Explorando a capacidade de microcontroladores para operações de Internet de Valor na Blockchain Neo

Vanessa F. da Silva[1], Maria Clicia S. de Castro[1], Igor M. Coelho[2]

[1]Instituto de Matemática e Estatística
Universidade do Estado do Rio de Janeiro (UERJ)
Rio de Janeiro, RJ – Brasil

[2]Instituto de Computação
Universidade Federal Fluminense (UFF)
Niterói, RJ – Brasil

vanessa.fernandes.silva@hotmail.com, clicia@ime.uerj.br, imcoelho@ic.uff.br

Abstract. *This paper addresses transaction structure aspects in Blockchain Neo from the perspective of low energy consumption devices. With the Internet of Valor, these devices will transfer values and make payments quickly, securely, and verifiable through digital signatures and smart contracts on the blockchain. Besides classic cryptocurrencies, new types of tokens (such as non-fungible tokens) have received wide interest from industry, and also for increasingly complex applications in society, demanding greater computational capacity of networks and involved devices, so as increasingly strict requirements of privacy and security. Finally, we investigated a prototype integrated into a local experimental network installed in a national university to validate the proposal.*

Resumo. *Este artigo aborda aspectos de estrutura das transações na Blockchain Neo, pela perspectiva de dispositivos de baixo consumo energético. Com a Internet de Valor, esses dispositivos serão capazes de transferir valores e realizar pagamentos de forma rápida, segura e verificável através de assinaturas digitais e contratos inteligentes na blockchain. Além de criptoativos clássicos, novos tipos de tokens (como tokens não-fungíveis) tem recebido amplo interesse na indústria e também para aplicações cada vez mais complexas na sociedade, exigindo maior capacidade computacional das redes e dos dispositivos envolvidos, bem como requisitos cada vez mais rígidos de privacidade e segurança. Finalmente, experimentamos um protótipo integrado a uma rede experimental local, instalada em uma universidade nacional, para validar a proposta.*

1. Introdução

A transferência de valores têm uma função fundamental na economia mundial. A Internet de Valor (IoV) é um conceito que vem sendo desenvolvido para permitir a transferência e o armazenamento de valores tão rapidamente quanto se transferem outras informações [Truong et al. 2018]. De forma geral, valores são atualmente transferidos em escala global de modo caro, pouco confiável e lento. Em geral, pode demorar dias e envolve vários procedimentos intermediários de terceiro para validar e processar as transações. Nesse sentido, a tecnologia Blockchain [Nakamoto 2008] surge como uma plataforma capaz de viabilizar processos de confiança sem o uso de intermediários.

Através da IoV uma transação de valor ocorrerá instantaneamente e permitirá a troca de qualquer ativo valioso, incluindo ações, votos, contratos digitais, valores mobiliários, propriedade intelectual, *tokens* fungíveis e não-fungíveis (NFTs), descobertas científicas, entre outros [Coelho et al. 2021]. A tecnologia Blockchain viabiliza a implementação prática da IoV, devido a sua capacidade de resistir a ataques de gasto duplo, bem como sua abertura a qualquer participante interessado com acesso à Internet. Para isso, é importante investigar a capacidade computacional necessária para a implementação prática da IoV.

A tecnologia Blockchain possui propriedades que impedem a corrupção dos valores existentes. A Blockchain é baseada numa base de dados pública e distribuída de registros de todas as transações ou eventos digitais que são executados e compartilhados entre as partes participantes [Di Pierro 2017]. Ao invés desses dados serem registrados em um computador central, a tecnologia blockchain possibilita que os dados sejam armazenados em múltiplos de computadores geograficamente dispersos. Cada computador da rede possui uma cópia integral da base de dados, tornando assim, as informações registradas extremamente seguras e confiáveis. Outra vantagem do armazenamento distribuído é não possuir um ponto único de acesso inviabilizando ataques. Por essa razão, a tecnologia também é denominada de *Distributed Ledger Tecnology* (DLT) [1], especialmente quando não há um conceito formal de blocos (também chamadas de não-blockchain).

Uma blockchain é formada por uma cadeia de blocos. Cada bloco armazena um conjunto de transações e está associado a um período de tempo. Quando é gerado um novo bloco de transações, ele se liga ao anterior por um elo, um código chamado *hash* [Nofer et al. 2017]. Em blockchains públicas e com suporte a mineração, como Bitcoin [Nakamoto 2008] e Ethereum [Wood 2014], todas as transações são criptografadas usando algoritmos matemáticos e verificadas pelos mineradores. Os mineradores verificam e registram as transações no bloco, se a maioria simples $(50\% + 1)$ da rede concordar que aquela transação é legítima [Nakamoto 2008]. Outras blockchains adotam um modelo de consenso bizantino [Lamport et al. 2019], geralmente inspirados no algoritmo *Practical Byzantine Fault Tolerance* [Castro et al. 1999]. Assim, evitam o processo de mineração e reduzem o número de elementos necessários para atingir um consenso. Um exemplo de blockchain pública deste tipo é a Neo Blockchain [Hongfei, Da and Zhang, Erik 2015]. Aspectos formais mais detalhados de definição da blockchain podem ser encontrados nas normas ISO [ISO 2020].

Neste trabalho, consideramos o uso de microcontroladores [Davies 2008], de baixo custo de aquisição e baixo custo energético, para operar dentro de um protótipo de rede IoV. Embora os experimentos possam ser feitos em redes públicas, isso introduz custos de aquisição de *tokens*, que são atrelados à volatilidade do mercado. Consideramos uma instância da blockchain Neo executando em uma universidade nacional, em modelo público-permissionado. O uso da blockchain Neo, bem como da plataforma de experimentação NeoCompiler Eco [Coelho and Coelho 2021], permite a utilização completa e monitoramento da rede, favorecendo abordagens experimentais sem a necessidade de pagar efetivamente por *tokens* de uma rede pública descentralizada. Assim, implementamos em microcontrolador e testamos operações da máquina virtual da Neo, chamada de NeoVM, que permite computação Turing-completa e de forma determinística (veri-

[1] `https://www.itu.int/en/ITU-T/focusgroups/dlt/`

ficável) [Malamud and Rostek 2017]. Dentre as contribuições deste artigo elencamos: (i) levantamento de aplicações, motivação para redes permissionadas e Internet de Valor; (ii) apresentação de um detalhamento da estrutura de transações na Neo Blockchain; e (iii) implementação e testes de um protótipo integrado a uma rede permissionada.

Este artigo está organizado em cinco seções, incluindo esta Introdução. Seção 2 aborda aspectos teóricos da Internet de Valor, na perspectiva do uso de microcontroladores para aplicações blockchain. A Seção 3 apresenta a estrutura das transações na blockchain Neo, bem como *opcodes* importantes para efetuar IoV na máquina virtual NeoVM. Na Seção 4 é apresentado um experimento prático, utilizando a infraestrutura computacional do ALODE, instalado em uma universidade nacional. Finalmente, a Seção 5 conclui o trabalho e apresenta perspectivas futuras.

2. Internet de Valor e Microcontroladores

A internet de valor representa um novo paradigma onde os dispositivos são capazes de trocar valor diretamente, possibilitando o surgimento de uma nova geração da Internet. Junto com a Internet das Coisas (IoT) [Weber and Weber 2010], a IoV traz novos recursos para dispositivos, hoje em dia por meio das DLT's. Embora existam propostas específicas de DLT para IoT [Silvano and Marcelino 2020], atualmente a tecnologia mais conhecida e explorada é a Blockchain, empregada em plataformas como Bitcoin [Nakamoto 2008], Ethereum [Wood 2014] e Neo [Hongfei, Da and Zhang, Erik 2015].

A interação de dispositivos na IoV ocorre através de regras pré-estabelecidas, em um conceito denominado de Contrato Inteligente [Szabo 1997]. Embora o Bitcoin permita diversos tipos de operações (denominadas *opcodes* na linguagem *Script*[2]), o suporte para contratos inteligentes é ainda bastante limitado, não permitindo execução de *loops* nem operações mais complexas em sua máquina virtual. Por essa razão, o Ethereum trouxe inovações em sua máquina virtual Turing-Completa, o Ethereum Virtual Machine (EVM), o que também demandou a criação do conceito de *custo de execução na rede*, o *GAS*. Esse custo varia de acordo com a capacidade de mineração da rede. Ele é ajustado em momentos de pico de execução, estando sujeito à volatilidade do custo do Ether, que é o ativo global da rede Ethereum. A Neo também permite a criação de contratos inteligentes através de sua máquina virtual Turing-Completa, a NeoVM[3]. Assim como a EVM, ela exige que custos computacionais sejam pagos em *GAS*. Desta forma, a implementação de máquinas virtuais Turing-completas em dispositivos de baixo custo energético para IoV ainda se mostra um desafio [da Silva et al. 2019].

O conceito de *home ledger* [da Silva et al. 2019] aborda aplicações de blockchains para uso restrito, como residências e condomínios, de forma a permitir trocas de ativos como energia elétrica [Zia et al. 2020, Gabrich et al. 2020] e dados internos de segurança. O conceito aproxima DLT e IoV para usuários domésticos, melhorando a capacidade de auditoria para dispositivos IoT locais, como travas inteligentes, armazenamento em nuvem local [Martini and Choo 2013] e câmeras auditáveis [Khan et al. 2020].

Em [da Silva et al. 2019] também é possível analisar os resultados de alguns experimentos realizados com o dispositivo usado neste trabalho, como por exemplo o microcontrolador ESP-8266. Um microcontrolador é um *chip* de circuito integrado único,

[2]https://en.bitcoinwiki.org/wiki/Script

[3]github.com/neo-project/neo-vm

que contém processador, memória, periféricos de entrada e saída, temporizadores e dispositivos de comunicação serial que podem ser programados. São utilizados em sistemas embarcados, que produzem uma sequência de tarefas pré-estabelecidas, controladas pelos dispositivos em questão. Atualmente, muitos microcontroladores já possuem capacidade de comunicação WiFi, o que facilita a aplicação do conceito de IoT [Davies 2008]. O microcontrolador utilizado neste trabalho pode ser visto na Figura 1.

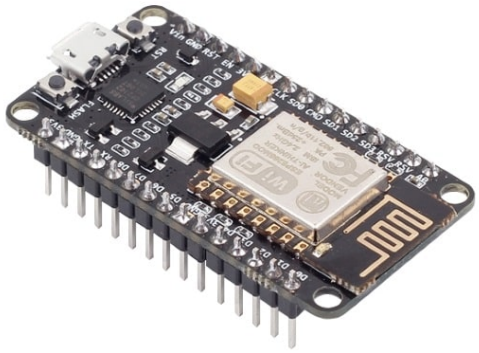

Figura 1. Microcontrolador ESP-8266

2.1. BIoT: Blockchain e IoT

Um tópico bastante explorado na literatura recente envolve o conceito de BIoT. BIoT é o termo utilizado para nominar a combinação entre a tecnologia Blockchain e IoT. A IoT é o conceito onde objetos estão conectados à Internet e entre si, permitindo que pessoas e coisas se conectem a qualquer hora, em qualquer lugar, com qualquer coisa e qualquer pessoa. Assim, são capazes de coletar, armazenar, compartilhar e processar dados. Desta forma, a IoT precisa adotar um modelo eficiente de segurança que seja capaz de garantir mais proteção na comunicação entre os dispositivos. É neste cenário, que a tecnologia Blockchain surge para permitir que os dispositivos IoT operem sem a necessidade de um servidor central, o que irá garantir mais agilidade, confiabilidade e segurança na troca de informações [Kamran et al. 2020].

Em [Monteiro et al. 2021], é apresentado o Modelo de Rastreabilidade Pervasiva de Agroquímicos (MRPA), que atua na combinação de múltiplos sensores de IoT, Machine Learning e Blockchain a fim de rastrear as embalagens de agroquímicos. O modelo possui o intuito não apenas de combater roubos nas propriedades rurais, mas também questões como poluição do meio ambiente, saúde da população, uso exagerado, evasão fiscal, perdas financeiras para o agricultor e para o estado. Esse modelo faz uso de sensores e microcontroladores para monitorar o estado e a localização de defensivos agrícolas, monitorando assim, o processo completo de rastreabilidade da cadeia. Após a recepção e processamento desses dados IoT, eles são gravados em uma rede Blockchain, permitindo um registro permanente imutável de todos os eventos ocorridos durante a movimentação das embalagens de defensivos agrícolas.

3. Transações na Neo Blockchain

Operações na Internet de Valor se dão através de transações em uma plataforma blockchain. Devido à escassez de material em português (e até mesmo em inglês ou chinês na documentação oficial do projeto), várias informações importantes são obtidas somente através do código-fonte do projeto. Por essa razão, introduziremos detalhadamente a estrutura das transações na blockchain Neo.

Na blockchain Neo, uma transação é uma estrutura de dados assinada com uma instrução/*script* NeoVM para toda a rede Neo. Por exemplo, um usuário (remetente) indica que deseja transferir ativos para outro endereço (destinatário). Assim como o Bitcoin, na versão Neo v2 Legacy considerada neste trabalho, ativos globais são protegidos contra gasto duplo através de uma estrutura de *Unspent Transaction Outputs* (UTXO).

Carteiras, contratos inteligentes e contas interagem com a rede por meio das transações. Cada bloco na blockchain contém uma ou mais transações, tornando cada bloco um lote de transações. Esses blocos são assinados pelas réplicas de consenso.

As transações na NEO possuem três partes importantes, que são abordadas neste artigo: entrada, saída e *scripts*. O ciclo de vida de uma transação engloba todo o processo, desde a sua criação, submissão via *peer-to-peer*, até a eventual inclusão da transação na blockchain por uma réplica de consenso [Zhang 2021].

Existem vários tipos de transação, dentre elas a *Contract Transaction* desempenha uma função fundamental na troca de ativos. Essa transação é responsável por formalizar em código a transferência de valor de um ou mais *inputs* (fonte dos fundos) para um ou mais *outputs* (destino dos fundos) [NGD 2021]. A estrutura geral de uma transação é apresentada na Tabela 1.

Tabela 1. Estrutura de uma transação Contract Transaction na Neo.

Campo	Descrição	Tamanho
Type	O tipo da transação	1 Byte
Version	Versão de compatibilidade	1 Byte
Attribute	Recursos adicionais para a transação	1 Byte
Input	O(s) *input(s)* da transação	*varbytes*
Output	O(s) *output(s)* da transação	*varbytes*
Script	Testemunhas que validam a transação	*varbytes*

Na Tabela 1 *varbytes* indica que um objeto mais complexo é serializado com número variável de *bytes*, sendo prefixado pelo tamanho do campo (em *bytes*) e pelo o número de repetições (como um *array*).

3.1. Type

O campo *type* é responsável por identificar o tipo da transação, ou seja, identifica as regras da transação para todos os nós na rede. Assim, é possível que esses nós decidam como verificar as transações, ou até mesmo descartá-las havendo incompatibilidade. Existem nove diferentes tipos de transações na Neo v2 Legacy. Cada uma possui propostas distintas bem como características distintas (veja Tabela 2).

Tabela 2. Tipos de transação e usos na blockchain.

Tipo	Ativos	SC	dBFT	Descrição	Código
MinerTransaction			✓	A primeira transação de um bloco, usada para distribuir as recompensas pela geração do bloco (somente nós validadores)	`0x00`
Register Transaction	✓			Registra ativos globais (como NEO ou GAS)	`0x40`
IssueTransaction	✓			Emite ativos globais (como NEO ou GAS)	`0x01`
ClaimTransaction	✓			Reivindica GAS a partir de um saldo em NEO	`0x02`
StateTransaction			✓	Inscreve-se como candidato a validador ou vota em nós de consenso	`0x90`
EnrollmentTransaction			✓	Inscreve-se como candidato a validador	`0x20`
ContractTransaction	✓	✓		Transação para troca de ativos	`0x80`
PublishTransaction		✓		Publica contratos inteligentes	`0xd0`
InvocationTransaction		✓		Invoca um contrato inteligente implantado na rede	`0xd1`

Na Tabela 2, as colunas **Tipo** e **Descrição** apresentam cada tipo de transação; o campo **Código** codifica o primeiro byte desse tipo de transação; e os campos **Ativos**, **SC** e **dBFT** descrevem a participação daquele tipo de transação, respectivamente: para operações de manipulação de ativos globais; para gestão de contratos inteligentes (do inglês, *Smart Contracts*); e para a operação de consenso bizantino da rede (do inglês, *delegated Byzantine Fault Tolerance*). Devido à contextualização deste trabalho em operações para IoV, focamos na transação código `0x80`, a *Contract Transaction*.

3.2. Version

O campo *version* permite atualizações na estrutura da transação com compatibilidade com versões anteriores. Atualmente, as versões zero e um são compatíveis com *Invocation-Transaction* e a versão zero é a única versão com suporte para qualquer outro tipo de transação.

3.3. Attribute

Dependendo do tipo da transação, é possível adicionar *attribute*. Para cada *attribute* existe algumas variações para *usage type*, onde pode ser especificado junto com o dado externo e o tamanho do mesmo (veja Tabela 3).

Tabela 3. Atributo

Campo	Descrição
Usage	Tipo de uso de *attribute*
Length	Tamanho do dado (se requerido)
Data	Dados externos anexados para tipo de uso

Na Tabela 3, observamos que informações adicionais em uma transação podem entrar em um campo `Data`, a partir da marcação específica de *Usage* (tipicamente é utilizado um tipo denominado *REMARK=0xf0*). Outro uso comum do campo atributo é para anexar novas testemunhas (do inglês, *witness*) à transação (basta adicionar o hash da testemunha no *Data* e marcar o *Usage* como tipo *SCRIPT=0x20*). Isso permite que múltiplos indivíduos assinem a transação simultaneamente, permitindo transações com diversas origens e destinos em uma mesma transação.

Como abordado anteriormente, cada transação de *attribute* pode ter diferentes *usages*. Veja demais opções na Tabela 4.

Os campos *ContractHash*, *ECDH02*, *ECDH03* e *Hash1 - Hash15* têm o comprimento dos dados fixado em 32 Bytes e o campo de comprimento é omitido. O campo *Script* tem o comprimento dos dados fixado em 20 Bytes e o campo de comprimento também é omitido. Os campos *Vote*, *DescriptionUrl* e *Description* têm o comprimento dos dados definido e o mesmo não deve exceder a 255 Bytes. O campo *Remark - Remark15* tem o comprimento dos dados definido e o mesmo não deve exceder 65535 bytes.

3.4. Input

Os campos *input* e *output* são os elementos essenciais de uma transação. As transações são relacionadas umas às outras por estes dois elementos. Os *inputs* de uma transação especifica a origem dos ativos, ou seja, nele está contido a informação de saída de ativos

Tabela 4. Tipos de uso

Campo	Valor	Descrição
ContractHash	`0x00`	Valor hash do contrato
ECDH02	`0x02`	Chave pública para troca de chave ECDH
ECDH03	`0x03`	Chave pública para troca de chave ECDH
Script	`0x20`	Inclui Testemunha complementar na transação
Vote	`0x30`	Para votar
DescriptionUrl	`0x81`	Endereço de URL da descrição
Description	`0x90`	Descrição breve
Hash1–Hash15	`0xa1-0xaf`	Usado para armazenar hash personalizados
Remark–Remark15	`0xf0-0xff`	Observações

não gastos ou recebidos de uma transação anterior (também conhecidos como UTXO). Cada transação pode ter zero ou até 65536 *inputs*. A estrutura de dados de *input* pode ser observada na Tabela 5:

Tabela 5. Estrutura do campo Input

Campo	Tamanho	Descrição
PrevHash	32 bytes	Hash da transação anterior
PrevIndex	2 bytes	Índice na transação anterior

Os campos *PrevHash* e *PrevIndex* são respectivamente, o hash e o índice da transação que deu origem ao saldo atual em little endian do remetente. A combinação de *PrevHash* e *PrevIndex* é chamada de *Coin Reference*.

3.5. Output

O campo *output* é composto pelo Hash do ativo em little endian, o valor a ser transferido no formato fixed8 em little endian e o script hash do endereço do destinatário também em little endian. Cada transação pode ter até 65536 *outputs*. A estrutura de dados de *output* pode ser observada na Tabela 6:

Tabela 6. Estrutura do campo output

Campo	Tamanho	Descrição
AssetId	32 bytes	Id do ativo global (NEO/GAS/...)
Value	10 bytes	Valor que deseja transferir (em *little-endian fixed8*)
ScriptHash	20 bytes	Endereço do destinatário

3.6. Script

O campo *Script* é responsável por verificar a validade e integridade da transação. Um objeto na matriz de *scripts* é tipicamente referido como *witness* (ou *testemunha*). Há dois campos para cada *witness* na matriz de scripts, como pode ser observado na Tabela 7:

Antes de cada transação ser adicionada a um bloco, é necessário assina-la digitalmente para garantir que a mesma não seja modificada durante a transmissão. A NEO

Tabela 7. Estrutura do campo Script

Campo	Descrição
InvocationScript	Envia assinaturas de validação para o script de verificação
VerificationScript	Envia a(s) chave(s) pública(s) correspondente(s) ao contrato

usa o método de assinatura digital por curva elíptica ECDSA *secp256r1* [NIST 2013]. Assim, o campo *InvocationScript* executa as instruções de operação da pilha, fornecendo a(s) assinatura(s) para o script de verificação.

O campo *InvocationScript* é composto da seguinte maneira:

1. `0x40` (PUSHBYTES64)
2. assinatura (com 64 bytes)

Nesse processo acima, o hexadecimal 40 equivale a 64 em decimal. Assim, é empilhado 64 Bytes e em seguida é empilhada a assinatura de mesmo tamanho. O *invocationScript* pode enviar mais de uma assinatura para contratos do tipo *CHECKMULTISIG=0xAE*, ou seja, contratos com várias assinaturas. Assim, o processo acima será repetido para o total de assinaturas.

O *verificationScript* com uma única assinatura é realizado assim:

1. `0x21` (PUSHBYTES33)
2. Chave pública
3. `0xAC` (CHECKSIG)

Neste processo, o hexadecimal 21 equivale a 33 em decimal. Logo, empilha-se 33 bytes, a chave pública de mesmo tamanho correspondente a assinatura e o opcode `CHECKSIG=0xAC` que verifica a chave pública e assinatura dada.

No caso de contrato com várias assinaturas, o *verificationScript* é realizado da seguinte forma:

1. `0x52` (PUSH2) a `0x60` (PUSH16)
2. `0x21` (PUSHBYTES33)
3. Chave pública 1
4. ... empilha demais chaves
5. `0x52` (PUSH2) to `0x60` (PUSH16)
6. `0xAE` (CHECKMULTISIG)

Neste processo, a etapa um indica a quantidade necessária de assinaturas, podendo variar de 2 a 16 assinaturas. A etapa dois empilha 33 bytes e em seguida a primeira chave pública é empilhada. As etapas dois e três se repetem de acordo com o número de chaves públicas. A etapa quatro indica a quantidade total de chaves públicas para as assinaturas. Finalmente, a última etapa empilha o opcode `0xAE` que verifica as assinaturas com as chaves públicas fornecidas.

Após os dados da transação serem preenchidos, ela é transmitida por toda a rede, via peer-to-peer. Então, os nós de consenso verificam essa nova transação afim de validá-la. Finalmente, após um consenso bem-sucedido, esta transação é inserida em um bloco [NGD 2021].

4. Experimentos na Rede Pública-Permissionada Neo ALODE

Experimentos foram realizados na rede Neo ALODE para a transferência de ativos utilizando um microcontrolador ESP8266, que possui WiFi integrado. Nos testes, a programação do microcontrolador foi feita usando a IDE do Arduino utilizando a comunicação via cabo micro-usb. Também foi utilizada a API do neoscan [COZ 2021] para acessar dados do explorer (também disponibilizado publicamente pela rede ALODE). Todos os dados são fornecidos por meio de solicitações GET, como no exemplo abaixo:

```
http://alode.ic.uff.br:4000/api/main_net/v1/get_balance
/AK2nJJpJr6o664CWJKi1QRXjqeic2zRp8y
```

A solicitação realizada foi *get address last transactions* que retorna os últimos 15 modelos de transação na cadeia para o endereço selecionado. Desta forma foi possível obter de forma prática o *hash* e o *index* da transação que deu origem ao fundo (UTXO) e o *hash* do ativo. Veja um exemplo na Figura 2.

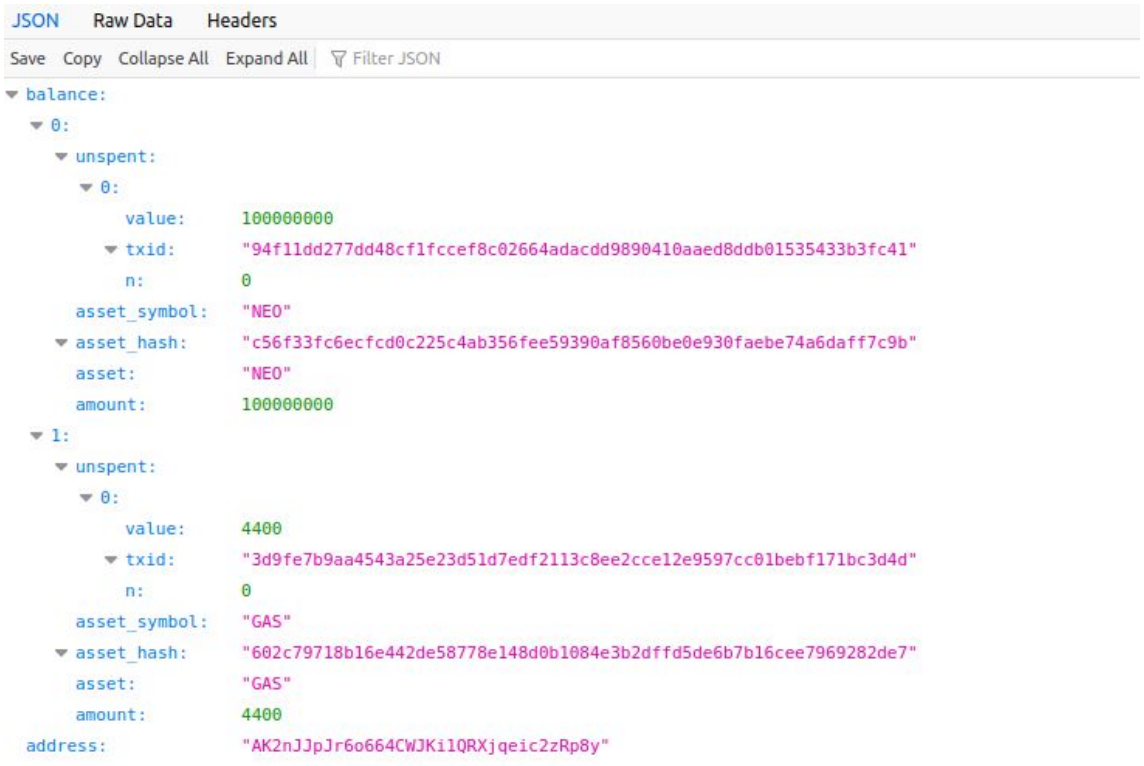

Figura 2. Resposta JSON no NeoScan para *get address last transaction*

Munido de todos os dados do cabeçalho como *type*, *version*, *attribute*, *input* e *output* foi gerado um SHA256. A resposta obtida do SHA256 foi assinada pela testemunha utilizando a criptografia de curva elíptica disponível para microcontroladores [MacKay 2021]. Para a realização do *InvocationScript* foi necessário informar o número de testemunhas, o tamanho do *InvocationScript* (total de bytes), o tamanho da

assinatura e a assinatura. Para o *VerificationScript* foi necessário o tamanho do *VerificationScript* (total de bytes), o tamanho da chave pública, a chave pública de quem assinou e o opcode `0xac`.

Agora, com os dados do cabeçalho, a assinatura da testemunha e a chave pública da mesma, foi possível fazer uma requisição do tipo POST na API da neocompiler [Coelho and Coelho 2021]. O método utilizado para o envio foi o *sendrawtransaction*, que envia uma transação para a Blockchain e retorna um JSON com o *hash* da transação, caso todas as condições sejam satisfeitas.

4.1. Transação de IoV na rede ALODE

Efetuamos, a seguir, uma operação IoV de transferência própria (*self-transfer*) no valor de 100.000.000 NEO, saindo do endereço *AK2nJJpJr6o664CWJKi1QRXjqeic2zRp8y* (*scripthash big-endian e9eed8dc39332032dc22e5d6e86332c50327ba23*), com destino ao próprio endereço (estratégia adotada pela facilidade de reprodução em testes sucessivos). O Código 1 apresenta o JSON RPC com a transação enviada para a rede.

```
{ "jsonrpc": "2.0", "id": 5, "method":
   "sendrawtransaction", "params": [
   "8000000194f11dd277dd48cf1fccef8c02664adacdd9890410aaed
8ddb01535433b3fc410000019b7cffdaa674beae0f930ebe6085af90
93e5fe56b34a5c220ccdcf6efc336fc50000c16ff286230023ba2703
c53263e8d6e522dc32203339dcd8eee9014140495f8d7262c446e5f6
1455a386f5350c3004fb8e4a47d3ccb1d5484ce05ff71dc0ebe9ccfd
37e46284d1c0f92ec462fca659ed8fbeeaf1d437229e1b6022a29e23
21031a6c6fbbdf02ca351745fa86b9ba5a9452d785ac4f7fc2b7548c
a2a46c4fcf4aac"]
}
```

Código 1. JSON de envio de transação via RPC *sendrawtransaction*

Observamos que, após o envio da transação, caso a mesma esteja corretamente assinada, ela é propagada na rede e o resultado é visualizado no arduino através de uma chamada posterior à API do neoscan. Devido ao modelo UTXO adotado no Neo v2 Legacy, a consolidação de saldo nas carteiras ocorre tipicamente através de nós de RPC da rede, que agregam transações não gastas de um usuário específico (através de seu endereço/*scripthash*). Isso permite que uma carteira local gere transações estruturalmente corretas e devidamente assinadas, sem a necessidade de sincronizar toda a cadeia de blocos, o que seria impraticável para um dispositivo bastante simples como o ESP8266.

4.2. Desafios encontrados

Dentre os diversos desafios encontrados, percebemos que ao incrementar a carga de bibliotecas e códigos complexos no Arduino, ele frequentemente trava devido à expiração de relógios internos de controle (conhecidos como *watchdogs*), exigindo uma série de precauções como liberações dos contadores através de métodos *yield()*.

Felizmente foi possível efetuar todo o processo inteiramente no Arduino, desde as chamadas à API do neoscan para consulta dos UTXOs, até a formação da transação e

posterior assinatura digital, embora esse processo tenha exigido recursos cada vez maiores do equipamento. Efetivamente, foram consumidos $309,408$ bytes ($226,019$ em formato comprimido), e a gravação do programa final levou $19,9$ segundos.

Após a gravação do programa no micro-controlador, foi efetuada uma transação na rede ALODE. A transação levou aproximadamente três segundos para ser formada, assinada e enviada, através do endpoint RPC `http://alode.ic.uff.br:30337`. O protocolo $http$ é tipicamente utilizado para fins de transmissão RPC na blockchain do Neo, dado que as transações são públicas e com não-repúdio garantido via assinatura digital. No contexto de micro-controladores, uma vantagem do uso de $http$ em relação ao $https$ é o seu menor custo de processamento, pois certificados digitais não são necessários.

A rede ALODE é uma rede experimental blockchain público-permissionada com quatro nós de consenso, dividida (no momento da escrita) entre duas máquinas hospedadas na infraestrutura de pesquisa em pós-graduação de instituições do estado do Rio de Janeiro: (i) Universidade do Estado do Rio de Janeiro, através de uma máquina virtual com parceria do Programa de Pós-Graduação em Ciências Computacionais e apoio do Laboratório do IME (LabIME), e (ii) Universidade Federal Fluminense, através de uma máquina virtual com parceria do Laboratório de Inteligência Computacional (LabIC-UFF). A rede está configurada para blocos de cinco segundos, portanto a transação só é confirmada definitivamente após um ciclo de consenso da rede (aproximadamente cinco segundos mais o tempo de transmissão de três segundos).

A execução do programa ocorreu dentro do limite máximo estipulado de 10 segundos, se mostrando promissora para aplicações práticas dentro do escopo considerado. Vale ressaltar que, durante o desenvolvimento do trabalho, equipamentos mais simples como o Arduino Uno foram descartados por não conseguirem assinar digitalmente em tempo hábil (tempo superior a 10 segundos), o que inviabilizaria as demais etapas de formação formação e submissão de transações.

5. Conclusões e Trabalhos Futuros

Neste trabalho, abordamos a estrutura de uma transação na Neo Blockchain e experimentamos com uma rede local a capacidade de transferência de valor de dispositivos de baixo consumo energético. Essa prova de conceito demonstra a capacidade de dispositivos IoT de baixo custo se tornarem dispositivos IoV, com capacidade de interação com uma rede público-permissionada hospedada em um ambiente controlado.

Dentre os desafios encontrados pelo caminho, elencamos dois principais: limitação dos equipamentos utilizados e falta de informações técnicas disponíveis em inglês ou português (para a Neo Blockchain). Observamos que a tecnologia Blockchain ainda se encontra em fase de amadurecimento, com diversas mudanças em andamento (como o lançamento futuro da rede Neo v3 prevista para 2021). Isso traz desafios, porém também novas oportunidades, dentre as quais esperamos que esse artigo possa ajudar também através de novas informações para o público brasileiro.

Nos experimentos, o dispositivo utilizado ESP8266 apresentou capacidade computacional suficiente para realização da proposta, formando uma transação, interagindo com nós de uma rede público-permissionada, assinando e enviando a transação para

a rede. Como trabalhos futuros, esperamos expandir a proposta para outros tipos de transação, especialmente transações ligadas à execução de contratos inteligentes e votação na rede (para escolha de nós validadores), de forma a fornecer um arcabouço capaz de abarcar aplicações de interesse no escopo de IoV, especialmente aquelas com interesse regional/nacional.

Referências

Castro, M., Liskov, B., et al. (1999). Practical byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186.

Coelho, I. M. and Coelho, V. N. (2021). Neocompiler eco: experimentação de consenso em blockchain e contratos inteligentes. In *Anais do VI Workshop do testbed FIBRE*, pages 57–67. SBC.

Coelho, V. N., Oliveira, T. A., Tavares, W., and Coelho, I. M. (2021). Smart accounts for decentralized governance on smart cities. *Smart Cities*, 4(2):881–893.

COZ (2021). Neo scan. `https://neoscan.neocompiler.io/`. acessado em 25/06/2021.

da Silva, V. F., Coelho, M. N., Coelho, B. N., Coelho, V. N., and Coelho, I. M. (2019). A home ledger approach for iot enabled devices. In *2019 31st International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, pages 227–233. IEEE.

Davies, J. H. (2008). *MSP430 microcontroller basics*. Elsevier.

Di Pierro, M. (2017). What is the blockchain? *Computing in Science Engineering*, 19(5):92–95.

Gabrich, Y. B., Coelho, I. M., and Coelho, V. N. (2020). Sharing electricity in brazil: a crypto-currency for micro/mini-grid transactive energy. In *2020 6th IEEE International Energy Conference (ENERGYCon)*, pages 973–978. IEEE.

Hongfei, Da and Zhang, Erik (2015). Neo: A distributed network for the smart economy. `https://github.com/neo-project/docs/blob/master/en-us/whitepaper.md`.

ISO, I. S. O. (2020). Blockchain and distributed ledger technologies 22739:2020. https://www.iso.org/obp/ui#iso:std:iso:22739:ed-1:v1:en (acessado em 5 de Maio de 2021).

Kamran, M., Khan, H. U., Nisar, W., Farooq, M., and Rehman, S.-U. (2020). Blockchain and internet of things: A bibliometric study. *Computers & Electrical Engineering*, 81:106525.

Khan, P. W., Byun, Y.-C., and Park2, N. (2020). A data verification system for cctv surveillancecameras using blockchain technology insmart cities. *Eletronics*.

Lamport, L., Shostak, R., and Pease, M. (2019). The byzantine generals problem. In *Concurrency: the Works of Leslie Lamport*, pages 203–226.

MacKay, K. (2021). Micro-ecc library. `https://github.com/kmackay/micro-ecc`. acessado em 25/06/2021.

Malamud, S. and Rostek, M. (2017). Decentralized exchange. *American Economic Review*, 107(11):3320–62.

Martini, B. and Choo, K.-K. R. (2013). Cloud storage forensics: owncloud as a case study. *Digital Investigation*, 10(4):287 – 299.

Monteiro, E. S., Mignoni, M. E., Righi, R. R., da Costa, C. A., Kunst, R., and Alberti, A. (2021). Combinando internet das coisas, inteligência artificial e blockchain para monitorar a cadeia de agroquímicos. In *Anais do XIII Simpósio Brasileiro de Computação Ubíqua e Pervasiva*, pages 61–70. SBC.

Nakamoto, S. (2008). Bitcoin:a peer-to-peer electronic cash system.

NGD (2021). Neo–tutorial: Neo global development. `https://neo-ngd.github.io/NEO-Tutorial/`. acessado em 25/06/2021.

NIST (2013). Digital signature standard (dss). `https://csrc.nist.gov/publications/detail/fips/186/4/final`.

Nofer, M., Gomber, P., Hinz, O., and Schiereck, D. (2017). Blockchain. *Business & Information Systems Engineering*, 59:187–183.

Silvano, W. F. and Marcelino, R. (2020). Iota tangle: A cryptocurrency to communicate internet-of-things data. *Future Generation Computer Systems*, 112:307–319.

Szabo, N. (1997). Formalizing and securing relationships on public networks. *First monday*.

Truong, N. B., Um, T., Zhou, B., and Lee, G. M. (2018). Strengthening the blockchain-based internet of value with trust. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7.

Weber, R. H. and Weber, R. (2010). *Internet of things*, volume 12. Springer.

Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. Technical report.

Zhang, E. (2021). Transaction details for neo v2 legacy. `https://docs.neo.org/v2/docs/en-us/tooldev/transaction/transaction.html`. acessado em 25/06/2021.

Zia, M. F., Benbouzid, M., Elbouchikhi, E., Muyeen, S., Techato, K., and Guerrero, J. M. (2020). Microgrid transactive energy: Review, architectures, distributed ledger technologies, and market analysis. *Ieee Access*, 8:19410–19432.