Onvifsense: ONVIF Network Device Accessibility Application

Yogiraj Awati¹, Abhijeet Bhintade², Rahul Gutal³, Saurabh Taware⁴
¹,²,³,⁴ Smt. Kashibai Navale College of Engineering, University of Pune

Abstract—In the world of disparate network devices and diverse standards for communication has led to limitations in interoperability between various networking interfaces. Open Network Video Interface (ONVIF) is committed to setting standards for the communication of network endpoint devices and achieve a coherent exchange of information between them. OnvifSense is a proposed system framework which manages all ONVIF compatible devices available in the network. ONVIF devices support WS-Discovery. This paper addresses OnvifSense as a central node application where multiple clients will query for particular services related to ONVIF device. Simultaneously it will discover devices in the network and respond to respective clients with multiple attributes of device viz. location, name, capabilities and functionalities such as local storage, supporting video/audio format etc.

Keywords—ONVIF, Web Service, SOAP, XML, WS-Security, WS-Discovery.

I. INTRODUCTION

The ONVIF specification defines a common protocol for the exchange of information between IP-based video surveillance devices. This was founded by Bosch Security Systems, Axis Communications and Sony Corporation in 2008. This specification elaborates a network model, communication at interfaces and data structure for message exchanges [1]. Network layer IP security for web services is provided by this standard. The construct of IP-Video devices vary in technical configurations which are dependent on the vendor, video compression/encoding designs, compatible network protocols etc. In such cases interoperability faces major issues and restrictions. To solve this issue ONVIF standard was set with an underlying platform as web services [2]. It concentrates on the automatic device discovery, Meta data transfer with integrity which uses core technologies like Web services, Real Time Streaming Protocol (RTSP). This standard supports services like video analytics, audio control and replay of recordings, real time video/audio streaming [4]. In this paper attempt has been made to introduce the novel concept of OnvifSense application.

II. MOTIVATION

To our knowledge various softwares are available in market to interact with NVTs in network. Existing softwares focuses on a single client interacting with a single device. They don’t provide a facility where a client interacts simultaneously with multiple devices and hence one-to-many relationship is not achieved. In this context this is a prominent issue of providing large-scale device interoperability and flexibility among network devices in large scale organisations. Here an attempt has been made to provide a solution where a single client interacts with many devices through OnvifSense application which helps in accomplishing one-to-many relationship.

III. BASICS OF ONVIF

A. ONVIF implied Services

Every ONVIF compatible device should support at least one of the following services such as 1) Network Video Storage (NVS) – defines the structure for streamed media and rules for accessing it. 2) Network Video Transmitter (NVT) - provides one or more video streams from distinct sources. 3) Network Video Analytics (NVA) – delivers the function of analysis of videos and audios 4) Network Video Display (NVD) – represents graphical representation between device and human operator [3].

B. Web Service Implementation

Web Service Description Language (WSDL) [6] is used to describe services on ONVIF devices. The benefit of this is that various devices can interact irrespective of manufacturers. The communication between client and device takes place over SOAP protocol [9]. Platforms on which SOAP can be enforced are RSTP, HTTP and HTTPS [3]. XML is the foundation of SOAP and defines the structure of data transfer.
Figure 1 depicts the basic architecture for interaction between device and client. WSDL defines the functionality of the service provided by the device. WSDL compiler is used for generating platform specific code which intends to integrate web service into a client side application.

C. Device Discovery

ONVIF specification follows WS-I basic profile 2.0. It uses the concept of exploring the network to find ONVIF capable devices.

D. Discovery Communication

Communication between client and the device which is implementing required target service is based on WS-Discovery multicast protocol.

Figure 2 illustrates client sending specific multicast probe message into the network. A unicast probe match message is sent by the concerned device to client based on probe message parameters. Device service address is provided on device discovery. This protocol supports remote discovery proxy even if the client and device is located in different network domains.

With reference to figure 3, NVT which is considered as an ONVIF compatible device will send multicast hello message as soon as it connects to the network. Clients will send multicast probe message encapsulated with requirements. NVT may receive multicast probe messages from different clients. Then probe match will be unicasted to respective client depending on matching rule. If a client knows the name of a device then it sends a multicast resolve message. The device checks name in the resolve message and if it matches, a resolve match message is generated and sent.
When NVT leaves the network, a bye message will be sent all over the network which informs to nodes that now the device will not be available for further communication. Polling refers to continuous checking of device, whether it is present in the network or not. This is reduced with the help of Hello/Bye messages [5]. Transport of messages takes place over UDP.

IV. OVERVIEW OF ONVIFSENSE APPLICATION

This paper elaborates a centralized application “OnvifSense” where multiple clients simultaneously send probe messages and OnvifSense takes the responsibility of further communication with ONVIF compatible devices like implements SOAP protocol, WS-Discovery, responding to clients with the capabilities of interested device. Thus this idea helps to reduce considerable overhead on Client to follow various handshaking signals. Queuing of requests from distinct clients helps to increase overall efficiency and throughput of the system. At this single point, technical administrator may be authorised with the privilege of managing clients and devices in the network.

Interiors of OnvifSense consist of two components namely 1) Discovery 2) Service. This division helps to attain modularity in the application.

A. Discovery component

Discovery component facilitates the discovery of ONVIF devices. Discovery can be made by using the attributes a) Types and Scopes b) Types only c) Scopes only.

1. Discovery by Types and Scopes:

To exemplify, if a client in an organisation of thousands of IP-based ONVIF-cameras is interested in finding all the cameras from Sony manufacturer having 4 mega pixels, 1 Gigabyte of storage, then the client will just fire a probe message to OnvifSense application. This message consists Types: Camera and Scopes: 4 mega pixels, 1 Gigabyte of storage. OnvifSense will already have the list of devices present in the network. Therefore this probe message received from the client is broadcasted to all devices. On the device side probe matching is done to see whether they satisfy the requirement and generate a probe match message and send it to the application. In the probe match message format device IP address is returned and each service on the device is associated with its specific network URL (Uniform Resource Located) address. Then the application forwards IP address of the device to the requesting client.

2. Discovery by Types only:

If a client prefers to discover devices based on device types only, then this mode is used. The scope parameter in the probe message will be null. Thus at the end of discovery, client will be equipped with list of device IPs, based on types such as NVT, camera, NVA, Network Video Storage etc.

3. Discovery by Scopes only:

If the client wants to search devices based on scope parameters such as location, hardware, name etc. then this mode is preferred. The type parameter in the probe message will be null. Thus at the end of discovery, client will be equipped with list of device IP, based on scopes.

B. Service component

After the discovery of desired device, the client has its IP address. In order to access the services on the device, client sends a SOAP message which contains the device IP. Services may include getting device information, making device backups, rebooting the device, upgrade firmware etc. OnvifSense application will have a list of array containing URLs (Uniform Resource Locator) of the services as received by a probe match message in the discovery component. Hence array containing the URLs of the services corresponding to queried IP is responded to client. Thus by clicking URL of interested service, the client can approach the device.

![Figure 4: System Architecture](image-url)
Figure 4 explains the basic design of our present system. All clients will be interacting with the OnvifSense application and it will communicate with all devices. Considering JAVA as underlying programming language, communication between various interfaces will take place with the help of socket and port numbers [8]. This application will continuously monitor various devices in the network.

A) Client

Each client will have an endpoint reference in the form of Universally Unique Identifier (UUID) which is static and invariant. The client will generate a Probe message which includes parameters namely type and scope. Type refers to the type of device wherein the client is interested to communicate with devices like NVT, IP-camera, NVA, etc. Scope refers to property of a device which may include hardware, location, video analytics value like streaming and recording etc. For example scope of IP-camera in Engineering College may be computer department. Hence IP-cameras in computer department will respond to probe message fired by the application. Then the client will wait for certain amount of time for receiving response from the application. Once the client receives a list of IP addresses, it can use them to access the functionality of those devices. To render the service component of OnvifSense, client has to fire the IP address of the device.

B) ONVIF Device

Once a device gets connected to the network, it sends a Hello message which contains its IP address. Afterwards all ONVIF compliant devices and OnvifSense application will be informed that a new device is located in the network. Now a device will receive multicast probe from the application where the device processes and verifies whether it meets with the type and scope parameters in the probe message. If it matches then probe match message will be generated and send to application. Similarly, for a resolve message, a resolve match message will be replied. A SOAP fault message is sent if device doesn’t adhere to matching rule.

When a device leaves the network, a Bye message will be broadcasted. Therefore when a client desires to interact with a device that has left the network, OnvifSense replies with a fault message that implies concerned device is not available to access the expected services.

C) OnvifSense Application

Initially, the application will receive probe messages from clients. These messages help our application to discover our targeted devices.

To handle multiple probe messages from a single client, the application maintains probe messages in a queue [7]. Hence for each client distinct queue is maintained. These queues are processed to get respective probe match messages from the devices in the network. OnvifSense senses a device connected to network by Hello message and it perceives that a device is disconnected from the network by receiving a Bye message.

Database schema of the application contains two tables. History table maintains a history of the communication between client and device and second table maintains types, scopes, and the array of URLs of the services corresponding to IP of the device. This table will be used for administrative purpose, statistical reports and interaction analysis. Configuration table is used by discovery and service component implemented by the application. For the first time communication between application and device, the corresponding entries in the databases are made for future interaction with the same device. When a probe message is received, application searches configuration table on the basis of types and scopes. If communication has already taken place, it retrieves IP address, types and scopes. If the search fails, then it will broadcast probe message in the network. The database is updated on receiving a Bye message from the device by deleting the corresponding entry from the configuration table.

Whenever a reference is made to the database, a “check message” is generated and sent to the respective device. An acknowledgement message is replied by the device. No message is responded if the device has suddenly gone down or a network crash happens. This technique maintains integrity with database and devices connected to the network. Hence at any point, the configuration table contains the entries of the devices that are currently available in the network.

D) Message Exchange Flow

With reference to figure 5, client resides in network 1, devices are present in network 2 and OnvifSense is present in public network so that both clients and devices can access the application. In addition to multicast Hello message, NVT will also send a remote Hello message to the OnvifSense as soon as it connects to the network 2. OnvifSense responds with Hello response message to the device as an acknowledgement which implies the device is ready for service. When a Hello message is received, OnvifSense records types, scopes and the services of the device. Clients are preconfigured with the application address and will send unicast probe message encapsulated with requirements.
Based on types and scopes present in probe message, it will find a match in the database and respond with probe match message. When device leaves the network, in addition to multicast Bye message, a remote Bye message is sent to the application. Device is acknowledged with a Bye response message by the application.

It is supported with password digest which is base64 encoding of the SHA-1 value that is formed by the combination of three parameters namely nonce, password, creation timestamp [2]. Password is UTF-8 encoded. A nonce is a random value which is included in Username Token. Thus for message exchanges, Username Token is included in SOAP messages. It is recommended to configure devices for Transport Level Security.

VI. IMPLEMENTATION CONSIDERATIONS

The proposed system “OnvifSense” utilizes the networking devices present in the same network domain. For discovering devices from different networks, developer may utilize the concept of discovery proxy. OnvifSense is not concerned about devices connecting or leaving the network. Therefore ONVIF devices should compulsorily implement Hello/Bye message while connecting or leaving network respectively. Present framework is built considering IPv4 address compatible devices. For IPv6 addresses necessary modifications are required.

The database which keeps track of client and ONVIF device communicated should be constantly updated unless there is a possibility of crashing the network.

VII. CONCLUSION

This paper introduces the idea of OnvifSense application, that eases the interoperability and communication between various network devices which are ONVIF standardized. The key components of OnvifSense are device discovery and services which utilize scope and attribute types of the device. Burst data is concurrently generated by the large number of broadcast messages exchange between multiple clients and the devices in the network. The overhead of excessive traffic generated in the network due to broadcasting probe messages is avoided using this application. Database resembles cache for frequently queried devices.

Developers would be benefited by the framework to develop more complex utility apps by directly using primitives in their code. Users would find it simple and handy to use application for performing various events and actions. Hence, it ensures to standardize communication between network devices to ensure interoperability between network products for the security market.
VIII. ABBREVIATIONS

ONVIF - Open Network Video Interface
RTSP - Real Time Streaming Protocol
SOAP - Simple Object Access Protocol
WSDL – Web Service Description Language
OV1 – ONVIF Device 1
PM – Probe Match
WS - Web Service
UDP – User Datagram Protocol
NVA – Network Video Analyser
NVT – Network Video Transmitter
IP – Internet Protocol
IPv4 – Internet Protocol Version
HTTP – Hyper Text Transfer Protocol
HTTPS - Hyper Text Transfer Protocol Secure

Acknowledgement

This project is sponsored by Persistent Systems Private Limited (PSPL). Authors are thankful to Mr Umesh Jaiswal (PSPL) for his vital guidance. We also convey our gratitude to Prof Mr S.P. Pingat and Prof Mrs S.V. Dhabade from Smt. Kashibai Navale College of Engineering, Pune for motivating us to write this research paper.

REFERENCES