DOI: 10.18721/JCSTCS.13201
УДК 004

DATA PROCESSING BY END DEVICES IN IoT SYSTEMS

A.Yu. Anufrienko

National Research University Higher School of Economics,
Moscow, Russian Federation

This paper presents a correlation method for processing data on end devices and reducing the amount of data transmitted over the network. Instead of expensive and complex network devices, developers can use cheap and proven low-speed Internet of Things (ZigBee, NB IoT, BLE) solutions for data transfer. The novelty lies in one of the features of this approach: the use of components for analysis, rather than a complete copy of the signals, as well as processing directly on the sensor. The advantage of this approach allows you to reduce the number of operations and complexity of implementation, in contrast to other methods focused on the cloud computing paradigm. We provide results for correlation values and the number of logical elements (LE) when implemented on the FPGA, depending on the number of elements in the correlator. This allows to maintain a balance between the required calculation accuracy and spent hardware resources, as well as to simplify the end device.

Keywords: Internet of Things, industrial IoT, correlation, FPGA, matched filter, autocorrelation.

Citation: Anufrienko A.Yu. Data processing by end devices in IoT systems. Computing, Telecommunications and Control, 2020, Vol. 13, No. 2, Pp. 7–13. DOI: 10.18721/JCSTCS.13201

ОБРАБОТКА ДАННЫХ КОНЕЧНЫМИ УСТРОЙСТВАМИ В СИСТЕМАХ ИНТЕРНЕТА ВЕЩЕЙ

А.Ю. Ануфриенко

Национальный исследовательский университет «Высшая школа экономики»,
Москва, Российская Федерация

Представлен корреляционный метод для обработки данных на конечных устройствах и сокращения объема передаваемых по сети данных. Вместо дорогих и сложных сетевых устройств разработчики могут применять дешевые и проверенные низкоскоростные решения Интернета вещей (ZigBee, NB IoT, BLE) для передачи данных. Новизна состоит в одной из особенностей этого подхода — использовании для анализа не полной копии сигналов, а их компонентов, а также обработке непосредственно на сенсоре. Данный подход позволяет уменьшить количество операций и сложность реализации в отличие от иных методов, ориентированных на парадигму облачных вычислений. Приведены результаты для значений корреляции и количества логических элементов (LE) при реализации на ПЛИС в зависимости от количества элементов в корреляторе. Это позволяет соблюдать баланс между требуемой точностью расчета и затраченными аппаратными ресурсами, а также упростить конечное устройство.

Ключевые слова: Интернет вещей, промышленный Интернет вещей, корреляция, ПЛИС, согласованный фильтр, автокорреляция.

Ссылка при цитировании: Anufrienko A.Yu. Data processing by end devices in IoT systems // Computing, Telecommunications and Control. 2020. Vol. 13. No. 2. Pp. 7–13. DOI: 10.18721/JCSTCS.13201

Introduction

Data are generating by End devices in Internet of Things systems: mainly group of sensors, social media, and applications. This massive data generation results in ‘‘big data’’, but not all kinds of data are valuable. Generally, the structure of IoT consists of five layers: Perception Layer, Network Layer, Middleware Layer, Application Layer, and Business Layer. Some of the Internet of Things architectures targeted to cloud computing at the center and a model of end-to-end interaction among various stakeholders in a cloud-centric IoT approach [1]. Cloud computing frees the enterprise and the end user from the specification of many details. This feature becomes a problem for latency-sensitive (industrial) applications, which require minimized delay. Fog computing extends the cloud computing paradigm to the edge of the network. The Fog vision was conceived to address applications and services that do not fit the paradigm of the Cloud well [3].

Typical architecture and components of IoT systems are presented in [1–4]. The systems include modules consisting of sensors, actuators and modems – devices that generate and transmit data. A number of sensors read and report the status of monitored objects. Industrial equipment may have thousands of points for data generation. The module may also have actuators for affecting the logical state of the tool. Modems transmit data to the next level – gate. The gate is usually a hardware component that interacts with a number of modules. The gate also interacts with a platform where the data are saved, processed and provided to end users. The platform (web-based platform) has a number of core components like storage systems, databases, AI and BI tools and an app engine support. So, one of specific devices in the system is the module because the quality of the final results and big data are depending on this device. Typical case for IoT system is transmitting data from modules to an IoT platform (cloud) as is, and deep processing with BI and AI tools.

The volume of generated data is often large, so its transfer to the cloud is limited by the network bandwidth. In industrial IoT applications 100 % of data should be analyzed, but not 100 % of the data should be saved. In addition, there are a lot of other applications (connected cars, smart city, assessments) when several groups of sensors are used in the tests or operations. In aviation, an aircraft engine can have as many as 250 sensors. A twin-engine aircraft on a 12-hour flight can produce up to 844 TB of data [5]. Widely used IoT wireless technologies have typical throughput of 10–250 kbps, and end devices may be autonomous (have an autonomous power supply) and low powered. In addition, there are not enough storage systems for terabytes and petabytes of raw data. This example demonstrates the complexity of using cloud-oriented approaches to analyze high-speed data streams.

The purpose of this paper is to review the existing data processing methods (cloud-edge) and research required computing resources and correlation efficiency depending on the complexity of data processing when processing on devices.

Clouds and endpoint architectures

One of the most commonly used approaches for IoT systems is the sampling rate adaptation [6–9]. A sampling rate is a rate at which a new sample is taken from a continuous signal provided by the sensor board. This rate can be adapted according to the input acquired from the monitoring area. If no significant change is noticed for a certain period of time, the sampling rate could be reduced for the upcoming period, and in contrast, if an event is detected, the sampling rate is increased. This sampling rate adaptation is based on event detection [7]. Data reduction approaches focus solely on reducing the number of transmissions while maintaining a fixed sampling rate [9]. The most popular of them all is the dual prediction scheme [10]. A prediction model capable of forecasting future values is trained and shared between the source and the destination, thus enabling the source sensor node to transmit only the samples that do not match the predicted value.

Another variation is a spatial-temporal correlation based approach for sampling and transmission rate adaptation in cluster-based sensor networks [11]. The correlation between sensor nodes and the new sampling rates of each sensor is calculated. This approach does not require any algorithm to be implemented on the sensor level, the only task performed by the sensors consists exclusively in sampling and transmission. All the work is done on the Cluster-Head (CH) level, where at the end of each round (duration predefined by the user) the CH runs an algorithm that finds the spatial correlation among the data reported by the sensors belonging to the same cluster. Then, it transmits to one of them its new sampling rate for the next round according to its level of correlation with other neighboring sensors in the cluster. The sampling rate scheduling follows a strict protocol that keeps the sampling rate of the sensors showing high correlation with a large number of nodes at an optimal maximum level [11].

In paper [12], the authors propose to capture such sensor data correlation changes to improve the performance of IoT (Internet of Things) equipment for anomaly detection. In a feature selection method, first cluster correlated sensors together to recognize the duplicated deployed sensors according to sensor data correlations, and monitor the data correlation changes in real time to select the sensors with correlation changes as the representative features for anomaly detection. Curve alignment and dynamic time warping (DTW) [13] are methods used for measuring similarity between two time series (data sequences). However, DTW methods do not assume a consistent time lag, and calculate an optimal matching between two given time series with certain restrictions to maximize a measure of their similarity [12]. But these methods involve working with big data at the cloud side.

There are two key issues:

1) Limited IoT network throughput and large amount of data.
2) Limited calculation resources near the sensors.

The general task for an IoT system is a reliable transmission of the data from the sensors to the platform for further analysis and visualization as Fig. 1a shows. As already mentioned, not all data should be transferred and stored for subsequent processing. For deep and precision analyzing, critical modes possess the utmost importance, especially at the assessment stage.

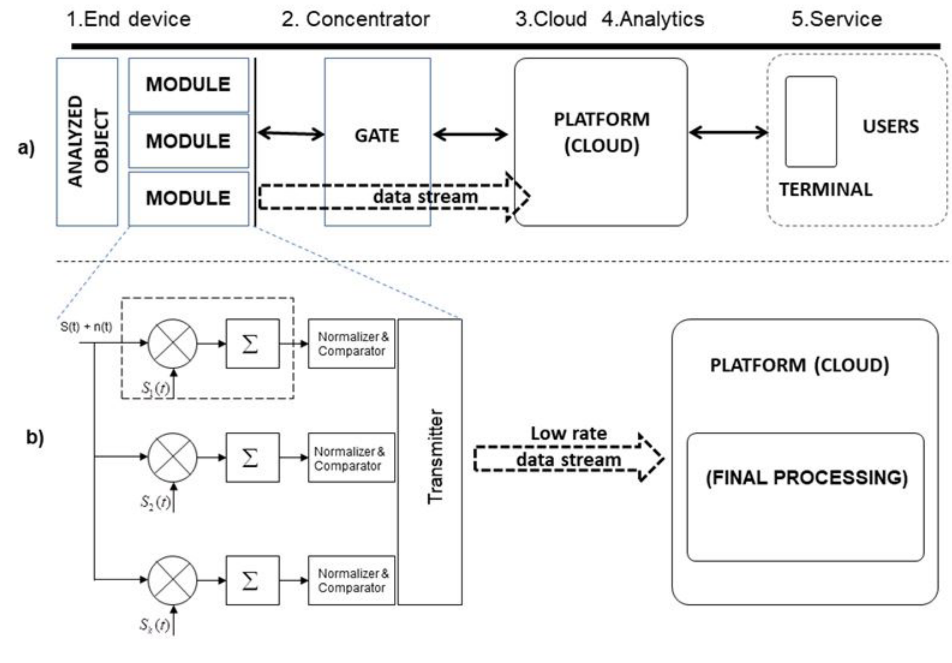

Fig. 1. Typical architecture of IoT systems (a) and simplified block diagram for a matched filter in a module and low rate data transmission to a cloud (b)

Design and results

To achieve the required results, we could use correlation methods. For many years, correlation methods have been applied in radar and sonar systems for range and position finding in which transmitted and reflected waveforms are compared. In robotic vision, they are used for remote sensing by satellite in which data from different images are compared. One of the applications of correlation is correlation detection implemented by the matched filter, which maximizes S/N ratio at its output. And output result of the matched filter is the autocorrelation at lag zero of input signal and its locally saved copy. But in IoT systems, we operate only with random signals, opposite to radar applications. The cross-correlation [14] between two digital sequences, each containing N data and normalized to the number of samples might be written as:

$$r_{12}(j) = \frac{1}{N}\sum_{n=0}^{N-1} x_1(n) * x_2(n+j), \tag{1}$$

where, the correlation should be calculated with lags. In case when sequence $x_1(n) = x_2(n)$, the process is known as autocorrelation and can be written as:

$$r_{11}(0) = \frac{1}{N}\sum_{n=0}^{N-1} x_1^2(n) = S, \tag{2}$$

where S is normalized energy of signal.

The cross-correlation values computed according to the above equations depend on the absolute values of the data. But it is often necessary to measure cross-correlations in a fixed range [–1; +1]. This can be achieved by normalizing the values by an amount depending on the energy content of the data. And the normalized expression for r_{12} becomes:

$$\rho_{12}(j) = \frac{r_{12}(j)}{\frac{1}{N}\sqrt{[\sum_{n=0}^{N-1} x_1^2(n) * \sum_{n=0}^{N-1} x_2^2(n)]}}, \tag{3}$$

where "+1" means complete coincidence (100 % correlation). Despite the fact that the result in the range [–1; +1] is convenient for understanding and analysis, the computational complexity of the denominator (3) is high and requires relatively large computational resources, especially the division operation.

In this paper, we analyze required computing resources and correlation efficiency depending on the complexity of the filter. Matched filters are detecting signals by comparing (determining the correlation) a known signal or pattern with a received signal. So, the number of samples of a known signal (saved copy) defines the number of taps of the matched filter. On-sensor processing concept means a combination of sensor and processor functions in a single device (System-on-chip). The module consists of a sensor, an analog-to-digital converter (ADC) and a fast processor. There are several types of devices suitable for the prototyping tasks – microcontrollers (MCU), digital signal processors (DSP) and field programmable gate arrays (FPGA). Most microcontrollers have a built-in sensors and ADC, but the operation frequency of the MCU is limited. In parallel processing, when a matched filter stores a set of local signals, implementation on the DSP is not the optimal solution. FPGA devices are more optimal for fast parallel processing in case of a matched filter. Cyclone IV FPGA family [16] was used as a base for system prototyping. The models were implemented with MATLAB. FPGA implementation used Verilog HDL. As it was said earlier, Signal-to-Noise ratio at filter's output depends on the quality of a stored copy of a signal. In a digital

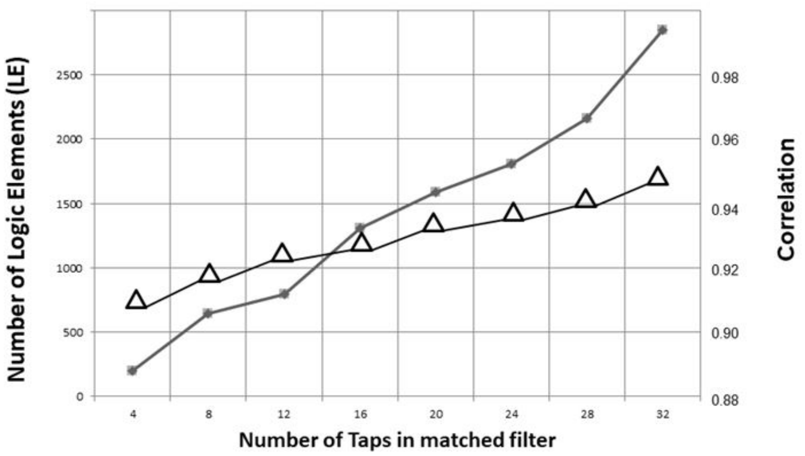

Fig. 2. Correlation values and number of logic elements (LE) depending on the number of taps in the correlator (△) – correlation; (♦) – N of LE

system, it means a number of samples of one signal. And the number of samples defines the number of taps in the filter. We used triangular waveforms described by 4, 8, 12, 16, 20, 24, 28 and 32 samples in the experiment. Thus, eight 14-bit matched filters were implemented in the FPGA in accordance with the models made in MATLAB. The use of FPGA resources is analyzed only for multipliers and integrators, shown in Fig. 1b. The amount of logical resources required for the division and square root operations is constant. The results of modeling and implementation are shown in Fig. 2.

199 LEs, 63 registers are required for the 4-tap filter, and a correlation value of 0.91 is reached. And 2646 LEs, 477 registers are required for the 32-tap filter, and a correlation value of 0.948 is reached. In case we need to analyze 100 types of signals in a data stream, 19 900 LEs will be necessary in the first case and 264 600 LEs in the second. Of course, operation frequency will be lower in the second case. Fmax for Cyclone IV is equal to 133 MHz, and the presented method gives a delay of 2 clocks, which is unattainable for MCUs and DSPs, and with parallel processing on the FPGA the performance gain will be more significant. The cost of an FPGA device suitable for implementing 100 32-tap filters can be 10 times higher than that of a simple FPGA device. For ASIC implementation, the number of gates is also crucial.

Conclusion

The key results presented in the article are summarizing fast data processing based on correlation with orientation to FPGA/ASIC. Correlation processing allows to reduce the amount of data transmitting from a sensor to the cloud and to simplify the IoT network architecture. Sensor-based data processing has more advantages than the cloud computing approach, where less than 100 % of the data is required for transmission, storage, and analysis. The applications are especially important in industrial systems (industrial IoT). Due to the large number of multiplications and divisions, correlators require a large amount of hardware resources.

Simulation results show the effectiveness of event detection. The dependence of the required hardware resources of the FPGA on the correlation value increases non-linearly. As the number of taps increases, the performance of the system (Fmax) decreases, so it is important to maintain a balance between accuracy and resource consumption. However, instead of expensive and complex network devices (Fog, Edge, Cloud computing), engineers can use cheaper IoT solutions. In future designs, it is more promising to use SoC solutions that include sensors, ADCS, microcontrollers and logic cores.

REFERENCES

1. **Khan R., Khan S.U., Zaheer R., Khan S.** Future internet: The Internet of Things architecture, possible applications and key challenges. *Proceedings of the 10th International Conference on Frontiers of Information Technology (FIT '12)*, 2012, Pp. 257–260.

2. **Weyrich M., Ebert C.** Reference architectures for the Internet of Things. *IEEE Software*, 2016, Vol. 33, No. 1, Pp. 112–116.

3. **Bonomi F., Milito R., Natarajan P., Zhu J.** Fog computing: a platform for Internet of Things and analytics. *Big Data and Internet of Things: A Road Map for Smart Environments*. Springer, Berlin, Germany, 2014, Pp. 169–186.

4. **Marjani M., et al.** Big IoT data analytics: Architecture, opportunities, and open research challenges. *IEEE Access*, 2017, Vol. 5, Pp. 5247–5261.

5. Engines in the Data Cloud. Available: *https://www.digitalcreed.in/engines-data-cloud/, April, 10, 2018.*

6. **Bhuiyan M.Z.A., Wu J., Wang G., Wang T., Hassan M.M.** e-sampling: Event-sensitive autonomous adaptive sensing and low-cost monitoring in networked sensing systems. *ACM Trans. Auton. Adapt. Syst.*, 2017, Vol. 12.

7. **Harb H., Makhoul A.** Energy-efficient sensor data collection approach for industrial process monitoring. *IEEE Trans. Ind. Informat.*, 2018, Vol. 14, No. 2, Pp. 661–672.

8. **Tayeh G.B., Makhoul A., Laiymani D., Demerjian J.** A distributed real-time data prediction and adaptive sensing approach for wireless sensor networks. *Pervasive Mobile Comput.*, 2018, Vol. 49, Pp. 62–75.

9. **Tayeh G.B., Makhoul A., Demerjian J., Laiymani D.** A new autonomous data transmission reduction method for wireless sensors networks. *Proceedings of the IEEE Middle East North Afr. Commun. Conf. (MENACOMM)*, 2018, Pp. 1–6.

10. **Braten A.E., Kraemer F.A., Palma D.** Adaptive, correlation-based training data selection for IoT device management. *Proceedings of the 6th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, Granada, Spain, 2019, Pp. 169–176.

11. **Tayeh G.B., Makhoul A., Perera C., Demerjian J.** A spatial-temporal correlation approach for data reduction in cluster-based sensor networks. *IEEE Access*, 2019, Vol. 7, Pp. 50669–50680.

12. **Su S., Sun Y., Gao X., Qiu J., Tian Z.** A correlation-change based feature selection method for IoT equipment anomaly detection. *Applied Sciences*, 2019, Vol. 9(3), P. 437.

13. **Kim S., Lee H., Ko H., Jeong S., Byun H., Oh K.** Pattern matching trading system based on the Dynamic TimeWarping Algorithm. *Sustainability*, 2018, No. 10, P. 4641.

14. **Ifeachor E., Jervis B.** *Digital signal processing: A practical approach*. Hardcover, 2nd ed., USA: Prentice Hall, 2001, Pp. 184–245.

15. **Oppenheim A.V., Schafer R.W., Buck J.R.** *Discrete-time signal processing*. Hardcover, 2nd ed., USA: Prentice Hall, 1998. Pp. 746–753.

16. Cyclone IV Device Handbook, 490 p. Available: *https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/cyclone-iv/cyclone4-handbook.pdf.* (Accessed: March 2016).

Received 03.04.2020.

СПИСОК ЛИТЕРАТУРЫ

1. **Khan R., Khan S.U., Zaheer R., Khan S.** Future internet: The Internet of Things architecture, possible applications and key challenges // Proc. of the 10th Internat. Conf. on Frontiers of Information Technology. 2012. Pp. 257–260.

2. **Weyrich M., Ebert C.** Reference architectures for the Internet of Things // IEEE Software. 2016. Vol. 33. No. 1. Pp. 112–116.

3. **Bonomi F., Milito R., Natarajan P., Zhu J.** Fog computing: a platform for Internet of Things and analytics // Big Data and Internet of Things: A Road Map for Smart Environments. Springer, Berlin, Germany, 2014. Pp. 169–186.

4. **Marjani M., et al.** Big IoT data analytics: Architecture, opportunities, and open research challenges // IEEE Access. 2017. Vol. 5. Pp. 5247–5261.

5. Engines in the Data Cloud // URL: https://www.digitalcreed.in/engines-data-cloud/, April, 10, 2018.

6. **Bhuiyan M.Z.A., Wu J., Wang G., Wang T., Hassan M.M.** e-sampling: Event-sensitive autonomous adaptive sensing and low-cost monitoring in networked sensing systems // ACM Trans. Auton. Adapt. Syst. 2017. Vol. 12.

7. **Harb H., Makhoul A.** Energy-efficient sensor data collection approach for industrial process monitoring // IEEE Trans. Ind. Informat., 2018, Vol. 14, No. 2, Pp. 661–672.

8. **Tayeh G.B., Makhoul A., Laiymani D., Demerjian J.** A distributed real-time data prediction and adaptive sensing approach for wireless sensor networks // Pervasive Mobile Comput. 2018. Vol. 49. Pp. 62–75.

9. **Tayeh G.B., Makhoul A., Demerjian J., Laiymani D.** A new autonomous data transmission reduction method for wireless sensors networks // Proc. of the IEEE Middle East North Afr. Commun. Conf. 2018. Pp. 1–6.

10. **Braten A.E., Kraemer F.A., Palma D.** Adaptive, correlation-based training data selection for IoT device management // Proc. of the 6th Internat. Conf. on Internet of Things: Systems, Management and Security. Granada, Spain, 2019. Pp. 169–176.

11. **Tayeh G.B., Makhoul A., Perera C., Demerjian J.** A spatial-temporal correlation approach for data reduction in cluster-based sensor networks. IEEE Access, 2019, Vol. 7, Pp. 50669–50680.

12. **Su S., Sun Y., Gao X., Qiu J., Tian Z.** A correlation-change based feature selection method for IoT equipment anomaly detection // Applied Sciences. 2019. Vol. 9(3). P. 437.

13. **Kim S., Lee H., Ko H., Jeong S., Byun H., Oh K.** Pattern Matching Trading System Based on the Dynamic Time Warping Algorithm // Sustainability 2018, No. 10, P. 4641.

14. **Ifeachor E., Jervis B.** Digital signal processing: A practical approach. Hardcover, 2nd ed., USA: Prentice Hall, 2001. Pp. 184–245.

15. **Oppenheim A.V., Schafer R.W., Buck J.R.** Discrete-time signal processing. Hardcover, 2nd ed., USA: Prentice Hall, 1998. Pp. 746–753.

16. Cyclone IV Device Handbook, 490 p. // URL: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/cyclone-iv/cyclone4-handbook.pdf. (Accessed: March 2016).

Статья поступила в редакцию 03.04.2020.

THE AUTHOR / СВЕДЕНИЯ ОБ АВТОРЕ

Anufrienko Alexander Yu.
Ануфриенко Александр Юрьевич
E-mail: alexanuf@mail.ru