Copyright Notice

Institute of Communication Networks and Computer Engineering
University of Stuttgart
Pfaffenwaldring 47, D-70569 Stuttgart, Germany
Phone: ++49-711-685-68026, Fax: ++49-711-685-67983
Email: mail@ikr.uni-stuttgart.de, http://www.ikr.uni-stuttgart.de

Towards Zero Factory Downtime: Edge Computing and SDN as Enabling Technologies

Keerthana Govindaraj*, Dennis Grewe*, Alexander Artemenko*, Andreas Kirstaedter‡
*Robert Bosch GmbH, Corporate Sector Research and Advance Engineering, Renningen, Germany
{keerthana.govindaraj, dennis.grewe, alexander.artemenko}@de.bosch.com
‡University of Stuttgart, Institute of Communication Networks and Computer Engineering, Stuttgart, Germany
{andreas.kirstaedter}@ikr.uni-stuttgart.de

***Abstract*—Future factory automation systems are expected to process vast amounts of data and orchestrate complex cyber-physical components. Edge Computing (EC) is a promising approach to address the requirements set by upcoming industrial systems. While EC caters to the computation requirements, it requires a solution to perform flexible network management of these computation resources. Software-Defined Networking (SDN) is a promising candidate to tackle such challenges. While most of the related work on EC and SDN focuses on multimedia or automotive applications, this paper presents the relevance of both paradigms for industrial applications. By introducing two most prominent industrial use cases, namely *proactive system surveillance* and *intelligent technical assistance*, this paper discusses the challenges involved and proposes a solution space for realizing these applications using the combination of EC and SDN. Furthermore, it presents future research directions regarding the combination of both paradigms in the context of factory automation.**

***Index Terms*—factory automation, edge computing, latency, flexibility**

I. Introduction

The interconnection of billions of devices as part of the so-called Internet of Things (IoT) is finding its way into the manufacturing industry. *Industry 4.0* describes the current trend towards fully automated and on-demand reconfigurable factory by linking different domains together – e.g., IoT devices, production lines, or other machines – to form a large *heterogeneous system*.

Smart factory, also termed as Industry 4.0, aims at automating the factory on multiple levels to increase the flexibility of the production process and at *zero factory downtime* to achieve maximum utilization of all the resource. Interconnecting various sensors and actuators along with other devices enables new features and functions which were until recently not possible.

In the past years, machines have been equipped with dedicated hardware (thick-clients) to attain a certain level of automation, e.g., autonomous transport systems [1]. While such development makes the systems hard to maintain and inflexible to react to emerging needs, the *cloud computing* paradigm has found its way into the manufacturing industry. It offers centralized resources to perform computationally intensive operations, for example *predictive maintenance* – automated detection of conditions which may lead to malfunctions or production errors even before they occur; *flexible manufacturing* – production of highly personalized *batch-size-one* products. Thus, in this case the machines are no longer equipped with dedicated computing hardware (thin-clients), but just have connectivity units to be able to access the cloud environment [2]. However, the introduction of a central computing entity presents a vulnerability in the system, especially in the case of multiple access that can lead to an increased latency in the data communication in worst case.

In recent years, research groups in academia and industry have been investigating the *Edge Computing* (EC) paradigm to integrate it into factory automation (e.g., [3]). EC is an optimization of the cloud computing paradigm, in which the computational resources, storage, and services are brought to the edge of the network, in order to reduce the response delays experienced by the end devices and the communication bandwidth required to exchange data [4]. However, the network resources require a simple and flexible management to deal with the low latency and reliability requirements in addition to the huge data transmissions involved. Software-Defined Networking (SDN) is a new networking paradigm that decouples network control from the data forwarding hardware. The network intelligence is logically located in software-based controllers (control plane) and the network devices become mere packet forwarding entities (data plane) [5]. Hence, it enables a new level of network management including better control, higher flexibility, and scalability. Integrating SDN mechanisms with EC helps in providing the required computation resources and in satisfying the unique quality of service requirements of the applications.

While recent publications in EC focus on specific problems mainly related to offloading possibilities to optimize response times or energy consumption, there are very few works available concentrating on the orchestration of tasks in a dynamic and distributed EC infrastructure. Being since a long time in theoretical research, SDN has seen many novel approaches for network management, control, fast reconfiguration, healing, network function abstraction, placement, etc. However, there are only few works that take those approaches to the factory floor where they can be highly beneficial. This paper introduces the new architectural perspective of EC in the industry and the aspects of SDN that assist the network management in this novel EC architecture.

978-1-5386-6876-4/18/$31.00 ©2018 IEEE

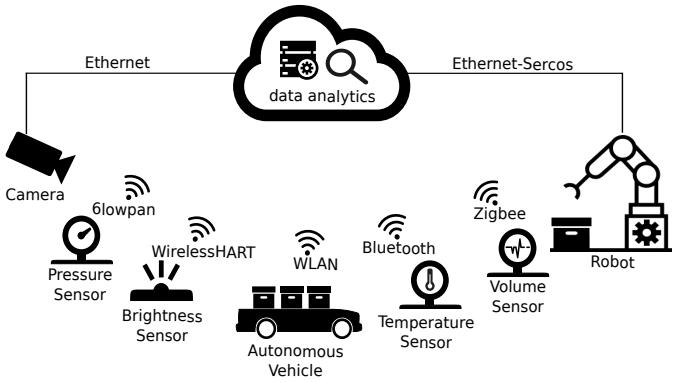

Fig. 1. Example of pro-active system surveillance based on the fusion of data from all factory devices communicating via different communication protocols.

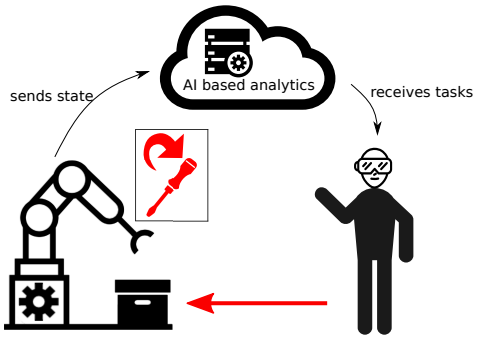

Fig. 2. Example of an intelligent technical assistance scenario in which a human is guided to a malfunctioning machine using AR technologies.

II. Motivation: Zero-factory downtime

A smart factory constitutes multiple components ranging from resource constrained IoT devices up to autonomous machines like programmable logical control (PLC) driven robotic arms or automated guided vehicles (AGV). Such a system is configured and planned to achieve an efficient production environment and to increase the productivity. However, unexpected failures of machines or malfunctioning of components within a production line, regardless of their complexity, may result in outages or a production halt, thus violating production deadlines. Moreover, the safety requirements specified in ISO 10218 [6] must nevertheless be fulfilled to ensure human safety. Furthermore, such outages result in both maintenance cost as well as losses. Hence, it is important to ensure a seamless and continuous functioning of the processes in a factory. This is termed as *zero-factory downtime* condition. The following subsections introduce most prominent use cases within a smart factory and outline their corresponding requirements to achieve the overall goal of *zero-factory downtime*.

A. Pro-active System Surveillance

The *pro-active system surveillance*, illustrated in Fig. 1, describes a cloud based virtual sensor system that keeps track of all activities in the factory presenting digital twins of all devices including their tasks, states, and sensed data. Each of the sensors and machines are equipped with connectivity modules capable of communicating with other devices by choosing from a variety of wired and wireless networking technologies such as cellular (e.g., 3G, LTE), WLAN (e.g., 802.11abgn), Bluetooth, ultra-wide band, Low-Power Wide-Area Network as well as future technologies (e.g., 5G and 5G Device-to-Device communication), thus forming a heterogeneous system of devices. Based on the fusion of all data, an environmental model is computed in a cloud server to provide a detailed overview of the current state of the factory. In case a problem is detected, re-planning and re-configuration of the production line is triggered, including reallocation of resources, in order to avoid production downtimes. This scenario illustrates exemplary requirement of communication bandwidth and computation resources to perform the complex data fusion algorithm and process it in order to guarantee low response times to react to events in real-time. A single electronic control unit production line can generate up to 500 gigabytes of data everyday and can require a computation unit with a capacity of up to 1 tera FLOPS to perform data analytics and fulfill the requirements of *zero-factory downtime*. Apart from that, it involves the requirement of discovery and reallocation of devices and resources, and simultaneously ensuring safety and security aspects.

B. Intelligent Technical Assistance

The *intelligent technical assistance* use case describes a system to assist technicians to perform maintenance operations or to resolve issues of machines in a factory. For example, a surveillance system (as introduced in Section II-A) can trigger the need for such an assistance. As depicted in Fig. 2, a technician needs to be guided to the machine to perform certain operational steps. In this scenario, the technician is wearing glasses supporting Augmented Reality (AR). Information such as operational steps computed by a function in a cloud server are augmented by the glasses.

Furthermore, an enhanced version of the intelligent technical assistance may include contribution from a powerful artificial intelligence (AI) system or an expertise of a person (a remote expert) located far away from the factory. Such a system helps resolve issues faster as well as avoid extra delay and travel expenses for experts. This scenario illustrates an exemplary requirement of network bandwidth to transmit a video frame of 500 megabytes per second and computation resources of 100 giga FLOPS to perform video processing and additionally rendering of the augmentations in 20 milliseconds to provide the user a jitter-free experience [7]. Apart from these, it presents safety and security requirements and the nearly random mobility of the technician makes it all more challenging.

C. Flexible Production Planning

The *flexible production planning* is a use case in which the production line is re-planned and re-structured entirely or partly based on the dynamically changing requirements.

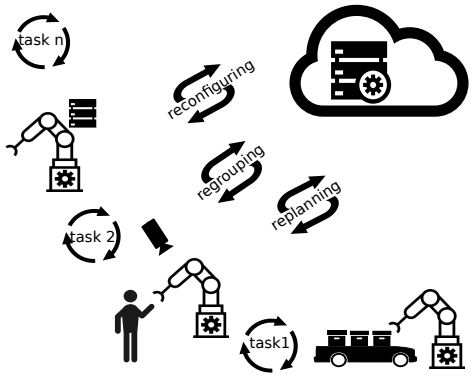

Fig. 3. Example for a flexible production planning. Based on the product to be produced, machines have to be re-allocated in their position as well as to be re-configured.

This includes machines, sensors, devices, and other participants required to manufacture a product. For example, after a surveillance system (as introduced in Section II-A) detects a problem caused by a machine, re-planning and re-configuration is triggered and computed in the cloud backend in order to avoid production downtimes. Fig. 3 illustrates the re-configuration and re-planning of a production line exemplary. This scenario describes the main requirements ranging from real-time monitoring and discovery of available resources, computational resources in order to re-configure and re-plan a production line, as well as safety and security aspects.

Such a model allows to detect and to react to conditions and thus to avoid outages of machines. However, the vital data from several sources needs to be gathered and processed in real-time in order to compute an up-to-date environmental model of the factory. This scenario illustrates the major requirements ranging from network aspects such as bandwidth and latency, computational resources in order to process a huge amount of data generated within a factory (dozens of TB of data to be processed per day, the trend is increasing), as well as safety and security aspects.

III. Challenges

The requirements presented by the use cases (cf. Section II) highlighted challenges that need to be addressed by a factory system in order to achieve the goal of *zero-factory downtime*.

The applications in the factory use cases vary in their requirements, such as the information required to process data or the complexity of the processing steps. For example, the data generated by all the sensors in a proactive surveillance system (cf. Section II-A) need to be available in time, based on certain **quality of service** (QoS) requirements defined by each application. Similarly, the AR data required by a mechanic in the intelligent technical assistance use case (cf. Section II-B) has to be displayed in time to ensure a certain level of **quality of experience** (QoE).

The production of *batch-size-one products* describes one of the major goals in smart factories. It demands **flexibility** in the system in order to react to the constantly changing applications and their requirements. For example, multiple robotic arms need to be re-organized to realize a suitable production line on demand. Similarly, in order to ensure a zero downtime, the factory system needs to react to malfunctioning components and must have the flexibility to take over or assign tasks to other resources, and thus meet the tight production deadlines.

In manufacturing environments in which machines and human workers need to coexist, a system need to ensure safety requirements at all time. Malfunctioning of devices or the entire system can lead to serious injuries. Safety is in turn dependent on the **robustness** of the system. However, the heterogeneity of devices and applications within a smart factory makes it more challenging.

Similarly, in order to cater to the QoS, QoE, and robustness requirements, **availability** of services, data, and resources in time becomes very crucial. Moreover, a flexible manufacturing environment has a varying demand on the availability in terms of location, time, etc.

The heterogeneity in the system (cf. Section II-A) introduces new challenges regarding **deployment & orchestration strategies**. This includes suitable allocation of resources and placement of services in the network to fulfill the production task as well as to dynamically manage and orchestrate these entities without affecting the already running applications.

A flexible factory production system also includes numerous mobile devices, such as AGVs and technicians using AR glasses (cf. Section II) that makes it inevitable to address **mobility**. While the participants are moving through the factory, they need to be able to get and provide access to information corresponding to other participants in the network without interruptions.

A system with a few participants and applications is relatively simple in its design. However, it is hard to achieve **scalability** in a system especially when the number of participants is large. For example, the production of several batch-size-one items involves re-configuration of multiple production lines, machines, sensors, and workers. Therefore, the resources involved in manufacturing the product need to be allocated efficiently, as the deployment infrastructure cannot always be over-provisioned.

In factory automation, the network is expected to host a various number of devices communicating through different access technologies, while forming a heterogeneous system. In order to fulfill the production task, the support of **interoperability** describes an important challenge to offer and share information across other participants in the network.

In order to keep a vital overview of the factory and its tasks, another challenge is described in **handling and processing the large amount of data** created by the entities. Especially in high-speed production lines, sensor data need to be processed and decision mechanisms need to react in a short time. However, the transmission of all data towards a centralized cloud server is not suitable due to time constraints, whereas processing of data directly at the sensors is not possible due to the limited resource capacities.

Last but not least, missing **security & privacy** mechanisms

describe a huge showstopper in the manufacturing industry. This includes the communication between the factory entities as well as the access to data outside of the factory. Manipulating the factory devices or accessing data by any third party may lead to fatal consequences or in a production halt that needs to be prevented.

IV. Edge Computing and SDN Enhanced Factory Automation

In recent years, the rapid development in information technology such as network architectures or communication solutions has been a crucial success factor to tackle the challenges in automating factories. One of the promising paradigms is Edge Computing. EC is a specialization of the cloud computing paradigm and shifts most of the functionality to the edge of the network – as part of Edge Server (ES) – and thus closer to the consumers. As a result, EC promises to improve the network performance by providing computational resources at the edge of the network, and thus reducing the load in the core network. Moreover, the resource constrained devices which require fast responses mainly benefit by offloading their task to a powerful ES nearby. For example, the sensor devices introduced in Section II-A, which need to perform complex data processing in time, are able to offload their task to ES instead of sending all the data towards centralized computing centers [8]. The AR use case presented in Section II-B can improve the performance by offloading computation intensive augmentation tasks to an ES instead of processing on its own hardware [9].

Though on-premises computing can overcome security and privacy threats in factory scenarios in addition to providing desired computation resources to the applications, it does not offer the benefits of cloud computing services. Moreover, on-premises computing also requires programming all the dependencies required by the applications. The EC brings these benefits of cloud computing closer to the devices. Furthermore, the deployment of multiple ESs in the vicinity increases the availability of services and decreases the risk of single-point of failure. Another ES in the vicinity can react and take over tasks or assist an ES in resolving a task cooperatively, if there is any overload or problem on a certain ES. The introduction of EC in factory automation offers operators a new level of flexibility.

Since sensitive data are transferred only between in-factory components and remain within ESs deployed locally, the tasks offloaded to an ES instead of a centralized cloud are naturally prevented from threats such as Denial-of-Service or Spoofing attacks. Moreover, the EC paradigm offers customized solutions for *privacy* and *security* issues corresponding to the applications.

While EC is a promising candidate to overcome many challenges in factory automation (e.g., *data processing*, *availability of data and services*, *short response/reaction times*, etc.), the deployment of multiple ESs becomes complicated when the system needs to be dynamic and flexible. For example, tasks are varying in their complexity (e.g., nodes, machines involved) and their requirements. This becomes even more challenging when large number of devices are mobile. SDN is a promising candidate to complement the EC paradigm. It simplifies the management of network components and resources by decoupling control and data plane. Especially in wireless and mobile scenarios, SDN is able to make networks more controllable and programmable, update routing tables according to the often predictable mobility pattern of the nodes and thus selecting the most appropriate paths from or to the end users (e.g., [10]).

Besides the advantages described by adopting EC and SDN in factory automation, the solution space can be exploited by enhancing ES instances with virtualization and caching capabilities.

A. Virtualized Services at the Edge

The heterogeneous characteristics of future factory systems result in a various number of applications and their requirements deployed on ESs. Virtualization technologies such as virtual machines and Linux containers [11] allow the computation resources and storage on an ES to be managed and allocated as per the requirements of the devices thus making the system *flexible* and *scalable*. For example, in case of a re-configuration of the production line (cf. Section II-A), tasks need to be moved to a different ES, depending on the application load or the mobility of the nodes. Virtualization favors the tasks to be easily stopped and started multiple times or even migrated across the network, and thus facilitates *mobility*.

B. Data Caching at the Edge

Caching strategies have shown a positive gain in reducing the time required for accessing information in the network. In EC, ESs can be used as intermediate nodes to store relevant data and services corresponding to the devices. Furthermore, tasks can be grouped together based on their data characteristics (time-sensitive data, popular data, etc.) that can be used by caching strategies to store data on suitable ESs (cf. [12]). As a result, caching strategies ensure the *availability* of data closer to the consumers and help satisfy the *quality of service* requirements.

V. Research Directions

Introduction of Edge Computing and Software-Defined Networking in factory automation is a promising solution and a door opener for upcoming factory use cases. However, it brings in new challenges and research directions. This section introduces the open challenges, provides an overview of the related work and outlines the open research directions.

A. Virtualization

While virtualization technologies offer flexibility to a system (cf. Section IV), they have some drawbacks such as adding response delay due to the introduction of extra abstraction layers. With the advent of EC, various light-weight virtualization approaches such as serverless computing technologies (e.g.,

Amazon Lambda [13]) or unikernels [14] are gaining importance as they can provide lower response times. This is due to their smaller footprints and faster boot-up times. However, being designed for stateless applications, these technologies present a serious limitation to the applications that rely on their state to resume on another ES. Furthermore, these applications rely on complex data processing (e.g., visual inspection, fusion of sensor data, etc.) and require short response times. Currently, a real-time image processing of video frames is difficult to achieve even with the most powerful CPUs [15]. While GPUs have shown performance improvements in terms of execution time, GPU virtualization is still in its early stage (cf. [16], [17]) and remains an open research direction.

B. Deployment strategies

Offloading is an integral part of EC and mainly focuses on improving the device performance in terms of attaining short response times and reducing energy consumption. There exists numerous works that deal with different strategies such as partial offloading, using the cloud as a fallback option as well as splitting the task amongst multiple ESs [8]. However, the data from multiple devices communicating via different protocols need to be sent to an ES, aggregated, and analyzed to produce the output required by an actuator on the consumer end and sent back within a required time window (cf. Section II-A). The problem here is not limited to offloading, rather extends to selecting an ES that is suitable for multiple devices simultaneously. Current research activities present several strategies for an optimized placement of an operating environment to maximize the utilization of resources [18]. However, in our use cases, the tasks and their requirements constantly change and thus another serving ES needs to be selected within a short time based on the current requirements and system state. This challenge is new and very specific to the system that requires coordinated functioning of multiple participants with the help of EC and thus represents an open research direction for now.

C. Mobility support

Many of the devices described in Section II are mobile in nature. The routes of these devices constantly change depending on the tasks to be accomplished. From a networking perspective, mobility introduces connectivity challenges such as ensuring acceptable handover times required to guarantee the functional operation of an offloaded application. SDN provides a solution with respect to finding an alternative route to the ES in order to maintain the QoS requirement [10]. Due to mobility, the distance between a client device and a serving ES may increase over the time. As a consequence, the required QoS becomes difficult to guarantee. Migration of the offloaded application from the serving ES to a closer one, in terms of the communication distance, can solve the problem. This process in which the service follows the mobile device is known as live migration and it is gaining importance in EC [19]. However, the most crucial aspect remains the zero service downtime requirement which is a prerequisite for a *zero factory downtime*. To the best of our knowledge, there is no solution for this yet. Therefore, the zero downtime service migration represents an important research direction for the factory automation.

D. Resource orchestration

Based on an application characterization, an appropriate amount of CPU, RAM, and storage resources in addition to the network resources must be reserved for each application. Barbarossa et al. in [20] proposed a joint radio and computation resource optimization scheme in a single base station to minimize the transmission power of mobile devices. Kusic et al. [21] developed a dynamic resource provisioning framework for a multi-objective optimization in a virtualized computing environment. However, as described in Section II-A, the factory tasks map changes very frequently thus varying the application requirements. The resources must be intelligently reallocated to continue to serve all applications already running and at the same time meet the QoS requirements of the new applications as well as satisfy the priority demands to ensure safety and reliability of the production process. To support zero factory downtime, the re-planning and reallocation algorithms must have fast convergence times which still remains an open research direction.

E. Context-based caching

As described in Section IV, caching reduces the time required to access data. But, what needs to be cached and when plays an important role in increasing the cache hit rate. Authors in [22] propose a scheme to pro-actively prefetch the content from the backbone to store it on the ESs. The data generated by the sensors in one location may be required in another location to make an appropriated decision (cf. Section II-A). Thus, context-aware caching strategies offer the possibility to store corresponding data within caches in the vicinity of devices and therefore provide quick access. Towards such schemes, the authors in [12] propose a context-aware grouping scheme to access relevant data to increase the cache hits. However, other entities such as applications in the network should also be taken into account. For example, when a mobile device moves and intends to connect to another edge server, the base application needs to be pre-fetched in time to enable a smooth and quick transition.

F. Reliable network management

Reliability and in turn safety is of at most importance in industrial scenarios as failures can have fatal consequences. SDN is considered to provide a flexible networking solution and thus can be programmed to guarantee reliability by performing failure detections and recoveries in the SDN data and control planes. However, the processing time involved in failure detection and recovery must be below the acceptable levels of industry safety requirements. Song et al. in [23] summarize different reliability solutions available on data and control planes and propose a strategy to improve the reliability of a control path between the controllers and switches.

However, mobility increases the complexity in achieving a reliable network connectivity between the devices and ESs in manufacturing environments and remains an open research direction.

G. Security & Privacy

In industrial scenarios, EC eliminates the major security and privacy threats involved in transmitting data outside the factory premises. However, EC in factory automation introduce new challenges such as: *trust* - end device as well as ESs, need to validate their genuinity using a trust model located centrally before starting a service; *authentication* - since multiple devices belonging to different players need to cooperate in a factory environment, the system requires a standard and energy efficient authentication scheme suitable for resource constrained mobile devices; *data encryption* - the devices need to encrypt their data before transmission in order to avoid data modification or spying from an infected device within a factory. Mukherjee et al. [24] explain the shortcoming of schemes used in cloud computing due to the resource constraint of end devices as well as a lack of centralized entity for management. Very few schemes related to EC listed in [24] are applicable in factory environment due to its unique requirements. Moreover, they are still in its nascent stage presenting a large scope for further research.

While the above list of presented challenges and research directions is extensive, it is not complete. With introduction of new wireless and wired network communication technologies (e.g., time-sensitive networking, network slicing, etc.), new challenges may arise.

VI. Conclusion

This paper has introduced a vision of *zero-factory downtime* with two most prominent use cases, namely *pro-active system surveillance* and *intelligent technical assistance* along with the challenges involved in realizing it. The solution space introduced in Section IV has described Edge Computing and Software-Defined Networking as promising technologies addressing the factory automation challenges such as availability, reliability, quality of service, security, privacy, and interoperability. This paper has shown that there are still open research directions such as context-based selection of information and server instances, dynamic allocation and orchestration of network resource, and massive data processing while participants are mobile. Especially in the manufacturing industry, safety and security mechanisms define important roles to provide maximum protection for humans in the factory and the surrounding machines. Future work needs to address the open research directions by setting up the use case scenarios in both simulation and real world proof of concept prototypes. Such environments will form the basis to further investigate and elaborate novel mechanisms with respect to the introduced directions.

References

[1] M. Gath, O. Herzog, and S. Edelkamp, "Autonomous and flexible multiagent systems enhance transport logistics," in *International Conference Expo on Emerging Technologies for a Smarter World*, Oct 2014.

[2] G. Mohanarajah, D. Hunziker, R. D'Andrea, and M. Waibel, "Rapyuta: A cloud robotics platform," *IEEE Transactions on Automation Science and Engineering*, April 2015.

[3] J. Rambach, A. Pagani, M. Schneider, O. Artemenko, and D. Stricker, "6dof object tracking based on 3d scans for augmented reality remote live support," *Computers*, Jan 2018.

[4] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing*, Oct 2009.

[5] B. A. A. Nunes, M. Mendonca, X. N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Communications Surveys Tutorials*, 2014.

[6] ISO, "ISO 10218 - robots and robotic devices - safety requirements for industrial robots," in *ISO/TC 299 Robotics*, July 2011.

[7] S. Melnyk, A. Tesfay, H. Schotten, J. Rambach, D. Stricker, M. Petri, M. Ehrig, T. Augustin, N. Franchi, G. Fettweis, O. Artemenko, M. Schneider, and M. Aleksy, "Next generation industrial radio lan for tactile and safety application," in *ITG-Fachtagung: Mobilkommunikation Technologien und Anwendungen, Osnabrck, Germany*, May 2017.

[8] M. A. Khan, "A survey of computation offloading strategies for performance improvement of applications running on mobile devices," *Journal of Network and Computer Applications*, vol. 56, 2015.

[9] M. Schneider, J. Rambach, and D. Stricker, "Augmented reality based on edge computing using the example of remote live support," in *IEEE International Conference on Industrial Technology (ICIT)*, March 2017.

[10] M. Yang, D. Z. L. Li, Y.and Jin, X. Wu, and A. V. Vasilakos, "Software-defined and virtualized future mobile and wireless networks: A survey," *Mobile Networks and Applications*, 2015.

[11] Canonical Ltd., "Linux containers project page," 2018. [Online]. Available: https://linuxcontainers.org/

[12] N. Mohan, P. Zhou, K. Govindaraj, and J. Kangasharju, "Managing data in computational edge clouds," in *Proceedings of the Workshop on Mobile Edge Communications (MECOMM)*, 2017.

[13] Amazon Web Services, Inc., "Amazon lambda project page - run code, not servers - serverless computing," 2018. [Online]. Available: https://aws.amazon.com/lambda/

[14] R. Morabito, V. Cozzolino, A. Y. Ding, N. Beijar, and J. Ott, "Consolidate iot edge computing with lightweight virtualization," *IEEE Network*, Jan 2018.

[15] I. K. Park, N. Singhal, M. H. Lee, S. Cho, and C. Kim, "Design and performance evaluation of image processing algorithms on gpus," *IEEE Transactions on Parallel and Distributed Systems*, Jan 2011.

[16] C. Y. Yeh, C. Y. Kao, W. S. Hung, C. C. Lin, P. L.iu, J. J. Wu, and K. C. Liu, "Gpu virtualization support in cloud system," in *Grid and Pervasive Computing*, 2013.

[17] S. Iserte, F. J. Clemente-Castelló, A. Castelló, R. Mayo, and E. S. Quintana-Ortí, "Enabling gpu virtualization in cloud environments," in *Proceedings of the International Conference on Cloud Computing and Services Science*, ser. CLOSER 2016, 2016.

[18] L. Zhao and J. Liu, "Optimal placement of virtual machines for supporting multiple applications in mobile edge networks," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, 2018.

[19] A. Machen, S. Wang, K. K. Leung, B. J. Ko, and T. Salonidis, "Live service migration in mobile edge clouds," *IEEE Wireless Communications*, February 2018.

[20] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, "Joint allocation of computation and communication resources in multiuser mobile cloud computing," in *IEEE Workshop on Signal Processing Advances in Wireless Communications*, June 2013.

[21] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," in *International Conference on Autonomic Computing*, June 2008.

[22] T. Hou, G. Feng, S. Qin, and W. Jiang, "Proactive content caching by exploiting transfer learning for mobile edge computing," in *IEEE Global Communications Conference (GLOBECOM)*, Dec 2017.

[23] S. Song, H. Park, B. Y. Choi, T. Choi, and H. Zhu, "Control path management framework for enhancing software-defined network (sdn) reliability," *IEEE Transactions on Network and Service Management*, June 2017.

[24] M. Mukherjee, R. Matam, L. Shu, L. Maglaras, M. A. Ferrag, N. Choudhury, and V. Kumar, "Security and privacy in fog computing: Challenges," *IEEE Access*, 2017.