A Fuzzy-based Dynamic Method for Efficient Sharing Bandwidth in Local Physical Network

로컬 물리적 네트워크에서 효율적인 대역폭 공유를 위한 퍼지 기반의 동적 방법

JDCS 디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 18, No. 2, pp. 411-422, Apr. 2017

로컬 물리적 네트워크에서 효율적인 대역폭 공유를 위한 퍼지 기반의 동적 방법

Linh Van Ma[1] · 박상현[1] · 장종현[2] · 박재형[1] · 김진술[1*]
[1]전남대학교 전자컴퓨터공학과
[2]한국전자통신연구원

A Fuzzy-based Dynamic Method for Efficient Sharing Bandwidth in Local Physical Network

Linh-Van Ma[1] · Sanghyun Park[1] · Jong-hyun Jang[2] · Jaehyung Park[1] · Jinsul Kim[1*]
[1]School of Electronics and Computer Engineering, Chonnam National University, Gwangju, 61186, Korea.
[2]Electronics and Telecommunications Research Institute, Daejeon, 34129, Korea.

[요 약]

현재 정책은 대역폭 공유에 대하여 평균 처리량을 높이고 로컬 네트워크의 대역폭 사용률을 향상시킨다. 하지만 이러한 정책은 각각 네트워크 흐름에 대역폭을 할당해주는 중앙 관리에서 사용자 특성 기반의 리소스를 할당 할 수 없다. 따라서 사용자 특성기반의 서비스를 보장하지 않기 때문에 불평등한 대역폭 할당이 발생된다. 따라서 본 논문에서는 제한된 대역폭 네트워크에서 대역폭을 공유하는 새로운 방법을 제안한다. 제안한 방법은 이상적인 퍼지 시스템을 사용하여 제한된 장치의 현재 사용량에 따른 대역폭 요청여부를 추전하고 결정한다. 본 논문에서는 OPNET의 비디오 스트리밍 시뮬레이션과 WebRTC의 실시간 비디오 스트리밍으로 구성된 두 가지 실행을 수행한다. 수행한 실험 결과를 통해 제안 된 방법이 네트워크 환경에서 사용자의 요구사항을 기반으로 대역폭 사용을 유연하게 공유 할 수 있다는 것을 확인 할 수 있었다.

[Abstract]

Current policies for sharing bandwidth increase average throughput and improve utilization of the bandwidth in the local network. However, with these policies, a central administer, which is responsible for allocating bandwidth to each network flow, cannot allocate resources based on user characteristics. Thus, it leads to unfair bandwidth allocation because it does not guarantee services based on user characteristics. Therefore, we propose a novel negotiation method to share the bandwidth in a limited bandwidth network, in which, a user negotiates with other users to gain more resource. Ideally, we use a fuzzy system to infer and determine whether a device will request bandwidth or not based on the current usage of the given device. We conduct two experiments consisting of a video streaming simulation in OPNET and a real-time video streaming in WebRTC. The results of the experiment indicate that the proposed method can flexibly share the bandwidth utilization based on user's requirement in the network.

색인어 : 대역폭 사용, 퍼지 시스템, 네트워크 전송, 서비스 품질, 비디오 스트리밍
Key word : Bandwidth usage, Fuzzy System, Network Transmission, Quality of Service, Video Streaming

http://dx.doi.org/10.9728/dcs.2017.18.2.411

RReceived 22 April 2017; **Revised** 27 April 2017
Accepted 28 April 2017

***Corresponding Author; Jinsul Kim**

Tel: +82-62-530-1808
E-mail: jsworld@chonnam.ac.kr

Ⅰ. Introduction

Communication technologies depend on both system software and system hardware. For example, materials have a great impact on the limitations of the system hardware, and thus, system software should be effectively developed and managed to overcome these hardware limitations [1]. Several studies have contributed to the field of media transmission in order to optimize bandwidth utilization for multimedia in a network, such as an effective bandwidth based scheduling [2] and a low transmission overhead framework [3]. Thus, an improvement in the bandwidth efficiency is necessary because it leads to better quality of service in network communication applications.

Today, the majority of traffic present on the Internet is comprised of Transmission Control Protocol (TCP) flows [4]. Standard TCP does not provide any mechanism to control the bandwidth allocated to a particular flow, and two connections with the same round-trip time generally receive an equal share of bandwidth at a particular bottleneck link. This equitable bandwidth sharing is desirable if the connections belongs to different users in a network, but it may not maximize user satisfaction if the competing flows belong to the same user. It is conceivable for a user to want to prioritize different applications and to distribute bandwidth according to his or her preferences and utilization. Bandwidth fluctuations and uncertainty cause the work of distributing bandwidth to be more difficult, and we sometimes cannot handle the distribution when applications continuously join and leave the network. Thus, we need an intelligent system that can make decisions automatically.

In this paper, we consider a problem in Local Area Network (LAN). Suppose we have three clients A, B, C in the LAN which are connected to a Switch to a Router which acts as a gateway to the internet. If User A has a lot of downloads running and User B uploads files all day and night, which leaves user A has a small amount of bandwidth since A and B already used up all bandwidth which the internet connection offers. The solution is that we need to share available bandwidth fairly among clients. Sharing resources provides high average throughput and improves utilization of the resources. A common mechanism to allocated bandwidth is to allow for weighted fair sharing of bandwidth among different users. For instance, a network manager usually divides an imbalance of bandwidth to all devices in a local network. He may decide to set aside one-fourth of the available bandwidth for a user with a video streaming application, to assign another fourth of the bandwidth for a voice application, and to allocate the remaining bandwidth for web browsing. However, some applications require a minimum guaranteed bandwidth to be allocated regardless of the current link capacity, and multimedia streaming applications are a prime example of such applications since they generally require constant playback at a particular rate and are sensitive to fluctuations in the received rate. Many online games [5] also have strict minimal bandwidth requirements to achieve adequate usability. These applications can suffer from severe performance degradation if they fail to receive a minimum desired bit rate, so it may be desirable to specify a minimum bit rate for these applications regardless of the total link capacity. However, such approach to delivering bandwidth is not flexible because a user sometimes remains in a rest state, but no other user can access the allocated resources.

To this end, fuzzy logic [6-10] is a suitable choice for such an approach by computing results based on degrees of truth rather than on the usual true or false determination. In fact, obtaining formal analytical models for fairness sharing bandwidth is difficult because we must consider many characters of network devices. Besides, some of those characters often change in the uncertainty of network communication. Additionally, human often operates using fuzzy evaluations in everyday situations. For example, when an engineer implements a system, he must consider many network connection parameters such as TCP receive window size. It is sometimes a difficult task because he must build the network model and solve non-linear optimization problems. Instead, he estimates the parameters based on his previous experience such as what he has done before. Especially, in the network communication, we always have to evaluate network metric to prevent network issues in advance.

This paper focuses on effectively improving the bandwidth for a group of users on the local physical network. Each user requires a fuzzy system with the ability to estimate the current network state, specifically whether there is a low, medium or high sending/receiving bit rate. Then, a method is designed to reduce or increase the resource consumption of a user. Our work has two main contributions. First, we propose a bandwidth-sharing system that uses fuzzy control to share the bandwidth among different users on a local network. Second, we also propose a method to decrease the bandwidth usage for streaming video.

The rest of this paper is organized as follows. In Section II, we introduce previous works related to our research. In Section III, we describe our proposed system using a Mamdani fuzzy model. In the next section, we present the fuzzy method for sharing bandwidth. Section V discusses a simulation carried out in Optimized Network Engineering Tool (OPNET) and a real-time video streaming experiment in (Web Real-Time Communication) WebRTC. Finally, the last section provides the conclusion as well as the corresponding discussion.

Ⅱ. Related Works

Network transmission technologies are constantly evolving to provide users with added functionality. The Internet handles so-called elastic traffic where the rate of flows adjusts to fill the available bandwidth [11], and it can be drawn as a network with a set of links L where each link $l \in L$. The performance of such a network has a capacity greater than zero, and the components seek to allocate link bandwidth to a set of flows in order to meet some shared objective, such as the maximum throughput, maximum-minimum fairness, proportional fairness, and weighted shares.

Fuzzy control logic [12] can be used to formulate a connection admission control system using type-2 fuzzy logic. This system can combine the input rate of real-time voice and video traffic and non-real-time data traffic to determine the connection admission combining the experience derived from many experts so that an acceptable decision boundary can be obtained. Also, it provides an interval decision from which a soft decision can be made based on the design tradeoff between the cell loss ratio and bandwidth utilization.

Data are transmitted on a packet-by-packet basis over the internet, and a packet delivered over the internet is either received correctly or is lost. Losses are mainly caused by network congestion and queuing delay. Hence, a new multimedia streaming TCP-friendly protocol that combines forward estimation of network conditions with information feedback control was proposed to optimally track the network conditions. In order that resource allocation can be carried out for multimedia streaming over the internet [13]. The authors also proposed a novel resource allocation scheme to adapt the media rate to the estimated network bandwidth using each media's rate-distortion function under various network conditions.

The authors in references [14-16] constructed a high-bandwidth overlay tree for streaming services to improve the tree bandwidth of peer-to-peer multimedia streaming if bandwidth bottlenecks occur at intermediate links in paths instead of edge links near hosts. They realized that underlay information, such as link connectivity and link bandwidth, is important in tree construction because two seemingly disjoint overlay paths may share common links on the underlay. They then proposed an approximation algorithm for Internet-like topologies to obtain high tree bandwidth and low link stress with low penalties in an end-to-end delay.

Adaptive bit rate streaming over Hypertext Transfer Protocol (HTTP) [17-20] is a technique used to stream multimedia over computer networks. It works by detecting a user's bandwidth and Central Processing Unit (CPU) capacity in real-time and adjusting the quality of the video stream accordingly. This method requires the use of an encoder that encodes a single video source at multiple bit rates, and the playback client switches between the different encodings depending on the available resources. This technology has recently been adopted for Internet video delivery and is expected to be more broadly deployed over the next few years.

Research on WebRTC [21-23] discusses some of the mechanisms used for WebRTC to handle packet loss and control loop APIs in the video communication path. Their analyses indicate that WebRTC audio and video engines dynamically adjust the bit rate of the media streams to match the conditions of the network link between the peers. The sending bit rate of a media source, such as the video frame rate or resolution in video streaming, varies continuously as the available bandwidth in the network fluctuates, and the service can even be switched off when the bandwidth does not satisfy its requirements. On the same idea of this, max-min fairness [24] allocates resource in a feasible region with constraints, and it attempts to increase the allocation of any participant necessarily which results in the decrease in the allocation of some other participant with an equal or smaller allocation. Though each participant has their characteristic, some devices design to only work efficiently in high bandwidth with powerful computation CPU and graphic processing. In the another hand, some devices only works in low bandwidth as it is compatible with the working-class who has low wages. Therefore, allocating bandwidth equally in these cases is unfairness. This kind of thought urges us to have a method, in which, a device can decide to share a resource in an inequality bandwidth allocation environment.

Ⅲ. System Overview of Proposed Method

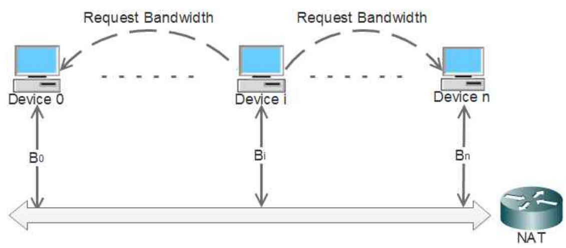


그림 1. 로컬 네트워크간의 협상방법 개요.
Fig. 1. Overview of the negotiation method in the local network.

Suppose we have N consuming bandwidth devices on a local network, and each device consumes bandwidth (upload or download speed) with corresponding values $B_i, i \in N$. For simplicity, the bandwidth will refer only to the upload speed because we usually have slower uploading than downloading bit rate. However, our proposed method can also be applied for both uploading and downloading scenarios. Devices $0,...,i-1$ and $i+1,...,n$ consume bandwidth at a given time, and i subscribes to a service. If the bandwidth available for i is less than the minimum requirements of the service, $B_{M_i}, B_{M_i} > B_i$, then i cannot establish fair service. In this case, device i sends a bandwidth request to all of the devices in the local network except itself as shown in Fig. 1. The request begins a negotiation process between the requester and receiver. It ends when either side achieves a satisfactory condition such as the requester stops sending requests or the receiver cannot share resources. To assist the system in working as described above, we implement a fuzzy system for each device that estimates the device's requirements and sharing capacity to produce a decision to request or share.

We usually allocate bandwidth utilization by using a center server, which centrally manages all metric information of local devices. It allocates resources on each device depends on an available resource and a device's needed. Instead, we solve the problem in decentralizing approach to let a device control itself, including making a bandwidth request and deciding whether share bandwidth use or not. In this way, we make a competition with constraints between devices on the local network as the real competition between employees in a company.

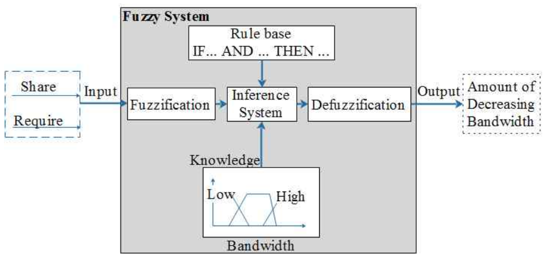


그림 2. 퍼지시스템을 이용한 대역폭 리소스 추정.
Fig. 2. Estimating bandwidth resources with a fuzzy system.

The proposed fuzzy system has two input variables consisting of the amount of resources shared and requested. The output of the system is the amount of bandwidth by which a sharing device can reduce its consumption. As shown in Fig. 2 , each device estimates the resource usage of the current state of network utilization by using fuzzy linguistics value, and this process is referred to as fuzzification. Then, human knowledge (expert knowledge judging the trends of external factors) and a set of constraints are used as rules to calculate the output, and this process is called fuzzy inference. Finally, defuzzification defuses the fuzzy values to the real sharing amounts.

Roughly speaking, the system analyzes a device's current required or shared bandwidth and then determines the best estimation request or sharing scheme for the device. Put in another way, the implementation procedures used in this study include the following items:

1) Establishing equations to estimate resources and decide whether to share or request bandwidth.

2) Setting up the fuzzy rules between the sharing and requesting devices.

3) Establishing algorithms to evaluate the required bandwidth for each device according to the current usage and the fuzzy inference rules.

4) Establishing a method to decrease/share bandwidth.

Ⅳ. Fuzzy Method for Sharing Bandwidth

On the basis of the proposed system, the amount of required and shared resources are expressed as vague estimations. In this approach, we use triangular fuzzy numbers to characterize the amount of bandwidth of the requesting or sharing device.

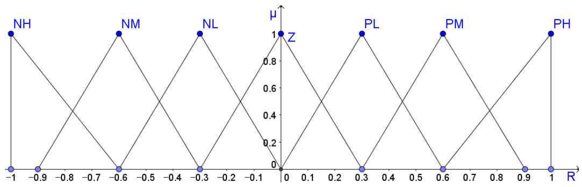


그림 3. 삼각형 퍼지 수의 입력 멤버쉽 함수 (각 퍼지 수 삼각형 상단에 표시됩니다).
Fig. 3. Input membership function of triangular fuzzy numbers (Each fuzzy number is given at the top of a triangle).

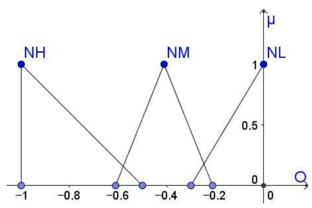


그림 4. 삼각형 퍼지 수의 출력 멤버쉽 함수 (각 퍼지 수 삼각형 상단에 표시됩니다).
Fig. 4. Output membership function of triangular numbers (Each fuzzy number is given at the top of a triangle).

A triangular fuzzy number is a particular case of fuzzy sets in a Mamdani fuzzy model. It has a triangle-shaped membership function that can be viewed as a probability distribution [9]. Suppose that we have three points $q=(a_1, a_2, a_3), a_1 \le a_2 \le a_3 \in \mathbb{R}$. Then, q is a triangular fuzzy number with membership function $\mu_q(x) \in [0,1]$.

The vague values are estimated by using the linguistic terms to evaluate the amount of bandwidth required and shared for each device. Thus, we define a symbol as positive or negative according to the increase or decrease in bandwidth utilization. Seven linguistic sets are allowable to describe the variables of one's subjective judgment: (1) Negative High, (2) Negative Low, (3) Negative, (4) Zero, (5) Positive Low, (6) Positive Medium, (7) Positive High (with abbreviations of NH, NL, Z, PL, PM, PH) in increasing order. Furthermore, the linguistic sets can be quantified with the corresponding triangular fuzzy numbers, as shown in Fig. 3 depicting the input membership functions and Fig. 4 depicting the output membership function.

4-1 Fuzzification Input

We name B_S is the current sending bit rate of a device that can share bandwidth. B_R is the current sending bit rate of a device that intends to subscribe to a specific service but cannot reach the service due to low bandwidth because almost all bandwidth has been consumed by other devices. We also name B_M as the minimum requirement for the bandwidth to satisfy a specific service requirement ($B_R < B_M < B_S$). Finally, we assign $\sigma_S \sigma_R$ as input values of sharing and request which correspond to fuzzy variables $q_S q_R$.

By comparing the current sending speed with the minimum requirement, we form scalar values of the input system by using the two equations below.

$$\sigma_S = \frac{B_M - B_S}{B_R}. \quad (1)$$

$$\sigma_R = \frac{B_M - B_R}{B_R}. \quad (2)$$

In our assumption, the requester only makes a request when its bandwidth use is lower than the utilization of the receiver or a certain amount of use, such as an average of bandwidth use, a minimum requirement of requester subscribing service. In contrast, the receiver only processes a request if its use is greater than the required. So that, sharing in (1) and requiring in (2) has range $\sigma_S \sigma_R \in [-1,1]$, and the negative value in (1) indicates that usage can be reduced while the positive value in (2) indicates that resources need to be negotiated with others and bandwidth cannot be shared with other devices on the network and vice versa. The input value in fuzzy system is represented using fuzzy numbers, as shown in Fig. 3.

4-2 Fuzzy Inference

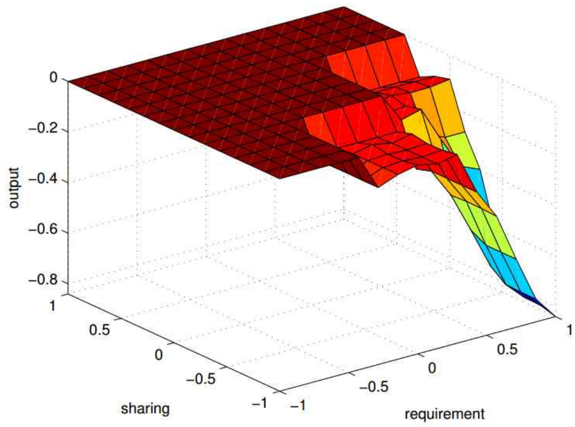


그림 5. 퍼지 시스템의 입/출력 사이의 관계.
Fig. 5. Relationship between input and output of the fuzzy system.

표 1. 대역폭 퍼지 추론.
Table. 1. Fuzzy inference of bandwidth.

q_R/q_S	NH	NM	NL	Z	PL	PM	PH
NH	Z	Z	Z	Z	Z	Z	Z
NM	Z	Z	Z	Z	Z	Z	Z
NL	Z	Z	Z	Z	Z	Z	Z
Z	NL	NL	Z	Z	Z	Z	Z
PL	NL	NL	Z	Z	Z	Z	Z
PM	NM	NM	NL	Z	Z	Z	Z
PH	NH	NH	NM	Z	Z	Z	Z

As mentioned above, we have two input values σ_R and σ_S in the fuzzy system. The output value of the system σ_O is the amount of bandwidth that a device can reduce to share its bandwidth usage. The output value corresponds to fuzzy variable q_O. Therefore, we have a set of fuzzy rules as follows,

If q_R is PH and q_S is NH, then q_O is NH.

If the requiring is Positive High, this means it highly needs resources. If the sharing is Negative High, this means it is very willing to share or decrease bandwidth utilization. Finally, when the output is Negative High, this means the sharing device will reduce utilization at a high rate. We probably do nothing if σ_R is negative because we do not want to decrease the bandwidth of the requesting device when it is the requester. Also, in the case where σ_S is positive, the sharing device needs more bandwidth and it cannot share bandwidth with others. The output q_O is Z describes keeping the device in the current state. Therefore, we only mention rules which are not Z. It includes ten rules which differ from Z, and two special Z rules are highlighted in the bold words because it is in the region of positive q_R and negative q_S. The discussion of the rules between input and output is shown in Table 1.

4-3 Defuzification Output System

On the basic of human knowledge, the triangular-shaped output membership function in Fig. 4, and fuzzy rules in Table. 1, the system calculates output value σ_O using the centroid method, as it corresponds to fuzzy output variable q_O. After several steps, we have a crisp value $\sigma_O \in [-1,0]$ of the output system, but it is a standardized value. Thus, we use (3) to calculate the real sharing output amount B_D of the system.

$$B_D = (-1)\sigma_O(B_S - B_M). \quad (3)$$

The max-min inference method is formed as:

$$\mu_O^i = \max_k\left[\min_j\left[\mu_S^{j,k}, \mu_R^{j,k}\right]\right], \quad (4)$$

where μ_O^i is the value of an output membership function μ_O of i^{th} device, $\mu_S^{j,k}$ is an input value S at k^{th} rule, and $\mu_R^{j,k}$ is an input value R at k^{th} rule. $k,j \in \{1,2,3,\ldots,49\}$ is the fuzzy rule's index as shown in Table 1, and $i \in [1,\ldots,N]$ with N is the number of devices in a network. The relationship between the system input and output is shown in Fig. 5. It shows that fuzzy output σ_O is only smaller than 0 if a fuzzy input sharing is negative and fuzzy input requiring is positive.

Ⅴ. Experiment with Video Streaming

We can apply the proposed method to any networks where it has a limited throughput. For example, a computer usually has one or two network interfaces though it always has a limitation on sending and receiving bit-rate. Meanwhile, many running applications in the computer compete with other to have the bandwidth. Besides, we can use various of network applications, which send and receive data over the network, to demonstrate the efficiency of our method. However, we chose video streaming as a simulated method because it is sensitive to network condition. A small change to a network immediately affects the quality of the streaming.

5-1 Fuzzy Method for Video Streaming Experiment

Suppose that x is the total horizontal size of the active pixels, y is the total vertical size of active pixels, r is the video frame rate or refresh rate, and $\alpha \in R^+$ is a factor that describes an original sending bit rate affected by environmental network factors. Thus, the sending bit rate is formed in (4).

$$f(x,y,r) = \alpha xyr. \quad (5)$$

In the sharing device, from an original point of resolution and frame rate (x_0, y_0, r_0), we reduce to a new point (x_i, y_i, r_i) in several steps, which is referred to as a decreasing process. From (3) and (5), we yield $B_D = \alpha(x_0 y_0 r_0 - x_i y_i r_i)$, then we divide the decreasing process into two cases as follows.

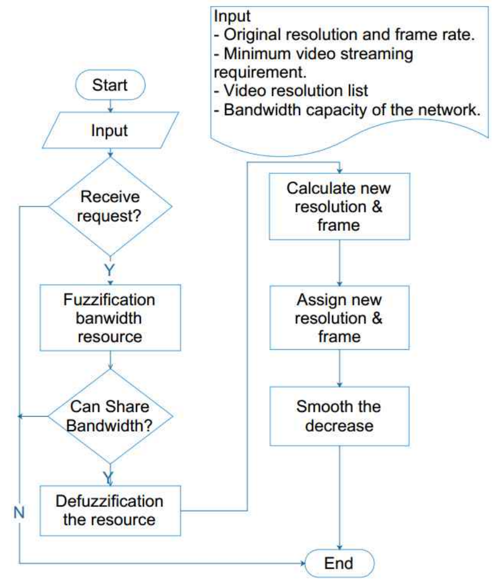


그림 6. 공유장치의 대역폭 감소 과정.
Fig. 6. Reduction bandwidth process of sharing device.

First, we keep resolution $x_i = x_0, y_i = y_0$, and reduce the frame rate until it satisfies (6), with $[a] = \min\{k \in Z, |k| \le |a|\}$, f and F as a minimum and maximum frame rate requirement for video streaming, respectively. Second, if we cannot find the appropriate frame rate, then we seek to find the next resolution in the resolution list that has a lower rate and also satisfies (6).

$$f \le r_i = \left\lfloor \frac{\alpha x_0 y_0 r_0 - \beta B_D}{\alpha x_i y_i} \right\rfloor \le F, \quad (6)$$

where $\lfloor . \rfloor$ is the floor function. After the sharing devices decrease their bandwidth use, the requiring has an amount βB_D bps where $\beta \in [0,1]$ is a factor with which the environmental network conditions have an effect on the reducing amount. If

$\beta\sum_i B_{Di} + B_R > B_M$ which means that the requester satisfies the subscribing service requirement with B_{Di} as the amount of decreasing bandwidth at i^{th} sharing device. We seek to find a new appropriate rate for the requesting device.

Second, we keep the video frame rate and find an instance in the resolution list (x_i, y_i) that satisfies (7). If we cannot find the appropriate resolution, then we increase the video frame rate that also satisfies (7).

$$f \le r_i = \left\lfloor \frac{\beta\sum_i B_{Di} + B_R}{\alpha x_i y_i} \right\rfloor \le F. \qquad (7)$$

If we directly decrease the rate from (x_0, y_0, r_0) to (x_i, y_i, r_i), then the user will experience inconvenience. Thus, we must smooth the decrease by adding more than one intermediate decreasing step. Suppose that (x_j, y_j, r_j) is an intermediate step between step 0 and step i.

S_D is assigned as the number of intermediate steps. Since the resolution has a higher influence on quality during streaming than the frame rate, the total intermediate steps in the sharing devices is calculated by (8). If (x_j, y_j) is unchanged and r_j is given by (9) and satisfies (6), where $r_0 > r_i$.

$$p = S_{D+}\left(\frac{[r_0 - r_i]}{5} + 1\right). \qquad (8)$$

$$r_i = \left\lfloor \frac{r_0 - r_i}{p} \right\rfloor + r_0. \qquad (9)$$

In (8), we yield $p \ge 1$ even if $S_D = 0$ and $r_0 = r_{i+1}$ or $r_0 = r_{i-1}$. Thus, given time t, each step must take $\Delta = t/p$ seconds.

Finally, the negotiation process to gain the resource is described in Fig. 6 with following steps:

Step 1: Each device determines its current state in the network, such as the current resolution for video streaming, frame rate, minimum bandwidth requirement of video streaming, and network bandwidth capacity.

Step 2: According to the current streaming state, the device decides whether to make a resource request or not.

Step 3: Sharing device receives a requested resource and carries out the fuzzification process to estimate the amount of resources shared/required.

Step 4: The sharing device conducts defuzzification to estimate the decreasing utilization bit rate.

Step 5: New resolution and frame rate are calculated.

Step 6: If the requesting device can achieve a positive streaming video quality after receiving responses from the sharing devices, then we move to the next step. Otherwise, the requesting device stops making resource request.

Step 7: Assigning new resolution and frame rate to prepare step 8.

Step 8: At the sharing device, we conduct a smooth decreasing process at step 9. At the requesting device, we do not have any further action.

Step 9: Comparing the new value and the initial value of the video streaming rate to produce the decreasing process given in (8) and (9).

The requiring device does not have any fuzzy calculations. It receives a response from the sharing device with an amount of sharing bandwidth. After that, it finds the matched video resolution to the given shared bandwidth without calculating of currently available bandwidth once again.

5-2 Simulation Video Streaming in OPNET

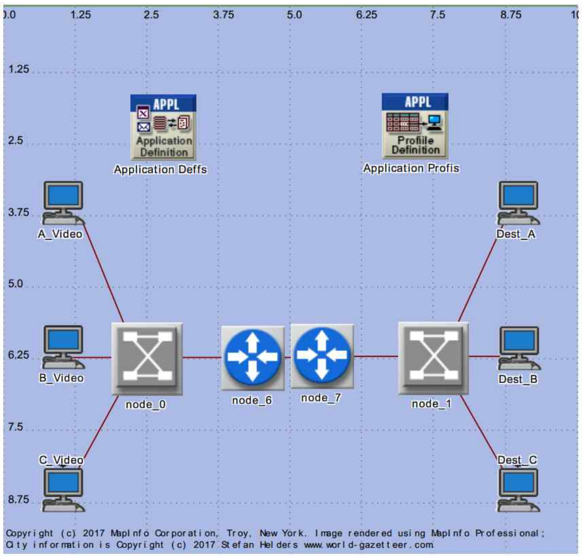


그림 7. 로컬 네트워크에 3개의 장치가 있는 OPNET의 비디오 스트리밍 시뮬레이션.

Fig. 7. Video streaming simulation in OPNET with three devices in the local network.

For the simulation, we use a common network simulation tool to describe how the proposed method works. The simulation is conducted as shown in Fig. 7 with three devices having a video conference from a local physical network to three other devices in another local network without the use of any encoding and decoding method in the video transmission. In each device, we implement the fuzzy system that assists them in negotiating resources by sending the requested resource to other. On the basic of the estimation, the sharing devices reduce the resolution in a fixed resolution list or it reduces the frame rate to the given minimum value. Similarly, the requesting devices increases its video streaming resolution or frame rate depending on the decrease of the sharing devices. In the simulation, we increase the minimum requirement for video streaming of the requesting device after it gets a certain streaming resolution. We stop the increasing when the sharing cannot share bandwidth.

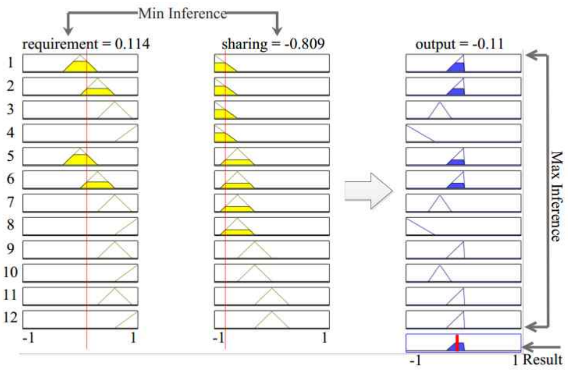


그림 8. 12개의 규칙, 2개의 입력, 1개의 출력으로 이루어진 퍼지 시스템 그림.
Fig. 8. Depiction of the fuzzy system with twelve rules, two inputs and one output.

표 2. OPNET 스트리밍 실험에서의 비디오 스트리밍 목록.
Table. 2. List of video streaming resolutions in the OPNET streaming experiment.

	Horizontal/Vertical/Pixels				Horizontal/Vertical/Pixels		
1	128	128	16384	6	432	128	55296
2	160	144	23040	7	320	200	64000
3	224	144	32256	8	320	224	71680
4	256	192	49152	9	320	256	81920
5	280	192	53760	10	400	240	96000

표 3. H:수평 픽셀, V:수직 픽셀, F:프레임속도 이 3가지 장치에서의 협상 프로세스의 해상도 감소 결과.
Table. 3. Result of decreasing the resolution in the negotiation process with three devices where H: Horizontal pixels, V: Vertical pixels, and F: Frame rate.

Step	Stream A (H/V/F)			Stream A (H/V/F)			Stream A (H/V/F)		
0	400	240	30	320	224	35	0	0	0
-	-	-	-	320	224	33	-	-	-
1	400	240	27	320	224	31	160	144	24
2	400	240	25	320	224	29	160	144	38
3	400	240	24	320	224	28	256	192	21
4	400	240	23	320	224	27	256	192	24
5	400	240	22	320	224	26	256	192	27

We use the list of video resolutions given in Table 2 with the minimum refresh rate requirement of $f=20$ and maximum refresh rate of $F=40$, which means that the video frame rate cannot decrease below the minimum or increase above the maximum value in any reducing/increasing rate process. To reduce the complexity, we chose $\alpha=1, \beta=1$. Both source and destination have no encoder and decoder, and the bandwidth is set to 8Mbytes.

Fig. 8 depicts the results of the defuzification process of the fuzzy system in Matlab using the fuzzy centroid method with the requiring/sharing input, and the output of the system.

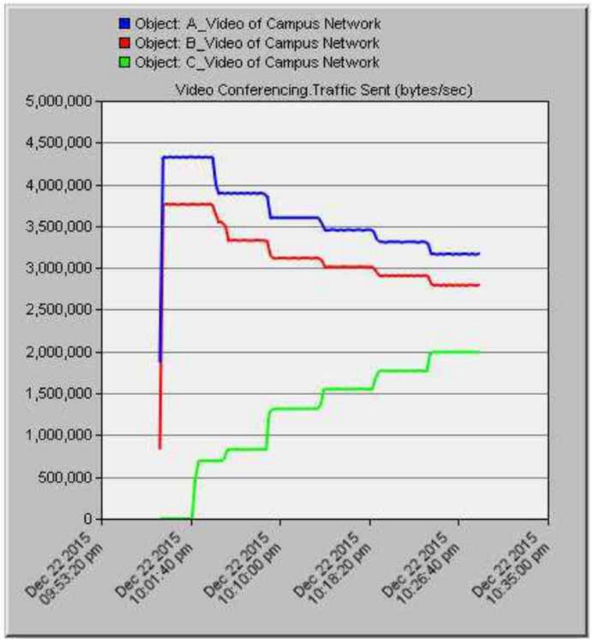


그림 9. 세 장치에서 전송 비트 전송률을 나타낸 시뮬레이션 결과.
Fig. 9. Result of the simulation with the sending bit rate in three devices.

The simulation is run within thirty minutes. After every five minutes, we increase the minimum requirement of device C. From the beginning of the simulation, device C does not have sufficient bandwidth to create a video streaming service. However, after five steps, the streaming rate increases instantly as A and B share their usage. The detailed results of the negotiation are shown in Table 3, and the results of the simulation are depicted in Fig. 9. In the result of the simulation, the intermediate step between step 0 and 1 is the result of the decreasing resolution process.

Besides, we simulated the system with some devices which vary from three to one-hundred-twenty. The system converges if all of the requesting devices stop sending a resource request to the other devices in the local network. If a current streaming resolution of the sharing device is close to its video bandwidth minimum requirement, or the minimum requirement of the requesting device is much lower than the current bandwidth usage of the sharing devices, then the requesting device just sends a few requests. Therefore, the minimum requirement of each device is necessary to speed up system convergence. So far, the proposed system works correctly in any conditions of the network because we prevent the sharing device from degrading resource. That means the requesting device cannot require the sharing device reduce the most of its resource. The sharing device cannot also decrease its resource use below its minimum bandwidth requirement.

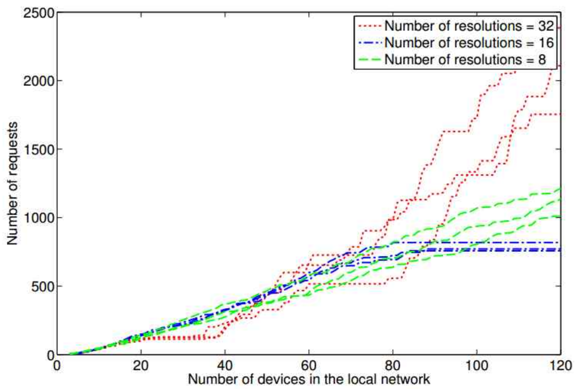
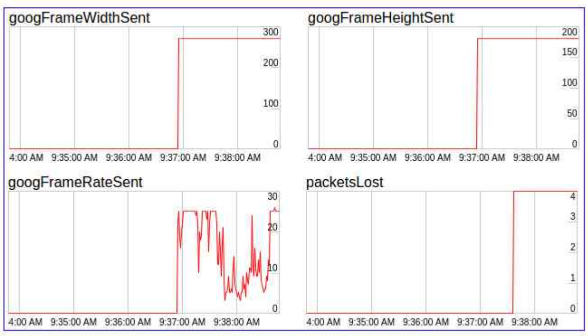


그림 10. 다수의 장치와 해상도를 기반으로 한 시스템 집합도.
Fig. 10. System convergence based on a number of devices and a number of resolutions.

(a) - Original video streaming on EasyRTC.

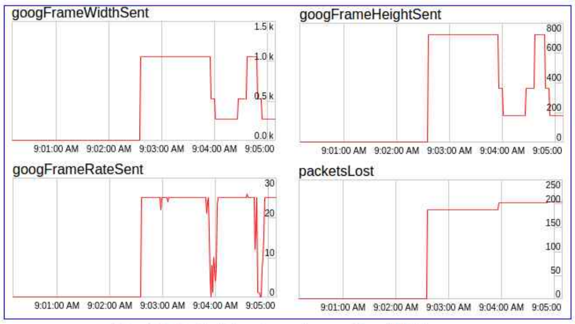


(b) - Video Streaming on EasyRTC using the proposed method.

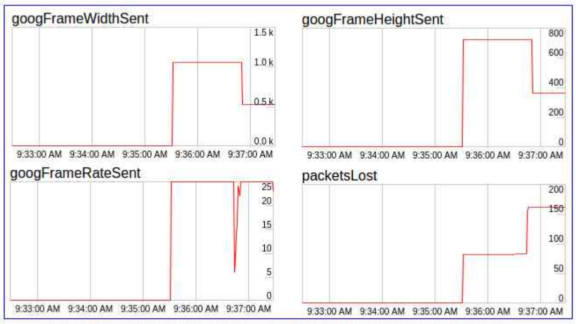
그림 11. 필요한 장치에 대한 통계.
Fig. 11. Statistics for requiring device.

Fig. 10 depicts the simulation results of the above-described tests. In which, each device has the same number of resolutions, such as eight, sixteen and thirty-two. Given the number of resolutions, we run the test three times. As a result, the number of resource requests increases along with the number of devices. Given the number of devices in the local network, the number of resource requests also increases along with the number of resolutions. However, the number of requests can decrease even though the number of resolutions increases as shown in the difference between the green line and the blue line. The reason is that the fuzzy system estimation based on the difference between a resolution of the request and sharing device. In the given resolution list, each resolution in the first eight-resolution is closer to other than the next second eight-resolution since we order thirty-two resolutions from a small spatial resolution to a high spatial resolution in the experiment. The higher spatial resolution, the more difference between two adjacent resolutions.

5-3 Video Streaming Experiment in WebRTC



(a) - Original video streaming on EasyRTC.



(b) - Video Streaming on EasyRTC using the proposed method.

그림 12. 공유 장치에 대한 통계.
Fig. 12. Statistics for sharing device.

On the basis of the theoretical background and the results of the simulation, we set up an experiment in EasyRTC [25], installed in Ubuntu 12.04. The limited uploading bandwidth of the network is set to 2Mbps using the open source Wondershaper

[26]. We use the open source WebcamStudio [27] to create a virtual webcam device for which we can control the resolution and frame rate of the video. In the result of the experiment, we compare between the current implementation of EasyRTC and the implementation of our algorithm.

The experiment conducted by two clients which run in the Chrome web browser on Ubuntu 12.04, with video streaming with a client also running in the Chrome browser on another computer on the same LAN. When the connection is established between the two clients and its destination, we access the Chrome WebRTC statistics to obtain the measurement parameters at *chrome://webrtc-internals/*. We set up the experiment as the following, one of the two clients, which acts as a sharing device, has high definition video streaming within one minute before another client starts streaming. The another client acts as a requiring device. After two clients join in the network, they compete with another for bandwidth.

Fig. 11-(a) and Fig. 12-(a) shows that video frame rate, height, and width fluctuate along with the time because the sharing device and requiring device compete with another for bandwidth in the original implementation of WebRTC. It results in increasing package lost. In contrast, Fig. 11-(b) and Fig. 12-(b) do not have those fluctuations, it also reduces package lost. At the time of changing frame rate in Fig. 12-(b), the method detects that it cannot reduce the frame rate, so that, it decreases video streaming resolution. From that time, the requiring device can start having video streaming. By using this method, we detect an appropriate initial resolution before establishing the streaming service. Thus, the method reduced the package loss and the computing resources needed on both sides of the streaming service.

Regardless of the encoding/decoding process during streaming, our proposal is similar to WebRTC, which reduces the resolution to save bandwidth with a different approach in the local physical network when each client knows each other. The clients can negotiate with others by sending a request whenever needed, which is similar to a business trade. Moreover, the involvement of the fuzzy system makes the process closer to human activity with greater precision and more flexibility than a coding system.

In summary, the results of this experiment showed the fuzzy control theorem can be used to flexibly estimate the amount of bandwidth network to share or request a given transmission bit rate. In the simulation, each sharing device gives just a small amount of bandwidth that does not affect much its quality of service, but it has a sufficient impact on the streaming video condition of the requesting device.

Ⅵ. Conclusion

In this paper, we presented a flexible method to share the bandwidth of a local network using fuzzy control theory. We used a Mamdani fuzzy model with two input parameters and output of the defuzzification process calculated using the Centroid method with twelve established rules. The triangular-shaped membership function is used to quantify the grade of the membership for the fuzzy input and output. Based on our knowledge and the condition of the system, we can change the fuzzy set to adapt to our use. With the evident input of the system, we transform these inputs with a calculation in the process of the fuzzy system. Thus, we tried to form an appropriate fuzzy input and a corresponding output that is relevant to the input. Also, we proved our formulation by analyzing the reasoning for those formulations and showed that the result of the output system is consistent with the input. The results of the experiment showed that our method reduced the package lost as well as the fluctuation of video streaming resolution. Besides, each sharing device just gives a small part of its bandwidth without having much effect on its quality of service, but the total decreasing amount is higher if we have many devices involved.

In future works, we will extend the work on a protocol sharing between devices such as a computer, mobile phone, printer, set-top-box, and many more devices on the same physical network.

Acknowledgments

This work was supported by 'The Cross-Ministry Giga KOREA Project' grant from the Ministry of Science, ICT and Future Planning, Rep. of Korea(GK16P0100, Development of Tele Experience Service SW Platform based on Giga Media). Also, this research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2017-2016-0-00314) supervised by the IITP(Institute for Information & communications Technology Promotion).

Reference

[1] H. Kim and N. Feamster, “Improving network management with software defined networking,” *IEEE Communications Magazine*, Vol.51, No.2, pp.114–119, February 2013.

[2] S.H. Kang and A. Zakhor, “Effective bandwidth based

scheduling for streaming media," *IEEE transactions on Multimedia*, Vol.7, No.6, pp.1139–1148, November 2005.

[3] H. Qi, M. Stojmenovic, K. Li, Z. Li, and W. Qu, "A low transmission overhead framework of mobile visual search based on vocabulary decomposition," *IEEE Transactions on Multimedia*, Vol.16, No.7, pp.1963–1972, August 2014.

[4] P. Mehra, A. Zakhor, and C. De Vleeschouwer, "Receiver-driven bandwidth sharing for tcp," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE*, San Francisco, California, USA, pp.1145–1155, 2003.

[5] C.Y. Huang, K.T. Chen, D.Y. Chen, H.J. Hsu, and C.H. Hsu, "Gaminganywhere: The first open source cloud gaming system," A*CM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, Vol.10, No.1s, p.10, January 2014.

[6] R.R. Yager and L.A. Zadeh, *An introduction to fuzzy logic applications in intelligent systems*, Springer Science & Business Media, 2012.

[7] R. Babuška, *Fuzzy modeling for control*, Springer Science & Business Media, 2012.

[8] E. Czogala and J. Leski, *Fuzzy and neuro-fuzzy intelligent systems*, Physica, 2012.

[9] H. Hellendoorn and D. Driankov, *Fuzzy model identification: selected approaches*, Springer Science & Business Media, 2012.

[10] R. Palm, D. Driankov, and H. Hellendoorn, *Model based fuzzy control: fuzzy gain schedulers and sliding mode fuzzy controllers*, Springer Science & Business Media, 2013.

[11] A. Johnston, J. Yoakum, and K. Singh, "Taking on webrtc in an enterprise," *IEEE Communications Magazine*, Vol.51, No.4, pp.48– 54, April 2013.

[12] Q. Liang, N.N. Karnik, and J.M. Mendel, "Connection admission control in atm networks using survey-based type-2 fuzzy logic systems," *IEEE Transactions on Systems, Man, and Cybernetics*, Part C (Applications and Reviews), Vol.30, No.3, pp.329–339, August 2000.

[13] Q. Zhang, W. Zhu, and Y.Q. Zhang, "Resource allocation for multimedia streaming over the internet," *IEEE Transactions on Multimedia*, Vol.3, No.3, pp.339–355, September 2001.

[14] Y.h. Chu, S.G. Rao, S. Seshan, and H. Zhang, "A case for end system multicast," *IEEE Journal on selected areas in communications*, Vol.20, No.8, pp.1456–1471, October 2002.

[15] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, Scalable application layer multicast, in S*IGCOMM '02 Proceedings of the 2002*, Pittsburgh, Pennsylvania, USA, pp.205-217, 2002.

[16] X. Jin, W.P.K. Yiu, S.H.G. Chan, and Y. Wang, "On maximizing tree bandwidth for topology-aware peer-to-peer streaming," *IEEE Transactions on Multimedia*, Vol.9, No.8, pp.1580–1592, December 2007.

[17] S. Akhshabi, A.C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http," in *Proceedings of the second annual ACM conference on Multimedia systems*, San Jose, CA, USA, pp.157–168, 2011.

[18] S. Lederer, C. Müller, and C. Timmerer, "Dynamic adaptive streaming over http dataset," in *Proceedings of the 3rd Multimedia Systems Conference*, Chapel Hill, North Carolina, pp.89–94, 2012.

[19] O. Oyman and S. Singh, "Quality of experience for http adaptive streaming services," *IEEE Communications Magazine*, Vol.50, No.4, April 2012.

[20] T. Stockhammer, "Dynamic adaptive streaming over http–: standards and design principles," in *Proceedings of the second annual ACM conference on Multimedia systems*, San Jose, CA, USA, pp.133–144, 2011.

[21] Linh VM, Jang JH, Kim J, "Adjusting Local Network Speed by Using Fuzzy Theory with An Illustration in WebRTC Environment," *The Journal of Digital Contents Society*, Vol. 16, No. 6, pp. 917-925, December 2015.

[22] Linh VM, S. Park, J. Jang, J. Park, J. Kim, "A Dynamical Load Balancing Method for Data Streaming and User Request in WebRTC Environment," *The Journal of Digital Contents Society*, Vol. 17, No. 6, pp. 581-592, December 2016.

[23] A.B. Johnston and D.C. Burnett, WebRTC: APIs and RTCWEB protocols of the HTML5 real-time web, Digital Codex LLC, 2012.

[24] B. Radunović and J.Y.L. Boudec, "A unified framework for max-min and min-max fairness with applications," *IEEE/ACM Transactions on Networking (TON)*, Vol.15, No.5, pp.1073–1083, October 2007.

[25] Priologic Software Inc. Easyrtc open source. Available: https://github.com/priologic/easyrtc/.

[26] V. Mulhollon, Wondershaper: Simple network traffic shaping tool for ubuntu linux. vlm@debian.org, 2004.

[27] W. Team, Webcamstudio for gnu/linux. soylent.tv@gmail.com, 2014.

Linh Van Ma

2013 : Hanoi University of Science and Technology, Vietnam (B.S. Degree).

2013~2015: Researcher and Developer at Samsung Vietnam Mobile R&D Center - SVMC.

2015~now : School of Electronics and Computer Engineering - Smart & Media Computing Laboratory.

※Research Interests : Mobile Cloud Computing, Next Generation of Mobile Platform, Mobile Operating System and Peer-toPeer Network

박상현 (Sanghyun Park)

2010 : Computer and Information from the University of Korea Nazarene.

2012 : Chonnam National University, School of Electronic and Computer Engineering(M.S Degree)

2010~2012 : Engineer in System Development Team of Media Flow Company.

2014~now : Doctoral candidate in Electronics and Computer Engineering at Chonnam National University.

※Research Interests : Interactive Media, Systems Development, Embedded of systems, Digital Media and Cloud computing.

장종현 (Jong-hyun Jang)

1988 : Kyungpook National University of Electronic Engineering(B.S Degree).

2000 : Chungnam National University of Computer Science (M.S Degree)

2004 : Hankuk University of Foreign Studies of Information and Communication (Ph.D Degree).

1988~1994: Assistant Manager in Daewoo Telecom Co., Ltd.

1994~now: Researcher in Real and Emotional Sense Platform Research Section, Electronics and Telecommunications Research Institute (ETRI).

※Research Interests : Real-time Middleware for Telecommunication Systems, Home Networking Systems, and Real-Sense Media Services.

박재형 (Jaehyung Park)

1991 : Yonsei University (B.S. Degree).

1993 : Korea Advanced Institute of Science and Technology (KAIST) (M.S Degree).

1997 : Korea Advanced Institute of Science and Technology (KAIST) (Ph.D Degree).

1997~1998: Researcher in Center for Artificial Intelligence Research(CAIR), KAIST.

1998~2002: Senior Researcher in Network Laboratory, Electronics and Telecommunications Research Institute (ETRI).

2002~now : Professor in Chonnam National University, Gwangju, Korea.

※Research Interests : Internet Protocols, Multicast Routing, Wireless Mesh/Ad-hoc Networks, and Network Security/Cryptography.

김진술 (Jinsul Kim)

2001 : University of Utah, Salt Lake City, Utah, USA (B.S. Degree).

2005 : Korea Advanced Institute of Science and Technology (KAIST) (M.S Degree).

2008 : Korea Advanced Institute of Science and Technology (KAIST) (Ph.D Degree).

2005~2008: Researcher in IPTV Infrastructure Technology Research Laboratory, Broadcasting/Telecommunications Convergence Research Division, Electronics and Telecommunications Research Institute (ETRI).

2009~2011: Professor in Korea Nazarene University, Chon-an, Korea.

2011~now: Professor in Chonnam National University, Gwangju, Korea.

※Research Interests : QoS/QoE, Measurement/ Management, IPTV, Mobile IPTV, Smart TV, Multimedia Communication and Digital Media Arts.

http://dx.doi.org/10.9728/dcs.2017.18.2.411