

IoTP: On Supporting IoT Data Aggregation Through Programmable Data Planes

André L. R. Madureira¹, Leobino N. Sampaio¹(Orientador)

¹Programa de Pós-Graduação em Ciência da Computação (PGCOMP)
Departamento de Ciência da Computação – Universidade Federal da Bahia (UFBA)
Salvador – BA – Brasil

{andre.romano, leobino}@ufba.br

Abstract. *In this dissertation, we introduced the Internet of Things Protocol (IoTP). An L2 communication protocol for IoT programmable data planes that supports the implementation of data aggregation algorithms inside hardware switches, at the network level. Through these features, IoTP provides support for the design of efficient and adaptable aggregation schemes that can adjust their operation according to network status and based on the different communication technologies used by IoT devices. We implemented IoTP using the P4 language and conducted emulation-based experiments through Mininet environment. Our findings show that IoTP accomplishes a 78% improvement in network efficiency, as well as allowing control over the average delay generated by data aggregation techniques.*

1. Introduction

Network packet aggregation is a frequently used technique in the Internet of Things (IoT). Such an approach combines several packets with heterogeneous requirements and characteristics into a single packet transmitted to the next network node. By reducing the number of packets on the network, this aggregation technique, in general, also reduces the devices' energy consumption, increases the network's efficiency, and eliminates the packets recurring headers [Akyurek and Rosing 2018]. When performing aggregation, IoT devices that operate as sink nodes require more computational resources and present a higher energy consumption than other network nodes [Akyurek and Rosing 2018]. Therefore, hierarchical (i.e., tree topology) and clustering strategies carry out the aggregation through a set of interconnected IoT devices [Rahman et al. 2016, Shen et al. 2017, Wissingh et al. 2017], instead of solely sink nodes. Besides energy constraints, devices' computational resource limitations also restrict algorithm execution and aggregation efficiency.

Solutions like the Priority-based Dynamic Data Aggregation scheme (PDDA) face resource constraint-related problems by adopting multilayered hierarchical big data aggregation. They enable the deployment of different aggregation schemes according to the available computational resources (e.g., memory and data processing capability) of the devices in each layer of the network [Karim and Al-kahtani 2016]. Although adopting different techniques, these approaches, however, overlook the specifics of the communication technologies involved (e.g., BLE, 802.11, and LoRa). Conversely, other proposals implement aggregation mechanisms that adapt themselves to the communication link by coupling the solution to the supported network protocol. But, the degree of dependence on one specific network protocol makes them harder to apply in different networking

scenarios. Therefore, the literature reports some attempts to use generic architectures through which different communication technologies and heterogeneous IoT devices are integrated to facilitate both data transmission and aggregation [Shen et al. 2017].

Despite the contributions of the initiatives mentioned above, current data aggregation strategies can reap the benefits of data plane programmability through SDN/P4 language technologies that can provide a richer set of network information that had not been available so far. Based on the features of the communication link, algorithms can rely on optimization techniques to accomplish a higher data aggregation efficiency. Thus, this investigation’s main problem statement is **how to process network and device information to support high-efficiency data aggregation on IoT networks?**

The main contributions of this work are the following: i) A protocol that allows data aggregation solutions to be deployed directly in the data plane through programmable hardware-based switches; ii) A protocol that provides support for data aggregation algorithms that can use the network information provided by the proposed protocol to reduce network communication overhead while improving network’s efficiency¹; iii) A protocol that can prioritize IoT data retrieval by controlling data aggregation delays.

2. Internet of Things Protocol (IoTP)

The architecture proposed in this dissertation aims to mitigate congestion and improve the efficiency of IoT networks by performing aggregation of IoT data as it traverses the network switches. Such aggregation improves network efficiency as it reduces the overhead caused by the repeated packet headers of network protocols. To improve the interoperability and make the protocol more generic, we designed IoTP² to be independent of routing protocols and link technologies, by redefining the link layer processing (Layer 2). Due to these features, IoTP can coexist with any other IoT communication technologies, such as BLE and ZigBee, due to the programmable data plane low-level header processing.

2.1. Protocol Header and Packet Payload

To make L2 processing feasible, the protocol consists of a fixed header made up of five fields and a stack of data blocks, as described in Figure 1. The IoTP protocol header is formed by the following fields:

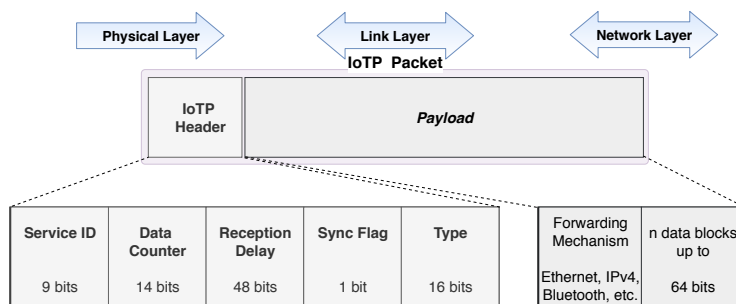


Figure 1. IoTP Packet Structure [Madureira et al. 2020]

¹Network efficiency indicates the communication overhead associated with a protocol, is defined as the ratio of a total payload by total data transmitted. Refer to Section 3 for further details.

²<https://github.com/insert-br/iotp/tree/main/exercises/iotp>

- **Service ID:** This 9-bit field identifies a set of similar data belonging to the same IoT application. Thus, network switches group incoming data into a single packet.
- **Data Counter:** IoTP enables IoT devices to send multiple blocks of data in a single packet, as long as the limit set by the network's MTU is respected. To this aim, this 14-bit field identifies the number of data blocks a packet carries.
- **Reception Delay:** This 48-bit field indicates the time taken from the IoT sensor data capture to the reception of this information by the IoTP gateway. Thus, the IoTP switch stores IoT data internally and, when the condition to send the packet is met, the IoTP switch updates the packet delay and sends the aggregated data.
- **Sync Flag:** When an IoTP packet with Sync Flag enabled is received by the switch, it sends all the internally stored data that is associated with the service ID of the Sync Flag packet to the IoTP gateway. By periodically sending packets with Sync Flag, the gateway is able to control the trade-off between the data retrieval delay and the efficiency of the data aggregation strategies.
- **Type:** This 16-bit field informs which forwarding mechanism should be used to forward IoTP packets. Therefore, IoTP can be used on any network, regardless of the underlying communication and routing technologies. For instance, the forwarding mechanism can be another protocol header such as BLE, 5G, and IP.

Each IoTP packet contains a finite set of up to n blocks of data as payload. Each block carries the data collected by the IoT devices, and can contain up to 64 bits of information. Thus, all elements that compose an IoTP network (such as switches, gateways and IoT devices) are aware of the amount and format of the data carried by each packet, which facilitates the use of aggregation strategies.

2.2. Protocol Operation

To assess IoTP effectiveness, we implemented a derived version of the Accretion aggregation algorithm [Kim et al. 2006]. In our implementation, IoTP nodes receive various packets from other network devices and aggregate their data according to the service ID of the packets. The algorithm has two steps: i) data storage, and ii) data transmission.

First, the aggregation strategy stores the data sent by the IoT devices, as well as the delays that the data suffered from the moment it was captured to the time it has reached the IoTP switch (see Algorithm 1). In addition to that, the IoTP switch stores delays caused by the aggregation strategy that executes alongside the IoTP protocol, within the IoTP switch. After the storage phase ends, the conditions for sending the data are verified. The switch calculates the cumulative data aggregation delay, adds this information to the delays reported by the IoT devices, and sends the stored data (see Algorithm 2).

The aggregation strategy implemented within the IoTP's switches considers two conditions for sending the stored data: (i) minimum amount of data blocks reached or (ii) an IoTP packet with Sync Flag enabled is received by the switch.

In the first condition, each service ID of the IoTP protocol is associated with a minimum amount of data, called "Trigger Counter". When this minimum amount of storage is reached, the switch applies the aggregation strategy to the locally stored data and sends the aggregated data to the IoTP gateway. The minimum number of data blocks for each service ID is defined by the SDN controller, according to the characteristics of the received data and the IoT applications requirements.

Algorithm 1: Data Storage.

```
1 Function AggregateReceivedData (pkt) :  
2   if pkt.dataCounter > 0 then  
3     STOREDELAY(pkt.serviceID, pkt.delay)  
4     foreach data in pkt.dataList do  
5       | STOREDATA(pkt.serviceID, data)  
6     end  
7   end  
8 end
```

Algorithm 2: Aggregation and Data Send.

```
1 Function SendAggregatedData (pkt) :  
2   id = pkt.id  
3   sf = pkt.syncFlag  
4   len = LENSTOREDDATA(id)  
5   if len >= MINDATA(id) or sf == 1 then  
6     pkt.delay = STOREDDELAYS(id)  
7     pkt.delay += TIMEDATAINSWITCH(id)  
8     pkt.dataCounter = len  
9     foreach data in STOREDDATA(id) do  
10    | pkt.dataList.push(data)  
11    end  
12    SENDPACKET(pkt)  
13  end  
14 end
```

The second condition enables the IoTP to control delays of the aggregation strategy. However, unlike the End-to-End and Accretion algorithms [Kim et al. 2006], IoTP does not use timers to perform such control. Instead, the delays are controlled by sending packets with Sync Flag enabled to the IoTP switch. As such packets are received, the switch retrieves all stored data associated with the service ID. Then, the aggregation strategy is applied over the data and the resulting data aggregate is sent to the IoTP gateway. So, the IoTP gateway can control the frequency of data reception by sending IoTP packets with Sync Flag periodically (see Algorithm 3).

3. IoTP Evaluation

In order to assess IoTP's effectiveness, we conducted experiments comparing End-to-End (E2E) and IoTP-based³ aggregation strategies. The experiments were conducted in the Mininet/BMv2 emulation environment using a trace⁴ created from the Intel Berkeley

³For the sake of simplification, this section uses the term IoTP to refer to the aggregation algorithm implemented in the IoTP switch.

⁴The trace contains data from 54 real IoT devices that measure room temperature, humidity, light and battery voltages for each device. Each Mininet host emulated an IoT device of the trace. Refer to the full dissertation manuscript for further details.

Algorithm 3: SyncFlag Packet Sending

```
1 Function SendSyncFlag (pkt) :  
2   for id in serviceIDList do  
3     receptionTime = TIMELASTRECEPTION(id)  
4     if ACTUALTIME - receptionTime >= RECEPTIONTIMEOUT(id)  
5       then  
6         pkt.id = id  
7         pkt.delay = 0  
8         pkt.dataCounter = 0  
9         pkt.syncFlag = 1  
9         SENDPACKET(pkt)
```

Research Lab dataset⁵ [Madden et al. 2004]. In the emulation environment, we connect a set of IoT devices and an IoTP gateway to the same switch. We connected the IoT devices to the switch via Bluetooth Low Energy (BLE) wireless link technology (1 Mbps, 6ms delay), whereas the IoTP gateway was connected via IEEE 802.3u (100 Mbps, 0ms delay). All remaining emulation parameters were adopted from the Mininet’s defaults.

3.1. Evaluation Methodology

We organized the performance evaluation of the IoTP in three steps. First, we carried out a conformity test to establish a baseline that allows us to distinguish the emulation environment performance from the evaluated aggregation mechanisms and protocol stacks. Then, we applied the full factorial design with k factors (2^k design) [Jain 1990]. Finally, we considered the most effective factors for our performance analysis, as described in Table 1. The factors were chosen according to the state-of-the-art literature recommendations. Only the minimum and maximum levels were taken into account for the factorial design phase. For statistical purposes, we replicated the experiment 20 times, extracted the average, the standard deviation and calculated the 95% confidence level.

Table 1. Full Factorial Design.

	Factors	Min - Max	Unit
<i>A</i>	Data Aggregation	10 - 50	data blocks
<i>B</i>	Number of IoT Devices	10 - 50	devices
<i>C</i>	IoTP Sync Flag Send Interval	0 - 0.1	seconds

We noticed that all factors (*A*, *B* and *C*) have a big impact over the Average Delay and that they interact with each. Thus, IoTP’s performance analysis considered the combined use of the three factors, and the following performance metrics were used to evaluate the proposed solution:

- **Network efficiency (%)**: Ratio between the total payload and the total data sent, which serves as an indicator of efficiency in data transmission on the network, as described by [Zechinelli-Martini et al. 2011] ;

⁵<http://db.csail.mit.edu/labdata/labdata.html>.

- **Total rate of packets sent by the switch (pps):** Total number of packets sent by the switch to the gateway divided by the duration of each emulation run;
- **Average delay (ms):** Time interval between the capture of the data by the IoT sensor and its reception by the gateway.

3.2. Comparative Analysis

In our experiments, we evaluated the effects of two IoTP strategies that interfere with data aggregation: (i) periodically sending IoTP packets with Sync Flag (SF) enabled (i.e., SF IoTP strategy), and (ii) not sending such SF packets (i.e., Non-SF IoTP strategy). When comparing SF IoTP with Non-SF IoTP, we noted that there is an increase in the number of packets received by the IoTP gateway when SF IoTP packets are sent, as depicted in Figure 2. The SF IoTP strategy works by sending SF IoTP packets periodically to inform the switch that we need to obtain data from IoT devices right away to satisfy a particular deadline. Therefore, the IoTP switch attempts to meet this demand by sending a larger number of packets with a small payload. That is, in SF IoTP, we are prioritizing the deadline over the efficiency of the aggregation strategy implemented within the IoTP switch. In Non-SF IoTP, we prioritize the aggregation efficiency over the deadline.

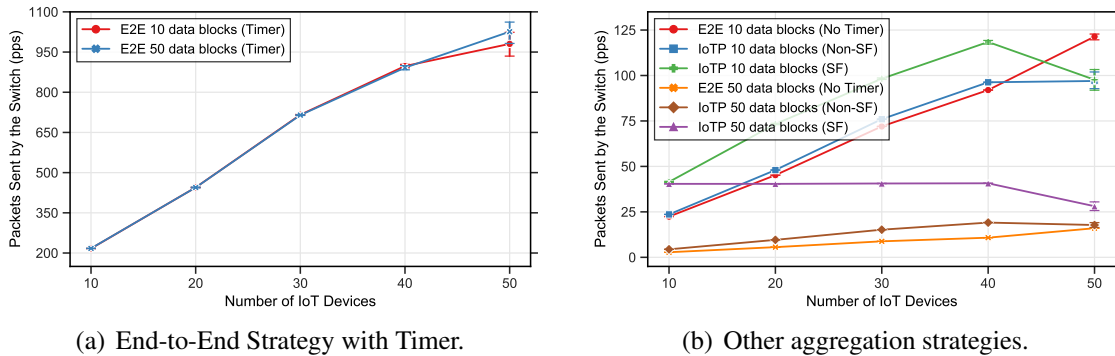
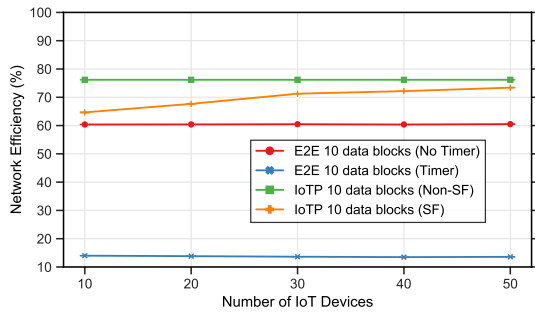


Figure 2. Packets sent by the switch [Madureira et al. 2020]

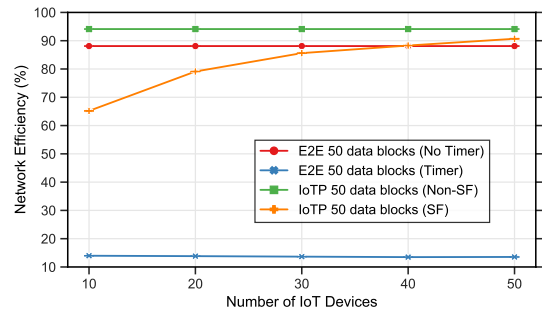
Statistically, E2E and IoTP presented a similar reduction in the total number of packets sent over the network when compared to no data aggregation scenarios⁶, with E2E presenting a slight advantage over IoTP. As a result of such reduction, the network efficiency is expected to increase [Akyurek and Rosing 2018]. It can be noted that, the network efficiency was improved by using IoTP, since the protocol achieved a higher network efficiency than E2E. We also remark that the IoTP's network efficiency decreases when it is periodically sending Sync Flag packets (i.e., SF IoTP), as depicted in Figure 3. Due to the use of Timers in E2E and Sync Flags in SF IoTP, the average aggregation delay gets a higher priority than data aggregation efficiency.

In respect to the average delay, IoTP presented lower values when aggregating up to 10 data blocks in comparison with the E2E strategy, as depicted in Figure 4. However, when IoTP operates with 50 data blocks, the E2E could achieve the lowest average delay. We believe that this difference results from the fact that, unlike E2E, IoTP does not operate with timers. IoTP is a reactive protocol that controls the average delay by receiving Sync Flag packets (i.e., SF IoTP). Thus, control of the average IoTP delays depends on the

⁶Network performance of the no data aggregation scenarios are further detailed in the dissertation.



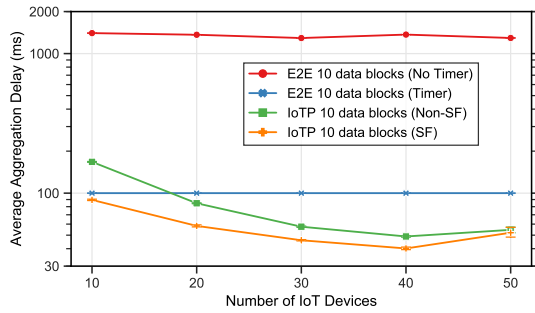
(a) Up to 10 aggregated data blocks.



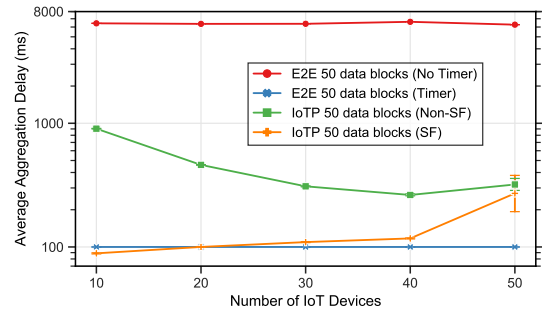
(b) Up to 50 aggregated data blocks.

Figure 3. Network efficiency [Madureira et al. 2020]

timing of sending Sync Flag packets and network characteristics such as propagation delay and link transmission capacity.



(a) Up to 10 aggregated data blocks.



(b) Up to 50 aggregated data blocks.

Figure 4. Average Delay [Madureira et al. 2020]

4. Conclusion

In this dissertation, we present IoTP, a low overhead protocol for IoT sensor networks that can improve network efficiency, control the average delay induced by data aggregation and reduce the overhead that repeated protocol headers cause on the network. Our results show that IoTP has a 78% better network efficiency and is 5 times faster in terms of average delay when compared to the End-to-End data aggregation strategy.

5. Publications

The following scientific publications were a direct result of this research's efforts:

1. Madureira, André Luiz R.; Araújo, Francisco Renato C.; Sampaio, Leobino N. On supporting IoT data aggregation through programmable data planes. *Computer Networks*, v. 177, p. 107330, 2020. doi: 10.1016/j.comnet.2020.107330 (A1)
2. Madureira, A. L. R.; Araújo, Francisco Renato C.; Sampaio, L. N. Um Protocolo IoT para Redução de Tráfego em Redes de Plano de Dados Programáveis. In: *Anais do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, 2020, Rio de Janeiro. SBC, 2020. v. 38. p. 826-839. doi: 10.5753/sbrc.2020.12328 (A4)

3. Madureira, A. L. R.; Araújo, Francisco Renato C.; Prates, L. N. B.; Sampaio, Leobino. NDN-ADAP: Uma Arquitetura para Encaminhamento Eficiente de Pacotes em Redes de Dados Nomeados. In: Anais do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 2020, Rio de Janeiro. v. 38. p. 812-825. doi: 10.5753/sbrc.2020.12327 (A4)
4. Madureira, Andre Luiz R.; Araujo, Francisco Renato C.; Araujo, Guilherme B.; Sampaio, Leobino Nascimento. NDN Fabric: Where the Software-Defined Networking Meets the Content-Centric Model. *IEEE Transactions on Network and Service Management*, v. 18, p. 374-387, 2021. doi: 10.1109/TNSM.2020.3044038 (A1)

References

- Akyurek, A. S. and Rosing, T. S. (2018). Optimal packet aggregation scheduling in wireless networks. *IEEE Transactions on Mobile Computing*, 17(12):2835–2852.
- Jain, R. (1990). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. John Wiley & Sons.
- Karim, L. and Al-kahtani, M. S. (2016). Sensor data aggregation in a multi-layer big data framework. In *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 1–7.
- Kim, K., Ganguly, S., Izmailov, R., and Hong, S. (2006). On packet aggregation mechanisms for improving voip quality in mesh networks. In *2006 IEEE 63rd Vehicular Technology Conference*, volume 2, pages 891–895.
- Madden, S., Bodik, P., Hong, W., Guestrin, C., Paskin, M., and Thibaux, R. (2004). Intel Lab Data. Disponível em: <http://db.csail.mit.edu/labdata/labdata.html>. Último acesso em: 18 de maio de 2019.
- 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.
- Rahman, H., Ahmed, N., and Hussain, I. (2016). Comparison of data aggregation techniques in internet of things (iot). In *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pages 1296–1300.
- Shen, Y., Zhang, T., Wang, Y., Wang, H., and Jiang, X. (2017). Microthings: A generic iot architecture for flexible data aggregation and scalable service cooperation. *IEEE Communications Magazine*, 55(9):86–93.
- Wissingh, B., D’Acunto, L., and Trichias, K. (2017). In-network data aggregation in icn: Demo paper. In *2017 8th International Conference on the Network of the Future (NOF)*, pages 129–131.
- Zechinelli-Martini, J. L., Buccioli, P., and Vargas-Solar, G. (2011). Energy aware data aggregation in wireless sensor networks. In *2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (Wireless VITAE)*, pages 1–5.