We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Network Coding-Assisted Retransmission Scheme for Video-Streaming Services over Wireless Access Networks

Aleš Švigelj and Melisa Junuzović

Additional information is available at the end of the chapter

Abstract

Video-streaming services, such as Internet protocol television, promising the delivery of multimedia contents over wireless access networks to clients whenever and wherever, are becoming more and more popular. However, scarce radio resources, lossy characteristics of wireless links and high bandwidth demands pose the never-ending challenges for provisioning of real-time streaming services over wireless networks in a timely and reliable manner. Furthermore, a wireless channel may suffer from interference and multipath fading, which may cause random packet losses. In addition, wireless link layer does not provide a retransmission mechanism for multicast/broadcast traffic. This would significantly impact the clients' quality of experience of streaming services. Traditional unicast retransmission solutions improve client's quality, at the bandwidth expense, because every lost packet must be retransmitted separately. This chapter presents and practically evaluates a retransmission scheme for video-streaming services over last mile wireless networks. It is based on network coding techniques that increase the overall performance by means of reducing the number of physical transmissions, in comparison to traditional unicast retransmission approach, resulting in reduced bandwidth consumption. Thus, the Internet service providers can increase the number of clients over the same infrastructure or, alternatively, offer more services to the clients.

Keywords: retransmission, wireless network, broadcast, network coding, streaming, IPTV

1. Introduction

The provision and efficient delivery of high-quality real-time streaming services, such as Internet protocol television (IPTV) over wireless networks (e.g., Wi-Fi) in the last mile, introduces never-ending challenges. Easy and fast deployment and relatively low deployment costs

of Wireless IEEE 802.11 networks, where one antenna covers several different clients/groups of clients, stimulate the usage of Wi-Fi technology in underdeveloped countries and rural areas. However, the bottleneck in the content delivery is due to spectrum limitations in wireless medium shifted to the last mile. Thus for ISPs, it is of paramount importance to increase the last mile access performance as in such cases, the bandwidth reduction means better services for existing clients, or alternatively, additional clients over the same infrastructure. Thus in this chapter, we are addressing the efficient and quality delivery of video-streaming services such as IPTV using the IEEE 802.11 wireless broadcast network in the last mile [1].

By using IEEE 802.11 wireless network in the last mile, the video content can be delivered using unicast or broadcast/multicast mechanism. Due to the fact that unicast mechanism, which delivers the content to clients individually and supports retransmissions (and back off) that assure reliable content delivery, is a preferred solution in currently deployed systems. On the other hand, multicasting can be used instead of unicasting in order to reduce the bandwidth. The bandwidth can be significantly reduced in particular, when multiple clients watch the same video stream, as the server, instead of sending multiple packets to different clients, has to send only one packet. All broadcast packets are delivered to all clients only under ideal wireless channel conditions. However, the packet losses frequently occur in the practical wireless environment; thus, some sort of retransmission mechanism is necessary to ensure reliability. 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 in network coding [2], numerous papers appeared on this subject and significant progress has been made in applying NC to different networks. NC has already been successfully used to increase throughput as shown in practical deployment in [3], where a reliable and scalable live streaming solution is presented. It is based on wireless multicast with real-time network coding in hyper-dense Wi-Fi spaces. A timely delivery scheme that uses a minimum amount of feedback from the clients is at the core of this approach. It generates coded repair packets, which are useful to a large number of clients. Packet loss estimation is used in the scheme, which is able to operate well with a very limited amount of feedback. Besides, all coded packets are linear combinations of original packets over the Galois field of size 2. Nevertheless, they are focusing on satisfying different clients' requests regarding the same flow, while we are focusing on satisfying different clients regarding different flows, using different approach of NC and feedbacks.

In [4], for the particular case of wireless mesh network, a network coding and scheduling scheme for transmitting several video streams over a wireless mesh network is proposed. Their main outcome is that in a network coding capable wireless network, the transmission of video streams should be optimized also, and even more important than for network throughput, for video quality. All approaches mentioned in [4–6] are designed for intersession network coding, by combining different flows. They compute a packet that must be decoded by the next hop of the head of the sender's output queue, based on a first in first out (FIFO) management for the requests of each client. However, there is no such restriction in our proposal and all client requests are taken into consideration, which significantly changes the complexity and the nature of the coding problem.

Retransmission schemes based on NC are discussed in [7–9]. Their key idea, in order to combine different lost packets with network coding to achieve retransmission, is to use the feedback of lost packets information. Their approaches by broadcasting combined packets effectively save the number of transmissions and advance the efficiency of transmissions. However, in practical real-time streaming applications, those approaches have their own drawbacks and limitations.

Depending on the NC application, the implementation affects different open systems interconnection (OSI) layers. In this chapter, we propose transmission scheme that is integrated in the application layer and is adapted for video-streaming applications such as IPTV. It can be used for increasing network throughput of mobile and static wireless networks. We also evaluate its performance using practical Wi-Fi test bed that comprises a streaming server and multiple Wi-Fi clients.

The rest of the chapter is organized as follows. In this section, we introduced the video-streaming challenges in wireless medium using IPTV. Section 2 discusses the packet loss recovery techniques in streaming wireless networks. Section 3 proposes concept of transmission scheme for reliable video-streaming service using network coding and gives an overview about implementation details. Practical evaluation of the proposed solutions is described in Section 4, and finally, Section 5 concludes the chapter.

1.1. Wireless medium challenges

The main advantages of wireless networks over wired counterparts include ease of deployment and mobile clients support. However, the unique characteristics of video data amplify the difficulty of video transmission and thus impose a number of challenges. Some of the main challenges of wireless networking and their impact on video communication are discussed and highlighted in the following.

- Shadowing and multipath fading: they are common wireless effects. Shadowing is caused by obstacles between the transmitter and the receiver that attenuate signal power through absorption, reflection, scattering, and diffraction. It occurs over a relatively large time scale. On the other hand, multipath fading is due to multipath propagation: different path signals are added constructively or destructively. Multipath fading results in rapid fluctuation of signal amplitude within the order of a wavelength. The presence of shadowing and multipath fading results in time-varying channel conditions and location-dependent packet erasures. They cause burst errors in the form of multiple lost packets.

- Limited bandwidth: wireless networks are limited in capacity. Although, the 802.11 products are advertised as having a high data rate. Still, "protection" mechanisms such as binary exponential back off, rate adaptation, and protocol overheads cut the throughput by 50% or more. Moreover, due to the shared nature of the wireless medium, the actual bandwidth available to individual clients can even be much lower, posing a great obstacle for providing high bandwidth video services.

- Interference: wireless medium is shared among multiple clients; thus signals, which arrive at a receiver from other concurrent transmissions, although attenuated, cause the

interference for the receiver. It is a common effect in WLANs because they operate in the unlicensed 2.4/5 GHz industrial, scientific, and medical (ISM) frequency band. Interference affects the quality of a wireless link, and consequently, its error rate and achievable capacity.

1.2. Video-streaming challenges

Video services and applications have become the driving force in the development and widespread deployment of wireless broadband access technologies and high speed local area networks. Convenient and cable-free connectivity can be achieved by using wireless local area networking (WLAN) technologies. Using the wireless networks, it also can provide mobility to portable TVs.

As conventional service architectures, network structures and protocols have not been designed to consider the high data rate and real-time transmission requirements of digital video, they lack to provide a robust medium distribution.

Video-streaming applications have distinctive QoS requirements, such as high bandwidth requirement, delay sensitiveness, and loss tolerance. We list the challenging QoS issues as follows:

- Bandwidth: transmission of video sequences typically has a minimum bandwidth requirement in order to achieve acceptable presentation quality. Therefore, supporting the delivery of video over time-varying wireless links could be very unreliable. The challenge lies in keeping the quality degradation to a level that is hardly noticeable or tolerable while utilizing the wireless resources efficiently.

- Delay: in contrast to data transmission, which is usually not subjected to strict delay constraints, video-streaming requires bounded end-to-end delay. Each video frame needs to arrive at the client to be decoded and displayed, before its play out deadline. Otherwise, it is useless. If the video packet does not arrive on time, the play out process will have to be temporally paused, which is annoying to human eyes and deteriorates the overall streaming quality. Consequently, video-streaming applications are usually known to be very sensitive to delay.

- Packet loss: video-streaming technology is tolerant to a certain level of loss, since the visual quality will still be acceptable if the packet loss ratio is kept below a certain threshold. However, loss of packets can potentially make the presentation displeasing to human eyes, especially when some of the key video frames are lost, which could make the presentation impossible.

- Unreliability: Wi-Fi does not provide a retransmission mechanism for broadcast/multicast transmissions (with multiple clients). Wi-Fi multicast traffic could normally experience a packet loss rate of 5%. As a loss of even a single video packet can result as an error that propagates for many video frames, this is a serious problem. Thus, it is quite normal that multicast video applications fail completely when they operate on a Wi-Fi network although they work fine on a wired network.

1.3. IPTV

Internet protocol television (IPTV) is a system that is used to deliver the Internet television services across the Internet protocol (IP) infrastructure. IPTV integrates voice, video, and data delivery into a single IP framework, to enable interactive broadcasting services to the subscribers. It promises significant advantages for both service providers and clients.

In the past few years, IPTV has gained an unremarkable growth rate. IPTV services are becoming very popular among telecommunication companies, most of all because they can provide television programs anytime and anywhere. IPTV can support broadcast and unicast services, such as broadcasting of TV channels, Video on demand (VoD) or time-shifted television. Unlike conventional broadcasting systems, IPTV broadcasts will not be restricted by the limited number of channels in the broadcast/radio spectrum.

Besides, IPTV will provide its clients with the opportunity to access and interact with a wide variety of high quality on demand video content over the Internet. Since IPTV is considered as a real-time broadcast service over the Internet, the success of the IPTV service depends on the quality of experience (QoE) perceived by the end clients.

The IPTV system has to support a wide range of transmission channel characteristics (i.e., varying packet loss rates and transmission delays) between the content providers and the end clients. The characteristics of video traffic as well as the high-quality requirements of IPTV broadcast impose strict requirements on transmission delay. IPTV framework has to provide mechanisms to satisfy the stringent delay, jitter, and bandwidth requirements over lossy transmission channels.

The architecture of IPTV is composed of five main parts as depicted in **Figure 1**: (i) head network, (ii) core network, (iii) distribution network, (iv) access network, and (v) customer network.

Transport network includes the core network, distribution network, and access network. IPTV service provider transports the IPTV streams to client end by using all entities of the transport
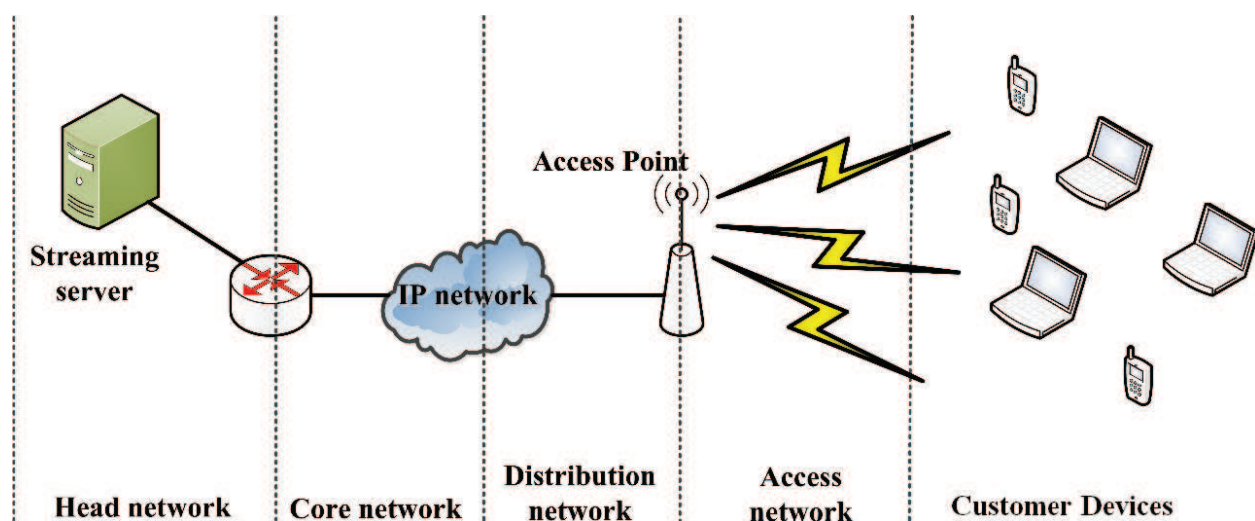

Figure 1. IPTV architecture.

network. Among all these entities, major bandwidth restricting entity in transport network is the access point (AP) because of limited buffer size of queues. To avoid bandwidth restriction, wired access links such as cable and digital subscriber line (DSL) are preferred by IPTV service providers.

Using WLAN for IPTV distribution improves the mobility and flexibility of IPTV network deployment; however, it has its challenges as described in previous sections and it can impact the clients' quality of experience when watching an IPTV program.

Error correction schemes such as forward error correction (FEC) and/or retransmissions can be used to improve reliability of an IPTV program. Both of them are explained in more detail below.

2. Packet loss recovery

Considering the earlier mentioned strict quality requirements of video, strong error recovery techniques are required to protect the content that is delivered to end clients and recover from the occasional packet losses. As it can be seen from the previous examples, there are two possibilities either to prevent packet loss or, alternatively, when packet loss does happen, reduce the noticeable effects of packet loss or restore the missing packets.

In other words, one wants to provide a mechanism to recover from packet loss or to have an error resiliency mechanism that reduces errors due to packet loss. Numerous techniques for providing error resiliency against packet loss can be divided in two categories:

• Techniques that provide recovery for packet loss.

• Techniques that try to reduce the impact or effect of the packet loss.

Two different techniques aimed to recover packet losses during the delivery of IPTV services are FEC and retransmission. The FEC approach operates by adding redundant information to the data at the server, while the retransmission approach recovers from packet losses by requesting retransmission from the server.

2.1. Forward error correction (FEC)

An FEC-based error recovery protocol uses redundant information to allow the client to correct packet losses. With this redundant information, the clients can recover from packet losses nearby the client. The amount of data that can be reconstructed is determined by the amount of loss and on the amount of redundant data. As an FEC-based recovery mechanism is suitable in networks that only allow unidirectional traffic (e.g., satellite broadcast) or environments where the latency from the client to the server is relatively high (e.g., cellular networks), it does not require any feedback from the server to the client. It can be applied from the physical layer up to the application layer according to the OSI reference model. For IPTV streaming applications, FEC can be offered at network layer, transport layer, or application layer FEC. A disadvantage of a FEC protection mechanism is that the applied FEC protection

scheme might be insufficient (too weak) for certain clients, while at the same time can be superfluous for some other clients, thus being a waste of bandwidth.

2.2. Retransmissions

Packet retransmission is the technique of retransmitting packets that are considered lost. A packet retransmission requires communication between the client and server, as the client needs to explicitly or implicitly request the packet retransmissions from server. The transmission control protocol (TCP) is a well-known protocol using packet retransmission. Bi-directional communication is required at packet retransmission mechanism in order to allow the client to indicate packet loss and to identify which packet needs to be retransmitted. It is typically achieved by applying sequence numbers. To indicate the loss of packets, two types of messages can be used:

- Acknowledgment (ACK) messages can be used to implicitly indicate the loss of a packet by acknowledging the reception of one packet or a sequence of packets. As the missing packet will never be acknowledged, these messages can then be used to implicitly determine packet loss (e.g., TCP uses ACK messages).

- Negative acknowledgment (NACK) messages are used to explicitly indicate that one or more packets were not received. As feedback from client to server is kept to a minimum, NACK messages are used in constrained networks with many clients per server.

A packet retransmission mechanism is adaptive to variable network conditions. When there is no loss in the network, there will be no packet retransmission, hence no additional required bandwidth. Because packet retransmission mechanisms introduce delay, they are only suitable for applications with nonstrict delay requirements.

Using packet retransmissions for (live) IPTV services has also the drawbacks as IPTV streams are not likely to get adapted to congestion. Thus, if congestion occurs during transport, the packet retransmissions may contribute to the network congestion if the packet retransmission is enabled, thus further lowering bandwidth available for IPTV stream delivery.

2.3. Network coding

In the past few years, network coding (NC) has become popular in both wired and wireless networks as a mechanism for increasing the network throughput. Proposed by Ahlswede et al. in [2], it is a technique for increasing the throughput of the network by encoding multiple packets into the same packet, either from the same or from different streams.

Using the NC, the broadcast nature of the wireless medium is exploited. In the wireless physical medium, all data transmitted by the source is in fact inherently broadcasted to all destinations, even when it is not intended. For no extra cost, other receivers will receive packets not addressed to them. In the case where a packet is lost at the receiver, other receivers may receive it successfully. Keeping those packets by receivers for some time enables NC to reduce the number of packets to be transmitted.

NC reduces the number of transmissions by replacing the conventional store-and-forward paradigm in relaying scenarios with a process-and-forward approach. Instead of forwarding packets in the same form as they are received, the intermediate nodes combine the received outgoing packets using an algebraic function such as XOR. All nodes store all the received packets for possible packet decoding purposes. As a result, a typical NC mechanism is composed of two main operations [6, 10]:

- Coding outgoing packets on intermediate nodes.

- Decoding incoming packets upon their reception on nodes.

The main concept of NC is presented through examples and is also sketched in **Figure 2**. Let us consider the situation where destination node D_1 can only hear transmissions from relay node R and source S1. Similarly, destination D_2 can only hear transmissions from source S_2 and relay node R. S1 has a packet p_1 for D_2 and S_2 has a packet p_2 for D_1. S_1 and S_2 send their packets separately to R, using two transmissions. In the first example (i.e., **Figure 2** (without NC)), the relay node R cannot transmit two packets at the same time. Thus, again two transmissions are used to transmit packets p_1 and p_2.

Referring to the **Figure 2** (with NC), it shows the case of the same network but using NC. The relay node R has enough processing power to code two packets, that is, the R node performs a coding operation, for example, linear operation over the p_1 and p_2; thus creating a coded packet of the same size as largest of p1 and p2. Node R can now by sending only one packet forward both p_1 and p_2 to its destination nodes. D_1 can decode the coded packet as it is familiar with the content of p_1 and similarly D_2 can also decode coded packet as it knows the content of p_2. Instead of using four transmissions, the broadcast nature of the media has been embraced and only three transmissions were needed.

The slight overhead that can be seen here is the buffer space requirement as well as additional processing in encoding and decoding the packet. With the advancement in solid states like high speed processors, and high speed and large capacity memories, these overheads are well taken care of. The bandwidth is limited especially in wireless networks. Using NC, we can efficiently utilize the bandwidth of the network.
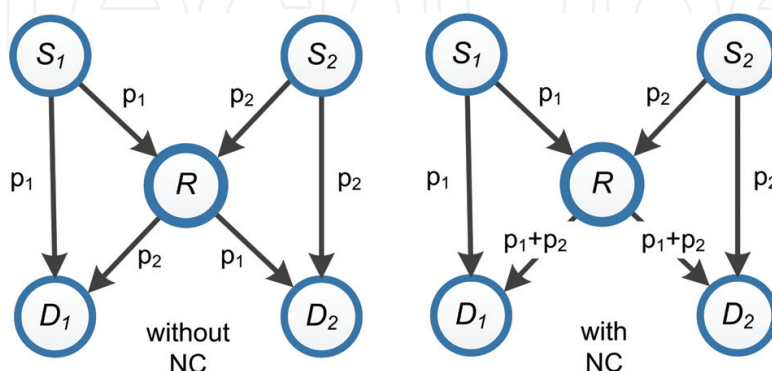

Figure 2. Network coding example.

3. Network coding-assisted retransmission scheme

3.1. Concept

Consider the system setup illustrated in **Figure 3**. 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 *Client* 1 and *Client* 2, respectively. Packets that are not successfully delivered in the first transmission are retransmitted. Clients signal the server on the not received packets using signalization, for example, in our case with signalization.

Figure 4 shows similar system. The difference is in the retransmission concept. In the first system, every packet that has not been received is retransmitted individually (e.g., five retransmissions). The second system codes multiple packets together and generally uses fewer transmissions (e.g., three).

For the system depicted in **Figure 4**, the design parameters were selected as follows:

- Only packets that require retransmission are transmitted.

- Packets are transmitted as late as possible, that is, after the retransmission timeout (RTO) or opportunistically.
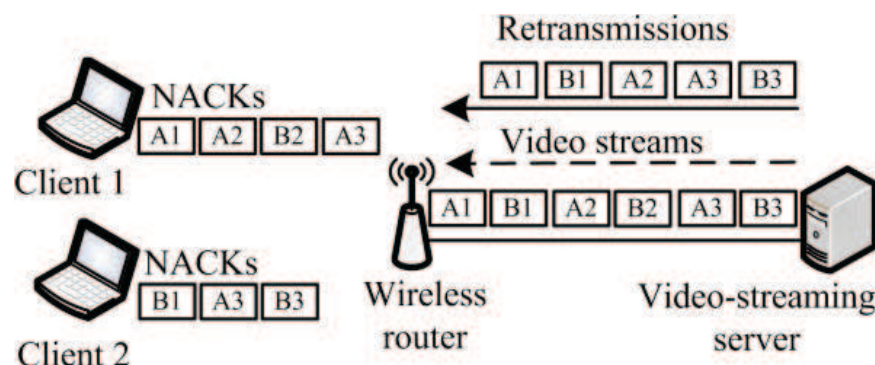

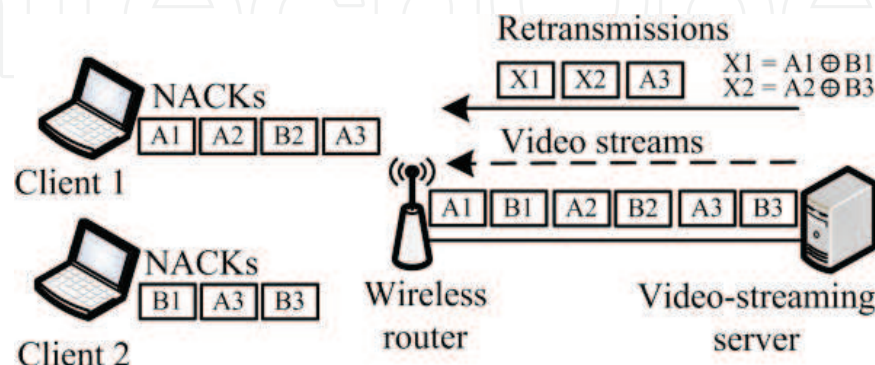**Figure 3.** Retransmissions using traditional retransmission scheme.

Figure 4. Retransmissions using NC-assisted retransmission scheme.

- All coded packets must be decodable on all the clients.

- Packets not received by any of the clients are sent out as they are, that is, not coded.

The streaming server keeps track of packets received for each of the clients. Information about unsuccessful received packet is provided through the negative acknowledgement (NACK) packet. Current status of the received packets at individual clients is presented in a transition table (e.g., **Table 1(a)**). M rows in the table correspond to packets that have not been received by at least one of the clients, and N columns correspond to the clients. Packets that have been received by all clients are not presented in the table. Packets with the lowest index have been present in the table for the longest period. There are three different states that indicate packet status:

- *State* 1: packet successfully received by the client.

- *State* 0: packet intended for the corresponding client not received means that packet needs to be retransmitted.

- *State* 2: packet not intended for the corresponding client not received means that packet does not need to be retransmitted. This state is required in the coding procedure as we explained below.

Packet status is represented in the transition table for the time needed to receive feedback from all the clients as explained in next section. Transmission and coding processes are based on the coding table (e.g., **Table 1(b)**) that is obtained from transition table and where only packets that require retransmissions are represented. As soon as packet status is noted in the coding table, the packet can be opportunistically transmitted. In coding table, two states are used to describe the packet status of the clients. Namely, not received packet is indicated by state 0, while packet received on the corresponding client is indicated by state 1. However, in the coding table, we are not presenting the packets that do not need to be retransmitted as they were not transcript from transition to coding table.

NC algorithm for making decisions on which packets to code together is presented in Algorithm 1. The algorithm is called after native packet is broadcasted. First, it checks if the first

(a) Transition table		
	C_1	C_2
A_1	0	1
B_1	1	0
A_2	0	1
B_2	2	1
A_3	0	2
B_3	1	0

(b) Coding table		
	C_1	C_2
A_1	0	1
B_1	1	0
A_2	0	1
A_3	0	1
B_3	1	0

Table 1. Transition and coding tables.

packet has exceeded *RTO*. Second, algorithm inspects the statuses of the first packet on the clients using the coding table (that is first row) and decides if it is codable. Packet is considered codable, if it has been received at least by one of the clients. Otherwise, it is transmitted as is. Last, the algorithm tries to code the packet with as many packets as possible (even the ones that have not reached the *RTO*) but prioritizes packets that have been waiting for a longer period.

Definition 1. Algorithm considers two packets codable, if coded packet can be decoded by all the clients.

In practice, this means that all sums over the corresponding packet columns from the coding table are not less than the number of coded packets $N_{CODED} - 1$. As shown in [5], client can decode coded packet if it has previously received at least $N_{CODED} - 1$ native packets coded in the encoded packet.

Algorithm 1

1: **if** (P_1 exceed RTO)

2:     **if** (P_1 is codable)

3:        k = 1;

4:        coded_packets [k] = &P_1;

5:        temp_coding = coding_table [first_row];

6:        coding = temp_coding [];

7:        **for** (m = 2; m <= M; m++)

8:          codable = 1;

9:          **for** (n = 1; n <= N; n++)

10:            temp_coding[n] = temp_coding[n] + coding table [m, n];

11:            **if** (temp_coding [n] < k)

12:              codable = 0;

13:              temp_coding = coding;

14:              break;

15:            **end if**

16:          **end for**

17:          **if** (codable)

18:            k++;

19:            coding = temp_coding;

```
20:                coded_packets [k] = &P_m
21:            end if
22:        if (k == N)
23:            break;
24:        end if
25:    end for
26:  end if
27:  if (k > 1)
28:      encode packets from coded_packets
29:      sent encoded_packet;
30:  else
31:      sent P_1 uncoded;
32:  end if
33:      update coding_table;
34:  end if
```

Example 1: Let us use the example from **Table 1 b** to explain the algorithm in practice. When algorithm is initiated, it first checks if the *RTO* for the first packet A_1 has exceeded. Then, it checks for the packet A_1 if it is codable. Since these conditions are fulfilled, algorithm examines if the second packet B_1 is codable with A_1. Referencing to the *Definition* 1, packets are codable, that is, sum of their corresponding columns is not less than *k*. In this example, only two packets can be coded together, hence the algorithm stops the coding procedure for packet A_1. Packets A_1 and B_1 are coded together and sent out using one transmission. If there were more clients, the algorithm would look for other coding opportunities trying to code more packets together. Further, when *RTO* is reached for packet A_2 algorithm, tries to code it with the next packet in the coding table A_3. Packets do not meet requirement from *Definition* 1, hence algorithm matches A_2 against the next packet, that is, B_3. In this case, packets are codable, thus, sent out in one transmission as together coded packet. Since there are no more packets left to code it to, lastly, packet A_3 is sent out as is. However, in such cases, where there is only one packet left in the coding table, is rarely encountered in practice. Simultaneously with packet transmissions, there are requests for retransmissions for new packets received. Thus, new packets are coming in the coding table to which A_3 would be matched for new coding opportunities.

In a given *Example 1* using proposed approach, five retransmitted packets are transmitted using only three transmissions. Even higher gains can be obtained in cases with more clients.

3.2. Implementation details

This section describes implementation details of the mechanisms required on the server and client.

3.2.1. Client side

On the client, *signalization*, *overhearing,* and *packet decoding* are the main processes and are described in greater detail in the following:

- *Signalization*: NACK packets are the only stand-alone packets used for signalization purposes and inform server in which packets have not been received on the clients. Two types of *NACK* packets are foreseen: hard *NACK* and soft *NACK*. Hard *NACK* is a packet sent by the client that has missed a packet from a stream currently in use (i.e., marked with 0 in the transition table). Soft *NACK* is a packet sent by the client that has not received packet for stream that is not intended for it (i.e., marked with 2 in the transition table). Soft *NACKs* only inform the server on the status of the received packets, though the packet not received is not required by the client. *NACK* packets are sent to the server using unicast mechanism, as it is a reliable mechanism and packets are unlikely to get lost.

- *Overhearing*: all clients listen to all the transmissions, even the ones not intended for them and store all packets in the packet pool, for decoding purposes.

- *Decoding*: with every received packet, the client checks if it is native or coded packet. In the case when a coded packet is received, the process checks the packet pool. If enough packets have been received (and stored in the packet pool), the client has enough information and it decodes the coded packet. Using previously stored packets and the *XOR* operation against coded packet, a native packet, previously not received is obtained [6].

3.2.2. Server side

Broadcasting, *NACK handling,* and *coding* processes take place on the server side, namely:

- *Broadcasting*: all packet transmissions from the server to clients are carried out using broadcast.

- *NACK handling*: for every *NACK* packet that server receives updates in transition table are made. Packet status is recorder in transition table only for a short time in order to gather *NACKs* for the packet from all the clients.

- *Coding*: after every broadcast of native packet server searches for coding opportunities in the coding table. If an opportunity is found, first packets are coded together with as many packets as possible sent out in one transmission to all clients. Otherwise, it is not coded than sent out as is.

3.2.3. Overhead estimation

NC-assisted retransmission scheme brings an additional overhead into the network in terms of additional packets and headers. They are depicted in **Figure 5**. All packets have additional
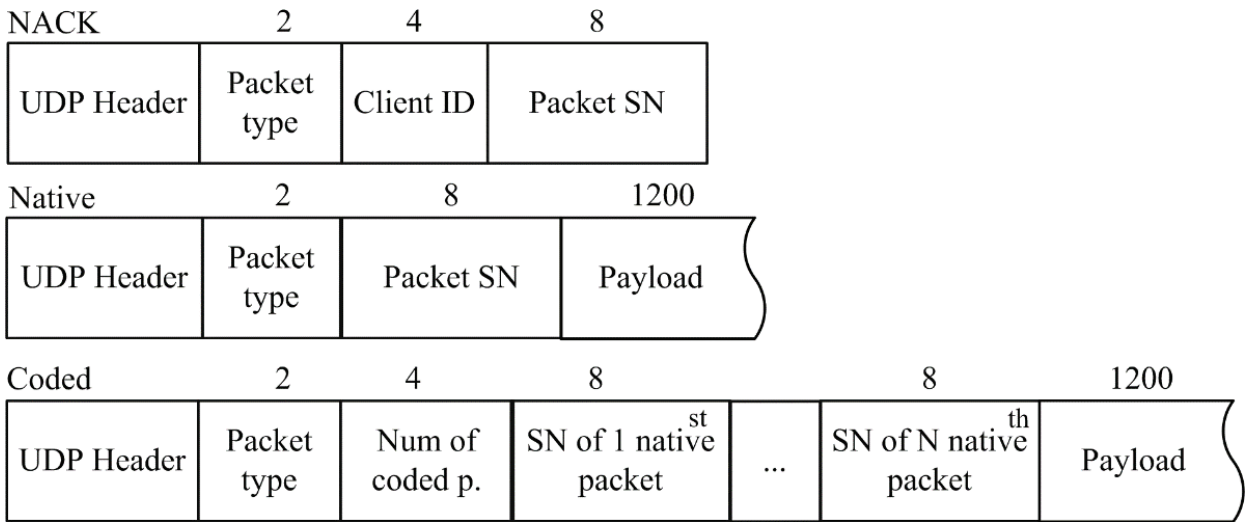
NACK	2	4	8
UDP Header	Packet type	Client ID	Packet SN

Native	2	8	1200
UDP Header	Packet type	Packet SN	Payload

Coded	2	4	8		8	1200
UDP Header	Packet type	Num of coded p.	SN of 1st native packet	...	SN of Nth native packet	Payload

Figure 5. Packet headers.

headers such as *packet type* field indicating that a packet is signalization (*NACK*), native, or coded packet (2 bytes). Native packet has additional *sequence number* (8 bytes) of a packet used for identification. In the case of a coded packet, *sequence number* for each of the encoded packet is given. Also, coded packet carries information on how many packets are coded in the encoded packet (4 bytes). Native packet has *Payload* (1200 bytes) field where video content is placed. Coded packet payload is the same as the native packet and it is composed of *XOR-ed* payloads of every encoded native packet.

Signalization (*NACK*) packet contains the network coding unique information such as: *Client ID* (4 bytes) field, which identifies the sender of the client.

4. Practical evaluation

4.1. Wi-Fi test bed

In order to evaluate the proposed scheme, we deployed IEEE 802.11 network test bed consisted of streaming server, access point, and multiple Wi-Fi clients. Server and clients were laptops with different hardware, for example, different wireless interfaces. For the access point, we used wireless router Linksys WRT54GL. Server was connected to the wireless router via Ethernet interface while clients were connected via WLAN interface. In order to perform tests, we used a third party firmware (i.e., DD-WRT) in our wireless router, because the original firmware was not capable of transmitting a multicast channel through it. Wireless router configuration was fixed during the experiments. Packets size was constant (i.e., 1210 bytes). Inter-arrival time between packets was also constant (i.e., 6 ms), resembling video-streaming application. Results were presented for the shortest time intervals to observe steady state conditions. Two client setups, mobile and static were used for the evaluation and are presented in the next sections.

4.1.1. Mobile client setup

In this scenario, experiment was set up with server and six clients, where each client position has changed once, during the experiment.

Server streamed one UDP 608 MB stream. If it streamed one packet every 6 ms (inter-arrival time between packets) that would be approx. 166 packets per second. The size of a single packet is 1200 bytes. Thus, the data rate was 1.525 Mbps.

The experiment started with all clients placed in the room with the router. Every 7 min, one client at a time was relocated to the nearby room. By moving clients to the nearby room, we wanted to observe how delivery probabilities were changing during the time. We assumed that delivery probabilities would decrease, due to wall obstacles between the server and the clients. Further, obstacles had caused packet loss, which we supposed would affect delivery probability of single clients in the nearby room.

Next, with decreased delivery probabilities, we expect more coding opportunities and higher bandwidth reduction, which we will discuss later in the result section.

4.1.2. Static client setup

In the second scenario, where clients were static and located in one room, four experiments were carried out, each with different number of clients (4, 6, 8, and 10). Server streamed one UDP stream in total of 122 MB and 1.525 Mbps data rate. Value of data rate is calculated as in for mobile client setup. Using this architecture setup, we wanted to observe if the number of clients will affect improvements in the bandwidth.

We expect to reduce more of the bandwidth, because with more clients we assume more coding opportunities, which means fewer transmissions and therefore better bandwidth reductions.

4.2. Evaluation metrics

Several performance metrics were used to evaluate the performance of the proposed scheme. All the presented results were observed over the sample period T_S (i.e., 20 s).

N_N is the number of native packets sent. N_R notes the number of retransmitted native packets. N_{RNC} is the number of transmissions performed to retransmit N_R packets, where $N_R \geq N_{RNC}$ if network coding is used. N_{Di} presents number of successfully received packets at i-th client, while N_C presents the number of clients.

Bandwidth reduction B_R is calculated as the proportion of difference of the N_R and N_{RNC} packets to the sum of N_N and N_R packets. We use B_R to show the share of bandwidth we reduced by using proposed approach.

$$B_R = \frac{N_R - N_{RNC}}{N_N + N_R} \cdot 100\% \tag{1}$$

Client delivery probability DP_i is defined for individual clients as:

$$DP_i = \frac{N_{D_i}}{N_{N_i}} \cdot 100\% \tag{2}$$

Delivery probability DP is an average of client delivery probabilities:

$$DP = \frac{1}{N_C} \sum_{i=1}^{N_C} DP_i \tag{3}$$

Client gain G_i is defined for ith client as:

$$G_i = 1 - \frac{N_{Ri}}{N_{RNC_i}} \cdot 100\% \tag{4}$$

Gain G is average client gains:

$$\mathrm{G} = \frac{1}{N_C} \sum_{i=1}^{N_C} G_i \tag{5}$$

Coding table size (CTS) refers to the number of packet-statuses on the clients noted in the table (note that only packet that requires retransmission is present there). Size is sampled periodically, after broadcasting native packet.

C_{AVG} is the average number of native packets sent in one retransmission (average number of encoded packets).

4.3. Experimental results

Results are presented individually for two client's setups that we explained above, that is, mobile and static. Mobile client's results give more detail results of described performance metrics above, while static client's results just discuss the number of clients depended of the gain.

4.3.1. Mobile client results

Bandwidth reduction B_R of proposed approach is presented in **Figure 6**. Straight lines in the figure indicate transition periods at which clients were relocated and divide figure in equal rectangles. For example, in the second rectangle from the left, five clients were located in the router room and one was in the room nearby. From this figure, we can observe that B_R increases over the time. The results show that in the given testbed, we have been able to save 6% of total consumed bandwidth per average. The value in observed time interval ranged between 4 and 8%.

DP is shown in **Figure 7** and we can see that it is over time lowering. This is expected because we were moving clients away from the router over the time.
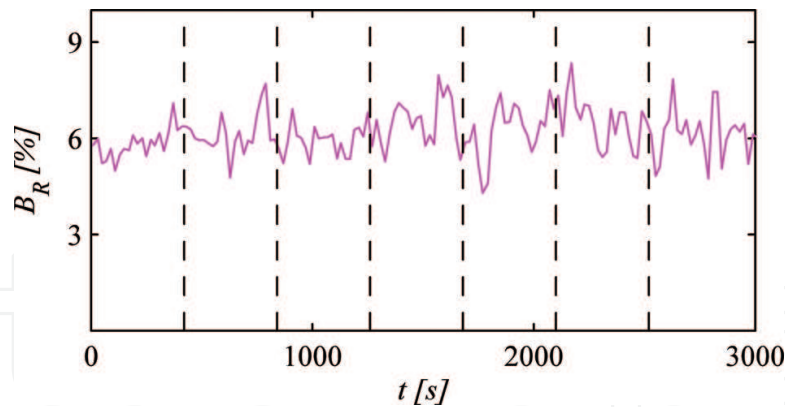
Figure 6. Bandwidth reduction B_R [%] in dependency of time t [s].
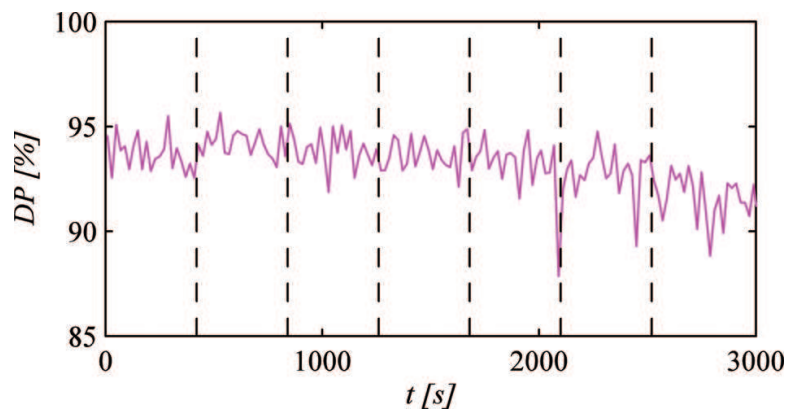

Figure 7. Delivery probability DP [%] in dependency of time t [s].

In the following, we show relations between average numbers of coded packets C_{AVG}, the number of transmissions performed to retransmit N_R packets, and gain G depicted in **Figure 8**. If C_{AVG} falls, N_{RNC} increases, and G decreases. For example, let us examine graphs in the time interval between 2100 and 2180s where very apparent case of described situation is shown. If more native packets are coded together, fewer packets need to be transmitted, thus G is increased. For example, for t = 2111 s, C_{AVG} = 1.831, N_{RNC} = 350, and G = 45%. Contrarily for t = 2130 s, C_{AVG} = 1.23, N_{RNC} = 1278, and G = 19%.

In the next step, we investigate DP versus the G as depicted in **Figure 9**. DP values range from 80 to 96%, while the corresponding values for G are from 18 to 46%. 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.

4.3.2. Static clients results

We investigate how the proposed scheme performs with different number of clients (i.e., 4, 6, 8, and 10). Results are shown in **Figure 10** for B_R. Observed interval is between 0th and 250th. Results show that highest B_R values are observed when the number of clients is the highest
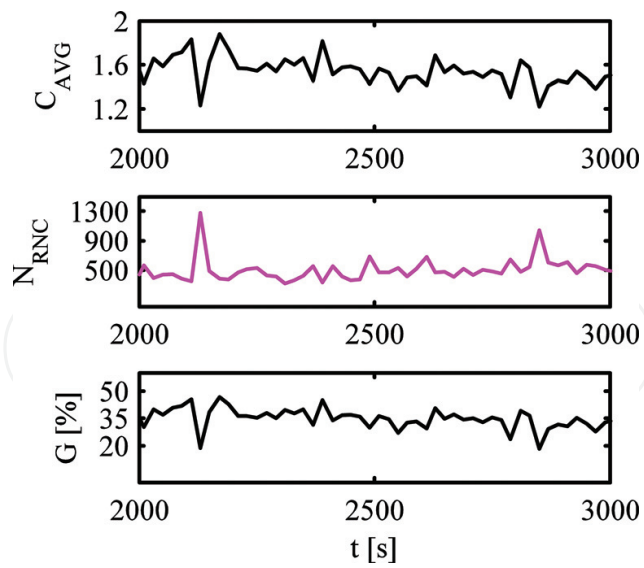
Figure 8. Average number of coded packets CAVG, coding table size CTS, and gain G [%] over time.
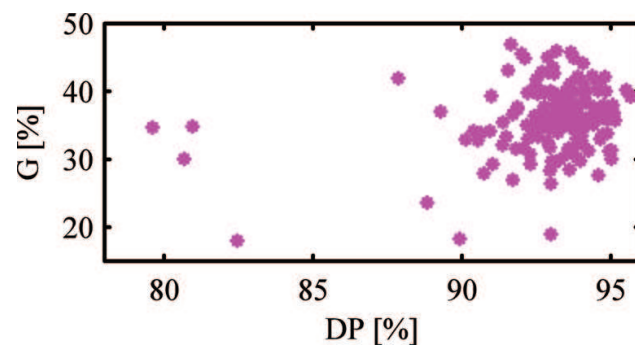

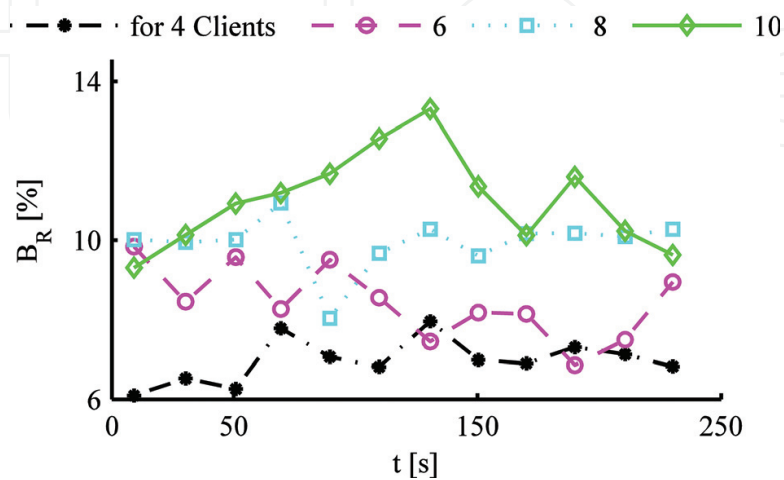**Figure 9.** Gain in G [%] dependency of delivery probability DP [%].

Figure 10. Bandwidth reduction B_R [%] over time for different number of clients.

(e.g., average B_R values for 4, 6, 8 and 10 clients are: 7, 9, 10, and 13%, respectively). When the number of clients increases, B_R improves at the same time, that is, when the number of clients is higher, there are more packets that get lost, hence there are more entries in the coding table CTS and coding algorithm is able to find more coding opportunities.

5. Conclusion

In this chapter, the approach for improving bandwidth in the last mile wireless broadcast network using network coding was investigated theoretically and practically. We have evaluated the performance of our approach for video-streaming services, such as IPTV, by designing and implementing a specific system solution, for a lossy wireless scenario. Our proposed schemes combine different lost packets from different clients in such a way that all clients are able to recover their lost packets with one transmission by the server. In particular, network coding is used to code and decode packets, as the new way of transmitting packets through the network. Instead of just store-and-forward technique, the server is now able to code packets in a more clever way, and transmit them over the broadcast network.

The implementation of such architecture, in a real scenario, was shown to be feasible. We tested the proposed approach with two different scenarios. It was found that the proposed approach in both cases decreases the number of transmitted packets. As compared to traditional unicast retransmission approach, where every packet is retransmitted separately, the reduction in bandwidth using the proposed network coding assisted retransmission scheme was up to 15%. This is a significant result and the developed scheme should be of great interest to be implemented in throughput demanding systems. In addition, results show that proposed approach saves bandwidth in various situations where users have different terminal equipment and different link quality. We observed the highest gains with high number of clients (i.e., 10 clients), which is also the case in the envisioned application where even higher number of different concurrent streams is anticipated.

Author details

Aleš Švigelj* and Melisa Junuzović

*Address all correspondence to: ales.svigelj@ijs.si

Jožef Stefan Institute, Ljubljana, Slovenia

References

[1] Junuzović M, Alič K, Švigelj A. Wireless broadcast transmission scheme for reliable video-streaming service. WSEAS Transactions on Communications. 2015;**14**:256-266

[2] Ahlswede R, Cai N, Li S-YR, Yeung RW. Network information flow. IEEE Trans. on Information Theory. 2000;**46**(4):1204-1216. DOI: 10.1109/18.850663

[3] Ferreira D, Costa RA, Barros J. Real-time network coding for live streaming in hyperdense WiFi spaces. IEEE Selected Areas in Communications. 2014;**32**(4):773-781. DOI: 10.1109/JSAC.2014.140409

[4] Seferoglu H, Markopoulou A. Video-aware opportunistic network coding over wireless networks. IEEE Journal on Selected Areas In Communications. 2009;**27**(5):713-728. DOI: 10.1109/JSAC.2009.090612

[5] Katti S, Rahul H, Hu W, Katabi D, Médard M, Crowcroft J. XORs in the air: Practical wireless network coding. IEEE/ACM Transactions on Networking. 2008;**16**(3):497-510. DOI: 10.1109/TNET.2008.923722

[6] Alič K, Pertovt E, Švigelj A. Bearing-opportunistic network coding. International Journal of Computers, Communications & Control. 2015;**10**(2):154-164. DOI: 10.15837/ijccc.2015.2.457

[7] Nguyen D, Tran T, Nguyen T, Bose B. Wireless broadcast using network coding. IEEE Transactions on Vehicular Technology. 2009;**58**(2):914-925. DOI: 10.1109/TVT.2008.927729

[8] Xiao X, Lu-Ming Y, Wei-Ping W, Shuai Z. A wireless broadcasting retransmission approach based on network coding. In: ICCSC 2008. 4th IEEE International Conference on Circuits and Systems for Communications; 26–28-05-2008. Shanghai, China. Shanghai: IEEE; 2008. pp. 782-786. DOI: 10.1109/ICCSC.2008.171

[9] Fang W, Liu F, Liu Z, Shu L, Nishio S. Reliable broadcast transmission in wireless networks based on network coding. In: IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS); 10–15-04-2011; Shanghai, China. Shanghai, China: IEEE; 2011. pp. 555-559. DOI: 10.1109/INFCOMW.2011.5928875

[10] Alič K, Švigelj A. Self-adaptive practical opportunistic network-coding procedure for static wireless mesh networks. Ad-hoc & Sensor Wireless Networks. 2017;**36**(1–4):87-105