An analysis of the TR069 (CWMP) protocol

Ilija Basicevic
Faculty Of Technical Science
University Of Novi Sad
Novi Sad, Serbia
ilibas@uns.ac.rs

***Abstract*—This paper presents a short analysis of the TR-069 (CWMP) protocol. The protocol is widely used in consumer electronics and the Internet of Things (IoT), and this paper is an attempt to explain its success. The purpose of the protocol, its architecture, integration with other protocols, and the technologies that it relies on are discussed. A parallel is drawn with some other management protocols, primarily the Simple Network Management Protocol (SNMP), where possible.**

Keywords*—*device management, TR-069, remote monitoring, Internet management

I. INTRODUCTION

TR-069 is an established protocol for device configuration and monitoring in the consumer electronics field. As this field has grown significantly in the last two decades, the spectrum of devices that can be managed by TR-069 has also widened. For example, TV operator probably uses this protocol to monitor set top box at the reader's home. The protocol is published by the Broadband Forum [1]. The first version appeared in 2004 and the current version is 1.4.

The two sides in the TR-069 communication are CPE (Customer Premises Equipment) and ACS (Auto-Configuration Server), see Fig. 1. CPE is a managed device that contains at least one CWMP endpoint and is located in the end user's network. The ACS performs auto-configuration and other management functions on the CPE over the broadband connection. The ACS is located in the operator network or datacentre.

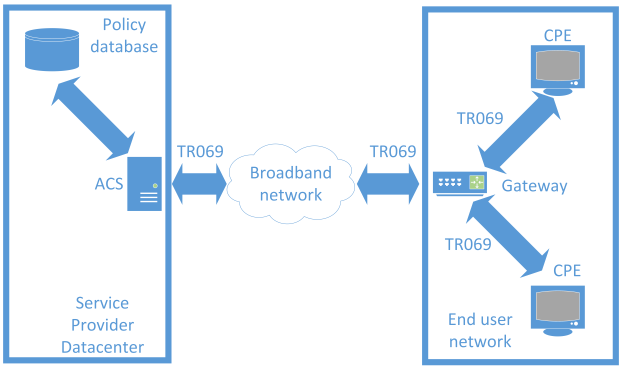

Fig. 1: TR069 network topology.

The protocol integrates the following functions: auto-configuration and dynamic provisioning, management of software and firmware, monitoring of device state and performance, and diagnostics. The protocol enables an ACS to provision a CPE (or a set of CPEs) based on different criteria. It also allows for the control of downloads of the CPE software/firmware image files. The download can be initiated by both sides. There are also mechanisms to manage modular software and execution environment on a CPE such as installation, update, uninstallation and inventory of software modules. The TR-069 protocol includes mechanisms which enable an ACS to monitor the status and performance statistics of a CPE. Also, it is important that this protocol enables a CPE to send information to an ACS which makes it possible for the ACS to diagnose and resolve different operational issues. The last feature is used when the aforementioned set top box malfunctions and the reader calls the operator's call center for technical support.

II. RELATED WORK

The Internet standard for device management and monitoring is SNMP [2]. Thus, the TR-069 features overlap with those of the SNMP, but TR-069 has been developed two decades later and consequently uses more modern technologies than SNMP, such as HTTP, XML, and SOAP. It is also less general than SNMP, as it is custom tailored for scenarios in consumer electronics systems. Its architecture is simpler compared to SNMP, which is probably one of the reasons for its success. Both TR-069 and SNMP are application level protocols. That fact, among other things, enables protocol operation (e.g. device management), when the managed device and the management agent are not in the same network (there can be several hops between them). (Early WAN management protocols have been situated on data link level, which has been possible in the environment with uniform network technology - levels 1 and 2 [3]).

Network Configuration Protocol (NETCONF) is a network management protocol developed by IETF. The current version is published in [4] and has the status of an Internet standard. The protocol has been developed as a result of a search in IETF for a device network configuration solution (see [5]), as it has been concluded that SNMP serves well for device monitoring, but for device configuration the network operators often use proprietary solutions.

YANG [6] is a data modelling language used for modelling configuration and state data. It is popular today and supports retrieving information using NETCONF and RESTCONF [7]. The language provides descriptions of network nodes and their interactions. In combination with NETCONF, YANG is a tool that administrators can use to automate configuration tasks in a heterogeneous network.

Open Mobile Alliance Device Management (OMA DM) [8] is a device management protocol designed for the management of mobile devices such as mobile phones, PDAs and tablet computers. The protocol is specified by OMA Device Management working group as well as the Data Synchronization working group. It supports the following operations: provisioning, data configuration, software upgrade and fault management.

OMA Lightweight Machine to Machine (LwM2M) [9] is a protocol from OMA alliance for machine to machine and IoT device management. Originally it used Constrained Application Protocol (CoAP) for transport but later versions included support for other protocols (UDP, TCP, TLS, Message Queueing Telemetry Transport - MQTT).

III. THE PROTOCOL STACK

The TR-069 protocol stack is presented in Fig. 2. The foundation is TCP/IP protocol, which provides transport functionality. (As mentioned later, for notification purpose and for sending the connection request, TR-069 can use UDP, too). Above TCP/IP is SSL/TLS which provides security. Residing on SSL/TLS is HTTP which provides client/server transactions that are the building elements of the TR-069 dialogue, corresponding to the session layer of the ISO OSI stack. Above HTTP is SOAP which provides transfer syntax – thus positioning it it in the presentation layer. Above SOAP are the remote procedure call (RPC) methods. Having in mind the ISO OSI application level architecture, the RPC methods would correspond to a Specific Application Service Element (SASE) in ISO OSI. On the top is the TR-069 management application (CPE or ACS) - the user application process. Thus the application is according to the ISO OSI in the local system environment (LSE), and the stack below is in the OSI environment (OSIE).

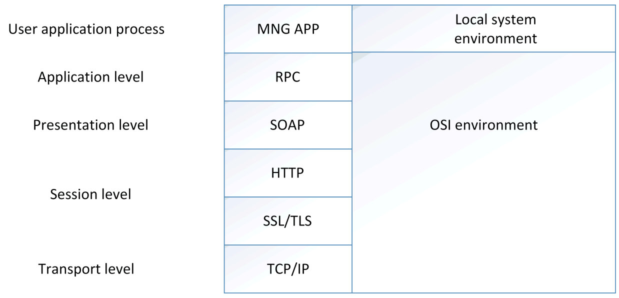

Fig. 2: TR-069 protocol stack.

IV. INTEGRATION WITH OTHER PROTOCOLS

Requests and responses are sent between a CPE and an ACS using HTTP or HTTPS, in the form of remote procedure calls (RPC), while SOAP 1.1 is used as the transfer syntax. Both CPE and ACS can take the role of client or server, but an interesting protocol design decision is that the communication is always started by a CPE. When needed, an ACS can send an asynchronous connection request which will result in the CPE reporting to the ACS, though. This is used when near-real-time reconfiguration of a CPE is required. For example, if the user subscribes to a service, that service becomes immediately accessible to them, without waiting for the next periodic contact. (But in that case, the TR-069 session is also started by the CPE.)

In certain cases, it is not possible for an ACS to send a connection request to a CPE directly. If a CPE is behind a firewall or a gateway, which prevents direct communication from an ACS to a CPE, TR-069 makes use of the XMPP protocol. Both the ACS and the CPE register on the XMPP server, and then the ACS can use it to relay the connection request (in the form of XMPP IQ Stanza) to the CPE. The CPE and the ACS do not have to be registered to the same XMPP server, it can be a cluster of XMPP servers instead.

Alternatively, if XMPP is not available, TR-069 can use Universal Plug and Play Internet Gateway Device framework (UPnP IGD). Using that framework a CPE can obtain a WAN IP address and add a port mapping on the gateway to enable traversal of the connection request from the ACS.

Still another possibility is the use of STUN, when a CPE is behind a NAT gateway. This is another case when an ACS cannot send the asynchronous connection request to an CPE (see also [10]). The standard prescribes the following procedure. When a CPE discovers (using STUN) that it is behind a NAT gateway with private addresses, it is required to keep open a NAT binding (by sending periodic STUN binding requests, based on the binding timeout it has discovered previously), and send to the ACS the public IP address and port associated with the binding. The ACS sends UDP Connection Request to the CPE using that address information when required.

In these cases, the gateway or firewall does not have to be managed by TR-069.

TR-069 can be integrated with DHCP, in a way that in the context of DHCP Offer message from the DHCP server, the CPE receives not only the new address it will use but also the address of the ACS server. Also, when sending the Inform message, the CPE can include the identification of the DHCP server, which can be checked later in communication between the ACS and the managed gateway. It is recommended that when a CPE device reports for the first time it is connected to a specific gateway that the ACS cross-checks that information either by explicitly soliciting the information from the gateway, or by using its own table - in the case when it previously subscribed for event information from that gateway. The next time, and any subsequent time when the CPE reports that it has been connected to that gateway, the cross check is not necessary.

TR-069 has a proxy feature that further extends its applicability by allowing non-TR-069 devices to be managed by an ACS through a TR-069 CPE acting as a proxy. Two types of proxied devices are distinguished. In the case when simple ones, such as power switches and binary sensors are used, a CPE proxy contains an embedded object in its data model. More complex devices, such as routers, and set top boxes, are presented by having a TR-069 CPE device in the CPE proxy.

TR-069 communication is secured using the TLS protocol (as is the case with SNMPv3 and LwM2M as well). This protocol is also used for authentication. If TLS authentication is not used, an ACS must authenticate a CPE using HTTP authentication. If TLS is used for encryption, an ACS should use the basic authentication scheme whereas if TLS is not used, an ACS must use digest authentication. The ACS chooses the authentication scheme by providing a basic or digest authentication challenge.

Although two sides authenticate each other, the operators can additionally restrict the IP addresses from which a connection request can reach the CPE, by using a firewall or setting the routing table. Since linux operating system is common in this field, iptables can be used for this restriction.

V. COMMUNICATION LIFE-CYCLE

The life cycle of a TR-069 communication between the two endpoints is as follows. The first session is initiated by the CPE. We can say that the CPE reports for the first time to the ACS. The CPE uses the ACS URL it obtained through the configuration, or through DHCP address allocation (explained earlier in the text). In the first session, the CPE sends to the ACS the management URL (in the ManagementServer.ConnectionRequestURL parameter). The ACS can program future communication (using the ScheduleInform). Also, the ACS can afterwards at any time use the CPE management URL to send the connection request. If the CPE management URL changes, it is mandatory for the CPE to report that to the ACS.

The MSC of an example TR-069 session is given in Fig. 3. It can be seen that the CPE opens the TCP connection to the ACS. The SSL initiation follows. After that, the CPE sends Inform request in the HTTP Post method. The ACS sends an Inform response in the HTTP response. After the TR-069 session has thus started, the CPE sends an empty HTTP Post message, only to make it possible for the ACS to send a TR-069 request, in an HTTP response. In this example, the ACS sends a GetParameterAttributes request. In the next step, the CPE sends a TR-069 GetParameterAttributes response, in an HTTP Post message. It can be seen how HTTP and TR-069 transactions overlap - the CPE sends an HTTP Post request carrying TR-069 GetParameterAttributes response, and the ACS sends an HTTP response carrying the next TR-069 request, and so on. In the last step, the ACS sends an empty HTTP response, which is a signal to the CPE to close the TCP connection (and the TR-069 session).

To maintain the session state, the ACS uses a session cookie. It is recommended that for a sequence of transactions that comprise a single session, the CPE maintains a single TCP connection that persists throughout the duration of the session.

TR-069 can use port 7547 which has been assigned by IANA for the CPE WAN management protocol.

VI. THE PROTOCOL SYNTAX

When it appeared, SNMP inherited the fetch-store paradigm from an earlier Internet management protocol High-Level Entity Management System (HEMS) [11]. It was a novel paradigm in the field of device management protocols [3]. Instead of having a specific request type for each operation type, it has a set of generic requests and responses – primarily for getting and setting the values of data model parameters. The specific operation that will be initiated depends on the variable onto which the generic request is applied. For example, if a device reboot is required, a set request with the value 0 would be applied to the variable which contains the time till the reboot, and consequently, reboot will immediately be started. This paradigm results in a protocol design that is extendable and stable, with a small set of commands. Consequently, the command set of SNMP comprises the generic commands, such as GetRequest for retrieving a value, GetNextRequest for iterative retrieval, SetRequest for setting the value of a parameter, Trap, and InformRequest which are used for event notification, and Response which is a response to any of the aforementioned commands, except Trap for which no response is expected.

On the other hand, in [5] it has been noted that there is a semantic mismatch between the data centric model of SNMP and the task oriented model which is preferred by human operators. This resulted in the development of a non-trivial code which performs mapping between these two models and is integrated into management applications.

The TR-069 has a kind of hybrid model. There is a set of generic commands: Get/SetParameterValues, Get/SetParameterAttributes, Add/DeleteObject, but it also contains commands for some specific operations, such as Reboot, FactoryReset, Download, ScheduleDownload or Upload.

A CPE connects to the ACS on initial installation, after reset, after a defined timeout, by schedule defined by the ACS in ScheduleInform, upon a connection request from the ACS, upon a change of the ACS URL, when the values of parameters for which the ACS requested monitoring have been changed by a third party, upon the termination of file up/download – if requested in the CPE configuration.

The CPE-ACS session starts with the CPE sending an Inform message. It contains connection reason, current time stamp on the CPE, the number of attempts, and the values of parameters for which ACS requested monitoring, and which were modified by a third party since the last connection.

The baseline data model template is given in the TR-106 standard [12], and it is specified in XML (YANG and OMA DM are also XML based). There are additional standards, specific to the field of the consumer system (e.g. TR-135 [13] for set top boxes). Also, as noted, TR-069 makes use of SOAP, and SOAP is XML based. On the other hand, in the case of SNMP, the standard for data is MIB (Management Information Base), which specifies that the information is encoded using ASN.1. An important difference between these two approaches is that the XML model in TR-069 and YANG is readable by humans, while ASN.1 binary coded data are not. A similar differentiation existed in the 2000s in the VoIP world [14], at the time of market competition between H.323 and SIP protocols, as H.323 uses ASN.1, while SIP messages contain text coded in UTF-8 and are based on HTTP.

TABLE I: TR-069 Methods

Mandatory Methods	
CPE Methods	
	GetRPCMethods GetParameterValues SetParameterValues GetParameterAttributes SetParameterAttributes AddObject DeleteObject Download Reboot
ACS Methods	
	Inform TransferComplete AutonomousTransferComplete
Optional Methods	
CPE Methods	
	ScheduleInform Upload FactoryReset GetAllQueuedTransfers ScheduleDownload CancelTransfer ChangeDUState
ACS Methods	
	RequestDownload DUStateChangeComplete AutonomousDUStateChangeComplete

VII. THE PROTOCOL METHODS

TR-069 methods are given in Table I.

A. Mandatory Methods

GetRPCMethods is a method that can be called either by a CPE or an ACS, to discover the set of RPC methods supported by the other side in the communication.

a) CPE Methods: The CPE methods can be used by an ACS to perform different operations, such as obtaining and modifying values of parameters on CPE (GetParameterValues and SetParameterValues), discovering which parameters are existing on a CPE (GetParameterNames), reading and obtaining attributes (which refer to the access control information including notification type) associated with specific parameters on a CPE (GetParameterAttributes and SetParameterAttributes).

There are also methods that are used to create or delete an instance of a multi-instance object (AddObject and DeleteObject). For example, if there is an object Top.Group.Object, AddObject can be used to create a new parameter Top.Group.Object.i.Parameter, where i is the return value of AddObject method. Afterwards, Top.Group.Object.i. can be passed as an argument to DeleteObject, to delete that instance of the object.

Download method is used by the ACS to cause the CPE to download a specified file from the designated location.

Reboot method causes the CPE to reboot.

b) ACS Methods: We have already mentioned the Inform method which a CPE must call to initiate a transaction sequence whenever a session with an ACS is established. Whenever a file transfer requested by the ACS (by an earlier Download, ScheduleDownload or Upload method call) is completed, the CPE calls the TransferComplete method. Whenever a file transfer which was not requested by the ACS is completed, the CPE calls the AutonomousTransferComplete method.

B. Optional Methods

a) CPE Methods: An ACS can request from a CPE to send an Inform method sometime in the future (independent from periodic Inform method calls) using ScheduleInform. Upload method is used by the ACS to request the CPE to upload a specified file to the specified location. FactoryReset method is used by the ACS to reset the CPE to its factory default state.

When necessary an ACS can request from a CPE to send the status of all queued downloads and uploads, including autonomous transfers, using GetAllQueuedTransfers method.

ScheduleDownload is an advanced version of Download method, as it allows an ACS to specify one or two time windows in which download should be performed. File transfers requested by the ACS (using Download, ScheduleDownload or Upload method call) can be cancelled with CancelTransfer method.

An ACS can request installing a new Deployment Unit (DU), updating an existing DU, or uninstalling an existing DU, using ChangeDUState method.

b) ACS Methods: Using the RequestDownload method, a CPE can request a file download from the ACS, which may result in the ACS calling the Download method to initiate the download.

There are also two methods related to the DU state changes, which are used by a CPE to inform the ACS about the completion (successful or unsuccessful) of a DU state change. DUStateChangeComplete method is used if the state was requested by an ACS, and AutonomousDUStateChangeComplete if it was not.

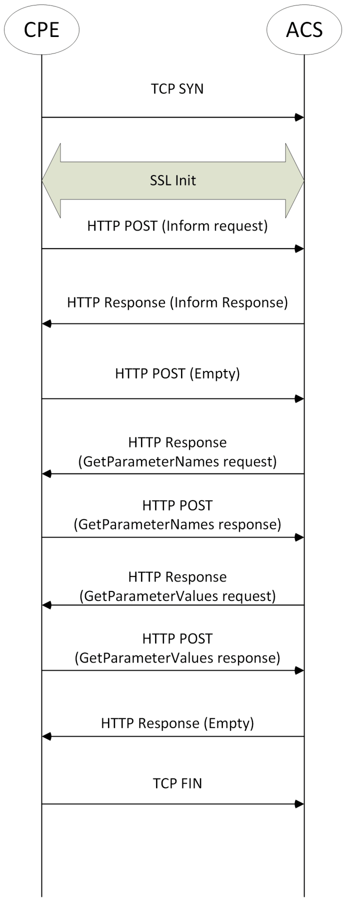
CPE
ACS
TCP SYN
SSL Init
HTTP POST (Inform request)
HTTP Response (Inform Response)
HTTP POST (Empty)
HTTP Response
(GetParameterNames request)
HTTP POST
(GetParameterNames response)
HTTP Response
(GetParameterValues request)
HTTP POST
(GetParameterValues response)
HTTP Response (Empty)
TCP FIN

Fig. 3: Message sequence chart of a TR-069 session.

VIII. MECHANISMS FOR MONITORING OF CPE PARAMETERS

There are several levels of notification on a CPE, which is defined for each parameter. The notification can be disabled, or it can be passive, active, lightweight passive or lightweight active. Thus, lightweight notification can be combined with active/passive notification as well. Passive notification means that when a parameter value is changed, the new value will be sent to the ACS in the next Inform which is sent (and which was scheduled earlier or is a periodic request). Active notification means that when a parameter value changes that would cause the establishment of a new session and sending of the new parameter value in the Inform. Lightweight notification is a UDP based mechanism which is not reliable, and complements the existing notification mechanisms.

Two message types are defined in the SNMP protocol for asynchronous event notification. One is Trap, used for reporting without confirmation, and the other is Inform-Request for confirmed notification delivery.

IX. THE PERSPECTIVES FOR TR-069 IN TODAY'S COMPUTING WORLD

As noted, TR-069 has already claimed an important position in the consumer electronics field in the previous period. The use of this protocol in the management of set top box devices was mentioned throughout the text, see also [15], [16] for more about this topic. Refs. [17] and [18] present an implementation of the protocol.

In a race with some other protocols (e.g. OMA-DM [8], LwM2M [9]) TR-069 is more and more often becoming the protocol of choice for device management in IoT systems ([19], [20], [21]) which is very important for the future of this protocol. An important segment of IoT is the smart home gateway, also named Machine-Type Communication Gateway (MTCG). MTCG acts as a bridge between smart objects and the Internet. Due to the significant expansion of the IoT domain, a separate configuration of each device is not viable anymore, as the number of devices is increasing. This is the place in IoT architecture where TR-069 can fit. There are also proposals for its use in LTE, [22], and Wi-Fi, [23]. In [23] a method is presented which ensures that all devices in a Wi-Fi mesh network have a consistent firmware version, which improves the stability of the system.

On the other hand, following the success of TR-069 and the sophistication of consumer network, the Broadband Forum came with its successor TR-369, [24] which allows for more complex communication scenarios (whereas in TR-069 there is one ACS that controls the CPEs, in TR-369 there can be several controllers controlling one device). TR-369 is sometimes referred to as TR-069 for IoT devices. The ability to have several controllers with different permissions controlling one device enables multiple providers, vendors and end-users to interact with the managed devices. Another difference from TR-069 is that TR-369 can be based on other protocols (Websockets, COAP, MQTT, Simple Text Oriented Messaging Protocol -STOMP), not just HTTP.

Also, an effort has been made to make less overhead in terms of communication cost, as for precise monitoring, a larger number of messages is required which incurs a network cost. With TR-369 more precise monitoring can be achieved and with less impact on network traffic than is the case with TR-069, as it is more lightweight in terms of communication cost. Among other elements, the number of communication handshakes has been reduced. The communication model used in TR-069 is based on the idea that a CPE would open a session to the ACS when required, exchange data and close the session as soon as possible. On the other hand, in TR-369, once opened a session remains open all the time.

X. CONCLUSION

This paper contains a short analysis of the TR-069 protocol, with respect to the protocol architecture, position in the protocol stack, and its reliance on the use of other Internet protocols. With respect to the protocol design, a parallel is made to the SNMP protocol, which is the most widely known Internet management protocol.

Compared to SNMP, TR-069 uses more recent protocols and technologies, such as HTTP, SOAP and XML. Both protocols are application-level protocols. While TR-069 messages are coded in XML and readable by humans, SNMP messages are binary coded in ASN.1 and not readable by humans.

In the Internet ecosystem, there can be obstacles to the bidirectional communication between a CPE and an ACS. The standard acknowledges this situation and provides several mechanisms to overcome it.

The TR-069 protocol achieved significant success in the consumer electronics field due to a good custom tailored design and the use of relevant contemporary technologies, which among other allow for secure operation. The protocol entered the IoT field as well. However, its successor TR-369 carries some improvements that make it even more suitable for IoT applications. For many scenarios though, TR-069 is still the protocol of choice.

Acknowledgment

The author would like to thank Marko Vucicevic for providing advice on the integration of the TR-069 protocol.

This work was partially supported by the Secretary of Science and Technology Development of Vojvodina Province under Grant 142-451-2406/2022-01/01.

References

[1] "TR-069 CPE WAN Management Protocol," 2020.

[2] J. Case, R. Mundy, D. Partain, and B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework," 1999, RFC: 2570.

[3] D. E. Comer, *Internetworking with TCP/IP Vol.1: Principles, Protocols and Architectures*. Upper Saddle River, NJ: Prentice Hall, 2000.

[4] R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman, "Network Configuration Protocol (NETCONF)," 2011, RFC: 6241.

[5] J. Schoenwaelder, "Overview of the 2002 IAB Network Management Workshop," 2003, RFC: 3535.

[6] M. Bjorklund, "The YANG 1.1 Data Modeling Language," 2016, RFC: 7950.

[7] A. Bierman, M. Bjorklund, and K. Watsen, "RESTCONF Protocol," 2017, RFC: 8040.

[8] "OMA Device Management V2.0," 2012.

[9] "OMA Lightweight M2M (LwM2M) V1.2," 2020.

[10] M. Savić, I. Papp, I. Rešetar, D. Majstorović, and D. Spasojević, "Implementation of server push mechanism based on tr-069 and tr-111 protocols for network device management," in *2013 21st Telecommunications Forum Telfor (TELFOR)*, 2013, pp. 361–364.

[11] C. Partridge and G. Trewitt, "THE HIGH-LEVEL ENTITY MANAGEMENT SYSTEM (HEMS)," 1987, RFC: 1021.

[12] "TR-106 – Data Model Template for CWMP Endpoints and USP Agents," 2022.

[13] "TR-135 Data Model for a TR-069 Enabled STB," 2012.

[14] I. Basicevic, M. Popovic, and D. Kukolj, "Comparison of sip and h.323 protocols," in *Proceedings of the 2008 The Third International Conference on Digital Telecommunications*, ser. ICDT '08. USA: IEEE Computer Society, 2008, p. 162–167.

[15] B. Trifunović, V. Mihailović, N. Ignjatov, R. Simikić, and I. Velikić, "Android development framework for tr-069-based services," in *2014 IEEE Fourth International Conference on Consumer Electronics Berlin (ICCE-Berlin)*, 2014, pp. 492–494.

[16] D. Knezevic, I. Ostojic, I. Papp, and M. Savic, "Integration mechanism for live stream qos monitoring in android-based iptv set-top box," in *2014 IEEE Fourth International Conference on Consumer Electronics Berlin (ICCE-Berlin)*, 2014, pp. 489–491.

[17] V. Mihailović, N. Ignjatov, M. Bojan, and T. Nikola, "Adaptive build system for tr-069 consumer device agent," in *2014 IEEE Fourth International Conference on Consumer Electronics Berlin (ICCE-Berlin)*, 2014, pp. 314–317.

[18] M. Ćetkovic, N. Nemet, T. Samardžić, and N. Teslić, "Auto-configuration server architecture with device cloud cache," in *2014 IEEE Fourth International Conference on Consumer Electronics Berlin (ICCE-Berlin)*, 2014, pp. 296–298.

[19] Y. Chahid, M. Benabdellah, and A. Azizi, "Internet of things protocols comparison, architecture, vulnerabilities and security: State of the art," in *Proceedings of the 2nd International Conference on Computing and Wireless Communication Systems*, ser. ICCWCS'17. Association for Computing Machinery, 2017.

[20] L. Reinfurt, U. Breitenbücher, M. Falkenthal, F. Leymann, and A. Riegg, "Internet of things patterns," in *Proceedings of the 21st European Conference on Pattern Languages of Programs*, ser. EuroPlop '16. New York, NY, USA: Association for Computing Machinery, 2016.

[21] M. Štůsek, P. Masek, D. Kovac, A. Ometov, J. Hosek, F. Kröpfl, and S. Andreev, "Remote management of intelligent devices: Using tr-069 protocol in iot," 06 2016.

[22] M.-Y. Wu, Y.-H. Lin, T.-H. Tseng, C.-M. Hsu, K.-S. Hsu, and H.-C. Young, "A qos monitoring system for lte small cells," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, 2018, pp. 1–4.

[23] K. S. Huang, "An enhanced tr-069 firmware upgrade method of wi-fi mesh system," in *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, 2019, pp. 655–659.

[24] "TR-369 – User Services Platform (USP)," 2022.