A Survey of Low-Latency IoT System Using FPGA Accelerator

F P Mahimai Don Bosco¹*, E Chitra², and S Ryan Ebenezer¹

¹Department of Electronics and Communication Engineering, Loyola-ICAM College of Engineering and Technology, Chennai, Tamil Nadu, India
²Department of Electronics and Communication Engineering, Kattankulathur Campus, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India
Email: *mahimaidonbosco.fp@licet.ac.in

Abstract. Internet of Things has taken its place in the world of technology fairly in the previous few years. It is assumed that there will be approximately 4 billion IoT devices interconnected by the year 2030. IoT has not widespread full feathered in all the fields of application. However, the future holds a wide spectrum of implementations and dependency in IoT, which demands digital computing parameters such as faster processing of data, reduced latency and parallel processing of multiple data channel simultaneously. This publication provides a solution to satisfy these parameters, using FPGA (field-programmable gate array) accelerators in the IoT systems.

In IoT, it is necessary to achieve data-centric parameters such as higher bitrate at a seamless flow rate avoiding data congestion and data traffic. The predictability of the endpoint is another important parameter to be considered in an IoT system. In this paper, we will discuss the use of Constrained Application Protocol and speculate the possibility of enhancing the performance parameters such as latency and predictability by accelerating the cloud servers with FPGA Accelerator.

Keywords: Field Programmable Gate array (FPGA), Constrained Application Protocol (CoAP), Hardware acceleration.

1. Introduction
The Internet of Things has fundamentally branched out from simple applications using menial sensors and processing units to extraordinarily complex applications such as automated locomotives. IoT systems have not been implemented commercially in a wide variety of fields [1]. Therefore, the number of data transfers in IoT systems is relatively lower than the potential [2]. Therefore, systems use low-frequency message transfer. Usage of lower frequency of message transfer between the endpoint devices comes with a low risk-probability of data traffic, congestion. Complicated IoT applications tend to produce a large amount of data transfer, resulting in Data congestion and Data traffic [3].

Hardware acceleration in IoT systems can be achieved by implementing [4] various methods; one such method is by accelerated processing of the Application protocol used in the system [5]. In this survey [6], we are using the CoAP protocol for reasons discussed as follows. CoAP (Constrained Application Protocol) is the protocol used in IoT systems [7]. Hardware acceleration through the CoAP application protocol can be implemented [8] by either accelerating the network or simply accelerating the processing by allocating the accelerator [9] to compliment the CPU in handling the overhead data to be processed, thus reducing the overload of the CPU; this method evidentially results in drastic [10]
2. **CoAP Accelerator**

The FPGA based CoAP accelerator improves the performance of the system by handling the Handshake protocol handled by the CPU [11]. The CoAP RTS and CTS semantics are passed in CoAP messages. Request and response information [12], such as the URI and consignment media type, are carried as CoAP options for further packet parsing and filtering [13]. In an IoT system, the request/response model [14] is highly dynamic and increases the latency between two packets [15]. This paper’s accelerator architecture design is so that the accelerator handles the request/response, thus reducing the CPU’s overhead. The handshake request/response model between client and server in a CoAP network is shown in Figure 1.

![CoAP Request/Response model](image)

**Figure1:** CoAP Request/Response model

2.1. **Constrained Application Protocol**

The Constrained Application Protocol (CoAP) is designed especially for solutions implemented in a constrained environment with constraints such as limited power resources, bandwidth, processing capabilities, etc. CoAP is a derivative of the HTTP protocol. CoAP is preferred over HTTP in constrained environments due to the processing constraint, such as the nodes used in IoT networks often have 8-bit microcontrollers with small amounts of ROM and RAM. CoAP is a lighter derivative version of HTTP that can satisfy the constraints. CoAP is developed for applications with constrained environments where power and bandwidth are limited. The CoAP also is inclusive of Methods such as GET, HEAD, POST, PUT, and DELETE and CONNECT, which are highly resourceful in applications such as the Internet of Things. The CoAP uses UDP as the transport layer protocol.

2.2. **Field Programmable Gate Array**

An FPGA is a reconfigurable programming unit. It is a derivative unit of its principle predecessors PLA and PAL, and an FPGA is advantageous in implementing algorithmic architectures for specific applications for the reasons listed below. FPGA, alike many other processing units, consists of digital units such as Flip-Flops, Registers, LUT, and RAM in a reconfigurable fashion. These resources are synthesized and dumped into an FPGA. FPGAs are extremely versatile and highly compatible with real-time applications. Therefore, FPGAs are preferred to other alternatives such as GPU.

FPGAs have limited storage capability; however, storage is not a constraint in the applications of IoT. 1. Parallel computing is the differentiating and pivotal feature of FPGA. The processor can accept multi-channel input data and compute in a parallel manner to parse the CoAP packets for further processing. 2. One of the significant features of FPGA is the Reconfigurable Hardware Architecture, its benefits the Internet of Things to be versatile as it should be without any physical hardware modification.
3. Low power consumption is the biggest challenge in designing an IoT system; FPGA helps IoT devices be energy-efficient yet faster than a conventional IoT system paradigm. The parallel processing capability of FPGA allows the IoT systems to open up to handling multiple sensors/actuators simultaneously. Applications such as Augmented Reality, Virtual reality and automated locomotives require a huge amount of data to be sent every fraction of a second. When processed with IoT CPU, such loads of data cause an enormous amount of latency that is not desirable for such critical applications of IoT. Even though FPGAs' clock speed is lower (60MHz), its ability is unlocked with the help of its multithreading and the throughput it offers. The data is processed and transmitted to a cloud computing server for further processing. This publication focuses specifically on FPGA-based IoT accelerators.

2.3. Advanced eXtensible Interface
Advanced extensible Interface, or AXI, is part of ARM's AMBA specifications. It acts as the system bus protocol for the communication between the different modules within the design architecture, thus implementing Inter-Process Communication between the Processing modules. The AXI is pivotal in an FPGA CoAP accelerator. The CoAP accelerator parses the CoAP data at high speeds within the Processing modules. Thus the AXI ensures the imminent interconnect and data parsing. The Accelerator Adapter IP core accessories accelerators developed using the Xilinx® Vivado High-Level Synthesis (HLS) tool.

2.4. General Design
The IoT accelerator's general design architecture consists of a different block to parse and process the CoAP data packets. Initially, the blocks can be divided into the coap_ingress and coap_egress. The coap_ingress consists of blocks responsible for parsing the CoAP header and store the URI. The coap_ingress handles the incoming request-to-send from the endpoint for every packet to be received by the server; it also consists of the General control configuration register. The coap_egress is responsible for compiling the information processed and push the data into the network, which is further transmitted to the respective endpoint. The general block design is shown in Figure 2 as follows.

![Figure 2: IoT Accelerator general design](image)

The udp_stack_ingress is the first step of filtering [4]. The received data packet is dissected into the header and the message. The parsed UDP and IP header is stacked down for filtering. The header is parsed to verify whether the received packet belongs to the respective destination, thus avoiding redundant data processing. The coap_header_parser extracts the header from the message; it parses information from the header such as Version (Ver), Type (T), Token Length (TKL), Code, Message ID. Figure 3 shows the CoAP header format.
Figure 3: CoAP Header Format

The header fields extracted by the `coap_header_parser` are fed into the `coap_message_processor`. The `coap_application-data` stores the URI of the endpoint. The CPU uses this URI for further information processing.

2.5. Hardware Acceleration

Hardware acceleration is the process in which the Accelerator co-processes the data alongside the CPU; in our case, the IoT accelerator itself is the supporting hardware implementation, which supports hardware acceleration which parses the CoAP header to complete the RTS, CTS interchange and thus accelerating the data processing. Hardware acceleration drastically reduces latencies due to its processing fashion, and the packet parsing and improves the performances of the system.

The accelerator we use in our survey is an FPGA that is extremely powerful in handling multithreaded server processing requests, thus accelerating higher standards and reducing delays.

3. Evaluation

The primary objective of the accelerator is to reduce delays and improve predictability. A simple IoT system is equipped with a CoAP blaster, which can vary the number of messages per second (mps). The test is run with and without an accelerator.

Figure 4: Latency Evaluation

The results were satisfying as the latency was reduced from an average of 160 μs to 40 μs. It was also observed that the accelerator varied its speed concerning the messages per second. Figure 4 shows a box-and-whisker graph of accelerator vs. server latency.
The brown boxes depict the latency of the server, and the purple boxes depict the Accelerator latency. It can be comprehended that the variation in the accelerator output is significantly lesser compared to the server performance; with this minimal variation in latency, the behavior of the endpoint device is easily evaluated. The Table 1 provides information about the IoT system's accelerated performance for each level of Messages per Second.

Table 1: Performance of IoT system with and without accelerator

| Rate (MPs) | Accelerator | Server | Speed-up |
|-----------|-------------|--------|----------|
| 2500      | 41.37       | 140.41 | 239.40%  |
| 5000      | 41.75       | 153.92 | 268.67%  |
| 7500      | 40.57       | 161.95 | 299.18%  |
| 10000     | 40.89       | 167.17 | 308.83%  |

4. Conclusion

From this, we can conclude that this accelerator model using CoAP processing has achieved both its objectives of Reduced Delay and Improved Predictability. We observed and analyzed the working and the improvements offered by an IoT Accelerator to an IoT system. Upon implementing industry standard IoT application, such Accelerators are required to be installed alongside to avoid data malfunction. It also lay the foundations for intensive cross-applications like IoT x AR etc.

References

[1] L. R. B. Brasilino and M. Swany, "Low-Latency CoAP Processing in FPGA for the Internet of Things," 2019 International Conference on the Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Atlanta, GA, USA, 2019, pp. 1057-1064, doi: 10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00182.
[2] AXI4-Stream Accelerator Adapter v2.1 (November 18, 2015), Vivado Design Suite, https://www.xilinx.com/support/documentation/ip_documentation/axis_accelerator_adapter/v21/pg081-axis-accelerator-adapter.pdf
[3] Shelby, Z., Hartke, K., Bormann, C. (June 2014) Internet Engineering Task Force (IETF), Standards Track, https://tools.ietf.org/html/rfc7252
[4] T. Gomes, F. Salgado, S. Pinto, J. Cabral and A. Tavares, "A 6LoWPAN Accelerator for the Internet of Things Endpoint Devices," in IEEE Internet of Things Journal, vol. 5, no. 1, pp. 371-377, Feb. 2018, doi: 10.1109/JIOT.2017.2785659.
[5] Brasilino, L.R. and Swany, M., 2019, July. Low-Latency CoAP Processing in FPGA for the Internet of Things. In 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) (pp. 1057-1064). IEEE.
[6] Abd-Elkader, Ahmed AH, Mostafa Rashdan, El-Sayed AM Hasanean, and Hesham FA Hamed. "Advanced implementation of Montgomery Modular Multiplier." Microelectronics Journal 106 (2020): 104927.
[7] Lombardi, Marco, Francesco Pascale, and Domenico Santaniello. "Internet of Things: A General Overview between Architectures, Protocols and Applications." Information 12, no. 2 (2021): 87.
[8] Premsankar, G. and Di Francesco, M., 2020. Advances in Cloud Computing, Wireless Communications and the Internet of Things. In Analytics for the Sharing Economy: Mathematics, Engineering and Business Perspectives (pp. 71-94). Springer, Cham.
[9] Mousavi, S.K., Ghaffari, A., Besharat, S. and Afshari, H., 2021. Security of the Internet of things based on cryptographic algorithms: a survey. Wireless Networks, 27(2), pp.1515-1555.

[10] Bouras, M.A., Farha, F. and Ning, H., 2020. Convergence of computing, communication, and caching in the Internet of things. Intelligent and Converged Networks, 1(1), pp.18-36.

[11] Safarini, O.A.S., Internet of Things: Standardizations, IoT Elements, Protocols, Architectural Design Choices, Challenges and Future Efforts.

[12] Rao, V. and Prema, KV, 2020. A review on lightweight cryptography for Internet-of-Things based applications. Journal of Ambient Intelligence and Humanized Computing, pp.1-23.

[13] Karad, S. and Thakur, R., 2021. Efficient monitoring and control of wind energy conversion systems using Internet of things (IoT): a comprehensive review. Environment, Development and Sustainability, pp.1-18.

[14] Schweissguth, E., Parzyjegla, H., Danielis, P., Mühl, G. and Timmermann, D., 2020. Real-time Publish/Subscribe for the Industrial Internet of Things.

[15] Paudel, N. and Neupane, R.C., 2021. General Architecture for a Real-Time Monitoring System Based on the Internet of Things. Internet of Things, p.100367.