
UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

PROGRAMA DE PÓS GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ACQUILA SANTOS ROCHA

# Secure D2D Caching Framework Based on Trust Management and Blockchain for Mobile Edge Caching - A Multi Domain Approach

Goiânia
2023

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

# TERMO DE CIÊNCIA E DE AUTORIZAÇÃO (TECA) PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES

# E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a Lei 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo das Teses e Dissertações disponibilizado na BDTD/UFG é de responsabilidade exclusiva do autor. Ao encaminhar o produto final, o autor(a) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

## 1. Identificação do material bibliográfico

[ x ] Dissertação [ ] Tese [ ] Outro*:_____

*No caso de mestrado/doutorado profissional, indique o formato do Trabalho de Conclusão de Curso, permitido no documento de área, correspondente ao programa de pós-graduação, orientado pela legislação vigente da CAPES.

**Exemplos**: Estudo de caso ou Revisão sistemática ou outros formatos.

## 2. Nome completo do autor

Acquila Santos Rocha

## 3. Título do trabalho

Secure D2D Caching Framework Based on Trust Management and Blockchain for Mobile Edge Caching - A Multi Domain Approach

## 4. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador)

Concorda com a liberação total do documento [ X ] SIM [ ] NÃO[1]

**[1]** Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante:
**a)** consulta ao(à) autor(a) e ao(à) orientador(a);
**b)** novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo da tese ou dissertação.
O documento não será disponibilizado durante o período de embargo.
Casos de embargo:
- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;

- Publicação da dissertação/tese em livro.

**Obs. Este termo deverá ser assinado no SEI pelo orientador e pelo autor.**

Documento assinado eletronicamente por **Vinicius Da Cunha Martins Borges**, **Professor do Magistério Superior**, em 18/09/2023, às 08:50, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Acquila Santos Rocha**, **Discente**, em 18/09/2023, às 08:52, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4047971** e o código CRC **875BDF31**.

**Referência:** Processo nº 23070.041361/2023-82 SEI nº 4047971

ACQUILA SANTOS ROCHA

# Secure D2D Caching Framework Based on Trust Management and Blockchain for Mobile Edge Caching - A Multi Domain Approach

Master's thesis submitted to the Graduate Program of Instituto de Informática of Universidade Federal de Goiás, in partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

**Area:** Computer Science.

**Mentor:** Prof. Dr. Vinicius da Cunha Martins Borges

**Co-Mentor:** Dr. Billy Anderson Pinheiro

Goiânia
2023

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Rocha, Acquila Santos
Secure D2D Caching Framework Based on Trust Management and Blockchain for Mobile Edge Caching - A Multi Domain Approach [manuscrito] / Acquila Santos Rocha. - 2023.
CV, 105 f.

Orientador: Prof. Dr. Vinicius Cunha Martins Borges; co-orientador Dr. Billy Anderson Pinheiro.
Dissertação (Mestrado) - Universidade Federal de Goiás, , Programa de Pós-Graduação em Ciência da Computação, Goiânia, 2023.
Bibliografia. Apêndice.
Inclui siglas, símbolos, gráfico, tabelas, algoritmos, lista de figuras, lista de tabelas.

1. D2D Caching. 2. Trust management. 3. Blockchain. 4. Security. 5. 5G Networks. I. Cunha Martins Borges, Vinicius, orient. II. Título.

CDU 004

UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

**ATA DE DEFESA DE DISSERTAÇÃO**

Ata nº **13** da sessão de Defesa de Dissertação de **Acquila Santos Rocha**, que confere o título de Mestre em Ciência da Computação, na área de concentração em Ciência da Computação.

Aos dezoito dias do mês de agosto de dois mil e vinte e três, a partir das catorze horas, na sala 257 do INF, realizou-se a sessão pública de Defesa de Dissertação intitulada **"Secure D2D Caching Framework Based on Trust Management and Blockchain for Mobile Edge Caching - A Multi Domain Approach"**. Os trabalhos foram instalados pelo Orientador, Professor Doutor Vinicius da Cunha Martins Borges (INF/UFG) com a participação dos demais membros da Banca Examinadora: Doutor Billy Anderson Pinheiro, coorientador; Professor Doutor Weverton Luis da Costa Cordeiro (UFRGS), membro titular externo; Professor Doutor Sérgio Teixeira de Carvalho, membro titular interno. A participação dos membros Billy Anderson Pinheiro e Weverton Luis da Costa Cordeiro e discente Acquila Santos Rocha ocorreram por meio de videoconferência. Durante a arguição os membros da banca não fizeram sugestão de alteração do título do trabalho. A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Dissertação, tendo sido o candidato **aprovado** pelos seus membros. Proclamados os resultados pelo Professor Doutor Vinicius da Cunha Martins Borges, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora, aos dezoito dias do mês de agosto de dois mil e vinte e três.

TÍTULO SUGERIDO PELA BANCA

Documento assinado eletronicamente por **Weverton Luis da Costa Cordeiro**, **Usuário Externo**, em 18/08/2023, às 16:26, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Sérgio Teixeira De Carvalho**, **Professor do Magistério Superior**, em 18/08/2023, às 16:26, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Billy Anderson Pinheiro**, **Usuário Externo**, em 18/08/2023, às 16:26, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Vinicius Da Cunha Martins Borges**, **Professor do Magistério Superior**, em 18/08/2023, às 16:26, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Acquila Santos Rocha**, **Discente**, em 18/08/2023, às 16:36, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **3927932** e o código CRC **8C2C0FCD**.

**Referência:** Processo nº 23070.041361/2023-82 SEI nº 3927932

**Acquila Santos Rocha**

Graduated in Computer Science from UFG - Universidade Federal de Goiás. Participated in the UFG Scientific Research Initiation Program in 2019-2020. Was a Teaching Assistant for the subjects of Discrete Mathematics in 2017 and Analysis and Design of Algorithms from 2018 to 2019. Currently working as a Software Engineer at Amazon.

*I dedicate this work to my family, my friends, and my advisor for all their support and attention during this journey*.

# Special Thanks

First and foremost, I want to express my deep gratitude to my family for their unwavering support and encouragement throughout this journey, with a special mention of my mother, Jane Darley Alves dos Santos, and my father, João Alves Rocha. They have consistently been my pillars of strength, offering wisdom, affection, and undivided attention in the most challenging moments. The values and lessons they instilled in me have been instrumental in the successful completion of this endeavor.

I extend heartfelt thanks to my circle of friends, whose love and support have been invaluable in guiding me through critical decisions and providing unwavering companionship during the toughest times. I firmly believe that without their presence, none of this would have been achievable.

My gratitude also goes to my mentor, Vinicius Cunha, whose guidance and support have played a pivotal role in my academic journey. The advice and knowledge imparted by him will forever remain a beacon in my academic and professional pursuits.

I'd like to acknowledge and appreciate all the educators who have contributed to my educational journey. Your mentorship and teachings have transformed my critical and theoretical thinking, shaping me into the person I am today.

"It's difficult in times like these: ideals, dreams and cherished hopes rise within us, only to be crushed by grim reality. It's a wonder I haven't abandoned all my ideals, they seem so absurd and impractical. Yet I cling to them because I still believe, in spite of everything, that people are truly good at heart"

**Anne Frank**,
*The Diary of Anne Frank - 1944*.

# Resumo


Rocha, Acquila Santos. **Secure D2D Caching Framework Based on Trust Management and Blockchain for Mobile Edge Caching - A Multi Domain Approach**. Goiânia, 2023. 104p. Dissertação de Mestrado. Programa de Pós Graduação em Ciência da Computação, Instituto de Informática, Universidade Federal de Goiás.

Comunicação Dispositivo-a-Dispositivo (D2D), combinado com cache de borda e computação de borda móvel, é uma abordagem promissora que permite a transferência de dados da rede móvel sem fio. No entanto, a segurança do usuário ainda é uma questão em aberto na comunicação D2D. Vulnerabilidades de segurança ainda são possíveis devido às interações diretas, espontâneas e fáceis entre usuários não confiáveis e diferentes graus de mobilidade. Esta dissertação abrange o design de um framework multicamada que combina diversas tecnologias inspiradas na blockchain para criar um framework de cache D2D multidomínio seguro. No que diz respeito ao aspecto intradomínio, estabelecemos o framework de cache D2D seguro inspirado em gerenciamento de confiança e blockchain (SeCDUB) para melhorar a segurança da comunicação D2D no cache de vídeo, por meio da combinação de observações diretas e indiretas. Além disso, os conceitos de blockchain foram adaptados ao cenário dinâmico e restrito das redes D2D para impedir a interceptação e alteração de dados de observações indiretas. Essa adaptação considerou o desenvolvimento de uma Abordagem de Clusterização (CA) que permite um blockchain escalável e leve para redes D2D. Foram usados dois modelos matemáticos de incerteza diferentes para inferir valores de confiança direta e indireta: inferência bayesiana e a Teoria de Dempster Shafer (TDS), respectivamente. No que diz respeito à abordagem interdomínio, desenvolvemos o framework Trust in Multiple Domains (TrustMD). Essa abordagem combina o armazenamento de confiança de borda com a blockchain para o gerenciamento de armazenamento distribuído em uma arquitetura de várias camadas, projetada para armazenar eficientemente dados de controle de confiança na borda em diferentes domínios. Quanto aos resultados coletados, realizamos simulações para testar a abordagem intradomínio do SecDUB. A abordagem de clusterização proposta desempenha um papel fundamental na mitigação do overhead do SecDuB, bem como no tempo de consenso. Os resultados do TrustMD demonstraram um aumento significativo

no goodput, atingindo 95% do throughput total da rede quando comparado com a abordagem que emprega apenas o SecDUB. Mesmo que tenha havido um aumento de 7% no overhead D2D, o TrustMD controla efetivamente os níveis de latência, resultando em uma ligeira diminuição de 1,3 segundos.Portanto, os resultados alcançados indicam que o TrustMD gerencia a segurança de forma eficiente sem comprometer o desempenho da rede, reduzindo a taxa de falsos negativos em até 31% no melhor cenário. Na verdade, a combinação do SecDUB e do TrustMD oferece uma solução de segurança escalável e eficaz que impulsiona o desempenho da rede e garante proteção robusta.

**Palavras–chave**

D2D Caching, Gestão de Confiança, Blockchain, Segurança, Redes 5G, Computação de Borda Móvel, Multi-domínio.

# Abstract

Rocha, Acquila Santos. **Secure D2D Caching Framework Based on Trust Management and Blockchain for Mobile Edge Caching - A Multi Domain Approach**. Goiânia, 2023. 104p. MSc. Dissertation. Programa de Pós Graduação em Ciência da Computação, Instituto de Informática, Universidade Federal de Goiás.

Device–to-Device communication (D2D), combined with edge caching and mobile edge computing, is a promising approach that allows offloading data from the wireless mobile network. However, user security is still an open issue in D2D communication. Security vulnerabilities remain possible owing to easy, direct and spontaneous interactions between untrustworthy users and different degrees of mobility. This dissertation encompasses the designing of a multi-layer framework that combines diverse technologies inspired in blockchain to come up with a secure multi domain D2D caching framework. Regarding the intra-domain aspect we establish Secure D2D Caching framework inspired on trUst management and Blockchain (SeCDUB) to improve the security of D2D communication in video caching, through the combination of direct and indirect observations. In addition, blockchain concepts were adapted to the dynamic and restricted scenario of D2D networks to prevent data interception and alteration of indirect observations. This adaptation considered the development of a Clustering Approach (CA) that enables scalable and lightweight blockchain for D2D networks. Two different uncertainty mathematical models were used to infer direct and indirect trust values: Bayesian inference and the Theory Of Dempster Shafer (TDS) respectively. Regarding the inter-domain approach we developed Trust in Multiple Domains (TrustMD) framework. This approach combines edge trust storage with blockchain for distributed storage management in a multi layer architecture, designed to efficiently store trust control data in edge across different domains. Regarding the collected results, we performed simulations to test SecDUB's intra-domain approach. The proposed clustering approach plays a key role to mitigate the SecDuB overhead as well as the consensus time. TrustMD results demonstrated a significant enhancement in goodput, reaching at best, 95% of the total network throughput, whicle SecDUB achieved approximately 80%. Even though there was a 7% increase in D2D overhead, TrustMD effectively keep control of latency levels, resulting in a slight decrease of 1.3 seconds. Hence, the achieved results indicates that TrustMD efficiently manages security

without compromising network performance reducing false negative rate up to 31% on the best case scenario. Actually, the combination of SecDUB and TrustMD offers a scalable and effective security solution that boosts network performance and ensures robust protection.

**Keywords**

D2D Caching, Trust management, Blockchain, Security, 5G Networks, Mobile Edge Computing, Multi-domain.

# Publications

**Acquila Santos Rocha**, **Billy Anderson Pinheiro**, **Vinicius C.M. Borges**, *Secure D2D caching framework inspired on trust management and blockchain for Mobile Edge Caching*, Pervasive and Mobile Computing, Volume 77, 2021, 101481, ISSN 1574-1192, https://doi.org/10.1016/j.pmcj.2021.101481 (https://www.sciencedirect.com/science/article/pii/S1574119221001115) **[Qualis A2]**.

**Acquila Santos Rocha**, **Billy Anderson Pinheiro**, **Vinicius C.M. Borges**, *Framework de caching D2D seguro baseado em gerenciamento de confiança e blockchain para borda da rede móvel*, III Brazilian Workshop on Smart Cities - WBCI (part of CSBC2022), available on: https://www.youtube.com/watch?v=H5JU8dJ0jNQ accessed on 3rd of August of 2023.

# Contents

# List of Figures

# List of Tables

# List of Algorithms

# Glossary

**3GPP** Third Generation Partnership Project. 22
**5G** Fifth-generation of Mobile Networks. 22, 23, 49, 51

**ADT** Average Degree of Trust. 65, 86

**BS** Base Station. 23, 29, 36, 38, 40, 51

**CDN** Content Delivery Network. 23, 24
**CH** Cluster Head. 39, 42–46, 55–57, 64, 67, 76, 77
**CM** Cluster Member. 42, 44, 46, 47, 101

**D2D** Device–to-Device communication. 22–24, 26–30, 32–46, 50, 51, 63–65, 67, 68, 72, 74, 77, 83, 86, 88–91
**DC** Domain Chain. 54, 58, 59, 61, 62, 73, 79
**DCO** Domain Controller. 53, 54, 58–62, 77
**DLP** Discrete Logarithm Problem. 64
**Domain-CM** Domain Chain Manager. 54, 59, 61, 62
**Domain-HC** Domain Handover Component. 53, 54, 59
**Domain-QC** Domain Query Component. 53, 54, 61, 62
**DRL** Deep Reinforcement Learning. 50, 51, 91
**DT** Degree of Trust. 24, 29, 101
**DTO** Data Transfer Object. 54

**EC** Edge Chain. 53, 54, 56, 57, 59, 61, 62, 76, 79
**ECDH** Elliptic Curve Diffie Hellman. 64
**ECDLP** Eliptic Curve Discrete Logarithm Problem. 64
**ECDSA** Eliptic Curve Digital Signature Algorithm. 64
**Edge-CM** Edge Chain Manager. 53, 54, 56, 57, 59, 61, 62
**Edge-QC** Edge Query Component. 53, 54, 60, 62
**ETSI** European Telecommunications Standards Institute. 59

**FNR** False Negative Rate. 65, 86

**Inter-TUF** Inter Domain Trust Update Flow. 54, 58, 73, 79

**Intra-TUF** Intra Domain Trust Update Flow. 53, 54, 76, 79

**LS** Location Service. 59

**M2M** Machine to Machine. 22
**MEC** Mobile Edge Computing. 22, 23, 26, 27, 29, 36, 48, 58, 59, 82, 86, 89–91
**MEH** MEC Host. 23, 25, 26, 53–59, 61, 62, 77, 86

**NFV** Network Function Virtualization. 22
**NS-3** Network Simulator 3. 63, 64, 73

**PBFT** Pratical Byzantine Fault Tolerance. 34, 37, 43, 45, 50, 66, 68
**PoW** Proof-of-work. 34, 44, 45
**ProSoCaD** Probabilistic Social Cascade For caching in D2D communication. 28, 29, 38, 40, 42, 43, 63–66, 69, 71, 88, 89

**QoS** Quality of Service. 22, 23

**RNIS** Radio Network Information Service. 59
**RPGM** Reference Point Group Mobility. 64

**SDN** Software Defined Networking. 22, 23
**SecDUB** Secure D2D caching based on Trust Management. 17, 26, 35, 36, 48, 51, 53–55, 63–66, 69–74, 76, 77, 81, 83–86, 88–90

**TDS** Theory of Dempster-Shafer. 31, 32, 56, 57, 62, 100–102
**TPS** Transaction per Second. 78, 81
**TQF** Trust Query Flow. 53, 54, 60, 62, 76, 77, 79, 80
**TrustMD** Trust in Multiple Domains. 17, 35, 48, 49, 53, 54, 72–78, 80–82, 85, 86, 88–90

**UE** User Equipment. 23, 34, 36, 38, 46, 48, 50, 52–56, 59, 60, 62, 76, 91
**URLLC** Ultra-reliable Low Latency Communication. 22

**VANET** Veicular Ad Hoc Networks. 49, 50

CHAPTER 1

# Introduction

Future wireless and mobile networks predict a steady growth in the number of connected devices, which will increase the amount of traffic and boost the demand for higher transmission rates. It is estimated that by 2023 global video traffic in mobile networks will show significant growth due to the emergence of high network demand applications such as smart car navigation systems and virtual reality. It is estimated that Machine to Machine (M2M) connections will be half of the global connected devices connections and over 70 percent of the global population will have mobile connectivity which contributes to a peak in mobile connections. This context represents a great challenge for the next generation of wireless networks [14] in terms of bandwidth and latency.

This extensive communication scenario imposes a series of new requirements in the Fifth-generation of Mobile Networks (5G). Considering this, the Third Generation Partnership Project (3GPP) has specified an architecture for 5G mobile network core that supports high data rate, high speed, and low latency, therefore enabling mobile applications based on Device–to-Device communication (D2D) communications [56, 30, 23]. To achieve the requirements and leverage Quality of Service (QoS), new core softwarization technologies are being explored, such as Software Defined Networking (SDN) and Network Function Virtualization (NFV). With these technologies 5G can meet the great demand for a fast forwarding information base in the data plane and context-aware decision-making in control planes [19].

Although 5G-core softwarization facilitates the diverse requirements of new communication scenarios, there is some drawbacks of a generic cloud service provisioning infrastructure. Conventional cloud computing architecture fails to provide high quality services, owing to the geographical placement of data-centers alongside with limited access capacity which makes it difficult to meet demands on larger scale [44]. In this scenario, Mobile Edge Computing (MEC) has emerged as a shift from centralized cloud computing [67], where the integration of 5G and MEC: (1) enable Ultra-reliable Low Latency Communication (URLLC), (2) improve security and also (3) contributes in data offloading of network core [22]. MEC has emerged as a shift from centralized cloud com-

puting, where MEC edge caching servers are distributed in the mobile network, to cache nodes closer to the mobile users that can store popular contents [67].

As an offloading opportunity MEC service providers can set up Content Delivery Network (CDN) at the mobile edge so that edge can store popular content. The core differences between traditional CDN and MEC-based CDN relies mainly on the ability to quickly process/store data in edge and consequently accelerate service and content provision by placing contents near the requesting User Equipment (UE) [16]. However MEC-based CDN networks can be compromised due to mobility, since as devices travel across cells boundaries, the network must be able to locate devices keeping the QoS and security levels [2].

The implementation of SDN can be a great deal for mobility management in 5G and beyond networks [36]. Usually each SDN domain is managed by a SDN controller responsible to be the head of the underlying network. In a mobile environment, nodes might travel across different domains, with that said MEC-based CDN systems should be capable of follow privacy and security requirements [45]. Also envisioning an architecture that still complies with mobility requirements, MEC Host (MEH) collaboration emerges as an opportunity to enhance edge security capabilities regarding mobility, by extending services across a larger area [65, 21].

In the light of the growth in traffic, content caching in UE is itself one of the alternatives for improving the efficiency of the use of mobile communication, since stored cache files can be distributed among nearby UEs through opportunistic connections. D2D communication is a promising communication paradigm to minimize the UEs interaction with the Base Station (BS) and enhance offload. Traditional D2D use cases incorporate surrounding observations dedicated to monitoring ecological disasters [1]. However, social development has brought new functionalities to D2D, including social networking, health care services and delay tolerant emergency services [56]. In the light of the 5G communication, D2D is proposed as an alternative to stimulate edge communication, through social networking and caching technologies, avoiding overloading the network core. The D2D connections are opportunistic and seek to increase the offloading of data in the BS as it implements MEC for automatic distribution of content between the nodes. Furthermore, D2D can, as a practical example, expand the coverage of the cellular network through multi-hop routing (relay communication).

Although D2D is a mobile edge communication technology that will significantly contribute to the success of 5G networks, its effectiveness depends a good deal on user security in D2D networks. As there is a direct and opportunistic interaction between neighboring devices, ordinary users will be under the threat of malicious users. In addition, the system of re-transmission of packets with multiple hops in environments with many nodes (i.e. dense scenarios) leads to a greater number of users, thus attracting more

malicious individuals, that is, a greater number of attacks. Finally, user mobility causes volatility which make it difficult to identify the malicious users' accurately. Hence, security in D2D is a challenging task.

Secure and distributed communication in D2D systems encourages users to adopt the technology and enhances the useful data offloading in the core network, which improves spectral efficiency and reduces the communication latency of devices with a base station [54]. This solution enables autonomous security and distributed management, even in the absence of cellular coverage. To achieve this goal, we have set out a strategy for valid video content caching, based on the combination of two concepts: collaborative trust management and blockchain.

Trust management in communication has been used in different multi hop networks such as D2D for different purposes. These approaches have been adopted by [57, 9, 25, 3, 43, 39, 59] who recommended strategies for collaborative trust management that focused on improving the security and efficiency of packet delivery, traffic routing, clustering and D2D communication. The authors [12] and [64], define the user's trustworthiness as a value called Degree of Trust (DT). By assessing trust on the basis of behavioral criteria for assessing trust, we are able to shape a user's pattern of behavior when the DT shows that the evaluated user is likely to engage in malicious activities [8]. In other words, the malicious user is someone whose reputation is assessed as "weak" on the basis of the amount of invalid shared content.

When it comes for collaborative trust assessment, in the particular context of D2D networks, the collection of opinions from neighboring nodes in the control plan is subject to interceptions and falsifications when security systems are not applied, a factor that has an adverse effect on the trust management mechanism and undermines its efficiency. The blockchain has various features that trust management can benefit from, however the literature ignores the fact that indirect behavior observations, shared between different nodes, can be intercepted and falsified owing to the lack of a suitable security mechanism for the task. Moreover, the traditional blockchain technology cannot be fully applied in D2D networks, since D2D nodes have various degrees of mobility (vehicles, trains and bus passengers, cyclists, etc) and are subject to a wide range of restrictions on processing power and energy. Hence, it is necessary to adapt the concepts of blockchain to the context of D2D video caching.

By using trust on the basis of behavioral criteria, we can shape the user's behavior pattern in network and predict its likelihood to engage in malicious activities [8]. Analyzing the state of art in edge caching systems and CDN networks, we can highlight authors efforts to enhance security and privacy using trust assessment techniques [47, 60, 61, 25, 39, 32, 70, 63, 4, 58]. However, we could not evidence a work focused on a multi-domain architecture in which user behavior/trustworthiness is retrievable

across different areas and domains. In mobile scenarios, users are more likely to cross unknown areas or domains where they will communicate with untrustworthy users, since a trust/behavioral information regarding this user is not yet present (Figure 1.1). Using a simple intra-domain trust assessment approach in high movement scenarios, the time needed to build neighbor nodes history may be longer than the time the node will remain in that same area, allowing malicious nodes to freely perform attacks in new vulnerable nodes. A way to mitigate this issue is by allowing devices to securely/rapidly retrieve trustworthiness values from edge MEC Host. For that matter we propose to create a multi-domain framework capable of securely distribute/persist trust information across different areas and domains and hopefully guarantee low latency trust query operations in MEHs.

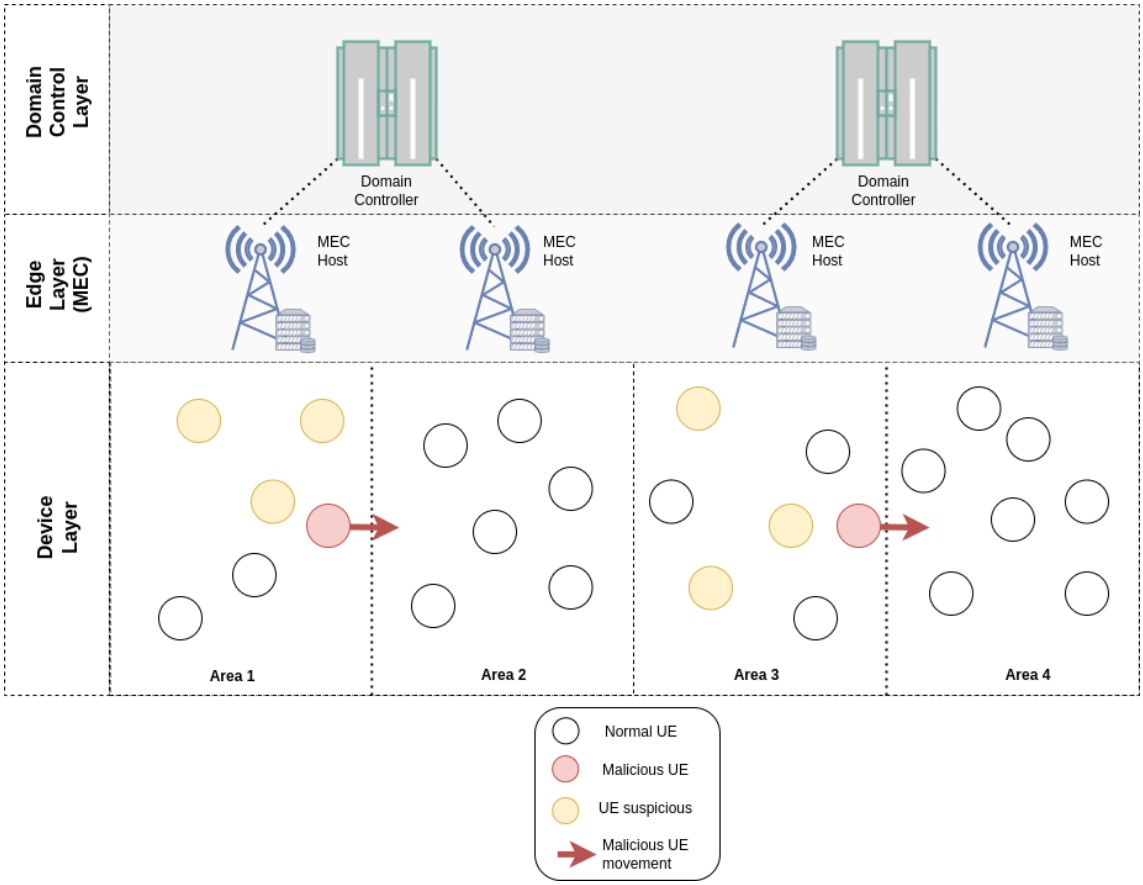


Figure 1.1: Problem Outline

Enabling multi-domain trust control information distribution in the network edge can potentially enhance trust assessment and leverage trust to the network edge layer, maintaining reasonable latency and availability levels. However, in multi-domain communication we need to address privacy/security concerns regarding data-leakage. In the occurrence of a domain-specific control information data-leakage (topology, traffic, user identifiers, etc), the attacker can obtain enough data to analyze network, track/steal

user data and also denial domain's internet service [65].

To address the security concerns and deploy the distributed framework using a multi-domain collaborative MEC scheme we can rely on blockchain as a source of privacy and security for inter-domain communication [65]. Generally the existing blockchain-based trust assessment and distribution approaches do not consider the constraints of standard blockchain architecture regarding latency and overall scale in distinct mobility scenarios. Although blockchain can ensure trustworthiness and data immutability, the storage burden of a single blockchain in a large network may become unmanageable after some period.

With a blockchain architecture we can benefit from the network hierarchical nature and interconnect each domain-specific MEH with intra-domain chains, controlled by a main chain at the upper level, within the Domain Control Layer. Hierarchical blockchain has the capability to accommodate data from control plane and promote on-chain scalability to achieve cross-chain edge data sharing [4].

Leveraging the defined scenario as a foundation, we present a framework crafted to tackle security issues in Mobile Edge Computing. This framework has been architected to combat a spectrum of security threats, including **content poisoning** and **cache pollution**. By doing so, it proactively fortifies MEC ecosystems against these threats, ensuring the integrity, confidentiality, and availability of data and services within the dynamic D2D communication paradigm. By employing cryptography, we establish a shield that envelops both the data and control planes, thus erecting defenses against attacks such as **man-in-the-middle** and **content spoofing** that may attempt to exploit vulnerabilities inherent in the caching processes.

This work is subdivided in two main steps. The first step encompasses the building Secure D2D caching based on Trust Management (SecDUB), an architecture which the main objective is to establish and evaluate a secure D2D caching framework based on collaborative trust management and blockchain, responsible to generate the device layer trustworthiness data that will be distributed across different MEHs and domains using the architecture of the second step. The second step encompasses the extension of this framework to an edge layer flexible multi-domain architecture, capable of raise the level of multi-domain network security considering security and privacy constraints. For each step we did a different systematic review focusing on what was conceptually important for each step. The methodological structure applied during both systematic review followed the recommendations of [31], a study which establishes a sequence of necessary steps for the development of reliable, consistent, auditable and reproducible systematic reviews.

## 1.1 Objective

The overall objective of this work is to establish and evaluate a secure multi-domain D2D caching framework based on collaborative trust management and blockchain. With the combination of trust management and blockchain technology, we seek to raise the level of network security and achieve the following objectives:

1. Reduce the distribution of malicious content in a D2D and Collaborative MEC scheme based on multi-domain architecture.
2. Increase the amount of useful data offloading, where caching of potentially malicious content can be avoided.
3. Enhance trust assessment in mobile environments.
4. Evaluate the capacity of the scheme to increase overhead as well as making a trade-off between offloading and security.

## 1.2 Contributions

We can summarize the contribution of this work into following aspects:

1. Combine direct and indirect trust for assessing of D2D nodes in video content caching.
2. Design a new lightweight intra-domain blockchain-inspired security framework to coordinate and audit evidence of indirect behavior in a secure way
3. New clustering approach that logically organizes the D2D nodes to support the lightweight blockchain-inspired framework, based on proximity, trust assessment and influence (online and offline social metrics).
4. Secure mechanism to distribute trust control information through different network domains in edge.

CHAPTER 2

# Conceptual Background

The aim of this chapter is to describe the main concepts of this our proposal. First, we set out the scenario of caching based on D2D communication as well as the adopted D2D video caching approach, and then we describe the adopted trust scheme and the blockchain-derived concepts derived.

## 2.1 Security Issues in D2D Video Caching

The D2D nodes in a video caching scheme can play two roles, i.e. that of the sender and receiver of a video. One D2D node can be a sender and a receiver at the same time. We introduce two different classifications for a D2D user/sender to show a security issue involving video caching in D2D communication networks: (1) the D2D Sender and (2) the malicious D2D Sender. User (1) is considered to be an ordinary D2D user, that receives and shares videos normally. On the other hand, user (2) deliberately shares invalid or malicious videos, and thus impairs the caching mechanism and reduces the system capacity for providing a useful flow of information [61].

It is important to stress out that malicious users may harm the caching mechanism by distributing invalid content in this scenario; thus the overall objective of our framework is to avoid communication between normal with malicious D2D nodes.

Probabilistic Social Cascade For caching in D2D communication (ProSoCaD) [17] is adopted as D2D video caching approach, since ProSoCaD coordinates content sharing and D2D caching. This approach takes into consideration online social relationships (e.g. facebook) to select the most influential nodes, which is also an evidence of trustworthiness. It is worth pointing out that our framework can be adapted to other D2D video caching approaches that take into account distinct node selection criteria or aspects, such as contextual [33] (content of interest or node behavior) and network performance [34] (link quality or transmission rate).

The ProSoCaD caching scheme is performed periodically. It combines the popularity of video content, online social relationships and offline social interactions (contact time and number of adjacent neighbors) to select the best located cache receiver devices,

which will proactively receive video caching via D2D communication or via LTE communication from the BS. In other words, the most influential devices are selected, based on the greatest and stable number of D2D connections (number of neighbors and contact time) and the greatest number of online social relationships.

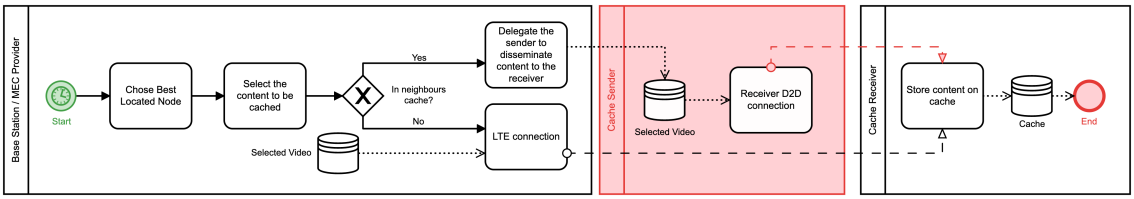


Figure 2.1: ProSoCaD caching

As it can be seen in Figure 2.1, the MEC Provider performs the selection of the best located cache receiver for a certain content, and search for a cache sender in the neighborhood of the selected receiver. If there is no neighbor D2D node (sender) which can send the selected content, the MEC Provider will send the selected content to the receiver node.

When analyzing the ProSoCaD system [17] during the caching process, the cache receiver takes no part in that decision, and thus, it vulnerable to malicious users, because there is no guarantee that the cache sender is trustworthy.

## 2.2 Collaborative Trust

As defined earlier, the user's trustworthiness is defined as Degree of Trust. In this context, a question arises: how can a security mechanism be designed that estimates the degree of trust of the nodes? We adopted an approach that involved using five trust properties in networks [12, 11, 64] as follows:

- **Dynamics** (the trust of a node must change in accordance with the pattern of behavior of your network).
- **Subjectivity** (different nodes may have different trust values for the same node under observation, that is, they depend on the experience of each node with the observed node).
- **No Transitivity** (if $A$ trusts $B$, and $B$ trusts $C$, $A$ should not necessarily trust $C$).
- **Context Dependency** (the trust assessment depends on the analysis context).
- **Asymmetry** (If $A$ trusts in $B$, it does not imply that $B$ trusts $A$).

The following features were taken into account to obtain the described trust properties: (R1) A decision-making procedure to determine an entity's degree of trust

should be distributed on the basis of cooperative assessments and uncertain/incomplete evidence. (R2) Trust must be determined in a highly dynamic and secure manner, by avoiding attacks and not harming the quality of effective communication, while capturing the components that demonstrate the aggregation of trust. (R3) A trust-based decision should not assume that all nodes are cooperative. (R4) Trust must be established in a self-organized and secure manner, so as not to be disturbed by the dynamics and maliciousness of relationships in D2D networks. This collaboration refers to the concept of trust attributed to the reputation of a single node in the network. (R5) Trust management must strike a balance between the problems of security and performance, in a D2D communication environment where resources are restricted, but security vulnerabilities are apparent.

The trust-based collaborative assessment must gather together all these features. Thus, trust can be assessed by combining two types of evaluation: trust through direct and trust through indirect observations. Let $T^D_{A,B}$ be the trust for direct observations and $T^I_{A,B}$ the trust for indirect observations of an user $A$ over an observed user $B$ [57, 25]. We calculate the degree of trust $T_{A,B}$ as follows:

$$T_{A,B} = \omega T^D_{A,B} + (1 - \omega) T^I_{A,B} \tag{2-1}$$

where $\omega$ is an associated weight to $T^D$, whose value ranges between $0 \leq \omega \leq 1$. We define a threshold of trust $\kappa$ for an observer node A that considers an observed node B to be trustworthy, therefore B is trustworthy if $T_{A,B} \geq \kappa$, where $0 \leq \kappa \leq 1$.

## Direct Trust

Trust by direct observation is an estimated degree of reliability that a user has that is greater than another based on behavior; thus direct trust can be estimated through observations of a node within its D2D neighborhood when interacting directly with them. Through multiple digital evidence collection, the observing user can evaluate the trust value by exploiting Bayesian inference, which is a method of statistical inference using Bayes' theorem to update the probability for a hypothesis when more evidence becomes available. Through Bayesian Inference the system is capable of producing a probability estimation of an unknown random variable [57, 25, 3, 43]. In our approach, the random variable represents the level of trust and is assumed to follow the $\beta$ distribution [37].

The $\beta$ distribution is characterized by the use of two parameters $\alpha$ and $\beta$, used to represent the quantification of normal and inappropriate behaviors, respectively. Let user $A$ evaluates user $B$, $T^D_{A,B}$ the trust value by direct observations of $A$ over $B$ and $\rho$ the probability of $B$ act maliciously. The trust value $T^D_{A,B}$ in the interval $[0, 1]$, is the estimate

of a random variable $\theta = 1 - \rho$. Since it is assumed that it will obey a $\beta$ distribution, the trust value can be represented with the mathematical expectation of the $\beta$ distribution.

$$T^{D}_{A,B} = E[\theta] = \frac{\alpha_{A,B}}{\alpha_{A,B} + \beta_{A,B}} \tag{2-2}$$

We establish that $\alpha_{A,B}$ and $\beta_{A,B}$ correspond to the collected values of evidence of good and bad behavior respectively, where initially $\alpha_{A,B} = \beta_{A,B} = 1$ thus $T^{D}_{A,B} = E[\theta] = 0.5$.

We assume that in a system of content caching, the direct trust of the sender is updated by the receiver with each shared video in a successive way through $\beta$ distribution, where trustworthy users transmit valid and truth video. By means of this approach, users seek to identify malicious users through the degree of trust, in order to avoid contact and hence prevent the distribution of false and/or malicious content. Thus, we employ two types of evidence as a parameter for assessing direct trust: (E1) the amount of invalid content transmitted and (E2) the amount of valid content transmitted, whereas E1 is evidence of malicious behavior, E2 is evidence of good behavior.

However, if only direct observations are relied on, this ignores the R1 rule outlined by [12], which refers to trust management cooperatively. For this reason, user $A$ must also take into account the historical of other users about user $B$ to calculate the trust level of user $B$, i.e. indirect observations. As a result of this, we achieve a more dynamic and, consequently less biased assessment.

## Indirect Trust

Trust by indirect observations is the estimated degree of trustworthiness based on the collection of recommendations about behavior from other nodes. Indirect observations can mitigate scenarios where an observed node acts normally for some nodes but maliciously for others so that it can confuse the trust mechanism and erroneously raise its trust value. By means of indirect observations, users are able to collect the observations of other users about a particular node, and then combine this evidence to make an indirect trust value decision. However, not all users are trustworthy, so the evidence/observation they provide may be erroneous or malicious. To deal with this, Theory of Dempster-Shafer (TDS) is a generalization of the classical probability theory, where an item of observation can be associated with more than one event, which makes it possible to represent uncertainty with greater precision, since it does not require an assumption to be made about events [48]. The versatility of the TDS, combined with the possibility of a probabilistic combination of evidence, offers a reliable resource for trust assessment based on reputation, that is, based on indirect observations.

TDS is an effective way to handle the problem of uncertainty by combining independent observations from distinct nodes. The ideas underlying TDS are derived from

two concepts: degrees of beliefs about an indirect observation obtained from different intermediate observer nodes, and the way these degrees of beliefs can be combined. The methodology put forward by [57] employs TDS, in which the calculation of degree of trust is mostly based on the direct trust value of the observer on the intermediate observer and the indirect observations sent by the intermediate observer. The set of hypotheses in TDS is named the problem domain, or Discernment Frame, represented by $\Omega$. Taking as an example an universe of two hypotheses $H$ and $\bar{H}$, the picture of discernment would be: $\Omega = \{H, \bar{H}\}$. The set of all possible combinations of $\Omega$ is called as Power Set and it is represented by $2^{\Omega}$, the elements of $2^{\Omega}$ are also called focal elements.

For our model, the received evidence follows to two hypotheses: $H = \{1\}$ and $\bar{H} = \{0\}$, where 0 and 1 corresponds to untrustworthy and trustworthy respectively. Consequently, the power set is given by $2^{\Omega} = \{\emptyset, H, \bar{H}, U\}$, where $U = \Omega$ is called as the Universe Set and represents the entire frame of discernment, that is, it represents both trust and non-trust. The mass, or Basic Probability Assignment (BPA) is a function, or measure associated with $2^{\Omega}$.

Each evidence (focal element) has a mass $m(s)$ associated with a hypothesis $s \subseteq S$. Bearing in mind that an element of the set $C_v$ is observation of indirect behavior, we consider that the associated mass with a hypothesis corresponds to the degree of direct trust of the node regarding the evidence. The belief value in a $S \subseteq \omega$ hypothesis is reached by sum the masses $m(s)$ according to TDS. So we call $Bel(S)$ as the belief value associated with each hypothesis and describe it mathematically according to the Equation 2-3.

$$Bel(S) = \sum_{s \subseteq S} m(s) \tag{2-3}$$

It is important noting that the $Bel(S)$ belief value under the $S$ subset does not imply that the belief under its complement $\bar{S}$, is $Bel(\bar{S}) = 1 - Bel(S)$, this is the greatest difference between Dempster Shafer's Theory and the standard probability theory.

The indirect trust value is calculated in accordance with the system of the TDS, which is a mathematical combination of the degrees of belief from each intermediate observer. Hence, if this approach is adopted, a less trustworthy D2D node will have less influence on the indirect trust calculation of other nodes. Figure 2.2 illustrates the scenario in which the TDS is applied.

Figure 2.2 shows $A$, $n_1$, $n_2$, $n_3$ that are observing nodes and $B$ the node under evaluation. Let $T_{n1,B} = 0.35$, $T_{n2,B} = 0.75$ and $T_{n3,B} = 0.87$ be the Degree of Trust of node $B$ for nodes $n_1$, $n_2$ and $n_3$ respectively and suppose that $T_{n1,B} \leq \kappa$, $T_{n2,B}$ and $T_{n3,B} \geq \kappa$, where $\kappa = 0.5$, therefore $n_1$ doesn't trust $B$ while $n_2$ and $n_3$ trust. In the context of D2D networks, user $A$ has no certainty about $n_1$, $n_2$ or $n_3$ trustworthiness, so it is therefore not advantageous to place full trust in the observations sent by them. Thus upon receiving

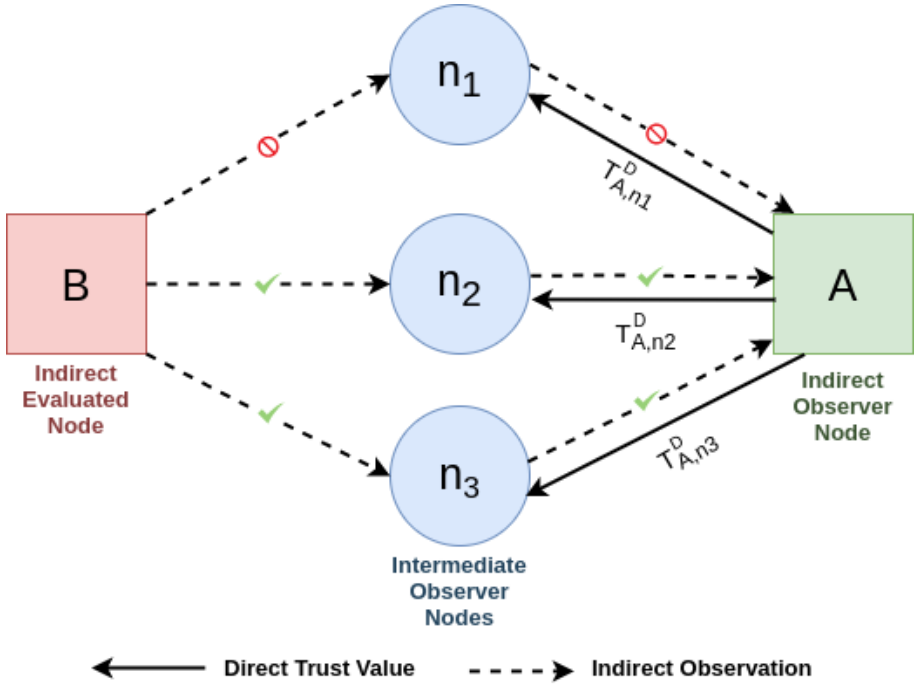


Figure 2.2: Indirect Trust Scenario

conflicting observations from different sources, with user $A$ uses the direct trust value of $n_2$ and $n_3$ as the probability value assigned to the indirect trustworthiness of $B$ [57]. However, in the context inherent to D2D networks, the collection of observations from neighboring nodes is subject to interceptions and falsifications when security approaches are not used, a factor that deteriorates the trust management mechanism and undermines the efficiency of the system to which it is applied. To have more knowledge regarding this theory and how it is applied on our proposal, please read the Appendix A.

## 2.3 Blockchain

Blockchain is a technology that enables secure and distributed management of an immutable record of transactions, called ledger. The ledger comprises a chain of blocks arranged in chronological order, which is stored in a distributed manner, through the management of a consensus protocol. The purpose of this protocol is to maintain the integrity of the transactions, by means of encryption, authentication and distributed consensus algorithms [24].

Transactions represent an agreement between two nodes, which may involve the transfer of data or completion of a task. Once the transaction is created, a participant signs and disseminates it between the nodes. The nodes are responsible for determining whether the transactions are valid or not. This means the nodes must reach an agreement about whether the transactions are valid or not, to ensure that there will be no diver-

gences. There are different consensus mechanisms for reaching agreement, depending on the type of blockchain. The most well-known is the Proof-of-work (PoW), which requires solving a computationally complex mathematical problem that will try to find an arbitrary number (i.e nonce) to vary the input and obtain a hash value. Other consensus approaches including the Pratical Byzantine Fault Tolerance (PBFT) algorithm, that seeks to reach consensus through the exchange of messages is characterized by the increasing completion time depending on the number of nodes involved [52]. Although blockchain protocols are well-defined, the domain in which they are applied has a great influence on different blockchain implementations. For this reason, blockchain governance deals with how nodes come together to maintain the blockchain inputs.

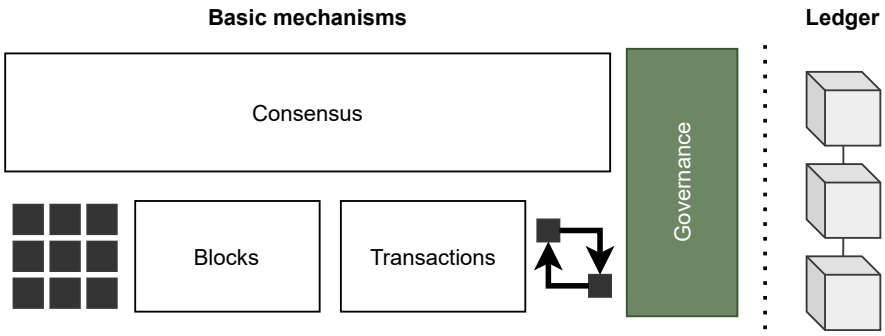


Figure 2.3: Blockchain high level architecture

The blockchain introduced by [40], encompasses all the nodes in the same network (i.e. thousands of nodes) and this was later categorized as a public blockchain, in which the particular features involve the participation of several nodes in maintaining the same ledger. However, we established a permissioned blockchain-inspired design to deal with the dynamics of D2D and edge communication interactions. For D2D scenario, we restrict the number of nodes per ledger through dynamic clustering, since the ledger is only read and written by the cluster nodes. An UE node is included to a cluster according to the clustering criteria of our proposed scheme and whether it is authorized by the cluster head. In our approach, blockchain is used to store the history of indirect observations sent by nodes of a cluster in a certain fraction of time. With this approach we enable indirect observation messages to be distributed and still persisted securely in the ledger.

CHAPTER 3

# Secure D2D caching in Multi-Domain Edge based on Trust Management and Blockchain

This dissertation encompasses the designing of a multi-layer framework that combines diverse technologies inspired in hierarchical blockchain to come up with a secure multi domain D2D caching framework. This chapter seeks to present the concepts and describe the architecture following the design of two different modules: Secure D2D caching based on Trust Management (SecDUB) (Section 3.1) and Trust in Multiple Domains (TrustMD), a multi domain distributed trust framework (Section 3.2).

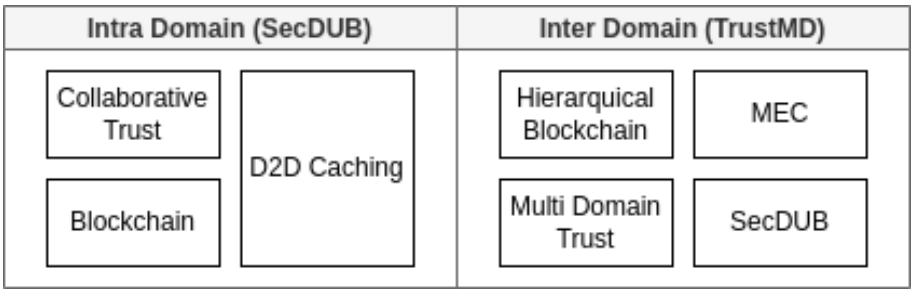


Figure 3.1: Overall Architecture [1]

Figure 3.1 briefly describes the proposed overall architecture. SecDUB builds an intra domain secure caching framework in which devices interact through D2D connection in order to collect trust evidences and then shape the trustworthiness of a neighbor node. TrustMD presents an inter domain framework that interacts with SecDUB to distribute trust information in edge by different domains in securely manner.

[1] https://github.com/DJAcquila/tmd

## 3.1 Secure D2D caching based on Trust Management (SecDUB)

This section describes the proposed D2D caching framework, which combines three models: (1) D2D Video Caching, (2) collaborative trust management and (3) security. Through this scheme, our aim is to enable UEs to maintain a minimal or non-existent contact with untrusted or malicious users, and thus increase the overall degree of security and mitigate the spread of malicious content.

We seek to meet the trust requirements in distributed networks, by adopting an approach which ensures that each node is responsible for assessing the trust of the nodes to which it interacts, on the basis of direct and indirect evaluations. In our approach, direct trust is calculated in terms of the experience between nodes during caching and video sharing and the indirect trust values are updated on the basis of the opinions of other nodes. Security mechanisms devised by pre-existing blockchain models are deployed to allow distributed and secure sharing of indirect trust information, and also, to provide historical evidence of indirect behavioral patterns.

The use of blockchain-inspired security mechanisms in trust management is one of the main research contributions of this work. An immutable ledger of video sharing ensures the distribution of secure trust information, through the use of encryption, authentication and a distributed consensus, as it provides a distributed ledger among the nodes of the blockchain network securely. Thus, the combination of these concepts in blockchain makes it possible to disseminate information without the need for a centralized trusted management entity (i.e., MEC Provider or BS). However, maintaining a distributed ledger between mobile devices requires clustering strategies to coordinate the group of nodes that are responsible for that ledger.

This section is subdivided as follows: Section 3.1.1 concentrate on SecDUB related works and compare them with the proposed solution utilizing a specific categorization; Section 3.1.2 desiccates SecDUB's architecture highlighting the most important points and characteristics; Section 3.1.4 describes the security management model; Section 3.1.5 highlights the clustering scheme and topological architecture of the proposed D2D secure caching scheme; Section 3.1.6 refers to the description of the agreement model used by the blockchain-inspired scheme.

### 3.1.1 Related Work

[25] put forward a method to improve efficiency and security in the context of content delivery based on D2D communications. A user's trust is represented by a real value between 0 and 1, and calculated through direct and indirect observations. The

authors decided to use Bayesian Inference and the Dempster Shafer Theory, to assess trust by direct and indirect observations respectively. A deep Q-learning model was designed to tackle the problem of secure caching in D2D, which was used to evaluate different metrics such as receiving capacity, signal quality and the trust value of the transmitter. It was also employed to make decisions such as: choosing the best sender and determining whether or not the provider should cache the new content. However, the authors ignored the fact that indirect behavior observation shared among mobile devices, through an insecure communication channel, can be intercepted and falsified, owing to the lack of a suitable security mechanism.

[39] designed a trust model based on direct and indirect observations that helps to detect malicious service providers, by highlighting the importance of collaborative trust assessment in the D2D communications environment. The authors also examined the impact of trust-distortion attacks and built a trust assessment architecture capable of resisting these attacks, by handling contradictory recommendations sent by dishonest recommenders. The devised mechanism evaluates the trustworthiness of the nodes by means of different evaluation intervals and keeping track of any marginal or mistaken misbehavior.

The falsification and interception of indirect observations leads to the degradation of trust values, which, as a result, impairs the D2D communication. However, it was clear that both [25] and [39] ignored the security features of the indirect observations that were shared in an insecure communication channel. It should also be noted that trust-based communications are subject to a series of attacks, which can impair performance in decision-making [8] and the results obtained by these authors do not provide evidence of the effects of these attacks on the assessment of trust performance.

It is worth highlighting the following works on blockchain for D2D: [69] and [35]. The authors seek to employ blockchain-based concepts in the context of D2D communication to address security vulnerabilities and compensate for the lack of a trusted third party managing entity, while addressing requirements such as offloading and the optimization of D2D resource sharing.

[69] developed a model, which employs blockchain as an incentive mechanism for caching in D2D. The blockchain transactions refer to the content share among two devices. The PBFT consensus protocol was employed in order to reach consensus under transactions in a faster and optimized way. The authors also considered the use of a Deep Reinforcement Learning algorithm to optimize the selection of nodes responsible for consensus and allocation of caching resources, in order to decrease latency and increase system scalability. However, the authors ignored node's trustworthiness and its impact on D2D caching.

[35], designed a blockchain-based consensus protocol in a mobile edge network

D2D environment, where the blockchain acts as a trusted third party to maintain transactions between users while exchanging content. Each transaction represents a different communication process between two nodes and is broadcasted to the blockchain nodes in a secure manner whenever a new transaction is created. In a predefined time period, the BS will collect some transactions and elect the most trustworthy node to be the consensus leader. The trust of the nodes is assessed by transforming the problem into a Markov decision-making process, which can be solved by means of reinforcement learning. Despite using a trust assessment model, the approach is centralized in BS and does not employ a collaborative trust mechanism, which does not allow nodes to assess their neighbors' trust in a distributed and decentralized way.

Table 3.1: Comparison of SecDUB Related Works

| | **Secure Caching** | **Blockchain based approach** | **Collaborative Trust** |
|---|---|---|---|
| [25] | X | | X |
| [39] | | | X |
| [69] | X | X | |
| [35] | | X | |
| **Proposed** | X | X | X |

Table 3.1 compares the related works discussed in this section in terms of the key features employed (Secure Caching, Blockchain-Based Technologies and collaborative Trust). These works underline the importance of a trustworthy and secure communication in the context of D2D. However, some of the approaches adopted failed to take into account the importance of collaborative trust and none of them addressed the question of the vulnerabilities of indirect trust observations that are shared with neighboring devices, which can impair the assessment of trust performance. In addressing this factor, we seek to make an improvement in collaborative trust assessment, by ensuring a secure distribution of observations by means of blockchain technology. We also seek to evaluate the trust mechanism in the presence of attacks that are designed to cause its malfunction [8] by investigating the effects of this kind of malicious behavior on performance.

### 3.1.2 Architecture

Figure 3.2 represents the planned architecture, and highlights the main actors and components of the scheme and their roles in the secure cache framework, as well as showing how this scheme differs from ProSoCaD.

While operating the caching, the receiver will only deal with content received from a trustworthy neighboring UE. Cache Sender is responsible for sending the video to its most influential receiver. Cache Receiver will only accept the video If the sender is trustworthy and acts in accordance with the trust management strategy. If the sender

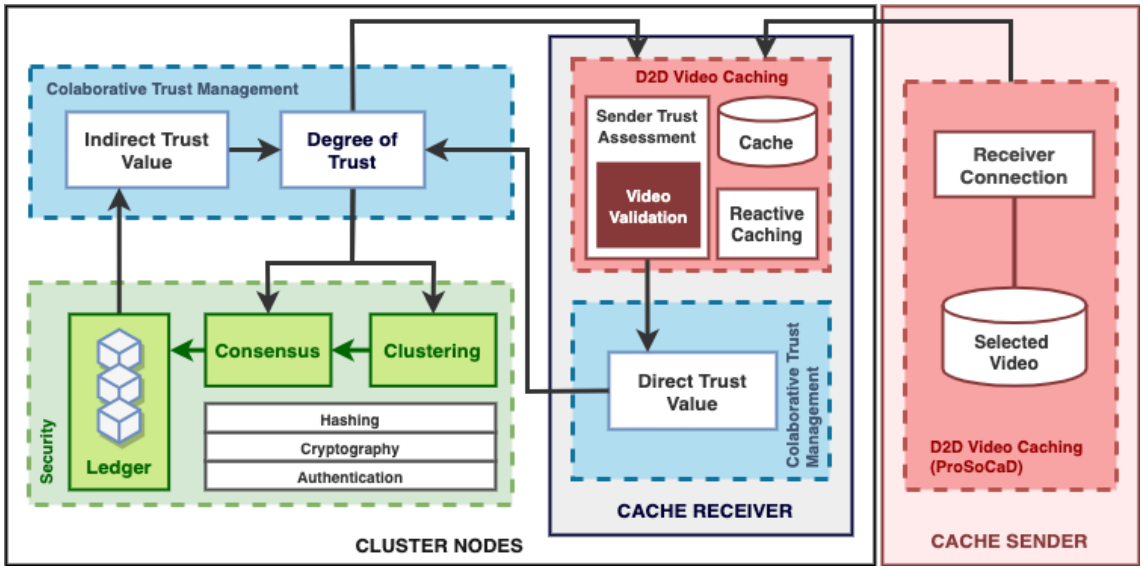


Figure 3.2: The SeCDuB Architecture

is found to be trustworthy, upon receiving the video, the receiver checks the validity of the video, which defines the evidence of the kind of behavior required for direct trust assessment. A valid video provides evidence of the sender´s good behavior, while an invalid video provides evidence of bad behavior. When a node updates the direct or indirect trust values of the other nodes, the degree of trust of the observed node also changes, which leads to influencing clustering, caching and consensus mechanisms.

It is mandatory for the Cache Receiver to be a Cluster Node that must report the sending of a video as a transaction to Cluster Head (CH) which is responsible for starting the consensus process. However, it is not necessary for a Cache Sender to be a Cluster Node. During the consensus, the remainder of the Cluster Nodes are responsible for validating transactions in a distributed manner and sharing indirect behavioral evidence among the cluster nodes, which means that at the end of the consensus, the created block has a record of good or bad behavior, in accordance with each sender´s cluster node trust value. In this scheme, the indirect trust evidence is shared in a secure communication channel and recorded in the distributed ledger by means of a verification and validation mechanism. Hence, when receiving a copy of the ledger, an old/new member of the cluster can update/calculate the indirect trust value from the historic of indirect trust provided by the ledger. It is worth noting that our framework enables different strategies to be adopted a) to select the most influential/central D2D cache nodes, b) to calculate the indirect and direct trustworthiness value, c) to form a consensus and d) to create a ledger. In the next subsections we highlight the main contributions related to the security module.

### 3.1.3 Sender Trust Assessment

The most of the caching schemes, including ProSoCaD, are capable of selecting the most central or influential devices, which will proactively receive video via D2D communication, but do not guarantee that the cache provider is trustworthy or offer a quality guarantee of the transmitted content. This is even more problematic in out-of-coverage D2D scenarios, where the BS is unable to manage the communication. For this reason, it is essential for users to have the ability to self-manage the communication so that they can protect their privacy and avoid the problem of having unwanted storage of invalid and potentially malicious content in their devices. We addressed this potential weakness, by adjusting the caching system, to ensure that content is only sent if the transmitter trusts the receiver. Through a trustworthy reactive caching, the receiver node can restrict its searches to trusted neighbors that can provide the content. We adopted the scheme described in Figure 3.3.

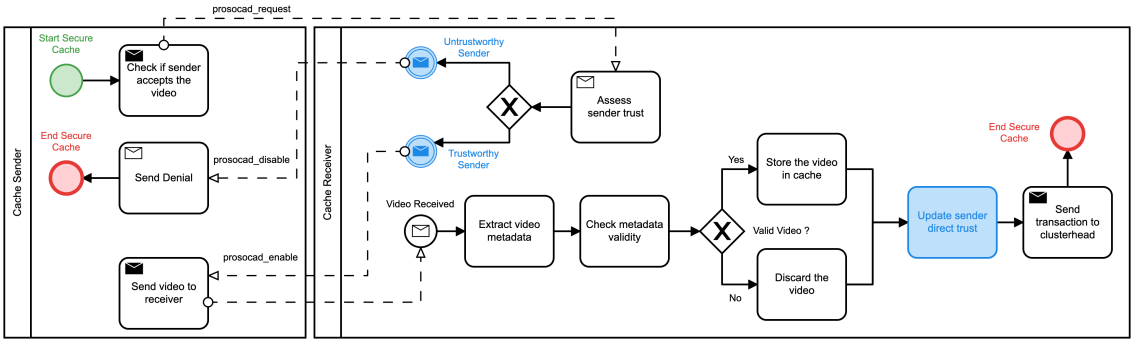


Figure 3.3: Secure Cache Scheme

Let $c$ be the content selected to be sent by $A$ to a receiver $B$ on its neighborhood. Before actually sending the video, $A$ shares a $PROSOCAD\_REQUEST$ message with $B$, meaning that it has a video selected to be shared through the ProSoCaD caching system. Upon receiving the message, $B$ can decide whether or not to trust $A$, by assessing the node trustworthiness through the degree of trust (Eq. 2-1).

If $A$ is trustworthy, $B$ sends a confirmation message $PROSOCAD\_ENABLE$, meaning that the content can be sent. On the other hand, if $B$ does not trust $A$, a message $PROSOCAD\_DISABLE$ is sended and the video sharing is avoided.

In a scenario where $B$ trusts $A$, upon receiving the video, $B$ checks whether the video is valid. A valid video provides an observation of $A$'s good behavior, while an invalid video provides an observation of bad behavior. In this way, if the video is found to be invalid, this will reduce the direct trust value of sender $B$ according to $\beta$ distribution. In contrast, If the video is found to be valid, the direct trust value of $B$ will increase according to $\beta$ distribution. We employ a metadata analysis algorithm to check the validity

of the video. Metadata can be regarded as "data about data", that is, it is information extracted from the video file itself, which allows it to be identified or classified [10]. We use information that can be extracted from the video itself to mitigate the risk of any falsification.

It should be noted that the metadata considered for our scheme are: size, data type, duration, resolution of streaming video, number of audio streams, number of video streams, total size of the audio streams and total size of the video streams. For the sake of simulation, we considered each of these fields 4 bytes long, totaling 32 bytes for metadata. All the files have a hexadecimal header that corresponds to the data type of the file signature, which means that all the previously known file types have a standard signature. The receiver must extract the information directly from the video to check its validity. For example, a bad metadata is one that identifies a file that does not correspond to a video. For example, if a video is identified with the header signature `0x25504446`, it is not a valid video, because this header identifies a PDF file. It should be stressed that the content will be considered to be invalid if any of these metadata are not included.

### 3.1.4 Security Management Model

The planned security management model combines concepts adapted from blockchain technology to provide a secure means of communication for the dissemination of the indirect trust control information. A trustworthy and immutable ledger for video sharing assists in the distribution of the secure trust information, through the use of encryption, hashing, authentication and distributed consensus, as it means the ledger is shared among the nodes of the blockchain network in a secure way. Thus, the combination of these concepts in blockchain makes it possible to disseminate control information without the need for a centralized trusted management entity. [40] sets up a network where the nodes have more time and energy available to operate in accordance with a highly complex consensus protocol through the traditional blockchain approach.

D2D nodes do not have high energy availability and their interactions dynamically, which makes it very hard to apply the traditional blockchain protocols. To overcome this, we recommend making adjustments to the blockchain concepts (e.g. ledger and consensus) to speed up the decision-making process and reduce the processing and network overhead, while improving security and providing trust data auditability. The adjustments needed to enable this integration are briefly summarized in the following paragraphs.

The computational costs of the consensus protocol should be lower so that there can be a reduction in the time required for its completion. This involves enabling its integration in mobile environments, where the interaction between the D2D devices have certain dynamism on account of the mobility. In traditional blockchain, during each

round of consensus, several transactions are validated and added to a block. However, the waiting time for adding a group of transactions and validating all of them can lead to loss of contact between D2D nodes. Each transaction of video sharing should be instantly validated so that the historic observation for trust degree assessment can be speedily updated. Hence, each transaction represents the transmission of a video between a transmitter and a receiver via D2D.

Moreover we designed a distributed clustering scheme as a means of arranging the nodes responsible for managing the blockchain, and ensuring that each cluster is responsible for a different blockchain network. A network with a smaller number of nodes makes the consensus process more agile, while reducing the communication and processing overhead, which is also desirable for D2D networks with restricted resources. The next subsection describes all the parts of the blockchain model.

### 3.1.5 Clustering Scheme for the Blockchain Network

We designed a scheme for clustering, which seeks to adapt the blockchain network to the D2D caching environment, considering the limited storage and computation [42]. In this way, our clustering scheme incorporated three factors: a) the restricted resources of D2D devices, b) the selection of CHs based on the level of influence provided by the ProSoCaD caching approach and c) trustworthiness. In light of this, we created a clustering scheme that seeks to provide a trustworthy blockchain network aligned with a D2D video caching approach. In carrying out a distributed cluster management, the nodes must agree on the same CH by forming a consensus.

The CH is the node responsible for coordinating the initial formation of the group, storing the cluster management information, keeping the list of Cluster Member (CM) updated and overseeing the addition of new blocks. Nodes can be classified in three types of status during the cluster formation: (1) *stand_alone*, (2) *cluster_member* and (3) *cluster_head*. In (1) the nodes not linked to any cluster, but look for a CH to connect to. In (2) the nodes are active participants in a cluster, but do not play the role of a CH or provide indirect observations. In (3) the nodes coordinate the input and output of nodes in the clusters.

We divided the clustering protocol into two operation modes: (1) configuration and (2) maintenance. The configuration phase corresponds to the initial state of the nodes, meaning the time they operate in *stand_alone* mode.

During configuration stage, the nodes attempt to link with neighboring CHs as follows: if more than one nearby CH is identified, they must decide to join the CH candidate that is closest. If no CH is identified, but a stand-alone neighbor instead, the process of forming a new cluster begins, which consists of two stages: (a) Preparation and

**Algorithm 3.1:** Clustering Configuration

**Data:** $C$ - List of nearby CHs
**Result:** $ch$ - ID of the selected CH

$ch \leftarrow \varnothing$
**if** $C \neq \emptyset$ **then**
  $C_d \leftarrow \emptyset$
  **for** *each* $u \in C$ **do**
    $d_u \leftarrow u.distance$ ▷ Distance to node $u$
    $id_u \leftarrow u.id$ ▷ ID of node $u$
    **if** $id_u$ *is trustworthy* **then**
      $C_d \leftarrow \{d_u, id_u\} \cup C_d$
    **end**
  **end**
  $ch \leftarrow min(d_u \in C_d)$ ▷ Select CH with the minimum distance
**else**
  $SA \leftarrow \emptyset$ ▷ List of nearby *Stand Alone* nodes
  **while** $|\, SA \,| \leq 2$ **do**
    $SA \leftarrow searchNeighbors() \cup SA$
  **end**
  $ch \leftarrow maximize\{\sum_j (w_{ij})\}$ [17] ▷ Select the most influential node
**end**
**return** $ch$

(b) Election. The preparation phase corresponds to the coalescence of the adjacent nodes, until each node has at least three neighboring nodes that are linked. The PBFT protocol requires $3f+1$ nodes, where $f$ is the number o failure nodes, so 4 is the minimum number of nodes to support 1 failure node and reach agreement.

During the election phase, the most influential node among the candidates is elected CH by means of the centrality metric recommended by [17]. Hence, the configuration phase allows a faster CH selection or election by combining two metrics to choose the CH, i.e. the closest and more influential CH.

CHs send periodic messages to neighboring nodes to arrange the addition of new members, by informing them that it is a CH. Upon receiving one of these messages, nodes in *stand alone* request the CH to enter the cluster if the CH is trustworthy. Video D2D caching and CH selection are aligned to enable the clustering protocol to select the most influential nodes as CHs for the content-sharing network. Thus, there is a tendency for the most influential nodes to be CHs which means they are more susceptible to receiving content for caching in accordance with ProSoCaD. Moreover, this approach helps to select more trustworthy CHs, which corroborates the security features of the system (Algorithm 1).

The maintenance phase corresponds to the period of time after the CH is chosen,

**Algorithm 3.2:** Clustering Maintenance - Node Dismemberment

**Data:** $C$ - List of nearby CHs, $\varphi_{id}$ - Selected CH id
**Result:** Dismemberment decision

$u \leftarrow maximize\left\{\sum_j (w_{ij})\right\}$ [17] ▷ Select the most influential node
**if** *ch is not reachable* **then**
  **return** *true* ▷ Check CH connectivity
**end**
**if** *ch is trustworthy* **then**
  **return** *true* ▷ Check CH trustworthiness
**end**
**if** $C \neq \emptyset$ ***and*** $u.id \neq ch.id$ **then**
  **return** *true* ▷ Check CH influence
**end**
**return** *false*

and involves carrying out the tasks required for blockchain and cluster management such as distributed consensus and the dismemberment of CMs. There are three situations which cause a CM to be dismembered from its cluster: a) loss of connectivity, b) finding a more influential CH or if the CH becomes untrustworthy in terms of the trust value and c) the trust threshold $\kappa$.

When a CM $A$ leaves the cluster, it is removed from the group and stops participating in that blockchain network, at that moment, there is no need to store the ledger and therefore $A$ must exclude it. If $A$ finds a new cluster, it must request to the CH the ledger for the blockchain network operated in this group.

The next subsection deals with the consensus protocol, and how the blockchain-based network can be adapted to define the topology. Each cluster is responsible for maintaining a distinct blockchain network which persists until the group is disbanded.

### 3.1.6 Consensus Protocol for D2D Networks

D2D communication can be characterized as a system of direct and opportunistic interactions caused by the dynamic interactivity of the model, and consists of nodes with little energy capacity and computational power. It is notable that consensus algorithms based on excessive consumption of time and energy, such as Proof-of-work [40], do not adapt easily to D2D communication.

Proof-of-work requires a computationally-complex mathematical process to find a random hash value. Owing to the complexity and randomness, a brute force algorithm needs to be is used, which requires an enormous amount of energy and time, as well as not being feasible for dynamic and limited scenarios such as D2D networks. As a result, many resources are wasted after the end of the consensus because the network or group of nodes

no longer exists. The Pratical Byzantine Fault Tolerance has been used by Hyperledger Fabric [27], since the algorithm can handle $1/3$ of malicious nodes [7]. The PBFT seeks to reach consensus through an exchange of messages between nodes and the maintenance of consensus is characterized by the increasing time of completion depending on the number of nodes involved [52]. However, energy consumption and average completion time are much lower than PoW, which suggests that PBFT is more suitable for dynamic scenarios such as D2D. Furthermore, we have adapted the PBFT to the clustering scheme employed in this article and thus, the consensus process is still lighter and more decentralized.

In the context of permissioned blockchains, distributed consensus plays the role of coordinating the activities of a network to confirm the agreement of the cluster members about changes in the ledger. The ledger is used as proof of a user's participation, or in other words, users who store an inconsistent or false ledger are unable to participate in the consensus process. Before gaining access to the ledger, a user must be accepted as a member of the cluster that possesses it. We use the notation $\lambda$ to refer to the ledger, users must have the same $\lambda$ in order to participate in the consensus.

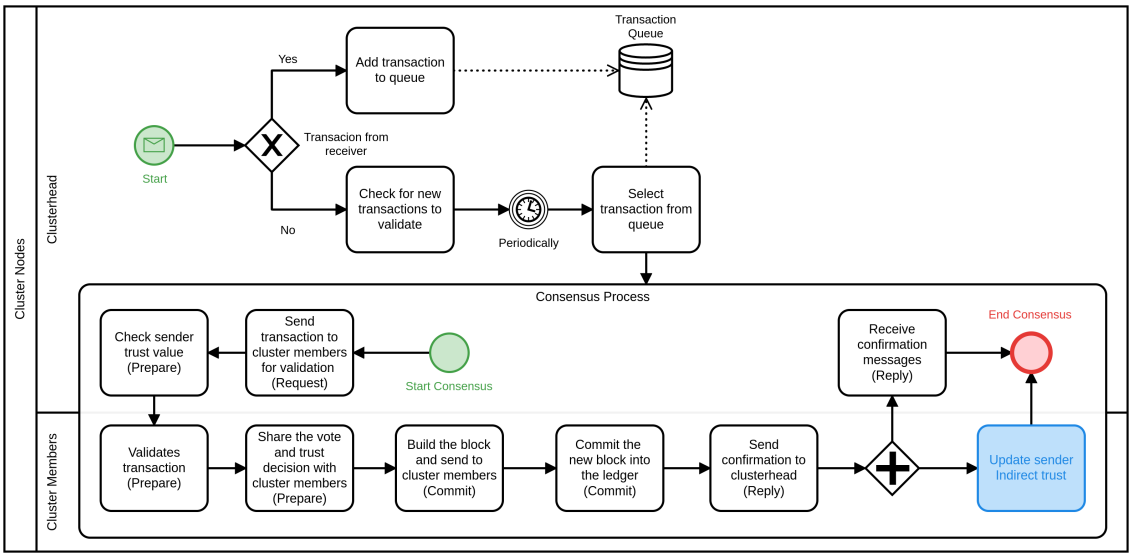


Figure 3.4: Consensus Strategy

Figure 3.4 describes the overall flow of the consensus strategy, from the receiving of a transaction to the indirect trust assessment. Sending a video results in the creation of a transaction which is subsequently validated through the consensus strategy. The protocol is operated in a distributed manner, and is divided into four stages: request, prepare, commit and reply. Unlike the classic PBFT, a new round of consensus is initiated by the CH, which is the cluster/consensus leader.

A new block is added whenever a transaction is triggered and a new round of consensus is required for its addition. A transaction corresponds to the sending

of content from the cache D2D sender to the cache D2D receiver and is generated by the UE upon receiving the content. The transactions have the following format: $t_x = (\tau, \mu_v, h_\mu, d_v, id_s, id_r)$, where $\tau$ is the 4-byte long transaction creation timestamp, $\mu_v$ represents the 32-byte long video metadata extracted by the receiver, $h_\mu$ is the 32-byte long metadata hash, $d_v \in \{0, 1\}$ is the 4-byte long verification of metadata validity, $id_s$ is the 4-byte long IP of the transmitting node of the video and $id_r$ is the 4-byte long IP of the receiving node, totaling 80 bytes for each transaction.

The $t_x$ transaction is sent by the receiver to the CH which is responsible for initiating the consensus. The CH maintain a list of transactions $Q_t$ to manage which transaction will be validated in each round of consensus, so that $Q_t = (t_x^1, t_x^2, ..., t_x^i)$, where $t_x^i$ is transaction $t_x$ at time $i$. When the consensus has started, we can divide it into four phases: (1) request, (2) prepare, (3) commit and (4) reply.

The ***request phase*** is the beginning of the consensus, when the CH starts the process of creating and checking the block and this involves notifying all the CMs through a message $m^{req}$. Let $m^{req}$ be defined as follows: $m^{req} = \{h(t_x), \tau, t_x\}$, where $h(t_x)$ is the transaction hash, to verify the integrity of the transaction, $\tau$ is the initialization timestamp of the consensus and $t_x$ the transaction that triggered the consensus process.

The ***prepare phase*** consists of the period after receiving $m^{req}$, when the CMs will share votes about the trustworthiness of the sender and verify the video metadata. This process is equivalent to the one processed by the receiver upon receiving a video. This transaction is validated by each of the CMs that check the video metadata. Let $n_a$ be a UE who received the message $m^{req}$ during initialization and $n_s$ the sender UE referred by IP $id_s$ in transaction $t_x \in m^{req}$. The message shared by $n_a$ during the preparation phase has the following format: $m_{n_a}^{pre} = \{h(t_x), t_x, d_{tx}, d_s\}$, where $t_x$ corresponds to the transaction under evaluation, $d_s$ the trustworthiness vote of $n_a$ where $d_s = 1$ if the node trusts the sender and $d_s = 0$, otherwise, and $d_{tx}$ the result of the metadata evaluation using the same methodology described in 3.1.3.

The ***commit phase*** corresponds to the step following the receipt of $m_{n_i}^{pre}$ messages from all $n_i$ users. Upon receiving the messages, each UE will have a set of votes $Cv_{n_s}$ that represents the opinion of each member of the cluster on the trust of $n_s$, where each $t_d \in Cv_{n_s}$ is a tuple $(id, d)$ corresponding to the node's IP and its vote about trust on the transmitter. This set is used for indirect trust assessment and therefore it is added to the newly created block header, so that it can be accessed at any time. The messages exchanged between nodes in the commit phase have the following format: $m_{n_a}^{com} = \{h(t_x), h(b_x), t_x\}$, where $h(b_x)$ is the hash of the new block, and $t_x$ the transaction in question. All nodes must generate the same $b_x$, to maintain the consistency of the added blocks and the ledger chronology. If a node finds that the generated hash by it is different from that generated by the majority, it must request the correct block to the CH. Finally,

CMs must send a confirmation to all the CMs, to show the current round of consensus has been completed.

### 3.1.7 Block Structure of the Ledger

Once the consensus is finished, a new block is created and added to the ledger. Each added block has the same structure as that described in Figure 3.5, which can be divided into two parts: (1) Header and (2) Data. A block header is used to identify a particular block on an entire ledger where each block is identified by a single header, and each of these blocks is identified by its block header hash individually. The block data field comprises the transactions and the data that has been properly stored in the block.

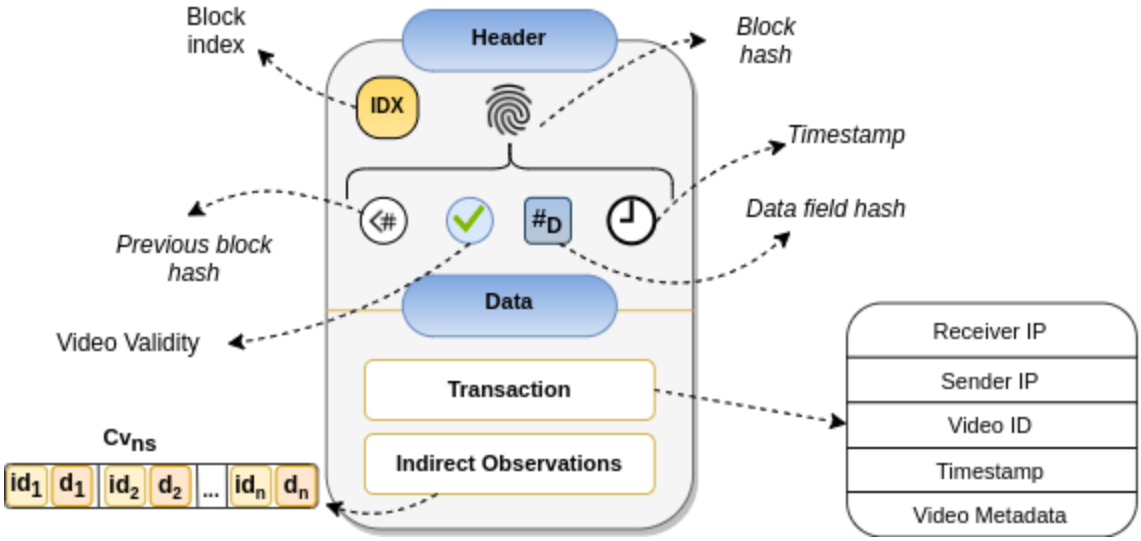


Figure 3.5: Block Structure

The block header contains a 4-byte long block Index number, 32-byte previous block hash, 4-byte long video validity decision, 32-byte long data field hash and a 4-byte long timestamp of the block, totaling 76 bytes. The primary identifier of each block is the cryptographic hash, which is basically a digital fingerprint created by hashing all the block headers.

The block data contains the transaction (80-bytes) related to the consensus round that originated the given block and the indirect set of observations $Cv_{n_s}$, that represents the opinions of each member of the cluster about the sender $n_s$ trustworthiness, where each $t_d \in Cv_{n_s}$ is a tuple $(id, d)$ corresponding to the node's IP and its vote about the sender trustworthiness, totaling (8-bytes) for each $t_d$. With the aid of the authentication and cryptography mechanisms, each CM can certify the origin of each indirect observation, in the knowledge that the information included in each block was not altered or falsified by any third party unrelated to the cluster, since all the messages exchanged between

any cluster member are encrypted. The use of a cryptography hash as a block identifier also guarantees the voter integrity, since any alteration in a block triggers changes in the following ledger blocks.

## 3.2 A multi domain edge distributed trust framework based on blockchain (TrustMD)

This section describes Trust in Multiple Domains proposal, a framework that extends SecDUB capabilities to the edge and domain layers. This approach combines edge trust storage with blockchain concepts and distributed storage management in a multi layer architecture, designed to efficiently store trust data in edge. Through this scheme, we seek to safely store trust information from nodes between distinct network edges and domains and thus provide trust information to a broader area. Also we seek to avoid high latency in edge, enabling secure MEC Layer collaborative communication. To enable TrustMD we propose to utilize a hierarchical blockchain approach, that works across Domain and MEC layers. Hierarchical blockchain has the capability to accommodate control information from control plane and promote on-chain scalability to achieve cross-chain edge data sharing [4].

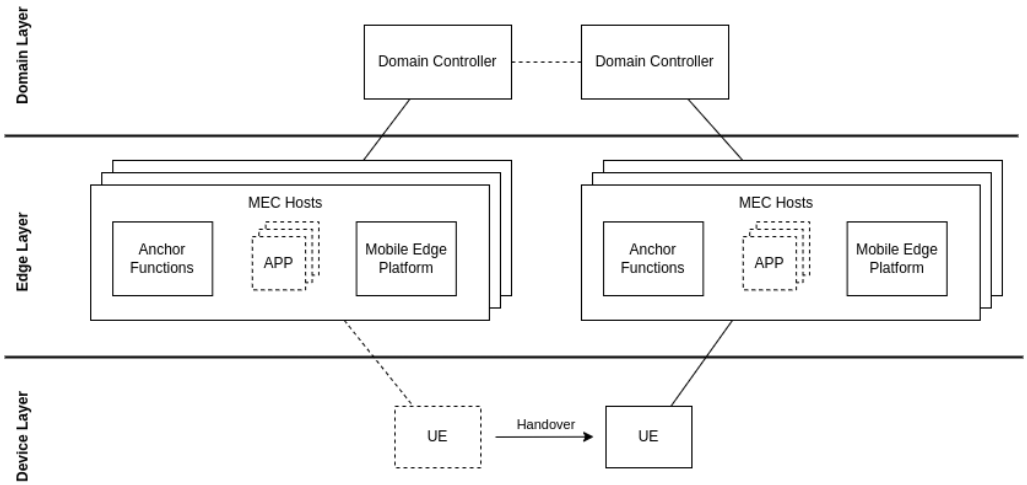


Figure 3.6: TrustMD scenario. Adapted from [20]

Figure 3.6 shows TrustMD scenario and the overview of the multi-layer network in the cross domain trustworthiness problem. Considering this design, the occurrence of an inter-domain handover of a malicious node will not harm the trustworthiness model, since the node's trustworthiness data is distributed in a multi-domain fashion across all the network. With TrustMD, UEs can have access to other domain's trust control data through the upper-level chain in Domain Layer.

This chapter is subdivided as follows: section 3.1.1 details TrustMD related works and compare them with the proposed solution utilizing a specific categorization. Section 3.2.2 desiccates TrustMD high level architecture highlighting the most important points and characteristics. Remaining sections 3.2.3, 3.2.4 and 3.2.5 encompass the main operations of TrustMD architecture: Edge Trust Update, Inter Domain Handover Trust Update and Trust Query respectively.

### 3.2.1 Related Work

This section covers the related work that helped create the scientific foundation of TrustMD and the problematic of multi-domain distributed storage. We subdivided our related works in three categories that summarizes core components of our inter-domain approach: (1) trust assessment; (2) edge caching; (3) blockchain. Subsection 3.2.1 encompasses those works in which the trust assessment was a core component of their approach. Subsection 3.2.1 contains the studies related secure content caching and its implications. Subsection 3.2.1 is composed of few studies that tried to incorporate a multi-domain architecture in the context of 5G and beyond edge computing.

#### Trust assessment

Authors of [32] proposed a new way to detect and blacklist insider attackers in Veicular Ad Hoc Networks (VANET). Their approach uses a blockchain-based trust score framework to track the behavior of participating nodes. If a node is found to be behaving suspiciously, it is added to a blacklist. This blacklist is dynamically updated based on the trust scores reported by neighboring nodes. Authors argue that trust-based solutions can help to detect selfish nodes that act as blackholes in the network. These nodes intentionally drop messages, which can disrupt routing algorithms and cause traffic congestion. The proposed framework is implemented in a flat architecture in the network edge. This means that all nodes have equal access to the blockchain, which makes it more decentralized. However, it also means that the blockchain can become overloaded if there are a large number of nodes in the network. Another limitation of the proposed framework is that it does not consider trust distribution among different network domains. This means that a node in one domain may not be able to trust a node in another domain, even if they have a high trust score. Overall, the proposed framework is a promising new approach to detect and blacklist insider attackers in VANET. However, it has some limitations such as blockchain scalability with increasing number of data and nodes, and trust distribution among different network domains.

In [70], authors combined blockchain and deep learning for the VANET network to investigate access control and computation offloading. They consider a general VANET

scenario where multiple vehicles can offload their tasks to an edge or cloud server for collaborative performance. To improve security, authors designed a hierarchical distributed software-defined VANET framework based on the blockchain. Although this is a fair design for offloading task, it is not well suited for the trust distribution context because: (1) there is only one blockchain to operate through the whole multi-domain architecture and consequently, the domain controllers are the only entities holding all blockchain data.

In [63], authors proposed *Social-Chain* a novel blockchain-based decentralized system for trust evaluation in Pervasive Social Networks. The blockchain store trust evidences and nodes use the data recorded in ledger as a source for trust assessment. The proposed system is fairly suitable for Pervasive Social Networks, but wasn't tested in scenarios with considerable mobility and different network domains.

**Edge caching in 5G networks**

In [60], authors argue that caching services are vulnerable to a variety of attacks, including man-in-the-middle, content tampering, and DDoS. To mitigate the impact of these attacks, they propose a secure edge caching scheme for content providers and UEs. In the proposed scenario, malicious users forge or alter cached content. The proposed solution is a Deep Reinforcement Learning (DRL) approach to optimize the caching payment strategy. This strategy focuses on the content rather than the user's behavior. With a mechanism that can classify users based on their behavior on the network, it would be able to block users with inappropriate behavior, preventing the sharing of harmful content.

In [68], authors developed a model, which employs blockchain as an incentive mechanism for caching in D2D. The blockchain transactions refer to the content share among two devices. The PBFT consensus protocol was employed in order to reach consensus under transactions in a faster and optimized way. The authors also considered the use of a Deep Reinforcement Learning algorithm to optimize the selection of nodes responsible for consensus and allocation of caching resources, in order to decrease latency and increase system scalability. However, the authors didn't considered node's trustworthiness and its impact on D2D caching.

Due to the complexity of trust management (Section 2.2) and the limited caching capacities of edge nodes, designing an efficient edge caching scheme for mobile users becomes a challenge. Addressing this challenge authors in [61] proposed a novel blockchain-based trustworthy edge caching scheme for mobile users. The blockchain is used to supervise the caching transactions between the edge nodes and mobile users without central authority avoiding caching tampering and poisoning. To optimize the blockchain and avoid massive storage overhead, authors proposed a expired transaction approach in which blocks with expired transactions could be ignored, enabling any entity

to validate the transactions from an intermediate block instead. Although the efforts for optimization authors didn't consider a multi-domain strategy and all edge data is handled by a single blockchain network.

In [18] authors integrate Deep Reinforcement Learning and permissioned blockchain into vehicular networks for intelligent and secure content caching. In the user plane, vehicles collect information of neighbors and road situation, such as road maintenance information, parking lot occupancy and video content. At the edge plane, BSs are distributed in a specific area to work as edge servers with communication and computing capability to accommodate Deep Reinforcement Learning and blockchain operations. The blockchain is utilized to empower distributed and secure content caching in edge coupled with a Deep Reinforcement Learning approach to optimize the content caching performance in edge taking in consideration mobility in device plane. However the proposed architecture do not handle multi-domain scenario and there is only one chain to handle all data in edge layer.

In [47] authors explored a secure D2D caching approach inspired on blockchain and trust management management called SecDUB. The collaborative trust model aimed to mitigate the transmission of invalid content, through the collection of indirect and direct observations. In addition, blockchain concepts were adapted to the dynamic and restricted scenario of the D2D communication in order to avoid the tampering of indirect observations and protect the control plane from data forgery and falsification. SecDUB operates only in D2D context which means that, the trust information related to a set of users will be discarded and also be limited to one specific edge host and domain, making this approach prone to multi-domain malicious users.

**Blockchain**

In [41], authors state that distributed ledger technology remains largely incompatible with the network virtualization paradigm, by its nature as functioning as an immutable record store. They stated that blockchain systems have focused on efforts to scale, disregarding its application in temporary storage. With that mindset they developed a permissionless multi-chain blockchain design for 5G core with *Lifecycle Control* and *Network Function Compatibility*. Blockchain is used solely as secure, decentralized storage, rather than a mechanism of direct policy and incentive control. Although the design is fairly compatible to work as a 5G core storage function, the multi-chain design was not tested with a hierarchical structure with multiple network layers.

In [4] authors proposed multi edge chain structure that accommodates edge control data to achieve cross-chain edge data sharing for heterogeneous blockchain systems. Each edge blockchain run independently, and collaborative edge computing results on chain storage can be performed concurrently. In this paper authors introduced the concept

of side-chains to mobile edge computing enabling trustworthy data consistency between different edge entities. Proposed solution encompasses a good sidechain approach for a generic edge computing-as-a-service data sharing scheme, but it didn't test the results upon high mobility scenarios.

In [58] authors designed a blockchain-powered framework that delivers trusted collaborative edge computing services, wanting to establish trustworthiness among all participants, due to the heterogeneity of all participants. This accountability scheme allows to establish a trust reputation system for all stakeholders, which can be further used for reliable selection of participants. The blockchain system is enabled in what is called *CEC Layer* corresponding to the edge layer, what could raise a scalability issue since there is only one blockchain to operate through the whole multi-layer architecture. Also the authors didn't investigate the capabilities of this same model as a distributed trust storage mechanism for UE data.

Table 3.2: Related work comparison

| | **Blockchain based approach** | **Trust Management** | **Multiple Domains** | **Caching** | **Scalability** |
|---|---|---|---|---|---|
| **[32]** | X | X | | | |
| **[70]** | X | X | | | X |
| **[63]** | X | X | | | |
| **[60]** | | X | | X | |
| **[68]** | X | | | X | |
| **[61]** | X | X | | X | X |
| **[18]** | X | | | X | |
| **[47]** | X | X | | X | X |
| **[41]** | X | | X | | X |
| **[4]** | X | X | X | | |
| **[58]** | X | X | X | | |
| **Proposed** | X | X | X | X | X |

By the extent of these related work reviews we can notice that when blockchain is proposed the majority of authors did not considered using it as a secure way to extend control data in a multi-layer architecture what would facilitate data sharing between different network domains [41, 32, 18, 68]. In the case when a multi-domain data sharing is proposed the architecture is not suited for trust information distribution since it would be better to have the data closer to the devices to avoid latency issues [70, 58, 4]. Therefore we can see the state of art lacks attention in secure multi-domain data sharing in the UE trust assessment scope, which requires improvements regarding latency goodput and overhead in mobile communication. In addition to these factors, scalability also depends on the way that trust information is persisted. If trust information is persisted in a centralized manner, then the scalability of the data sharing process will be limited by the capacity of the central server. However, if trust information is persisted in a distributed manner, then the scalability of the data sharing process can be improved. We need a

way to efficiently persist trust information enabling trust sharing among a larger group of mobile users, taking into account the increasing growth of data and interactions in next generations of mobile networking [14].

### 3.2.2 Architecture

We show the high level design of TrustMD proposed architecture in Figure 3.7. As schematized, we subdivided TrustMD approach in components, each component has a specific responsibility and acts within one of the three main actors of this framework: UE, MEH and Domain Controller (DCO).

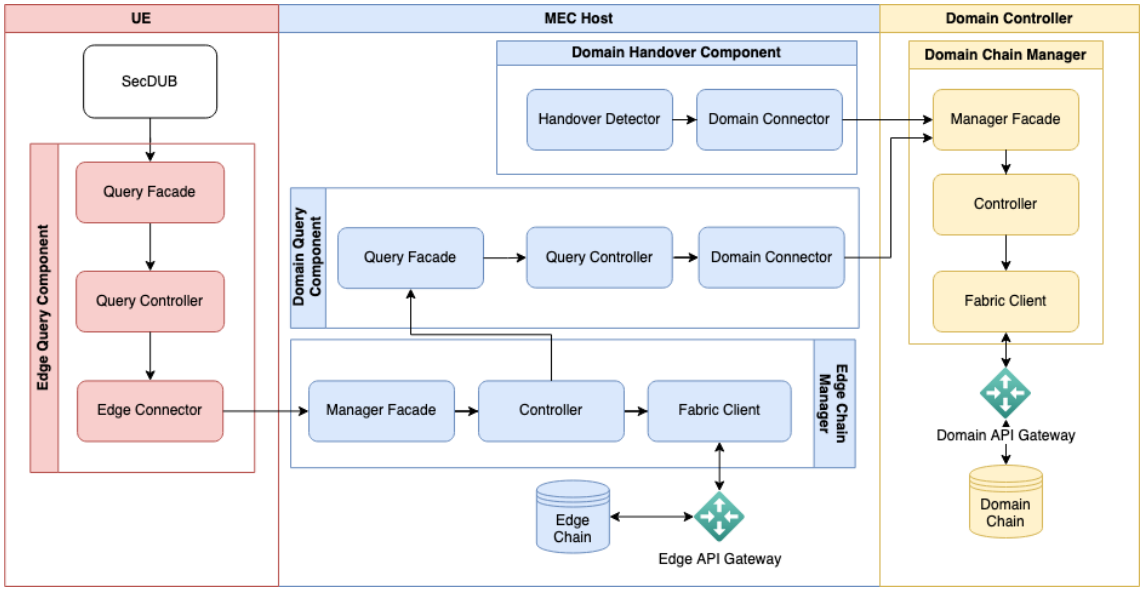


Figure 3.7: TrustMD High Level Design

UEs have two components: SecDUB (section 3.1) and Edge Query Component (Edge-QC). SecDUB is the component responsible to collect trustworthiness data and share it with the MEH, starting the Intra Domain Trust Update Flow (Intra-TUF) (section 3.2.3) in TrustMD framework. Edge-QC is the component in UE responsible to interact with MEH to query edge trust values of a node, starting the Trust Query Flow (TQF) (section 3.2.5).

MEHs have three components: Edge Chain Manager (Edge-CM), Domain Query Component (Domain-QC) and Domain Handover Component (Domain-HC). Edge-CM operates in both Intra-TUF and TQF as a bridge to perform any action in the Edge Chain (EC), so it responsible to interact with the EC and perform requested operations. EC is the distributed ledger present in MEHs alongside the same domain and stores UE trust values. Domain-QC is part of TQF (section 3.2.5) and operates in a analogously with Edge-QC, which the main responsibility is to interact with its DCO to query a node's domain trust

value stored in the upper level chain. Domain-HC is a standalone component of MEHs, which plays role of detecting domain handover events and act to update the domain-changing UE trust value in the upper level chain. Domain-HC is part of the Inter Domain Trust Update Flow (Inter-TUF) section 3.2.4.

DCOs, on the other hand, have Domain Chain Manager (Domain-CM) component that acts as part of both Inter-TUF and TQF as a bridge to perform any action in the Domain Chain (DC). Domain Chain is the upper level distributed ledger present in DCOs alongside the same domain and it stores domain-changing UE trust values.

The chain manager components, Edge-CM and Domain-CM, consist of three sub-components each: *Manager Facade*, *Controller*, and *Fabric Client*. These sub-components handle atomic operations independently. *Manager Facade* acts as an interface between TrustMD components and the chain manager components, providing Data Transfer Object (DTO)s for interactions with Edge-CM and Domain-CM. The *Controller* handles pre-processing of operations within the chain and executes requested actions. The *Fabric Client* interacts with the Fabric API Gateway to perform actions and submit transactions to EC or DC.

Similarly, the query components, Edge-QC and Domain-QC, also consist of three sub-components: *Query Facade*, *Query Controller*, and *Connector*. The *Query Facade* and *Query Controller* have roles similar to *Manager Facade* and *Controller* but specifically for query operations. The *Connector* serves as a bridge between the Query components and Manager components, enabling the execution of query operations by calling Edge-CM and Domain-CM.

The Domain Handover Component comprises two sub-components: *Handover Detector* and *Domain Connector*. The *Handover Detector* monitors handover events and takes appropriate actions in response to these events. The *Domain Connector* performs a role similar to the *Domain Connectors* in Domain-QC, enabling connectivity and communication between components in different domains. The following sections detail each flow mentioned earlier, connecting activities to a specific component.

### 3.2.3 Intra Domain Trust Update Flow (Intra-TUF)

Figure 3.8 highlights the activity diagram of Intra-TUF. This flow starts at the UE in Device Layer and it is designed to securely update the neighbors trustworthiness data collected by different UE's. The update action is projected to be a standalone entry point for trust data collection in a MEH, but here we project this flow to match the EC update with the intra domain SecDUB approach.

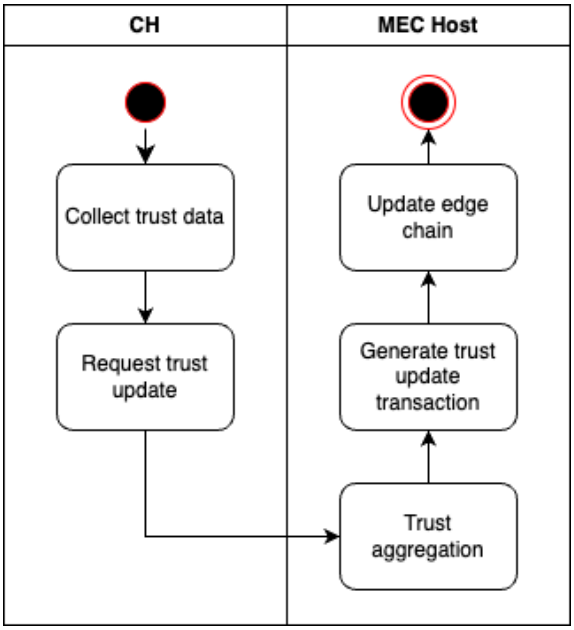


Figure 3.8: Trust update activity diagram

### Collect Trust Data

Our proposal currently lacks a self-service mechanism for gathering device trust evidence in the context of MEC Host (MEH). As a result, behavioral information must be collected from the device layer. The current state of our proposal does not propose a MEC Host self-service mechanism for device trust evidence gathering. Thus, the behavioral information needs to be collected from device layer. Essentially the design could be coupled with any behavioral framework or mechanism capable of assessing trust of a node upon other node, but here we are proposing to use SecDUB as the behavioral collecting approach. This situation leverages two possible scenarios for trust assessment in MEH: (1) expect trust updates from all devices in device layer or (2) assign trust update responsibility to a specific set of entities. Approach (1) is not advantageous considering the overhead added to the edge layer due to frequent updates from multiple users. Approach (2) seems like the most reasonable way of handling in terms of overhead, leading to the next challenge: what is the ideal set of UE to assign trust update responsibility? CH has direct access to its peers trustworthiness data stored in SecDUB's ledger.

Also, in SecDUB we choose each CH utilizing a clustering approach in which trustworthiness of an UE is assessed in order to make this node a CH, making the CH the most suitable device to provide MEH with trustworthiness information (section 3.1.5).

### Request Trust Update

Each CH can collect trust data from SecDUB's ledger and periodically send it to its MEH. MEC Hosts would be capable of leveraging each evaluated node trustwor-

thiness with the help of TDS. However, MEHs initially do not know devices behavioral information. To overcome this limitation, CHs share their direct trust values to its associated MEH. Here we assume that the connection between an UE and the MEH occurs through a secure channel, allowing share of control information with CH.

Let $\Delta_{Intra-TUF}$ be the update interval in which a CH send the Intra Domain Trust Update (IaTU) message to the MEH. The message contains the following fields $IaTU = \{\tau, TD, c_{size}, l_{size}\}$, where $\tau$ the timestamp of these trust values, $TD$ corresponds to the set of most recent trust decisions collected by the CH, $c_{size}$ the number of peers within CH in cluster and $l_{size}$ is the size of CH's ledger in number of blocks.

Each element in $td \in TD$ is a tuple containing: (1) the opinion/decision $d_{UE_o,UE_e}$ of an opinator node $UE_o$ regarding $UE_e$ trustworthiness; (2) the direct trust value $T^D_{CH,UE_o}$ of CH upon the opinator $UE_o$.

### Trust Aggregation

Upon receiving IaTU, the Edge-CM component in MEH is activated to operate the trust update in EC. Firstly, it executes the trust aggregation procedure to calculate the uncertainty of received trust opinions and normalize the data. Trust aggregation will proceed similarly to trust assessment of indirect observations (section 3.1.6), but using direct trust observation of CH about its set of cluster members. At the end of trust aggregation, is expected from Edge-CM the output of the highest belief hypothesis regarding each evaluated UE aiming to create one transaction per evaluated UE. First Edge-CM builds the decision matrix $M_{UE_e,td}$ that maps an evaluated node $UE_e$ to a set of $td$ originated from the request message. Based on the trust decisions regarding a specific node, by the help of TDS (section 2.2), Edge-CM will find the degree of belief in each hypothesis (*trustworthy* or *untrustworthy*), using $T^D_{CH,UE_o}$ the direct trust value of CH as the associated evidence mass.

At the end of the aggregation process, for each evaluated node, Edge-CM will have a set of belief values upon trustworthiness hypothesis of each evaluated node (Equation 2-3). Upon this data, Edge-CM will maximize the degree of belief to find the highest belief among all set of hypothesis, which is the resulted degree of belief upon each $UE_e$ trustworthiness hypothesis (Equation 3-1). This information will feed EC and trigger the chain update process.

$$bel = \max \sum_{s \subseteq S} m(s) \tag{3-1}$$

### Generate Trust Update Transaction

When trust aggregation finishes the next step is prepare to update the EC with the new trustworthiness hypothesis. To start this procedure, MEH will create a new Intra Domain Trust Update Transaction ($tx_{Intra}$), for each evaluated node $UE_e$ with the following format:

$$tx_{Intra} = (\tau, \xi, id_{MEH}, id_{CH}, id_{UE_e})$$

where $\tau$ is the 4-byte long transaction creation timestamp, $\xi = (bel, d_{MEH,UE_e})$ the 8-byte long trustworthiness decision upon evaluated node, $id_{MEH}$ is the 4-byte long IP of MEH, $id_{CH}$ is the 4-byte long IP of CH source of trustworthiness data and $id_{UE_e}$ is the 4-byte long IP of the evaluated node $UE_e$, totaling 24 bytes per transaction. Each $tx_{Intra}$ is multicasted to MEHs peers in the same domain and added to the transaction pool $Tp$ managed by Edge-CM, to eventually start the EC update activity.

### Update Edge Chain

To update the edge, we propose a permissioned-based architecture, in which the EC is designed on top of the Hyperledger Fabric distributed ledger framework [6]. Hyperledger Fabric is a blockchain framework that offers a secure, private, flexible, scalable and resilient platform for implementation of multi-purpose distributed solutions. Also Fabric can leverage consensus protocols that do not require a native cryptocurrency as a incentive or costly mining smart contract execution, reducing some significant risk/attack vectors.

Fabric operates with plug-and-play membership and consensus services that implements smart contracts (chaincoodes) to regulate interactions among parties participating in the framework. Typically a transaction flow in fabric follows four phases: (1) proposal, (2) endorsement, (3) ordering and (4) commit. Bringing to our scenario, a client MEH who originated a transaction $tx_{Intra}$, submits the transaction proposal to one or more peers in network to collect endorsements. Transactions will be simulated, endorsed and then sent back to the client MEH, which then submits to the ordering service. Ordering service will execute the consensus mechanisms and then submit it to a committing peer, responsible to commit the transaction in ledger.

To avoid updating EC with trust evidence of high degree of uncertainty, we propose to deploy a Fabric chain-code that checks if the degree of belief $bel \in \xi$, in $tx_{Intra}$ is greater than a pre-defined threshold ($\gamma_b^{Intra}$) prior to the update operation, updating the trust decision regarding node $UE_e$ if so. This configuration reduces uncertainty of trust assessment by prioritizing the most "believable" evidences following TDS rules.

We can subdivide transactions in blockchain layer between two different layers. Transactions committed to the ledger are considered on-chain transactions, on the other hand, transactions managed by the ledger, but stored in a standalone collection database are considered off-chain transactions. As the edge layer handles a lot of data, size of the blockchain network may increase at exponential speed. In general on-chain transactions takes longer time to execute then off-chain transactions, which increases the query process of on-chain data. Also, storage capacity related issues may lead to scalability problems. To bring scalability in data operations, we employ an off-chain storage mechanism to enhance data query in MEC layer and avoid potential scalability issues [26, 28].

### 3.2.4 Inter Domain Trust Update Flow (Inter-TUF)

Figure 3.9 highlights the activity diagram of Inter Domain Trust Update Flow (Inter-TUF). This flow starts at the MEH in the Edge Layer and it is designed to securely update the Domain Chain upon a detected Inter Domain Handovers. The update action is projected to be a standalone entry point for trust data collection at the Domain Layer level. We choose a reactive approach operation to avoid unnecessary updates at the upper level chain maintaining overhead at reasonable levels in DCOs, since handovers to other domains occur less frequently.

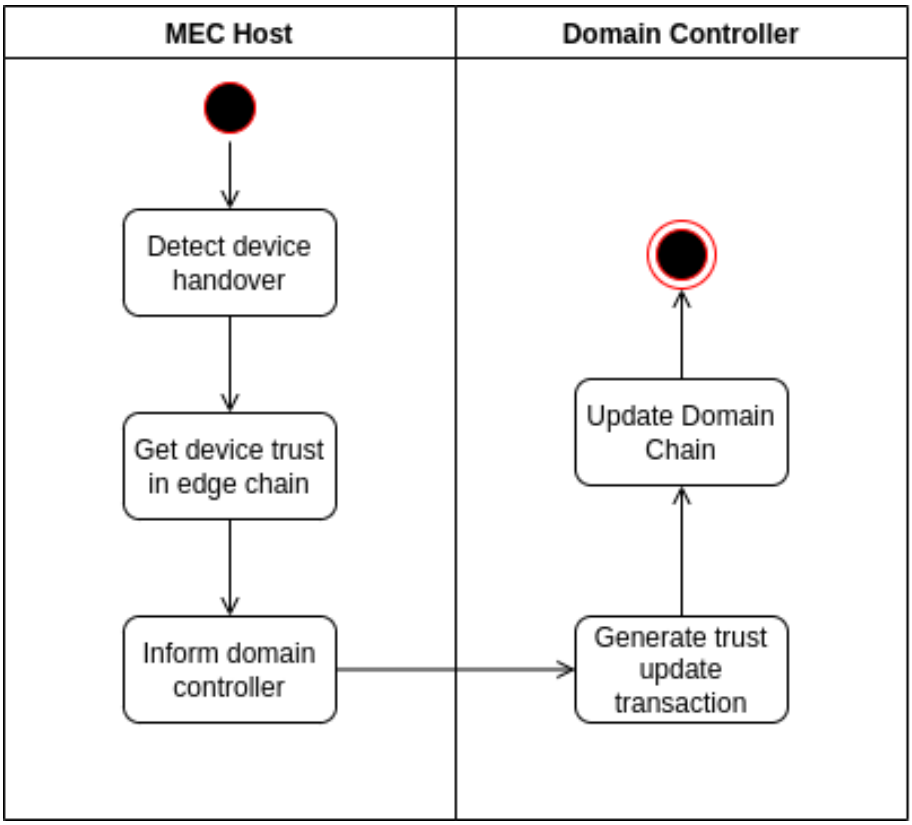


Figure 3.9: Inter domain handover trust update activity diagram

### Detect UE Handover

The European Telecommunications Standards Institute (ETSI) proposes these two MEC APIs: Radio Network Information Service (RNIS) and Location Service (LS). The RNIS is a service that provides radio network related information to MEC applications and to MEC platforms and LS leverages zonal presence service. There are several approaches in literature that deals with inter-domain handover utilizing both RNIS and LS to efficiently detect handover events [50, 38, 51]. RNIS and LS can be used to predict service relocation, identify capable MEHs and trigger handover actions to the underlying network.

It is out of this work scope propose a handover detection/prediction mechanism. With that said, we assume that the Domain-HC present in the source MEH will detect an event with the following data:

$$hEvent_{sMEH} = (\tau, id_{sMEH}, id_{dMEH}, id_{UE_h}, id_{DCO})$$

where $\tau$ is the event timestamp, $id_{sMEH}$ the IP of the source MEH, $id_{dMEH}$ the IP of the destination MEH in the new domain network, $id_{UE_h}$ the IP of the UE doing handover and $id_{DCO}$ the identification of the new UE domain.

### Get Device Trust in EC and request trust update

Once $hEvent_{sMEH}$ is detected by Domain-HC, MEH will utilize Edge-CM to trigger a query transaction to retrieve trustworthiness evidence of node $id_{UE_h}$ among all transactions originated by $id_{sMEH}$. Considering a set of trust evidences $E$, one trust evidence $\xi$ is extracted from EC as the tuple $(bel_{UE_h}, d_{MEH,UE_h}) \in E$. When $\xi$ is retrieved from EC, MEH send an Inter Domain Trust Update (IeTU) request to the Domain Controller of its associated network.

### Generate Trust Update Transaction

Upon receiving IeTU message, DCO activates de chain manager Domain-CM to operate the trust update in DC. The starting of Inter domain update procedure is given by the DCO that received IeTU request. The referred DCO will be responsible to create a new Inter Domain Trust Update Transaction ($tx_{Inter}$), with the following format:

$$tx_{Inter} = (\tau, \xi, d_{sMEH}, id_{dMEH}, id_{UE_h}, id_{DCO})$$

where $\tau$ is the 4-byte long timestamp, $\xi$ the 8-byte long trustworthiness evidence of the evaluated UE, $id_{sMEH}$ the 4-byte long IP of the source MEH, $id_{dMEH}$ the 4-byte long IP of the destination MEH in the new domain network, $id_{UE_h}$ the 4-byte long IP of

the evaluated UE and $id_{DCO}$ the 4-byte long identification of the new UE domain, in total of 28 bytes.

#### Update Domain Chain

The trust update in domain layer will follow the same idea as it is at the edge level, where the Hyper Ledger Fabric will serve as the blockchain framework that handles the distributed ledger and leverages the underlying mechanisms of the blockchain technology. The Domain Controller that received the request will submit the $tx_{Inter}$ transaction proposal triggering Fabric mechanisms. To validate $tx_{Inter}$ transactions, we also propose a verification upon each transaction hypothesis degree of belief, by checking if its value is greater than a pre-defined threshold $\gamma_b^{Inter}$.

### 3.2.5 Trust Query Flow (TQF)

Figure 3.10 highlights the activity diagram of Trust Query Flow (TQF). This flow is triggered by an UE in Device Layer and it is designed to efficiently/securely retrieve reliable trust data from Edge Layer. It is expected that by the end of the flow the client UE will be able to assess other node's trust, even if the assessed node was out of client UE domain.

TQF is projected to be a synchronous request/reply data flow in which UEs inside a specific domain can get inter-domain trust evidence from edge. The capability offered by TQF to an UE promise to enhance trust assessment in edge layer and restrict the impact of malicious/untrustworthy nodes during UE caching.

In our approach, we propose that TQF takes place when an UE receives a caching request from an unknown neighbor so it cannot distinguish whether the requester is trustworthy or not, as it does not have what it takes to build the neighbor trust values (section 3.1.3). It is good to stress out that, a node primarily relies on SecDUB's trust values collected through direct and indirect observation of its neighbors, it queries for trust in MEC layer just in case it has no history regarding the evaluated UE. This approach avoids excessive queries in upper layers.

#### Query for device trustworthiness in Edge Chain

The flow starts with a device $UE_r$ at the Device Layer activating Edge Query Component component. The Edge Query Component is responsible to build/send a Trust Query Request (TQ-Request) and synchronously wait for a Trust Query Response (TQ-Response). TQ-Request contains a IP identification of the evaluated node $id_{UE_e}$ and the IP identification of the requester node $id_{UE_r}$. On the other hand, TQ-Response contains both request and evaluated nodes identifiers and the queried transaction.

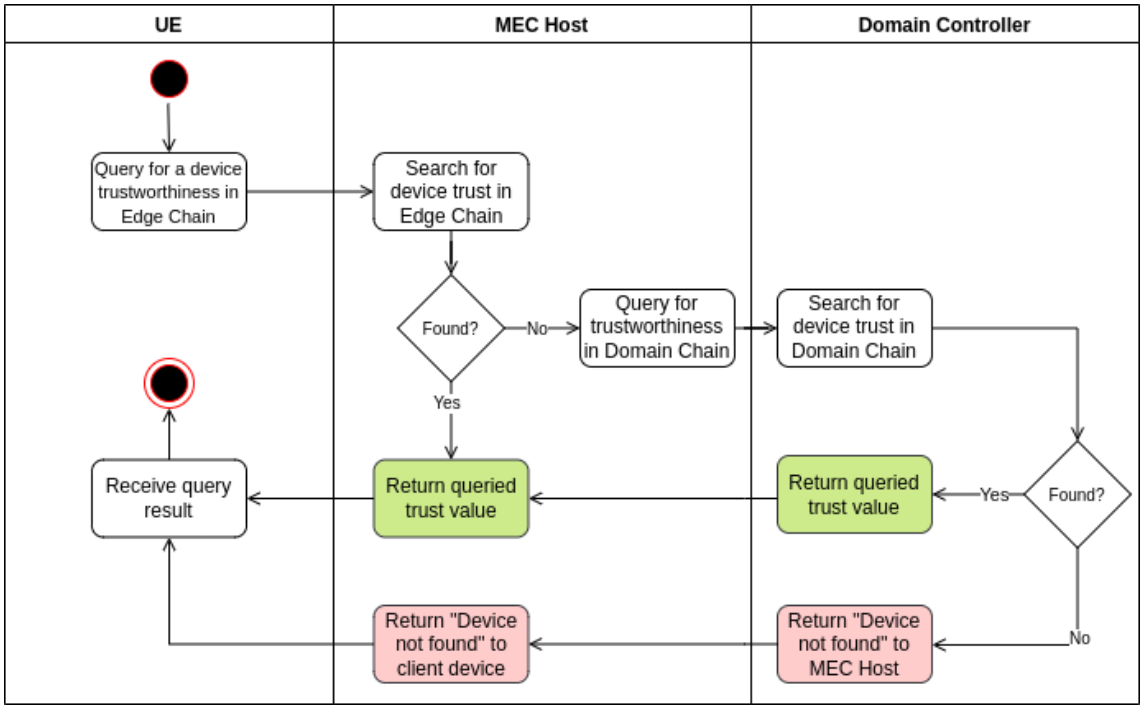


Figure 3.10: Trust query activity diagram

### Search for device trust in Edge Chain

Upon receiving a TQ-Request, MEH activates the Edge Chain Manager component to manage Edge Chain interaction regarding $UE_r$ request. Edge Chain Manager execute the query looking for trust data regarding $UE_e$ trustworthiness. The query follows the same procedure described earlier, in which the evaluated node id $id_{UE_e}$ and its MEH identification $id_{MEH}$ are the available fields for query execution [62].

If Edge Chain Manager finds a transaction in the query operation, it stops the query process, builds and finally sends TQ-Request with the queried transaction. Else if no trust data regarding this transaction was found in Edge Chain, the process is delegated to the upper level chain, by activating Domain Query Component component in MEH. Domain Query Component is responsible to build/send a Edge Trust Query Request (ETQ-Request) and synchronously wait for a Edge Trust Query Response (ETQ-Response). This Domain Query Component request is basically TQ-Request enriched with edge requester IP identification $id_{MEH}$. The same logic follows for Domain Query Component response, which is TQ-Response enriched with $id_{DCO}$ domain identification.

### Search for device trust in Domain Chain

Upon receiving a ETQ-Request, DCO activates the chain handler Domain-CM to manage DC interaction regarding $id_{MEH}$ request. Domain-CM executes the query looking

for trust data related to $UE_e$ in a similar way described in the previous section.

If Domain Chain Manager finds a transaction with $id_{UE_e}$ trust data, it stops the query process, builds and finally sends ETQ-Response reply back to requester MEH. The reply is handled by Domain Query Component in MEC Host that builds the final TQ-Response and sends it back to requester UE. Else if not trust related data Edge Chain Manager is found in Domain Chain, the overall query process return an empty $\xi$ in ETQ-Response.

**Receive query result**

The flow finishes when Edge-QC receives the final reply ETQ-Response from the requested MEH. Upon receiving the reply the requester UE extract $\xi$ from the message and act accordingly.

As stated before, $\xi$ corresponds to the trustworthiness decision upon the evaluated node. By the means of trust's conceptual definition, trust is uncertain due to its subjectivity and non-transitivity (section 2.2), that is, UE's ideally cannot blindly trust the MEHs and DCOs opinions. This statement is even more clear considering that MEH's opinion is based on another UEs opinions. Addressing this point we propose trustworthiness assessment on MEHs ($T_{UE,MEH}$) and domains ($T_{UE,DCO}$) as well. This trust assessment will follow the rules of trust by direct observations (Equation 2-2). In that way a trustworthy MEH or DCO transmit assertive trust decisions to its UEs.

The trust of a MEC Host or a Domain Controller is updated by the receiver within the end of a Trust Query Flow. If a requester UE receives a TQ-response wrapping a Edge Chain transaction it means that the response was given by the edge layer, otherwise the response was given by the domain layer. The trust update of MEC Hosts and Domain Controllers happens accordingly with these type of responses. Thus, if a MEC Host considers an evaluated node trustworthy when it its not, the requester UE will decrease the degree of trust upon the MEH that provided the trustworthiness decision. The same logic follows for domain trustworthiness assessment upon the receipt of a domain response at the and of Trust Query Flow.

An UE queries for trust in edge only if trust the associated MEC Host (MEH) and Domain Controller (DCO). We define trust thresholds $\kappa'_{MEH}$ and $\kappa'_{DCO}$ for MEH and DCO trustworthiness respectively, therefore MEH is trustworthy if $T_{UE,MEH} \geq \kappa'_{MEH}$ and DCO is trustworthy if $T_{UE,DCO} \geq \kappa'_{DCO}$.

Finally, the joint combination of $T_{UE,MEH}$, $T_{UE,DCO}$ and ETQ-Response enables the requesting UE to update the indirect trust value of the evaluated node. This calculus follows Theory of Dempster-Shafer rules and considers $T_{UE,MEH}$ or $T_{UE,DCO}$ as the source of mass (Appendix A), depending on what layer ETQ-Response originated.

CHAPTER 4

# Results

In this chapter we present the results regarding the proposed secure multi domain D2D caching framework. As discussed earlier, we subdivided the proposal upon intra domain and inter domain approaches. In the current stage of the project we have delivered intra-domain approach results described in section 3.1 named. SecDUB results were conducted by comparison of ProSoCaD base model [17] and the proposed intra-domain scheme.

## 4.1 Scenario

The simulations were carried out using the Network Simulator 3 (NS-3) [46], widely used and recognized in the academic/industrial network community. We built a scenario of wireless mobile networks with some foreseen characteristics in 5G networks, enabling the representation of cellular network components, such as EPC (Evolved Packet Core) and eNB (eNodeB), and ad hoc communication along the lines of D2D. Table 4.1 shows the parameters of the system and the simulation environment.

Each UE was configured with two wireless communication interfaces, i.e. the first interface for communication with the cellular network structure and second interface for D2D communication in *adhoc* mode. The mobile network was simulated using the LENA model, which implements the LTE [5] standard in NS-3. We employed the physical layer model which is mostly used by the community, *YansWifiPhy*.

The D2D communication model adopted is outband, that is, a Wi-fi ad-hoc mode is employed. The D2D devices must be at the communication range with each other in the Wi-Fi ad hoc mode in order to communicate (D2D connection). All nodes in this scenario are in the same cellular cell and they are served by one Base Station, however not all devices are reachable through D2D connection. ProSoCaD showed advantages in relation to other proposals in terms of: data offloading, transfer time and cache hit rate. We modified the protocol structure so that video transmission occurs when the receiver trusts the transmitter in order to adapt trust management to caching policies.

Table 4.1: Simulation Parameters

| **Parameters** | **Value** |
|---|---|
| MAC Layer | IEEE 802.11g |
| D2D Communication Mode | Ad Hoc (Outband) |
| PHY Layer (NS-3) | *YansWifiPhy* |
| Mobility Model | RPGM (Bonn Motion) |
| Area Topology | 2000m x 2000m |
| Maximum Distance between nodes | 200m |
| Maximum Speed | 1.6 m/s |
| Minimum Speed | 1.0 m/s |
| Associated Weight with $T^D$ ($\omega$ in Eq. 2-1) | 0.5 |
| Associated Weight with $T^I$ (1-$\omega$ in Eq. 2-1) | 0.5 |
| Trustworthiness Threshold ($\kappa$ in Eq. 2-1) | 0.5 |
| CH update interval $\Delta_{Intra-TUF}$ | 15 updates/min |
| Total Simulation Time | 100s |
| Cryptographic Hash Standard | SHA-256 |
| Algorithm For Encryption and Authentication | ECDH and ECDSA |

We adopt Reference Point Group Mobility (RPGM)[53] to model the user mobility similar to ProSoCaD. The RPGM simulates the movement of a group of individuals, in scenarios with low level of mobility, similar to shows, conferences or events. Groups move to pre-defined locations and when they reach that location, they take a random time break, until they move again. The mobility is random, but limited to the perimeter of the group, so that the meeting of two groups can lead to the separation of its members or the creation of a new group.

In view of the low computing power of mobile devices and the volatility of D2D communication, performance is essential. The Eliptic Curve Discrete Logarithm Problem (ECDLP) is employed as the basis for the generation of Eliptic Curve Digital Signature Algorithm (ECDSA) and Elliptic Curve Diffie Hellman (ECDH) keys, which guarantees greater performance than Discrete Logarithm Problem (DLP) [29].

## 4.2 Evaluation Parameters

We used different scenarios to assess the proposed framework described in this article, each of which envisions possible network configurations, which can impact on the system's performance in different ways. We defined that malicious users share only false video at most scenarios, that is, content with invalid metadata. However, there are scenarios that malicious users share true and false videos alternatively and randomly.

The evaluation criteria fundamentally analyze the results obtained, between two aspects: (1) network performance and (2) efficiency of the security framework. The network performance evaluation seeks to evaluate the impact of SecDUB on data traffic

and control overhead, taking into account traffic performance parameters such as Packet Lost Rate (PLR), Throughput and Goodput. It is worth to point out that goodput is the amount of serviceable data traffic (bits) delivered by the network per unit of time (excluding transmitted bytes of invalid videos). We also analyzes the impact of the security mechanism on control overhead, which is calculated based on the amount of control messages related to the consensus protocol and clustering.

The efficiency of the security framework analyzes how the behavior of mobile user (malicious or not) impacts on the SecDUB trust model and therefore, what it is the impact of the degree of trust in the distribution of invalid content. To reach this, we evaluated the Average Degree of Trust (ADT) of all D2D nodes and also the False Negative Rate (FNR) when the behavior of malicious node changed in the collaborative trust management model according different types of attacks. In this context, a false negative occurs when an invalid (false) video is transmitted by a malicious user, that is, when the receiving node mistakenly classifies a malicious transmitting node as trustworthy.

Some experiments were conducted based on these metrics, the Average Degree of Trust of the nodes and (2) the False Negative Rate. ADT aims to analyze the influence of sending invalid content on the degree of trust of the nodes, whereas FNR aims to investigate the efficiency of the collaborative trust management model on preventing the transmission of invalid video. We calculate FNR with the Equation 4-1, where $s_{invalid}$ corresponds to invalid content successfully transmitted and $ns_{invalid}$ corresponds to invalid content not transmitted, i.e. blocked by the SecDUB.

$$\text{FNR} = \frac{s_{invalid}}{s_{invalid} + ns_{invalid}} \tag{4-1}$$

We performed 10 runs and averaged the results in simulations. The results employ error bars with confidence interval based on significance level of 95%.

## 4.3 Analysis of SecDUB Network Performance

We assess the impact of the SecDUB on communication as a whole (data and control plane) in this subsection. When adding a layer of security over any communication system, it is necessary to achieve a trade-off between security and performance. However, the overall performance needs to be comparable to ProSoCaD[17], since it makes no sense to impose security solutions that deteriorate significantly the traffic performance. In addition, we emphasize that we evaluated the proposal in a stressful scenario with a high number of malicious users similar to [55, 66], in which it is notable the exacerbated growth in number of attacks. In order to simulate this situation, we set 50% of the nodes

as malicious, where malicious nodes send only invalid videos during the entire simulation time.

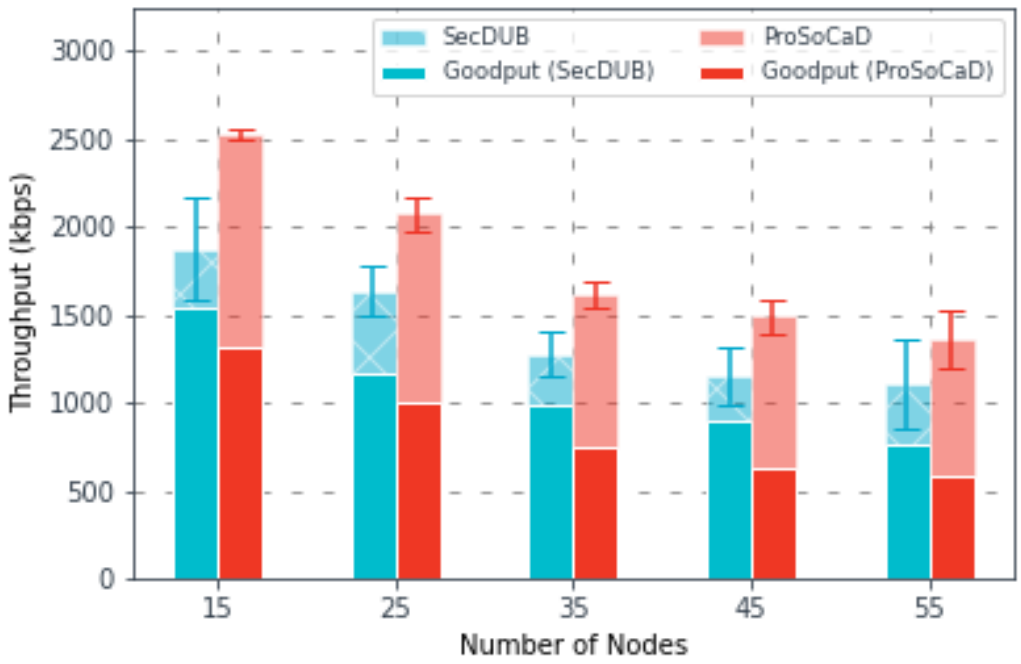


Figure 4.1: Average Throughput and Goodput

We use the performance evaluation parameters: throughput, goodput, Packet Loss Rate and overhead. In addition, we consider the average time to complete the consensus in order to analyze the behavior of the PBFT by increasing the number of nodes. The useful flow corresponds to the percentage of the flow corresponding to the sending of valid contents. We measured these parameters in the simulation by varying the number of nodes in order to assess how SecDUB impacts the scalability of the network as well as to different degrees of node density, thus dividing this experiment into five groups of scenarios: (T1) 15 nodes, (T2) 25 nodes, (T3) 35 nodes, (T4) 45 nodes, (T5) 55 nodes.

We can notice in Figure 4.1 that the throughput decreases when SecDUB is used, by an average of 17.96% compared to ProSoCaD, while there was a relative average growth of 6.31% in the PLR illustrated by Figure 4.2, which can be justified by the SecDUB employement. However, goodput is higher when the SecDUB is used in all groups of tests. The goodput with SecDUB framework represents 80% of the throughput in category T1, while the goodput approximately reaches 50% with ProSoCaD. The goodput of the proposed framework achieves 70% of the throughput on average while ProSoCaD achieves on average 40%. Despite decreasing the throughput, SecDUB was

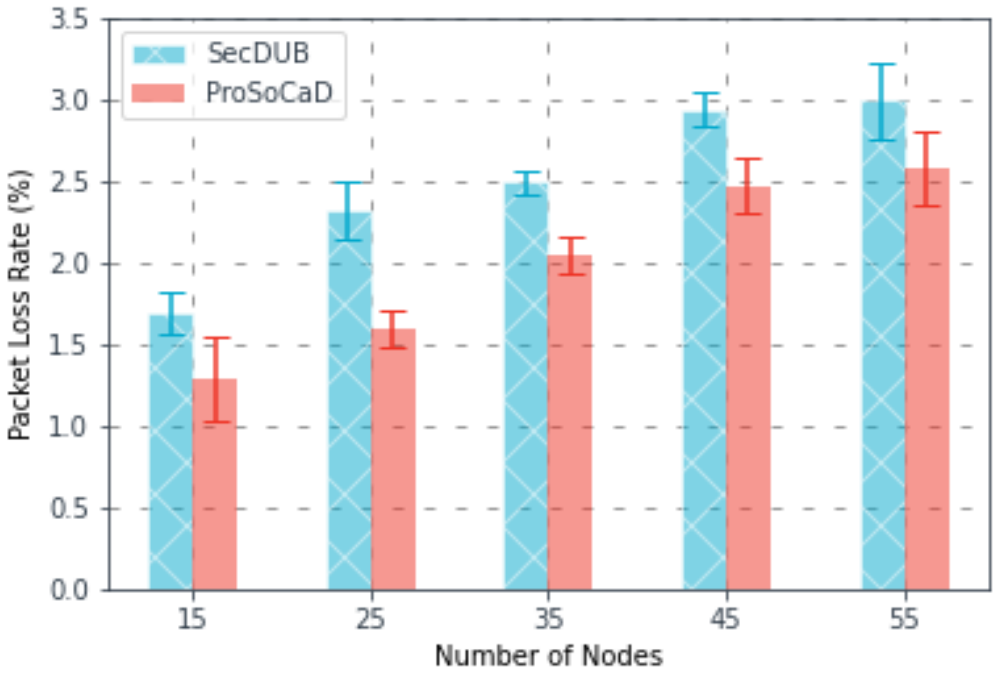


Figure 4.2: Average Packet Lost Rate

successful in improving significantly the system's utility by increasing the goodput through the diminishing of the sharing of invalid video in the D2D communication.

Figure 4.3 illustrates the growth in the average cluster size (blue line) and the average number of clusters (red line) by increasing the number of nodes in the network. We can observe that average cluster size increases as increase the number of nodes, i.e. the greater the number of nodes, the greater the proximity between them, so the clusters are formed with more nodes. Nevertheless, it works differently in average number of clusters (red line), the number of clusters does not vary too much with the increase of number of nodes.

The number of clusters increases at scenarios of 15 and 25 nodes but it decreases in scenarios of 35 and 45 nodes and, it comes back to increase in at scenario of 55 nodes. A D2D network which has higher number of clusters and a lower number of nodes per cluster presents lesser message exchange than a scenario with few groups but many nodes per cluster. Hence, a higher density of these scenarios does not result in an increase in the number of clusters in red line of Figure 4.3, since the nodes are distributed among the clusters fairly.

This can be justified by the proposed clustering scheme, since the cluster formation and maintenance seeks to select CHs which are more influent and close to the CMs (distance in hops - cluster radius). It is important to remember that the degree of the influ-

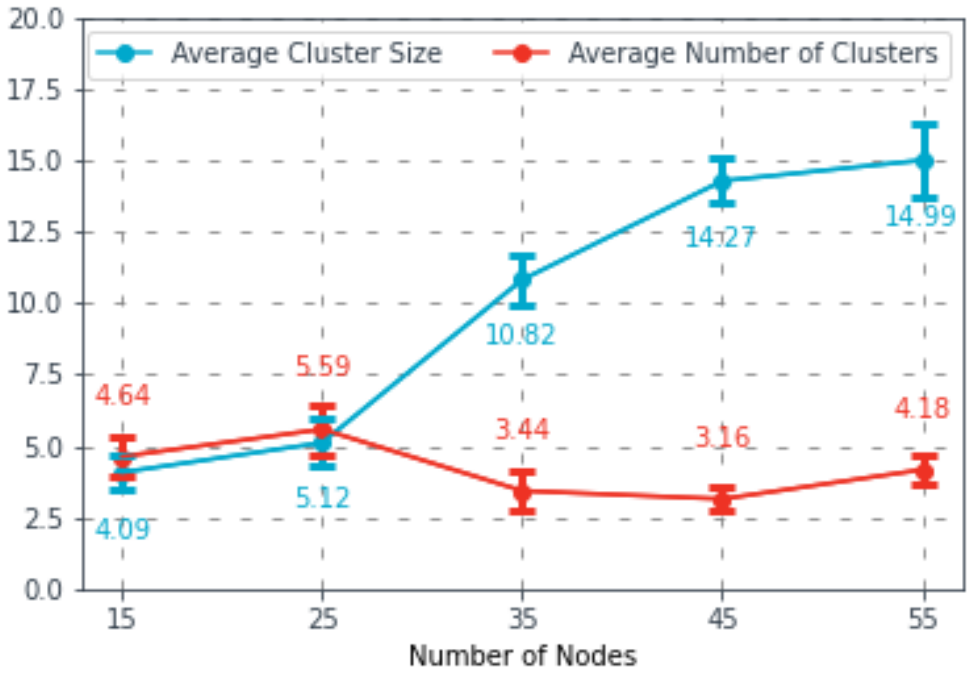


Figure 4.3: Average Cluster Size Average

ency of the nodes are based on social centrality metrics like number of adjacent neighbors and contact time. Thus, higher number of nodes higher the number of nodes with more influence. In other words, clustering scheme seeks an equilibrium between the number of nodes per cluster and the number of the clusters at the D2D networks.

A scenario with more clusters, but with a lower number of nodes per cluster, presents lesser consensus message exchange than a scenario with few clusters but many nodes per cluster. Nonetheless, as the Cluster Members (CMs) participating in the consensus increases, the greater the quantity of CMs and more messages are needed until its conclusion, that is, the greater the aggregated overhead by the consensus control messages, as it can be seen in Figure 3.4 previously shown. The same logic can be applied to the time of completion of the consensus in Figure 4.4, considering that the more messages exchanged, the more time is required for all stages of the consensus to be completed.

Figure 4.4 shows that the consensus protocol varies the average completion time between 0.20s up to 1.85s. Although the PBFT protocol is characterized by higher communication overhead, since maintaining the protocol requires a lot of message exchange, the average completion time of consensus is not significantly high. The clustering scheme and adopted concepts of the blockchain, such as one transaction by block, PBFT and blockchain based on clustering helped to simplify the consensus process and therefore, it

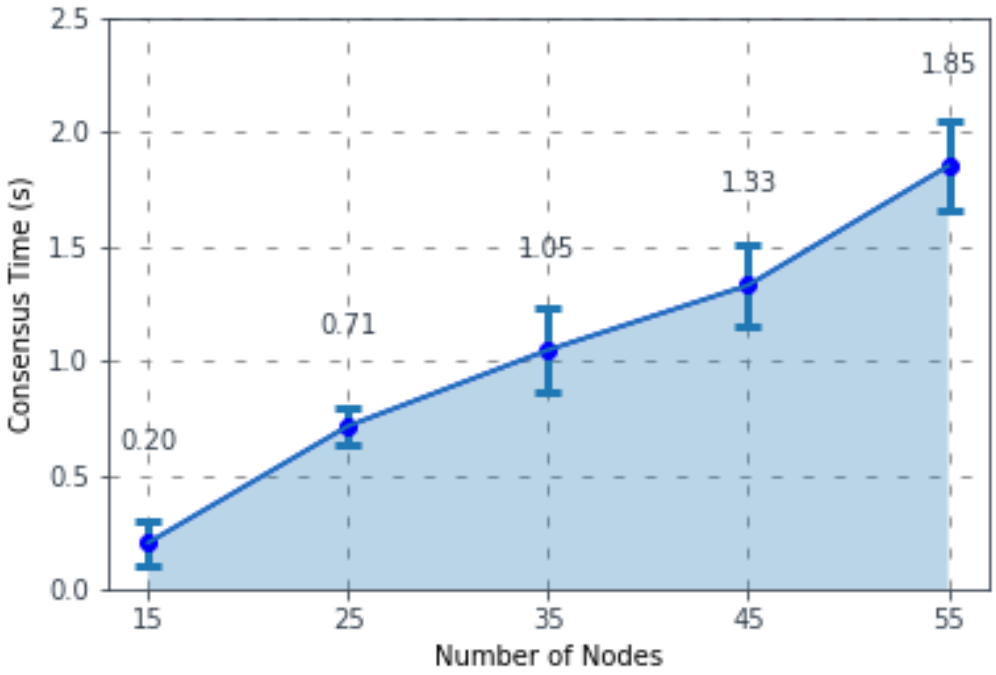


Figure 4.4: Average Consensus Completion Time

does not increase significantly completion time.

As also a result of the proposed clustering scheme, the SecDUB achieves lesser degradation of the overall performance (for both throughput and PLR illustrated in Figures 4.1 and 4.2) when the number of nodes is increasing, since the difference between SecDUB and ProSoCaD diminishes as increase the number of nodes. For instance, the average throughput minimum and maximum achieved by SecDUB is about 1100 and 1800 Kbps respectively, whereas the average throughput minimum and maximum achieved by ProSoCaD is about 1400 and 2500 Kbps respectively. Hence, the SecDUB provides a more scalable solution due to the proposed clustering scheme.

It is important to stress out that the number of control messages increases due to the growth in the number of nodes, as it is shown in Figure 4.5. However, the increase of overhead is not directly proportional to the increase of number of nodes, i.e. there was a slighter increase in the scenarios 35, 45 and 55 nodes. This can be explained by the same reasons discussed before in Figure 4.3: a higher density of these scenarios does not result in an increase in the number of clusters in red line of Figure 4.3, since the proposed clustering scheme achieves an trade-off between the number of nodes per cluster and the number of the clusters, thus the nodes are distributed among the clusters fairly (i.e. increase of the number of nodes in blue line of Figure 4.3 while similar number of clusters in red line are reached).

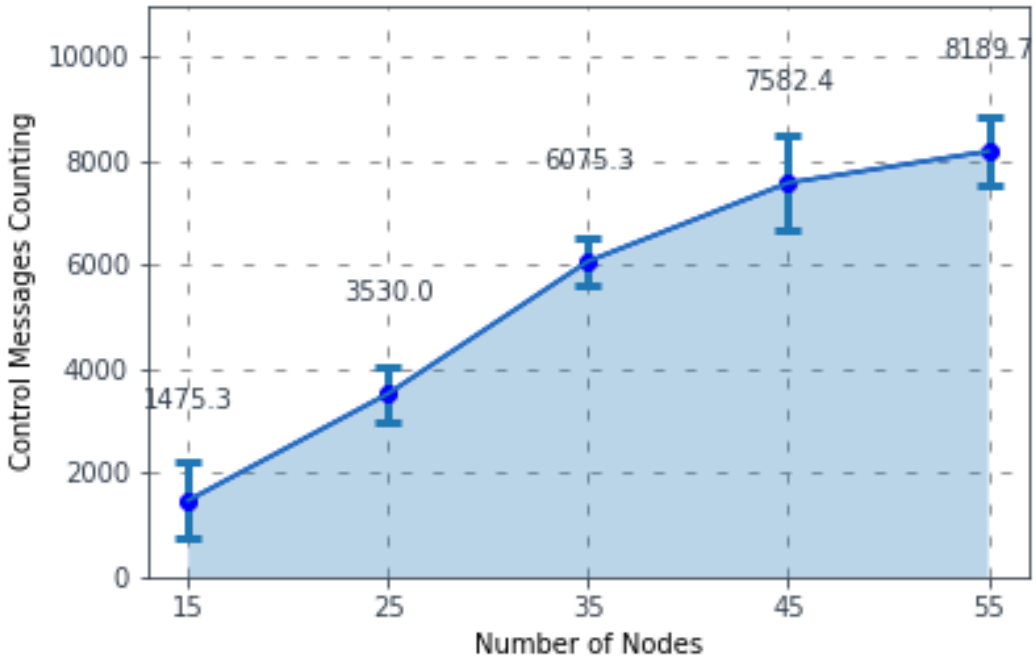


Figure 4.5: Average Control Messages Counting

The persistence of the metadata in the ledger is not critical, as nodes participate in sporadic and shorter events and they may fall out of their cluster. Even though every block has a single transaction in our proposal, the block metadata is about 156 bytes (header+data). The size of the video metadata is tiny and fixed (32 bytes), whereas the number of indirect observations (vote) is a variable value because it depends on the number of nodes in the cluster (blockchain network). Figure 4.6 describes the average ledger size and average block size for the different combination of nodes. We can notice that both average ledger size and average block size follows a similar tendency, i.e. they increase proportionally to the number of nodes, however the absolute increase is not significant. Besides, clustering is well-known for improving the scalability and aid to reduce the impact of the increase of nodes in the overhead. Extrapolating the values in the graph for the densest scenario (55 nodes), in four hours (this is reasonable duration time for our scenario, e.g. shows and events in sport arenas and restaurants) of simulation following a linear growth, the average size of the ledgers would be approximately 17 MB. Thus, the metadata does not cause the blockchain to explode or any scalability issue.

Figure 4.7 depicts the latency (i.e. the amount of time consumed to send a video content from the source to the destination). The graph shows an interesting finding, the SecDUB results in a very similar latency in more dense scenarios (35, 45 and 55 nodes), whereas SecDUB decreases the latency in sparse scenarios (15 and 25 nodes). Although

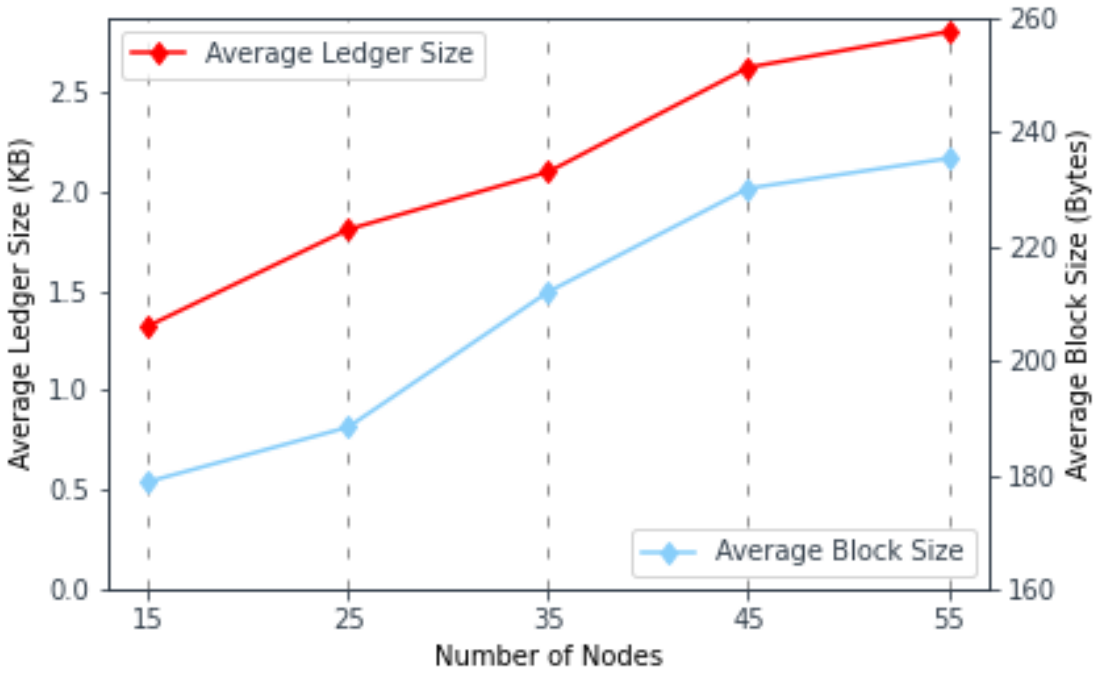


Figure 4.6: Block/Ledger Average size

SecDUB incurs more time to assess a valid video due to overhead of communication and processing of clustering and consensus, SecDUB avoids malicious transmission (i.e. invalid videos), which results in a fewer average total of Transactions Per Second (TPS) than ProSoCaD. It is worth pointing out that the average total of TPS in ProSoCaD (dotted red line) is the sum of number of valid videos (valid TPS) plus number of invalid videos (invalid TPS). In other words, the average total number of TPS in SecDUB achieves the fewest number of TPS (dotted blue line), due to the fact that SecDUB takes into account only valid TPS, because SecDUB avoid invalid TPS. Therefore, the number of invalid TPS in ProSoCaD can be calculated diminishing the average total of TPS in ProSoCaD by the average total of TPS in SecDuB. Hence, the fewest transactions aid to diminish slightly the end-to-end latency at the most scenarios in SecDUB, despite of the overhead incurred by SecDUB.

Even though ProSoCaD increases the throughput and TPS, a percentage of transmitted video data traffic is invalid. It can be noticed that the invalid TPS (i.e. difference between valid TPS and total TPS) increases as the number of nodes increases. For instance, when ProSoCaD is used, there are 10 and 15 invalid TPS in the scenarios with 15 and 55 nodes, respectively. It is worth pointing out that the offload achieved by ProSoCaD can be significantly harmed by these invalid TPS. Therefore, the network bandwidth is wasted due to transmission of non-useful video traffic in ProSoCaD. SecDUB enables

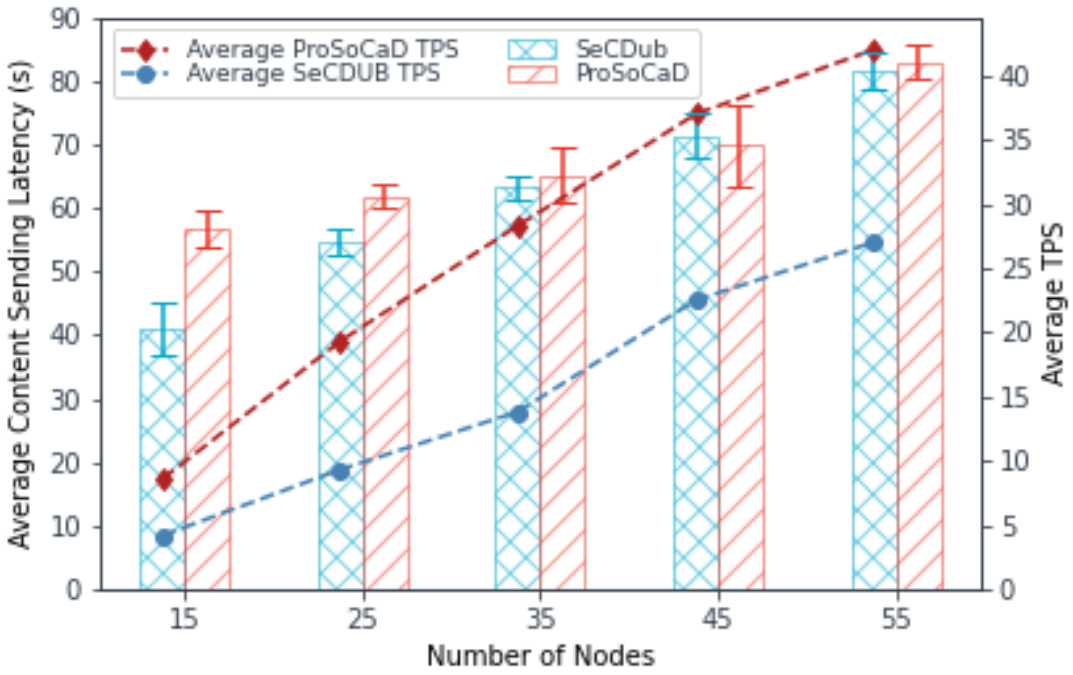


Figure 4.7: Average TPS and latency

to improve the goodput and latency as well as to achieve higher serviceable offloading, since that network bandwidth wasted with invalid videos can be used for valid videos. Besides, the clustering scheme divides the entire D2D network in smaller groups, named clusters. Each cluster carries out a separated and independent instance of consensus with a small number of nodes (average 15 nodes by cluster in a network scenario with 55 nodes), less overhead and less consensus time than a flat network. Therefore, the impact of increase in the number of transactions on the video content latency or consensus time added by the SecDUB are mitigated by the clustering scheme.

## 4.4 Analysis of TrustMD Network Performance

We assess the impact of TrustMD approach on the D2D communication in data and control plane. As it was done to SecDUB assessment (section 4.3), with TrustMD we seek to achieve higher trust levels with low computation costs and no significant traffic deterioration achieving considerably great balance between security and performance. To assess these results, we compare TrustMD approach with SecDUB at the same simulation parameters. We evaluated the proposed approach within a stressful scenario with 50% of malicious nodes [55, 66], where malicious nodes send only invalid videos during the entire simulation time.

To assess TrustMD scenarios, the inter-domain communication were implemented by enabling two different domains. Since NS-3 does not provide domain handover protocol or service in Lena module, the domain handover was carried out in simulation as follows: the Domain Chain procedure is performed by triggering the Inter Domain Trust Update Flow to update the trust information at the domain level. The update frequency $\Delta_{Intra-TUF}$ of all CHs in environment was set to 15 updates a minute, as the higher assessed update frequency rate, which implicates in better security performance of TrustMD (see Section 4.6).

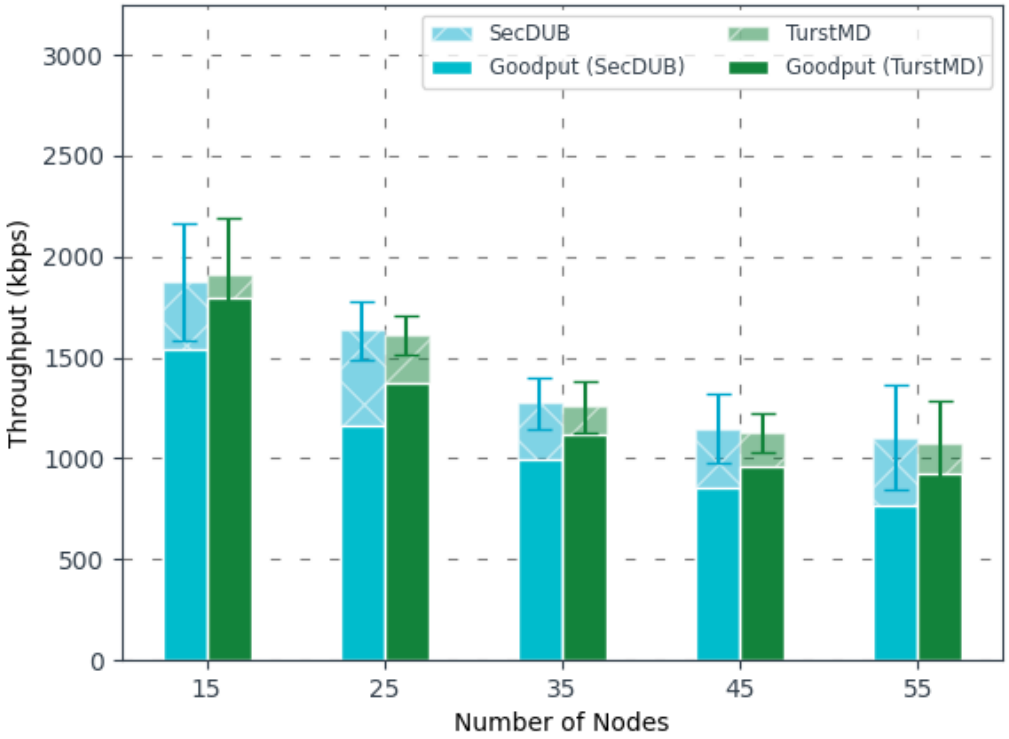


Figure 4.8: Average Throughput and Goodput of SecDUB and TrustMD

With the help of NS-3 we assessed throughput, goodput, packet loss rate and overhead to understand TrustMD network performance. We performed load tests using Hyperledger Caliper [13] with varying settings of transaction loads and varying number of blockchain peers to evaluate blockchain network performance.

We measured TrustMD network performance by varying the number of nodes and assessing how it impacts the scalability of the network with different degrees of node density, when compared with SecDUB. We can notice by the Figure 4.8 and Figure 4.9 that the throughput and Packet Loss Rate do not change considerably when compared

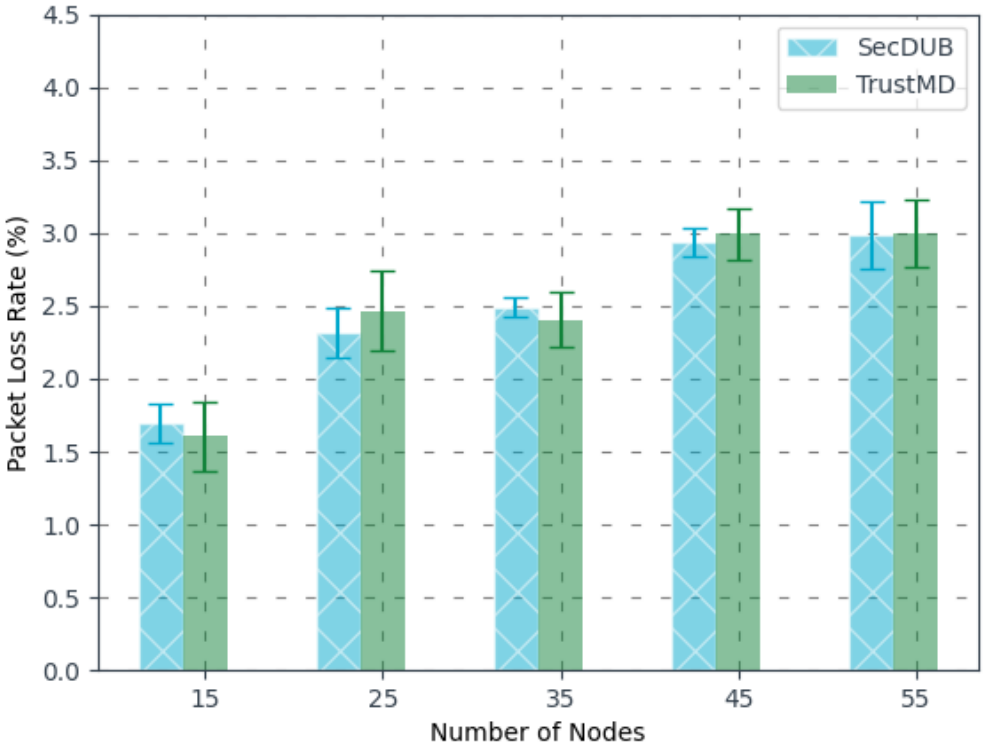


Figure 4.9: Average Packet Loss Rate of SecDUB and TrustMD

TrustMD with SecDUB. These results are an indication that TrustMD does not cause D2D network degradation.

Goodput is considerably higher when compared to SecDUB, achieving values close to 95% of goodput for the lowest density (15 nodes) scenarios and close to 85% on the other scenarios (moderate and higher density networks), representing an average increasing of 11% in goodbput, comparing it with SecDUB. These results indicate that TrustMD was successful in improving significantly the system's utility by increasing the goodput through the diminishing of the sharing of invalid video in the D2D communication.

We can see that there is a brief decrease in the percentage difference between the goodput between TrustMD and SecDUB in denser scenarios. This behavior can be attributed to the inherent challenges of trust establishment and information dissemination in networks with greater number of nodes. As the number of nodes and users increases, the complexity of trust management also escalates. In such environments, there is a higher likelihood of encountering nodes with no prior interactions or trust history, leading to a prevalence of neutral trust degrees. The presence of a significant number of users with a neutral degree of trust creates a window of opportunity for attackers to exploit

this uncertainty and launch malicious attacks before TrustMD can effectively distribute trust information. These early attacks can result in compromised interactions and reduced goodput. Despite the decrease in the difference in goodput between these two approaches, it is important to note that TrustMD constantly enhance goodput on all assessed scenarios.

Trust in Multiple Domains results on increasing device goodput and slightly increasing packet loss rate. We were able to achieve 95% of goodput in relation to the total throughput when the trust information is shared with upper layer chains. It is also important to note how the transposition of the overhead from the device layer to the domain layer not only increased the goodput but maintained the network quality levels while preserving the packet loss rate, which achieved an increasing average of approximately 11% for goodput and increasing average slightly of 1% for packet loss rate.

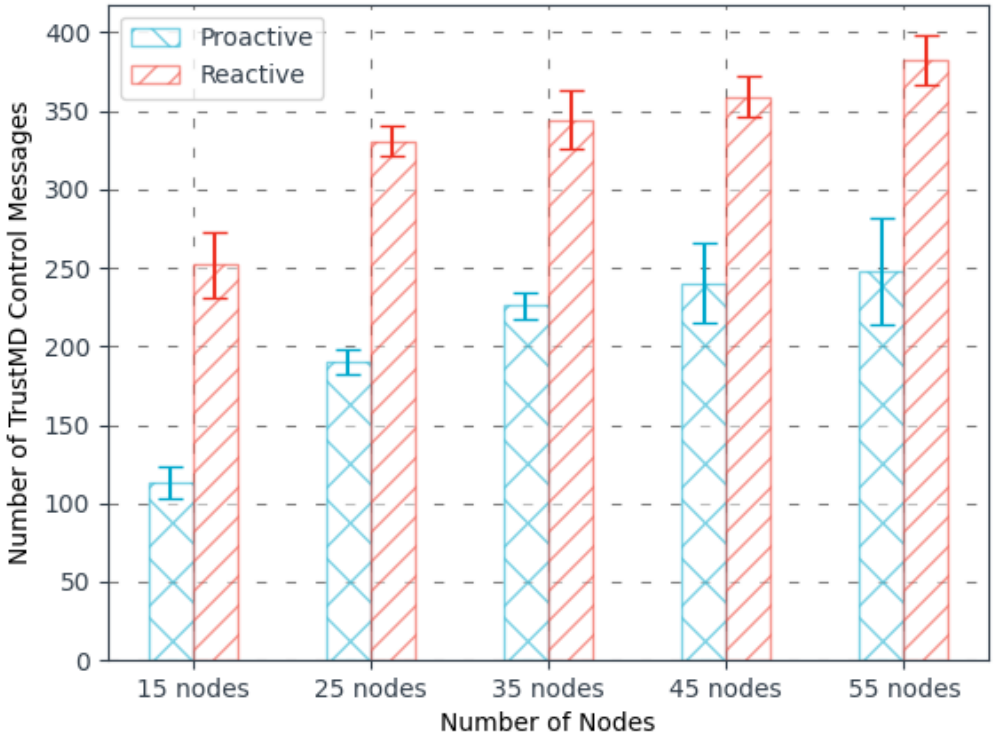


Figure 4.10: TrustMD overhead

With the data highlighted in graphs 4.8 and 4.9 it can be noticed that combining edge trust information storage with blockchain and distributed management in a multi layer architecture, we could safely distribute information across distinct nodes in network different domains and thus distribute trust information to a broader area. Results of

goodput may also be reflected on what is observed on the security performance analysis of Trustmd approach in Section 4.6.

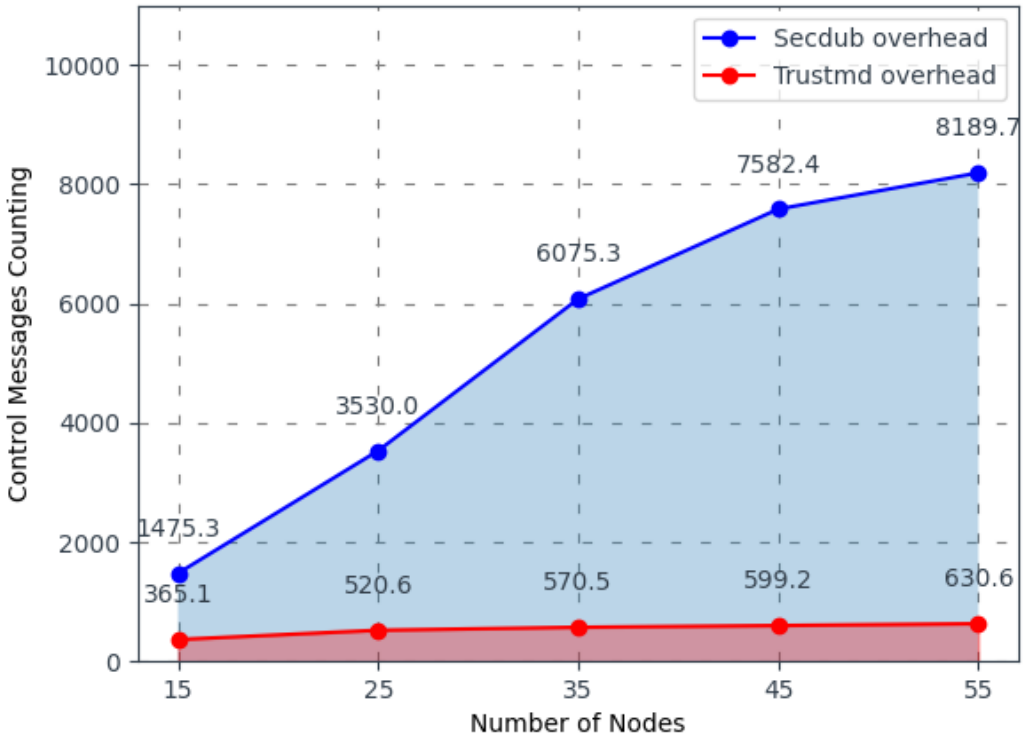


Figure 4.11: Comparing with SecDUB

Figure 4.10 highlights the overhead that TrustMD imposes on the device layer, considering a fixed $\Delta_{Intra-TUF}$ to all CHs in environment. Here the overhead is classified between *proactive* and *reactive* overhead. This classification is important to understand the actual burden TrustMD imposes to the overall content delivery network performance. In that way, *proactive* overhead refers to the control traffic generated by operations in which an UE acts by causing change and not only reacting to a change when it happens. *Reactive* overhead describes the one generated by operations in which an UE acts by reacting to a change or event. In TrustMD a *proactive* overhead is strictly related to Intra Domain Trust Update Flow (Section 3.2.3) when a CH requests a proactive trust update in Edge Chain. On the other way, *reactive* overhead refers to the overhead strictly related to Trust Query Flow when a UE requests a trust decision upon a node with neutral trust value UE during the sharing of a content.

As we can see by the results, the *reactive* overhead reassembles more burden to the device layer than the *proactive* one. This can be explained by the inherent nature of each associated TrustMD operation. *Proactive* TrustMD operations are generated

on a fixed rate ($\Delta_{Intra-TUF}$) and in a limited set of Cluster Heads. On the other hand, *reactive* operations are generated by the extent of content sharing, depending on each node behavior in the network. With that said it is possible to correlate *reactive* overhead with the number of CHs and the size of each cluster in SecDUB.

When the number of node increases it is expected higher number of TrustMD TQF calls, but in 35 and 45 nodes scenario we can observe a slightly slow growth. As the higher is trust distribution on the device layer itself due to the nodes interaction, the lesser number of queries to the upper layers (MEH, DCO). As we stated before, a higher density of nodes does not result in an increase in the number of clusters, in that way, each node knows trust information (indirect and direct trust) more frequently with the help of SecDUB.

Hence, the incurred overhead added by TrustMD is small compared to SecDUB, which represents 7% of the incurred SecDUB overhead on average. In Figure 4.11 we highlight the total overhead (*proactive* + *reactive*) of Trust in Multiple Domains proposal and compared it with SecDUB. As we can see, TrustMD overhead in D2D communication is negligible compared with SecDUB overhead. We highlight the importance of TrustMD's low overhead when crossing this data with throughput, goodput and packet loss rate values shown in figures 4.8 and 4.9. With low overhead, D2D layer is able to operate without performance disruption but still benefit from TrustMD's trust information distribution power.

Figure 4.12 illustrates the latency, representing the amount of time taken to transmit video content from the source to the destination. Notably, both TrustMD and SecDUB demonstrated comparable latency results across all the evaluated scenarios. However, in sparse scenarios, TrustMD showcased a slight reduction in latency, while in dense scenarios, there was a marginal increase. On average, TrustMD exhibits a slightly decrease of 1.3*s* in latency, comparing with SecDUB.This discrepancy can be attributed to the growing *reactive* overhead, arising from the escalating interactions among D2D nodes in dense environments.

Despite the slightly higher latency observed in dense scenarios with TrustMD, it is crucial to emphasize the framework's paramount focus on ensuring secure communication. By effectively preventing the propagation of malicious content, TrustMD safeguards the network from potential security threats while keeping the impact on latency and overhead at a manageable level. This ability to maintain secure communication in dynamic and dense scenarios is of utmost importance, as it directly contributes to fostering trust among users and devices in multi-domain environments.

When it comes to the blockchain performance we evaluated the blockchain network using Hyperledger Caliper and varying the number of peers in each simulation. Supported by the Linux Foundation, Caliper is a tool that is used as a performance

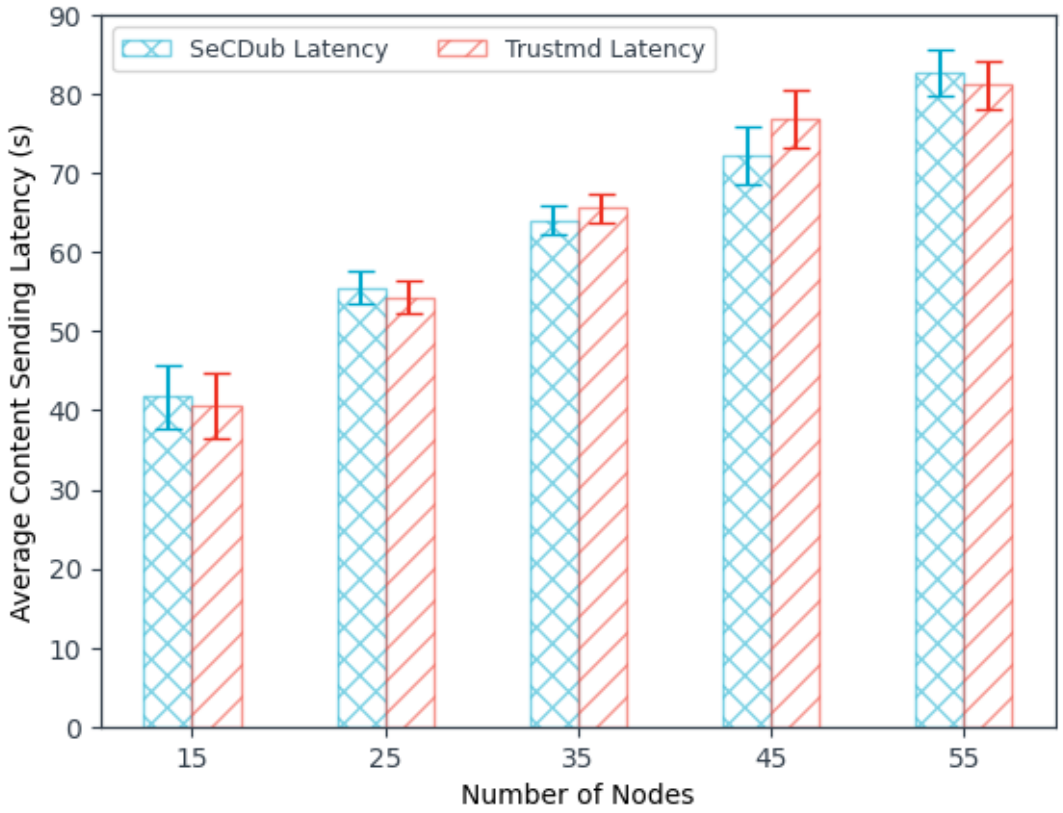


Figure 4.12: TrustMD Average Latency

benchmark framework for permissioned blockchains. The tool allows for testing different blockchains with similar environments. It is able to track metrics such as throughput, latency and success rate. This is done by listening to transaction timestamps and then calculating the metrics based on those timestamps [13].

Throughput is measured as Transaction per Second (TPS) and is how fast transactions are committed to the ledger successfully. Latency is measured in seconds and it corresponds to the amount of time between transactions being sent and them being received. The rate at which transactions are created in the blockchain system is a key factor for performance tests.

It may be desired to send transactions at a specified rate or follow a specified profile. In our case we utilized a preset Caliper run profile, called fixed-load. This run profile aims to maintain a defined set of transactions within the system by modifying the driven TPS. The result is the maximum possible TPS for the system whilst maintaining the pending transaction load.

To measure these metrics and understand TrustMD chaincode efficiency, we vary the number of blockchain peers and transaction load in each simulation during 60 seconds of test. Transaction load vary between 300, 400, 500 and 600 transactions. We ran tests

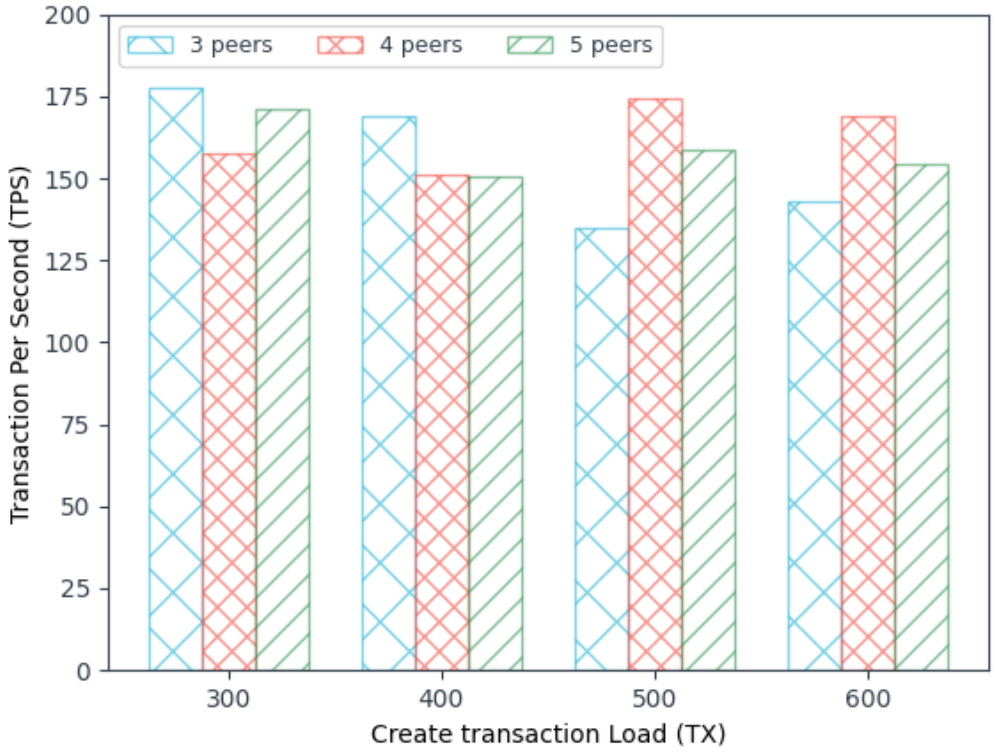


Figure 4.13: Throughput of Update Chaincode Transactions

with those configuration with topology variation between 3, 4 and 5 peers. Here we highlight average latency and throughput of the hyperledger trustmd chaincode. We ran these scenarios for two fundamental chaincode functions: *Update* and *Query*.

*Update* refers to the operation in which peers send transactions designated to update/create a trust asset within its ledger. This operation is present in Intra Domain Trust Update Flow (section 3.2.3), but also in Inter Domain Trust Update Flow (section 3.2.4). *Query* refers to operations in which peers send transactions specifically designated to get a trust asset within a ledger, based on the asset index. These operations are present mainly on Trust Query Flow(section 3.2.5)

Greater the latency of *Query* operations in Edge Chain, greater the waiting time to determine a node trustworthiness and consequently, worse the network throughput and goodput. Higher is *Query* latency in Domain Chain, worse is the distribution of trust information and therefore worse the goodput. The same logic can be replicated to the network security performance since longer it takes to distribute information, more nodes can be harmed by malicious nodes. It is also true for *Update* transactions, since higher is the transaction latency, lower is the trust information distribution rate and consequently the network will be more susceptible to attacks.

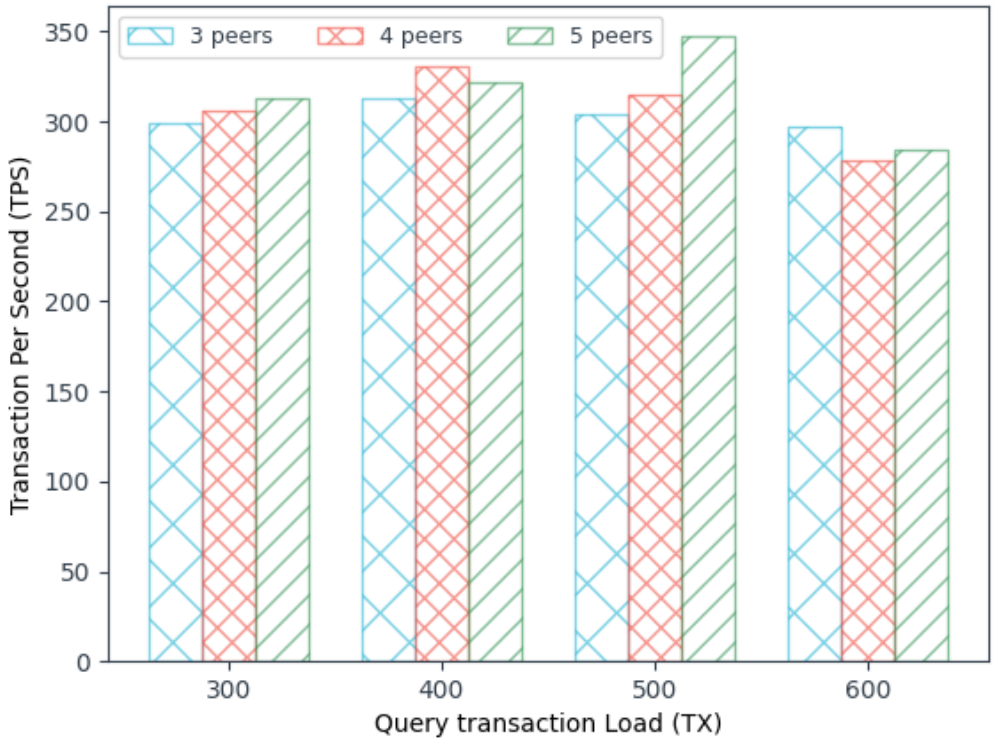


Figure 4.14: Throughput of Query Chaincode Transactions

In Figure 4.13 we can observe that we had a peak of approximately 175 TPS, with 140 TPS in average for all assessed scenarios of *Update* transactions. Considering TrustMD use case described in section 3.2, this load test describes a highly stressed scenario. TrustMD operations occurs in a fixed/controlled rate (Section 3.2.3), during content sharing (Section section 3.2.5) or during handover (Section 3.2.4) and give those limitations, the assessed tests corresponds to a highly stressed scenario.

TrustMD *Update* transactions are originated proactively with the fixed periodic update (Section 3.2.3) in Intra Domain scenario or reactively in the event of an inter domain handover (Section 3.2.4), indicating that even in stressed scenarios TrustMD presented good results. In Figure 4.14 similar analysis can be applied in which we observed that *Query* operation achieved a peak of almost 350 TPS and approximately 300 TPS in average. Query transactions are generated during Trust Query Flow which originates from reactive requests made by UEs when a new UE neighbor appears (see Section 3.2.5).

When it comes to latency, for *Update* transactions, in Figure 4.15 we observe that latency increase linearly with increasing transaction load, achieving a peak of approximately 2.5 seconds in the worst scenario and 1.0 seconds on the best case scenario.

In Figure 4.16 we observe similar behavior for *Query* transactions with a peak latency of approximately 1.4 seconds in worst case and 0.5 seconds in the best case scenario. Performing stressed load tests in the blockchain network showed that TrustMD chaincode was capable to sustain latency below 3 seconds during *Update* transactions and below 2 seconds during *Query* transactions with the highest assessed transaction load. This results explains how TrustMD didn't considerable harm the overall content distribution mechanism and/or impact SecDUB performance.

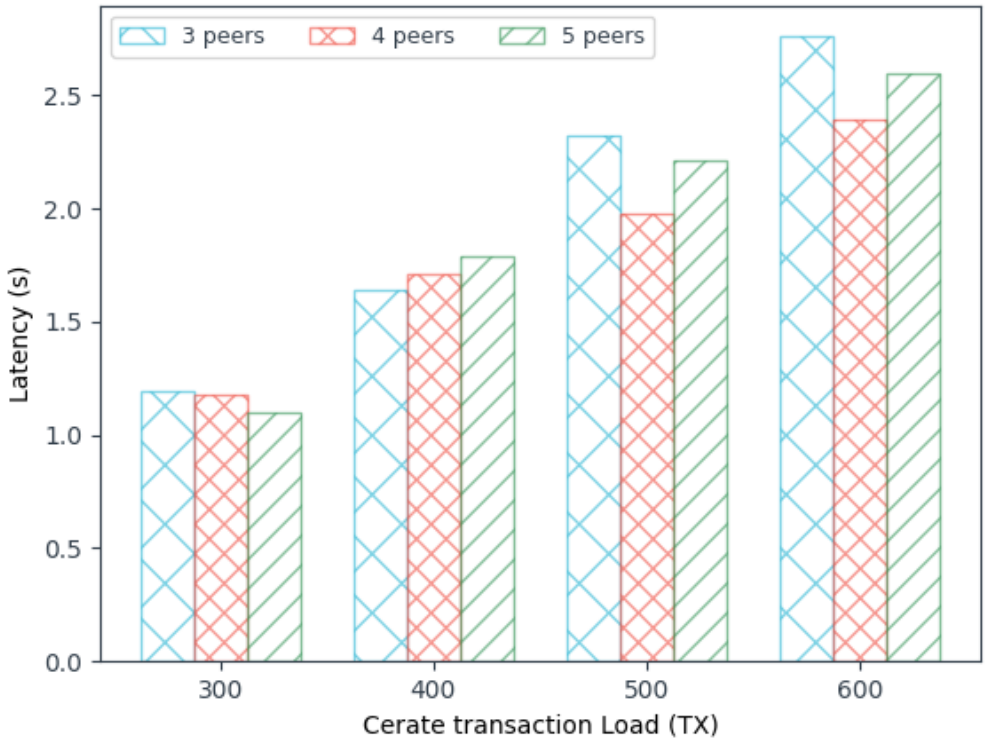


Figure 4.15: Latency of Update Chaincode Transactions

As we can see in Figures 4.13 and 4.14 *Update* transactions presents lower overall TPS than *Query* transactions. This is an expected result, since *Update* transactions encompasses a sequence of operations that characterize a procedure of greater complexity than *Query* transactions and by consequence has higher latency and lower TPS. Even with relatively higher latency, *Update* operations do not impact the critical flow of the proposal, that is, there is no direct impact on the content distribution. The situation is opposite when we consider *Query* transactions, since its latency has direct impact on content distribution.

In summary, the success of TrustMD depends on the combination of all the factors presented. So TrustMD must be able to operate with low overhead, and high

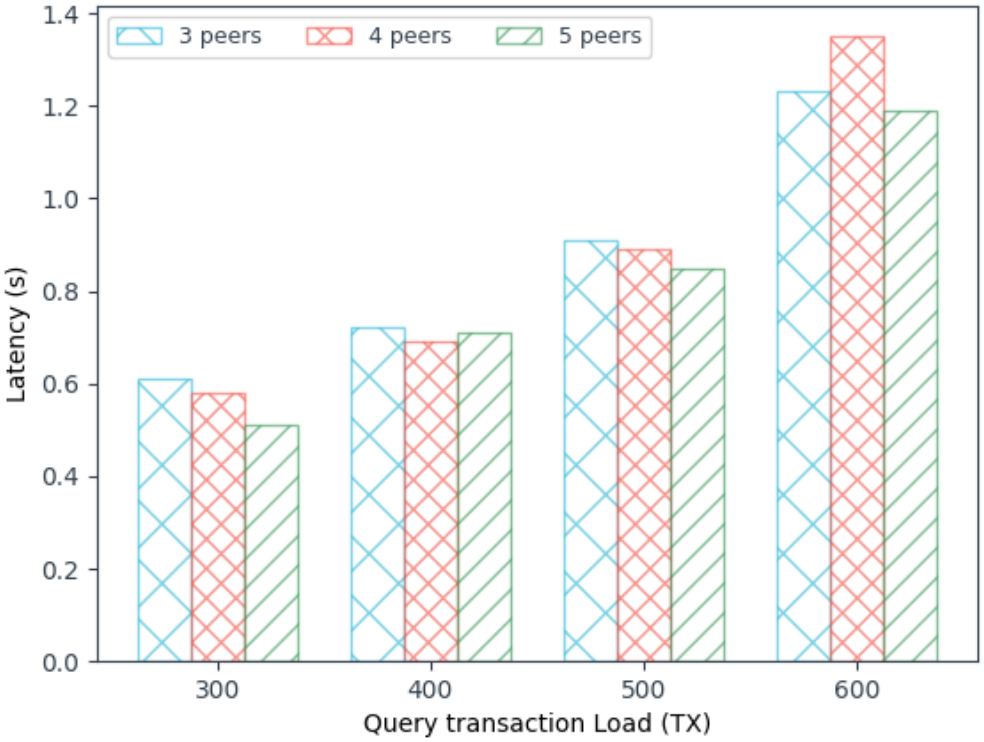


Figure 4.16: Latency of Query Chaincode Transactions

chaincode efficiency, on both edge and domain layers to achieve the higher goodput. As we can see, the chain performance was assessed against an stressed scenario, with high transaction load. To achieve the transaction execution rate presented by the executed caliper tests in a real mobility scenario, would be required high density of nodes with high transaction rate in an extremely high mobility scenario. Furthermore, there might be scalability problems concerning storage capacity. The implementation of off-chain storage mechanism in the Mobile Edge Computing (MEC) layer, enabled by Hyperledger Fabric. This approach considerably improved data query efficiency and mitigate potential scalability issues. Therefore the chaincode efficiency results confirms that TrustMD was able to scale both in number of nodes and in transaction rate.

## 4.5 Analysis of SecDUB security performance

In this section, we evaluated the efficiency of the collaborative trust management model in face of different malicious nodes behaviors. In the previous experiments the malicious users only behaved maliciously, however in a real scenario nodes may behave differently over time. In the context of trust mechanisms, malicious users can perform

attacks by intentionally varying their behavior in order to elevate their trust values among proximate users. Good trust strategies are able to perform well even in face of that type of behaving and avoid the effectiveness of malicious actions. To assess the proposed trust mechanism through the presence of such disturbances, we evaluate our approach using one form of behavior variation called *On-Off attack* in which nodes shall manipulate the trust mechanism individually, by intentionally altering their behavior [8]. The nodes randomly provide good and bad services in order to avoid the risk of being labeled as malicious.

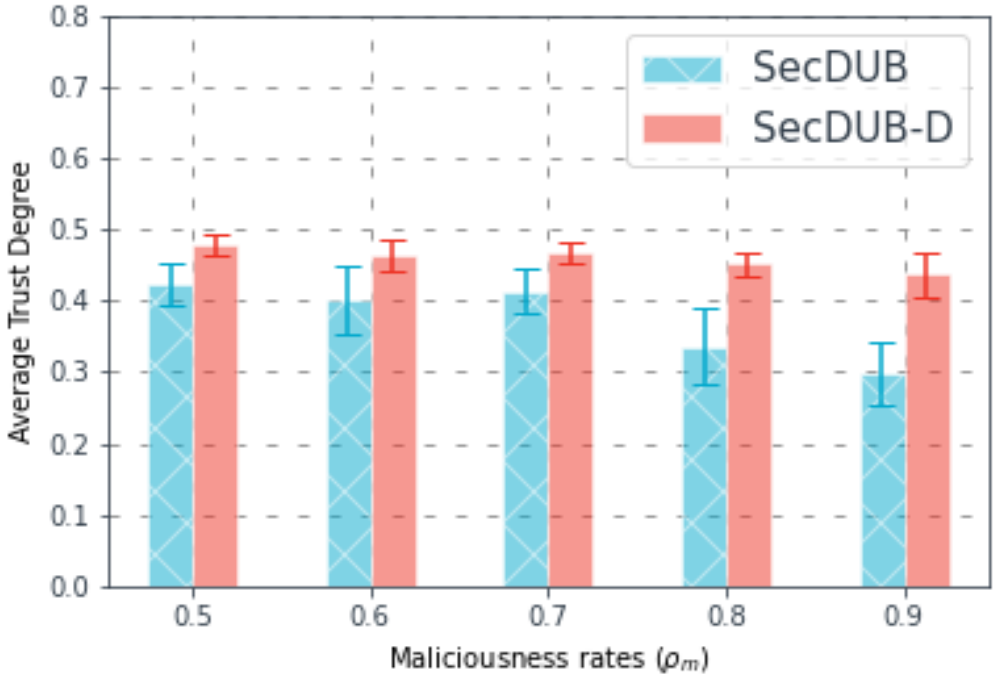


Figure 4.17: Average trust degree of malicious nodes

It is important to point out that, to the best of our knowledge, most of the works that take into account a collaborative trust model do not assess its performance [57, 9, 25, 3, 43, 39] in face of trust mechanism attacks. In each simulation, we configure a set of malicious nodes, corresponding to the group of users responsible for sharing invalid content. For all the experiments in this section, we set up a scenario with 27 users of the D2D network corresponding to the configuration pattern of [17].

We employed two variants of our proposal: (1) SecDUB-D - which employ only the trust through direct observations and (2) SecDUB - which employ trust through direct and indirect observations. We applied FNR and ADT assessment metrics while varying malicious nodes behavior. The following paragraphs detail and discuss the results for each one of the scenarios described.

To simulate this attack, we have defined different levels of maliciousness ($\rho_m$) that model the simulation scenario with the on/off attack, which limit the likelihood that malicious nodes send invalid content with each interaction. That is, with $\rho_m = 0.6$, the probability that malicious nodes will transmit invalid content is 60% with each interaction. In the tests performed, we used maliciousness rates ranging from 0.5 to 0.9. With this configuration, we were able to simulate different scenarios of random behavior and compare the performance of the approaches in different perspectives of randomness. The results are illustrated in Figures 4.18 and 4.17.

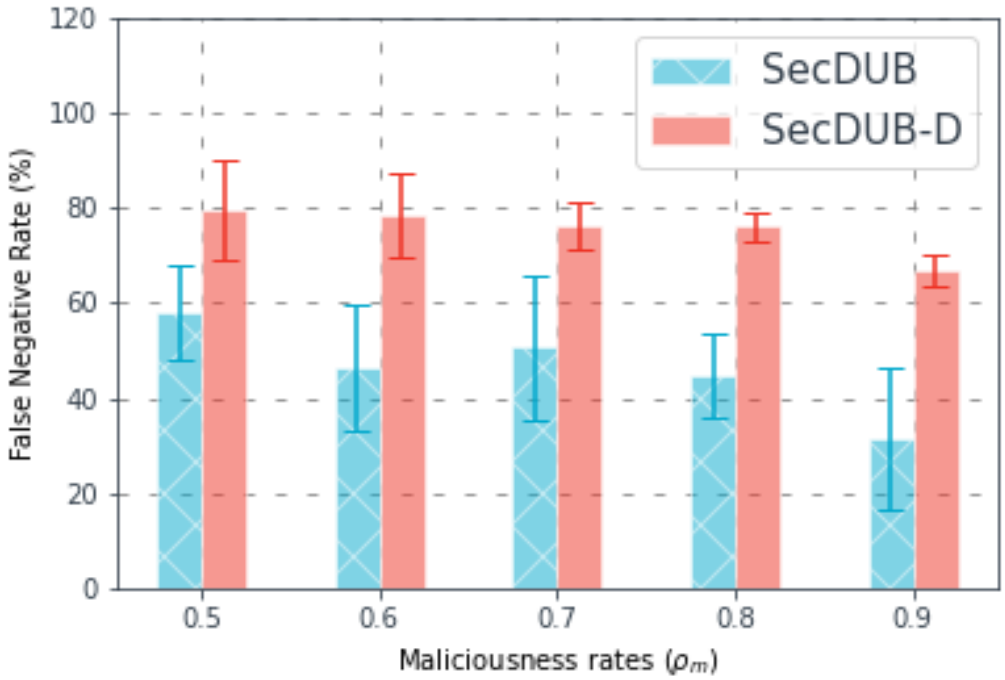


Figure 4.18: False negative rate

Through Figures 4.17 and 4.18, we can see the comparison between SecDUB and SecDUB-D, given the average degree of trust of the malicious nodes and the rate of false negatives, respectively, varying according to $\rho_m$. We observed that the random malicious behavior was more impacting in the approach without trust by indirect observations (SecDUB-D), considering that with $\rho_m = 0.5$ the average degree of trust of the malicious nodes remained close to 0.5, showing a higher level of uncertainty when comparing the complete SecDUB approach. However, the different levels of randomness had a considerable impact on SecDUB, since the false negative rate grew by an average of 45% between $\rho_m = [0.5, 0.9]$. When we compare these approaches, it is noticeable the importance of observation distribution on uncertainty assessment, because even when the difference in the average degree of trust of malicious nodes in SecDUB and SecDUB-D is very small,

SecDUB was able to reduce the rate of false negatives much more significantly. In this scenario, the difference between the SecDUB and SecDUB-D degrees of trust is approximately 0.05, but the difference between false negative rates is approximately 45%.

## 4.6 Analysis of TrustMD security performance

To assess TrustMD security performance we utilize similar configuration of what was proposed in SecDUB assessment (section 4.5). We compared the complete SecDUB approach, of direct and indirect trust, with results obtained from TrustMD implementation. To assess TrustMD security results we evaluated its performance considering different update rates ($\Delta_{Intra-TUF}$). The main objective of these tests is understand the influence of the update frequency rate on trust information distribution and attack mitigation.

The hypothesis considers that a higher update rate indicates greater trust efficiency and consequently improves security performance. Ideally as higher the update frequency lower the false negative rate, hence better the trustworthiness distribution across the network.

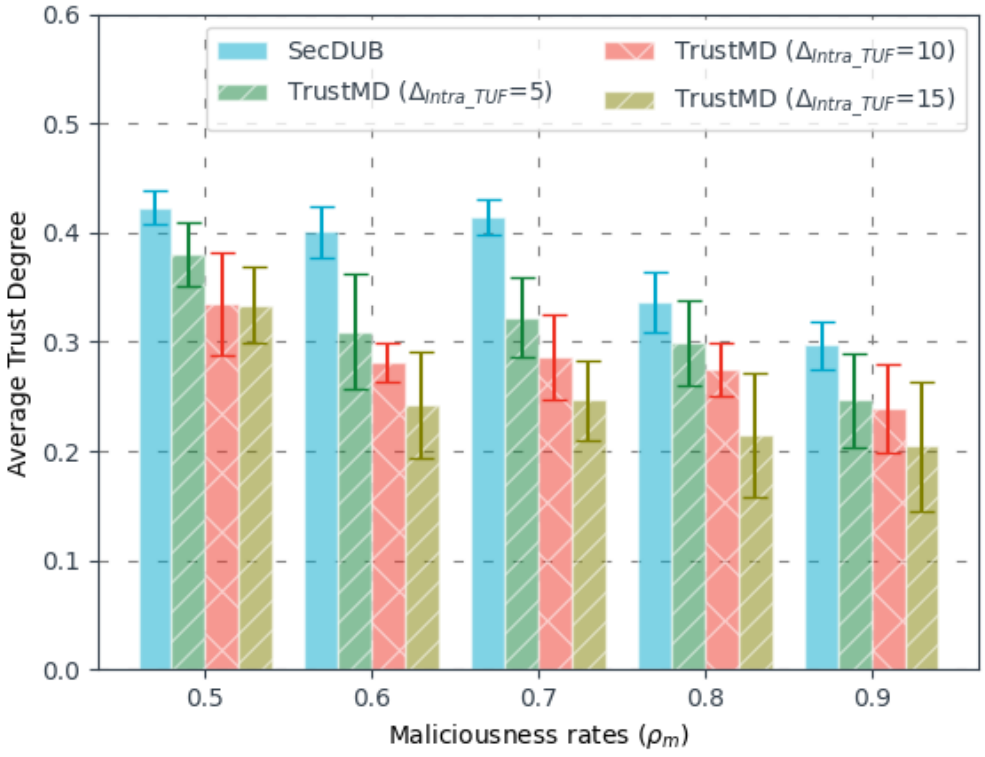


Figure 4.19: Average trust degree of malicious nodes

To assess Trust in Multiple Domains security performance, we also considered False Negative Rate and Average Degree of Trust. As it was done for SecDUB (section 4.5), we evaluated TrustMD using *On-Off attack* [8], in which nodes randomly provide good and bad services in order to avoid being labeled as malicious.

The first metric assessed here is the Average Degree of Trust of nodes considering the comparison of TrustMD approach with SecDUB, using On-Off behavior and varying the level of maliciousness $\rho_m$ (refers to section 4.5). The difficulty of a trust model to accurately identify a malicious node when employing random malicious behavior adhere to the fact that probabilistic models like SecDUB rely on collecting evidence to model user behavior. With contradictory or mixed evidence, greater uncertainty and therefore more difficult to predict which users are malicious due to behavioral variance.

As we can see in Figures 4.19 and 4.20 SecDUB shows higher level of uncertainty when comparing the complete SecDUB approach, as TrustMD helps to better distribute trust information among greater sets of nodes. However, the different levels of randomness had a considerable impact on TrustMD as it had on SecDUB, since the false negative rate grows in an inversely proportional rate considering the assessed interval $\rho_m = [0.5, 0.9]$. Also is worth mention that these results highlight how TrustMD manages to decrease the aggregate trust value as well as false negatives by sharing trust information between each MEC Host and different domains.

As we can see on the assessed metrics, TrustMD operated better on scenarios with higher frequency update configurations. With $\rho_m = 0.5$, $\Delta_{Intra-TUF} = 5$ configuration presented a decrease of approximately 22% on device trust value, while $\Delta_{Intra-TUF} = 10$ and $\Delta_{Intra-TUF} = 15$ configurations of Trust in Multiple Domains, decreased trust values on approximately 35% when comparing with SecDUB. On the most malicious scenario $\rho_m = 0.9$, the device trust value decreased approximately 18% on $\Delta_{Intra-TUF} = 5$ and approximately 36% on $\Delta_{Intra-TUF} = 15$ scenarios. Regarding the false negative rate, higher trust update configurations presented better results in average, with the exception of $\rho_m = 0.5$ scenario, in which $\Delta_{Intra-TUF} = 10$ scenario presented a decrease of 25% on false negative rate and $\Delta_{Intra-TUF} = 15$ configurations presented a decrease of 31% when comparing with SecDUB.

In conclusion, when we comparing the security performance of these approaches in terms of trust modeling, it is noticeable the importance of sharing trust information among the edge. Beyond distributing trust among different domains, TrustMD helps to start the network with different level of security, since once a D2D node doesn't know your neighbor trustworthiness, it can get this data by querying edge data directly with the Mobile Edge Computing Server.

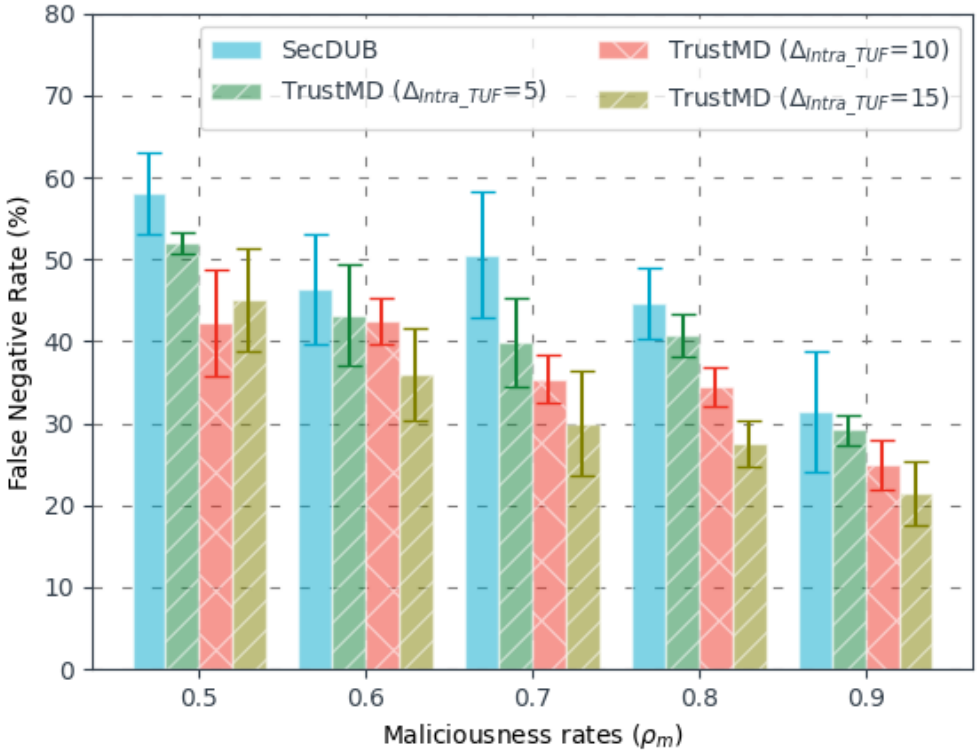


Figure 4.20: False negative rate

CHAPTER 5

# Conclusion and Future Work

This chapter presents the final considerations about the work, highlighting highlights and low lights of the work and final thoughts upon future work and directions of this research.

## 5.1 Conclusion

The current dissertation, which is based on studies/results and the systematic literature review, confirms the hypothesis that the secure trustworthiness data distribution among different domains is an effective way of mitigating malicious acts towards D2D caching. Thinking on a two way step approach, we subdivided the proposal execution between two trust models that work between three layers: Device Layer, Edge Layer and Domain Control Layer. SecDUB framework is proposed for the device layer as a way of leveraging trustworthiness to Device–to-Device communication nodes, by using trust on the basis of behavioral criteria, in a way to shape users behavior pattern in network and predict the likelihood to engage in malicious activities. Across the Edge and Domain Control layers we leverage a secure distribution mechanism called TrustMD, enabling multi-domain trust control information distribution in the network edge.

SecDUB is a collaborative trust model that aims to mitigate the transmission of invalid content, through the collection of indirect and direct observations. In addition, blockchain concepts were adapted to the dynamic and restricted scenario of the D2D communication to avoid the modification of indirect observations. Essentially, the evaluation assessed two distinct scopes: network performance with/without the security framework and the trust mechanism performance when is under attack of a high number of malicious nodes. The architecture was tested in simulated environment considering different evaluation scenarios.

In terms of network performance, we varied the number of nodes between dense and sparse scenarios comparing the proposed solution performance with ProSoCaD [17] D2D caching baseline model. We noticed the average throughput flow was lower for most of the configurations varying the number of nodes when SecDUB is used, but it

is similar with dense scenarios. However, the average goodput of our proposal is greater than ProSoCaD, that is, despite decreasing the throughput, the useful service is higher in our proposal, which demonstrates the benefits of the collaborative trust management model, for example, lesser wasting of network bandwidth and higher useful offloading. In addition, the impact on latency is low. Furthermore, the overhead aggregated by SeCDuB is not increased linearly with the increase of the number of nodes, which is closely related to the proposed clustering scheme. In this context, the current approach achieves better results in dense scenarios, considering that the difference between average throughput and packet loss decreased with an increase of the number of nodes. Hence the proposed clustering scheme helps to mitigate the impact of SecDUB on traffic performance and overhead. Furthermore, the clustering helps to provide an agile and lightweight blockchain with a non-significant average consensus time.

Regarding the security mechanism performance, we compare the SecDUB to its version without using of trust by indirect observations, varying the malicious nodes behavior. In this assessment, we observed that indirect trust improved the overall system efficiency, by decreasing the false negative rate and the average degree of trust of malicious nodes. To assess the trust mechanism performance, we simulated scenarios where the nodes change behaviors individually.

Trust in Multiple Domains serves as a secure distributed framework with the primary purpose of safely storing trust information among nodes across different clusters, network edges and domains, thus facilitating the dissemination of trust information over a wider area. By employing this approach, the framework effectively circumvents high latency issues at the edge, paving the way for secure collaborative communication within the Mobile Edge Computing Layer. To enable seamless TrustMD operations, blockchain technology was integrated into the edge infrastructure, spanning both the Domain and MEC layers. This integration leverages blockchain's capability to accommodate control information from the control plane and promotes on-chain scalability, enabling efficient cross-chain edge data sharing [4].

In terms of network performance, we varied the number of nodes between dense and sparse scenarios comparing the proposed solution performance with SecDUB as our baseline. The results indicates that TrustMD does not cause D2D network degradation and in addition, goodput is considerably higher when compared to SecDUB. In that way, TrustMD was successful in improving significantly the system's utility by increasing the goodput by diminishing the sharing of invalid video in the D2D communication. The incurred overhead of TrustMD is small compared to SecDUB and does not harm the traffic performance in device layer. With the low overhead, D2D layer is able to operate without performance disruption but still benefit from TrustMD's trust information distribution potential.

The usage of Hyperledger blockchain distributed across domain and edge layers proved to be a good approach to spread trust information on different levels of the network. We evaluated blockchain performance using Hyperledger Caliper by varying the number of peers in each simulation [13]. On-chain performance was assessed with an stressed scenario, with higher transaction load than a normal mobile environment. Blockchain efficiency results contributed to overall content distribution mechanism and/or impact SecDUB performance while still increases goodput.

Regarding the effectiveness of TrustMD trustworthiness mechanism, we conducted a comparative analysis with Secure D2D caching based on Trust Management. Notably, TrustMD not only facilitates the distribution of trust across different domains but also enhances security while keeping device-level overhead at a minimum. The resilience of TrustMD was evident in the context of inter-domain handover involving a malicious node, as the multi-domain distribution of nodes' trustworthiness data throughout the network proved non-harmful. Leveraging the upper-level chain in the Domain Layer, TrustMD enabled access to trust control data from other domains, harnessing the high availability power of Mobile Edge Computing to make more informed trust decisions with a lower false negative rate at the Device–to-Device communication layer.

The integration of Secure D2D caching based on Trust Management (SecDUB) with Trust in Multiple Domains (TrustMD) yielded interesting advancements in secure Device–to-Device communication content sharing. By combining the collaborative trust management approach of SecDUB with the efficient distributed edge storage mechanism of TrustMD, a promising solution emerged, significantly enhancing the security and privacy of users engaged in Device–to-Device communication communication. Our comprehensive analysis of the results revealed an increase on goodput, with performance reaching up to 95% in the best-case scenario presenting an average increase of approximately 11% comparing with SecDUB, consequently elevating the overall network quality. This achievement can be attributed to a marked decrease in the false negative rate, a direct consequence of the trust mechanism's assertiveness. It is worth highlighting that these notable results were attained harming slightly content sharing latency (decrease of 1.31%) and packet loss rate (1%), despite the additional overhead incurred. Notably, the overhead introduced by SecDUB pertains solely to maintaining clustering for information sharing, while TrustMD incur overhead through *proactive* updates ($\Delta_{Intra-TUF}$ ) and *reactive* interactions among nodes in content distribution, representing 7% of the average incurred SecDUB overhead. To conclude, the combination of SecDUB and TrustMD has proven to be an advantageous approach for enhancing edge security and securely disseminating trust information at the edge.

## 5.2 Future Work

Throughout the course of our research, we explored various potential directions for advancing the work, focusing on both the refinement of our proposal and the broader domain of study. The subsequent items delineate these promising directions:

(i) Optimizing the Consensus Protocol for Diverse Node Mobility Scenarios: One key direction involves the optimization of our consensus protocol, particularly in the context of nodes exhibiting varying patterns of mobility. This initiative seeks to enhance the robustness and adaptability of our protocol to accommodate different node movement scenarios.

(ii) Strengthening Trust Mechanisms to Mitigate Clusterhead Vulnerabilities: Another critical aspect involves fortifying our trust mechanisms to effectively prevent potential clusterhead vulnerabilities. This endeavor aims to enhance the security and integrity of the network by addressing vulnerabilities in clusterhead selection processes.

(iii) Performance Evaluation of Trust Mechanisms Against Diverse Attack Categories: To provide a comprehensive assessment of our trust mechanisms, it is imperative to rigorously evaluate their performance against a broader spectrum of attack categories, as exemplified in [8]. This evaluation will ensure that our security measures remain robust and effective in diverse threat scenarios.

(iv) Enhancing Decision Mechanisms via Threshold Optimization: The improvement of our decision-making mechanisms is a crucial area for development. By fine-tuning and optimizing the decision thresholds, we aim to enhance the precision and efficiency of our decision-making processes, leading to more effective network management.

(v) Enhancing Trust Mechanisms through Deep Reinforcement Learning on Edge Servers: An innovative avenue we intend to explore is the integration of Deep Reinforcement Learning within our trust mechanisms, particularly on edge servers. This novel approach can potentially elevate the trust mechanisms to a new level of sophistication, enabling dynamic adaptation and intelligent decision-making in real-time.

To implement the framework proposed in this dissertation in a real-world environment, D2D communication must be enabled on devices within the cellular network coverage. Devices located in this layer should be able to connect directly using routing protocols for ad hoc wireless mobile networks, such as OLSR [15]. Communication between a UE and a MEC server is outlined in [20]. However, to enable the proposed collaboration between MEC servers from different edge locations, it should be enabled to allow data distribution among servers within the same domain on the same blockchain network. Furthermore, the exchange of information between controllers from different domains should be enabled to make inter-domain information exchange possible. However, this type of sharing is not yet foreseen at the time of writing this dissertation.

These directions represent the forward-looking trajectory of our research, with the overarching goal of advancing the field of study while simultaneously refining our proposal to ensure its efficacy and resilience in dynamic and challenging environments.

# Bibliography


[1] ALI, K.; NGUYEN, H. X.; SHAH, P.; VIEN, Q.-T.; BHUVANASUNDARAM, N. **Architecture for public safety network using d2d communication**. In: *2016 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, p. 206–211, 2016.

[2] ALJERI, N.; BOUKERCHE, A. **Mobility management in 5g-enabled vehicular networks: Models, protocols, and classification**. *ACM Computing Surveys (CSUR)*, 53(5):1–35, 2020.

[3] ALNUMAY, W.; GHOSH, U.; CHATTERJEE, P. **A trust-based predictive model for mobile ad hoc network in internet of things**. *Sensors*, 19(6), 2019.

[4] BAI, F.; SHEN, T.; YU, Z.; ZENG, K.; GONG, B. **Trustworthy blockchain-empowered collaborative edge computing-as-a-service scheduling and data sharing in the iioe**. *IEEE Internet of Things Journal*, p. 1–1, 2021.

[5] BALDO, N.; MIOZZO, M.; REQUENA-ESTESO, M.; NIN-GUERRERO, J. **An open source product-oriented lte network simulator based on ns-3**. In: *Proceedings of the 14th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '11, p. 293–298, New York, NY, USA, 2011. Association for Computing Machinery.

[6] CACHIN, C.; OTHERS. **Architecture of the hyperledger blockchain fabric**. In: *Workshop on distributed cryptocurrencies and consensus ledgers*, volume 310, p. 1–4. Chicago, IL, 2016.

[7] CASTRO, M.; LISKOV, B. **Practical byzantine fault tolerance**. In: *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, OSDI '99, p. 173–186, USA, 1999. USENIX Association.

[8] CHAHAL, R. K.; KUMAR, N.; BATRA, S. **Trust management in social Internet of Things: A taxonomy, open issues, and challenges**. *Computer Communications*, 150:13–46, Jan. 2020.

[9] CHATTERJEE, P.; GHOSH, U.; SENGUPTA, I.; GHOSH, S. K. **A trust enhanced secure clustering framework for wireless ad hoc networks**. *Wireless Networks*, 20(7):1669–1684, 2014.

[10] CHAUHAN, S.; PANDA, N. K. **Chapter 7 - metadata**. In: Chauhan, S.; Panda, N. K., editors, *Hacking Web Intelligence*, p. 133 – 146. Syngress, Boston, 2015.

[11] CHEN, Y.; YANG, J.; TRAPPE, W.; MARTIN, R. P. **Detecting and localizing identity-based attacks in wireless and sensor networks**. *IEEE Transactions on Vehicular Technology*, 59(5):2418–2434, Jun 2010.

[12] CHO, J.-H.; SWAMI, A.; CHEN, R. **A survey on trust management for mobile ad hoc networks**. *IEEE Communications Surveys & Tutorials*, 13(4):562–583, 2011.

[13] CHOI, W.; HONG, J. W.-K. **Performance evaluation of ethereum private and testnet networks using hyperledger caliper**. In: *2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS)*, p. 325–329, 2021.

[14] CISCO, V. **Cisco annual internet report (2018–2023) white paper**. https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html, 2019. Acessado em: 03-03-2020.

[15] CLAUSEN, T.; JACQUET, P.; ADJIH, C.; LAOUITI, A.; MINET, P.; MUHLETHALER, P.; QAYYUM, A.; VIENNOT, L. **Optimized Link State Routing Protocol (OLSR)**, 2003. Network Working Group.

[16] CUI, M.; FEI, Y.; LIU, Y. **A survey on secure deployment of mobile services in edge computing**. *Security and Communication Networks*, 2021, 2021.

[17] D. S. MORAES, F.; CARDOSO, K. V.; BORGES, V. C. M. **Improving video content access with proactive d2d caching and online social networking**. In: *2017 IEEE Symposium on Computers and Communications (ISCC)*, p. 1043–1048, July 2017.

[18] DAI, Y.; XU, D.; ZHANG, K.; MAHARJAN, S.; ZHANG, Y. **Deep reinforcement learning and permissioned blockchain for content caching in vehicular edge computing and networks**. *IEEE Transactions on Vehicular Technology*, 69(4):4312–4324, 2020.

[19] DO, T.-X.; KIM, Y. **Control and data plane separation architecture for supporting multicast listeners over distributed mobility management**. *ICT Express*, 3(2):90–95, 2017. Special Issue on Patents, Standardization and Open Problems in ICT Practices.

[20] ETSI, G. **Mobile edge computing (mec); end to end mobility aspects**. *ETSI Standards Search*, 2017.

[21] ETSI, G.; OTHERS. **Multi-access edge computing (mec); phase 2: Use cases and requirements**. *ETSI Standards Search*, 2018.

[22] FILALI, A.; ABOUAOMAR, A.; CHERKAOUI, S.; KOBBANE, A.; GUIZANI, M. **Multi-access edge computing: A survey**. *IEEE Access*, 8:197017–197046, 2020.

[23] GARCIA, M. H. C.; MOLINA-GALAN, A.; BOBAN, M.; GOZALVEZ, J.; COLL-PERALES, B.; ŞAHIN, T.; KOUSARIDAS, A. **A tutorial on 5g nr v2x communications**. *IEEE Communications Surveys Tutorials*, 23(3):1972–2026, 2021.

[24] GREVE, F.; SAMPAIO, L.; ABIJAUDE, J.; COUTINHO, A.; VALCY, Í.; QUEIROZ, S. **Blockchain e a revolução do consenso sob demanda**. *SRBC*, p. 201–252, 2018.

[25] HE, Y.; YU, F. R.; ZHAO, N.; YIN, H. **Secure social networks in 5g systems with mobile edge computing, caching, and device-to-device communications**. *IEEE Wireless Communications*, 25(3):103–109, JUNE 2018.

[26] HONAR PAJOOH, H.; RASHID, M.; ALAM, F.; DEMIDENKO, S. **Hyperledger fabric blockchain for securing the edge internet of things**. *Sensors*, 21(2):359, 2021.

[27] IBM. **Hyperledger – open source blockchain technologies**.

[28] JAYABALAN, J.; JEYANTHI, N. **Scalable blockchain model using off-chain ipfs storage for healthcare data security and privacy**. *Journal of Parallel and Distributed Computing*, 164:152–167, 2022.

[29] JOHNSON, D.; MENEZES, A.; VANSTONE, S. **The elliptic curve digital signature algorithm (ecdsa)**. *International Journal of Information Security*, 1(1):36–63, Aug 2001.

[30] KIM, J.; KIM, D.; CHOI, S. **3gpp sa2 architecture and functions for 5g mobile communication system**. *ICT Express*, 3(1):1–8, 2017.

[31] KITCHENHAM.; CHARTERS, S. **Guidelines for performing systematic literature reviews in software engineering**. *EBSE Technical Report*, 2, 01 2007.

[32] KUDVA, S.; BADSHA, S.; SENGUPTA, S.; LA, H.; KHALIL, I.; ATIQUZZAMAN, M. **A scalable blockchain based trust management in vanet routing protocol**. *Journal of Parallel and Distributed Computing*, 152:144–156, 2021.

[33] KUMAR, S.; MISRA, S. **Joint content sharing and incentive mechanism for cache-enabled device-to-device networks**. *IEEE Transactions on Vehicular Technology*, 70(5):4993–5002, 2021.

[34] LEE, M.-C.; MOLISCH, A. F. **Caching policy and cooperation distance design for base station-assisted wireless d2d caching networks: Throughput and energy efficiency optimization and tradeoff**. *IEEE Transactions on Wireless Communications*, 17(11):7500–7514, 2018.

[35] LI, Q.; SUN, Y.; TIAN, T.; YANG, R.; MENG, L.; ZHANG, Y.; YU, F. R. **Research on Security of D2D Resource Sharing Based on Blockchain in Mobile Edge Network**. In: *2020 12th International Conference on Communication Software and Networks (ICCSN)*, p. 202–206, June 2020. ISSN: 2472-8489.

[36] MAAN, U.; CHABA, Y. **Accurate cluster head selection technique for software defined network in 5g vanet**. *Wireless Personal Communications*, 118(2):1271–1293, 2021.

[37] MCBRIDE, W. J.; MCCLELLAND, C. W. **Pert and the beta distribution**. *IEEE Transactions on Engineering Management*, EM-14(4):166–169, Dec 1967.

[38] MONIR, N.; TORAYA, M. M.; VLADYKO, A.; MUTHANNA, A.; TORAD, M. A.; EL-SAMIE, F. E. A.; ATEYA, A. A. **Seamless handover scheme for mec/sdn-based vehicular networks**. *Journal of Sensor and Actuator Networks*, 11(1):9, 2022.

[39] MOVAHEDI, Z.; HOSSEINI, Z. **T-D2D: A trust model for service offloading in device-to-device communication**. *Transactions on Emerging Telecommunications Technologies*, 30(10):e3686, 2019.

[40] NAKAMOTO, S. **Bitcoin: A peer-to-peer electronic cash system**. *Cryptography Mailing list at https://metzdowd.com*, 03 2009.

[41] PLATT, S.; SANABRIA-RUSSO, L.; OLIVER, M. **Conte: A core network temporal blockchain for 5g**. *Sensors*, 20(18), 2020.

[42] RAGHAV.; ANDOLA, N.; VENKATESAN, S.; VERMA, S. **Poewal: A lightweight consensus mechanism for blockchain in iot**. *Pervasive and Mobile Computing*, 69:101291, 2020.

[43] RAJU, L.; REDDY, C. **Security improvisation through node trust prediction approach in mobile ad hoc networks**. *International Journal of Interactive Mobile Technologies (iJIM)*, 13:40, 2019.

[44] RANAWEERA, P.; JURCUT, A.; LIYANAGE, M. **Mec-enabled 5g use cases: A survey on security vulnerabilities and countermeasures**. *ACM Comput. Surv.*, 54(9), oct 2021.

[45] RAZA, S. M.; THORAT, P.; CHALLA, R.; CHOO, H. **On demand inter domain mobility in sdn based proxy mobile ipv6**. In: *2017 International Conference on Information Networking (ICOIN)*, p. 194–199. IEEE, 2017.

[46] RILEY, G. F.; HENDERSON, T. R. **The ns-3 Network Simulator**, p. 15–34. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[47] ROCHA, A. S.; PINHEIRO, B. A.; BORGES, V. C. **Secure d2d caching framework inspired on trust management and blockchain for mobile edge caching**. *Pervasive and Mobile Computing*, 77:101481, 2021.

[48] SENTZ, K.; FERSON, S.; OTHERS. **Combination of evidence in Dempster-Shafer theory**, volume 4015. Citeseer, 2002.

[49] SHAFER, G. **A mathematical theory of evidence**, volume 42. Princeton university press, 1976.

[50] SHAH, S. D. A.; GREGORY, M. A.; LI, S.; FONTES, R.; HOU, L. **Sdn-based service mobility management in mec-enabled 5g and beyond vehicular networks**. *IEEE Internet of Things Journal*, 2022.

[51] SHAH, S. D. A.; GREGORY, M. A.; LI, S.; FONTES, R. D. R. **Sdn enhanced multi-access edge computing (mec) for e2e mobility and qos management**. *IEEE Access*, 8:77459–77469, 2020.

[52] SUKHWANI, H.; MARTÍNEZ, J. M.; CHANG, X.; TRIVEDI, K. S.; RINDOS, A. **Performance modeling of pbft consensus process for permissioned blockchain network (hyperledger fabric)**. In: *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, p. 253–255. IEEE, 2017.

[53] WANG, K. H.; BAOCHUN LI. **Group mobility and partition prediction in wireless ad-hoc networks**. In: *2002 IEEE International Conference on Communications. Conference Proceedings. ICC 2002 (Cat. No.02CH37333)*, volume 2, p. 1017–1021 vol.2, April 2002.

[54] WANG, M.; YAN, Z. **A Survey on Security in D2D Communications**. *Mobile Networks and Applications*, 22(2):195–208, Apr. 2017.

[55] WANG, T.; QIU, L.; SANGAIAH, A. K.; LIU, A.; BHUIYAN, M. Z. A.; MA, Y. **Edge-computing-based trustworthy data collection model in the internet of things**. *IEEE Internet of Things Journal*, 7(5):4218–4227, 2020.

[56] WANG, T.; SUN, Y.; SONG, L.; HAN, Z. **Social data offloading in d2d-enhanced cellular networks by network formation games**. *IEEE Transactions on Wireless Communications*, 14(12):7004–7015, 2015.

[57] WEI, Z.; TANG, H.; YU, F. R.; WANG, M.; MASON, P. **Security enhancements for mobile ad hoc networks with trust management using uncertain reasoning**. *IEEE Transactions on Vehicular Technology*, 63(9):4647–4658, 2014.

[58] WU, B.; XU, K.; LI, Q.; REN, S.; LIU, Z.; ZHANG, Z. **Toward blockchain-powered trusted collaborative services for edge-centric networks**. *IEEE Network*, 34(2):30–36, 2020.

[59] WU, X.; LIANG, J. **A blockchain-based trust management method for internet of things**. *Pervasive and Mobile Computing*, 72:101330, 2021.

[60] XU, Q.; SU, Z.; LU, R. **Game theory and reinforcement learning based secure edge caching in mobile social networks**. *IEEE Transactions on Information Forensics and Security*, 15:3415–3429, 2020.

[61] XU, Q.; SU, Z.; YANG, Q. **Blockchain-based trustworthy edge caching scheme for mobile cyber-physical system**. *IEEE Internet of Things Journal*, 7(2):1098–1110, 2020.

[62] YAN, T.; CHEN, W.; ZHAO, P.; LI, Z.; LIU, A.; ZHAO, L. **Handling conditional queries and data storage on hyperledger fabric efficiently**. *World Wide Web*, 24(1):441–461, 2021.

[63] YAN, Z.; PENG, L.; FENG, W.; YANG, L. T. **Social-chain: Decentralized trust evaluation based on blockchain in pervasive social networking**. *ACM Trans. Internet Technol.*, 21(1), jan 2021.

[64] YAN LINDSAY SUN.; WEI YU.; ZHU HAN.; LIU, K. J. R. **Information theoretic framework of trust modeling and evaluation for ad hoc networks**. *IEEE Journal on Selected Areas in Communications*, 24(2):305–317, 2006.

[65] YANG, H.; LIANG, Y.; YUAN, J.; YAO, Q.; YU, A.; ZHANG, J. **Distributed blockchain-based trusted multidomain collaboration for mobile edge computing in 5g and beyond**. *IEEE Transactions on Industrial Informatics*, 16(11):7094–7104, 2020.

[66] YIN, X.; LI, S. **Trust evaluation model with entropy-based weight assignment for malicious node's detection in wireless sensor networks**. *EURASIP Journal on Wireless Communications and Networking*, 2019(1):1–10, 2019.

[67] YU, B.; ZHANG, X.; YOU, I. **Collaborative cache allocation and transmission scheduling for multi-user in edge computing**. *IEEE Access*, 8:163953–163961, 2020.

[68] ZHANG, R.; YU, F. R.; LIU, J.; HUANG, T.; LIU, Y. **Deep reinforcement learning (drl)-based device-to-device (d2d) caching with blockchain and mobile edge computing**. *IEEE Transactions on Wireless Communications*, 19(10):6469–6485, 2020.

[69] ZHANG, R.; YU, F. R.; LIU, J.; XIE, R.; HUANG, T. **Blockchain-Incentivized D2D and Mobile Edge Caching: A Deep Reinforcement Learning Approach**. *IEEE Network*, 34(4):150–157, July 2020.

[70] ZHENG, X.; LI, M.; CHEN, Y.; GUO, J.; ALAM, M.; HU, W. **Blockchain-based secure computation offloading in vehicular networks**. *IEEE Transactions on Intelligent Transportation Systems*, 22(7):4073–4087, 2021.

APPENDIX A

# Theory of Dempster Shafer in Indirect Trust Assessment

The Theory of Dempster-Shafer (TDS) is a generalization of the classical probability theory, where an evidence can be associated with more than one event, making it possible to represent uncertainty with better precision, since it does not require assumptions regarding any event[48]. The versatility of the TDS, combined with the possibility of probabilistic combination of evidence, characterize a good resource for trust assessment based on recommendation, that is, based on indirect observations.

TDS has the ability to combine and aggregate evidence that reinforces the hypotheses. In this sub-section, we describe the basic concepts of TDS, exemplifying them according to the proposed model.

The set of hypotheses in TDS is named the problem domain, or **Discernment Frame**, represented by $\Omega$. Taking as an example an universe of two hypotheses $H$ and $\bar{H}$, the picture of discernment would be: $\Omega = \{H, \bar{H}\}$. The set of all possible combinations of $\Omega$ is called as **Power Set** and it is represented by $2^{\Omega}$, the elements of $2^{\Omega}$ are also called **focal elements**.

For our model, the received evidence follows to two hypotheses: $H = \{\text{Trustworthy}\}$ and $\bar{H} = \{\text{Untrustworthy}\}$. Consequently, the power set is given by $2^{\Omega} = \{\emptyset, H, \bar{H}, U\}$, where $U = \Omega$ is called as the Universe Set and represents the entire frame of discernment, that is, it represents both trust and non-trust. The mass, or Basic Probability Assignment (BPA) is a function, or measure associated with $2^{\Omega}$.

Let $S \subseteq \Omega$ be a subset of the hypotheses, the associated mass by evidence, with the subset $S$ represents the measure of belief in the subset, that is, how much we believe that given an evidence a specifc hypothesis will happen. For the scenario we created, where $\Omega = \{H, \bar{H}\}$, we have three possible subsets of hypotheses, so that each evidence represents an associated mass with each of the hypothesis sets:

$$(1)\ S = \{H\}$$
$$(2)\ S = \{\bar{H}\}$$
$$(3)\ S = U = \{H, \bar{H}\}$$

Formally, the description of the mass is defined in three properties: (P1) the mass represents a mapping of the Power Set for an interval between 0 and 1, (P2) the mass for the set $\emptyset$ is 0 and (P3) a sum of the masses assigned to each subset of $2^{\Omega}$ is 1:

$$(P1)\ m : 2^{\Omega} \rightarrow [0, 1]$$
$$(P2)\ m(\emptyset) = 0$$
$$(P3)\ \sum_{S \subseteq \Omega} m(S) = 1$$

Each evidence (focal element) has a mass $m(s)$ associated with a hypothesis $s \subseteq S$. Bearing in mind that an element of the set $C_v$ is observation of indirect behavior, we consider that the associated mass with a hypothesis corresponds to the degree of direct trust of the node regarding the evidence. Figure 2.2 shows an example of this, where $A$ is the node that assesses the indirect trust of $B$ in a scenario where $n_1$ believes that $B$ is trustworthy. The associated mass with the hypothesis that $B$ is trustworthy is equal to the degree of trust by direct observations from the user $A$ on $n_1$, that is $T^{D}_{n_1}$. The association of the direct trust value in the calculation of the indirect trust is also used by [3, 57].

The belief value in a $S \subseteq \omega$ hypothesis is reached by sum the masses $m(s)$ according to TDS. So we call $Bel(S)$ as the belief value associated with each hypothesis and describe it mathematically according to the Equation A-1.

$$Bel(S) = \sum_{s \subseteq S} m(s) \tag{A-1}$$

It is important noting that the $Bel(S)$ belief value under the $S$ subset does not imply that the belief under its complement $\bar{S}$, is $Bel(\bar{S}) = 1 - Bel(S)$, this is the greatest difference between Dempster Shafer's Theory and the standard probability theory.

Figure 2.2 shows an example for indirect observations based on TDS, node $A$ is evaluating the trust of node $B$, so $T^{I}_{A,B}$ is the DT for indirect observations of $A$ over $B$. Let $Cv_B = \{(A, 1), (n_1, 1), (n_2, 0)\}$ be the set of extracted votes from the last block, after the end of the distributed consensus, where each $t_d \in Cv_B$ is a tuple $(id, d)$ corresponding to the node's IP and its vote on the trustworthiness of $B$. That is, $A$, $n_1$ and $n_2$ are CMs from the same cluster (blockchain network), where $B$ is trustworthy for $n_1$, while $B$ is not for $n_2$. We take into consideration the direct trust value of $A$ over $n_1$ and $n_2$, as the mass of each

of the associated nodes with a specific hypothesis in order to be able to apply TDS in our proposal. For example, let $T^D_{A,n_1}$ be the mass of $n_1$ on the hypothesis $H$ and $T^D_{A,n_2}$ the mass of $n_2$ on the hypothesis $\bar{H}$, the scenario configuration, according to the TDS, is as follows:

$$\begin{aligned} m_{n_1}(H) &= T^D_{A,n_1} \\ m_{n_1}(\bar{H}) &= 0 \\ m_{n_1}(U) &= 1 - T^D_{A,n_1} \end{aligned} \tag{A-2}$$

$$\begin{aligned} m_{n_2}(H) &= 0 \\ m_{n_2}(\bar{H}) &= T^D_{A,n_2} \\ m_{n_2}(U) &= 1 - T^D_{A,n_2} \end{aligned} \tag{A-3}$$

The degree of belief in each hypothesis is given by the equation 2-3 and for the mentioned case above we would have the following configuration:

$$\begin{aligned} Bel_{n_1}(H) &= m_{n_1}(H) \\ Bel_{n_1}(\bar{H}) &= m_{n_1}(\bar{H}) \end{aligned} \tag{A-4}$$

$$\begin{aligned} Bel_{n_2}(H) &= m_{n_2}(H) \\ Bel_{n_2}(\bar{H}) &= m_{n_2}(\bar{H}) \end{aligned} \tag{A-5}$$

However, we need a method to combine observations, that is, to aggregate different values of belief in each hypothesis of the discernment frame. We need to combine the belief degrees $Bel_{n_1}$ and $Bel_{n_2}$ associated with each hypothesis $S \subseteq \Omega$ in the example of Figure 2.2. To reach this, we use the **Dempster's rule of Combination**.

The Dempster's rule of Combination associates different belief functions through mass aggregation[49].

Let $m_1$ and $m_2$ two masses, $s_1$ and $s_2$ two focal elements. We combine different observations as follows:

$$Bel(S) = m_1(S) \oplus m_2(S) = \frac{\sum_{s_1 \cap s_2 = S} m_1(s_1) m_2(s_2)}{1 - K}, \; para \; S \neq \emptyset \tag{A-6}$$

where $K$ represents the associated mass with the conflict and is defined as follows:

$$K = \sum_{s_1 \cap s_2 = \emptyset} m_1(s_1) m_2(s_2) \tag{A-7}$$

the denominator in Equation A-6 represents a normalization factor [49] pg. 64-66.

The proposed scenario in Figure 2.2 is used to exemplify the Dempster Combination Law, where $T^D_{A,n_1} = 0.85$ and $T^D_{A,n_2} = 0.3$ and $Cv_B = \{(A,1),(n_1,1),(n_2,0)\}$ is the set of extracted votes after finalization of the distributed consensus. Through $Cv_B$ we conclude that $n_1$ trusts $B$, but $n_2$ does not, that is, $(n_1,1) \in Cv_B$ is evidence for the $H$ =*trustworthy*, while $(n_2,0) \in Cv_B$ is evidence for the hypothesis $\bar{H}$ =*untrustworthy*. Thus obeying the properties P1 to P3, the association of masses is as follows:

$$m_{n_1}(H) = 0.85 \quad m_{n_1}(\bar{H}) = 0 \quad m_{n_1}(U) = 0.15$$
$$m_{n_2}(H) = 0 \quad m_{n_2}(\bar{H}) = 0.3 \quad m_{n_2}(U) = 0.7$$

being $m_{n_1}(H) = T^D_{A,n_1}$ and $m_{n_2}(\bar{H}) = T^D_{A,n_2}$. We employs the Combination Law (Equation A-6) to calculate the trust value based on indirect observations, obtaining the following sums:

$$Bel(H) = m_{n_1}(H) * m_{n_2}(H) + m_{n_1}(H) * m_{n_2}(U) + m_{n_2}(H) * m_{n_1}(U)$$
$$Bel(\bar{H}) = m_{n_1}(\bar{H}) * m_{n_2}(\bar{H}) + m_{n_1}(\bar{H}) * m_{n_2}(U) + m_{n_2}(\bar{H}) * m_{n_1}(U)$$
$$Bel(U) = m_{n_1}(U) * m_{n_2}(U)$$

we hide the denominator $1 - K$ in order to simplify the example and the organization of the calculations, however it is considered in the calculation of the combination of evidence.

$$Bel(H) = 0,85 * 0 + 0,85 * 0,7 + 0 * 0,7 = 0,595$$
$$Bel(\bar{H}) = 0 * 0,3 + 0 * 0,7 + 0,3 * 0,15 = 0,045$$
$$Bel(U) = 0,15 * 0,7 = 0,105$$

With the Dempster's rule of combination, the indirect trust from $A$ over $B$, is 0.595. Generalizing this situation, for $n$ distinct nodes the indirect trust of $B$ is the belief value aggregated to the $H$ hypothesis, calculated through the following equation:

$$T^I_{A,B} = m_1(H) \oplus m_2(H) \oplus \cdots \oplus m_n(H) \tag{A-8}$$

At the end of the trust assessment by indirect observations, the overall trust value is updated according to Equation 2-1. For example, Figure 2.2 show the degree of trust by direct observations from $A$ over $B$ is given by $T^D_{A,B} = 0.4$ and that the associated weight with $T^D$ is $\omega = 0.6$. Also consider the scenario used in this sub-section (Figure 2.2), where $T^I_{A,B} = 0.595$. Thus, the degree of overall trust is the result of the aggregation $T^D_{A,B}$ and $T^I_{A,B}$, calculated as follows:

$$T_{A,B} = \omega T^{D}_{A,B} + (1 - \omega) T^{I}_{A,B}$$
$$T_{A,B} = 0.6 * 0.4 + (1 - 0.6) * 0.59$$
$$T_{A,B} = 0.6 * 0.4 + 0.4 * 0.59$$
$$T_{A,B} = 0.24 + 0.236$$
$$T_{A,B} = 0.476$$

that is, the overall degree of trust of the $A$ node over the $B$ node is $T_{A,B} = 0.476$. Assuming that the trustworthiness threshold is $\kappa = 0.5$, as $T_{A,B} = 0.476 < \kappa$, we conclude that the node $A$ does not trust $B$.