Network Coding-Based Next-Generation IoT for Industry 4.0

Goiuri Peralta, Raul G. Cid-Fuentes, Josu Bilbao and
Pedro M. Crespo

Additional information is available at the end of the chapter

Abstract

Industry 4.0 has become the main source of applications of the Internet of Things (IoT), which is generating new business opportunities. The use of cloud computing and artificial intelligence is also showing remarkable improvements in industrial operation, saving millions of dollars to manufacturers. The need for time-critical decision-making is evidencing a trade-off between latency and computation, urging Industrial IoT (IIoT) deployments to integrate fog nodes to perform early analytics. In this chapter, we review next-generation IIoT architectures, which aim to meet the requirements of industrial applications, such as low-latency and highly reliable communications. These architectures can be divided into IoT node, fog, and multicloud layers. We describe these three layers and compare their characteristics, providing also different use-cases of IIoT architectures. We introduce network coding (NC) as a solution to meet some of the requirements of next-generation communications. We review a variety of its approaches as well as different scenarios that improve their performance and reliability thanks to this technique. Then, we describe the communication process across the different levels of the architecture based on NC-based state-of-the-art works. Finally, we summarize the benefits and open challenges of combining IIoT architectures together with NC techniques.

Keywords: industry 4.0, IoT, IIoT, latency, network coding, fog computing, multicloud

1. Introduction

Industry 4.0, that is, the fourth industrial revolution, represents industry and manufacturing digitalization bringing with it, among other things, the so-called smart factories. This transformation comes through the adoption of the Internet of Things (IoT) [1], which gives rise to

the Industrial IoT (IIoT) and allows to interconnect humans, machines, and smart devices, as well as to share huge amounts of data among them.

In order to cope with big data and predictive analytics [2], cloud computing is becoming another key enabler due to its computing, storage, and networking capabilities. It allows us to obtain meaningful information and valuable insights which will increase the efficiency, productivity, and performance of manufacturing processes and services. Several IIoT applications, such as system control, anomaly detection, or robot guidance, are time-critical, and therefore, they require millisecond response times. Thus, low-latency communications, as well as real-time analysis and monitoring, are indispensable for immediate decision-making.

Although the cloud offers high scalability, flexibility, and responsiveness, cloud-based analytics may introduce excessive latency, which would compromise the performance of time-critical applications. In order to accomplish a trade-off between latency and computation, IIoT deployments are moving cloud capabilities downwards to fog nodes to perform early analytics and minimize latency. Furthermore, most delay-critical applications not only require low-latency communications but also ensure high reliability. A promising technique that increases network reliability while reducing end-to-end latency is network coding (NC). Its properties are particularly beneficial for enhancing the robustness and reducing delays of wireless sensor network (WSN) communications [3]. Moreover, it improves the efficiency of distributed storage systems, regarding both data download speed and redundancy [4].

In this chapter, we overview next-generation IIoT systems, which must provide low-latency communications as well as ensure their reliability in order to allow the performance of on-premise advanced cloud analytics for time-critical IIoT applications, that is, to bring the cloud to the fog (see **Figure 1**). This objective can be achieved by implementing a three-layer architecture based on IoT nodes, fog nodes and a multicloud environment, and also by exploiting the advantageous properties of NC techniques across the architecture.
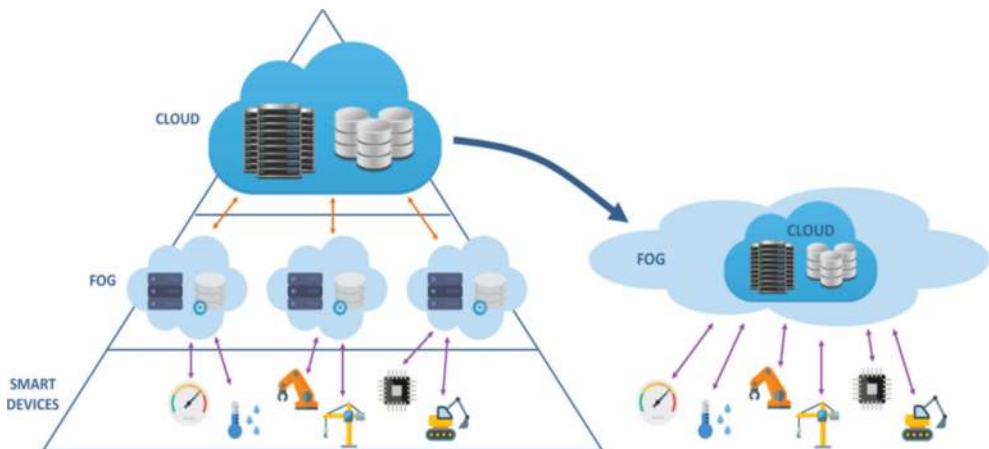

Figure 1. Bringing the cloud to the fog.

The rest of this chapter is organized as follows. First, we overview next-generation IIoT architectures, briefly describing and comparing the different layers as well as providing different use-cases in which these architectures are integrated. Next, we introduce some NC approaches and describe the benefits of NC regarding different scenarios. Then, we describe the communication process across the different levels of the architecture. We also summarize the benefits of merging IIoT architectures and NC techniques. Finally, we discuss existing issues and open challenges, and we report the final conclusions of the chapter.

2. Next-generation IIoT architectures

Nowadays, due to its scalability and big data management capabilities, cloud-based architectures are most widely used in Industry 4.0 applications. However, the integration of the IoT into industrial environments poses new challenges, which implies an architectural adaptation. As previously mentioned, IIoT applications are mostly delay-sensitive and require instant decision-making. This has led to the integration of fog nodes into the industrial systems in order to perform early analytics and closed-loop control. Moreover, systems of this nature must be robust. Thus, with the aim of providing a fault-tolerant architecture and guarantee system reliability, multicloud deployments are emerging as a promising solution. In addition to the latter, they enable to use the connections under the best conditions and therefore, delays can be reduced. Dependencies on a single cloud provider can also be avoided.

It can be said that next-generation IIoT architectures, as shown in **Figure 2**, will consist of three layers, composed of IoT or smart devices, fog nodes and multiple clouds. The lowest layer, comprised of a variety of end-nodes, is responsible for sending taken measurements to actuators or fog devices. In the fog layer, time-critical analytics, as well as closed-loop control, can be performed. Finally, cloud servers are in charge of heavy data analytics and compute-intense workloads that manage a vast amount of data.

2.1. Architecture design

A description and comparison of the layers that comprise next-generation IIoT architectures are next provided.

2.1.1. WSN

WSNs can be considered the main communication technologies of IIoT due to the flexibility they offer to connect and manage a large number of sensors and actuators, independently of their location. A WSN consists of several IoT nodes, including sensors, actuators, and smart devices, which take several measurements. These devices are mainly battery, storage, and processing power constrained. This layer is responsible for gathering sensor data, such as machine temperature or vibration measurements, and for uploading them. It also receives instructions from the upper layers in order to perform a corresponding task or action.
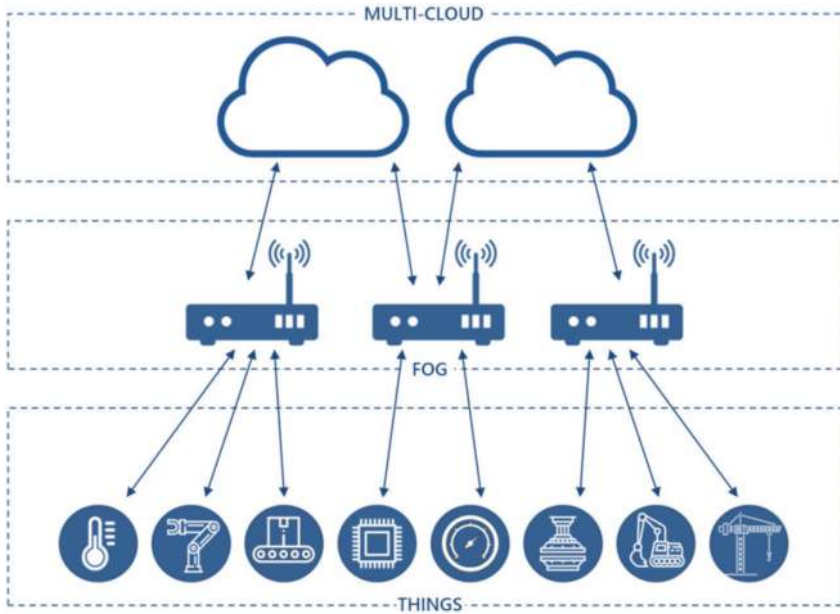
Figure 2. Next-generation IIoT architecture.

2.1.2. Fog

The fog can be considered as an intermediate layer between the cloud and IoT devices and so, it extends cloud computing capabilities to the edge of the network [5]. One of its main advantages is its closeness to the end-nodes, which makes possible to reduce communication latency and to enable real-time service support. Since fog computing allows early data processing, the amount of data sent to the cloud can be reduced. In addition, its mobility and location-awareness enable to deliver rich services to moving devices [6].

2.1.3. Multicloud

The cloud can be described as several distributed remote servers which can be accessed via the Internet to store and manage big amounts of data [7]. Cloud computing enables the remote on-demand use of computing resources, that is, networks, servers, storage, applications, and services. It provides virtualized, elastic, and controllable services and powerful computational capabilities, enabling complex application systems at lower costs. The deployment of more than one cloud, in addition to the mentioned advantages, provides fault tolerance against service outages, and the system security level is improved since it is possible to store the information divided into different clouds. Furthermore, application requirements can be better adapted to available cloud resources and connectivity conditions.

2.1.4. Comparison

With the adoption of IoT, the number of things connected to the Internet is expected to grow up to 20 billion in 2020 [8]. Thus, a scalable architecture is required in order to adapt to such a huge number of devices. Moreover, the need for large amounts of data to be accessed more quickly is ever-increasing, where the inherent latency of the cloud can be detrimental. Latency issues become highly damaging, particularly for IIoT time-critical applications. Autonomous decisions are required in order to prevent failures or optimize production, and thus, milliseconds matter when trying, for instance, to prevent manufacturing line downtimes or to get the right decision in autonomous vehicles.

Processing data directly in the end-devices would be the best solution in order to provide the lowest latency and jitter. However, the constrained nature of these nodes inhibits the performance of more advanced processing and analytics. Thus, fog computing can be the most suitable solution for applications that cannot afford the delay caused by the round trip to the cloud server. Nonetheless, fog computing requires local management of redundancy and data backup. Moreover, the integration of devices capable of performing remote data analytics implies the increase of the architecture complexity, as well as of the associated costs in hardware and software investments. **Table 1** shows the most significant differences between WSNs, fog computing, and cloud computing.

2.2. Application use-cases

The three-layer architecture enables to exploit the efficiency and scalability of the fog while benefiting from the powerful storage and computing resources of the cloud. Next, we show some use-case examples.

2.2.1. Smart energy

Wind energy-based smart grids require data analysis and real-time decision making. In a large wind farm, the health of the turbines is monitored by analyzing data collected by

Feature	WSNs	Fog computing	Cloud computing
Latency	Very low	Low	High
Delay jitter	Very low	Low	High
Server location	—	Local	Internet
Client–server distance	—	One hop	Multiple hops
Location awareness	Yes	Yes	No
Distribution	Highly distributed	Distributed	Centralized
Mobility awareness	Guaranteed	Supported	Limited
Real-time interactions	Guaranteed	Supported	Limited

Table 1. Comparison between WSN, fog, and cloud computing [9].

numerous sensors [10]. Each turbine can be monitored locally, that is, in the fog, and the collective performance can be improved by processing data on remote servers in the cloud. Thus, it enables to combine real-time response for early actions and advanced analyses for a deeper view of the whole wind farm. It can increase energy output, decrease operational costs, and increase turbine uptime.

2.2.2. Smart transportation

Self-driving vehicles, for instance, are equipped with an on-board system that, through real-time data analysis, allows controlling the car without human interaction. In such systems, highly reliable and low latency communication is crucial. Thus, critical decisions that require an instantaneous response are better managed with fog computing [11]. However, for monitoring the tracking performance of a truck fleet [12], as there is no need for real-time analytics, cloud computing is more suitable. Advanced cloud analytics based on information gathered from different parts of the truck can bring insights to improve the maintenance and lower repair costs.

2.2.3. Smart manufacturing

Smart factories are able to perform predictive maintenance of their machines or improve product quality by real-time sensor analysis [13]. Fog computing is crucial for these delay-critical data processing. However, cloud computing can provide an overall system management as well as machine learning analytics that require greater computing power.

2.2.4. Smart cities

Fog computing can provide a fast, real-time, and location-aware solution for many IoT use cases of smart cities, such as smart buildings [14]. Several sensors gather diverse measurements like temperature, energy usage, humidity, parking occupancy, air quality, elevators, smoke, and so on. The efficiency of the system can be improved by managing critical data at the fog layer in real time, as in traffic control, and by performing big data analytics in the cloud.

3. Next-generation communications

Most IIoT systems are deployed in harsh environments, where different devices within the architecture can be connected and disconnected from the network any time. Thus, besides providing low-latency communications, it is crucial to strengthen these communications in order to ensure a robust and highly reliable environment.

3.1. Issues

IIoT networks require system reliability, data availability and high communication quality. This may be difficult to achieve due to inherent constraints of these scenarios. WSNs, for example, may suffer from noise or multipath interferences, among others, which cause packet loss and inevitably degrades the quality of the communications. Moreover, the dynamic topology of these architectures in which devices connect intermittently, can destabilize

communications and introduce variable delays. In order to overcome these issues, the integration of NC techniques across the shown architecture can be a suitable solution due to its properties.

3.2. Network coding

NC breaks with the traditional store-and-forward transmission model [15] by allowing any intermediate node to recombine incoming packets into coded ones, which are decoded at destination. Its properties make it a promising solution to improve the performance of wireless and peer-to-peer networks. It exploits the broadcast nature of the wireless medium [3], which facilitates node cooperation to provide significant benefits in terms of communication robustness, stability, throughput, and latency.

Moreover, dependency on obtaining a particular packet is removed by applying NC since it is sufficient to get enough linearly independent packet combinations in order to recover the required data. Thus, in distributed systems, such as P2P [16] or multicloud environments [17], the use of NC can reduce additional data download or access delays in highly loaded conditions as well as improve the performance of data recovery and acquisition [4].

3.2.1. Advanced techniques

Advanced NC techniques are based on the widely used NC approach random linear network coding (RLNC) [18], where the received κ packets are linearly combined with randomly chosen coefficients from a finite field or Galois Field (GF) $\mathbb{F}_q$, before forwarding them. The performance of RLNC is influenced by the following coding parameters: finite field size, generation size, and coding vector density [19], among others. Eq. (1) represents the encoding process, where coding coefficients ($\alpha_{i,j} \in \mathbb{F}_q$) multiply incoming packets $M_i (\forall i = 1, \ldots, K)$ to generate corresponding encoded packets X_i.

$$
\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_K \end{bmatrix} = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,N} \\ \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{K,1} & \alpha_{K,2} & \cdots & \alpha_{K,N} \end{bmatrix} \cdot \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_K \end{bmatrix} \tag{1}
$$

Several variants of this technique have been developed in order to adapt it to different scenarios and application requirements. Perpetual codes [20], for instance, can be considered as a supplement of RLNC. In manifold scenarios, particularly for large generation sizes, they can substantially increase the throughput due to their sparsity and the possibility of structured decoding.

For heterogeneous networks with devices of different resources, fulcrum codes [21] allow to use binary GF operations in the network to achieve reduced overhead and computational cost, and reach compatibility with heterogeneous devices and data flows in the network, while providing the opportunity of employing higher coding finite fields end-to-end for greater performance. On the other hand, systematic coding [22] allows sending coded packets along with original ones, that is, uncoded packets, which can help to reduce overhead and improve the real-time decoding performance.

For low-delay applications such as real-time control applications, on-the-fly or sliding window coding can be the most suitable solutions. Unlike block codes where all packets in a block need to be present to start generating useful coded packets, on-the-fly codes [23] are able to encode data while they become available and these packets are progressively decoded. Sliding window codes [24] are more flexible since they remove the limitation of fixed blocks by creating a variable-sized sliding window.

Finally, Tunable Sparse Network Coding (TSNC) [25], unlike RLNC that applies a fixed coding density for the entire process, tunes the density of coded packets during transmission to adjust to the trade-off between real-time performance and reliability. TSNC proposes to increase the coding density as the destination node receives more linearly independent packets since the probability of receiving innovative packets is lower, thereby reducing coding complexity.

3.2.2. NC benefits

This section lists different benefits of NC, showing its suitability for IIoT systems and applications.

- *Distributed storage systems*: IIoT systems require high reliability and availability. Thus, data must be distributed and stored in such a way as to ensure fault tolerance, for example in the event of a server failure. Packet loss, delay, and bandwidth fluctuation can hinder data distribution. The main benefit of NC over P2P environments is in relation to the coupon collector problem [16], being able to solve this issue due to the redundancy introduced in packet transmissions. Therefore, the performance of data streaming is enhanced since download times are minimized. With NC, the performance of the system depends much less on the underlying topology and schedule.

  NC can help to increase the reliability of distributed storage systems like multicloud deployments [17]. In case of data loss, the amount of redundant data required for repair is minimized. In addition, each cloud is used at its maximum speed even in highly loaded conditions or dynamically changing environments. Thus, NC improves storage efficiency in terms of data retrieval time and storage space.

- *Dynamic topologies*: NC techniques can be helpful for efficient content distribution [26] in changing environments. For example in Vehicular Ad-Hoc Networks (VANETs), in order to avoid possible accidents, vehicles exchange road state information among them. Even in dynamic road changing conditions, VANET applications, such as traffic live video broadcast, must guarantee a correct data reception. Since NC enhances network performance and reduces the number of required data transmissions, it can reduce transmission delays.

  Due to the previous and together with its decentralized nature and robustness, NC can be extrapolated also to dynamic IIoT architectures, where end-nodes may connect periodically in order to save power or they can connect to different access points.

- *Constrained environments*: the use of NC has been extended also to constrained environments. In satellite communications, for instance, bandwidth is usually limited and round-trip

delays are high. The properties of this technique can be advantageous particularly in mul-tibeam satellites [27]. On the one hand, NC improves throughput and bandwidth usage. On the other hand, it does not require any change at the physical layer and thus, it easies the implementation on already deployed satellite systems. NC techniques can also be used in applications aimed at energy-efficient data transmissions, such as Wireless Body Area Networks (WBANs), since they can provide reliable communications under low-energy constraints [28]. Therefore, IIoT applications can also profit from this technique as the majority of end-devices are resource constrained.

- Poor quality channels: NC can improve the transmission performance in environments with unstable channel conditions which quality may not meet end-user quality of ser-vice (QoS) requirements, such as delay and reliability. An example of the previous are Underwater Sensor Networks (UWSNs) and Power Line Communication (PLC) systems. In UWSNs, the acoustic communications suffer high error rates and long propagation delays, which require efficient error recovery. NC can exploit the broadcast property of acoustic channels, improving data throughput [29]. PLC systems, on the other hand, are able to provide multicast and broadcast services by exploiting existing electrical wires. Due to the similarities between power line and wireless channels, NC protocols can be applied in order to achieve the implementation constraints [30] and provide reliable communica-tions in harsh environments.

IIoT systems that relay on WSNs may deal with interferences or channel contention that cause QoS issues. Thus, NC-based techniques can help to improve channel resources as well as data rate while maintaining QoS.

4. NC over IIoT architectures

In this section, we introduce NC into next-generation IIoT architectures reviewing related state-of-the-art works. We also outline some of the most relevant benefits and challenges.

4.1. Communication process

In IIoT systems, not only low-latency communications between end-nodes (things) and the cloud must be guaranteed but the whole system must be robust, including the connections and the provided service. Next, the communication process across the architecture is described.

4.1.1. Things

Implementing NC techniques through the WSN, communication latency can be reduced [31] and its robustness [32, 33] improved. Here, sensors and actuators combine their mea-surements and transmit them across the network. By using NC, intermediate nodes recode received data and send them to one or more gateways which compose the fog layer. These devices are then in charge of uploading incoming data to the multicloud framework.

4.1.2. Fog

Fog nodes can be any device with computing, storage, and network connectivity, such as controllers, routers, gateways, and so on. They can be deployed anywhere with a network connection, for instance, alongside a factory floor. They are interconnected among them, with the IoT devices and also with cloud servers, forming a distributed network. Therefore, NC-based techniques can be extrapolated from WSNs to the communication between the devices within the fog layer [34, 35].

The fog layer is responsible for gathering data from end-devices and for distributing coded packets to the different clouds that comprise the multicloud deployment. The use of NC has also been demonstrated to be beneficial for data distribution [36]. Moreover, this technique is advantageous for distributed storage systems, since it can achieve an optimal trade-off between storage and repair traffic. Thus, it can also help to deal with fog storage nodes that may continuously leave the network without a replacement [37].

4.1.3. Multicloud

Clouds within the multicloud deployment are responsible for storing incoming network-coded data from the lower layer. NC-based techniques can improve the process of lost data recovery, as well as enhance the efficiency of data redundancy [38]. As an example, **Figure 3** illustrates the repair operation in case of a cloud failure using exact minimum-storage regenerating (EMSR) codes. A file is divided into for fragments, and both original and coded chunks are distributed as shown in the figure. Assuming Cloud 1 is down (A and B are lost), the surviving nodes XOR their own chunks to create new encoded ones in order to make possible the reconstruction of A and B.
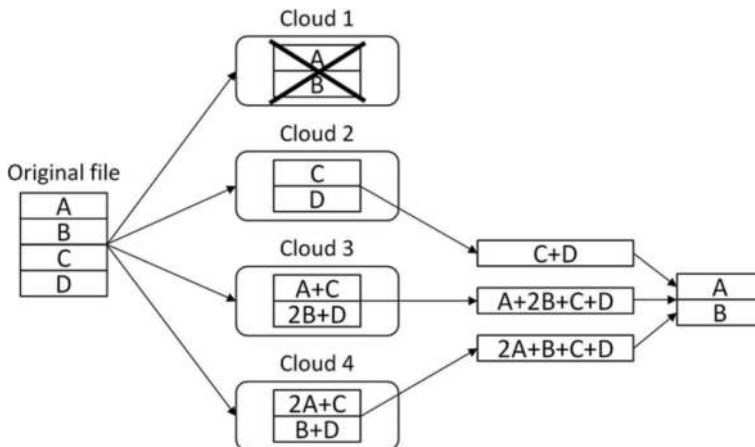

Figure 3. Repair process with EMSR codes [17].

Clouds are then able to perform required operations for decision-making or further analytics. Data or action commands are transmitted to the corresponding devices based on the results of the performed analysis. The cloud response time can also be reduced, as in the upstream communication, due to the implementation of NC. In [39], for instance, if a file has been divided into g fragments, the number of coded packets P_i stored on each of the N clouds is calculated in order to achieve optimal scheduling during data retrieval (see Eqs. (2) and (3), where R_i defines the download rate and α_i denotes the minimum part of the data to be stored).

$$\alpha_i = \frac{R_i}{\sum\limits_{i=1}^{N} R_i} \tag{2}$$

$$P_i \geq \alpha_i \cdot g \tag{3}$$

If the information is stored over distributed untrusted platforms, such as public clouds, the inherent use of NC schemes can provide, in addition to fault tolerance, a security level against eavesdroppers [40]. However, it is necessary to find a trade-off between fault-tolerance and security [41], since the more redundant data, the more vulnerable the system becomes.

4.2. Summary

We overview the most relevant benefits the integration of IIoT architectures and NC provides. While this approach can bring significant advantages, it also poses some issues. We identify and describe future challenges that may arise with the implementation of next-generation IIoT architectures.

4.2.1. Benefits

As the IIoT architecture relies on a multicloud deployment, the reliability and availability of the entire system can be enhanced. Data are distributed across different clouds, and so, the possibilities of suffering a cyber-attack are reduced. Moreover, this information is stored differently from the original form. Thus, data privacy is improved. The multicloud environment, due to its fault tolerance, increases the robustness in case of service outages. Due to the implementation of multiple clouds, this architecture enables to distribute data to the most convenient cloud, which makes possible not only to choose the service provider that better fits the moment requirements but to use the connections under the best conditions. Thus, it helps to identify the right service architecture to optimize latency, location, and cost.

The use of NC-based techniques can enhance the performance of the communications over congested WSNs, as well as of the data distribution and recovery processes over multicloud deployments. Since data redundancy is more efficient, reliability and availability of the provided service are improved. Besides, the integration of NC-based techniques into the architecture can lead to the reduction of end-to-end latency.

All of the above advantages allow the computational power of the cloud to be available closer to the end nodes, improving the performance of delay-sensitive IIoT applications.

4.2.2. Challenges

As mentioned throughout the chapter, next-generation IIoT architectures aim to reduce the end-to-end communication latency and to increase the system reliability by merging both fog and multicloud-based schemes as well as NC techniques. However, the use of complex NC schemes can result in extra delays taking into account that IoT devices have limited computational resources. Thus, in order to exploit the benefits of this technique, it is crucial to choose the most suitable coding parameters as well as to design simpler coding schemes and adaptable scheduling and routing algorithms.

IIoT systems must also provide scalability and flexibility. A cloud environment is inherently a scalable architecture due to its capability to manage network topology variations while handling big amounts of data. However, in architectures such as the proposed, as devices may be intermittently connected to the network, not only the architecture itself needs to be scalable and adaptable to changing environments, but also the coding techniques.

5. Conclusions

This chapter overviews next-generation IIoT systems which, in order to satisfy the demands of Industry 4.0 applications, must ensure low-latency and highly reliable communications. This will enable advanced analytics for time-critical IIoT applications. The previous objective can be achieved on one hand, by implementing a three-layer architecture based on IoT devices, fog nodes, and a multicloud deployment. On the other hand, the use of NC techniques across this architecture can improve the communication quality and increase the system reliability. In this chapter, we describe next-generation IIoT architectures and provide different application use-cases where they can be applied. We also review NC-based techniques and the benefits of this technique for different scenarios. Next, we describe the introduction of NC for the communications across the architecture. We also outline the advantages of the approach and finally, we present some challenges that may arise, such as the design of scalable and adaptive coding schemes and routing algorithms, and which may inspire future research lines.

Acknowledgements

Author details

Goiuri Peralta[1]*, Raul G. Cid-Fuentes[1], Josu Bilbao[1] and Pedro M. Crespo[2]

*Address all correspondence to: gperalta@ikerlan.es

1 IK4-Ikerlan Technology Research Center, Arrasate-Mondragón, Spain

2 University of Navarra (Tecnun), Donostia-San Sebastián, Spain

References

[1] Da Xu L, He W, Li S. Internet of things in industries: A survey. IEEE Transactions on Industrial Informatics. 2014;**10**(4):2233-2243. DOI: 10.1109/TII.2014.2300753

[2] Yue X, Cai H, Yan H, Zou C, Zhou K. Cloud-assisted industrial cyber-physical systems: An insight. Microprocessors and Microsystems. 2015;**39**(8):1262-1270. DOI: 10.1016/j.micpro.2015.08.013

[3] Fragouli C, Boudec JYL, Widmer J. Network coding: An instant primer. ACM SIGCOMM Computer Communication Review. 2006;**36**(1):63-68. DOI: 10.1145/1111322.1111337

[4] Dimakis AG, Godfrey PB, Wu Y, Wainwright MJ, Ramchandran K. Network coding for distributed storage systems. IEEE Transactions on Information Theory. 2010;**56**(9):4539-4551. DOI: 10.1109/TIT.2010.2054295

[5] Bonomi F, Milito R, Zhu J, Addepalli S. Fog computing and its role in the internet of things. In: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing (MCC '12); 17 August 2012. ACM; 2012. pp. 13-16

[6] Peralta G, Iglesias-Urkia M, Barcelo M, Gomez R, Moran R, Bilbao J. Fog computing based efficient IoT scheme for the industry 4.0. In: International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM); 24-26 May 2017. IEEE; 2017. pp. 1-6

[7] Zhang Q, Cheng L, Boutaba R. Cloud computing: State-of-the-art and research challenges. Journal of Internet Services and Applications. 2010;**1**(1):7-18. DOI: 10.1007/s13174-010-0007-6

[8] Gartner Inc. Analysts to Explore the Value and Impact of IoT on Business. Gartner Symposium/ITxpo [Internet]. 2015. Available from: https://www.gartner.com/newsroom/id/3165317 [Accessed: January 10, 2018]

[9] Cisco Blog. IoT, from Cloud to Fog Computing [Internet]. 2015. Available from: https://blogs.cisco.com/perspectives/iot-from-cloud-to-fog-computing [Accessed: January 05, 2018]

[10] FOGHORN. Wind Turbine Optimization [Internet]. 2018. Available from: https://www.foghorn.io/wind-turbine-optimization/ [Accessed: January 10, 2018]

[11] Automation Alley. Fog computing: A New Paradigm for the Industrial IoT [Internet]. Available from: https://www.automationalley.com/Blog/October-2017/Fog-Computing-A-New-Paradigm-for-the-Industrial-Io.aspx [Accessed: January 05, 2018]

[12] Thorn Technologies. How Edge Computing and the Cloud will Power the Future of IoT [Internet]. Available from: https://www.thorntech.com/2017/11/edge-computing-and-the-cloud-future-of-iot/ [Accessed: January 10, 2018]

[13] Fog in the Factory [Internet]. Available from: https://industrial-iot.com/2017/01/fog-in-the-factory/ [Accessed: January 05, 2018]

[14] Dutta J, Sarbani R. IoT-fog-cloud based architecture for smart city: Prototype of a smart building. In: 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence; 12-13 January, 2017. IEEE; 2017. pp. 237-242

[15] Ahlswede et al. Network information flow. IEEE Transactions on Information Theory. 2000;**46**(4):1204-1216. DOI: 10.1109/18.850663

[16] Wang M, Li B. Network coding in live peer-to-peer streaming. IEEE Transactions on Multimedia. 2007;**9**(8):1554-1567. DOI: 10.1109/TMM.2007.907460

[17] Chen HCH, Hu Y, Lee PPC, Tang Y. NCCloud: A network-coding-based storage system in a cloud-of-clouds. IEEE Transactions on Computers. 2014;**63**(1):31-44. DOI: 10.1109/TC.2013.167

[18] Ho et al. A random linear network coding approach to multicast. IEEE Transactions on Information Theory. 2006;**52**(10):4413-4430. DOI: 10.1109/TIT.2006.881746

[19] Heide J, Pedersen MV, Fitzek FHP, Medard M. On code parameters and coding vector representation for practical RLNC. In: IEEE International Conference on Communications (ICC); 5-9 June 2011. IEEE; 2011. pp. 1-5

[20] Heide J, Pedersen MV, Fitzek FH, Médard M. A perpetual code for network coding. In: 79th Vehicular Technology Conference (VTC Spring); 18-21 May 2014. IEEE; 2015. pp. 1-6

[21] Lucani DE, Pedersen MV, Ruano D, Sørensen CW, Fitzek FH, Heide J, et al. Fulcrum network codes: A code for fluid allocation of complexity. 2014; arXiv:14046620

[22] Shrader B, and Jones NM. Systematic wireless network coding. In: Military Communications Conference (MILCOM); 18-21 October 2009. IEEE; 2010. pp. 1-7

[23] Tournoux PU, Lochin E, Lacan J, Bouabdallah A, Roca V. On-the-fly erasure coding for real-time video applications. IEEE Transactions on Multimedia. 2011;**13**(4):797-812. DOI: 10.1109/TMM.2011.2126564

[24] Li B, Bi S, Zhang R, Jiang Y, Li Q. Random network coding based on adaptive sliding window in wireless multicast networks. In: 83rd Vehicular Technology Conference (VTC Spring); 15-18 May 2016. IEEE; 2016. pp. 1-5

[25] Feizi S, Lucani DE, Sørensen CW, Makhdoumi A, Médard M. Tunable sparse network coding for multicast networks. In: International Symposium on Network Coding (NetCod); 27-28 June 2014. IEEE; 2014. pp. 1-6

[26] Ahmed S, Kanhere SS. VANETCODE: Network coding to enhance cooperative downloading in vehicular ad-hoc networks. In: Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing (IWCMC); 3 July 2006. ACM; 2006. pp. 527-532

[27] Vieira F, Lucani DE, Alagha N. Codes and balances: Multibeam satellite load balancing with coded packets. In: IEEE International Conference on Communications (ICC); 10-15 June 2012. Ottawa: IEEE; 2012. pp. 3316-3321

[28] Arrobo GE, Gitlin RD. Improving the reliability of wireless body area networks. In: Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'11); 30 August–3 September 2011. Boston: IEEE; 2011. pp. 2192-2195

[29] Wu H, Chen M, Guan X. A network coding based routing protocol for underwater sensor networks. Sensors. 2012;**12**(4):4559-4577. DOI: 10.3390/s120404559

[30] Bilbao J, Crespo PM, Armendariz I, Médard M. Network coding in the link layer for reliable narrowband powerline communications. IEEE Journal on Selected Areas in Communications. 2016;**34**(7):1965-1977. DOI: 10.1109/JSAC.2016.2566058

[31] Douik A, Sorour S, Al-Naffouri TY, Yang HC, Alouini MS. Delay reduction in multi-hop device-to-device communication using network coding. In: International Symposium on Network Coding (NetCod); 22-24 June 2015. IEEE; 2015. pp. 6-10

[32] Zhan C, Xu Y. Broadcast scheduling based on network coding in time critical wireless networks. In: IEEE International Symposium on Network Coding (NetCod); 9-11 June 2010. IEEE; 2010. pp. 1-6

[33] Lun DS, Médard M, Koetter R, Effros M. On coding for reliable communication over packet networks. Physical Communication. 2008;**1**(1):3-20. DOI: 10.1016/j.phycom.2008.01.006

[34] Li S, Maddah-Ali MA, Avestimehr AS. Coding for distributed fog computing. IEEE Communications Magazine. 2017;**55**(4):34-40. DOI: 10.1109/MCOM.2017.1600894

[35] Marques B, Machado I, Sena A, Castro MC. A communication protocol for fog computing based on network coding applied to wireless sensors. In: International Symposium on Computer Architecture and High Performance Computing Workshops (SBAC-PADW). IEEE; 2017. pp. 109-114

[36] Sipos M, Fitzek FHP, Lucani D, Pedersen M. Dynamic allocation and efficient distribution of data among multiple clouds using network coding. In: 3rd International Conference on Cloud Networking (CloudNet); 8-10 October 2014. IEEE; 2014. pp. 90-95

[37] Cabrera Guerrero J, Lucani D, Fitzek F. On network coded distributed storage: How to repair in a fog of unreliable peers. In: International Symposium on Wireless Communication Systems (ISWCS); 20-23 September 2016. IEEE; 2016. pp. 188-193

[38] Dimakis AG et al. Network coding for distributed storage systems. Transactions on Information Theory. IEEE. 2010;**56**(9):4539-4551. DOI: 10.1109/CloudNet.2014.6968974

[39] Sipos M, Heide J, Lucani D, Pedersen M, Fitzek F, Charaf H. Adaptive network coded clouds: High speed downloads and cost-effective version control. IEEE Transactions on Cloud Computing. 2015;**99**:1-1. DOI: 10.1109/TCC.2015.2481433

[40] Oliveira PF, Lima L, Vinhoza TTV, Barros J, Medard M. Coding for trusted storage in untrusted networks. IEEE Transactions on Information Forensics and Security. 2012;**7**(6):1890-1899. DOI: 10.1109/TIFS.2012.2217331

[41] Ostovari P, Wu J. Fault-tolerant and secure distributed data storage using random linear network coding. In: 14th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt); 9-13 May 2016. IEEE; 2016. pp. 1-8