World Applied Sciences Journal 29 (Computer Sciences, Engineering and Its Applications): 200-207, 2014
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2014.29.csea.2267

Traffic Policing Approach for Wireless Video Conference Traffic

V. Khanaa, K.P. Thooyamani and R. Udayakumar

School of Computing Science, Bharath University - 73, India

Abstract: The subject of traffic policing for computer communication networks has been studied extensively in the literature. However, the constant development of new multimedia applications which are "greedy" in terms of bandwidth and Quality of Service requirements calls for new approaches to the traffic policing problem. In this work, we introduce a new video model for single H.263 videoconference sources and we use it in order to propose a new traffic policing approach for wireless videoconference traffic. We study well-known traffic policing mechanisms which still present interesting, unsolved problems when servicing video traffic and propose, to the best of our knowledge, for the first time in the relevant literature that the token generator is based on a traffic model and not on a fixed rate. The proposed approach shows significant improvement in the results obtained by all the traffic policing mechanisms and hence, shows that dynamic traffic policing can provide much higher efficiency than the widely used static approach.

Key words: Traffic policing · Token bucket · Traffic modeling · Wireless videoconference · H.263 video encoding

INTRODUCTION

TRAFFIC from video services, especially videoconference traffic, is expected to be a substantial portion of the traffic carried by emerging wired and wireless networks.

The explosive growth of wireless multimedia applications, in particular, calls for new sets of traffic control procedures to be implemented in order for the networks to cope with the bursty new applications, which have strict Quality of Service (QoS) requirements. For Variable Bit Rate (VBR) coded video, statistical source models are needed to design networks which are able to guarantee the strict QoS requirements of the video traffic. Video packet delay requirements are strict, because delays are annoying to viewer. Whenever the delay experienced by a video packet exceeds the corresponding maximum delay, the packet is dropped and the video packet dropping requirements are equally strict [1, 2].

There are three areas where single video source models are useful:

Studying what types of traffic descriptors are needed for parameter negotiation with the network at call setup, testing rate control algorithms and predicting the QoS degradation caused by congestion on an access link [3]. Hence, the problem of modelling video traffic, in general and videoconferencing, in particular, has been extensively studied in the literature. In various differences in successive video frame sizes of VBR video traffic were investigated. In different approaches are proposed for MPEG traffic, based on the lognormal, Gamma and a hybrid Gamma/lognormal distribution model. In the authors show that H.261 videoconference sequences generated by different hardware coders, using different coding algorithms, have gamma marginal distributions and use this result to build a Discrete Autoregressive (DAR) model of order one, which works well when several sources are multiplexed [4]. This result was also employed by which proposes an Autoregressive Model of order one for sequences of H.261 encoding. Although H.264 has been recently introduced as the new video coding standard, H.263 is still very widely used in a variety of applications.H.263 is a video standard that can be used for compressing the moving picture component of audiovisual services at low bit rates. Hence, in this work, we use our results in along with the work in on the GBAR(1) model in order to exploit the properties of the Pearson V distribution and be able to use a modified

Corresponding Author: V. Khanaa, School of Computing Science, Bharath University - 73, India.

GBAR(1) model for single H.263 videoconference traces. We then proceed to utilize the model in our proposal for traffic policing of wireless videoconference sources, which is the main contribution of this work [5]. To the best of our knowledge, this is the first work in the literature which addresses the problem of modelling single H.263 videoconference traces.

Very few other papers in the literature have studied the modelling of H.263 traffic, but all of them have actually used low-rate versions of movies (i.e. not videoconference traces) for their work and used the gamma distribution as the basis for their model. The DAR model was built based on extensive results showing that H.263 videoconference sequences have Pearson V marginal distributions. Since the Pearson V distribution is specified by two parameters which can be easily estimated from the mean and variance of the number of packets (cells) per frame, only those two moments and the autocorrelation coefficient are needed to build a DAR model for multiplexed VBR videoconference sources. However, as shown in, the DAR model fails to describe a single VBR source. The reason is that, due to the high autocorrelation of the video trace (in videoconference sources, autocorrelation is typically very high), the mean time between cell rate changes in the model is large in the actual trace, however, the rate fluctuates. This makes the DAR model inappropriate for traffic policing purposes, as each source needs to be monitored separately so that the system makes sure that the source conforms to its declared traffic parameters[6].

Proposed System: In this work, we have proposed a new traffic policing approach for videoconference traffic transmission over wireless cellular networks. We first built a new model for single H.263 videoconference sources and then proceeded to use the model in order to improve, with several modifications that we proposed, the standard traffic policing mechanisms used in the Token Bucket, Jumping Window and Sliding Window methods. To the best of our knowledge, this is the first work in the relevant literature using an adaptive probabilistic token generation rate based on a traffic model, instead of a fixed rate. Our results show that our dynamic approach provides significantly better results in policing the burstiness of video traffic sources than the static traffic policing mechanisms which have been widely presented and studied in the literature.

The proposed system is divided into four modules as shown in the following figure. The modules are implemented in. Net.

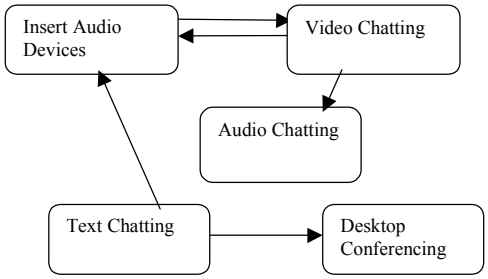

Fig. 1: Architecture of the Proposed System

Modules: This section contains the following modules namely,

- Capturing Video/Audio Devices Module
- Video conferencing Module
- Audio Conferencing Module
- Traffic Control Module

Capturing Video/Audio Devices Module: In this Module, The Project Will Capture the External Hardware Devices such as Web Camera, Head phone to Start the Conferencing. Capturing Video/Audio Devices Module is the first module of video conferencing the first module start with connecting the external hardware devices like video camera for capturing the video images and then audio devices like head phone for capturing voice [7, 8]. Video is used in technical discussions to represent the participating persons either directly or by pointing to their participation symbol used in the session (e.g., in the form of a picture or animation) as well as graphics, which are part of the presentation. Gestures and mimics are information carriers; their transmission creates a more natural way of direct communication and is therefore considered a very important feature. Workstations, PCs, or video boards can be used for display. For conferences with more than three or four participants, the space required to display the concurrent representation of all members can quickly turn into a big problem in particular when additional applications, such as shared editors or drawing boards, are used. For this reason, mechanisms are required to allow easy manipulation of the representation of each participant, i.e., to zoom in or out or bring active activities into the foreground [9].

Would be required for special applications even if all workplace computers were equipped with appropriate video hardware. Systems that support conference rooms with video walls attempt to support the entire wealth of human communication.

Video Conferencing Module: Video Conferencing Module Is The Second Module Of Conferencing Here The Conversation Should Be Made By Connecting Two System. With An IP Address And Port Number. Audio Plays A Very Important Role In The Description And Explanation Of The Information Visually Represented In A Conference. Often, Audio Itself Is More Important Than Video [10, 11]. For This Reason, A High-Quality Audio Transmission With A Full-Duplex Communication Means And Echo Suppression Is Desirable, If Space Information (Stereo) Can Be Transmitted In Addition, Or If The Active Speaker Can Be Allocated To The Pertaining Image Information, Then The Usability Can Be Further Improved. Conference Applications Require A Very Low Delay In The Underlying Network As Well As A High Available Bandwidth For The Media Transmission That Cab Be Highly Data Intensive To Ensure Acceptable Interactivity Between The Users. In Addition, They Require A Service For The Distribution Of Messages Along With The Transmission Of Both Data And Control Information To All Participants (Distributed Messaging).

Audio Conferencing Module: In this Module the Following Operations are done,

- Select the Audio Conference From the Conference Menu
- Enable the Audio By Connecting the IP Address As well As Port Address

Traffic Control Module: In this Module, the Video Traffic is controlled whenever the conferencing is going. The Algorithm Has Done The Job Of Controlling The Dynamic Video Traffic.

Operations in Traffic Control Module

Model Validation: To Validate Our Model, We Use A Generic Buffer Model, As Was Done , With The Necessary Modifications Due To The Different Video Encoding Schemes.Video frames arrive in integer multiples of 40 ms (P frames arrive after 40 ms intervals, PB frames arrive after 80 ms intervals and I frames arrive after 160 ms intervals). Each frame contains a certain number of cells, the exact amount of which is determined by either the data trace or by a realization of the GBAR(1) model in order to get the respective PT5-distributed values. All of the cells have the same size (equal to 48 bytes in our work, but any size can be chosen without any effect on our modelling and traffic policing results).

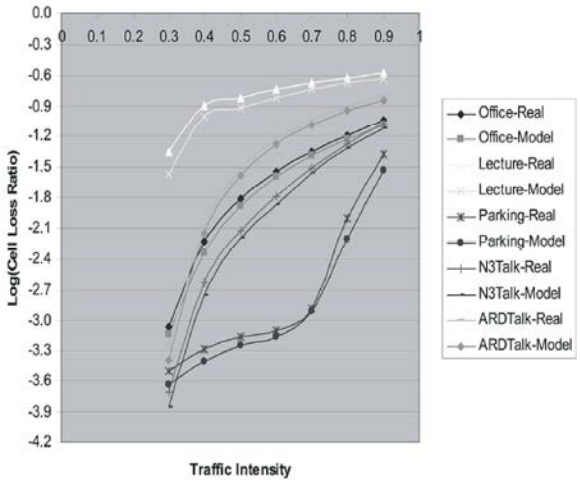

Fig. 2: Comparison Of Cell Loss Rates

The above Fig shows that the cell loss rates computed with the use of the model are very close to the cell loss rates computed from the actual data, regardless of the traffic intensity. These results refer to the infinite buffer case. The reason that the cell loss rate is highest in the case of lecture traffic is that the lecture camera trace is by far the most bursty of all traces under study (burstiness = peak/mean = 9.32 for the lecture camera trace) and it has the highest standard deviation/ mean ratio, which means that the values of the video frame sizes fluctuate significantly from the mean, hence adding to the bursty behavior of the trace. The reasons for the slight difference in the Parking trace curves in Fig. 2 are: 1) the very small, compared to all other traces, ratio of the standard deviation versus the mean, for this trace; this causes a much slower increase in the cell loss ratio for lower values of traffic intensity and 2) the fact that the Parking trace has, by far, the largest peak among all traces under study; this becomes a problem when traffic intensity increases, as the system is unable to cope with the large peaks when the channel is heavily loaded and hence, the cell loss ratio increases abruptly.

Similar results to those in Fig were obtained when computing the mean queue lengths of the model and the actual traces in an infinite buffer and when computing the cell loss ratio for the finite buffer case. As already explained, the quality of our modeling results is such that the model can be very effectively used in a new proposal for traffic policing of wireless videoconference traffic.

Channel Error Model: We use a simplified Fritch man Markov model to emulate the process of packet transmission errors. The Markov model used in for the

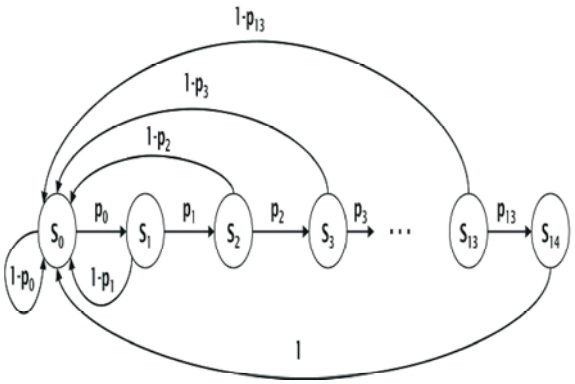

Fig. 3: Channel Error Model

downlink channel is presented in Fig. 3 and comprises of 15 states. State s0represents the "good state" and all other states represent the "bad states."

When the channel is in state s0, it can either remain in this state or make the transition to state s1 (with probability p0). When the channel is in a bad state, the transition is either to the next higher state or back to state s0,based on the status of the currently received packet. With this model, it is only possible to generate burst errors of at most length N _ 1, where N is the number of states(therefore, the maximum burst error in our model is 14 slots). The transition probabilities ðp0; p1; . . . p13Þ of the error model. The transition probabilities ðp1; . . . p13Þ of the error model have been adopted. The only difference in our model exists in probability p0, which is, in our model, slightly smaller (0.00146) than in [29] (0.001469).

The reason is that, the probability that the downlink channel is in a good state is considered in the authors' simulations to be 0.994.

Given that in our work we assume a very strict QoS requirement of 0.01 percent maximum video packet dropping [27] (if a video packet fails to be transmitted before the arrival of the next video frame due to congestion or channel errors, the packet is dropped), the use of the0.994 value for p good (i.e., p So) would prohibitive for our system to achieve the desired video QoS.Therefore, we have chosen in our study the value of the probability p bad,i.e., the steady-state probability that the channel is in bad state, to be equal to 5 _ 10_5; this value has been chosen in order to test an "almost worst" case scenario for our system ,given the maximum allowed video packet dropping.

Approaches For Traffic Control

The Token Bucket Approach

The Function of the Standard Mechanism: The token bucket mechanism has been widely studied in the recent past. The reason for its popularity is its ability to verify easily whether a source conforms to its declared (at call setup) traffic parameters. The token bucket is, along with the leaky bucket, the predominant methods for network traffic shaping. The two methods have different properties and are used for different purposes.

The leaky bucket imposes a hard limit on the source transmission rate, whereas the token bucket allows a certain amount of burstiness (which is necessary for video traffic) while imposing a limit on the average source transmission rate .

The basic idea behind the token bucket approach can be described by the following:

Tokens are put into the bucket at a certain rate. The bucket has a limited capacity. Each token represents a permission to the source to send a certain number of bytes into the network After each transmission from the source, tokens which correspond to the packets transmitted by the source are removed from the bucket.Arriving packets of K bytes are conforming and therefore, are immediately processed if there are tokens equivalent to K bytes in the bucket. If the current number of accumulated tokens (i.e., its equivalent in bytes) is less than the corresponding number of packets, the exceeding number of packets is nonconforming.

Nonconforming packets either wait until the bucket has enough tokens for them to be transmitted (traffic shaping) or they are marked as nonconforming in order to be discarded in the case of network congestion or they are discarded, when their bandwidth needs exceed the token bucket size.

If no packets wait to be transmitted, tokens can be accumulated up to the size of the token bucket. If the bucket fills with tokens and the source remains inactive or transmits at a rate lower than the token generation rate, the token buffer overflows and new incoming tokens are discarded and therefore, cannot be used by future source packets. In this way, the token bucket mechanism imposes an upper bound on the source's burst length, equal to the token bucket size, i.e., a token bucket permits burstiness, but bounds it.

The Dual Token Bucket Approach: Dual token buckets have been proposed and studied extensively in the literature (e.g., [13], [24], [26]). For the peak rate policer, the token generation rate was equal to the declared peak rate of the source, as was the token bucket size.

For the mean rate policer, the token generation rate was equal to the mean rate of the source, but two methods of generating tokens were implemented. The first method was the widely used static method of generating tokens

at a token rate equal to the mean (i.e., mean frame size divided by 80 ms), while the second method was our proposed method of generating tokens with the use of the Pearson V-based variant of the GBAR model for each video trace (tokens are generated by using the model and put in the bucket every 80 ms; the parameters for the model are computed by the declaration of each source, at the system entrance, of its mean, peak and standard deviation parameters).

The reason for this significant improvement is the quality of the video traffic model, which is able to "capture" the behaviour of the video traces, including the variations in the traces' bit rate, whereas the static mechanism generating tokens with a rate equal to the mean fails to do so. It needs to be stressed that in this study, we consider the very strict QoS requirement that a video packet is dropped if it fails to be transmitted before the arrival of the next video frame (and an upper bound of 0.01 percent is set on video packet dropping), since we want to test the system "at its limit" in order to study the efficiency of our dynamic traffic policing approach.

However, depending on the video application and on buffer availability, the upper bound on packet delay can be more lenient and in this case, the second policer will be responsible for the traffic shaping in the system, with a significant potion of the traffic shown as dropped being actually transmitted in that case, when tokens become available in the bucket.

These initial results showed that the idea of using probabilistic token generation instead of the widely used static one was very promising.

As the bucket size increased, two facts became evident from the results: video packet dropping is decreased and the difference between the static approach and our proposed method is also decreased.

The reason for both the above results is the same: as the bucket size increases, more tokens can be accumulated when small video frames are transmitted by the source and the accumulated tokens help the system to overcome bursts of large video frames more smoothly.

Therefore, video packet dropping decreases and the advantage of the use of the model slightly recedes. If we consider an infinite token bucket size, the packet dropping with both mechanisms would be zero, for "behaving" sources.

The Triple Token Bucket Approach: An important conclusion derived from the results is that, although our approach provides much better results than the static one, the video packet dropping results for both methods are unacceptably high; a strict QoS requirement for videoconference traffic sets the acceptable percentage for packet dropping at just 0.01 percent.

It needs to be noted here that in all our results, the packet dropping enforced by the policers is added to the losses due to errors in the wireless channel (however, the latter are much smaller, due to the value of p good in the channel error model, which is explained).

The very high packet dropping denotes the poor performance of the dual token bucket approach, as it is unfair to sources which conform to their declared traffic parameters.

The reasons for this result will be discussed in more detail when we present our results on another policing mechanism.

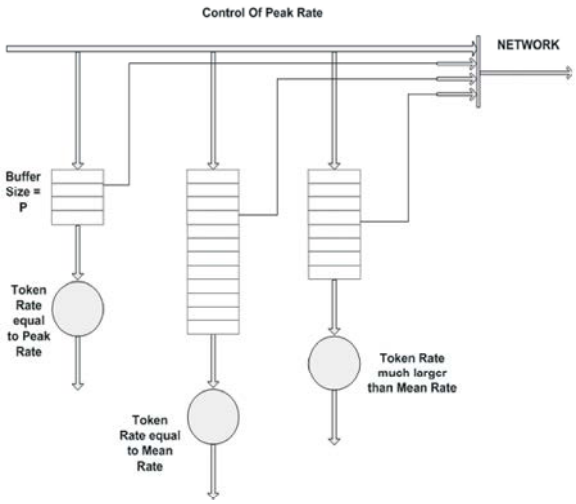

Fig. 4: The Proposed Triple Token Bucket Mechanism

Hence, three solutions can be applied to deal with the problem of high packet dropping from the second policer of the dual token bucket:

First Solution: The bucket size can be considered infinite, which, as explained above, is a bad solution.

Second Solution: Instead of a dual token bucket, a triple token bucket can be implemented. Triple token buckets have been used in the literature but not as often as dual token buckets. The triple token bucket mechanism uses one token bucket in series with two token buckets connected in parallel. The first token bucket in series controls the peak rate and the system of the two parallel token buckets consists of one bucket which has a mean token generation rate larger than that of the source and a

small token bucket size and another bucket which has a mean token generation rate close to that of the source but a large token bucket size.

The very significant differences between the percentages can be explained by the equally large differences in the standard deviation of each trace.

The trace with the largest standard deviation in comparison to its mean (lecture camera trace) is the most "greedy" one in terms of required token generation rate, while the trace with the smallest standard deviation in comparison to its mean (parking camera trace) is the less greedy one.

The greediness of sources with high standard deviation/mean ratio can be explained by the fact that the frame sizes of such sources have a larger variability than that of the frame sizes of sources with low standard deviation/mean ratio; this variability needs to be "covered" by the token generation rate of the third bucket, in order to keep the video packet dropping as low as 0.01 percent.

Third solution: We implemented again a triple token bucket, almost identical to the one described in the second solution, with one significant difference: instead of using a fixed token bucket size for the third policer, which is solely responsible for the video packet dropping in the absence of network congestion, we allowed the bucket size to change dynamically over the simulation time. It should be noted that The token generation rate mechanism for the third policer is static. As the mechanism needs to generate tokens at a much higher rate than the mean of the source, there is no use in applying a model-based token generator.

The dynamic triple token bucket mechanism (third solution) does produce better results than the static one (second solution) for large time windows (in the order of tens of frames); however, this is not a good choice, as the third token bucket is the most "tolerant" one and it should at least control burstiness, which is impossible if the rate control is not often performed. Therefore, we conducted all our simulation tests for this mechanism with relatively small time windows.

All our results for the token bucket controlling the mean source generation rate show that, even with the significant improvement provided by our scheme, a high percentage of the source information will be marked as nonconforming and will be discarded in the case of network congestion.

Therefore, a strict Call Admission Control mechanism must be implemented at the entrance of the network to prevent congestion.

Jumping Window: The Jumping Window mechanism uses windows of a fixed length T side by side through time.

A new window starts immediately after the conclusion of the previous one. During a window, only K bytes (or packets) can be submitted by the source to the network. In the case that a source attempts to transmit more than K bytes, the excessive traffic is dropped (or marked as nonconforming, as in the case of the Token Bucket).

The mechanism is implemented with the use of a token counter, similar to the one of the Token Bucket and in each new window, the associated packet counter is restarted with an initial value of zero . In our study, we use a modification of the Jumping Window mechanism, in order to implement a more dynamic traffic policing mechanism: in the case that less than K bytes are transmitted by the source within one window, the token counter is not restarted, but starts with an initial value equal to the remaining tokens.

Therefore, in our study, the jumping window mechanism can be regarded as an extension of a token bucket mechanism implemented over a longer time window than the ones.

Similarly to the results presented in presents the loss for a source transmitting the N3 TALK trace when two Jumping Window mechanisms are used for policing. The peak rate is policed by a token bucket with generation rate and size equal to P0, where P0 denotes the peak amount of traffic transmitted by the source within one time window.

We attempted to police the mean rate with various sizes of token buckets with a generation rate equal to either the declared mean of the source, or a token generator based on our model.

The results presented in show again clearly that the loss experienced by the trace is lower with the use of our proposed traffic policing scheme, for all bucket sizes examined (the results for the other four traces are similar in nature).

However, the loss decrease with the use of the model here is smaller (both in percentage of the loss achieved with the static mechanism and in absolute numbers) than the loss decrease in the case of the token bucket mechanism.

Sliding Window: The Sliding Window (Moving Window) mechanism is similar to the Jumping Window, but more stringent and more complex to implement.

This mechanism again ensures that the maximum number of bytes transmitted by a source within any given time interval of duration equal to the fixed window size, T, is upper bounded by K byte The difference with the Jumping Window mechanism is that each video frame size is remembered for the width of exactly one window, starting with the specific video frame And ending T frames later.

This mechanism can be interpreted as a window, which is steadily moving along the time axis, with the requirement that the frame sizes of T frames are stored for the duration of one window. This is the reason that the implementation complexity is considerably higher than for the other two mechanisms (Token Bucket and Jumping Window), as the complexity is directly related to the window size; also, since the content of successive time windows differs by just one frame.

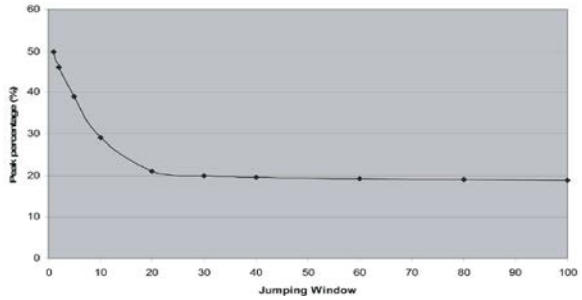

Fig. 5: Jumping Window Size Versus Percentage Of Peak Traffic, For The Office Camera Trace

Mechanism enforces the strictest bandwidth enforcement policy compared to the Token Bucket and the Jumping Window mechanisms.

Again, as in the case of the Jumping Window mechanism, we use, in our study, a modification of the Sliding Window mechanism, in order to implement a more dynamic policy: in the case that less than K bytes are transmitted by the source within one window W, the tokens left in the bucket are not discarded, but they are added to the token bucket of the next window (W+1), which has one more frames to police before it ends.

Tables present the results for a dual and triple Sliding Window mechanism, respectively, similar to the dual and triple Jumping Window mechanisms.

The Sliding Window size has been chosen equal to the Jumping Window (40 frames) for the purpose of comparison. With the use of this mechanism, we observe from Table that the packet loss results for the dual policer are higher than those achieved with the Jumping Window mechanism.

The reason for this result is the significantly large window size; in our discussion of Table when the Jumping Window mechanism was used, we explained that when the system is studied at fixed, consecutive, but more widely dispersed time periods, the token generator based on the model naturally tends to behave similarly to the static one, as they have the same mean generation rate.

Also, as already mentioned, when the Sliding Window policer is used instead of the Jumping Window one, the consecutive periods (windows) differ by just one video frame.

The combination of a large window and a frame-by-frame progress enforce the very strict nature of the Sliding Window mechanism, in comparison to the Jumping Window.

Bandwidth of Self-Similar Sources," ACM Trans. Modeling and Computer Simulation.

CONCLUSION

Based on our results when studying the Token Bucket, Jumping Window and Sliding Window schemes, we believe that in the case when all users are considered equal by the provider (i.e., in the case that all users are charged equally for a given service), the better choice is the triple Jumping Window mechanism which is the more "lenient" of the three in terms of "clipping" videoconference sources which generally conform to their declared traffic parameters but, at times, have large bursts due to the nature of video traffic.

Our traffic policing approach is also applicable for the new video coding standard, H.264. In our latest work , we have shown that traffic from multiplexed H.264 videoconference sources can be modelled with DAR based on the Pearson V distribution, as H.264 videoconference sequences have Pearson V marginal distributions.

The only difference with H.263 is that the modelling is more complex due to the fact that I, P and B frames need to be separately modeled for H.264 sources, whereas H.263 traces can be modeled as a whole.

Since all of our results confirm that the idea of using probabilistic token generation is very promising, the next step in our work will be to implement the modified GBAR(1) model to single H.264 videoconference sources and study the implementation of our proposed traffic policing approach for that type of videoconferencing traffic.

REFERENCES

1. Fitzek, F.H.P. and M. Reisslein, 2001. "MPEG-4 and H.263 Video Traces for Network Performance Evaluation," IEEE Network, 15(6): 40-54.
2. Le Boudec, J.Y., 2002. "Some Properties of Variable Length Packet Shapers," IEEE/ACM Trans. Networking, 10(3): 329-337.
3. Raghunathan, V., S. Ganeriwal, M. Srivastava and C. Schurgers, 2004. "Energy Efficient Wireless Packet Scheduling and Queuing," ACM Trans. Embedded Computer Systems, 3(1): 3-23.
4. Procissi, G., A. Garg, M. Gerla and M.Y. Sanadidi, 2002. "Token Bucket Characterization of Long-Range Dependent Traffic," Computer Comm., 25(11/12): 1009-1017.
5. Fiddler, M. and V. Sander, 2004. "A Parameter Based Admission Control for Differentiated Services Networks," Computer Networks, 44(4): 463-479.
6. McKenzie, E., 2008. "Autoregressive Moving-Average Processes with Negative-Binomial and Geometric Marginal Distribution," Advances, 7(1): 95-112.
7. Ors, T. and S.P.W. Jones, 1995. "Performance Optimizations of ATM Input Control Using an Adaptive Leaky-Bucket Mechanism,"Proc. Third IFIP Workshop Performance Modeling Evaluations ATM Networks.
8. Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. Application of Soft Computing Techniques in weather forecasting : Ann Approach, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1845-1850.
9. Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. Improving Web Information gathering for personalised ontology in user profiles, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1675-1679.
10. Udayakumar, R., A. Kumaravel, Rangarajan, 2013. Introducing an Efficient Programming Paradigm for Object-oriented Distributed Systems, Indian Journal of Science and Technology, ISSN: 0974-6846, 6(5S): 4596-4603.
11. Saravanan, T., V. Srinivasan and R. Udayakumar, 2013. A Approach for Visualization of Atherosclerosis in Coronary Artery, Middle-East Journal of Scientific Research, ISSN:1990-9233, 18(12): 1713-1717.