

Entropy Coding-based Lossless Compression of Asynchronous Event Sequences —Supplementary Material—

Ionut Schiopu and Radu Ciprian Bilcu
Huawei Technologies Oy (Finland) Co. Ltd, Tampere Handset Camera Innovation Lab

Algorithm E1: Modified TTP_e Encoder of N_e

Input: $N_e; \hat{N}_e; e = (1; 2);$

```

1 if  $N_e > 0$  then
2   Encode  $e_n = 0$  using  $DTeb_0$ ; Update  $DTeb_0$ ;
3    $e = N_e - \hat{N}_e; e = 1 + 2;$ 
4   if  $j_{ej} < 1$  then // R1 Range
5     Encode  $e_n = 0$  using  $DTeb_1$ ; Update  $DTeb_1$ ;
6     Encode  $e_n = j_{ej}$  using  $Ee_1$ ; Update  $Ee_1$ ;
7   else
8     Encode  $e_n = 1$  using  $DTeb_1$ ; Update  $DTeb_1$ ;
9     if  $j < e - 1$  then // R2 Range
10      Encode  $j_{ej} - 1$  using  $Ee_2$ ; Update  $Ee_2$ ;
11    else // R6 Range
12      Encode  $e - 1$  using  $Ee_2$ ; Update  $Ee_2$ ;
13      Encode  $x = j_{ej} - e - 2$  using Alg. E2;
14    Encode  $e_n = \text{sgn}(j_{ej})$  using  $DTeb_5$ ; Update  $DTeb_5$ ;
15 else
16   Encode  $e_n = 1$  using  $DTeb_0$ ; Update  $DTeb_0$ ;

```

Algorithm E2: Modified EGC Encoder of x

Input: x

```

1 if  $x = 1$  then
2   Encode  $x_n = 1$  using  $BR4$ ; Update  $BR4$  using (1);
3 else
4    $N = \text{blog}_2(x) + 1; // (x)_{(10)} = 1b_{N-1} \dots b_1b_0$ 
5   for  $i = 1; 2; \dots; N$  do // Unary rep. of  $N$ 
6     Encode  $x_n = 0$  using  $BR4$ ; Update  $BR4$ ;
7   Encode  $x_n = 1$  using  $BR4$ ; Update  $BR4$ ;
8   Binarize  $(x)_{(10)}$  as  $1b_{N-1} \dots b_1b_0$ ;
9   for  $i = N - 1; N - 2; \dots; 0$  do // Last  $N$  digits
10    Encode  $x_n = b_i$  using  $BR5$ ; Update  $BR5$ ;

```

Algorithm E3: Modified TTP_y Encoder of y

Input: $y; \hat{y}; [1; W]; e = (1; 2; 3);$

```

1  $y = y - \hat{y}; y = 1 + 2 + 3;$ 
2 if  $j_{yj} < y$  then
3   Encode  $y_n = 0$  using  $DTyb_0$ ; Update  $DTyb_0$ ;
4   if  $j_{yj} < 1$  then // R1 Range
5     Encode  $y_n = 0$  using  $DTyb_1$ ; Update  $DTyb_1$ ;
6     Encode  $y_n = j_{yj}$  using  $Ey_1$ ; Update  $Ey_1$ ;
7   else
8     Encode  $y_n = 1$  using  $DTyb_1$ ; Update  $DTyb_1$ ;
9     if  $j < 1 + 2$  then // R2 Range
10      Encode  $y_n = 0$  using  $DTyb_2$ ; Update  $DTyb_2$ ;
11      Encode  $j_{yj} - 1$  using  $Ey_2$ ; Update  $Ey_2$ ;
12    else // R3 Range
13      Encode  $y_n = 1$  using  $DTyb_2$ ; Update  $DTyb_2$ ;
14      Encode  $j_{yj} - 1 - 2$  using  $Ey_3$ ; Update  $Ey_3$ ;
15 else // R5 Range
16   Encode  $y_n = 1$  using  $DTyb_0$ ; Update  $DTyb_0$ ;
17    $y_2 = y - y; n_2 = \text{dlog}_2(W - y_2 + 1) + e;$ 
18   Binarize  $(W - y)_{(10)}$  as  $b_{n_2-1} \dots b_1b_0$ ;
19   for  $i = 0; 1; \dots; n_2 - 2$  do
20     Encode  $y_{n+i} = b_i$  using  $ByL$ ; Update  $ByL$ ;
21   if  $b_{n_2-1} \dots b_1b_0 = W - y_2 + 1 - 2^{n_2-1}$  then
22     Encode  $x_n = b_{n_2-1}$  using  $BR5$ ; Update  $BR5$ ;

```

Algorithm E4: Modified TTP_x Encoder of x

```

Input:  $x; \hat{x}; c_x$  (to signal sorted  $x$ );  $[1; H]; \quad = (1; 2; 3);$ 
1  $x = X \quad \hat{x}; \quad x = 1 + 2 + 3;$ 
2 if  $j_{xj} < x$  then
3   Encode  $x_n = 0$  using  $DTxb_0$ ; Update  $DTxb_0$ ;
4   if  $j_{xj} < 1$  then // R1 Range
5     Encode  $x_n = 0$  using  $DTxb_1$ ; Update  $DTxb_1$ ;
6     Encode  $x_n = j_{xj}$  using  $Ex_1$ ; Update  $Ex_1$ ;
7   else
8     Encode  $x_n = 1$  using  $DTxb_1$ ; Update  $DTxb_1$ ;
9     if  $j_{xj} < 1 + 2$  then // R2 Range
10      Encode  $x_n = 0$  using  $DTxb_2$ ; Update  $DTxb_2$ ;
11      Encode  $j_{xj} - 1$  using  $Ex_2$ ; Update  $Ex_2$ ;
12    else // R3 Range
13      Encode  $x_n = 1$  using  $DTxb_2$ ; Update  $DTxb_2$ ;
14      Encode  $j_{xj} - 1 - 2$  using  $Ex_3$ ; Update  $Ex_3$ ;
15  if  $c_x$  then // Was  $x$  sorted?
16    Encode  $sgn(j_{xj})$  using  $DTxb_s$ ; Update  $DTxb_s$ ;
17 else
18   Encode  $x_n = 1$  using  $DTxb_0$ ; Update  $DTxb_0$ ;
19    $x_1 = \hat{x} \quad x; n_1 = \lceil \log_2 x_1 \rceil;$ 
20    $x_2 = \hat{x} + x; n_2 = \lceil \log_2 (H - x_2 + 1) \rceil;$ 
21   if  $x_1 < 1$  then // Deterministic case R5
22     Binarize  $(H - x)_{(10)}$  as  $\overline{b_{n_2-1} \dots b_1 b_0}_{(2)}$ ;
23     for  $i = 0; 1; \dots; n_2 - 2$  do
24       Encode  $x_{n+i} = b_i$  using  $BxL$ ; Update  $BxL$ ;
25     if  $\overline{b_{n_2-1} \dots b_1 b_0} = H - x_2 + 1 - 2^{n_2-1}$  then
26       Encode  $x_n = b_{n_2-1}$  using  $BR5$ ; Update  $BR5$ ;
27   else if  $x_2 > H$  then // Deterministic case R4
28     Binarize  $(x - 1)_{(10)}$  as  $\overline{b_{n_1-1} \dots b_1 b_0}_{(2)}$ ;
29     for  $i = 0; 1; \dots; n_1 - 2$  do
30       Encode  $x_{n+i} = b_i$  using  $BxL$ ; Update  $BxL$ ;
31     if  $\overline{b_{n_1-1} \dots b_1 b_0} = x_1 - 2^{n_1-1}$  then
32       Encode  $x_n = b_{n_1-1}$  using  $BR4$ ; Update  $BR4$ ;
33   else
34     if  $x = \hat{x} \quad x;$  then // R4 Range
35       Encode  $x_n = 0$  using  $DTxb_1^d$ ; Update  $DTxb_1^d$ ;
36       Binarize  $(x - 1)_{(10)}$  as  $\overline{b_{n_1-1} \dots b_1 b_0}_{(2)}$ ;
37       for  $i = 0; 1; \dots; n_1 - 2$  do
38         Encode  $b_i$  using  $BxL$ ; Update  $BxL$ ;
39       if  $\overline{b_{n_1-1} \dots b_1 b_0} = x_1 - 2^{n_1-1}$  then
40         Encode  $b_{n_1-1}$  using  $BR4$ ; Update  $BR4$ ;
41     else // R5 Range
42       Encode  $x_n = 1$  using  $DTxb_1^d$ ; Update  $DTxb_1^d$ ;
43       Binarize  $(H - x)_{(10)}$  as  $\overline{b_{n_2-1} \dots b_1 b_0}_{(2)}$ ;
44       for  $i = 0; 1; \dots; n_2 - 2$  do
45         Encode  $b_i$  using  $BxL$ ; Update  $BxL$ ;
46       if  $\overline{b_{n_2-1} \dots b_1 b_0} = H - x_2 + 1 - 2^{n_2-1}$  then
47         Encode  $b_{n_2-1}$  using  $BR5$ ; Update  $BR5$ ;

```

Algorithm D1: Modified TTP_x Decoder of x

```

Input:  $\hat{x}; [1; H]; c_x; \quad = (1; 2; 3);$ 
Output:  $x;$ 
1  $b_0$  Decode using  $DTxb_0$ ; Update  $DTxb_0$ ;
2 if  $b_0 = 0$  then
3    $b_1$  Decode using  $DTxb_1$ ; Update  $DTxb_1$ ;
4   if  $b_1 = 0$  then  $x$  Decode using  $Ex_1$ ; Update  $Ex_1$ ;
5   else
6      $b_2$  Decode using  $DTxb_2$ ; Update  $DTxb_2$ ;
7     if  $b_2 = 0$  then // R2 Range
8        $x = 1 +$  Decode using  $Ex_2$ ; Update  $Ex_2$ ;
9     else  $x = 1 + 2 +$  Decode using  $Ex_3$ ; Update  $Ex_3$ ;
10     $b_s = 1;$ 
11    if  $c_x$  then  $b_s$  Decode using  $DTxb_s$ ; Update  $DTxb_s$ ;
12     $x = (b_s = 0) ? (\hat{x} - x) : (\hat{x} + x);$ 
13 else
14    $x = 0; x_1 = \hat{x} \quad 1 \quad 2 \quad 3; x_2 = \hat{x} + 1 + 2 + 3;$ 
15    $n_1 = \lceil \log_2 x_1 \rceil; n_2 = \lceil \log_2 (H - x_2 + 1) \rceil;$ 
16   if  $x_1 < 1$  then // Deterministic case R5
17     for  $i = 0; 1; \dots; n_2 - 2$  do
18        $b$  Decode using  $BxL$ ; Update  $BxL$ ;
19       if  $b = 1$  then  $x = (1 - i)jx$ ;
20     if  $x = H - x_2 + 1 - 2^{n_2-1}$  then
21        $b$  Decode using  $BR5$ ; Update  $BR5$ ;
22       if  $b = 1$  then  $x = (1 - (n_2 - 1))jx$ ;
23   else if  $x_2 > H$  then // Deterministic case R4
24     for  $i = 0; 1; \dots; n_1 - 2$  do
25        $b$  Decode using  $BxL$ ; Update  $BxL$ ;
26       if  $b = 1$  then  $x = (1 - i)jx$ ;
27     if  $x = x_1 - 2^{n_1-1}$  then
28        $b$  Decode using  $BR4$ ; Update  $BR4$ ;
29       if  $b = 1$  then  $x = (1 - (n_1 - 1))jx$ ;
30   else
31      $b_1^d$  Decode using  $DTxb_1^d$ ; Update  $DTxb_1^d$ ;
32     if  $b_1^d = 0$  then // R4 Range
33       for  $i = 0; 1; \dots; n_1 - 2$  do
34          $b$  Decode using  $BxL$ ; Update  $BxL$ ;
35         if  $b = 1$  then  $x = (1 - i)jx$ ;
36       if  $x = x_1 - 2^{n_1-1}$  then
37          $b$  Decode using  $BR4$ ; Update  $BR4$ ;
38         if  $b = 1$  then  $x = (1 - (n_1 - 1))jx$ ;
39     else // R5 Range
40       for  $i = 0; 1; \dots; n_2 - 2$  do
41          $b$  Decode using  $BxL$ ; Update  $BxL$ ;
42         if  $b = 1$  then  $x = (1 - i)jx$ ;
43       if  $x = H - x_2 + 1 - 2^{n_2-1}$  then
44          $b$  Decode using  $BR5$ ; Update  $BR5$ ;
45         if  $b = 1$  then  $x = (1 - (n_2 - 1))jx$ ;
46 Return  $x;$ 

```

Algorithm D2: Modified TTP_e Decoder of N_e

Input: $\hat{N}_e; \quad = (1; 2);$
Output: $N_e;$

```

1  $b_0$  Decode using  $DTebo$ ; Update  $DTebo$ ;
2 if  $b_0 = 0$  then
3    $b_1$  Decode using  $DTe b_1$ ; Update  $DTe b_1$ ;
4   if  $b_1 = 0$  then  $e$  Decode using  $Ee_1$ ; Update  $Ee_1$ ;
5   else
6      $e$  Decode using  $DTe b_2$ ; Update  $DTe b_2$ ;
7     if  $e = 2 - 1$  then
8        $e = 1 + 2 - 2$  + Decode using Alg. 3;
9     else  $e = 1 + e$ ;
10   $b_s$  Decode using  $DTe b_s$ ; Update  $DTe b_s$ ;
11  Return  $N_e = (b_s = 0) ? (\hat{N}_e - e) : (\hat{N}_e + e);$ 
12 else
13  Return  $N_e = 0$ 

```

Algorithm D3: Modified EGC Decoder of x

Output: x ;

```

1  $x$  Decode using  $BR4$ ; Update  $BR4$ ;
2 if  $x = 0$  then
3    $N = 1$ ;  $x$  Decode using  $BR4$ ; Update  $BR4$ ;
4   while  $x = 0$  do
5      $N = N + 1$ ;  $x$  Decode using  $BR4$ ; Update  $BR4$ ;
6   for  $i = N - 1; N - 2; \dots; 0$  do
7      $x = (x - 1) +$  Decode using  $BR5$ ; Update  $BR5$ ;
8 Return  $x$  ;

```

Algorithm D4: Modified TTP_y Decoder of y

Input: $\hat{y}; [1; W]; \quad = (1; 2; 3);$
Output: $y;$

```

1  $b_0$  Decode using  $DTyb_0$ ; Update  $DTyb_0$  using (1);
2 if  $b_0 = 0$  then
3    $b_1$  Decode using  $DTyb_1$ ; Update  $DTyb_1$ ;
4   if  $b_1 = 0$  then // R1 Range
5      $y = \hat{y} +$  Decode using  $Ey_1$ ; Update  $Ey_1$ ;
6   else
7      $b_2$  Decode using  $DTyb_2$ ; Update  $DTyb_2$ ;
8      $y = \hat{y} + 1$ ;
9     if  $b_2 = 0$  then // R2 Range
10       $y = y +$  Decode using  $Ey_2$ ; Update  $Ey_2$ ;
11    else // R3 Range
12       $y = y + 2 +$  Decode using  $Ey_3$ ; Update  $Ey_3$ ;
13 else // R5 Range
14    $y = 0; n_2 = \log_2 (W - \hat{y} - 1 - 2 - 3 + 1)e;$ 
15   for  $i = 0; 1; \dots; n_2 - 2$  do
16      $b$  Decode using  $ByL$ ; Update  $ByL$ ;
17     if  $b = 1$  then  $y = (1 - i)jy$ ;
18   if  $y = W - \hat{y} - 1 - 2 - 3 + 1 - 2^{n_2 - 1}$  then
19      $b$  Decode using  $BR5$ ; Update  $BR5$ ;
20     if  $b = 1$  then  $y = (1 - (n_2 - 1))jy$ ;
21 Return  $y;$ 

```
