Digital Fountain Technical Report DF2002-03-001

Constant-Rate Server Output in WEBRC

Vivek K Goyal
39141 Civic Center Drive, Suite 300
Fremont, CA 94538
vivek@digitalfountain.com

March 12, 2002
(Revised May 17, 2002)

Abstract

With the initial design of WEBRC, as described in the -00 and -01 IETF RMT building block drafts [1, 2], the server output rate varies in time with period `TSD` (recommended to be 10 seconds). The variation is quite significant; when the recommended multiplicative decrease value `P` = 0.75 is used, it causes the average output rate to be *at least* 13.1% less than the peak rate. This document describes how increasing the number of wave channels by one and changing the shape of the initial portion of the wave makes it possible to have constant-rate server output. (A possibility that had been discussed but is not described in this document would not increase the number of wave channels and would make the variation *at most* 13.1%.)

For the sake of brevity, this document is not self-contained. Those unfamiliar with WEBRC should see [2, 3] for terminology.

1 Basic Principles

The cumulative server output in the original design of WEBRC is shown in Fig. 1. The basic idea pursued here is to fill in the gaps between saw teeth to make the cumulative output have constant rate SR. In addition, to prevent join time outs on the top wave, we would like each wave channel to have a minimum rate of BCR for the full duration of every time slot in which it is active. (We will abandon the latter requirement when SR/BCR is small.) An example of the cumulative server output is shown in Fig. 2.

Suppose in the original design a wave channel starts in the middle of time slot 1. Use time coordinates such that this time slot is $t \in [\mathtt{TSD}, 2 * \mathtt{TSD}]$. At first glance, it seems that the alteration is merely as follows:

- Compute the time t_0 at which the wave should crest. (This is slightly different than when the wave would start in the original design. In the original design, the time at which a wave starts is such that the rate of the top wave plus the rates of the $N-1$ lower waves plus the rate of base channel equals SR.[1] In the new design, there is one fewer wave channel "below" the crested wave and the sum should equal $\mathtt{SR} - \mathtt{BCR}$.) t_0 would be in time slot 1.
- Fill in from t_0 back to the beginning of time slot 1.
- Fill in (with slightly different form) from the end of time slot 0 back to time $t_1 = t_0 - \mathtt{TSD}$.
- The rate equals BCR from the beginning of time slot 0 to time t_1.

This is generally true, i.e., true for most values of SR/BCR. But for some values of SR/BCR, the calculation of t_0 gives a value greater than $2 * \mathtt{TSD}$. Then, in the computation of t_0 there is a different sum and it should equal a different value $(\mathtt{SR} - 2 * \mathtt{BCR})$. The t_0 obtained with the computation implied above is shown in Fig. 3. For reasonable (large) values of SR, $t_0/\mathtt{TSD}$ is rarely larger than 2, so it is easy to miss the fact that there are multiple cases. Specifically, t_0 is larger than TSD for the rare values of SR such that

$$1 + \frac{1}{\mathtt{P}} + \frac{1}{\mathtt{P}^2} \cdots + \frac{1}{\mathtt{P}^k} \; < \; \frac{\mathtt{SR}}{\mathtt{BCR}} - 1 \; < \; p + 1 + \frac{1}{\mathtt{P}} + \frac{1}{\mathtt{P}^2} \cdots + \frac{1}{\mathtt{P}^k}$$

for an integer k.

2 Computation of Wave Channel Rates

The sender and receiver should compute the number of active wave channels as follows:

$$\mathtt{N} \; = \; \left\lceil \log_{1/\mathtt{P}} \left(1 + \frac{1}{\mathtt{P}} \left(\frac{1}{\mathtt{P}} - 1 \right) \frac{\mathtt{SR}}{\mathtt{BCR}} \right) \right\rceil - 1. \tag{1}$$

This is derived from the following fact: N should be smallest integer such that the rate at the end of each time slot, without truncation of waves,

$$\mathtt{BCR} \left(\mathtt{P} + 1 + \frac{1}{\mathtt{P}} + \cdots + \frac{1}{\mathtt{P}^{\mathtt{N}-1}} \right),$$

is at least SR. This is the only calculation given in this document that is relevant to the receiver.

[1] All rates are in packets per second.

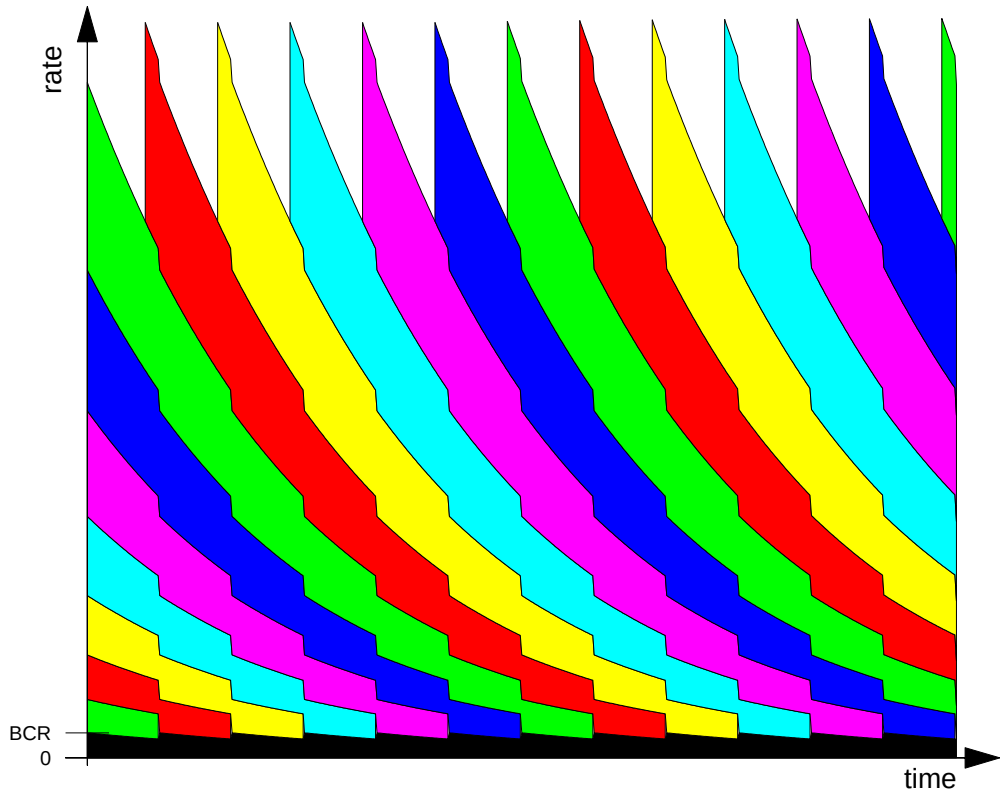


Figure 1: Cumulative output in the original design.

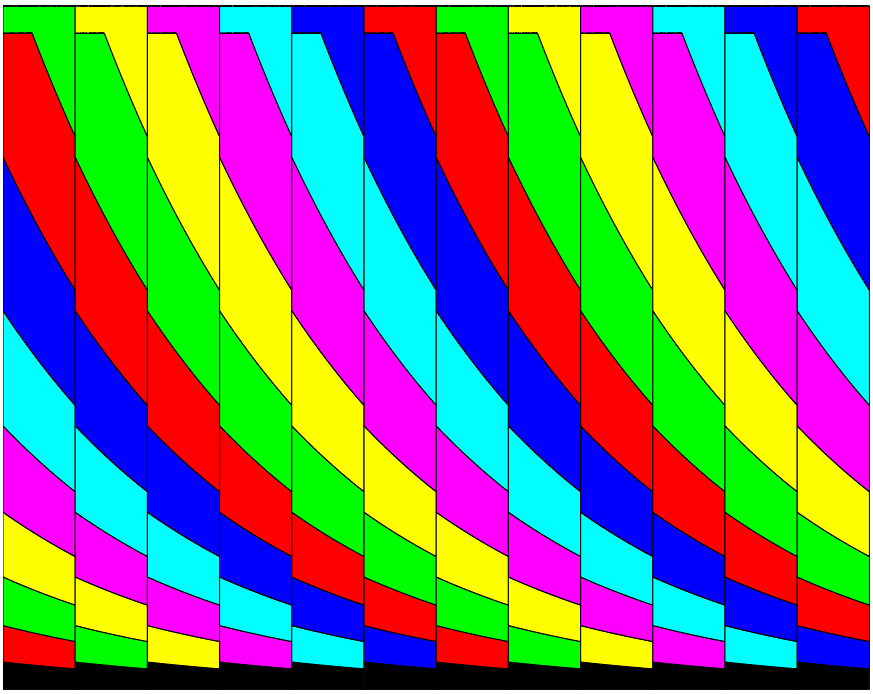


Figure 2: Example of cumulative output with the new design.

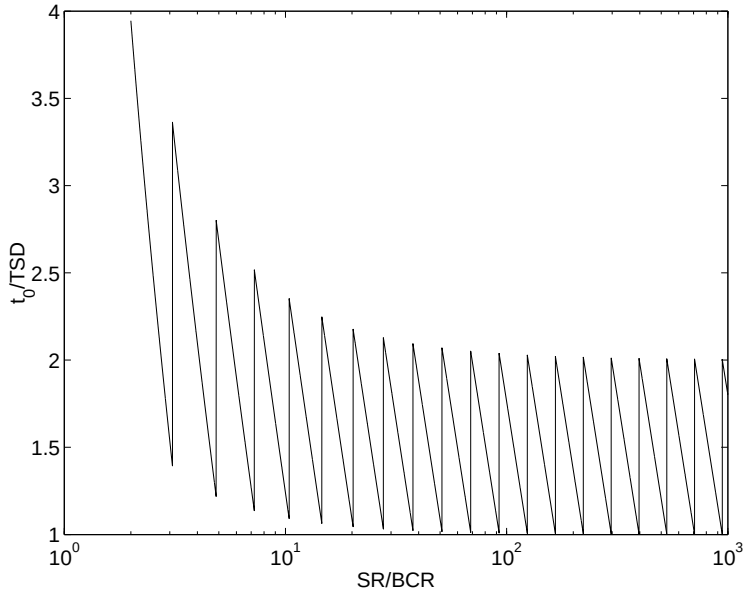


Figure 3: The time after which the rate decays exponentially is denoted t_0. The plot shows that t_0 as defined in (3) may exceed $2*\texttt{TSD}$. This is a roundabout demonstration that sometimes more than two time slots must be affected by the change to constant rate output. The values of $\texttt{SR}/\texttt{BCR} \geq 5.2$ are divided into "generic" and "secondary" cases based on whether $t_0/\texttt{TSD}$ exceeds 2.

The simplest way to make further calculations to proceed under the assumption that only two time slots are affected by making the server output constant. Full details on the packet scheduling are given for this "generic" case in Section 2.1. Sections 2.2 and 2.3 summarize other cases. Complete Matlab code is given for all cases in Appendix B.

2.1 The generic case

A convenient intermediate variable is the maximum wave channel rate MWCR. In the generic case, at the crest of the wave there are $\texttt{N}-2$ lower waves and the base channel below, plus a wave at rate BCR above.[2] Thus,

$$\texttt{MWCR}\left(1+\texttt{P}+\texttt{P}^2+\cdots+\texttt{P}^{\texttt{N}-1}\right) \;=\; \texttt{SR}-\texttt{BCR},$$

or

$$\texttt{MWCR} \;=\; \frac{1-\texttt{P}}{1-\texttt{P}^{\texttt{N}}}(\texttt{SR}-\texttt{BCR}). \tag{2}$$

Noting that $\texttt{N}*\texttt{TSD}-t_0$ is the time for the rate on a channel to decay from MWCR to BCR, the breakpoint after which the wave is a simple exponential decay is

$$t_0 \;=\; \texttt{TSD}\left(\texttt{N}-\log_{1/\texttt{P}}\frac{\texttt{MWCR}}{\texttt{BCR}}\right). \tag{3}$$

This yields $t_0 \geq \texttt{TSD}$. The hypothesis of the generic case is true as long as $t_0 \leq 2*\texttt{TSD}$ is satisfied.

[2] "Below" and "above" are with respect to diagrams like Fig 2.

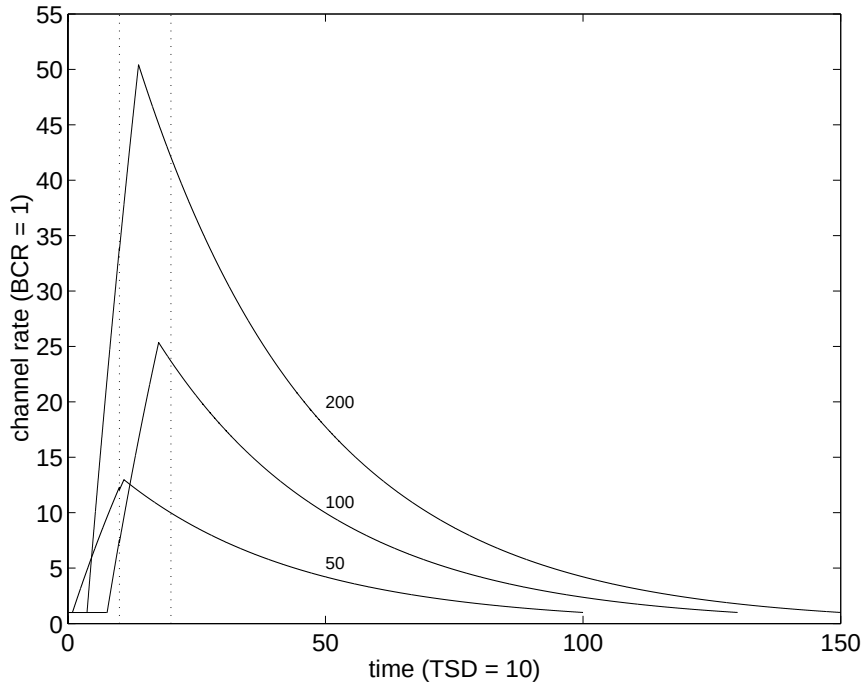


Figure 4: Waves designed with (4). The curve labels are values of SR. The dotted lines are at times TSD and $2 * \mathtt{TSD}$.

For the generic case, the rate $R(t)$ on the wave channel at time $t \in [0, \mathtt{N} * \mathtt{TSD}]$ is given by

$$R(t) \;=\; \begin{cases} \mathtt{BCR}, & 0 \le t < t_0 - \mathtt{TSD}; \\ \mathtt{SR} - \mathtt{BCR} * \mathtt{P}^{t/\mathtt{TSD}} - \mathtt{BCR} * \dfrac{(1/\mathtt{P})^{\mathtt{N}-1} - 1}{(1/\mathtt{P}) - 1} * \mathtt{P}^{t/\mathtt{TSD}-1}, & t_0 - \mathtt{TSD} \le t < \mathtt{TSD}; \\ \mathtt{SR} - \mathtt{BCR} - \mathtt{BCR} * \dfrac{(1/\mathtt{P})^{\mathtt{N}-1} - 1}{(1/\mathtt{P}) - 1} * \mathtt{P}^{t/\mathtt{TSD}-1}, & \mathtt{TSD} \le t < t_0; \\ \mathtt{BCR} * (1/\mathtt{P})^{\mathtt{N}-t/\mathtt{TSD}}, & t_0 \le t \le \mathtt{N} * \mathtt{TSD}. \end{cases} \tag{4}$$

Fig. 4 shows some examples of waves generated with (4) and Fig. 2 shows how stacking the waves makes the cumulative rate constant.

Since packet transmissions are discrete events, the fluid model for the wave channel rate (4) and the base channel rate

$$R_{\mathrm{base}}(t) \;=\; \mathtt{BCR} * \mathtt{P}^{t/\mathtt{TSD}}, \qquad t \in [0, \mathtt{TSD}) \tag{5}$$

must be converted to times for packet transmissions. According to the fluid model, the number of packets sent in one time slot is $K = \mathtt{TSD} * \mathtt{SR}$. We henceforth assume that this quantity is integral. This allows the scheduling of packet transmissions on all channels in all time slots to be described in terms of a precomputation involving one period of the base channel and one wave.

The conversion of the fluid-model rates to packet transmission times is demonstrated graphically in Fig. 5. Build a curve with support $[0, (\mathtt{N}+1) * \mathtt{TSD}]$ by taking one period of the base channel rate followed by the time-reversed rate of one wave.[3] The area under this curve is the total number of

[3]Though it may seem arbitrary, there are reasons for taking the base channel period first and using the time-

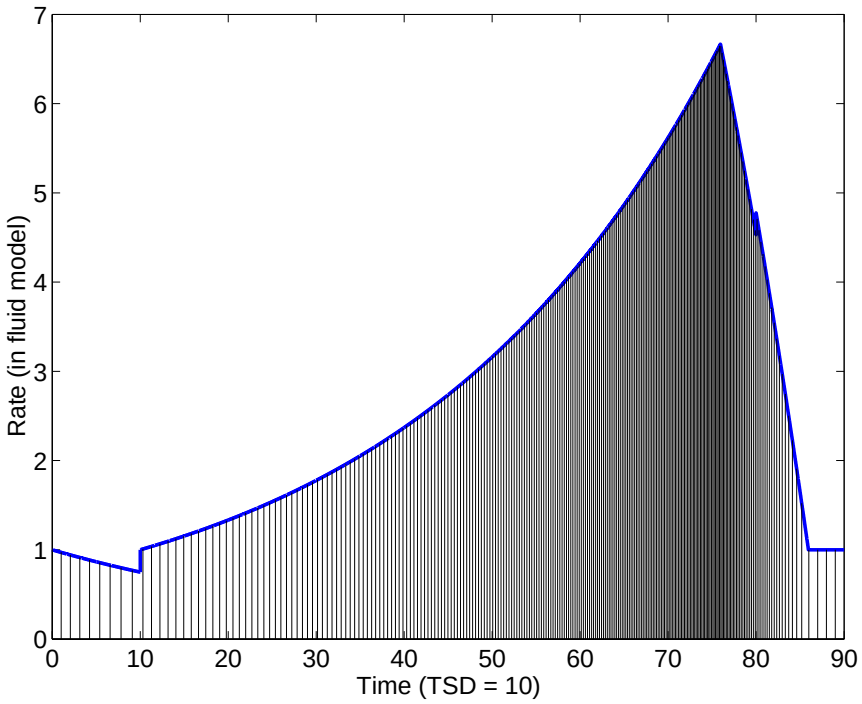


Figure 5: The conversion between rates from the fluid model and packet transmission times with $\texttt{TSD} = 10$, $\texttt{SR} = 25$, $\texttt{BCR} = 1$, and $\texttt{P} = 0.75$. The plotted curve corresponds to one period of the base channel followed by one time-reversed wave. Vertical lines indicate packet transmission times. They divide the area under the curve into $\texttt{TSD} * \texttt{SR} = 250$ unit-area regions.

packets to be transmitted in one time slot. Each packet transmission corresponds to one unit of fluid, so the packet transmission times should divide this area into K unit-area regions.

Mathematically, determining the transmission times that make for equal areas in Fig. 5 corresponds to solving for the times at which the integral of the rate curve crosses integer boundaries. The first base channel packet is sent at time $b_0 = 0$. The next is sent at the time b_1 that satisfies

$$1 \;=\; \int_0^{b_1} R_{\text{base}}(\tau)\, d\tau \;=\; \int_0^{b_1} \texttt{BCR} * \texttt{P}^{\tau/\texttt{TSD}}\, d\tau.$$

Continuing this process, a total of $L = \lceil \frac{\texttt{BCR}*\texttt{TSD}(1-\texttt{P})}{-\ln \texttt{P}} \rceil$ packets are sent at times

$$b_k \;=\; \texttt{TSD} \log_{\texttt{P}} \left(1 + \frac{\ln \texttt{P}}{\texttt{BCR} * \texttt{TSD}} * k \right), \qquad k = 0,\, 1,\, \ldots,\, L-1. \tag{6}$$

Conceptually, this process is continued to find all of the wave channel transmission times. The key differences are integrating from the end of the wave rather than the beginning and accounting

reversed wave. Taking the base channel first makes the first packet of each time slot a base channel packet; thus, every receiver, regardless of rate, learns of time slot boundaries as soon as possible. Using the time-reversed wave causes the first packet of a wave to occur as late as possible while being consistent with the fluid model; this is important because receivers cannot join a wave until the indication of a new time slot reveals that the wave has started. Finally, with this configuration it is possible to precompute the b_ks and s_ks that depend on TSD, BCR, and P but are independent of SR.

for the difference between the number of base channel packets per time slot according to the fluid model,

$$I_0 \;=\; \int_0^{\mathtt{TSD}} R_{\text{base}}(\tau)\,d\tau \;=\; \int_0^{\mathtt{TSD}} \mathtt{BCR} * \mathtt{P}^{\tau/\mathtt{TSD}}\,d\tau \;=\; \frac{\mathtt{BCR} * \mathtt{TSD}(1-\mathtt{P})}{-\ln \mathtt{P}},$$

and the number of transmitted packets $L = \lceil I_0 \rceil$.

Let

$$S(s) \;=\; \int_{\mathtt{N}*\mathtt{TSD}-s}^{\mathtt{N}*\mathtt{TSD}} R(\tau)\,d\tau.$$

(s is time running backward from the end of the wave.) $S(s)$ is the fluid-model approximation to the number of packets to be sent in the last s seconds of the wave. The desired transmission times (measured backward from the end of the wave) are the solutions to

$$I_0 + S(s_k) \;=\; k, \qquad k = L,\, L+1,\, \ldots,\, K-1. \tag{7}$$

Solving (7) for the s_ks is complicated by the fact that $R(t)$, $t \in [0, \mathtt{N} * \mathtt{TSD}]$ is described separately on four subintervals (see (4)). The remainder of this section details the computations of the s_ks.

For the last time range in (4), the integral of $R(t)$ has a simple form:

$$S(s) \;=\; \frac{\mathtt{BCR} * \mathtt{TSD}}{-\ln \mathtt{P}} \left(\left(\frac{1}{\mathtt{P}} \right)^{s/\mathtt{TSD}} - 1 \right) \qquad \text{for } s \in [0, \mathtt{N} * \mathtt{TSD} - t_0].$$

Thus for $k = L$ and some larger values of k, (7) becomes

$$I_0 + \frac{\mathtt{BCR} * \mathtt{TSD}}{-\ln \mathtt{P}} \left(\left(\frac{1}{\mathtt{P}} \right)^{s_k/\mathtt{TSD}} - 1 \right) \;=\; k. \tag{8}$$

The range of k for which (8) is valid is determined based on the number of fluid-model base channel packets I_0 and the number of fluid-model packets sent in the last $\mathtt{N} * \mathtt{TSD} - t_0$ seconds of a wave. The latter quantity is given by

$$I_1 \;=\; \int_{t_0}^{\mathtt{N}*\mathtt{TSD}} R(\tau)\,d\tau \;=\; S(\mathtt{N} * \mathtt{TSD} - t_0) \;=\; \frac{\mathtt{BCR} * \mathtt{TSD}}{-\ln \mathtt{P}} \left(\left(\frac{1}{\mathtt{P}} \right)^{\mathtt{N} - t_0/\mathtt{TSD}} - 1 \right).$$

The sum of the number base channel packets in one time slot and the number of wave channel packets in the portion of the wave under consideration is denoted $K_1 = \lceil I_0 + I_1 \rceil$. Thus (8) is solved to obtain

$$s_k \;=\; \mathtt{TSD} * \log_{1/\mathtt{P}} \left(1 - \frac{\ln \mathtt{P}}{\mathtt{BCR} * \mathtt{TSD}} \,(k - I_0) \right), \qquad k = L,\, L+1,\, \ldots,\, K_1 - 1. \tag{9}$$

Computing and inverting $S(s)$ gets uglier from here on. For $s \in (\mathtt{N} * \mathtt{TSD} - t_0, (\mathtt{N}-1)\mathtt{TSD}]$, $S(s)$ has a fixed component I_1 plus

$$\begin{aligned} \int_{t_0-\tau}^{t_0} R(t)\,dt \;&=\; \int_{t_0-\tau}^{t_0} \left(\mathtt{SR} - \mathtt{BCR} - \mathtt{BCR} * \frac{(1/\mathtt{P})^{\mathtt{N}-1} - 1}{(1/\mathtt{P}) - 1} * \mathtt{P}^{t/\mathtt{TSD}-1} \right) dt \\ &=\; \underbrace{(\mathtt{SR} - \mathtt{BCR})}_{A}\,\tau - \underbrace{\mathtt{BCR} * \frac{(1/\mathtt{P})^{\mathtt{N}-1} - 1}{(1/\mathtt{P}) - 1} * \frac{\mathtt{TSD}}{-\ln \mathtt{P}} * \mathtt{P}^{t_0/\mathtt{TSD}-1} \left(\mathtt{P}^{-\tau/\mathtt{TSD}} - 1 \right)}_{B}, \end{aligned}$$

where $\tau = s - (\mathtt{N} * \mathtt{TSD} - t_0)$. The total fluid contribution of the latter for this portion of the wave is

$$I_2 \;=\; \int_{\mathtt{TSD}}^{t_0} R(t)\,dt \;=\; A(t_0 - \mathtt{TSD}) - B\left(\mathtt{P}^{-(t_0-\mathtt{TSD})/\mathtt{TSD}} - 1\right).$$

Thus the number of packets for one period of the base channel plus the number of packets for the last $(\mathtt{N}-1)\mathtt{TSD}$ seconds of a wave is $K_2 = \lceil I_0 + I_1 + I_2 \rceil$ and scheduling the packets for $s \in (\mathtt{N} * \mathtt{TSD} - t_0, (\mathtt{N}-1)\mathtt{TSD}]$ amounts to solving

$$I_0 + I_1 + A\tau_k - B(\mathtt{P}^{-\tau_k/\mathtt{TSD}} - 1) \;=\; k, \qquad k = K_1,\, K_1+1,\, \ldots,\, K_2 - 1 \tag{10}$$

for the τ_ks. There is no elementary closed form solution; techniques for finding approximate solutions are described in Appendix A. The τ_ks in this calculation are transmission times measured backward in time from time t_0. We thus obtain $s_k = \tau_k + \mathtt{N} * \mathtt{TSD} - t_0$, $k = K_1,\, K_1+1,\, \ldots,\, K_2 - 1$.

The calculations are very similar for the next interval, $s \in ((\mathtt{N}-1)\mathtt{TSD}, (\mathtt{N}+1)\mathtt{TSD} - t_0]$. We are interested in solutions to

$$S(\tau + (\mathtt{N}-1)\mathtt{TSD}) \;=\; I_0 + I_1 + I_2 + \int_{\mathtt{TSD}-\tau}^{\mathtt{TSD}} R(t)\,dt \;=\; k \qquad \text{for } \tau \in [0,\, 2 * \mathtt{TSD} - t_0].$$

Integrating yields

$$I_0 + I_1 + I_2 + A'\tau_k - B'(\mathtt{P}^{-\tau_k/\mathtt{TSD}} - 1) \;=\; k \tag{11}$$

where $A' = \mathtt{SR}$ and

$$B' \;=\; \frac{\mathtt{BCR} * \mathtt{TSD}}{-\ln \mathtt{P}} * \left(\mathtt{P} + \frac{(1/\mathtt{P})^{\mathtt{N}-1} - 1}{(1/\mathtt{P}) - 1}\right).$$

The range of integers k for which we solve (11) depends on how many packets are transmitted in this time interval. Paralleling the previous calculations, let

$$I_3 \;=\; \int_{t_0-\mathtt{TSD}}^{\mathtt{TSD}} R(t)\,dt \;=\; \mathtt{SR}(2 * \mathtt{TSD} - t_0) - \frac{\mathtt{BCR} * \mathtt{TSD}}{-\ln \mathtt{P}} * \left(\mathtt{P} + \frac{(1/\mathtt{P})^{\mathtt{N}-1} - 1}{(1/\mathtt{P}) - 1}\right) * \left(\mathtt{P}^{t_0/\mathtt{TSD}-2} - 1\right)$$

and $K_3 = \lceil I_0 + I_1 + I_2 + I_3 \rceil$. Then (11) is solved for $k = K_2,\, K_2+1,\, \ldots,\, K_3 - 1$. Of course, (11) is of the same form as (10) and solutions can be approximated similarly. This process yields $s_k = \tau_k + (\mathtt{N}-1) * \mathtt{TSD}$, $k = K_2,\, K_2+1,\, \ldots,\, K_3 - 1$.

In the final interval $s \in ((\mathtt{N}+1)\mathtt{TSD} - t_0, \mathtt{N} * \mathtt{TSD}]$, the fluid-model rate is BCR so the time between packets is $1/\mathtt{BCR}$. To correctly patch together the intervals, we seek solutions to

$$I_0 + I_1 + I_2 + I_3 + \mathtt{BCR} * \tau_k \;=\; k, \qquad k = K_3,\, K_3+1,\, \ldots,\, K - 1. \tag{12}$$

The solutions are

$$\tau_k \;=\; \frac{1}{\mathtt{BCR}}\,(k - I_0 - I_1 - I_2 - I_3)$$

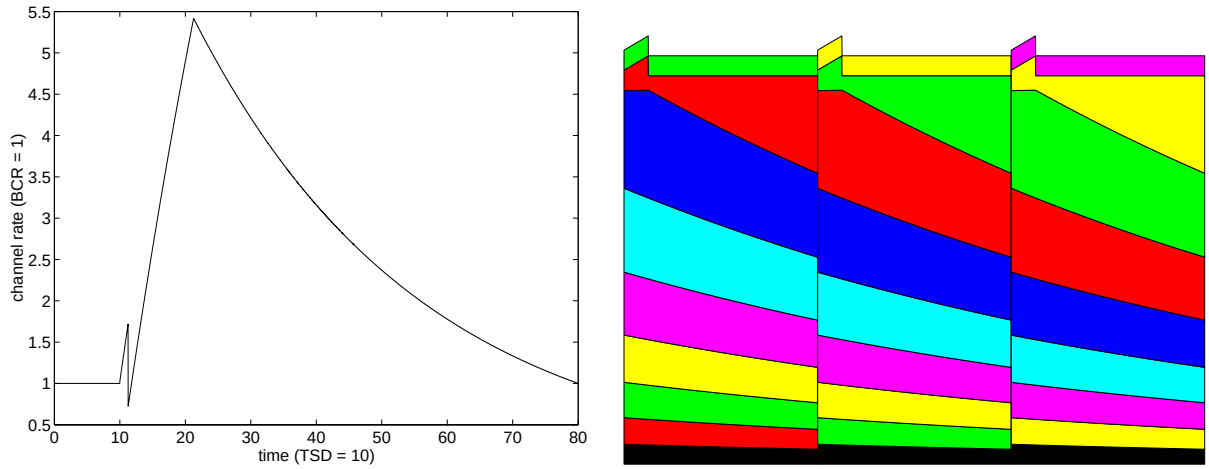
and yield

$$s_k \;=\; \frac{1}{\mathtt{BCR}}\,(k - I_0 - I_1 - I_2 - I_3) + (\mathtt{N}+1) * \mathtt{TSD} - t_0, \qquad k = K_3,\, K_3+1,\, \ldots,\, K - 1. \tag{13}$$

As a sanity check for these calculations, let

$$I_4 \;=\; \int_0^{t_0-\mathtt{TSD}} R(t)\,dt \;=\; \mathtt{BCR}(t_0 - \mathtt{TSD})$$

and verify $I_0 + I_1 + I_2 + I_3 + I_4 = \mathtt{TSD} * \mathtt{SR}$.



(a) (b)

Figure 6: Waves incorrectly designed with (4). (a) The rate of the wave channel does not stay above BCR. (b) The cumulative rate is not constant.

2.2 The secondary case

When (3) yields $t_0 > 2 * \mathtt{TSD}$, the crest of the wave does not occur until the third active time slot. (We will abandon this methodology for very low values of SR so that we do not have to consider $t_0 > 3 * \mathtt{TSD}$.) An example of a ratio SR/BCR that yields $t_0 > 2 * \mathtt{TSD}$ is 20.5. The wave shape given by (4) and the resulting cumulative rate are shown in Fig. 6. Clearly the calculations in Section 2.1 do not apply for this value of SR/BCR.

At the crest of the wave, there are $\mathtt{N}-3$ lower waves and the base channel below, plus two waves at rate BCR above. Thus,

$$\mathtt{MWCR}\left(1 + \mathtt{P} + \mathtt{P}^2 + \cdots + \mathtt{P}^{\mathtt{N}-2}\right) \;=\; \mathtt{SR} - 2 * \mathtt{BCR},$$

or

$$\mathtt{MWCR} \;=\; \frac{1-\mathtt{P}}{1-\mathtt{P}^{\mathtt{N}-1}}(\mathtt{SR} - 2 * \mathtt{BCR}). \tag{14}$$

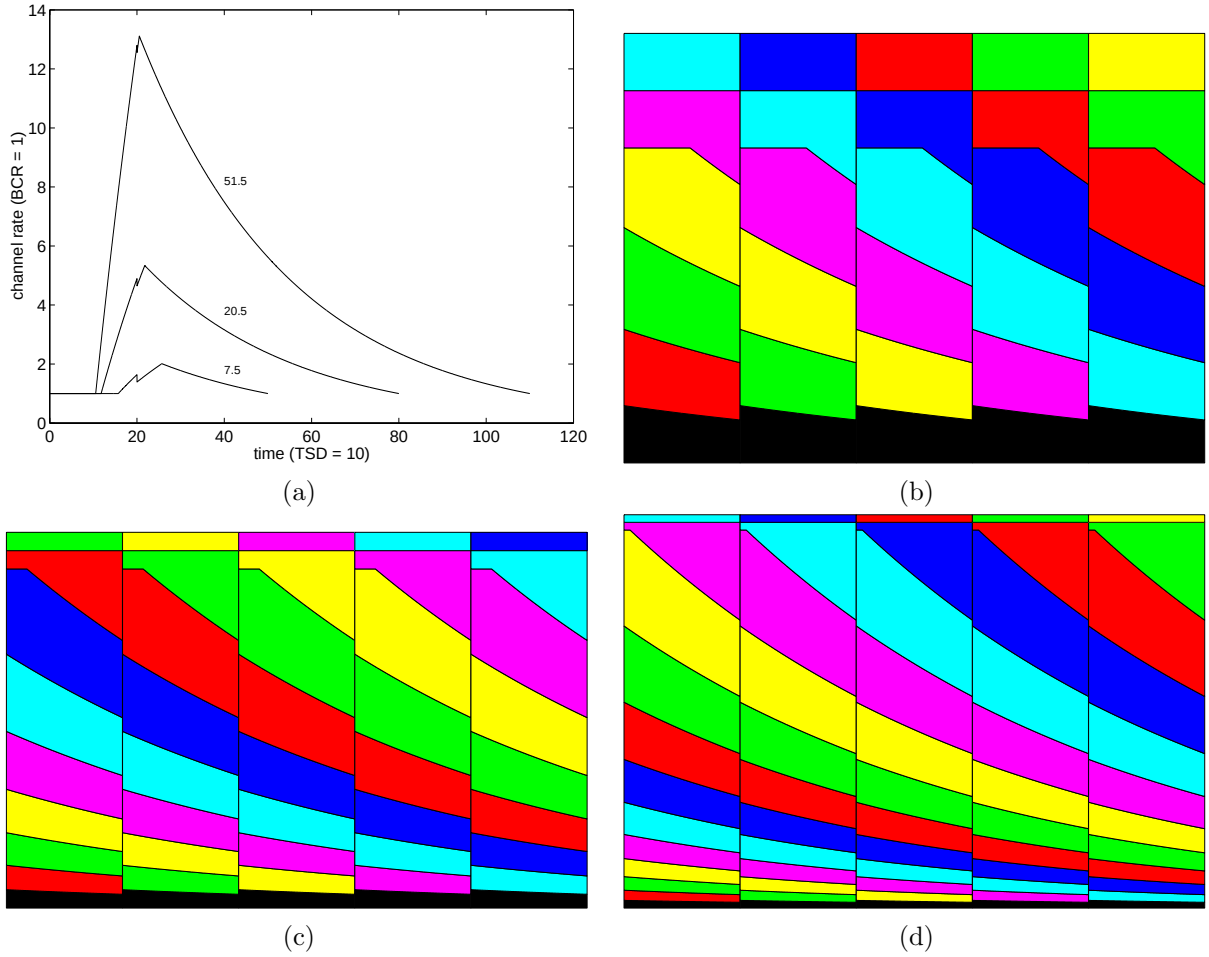
The computation of t_0, which again means the time after which the wave rate is simply a decaying exponential function, uses this new value of MWCR:

$$t_0 \;=\; \mathtt{TSD}\left(\mathtt{N} - \log_{1/\mathtt{P}} \frac{\mathtt{MWCR}}{\mathtt{BCR}}\right). \tag{15}$$

The analogue to (4) is

$$R(t) \;=\; \begin{cases} \mathtt{BCR}, & 0 \le t < t_0 - \mathtt{TSD}; \\ \mathtt{SR} - \mathtt{BCR} - \mathtt{BCR} * \dfrac{(1/\mathtt{P})^{\mathtt{N}-1} - 1}{(1/\mathtt{P}) - 1} * \mathtt{P}^{t/\mathtt{TSD}-1}, & t_0 - \mathtt{TSD} \le t < 2 * \mathtt{TSD}; \\ \mathtt{SR} - 2 * \mathtt{BCR} - \mathtt{BCR} * \dfrac{(1/\mathtt{P})^{\mathtt{N}-2} - 1}{(1/\mathtt{P}) - 1} * \mathtt{P}^{t/\mathtt{TSD}-2}, & 2 * \mathtt{TSD} \le t < t_0; \\ \mathtt{BCR} * (1/\mathtt{P})^{\mathtt{N}-t/\mathtt{TSD}}, & t_0 \le t \le \mathtt{N} * \mathtt{TSD}. \end{cases} \tag{16}$$

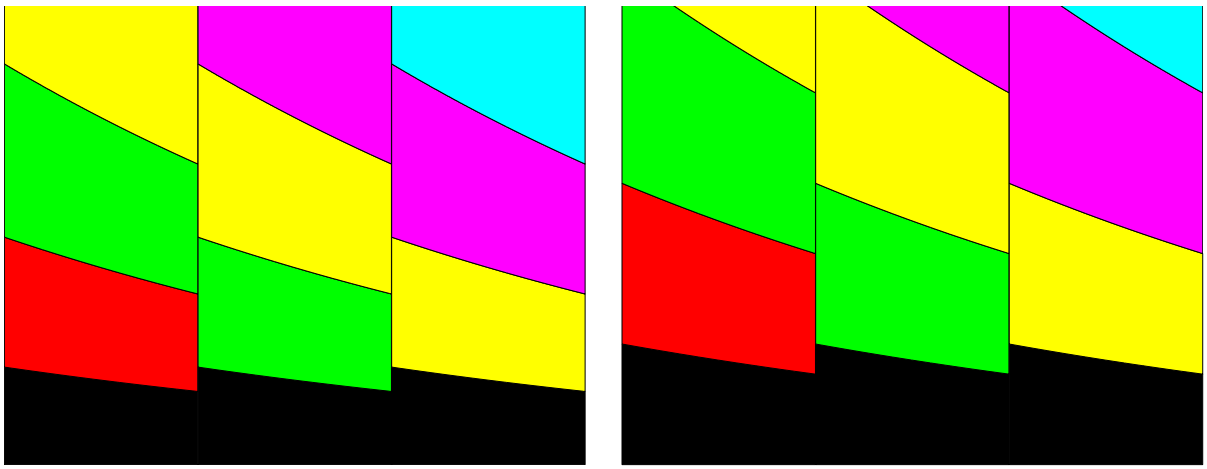
Examples of waves designed with (16) and how they stack up are given in Fig. 7. The various integrations of (16) for scheduling packets are omitted. However, all the corresponding equations can be found in the Matlab code in Appendix B.



(a) (b)

(c) (d)

Figure 7: Waves designed with (16). (a) Waves labeled by their SR values. (b) Stack for SR = 7.5. (c) Stack for SR = 20.5. (d) Stack for SR = 51.5.



(a) $\mathtt{BCR}(1 + 1/\mathtt{P} + \cdots + 1/\mathtt{P}^{\mathtt{N}-1}) < \mathtt{SR}$ (b) $\mathtt{BCR}(1 + 1/\mathtt{P} + \cdots + 1/\mathtt{P}^{\mathtt{N}-1}) > \mathtt{SR}$

Figure 8: Wave designs for very low SR/BCR.

2.3 Very low SRs

Computations (14)–(15) can yield $t_0 > 3 * \mathtt{TSD}$, which means that (16) does not actually give constant server output. However, this occurs only with $\mathtt{SR}/\mathtt{BCR} < 5.084$ (and hence $\mathtt{N} \leq 4$). Total server output rates this low (41.6 kbps with the default base channel rate of 8192 bps) are not very interesting for multiple rate congestion control. Furthermore, it is silly to have a regime where a wave that is active for four or less time slots is constant for at least two slots. Thus I propose a cutoff of $\mathtt{SR}/\mathtt{BCR} = 5.2$ below which we abandon the constraint that the minimum rate on a wave channel is BCR for the entire duration of each active time slot.

Equation (1) is still valid for determining N. Then, there are is a qualitative difference depending on whether $\mathtt{BCR}(1 + 1/\mathtt{P} + \cdots + 1/\mathtt{P}^{\mathtt{N}-1})$ exceeds SR. The two situations are shown in Fig. 8. As shown in Fig. 8(a), when $\mathtt{BCR}(1 + 1/\mathtt{P} + \cdots + 1/\mathtt{P}^{\mathtt{N}-1})$ does not exceed SR, the server output limit affects the rate only for the first time slot. The wave channel rate is given by

$$R(t) \;=\; \begin{cases} \mathtt{SR} - \mathtt{BCR} * \dfrac{(1/\mathtt{P})^{\mathtt{N}} - 1}{(1/\mathtt{P}) - 1} * \mathtt{P}^{t/\mathtt{TSD}}, & 0 \leq t < \mathtt{TSD}; \\ \mathtt{BCR} * (1/\mathtt{P})^{\mathtt{N}-t/\mathtt{TSD}}, & \mathtt{TSD} \leq t \leq \mathtt{N} * \mathtt{TSD}. \end{cases} \tag{17}$$

If $\mathtt{BCR}(1 + 1/\mathtt{P} + \cdots + 1/\mathtt{P}^{\mathtt{N}-1})$ exceeds SR, as shown in Fig. 8(b), there is a breakpoint in the second time slot as in the generic case studied in Section 2.1. Let

$$\mathtt{MWCR} \;=\; \frac{1 - \mathtt{P}}{1 - \mathtt{P}^{\mathtt{N}}} * \mathtt{SR} \tag{18}$$

and

$$t_0 \;=\; \mathtt{TSD} \left(\mathtt{N} - \log_{1/\mathtt{P}} \frac{\mathtt{MWCR}}{\mathtt{BCR}} \right). \tag{19}$$

Then

$$R(t) \;=\; \begin{cases} 0, & 0 \leq t < t_0 - \mathtt{TSD}; \\ \mathtt{SR} - \mathtt{BCR} * \dfrac{(1/\mathtt{P})^{\mathtt{N}} - 1}{(1/\mathtt{P}) - 1} * \mathtt{P}^{t/\mathtt{TSD}}, & t_0 - \mathtt{TSD} \leq t < \mathtt{TSD}; \\ \mathtt{SR} - \mathtt{BCR} * \dfrac{(1/\mathtt{P})^{\mathtt{N}-1} - 1}{(1/\mathtt{P}) - 1} * \mathtt{P}^{t/\mathtt{TSD}-1}, & \mathtt{TSD} \leq t < t_0; \\ \mathtt{BCR} * (1/\mathtt{P})^{\mathtt{N}-t/\mathtt{TSD}}, & t_0 \leq t \leq \mathtt{N} * \mathtt{TSD}. \end{cases} \tag{20}$$

Again, the specifics of scheduling packets to satisfy (17) or (20) are given in the Matlab code in Appendix B.

3 Suggested Server Implementation

Evaluating (6), (9), and (13) and solving (10) and (11) gives values over the reals as the times at which to transmit packets. Since we intend to have constant rate output and packet transmissions could turn out to be bursty for reasons unrelated to WEBRC, these times should be used only to determine an ordering of packets for one period of TSD seconds and then discarded. This ordering is determined by representing each transmission time as a quotient and remainder modulo TSD and then sorting the remainders. This is detailed below.

Recall that in each time slot the server sends L packets on the base channel at times $\{b_k\}_{k=0}^{L-1}$ given by (6). Represent these times as (channel,time) pairs $\{(\mathtt{T}, b_k)\}_{k=0}^{L-1}$.

Without loss of generality, suppose the wave channel for which we have made computations is channel $\mathtt{N}-1$. This means that time 0 is the beginning of a time slot numbered 0. The wave channel

packet transmission time s_k (measured from the end of the wave) is converted to a (channel,time) pair as follows. Let $\mathtt{N} * \mathtt{TSD} - s_k = m * \mathtt{TSD} + w_k$ for $m \in \mathbb{Z}$ and $w_k \in [0, \mathtt{TSD})$. Transmission of a packet on wave channel $\mathtt{N}-1$ at time $\mathtt{N} * \mathtt{TSD} - s_k$ implies a packet is sent on wave channel $\mathtt{N} - 1 - m \bmod \mathtt{T}$ at time w_k. Thus we have the mapping of s_k to $(\mathtt{N} - 1 - m \bmod \mathtt{T}, w_k)$.

The b_ks and w_ks are all in the interval $[0, \mathtt{TSD})$. By sorting the (channel,time) pairs by time, we get a sequence of transmission events for time slot 0. The same sequence can be used in time slot $i \neq 0$ if all channels $\mathtt{CN} \neq \mathtt{T}$ are replaced by $\mathtt{CN} + i \bmod \mathtt{T}$. There is no need to retain the times in the (channel,time) pairs after the sorting.

A Solving $A\tau - B(\mathtt{P}^{-\tau/\mathtt{TSD}} - 1) = C$

Although there is no elementary closed-form solution, any number of root finding methods could be used to solve

$$A\tau - B(\mathtt{P}^{-\tau/\mathtt{TSD}} - 1) \;=\; C. \tag{21}$$

To have a formula (rather than a recursion) for τ, one can replace (21) with an approximate equation that has an elementary closed-form solution. For this purpose it is important to note that we are interested in $\tau \in [0, \mathtt{TSD})$. Thus we get reasonable accuracy with Taylor approximations of $\mathtt{P}^{-\tau/\mathtt{TSD}}$ about $\tau = 0$.

A first-order Taylor approximation is

$$\mathtt{P}^{-\tau/\mathtt{TSD}} \;=\; e^{-\tau(\ln \mathtt{P})/\mathtt{TSD}} \approx 1 - \frac{\ln \mathtt{P}}{\mathtt{TSD}}\tau.$$

This turns (21) into a linear equation with approximate solution

$$\tau \approx \frac{C}{A + B\frac{\ln \mathtt{P}}{\mathtt{TSD}}}. \tag{22}$$

A second-order Taylor approximation is

$$\mathtt{P}^{-\tau/\mathtt{TSD}} \;=\; e^{-\tau(\ln \mathtt{P})/\mathtt{TSD}} \approx 1 - \frac{\ln \mathtt{P}}{\mathtt{TSD}}\tau + \frac{1}{2}\left(\frac{\ln \mathtt{P}}{\mathtt{TSD}}\right)^2 \tau^2,$$

which yields

$$\underbrace{-\frac{B}{2}\left(\frac{\ln \mathtt{P}}{\mathtt{TSD}}\right)^2}_{a}\tau^2 + \underbrace{\left(A + B\frac{\ln \mathtt{P}}{\mathtt{TSD}}\right)}_{b}\tau - \underbrace{C}_{-c} \;=\; 0.$$

This is a standard quadratic equation; the desired solution is

$$\tau \;=\; \frac{-b + \sqrt{b^2 - 4ac}}{2a}.$$

The process of increasing the degree of the Taylor expansion can be continued to degree four; beyond the quadratic case there is no closed-form solution to the approximate polynomial equation that is obtained.

Since $\tau/\mathtt{TSD}$ is not always small enough for fast convergence of the Taylor series about zero, a better way to improve accuracy is to use Taylor approximations about different points. Let τ_0 be

given by (22) and expand $f(\tau) = \mathtt{P}^{-\tau/\mathtt{TSD}}$ about τ_0 to get

$$\begin{aligned}\mathtt{P}^{-\tau/\mathtt{TSD}} &= e^{-\tau(\ln \mathtt{P})/\mathtt{TSD}} \approx f(\tau_0) - \frac{\ln \mathtt{P}}{\mathtt{TSD}} f(\tau_0)(\tau-\tau_0) + \frac{1}{2}\left(\frac{\ln \mathtt{P}}{\mathtt{TSD}}\right)^2 f(\tau_0)(\tau-\tau_0)^2 \\ &= \mathtt{P}^{-\tau_0/\mathtt{TSD}} - \frac{\ln \mathtt{P}}{\mathtt{TSD}}\mathtt{P}^{-\tau_0/\mathtt{TSD}}(\tau-\tau_0) + \frac{1}{2}\left(\frac{\ln \mathtt{P}}{\mathtt{TSD}}\right)^2 \mathtt{P}^{-\tau_0/\mathtt{TSD}}(\tau-\tau_0)^2.\end{aligned}$$

Substituting this approximation in (21) gives

$$\frac{1}{2}B\left(\frac{\ln \mathtt{P}}{\mathtt{TSD}}\right)^2 \mathtt{P}^{-\tau_0/\mathtt{TSD}}(\tau-\tau_0)^2 - \left(A + B\frac{\ln \mathtt{P}}{\mathtt{TSD}}\mathtt{P}^{-\tau_0/\mathtt{TSD}}\right)(\tau-\tau_0) + \left(A\tau_0 + B(\mathtt{P}^{-\tau_0/\mathtt{TSD}} - 1) - C\right) \;=\; 0.$$

Solving this quadratic for $\tau - \tau_0$ gives a very good approximation for τ. This is implemented in the Matlab code in Appendix B.6.

B Matlab code

B.1 packetSchedule()

This is the main function.

```
function [CN,T,t] = packetSchedule( SR, BCR, P, TSD, Q )
% packetSchedule( SR_P, BCR_P, P, TSD, Q )
%  Determine the sequence of CNs for a single time slot.
%  This sequence is used in subsequent time slots by incrementing
%  modulo T all of the CNs that are different from T.
%
%  Defaults:  SR_P  = 100
%             BCR_P = 1
%             P     = 0.75
%             TSD   = 10
%             Q     = 30

if ~exist('Q','var'),
   Q = 30;
   if ~exist('TSD','var'),
      TSD = 10;
      if ~exist('P','var'),
         P = 0.75;
         if ~exist('BCR','var'),
            BCR = 1;
            if ~exist('SR','var'),
               SR = 100;
            end
         end
      end
   end
end

b = baseChannelTimes( BCR, P, TSD );
[w,N] = waveChannelTimes( SR, BCR, P, TSD );
T = N + Q;

times = [b,mod(N*TSD-w,TSD)];
CN = [T*ones(1,length(b)),floor(w/TSD)];
[t,j] = sort(times);
CN = CN(j);
```

B.2 baseChannelTimes()

This computes the transmission times for the base channel. It is subordinate to packetSchedule().

```
function [b,L] = baseChannelTimes( BCR_P, P, TSD )
% baseChannelTimes( BCR_P, P, TSD )
%  Compute a vector of packet transmission times for the base
%  channel for one time slot beginning at time 0.
%
%  Defaults:  BCR_P = 1
%             P     = 0.75
%             TSD   = 10

if ~exist('TSD','var'),
   TSD = 10;
   if ~exist('P','var'),
      P = 0.75;
      if ~exist('BCR_P','var'),
         BCR_P = 1;
      end
   end
end

L = ceil( -BCR_P*TSD*(1-P)/log(P) );
b = TSD * log( 1 + log(P)/BCR_P/TSD * (0:L-1) ) / log(P);
```

B.3 waveChannelTimes()

This computes the transmission times for a single wave. It is subordinate to packetSchedule().

```
function [w,N] = waveChannelTimes( SR, BCR, P, TSD, makePlot )
% waveChannelTimes( SR_P, BCR_P, P, TSD )
%  Compute a vector of packet transmission times for a single
%  wave that starts at time 0.
%
%  The result is accurate for any SR_P >= BCR_P.  However, for
%  certain values of SR_P/BCR_P the actual heavy lifting is done
%  by waveChannelTimes_secondary() or waveChannelTimes_low().
%
%  Defaults:  SR_P  = 100
%             BCR_P = 1
%             P     = 0.75
%             TSD   = 10

if ~exist('makePlot','var')
   makePlot = 0;
   if ~exist('TSD','var'),
      TSD = 10;
      if ~exist('P','var'),
         P = 0.75;
         if ~exist('BCR','var'),
            BCR = 1;
            if ~exist('SR','var'),
               SR = 100;
            end
         end
      end
   end
end
```

```
if (SR/BCR < 5.2),
   [w,N] = waveChannelTimesLowRate( SR, BCR, P, TSD, makePlot );
else
   N = ceil( log( 1 + (1/P)*(1/P-1)*(SR/BCR) )/log(1/P) ) - 1;
   MWCR = (1-P)/(1-P^N)*(SR - BCR);
   t0 = TSD*(N - log(MWCR/BCR)/log(1/P));
   if (t0/TSD > 2),
      [w,N] = waveChannelTimesSecondary( SR, BCR, P, TSD, makePlot );
   else
      kappa = ( (1/P)^(N-1) - 1 ) / ( (1/P) - 1 );
      I0 = -BCR*TSD*(1-P)/log(P);
      I1 = -BCR*TSD/log(P) * ((1/P)^(N-t0/TSD) - 1);
      I2 = (SR-BCR)*(t0-TSD) + BCR*TSD/log(P) * kappa * (1-P^(t0/TSD-1));
      I3 = SR*(2*TSD-t0) + BCR*TSD/log(P) * (P + kappa) * ((1/P)^(2-t0/TSD)-1);
      I4 = BCR*(t0-TSD);
      L = ceil( I0 );

      % part 4/4 of wave:
      K1 = ceil( I0+I1 );
      s1 = TSD * log( 1 - log(P)/BCR/TSD * ((L:(K1-1)) - I0) ) / log(1/P);

      % part 3/4 of wave:
      K2 = ceil( I0+I1+I2 );
      A = SR-BCR;
      B = -BCR * kappa * TSD/log(P) * P^(t0/TSD-1);
      tau = solveIntegerCrossings( A, B, (K1:(K2-1)) - I0 - I1, P, TSD );
      s2 = (N*TSD-t0)+tau;

      % part 2/4 of wave:
      K3 = ceil( I0+I1+I2+I3 );
      A = SR;
      B = -BCR * (P + kappa) * TSD/log(P);
      tau = solveIntegerCrossings( A, B, (K2:(K3-1)) - I0 - I1 - I2, P, TSD );
      s3 = (N-1)*TSD+tau;

      % part 1/4 of wave:
      K4 = round(TSD*SR);
      tau = ((K3:(K4-1)) - I0 - I1 - I2 - I3)/BCR;
      s4 = (N+1)*TSD-t0+tau;

      w = [s1, s2, s3, s4];
      if makePlot,
         figure
         plot( (L+1):K1, s1, 'b.' );
         hold on
         plot( (K1+1):K2, s2, 'r.' );
         plot( (K2+1):K3, s3, 'g.' );
         plot( (K3+1):K4, s4, 'm.' );
         title(['Primary case: SR = ',num2str(SR),', BCR = ',num2str(BCR)])
      end
   end
end
```

B.4 waveChannelTimesSecondary()

This handles the $t_0/\texttt{TSD} > 2$ special case for waveChannelTimes().

```
function [w,N] = waveChannelTimesSecondary( SR, BCR, P, TSD, makePlot )
% waveChannelTimesSecondary( SR_P, BCR_P, P, TSD )
%  Compute a vector of packet transmission times for a single
%  wave that starts at time 0.
%
%  This function is for the secondary case in which SR_P/BCR_P >= 5.2
%  and t0/TSD > 2.
%
%  Defaults:  SR_P  = 51.5
%             BCR_P = 1
%             P     = 0.75
%             TSD   = 10

if ~exist('makePlot','var'),
   makePlot = 0;
   if ~exist('TSD','var'),
      TSD = 10;
      if ~exist('P','var'),
         P = 0.75;
         if ~exist('BCR','var'),
            BCR = 1;
            if ~exist('SR','var'),
               SR = 51.5;
            end
         end
      end
   end
end

N = ceil( log( 1 + (1/P)*(1/P-1)*(SR/BCR) )/log(1/P) ) - 1;
MWCR = (1-P)/(1-P^(N-1))*(SR - 2*BCR);
t0 = TSD*(N - log(MWCR/BCR)/log(1/P));

kappa = ( (1/P)^(N-1) - 1 ) / ( (1/P) - 1 );
lambda = ( (1/P)^(N-2) - 1 ) / ( (1/P) - 1 );
I0 = -BCR*TSD*(1-P)/log(P);
I1 = -BCR*TSD/log(P) * ((1/P)^(N-t0/TSD) - 1);
I2 = (SR-2*BCR)*(t0-2*TSD) - BCR*TSD/log(P) * lambda * (P^(t0/TSD-2) - 1);
I3 = (SR-BCR)*(3*TSD-t0) - BCR*TSD/log(P) * kappa * P * (1 - P^(t0/TSD-3));
I4 = BCR*(t0-TSD);
L = ceil(I0);

% part 4/4 of wave:
K1 = ceil( I0+I1 );
s1 = TSD * log( 1 - log(P)/BCR/TSD * ((L:(K1-1)) - I0) ) / log(1/P);

% part 3/4 of wave:
K2 = ceil( I0+I1+I2 );
A = SR-2*BCR;
B = -BCR * lambda * TSD/log(P) * P^(t0/TSD-2);
tau = solveIntegerCrossings( A, B, (K1:(K2-1)) - I0 - I1, P, TSD );
s2 = (N*TSD-t0)+tau;

% part 2/4 of wave:
K3 = ceil( I0+I1+I2+I3 );
```

```
A = (SR-BCR);
B = -BCR * kappa * TSD/log(P) * P;
tau = solveIntegerCrossings( A, B, (K2:(K3-1)) - I0 - I1 - I2, P, TSD );
s3 = (N-2)*TSD+tau;

% part 1/4 of wave:
K4 = round( TSD*SR );
tau = ((K3:(K4-1)) - I0 - I1 - I2 - I3)/BCR;
s4 = (N+1)*TSD-t0+tau;

w = [s1, s2, s3, s4];
if makePlot,
   figure
   plot( (L+1):K1, s1, 'b.' );
   hold on
   plot( (K1+1):K2, s2, 'r.' );
   plot( (K2+1):K3, s3, 'g.' );
   plot( (K3+1):K4, s4, 'm.' );
   title(['Secondary case: SR = ',num2str(SR),', BCR = ',num2str(BCR)])
end
```

B.5 waveChannelTimesLowRate()

This handles the SR/BCR < 5.2 special case for waveChannelTimes().

```
function [w,N] = waveChannelTimesLowRate( SR, BCR, P, TSD, makePlot )
% waveChannelTimesLowRate( SR_P, BCR_P, P, TSD )
%  Compute a vector of packet transmission times for a single
%  wave that starts at time 0.
%
%  This function is for the low rate case in which SR_P/BCR_P < 5.2.
%
%  Defaults:  SR_P  = 4.5
%             BCR_P = 1
%             P     = 0.75
%             TSD   = 10

if ~exist('makePlot','var'),
   makePlot = 0;
   if ~exist('TSD','var'),
      TSD = 10;
      if ~exist('P','var'),
         P = 0.75;
         if ~exist('BCR','var'),
            BCR = 1;
            if ~exist('SR','var'),
               SR = 4.5;
            end
         end
      end
   end
end

N = ceil( log( 1 + (1/P)*(1/P-1)*(SR/BCR) )/log(1/P) ) - 1;
alpha = ( (1/P)^N - 1 ) / ( (1/P) - 1 );
I0 = -BCR*TSD*(1-P)/log(P);
L = ceil(I0);
```

```
if alpha < SR/BCR,
   I1 = -BCR*TSD/log(P) * ((1/P)^(N-1) - 1);
   I2 = SR*TSD - BCR*TSD/log(P) * alpha * P * (1 - (1/P));

   % part 2/2 of wave:
   K1 = ceil( I0+I1 );
   s1 = TSD * log( 1 - log(P)/BCR/TSD * ((L:(K1-1)) - I0) ) / log(1/P);

   % part 1/2 of wave:
   K2 = round( TSD*SR );
   A = SR;
   B = -BCR * alpha * TSD/log(P) * P;
   tau = solveIntegerCrossings( A, B, (K1:(K2-1)) - I0 - I1, P, TSD );
   s2 = (N-1)*TSD+tau;

   w = [s1, s2];
   if makePlot,
      figure
      plot( (L+1):K1, s1, 'b.' );
      hold on
      plot( (K1+1):K2, s2, 'r.' );
      title(['Case in Fig. 7(a): SR = ',num2str(SR),', BCR = ',num2str(BCR)])
   end
else
   MWCR = (1-P)/(1-P^N)*SR;
   t0 = TSD*(N - log(MWCR/BCR)/log(1/P));
   kappa = ( (1/P)^(N-1) - 1 ) / ( (1/P) - 1 );
   I1 = -BCR*TSD/log(P) * ((1/P)^(N-t0/TSD) - 1);
   I2 = SR*(t0-TSD) - BCR*TSD/log(P) * kappa * (P^(t0/TSD-1) - 1);
   I3 = SR*(2*TSD-t0) - BCR*TSD/log(P) * alpha * P * (1 - P^(t0/TSD-2));

   % part 3/3 of wave:
   K1 = ceil( I0+I1 );
   s1 = TSD * log( 1 - log(P)/BCR/TSD * ((L:(K1-1)) - I0) ) / log(1/P);

   % part 2/3 of wave:
   K2 = ceil( I0+I1+I2 );
   A = SR;
   B = -BCR * kappa * TSD/log(P) * P^(t0/TSD-1);
   tau = solveIntegerCrossings( A, B, (K1:(K2-1)) - I0 - I1, P, TSD );
   s2 = (N*TSD-t0)+tau;

   % part 1/3 of wave:
   K3 = round( TSD*SR );
   A = SR;
   B = -BCR * alpha * TSD/log(P) * P;
   tau = solveIntegerCrossings( A, B, (K2:(K3-1)) - I0 - I1 - I2, P, TSD );
   s3 = (N-1)*TSD+tau;

   w = [s1, s2, s3];
   if makePlot,
      figure
      plot( (L+1):K1, s1, 'b.' );
      hold on
      plot( (K1+1):K2, s2, 'r.' );
      plot( (K2+1):K3, s3, 'g.' );
      title(['Case in Fig. 7(b): SR = ',num2str(SR),', BCR = ',num2str(BCR)])
   end
```

```
end
```

B.6 solveIntegerCrossings()

This makes the calculation described in Appendix A. It is used by waveChannelTimes(), waveChannelTimesSecondary(), and waveChannelTimesLowRate().

```
function tau = solveIntegerCrossings( A, B, C, P, TSD )
%tau = solveIntegerCrossings( A, B, C, P, TSD )
%
%  Find approximate solution to A*tau + B*(1-P^(-tau/TSD)) = C.
%
%  The normal usage is to pass a vector C of integer-spaced values,
%  in which case the equation is solved for each component of C.
%
%  Implemented with a quadratic Taylor expansion about tau0, where tau0
%  is the approximate solution obtained with a linear Taylor expansion.

if ~exist('TSD','var'),
   TSD = 10;
   if ~exist('P','var'),
      P = 0.75;
   end
end

tau0 = C/(A + B*log(P)/TSD);
ftau0 = P.^(-tau0/TSD);

a = 0.5*B*(log(P)/TSD)^2 * ftau0;
b = -(A + B*log(P)/TSD * ftau0);
c = C + B*(ftau0 - 1) - A*tau0;
zeta = (-b-sqrt(b.^2-4*a.*c) )./(2*a);
tau = tau0 + zeta;
```

Acknowledgment

In the original version of this document, transmission times for the base and wave channels were computed independently. Armin Haken demonstrated the possibility of off-by-one errors in the number of wave channel packets produced. This was traced to the difference between I_0 and L and inspired the more elegant, unified treatment given here.

References

[1] M. Luby, V. K Goyal, and S. Skaria. Wave and equation based rate control: A massively scalable receiver driven congestion control protocol. IETF Reliable Multicast Transport Working Group Internet-Draft, October 2001. Document `draft-ietf-rmt-bb-webrc-00.txt` superceded by [2].

[2] M. Luby and V. K Goyal. Wave and equation based rate control building block. IETF Reliable Multicast Transport Working Group Internet-Draft, March 2002. Available online at `http://www.ietf.org/internet-drafts/draft-ietf-rmt-bb-webrc-01.txt`. Work in progress. Expires Sept. 2002.

[3] M. Luby, V. K Goyal, S. Skaria, and G. B. Horn. Wave and equation based rate control using multicast round trip time. In *Proc. ACM SIGCOMM*, Pittsburgh, PA, August 2002. To appear.