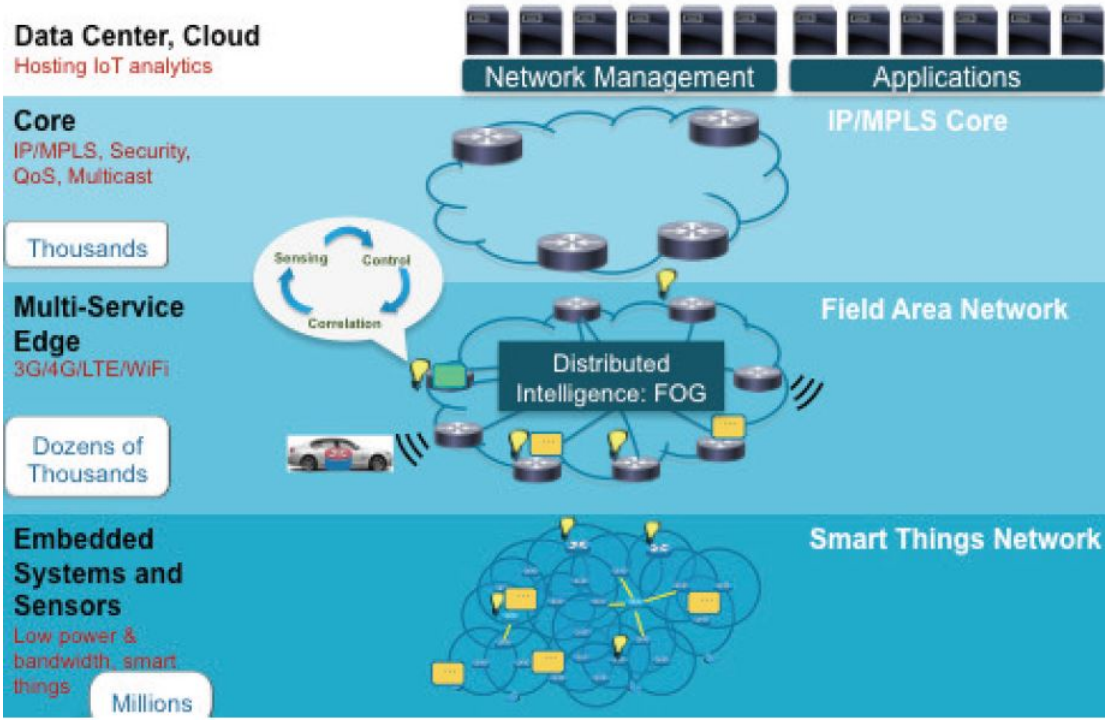

Figura 2.2: IoT e Fog Computing. (BONOMI et al., 2012)

A figura 2.2 mostra uma arquitetura de Fog Computing, distribuída. No topo temos a nuvem, o núcleo da rede com seus serviços de rede e serviços computacionais para Big Data, como armazenamento, agregação, filtragem e limpeza de dados. E então, embaixo, temos a borda da rede e suas redes de acesso, com aplicativos para processamento de dados em movimento. Na borda da rede temos Fog Computing com uma estrutura distribuída para processamento e armazenamento. Nos pontos finais temos grandes ecossistemas, com sistemas embarcados e sensores, que rodam aplicações para analisar e filtrar os dados em tempo real.

Atualmente, o que acontece é que os dados são gerados em grande volume e são enviados para a Nuvem. A tendência atual é que a análise dos dados deve acontecer na borda, sendo filtrada em cada camada. Essa análise não pode mais acontecer na nuvem, pois a quantidade de dados gerados é muito grande. Como consequência, o modelo de computação mudou. Uma arquitetura distribuída traz maior escalabilidade, maior confiabilidade, respostas mais rápidas, flexibilidade para encadear processos e redução de custos. A mudança de Ş ondeŤ acontece a computação já está ocorrendo, a próxima mudança é Ş comoŤ a computação vai acontecer. O paradigma mudou, com a inteligência indo para a borda da rede. Atualmente, com computação em nuvem, capturamos e armazenamos os dados na nuvem, estes dados são analisados em grandes data warehouses, e então são realizadas decisões em cima delas. Mas é muito custoso e difícil mover dados, ainda mais agora com muito mais dados sendo gerados. O futuro precisa que as aplicações se movam para os dados, não que os dados se movam para as aplicações de análise, como acontece atualmente. A diferença essencial entre Computação em Nuvem e Fog Computing é que (ENESCU, 2014):

- Em Computação em Nuvem se armazena os dados, eles são analisados, e são tomadas decisões e ações em cima da análise;
- Em Fog Computing os dados são analisados já na borda, notificações e ações são tomadas, e por fim os dados, já filtrados, são armazenados.

2.5 SÍNTESE

Fog Computing é uma arquitetura distribuída que complementa e estende a Computação em Nuvem. Algumas aplicações que estão surgindo não tão compatíveis com o modelo centralizado da nuvem. Diversas previsões sugerem um grande aumento no número de dispositivos e de tráfego que vai circular pela rede, e isso pede uma mudança de paradigma da computação. Não apenas é desejável ter uma arquitetura distribuída mais perto dos clientes, para reduzir a latência em aplicações. A arquitetura distribuída de Fog Computing é necessária, pois a Nuvem pode não aguentar a enorme quantidade de dados que estão sendo gerados e que serão gerados com a Internet das Coisas.

Ter essa arquitetura distribuída se tornou interessante agora pelo fato dos dispositivos finais estarem cada vez mais potentes e baratos, o que abre a oportunidade de utilizar seus recursos para executarem tarefas que eram majoritariamente funções da nuvem. Mas partir da arquitetura centralizada atual para a arquitetura distribuída em Fog não é algo trivial. Os dispositivos e aplicações da Internet das Coisas possuem requisitos e recursos heterogêneos, o que significa que pode ser difícil criar protocolos e aplicações que sejam interconectáveis e que funcionem bem para diferentes tipos de dispositivos.

Fog Computing usa uma multidão de dispositivos, que podem trabalhar em conjunto,

para entregar serviços que são majoritariamente serviços da nuvem, como armazenamento, processamento, comunicação e gestão da rede. Fog Computing traz mudança de paradigma na computação, com aplicações indo até os dados, ao contrário da Computação em Nuvem, onde os dados vão até as aplicações.

3 FOG COMPUTING COMO ARQUITETURA DE REDE DISTRIBUÍDA

Vários sistemas da Internet das Coisas incorporam componentes tanto do cibernético como do físico, o que leva ao nome de sistemas ciberfísicos, também chamados de CPS (Cyber-Physical Systems). Isso significa que se pode sentir o que há no mundo físico, como também atuar e controlar. Por causa do forte acoplamento entre o cibernético e o físico, vemos em IoT e CPS a necessidade de controle e configuração da rede, geralmente pelos clientes e no lado dos clientes.

Há discussões recentes no Brasil e Estados Unidos sobre regulamentação e neutralidade da rede (RAMOS, 2014)(UN, 2015). O tema sugere algumas regulamentações da rede que poderiam permitir os operadores da rede regular algumas atividades dos clientes por motivos não técnicos (por exemplo, por preço, que é uma forma indireta de gerenciar a rede). Ou seja, o tratamento e diferenciação de bytes pelo operador no interior da rede poderia acontecer por razões não técnicas. Agora, se a inovação da rede está vindo dos clientes (por Fog Computing) dentro de suas residências, e diretamente iniciadas e controladas pelos utilizadores finais e não pelos operadores de rede, isso pode ser um processo bastante diferente. Em outras palavras, as escolhas e decisões em Fog Computing podem ser feitas pelos usuários finais. Na presente data deste trabalho ainda não sabemos o que irá acontecer, mas a implicação potencial é como o controle e configuração da rede pode precisar migrar ainda mais rapidamente do que se imagina, para esse controle voltado ao cliente final.

Por Internet das Coisas e Fog Computing serem ambos emergentes, a arquitetura e domínios de aplicação não possuem um material padrão para serem estudados. Por isso, é interessante ilustrar pontos genéricos através de estudos de casos específicos.

3.1 CONTROLE E CONFIGURAÇÃO VOLTADOS AO CLIENTE

Dispositivos finais, como um computador, smartphone, ou mesmo algum servidor que possui conteúdo a ser disponibilizado, não possuem muita visibilidade da rede ou controle dentro da rede. Eles só podem ver pouco do que está acontecendo, mas precisam medir e inferir o estado da rede para tomarem decisões sobre como fazer o controle de congestionamento, usado pelo protocolo TCP. O dispositivo final só pode controlar a si mesmo, não sendo possível tomar ações pelos outros clientes, e, na verdade, não consegue sequer ver o que está acontecendo com os outros clientes. Em outras palavras, o dispositivo final possui pouquíssima visibilidade e controle limitado.

3.1.1 Princípios Básicos

O primeiro ponto sobre controle da rede que será estudado é sobre controle de congestionamento. O que queremos é extrair algumas das ideias e princípios mais fundamentais para controle da rede pelo cliente final, que estão intimamente relacionadas com a ideia de Fog Computing e o controle e configuração feitos do lado do cliente e pelo cliente. Existem cinco princípios básicos para controle da rede feito pelos clientes na borda da rede (CHIANG, 2012).

O controle de congestionamento do TCP, sem dúvidas, é umas das tecnologias mais importantes que faz a internet funcionar há quase 30 anos, e funciona baseado em alguns princípios. O primeiro princípio é o do controle do dispositivo final. Para isso são necessárias algumas informações de dentro da rede, que chegam via feedback negativo. Este feedback negativo permite ativar o controle de congestionamento pelo cliente, no caso, controlando a velocidade de transmissão. E, assim, indiretamente controlando o congestionamento na rede, feito pelo usuário final. O feedback é recebido através da presença ou da ausência, bem como pela cronometragem de pacotes de confirmação de recebimento, conhecidos como ACK.

O segundo princípio é mais particular do controle de congestionamento do TCP. Trata-se da janela deslizante (Sliding Window). Este é um tipo de princípio que pode ser chamado de incentivante. Na linguagem de teoria dos jogos, pode-se dizer que se cria um mecanismo de frear a natureza egoísta dos utilizadores finais, ao invés de utilizar uma abordagem de grande penalidade. Em outras palavras, o usuário final é recompensado pelo bom comportamento, ao contrário de ser penalizado (por exemplo, ser bloqueado) pelo seu egoísmo , ao aumentar ou diminuir o tamanho da janela de transmissão, que é essencialmente controlar a taxa de transmissão.

O terceiro princípio é sobre aumentar a taxa de transmissão de uma maneira aditiva, e reduzir a taxa de transmissão de uma maneira multiplicativa. Obviamente, não se pode só aumentar o tamanho da janela de transmissão, pois é assim que o congestionamento começa. Os engenheiros e pesquisadores sugeriram que devia ser feito um aumento mais conservador e uma diminuição mais agressiva. A ideia é que quando se tenta aumentar a taxa de transmissão, existe o risco de causar congestionamento, então, deve-se aumentar a velocidade de transmissão lentamente. Mas quando se sabe que há algo errado, infere-se que os dispositivos finais estão causando congestionamento coletivamente, e devem se penalizar rapidamente para tentar cuidar da situação o mais rápido possível.

O quarto princípio refere-se a inferir o congestionamento. Como saber que se deve diminuir a taxa de transmissão? Como saber que há algo errado na rede? Lembrando, se deseja um controle baseado no usuário final e não há visibilidade de toda a rede. A operadora de rede possui esta visibilidade, mas não é a operadora de rede que controla a taxa de transmissão. Em TCP, basicamente, pode-se inferir congestionamento por dois tipos de feedback negativo. Um deles é pela perda de pacote. O outro é o atraso. Como saber que há perda de

um pacote? Se não há ninguém para dizer explicitamente que há uma perda, é necessário um palpite, uma inferência sobre a perda de pacotes. Atraso é sem dúvidas mais fácil de medir. Por meio do uso de um temporizador, se faz a cronometragem o tempo entre o envio do pacote e o recebimento do pacote de confirmação ACK. A diferença dos tempos é chamada de Round-Trip Time (RTT), que é usado para medir o atraso. (CHIANG, 2012)

Uma questão que é levantada é qual feedback deve ser usado, o de perda de pacote ou o de atraso. O consenso emergente nos últimos anos de pesquisa diz que se pode usar ambos (HA; RHEE; XU, 2008). E, certamente, se é possível sentir a perda, deve-se fazer o uso dessa sugestão. Mas o controle de congestionamento baseado em atraso pode ser mais útil. Existem várias razões para isso, mas de forma intuitiva, é possível saber que quando uma perda acontece, já há um congestionamento grave ocorrendo a ponto de transbordar o buffer dos equipamentos de rede. E, considerando o atraso, pode-se ter um indicador mais precoce, possibilitando tomar medidas para amenizar um congestionamento antes que ele fique muito ruim.

Então, finalmente, há o quinto princípio. Como já dito, uma das funções de controle e congestionamento pelo usuário final é de estimar. Estimar a perda de pacotes e o atraso por meio de temporizadores. Se colocarmos tudo isso junto, teremos inferência e medição voltados ao usuário final, que por sua vez impulsiona o controle e configuração voltados ao usuário final. Neste caso, o usuário final está apenas alterando sua taxa de transmissão, mas não altera a rota de seus pacotes. (CHIANG, 2012)

Estes princípios formam a essência de controle e configuração voltados ao usuário final. TCP é um protocolo antigo, de sucesso, que ilustra lições interessantes. Servem de inspiração para um controle efetuado pelo cliente final, que tenha a decidir sua própria taxa de transmissão, mesmo quando se tem visibilidade limitada de toda a rede, e isso é o que se deseja em Fog Computing.

Existem mais algumas considerações. Atualmente, redes celulares e redes sem fio são parte importante do cotidiano de uso de muitas pessoas. E na Internet das Coisas, muitas dessas coisas estão conectadas sem fio. Canais sem fio estão sujeitos a grandes flutuações, que podem acabar corrompendo alguns pacotes, e está perda de pacote não relacionada ao congestionamento. Pode ser devido ao uso de canais ruins, que estão sofrendo alguma interferência, que pode sumir a qualquer momento. Neste caso, reduzir a taxa de transmissão não melhoraria a conexão. Portanto, existem questões complicadas.

3.1.2 HetNets

A coexistência de diferentes tipos de redes é um tema importante atualmente, com a possibilidade de escolha entre redes, por exemplo, WiFi, 4G, 3G. O termo HetNets (Heterogeneous Networks) se refere a redes que conectam diversos tipos de dispositivos usando diferentes tipos de protocolos ou sistemas operacionais , ou seja, são redes heterogêneas

(ARYAFAR et al., 2013). Com a Internet das Coisas, de fato as "coisas" possuem múltiplas tecnologias de acesso a rádio, que são chamadas de RAT. Ao se ter várias RATs (Radio Access Technologies), uma pergunta que surge é sobre qual tecnologia o dispositivo final deve usar. Ou talvez mais importante que isso, seria a pergunta sobre quem decide qual tecnologia deve ser usada, pois poderia ser a operadora de rede ou o usuário final. A operadora de rede decidir qual meio de acesso a rádio o cliente deve usar pode estar relacionada ao quanto tráfego o cliente está usando, ou pode estar relacionado a quanto o cliente paga para utilizar a rede, ou mesmo pode estar relacionado a uma possível regulamentação. A outra alternativa é o cliente decidir, não necessariamente um humano tomando a decisão, mas um dispositivo na borda da rede.

Existem algumas vantagens para ser o cliente o responsável por decidir qual RAT usar. Por exemplo, se cada tecnologia é controlada por operadoras diferentes, elas não podem ter qualquer incentivo para trabalhar com empresas concorrentes. E, quando o cliente final decide por ele mesmo, estará apenas seguindo suas próprias escolhas. Mas existem preocupações, como ziguezagues, que ocorreriam caso o cliente mudasse de tecnologia várias vezes. Especialmente em tecnologias sem fio, várias alterações na escolha de tecnologias a rádio poderiam levar a um efeito de ondulação que poderia retardar a todos. Portanto, não há vantagem óbvia nessa escolha.

Então, voltando a pergunta central, que é se o usuário final tem o poder de decidir qual das tecnologias ele vai escolher para manter sua conexão, qual ele deveria escolher? Esta escolha não pode levar a ziguezagues, rippling effect, e que possa convergir para algo que seja eficiente para se controlado. Por assim dizer, se deve analisar o tradeoff.

O compartilhamento de throughput pelos clientes pode ser feito de diferentes maneiras. Podemos dividi-los em duas classes (ARYAFAR et al., 2013):

- Class-1 Throughput : O throughput ω recebido pelo usuário i no em algum RAT k depende de quantos usuários estão conectados e quais são suas taxas. Na equação 3.1, R é a taxa bruta de transmissão de dados e ω é o que efetivamente se pode alcançar fora do modelo de interferência. Um exemplo é o DCF do WiFi (aproximado). O modelo para este canal WiFi (imagem) diz que é basicamente a média harmônica de suas taxas de dados bruta, e essa é a taxa de transferência que se pode obter. Portanto, nessa classe, diz que o que se pode efetivamente obter depende da taxa de dados bruta de todos os outros utilizadores.

$$\omega_{i,k} = f_k(R_{1,k}, R_{2,k}, ..., R_{n_k,k}) \; \forall i \in N_k \tag{3.1}$$

- Class-2 Throughput: Este outro modelo é mais simples. O throughput ω recebido depende da própria taxa de dados bruta $R_{i,k}$ e o número de concorrentes na mesma frequência ($f_k(nk)$). Por exemplo, múltiplo acesso por divisão de tempo (TDMA), onde existe um determinado número de usuários em uma mesma frequência. No

TDMA os usuários compartilham o tempo, por meio de turnos. Portanto, neste modelo, o throughput só depende do total de usuários em uma mesma faixa, e o quão rápido eles transmitem não importa.

$$\omega_{i,k} = R_{i,k} \times f_k(n_k) \; \forall i \in N_k \tag{3.2}$$

A classe 1 pode ser chamada de throughput sharing (compartilhamento de throughput) e a classe 2 pode ser chamada de time sharing (compartilhamento de tempo).

3.1.2.1 Controle de HetNets pelo Cliente

Em (ARYAFAR et al., 2013) o autor mostra uma forma muito simples de controle pelo cliente,de uma maneira que está de acordo com as características de Fog Computing, com o cliente podendo decidir se quer efetuar o controle.

O algoritmo é: se $\frac{\omega_{i,k'}}{\omega_{i,k}} > \eta$ para T intervalos de tempo, então troca-se de RAT com probabilidade p.

Se o cliente i está no RAT k, por exemplo WiFi, e se pergunta se deve ir para o RAT k' (por exemplo, LTE). A primeira coisa a se fazer é observar a taxa ω (throughput) atual ($\omega_{i,k}$). O cliente não sabe o que os outros clientes estão fazendo, o cliente só pode observar ele mesmo e controlar ele mesmo. A pergunta que o cliente deve fazer é o que ele vai ganhar com a mudança de RAT. $\omega_{i,k}$ pode ser medido diretamente mas $\omega_{i,k'}$ deve ser inferido, pode ser por mandar um sinal de testes e medir alguns resultados (como medir é outro tópico, assume-se que é possível medir com precisão). Olhando para a razão entre os dois deve-se ver qual seria o ganho e se seria melhor trocar de RAT. η indica o quão melhor é preciso estar para realizar a mudança. Se $\eta = 2$ indica que o throughput seria o dobro do que se tem atualmente. Um η maior significa um cliente mais conservador, que não muda de RAT a não ser que o outro RAT esteja realmente bem mais rápido. Se η for pequeno (não menor que 1, afinal, se for menor que 1 significa que o throughput atual é melhor do que se trocasse de RAT), por exemplo $\eta = 1,1$, significa um cliente agressivo. Um RAT com throughput 10% maior seria suficiente para realizar uma troca. Esta comparação é feita em T intervalos de tempo, e a troca de RAT seria realizada de uma maneira aleatória com probabilidade p, pois, caso contrário haveriam diversos clientes trocando de RAT ao mesmo tempo, e $\omega_{i,k'}$ seria bem diferente do esperado. Isto é um algoritmo de controle HetNet feito pelo cliente, feito de uma que se encaixa com os princípios de Fog Computing.

3.2 MEDIÇÕES E INFERÊNCIAS SOBRE O ESTADO DA REDE VOLTADOS AO CLIENTE

Nesta seção serão abordadas maneiras de como um cliente final pode medir e inferir o estado da rede, fazendo tudo no lado do cliente, na borda da rede. Fazer a medição do estado da rede dentro da rede (na nuvem) é uma abordagem muito diferente. Fazê-la no lado do cliente, seguindo os princípios de Fog Computing, abre oportunidades únicas. Às vezes pode não haver escolha, a não ser realizar a medição do lado do cliente, por ser necessária uma tomada de decisão em tempo real. Pode ser preferível executar uma ação em tempo real, mas com um pouco de ruído e não perfeita, antes que seja tarde demais para tomar uma decisão em um sistema ciberfísico, do que esperar por dados mais precisos vindo da nuvem. Ao mesmo tempo, por ser do lado do cliente, tais decisões e medições tendem a ser mais desafiadoras, abrindo preocupações com o tradeoff de quão preciso são os dados recebidos versus quantos recursos serão gastos fazendo as medições.

Na borda da rede, podemos ter diversos dispositivos fazendo medições sobre a rede e eles podem colaborar uns com os outros, ou apenas um dispositivo cliente medindo as condições da rede. E eles podem fazer a medição direta sobre a própria rede, ou podem tentar fazer uma inferência indireta sobre outras redes, por exemplo, o cliente querer saber o estado de uma RAT sem estar nela.

Medição direta por apenas um cliente é o resultado clássico, como o controle de congestionamento do TCP, que deve inferir o estado da rede antes de efetuar o controle, usando temporizadores para inferir perdas ou calcular atraso.

Mobile Crowdsensing é o caso onde diversos dispositivos se ajudam para medir o estado da rede (GANTI; YE; LEI, 2011). Inferência indireta é um caso de crowdsensing mais difícil, e quando se quer uma inferência indireta feita apenas por um dispositivo, temos o caso mais desafiador.

Para começar a estudar esses casos, é preciso primeiro começar respondendo uma pergunta básica, sobre o que, afinal, se deseja inferir. Existem pelo menos três coisas que gostaríamos de inferir.

O primeiro seria o throughput $\omega_{i,k}$ falado na seção 3.1.2, sobre HetNets. Também é desejável inferir a carga da rede, por exemplo, dos resource blocks (blocos de recursos - pequenos blocos na banda de frequência e tempo) de redes 4G. Saber quantos blocos de recursos estão sendo utilizados e quantos estão livres é uma maneira de medir a carga de rede. Ou pode-se fazer a inferência no nível de aplicação, para entender o que está acontecendo na sessão. Essas três coisas são desejáveis de serem medidas e inferidas.

A primeira coisa a ser estudada será a inferência de throughput. Existe uma série de quantidades que se pode tentar inferir, alguns estão na camada de radiofrequência e outros na camada física , e uns são mais úteis que os outros. As terminologias RSSI, SNR, RSRP, e

RSRQ são usadas para medição do nível de RF, e que podem se relacionar com o throughput e com quanto se pode atingir em termos de bits por segundo.

Uma terminologia comum é o RSSI (Received Signal Strenght Indicator –Indicador da força do sinal recebido). O valor do RSSI fica tipicamente entre $-100dBm$ e $-60dBm$. O RSSI é frequentemente usado em redes WiFi, e o indicador é geralmente representado por uma percentagem. RSSI é apenas uma medição da intensidade do sinal, não mede a intensidade da interferência. (SAUTER, 2010)

Em redes sem fio, a razão entre o sinal e o ruído é uma medida mais interessante. De fato, existe a razão chamada de SNR (Signal to Noise Ratio), que mede exatamente a proporção entre o sinal útil e o ruído. Então, SNR tende a ser mais útil, por não olhar apenas a intensidade do sinal, mas por também olhar a intensidade da interferência e do ruído.

Em redes celulares existem termos com diferenças sutis, mas pode-se classificar o RSRP (Reference Signal Received Power) como um equivalente de rede celular ao RSSI, e o RSRQ (Reference Signal Received Quality) como equivalente ao SNR. Qualidade aqui se refere à razão, portanto, pode-se dizer em geral que RSRQ é um indicador mais confiável da qualidade do sinal que importa para o cálculo de throughput. Quantos bits por segundo se pode transmitir é diretamente impactado pelo tipo de codificação em esquemas de modulação, que, por sua vez, é impactado por medidas como SNR ou RSRQ.

Estas quantidades podem ser medidas do lado do cliente. Seguindo uma tabela de referência, é possível inferir qual "nível de modulação" pode ser alcançado. Modulações de alta ordem permitem enviar mais bits por bloco (chunk) de energia, que dá uma maior taxa bits por segundo. Se o canal estiver "limpo", com pouco ruído, é possível enviar bits a uma taxa mais alta, enquanto em um canal com muito ruído e interferência, a transmissão terá taxa de transmissão mais devagar. Então, ao inferir SNR ou RSRQ, se pode ter uma noção de quanto throughput pode ser alcançado. E se pode usar a inferência de throughput para uma variedade de coisas.

3.2.1 Rate Adaptation

Pode-se usar a inferência de throughput para adaptação de taxa (Rate Adaptation), trocando os esquemas de modulação e de codificação. Ao se ter um melhor SNR ou RSRQ, um cliente pode ser capaz de mudar sua modulação para uma de ordem maior. Mas às vezes é necessário um grande número de medições para ter uma boa precisão em determinar as condições do canal.(PEFKIANAKIS et al., 2013)

Existem alguns esquemas propostos. Uns tentam adaptar a taxa baseado em SNR (SNR-Triggered), tanto por adaptação de taxa pacote-por-pacote baseado na melhor taxa corrente , ou em sistemas baseados em perda de pacote (Packet Loss-Rriggered), onde se tem um histórico de janela de entrega de pacotes bem e mal sucedidos para diferentes taxas de trans-

missão.

- SNR-Triggered: infere informação do estado do canal, no receptor, baseado na força do sinal de pacotes de controle. É uma adaptação de taxa p̈acote-por-pacoteb̈aseada na melhor taxa (uma função de tempo de coerência) empírica ou derivada pelo último SNR recebido. Possui um histórico informando basicamente como foi o desempenho anterior, se era possível ter uma comunicação mais rápida ou se a taxa estava alta demais, e agora é melhor reduzi-la (CAMP; KNIGHTLY, 2010);
- Packet Loss-Triggered: usa o passado recente de consecutivos pacotes perdidos como o maior indicador sobre o que fazer ou não fazer com a adaptação de taxa. Por exemplo, se aumenta a taxa caso 10 pacotes consecutivos tenham sido enviados com sucesso, e diminui a taxa caso 2 pacotes consecutivos tenham sido perdidos. Um exemplo é o Robust Rate Adaptation Algorithm (RRAA) (WONG et al., 2006)).

3.2.2 Resource Block / Inferência de carga da rede

Outro tema que se pode querer inferir é a condição de carga da rede. Quando se tenta inferir a carga de rede em tempo real do lado do cliente, se tem uma gama de opções e ações que podem ser tomadas. Resumidamente, podem ser classificadas em: Medição Passiva, Probing Ativo, Correlação de performance da aplicação e histórico de data mining.

Medição passiva, inclui, por exemplo, medir o tempo de ida e volta (Round-Trip Time), ou as razões SNR ou RSRQ.

Active Probing é quando se envia um trem de pacotes (Packet Train) de prova. Alguns estudos foram feitos sobre quais Packet Train devem ser enviados, onde basicamente se envia alguns pacotes, talvez uma sequência deles, separados por alguns instantes, e cada um deles tem certa magnitude (por exemplo, quantidade de Kb). Então, se mede cuidadosamente o que acontece com esses pacotes, em particular seu RTT. Eles são pacotes ativos de teste, o que significa que são uma sobrecarga (overhead), não representando uma informação real que se deseja enviar, realmente servindo apenas como medição.

Outra maneira é olhando para a performance na camada de aplicação, variando de o número de telas congeladas até latência de buffer necessário para a reprodução de um vídeo. Diversas métricas residem na camada de aplicação e tentam correlacionar o que acontece dentro da rede com essas métricas a nível e usuário.

O quarto meio de inferir a condição de carga da rede é pelo histórico, com data mining e o aprendizado de máquina (machine learning). Se observa o que está acontecendo em um determinado horário, dia e/ou local, em uma determinada aplicação. E se usa um riqueza de dados históricos para ajudar com a limitação de poucos ou apenas um cliente tentando medir a rede. Pode-se olhar uma tabela (lookup table) para ajudar em um começo de estimações futuras.

Em particular, se quiser usar pacotes de prova (Active Probe), que incorre overhead, ter vários clientes trabalhando juntos pode ajudar. Ao invés de colocar todas as active probes em um único cliente, por exemplo em algum setor (célula), pode-se espalhá-los em vários deles. Então, enquanto um está realizando um active probe, os outros clientes podem estar em espera, aguardando seu turno. Desta forma, espalha-se o overhead em vários dispositivos tentando fazer medidas correlatas sobre o mesmo setor.

3.2.3 Inferência de uso de espectro

Inferência de utilização do espectro para rádio cognitivo ou para inferir espaços não utilizados foram bem estudadas (LIANG et al., 2011) (TU; CHEN; PRASAD, 2009). É útil em redes de tomada de decisões voltadas ao cliente (redes de rádio cognitivo como um exemplo). Técnicas envolvem um transmissor monitorando ativamente o espectro para inferir a disponibilidade via Bayesian Risk em uma informação a priori (QUER et al., 2010). Trabalhos adicionais para examinar a disponibilidade de espectro em nível de link de rede para inferir disponibilidade nó-a-nó foi estudado associado com diversidade cooperativa (LIANG et al., 2011). É possível receber a ajuda de um banco de dados centralizado dentro da nuvem, e então, com a interface Nuvem-Fog, usar isso para a tomada de decisão voltadas ao cliente e rádio cognitivo com sensores no lado do cliente.

3.2.4 Inferência no nível de Aplicação

O terceiro tipo é inferência no nível de aplicação. Isto é mais difícil, porque não dá para saber qual aplicação está sendo usada por um cliente, bem como não dá para saber o que está dentro dos pacotes. A abordagem padrão, mais robusta, é chamada de DPI (Deep Packet Inspection) (DAINOTTI; PESCAPE; CLAFFY, 2012). Uma abordagem mais leve é a análise de porta: as comunicações acontecem em um determinado número de porta, e é possível saber alguns aplicativos associados, que possuem sempre ou quase sempre um número específico. Ou se pode também utilizar a abordagem de aprendizagem de máquina, com técnicas para entender suas assinaturas ou padrões temporais. Certas aplicações tendem a gerar pacotes de um jeito particular, e isso dá uma assinatura temporal para inferir a aplicação (HAFFNER et al., 2005). Inferência na camada de aplicação na borda da rede pode ser uma boa alternativa, pois estando dentro da rede é difícil e custoso inferir o estado da rede em nível de aplicação, enquanto estar no lado do cliente ou perto da borda, como no próprio smartphone ou mesmo um gateway (roteador WiFi em casa), se está muito perto onde todas as sessões estão acontecendo. É mais barato e mais viável, sem violação de privacidade, para entender e classificar os atributos de cada aplicação para cada sessão.

Deep Packet Inspection pode comparar o conteúdo dos pacotes após lê-los, e isso cria grande preocupação com a privacidade e potenciais limites legais. Além de preocupações com privacidade, está se tornando menos útil devido ao aumento de comunicações cripto-

grafadas, e os pacotes podem ser encapsulados em outras camadas, o que torna cada vez mais difícil usar DPI dentro da rede. Então, ao invés de por o DPI dentro da nuvem, pode ser melhor usar DPI no cliente ou perto do cliente.

Outra maneira de inferir a rede na camada de aplicação é através de análise de porta (DAINOTTI; PESCAPE; CLAFFY, 2012). Uma desvantagem é que aplicações podem trocar o número da porta usada por diversos motivos, como por exemplo, ao estar atrás de um firewalls. E muitas vezes uma porta pode ser usada por muitas aplicações diferentes (problemas de porta comum). Seja pelo robusto DPI ou a leve análise de porta, uma vez que o tráfego sai da borda, torna-se cada vez mais difícil de fazer inferência no nível de aplicação, e a borda tende a ser o local ideal para estuda-lo.

Seja qual for a camada que se deseja inferir, ou independente de se inferir throughput, resource block ou aplicações, é sempre melhor ter não apenas um, mas ter N dispositivos trabalhando em conjunto, uma multidão de colaboração, trabalhando para terem uma medida mais precisa da rede. É sempre mais difícil ir de inferência direta para indireta, com a inferência indireta feita por apenas um dispositivo sendo a mais difícil, mas, com uma combinação de técnicas mencionadas, essas inferências se tornam cada vez mais interessantes e realizáveis. E, certamente, para a inferência na camada de aplicação, a borda se torna a principal fonte de ações potencialmente efetivas, quando se fala de medições e inferências.

3.3 POOL DE RECURSOS E CACHING NA BORDA DA REDE

Esta seção aborda pooling e caching na borda da rede. Aqui será explorado como usar o armazenamento em cache na borda, armazenar o conteúdo mais próximo do cliente e evitar armazená-lo na nuvem, evitando atravessar todo o caminho até a nuvem. A vantagem é essencialmente diminuir a latência ou mesmo mitigar alguns gargalos que podem acontecer com todos enviando dados para a nuvem. Pooling de recursos distribuído se trata de combinar os dispositivos de borda, com suas capacidades computacionais, de armazenamento, de conexão, de sensoriamento, para fazer um pool de recursos descentralizado, usando algo similar a redes de malha (redes Mesh). Um usuário pode ter bastante armazenamento, o outro pode ter grande poder de processamento, e o problema é como gerenciar tal rede para criar uma computação decentralizada, para trazer mais robustez ao sistema, bem como melhorar a performance e torna-los mais eficientes em energia. Alguns problemas e soluções que serão mostrados tratam disso, e será mostrado que não se trata de um problema trivial para ser resolvido, trazendo tradeoffs entre as capacidades de robustez, performance, eficiência, etc.

A fim de compreender por que precisamos de pooling e caching na borda da rede, devemos dar uma olhada na arquitetura de computação móvel. A computação em nuvem é às vezes percebida como se os clientes interagissem diretamente com ela, usando seus serviços para armazenamento ou alguma outra tarefa computacional. Especialmente quando se trata

de comunicação por telefonia móvel, há um caminho que deve ser atravessado primeiro. Primeiro há a conexão com a estação-base, em seguida com a rede de telefonia celular, e por fim a internet, até chegar na nuvem. Isto abre uma série de oportunidades em termos de offloading de algumas tarefas da nuvem para mais perto da borda, nos dispositivos ou na borda em si. Assim, nessa seção, serão mostradas algumas maneiras de como isso pode ser alcançado, a fim de permitir que um pouco das tarefas que a nuvem executa seja direcionada para a borda da rede, a fim de reduzir a carga na nuvem e diminuir a latência entre processos. Isso é muito importante, pois na Internet das Coisas teremos mais dispositivos do que nunca. Não apenas teremos mais alguns bilhões de dispositivos, eles também consomem mais dados do que nunca, e de acordo com a Cisco, o número de dados trafegando será de aproximadamente 100 exabytes por mês (Cisco VNI, 2015b). Este número é realmente grande, e que põe muita carga sobre a infraestrutura em nuvem.

Por que não usar a nuvem? Embora a nuvem tenha revolucionado a maneira como construímos sistemas e todo mundo esteja tentando usá-la (pois realmente traz algumas facilidades), este aumento no número de dispositivos e de dados trafegados está fazendo a nuvem se tornar muito cara para ser construída, mantida e usada. Obviamente, alguns projetos continuarão a ser executados nela, mas o custo de manter uma arquitetura global em nuvem é cada vez maior. Além disso, os clientes estão localizados relativamente longe da nuvem, o que gera o problema de latência. Quanto mais dispositivos, mais dados eles consomem, mais recursos computacionais eles precisam, e mais surgem problemas relacionados a latência. Portanto, existem alguns argumentos para que a nuvem seja desafogada. E se é possível ficar mais perto da borda para offloading de algumas das tarefas da nuvem, pode ser uma boa ideia fazer isso.

Então, há uma oportunidade de mudar o cenário atual, devido à entrada de novos dispositivos de borda. Recentemente uma grande quantidade de armazenamento começou a estar disponível, seja em smartphones, tablets, ou qualquer outro dispositivo móvel. E eles também estão cada vez com maior poder computacional, fazendo da computação móvel algo extremamente poderoso. São baratos, em relação ao custo de construir e manter uma infraestrutura de Computação em Nuvem, e eles estão sempre nas proximidades. Por que não explorar este novo cenário para realizar algumas tarefas sem ter que depender da nuvem? A oportunidade, essencialmente, é descarregar (offload) o trabalho para a borda. É possível usar a borda, por exemplo, para combinar diversos dispositivos para um armazenamento distribuído. Ou para um processamento em conjunto, descarregando a rede e realizando algumas tarefas computacionais em dispositivos próximos, sem a necessidade de ir e voltar da nuvem. E também há um ponto importante, que é muito discutido hoje em dia, que é sobre a segurança e privacidade. Realizar as tarefas computacionais mais perto da borda, ou no dispositivo em si, permite mais controle sobre a segurança e preserva a privacidade, o que é, na maioria dos casos, muito mais difícil, quando se tem que confiar ou descarregar parte de um trabalho para um provedor terceirizado na nuvem. Ao delegar tarefas para um provedor na nuvem, de alguma maneira se tem que confiar que a companhia seguirá as polí-

ticas de tratamento de dados e que realizará as tarefas computacionais do jeito desejado pelo consumidor.

Tendo a habilidade de fazer isso na borda permite ter mais controle sobre a segurança e mais privacidade do que antes. Isto é um tópico muito importante que será explorado no capítulo 4.

Estas oportunidades podem ser limitadas a três questões: a primeira é sobre o que se deseja descarregar (offload) para a borda ou perto da borda. Usualmente se trata de conteúdo (armazenar algo nos dispositivos) ou tarefas (computar algumas tarefas por perto e descarregar um pouco da computação); a segunda é sobre onde realizar o offloading. Como dito, existem múltiplos dispositivos perto com diferentes características, e se deve saber onde armazenar, para onde delegar tarefas, e onde recuperá-las. Saber quão seguros e confiáveis são os dispositivos é algo fundamental para onde realizar o offload; a terceira é sobre o quando. Aqui temos o problema do sincronismo, especialmente em sistemas distribuídos. É preciso saber onde o descarregamento irá acontecer e se será sincronizado, para a computação ser coerente e ter o retorno de resultados precisos e consistentes (KUMAR et al., 2013).

Muito disso tem que ser coberto em qualquer solução, e essas três questões podem resumir o que todo sistema usar para pooling e caching na borda da rede.

3.3.1 Princípios Básicos

Quando se trata de sistemas distribuídos, existem desafios como disponibilidade, consistência e tolerância a falhas. O teorema de CAP diz essencialmente que só se pode ter duas das três capacidades. Se deseja ter as três capacidades, existem soluções para as três capacidades, mas uma sempre virá como tradeoff (BREWER, 2012).

A disponibilidade permite garantir que se há uma coleção de dispositivos distribuídos, e se algum conteúdo é colocado em um dos dispositivos, se deve ser capaz de recuperá-lo de qualquer lugar, em qualquer ponto da rede.

Consistência está relacionada com garantir que, se algum conteúdo é mudado para outra versão, o sistema deve ter (fazer) uma cópia consistente em todo o sistema.

E a tolerância a falha diz que se um ou mais nós apresente alguma falha, se deve ser capaz de recuperar o conteúdo em outro nó.

Trabalhos e discussões a respeito de sistemas distribuídos que tentam olhar e resolver esses problemas são muito relevantes para Fog Computing, em pooling e caching de borda em particular.

Por exemplo, para disponibilidade, pode-se distribuir cada cópia em todos os lugares, e se é necessário recuperar algo, haverá uma cópia por perto. Para consistência, usam-se alguns métodos de sincronização que garante que as cópias estão sincronizadas, e se

alguma mudança acontecer, elas serão sincronizadas e atualizadas, impedindo que se acesse cópias não sincronizadas. E tolerância a falha, de algum modo relacionado a redundância e réplicas, então se o conteúdo for replicado através de múltiplos componentes do sistema, se um falhar haverá uma cópia, quem sabe sincronizada, em diferentes outros nós que possam ser acessados. Também existem mecanismos de auto cura que podem ser usados para recriar conteúdo, e alguns sistemas também usam isso.

3.3.2 Estado da Arte

Agora será analisado um pouco do estado da arte da abordagem de offloading computacional. Uma das abordagens é a virtualização, que é feita essencialmente usando máquinas virtuais. Se existe uma máquina com poder de processamento muito maior que um dispositivo próximo, pode ser desejável descarregar algumas tarefas para aquela máquina. Alguns sistemas podem permitir que isso seja feito, rodando uma máquina virtual com recursos dedicados.

Abordagem	Prós	Contras	Exemplo
Virtualização	Isolamento de Recursos; Sem necessidade de software especial no cliente	Atrasos pela inicialização da VM	Cloudlets MAUI CloneCloud MobiCloud
Agentes móveis (Cyber Foraging)	Implementação dinâmica	Problemas com gerenciamento de agentes	Scavenger

Tabela 3.1: Visão geral de abordagens de offloading. (FERNANDO; LOKE; RAHAYU, 2013)

Então, essencialmente, um cliente não precisa de nenhum software especial, e a tarefa será executada na máquina virtual com recursos dedicados, retornando os resultados para o cliente após a computação. Um problema é que leva um tempo para a máquina virtual ser rodada (ligada), o que causa um atraso.

Uma outra abordagem é Cyber Foraging que é essencialmente uma maneira de aproveitar o ambiente ad hoc de múltiplos dispositivos, que possuem diferentes interfaces de comunicação, diferentes CPUs e diferentes propriedades (KRISTENSEN, 2010). Procura-se um meio de descarregar algumas tarefas e encontrar uma maneira de receber os resultados a tempo. O benefício dessa abordagem é o aproveitamento dos sistemas ad hoc, que são bem dinâmicos. O problema é a dificuldade de gerenciar tal ambiente.

Antes de continuar com a descrição de alguns sistemas, é importante lembrar que eles possuem outros tradeoffs e outras considerações quando eles foram projetados. Uma dessas considerações são os protocolos de comunicação. Cada dispositivo tem um ou múltiplos protocolos de comunicação. Então, é muito importante saber qual protocolo é utilizado. Por exemplo, quando se trata de WiFi se sabe que tem boa cobertura e é energeticamente efi-

ciente, porém está sujeito a congestionamento. Já o 3G possui grande consumo de energia e menor taxa de transferência do que WiFi, mas possui boa cobertura. 4G LTE tem bom throughtput, mas ainda não é suportado por muitos dispositivos e tem baixa cobertura, dependendo da localidade. Então, diversas considerações devem ser levadas em conta na hora de projetar sistemas que suportem virtualização ou Cyber Foraging, ou qualquer outra tarefa de offload, pois se deve saber qual protocolo de comunicação é mais eficiente para executar determinada tarefa, e isso não é trivial (FERNANDO; LOKE; RAHAYU, 2013).

3.3.3 Cloudlets

Cloudlets são "pequenas nuvens" basicamente construídas em virtualização (SATYANARAYANAN et al., 2009). É um conceito que envolve proximidade, infraestrutura computacional e que pode ser aproveitada por dispositivos móveis. Um dispositivo móvel geralmente está longe da nuvem, que é uma infraestrutura poderosa e que todo mundo está usando-a. Então, a ideia de Cloudlets é que elas são pequenas nuvens que não precisam de toda a infraestrutura e poder da nuvem, e os dispositivos móveis se conectariam ao Cloudlet ao invés da nuvem. Agora, cada Cloudlet deve ser capaz de prover recursos capazes de servir um número limitado de dispositivos móveis que estejam próximos, e se conectaria a nuvem quando fosse necessário. Dispositivos se conectariam a uma Cloudlet ao invés da nuvem, para offload da computação ou para requerer computução, mas a Cloudlet seria um servidor em escala muito menor, necessitando de muito menus recursos para mantê-la.

Cloudlets podem ser chamados de "data center in a box", ou seja, um mini data center. É autogerido, pode ser um pequeno cluster de computadores interconectados, podendo ser muito poderoso. E então dispositivos móveis próximos podem tentar encontrar e se conectar a ele. E caso ele não seja poderoso o bastante, ou os recursos estejam esgotados ou não tenha Cloudlets por perto, continua sendo sempre possível se conectar diretamente a nuvem. A chave é que para cada um dos clientes, se tenha uma iniciação de uma máquina virtual que irá dedicar recursos computacionais a eles. Como dito, é um exemplo de virtualização para descarregar tarefas.

3.3.3.1 Arquitetura

A figura 3.1 mostra um exemplo de um sistema chamado de Kimberley (SATYANARAYANAN et al., 2009), em que dispositivos móveis rodariam uma instância de um agente Kinberley, e usariam o protocolo Avahi , que é uma forma de descobrir serviços de rede, para descobrir a infraestrutura do servidor. A Cloudlet irá instanciar uma VM para cada um dos clientes, realizar os cálculos e retornar os resultados. E, há basicamente uma pilha de software (software stack) no dispositivo móvel e uma infraestrutura dedicada para permitir a criação de máquinas virtuais para cada cliente. Os desafios de Cloudlets são bem alinhados com os problemas já discutidos, sobre o "o que", o "onde" e o "quando".

Além disso, a ideia é servir diferentes tipos de dispositivos. Como já se sabe, dispositivos na Internet das Coisas possuem capacidades e características distintas, e VMs são usadas para permitir que se tenha poucas restrições no lado de software, e de uma maneira que clientes não necessitem fazer nada. Uma VM em uma Cloudlet irá executar tarefas usando recursos dedicados, retornando os resultados aos clientes.

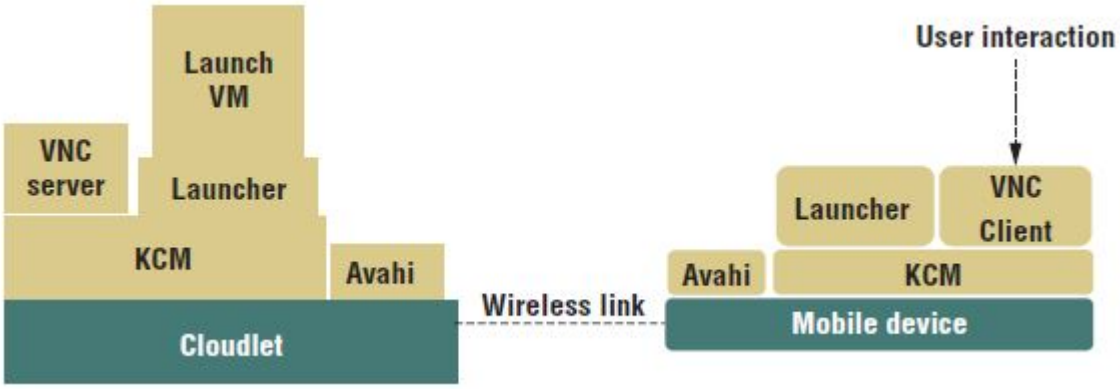

Figura 3.1: Componentes do Kimberley. (SATYANARAYANAN et al., 2009)

3.3.3.2 Síntese de máquinas virtuais (VMs)

Em detalhes, o processo ocorre assim (SATYANARAYANAN et al., 2009): o primeiro passo é o dispositivo móvel descobrir e se conectar com uma cloudlet, negociando o uso de recursos da cloudlet. Em seguida a cloudlet inicia uma VM que irá executar as tarefas desejadas, onde o cliente poderá executar diversas tarefas. Uma vez terminado o processo, pode-se criar um resíduo (VM Resudue), que é essencialmente uma maneira de manter alguma sessão (estado), e o cliente pode receber esse resultado e sair. Após algum tempo, a VM pode ser destruída. Isso é uma maneira dinâmica de alocar VMs, ilustrada na figura 3.2.

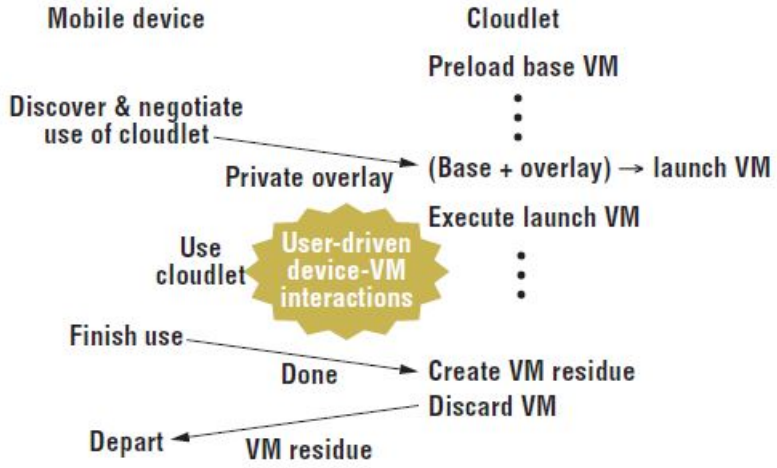

Figura 3.2: Síntese dinâmica de uma VM. (SATYANARAYANAN et al., 2009)

Então, esta é uma ideia para isolar recursos, permitir offloading de tarefas, e essencialmente cobre todos os objetivos que são desejáveis quando se quer descarregar alguma tarefa para a borda da rede.

Os autores também sugerem uma maneira de melhorar o desempenho, encontrando uma forma de transferir dados em taxas mais rápidas, usando uma maior largura de banda para redes sem fio de curto alcance. E enfatizam que ainda é preciso muita pesquisa para suportar dispositivos móveis. Falam sobre reduzir os tempos de descompressão das VMs, é isto é mais sobre uma maneira particular de iniciar VMs e transformar os resultados. Abordam sobre uma maneira de permitir multitarefas serem feitas em paralelo, que pode acelerar o desempenho, e uma melhor maneira de gerenciar VMs, aproveitando um pouco da localidade temporal e padrões de mobilidade para encontrar uma maneira de rodar as VMs antecipadamente. (SATYANARAYANAN et al., 2009)

3.3.3.3 Comparação entre Cloudlets e a Nuvem

A tabela 3.2 mostra como Cloudlet contrasta com nuvem, e como é diferente da nuvem. Em geral, mostra que uma Cloudlet é muito menor que a nuvem, sendo Cloudlet apenas "soft state", que significa que a virtual machine é construída e destruída, não havendo algo dedicado por tempo indeterminado. São auto gerenciadas, ao contrário de gerenciado por uma equipe profissional 24 horas. É basicamente menor, abastecendo uma região geograficamente menor, e não requer algo tão grande como uma sala de máquinas com controle de temperatura e potência. Cloud é centralizada, com empresas grandes, enquanto Cloudlets são descentralizadas, sendo propriedade de negócios locais. Possuem latência e largura de banda de uma rede LAN, enquanto a nuvem possui latência e banda da internet. E o número de usuários é muito menor do que as centenas de milhares que usam a nuvem ao mesmo tempo.

Portanto, isso mostra que já foi feito um trabalho que descarrega as tarefas da nuvem em pequenas entidades mais perto dos usuários, mais perto da borda, livrando os clientes finais de terem que ter comunicações atravessando longos caminhos para ter suas requisições atendidas. (SATYANARAYANAN et al., 2009)

3.3.4 Cyber Foraging

Cyber Foraging é uma ideia de particionamento de tarefas, delegando-as paralelamente a múltiplos dispositivos vizinhos, que são chamados de surrogates (substitutos). Essa técnica envolve, primeiramente, identificar e detectar surrogates, passando para negociação (similarmente como com cloudlets) e então o planejamento das tarefas a serem executadas. Ou seja, objetivos muito semelhantes, mas diferentes maneiras de interação. Novamente, as perguntas feitas são "o que se deseja descarregar", "quais tarefas", "quais surrogates serão usadas"

	Cloudlet	Nuvem
Estado	Soft State	Hard e Soft State
Gerência	Autogerido; Pouca ou nenhuma atenção profissional	Administrado por profissionais, 24 horas
Ambiente	"Datacenter in a box" como premissa	Casa de máquinas com condicionamento de energia e refrigeração
Propriedade	Descentralizado, propriedade de negócios locais	Centralizado e propriedade de grandes empresas
Rede	Latência e largura de banda de LAN	Latência e largura de banda da internet
Compartilhamento	Poucos usuários ao mesmo tempo	Milhares de usuários ao mesmo tempo

Tabela 3.2: Comparação entre Cloudlet e Nuvem. (SATYANARAYANAN et al., 2009)

e “quando estarão disponíveis” que é basicamente planejar (agendar). Essas três questões (o que, onde, quando) sempre estão em mente quando se planeja esses tipos de sistema.

Scavenger é um sistema usado para demonstrar Cyber Foraging (KRISTENSEN, 2010). O “o que” em Scavenger é qual tarefa computacional será escolhida para ser descarregada. Ele precisa de surrogates, que é o “onde”, e o “quando” é usando perfis de tarefas usando os surrogates certos. O onde e o quando são bem conectados aqui.

A figura 3.3 ilustra uma rede ad hoc onde poderia ser usado Cyber Foraging. Um computador dedicado é mostrado como um surrogate dedicado, que geralmente possui mais poder de processamento que os outros dispositivos ilustrados. Os dispositivos poderiam usar esse computador para solicitar que ele execute algumas tarefas para eles.

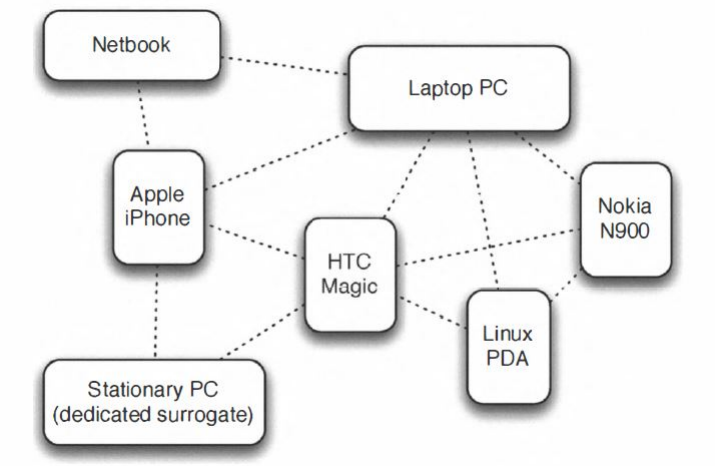

Figura 3.3: Um ambiente de computação pervasiva com rede ad hoc onde pode haver cyber foraging. (KRISTENSEN, 2010)

É interessante notar como Cyber Foraging é uma técnica de computação pervasiva, onde pequenos dispositivos móveis podem solicitar a computadores mais robustos na vizinhança para efetuarem tarefas mais computacionalmente intensas.

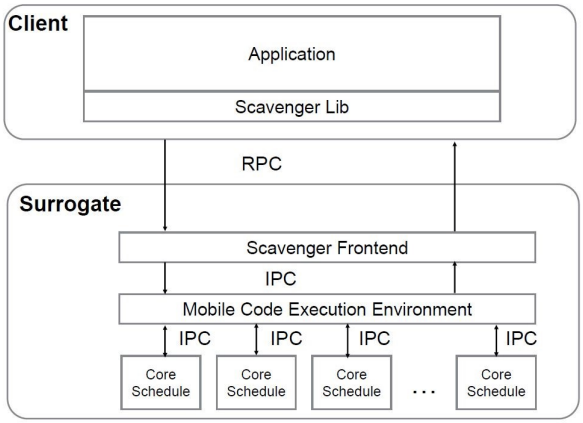

Figura 3.4: Arquitetura do Scavenger. (KRISTENSEN, 2010)

A figura 3.4 mostra a arquitetura de Scavenger, que é bem simples, composta da aplicação, as trends nos clientes (dispositivos móveis), e a comunicação com os surrogates é feita por RPC (Remote Procedure Calls). O surrogate terá algum tipo de estrutura para receber procedure calls e processar em um ambiente de execução, com algo dedicado para isso, que irá resolver as tarefas que as aplicações requisitam e retornam o resultado para a aplicação, bem similar às aplicações em Cloudlets discutidas anteriormente em virtualização.

Os autores também fornecem uma comparação de uma tarefa de edição de imagem feita em um dispositivo autônomo e quando ela usa surrogates. Na imagem 3.5, as partes claras representa a tarefa executada pela CPU, e as partes mais escuras o overhead na transmissão pela rede. Podemos ver que usando um celular sem um surrogate, demora muito tempo para ser executada, e ao usar surrogates a diferença é significativa. Os surrogates são muito mais poderosos e executam a tarefa com muito mais rapidez e eficiência. Existe também a possibilidade de melhoramento no overhead da rede, reduzindo-o.

Os resultados do artigo mostram que quando um dispositivo tenta executar uma tarefa relacionada a edição de imagem e sem o uso de surrogates, a utilização da CPU chega a 100%, e com surrogates a tarefa é realizada em muito menos tempo e é muito mais eficiente. Boa parte do atraso é causada pelo overhead da rede, não pela CPU, e se há a oportunidade de melhorar a performance do sistema usando melhores conexões e melhores protocolos, que podem vir com alguns tradeoffs.

Scavenger foi projetado para alcançar eficiência computacional por delegar tarefas e telas executadas em dispositivos (surrogates) diferentes e mais potentes, e as combinando e enviado o resultado de volta ao cliente. Então, a tarefa é paralelizada pela biblioteca Scavenger no lado do cliente e a ideia é achar uma maneira de gerir e agendar essas tarefas para serem executadas em múltiplos nós, dispositivos ou circuitos. Reduz o consumo de bateria,

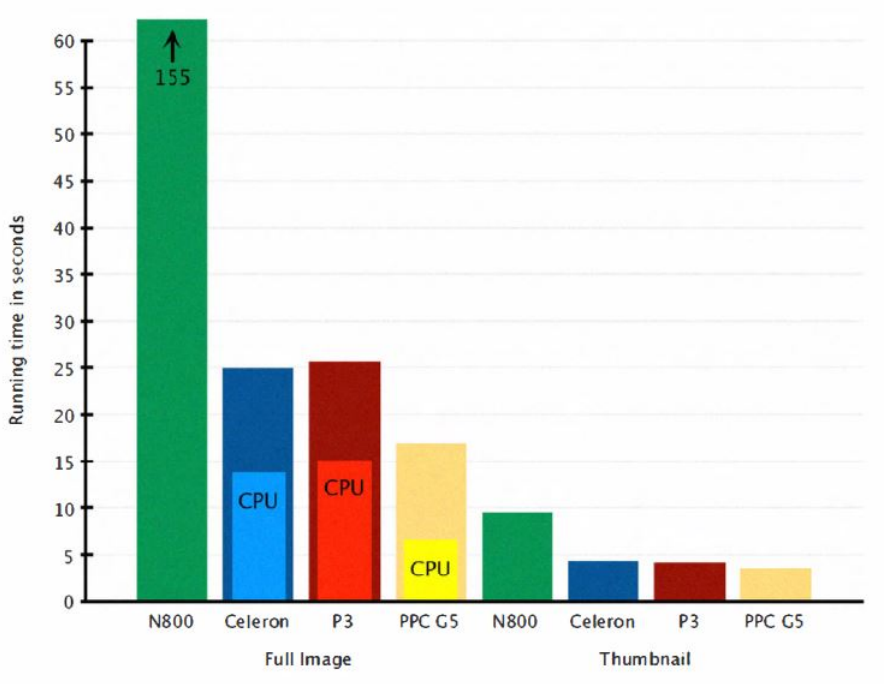

Figura 3.5: Benchmark com operações realizadas com Scavenger. (KRISTENSEN, 2010)

pois afinal a tarefa não é executada no próprio dispositivo. Algo também tratado pelo artigo é o desenvolvimento de um modelo, que permite aos desenvolvedores especificar qual parte do código devem ser executadas no lado do cliente. Se o desenvolvedor sabe o quão complicado é uma tarefa, ele está em uma posição muito melhor de delegar essas tarefas dentro do próprio código, por meio de bons comentários e técnicas padronizadas para delegar essas tarefas para dispositivos próximos (KRISTENSEN, 2010).

Isso completa uma visão geral de técnicas que estão disponíveis, as possibilidades e oportunidades se tratando de pooling de recursos e caching na borda da rede. A motivação para isso é o rápido aumento no uso de serviços famintos por dados, que nos força a pensar nesse novo paradigma, de usar alguns recursos inexplorados e ociosos na borda da rede, para servir como uma alternativa ou complemento às tecnologias em nuvem.

3.4 SÍNTESE

A ideia de Fog Computing é que os usuários finais ou dispositivos na borda da rede realizem tarefas que são majoritariamente funções da nuvem. Tarefas de controle e configuração da rede, armazenamento, processamento e gerenciamento podem ser feitas pelos clientes finais, e isso ajuda a desafogar o tráfego intenso que é estimado para os próximos anos, devido a chegada da Internet das Coisas.

Para o controle de congestionamento da rede já temos o protocolo TCP, que é um controle baseado no usuário final e que possui princípios que estão alinhados com Fog Computing,

ou seja, que o cliente final seja responsável por tarefas de controle. Podemos usar alguns princípios básicos do TCP para realizarmos controle da rede feitos na borda da rede. Isso inclui princípios como:

- O controle da velocidade de transmissão, feito pelo cliente, e não pelo núcleo da rede. Por meio de um feedback, como pacotes de recebimento ACK, é possível inferir o estado da rede e tomar medidas adequadas;
- Uso do protocolo de janelas deslizantes, para que o cliente incremente gradativamente sua velocidade de transmissão ao receber pacotes de confirmação (dentro de um prazo estipulado) e controle sua taxa de transmissão;
- Aumentar a taxa de transmissão de maneira aditiva, e reduzir de maneira multiplicativa. A ideia é que os usuários aumentem gradativamente a velocidade de transmissão e se penalizem rapidamente caso seja percebido que a rede está ficando congestionada;
- Inferir o estado da rede por meio de feedback negativo, como atraso ou perda de pacotes.
- Estimar a perda de pacotes e o atraso por meio de temporizadores.

Colocando tudo isso junto, se tem inferência controle da rede feitas pelos usuários finais, na borda da rede.

Além disso, clientes finais podem ter diversas opções de rede para se conectarem. Os dispositivos atuais permitem que eles se conectem, por exemplo, à redes 3G, 4G, WiFi, tudo no mesmo dispositivo. Uma tarefa de controle é escolher qual dessas redes o usuário deve utilizar. Isso também pode ser feito pelo usuário final, pelo chamado controle de HetNets. Deixar esse controle para ser feito pelo núcleo da rede pode trazer algumas desvantagens, como por exemplo empresas concorrentes não terem o incentivo para cooperarem umas com as outras. Deixar isso para o cliente final é menos custoso, pois ele apenas estará seguindo suas próprias escolhas. Isso não necessariamente significa que será uma pessoa escolhendo a cada momento qual rede utilizar, pode haver um dispositivo que rode um algoritmo para seguir a melhor opção. Foi mostrado um algoritmo simples na seção 3.1.2.1 que demonstra que o usuário final pode fazer essa escolha e que ela pode ser eficiente.

Clientes finais em Fog Computing também podem ser responsáveis pela medição do estado da rede. Isso pode ser interessante, pois esperar resultados vindo da nuvem podem ter latência não compatível com algumas aplicações. Às vezes é melhor realizar uma medição imperfeita, mas com menos latência, em Fog, do que uma medição mais precisa, mas que demore muito para chegar ao dispositivo final. As medições podem ser diretas ou indiretas, e realizadas por um ou mais dispositivos, que podem trabalhar em conjunto, para um resultado mais preciso e eficiente.

Os itens que são interessantes de serem medidos são o throughput, a carga da rede, por meio de inferência do uso do espectro, e quais aplicações estão usando a rede. Medidas como RSSI, SNR, RSRQ e RSRP são utilizadas para inferir o throughput, e essa inferência pode ser utilizada para adaptação de taxa de transmissão. A adaptação de taxa de transmissão pode ocorrer por meio de medições de SNR, que indicam se a transmissão poderia ser mais rápida ou deveria ter sido mais devagar, baseado no valor da razão SNR, ou por meio de indicadores de perda de pacote, onde verifica-se a quantidade de pacotes consecutivos que foram entregues ou não, para aumentar ou reduzir a taxa.

Inferência sobre a carga da rede são feitas através de Medição Passiva, Probing Ativo, Correlação de performance da aplicação e Data Mining. A medição passiva consiste em apenas observar o RTT ou as razões SNR ou RSRQ. Probing ativo é quando se utiliza uma carga de prova e se mede o que acontece com essa carga. Correlação de performance é medir o que acontece no nível de aplicação, por exemplo, verificando a latência do buffer no carregamento de um vídeo. E o Data Mining utiliza um histórico sobre o que acontece na rede em determinado horário e local para aprender determinados padrões sobre a rede, usando uma abordagem de aprendizagem de máquina.

Também pode-se medir a carga da rede por meio de inferência de uso de espectro, onde técnicas utilizam ativamente um sensor para medir a disponibilidade de espectro.

Inferências no nível de aplicação podem usar DPI ou análise de porta. DPI usa um método para ler os pacotes e descobrir qual aplicação está enviando tais pacotes. O uso da inspeção DPI tem caído, pelo fato de poder causar violação de privacidade e pelo aumento do uso de criptografia nas comunicações, que inviabilizam seu uso. A análise de porta é uma alternativa mais simples e leve, porém pode ser ineficaz, visto que números de porta das aplicações podem ser trocados por diversos motivos. Por isso, usar DPI ou análise de porta pode ser mais útil no lado dos clientes ou nos clientes, do que na nuvem.

Pooling e caching na borda da rede pode reduzir a latência, evitar que todos os dados trafeguem para a nuvem, prevenindo gargalos, e promover offloading computacional. Pooling de recursos na borda pode ser feito através de virtualização ou cyber foraging. Dispositivos podem pedir uma ajuda a outros dispositivos próximos para realizar tarefas computacionais, a fim de ter mais poder de processamento ou mesmo utilizar uma maior largura de banda.

A virtualização pode ser por meio de Cloudlets, que são pequenas nuvens. Funcionam como se fossem um pequeno data center, mas menos potente que a nuvem e servindo um menor número de usuários ao mesmo tempo. Pode servir usuários da borda com uma latência muito menor e com largura de banda de uma rede local, sofrendo menos atrasos e evitando o congestionamento da rede. Os usuários de Cloudlets podem requisitar arquivos em cachê que estão armazenados nelas, ou mesmo descarregar parte de um processamento para elas. Isto é muito interessante, pois dispositivos móveis, apesar de serem muito mais potentes do que anos atrás, ainda podem ter pouco desempenho em tarefas computacionalmente exigentes,

como tratamento de imagens.

Cyber Foraging expande a ideia de offloading computacional para diversos dispositivos, onde tarefas podem ser particionadas e serem delegadas paralelamente a dispositivos vizinhos, chamados de surrogates. Pode-se usar recursos inexplorados ou ociosos que estão na borda da rede para servirem como ajuda a outros dispositivos. O uso de cyber foraging pode diminuir bastante o tempo de processamento de algumas tarefas, podendo melhorar a performance de algumas aplicações na borda.

Tais abordagem podem servir como alternativa ou complemento às tecnologias usadas em computação em nuvem, reduzindo o tráfego para o núcleo da rede ao usar recursos que estão mais perto dos clientes finais.

4 SEGURANÇA E PRIVACIDADE EM FOG COMPUTING E INTERNET DAS COISAS

Com o uso cada vez maior de aparelhos conectados à internet e a projeção de enorme crescimento de dispositivos e sensores que vão estar interconectados, é preciso atenção sobre como esses dispositivos vão interagir uns com os outros e com humanos e como se pode ter um ambiente ciberfísico seguro e com privacidade.

Os desafios com a Internet das Coisas são maiores, especialmente por não temos apenas o mundo virtual (cibernético) como na internet comum, mas também temos os sistemas ciberfísicos, com Internet das Coisas e Fog Computing. Os atuadores virtuais podem alterar diretamente uma entidade física, o que é muito próximo de ter um ser humano envolvido na ação: pode ser o computador de bordo de carro realizando uma manobra, aparelhos médicos injetando medicamentos em pessoas, um dispositivo industrial que esteja próximo a trabalhadores. Essa capacidade de atuar no mundo físico é uma das principais vantagens da Internet das Coisas, mas ao mesmo tempo apresenta uma maior preocupação com a segurança. Muitas dessas coisas não podem ser atualizadas com certa frequência, diferentemente de um computador pessoal com um sistema operacional, onde geralmente é possível baixar atualizações e patches de segurança quando brechas e falhas são encontradas. Essas atualizações fornecem soluções ou soluções parciais para alguns problemas. Mas na Internet das Coisas muitas dessas coisas são fabricadas e mantidas por organizações que não são necessariamente grandes empresas de tecnologia, e elas podem não ter a capacidade ou o incentivo para fornecer atualizações. Outras "coisas"podem ser difíceis para atualizar e corrigir. Sendo assim, agora ao invés de assumirmos que a maior parte dos dispositivos disponíveis podem ser seguros, sendo atualizados com alguma frequência , temos que assumir que intrinsecamente grande parte deles não são e não serão seguros.

Mesmo com esse temor de diversos dispositivos inseguros sendo usados aos bilhões, ao mesmo tempo surge uma oportunidade: com Fog Computing e sua arquitetura para prover serviços na borda da rede, perto do cliente, é possível ter mais confiança sobre a privacidade e a segurança da comunicação, uma vez que os dados não precisam percorrer um longo caminho entre o emissor e o destinatário. Os dados não precisam correr um país inteiro atravessando diversas redes e dispositivos. Quanto maior o caminho e mais elementos de redes envolvidos, maiores são as chances de a comunicação ser escutada, comprometida e alterada por entidades não autorizadas. Mantendo a comunicação localmente, maior a chance de termos privacidade e segurança.

O fato de que estes dispositivos serão fabricados e mantidos por empresas que podem não ser de alta tecnologia pode aumentar ainda mais o número de dispositivos com segurança fraca. Geladeiras inteligentes, que são exemplos de “coisas” na Internet das Coisas, já foram hackeadas para enviarem spam, mas danos causados a indústrias e a consumidores podem ser

bem menos inocentes, podendo trazer perda de negócios ou danos físicos (FINLEY, 2015). Isso pode ficar cada vez mais preocupante e perigoso em um mundo ciberfísico.

Este capítulo trata de várias questões sobre segurança e privacidade, mostrando quais são as principais preocupações no mundo da Internet das Coisas e como esses problemas podem ser minimizados.

4.1 SEGURANÇA NOS SERVIÇOS DE REDE

Estudo da Hewlett-Packard afirma que 70% dos dispositivos IoT não criptografam a comunicação, sendo isso ao mesmo tempo uma brecha de segurança e de privacidade (SMITH; MIESSLER, 2014). Isto é um perigo não apenas na internet, mas também em redes locais, onde se imagina estar em um ambiente mais seguro. Outro estudo da HP para segurança residencial afirma que 50% dos dispositivos estudados exibem uma fraca configuração ou implementação de SSL/TLS, o que é bastante grave, uma vez que o principal foco desses sistemas é a segurança (HP, 2015). Dados e credenciais são enviados em claro e atualizações de firmware para os dispositivos não são criptografadas, possibilitando que qualquer um com acesso à rede possa escutar a comunicação e tenham a possibilidade de alterar dados. Uma pessoa que esteja dentro do raio de recepção da transmissão via rádio, passando ao lado de um edifício ou casa, pode espionar a comunicação entre dispositivos. Isto não é facilmente feito na nuvem. Mas com a Internet das Coisas e os diversos dispositivos conectados sem fio, agora basta uma pessoa maliciosa estar perto fisicamente de algum aparelho que envie dados sem fio que haverá chance de violação de privacidade. Não usar sockets seguros, como SSL/TSL, pode exacerbar o problema.

Criptografia no transporte de dados é crítico para todas as comunicações que trafegam pela internet, para que proteja não apenas dados sensíveis como credenciais, informações pessoais, informações bancárias, configurações de segurança, mas também qualquer tipo de informação que não deva ser vazada.

Não apenas deve-se pensar na segurança na camada de transporte, usando SSL/TLS. Outras camadas da arquitetura de redes de computadores também têm um papel importante. Segurança usando criptografia pode ser utilizada também nas camadas de rede e de aplicação. Todas as camadas possuem desafios para proverem uma comunicação confiável na Internet das Coisas.

4.1.1 Segurança na camada de Aplicação

Prover segurança na camada de rede usando IPSec ou na camada de transporte usando TLS ou DTLS tem muitas vantagens. As principais são: primeiro, o mesmo mecanismo padrão e a mesma implementação podem ser compartilhadas por todas as aplicações, resul-

tando em reuso de código e redução do tamanho do código;Segundo, programadores não precisam se preocupar com a implementação de nenhum mecanismo de segurança. Isso simplifica significantemente o desenvolvimento de aplicativos, também em presença de comunicações seguras. Infelizmente, IPSec e (D)TLS possuem seus próprios inconvenientes. O principal problema que é comum a ambas abordagens é a impossibilidade de assegurar uma segurança de fim a fim completa, quando aplicativos de comunicações são retransmitidos por nós intermediários que funcionam na camada de aplicação (por exemplo, no uso de proxy). Nesse caso, a segurança de usuário final até usuário final ainda pode ser provida pelas camadas de transporte ou de rede, mas somente na presença de sistemas intermediários bem confiáveis. Mas segurança geral é complicada pelo manuseio de tal gerenciamento de confiança nó-a-nó.

Uma abordagem diferente para prover uma completa segurança fim a fim é forçar segurança diretamente na camada de aplicação. Isso simplifica os requerimentos das camadas abaixo, e provavelmente reduz custos.

A maior desvantagem de prover segurança no nível de aplicação são as complicações introduzidas pelo desenvolvimento da aplicação e o tamanho geral do código causado por um ruim reuso de códigos de software. Isso é principalmente causado pela falta de protocolos de segurança bem definidos e adotados na camada de aplicação. Exemplo de padrões que podem ser usados para esse propósito são S/MIME e SRTP.

S/MIME é um padrão para prover autenticação, integridade, não repúdio da origem e confidencialidade para os dados da aplicação. Não foi originalmente criado para esse fim, mas pode ser usado para a segurança de qualquer dado de aplicação e encapsulado em qualquer protocolo de aplicação ou transporte. SRTP é outro protocolo que provê confidencialidade e autenticação de mensagem. É uma extensão de RTP especificamente desenvolvido para tratar de comunicação de dados em tempo real (vídeos ou comunicações por voz), mas pode ser reusado em outros cenários de aplicação. Funciona de uma maneira “por pacote” e é usualmente encapsulado em UDP. Mas ainda é necessário mais investigação para saber qual protocolo é mais indicado para proteger os dados na camada de aplicação em uma rede com dispositivos limitados como em IoT. (CIRANI; FERRARI; VELTRI, 2013)

Ataques em camadas mais baixam focam nos canais de comunicação, na banda de transmissão, em computadores e roteadores. Ataques de negação de serviço (DoS) na camada de aplicação, entretanto, tem intenção de esgotar recursos de um computador, como CPU. Como mostrado em X, ataques na camada de aplicação podem facilmente esgotar um computador com recursos computacionais limitados por fazer inúmeras requisições computacionais custosas. Com milhares de dispositivos da Internet das Coisas equipados com componentes com recursos limitados, eles podem ser vitimas potenciais ataques DoS na camada de aplicação. (WANG; LU, 2013)

4.1.2 Segurança na camada de Transporte

A OWASP trata a falta de encriptação como um dos Top 10 desafios de segurança da Internet das Coisas (OWASP, 2014). Falta de encriptação na camada de transporte permite que dados sejam vistos ao trafegar redes locais ou a internet. Falta de encriptação é mais comum em redes locais, já que é fácil de assumir que tráfego de redes locais não serão largamente visíveis, entretando no caso de uma rede local sem fio, configurações erradas da rede sem fio podem fazer o tráfego ser visível para qualquer um dentro do raio da rede. Diversos problemas com encriptação são facilmente descobertos por simplesmente olhar o tráfego de rede e procurar por dados legíveis. Ferramentas automatizadas também ajudam a procurar por implementações corretas de encriptação como SSL e TLS. Checagem de falta de encriptação na camada de transporte inclui (OWASP, 2014):

- Revisar o tráfego de rede do dispositivo, seu aplicativo móvel e qualquer conexão com a nuvem para determinar se qualquer informação é passada em texto em claro;
- Revisar o uso de SSL ou TLS para garantir que está atualizado e bem implementado;
- Revisar o uso de qualquer protocolo de encriptação para garantir que eles são recomendados e aceitados.
- Encriptação suficiente na camada de transporte requer:
- Garantir que dados são encriptados usando protocolos como SSL e TLS quando trafegando redes;
- Garantir que outros protocolos padronizados são usados caso SSL ou TLS não estiverem disponíveis;
- Garantir que apenas protocolocos de encriptação padronizados são utilizados, evitando uso de protocolos de encriptação proprietários.

A troca de dados entre de aplicações na atual arquitetura IP pode ser protegida, na camada de transporte, utilizando os protocolos TLS e DTLS. O TLS é o protocolo de segurança mais utilizado, baseando-se no protocolo de transporte TCP e provendo, para a camada de aplicação, a mesma comunicação orientada ao fluxo (DIERKS; RESCORLA, 2008). Já o DTLS foi introduzido mais recentemente para prover um serviço de segurança similar ao TLS, mas rodando com base no protocolo UDP. DTLS é um protocolo de segurança de referência para sistemas IoT, uma vez que utiliza UDP e não sofre de problemas gerados pelo uso de TCP em casos onde os recursos de rede são limitados (MODADUGU; RESCORLA, 2012).

A principal vantagem de proteger as comunicações na camada de transporte usando DTLS consistem em permitir um controle de acesso mais preciso. Operações na camada de

transporte permitem que aplicações selecionem facilmente qual serviço de segurança deve ser iniciado. Outra vantagem prática é que a adoção de DTLS pode permitir o reuso da larga experiência e de implementações que vieram com o TLS. Por essas razões,DTLS recentemente recebeu uma atenção significante para proteger a comunicação de aplicações de rede/nós limitados (SHELBY; HARTKE; BORMANN, 2014).

Ambos IPSec e DTLS proveem as mesmas características de segurança com seus próprios mecanismos em diferentes camadas. Mas ainda restam alguns problemas que devem ser enfrentando para fazer o DTLS mais amigável para dispositivos limitados. O mais relevante deles é relacionado ao tamanho limitado do pacote, imposto por protocolos de base como o IEEE 802.15.4. De fato, como para IPSec, DTLS introduz overhead durante ambas as fases de negociação (handshake) e transporte de dados. DTLS oferece fragmentação na camada de handshake, entretanto, isso pode adicionar um overhead significante. Uma solução que pode ser usada é usar a fragmentação oferecida por IPv6 ou 6LoWPAN.

Um problema de usar DTLS ou IPSec é que comunicações seguras fim a fim não são garantidas quando nós intermediários como proxies ou gateways no nível de aplicação são introduzidos. IPSec e DTLS proveem comunicações seguras na camada de rede e transporte, respectivamente, e, em presença de uma comunicação com vários nós (hop)no nível de aplicação, eles podem garantir segurança somente em cada nó. Adicionalmente, algumas complicações ao prover segurança fim a fim podem surgir mesmo quando a conectividade é realizada diretamente nas camadas de rede e transporte. Existem cenários que uma parte da rede (interna) composta por dispositivos limitados é interconectada na camada de rede para o resto (externo) da rede, que é o caso da internet. Mesmo que a proteção dos dados possa ser garantida pelos protocolos IPSec ou DTLS, outros ataques a rede, como flooding, podem ocorrer por causa da assimetria dos recursos disponíveis nos sistemas finais. Por exemplo, um cliente final robusto ligado à Internet pode atacar um dispositivo limitado por tentar consumir todos os recursos do limitado poder de processamento do dispositivo . A fim de garantir um nível adequado de proteção , também contra este tipo de ataques , um gateway intermediário de segurança pode ser exigido na borda da rede interna . Um gateway de segurança pode agir como controlador de acesso , provendo acesso à rede interna só para nós confiáveis. (CIRANI; FERRARI; VELTRI, 2013)

o caso de comunicação fim-a-fim no nível de aplicação baseado em CoAP , uma solução pode ser a de exigir que o nó externo encapsule CoAP, DTLS ou tráfego IP dentro de um túnel DTLS estabelecido entre o nó externo e o gateway de segurança (BRACHMANN et al., 2012).

4.1.3 Segurança na camada de Rede

IPSec é um conjunto de protocolos para proteger a Internet , autenticando e criptografando cada pacote IP de uma sessão de comunicação (SEO; KENT, 2005). Esses serviços

de segurança são implementados por dois protocolos de segurança do IPSec: Authentication Header (AH), que provê integridade e autenticação, e Encapsulated Security Payload (ESP), que pode prover confidencialidade, autenticação e integridade. IPSec inclui a negociação de chaves criptográficas usadas para a criptografia. Pode ser usando, na camada de rede, para proteger fluxo de dados entre um par de clientes finais, entre um par de gateways de segurança ou entre um gateway de segurança e um cliente final . Ele protege o diálogo mesmo quando a aplicação não incluir segurança. Na falta de IPsec, TLS/SSL ou DTLS devem ser incluídas na aplicação para proteger as comunicações.

IPSec pode ser usado na presença de dispositivos com recursos limitados, se é selecionado um algoritmo apropriado que garanta tanto usabilidade e níveis de segurança suficientes. Isso significa que, a partir de um ponto de vista algorítmico, o problema se move a partir do próprio protocolo IPSec para a real algoritmos criptográficos. As chaves e os algoritmos de criptografia selecionados usado por IPSec para garantir um comunicação são chamados IPSec Security Association (SA). Para estabelecer uma SA, o IPSec pode ser pré-configurado (especificando um algoritmo de chave pré-compartilhada, função hash e criptografia) ou pode ser dinamicamente negociado pelo protocolo IPSec Internet Key Exchange (IKE). Infelizmente, como o Protocolo IKE foi projetado para nós de Internet padrão, ele usa criptografia assimétrica, que é pesados computacionalmente para dispositivos muito limitados, que são bastante usados na Internet das Coisas. Por esta razão, extensões adequadas do IKE devem ser consideradas fazendo uso de algoritmos mais leves. Outros problemas relacionados com a implementação de IPSec nos nós limitados da Internet das coisas incluem overhead de dados, configuração e aspectos práticos de implementação. Sobrecarga (overhead) de dados é introduzido pela o encapsulamento cabeçalho extra de IPSec AH e/ou ESP. No entanto, isso pode ser limitada pela aplicação técnicas de compressão no cabeçalho, à semelhança do que é feito em 6LoWPAN para o cabeçalho IP.

Em relação aos aspectos práticos , é necessário observar que o IPSec é frequentemente projetado para VPNs , assim, o que torna difícil para eles para ser dinamicamente configurável por um aplicativo. Além disso , implementações existentes também são dificilmente compatíveis entre si e muitas vezes exigem a configuração manual para interoperarem. (CIRANI; FERRARI; VELTRI, 2013)

Uma alternativa ao uso de IKE e IPsec é o Host Identity Protocol (HIP) (MOSKOWITZ et al., 2015). O principal objetivo do HIP é de dissociar as duas funções de localizadores de clientes finais (para fins de roteamento) e identificadores de clientes, atualmente desempenhadas por endereços IP. HIP introduz um novo namespace entre IP e camadas superiores especificamente para a identificação de host com base em criptografia pública. (CIRANI; FERRARI; VELTRI, 2013)

4.2 CONFIGURABILIDADE DE SEGURANÇA

Passando para um mundo onde teremos tantos dispositivos, a quantidade de senhas que devem ser lembradas passa a ser enorme. Imaginando que se tem uma senha para cada dispositivo ou uma conta individual em cada dispositivo, seria comum esquecer senhas quando se tem dezenas delas. Não havendo uma senha diferente para cada dispositivo, então teremos que repetir senhas, teremos que confiar em outro dispositivo ou entidade para efetuar a autenticação, ou teremos que ignorar alguma fase de autenticação, e nenhuma é uma solução perfeita. Senhas fracas são um dos problemas da internet atual, e pode continuar sendo um problema na Internet das Coisas. De acordo com um relatório da HP, cerca de 80% dos dispositivos testados não exigem senha de complexidade e comprimento suficientes (SMITH; MIESSLER, 2014). Com a Internet das Coisas, onde pessoas vão possuir muito mais dispositivos, ter uma senha diferente para cada dispositivo não é algo viável, pois é difícil lembrar todas as diferentes senhas então há um tradeoff entre segurança e usabilidade.

Outro estudo revelou que 13% dos mais de meio milhão de dispositivos embarcados têm as senhas de administrador configuradas com senhas padrão de fábrica, onde é fácil descobrir quais são essas senhas padrão. Utilizar senhas que vem de fábrica é preocupante pois uma rápida pesquisa na internet pode mostrar quais são essas senhas. (CUI; STOLFO, 2010)

Configurabilidade de segurança não se trata apenas de senhas fortes. Se trata de permitir que um usuário possa tomar diversas atitudes com relação à segurança do dispositivo. Temos uma configurabilidade de segurança insuficiente quando um usuário tem pouca ou nenhuma habilidade de alterar as configurações segurança do dispositivo. Por exemplo, uma interface web que não permite criar diferentes contas com diferentes níveis de acesso (acesso granular), não obrigue o uso de senhas fortes (não deixa que o usuário registre senhas pequenas e fáceis como uma sequência de números) ou que não tenha encriptação suficiente (protocolos de segurança são mal implementados ou inexistentes). Usuários maliciosos aproveitar a falta de controle de contas para ter acesso a dados e controle do dispositivo, ou também se aproveitar da falta de encriptação ou de senhas fortes para acessar o dispositivo e comprometer dados. De acordo com a OWASP, para analisar se a configurabilidade de segurança é deve-se, na interface administrativa do dispositivo (OWASP, 2014):

- Procurar se há opções para melhorar a segurança, como forçar a criação de senhas fortes;
- Checar se há opções para habilitar um acesso granular ao dispositivo, separando usuários comuns de administradores;
- Checar se existem opções para selecionar diferentes métodos de encriptação;
- Buscar por opções para ativar registros (logs) de eventos de segurança;

- Procurar por opções para ativar alertas e notificações caso algum evento de segurança ocorra.

Fabricantes e desenvolvedores de dispositivos IoT devem possibilitar que um usuário tenha diversas opções para configurar a segurança do dispositivo, implementando as opções citadas acima. Usuários devem buscar e fazer uso dessas opções para reduzir os riscos de comprometimento do dispositivo e de seus dados.

4.3 AUTENTICAÇÃO E AUTORIZAÇÃO

Autenticação e autorização são palavras que são vistas continuamente, mas que possuem mais de um significado. Uma delas é a gestão de confiança, que se trata de especificar quem detém qual nível de permissão (ROMAN; ZHOU; LOPEZ, 2013). Por exemplo, casas, edifícios e cidades inteligentes são interligados, o que permite que sejam controlados remotamente por alguém que tenha autorização. Quem pode controlar a fechadura da porta, a TV ou a iluminação da casa, ou seja, quem dentro de uma família e seus convidados pode ter esse controle e diferentes níveis de permissão.

Outro significado é o gerenciamento de identidade. Aqui se trata de garantir que uma pessoa ou dispositivo seja realmente quem ele diz que é. É essencial o desenvolvimento de mecanismos de gerenciamento de identidade para criar serviços confiáveis, provendo confiabilidade entre múltiplas entidades. O ambiente na Internet das Coisas é bastante dinâmico, uma entidade pode não saber previamente com qual outra entidade irá interagir. Dentro de um mundo com bilhões de dispositivos, é necessário gerenciar as identidades de um modo que seja facilmente escalável, de modo a permitir rápida identificação das entidades. É importante lembrar que os dispositivos da Internet das Coisas podem modificar entidades físicas, e o impacto de um dispositivo malicioso falsificar uma identidade é ainda pior do que antes, já que ele pode enviar comandos maliciosos para efetuar ações físicas, podendo ferir alguém. (KANUPARTHI; KARRI; ADDEPALLI, 2013)

Os dispositivos da Internet das Coisas são heterogêneos e é um desafio chave resolver essa questão de como autenticar e autorizar uma grande variedade de dispositivos, que podem ter características muito distintas. Não apenas é necessário criar algoritmos criptográficos eficientes para dispositivos limitados em processamento, bateria e/ou largura de banda, também são necessários protocolos de segurança leves para prover um canal de segurança fim-a-fim seguro.

A OWASP classifica autenticação e autorização insuficientes um dos Top 10 problemas de segurança em sistemas IoT. Falhas na autenticação estão presentes quando senhas fracas são usadas ou fracamente protegidas, utilização de métodos de recuperação de senha mal implementados, falta de acesso granular, entre outros. Estas falhas permitem que um usuá-

rios mal intencionados invada dispositivos e possivelmente roube ou corrompa dados. Falhas desse tipo podem ser encontradas por meio de uma inspeção manual ou usando ferramentas automatizadas. Uma inspeção deve procurar (OWASP, 2014):

- Determinar os requisitos para criação de senhas, se são exigidas senhas complexas, se há histórico de senhas e se as senhas expiram. Senhas seguras dificultam o acesso não autorizado;
- Revisar o tráfego de rede para saber se credenciais são transmitidas sem criptografia (texto em claro). Qualquer pessoa dentro da área de alcance de uma transmissão sem fio poderá ver as credenciais caso não sejam criptografadas;
- Se uma nova autenticação é necessária para alguns controles mais finos, para evitar atacantes que de algum modo conseguiram uma sessão como administrador, mas não possuem a senha.
- Uma revisão de autorização: verificar se a interface permite separação de funções em usuários e administradores (acesso granular) e se esse controle de acesso funciona.

  Ou seja, para melhorar os métodos de autenticação e autorização nos dispositivos deve-se exigir senhas fortes, permitir acesso granular, garantir que as credenciais sejam devidamente protegidas por meio de criptografia, garantir que mecanismos de recuperação de senha são seguros, requisitar nova autenticação para ações sensíveis e, se possível, implementar autenticação em duas fases.

4.4 SEGURANÇA EM INTERFACES

Existem muitas interfaces diferentes que são utilizadas em diferentes dispositivos, e muitos deles são vulneráveis a diversos ataques. Ter uma rede doméstica ou de pequenos escritórios atrás de um firewall não é uma garantia que vulnerabilidades não possam ser exploradas. A OWASP classifica Interface web e Cloud interface como um dos Top 10 problemas de segurança da Internet das Coisas (OWASP, 2014) .

4.4.1 Segurança na interface Web

Um estudo revelou que 60% dos dispositivos testados apresentaram alguma falha em suas interfaces web, como vulnerabilidade a ataques XSS (Cross-Site Scripting) ou XCS (Cross-Channel Scripting), gerenciamento de sessão inadequado e credenciais de fábrica fracas (SMITH; MIESSLER, 2014). Ataques XSS ou XCS permitem que um invasor execute remotamente um script de ataque em um dispositivo por meio de uma interface web. Interfaces web inseguras também apresentam questões como possibilidade de enumeração

de contas (identificar contas válidas) e falta de bloqueio de conta quando um número de tentativas de acesso é atingido. Interfaces Web inseguras ocorrem predominantemente em redes internas, mas riscos vindos de um usuário interno podem ser tão perigosos quanto o de um usuário externo, levando a perda ou corrupção de dados até a completa perda do controle sobre o dispositivo.

Existem diversas medidas para minimizar as falhas com interfaces Web inseguras. Testes manuais ou com aplicativos ajudam a identificar potenciais vulnerabilidades. Aplicações como ZAP e DAST são ferramentas que ajudam a procurar por potenciais vulnerabilidades em interfaces web, quem fazem uma varredura por portas que o dispositivo está escutando (e que podem potencialmente serem usadas por usuários maliciosos). Para minimizar os riscos de uma interface web insegura deve-se (OWASP, 2014):

- Assegurar que contas e senhas de fábrica sejam trocadas no primeiro uso, e com senhas fortes;
- Bloquear acesso após seguidas tentativas de login, impedindo novas tentativas de acesso durante um determinado tempo ou até que alguma ação seja completada;
- Evitar que contas válidas possam ser identificadas em páginas de cadastro ou usando métodos de recuperação de senha;
- Revisar a interface a procura de possíveis campos onde poderiam ser efetuados ataques como SQL Injection e Cross-Site Scripting.
- Certificar que algum método de criptografia é utilizado, evitando vazamento de credenciais e dados na rede interna ou externa.

Qualquer pessoa ao adquirir um dispositivo conectado a web e que possua alguma interface de gerenciamento deve estar atento a essas questões, para que informações sensíveis não sejam expostas e que os riscos de impactos em negócios ou questões pessoais sejam minimizados.

4.4.2 Segurança na interface de acesso à Nuvem

A segurança é outra preocupação na hora de definir a Cloud Interface, ou interface de acesso à nuvem. Autenticação insuficiente, credenciais fáceis de serem adivinhadas, falta de criptografia, identificação de contas válidas por meio de requisições de recuperação de senha: todas são potenciais brechas. Estudo da HP sobre dispositivos de segurança doméstica mostrou que 70% dos dispositivos testados apresentavam falhas que permitiam que usuários tivessem tentativas ilimitadas para adivinhar contas válidas.

Para uma Cloud interface segura, é preciso (OWASP, 2014):

- Usuários e senhas sejam atualizados no primeiro uso;
- Garantir que não se possa identificar contas válidas usando, por exemplo, métodos de recuperação de senha;
- Bloquear tentativas de acesso após falhas sucessivas; garantir que a interface não seja suscetível a SQLi, XSS ou CSRF;
- Garantir que credenciais não sejam expostas na internet (usar criptografia);
- Se possível, Implementar autenticação em duas fases.

Podemos usar a rede Fog para deixar a nuvem mais segura. Por exemplo, antes de enviar algo para a nuvem, pode-se dividir arquivos em pequenos pedaços e espalhar por diversos provedores de nuvem diferentes, de modo que nenhum deles possa recuperar completamente o arquivo. Dessa maneira estamos dissociando o armazenamento maciço que fica na nuvem, de maneira que ela continue na nuvem, mas efetuando um controle de privacidade, que não é a função primária de armazenamento, e colocando na Fog. Portanto, um software em Fog pode executar o retalhamento dos arquivos e em seguida espalhá-lo em vários provedores da nuvem. Isso é uma situação onde a privacidade é melhorada numa interface entre Nuvem e Fog. A nuvem ainda está fazendo o armazenamento, mas a Fog está controlando o que é espalhado e o que pode ser juntado novamente nos dispositivos cliente.

4.5 SEGURANÇA DE SOFTWARES E FIRMWARES

Como dito na introdução do capítulo, dispositivos IoT são intrinsecamente não atualizáveis. As empresas que desenvolvem e vendem esses dispositivos podem não ter o incentivo ou a capacidade para atualizá-los, deixando-os com softwares e firmwares inseguros. Mesmo dispositivos novos são vendidos com software antigo. Uma pesquisa relevou que roteadores domésticos possuem software quatro ou cinco anos mais antigos que os dispositivos (SCHNEIER, 2014). Existem diversas vulnerabilidades típicas de software, como buffer overflow, falhas em código ou máquina de estados. Vulnerabilidades em dispositivos são encontradas diariamente e a não possibilidade de atualização é, por si só, uma grande vulnerabilidade. Mas dispositivos que podem ser atualizados também podem ser afetados na hora de baixar atualizações. Atualizações de segurança podem ser comprometidas se os arquivos ou a conexão de rede não forem protegidas por encriptação.

Relatório da Symantec indica que novos worms em Linux tem como alvo dispositivos antigos sem correções. O pensamento de inovar em novos dispositivos e novos gadgets é crescente, enquanto dispositivos antigos são esquecidos desatualizados. Se eles não estiverem ligados em rede, como antigamente, o problema é menor, mas na hipótese de estarem em rede, é possível que as pessoas não os desliguem ou eles não se desligam automaticamente.

HP relata que diversos sistemas possuem problemas com atualizações de firmware, como transmitir updates sem encriptação e sem encriptar os arquivos de atualização. Em 60% dos dispositivos testados não havia nenhuma opção de atualização automática e em apenas 30% dos dispositivos testados havia uma opção para aceitar ou recusar a última atualização de firmware (HP, 2015). Mesmo que seja possível aplicar atualizações nos dispositivos, raramente isto é feito. O usuário raramente é avisado que existem atualizações disponíveis, e, mesmo que seja, talvez não saiba como aplicar a atualização.

A Internet das Coisas vai deixar as coisas piores: uma imensidão de dispositivos conectado em rede e distribuídos em vários locais, de fábricas a residências e edifícios, que com o tempo vão se tornar antigos e com vulnerabilidades, mas que não serão atualizados. Usuários maliciosos terão esses dispositivos como alvos fáceis, com brechas para capturarem dados sigilosos ou algum outro dano virtual ou físico.

OWASP classifica Software e Firmware Inseguros como um dos Top 10 problemas de segurança na Internet das Coisas. Para termos software e firmware seguros deve-se empenhar que (OWASP, 2014):

- O dispositivo tenha a capacidade de ser atualizado;
- O dispositivo tenha algum mecanismo de atualização automática;
- As atualizações e a transmissão dos arquivos de atualização sejam criptografados;
- O arquivo de atualização não exponha dados sensíveis, como credenciais;
- A atualização seja assinada e verificada antes de ser aplicada no dispositivo;
- O servidor de atualização seja seguro;
- Vendedores de sistemas embarcados façam um design melhor de seus sistemas e de preferência com driver feito com código aberto para serem mais facilmente atualizáveis (SCHNEIER, 2014).

4.6 SEGURANÇA FÍSICA E PROBLEMAS DE SEGURANÇA EM DISPOSITIVOS COM RECURSOS LIMITADOS

Muitos dispositivos na Internet das Coisas são limitados em recursos. Eles podem possuir problemas como limitação de poder de processamento, quantidade de memória ou duração de bateria. Podem não ter qualquer tela de toque ou outra interface, ou ainda ter capacidade de comunicação limitada. Estes dispositivos estão sujeitos a falhas de hardware, a ataques como buffer overflow. Dispositivos sem fio estão sujeitos a jamming e interferências de sinal (WANG; LU, 2013).

Com tais limitações de hardware é um desafio criar sistemas e protocolos eficientes para que os serviços de rede sejam entregues adequadamente. (RASHWAN; TAHA; HASSANEIN, 2014) apresentou novas métricas para avaliar o impacto de medidas de segurança em sistemas móveis.Observaram vários comportamentos de performance não intuitivos que criam uma necessidade de revisitar como funções de segurança são avaliados e selecionados em comunicações móveis, especialmente quando relacionadas a garantias baseadas em tempo. A métrica sugerida mostrou ser mais informativa no contexto de comunicações e razoável candidato a métrica em um design futuro de protocolo.

Por vezes, dispositivos possuem um sistema operacional que pode ser acessado fisicamente e alterado através de diferentes meios, como portas USB, cartões SD ou mesmo ter acesso por meio sem fio. Por terem diversas interfaces, podem apresentar problemas com segurança física, sendo vulnerável a ataques por pessoas que possuem acesso físico ao dispositivo (ROMAN; ZHOU; LOPEZ, 2013).

Atacantes podem usar vetores como portas USB, cartões SD ou outro meio de armazenamento para acessar o sistema operacional e potencialmente qualquer dado armazenado no dispositivo.

Fragilidades na segurança física estão presentes quando um atacante pode desmontar um dispositivo para facilmente acessar mídia de armazenamento e qualquer dado armazenado. Fragilidades também estão presentes quando portas USB ou outra porta externa pode ser usada para acessar o dispositivo usando recursos projetados para configuração ou manutenção. Segurança física insuficiente pode levar ao comprometimento do dispositivo e qualquer dado armazenado nele.

A OWASP classifica a fraca segurança física como um dos Top 10 problemas relacionados a Internet das Coisas. Para determinar a qualidade da segurança física de um dispositivo deve-se rever (OWASP, 2014):

- Quão facilmente um dispositivo pode ser desmontado e a mídia de armazenamento pode ser acessada ou removida;
- Se o uso de portas externas como USB para determinar se dados podem ser acessados sem precisar desmontar o dispositivo;
- O número de portas externas físicas para determinar se todas são necessárias para o bom funcionamento do dispositivo;
- A interface administrativa para determinar se portas externas como USB podem ser desativadas;
- A interface administrativa para determinar se capacidades administrativas podem ser limitadas para acesso local somente.

4.7 PRIVACIDADE NA INTERNET DAS COISAS E EM FOG COMPUTING

Além das questões de segurança que foram mencionadas, existem também as questões de privacidade. Estamos entrando em um momento onde teremos casas inteligentes, com uma série de dispositivos que estão todos conectados. Podemos ter aparelhos monitorando o movimento de nossos olhos, escutando o que falamos e vigiando nossos movimentos, fornecendo e armazenando uma série de dados sobre nosso comportamento. Certamente, a ideia de Internet das Coisas é que ela possui vários sensores. Os sensores em dispositivos da Internet das Coisas dão origem a uma série de preocupações com a privacidade, assim como os atuadores dão origem a muitas preocupações sobre segurança, pela habilidade de mover objetos físicos.

Surge a dúvida de quem deve controlar os dados de comportamento. Talvez deva-se armazená-los temporariamente no interior da Fog usando um dispositivo de armazenamento compartilhado, e em seguida distribuindo-os em múltiplas nuvens diferentes. Foi visto que uma arquitetura distribuída, como Fog Computing para a Internet das Coisas, requer um mecanismo de segurança mais complexo. Porém, em algumas áreas, esta arquitetura distribuída provê benefícios imediatos: gerenciamento de dados e privacidade. A ideia chave é que visto que os dados serão processados na borda da rede, cada entidade detém mais controle sobre os dados que ele gera e processa. Cada entidade pode controlar a granularidade dos dados que ele produz, ou seja, qual vai ser a precisão e qual informação vai ser enviada para uma entidade externa, e também definir para quem ou em qual situação revelará seus dados, definindo sua própria política de acesso. Ter o poder de controlar a informação na borda é ótimo, mas ter diversos dispositivos na borda pode ser um perigo caso alguma entidade vigie usuários sem sua permissão, fazendo uso de vários sensores para gerar perfis de usuários.

É necessário ir além de políticas de acesso centradas no usuário e mecanismos que controlam a granularidade dos dados. Aspectos de usabilidade vem à tona quando pessoas estão envolvidas. Preocupações com a coleta de informações pessoais, informações financeiras, geolocalização ou dados sobre saúde já fazem parte da internet tradicional. Agora com a Internet das Coisas e seus diversos sensores, há a preocupação sobre a coleta contínua de informação, como hábitos, trajetos frequentes e condição física, que se não coletam dados sensíveis diretamente, possibilitam que entidades possam inferir informações através de outros dados coletados. Tamanha quantidade de dados possibilita uma nova forma de análise que antes não era possível. Pesquisadores começam a mostrar que sensores de smartphones conseguem inferir o humor do usuário, níveis de stress, personalidade, padrões de sono, quantidade de exercícios, entre outros. Apesar de haver uma ótima oportunidade de fazer bom uso desses dados, como para diagnosticar doenças, há o risco de os dados serem usados por entidades ou por vias não autorizadas. É possível que tais dados sejam usados em práticas discriminatórias, beneficiando alguns grupos e desfavorecendo outros.

Preocupações sobre privacidade são uma das Top 10 questões sobre Internet das Coi-

sas, de acordo com a organização OWASP. Para minimizar os problemas com privacidade violada, fabricantes de dispositivos e usuários deve-se garantir que (OWASP, 2014):

- Apenas os dados realmente necessários para a funcionalidade do dispositivo sejam coletados;
- Qualquer dado coletado seja desidentificado ou que seja anônimo;
- Todos os dados coletados sejam protegidos por criptografia;
- O dispositivo e seus componentes protejam as informações pessoais;
- Apenas pessoas autorizadas devem ter acesso a informações pessoais;
- Limites sejam impostos para a coleta de dados;
- Que usuários finais sejam avisados caso os dados coletados sejam maiores do que esperado.

4.8 SÍNTESE

Com a chegada da Internet das Coisas, muito mais dispositivos vão estar conectados à internet. Estamos em um momento onde cada vez mais pessoas usam a internet e temos muitos desafios em relação a segurança e privacidade. Com o aumento do número dos dispositivos e com o novo ambiente onde eles podem modificar o ambiente físico, o que é chamado de ambiente ciberfísico, os desafios em relação à segurança e privacidade se tornam ainda maiores. Foram apresentados nesse capítulo alguns desafios de segurança e privacidade em Fog Computing e Internet das Coisas e algumas medidas para atenuar alguns problemas.

Existem diversas soluções para melhorar a segurança dos serviços de rede. Na camada de aplicação existem protocolos como S/MIME e SRTP, que provêm autenticação, integridade, não repúdio da origem e confidencialidade para os dados da aplicação. Na camada de transporte temos os protocolos SSL e TLS para serem usados para encriptar mensagens. Na Internet das Coisas é interessante o uso do protocolo DTLS, que é similar ao TLS e funciona melhor em dispositivos com poucas capacidades computacionais. Na camada de rede há o protocolo de segurança IPSec, provendo serviços de integridade, autenticação e confidencialidade. O problema é que o IPSec, com seu protocolo IKE para troca de chaves criptográficas, é um pouco pesado computacionalmente para dispositivos limitados de IoT e introduz overhead.

Não só problemas relacionados a protocolos ou criptografia fazem parte das preocupações de segurança. Alguns desafios são mais relacionados ao uso ou ao desenvolvimento dos dispositivos. Problemas relacionado a configurabilidade de segurança estão presentes

na Internet das Coisas. Usuários devem fazer uso de opções de segurança no dispositivo e desenvolvedores devem implementar essas medidas em seus dispositivos.

Problemas de autenticação e autorização também fazem parte da Internet das Coisas. Usuários e desenvolvedores devem estar atentos ao uso e criação de senhas e acesso granular a funções administrativas, bem como se preocuparem com a encriptação de credenciais que trafegam pela rede.

Interfaces Web e interfaces de acesso à nuvem podem possuir diversas vulnerabilidades. Assegurar que senhas sejam fortes e que a interface não seja suscetível à ataques como XSS e SQLi são fundamentais, bem como o uso de criptografia.

Dispositivos desatualizados são um grande perigo na Internet das Coisas, e esse problema tende a se atenuar. Softwares e Firmwares desatualizados abrem brechas para que dispositivos sejam facilmente atacados, mas o grande problema é que os dispositivos da Internet das Coisas provavelmente não serão atualizados. Desenvolvedores de sistemas IoT devem se empenhar para proverem atualizações automáticas para seus dispositivos, visto que usuários não possuem costume de atualizá-los.

A enorme quantidade de dispositivos espalhados por toda parte, na Internet das Coisas, abre brechas para que esses dispositivos sejam acessíveis fisicamente, que uma pessoa possa tocá-los e modificá-los. Pessoas maliciosas podem usar as portas de conexão do dispositivo para alterarem suas configurações. Por isso, desenvolvedores devem tomar cuidado com o design de seus dispositivos, não facilitando o acesso aos meios de armazenamento.

Não bastante, dispositivos na Internet das Coisas podem ser muito limitados em seus recursos, o que os torna sujeito a falhas de hardware e a ataques de, por exemplo, buffer overflow. É um grande desafio criar medidas de segurança que sejam computacionalmente leves para esses dispositivos.

5 CONCLUSÕES

Fog Computing é uma arquitetura distribuída que complementa e estende a Computação em Nuvem. Algumas aplicações que estão surgindo não tão compatíveis com o modelo centralizado da nuvem. Diversas previsões sugerem um grande aumento no número de dispositivos e de tráfego que vai circular pela rede, e isso pede uma mudança de paradigma da computação. Ter essa arquitetura distribuída se tornou interessante agora pelo fato dos dispositivos finais estarem cada vez mais potentes e baratos, o que abre a oportunidade de utilizar seus recursos para executarem tarefas que eram majoritariamente funções da nuvem. Mas partir da arquitetura centralizada atual para a arquitetura distribuída em Fog não é algo trivial. Os dispositivos e aplicações da Internet das Coisas possuem requisitos e recursos heterogêneos, o que significa que pode ser difícil criar protocolos e aplicações que sejam interconectáveis e que funcionem bem para diferentes tipos de dispositivos.

Fog Computing usa uma multidão de dispositivos, que podem trabalhar em conjunto, para entregar serviços que são majoritariamente serviços da nuvem, como armazenamento, processamento, comunicação e gestão da rede. Fog Computing traz mudança de paradigma na computação, com aplicações indo até os dados, ao contrário da Computação em Nuvem, onde os dados vão até as aplicações.

A ideia de Fog Computing é que os usuários finais ou dispositivos na borda da rede realizem tarefas que são majoritariamente funções da nuvem. Tarefas de controle e configuração da rede, armazenamento, processamento e gerenciamento podem ser feitas pelos clientes finais. Podemos usar alguns princípios básicos do TCP para realizarmos controle da rede feitos na borda da rede. Isso inclui princípios como: controle da velocidade de transmissão feito pelo cliente; uso do protocolo de janelas deslizantes; aumentar a taxa de transmissão de maneira aditiva, e reduzir de maneira multiplicativa; inferir o estado da rede por meio de feedback negativo; estimar a perda de pacotes e o atraso por meio de temporizadores. Colocando tudo isso junto, se tem inferência e controle da rede feitas pelos usuários finais, na borda da rede. Também há o controle de HetNets, onde o cliente final decide qual melhor RAT é melhor ser usado em determinado momento. Um algoritmo simples foi apresentado para esse caso na seção 3.1.2.1.

Clientes finais em Fog Computing também podem ser responsáveis pela medição do estado da rede. Isso pode ser interessante, pois esperar resultados vindo da nuvem podem ter latência não compatível com algumas aplicações. Os usuários finais podem trabalhar em conjunto, para um resultado mais preciso e eficiente. Os itens que são interessantes de serem medidos são o throughput, a carga da rede, por meio de inferência do uso do espectro, e quais aplicações estão usando a rede. Medidas como RSSI, SNR, RSRQ e RSRP são utilizadas para inferir o throughput, e essa inferência pode ser utilizada para adaptação de taxa de transmissão. A adaptação de taxa de transmissão pode ocorrer por meio de medições

de SNR por meio de indicadores de perda de pacote. Inferência sobre a carga da rede são feitas através de Medição Passiva, Probing Ativo, Correlação de performance da aplicação e Data Mining. A medição passiva consiste em apenas observar o RTT ou as razões SNR ou RSRQ. Probing ativo é quando se utiliza uma carga de prova e se mede o que acontece com essa carga. Correlação de performance é medir o que acontece no nível de aplicação, e o Data Mining utiliza um histórico sobre o que acontece na rede em determinado horário e local para aprender determinados padrões sobre a rede. Também pode-se medir a carga da rede por meio de inferência de uso de espectro, onde técnicas utilizam ativamente um sensor para medir a disponibilidade de espectro. Inferências no nível de aplicação podem usar DPI ou análise de porta. Pooling e caching na borda da rede pode reduzir a latência, evitar que todos os dados trafeguem para a nuvem, prevenindo gargalos, e pode promover offloading computacional. Pooling de recursos na borda pode ser feito através de virtualização ou cyber foraging. Dispositivos podem pedir uma ajuda a outros dispositivos próximos para realizar tarefas computacionais, a fim de ter mais poder de processamento ou mesmo utilizar uma maior largura de banda. A virtualização pode ser por meio de Cloudlets, que são pequenas nuvens. Funcionam como se fossem um pequeno data center, mas menos potente que a nuvem e servindo um menor número de usuários ao mesmo tempo. Cyber Foraging expande a ideia de offloading computacional para diversos dispositivos, onde tarefas podem ser particionadas e serem delegadas paralelamente a dispositivos vizinhos, chamados de surrogates. Pode-se usar recursos inexplorados ou ociosos que estão na borda da rede para servirem como ajuda a outros dispositivos.

Com a chegada da Internet das Coisas, muito mais dispositivos vão estar conectados à internet. Com o aumento do número dos dispositivos e com o novo ambiente onde eles podem modificar o ambiente físico, os desafios em relação à segurança e privacidade se tornam ainda maiores. Na camada de aplicação existem protocolos como S/MIME e SRTP, que provêm autenticação, integridade, não repúdio da origem e confidencialidade para os dados da aplicação. Na Internet das Coisas é interessante o uso do protocolo DTLS, que é similar ao TLS e funciona melhor em dispositivos com poucas capacidades computacionais. Na camada de rede há o protocolo de segurança IPSec, provendo serviços de integridade, autenticação e confidencialidade. Alguns desafios são mais relacionados ao uso ou ao desenvolvimento dos dispositivos. Problemas relacionado a configurabilidade de segurança estão presentes na Internet das Coisas. Usuários devem fazer uso de opções de segurança no dispositivo e desenvolvedores devem implementar essas medidas em seus dispositivos. Usuários e desenvolvedores devem estar atentos ao uso e criação de senhas e acesso granular a funções administrativas, bem como se preocuparem com a encriptação de credenciais que trafegam pela rede. Interfaces Web e interfaces de acesso à nuvem pedem senhas sejam fortes e que a interface não seja suscetível à ataques como XSS e SQLi. Softwares e Firmwares desatualizados abrem brechas para que dispositivos sejam facilmente atacados. Desenvolvedores de sistemas IoT devem se empenhar para proverem atualizações automáticas para seus dispositivos. Desenvolvedores devem tomar cuidado com o design de seus dispositivos, não

facilitando o acesso aos meios de armazenamento, evitando que usuários maliciosos tenham acesso ao sistema operacional ou dados sigilosos. Não bastante, dispositivos na Internet das Coisas podem ser muito limitados em seus recursos, o que os torna sujeito à falhas de hardware e a ataques de, por exemplo, buffer overflow. É um grande desafio criar medidas de segurança que sejam computacionalmente leves para esses dispositivos. E, por fim, existe a preocupação com a privacidade. Desenvolvedores devem se empenhar para usar e armazenas apenas os dados necessários para o funcionamento da aplicação, e dados devem ser sempre criptografados para que seja muito difícil uma entidade externa capturar e utilizar esses dados.

5.1 TRABALHOS FUTUROS

Fog Computing e Internet das Coisas abrem um mundo de oportunidades. É preciso um estudo mais aprofundado sobre as funcionalidades de uma arquitetura distribuída na borda da rede. Existem muitos desafios relacionados a IoT, como buscar protocolos mais eficientes para dispositivos limitados em recursos, e aplicações que usariam a arquitetura distribuída de Fog Computing, como tráfego urbano inteligente e cidades inteligentes.

ABBASOV, B. Cloud computing: State of the art reseach issues. In: *Application of Information and Communication Technologies (AICT), 2014 IEEE 8th International Conference on*. [S.l.: s.n.], 2014. p. 1–4.

ACCETURE. *The Internet of Things: The Future of Consumer Adoption*. [S.l.], 2014. Disponível em: <http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture-Internet-Things.pdf>.

ARYAFAR, E. et al. Rat selection games in hetnets. In: *INFOCOM, 2013 Proceedings IEEE*. [S.l.: s.n.], 2013. p. 998–1006. ISSN 0743-166X.

BONOMI, F. et al. Fog computing: A platform for internet of things and analytics. In: BESSIS, N.; DOBRE, C. (Ed.). *Big Data and Internet of Things: A Roadmap for Smart Environments*. Springer International Publishing, 2014, (Studies in Computational Intelligence, v. 546). p. 169–186. ISBN 978-3-319-05028-7. Disponível em: <http://dx.doi.org/10.1007/978-3-319-05029-4_7>.

BONOMI, F. et al. Fog computing and its role in the internet of things. In: *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*. New York, NY, USA: ACM, 2012. (MCC '12), p. 13–16. ISBN 978-1-4503-1519-7. Disponível em: <http://doi.acm.org/10.1145/2342509.2342513>.

BRACHMANN, M. et al. End-to-end transport security in the ip-based internet of things. In: *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*. [S.l.: s.n.], 2012. p. 1–5.

BREWER, E. Cap twelve years later: How the "rules"have changed. *Computer*, v. 45, n. 2, p. 23–29, Feb 2012. ISSN 0018-9162.

CAMP, J.; KNIGHTLY, E. Modulation rate adaptation in urban and vehicular environments: Cross-layer implementation and experimental evaluation. *Networking, IEEE/ACM Transactions on*, v. 18, n. 6, p. 1949–1962, Dec 2010. ISSN 1063-6692.

CHIANG, M. *Networked Life: 20 Questions and Answers*. New York, NY, USA: Cambridge University Press, 2012. ISBN 1107024943, 9781107024946.

CHUI, M.; LOFFLER, M.; ROBERTS, R. *The Internet of Things*. [S.l.], 2010. Disponível em: <http://www.mckinsey.com/insights/high_tech_telecoms_internet/the_internet_of_things>.

CIRANI, S.; FERRARI, G.; VELTRI, L. Enforcing security mechanisms in the ip-based internet of things: An algorithmic overview. *Algorithms*, v. 6, n. 2, p. 197, 2013. ISSN 1999-4893. Disponível em: <http://www.mdpi.com/1999-4893/6/2/197>.

Cisco VNI. *Global Mobile Data Traffic Forecast Update 2014Ű2019*. [S.l.], 2015. Disponível em: <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html>.

Cisco VNI. *The Zettabyte Era: Trends and Analysis*. [S.l.], 2015. Disponível em: <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/VNI_Hyperconnectivity_WP.pdf>.

CUI, A.; STOLFO, S. J. A quantitative analysis of the insecurity of embedded network devices: Results of a wide-area scan. In: *Proceedings of the 26th Annual Computer Security Applications Conference*. New York, NY, USA: ACM, 2010. (ACSAC '10), p. 97–106. ISBN 978-1-4503-0133-6. Disponível em: <http://doi.acm.org/10.1145/1920261.1920276>.

DAINOTTI, A.; PESCAPE, A.; CLAFFY, K. Issues and future directions in traffic classification. *Network, IEEE*, v. 26, n. 1, p. 35–40, January 2012. ISSN 0890-8044.

DIERKS, T.; RESCORLA, E. *The Transport Layer Security (TLS) Protocol Version 1.2*. IETF, ago. 2008. RFC 5246 (Proposed Standard). (Request for Comments, 5246). Updated by RFCs 5746, 5878, 6176. Disponível em: <http://www.ietf.org/rfc/rfc5246.txt>.

ENESCU, M. *From Cloud to Fog & The Internet of Things*. 2014. Disponível em: <http://pt.slideshare.net/MichaelEnescu/michael-enescu-keynote-chicago2014fromcloudtofogandiot>.

FERNANDO, N.; LOKE, S. W.; RAHAYU, W. Mobile cloud computing: A survey. *Future Generation Computer Systems*, v. 29, n. 1, p. 84 – 106, 2013. ISSN 0167-739X. Including Special section: AIRCC-NetCoM 2009 and Special section: Clouds and Service-Oriented Architectures. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167739X12001318>.

FINLEY, K. *Hacked Fridges Aren't the Internet of Things Biggest Worry*. 2015. Disponível em: <http://www.wired.com/2015/03/hacked-fridges-arent-internet-things-biggest-worry/>.

GANTI, R.; YE, F.; LEI, H. Mobile crowdsensing: current state and future challenges. *Communications Magazine, IEEE*, v. 49, n. 11, p. 32–39, November 2011. ISSN 0163-6804.

GARTNER. *IT Glossary: Internet of Things*. [S.l.], 2013. Disponível em: <http://www.gartner.com/it-glossary/internet-of-things/>.

GIL, A. *Como elaborar projetos de pesquisa*. Atlas, 2010. ISBN 9788522458233. Disponível em: <https://books.google.com.br/books?id=HSGHRAAACAAJ>.

GONSALVES, E. *Conversas sobre Iniciação à Pesquisa Científica*. Alinea, 2001. ISBN 9788575165492. Disponível em: <https://books.google.com.br/books?id=h-EbqAAACAAJ>.

HA, S.; RHEE, I.; XU, L. Cubic: A new tcp-friendly high-speed tcp variant. *SIGOPS Oper. Syst. Rev.*, ACM, New York, NY, USA, v. 42, n. 5, p. 64–74, jul. 2008. ISSN 0163-5980. Disponível em: <http://doi.acm.org/10.1145/1400097.1400105>.

HAFFNER, P. et al. Acas: Automated construction of application signatures. In: *Proceedings of the 2005 ACM SIGCOMM Workshop on Mining Network Data*. New York, NY, USA: ACM, 2005. (MineNet '05), p. 197–202. ISBN 1-59593-026-4. Disponível em: <http://doi.acm.org/10.1145/1080173.1080183>.

HALLER, S. *Internet of Things: An Integral Part of the Future Internet*. [S.l.], 2009. Disponível em: <http://services.future-internet.eu/images/1/16/A4_Things_Haller.pdf>.

HAMDI, M. Security of cloud computing, storage, and networking. In: *Collaboration Technologies and Systems (CTS), 2012 International Conference on*. [S.l.: s.n.], 2012. p. 1–5.

HP. *How safe are home security systems?* [S.l.], 2015. Disponível em: <http://www8.hp.com/h20195/V2/GetPDF.aspx/4AA5-7342ENW.pdf>.

KANUPARTHI, A.; KARRI, R.; ADDEPALLI, S. Hardware and embedded security in the context of internet of things. In: *Proceedings of the 2013 ACM Workshop on Security, Privacy & Dependability for Cyber Vehicles*. New York, NY, USA: ACM, 2013. (CyCAR '13), p. 61–64. ISBN 978-1-4503-2487-8. Disponível em: <http://doi.acm.org/10.1145/2517968.2517976>.

KRISTENSEN, M. Scavenger: Transparent development of efficient cyber foraging applications. In: *Pervasive Computing and Communications (PerCom), 2010 IEEE International Conference on*. [S.l.: s.n.], 2010. p. 217–226.

KUMAR, K. et al. A survey of computation offloading for mobile systems. *Mob. Netw. Appl.*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 18, n. 1, p. 129–140, fev. 2013. ISSN 1383-469X. Disponível em: <http://dx.doi.org/10.1007/s11036-012-0368-0>.

LIANG, Y.-C. et al. Cognitive radio networking and communications: an overview. *Vehicular Technology, IEEE Transactions on*, v. 60, n. 7, p. 3386–3407, Sept 2011. ISSN 0018-9545.

MELL, P.; GRANCE, T. The NIST definition of cloud computing. 2011.

MODADUGU, N.; RESCORLA, E. *Datagram Transport Layer Security Version 1.2*. IETF, 2012. RFC 6347 (Proposed Standard). (Request for Comments, 6347). Disponível em: <http://tools.ietf.org/rfc/rfc6347.txt>.

MOSKOWITZ, R. et al. *Host identity protocol version 2 (HIPv2)*. [S.l.], 2015.

OWASP. *OWASP Internet of Things Top 10*. 2014. Disponível em: <https://www.owasp. org/index.php/OWASP_Internet_of_Things_Top_Ten_Project>.

PEFKIANAKIS, I. et al. Toward history-aware robust 802.11 rate adaptation. *Mobile Computing, IEEE Transactions on*, v. 12, n. 3, p. 502–515, March 2013. ISSN 1536-1233.

QUER, G. et al. Cognitive network inference through bayesian network analysis. In: *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*. [S.l.: s.n.], 2010. p. 1–6. ISSN 1930-529X.

RAMOS, P. H. S. Neutralidade da rede e o marco civil da internet: um guia para interpretacao. Sep 2014.

RASHWAN, A.; TAHA, A.-E.; HASSANEIN, H. Characterizing the performance of security functions in mobile computing systems. *Internet of Things Journal, IEEE*, v. 1, n. 5, p. 399–413, Oct 2014. ISSN 2327-4662.

ROMAN, R.; ZHOU, J.; LOPEZ, J. On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, Elsevier, v. 57, n. 10, p. 2266–2279, 2013.

SATYANARAYANAN, M. et al. The case for vm-based cloudlets in mobile computing. *Pervasive Computing, IEEE*, v. 8, n. 4, p. 14–23, Oct 2009. ISSN 1536-1268.

SAUTER, M. *From GSM to LTE: An Introduction to Mobile Networks and Mobile Broadband*. Wiley, 2010. 160 p. (Wiley Online Library: Books). ISBN 9780470978221. Disponível em: <https://books.google.com.br/books?id=uso-6LN2YjsC>.

SCHNEIER, B. *The Internet of Things Is Wildly Insecure, And Often Unpatchable*. 2014. Disponível em: <http://www.wired.com/2014/01/theres-no-good-way-to-patch-the-internet-of-things-and-thats-a-huge-problem/>.

SEO, K.; KENT, S. Security architecture for the internet protocol. 2005.

SHELBY, Z.; HARTKE, K.; BORMANN, C. The constrained application protocol (coap). 2014.

SMITH, C.; MIESSLER, D. *Internet of Things Research Study*. [S.l.], 06 2014. Disponível em: <http://www8.hp.com/h20195/V2/GetPDF.aspx/4AA5-4759ENW.pdf>.

TU, S.-Y.; CHEN, K.-C.; PRASAD, R. Spectrum sensing of ofdma systems for cognitive radio networks. *Vehicular Technology, IEEE Transactions on*, v. 58, n. 7, p. 3410–3425, Sept 2009. ISSN 0018-9545.

UN. *UN expert applauds US decision guaranteeing net neutrality*. 2015. Disponível em: <http://www.un.org/apps/news/story.asp?NewsID=50200\#.VYd-IPlVhBd>.

WANG, W.; LU, Z. Cyber security in the smart grid: Survey and challenges. *Computer Networks*, Elsevier, v. 57, n. 5, p. 1344–1371, 2013.

WONG, S. H. Y. et al. Robust rate adaptation for 802.11 wireless networks. In: *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*. New York, NY, USA: ACM, 2006. (MobiCom '06), p. 146–157. ISBN 1-59593-286-0. Disponível em: <http://doi.acm.org/10.1145/1161089.1161107>.

ZANELLA, A. et al. Internet of things for smart cities. *Internet of Things Journal, IEEE*, v. 1, n. 1, p. 22–32, Feb 2014. ISSN 2327-4662.

ZHU, J. et al. Improving web sites performance using edge servers in fog computing architecture. In: *Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on*. [S.l.: s.n.], 2013. p. 320–323.