

# Management of IoT Devices in Home Network via Intelligent Home Gateway Using NETCONF

Savita Vijay<sup>(✉)</sup> and M. K. Banga

Department of Computer Science and Engineering, Dayananda Sagar University,  
Kudlu Gate, Bangalore, India  
{savitavijay-cse, chairman-cse}@dsu.edu.in

**Abstract.** Internet of things (IoT) is surrounded by heterogeneous entities such as sensors, mobile devices and actuators in a constrained environment which are running on very low power and lossy networks. These entities are also of very small memory and can handle small computational overhead. To this end, complete IoT system and different devices which are working in home network management system will be presented in this paper. Applications for home network are considered under different architectures and their designs are discussed. Conventional simple network management protocol (SNMP) is generally applied for network management but it is not optimal due to lack of flexibility in configuring devices and lack of capabilities in managing operations in an IoT network. A design of smart washing machine device using the IoT design methodology is being discussed using network configuration protocol (NETCONF) and yet another next generation (YANG) data modeling language to illustrate how it would be a better alternative for managing home network.

**Keywords:** Internet of things (IoT)  
Intelligent home gateway network management (IHGNM)  
Simple network management protocol (SNMP)  
Network configuration protocol (NETCONF) · Netopeer

## 1 Introduction

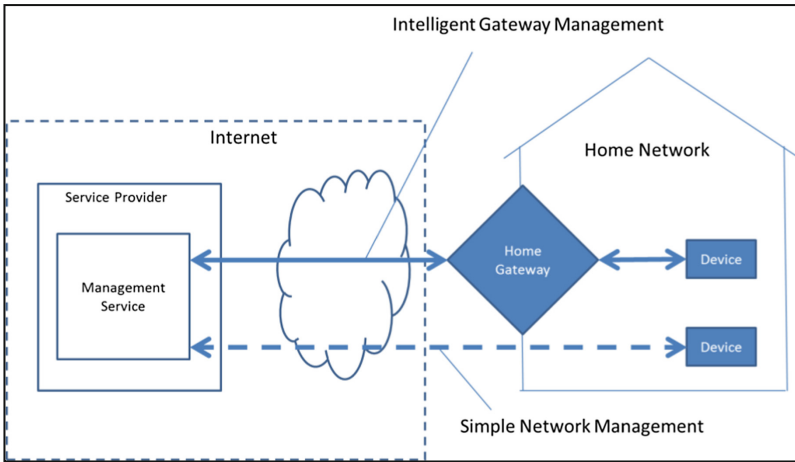
There are number of heterogeneous devices present in home network which are expected to be connected. Here, devices are based on different hardware platforms, controller services are also of different nature, the software components that enable network access on them are also of different nature for each device. For example, health and lifestyle wearable IoT devices like smartwatch, wristband have different capability in terms of memory usage, power consumption, processing speed as compared to smart home appliances like washing machine.

IoT devices can be broadly classified on the basis of their key characteristics like communication pattern, memory usage, data processing capability and power consumption. For example, devices like smart keys or a smart washing machine doesn't need to be connected always and are switched on to perform certain tasks when required; these devices consume less power for communication. This paper is focused on such low

power or normally off devices in a home network as they need a gateway for effectively managing operations and communication with internet.

Particularly, there are two common approaches for managing devices in home automation system: simple network management [1] and intelligent home gateway network management (IHGNM).

The main difference between them is involvement of the intelligent home gateway for management of the devices as shown in Fig. 1.



**Fig. 1.** Various components in home automation services architecture

Each of these approaches is as per the application. If the devices have enough resources that it supports a direct connection to the internet in a secure manner and does not support multiple device classes, then implementation is possible with light weight machine to machine (LWM2M) [2]. LWM2M is a remote device management standard. In this, IoT devices can be directly managed. This is an example of simple network management.

In IHGNM architecture, multiple device classes from low resource constrained to high resource constrained devices are considered. The proposed architecture monitors large number of heterogeneous devices in a home network. Here, IHGNM architecture for a high resource constrained device (i.e. washing machine) in a home network is discussed.

The rest of the paper is organized as follows: Sect. 2 describes IoT system components for home network management system and different devices for the same. Sections 3 and 4 describes intelligent home gateway design approach and study of same using netopeer tools. Section 5 gives conclusion and future research possible.

## 2 IoT System Components for Home Network Management System

### 2.1 A Home Network Management IoT System Comprises of Following Components

**Device:** An IoT device allows identification, remote sensing, actuating and monitoring from remote locations capability. Generally, IoT devices have unique identities, can exchange data with other connected devices and different web or mobile applications (directly or through Intelligent gateways), or collect data from other devices and store the data in local databases and process the data locally. Data can also be processed on cloud based application backend (like Amazon Web Services, Microsoft Azure) or on centralized servers. In different IoT design levels, we can perform some tasks locally and other tasks within IoT infrastructure, based on temporal and space constraints (i.e. memory, communication latencies and speeds, processing capabilities and deadlines).

**IoT software:** On IoT devices, some software components are required and installed for accessing the information from different sensors running in the home management system. They are also responsible for storing different sensor information or controlling the actuators connected to the devices. To enable network access on the device, resources are required.

**Native controller service:** This service is the native service that runs on the device. It interacts with the web services. For controlling the devices, it sends data from the device to the web service and receives commands from the application (via web services).

**IoT database:** Database for storing the collected data can be either local or it can be on cloud.

**IoT data security:** On local network, it will be done with symmetric key encryption decryption technique. And if it will run on internet then asymmetric key encryption decryption technique will be used.

**Web service:** Web services work as a connection link between all the IoT system components i.e. IoT device, IoT application, database for storing collected data and analysis components. Web services can be either implemented using hypertext transfer protocol (HTTP), constrained application protocol (CoAP) or representational state transfer (REST) principles or if it's a real time application then WebSocket protocol can also be used.

**Analysis component:** The Analysis component is responsible for analyzing the IoT data and generating results. Generally, results are published in a format that user can easily understand. Local and cloud, at both places, analysis can be done and then results are kept in local and cloud databases.

**Applications:** Users can control and monitor various features of IoT system with IoT applications only. To perform any operation on the IoT system or to see its status and to view the processed data, applications are used.

## 2.2 Device Heterogeneity and Applicable Management Approaches

There are numerous IoT devices across different applications in a home network as illustrated in Table 1. Based on the characteristics and constraints of devices in a home network, classification of devices is being done and appropriate management approach is recommended as highlighted in Tables 2 and 3 below.

**Table 1.** IoT devices for home network management system [3–5]

IoT device type	Device name
Smart lighting	Internet protocol enabled lights (tubelight, bulb), solid state lighting (LED lights)
Smart appliances	Refrigerator, air conditioner, television, television remote, music system, washer/dryer
Intrusion detection	Security camera, door sensor
Smoke/gas detectors	Smoke detectors (optical detection, ionization or air sampling technique), gas detector
Health and lifestyle	Wearable IoT devices (smartwatches e.g. moto 360 smart watch), smart glasses e.g. Google glass, wristbands (fitbit), fashion electronics (electronics in clothing and accessories, smart shoes), wearable ubiquitous healthcare monitoring system (integrated electrocardiogram (ECG)), accelerometer and oxygen saturation (SpO2) sensors)

**Table 2.** Classes of constrained IoT devices-Class 0, Class 1, Class 2 [6]

Class	RAM	Flash	Description
Class 0	<1 KB	<100 KB	Use gateway for basic communication need
Class 1	Approx 10 KB	Approx 100 KB	Use protocol stack as per IoT devices using CoAP. Can interact with other devices without the need of gateway
Class 2	Approx 50 KB	Approx 250 KB	These devices support regular IPv4 and IPv6 protocol. They function similar to other network devices

Device heterogeneity could be due to multiple aspects. This study is focusing on heterogeneity in terms of:

- (1) Characteristics of the device: In [9], device classification is done.

Depending on (i) memory usage and data processing capabilities and (ii) strategies for power consumption because existing management technologies use different protocol stacks and different protocol stacks consumes different amount of memory and

power. Class 0 devices cannot support simple network management because they lack resources require for proper communication and they cannot support any security standards also. However, both Class 1 and Class 2 devices support both simple and intelligent gateway management approaches.

**Table 3.** Devices and corresponding management approaches

	Simple network management	Intelligent home gateway network management
Suitable devices	IoT devices communicate directly to cloud	Communication between device to device
	Devices which can share background data	Communication between IoT device and gateway
	Class 1 devices, Class 2 devices	Class 0 devices, Class 1 devices
	Devices which are always on	Low power devices which are normally off [7, 8]

Table 4 lists and categorizes general strategies for power usage. Low-power or normally-off devices are not recommended in direct management approach as they cannot maintain the connection with the simple management service.

- (2) Communication pattern of devices: In home gateway network system, IoT devices can communicate in between or through gateway.

**Table 4.** Strategies of using power for communication

Name	Strategy	Ability to communicate
Class 0	Normally off	Reattach when required
Class 1	Low power	Appears connected, perhaps with high latency
Class 2	Always on	Always connected

### 3 Design Discussion

In the home network, devices are connected and controlled by the home gateway or directly managed by the remote management platform (RMP) running on the internet.

To manage multiple devices within a single system requires enhanced management capabilities because at home we have different IoT devices; which use sensors, different software and data collection, data analysis services and interfaces to interact with users.

#### 3.1 Simple Network Management System

Simple management services manage IoT devices directly without any gateway. The application running remotely on internet can communicate directly with the devices. Performance and the latency introduced because of gateways will be minimized here. The majority of devices that primarily exchange real-time sensory and control data in

small but numerous messages, direct management should be preferred in them, due to the aforementioned advantages.

Class 2 of IoT devices directly communicate to central servers for data storage. It supports IPv6 protocol. These classes of devices use powerful processors. They are not constrained by battery power. They also support gateway functionalities wherein they support different types of communication ports such as digital subscriber line (DSL), WiMax, WiFi etc. They support multiple sensor devices.

### **3.2 Intelligent Home Gateway Network Management**

In IoT intelligent home gateway network management system, all the home network devices are connected to gateway and gateway is connected to internet as shown in Fig. 1. In the given diagram, every device that perform sensing and/or actuation, stores the collected data on their datastores, perform analysis and hosts the application on gateway.

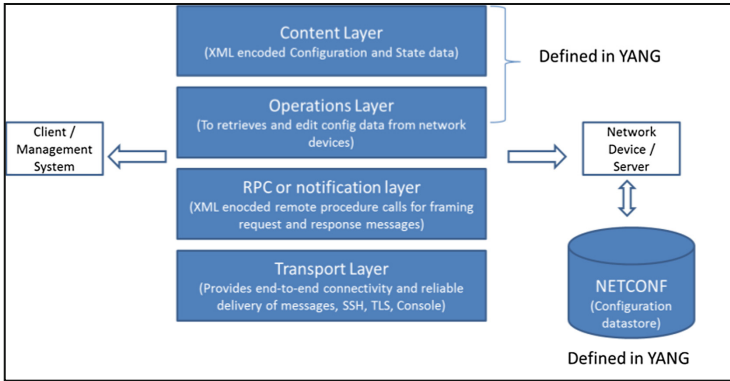
#### **3.2.1 Intelligent Home Gateway Network Management with NETCONF and YANG**

There are two standard protocols for network management viz. network configuration protocol (NETCONF) [10, 11] and simple network management protocol (SNMP).

#### **3.2.2 Introduction to NETCONF and YANG**

In 2002, internet architecture board (IAB) workshop held, in that workshop it was concluded that SNMP is not suitable for configuration management. This is documented in RFC 3535. That was a point to initiate research on NETCONF and YANG.

NETCONF is an internet engineering task force (IETF) network management protocol and recorded in RFC 4741. It is a session based network management protocol. It provides mechanisms to install, manipulate, and delete the configuration of network devices. Its operations are running on top of a simple remote procedure call (RPC) layer. It uses XML based remote procedure calls for framing request and response messages. It works on secure shell transport layer protocol. It also supports block extensible exchange protocol (BEEP) which is a transport layer protocol. Transport layer provides port to port connectivity and reliable delivery of messages. It can also replace command line interpreter (CLI) based programming interfaces like perl running over secure shell (SSH). It uses structured schema driven data and provides error return information also in structured format, which even CLI cannot provide. Figure 2 shows the layered architecture of NETCONF protocol.



**Fig. 2.** NETCONF protocol layers

The content layer consists of state data and configuration of each device running in home like TV, refrigerator in XML format. YANG is a data modeling language used to model configuration and state data of devices manipulated by NETCONF protocol [8, 9]. The definition of data exchanged between NETCONF client and server i.e. device and management system running on gateway. NETCONF operation <get-config> retrieves the configuration data of devices, and the operation <get> retrieves the state and configuration data of devices. On every device a NETCONF configuration datastore is running to keep configuration data. Here, the client and server maintains the NETCONF session for communication. Client manages the server (device) with ‘hello’ message exchange to share each other capabilities. Client can then send n number of requests to the server for retrieving and editing the configuration data. NETCONF allows management client to discover the capabilities of the device and also access of its native capabilities.

NETCONF defines on the devices one or more configuration datastores. A configuration store contains all the configuration information to bring the device from its initial state to the operational state. By default a <running> configuration store is present. Additional configuration datastores as per the device need such as <startup> and <candidate> can be defined in the capabilities.

NETCONF is a connection oriented protocol and uses SSH or transport layer security (TLS) transport protocol for providing security features on server like authentication, data integrity and confidentiality. NETCONF connection persists between protocol operations.

### 3.2.3 YANG

YANG is a data modeling language. It is a standard defined by the IETF in the network modeling (NETMOD) working group. YANG can said to be tree-structured. Configuration data is structured into the tree and data can be of complex type such as lists and unions. It is used to model configuration and state data manipulated by the NETCONF protocol.

YANG modules define configuration data, remote procedure calls and state data that can be issued and it decides the format of notifications also. Whatever data is exchanged

between client and server, format of that data is decided by the NETCONF protocol. A YANG module comprises of number of ‘leaf’ nodes which are specified using the ‘leaf’ or ‘leaf list’ constructs. Leaf nodes are organized using ‘container’ or ‘list’ constructs. On data nodes constraints and data validation can also be defined. YANG module can use other modules also by introducing their definition in it. ‘config’ statement is used to model both configuration data and state data.

### 3.2.4 SNMP and NETCONF

SNMP is also widely used network management protocol which is responsible for monitoring and configuring network devices such as router, printer, scanner, server etc. Table 5 shows the comparison in between SNMP and NETCONF, and shows the suitability of NETCONF over SNMP.

**Table 5.** Comparison of SNMP and NETCONF protocols [12]

S. no.	SNMP	NETCONF
1	SNMP uses user datagram protocol (UDP). UDP is a transport layer connectionless protocol which makes SNMP unreliable	It uses SSH protocol. It ensures reliable delivery of messages
2	It is stateless in nature. Management application should be intelligent to manage the device	NETCONF is session-based protocol
3	Generally lacks writeable objects without which device configuration is not possible	SNMP can only be used for device monitoring and status polling while NETCONF allows to retrieve state or configuration data of network devices
4	Very difficult to differentiate between configuration and state data	<get-config> retrieve configuration data only. <get> retrieve both state and configuration data
5	It does not support easy retrieval and playback of configurations	NETCONF gives access to the native capabilities of the device
6	Latest version of SNMP providing security support is very complex	NETCONF uses SSH or TLS for security services

## 4 Study of Intelligent Home Gateway Network Management Through NETCONF and YANG Using Netopeer Tools

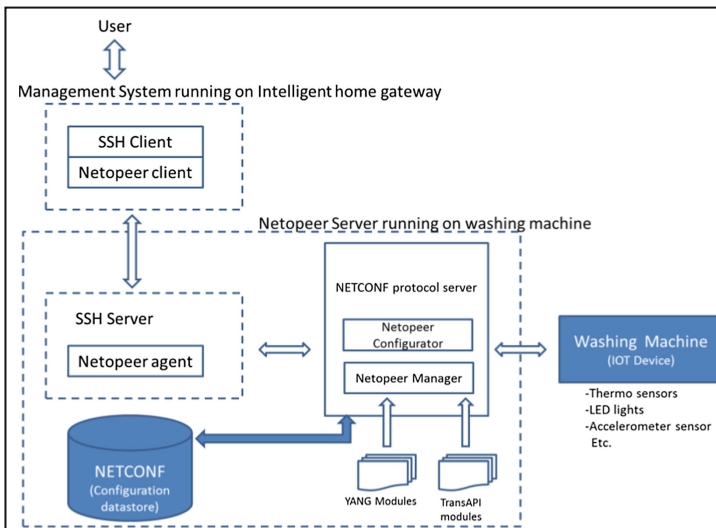
We have heard about the smart devices and at home also we have electronic devices like washing machine, dishwasher, refrigerator, etc. Initially Amazon launched Amazon dash button for washing machines. Using this button, washing machine prompts users to order more detergent for itself when it is running low. Afterward, around 40 services were launched in the form of Amazon dash replenishment service also known as DRS buttons. They are linked to the Amazon account and allow anyone in the home to instantly order staples from toothbrush heads, kitchen rolls and washing up liquid to coffee. Internet of things is big technology space and in home network also there are



series of interconnected devices that look to automate tasks such as lights that respond to presence or timers or smart thermostats that save energy by only putting the heating on when people are in the house.

A design of smart washing machine device using the IoT design methodology is being discussed. The purpose of the washing machine automation system is to control the machine in a home remotely using a gateway running management system application and connected to the internet and other IoT devices running in home network. Implementation tool considered is netopeer. Netopeer is a set of open source NETCONF tools built on libnetconf library [13].

Figure 3 shows how to manage a washing machine using the netopeer tools. It includes:



**Fig. 3.** Washing machine management with NETCONF using netopeer tool

1. **Netopeer-server:** Netopeer-server is a NETCONF protocol server that runs on the washing machine. It provides an environment for configuring the washing machine using NETCONF RPC operations and also retrieving the state data from the washing machine.

On washing machine, three services are running, it includes:

- (a) Native controller service – runs as a native service on washing machine. Gets the current mode (auto/manual), current state (on/off/wash/spin) and sends to the netopeer-cli.
- (b) Mode service: Every machine includes auto and manual modes.
  - i. In auto mode, system measures clothes in machine and switch it on if clothes are there.
  - ii. In manual mode, the system provides the option of manually and remotely switching on/off the machine.

- (c) There are four states of washing machine we are considering:
- i. On
  - ii. Off
  - iii. Wash (for washing clothes)
  - iv. Spin (for spinning clothes).
2. **Netopeer-agent:** Netopeer-agent is the NETCONF protocol agent running as a secure shell subsystem. It accepts incoming NETCONF connection and passes the NETCONF RPC operations received from the NETCONF client (running on gateway) to the netopeer-server (running on washing machine). It is also responsible for authentication and checking integrity of message (if required) in the request message. It checks that request message is for washing machine only and checks the syntax of message.
  3. **Netopeer-cli:** It is a NETCONF client that provides a command line interface for interacting with the device running netopeer-server. The operator can use the netopeer-cli from the gateway management system to send NETCONF RPC operations for configuring the washing machine and retrieving the state information.
  4. **Netopeer-manager:** Netopeer-manager allows managing the YANG and libnetconf transaction API (TransAPI) modules on the washing machine. With netopeer-manager modules can be loaded or removed from the washing machine.
  5. **Netopeer-configurator:** Netopeer-configurator is a tool that be used to configure the netopeer-server.

#### 4.1 Steps for Managing Washing Machine with NETCONF-YANG

- i. Create a YANG module of the washing machine in home management system that defines the configuration and its state data on its hierarchical tree structure [14].
- ii. Compile the YANG model with the 'inctool'. Inctool is part of libnetconf library. Whatever the changes done in the configuration file of washing machine, to reflect those changes in actual washing machine device, TransAPI framework is used. The 'Inctool' generates the TransAPI module (callbacks C file) and the YIN file. The callbacks C file contains the functions for making the change on the washing machine. YIN file contains an XML representation of the YANG module.
  - Inctool – model washingmachine.yang convert
  - Inctool – model washingmachine.yang validation
  - Inctool – model washingmachine.yang transapi – paths washingmachine.paths
- iii. Fill in the IoT device management code in the transaction API module also known as callbacks C file. This file comprises of configuration callbacks, RPC callbacks and state data callbacks.
- iv. Below commands are issued to build the callbacks C file as a result generate (.so) library file.
  - Autoreconf
  - ./configure
  - make

- v. Netopeer manager tool loads the YANG module and .so binary generated by TransAPI into the washing machine.
  - Sudo netopeer - manager add – name washingmachine – model washingmachine.yin – datastore/home/ubuntu/washingmachine.xml
- vi. The user can now connect from the management system running on gateway to the netopeer server using the netopeer-cli.
  - netopeer-cli
  - netconf> connect
  - Host: localhost
  - Password:
- vii. User can issue NETCONF commands from the netopeer client, CLI. Commands can be issued to change the configuration data of washing machine, get operational data or execute an RPC on it.
  - netconf> get
  - ..
  - ..
  - netconf> get-config running
  - netconf> edit-config running
  - ..
  - ..
  - netconf> user-rpc.

## 5 Conclusion and Future Directions

This paper discusses IoT system for intelligent home gateway network management system to handle different IoT devices in the network. Complete design of IoT devices, their software and services running on them are discussed. Here, the study is done by choosing washing machine device from home network system and how it can be managed using the architecture involving NETCONF protocol, YANG data modeling language and netopeer tools. With this architecture, we can manage multiple device classes especially devices which are of limited resource capability like low power, slow processing, and less communication capability. NETCONF protocol manages the operation and configuration of devices in home network in a reliable and efficient manner as it is session based thus reduces the traffic to the home gateway and also easily retrieves states and configuration of devices. As future research, some work can be done around introducing artificial intelligence into the IoT devices of the system.

## References

1. Pham, C., Lim, Y., Tan, Y.: Management architecture for heterogeneous IoT devices in home network. IEEE (2016)
2. Rao, S., Chendanda, D., Deshpande, C., Lakkundi, V.: Implementing LWM2M in constrained IoT devices. In: ICWiSe, pp. 52–57 (2015)

3. Caldeira, J.M.L.P., Rodrigues, J.J.P.C., Lorenz, P.: Toward ubiquitous mobility solutions for body sensor networks on healthcare. *IEEE Commun. Mag.* **50**, 108–115 (2012)
4. Chung, W.Y., Lee, Y.D., Jung, S.J.: A wireless sensor network compatible wearable U-healthcare monitoring system using integrated ECG, accelerometer and SpO<sub>2</sub>. In: International Conference of the IEEE Engineering in Medicine and Biology Society (2008)
5. Bahga, A., Madiseti, V.: Internet of Things, A Hands on Approach (2015)
6. ITU-T: Overview of the Internet of Things, Y.2062 (2012)
7. Ersue, M., Romascanu, D., Schoenwaelder, J.: Management of networks with constrained devices: problem statement and requirements, RFC 7547 (2015)
8. Bormann, C., Ersue, M., Keranen, A.: Terminology for constrained-node networks, RFC 7228 (2014)
9. Sehgal, A., Perelman, V., Kuryla, S., Schonwalder, J.: Management of resource constrained devices in the internet of things. *IEEE Commun. Mag.* **50**, 144–149 (2012)
10. Enns, R., Bjorklund, M., Bierman, A.: Network configuration protocol (NETCONF), RFC 6241 (2011)
11. Schönwälder, J., Björklund, M., Shafer, P.: Network configuration management using NETCONF and YANG (2010)
12. Harrington, D., Preshun, R., Wijnen, B.: An architecture for describing simple network management protocol (SNMP) management frameworks, RFC 3411 (2002)
13. libnetconf (2014). <https://github.com/CZ-NIC/libnetconf>
14. Tschofenig, H., Arkko, J., Thaler, D., McPherson, D.: Architectural considerations in smart object networking, RFC 7452 (2015)