I. J. Computer Network and Information Security, **2016, 8, 1-10**
Published Online August 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2016.08.01

Modelling and Implementation of Network Coding for Video

Can Eyupoglu
Istanbul Commerce University, Department of Computer Engineering, Istanbul, 34840, Turkey
E-mail: ceyupoglu@ticaret.edu.tr

Ugur Yesilyurt
Takasbank, Department of Application Development, Istanbul, 34381, Turkey
E-mail: uyesilyurt@takasbank.com.tr

***Abstract*—**In this paper, we investigate Network Coding for Video (NCV) which we apply for video streaming over wireless networks. NCV provides a basis for network coding. We use NCV algorithm to increase throughput and video quality. When designing NCV algorithm, we take the deadline as well as the decodability of the video packet at the receiver. In network coding, different flows of video packets are packed into a single packet at intermediate nodes and forwarded to other nodes over wireless networks. There are many problems that occur during transmission on the wireless channel. Network coding plays an important role in dealing with these problems. We observe the benefits of network coding for throughput increase thanks to applying broadcast operations on wireless networks. The aim of this study is to implement NCV algorithm using C programming language which takes the output of the H.264 video codec generating the video packets. In our experiments, we investigated improvements in terms of video quality and throughput at different scenarios.

***Index Terms*—**Network coding, video streaming, wireless networks.

I. Introduction

Wireless networks suffer from low throughput and cannot provide high quality video because of the irregular and time varying nature of wireless channel [1–4]. In video streaming over wireless, there are many researches related to cross layer design, wireless networking, video compression and streaming [1, 5–7].

Network coding is a new research field and can be used in many applications. Furthermore, it is applied to practical networking systems. In network coding, intermediate nodes may send out packets which are linear combinations of previously received information. There are two main benefits of network coding. These are throughput improvement and high robustness [8]. Moreover, network coding is utilized in wireless networks [9–12], P2P file distribution [13, 14], ad-hoc sensor networks [15], network tomography [16, 17] and network security [18–20].

In order to improve network throughput, Network Coding for Video (NCV) has been proposed and seen as a promising method [21, 22]. The main idea of NCV algorithm is to select the best network code in order for improving video quality. Different scheduling schemes of video packets are generated by NCV algorithm ensuring an increase in throughput. Each video packet has a different contribution in a video signal. The best network code is reconstructed by taking the importance of different chosen packets and their corresponding flows [1, 23].

The rest of the paper is organized as follows. Section II discusses related work. In Section III, Network Coding for Video algorithm is explained. Section IV investigates the simulation results that show the improvements in video quality and throughput. Finally, conclusions being under study are summarized in Section V.

II. Related Work

A wireless mesh network example is shown in Fig. 1. I is an intermediate node. A, B and C are receiving or sending nodes. In video streaming over wireless mesh networks, intermediate nodes can forward packets to other intermediate nodes or clients.

In order to maximize video quality and throughput, many algorithms have been developed and they can be used at intermediate nodes. Network coding operations are performed at intermediate nodes. In addition, intermediate nodes combine packets from several incoming streams into a single outgoing packet. Then, this outgoing packet is broadcasted to the entire neighborhood. Therefore, it reaches lots of nodes concurrently. It is assumed that all nodes can overhear all transmissions in their neighborhood. Besides, they can decode network coded packets using overheard packets [1, 24, 25].

It is observed that when video streams are transmitted, video quality must also be considered in addition to the best choice. In order for selecting the codes which contribute to the video stream quality, the importance and deadlines of video packets must be taken into consideration [1].

There are many terminologies used in network coding. These are shown in Fig. 2.

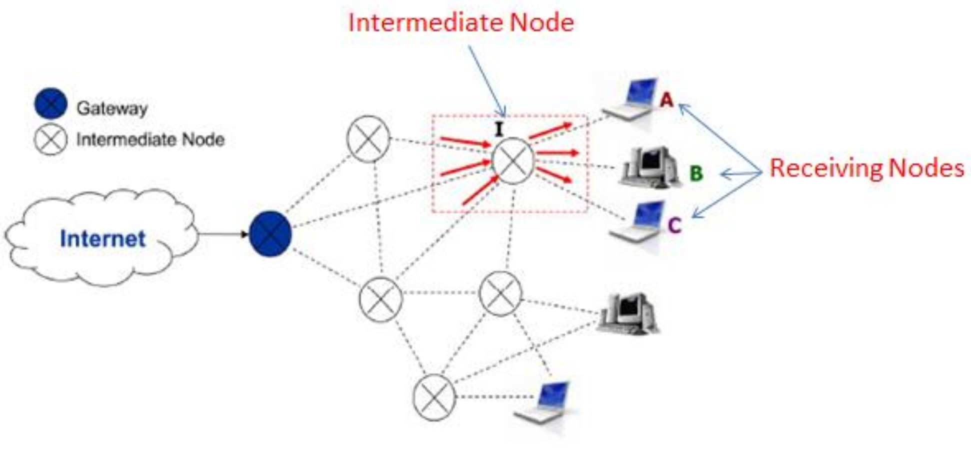


Fig.1. A Wireless Mesh Network Example.

Primary packet is the packet selected from the Tx queue before network coding. It must be included in all network codes. Furthermore, it can be thought as the main packet in a given time-slot. Side packet is the packet in the Tx queue, other than the primary, included in the network code. In network code, one primary and all side packets are XOR-ed together into a single packet. Active packet is the packet in the Tx queue that can be considered as primary. Inactive packet is the packet in the Tx queue that cannot be considered as primary. Target node is the intended recipient of the primary packet. Tx queue is the output queue of the transmitting node. Rx buffer is the receive queue of the receiving node. It stores received packets, destined to this node. Virtual buffer is also maintained at a receiving node. It stores overheard packets, destined to other nodes [1].

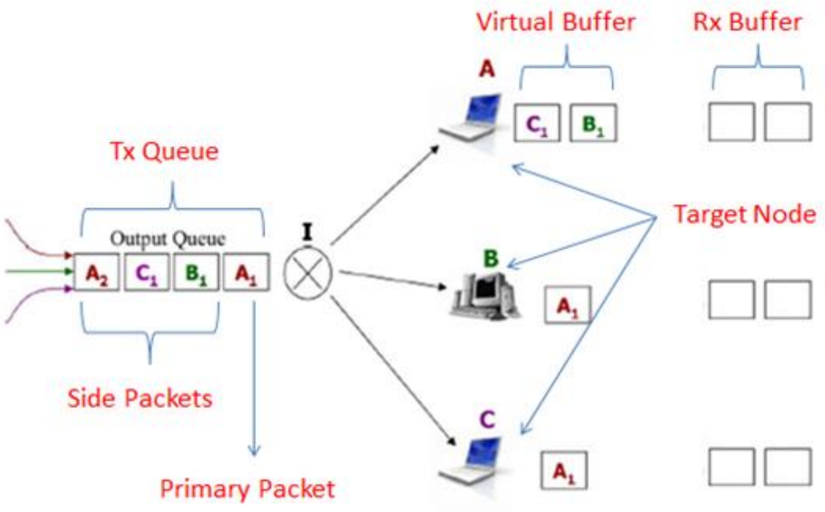


Fig.2. Network Coding Terminologies.

In code selection, it is considered that an intermediate node receives *N* packets from different video streams. Then, it forwards these packets to *N* nodes in its neighborhood. The intermediate node maintains a Tx queue with incoming packets. In order for transmission, a packet is selected from the Tx queue at a given time slot. The selected packet is called the primary packet. In addition, its destination is called the target node. The primary packet is considered as the main packet which is desired to transmit during a time slot. The primary packet might become the first packet or any packet in Tx queue that is marked as active depending on the network coding scheme. Besides, all packets are considered as candidate side packets. The side and primary packets are XOR-ed together into a single packet. This is called the network code. The main point is to choose the best network code in order to maximize the total video quality and throughput [1].

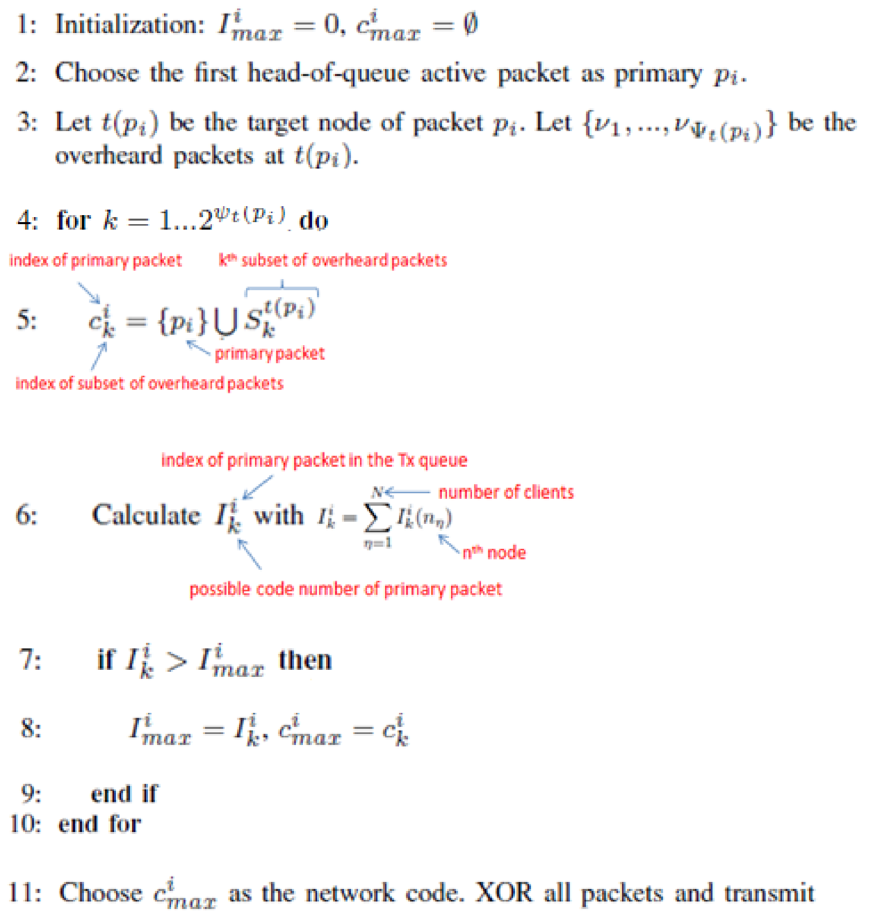
III. Method

A. NCV Algorithm

NCV algorithm is one of the methods developed for improving network throughput. Moreover, it selects the network codes maximizing video quality. NCV guarantees throughput increase thanks to creating various scheduling schemes of video packets. In a video signal, each video packet has a different contribution. Furthermore, taking the importance of different chosen packets and their corresponding flows, the best network code is formed.

In NCV, the packets which come to an intermediate node are held on Tx queue. The queue has FIFO algorithm. The first packet of the queue is the primary packet and others are side packets. The primary packet and side packets are XOR-ed to create possible network codes. The target node is the intended recipient of the primary packet. If the target node can get the network code successfully, it can decode it.

Nodes have two kinds of buffers. One of them is virtual buffer which is for overheard packets. Overheard packets are related to previous transmissions. Other one is Rx buffer which is for received packets. The primary packet which comes to target node is stored in Rx buffer and ACK is sent to the intermediate node. The time between the beginning of transmission and getting an ACK is round trip time (RTT). In this time period, the sent packets are marked inactive in Tx queue. When ACK comes from the related node, the packets which are in network code are erased from Tx queue. If the ACK doesn't come during RTT, the video packets will be active again and the process will be repeated. The packets which are in Tx queue are held until the packets are transmitted or the deadline is missed. Nodes can encode and decode in real time [1]. NCV algorithm is shown in Fig. 3.



```
1: Initialization: $I^i_{max} = 0$, $c^i_{max} = \emptyset$
2: Choose the first head-of-queue active packet as primary $p_i$.
3: Let $t(p_i)$ be the target node of packet $p_i$. Let $\{\nu_1, ..., \nu_{\Psi_t(p_i)}\}$ be the
   overheard packets at $t(p_i)$.
4: for $k = 1...2^{\psi_t(p_i)}$ do
5:     $c^i_k = \{p_i\} \bigcup S^{t(p_i)}_k$
6:     Calculate $I^i_k$ with $I^i_k = \sum_{\eta=1}^{N} I^i_k(n_\eta)$
7:     if $I^i_k > I^i_{max}$ then
8:         $I^i_{max} = I^i_k$, $c^i_{max} = c^i_k$
9:     end if
10: end for
11: Choose $c^i_{max}$ as the network code. XOR all packets and transmit
```

index of primary packet
kth subset of overheard packets
primary packet
index of subset of overheard packets
index of primary packet in the Tx queue
number of clients
nth node
possible code number of primary packet

Fig.3. NCV Algorithm [1].

• In the first step of the algorithm, we initialize the maximum improvement to zero and the best network code to empty set. In the beginning, there is no calculated value.

• Second, we choose the first active packet in Tx queue at the intermediate node as a primary packet.

• Third, we find the target node of the chosen primary packet and create a subset of overheard packets in the target node.

• Then, *k* is the number of possible network codes. The maximum value of *k* is denoted by 2 over number of overheard packets.

• The chosen primary packet and the subset of overheard packets are combined and thus possible network code is formed.

• For every possible network code, the improvement at each client is calculated and summed up.

• In 7th step, we compare the new improvement value and the previous one.

• This comparison continuous until finding the maximum improvement. Then, we choose the best network code that provides the maximum improvement.

• At the end of the algorithm, we find the best network code and then all packets in this code are XOR-ed and transmitted to clients.

The step 6.1 and 6.1.1 of NCV algorithm are shown in Fig. 4 and 5, respectively.

$$\underset{\text{index of network code}}{\overset{\text{index of primary packet}}{I_k^i}}(n_\eta) = \sum_{\underset{\text{packet}}{l=1}}^{\overset{\text{The number of original packets in network code}}{L_k}} \underbrace{(1-P(l))}_{\text{the probability that a packet is not lost}} \overbrace{\Delta(l)}^{\text{improvement in video quality}} \underbrace{\gamma(l)}_{\text{importance of flow}} \overbrace{g_l^k(n_\eta)}^{\text{targeted node or not}} \underbrace{d_l^k(n_\eta)}_{\text{decodability at target node}}$$

Fig.4. The Step 6.1 of NCV Algorithm.

• According to the step 6.1, first, the improvement of each code is calculated to find the best network code. To calculate the improvement of each code, the lost probability of packets, distortion value of packets, priority of the corresponding flow and indicator functions of network code are considered. Indicator functions are related to the decodability of the packet and also whether the current node is the real target node.

• If the priorities of all flows are equal, the value of priority of flow becomes 1 for all. We define indicator function *d* if the code is decodable at the node becomes 1 or 0 otherwise. If the packet is targeted to the node, indicator function *g* becomes 1 or 0 otherwise.

$$\overset{\text{overall loss probability}}{P(l)} = P\{\underbrace{FTT'}_{\text{forward trip time}} > \underbrace{\overbrace{t_d(l)}^{\text{deadline of packet}} - \overbrace{t_c}^{\text{current time}}}_{\tau}\}$$

$$P\{FTT' > \tau\} = \underbrace{\varepsilon_F}_{\text{loss probability}} + \overbrace{(1-\varepsilon_F)\int_{\tau}^{\infty} p_F(t)dt}^{\text{the probability of arriving late}}$$

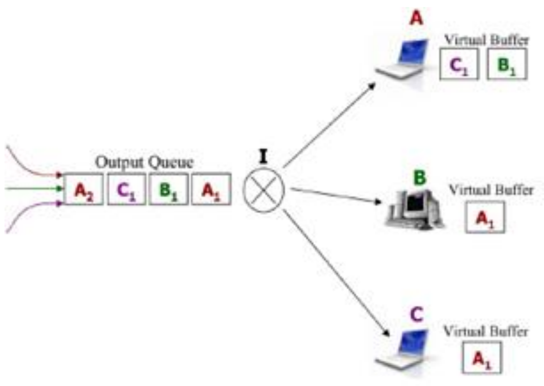
Fig.5. The step 6.1.1 of NCV Algorithm.

• According to the step 6.1.1, overall loss probability of a packet is related to the remaining time to the deadline of a packet and forward trip time (FTT) which is based on delay and loss. If FTT is greater than the remaining time, the packet will be lost.

• Loss probability is calculated with adding loss probability due to noise, fading and interference and the probability of arriving late after its deadline.

B. *NCV Algorithm Examples*

In the first example (Fig. 6), we try to choose the best network code for A_1 primary packet. There are four possible network codes. It takes account two side nodes B_1 and C_1 besides the primary packet A_1. If we take account A_2, the number of possible network codes will increase and will be 8. The number of possible codes can be determined with 2 over number of side packets.



Network Codes	Decodability
$c_1 = A_1$	only A decodes
$c_2 = A_1+B_1$	A and B decode
$c_3 = A_1+C_1$	A and C decode
$c_4 = A_1+B_1+C_1$	only A decodes

Fig.6. The First Example of NCV Algorithm.

c_1 network code contains only A_1 packet and it can be decoded by only node A because A_1 can be decoded without any help of overheard packets. Other nodes ignore A_1 because A_1 is not related to other nodes.

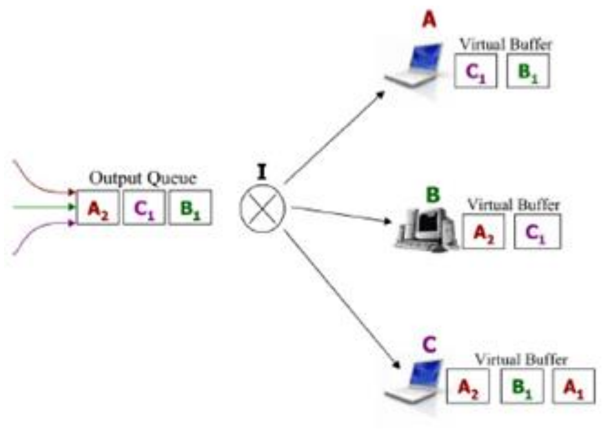
c_2 network code contains A_1 and B_1 packets. This code can be decoded by A and B nodes. With the help of the overheard packets in node A, c_2 network code can be decoded because node A has overheard packet B_1. In addition, c_2 network code can be decoded with the help of the overheard packet in node B because node B has overheard packet A_1.

c_3 network code contains A_1 and C_1 packets. A and C nodes can decode this code. With the help of the overheard packets in node A, c_3 network code can be decoded because node A has overheard packet C_1. Besides, c_3 network code can be decoded with the help of the overheard packet in node C because node C has overheard packet A_1.

c_4 network code contains A_1, B_1 and C_1 packets. Only A node can decode this code. With the help of the overheard packets in node A, c_4 network code can be decoded because node A has overheard packet B_1 and C_1. Others cannot decode it.

As a result of this scenario, c_2 and c_3 are the best possible network codes. If you ask that which code will be selected? It depends on the priority of B_1 and C_1. The code which has high priority is selected.

In the second example (Fig. 7), we try to choose the best network code for B_1 primary packet. There are four possible network codes. Because it takes account two side nodes C_1 and A_2 besides the primary packet B_1.



Network Codes	Decodability
$c_1 = B_1$	only B decodes
$c_2 = B_1 + C_1$	B and C decode
$c_3 = B_1 + A_2$	A and B decode
$c_4 = B_1 + C_1 + A_2$	A, B, C decode

Fig.7. The Second Example of NCV Algorithm.

c_1 network code contains only B_1 packet and it can be decoded by only node B, since B_1 can be decoded without any help of overheard packets. Other nodes ignore B_1, because B_1 is not related to other nodes.

c_2 network code contains B_1 and C_1 packets. This code can be decoded by B and C nodes. With the help of the overheard packets in node B, c_2 network code can be decoded because node B has overheard packet C_1. Furthermore, with the help of the overheard packet in node C, c_2 network code can be decoded because node C has overheard packet B_1.

c_3 network code contains B_1 and A_2 packets. A and B nodes can decode this code. With the help of the overheard packets in node A, c_3 network code can be decoded because node A has overheard packet B_1. In addition, with the help of the overheard packet in node B, c_3 network code can be decoded because node B has overheard packet A_2.

c_4 network code contains B_1, C_1 and A_2 packets. A, B and C nodes can decode this code. With the help of the overheard packets in node A, c_4 network code can be decoded because node A has overheard packet C_1 and B_1. Also, with the help of the overheard packets in node B, c_4 network code can be decoded because node B has overheard packet A_2 and C_1. Finally, with the help of the overheard packets in node C, c_4 network code can be decoded because node C has overheard packet A_2 and B_1. As a result of this scenario, c_4 is the best possible network code.

C. NCV Algorithm Simulation

This simulation (Fig. 8) shows how the improvement is calculated. In Tx queue, every packet has an index. In the first step, when considering the overheard packets, there are two possible network codes. These are $c_1 = A_1$ and $c_2 = A_1 + B_1$. To take improvements into account at each client, we look at the possible network codes of primary packet A_1. Besides, these improvement values are summed up for each possible network code.

In the second step (Fig. 9), after A_1 and B_1 are sent and removed from Tx queue, C_1 becomes the next primary packet which has the index number 3. Furthermore, in this situation, there is only one possible network code which is $c_1 = C_1$. Then, we can see the changes in *i* and *k* values.

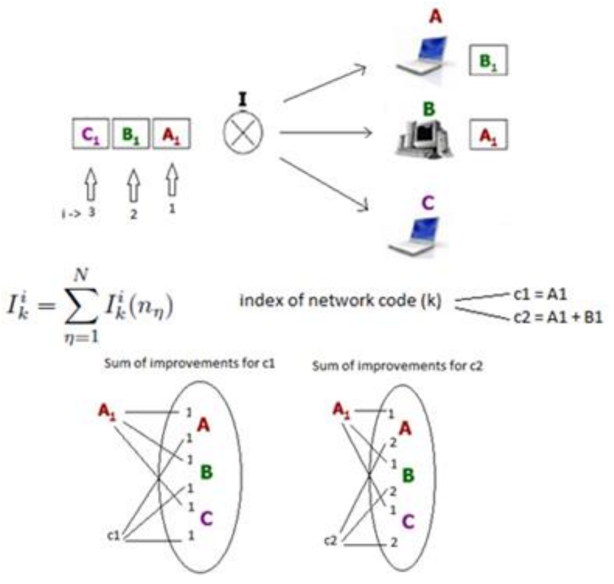

A
B1
I
C1 B1 A1
B
A1
i -> 3
2
1
C
$I_k^i = \sum_{\eta=1}^{N} I_k^i(n_\eta)$
index of network code (k)
c1 = A1
c2 = A1 + B1
Sum of improvements for c1
Sum of improvements for c2
A1
A
B
C
c1
c2

Fig.8. The First Step of NCV Algorithm Simulation.

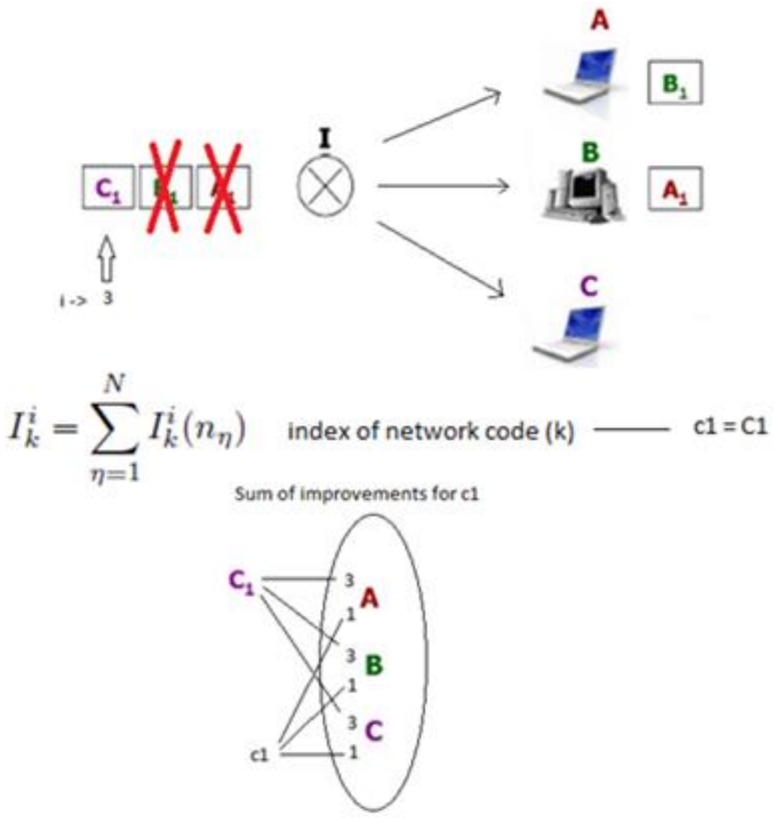
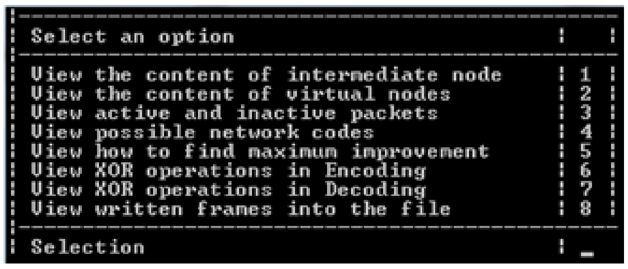

A
B
C
B_1
A_1
C_1
B_1
A_1
I
i -> 3
$I_k^i = \sum_{\eta=1}^{N} I_k^i(n_\eta)$
index of network code (k)
c1 = C1
Sum of improvements for c1
C_1
c1
3
1
3
1
3
1

Fig.9. The Second Step of NCV Algorithm Simulation.

IV. Results and Discussion

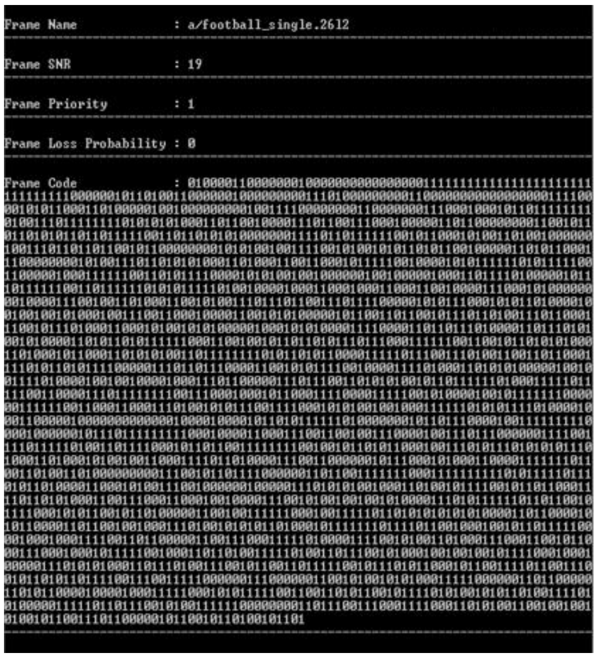
In this study, NCV algorithm which takes the output of the H.264 video codec [26] generating the video packets is implemented in C programming language using Dev-C++. In addition, XEmacs text editor is used to view the contents of video frames. The main menu of developed program is shown in Fig. 10.



```
 ---------------------------------------------------------
| Select an option                               |   |
 ---------------------------------------------------------
| View the content of intermediate node          | 1 |
| View the content of virtual nodes              | 2 |
| View active and inactive packets               | 3 |
| View possible network codes                    | 4 |
| View how to find maximum improvement           | 5 |
| View XOR operations in Encoding                | 6 |
| View XOR operations in Decoding                | 7 |
| View written frames into the file              | 8 |
 ---------------------------------------------------------
| Selection                                      | _
```

Fig.10. The Main Menu of Developed Program.

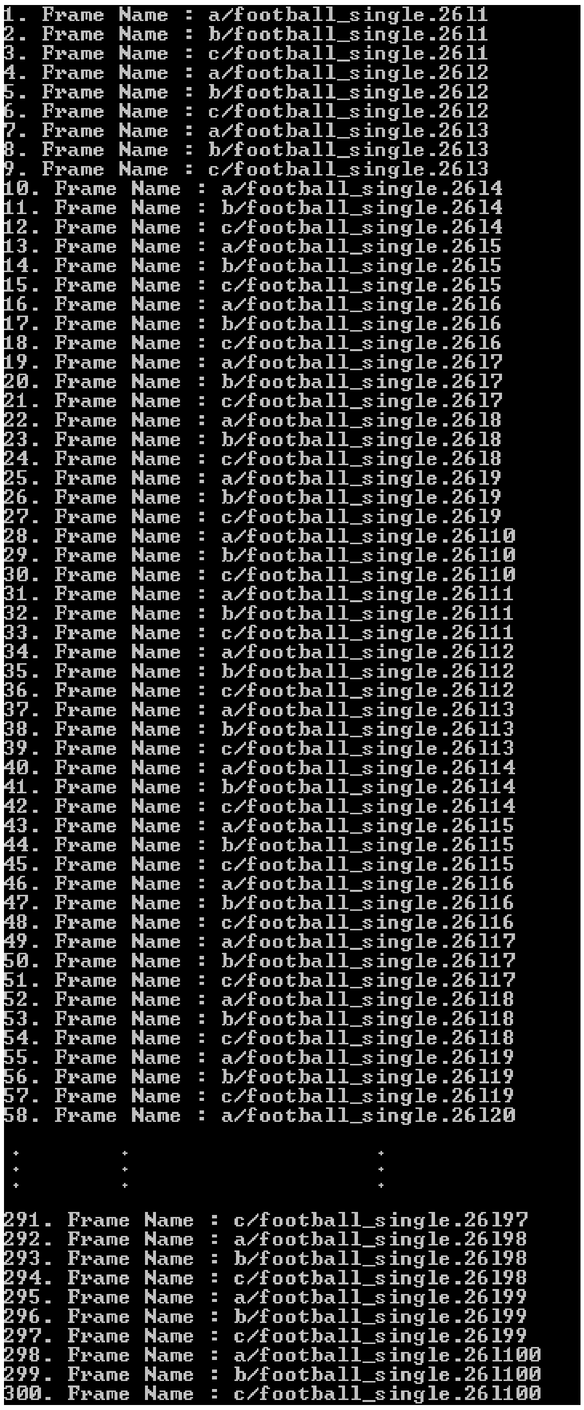
Using the main menu, desired results can be viewed by entering the number of requested option. In testing process, a football video was used and the frames were read from video packet files correctly. The system assigned names and other attributes. Besides, it converted the characters in video packet file into binary format. As an example, the attributes of a frame is shown in Fig. 11.



```
Frame Name             : a/football_single.2612
-----------------------------------------------------------------------------------
Frame SNR              : 19
-----------------------------------------------------------------------------------
Frame Priority         : 1
-----------------------------------------------------------------------------------
Frame Loss Probability : 0
-----------------------------------------------------------------------------------
Frame Code             : 0100001100000000100000000000000000111111111111111111111
1111111110000001011010011000000100000000011101000000000011000000000000000001111100
0010101100011010000010010000000001001111000000000110000000011100010001011011111111
01001110111111110101010100011011001000011101100111000100000110110000000011001011
01101010110110111110011011010101000000011110110111111001011000101001101001000000
10011101101101100101100000000010101001001111001010010101101011001000001101011000 1
1100000000101001110110101010001101000110011000101111100100001010111111010111111 00
11000001000111111001101011110000101010010010000001001000000100011011111010000001011
10111111001101111110101011111010010000100011000100011000110010000111000101000000
0010000111001001101000110010100111011101100111011110000001010111000101011010000 10
010010010100010011100110001000011001010100000010110011011001011101101001110110001
110010111010001100010100101010000001000101010000111100001101101110100001101110101
0010100001101011010111111000110010010110110101110111000111111001100101101010100 00
11010001011000110101010011011111111010110101100001111101110011101001100110110001
11101011010111100000111011011100001100101011110010000111101000110101010000010010
0111101000010010010000100011101100000111011100110101010010110111111010001111011
111001100001110111111110011100010001011000111100001111100101000010010111111100000
0011111110011000110001110100101011100111100010101001001000111111010101111010000 10
001100000100000000000001000010000101101011111101000000001011011100001001111111110
0001000000101110111111111000100001100011100110010011100001001110111000000111001
1110111110100110111100010110110011111110010010110101100010011101011101010101110
1000110100010100100110001111011010000111001100000010111000101000110000111111011
00110100110100000000011100101101111000000110110011111110001111111111010111110111
01011010000110001010011100100000010000011101010100100011010010111100101101100011
11011010100011001110001100010010000111001010010010010100001110101111110110110010
111100010101100101101000000110010011111110001001111101101010101010000011011000010
1011000011011001001000111010010101011010001011111111011110110010001001011011111100
00100010001111001101100000011001110001111101000011110010100110100011100011001011 0
00111000100010111110010001101101001111101001101110010100010010010010111100010001
00000111010101000110111010011100101100110111110010111010110001011001111011001110
010110101101111001110011111000000110000000110010100101010001111100000001101100000
11010110000100001000111110001010111110011001101011001011110101001010110100111101
01000001111101101110010100111111000000001101110011100011110001101010011001001001
010010110011101100000010110010110100101101
-----------------------------------------------------------------------------------
```

Fig.11. The Attributes of a Frame.

We confirmed that there was no missing packet after reading from the file. There were 300 packets in the intermediate node queue and 100 packets for each node. These packets are shown in Fig. 12. Then, it was seen that the active and inactive packets were listed correctly after creating each network code. These packets are shown in Fig. 13.



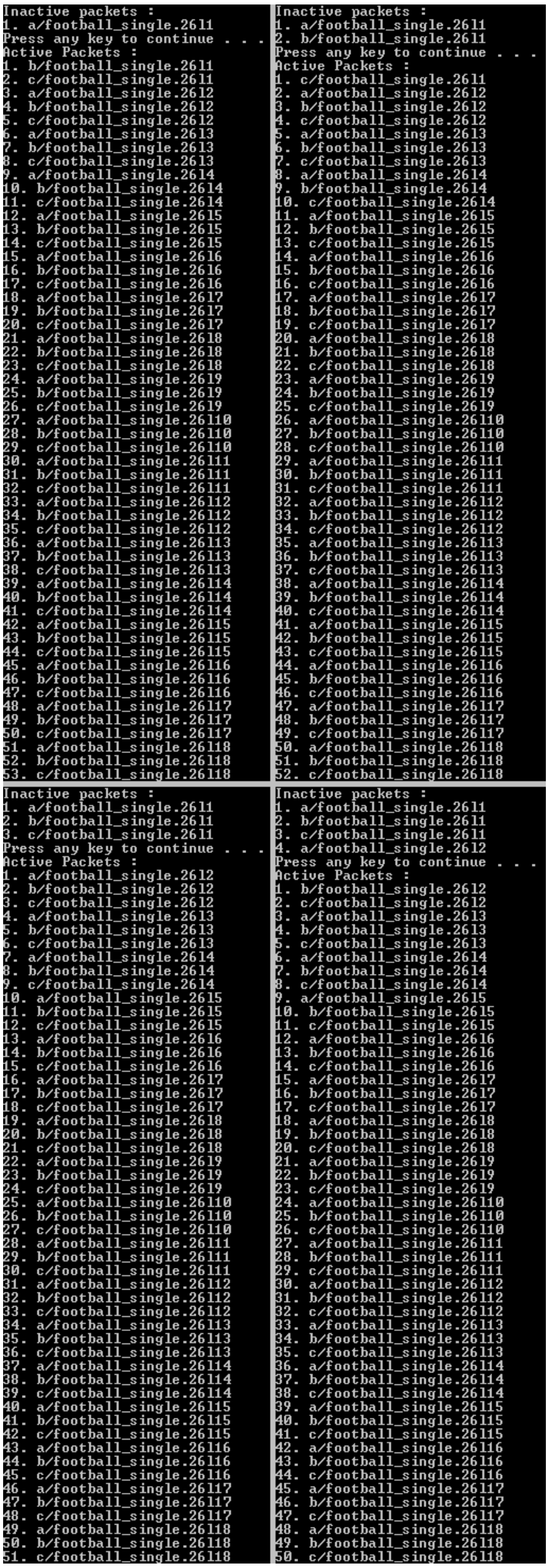
```
1. Frame Name : a/football_single.2611
2. Frame Name : b/football_single.2611
3. Frame Name : c/football_single.2611
4. Frame Name : a/football_single.2612
5. Frame Name : b/football_single.2612
6. Frame Name : c/football_single.2612
7. Frame Name : a/football_single.2613
8. Frame Name : b/football_single.2613
9. Frame Name : c/football_single.2613
10. Frame Name : a/football_single.2614
11. Frame Name : b/football_single.2614
12. Frame Name : c/football_single.2614
13. Frame Name : a/football_single.2615
14. Frame Name : b/football_single.2615
15. Frame Name : c/football_single.2615
16. Frame Name : a/football_single.2616
17. Frame Name : b/football_single.2616
18. Frame Name : c/football_single.2616
19. Frame Name : a/football_single.2617
20. Frame Name : b/football_single.2617
21. Frame Name : c/football_single.2617
22. Frame Name : a/football_single.2618
23. Frame Name : b/football_single.2618
24. Frame Name : c/football_single.2618
25. Frame Name : a/football_single.2619
26. Frame Name : b/football_single.2619
27. Frame Name : c/football_single.2619
28. Frame Name : a/football_single.26110
29. Frame Name : b/football_single.26110
30. Frame Name : c/football_single.26110
31. Frame Name : a/football_single.26111
32. Frame Name : b/football_single.26111
33. Frame Name : c/football_single.26111
34. Frame Name : a/football_single.26112
35. Frame Name : b/football_single.26112
36. Frame Name : c/football_single.26112
37. Frame Name : a/football_single.26113
38. Frame Name : b/football_single.26113
39. Frame Name : c/football_single.26113
40. Frame Name : a/football_single.26114
41. Frame Name : b/football_single.26114
42. Frame Name : c/football_single.26114
43. Frame Name : a/football_single.26115
44. Frame Name : b/football_single.26115
45. Frame Name : c/football_single.26115
46. Frame Name : a/football_single.26116
47. Frame Name : b/football_single.26116
48. Frame Name : c/football_single.26116
49. Frame Name : a/football_single.26117
50. Frame Name : b/football_single.26117
51. Frame Name : c/football_single.26117
52. Frame Name : a/football_single.26118
53. Frame Name : b/football_single.26118
54. Frame Name : c/football_single.26118
55. Frame Name : a/football_single.26119
56. Frame Name : b/football_single.26119
57. Frame Name : c/football_single.26119
58. Frame Name : a/football_single.26120

 .        .                    .
 .        .                    .
 .        .                    .

291. Frame Name : c/football_single.26197
292. Frame Name : a/football_single.26198
293. Frame Name : b/football_single.26198
294. Frame Name : c/football_single.26198
295. Frame Name : a/football_single.26199
296. Frame Name : b/football_single.26199
297. Frame Name : c/football_single.26199
298. Frame Name : a/football_single.261100
299. Frame Name : b/football_single.261100
300. Frame Name : c/football_single.261100
```

Fig.12. The Packets in the Intermediate Node Queue.

After that possible network codes were created correctly (Fig. 14). The developed program found the best network code that provides maximum improvement according to each improvement values and the same process was performed for every possible network code frame groups (Fig. 15). Then, the maximum code frames were XOR-ed correctly at encoding process and the best network code was obtained (Fig. 16). The best network code and the overheard packet code were XOR-ed accurately in order to obtain desirable frames (Fig. 17). Then, the transmission occurred and the frames were placed in the result folder after network coding (Fig. 18). Finally, the contents of frames were viewed the same as the original using XEmacs text editor (Fig. 19).



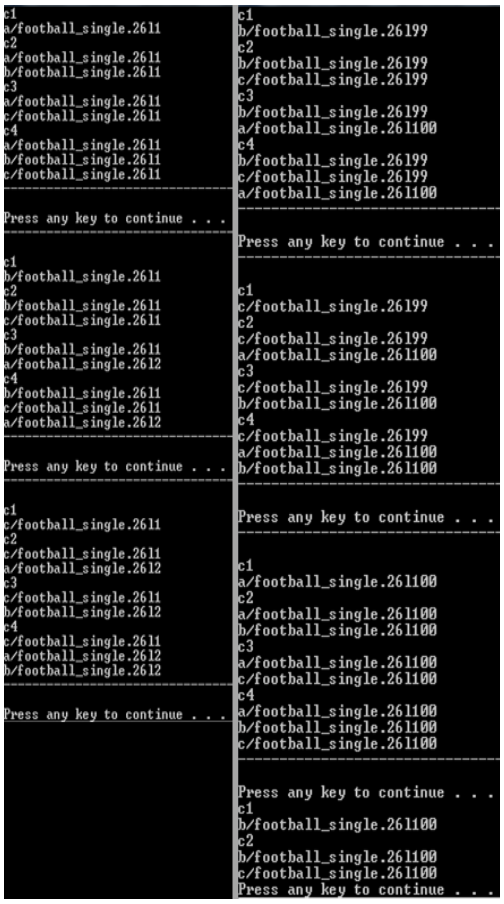
```
Inactive packets :
1. a/football_single.2611
Press any key to continue . . .
Active Packets :
1. b/football_single.2611
2. c/football_single.2611
3. a/football_single.2612
4. b/football_single.2612
5. c/football_single.2612
6. a/football_single.2613
7. b/football_single.2613
8. c/football_single.2613
9. a/football_single.2614
10. b/football_single.2614
11. c/football_single.2614
12. a/football_single.2615
13. b/football_single.2615
14. c/football_single.2615
15. a/football_single.2616
16. b/football_single.2616
17. c/football_single.2616
18. a/football_single.2617
19. b/football_single.2617
20. c/football_single.2617
21. a/football_single.2618
22. b/football_single.2618
23. c/football_single.2618
24. a/football_single.2619
25. b/football_single.2619
26. c/football_single.2619
27. a/football_single.26110
28. b/football_single.26110
29. c/football_single.26110
30. a/football_single.26111
31. b/football_single.26111
32. c/football_single.26111
33. a/football_single.26112
34. b/football_single.26112
35. c/football_single.26112
36. a/football_single.26113
37. b/football_single.26113
38. c/football_single.26113
39. a/football_single.26114
40. b/football_single.26114
41. c/football_single.26114
42. a/football_single.26115
43. b/football_single.26115
44. c/football_single.26115
45. a/football_single.26116
46. b/football_single.26116
47. c/football_single.26116
48. a/football_single.26117
49. b/football_single.26117
50. c/football_single.26117
51. a/football_single.26118
52. b/football_single.26118
53. c/football_single.26118
```

```
Inactive packets :
1. a/football_single.2611
2. b/football_single.2611
Press any key to continue . . .
Active Packets :
1. c/football_single.2611
2. a/football_single.2612
3. b/football_single.2612
4. c/football_single.2612
5. a/football_single.2613
6. b/football_single.2613
7. c/football_single.2613
8. a/football_single.2614
9. b/football_single.2614
10. c/football_single.2614
11. a/football_single.2615
12. b/football_single.2615
13. c/football_single.2615
14. a/football_single.2616
15. b/football_single.2616
16. c/football_single.2616
17. a/football_single.2617
18. b/football_single.2617
19. c/football_single.2617
20. a/football_single.2618
21. b/football_single.2618
22. c/football_single.2618
23. a/football_single.2619
24. b/football_single.2619
25. c/football_single.2619
26. a/football_single.26110
27. b/football_single.26110
28. c/football_single.26110
29. a/football_single.26111
30. b/football_single.26111
31. c/football_single.26111
32. a/football_single.26112
33. b/football_single.26112
34. c/football_single.26112
35. a/football_single.26113
36. b/football_single.26113
37. c/football_single.26113
38. a/football_single.26114
39. b/football_single.26114
40. c/football_single.26114
41. a/football_single.26115
42. b/football_single.26115
43. c/football_single.26115
44. a/football_single.26116
45. b/football_single.26116
46. c/football_single.26116
47. a/football_single.26117
48. b/football_single.26117
49. c/football_single.26117
50. a/football_single.26118
51. b/football_single.26118
52. c/football_single.26118
```

```
Inactive packets :
1. a/football_single.2611
2. b/football_single.2611
3. c/football_single.2611
Press any key to continue . . .
Active Packets :
1. a/football_single.2612
2. b/football_single.2612
3. c/football_single.2612
4. a/football_single.2613
5. b/football_single.2613
6. c/football_single.2613
7. a/football_single.2614
8. b/football_single.2614
9. c/football_single.2614
10. a/football_single.2615
11. b/football_single.2615
12. c/football_single.2615
13. a/football_single.2616
14. b/football_single.2616
15. c/football_single.2616
16. a/football_single.2617
17. b/football_single.2617
18. c/football_single.2617
19. a/football_single.2618
20. b/football_single.2618
21. c/football_single.2618
22. a/football_single.2619
23. b/football_single.2619
24. c/football_single.2619
25. a/football_single.26110
26. b/football_single.26110
27. c/football_single.26110
28. a/football_single.26111
29. b/football_single.26111
30. c/football_single.26111
31. a/football_single.26112
32. b/football_single.26112
33. c/football_single.26112
34. a/football_single.26113
35. b/football_single.26113
36. c/football_single.26113
37. a/football_single.26114
38. b/football_single.26114
39. c/football_single.26114
40. a/football_single.26115
41. b/football_single.26115
42. c/football_single.26115
43. a/football_single.26116
44. b/football_single.26116
45. c/football_single.26116
46. a/football_single.26117
47. b/football_single.26117
48. c/football_single.26117
49. a/football_single.26118
50. b/football_single.26118
51. c/football_single.26118
```

```
Inactive packets :
1. a/football_single.2611
2. b/football_single.2611
3. c/football_single.2611
4. a/football_single.2612
Press any key to continue . . .
Active Packets :
1. b/football_single.2612
2. c/football_single.2612
3. a/football_single.2613
4. b/football_single.2613
5. c/football_single.2613
6. a/football_single.2614
7. b/football_single.2614
8. c/football_single.2614
9. a/football_single.2615
10. b/football_single.2615
11. c/football_single.2615
12. a/football_single.2616
13. b/football_single.2616
14. c/football_single.2616
15. a/football_single.2617
16. b/football_single.2617
17. c/football_single.2617
18. a/football_single.2618
19. b/football_single.2618
20. c/football_single.2618
21. a/football_single.2619
22. b/football_single.2619
23. c/football_single.2619
24. a/football_single.26110
25. b/football_single.26110
26. c/football_single.26110
27. a/football_single.26111
28. b/football_single.26111
29. c/football_single.26111
30. a/football_single.26112
31. b/football_single.26112
32. c/football_single.26112
33. a/football_single.26113
34. b/football_single.26113
35. c/football_single.26113
36. a/football_single.26114
37. b/football_single.26114
38. c/football_single.26114
39. a/football_single.26115
40. b/football_single.26115
41. c/football_single.26115
42. a/football_single.26116
43. b/football_single.26116
44. c/football_single.26116
45. a/football_single.26117
46. b/football_single.26117
47. c/football_single.26117
48. a/football_single.26118
49. b/football_single.26118
50. c/football_single.26118
```

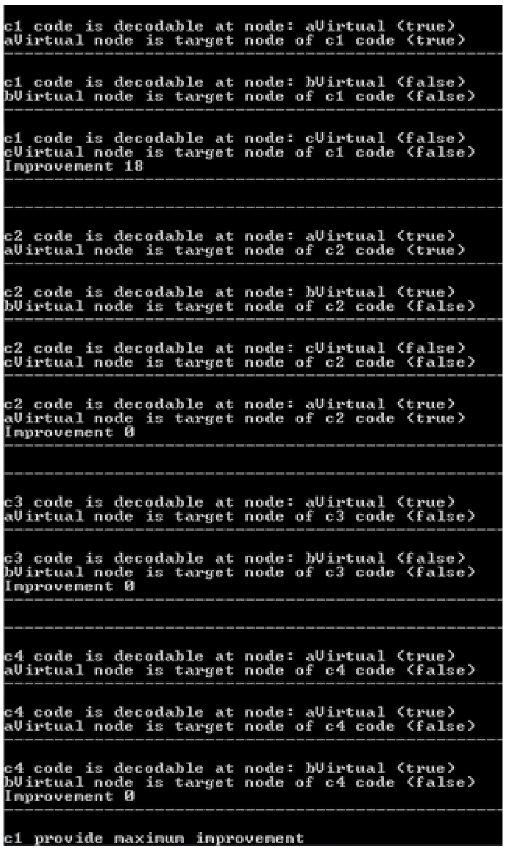
Fig.13. The Active and Inactive Packets.



```
c1
a/football_single.2611
c2
a/football_single.2611
b/football_single.2611
c3
a/football_single.2611
c/football_single.2611
c4
a/football_single.2611
b/football_single.2611
c/football_single.2611
------------------------------------------
Press any key to continue . . .
c1
b/football_single.2611
c2
b/football_single.2611
c/football_single.2611
c3
b/football_single.2611
a/football_single.2612
c4
b/football_single.2611
c/football_single.2611
a/football_single.2612
------------------------------------------
Press any key to continue . . .
c1
c/football_single.2611
c2
c/football_single.2611
a/football_single.2612
c3
c/football_single.2611
b/football_single.2612
c4
c/football_single.2611
a/football_single.2612
b/football_single.2612
------------------------------------------
Press any key to continue . . .
```

```
c1
b/football_single.26199
c2
b/football_single.26199
c/football_single.26199
c3
b/football_single.26199
a/football_single.261100
c4
b/football_single.26199
c/football_single.26199
a/football_single.261100
------------------------------------------
Press any key to continue . . .
c1
c/football_single.26199
c2
c/football_single.26199
a/football_single.261100
c3
c/football_single.26199
b/football_single.261100
c4
c/football_single.26199
a/football_single.261100
b/football_single.261100
------------------------------------------
Press any key to continue . . .
c1
a/football_single.261100
c2
a/football_single.261100
b/football_single.261100
c3
a/football_single.261100
c/football_single.261100
c4
a/football_single.261100
b/football_single.261100
c/football_single.261100
------------------------------------------
Press any key to continue . . .
c1
b/football_single.261100
c2
b/football_single.261100
c/football_single.261100
Press any key to continue . . .
```

Fig.14. Possible Network Codes.



```
c1 code is decodable at node: aVirtual (true)
aVirtual node is target node of c1 code (true)
------------------------------------------
c1 code is decodable at node: bVirtual (false)
bVirtual node is target node of c1 code (false)
------------------------------------------
c1 code is decodable at node: cVirtual (false)
cVirtual node is target node of c1 code (false)
Improvement 18
------------------------------------------

------------------------------------------
c2 code is decodable at node: aVirtual (true)
aVirtual node is target node of c2 code (true)
------------------------------------------
c2 code is decodable at node: bVirtual (true)
bVirtual node is target node of c2 code (false)
------------------------------------------
c2 code is decodable at node: cVirtual (false)
cVirtual node is target node of c2 code (false)
------------------------------------------
c2 code is decodable at node: aVirtual (true)
aVirtual node is target node of c2 code (true)
Improvement 0
------------------------------------------

------------------------------------------
c3 code is decodable at node: aVirtual (true)
aVirtual node is target node of c3 code (false)
------------------------------------------
c3 code is decodable at node: bVirtual (false)
bVirtual node is target node of c3 code (false)
Improvement 0
------------------------------------------

------------------------------------------
c4 code is decodable at node: aVirtual (true)
aVirtual node is target node of c4 code (false)
------------------------------------------
c4 code is decodable at node: aVirtual (true)
aVirtual node is target node of c4 code (false)
------------------------------------------
c4 code is decodable at node: bVirtual (true)
bVirtual node is target node of c4 code (false)
Improvement 0
------------------------------------------
c1 provide maximum improvement
```

Fig.15. Finding the Best Network Code According to Each Improvement Values.

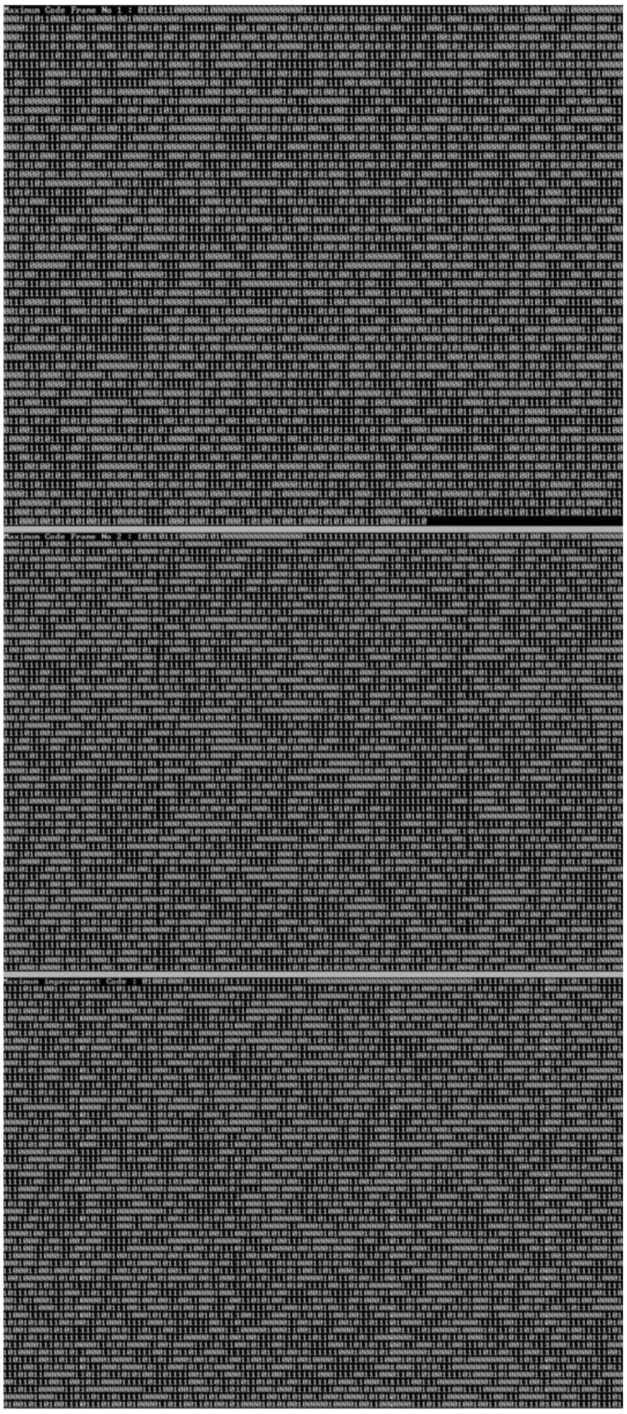
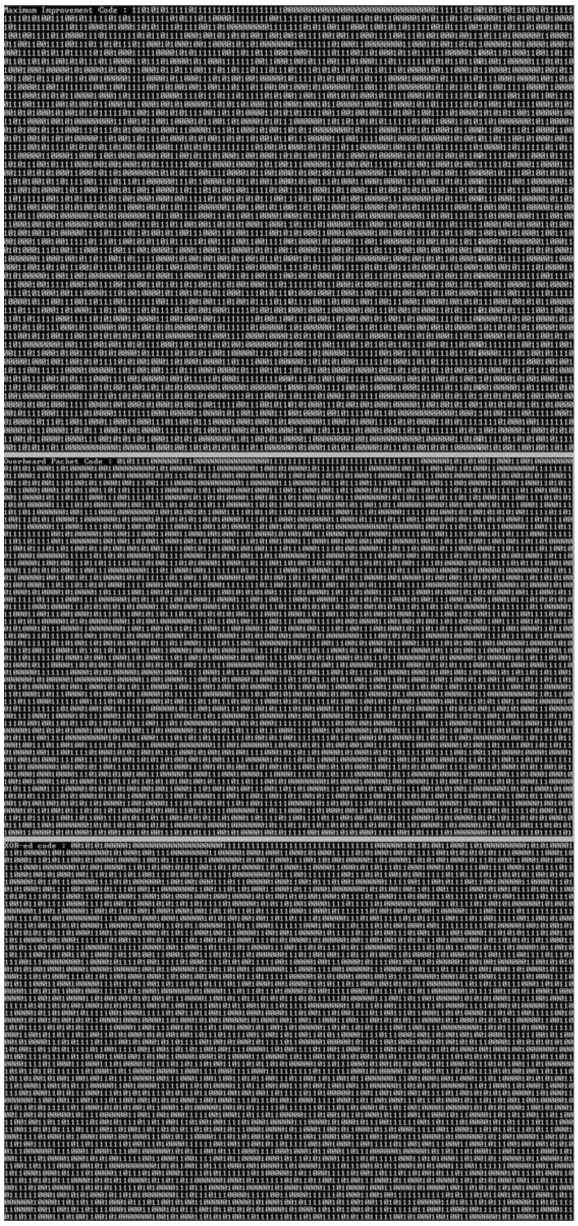
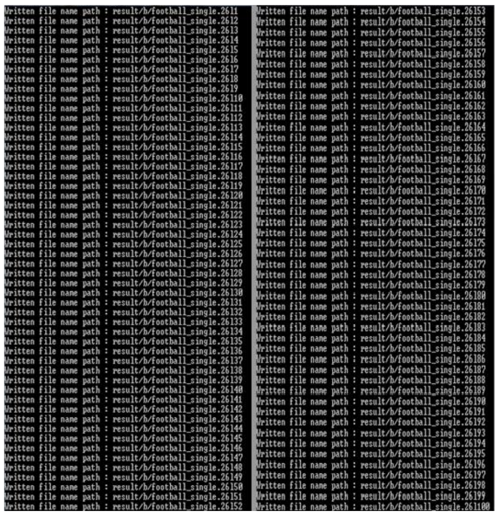


Fig.16. XOR-ing Maximum Code Frames and Obtaining the Best Network Code



```
Maximum Improvement Code : [illegible]
Overheard Packet Code : [illegible]
XOR-ed code : [illegible]
```

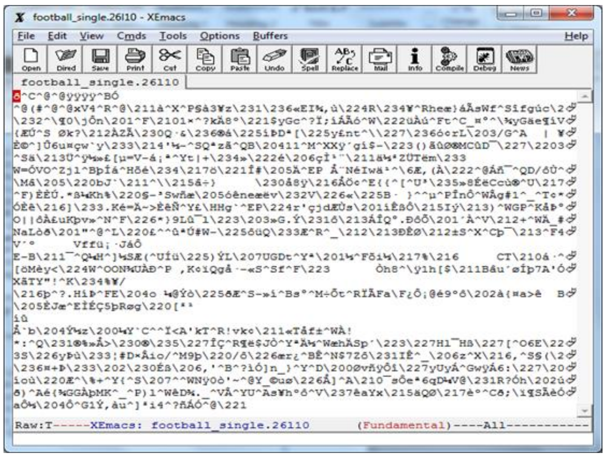
Fig.17. XOR-ing the Best Network Code and the Overheard Packet Code.



```
Written file name path : result/b/football_single.2611
Written file name path : result/b/football_single.2612
Written file name path : result/b/football_single.2613
Written file name path : result/b/football_single.2614
Written file name path : result/b/football_single.2615
Written file name path : result/b/football_single.2616
Written file name path : result/b/football_single.2617
Written file name path : result/b/football_single.2618
Written file name path : result/b/football_single.2619
Written file name path : result/b/football_single.26110
Written file name path : result/b/football_single.26111
Written file name path : result/b/football_single.26112
Written file name path : result/b/football_single.26113
Written file name path : result/b/football_single.26114
Written file name path : result/b/football_single.26115
Written file name path : result/b/football_single.26116
Written file name path : result/b/football_single.26117
Written file name path : result/b/football_single.26118
Written file name path : result/b/football_single.26119
Written file name path : result/b/football_single.26120
Written file name path : result/b/football_single.26121
Written file name path : result/b/football_single.26122
Written file name path : result/b/football_single.26123
Written file name path : result/b/football_single.26124
Written file name path : result/b/football_single.26125
Written file name path : result/b/football_single.26126
Written file name path : result/b/football_single.26127
Written file name path : result/b/football_single.26128
Written file name path : result/b/football_single.26129
Written file name path : result/b/football_single.26130
Written file name path : result/b/football_single.26131
Written file name path : result/b/football_single.26132
Written file name path : result/b/football_single.26133
Written file name path : result/b/football_single.26134
Written file name path : result/b/football_single.26135
Written file name path : result/b/football_single.26136
Written file name path : result/b/football_single.26137
Written file name path : result/b/football_single.26138
Written file name path : result/b/football_single.26139
Written file name path : result/b/football_single.26140
Written file name path : result/b/football_single.26141
Written file name path : result/b/football_single.26142
Written file name path : result/b/football_single.26143
Written file name path : result/b/football_single.26144
Written file name path : result/b/football_single.26145
Written file name path : result/b/football_single.26146
Written file name path : result/b/football_single.26147
Written file name path : result/b/football_single.26148
Written file name path : result/b/football_single.26149
Written file name path : result/b/football_single.26150
Written file name path : result/b/football_single.26151
Written file name path : result/b/football_single.26152
```

```
Written file name path : result/b/football_single.26153
Written file name path : result/b/football_single.26154
Written file name path : result/b/football_single.26155
Written file name path : result/b/football_single.26156
Written file name path : result/b/football_single.26157
Written file name path : result/b/football_single.26158
Written file name path : result/b/football_single.26159
Written file name path : result/b/football_single.26160
Written file name path : result/b/football_single.26161
Written file name path : result/b/football_single.26162
Written file name path : result/b/football_single.26163
Written file name path : result/b/football_single.26164
Written file name path : result/b/football_single.26165
Written file name path : result/b/football_single.26166
Written file name path : result/b/football_single.26167
Written file name path : result/b/football_single.26168
Written file name path : result/b/football_single.26169
Written file name path : result/b/football_single.26170
Written file name path : result/b/football_single.26171
Written file name path : result/b/football_single.26172
Written file name path : result/b/football_single.26173
Written file name path : result/b/football_single.26174
Written file name path : result/b/football_single.26175
Written file name path : result/b/football_single.26176
Written file name path : result/b/football_single.26177
Written file name path : result/b/football_single.26178
Written file name path : result/b/football_single.26179
Written file name path : result/b/football_single.26180
Written file name path : result/b/football_single.26181
Written file name path : result/b/football_single.26182
Written file name path : result/b/football_single.26183
Written file name path : result/b/football_single.26184
Written file name path : result/b/football_single.26185
Written file name path : result/b/football_single.26186
Written file name path : result/b/football_single.26187
Written file name path : result/b/football_single.26188
Written file name path : result/b/football_single.26189
Written file name path : result/b/football_single.26190
Written file name path : result/b/football_single.26191
Written file name path : result/b/football_single.26192
Written file name path : result/b/football_single.26193
Written file name path : result/b/football_single.26194
Written file name path : result/b/football_single.26195
Written file name path : result/b/football_single.26196
Written file name path : result/b/football_single.26197
Written file name path : result/b/football_single.26198
Written file name path : result/b/football_single.26199
Written file name path : result/b/football_single.261100
```

Fig.18. Placing the Frames in Result Folder after Network Coding.


football_single.26110 - XEmacs
File Edit View Cmds Tools Options Buffers Help
Open Dired Save Print Cut Copy Paste Undo Spell Replace Mail Info Compile Debug News
football_single.26110

```
[illegible]
Raw:T-----XEmacs: football_single.26110      (Fundamental)----All-----------
```

Fig.19. Viewing the Contents of Frames after Network Coding.

V. CONCLUSION

In this study, in order to increase throughput and video quality during video streaming over wireless networks, NCV algorithm is used. This algorithm is implemented in C programming language and takes the output of the H.264 video codec that generates the video packets.

The important point is to transmit flows which increase both video quality and throughput. Additionally, the deadlines of packets, the condition of the network, overheard packets on neighbor nodes and the importance of video packets are taken into account. In NCV, many flows may come to an intermediate node. In this situation, network codes are generated by looking at the buffers of clients. NCV algorithm chooses the best network code from all possible codes. The content of assistive virtual buffers of clients is utilized to get an intended primary packet from the network code. In addition to chosen primary packet, the chosen side packets must be overheard by the target node. When choosing the right network code, the most important problem is to determine whether the candidate codes can be decoded at target node or not. The network code is decoded by all neighboring nodes simultaneously. Improvement is calculated for each possible code on all receiving nodes. Finally, the improvement which has the biggest value gives the network code and this code improves video quality. According to chosen network codes, video transmission occurs. It is observed that NCV algorithm improves video quality and throughput according to the experiments performed in this study.

ACKNOWLEDGMENT

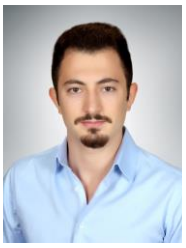
We would like to thank our project advisor Assist. Prof. Dr. Sıla Ekmekçi FLIERL for continuous assistance and contribution throughout the project process.

REFERENCES

[1] H. Seferoglu and A. Markopoulou, "Video-Aware Opportunistic Network Coding over Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol.

27, no. 5, pp. 713–728, June 2009.
[2] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, “XORs in the Air: Practical Wireless Network Coding,” *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 497–510, June 2008.
[3] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, “Link-level measurements from an 802.11b mesh network,” *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '04)*, Portland, OR, pp. 121–132, Aug. 2004.
[4] H. Shiang and M. van der Schaar, “Multi-user video streaming over multi-hop wireless networks: A distributed, cross-layer approach based on priority queuing,” *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 4, pp. 770–785, May 2007.
[5] B. Girod and N. Farber, “Wireless Video,” *Compressed Video over Networks*, A. Reibman and M.-T. Sun, Eds. New York: M. Dekker, pp. 124–133, 2000.
[6] Special issue on “Advances in Wireless Video,” *IEEE Wireless Communications Magazine*, Aug. 2005.
[7] B. Girod, J. Chakareski, M. Kalman, Y.J. Liang, E. Setton, and R. Zhang, “Advances in Network-adaptive Video Streaming,” *Proceedings of IWDC 2002*, Capri, Italy, Sept. 2002.
[8] C. Fragouli. J.Y. Le Boudec, and J. Widmer, “Network coding: an instant primer,” *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 63–68, Jan. 2006.
[9] Y. Wu, P.A. Chou, and S.-Y. Kung, “Information exchange in wireless networks with network coding and physical-layer broadcast,” *Technical Report MSR-TR-2004-78*, Microsoft Research, Aug. 2004.
[10] C. Fragouli, J. Widmer, and J.Y. Le Boudec, “A network coding approach to energy efficient broadcasting: from theory to practice,” *Technical Report LCA-REPORT-2005-009, accepted at Infocom 2006*, EPFL, July 2005.
[11] J. Widmer, C. Fragouli, and J.Y. Le Boudec, “Energy efficient broadcasting in wireless ad hoc networks,” *Proceedings of First Workshop on Network Coding*, March 2005.
[12] J. Widmer and J.Y. Le Boudec, “Network coding for efficient communication in extreme networks,” *Proceedings of Workshop on delay tolerant networking and related networks (WDTN-05)*, Philadelphia, PA, Aug. 2005.
[13] Avalanche: File swarming with network coding. http://research.microsoft.com/pablo/avalanche.aspx.
[14] C. Gkantsidis and P. Rodriguez, “Network coding for large scale content distribution,” *Proceedings of IEEE Infocom*, Miami, FL, Mar. 2005.
[15] D. Petrovic, K. Ramchandran, and J. Rabaey, “Overcoming untuned radios in wireless networks with network coding,” *Proceedings of First Workshop on Network Coding, Theory, and Applications (NetCod)*, Italy, Apr. 2005.
[16] C. Fragouli and A. Markopoulou, “A network coding approach to overlay network monitoring,” *Proceedings of Allerton Conference*, Sept. 2005.
[17] T. Ho, B. Leong, Y. Chang, Y. Wen, and R. Koetter, “Network monitoring in multicast networks using network coding,” *Proceedings of International Symposium on Information Theory (ISIT)*, 2005.
[18] N. Cai and R.W. Yeung, “Secure network coding,” *Proceedings of International Symposium on Information Theory (ISIT)*, 2002.
[19] K. Bhattad and K.R. Nayayanan, “Weakly secure network coding,” *Proceedings of First Workshop on Network Coding, Theory, and Applications (NetCod)*, Apr. 2005.
[20] S-Y. R. Li, R.W. Yeung, and N. Cai, “Linear network coding,” *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.
[21] Q. Dong, J. Wu, W. Hu, and J. Crowcroft, “Practical Network Coding in Wireless Networks,” *Proceedings of MobiCom’07*, Montréal, Québec, Canada, pp. 1–4, Sep. 2007.
[22] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, “Network information flow,” *IEEE Transactions on Information Theory*, vol. 46, no. 4, July 2000.
[23] H. Seferoglu and A. Markopoulou, “Opportunistic network coding for video streaming over wireless,” *Proceedings of the 16thPacket Video Workshop*, Lausanne, Switzerland, pp. 191–200, Nov. 2007.
[24] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, “XORs in the air: Practical wireless network coding,” *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM’06)*, Pisa, Italy, pp. 243–254, Sept. 2006.
[25] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Medard, “The Importance of Being Opportunistic: Practical Network Coding For Wireless Environments,” *Proceedings of Allerton Conference*, 2005.
[26] ITU-T Rec., H.264: Advanced video coding for generic audiovisual services, 2010.

Authors’ Profiles



Can Eyupoglu received his B.Sc. degree (with high honor) in Computer Engineering and Minor degree in Electronics Engineering from Istanbul Kultur University, Istanbul, Turkey in 2012. He completed his M.Sc. degree in Computer Engineering from Istanbul University, Istanbul, Turkey in 2014. He is a Ph.D. candidate in Istanbul University, Department of Computer Engineering. He has been working as a Research Assistant in Istanbul Commerce University, Department of Computer Engineering since 2013. His research interests include optical networks, network security, computer arithmetic, big data and image processing.

Ugur Yesilyurt received his B.Sc. degree (with honor) in Computer Engineering from Istanbul Kultur University, Istanbul, Turkey in 2012. He has been working as a Software Specialist in Takasbank, Department of Application Development since 2013. His research interests include web programming, software architecture and database management.