**Internet Engineering Task Force** **RMT WG**
**INTERNET-DRAFT** **M. Luby/Digital Fountain**
**draft-ietf-rmt-bb-webrc-01.ps** **V. Goyal/Digital Fountain**
**1 March 2002**
**Expires: September 2002**

# Wave and Equation Based Rate Control building block

### Status of this Document

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as a "work in progress".

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

To view the list Internet-Draft Shadow Directories, see http://www.ietf.org/shadow.html.

This document is a product of the IETF RMT WG. Comments should be addressed to the authors, or the WG's mailing list at rmt@lbl.gov.

### Abstract

Wave and Equation Based Rate Control (WEBRC) provides rate and congestion control for data delivery. WEBRC is specifically designed to support protocols using IP multicast, but could also be used to provide support to protocols that use unicast. WEBRC provides multiple rate congestion controlled delivery to receivers, i.e., different receivers joined to the same session may be receiving packets at different rates depending on their individual bandwidth connection to the sender and on competing traffic along that connection. WEBRC requires no feedback from receivers to the sender, i.e., it is a completely receiver-driven congestion control protocol. Thus, WEBRC is designed to scale to potentially massive numbers of receivers attached to a session from a single sender. Furthermore, because each individual receiver adjusts to the available bandwidth between the sender and that receiver, there is the potential to deliver data to each individual receiver at the fastest possible rate for that receiver, even in a highly heterogeneous network architecture, using a single sender.

## Table of Contents

1. Introduction . . . . . 3
2. Rationale . . . . . 4
3. Functionality . . . . . 4
3.1. Sender Operation . . . . . 6
3.1.1. Sender inputs and initialization . . . . . 6
3.1.2. Sending packets to the session . . . . . 7
3.2. Receiver Operation . . . . . 8
3.2.1. Receiver inputs and initialization . . . . . 8
3.2.2. Receiver measurements and calculations . . . . . 9
3.2.2.1. Average loss probability . . . . . 9
3.2.2.2. Average round-trip time . . . . . 9
3.2.2.3. Rate Equation . . . . . 10
3.2.2.4. Epochs . . . . . 10
3.2.2.5. Average reception rate . . . . . 10
3.2.2.6. Slow start . . . . . 11
3.2.2.7. Target rate . . . . . 11
3.2.3. Receiver events . . . . . 11
3.2.3.1. Epoch change . . . . . 12
3.2.3.2. Time slot change . . . . . 12
3.2.3.3. Join a wave channel . . . . . 12
3.2.3.4. Loss event . . . . . 12
3.2.3.5. Exceptional timeouts . . . . . 12
4. Applicability Statement . . . . . 13
4.1. Environmental Requirements and Considerations . . . . . 13
5. Packet Header Fields . . . . . 14
5.1. Short Format Congestion Control Information . . . . . 15
5.2. Long Format Congestion Control Information . . . . . 15
6. Requirements From Other Building Blocks . . . . . 16
7. Security Considerations . . . . . 17
8. IANA Considerations . . . . . 17
9. Intellectual Property Issues . . . . . 17
10. References . . . . . 17
11. Authors' Addresses . . . . . 19
12. Full Copyright Statement . . . . . 20

# 1. Introduction

This document specifies Wave and Equation Based Rate Control (WEBRC). WEBRC is a congestion control building block that is designed to be massively scalable when used with the IP multicast network service. WEBRC is also suitable as the basis for unicast congestion control, but this is outside the scope of this document. WEBRC is designed to compete fairly with TCP and similar congestion-controlled sessions. WEBRC can be used as a congestion control protocol for any type of data delivery, including reliable content delivery and streaming delivery.

This document describes a building block as defined in RFC3048 [17] that conforms to RFC2357 [14].

WEBRC is a receiver-driven congestion control protocol in the spirit of [16] and [3]. This means that all measurements and decisions to raise or lower the reception rate are made by each individual receiver, and these decisions are acted upon by sending join and leave messages for channels to the network. A receiver using WEBRC adjusts its reception rate without regard for other concerns such as reliability. This is different from TCP, where the congestion control protocol and the reliability protocol are intricately interwoven with one another.

WEBRC takes the same basic equation-based approach as TFRC [5]. In particular, each WEBRC receiver measures parameters that are plugged into a TCP-like equation to compute the receiver target reception rate, and adjusts its reception rate up and down to closely approximate the target reception rate. The sender sends packets to multiple channels called wave channels. Each wave channel follows the same pattern of packet rate transmission spread out over equally spaced intervals of time. The pattern of each wave is that it starts at a high rate that decreases gradually and continually over a long interval of time. (Picture an infinite sequence of waves.) The receiver increases its reception rate by joining the next wave channel earlier in the descent of the wave than it joined the previous wave channel, and the receiver decreases its reception rate by joining the next wave channel later in the descent of the wave than it joined the previous wave channel.

WEBRC is designed for applications that use a fixed packet size, and vary their packet reception rate in response to congestion. WEBRC is designed to be reasonably fair when competing for bandwidth with TCP flows, where a flow is ‘‘reasonably fair’’ if its reception rate is generally within a factor of two of the reception rate of a TCP flow under the same conditions. However WEBRC has a much lower variation of throughput over time compared to TCP, which makes it more suitable for applications such as telephony or streaming media where a relatively smooth rate is of importance. Furthermore, WEBRC is designed to be massively scalable in the sense that the sender is insensitive to the number of receivers joined to a multicast session. The penalty of having smoother throughput than TCP while competing fairly for bandwidth is that WEBRC responds slower than TCP to changes in available bandwidth.

The receiver measures and performs the calculation of congestion control parameters (e.g., the average loss probability) and makes decisions on how to increase or decrease its rate based on these parameters. The receiver-based approach is well suited to an application where the sender is handling many concurrent connections and therefore WEBRC is suitable as a building block for multicast congestion control.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119.

## 2. Rationale

WEBRC provides congestion control for massively scalable protocols using the IP multicast network service. The congestion control that WEBRC provides is common to a variety of applications, including reliable content delivery and streaming applications.

WEBRC is designed to provide congestion control for all packets that are sent to a session. A session comprises multiple channels originating at a single sender that are used for some period of time to carry packets pertaining to the transmission of one or more objects that can be of interest to receivers. The logic behind defining a session as originating from a single sender is that this is the right granularity to regulate packet traffic via congestion control. The rationale for providing congestion control that uses multiple channels within the same session is that this allows the data on the channels to be layered, and a receiver joins and leaves channels in a layered order during its participation in the session. There are advantages to layered data for streaming, where more data can be sent to the lower layers and incrementally important data to higher layers. For reliable content delivery, as described in [12] an application can send in packets encoded data generated from an object in such a way that the arrival of enough packets by a receiver is sufficient to reliably reconstruct original object. A primary advantage of WEBRC is that each receiver controls it reception rate independent of other receivers. Thus, for example, a receiver with a slow connection to the sender does not slow down the receivers with faster connections.

There are coding techniques that provide massively scalable reliability and asynchronous delivery which are compatible with WEBRC. When combined the result is a massively scalable, reliable, asynchronous content delivery protocol that is network friendly. WEBRC also provides congestion control that suitable for streaming applications.

WEBRC avoids using techniques that are not massively scalable. For example, WEBRC does not provide any mechanisms for sending information from receivers to senders, although this does not rule out protocols that both use WEBRC and do require sending information from receivers to senders.

WEBRC provides congestion control that can be tuned for different applications that may have differing application requirements. For example, a content delivery protocol may aggressively strive to use all available bandwidth between receivers and the sender, and thus it must drastically back off its rate when there is competing traffic. On the other hand, a streaming delivery protocol may strive to maintain a constant rate instead of trying to use all available bandwidth, and thus it may not back off its rate as fast when there is competing traffic.

WEBRC does not provide any support beyond congestion control, and thus it must be combined with other building blocks to provide a complete protocol instantiation. For example, WEBRC does not provide any means that can be used to identify which session each received packet belongs to. As another example, WEBRC does not provide support for identifying which object each packet is carrying information about.

## 3. Functionality

A WEBRC session comprises a logically related set of channels originating from a single sender that are used for some period of time to carry data packets with a header carrying WEBRC Congestion Control Information. At the network layer, a channel can be uniquely identified by a (sender IP address, multicast group address) pair.

For each WEBRC session, the channels within the session are mapped uniquely to consecutive channel numbers. In each packet sent to a channel, the channel number that corresponds to the channel is carried in the WEBRC Congestion Control Information. A WEBRC receiver uses the channel number to determine which channel within a session a packet is received from. When packets are received, they are first checked to see that they belong to the appropriate session before WEBRC is applied. For example, if LCT [11] is being used with the session, then the sender IP address together with the Transport Session Identifier supported by LCT would be used to determine which session a received packet belongs to. The particular details of how this filtering is performed it outside the scope of this document. In the remainder of this document, channels will be referred to by their channel number in the scope of the session.

At the sender, time is partitioned into time slots, each of duration TSD seconds. There are a fixed number T of time slot indices associated with a session. As time progresses, the current time slot index increases by one modulo T each TSD seconds. The current time slot index CTSI is carried in the WEBRC Congestion Control Information. This allows receivers to perform very coarse grained synchronization within a session.

WEBRC congestion control is achieved by having the sender send packets associated with a given session to several different channels. Individual receivers dynamically join and leave these channels according to the network congestion as experienced by the receiver. These congestion control adjustments are performed at each receiver independently of all other receivers, and in the case of multicast without any impact on the sender. A packet sequence number is carried in the WEBRC Congestion Control Information. The packet sequence numbers are consecutively numbered per channel and are used by receivers to measure packet loss.

The channels associated with a session consist of one base channel and T wave channels. The packet rate for each channel varies over time. For the base channel, packets are sent to the channel at a low rate BCR_P at the beginning of a time slot and this rate gradually decreases to P*BCR_P at the end of the time slot, where $P < 1$ is a constant defined later. This pattern for the base channel repeats over each time slot. For each wave channel i, packets are sent to channel i starting at a high rate MWCR_P and decreasing over time by a fixed fraction P per time slot until a rate of BCR_P is reached at the end of time slot i. Then, for a period of time called the quiescent period, no packets are sent to wave channel i, and thereafter the whole cycle repeats itself, where the duration of the cycle is T*TSD seconds. Thus, the wave channels are going through the same cyclic pattern of packet rate transmission spaced out evenly by TSD seconds.

Before joining a session, the receivers MUST obtain enough of the session description to start the session. This MUST include the relevant session parameters needed by a receiver to participate in the session and perform WEBRC congestion control. The session description is determined by the sender and is typically communicated to the receivers out of band. How receivers obtain the session description is outside the scope of this document.

When a receiver initiates a session, it first joins the base channel. The packets in the base channel help the receiver orient itself in terms of what the current time slot index is, which in turn allows the receiver to know the relative rates on the wave channels. The receiver remains joined to the base channel for the duration of its participation in the session.

After joining a session the receiver adjusts its rate upwards and downwards by joining wave channels in sequence, from the lowest rate wave channel and moving towards the higher rate wave channels. Since the relative ordering among the channels with respect to their rate depends

on the current time slot index, it is important that the receiver continually monitor the current time slot index contained in received packets. The reception rate at the receiver is determined by how early each wave channel is joined by the receiver: the earlier the receiver joins a channel with respect to when its wave started, the higher the reception rate. When the receiver wants to decrease its rate, it joins the next wave channel at a later time relative to when it joined the previous wave. When the receiver wants to increase its rate, it joins the next wave channel at an earlier time relative to when it joined the previous wave.

Once the receiver is joined to a wave channel, the receiver remains joined to the wave channel until the channel goes quiescent, at which point the receiver MUST leave the channel.

The way the receiver adjusts its reception rate is inspired by TFRC [5]. The receiver at all points in time maintains a target reception rate, and the receiver is allowed to join the next wave channel if after joining its anticipated reception rate would be at most its target reception rate. The target rate is continually updated based on a set of measured parameters. The primary parameters are the average loss probability (measured in much the same way as described in TFRC) and the average round-trip time (measured as described below).

## 3.1. Sender Operation

The sender operation is by design much simpler than the receiver operation.

### 3.1.1. Sender inputs and initialization

The primary input to the sender for the session is MSR_b. MSR_b is the maximum sender transmission rate in bits per second at any point in time in aggregate to all channels. This is also the maximum rate in bits per second that any receiver could receive data from the session at any point in time.

The secondary inputs to the sender are listed below. These are secondary inputs because in general the values for these inputs will be fixed to default values that will not change, or because they are set based on non-WEBRC considerations.

- LENP_B is the length of packets in bytes sent to the session. The value of LENP_B depends on the complete protocol, but in general this should be set to as high a value as possible without exceeding the MTU size for the network that would cause fragmentation.
- BCR_P is the transmission rate on the base channel at the beginning of a time slot in packets per second. The default value for BCR_P is 1.
- TSD is the time slot duration measured in seconds. The default value for TSD is 10.
- QD is the minimum quiescent period duration measured in seconds. The default value for QD is 300.
- P is the multiplicative drop in every channel rate over each time slot. The default value for P is 0.75.

From these inputs the following fixed sender parameters can be derived as follows.

- MSR_P = MSR_b/(8*LENP_B) is the maximum sender transmission rate in packets per second.

- o BCR_b = 8*LENP_B*BCR_P is the rate of the base channel at the beginning of a time slot in bits per second.
- o N = floor{log_{1/P}((1+(1-P)/P^2)*(MSR_P/BCR_P))}-1 is the number of active time slots for a wave channel. A wave channel is active in any time slot if it is not quiescent. N is also the number of wave channels active in every time slot.
- o Q = ceil(QD/TSD) is the number of quiescent time slots for a wave channel.
- o T = N + Q is the total number of time slots in a cycle. T is also the total number of wave channels.
- o For the base channel CN = T and for the wave channels CN = 0,1,...,T-1. The sender has the description of the channels assigned to the session and the mapping between the channels and the CNs.
- o C = TSD*T is the total duration in seconds of a cycle.

#### 3.1.2. Sending packets to the session

The sender keeps track of the current time slot index CTSI. The value of CTSI is incremented by 1 modulo T each TSD seconds. The value of CTSI is placed into each packet in the format described in Section 5. For each packet sent to the session, the sender also places the channel number CN of the channel into the packets in the format described in Section 5. Recall that CN = T for the base channel and CN = 0,1,...,T-1 for the wave channels.

For each packet sent to the session, the sender calculates and places a packet sequence number PSN into the packet. The value of PSN is scoped by CN, and the value of PSN is consecutively increasing within each channel. Furthermore, for each wave channel, the last packet sent before the channel becomes quiescent must have the maximum possible PSN value. When the short format for Congestion Control Information is used (see Section 5.1), this implies that for any wave channel the PSN values are 65 535, 65 534, 65 533, ..., going backward in time starting at the last packet sent to the channel just before the channel becomes quiescent. Similarly, when the long format for Congestion Control Information is used (see Section 5.2), the PSN for the final packet of any wave is 4 294 967 295. The PSN of the initial packet of a wave thus depends on TSD, P, BCR_P and MSR_P. The format for the PSN within packets is described in Section 5.

Packets are sent to the base channel at a rate of BCR_P packets per second at the beginning of a time slot decreasing at a constant relative rate till the rate reaches P*BCR_P at the end of a time slot. To explain the packet rate for wave channels, suppose for simplicity that MSR_P/BCR_P = 1 + (1/P) + (1/P)^2 + ... + (1/P)^N = ((1/P)^{N+1}-1)*P/(1-P). For all i = 0,1,...,T-1, the rate for wave channel i behaves as follows. At the beginning of time slot i+Q+1 modulo T, packets are sent to wave channel i at the rate of BCR_P*(1/P)^N. The rate of packets sent to wave channel i steadily decreases geometrically by a factor of P during each of the next N time slots until the end of time slot i when the rate reaches BCR_P. During the Q time slots i+1 modulo T, i+2 modulo T, ..., i+Q modulo T, wave channel i is quiescent, i.e., no packets are sent to the channel.

### 3.2. Receiver Operation

The bulk of the complexity in WEBRC is in the receiver operation. For ease of explanation, suppose for the moment that during the reception there is no packet loss and packets are arriving at exactly the rate at which they were sent. The sender transmission rate to the channels is designed so that the receiver reception rate behaves as follows.

Upon entering a session, the receiver immediately joins the base channel. When the receiver wants to increase its rate, it consecutively joins wave channels from the lowest rate channel to the highest rate channel. (Recall that the designations of lowest to highest change at time slot boundaries.) When the receiver wants to maintain its current reception rate and it is already joined to NWC wave channels, if the receiver joins channel i-1+NWC modulo T sometime during time slot i then the receiver joins channel i+NWC modulo T TSD seconds later in time slot i+1. As each wave channel becomes quiescent the receiver leaves the channel.

Suppose the receiver wants to decrease its rate till it is joined to just the base channel. Assume that a receiver is joined to the NWC wave channels i, i+1 modulo T, ..., i+NWC-1 modulo T at the beginning of time slot i. Then, the aggregate packet reception rate of the receiver over the next NWC time slots will behave as follows. At the beginning of time slot i the receiver reception rate is BCR_P*(1 + (1/P) + (1/P)^2 + ... + (1/P)^NWC). Then the receiver reception rate decreases by a factor of P over the duration of each time slot, and at the end of each time slot the reception rate decreases by an additive amount of P*BCR_P. At the end of time slot i+NWC-1 mod T, the receiver reception rate is BCR_P*(1+P), and at the beginning of time slot i+NWC mod T the receiver is joined only to the base channel and its reception rate is BCR_P.

### 3.2.1. Receiver inputs and initialization

Before joining a session the receiver MUST know the mapping between the CNs and the channels. Upon joining the session, it should have the values of LENP_B, BCR_P, TSD, P, N, Q and T. Some of these values may be obtained or measured once the receiver has joined the session. For example, the receiver MAY obtain LENP_B and T from the first packet received from the base channel, and the receiver MAY measure BCR_P once it is joined to the base channel. The values of P, Q and TSD MAY be fixed to default values built into the receiver that do not change from session to session, and the value of N MAY be computed as T-Q.

When a receiver first joins a session, it MUST first join just the base channel and start receiving packets to determine the current time slot index. If during the course of the session the receiver continually loses a high fraction of the packets from the base channel even when the receiver is only joined to the base channel, the receiver SHOULD leave the session.

The receiver MAY also have other individually set parameters that may be used to determine its behavior. Two such parameters are:

- MRR_b is the maximum receiver reception rate in bits per second. This may be used to determine the maximum reception rate this receiver is willing to reach. Thus, the maximum reception rate that the receiver can possibly achieve in the session is the minimum of MSR_b and MRR_b. A recommended value of MRR_b for a receiver is the bandwidth capacity of the last link to the receiver. MRR_P is the maximum receiver reception rate in packets per second, i.e., MRR_P = MRR_b/(8*LENP_B).
- CONNEC is the number of virtual connections that the receiver is making to the sender to receive data from the session. This affects how competitive the receiver will be compared to other receivers (both TCP and WEBRC receivers) in competing for bandwidth over bottleneck links. The standard default value for CONNEC is 1.

### 3.2.2. Receiver measurements and calculations

As outlined in the introduction, the way a receiver adjusts its reception rate is inspired by TFRC [5]. The receiver at all points in time maintains a target reception rate, and the receiver is allowed to join the next wave channel if joining would increase its reception rate to at most its target reception rate. The target rate is continually updated based on a set of measured parameters. The primary parameters are the average loss probability LOSSP and the average round-trip time ARTT.

#### 3.2.2.1. Average loss probability

The average loss probability LOSSP of the receiver is maintained in a manner very similar to that described in TFRC [5]. The calculation of LOSSP as the inverse of a weighted average is exactly as described in TFRC, with the same definition of loss events. One difference is the number of inter-loss periods that are used to compute the average loss event rate:

- In TFRC, it is recommended that the loss probability be averaged over the previous 8 loss events.
- In WEBRC, it is recommended that the loss probability LOSSP be averaged over the previous channel periods that include 8 channel periods where there was at least one loss event. A channel period is defined as the time between when a wave channel is joined till the time the next wave channel is joined. Thus, since a channel period is generally around TSD seconds, the loss probability will generally be averaged over at least 8*TSD seconds.

The other difference is in the weights:

- In TFRC, it is recommended that the weights (1, 1, 1, 1, 0.875, 0.625, 0.375, 0.125) be used for the 8 inter-loss periods (from most recent to most distant past).
- In WEBRC, it is recommended that these same 8 values be used for the loss events that are 1,2,...,8 lossy channel periods into the past. In addition, it is recommended that the weight for each inter-loss period be multiplicatively adjusted by the factor Pˆmax{0,LOSS_NWC-NWC-1}, where NWC is the number of wave channels the receiver is currently subscribed to and LOSS_NWC is the number of wave channels the receiver was subscribed to at the end of the inter-loss period in question.

#### 3.2.2.2. Average round-trip time

The receiver maintains an average round-trip time, ARTT, as the exponentially weighted moving average of RTT measurements. Each time the receiver joins a channel (either the base channel at the beginning or wave channels continually), it makes a measurement of the round-trip time RTT as follows. When the receiver sends the join for the channel it records the current time X and sets a Boolean variable JOINING to true. When the first packet is received from the channel the receiver records the current time Y and resets the value of JOINING to false. If it is the base channel that has been joined, RTT is set to Y - X. Otherwise, when the receiver receives a second packet from the channel it records the current time Z. Then, the value of RTT is set to 3*Y/2 - Z/2 - X. (Note that this value can be negative.)

When the kth RTT measurement is made, the value of ARTT is updated to max{ARRT/2, (1-delta)*ARTT + delta*RTT}, where delta = alpha/(1-(1-alpha)ˆk). The recommended value for alpha is 0.1.

It is RECOMMENDED that the RTT measurement be adjusted appropriately for packets lost between the first and second received packets.

#### 3.2.2.3. Rate Equation

The receiver calculates the reception rate REQN based on the TCP-equation as follows: REQN = CONNEC/(ARTT*sqrt{LOSSP}(0.816 + 7.35*LOSSP*(1+32*LOSSP^2))). This equation comes from TFRC [5].

#### 3.2.2.4. Epochs

The receiver makes decisions on whether or not to join another wave channel at equally spaced units of time called epochs. The duration of an epoch in seconds, EL, is set to be a small fraction of TSD, so that decisions to join a channel can be made at a much finer granularity than TSD. A standard setting is EL = TSD/20. Thus, if TSD = 10, then EL = 0.5.

#### 3.2.2.5. Average reception rate

There are two averaged versions of reception rate maintained by the receiver: TRR_P, the true reception rate, and ARR_P, the anticipated reception rate. These are used for different purposes and thus are calculated quite differently.

The true reception rate TRR_P is used to ensure that the receiver does not increase its reception rate too quickly above its current reception rate. TRR_P is calculated based on the measurement of RR_P, where RR_P is the receiver reception rate in packets per second measured at the beginning of an epoch averaged over the epoch that just ended. TRR_P is initialized to BCR_P. TRR_P is updated to (1-beta)*TRR_P + beta*RR_P at the beginning of each epoch after RR_P is measured for the previous epoch.

The anticipated reception rate ARR_P is the receiver's estimate of the total instantaneous rate of the currently subscribed channels. It is used to compare against the target rate to decide whether or not the receiver can increase its reception rate by joining the next channel. ARR_P is calculated based on a measurement IRR_P and on the number of subscribed wave channels NWC. IRR_P is what the ideal receiver reception rate should have been in packets per second measured at the beginning of the epoch averaged over the previous epoch, i.e., IRR_P includes both received and lost packets. ARR_P is initialized to BCR_P and NWC is initialized to 0. ARR_P, IRR_P and NWC are updated as follows:

- At the beginning of each epoch, IRR_P is measured over the previous epoch and then ARR_P is updated to P^(EL/TSD)*(1-gamma)*ARR_P + gamma*IRR_P.
- When a join is made to a wave channel, NWC is updated to NWC+1 and then ARR_P is multiplicatively increased by the factor ((1/P)^(NWC+1)-1)/((1/P)^NWC-1). (Joins happen at epoch boundaries; this adjustment is in addition to the adjustment above.)
- When a new time slot index is detected, ARR_P is additively increased by (1-P)*BCR_P to account for the change in rate on the base channel. In addition, for each wave channel that has just been detected to be quiescent and thus had a leave message sent, ARR_P is additively decreased by BCR_P and NWC is decremented by 1. (Normally, a change of time slot index indicates that one wave channel has become quiescent.)

Consider for the moment what happens if gamma = 0 and ARR_P is an accurate estimate of the total rate of the subscribed channels. The adjustments to ARR_P upon joining and leaving wave channels, with the passage of epochs, and with the detection of time slot changes will then cause ARR_P to remain an accurate estimate. In practice, gamma MUST be positive; allowing an influence of IRR_P prevents ARR_P from drifting away from being an accurate estimate of the total subscribed rate.

The motivation for separate estimates TRR_P and ARR_P is as follows. TRR_P alone is not appropriate for comparison with the TFRC-inspired target rate because there is a considerable lag before it reflects the rate increase resulting from joining the next wave channel. However, TRR_P is needed because ARR_P alone does not reflect the amount of data getting to the receiver, and a large gap between the subscribed rate and the received rate should not be allowed.

The recommended values of beta and gamma depend on whether the receiver is in slow start mode. Slow start is indicated by TRR_P < SSR_P/zeta^2, where SSR_P is defined in the following subsection and zeta = ((1/P)^(NWC+2)-1)/((1/P)^(NWC+1)-1). It is recommended that beta and gamma equal P/(1+P) in slow start and equal (2/3)*(-c/2 + sqrt(c^2/4 + 3*c/2), where c = 1 - cos(6.283*EL/TSD), otherwise.

#### 3.2.2.6. Slow start

WEBRC uses a slow start mechanism to quickly ramp up its rate at both the beginning of the session and in the middle of a session when the rate drops precipitously. To enact this, the receiver maintains the following parameters:

- SSMINR_P is the minimum allowed slow start threshold rate in packets per second. The recommended value for SSMINR_P is BCR_P*(1+1/P+1/P^2).
- SSR_P is the slow start threshold rate in packets per second. SSR_P is initialized to MRR_P*P^2.

#### 3.2.2.7. Target rate

The target rate TRATE is computed as TRATE = min{max{SSR_P, REQN}, TRR_P*((1/P)^(NWC+3)-1)/((1/P)^(NWC+1)-1), MRR_P}.

### 3.2.3. Receiver events

There are various receiver events, some of which are triggered by the passing of time on the receiver, and others by events such as packet reception, detection of packet loss, reception of a first packet from a channel, and exceptional time-outs.

#### 3.2.3.1. Epoch change

This is an event that is driven by the passage of time on the receiver, which occurs each EL seconds. When this happens, TRR_P and ARR_P are computed as described in Section 3.2.2.5. Immediately after these updates, a decision is made about whether to join a wave channel as described in Section 3.2.3.3.

#### 3.2.3.2. Time slot change

This is an event that is driven by the reception of a packet with a new CTSI value. When a packet with a new CTSI = i is received, a leave is sent for wave channel i-1 modulo T, NWC is updated to NWC-1, and ARR_P is updated to ARR_P - P*BCR_P as described in Section 3.2.2.5.

#### 3.2.3.3. Join a wave channel

At the beginning of each epoch, after updating the values of ARR_P and TRR_P as described in Section 3.2.2.5, the receiver decides whether or not to join a wave channel as follows:

- o If there is a loss event in progress (LOSS_EVENT = true) or there is a join of a channel in progress (JOINING = true), then no join of a channel is attempted.
- o The receiver calculates REQN as described in Section 3.2.2.3.
- o The receiver calculates TRATE as described in Section 3.2.2.7.
- o The receiver checks if TRATE > ARR_P*((1/P)^{NWC+2}-1)/((1/P)^{NWC+1}-1). In this case, subscribing to an additional wave channel is not expected to increase the subscribed rate over the target rate. Suppose CTSI = i and NWC = J. If the inequality is true then the receiver joins the wave channel with CN = i+J modulo T, updates NWC to NWC+1, and then updates ARTT to ARR_P*((1/P)^{NWC+1}-1)/((1/P)^NWC-1) as described in Section 3.2.2.5.

#### 3.2.3.4. Loss event

Each time the receiver detects a lost packet (based on the sequence numbers in the packets scoped by the channel number), the receiver records the start of a new loss event, and sets a Boolean variable LOSS_EVENT to true that will automatically reset to false after ARTT seconds. All subsequent packet loss for a period of ARTT seconds is considered as part of the same loss event. When a start of a loss event is detected, the value of SSR_P is updated to max{SSMINR_P, TRR_P*P^2}.

It is RECOMMENDED that the receiver account for simple misordering of packets without inferring a loss.

#### 3.2.3.5. Exceptional timeouts

These are timeouts when the packet reception behavior is far from what it should be and should trigger a drastic event by the receiver. Exception timeouts include events such as when no packets are received for a long time, there is no change in the current time slot index for a long time, or no first or second packet is received after join to channel for a long time.

## 4. Applicability Statement

WEBRC is intended to be a congestion control scheme that can be used in a complete protocol instantiation that delivers objects and streams (both reliable content delivery and streaming of multimedia information). A WEBRC session comprises a logically related set of one or more channels originating at a single sender that are used for some period of time to carry packets pertaining to the transmission of one or more objects that can be of interest to receivers. WEBRC congestion control is to be performed over the aggregate of all packets sent to the session.

WEBRC is most applicable for delivery of objects or streams of substantial length, i.e., objects or streams that range in length from hundreds of kilobytes to many gigabytes, and whose transfer time is in the order of tens of seconds or more.

## 4.1. Environmental Requirements and Considerations

WEBRC can be used with both multicast and unicast networks. However, the scope of this document is limited to multicast. WEBRC requires connectivity between a sender and receivers, but does not require connectivity from receivers to the sender.

WEBRC inherently works with all types of networks, including LANs, WANs, Intranets, the Internet, asymmetric networks, wireless networks, and satellite networks. Thus, the inherent raw scalability of WEBRC is unlimited. However, in some network environments varying reception rates to receivers may not be advantageous, e.g., when the network is a satellite network with a fixed amount of bandwidth dedicated to a session.

Receivers join and leave channels using the appropriate multicast join and leave messages. For IPv4 multicast, IGMP messages are used by receivers to join and leave channels. For IPv6, MLDv2 messages are used by receivers to join and leave channels. This is the only dependency of WEBRC on the IP version.

WEBRC requires receivers to be able to uniquely identify and demultiplex packets associated with a session in order to effectively perform congestion control over all packets associated with the session. How receivers achieve this is outside the scope of this document.

WEBRC is presumed to be used with an underlying network or transport service that is a ‘‘best effort’’ service that does not guarantee packet reception, packet reception order, and which does not have any support for flow or congestion control. For example, the Any-Source Multicast (ASM) model of IP multicast as defined in RFC1112 [4] is such a best effort network service. While the basic service provided by RFC1112 is largely scalable, providing congestion control or reliability should be done carefully to avoid severe scalability limitations, especially in the presence of heterogeneous sets of receivers.

There are currently two models of multicast delivery, the Any-Source Multicast (ASM) model as defined in RFC1112 [4] and the Source-Specific Multicast (SSM) model as defined in [6]. WEBRC works with both multicast models, but in a slightly different way with somewhat different environmental concerns. When using ASM, a sender S sends packets to a multicast group G, and the WEBRC channel address consists of the pair (S,G), where S is the IP address of the sender and G is a multicast group address. When using SSM, a sender S sends packets to an SSM channel (S,G), and the WEBRC channel address coincides with the SSM channel address.

A sender can locally allocate unique SSM channel addresses, and this makes allocation of channel addresses easy with SSM. To allocate channel addresses using ASM, the sender must uniquely chose the ASM multicast group address across the scope of the group, and this makes allocation of WEBRC channel addresses more difficult with ASM.

WEBRC channels and SSM channels coincide, and thus the receiver will only receive packets sent to the requested WEBRC channel. With ASM, the receiver joins a channel by joining a multicast group G, and all packets sent to G, regardless of the sender, may be received by the

receiver. Thus, SSM has compelling security advantages over ASM for prevention of denial of service attacks. In either case, receivers SHOULD use mechanisms to filter out packets from unwanted sources.

WEBRC assumes that the packet route between the sender and a particular receiver is the same for all channels associated with a session. For SSM this assumption is true because the multicast tree is a shortest path tree from each receiver to the sender and generally this path changes infrequently. For ASM there are some issues that if not properly considered may invalidate this assumption. With ASM, the packet route between the sender and receivers may initially be through the Rendezvous Point (RP) and then switch over to the shortest path to the sender as packets start flowing in a channel. The first issue is that the RP may not be the same for all channels associated with a session, and thus the first packets sent to the channels may follow a route that depends on the RP of the channel. This depends on the RP configuration for the sender. If the sender registers all channels associated with the session with the same RP then the assumption is true, but if the sender registers different channels with different RPs then the assumption may not be true. Thus, it is RECOMMENDED that the sender register all channels associated with a session with the same RP. Another issue is that when the channel switches over from the RP to the sender-based tree then the route to the receivers may vary within a channel. Furthermore, this may cause either the receipt of duplicate packets at receivers or loss of packets depending on the smoothness of the switchover. Thus, it is RECOMMENDED that the RP be placed as close as possible to the sender. The best location for the RP is that it be the first-hop router closest to the sender, in which case the path to the sender and the path to the RP is the same for each receiver and the problems mentioned above are eliminated. The consequences of this assumption not being true are that the receiver reaction to congestion may not be appropriate. Generally, the WEBRC receiver will act conservatively and reduce its reception rate too much if this assumption is not true, but there can be cases where the receivers will act inappropriately.

Some networks are not amenable to the WEBRC congestion control protocol. In particular, for a satellite or wireless network, there may be no mechanism for receivers to effectively reduce their reception rate since there may be a fixed transmission rate allocated to the session.

# 5. Packet Header Fields

Packets sent to a session using WEBRC MUST include Congestion Control Information fields as specified in this section. This document specifies short and long formats for the Congestion Control Information, and it is RECOMMENDED that protocol instantiations use one of these two formats. Other formats for the Congestion Control Information fields MAY be used by protocol instantiations, but all protocol instantiations are REQUIRED to use these fields in a format that is compatible with the interpretations of these fields. Thus, if a protocol does use a different format for the fields in the Congestion Control Information then it MUST specify the lengths and positions of these fields within the packet header.

All integer fields are carried in "big-endian" or "network order" format, that is, most significant byte (octet) first. All constants, unless otherwise specified, are expressed in base ten.

## 5.1. Short Format Congestion Control Information

The short format for the Congestion Control Information is shown in Fig. 1. The total length of the short format is 32-bits.

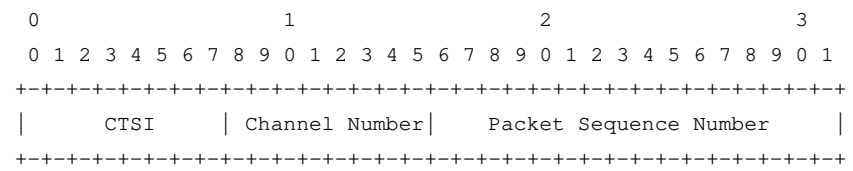


Fig. 1 - Short format for Congestion Control Information

The function of each field in the Congestion Control Information is the following.

Current Time Slot Index (CTSI): 8 bits

CTSI indicates the index of the current time slot. This must be sent in each packet within the session. The Current Time Slot Index increases by one modulo T each TSD seconds at the sender, where T is the number of time slots associated with the session and TSD is the time slot duration. Note that T is also the number of wave channels associated with the session, and thus T MUST be at most 255.

Channel Number (CN): 8 bits

CN is the channel number that this packet belongs to. CN for the base channel is T, and the CNs for the wave channels are 0 through T-1. Thus, T+1 channels in total are used, and thus T MUST be at most 255.

Packet Sequence Number (PSN): 16 bits

The PSN of each packet is scoped by its CN value. The sequence numbers of consecutive packets sent to the base channel are numbered consecutively modulo 2^16. The same sequence of PSNs are used for each wave channel in each cycle. The sequence numbers of consecutive packets sent to a wave channel are numbered consecutively modulo 2^16 within each cycle, ending with the last packet sent to the channel before the channel goes quiescent with PSN = 2^16-1.

## 5.2. Long Format Congestion Control Information

The long format for the Congestion Control Information is shown in Fig. 2. The total length of the long format is 64-bits.

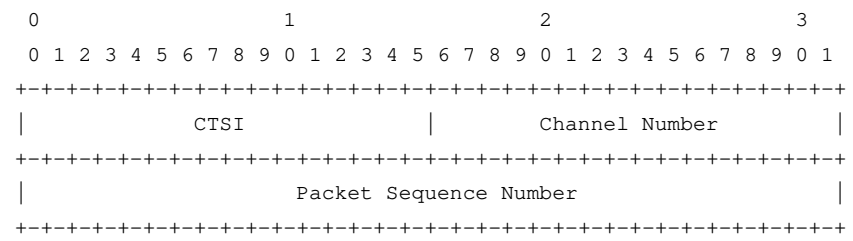


Fig. 1 - Long format for Congestion Control Information

The meaning of each field for the long format is the same as for the short format, the only difference is that each field is twice as long.

Current Time Slot Index (CTSI): 16 bits

CTSI indicates the index of the current time slot. This must be sent in each packet within the session. The Current Time Slot Index increases by one modulo T each TSD seconds at the sender, where T is the number of time slots associated with the session and TSD is the time slot duration. Note that T is also the number of wave channels associated with the session, and thus T MUST be at most 65 535.

Channel Number (CN): 16 bits

CN is the channel number that this packet belongs to. CN for the base channel is T, and the CNs for the wave channels are 0 through T-1. Thus, T+1 channels in total are used, and thus T MUST be at most 65 535.

Packet Sequence Number (PSN): 32 bits

The PSN of each packet is scoped by its CN value. The sequence numbers of consecutive packets sent to the base channel are numbered consecutively modulo 2^32. The same sequence of PSNs are used for each wave channel in each cycle. The sequence numbers of consecutive packets sent to a wave channel are numbered consecutively modulo 2^32 within each cycle, ending with the last packet sent to the channel before the channel goes quiescent with PSN = 2^32-1.

## 6. Requirements From Other Building Blocks

As described in RFC3048, WEBRC is a building block that is intended to be used, in conjunction with other building blocks, to help specify a protocol instantiation.

WEBRC does not provide higher level session support, e.g., how receivers obtain the necessary session description and how the receivers demultiplex received packets based on their session. There is support provided by other building blocks that can be used in conjunction with WEBRC to provide some of this support. For example, LCT [11] can provide some of the higher level in-band session support that may be needed by receivers, and the WEBRC Congestion Control Information (CCI) required in each packet can be carried in the CCI field of the LCT header [11].

WEBRC does not provide any type of reliability, and in particular does not provide support for retransmission of loss packets. Reliability can be added by independent means, such as by the use of FEC codes as described in [12] and specified in the FEC building block [13].

## 7. Security Considerations

WEBRC can be subject to denial-of-service attacks by attackers that try to confuse the congestion control mechanism for receivers by injecting forged packets into the multicast stream. This attack most adversely affects network elements and receivers downstream of the attack, and much less significantly the rest of the network and other receivers. Because of this and because of the potential attacks due to the use of FEC described above, it is RECOMMENDED that some form of packet authentication such as TESLA [15] be used to protect against such attacks and that Reverse Path Forwarding checks be enabled in all network routers and switches along the path from the sender to receivers to limit the possibility of a bad agent injecting forged packets into the multicast tree data path.

A receiver with an incorrect or corrupted implementation of WEBRC may affect health of the network in the path between the sender and the receiver, and may also affect the reception rates of other receivers joined to the session. It is therefore RECOMMENDED that receivers be required to identify themselves as legitimate before they receive the session description needed to join the session.

Another vulnerability of ALC is the potential of receivers obtaining an incorrect session description for the session. The consequences of this could be that legitimate receivers with the wrong session description are unable to correctly receive the session content, or that receivers inadvertently try to receive at a much higher rate than they are capable of, thereby disrupting traffic in portions of the network. To avoid these problems, it is RECOMMENDED that the receiver authenticate the session description, for example by using either the ESP (with enabled authentication service) [9] or AH [8] protocols of IPSEC [7] to ensure the authenticity of the session description.

## 8. IANA Considerations

No information in this specification is subject to IANA registration.

## 9. Intellectual Property Issues

Digital Fountain maintains all rights to the WEBRC technology, but will grant a royalty-free worldwide license to the WEBRC technology once it has been standardized.

WEBRC may be used with other protocols which are proprietary, or have pending or granted patents.

## 10. References

[1] Bradner, S., ‘‘The Internet Standards Process -- Revision 3,’’ RFC2026, October 1996.

[2] Bradner, S., ‘‘Key words for use in RFCs to Indicate Requirement Levels,’’ RFC2119, March 1997.

[3] Byers, J.W., Frumin, M., Horn, G., Luby, M., Mitzenmacher, M., Roetter, A., and Shaver, W.,

‘‘FLID-DL: Congestion Control for Layered Multicast,’’ Proc. Second International Workshop on Networked Group Communications (NGC 2000), Palo Alto, CA, November 2000, pp. 71-81.

[4] Deering, S., ‘‘Host Extensions for IP Multicasting,’’ RFC1112, August 1989.

[5] Handley, M., Padhye, J., Floyd, S., Widmer, J., ‘‘TCP Friendly Rate Control (TFRC): Protocol Specification,’’ Internet Draft draft-ietf-tsvwg-tfrc-03, July 2001, a work in progress.

[6] Holbrook, H. W., ‘‘A Channel Model for Multicast,’’ Ph.D. Dissertation, Stanford University, Department of Computer Science, Stanford, California, August 2001.

[7] Kent, S., Atkinson, R., ‘‘Security Architecture for the Internet Protocol,’’ RFC2401, November 1998.

[8] Kent, S., Atkinson, R., ‘‘IP Authentication Header,’’ RFC2406, November 1998.

[9] Kent, S., Atkinson, R., ‘‘IP Encapsulating Security Payload (ESP),’’ RFC2406, November 1998.

[10] Luby, M., Gemmell, J., Vicisano, L., Rizzo, L., Crowcroft, J., ‘‘Asynchronous Layered Coding protocol instantiation,’’ Internet Draft draft-ietf-rmt-pi-alc-06, February 2002, a work in progress.

[11] Luby, M., Gemmell, J., Vicisano, L., Rizzo, L., Handley, M., Crowcroft, J., ‘‘Layered Coding Transport building block,’’ Internet Draft draft-ietf-rmt-bb-lct-04.txt, February 2002, a work in progress.

[12] Luby, M., Gemmell, Vicisano, L., J., Rizzo, L., Handley, M., Crowcroft, J., ‘‘The Use of Forward Error Correction in Reliable Multicast,’’ Internet Draft draft-ietf-rmt-info-fec-02.txt, February 2002, a work in progress.

[13] Luby, M., Gemmell, J., Vicisano, L., Rizzo, L., Handley, M., Crowcroft, J., ‘‘Forward Error Correction building block,’’ Internet Draft draft-ietf-rmt-bb-fec-06.txt, February 2002.

[14] Mankin, A., Romanow, A., Bradner, S., Paxson V., ‘‘IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols,’’ RFC2357, June 1998.

[15] Perrig, A., Canetti, R., Song, D., Tygar, J.D., ‘‘Efficient and Secure Source Authentication for Multicast,’’ Network and Distributed System Security Symposium, NDSS 2001, pp. 35-46, February 2001.

[16] Vicisano, L., Rizzo, L., Crowcroft, J., "TCP-like Congestion Control for Layered Multicast Data Transfer", Proc. IEEE Infocom ’98, San Francisco, CA, March-April 1998, pp. 996-1003.

[17] Whetten, B., Vicisano, L., Kermode, R., Handley, M., Floyd, S., Luby, M., ‘‘Reliable Multicast Transport Building Blocks for One-to-Many Bulk-Data Transfer,’’ RFC3048, January 2001.

## 11. Authors' Addresses

Michael Luby
luby@digitalfountain.com
Digital Fountain
39141 Civic Center Drive
Suite 300
Fremont, CA, USA, 94538

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

## 12. Full Copyright Statement

Copyright (C) The Internet Society (2001). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."