

# An Integrated Software Architecture for Bandwidth Adaptive Video Streaming

T. Arsan

**Abstract**—Video streaming over lossy IP networks is very important issues, due to the heterogeneous structure of networks. Infrastructure of the Internet exhibits variable bandwidths, delays, congestions and time-varying packet losses. Because of variable attributes of the Internet, video streaming applications should not only have a good end-to-end transport performance but also have a robust rate control, furthermore multipath rate allocation mechanism. So for providing the video streaming service quality, some other components such as Bandwidth Estimation and Adaptive Rate Controller should be taken into consideration. This paper gives an overview of video streaming concept and bandwidth estimation tools and then introduces special architectures for bandwidth adaptive video streaming. A bandwidth estimation algorithm – pathChirp, Optimized Rate Controllers and Multipath Rate Allocation Algorithm are considered as all-in-one solution for video streaming problem. This solution is directed and optimized by a decision center which is designed for obtaining the maximum quality at the receiving side.

**Keywords**—Adaptive Video Streaming, Bandwidth Estimation, QoS, Software Architecture.

## I. INTRODUCTION

IN general, users want maximum speed and minimum delay even using a minimum bandwidth required network applications. On the other hand, in real-time applications such as video streaming, TV broadcasting or video-conferencing, the service quality must be maximized for satisfying the users. Some users never know what exactly happens after starting a real-time application. They only take the quality into consideration, other facts are not important for them. The more you pay for the bandwidth, the more you expect. Unfortunately, today's technologies cannot provide the best solutions for video streaming users that use best effort delivery networks.

Many researches have been done for better network performance. Especially, one of the most important network devices - a router uses path determination methodology. It compares a destination address to the available routes in its routing table and selects the best path. Although network devices try to find a best path to the destination, the algorithms they used are restricted to their ability. There will be always a better path available in the Internet. *In 30-80% of*

*the cases, the best paths found by classic routing algorithms are suboptimal from a media perspective [1, 2].*

If we want to use the Internet, we should find the most convenient protocol for video streaming. It is not possible to use lower-layer transport layer protocols in video streaming, because Transmission Control Protocol-TCP has huge amount of delay while retransmitting lost packets, and User Datagram Protocol-UDP does not guarantee packet delivery and packet loss recovery. An upper-layer transport protocol will be necessary. In this point, Real-time Transport Protocol-RTP and Real Time Control Protocol-RTCP protocols may be helpful, RTP is designed to provide end-to-end transportation and to support real-time applications, but RTP does not guarantee the best video streaming performance. RTCP, part of the RTP specification, provides QoS feedback and monitors the QoS. RTCP is not intended to support all of an application's control signaling requirements. Still one of the Session Control protocols such as Real Time Streaming Protocol-RTSP or a convenient control mechanism, may therefore be needed. RTSP is a session control protocol and its aim is to provide video streaming over the Internet. All these algorithms try to find a best solution for routing the packets, and best effort delivery, but none of them guarantees video streaming or briefly quality of service routing. There should be a mechanism to help regulating video streaming packets before or after network devices' routing for considering quality of service routing, resource reservation and maximizing the video quality at receiving side.

Video streaming system has some constraints such as bandwidth, congestion, delay and etc. The Internet has also a heterogeneous structure and performance measurement is not considered as the primary goal of its architecture. To obtain a better video streaming system, end-to-end transport performance should be optimized. It is possible to understand the network characteristics by using a bandwidth estimator. This qualification makes packet forwarding more effective and evaluates the performance of video streaming via the Internet. Some bandwidth estimation tools consider bandwidth-estimation techniques for networking solutions, but we need to consider bandwidth-estimation tools as video streaming point of view.

Rate controller has an important role in video streaming. If the aim is to maximize the quality at the receiving side, streaming rate should be well regulated and should be optimized. On the other hand, rate should be allocated for an optimized path. If we consider multipath scenario, it is important to determine which paths should ideally be used for

Manuscript received November 30, 2007.

T. Arsan is with Kadir Has University, Computer Engineering Department, Cibali Merkez Kampusu, 34230 Cibali-Fatih Istanbul, Turkey (phone: 90-212-533 65 32 ext.1405; fax: 90-212-533 57 53; e-mail: arsan@khas.edu.tr).

the streaming application, in a given available network topology between a streaming server and a client. This is different from classic routing algorithms that find the best path given some established network metrics. Classic routing algorithms may be optimal in terms of network utilization, but it is certainly suboptimal from the point of view of the media streaming application.

There are some alternative mechanisms such as joint-cross channel coding, dynamic resource allocation, cross layer optimization and network adaptive video transmission to improve quality level at receiver. Moreover, upper-layer transport protocol usage, bandwidth estimation tools and rate controller are taken into consideration separately as a solution for effective video streaming in the literature, but there is no joint solution on bandwidth estimation and rate control available and only a few control engineering oriented solutions available in the literature such as PID controller, fuzzy controller or neural network controller.

In this paper, an overview for video streaming concept is given and special architectures for bandwidth adaptive video streaming are introduced as a total solution. While Bandwidth Adaptive Streaming Controller uses bandwidth estimation, adaptive rate controllers, and multipath rate allocation algorithm, QoS controller calculates delay and packet loss values. Decision Center optimizes rate-delay-packet loss oriented cost function and determines optimal rate and optimal path. The rest of this paper is organized as follows. In the following section, video streaming overview is provided. We give a survey on bandwidth estimation tools in Section 3. We introduce complete system architecture of bandwidth adaptive video streaming system in Section 4 and we give the details of software information model of bandwidth adaptive video streaming in Section 5. We conclude the paper with research contributions in Section 6.

## II. VIDEO STREAMING OVERVIEW

Streaming is very popular in today's best effort delivery networks. Delivering audio or video content over the Internet can be achieved by two methods: progressive download and real-time streaming. If the content size is short, the progressive download method is used in general. In this method, media content is directly downloaded from a server into storage units of a client, but in real-time streaming, client software plays media content without storing the content into any storage units. Real-time streaming can be easily explained as delivering media from a server to a client over a network in real-time. The client software is responsible for playing the media as it is delivered. There are two main types of delivery options for real-time streaming: live and on-demand. Simulated live is also another method that is used with live streaming or on-demand as a part of them to add some extra materials such as prerecorded scenes, concerts, interviews and lectures, but this event is not live and there is no need any broadcasting tools. If the media content contains live events, this type of streaming is called live streaming. On the other hand, if the media contents can be provided on user's demand, it is called on-demand streaming.

### A. Video Streaming Architectures

There are several common media streaming architectures available in the commercial use. These architectures are Single Sender - Single Receiver Streaming System, Single Sender - Multiple Receivers Streaming System, and Multiple Senders -Single Receivers Streaming System as seen in Fig. 1.

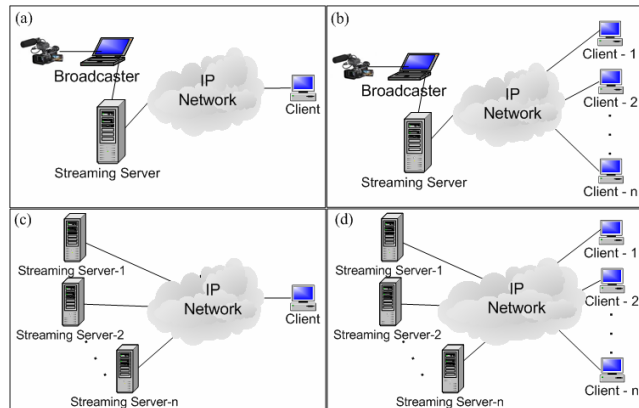


Fig. 1 Video Streaming Architectures:

(a) Single Sender-Single Receiver (b) Single Sender-Multiple Receivers (c) Multiple Senders-Single Receiver (d) Multiple Senders-Multiple Receivers

The first system is the most common streaming architecture, but the second system is the typical broadcast architecture. Although, in the first system the bandwidth will be enough for streaming with the constraints of delay and packet recover acknowledgement, in the second system there is a need for bandwidth regulation and adaptation for network conditions. The third and fourth systems will be popular in the near future, because of its distributed system architecture structure. This system should have a robust scheduling structure, because it is necessary to send the media content to client in a certain hierarchical order, and also client should put the received packet in correct order to have the media content. In recent commercial applications, a new concept is in use: Relaying Streaming Media. The main idea is to listen to the incoming stream and then send that stream to corresponding destinations. The main goal is to reduce the Internet bandwidth consumptions.

In general, streaming technologies support the latest digital media standards such as AAC, H.264, MP3, MPEG-4 and 3GPP. Client software uses a protocol rather than TCP/IP, because of its long and bandwidth greedy acknowledgement procedure. These protocols are Real-Time Transport Protocol (RTP) or Real-Time Streaming Protocol (RTSP) etc. and these protocols run in the application layer of the Internet, and control the streaming of media content. First, the client sends a RTP or RTSP request to server and server receives the request and calls necessary software modules to satisfy the request and to start the streaming.

### B. Typical Internal Structure of Video Streaming

Typical video streaming system includes broadcaster, repository for archiving, streaming server and a receiver unit

or a client as seen Fig. 2.

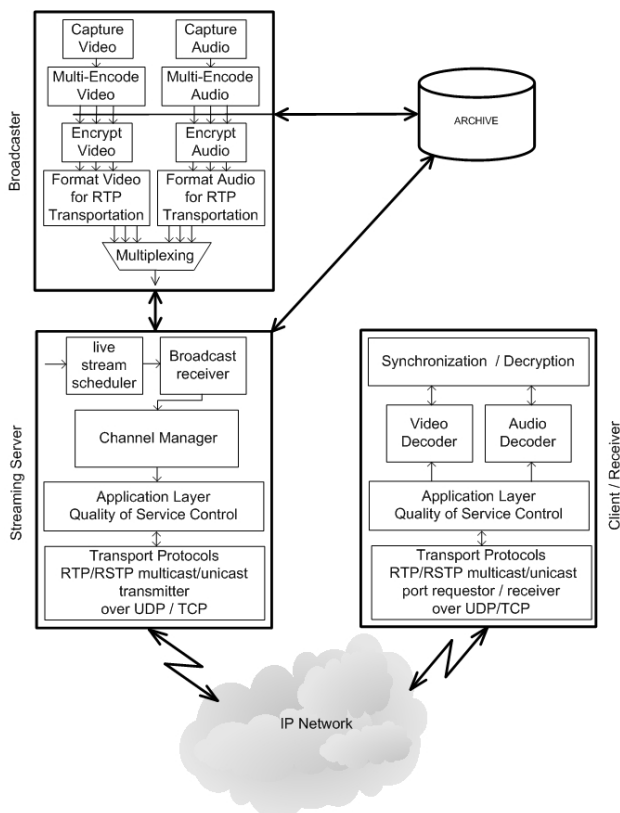


Fig. 2 Internal Structure of Video Streaming

The main idea is to capture video and audio signals from the source of the raw video and audio and put these media content into correct form to transmit on IP network. Raw video and audio must be compressed before transmission on IP network. This makes data transmission effective, but it is also noted that transmission of the packets should be under control of the quality of service control. Application layer Quality of Service control tries to achieve this task. It includes congestion control and error control techniques for successful transmission of the packets on IP networks. Congestion control is used for preventing packet loss and reducing the delay. Error control is used for improving video presentation quality, if the packet loss is occurred. Forward Error Correction (FEC), retransmission, error-resilient encoding and error concealment are some examples of error control mechanism. To implement the some methods, continuous media distribution services are necessary. In the Internet, IP protocol can successfully try to achieve this task by means of its best-effort delivery structure. These services also include network filtering, application-level multicast and content replication [3, 4].

In video streaming technology, streaming servers have one of the most important roles. Streaming servers offer quality streaming services, and contain transport protocols, operating system and storage system. Although, quality streaming can

be successfully achieved by the streaming servers, there will be a media synchronization mechanism to put together the video and audio. Otherwise, video and audio can not be synchronized at receiver's media playing tools. Finally, some protocols are needed to regulate the transmission of the packet by using one of the lower-layer transport protocols such as User Datagram Protocol-UDP or Transmission Control protocol-TCP, or a higher-level transport protocol such as Real-time Transport Protocol-RTP and/or Real Time Control Protocol-RTCP or a session layer protocol such as Real Time Streaming Protocol-RTSP or Session Initiation Protocol-SIP.

### III. BANDWIDTH ESTIMATION

Bandwidth is the amount of data that can flow through a network connection in a given period of time. Bandwidth is one of the most important factors in network engineering. It is used to analyze network performance and design new networks. On the other hand, there are any other ideas produced by the Internet users and Internet application developers. Streaming video and audio over a network requires huge amounts of bandwidth. IP telephony systems also need bandwidth. The networking professional should find an excellent solution for developing technologies. They generally make use of some analysis methods. Bandwidth estimation is one of these methods. Network engineers use bandwidth estimation tools for analyzing network performance, optimizing end-to-end transport performance and network devices' abilities. On the other hand, application developers such as video streaming and IP telephony technology developers are also curious about bandwidth concepts and bandwidth estimation methodology. Different types of bandwidth estimation tools are developed so far. They considered some metrics that network professionals prefer and/or some metrics that video streaming/network related application developers prefers. There are two major end-to-end measurement methods available as Active Methods or Passive Methods.

#### A. Active Methods

In Active Methods, the idea is to send dummy packets from sender to receiver and try to understanding the network characteristics by the help of transmitted packets. Although, Internet Control Message Protocol – ICMP based traditional tools are available such as *ping* and *traceroute* today, but it is not possible to obtain satisfactory results by using these tools. Especially, there are no details whether the packet was lost or response was lost in ICMP-based tools, because packet loss rates on the forward and reverse path are different from each other. Whereas customer oriented tools are helpful to evaluate the performance of Internet applications.

Different types of techniques exist for active bandwidth estimation such as

- Variable Packet Size Probing
- Packet Pair / Train Dispersion Probing
- Self-Loading Periodic Streams
- Trains of Packet Pairs.

There are different classifications available for active bandwidth estimation tools such as

- Per-hop Capacity Estimation Tools
- End-to-end Estimation Tools
- Available Bandwidth Estimation Tools
- TCP Throughput and Bulk Transfer Measurement Tools
- Intrusiveness of Bandwidth Estimation Tools

A classification of publicly available bandwidth estimation tools is given in Table I [5]. It is important to note that *cprobe* was the first end-to-end available bandwidth estimation tool in the literature. On the other hand, *treno* is the first tool to measure the Bulk Transfer Capacity of a path, and *cap* is the first canonical implementation of the Bulk Transfer Capacity measurement methodology. *NIMI* – National Internet Measurement Infrastructure of Vern Paxson uses *cap* to estimate Bulk Transfer Capacity of a path [5]. *Pathchar*, *pathload*, *Sting* and *pathChirp* are some today's popular examples for publicly available end-to-end bandwidth estimation Active Methods.

*cprobe* [6] estimated the available bandwidth based on the dispersion of long packet trains at the receiver, but then it is shown that the dispersion of long packet trains does not measure the available bandwidth in a path: instead it measures a different throughput metric, *pathcar* is a tool that is to estimate Internet link characteristics by measuring the round trip time (RTT) of packets sent from a single host. It is possible to obtain the per-hop capacity characteristics such as latency, bandwidth, queue and delays of any selected links. *clink*, *pchar* and the *tailgating* techniques are the examples of measuring the per-hop capacity methods. *Sting* is a TCP-based network measurement tool that measures the packet loss rate from sender to receiver and vice versa. *Sting* is developed under the lights of *NIMI* and it is similar to ICMP-based tools, but in this method TCP's error control mechanism can be used and it is possible to understand the direction in which a packet was lost.

*Pathload* is another active measurement tool that estimates the available bandwidth of a network path [7]. The main idea is that the one-way delays of a periodic packet stream show increasing trend, when the stream rate is larger than the available bandwidth. A sender process and a receiver process are running in *Pathload*. Periodic packet streams are achieved by UDP-User Datagram Protocol. On the other hand, TCP connection is used as control channel between two end-points. This algorithm has a very good approach for estimating bandwidth. It uses equation of  $(R = L / T)$  where  $T$  is transmission period,  $L$  is packet size and  $R$  is the transmission rate. Minimum transmission period for back-to-back minimum-sized packets is around 15-30 $\mu$ s. Therefore  $T_{min}$  is selected as 100 $\mu$ s. Minimum allowed packet size is 96bytes. At the beginning,  $L$  is set to 96bytes and considering target streaming rate,  $T$  is computed from the equation of  $(R = L / T)$ . In this point, it is also noted that selecting stream length as 100 packets rarely causes packet losses, and 100 packets also provides adequate number of delay measurements.

*pathChirp* is based on the concept of self-induced congestion and it is used the Self-Loading Periodic Streams methodology called self-loading packet chirps [8]. *Pathload* uses adaptive search method, for that reason it has a long convergence time. Both algorithms use different methodology, and their outputs are different from each other. *pathChirp* provides a single estimate of available bandwidth, whereas *pathload* provides minimum and maximum bounds on the available bandwidths. The idea of *pathChirp* is to use exponentially spaced highly efficient chirp probing train. The task is to estimate the available bandwidth over the path based on queuing delays of chirp probe packets transmitted from sender to the receiver and then conducting a statistical analysis at the receiver. As considered in most researches, congestion supposes to occur at the edge of the network close to the source or receiver. Data packets may encounter two congested queues, one on each end of their paths. *pathChirp* is a tool that presents a robust solution for these kinds of congested queues.

TABLE I  
CLASSIFICATIONS OF BANDWIDTH ESTIMATION TOOLS IN DETAILS

TOOL	AUTHOR	MEASUREMENT METRIC	METHODOLOGY
<i>pathchar</i>	Jacobson	Per-hop Capacity	Variable Packet Size
<i>clink</i>	Downey	Per-hop Capacity	Variable Packet Size
<i>pchar</i>	Mah	Per-hop Capacity	Variable Packet Size
<i>bprobe</i>	Carter	End-to-End Capacity	Packet Pairs
<i>nettimer</i>	Lai	End-to-End Capacity	Packet Pairs
<i>pathrate</i>	Dovrolis – Prasad	End-to-End Capacity	Packet Pairs & Trains
<i>Sprobe</i>	Saroiu	End-to-End Capacity	Packet Pairs
<i>Cprobe</i>	Carter	End-to-End Available Bandwidth	Packet Pairs
<i>Pathload</i>	Jain - Dovrolis	End-to-End Available Bandwidth	Self-Loading Periodic Streams
<i>IGI</i>	Hu	End-to-End Available Bandwidth	Self-Loading Periodic Streams
<i>pathChirp</i>	Ribeiro	End-to-End Available Bandwidth	Self-Loading Packet Chirps
<i>Treno</i>	Mathis	Bulk Transfer Capacity	Emulated TCP Throughput
<i>Cap</i>	Allman	Bulk Transfer Capacity	Standardized TCP Throughput
<i>Sting</i>	Savage	Achievable TCP Throughput	TCP Connection
<i>Ttcp</i>	Muuss	Achievable TCP Throughput	TCP Connection
<i>Iperf</i>	NLANR	Achievable TCP Throughput	Parallel TCP Connections
<i>Netperf</i>	NLANR	Achievable TCP Throughput	Parallel TCP Connections

### B. Passive Methods

While dummy packets are sent to receiver from sender in active methods, there is no injected packet in passive methods. The idea is to observe traffic already present in the network and then estimate the bandwidth of the network. This technique is called non-intrusive measurement technique, and it is used for detecting shared upstream congestion and discovering bottleneck (significant queuing) router link speeds [9]. This method considers Probability Distribution Function (PDF) of packet inter-arrival in a TCP flow. The PDF shows behavior of spike, spike bump, spike train and train of spike bumps. These characteristic behaviors are interpreted as a bottleneck with no substantial cross traffic, a low bandwidth bottleneck followed by a high bandwidth bottleneck, traversed bottleneck shared with a substantial amount of cross traffic and a low bandwidth upstream bottleneck shared with a substantial amount of cross traffic respectively.

In this method, one of the most important points is a clustering problem that detect shared bottleneck. In the receiving part of the end-to-end system, there is an observer that watches the arrivals of packets at some link. After all these steps, minimization of R nyi Entropy, which is a generalized version of Shannon Entropy formula, is used for discriminating between bottleneck sharing and non-sharing flows. Now the problem will change into an optimization (briefly a minimization) problem and then a cost function is selected and an algorithm for optimization is proposed.

## IV. SOFTWARE DESIGN OF AN INTEGRATED ARCHITECTURE

Rate control concept is basically related to application layer Quality of Service (QoS) control techniques. Application layer QoS techniques try to manipulate congestion and error. If we consider video streaming, congestion control takes the form of rate control. It is known that packet loss, delays, and network congestion may affect on video presentation quality. It is necessary to reduce the packet loss and delay by using congestion control techniques, so rate control techniques. The idea is so simple: to maximize the quality at receiver side.

In some conditions, especially lossy IP networks, rate control techniques should not only meet the bandwidth requirements but also assures other quality of service metrics such as delay, delay-jitter and cost.

In this study, as shown in Fig.3, a special architecture for bandwidth adaptive video streaming is implemented by using two different controllers: Bandwidth Adaptive Streaming (BAS) controller and Quality of Service (QoS) controller. The outputs of these two controllers are collected at a Decision Center. Decision Center is the heart of rate optimization and allocation by considering bandwidth, delay and packet loss metrics.

Firstly, we set the reference signal for convenient value, and then we run Bandwidth Estimation tools called *pathchirp*. *pathchirp* is the most important agent in this system, because it determines available bandwidth and regulates reference signal. On the other hand, *pathchirp* helps convenient path selection. Neural Network Rate Controller makes this

procedure more intelligent. It has a performance index function to optimize the performance.

Secondly, we try to guarantee the Quality of Service at the receiving side. We use the feedback signal of RTP called RTCP. RTCP includes the values for determining the values of packet loss rate and round-trip propagation delay. We have obtained these two values by the help of Packet Loss Observer and Delay Controller.

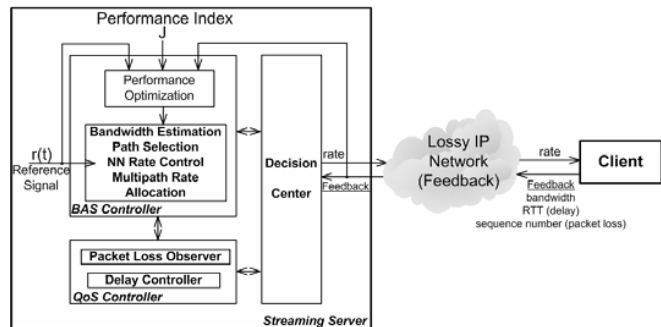


Fig. 3 Bandwidth Adaptive Video Streaming Architecture in Video Streaming Topology

### A. Bandwidth Adaptive Streaming (BAS) Controller

BAS controller is responsible for end-to-end bandwidth estimation, adaptive rate control and multipath rate allocation. This controller also selects the best path to destination by using extended version of bandwidth estimation algorithm for effective video streaming. In order to control the rate and to assure the service quality, feedback of related metrics should be provided. BAS controller needs bandwidth estimation value to determine the path and to regulate and to allocate the rate.

#### 1) Bandwidth Estimation and Path Selection

In BAS Controller, *pathChirp* algorithm which is one of the best effective publicly available bandwidth estimation tools is considered as a bandwidth estimator [8]. Original *pathChirp* algorithm estimates the best available network connections bandwidth. It does not however, search any better bandwidth in the entire topology. BAS controller first triggers *pathChirp* algorithm to search and estimate the bandwidth of all available networks. After determining and selecting the best path in the current topology, the bandwidth estimation value is provided to Proportional-Integral-Derivative (PID) rate controller which tries to regulate the rate of both live stream scheduler, adaptive encoder and the network.

It is important to find a solution for the problem of streaming path allocation in a multi-path network, and it is almost perfect to take media aware metrics into account during the decision process. There were some previous works [10] about path selection methodology based on experimental data. These are some rules that consider network metrics such as available bandwidth, loss rate and hop distance, and media aware metrics such as link connections and video distortion. It will be convenient to consider media-specific metrics to provide a more general framework for the analysis of joint

path selection and rate allocation. It will be also logical to consider a multi-path network model that supports multiple media flows, and a streaming server that can adapt the media source rate to the transmission conditions. A generic video distortion metric can be proposed, which encompasses both the source distortion (mostly driven by the encoding rate), and the channel distortion, dependent on the loss probability [2].

## 2) Rate Control and Rate Allocation

It is possible to use different type of controllers as rate controller. We consider traditional Proportional-Integral-derivative (PID) Controller and Neural Network Predictive Controller. These algorithms are based on control engineering theory and implementation.

### a. PID Controller as Rate Controller

General structure of PID controller is given Fig.6. PID controller is a traditional controller [11] that mostly used position, motion, speed and process control, and it gives good enough performance as a rate controller [12]. Input of the PID controller is the error between the reference value and the system output. We can determine a rate as a reference value and we can compare this rate with estimated bandwidth value that is provided by bandwidth estimation tool, and we try to find new rate value. BAS controller should be supported by a parameter optimization tool. This would be necessary to obtain controller parameters which make the controller give the best performance values.

There is only a few control engineering oriented solutions are available in the literature. PD (proportional-derivative) controller is first used as application-level feedback controller with conventional Earliest Deadline First-EDF scheduler in control-theoretical approach of EDF called BC-EDF (Buffer-Controlled EDF) [12]. It is proposed that control-theoretic buffer-occupancy feedback control with an end-to-end feedback scheduling mechanism is used together to achieve real-time streaming. Thus, the playback quality on the client side is improved. On the other hand, this performance is relatively poor, if it compares with RTCP feedback based transmission rate control approach for wireless media streaming [13]. Three major problems (client buffer underflow and client buffer overflow prevention, and network buffer congestion control) are considered, extended RTCP feedback is used for solution. The sequence number of the last transmitted packet, Highest Transmitted Sequence Number (HTSN), is considered and RTCP feedback based transmission rate control algorithm is produced. The results are compared with BC-EDF algorithm, and the buffer fill level and overflow conditions are relatively reduced in RTCP feedback based approach.

In this study, both PID controller and RTP/RTCP feedback based approach are taken into consideration for better video streaming performance. Internal structure of PID controller is given in Fig. 4. PID controller is simple, but it is easy to use and effective controller in control engineering. Different types of PID controllers are proposed such as fuzzy PID, auto-tuning PID, self-tuning PID etc. so far, but we consider

default structure. PID controller characterized by the equation of

$$u(t) = (K_p + K_i \int e(\tau) d\tau + K_d \frac{de(t)}{dt}) e(t)$$

where  $u(t)$  is control signal,  $e(t)$  is error signal between system output  $y(t)$  and the reference signal  $u(t)$  and describes as  $e(t) = y(t) - r(t)$  and  $K_p$ ,  $K_i$ , and  $K_d$  are proportional, integral and derivative parameters of PID controller respectively.

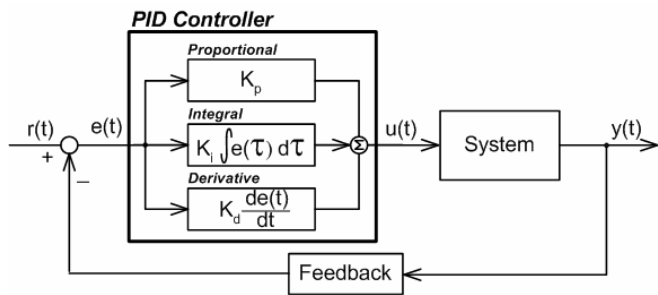


Fig. 4 General Structure of PID Controller in a control system

Performance optimization of PID controller is provided in different ways. Most of them are experimentally obtained. In this work, global optimization method is used for optimization. This is a simple method and yielding the idea of preparing a look up table for PID parameters obtained in different reference and performance values. Parameter optimizer easily selects the parameter value set ( $K_p$ ,  $K_i$ , and  $K_d$ ) which gives the best performance. The performance index of such as system is given by

$$J = \int_0^{\infty} e^2(t) dt$$

under the assumption of integral of squared-error (ISE). The value of  $J_n$  can be minimized adjusting the PID parameters,

$$J_n = \min_{K \text{ stabilizing}} J_n(K)$$

where  $K$  is the function of  $K_p$ ,  $K_i$  and  $K_d$  and represented as  $K = \{K_p, K_i, K_d\}$ . That means we try to find convenient  $K_p$ ,  $K_i$  and  $K_d$  which makes performance index function  $J$  minimized.

As an adaptive algorithm, Fuzzy Logic Controller scheme with feed-forward scaling factors was also used in fuzzy-logic based video rate control [14].

### b. Neural Network Predictive Rate Controller

A Neural Network (NN) approach that supported by a predictive control algorithm may provide a relatively better solution for rate control. NNs become the focus of great attention, because of its capacity to treat complex and nonlinear problems. The Internet has also a heterogeneous

structure. It is therefore possible to adapt NN approach to video streaming rate control. Backpropagation algorithm is mostly used for learning about the network topology and it can easily be applied as:

1. Initialize weights to small random values.
2. Propagate the input signal forward throughout the network.
3. Compute the sensitivity function for each weight of the network.
4. Update weights.
5. Go to step 2 and repeat the procedure until end criteria is reached.

Predictive control was established as a powerful technique, it became a potential for applications [15]. The idea is based on input and output past information, and then future control actions is predicted along a moving prediction horizon. The algorithm includes the following steps:

1. Predict the system output in period of time via Taylor series,
2. Obtain control signal series by minimizing the error between system output and future system output by considering constraints,
3. Select the first element in the series as a control signal and repeat same procedure in next future period of time,

Neural Network Predictive controller is a joint solution of backpropagation algorithm and moving horizon predictive algorithm. The general structure of NN Predictive Controller is given in Fig. 5.

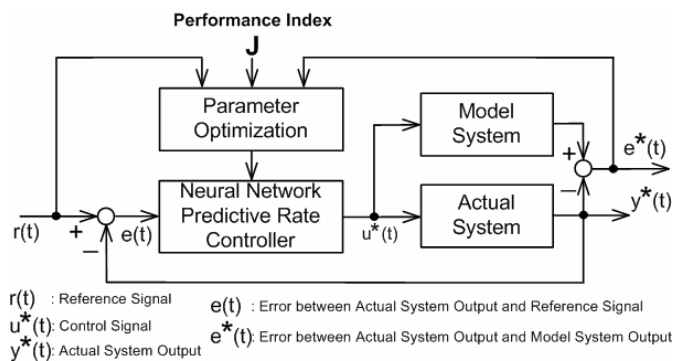


Fig. 5 Neural Network Predictive Rate Controller Principle

Finding the optimal rate allocation in multi-path networks is in general a complete problem in generic scenarios. However, we show that a careful analysis of the video distortion evolution allows deriving a linear complexity algorithm for the joint optimal path selection, and flow rate allocation under common network assumptions. In other words, our main objective is to jointly find:

- (i) The optimal encoding or streaming rate of a video stream so that the quality at receiver is maximized,
- (ii) Which network paths should be used for relaying the video stream to the client?

Interestingly enough, our conclusions demonstrate that the answer to these two questions is represented by a careful tradeoff among available network bandwidth (translated into video encoding rate), transmission loss process, and number of utilized paths. And, contrary to the commonly admitted opinion, flooding the network in using all the possible paths rarely provides an efficient strategy. Main findings can be briefly summarized as follows:

- A general framework for media streaming analysis in multi-path networks, which encompasses network and media aware metrics has been done,
- The first theoretical analysis on the optimality of number, and choice of network paths, in terms of media streaming QoS has been presented,
- A linear time media aware routing algorithm has been provided. The outputs of this algorithm give the optimal set of network paths to be used in the streaming process, along with the corresponding rate distribution.

One of the most important points for developing such a method is a clustering problem that detect shared bottleneck. The details of the algorithm can be found in [2]. It is also very important to mention that bandwidth estimation tools that are described in Section 3 can also be used for detecting bottlenecks and end-to-end bandwidth of a network.

#### B. QoS Controller: Packet Loss Observer and Delay Controller

QoS controller tries to guarantee the service quality at the receiving side by considering RTT for calculating the delay and sequence number for observing packet loss. QoS controller needs delay and packet loss values to assure QoS at receiving client. Packet loss observation and delay control can be handled within the framework of RTCP protocol.

In the proposed structure, QoS controller uses the feedback values of Round Trip Time-RTT and sequence number which are the feedback metrics of upper-level transport layer protocols *RTP/RTCP*. RTP and RTCP protocols are used for video streaming and QoS controller calculates delay by using RTT and packet loss by using sequence number and tries to make the video streaming much more efficient and makes the video quality reasonable at the receiver side for the available conditions. Both outputs coming from BAS Controller and the QoS Controller are sent to Decision Center. Decision Center runs an optimization algorithm to calculate the best rate to send to adaptive scheduler and adaptive encoder as seen in Fig. 6.

We have used Real-time Transport Protocol-RTP and Real-time Transport Control Protocol-RTCP protocols for transmitting the video streaming signals. RTP and RTCP are described by RFC 3550, and Extended RTP profile for RTCP-based feedback is described by RFC 4586. There are some clues in RFC 3550 such as

- RTP provides end-to-end delivery services for data with real-time characteristics, such as interactive audio and video.
- RTP does not provide quality of service guarantees, but relies on lower-layer services to do so.
- RTP header contains timing information and a sequence number that allow the receivers to reconstruct the timing produced by the source. The sequence number can also be used by the receiver to estimate how many packets are being lost.

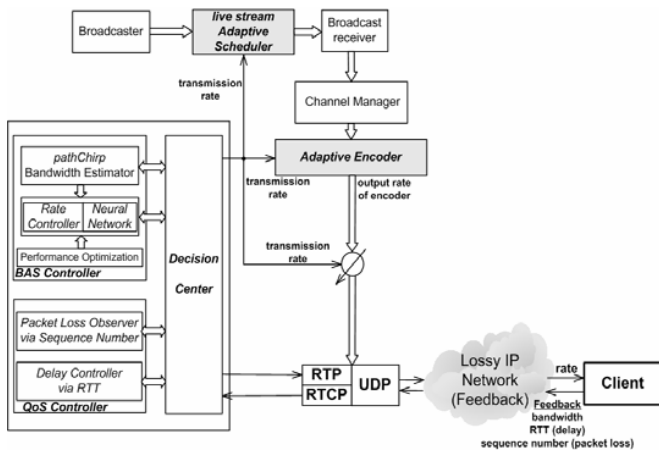


Fig. 6 BAS Controller, QoS Controller and Decision Center with Encoder and Transmitter

- There is no single congestion control mechanism that will work for all. Therefore, congestion control SHOULD be defined in each RTP profile as appropriate.
- RTCP feedback used more or less 5% of the session bandwidth, so it behaves in a network-friendly manner.
- It is possible to implement encoding and profile independent quality monitors.
- It is possible to calculate packet loss rate.
- It is possible to calculate the average payload data rate and average packet rate over an interval.
- It is possible to calculate a second short-term measure of network congestion by inter-arrival jitter.
- It is possible to make jitter calculation by RTP timestamp.
- It is possible to calculate delay (round-trip propagation).

There are also some more details in the documents, but these are the most important for our video streaming architecture. We consider packet loss and delay for maximizing quality of service. As we mentioned above, it is possible to calculate both packet loss ratio and delay.

According to RFC 2550 packet loss rate can be calculated as “The difference in the cumulative number of packets lost gives the number lost during that interval. The difference in the extended last sequence numbers received gives the number of packets expected during the interval. The ratio of these two is the packet loss fraction over the interval. The lost rate per

second can be obtained by dividing the loss fraction by the difference in NTP timestamps, expressed in seconds. The number of packets received is the number of packets expected minus the number lost.”

According to RFC 2550 round-trip propagation delay can be calculated as “RTCP sender and RTCP receiver report contain Delay since Last Sender Report-DLSR field. Source can compute the round-trip propagation delay to receiver by recording the time A when this reception report block is received. It calculates the total round-trip time A-LSR using the Last Sender Report timestamp (LSR) field, and then subtracting this field to leave the round-trip propagation delay as A-LSR-DLSR. This may be used as an approximate measure of distance to cluster receivers, although some links have very asymmetric delays.”

### C. Decision Center and Optimal Rate Determination

Decision Center is the heart of this architecture, because all controller information is gathering here. It should therefore have a mechanism to determine optimal rate by considering Bandwidth Adaptive Streaming Controller output (rate ratio) and QoS Controller outputs (delay and packet loss). Decision center also considers a performance index function. The constraints are also rate ratio, delay and packet loss. If other metrics are necessary such as jitter and congestion, it is also possible to calculate such these values by RTP/RTCP sender-receivers reports.

In this work, the aim is to minimize transmission cost while limiting delay and packet loss. The transmission packet loss (L), delay (D) and rate (R) are constraints for transmission cost function. We have to consider encoder (E) parameters and network (N) parameters. If performance index is characterized by one of these three costs,

$$J = \int_0^{\infty} |e(t)| dt, \quad J = \int_0^{\infty} e^2(t) dt \quad \text{or} \quad J = \int_0^{\infty} te(t) dt$$

we try to obtain the minimized cost function as

$$\min J(E, N)$$

We have the constraints as

$$L(E, N) \leq L_0 \quad \text{Packet Loss}$$

$$D(E, N) \leq D_0 \quad \text{Delay}$$

considering Lagrange multipliers as  $J(E, N) + aL(E, N) + bD(E, N)$  where a and b weighting parameters, and try to find a solution for this performance index by Lagrange Optimization Method.

### V. A SOFTWARE INFORMATION MODEL OF BANDWIDTH ADAPTIVE VIDEO STREAMING

A software information model of bandwidth adaptive video streaming system is given in Fig. 7. This model includes Broadcaster Component, Streaming Server Component and



Clients Component. All the methods regarding to component functionalities are shown inside each component. All components run a middleware bus software which enables transparent distributed communication between them. Now it is time to explain these components one by one.

Broadcaster Component captures the video and audio signals separately, and then encodes and encrypts these signals in convenient forms. At this point, data is ready to transmit to a streaming server over convenient transport algorithm such as RTP over UDP or TCP. If needed, this data can be stored at the repository for archiving. This repository can be jointly used by a broadcaster and/or streaming server components.

The Streaming Server Component has the most complex, and at the same time an integrated structure. This component includes BAS-Bandwidth Adaptive Streaming Controller, QoS-Quality of Service Controller and a Decision Center subcomponents. BAS includes Bandwidth Estimator, Predictive Rate Controller and Performance Optimizer components. Bandwidth Estimator object is used for determining available bandwidth by means of a bandwidth estimation algorithm called *pathchirp* [8]. It also regulates encoder and scheduler for the received video information. Predictive Rate Controller object predicts transmission rate and forwards broadcasts. Performance Optimizer object tries to find the best network path to destination. It also maximizes transmission rate to Client Components. QoS Controller subcomponent contains two important objects, Packet Loss Observer and Delay Controller. Using an RTP feedback

protocol RTCP with its frame fields, Packet Loss Observer object calculates packet loss ratio and average distortion. Delay Controller object calculates variable packet delay and acceptable packet loss. Decision Center of Streaming Server Component is the heart of this software information model which has many tasks to control and regulate the streaming. Decision Center receives BAS Controller and Qos Controller information and processes them to generate the following important functionalities: get predicted transmission rate, select performance index, get optimized transmission rate, get optimized network path, get packet loss ratio, get delay time, regulate transmission rate, receive broadcast, forward broadcast, and choose convenient transport protocol and then send it to destination client components.

Clients Component on the client side is the main component which generates client video requests, sends them to Streaming Server, and receives processed streaming video from Streaming Server Component. It requests live or video on demand streaming, receives their broadcasts, chooses transport protocol, requests streaming server port, chooses decryption standard, as well as supplying rate feedback to bandwidth estimator. All communication between these components are seamlessly accomplished through the middleware bus.

## VI. RESEARCH CONTRIBUTIONS AND CONCLUSIONS

In this work, first, an overview of both video streaming methodology and bandwidth estimation tools are given, and

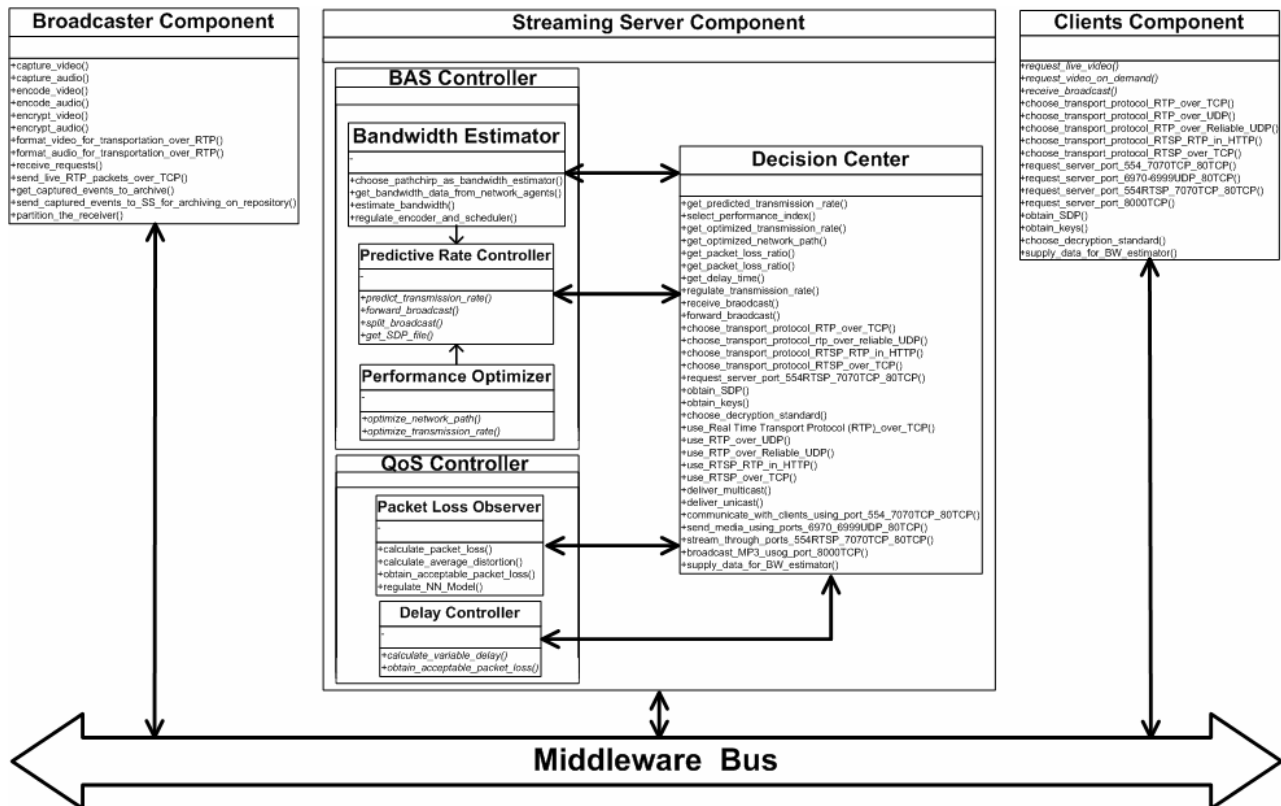


Fig. 7 Software Information Model of a Bandwidth Adaptive Video Streaming

bandwidth adaptive video streaming architecture is introduced. This streaming server architecture includes Bandwidth Adaptive Streaming (BAS) Controller, QoS Controller and Decision Center components and their related objects. While Bandwidth Adaptive Streaming Controller tries to obtain the best path, an acceptable rate and optimized rate allocation by making use of *pathChirp* bandwidth estimation tool, the QoS controller calculates delay and packet loss values. A novel and carefully designed Decision Center determines the path, and optimizes the rate under the delay and packet loss constraints, by means of client produced RTCP feedback values. A global optimization method is used for optimizing parameters. A neural network predictive controller is used for video rate control. We have developed a complete distributed software information model with Broadcaster Component, Streaming Server Component and Client Components communicating over a middleware bus. PID (Proportion Integral Derivative) controller, Neural Network Predictive controller and in an all-in-one solution within our Integrated Software Architecture is now being implemented by using *ns-2* Network Simulator [16].

## REFERENCES

- [1] S. Savage, A. Collins, E. Hoffman, "The end-to-end effects of internet path selection," in Proceedings of ACM SIGCOMM, 1999, pp. 289-299.
- [2] D. Jurca and P. Frossard, "Media-Specific Rate Allocation in Multipath Networks", Signal Processing Institute Technical Report TR-ITS-2005.032, 2005.
- [3] J. Chakareski and P. Frossard, "Distributed Streaming Via Packet Partitioning", Proceedings of IEEE ICME, July 2006.
- [4] J.P. Wagner, J. Chakareski and P. Frossard, "Streaming of Scalable Video from Multiple Servers using Rateless Codes". Proceedings of IEEE ICME, July 2006.
- [5] R. S. Prasad, M. Murray, C. Dovrolis and K. Claffy, "Bandwidth estimation: metrics, measurement techniques and tools", IEEE Network, Nov 2003, Vol. 17, No. 6, pages 27-35.
- [6] R. L. Carter and M. E. Crovella. Measuring Bottleneck Link Speed in Packet-Switched Networks. In Performance Evaluation, 27,28:297-318, 1996.
- [7] M. Jain and C. Dovrolis, "Pathload: A measurement tool for end-to-end available bandwidth", In Proceedings of Passive and Active Measurements (PAM) Workshop, Mar. 2002.
- [8] V. Riberio, R. Riedi, R. G. Baraniuk, J. Navratil, and L. Cottrell, "pathChirp: Efficient Available Bandwidth Estimation for Network Paths" Passive Active Measurement Workshop -- PAM2003 (Apr. 2003).
- [9] D. Katabi and C. Blake, "Inferring Congestion Sharing and Path Characteristics for Packet Interarrival times", MIT-LCSTR - 828, December 2001.
- [10] J. Apostolopoulos, T. Wong, W. Tan, and S. Wee, "On multiple description streaming with content delivery networks," in *Proceedings of IEEE INFOCOM*, vol. 3, 23-27 June 2002, pp. 1736-1745.
- [11] K. J. Astrom, and T. Hagglund, Tore, "PID Controllers: Theory, Design, and Tuning", Research Triangle Park, North Carolina, Instrument Society of America, 1995.
- [12] C. Wang, J. Ho, R. Chang, S. Hsu, "A feedback-Controlled EDF Scheduling Algorithm for Realtime Multimedia Transmission", Technical Report TR-IIS- 01-008, Institute of Information Science, Academia Sinica, Taipei, Taiwan, 2001.
- [13] N. Baldo, M. Kampmann, U. Horn, and F. Hartung, "RTCP Feedback Based Transmission Rate Control For 3g Wireless Multimedia Streaming", IEEE International Symposium on Personal, Indoor and Mobile Radio Communications 2004 (PIMRC2004), Barcelona, Spain, September 2004.
- [14] Y. Saw, P. M. Grant, and J. M. Hannah, "A Comparative Study of Nonlinear Video Rate Control Techniques: Neural Networks and Fuzzy Logic", IEEE, 1998.
- [15] E. Ronco, T. Arsan and P. J. Gawthrop, Open-loop Intermittent Feedback Control: Practical Continuous-time GPC, IEE Proceedings of Control, Theory and Applications, Vol.146, Issue 5, p.426-434, September 1999.
- [16] K. Fall, K. Varadhan (Editors), The ns Manual (formerly ns Notes and Documentation), The VINT Project A collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC.

**Taner Arsan** was born in Turkey, on 05.12.1968. He received the Electronics and Communication Engineering degree from ITU-Istanbul Technical University, Istanbul, Turkey, in 1990. He completed his M.Sc. and Ph.D. thesis at Control and Computer Engineering Department of ITU, in 1994 and 1999, respectively. He had been to Centre for Systems and Control - University of Glasgow (Scotland-UK) between September 1997 and July 1998 as a visiting scholar for his Ph.D. thesis research. He worked as Research and Teaching Assistant at Control and Computer Engineering Department of ITU from 1991 to 2000.

In April 2000, Dr. Arsan became an Assistant Professor at the Computer Engineering Department of Kadir Has University, Istanbul, Turkey. He is also Advisor to the President of Kadir Has University and Vice Chairman of Computer Engineering Department of Kadir Has University.

His research areas are Self-tuning Algorithms, Microcontroller-based Auto-tuning System Implementation, Generalized Predictive Control and its Applications, Software Architecture for Neural Network Systems, Computer Networks Education and Video Streaming over Lossy IP Networks.