Optimal Bandwidth-Buffer Trade-off for VBR Media Transmission over Multiple Relay-Servers

R.I. Chang, M.C. Chen, M.T. Ko and J.M. Ho
Institute of Information Science, Academia Sinica, Nankang, Taipei, Taiwan
E-mail: [william, mcc, miko, hooho]@iis.sinica.edu.tw

Abstract
Given a client buffer, the minimum bandwidth required to transmit a pre-recorded VBR media can be computed in O(n) (n is the frame number). As n is usually very large, this resource management procedure is not suitable for online computation. We have proposed an O(n log n) algorithm to characterize the bandwidth-buffer tradeoff for the optimal resource management. However, it cannot be applied to a general multimedia network with multiple relay-servers. In this paper, we extend our problem model to consider relay-servers. In each relay-server, an O(n log n) algorithm is proposed to decide the optimal bandwidth-buffer tradeoff. With these pre-computed tradeoff functions, an O(m) QoS control procedure is designed to allocate the suitable bandwidth for the available buffer where m is the number of relay-servers in the assigned transmission path.

1. Introduction
To support continuous media playback, requests in a multimedia system [3-4] require guaranteed QoS control and resource management (QC-RM) in disks and networks [5-7]. Different from the CBR traffic, media data are usually VBR due to the compression technology applied [1-2]. It makes the design of a good multimedia scheduler more complicated. In a general multimedia network with multiple nodes, the available resources are limited and various in different node and connection. For a coming request, the system needs to know the available resources for deciding the admission of the coming request. To admit as many requests as possible, a good QC-RM procedure should be provided [11,18]. Based on DAVIC [24], client’s request is sent to a level-1 gateway (called AS, application-server) to allocate a suitable information flow (called TP, transmission path). The minimum bandwidth requirement for the allocated TP should be decided to satisfy the resource constraints and optimize the system performance. If the available resources are not enough to support this coming request, the request will be rejected. In the previous years, different approaches [7-15] were proposed to minimize the required resources in transmitting a pre-recorded VBR media stream. In [7], we presented a linear-time traffic smoothing algorithm based on the Lazy scheme (L-scheme) and the Aggressive scheme (A-scheme). We can decide the minimum client buffer and delay time required to transmit the VBR media by the allocated bandwidth. Besides, the network idle rate is minimized under the available client buffer. These optimal resource requirements can be decided in O(n) time (n is the number of frames) [7]. However, as n is usually very large (n = 216000 for a 2-hours movie), this QC-RM procedure is not suitable for online computation. Recently, some approaches have been proposed to off-line compute the optimal tradeoffs among different resources [16-17]. Whenever a new request is presented, the admission control procedure can easily check the required resources against the available resources and decides to admit this new request or not. Given a pre-recorded VBR media, a native algorithm requires O(n^2) time complexity to compute the optimal bandwidth-buffer tradeoff (BBT). It is really time-consuming. In [16], given a media stream, we presented an O(n log n) algorithm to characterize this BBT under the minimum delay time and the minimum bandwidth idle rate. This function depends only on the considered media stream and can be applied to the transmission from any server to any client. The QoS control procedure takes only O(1).
However, this method does not consider the network model with multiple relay-servers (RSs), such as the head-end nodes for a cable modem system. In this paper, we extend the problem model to consider additional RSs. In each RS, there is an incoming-transaction schedule (ITS) and an outgoing-transaction schedule (OTS). Given a media stream, our proposed algorithm can compute the optimal BBT to transmit the VBR media on any TP. Assume that the assigned TP contains m RSs. Based on the pre-computed tradeoff functions, an O(m) QoS control procedure can be designed to allocate the suitable bandwidth for the available buffer in each RS. Notably, these pre-computed tradeoff functions can be applied to any TPs allocated. It is different from the on-line computation approach which requires O(mn) computation time to make the admission control.

Fig. 1. (a) The physical layout of considered system. (b) A possible transmission path \(P_0P_{m-1}...P_3P_i\).

2. System model and problem definition
The physical layout of the considered system architecture is shown in Fig. 1(a). It is a general multimedia network with multiple RSs between the client and the content-servers (servers, for short) with a special AS [24]. The AS contains the complete information of the available resources in the system. As the
operation steps shown in DAVIC [24], client’s request should be sent to the AS to determine a suitable TP (in sequence, the suitable content-server, RSs and the client) in order to support as many requests as possible. Given available buffers in the relay-servers, the best TP is the one requires minimum allocated bandwidths on its relay-servers. Fig. 1(b) shows a simple example with possible TPs. Notably, the tradeoff functions depend only on the media stream and can be applied to different TPs. In the following we consider a certain pre-recorded VBR media $V = \{f_1, f_2, ..., f_n\}$ (where $n$ is the frame number and $f_i$ is the related frame size). Assume that the media stream starts the playback at time 0. For $V$, the cumulative-playback-function (CPF) $F(t)$ for the time $t$ is defined as $F(t) = F(t-1) + f$, and $F(0) = 0$. A TP is denoted by $P_1P_2P_3...P_{n+1}$, where $P_n$ is a content server, $P_1$ is a client and the other $P_{i}$s are relay-servers. For the TP, the following notations are used.

$T_i$: the incoming transmission schedule for the $i$-th network node $P_i$.

$b_i$: the available buffer in $P_i$.

$r_i$: the bandwidth allocated for $T_i$.

The media $V$ is stored on the server $P_n$ based on some data layout schemes [6,20]. When a request is accepted for serving, the related media data can be successfully retrieved from the storage system to the server buffer at the proper time [5-6,20-21]. Then, the transmission schedule $T_{m+1}$ is applied to transmit the media data from the server buffer to $P_{m+1}$. According to the transmission schedule $T_m$, media data in the buffer of $P_{m+1}$ are transmitted to $P_i$. At last, media data are transmitted to the client $P_1$. The client consumes media data in the client buffer frame-by-frame for continuous playback. Namely, the OTS of $P_1$, $T_0$ is exactly $F$.

In this paper, we focus on QC-RM in network transmission. For a given TP, we want to determine the $T_0$ for $0 < i < m$, which require minimum bandwidth according to the available resource in the relay-servers. Further, we want to determine the bandwidth-buffer trade-off on each relay-server that facilitates efficient QoS admission control.

Basically, with $T_0 = F$ is given, $T_1$, $T_2$, ..., $T_{n+1}$ are determined in order. First, $T_1$ with minimum bandwidth is determined according to $T_0$, the OTS of $P_1$ and $b_1$, the available buffer in $P_1$. After $T_1$ is determined, $T_{i+1}$ with minimum bandwidth can be determined according to $T_i$ and $b_{i+1}$, and so on. In fact, the bandwidth-buffer trade-off on $P_{n+1}$ is determined in the same order.

Using the algorithm in [7], given the CPF $T_0 = F$, we can construct an optimal OTS $T_1$ to minimize the delay-time $d_1$, the incoming bandwidth $r_1$, and the related network idle rate $u_1$ under the given buffer $b_1$. For the RS $P_1$, if its OTS $T_1$ is provided the optimal OTS $T_1$ can be computed by viewing this $T_1$ as a kind of CPF for $P_2$ (just like the CPF $T_0$ for $P_1$). However, in a relay server, the related optimal transmission schedules (incoming or outgoing) are generally not unique. As shown in Fig. 2, given a buffer, there are at least three transmission schedules (LA, LAL, LA-Lazy), and S-LA (smoothed-LA) [7]). These optimal transmission schedules have the same bandwidth requirement and the same bandwidth idle rate. In previous approaches [7-17], the computation of the OTS would depend not only on the buffer specified but also on the related OTS. Different OTSs would lead to different ITSs and require different resources. Although we can compute the optimal ITS for each given OTS, there are infinite possible OTSs should be considered. Thus, the best ITS to minimize the bandwidth allocated for these OTSs cannot be determined by previous approaches [7-15]. In [16] we presented an algorithm to explore the optimal BBT between two nodes; however it is valid only for a specific OTS. Without loss of generality, in the following we only present the algorithm for the transmission problems for $P_2$. Our proposed method can compute the optimal transmission schedule and BBT. The same idea can be extended to the $i$-th RS.

![Fig. 2. Given the buffer, there are lots of OTSs have the same optimal bandwidth requirement.](image)

### 3. Optimal bandwidth-buffer tradeoff

Let $T^A$ and $T^{LA}$ denote the optimal ITSs for relay-server $P_1$ that are obtained by the LA algorithm and the LAL algorithm in [7] according to available buffer $b_1$. Notice that by the properties of L-scheme and A-scheme [7], transmission schedules $T_i$ satisfying $T^A(t) \geq T_i(t) \geq T^{LA}(t)$ for any time $t$ are all the ITSs of $P_1$ that have the same delay time $d_1$, the same bandwidth $r_1$ and the same bandwidth idle rate $u_1$ as $T^A$ and $T^{LA}$. Namely, these optimal ITSs for $P_1$ are the possible optimal OTSs for $P_2$. Besides, all these optimal ITSs $T_1$ start the connection at the same start-connection-time $s_1$ and end the connection at the same end-connection-time $e_1$. Among these optimal ITSs of $P_1$ we want to choose one such that the required bandwidth $r_2$ of $P_2$ is minimized in order the multimedia system to support as many requests as possible.

![Fig. 3. Specifying an ITS, we can easily construct an OTS to decide the minimum buffer.](image)

Assume that an ITS $T_2$ is specified $(T_2(t) \geq T^{LA}(t))$. Then a feasible OTS $T_1$ must satisfy $T_1(t) \geq T_2(t)$ and the required buffer at $P_2$ is equal to $\max[T_2(t) - T_1(t); \forall t]$. It is easy to see that the OTS $T_1(t) = \min[T_2(t) - T^{LA}(t)]$, which is feasible, has the minimum buffer requirement at $P_2$. A simple example to illustrate these relations is shown in Fig. 3. Notice that the buffered data size $T_1(t) - T_2(t) > 0$ only when $T_2(t) \geq T^{LA}(t)$, at which $T_1(t) = T^{LA}(t)$. Namely, the minimum required buffer is in fact equal to $\max[T_2(t) - T^{LA}(t); \forall t]$. When $T^{LA}(t)$ is determined, the minimum required buffer at $P_2$ depends on $T_2(t)$ only.
On the other hand, given the available buffer size $b_2$ at $P_2$, by the above argument, $T_3(t)$ has to be between $T^{LA}(t)$ and $T^{LA}(t)+b_2$, as shown in Fig. 3(a). When the bandwidth $r_2$ of $T_2$ is specified, to minimize the required buffer, the value of $T_3(t)$ needs to be as small as possible under the constraints $T_3(t) \geq T^{LA}(t)$ and the initial value $T_3(0) = T^{LA}(0) = T^{LA}(t) = W/I$. With these constraints, it is obvious that $r_2$ must be smaller than or equal to $r_1$. In fact, for any $r_2 \leq r_1$, the optimal transmission schedule $T_2$ and $T_3$ can be computed by the following O(n) algorithm.

1. Pre-compute $T^{LA}$ and $T^{LA}(0)$. Initialize $T_3(0) = W/I$.
2. $T_3(t) = \max\{T_3(t-1) - r_2, T^{LA}(t)\}$ for all $t$.
3. $T_3(t) = \min\{\max\{T_3(t+1) - r_2, T^{LA}(t)\}, T^{LA}(t)\}$ for all $t$.

Notice that if $T_3(t) > T^{LA}(t) + b_2$ for some $t$, it implies the bandwidth $r_2$ is too small that the required buffer exceeds $b_2$. In summary, when $r_1$ is given, for any $r_2 \geq r_1$, we may construct an optimal transmission schedule $T_2$ with bandwidth $r_2$ such that its required buffer at $P_2$ is minimum. Furthermore, the minimum required buffer depends on $T^{LA}$ and $r_2$ only.

When the available buffer size $b_2$ is close to 0, the data incoming-transmitted should be outgoing-transmitted immediately. The ITS bandwidth $r_2$ would be close to the OTS bandwidth $r_1$. On the other hand, if the available buffer size $b_2$ is the same as the media size $W/I$, we can apply the well-known stored-and-forward scheme and the required bandwidth can be very low.

We have shown that the buffered data size in $P_2$ depends on $T^{LA}$ and the ITS $T_2$. As we know, $T^{LA}$ is an on-off function with interleaved on-transmission segments (on-segment, for short) and off-transmission segments (off-segment, for short) shown in Fig. 4(a). Notably, there are at most $n$ on-segments and $n+1$ off-segments. A simple observation is that the maximum buffer of $T^{LA}$ is at the start point of some on-segment in $T^{LA}$. The start points of on-segments in $T^{LA}$ are called the buffer points. In Fig. 4, we mark each buffer point by a "box". Then, when we compute the BBT on $P_2$, we can simply consider the buffer changes at buffer points of $T^{LA}$ as $r_2$ changes. Consider first the boundary condition that $b_2$ is close to $W/I$. In this case, the ITS $T_2$ has one on-segment. By decreasing the available buffer, the off-segments are introduced and started at the points called the segment-points. From the definition of $T_2$, they are just the start points of the off-segments on $T^{LA}$. As shown in Fig. 4 we mark each segment-point by a "circle". There are at most $n$ buffer-points and $n$ segment-points. In each on-segment of $T_2$, we define the buffer-point of $T^{LA}$ that has the maximum buffered data in this on-segment as the segment-buffer-point (SBP). Notably, this SBP must be one of these buffer-points in the related on-segment. The SBP that achieves the maximum buffer requirement $b_2$ is called the maximum-buffer-point (MBP, or schedule-buffer-point) for the schedule $T_2$. The on-segment with the MBP is called the maximum buffer segment. Now, we consider the maximum buffer segment in $T_2$ and try to decrease the available buffer from $b$ to $b'$ as shown in Fig. 4(b). Assume that the MBP is not changed and no new off-segment is created. As shown in Fig. 4(b), in some range of the available buffers (i.e., $[b', b]$), the required bandwidth is increased by a constant slope $k$ when the available buffer is increased. This linear tradeoff slope is called the buffer-decreasing-slope (BDS, or bandwidth-increasing-slope). The same results can be applied to the SBPs in other on-segments. Thus, although the MBP may switch to another segment (or shift to another index point), the available buffer is also linearly increased and continuously changed when the required bandwidth is linearly decreased. Notably, as the applied BDS may be changed (i.e., the MBP is changed), the function would be piecewise-linear. The optimal BBT is piecewise-linear and continuously decreasing.

![Fig. 5. When the bandwidth rate increases, a new off-segment would be inserted at the index $t'$.](image)

To formulate the optimal BBT, we should identify the bandwidth rates at which the related BDS would be changed. Besides, as the MBP may switch to any other on-segments, we should keep track the changes of the BBT in each on-segment. In this section, we extend the BDS concept $k = (b' - b) / (r' - r)$ to each on-segment. From this extended definition of BDS $k$ (the difference between the SBP and the end-transmission point in the related on-segment), the slope may change when the end-transmission point is changed or the SBP is changed. The first case may happen when the related on-segment is separated. Look into the case that a segment is separated as shown in Fig. 5. The obtained transmission schedule $T_2$ with the bandwidth rate $r'$ just touches $T^{LA}$ at the index $t'$. When the bandwidth rate increases from $r'$ by a small value, a new off-segment is inserted with the start point $t'$. Such a bandwidth rate $r'$ is called a segment-separating-rate (SSR). Notably, the original on-segment is separated into the right sub-segment and the left sub-segment. If the SBP is at the right sub-segment (as shown in Fig. 5(a)), the related end-transmission point is not changed and the BDS would not change. If the SBP is at the left sub-segment (as shown in Fig. 5(b)), the related end-transmission point is changed to $t'$. Thus, the BDS would be changed to $k'$. We can find that $k' < k$. In Fig. 5, the related changes of the BBT in each on-segment are also shown. Notably, when the bandwidth rate is increased, segments may be separated further. The number of on-segments is increased from 1 to $n$ when the available buffer is decreased from $W/I$ to 0. The new added on-segment will
introduce a new BBT as shown in Fig. 5. We select the maximum buffer requirement as the available buffer. If the on-segment is not separated, we should consider the second case with the changed SBP. This case is happened at the equal-buffer-rate (EBR) as shown in Fig. 6. The EBR \( r' \) is defined as the slope of the line segment from \( T^{BL}(r') \) to \( T^{AL}(r') \). The index \( i' \) and \( j' \) are two different buffer-points defined in the section 3.2. Assume that these two points are at the same on-segment for the transmission schedule \( T'_2 \) with the bandwidth rate \( r' \). There is a parallelogram between the transmission schedule \( T'_2 \) and the line segment from \( T^{BL}(r') \) to \( T^{AL}(r') \). From this parallelogram, we can easily prove that the buffered data size at time \( r' \) is the same as the buffered data at time \( j' \). Thus, we call \( r' \) as an EBR. If the bandwidth rate is slightly smaller than \( r' \), the buffered data at time \( r' \) would be larger than that at time \( j' \). Assume that \( r' \) is just the SBP for the related on-segment (as shown in Fig. 6). When the bandwidth rate is slightly larger than \( r' \), the SBP would be changed from \( i' \) to \( j' \). Thus, the BDS is changed from \( k \) to \( k' \). It can be easily proved that the value of BDS would be decreased (\( k' < k \)). We have a decreasing BDS for the optimal BBT.

4. Proposed optimal tradeoff algorithm

Notably, in the above section, we consider only the BBT in each on-segment. However, the MBP may also shift from one on-segment to another on-segment. This case should be handled by combining the BBT in each on-segment to find the maximum buffer requirement. It is the basic idea of our proposed algorithm. Before describing our proposed O(nlogn) algorithm, we first identify all the SSRs and the related EBRs in each on-segment by a linear-time procedure. Then, a construction algorithm of the optimal BBT function is proposed. Based on this function, an O(1) QoS control procedure is designed to allocate the minimum bandwidth for the available buffer in the RS [16].

![Fig. 6. When the bandwidth rate is slightly larger than \( r' \), the segment-buffer-point would be changed from \( i' \) to \( j' \).](image)

4.1. Identify the SSRs and the related EBRs

As we know, both \( T^{BL} \) and \( T^{AL} \) can be represented by a set of on-segments and off-segments. The start point of an on-segment is an inner-corner of the transmission schedule. An outer-corner of the transmission schedule is the start point of an off-segment. By increasing the bandwidth rate from 0 to \( \infty \), a linear-time algorithm with an O(n) heap structure is proposed to exploit all the SSRs. It is similar to construct the convex upper envelope of the outer-corners in \( T^{AL} \) as shown in Fig. 7(a). All these outer-corners are under the convex upper envelope to represent these on-segments by a tree structure. Based on this tree structure, we can identify the related EBRs in each on-segment. As shown in Fig. 7(b), it is similar to hierarchically construct the the convex lower envelope of the inner-corners in \( T^{BL} \) based on the tree structure of on-segments. In this paper, we denote the outer-corners of \( T^{AL} \) by \( c_{k,i} \) for \( k = 1 \) to \( p \). The inner-corners of \( T^{BL} \) is denoted by \( c'_{k,j} \) for \( k = 1 \) to \( q \). Notably, \( p \leq n \) and \( q \leq n \). To construct a tree structure of all these on-segments, we first define the angle that counterclockwise from a line segment to its end point \( x \) (line 22) as \( \pi \). The step-by-step description of the proposed construction algorithm is:

1. Push the point \( c'_{k,1} \) to the heap twice as the line \( c'_{k,1} \) to \( c'_{k,2} \), \( c'_{k,2} \) to \( c'_{k,3} \), and so on. Pop the point \( c'_{k,1} \) from the heap. Whether the angle that counterclockwise from line \( c'_{k,1} \) to \( c'_{k,2} \) to \( c'_{k,3} \) is less than or equal to \( \pi \).

2. If the angle is less than or equal to \( \pi \) the convex upper envelope from \( c'_{k,1} \) to \( c'_{k,2} \) together with the new segment \( c'_{k,2} \) is the resulted convex upper envelope from \( c'_{k,1} \) to \( c'_{k,2} \). Push the points \( c'_{k,1} \) and \( c'_{k,2} \) to the heap.

3. If the angle is larger than \( \pi \) the convex upper envelope from \( c'_{k,1} \) to \( c'_{k,2} \) is the resulted convex upper envelope from \( c'_{k,1} \) to \( c'_{k,2} \). Push the points \( c'_{k,1} \) and \( c'_{k,2} \) to the heap.

(6) \( k = k - 1 \). Go to step (2).

![Fig. 7. (a) Construct the segment-separating-tree. (b) Construct the equal-buffer-rates in each separating segments.](image)
4.2. Constructing the optimal tradeoff

With the data structures for the SSRs and the related EBRs, we can maintain the structure of separating on-segments and the related SBPs/sizes as follows. We process the stored SSRs in a decreasing order (from root to leaf). Between two SSRs, the related EBRs are considered. The processing steps are as follows:

1. If the selected rate is an SSR, the changes of the decreasing largest buffers are as shown in Fig. 5. A new on-segment is introduced and the related BDS may be changed.

2. If the selected rate is an EBR, we can just change the SBP. Fig. 6 is the related changes of the decreasing largest buffers. Notably, the related BDS of the largest buffer is changed.

3. The maximum buffer is just the upper envelope of these largest buffers for different on-segments. We maintain the optimal BBT by finding a new upper envelope from the original upper envelope and the new added on-segment with the decreased BDSs as shown in Fig. 8.

When a rate (SSR or EBR) is selected for processing, one or two lines are inserted to the original upper envelope. These lines represent the possible changes of BDS for the BBT. By finding the intersection points of the added lines to the original upper envelope, we can construct the new upper envelope in $O(\log n)$ time.

Since there are $O(n)$ such rates should be considered, the time complexity of the proposed algorithm is $O(n \log n)$. We can extend the same idea to the $i$-th relay server to compute the related tradeoff functions for QC-RM.

5. Concluding remarks

In this paper, an algorithm is proposed to decide the optimal BBT for a general multimedia network with multiple RSs. This approach shows great flexibility to allow various clients and relay servers to set up their best transmission schedules. As the required initial delay depends only on the transmission rate and the end-point of the first on-segment, we can easily apply the same idea to decide the optimal bandwidth-delay tradeoff.

References


