COMPUTATION OF THE CHARACTERISTIC CURVES FOR VIDEO SOURCES

by

Subir Varma

Emerging Technologies, Dept C70A/673

Networking Systems Architecture, IBM Corp., RTP, NC 27709

email: *subirv@vnet.ibm.com*

Ph. No.: (919) 254-7493

ABSTRACT

A video source can be described by its characteristic curves, namely the peakedness, burstiness and loss curves. These curves can be used at the time of connection set up to provide Traffic Descriptors (TDs) to the network, and request Quality of Service (QOS) parameters from it. In another paper [Va] we provided techniques for obtaining the TD and QOS parameters, if the peakedness, burstiness and loss curves are known. The objective of this paper is provide efficient algorithms for computing these curves.

1. Introduction

Consider a compressed video source of duration T and frame rate FR, such that the i^{th} frame is of size f_i bits. In order to describe this source to a network, the following Traffic Descriptors (TD) have to be specified:

- The peak rate C at which the data is sent into the network.
- The leaky bucket parameters (σ, ρ) such that the rate $m^C(t), 0 \leq t \leq T$ of the input into the network, satisfies the inequality

$$\int_{t_1}^{t_2} m^C(t) \leq \sigma + (t_2 - t_1)\rho, \quad 0 \leq t_1 \leq t_2 \leq T, \tag{1.1}$$

leaky-bucket constrained source

The source also requests the following Quality of Service (QOS) parameters from the network:

- The maximum end-to-end delay Δ,
- The maximum cell delay variation δ, and
- The maximum cell loss ratio ϵ, that it can tolerate.

In [Va] we described a procedure for obtaining the TD and QOS parameters for such sources, which consists of the following steps:

(1) : Send the video trace into a single server queue with rate C, and let $s(C)$ be the maximum queue length observed. Obtain a plot of $s(C)$ vs C by repeating this procedure for different values of C. We will refer to this plot as the *peakedness* curve. Assuming that S_{TR} is the amount of buffer available in the transmitter, choose a peak rate C such that $s(C) = S_{TR}$. The end-to-end delay Δ and jitter δ are then obtained in a straight forward manner by making use of relations between the sender and receiver buffer occupancies [Va].

(2) : Let $m^C(t), 0 \leq t \leq T$ be the rate at which packets are sent into the network, after the peak rate smoothing. We need to obtain the parameters (σ, ρ) such that (1.1) is satisfied. Unfortunately, as several researchers have pointed out [Gr], [LoVa], [Va], there is no single pair (σ, ρ) that uniquely characterizes a source. In fact for every value of ρ in the range $0 \leq \rho \leq \max_{0 \leq t \leq T} m^C(t)$, there is a σ such that the source is (σ, ρ) conformant. Hence the set of conformant (σ, ρ) pairs describe a curve which has been

referred to as the *burstiness curve* by Low and Varaiya [LoVa]. In [Va] we describe some procedures by which the best (σ, ρ) pair can be obtained, by combining the information in the burstiness curve with other information about network behavior.

(3) : The burstiness curve is a sufficient description of a traffic source if it cannot tolerate any loss in the network. However there are sources that can tolerate a certain amount of loss, in return for which they reduce their network costs by consuming less resources. The loss parameter ϵ, introduces a third dimension in the traffic descriptor, so that a different burstiness curve can be drawn for each value of ϵ. These curves have been referred to as *loss curves* by Wong and Varaiya [WoVa]. For a given value of ϵ, procedures are provided in [Va] to obtain the best (σ, ρ) pair.

The peakedness, burstiness and loss curves can be obtained through simulation based techniques, as described in [Va]. However simulation can be very time consuming especially when the packet size is small. The objective of this paper is provide efficient algorithms for computing these curves, that do not rely on simulation. The rest of the paper is organized as follows: In Section 2 we provide algorithms for computing the peakedness curve, while in Section 3 we provide algorithms for computing the burstiness curve. These algorithms are extended in Section 4 to obtain the loss curves.

2. Computation of the Peakedness Curve

In order to compute the peakedness curve, we need to obtain the maximum queue length observed when the video trace is passed through a single server queue with deterministic service time C. We repeat this procedure for a range of values of C, and then connect the resulting points together. For each value of C we also compute the values $(s_i^C, t_i^C), 1 \leq i \leq b(C)$, where $b(C)$ is the number of busy periods, s_p^C is the time instant at which the i^{th} busy period commences while t_i^C is when it ends. This information will be used in computing the burstiness curves in the next section.

Note that the number of frames in the video trace is given by $N = FR \times T$. Also let τ be the constant interval between two frames so that $\tau = \frac{1}{FR}$. Then the i^{th} frame occurs at time $i\tau, 0 \leq i \leq N$ and is of size f_i bits. Let $f_{\max} = \max_{0 \leq i \leq N} f_i$, be the size of the biggest frame in the trace. As long as the peak rate C satisfies $C > \frac{f_{\max}}{\tau}$, we have

$$s(C) = f_{\max}, \quad b(C) = N, \quad s_i^C = i\cancel{T} \quad \text{and} \quad t_i^C = \frac{f_i}{C} \tag{2.1}$$

However for $C \leq \frac{f_{\max}}{\tau}$, neighboring frames collide with one another, thus leading to larger maximum queue lengths and a smaller number of busy periods. Let $f_i'(C) > f_i$ be the size of the queue just after the instant at which the i^{th} frame arrives, assuming that the peak rate is C. Note that since the maximum queue always occurs just after the arrival of a frame, we have that

$$s(C) = \max_{0 \leq i \leq N} f_i'(C)$$

The following algorithm computes $f_i'(C), 0 \leq i \leq N$ and $(s_j^C, t_j^C), 1 \leq j \leq b(C)$.

(0) : Let the index i denote the i^{th} frame and let the index j denote the j^{th} busy period. Let $s_0^C = 0, f_0'(C) = f_0$ and initialize $i = j = 0$.

(1) : If $\frac{f_i'(C)}{C} < \tau$ then set

$$f_{i+1}'(C) = f_{i+1}, \quad t_j^C = i\overset{\tau}{\cancel{T}} + \frac{f_i'}{C}$$

Also let $j \leftarrow j + 1$ and $s_j^C = \overset{(i+1)\tau}{i\cancel{T}}$. If $\frac{f_i'(C)}{C} \geq \tau$, then set

$$f_{i+1}'(C) = f_{i+1} + f_i'(C) - C\tau \tag{2.2}$$

(2) : If $i = N$ then go to step (3), otherwise set $i \leftarrow i + 1$ and go back to step (1).

(3) : Set

$$s(C) = \max_{0 \leq i \leq N} f_i'(C) \quad \text{and} \quad b(C) = j$$

The peakedness curve obtained by applying this procedure will be an upper bound to the exact curve. In order to show this, the following proposition is used.

Proposition 1. *The peakedness curve is piecewise linear and convex.*

Proof: Assume that the largest queue length occurs in the j^{th} busy period for $C = C_0$, the frames in that busy period are $\{f_i, \ldots, f_{i+m}\}$, and the maximum occurs at arrival of that the maximum occurs to that frame (the maximum may lie

the f^{th}_{i+k} frame. Further assume that this continues to be the case until C decreases to $\overline{C}_0$.

By recursively applying (2.2) k times, it is easy to see that (starting with $f'_i = f_i$),

decreasing function

$$s(C) = f'_{i+k}(C) = \sum_{l=0}^{k} f_{i+l} - kC\tau, \quad \overline{C}_0 \le C \le C_0 \tag{2.3}$$

This is clearly the equation of a straight line with slope $k\tau$.

In order to show the convexity, it is sufficient to show that the slope of the linear segements that constitute $s(C)$ increases as C decreases. There are three distinct scenarios that can occur at $C = \overline{C}_0$:

- The new maximum occurs in the (same) j^{th} busy period again, but at the $(i+\overline{k})^{th}$ frame. Note that is clear from (2.3) that for this to be true, $\overline{k} > k$, so that the slope increases.
- The number and duration of the busy periods changes, but the new maximum continues to occur at the $(i+k)^{th}$ frame in the old j^{th} busy period. In this case, the only way that the peakedness curve will change slope is if the busy period just prior to the j^{th} busy period, coalesces into it, and forms a larger busy period. In this case, the number of frames $\overline{k}$ that occur in the new busy period before the tagged frame is clearly larger than k, which leads to an increase in slope of the peakedness curve.
- The number and duration of the busy periods changes, and the new maximum now occurs at the $(\overline{i}+\overline{k})^{th}$ frame in the $\overline{j}^{th}$ busy period. If the $\overline{j}^{th}$ busy period remains unchanged over the interval $[\overline{C}_0, C_0]$, then the change in maximum would occur due to the contribution of the second term in the LHS of (2.3). If this is the case then clearly $\overline{k} > k$. If the $\overline{j}^{th}$ busy period coalesces into the busy period just before it, for some $C \in [\overline{C}_0, C_0]$, then the change in maximum could occur due to contributions from both the terms in RHS of (2.3). However a little thought should convince the reader that the contribution due to the frame sizes in $(\overline{j}-1)^{th}$ busy period in the first term of (2.3) for the $\overline{j}^{th}$ busy period is zero, thus implying an increase in slope. ∎

As a result of Proposition 1, the peak rate C obtained by solving the equation $s(C) = S_{TR}$ (see [Va]), will lead to a maximum buffer fill level less than S_{TR}, i.e., there will not be any data loss due to the fact that peakedness curve obtained is not the exact one.

If the algorithm runs for all C then it must be exact

$L(C)$ = # busy periods for peak rate C

3. Computation of the Burstiness Curve

As a result of peak rate shaping, we obtain a waveform $m^C(t), 0 \le t \le T$, such that $m^C(t) = C, t \in [s_i^C, t_i^C], 1 \le i \le b(C)$ and $m^C(t) = 0$ otherwise. The strategy that we will pursue to obtain the burstiness curve is to analytically derive the maximum queue length when $m^C(t)$ is fed into a single server queue with deterministic service rate ρ, for a set of values of ρ in the range $[0, C]$. Furthermore we show that the entire curve can be obtained by connecting these points with linear segments.

The total amount of data contained in the video source M, is given by

termination time of busy period

start time of busy period

$$M = C \sum_{i=1}^{b(C)} (t_i^C - s_i^C)$$

It is clear that $\sigma(0) = M$, since if the video source is sent into a single server queue with rate 0, then the maximum queue length is the same as the amount of data. Note that $\sigma(\rho) = 0$ for $\rho \ge C$. For the case when $\rho \in (0, C)$, the analysis is complicated by the fact that there are several busy periods in the interval $[0, T]$. Hence in order to obtain the maximum queue length, we need to consider every busy period and calculate the maximum queue length achieved in each. Our task is simplified by the following result:

Proposition 2. *The burstiness curve is piecewise linear.* ?

Proof: Note that as the value of ρ increases from 0 to C, the number of busy periods increases from 1 to L, where L is the number of packets. Consider the case when the maximum queue size occurs in the i^{th} busy period. The maximum always occurs at times t_p^C at which the rate of $m^C(t)$ changes from C to 0. Assuming that the i^{th} busy period started at $t = s_i$, the maximum queue length σ is thus given by

because from that point there will be no utility

$$\sigma = \int_{s_i}^{t_p^C} m^C(t)dt - \rho(t_p^C - s_i) \qquad (3.1)$$

?

As long as the number of busy periods does not change σ and ρ satisfy equation (3.1), which is an equation of a straight line with slope $t_p^C - s_i$, which shows that the burstiness curve is piecewise linear. It is also clear that the values of ρ at which the slope changes are precisely the same values at which the number of busy periods changes. ∎

? probably wrong

By virtue of this proposition, we only need to compute points on the burstiness curve at places where the slope changes, since the entire curve can be obtained by joining these points with linear segments. The set of these points is defined as $\{\rho_k\}_{k=0}^{S}$. Note that $\rho_0 = 0$ and $\rho_S = C$.

The following is a brief summary of the algorithm to compute the burstiness curve: Let us assume that we have obtained the burstiness curve in the interval $[0, \rho_k)$. In order to extend it to the interval $[\rho_k, \rho_{k+1})$, we proceed in the following three steps:

- Identify all the busy periods for $\rho = \rho_k$. Assume that there are $n(\rho_k)$ such busy periods, and the i^{th} one spans the interval $[s_i(\rho_k), t_i(\rho_k)], 1 \leq i \leq n(\rho_k)$.
- Identify the value ρ_{k+1} at which the number of busy periods increases to $n(\rho_k) + 1$ $(= n(\rho_{k+1}))$. wrong
- In each busy period, obtain the maximum queue length, say $q_i(\rho_k), 1 \leq i \leq n(\rho_k)$. Then $\sigma(\rho_k)$ and the slope of the straight line between ρ_k and ρ_{k+1} can be obtained by formulae that are given in the detailed description.

A detailed description of the algorithm now follows:

s_i^C = starting time of busy period i for trace because of peak rate shaping

(0) : Initialize the algorithm by setting $\rho' = 0$, $b = 0$, $s_0(\rho') = 0$ and $t_0(\rho') = 0$.

(1) : In the rate function $m^C(t), 0 \leq t \leq T$, find the smallest time $s_i^C > t_b(\rho')$ such that $m^C(t) = C$, If no such s_i^C exists, then all the busy periods have been identified for $\rho = \rho'$, hence go to step (4). Otherwise set $b \leftarrow b + 1$ and $s_b(\rho') = s_i^C$.

(2) : For $j = i, \ldots, h$, compute the quantities $p_{ij}^b(\rho')$ and $\overline{p}_{ij}^b(\rho')$ by

maybe (h-1) ?

ρ' = current rate

$s_b(\rho')$ = start time of busy period with number b for current rate ρ'

$$p_{ij}^b(\rho') = \overbrace{C \sum_{a=i}^{j} (t_a^C - s_a^C)}^{\text{total bits sent during i,j busy periods}} - \overbrace{(t_j^C - s_i^C)\rho'}^{\text{total bits serviced}} \tag{3.2a}$$

difference

$$\overline{p}_{ij}^b(\rho') = C \sum_{a=i}^{j} (t_a^C - s_a^C) - (s_{j+1}^C - s_i^C)\rho' \tag{3.2b}$$

it is used to identify the busy periods for the specific ρ

Note that either,

- h is the smallest integer greater than i, that satisfies $\overline{p}^{b}_{ih}(\rho') < 0$, or
- $t^C_h = T$.

The maximum queue size over the b^{th} busy period is given by

$$q_b(\rho') = \max_{i \le j \le h} p^b_{ij}(\rho') \tag{3.3}$$

If $\overline{p}^{b}_{ih}(\rho') < 0$, then the time at which the busy period ends, $t_b(\rho')$ is given by,

$$t_b(\rho') = s^C_i + \frac{C\sum_{a=i}^{h-1}(t^C_a - s^C_a)}{\rho'}, \tag{3.4}$$

otherwise, set $t_b(\rho') = T$ if $t^C_h = T$. ? Probably $t_b(\rho') = s_i^C + \frac{C\sum_{a=i}^{b(c)}(t_a^C - s_a^C)}{\rho'}$

(3) : If $t_b(\rho') = T$, then go to step (4), otherwise go back to step (1).

(4) : Set $n(\rho') = b$. If $n(\rho') = 0$ then STOP. Otherwise compute the maximum queue length achieved for $\rho = \rho'$ as ↳ Can this ever occur?

$$q(\rho') = \max_{1 \le b \le n(\rho')} q_b(\rho')$$

If the maximum is achieved at $b = B$, then identify $(s^C_{i(B)}, t^C_{j(B)})$, such that $q_B(\rho') = p^B_{i(B)j(B)}(\rho')$. Then the burstiness curve satisfies the following equation for $\rho \in [\rho', \rho'')$, where ρ'' is computed in the next step.

$$\sigma(\rho) = C\sum_{a=i(B)}^{j(B)}(t^C_a - s^C_a) - \rho(t^C_{j(B)} - s^C_{i(B)}), \quad \rho' \le \rho < \rho'' \tag{3.5}$$

(5) : Let $\rho'' > \rho'$ be smallest value of ρ at which the B^{th} busy period splits apart into two or more busy periods. In order to compute this quantity, first compute the quantities $\rho''_{i,j}, j = i+1, \ldots, h-1$ for the B^{th} busy period given by

$$\rho''_{i,j} = \frac{C\sum_{a=i}^{j}(t^C_a - s^C_a)}{(s^C_{j+1} - s^C_i)}$$

Max Busy Period

Then,

$$\rho'' = \min[C, \min_{i \leq j \leq (h-1)} \rho''_{i,j}] \tag{3.6}$$

(6) : If $\rho'' = C$, then STOP. Otherwise replace ρ' by ρ'', set $b = 0$, $s_0(\rho') = 0$, $t_0(\rho') = 0$ and go back to step (1).

We now provide justifications for the steps in this algorithm:

- The quantities $p^b_{ij}(\rho')$ in step (2) are the queue sizes at the end of the interval $[s^C_i, t^C_j]$ while the quantities $\overline{p}^b_{ij}(\rho')$ are the queue sizes at the end of the interval $[s^C_i, s^C_{j+1}]$. Note that it is sufficient to obtain the queue sizes at the time instants at which the rate $m^C(t)$ changes, since the maximum and minimum queue sizes always occur at these rate change instants. Note that if $\overline{p}^b_{ih}(\rho') < 0$, then $t_b(\rho')$ satisfies the equation

$$C \sum_{a=i}^{h-1} (t^C_a - s^C_a) = (t_b(\rho') - s^C_i)\rho'$$

which leads to (3.4).

- In step (5), the quantity ρ''_{ij} satisfies the equation $j = (i+1), \ldots, (h-1)$ or $(h-2)$?

$$C \sum_{a=i}^{j} (t^C_a - s^C_a) - (s^C_{j+1} - s^C_i)\rho' = 0$$

and so is interpreted as the minimum value of ρ' at which the queue size at the end of the interval $[s^C_i, s^C_{j+1}]$ becomes zero, i.e., the B^{th} busy period ends at $t = s^C_{j+1}$. Note that it is sufficient to consider only the B^{th} busy period, since if any of the other busy perids were to break up, the maximum queue would still be achieved in the B^{th} busy period.

4. Computation of the Loss Curve

We assumed during the computation of the burstiness curve that there is an infinite amount of buffer space available in the single server queue with rate ρ. This implies that if the network reserves resources based on the burstiness curve, then the source does not experience any loss. However this can lead to excessive resource demands, since the burstiness curve is governed by the tail behavior of the source traffic statistics. In order to reduce resource consumption, the source may be willing to tolerate a small amount of loss. In order to realize this, it is necessary to obtain the burstiness curve under the constraint that at most β bits of buffer space is available in the single server queue with rate ρ. The plot of β vs ρ, for a fixed value of the loss ϵ, is known as the *loss curve*. In order to obtain the loss curve we first compute an ϵ vs β curve for different values of ρ. The loss curve can be obtained from this set of curves by fixing ϵ and then reading off the β values as ρ changes.

Once again we start with the waveform $m^C(t), 0 \leq t \leq T$, such that $m^C(t) = C, t \in [s_i^C, t_i^C], 1 \leq i \leq b(C)$ and $m^C(t) = 0$ otherwise. We will denote the number of bits lost in the interval $[t_1, t_2]$ when the buffer size is β and the rate of the single server queue is ρ, by $L(t_1, t_2, \beta, \rho)$, with the convention that if $t_1 = 0, t_2 = t$, then the notation $L(t, \beta, \rho)$ is used. The fraction of lost traffic $\epsilon(\beta, \rho), 0 \leq \epsilon \leq 1$ is given by

$$\epsilon(\beta, \rho) = \frac{L(T, \beta, \rho)}{M}, \quad 0 \leq \rho \leq C, \quad 0 \leq \beta \leq \sigma(\rho) \tag{4.1}$$

It is only necessary to consider the range $0 \leq \beta \leq \sigma(\rho)$, where $\sigma(\rho)$ is the burstiness curve, since if $\beta > \sigma(\rho)$, then $\epsilon(\beta, \rho) = 0$. Note that if $\beta = 0$, then

$$\epsilon(0, \rho) = \frac{C - \rho}{C} \quad 0 \leq \rho \leq C \tag{4.2}$$

since if $\beta = 0$, then the maximum amount of data that is able to get through without being dropped is

$$L(T, 0, \rho) = (C - \rho) \sum_{i=1}^{b(C)} (t_i^C - s_i^C)$$

It is also easy to see that

$$\epsilon(\beta, 0) = \begin{cases} 1 & \text{if } 0 \leq \beta < M \\ 0 & \text{if } \beta \geq M \end{cases} \qquad \frac{M-\beta}{M} \tag{4.3}$$

Consider the b^{th} busy period and let $[s_i^C, t_i^C]$, $[s_{i+1}^C, t_{i+1}^C]$ and $[s_{i+2}^C, t_{i+2}^C]$ be the first three intervals in that busy period for which $m^C(t) = 1$. We will obtain expressions for the lost data in the following three cases:

(1) : *Loss of data in the first interval, but not in the second and third:* Let x_1 be the time at which the queue length first exceeds β during the first interval. It is clear that

$$\beta = (C - \rho)(x_1 - s_i^C)$$

so that

$$x_1 = s_i^C + \frac{\beta}{C - \rho}$$

The amount of data lost is given by

$$\begin{aligned} L(s_i^C, t_i^C, \beta, \rho) &= (C - \rho)(t_i^C - x_1) \\ &= (C - \rho)(t_i^C - s_i^C) - \beta \\ &= \int_{s_i^C}^{t_i^C} (m(t) - \rho)dt - \beta \end{aligned} \tag{4.4}$$

(2) : *Loss of data in the second interval, but not in the first or third:* Let the queue length at time t be denoted by $q(t)$. Let x_2 be the time at which the queue length first exceeds β during the second interval. It is clear that

$$\beta = q(s_{i+1}^C) + (C - \rho)(x_2 - s_{i+1}^C)$$

so that

$$x_2 = s_{i+1}^C + \frac{\beta - q(s_{i+1}^C)}{C - \rho}$$

The amount of data lost is given by

$$\begin{aligned} L(s_{i+1}^C, t_{i+1}^C, \beta, \rho) &= (C - \rho)(t_{i+1}^C - x_2) \\ &= (C - \rho)(t_{i+1}^C - s_{i+1}^C) - \beta + q(s_{i+1}^C) \end{aligned} \tag{4.5}$$

Substituting

$$q(s_{i+1}^C) = (C - \rho)(t_i^C - s_i^C) - \rho(s_{i+1}^C - t_i^C)$$

we obtain

$$L(s_{i\cancel{+1}}^C, t_{i+1}^C, \beta, \rho) = \int_{s_i^C}^{t_{i+1}^C} (m(t) - \rho)dt - \beta \tag{4.6}$$

(3) : *Loss of data in the first and second intervals, but not in the third:* From (4.4), the amount of data lost in the first interval is given by

$$L(s_i^C, t_i^C, \beta, \rho) = (C - \rho)(t_i^C - s_i^C) - \beta \tag{4.7}$$

while the amount of data lost in the second interval is given by (4.5) with

$$q(s_{i+1}^C) = \beta - \rho(s_{i+1}^C - t_i^C) \tag{4.8}$$

Substituting (4.8) into (4.5), we obtain

$$L(s_{i+1}^C, t_{i+1}^C, \beta, \rho) = (C - \rho)(t_{i+1}^C - s_{i+1}^C) - \rho(s_{i+1}^C - t_i^C) \tag{4.9}$$

Combining (4.7) with (4.9), we obtain

$$L(s_i^C, t_{i+1}^C, \beta, \rho) = \int_{s_i^C}^{t_{i+1}^C} (m(t) - \rho)dt - \beta \tag{4.10}$$

(4) : *Loss of data in the first and third intervals, but not in the second:* Once again the amount of data lost in the first interval is given by

$$L(s_i^C, t_i^C, \beta, \rho) = (C - \rho)(t_i^C - s_i^C) - \beta \tag{4.11}$$

while the amount of data lost in the third interval is given by

$$L(s_{i+1}^C, t_{i+1}^C, \beta, \rho) = (C - \rho)(t_{i+2}^C - s_{i+2}^C) - \beta + q(s_{i+2}^C)$$

Note that

$\beta -$

$$q(s_{i+2}^C) = (C - \rho)(t_{i+1}^C - s_{i+1}^C) - \rho(s_{i+1}^C - t_i^C) - \rho(s_{i+2}^C - t_{i+1}^C)$$

so that

$$L(s_i^C, t_{i+2}^C, \beta, \rho) = \int_{s_i^C}^{t_{i+2}^C} (m(t) - \rho)dt - \beta \tag{4.12}$$

The following result now follows (a related result was given in [WoVa] without proof).

Proposition 3. *Let $\{s_{i'}^C\}$ be a sub-sequence of the sequence $\{s_i^C\}$ such that the busy periods that start at the instants $s_{i'}^C$ experience cell loss. Furthermore let $\{\tau_{i'}^C\}$ be the time instant at which the last interval (with input rate C) experiencing cell loss ends, for the busy period that starts at $s_{i'}^C$. Then*

$$L(T,\beta,\rho) = \sum_{i'} [\int_{s_{i'}^C}^{\tau_{i'}^C} (m(t)-\rho)dt - \beta], \quad 0 \le \rho \le C, \quad 0 \le \beta \le \sigma(\rho) \tag{4.13}$$

Proof: The proof is by straighforward induction from equations (4.4), (4.6), (4.10) and (4.12). ∎

Proposition 3 has the following Corollary.

Corollary 1. *For a fixed value of ρ, the ϵ vs β curve is piecewise linear and convex.*

Proof: We first show that, for the b^{th} busy period that experiences loss, the time τ_b^C in (4.13) is the same as the time at which the maximum queue length was achieved in that busy period. Assume that the maximum queue length q_b was achieved at time t_b. Then the queue length p_i at the end of the i^{th} interval with rate C in that busy period (that occurs after t_b), satisfies the equation (assuming that there are l such intervals),

$$p_i = q_b + \int_{t_b}^{t_i^C} (m(t)-\rho)dt \quad i = 1, \ldots, l$$

Since by assumption $p_i < q_b, i = 1, \ldots, l$, it follows that

$$\int_{t_b}^{t_i^C} (m(t)-\rho)dt < 0 \quad i = 1, \ldots, l \tag{4.14}$$

Consider the case when the value of β decreases to the point such that $\beta < q_b$, then the b^{th} busy period will experience traffic loss. In this case $q_b = \beta$ and the queue lengths in the succeeding intervals in that busy period satisfy the equation

$$p_i = \beta + \int_{t_b}^{t_i^C} (m(t)-\rho)dt \quad i = 1, \ldots, l \tag{4.15}$$

Combining (4.15) with (4.14) it follows that $p_i < \beta, i = 1, \ldots, l$, which implies that the last interval that experiences loss ends at t_b, i.e.

$$\tau_b^C = t_b. \tag{4.16}$$

Consider the case when β decreases from $\sigma(\rho)$ towards zero. The interval $[0, \sigma(\rho)]$ can be divided into sub-intervals $[\beta_{i-1}(\rho), \beta_i(\rho)], i = 1, \ldots, g(\rho)$, such that $\beta_0(\rho) = 0, \beta_g(\rho) = \sigma(\rho)$ and the number of busy periods experiencing loss $\overline{n}_i(\rho)$, is constant over the i^{th} such interval. It is clear from (4.13) and (4.16) that the slope of the $L(T, \beta, \rho)$ vs β curve is also $\overline{n}_i(\rho)$ in the interval $[\beta_{i-1}(\rho), \beta_i(\rho)]$ which implies piecewise linearity. Convexity follows from the fact that $\overline{n}_i(\rho)$ increases as i decreases, since there are more busy periods that experience loss for smaller values of β. ∎

Recall that the number of busy periods $n(\rho)$ (for $\beta \geq \sigma(\rho)$) was computed in the algorithm for the burstiness curve in the last section. Note that $\overline{n}_i(\rho) > n(\rho), 1 \leq i \leq g(\rho) - 1$ and $\overline{n}_g(\rho) = n(\rho)$, since data loss can lead to an increase in the number of busy periods, even if ρ is fixed. In fact $n_1(\rho) = b(C), 0 \leq \rho \leq C$. Based on the results in Proposition 3 and Corollary 1 a quick approximation to the exact curve can be readily obtained from the data produced by the algorithm for the burstiness curve in the following way:

Proposition 4. *Re-arrange the sequence of maximum queue length in the b^{th} busy period, $\{q_b(\rho)\}_{b=1}^{n(\rho)}$, obtained in (3.3), in increasing order, say $\{q_{[b]}(\rho')\}_{b=1}^{n(\rho)}$. Set $\beta_i = q_{[i]}(\rho'), 1 \leq i \leq n(\rho)$. Then the data lost for $\beta \in (\beta_{k-1}, \beta_k]$ is approximately given by*

$$L(T, \beta, \rho) \approx \sum_{b=k}^{n(\rho)} q_{[b]}(\rho) - (n(\rho) - k + 1)\beta. \tag{4.17}$$

Proof: When $\beta \in (\beta_{k-1}, \beta_k]$, then only the busy periods for which the maximum queue length $q_b(\rho)$ exceeds β_{k-1} will experience loss. Ignoring the fact that more busy periods are created due to decreasing β, there are approximately $n(\rho) - k + 1$ such busy periods, with maximum queue lengths given by $\{q_{[b]}(\rho)\}_{b=k}^{n(\rho}$. Equation (4.17) then directly follows by applying (4.13) and (4.16). ∎

This approximation should work well for small values of loss, since the effect of increase in busy periods is not apparent until β becomes sufficiently small. The exact algorithm now follows.

(0) : Initialize the algorithm by setting $\beta' = \sigma(\rho)$, $A(T, \beta', \rho) = 0$, $b = 0$, $c = 0$, $s_0(\beta') = 0$ and $t_0(\beta') = 0$. Also, let Π be the ordered list of maximum queue lengths such that

$$\Pi = \{q_{[1]}(\rho), \ldots, q_{[n(\rho)]}(\rho)\}$$

(1) : In the rate function $m^C(t), 0 \leq t \leq T$, find the smallest time $s_i^C > t_b(\beta')$ such that $m^C(t) = C$, If no such s_i^C exists, then all the busy periods have been identified for $\beta = \beta'$, hence go to step (4). Otherwise set $b \leftarrow b + 1$ and $s_b(\beta') = s_i^C$.

(2) : For $j = i, \ldots, h$, recursively compute the quantities $r_{ij}^b(\beta')$ and $\overline{r}_{ij}^b(\beta')$ as follows: Initialize

$$r_{ii}^b(\beta') = C(t_i^C - s_i^C) - \rho(t_i^C - s_i^C)$$

$$\overline{r}_{ii}^b(\beta') = C(t_i^C - s_i^C) - \rho(s_{i+1}^C - s_i^C)$$

Assume that $r_{ij}^b(\beta')$ and $\overline{r}_{ij}^b(\beta')$ have already been computed, then

$$r_{i,j+1}^b(\beta') = \min[\overline{r}_{ij}^b(\beta') + (C - \rho)(s_{j+1}^C - t_{j+1}^C), \beta'] \tag{4.18}$$

$$\overline{r}_{i,j+1}^b(\beta') = r_{i,j+1}^b(\beta') - \rho(s_{j+2}^C - t_{j+1}^C) \tag{4.19}$$

Note that either,

- h is the smallest integer equal to or greater that i, that satisfies $\overline{r}_{ih}^b(\rho') < 0$, or
- $t_h^C = T$.

Compute the quantities $p_{ij}^b(\rho), j = i, \ldots, h$ as given by equation (3.2a). If $\max_{1 \leq j \leq h} r_{ij}^b(\beta') < \beta'$, then the b^{th} busy period does not experience any loss. In this case if $\max_{1 \leq j \leq h} r_{ij}^b(\beta')$ is not already in Π, then add it to that list. If $\max_{1 \leq j \leq h} r_{ij}^b(\beta') = \beta'$, then the b^{th} busy period does experiences loss. In this case, let $c \leftarrow c + 1$, and

$$A(T, \beta', \rho) = A(T, \beta', \rho) + \max_{i \leq j \leq h} p_{ij}^b(\rho) \tag{4.20}$$

If $\bar{r}^b_{ih}(\beta') < 0$, then the time at which the busy period ends, $t_b(\beta')$ is given by,

$$t_b(\beta') = s_i^C + \frac{r^b_{i,(h-1)}(\beta')}{\rho},$$

otherwise, set $t_b(\beta') = T$ if $t_h^C = T$.

(3) : If $t_b(\rho') = T$, then go to step (4), otherwise go back to step (1).

(4) : Set $n(\beta') = c$. If $n(\rho') = 0$ then STOP. Otherwise the equation for the loss in the range $\beta \in (\beta'', \beta']$ is given by

$$L(T, \beta', \rho) = A(T, \beta', \rho) - n(\beta')\beta \tag{4.21}$$

where β'' is obtained by deleting β' from $\mathit{I\!I}$, and choosing the largest remaining number left.

(5) : If I is empty, then STOP. Otherwise set $\beta' \leftarrow \beta''$, $A(T, \beta', \rho) = 0$, $b = 0$, $c = 0$, $s_0(\rho') = 0$, $t_0(\rho') = 0$ and go back to step (1).

We now provide justifications for the steps in this algorithm:

- Note that from Proposition 3 and Corollary 1, the slope of the curve remains fixed as long as the number of busy periods with losses does not change. The algorithm basically works by keeping track of the number of the busy periods, and the maximum queue lengths that are achieved in those busy periods. These maximum queue lengths are put in the list I, which expands as β decreases, since there are more busy periods created.
- The quantities $r^b_{ij}(\beta)$ are the queue sizes in the b^{th} busy periods, at the end of the j^{th} interval with rate C, while the quantities $\bar{r}^b_{ij}(\beta)$ are the queue sizes at the end of the j^{th} interval with rate zero.

Even though the algorithm computes the ϵ vs β curve for $\epsilon \in [0, \frac{C-\rho}{C}]$, in most practical situations an application will not be able to tolerate a loss of more than a few percent. Hence we may stop the algorithm when ϵ reaches that threhold. Once the ϵ vs β curves has been computed, the β vs ρ curves (i.e. the loss curves) can be easily got, by fixing ϵ and obtaining the value β, for each ρ, at which the loss equals ϵ. In fact the algorithms for the burstiness curve and the ϵ vs β curve can be run in alternation in the following way to obtain a three-dimensional plot of ϵ vs β vs ρ.

(a) : Compute the value $\sigma(\rho_k)$ for $\rho = \rho_k$ by using the burstiness algorithm.

(b) : Fix $\rho = \rho_k$ and obtain the ϵ vs β curve using the algorithm in this section. The maximum queue lengths computed in (a) can be used as input.

(c) : Set ρ to the next higher value ρ_{k+1} and go back to step (a).

5. Conclusions

In this paper we provided several efficient algorithms for computing the characteristic curves for video sources. We envision that these algorithms would form part of a software tool that can be used to process video traces and extract useful information from them.

REFERENCES

[Cr] R. L. Cruz, "A calculus for network delay, Part I: Network elements in isolation," *IEEE Transactions on Information Theory*, Vol 37, No. 1, pp. 114-131, (Jan, 1991).

[Gr] M. Graf, "Traffic shaping of VBR video in ATM end systems," *Proc. 4th Open Workshop on High Speed Networks*, Brest, France, (Sept, 1994).

[LoVa] S. Low and P.P. Varaiya, "A simple theory of traffic and resource allocation in ATM," *INFO~~COM~~'91*, pp. 1633-1637, (1991).
GLOBECOM

[Va] S. Varma, "Specifying traffic descriptor and QOS parameters for video sources," *In Preparation*, (1994).

[WoVa] M.K. Wong and P.P. Varaiya, "A deterministic fluid model for cell loss in ATM networks," *INFOCOM'93*, pp. 395-44, (1993).