

Agregação e Desagregação de Dados IoT em Redes Definidas por Software Utilizando P4

Humberto C. C. Leão¹, André Riker¹, Antônio J. G. Abelém¹

¹Grupo de Pesquisa em Redes de Computadores e Comunicação Multimídia (GERCOM)
Universidade Federal do Pará (UFPA) – Pará – Brasil

{humbertoleao, afr, abelem}@ufpa.br

Abstract. *Low-Power Internet-of-Things (LPIoT) networks are characterized by a large number of IoT nodes with resource limitations. Due to its limitations, it is common to use data aggregation techniques in LPIoT data traffic. However, any aggregated payload needs to be disaggregated before the data is delivered to IoT applications. This work proposes a new strategy for aggregating and disaggregating IoT data in software-defined networks (SDN) using the P4 language.*

Resumo. *Redes Low-Power Internet-of-Things (LPIoT) são caracterizadas por volumosa quantidade de nós IoT com limitações de recursos. Por conta de suas limitações, é comum a utilização de técnicas de agregação de dados no tráfego de dados LPIoT. No entanto, toda carga útil agregada precisa ser desagregada antes que os dados sejam entregues às aplicações IoT. Este trabalho propõe uma nova estratégia de agregação e desagregação de dados IoT em redes definidas por software (SDN) utilizando a linguagem P4.*

1. Introdução

Low-Power Internet-of-Things (LPIoT), ou Internet das Coisas de Baixa Potência, em tradução livre, trata-se de uma rede IoT composta por nós com limitações em seus recursos, como energia, processamento e transmissão. Normalmente essas redes são projetadas para aplicações como cidades inteligentes, monitoramento de edifícios e medição inteligente de energia, para citar alguns.

O *Constrained Application Protocol (CoAP)* é um protocolo criado pela *Internet Engineering Task Force (IETF)*, especialmente para ser utilizado em redes IoT com limitações nos nós, ou na própria rede.

Devido ao grande tráfego de dados em redes LPIoT, estas geralmente aplicam agregação de dados, um mecanismo com o objetivo de coletar vários pacotes de dados pequenos e agrupá-los, formando um único pacote maior, a fim de reduzir a quantidade de dados trafegados na rede, com o objetivo principal de minimizar o gargalo de tráfego, aumentar a vida útil da rede, entre outros benefícios.

Dados agregados necessitam ser desagregados antes que sejam entregues às suas respectivas aplicações. Esse processo tem alto custo em termos de processamento, por isso a eficiência dos processos de agregação e desagregação depende de alguns fatores como projeto da rede, por exemplo.

Em Redes Definidas por Software (SDN) o plano de controle é fisicamente separado do plano de dados. Essa separação permitiu maior programabilidade aos dispositivos de encaminhamento [Bosshart et al. 2014]. O P4 trata-se de uma linguagem de programação que faz parte da tecnologia SDN, para proporcionar maior independência de protocolos e dispositivos [Bosshart et al. 2014]. onde o programador define o conjunto de recursos da rede, escreve o código, que é então compilado e inserido no dispositivo de rede. Por isso o P4 fornece uma maneira de configurar os *pipelines* de processamento de pacotes para processar pacotes no plano de dados. [Wang et al. 2019]

Neste trabalho utilizamos a linguagem P4 para desenvolver programas que implementam lógicas de agregação e desagregação de dados. A agregação é utilizada para concatenar pacotes pequenos em um único pacote grande. Do outro lado há um gatilho que executa o programa de desagregação sempre que o *switch* recebe pacotes agregados.

2. Trabalhos Relacionados

Soluções de agregação de dados foram projetadas e amplamente aplicadas em redes IoT. A maioria dessas soluções permite a agregação de dados em redes IoT homogêneas de baixa potência, porém sem considerar como os dispositivos SDN podem agregar ou desagregar um tráfego proveniente de redes IoT.

Para permitir que a agregação de dados seja realizada não apenas pelos dispositivos IoT de baixa potência, mas também pelos dispositivos SDN, [Madureira et al. 2020] propõem uma solução onde dispositivos SDN realizam agregação no plano de dados do *switch* P4. [Kim et al. 2017] também propõem uma solução para permitir o processamento no plano de dados, onde a agregação de dados e esquemas de publicação-assinatura estão incluídos nas funcionalidades propostas. [Anbalagan et al. 2020] propõem uma solução de agregação de dados, onde a SDN inicia a agregação de tráfego entre LTE e um ponto de acesso (AP) WLAN ótimo. A seleção do AP ideal visa evitar religações frequentes e serviços privados. Essa abordagem aumenta o rendimento do Equipamento do Usuário e reduz a pressão de tráfego sobre o espectro licenciado.

Ambos os dispositivos IoT e SDN podem realizar agregação de dados para aumentar o desempenho da rede, mas o esquema de desagregação de dados é abordado por poucos trabalhos, sendo que todo o tráfego agregado por dispositivos SDN e IoT deve ser desagregado e entregue para vários destinos. A desagregação de dados representa um desafio porque requer bom poder de processamento, o que pode representar um gargalo para o desempenho de uma rede IoT SDN, portanto a taxa de desagregação nestes dispositivos é crucial para o desempenho da rede.

[Wang et al. 2019] apresentam uma solução de manipulação de cabeçalho capaz de transformar os switches P4 em agregador e desagregador de dados. Essa solução permite que o equipamento SDN agregue pequenos pacotes IoT em um grande, transmita-os pela rede e desagregue-os de volta aos pequenos pacotes originais. Uma das contribuições desta solução consiste no fato de que a agregação e desagregação de pacotes são realizadas puramente em um Circuito Integrado de Aplicação Específica, do inglês *Application-Specific Integrated Circuit* (ASIC). Este trabalho mostra que a agregação e a desagregação realizadas no plano de controle resultam em uma taxa de transferência inferior ao realizado no ASIC.

[Wang et al. 2020] abordam a taxa de desagregação de dados e propõem uma

solução para realizar a agregação e desagregação de dados nos switches P4 SDN de forma eficiente. Este trabalho apresenta dois métodos para desagregação de dados. O primeiro é um método baseado em reenvio e clone que pode desagregar um pacote agregado com N cargas úteis em N pacotes, reenviando o pacote agregado para o analisador N vezes. A cada vez que é reenviado, é clonado para extrair uma carga útil diferente. O segundo é um método baseado em *multicast*, que define metadados especiais para os pacotes agregados, para que o *switch* saiba que o pacote agregado deve ser processado como um pacote *multicast*. Durante o processamento *multicast*, é indicado o número de cópias necessárias para extrair todos os *payloads*, onde cada cópia possui um metadado para indicar qual *payload* deve ser extraído.

[Lin et al. 2018] estudam a falta de desempenho no processo de desagregação, mais especificamente focam no método de reenvio e no método baseado em clones. Nesse método, desagregar um pacote N requer passar pelo pipeline de entrada N vezes, o que diminui o processamento de pacotes no *switch*. Como a taxa de processamento é menor, os *switches* SDN são mais suscetíveis a perda de pacotes. Assim, os autores propõem um mecanismo de enfileiramento para reduzir o descarte de pacotes no processo de desagregação. Com base na teoria das filas, os *buffers* fornecidos na porta de entrada resultam em uma taxa de processamento de pacotes menor na desagregação do que seria necessário sem os *buffers*.

Uma lacuna encontrada nestes trabalhos é que há pouca flexibilidade do algoritmo para lidar com agregação e desagregação de pacotes com quantidade variável de *payloads*. Além do que geralmente utiliza-se a memória do *switch* para armazenamento dos pacotes agregados, o que limita a quantidade possível de *payloads* agregados em cada pacote.

3. Proposta

Este trabalho considera que uma rede possui *switches* SDN responsáveis por executar a agregação e desagregação de dados. A desagregação de dados pode causar redução na taxa de transferência, pois consome certo tempo computacional. Isso significa que, para cada *payload* dentro de um pacote agregado, é necessário criar e enviar uma mensagem contendo um novo cabeçalho e o *payload* desagregado. Esse processo pode causar perda de desempenho na rede, principalmente se a desagregação de uma grande quantidade de dados for realizada por um único *switch* SDN. Por essa razão propomos que dois ou mais *switches* SDN rodando programas P4 com nossa proposta para agregação e desagregação, recebam a carga distribuída uniformemente por um balanceador de carga, conforme ilustrado na (Figura 1), a fim de manter a maior vazão possível, evitar sobrecarga e maximizar o rendimento da rede.

Consideramos que os switches P4 serão conectados à rede IoT utilizando Ethernet, no entanto nossa especificação pode ser facilmente estendida para outros protocolos. A (Figura 2) apresenta a estrutura da mensagem utilizada no tráfego IoT. Como pode ser observado, uma única mensagem IoT pode agregar muitos *payloads*. Cada *payload* tem um campo de 2 bytes para identificar o tipo de dados e 2 bytes para os dados, em virtude de tratar-se de uma rede LPIoT onde os dados são pequenos, o que pode ser alterado no futuro caso seja necessário. Consideramos o CoAP como o protocolo de aplicação, tendo 7 bytes para o cabeçalho CoAP e introduzimos um campo de 1 byte para indicar o número de cargas úteis que cada mensagem CoAP carrega. Além disso, a pilha de protocolos é

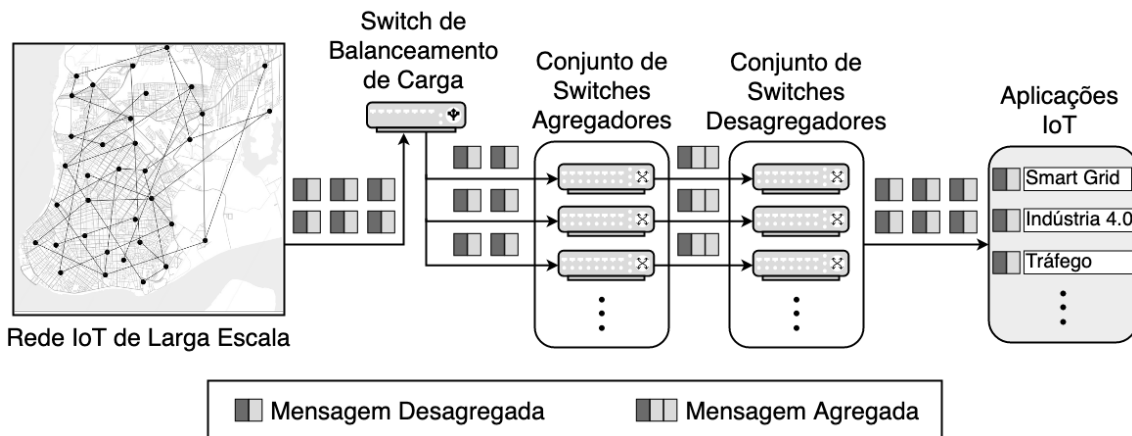


Figura 1. Esquema de Agregação e Desagregação Proposto

completada por UDP, IPv6 e IEEE 802.15.4 com 8, 7 e 33 bytes, respectivamente.

PHY MAC	IPv6	UDP	CoAP	Payload Length	Payload 1		...	Payload N		(bytes)
					Type	Data		Type	Data	
33	7	8	7	1	2	2		2	2	

Figura 2. Formato da Mensagem IoT

É importante notar que o formato de mensagem presente na (Figura 2) é válido para a comunicação interna na rede IoT, porém o tráfego IoT assume outros formatos quando o tráfego cruza o roteador de borda, que possui várias interfaces de rede e é responsável por manter a comunicação entre os dispositivos IoT e a rede externa. No escopo do projeto consideramos IPv4 e IPv6 como protocolos de rede e UDP para protocolos de transporte, uma vez que os pacotes CoAP são processados como datagramas no escopo deste projeto.

3.1. Agregação de Pacotes

A lógica de agregação dos *payloads* IoT mostrado na (Figura 3) implementa o modelo de armazenamento e encaminhamento em linguagem P4 que, ao receber um pacote IoT, extrai seu conteúdo, armazena seu *payload* na memória do *switch* usando um registrador e sinaliza ao *switch* que ainda não está agregando pacotes. Então verificamos se a capacidade máxima de *payloads* que podem ser agregados em um único pacote foi atingida, através de um parâmetro denominado `MAX_IOT_PAYLOAD`. Caso `MAX_IOT_PAYLOAD` não tenha sido atingido, armazenamos o *payload* do pacote recebido, atualizamos o contador de *payloads* salvos e marcamos o pacote para descarte. Ao se detectar que não há mais espaço para armazenamento de *payloads*, ou seja, quando `MAX_IOT_PAYLOAD` é atingido, inicia-se a construção do pacote agregado, usando a recirculação de pacotes para buscar cada *payload* armazenado na memória do *switch* e adicioná-lo ao pacote agregado. A recirculação de pacotes continua até que se busque todos os *payloads* armazenados. Então o *switch* monta um único pacote com todos eles, zera a contagem de *payloads* armazenados e envia o pacote agregado na rede.

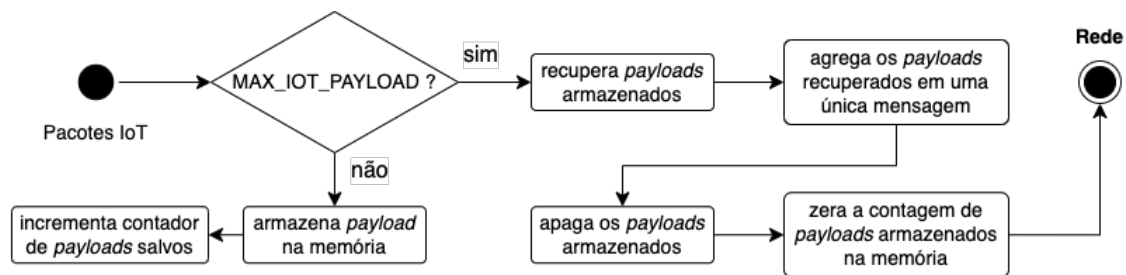


Figura 3. Lógica de Agregação de Pacotes

3.2. Desagregação de Pacotes

A (Figura 4) mostra a lógica de desagregação, que foi implementada em linguagem P4 e é disparada sempre que o *switch* recebe um pacote agregado. Quando isso ocorre, extraímos o primeiro *payload* agregado para uma estrutura de metadados e os *payloads* restantes são mantidos na estrutura do pacote. Então verificamos se ainda há mais de um *payload* no pacote restante. Se sim, clonamos este pacote, adicionamos o primeiro *payload* extraído de volta ao pacote clonado, apagamos os *payloads* restantes deste e o enviamos para a rede. Paralelamente a isso, no pacote original, agora com um *payload* a menos, já que o primeiro *payload* foi extraído para a estrutura de metadados, verificamos se ainda há mais de um *payload* para desagregar. Caso haja, recirculamos esse pacote e repetimos o processo até que permaneça apenas um *payload* no pacote original, então o enviamos para a rede.

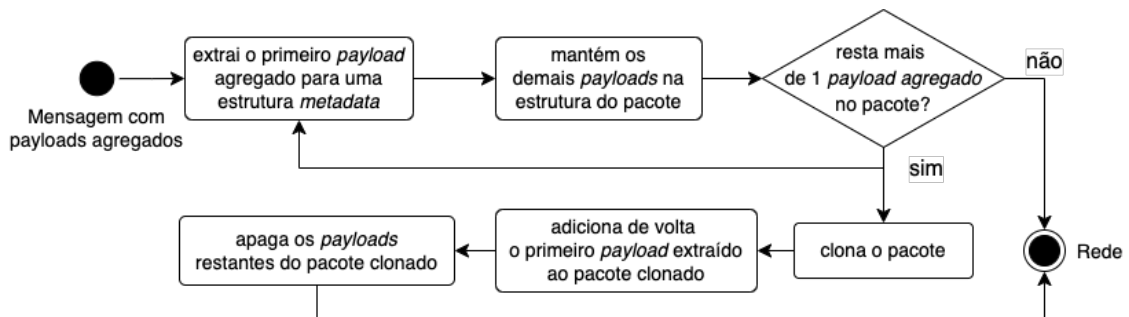


Figura 4. Lógica de Desagregação de Pacotes

A estratégia de desagregação proposta não envolve o armazenamento de pacotes agregados na memória do *switch*, por isso não há limite para o número de *payloads* agregados no pacote recebido para desagregação. Entretanto há uma relação entre flexibilidade na agregação de um grande número de *payloads* e eficiência no processo de desagregação. Como a desagregação envolve a recirculação de pacotes para cada *payload* desagregado e a análise de cada *payload* que está sendo desagregado, pode haver uma degradação de desempenho quando houver um número muito alto de *payloads* agregados.

Em nossa proposta o algoritmo pode ser facilmente modificado para lidar com agregação de uma quantidade variada de *payloads* apenas com ajuste de um parâmetro, diferentemente do que encontrado no estado da arte, onde esse número é pré-definido pelo código-fonte. Também é possível, por meio do modelo da rede, orquestrar a agregação

e desagregação, para direcionar o fluxo para os agregadores e desagregadores de modo a manter a maior vazão possível.

4. Considerações Finais

Agregação de dados é muito utilizada em redes IoT para reduzir a quantidade de dados trafegados na rede e, com isso, reduzir o gargalo de tráfego e aumentar a vazão. Neste trabalho apresentamos uma proposta de agregação e desagregação de dados implementada na linguagem P4 em *switches* SDN, como elementos de uma rede LPIoT com limitações de recursos. Propomos um algoritmo que permite flexibilidade na quantidade de *payloads* agregados, com a utilização de um parâmetro que determina o máximo de *payloads* agregados em um único pacote, sem a utilização da memória do *switch* para armazenar os dados agregados, o que remove a limitação do tamanho do pacote agregado. Apresentamos, ainda, uma proposta de balanceamento para distribuição uniforme de carga entre os *switches* agregadores e desagregadores, com o objetivo de aumentar a vazão de dados e evitar sobrecarga da rede.

Referências

- Anbalagan, S., Kumar, D., Raja, G., Ejaz, W., Bashir, A. K., et al. (2020). Sdn-assisted efficient lte-wifi aggregation in next generation iot networks. *Future Generation Computer Systems*, 107:898–908.
- Bosshart, P., Daly, D., Gibb, G., Izzard, M., McKeown, N., Rexford, J., Schlesinger, C., Talayco, D., Vahdat, A., Varghese, G., et al. (2014). P4: Programming protocol-independent packet processors. *ACM SIGCOMM Computer Communication Review*, 44(3):87–95.
- Kfoury, E. F., Crichigno, J., and Bou-Harb, E. (2021). An exhaustive survey on p4 programmable data plane switches: Taxonomy, applications, challenges, and future trends. *IEEE Access*, 9:87094–87155.
- Kim, K.-W., Min, S.-G., and Han, Y.-H. (2017). A programmable data plane to support in-network data processing in software-defined iot. In *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 855–860. IEEE.
- Lin, Y.-B., Wang, S.-Y., Huang, C.-C., and Wu, C.-M. (2018). The sdn approach for the aggregation/disaggregation of sensor data. *Sensors*, 18(7):2025.
- Madureira, A. L. R., Araújo, F. R. C., and Sampaio, L. N. (2020). On supporting iot data aggregation through programmable data planes. *Computer Networks*, 177:107330.
- Pourghebleh, B. and Navimipour, N. J. (2017). Data aggregation mechanisms in the internet of things: A systematic review of the literature and recommendations for future research. *Journal of Network and Computer Applications*, 97:23–34.
- Wang, S.-Y., Li, J.-Y., and Lin, Y.-B. (2020). Aggregating and disaggregating packets with various sizes of payload in p4 switches at 100 gbps line rate. *Journal of Network and Computer Applications*, 165:102676.
- Wang, S.-Y., Wu, C.-M., Lin, Y.-B., and Huang, C.-C. (2019). High-speed data-plane packet aggregation and disaggregation by p4 switches. *Journal of Network and Computer Applications*, 142:98–110.