Review of the IoT Protocol Literature at the Application Layer

Felipe M. D. dos Santos, Ronaldo P. Borges, Rafael A. Leite.

***Resumo* - A *Internet of Things* (IoT) tem como finalidade a conexão entre usuários com objetos e dispositivos diversos, como aparelhos pessoais (smartphones, notebooks, smartwatches, tablets etc), equipamentos eletrônicos embutidos com sensores e outros elementos ambientais. Que estejam sempre conectados a uma rede comum. Embora a IoT traga diversos benefícios, também tem gerado aumento de tráfego na rede nos últimos anos, afetando a infraestrutura desta mesma rede. Além disso, a implementação de dispositivos apresenta muitas dificuldades, riscos e questões de segurança que devem ser considerados. Para reduzir tais problemas é essencial entender quais protocolos são mais adequados para cada solução. Levando em consideração esses problemas, este artigo apresenta uma revisão dos protocolos da camada de aplicação a fim de escolher o melhor protocolo para ser aplicado em uma plataforma de monitoramento de gás de cozinha.**

***Palavras-chave*—Protocolos IoT, Camada de Aplicação, CoAP, MQTT, XMPP, WebSocket.**

I. INTRODUÇÃO

A INTERNET das Coisas tem como principal finalidade a conexão entre usuários com objetos e dispositivos diversos, como atuadores, veículos, smartphones, eletrodomésticos, brinquedos, câmeras, equipamentos médicos, sistemas industriais, prédios, entre outros [1].

A previsão é que mais de 75 bilhões de aparelhos estejam conectados até o final de 2025 [2], como mostra a Fig. 1 [2, Fig. 1]. Este aumento de tráfego impulsiona carga na rede existente e na infraestrutura de hardware. Para reduzir a carga sobre a rede e sua infraestrutura é essencial entender quais protocolos são mais adequados para cada solução.

As redes de computadores funcionam com uma arquitetura em camadas e existem diversos protocolos para IoT que atuam de formas e em camadas diferentes. Essas camadas são conhecidas como a pilha de protocolos *Transmission Control Protocol/Internet Protocol* (TCP/IP) que, dependendo da finalidade e meios de conexão, os protocolos são combinados de formas diferentes [3].

Para usar uma página web, por exemplo, um navegador de internet usa o *HyperText Transfer Protocol* (HTTP) para receber as páginas e a combinação de protocolos na pilha pode ser HTTP, TCP, IP e *Ethernet* [4], cada protocolo em sua respectiva camada, como pode ser visto no Tabela 1, que apresenta uma referência do modelo TCP/IP e os protocolos IoT mais usados na camada de aplicação, conforme mostrado na Tabela 2.

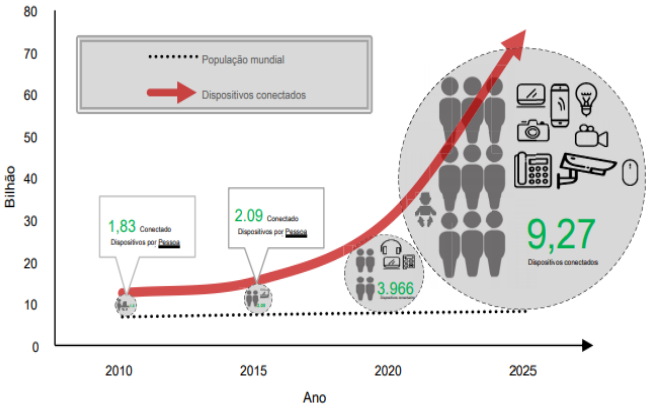

Figura 1. Estimativa de dispositivos conectados por pessoa até 2025.

Pode-se tomar como exemplo de solução da internet das coisas um dispositivo medidor de gás de cozinha, por meio do qual é possível realizar o monitoramento do consumo em tempo real via internet e gerar informações, tais como previsão de troca, custo por preparo, entre outras [Fig 2]. O dispositivo em comento pode: I - Utilizar rede de dados móveis e tornar relevante o consumo de dados de uma franquia; II - Utilizar uma rede *Wireless Fidelity* (Wi-Fi) e trazer a preocupação com a segurança de dados nas redes de computadores; e, III - Utilizar baterias e tornar pertinente a análise do gasto energético.

Para escolher o protocolo mais adequado para uma solução torna-se necessário entender os requisitos da mesma e o que cada um dos protocolos pode oferecer.

No caso da solução em questão, segundo [3], o software da plataforma utilizada atua apenas na camada de aplicação onde o protocolo é distribuído por diversos sistemas finais e utiliza o protocolo para a troca de pacotes de dados com a aplicação em outro sistema final, podendo ser uma pilha dos protocolos *Constrained Application Protocol* (CoAP), *User Datagram Protocol* (UDP), IP e 802.11. Quando o desenvolvedor cria sua solução ele tem controle total na camada de aplicação e apenas utiliza os serviços das outras

camadas [3], portanto, neste estudo não é necessário verificar os protocolos que atuam em outras camadas do TCP/IP.

Fig 2. Dispositivo para o monitoramento do Gás de Cozinha.

TABELA I
MODELO TCP/IP E SEUS PRINCIPAIS PROTOCOLOS

Camadas	Protocolos			
Aplicação	MQTT	CoAP	AMQP	HTTP
	REST	XMPP		
Transporte	TCP	UDP		
Rede	IP	ICMP		
Enlace de Dados/Física	DSL	802.11	SONET	Ethernet

Neste contexto, este artigo tem como **objetivo** realizar uma revisão de literatura sobre protocolos de rede da camada de aplicação utilizados em IoT com o intuito de elucidar qual é o ideal para este tipo de solução.

Considerando custos de infraestrutura como serviços de internet e equipamentos, a **relevância** desta investigação está em fornecer uma visão ampla e sistemática sobre as opções disponíveis atualmente.

II. METODOLOGIA

Há inúmeras formas de expressar conhecimento sobre um determinado assunto. Esta é uma pesquisa de cunho descritivo, posto que, conforme o descrito em [5], identifica, descreve e categoriza modelos teóricos e empíricos, apontados na literatura científica, sendo que faz-se uma análise de cada elemento constitutivo do assunto e sua relação com o todo, através de uma Revisão Sistemática de Literatura (RSL). Tal análise engloba descrição, classificação e definição do assunto, tendo em vista a estrutura, a forma, o objetivo e a finalidade do tema [6].

Para que os objetivos fossem atingidos perante os dados obtidos, optou-se por uma metodologia de RSL que se constitui de 3 (três) das 4 (quatro) fases distintas e complementares, apresentadas por [7] para análise de literatura científica.

(I) Avaliação da fonte de informação: Nessa fase, foi escolhida a base de dados IEEE (ieeexplore.ieee.org), pois é a base com enfoque nos periódicos e publicações de áreas e de disciplinas relacionadas ao desenvolvimento de recursos tecnológicos e a coleta de dados em ambientes digitais [8].

(II) Pesquisa e localização na literatura dos recursos de informação: As palavras chave utilizadas nos comandos de busca referentes ao tema foram *IoT and protocols and comparison and "application layer",* aplicados a base de dados IEEE, considerando todos os anos. Foram encontrados 17 (dezessete) artigos, após a leitura, 9 (nove) foram excluídos e 8 (oito) foram selecionados para compor a revisão bibliográfica.

(III) Escrita da literatura selecionada: Nessa fase, os artigos foram analisados e agrupados em uma tabela, (Tabela II) delimitando quais protocolos cada autor avaliou e que tipo de testes foram feitos. Nos resultados pode-se fazer uma análise geral dos resultados que cada autor obteve, conforme os critérios largura de banda [9],[10],[14], segurança [2],[15],[16] tempo [9],[10],[14] e consumo de energia [16],[18].

TABELA II
CARACTERIZAÇÃO DO ACERVO DE REVISÃO, SEGUNDO AUTOR, ANO, PROTOCOLOS E TIPOS DE TESTES ADOTADOS

Autor/ano	**Protocolos**	**Tipos de Testes**
Deva P. Seetharam, 2017	CoAP, MQTT e REST	Largura de banda e tempo gasto.
Syed Rameez Ullah Kakakhel et al., 2019	MQTT, HTTP, CoAP e XMPP	Conexão, desempenho, segurança, atributos operacionais, suporte de mensagem e carga útil.
Lavinia N stase, 2017	CoAP, MQTT e XMPP	Segurança.
Bardia Safae et al., 2017	CoAP e MQTT	Segurança.
Paridhika Kayal & Harry Perros, 2017	CoAP, MQTT, XMPP e WebSocket	Tempo de resposta variando a carga de tráfego.
Guilherme M. B. Oliveira et. al., 2018	MQTT e WebSocket	Tempo de resposta e consumo.
Thays Moraes et al., 2019	AMQP, CoAP e MQTT	Tamanho da mensagem e perda de pacotes
Kanchana P. & Naik Rakesh Joshi U, 2018	CoAP e HTTP	Consumo de energia e tempo de resposta

III. PROTOCOLOS

CoAP

O CoAP foi desenvolvido pelo Grupo de Trabalho *RESTful Environment* (CoRE) da *IETF Constraint*, especificamente para dispositivos restritos como microcontroladores com menos de 1 *kilobyte* (kB) de *Random Access Memory* (RAM) [9]. Isso se deve ao emprego do protocolo UDP em vez do protocolo TCP na pilha de protocolos do CoAP e ao mapeamento de campos string existentes de HTTP em campos com dimensões tão pequenas quanto um bit. Consequentemente, os pacotes CoAP são muito menores do que os pacotes HTTP [2].

Com o CoAP os clientes podem enviar solicitações de recursos GET, PUT, POST e DELETE para o servidor. Devido usar o UDP na sua estrutura e ele não ser confiável o CoAP adota dois tipos de mecanismos de qualidade de serviço. A "Mensagem confirmada" que fornece confiabilidade ao implementar o mecanismo de retransmissão na própria camada de aplicativo e "Mensagem não confirmada", é só enviar e esquecer. Não há necessidade de enviar uma mensagem de confirmação. Isso significa que se algum pacote for perdido, ele não será re-transmitido [9].

MQTT

O *Message Queuing Telemetry Transport* (MQTT) foi desenvolvido inicialmente pela *International Business Machines Corporation* (IBM) em 1999 e implementado em uma variedade de setores [10]. Diferente do CoAP ele usa TCP como protocolo de camada de transporte.

Neste modelo, vários clientes que atuam como *publisher* e *subscriber* se conectam ao corretor central. Cada assinante assina um tópico registrado *broker*. Cada *publisher* publica em um tópico específico e o *broker* transmite essa mensagem para todos os clientes assinados. Os clientes podem enviar dois tipos de mensagem: *PUBLISH* ou *SUBSCRIBE* [9]. O MQTT funciona sobre TCP e tem 3 (três) níveis de qualidade de serviço (QoS): QoS-0 (MQTT0), a mensagem é enviada apenas uma vez sem confirmação; QoS-1 (MQTT1), a mensagem é enviada e sua entrega é garantida (talvez com duplicatas); e QoS-2 (MQTT2), apenas uma mensagem é garantida para ser entregue aos assinantes sem duplicatas [11].

HTTP

O HTTP é um protocolo simples, do tipo solicitação-resposta, que roda sobre TCP. Ele especifica quais mensagens os clientes podem enviar para os servidores e quais respostas recebem de volta. Os cabeçalhos de solicitação e resposta são dados em *American Standard Code for Information Interchange* (ASCII), assim como no *Simple Mail Transfer Protocol* (SMTP) [4].

O HTTP é definido no [RFC 1945] e no [RFC 2616]. Cabe destacar que o HTTP é executado em dois programas: um cliente e outro servidor. Os dois, executados em sistemas finais diferentes, conversam entre si por meio da troca de mensagens HTTP. Tal protocolo define a estrutura dessas mensagens e o modo como o cliente e o servidor as trocam. Para aprofundar em HTTP, devemos revisar a terminologia da *world wide web* (Web) [3].

XMPP

O *Extensible Messaging and Presence Protocol* (XMPP) padronizado pela IETF 3 é composto por um conjunto de tecnologias abertas utilizadas principalmente para chat, mensagens instantâneas, vídeo e chamadas de voz [12]. No entanto, o XMPP não foi projetado para trocas de mensagens de alto desempenho dentro do mesmo modo e é mais apropriado quando usado para comunicação entre nós ou com aplicativos baseados na Internet [13].

REST

O *Service Representational State Transfer* (REST) é uma arquitetura que usa HTTP como protocolo de camada de aplicação, e fornece vários serviços da web para troca de dados e comunicação. Assim como o Coap, ele usa métodos do HTTP como GET, PUT, POST, DELETE etc. GET e POST que são usados para recuperar e enviar dados para o servidor [9].

WebSocket

O protocolo *WebSocket* atua na cama de transporte e foi desenvolvido para atender a dados constantes troca entre cliente e servidor não suportada anteriormente a partir de canal de HTTP O protocolo consiste em uma comunicação bidirecional completa e comunicação que funciona através de um único soquete, com protocolo HTTP [14]. Além disso, ele opera sobre TCP como uma atualização para uma conexão HTTP padrão, permitindo comunicação full-duplex de baixa latência entre um servidor e um cliente [10].

IV. RESULTADOS

Nos tópicos abaixo, os 8 (oitos) artigos citados na tabela 2 são analisados para julgar os protocolos, descrevendo os pontos positivos e negativos conforme os critérios largura de banda, segurança, tempo e consumo de energia.

1.Largura de banda

A largura de banda é uma propriedade física do meio de transmissão, que depende, por exemplo, da construção, da espessura e do comprimento do meio, seja ele fio ou fibra [4]. A largura de banda também é transmitida por uma rede Wi-Fi ou 4G, que será o foco dos estudos apresentados.

O protocolo escolhido de forma não apropriada pode sobrecarregar facilmente a rede existente e a infraestrutura de computação. A exemplo desses protocolos para cargas úteis pequenas de até 10 kB, temos o CoAP que, segundo testes de [9], nos quais foi utilizado o software *Wireshark,* é mais rápido em comparação aos protocolos MQTT e REST, haja vista que fragmenta a carga útil em partes iguais.

É notório que tal fragmentação é boa para cargas úteis pequenas, pois nenhum pacote transmitido e recebido é menor, porém, à medida que o tamanho da carga útil aumenta, o CoAP perde sua vantagem e o REST se mostra melhor para essas cargas úteis como mostra a figura 3 [9].

Já nos testes realizados por [10], no projeto de um aplicativo para estacionamento inteligente, se optou por uma simulação executada em três threads de cliente, quais sejam PUT, DELETE e GET, e em máquina, com sistema Ubuntu

14.04 de 64 bits com *Central Process Unit* (CPU) Intel Core i5-5200U a 2,20 *GigaHertz* (GHz) com 4 (quatro) núcleos. Como resultado, o atraso de propagação é zero. Com os testes finalizados pode-se observar que MQTT e CoAP possuem um tempo de resposta muito maior, uma vez que utilizam uma fila de mensagens para processar solicitações de clientes, porém o CoAP tem melhor desempenho, pois à medida que foi aumentada a utilização do servidor, o tempo médio de resposta dos protocolos aumenta.

Exceto o MQTT com *WebSocket* na camada de transporte, que segue uma tendência oposta. Isso ocorre por ele multiplexar (técnica que consiste na combinação de dois ou mais canais de informação por apenas um meio de transmissão usando um dispositivo chamado multiplexador) vários fluxos na mesma conexão devido às taxas de chegada mais altas, resultando em menos terminações de conexão[14]. Na Fig. 3 [9, Fig 5] são apresentados 5 (cinco) cenários diferentes como o MQTT2 que deve cuidar da entrega da mensagem sem duplicação da mensagem. os outros protocolos estão divididos em uma rede Wi-Fi e 4G.

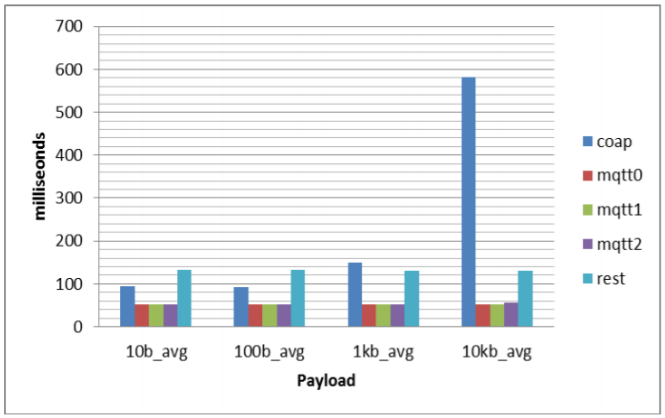

Fig. 3 Tempo de upload em cargas até 10 kb.

2. Segurança

Como a camada de aplicação é a "porta" para fora da rede, inúmeras ameaças podem surgir. Ela garante a autenticação do usuário e acesso a dados pessoais e sigilosos; portanto, o protocolo deve fornecer mecanismos para evitar que invasores e usuários mal-intencionados obtenham acesso ao sistema. Dito isso o autor [15] fala do fato do MQTT não possuir mecanismos de segurança impostos, já que foi projetado para operar em redes seguras,sendo uma má ideia criar uma rede MQTT globalmente, pois conforme o tamanho da árvore de tópicos aumenta, a complexidade aumenta. Outro sim é que o MQTT depende da criptografia *Secure Sockets Layer/Transport Layer Security* (SSL/TLS), caso contrário, o nome de usuário e a senha seriam enviados em texto não criptografado.

Embora seja um mecanismo muito confiável, a desvantagem desse mecanismo é que SSL/TLS é um protocolo caro para uso em dispositivos restritos. Hoje em dia, alguns brokers aceitam clientes anônimos e, portanto, o nome de usuário e senha não são mais necessários, mas não é desejável na maioria dos casos[15],[16],[19].

Segundo [2],[11] os protocolos baseados em UDP, como CoAP, fornecem transações menos confiáveis porque a perda ou falha de um pacote não é importante para o transmissor. Para superar esse desafio,uma abordagem de duas camadas em sua arquitetura é explorar a: 1) Camada de transação; e, 2) Camada de solicitação/resposta.

Já no protocolo XMPP, a IETF propõe o suporte nativo para autenticação via *Simple Authentication and Security Layer* (SASL) e segurança de transporte com TLS. A vantagem do SASL é que ele fornece um conjunto de métodos de autenticação a partir dos quais o cliente pode escolher aquele que melhor se adapta às necessidades específicas; no entanto, a desvantagem é que um mecanismo fraco ainda pode ser escolhido[15].

3. Tempo

Segundo [10] os protocolos MQTT e CoAP possuem um tempo de resposta muito maior que o XMPP e MQTT com *WebSockets*, uma vez que utilizam uma fila de mensagens para processar solicitações de clientes. O CoAP tira proveito do uso de UDP para comunicação e, portanto, tem um desempenho melhor com menor utilização, já o MQTT tem melhor desempenho em utilizações de servidor superiores, utilizando alguns recursos extras de otimização do protocolo.

Em uma comparação feita entre XMPP e *Websocket* pode-se observar que o XMPP tem melhor desempenho com menor utilização do servidor, pois transfere mensagens diretamente por meio de uma conexão TCP enquanto o MQTT com *WebSocket* usa um *handshake WebSocket* extra. Conforme a carga aumenta, o MQTT com *WebSocket* tira proveito da multiplexação e, portanto, o tempo médio de resposta diminui.

4. Consumo de energia

Embora o CoAP tenha cargas úteis pequenas, ao usar UDP como protocolo de transporte, o consumo de energia e a vida útil de uma bateria, por exemplo, aumenta com a diminuição do tamanho do cabeçalho do pacote em *Wireless Sensor Networks* (RSSF)[18]. Já o MQTT foi adotado pelo Facebook pois requer baixo consumo de energia, o que evita o esgotamento das baterias do smartphone[16].

Feito essas análises e considerando os critérios *largura de banda*, *segurança tempo* e *consumo de energia*, o protocolo que mais se destacou foi o **MQTT com *WebSocket***, já que tem a maior garantia de entrega dos dados.

Como o MQTT perde qualidade de tempo à medida que o servidor aumenta, o *Websocket* segue uma tendência oposta, pois ele multiplexa vários fluxos na mesma conexão devido às taxas de chegada mais altas. Então, com maior utilização do servidor, o MQTT com *WebSocket* supera os outros três protocolos, além disso fornece escalabilidade horizontal, enquanto esse recurso está ausente no CoAP e no MQTT.

V. CONSIDERAÇÕES FINAIS

O objetivo da presente revisão sistemática de literatura foi identificar comparar os protocolos da camada de aplicação, CoAP, MQTT, HTTP, XMPP e MQTT com

Websocket, analisando seus aspectos de largura de banda, segurança e tempo. De modo geral, pode se **evidenciar** que na literatura analisada, o que mais se destacou foi o MQTT com *Websocket*, sendo assim o mais adequado para o exemplo de solução da internet das coisas "dispositivo medidor de gás de cozinha".

Os resultados aqui apresentados são **importantes** por fornecerem uma visão ampla do potencial de cada protocolo e sua respectiva função, já que cada protocolo tem sua forma e objetivo de ser usado.

Todavia e considerando que o estudo tem a **limitação** de usar apenas uma base de dados, sugere-se que novos estudos sejam realizados para que se possa identificar de forma mais precisa os resultados fazendo buscas em outras bases de dados.

Referências

[1] B. Chen *et al*., "Edge Computing in IoT-Based Manufacturing," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 103–109, 2018.

[2] B. Safaei *et al*, "Reliability side-effects in Internet of Things application layer protocols," *2017 2nd International Conference on System Reliability and Safety (ICSRS)*, 2017.

[3] J.F. Kurose and K.W. Ross, *Redes de computadores e a internet: uma abordagem top-down*. Pearson, 2013.

[4] A.S. Tanenbaum and D.J. Wetherall, *Redes de computadores*. Pearson Prentice Hall, 2011.

[5] P. Cronin, F. Ryan, and M. Coughlan, "Undertaking a literature review: a step-by-step approach," British Journal of Nursing, vol. 17, no. 1, pp. 38–43, 2008.

[6] E.M. Lakatos and M. de A. Marconi, *Fundamentos de metodologia científica*. Editora Atlas S.A., 2010.

[7] J. Rowley and F. Slack, "Conducting a literature review," *Management Research News*, vol. 27, no. 6, pp. 31–39, 2004.

[8] F.M. Moreira, F.D.A. Rodrigues, and R.C.G. Sant'Ana, "Análise de Domínio da Produção Científica Sobre Coleta de Dados No Contexto Do Institute Of Electrical And Electronics Engineers," *Complexitas – Revista de Filosofia Temática*, vol. 3, no. 1, p. 28, 2019.

[9] U. Tandale, B. Momin, and D.P. Seetharam, "An empirical study of application layer protocols for IoT," *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, 2017.

[10] P. Kayal and H. Perros, "A comparison of IoT application layer protocols through a smart parking implementation," *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, 2017.

[11] T. Moraes *et al.*, "Performance Comparison of IoT Communication Protocols," *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019.

[12] S.N. Swamy, D. Jadhav, and N. Kulkarni, "Security threats in the application layer in IOT applications," *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2017.

[13] L. Sikic *et al*, "A Comparison of Application Layer Communication Protocols in IoT-enabled Smart Grid," *2020 International Symposium ELMAR*, 2020.

[14] G.M.B. Oliveira, *et al*, "Comparison Between MQTT and WebSocket Protocols for IoT Applications Using ESP8266," *2018 Workshop on Metrology for Industry 4.0 and IoT*, 2018.

[15] Kakakhel, Syed Rameez Ullah, et al. "A Qualitative Comparison Model for Application Layer IoT Protocols." 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC), 2019, doi:10.1109/fmec.2019.8795324.

[16] Security in the Internet of Things: A Survey on Application Layer Protocols - Lavinia Nastase2017 21st International Conference on Control Systems and Computer Science (CSCS) - 2017

[17] K. P. Naik and U. R. Joshi, "Performance analysis of constrained application protocol using Cooja simulator in Contiki OS," 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), 2017.

[18] U. Tandale, B. Momin, and D. P. Seetharam, "An empirical study of application layer protocols for IoT," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), 2017.