Wireless Cyber-Physical Systems Performance Evaluation through a Graph Database Approach

Mohamed Kashef*, Yongkang Liu, Karl Montgomery, and Richard Candell
National Institute of Standards and Technology (NIST)
Gaithersburg, Maryland 20899
Email: mohamed.kashef, yongkang.liu, karl.montgomery, richard.candell@nist.gov

ABSTRACT

Despite the huge efforts to deploy wireless communications technologies in smart manufacturing scenarios, some manufacturing sectors are still slow to massive adoption. This slowness of widespread adoption of wireless technologies in cyber-physical systems (CPS) is partly due to not fully understanding the detailed impact of wireless deployment on the physical processes especially in the cases that require low latency and high reliability communications. In this paper, we introduce an approach to integrate wireless network traffic data and physical processes data in order to evaluate the impact of wireless communications on the performance of a manufacturing factory work-cell. The proposed approach is introduced through the discussion of an engineering use case. A testbed that emulates a robotic manufacturing factory work-cell is constructed using two collaborative-grade robot arms, machine emulators, and wireless communication devices. All network traffic data is collected and physical process data, including the robots and machines states and various supervisory control commands, is also collected and synchronized to the network data. The data is then integrated where redundant data is removed and correlated activities are connected in a graph database. A data model is proposed, developed, and elaborated; the database is then populated with events from the testbed, and the resulting graph is presented. Query commands are then presented as a means to examine and analyze network performance and relationships within the components of the network. Moreover, we detail the way by which this approach is used to study the impact of wireless communications on the physical processes and illustrate the impact of various wireless network parameters on the performance of the emulated manufacturing work-cell. This approach can be deployed as a building block for various descriptive and predictive wireless analysis tools for CPS.

1 Introduction

Smart manufacturing and modern factories require interactions and collaborations between various distributed equipment, products, and logistics to accomplish unprecedented levels of productivity and operational efficiency. Wireless communication is among the enabling technologies to achieve this vision [1]. Due to an increased demand for ease of installation, reduced costs of deployment and maintenance, and flexibility, wired networks are being replaced with wireless networks. This presents a real challenge for the networks and control systems. Compared with wired connections, wireless links have their unique advantages in connecting field sensors and actuators with reduced cabling cost and natural support of mobility [2]; however, most current communication systems lack the latency and reliability support [3] mandated by factory requirements [4,5]. More specifically, the adoption of wireless in cyber-physical systems (CPS) requires innovative methods and approaches to quantify the impact of wireless technologies in terms of production efficiency and measure the cost of wireless link failures on performance [6].

In modern CPS, a large volume of heterogeneous data is generated and transferred within a variety of equipment, sensors, controllers and computing platforms. Data analytics for CPS play a critical role in improving factory operation and product quality, reducing machine downtime, and enhancing manufacturing efficiency [7, 8]. Generally, data analytics performs the task of extracting information, analyzing performance, and predicting production forecast. In [7], the life cycle of data analytics in CPS includes data acquisition, prepossessing, storage, and analysis. Data acquisition includes adding various points of data collection and the data transfer to the storage and processing units. Afterwards, typical prepossessing

*Address all correspondence to this author.

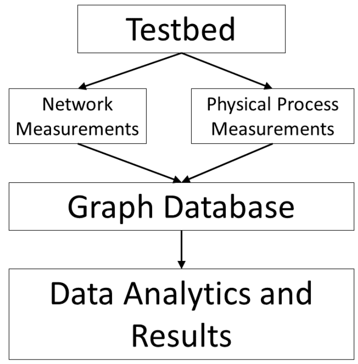


Fig. 1: Graph database approach for wireless network performance evaluation

techniques for cleaning, integration, and compression are deployed because of the big volume, redundancy, and heterogeneity of the raw data. Finally, data analysis is performed for data modelling and visualization.

Database management systems play the role of organizing data efficiently and effectively. Two types of databases exist, namely, relational and non-relational. The non-relational databases are also known as NoSQL and are used often to store semi-structure and unstructured data. NoSQL databases include various data models such as key-value stores, columnar databases, document stores, and graph databases (GDBs) [9]. Therefore, non-relational databases are more suitable for managing heterogeneous data in industrial settings [10]. A GDB is a NoSQL database that uses nodes, edges, and properties to store and present data. The GDB does not enforce a particular schema by having a data model that allows any node type to have different set of properties, and similarly, the relationships. Specifically, not each property or relationship corresponding to a specific node type is required for each instance of the GDB implementation. The relationships within a GDB can be efficiently queried because they are persistently stored within the database. In a GDB, queries can be made based on relationships. This, in particular, presents an advantage when storing information regarding systems with correlations that are apparent but difficult to visualize or quantify.

In order to improve the deployment of industrial wireless in CPS, effective and easy-to-use strategies have to be offered for the test and evaluation of such systems in a way that correlates network performance with operational performance. The needed data for this purpose is collected from various CPS activities and networks, and is generally found to be of large amounts, heterogeneous, and correlated. As a result, we present a novel method to simultaneously capture network and operational event information using a GDB. The use of a GDB allows for more intuitive inferences to be made through the stored relationships and graph theoretic models [11].

The main benefits of deploying a GDB approach are as follows: i) it allows to have direct relationships between the corresponding messages, transactions and physical actions which allows faster database querying compared to the relational databases (i.e., accessing a vertex of GDB and its neighbors can occur with a simple memory lookup through a pointer attached to each vertex), and ii) it offers a clear explanation of the impact of wireless communications individual events on the corresponding individual physical action not based on statistical criteria.

In [12], we have presented the wired baseline design of the National Institute of Standard and Technologies (NIST) collaborative two-robot machine-tending work-cell testbed. In this paper, we extend the usage of this testbed through deploying a wireless network for the robotic supervisory control traffic. We then propose the complete data analytics approach shown in Fig. 1 to evaluate the performance of the wireless network and its impact on physical activities of the work-cell. In [13], we have presented a GDB implementation to measure wireless performance of the network without yet correlating that to the physical actions of the work-cell. In summary, the contributions of this work are as follows:

1. A wirelessly connected collaborative two-robot machine-tending work-cell testbed is described and the method to collect the network traffic and physical actions data of the testbed is elaborated.
2. A GDB application for the capture and analysis of the cyberphysical system performance of a manufacturing workcell utilizing the Neo4j database platform is explained in detail where a proposed data model is described to correlate both network and operational events.
3. Numerical results are obtained to validate the proposed approach and to describe the impact of wireless communications on this specific use case.

This paper discusses multiple components through the discussion of an engineering use case. Some of these components are generic and can be further used while the rest can be more specific for this use case. These components include the general framework using a GDB, the data model design and implementation, the testbed implementation and data collection, and data analysis. The introduction of a GDB approach for analyzing industrial CPS, which achieves the one-to-one mapping between the network activities and the corresponding physical actions, is needed to understand industrial wireless network impacts on the physical activities. To the best of our knowledge, this framework is the first to achieve this mapping on an experimental study. Hence, the idea of using GDB is generic and can be used in many industrial scenarios. The data model introduced can be widely adopted in multiple use cases that deploy industrial wireless networks for supervisory control of robots and machines including machine tending and pick-and-place applications. This model and the implemented scripts that build the GDB using the collected network data, robots state feedback, and the machine status can be suitable for these supervisory applications. On the other hand, the exact use case implementation and the ensuing data analysis are more specific to the engineering use case where we try to introduce the proposed framework and the date while being implemented in an experimental study. Some ideas for data collection and synchronization can be generic but they are already examined in the literature as well.

Our paper is organized as follows: in Section 2, we present a review of the related literature where we start by the state-of-the-art of industrial wireless in Section 2.1, followed by discussing the importance of industrial data analytics in Section 2.2. We discuss the graph database applications and advantages in Section 2.3 followed by examples of its deployment in industrial data analytics in Section 2.4. In Section 3, the use case and the testbed setup are briefly presented. In Section 4, we start by introducing the justifications for selecting the GDB approach and the use of the Neo4j tool. We then present the GDB related architecture in Section 4.1 and various data processing stages in Section 4.2. We then present the results of our analysis in Section 5, followed by conclusions and future direction in Section 6.

2 Related Work

2.1 State-of-the-art of Industrial Wireless

Wireless communications in industrial environments enables machine-to-machine (M2M) information exchange to improve production efficiency and safety. Generally, industrial wireless services have various requirements in different use cases. Ahmadzai et al. summarized CPS requirements for M2M systems and called out high connectivity within the factory coupled with autonomy indicating needs for reliable wireless systems [14]. A recent technical report, in [15], identifies specific service requirements in different classes of industrial wireless applications with finely tuned metric thresholds. Pang et al. discussed the possibility of new wireless techniques to support high performance industrial services such as cycle time (1 ms to 10 us), reliability (medium 10^{-6} to high 10^{-9}), and scale (100's to 10,000's) [16]. Industrial wireless system design also faces challenges such as the transmission loss in radio propagation and diverse interference on the factory floor. A series of channel measurement campaigns have shown unique radio channel features in real industrial sites. In [17], the channel performance was analyzed within two different factories showing multipath environment clearly for different antenna and polarization types. Both active and passive channel measurement data was collected in different sites and identified the huge diversity of radio propagation characteristics that vary with the factory layouts and production activities [18]. Besides, the factory contains various sources of interference that may impair wireless links. The authors in [19] enumerate few sources such as motors, frequency converters, voltage regulators, welding equipment, and office computers. Therefore, efficient evaluation methods are needed to verify existing and emerging industrial wireless solutions. The authors in [17,20] discussed the performance of IEEE 802.15.4 radios in real industrial environments with strong multi-path fading and concluded that existing solutions could not overcome the log-normal shadowing, measured in 7 – 12 dB, as they did not employ strong channel coding in their receivers. There is a real need to have a testbed that can evaluate the performance of wireless solutions in support of industrial applications [21].

When industrial status information is collected at the field device or a control command is generated by the controller for remote actuators, the intermediate network has to timely and reliably transmit these data, especially in wireless links. However, connecting the wireless system information flows to the operation of the physical system is not always evidently apparent. Current studies do not unveil hidden dependencies in such complex systems.

In Table 1, we compare this work to the most related works in the literature for industrial wireless system evaluation. The comparison includes the application domain, the test use case, the system setup, the data collection approach, and the evaluation criteria. Our work is the only work that achieves the one-to-one mapping between physical actions to network activities. Furthermore, it introduces hardware systems for both physical and network domains to capture the realistic impacts of hardware on CPS system performance.

2.2 Importance of Industrial Data Analytics

Industrial data analytics play an essential role in achieving the smart factory vision and improving decision-making in various industrial applications. Five main industrial data methodologies are generally studied including highly distributed data ingestion, data repository, large-scale data management, data analytics, and data governance [30]. Industrial data processing offers valuable information about various sections of industrial applications including inefficiencies in industrial processes, costly failures and down-times, and effective maintenance decisions [31, 32]. The industrial data analytics are

Table 1: Feature comparison with existing industrial wireless evaluation platforms

	Application Domain	Use Case	System Setup			Data Collection		Evaluation	
			Physical System	Wireless Network	RF Factor	Physical	Network	Data Process	CPS Metrics
This work	Factory Automation	Robotic Work-cell	HW (PLC, robots, 10 - 125 Hz updates)	HW (WLAN IEEE 802.11b/g/n as presented)	Indoor (Lab), managed WLAN interference radios in 2.4 GHz	Device logging, remote access	Ethernet /WLAN sniffer	Pipeline, scripts, graph database	Control delay, system failure
Aminian 2013 [22]	Process Automation	Dual-Tank level control as presented	SW (by Simulink)	SW (Wireless Mesh with IEEE 802.15.4), HIL (tentative)	SW	Simulator logging	Simulator logging	Visual inspection, scripts	Control, I/O
Jecan 2018 [23]	Process Automation	Industrial Wireless Network	No	HW (WirelessHART plus ISA100.11a)	Indoor (Lab)	No	Network manager	scripts	No
Ding 2015 [24]	Process Automation	Wireless Sensor & Actuation	HW (valve control, 1 Hz updates)	HW (WirelessHART, ISA100.11a)	Indoor (factory), managed Zigbee and WLAN in 2.4 GHz	No	Packet sniffer	scripts	No
Liu, Q 2018 [25]	Process Automation	Wireless medical telemetry	HW (operation room surgical monitoring)	HW (WLAN IEEE 802.11b/g/n)	Indoor (multiple rooms separated by walls)	Health database (MySQL)	WLAN signal analyzer, JPerf	scripts	Database requests
Fink 2013 [26]	Robotics	Robot teams	HW (mobile AGVs)	HW (IEEE 802.15.4)	Indoor (two office/lab buildings)	Location reported at 5 Hz	Signal strength reported at 5 Hz	scripts	No
Liang 2019 [27]	Factory Automation	AGV, safety	HW (mobile AGVs)	HW (WIA-FA)	Indoor (industrial sites)	Device logging, field measures	Signal analyzer, spectrum analyzer, network analyzer	scripts	Motion distance error
Candell 2015 [28]	Process Automation	Chemical process control	HIL (process simulator, PLC, sensors)	HW (IEEE 802.15.4-TDMA)	Indoor (Lab)	Simulator logging	No	scripts	Process control
Liu, Y. 2016 [29]	Process Automation	Chemical process control	SW (process simulator)	SW (IEEE 802.15.4-TDMA)	SW (PER-SNR curves)	Simulator logging	Simulator logging	scripts	Process control safety

Notes: HW: hardware testbed; SW: software simulation; HIL: hardware-in-the-loop simulation.
"scripts" stands for the data processing that uses specific code/program to treat experiment data in the performance evaluation.

generally deployed for improving factory operations through improving machinery utilization and predicting production demands, improving product quality by analyzing market demands and reducing defective products, and enhancing supply chain efficiency by analyzing risk factors and making accurate logistic plans and schedules [7]. The methods of industrial data analytics can be split into different categories such as descriptive, diagnostic, predictive, and prescriptive analytics [7]. Descriptive and diagnostic analytics are responsible for analyzing historic data and the causes of events and behaviors. Predictive and prescriptive analytics require more processing power, anticipate the trends of data, and deploy the historical data in making decisions to achieve production goals. Examples of industrial data analytics frameworks can be found in [33–36]. In [34], a platform for performing industrial big data analysis is presented where the performance requirements are introduced to achieve a cost-effective operation. In [33], a manufacturing big data solution for active preventive maintenance in manufacturing environments is proposed. Various other frameworks for industrial data analysis can be found in [35, 36], where the importance of using data analysis in decision making is emphasized.

2.3 Advantages of the GDB Approach

Multiple surveys about GDBs have been presented to describe the associated models, tools, and their features such as [11,37,38]. The advantages of deploying GDB include having a more natural approach of data modelling and keeping data properties connected to nodes and relationships [11]. Moreover, GDBs offer graphical and visualization interfaces to data and are able to keep the time-related information of events through various graph paths [38]. Also, an extended list of applications and implementations of GDBs is presented in [39] to show their use on enterprise data, social networks, and determining security and access rights. It was found that GDBs provide the much needed structure for storing data and incorporating a dynamic data model. In general, the use cases, in which GDBs perfectly improve the data management, include path finding with weighted and time-related path properties, mapping dependencies of various system components to capture potential weak points, and communications between various networked elements [38]. On the other hand, query languages are used to extract data, including traversing the database, comparing node properties, and subgraph matching [37,40]. The performance of different GDB tools and methodologies is analyzed and compared in [41,42]. Various aspects of functionality differentiate the performance of query languages such as subgraph matching, finding nodes connected by paths, comparing and returning paths, aggregation, node creation, and approximate matching and ranking [40].

2.4 GDB for Industrial Data Analysis

Due to their advantages including scalability, efficiency, and flexibility, GDBs are widely adopted in various industry-related applications and use cases such as network operations, fraud detection, and asset and data management [43]. In [44], authors have proposed a new object tracking approach for surveillance applications. The GDB approach is selected to contribute to the scalability of the proposed scheme and support the required connectivity analysis for the object tracking. Moreover, relationships in social networks have been modeled using a GDB for structural information mining and marketing [45]. On the other hand, GDBs are also deployed in business solutions for scenarios with multiple large data sources which require distributed processing in decision making for various problems such as fraud detection, trend prediction, and product recommendation [46].

In [8], it was shown that GDB and Neo4j can be used in network security-related applications because the network characteristics are in compliance of GDB concept of nodes and relationships. It was stated that Neo4j is selected to efficiently query and analyze the data where the query results can be visualized directly. Moreover, in [47], Neo4j was also used to build a model for a power grid network analysis where experimental results compared the performance to an example relational database system. In [48], an efficient and secure information retrieval framework for content centric networks used the Neo4j graph database to improve the efficiency of storing and processing large-scale data. In [49], Neo4j was used to analyze network vulnerability to guarantee the accuracy of the attack graph generation and analysis process.

Moreover, the use of GDB, and more specifically Neo4j, in analyzing time stamped data logs has been demonstrated in [50, 51]. In [50], a GDB approach has been used for analyzing network log files from different sources in real-time. The data from different network layers has been exported and combined is a single graph in order to detect anomalies in network performance. In [51], the business event logs monitoring is demonstrated where a loan application was exported to a GDB in order to facilitate the business decision making process based on the available data. In our work, we introduce the application of a GDB approach for analyzing industrial CPS, which achieves the one-to-one mapping between the network activities and the corresponding physical actions.

3 Case Study: Robotic Machine-tending

The NIST industrial wireless testbed and measurement system provides a reusable framework that can be utilized to evaluate numerous wireless technologies. In this section, a two-robot machine-tending work-cell case study is presented to introduce the proposed procedures in the data workflow. This section reviews the design of the work-cell, measurement system, and equipment used that serve in a typical evaluation case. In this section, we detail the physical system implementation of the discussed engineering use case where our evaluation approach is applied

3.1 Work-cell Implementation and Measurements

In the testbed, the two-robot pick-and-place task is performed that can be tailored to other use-cases and wireless technologies. A work-cell was constructed to perform a dual robot pick-and-place task that is controlled by a supervisor programmable logic controller (PLC). The robots have six degrees-of-freedom and utilize Modbus/TCP communication messages to receive and execute assigned tasks from the supervisor PLC. There are also four emulated computers numerically

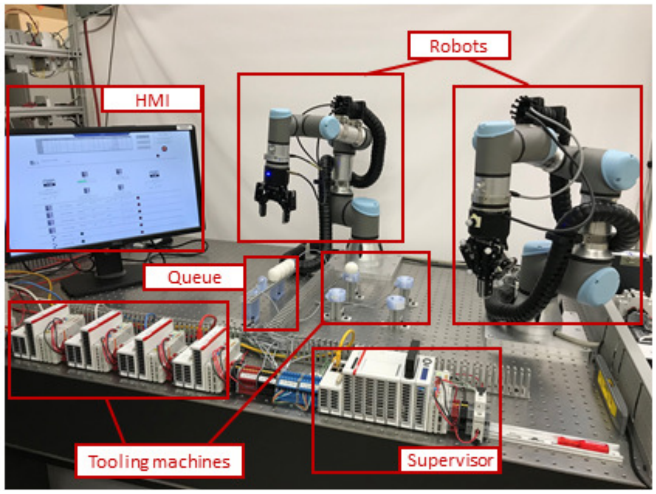


Fig. 2: Collaborative work cell testbed

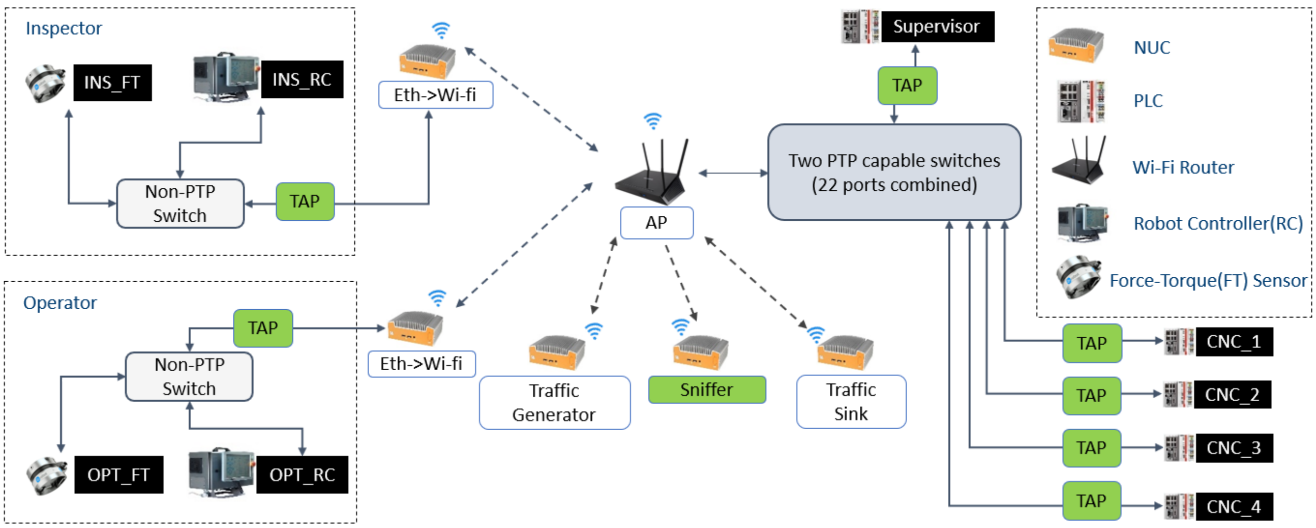


Fig. 3: Network diagram of the testbed. Solid lines represent Ethernet based communications and dashed lines represent wireless based communications.

controlled (CNC) machines that detect the physical state of the testbed with proximity sensors. To detect whether the CNC machine is free or occupied, proximity sensors are positioned at the inner radius of the cups that hold the parts.

Fig. 2 depicts the robots, supervisor PLC, and CNC machines in the testbed in relation to the physical queue ramp that supplies parts to the CNC machines. A human-machine interface (HMI) is used to add tasks to the job queue. To perform the pick-and-place task, the operator moves parts from the queue ramp to each of the CNC machines, and back to the queue ramp. Between each Operator movement, the Inspector performs a force seeking detection to determine if the part is in the correct location before the Operator proceeds. More detail regarding the workflow of the testbed can be found in [12].

All communicating devices in the testbed were originally designed to use Ethernet TCP/IP for communications. To enable wireless in the testbed, Ethernet-WiFi bridges are utilized for wireless communications through a common access point (AP). It is also possible to use other industrial wireless technologies such as Ethernet-Zigbee converters. To establish the Ethernet-WiFi bridges, small form factor computers, called NUCs, are used as they allow for flexibility in the work-cell. For the experiment in this paper, the NUCs used a single antenna for wireless communications using IEEE 802.11n (Wi-Fi) in the 2.4 GHz ISM band. Each NUC is configured to be one of the following forms: a bridge, a wireless sniffer, a traffic generator, or a traffic sink. These configurations in the network are shown in Fig. 3.

Three different types of measurements (network, robot, and PLC state data) are collected in the testbed. Network, wired, and wireless traffic data are captured using seven test access point (TAP) devices and a wireless sniffer, shown in Fig. 3 with green labels. A machine running UBUNTU 18.04, not shown, is used to capture the data from all wired network traffic in the testbed. The position data and robot state data are captured from the robot controllers using the real time data exchange

(RTDE) protocol [52]. RTDE data is captured on a Linux data capture workstation. Lastly, the PLC state data is captured locally on the supervisor PLC during each trial of the experiment. These three types of measurements (network, robot, and PLC state data) share a precise time stamp that originates from the synchronization to the grand master time server. We adopted the IEEE 1588 precision time protocol (PTP) to allow synchronized distributed clocks to stamp the accurate time on all measurement devices in the testbed [53].

3.2 Equipment Used

The following pieces of equipment are illustrated in Fig. 3. Intel Core i7-equipped NUCs running UBUNTU 14.04, that enable wireless communications in the testbed, are used; the NUCs communicate through a common AP. The AP is a Netgear AC1900 wireless router capable of IEEE 802.11ac 4x4; however, MIMO is not used for the experiment in this paper. For the wired communications, two Cisco IE 4000 industrial grade Ethernet switches that are PTP compatible for time synchronization are used. The collaborative robots that perform the pick-and-place task are Universal Robots "UR3" CB series. The collaborative robots are equipped with OnRobot HEX-H Force/Torque Sensors, which are used by the Inspector to inspect parts. The supervisor PLC is a Beckhoff CX2020 with an EL6688 PTP module for time synchronization. The CNC simulators are Beckhoff CX9020 PLCs. The seven TAP devices on the testbed are SharkTap Gigabit Network Sniffers. To synchronize the timing of the devices while taking measurements, a Meinberg Lantime M900 grand master time server is used. Lastly, the Operator and Inspector each use a D-link DGS-108 8-port unmanaged Ethernet switch for wired communications between the robot controller and the force-torque sensor. These switches are also used to enable the Linux workstation to collect RTDE data through a wired connection.

4 Application of GDB in the Robotic Machine-Tending

A GDB was built to manage data collected from testbed measurements of both network traffic and physical operations. In this section, we briefly introduce graph components developed for our testbed and the data processing flow that transforms measurement results to graph entities.

In order to justify using the proposed approach, we start by stating and defining the collected data characteristics and the requirements for the deployed database approach in handling the data for the goal of our study as follows:

1. Heterogeneous data - The collected data from industrial wireless communications system is heterogeneous in different aspects as follows
    (a) Different sources: We collect network data at various network nodes in the system. Also, collected data using wireless sniffer describes the wireless physical environment. Data from the supervisor controller is also collected which includes the system states and the supervisory commands. Data from the robots is used to describe the physical actions taken.
    (b) Different formats: The data includes different file formats such as packet capture (PCAP) files, and data that comes from different PLC and robot controllers is stored in the format of comma separated value (CSV) files. Another example is the time stamp format from different devices.
    (c) Different rates: data packets can be both periodic and event driven. Also, the robot state feedback is periodic with a different update rate than the update rate of the PLC state.
2. Entities are interrelated - This is the main requirement and challenge in this work where the goal is to obtain the direct one-to-one connection between physical actions and their corresponding entities including network activities, the physical wireless environment through sniffer reports, and the physical system state.
3. Various entity types - The data model will consider two types of system entities, namely dynamic and static. The class of static entities covers testbed setup profiles which contain testbed components, network interfaces, and their settings. These entities are normally predetermined or collected in the initialization of each measurement. The class of dynamic entities captures various system events such as machine status reports, network traffic, and information flows in the testbed. These entities are dynamically added into the data set whose quantities and properties are determined by the measured data.
4. Data Model with multiple abstraction mechanisms - The considered data model and the corresponding queries should encompass multiple levels of abstraction including traffic data level, physical hardware level, physical environment level, physical actions level, and the interactions between these various levels. The network database system must allow for the categorization and labeling through these levels.
5. Time travel queries - The data model and the resulting database should allow for direct querying for temporal variations of the studied entities. Hence, temporal relationships between data packets and the corresponding physical actions should be stored and directly accessible.
6. Efficient path and relationship queries - Given the requirement of having interrelated nodes, the query language should allow for path and relationship queries to directly extract this information. These types of queries are used for calculating

various system metrics and hence should be performed in an efficient manner.

Given these discussed requirements, the graph database approach is selected for data management due to the following reasons. First, the data model is defined such that nodes of same type may have different sets of properties, and hence, the GDB offers the ability to store data without an enforced schema such that there is more flexibility of how the data is organized and accessed in the most suitable way for the application [54]. The GDB also allows one to gain insights from the relationships between data points or in applications where the information available to end users is determined by their connections to others. Furthermore, the GDB intuitively display data, thus, visual inspection of certain data connections can be performed [9, 38, 55]. More specifically, the collected data coming out of the testbed are in table format. However, to perform queries that traverse these tables requires expensive processing; hence building a connected graph once makes the traversing easier and obtaining the queries more direct.

Regarding the use of Neo4j in the proposed GDB approach, we found through studying the literature that Neo4j offers the following: neighborhood queries or graph traversal, the ability to be embedded within our analytical tools in Python, the query complexity not dependent on the graph size, and the use of Cypher querying language [55–57]. More specifically, the stored records in Neo4j are linked with direct pointers to avoid maintaining an additional dedicated indexing structure to traverse the graph and consequently, the query complexity does not depend on the graph size. Instead, it only depends on how large the visited subgraph is. Moreover, in Table 3 and Section 4.8 in [55], a comparison between Neo4j and other GDB approaches is summarized where the criteria of comparison are the used model, the record storage properties, storing edges, and data distribution. It was stated in [55] that Neo4j is the most popular graph database system, according to different database rankings. On the other hand, the Cypher syntax is used for querying the graph database structure in Neo4j. Cypher is a declarative query language that allows users to specify which actions they want to perform, such as, matching, inserting, updating, and deleting graph data. The syntax is in the ASCII format, which provides a well-known and legible way to collapse patterns of nodes and relationships within graph data sets [56]. It was stated in [57] that Neo4j can store hundreds of trillion entities. Neo4j can support the operations of storage, query, backup, and redundancy for large-scale data. Although these characteristics can be found in other tools, we found that Neo4j satisfies them well, and hence, suitable for our application.

4.1 Reference Data Model

In a GDB, the data model, which can be roughly analogous to the "schema" of relational databases, illustrates how data records are organized and stored in a graph. However, unlike a fixed schema, the data model of GDBs has more flexibility of depicting diverse data types, content, and connections between different entities whose structure and property profile can update and evolve with more data and/or better observation. A data model contains different node types with specific properties in the graph and various relationships between them. In [13], we identified the requirements of a GDB data model and built a graph containing nodes and relationships that mainly exhibit information around networked industrial devices in a factory work-cell. In this paper, we further populate the earlier-defined work-cell data graph by introducing additional node types characterizing physical actions that are newly captured. Accordingly, we update the relationships, such as associating individual quality of service (QoS) report data from the wireless sniffer with the packets captured at the collocated receiver. The updated data model provides a comprehensive view of production operations, information flows, and wireless channel variations in the testbed, which facilitates further analysis work.

As shown in Fig. 4, an example is illustrated here that summarizes nodes, relationships, and their key properties used in the GDB. We will elaborate the definition and use of these entities in the remainder of this section.

Node Design

To effectively depict testbed operations in the measurement, we define a series of node types in the graph. Nodes are GDB elements that are used to identify testbed components, device states, and messages that store snapshots of the testbed for further analysis. They can be found in two main classes depending on what type of objects the node represents.

The class of *static* nodes covers testbed setup profiles, which contain testbed components, network interfaces, and their settings. These entities are normally predetermined or collected in the initialization of each measurement. They usually remain constant in each round of measurements. In our data model, this class of nodes include the following.

Actor A physical component within the factory work-cell such as a robot, PLC, or other networked item.

NtwkID A network address item for an actor such as an Internet Protocol (IP) address.

SMS An spectrum management system (SMS) observes and records significant spectral events within the work-cell and may report those events to actors within the work-cell.

Sniffer Measurement device that records all transmissions conducted over the wireless medium and includes the wireless header information for each wireless transmission detected.

Adapter Device that serves to connect an actor to a network (adapters are divided into sub-categories depending on the type of interface to a network).

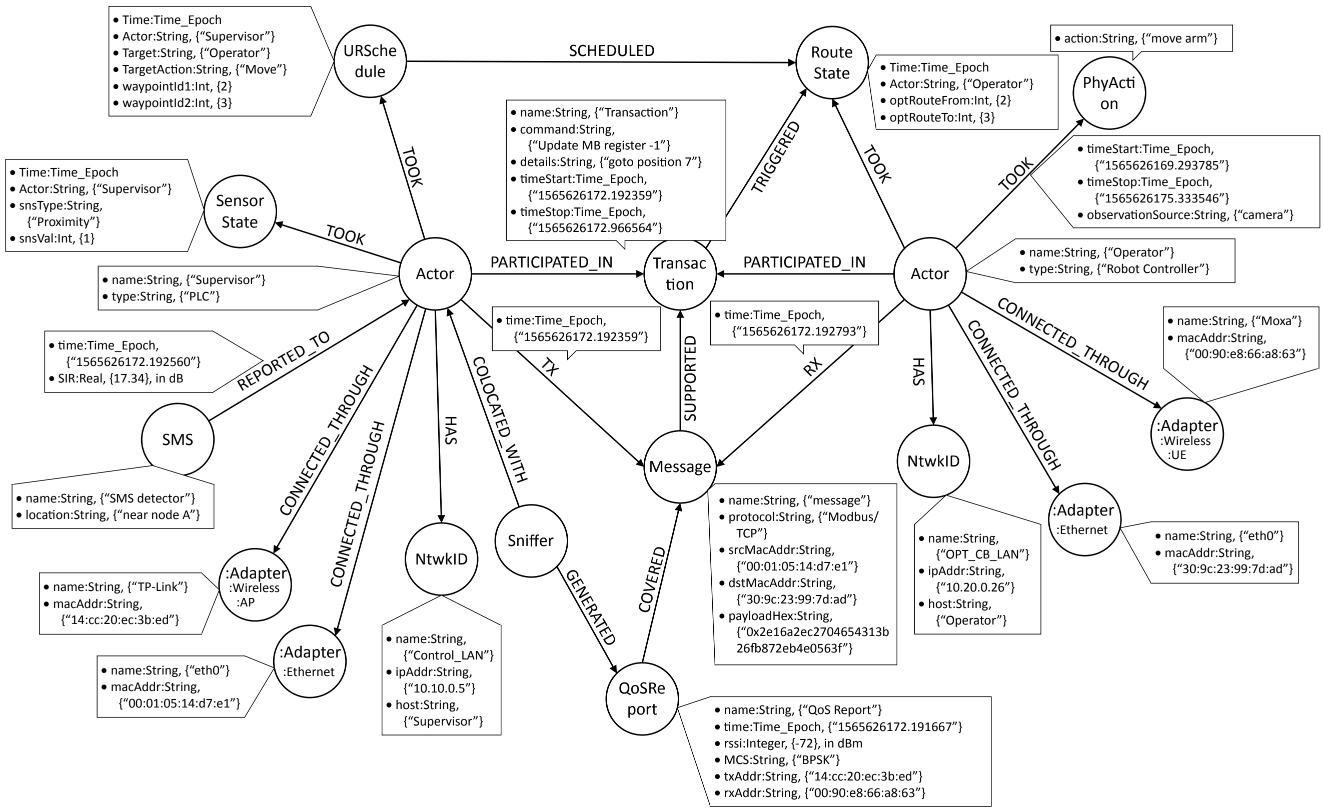


Fig. 4: The data model of the graph database used for each operational run of the NIST wireless factory testbed. The graph is organized into nodes and edges, where the edges signify relationships among network elements and physical operational elements.

Adapter:Ethernet A subcategory of adapter representing an Ethernet interface.
Adapter:Wireless A subcategory of adapter representing a wireless interface.
Adapter:Wireless:AP A subcategory of adapter representing a wireless access point interface.
Adapter:Wireless:UE A subcategory of adapter representing a wireless user equipment interface.

The class of *dynamic* nodes in the graph captures various system events such as machine status reports, network traffic, and information flows in the testbed. These nodes are dynamically added into the graph, and that quantities and properties are determined by the measured data. In our data model, this class of nodes include the following.

Transaction A complete information exchange between two or more actors (multiple actors may participate in a transaction).
Message A network transmission event that occurs between two actors (messages are essentially packet transmissions captured at the transport layer; multiple messages support a transaction).
QoSReport Quality of service report of a message (not all messages will have a QoS report).
Physical Action (PhyAction) A physical occurrence within the factory work-cell associated with Actors through multiple time-based relationships.
PhyAction:URSchedule A subcategory of PhyAction representing a schedule decision made by the supervisor PLC for a robot
PhyAction:SensorState A subcategory of PhyAction representing a real-time reading of the proximity sensor state in a CNC
PhyAction:RouteState A subcategory of PhyAction representing a real-time reading of the action route in a robot

Graph Relationships

A relationship in the graph denotes an action taken to associate two nodes, either homogeneous or heterogeneous ones, which shows their connections in the topology, timeline, or affiliation. We identify the following relationships in the testbed.

PARTICIPATED_IN Actors will participate in transactions. A transaction exists for each logical set of messages between actors such as the setting of a Modbus register or the sending of a command to a robot. Therefore, actors will participate

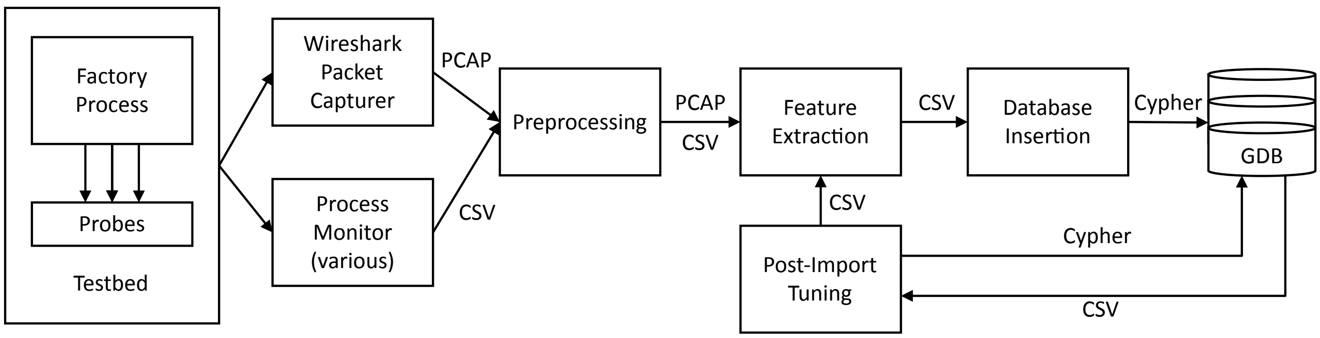


Fig. 5: Data processing flow from factory work-cell to database

in many transactions, and multiple actors may participate in a single transaction.

SUPPORTED Messages (i.e., packets between actors) are associated with transactions through the SUPPORTED relationship. Depending on the protocol and the quality of the channel, a single transaction could have one or many messages connected through this relationship.

TX/RX An actor may either transmit (TX) or receive (RX) a message. Both the TX and RX relationships contain a timestamp in the format of an epoch time which is a floating point number in seconds since January 1, 1970, with a resolution of microseconds.

TOOK When an actor performs a physical action, a TOOK relationship is created between the actor and the physical action node. This relationship contains start and stop time properties as well as the source of the observation such as a networked camera.

REPORTED_TO An SMS may be a passive or active listener within a work-cell. When an SMS operates as an active listener, spectral reports from the SMS may be sent to an actor such that the actor can respond intelligently to the spectral event. Reports from an SMS to an actor are captured within this relationship.

COVERED A wireless sniffer keeps monitoring the working wireless channel(s) and extracts the real-time link QoS information from the sniffed wireless packets, such as the received signal strength indicator (RSSI). A COVERED relationship links the QoSReport node with the concurrent Message node received at the same spot. Not all Message nodes have such a relationship with QoSReport which depends on the availability of the sniffer collocated with the receiver and any wireless sniffer data reported during the transmission.

Other relationships shown in Fig. 4 but not explained above are considered self-explanatory.

Closer Examination

The graph data model is designed in a way where nodes and relationships are centered around Actors. Actors have dual roles in the work-cell operations. In the factory system, Actors participate in the production operations. In the example of Fig. 4, two Actor nodes are presented. In this case, Actor "Supervisor" is the supervisory controller, and Actor "Operator" is a robot arm. The Supervisor schedules the production, collects the other Actors' states, and hosts supportive services, such as SMS. The Operator follows the instructions of the Supervisor and moves parts between work stations. Meanwhile, Actors also act as communication nodes that exchange messages between each other through various network interfaces. In Fig. 4, Actors participate in a transaction, which, in this example, is a Modbus/TCP exchange. The transaction itself is associated with one or more messages (i.e., packets). Each message associated with a transaction manifests itself as a node in the graph. Multiple message nodes will exist for each transaction. Additionally, QoS reports may be associated with each actor node through a collocated sniffer node.

Dynamic event nodes in the measurement, i.e., physical actions, network messages, information transactions, and QoSReport records, have timestamps representing "measurement time" of the recorded events. Once a new event occurs, a proper relationship would be added between the actor and the physical/network event node. All timestamps are accurately synchronized to the grand-master clock.

4.2 Information Workflow

A multi-stage workflow is deployed to feed the graph with instances of nodes, relationships, and their properties that are extracted from measurement data, as shown in Fig. 5. In the measurement data set, network data from distributed network probes in selected links is stored in packet capture (PCAP) files, while operational data that comes from different PLC and robot controllers is stored in the format of comma separated value (CSV) files. The whole process contains four steps including data preprocessing, feature extraction, database insertion, and post-import tuning. Such conversion from raw data

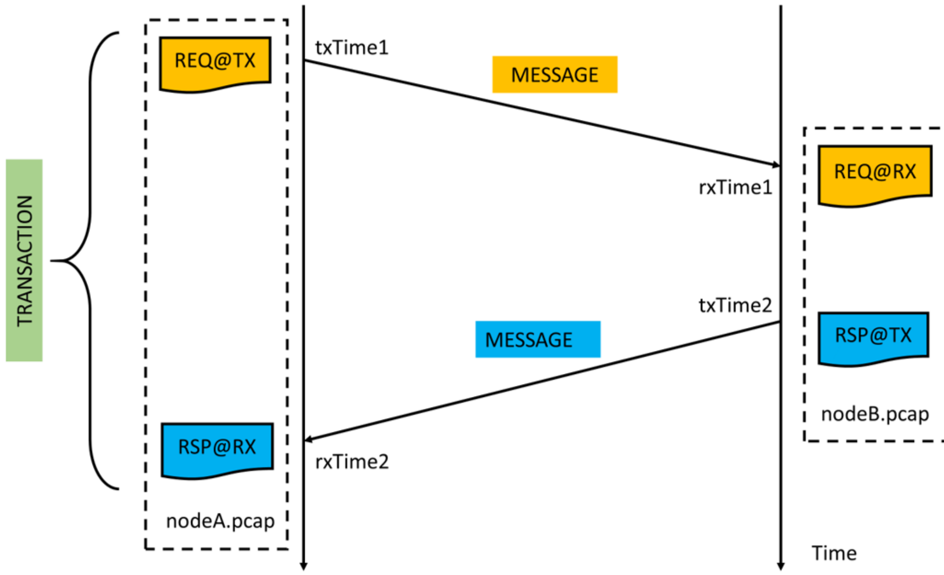


Fig. 6: Timeline illustration of multiple network captures in a control command transaction

to the ready-to-go graph has been done by running automated scripts on a host machine that maintains data repositories of measurement results and deploys the Neo4j desktop application. The functions and operation features in individual steps are discussed next.

Data Preprocessing

Data preprocessing is the first step where measurement data is verified, cleaned, and formatted to facilitate the following processing steps. As fore-mentioned, measurement data contains results collected from heterogeneous modules/devices in the testbed, which may adopt different data types, sampling rates, time and metric resolution, and file formats. For example, different machines may represent and store the record timestamps in various formats depending on local clock settings. Once we obtained the data, we unified the time representation in the entire data set using the time epoch that has microsecond resolution. In another example, we deployed packet filters to remove unrelated packet captures. In treating measurement data, e.g., experimenting with single or double wireless interference links, we managed to reduce the sniffer data in the order of gigabytes to only a few megabytes while keeping all signaling handshakes of interest in the studied links.

Feature Extraction

Feature extraction refers to the process of extracting relevant information from measurement data and prepare the data for insertion into the database. Nodes and relationships are defined by a set of features that share common views. We developed bash and Python scripts that pick the desired features to produce CSV files that are ready for insertion into the Neo4j database. In this step, a bash script was developed running the tshark tool to extract fields of protocol headers in packet captures and save the field information into CSV files. Each line in these CSV records will create one Message node instance as a sender or receiver copy of the packet through a link. A Python script was also used to detect state switches in the physical action data and label these moments that were triggered by testbed communications.

For example, Fig. 6 illustrates the timing information that is extracted from network capture data and used as features in the created nodes, e.g., Transactions and Messages, and relationships, e.g., TX and RX. A complete message transaction in the PLC-PLC/UR3 link includes two messages, i.e., a request and a response. In the measurement data, there are four packet copies corresponding to one transaction regardless of retransmissions or packet loss. Therefore, in the feature extraction step, our bash script calls tshark to dissect packet captures to obtain these four timestamps which will be used later to pair the transmitted and received packet copies.

Graph Insertion

We load the prepared data into the Neo4j GDB using bulk importing, which can create one or multiple nodes and/or relationships in the graph by reading a CSV file once. Neo4j uses Cypher to construct GDB queries to import data. As the output of feature extraction, each line of the CSV file can create one new node entity and/or generate the criteria of linking two qualified nodes for a new relationship. Properties of new entities can be assigned explicitly by the column values of records or inferred from predetermined rules such as some fixed combination of nodes and edges in the graph. Multiple types of nodes can be created from the same data file using one common node template in which each node type has its own subgroup of properties. For example, Modbus and ADS packets use the same Message node structure in our graph to manage the common transmission information such as IP addresses and TCP session identification. Meanwhile, each of these Messages maintains its own application layer header information in the node properties, e.g., Modbus register addresses and ADS function codes.

Post-Import Tuning

Post-import tuning refers to any additional modification in the graph after CSV data is imported. This step treats a few cases where raw data work with current graph insights to obtain new ones. First, in the additive insertion case, i.e., when new data is added into the graph, it links the newly added nodes to the existing ones following necessary relationships between them. Time series data often use this method to link consecutive event nodes in the recorded process. Second, it is the case in which higher level features are needed in the graph that can be abstracted from the imported data. For example, Transaction nodes are built upon Message nodes who participate in the same application transactions; Message nodes themselves are also the summary of packet data, i.e., packet copies at the wireless transceivers. Third, it can feed feature extraction with pieces of information in the current graph for purposes such as coupling data records. For example, coupling QoS reports and Messages in their observation windows used to be an extremely time-consuming process. On one hand, each Message raw data, i.e., the transmitter or receiver copy, contains only half of the transmission time window information. On the other hand, the Cypher query takes a long time to find all eligible relationships as Neo4j would generate a huge Cartesian product when treating the large sample set. We solved this issue by obtaining qualified Message nodes and feeding them into feature extraction where a more efficient Python script finds all Message-QoS Report pairs and later presents them in the graph as new COVERED relationships.

The above four data processing steps can be performed through multiple iterations to treat data and refine the graph according to the data complexity and requirements.

5 Results

Once the data resides within the database, we apply queries to extract information for the evaluation of work-cell performance and visualization of network and operational events within the work-cell. By tracking paths through the relationships within the graph, discerning how a network event such as interference relates to physical events such as position uncertainty or part throughput is possible. Various impairments may be introduced as a part of work-cell operation. Examples of such impairments include competing wireless traffic, radio interference, and reflections and diffraction due to the multi-path environment [58]. We have shown that it is feasible to implement such impairments and measure the resulting physical performance manifestation [12].

This section is introduced to show the type of results that can be obtained for the investigated use case using the GDB approach. These quantities include the impact of wireless transaction latency on the corresponding physical action processing time and the correlation between these quantities. This section introduces the realized data model of the implemented GDB as well to verify that the implemented GDB follows the intended data model. Although the obtained results reflect the performance of the testbed, and hence, can be obtained using other approaches. It will be overly complicated to obtain the one-to-one connection between the physical actions and the network activities using another approach to the best of our knowledge. Traversing the tables of the collected data for each query will be time consuming if the GDB is not built. Hence, we assert that our approach can achieve these results in an efficient way.

In the following subsections, we show results from an experimental scenario of the NIST industrial wireless testbed. In this scenario, two wireless links are used to connect the robot controllers and the wireless AP that is connected to all the other actors in the testbed. The wireless nodes are IEEE 802.11b/g/n devices. During each run of this experimental scenario, the production of 20 parts was emulated, which resulted in 12 minutes of network activity. We performed 4 different experimental cases with respect to the communications network, namely, 1) wired baseline where all links are connected using Ethernet cables to act as a benchmark for performance comparison, 2) wireless baseline where the robots traffic is the only traffic transferred over the wireless network, 3) with 2500 packets per second (pps) wireless traffic where a pair of a source and a sink operate simultaneously with the robots traffic over the wireless network, and 4) with 2x1250 pps traffic where two communications pairs of a source and a sink that each source generates 1250 pps traffic simultaneously with the robots traffic. These external traffic pairs have packet size of 1000 Bytes.

5.1 Realized Schema

After populating the database with data captured from the experiment runs, the resulting realized schema is shown in Fig. 7. The schema visualization is produced by invoking the command

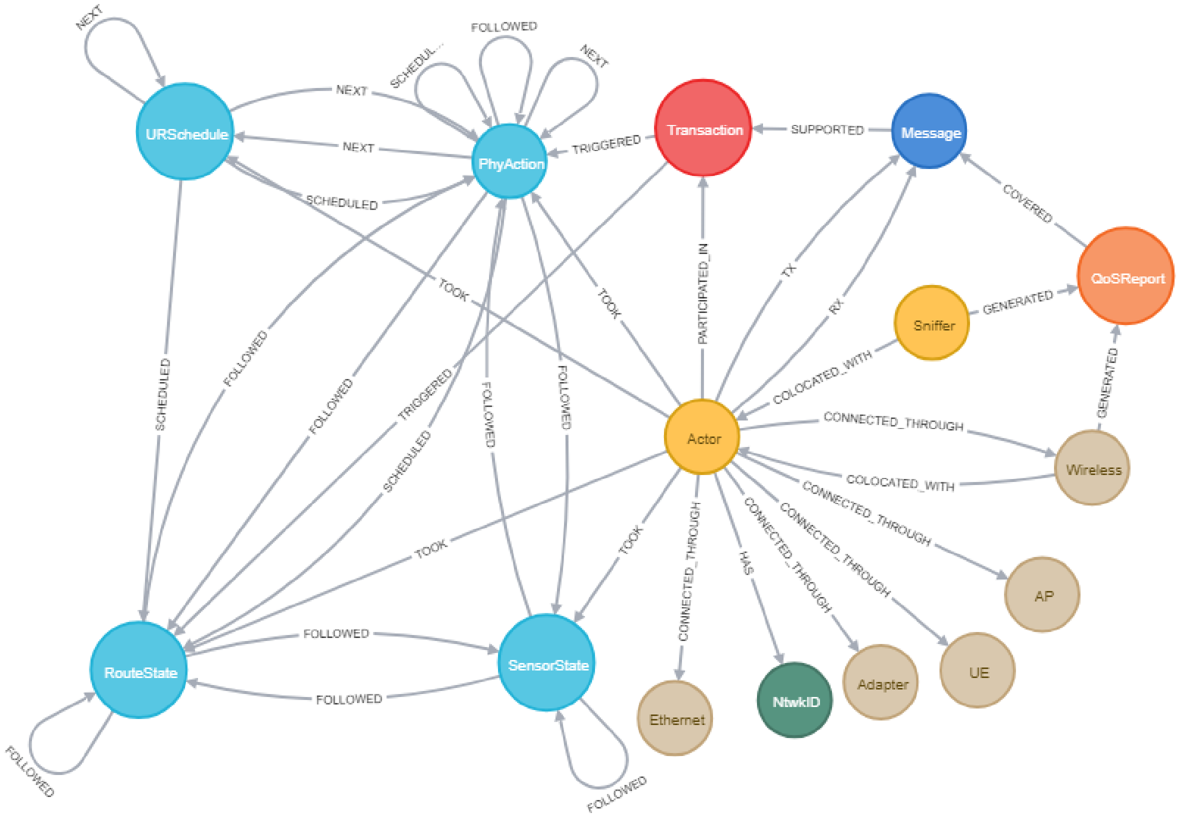


Fig. 7: Realized schema of the graph database fully populated after capturing network and operational data from the NIST industrial wireless testbed

```
call db.schema.visualization()
```

in Neo4j. It is important to note that, in comparison to the data model shown in Fig. 4, a realized schema shows only one representation of each node and relationship. Where label inheritance is employed, such as the case for different adapter types, relationships are reproduced; however, this is a result of the visualization tool rather than the schema itself. Fig. 7 serves, therefore, to validate that the intended data model was indeed realized by the insertion of event data from the testbed. In the realized schema, inherited labels are shown as separate nodes.

5.2 Physical Actions Processing

In this subsection, we use the extracted data from the GDB to study the impact of the wireless communications on the physical action performance. We focus our analysis in this subsection on the URSchedule and RouteState progress over time where URSchedule is the dynamic node to represent a physical action decision at the supervisor and RouteState is the dynamic node to represent a physical action command received by one of the robots where the command parameters are stored at the robot registers. Note here that the transaction between a robot controller and the supervisor is initiated by a request message from the robot controller and terminated by correctly receiving a response message from the supervisor to the robot controller as well.

The supervisor takes decisions based on available information about the testbed. Once it makes a decision, it is reflected on the value of the URSchedule. We define the supervisor processing time as the time from the instant the decision is taken to the instant when the wireless transaction is initiated to request a new physical action and it is denoted by T_{Sup}. Then, the transaction latency is the total time spent by all the wireless packets corresponding to an action such that it is the time between the instant, at which the wireless transaction is initiated by the robot controller to require a new action until the data arrives from the supervisor at the intended robot controller. The wireless transaction latency is denoted by T_{W}. The robot processing time is the time between the instant the wireless data is received by the robot controller to the instant when the required action is updated in the RouteState register indicating the physical action starts. The robot processing time is denoted by T_{Rob}. The total physical action time, which represents the time needed for a supervisor command to be reflected

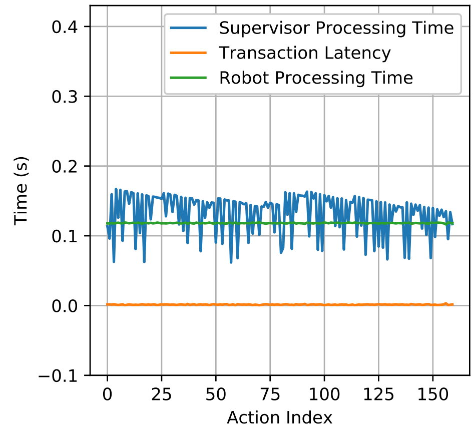


Fig. 8: Wired baseline physical action time

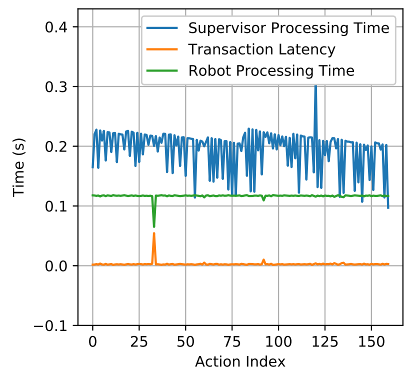


Fig. 9: Wireless baseline physical action time

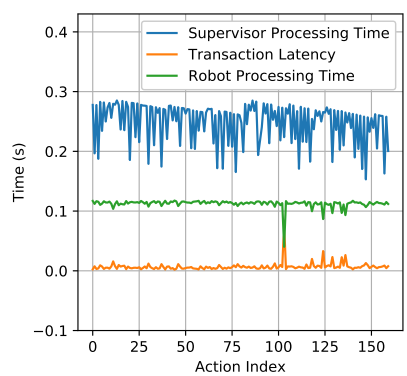


Fig. 10: Wireless with 2500 pps traffic physical action time (run 1)

at the corresponding robot, is denoted by T_{Act} and evaluated through

$$T_{\text{Act}} = T_{\text{Sup}} + T_{\text{W}} + T_{\text{Rob}}. \tag{1}$$

In Fig.8-12, we present the values of the three components of the total physical action time for each run of the testbed. The horizontal axis represents the action index for all the operator and inspector actions while the corresponding time components are shown in the vertical figure axis.

In Fig. 8, the value of the transaction latency is almost fixed by deploying a wired channel while in Fig. 9 more fluctuations start to appear because of introducing a wireless channel in the wireless baseline. In this case, the channel is relatively good so very few fluctuations happen. In later Figs. 10-12, the wireless interference is introduced where these fluctuations increase significantly. The RouteState register update happens once during the robot program scan cycle in a periodic fashion

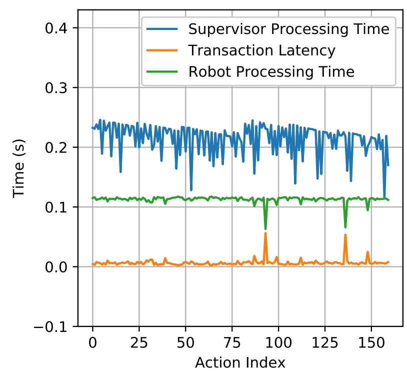


Fig. 11: Wireless with 2500 pps traffic physical action time (run 2)

Fig. 12: Wireless with 2x1250 packets/s traffic physical action time

almost every 120 ms. In this same loop, the transaction initiation happens as well, and hence, the sum of these two quantities ($T_{\mathrm{W}} + T_{\mathrm{Rob}}$) equals the loop time of approximately 120 ms. That is why when the transaction latency increases with certain amount of time, the corresponding robot processing time dips with exactly the same amount of time. Such observations are captured in the figures 8-12.

On the other hand, most of the randomness in the total physical action time results from the supervisor processing time. We also notice that the randomness in the supervisor processing time is not impacted by the wireless channel where multiple runs of the same wireless case with the same interfering traffic have completely different supervisor processing time performance as shown in Fig. 10 and 11.

5.3 Stochastic Distribution of Physical Action Time

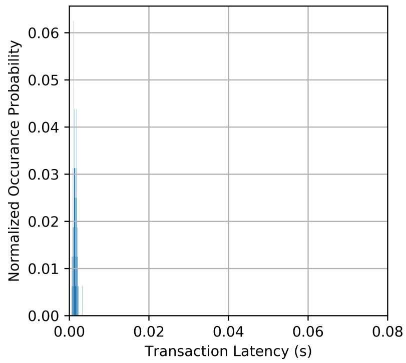
In this subsection, we present the normalized histograms of the transaction latency and the total physical action time in Figs. 13 and 14, respectively. In Fig. 13, we notice the clear impact of the communication channel and interference on the histogram of the transaction latency where the mean and the variance are clearly impacted by the wireless parameters. On the other hand, the total physical action time is not impacted directly by the communications channel in this use case due to the fact that the robot processing loop compensate of any transaction latency below the loop time of 120 ms.

5.4 Timeline Visualization

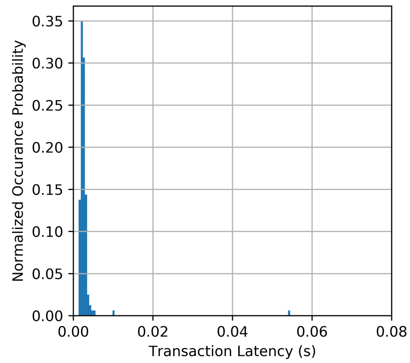
Finally, we present a simple visualization result achieved by processing the data through the GDB. In Fig. 15, we draw the timeline of various events related to the physical actions. These events happen at the supervisor and the corresponding robot where the mapping is achieved through creating and querying the GDB. The detailed timeline is just a zoomed in version to the data in the left where URSchedule value changes can be captured through this visualization. The main connection between these events as shown in the schema is the triggering transaction.

5.5 Discussion

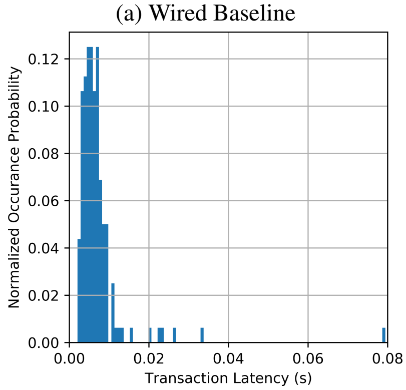
In the studied specific use case of machine-tending supervisory control, the impact of network latency on physical action is the main metric to study for supervisory control applications. Other metrics, such as reliability, can be beneficial in other



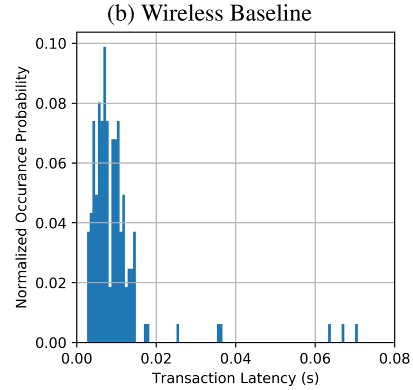
(a) Wired Baseline



(b) Wireless Baseline

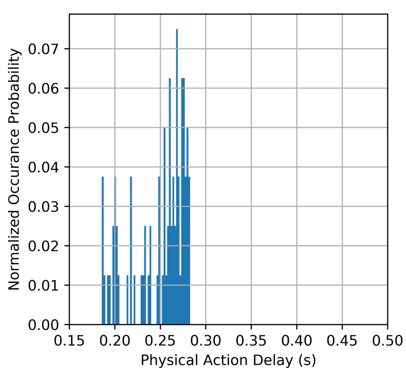


(c) With 2500 pps Traffic

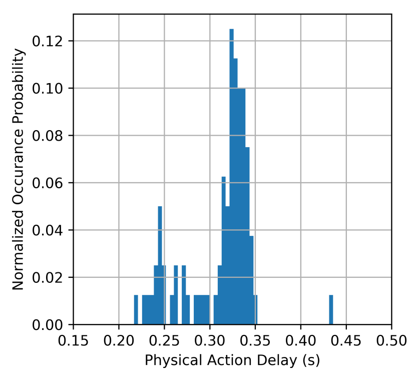


(d) With 2x1250 pps Traffic

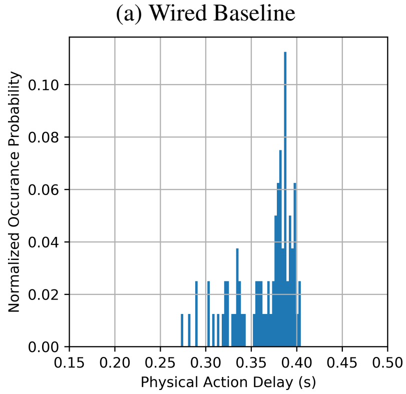
Fig. 13: Histograms of Transaction Latency for Various Experimental Scenarios



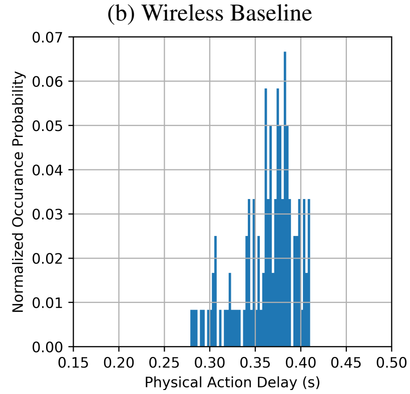
(a) Wired Baseline



(b) Wireless Baseline

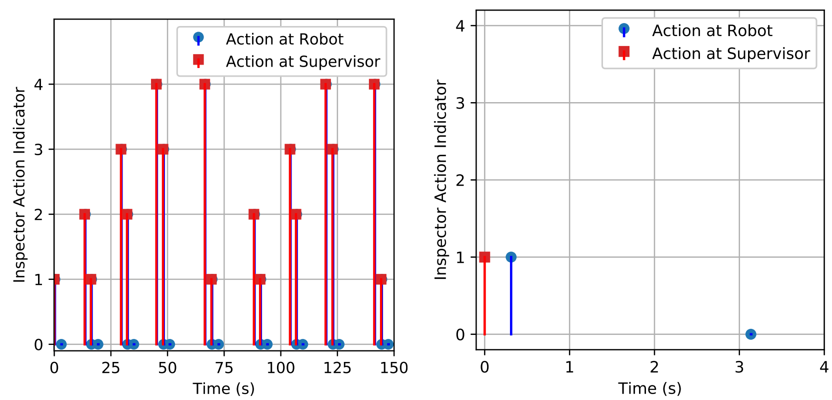


(c) With 2500 pps Traffic



(d) With 2x1250 pps Traffic

Fig. 14: Histograms of Physical Action Time for Various Experimental Scenarios



(a) Action Timeline

(b) Action Timeline-Detailed

Fig. 15: Timeline of various physical action related events

applications as feedback control. Hence, we assessed the performance through latency related metrics in this use case. In future work, we plan to modify the use case to allow for further metrics evaluation. The current set of results illustrates the use of our approach in achieving latency results and evaluate interference impact on this specific use case.

6 Conclusions

We have presented in this paper a novel approach to capturing network and operational event information from a factory work-cell with the purposes of 1) capturing and storing network and operational events, 2) calculating performance metrics of the network, and 3) discovering performance dependencies between the network and the physical assembly of the work-cell. Using a graph database, we have demonstrated that it is possible to construct such a database, compute network performance metrics and discover correlations. We have also developed the capability of examining the correlation between network events and the performance of physical actions. We have tested this approach in an emulated robotic manufacturing factory work-cell that has two collaborative grade robot arms for a pick-and-place task. We have shown that wireless transaction latency has a minimal impact on the physical actions processing time in this use case. This behavior is expected to occur in many similar use cases in which the physical action is performed after a loop scan for the action triggering parameters.

The future progress and measurement data will be released in the NIST public domain repository as a reference for industrial traffic modeling efforts and comparative studies on industrial wireless technologies [59]. Deploying this approach allows for having a direct connection between network packets and physical actions. Initially, this can help in the wireless network design by allowing the network to react to the events that may cause physical actions disturbance especially in mission-critical applications. In future, the results can be used directly in control domain where the control loops will be allowed to react to changes in physical parameters through adapting wireless networks parameters. Furthermore, by deploying artificial intelligence, performing predictive analysis can allow the system to take the corrective actions early enough not to disturb the physical process.

Disclaimer

Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

References

[1] Martinez, B., Cano, C., and Vilajosana, X., 2019. "A square peg in a round hole: The complex path for wireless in the manufacturing industry". *IEEE Communications Magazine,* **57**(4), Apr., pp. 109–115.

[2] Huang, V. K. L., Pang, Z., Chen, C. A., and Tsang, K. F., 2018. "New trends in the practical deployment of industrial wireless: From noncritical to critical use cases". *IEEE Industrial Electronics Magazine,* **12**(2), June, pp. 50–58.

[3] Vilajosana, X., Cano, C., Martínez, B., Tuset, P., Melià, J., and Adelantado, F., 2018. "The wireless technology landscape in the manufacturing industry: A reality check". *ArXiv,* **abs/1801.03648**.

[4] Kagermann, H., Wahlster, W., and Helbig, J., 2013. Recommendations for implementing the strategic initiative industrie 4.0, industrie 4.0 working group.

[5] Barnard Feeney, A., Frechette, S., and Srinivasan, V., 2017. *Cyber-Physical Systems Engineering for Manufacturing*. Springer International Publishing, Cham, pp. 81–110.

[6] Huang, V. K. L., Pang, Z., Chen, C. A., and Tsang, K. F., 2018. "New trends in the practical deployment of industrial wireless: From noncritical to critical use cases". *IEEE Industrial Electronics Magazine,* **12**(2), June, pp. 50–58.
[7] Dai, H.-N., Wang, H., Xu, G., Wan, J., and Imran, M., 2019. "Big data analytics for manufacturing internet of things: opportunities, challenges and enabling technologies". *Enterprise Information Systems*, June, pp. 1–25.
[8] Tao, F., Qi, Q., Liu, A., and Kusiak, A., 2018. "Data-driven smart manufacturing". *Journal of Manufacturing Systems,* **48**, July, pp. 157–169.
[9] Drake, M., 2019. A Comparison of NoSQL Database Management Systems and Models.
[10] Lade, P., Ghosh, R., and Srinivasan, S., 2017. "Manufacturing analytics and industrial internet of things". *IEEE Intelligent Systems,* **32**(3), May, pp. 74–79.
[11] Angles, R., and Gutierrez, C., 2008. "Survey of graph database models". *ACM Comput. Surv.,* **40**(1), Feb., pp. 1:1–1:39.
[12] Liu, Y., Candell, R., Kashef, M., and Montgomery, K., 2019. "A collaborative work cell testbed for industrial wireless communications — the baseline design". In 2019 IEEE 28th International Symposium on Industrial Electronics (ISIE), pp. 1315–1321.
[13] Candell, R., Kashef, M., Liu, Y., Montgomery, K., and Foufou, S., 2020. "A graph database approach to wireless IIoT work-cell performance evaluation". In 2020 IEEE International Conference on Industrial Technology (ICIT). (preprint, URL: https://www.nist.gov/publications/graph-database-approach-wireless-iiot-work-cell-performance-evaluation).
[14] Ahmadi, A., Moradi, M., Cherifi, C., Cheutet, V., and Ouzrout, Y., 2019. "Wireless Connectivity of CPS for Smart Manufacturing: A Survey". In International Conference on Software, Knowledge Information, Industrial Management and Applications, SKIMA.
[15] Montgomery, K., Candell, R., Liu, Y., and Hany, M., 2019. Wireless User Requirements for the Factory Work-cell. Tech. rep. URL: https://www.nist.gov/publications/wireless-user-requirements-factory-workcell.
[16] Pang, Z., Luvisotto, M., and Dzung, D., 2017. "Wireless High-Performance Communications: The Challenges and Opportunities of a New Target". *IEEE Industrial Electronics Magazine,* **11**(3), pp. 20–25.
[17] Damsaz, M., Guo, D., Peil, J., Stark, W., Moayeri, N., and Candell, R., 2017. "Channel modeling and performance of Zigbee radios in an industrial environment". *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS*.
[18] Candell, R., Remley, K. A., Quimby, J. T., Novotny, D., Curtin, A., Papazian, P. B., Kashef, M., and Diener, J., 2017. Industrial wireless systems radio propagation measurements. Tech. rep. doi: 10.6028/nist.tn.1951.
[19] Islam, K., Shen, W., and Wang, X., 2012. "Wireless sensor network reliability and security in factory automation: A survey". *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*.
[20] Peil, J., Damsaz, M., Guo, D., Stark, W., Candell, R., and Moayeri, N., 2017. Channel Modeling and Performance of Zigbee Radios in an Industrial Environment (NIST-TN-1941). Tech. rep., NIST, Gaithersburg, MD.
[21] Lu, C., Saifullah, A., Li, B., Sha, M., Gonzalez, H., Gunatilaka, D., Wu, C., Nie, L., and Chen, Y., 2016. Real-Time Wireless Sensor-Actuator Networks for Industrial Cyber-Physical Systems. doi: 10.1109/JPROC.2015.2497161.
[22] Aminian, B., Araujo, J., Johansson, M., and Johansson, K. H., 2013. "GISOO: A virtual testbed for wireless cyber-physical systems". In IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society, IEEE, pp. 5588–5593.
[23] Jecan, E., Pop, C., Padrah, Z., Ratiu, O., and Puschita, E., 2018. "A dual-standard solution for industrial Wireless Sensor Network deployment: Experimental testbed and performance evaluation". In 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS), IEEE, pp. 1–9.
[24] Ding, Y., Hong, S. H., Lu, R., Kim, J., Lee, Y. H., Xu, A., and Xiaobing, L., 2015. "Experimental investigation of the packet loss rate of wireless industrial networks in real industrial environments". In 2015 IEEE International Conference on Information and Automation, IEEE, pp. 1048–1053.
[25] Liu, Q., Ma, L., Fan, S.-Z., Abbod, M. F., Lu, C.-W., Lin, T.-Y., Jen, K.-K., Wu, S.-J., and Shieh, J.-S., 2018. "Design and Evaluation of a Real Time Physiological Signals Acquisition System Implemented in Multi-Operating Rooms for Anesthesia". *Journal of Medical Systems,* **42**(8), aug, p. 148.
[26] Fink, J., Ribeiro, A., and Kumar, V., 2013. "Robust Control of Mobility and Communications in Autonomous Robot Teams". *IEEE Access,* **1**, pp. 290–309.
[27] Liang, W., Zheng, M., Zhang, J., Shi, H., Yu, H., Yang, Y., Liu, S., Yang, W., and Zhao, X., 2019. "WIA-FA and Its Applications to Digital Factory: A Wireless Network Solution for Factory Automation". *Proceedings of the IEEE,* **107**(6), jun, pp. 1053–1073.
[28] Candell, R., 2015. "A Research Framework for Industrial Wireless Deployments". In Proceedings of 2015 ISA Instrumentation Symposium.
[29] Liu, Y., Candell, R., Lee, K., and Moayeri, N., 2016. "A simulation framework for industrial wireless networks and process control systems". In 2016 IEEE World Conference on Factory Communication Systems (WFCS), IEEE, pp. 1–11.
[30] Wang, J., Zhang, W., Shi, Y., Duan, S., and Liu, J., 2018. "Industrial big data analytics: Challenges, methodologies, and applications". *CoRR,* **abs/1807.01016**.

[31] Lee, J., 2015. *Industrial Big Data (Mechanical Industry Press, China)*. 07.
[32] Raptis, T. P., Passarella, A., and Conti, M., 2019. "Data management in industry 4.0: State of the art and open challenges". *IEEE Access,* **7**, pp. 97052–97093.
[33] Wan, J., Tang, S., Li, D., Wang, S., Liu, C., Abbas, H., and Vasilakos, A. V., 2017. "A manufacturing big data solution for active preventive maintenance". *IEEE Transactions on Industrial Informatics,* **13**(4), Aug, pp. 2039–2047.
[34] (GE). Unlocking machine data to turn insights into powerful outcomes. Online. URL https://www.ge.com/digital/, Accessed 2019-07-01.
[35] Brian Courtney. Industrial big data analytics: The present and future. Online. URL https://www.isa.org/intech/20140801/, Accessed 2019-07-01.
[36] (ABB). Big Data and decision-making in industrial plants. Online. URL https://new.abb.com/cpm/production-optimization/big-data-analytics-decision-making, Accessed 2019-07-01.
[37] Kumar Kaliyar, R., 2015. "Graph databases: A survey". In International Conference on Computing, Communication Automation, pp. 785–790.
[38] Vyawahare, H. R., and Karde, P. P., 2015. "An overview on graph database model". *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE),* **3**, August, pp. 7454–7457.
[39] Satone, K. N., 2014. "Modern graph databases models". *International Journal of Engineering Research and Applications (IJERA)*, pp. 19–24.
[40] Wood, P. T., 2012. "Query languages for graph databases". *SIGMOD Record,* **41**, pp. 50–60.
[41] Jadhav, P. S., and Oberoi, R. K., 2015. "Comparative analysis of graph database models using classification and clustering by using weka tool".
[42] Macko, P., Margo, D., and Seltzer, M., 2013. "Performance introspection of graph databases". In Proceedings of the 6th International Systems and Storage Conference, SYSTOR '13, ACM, pp. 18:1–18:10.
[43] Webber, J., and Robinson, I., 2015. "The top 5 use cases of graph databases". *white paper, Neo4j*.
[44] Küçükkeçeci, C., and Yazici, A., 2019. "Multilevel object tracking in wireless multimedia sensor networks for surveillance applications using graph-based big data". *IEEE Access,* **7**, pp. 67818–67832.
[45] Gomez-Rodriguez, M., Leskovec, J., and Krause, A., 2012. "Inferring networks of diffusion and influence". *ACM Trans. Knowl. Discov. Data,* **5**(4), Feb., pp. 21:1–21:37.
[46] Skhiri, S., and Jouili, S., 2013. *Large Graph Mining: Recent Developments, Challenges and Potential Solutions*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 103–124.
[47] Kan, B., Zhu, W., Liu, G., Chen, X., Shi, D., and Yu, W., 2017. "Topology Modeling and Analysis of a Power Grid Network Using a Graph Database". *International Journal of Computational Intelligence Systems,* **10**(1), p. 1355.
[48] T.K., A. K., Thomas, J. P., and Parepally, S., 2017. "An efficient and secure information retrieval framework for content centric networks". *Journal of Parallel and Distributed Computing,* **104**, jun, pp. 223–233.
[49] Barik, M. S., Mazumdar, C., and Gupta, A., 2016. "Network Vulnerability Analysis Using a Constrained Graph Data Model". pp. 263–282.
[50] Diederichsen, L., Choo, K.-K. R., and Le-Khac, N.-A., 2019. "A graph database-based approach to analyze network log files". In Network and System Security, J. K. Liu and X. Huang, eds., Springer International Publishing, pp. 53–73.
[51] Esser, S., 2019. "Using graph data structures for event logs".
[52] Universal Robots. Real-Time Data Exchange (RTDE) Guide. Online. URL https://www.universal-robots.com/how-tos-and-faqs/how-to/ur-how-tos/real-time-data-exchange-rtde-guide-22229, Accessed 2020-01-08.
[53] Technical Committee, 2008. "IEEE standard for a precision clock synchronization protocol for networked measurement and control systems - redline". *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002) - Redline*, July, pp. 1–300.
[54] Robinson, I., Webber, J., and Eifrem, E., 2015. "Graph database internals". In *Graph Databases*, second edi ed. O'Relly, ch. 7, pp. 149–170.
[55] Besta, M., Peter, E., Gerstenberger, R., Fischer, M., Podstawski, M., Barthels, C., Alonso, G., and Hoefler, T., 2019. "Demystifying Graph Databases: Analysis and Taxonomy of Data Organization, System Designs, and Graph Queries".
[56] Fosic, I., and Solic, K., 2019. "Graph Database Approach for Data Storing, Presentation and Manipulation". In 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), IEEE, pp. 1548–1552.
[57] Huang, H., and Dong, Z., 2013. "Research on architecture and query performance based on distributed graph database Neo4j". In 2013 3rd International Conference on Consumer Electronics, Communications and Networks, IEEE, pp. 533–536.
[58] Candell, R., Remley, C., Quimby, J., Novotny, D., Curtin, A., Papazian, P., Koepke, G., Diener, J., and Kashef, M., 2017. Industrial wireless systems: Radio propagation measurements. Tech. rep., National Institute of Standards and Technology, Gaithersburg, MD.
[59] Candell, R., Hany, M., Liu, Y., and Montgomery, K. Reliable, High Performance Wireless Systems for Factory Automation. URL https://www.nist.gov/programs-projects/reliable-high-performance-wireless-systems-factory-automation.