

MONITORAMENTO REATIVO DE PROTOCOLOS EM SISTEMAS IOT SOBRE O MQTT

Gustavo Oliveira Dias

Universidade Estadual do Norte do Paraná
Bacharelado em Ciência da Computação

Orientador: Prof. Me. Wellington Aparecido Della Mura

Banca de Defesa do Trabalho de Conclusão de Curso
24 de novembro de 2017

Roteiro da Apresentação

- 1 Motivação;
- 2 Formulação e escopo do problema;
- 3 Objetivos;
- 4 Método proposto;
- 5 Estudo de Caso;
- 6 Conclusão;

O que é Internet das Coisas?

Aplicações que **armazenam** e **processam** dados que **representam virtualmente** objetos do mundo físico, provendo **interação** e **comunicação** entre dispositivos **heterogêneos** conectados.

- Em 2011, movimentou cerca de US\$44 bi;
- Crescimento previsto em US\$290 bi para 2017;
- Até 2020, haverá em torno de 34 bi de dispositivos.

Desafio: lidar com a **comunicação** entre tantos dispositivos conectados.

Protocolos de comunicação

Conjunto de **regras-padrão** para **transmissão de informação** entre computadores.

- Formato;
- Sincronização;
- Sequência;
- Detecção de erros e falhas.

MQTT (*Message Queue Telemetry Transport protocol*)

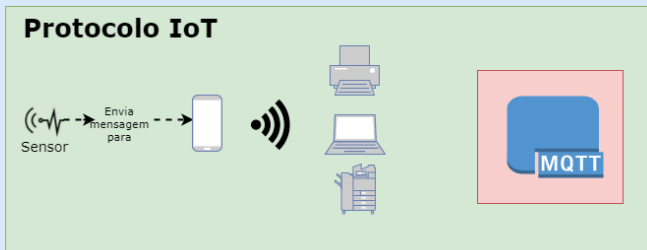
Ideal para dispositivos com **baixo poder de processamento** em ambientes com **largura de banda limitada**.

Características:

- Utiliza TCP;
- Cabeçalho de apenas 2 bytes;
- Estratégia *publish/subscribe*;
- Clientes se inscrevem em tópicos;
- Implementa um servidor próprio: *broker*.
- 3 níveis de *Quality of Service (QoS)*;

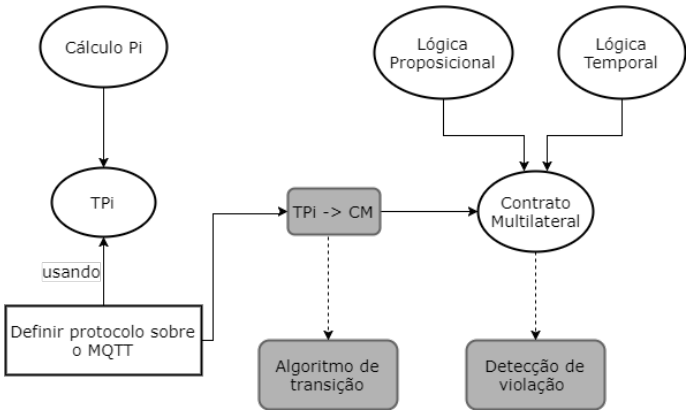
Motivação - Protocolos IoT

Aplicação IoT



É possível **detectar** e **identificar** o responsável por uma **violação** do protocolo?

Os protocolos IoT podem ser **modelados** por meio de **formalismos**?



Objetivo geral

Mecanismo para **formalização** e **monitoramento reativo** de sistemas de IoT que utilizam o protocolo **MQTT**.

Objetivos específicos:

- Definir **métodos** para representação de protocolos em **contratos multilaterais** a partir do **modelo** descrito em **TPi**;
- Representar o modelo em TPi do MQTT como um **contrato multilateral**;
- Aplicar o algoritmo de monitoramento a um estudo de caso real.

Representação formal - TPi

O que é o TPi?

Uma linguagem temporal para modelagem de processos.

QoS nível 0 "(*At most once*)" :

$Cliente(Publish) \mid Servidor() \mid ClienteSub()$, em que :

$Cliente(z) \stackrel{def}{=} \bar{c}\langle z \rangle$

$Servidor() \stackrel{def}{=} c(x).\overline{pub}\langle x \rangle$

$ClienteSub() \stackrel{def}{=} pub(y)$

Contratos multilaterais

Acordo entre duas ou mais partes baseado em **compromissos mútuos**.

Composto por:

- **Ação:** o que cada parte deverá fazer;

$(A_QoS0enviarPublish, C, S, ?)$

- **Compromisso:** garantia de cumprimento das ações entre as partes envolvidas;

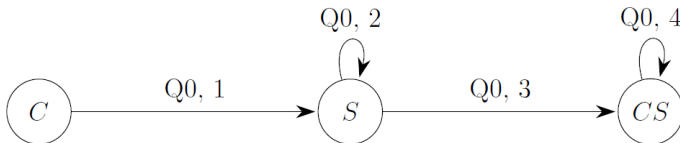
$c = (C_QoS0delivery, C, CS, 4,$
 $\{(A_QoS0enviarPublish, tr), (A_QoS0receberPublish, fi),$
 $(A_QoS0publicarPacote, tr), (A_QoS0receberPacote, fi)$

- **Grafo de compromisso:** forma de visualizar os compromissos entre as partes.

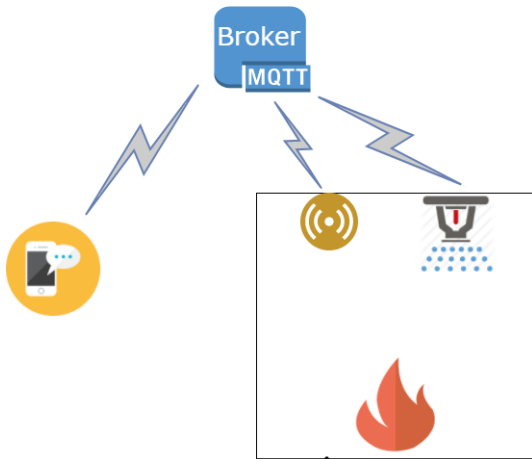
QoS 0 - "At most once"

A comunicação baseia-se em um *broker* **recebendo** e **publicando** mensagens.

| Compromissos | Classificação das ações e compromissos | | Rótulos |
|-------------------------------|--|-----------------------------|---------|
| | <i>tr</i> | <i>fi</i> | |
| <i>C_QoS0delivery</i> (Q0) | <i>A_QoS0enviarPublish</i> | | Q0, 1 |
| | | <i>A_QoS0receberPublish</i> | Q0, 2 |
| | <i>A_QoS0publicarPacote</i> | | Q0, 3 |
| | | <i>A_QoS0receberPacote</i> | Q0, 4 |



Sistema IoT de **detecção** e **supressão** de incêndio.



Gustavo Oliveira
Dias

Motivação do
Trabalho

IoT
Protocolos
MQTT

Formulação e
escopo do
problema

Estrutura

Objetivos

Método Proposto

TPI
Contratos multilaterais

Estudo de caso

Modelagem e conversão
Aplicação do algoritmo

Conclusão

| Mensagem | Tópico MQTT | Publisher | Subscriber | Ação do subscriber |
|--------------------------|------------------------|--------------------|------------------|-------------------------|
| <i>Fire Detection</i> | <i>Fire/Detected</i> | Sensor de incêndio | <i>App</i> | Exibir notificação |
| <i>Sprinkler Request</i> | <i>Sprinkler/StReq</i> | <i>App</i> | <i>Sprinkler</i> | Ler status |
| <i>Sprinkler Reply</i> | <i>Sprinkler/StRep</i> | <i>Sprinkler</i> | <i>App</i> | Enviar mensagem |
| <i>Sprinkler Start</i> | <i>Sprinkler/Start</i> | <i>App</i> | <i>Sprinkler</i> | Ativar <i>sprinkler</i> |
| <i>Sprinkler Start</i> | <i>Sprinkler/Start</i> | Sensor de incêndio | <i>Sprinkler</i> | Ativar <i>sprinkler</i> |

O sistema é baseado em três processos, que são **modelados** e convertidos em um **contrato**.

- Detecção/Alerta;
 - Checagem;
 - Supressão;
-
- O **aplicativo** assume o papél **Ap** durante a detecção/alerta e checagem, e recebe o papel **Ap'** no compromisso de supressão;
 - Os agentes **broker**, **sprinkler** e **sensor** assumem os papéis **Br**, **Sp** e **S**, respectivamente.

Estudo de caso - Modelo para Detecção/Alerta

$Sensor(FireDetection) \mid Broker() \mid App() \mid Sprinkler() :$

$Sensor(z) \stackrel{def}{=} \overline{fd}\langle z \rangle$

$Broker() \stackrel{def}{=} fd(x).\overline{fd'}\langle x \rangle$

$App() \stackrel{def}{=} fd'(y)$

$Sprinkler() \stackrel{def}{=} fd'(y')$

Estudo de caso - Conversão do modelo de Detecção/Alerta

Conjunto de ações : $\{$ $(\overline{fd}\langle z \rangle, S, Br, ?),$
 $(fd(x), Br, Br, ?),$
 $(\overline{fd'}\langle x \rangle, Br, Ap, ?),$
 $(fd'(y), Ap, Ap, ?),$
 $(\overline{fd'2}\langle x \rangle, Br, Sp, ?),$
 $(fd'2(y'), Sp, Sp, ?)\}$

Compromisso:

$$m_1 = (C_detectarIncendio, Sensor, Sprinkler, 6,$$
$$\{(\overline{fd}\langle z \rangle, tr), (fd(x), fi), (\overline{fd'}\langle x \rangle, tr), (fd'(y), fi),$$
$$(\overline{fd'2}\langle x \rangle, tr), (fd'2(y'), fi) \})$$

Estudo de caso - Conversão do modelo de Detecção/Alerta

- 1 Atribuir um nome ao compromisso;

$$name = C_detectarIncendio$$

- 2 Definir o agente e a mensagem que dão início ao processo;

$$P_{ini} = (Sensor, FireDetection)$$

- 3 Identificar todos os agentes.

$$\mathcal{P} = \{Sensor, Broker, App, Sprinkler\}$$

Estudo de caso - Conversão do modelo de Detecção/Alerta

- 4 Agrupar duplas compostas por cada agente e sua definição;

$$\mathcal{D} = \{(Sensor, \{\overline{fd}\langle z \rangle\}), \\ (App, \{fd'(y)\}), (Sprinkler, \{fd'(y')\})\}$$

- 5 Identificar a ordem de execução das ações.

$$\mathcal{A} = \{(\overline{fd}\langle z \rangle, Sensor), (fd(x), Broker), (\overline{fd'}\langle x \rangle, Broker), \\ (fd'(y), App), (fd'(y'), Sprinkler)\}$$

Estudo de caso - Conversão do modelo de Detecção/Alerta

- Quando uma mensagem é publicada para mais de um inscrito, então novas ações de saída são adicionadas;
- Ações de entrada podem ser modificadas.

Conjunto de ações : $\{$ $(\overline{fd}\langle z \rangle, S, Br, ?),$
 $(fd(x), Br, Br, ?),$
 $(\overline{fd'}\langle x \rangle, Br, Ap, ?),$
 $(fd'(y), Ap, Ap, ?),$
 $(\overline{fd'2}\langle x \rangle, Br, Sp, ?),$
 $(fd'2(y'), Sp, Sp, ?)\}$

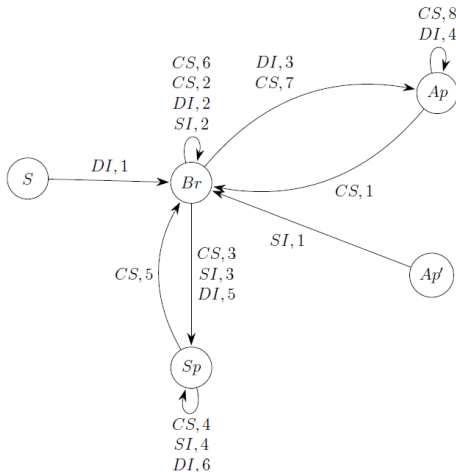
Estudo de caso - Conversão do modelo de Detecção/Alerta

- O agente contido em $P_{ini} = (Sensor, FireDetection)$ é o **expedidor** do compromisso;
- O agente que envia a última ação é o **recebedor**;
- Ações de saída são do tipo **trigger** e ações de entrada são do tipo **finish**.

Compromisso:

$$m_1 = (C_detectarIncendio, Sensor, Sprinkler, 6, \\ \{(\overline{fd}\langle z \rangle, tr), (fd(x), fi), (\overline{fd'}\langle x \rangle, tr), (fd'(y), fi), \\ (\overline{fd'2}\langle x \rangle, tr), (fd'2(y'), fi) \})$$

Estudo de caso - Grafo de compromisso



$$ordercommitment = \{(m_2), (m_1 \cdot m_2), (m_1 \cdot m_2 \cdot m_3)\}$$

Estudo de caso - Cenário da violação

- Um incêndio foi detectado pelo sensor;
- O aplicativo verificou que o status do *sprinkler* consta como **desativado**;
- O compromisso de Supressão é disparado;
- Porém, o *sprinkler* continua **desativado**.

Estudo de caso - Aplicação do algoritmo

1 Entradas:

- Contrato;
- Ação perdida: $a_{miss} = A_receberAtivacao(Sl_2)$
- Última ação feita: $a_{done} = A_publicarAtivacao(Sl_1)$

2 Localizar todas as ações que **não ocorreram** após a violação;

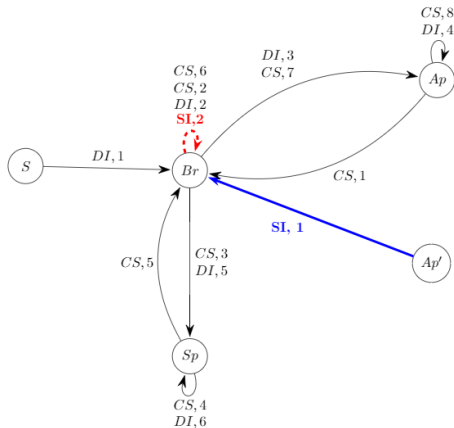
$$E_{not_occur} = \{Sl_3, Sl_4\}$$

3 Encontrar todas as ações que **ocorreram** antes da violação;

$$E_{cut} = \{DI_1, DI_2, DI_3, DI_4, DI_5, DI_6, CS_1, CS_2, CS_3, CS_4, CS_5, CS_6, CS_7, CS_8\}$$

4 Montar o grafo restruturado.

- Se a mensagem "*StartSprinkler*" enviada por Ap' ao $Br(SI, 1)$ estiver incorreta, então **Ap'** é o responsável;



Gustavo Oliveira
Dias

Motivação do
Trabalho

IoT
Protocolos
MQTT

Formulação e
escopo do
problema

Estrutura

Objetivos

Método Proposto

TPI
Contratos multilaterais

Estudo de caso

Modelagem e conversão
Aplicação do algoritmo

Conclusão

- Proposta: método para monitoramento reativo de protocolos IoT sobre o MQTT;
- É importante que violações sejam detectadas, assim como localizar o responsável;
- A solução proposta consiste em:
 - 1 Descrever o modelo por meio do TPI;
 - 2 Desenvolver um método de conversão para representação em contrato multilateral;
 - 3 Aplicar o algoritmo de monitoramento;
- O que foi feito:
 - 1 Aplicação da proposta para comunicação QoS 0.

Gustavo Oliveira
Dias

Motivação do
Trabalho

IoT
Protocolos
MQTT

Formulação e
escopo do
problema
Estrutura

Objetivos

Método Proposto

TPI
Contratos multilaterais

Estudo de caso
Modelagem e conversão
Aplicação do algoritmo

Conclusão

■ Questões pendentes:

- 1 Representação em TPI dos papéis, suas propriedades;
- 2 Representação das ordens de execução dos compromissos;

Trabalhos futuros:

- Determinar meios para representação em TPI das propriedades pendentes;
- Adaptar o método de conversão para casos mais complexos;
- Estender o método proposto para os níveis 1 e 2 de QoS;
- Desenvolver um **broker** MQTT para implemente o monitoramento reativo.

MONITORAMENTO REATIVO DE PROTOCOLOS EM SISTEMAS IOT SOBRE O MQTT

Gustavo Oliveira Dias

Universidade Estadual do Norte do Paraná
Bacharelado em Ciência da Computação

Orientador: Prof. Me. Wellington Aparecido Della Mura

Banca de Defesa do Trabalho de Conclusão de Curso
24 de novembro de 2017

QoS Level 0 Protocol:

$Client(Publish) \mid Server()$, where:

$$Client(z) \stackrel{\text{def}}{=} \bar{c}\langle z \rangle$$

$$Server() \stackrel{\text{def}}{=} c(x).\overline{pub}\langle x \rangle$$

QoS Level 1 Protocol:

$Client(Publish) \mid Server()$, where:

$$Client(z) \stackrel{\text{def}}{=} \bar{c}\langle z \rangle.\text{timer}^t(c'(y), Client(Publish_{DUP}))$$

$$Server() \stackrel{\text{def}}{=} !c(x).\overline{pub}\langle x \rangle.\bar{c}'\langle Puback \rangle$$

Gustavo Oliveira
Dias

Motivação do
Trabalho

IoT
Protocolos
MQTT

Formulação e
escopo do
problema

Estrutura

Objetivos

Método Proposto

TPi
Contratos multilaterais

Estudo de caso

Modelagem e conversão
Aplicação do algoritmo

Conclusão

$$P, Q ::= \bar{x}(y).P \mid \mathbf{timer}^t(x(y).P, Q) \mid !P \mid (\nu x)P \mid (P \mid Q) \mid (P + Q) \mid \mathbf{0} \mid A(x)$$