Multimed Tools Appl
DOI 10.1007/s11042-016-3623-8

Analysis and modelling of traffic produced by adaptive HTTP-based video

Arkadiusz Biernacki[1]

Received: 26 October 2015 / Revised: 28 March 2016 / Accepted: 16 May 2016

Abstract The increase of HTTP-based video popularity causes that broadband and Internet service providers' links transmit mainly multimedia content. Network planning, traffic engineering or congestion control requires an understanding of the statistical properties of network traffic; therefore, it is desirable to investigate the characteristic of traffic traces generated by systems which employ adaptive bit-rate streaming. Our first contribution is an investigation of traffic originating from 120 client-server pairs, situated in an emulated content distribution network, and multiplexed onto a single network link. We show that the structure of the traffic is distinct from the structure generated by the first and second generation of HTTP video systems, and furthermore, not similar to the structure of general Internet traffic. The obtained traffic exhibits negative and positive correlations, anti-persistence, and its distribution function is skewed to the right. Our second contribution is an approximation of the traffic by ARIMA/FARIMA processes blue and artificial neural networks. As we show, the obtained traffic models are able to enhance the performance of an adaptive streaming algorithm.

Keywords Multimedia service management · Performance management · Stochastic processes · IP networks

1 Introduction

During the past years, web-based video sharing services like YouTube, Hulu or Dailymotion have become very popular. The users of YouTube, which allows for the distribution of user-produced multimedia content, alone request millions of videos every day [1]. Consequently,

✉ Arkadiusz Biernacki
arkadiusz.biernacki@polsl.pl

1 Institute of Computer Science, Silesian University of Technology, Akademicka 16, 44-100, Gliwice, Poland

a popularity of this kind results in a drastic shift in Internet traffic statistic, which reports an increase in traffic from Web-based video sharing services [2].

Video streaming in the above-mentioned services is HTTP-based; therefore, being transported using TCP. HTTP and TCP are general purpose protocols and were not primarily designed for streaming of multimedia. Thus, attempts are being made to adapt the delivery of multimedia content to the Internet environment. One of such attempts tries to introduce an additional layer of application control to transmitted video traffic [3]. Since TCP is designed to deliver data at the highest available transmission rate, it may sometimes be reasonable for a sender to provide additional flow control if it is not strictly necessary for application data to reach a receiver as fast as TCP would otherwise allow. Therefore, the application may limit the rate at which data is passed to a network stack for transmission, and, if the video bit rate is lower than the end-to-end available bandwidth, the traffic characteristic will differ from a standard TCP flow. In order to limit data rate, modern video players produce ON-OFF cycles, where during the ON time a block of data is transferred at the end-to-end available bandwidth that can be used by TCP, and during the OFF time, the TCP connection remains idle. Furthermore, the players implement stream-switching (or multi-bit-rate): the content, which is stored at a web server, is encoded at different bit-rate levels, then an adaptation algorithm selects the video level, which is to be streamed, based on a state of a video player, for example on the length of the player buffer, or on the state of network environment, for example on the amount of available bandwidth [3].

An HTTP server usually transmits multiple streams simultaneously. Such a situation takes place when, for instance, there are several concurrent sessions initiated by clients located within an Internet Service Provider (ISP) network, as presented in Fig. 1. In such a scenario, the overlapped ON-OFF cycles, TCP reliability and retransmission mechanisms in combination with application level flow control lead to a new network traffic characteristic, which may be significantly different from classical models of aggregated traffic where traffic is depicted as a stochastic process with a positive and slowly decaying autocorrelation function [4]. Taking into account that the traffic from adaptive video services embedded in web pages already takes an important part of all traffic, the above scenario is quite possible.

Therefore, in this work we investigate the characteristic of aggregate network traffic produced by 120 server-client pairs. We tried to estimate the statistical properties of the traffic, examining for this purpose its distribution, autocorrelation, and possible self-similarity what is the first contribution of our work.

Statistical analysis of Internet traffic is important for, e.g.: network planning, resource allocation or detection of network anomalies. Therefore, the obtained results may be useful to video content providers or ISPs, which have an access to traffic management withing their networks. They may apply the knowledge about the traffic characteristic, for instance,

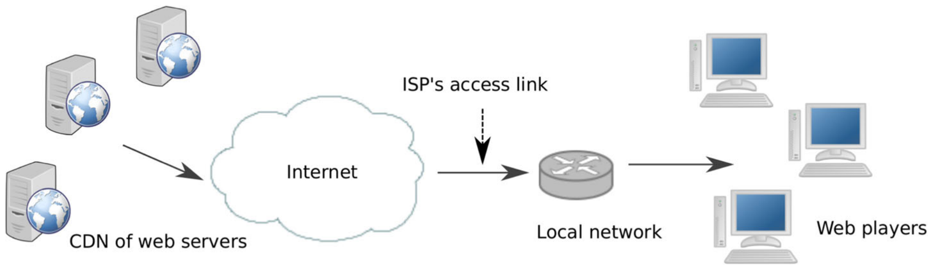


Fig. 1 HTTP streaming scenario. Video traffic is multiplexed on the ISP access link

for predicting the quality of service degradation caused by congestions or for adjusting the control rate algorithms. Thus, ISPs may, to a certain extent, prevent play-out stalls caused by buffer under-runs of a video player, reduction of a video bit-rate or frequent switching between different bit-rates. Also developers of video players may consider enhancing their play-out algorithms with predictions of network traffic [5–7].

To demonstrate a practical application of our statistical analysis, we approximated the traffic trace with ARIMA/FARIMA processes and with artificial neural networks (ANNs). We applied the obtained models to enhance the performance of an adaptive streaming algorithm. We show that the streaming algorithm obtains better performance using the estimations provided by the models than relaying only on the past network measurements blue what is the second contribution of our work.

We conduct the study using an emulation model which allows us to methodologically explore the behaviour of the examined system over a wide range of parameter settings, which would be a challenging task to conduct such experiments only on a real-network. Simultaneously, as the emulation is performed in a laboratory environment, we are able to preserve much of the network realism because we conduct experiments, using real hardware and software, which permits us to maintain a decent level of accuracy for the obtained results.

2 Theoretical background

2.1 Application level flow control

One of the popular video transmission methods is the progressive download, which is simply a transfer of a video file from an HTTP server to a client where the client may begin playback of the file before the download is completed. However, as the HTTP server progressively sends (streams) the whole video content to the client and usually does not take into account how much of the data has been already sent in advance, an abundance of data can overwhelm the video player and lead to a large amount of unused bytes if a user interrupts the video play-out [8]. To avoid such an undesirable situation, the video file is divided into chunks of fixed length and the server pushes them sequentially to the client at a rate little higher than the video-bit rate of the transmitted content. As a result, the transmitted traffic creates an ON-OFF pattern, where ON and OFF periods have a constant length.

The extension of this idea is an adaptive streaming, which offers more flexibility when a network environment is less stable, e.g. in wireless mobile networks. With this approach, it is possible to switch the media bit rate (and hence the quality) after each chunk is downloaded and adapt it to the current network conditions [9]. This technique has commenced the development of a new generation of HTTP-based streaming applications which implement client-side play-out algorithms, trying to deliver a continuous stream of video data to an end user by mitigating unfavourable network conditions. In this approach, a video stream is also divided into segments, but this time, they are encoded in multiple quality levels, called representations. Based on an estimation of the available throughput, the client may request subsequent segments at different quality levels which depend on network conditions between the client and server, as drafted in Fig. 2. The algorithm deciding which segment should be requested in order to optimize the viewing experience is the main component and a major challenge in adaptive streaming systems because the client has to properly estimate, and sometimes even predict, network conditions or the dynamics of the available throughput. Furthermore, the client has also to control a filling level of its local buffer in order to

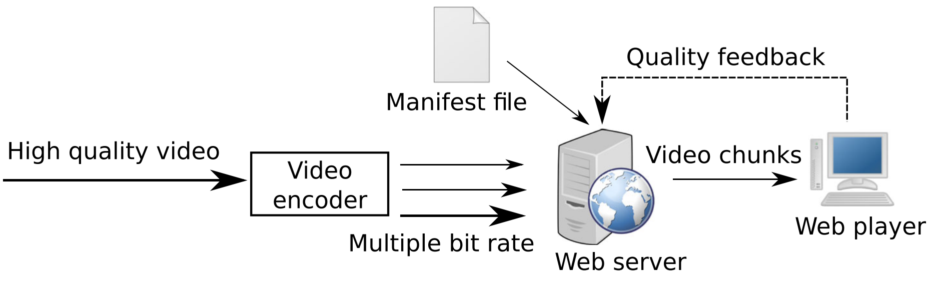


Fig. 2 Architecture of a video adaptive system based on HTTP

avoid underflows resulting in playback interruptions. Simultaneously, users expect usually the best possible quality of the stream, while avoiding unnecessary quality fluctuations, and minimisation of the delay between their requests and the start of a playback. The transmitted traffic also creates an ON-OFF pattern, but this time, the duration of the ON and OFF periods is not constant anymore, but is a discrete random variable, which values are dependent on the logic of the play-out algorithm.

The stream-switching technique is employed today less or more in some proprietary video players, among others in the Apple HTTP-based streaming [10], Microsoft IIS Smooth Streaming [11] or Adobe Dynamic Streaming [12]. Moreover, the technique is also adopted by the Dynamic Adaptive Streaming over HTTP (DASH), which is a MPEG standard pursuing the interoperability between devices and servers of various vendors [3].

As the above adaptive video systems do not share many details about the employed algorithms, in our further experiments we implemented several adaptive algorithms found in the literature.

The first implementation uses the Microsoft Smooth Streaming (MSS) algorithm, which is based on the open source version of the algorithm of the MSS video player and is extensively described in [13].

The second heuristic relies on the *PANDA* (Probe AND Adapt) algorithm proposed in [14]. The algorithm takes TCP download throughput as an input to the adaptive algorithm only if the measurement is an accurate indicator of the fair-share bandwidth. The video quality is not directly related to network bandwidth but to its average, which in turn determines the selected video bit-rate and the request time between video chunks.

The third implementation is based on the heuristic proposed in [7], which is a hybrid receiver-driven adaptation algorithm taking into account a buffer filling level and throughput estimations.

The last approach, referred in the literature as *Festive*, is based on the implementation described in [15]. *Festive* uses randomized scheduling and bit-rate selection based on the state of the video player.

The above algorithms were implemented in the open-source software described in [16].

2.2 Multiplexed traffic and its characteristic

The packets of the different video flows are usually placed on a single physical link – for example, a packet is transmitted for one connection, then a packet for a second, another for the first, then two packets in a row for a third, and so forth. This intermingling is referred to as “statistical multiplexing” in the packet network literature or as “superposition” in the mathematical literature of stochastic processes.

As it was shown in many research works, statistical properties of the stochastic process which forms the network traffic have significant influence on the network functionality and efficiency. Network performance, as captured by throughput, packet delay and loss rate, degrades gradually with increased long range dependence (LRD) or self-similarity of aggregated traffic [17]. Simultaneously, network routers and other middle-boxes servicing such traffic experience larger queueing delay and response time [18]. Therefore, from the traffic engineering point of view, an estimation of the degree of LRD for the examined video traffic may be useful, e.g. for an assessment of video quality received by users or for determining how well congestion control is able to shape the traffic into an on-average constant output stream while conserving information.

The amount of multiplexed traffic sent within a certain time period is represented in our work as a counting process. From a mathematical point of view, let N_t be a stochastic process which values represent a cumulative number of bytes sent until time t. Assuming that $N_t > 0$ and $N_t \in \mathbf{I}$, we are interested in the difference

$$C_n = N_{t+\Delta t} - N_t, \tag{1}$$

which is to be interpreted as a number of bytes sent in the interval $[t, t + \Delta t]$.

For the above described stochastic process, we computed the distribution function, autocorrelation and the degree of LRD.

2.3 Long-memory processes

There are several way of characterising long-memory processes. A widespread definition is in terms of the autocorrelation function $\gamma(k)$. We define a process as long-memory if for $k \to \infty$

$$\gamma(k) \sim k^{-\alpha} L(k), \tag{2}$$

where $0 < \alpha < 1$ and $L(k)$ is a slowly varying function at infinity. The degree of long-memory is given by the exponent α; the smaller α, the longer the memory. By contrast, one speaks of short range dependent process if the autocorrelation function decreases at a geometric rate and $\alpha > 1$.

Long-memory is also discussed in terms of the Hurst exponent H, which is simply related to α from (2). For a stochastic process, $H = 1 - \alpha/2$ or $\alpha = 2 - 2H$. When $H \in (.5, 1]$, the process is positively correlated which implies that it is persistent and is characterised by long-memory effects on all time scales, i.e. the realisation of the process has been up or down in the last period then the chances are that it will continue to be up or down, respectively, in the next period. On the other hand, when $H \in [0, 0.5)$, we have long-term anti-persistence which means that whenever the realisation of the process has been up in the last period, it is more likely that it will be down in the next period. When $H = 1/2$, and the autocorrelation function decays faster than k^{-1}, then the process has no memory.

While the Hurst parameter is perfectly well-defined mathematically, it may be a difficult property to measure in real life. Tests for the LRD usually require a considerable amount of data because the measurement should be done at tails of a distribution, where not so much data are available. Furthermore, different methods of the Hurst parameter estimation often give inconclusive or even contradictory results. The assessment results may be biased by trends, periodicity and corruptions in the data. Therefore, some authors suggested to apply a “portfolio” of estimators instead of relying on a single estimator, which could give a misleading assessment caused by properties of the process under investigation [19]. Thus, in this paper, we employ three widespread, well-known techniques, which have been used for some time, to estimate the Hurst exponent: R/S, Aggregated Variance (AV) and Differenced Aggregated Variance (DAV). All the chosen techniques have freely available code and are implemented, among others, in Rmetrics software [20], which is a part of the Cran R environment [21].

The empirical investigation of LRD processes must take into account the presence of high-frequency autocorrelations. When doing a statistical analysis, it is therefore important to try to eliminate or, at least minimize, such linear dependencies, since it can bias the Hurst parameter and classify a process as having a long-term "memory" when it is, in reality, a short-term "memory" effect. In order to eliminate, or at least reduce, the linear dependencies, the autocorrelation and the estimation of the Hurst parameter were obtained for the de-trended counting process C_n (1) computed as

$$R_n = ln(C_{n+1}/C_n), \tag{3}$$

where Δt in (1) was set to 0.1 s.

3 Traffic prediction

In the common approach to the problem of network bandwidth prediction (and other time series in general), we have the measurements of previous bandwidth $\{x_t, x_{t-1}, \dots, x_{t-(n-1)}\}$ and we wish to predict the value of x_{t+m}, $m > 0$. For this purpose, we apply the predictor $\hat{x}_{t+m}$ which is based on the observations of the past bandwidth measurements and may be written as

$$\hat{x}_{t+m} = \phi(x_t, x_{t-1}, \dots, x_{t-(n-1)}). \tag{4}$$

The problem is to choose ϕ so that $\hat{x}_{t+m}$ is, in some sense, "closest" to x_{t+m}.

Conventionally, the video adaptive algorithm uses prediction of network bandwidth for the nearest future x_{i+1}. The prediction is usually an average of past bandwidth measurements:

$$\hat{x}_{t+1} = \frac{1}{n}\sum_{k=0}^{n-1} x_{t+k}. \tag{5}$$

This estimator (5) is applied in the base version of adaptive play-out algorithms [9, 14]. In our work, we try to improve the prediction employing (F)ARIMA and ANNs models.

We adopt the usual measure of closeness, namely root-mean-square error (RMSE) which is a statistical indicator for assessing goodness of different parameters of the model. The RMSE represents a sample standard deviation of the differences between the predictor $\hat{x}_{t+m}$ and the observed values x_{t+m}. Mathematically, it can be described as

$$RMSE = \sqrt{\sum_{t=1}^{N}(x_{t+m} - \hat{x}_{t+m})^2/N}, \tag{6}$$

where N denotes the number of predictions. The lower is the RMSE value, the better the model describes empirical data.

3.1 ARIMA and FARIMA models

ARIMA processes are considered as a classical and universal modelling tools. They can capture the linear and short-range dependencies occurring in time series, but when used for general Internet traffic which exhibits LRD, the ARIMA processes have rather poor performance. Nevertheless, due to their versatility and relative easiness of use, they are extensively applied for modelling and forecasting of network traffic, e.g. [22, 23]. In order to capture LRD of network traffic and obtain better accuracy of a model, some research use FARIMA processes [24, 25]. However, the FARIMA processes are computationally more complex and require more elaborate algorithms for estimation of their parameters.

A time series $\{x_t\}$ is an autoregressive (AR) process of order p, if

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \ldots + \alpha_p x_{t-p} + w_t \tag{7}$$

where $\{w_t\}$ is white noise and the α_i are the model parameters.

Applying the backward shift operator

$$\mathbf{B}^n x_t = x_{t-n},$$

Equation (7) can be expressed as a polynomial of order p

$$\theta_p(\mathbf{B}) x_t = (1 - \alpha_1 \mathbf{B} - \alpha_2 \mathbf{B}^2 - \ldots - \alpha_p \mathbf{B}^p) x_t = w_t \tag{8}$$

A moving average (MA) process of order q is a linear combination of the current white noise term and the q most recent past white noise terms, which is defined by

$$x_t = (1 + \beta_1 \mathbf{B} + \beta_2 \mathbf{B}^2 + \ldots + \beta_q \mathbf{B}^q) w_t = \phi_q(\mathbf{B}) w_t \tag{9}$$

When AR (8) and MA (9) terms are added together in a single expression, we obtain a time series $\{x_t\}$ which follows an autoregressive moving average (ARMA) process of order (p, q), denoted as ARMA(p,q)

$$\theta_p(\mathbf{B}) x_t = \phi_q(\mathbf{B}) w_t, \tag{10}$$

where θ_p and ϕ_q are polynomials of orders p and q, respectively.

A series $\{x_t\}$ is integrated of order d, if the dth difference of $\{x_t\}$ is white noise $\{w_t\}$, i.e.

$$(1 - \mathbf{B})^d x_t = w_t.$$

A time series $\{x_t\}$ follows an ARIMA(p,d,q) process if the dth differences of the $\{x_t\}$ series are an ARMA(p,q) process (10). If we introduce $y_t = (1 - \mathbf{B})^d x_t$, then $\theta_p(\mathbf{B}) y_t = \phi_q(\mathbf{B}) w_t$.

We can now substitute for y_t to obtain the more succinct form for an ARIMA(p,d,q) process as

$$\theta_p(\mathbf{B})(1 - \mathbf{B})^d x_t = \phi_q(\mathbf{B}) w_t. \tag{11}$$

A fractionally differenced ARIMA process $\{x_t\}$, FARIMA(p,d,q), has the form of (11) for some $-\frac{1}{2} < d < \frac{1}{2}$.

3.2 Artificial neural networks

ANNs have been successfully applied in many disciplines, including time series prediction. Similarly to ARIMA and FARIMA processes, an ANN for prediction of time series uses a sliding window containing n most recent measurements, see (4). However, unlike linear ARIMA and FARIMA, ANNs can handle also non-linear phenomena in time series [26, 27]. Thus, ANNs have been also widely applied for network traffic modelling and prediction [28, 29].

There have been a number of different architectures for ANNs, however for our purposes, we employ only multilayer perceptrons (MLP) network and radial basis function (RBF) network with a single hidden layer. As we will show, even these basic architectures allow to achieve a quite decent approximation of network traffic and a fair improvement of an adaptive video play-out. In both mentioned architectures, each of the layers is fully connected to the next one. In the case of the MLP network, we apply commonly used logistic function

$$\sigma(x) = \frac{1}{1+e^{-x}}.$$

The output of the MLP network is given as

$$x_t = \sum_{i=1}^{h} w_i^2 \sigma \left(\sum_{j=1}^{n} w_{i,j}^1 x_{t-j} + b_{i,j}^1 \right) + b^2,$$

where n is the number of inputs and h is the number of neurons in the hidden layer. The variables $\{w_{i,j}^1, w_i^2\}$ denote the weights of the connections between the input and the hidden layer, and the hidden layer and the output layer respectively. The variables $\{b_{i,j}^1, b^2\}$ denote the bias terms added to the hidden and output layers respectively (Fig. 3a).

RBF networks as an activation function use radial basis functions. For our purpose, we employed popular Gaussian function

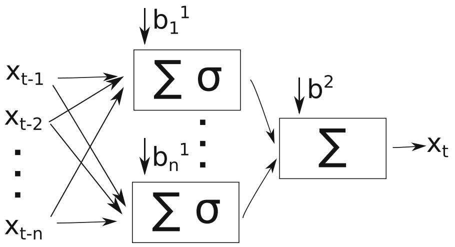
$$\sigma(\|\mathbf{x} - \mathbf{c}_i\|) = \exp\left[-\beta \|\mathbf{x} - \mathbf{c}_i\|^2\right],$$

where $\|\mathbf{x}\|$ denotes a norm of vector $\mathbf{x}$. The output of the RBF network is a linear combination of radial basis functions of the inputs and the weights of connections between neurons

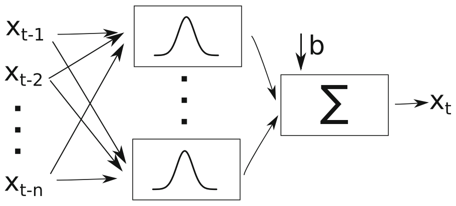
$$x_t = \sum_{i=1}^{h} w_i \sigma(||\mathbf{x} - \mathbf{c}_i||) + b$$

where w_i are the weights between hidden and output layers, h is the number of neurons in the hidden layer and b is a bias term (Fig. 3b).

Similarly to (F)ARIMA models, the neural network should minimize the RMSE defined in (6). For this purpose, the neural network is said to be trained in order to iteratively decrease the RMSE. For this purpose, from the variety of available approaches, we chose the popular back-propagation algorithm [30].



(a) Multilayer perceptrons with logistic function as an activation function



(b) Radial basis function network with Gaussian function as an activation function

Fig. 3 ANNs used in the experiments. Each network consists of three layers: input, hidden and output. Every subsequent layers are fully connected

4 Previous works

Modeling of video traffic has been extensively studied in the literature. Many measurement experiments have demonstrated that the real-world traffic in modern communication networks exhibits LRD and self-similarity [4]. In particular, these phenomena were observed also for traffic generated by video [31] and other multimedia applications, e.g. internet telephony or video-conferencing [32].

Due to the popularity of video sharing services, e.g. YouTube, most of the studies dedicated directly to HTTP streaming were reported in the last few years. At the beginning, the majority of the works was measurement-based, focusing on the characterisation of various aspects of HTTP video and its usage patterns. On the one hand, there are publications based on crawling web video sites for an extended period of time [33–35]. These works, which examined video popularity and users' behaviour, showed that statistics such as length, access patterns, growth trend, and active life span were quite different in comparison to traditional video streaming applications. On the other hand, we have works based on user traffic trace analysis including deep packet inspection, e.g. [8, 36–38]. Some works tried to investigate details of traffic characteristics, e.g. in [39] the authors examined the periodic behaviour of traffic generated by Microsoft Smooth Streaming and Netflix.

In [36], the authors provided formulas for average intensity and variability of network traffic. However, the proposed analytical model takes into account only non-adaptive streaming. To our best knowledge, none of the work has been focused on an investigation of aggregated adaptive video traffic.

Traditional aggregated traffic model assumed that the LRD and self-similar traffic can be generated using two state Markov ON-OFF source models with sojourn times drawn from two individually parametrised Pareto distributions. In the ON state, a source emits packets on a peak rate while in the OFF state no packets are emitted [4]. Taking into account the theoretical results presented in [40], which suggest that aggregating a large number of ON-OFF sources with the same heavy-tailed distribution in the two states results in a self-similar process, Willinger made the conjecture that heavy-tailed ON-OFF behavior provides the physical basis for the self-similarity observed in packet data traffic.

Nonetheless, contrary to these above assumptions, as it was mentioned in the Section 2.1, the HTTP video traffic has a more complex structure. In [36], the authors identified different streaming strategies employed by an HTTP server and showed that the strategies vary with the type of the application, i.e. they are dependent on the type of a web browser or a mobile application, and the type of a container used for video streaming. These findings are in agreement with the results presented in [8], where the authors found that the character of ON-OFF cycles depends on the available bandwidth of a network. The cycles are observed when the end-to-end available bandwidth exceeds the transmitted video bit rate by a certain percent.

Furthermore, though the transfer rate control implemented at the application level in video adaptive systems may appear analogous to the TCP congestion control, there are some key differences between these two logics, which may have an impact on a traffic characteristic: the two control algorithms operate at different levels in the protocol stack; TCP is a connection-oriented protocol while video adaptation is a connectionless protocol and TCP operates at the packet level whose size is about 1 KB, while video system operates at the level of segments, whose size is usually hundreds of kilobytes.

The described differences in the control logic between adaptive video and TCP are apparent when several video clips are multiplexed and streamed through a single network link. In such a scenario, the video players compete among themselves for the available bandwidth needed to transfer video chunks of variable size [41, 42]. The competition introduces additional video bit-rate oscillations, which translate into a higher probability of a buffer under-run of a video player and, consequently, video re-buffering. This may influence the characteristic of the measured traffic.

Taking into account the above factors, the abundant literature with propositions of models for TCP and Internet traffic may not directly be applicable to HTTP adaptive video, as traffic generated by these systems will probably be significantly different from the patterns produced by multiplexed exponentially or Pareto distributed ON-OFF sources.

Therefore, in our work, we try to estimate characteristic of the aggregated traffic produced by systems transmitting adaptive HTTP video generated by several popular play-out algorithms. We compute traffic intensity distribution, its autocorrelation and test it against LRD. Then, we try to fit to the examined traffic traces ARIMA/FARIMA processes and ANNs, which are universal time series models. Finally, we show that the obtained traffic models are able to enhance the performance of a simple adaptive algorithm used for video play-out.

ARIMA processes are considered as a classical and universal modelling tools. They can capture the linear and short-range dependencies occurring in time series. However, when used for general Internet traffic which exhibits LRD, the ARIMA processes have rather a poor performance. Nevertheless, due to their versatility and relative easiness of use, they are extensively applied for modelling and forecasting of network traffic, e.g. in [43] for Ethernet traffic, in [44] for traffic from websites, or in [22, 23] for general Internet traffic. However, as it was mentioned, network traffic has LRD, therefore, in order to capture its properties and overcome ARIMA limitations, researches sometimes use FARIMA processes [24, 25]. Nevertheless, one must note that FARIMA processes are computationally more complex and require more elaborate algorithms for estimation of their parameters.

ANNs have been widely applied to forecasting of time series including also different types of network traffic. Internet traffic prediction based on MLP network can be found, among others, in [45, 46] while RBF network is applied, among others, in [47, 48].

Generally, there are no conclusive guidelines regarding the process of network traffic prediction. Some authors state that the performance of traffic prediction model can be improved by multiplying the number of layers instead of increasing the number of neurons [48]. Nevertheless, other research reveals that more complex algorithms are not necessarily better, and there exists a specific range of operating parameters where predictions are generally more accurate [49]. Some researchers find that MLP networks give a little better results compared to RBF networks [23, 48], while in other works the performance of both networks are similar and depends on a traffic trace used for modelling [50]. The comparison presented in [23] shows that of FARIMA processes and ANNs have similar approximation errors. However, the best results are achieved for hybrid solutions which employ both FARIMA and ANNs simultaneously [23, 51, 52].

5 Experiments

5.1 Laboratory set-up

In order to capture the aggregated traffic, which was generated by an adaptive video system, we prepared a test environment emulating simplified content distribution system (CDN). The environment consists of: a network environment emulator (NEE), which in a real world

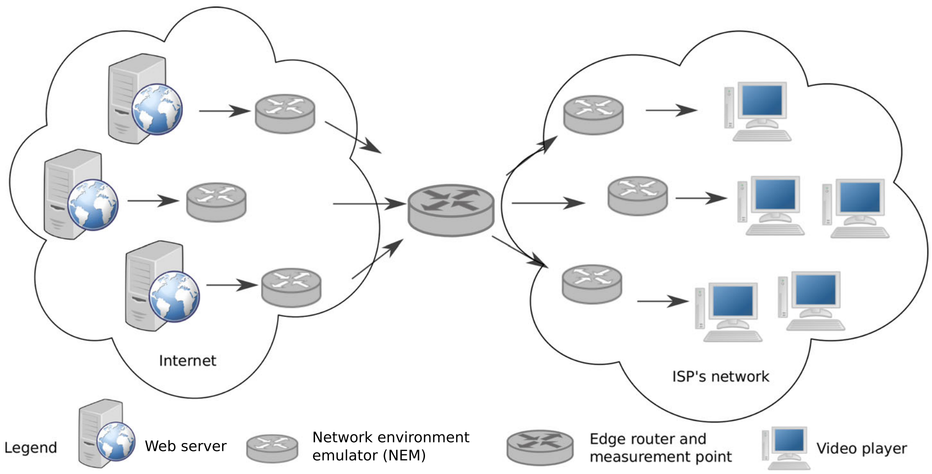

Internet
ISP's network
Legend
Web server
Network environment emulator (NEM)
Edge router and measurement point
Video player

Fig. 4 Experiment set-up which includes: network environment emulators, web servers, video players, a measurement point located in an edge router. The above elements constitute a CDN of a video provider and an end users' network

is applied to deliver video to multiple users; web servers; video players and a measurement point located in an edge router, as shown in Fig. 4.

As an NEE, we used a network emulation node based on the built-in Linux Kernel module *netem* [53]. The module is capable of altering the network QoS parameters such as network bandwidth or network delays; thus, it allows to test data transmission in different network environments.

The role of the web server plays Apache [54], which stores the video clips as a set of chunks which length is a parameter of the experiments. Each of the three servers used in the experiment for emulating the CDN has assigned a NEE which mimics a different network environment so that there are different packet delays and network bandwidth between the servers and the edge router. The delays and the bandwidth are uniformly distributed random variables which take their values between 0.01 s and 0.2 s for the delay, and 4 MB/s to 12 MB/s for the bandwidth. Taking into account that according to [2] about 65 %-80 % of current network traffic is generated by video services, we allocated a quota of 25 % of the bandwidth for background traffic which was generated by a tool presented in [55].

As a video player, we chose VLC media player with a DASH plug-in [16]. Both the player and the plug-in have an open-source code, thus, it is possible to manipulate or completely change the adaptation logic without affecting the other components. As a consequence, the plug-in enables the integration of a variety of adaptation logics making it attractive for performance comparison of different adaptive streaming algorithms and their parameters. As it was mentioned in Section 2.1, we replaced the default logic implemented in the plug-in with the implementation of four algorithms described in the literature. The players were divided into three groups. Each group has assigned an NEE which imposes on the group different network properties so that the NEE of the first group emulates properties of a wired network, the NEE of the second group emulates a wireless network and the NEE of the third group mimics a wireless mobile network. Table 1 summarizes basic parameters of the experiment.

Table 1 Simulation parameters

Parameter	Value(s)
Number of web servers	3
Number of NEE	6
Number of video players	120
Bw. at the edge router	23 MB/s
Bw. at the ISP's side (per a server)	4 - 12 MB/s
Packet delays at the ISP's side	0.01 - 0.2 s
Bw. at the client side (per a player)	0.16 - 1 MB/s
Packet delays at the client side	0.01 - 0.6 s

We transmitted several video files, acquired from [56] and presented in Table 2, through the simulation environment with variable network bandwidth and packet delays. Using the edge router with installed capturing software, based on Tcpdump and Libcap [57], we obtained five aggregated traffic traces. The length of each trace was trimmed to 10 minutes. Then, they were converted to time series represented by a point process C_n, which was defined in (1). As we want to apply the results of the analysis to enhance an adaptive play-out algorithm, we are interested in properties of relatively short time traces which last up to 10 minutes. Hence, we do not introduce users' churn or take into account switching of the players between different CDN servers. For the same reason, we also do not take into account video length distribution, its popularity or the patterns of users' access, which could possibly lead to a more complex model of aggregated video traffic on a longer term scale. All further described experiments were performed for each of five traces separately. Their results are presented as box plots which are able to depict groups of data through their quartiles, what allows investigating the variation of the data. In the case of autocorrelation and density, we show only averaged values. The figures with traffic intensities are based on a single, randomly selected network trace.

6 Results and discussion

6.1 Traffic characteristics

A visual assessment indicates that the pattern of the aggregated traffic presented in Fig. 5 has rather an irregular structure regardless of the aggregation scale. At the macro-scale, the traffic remains within a narrow range; nevertheless, from time to time there are periods of increased variability when the oscillation amplitude of the traffic increases, e.g. in the first

Table 2 Video clips used during the experiments

Name	Genre	Bitrate levels
Big Buck Bunny	anim.	100 kbit/s – 320x240, 300 kbit/s – 480x360, 700 kbit/s – 854x480, 1.2 Mbit/s – 1280x720, 2.5 Mbit/s – 1920x1080
Elephants Dream	anim.	
Red Bull Playstreets	sport	
The Swiss Account	sport	
Valkaama	movie	
Of Forest and Men	movie	

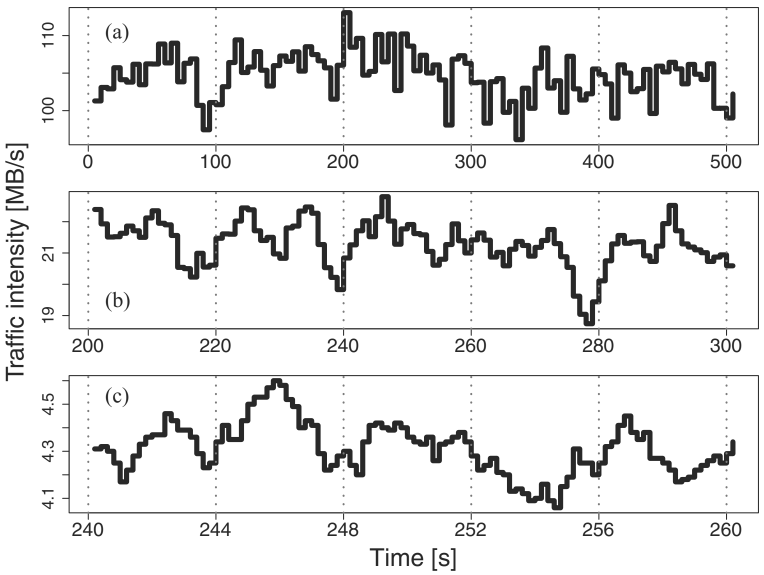


Fig. 5 Video traffic intensity in a **a**) 10 s **b**) 1 s **c**) 0.1 s time scale. Total number of players = 120, bandwidth limit = 23 MB/s

120 s of the trace as shown in Fig. 5a. Then, this period is succeeded by 80 s of relatively lower volatility. When we examine Fig. 5b, we can observe similar patterns. Fragment of the trace with increased variability between 200 s and 240 s is followed by a fragment of lower variability lasting about a dozen of seconds. The pattern repeats itself also on micro-scale depicted in Fig. 5c.

The correlation coefficients alternate between negative and positive values, see Fig. 6a (the first element of the autocorrelation, which is always equal to one, was removed). A high amount of a negative correlation indicates the anti-persistent nature of the examined traffic traces, what has a consequence that an increasing trend in the past is likely to be followed by a decreasing trend.

The distribution of the traffic intensity is negatively-skewed – the right tail of the distribution, presented in Fig. 6b, drops abruptly, what is the result of the bandwidth caps imposed in the experiment, while the left tail decays hyperbolically, breaking its slide for the parameters slightly above zero.

As presented in Fig. 6c, the Hurst parameter oscillates between 0.35 and 0.44, what confirms the anti-persistent nature of the traffic revealed by the autocorrelation analysis.

6.2 Traffic prediction with ARIMA and FARIMA models

To find the best fit of the ARIMA and FARIMA models to the obtained traffic traces, we employed the Box-Jenkins procedure. The procedure is described in details in literature, for instance in [58] in Section 2.3. The procedure includes several steps: an estimation of the model order, an estimation of the model coefficients, a diagnostic check of the obtained model and its prediction accuracy. We divided the examined trace into five, ten minutes length, sub-traces. Than, the prediction were repeated five times for each sub-trace. The

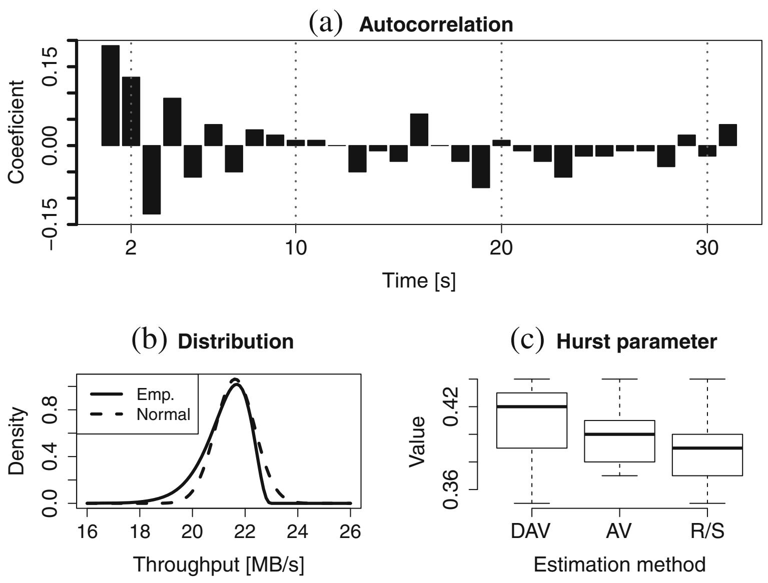


Fig. 6 Estimated **a**) autocorrelation **b**) density **c**) Hurst parameter for the examined network traces. Total number of players = 120, bandwidth limit = 23 MB/s

models were trained on the first half of each sub-trace and tested against the RMSE (6) on the second half of the sub-trace.

To determine the order of the models, we use Akaike's Information Criterion (AIC), presented, among others, in [59] in section 8.6, and defined as

$$\text{AIC} = -2\log(L) + 2(p + q + k + 1), \tag{12}$$

where L is the maximum value of the likelihood function for the model and k is the number of estimated parameters in the model. As the algorithm uses a stepwise search to traverse the model space, the values of p and q are selected by minimizing (12), rather than checking all their possible combinations. However, in order to avoid degeneracy of the model, we imposed a limit on the parameters number so that $p, q \leq 3$. From the set of candidate models, we chose one ARIMA (A2) and two FARIMA (F1 and F2) models, as presented

Table 3 Exemplary parameters of the examined models for a randomly selected experiment.

Model	Short	Parameters		
		α	d	β
ARIMA(4,0,0)	A1	$\alpha_i = 0.25$	0	0
ARIMA(2,1,1)	A2	$\alpha_1 = 0.15$	1	−0.42
		$\alpha_2 = 0.29$		
FARIMA(1,1,1)	F1	0.31	0.90	−0.13
FARIMA(2,1,1)	F2	$\alpha_1 = 0.43$	0.90	−0.21
		$\alpha_2 = -0.26$		

Except of the A1 model, the parameters of other models vary in different experiments in Table 3. The ARIMA(A1), plays a role of a reference model being an implementation of the simple bandwidth predictor specified in (5).

For the estimation of ARIMA parameters, we used the routines implemented in the *arima* library, which is a part of the standard *CRAN R* distribution [21]. While estimating the FARIMA parameters, we had to apply additional calculation, as the *arima* routines operate only on integer values of the differencing parameter d. Firstly, using the relationship between the differencing parameter d and the long-memory parameter α from (2), we estimated the value of d, which according to [60] section 8.2, is

$$d = \frac{1-\alpha}{2} \rightarrow d = H - \frac{1}{2}.$$

From the analysis presented in Fig. 6, the median of Hurst parameter is about 0.40 what implies that the differencing parameter $d = -0.1$. After the estimation of the above parameter, the time series was fractionally differenced. The calculation was based on the binomial expansion of $(1-B)^d$ which is given by

$$(1-B)^d = 1 - dB + \frac{d(d-1)}{2!}B^2 - \frac{d(d-1)(d-2)}{3!}B^3 + \ldots,$$

and curtailed at some suitably large lag L, which we set to 40 following [60] section 8.2. Thus, for $d = -0.10$ we get

$$y_t = x_t + 0.1\,x_{t-1} + 0.055\,x_{t-2} + \ldots + 0.001334\,x_{t-40}.$$

To the fractionally differenced time series $\{y_t\}$, we applied the *arima* library in order to get estimates of the model order and the values of the p and q parameters. The estimated exemplary parameters for the models are presented in Table 3 and the basic steps of the model preparation are given in Fig. 7a.

6.3 Traffic prediction with neural networks

The key elements of an ANN are the input variables, the number of hidden layers and number of nodes for each layer. As we have already defined in Section 3.2 the basic architecture of the ANNs which consist of three layers (including one hidden layer), thus, we focus here on the number of input and hidden nodes.

In order to estimate the number of input neurons, it is assumed that the analysed data is produced by an unknown M-dimensional dynamic system. Thus, using the embedding theorem [61], the purpose is to identify a simpler N, $N < M$ dimensional system which

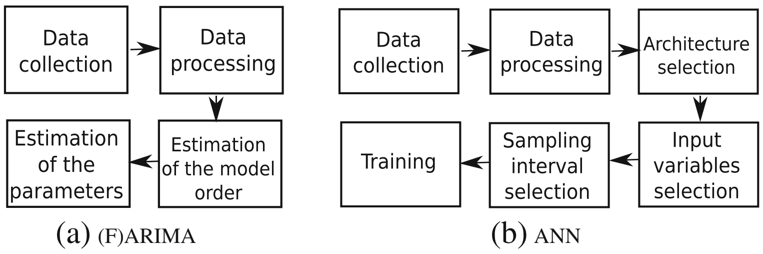

Data collection
Data processing
Estimation of the model order
Estimation of the parameters
(a) (F)ARIMA
Data collection
Data processing
Architecture selection
Input variables selection
Sampling interval selection
Training
(b) ANN

Fig. 7 Workflow for the models preparation

Table 4 Basic parameters of the ANNs applied in the experiments

Model	Short	Parameters		
		layers	neurons	τ
Multi-Layer Perceptron network	MLP	3	(7,9,1)	1
Radial Basis Function network	RBF	3	(6,6,1)	1

could be equivalent to the original system. Furthermore, the data are obtained from the past values $\{y_t\}$ with a sampling rate $\tau,\ \tau \in \mathbf{N}$. Thus, taking into account (4), we have

$$\hat{x}_{t+m} = \phi(x_t, x_{t-1}, \ldots, x_{t-(n-1)}) \equiv \theta(y_t, y_{t-\tau}, \ldots, y_{t-(N-1)\tau}), \tag{13}$$

where n indicates the number of input nodes for the network.

For the selection of the equivalent system dimension N, we chose a classical method called False Nearest Neighbors (FNN) [62]. The main idea is to examine how the number of neighbours of a point along a signal trajectory changes with increasing embedding dimension. If the embedding dimension is too low, many of the neighbors will be false. However, if the embedding dimension is appropriate, the neighbours become real. Therefore, by examining how the number of neighbours change as a function of dimension, the algorithms determine the correct embedding dimension.

For the selection of τ we applied Mutual Information (MI) method [63]. This procedure, which can be an equivalent of the correlation function but in a non-linear setting, uses the MI function. The first local minimum of the MI function is considered to be the sampling rate τ.

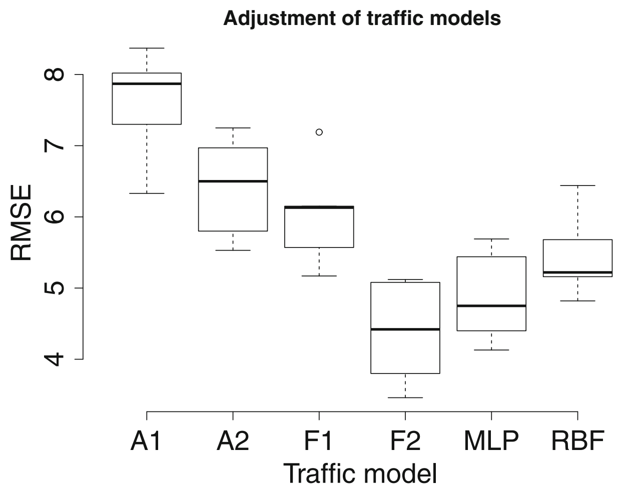


Fig. 8 RMSE of the analysed traffic models

Table 4 presents the summary of basic networks attributes and the basic steps of the model preparation are given in Fig. 7b. We performed the computations in *CRAN R* environment employing *nnet* package [64].

6.4 Models performance

The RMSE for the obtained models is presented in Fig. 8. From the auto-regressive models, the best fit model was obtained for FARIMA(F2) with the RMSE at about 4.5. The ARIMA(A2) and FARIMA(F1) gained RMSE more than 6, while the highest RMSE, nearly 8, had the ARIMA(A1) model. The results should not be surprising: the best scores acquired the most elaborate auto-regressive model, the poorest score belongs to the simplest one.

The models based on ANNs achieve similar performance to fractional auto-regressive models with their scores positioned between the results achieved by FARIMA(F1) and FARIMA(F2). The results of model based on MLP network is slightly better compared to the one based on RBF network.

Summarising, our experiments show that aggregated traffic exhibits usually long-memory and anti-persistence. The aggregated traffic traces have a fairly irregular structure with many sudden drops and jumps, which can be observed in time plots at a one-second scale. As a result, the traffic distribution is negatively-skewed, thus differing significantly from the classical Gaussian or Pareto distributions which are often employed in the engineering of network traffic, e.g. [65]. Nevertheless, the traffic trace can modelled by ARIMA and FARIMA models better than by a simple approximation defined in (5) and realised by the ARIMA(A1) model. The approximation errors achieved by the models based on ANNs are comparable with FARIMA models.

7 Models application

We applied our traffic models presented in Table 3 to enhance the functionality of a simple algorithm which adapts its play-out quality to measured throughput of a network [9]. Its major points are presented as Algorithm 1.

The algorithm in the Line 1 calls a function which estimates average network throughput n_b. The bandwidth estimation may be acquired directly from measurement of network traffic, e.g. measuring the time needed to download a chunk of video, or from a model of network traffic. In this case, the estimator returns the output of the models detailed in Table 3, taking into account that the output of ARIMA(A1) is just an averaged measurement of network traffic intensity in several previous periods (5).

When the video bit-rate v, which is needed for a smooth video play-out, is lower than the measured or estimated network throughput n_b reduced by ΔL, Line 2, the algorithm reports that the video quality level q should be increased, i.e. the chunk download module asks the server for bigger chunks, encoded in higher quality. When the throughput is not sufficient for the given level of video quality, Line 5, the opposite situation takes place: the quality level q is decreased and the download module is instructed to obtain chunks of poorer quality what simultaneously demands less network throughput. The parameter ΔL marks a region of network throughput for which there is no need to switch the quality to a higher level. As a result, the parameter plays a stabilising role and prevents switching the quality levels too frequently, which could have a negative impact on the overall video quality perceived by users. The constants $Q_{\max}$ and $Q_{\min}$ define a range of available levels of the quality.

Algorithm 1 Adaptation of video play-out based on a bandwidth estimation.

```
1: $v \leftarrow$ bandwidthEstimator()
2: if $v < n_b - \Delta L$ then
3:     $q \leftarrow q + 1$
4: end if
5: if $v > n_b$ then
6:     $q \leftarrow q - 1$
7: end if
Ensure: $q \in \{Q_{\min}, Q_{\max}\}$
8: return $q$
```

For a performance evaluation of Algorithm 1, which implements six different approaches of network bandwidth estimation, we employed two measures, applied also in [14].

This first measure describes how effectively the algorithm utilises available network bandwidth by computing the value of the following formula

$$\text{efficiency} = \frac{\sum_i^N (q_i/\tilde{q_i})}{N}. \tag{14}$$

Equation (14) computes the relation between the quality level q_i of the chunk to the theoretical quality level $\tilde{q_i}$ which is possible to achieve for the i-th chunk in given network conditions. The minimum value of the formula is $Q_{\min}/Q_{\max}$ if the play-out quality of every chunk is $Q_{\min}$, although the network conditions allow for Q_{max} quality. The value of the formula can reach one if, and only if, for every chunk $q_i = \tilde{q_i}$.

The play-out algorithm may try to maximise the value of (14) by adjusting the play-out quality to given network conditions as frequently as it is possible. Such behaviour will result in rapid oscillations of video quality, what will be negatively perceived by users [66, 67]. For this reason, we introduce the second measure which sums the quality switches:

$$\text{switches} = \sum_i^{N-1} |q_{i+1} - q_i|. \tag{15}$$

As Fig. 9 A shows, the traffic models are able to improve the efficiency of network bandwidth utilisation (14). The algorithm relying on an estimation of network bandwidth

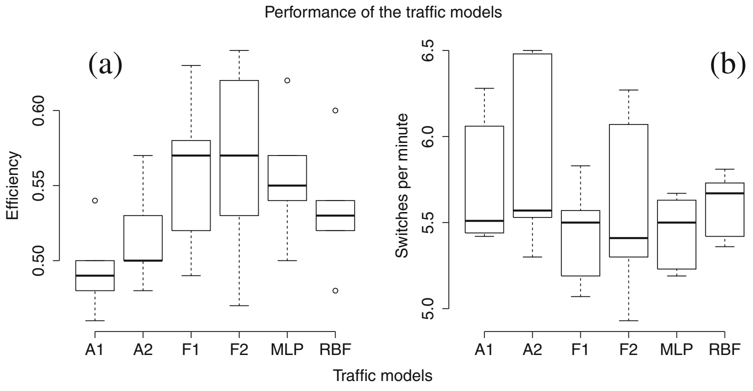


Fig. 9 Performance comparison of traffic models applied to play-out of adaptive video

provided by any of the examined models achieves a better score than the algorithm relying solely on direct network measurements, what is denoted here as ARIMA(A1). The FARIMA models achieve the best scores, nonetheless their results are subjected to higher variability compared to the ARIMA outcomes. Therefore, as Fig. 9a presents, there are sporadic situations, where the estimation delivered by the models are inaccurate and worsens the play-out quality. Probably during the play-out, models accuracy gradually deteriorates and their parameters need more frequent recalibration, e.g. every several minutes. As it could be expected from the comparison of models approximation errors, the efficiency of algorithms based on the ANN models is on an average a little worse compared to FARIMA models, however, is better compared to ARIMA models.

The frequency of quality switches remain at the similar level, regardless of the implemented estimator, see Fig. 9b. However, it is the good news: the improvement in throughput utilisation is a result of employment of the traffic models and does not come at the cost of a decrease of play-out stability.

8 Conclusions

In this paper, we examined statistical properties of aggregated traffic generated by 120 client-server pairs in a video adaptive system. The obtained results are at odds with the measurements of general Internet traffic, which showed that the Internet traffic exhibits persistence with Hurst parameter greater than 0.5. Our investigation of the autocorrelation function shows that the aggregated traffic has negative and positive correlations which exist for at least several seconds. The correlations are confirmed by estimations of the Hurst parameter, which in the most cases has values below 0.5, indicating the anti-persistence in the examined traffic traces and a negatively-skewed distribution function of the traffic intensity. Consequently, compared to the previous measurements, the contemporary video traffic and Internet traffic, which to large extend is composed of the video traffic, may exhibit more irregularities, has abrupt changes accompanied by sudden drops and jumps, what may sometimes pose a challenge to algorithms responsible for congestion management in computer networks.

The obtained traffic characteristic may be described by (F)ARIMA models and ANNs. The ARIMA model is computationally simpler, nevertheless, the FARIMA model fits better the long-range dependent data. As we showed, not only FARIMA, but also universal ARIMA model can improve the performance of a simple adaptive play-out algorithm. The models based on the ANNs achieve results comparable with the FARIMA models.

There may be doubts, how will the models behave when, e.g. the architecture of network traffic, or the number of video players, or their proportions change. Of course, there will be need to recompute parameters of the models and update them at certain time intervals. Our examination, repeated five times, show that in the most cases the models are able to improve the performance of the play-out algorithm. Due to their universality, the models with great probability should handle also traffic generated by systems with different configuration.

There is also a further field of improvement for the traffic models. Some authors report that the best results are usually obtained by hybrid techniques, e.g. combination of (F)ARIMA and different types of neutral networks [22, 23]. Thus, these more elaborate approaches may give a better adjustment to network data and provide better input for adaptive algorithms.

The obtained results may be useful for video content or Internet service providers or any other network operators which have access to traffic management and may impose additional traffic engineering procedures to assure an adequate level of quality for users. Secondly, as developers of the video players are free to employ their own adaptation strategies, potentially they may take into account prediction of network traffic in order to enhance video play-out.

References

1. Apple HTTP Live Streaming Resources - Apple Developer. https://developer.apple.com/streaming/
2. Apache Apache Web Server. www.apache.org
3. Adobe Adobe HTTP Dynamic Streaming. www.adobe.com/products/httpdynamicstreaming
4. Ahmed S (2009) IP Traffic forecasting using focused time delay feed forward neural network. Bahria University Journal of Information & Communication Technology 2(1):1
5. Adebiyi AA, Adewumi AO, Ayo CK (2014) Comparison of ARIMA and artificial neural networks models for stock price prediction. J Appl Math 2014:1–7
6. Akhshabi S, Anantakrishnan L, Dovrolis C, Begen AC (2012) What Happens When HTTP Adaptive Streaming Players Compete for Bandwidth? In: Proc. NOSSDAV
7. Akhshabi S, Begen AC, Dovrolis C (2011) An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http. ACM MMSys 11:157–168
8. Alcock S, Nelson R (2011) Application flow control in YouTube video streams. ACM SIGCOMM Computer Communication Review 41(2):24–30
9. Ameigeiras P, Ramos-Munoz JJ, Navarro-Ortiz J, Lopez-Soler JM (2012) Analysis and modelling of YouTube traffic. Transactions on Emerging Telecommunications Technologies 23(4):360–377
10. Abhari A, Soraya M (2010) Workload generation for YouTube. Multimedia Tools and Applications 46(1):91–118
11. Biernacki A (2007) Multi-Scale Modelling of VoIP Traffic by MMPP. In: Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications, pages 55–60. Springer
12. Botta A, Dainotti A, Pescapè A (2012) A tool for the generation of realistic network workload for emerging networking scenarios. Comput Netw 56(15):3531–3547
13. Bouten N, Schmidt RO, Famaey J, Latré S, Pras A, Turck FD (2015) QoE-Driven In-Network Optimization for Adaptive Video Streaming Based on Packet Sampling Measurements. Comput Netw
14. Cisco (2015) Cisco visual networking index: forecast and methodology, 2014-2019 Technical report
15. Core Team R (2015) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing Vienna Austria
16. Cowpertwait PSP, Andrew VM (2009) Introductory time series with R Springer Science & Business Media
17. Cheng X, Dale C, Liu J (2008) Statistics and social network of youtube videos. In: Quality of Service, 2008. IWQos 2008 16th International Workshop on, pp 229–238
18. Cortez P, Rio M, Rocha M, Sousa P (2012) Multi-scale Internet traffic forecasting using neural networks and time series methods. Expert Syst 29(2):143–155
19. Dethe CG, Wakde DG (2013) On the prediction of packet process in network traffic using FARIMA time-series model. J Indian Inst Sci 84(1 & 2):31
20. Famaey J, Latré S, Bouten N, Meerssche WVD, Vleeschauwer BD, Leekwijck WV, Turck FD (2013) On the merits of SVC-based HTTP adaptive streaming. In: Integrated Network Management (IM 2013) IFIP/IEEE International Symposium on, pages 419–426 IEEE, p 2013
21. Falk M, Marohn F, Michel R, Hofmann D, Macke M, Tewes B, Dinges P (2012) A first course on time series analysis: examples with SAS

22. Gadre V, Desai UB et al. (2012) MUltifractal based network traffic modeling. Springer Science & Business Media
23. Gowrishankar S, Satyanarayana PS (2008) A time series modeling and prediction of wireless network traffic. International Journal of Interactive Mobile Technologies (iJIM) 3(1):53–62
24. Hemminger S (2005) Network emulation with NetEm. In: Linux Conf Au, pp 18–23
25. Jacobson V, Leres C, McCanne S (2015) TCPDUMP public repository. Web page at http://www.tcpdump.org
26. Jiang J, Sekar V, Zhang H (2012) Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive. In: Proceedings of the 8th international conference on Emerging networking experiments and technologies, pages 97–108. ACM
27. Khashei M, Bijari M (2012) A new class of hybrid models for time series forecasting. Expert Syst Appl 39(4):4344–4357
28. Khashei M, Bijari M, Ardali GAR (2009) Improvement of auto-regressive integrated moving average models using fuzzy logic and artificial neural networks (ANNs). Neurocomputing 72(4):956–967
29. Katris C, Daskalaki S (2015) Comparing forecasting approaches for Internet traffic Expert Systems with Applications
30. Kantz H, Schreiber T (2004) Nonlinear time series analysis, volume 7 Cambridge university press
31. Liebeherr J, Burchard A, Ciucu F (2012) Delay bounds in communication networks with heavy-tailed and self-similar traffic. IEEE Trans Inf Theory 58(2):1010–1024
32. Lazaris A, Koutsakis P (2010) Modeling multiplexed traffic from h. 264/AVC videoconference streams. Comput Commun 33(10):1235–1242
33. Liu Y, Lee J (2015) An Empirical Study of Throughput Prediction in Mobile Data Networks. In: 2015 IEEE Global Communications Conference (GLOBECOM), pages 1–6. IEEE
34. Lederer S, Müller C, Timmerer C (2012) Dynamic adaptive streaming over HTTP dataset. In: Proceedings of the 3rd Multimedia Systems Conference, pp 89–94
35. Li Z, Zhu X, Gahm J, Pan R, Hu H, Begen AC, Oran D (2014) Probe and adapt Rate adaptation for http video streaming at scale. IEEE J Sel Areas Commun 32(4):719–733
36. Microsoft Microsoft Smooth Streaming. Web page at http://www.iis.net/download/smoothstreaming
37. Miller K, Al-Tamimi A, wolisz A (2015) Low-Delay Adaptive Video Streaming Based on Short-Term TCP Throughput Prediction. arXiv preprint. arXiv:hep-th/1503.02955
38. Murad ST, Levy JB (1986) Using renewal processes to generate long-range dependence and high variability. Dependence in probability and statistics 11:73–89
39. Mirzaei MM, Mizanian K, Rezaeian M (2014) Modeling of self-similar network traffic using artificial neural networks. In: Computer and Knowledge Engineering (ICCKE), 2014 4th International eConference on, pages 741–746. IEEE
40. Miguel MLF, Penna MC, Nievola JC, Pellenz ME (2012) New models for long-term Internet traffic forecasting using artificial neural networks and flow based information. In: Network Operations and Management Symposium (NOMS) IEEE pages 1082–1088 IEEE, p 2012
41. Müller C, Timmerer C (2011) A VLC media player plugin enabling dynamic adaptive streaming over HTTP. In: Proceedings of the 19th ACM international conference on Multimedia, pp 723–726
42. Martin J, Yunhui F, Wourms N, Shaw T (2013) Characterizing Netflix bandwidth consumption. In: consumer Communications and Networking Conference (CCNC) IEEE, pages 230–235 IEEE
43. Ni P, Eichhorn A, Griwodz C, Halvorsen P (2009) Fine-grained scalable streaming from coarse-grained videos. In: Proceedings of the 18th international workshop on Network and operating systems support for digital audio and video, pages 103–108. ACM
44. Park K, Kim G, Crovella ME (1997) Effect of traffic self-similarity on network performance. In: Voice, Video, and Data Communications, pp 296–310
45. Rutka G (2015) Some Aspects of Traffic Analysis used for Internet Traffic Prediction. Elektronika ir Elektrotechnika 93(5):7–10
46. Rutka G (2015) Neural network models for Internet traffic prediction. Elektronika ir Elektrotechnika 68(4):55–58
47. Rob JH, Athanasopoulos G (2014) Forecasting: principles and practice OTexts
48. Rajnish KY, Balakrishnan M (2014) Comparative evaluation of ARIMA and ANFIS for modeling of wireless network traffic time series. EURASIP J Wirel Commun Netw 2014(1):1–8
49. Rangarajan G, Ding M (2001) An integrated approach to the assessment of long range correlation in time series data arXiv preprint cond-mat/0105269
50. Rumelhart DE, Hinton GE, Williams RJ (1988) Learning representations by back-propagating errors. Cognitive modeling 5(3):1
51. Rao A, Lim YS, Barakat C, Legout A, Towsley D, Dabbous W (2011) Network Characteristics of Video Streaming Traffic. In: CoNEXT, Tokyo, Japan
52. Rhodes C, Morari M (1997) The false nearest neighbors algorithm: an overview. Comput Chem Eng 21:S1149–S1154

53. Ramos-Munoz JJ, Prados-Garzon J, Ameigeiras P, Navarro-Ortiz J, Lopez-Soler JM (2014) Characteristics of mobile youtube traffic. IEEE Wirel Commun 21(1):18–25
54. Sodagar I (2011) The mpeg-dash standard for multimedia streaming over the internet. Multimedia IEEE 18(4):62–67
55. Stockhammer T (2011) Dynamic adaptive streaming over HTTP–: standards and design principles. In: Proceedings of the second annual ACM conference on Multimedia systems, pp 133–144
56. Szmit M, Szmit A, Kuzia M (2013) Usage of RBF Networks in prediction of network traffic. In: FedCSIS Position Papers, pp 63–66
57. Satoda K, Yoshida H, Ito H, Ozawa K (2012) Adaptive video pacing method based on the prediction of stochastic TCP throughput. In: Global Communications Conference (GLOBECOM) IEEE, pages 1944–1950 IEEE
58. Takens F (1981) Detecting strange attractors in turbulence Springer
59. Valipour M, Banihabib ME, Behbahani SMR (2013) Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of Dez dam reservoir. J Hydrol 476:433–441
60. Villa BJ, Heegaard PE (2013) Detecting period and burst durations in video streaming by means of active probing. In: International Journal of Computer and Communication Engineering, volume 2, pages 460–467. IACSIT Press
61. Venables WN, Ripley BD (2002) Modern Applied Statistics with S, 4th edition. Springer, New York. ISBN 0-387-95457-0
62. Wuertz D, et all (2015) Rmetrics
63. Willinger W, Taqqu MS, Sherman R, Wilson DV (1997) Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level. IEEE/ACM Trans Networking 5(1):71–86
64. YouTube YouTube Statistics. http://www.youtube.com/yt/press/en/statistics.html
65. Zou XK, Erman J, Gopalakrishnan V, Halepovic E, Jana R, Jin X, Rexford J, Sinha RK (2015) Can Accurate Predictions Improve Video Streaming in Cellular Networks? In: HotMobile
66. Zhani MF, Elbiaze H, Kamoun F (2009) Analysis and prediction of real network traffic. J Netw 4(9):855–865
67. Zink M, Künzel O, Schmitt J, Steinmetz R (2003) Subjective impression of variations in layer encoded videos. In: Quality of Service–IWQoS 2003, pages 137–154. Springer

Arkadiusz Biernacki received the M.Sc. and Ph.D degree in Computer Science from the Silesian University of Technology, Poland, in 2002 and Ph.D. 2007 respectively. From 2007 he is an Assistant Professor at the Silesian University of Technology. His research interests focus on network traffic modelling and computer system simulations.