



Gerenciamento de configurações chave-valor orientado à Internet das Coisas

Palavras-Chave: SISTEMAS DISTRIBUÍDOS, GERENCIAMENTO DE CONFIGURAÇÕES, INTERNET DAS COISAS

Autores:

LUCAS VALENTE VIEGAS DE OLIVEIRA PAES – IC, UNICAMP

Prof^(a). Dr^(a). LUIZ FERNANDO BITTENCOURT (orientador(a)) – , UNICAMP

INTRODUÇÃO:

A invenção da Internet das Coisas (IoT) somada ao rápido progresso na área de sistemas distribuídos apontam para a transformação do modo de vida nas próximas décadas. É muito provável que, em um futuro próximo, casas, indústrias e cidades serão automatizadas através de sensores e atuadores. No entanto, para que tal cenário seja possível, ainda há a necessidade de investigar certos obstáculos técnicos, dentre eles, o de gerenciamento, propagação e replicação de configurações de tais dispositivos.

Nesse contexto, o presente projeto de Iniciação Científica estuda como os Conflict-Free Replicated Datatypes (CRDTs) [1], uma espécie de estrutura de dados distribuída, se comporta nessa situação. Para tal, foi especificado um protocolo básico de sincronização de configurações em uma rede IoT, que foi simulado no simulador de rede *ns-3*.

METODOLOGIA:

O trabalho foi desenvolvido em seis etapas, elaboradas a seguir.

1. LEVANTAMENTO DE CASOS DE USO

Com o objetivo de entender melhor o problema em questão, foi feito um levantamento de casos de uso com base na literatura sobre o assunto antes de desenvolver-se uma solução. Os três principais casos de uso identificados foram: gerenciamento automático de ar-condicionado central via detecção de presença [2], controle integrado de dispositivos em casa inteligente [3], monitoramento de desempenho de aparelhos industriais [4, 5].

2. ESTUDO DE TÉCNICAS DE DESENVOLVIMENTO DE SISTEMAS DISTRIBUÍDOS

Além dos casos de uso para IoT, também foram exploradas técnicas comumente usadas em sistemas distribuídos, visando compreender melhor erros comuns a eles associados, bem como suas características particulares. Especificamente, foram estudados problemas de consistência,

disponibilidade e tolerância a particionamento [6], o CRDT LWW-Set [1], que implementa um conjunto de elementos que pode ser distribuído em vários dispositivos sem a necessidade de um algoritmo explícito de coordenação, e o algoritmo RAFT para coordenação entre réplicas em sistemas distribuídos.

3. ESPECIFICAÇÃO INICIAL DE UM PROTOCOLO DE GERENCIAMENTO DE CONFIGURAÇÕES

Após os estudos iniciais, foi desenvolvido um rascunho inicial de um protocolo de comunicação entre dispositivos na rede. O protocolo define dois tipos de dispositivos, *Fornecedor* e *Consumidor* de configurações. Os Fornecedores são responsáveis por armazenar e prover configurações indexadas por chaves (nomes) aos Consumidores em uma rede utilizando o protocolo CoAP. Os Consumidores podem requisitar ou modificar valores de configuração nos Fornecedores através de sua chave. Assumiu-se que a rede utilizada era 100% segura para evitar preocupações iniciais com segurança, criptografia e outras complexidades.

Após a definição geral do protocolo, sua implementação foi dividida em quatro etapas:

- Handshake de conexão: início do ciclo de vida do protocolo, durante o qual um Consumidor que entra na rede anuncia sua entrada;
- Requisição por fornecedores: o consumidor que acaba de entrar na rede solicita pelos fornecedores disponíveis;
- Requisição por configurações: o consumidor requisita uma ou mais chaves de configuração a um fornecedor;
- Atualização de configurações: um fornecedor informa outro fornecedor sobre a disponibilidade de novos valores para determinada chave. O recipiente pode escolher propagar ou não essa mensagem pela rede.

4. IMPLEMENTAÇÃO DO SIMULADOR

Depois de especificar formalmente o protocolo em termos de requisições CoAP, foram desenvolvidas simulações utilizando o simulador *ns-3* [7].

Inicialmente, o protocolo foi implementado sobre UDP ao invés de CoAP, dado que não há uma implementação de tal protocolo no *ns-3* e desenvolvê-lo do zero não é trivial. Portanto, as primeiras simulações não contam com confiabilidade de entrega de pacotes.

A implementação consiste em três principais componentes

- Função principal: responsável por inicializar a terminar a simulação, configurando a estrutura geral da rede, hiperparâmetros de simulação e *tracing*;
- Definição dos pacotes: implementação de pacotes *ns-3* que carregam os metadados necessários do protocolo. O pacote possui um header que armazena se o pacote é requisição ou resposta, identificador do tipo do pacote (solicitação de configuração, atualização de configuração, listagem de peers). O corpo do pacote varia dependendo de seu tipo, podendo carregar valores de chaves e lista de peers.
- Aplicação: instalada em cada um dos nós disponíveis na rede simulada, podendo ser um Consumidor ou um Fornecedor ao mesmo tempo. A aplicação mantém um conjunto de peers e

chaves conhecidos, implementando um CRDT LWW-Set. A aplicação implementa toda a lógica de requisição e resposta de pacotes de acordo com a especificação do protocolo.

As simulações foram parametrizadas com D (densidade, ou número de nós conhecidos por um nó na rede) e N (número total de nós na rede). O parâmetro D foi variado de 1 a 10 com N=50, enquanto N foi variado de 10 a 110, de 10 em 10, com D fixado em 2.

5. COLETA DE DADOS

A coleta de dados foi implementada a partir da geração arquivos de *trace* padronizados. Cada linha do arquivo de *trace* foi produzida no formato “tempo origem evento chave=valor”, onde “tempo” é o momento em que o evento foi emitido, “origem” é o identificador do nó de rede em que o evento foi emitido e chave e valor podem vários a depender do tipo do “evento”. Os eventos de simulação de acordo com cada parâmetro foram armazenados em arquivos independentes para análise posterior.

6. ANÁLISE DE DADOS

Foi desenvolvido um script de análise de dados que converte os arquivos de *trace* em métricas de performance do protocolo para cada cenário, percorrendo os eventos registrados linha por linha. Dessa forma, é possível encontrar possíveis relações entre os parâmetros de simulação e a performance.

As métricas coletadas foram:

- Tempo médio para que um valor de configuração chegue em um nó qualquer;
- Número médio de saltos para que um valor de configuração chegue em um nó qualquer;
- Número de nós que, ao final da simulação, possuíam determinado valor de configuração, bem como o número de nós não encontrados devido à perda de pacotes.

RESULTADOS E DISCUSSÃO:

Primeiramente, observou-se que o tempo de propagação total aumenta linearmente conforme a densidade de simulação aumenta. Isso ocorre provavelmente devido ao fato de que a quantidade de pacotes que trafega na rede aumenta significativamente, apesar do número de saltos de um nó a outro diminuir.

Além disso, nota-se que valores muito baixos ou muito altos do parâmetro D resultam em baixa confiabilidade. No primeiro caso, isso se deve à baixa redundância de caminhos na rede. No segundo, isso se deve ao excesso de congestionamento devido ao excesso de pacotes enviado. Dessa forma, estimou-se que um bom valor para o parâmetro é D=3.

CONCLUSÕES:

O estudo discute o problema da configuração de dispositivos IoT, apresentando um protocolo que se propõe a solucionar parte deste problema. O protocolo foi implementado e executado no

simulador de eventos discretos *ns-3*, e seu desempenho foi avaliado através de métricas concretas conforme a variação dos parâmetros experimentais.

Assim, foi possível compreender mais a fundo o comportamento de aplicações IoT, bem como estabelecer um ponto de partida inicial para o desenvolvimento de protocolos de finalidade similar no futuro.

BIBLIOGRAFIA

- [1] Shapiro, MARC; PREGUIÇA, Nuno; BAQUERO, Carlos; e ZAWIRSKI, Marek. **A comprehensive study of convergent and commutative replicated data types**. INRIA [Research Report] RR-7506, p. 14, 2014.
- [2] AGARWAL, Yuvraj et al. **Occupancy-driven energy management for smart building automation**. in Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building, New York, NY, USA, 2010, BuildSys '10, p. 1-6, Association for Computing Machinery.
- [3] **Home Assistant**. <https://www.home-assistant.io/>, acessado em 9 de Setembro de 2021.
- [4] LEE, Gil-Yong et al. **Machine health management in smart factory: A review**. Journal of Mechanical Science and Technology, vol. 32, no. 3, pp. 987-1009, Mar 2018.
- [5] **Tractian**. <https://tractian.com/>, acessado em 11 de Setembro de 2021.
- [6] BAILIS, Peter et al. **Highly available transactions: Virtues and limitations**. Proc. VLDB Endow., vol. 7, no. 3, pp. 181-192, nov 2013.
- [7] **ns-3 network simulator**. <https://www.nsnam.org/>, acessado em 16 de Dezembro de 2021.