Wireless broadcast transmission scheme for reliable video-streaming service

MELISA JUNUZOVIĆ, KEMAL ALIČ, ALEŠ ŠVIGELJ
Department of Communication Systems, Jozef Stefan Institute
Jozef Stefan International Postgraduate School
Jamova 39, 1000 Ljubljana
SLOVENIA
{melisa.junuzovic, kemal.alic, ales.svigelj}@ijs.si

Abstract: Recently, there has been growing interest in real-time streaming services such as IPTV over wireless access networks as it promises to deliver multimedia contents to clients whenever and wherever. However, providing those services over wireless networks in a timely and reliable manner is challenging due to high bandwidth demands, scarce radio resources, and lossy characteristics of wireless networks. A wireless channel may suffer from multi-path fading and interference, which may cause random packet losses; to make things worse, wireless link layer does not provide retransmission for multicast/broadcast traffic. This would impact the clients' quality of experience of an IPTV programme. Unicast retransmission solutions improves client's quality but at the bandwidth expense. This paper presents a transmission scheme for video streaming applications over last mile wireless networks, relaying on the use of Network Coding techniques to increase the overall performance, by means of reducing the number of physical transmissions resulting in reduced bandwidth consumption. Due to the bandwidth reduction, Internet Service Providers can increase the number of clients over the same infrastructure or, alternatively, offer more services to the clients. Results from practical testbed show that the number of transmissions can be significantly reduced and total bandwidth required to deliver the same content can be reduced for up to 15 %.

Key-Words: - Retransmission, wireless network, broadcast, network coding, streaming, IPTV

1 Introduction

The provision of high quality streaming services over wireless systems with increased demands for scarce radio resources, introduces never-ending challenges of efficient delivery of streaming video content, in particular when wireless technology (e.g. Wi-Fi) is used in the last mile. Wireless IEEE 802.11 networks, where one antenna covers several different clients/groups of clients, are often used due to its ease and fast deployment and relatively low deployment costs, especially in underdeveloped countries and rural areas. However, spectrum limitations in wireless medium shift the bottleneck in content delivery to the last mile. Increasing the last mile access performance in such cases is for Internet Service Providers (ISPs) of paramount importance as bandwidth reduction means additional clients over the same infrastructure or, alternatively, additional services that can be offered.

In this paper we are interested in the efficient and quality delivery of video-streaming services such as Internet Protocol Television (IPTV) in the last mile over the IEEE 802.11 wireless broadcast network.

In general, video content can be delivered over IEEE 802.11 wireless network using broadcast/multicast or unicast mechanism [1]. The unicast mechanism delivers the content to clients individually and supports retransmissions (and back-off) that assure reliable content delivery and is thus preferred solution in currently deployed systems. Instead of unicasting multiple streams of video content, a goal of broadcasting and multicasting is to reduce bandwidth [2]. Broadcast reduces bandwidth when multiple clients watch the same video stream, since the server has to send only one packet, instead of multiple packets to different clients. Under ideal wireless channel conditions [3] all broadcast packets are delivered to all clients. In the practical wireless environment the packet losses frequently occur [4] which causes degraded Quality of Services (QoS) and Quality of Experience (QoE) of the user [5] [6]; thus, some sort of retransmission mechanism is necessary to ensure reliability. However, broadcast does not assure reliable content delivery as it does not support retransmissions. In order to support reliability and/or reduce bandwidth consumption different retransmission schemes have been proposed, where some of them also consider Network Coding (NC).

Since the pioneering work of NC [7] numerous papers appeared on this subject and significant progress has been made in applying NC to different networks. For example NC is increasing throughput in satellite [8] and P2P networks [9], improving delivery reliability over the lossy links either in wireless networks over TCP [10] or in Delay Tolerant Networks such as deep space links [11]. NC is also used in ad-hoc networks for increasing throughput [12], for bandwidth allocation in wireless networks [13], for public safety communication over LTE networks [14], for enhancing TCP performance in wireless sensor networks [15] etc.

Depending on the NC application the implementation affects different OSI layers. In multicast scenarios NC is typically implemented in the application layer while two stage NC for increased spectrum efficiency is deployed in the physical layer [16].

NC concept can be used in the transmission scheme to reduce bandwidth consumption. Let us illustrate NC using an elementary examples from Fig.1 and Fig.2. Consider two clients and a broadcasting server that had transmitted packets A and B. Assume that Client 1 has not successfully received packet A and Client 2 has not successfully received packet B. Instead of retransmitting packets A and B separately as in Fig.1, the server codes packets A and B together using linear operation (e.g. XOR) into a single packet and transmits the coded packet as it is depicted in Fig.2. Clients can now obtain their lost packets by performing decoding operation using packets that they already have stored in their packet pools.

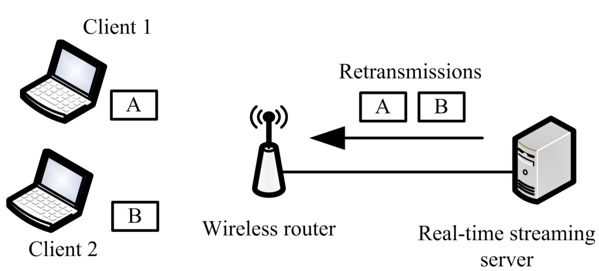

Fig.1. Retransmissions without NC

(E.g. Client 1 can decode coded packet using packet B from its pool and thus obtain packet A. In the same way Client 2 can obtain packet B).

By applying the NC mechanism and code packets together bandwidth reduction can be achieved. Even higher gains can be obtained in case with more clients.

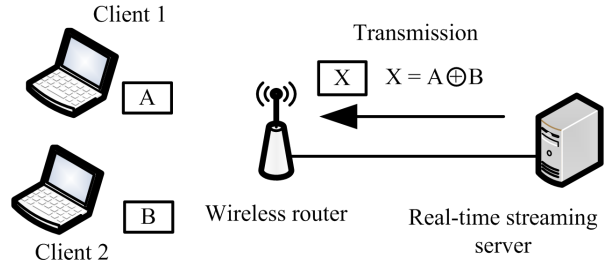

Fig.2. Transmission with NC

NC has already been successfully used to increase throughput as show in practical deployment in [17]. In [17] reliable and scalable live streaming solution based on wireless multicast with real-time network coding in hyper dense Wi-Fi spaces is presented. At the core of this approach is a timely delivery scheme that uses a minimum amount of feedback from the receivers to generate coded repair packets that are simultaneously useful to a large number of clients. This scheme uses packet loss estimation to be able to operate well with a very limited amount of feedback.

Moreover, in this scheme, all coded packets are linear combinations of original packets over the Galois field of size 2. However, they consider problem of satisfying different clients requests regarding the same flow while our approach considers problem of satisfying different clients regarding different flows with different approach of NC and feedbacks.

Feedback information used to design network coding solutions is presented also in practical deployment called Coding Opportunistically (COPE) [18]. They considered wireless mesh networks where feedback information is obtained by overhearing transmissions from the neighbouring nodes. In COPE opportunistic coding is presented where intermediate node looks for opportunities to encode as many packets as possible from different flows, ensuring that the recipients can decode their packets, so as to increase the information content of each broadcast transmission and therefore the throughput for data applications. Similar to COPE, in [19] authors presented Bearing Opportunistic Coding (BON) algorithm that can be used in static wireless mesh networks such as Wireless Sensor Networks and metropolitan Wi-Fi to increase network performance in terms of a higher throughput and a lower delay. In comparison to the COPE and the case where network coding is not used at all, BON increases the network goodput and decreases the delay. However, both COPE and BON were designed for regular non-real-time traffic and hence the reliability of data transfer was granted higher priority than the timely delivery of data.

For the particular case of wireless mesh network, [20] propose a network coding and scheduling scheme for transmitting several video streams over a wireless mesh network. Their key insight is that the transmission of video streams in a network coding-capable wireless network should be optimized not only for network throughput but also, and more importantly, for video quality. Both [18], [19] and [20] are designed for inter-session network coding, by combining different flows. They use a FIFO management for the requests of each receiver and compute a packet that must be decoded by the next hop of the head of the sender's output queue. In our proposal, there is no such restriction and all the requests from each receiver are taken into consideration, which significantly changes the nature and the complexity of the coding problem.

Further, approach in [21] considers intra-session network coding. The idea is to stream multimedia content from a single source to multiple receivers with direct or multihop connections to the source. Moreover in [21] retransmissions are not included but random linear codes that incorporate redundancy already when transmitting the packets for the first time.

Retransmission schemes based on NC are discussed in [22], [23] and [24]. Their key idea is to use the feedback of lost packets information to combine different lost packets with network coding to achieve retransmission. By broadcasting combined packets, their approaches can effectively save the number of transmissions and advance the efficiency of transmissions. However, those approaches have their own drawbacks and limitations making them impractical for real-time streaming applications.

We are interested in the transmission scheme that is integrated in the application layer and is adapted for video-streaming applications such as IPTV. In this paper we propose wireless broadcast transmission scheme for reliable video streaming service. We also evaluate its performance using practical Wi-Fi testbed, which comprise streaming server and multiple Wi-Fi clients.

The paper is structured as follows. Section 2 proposes concept of transmission scheme for reliable video streaming service using Network Coding. Section 3 gives overview about implementation details at server side and client side. Section 4 describes performance metrics and results. And finally, Section 5 concludes the paper.

2 Concept of video-streaming transmission scheme using NC

Consider the system setup illustrated in Fig.3. The system is intended for transferring multiple video streams to multiple clients. In depicted example streaming server is streaming two channels (streams A and B) to two clients (Client 1 and Client 2). Wireless router broadcasts the content over the wireless media. All clients listen to all transmissions and store all the received packets (even the ones not intended for them).

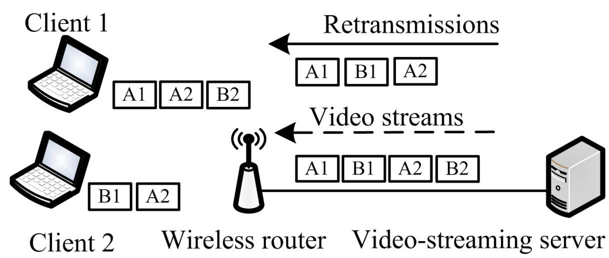

Fig.3. Traditional retransmission approach

System from Fig.4 has the same setup. Dissimilarity of the two systems is reflected in the way of implementation of transmission scheme. System from Fig.3 is using traditional retransmission approach where every lost packet is retransmitted separately (three packets are retransmitted) while system from Fig.4 is using transmission scheme with NC (only two packets are transmitted).

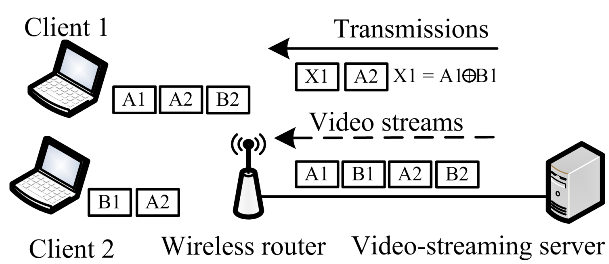

Fig.4. Transmission approach using NC

Given the described system from Fig.4 the design parameters for development for the transmission scheme using NC were:

1. Only packets that require retransmission will be transmitted
2. The algorithm transmits packets as late as possible, but still assuring the Quality of Service (QoS), i.e. after the Retransmission Timeout (RTO) or opportunistically.
3. All coded packets must be decodable by all clients

4. Packets that have not been received by any client will be sent out as they are, i.e. not coded.

The streaming server records the status of the received and not-received packet for each of the client. Information about the not received packets is provided through the Negative Acknowledgement (NACK) packets. Current status of the received packets at individual clients is represented in a transition table depicted in Table 1.a. Transition table has M rows that correspond to packets and N columns that correspond to the clients.

Table 1. Transition and coding tables

a) Transition table

	C_1	C_2
A_1	0	1
B_1	1	0
A_2	0	2
B_2	2	1

b) Coding table

	C_1	C_2
A_1	0	1
B_1	1	0
A_2	0	0

Packets with the lowest index have been represented in the table for the longest period. Three different states are noted in the transition table:

1. Packet received successfully is represented with state 1.
2. Not received packet intended for the corresponding client is represented with state 0 and means that packet has to be retransmitted.
3. Not received packet not intended for the corresponding client is represented with state 2 and means that packet does not have to be retransmitted for this client.

For the transmission and coding process transition table is transcript into the coding table (i.e. Table 1.b) where only packets that require retransmissions are represented. Here, two states are used to describe the status of the received packets on the clients. State 0 indicates that a packet has not been received and state 1 indicates that packet has been received on the client. Moreover, packets that do not need to be retransmitted are not presented in the coding table such as in packet B_2 case (packet has not been received by the C_1 but it does not require stream B so the packet has not have to be retransmitted).

NC algorithm for making decisions on which packets to code together is presented in Alg.1.

Algorithm 1. *NC Algorithm*

```
while (packets for retransmission)
    number_of_coded_packets = 1;
    if (P1 exceed RTO && P1 is codable)
       k = 1;
       coded_packets [k] = &P1;
       for (m = 2; m <= M; m++)
          packets_codable = 1;
          for (n = 1; n <= N ; n++)
             codable = coding table [1,n] +
                     + coding_table [m, n];
             if (codable < number_of_coded_packets)
                packets_codable = 0;
                break;
             end if
          end for
          if (packets_codable)
             number_of_coded_packets++;
             k ++;
             coded_packets [k] = &Pm
          end if
          if (number_of_coded_packets == N)
             break;
          end if
       end for
    end if
    if (number_of_coded_packets > 1)
       encode packets from coded_packets
       sent encoded_packet;
    else
       sent P1 uncoded;
    end if
       update coding_table;
end while
```

The algorithm is called after native packet is broadcasted. First it checks if there are available packet for retransmissions. In case of available packets it checks if the first packet from the coding table has exceeded RTO. If it has not, algorithm stops coding procedure. Otherwise, it takes the status of the first packet (that is first row) and checks if it is codable. Packet is considered codable, if it has been received at least by one of the clients. In the case that packet is not codable it is retransmitted as is. Otherwise, algorithm searches for coding opportunities with the rest of the packets intended for retransmission (even the ones that have not reached the RTO).

Algorithm looks to code with the first packet as many packets as possible, but prioritizes packets that have been waiting in the retransmission queue for a longer period. If it finds two packets codable, it will try to find the third one, than the fourth one etc. After algorithm is done, coded packet will be broadcasted to all clients. We assume that two packets are codable if coded packet can be decoded by all the clients. In practice, this means that all sums over the corresponding packets columns from the coding table are not less than number of coded packets N – 1.

Client can decode coded packet if it has previously received at least N – 1 native packets coded in the encoded packet.

Let us also use the example from the Table 1.b to explain the algorithm in practice. First, algorithm checks the status of the first packet A_1 using the coding table. Assume that RTO has expired for this packet. Packet is codable, since it was received by one of the clients. Second, algorithm checks status of the next packet B_1 with the help of the coding table and looks if the two are codable. These two packets are codable, as their coded packet can be decoded by all clients. Since the two packets is the maximum packets we can code together when only two clients are presented the algorithm stops the coding procedure. If there were more than two clients, the algorithm would look for other coding opportunities trying to code more packets together. Further, packets A_1 and B_1 are coded together and sent out in one transmission.

Last, packet A_2 is sent out as is, since there are no more packets left to code it to. In practice, such cases, where there is only one packet in the coding table left, are rarely encountered. Simultaneously, as packets are transmitted there are requests for retransmissions for new packets received. Given that, the algorithm would have new packets in the coding table to which A_2 would be matched for coding opportunities.

In a given example two transmissions are required with the use of NC in contrast to the traditional approach where server would need three retransmissions. In the cases with more clients, we would have more coding opportunities which means less transmissions and higher gains i.e. bandwidth reductions, A more extensive example is provided in the following. Assume the case with three clients watching three different IPTV streams. NACK packets for the first two packets of each stream have been received and the coding table describing the state is given in Table 2.

Table 2. Coding table with three clients

	C_1	C_2	C_3
A_1	0	1	1
B_1	0	0	0
C_1	1	1	0
A_2	0	0	1
B_2	1	0	1
C_2	1	1	0

For the simplicity reasons we do not include dynamics of the system into the account, thus not updating for the example the coding table, which size is in practice constantly changing. Algorithm first takes packet A_1 and checks the RTO and the codability of the packet. Since both conditions are meet, algorithm searches for coding opportunities that is looks for possible coding matches. Packet B_1 has not been received by any of the clients, so it is not codable. Next, packet C_1 is checked. Packets A_1 and C_1 are codable, hence C_1 is added to the coding array. Since upper limit for the maximal number of coding packets has not been reached, algorithm searches for new coding opportunities. Packet A_2 is codable, but does not meet conditions to be coded with A_1 and C_1 together. Next packet waiting for retransmission is packet B_2 which is codable and meets the selected conditions to be coded with packet A_1 and C_1. Packets A_1, C_1 and B_2 are coded together and sent out in one transmission. Algorithm stops coding procedure here, as three packet are the maximum that can be coded together. Each of the selected packets has not been received on only one of the clients, hence each client has enough information to decode the coded packet. When RTO is reached for packet B_1, the packet is sent out as is as the packet is not codable. When RTO is reached for A_2 the packet is taken for retransmission. Last packet in the coding table (C_2) is matched with the coding conditions. Since the packets A_2 and C_2 are codable and there are no more packets left in the coding table the two packets are coded together and sent out in one transmission. In a given example we needed three transmissions to recover lost packet. If NC would not be used, six retransmissions would be required to achieve the same, thus saving half of bandwidth.

3 Implementation details

In order to make proposed scheme deployable, several supporting mechanisms have been implemented on the client side, and on the server side and are explained in more detail below.

3.1 Client side

Client side runs two processes in parallel: signalization and decoding process depicted in Fig.5.

3.1.1 Signalization

Clients use NACK packets as the signalization mechanism to inform server which packets they have not received. NACK packets are sent to the server using unicast mechanism, as it is reliable mechanism and packets are unlikely to get lost. Two types of NACK packets are used:

- Hard NACK is a packet sent by the client that has lost packet for its video stream and it is marked with state 0 in the transition table.
- Soft NACK is a packet sent by the client that has lost packet for stream that is not intended for him and it is marked with state 2 in the transition table. Based on the NACK packets the transition and coding table are built that are needed for coding and transmission purposes.

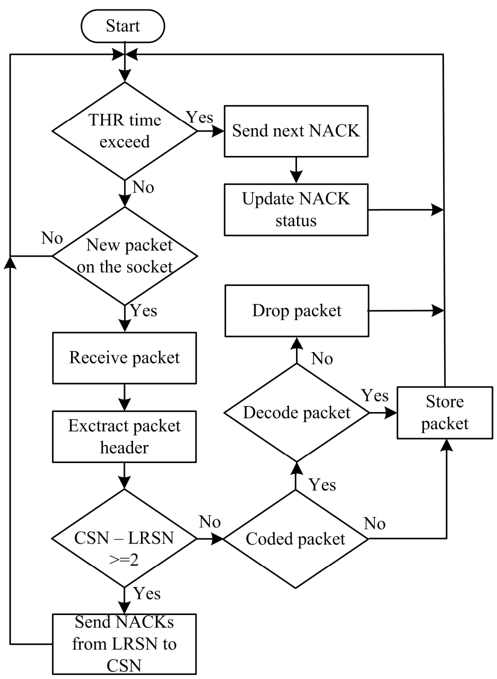

Fig.5. Signalization and decoding processes

When packet is received from the server, its header is extracted and examined. If there is a gap between sequence number of the current received packet (CSN) and last received packet (LRSN), the retransmission requests (i.e. NACKs) are sent to the server for packets that are not received. In order to reduce NACK bursts between lost packets we implemented forced retransmissions request. We added control timer that measures time when the last packet is received and compare it with predefined threshold time T_{THR} whose value is given Table 4. If this time is exceeded, next NACK is forcedly sent and the status of last NACK sent is updated.

The system state is assumed from the NACK messages collection. In case that NACK message is not received the system state is not recorded correctly. This results in an incorrect packet status in the transition table and possibly later on in the coding table. The coding process explained through Alg.1 thus make coding decisions that may lead to some of the clients being unable to decode the packet. We minimised the possible effects of the phenomena by setting the Wi-Fi unicast setting to three retransmissions.

3.1.2 Decoding

Client listens to all the transmissions, even the ones not intended for them and stores all packets in the packet pool, for decoding purposes.

With every received packet the decoding process checks if it is native or coded packet. If the packet is native its copy is stored into the packet pool for decoding purposes. It does so for every received native packet, as all the received packets are potentially needed for further decoding purposes. In the case when a coded packet is received, the process has to determine whether packet is decodable. If is not, the coded packet can not be decoded and it is simply dropped. If the node has enough information, it decodes the coded packet using previously stored packets with the XOR operations, thus gaining a native packet that has not been received before.

3.3 Server side

At the server side two processes run in parallel: handling NACK packets and broadcasting native/coded packets as depicted in Fig.6 and Fig.7, respectively.

3.3.1 Handling NACK packets

For every received NACK additional control is added in terms of delay. NACK packets that arrive too late, are dropped because they are considered useless in the coding process. When NACK packet arrives the process determines delay T_{ND} as depicted in Fig.6. T_{ND} is measured for *i -th* packet as the difference between time when NACK is received and time when native packet for corresponding NACK is sent.

Moreover, T_{ND} is compared to defined threshold time T_{THR} whose value is given in Table 4. If T_{ND} exceeds T_{HR} that means that NACK has delayed and such a NACK is no longer useful, therefore it is dropped. Otherwise, new NACK is added to the transition table and updates in transition and coding table are made.

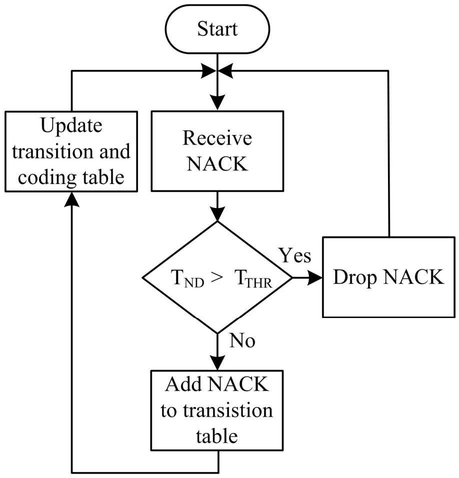

Fig.6. Handling NACK packets process

3.3.2 Broadcasting native/coded packets
Broadcasting native/coded packet process is depicted in Fig.7. After native packet is broadcasted process start to searching for coding opportunities using previously described algorithm in Alg.1. If requirements from Alg.1 are met, i.e. coding opportunity is found, native packets are encoded together using XOR operation and broadcasted as a one coded packet to all clients. Otherwise, if no coding opportunity is found first packet from coding table is retransmitted as is, and process continue further from beginning i.e. broadcasting native packet.

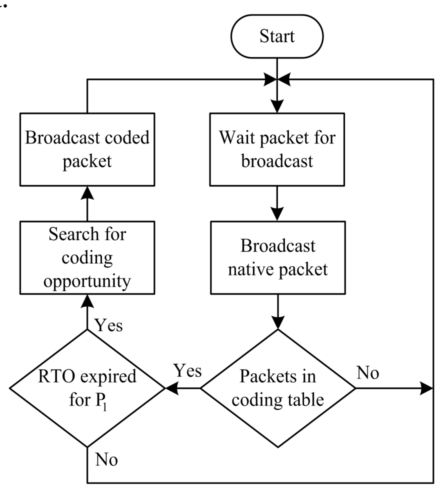

Fig.7. Broadcasting native/coded packet process

3.4 Overhead estimation
NC brings an additional overhead into network in terms of additional packets and headers added as depicted in Fig.8. The additional overheads can be compensated with the increased coding opportunities.

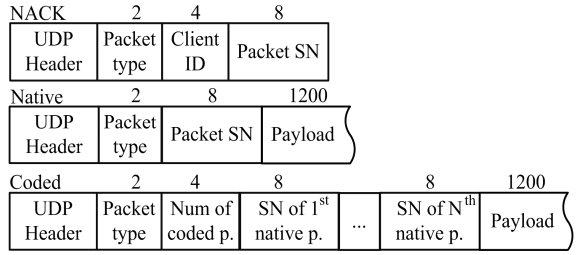

Fig.8. Additional headers

Additional headers are added for three types of packets:

- NACK packet has three types of additional headers: *Packet type*, *Client ID* and *Packet Sequence Number*. Packet type is used to indicate which type of packet is sent, i.e. NACK, native or coded packet. Moreover it defines type of NACK i.e. soft or hard. *Client ID* is used to know from which client is NACK received. *Packet Sequence Number* is used for the identification of packet.
- Native packet also has three types of additional headers: *Packet type*, *Packet Sequence Number* and *Payload*. First two headers are the same as in the preceding case. *Payload* filed represents media content added to every packet.
- Additional headers added for the coded packet depend on the number of native packets that are coded together. As in the preceding cases, *Packet type* represents type of the packet i.e. coded packet. Further, number of packets that are coded together is represented with the field *Number of Coded Packets*. *Packet Sequence Number* represents sequence number of native packet inside of a coded packet. Number of these fields is not larger than total number of clients. Coded packet just like native packet has *Payload* but in this case it is composed of XOR-ed payloads of every native packet.

4 Performance evaluation

4.1 Experimental setup
Practical wireless testbed deployed for proposed scheme consist of streaming server, wireless router and 7 clients (i.e. laptops). Clients were arranged

arbitrarily around wireless router (within the same room) and their positions were fixed throughout the entire experiment. Wireless router used was Linksys WRT54GL with DD-WRT firmware and fixed wireless parameters as in Table 3.

Server was connected to the wireless router via Ethernet interface while clients (laptops of different brands) were connected via WLAN interface. Server streamed one UDP stream in total 608 MB of data at 1.525 Mbps rate.

Table 3. Wireless router configuration

Parameter	Value	Comment
Wireless Mode	AP	Access Point
Wireless Network Mode	G-Only	802.11 g
Wireless Channel	5	5th channel
Basic Rate	All	Compatibility with devices
Transmission Fixed Rate	Auto	Auto

Packets size was constant (i.e. 1210 bytes). To all outgoing packets transmission scheme related headers were added. Inter-arrival time between packets was constant (i.e. 0.3 s).

Results were gathered through a one hour experiment and are presented for the time interval between 1000th and 1500th second to observe steady state condition.

Parameters used for the proposed testbed at server and client side are presented in the Table 4.

Table 4. Parameters as used in testbed at server and client side

	Para-meter	Value	Parameter Full Name
Server side	RTO	300 ms	Retransmission timeout
	T_i	6 ms	Packet inter-arrival time
	T_{THR}	300 ms	Threshold time
Client side	T_i	6 ms	Packet inter-arrival time
	T_{THR}	150 ms	Threshold time

4.2 Performance metrics

Several performance metrics were used to evaluate the performance of the proposed scheme:

1. Number of retransmitted packets N_R is the number of native packets that are retransmitted with no coding while number of transmitted packets N_T is the number of packets that are transmitted using network coding.

2. N_N is the number of native packets sent.

3. Bandwidth reduction B_R is calculated as the proportion of difference of the N_R and N_T packets to the sum of N_N and N_R packets. We use B_R to show how much bandwidth in percentage we reduced by using the proposed scheme.

$$B_R = \frac{N_R - N_T}{N_N + N_R} * 100\% \quad (1)$$

4. Delivery probability of *i-th* client is defined as the proportion of successfully delivered packets N_{Di} to the total packets that were retransmitted N_{TOi} in percentages.

$$DP_i = \frac{N_{D_i}}{N_{TO_i}} * 100\% \quad (2)$$

Delivery probability *DP* is average of delivery probabilities over single clients (N_C).

$$DP = \frac{1}{N_C} \sum_{i=1}^{N_C} DP_i \quad (3)$$

5. In the evaluated scenarios we primarily observed gain G_i which is on *i-th* client defined as the proportion of transmitted packets N_{Ti} to the retransmitted packets N_{Ri}.

$$G_i = 1 - \frac{N_{T\,i}}{N_{R_i}} * 100\% \quad (4)$$

Gain *G* is average of gains of single clients (N_C).

$$G = \frac{1}{N_C} \sum_{i=1}^{N_C} G_i \quad (5)$$

6. With coding table size *CTS* we refer to the number of packet statuses on the clients. It is sampled periodically.

All the presented results were observed over the sample period T_S (i.e. *5 s*).

4.3 Results

Bandwidth reduction B_R of proposed scheme is shown in Fig.9. We can see that during the given time interval bandwidth reduction is between *7 %* and *14.75 %*.

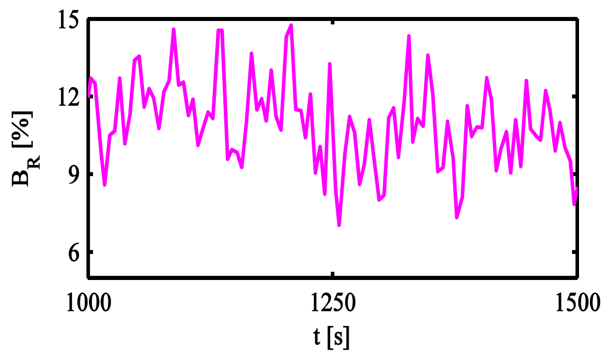

Fig.9. Bandwidth reduction B_R [%]

Fig.10 shows delivery probabilities of three different clients during the observed time interval. Clients on graph were selected based on worst, medium and best delivery probabilities. We wanted to observe how these delivery probabilities of individual clients are changing during the time and how they affect another parameters which is explained further in Fig.11. It shows in more detail the dependencies between average delivery probability (*DP)*, coding table size (*CTS)* and gain (*G)*. As the *DP* decreases coding table size *CTS* increases i.e. number of packets in the coding table increases.

This is because more NACKs are sent by clients. Moreover as the *DP* and *CTS* decreases, so does the coding gain *G*. This is because when *DP* is low there are more occasions where the same packet is not received by multiple clients. This affects the number of coding opportunities i.e. there are fewer coding opportunities, and consequently the *G* is lower.

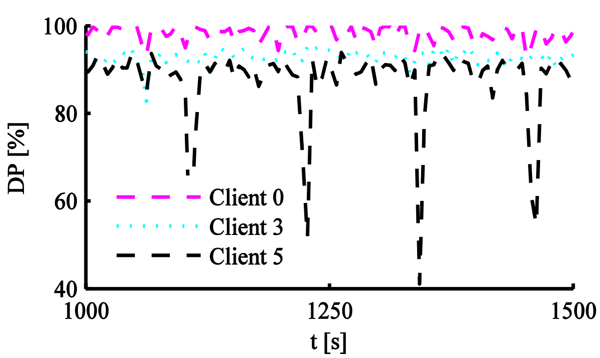

Fig.10. Delivery probabilities of three selected clients [%]

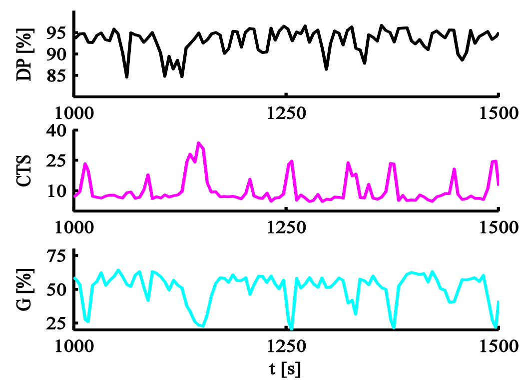

Fig.11. Dependencies of delivery probability *DP* [%], coding table size *CTS* and gain *G* [%]

In the following, we will also show relations between average coded packets/s and number of retransmissions/s depicted in Fig.12 and Fig.13, respectively. Average coded packet is the number of native packets coded together per second.

If we compare the two figures, paying the attention to the time interval from 1050th to 1150th second, we can conclude that if more native packets are coded together, fewer packets need to be retransmitted. I.e. for *t = 1052 s* number of average coded packets is *2.77* while number of retransmissions is *80*. Contrarily for *t = 1088 s* number of average coded packets is *1.71* while number of retransmissions is *262*.

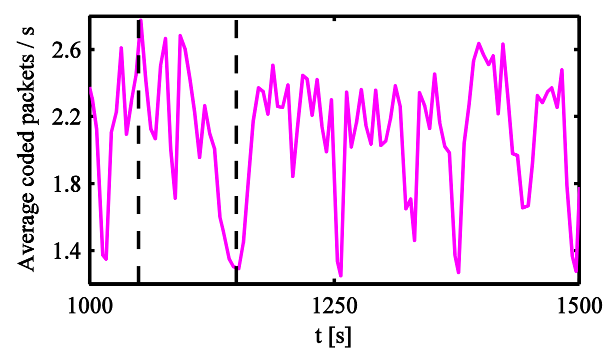

Fig.12. Average number of coded packets/s

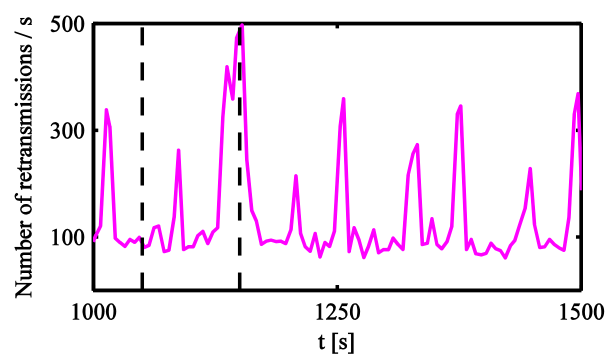

Fig.13. Number of retransmissions/s

In the next step we investigate *DP* versus the *G* as depicted in Fig.14. *DP* values range between *78.61 %* and *96.75 %* while the corresponding values for *G* range between *16.44 %* and *65. 89 %*. The majority of points from graph are located in the area with high *DP* and high *G*. There are several points that are far away from the cluster.

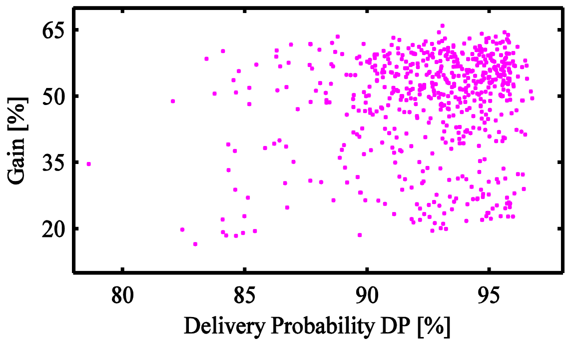

Fig.14. DP with respect to Gain

In the Fig.15 we investigate the percentage of number of all native packets that are coded together. Result is presented as histogram where x-axis represent number of native packets coded together (*2*, *3*, *4*, *5*) and y-axis represents percentage of native packets encoded together in coded packet (*31.9 %*, *15.4 %*, *5.8 %*, *0.1 %*), respectively.

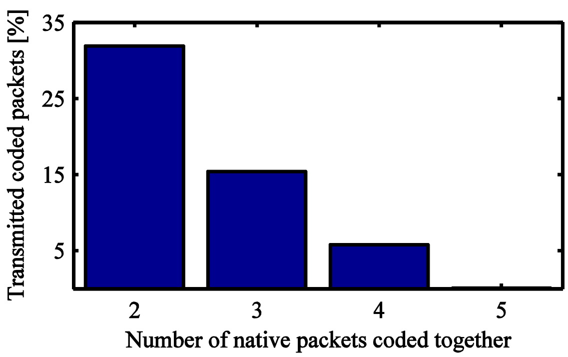

Fig.15. Histogram of transmitted coded packets [%]

Furthermore, the coding and decoding require additional delay. Still, we expect that the additional delay introduced by our scheme will not affect the Quality of Experience (QoE) on the client side as all the operations will be carried out within the buffer time of the stream which is in our case 0.3 s (RTO).

5 Conclusion

In this paper we proposed wireless broadcast transmission scheme for reliable video streaming service. The main contribution of this work is the use of a XOR-ed transmission scheme based on the information gathered from explicit NACK messages, sent by remote clients. From this information, the server generates different combinations of the transmitted packets, aiming that all of them are to be retrieved in every intended destination. Using the proposed approach, the overall number of transmissions is reduced and the wireless link is less utilized.

We showed using wireless testbed that our solution compared to no NC retransmission approach reduces the bandwidth up to 15 %. This bandwidth reduction in practice is important for ISPs as they can offer services to higher number of clients using the same equipment or, alternatively, they can introduce new services.

With the proposed scheme high gains can be obtained for different delivery probabilities. Higher gains can be obtained with higher number of clients, which is usually the case in the real environment, as we have more different streams. With more clients, more different packets can be coded together which results in more coding opportunities and fewer transmissions.

The presented approach can be implemented in the wireless broadcast network when wireless technology is used for the last mile access. Our scheme introduces bandwidth reduction for video-streaming applications such as IPTV.

References:

[1] H.-H. Yang, J.-C. Yu, Y.-D. Wang, H.-J. Yang, H.-C. Lin, and W.-Y. Dzan, "An Implementation of Designing Media Streaming System for Live Broadcast," *WSEAS Transactions on Communications,* vol. 9, pp. 226-235 2010.

[2] O. S. Badarneh and M. Kadoch, "Video Multicast Over Wireless Ad Hoc Networks," *WSEAS Transactions on Communications,* vol. 11, 2012.

[3] A. Roy, M. I. Islam, and M. R. Amin, "Performance Evaluation of Voice-Data Integrated Traffic in IEEE 802.11 and IEEE 802.16e WLAN," *WSEAS Transactions on Communications,* vol. 12, p. 352, 2013.

[4] C. Fragouli, J.-Y. L. Boudec, and J. Widmer, "Network coding: an instant primer," *ACM SIGCOMM Computer Communication Review,* vol. 36, pp. 63-68, 2006.

[5] S. Battilotti, F. D. Priscoli, C. G. Giorgi, A. Pietrabissa, S. Monaco, M. Panfili, *et al.*,

"Approaches for Future Internet architecture design and Quality of Experience (QoE) Control " *WSEAS Transactions on Communications*, vol. 14, pp. 62-73, 2015.
[6] B.-B. Chen, H.-C. Chen, and D.-H. Guo, "Adaptive Wireless Networks QoS Evaluation Analysis Through Enhanced Parameters Tuning Algorithms," *WSEAS Transactions on Communications*, vol. 13, pp. 217-228, 2014.
[7] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow " *Information Theory, IEEE Transactions on* vol. 46, pp. 1204 - 1216, 2000.
[8] T. Matsuda, T. Noguchi, and T. Takine, "Survey of Network Coding and Its Applications," *IEICE Transactions on Communications*, vol. E94-B, pp. 698-717, 2011.
[9] D. Niu and B. Li, "Asymptotic optimality of randomized peer-to-peer broadcast with network coding," presented at the INFOCOM, 2011 Proceedings IEEE 2011.
[10] C. Jing, S. Xue-peng, and L. Fang, "Effective Retransmission in Network Coding for TCP," *International Journal of Computers Communications & Control*, vol. 6, pp. 53-62, 2011.
[11] S. Haoliang, L. Lixiang, and H. Xiaohui, "A Network Coding based DTN Convergence Layer Reliable Transport Mechanism over InterPlaNetary Networks," *International Journal of Computers, Communications & Control*, vol. 6, pp. 236-245, 2011.
[12] M. Osama and A. Shawish, "Enhanced Network Coding Scheme for Efficient Multicasting in Ad-Hoc Networks," presented at the 7th WSEAS European Computing Conference (ECC 2013), Dubrovnik, Croatia, 2013.
[13] P. Chaikijwatana and T. Tachibana, "VCG auction-based bandwidth allocation with network coding in wireless networks," in *The 10th WSEAS International Conference on Applied Computer and Applied Computational Science (ACACOS '11)*, Venice, Italy, pp. 104-109.
[14] C. Tata and M. Kadoch, "Secure Network Coding based Data Splitting for Public Safety D2D communications over LTE Heterogeneous Networks " presented at the 8th International Conference on Communications and Information Technology (CIT '14), Tenerife, Spain, 2014.
[15] C. Yi-Cheng and H. Ya-Yi, "Adaptive network coding scheme for TCP over wireless sensor networks " *International Journal of Computers, Communications & Control*, vol. 8, pp. 800-811, 2013.
[16] K. Yasami, A. Razi, and A. Abedi, "Analysis of Channel Estimation Error in Physical Layer Network Coding," *Communications Letters, IEEE*, vol. 15, pp. 1029 - 1031, 2011.
[17] D. Ferreira, R. A. Costa, and J. Barros, "Real-Time Network Coding for Live Streaming in Hyper-Dense WiFi Spaces," *Selected Areas in Communications, IEEE Journal on*, vol. 32, pp. 773 - 781, 2014.
[18] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding," *IEEE/ACM Transactions on networking*, vol. 16, pp. 497 - 510, 2008.
[19] K. Alič, E. Pertovt, and A. Švigelj, "Bearing-opportunistic network coding," *International Journal of Computers, Communications & Control*, vol. 10, pp. 154-164, 2015.
[20] H. Seferoglu and A. Markopoulou, "Video-Aware Opportunistic Network Coding over Wireless Networks," *IEEE Journal On Selected Areas In Communications*, vol. 27, pp. 713 - 728, June 2009.
[21] P. Vingelmann, F. H. P. Fitzek, M. V. Pedersen, J. Heide, and H. Charaf, "Synchronized multimedia streaming on the iphone platform with network coding," *Communications Magazine, IEEE*, vol. 49, pp. 126 - 132, 2011.
[22] W. Fang, F. Liu, Z. Liu, L. Shu, and S. Nishio, "Reliable Broadcast Transmission in Wireless Networks Based on Network Coding," presented at the Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on Shangai, 2011.
[23] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Wireless Broadcast Using Network Coding," *Vehicular Technology, IEEE Transactions on*, vol. 58, pp. 914 - 925 2009.
[24] X. Xiao, Y. Lu-Ming, W. Wei-Ping, and Z. Shuai, "A Wireless Broadcasting Retransmission Approach Based on Network Coding," presented at the Circuits and Systems for Communications, 2008. ICCSC 2008. 4th IEEE International Conference on Shangai, 2008.