# Negotiated Bandwidth of Streaming Video by Using Fuzzy Logic in WebRTC

**Ma Van Linh***, **Jinsul Kim****

**Abstract**

WebRTC is one of the most recent technologies that supports browser-to-browser for voice calling, video chat, and P2P file sharing without the need of either internal or external plugins. However, there has some limitation which lets our development deal with many problems. This research will focus just on a small field of that problem which is bandwidth. While the bandwidth for downloading and uploading is fixed by providing service, but the number of users in a certain area is increasing largely by the time. In this paper, we propose a model to overcome the limitation of the bandwidth based on fuzzy control to adjust utilized bandwidth by changing frame rate and resolution of streaming video.

**Keywords:** Fuzzy theory, control bandwidth, streaming video.

*School of Electronics& Computer Engineering, Chonnam National University, Master Course linh.mavan@gmail.com
**School of Electronics& Computer Engineering, Chonnam National University, Professor(Corresponding author) jsworld@jnu.ac.kr

## 1. Introduction

Along with the development of human beings, technologies are playing in an important part, especially Network communication technology. Nowadays, a real time communication is the need of our requirement. By the limitation of network infrastructure, we can't develop beyond those. Suppose the situation that, in a call center only has a 5x5 internet connection. We had two persons with 2 subscribed streams coming in just fine (perfect clarity) and the 3rd one was coming in audio-only because of bandwidth issues. But everybody was publishing and subscribing while physically sitting in the same room on the same network. The problem was that the first 2 publishers and subscribers had already completely saturated the network by the time the 3rd person came on. If we were able to set an absolute bandwidth use cap on all the subscribed streams, this situation would not happen. However, this solution isn't flexible and it might cause redundant network bandwidth, the bandwidth isn't maximized usable. Currently, we have no API to do so. We can have a several ways for controlling bandwidth. In this paper, we try to control bandwidth by reducing the frame rate and resolution of video. While we can reduce device's CPU usage by reducing the frame rate and the bandwidth for transferring data is reduced too.

Recently, fuzzy control is somehow used for the Intelligence Control System. It's based on the human mind for making a decision.

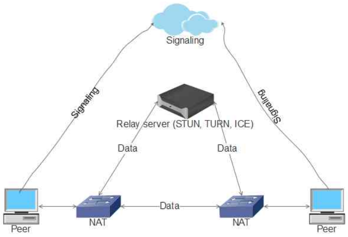
## 2. Related research

The valuating the performance[1] of various topologies using endpoints implementing WebRTC, is specifically evaluated the performance of the congestion control currently implemented and deployed in these web-browser, Receive-side Real-Time Congestion Control (RRTCC). They use transport impairments like varying throughput, loss and delay, and varying amounts of cross-traffic to measure the performance. Their results show that RRTCC is performant when by itself, but starves when competing with TCP. When competing with selfsimilar media streams, the late-arriving flow temporarily starves the existing media flows.

Scheduling research[2] in cloud environment, in which have features such as large-scale, diversity, and heterogeneity. The resources of user the user requirements for cloud computing resources are commonly characterized by uncertainty and imprecision. Also Many users generally cannot give a precise requirement in accordance with the attitude of the tasks. Therefore, they propose a dynamic resource scheduling method based on fuzzy control theory with resource requirements prediction, and the relationships between resource availability and the resource requirements.

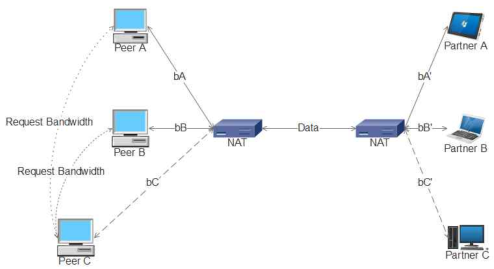
## 3. System Architecture.

In the case of webRTC, every participant will join as a peer. Thus, each participant has an equality to take network resources. Based on the quality of the network, webRTC will automatically deliver resources to each peer. In Figure 1, whenever a peer connection is established, those peers can't be re-configured (Signaling in webRTC is just a machine by which peers send control message to each other to establishing a communication protocol). Therefore, we propose a model where peers negotiated bandwidth with each other rather than using server for limit the transportation package.




[Figure 1] Connection between peers in webRTC

In the figure Figure 2 shows that, device A and B are using network by using WebRTC to share their data (video, audio) with their partners and their utilized bandwidth is bA and bB respectively, by the time device C join into the network. It only gets the audio signal which is described the dash arrow with bandwidth bC, instead of full video because of bandwidth limitation. Device C will communicate with A and B for negotiating network resources.




[Figure 2] Requesting bandwidth overview

## 3.1 Fuzzy inference

Referring to the levels of the peers have taken and the network resource requirements of new upcoming peers. We define the symbol as positive or negative in accordance with the increasing or decreasing network resources. The input variables are the requirements for resources and the need of resources of the established peers, the output variables are the resources that the new peer need to be increased or current peers need to be described frame rate or resolution of streaming video. The fuzzy values of the input and output variables are described as Positive High, Positive Medium, Positive Low, Normal, Negative Low, Negative Medium, Negative High. We can abbreviate those words as PH, PM, PL, N, NL, NM, NH by descending orders.

Naming the bandwidth requirement is bR and the need bandwidth of established peer is bN, which are the input of the system, and the output of system is named as bandwidth output bO which is the ability the peers can decrease frame rate or video streaming resolution. Therefore, we can express the set of rule as follows:

*If bR is PH and bN is NH, then bO is NH*

If the peer need high bandwidth and the established peers don't need much bandwidth, then it will reduce the frame with high rate. Probably, we do nothing if bR is kind of Negative because we don't want to decrease the bandwidth of upcoming peer. Also in the case bN is kind of Positive which means the established peers need more bandwidth, they can't share bandwidth with others. Thus, the output is N (do nothing). We can have more details of the input and output results in Table 1.

<Table 1> Request bandwidth based on fuzzy rules

| bR/bN | 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 |

## 3.2 Negotiating algorithm

The process for negotiating bandwidth resource is a dynamic algorithm based on the request of new coming peers and the present peers, which is described in Figure 3,

**Step 1**: Starting video streaming via webRTC and make fuzzy estimation of bandwidth resource.

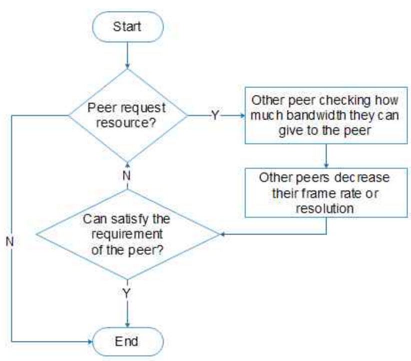
**Step 2**: Sending request resource to other peer in the same network if need.

**Step 3**: Other peers checking their current usage resource and how much they can share by fuzzy variable

**Step 4**: Other peers decease their frame or resolution based on previous step

**Step 5**: Others peers send responding message to tell that they have been shared bandwidth, the peer checks the satisfy bandwidth condition aiming whether need to send request again or not.

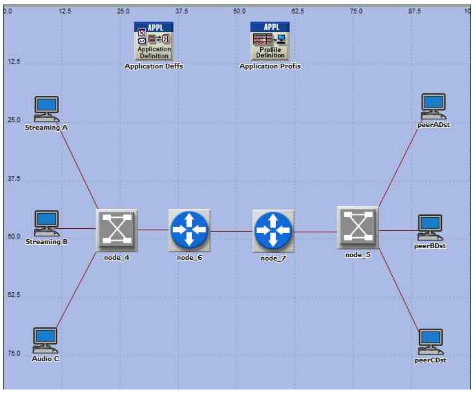
**Step 6**: End the negotiating process.



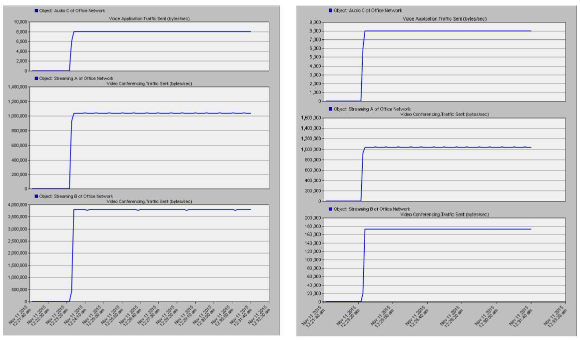

[Figure 3] Negociating bandwidth algorithm

## 4. Simulation experiments

In the figure 4, Streaming A streams with peerADst with Video Conferencing (Frame interarrival time: 30 frames/sec, Frame Size Information: 128*240, Type of service: Best effort<0>), Streaming B streams with peerBDst with Video Conferencing (Frame interarrival time: 15 frames/sec, Frame Size Information: 352*240, Type of service: Standard <2>), Audio C streams with peerCDst with Voice (PCM Quality Speech).




[Figure 4] Simulation network



Before changing resolution and frame rate in B

After the changing is applied in B

[Figure 5] Simulation results

Streaming B has a lower frame rate than Streaming A but the frame size of the B is more quality than the A. As you can see in the left of figure 5, The packages which the B sent is greater than the rest that means it takes more bandwidth than others. After reducing resolution size in B to 128*120 and frame rate to 10 frames/sec, the total packages is sent in B and the local network through the switch is reduced which is shown in the right figure 5.


## Acknowledgement

This research was partially supported by the IT R&D program of MSIP(Ministry of Science, ICT and Future Planning) / NIPA(National IT Industry Promotion Agency) [12221-14-1001, Next Generation Network Computing Platform Testbed] and also supported by Basic Science Research Program through the National Research Founda-tion of Korea(NRF) funded by the Ministry of Education, Science and Technol-ogy(NRF-2013R1A1A2013740)


## References


[1] Singh., & Varun., & Albert AbelloLozano., & Johannes Ott. (2013). Performance analysis of receive-side real-time congestion control for WebRTC. Packet Video Workshop (PV), 2013 20th International, 1-8.

[2] VChen., & Zhijia., & YuanchangZhu., & Yanqiang Di., & ShaochongFeng. (2015). A dynamic resource scheduling method based on fuzzy control theory in cloud environment. Journal of Control Science and Engineering. 2015(34), 1-10.

[3] Vogt., & Christian., & Max Jonas Werner., & Thomas C. Schmidt. (2013). Leveraging WebRTC for P2P content distribution in web browsers. Network Protocols (ICNP), 2013 21st IEEE International Conference, 1-2.

[4] VMassoulie., & Laurent., & ames W. Roberts. (2000). Bandwidth sharing and admission control for elastic traffic. Telecommunication systems, 15(1), 185-201.