

## Adaptive RTP Rate Control Method

Uras Tos

*Department of Computer Engineering  
Izmir Institute of Technology  
Izmir, Turkey  
urastos@iyte.edu.tr*

Tolga Ayav

*Department of Computer Engineering  
Izmir Institute of Technology  
Izmir, Turkey  
tolgaayav@iyte.edu.tr*

**Abstract**—In this paper, we present an adaptive method for maximizing network bandwidth utilization for the real-time applications. RTP protocol is chosen as the transport protocol and the network utilization is provided by increasing and decreasing the transmission rate of the RTP traffic. Our method employs a PID controller that keeps the RTP packet loss fraction at a predefined reference point. Packet loss fraction parameter is gathered from the RTCP receiver reports and fed into the PID controller that controls the transmission rate of the RTP traffic. Simulations are performed to demonstrate the effectiveness of the method under a bottleneck network configuration with background UDP traffic competing against the RTP traffic for the available bandwidth.

**Keywords**—adaptive; RTP; RTCP; simulation; feedback control;

### I. INTRODUCTION

TCP and UDP are the popular transmission protocols used on the Internet today. Of these two popular protocols, TCP has rate control capabilities that allow a TCP traffic to adapt itself to the changing network conditions. On the other hand, UDP lacks such a feature. Nevertheless, the lightweight nature of the UDP protocol makes UDP more suitable to real-time applications. However, increasing bandwidth requirements of real-time applications might in turn cause the networks flooded with uncontrolled rate UDP traffic. In such a network environment, the traffic with uncontrolled rate results in unavoidable congestion problems.

In order to address the transmission rate control need for the real-time applications, we have used RTP protocol defined by IETF. RTP is a networking protocol specifically designed for real-time application needs. Especially in multimedia audio and video streaming applications, in which the need of end-to-end QoS for efficient transmission is critical, transmission rate control is necessary [1], [2], [3]. Even though RTP does not have rate control functionality implemented in the specification, it has the means to gather information about the network state by the RTCP status packets. RTP works in conjunction with RTCP, which delivers periodic report packets to both parties of a RTP communication about the state of the network. However, the interpretation of the information contained in these feedback reports is left to the application that employs RTP as the transport protocol. The scope of our work is interpreting

these feedback reports to create and an adaptive rate control scheme for RTP traffic source.

The rate control method proposed in this paper is based on sender adaptation. Sender adaptation is a transmission rate adjustment scheme in which the traffic source adjusts the packet transmission rate to respond changing conditions of the network. In our method, sender adaptation is applied to RTP traffic and adaptation is accomplished by altering the transmission rate of the RTP packets by a PID controller that keeps the packet loss rate of the RTP traffic at a predefined value. Packet loss rate is gathered from RTCP receiver report packets.

This paper is organized as follows. Section 2 provides related work on transmission rate control. In Section 3, we describe the general definition of the PID feedback control system that is the basis our method. In Section 4, we provide detailed information about our method and how the RTP transmission rate controlled by the PID controller. In Section 5, the simulation environment and the simulation results indicating the benefits of our method.

### II. RELATED WORK

Literature [4] proposes PA-UDP, a method to maximize data transfer over high throughput network links. PA-UDP uses UDP for data transmission and TCP for control packets that carry network statistics. PA-UDP also extends its adaptation features over the CPU and disk performance of the endpoints. Literature [5] implements a rate control method for RTP traffic with additive increase multiplicative decrease (AIMD) approach. AIMD algorithm increments and decrements the transmission rate at a constant pace throughout the course of network transmission. Literature [6] provides means to control RTP flow by employing low pass filters and a constant increase/decrease method that depends on the last known state of the network. Literatures [7] and [8] propose adaptive methods for sender rate adaptation which is essentially a modified AIMD algorithm. This method perform calculates increment and decrement amounts as a function of the current and previous network state. Literature [9] also uses a similar approach for adjusting RTP transmission rate. This method proposes an improved

determination of increment and decrement amounts for the RTP transmission rate.

Our motivation is to propose a method that has a slightly different aim than the previous work. Previous work focus on TCP friendliness as the primary goal. The motivation behind this is ensuring that the controlled traffic coexists with TCP traffic peacefully and does not steal bandwidth from competing TCP traffic.

Proposed AIMD algorithms that control RTP traffic rate relies on additively increasing and multiplicatively decreasing the RTP traffic rate according to the network state described in the RTCP reports. An intelligent algorithm should not only rely on the present state and the last state before the present state. Doing so means that the algorithm loses track of the history of the network status. In the de facto scenario that the RTCP receiver reports are generated every 500ms, it takes at least two subsequent receiver reports to gather information about the trend in the changing condition of the network. Using the network status information gathered only in the last 1 second might not be enough to correctly understand the trend of the traffic flow going on at that moment. Our proposal originates from this issue. Next section describes our method to control the rate of the RTP traffic in such a way that includes the information learned from the history of the changing network status trends.

### III. FEEDBACK CONTROL SYSTEM

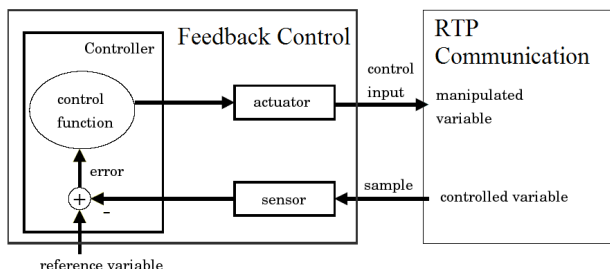


Figure 1. Feedback Control System

To introduce an autonomic behavior, we first map the classical feedback control to the RTP communication system. A typical feedback control applied to the communication system is shown in Figure 1. The major components are control related variables and a feedback control loop.

Control related variables include controlled variable, manipulated variable and reference variable. Controlled variable is a performance metric that characterizes the system performance over a period. The communication system controls the controlled variable to achieve the desired performance. For example, packet loss fraction and round trip delay are typical controlled variables in feedback communication systems.

Reference variable indicates the desired system performance in terms of a controlled variable and it is defined by the user. The difference between the reference variable and the corresponding controlled variable is called the error. For example, if a system sets its reference variable to 0.05 and the current controlled variable is 0.2, then the system can be said to have an error of -0.15. Manipulated variable is system attribute that is dynamically changed by the controller. Manipulated variable should be effective for performance control, i.e. changing its value should affect the system's controlled variables.

Feedback control communication system has a feedback control loop that is invoked at every new measurement of the measured variable. The loop is composed of a Sensor, a Controller, and an Actuator. The sensor measures the controlled variables and feeds the samples back to the Controller. The controller compares the reference variable with corresponding controlled variables to get the current errors, and calls the *control function* to compute a control input, the new value of the manipulated variable based on the errors. The control algorithm is a critical component with significant impacts on the performance and hence is the core of the design of a feedback control system. Notice that control theory may enable us to derive the control algorithm and analytically prove that the algorithm provides the desired system performance. Finally, the actuator changes the manipulated variable based on the newly computed control input.

### IV. ADAPTIVE CONTROL OF THE TRANSMISSION RATE

RTCP receiver report packets contain fields that carry information about the present state of the network. One of the metrics contained in these reports is the loss fraction. Loss fraction represents the fraction of the RTP packets lost during transmission in between two subsequent RTCP receiver reports. Assessing the network condition over the changes in the loss fraction value is the foundation of the method presented in this paper.

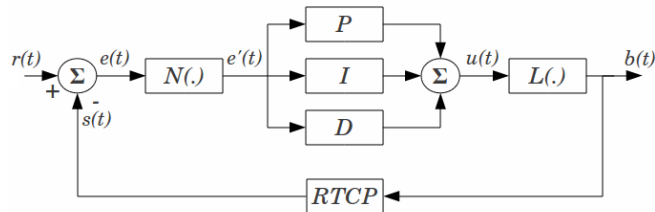


Figure 2. PID controller used in the proposed method.

In order to autonomously adapt the RTP transmission rate, a PID controller is implemented as shown in Figure 2. In the PID controller, RTP packet loss fraction, denoted with  $s(t)$ , is gathered from the RTCP receiver reports and used as the measured variable. Since the value of  $s(t)$  is provided by the

RTCP receiver reports, we assume that there is no measurement error. The error  $e(t)$ , the value difference between  $r(t)$  and  $s(t)$  is normalized and fed into the PID controller as the normalized error, denoted with  $e'(t)$ . The controller starts the RTP traffic with the minimum transmission rate ( $B_{min}$ ) and measures the value of the RTP packet loss fraction gathered by the RTCP receiver reports. After the measurement, the controller compares the measurement against the reference value. P, I and D components of the controller performs the necessary calculations according to the error generates a new RTP transmission rate to be used by the RTP sender. Manually tuned system parameters of the PID controller can be seen in Table I and the inner workings of the PID controller is explained throughout this section.

Table I  
SYSTEM PARAMETERS AND THEIR RESPECTIVE VALUES

| Parameter                               | Value    |
|---|----------|
| Proportional gain ( $K_p$ )             | 1200000  |
| Integral gain ( $K_i$ )                 | 410000   |
| Derivative gain ( $K_d$ )               | 150000   |
| Error constant ( $E_c$ )                | 0.15     |
| Reference variable ( $r(t)$ )           | 0.05     |
| Sampling period ( $SP$ )                | 500 ms   |
| Minimum transmission rate ( $B_{min}$ ) | 100 kb/s |
| Maximum transmission rate ( $B_{max}$ ) | 5 Mbit/s |

At any given time the packet loss fraction indicated by the RTCP receiver reports ranges from 0 to 1. In this application, 0.05 is chosen as the reference variable to keep the RTP packet loss at a relatively minimum, while allowing enough packet loss to enable the PID controller to manipulate the transmission rate. However, there is a significant difference in the absolute values of the maximum values of positive and negative error values. If the error is fed into the controller without any form of normalization, this would cause an unwanted biasing effect. This effect results in the controller to run in the  $b(t)$  increment direction slower than it runs in the  $b(t)$  decrement direction. Eliminating this unfair operation, error values are processed in an error normalization function ( $\mathcal{N}(e(t))$ ).

Let  $t \in \mathbb{N}$  denote the discrete time index, i.e. the actual time  $\tilde{t}$  can be computed as  $\tilde{t} = SPt$ .

$$e(t) = r(t) - s(t) \quad (1)$$

$$e'(t) = \mathcal{N}(e(t)) = \begin{cases} E_c & \text{if } e(t) = r(t) \\ e(t) & \text{if } e(t) < r(t) \end{cases} \quad (2)$$

$$P(t) = K_p e'(t) \quad (3)$$

$$I(t) = K_i \sum_{i=0}^t e'(i) \quad (4)$$

$$D(t) = K_d (e'(t) - e'(t-1)) \quad (5)$$

$$u(t) = P(t) + I(t) + D(t) \quad (6)$$

The PID controller consists of three components [10]. First component of the PID controller is  $P(t)$ , namely the proportional component. The output of the proportional component is the multiplication of proportional gain, denoted by  $K_p$ , and the normalized error value. Second component is  $I(t)$ , the integral component. Integral component is the product of integral gain, denoted by  $K_i$  and the sum of normalized error values from time 0 to time  $t$ . Final component is  $D(t)$ , the derivative component. Derivative component outputs the product of the derivative gain,  $K_d$ , and the difference between current and previous value of the normalized error. After each component of the PID controller is calculated, the sum of outputs is fed into the limiting function,  $\mathcal{L}(u(t))$ .

In the cases where the packet loss fraction cannot be decreased within the acceptable limits even though the RTP transmission rate is slowed down to the point of nearly stopping, the PID controller might blindly continue to decrease the RTP transmission rate and eventually stop it. This is an unwanted situation. The PID controller gets the feedback from the RTCP receiver reports and for the RTCP receiver reports to be generated, there needs to be an RTP traffic. If the PID controller is let to decrease the RTP transmission rate to the point of stopping, the whole operation of the system is crippled. Therefore, a limiting function, denoted by  $\mathcal{L}(u(t))$ , is implemented on the RTP traffic reduction. If the adjusted RTP transmission rate reaches to the point of the minimum allowed transmission rate ( $B_{min}$ ), the RTP transmission rate is limited to  $B_{min}$  and is not allowed to decrease further more. Similarly, the RTP transmission rate cannot exceed the bandwidth of the link, namely  $B_{max}$ . In this manner, the output of the limiting function is used as the new transmission rate ( $b(t+1)$ ) for the RTP traffic.

$$\mathcal{L}(u(t)) = \begin{cases} B_{max} & \text{if } u(t) > B_{max} \\ B_{min} & \text{if } u(t) < B_{min} \\ u(t) & \text{if } B_{min} \leq u(t) \leq B_{max} \end{cases} \quad (7)$$

$$b(t+1) = \mathcal{L}(u(t)) \quad (8)$$

Once  $b(t+1)$  is calculated, the RTP traffic rate is immediately set at this value and the RTP traffic continues to flow. The PID controller waits for the sampling period ( $SP$ ) amount of time until a new RTCP receiver report packet is received. Upon the arrival of the new RTCP receiver report packet, the PID controller calculates the new traffic rate according to the new packet loss fraction. This autonomous operation continues to run as long as there is RTP traffic flow on the network.

The results that demonstrate the effectiveness of the proposed method are presented with various simulation scenarios in the next section.

## V. SIMULATION

As a widely used application for network simulation, all of the simulation work presented in this paper is performed by NS, the discrete event network simulator.

### A. Simulation Setup

With the aim of demonstrating the capabilities of the proposed method, a simple bottleneck network topology is created. Figure 3 shows the visual representation of this network topology. Nodes  $s_1$  and  $s_2$  represent RTP traffic source and background traffic source, respectively. Nodes  $r_1$  and  $r_2$  denote the topology routers. Finally nodes  $d_1$  and  $d_2$  represent the destinations for RTP and background traffic.

All links except the link between  $r_1$  and  $r_2$  are 10Mb/s duplex link with 10ms propagation delay. The bottleneck link between  $r_1$  and  $r_2$  is a 5Mb/s duplex link with 10ms propagation delay. Node-to-router link bandwidths are selected higher than the router-to-router link bandwidth to avoid any potential bottleneck over any link other than the bottleneck link. Placing the sources on opposite sides of the bottleneck link might affect the transmission of the RTCP packets that carry the receiver reports. Therefore the sources are placed on the same side of the bottleneck link.

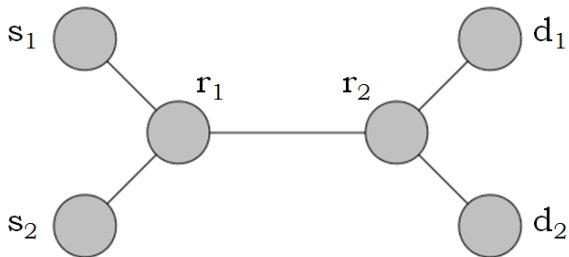


Figure 3. Network topology of the simulation.

The sender can only gather information about the network condition once the RTP traffic starts and the RTCP receiver reports begin to arrive. Therefore, before RTP traffic starts to flow, i.e.  $t=0$ , the sender does not have any information about the available bandwidth. However, despite the lack of this information, sender needs to set an initial transmission rate to start the traffic until the first RTCP receiver report packet arrives. We are only interested in how our method adjusts the transmission rate over the long run, therefore any initial transmission rate is as good as the other. For this reason, we start the RTP traffic with the minimum transmission rate,  $B_{min}$ , which is already defined in the PID controller.

In any given simulation scenario, the PID controller requires a certain amount of time to reach full utilization of the available bandwidth. This transient state might affect the user experience. The simulations should run long enough to let the PID controller past the transient state. On the other

hand, unnecessarily longer simulations may generate redundant data and make it difficult to see the operation of the PID controller when the data is plotted. Therefore, simulations are limited to 90 seconds, which is an appropriate value for simulations presented in this paper.

### B. Simulation Results

The goal of the simulations study is to demonstrate how well the proposed method keeps the RTP traffic rate at the maximum while keeping the RTP packet loss under control for any condition of the network. Simulating the changing conditions of the network is accomplished by introducing a background traffic source. This background traffic competes against the RTP traffic for the bandwidth utilization.

For background traffic generation we chose an UDP source carrying constant bit rate (CBR) traffic. A TCP source might also be incorporated for background traffic generation. However, TCP's rate control scheme might cloud the indication that whether increase or decrease of the packet loss is caused by the proposed algorithm or TCP rate control methods. In this case UDP's lack of any form of rate controls is the primary reason why UDP is chosen as the transport protocol for the background traffic. TCP-friendly rate control is out of the scope of this work, therefore no TCP source is used in the simulations.

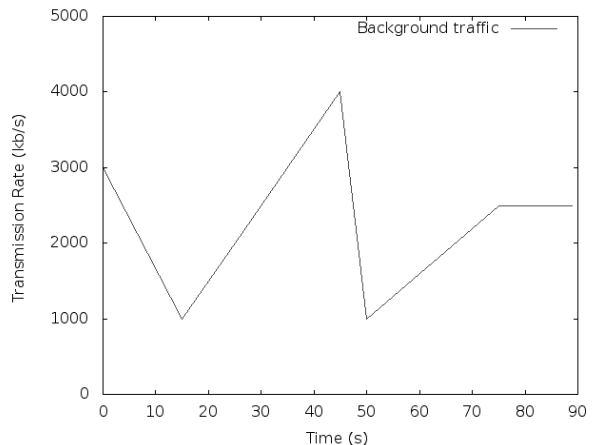


Figure 4. Background traffic transmission rate over the timespan of the simulation.

The transmission rate pattern of the background traffic is predefined and varies throughout the simulations. In all of the simulations, this same UDP source with the same pattern is used in conjunction with the RTP traffic source, which is also carrying CBR payload. The transmission rate pattern of the background traffic can be observed in Figure 4.

For the first simulation, we let the RTP source run with an uncontrolled constant traffic rate. The constant transmission rate of the traffic is set at 4 MB/s at the beginning of the simulation and kept at the same rate until the simulation

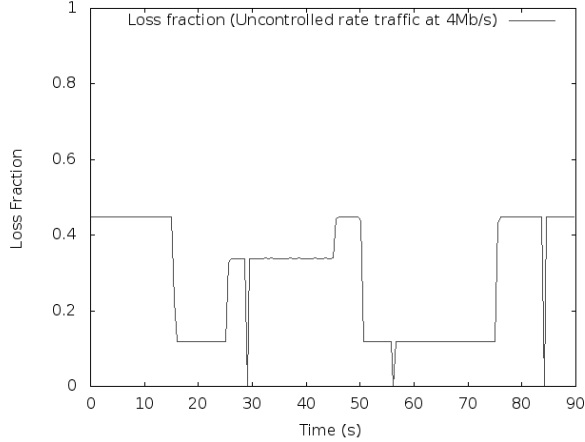


Figure 5. RTP packet loss fraction of 4Mb/s constant traffic ( $MV=4\text{Mb/s}$ ) over the timespan of the simulation.

ends. The result is the RTP packet loss throughout the simulation. As it can be seen in Figure 5, constant and uncontrolled high transmission rate of the RTP traffic results in high packet loss.

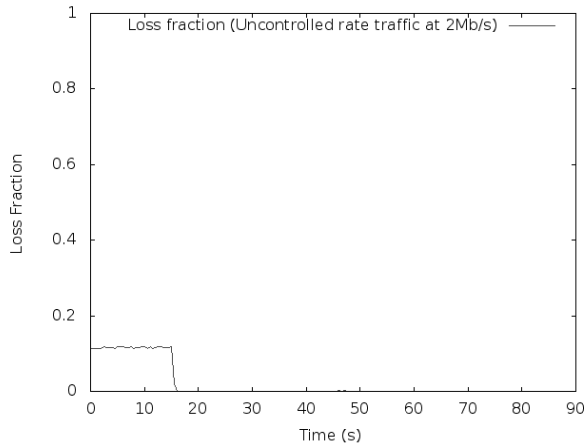


Figure 6. RTP packet loss fraction of 2Mb/s constant traffic ( $MV=2\text{Mb/s}$ ) over the timespan of the simulation.

In the second simulation, a constant and uncontrolled RTP transmission rate is employed again, with the difference that this time the transmission rate is limited at 2 Mb/s. This simulation resulted in low packet loss (Figure 6), however the throughput of the RTP traffic is low and is not enough to fully utilize the available bandwidth.

In the final simulation, the RTP transmission rate is controlled by the PID controller. Figure 7 shows how the PID controller adapts the rate of the RTP traffic in response to the changes in the background traffic. As the background traffic increases and decreases, the RTP transmission rate is adjusted accordingly. At the same time, the packet loss is controlled and kept at the reference point (Figure 8).

At the points where background traffic rate increases and causes the RTP traffic to suffer packet losses, the RTP traffic rate is autonomously adapted to this change and in turn enables the RTP packet losses to decrease. By keeping the utilization high and packet losses low, the proposed method integrates QoS capabilities to the network.

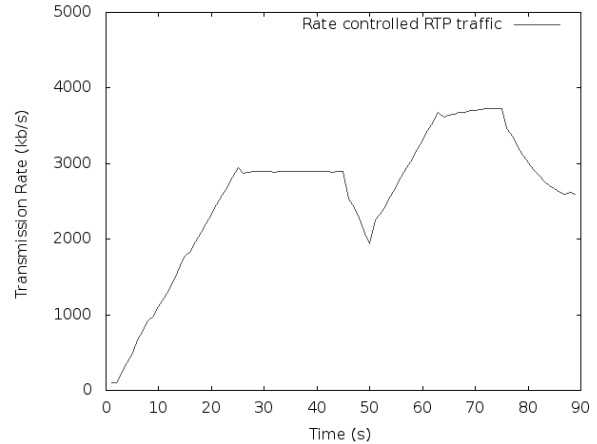


Figure 7. RTP transmission rate of PID controlled traffic over the timespan of the simulation.

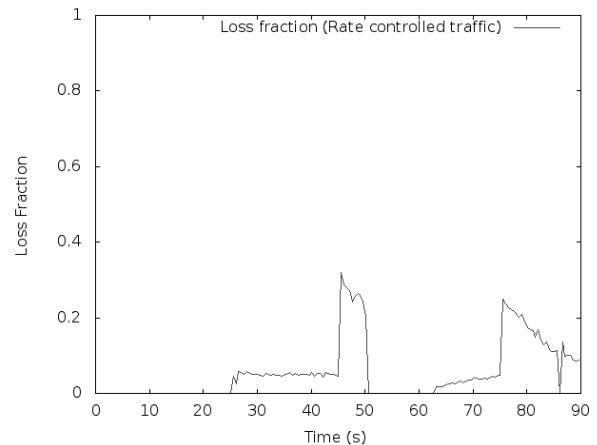


Figure 8. RTP packet loss fraction of PID controlled traffic over the timespan of the simulation.

## VI. CONCLUSION

We have presented an approach on maximizing the bandwidth utilization while keeping packet loss under control for the RTP traffic in situations with varying background traffic. In the proposed method, the loss rate of the transmitted RTP packets is gathered from the RTCP receiver report packets and fed into a PID controller. PID controller modifies the transmission rate of the RTP packets to keep the loss rate at a predefined reference point therefore adding QoS capabilities

to the network. Simulations show that the proposed method successfully controls the transmission rate of the RTP traffic albeit with a small amount of response delay. Proposed method only inputs the loss fraction as the measured variable for the PID controller. This results in the system to adjust the sending rate once the packet losses start to occur. Packet loss is usually preceded by increasing transmission delays [11]. Future work may focus on incorporating another measured variable, namely the transmission delay. Since feedback control paradigms that employ more than one measured variable are not uncommon [12], using transmission delay as a measured variable might provide the controller the ability to adjust the rate of the RTP transmission even before the actual packet losses occur. Additionally, intelligent parameter training might eliminate the manual tuning of PID controller parameters and make the system suitable for different network configurations. Comparison with present transmission rate control methods may also be considered. Please note that control theoretical analysis is needed such that the stability properties of the system must be evaluated.

#### REFERENCES

- [1] C. Bouras and A. Gkamas, "Multimedia transmission with adaptive qos based on real-time protocols," *International Journal of Communication Systems*, vol. 16, no. 3, pp. 225–248, 2003. [Online]. Available: <http://dx.doi.org/10.1002/dac.585>
- [2] M. Wagner, I. Kopilovic, S. Dohla, N. Farber, and H. Fuchs, "Towards an rtp profile for iptv," *Broadcasting, IEEE Transactions on*, vol. 55, no. 2, pp. 343–352, 2009.
- [3] P. Papadimitriou and V. Tsaoussidis, "A rate control scheme for adaptive video streaming over the internet," in *Communications, 2007. ICC '07. IEEE International Conference on*, 2007, pp. 616–621.
- [4] B. Eckart, X. He, and Q. Wu, "Performance adaptive udp for high-speed bulk data transfer over dedicated links," in *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, 2008, pp. 1–10.
- [5] A. Barberis, C. Casetti, J. C. D. Martin, and M. Meo, "A simulation study of adaptive voice communications on ip networks," *Computer Communications*, vol. 24, no. 9, pp. 757–767, 2001. [Online]. Available: <http://www.sciencedirect.com/science/article/B6TYP-42WP80B-2/2/a19496339e1f93aba443578bd1d4de32>
- [6] P. Ling and L. ShaoWen, "An improved algorithm of rtp adaptive transmission control," in *Genetic and Evolutionary Computing, 2009. WGEN '09. 3rd International Conference on*, 2009, pp. 595–599.
- [7] D. Sisalem and A. Wolisz, "Towards tcp-friendly adaptive multimedia applications based on rtp," in *Computers and Communications, 1999. Proceedings. IEEE International Symposium on*, 1999.
- [8] D. Sisalem, "End-to-end quality of service control using adaptive applications," in *In IFIP Fifth International Workshop on Quality of Service (IWQOS '97, 1997*.
- [9] C. Wanxiang and L. Zhenming, "An modified rtp adaptive algorithm," in *Info-tech and Info-net, 2001. Proceedings. ICII 2001 - Beijing. 2001 International Conferences on*, 2001.
- [10] M. Johnson, M. Moradi, and E. Corporation, *PID control: new identification and design methods*. Springer, 2005.
- [11] W. Jiang and H. Schulzrinne, "Modeling of packet loss and delay and their effect on real-time multimedia service quality," in *PROCEEDINGS OF NOSSDAV '2000, 2000*.
- [12] C. Lu, J. A. Stankovic, S. H. Son, and G. Tao, "Feedback control real-time scheduling: Framework, modeling, and algorithms," *Real-Time Syst.*, vol. 23, pp. 85–126, July 2002. [Online]. Available: <http://dx.doi.org/10.1023/A:1015398403337>