A Study on Real-time Communications Management Based on Packet Propagation Time Monitoring and Hop-by-Hop Packet Authentication for Process Automation

A Dissertation Submitted to
Department of Electronic and Information Engineering,
the Graduate school of Engineering of
Tokyo University of Agriculture and Technology
for the degree of Doctor of Philosophy in Engineering

March 2015

Hiroshi Miyata

ABSTRACT

In the past, Process Automation (PA) has maintained reliability and security by building a dedicated PA system with proprietary technology. However, some significant changes are emerging on PA due to technical innovation of Information Communication Technology (ICT) and Information Technology (IT). One of the typical changes is connecting facilities in a plant with a network infrastructure. In this dissertation, the network infrastructure is called Industrial Backhaul. The traffic transmitted to the Industrial Backhaul includes real-time communications and non-real-time communications. In addition, two characteristics of real-time communications exist such as static and dynamic. This dissertation introduces a study addressing "to provide dependable real-time communications management for PA operations over Industrial Backhaul".

There are a numbers of existing studies addressing real-time communications. However, the dependable real-time communications management is not yet satisfied, since an attacker can disrupt real-time communications by sending plenty of spoofing packets for example. This study calls such disrupting attacks "QoS Spoofing Attacks". To clarify the missing functions, this study identifies five essential functions to allow dependable real-time communications management based on feedback control concept, which is commonly used in PA. The functions are as follows: "Bandwidth Control", "Bandwidth Allocation", "Assessment of Real-time Communications", "Prevention of Unauthorized Bandwidth Allocation", and "Prevention of QoS Spoofing Attacks". As a consequence of survey of existing study, it is clarified that "Assessment of Real-time Communications" and "Prevention of QoS Spoofing Attacks" are missing.

This study defines providing the two functions of "Assessment of Real-time Communications" and "Prevention of QoS Spoofing Attacks" is the major challenge to achieve the dependable real-time communications management. This study defines two goals of "management method for real-time communications based on packet propagation time monitoring" and "Hop-by-Hop packet authentication method to protect the output queue" to satisfy the "Assessment of Real-time Communications" and "Prevention of QoS Spoofing Attacks" respectively.

Firstly, this dissertation proposes a management method for real-time communications based on packet propagation time monitoring. The method is designed based on OpenFlow. The proposed method monitors the packet propagation time in addition to the utilization of allocated bandwidth on a corresponding output queue of intermediate network equipment. The monitoring function allows detection and prediction of timeout of real-time communications to reallocate the bandwidth. The method was prototyped and evaluated. The evaluation indicated that monitoring packet propagation time and monitoring utilization of allocated bandwidth mutually complement to provide real-time communications even to the dynamic traffic. Thus, the proposed method achieves a goal of "management method for real-time communications based on packet propagation time monitoring".

Secondly, this dissertation analyzes the reason of QoS Spoofing Attacks and proposes a Hop-by-Hop packet authentication method to address QoS Spoofing Attacks. The attacks disrupt real-time communications by consuming the bandwidth allocated for real-time communications with the spoofing packets. To prevent the attacks, the intermediate equipment needs to discard the spoofing packets before the allocated bandwidth is consumed. Thus, this study proposes a lightweight Hop-by-Hop authentication method. The method was prototyped and evaluated. The evaluation indicated that the proposed method allows detection and discursion of spoofing packet while satisfying the requirement of PA for real-time communications.

This dissertation introduces the advanced method applying the Hop-by-Hop packet authentication method to OpenFlow based network environment. The advanced method allows flexible and scalable Hop-by-Hop packet authentication. The method is achieved by newly added Packet Authentication Plane into the OpenFlow environment. This method allows bandwidth control combined with flexible flow definition and scale out of packet authentication function. The method was prototyped and evaluated. The evaluation indicated that the proposed method allows bandwidth protection from spoofing packets with acceptable overhead for the real-time communications of PA. Thus, the proposed method achieves a goal of "Hop-by-Hop packet authentication method to protect the output queue".

The proposed methods satisfy the two goals "management method for real-time communications based on packet propagation time monitoring" and "Hop-by-Hop packet authentication method to protect the output queue". The two methods have high affinity to cooperate each other, since the methods can work on OpenFlow. As a consequence, the two missing functions to achieve the dependable real-time communications management for PA operations over an Industrial Backhaul are provided. This study revealed that the managing

functions such as monitoring propagation time, bandwidth utilization, discarding spoofing packet are useful and required for the Industrial Backhaul, which has strict requirement for real-time communications.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AH	IP Authentication Header
CM	CONFIG Manager
DCS	Distributed Control System
DiffServ	Differentiated Service
DSCP	Differentiated Service Code Point
ERP	Enterprise Resource Planning
ESP	IP Encapsulated Security Payload
ETSI	European Telecommunications Standards Institute
FA	Factory Automation
FF	Foundation Fieldbus
FF-H1	Foundation Fieldbus – H1
FF-HSE	Foundation Fieldbus – High Speed Ethernet
HMAC	Keyed-Hashing for Message Authentication
HMI	Human Machine Interface
ICS	Industrial Control System
ICT	Information Communication Technology
IntServ	Integrated Service
ISA	International Society of Automation
MAC	Message Authentication Code
MOM	Manufacturing Operations Management
NFV	Network Functions Virtualization
NTP	Network Time Protocol
OFC	OpenFlow Controller
OFS	OpenFlow Switch
OF-CONFIG	OpenFlow Management and Configuration Protocol
OS	Operating System
OVSDB	Open vSwitch Database
PA	Process Automation
PCoIP	PC over IP

PCP	Priority Code Point
PHB	Per Hop Behavior
PID	Proportional-Integral-Derivative
PLA	Packet Level Authentication
PLC	Programmable Logic Controller
pmqFlow	Propagation time monitoring QoS with OpenFlow
Pub/Sub	Publisher/Subscriber
QoS	Quality of Service
RAIDUS	Remote Authentication Dial In User Service
RCM	Real-time Communication Manager
RDP	Remote Desktop Protocol
RR	Receiver Report
RSVP	Resource Reservation Protocol
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
RTT	Round Trip Time
RSVP-SQoS	RSVP with scalable QoS protection
SDN	Software Defined Network
SR	Sender Report
TLS	Transport Layer Security
VID	Virtual LAN ID
VLAN	Virtual LAN
WSN	Wireless Sensor Networks

Chapter 1

INTRODUCTION

1.1 BACKGROUND

In the past, Process Automation (PA) has maintained reliability and security by building a dedicated PA system with proprietary technology. However, some significant changes are emerging on PA due to technical innovation of Information Communication Technology (ICT) and Information Technology (IT). One of the typical changes is connecting facilities in a plant with a network infrastructure. On the other hand, the threat of cyber attacks have been increasing as indicated by the reported cyber attacks on plants.

In this chapter, the features of PA are introduced at first. Secondly, the hierarchy models of PA are introduced to clarify the scope of this study. Thirdly, the emerging changes and concerns in the plant are introduced to lead the objectives of this study. Lastly, the structure of this dissertation is introduced.

1.1.1 PROCESS AUTOMATION

Industrial Automation is classified into two kinds of automation by the target of control. One is Process Automation and another is Factory Automation (FA). The PA has different characteristics from FA, since PA mainly deals with fluid while FA assembles parts to build a car for example.

In FA, many kinds of robots are utilized and the positioning and timing are important for assembling the product. The speed of each process impacts the efficiency of production. Thus, fast control is required. In contrast to FA, PA senses and controls the temperature, pressure, or flow volume with sensor and actuator (both devices are called Field Device) to make a product. The controls against the fluid (e.g., heating, cooling, opening the bulb and closing the bulb) cannot react as quickly as FA. The requirement for control communications is more moderate than the requirement of FA. In general, fieldbus for FA requires communications less than 10 ms while fieldbus for PA requires communications tens of ms.

Figure 1.1 represents the Proportional-Integral-Derivative Controller (PID) control loop, which is mainly utilized in PA. The PID control loop consists of:

(1) The sensor (e.g., temperature transmitter) obtains the measured value and transmit it to the PID controller;

(2) The PID controller calculates and transmit configuration request to the actuator (e.g., bulb positioner);

(3) The actuator configures as requested and transmits the configured value to the PID controller.

As described above, the PID control loop consists of three communications and three devices' internal processes. The loop is repeated periodically. The typical loop cycle is one second. In the fast case, the cycle is 0.3 ms.

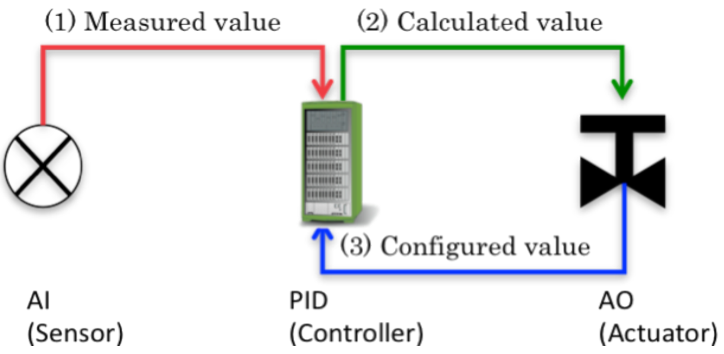


FIGURE 1.1 PID CONTROL LOOP IN PA

Since the PA deals with explosive material like oil and chemical plant, incorrect control or data loss can cause explosions of the plant. Thus, the reliability of the communication is important. Mostly, the PID control loop is operated periodically; the data that does not arrive on schedule will not be used for the PID control. It means, packets delayed for the schedule will be considered as unreached packets. Thus, the delay of the communications can be considered as a factor disrupting the communication reliability in terms of PID control. In FA, the data loss may stop the robot but not cause the serious problem as explosions. To summarize, the PA communications require packets arriving on schedule, while it does not require speed so strongly. On the other hand, the FA communications requires speed while the data loss does not cause a serious problem.

As mentioned above, PA and FA have different characteristics. This means requirements for communications are also different. As the result, the method developed for real-time communications of PA may not be useful for FA applications, or vice versa. Thus, this study focuses on PA as the target industrial automation.

1.1.2 Functional Hierarchy Model in Process Automation

IEC 62264-1[1] defines a functional hierarchy model and role-based equipment hierarchy model of a manufacturing system; these are widely utilized as the reference models of manufacturing system. The functional hierarchy model classifies the entire manufacturing functions into five functional levels. Each functional level has specific network requirements since each level has specific functions. The role-based equipment hierarchy model is close to a physical equipment model. This subsection clarifies the target of this study with a functional hierarchy model. Since the functional hierarchy model represents functional level with specific time scale, the model is useful to map the requirements for real-time communications. In addition, this subsection clarifies the target network of this study with a role-based hierarchy model, since it helps to understand the physical structure of a plant.

Table 1.1 represents the functional hierarchy model. The model consists of five levels from level 0 to level 4, from downstream to upstream. Each level has specific requirements for communication with an exception of level 0, which consists of tank and pipe, since it does not communicate. Level 4 is in charge of enterprise management system, which deals with high-level business planning such as order entry or shipping. It is a common IT system in the enterprise. Level 4 makes rough manufacturing plan. Level 3 is Manufacturing Operations Management system (MOM), which makes detailed manufacturing plan by breaking down the plan made by Level 4. MOM dispatches the detailed order to particular equipment and specifies the material with concrete schedules. Level 2 and Level 1 are in charge of actual manufacturing according to the plan made by Level 3. In Level 2, Human Machine Interface (HMI) configures the transmitter or other PA equipment to prepare the manufacturing. In addition, HMI monitors and controls the equipment. The transmitters and actuators are placed on Level 1. In Level 1, sensing and controlling is performed with the equipment to manufacture the desired quality of product. Actually, both Level 1 and Level 2 work together to provide actual manufacturing functions.

As shown in Table 1.1, from Level 4 to Level 1, the plan is gradually getting more detailed. Thus, the time scale is getting more detailed from Level 4 to Level 1. This means the requirement for real-time communications is getting stricter. In Level 1 and 2, that has the strongest real-time communications, dedicated fieldbus or network technologies are utilized.

Table 1. 1 Functional Hierarchy Model

Category	**Level**	**Functions**	**Time Frame**
Enterprise Information Network	4	Establishing the basic plant schedule for production, material use, shipping, etc.	Months, weeks, days
Industrial Control System	3	Workflow/Recipe control to produce the desired end products. Dispatching production, detailed production schedule, etc.	Days, hours, minutes, seconds, shifts
	2	Monitoring, supervisory control and automated control of the production process	Hours, minutes, seconds, subseconds
	1	Sensing and manipulating the production process	
	0	The actual production process	N/A

1.1.3 Role-based Equipment Hierarchy Model in Process Automation

The role-based equipment hierarchy model is close to a production equipment scheme. Thus, it is useful to understand the physical structure of a production system. Table 1.2 describes each component of a role-based hierarchy model. Enterprise is the top level of a role-based equipment hierarchy model, which determines the entire production plan. Level 4 functions are concerned with Enterprise and Site levels in general. The production plans determined by Enterprise are distributed to Sites, which manages the regional manufacturing facilities. The Site determines detailed manufacturing plan for the given region. This means that Site can run Level 4 function in detail for the given region. The production plan determined in the Site can be detailed in the Areas, which is under the Site, as required. It is broken down into the actual production execution schedule. In other words, the schedule specifies which personnel manufacture what kind of product, at which facility or equipment, at when, and with which materials. Thus, Area mainly has Level 3 function. The Work Center and Work Unit is responsible for actual manufacturing according to the detailed schedule determined by Level 3. This means that Work Center and Work Unit work together to perform Level 2 and Level 1 functions.

TABLE 1.2 ROLE-BASED EQUIPMENT HIERARCHY MODEL

Role	Functions	Example
Enterprise	The top level of a role-based equipment hierarchy. It is responsible for determining what product will be manufactured at which site. It provides Level 4 functions in general.	Collection of Sites and Areas
Site	Site is a group of production facilities, defined by physical, geographical, or logical point of view. grouping of. Site is recognized as a delegation of regional manufacturing facility. It may have Level 4 function.	Mizushima ethylene Plant, Anegasaki Chemical Plant
Area	Area is a group of production facilities within a Site, defined by physical, geographical, or logical point of view. It mainly have Level 3 function.	Assembling Building, Blending facility, Utility Facility
Work Center	Work center is a group of equipment scheduled by Level 3 or Level 4 functions. Production facility or equipment for specific purpose within the Area.	Steam Cracker #8, Catalytic Cracker #2, Storage Tank zone, Shipping Center
Work Unit	Work Unit is the lowest element of role-based equipment hierarchy model. It is a component of the Work Center such as sensor or actuator. It is scheduled by Level 3 functions.	Storage Tank, Rack

1.1.4 RECENT CHANGES IN PROCESS AUTOMATION

This subsection introduces the emerging changes in PA. In addition, this subsection also introduces emerging network technologies to discuss the applicability to the changes in PA.

A) Wireless Equipment

The PA users currently start to introduce wireless sensor networks (WSN) to the plants to reduce the installation cost of Field Devices such as sensors or actuators. The typical WSN technologies designed for PA are ISA100.11a [2] and WirelessHART [3]. The users can install the Field Devices quickly with less constraint of wiring. The WSN provides the advantage of higher accuracy and efficiency into PA operation, since WSN allows users to install required Field Devices to any place at anytime as required.

To utilize the data obtained by WSN, the data shall be transmitted to the devices that need the data or to the data storage servers in a central control room. The data is utilized in control loop (e.g., PID control) or in HMI to monitor the operation. A type of HMI working on mobile terminals, which are so called Mobile HMIs, is emerging to be used in the area where actual equipment or the facility is located in the plant (the area is called Field in this study). The Mobile HMI allows for operator efficient operation; staying beside the actual equipment and watching the current situation and historical related data. To utilize the Mobile HMI, the data of HMI shall be transmitted from the control room to the Field.

B) Industrial Backhaul

To transmit the obtained data from WSN to the control room, or HMI data from the control room to the Field, a network installed through out the plant is utilized. Such network is called Industrial Backhaul. The International Society of Automation (ISA) has developed a technical report [4] presenting the desired Industrial Backhaul architecture. The technical report introduces kinds of usage of Industrial Backhaul. One typical example is interconnecting between Sites, another example is interconnecting Areas inside the Site. This study focuses on interconnection Areas, since it requires more strict real-time communications. Figure 1.2 represents the abstracted overview of Industrial Backhaul with entities of role-based equipment hierarchy model.

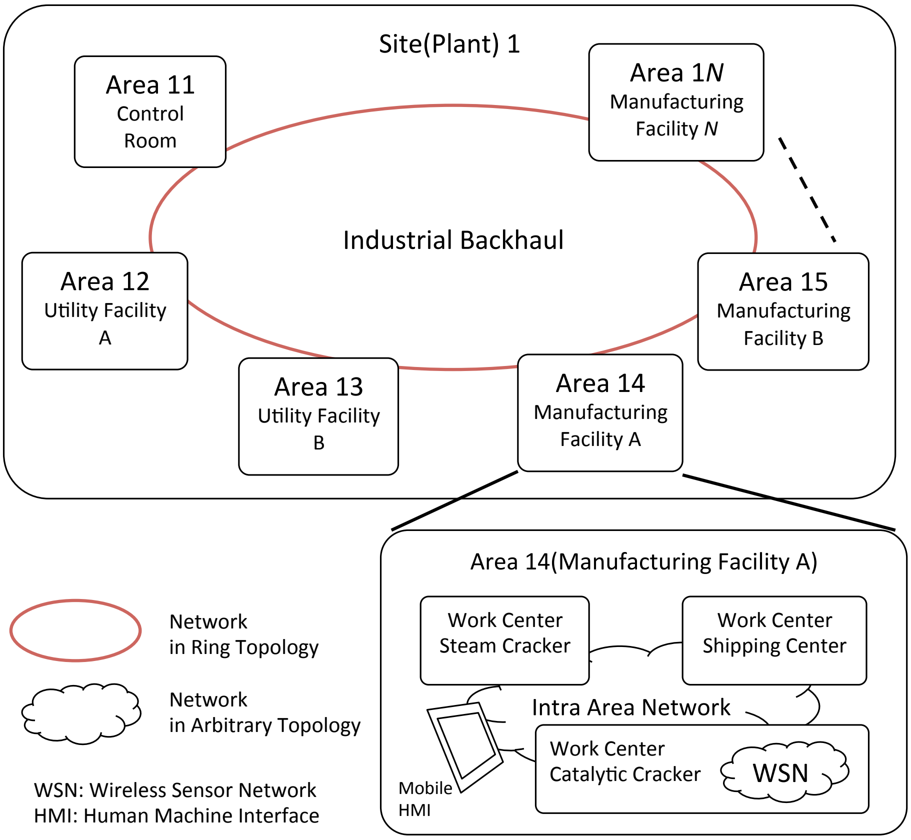


FIGURE 1.2 PLANT ROLL-BASE EQUIPMENT HIERARCHY MODEL WITH INDUSTRIAL BACKHAUL

Even within a Site, the Industrial Backhaul is installed on a vast site. Thus, it is a very long and expensive network. This means there is a constraint to install additional link. For example, the world largest class of plant, as known as Shell Nanhai [5], has nine Areas in a site of 260ha. Since the distance between sites is kilos meters, it is not possible to build mesh topology to provide network redundancy due to the cost restriction. Thus, the network is commonly built in a ring topology. Within the Area, a smaller network is built to interconnect Work Centers. The Mobile HMI would be attached to this intra Area network, and the WSN would be attached within the Work Center.

As described above, the intra Site network is in hierarchy architecture. Thus, the total number of involved Field Device is large. In Shell Nanhai, the total number of Field Devices is 16,000.

To run PID control or to utilize the Mobile HMI over the Industrial Backhaul, real-time communications are required. On the other hand, the industrial backhaul should be considered as an untrusted network since many kinds of users in many Areas transmits the packet onto the Industrial Backhaul. In addition, the transmitted data has a variety of characteristics depending on the application. This means the traffic may interfere each other. Thus, [4] recommends introducing QoS methods to provide real-time communications on the Industrial Backhaul.

C) Collaborative Operation

Manufacturing companies start to improve production efficiency by connecting a variety of equipment or functions in the production system. This kind of approach is represented by Industry 4.0 [6]. Industry 4.0 is known as the fourth industrial revolution. The approach collects and analyzes a variety of production related information to make an efficient production plan. In the past, PA collects the data from Field Devices to utilize for the equipment or for facility level optimization. Industry 4.0 expands the existing PA approach for larger scale optimization by connecting larger number of equipment or higher levels of functions.

Industry 4.0 allows timely production by exchanging production plan from order entry system to MOM system promptly. In addition, Industry 4.0 allows reliable shipping by collecting the maintenance information to avoid dispatching orders to the facility or equipment that would be maintained. These effects are just a part of examples. Users expect more effects for Industry 4.0. There are a variety of activities related to Industry 4.0 such as Industrial Internet Consortium [7] and Smart Manufacturing Leadership Coalition [8]. This means this is a promising approach to improve manufacturing efficiency.

D) Targeted Cyber Attacks

Cyber-attacks are changing and becoming more sophisticated day-by-day. In the past, the cyber attacks don’t have particular target or particular purpose. However, today the attacks have particular purpose on particular targets. Such recent cyber attacks utilize unreported security holes within the Operating System (OS). Thus, it is not easy to take preventive measures against the attacks. Plants are facing such dangerous situations as well as IT devices. The cyber attacks in the plant are represented by Stuxnet [9], which has been reported in 2010. Stuxnet is reported as a malware designed for particular industrial device of Programmable Logic Controller (PLC) to attack the nuclear power plant. After the report of Stuxnet, PA users have realized the possibility of cyber attacks targeting the plant.

The threat of cyber attacks is not limited to Stuxnet. Shodan [10] is a popular portal site for a control system. Shodan allows attackers to find a control device that is accessible from the Internet. After finding a control device, the attacker can log-in to the device and generate the control code to improperly control the device. Shodan has exposed the attackers interest to the control system. In addition, Shodan has made the barrier lower to attack the control system.

The cyber attacks on Plant enable the damage to not only the cyber side but also the physical side. For example, the modification of control algorithms and/or sensor data can cause improper command to an actuator that results in inefficient manufacturing or, in the worst-case scenario, explosions of the plant.

The attacks in the plants are also possible by losing the data as well as modification of control algorithms and/or data. As mentioned in Subsection 1.1.1, sensor data delayed for its schedule is not used for PID control. Thus, the delay of arrival has the same meaning as packet loss. In general, priority control and bandwidth control (both are called Quality of Service (QoS)) is utilized to provide real-time communications. The existing QoS methods classify the packets based on the attributes on the packet to differentiate the forwarding policy. This differentiation allows packets of real-time communications to be set with higher priority or to be assigned a dedicated queue with particular bandwidth. In other words, it is possible to generate packet spoofing real-time communications. As an example, the spoofing packets can disrupt real-time communications of proper packets by consuming the dedicated output queue. In this study, such attacks are called QoS Spoofing Attacks. For example, the devices attached to networks related to Level 3 functions could cause QoS Spoofing Attacks, since Level 3 related networks is not well managed as Level 2 related networks so far and the traffic of both Level 2 and Level 3 can go through the Industrial Backhaul. The attacks disrupting real-time communications are a new and unique threat in the plant, since such attacks can cause an irreparable accident.

1.1.5 Requirement for Real-time Communications

This subsection analyzes the requirements of real-time communications in PA. For this analysis, the PID control and Mobile HMI are targeted as the representing application of Level 1 and Level 2; those require strongest real-time communications.

In PA, the shortest cycle of PID control loop is 300 ms while typical cycle of PID control is one second. The PID control loop consists of three processes on sensor, PID controller, and actuator. In addition, the PID control loop consists of three communications; from sensor to PID controller, from PID controller to actuator, and from actuator to PID controller. The processing time depends on the actual device. With an assumption of practical processing time on sensor, PID controller, actuator correspond to 30, 45, and 90 ms, allowed time for each communication is 45 ms according to Eq. (1.1).

$$\frac{300-(30+45+90)}{3}=45 \tag{1.1}$$

In FF-HSE [11] specification, which is designed to utilize typical fieldbus protocol (FF-H1) on Ethernet, the typical packet size is up to 256 Byte. In Shell Nanhai, the world largest class of Petro chemical plant, there are nine Areas and 16,000 Field Devices. Thus, according to Eq. (1.2), each Area has 1,777 Field Devices on average.

$$\frac{16{,}000}{9}=1{,}777 \tag{1.2}$$

When evenly classifies the Field Devices to sensor and actuator, around 888 PID control loop could be built. When all the PID control loop run in 300 ms cycle, three times of 888 communications are performed in 300 ms. This means the PID control traffic causes up to 18.186 Mbps per Area from Eq. (1.3).

$$\frac{888\times3\times256\times8}{0.3}=18{,}186{,}240 \tag{1.3}$$

The traffic of PID control could be considered as static with no dynamics on geometrical and time aspect, since the sensors and actuators are fixed equipment in general. In addition, the related devices transmit packets with fixed size and fixed period.

Mobile HMI is an emerging application in PA. The PA users commonly require HMI in Distributed Control System (DCS) to respond within one second, since HMI is an interactive application. If the response time exceeds one second, the operator may repeat the same operation since he/she may consider the command is not transmitted properly for example. Such operation would cause mis-operation. The common HMI is fixed in the control room (it is called static HMI in this study). Thus, it is not dynamic from a geographical point of view. However, the Mobile HMI is an application that works on portable devices. The operators take the devices to a manufacturing place, which is so called Field, where Field Devices or actual manufacturing facilities are installed. The Mobile HMI could be developed separately from fixed HMI. However, as the easiest way, the HMI would work on remote desktop technology at the first phase of Mobile HMI. In this case, the desktop image is transmitted from control room to portal device in the Field. In general, the remote desktop technology transmits only required amount of packets to reduce the total traffic regarding remote desktop. Thus, the amount of Mobile HMI traffic differs time-by-time. In addition, the device appears, disappears and moves since it works on portable devices. This means, the traffic from Mobile HMI has dynamisms from both time and a geographical point of view.

1.2 Research Statement

As was previously mentioned, the PA production system will utilize Industrial Backhaul to transmit data of Level 1, 2 and 3 due to the deployment of WSN, Mobile HMI, and Industry 4.0. This data has different requirements for real-time communications. The Industrial Backhaul is required to satisfy all of the requirements simultaneously. Especially, the real-time communications are required not only by static traffic but also dynamic traffic. In addition, the Industrial Backhaul has restriction to install additional cable, thus the bandwidth needs to be utilized in an efficient manner. Even in the plant, the threats of the targeted cyber attacks become obvious. Since the control of PA depends on real-time communications, like PID control and HMI, disturbance on real-time communications could result in the irreparable accident, like explosions. Especially, the Industrial Backhaul could be a target of cyber attacks, since various users can transmit packets with various purposes. Thus, the real-time communications in the Industrial Backhaul shall be protected from the cyber attacks.

As the consequence of the situation mentioned above, dependable real-time communications management, that provides efficient bandwidth utilization and secure QoS functions, are required for PA operations over the Industrial Backhaul. The objective of this study is defined as "to provide dependable real-time communications management for PA operations over Industrial Backhaul". The targeted real-time communications traffic of this study is related to Level 1 or Level 2 functions.

To achieve the real-time communications management five essential functions are identified. The functions are: 1) bandwidth control, 2) bandwidth allocation, 3) assessment of real-time communications, 4) prevention of unauthorized bandwidth allocation, and 5) prevention of QoS Spoofing Attacks. As the result of a survey on existing research, it was found that sufficient methods for 3) assessment of real-time communications and 5) prevention of real-time communications are missing. Therefore, this study developed three novel methods to provide the missing functions. The methods are "Propagation time monitoring QoS with OpenFlow (pmqFlow)", "Secure QoS(sQoS)", and "OpenFlow based secure QoS(OFsQoS)".

pmqFlow addresses the efficiency of bandwidth utilization by dynamic bandwidth allocation while assessing the real-time communications. sQoS addresses prevention of QoS Spoofing Attacks by authenticating the packets on each hop. OFsQoS addresses the scalability and flexibility of sQoS to be utilized in practical Industrial Backhaul. Each method has been prototyped and evaluated. As the result of evaluation, the developed methods provide the required functions to manage the dependable real-time communications management. This dissertation introduces the design and evaluation of the developed methods in addition to the survey of related work.

1.3 CONTRIBUTIONS

Figure 1.3 depicts the contribution of this study for dependable real-time communications. Practically, the packet routing and amount of traffic change dynamically. From the real-time communications reliability for dynamic network perspective, the existing research provides dynamic configuration. However, the research does not allow assessing the real-time communications. This means, the management functions of the real-time communications cannot know the appropriate timing to reconfigure. In addition the existing research cannot know the appropriate amount of bandwidth to be allocated, since the research cannot know whether the allocated amount satisfies the real-time communications requirement. Thus, the network dynamism can impact the real-time communications.

From security perspective, some existing research provides secure bandwidth allocation. However, the QoS Spoofing Attacks are not prevented. This means malicious packets can impact the real-time communications by consuming the properly allocated bandwidth.

This study extends the accuracy of bandwidth allocation from the view point of allocation timing and allocation amount by assessing the propagation time of real-time communications. In addition, this study extends the independency of allocated bandwidth for real-time communications by preventing QoS Spoofing Attacks. The propagation time assessment and Prevention of QoS Spoofing Attacks are not provided by existing research. Thus, this study extends the dependability of real-time communications in dynamic networks by the two features. The extension satisfies the requirement of real-time communications in PA.

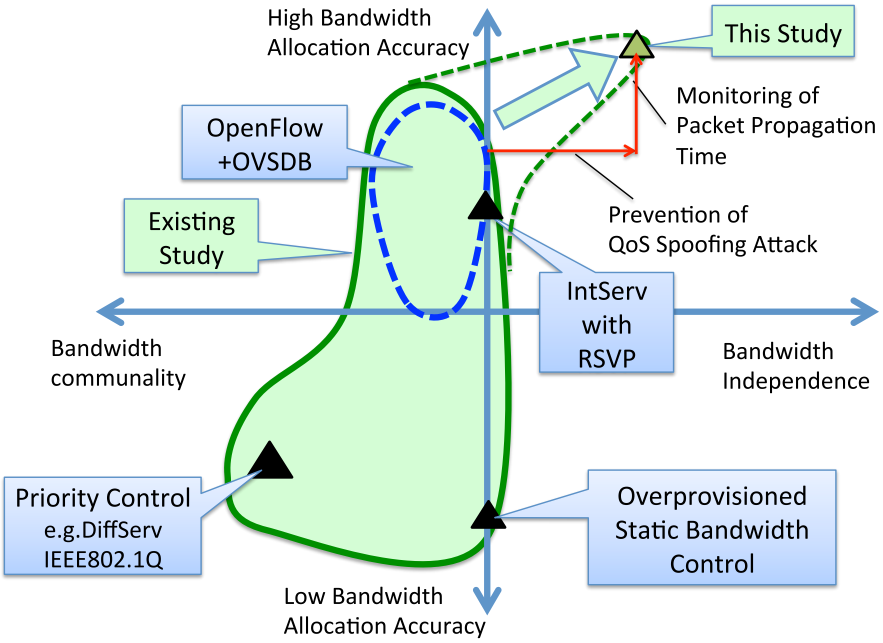


FIGURE 1.3 CONTRIBUTION OF THIS STUDY FOR DEPENDABLE REAL-TIME COMMUNICATIONS

1.4 ORGANIZATION OF DISSERTATION

This dissertation consists of seven chapters. This chapter describes background and objectives of this study. Chapter 2 presents existing related work. Chapter 3 presents concrete challenges to achieve the objectives. In Chapter 4, 5, and 6, the methods developed to satisfy the challenges are presented. Specifically, Chapter 4 describes the method to manage the real-time communications based on actual bandwidth usage and propagation time. Chapter 5 presents a packet authentication method on intermediate device to prevent QoS Spoofing. Chapter 6 presents a new plane to be added into OpenFlow available networks in addition to the Data Forwarding Plane and the Network Control Plane; the plane is called Packet Authentication Plane. The Packet Authentication Plane allows flexible installation of packet authentication method presented in Chapter 5. Lastly, Chapter 7 summarizes the achievements of this study and future challenges to conclude.

Chapter 2

RELATED WORK

2.1 REQUIRED FUNCTIONS FOR DEPENDABLE REAL-TIME COMMUNICATIONS MANAGEMENT

This subsection introduces the required functions to achieve dependable real-time communications management in the Industrial Backhaul, which has particular requirements in order to avoid a serious accident. The required functions are listed below. The first three functions come from a feedback control viewpoint. The feedback control is commonly used in PA since it is employed in the PID control loop. The last two functions come from a security viewpoint. In PA, corresponding security features are provided physically. Thus, the functions allow taking same approach with the control in PA. From the PA viewpoint, all of these five functions are necessary and sufficient for the dependable real-time communications management. The remaining subsections of this chapter introduce the related work addressing the functions. As well as summarize the current status of required functions to clarify the position of this study.

(1) BANDWIDTH CONTROL

To provide real-time communications for a particular packet flow within the physically shared network with variable traffic, the flow should be allocated dedicated bandwidth. Otherwise, other traffic could disrupt real-time communications of the flow. In this sense, bandwidth control is an essential function.

(2) Bandwidth Allocation

When using a bandwidth control function, particular amount of bandwidth needs to be allocated to the flow that requires real-time communications. The bandwidth allocation could be done either statically or dynamically. To deal with dynamic traffic, dynamic bandwidth allocation is required for efficient bandwidth utilization, since the static bandwidth allocation would allocate peak bandwidth regardless of the actual utilization. To provide dynamic allocation, automatic allocation is useful for timely configuration. Thus, automatic bandwidth allocation is required.

(3) Assessment of Real-time Communications

To manage the real-time communications, the actual situation needs to be assessed. The assessed information shall be feedback to configuration. In this study, the situation of real-time communications is assessed by propagation time of packets to know if the packet arrives within the allowed time. The assessment shall be done when configuration is made. In addition, the continuous assessment is needed while the real-time communications last. When the packet cannot arrive within allowed time, it is considered as timeout (Timeout). The threshold of Timeout depends on the flow.

(4) Prevention of Unauthorized Bandwidth Allocation

To use bandwidth control, the bandwidth is an essential resource of the network. And it is a limited resource in the Industrial Backhaul. If an attacker reserves the bandwidth improperly, the proper flows requiring real-time communications could not be allocated for the required bandwidth. Thus, improper bandwidth allocation needs to be prevented.

(5) Prevention of QoS Spoofing Attacks

Even when the bandwidth is allocated for proper flow, improper packets can consume the bandwidth by spoofing the real-time communication packets. Thus, the allocated bandwidth shall be protected from the spoofing packets. Most QoS methods inspect packet attributes to dispatch the packet to a particular bandwidth. Thus, the inspected attributes need to be protected from the spoofing. In addition, even the attributes are protected, plenty of copies of the proper packet could disrupt real-time communications; that are called replay attacks. The replay attacks shall be prevented.

2.2 BANDWIDTH CONTROL

The methods to differentiate the quality of communication are called QoS methods. The methods are classified into bandwidth control and priority control. The bandwidth control allocates dedicated bandwidth to a particular flow to avoid disturbance caused by other flows. The priority control has an assumption that the bandwidth is shared with multiple flows. Thus, the priority control classifies the multiple flows into the limited number of common queues on output port. It results in the flow aggregation. The typical priority control methods are IEEE802.1Q [12] and DiffServ.

Figure 2.1 depicts IEEE802.1Q header. IEEE802.1Q allows administrator to define up to 4,096 Virtual LAN (VLAN) with 12 bit of VLAN Identifier (VID). IEEE802.1Q also allows classifying packets into eight classes of priority, since it has only 3 bit of Priority Code Point (PCP). Thus, each output port has only eight queues. In addition, the queues are shared by traffic belonging to different VLANs.

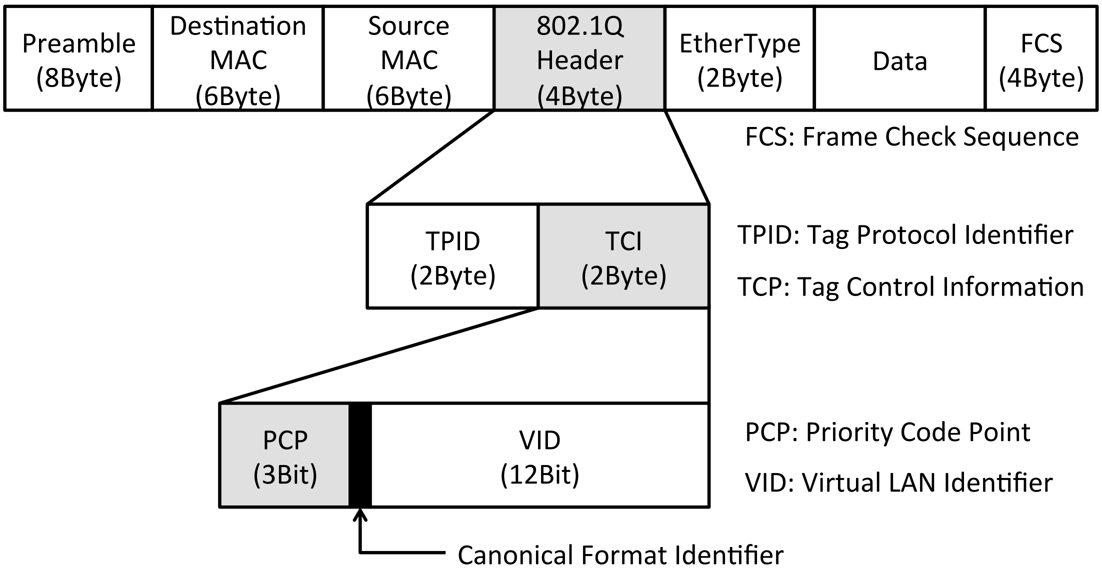


FIGURE 2.1 FORMAT OF IEEE802. 1Q HEADER

When assigning a queue to each real-time communication application, eight queues are not sufficient for Industrial Backhaul since there are more flows to be differentiated. For example, the typical PA application protocol of FF-HSE has three types of communications with different characteristics (i.e., Pub/Sub, Client/Server, Report Distribution). In addition, other traffic requires real-time communications like IP phone traffic, Mobile HMI traffic, network management traffic, Video Camera traffic, MOM applications, and so on.

DiffServ also allows priority control according to the DSCP value (6 bit) on each packet. Although the DSCP allows specifying 64 levels of priority, the DSCP values assigned for standard Per Hop Behavior (PHB) are only 32 as shown in Table 2.1. Table 2.2 represents the detail PHB allocated to the DSCP value in Pool 1 (Standard Action).

TABLE 2. 1 DSCP ALLOCATION SPACE

Pool	Codepoint Space	Assignment Policy
1	xxxxx0	Standard Action
2	xxxx11	Experimental or Local Use
3	xxxx01	Experimental or Local Use (*)

where x could be “0” or “1”
(*) may be utilized for future Standards PHB as necessary)

TABLE 2. 2 PHB MAPPING OF POOL 1 DSCP

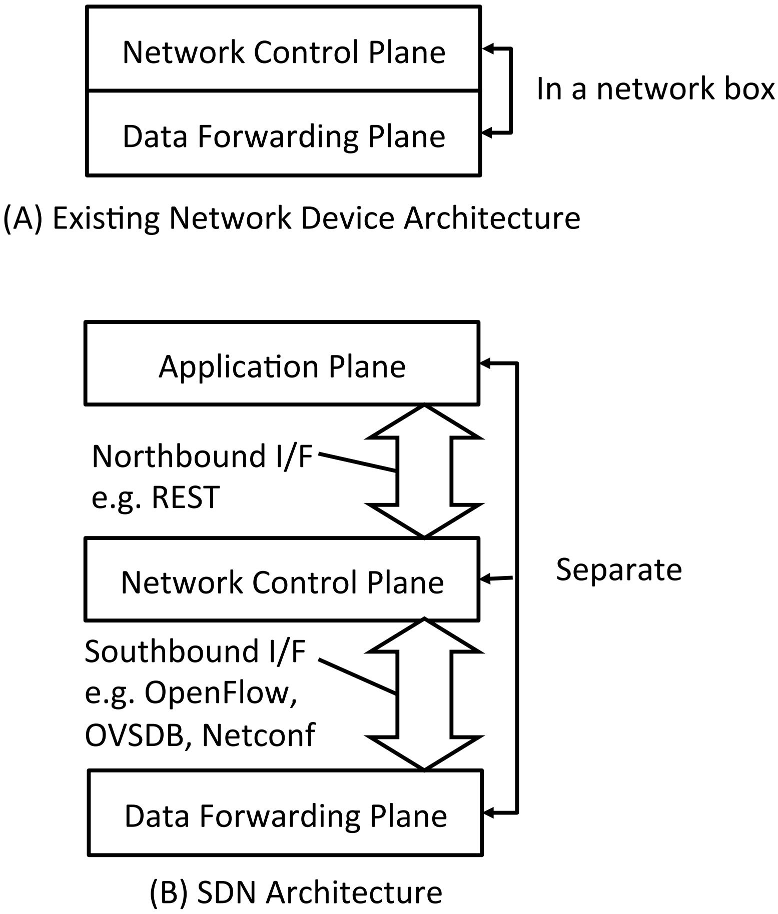
DSCP	PHB
000000	Default PHB
xxx000	Class Selector PHB NOTE: “xxx” represents class (1, 2, 3, 4, 5, 6, 7)
cccdd0	Assured Forwarding (AF) PHB NOTE:“ccc” represents class (1, 2, 3, 4) “dd” represents drop precedence (1, 2, 3)
101110	Expedited Forwarding (EF) PHB

As shown in Table 2.2, the DSCP value in Pool 1 could specify the class with top three bits. Since the output queue with dedicated bandwidth could be assigned to each class, the maximum number of commonly available queue is nine (when use all the Class Selector PHB). It is not sufficient for Industrial Backhaul as the same reason of PCP.

Since the priority control is designed to aggregate multiple flows into a class, when using priority control the real-time communications could be disturbed by other traffic. In addition, low priority packets could cause jitter of forwarding time on high priority packets. Thus, the priority control represented by IEEE802.1Q and DiffServ is not appropriate for the PA related flows in the Industrial Backhaul, which requires ensured real-time communications.

The bandwidth control method is represented by Integrated Service (IntServ) [13]. IntServ can eliminate the possibility of disturbance from other traffic by allocating the dedicated bandwidth to a particular flow. Thus, it fits to the purpose of providing ensured real-time communications.

A significant paradigm shift in network control functions, which is so called Software Defined Network (SDN), is currently emerging. Before SDN, static QoS configuration and static/dynamic routing control are mainly used for network configuration. Most of the dynamic routing control employs versatile routing algorithms. In contrast, SDN separates the network control functions and data forwarding functions. The network control functions provide network configuration to the data forwarding functions from the outside. This architecture allows users to operate network flexibly with user defined policy and algorithms. In addition, SDN allows applications to input their requirements on network operation by providing the interface to applications. Figure 2.2 (a) and (b) correspond to the existing network architecture and SDN architecture. In SDN architecture, the packet forwarding functions are called the Data Plane (this study calls it Data Forwarding Plane). The upper layer, which controls the network, is called the Control Plane (this study calls it Network Control Plane). In addition, the upper layer of the Network Control Plane is called the Application Plane, which sends particular requests to the Network Control Plane. The interface between the Network Control Plane and the Data Forwarding Plane is called the “southbound IF”. The typical protocols utilized for the southbound IF are OpenFlow [14], OVSDB [15], and NETCONF [16]. The interface between the Application Plane and the Network Control Plane is called the “northbound IF”. The typical protocol utilized for the northbound IF is REST. The SDN architecture mentioned above allows network control based on the traffic condition and request from applications. SDN allows packet routing based on the information other than destination MAC or IP address, in contrast to existing architecture. Moreover, SDN allows classifying packets based on any combination of packet attributes, in contrast to existing QoS methods represented by DiffServ [17], which classifies packets by Differentiated Service Code Point (DSCP) [18].



(A) Existing Network Device Architecture

(B) SDN Architecture

FIGURE 2.2 NETWORK ARCHITECTURE

In an OpenFlow based network, the functions controlling the network are centralized to a device, which is so called OpenFlow Controller (OFC), to enable flexible and sophisticated network control. OFC manages FlowTables in OpenFlow available Switch (OFS), which contains packet-forwarding rules. Originally, the OpenFlow specification was focusing on routing functions. However, OFS (compliant to version 1.3 or later) can guarantee minimum bandwidth for each output queue. In other words, OpenFlow can provide bandwidth control functions. The OpenFlow architecture allows OFSs to classify packets into flows to provide per flow routing control and per flow bandwidth control according to the rule provided by OFC.

The OpenFlow (Version 1.3.1) allows users to define flows by combination of either of 40 attributes in the packets. The examples of the attributes are source/destination address, protocols (TCP or UDP), source/destination port, DSCP, flow label [19]. In addition, the OFSs can prepare output queues with dedicated minimum bandwidth on any output network port.

Bandwidth control is achieved by specifying the output network port and output queue for each flow that is defined by users based on their detailed requirements. Thus, OpenFlow allows detailed and flexible bandwidth control. This means, the bandwidth control represented by OpenFlow and IntServ is appropriate for the PA related flows in the Industrial Backhaul.

2.3 BANDWIDTH ALLOCATION

To use the bandwidth control functions, a particular amount of bandwidth shall be allocated to the target flow. When allocating the bandwidth manually, it is difficult to configure the bandwidth in a timely manner, since the entire intermediate network devices on the path need to be configured. In general, when using static bandwidth control, the bandwidth is allocated more than the estimated amount to prepare the fluctuation of the traffic. Ref. [20] points that 50% of unused bandwidth is allocated in general when using static bandwidth control. Such allocation method is called over provisioning. Over provisioning in a static manner that makes a strong declaim in the bandwidth efficiency. Thus, such inefficiency is not allowed in the Industrial Backhaul within a vast site.

RSVP [21] has been designed to allocate bandwidth to a particular flow automatically. In RSVP specification, end devices reserve the required bandwidth on the routers in the path when the communications is initiated. Even though the reservation needs to follow the path change, RSVP cannot know the changes in path since RSVP is not involved to routing function. Thus, the bandwidth reservation procedure is repeated periodically rather than an event driven manner. This means the bandwidth control does not work for a while when the path is changed. It may allow packet loss or big jitter on real-time communications. Thus, the RSVP is not sufficient for PA related flows in the Industrial Backhaul, since bandwidth allocation shall be maintained in an event driven manner.

In OpenFlow environment, routing information can be centralized on OFC. The routing information is helpful to maintain bandwidth allocation automatically. However, bandwidth allocation is outside the scope of OpenFlow. Thus, OpenFlow itself cannot configure the bandwidth on output queue.

Ref. [22] proposes API to automate the QoS configuration on OpenFlow network. This approach allows the ability to combine routing configuration to QoS configuration. However, it does not observe the actual propagation time. In addition, the proposed approach does not allocate bandwidth dynamically but utilizes multiple output queues on which bandwidths are statically pre-allocated. The allocated bandwidth on OFS guarantees the minimum bandwidth exclusively. Thus, the bandwidth would not be used by un-assigned traffic. Preparing static exclusive bandwidth is not allowed in the Industrial Backhaul since it declines bandwidth efficiency. Thus, dynamic bandwidth allocation is desired.

To allow bandwidth configuration on OFSs, OF-CONFIG [23] and OVSDB, and [24] are developed. Combining these protocols with OpenFlow allows dynamic bandwidth configuration as required. However, both of them provide framework to configure the OFSs. Algorithms to allocate the bandwidth are required. In addition, OpenFlow allows OFC to monitor bandwidth usage of each flow.

2.4 Assessment of Real-time Communications

Ref. [22] and OpenQoS [25] proposes the means to assess the situation of network. However, both of them monitor the bandwidth utilization of OFSs rather than packet propagation time. OpenFlow and OVSDB allow monitoring the bandwidth utilization as well. Thus, the combination of OpenFlow and OVSDB or OF-CONFIG allows monitoring bandwidth utilization. However, both combinations do not allow real-time communications assessment. RTP [26] addresses to dynamic real-time communications. In [26], RTCP is defined to control RTP. RTCP provides reporting function of packet loss, jitter of arrival time, and propagation time by involving sender device and receiver device. Based on the report packet sender adjust the transmission rate. It is also possible to send the report to network management system and reconfigure the network equipment based on the report. The reported propagation time is based on Round Trip Time (RTT) as represented in Figure 2.3. In RTCP sender obtains the RTT (A in Figure 2.3) by subtracting the timestamp at transmission of Sender Report (SR) from the timestamp at reception of Receiver Report (RR). The receiver obtains the B by subtracting the timestamp at reception of SR from the timestamp at transmission of RR. The sender could obtain the RTT propagation time by subtracting B from A.

However, many of the real-time communications in PA utilize UDP one-way packet. For example, the PID control utilizes Publisher/Subscriber (Pub/Sub) type communications, which is one-way communication using UDP packet. In addition, some particular protocols designed for Remote Desktop, which would be used for Mobile HMI at the first phase, leverage UDP packet. Typical examples of such protocols are Remote Desktop Protocol (RDP) or PC over IP (PCoIP). The reported time cannot be used for such one-way communications since network does not guarantee symmetric path and symmetric traffic amount for both directions of a round trip packet. Thus, unrelated network devices and traffic can affect the RTT based propagation time. In addition, it is impossible to find the device to be reconfigured by end-to-end measurement. With the reasons mentioned above, RTCP is insufficient for managing real-time communications.

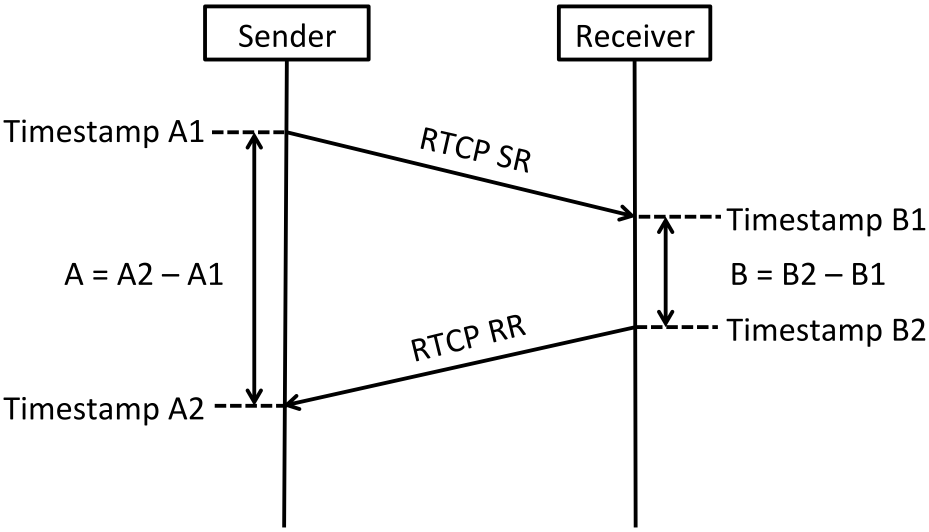


FIGURE 2.3 ROUND TRIP TIME MEASUREMENT FLOW IN RTCP

NTP [27] and IEEE1588 [28] are the protocols to synchronize the clock among the network devices. Those protocols also have a function to obtain the packet propagation time. However, the obtained propagation time is based on RTT as same as RTCP. Thus, with the same reason of RTCP, the NTP and IEEE1588 are insufficient for managing the real-time communications.

2.5 PREVENTION OF UNAUTHORIZED BANDWIDTH ALLOCATION

When allowing the bandwidth allocation automatically, a risk of improper bandwidth allocation by unauthorized user is concerned. If an unauthorized user can reserve the dedicated bandwidth improperly, a proper user cannot be allocated enough bandwidth required for real-time communications. It is considered a type of the real-time communications disturbance.

RSVP-SQoS [29] proposes a method to prevent the unauthorized bandwidth allocation by

RSVP. RSVP-SQoS authenticates path discovery message and bandwidth reservation message with a digital signature. In addition, RSVP-SQoS introduces acknowledge messages for proper message. This approach works for preventing unauthorized bandwidth allocation.

In OpenFlow available networks, bandwidth allocation is configured from the Network Control Plane to the Data Forwarding Plane by using OVSDB or OF-CONFIG. By using Transport Layer Security (TLS) [30] for the communications of OVSDB and OF-CONFIG, the devices on the Network Control Plane and the Data Forwarding Plane could be mutually authenticated. The Network Control Plane decides the bandwidth allocation based on its own algorithms or requests from Application Plane. The communications between the Application Plane and the Network Control Plane could be authenticated by TSL as well. In addition, the Network Control Plane could authorize the requests by involving the authorization server such as RADIUS [31] and Diameter [32]. Thus, the OpenFlow available networks could prevent the unauthorized bandwidth allocation.

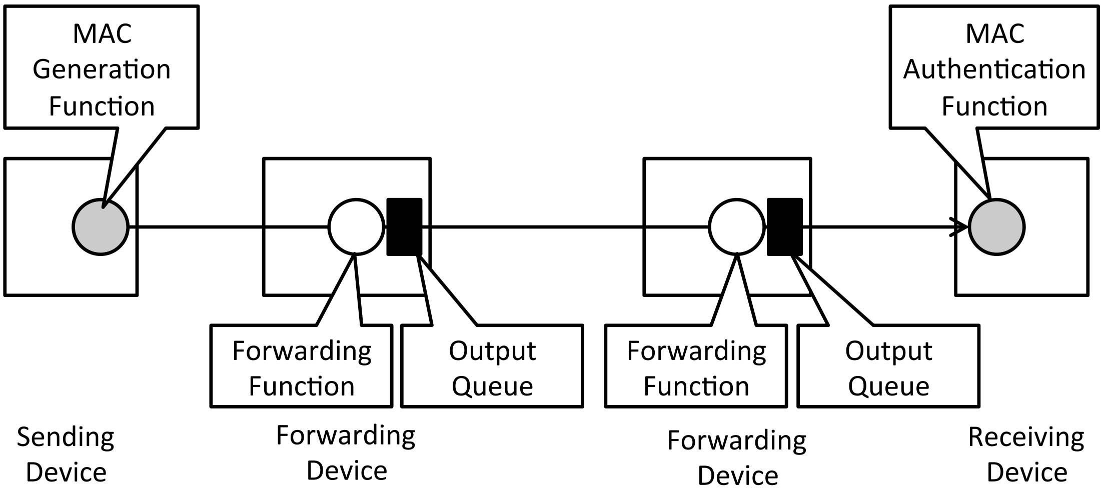
2.6 Prevention of QoS Spoofing Attacks

The intermediate equipment needs to distinguish the packet to classify in both bandwidth control and priority control method,. For example, IntServ introduces flow definition by transport connection between a given host pair: this means using packet attributes of (source address, destination address, protocol, source port, destination port). In addition, DiffServ utilizes DSCP to distinguish the packet priority. Thus, if an attacker transmits a number of improper packets that spoof the real-time communications packets, it is possible to disrupt real-time communications. In this study, such attacks are called QoS Spoofing Attacks. The essential reason of QoS Spoofing Attacks is the impossibility of spoofing packet detection on intermediate equipment before being consumed the output queue. Thus, a method to detect the spoofing packets on the intermediate equipment is required to prevent the QoS Spoofing Attacks.

The most famous packet authenticating security protocol is IPsec [33] including IP Authentication Header (AH) [34] and IP Encapsulated Security Payload (ESP) [35]. Especially, AH protects the packet including IP header. However, the AH does not protect the DSCP field, since DSCP value is can be changed by the intermediate network equipment in the path to the destination. In addition, IPsec is designed based on End-to-End security architecture. This means only the receiving device can authenticate the packets as represented in Figure 2.4. Thus, the intermediate network equipment cannot evaluate whether the packet is modified or not. The other security protocols such as Secure Socket Layer (SSL) [36], TLS cannot be used for packet authentication on the path with the same reason.

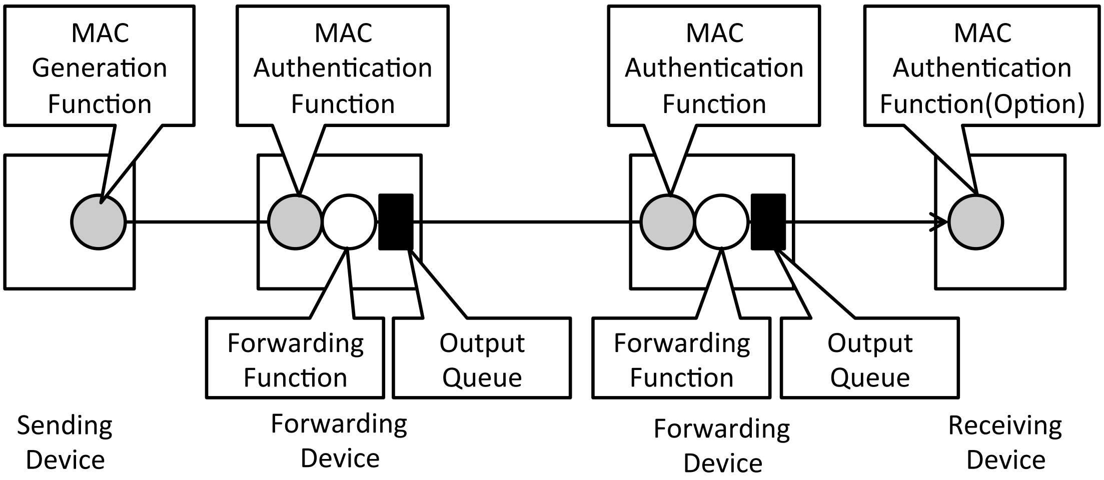
To prevent QoS Spoofing, Hob-by-Hop packet authentication is useful rather than End-to-End authentication. As Figure 2.5 indicates, the MAC Authentication Function shall authenticate packets before output queue is consumed. Lagutin has developed Packet Level Authentication (PLA) [37] to allow packet authentication on intermediate network devices. PLA has a good concept to detect the malicious packets by protecting the whole packet including IP header to IP payload. However, it does not provide sufficient QoS protection. PLA does not authenticate some IP header fields that could be changed in the path to the destination, like IPsec. Some QoS methods use the rewritable fields. For example, DiffServ utilizes DSCP filed to prioritize the packets. Ref. [38] indicates that Flow Label could be used for QoS as well. The field to classify the packet depends on the QoS method. Therefore, to prevent the QoS Spoofing Attacks, packet authentication shall allow protecting the field utilized by QoS methods somehow, while they could be rewritable. For example, specifying the protected field would be helpful to allow utilizing any field for QoS. However, PLA does not have such function. In addition, PLA can introduce proprietary priority control like PCP of IEEE801.Q. However, priority control is insufficient to ensure real-time communications as was previously mentioned. As the consequence, the PLA has good concept of hop-by-hop packet authentication. However, PLA is not appropriate since it does not work with bandwidth control.

Karlof et al. has developed TinySec [39] to provide Hop-by-Hop packet authentication as well as PLA. TinySec has the following features: a) TinySec is Designed for low power WSN and has small impact (10%) on energy consumption, delay, and bandwidth, b) TinySec generates Message Authentication Code (MAC) with secret shared key to verify packet integrity. c) TinySec authenticates packet sender, d) TinySec encrypts message, e) TinySec prevents replay attacks. It protects whole packet by using hash-based algorithm, which is similar to Keyed-Hashing for Message Authentication code (HMAC) [40]. However, it uses the proprietary packet format to address a). Therefore, it is not compliant to any of Ethernet, WiFi nor IP protocols. In addition, it does not provide QoS functionalities. Thus, TinySec is insufficient for Industrial Backhaul.



MAC: Message Authentication Code

FIGURE 2.4 PACKET AUTHENTICATION IN END-TO-END SECURITY



MAC: Message Authentication Code

FIGURE 2.5 PACKET AUTHENTICATION IN HOP-BY-HOP SECURITY

2.7 SUMMARY OF RELATED WORK

As was previously mentioned, a IntServ or OpenFlow available switch can provide sufficient “(1) Bandwidth Control”. OVSDB or OF-CONFIG can provide “(2) Bandwidth Allocation”. OVSDB or OF-CONFIG over TLS could provide “(4) Prevention of Unauthorized Bandwidth Allocation” as well as RSVP-SQoS. However, “(3) Assessment of Real-time Communications” cannot be satisfied, since existing work that provides propagation time obtains the propagation time based on RTT. In addition, “(5) Prevention of QoS Spoofing Attacks” could not be satisfied, since existing work that provides Hop-by-Hop packet authentication does not address making the authentication work with bandwidth control.

As a consequence of this subsection, to achieve the dependable real-time communications management, following two functions shall be satisfied: “(3) Assessment to Real-time Communications”, and “(5) Prevention of QoS Spoofing Attacks”.

Chapter 3

CHALLENGES AND GOALS

3.1 CHALLENGES

As described in Subsection 1.2, the objectives of this study is to provide the dependable real-time communications management by monitoring the packet propagation time and preventing disturbance on real-time communications for PA operations over an Industrial Backhaul. To achieve the objectives, five functions are required as introduced in Subsection 2.1. However, two required functions are still insufficient.

Thus, to achieve the objectives, this study works on these two functions of “Assessment of Real-time Communications”, and “Prevention of QoS Spoofing Attacks”. The detailed challenges are described below.

3.1.1 ASSESSMENT OF REAL-TIME COMMUNICATIONS

As shown in Subsection 2.2 to 2.4, bandwidth control function is helpful to provide real-time communications. The related work represented by [22] and [24] provides monitoring functions of output queue of OpenFlow switches to reconfigure the bandwidth allocation based on the actual queue usage. However, to assess whether the requirement for real-time communications is satisfied a method to monitor the actual propagation time is required, since the propagation time is the value to assess the real-time communications directly. In addition, the monitored range shall be limited to the path of the targeted real-time communication flow to avoid the impact of unrelated devices and traffic.

The means to monitor the bandwidth utilization obtains the utilization rate by dividing the difference of two accumulated transmitted traffic amounts by the difference of monitored times. The value obtained by this means is average value within the given time period. It cannot detect the peak of big traffic in relatively shorter period for monitoring cycle; such traffic is called spike traffic in this study. Since the spike traffic can cause Timeout of the real-time communications, the monitoring methods of propagation time shall be able to detect the spike traffic.

3.1.2 Prevention of QoS Spoofing Attacks

As shown in Subsection 2.6, the intermediate network equipment needs to classify the packets to differentiate the forwarding policy. To classify the packets on the intermediate network equipment, the attributes on the packets are inspected. However, the attackers can spoof the attributes of the packet, and it allows disrupting real-time communications. To prevent such disruption of QoS Spoofing Attacks, the intermediate network equipment needs to detect the spoofing packets before the output queue allocated to the real-time communications, such as PID control, is consumed. On the other hand, many of the existing packet authentication mechanisms take End-to-End security architecture. In such architecture, the intermediate network equipment cannot authenticate the packets. PLA and TinySec have addressed Hop-by-Hop authentication methods, however they are insufficient to protect the bandwidth control. Thus, Hop-by-Hop packet authentication method addressed protecting the bandwidth control is required to prevent the QoS Spoofing.

3.2 Goals

As mentioned in Subsection 1.2 the objectives of this study is to provide the dependable real-time communications management into the Industrial Backhaul. The dependable management requires reliability to ensure the real-time communications even for the dynamic traffic while satisfying the efficient bandwidth utilization. In addition, the management needs to protect the real-time communications from malicious traffic, since the Industrial Backhaul is considered as an untrusted network, which interconnects multiple Areas with different policies.

Based on the challenges mentioned above, this study develops two novel methods of "management method for real-time communications based on packet propagation time monitoring" and "Hop-by-Hop packet authentication method to protect the output queue" to address "Assessment of Real-time Communications" and "Prevention of QoS Spoofing Attacks" respectively. The detailed goal of each method is described below.

3.2.1 Management Method for Real-time Communications Based on Packet Propagation Time Monitoring

This method provides dynamic real-time communications management function based on the monitored packet propagation time automatically. This method shall allow measuring a particular range of path rather than measuring RTT of the packet, since the actual real-time communications in PA take one-way communications. This kind of measuring allows obtaining the accurate propagation time. This method shall allow detecting spike traffic to protect real-time communications of them. These are novel features of this method. A proposed method and detailed goals are described in Chapter 4.

3.2.2 Hop-by-Hop Packet Authentication Method to Protect the Output Queue

This method provides packet authentication on intermediate network equipment to prevent QoS Spoofing Attacks. This method needs to protect the bandwidth allocated to the real-time communications by detecting and discarding the spoofing packets on intermediate network equipment before the allocated bandwidth is consumed. In addition, the authentication process shall be provided within the allowed time for PA operation. This is a novel feature of this method. The methods are described in Chapter 5 and Chapter 6. In Chapter 5, Hop-by-Hop authentication method on the router is presented. In Chapter 6, integration of the Hop-by-Hop packet authentication method described in Chapter 5 with bandwidth control on OpenFlow available networks is presented. The detailed goals are described in each Chapter.

Chapter 4

PMQFLOW: OPENFLOW BASED REAL-TIME COMMUNICATIONS MANAGEMENT METHOD FOR DYNAMIC TRAFFIC IN PROCESS AUTOMATION

4.1 INTRODUCTION

PA users are currently interested in networked operations in plants. There are many facilities (Areas) in conventional plants and each is operated individually. Connecting Areas distributed within a vast site with networks allows geographically separated environments to be managed remotely. Thus, such networks can improve operational efficiency and safety since they enable equipment installed in physically and chemically dangerous zones to be managed as required. In addition, mobile operation terminals, which are so called mobile HMIs, would allow access to data and applications from places, where instrumental equipment is working. The places are so called Fields. Such data and applications used to be traditionally accessed from central operation rooms. Thus, Mobile HMIs are expected to improve operational efficiency in the field.

The networks connecting multiple Areas in plants are called the Industrial Backhaul. There are some concerns about the Industrial Backhaul. Since the Industrial Backhaul is shared by multiple users for multiple purposes, users are concerned with risks in real-time communications and security. The Industrial Backhaul is commonly built in a ring topology as indicated by the fact that many kinds of network switches designed for plants provide redundancy features based on the ring topology. This is because networks installed on vast Sites are too expensive to build in a mesh topology, which requires many links between Areas. Thus, the network resources in the Industrial Backhaul are limited and need to be used efficiently. To address these concerns, ISA has developed a technical report [4] that defines the network architecture for the Industrial Backhaul. The report suggests introducing QoS methods to the Industrial Backhaul to enable real-time communications.

The ISA-TR100.15.01-2012 Backhaul Architecture model [4] has introduced typical applications requiring real-time communications such as PID control using Field Devices (sensors and actuators), Mobile HMIs, and Video Cameras. PA in this study means all the applications required for plant operations including PID control, Video Cameras, and Mobile HMIs. These applications require real-time communications. However, the characteristics of real-time communications depend on the applications. For example, the traffic for PID control is static and has no mobility. The traffic for Video Cameras is dynamic and has no mobility. The traffic for mobile HMIs is dynamic and has mobility. Static HMIs monitor and operate equipment as required. PA users commonly require HMIs to respond within one second to avoid mis-operation. Thus, mobile HMIs are also required to respond within one second.

There has been much existing work on configuring network equipment to support dynamic real-time communications. However, it has not provided the means to confirm whether the given configurations satisfy the requirements for real-time communications. Thus, the configurations have not been dependable. In other words, it has been impossible to provide dependable management to real-time communications with the existing work.

The proposed method monitors the elapsed time to propagate packets (propagation time) within a particular range in a path and the utilization of output queues of network devices to detect changes in network status to reconfigure the bandwidth as required. The proposed method was prototyped and evaluated. As a result, it was confirmed that the proposed method could work to ensure dynamic real-time communications.

The rest of this chapter is structured as follows. Subsection 4.2 presents objectives and challenges with this proposed method. Subsection 4.3 presents an overview, Subsection 4.4 presents the design, Subsection 4.5 presents a prototype, Subsection 4.6 explains our evaluation, and Subsection 4.7 presents considerations. Subsection 4.8 concludes the chapter. The proposed method in this chapter is called propagation time monitoring QoS with OpenFlow (pmqFlow).

4.2 OBJECTIVES AND CHALLENGES

The Industrial Backhaul needs to connect Areas distributed throughout a Site, while providing real-time communications to dynamic traffic, as was previously mentioned. Thus, the main objective of pmqFlow was to ensure real-time communications in the Industrial Backhaul to dynamically change PA traffic.

Related work, such as that by Kim et al. [22] and Sonkoly et al. [24] has tried to improve existing QoS methods with automatic QoS configuration functions. However, it has not provided the means to assess whether the provided configurations satisfy real-time communications requirements. This means real-time communications have not been ensured. Thus, pmqFlow has three goals to achieve this objective.

4.2.1 Managing Real-time Communications Based on Packet Propagation Time

The proposed method should assess whether the provided configurations satisfy the requirements for real-time communications by monitoring the propagation time, as was previously mentioned. After assessment, the method should update the configuration as required to ensure real-time communications. The means of assessing the propagation time is a novel feature of pmqFlow.

4.2.2 Managing Real-time Communications Based on Bandwidth Utilization

Bandwidth Utilization has a strong impact on real-time communications. Thus, the monitoring function of Bandwidth Utilization should ensure real-time communications. When the utilized bandwidth exceeds the allocated amount, additional bandwidth should be allocated. When the utilized bandwidth falls below the allocated amount for a while, the unused bandwidth should be identified as over provisioned. Thus, the function should release the unused bandwidth to allocate it to other real-time communications. The observation of Bandwidth Utilization is a required feature to efficiently utilize limited bandwidth.

4.2.3 Recovering within One Second

The required reconfiguration should be completed within the allowed time for the Mobile HMI response time when detecting the timeout for real-time communications. The allowed time is one second, as was previously mentioned. It is impossible to completely prepare for all sudden changes when taking the dynamic configuration approach. Thus, this method should prevent continuous timeouts by reconfiguring the OFSs within one second. This approach minimizes the risk of duplicated operations.

The required performance for reconfiguration depends on the size of the Industrial Backhaul. This study defined a reference plant model in which the Industrial Backhaul connects ten Areas via OFSs and 18,000 Field Devices (site: 260 ha, Areas: 10, Field Devices: 18,000). The reference plant model was defined by referring to one of the world's biggest plants (Shell Nanhai). If pmqFlow satisfies the goals under these conditions, pmqFlow can be applied to most plants around the world.

4.3 Overview of pmqFlow

4.3.1 Fundamental Policy

There are many kinds of devices in plants, such as Field Devices, Video Cameras, and Mobile HMIs. Field devices impose especially strong constraints on computing resources. pmqFlow is designed to leverage intermediate equipment rather than modifying various existing end devices to make pmqFlow an end device agnostic approach.

"Managing real-time communications based on packet propagation time" without changing end devices, network equipment needs to provide functions to monitor propagation time and to dynamically allocate bandwidth to achieve goal (1). In addition, "Managing real-time communications based on bandwidth utilization", network equipment needs to provide functions to monitor bandwidth utilization and dynamic bandwidth allocation to achieve goal (2). OpenFlow available devices can provide a bandwidth utilization monitoring function in addition to ensured bandwidth control. OVSDB allows dynamic bandwidth allocation to the output queues of OFSs. Moreover, equipment needs to provide particular processing to measure the propagation time with network equipment. OpenFlow provides a common framework to modify and forward packets. Thus, the combination of OpenFlow and OVSDB can provide most of the functions required by pmqFlow, such as monitoring bandwidth utilization, reconfiguring bandwidth allocation, and processing functions. Thus, pmqFlow has been designed over this combination.

The Industrial Backhaul is expensive, as was previously mentioned. Thus, bandwidth should be allocated dynamically as required to improve the efficiency of bandwidth utilization. Static over provisioning, on the other hand, improves the reliability of real-time communications even when unpredicted occasions occur, which is a trade-off. pmqFlow has a policy that gives priority to dynamic bandwidth allocation to improve the efficiency of bandwidth utilization, while minimizing the risk of disruptions to real-time communications.

4.3.2 PROPOSED STRUCTURE

Figure 4.1 has an overview of pmqFlow. The Industrial Backhaul interconnects Areas distributed throughout the plant and is built with a ring topology, as was mentioned in Subsection 4.1. Each link can be several kilometers as the plant site is vast. Thus, the links can be comprised of optical fiber cable (1 Gbps). Each Area in Figure 4.1 is represented as an edge network. One of the connected Areas is the central control room. Eighteen hundred of the Field Devices are evenly distributed in the other nine areas (a totally of 18,000 Field Devices in 10 Areas). In addition, Mobile HMIs and Video Cameras are utilized in the plant. According to the Fieldbus Foundation [5] and Success Stories [41], 90 Static HMIs are used in huge scale plants like Shell Nanhai. Thus, nine Static HMIs are evenly assigned to each Area. Mobile HMIs are devices that complement static HMIs. Thus, the number of Mobile HMIs is supposed to be five (half the number of Static HMIs). Video Cameras are considered to be devices that complement sensors. Thus, the number of the Video Cameras is also supposed to be five as Mobile HMIs. In addition, 100 of the PCs are evenly located in each Area for ordinary office work, which do not require real-time communications. As was previously explained, the Industrial Backhaul needs to ensure dynamic and static real-time communications while allowing non real-time communications.

pmqFlow prepares a real-time communication manager (RCM), which monitors propagation time and bandwidth utilization on OFSs to reconfigure them as required to ensure real-time communications. RCM is placed on OFC to immediately utilize collected information.

The OpenFlow specifications recommend that a dedicated channel (secure channel) be prepared for communications between OFC and OFS. However, this is not permitted when dedicated networks are only installed for management purposes at vast plant sites. Thus, the author designed a secure channel so that it overlaid the Data Forwarding Plane provided by OFSs, as was proposed by Koide and Shimonishi [42]. OFSs can provide bandwidth control to the overlaid secure channel since they can deal with overlaid packets as a common flow.

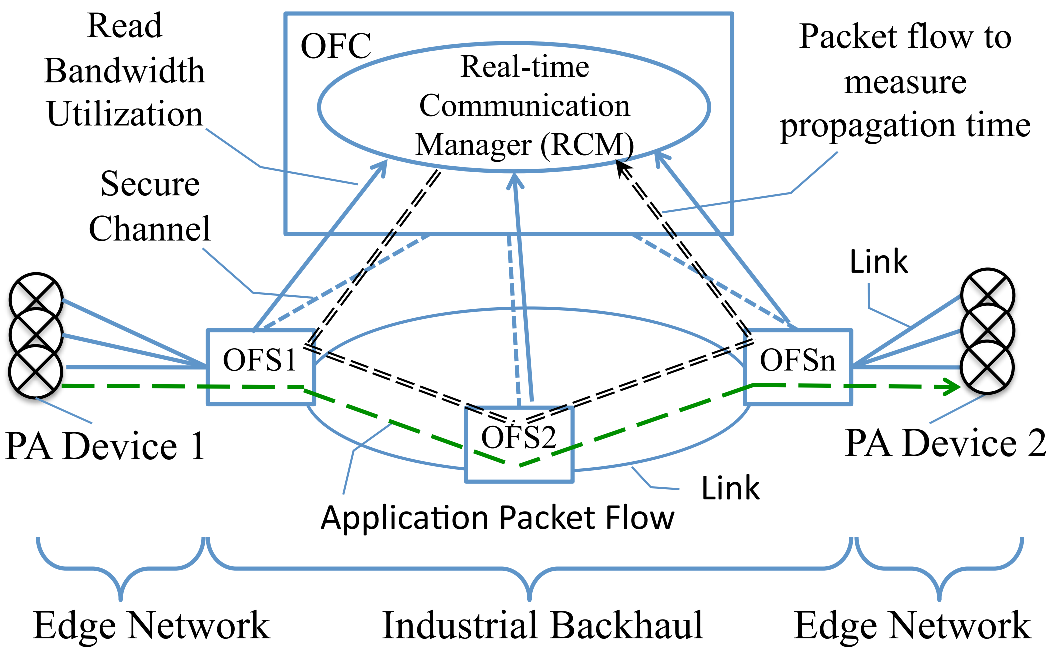


FIGURE 4.1 ABSTRACTED MODEL OF ENTIRE SYSTEM

4.4 DESIGN OF PMQFLOW

pmqFlow monitors the propagation time of packets and the bandwidth utilization of output queues to reconfigure the Allocated Bandwidth to manage real-time communications on the Industrial Backhaul. This subsection introduces the range of communication paths to be monitored and the means of carrying out measurements. In addition, a means of utilizing measured data is introduced. Finally, the role of each function is clarified.

4.4.1 PROPAGATION TIME MONITORING

The range to be monitored should be clarified to achieve the goal of (1) “Managing real-time communications based on packet propagation time”. Figure 4.2 breaks down the end-to-end propagation time into the entity level to analyze the delay making entity based on Bellovin [43] and Prasad et al. [44]. The end-to-end propagation time is represented as Tee. The time from the application to the network driver, and vice versa, is called the overhead. The time from the network driver to the network medium, and vice versa, is called serialization. TS means the sum of the overhead and serialization on the sending device (device-x). Overhead and serialization occur in the reverse direction on the receiving device (device-y). TR means the sum of the overhead and serialization on device-y. The transmitted packet is propagated by more than one medium and forwarded by more than one piece of network equipment such as the switch and router. Each required propagation time is represented as $T_{1,}$ T_2, …, T_n in Figure 4.2. In a similar way, each required forwarding time is represented as F_1, F_2, …, F_n. Their total values are represented as T_{total} and F_{total}. Tee is represented in Eq. (4.1), T_{total} is represented in Eq. (4.2), and F_{total} is represented in Eq. (4.3).

TS and TR would be fixed values when the loads of the devices were fixed. Since T_{total} is the physical propagation time, it is possible to obtain T_{total} when the path is defined. The path is completely managed by OFC in the OpenFlow environment. This means that the TS, TR, and actual T_{total} do not need to be measured since theoretical values can be used. However, F_{total} means the time to forward packets on network equipment such as OFSs. The time will be affected by the amount of traffic. The amount of traffic is not theoretically obtained by OFC, since it depends on applications. Thus, the time to be observed to obtain the propagation time is F_{total}.

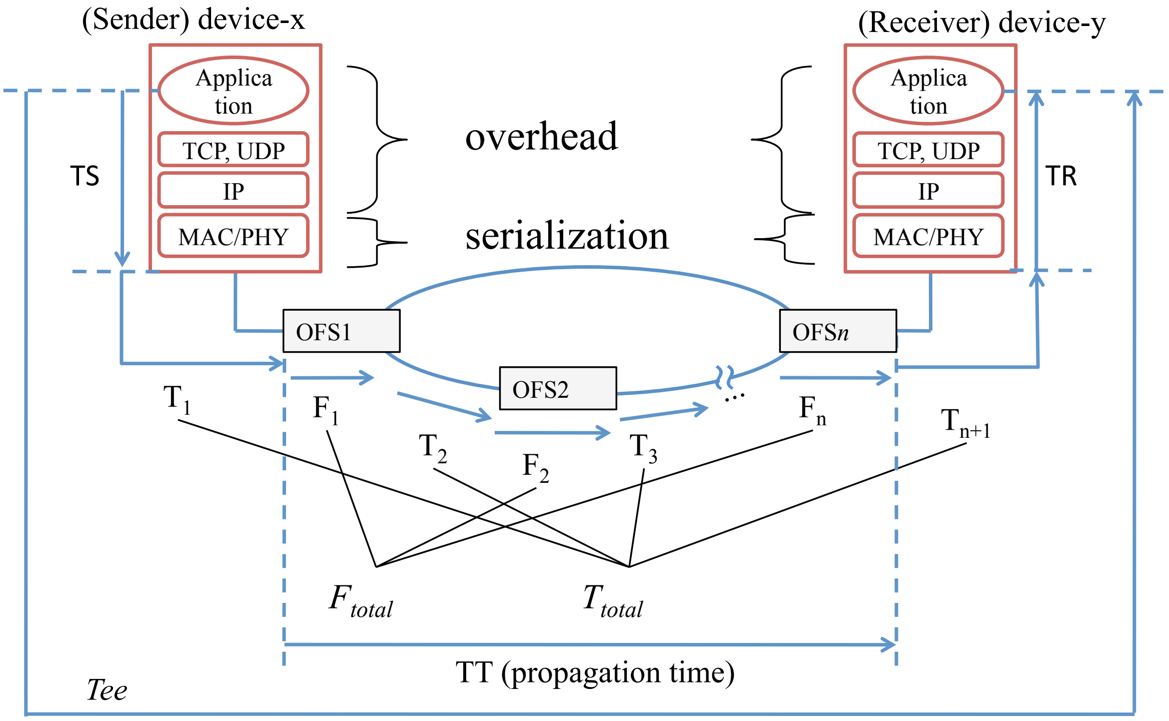


FIGURE 4.2 ANALYSIS OF PROPAGATION TIME

$$Tee = TS + T_{total} + F_{total} + TR \tag{4.1}$$

$$T_{total} = \sum_{k=1}^{n+1} Tk \tag{4.2}$$

$$F_{total} = \sum_{k=1}^{n} Fk \tag{4.3}$$

F_{total} in Figure 4.2 is the sum of the forwarding time from the closest OFS to the sender (OFS1) to the closest OFS to the receiver (OFSn). However, it is inefficient to measure the forwarding time on each OFS in a one-by-one manner. Thus, pmqFlow measures the value of TT, which is given in Eq. (4.4), where T_{total}' means T_{total} without T_1 and T_{n+1}, as shown in Eq. (4.5). TT is equivalent to the total required time from OFS1 to OFSn. Thus, TT is obtained as the difference in the passing times on OFS1 and OFSn.

$$TT = F_{total} + T_{total}' \tag{4.4}$$

$$T_{total}' = T_{total} - (T_1 + T_{n+1}) \tag{4.5}$$

OFC periodically transmits measurement packets to OFS1 to monitor the propagation time. OFS1 forwards the measurement packets to the next hop device. When the measurement packets arrive at OFSn, OFSn sends the packets to OFC. To obtain TT, OFS1 and OFSn obtain timestamps to place them in the packets. Obtaining timestamps in OFSs rather than OFC allows the impact of packet propagation time to be eliminated between OFS and OFC. The monitoring cycle should be defined for each flow based on the requirements for real-time communications.

pmqFlow involves an assumption that the target flows to be monitored are specified by 5-tuples in most cases. In other words, the UDP or TCP port number will be used to specify the target flow since the requirements for real-time communications depend on the application. In addition, the source address and destination address are required to specify the monitored range. However, specifying the address of each device would increase the monitoring load and would consume too much processing time on OFC. Thus, the addresses should be specified with aggregated addresses assigned to each Area to achieve goal (3) of “recovering within one second”. The aggregated address can be represented with a mask. For example, when using an

IPv4 address, the specified tuples would be {source_address=10.1.0.0/16, destination_address=10.0.0.0/24, protocol=UDP, source_port=any, destination_port=1090}; this is equivalent to {source_Area, destination_Area, application}.

4.4.2 Bandwidth Utilization Monitoring

Each flow needs to be assigned to a particular output queue on a particular output network port to provide real-time communications in an OpenFlow environment. Each output queue is allocated a particular bandwidth. While the traffic is less than the Allocated Bandwidth, forwarding is guaranteed. However, if the traffic exceeds the Allocated Bandwidth, the exceeded traffic shares the default queue. This could cause long delays or packet losses depending on the traffic situation in the default queue.

pmqFlow monitors bandwidth utilization for this reason. The targets of monitoring are output queues assigned to the target flow of all OFSs on the path. OFC periodically requests all OFSs to report utilization of the monitored queue and store the obtained data in the internal management table. The monitoring interval should be configured based on the requirements for the real-time communications of each flow. Goal (2) of "Managing real-time communications based on bandwidth utilization" is achieved with this monitoring function.

4.4.3 Monitored Data Analysis

pmqFlow controls network equipment based on the monitored data to satisfy real-time communications. The obtained propagation time is used to detect and predict Timeout. When the propagation time exceeds a predefined timeout value, the traffic is considered to exceed the Allocated Bandwidth in the corresponding output queue. Thus, additional bandwidth should be allocated to all corresponding output queues on the path of the flow. The Timeout value and the amount of bandwidth to be added should be defined based on the requirements for the real-time communications of each flow.

The bandwidth utilization that is obtained should be used to detect and predict situations in which traffic exceeds the Allocated Bandwidth (Bandwidth Exceedance). When Bandwidth Exceedance is detected or predicted, additional bandwidth should be allocated to the output queue on corresponding OFSs. In addition, when unused bandwidth for a particular period is detected, the bandwidth should be released. The particular period to release the unused bandwidth and amount of released bandwidth should be defined based on the requirements for the real-time communications of each flow.

When the change in propagation time or bandwidth utilization has a particular inclination, Timeouts can be prevented by prediction. Thus, pmqFlow is designed to predict Timeouts and Bandwidth Exceedance by statistically analyzing the obtained propagation time and bandwidth utilization. pmqFlow leverages the least square algorithm, which is commonly used to obtain inclination, to obtain the inclination of the change in obtained data. pmqFlow represents the inclination in a linear function to minimize calculations. The means to predict Timeouts is described below.

The value of Tee at time t (Tee(t)) is represented as Eq. (4.6) with the least square linear function

$$Tee(t) = at + b, \tag{4.6}$$

where “a” means the slope and “b” means the intercept. The predicted remaining time to Timeout occasion ($T_{timeout}$) is obtained with Eq. (4.7)

$$T_{timeout} = \frac{P_{max} - P_{now}}{a}, \tag{4.7}$$

where P_{now} means the current propagation time, P_{max} means the maximum limit of propagation time (Timeout value), and “a” means the slope derived with Eq. (4.6). When $T_{timeout}$ is smaller than a predefined value (minimum time to execute prevention procedure), additional bandwidth is allocated as well as Timeout detection.

Bandwidth utilization at time *t* is also obtained with a linear function like that in Eq. (4.7) to predict Bandwidth Exceedance. In this case, $T_{timeout}$ should be replaced with the time when Bandwidth Exceedance is predicted to occur. P_{max} should be replaced with the current bandwidth allocation. P_{now} should be replaced with the current bandwidth utilization.

Six values should be configured for each flow based on the requirements for real-time communications: i) a monitoring cycle, ii) a Timeout value, iii) the time to release unused bandwidth, iv) the unit amount of unused bandwidth to be released, v) the unit amount of bandwidth to be added when Timeout is detected or predicted, and vi) the minimum time to execute prevention procedure, as was previously mentioned. These values in pmqFlow are considered to be the parameters to tune the response time or efficiency of bandwidth utilization.

4.4.4 FUNCTIONAL MODEL

Figure 4.3 outlines the functional relationships of pmqFlow, where OFS1 and OFSn correspond to OFS1 and OFSn in Figure 4.1. The unique functions in pmqFlow, in contrast to the common environment integrated with the combination of OpenFlow and OVSDB, are indicated by the bold ellipses. The solid arrows represent informational interactions. The double dashed arrows represent the flow of measurement packets. The monitoring of propagation time is achieved by RCM and the packet handlers. The monitoring of bandwidth utilization is achieved by RCM and the OpenFlow stack. There is a detailed description of each function in what follows.

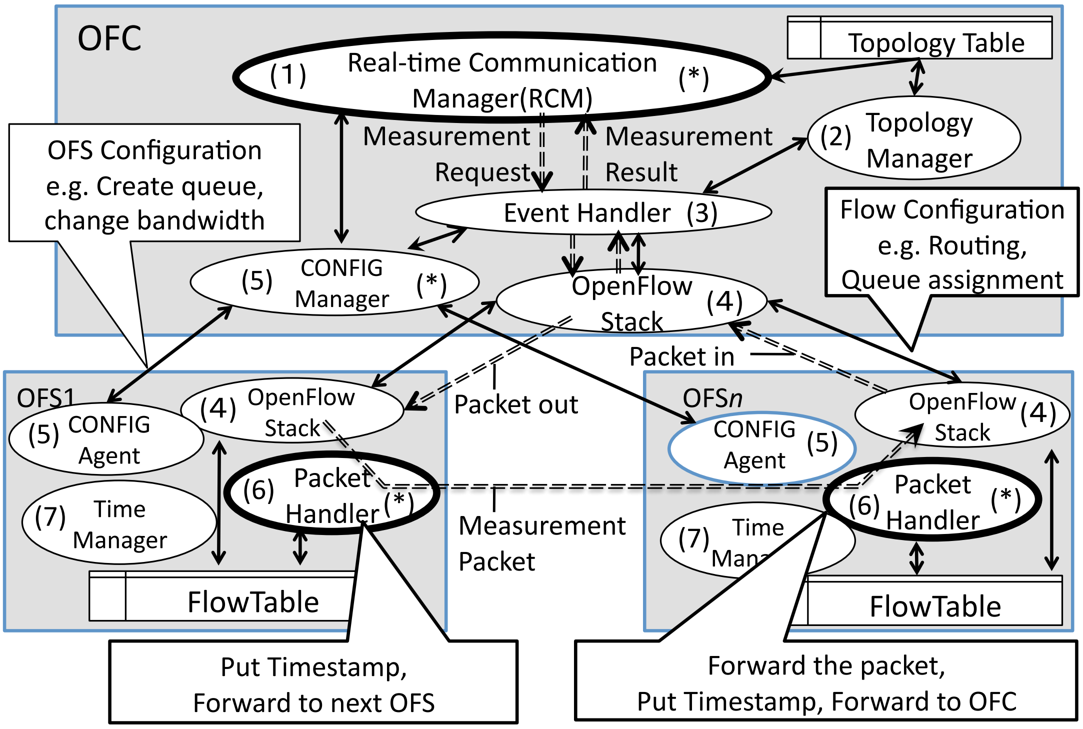


FIGURE 4.3 FUNCTIONAL RELATIONSHIP DIAGRAM

(1) Real-time Communication Manager (RCM)

RCM manages real-time communications and provides two unique monitoring functions such as propagation time monitoring and bandwidth utilization monitoring. In addition, RCM provides an analysis function for the obtained data to reconfigure the allocated bandwidth. pmqFlow achieves the goals of (1) “Managing real-time communications based on packet propagation time" and (2) "Managing real-time communications based on bandwidth utilization" with RCM. Since pmqFlow needs to monitor and respond promptly, RCM has been designed as an extension of OFC.

Propagation time monitoring is achieved with periodical measurement packets. RCM utilizes a "Packet_out" message to request OFS1 to transmit a measurement packet. When the measurement packet returns to RCM from OFSn, the "Packet_in" message is utilized. Both "Packet_out" and "Packet_in" are parts of OpenFlow compliant messages. OFC obtains the propagation time from the difference between two timestamps placed in OFS1 and OFSn. This monitoring function is a unique characteristic of pmqFlow.

RCM requests OFSs to provide statistics data to monitor bandwidth utilization. This monitoring function utilizes the "Flow_Stats" message, which is also part of the OpenFlow compliant message. RCM analyzes the obtained data, as described in Subsection 4.4.3. This monitoring function is a unique characteristic of pmqFlow and has not been introduced in research such as that by Sonkoly et al. [24].

(2) Topology Manager

Path information is required to monitor the propagation time and bandwidth utilization. Path information is also utilized for bandwidth allocation in the output queues on OFSs. This manager probes and manages the connecting information of network interfaces of OFSs and devices in the Topology Table.

(3) Event Handler

RCM was designed as a dedicated function for purposes of real-time communications management to achieve the goal of (3)"Recovering within one second". Thus, the events not related to this purpose should be handled by other functions. The Event Hander dispatches the events to appropriate functions such as RCM and the Topology Manager. In addition, this hander dispatches the events from other functions in OFC to appropriate OFSs.

(4) OpenFlow Stack

OpenFlow is a protocol to exchange information, such as routing configurations and event information, between OFC and OFS. Requests to send measurement packets, notations of packet arrivals, and bandwidth monitoring are achieved with OpenFlow messages. Thus, the OpenFlow stack is an essential entity for the goals of (1) “Managing real-time communications based on packet propagation time” and (2) “Managing real-time communications based on bandwidth utilization”.

(5) Config Manager/Agent

pmqFlow reallocates the bandwidth to the output queue in OFSs as required to achieve the goals of (1) “Managing real-time communications based on packet propagation time” and (2) “Managing real-time communications based on bandwidth utilization”. These Manager/Agent functions allow bandwidth reconfigurations. These reconfigurations are not enabled by OpenFlow.

(6) Packet Handler

pmqFlow requires OFS1 and OFSn to place timestamps on measurement packets to achieve the goal of (1) “Managing real-time communications based on packet propagation time”. Thus, pmqFlow is designed to leverage the Packet Handler in OFS to place timestamps in measurement packets, since it originally had functions to rewrite packets.

Since the Packet Hander in OpenFlow can rewrite IP addresses (including IPv4 and IPv6) and IPv6 has enough length to contain timestamps, the author designed IPv6 packets to be utilized as measurement packets and placed the timestamps in IPv6 address fields. OFS1 and OFSn specifically placed the timestamps in the source address field for the former and the destination address field for the latter. pmqFlow minimized the additional function by leveraging existing functions. This design worked for the goal of (3) “Recovering within one second”.

The packet rewriting function compliant with OpenFlow in OFS was limited to decreasing the Hop Limit or replacing the values in particular fields to static values. Thus, placing dynamic values (i.e., timestamps) involved unique processing by pmqFlow. The propagation time obtained by using this design contained the processing time to obtain timestamps and place the timestamps in packets. Thus, the predefined Timeout value needed to take into account such processing times. The processing values should specifically be added to predefined Timeout values.

The measurement packets need to trace all the output queues in the path that the target flow goes through. The timestamps should be taken before the measurement packets are forwarded at

OFS1 and after the packets are forwarded at OFSn. Thus, OFSn needs to obtain the timestamps after transmission has been completed from the specified output queue. pmqFlow leverages a function of IEEE1588 that allows obtaining the timestamps written in the network interface The first timestamp when a request to send a measurement packet (i.e., Packet_out) arrives at OFS1 needs to be detected to obtain an accurate forwarding time. Thus, the timestamp is obtained immediately in OFS1 right after the request is detected. Obtaining the timestamp only for the measurement packets avoids impact on actual packets in real-time communications since pmqFlow does not change the actual packets in real-time communications. In addition, this design helps to minimize additional load on OFS1. However, the duration from when requests are received to when timestamps are obtained is not included in the measured propagation time. Thus, the predefined Timeout value needs to take into account this duration. The duration should specifically be removed from the predefined Timeout value.

(7) Time Manager

pmqFlow obtains the propagation time from the difference between two timestamps to achieve the goal of (1) “Managing real-time communications based on packet propagation time”. The timestamps are obtained at different OFSs. Thus, the clocks on multiple OFSs involved in obtaining the propagation time should be synchronized. The time manager helps to synchronize the clocks between the devices that are involved.

4.5 PROTOTYPE

The author prototyped pmqFlow based on the previously explained design. The functions developed for pmqFlow are marked with an asterisk (*) in Figure 4.3. The purpose and detailed descriptions of the prototype are explained in what follows.

4.5.1 PURPOSE OF PROTOTYPE

The main purpose of the prototype was to confirm whether the design could achieve the two goals of (1) “Managing real-time communications based on packet propagation time” and (2) “Managing real-time communications based on bandwidth utilization” by confirming bandwidth reconfigurations based on the monitored data. In addition, another purpose of this prototype was to confirm whether the design could achieve the goal of (3) "Recovering within one second" by measuring the performance of the prototype.

4.5.2 Structure of Prototype

Figure 4.4 outlines the structure of the prototype. Each component of OFC, OFS1, OFS2, Dev1, Dev2, and Perf1 has an Intel Core i7-2700 CPU, with 8 Gbytes of DDR3 SDRAM. OFC had the Ubuntu 12.04.1 Operating System. The other devices had the Ubuntu Server 12.04.2 LTS Operating System. All of the network media were 100 Base-T to make the effect of pmqFlow noticeable.

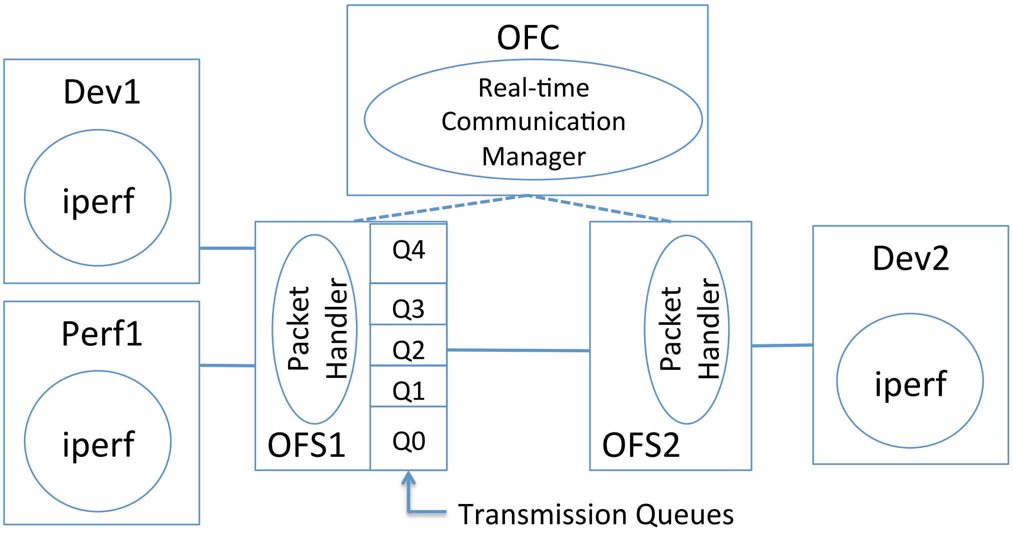


Figure 4.4 Prototype Structure and Evaluation Environment

The components in the prototype were based on those in Figure 4.3. OFC was developed on Trema-edge [45] since it supported IPv6. OFSs were developed on Openvswitch-2.1.2 [46] since it supported IPv6 and it allowed configuration via OVSDB. Openvswitch-2.1.2 was slightly modified to add functions required for pmqFlow. The details on the prototyped devices are in what follows.

(1) OpenFlow Controller

pmqFlow leverages the OpenFlow Stack, Event Handler, and Topology Manager provided by Trema-edge, since Trema-edge can provide sufficient functions. Thus, the developed functions in OFC were RCM and CONFIG Manager (CM).

RCM has a network analysis function as well as the two monitoring functions of propagation time monitoring and bandwidth utilization monitoring, as was described in Subsection 4.4. The network analysis function has five observation points: Timeout detection, Timeout prediction, Bandwidth Exceedance detection, Bandwidth Exceedance prediction, and unused bandwidth detection.

RCM requests OFS1 and OFS2 to place timestamps on the measurement packets. The author designed the use of UDP packets that had 0x5555 on both the Source Port field and the Destination Port field to distinguish measurement packets. Any fields and values would work well as long as packets could be distinguished. OFC registers appropriate rules by specifying the fields in which to place the timestamps into the FlowTables of both OFSs to allow OFS1 to place timestamps in the Source Address field and OFS2 to place them in the Destination Address field. OVSDB was chosen to configure the bandwidth size of the output queue on each OFS based on the design introduced in Subsection 4.4.1. Thus, CM was OVSDB compliant.

(2) OpenFlow Switches

Openvswitch2.1.2 provides the OpenFlow Stack and Packet Handler. In addition, it provides an OVSDB available agent as a CONFIG Agent. Since the OpenFlow Stack and CONFIG Agent have sufficient functions for pmqFlow, pqmFlow utilizes the provided OpenFlow Stack and CONFIG Agent. Thus, only the Packet Handler was developed.

The developed Packet Handler identifies measurement packets when received packets have a value of 0x5555 in both the Source Port and Destination Port as registered in the FlowTable. OFS1 rewrites the Source Address and OFS2 rewrites the Destination Address. Since the main purpose of the prototype was to confirm bandwidth allocation based on the measured propagation time and bandwidth utilization, OFS1 was defined as the target OFS for monitoring and reconfiguration. Thus, the timestamp obtained in OFS2 was that obtained before forwarding instead of the recorded timestamp in the output NIC of OFS1 and OFS2.

Since the propagation time is obtained by using the difference between the two timestamps, all involved OFSs need to synchronize their clocks. The Network Time Protocol (NTP) was chosen for synchronization in this prototype since NTP can provide sufficient accuracy (a few ms) for the required response time (one second) of Mobile HMIs.

4.6 EVALUATION

The behavior and performance of the proposed method in the environment introduced in Subsection 4.5 was evaluated. The evaluation environment is the same as that in Figure 4.4. The device that was to be monitored was OFS1. OFS2 was prepared to measure the propagation time without using the IEEE1588 available NIC. iperf (ver. 2.0.5) [47] was installed on Dev1, Dev2, and Perf1 to generate disturbance traffic. OFC monitored the propagation time between OFS1 and OFS2 to detect and predict the Timeout caused by the forwarding delay on OFS1. In addition, OFC monitored the bandwidth utilization of the output queue on OFS1 to detect unused bandwidth as well as to detect and predict Bandwidth Exceedance. Five output queues of Q0 to Q4 were configured on OFS1. The assigned traffic for each output queue is summarized in Table 4.1. Q1 is assigned to the assumed overlaid OpenFlow traffic between OFSs and OFC. Q2 is assigned to the assumed traffic of about 1500 Field Devices. The traffic assigned to Q1 and Q2 is considered to be static. Thus, static traffic and an equivalent amount to the allocated bandwidth are transmitted to Q1 and Q2. Dynamic traffic is assigned to Q3 and Q4. The assumed amounts of traffic per session are listed in the Assumed Traffic column in Table 4.1. Q0 means the default output queue. Traffic other than the traffic assigned to Q1 to Q4 utilizes Q0. In addition, the traffic assigned to Q1 to Q4 utilizes Q0 when it exceeds the allocated bandwidth. Measurement packets should be as large as possible to make the impact caused by disturbance traffic noticeable. Thus, the measurement packets were a maximum of 1384 bytes in length, which could be encapsulated in Packet_out messages without being fragmented. The results obtained from evaluations are presented in what follows.

TABLE 4. 1 CONFIGURATION OF PROPAGATION TIME MONITORING TEST

Queue ID	Application	Assumed Traffic [Mbps]	Allocated Bandwidth [Mbps]
Q4	Video Camera	0.7 to 2.5	Depends
Q3	Mobile HMI	0 to 8	Depends
Q2	PID Control	9	8
Q1	OpenFlow	2	2
Q0	Others	N/A	Others

Depends: defined in each evaluation

4.6.1 Managing Real-time Communications Based on Packet Propagation Time

Whether pmqFlow could detect and predict Timeouts by monitoring the propagation time to add bandwidth as required was evaluated. In addition, the effect of added bandwidth for real-time communications was evaluated. Mobile HMI traffic assigned to Q3 was monitored in this evaluation. Traffic from attachments of Mobile HMIs and frequent desktop refreshment was emulated. The monitoring cycle was one second to align with the goal of (3) "Recovering within one second". The predefined Timeout value was 5 ms. The following indicates how the predefined Timeout value was obtained. The target range of monitoring (TT) is given in Eq. (4.8).

$$TT = T_{ee} - (TS + TR) - (T_1 + T_{n+1}) \tag{4.8}$$

The measured propagation time obtained by pmqFlow (MT) is given in a strict sense in Eq. (4.9). The time to match the packet (FT) should be subtracted from TT. In addition, the processing time to obtain and place a timestamp (PT) should be added to TT.

$$MT = Tee - (TS + TR) - (T_1 + T_{n+1}) - FT + PT \tag{4.9}$$

The predefined Timeout value of MT (MTTO) is given in Eq. (4.10)

$$MTTO = TeeTO - (TS + TR) - (T_1 + T_{n+1}) - FT + PT, \tag{4.10}$$

where TeeTO means the timeout value of Tee in Figure 4.2. PT only occurs in measurement packets and the value does not depend on application traffic. Thus, pmqFlow treats PT as a fixed value.

MTTO is specifically obtained as follows. First, TeeTo is defined as one second to align it with the target for the response time of Mobile HMIs. The total processing time on the HMI server and mobile terminal is estimated to 0.7 s based on Rhee et al. [48]. This estimated time corresponds to the TS+TR in Figure 4.2. The assumed CPU on mobile terminals is equivalent to that of Intel Core i5. When the rest of the 1 s is allocated to the round trip time (RTT), 150 ms is assigned to each one-way communication. A relatively larger value than the Industrial Backhaul (i.e., 90 ms) is allocated to the Edge Network (i.e., T_1+T_{n+1}) since it is outside the target of monitoring and controlling. The remaining 60 ms is assigned to the Industrial Backhaul (i.e., TT). When dividing 60 ms with the maximum number of OFSs (10), 6 ms is the time allowed for forwarding in each OFS. The actual PT is 0.002 ms and FT is 0.011 ms. Thus, MTTO is 5.991 ms from Eq. (4.10). 5 ms is set for MTTO to more sensitively detect Timeouts. Note that PT is obtained by using the difference between the required forwarding time for measurement packets and non-measurement packets in OFS1; both packets are sent by OFC on Packet_out messages. The forwarding time is obtained by monitoring both sides of OFS1 with Wireshark [49]. Each forwarding time is the average of 100 trials. The observed times for measurement and non-measurement packets are summarized in Table 4.2. PT is 0.002 ms, which was derived from both average times.

TABLE 4. 2 MEASUREMENT RESULT TO OBTAIN PT

	Maximum	**Average**	**Minimum**
Measurement Packet	0.551	0.512	0.472
Non-measurement Packet	0.539	0.510	0.478

Unit: ms

FT was obtained from the difference in the forwarding times under two conditions. OFS1 has one FlowTable under the first condition. OFS1 has two FlowTables under the second condition, and the actual forwarding rule is registered in the second FlowTable. The forwarding time was obtained in the same way as PT. The obtained forwarding times are listed in Table 4.3. FT is 0.011 ms, which was derived from both average times.

TABLE 4. 3 MEASUREMENT RESULT FOR TO OBTAIN FT

	Maximum	**Average**	**Minimum**
One FlowTable	0.128	0.064	0.006
Two FlowTables	0.163	0.075	0.005

Unit: ms

RCM adds bandwidth to eliminate the Timeout occasion when Timeout occurs. In that case, RCM gradually increases bandwidth 1 Mbps at a time so that large amounts of unused bandwidth are not created. The unused period to release the bandwidth is defined as 10 s. When unused bandwidth is detected, RCM gradually reduces allocated bandwidth 1 Mbps at a time.

Since Q4 was not monitored, a static bandwidth of 2.5 Mbps was allocated and traffic of 2.5 Mbps was transmitted to it. To make Timeout noticeable, traffic of 100 Mbps was transmitted to Q0.

Traffic of 6.5 Mbps was transmitted as the base traffic of Mobile HMIs in this evaluation. In addition, traffic of 8 Mbps was periodically transmitted for 500 ms each second (Spike Traffic) to confirm bandwidth allocation. Spike Traffic can cause Timeout since it requires 8 Mbps at a time. However, this may be identified as 4 Mbps on average. Thus, it is considered that periodical monitoring of bandwidth utilization cannot detect this Bandwidth Exceedance, while the monitoring propagation time can detect such Spike Traffic. Thus, Spike Traffic detection was able to be evaluated.

Figure 4.5 plots the results obtained from evaluation. The bold solid line represents the Allocated Bandwidth. The thin solid line represents the Used Bandwidth. The Used Bandwidth is obtained by using the proposed monitoring function of Bandwidth Utilization. These lines correspond to the left scale. The dashed line represents the Propagation Time, which correspondents to the right scale. The open diamonds in the graph mean the detection of Timeout, the open triangles mean the detection of Bandwidth Exceedance, the open squares mean the prediction of Bandwidth Exceedance, and the open circles mean the detection of unused bandwidth.

Spike Traffic was transmitted after 100 s in this evaluation, which transiently caused a long propagation time of 567.523 ms. At this time, RCM detected Timeout to add bandwidth. RCM gradually added bandwidth. When the allocated bandwidth reached 15 Mbps, Timeout was eliminated. The amount of 15 Mbps was almost equivalent to the sum of base traffic (6.5 Mbps) and Spike Traffic (8 Mbps). The results from this evaluation indicated that pmqFlow could detect Timeout and allocate bandwidth to eliminate it. However, RCM could not predict Timeout.

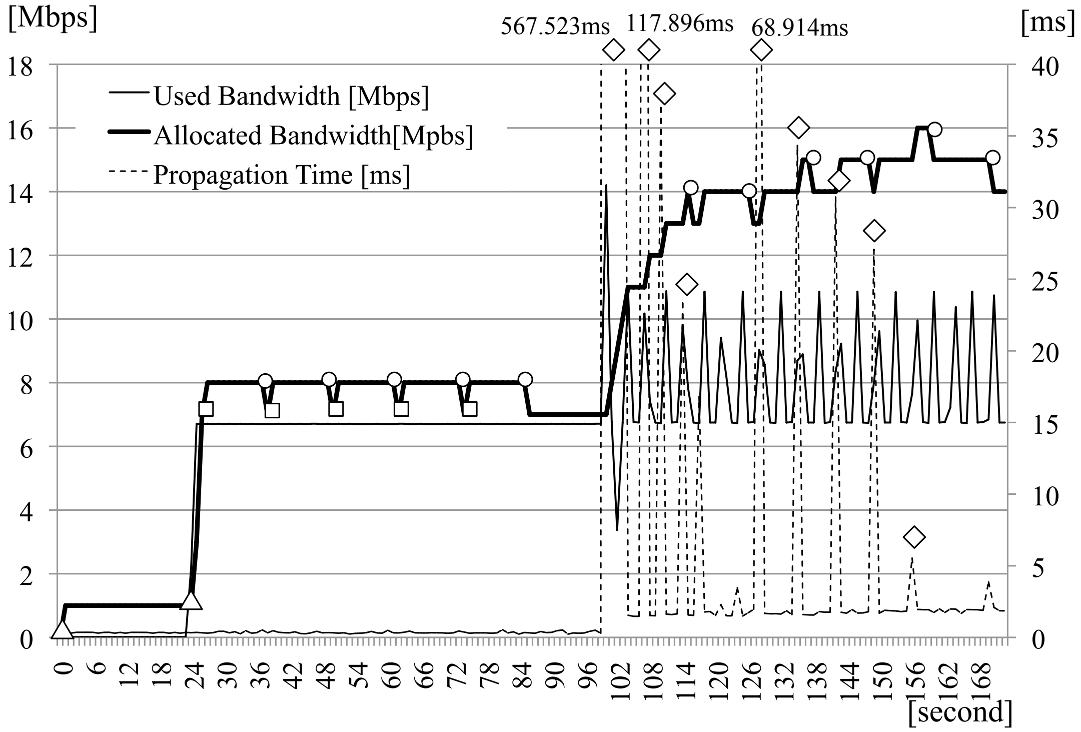


FIGURE 4.5 EVALUATION RESULT OF BANDWIDTH CONTROL BASED ON PROPAGATION TIME

4.6.2 MANAGING REAL-TIME COMMUNICATIONS BASED ON BANDWIDTH UTILIZATION

Whether pmqFlow could allocate and release bandwidth on monitored output queues based on actual Bandwidth Utilization was evaluated to achieve efficient Bandwidth Utilization. This evaluation monitored Video Camera traffic assigned to Q4. The traffic caused by added sessions, terminated sessions, and resolution changes were emulated. The monitoring cycle was one second to align with goal (3) of “Recovering within one second”. The Timeout was 5 ms. When Bandwidth Exceedance was detected or predicted, RCM allocated bandwidth. Bandwidth was released when unused bandwidth was detected. The predefined Timeout value and unit of released bandwidths were the same as those in the evaluation in Subsection 4.6.1.

Since Q3 was not monitored, a static bandwidth of 8 Mbps was allocated and traffic of 8 Mbps was transmitted to it. Traffic of 50 Mbps was transmitted to Q0 to avoid Timeout occasions in this evaluation.

Figure 4.6 presents the results obtained from evaluation. The lines and symbols have the same meanings as those in Figure 4.5. Figure 4.6 indicates that RCM detected Bandwidth

Exceedance to reallocate additional bandwidth around times of 24–40 s and 70–90 s; Bandwidth Utilization was suddenly increased during both periods. We can see that additional Bandwidth Utilization was predicted and additional bandwidth was reallocated around 90–95 s. The unused bandwidth was released after 105 s. These results indicate that pmqFlow could detect and predict Bandwidth Exceedance to allocate bandwidth. In addition, they also indicate RCM could detect and release unused bandwidth.

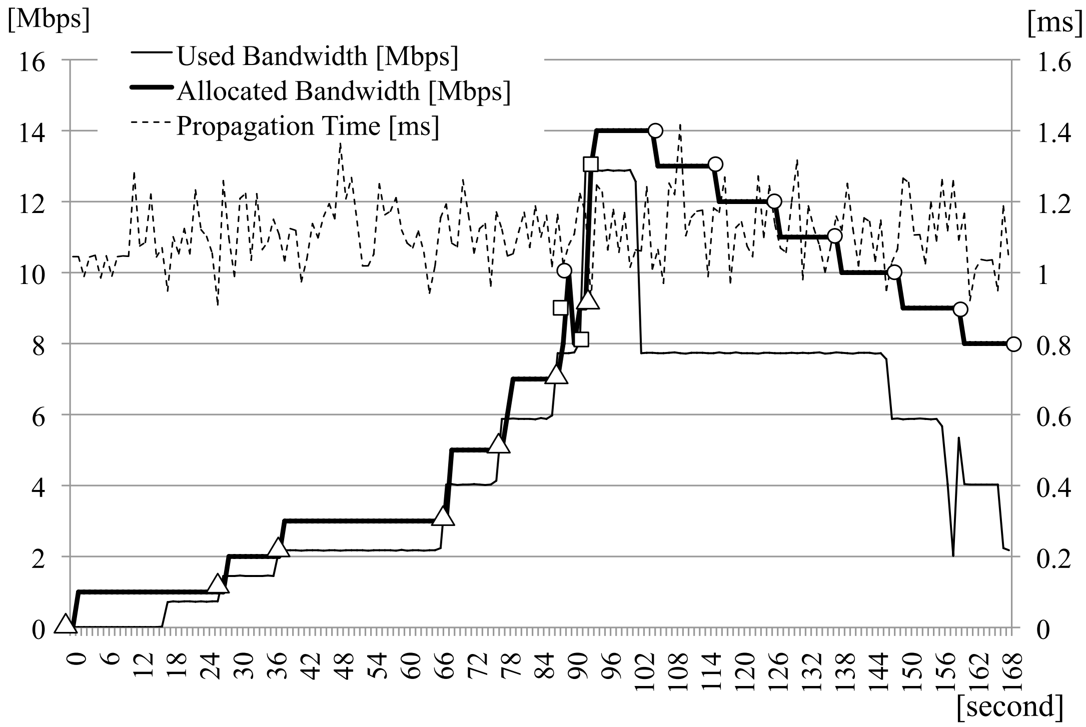


FIGURE 4.6 EVALUATION RESULT OF BANDWIDTH CONTROL BASED ON BANDWIDTH UTILIZATION

4.6.3 RECOVERING WITHIN ONE SECOND

RCM periodically performs a series of eight processes: A) transmitting measurement packets, B) receiving transmitted measurement packets, C) detecting and predicting Timeout, D) requesting statistics of output queues on OFS1, E) receiving statistics, F) detecting and predicting Bandwidth Exceedance, G) requesting bandwidth allocation of output queues (if required), and H) receiving the response for bandwidth allocation (if required). Thus, four evaluations (I–IV) are carried out to obtain the performance of pmqFlow. Each value represented below is the average of 100 measurements.

I. Time to monitor propagation time

Processes A) to C) are classified into propagation time monitoring. They are performed once in a monitoring cycle. The Timeout value is specified in Subsection 4.6.1. This processing time is obtained from the difference between two internal timestamps at RCM. The results are outlined in Figure 4.7. The required time was 1.794 ms on average.

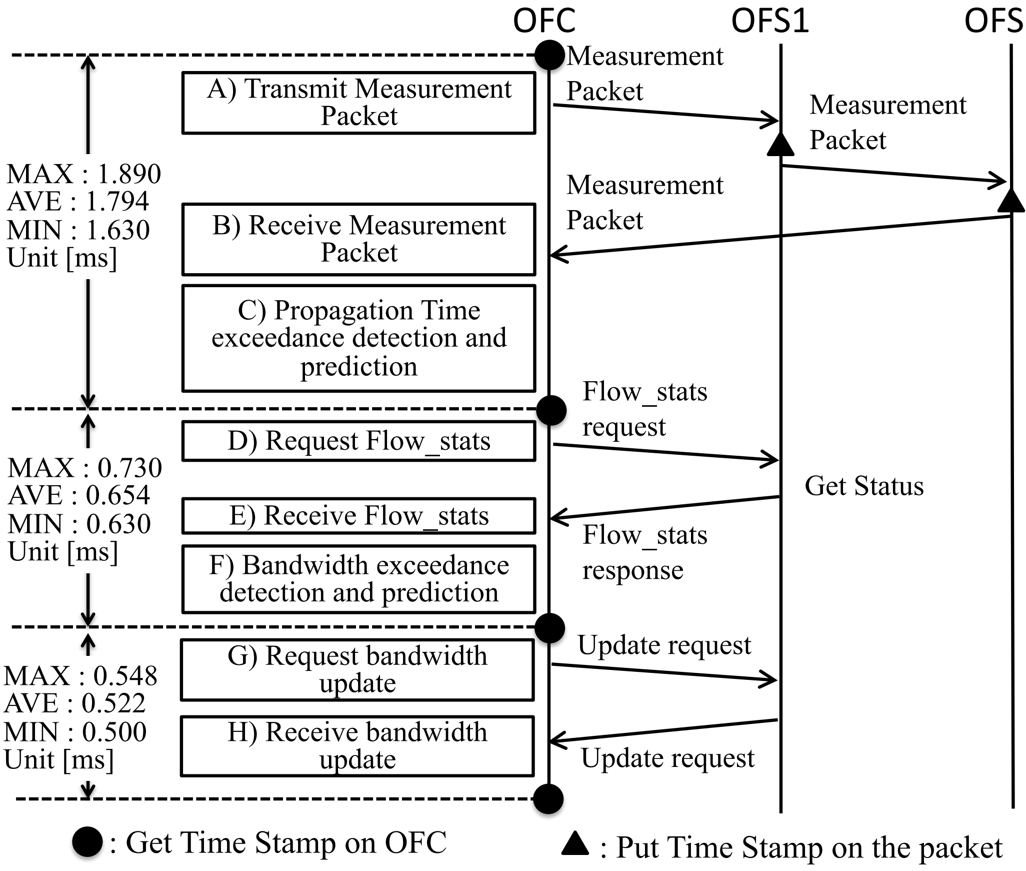


FIGURE 4.7 EVALUATION RESULT OF PROCESSING TIME OF RCM

II. Time to monitor bandwidth utilization

Processes D) to F) are classified into Bandwidth Utilization monitoring. Monitoring should be performed on all OFSs in the path of the target flow. This processing time is obtained from the difference between two internal timestamps at RCM. The required time was 0.654 ms on average.

III. Time to allocate bandwidth

Processes G) to H) are classified into bandwidth allocation. Allocation should be performed on all OFSs in the path of the target flow. This processing time is obtained from the difference between two internal timestamps at RCM. The required time was 0.522 ms on average. This was sufficiently small to achieve the goal of one second.

IV. Time to forward packets

There may be some additional OFSs between OFS1 and OFS2, as shown in Figure 4.1. The simple forwarding time on OFS was measured to obtain the additional time caused by packet forwarding on OFSn. The forwarding time was obtained from the difference between the timestamp on the incoming network port and the timestamp on the outgoing network port on OFS1. Wireshark obtained the timestamps. The obtained time was 0.324 ms on average.

The required time for the maximum path in the Industrial Backhaul, i.e., ten hops, was calculated with the required time obtained in the four evaluations from (I) to (IV). The value of (I) was obtained in directly connected OFS1 and OFS2. The forwarding time of eight OFSs (2.592 ms) should be added to obtain the maximum required time for A) – C). Thus, the maximum required time for (I) was 4.386 ms (1.794+2.592 ms). The required time for D) – F) on ten OFSs was 6.54 ms. Thus, the total monitoring time, A) – F), was 10.926 ms (4.386+6.54 ms). The time to allocate bandwidth on ten OFSs required 5.22 ms (corresponding to E) – F)). Thus, the total processing time from A) to F) should be 16.146 ms. This means it is possible to eliminate Timeout within one second if the monitoring cycle is configured to less than 983.854 ms.

4.7 CONSIDERATIONS

Figure 4.6 indicates that bandwidth allocation based on actual bandwidth utilization is achieved by detecting and predicting Bandwidth Exceedance. In addition, Figure 4.6 indicates that releasing unused bandwidth was achieved.

Figure 4.5 indicates that Bandwidth Utilization after 156 s (Timeouts were eliminated) was less than 11 Mbps. There is a large difference between the Allocated Bandwidth and the

Utilized Bandwidth. It shows that periodical monitoring of Bandwidth Utilization cannot detect the peak amount of utilization, since such monitoring obtains flattened average values. RCM, on the other hand, allocated the same amount of bandwidth as the actual peak utilization. This is because the propagation time monitoring function worked successfully. This means the monitoring propagation time complemented the Spike Traffic detection to eliminate Timeouts. The Spike traffic etection was impossible for the Bandwidth Utilization monitoring function.

Figure 4.5 indicates that Timeouts can occur since it is impossible to predict sudden increases in traffic. This is unavoidable when dynamically allocating bandwidth. Shorter monitoring cycles would be effective to reduce the impact of such sudden increases. Figure 4.5 also indicates that it took long periods to eliminate Timeouts. This is because RCM gradually increased bandwidth, 1 Mbps at a time. Adding larger bandwidths at a time would be effective to shorten the elimination period for Timeouts. Figure 4.6 indicates that RCM gradually reduced the unused bandwidth, 1 Mbps at a time, even when larger amounts of Bandwidth Utilization decreased. This was intended to reduce the impact on propagation time when traffic increased again. However, reducing the Allocated Bandwidth to the same amount as that in actual utilization or reducing larger bandwidths at a time would be effective for more efficient Bandwidth Utilization. Six values should be configured for each flow, as was explained in Subsection 4.4.3, based on the requirements for real-time communications: i) the monitoring cycle, ii) the timeout value, iii) the time to release unused bandwidth, iv) the unit amount of unused bandwidth to be released, v) the unit amount of bandwidth to be added when Timeout is detected or predicted, and vi) the minimum time to execute prevention procedures. These values would be helpful to quickly eliminate Timeouts or achieve efficient Bandwidth Utilization.

Figure 4.5 and Figure 4.6 indicate that RCM can predict Bandwidth Exceedance with a simple least squares algorithm, and it can predict Bandwidth Exceedance with the data increase slope shown in Eq. (4.7). This means Bandwidth Exceedance with a simple data increase can be predicted with the algorithm. For example, such simple data increase will occur when accidents at some facilities occur (e.g., cracks in tanks). In that case, multiple operators will access corresponding Video Cameras. This results in traffic increases. When considering normal operations, other algorithms such as machine learning algorithms may work well. For example, Field operators walk around the Field on schedule to observe the Field situation with Mobile HMIs. RCM can identify the amount of traffic and create schedules to predict Bandwidth Utilization with some kinds of machine learning algorithms.

Since the proposed method chooses flows as the monitoring units, the scalability of this method can be discussed with the number of flows. Based on the results from evaluation, the required time for periodical monitoring was 2.448 ms. Thus, RCM could monitor 404 flows in the previously mentioned 989.743 ms monitoring cycle.

Most of the inter Area communications in plants are between the central control room and each Area. Thus, communications are structured in a star topology of 1: 9, where one means the central control room and nine means the Areas. RCM can manage 44 real-time communications per Area to evenly divide the 404 flows into the nine Areas. pmqFlow specifies the monitored flow with a combination of {source_Area, destination_Area, application} to make the target flow independent of the number of devices or amount of traffic, as was described in Subsection 4.4.1. Thus, the number of monitored flows is almost equivalent to the number of applications communicating between the Area of the central control room and other Areas. Six flows should be monitored to protect the real-time communications of PID control, Mobile HMIs, and Video Camera traffic in both directions. While PID control utilizes Pub/Sub FF-HSE communications, FF-HSE has two more types of communications, such as Client/Server and Report Distribution. Even when RCM supports all FF-HSE communications and the overlaid secure channel, a 44 flow capacity per Area is sufficient for the world's biggest class plant (with 10 Areas, 16,000 Field Devices, 5 Mobile HMIs, 5 Video Camera). This means that the proposed method has enough scalability for most existing plants. Note that this number means the capacity when RCM sequentially processes monitoring since it is obtained by simply dividing the monitoring cycle by the required monitoring time. If more applications require real-time communications and the flows to be monitored is increased, the capacity can be extended by parallel monitoring.

The results from evaluations demonstrated that the proposed method could dynamically allocate and release bandwidth based on actual use. Since ordinary OpenFlow based networks provide functions that dynamically manage attached devices and control paths, pmqFlow can provide real-time communications even to portable devices like Mobile HMIs.

4.8 Conclusion

This chapter presented the design, prototype, and evaluation of pmqFlow, which provides essential functions to manage real-time communications for PA in the Industrial Backhaul. pmqFlow provides two novel functions such as "Managing real-time communications based on packet propagation time" and "Managing real-time communications based on bandwidth utilization”. The results from the evaluation confirmed that these two functions were effective to achieve the goal of (1) “Managing real-time communications based on packet propagation time” and that of (2) “Managing real-time communications based on bandwidth utilization”. In addition, both functions mutually complemented the shortcomings. Moreover, it was confirmed that the goal of (3) “Recovering within one second” was achieved since the total time that was required to monitor and reconfigure OFSs (16.146 ms per flow) was significantly small for the target time of one second.

A simple algorithm was utilized to predict Bandwidth Utilization in RCM in the prototyping and evaluation of pmqFlow. In addition, pmqFlow provided a capability to configure multiple parameters regarding monitoring and reconfiguration to adjust to the requirements of each application. Since the requirement for real-time communications depends on the application, appropriate algorithms and parameters should be selected for each application. For example, Figure 4.5 indicates that eliminating Timeout takes a certain period of time, which could be shortened by using a larger bandwidth increment. However, a larger bandwidth increment would decrease the efficiency of Bandwidth Utilization. In addition, Figure 4.6 indicates that the pace to release bandwidth is slow, and reducing greater amounts of bandwidth would improve the efficiency of Bandwidth Utilization. Greater amounts of reduced bandwidth, on the other hand, would also increase the possibility of Timeouts. The parameters can have a trade-off, as was shown here. Thus, the most suitable parameters and algorithms should be prepared to improve the dependability of real-time communications and the efficiency of Bandwidth Utilization.

RCM reallocates the bandwidth as required in pmqFlow. It needs to authorize and confirm the upper limit of bandwidth reallocation for each flow before reallocation is performed, since bandwidth is a finite resource. One of the means of authorization is communicating with the Authentication, Authorization and Accounting (AAA) server such as the RADIUS server or Diameter server. pmqFlow should be integrated with a kind of AAA server or some other means in the future. Another concern caused by the finite resource of bandwidth is the shortage of unallocated bandwidth. In this case, no more flows can be allocated to the required bandwidth. OFC needs to calculate another path when shortages occur, and allocate the bandwidth on a path and confirm if the path and allocated bandwidth satisfy the real-time communications requirements. If the path is not appropriate, OFC needs to consider moving the existing flow to another path while satisfying its real-time communications requirements. The algorithms to arrange bandwidths and paths should be worked on in the future.

Chapter 5

sQoS: The Design and Prototyping of Secure QoS for Process Automation

5.1 Introduction

The entire PA related manufacturing system consists of the functions described in Chapter 1. The levels of functional hierarchy are summarized in Table 5.1. The manufacturing system is classified into Enterprise Information Technology, which supports Enterprise Resource Planning (ERP), and the Industrial Control System (ICS).

Table 5.1 Levels in the Manufacturing System

Classification	Level	Role
Enterprise Information Technology	4	Establishing the basic plant schedule – production, material use, delivery and shipping. (e.g., ERP)
Industrial Control System	3	Work Flow/recipe control to produce the desired end products. Maintaining records and optimizing the production process. (e.g., MOM)
	2	Monitoring, supervisory control and automated control of the production process. (e.g., DCS)
	1	Sensing and manipulating the production process. (e.g., Fieldbus)
	0	The actual production process.

ICS includes MOM, DCS, and Fieldbus. ICS was conventionally separated based on regions and/or functions. Users are currently considering improving plant operations by introducing WSNs (e.g., ISA100.11a, WirelessHART) and the Industrial Backhaul, which interconnects the control room and WSNs based on the rapid evolution of Information Communication Technology (ICT). However, users are concerned with security risks since some cyber attacks targeting PA manufacturing systems, such as Stuxnet and Night Dragon [50], have been reported.

Packets in PA networks, such as DCS and Fieldbus, need to arrive at the destination on schedule. Packets that cannot arrive on schedule are not used for PID control, which is the most commonly used control method in PA. This means the delayed packets are equivalent to unreached packets from the PA perspective. Therefore, delay and jitter need to be minimized in the Industrial Backhaul and high levels of availability need to be provided.

The Industrial Backhaul is a network interconnecting geographically separated Areas that is shared by various applications. Methods of Quality of Service (QoS) are generally used to minimize latency and jitter. The QoS methods can prioritize or allocate exclusive bandwidth to particular packets such as PID control packets. However, attackers can spoof high priority packets since the existing QoS methods do not have any mechanisms to protect packet attributes to classify packets in intermediate equipment. If attackers spoof high priority packets, this causes large amounts of jitter or packet loss in actual PID control packets. As the result of these kinds of attacks, manufacturing processes will be suspended or physical explosions may occur. The author called such attacks QoS Spoofing Attacks in this study. The author investigated the latest trends by PA users and identified the possibility of QoS Spoofing Attacks on the Industrial Backhaul. Thus, the author designed and prototyped a method that prevented QoS Spoofing Attacks, which was called Secure QoS (sQoS).

The rest of this chapter is structured as follows. Subsection 5.2 introduces current technical trends in PA. Subsection 5.3 describes investigations into new threats of QoS Spoofing Attacks in PA. Subsequent subsections provide detailed descriptions in order of the goals (5.4), design (5.5), prototype (5.6), and evaluation (5.7). Finally, Subsection 5.8 summarizes the chapter and discusses future work.

5.2 TREND IN PROCESS AUTOMATION

A) Wireless Sensor Networks

PA users are expecting to utilize WSNs, such as ISA100.11a and WirelessHART. WSNs enable rapid installation at less cost compared to wired systems. In addition, WSNs allow dynamic installation of Field Devices such as sensors and actuators.

WSNs also have strong security features such as channel hopping and encryption. Thus, WSNs can be considered secure enough against attacks.

B) Industrial Backhaul

Industrial Backhaul is a network that interconnects distributed elements. Typical use of the Industrial Backhaul involves conveying data from a WSN to the control room. However, its use is not limited to WSN related control applications. PA users expect to utilize the Industrial Backhaul for other applications such as monitoring Video Cameras, VoIP, mobile worker terminals (Mobile HMIs) as well as PID control applications. The Industrial Backhaul could consist of wired networks, wireless networks, or mixed networks. The ISA-TR100.15.01-2012 Backhaul Architecture [4] provides guidelines on how to securely utilize the Industrial Backhaul in which the Industrial Backhaul is defined as an unsecure network since multiple users in multiple Areas transmits packets into it. Thus, the guidelines suggest validating high priority packets.

C) Cyber Attacks

Some cyber attacks have recently targeted PA systems and a typical one is Stuxnet, which is highly intelligent malware that utilizes unreported security holes. Stuxnet can hide itself and it targets particular devices of Programmable Logic Controllers (PLCs). There is a great possibility that Stuxnet was designed to incorrectly control nuclear plants. After Stuxnet was reported, PA users, such as nuclear plant users and chemical plant users, found that cyber attacks could target PA and the results would be irreparable. Consequently, the importance of cyber security in PA is strongly accepted, while the difficulty of completely preventing cyber attacks is also known.

5.3 QoS Spoofing Attacks in Process Automation

The whole PA system is classified into the five functional levels listed in Table 5.1. Each level has an individual purpose and requirements for the network. Lower levels generally require stricter network availability, security, and real-time communications. In practice, levels 1 through 2 are integrated together, and this was called converged “level 1+2” in this study since these levels have a close relationship. The communications between the WSN and the control room via the Industrial Backhaul can be mapped to the communication in “level 1+2”.

A common PA is achieved with a periodical PID control loop, as has been outlined in Figure 5.1. Three communications and three processes must be completed within a PID control loop. The typical PID control loop cycle is one second, while the shortest cycle is 300 ms. The packets need to arrive on schedule to achieve the control loop. Thus, the PA system requires real-time communications.

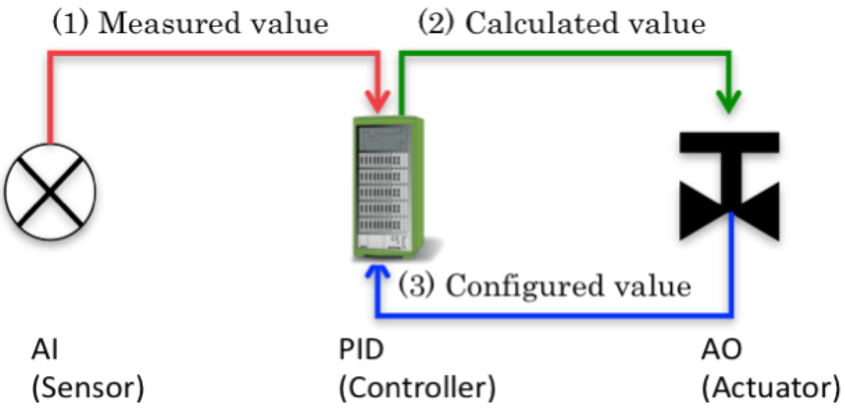


FIGURE 5.1 PID CONTROL IN PA

The PA was conventionally operated in a closed network with propriety protocols. WSNs and the Industrial Backhaul have recently encouraged PA users to operate with wider networks and open technologies. However, the impact of cyber attacks on PA systems will not be limited to the cyber side but will also affect the physical side, and damage will be critical like that caused by plant explosions. QoS Spoofing Attacks can result in such serious damage. This means that the threat of QoS Spoofing Attacks is serious and unique to PA. Thus, the risk of utilizing WSNs and the Industrial Backhaul should be carefully considered.

5.4 GOAL OF SQOS

The objective of sQoS is preventing QoS Spoofing Attacks. Five detailed goals are described in detail below.

(1) Spoofing Packet Detection

Intermediate network equipment needs to detect spoofing packets to prevent QoS Spoofing Attacks, as was previously mentioned. Authentication methods are helpful to detect spoofing packets.

(2) IP Compliancy

sQoS needs to be compliant with Internet protocols (IP) to be utilized on the Industrial Backhaul, since the Industrial Backhaul would be an IP based network to support a variety of applications as was described in [4].

(3) QoS method agnosticity

sQoS needs to be QoS method agnostic since the fields to classify the packets depend on the QoS method. sQoS needs to provide the possibility of selecting the fields to be authenticated to provide the features of being QoS method agnostic.

(4) Prevention of Replay Attacks

Attackers can transmit numerous copied proper packets to disrupt real-time communications since the authentication algorithms cannot detect packet duplication. These are called Replay Attacks. Thus, mechanisms to prevent Replay Attacks are required.

(5) Less Overhead

Since the motivation for sQoS is to protect real-time communications, the overhead time caused by sQoS must be short enough to satisfy the shortest control cycle of 300 ms under the assumed largest PA topology in the world's biggest plant (Shell Nanhai), which has ten routers and two switches, between Field Devices and the PID controller. The assumed practical processing times of the Sensor, PID controller, and Actuator correspond to 30, 45, and 90 ms. Thus, the allowed time for each communication is 45 ms from Eq. (5.1).

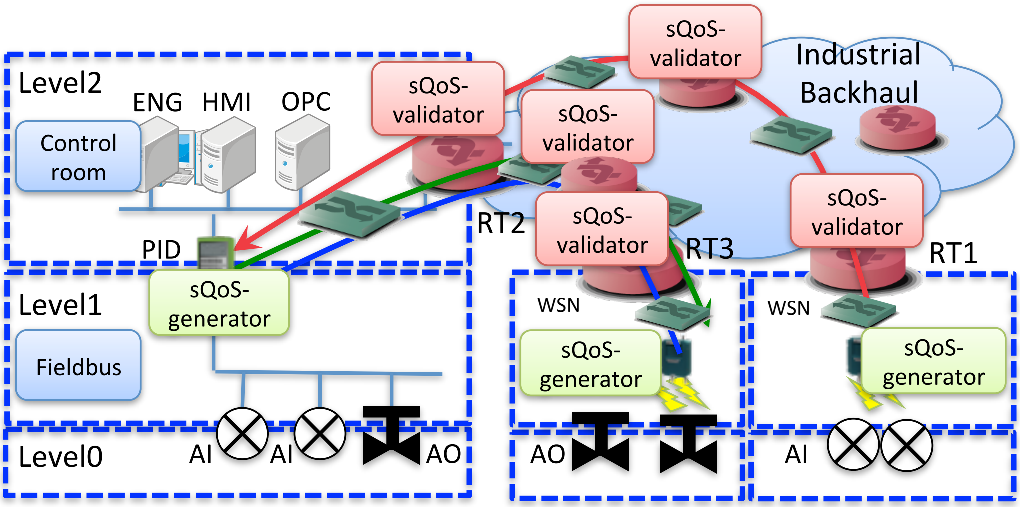
$$\frac{(300-(30+45+90))}{3} \tag{5.1}$$

5.5 Design

sQoS is a novel method that focuses on the prevention of QoS Spoofing Attacks by validating the real-time communications packets on each intermediate router. sQoS was designed to be compliant with IPv6 [51]. In addition, sQoS is QoS method agnostic by introducing a function that allows the QoS field to be selected. This chapter introduces the design and prototyping of sQoS.

5.5.1 Ground Design

Figure 5.2 outlines a functional map of sQoS in a PA system. There are two key entities in sQoS (i.e., the sQoS-generator and sQoS-validator). The sQoS-generator works on the packet originator. The sQoS-validator works on each router. The sQoS-validator can work on the packet receiver but this is not required since the packets have already arrived at the destination device and QoS operation has already finished for the given packets.



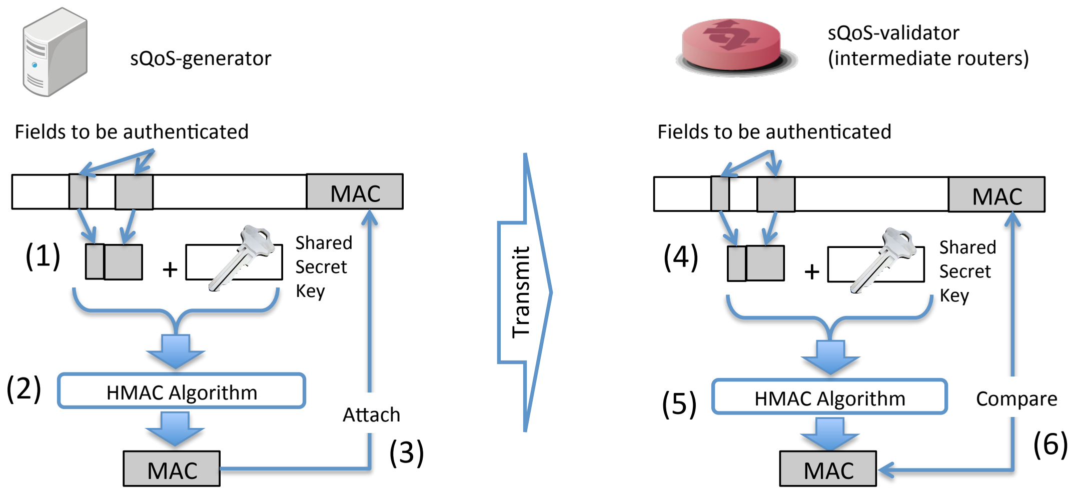
OPC: OLE for Process Control, HMI: Human Machine Interface, ENG: Engineering Terminal, PID: Proportional-Integral-Derivative Controller, AI: Analog Input, AO: Analog Output

Figure 5.2 Functional Map in PA System

5.5.2 Spoofing Packet Detection

Generating a Message Authentication Code (MAC) for a given packet with a shared secret key and placing the generated MAC into the packet should effectively detect spoofing packets. The algorithm is known as a Hash-Based Message Authentication Code (HMAC). An alternative method to authenticate packets is by using a digital signature, which uses private and public keys. Since sQoS needs to minimize the overhead of calculation, the author chose the lighter method of HMAC.

The HMAC based conceptual model is outlined in Figure 5.3. The fundamental procedure involves six steps. (1) The packet originator prepares the secret key and the value in specified fields of the given packet. (2) The packet originator generates the MAC of the prepared values with the HMAC algorithm. (3) The packet originator places the MAC into the packet and sends it. (4) When the packet arrives at an intermediate router, the router prepares the secret key and the value in the specified field of the given packet. (5) The router generates the MAC of the prepared values with the HMAC algorithm. (6) The router compares the generated MAC with the MAC in the packet. If the values are not equivalent, the packet can be considered to be spoofed. Thus, the router discards the packet. If the values are equivalent, the router forwards the packet based on a predefined QoS behavior (e.g., it prioritizes the packet or places it into the appropriate output queue with a particular bandwidth).



MAC: Message Authentication Code
HMAC: Keyed-Hashing for Message Authentication Code

Figure 5.3 Spoofing Packet Detection Method

5.5.3 IP COMPLIANCY

When sQoS utilizes the IPv6 Hop-by-Hop option header, it can be fully compliant with IPv6. It has been predicted that a variety of applications will share the Industrial Backhaul regardless of the IP versions according to [4]. This means that the Industrial Backhaul needs to be a dual stack network of IPv4 and IPv6. Therefore, an IPv6 Hop-by-Hop option header based sQoS can properly work on the Industrial Backhaul.

Figure 5.4 outlines the format of the Hop-by-Hop option extension header for sQoS (sQoS Header). The Hop-by-Hop option header is designed to be examined by intermediate routers in the IPv6 specifications. The nature of the Hop-by-Hop option extension header fits the concept of sQoS.

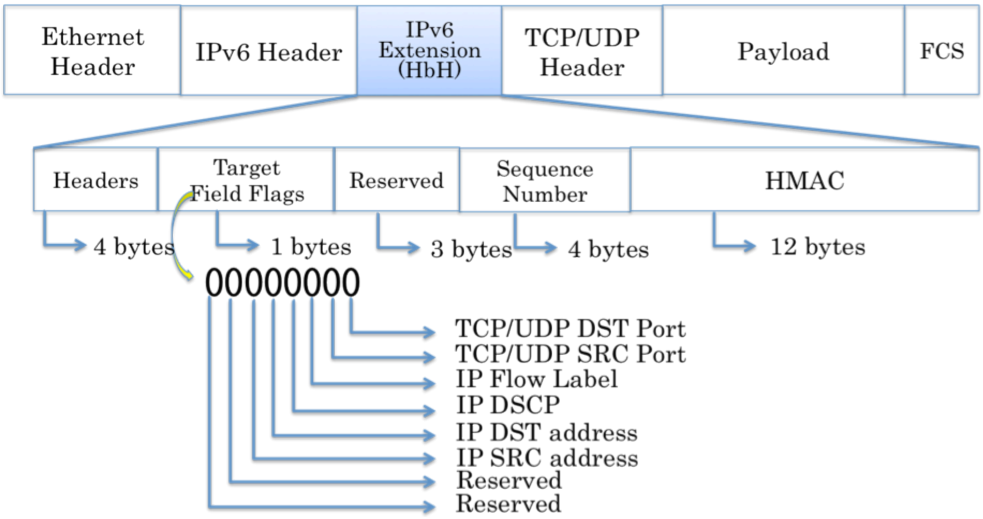


FIGURE 5.4 SQOS HEADER FORMAT

5.5.4 QOS METHOD AGNOSTICITY

The fields used to classify the packets depend on the QoS method. sQoS needs to be capable for any kinds of QoS methods. Thus, the sQoS Header has one byte flags, viz., Target Field Flags, to specify the fields to be authenticated. Two bits from the left most bits are reserved for future use. The third bit specifies the IP source and the fourth bit specifies the IP destination address field. The fifth bit specifies DSCP and the sixth bit specifies the Flow Label field. The seventh bit specifies the source and the eighth bit specifies the destination port field of the transport protocol (i.e., TCP/UDP).

5.5.5 PREVENTION OF REPLAY ATTACKS

Attackers can snoop the network and copy proper packets to resend. These are known as Replay Attacks. If there is no mechanism to prevent such Replay Attacks, it is possible to cause long delays or packet loss by sending many captured proper packets. The sQoS Header has four bytes of the sequence number field to prevent Replay Attacks. The sender places a number that linearly increases one by one in a packet. The routers record the number after they validate whether the number in the received packet is larger than the recorded one. If the number is the same or smaller than the recorded one, the routers discard the packet since the packet can be considered to be a replayed packet.

5.5.6 LESS OVERHEAD

A short processing time is required since the motivation for sQoS is making the PA packets arrive on schedule. Thus, sQoS choses the HMAC algorithm rather than a digital signature algorithm, since hash based HMAC is lighter than a digital signature algorithm, which is based on asymmetric key encryption.

5.6 PROTOTYPE

sQoS software was prototyped on Linux to evaluate its feasibility especially from the perspective of performance. This subsection introduces the prototype of sQoS. Figure 5.5 outlines the software architecture model in a router.

5.6.1 SQOS-GENERATOR

The packet originating applications in this prototyping have the function of the sQoS-generator, which generates the sQoS Header. After the sQoS Header is generated, the applications can place the header into the packet and send it with sendmsg( ) API. The sQoS-generator in Figure 5.2 is placed in the PID controller as well as the Sensor devices since the PID controller sends the packets with the sQoS Header to Sensors or Actuators.

The HMAC algorithm allows a hash algorithm to be selected. HMAC-SHA1 was chosen for this prototyping. Wang et al. [52] reported that SHA1 collision can occurred with 2^69 calculations. However, this is not a large threat from the QoS perspective since only one packet cannot create long delays in the packet. It is also possible to choose a stronger hash algorithm with a longer calculation time if required.

Key management was beyond the scope of this prototype. The author manually configured the secret key. It is possible to utilize existing key management protocols such as those used by Sandro and Hutchison [53]. Integration with such key management protocols should be done in further studies.

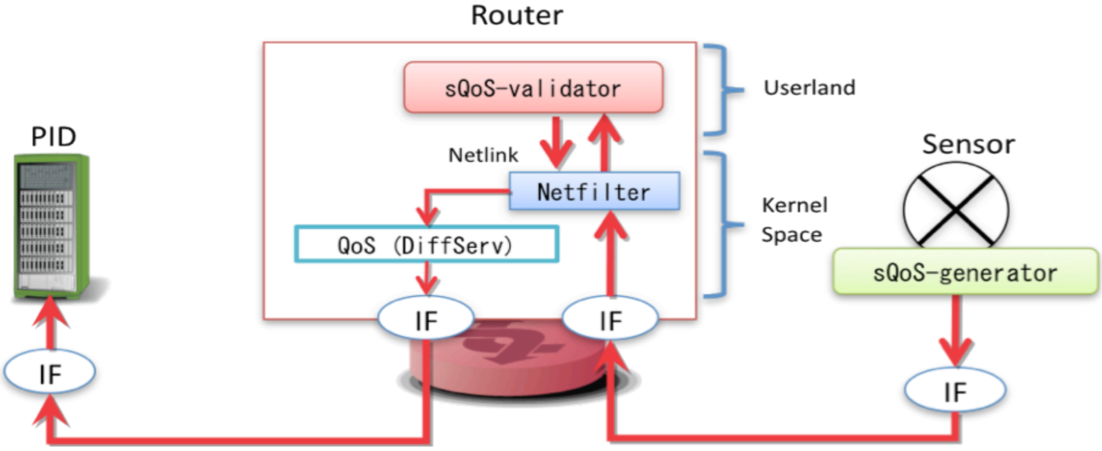
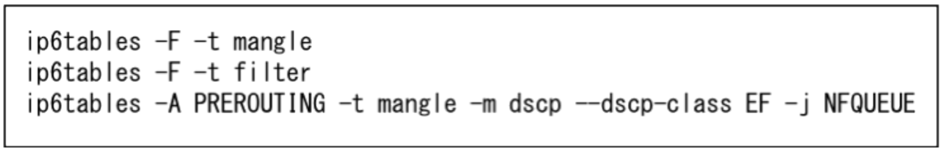


FIGURE 5.5 SQOS SOFTWARE ARCHITECTURE

5.6.2 SQOS-VALIDATOR

The sQoS-validator in this prototyping is implemented as a userland application. The router utilizes the ip6tables to select the packet to be validated. When a packet to be validated arrives, the packet is forwarded to the sQoS-validator via a Netlink socket. After validation, the sQoS-validator sends back the packet if the packet is valid. Otherwise the packet is discarded.

The ip6tables command is provided in Figure 5.6. It shows that EF class packets are selected to be validated at PREROUTING, which is the first stage of packet processing in the routers. This means the QoS process, such as the Per-Hop-Behavior in DiffServ, in the router is conducted after validation.



```
ip6tables -F -t mangle
ip6tables -F -t filter
ip6tables -A PREROUTING -t mangle -m dscp --dscp-class EF -j NFQUEUE
```

FIGURE 5.6 IP6TABLES COMMAND TO FORWARD THE PACKET TO APPLICATION

5.7 Evaluation

5.7.1 Evaluation Environment and Condition

The packets used for the evaluation were UDP over IPv6 with a size of 256 bytes. The typical control message size in PA is between 64 to 128 bytes. The 256 bytes was a little larger than the expected PID control packet size of 200 bytes, (i.e., IPv6 Header (40 bytes) + sQoS Header (24 bytes) + UDP Header (8 bytes) + control message (128 bytes)). The most time conscious communications in PA, (i.e., Publisher/Subscriber communications), utilize UDP as the transport protocol.

All implementations worked on Ubuntu-12.04LTS with openssl (libssl-dev 1.0.1-4ubuntu5.5) that ran on an Intel Core i7-2600 CPU (3.4 Ghz) with an 8 Gigabyte Memory.

A simple topology was built to measure the impact of latency on sQoS as outlined in Figure 5.7, where there are two routers between the Sensor and PID, viz., RT1 and RT2. There is a switch, such as SW1 to SW3, to capture the packets in each hop. A packet-capturing device is connected to each switch.

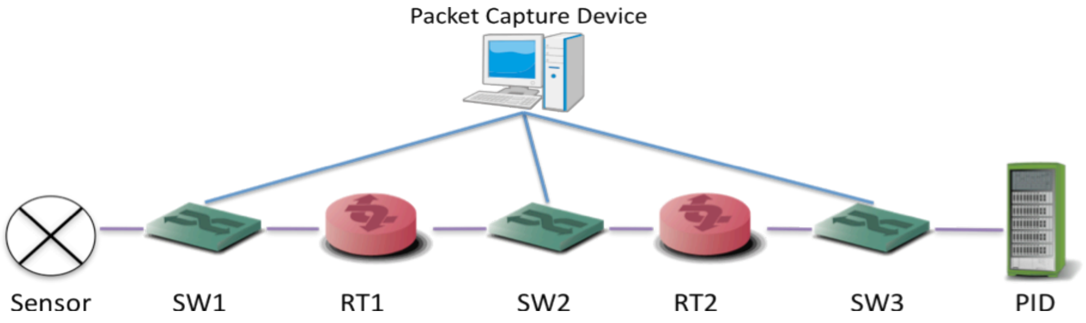


Figure 5.7 Topology to Evaluate the Impact of sQoS

5.7.2 Evaluation of sQoS-generator

The author ran two types of applications. The first placed the sQoS Header into the UDP over the IPv6 packet; the applications will be called sQoS-CL and sQoS-SV after this. The second types did not place the sQoS header; the applications will be called non-sQoS-CL and non-sQoS-SV after this. In this evaluation, no router validates the sQoS Header even though the arrived packet contains the sQoS Header.

The author measured RTT for both pairs of sQoS-CL/sQoS-SV and non-sQoS-CL/non-sQoS-SV (Figure 5.8). The difference between them is two sQoS-generators. When sQoS-CL starts to generates a packet, it records "timestamp 1" before it generates the sQoS Header and transmits. When the PID receives the packet, it replies to the packet with the sQoS Header, which is generated by sQoS-SV. Finally, the replied packet arrives at sQoS-CL, and then sQoS-CL records "timestamp 2". The difference between the two timestamps is the sum of two sQoS Header generations and round trip transmission time (the difference is called GT). The timestamps ("timestamp 3" and "timestamp 4") in the non-sQoS-CL/non-sQoS-SV case are recorded at the same timing as those in the sQoS-CL/sQoS-SV case. Thus, the difference between "timestamp 3" and "timestamp 4" is the pure RTT (the difference is called PT). The processing time per sQoS-generator (T_{gen}) is calculated with Eq. (5.2).

$$T_{gen} = \frac{GT - PT}{2} \tag{5.2}$$

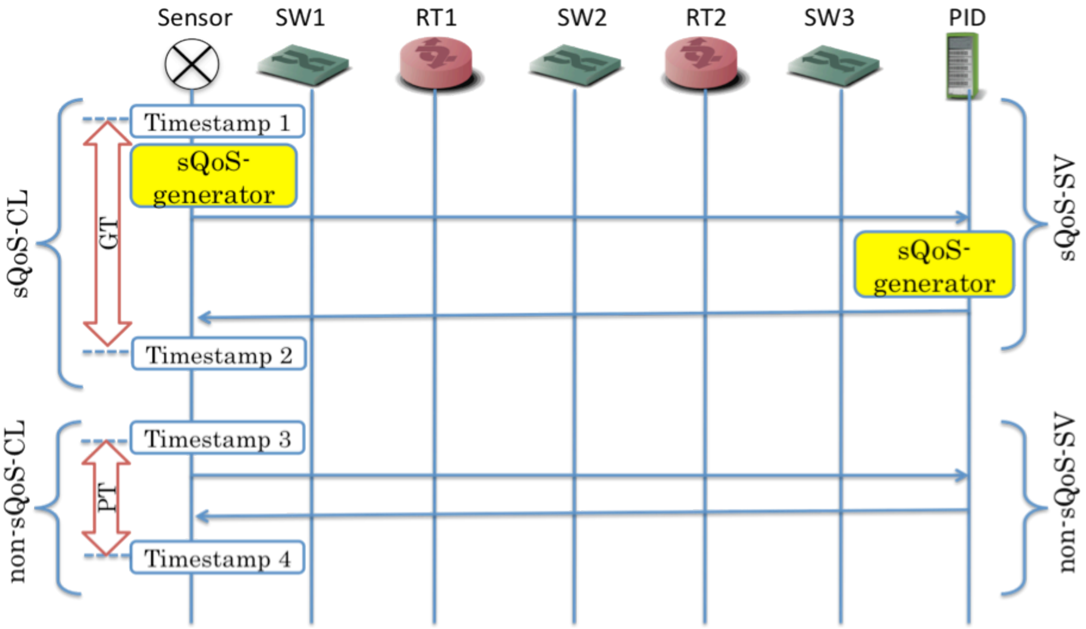


Figure 5.8 Sequence to evaluate sQoS-generator

5.7.3 Evaluation of sQoS-validator

The author built another topology outlined in Figure 5.9 for this evaluation, where the sQoS-validator worked on RT1 but not on RT2. The three switches were physically actually one switch, and the packets were captured from a mirroring port.

The timestamps of an outgoing packet from SW1 and an incoming packet to SW2 were recorded (“timestamp 1” and “timestamp 2”) to measure the entire processing time of RT1. The difference between the two timestamps was the sum of the sQoS validation time and packet forwarding time (the time difference was called VF). The timestamps of an outgoing packet from SW2 and an incoming packet to SW3 were recorded (“timestamp 3”) to measure the pure forwarding time. The difference between “timestamp 2” and “timestamp 3” was the pure forwarding time (the time difference was called PF). The processing time for sQoS validation (T_{val}) was calculated with Eq. (5.3).

$$T_{val} = VF - PF \tag{5.3}$$

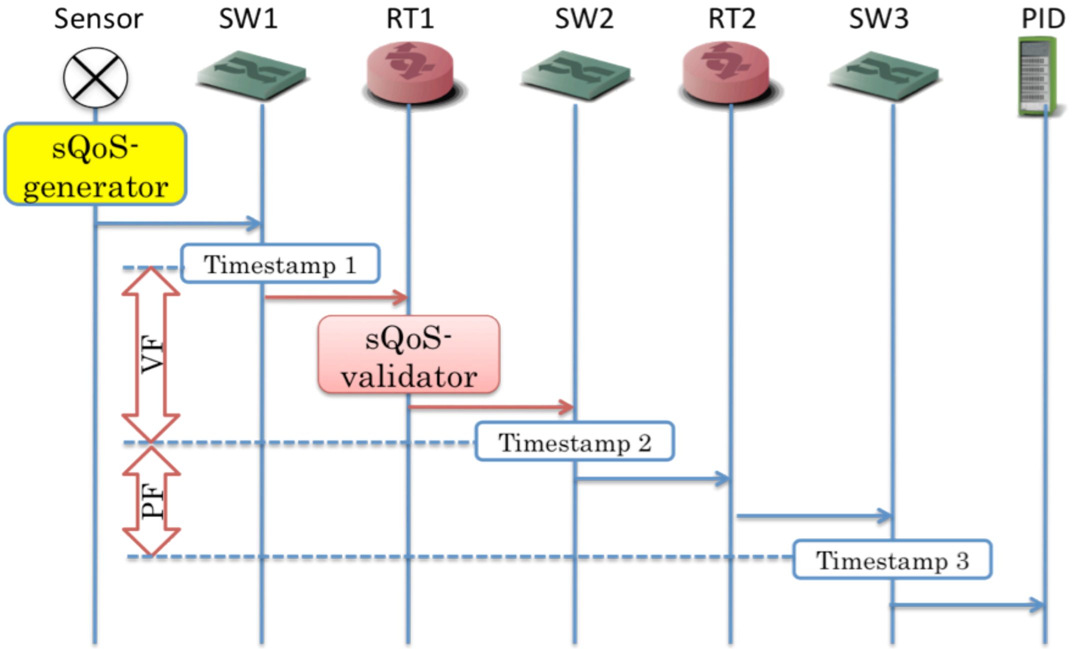


FIGURE 5.9 SEQUENCE TO EVALUATE SQOS-VALIDATOR

5.7.4 MEASUREMENT RESULT

The results from measurements are summarized in Table 5.2 and Table 5.3. The sQoS-generator took 66 microsecond for each process according to the results. The sQoS-validator took 131 microsecond for each process. The forwarding time for a router was 6 microsecond. The forwarding time for each switch was also measured, which was 4 microsecond. Thus, ten routers and two switches were assumed under the topology conditions between the Sensor and PID, and the End-to-End communication time was 1,444 microsecond, as shown in Eq. (5.4). This satisfied the allowed time of 45 ms.

TABLE 5. 2 LATENCY IMPACT OF SQOS-GENERATOR

	Average RTT	**Standard Deviation**	**Jitter**
GT (w/ sQoS-generator)	579	22	110
PT (w/o sQoS-generator)	447	16	110

Unit: microsecond

TABLE 5. 3 LATENCY IMPACT OF SQOS-VALIDATOR

	Average forwarding time	**Standard Deviation**	**Jitter**
VF (w/ sQoS-validator)	137	18	79
PF (w/o sQoS-validator)	6	16	91

Unit: microsecond

$$(66 + 131\times10 + 6\times10 + 4\times2) = 1{,}444 \quad (5.4)$$

5.8 CONCLUSIONS AND FUTURE WORKS

This chapter introduced a method of preventing QoS Spoofing Attacks, which are new and unique threats in PA networks. The method that was introduced provided five required features of: (1) Spoofing Packet Detection, (2) IP Compliancy, (3) QoS Method Agnosticity, (4) Prevention of Replay Attacks, and (5) Less Overhead. sQoS satisfied the goals of features (1) to (4).

The author prototyped and measured the overhead of the proposed method to evaluate (5). As a result, the overhead was sufficiently small to satisfy the shortest PID control cycle of 300 ms in the assumed typical PA network topology.

The author intends to evaluate the prototype under various conditions in the future since he only evaluated the prototype under limited conditions in this research, e.g., evaluate QoS protection in busy networks and/or when QoS Spoofing Attacks occur. In addition, measuring the overhead time with stronger HMAC algorithms, such as HMAC-SHA-256 [54], and measurements with a variety of packet sizes should be evaluated. Moreover, integration with appropriate key management protocols, which make the methods smarter, should be studied further. These results from evaluation and integration indicate that sQoS should prove usable in a variety of applications.

Chapter 6

OFsQoS: A OpenFlow Based Secure Method to Protect Bandwidth Control for Process Automation

6.1 Introduction

PID control is commonly used in PA, which is achieved through periodical data exchanges. Real-time communications are required to achieve correct PID control since data that do not arrive on schedule cannot be used for PID control. Thus, most existing PA systems have utilized proprietary wired communication technologies. In addition, the PA systems have been separated based on geographical conditions. Each separated PA system is called an Area. However, a new approach to connect such separated PA systems with networks has recently been emerging to improve flexibility, management efficiency, and economic efficiency based on the remarkable evolution of ICT. Typical examples of such approaches are introducing WSNs and the Industrial Backhaul, which is a plant wide network that interconnects distributed WSNs and existing PA systems, such as those in central control rooms. Plant users, on the other hand, are aware of threats by cyber attacks that target plants, such as those by Stuxnet and Night Dragon. These facts mean the co-existence of two requirements, i.e., encouraging network utilization and discouraging network utilization in plants. PA users, vendors, and integrators are facing a difficult situation in simultaneously satisfying both requirements.

The ISA-TR100.15.01-2012 Backhaul Architecture Model [4] has introduced a variety of devices and applications other than PID control that are utilized in plants such as Mobile HMIs and Video Cameras. However, installing dedicated cables for individual applications is not possible from the perspective of cost. Thus, multiple Areas with different security policies are attached to the Industrial Backhaul and a variety of applications transmit packets on it.

Methods of QoS are used to provide real-time communications in this environment. QoS methods generally require intermediate network equipment to classify packets based on the attributes in given packets to differentiate behaviors against these. For example, DiffServ requires the network equipment to investigate the field of DSCP to classify packets. However, the attributes to be investigated are designed to be rewritten by the intermediate equipment. Thus, if an attacker intentionally transmits improper high priority packets (Spoofing Packets), they can create additional delays or packet loss in proper high priority packets, such as PID control packets. These kinds of cyber attacks are called QoS Spoofing Attacks in this study. As Langner [9] and Miller and Rowe [50] have pointed out, cyber attacks targeting plants use security holes in Operating Systems or holes in security management. Unauthorized devices attached to Areas or the Industrial Backhaul could transmit Spoofing Packets. Delay or loss of PID control packets may result in unexpected blackouts or explosions at plants. Unexpected blackouts at plants result in enormous economic damage and explosions result in enormous damage to the environment and health of personnel or neighbors. Thus, QoS Spoofing Attacks on real-time communications should be prevented. The proposed method allows secure real-time communications over the Industrial Backhaul with a novel bandwidth protecting function on OpenFlow networks. While the ISA Model [4] mentions utilizing IPv6 and IPv4 on the Industrial Backhaul, this model assumes IPv6 will be used since IPv6 is expected to be widely used in the future.

The rest of this chapter is structured as follows. Subsection 6.2 presents challenges and objectives. Subsection 6.3 presents an overview of the proposed method and Subsection 6.4 presents the design of the proposed method. Subsection 6.5 presents the prototype, Subsection 6.6 presents the evaluation, and Subsection 6.7 presents considerations. Finally, Subsection 6.8 provides the conclusion.

6.2 CHALLENGES AND OBJECTIVES

There has been some research that has allowed intermediate equipment to authenticate packets. However, this research has not protected real-time communications by preventing spoofing of packet attributes used in QoS such as DSCP. Flow based bandwidth control is the key to ensuring real-time communications, as was previously mentioned. Thus, the main objective of the proposed method is to allow PID control over the Industrial Backhaul by protecting real-time communications from QoS Spoofing Attacks with a novel method of providing secure per flow bandwidth control. The three challenges and detailed goals are as follows.

(1) Per flow bandwidth control

Bandwidth control, which allows dedicated bandwidth allocation, is required to ensure real-time communications, as was described in Subsection 2.2. In addition, detailed per flow bandwidth control based on users or applications is required since packets from a variety of applications by a variety of users co-exist on an Industrial Backhaul. The flow is generally identified by five tuples of {source address, destination address, protocol, source port, destination port}. If some more attributes (e.g., DSCP) are used, the flexibility of flow definitions is increased, which allows the class of packets from the same application to be changed depending on the situation in the application (e.g., normal or urgent operation).

The proposed method allows the flow to be defined with the combination of a variety of fields such as the DSCP proposed by Nichols et al. [18] and flow labels as report by Hu and Carpenter [38]. In addition, the method allows an output queue to be assigned for each flow. This approach achieves flexible bandwidth control that allows flow definitions with an arbitrary combination of packet fields for each flow and bandwidth allocation in the defined flow. For example, a flow is identified by a combination of {source address, destination address}, and another flow is identified by a combination of {source address, destination address, destination port, DSCP}, while a different bandwidth allocated queue is assigned.

(2) Prevention of QoS Spoofing Attacks on intermediate equipment

QoS Spoofing Attacks are based on output queue consumption in intermediate equipment. Thus, End-to-End architecture based packet authentication cannot prevent these kinds of attacks. Intermediate equipment needs to detect spoofing packets before the output queue is consumed to prevent QoS Spoofing Attacks.

The intermediate equipment needs to classify the packets into predefined flows based on one or more attributes in the packets to achieve per flow bandwidth control. Thus, the intermediate equipment needs to detect the spoofed attributes to prevent QoS Spoofing Attacks. The proposed method allows arbitrary combinations of packets attributes, as was mentioned in (1).

Therefore, the function to detect spoofing packets needs to be able to cover such arbitrary combinations.

Even though the method can detect spoofing packets, attackers can cause long delays and packet losses by re-transmitting numerous previously sent proper packets (Replay Attacks). Thus, Replay Attacks need to be detected as well.

PLA and TinySec can detect spoofing packets and re-transmitted packets. However, these methods cannot work with per flow bandwidth control. OpenFlow can provide per flow bandwidth control. However, OpenFlow does not allow packet authentication on intermediate equipment. Thus, the proposed method should provide per packet authentication that allows intermediate equipment to detect spoofed attributes utilized for QoS classification with arbitrary combinations and detect retransmitted packets to protect the allocated bandwidth.

(3) Guaranteed small latency for PID control

Even though the proposed method detects spoofing packets to prevent QoS Spoofing Attacks, the proposed method should not disrupt real-time communications with its own processing. In other words, real-time communications of periodical PID control need to be provided even in environments where the proposed method is used. The author has specifically assumed the world's largest class of plant with ten Areas connected in a ring topology, which is commonly used in plants, as a plant reference model. The proposed method should allow a 300 ms PID control loop in such an environment. Items (1), (2), and (3) stated above were set as the goals for this proposed method.

6.3 OVERVIEW OF PROPOSED METHOD

6.3.1 BASIC POLICY

OpenFlow (Version 1.3.1) allows flexible flow definitions through arbitrary combinations of 40 tuples. In addition, OpenFlow allows packet processing rules to be specified for each flow such as bandwidth control or routing. Thus, the author chose OpenFlow intermediate equipment that was available to provide per flow bandwidth control. A new function should be introduced to accomplish per packet authentication, which is not provided by OpenFlow.

Methods of authenticating packet encryption and/or hash algorithms are widely deployed (e.g., IPsec). One typical authentication algorithm is HMAC. HMAC algorithms should be used in the proposed method. Since HMAC is based on hash algorithms, which require less calculation, it is helpful in guaranteeing small latency. Sequence numbers should be used in the proposed method to prevent Replay Attacks. Sequence numbers have also been introduced to IPsec or PLA.

The proposed method is based on IPv6 since this is expected to be deployed more widely in the Industrial Backhaul [4]. This study assumed the Industrial Backhaul would be installed throughout the vast Sites of plants. In addition, the Industrial Backhaul has a ring topology to provide redundancy at reasonable cost. Moreover, each Area is connected to the Industrial Backhaul via OFS. A dedicated network between OFC and OFSs is required in the existing OpenFlow environment in addition to the Data Forwarding Plane. The dedicated network is called a Secure Channel. It is difficult to provide an additional dedicated line only for the Secure Channel for networks installed at vast Sites. Thus, this study has assumed the Secure Channel is overlaid on the Data Forwarding Plane. Koide and Shimonishi [42] proposed a method of overlaying the Secure Channel based on a unique tunneling technique.

This proposed method had a policy to provide a bandwidth protection function to OpenFlow based networks by introducing per packet authentication. Such bandwidth protection is not provided by related work, as was explained in Subsection 2.6. The proposed method in this study is represented as OpenFlow Based Secure QoS (OFsQoS).

6.3.2 OVERVIEW

Figure 6.1 overviews the Industrial Backhaul with OFsQoS installed, which was used as the proposed structure. Sensors in typical PID control obtain data to send to the PID controller. The PID controller carries out calculations with the received data and sends configuration data to the actuator. The actuator sets the received data and sends back the configured value to the PID controller. This series of processes is called a PID control loop. The control loop is periodically repeated. The control loop consists of three Pub/Sub communications with different combinations of {source address, destination address}. A number of control loops run in a plant in parallel. Packets for Pub/Sub need to arrive at destinations on schedule; otherwise, the data are not used for PID control. Thus, this proposed method protects the real-time communications of Pub/Sub packets.

Sending devices in this structure place the authentication code in sent packets to detect spoofing packets. The authentication code is validated by the dedicated devices before the packets are forwarded by intermediate network equipment. OFS is chosen for the intermediate network equipment to provide flexible flow definitions and per flow bandwidth control. The network control functions in the OpenFlow environment are centralized in OFC and a simple forwarding function is provided by OFS to provide programmability to network control. This means that there is no programmability on OFS. OFC provides per flow forwarding rules to OFS, which stores the rules in the FlowTables to refer to when it processes (forwards or discards) packets. OFC does not need to provide per packet processing by using this structure, which helps to prevent increasing the load on OFC.

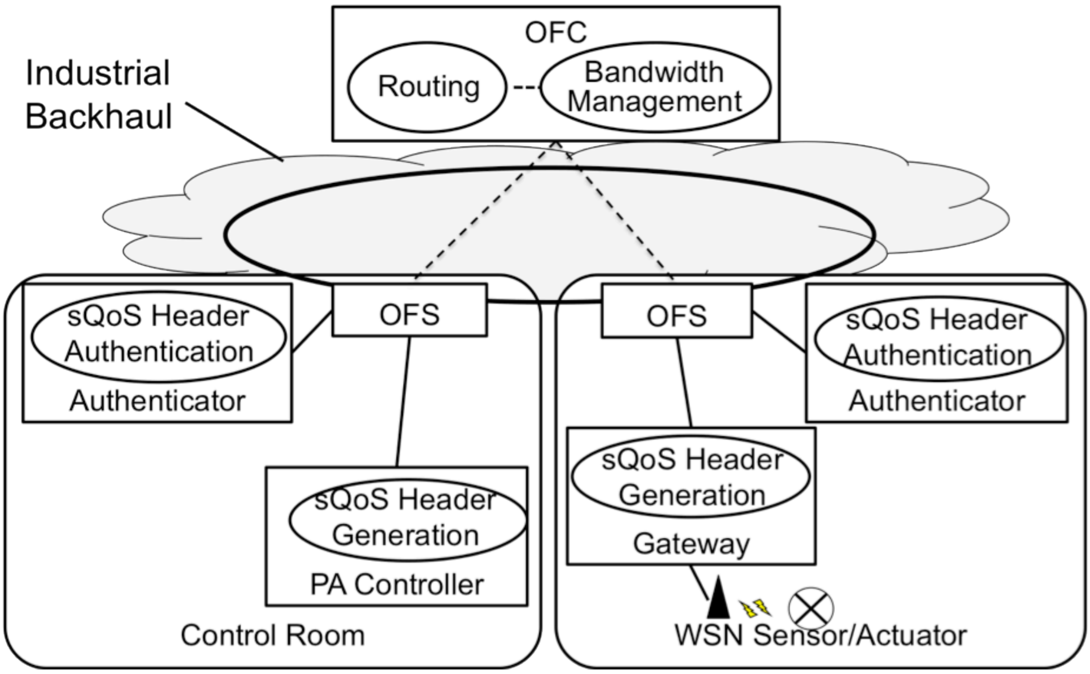


FIGURE 6.1 OFSQOS OVERVIEW

The per packet authentication provided by the proposed method cannot run on the Data Forwarding Plane since authentication requires per packet calculations. The Data Forwarding Plane is not designed to provide per packet calculations. Thus, authentication is not suitable for OFS. In addition, per packet authentication is not a kind of per flow processing, which is provided by the Network Control Plane. This means authentication is not suitable for OFC. Thus, pmqFlow introduces a new plane in addition to the Network Control Plane and the Data Forwarding Plane to achieve per packet authentication.

HMAC calculations need a secret key, which needs to be shared with all devices that generate MAC (i.e., packet generators) or validate MAC (i.e., authenticating intermediate equipment). When the key is leaked, an attacker can generate proper MAC and succeed in QoS Spoofing Attacks. Thus, when the key is leaked, it should be updated in all key holding devices. The key could be manually configured. However, the group key management protocol [55], which can simultaneously update the key in multiple devices, is helpful in reconfiguration when a key is leaked.

6.4 DESIGN

Figure 6.2 outlines the elements of OFsQoS. OFsQoS utilizes OFS for the Data Forwarding Plane and OFC for the Network Control Plane, as was mentioned in Subsection 6.3.1.

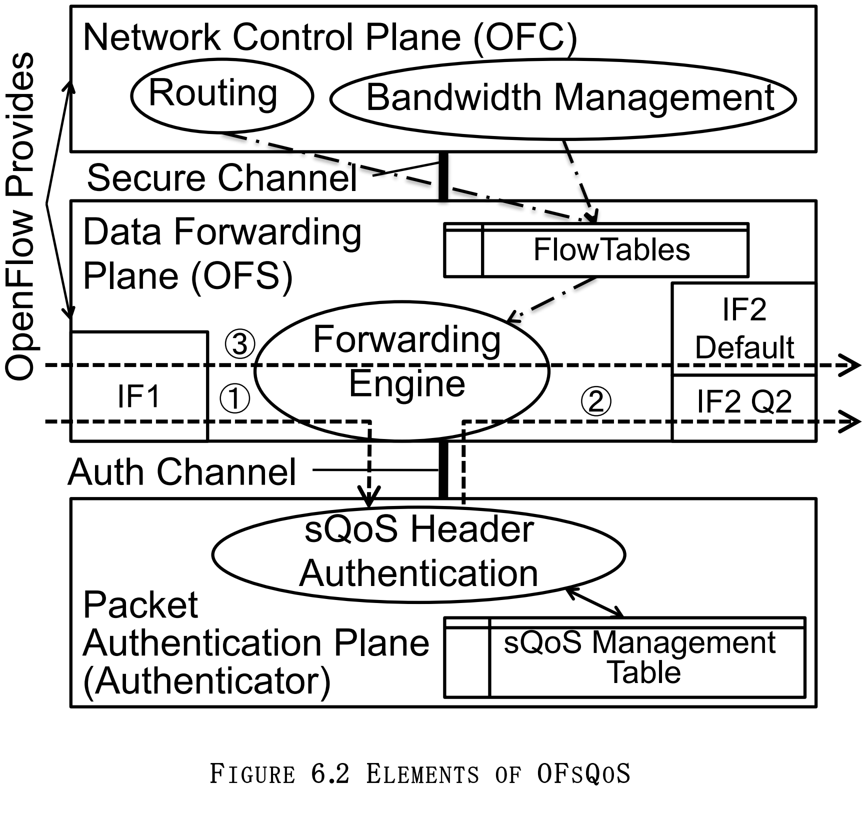


FIGURE 6.2 ELEMENTS OF OFSQOS

Per flow bandwidth control (Goal (1)) is achieved by assigning a flow to a bandwidth allocated output queue on OFSs from OFC. Neither PLA nor TinySec can work with such a bandwidth control function. An sQoS header is introduced to authenticate packets. This header is generated and attached to packets on the packet generator. The packets are authenticated by the function of “sQoS Header Authentication” on the “Packet Authentication Plane”, which is introduced by OFsQoS. After authentication, OFS dispatches the packets to an appropriate output queue. OFsQoS is designed to run “sQoS Header Authentication” on different devices from OFS since OFS does not have enough computing resources. OFsQoS prepares a dedicated communication path, which is so called “Auth Channel”, to transmit packets to “sQoS Header Authentication”. Detecting and discarding improper packets with the structure mentioned above protects real-time communications for PID control. The following has a detailed description of each function.

6.4.1 sQoS Header Generation

MAC should be placed on packets to prevent QoS Spoofing Attacks. Thus, OFsQoS utilizes an IPv6 Hop-by-Hop Option header, which was designed to contain information to be investigated by intermediate equipment. When there are constraints on the resources of WSN devices or the WSN protocol disallows the sQoS header from being directly placed, the gateway devices between WSN and the Industrial Backhaul should add the header on behalf of WSN end devices (e.g., sensors).

Figure 6.3 outlines the header format utilized in OFsQoS. The header is called an sQoS header. Spoofing Packets (e.g., with high priority value of DSCP) should be detected to prevent QoS Spoofing Attacks, as was mentioned in Subsection 6.3. In addition, the combination of attributes to identify the flow depends on the network operation policy, and thus the fields to be authenticated need to be specified according to the policy. Thus, OFsQoS prepares an “Authentication Flag” to specify the fields to be authenticated. The length of the “Authentication Flag” is one byte. Each bit from the right most bit is mapped to a destination port field, a source port field, a flow label field, a DSCP field, a destination IP address field, and a source IP address. The two left most bits are reserved for future use. This mapping allows arbitrary combinations of commonly utilized attributes to identify the flow.

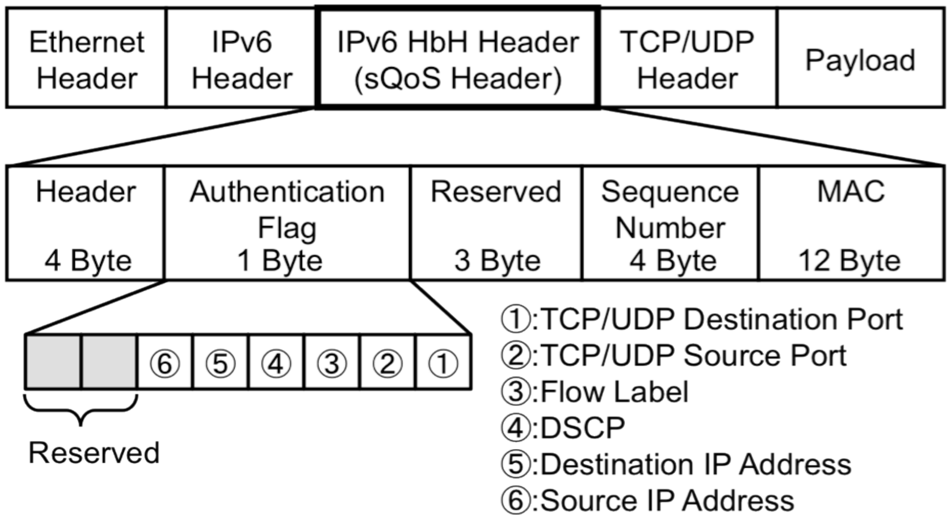


FIGURE 6.3 sQoS HEADER FORMAT

The packet generator places MAC calculated by the HMAC algorithm to prove that a given packet has been sent from a proper device and has not been improperly modified. The MAC is generated with target fields, an sQoS Header with zero padding MAC and a shared secret key. The calculated MAC is placed in the MAC field. Involving the sQoS Header for MAC calculations helps to protect the sequence number in the sQoS Header. In addition, increasing sequence numbers helps to add fluctuations to the MAC values. MAC without using sequence numbers could be identical with previously sent packets that had the same payload data, and attackers could succeed in QoS Spoofing Attacks. This MAC and Authentication Flag help to prevent QoS Spoofing Attacks, which is goal (2). OFsQoS utilizes SHA1 for the hash algorithm used in HMAC calculations (HMAC-SHA1), which is light and strong enough to prevent QoS Spoofing Attacks. HMAC-SHA1 helps to achieve small latency in goal (3). The MAC field should be 12 bytes in length, which has sufficient strength as IPsec.

OFsQoS places linearly increasing sequence numbers to allow sQoS Header Authentication to detect packets for Replay Attacks. sQoS Header Authentication compares the sequence numbers of the received packets with the stored sequence numbers, which are copied from previous authenticated packets. If the received numbers are equal to or smaller than those stored, the packets can be considered to be replayed packets and should be discarded. The sequence numbers are protected from spoofing attacks, since they are part of the sQoS Header, as was previously mentioned. The field of sequence numbers is 4 bytes in length, which can represent up to 4, 294, 967, 295. The three communications of PID control should be handled as individual flows. Thus, the sequence numbers can be used for 40.86 years for PID control flows with a 300 ms cycle according to Eq. (6.1). This is sufficient length for plants since plant lifetimes are generally up to 30 years. The sequence numbers work to prevent Replay Attacks, which is part of goal (2).

$$\frac{4{,}294{,}967{,}295 \times 0.3}{3600 \times 24 \times 365} = 40.86 \tag{6.1}$$

6.4.2 Packet Authentication Plane

OFsQoS introduces a new dedicated plane, which is so called Packet Authentication Plane, to provide per packet authentication. As Curtis et al. pointed out [56], the reliability of OFC is significantly important in OpenFlow based networks since network control is centralized on it. If authentication is provided on the Network Control Plane rather than the Packet Authentication Plane, attackers can cause DoS attacks by sending a number of improper high priority packets. Such improper packets could increase the load on OFC, since all such packets would be examined by OFC. The approach of using OFC for per packet authentication breaks the design of OpenFlow, which separates the Network Control Plane from the Data Forwarding Plane to make OFC secure. Thus, OFsQoS prepares a new dedicated plane for per packet authentication. OFsQoS utilizes OFSs for the Data Forwarding Plane and OFC for the Network Control Plane.

6.4.3 sQoS Header Authentication

This subsection describes sQoS Header Authentication on the Packet Authentication Plane. First, this function validates sequence numbers. If the received sequence numbers are equal to or smaller than the number of previous packets, the received packets should be discarded since they could be Replay Attack packets. If the received sequence numbers are larger than the previous ones, the MAC value should be validated.

MAC is validated with the HMAC-SHA1 algorithm for values of the field specified by the Authentication Flag, sQoS Header, and pre-shared secret key. sQoS Header Authentication calculates the MAC value. If the calculated value is equivalent to the MAC value in received packets, the received packets can be identified as proper packets that have been sent from proper devices and have not been modified. If the calculated MAC is different from that received, the received packets can be identified as improper packets. The improper packets should be discarded since there is no need to send error messages to the attacker. When the received packets are identified as proper packets, the sequence numbers in the received packets should be stored in corresponding entries of the sQoS Management Table. After that, the packets are sent back to the Data Forwarding Plane to be forwarded. A series of processes are required to authenticate the sQoS Header: sequence number validation, MAC validation, and sequence number storage, as was previously explained. Thus, OFsQoS is designed to provide all of these processes in sQoS Header Authentication since all of them are closely related. This sQoS Header Authentication helps to detect QoS Spoofing Attacks and Replay Attacks, which are parts of goal (2). In addition, the lightweight HMAC-SHA1 algorithm helps to achieve small latency, which is part of goal (3), like sQoS Header Generation.

6.4.4 AUTH CHANNEL

This subsection describes the dedicated communication path between sQoS Header Authentication and OFSs. Network appliances are generally placed on the Data Forwarding Plane and configure the route to make target packets go through appliances to provide various services (e.g., Deep Packet Inspection (DPI)) in OpenFlow networks. However, if OFsQoS places sQoS Header Authentication on the Data Forwarding Plane, the packets to be authenticated consume the bandwidth of the Data Forwarding Plane for the purpose of authentication. This means that bandwidth cannot be protected. Thus, the path from OFS to sQoS Header Authentication should be separated from the Data Forwarding Plane. OFsQoS prepares a dedicated channel for such a communication path, which is so called Auth Channel. The Auth Channel helps to protect the allocated bandwidth, which is part of goal (2).

OFS cannot inspect the Hop-by-Hop Option header in OpenFlow specifications. Thus, all the packets that should be dispatched to the protected queue should be transmitted to sQoS Header Authentication via the Auth Channel. The packets that should be dispatched to unprotected queues should be forwarded in the Data Forwarding Plane without passing the Auth Channel.

The packets in the Auth Channel should be forwarded without changing authentication with any attributes in any way. Two means can be used to achieve this forwarding function: encapsulation and native forwarding. OFsQoS uses native forwarding since it involves lighter processing and was originally prepared by OFS. OFsQoS requires OFS to have different forwarding rules for identical packets, as we can see from Figure 6.2. When a packet to be authenticated arrives at an OFS, the packet should be forwarded to the Auth Channel. When an identical packet arrives after authentication, it should be forwarded to an appropriate next hop on the Data Forwarding Plane. Note that both packets are completely identical. Network switches generally deal with packets with one rule. Thus, OFsQoS utilizes OpenFlow based routing control to provide different forwarding rules for packets. OpenFlow based routing control can classify identical packets before authentication and after authentication into different flows. This classification is achieved by using information from the ingress network port as flow matching conditions. A packet to be authenticated before authentication in this design is classified into a Suspicious Flow to forward to the Auth Channel (① in Figure 6.2). A packet after authentication is classified into an Authenticated Flow to forward to the Data Forwarding Plane (② in Figure 6.2). This routing control enables the Auth Channel to protect the output queue on Data Forwarding Plane.

6.5 PROTOTYPE

6.5.1 PURPOSE OF PROTOTYPE

OFsQoS was prototyped based on the previously explained design. The main purpose of this prototyping was to evaluate whether the proposed method could satisfy the requirements for real-time communications by providing: (1) per flow bandwidth control, (2) bandwidth protection by preventing QoS Spoofing Attacks on intermediate equipment, and (3) guaranteed small latency for PID control.

6.5.2 COMPONENT

Figure 6.4 outlines the components for the prototype. All devices have a Core i7-2600 CPU and a 8 GB of DDR3 SDRAM (PC3-10600). OFC utilized Ubuntu 12.04.1 and the others utilized Ubuntu Server 12.04.2 LTS. Each network interface was a Gigabit Ethernet.

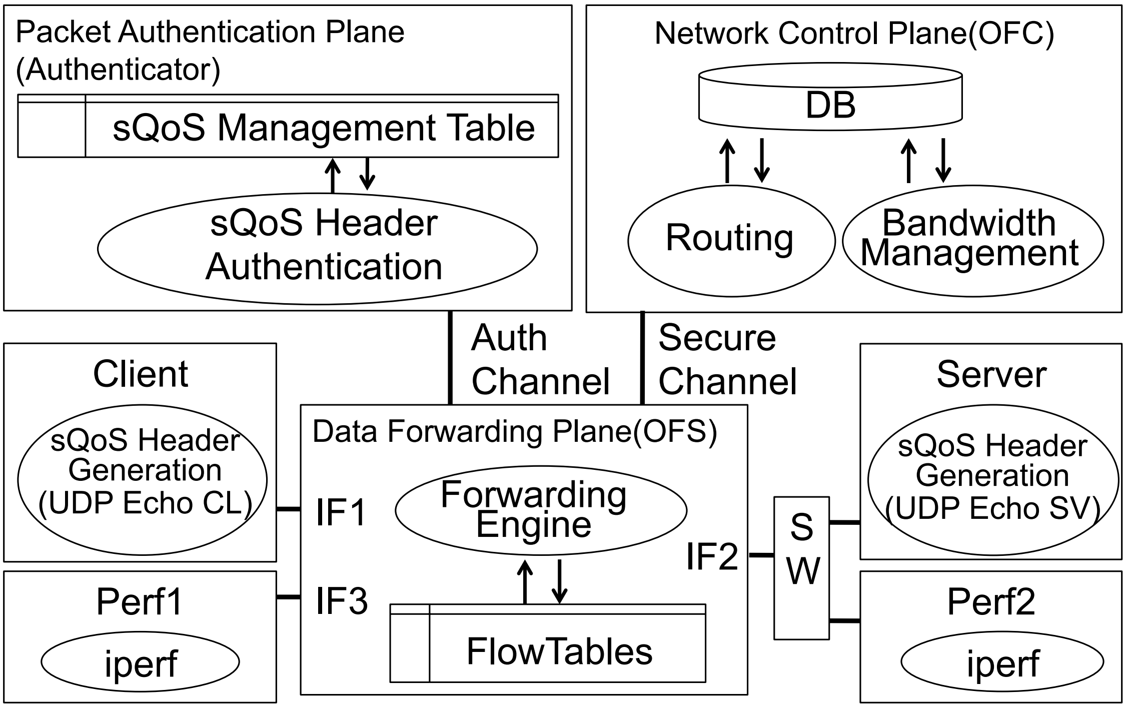

Packet Authentication Plane
(Authenticator)
sQoS Management Table
sQoS Header Authentication
Network Control Plane(OFC)
DB
Routing
Bandwidth Management
Auth Channel
Secure Channel
Client
sQoS Header Generation (UDP Echo CL)
Data Forwarding Plane(OFS)
IF1
Forwarding Engine
IF2
IF3
FlowTables
S W
Server
sQoS Header Generation (UDP Echo SV)
Perf1
iperf
Perf2
iperf

FIGURE 6.4 COMPONENTS OF PROTOTYPE SYSTEM

(1) OpenFlow Controller

OFC needed to support IPv6 since this method utilized IPv6. The Trema-edge was chosen in this prototyping since it was an IPv6 available SDK for OFC.

(2) OpenFlow Switch

OFS needed to support IPv6 since this method utilized IPv6. The implementation of an Openvswitch repository was chosen, since it was an IPv6 available OFS.

(3) Authenticator

sQoS Header Authentication worked in the Authenticator. The authenticating application needed to validate forwarded packets that arrived through the Auth Channel. However, the packets were forwarded as-is. This meant that packets were not destined for either the Authenticator or at sQoS Header Authentication. OFsQoS utilized ebtables [57], ip6tables [58], and netlink provided in Linux to make authenticating applications to be able to receive such packets.

(4) Client

The Client is a device on which the UDP Echo client application (UDP Echo CL) works. This client application transmits packets containing the sQoS Header. It indicates the RTTs of UDP echo packets obtained from timestamps when they are sent and received. The obtained time is used to evaluate the effect and impact of OFsQoS. Only UDP Echo CL application works on the Client to enable accurate evaluations. In addition, the Client is attached to the OFS directory.

(5) Server

The Server is a device on which the UDP Echo server application (UDP Echo SV) works. The server application uses UDP port 8000 for the Echo Service. The packets replied to by this server application have the sQoS Header. Since the packets from the Client and Server are classified into different flows, this server application generates the sQoS Header by itself and places the header into the reply packets.

(6) Perf1

Perf1 is a device that generates disturbance traffic. The iperf 2.0.5 client works on it. Perf1 is attached to the OFS directory to maximize the amount of traffic from Perf1.

(7) Perf2

Perf2 is a device that generates disturbance traffic. The iperf 2.0.5 server works on it. The packets from Perf1 are addressed to Perf2. Perf2 and the Server share a network port of OFS (IF2) due to the insertion of an additional switch to make the impact of disturbance packets noticeable.

6.6 EVALUATION

The configurations in Table 6.1 and Table 6.2 were installed onto the FlowTable 1 of OFS for the former and the FlowTable 2 of OFS for the latter to evaluate the behavior and performance of the proposed method. The match field represents the definition of each flow. OFS processes matched packets according to the commands in the Instruction field. Two FlowTables were prepared, as was previously mentioned. OFS refers to the FlowTables in the order of FlowTable 1 to 2. The number of rules means priority. Larger numbers have higher priority.

FlowTable 1 contains the rules to use the Auth Channel. FlowTable 2 contains rules to forward the packets in the Data Forwarding Plane for the Authenticated Flow, or flows do not need to be authenticated. A definition of Suspicious Flows is required to use the Auth Channel. Thus, packets that have 0x2E on the DSCP field are defined as Suspicious Flows (rule 8 in FlowTable 1). The value is probably used for QoS Spoofing Attacks, since it has the highest priority in DiffServ. Authenticated packets returning to OFS via the Auth Channel are dealt with as Authenticated Flows. Flows are defined to be forwarded to the Data Forwarding Plane according to FlowTable 2 (rule 9 in FlowTable 1). Concrete configurations and evaluation procedures are described below.

TABLE 6. 1 CONFIGURATION OF FLOWTABLE 1

Rule No.	**Match**							**Instruction**	
	Src Addr	**Dst Addr**	**Proto**	**Src Port**	**Dst Port**	**DSCP**	**INPUT IF**	**OUTPUT IF**	**Queue**
9	ANY	ANY	ANY	ANY	ANY	ANY	Auth Channel	Goto FlowTable 2	N/A
8	ANY	ANY	ANY	ANY	ANY	0x2E	ANY	Auth Channel	Default
0	ANY	ANY	ANY	ANY	ANY	ANY	ANY	Goto FlowTable 2	N/A

TABLE 6. 2 CONFIGURATION OF FLOWTABLE 2

Rule No.	**Match**							**Instruction**	
	Src Addr	**Dst Addr**	**Proto**	**Src Port**	**Dst Port**	**DSCP**	**INPUT IF**	**OUTPUT IF**	**Queue**
9	Client	Server	UDP	ANY	8000	0x2E	ANY	IF2	Q2
8	Server	Client	UDP	8000	ANY	0x2E	ANY	IF1	Q1
7	Client	Server	ANY	ANY	ANY	ANY	ANY	IF2	Default
6	Server	Client	ANY	ANY	ANY	ANY	ANY	IF1	Default
4	Perf2	Perf1	ANY	ANY	ANY	ANY	ANY	IF3	Default
3	Perf1	Perf2	ANY	ANY	ANY	ANY	ANY	IF2	Default
0	ANY	ANY	ANY	ANY	ANY	ANY	ANY	Secure Channel	Default

(1) Per flow bandwidth control

A flow of a UDP Echo Request with 0x2E of DSCP is assigned to output queue Q2 on IF2 (rule 9 in FlowTable 2) to confirm per flow bandwidth control in this proposed method and 50 Mbps is allocated to Q2. A flow of a UDP Echo Reply with 0x2E of DSCP is assigned to output queue Q1 on IF1 (rule 8 in FlowTable 2). Other traffic is assigned to default output queues on corresponding output interfaces (rules 7, 6, 4, and 3 in FlowTable 2).

The Client sends the UDP Echo Request to port 8000 (DSCP=0x2E) of the Server and the ICMP Echo Request (DSCP=0x00) to the Server every 300 ms with the above configuration to measure RTT. Perf1 simultaneously transmits disturbance traffic to UDP port 5000 (DSCP=0x00) on Perf2 in each measurement. The disturbance traffic was escalated in 100 Mbps steps. The packets for UDP Echo and ICMP Echo were 256 bytes to align them with PID control messages. The packets of disturbance traffic were set to 1500 bytes, which was the largest size of IPv6 over the Ethernet, to make the impact on latency and jitter noticeable. This procedure confirmed that UDP Echo packets and ICMP Echo packets were using different output queues by observing the impact on disturbance traffic. Figure 6.5 and Figure 6.6 outline the packet flows to evaluate ICMP Echo and UDP Echo.

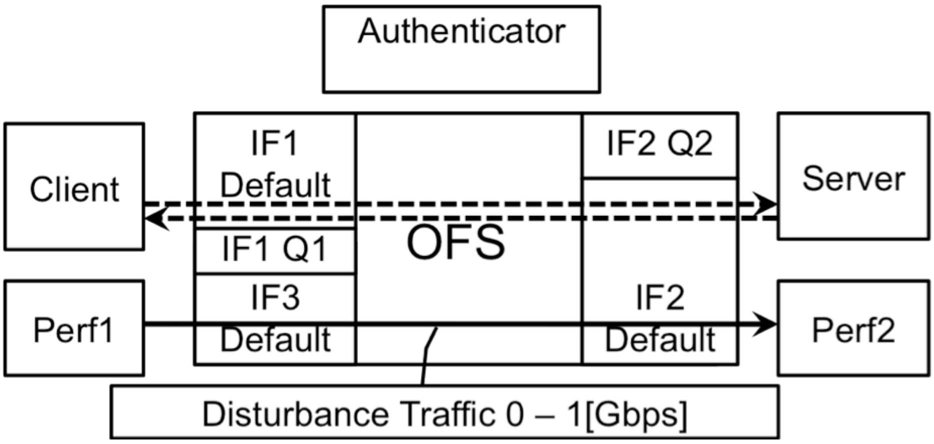


FIGURE 6.5 EVALUATION OF BANDWIDTH CONTROL EFFECT (ICMP ECHO)

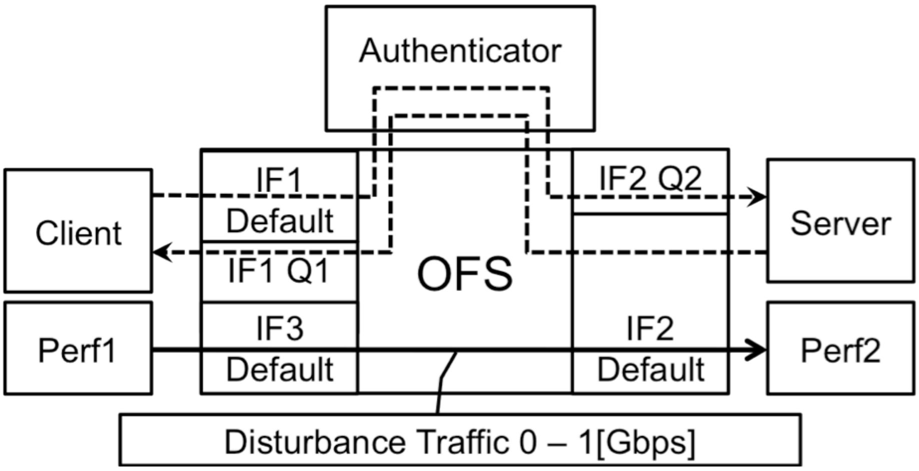


Figure 6.6 Evaluation of Bandwidth Control Effect (UDP Echo)

Figure 6.7 plots the results obtained from both evaluations. The values used in the figure are the average times for 100 measurements and they indicate that ICMP Echo packets were strongly affected by disturbance traffic.

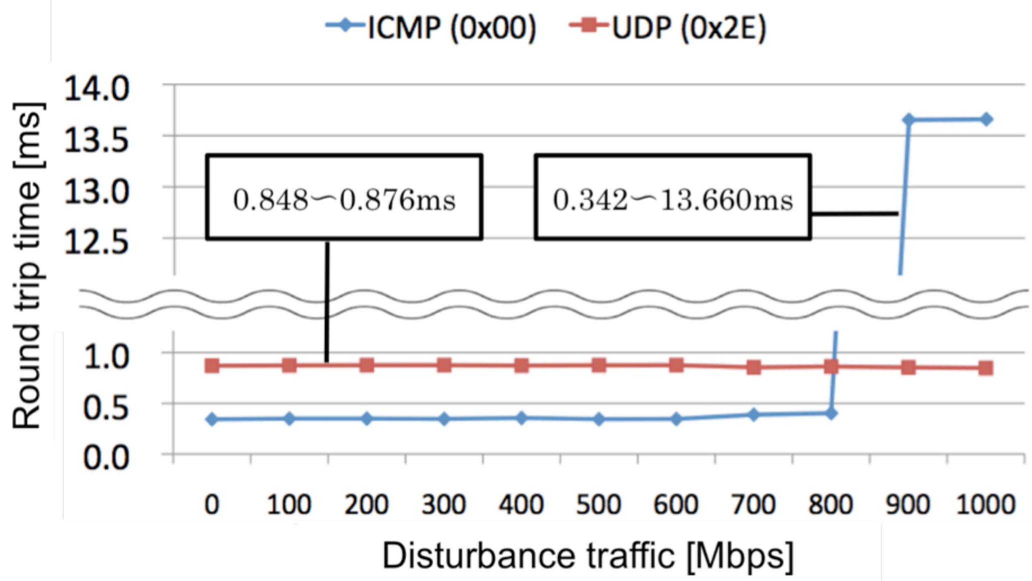


Figure 6.7 Evaluation of Bandwidth Control Effect

The RTT of ICMP Echo packets was 0.342 ms without disturbances, while it took 13.66 ms with 1 Gbps of disturbance. UDP Echo packets, on the other hand, were not affected. The RTT of UDP Echo packets was 0.848 to 0.876 ms regardless of the amount of disturbance traffic. This is because ICMP Echo packets and disturbance traffic used default output queue (rules 7 and 3 in FlowTable 2), while UDP Echo (DSCP=0x2E) used dedicated output queue after passing the Auth Channel (rule 9 in FlowTable 2). This evaluation confirmed that the proposed method could provide per flow bandwidth control to flows defined by arbitrary combinations of attributes, while using the Packet Authentication Plane.

(2) Bandwidth protection with preventing QoS Spoofing Attacks on intermediate equipment

Disturbance traffic was assigned to the output queue that was assigned to the protected UDP Echo to confirm whether the introduced Packet Authentication Plane could protect the bandwidth allocated to a particular flow. Two measured results were compared to confirm the effect of the Packet Authentication Plane; the first used the Packet Authentication Plane and the second did not. The former condition was called "with OFsQoS" and the latter was called "without OFsQoS". Table 6.3 was prepared as FlowTable 1 and Table 6.4 was prepared as FlowTable 2 for this evaluation.

Table 6.3 Configuration of FlowTable1 to skip OFsQoS

Rule No.	Match							Instruction	
	Src Addr	Dst Addr	Proto	Src Port	Dst Port	DSCP	INPUT IF	OUTPUT IF	Queue
0	ANY	ANY	ANY	ANY	ANY	ANY	ANY	Goto FlowTable 2	N/A

Table 6.4 Configuration of FlowTable 2 for Evaluation of Bandwidth Protection

Rule No.	Match							Instruction	
	Src Addr	Dst Addr	Proto	Src Port	Dst Port	DSCP	INPUT IF	OUTPUT IF	Queue
9	Client	Server	UDP	ANY	8000	0x2E	ANY	IF2	Q2
8	Server	Client	UDP	8000	ANY	0x2E	ANY	IF1	Q1
7	Client	Server	ANY	ANY	ANY	ANY	ANY	IF2	Default
6	Server	Client	ANY	ANY	ANY	ANY	ANY	IF1	Default
5	Perf1	Perf2	ANY	ANY	ANY	0x2E	ANY	IF2	Q2
4	Perf2	Perf1	ANY	ANY	ANY	ANY	ANY	IF3	Default
3	Perf1	Perf2	ANY	ANY	ANY	ANY	ANY	IF2	Default
0	ANY	ANY	ANY	ANY	ANY	ANY	ANY	Secure Channel	Default

Table 6.1 was used for evaluation with the Packet Authentication Plane, while Table 6.3 was used for evaluation without the Packet Authentication Plane. Rules 8 and 9 were removed from Table 6.1 for Table 6.3. The configuration in Table 6.3 allowed OFS to skip the Auth Channel and forward all packets according to FlowTable 2 (Table 6.4). Rule 5 (shadowed line) was added to Table 6.2 for Table 6.4. This added entry allowed disturbance traffic (DSCP=0x2E) to share Q2 (50 Mbps bandwidth) on IF2 with protected UDP Echo. Thus, UDP Echo would be disturbed when using Table 6.3 and Table 6.4. This condition will be called the "no-OFsQoS test" after this. The Packet Authentication Plane should discard disturbance packets when using Table 6.1 and Table 6.4. Thus, UDP Echo packets would be protected.

The Client sent a UDP Echo Request to port 8000 (DSCP=0x2E) of the Server every 300 ms to measure RTT to evaluate the above behaviors. Perf1 simultaneously transmitted disturbance traffic to UDP port 5000 (DSCP=0x2E) on Perf2 in each measurement. The disturbance traffic was escalated in 10 Mbps steps. Figure 6.8 plots both results obtained from the evaluation. The line chart in the figure represents RTT, which corresponds to the left-hand scale (units of ms). The bar chart represents the Packet Loss Rate (PLR), which corresponds to the right-hand scale (units of percent (%)).

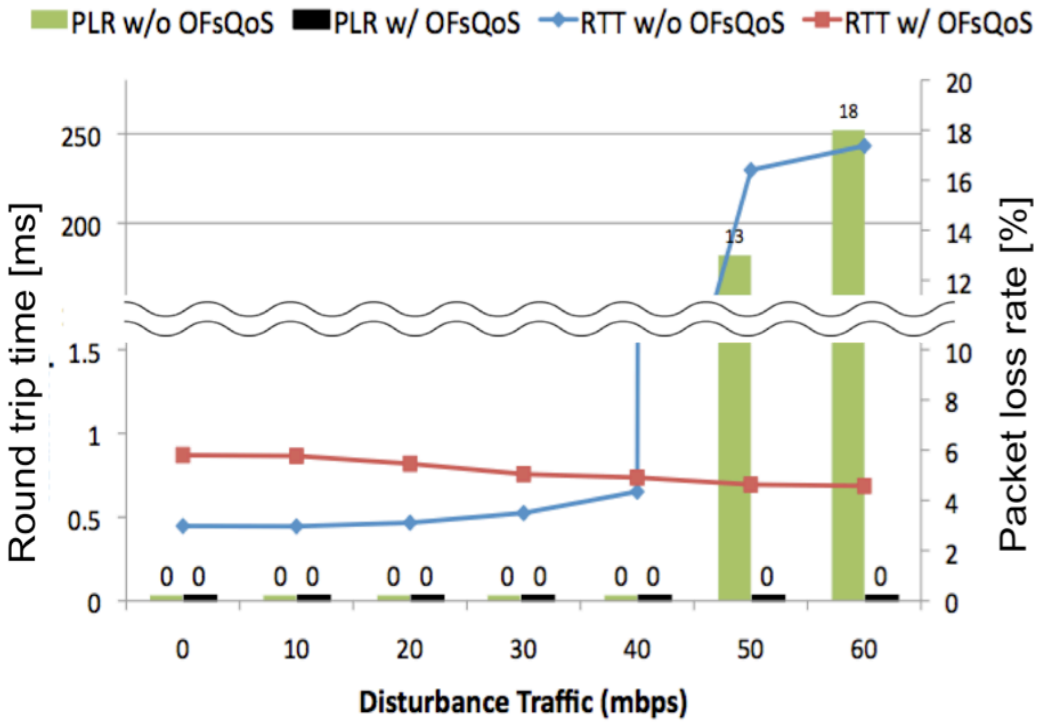


FIGURE 6.8 EVALUATION OF BANDWIDTH PROTECTION BY OFSQOS

RTT without the Packet Authentication Plane exceeded 200 ms and 13% of PLR was observed when disturbance traffic reached 50 Mbps (the allocated bandwidth of Q2 on IF2). However, no impact of disturbance traffic was observed with the Packet Authentication Plane, even when the traffic exceeded the allocated bandwidth of 50 Mbps. Note that the bandwidth of Q2 on IF2 was consumed by packets to be classified into Suspicious Flows if the Packet Authentication Plane was not used. The path was equivalent to a path where sQoS Header Authentication was placed on the Data Forwarding Plane; it was the path OFsQoS was designed to avoid by preparing the dedicated Auth Channel. The observed results confirmed that the Auth Channel and the Packet Authentication Plane worked well to protect the allocated bandwidth from improper packets.

(3) Guaranteed small latency for PID control

The RTT between the Client and Server were measured to evaluate the impact on latency caused by the proposed method under three different conditions of (A) generating and authenticating the sQoS Header, (B) generating the sQoS Header but not authenticating it, and (C) not generating the sQoS Header. One hundred measurements were done and the average, maximum, and minimum RTT for each condition were obtained. The average RTT for each condition were respectively labeled A_{AVE}, B_{AVE}, and C_{AVE}. The elapsed time on the Packet Authentication Plane (T_{VAL}) and the elapsed time to generate the sQoS header (T_{GEN}) were calculated with Eqs. (6.2) and (6.3).

$$T_{VAL} = \frac{A_{AVE} - B_{AVE}}{2} \tag{6.2}$$

$$T_{GEN} = \frac{B_{AVE} - C_{AVE}}{2} \tag{6.3}$$

Figure 6.9 outlines the measured RTT. The required time to use the Packet Authentication Plane was 0.212 ms (Eq. (6.4)) according to Eq. (6.2). The required time to generate the sQoS Header was 0.053 ms (Eq. (6.5)) according to Eq. (6.3).

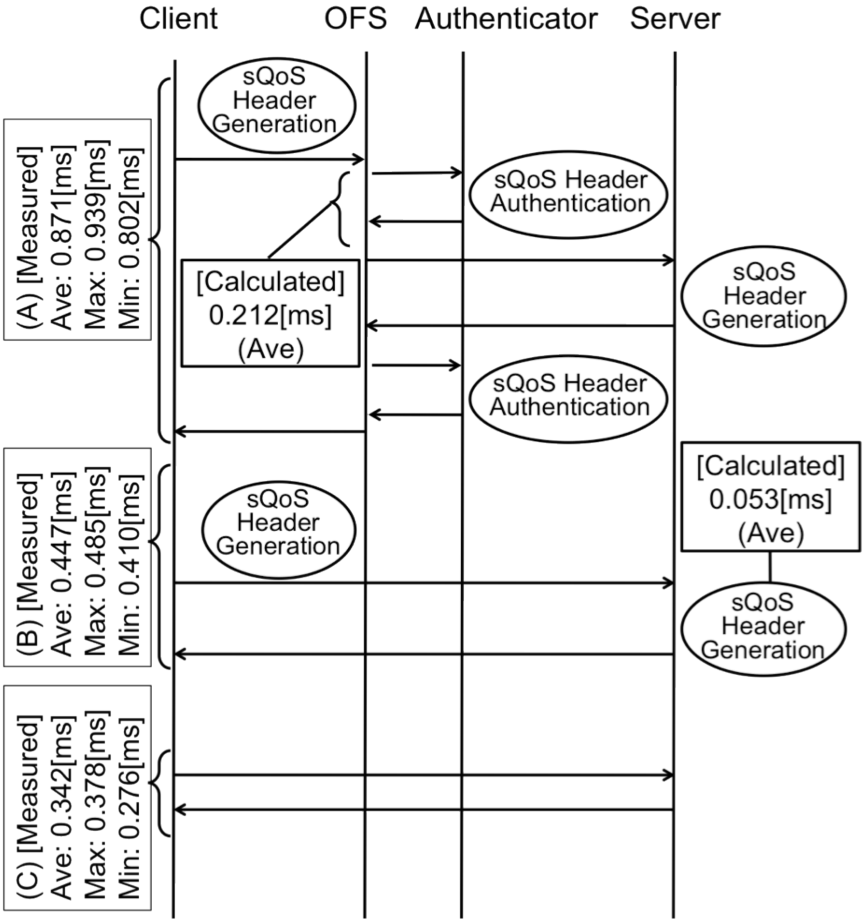

Client
OFS
Authenticator
Server
sQoS Header Generation
sQoS Header Authentication
sQoS Header Generation
sQoS Header Authentication
[Calculated] 0.212[ms] (Ave)
(A) [Measured] Ave: 0.871[ms] Max: 0.939[ms] Min: 0.802[ms]
(B) [Measured] Ave: 0.447[ms] Max: 0.485[ms] Min: 0.410[ms]
sQoS Header Generation
[Calculated] 0.053[ms] (Ave)
sQoS Header Generation
(C) [Measured] Ave: 0.342[ms] Max: 0.378[ms] Min: 0.276[ms]

FIGURE 6.9 EVALUATION OF LATENCY

$$\frac{0.871 - 0.447}{2} = 0.212 \tag{6.4}$$

$$\frac{0.447 - 0.342}{2} = 0.053 \tag{6.5}$$

The total overhead time was 2.173 ms (Eq. (6.6)) to transmit packets to the Packet Authentication Plane when there were ten OFSs, i.e., the maximum number of OFS in the assumed Industrial Backhaul. With the condition of the processing time of the Sensor was 30 ms, that of the the PID Controller was 45 ms, that of the Actuator was 90 ms, the allowed time to achieve the 300 ms PID control loop was 45 ms (Eq. (6.7)). Thus, it was confirmed that the proposed method could satisfy the required latency of goal (3).

$$0.212\times 10 + 0.053 = 2.173 \tag{6.6}$$

$$\frac{\left(300 - (30 + 45 + 90)\right)}{3} = 45 \tag{6.7}$$

6.7 CONSIDERATION

The evaluation confirmed that the proposed method could provide per flow bandwidth control based on OpenFlow. Neither TinySec, PCP, or PLA can provide such bandwidth control. TinySec does not have the QoS concept, and PLA and PCP can only classify packets into eight classes. In addition, PLA and PCP only provides priority control rather than bandwidth control.

The results obtained from evaluations indicated that the Packet Authentication Plane and the Auth Channel could protect bandwidth with per packet authentication, which required per packet calculations. The existing OpenFlow architecture cannot provide per packet processing. In addition, the results from evaluations demonstrated the effect of the Auth Channel by showing that the authentication function on the Data Forwarding Plane (like Deep Packet Inspection) did not work to protect bandwidth. However, if numerous packets, which are classified as Suspicious Flows, inflow into the network, they may exceed the bandwidth prepared for the Auth Channel. Adding another network port or using link aggregation for the Auth Channel would work in such cases to increase the bandwidth of the Auth Channel.

OFS has two options when the Auth Channel is unavailable: 1) forward the packets to the Data Forwarding Plane without passing the Auth Channel or 2) drop the packets. Option 1) is preferable from the perspective of the dependability of real-time communications since option 2) completely stops real-time communications. Thus, OFC needs to reconfigure the routing information on OFS when it detects the unavailability of the Auth Channel.

The evaluation revealed the explicit overhead of OFsQoS. However, the overhead was much smaller than the allowed time for PID control. This is because OFsQoS utilized a simple packet forwarding function, which OFS natively provided, for the Auth Channel and utilized a lightweight HMAC-SHA1 algorithm. The overhead and security strength were a trade-off. The small overhead in the evaluated environment indicated the possibility of utilizing a stronger algorithm for sQoS Header Authentication or packet encapsulation for the Auth Channel. In addition, it is possible to reduce the overhead by utilizing a lighter algorithm with a higher risk of the possibility of QoS Spoofing Attacks.

The proposed method could achieve all the goals according to the overall evaluation results. sQoS Header Authentication, on the other hand, needs to be reliable since crucial packets go through sQoS Header Authentication in OFsQoS.

The Packet Authentication Plane and the Auth Channel were implemented outside OFS in the presented prototyping. It would be possible to implement the Packet Authentication Plane and the Auth Channel inside OFS by introducing a virtualized environment into OFS as long as an Auth Channel with sufficient bandwidth separated the Packet Authentication Plane from the Data Forwarding Plane.

Since this proposed method can utilize appliances for OFS, the bottleneck of scalability will be in sQoS Header Authentication, which is provided as software. The results from evaluations indicated that the required time to use the Packet Authentication Plane was 0.212 ms. With the simple calculation in Eq. (6.8), 4,716 packets could be authenticated every second. According to Eq. (6.9), 707 pairs of client/server communications are available every 300 ms. The PID control loop, which consists of three communications, has 471 available loops every 300 ms according to Eq. (6.10).

$$\frac{1000}{0.212} = 4{,}716 \tag{6.8}$$

$$\frac{4{,}716\times 0.3}{2} = 707 \tag{6.9}$$

$$\frac{4{,}716\times 0.3}{3} = 471 \tag{6.10}$$

The world’s biggest class of plant [5] has 888 PID control loops obtained through simple calculations. Two sQoS Header Authentications can cover all the loops. Since the proposed method can provide an sQoS Header Authentication function in a scale-out manner, it is possible to provide appropriate scalability based on the size of the plant.

The proposed method can be utilized for multicast packets. However, if authentication is carried out after numerous copies are created, the copies can increase the load on sQoS Header Authentication. This means that attackers can succeed in DoS attacks on sQoS Header Authentication. sQoS Header Authentication should be optimally located to discard improper packets before they are copied to prevent such DoS attacks. For example, if the first OFS for an improper multicast packet generator leads multicast packets to sQoS Header Authentication, just one authentication is required. This is the same as that for unicast packets. Improper multicast packets for the OFS internal configuration need to be forwarded to sQoS Header Authentication when they arrive, as listed in Table 6.1. These approaches can prevent DoS attacks that use multicast packets.

This proposed method to provide secure real-time communications in the Industrial Backhaul built with intermediate equipment available with OpenFlow is expected to be widely deployed. The definition of the Authentication Flag in the sQoS header needs to be modified to specify PCP in the IEEE802.1Q Header to protect QoS in the Industrial Backhaul that is based on the existing technology of IEEE802.1Q with the proposed method. In addition, the OFS to which sQoS Header Authentication is attached should be inserted onto the edge of the Industrial Backhaul to authenticate the packets. The FlowTables in OFS also need to be configured to forward packets to sQoS Header Authentication via the Auth Channel. The main limitation of this approach is the number of classes (eight) since protected QoS is PCP based priority control.

OFC has full responsibility for per flow network control and OFS plays a limited role in packet processing based on the configurations provided by OFC. Thus, many methods have been proposed to protect OFC from cyber attacks. Benton et al. [59] pointed out the possibility of Man-in-the-Middle Attacks, Spoofing Attacks, and Snooping Attacks against OpenFlow communications between OFC and OFS. Thus, they proposed using TLS, which provides packet encryption and mutual authentication between OFC and OFS, to prevent pointed attacks. In addition, they pointed out the possibility of DoS attacks by sending huge numbers of packets that were reported from OFS to OFC to obtain processing rules. They proposed allowed flows be predefined to prevent DoS attacks.

Spoofing Neighbor Discovery (ND) packets on the Data Forwarding Plane in IPv6 networks can create confusion in the topology management function on OFC. This attack can be considered to be an attack on the Network Control Plane from the Data Forwarding Plane. Such ND packet spoofing can be prevented with SEND [60].

6.8 CONCLUSION

Many methods of preventing attacks on OFC have been proposed, as was previously explained. However, they cannot prevent QoS Spoofing Attacks, since the OpenFlow architecture is not designed to provide an authentication function that requires per packet calculations in OFS or OFC. Thus, OpenFlow should be integrated with the Packet Authentication Plane, which prevents QoS Spoofing, to allow PID control over the Industrial Backhaul.

The author has proposed a method of protecting real-time communications in PID control from QoS Spoofing Attacks, which spoof attributes in packets. The method that was designed was prototyped and evaluated. The results revealed that the newly added Packet Authentication Plane worked well to protect bandwidth while achieving the three main goals of this method that were: (1) per flow bandwidth control, (2) prevention of QoS Spoofing Attacks on intermediate equipment, and (3) guaranteed small latency for PID control. These achievements indicate that PA applications that require real-time communications can safely utilize the Industrial Backhaul.

The proposed method utilizes native per flow bandwidth control of OpenFlow to achieve these goals. However, as OpenFlow was not designed to provide per packet processing, the proposed method introduced a new plane that could provide per packet processing to detect spoofing packets and to protect bandwidth. In addition, the proposed method was designed to prevent DoS attacks on the Network Control Plane, which sends numerous packets to be authenticated, by separating the authentication function from OFC.

The prototyping and evaluation of OFsQoS utilized only one Authenticator. The reliability of the Authenticator should be considered to avoid single points of failure to utilize OFsQoS in real environments since all real-time communications go through the Authenticator. Thus, a method should be developed to make the Authenticator reliable. In addition, OFsQoS was designed to forward packets after authentication. This process adds 0.2 ms to the forwarding time for each authentication. This is the main reason there are limitations in applications that require faster real-time communications. Parallel processing of forwarding and authentication should be considered in the future to make OFsQoS more applicable.

Chapter 7

CONCLUSIONS AND FUTURE PLANS

7.1 SUMMARY OF ACHIEVEMENT

This study intended to provide dependable real-time communications management to allow utilization of applications requiring real-time communications represented by PID control and Mobile HMI over the Industrial Backhaul. Based on the feedback control method, commonly utilized in PA, this study has identified following five essential functions for the management: (1) Bandwidth Control, (2) Bandwidth Allocation, (3) Assessment of Real-time Communications, (4) Prevention of Unauthorized Bandwidth Allocation, and (5) Prevention of QoS Spoofing Attacks. As the consequence of the survey, "(3) Assessment of Real-time communications" and "(5) Prevention of QoS Spoofing Attacks" were not yet satisfied. Thus, this study defined two goals of developing methods to satisfy these two functions. The goals are; "management method for real-time communications based on packet propagation time monitoring" and "Hop-by-Hop packet authentication method to protect the output queue".

A method to reconfigure the bandwidth allocation based on the actual packet propagation time has been developed to address "management method for real-time communications based on packet propagation time monitoring" as presented in Chapter 4. The method is called pmqFlow. pmqFlow allows monitoring propagation time of the particular range of real-time communications path rather than monitoring entire RTT to avoid the impact of unrelated intermediate equipment and traffic. pmqFlow also allows detecting the impact of spike traffic to protect the entire traffic of the flow including spike traffic. Thus, the goal of "management method for real-time communications based on packet propagation time monitoring" was achieved. This means that the function of "(3) Assessment of Real-time communications" was satisfied.

A method to authenticate real-time communications packets in a Hop-by-Hop manner has been developed to address "Hop-by-Hop packet authentication method to protect the output queue” as presented in Chapter 5. The method is called sQoS. sQoS allows detecting and discarding the spoofing packets on intermediate network equipment within the allowed time for PID control. In addition, a method to protect the bandwidth with sQoS in OpenFlow available networks has been developed as presented in Chapter 6. The method is called OFsQoS. OFsQoS introduced the novel Packet Authentication Plane to protect the bandwidth by detecting and discarding the spoofing packet before the allocated bandwidth is consumed. In addition, the Packet Authentication Plane allows installing scalable packet authentication function while protecting the Data Forwarding Plane and the Network Control Plane. It was confirmed that OFsQoS could protect the bandwidth within the allowed time for PID control. Thus, the goal of “Hop-by-Hop packet authentication method to protect the output queue” was achieved. This means that the function of “(5) Prevention of QoS Spoofing Attacks” was satisfied.

The methods developed by this study satisfied the two functions of “(3) Assessment of Real-time communications” and “(5) Prevention of QoS Spoofing Attacks”. Thus, all of the required five functions for dependable real-time communicants management were satisfied. As a consequence, the objectives of this study, “to provide dependable real-time communications management for PA operations over Industrial Backhaul”, was achieved.

7.2 Findings from This Study

7.2.1 Feedback Control of Real-time Communications in Industrial Backhaul for PA

This study revealed that feedback control of real-time communications is effective. In addition, the control is required to achieve the dependable real-time communications in the Industrial Backhaul for PA.

The proposed method of pmqFlow monitors the actual propagation time. The method accesses whether the configuration on the network equipment satisfies the requirements for real-time communications. It is an essential function for the dependable real-time communications management. This study revealed that the propagation time monitoring function can assess the real-time communications and the assessment information is useful to reallocate the bandwidth to ensure the real-time communications. Especially, the propagation time monitoring function can detect spike traffic, a situation that utilized bandwidth exceeds the allocated bandwidth for a while, to protect the real-time communications including spike traffic itself. On the other hand, the propagation time monitoring function is not suitable to detect or predict the increase of traffic while the utilized bandwidth is less than allocated bandwidth, since the impact on propagation time is not visible in such situations. In addition, even when the Bandwidth Exceedance happens, the impact may be still invisible as long as enough unused bandwidth is available on default output queue. Thus, the propagation time monitoring function might not detect the Bandwidth Exceedance depending on the condition of default output queue. Even though the propagation time is still less than Timeout value, it is not ensured, since the propagation time can have impact (e.g., Timeout or packet loss) when traffic passing the default output queue increased. This means that the situation of Bandwidth Exceedance should be eliminated. Thus, it was revealed that utilizing propagation time monitoring function alone is not sufficient to prevent the Timeout in advance since it does not work well for detection and prediction of Bandwidth Exceedance.

The bandwidth utilization monitoring function such as [22] and [24], can detect the changes in bandwidth utilization regardless of the occasion of Bandwidth Exceedance. Thus, this bandwidth utilization monitoring function is useful to detect and predict the Bandwidth Exceedance. In other word, it is useful to prevent the Timeout or packet loss in advance. This study revealed that the bandwidth utilization monitoring functions could detect and predict the Bandwidth Exceedance to ensure the real-time communications. Bandwidth utilization monitoring function, on the other hand, cannot assess the achievement of real-time communications since it cannot know the actual propagation time. In addition, the bandwidth utilization monitoring function may not detect the Bandwidth Exceedance of spike traffic since the obtained bandwidth utilization is the average amount of monitored period. Thus, it was revealed that utilizing bandwidth utilization monitoring function alone is not sufficient to ensure the configuration and protect the real-time communications including spike traffic.

When allocating the bandwidth statically, the administrator needs to estimate all the real-time traffic completely for each flow in advance. It is difficult and requires huge effort since unexpected event, such as sudden equipment trouble or urgent operations, can happen in the real network. In general, to cover all traffic including such unexpected traffic, all of the possible traffic shall be listed up and accumulated strictly to allocate enough bandwidth to cover the obtained peak traffic statically. It is not efficient in terms of bandwidth utilization, since the allocated bandwidth is exclusive for assigned flow and some part of the bandwidth is unused in normal operations. This study proposed the dynamic bandwidth allocation to adjust to the actual traffic conditions rather than static bandwidth allocation covering peak traffic. This approach can provide both efficient bandwidth utilization and ensured real-time communications for dynamic traffic. It is an essential function for the dependable real-time communications management. From the evaluation presented in Subsection 4.6.3, it was shown that monitoring the traffic condition and reconfiguring the network equipment take up to about 16 ms in total. It was revealed that required time is acceptable for focused PA applications including Mobile HMI, which requires one second for response time.

The dynamic bandwidth allocation is applicable for the applications requiring shorter response time by adjusting monitoring cycle. In addition, the required time can be shorten by monitoring bandwidth utilization on OFSs in parallel and configuring bandwidth allocation on OFSs in parallel. According to Eq. (7.1), the minimum time to complete the possessing is about 3 ms. That means, the proposed method is applicable for applications that accept 3 ms of fixing time.

$$1.794 + 0.654 + 0.522 = 2.970 \tag{7.1}$$

This study revealed the combination of propagation time monitoring function and bandwidth utilization monitoring function complement each other to monitor the real-time communications by providing all of the features of; assessing the real-time communications, detecting spike traffic, detecting and predicting the Bandwidth Exceedance, and detecting the unutilized bandwidth. In addition, this study revealed that the dynamic bandwidth allocation helps to improve the bandwidth utilization while satisfying the real-time communications with quick reconfiguration. As a consequence, it was revealed that the combination of two monitoring functions and bandwidth reconfiguration has sufficient features and performance to allow the feedback control of real-time communications. In addition, the feedback control is required to provide the dependable real-time communications in the Industrial Backhaul for PA.

7.2.2 Hop-by-Hop Security Architecture in Industrial Backhaul for PA

This study reveled that the Hop-by-Hop security architecture is required to protect the real-time communications. In other words, the End-to-End security architecture, which is commonly utilized in the Internet, can provide limited security functions. Thus, the End-to-End security architecture is not sufficient to protect the real-time communications. In addition, it was revealed that the Hop-by-Hop Security architecture is acceptable for the real-time communications in the Industrial Backhaul for PA.

The proposed method of OFsQoS, which is presented in Chapter 6, provides Hop-by-Hop packet authentication by offloading the authentication function from the OpenFlow switch to the newly added Packet Authentication Plane. It was indicated that OFsQoS works with bandwidth control function to keep propagation time in constant value within proposed architecture. On the other hand, it requires 0.212 ms of additional time to use the Packet Authentication Plane once. This means that the Hop-by-Hop packet authentication is applicable for PA applications such as PID control. Moreover, the required time can be shortened to 0.131 ms when the authentication function is implemented in the OFS as shown in Chapter 5. However, the applicability of this method has a limitation on utilization for the applications requiring faster real-time communications such as FA.

OFsQoS provides per packet authentication. The per packet authentication has scalability concern since the Packet Authentication Plane needs to validate all packets classified into Suspicious Flow. The evaluation indicated that two devices of Authenticator is enough to support the busiest OFS that interconnects the Area of the central control room and the Industrial Backhaul with strict condition of 300 ms PID control cycle. If the PID control cycle is one second, the most typical cycle of PID control, only one Authenticator is enough.

As a consequence, this study revealed that per packet authentication in a Hop-by-Hop manner allows to protect real-time communications within the required time for PA. The authentication is required to provide the dependable real-time communications in the Industrial Backhaul for PA, which is utilized for serious operation.

7.3 FUTURE PLANS

This study focused on Level 1 and Level 2 functions. However, for more advanced process control, Level 3 functions will work with Level 2 and Level 1 closer. Thus, the real-time traffic on the Industrial Backhaul will be increased. In addition, dependability of PA operation on the Industrial Backhaul will be increased. In such situation, scalability and reliability of the proposed methods will be essential. Especially, the reliability and scalability of the key functions, such as Authenticator in OFsQoS, need to be provided.

There is a group that discusses Network Functions Virtualization (NFV) [61] in European

Telecommunications Standards Institute (ETSI). The group is discussing an architecture that allows network functions (e.g., DPI, IPS, Antivirus) to work on commodity PC hardware rather than vendor proprietary appliance hardware. In other words, the architecture allows virtualization of network functions from physical network appliance. The architecture will provide high availability and scalability. It is same approach as cloud computing.

There is a new working group in IETF that discusses how to chain the virtualized network functions. The working group is called Service Function Chaining (SFC) [62]. SFC allows packets to go through multiple network appliances such as DPI, IPS, and Antivirus sequentially for example.

Even though most of the applications for NFV can work in the Data Forwarding Plane not the Packet Authentication Plane, the NFV architecture will be useful for Authenticator in OFsQoS. In addition, some security functions (e.g., IPS, IDS and Firewall) will be used in the Industrial Backhaul to improve security. Thus, OFsQoS should align with NFV and SFC architecture in the future.

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere gratitude and special thanks to my supervisor Dr. Mitaro Namiki, a professor of Tokyo University of Agriculture and Technology. He has offered this study opportunity and strong support with many advices. This study could not be achieved without his support.

I am highly grateful and appreciated of the members of evaluation committee of my dissertation, Dr. Nariyoshi Yamai, a professor of Tokyo University of Agriculture and Technology, Dr. Toshiyuki Kondo, a professor of Tokyo University of Agriculture and Technology, Dr. Hiroshi Yamada, an assistant professor of Tokyo University of Agriculture and Technology, and Dr. Shinya Sugiura, an assistant professor of Tokyo University of Agriculture and Technology, for their acceptance of reviewing this dissertation. Their advices are valuable and helpful to improve this research.

I am grateful to Dr. Mikiko Sato, an assistant professor of Tokyo University of Agriculture and Technology. She has advised me from the attitude toward the research to finalizing academic papers. I would like to express my gratitude to all the members of my laboratory for their warm and friendly attitude. Especially, I highly appreciate Mr. Ryuichi Sakamoto and Mr. Ken-ichi Suzuki for their technical advices. I am thankful to Miss Kazumi Kikuchi for her various administrative supports.

I am grateful to Dr. Tetsuya Otani, Mr. Mitsunori Fukuzawa, Mr. Takaharu Matsumoto, Mr. Shinji Oda, Mr. Koji Demachi, and Mr. Mitsuru Iwaoka for their understanding of this study and warm support in my company, Yokogawa Electric Corporation.

I would like to express my special gratitude to Mr. Kazushi Sugyo, one of “trema-edge” developers, for his plenty of valuable advices to develop new functions over “trema-edge”.

I am grateful to Dr. Yasushi Takagi in Nippon Telegraph and Telephone Corporation and Dr. Sho Fujita in Yokogawa Electric Corporation for their variety of advices based on their experiences.

I am grateful to Ms. Erica Johnson, a director of the University of New Hampshire's Interoperability Laboratory, for her warm support to improve this dissertation especially from English language perspective.

At last, I would like to represent my special thanks go to my families for their warm and deep cheers. Especially, I am deeply thankful my wife for her acceptance of this study, taking various burdens, and complete moral support.

March 2015

ACHIEVEMENTS

JOURNAL PUBLICATIONS

[1] Hiroshi Miyata, Mitaro Namiki, and Mikiko Sato, "Study on OpenFlow Based Bandwidth Control Security for Process Automation," IEICE TRANSACTIONS on Information and Systems, Vol.J97-D, No.6, Jun. 2014, pp.1068-1081. (Corresponds to Chapter 6)

[2] Hiroshi Miyata, Mitaro Namiki, and Mikiko Sato, "OpenFlow based Real-time Communication Control Method for Dynamic Traffic in Process Automation," IEICE TRANSACTIONS on Information and Systems, Vol.J98-D, No.3, Mar. 2015, pp.(TBD). Accepted to be published in Mar.2015. (Accepted for Publication) (Corresponds to Chapter 4)

[3] Kenichi Suzuki, Hiroshi Miyata, Mikiko Sato, and Mitaro Namiki, "Design of a Real-time Network in Virtual Machine Environment with OpenFlow," IPSJ Journal, Vol.56, No.1, Jan. 2015, pp.1-13. Jan. 2015 (Accepted for Publication).

INTERNATIONAL CONFERENCE PAPERS

[1] H. Miyata, M. Mitaro, and M. Sato, "sQoS: The design and prototyping of secure QoS for process automation system," IECON 2013-39th Annual Conference of the IEEE, Nov. 2013, pp.5680-5685. (Corresponds to Chapter 5)

[2] H. Miyata, M. Mitaro, and M. Sato, "pmqFlow: Design of Propagation Time Measuring QoS System with OpenFlow for Process Automation," IECON 2014-40th Annual Conference of the IEEE, Oct.2014, pp.3693-3699. (Corresponds to Chapter 4)

JAPANESE DOMESTIC SYMPOSIUM PAPERS

[1] H. Miyata, M. Mitaro, and M. Sato, "A Fundamental Study of OpenFlow Network Applied to the Industrial Control System," IPSJ, in proceeding of Swopp2012 Tottori, vol.2012-OS-122 No.1, Jul. 2012，pp.1-9. (Corresponds to Chapter 5)

[2] H. Miyata, M. Mitaro, and M. Sato, "A Fundamental Study of OpenFlow Technology Adaptation to the Industrial Control System," IPSJ, in proceeding of 54th Programming Symposium， vol.2013, Jan. 2013， pp.37-48. (Corresponds to Chapter 5)

[3] H. Miyata, M. Mitaro, and M. Sato, "An Fundamental Study of QoS Authentication Service Plane with OpenFlow," IPSJ, in proceeding of Dicomo 2013， vol.2013, Jul. 2014, pp.483-492. (Corresponds to Chapter 6)

REFERENCES

[1] IEC SC65E, "IEC 62264-1 ed2.0: Enterprise-control system integration - Part 1: Models and terminology," IEC, May 2013.

[2] IEC SC65C, "IEC 62734 ed1.0: Industrial networks - Wireless communication network and communication profiles - ISA 100.11a," IEC, Oct. 2014

[3] IEC SC65C. "IEC 62591 ed1.0: Industrial communication networks – Wireless communication network and communication profiles – WirelessHART," IEC, Apr. 2010

[4] ISA, "ISA-TR100.15.01-2012 Backhaul Architecture Model: Secured Connectivity over Untrusted or Trusted Networks, " ISA, 2012

[5] Fieldbus Foundation, "Major end users improve their process performance," Fieldbus Report, vol. 5, issue 2, pp. 17-18, October 2006.

[6] Industry4.0, http://www.plattform-i40.de/sites/default/files/Report_Industrie%204.0_engl_1.pdf

[7] Industrial Internet consortium, http://www.industrialinternetconsortium.org

[8] Smart Manufacturing Leadership Coalition, https://smartmanufacturingcoalition.org

[9] R. Langner, "Stuxnet : Dissecting a Cyberwarfare Weapon," IEEE Security & Privacy vol. 9 no. 3 pp. 49-51, May/June 2011.

[10] shodan, http://www.shodanhq.com

[11] Fieldbus Foundation, "System Architecture, FF0581-1.3," Fieldbus Foundation, October, 2003.

[12] IEEE Std 802.1Q, "IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks", IEEE, May 2006.

[13] R. Braden, D. Clark, and S. Shenker, "Request for comments 1633: Integrated Services in the Internet Architecture: an Overview," IETF, http://tools.ietf.org/html/rfc1633, Jun. 1994

[14] N. Mckeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," ACM SIGCOMM Computer Communication Review vol.38, pp. 69-74, April 2008.

[15] B. Pfaff, and B. Davie, "Request for comments 7074: The Open vSwitch Database

Management Protocol," IETF, http://tools.ietf.org/html/rfc7074, Dec. 2013.

[16] R.Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman, "Request for comments 6241: Network Configuration Protocol (NETCONF)," IETF, http://www.ietf.org/rfc/rfc6241.txt, June 2011.

[17] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "Request for comments 2475: An Architecture for Differentiated Services," IETF, http://www.ietf.org/rfc/rfc2475.txt, December 1998.

[18] K. Nichols, S. Blake, F. Baker, and D. Black, "Request for comments 2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," IETF, http://tools.ietf.org/html/rfc2474, December 1998.

[19] S.Amante, B. Carpenter, S. Jiang, and J. Rajahalme, "Request for comments 6437: IPv6 Flow Label Specification," IETF, http://www.ietf.org/rfc/rfc6437.txt, November 2011.

[20] P. Cholda, A. Mykkeltveit, B. E. Helvik, O. J. Wittner, and R. Jajszczyk, "A SURVEY OF RESILIENCE DIFFERENTIATION FRAMEWORKS IN COMMUNICATION NETWORKS," Communications Surveys and Tutorials, IEEE, Vol. 9, Issue 4, pp. 32-55, February 2008.

[21] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Request for comments 2205: Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification," IETF, https://tools.ietf.org/html/rfc2205, Sep. 1997.

[22] W. Kim, P. Sharma, J. Lee, S. Banerjee, J. Tourrilhes, S. Lee, and P. Yalagandula, "Automated and scalable QoS control for network convergence," INM/WREN ’10, pp.1-6, San Jose, USA, April 2010.

[23] B. Sonkoly, A. Gulyás, F. Németh, J. Czentye, K. Kurucz, B. Novák and G. Vaszkun, "OpenFlow virtualization framework with advanced capabilities," Software Defined Networking (EWSDN), 2012 European Workshop on, pp. 18-23, Darmstadt, Germany, October 2012.

[24] B. Sonkoly, A. Gulyas, F. Nemeth, J. Czentye, K. Kurucz, B. Novak, and G. Vaszkun. "On QoS Support to Ofelia and OpenFlow," Software Defined Networking (EWSDN), 2012 European Workshop on, pp. 109-113, Darmstadt, Germany, October 2012.

[25] H. E. Egilmez, S. T. Dane, K. T. Bagci, and A. M. Tekalp, "Open QoS: In OpenFlow Controller Design for Multimedia Delivery with End-to-End Quality of Service over Software-Defined Networks," Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 1-8, Dec. 2012

[26] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "Request for comments 3550:

RTP: A Transport Protocol for Real-Time Applications," IETF, http://tools.ietf.org/html/rfc3550, IETF, July 2003.

[27] D. Mills, J. Martin, J. Burbank, and W. Kasch, "Request for comments 5905: Network Time Protocol version 4: Protocol and algorithms specification," IETF, http://tools.ietf.org/html/rfc5905, June 2010.

[28] IEEE Std IEEE1588-2008, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," IEEE, July 2008.

[29] V. Talwar, and K. Nahrstedt, "Securing RSVP for Multimedia Applications," MULTIMEDIA '00 Proceedings of the 2000 ACM workshops on Multimedia, pp.153-156, Nov. 2000.

[30] T. Dierks, and E. Rescorla, "Request for comments 5246: The Transport Layer Security (TLS) Protocol Version 1.2," IETF, http://tools.ietf.org/html/rfc5246, Aug. 2008.

[31] C. Rigney, S. Wilens, A. Rubens, and W. Simpson, "Request for comments 2865: Remote Authentication Dial In User Service (RADIUS)," IETF, http://www.ietf.org/rfc/rfc2865.txt, June 2000.

[32] V. Fajardo, J. Arkko, J. Loughney and G. Zorn, "Request for comments 6733: Diameter Base Protocol," IETF, http://www.ietf.org/rfc/rfc6733.txt, October 2012.

[33] S. Kent, and K. Seo, "Request for comments 4301: Security Architecture for the Internet Protocol," IETF, http://tools.ietf.org/html/rfc4301, December 2005.

[34] S. Kent, "Request for comments 4302: IP Authentication Header," IETF, http://tools.ietf.org/html/rfc4302, December 2005.

[35] S. Kent, "Request for comments 4303: IP Encapsulating Security Payload," IETF, http://tools.ietf.org/html/rfc4303, December 2005.

[36] Kant, Krishna, Ravishankar Iyer, and Prasant Mohapatra. "Architectural impact of secure socket layer on internet servers." Computer Design, 2000. Proceedings. 2000 International Conference on. IEEE, 2000.

[37] Dmitrij Lagutin, "Redesigning Internet The Packet Level Authentication architecture," Licentiate's Thesis - HELSINKI UNIVERSITY OF TECHNOLOGY, May 2008.

[38] Q. Hu, and B. Carpenter, "Request for comments 6294: Survey of proposed use cases for the IPv6 flow label," IETF, http://tools.ietf.org/html/rfc6294, June 2011

[39] C. Karlof, N. Sastry, and D. Wagner, “TinySec: Link Layer Security for Tiny Devices,” In Proceeding of SenSys '04 Proceedings of the 2nd international conference on Embedded networked sensor systems

[40] H. Krawczyk, M. Bellare, and R. Canetti, "Request for comments 2104: HMAC: Keyed-Hashing for Message Authentication," IETF, http://tools.ietf.org/html/rfc2104, February 1997.

[41] Success Stories, http://www.yokogawa.com/success/petrochem/suc-cnpc.htm

[42] T. Koide and H. Shimonishi, "A study on automatic construction mechanism of control network in OpenFlow-based network," IEICE Technical Report, NS2009-105, pp.19-24. 2010.

[43] S. Bellovin, "A Best-Case Network Performance Model," ATT Research, Tech. rep., February 1992.

[44] R. S. Prasad, C. Dovrolis and A. M. Bruce, "The effect of layer-2 store-and-forward devices on per-hop capacity estimation," INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies. Vol. 3, pp. 2090-2100, SanFrancisco, USA, March/April 2003.

[45] Trema-edge, https://github.com/trema/trema-edge/

[46] openvswitch, http://openvswitch.org/

[47] iperf, http://sourceforge.net/projects/iperf/

[48] J. Rhee, A. Kochut, and K. Beaty, "DeskBench:Flexible Virtual Desktop Benchmarking Toolkit," Integrated Network Management, 2009. IM '09. IFIP/IEEE International Symposium on, pp. 622-629, Long Island, USA, June 2009.

[49] Wireshark: http://sourceforge.net/projects/wireshark/

[50] B. Miller and D. C. Rowe, "A Survey of SCADA and Critical Infrastructure Incidents," SIGITE'12, October 11–13, 2012, Calgary, Alberta, Canada

[51] S. Deering, and R. Hinden, "Request for comments 2460: Internet Protocol, Version 6 (IPv6) Specification," IETF, http://tools.ietf.org/html/rfc2460, December 1998.

[52] X.Wang, Y. L. Yin, and Y. Yu, "Finding collisions in the full SHA-1," Advances in Cryptology–CRYPTO 2005, Springer Berlin Heidelberg, pp. 17-36, 2005.

[53] Rafaeli, Sandro, and David Hutchison. "A survey of key management for secure group communication." ACM Computing Surveys (CSUR) 35.3 (2003): 309-329.

[54] S. Kelly, and S. Frankel, "Request for comments 4868: Using hmac-sha-256, hmac-sha-384, and hmac-sha-512 with ipsec," IETF, http://tools.ietf.org/html/rfc4868, May, 2007.

[55] J. Mattsson, and T. Tian, "Request for comments 6043: MIKEY-TICKET: Ticket-Based Modes of Key Distribution in Multimedia Internet KEYing (MIKEY)," IETF,

http://www.ietf.org/rfc/rfc6043.txt, March 2011.

[56] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, "Devoflow: Scaling flow management for high-performance networks," SIGCOMM Comput. Commun. Rev. vol. 41, no. 4, pp. 254-265, August 2011

[57] ebtables, http://ebtables.sourceforge.net/br_fw_ia/br_fw_ia.html

[58] ip6tables, http://www.netfilter.org/projects/iptables/

[59] K. Benton, L. J. Camp, and C. Small, "Openflow vulnerability assessment," Proc. 2nd ACM SIGCOMM workshop on Hot topics in software defined networking, pp. 151-152, Hong Kong, China, August 2013.

[60] J. Arkko, J. Kempf, B. Zill, and P. Nikander, "Request for comments 3971: Secure Neighbor Discovery (SEND)," IETF, http://www.ietf.org/rfc/rfc3971.txt, March 2005.

[61] NFV, https://portal.etsi.org/Portals/0/TBpages/NFV/Docs/NFV_White_Paper3.pdf

[62] IETF SFC WG, https://datatracker.ietf.org/wg/sfc/charter/