Unified SNMP Interface for IoT Monitoring

1st Petr Matoušek
Faculty of Information Technology
Brno University of Technology
Brno, Czech Republic
matousp@fit.vutbr.cz

2nd Ondřej Ryšavý
Faculty of Information Technology
Brno University of Technology
Brno, Czech Republic
rysavy@fit.vutbr.cz

3rd Libor Polčák
Faculty of Information Technology
Brno University of Technology
Brno, Czech Republic
polcak@fit.vutbr.cz

***Abstract*—Internet of Things (IoT) is a network connecting various devices like sensors, actuators, and intelligent gadgets that monitor and control temperature, light, humidity, electrical power, and other physical quantities in a smart environment such as smart buildings. Data provided by IoT devices are essential for the management of the smart environment. So, it is important to be aware of the reachability of connected IoT devices, their state, volume of transferred data, connections they make, etc. Unfortunately, many IoT devices communicate directly over the data link layer (Layer 2) as ZigBee, Bluetooth, or WiFi. Traditional network monitoring techniques like SNMP, Netflow, or Syslog, however, require the full TCP/IP stack, so they cannot be directly applied on IoT networks. IoT devices are managed independently through vendor-specific solutions mostly implemented in the cloud. This leads to the divided network management where IP network devices are managed by a central network management system (NMS) while IoT devices are managed separately using proprietary applications. In order to include IoT devices into the network monitoring, two steps are required: (i) obtain IoT monitoring data, (ii) present these data in a standardized format supported by a common NMS. In this paper, we propose a solution based on the SNMP Proxy Agent that collects IoT information from IoT communication and the IoT log file on a local gateway. The agent converts gathered data into MIB objects that are provided to the SNMP monitoring system. Thus, information about IoT devices are fed to the locally deployed network management system. The paper demonstrates the proposed solution on the smart building where IoT data is obtained from MQTT packets and the Home Assistant log file.**

***Index Terms*—network monitoring, SNMP, Internet of Things, MIB objects, MQTT**

I. Introduction

The Internet of Things (IoT) network is a set of heterogeneous devices ("things", nodes) that usually communicate over data link (L2) technology as ZigBee, Z-Wave, Bluetooth, WiFi, or IEEE 802.15.4 [1] with an IoT gateway (hub, controller) that is connected to the Internet. The gateway forwards local IoT data to a cloud for further processing. A typical example of an IoT network is a set of IoT sensors and actuators used for the smart home control that monitor and control heating, air-condition, lights, or surveillance cameras. A smart home user can access the data and control functions in the cloud through a mobile app, web application, or an API, see Fig. 1.

Such a solution is suitable for a private home installation but it is impractical for large installations like smart buildings with tens or hundreds of heterogeneous IoT devices made by different vendors. These devices are typically managed by vendor-specific solutions running in the cloud. This leads to the divided network management: active network devices (switches, routers) plus end systems (desktops, printers, data storages) are managed by an on-site NMS while IoT devices are managed remotely using vendors' apps or APIs.

978-3-903176-32-4 © 2021 IFIP

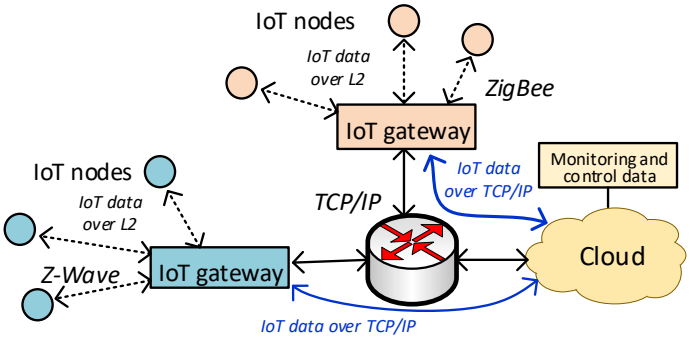

Fig. 1. Communication of IoT devices

Consequently, the network operator needs to apply multiple tools to monitor the complete network state. In case of a device or link failure, it is difficult for him to understand the full impact on the functionality of the system as a whole.

Integration of IoT devices monitoring into the central network monitoring and management system is difficult because most IoT devices do not support TCP/IP communication due to the limited software and hardware capabilities. So, traditional network management frameworks like SNMP [2], Netflow [3] or Syslog [4] cannot be directly applied on IoT networks. Nevertheless, the network admin should be informed about each device connected to the network, including IoT devices.

The main issue is how to obtain monitoring information about IoT devices that communicate on Layer 2 only. There are basically three sources of monitoring data about IoT devices that can be exploited: (i) IoT application protocols like Constrained Application Protocol (CoAP) [5] or Message Queuing Telemetry Transport (MQTT) protocol [6] that transmit data between the local IoT gateway and the cloud application, (ii) a log file at the local IoT gateway that records events of device activities, and (iii) an IoT device API if it is provided.

Having these sources of IoT monitoring data, we need to convert these data into a unified format supported by a NMS. Recall that IoT devices are not standardized and different vendors may implement IoT communication and behavior differently. As a solution, we propose an SNMP proxy agent

that transforms IoT monitoring data into Management Information Base (MIB) objects (also called SNMP objects). Having IoT monitoring data in MIB objects, we can use them for manifold purposes. Data can be visualized on a dashboard of the central NMS with information about the state and communication of IoT devices. They can be exploited for application monitoring when collecting not only network-related data but also IoT measured values like temperature, smart socket power, smart light luminance, etc. Security and system maintenance applications can use this data to detect anomalies and predict failures.

Activity monitoring of IoT communication can reveal privacy and security issues as many smart devices communicate with their vendors sharing various possibly sensitive information. IoT data may include information describing personal behavior like presence at home/office, daily habits related to power consumption, heating, and lights status [7], [8], sometimes even health issues [9]. As data is transmitted to the vendor's cloud, they move out of direct control of the local network administrator. Legally, the cloud operator is a processor (GDPR) or service provider (CCPA). Data can even move to another jurisdiction of a foreign country where the cloud is physically located. Such data transfers may violate data protection laws, for example, European GDPR [10], [11].

While the goal is not to replace the functionality of IoT applications, the collected monitoring data can be used for advanced functions, e.g., in an anomaly detection system that notifies an admin about unusual activities or in an IDS system that raises an alarm when the defined condition is violated.

A. Contribution

The paper overviews existing methods for IoT device monitoring and identifies requirements for IoT network management. The main contribution is a unified SNMP interface that integrates IoT monitoring with SNMP-compatible monitoring solutions. We show how real-time data is obtained from MQTT communication and IoT gateway log files. Having IoT data in the unified format opens new possibilities for advanced IoT management. Also, we give a definition of MIB objects suitable for IoT monitoring. The proposed approach is demonstrated on a case study of the smart building.

B. Structure of the Paper

The paper is structured as follows. Section II gives an overview of current IoT monitoring techniques. Section III discusses specific features of IoT device monitoring, typical requirements, and challenges to IoT monitoring. Section IV presents an architecture of the proposed SNMP proxy agent that serves as a unified interface for IoT monitoring. We show how monitoring data from IoT devices are obtained and transformed into MIB objects. Section V demonstrates our approach on a use case of the smart building. The last section concludes our work and discusses future directions.

II. Related Work

Smart buildings connect IoT devices like sensors and actuators into the network that monitors and controls environmental variables related to the building, e.g., climate (temperature, heating, humidity), lights (level of illumination, energy consumption), safety (fire detection, water leakage) or security (open/close windows and doors, surveillance) [12]. IoT sensors are usually controlled by a building management system (BMS). BMS systems are mostly implemented as proprietary solutions [13] that do not provide direct access to monitoring data. Thus, the network monitoring system is not informed about the health and operational status of IoT devices.

Researchers realized this limitation and have proposed several solutions for gathering IoT data and presenting them according to the requirements of NMS. One approach is based on monitoring and analysis of IoT communication transmitted by CoAP or MQTT protocols. Lindholm-Ventola et al. [14] followed up and implemented two methods for accessing IoT resources: (i) through a shared database where data from CoAP communication are inserted and then retrieved by an SNMP agent, or (ii) through the translation of CoAP messages to SNMP requests/responses. Translation requires mapping between MIB object identifiers (OID) and CoAP resources. Evidently, their approach is similar to ours, however, we also process IoT events from the log files on the IoT gateway.

CoAP communication is based on the client-server paradigm while MQTT protocol uses a publish-subscribe model where a publisher sends monitoring data to an MQTT broker that distributes data to subscribers. Savić [15] proposed an IoT-SNMP Bridge that translates MQTT messages into SNMP objects and transmits them to CloudSNMP. This is useful for providing IoT management as a service (IaaS). Our goal is different. Instead of building a separate IoT management system, we try to integrate IoT monitoring into the central NMS. Han et al. [16] use a different approach to MQTT. They created an SNMP proxy agent called SNMP+Sensor with an MQTT interface and connected the agent to the MQTT broker as a subscriber. Similarly to Savić they store MQTT data in a local database and translate them into SNMP objects. Unfortunately, they do not provide details about implementation, which avoids the direct comparison of our and their design.

There were also several attempts to extract monitoring data directly from an IoT device. Sehgal et al. [17] implemented a lightweight SNMP agent for resource-constrained devices. Their solution, however, requires a full TCP/IP stack implementation for transmitting SNMP messages. In our approach, IoT devices connected at Layer 2 are supported without any modification. Choi at al. [18] developed a system that transmits reduced SNMP messages directly over 6LoWPAN technology. Their solution is, however, not scalable since it requires a special SNMP agent running on every IoT device. Our approach does not aim at accessing IoT devices directly but obtains IoT data indirectly from the IoT gateway.

III. Specific Features of IoT Monitoring

The primary role of network monitoring and management is to provide an updated view of network resources, in terms of their status, availability, and performance. FCAPS model [19] defines five areas of network management consisting

of fault, configuration, accounting, performance, and security management. The model does not specify an implementation of these requirements. Common NMSs integrate various management and monitoring solutions like SNMP, Netflow data, or event logging. Such an approach works well for monitoring enterprise-class IP-based networks consisting of traditional networks and end-host devices. Recent increase of IoT deployment raises a need for the integration of IoT devices into the network monitoring systems.

A. Differences of IoT Communication

Traditional network monitoring obtains information either (1) directly from a device using an SNMP or Syslog agent, or (2) indirectly by observing network flows (Netflow) [20]. Obtained monitoring data is transmitted to the NMS system for analysis, filtering, and visualization, or is further processed to detect threats or anomalies. IoT monitoring works differently:

- Data from IoT devices is transmitted at Layer 2 to an IoT gateway. The IoT gateway relays the data to the cloud using an IP network as shown in Fig. 1. The control of IoT devices is done remotely. Moreover, simple network monitoring of IoT devices is not possible with existing IP-based tools because IoT devices mostly operate on L2.
- IoT monitoring uses vendor-specific solutions that include proprietary communication protocols, clouds (Amazon AWS, IBM Watson, Microsoft Azure IoT Hub, Google Cloud, Cisco IoT DM) [21], [22] and specific APIs to access monitoring and management data. There are also standardized IoT management platforms and protocols like Netconf, Restconf, CoAP, CWMP, or LwM2M [22], [23] which can be partially integrated into standardized NMS systems like Nagios or Zabbix. Still, many IoT devices communicate using proprietary protocols.
- IoT devices have specific communication patterns. Often, they produce a regular stream of real-time information either in push-based (MQTT) or pull-based (CoAP) communication models. The collected data thus may require adaptation, e.g, resampling, aggregation, or filtering, before it is injected into the monitoring system.
- IoT devices often communicate to remotely deployed control applications, mostly located at the cloud of the vendor. This raises legitimate concerns about security and privacy of IoT data which include not only security of IoT data transmissions from a local site to the cloud but also preservation and data protection in the cloud.

B. IoT Monitoring Challenges

To integrate the IoT environment into the existing network monitoring systems, the following issues should be addressed:

- *Access to IoT data* is possible either through the vendor's cloud (web applications, APIs) or by capturing and analysis of IoT messages. In this paper, we do not consider getting data from the cloud-based application. The proposed approach considers obtaining data either by decoding L2 protocols, e.g., Bluetooth, Z-Wave, or ZigBee, or IoT application protocols, e.g., MQTT and CoAP. In addition, log files of the IoT gateway are a fruitful source of information.
- *Unified format.* IoT covers a large range of various devices with specific functions and different parameters. To enable their monitoring it is not only necessary to get access to IoT data objects, but also transform the data into a unified well-defined and widely used format. Despite its age, the ASN.1 [24] is a flexible presentation layer language enabling to represent complex structured information. The ASN.1 is used for describing MIB objects in the SNMP ecosystem. It also easy to represent typical data objects in IoT devices, e.g., door/window sensors, thermometers, light sensors, heating regulators, etc. using ASN.1. This means to map a vendor-specific IoT device to representational MIB objects. Existing the ITU-T standard H.641 [25] defines MIB objects for general sensors to be used by a sensor network gateway that translates SNMP to a newly proposed sensor network management protocol. The standard defines two MIB objects for ZigBee network under a special MIB branch with prefix `itu-t(0).recommendations(0).h(8).h641(641).sensor-network-mgt(2).` However, the standard seems to be abandoned by IoT vendors. For this reason we propose a simple template for creating IoT MIB objects relevant to local IoT devices.
- *Continuous flow of IoT data.* IoT devices, namely sensors, constantly produce data readings. To observe several IoT data flows puts high demands on data processing and storage. Our approach separates the process of reading monitoring data as implemented by the IoT data extractor, see Fig. 2, from the presentation of MIB objects to the SNMP system. While the IoT extractor continuously observes IoT data and updates the corresponding MIB objects as necessary, the SNMP proxy agents provides monitoring data on demands as configured in the SNMP system. This reduces the monitoring load while keeping the actual values of IoT devices available to the monitoring system. In some case, the immediate reaction is necessary, which can be achieved using the SNMP Trap.
- *Application monitoring.* Integrated IoT environment monitoring provides similar information to traditional device network monitoring, such as information on IoT devices (address, status, availability), communication statistics (number of sent/received data), etc. Besides, it provides measured values like temperature, light intensity, power consumption. It offers the possibility to extend the monitoring with additional dashboards and analytical modules giving an insight into the controlled IoT environment.
- *IoT management.* The full IoT management also means the ability to configure IoT devices. The currently implemented SNMP proxy agent provides read-only data access. Device configuration and management can be realized by transforming SNMP SET commands to device-specific operations that reconfigure a device, e.g., turn a sensor on/off. We plan to research it in our future work.

IV. SNMP Proxy Agent

In this section, we introduce a general architecture of the proposed SNMP Proxy Agent. The approach is demonstrated on a smart home environment that utilizes MQTT communication and Home Assistant[1] as an IoT gateway that is the source of IoT events recorded in the log file. The idea of the unified SNMP interface for IoT monitoring is to provide information feeds for the existing network monitoring systems by implementing SNMP proxy agents for IoT devices and gateways. The current limitation of this approach is that the SNMP proxy agent does not support IoT device configuration.

The architecture of the SNMP proxy agent consists of the *IoT data extractor* that (i) reads IoT monitoring data from different local sources, namely locally captured communication and log files, (ii) extracts values of interest, and (iii) stores them in the IoT monitoring database. The second building block of the proxy agent is an *SNMP agent process*, which is a TCP server application that processes SNMP requests. After receiving the request, the agent process (i) retrieves the definition of the requested MIB object from a list of supported IoT MIB objects, (ii) queries the database for the current value, and (iii) replies with a MIB object and its value.

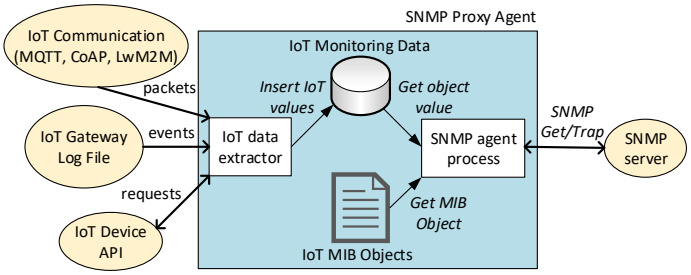

Fig. 2. Architecture of SNMP Proxy Agent

A. Extracting IoT monitoring Data

The first task is to obtain IoT readings, extract values of interest and store them in the local IoT objects database. IoT data format depends on the source (IoT management protocols, IoT gateway log files, IoT device API) and the type of an IoT device. We need to define an IoT data extractor with transformation rules that convert extracted data into ASN.1 for each data source and type.

We demonstrate the approach on (i) MQTT data and (ii) IoT events obtained from the Home Assistant log file.

1) Reading MQTT data: For IoT devices communicating using MQTT, we obtain monitoring values directly from PING and PUBLISH messages [6], [26]:

- The PING message (msgType=12) is regularly sent by a client (publisher) to inform a server (subscriber) that the client is alive. Although the message does not contain any measured value, it is used to update a timestamp of the last visibility of the device. PINGs can also reveal a new IoT device on the network.
- PUBLISH messages (msgType=3) are also sent by a client. The header contains the `Topic` which refers to an IoT object observed by IoT devices. For example, the light sensor contains *STATE* objects with values ON/OFF, *brightness*, *color* with RGB values, or *POWER*. Beside `Topic` header the PUBLISH message contains a `Message` header with the value related to the object. This can be a single value (in case of POWER object), or a composed value in JSON format (in case of STATE object, that includes *state, brightness, rgb*). Observing IoT values is important for IoT application monitoring.

The *IoT data extractor* retrieves source values and saves them with a timestamp to the database. Since SNMP does not support historical values by design, a new message updates the current value[2]. This helps to keep the database small, e.g., for each IoT object, there is only one entry.

2) Reading IoT events from the log file: When IoT devices communicate using L2 proprietary protocols, we rely on the IoT gateway. The gateway often creates a log file to record various device-related events or even a full log consisting of the data reading history. Suppose the *multilevel sensor* connected over Z-Wave technology to the Home Assistant. The Home Assistant logs every sensor activity into the log file, which in turn is observed by the *IoT data extractor*. The extractor retrieves data of interest and inserts them into the IoT monitoring database, as demonstrated below:

```
2019-11-03 09:18:30.380 Info, Node003, Received
  SensorMultiLevel report from node 3,
  instance 1, Temperature: value=85F
2019-11-03 09:20:29.134 Info, Node003, Received
  SensorMultiLevel report from node 3,
  instance 1, Temperature: value=84F
2019-11-03 09:20:29.882 Info, Node003, Received
  SensorMultiLevel report from node 3,
  instance 1, Luminance: value=11%
```

A simple parser detects *Temperature* and *Luminance* keywords in the log file, obtains values of interest, and puts them in the *IoT database*. Such a parser is required for all IoT protocols and log file formats that are to be supported.

B. Defining IoT MIB objects

SNMP monitoring works with MIB objects that are formally described by ASN.1 language and addressed by an object ID (OID) which refers to the specific MIB object of the IoT device. A MIB object definition provides an abstraction for IoT monitoring because monitoring data from IoT devices made by different vendors are represented using the same MIB object. This is similar to network monitoring, where MIB-2 defines a network interface using a single MIB object *ifEntry* [27]. This object provides information about the interface card regardless of vendor, type, or speed. Similarly, we create IoT MIB objects describing IoT devices regardless of the vendor or implementation. The ASN.1 description meets the unified format requirement as stated in Section III-B.

[1]See https://www.home-assistant.io/ (last access in January 2021)

[2]It is possible to define different update operations depending on the meaning of data, e.g., sum, average, etc.

IoT MIB objects can be described using either existing standardized or proprietary MIB objects. It is also possible to create a new IoT MIB definition with specific IoT device values. In both cases, the MIB with defined objects is uploaded into the NMS to correctly interpret and process IoT objects.

1) Existing MIBs: The existing standardized and proprietary MIBs already include definitions of many IoT-related objects like smart outlets, thermometers, sensors, see Table I.

OID	Group (Source)	Example
1.3.6.1.2.1.99.1	entitySensorMIB (IETF) [28]	entPhysSensorType,PhySensorValue, SensorOperStatus
1.3.6.1.4.1.9.9.91	ciscoEntitySensorMIB (Cisco)	entSensorType,entSensorStatus, entSensorValue
1.3.6.1.4.1.2.6.159.1.1.80	ibmSystemLMSensor (BM)	ibmSystemVoltageSensor, ibmSystemTemperatureSensor
1.3.6.1.2.1.229	energyObjectMib (IETF) [29]	eoPower, eoPowerOperState
1.3.6.1.4.1.318.1.1.10	environmentalMonitor (APC)	emsSmokeSensor,emsFluidSensor, emsDoorSensor
1.3.6.1.4.1.5528.100.4	netBotzSensors (netBotz)	tempSensor, humiSensor, cameraMotionSensor, doorSwitchSensor

TABLE I
MIBS WITH IOT RELATED OBJECTS

2) Creating a new IoT MIB: A new definition of IoT MIB objects can be created for a specific IoT device using ASN.1 language. The following example describes a multilevel sensor MIB object that observes temperature and luminance. Its definition may serve as a template for any IoT object.

```
multiSensorTable OBJECT-TYPE   -- OID: MySensor.1
  SYNTAX      SEQUENCE OF MultiSensorEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION "Table of multivalue sensors."
  ::= { MySensor 1 }

MultiSensorEntry ::= SEQUENCE {  -- OID: MySensor.1.1
  multiSensorIndex Integer32,       -- MySensor.1.1.1
  multiSensorSID   DisplayString,   -- MySensor.1.1.2
  multiSensorTemperature Integer32  -- MySensor.1.1.3
  multiSensorLuminance   Integer32, -- MySensor.1.1.4
}
multiSensorIndex  OBJECT-TYPE  -- OID: MySensor.1.1.1
  SYNTAX      Integer32
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION "A unique ID for each multilevel
               sensor."
::= { multiSensorEntry 1 }

multiSensorEntry  OBJECT-TYPE -- OID: MySensor.1.1
  SYNTAX      MultiSensorEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION "An entry related to a particular
               sensor."
  INDEX       { multiSensorIndex }
::= { multiSensorTable 1 }
```

The MIB object *multiSensorTable* is defined as a table of *MultiSensorEntry* objects using the SEQUENCE OF data type. Each *MultiSensorEntry* includes a set of single MIB objects *multiSensorIndex*, *multiSensorSID*, *multiSensorTemperature*, and *multiSensorLuminance*. Data type SEQUENCE OF defines an ordered set of multilevel sensor MIB objects that are accessible through the *multiSensorIndex*. This is a great benefit of ASN.1. When a new sensor is detected by an SNMP proxy agent, a new *MultiSensorEntry* is added to the *multiSensorTable* without SNMP agent re-configuration, see Table II. The pointer to the particular sensor is stored in

	SID MySensor.1.1.2	Temperature MySensor.1.1.3	Luminance MySensor.1.1.4
MultiSensorEntry	Node1	85	11
MultiSensorEntry	Node2	70	20

TABLE II
EXAMPLE OF MIB SENSOR OBJECTS.

multiSensorIndex object which is a part of the MIB OID. For example, to retrieve a temperature of the second multilevel sensor, the OID *MySensor.1.1.3.2* is used while the temperature of the first sensor is addressed by OID *MySensor.1.1.3.1*. User defined MIB objects can be registered under enterprise (1.3.6.1.4.1) or experimental (1.3.6.1.3) MIBs.

V. USE CASE: SMART BUILDING MONITORING

The IoT devices monitoring using SNMP was applied on the smart building environment as shown in Fig. 3.

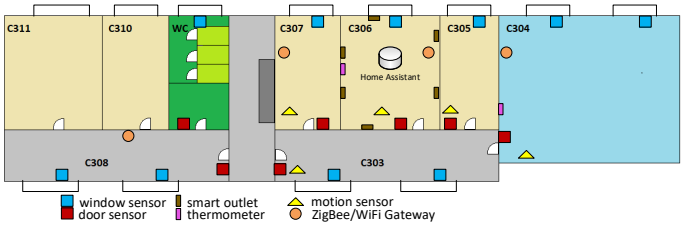

Fig. 3. IoT devices in the Smart Building

The deployment includes window and door sensors, motion sensors, smart outlets, and thermometers. IoT devices are connected to the Home Assistant either directly over WiFi or indirectly via ZigBee/WiFi gateway. Data from IoT devices is obtained by (i) reading the Home Assistant log file or by (ii) observing MQTT communication between smart outlets and the gateway. Altogether, we employed 29 IoT sensors of five types that required the definition of five new MIB objects [30]. In our experiments, we observed frequency of logs and MQTT messages processed by the IoT data extractor. The results of one-day MQTT communication for three MQTT-enabled sensors is given in Table III.

Sensor	Messages	Size (kB)	Pings	Publish
Thermometer	6.861	1 829	30 sec	30 sec
Multisensor	13.214	3170	10 sec	10 sec
Smart outlet	1.440	161	30 sec	10 sec

TABLE III
MQTT TRANSMISSION BY IOT SENSORS (1 DAY).

We notices, that IoT events in the Home Assistant log file or in MQTT messages appear every 10 or 30 seconds with keep-alive messages or state updates. This is useful for network state monitoring. Nevertheless, application values (temperature, alarm state, etc.) change with less frequency depending on the type of IoT devices. Using SNMP proxy

agent we can significantly reduce monitoring transmissions by polling SNMP agent every five minutes. In case of critical value change, a SNMP Trap message can be sent to the NMS.

The SNMP proxy agent was implemented using Net-SNMP library. It extracts data from both MQTT and the Home Assistant as described in Sec. IV-A. SNMP monitoring data are visualized by CloudView NMS, see Fig. 4.

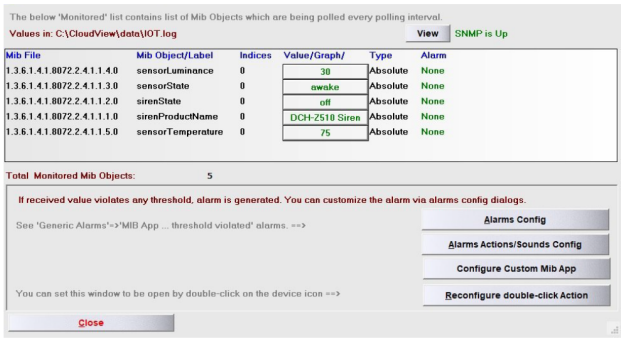

Fig. 4. Integrated IoT monitoring into SNMP

VI. Conclusion

In this paper, we proposed to unify IoT device monitoring and real-time data gathering by employing SNMP. SNMP proxy agents are deployed on IoT gateway devices or in a suitable network location with access to IoT traffic to achieve this. The newly implemented SNMP proxy agent collects IoT monitoring data from various sources, transforms data into MIB objects, and provides it via a standard SNMP interface.

The approach's main advantage is a unified view on connected IoT and non-IoT devices regardless of their communication protocol. The data can be visualized in the network monitoring system's dashboard or further processed by dedicated application monitoring systems, smart building management software, or security and diagnostic systems.

Our future work will focus on management of IoT devices using SNMP SET, the efficient analysis of gathered data as a part of IoT application monitoring, and protection of the privacy of collected IoT data.

Acknowledgment

The work is supported by the Brno University of Technology project "Application of AI methods to cyber security and control systems", no. FIT-S-20-6293. The authors thank Patrik Krajč and Kateryna Polishchuk for IoT testbed experiments.

References

[1] D. Hanes, G. Salqueiro, P. Grossetete, R. Barton, and J. Henry, *IoT Fundamentals. Networking Technologies, Protocol and Use Cases for the Internet of Things.* Cisco Press, 2017.

[2] D. Harrington, R. Presuhn, and B. Wijnen, "*An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*," IETF RFC 3411, December 2002.

[3] B. Claise, "*Cisco Systems NetFlow Services Export Version 9*," IETF RFC 3954, October 2004.

[4] R. Gerhards, "*The Syslog Protocol*," IETF RFC 5424, March 2009.

[5] Z. Shelby, K. Hartke, and C. Bromann, "*The Constrained Application Protocol (CoAP)*," IETF RFC 7252, June 2014.

[6] IEC, "Information technology – Message Queuing Telemetry Transport (MQTT)," International Organization for Standardization, Standard ISO/IEC 20922:2016, June 2016.

[7] F. Chen, J. Dai, B. Wang, S. Sahu, M. Naphade, and C.-T. Lu, "Activity Analysis Based on Low Sample Rate Smart Meters," in *Proceeding of the 17th ACM Conference KDDM*, New York, 2011, p. 240–248.

[8] M. A. Lisovich, D. K. Mulligan, and S. B. Wicker, "Inferring Personal Information from Demand-Response Systems," *IEEE Security Privacy*, vol. 8, no. 1, pp. 11–20, 2010.

[9] C. Welch, "Comcast is reportedly developing a device that would track your bathroom habits," 2019, *https://www.theverge.com/2019/5/21/18634466*.

[10] "Recommendations 01/2020 on measures that supplement transfer tools to ensure compliance with the EU level of protection of personal data," European Data Protection Board, Tech. Rep. R01/2020, 2020.

[11] "Recommendations 02/2020 on the European Essential Guarantees for surveillance measures," European Data Protection Board, Tech. Rep. R02/2020, 2020.

[12] N. Havard, S. McGrath, C. Flanagan, and C. MacNamee, "Smart Building Based on Internet of Things Technology," in *2018 12th International Conference on Sensing Technology (ICST)*, 2018, pp. 278–281.

[13] H. Chen, P. Chou, S. Duri, H. Lei, and J. Reason, "The Design and Implementation of a Smart Building Control System," in *2009 IEEE Int. Conference on e-Business Engineering*, 2009, pp. 255–262.

[14] H. Lindholm-Ventola and B. Silverajan, *CoAP-SNMP Interworking IoT Scenarios*, ser. Tampere University of Technology. Department of Pervasive Computing. Report, 2014, no. 3.

[15] M. Savić, "Bridging the SNMP gap: Simple network monitoring the internet of things," *Facta universitatis - series: Electronics and Energetics*, vol. 29, pp. 475–487, 01 2016.

[16] J. Han and S. Oh, "A study of IoT home network management system using SNMP," *International Journal of Control and Automation*, vol. 11, pp. 163–172, 05 2018.

[17] A. Sehgal, V. Perelman, S. Kuryla, and J. Schonwalder, "Management of resource constrained devices in the internet of things," *IEEE Communications Magazine*, vol. 50, no. 12, pp. 144–149, 2012.

[18] H. Choi, N. Kim, and H. Cha, "6LoWPAN-SNMP: Simple Network Management Protocol for 6LoWPAN," in *2009 11th IEEE International Conference on High Performance Computing and Communications*, 2009, pp. 305–313.

[19] ITU-T, "*TMN Management Functions*," M.3400, February 2000.

[20] B. Claise, B. Trammel, and P. Aitken, " Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information," IETF, RFC 7011, September 2013.

[21] J. Guth, U. Breitenbücher, M. Falkenthal, P. Fremantle, O. Kopp, F. Leymann, and L. Reinfurt, *A Detailed Analysis of IoT Platform Architectures: Concepts, Similarities, and Differences*, Singapore, 2018.

[22] S. Sinche, D. Raposo, N. Armando, A. Rodrigues, F. Boavida, V. Pereira, and J. S. Silva, "A Survey of IoT Management Protocols and Frameworks," *IEEE Comm. Surveys Tutorials*, vol. 22, no. 2, 2020.

[23] J. de C. Silva, J. J. P. C. Rodrigues, J. Al-Muhtadi, R. A. L. Rabêlo, and V. Furtado, "Management Platforms and Protocols for Internet of Things: A Survey," *Sensors*, vol. 19, 2019.

[24] O. Dubuisson and P. Fouquart, *ASN.1: Communication Between Heterogeneous Systems*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001.

[25] ITU-T, "*SNMP-based sensor network management framework*," H.641, 2012.

[26] O. Ryšavý and P. Matoušek, "An IPFIX Extension for MQTT Protocol Monitoring," FIT BUT, Tech. Rep. FIT-TR-2019-01, 2019.

[27] K. McCloghrie and M. T. Rose, "*Management Information Base for Network Management of TCP/IP-based internets:MIB-II*," IETF RFC 1213, March 1991.

[28] A. Bierman, D. Romascanu, and K. Norseth, "*Entity Sensor Management Information Base*," IETF RFC 3433, December 2002.

[29] M. Chandramouli, B. Claise, B. Schoening, J. Quittek, and T. Dietz, "*Monitoring and Control MIB for Power and Energy*," IETF RFC 7460, March 2015.

[30] P. Matoušek and P. Krajč, "Monitoring of IoT Devices Using SNMP," Tech. Rep. IT-TR-2020-10, 2021.