

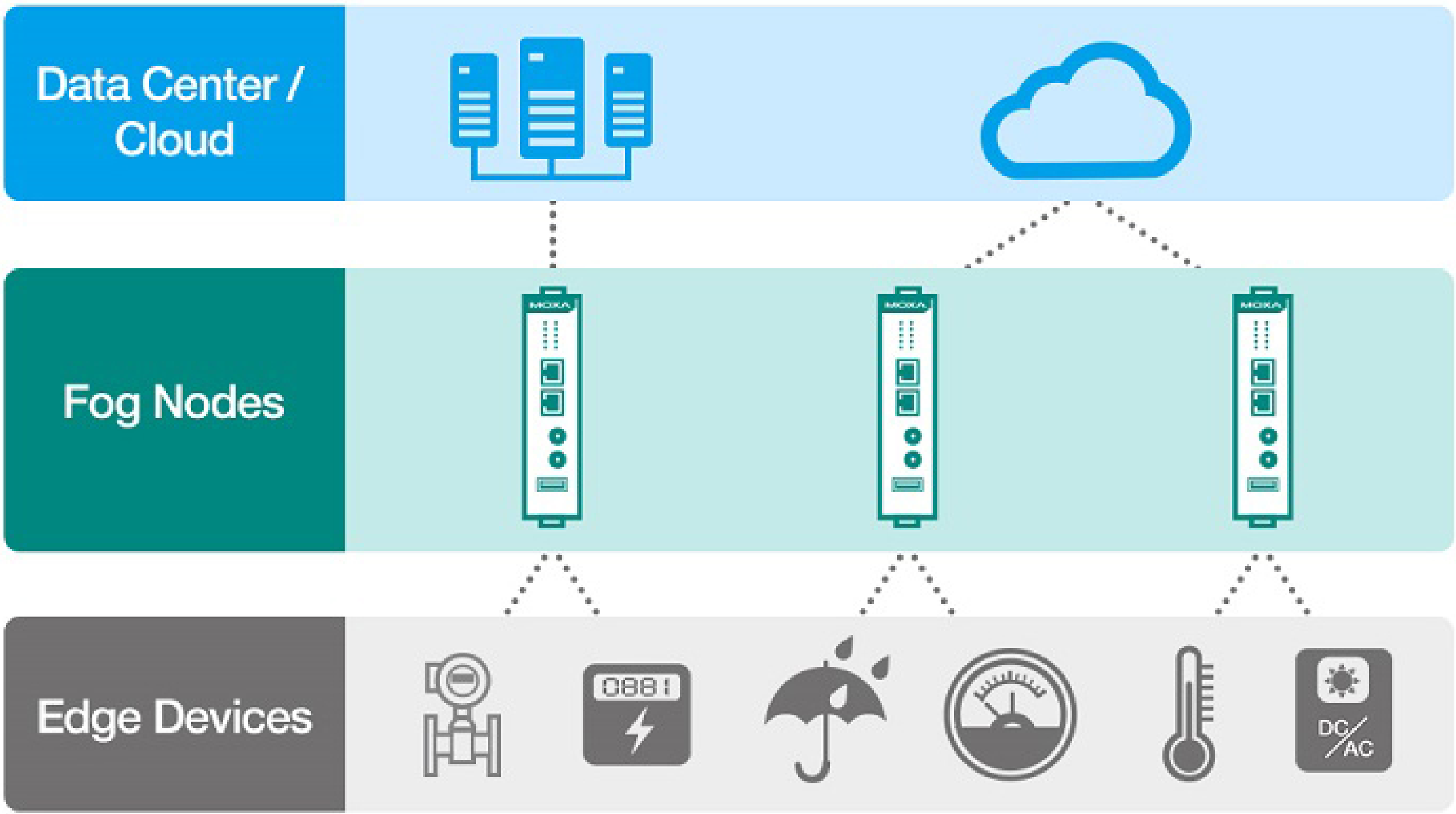
# Mapeamento de recursos sob computação em névoa

Felipe Brizola Bergues Duro  
Orientador: Prof. Dr. Sérgio Johann Filho  
Ciência da Computação  
✉ felipe.duro@acad.pucrs.br



## Introdução

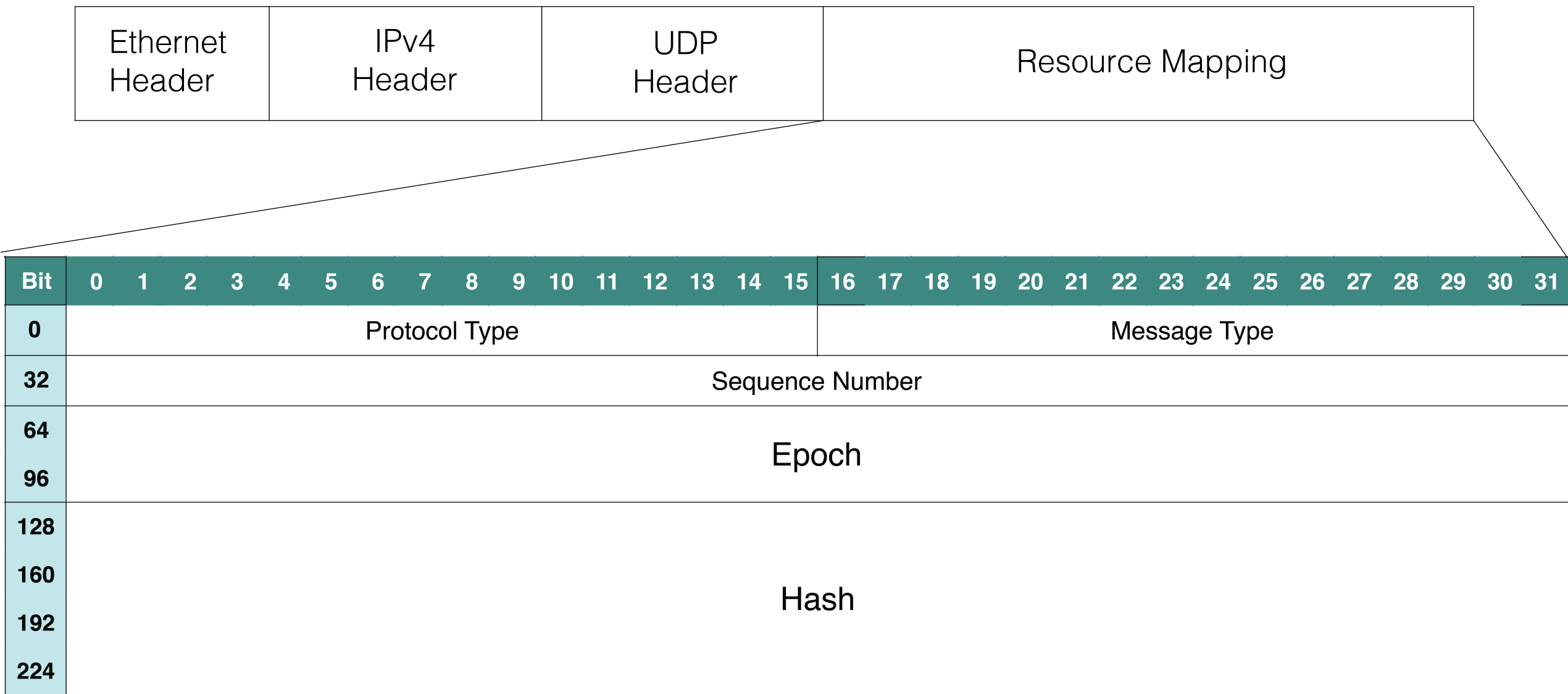
- A computação em névoa é um termo relativamente novo e teve sua primeira definição, dada pela Cisco Systems em 2012, como uma extensão do modelo de computação em nuvem que provê armazenamento, computação e serviços de rede entre dispositivos finais e os servidores na nuvem [2].
- A computação em névoa tornou-se um paradigma próprio deixando de ser um apêndice da computação em nuvem. Deste novo paradigma surgiu o termo de *fog node*, que abrange desde dispositivos finais com baixa capacidade computacional até servidores poderosos na nuvem.
- Os problemas referentes à padronização, descoberta e sincronização são os motivadores deste trabalho, uma vez que atualmente não existem mecanismos no qual um membro da névoa, seja ele um dispositivo com limitações de memória e processamento ou um computador robusto, mapeiem os recursos disponíveis e divulgue os seus na rede.
- Para que este objetivo fosse atingido, foi projetado um protocolo de rede simples que é executado em redes locais e que atua de maneira automatizada e descentralizada no processo de descobrimento e sincronização de recursos.
- A computação em névoa pode ser ilustrada pela imagem abaixo, na qual demonstra uma arquitetura base deste cenário.



- A comunicação entre os *fog nodes* e seus sensores, comumente chamados de *edge devices*, ocorre utilizando o protocolo CoAP - Constrained Application Protocol.
- O CoAP implementa a RFC-6690 - Constrained RESTful Environments (CoRE) Link Format - que possui como finalidade a realização de REST em nodos com recursos limitados[3].
- CoAP também implementa a RFC-5785 que define como URI padrão o prefixo `/.well-known/core`. Este prefixo é utilizado para que o servidor exponha suas políticas e recursos disponíveis[1].

## Protocolo Proposto

- A imagem a seguir ilustra a pilha de protocolos utilizados na implementação deste projeto, bem como o detalhamento estrutural do protocolo proposto, intitulado Resource Mapping.



- O campo Protocol Type é reservado para indicar a forma de encapsulamento do pacote.
- Atualmente existem dois tipos de Messages Types possíveis, keep alive e acknowledgement.
- Sequence Number tem o intuito de identificar o pacote enviado.
- O campo Epoch é utilizado para indicar alterações nos recursos providos pelo fog node. Sendo assim, quando um recurso é adicionado ou removido de um fog node, o campo Epoch é acrescido em uma unidade.
- O campo Hash utiliza a função de criptografia MD5 para validar a integridade dos demais campos contidos no pacote.
- Os fog nodes, a cada trinta segundos, enviam mensagens por multicast indicando que ainda estão em operação. Essas mensagens são do tipo keep alive e carregam consigo a época em que o fog node se encontra.
- Ao receber as mensagens de keep alive, o fog node valida se já possui este IP mapeado em sua lista de recursos globais. Caso não possua o IP, deverá, então, realizar uma requisição CoAP para a URI `/.well-known/core` a fim de obter os recursos providos pelo fog node. Caso já possua o IP mapeado, deverá, então, verificar se a época recebida é diferente da que possui. Havendo divergência entre as épocas, uma nova requisição CoAP para a URI `/.well-known/core` deverá ser realizada com o intento de atualizar a listagem de recursos providos pelo node.
- Após receber a mensagem de keep alive e processá-la, outra mensagem, desta vez de acknowledgement, é enviada de volta ao remetente do keep alive. Desta forma, o remetente passa a ter conhecimento que o fog node ainda está em operação.
- Para que seja possível saber quando um membro deixou de fazer parte da névoa, o mecanismo proposto funciona de tal forma que a cada três timeouts de acknowledgement o fog node é removido da lista de recursos globais.

## Experimentos

- O ambiente de testes da solução foi construído utilizando containers Docker e scripts Python.
- O gerenciamento dos containers na névoa é automatizado, portanto, é possível dinamicamente adicionarmos ou removermos containers da rede.
- O gerenciamento dos recursos de um determinado fog node também foi automatizado, sendo assim, é possível incluirmos ou retirarmos recursos dinamicamente.
- Os experimentos se mostraram satisfatórios, uma vez que foi possível criar um ambiente com mais de cem fog nodes e a sincronização dos recursos transcorreu normalmente. Esta limitação dá-se pelo fato do computador hospedeiro, que executa dos containers, não possuir recursos suficientes para a expansão da escalabilidade dos testes.

## Referências

[1] Eran Hammer-Lahav and Mark Nottingham. Defining Well-Known Uniform Resource Identifiers (URIs). RFC 5785, April 2010.

[2] Rodrigo Roman, Javier Lopez, and Masahiro Mambo. Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. *CoRR*, abs/1602.00484, 2016.

[3] Zach Shelby. Constrained RESTful Environments (CoRE) Link Format. RFC 6690, August 2012.