A Flexible Method for Sharing Network Bandwidth Utilization in Local Network by Applying the Advantage of Fuzzy Control: An Example on WebRTC Streaming

Linh Van Ma[1], Sanghyun Park[2], Jinsul Kim[3]
School of Electronics and Computer Engineering, Chonnam National University, 77 Yongbong-ro, Buk-gu, Gwangju 500-757, Republic of Korea
linh.mavan@gmail.com[1], sanghyun079@gmail.com[2], jsworld@jnu.ac.kr[3]

Abstract - *Limitation of resources and uncertainty of environment in the network communication field, leading us to a problem of how to use the most effective utilization of bandwidth. This research, therefore, focuses on a work of sharing and balancing bandwidth in a local network, some prior users subscribe its service first and consume much bandwidth capacity, it makes new coming users cannot gain its service because it does not have enough requiring amount of the service bandwidth. Thus, we try to distribute the bandwidth of the users who have much utilization, to the new coming, by that way we make a proportional usage with a negotiating resource process between users. In order to do that, we use fuzzy logic control method and set it up to each user, in which each one estimates its network state, and sends a bandwidth request to others, or shares its consumption to others by using methods of decreasing bandwidth usage. In this paper, we design a fuzzy system which uses some equations to calculate input, output system parameters; we also prove that the equations work well in all conditions of the network. In the experiment, we simulate to show how our method works by using OPNET with the fuzzy system in each user. A comparing result in an experiment between our method and original implementation of WebRTC, which is one of the most recent real-time communication technologies, shows that this approach improves video streaming between peers efficiency.*

Keywords: Low bandwidth, Sharing bandwidth, Fuzzy control logic, WebRTC real time communication, Reduce frame rate, Reduce resolution

1 Introduction

Along with the development of the Internet, bandwidth has a vital role in providing and qualifying network quality. Recently, streaming media over the Internet presents many challenges. On one hand, the current network provides fixed up and down bit stream for a group of local users, but they always demand a high quality of serving services with various kinds of requirements such as big data processing, parallel processing, HD streaming video services. On the other hand, the network has its own problem with inherent infrastructure which causes the work of changing the structure of the network is not easy with the involved of many components, otherwise, we must improve network management [1, 2]. Many previous studies have been conducted in the field of media transmission for optimizing bandwidth usage with streaming media, low transmission [3, 4]. Thus, improving efficient utilization of bandwidth is necessary. The more effective bandwidth uses, the better quality service is, in network communication application. For the above reason, this paper concentrates on improving bandwidth utilization for a group of users in the local network where each user knows other. Each user needs a fuzzy system which has an ability to estimate its current network state whether it has low, medium or high speed. Then the system designs a method to reduce or increase resource consumption. To form the input system, we compare values between current usage and service requirement. However, the process to produce the fuzzy output system depends on services which user subscribes to. Hence, we choose WebRTC video streaming service [5, 6, 7] to demonstrate and illustrate our research in the most straightforward way. Our work is primarily focused on network transmissions, such as file transfer and multimedia streaming with two main contributions. First, we proposed bandwidth sharing system, which uses the fuzzy control system to share bandwidth between different users on the local network according to requesting bandwidth users. The second contribution is a smoothing method for decreasing bandwidth usage, such as frame rate and resolution in the case of video streaming.

The rest of this paper is organized as follows. In Section II, related works on the packet scheduling problem and some other related fields, are introduced. In Section III, we describe our proposal system using Mandani fuzzy model, which includes a description of sharing and requiring bandwidth estimation with fuzzification and defuzification processes. In Section IV, we present the way of how our method works with the simulation of OPNET - a tool that provides performance management for computer networks and applications. Section V provides conclusions and discussions.

2 Related research

Making a plan to distribute bandwidth is an important task in a large computing system, based on [8], a decentralized, accurate, and low-cost system that predicts pairwise bandwidth between hosts [9] proposed an algorithm which constructs a distributed tree that embeds bandwidth measurements without any centralized component requirement. By that way, they determine the performance of distributed computing applications in a computing system.

Network coding brings substantial improvements in terms of throughput and delay in collaborative media streaming applications. To overcome this critical redundant transmission problem [10] addressed the problem of finding a suitable asynchronous packet scheduling policy in collaborative media streaming applications.

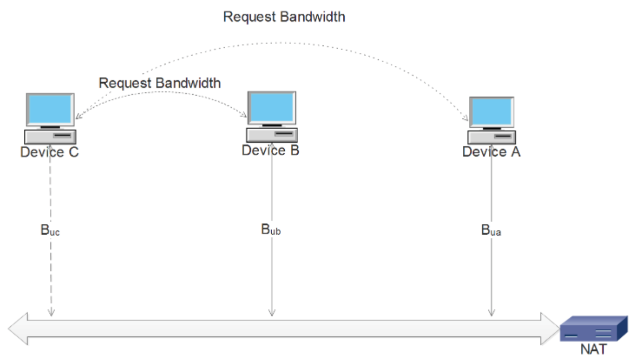

Fig.1. A local network with negotiating bandwidth proposal method.

In the field of fuzzy control logic [11] presented a connection admission control method that uses a type-2 fuzzy logic system. The system can combine the input rate of real-time voice, video traffic, and non-real-time data traffic in the decision of connection admission combine the experiences from lots of experts so that an acceptable decision boundary can be obtained. Also, it provides an interval decision, so that a soft-decision can be made based on a design tradeoff between cell loss ratio and bandwidth utilization.

3 Proposal system overview

In the proposal system has three devices A, B, and C in a local network, each device consumes bandwidth with value represents B_{ua}, B_{ub}, B_{uc} correspondingly. At first, device A and B are active at some services and they take much of bandwidth resource in the network, which describes by $B_{ua} \approx B_{ub} \gg B_{uc}$. For the above reason, device C cannot get a desired service because it does not have enough bandwidth for gaining the service. Hence, device C will negotiate with A and B for getting network resources by sending a request as shown in a dashed arrow Fig. 1. For better explanatory our research, we do not mention any case of complexity in the network infrastructure, and outside affection of the network.

3.1 Fuzzification input system

The proposal uses Mandani Fuzzy Model, and triangle shapes membership function. It has two input variables which are sharing ability and resource requirement of a client. The output variable of the system is the amount of bandwidth that user can reduce. Our method focuses on the client itself, thus, each client has their system for calculating and requesting resources. Depending on the current state of network utilization, each client estimates its fuzzy usage with a value; the process is so called Fuzzification. Then, the human knowledge and constraint rules involve for calculating output; the process is called Fuzzy Inference. Finally, the Defuzzification defuses output fuzzy values, from that, the system makes a decision decreasing bandwidth.

Assuming that, B_S is a current utilization bandwidth of a prior device which established a connection with some services and was consuming the bandwidth of the local network. B_R is a current utilization bandwidth of a device which tends to be subscribed to a specific service, however, it cannot reach the service because the bandwidth almost consumed by the device has B_S. Furthermore, we name B_M is a minimum required bandwidth to gain a specific service in the user. Thus, different users have different B_M values of their subscribed service.

By comparing the current consuming bandwidth value with minimum requirement, we form scalar values for the input system by using two equations below:

$$S = \frac{B_M - B_S}{B_S} \times 100. \qquad (1)$$

$$R = \frac{B_M - B_R}{B_M} \times 100. \qquad (2)$$

The sharing variable in (1) and requiring variable in (2) has range [-100, 100], the negative value in (1) means that it can reduce bandwidth usage, the positive value in (2) means that it need more bandwidth, and it cannot share bandwidth to other devices on the network and vice versa. In (2), if the users already established services and consumed most of the bandwidth in the network, then the device which tends to establish a new connection has a small bandwidth consumption when compares with minimum requirement, $B_R \ll B_M \Rightarrow R \approx 100$, it makes the sharing users reduce much more resource, and the serving service become low. In this case, we make a new equation which supports the user gains a service by making a comparison with B_M and B_S, and equation (2) becomes:

$$R = \frac{B_M}{B_S} \times 100. \qquad (3)$$

3.2 Fuzzy inference process

As mention above, we define the symbol as positive or negative in accordance with the increasing or decreasing bandwidth utilization. The input variables are the requirement and sharing ability, the output variable is the amount of bandwidth that the sharing user needs to decrease bandwidth usage. The fuzzy values of the input and output variables describe as Negative High, Negative Low, Negative, Medium, Normal, Positive Low, Positive Medium, Positive High or their abbreviation words are NH, NL, N, PL, PM, PH by increasing order.

In the system, we have two input variables, *R* represents the fuzzy value of requiring bandwidth, and *S* represents the fuzzy value of sharing ability. The output of the system *Op* is an amount of a device which can reduce bandwidth usage depends on the request from another device. Therefore, we have a set of fuzzy rules as follows,

If R is PH and S is NH, then Op is NH

If the requiring is positive high (it needs much bandwidth), and the sharing is negative high (it is eager to share resource), then the output is negative high (the sharing user reduces much of its utilization). Probably, we do nothing if R is kind of Negative because we do not want to decrease the bandwidth of the required device, meanwhile, it is requesting. Also, in the case S is kind of Positive which means the sharing devices need more bandwidth, they cannot share bandwidth with others. Thus, the output is N (keeping its current state). The discussion of those rules between input and output is shown in Table I.

Table 1. Fuzzy Inference of bandwidth

R/S	NH	NM	NL	N	PL	PM	PH
NH	N	N	N	N	N	N	N
NM	N	N	N	N	N	N	N
NL	N	N	N	N	N	N	N
N	NL	NL	N	N	N	N	N
PL	NL	NL	N	N	N	N	N
PM	NM	NM	NL	NL	N	N	N
PH	NH	NH	NM	NL	N	N	N

3.3 Defuzification output system

Based on human knowledge, triangle shapes membership function, and fuzzy rules which were given above, the system calculates output variable $\mu_O \in [-1, 1]$ by using the centroid method. Finally, we have the crisp value of the output system but it is still fuzzy value. Thus, we use (5) to calculate a real output amount *Dec*, of the system.

$$\mathrm{Dec} = (\mathrm{B_S} - B_M) \times \mu_O. \qquad (5)$$

The max-min inference method produces output μ_O variable, is formed as:

$$\mu_O^i = max\left[\min\left[\mu_S^i, \mu_R^i\right]\right]. \qquad (6)$$

With $\mu_O^i, \mu_R^i, \mu_S^i$ is values of output membership function corresponding to input value i^{th}, R and S. The relationship between input and output is shown in Fig. 2.

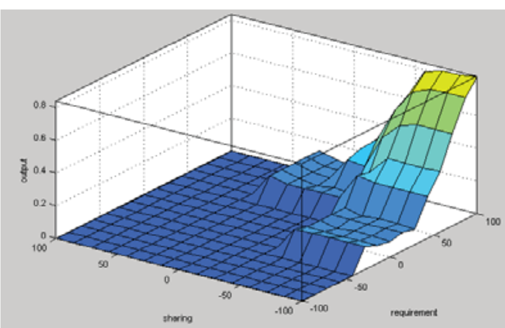

Fig. 2. Relationship between input and output of the fuzzy system

If the user does not have priorities in consuming bandwidth which means all users have equality to take resources, then we reform equation (1), (2) with Avrg is the average bandwidth consumption of the local network. Assuming that Avrg ≥ $\mathrm{Min_r}$,

$$\mathrm{S} = \frac{Avrg - B_S}{B_S} \times 100. \qquad (7)$$

$$\mathrm{R} = \frac{Avrg - B_R}{Avrg} \times 100. \qquad (8)$$

Using the same strategy above with equation (7) and (8), we have a final balance bandwidth usage after the negotiation in the local network because each sharing and requiring user tends to come *Avrg* value.

4 An example on WebRTC

Aiming to prove that our method stands out from other methods, we choose a technology which is one of the most advanced real-time communication technologies recently, WebRTC. While our method works well in low bandwidth condition, WebRTC also works well in the similar condition, such as the WebRTC audio and video engines work together with the underlying network transport to probe the available bandwidth and optimize delivery of the media streams [6]. However, DataChannel in WebRTC API transfers require additional application logic: the application must monitor the amount of buffered data and be ready to adjust as needed. Thus, it takes some time to evaluate and may cause package loss. In this section, we discuss an analysis of a negotiating algorithm, after that, we simulate our method for balancing bandwidth utilization in OPNET and illustrate the result by using open source EasyRTC [12].

A sending rate of video streaming with *r* represents frame rate or refresh rate, *y* represents total vertical lines of active pixels, *x* represents total horizontal of active pixels, and $\alpha \in R^+$ is a factor when the original rate is affected by network environment, it calculates by equation (9),

$$\mathrm{f(x, y, r)} = \alpha \times xyr. \qquad (9)$$

The strategy for reducing bandwidth utilization is that we reduce the frame rate until it gets minimum value Min_r (Max_r is the maximum frame rate), and then reduce the resolution. However, in the real world, the resolution of the digital camera has a discrete type, and the total horizontal, vertical pixel are somehow fixed in a resolution list corresponding to the type of camera, we cannot choose it randomly.

In the sharing device, from original resolution and frame rate (x_0, y_0, r_0), we reduce the rate to a new point (x_i, y_i, r_i) with several steps which is called smoothing process. From (5) and (9), we yield $\mathrm{Dec} = \alpha(\mathrm{x_0} y_0 r_0 - x_i y_i r_i)$, we divide the smoothing process into two cases:

First, we keep the resolution $\mathrm{x_i} = \mathrm{x_0}$, $\mathrm{y_i} = \mathrm{y_0}$ and reduce frame rate if it satisfies (10) with $[a] = \min\{z \in Z, |z| \leq |a|$, we move to the second step if the condition does not satisfy.

Second, finding next resolution in the resolution list which has lower rate and satisfying (10),

$$\mathrm{Min_r} \leq r_i = \left[\frac{\alpha \times x_0 y_0 r_0 - Dec \times \beta}{x_i y_i \times \alpha}\right] \leq Max_r. \qquad (10)$$

$$\mathrm{Min_r} \leq r_i = \left[\frac{Dec \times \beta}{x_i y_i \times \alpha}\right] \leq Max_r. \qquad (11)$$

In the requiring user, after the sharing user decreases bandwidth usage, it has an amount Dec $\times$ β bps with $\beta \in R^{+}$is a factor which the environment impacts on the reducing amount of the network. If $\beta \sum_i Dec_i > B_M$ which means, it satisfies the requirement of the service with *Dec*$_i$ is an amount of decreasing bandwidth at ith sharing user. Thus, the resolution of requiring device is calculated by equation (11). When we decrease the rate from (x_0, y_0, r_0) to (x_i, y_i, r_i) in just a while, it makes inconvenience to the user because of video vibrant resolution. Thus, we must smooth the decreasing process by spreading out the decreasing time *t*. Suppose that (x_i, y_i, r_i) is an intermediate step between step 0 and *i*th, if the resolution changes in step *j* then $r_j = Max_r$. Otherwise, (x_j, y_j) is invariable and r_j is given by (13), $r_0 > r_i$ because the resolution does not change. Assigning B_{Step} is the number of decreasing resolution steps. A resolution takes a higher efficient quality than the frame rate. Thus, we assign B_{Step} with a weight higher than the weight of step for decreasing the frame, the total steps in the process is,

$$\text{step} = B_{Step} + \left(\frac{[r_0 - r_i]}{4} + 1\right). \quad (12)$$

$$r_j = \left[\frac{r_0 - r_i}{step}\right] + r_i. \quad (13)$$

In equation (12), step ≥ 1 even if $B_R = 0$ and $r_0 = r_i + 1$ or $r_0 = r_i - 1$. Given a time *t*, each step must take $\Delta t = t/\text{step}$ seconds.

At the beginning of this section, we propounded the method which describes how the new coming user, C gets streaming video in the case bandwidth almost saturated by A and B. The process for gaining resource is a *negotiating algorithm* with following steps:

Step 1: Each device determines its current state in the network, such as streaming video resolution and frame rate, minimum video streaming requirement, camera resolution list, and network bandwidth capacity.
Step 2: According to the current streaming state, new device requests resource from current devices.
Step 3: In each current device do a fuzzification depending on its current state.
Step 4: In each current device do a defuzzification to estimate the decreasing utilization bit rate.
Step 5: Calculating new resource and frame rate for all devices.
Step 6: In this step, the algorithm drive into two ways. First, if new device can get streaming video after the previous steps, then we move to the next step. Otherwise, we back to step 4. Secondary, we try to make a proportion of resource utilization for every device. If the condition satisfies, then we move to the next step. Otherwise, we back to step 4.
Step 7: Assigning new resolution and frame rate for preparing step 8.
Step 8: In case of sharing device, we will make a smooth decreasing process to make the convenience. If it is a new coming device, its bandwidth usage is not decreased, and then we end the process.
Step 9: Comparing the new value and initial value to make a smoothing process which is given in equation (12).

4.1 Simulation and experiment

To show how our method works, in this section we use network simulation tool OPNET, to illuminate the negotiation process between devices in the network. The network topology in the simulation is shown in Fig. 3 with three devices A, B, and C. We conduct the experiment with an initial camera resolution. For less complexity, we choose $\alpha = 1$, $\beta = 1$ in (9), (10), and (11). Both video source and destination have no encoder and decoder, the transmit rate sets to 8 Mbytes.

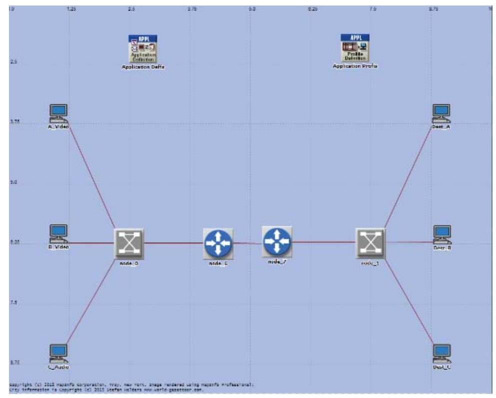

Fig. 3. Network topology of simulation

Three devices A, B, and C have video conferencing service with the imbalance bandwidth utilization. Each device negotiates with others and takes an average bandwidth utilization of the current network as the step of result is shown in Table II with one smoothing step between step 0 and 1.

Table 2. Balancing bandwidth usage in the local network

Step	Stream A (H/V/F)			Stream B (H/V/F)			Stream C (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

WebRTC works well in low bandwidth condition because it has an API for negotiating and assigning bandwidth amount to each peer bases on the network condition. The frame rate and resolution automatically change to match the available bandwidth. Thus, in the experiment, we only demonstrate the proportional bandwidth process and compare our research with the current implementation of WebRTC in EasyRTC [12]. The experiment uses EasyRTC server, which is installed on Ubuntu 12.04, the limited uploading speed of the network sets to 200Kbits by using an open-source Trickle. To create a virtual webcam device we use WebcamStudio open-source, in which we can control the resolution and frame rate of a video. Two hosts run on Chrome browser. It makes video streaming with two hosts in another computer which has Chrome browser. When the connection established between two hosts and its destination, we access to chrome WebRTC statistic for getting parameters measurement at *chrome://webrtc-internals/*.

Fig. 4 (a): A statistically in an original host using EasyRTC for video calling to another host in other computers.

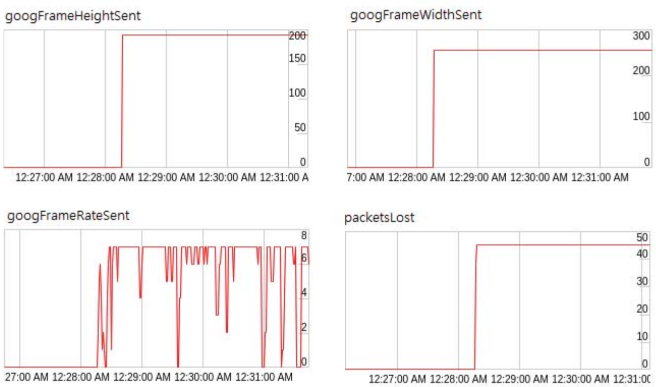

Fig. 4 (b): A statistically in a host which uses the negotiating algorithm for detecting resolution and frame rate before sending using EasyRTC for video calling to another host in other computers. In Fig. 17 and Fig 18 show that proposed method reduces a certain package lost with the same quality of information per frame, also, the frame rate is less variant than original EasyRTC implementation. By using the method, we detect the initial of the matching resolution before making streaming service. It does not only reduce package lost but it also reduces the computing resource in both sides of streaming service.

5 Conclusions

In this paper, we have presented the flexible method for sharing and balancing bandwidth of the local wired and wireless network using fuzzy control theory with the simulation using OPNET and the experiment using the implementation of WebRTC, EasyRTC. The method does not provide the best quality for each individual, but every user in the same group deserves to have a beneficial service with the lowest quality instead of losing service. By such way, we proportionate the quality service for every user in the same group where has no priority of serving service. Throughout the experimental section, the proposed method solved completely the problem lead on introduction section. Each sharing device just gives a small amount of bandwidth that does not have much effect on its service, but the total decreasing amount is higher when we have more and more involving users. So far, the experiment demonstrated the proportion of the bandwidth utilization beyond the requirement of the introduced problem.

In the future works, this method does not only apply for video streaming application, but it can also use to make a sharing protocol between devices, such as computer, mobile, printer, Set Top Box etc., in the same network.

6 Acknowledgements

This work (Grants No.C0218369) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2014, and also supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(NRF-2013R1A1A2013740)

7 References

[1] H. Kim and N. Feamster, "Improving network management with software defined networking," Communications Magazine, IEEE, Vol. 51., pp. 114-119, 2013.

[2] M. Baldi, S. Gai, and G. P. Picco, "Exploiting code mobility in decentralized and flexible network management," in Mobile Agents, pp. 13-26, 1997.

[3] S. H. Kang and A. Zakhor, "Effective bandwidth based scheduling for streaming media," Multimedia, IEEE Transactions on, Vol. 7., pp. 1139-1148, 2005.

[4] H. Qi, M. Stojmenovic, K. Li, Z. Li, and W. Qu, "A low transmission overhead framework of mobile visual search based on vocabulary decomposition," Multimedia, IEEE Transactions on, Vol. 16., pp. 1963-1972, 2014.

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

[6] A. Bergkvist, D. Burnett, and C. Jennings, "A. Narayanan," WebRTC 1.0: Real-time Communication Between Browsers," World Wide Web Consortium WD WD-webrtc-20120821, 2012.

[7] A. Johnston, J. Yoakum, and K. Singh, "Taking on WebRTC in an enterprise," Communications Magazine, IEEE, Vol. 51., pp. 48-54, 2013.

[8] LU SH, YIN JP, CAI ZP, ZHAO WT, "Efficient available bandwidth estimation for network paths," Journal of Harbin Institute of Technology, Vol. 1, No.032., 2008.

[9] Song S, Keleher P, Bhattacharjee B, Sussman A, "Decentralized, accurate, and low-cost network bandwidth prediction," INFOCOM, 2011 Proceedings IEEE, pp. 6-10, Apr 10. 2011.

[10] T S. H. Hao, E. Izquierdo, and Pengwei, "Bandwidth-efficient Packet Scheduling for Live Streaming with Network Coding," IEEE Transactions on Multimedia, 2016.

[11] Liang, Qilian, Nilesh N. Karnik, and Jerry M. Mendel, "Connection admission cont1rol in ATM networks using survey-based type-2 fuzzy logic systems," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions, Vol. 30 No. 3., pp.329-339, 2000.

[12] Priologic Software Inc, “EasyRTC Documentation”, 2014.