Songklanakarin J. Sci. Technol.
30 (1), 71-84, Jan. - Feb. 2008

Songklanakarin Journal of Science and Technology
http://www.sjst.psu.ac.th

Original Article

Adaptive quality control for multimedia communications

Santichai Chuaywong*, Sinchai Kamolphiwong, Thossaporn Kamolphiwong, Kevin Robert Elz and Suthon Sae-Wong

Centre of Network Research (CNR), Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University, Hat Yai, Songkhla, 90112 Thailand.

Received 18 April 2007 ; Accepted 26 February 2008

Abstract

Multimedia communications are communications with several types of media, such as audio, video and data. The current Internet has some levels of capability to support multimedia communications, unfortunately, the QoS (Quality of Service) is still challenging. A large number of QoS mechanisms has been proposed; however, the main concern is for low levels, e.g. layer 2 (Data Link) or 3 (Transport). In this paper, mechanisms for control the quality of audio and video are proposed. G.723.1 and MPEG-4 are used as the audio and video codec respectively. The proposed algorithm for adaptive quality control of audio communication is based on forward error correction (FEC). In the case of video communication, the proposed algorithm adapts the value of key frame interval, which is an encoding parameter of MPEG-4. We evaluated our proposed algorithms by computer simulation. We have shown that, in most cases, the proposed scheme gained a higher throughput compared to other schemes.

Keywords: Adaptive, multimedia communications, quality control, FEC, MPEG-4

1. Introduction

Internet is the big widely-used world network. There is a huge variety of services that are using the Internet, not only data (WWW, FTP, etc.), but also communications of other types of medium such as audio and video. Since the current Internet does not guarantee QoS, the packet loss problem always occurs. In multimedia communications, it is not guaranteed that the source media, transmitted by the sender, and the received media, at the receiver, will have the same quality because some packets may be dropped in the network. For data traffic, packet loss can be solved by retransmission, but this approach cannot be applied to real-time traffic such as audio and video. A number of packets lost can degrade media quality at the receiver end. So it is necessary for the sender to monitor the quality of media received by the receiver, and to adapt its transmission parameters in order to control the quality of the received media in acceptable level. There are a number of previous works proposing adaptation techniques for audio and video services, but there is still no perfect solution. In this paper, we propose new adaptive quality control algorithms for audio and video communications over the Internet, so call "CNR algorithm".

The remainder of this paper is organized as follows. In section 2, background information of fundamentals of related topics is given. In section 3, some of important related works are described. Then the proposed mechanism is presented the next section, together with the proposed algorithm in section 5. The simulation results are shown and compared in section 6, following with section 7: result discussion. Finally, the conclusion is presented in section 8.

2. Background information

In this section, some topics of necessary background information are described.

*Corresponding author.
Email address: suntichai@coe.psu.ac.th

2.1 Forward Error Correction (FEC)

When the audio traffic is sent over the internet, a problem that always occurs is the packet loss. If there are too many lost packets, audio quality can be degraded. So it is necessary to use the error recovery mechanism for reducing the effect of the packet loss. Forward Error Correction (FEC) is an effective error recovery mechanism with low latency.

FEC is separated into two classes, media independent FEC and media specific FEC. In the media independent FEC, when a set of packets is sent, it produces the additional packet. This additional packet is generated from the arithmetic operation of all packets in the same set. But this scheme cannot protect the loss of several packets which are in the same set.

Figure 1 shows the second FEC scheme, the media specific FEC. In this scheme, each packet contains the redundant version of the prior packets. But the audio codec of the redundant data may have lower bit rate than the primary codec, for bandwidth saving. For example, the primary codec is G.711 u-law (64 kbps) and the redundant codec is LPC (4.8 kbps). But if the bit rate of primary codec is low, the redundant data can use the same codec as the primary data. In Figure 1, if the packet numbered 3 is lost, the receiver can recovery this audio unit from the packet numbered 4.

In this paper, we focus on media specific FEC because it is more effective than the media independent FEC. The packet format for redundant audio data is defined in RFC 2198 (Perkins *et al.*, 1997). When the FEC mechanism is applied in audio communication, there are two values of loss rate to be considered. They are loss rate before reconstruction (L_b) and loss rate after reconstruction (L_a). L_b is only the packet loss rate and La indicates the loss of audio data.

2.2 MPEG-4

MPEG-4 (ISO/IEC 14496-2, 2001) is an open standard developed by the Moving Picture Expert Group (MPEG) for low bit rate digital media applications. There are several parts in MPEG-4 such as the system, video coding, audio coding, file format, etc. But in this paper, we focus on the part of video coding, which is officially called MPEG-4 Visual part. The input video of MPEG-4 encoder must be in the YCbCr color space.

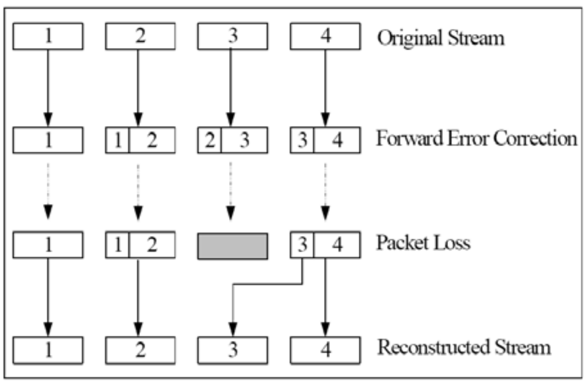

Figure 1. Media Specific FEC

1) MPEG-4 Frame Types

The compression method of the MPEG-4 encoder concerns both spatial and temporal redundancy, so it uses inter-frame prediction. In MPEG-4 standard, the video frame is called Video Object Plane (VOP). But for easier understanding, in this paper we will use the word frame instead of VOP. There are three types of MPEG-4 encoded frames, intra frame (I-frame), predicted frame (P-Frame) and bidirectional frame (Bframe). The dependency between each type of frames is shown in Figure 2.

I-frame is the intra-coded frame which is encoded by reducing the redundancy in the same frame only. P-frame is the forward prediction frame and it is depend on previous I-frame or P-frame. So it can reduce the temporal redundancy. B-frame is both forward and backward prediction frame. It is depend on previous and succeeding non B-frame. Since the encoding of B-frame has to wait for the succeeding frame, so this frame type is not suitable for real-time video transmission.

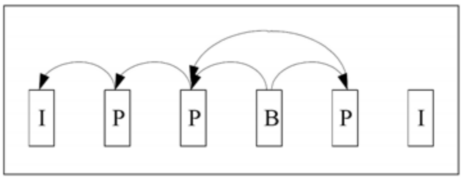

Figure 2. MPEG-4 Frame Types

Since the encoding process of MPEG-4 uses inter-frame prediction, so the lost of a video frame can affect the succeeding frames. Figure 3 shows the error propagation when a video frame is lost. The error propagation will be recovered only when an Iframe is received because the I-frame does not depend on other frames. So I-frame is often called key frame or reference frame and in this paper we use the words I-frame and key frame interchangeably.

The key frame interval is a parameter of the MPEG-4 encoder. It is the distance between adjacent I-frames. For example, in Figure 4, the key frame interval is 5 frames. If

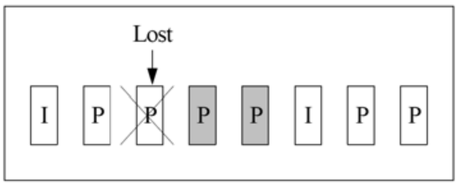

Figure 3. Error Propagation in MPEG-4 Decoder

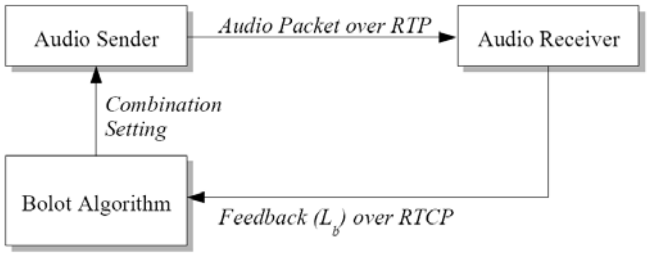

Figure 4. Diagram of adaptive error control with Bolot Algorithm

key frame interval is low, the error propagation will be decreased. But the video bit rate is increased because the lower value of key frame interval means that there are more Iframes. And the size of I-frame is much bigger than P-frame because I-frame stores information of the entire frame while P-frame stores only difference between current and previous frame.

2) Rate Control in MPEG-4 Encoder

The output bit rate of video encoder will be varied based on the characteristics of video such as motion, temporal redundancy, and spatial redundancy. Rate control involves modifying the encoding parameters in order to control the output video bit rate to meet the target bit rate. The most widely-used parameter is the quantization scale or step size because increasing quantization scale can reduce the encoded frame length and lead to the higher distortion (lower image quality). In the other hand, reducing quantization scale will reduce distortion but increase the encoded frame length (Richardson, 2003).

3. Related works

In this section, some important and well known related works are described and analysed as follows:

3.1 Adaptive FEC-Based Algorithm for Audio Communication

Since the FEC mechanism increases the bandwidth usage. Sending too much audio redundancy may be waste of bandwidth if the packet loss rate is low. So the adaptive algorithm is required to determine the quantity of audio redundancy. The previous works in adaptive FEC-based error control algorithm are Bolot algorithm (Bolot and Garcia, 1996), USF algorithm (Padhye *et al.*, 2000) and RCCS algorithm (Ji *et al.*, 2001).

1) Bolot Algorithm

Bolot *et al.* proposed an adaptive FEC-based algorithm to reduce the effect of audio packet loss in (Bolot and Garcia, 1996). They do several experiments to find a value

Table 1. Combination table used in Bolot algorithm

No.	Combination	Reward
0	(PCM)	1
1	(PCM, ADM4(1))	2.5
2	(PCM, GSM(1))	2.5
3	(PCM, LPC(1))	2.5
4	(PCM, ADM4(2))	6
5	(PCM, ADM4(1), ADM2(2))	6
6	(PCM, ADM4(1), ADM2(3))	10
7	(PCM, ADM4(1), ADM2(2), ADM2(3))	18

called "reward" of each combination of audio redundancy. The reward is the ratio of L_b and L_a.

$$\text{Reward} = L_b / L_a$$

Table 1 shows detail of combinations used in Bolot algorithm. This algorithm will change combination based on audio loss rate. The meaning of the syntax in the second column (combination), for example, syntax (PCM, ADM4 (1)) means each packet N contains the primary data (encoded with PCM) and the redundant version of packet N-1 (encoded with ADM4). The syntax of other combinations can be interpreted in the equivalent way. The reward of each combination is the empirical value that tells the effectiveness that each combination can recover the loss of audio data.

According to the diagram in Figure 4, Bolot algorithm does not use the real value of the loss rate after reconstruction (L_a). But it calculates L_a from L_b (feedback from the receiver) divided by the reward of the current combination and then uses the calculated L_a to determine which combination to use. The pseudo code of this algorithm is shown in Figure 5. Bolot algorithm will increase combination (increase selected combination number in Table 1) if L_a is over HIGH threshold and decreasing combination if L_a is lower than LOW threshold.

```
For each RTCP packet received do
   1.Calculate loss rate before reconstruction, Lb
   2.Calculate loss rate after reconstruction
      La = Lb / Reward associated with current
               combination number
   3. If (La > HIGH) then
         Increment combination
   4. If (La < LOW) then
         Decrement combination
```

Figure 5. Pseudo code of Bolot Algorithm

2) USF Algorithm

The USF Algorithm (Padhye *et al.*, 2000), improves some drawback of Bolot algorithm. In the paper of Padhey (Padhye *et al.*, 2000), the authors said that the Bolot has two drawbacks.First, it uses empirical value of reward to calculate L_a. This value may not be correct in all conditions. So the USF algorithm uses the real value of L_a. The second drawback of Bolot algorithm is the cyclical behavior. When the value of L_a in the current period is less than the HIGH threshold, the combination number is decreased. In the next period, the decreased combination is not reliable enough. So the value of L_a becomes over the HIGH threshold and the combination number is increased to the same value. The value of L_a in the next period will be less than HIGH threshold again and the combination number will be changed cyclically. In the USF algorithm, this problem is solved by using the difference of L_b in the current and previous periods. The combination number will not be decreased if the difference of Lb is not big enough.

In fact, the cyclical behavior of the Bolot algorithm occurs when the HIGH threshold is too closed to LOW threshold. Especially in this paper (Padhye *et al.*, 2000), the HIGH threshold and LOW threshold are the same value.

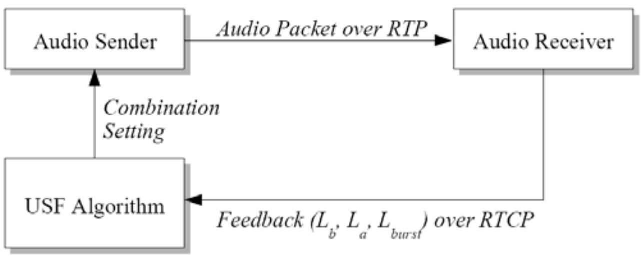

Figure 6. Diagram of adaptive error control with USF Algorithm

```
For each RTCP packet received do
   1.Calculate loss rate after reconstruction, La
   2. Calculate loss rate before reconstruction, Lb
   3. If (La > HIGH) then
         La = La - Lburst
   4. If (La > HIGH) then
         Increment combination
   5. If (La < LOW) then
         Loss difference = Lb(previous) - Lb
   6. If (Loss difference > MINIMUM_THRESHOLD) then
         Decrement combination
   7. Set Lb(previous) = Lb
```

Figure 7. The pseudo code of the USF algorithm

According to the diagram of USF algorithm in Figure 6, the receiver has to feed back 3 parameters to the senders, L_b, L_a and L_{burst}. L_{burst} is the loss rate after reconstruction that counts only packets in the burst loss periods. In the paper of Padhye (Padhye *et al.*, 2000), burst loss means the consecutive loss of 10 packets or higher. USF algorithm avoids increasing combination in the case of burst loss. This is not a good idea. In the case of burst loss, although increasing combination cannot recover all of the lost packets, it must be better than using the same combination. Further detail of this algorithm can be considered through its pseudo code in Figure 7.

3) RCCS Algorithm

Redundant Codec Combination Selection (RCCS) algorithm (Ji *et al.*, 2001) uses both loss rate and end-to-end delay to determine which combination of audio redundancy to use. But the authors did not explain how to find the end-to-end delay clearly. This algorithm uses not only the reward, but it also uses the penalty value. The penalty is the ratio of the end-to-end delay after and before reconstruction.

$$\text{Penalty} = \frac{\text{End-to-End Delay after Reconstruction}}{\text{End-to-End Delay before Reconstruction}}$$

Table 2 shows combination table used in RCCS algorithm. The initial rewards are taken from Bolot experiment but the penalties are defined in this algorithm. RCCS algorithm wants to avoid the drawback of Bolot algorithm that uses the empirical reward value. So, when receive the RTCP receiver report packet is received, this algorithm will change the value of reward of the current combination.

According to the diagram of the RCCS algorithm in Figure 8 and its pseudo code in Figure 9, there are many parameters used in this algorithm. The description of theses parameters is in Table 3. From the pseudo code, after calculating the value of L_b, L_a, D_b and D_a in step 1-2, the next step is updating reward and penalty with the following filter equations.

Table 2. Combination Table used in RCCS algorithm

No.	Combination	Initial Reward	Penalty
0	(G.711)	1	1
1	(G711,GSM(1))	2.5	1.5
2	(G711,G.723(1))	2.5	2
3	(GSM,G.723(1))	2.5	4
4	(G711,GSM(1),G.729(2))	6	2.4
5	(G729,G.723(1),LPC10(2))	6	4.5
6	(G711,GSM(1),G.729(2), LPC10(3))	18	3.4
7	(G711,GSM(1),G.723(2), LPC10(3))	18	4.5

Table 3. Description of parameters used in RCCS algorithm

Parameters	Description
R_i	Reward of combination number i
P_i	Penalty of combination number i
L_a (L_a')	Loss rate after reconstruction, L_a' is the predicted value
L_b	Loss rate before reconstruction
D_a	End-to-End Delay after reconstruction
D_b	End-to-End Delay before reconstruction
α	a Smoothing factor in the filter equation for updating R_i and P_i

$$R_i = \alpha(L_a / L_b) + (1-\alpha)R_{i-1}$$

$$P_i = \alpha(D_a / D_b) + (1-\alpha)P_{i-1}$$

In step 4, selecting the combination, this algorithm will consider each combination in the combination table and select the first combination that matches the following two conditions.

- Produce the value of L_a' lower than HIGH threshold and higher than LOW threshold.
- Produce the value of D_a' lower than Dmin (threshold of end-to-end delay).

If there is no combination that matches these conditions, this algorithm will not change the used combination.

4) Drawback of the Previous Works

Bolot algorithm does not use the real value of L_a for increasing or decreasing combination. But it calculates L_a from L_b divided by reward of the current combination. The value of reward got from the experiment, so it may not be correct in all environments.

USF algorithm improves the drawback of Bolot algorithm by using real value of L_a. But there is a new problem, the oscillation of L_a. If L_a changes with high oscillation, low L_a does not mean low network congestion. This behavior can affect the USF algorithm to be misunderstanding. In the step of decrement combination of this algorithm, it considers both L_a and L_b-difference. But this mechanism does not solve this problem because L_b can be changed with high oscillation too.

RCCS algorithm has a drawback in the step of selecting combination. This algorithm does not separate the increasing and decreasing conditions. So it is possible that this algorithm does not increase combination although the value of La is higher than HIGH threshold if there is no combination that matches the specified conditions.

3.2 Adaptation Techniques for Video Communication

The previous works on adaptation techniques for video communication can be separated into 3 schemes, rate adaptation, adaptive error correction and receiver-driven adaptation. In this paper, we will focus on the first two schemes because this paper focuses on point-to-point communication. According to previous works on rate adaptation and adaptive error control techniques, there is no algorithm that can control the video quality obviously.

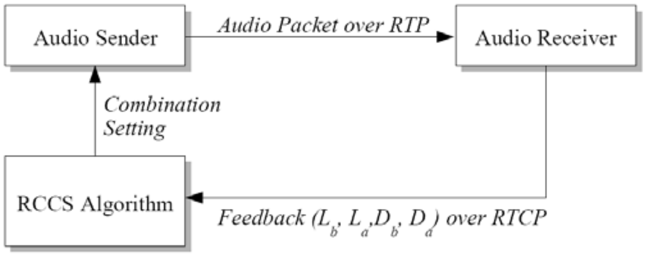

Figure 8. Diagram of adaptive error control with RCCS Algorithm

```
For each RTCP packet received do
1.Calculate packet loss and delay before reconstruction
Lb and Db
  Lb = Number of packet loss before reconstruction /
Number of
       packet expected
  Db = end-to-end delay before reconstruction

2.Calculate packet loss and delay after reconstruction La
and Da
  La = Number of packet loss after reconstruction /
Number of
       packet expected
  Da = end-to-end delay after reconstruction

3.Update reward and penalty of current combination with
Ri and Pi
  in combination table

4.If(La > HIGH or La < LOW)
    For each combination j in the combination table
       Calculate and predict La', La' = Lb/Rj
       Calculate and predict Da', Da' = Db*Pj
          If(La' < HIGH and La' > LOW)
             If(Da' < DMin)
                selected combination = j
                Dmin = Da'
5.combination = selected combination
```

Figure 9. Pseudo code of RCCS algorithm

Most of previous works on rate adaptation techniques change video bit rate based on the packet loss rate, feedback from the receiver. In (Busse *et al.*, 1996), the authors propose method for setting video bit rate. This method will reduce video bit rate when the network is congested (packet loss rate is too high) and increase video bit rate when loss rate is low enough. The value of video bit rate is adapted based on additive increase multiplicative decrease (AIMD) schemes. This method was evaluated by doing experiments in a real network. The experiments used vic (vic, 2005), a well-known video conferencing software, for transmitting video and it can specify the target bit rate. But there is no explanation about how to set the transmission parameters in order to meet the target bit rate.

In the paper of Sisalem and Schulzrinne (1998), (2000), the authors proposed the methods for adapting video bit rate based on TCP-friendly scheme. The formulas for calculating video bit rate use both packet loss rate and round trip time (RTT) between sender and receiver. But these two papers did not say how to set video parameters in order to meet the target bit rate.

IVS (Turletti and Huitema, 1996), video conferencing software, which uses H.261 as the video codec, also adapts video bit rate based on packet loss rate. It will reduce video bit rate when packet loss rate is too high and increase bit rate when packet loss rate is low enough. After calculating value of video bit rate, it will change video parameter in order to meet the target bit rate. There are 2 modes of adaptation, Privilege Quality (PQ) mode and Privilege Frame Rate (PFR) mode.

In PQ mode, frame rate will be adapted in order to meet the target bit rate. The quantization scale of each frame is constant, which means that the spatial quality of video frame is not changed. In the PFR mode, frame rate is constant, but quantization scale is varied. So the spatial quality of each video frame is varied too.

All of the rate adaptation techniques, described above, assume that reducing video bit rate can reduce network congestion and the loss rate of video packets will be reduced too. But this assumption may be not correct in some situations, especially in the almost fully utilized network. Besides, the packet loss rate does not actually reflect the video quality seen by the viewer.

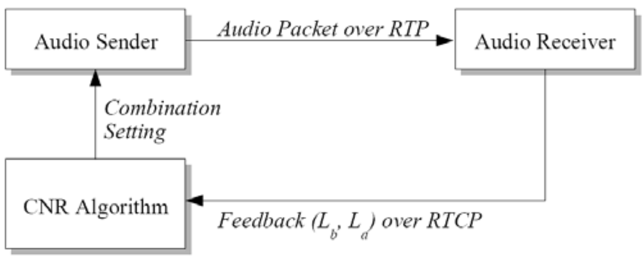

Figure 10. Diagram of adaptive error control with CNR Algorithm

```
For each receiver report packet received do

1.Update reward of the current combination
    //Let Rc stands for reward of the current combination
    Rc = Lb/La

2.if(Lb < LOW)
    increment count_Lb_under_low
  else
    count_Lb_under_low = 0

3.if(La > HIGH)
    selected_combination = MAX_COMBINATION;
    for(i = current_combination+1 to MAX_COMBINATION)
        Ri = reward of combination No.i
        Lai = Lb/Ri
        if(Lai <= HIGH)
            selected_combination = i
            break;
    current_combination = selected_combination;
    count_La_under_low = 0

  else if(La < LOW)
    increment count_under_low

  else
    count_La_under_low = 0

4.if((count_La_under_low >= MIN_UNDER_LOW)OR
     (count_Lb_under_low >= MIN_UNDER_LOW))
    decrement current_combination
    count_La_under_low = 0
```

Figure 11. Pseudo code of CNR algorithm

Table 4. Combination table used in CNR algorithm

No.	Combination Format	Initial Reward (Bolot Reward)
0	-	1
1	-1	2.5
2	-2	6
3	-1-2	6
4	-1-3	10
5	-1-2-3	18

The second scheme of adaptation technique, adaptive error control, can be separated into 2 methods. The first one uses FEC to reduce effect of packet loss (Bolot and Turletti 1996). But using FEC with video transmission seems to be inappropriate because the size of video data is much bigger than audio. Using FEC will increase large of bandwidth usage. The second method of error control is adapting the parameters of video encoder. For example, IVS reduces key frame interval when network is congested. But the paper of Turletti and Huitema, (1996), it did not describe how to set the value of key frame interval.

4. CNR Algorithm: Proposed adaptive quality control for audio communication

Since previous works on adaptive FEC-based algorithms have some drawbacks, so we propose a new adaptive error control algorithm, called CNR algorithm (CNR: Centre for Network Research, where this work has been done). CNR algorithm removes the drawback of Bolot algorithm by using real value of L_a. And the oscillation of L_a is solved by counting the number of periods that L_a lower than LOW threshold. CNR algorithm decreases conditions of the combination when low-loss period is big enough. The drawback of RCCS algorithm is solved by separating the condition for increment and decrement combination. According to the diagram of CNR algorithm shown in Figure 10, the receiver has to feedback Lb and La to the sender by the RTCP receiver report packet. Value of L_b is filled in the fraction lost field and L_a is put the RTCP header extension part.

Table 4 shows the combinations used in the CNR algorithm. All of the previous algorithms use different codec in primary and redundant audio data. But in CNR algorithm, the audio codec for both redundant and primary part is G.723.1 because the bit rate of this codec is low enough (6.3 kbps). The format of combination of audio redundancy is in the second column. For example, the syntax -1 means that there is a redundant version of packet N-1 in every packet N. The syntax -1-2 means that the redundant version of packet

N-1 and N-2 are placed in every packet N. The syntax of other combinations can be interpreted in the equivalent way. The third column is the initial reward, taken from Bolot reward. CNR algorithm used reward to predict the next combination when L_a becomes too high. But value of reward will be changed based on real network status when the sender receives the receiver report.

The pseudo code of CNR algorithm is shown in Figure 11. After receiving the value of L_a and L_b, the first step is updating reward of the current combination. The second step is counting the number of periods that L_b is lower than LOW threshold consecutively and saves in count_L_b_under_low. This variable will be checked in the part of decrement combination. The third step is checking the value L_a. If L_a is over the HIGH threshold, it means that the current combination is not robust enough, so it is necessary to find the next combination. The selected combination is the first upperorder combination that produces the value of predicted L_a (Lai in the pseudo code) not over the HIGH threshold.

If value of L_a is lower than LOW threshold, CNR algorithm does not decrease combination as in Bolot and USF algorithm, but it will count the number of periods that L_a is lower than LOW threshold consecutively and saves in count_L_a_under_low. To make sure that it is safety for trying to decrease combination, CNR algorithm will decrease combination when the value of count_L_a_under_low or count_L_b_under_low reaches to MIN_UNDER_LOW. This mechanism can solve the problem of L_a oscillation. We set the value of MIN_UNDER_LOW to 10 periods (50 seconds). The result in the next section show that using this value is more effective in loss protection, compared with other algorithm.

5. Adaptive key frame interval adjustment algorithm

The transmission of video over the internet can be affected by packet loss. Reducing the key frame interval is an approach that can minimize the effect of packet loss because it reduces the error propagation in P-frames. Although this approach will increase the video bit rate, but this problem can be solved by using rate control. In the MPEG-4 encoder that applies the rate control, reducing key frame interval will increase the mean quantization scale. The disadvantage of increasing quantization scale is the higher distortion. But the advantage is that the mean frame length will be decreased. When the encoded frame is bigger than the maximum transfer unit (MTU) of transmission link, it will be fragmented. Smaller frames, which have a lower number of fragments, will have higher probability of completely transmitting of the entire frame to the receiver than bigger frames.

It is hard to specify the value of key frame interval because of the tradeoff between quantity of distortion and error resilience. So we propose a new adaptive algorithm that changes the value of key frame interval based on network status. Figure 12 presents the architecture of the adaptive key frame interval adjustment. Since video is transmitted over RTP, so the receiver can send feedback information with RTCP. The information that the quality control unit uses for specifying the key frame interval are PSNR, reported by the encoder, and received frame rate, reported by the receiver.

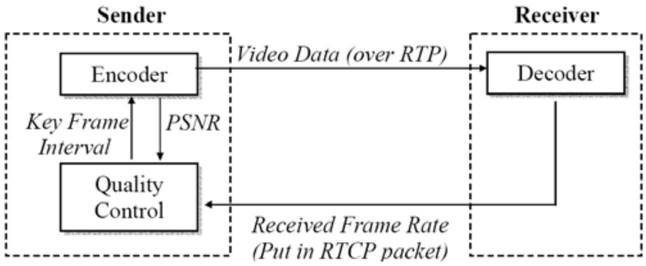

Figure 12. Architecture of adaptive key frame interval adjustment

```
When receive each RTCP receiver report:

Frcv = The value received frame rate in the
RTCP
      extension header.
if((Frcv < Flow) AND (PSNRavg >= MIN_PSNR))
//Decrement key-frame interval
  if(I > Ic)
    I = Ic
  else if(I > 1)
    I = I - 1
   Nh = 0

else if(Frcv > Fhigh)
   Nh = Nh+1

else
   Nh = 0
if(Nh == MIN_DURATION)
  //Increment key-frame interval
  if(I < Ic)
    I = I + 1
  else if(I < Fsnd)
    I = Ftran  //key-frame interval = 1 sec.
     Nh = 0
```

Figure 13. Pseudo code of the adaptive key frame interval adjustment algorithm

Since this mechanism is designed for point-to-point video communication, so the receiver sends the RTCP receiver report to the sender about every 5 seconds. In this packet, it puts value of received frame rate in the RTCP header extension part. The received frame rate is calculated from the correctly-decoded frames only. The corrupted frames and error prediction P-frames are removed, so the received frame rate, put in the receiver report packet, is the worst case value.

When the sender receives a receiver report, it uses the received frame rate in this packet in addition with the value of mean PSNR, got from the encoder, in order to specify the key frame interval based on the pseudo code in Figure 13. Parameters in the pseudo code are described in Table 5. This algorithm decreases key frame interval when the received frame rate is too low ($F_{rcv} < F_{low}$) except when the mean PSNR is too low ($PSNR_{avg} < MIN\_PSNR$). Because it

Table 5. Parameters in adaptive key frame interval adjustment algorithm

Parameters	Description
I	Key frame interval
I_c	Critical value of *I* that can suddenly changed the image quality of video frames
F_{tran}	Transmitted frame rate
F_{rcv}	Received frame rate in the receiver report
F_{low},F_{high}	Thresholds of F_{rcv} used to increase or decrease key frame interval.
$PSNR_{avg}$	Mean PSNR of Y components.
MIN_PSNR	Minimum PSNR of the acceptable image quality
Nh	Number of periods that F_{rcv} is more than F_{high}
MIN_DURATION	Threshold of N_h used to avoid the fluctuation of key frame interval

is useless to try to increase the video frame rate if there are too high distortion to see the content in video frames. When the received frame rate is high enough ($F_{rcv} > F_{high}$) for a specify number of periods (*MIN_DURATION*), this algorithm experiences that it is safe to try to increase the key frame interval in order to get better image quality.

In our proposed algorithm, the key frame interval may not be increased or decreased by one frame per RTCP periods because the changing of key frame interval in some regions has small effect on the image quality of video frames. Figure 14 is the graph of PSNR at different key frame interval resulted from the encoding of 320x240 pixel, 24 fps video when the target bit rate is fixed at 800 kbps. The content in this video scene has low motion which is the general characteristic of the input video in the video telephony or video conferencing environment.

According to this graph, mean PSNR of Y, Cb and Cr components are sharply decreased when key frame interval is lower than 5 frames. We use this value as the critical value

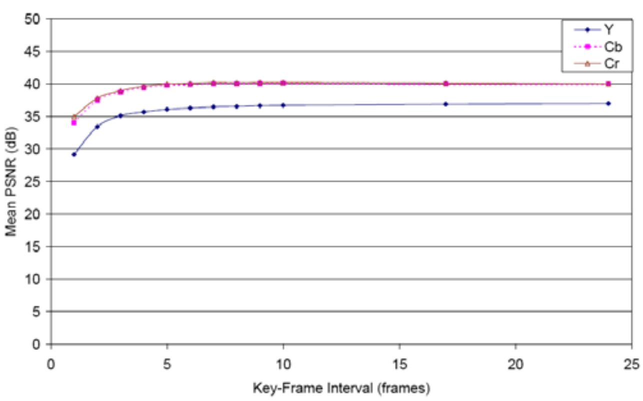

Figure 14. Mean PSNR at different key frame interval

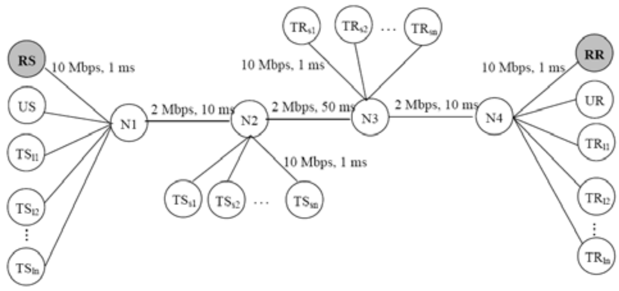

Figure 15. Topology of simulation model

of key frame interval which is I_c in the pseudo code in Figure 13. The value of I_c is not always 5 frames. It is dependent on the content of video frame and the rate control algorithm of the encoder. However, after trying to encode several low motion video sequences using XviD MPEG-4 Codec (XviD, 2005), we found that the value of I_c is about 4-6 frames.

The value of key frame interval is changed by one frame per RTCP period only when it is lower than I_c, otherwise it will jump between I_c and maximum value of key frame interval. We use the value of transmitted frame rate (in fps) as the maximum value of key frame interval. This means that the time of error propagation in the decoder will not be more than a seconds.

6. Evaluation of the proposed posed algorithms

6.1 Evaluation of CNR Algorithm

To evaluate the CNR algorithm and compare the performance with previous adaptive FEC-based algorithms, we setup an experiment and used NS-2 (2005) as a network simulator. All of the previous algorithms use different codec in primary and redundant audio data. But in this experiment, we would like to evaluate CNR algorithm and other algorithms in the case that the redundant and primary codec are G.723.1. There are 6 combinations used in this experiment as shown in Table 4. The HIGH threshold used for all algorithms is 5% because the loss of audio data more than 5% is considered harmful (Jayant and Christensen, 1981). In the case of RCCS algorithm, it is not described clearly about how to find the end-to-end delay. So we assume that all combinations have acceptable end-to-end delay.

In this experiment, we build a simulation model in NS-2. Figure 15 shows the topology of the simulation model. The RS node sends audio traffic over RTP to the RR node. This node selects the combination of audio redundancy based on the algorithm that we adjust. There are other nodes that send TCP and UDP traffic. They act as other users in the same network. In order to see the result in the different network environments, we separate this experiment into 4 parts. The difference between each part is the transmission behavior of the TCP nodes and number of TCP nodes.

1) Simulation Part 1

In this part, all TCP nodes transmit traffic with long period. The configuration of this part is as follows:

- RS transmits audio to RR over RTP and applies the adaptive error control algorithm.
- US sends UDP traffic to UR with bit rate 400 kbps, packet size 1024 bytes. US acts as a non-adaptive sender in the network.
- TS_{l1}-TS_{ln} and TS_{s1}-TS_{sn} send FTP/TCP traffic to TR_{l1}-TR_{ln} and TR_{s1}-TR_{sn} respectively.
- All TCP nodes start transmission at 100 seconds.
- Stop simulation at 1200 seconds.
- There are two rounds of simulating each algorithm.
  - 20 TCP Senders (TS_{l1}-TS_{l10} and TS_{s1}-TS_{s10}).
  - 40 TCP Senders Nodes (TS_{l1}-TS_{l20} and TS_{s1}-TS_{s20}).

Simulation Result

The experimental result consists of 3 graphs, Figure 16 shows the number periods that L_a > HIGH threshold compared among each algorithm, Figure 17 is the graph of average L_a and Figure 18 shows average transmission bit rate of each algorithm.

2) Simulation Part 2

In this part, RS and US have the same behavior as

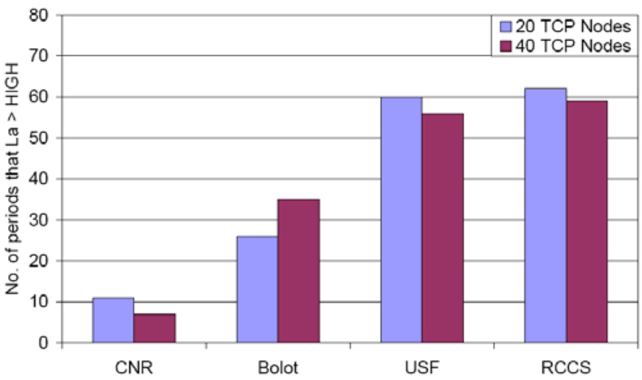

Figure 16. Number of periods that L_a over HIGH threshold (part 1)

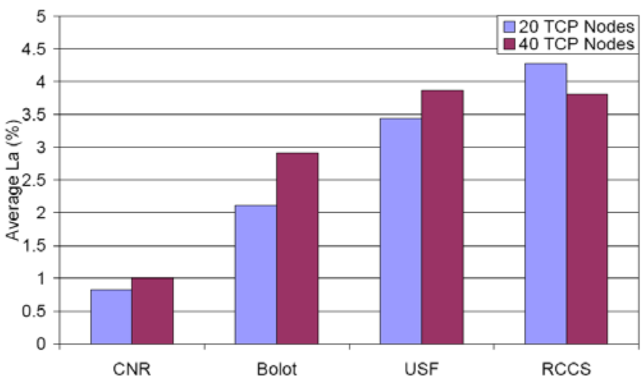

Figure 17. Average L_a (part 1)

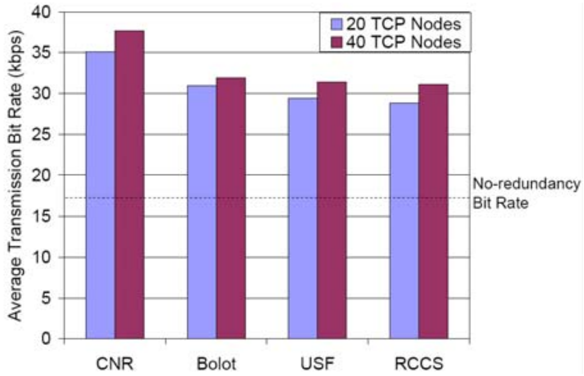

Figure 18. Average bit rate (part 1)

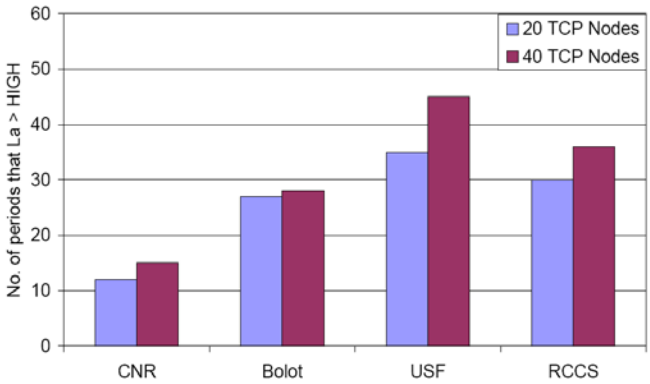

Figure 19. Number of periods that L_a over HIGH threshold (part 2)

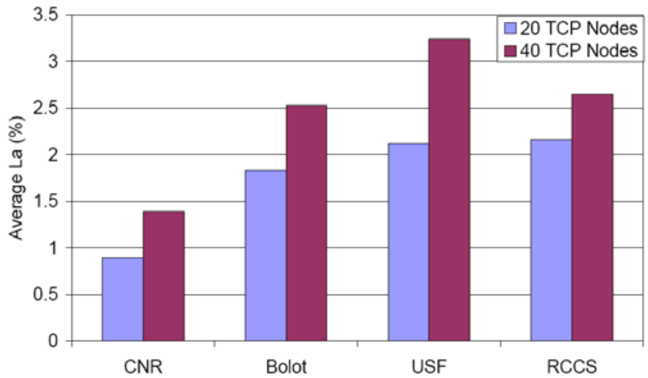

Figure 20. Average L_a (part 2)

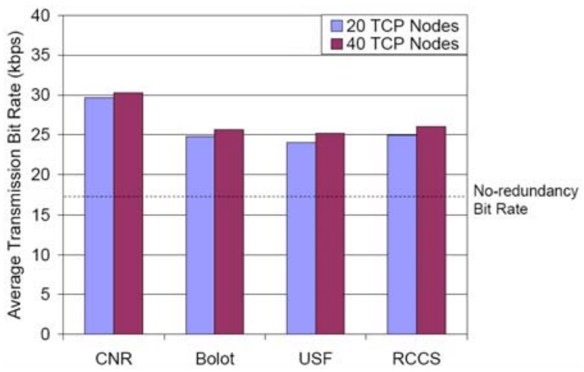

Figure 21. Average bit rate (part 2)

part 1. The difference is in the TCP nodes. The configuration of this part is as follows:

- TS_{l1}-TS_{ln} and TS_{s1}-TS_{sn} start and stop transmission every 200 seconds. We would like to see the result in the different network characteristic from part 1. In this part, the level of network congestion is not constant.
- Other configurations are same as part 1.

Simulation Result

The simulation result of this part is shown in the graph in Figure19, Figure 20 and Figure 21

3) Simulation Part 3

The configuration of this part is as follows:

- The parking-lot nodes (TS_{s1}-TS_{sn}) do not transmit.
- TS_{l1}-TS_{ln} start transmission at 100 seconds and stop at the end of simulation time (1200 seconds).
- All other configurations are the same as in Part 1.

Simulation Result

The simulation result of this part is shown in the graph in Figure 22, Figure 23 and Figure 24.

4) Simulation Part 4

The configuration of this part is as follow:

- The parking-lot nodes (TS_{s1}-TS_{sn}) do not transmit.
- TS_{l1}-TS_{ln} start and stop transmission every 200 seconds.
- All other configurations are the same as in Part 2.

Simulation Result

The simulation result of this part is shown in the graph in Figure 25, Figure 26 and Figure 27.

5) Summary of Simulation Results

The simulation results of CNR algorithm from 4 parts are summarised in Table 6. The result shows that CNR algorithm is the best in recovery the loss of audio data (low loss).

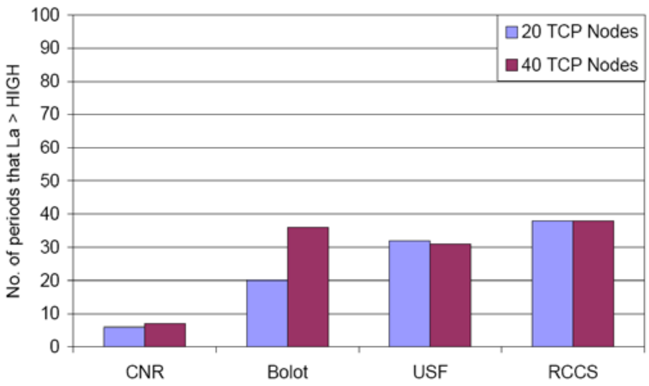

Figure 22. Number of periods that L_a over HIGH threshold (part 3)

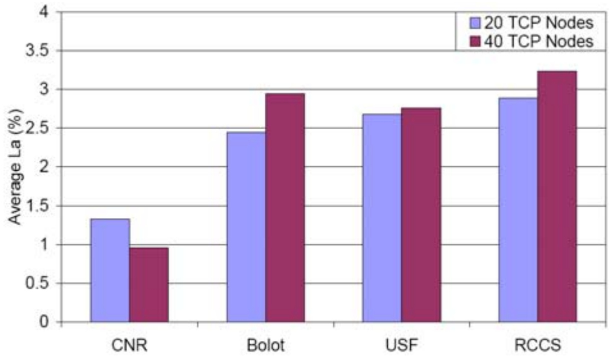

Figure 23. Average L_a (part 3)

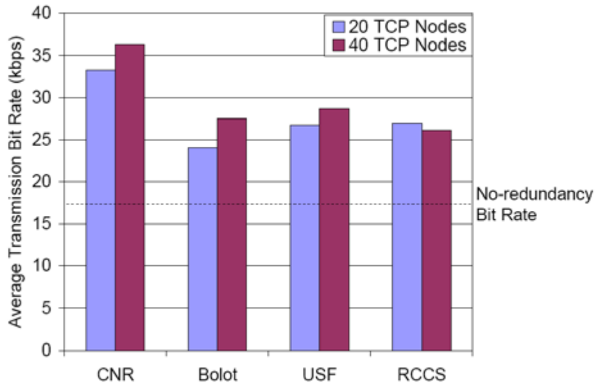

Figure 24. Average bit rate (part 3)

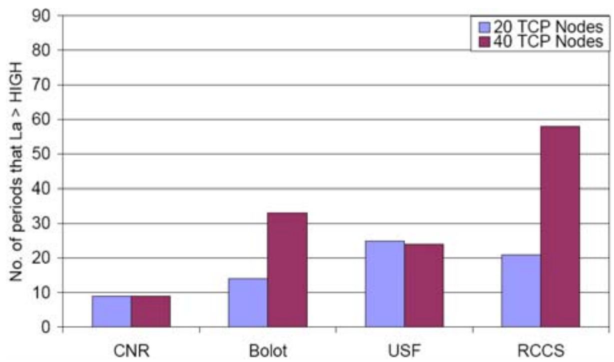

Figure 25. Number of periods that L_a over HIGH threshold (part 4)

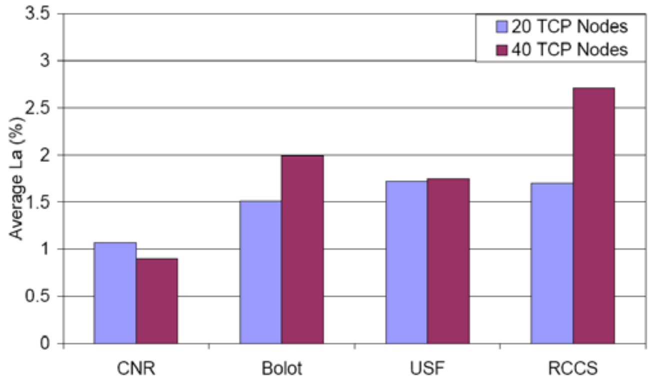

Figure 26. Average L_a (part 4)

Table 6. Summary of experimental result of CNR algorithm

Part	No. of periods that L_a > HIGH	Avg. L_a	Candidate Algorithm	R_{CNR} - $R_{candidate}$	
				20 TCP Senders	40 TCP Senders
1	Lowest	Lowest	Bolot	4.20	5.69
2	Lowest	Lowest	Bolot	4.91	4.62
3	Lowest	Lowest	Bolot	9.14	8.86
4	Lowest	Lowest	Bolot	5.14	5.63

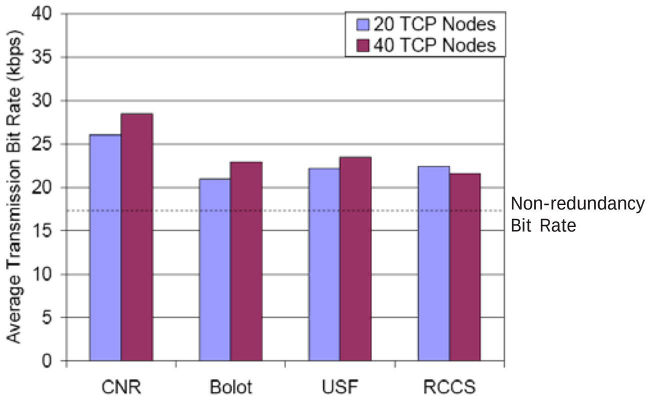

Figure 27. Average bit rate (part 4)

However, it has the average bit rate a little higher than other algorithms because CNR algorithm decreases combination slower than others. Fortunately, it is a reasonable figure because the slow decrement combination is used for avoid the oscillation of L_a.

6.2 Evaluation of Rate Adaptation Techniques

In this section, we will evaluate one of the rate adaptation techniques. We use AIMD (Busse *et al.*, 1996) as the case study. We evaluate this technique by the simulation in NS-2. The network topology of the simulation model is shown in Figure 28. RS transmits video packets to node RR with AIMD algorithm. We use the video trace from the encoding of the same video sequence as used in the graph in Figure 14, frame size 320x240 pixels, frame rate 24 fps and target bit rate is 800 kbps. TS_{l1} - TS_{l00} and TS_{s1} - TS_{s100} transfer file to TR_{l1} - TR_{l100} and TR_{s1} - TR_{s100} over FTP/TCP. This TCP traffic acts as the background traffic from other users in the network. We use TCP Reno in this simulation and set packet size to be 1024 bytes. The bit rate of TCP connections is varied based on its congestion control algorithm. US transmit UDP packets to UR with bit rate 400 kbps, packet size 1024 bytes.

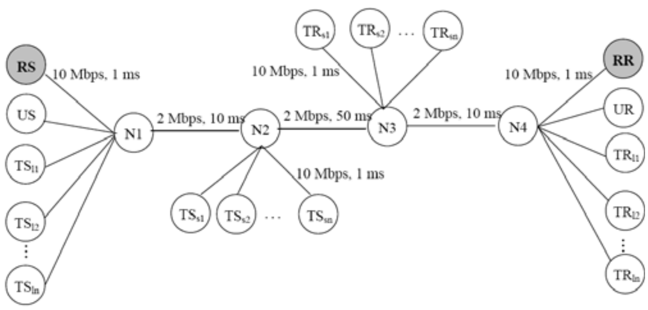

Figure 28. Topology of the simulation model

We run the simulation for 600 seconds and each node starts and stops transmission at the following time:

- RS and US start transmission at the beginning of the simulation.
- All senders stop transmission at 600 seconds.
- Value of AIMD parameters are specified as follow, $\alpha = 0.3$, $\mu = 0.875$, $\nu = 50$ kbps, $\lambda_c = 4\%$ and $\lambda_u = 2\%$. These values are suggested in the paper of Busse (Busse *et al.*, 1996). We set $b_{min} = 200$ kbps and $b_{max} = 2000$ kbps.
- AIMD algorithm focuses on setting value of bit rate only. But when we use MPEG-4 as the video codec, we must specify value of key frame interval. We set it to be 1 frame.

According the graph in Figure 29 that shows the changing of video bit rate and packet loss rate during the simulation. The adaptation of bit rate starts at 10 seconds because TCP senders start transmission. However, the decreasing of video bit rate never make the packet loss rate to be lower than 4% (λ_c). And AIMD stop reducing bit rate at the minimum value (200 kbps).

According to the graph in Figure 30, received frame rate is acceptable because it is always higher than 15 fps. But this is resulted from setting key frame interval to be 1 frame. So the video transmission is robust from packet loss. If we use other values, the experimental result will be different.

The graph in Figure 31 shows the received frame rate at difference key frame interval. When we set key frame interval to be 4 frames, most of received frame rates are lower than 15 fps. And it will be also reduced when we set key frame interval to be 24 fps. So, bit rate adaptation is not enough. The adaptive quality control should mention key frame interval.

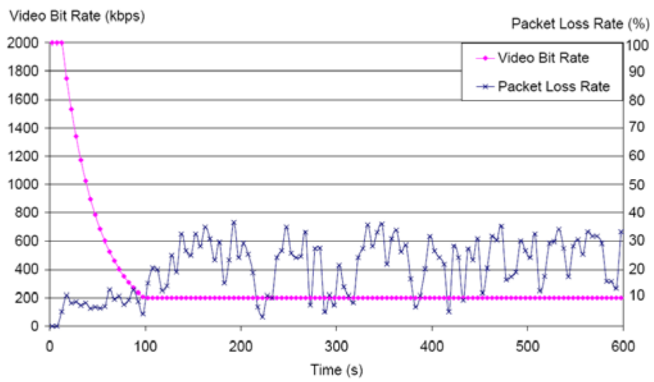

Figure 29. Video bit rate and packet loss rate

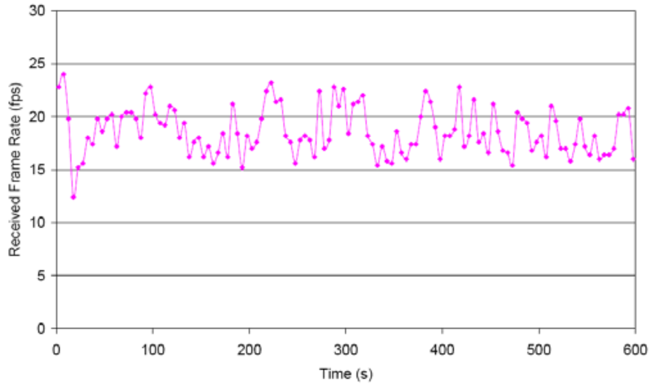

Figure 30. Received frame rate (key frame interval = 1 frame)

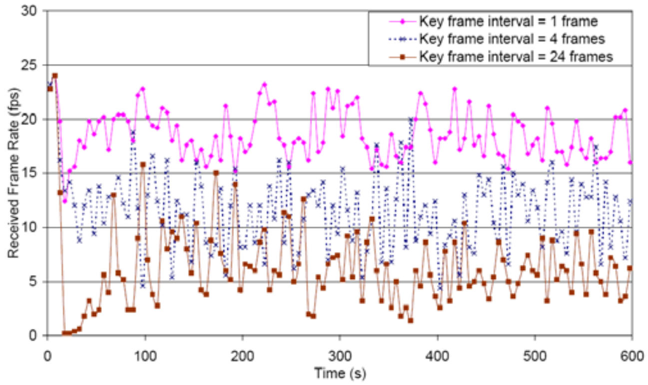

Figure 31. Received frame rate at different key frame interval

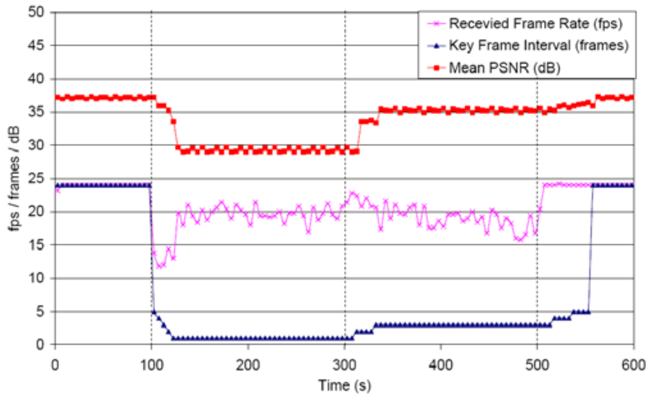

Figure 32. The adaptation of key frame interval and received frame rate

6.3 Evaluation of Adaptive Key Frame Interval Adjustment

1) Simulation Setup

To evaluate the key frame interval adjustment algorithm that we propose, we setup the experiment by using simulation in NS-2 and the topology of the simulation model is shown in Figure 28. RS transmits video packets to node RR with adaptive keyframe interval adjustment algorithm. We use the video trace from the encoding of the same video sequence as used in the graph in Figure 14, frame size 320x 240 pixels, frame rate 24 fps and target bit rate is 800 kbps. TS_{l1} - TS_{l20} and TS_{s1} - TS_{s20} transfer file to TR_{l1} - TR_{l20} and TR_{s1} - TR_{s20} over FTP/TCP. This TCP traffic acts as the background traffic from other users in the network. We use TCP Reno in this simulation and set packet size to be on 1024 bytes. The bit rate of TCP connections is varied based its congestion control algorithm. We run the simulation for 600 seconds and each node starts and stops transmission at the following times:

- RS and US start transmission at the beginning of simulation and stops at time 600 seconds.
- All TCP nodes (TS_{l1} - TS_{l20} and TS_{s1} - TS_{s20}) start transmission at time 100 seconds.
- TS_{l1} - TS_{l10} and TS_{s1} - TS_{s10} stop transmission at time 300 seconds.
- TS_{l11} - TS_{l20} and TS_{s11} - TS_{s20} stop transmission at time 500 seconds.

So there is no TCP traffic during 0-100 seconds and 500-600 seconds. There are 20 nodes that transmit TCP traffic during 100-300 seconds and 10 nodes during 300-500 seconds. So the loss rate of video in each time is varied and we can see the adaptation of key frame interval.

We set parameters in adaptive key-frame interval adjustment algorithm as follows. We set I_c to be 5 frames as described in the previous section. The value of F_{tran} is 24 fps, which is the frame rate of film production. We set F_{low} to be 15 fps which is widely-used frame rate of the video stream-ing in the Internet (Benedetti, 2005). Video seems to be not smooth when frame rate is less than 15 fps. F_{high} is set to 20 fps, which is safety enough to try to increase key frame interval and there is not much different smoothness between 20 fps and 24 fps video. MIN_PSNR is 30 dB because in general, a PSNR of at least 30 dB provides an acceptable quality image (Chen *et al.*, 1998), (Zelinski *et al.*, 2004), (Ramac and Varshney, 2000), (Karayiannis and Li, 2001).

And the value of MIN_DURATION is set to be 4 periods which is equivalent to 20 seconds. The objective of this experiment is to evaluate the adaptive key frame interval adjustment algorithm whether it can control quality of the received video.

2) Result and Discussion

According to the simulation results in the previous section, the result that we would like to see is the adaptation key frame interval and the received video frame rate, which is the graph in Figure 32. According to this graph, in the first 100 seconds, the transmitted and received video have the same frame rate because there are no nodes transmitting TCP traffic. And key frame interval is still 24 frames. From time 100-300 seconds, which has 40 TCP transmitting TCP traffic, key frame interval is decreased to be 1 frame. And it can keep the value of received frame rate not be below 15 fps. During the number of nodes transmitting TCP traffic is decreased to be 20 nodes and the network congestion is reduced. So the key frame interval is increased to be 3-4 frames. And the last 100 seconds, there is no nodes transmitting TCP traffic again, so the key frame interval is increased and stop at 24 frames.

When we consider the value of mean PSNR, which indicate the image quality of video frames, mean PSNR is often over 30dB (MIN_PSNR) except during time 170-320 seconds. During that time, mean PSNR is around 29 dB. But it does not influence the decision to decrement key frame interval because the value of key frame interval during time is 1 frame that is the lowest value.

7. Conclusion

In this paper, we present QoS (Quality of Service) adaptations to support multimedia communications. Most of the rate adaptation techniques in previous works, assumed that reducing video bit rate can reduce network congestion and the loss rate of video packets will be reduced too. Unfortunately this assumption may not be correct in some situations, especially in the almost fully utilized network. Besides, the packet loss rate does not actually reflect the video quality seen by the viewer. Moreover previous works on adaptive FEC-based (Forward Error Correction) algorithms also have some drawbacks. We have proposed new adaptive error control algorithm, called CNR algorithm (CNR: Centre for Network Research, where this work has been done). We have shown that, through various simulation scenarios, CNR algorithm removes the drawbacks of those previous algo-rithms, for example, increase the system throughput and reduce the rate oscillation. Moreover, CNR algorithm decreases conditions of the combination when low-loss period is big enough. In addition, we have proposed the key frame interval adjustment algorithm to gain a better traffic throughput. However, a small disadvantage of CNR is it requires a small higher bandwidth than others, but it is in a marginally acceptable figure.

Acknowledgement

This work was supported by NECTEC, Project ID: NT-B-22-T3-19-48-06

References

Benedetti, B. Video Basics. http://www.pierce.ctc.edu/bbenedet/digvideo/Documents/Videobasics/videobasics.htm [October 23, 2005]

Bolot, J. C. and Garcia, A.V., 1996. Control mechanisms for packet audio in the Internet. Proceedings of IEEE INFOCOM '96, vol. 1, San Francisco, USA, March 1996, 232-239.

Bolot, J. C. and Turletti, T. 1996. Adaptive error control for packet video in the Internet. Proceedings of IEEE ICIP '96, vol. 1, Lausanne, Switzerland, September 1996, 25-28.

Busse, I., Deffner, B. and Schulzrinne, H. 1996. Dynamic QoS control of multimedia applications based on RTP. Computer Communications, 9(1): 49-58.

Chen, T. S., Chang, C. C; and Hwang, M.S. 1998. A virtual image cryptosystem based upon vector quantization. IEEE Transactions on Image Processing, 7(10): 1485-1488.

Digital Video for the Web, http://stream.uen.org/medsol/digvid/html/3B_videoframerate.html[October 23, 2005]

http://www.matrox.tv/includes/pdfs/tutorial2.pdf [October 23, 2005]

ISO/IEC 14496-2. 2001. Coding of audio-visual objects - Part 2: Visual.

Jayant, N. and Christensen, S. W. 1981. Effect of Packet Losses in Waveform Coded Speech and Improvements due to an Odd-Even Sample-Interpolation Procedure. IEEE Transactions on Communications, COM-29(2): 101-109,

Ji M. K., Lee, S. H.; Choi, T. U.; Park, S. H.; Kang, J. G. ; and Chung, K.D. 2001. Selecting an audio redundancy codec combination for error control in Internet telephony. Proceedings of IEEE ICPADS 2001, Kyongju, South Korea, June 2001, 645–649.

Karayiannis, N. B. and Li, Y. 2001. A replenishment technique for low bit-rate video compression basedon wavelets and vector quantization. IEEE Transactions on Circuits and Systems for Video Technology, 11(5): 658-663.

Padhye, C., et al. 2000. A new adaptive FEC loss control algorithm for voice over IP applications. Proceedings of IEEE IPCCC 'OO, Phoenix, USA, February 2000, 307-313.

Perkins, C. S., et al. 1997. RTP Payload for redundant audio data, RFC2198, September 1997.

Ramac, L. C. and Varshney, P. K. Y. 2000. A wavelet domain diversity method for transmission of images over-wireless channels. IEEE Journal on Selected Areas in Communications, 18(6): 891-898.

Richardson, I. E. G. 2003. H.264 and MPEG-4: Video Compression Video Coding for Next-generation Multimedia, John Wiley & Sons, England.

Sisalem, D. and Schulzrinne, H. 1998. The loss-delay based adjustment algorithm: A TCP-friendly adaptation scheme. Proceedings of International Workshop on Network and Operating Systems Support for Digital Audio and Video, Cambridge, England, July 1998, 215-226.

Sisalem, D. and Schulzrinne, H. 2000. The Direct Adjustment Algorithm: a TCP-friendly adaptation scheme. Proceedings of International Workshop on Quality of Future Internet Services, Berlin, Germany, September 2000, 68-79.

The Network Simulator, NS-2. http://www.isi.edu /nsnam/ns/ [October 23, 2005]

Turletti, T. and Huitema, C. 1996. Videoconferencing in the Internet. IEEE/ACM Transactions on Networking Journal. 4(3):340-351.

UBC/LBNL Video Conferencing Tool (vic). http://www-nrg.ee.lbl.gov/vic/ [October 23, 2005]

XviD. http://www.xvid.org [October 23, 2005]

Zelinski, A. C.; *et al.* 2004. Automatic cost minimization for multiplierless implementations of discrete signal transforms. Proceedings of IEEE ICASSP'04, 5(May) : V-221-V-224.