ITUPublications

International Telecommunication Union

Standardization Sector

ITU-T Technical Report

(05/2023)

TR.NCDP

Session-layer network coding protocol for multicast data transmission

Technical Report ITU-T TR.NCDP

Session-layer network coding protocol for multicast data transmission

Summary

This Technical Report provides information about the session-layer multicast protocol based on network coding.

Multicast data transmission finds wide application in modern telecommunication networks, for example, for network management and control as well as for different network applications. The number of network service users and the number of services are increasing significantly, resulting in a tremendous network traffic growth. This in turn leads to an increase in network latency that could be inappropriate for some novel network services.

Therefore, one of the purposes of the emerging network protocols is to reduce the volume of network traffic without reducing the number of network services or their functionality. Network coding can solve this problem with bitwise XOR (eXclusive OR) addition, and a decrease of the number of packets transmitted through the network.

It is also important that such protocols be conceived taking into consideration implementation and testing. For this reason, it is convenient for the network coding protocol to be implemented as a session-layer protocol according to the Open Standards Interconnection (OSI) model. In this case, network coding functions are performed between the standard Transfer Control Protocol (TCP) / Internet Protocol (IP) stack of protocols and application-layer protocols.

Both of these aspects are considered in this Technical Report, which includes:

- Location of the protocol in the OSI model;
- Description of different network architectures where network coding protocol is applicable, and details of implementation;
- Coding and decoding procedures;
- Packet types and formats;
- Packet header processing;
- Signalling procedures and diagrams.

Keywords

Data transmission, multicast, network coding.

Note

This is an informative ITU-T publication. Mandatory provisions, such as those found in ITU-T Recommendations, are outside the scope of this publication. This publication should only be referenced bibliographically in ITU-T Recommendations.

Change Log

This document contains Version 4 of the ITU-T Technical Report on *"Session-layer network coding protocol for multicast data transmission"*

Editor:

Anastasia VYBORNOVA
The Bonch-Bruevich Saint-Petersburg State University of Telecommunications (SPbSUT)
Russian Federation
Tel: + 7-921-9237564
E-mail: a.vybornova@spbgut.ru

Artem Volkov
The Bonch-Bruevich Saint-Petersburg State University of Telecommunications (SPbSUT)
Russian Federation
Tel: + 7-981-8335077
E-mail: artemanv.work@gmail.com

Ammar MUTHANNA
The Bonch-Bruevich Saint-Petersburg State University of Telecommunications (SPbSUT)
Russian Federation
Tel: +7-952-21044-86
E-mail: ammarexpress@gmail.com

Alexey BORODIN
Rostelecom
Russian Federation
Tel: +7-985-3649319
E-mail: alexey.borodin@rt.ru

Evgeny TONKIKH
Russian Federation
Tel: +79036142576
E-mail: et@niir.ru

Sergey VLADIMIROV
The Bonch-Bruevich Saint-Petersburg State University of Telecommunications (SPbSUT)
Russian Federation
Tel: +7-951-6808102
E-mail: vladimirov.opds@gmail.com

Table of Contents

Technical Report ITU-T TR.NCDP

Session-layer network coding protocol for multicast data transmission

1 Scope

This Technical Report provides information on the session-layer multicast protocol based on network coding. The aim of this new protocol is to reduce the multicast traffic volume without reducing the volume of transmitted information.

2 References

[RFC 8406] IETF Request for comments 8406 (2018), *Taxonomy of Coding Techniques for Efficient Network Communications*.

3 Definitions

3.1 Terms defined elsewhere

This Technical Report uses the following term defined elsewhere:

3.1.1 network coding [RFC 8406]: A system where coding can be performed at the source as well as at intermediate forwarding nodes (all or a subset of them).

3.2 Terms defined in this Technical Report

None.

4 Abbreviations and acronyms

This Technical Report uses the following abbreviations and acronyms:

DCCP Datagram Congestion Control Protocol

ID Identifier

IETF Internet Engineering Task Force

IGMP Internet Group Management Protocol

IP Internet Protocol

MLD Multicast Listener Discovery

NC Network Coding

NCDP Network Coding Datagram Protocol

NWCRG Network Coding Research Group

OSI Open Systems Interconnection

PIM Protocol Independent Multicast

PN Packet Number

PT Packet Type

RTT Round Trip Time

SF Start Flag

SID Session Identifier

UDP User Datagram Protocol

5 Overview

As for the Internet Engineering Task Force (IETF) Network Coding Research Group (NWCRG), which also has worked on network coding, this research group is more concentrated on developing and standardizing methods and principles of network coding itself, rather than methods for network coding implementation in network protocols. Additionally, there were no responses from IETF NWCRG regarding ITU-T Q2/11 work on this document.

Multicast data transmission finds wide application in modern telecommunication networks, e.g., for network management and control as well as for the different network applications, including Internet radio, videoconferencing, video-on-demand and other content delivery services. An important benefit of multicast data transmission is that it reduces the amount of traffic transmitted through the network. The number of network service users increases significantly, as does the number of services, resulting in tremendous network traffic growth. This in turn leads to a network latency increase, that could be inappropriate for some novel network services, e.g., autonomous vehicles, telesurgery, augmented reality and other delay intolerant real-time services.

Therefore, one of the purposes of the emerging network protocols is to reduce network traffic volume without reducing the number of network services or their functionality. Network coding can solve this problem with bitwise XOR (eXclusive OR) addition, and a decrease of the number of packets transmitted through the network.

It is also important that such protocols be conceived taking into consideration implementation and testing. For this reason, it is convenient for the network coding protocol to be implemented as a session-layer protocol according to the Open Standards Interconnection (OSI) model. In this case, network coding functions are performed between the standard TCP/IP stack of protocols and application-layer protocols.

Network Coding Datagram Protocol (NCDP) provides the following advantages for multicast data transmission:

1) Reduction of the multicast traffic volume without reduction of the volume of transmitted information.
2) Can be implemented in existing networks.

6 Place of the network coding multicast protocol in the OSI model

NCDP is a session layer protocol; therefore, the coding functions are performed between the standard TCP/IP stack of protocols and application-layer protocols, as shown in Figure 6-1.

As NCDP is a multicast protocol, protocols without communication channel set-up should be used on the transport layer, e.g., User Datagram Protocol (UDP) or Datagram Congestion Control Protocol (DCCP). Session control functions such as connection control and packet numbering lie on the NDCP.

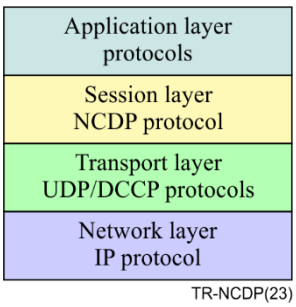

Figure 6-1 – NCDP in the OSI model

Multicast data transmission is based on IP group addresses. For the functions of multicast listener discovery and multicast routing, the following network protocols may be used: Protocol Independent Multicast (PIM), Internet Group Management Protocol (IGMP) (for IPv4), and Multicast Listener Discovery (MLD) (for IPv6). Also, NCDP's own control packets can be used for this purpose.

7 Network architecture of the network coding multicast protocol

7.1 Butterfly network architecture

Implementation of the NCDP for the "butterfly" network architecture is shown in Figure 2. Two sources, S_1 and S_2 send multicast packets to two receivers, R_1 and R_2. Basic multicast packet transfer is shown in black colour, and NCDP packet transfer is shown in blue colour. As Figure 7-1 implies, bitwise XOR (eXclusive OR) packets addition can reduce the number of packets in the network.

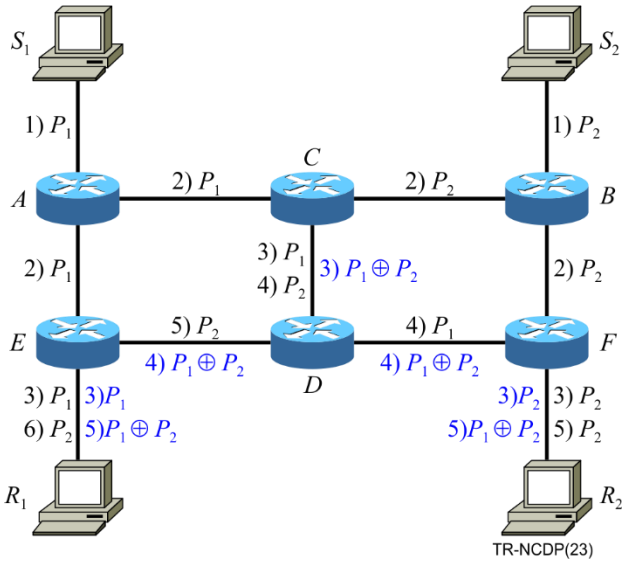

Figure 7-1 – Butterfly network topology and multicast packet transmission

The functions of the nodes in the "butterfly" network architecture are shown on Figure 7-2. Source nodes S_1 and S_2 send packets. Router C implements the network encoding procedure for NCDP packets passing through it. Destination nodes R_1 and R_2 provide network decoding of received NCDP packets. Routers A, B, D, E, and F act as protocol packet relay agents, routing them according to the multicast addresses.

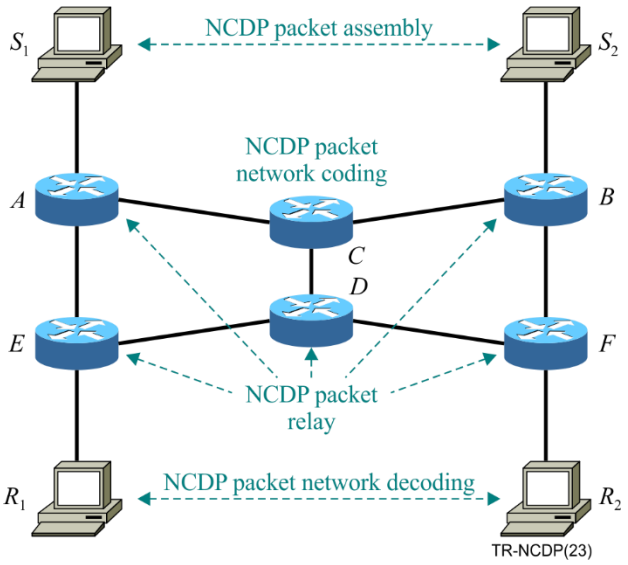

Figure 7-2 – NCDP in butterfly network

7.2 Diamond-shaped network architecture

In addition to the butterfly network topology, network coding can also be applied to the diamond-shaped topology, whose structure is shown in Figure 7-3. This topology is formed from the "butterfly" topology (Figure 7-1) by merging side routers that form "butterfly wings".

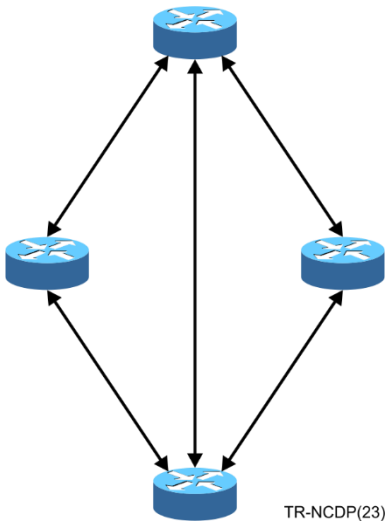

Figure 7-3 – General structure of a diamond-shaped network topology

A variation of a diamond topology, in which each side of the diamond belongs to one user subnetwork (Figure 7-4) is considered below. Source nodes S1 and S2 belong to subnetworks N_1 and N_2, which include the pairs of routers (G_1, G_2) and (G_1, G_3), respectively. Destination nodes R_1 and R_2 belong to the subnetworks N_4 and N_5, which include the pairs of routers (G_2, G_4) and (G_3, G_4), respectively. The internal subnet N_3 forms an additional route between routers G_1 and G_4. The packet routes in the network are shown in Table 7-1.

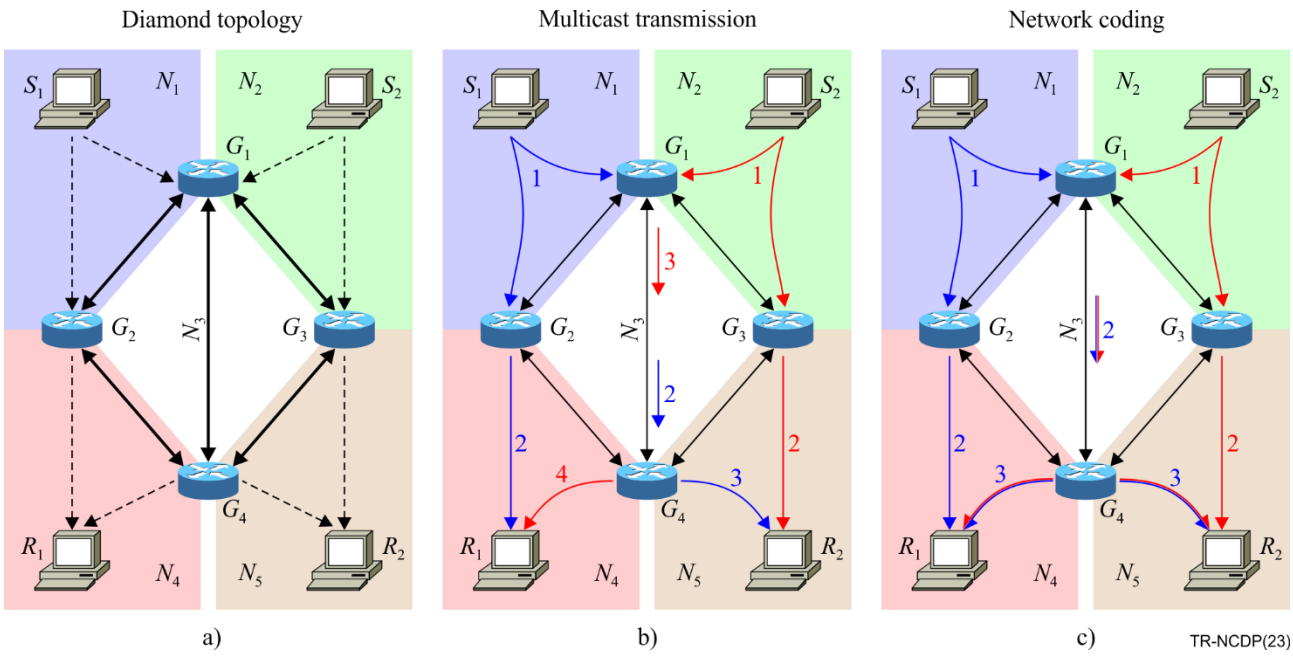

Figure 7-4 – Diamond-shaped network topology and multicast packet transmission

Table 7-1 – Packet routes in the diamond-shaped network topology

Source nodes	Destination nodes	
	R_1	R_2
S_1	G_2	G_1, G_4
S_2	G_1, G_4	G_3

Figure 7-4-b shows the multicast transmission of packets from sources S_1 and S_2 to destinations R_1 and R_2 without network coding. Figure 7-4-c shows the transmission of packets with network coding on the G_1 router and decoding on the destination nodes.

Comparing multicast transmission with and without network coding at Figure 7-4, one can see that, like in the case of the "butterfly" topology, network coding allows for a decrease in the number of packets that are transmitted in sections G_1-G_4. Without NCDP, it takes 4 steps for R_1 and R_2 to receive packets P_1 and P_2, while with NCDP, it takes only 3 steps.

8 Coding procedures

According to [RFC 8406], transmission systems with network coding are defined as systems where coding can be performed at the source as well as at intermediate forwarding nodes (all or a subset of them). Typically, network coding of data packets is represented by a reversible linear transformation of two or more data packets by means of mathematical operations. The main purpose of network coding is to reduce the volume of traffic transmitted over the network in order to reduce data transmission delay.

The main mathematical operation used in network coding systems is bitwise XOR (eXclusive OR), as the simplest reversible operation. Nevertheless, it is possible to use other reversible operations that provide unique decoding, for example, mathematical operations on finite Galois fields.

In the general case of transmission of N packets, the network coding procedure for forming an encoded packet is represented by the formula:

$$P_{NC} = P_1 * P_2 * \ldots * P_N$$

where * is a reversible operation or a set of operations for which $P * P = E$, where E is an identity element such that $P * E = P$.

To decode any of the P_i packets, one needs to know the encoded P_{NC} packet and the remaining N – 1 original packets. For example, to decode the P_1 packet, the following calculation is required:

$$P_1 = P_{NC} * (P_2 * \ldots * P_N)$$

In case of two sources, two packets P_1 and P_2 should be encoded. In the simplest case, the result of their network coding is represented by the packet $P_{NC} = P_1 + P_2 \pmod 2$. Thus, knowing the P_{NC} encoded packet and one of the original packets, it is possible to recover the second original packet. For example, if P_{NC} и P_1 are known:

$$P_2 = P_1 + P_{NC} \pmod 2 = P_1 + P_1 + P_2 \pmod 2 = P_2$$

In a more general case, it is necessary to take into account that P_1 and P_2 packets can have different sizes, and therefore, during encoding, they must be brought to the same size. In the simplest case, the smaller packet can be padded with zeros to the desired length. In this case, the network coding formula can be written as:

$$P_{NC} = ([P_1]+[P_2]_{L1}) \,||\, [P2]_{(L1\ldots L2)} \quad (1)$$

where || is a concatenation.

In this example, packet P_1 has an L_1 length that is less than the L_2 length of P_2.

In the case above, the resulting P_{NC} packet contains the part of the larger packet that is not encoded (in the example, $[P2]_{(L1\ldots L2)}$. To prevent this, the predefined constant A could be used as the third component of addition:

$$P_{NC} = (A+[P_1]+[P_2]_{L1}) \,||\, (A+[P_2]_{(L1..L2)}) \quad (2)$$

Another option for increasing the length of a smaller packet is to loop it using concatenation in a cyclic shift register. In this case the smaller packet is increased to the L_2 length of the larger P_2 packet using its own low-order bits:

$$P_{NC} = ([P_1]+[P_2]_{L1}) \,||\, ([P_1]_{(L2-L1)}+[P_2]_{(L1\ldots L2)}) \quad (3)$$

If necessary, an optional third component, a constant, could also be added:

$$P_{NC} = (A+[P_1]+[P_2]_{L1}) \,||\, (A+[P_1]_{(L2-L1)}+[P_2]_{(L1\ldots L2)}) \quad (4)$$

The examples for all the proposed options are considered below. All data are presented in hexadecimal notation.

Let packet P_1 of length $L_1 = 5$ bytes be equal to [23 A5 82 34 D2]. P_2 packet is $L_2 = 8$ bytes long and equal to [4F 28 56 AE D6 EA 39 48].

The calculation by formula (1) and the inverse transformation to decode the P_2 packet with known P_{NC} and P_1 is shown in the figure below.

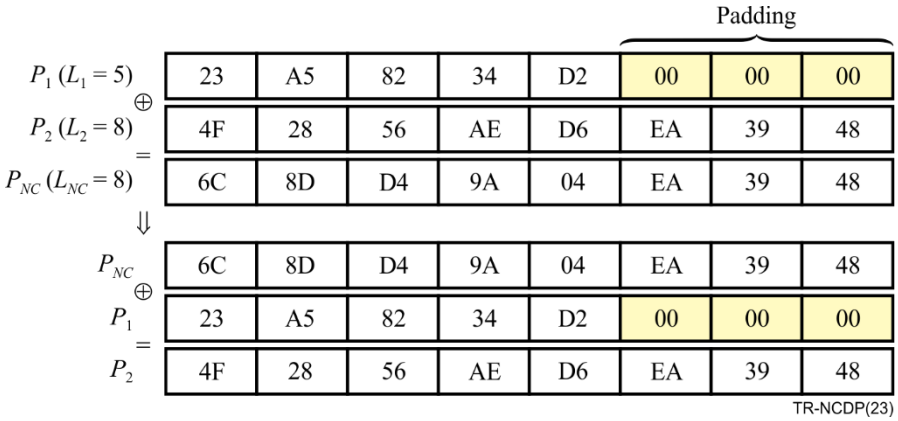

Figure 8-1 – Coding by formula (1) example

Calculation by formula (2) with constant A = 76_{HEX} and the inverse transformation to decode the P_2 packet with known P_{NC} and P_1 is shown in the figure below.

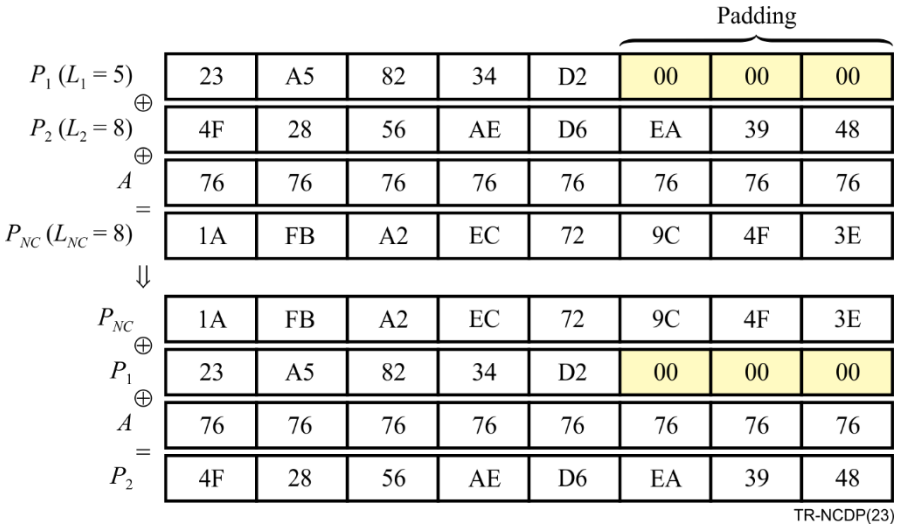

Figure 8-2 – Coding by formula (2) example

Calculation by formula (3) and the inverse transformation to decode the P_2 packet with known P_{NC} and P_1 is shown in the figure below.

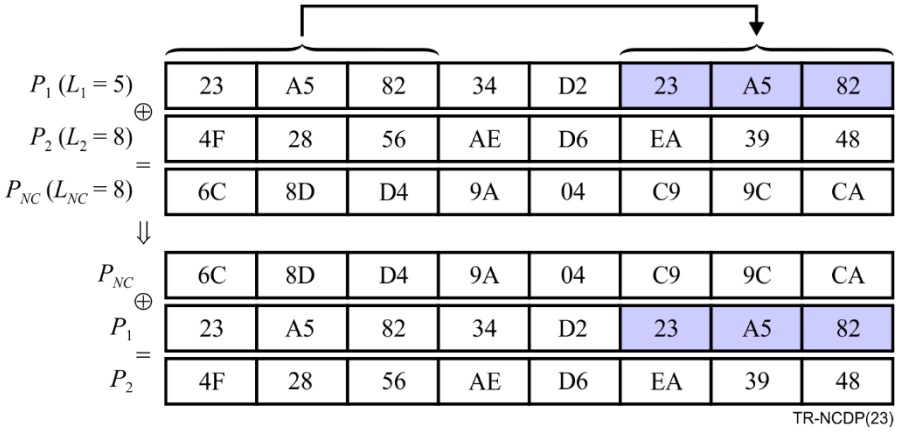

Figure 8-3 – Coding by formula (3) example

Calculation by formula (4) with constant A = 76_{HEX} and the inverse transformation to decode the P_2 packet with known P_{NC} and P_1 is shown in the figure below.

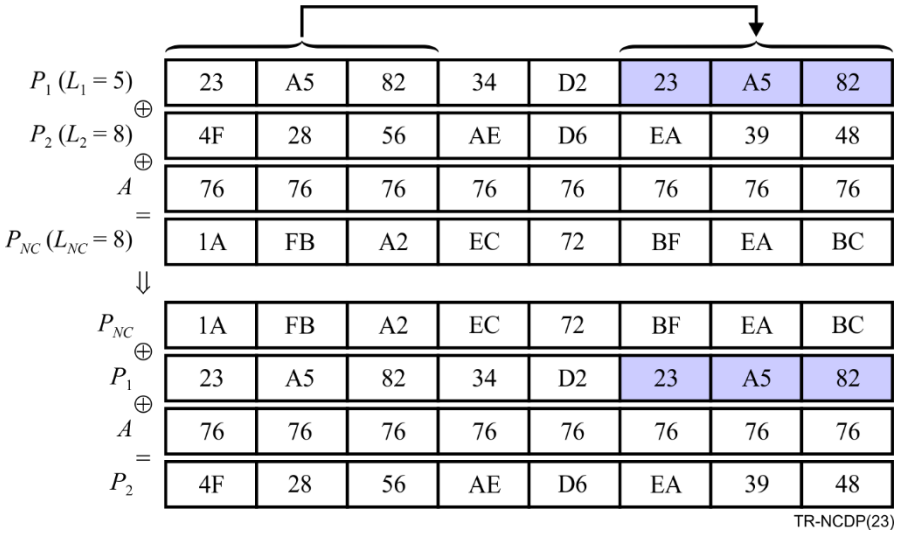

Figure 8-4 – Coding by formula (4) example

9 Packet Structure

An NCDP packet consists of a header, whose format and size characteristics depend on the NCDP packet type, and a data block, in which the upper layer protocol data are encapsulated.

The structure of the NCDP protocol header is shown in Figure 9-1. The header consists of two parts: a fixed part that is used for all NCDP packets, and a variable part that may be changed depending on the packet type (PT) and flag values.

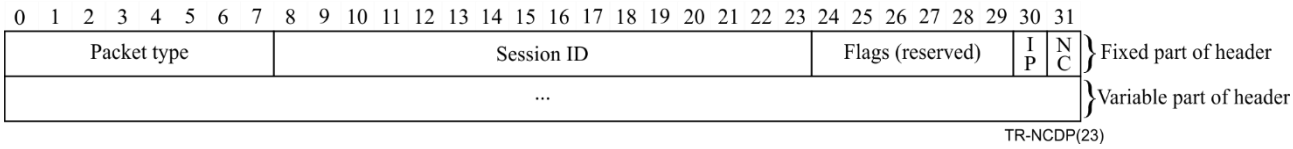

Figure 9-1 – NCDP packet header

The fixed part contains three fields: packet type (8 bits), session ID (16 bits) and Flags (8 bits). The Packet type is determined by its purpose. The session ID is used to distinguish packets of different multicast groups from each other. Flags indicate specific settings of packet transmission and depend on the type of the packet. In this Technical Report, two flags out of 8 are defined. The IP flag defines the version of the IP network protocol – IPv4 or IPv6. Depending on this, the source address fields will have different sizes. The network coding (NC) flag indicates whether the packet has gone through the network coding procedure or not.

At the moment, two types of packets are supported:

- Type 1 – data packet that is sent from the source to the destination. It may be coded using the network coding procedure while transmitted.
- Type 2 – control packet that contains the data of the beginning/end of the data flow transmission and the other information.

The variable part of the data packet header is shown in Figure 9-2. The header contains information about the sources. For each source, the Packet number, Data length, and IP address are indicated. For the packet that has not been coded yet (NC flag = 0) only one source is indicated. When the packet

has been coded at the Node C (Figure 7-2), the NC flag is set to 1 and the information about the second source is added.

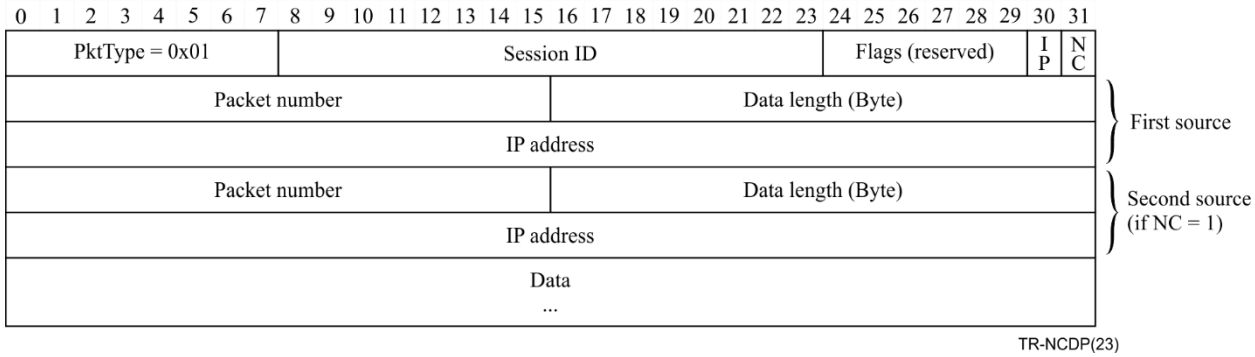

Figure 9-2 – NCDP Data packet header

Packets are numbered using the same principles as in the TCP. The packet number (PN) indicates the position of the first byte of the packet from the start of the data session. Packets are numbered cyclically by modulo 2^{16}.

Application layer Data length field is required because source nodes may transmit packets of different sizes. When such packets are added together during the network coding, the application layer data of the smaller packet is added with zero octets to the size of the larger packet. Since 16 bits are allocated for the length field, up to 64 KB of data can be transferred in one packet.

The control packet header is shown in Figure 9-3. The header contains the source IP address. To indicate the beginning / end of the transmission, an additional start flag (SF) is used in the fixed part of the header: "1" indicates the beginning of the transmission, "0" indicates the end of the transmission.

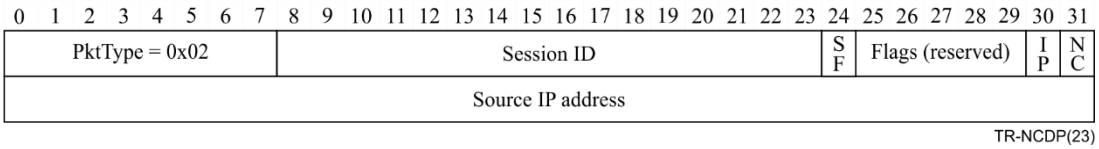

Figure 9-3 – NCDP Control packet header

Other types of packets, including upper-level protocol interoperability control, should be defined before the implementation of the protocol.

10 Protocol header processing

Since the NCDP packet header format depends on the purpose of the packet, its processing at the encoder (router) or decoder (destination node) starts when the packet type is determined.

Depending on it, processing is performed according to one of the two options described in clauses 10.1 and 10.2.

10.1 NCDP control packet processing

For a control packet, the SF flag, the Session ID, and Source IP address are defined. If the SF flag is equal to 1, the presence of an active session with the determined Session ID should be checked. If there is no session, then the memory areas intended for the accumulation and processing of data packets are initialized, then the node switches to the packet retransmission mode. If the session had already been established earlier by another source, then the node switches to data packet encoding

mode, and all the incoming data packets should be written to the memory buffer for further processing.

If the SF flag is equal to 0, the presence of an active session with the determined Session ID should be checked. If such a session is open, then the information source is marked as disabled. If this is the last source for that session, then the session ends, and the resources previously allocated to it are released. If there is no session for the received Session Identifier (SID), then the packet is recognized as erroneous and discarded.

The block diagram of the processing order of the header of the NCDP control packet is shown in Figure 10-1.

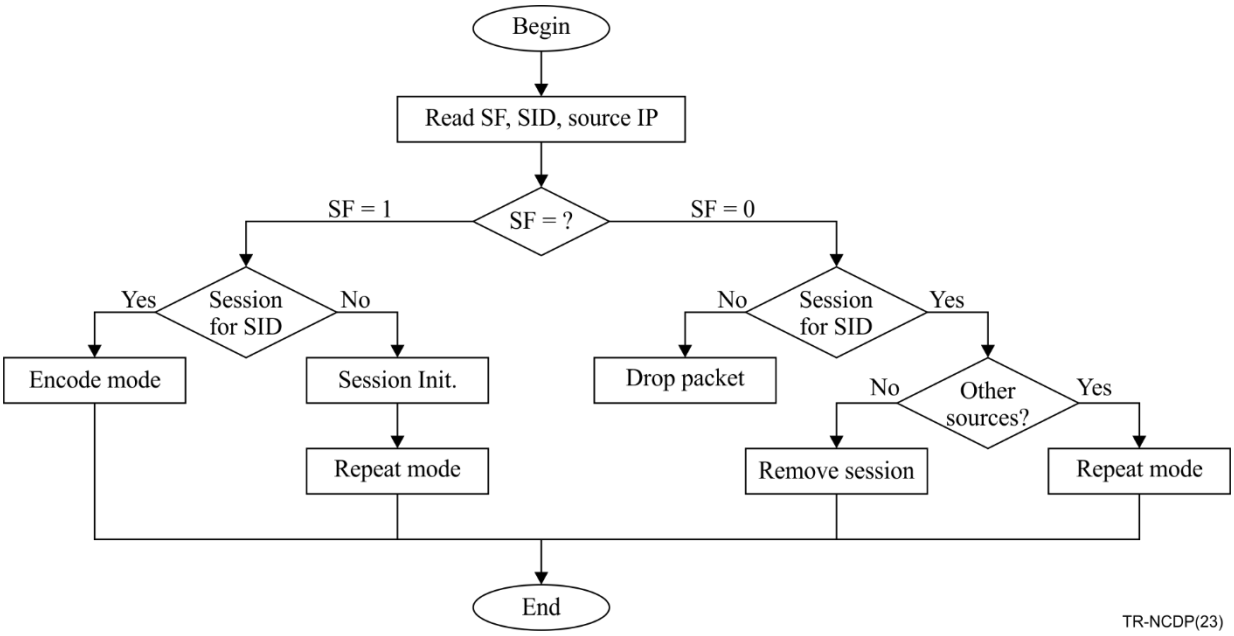

Figure 10-1 – NCDP control packet header processing on the encoder node

10.2 NCDP data packet processing

For a data packet, the Session ID is determined first, as shown in Figure 10-2. Then it should be checked if there is an open session with this Session ID. If there is no session, then the packet is forwarded according to the relevant multicast routing rules.

If a session with the given Session ID is open, then the NC flag should be checked.

If the NC flag is equal to 1, then this packet has already been encoded before and, therefore, is forwarded according to the routing rules.

If the NC flag = 0, then the number of active data sources is checked. If there is only one source of information, then the packet is sent to the next node according to the routing rules. If there are several sources, then it is possible to perform the network encoding procedure, and the received packet is sent to the memory buffer for the relevant Source IP.

As soon as two packets belonging to different sources appear in the memory buffers, the network encoding procedure is performed, and the generated encoded packet is sent to the next node. The procedure of the new encoded packet header generation is shown in Figure 10-3. The source that was registered earlier is indicated as the first source.

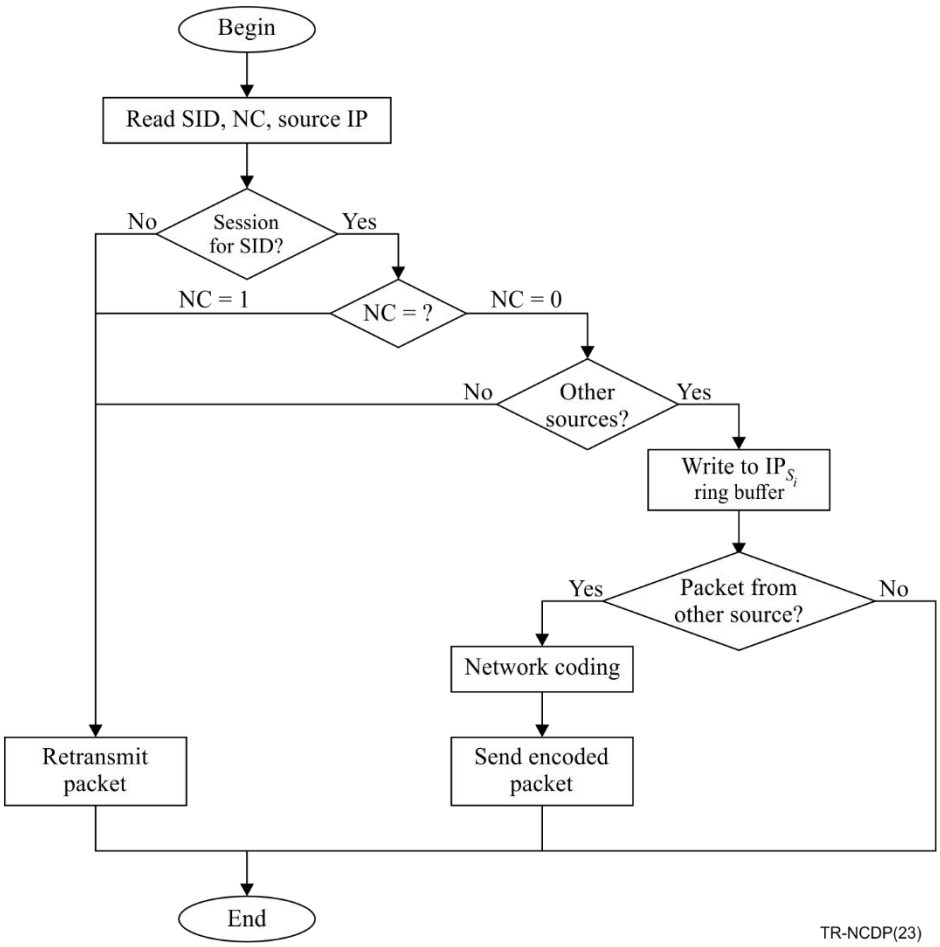

Figure 10-2 – NCDP data packet header processing on the encoder node

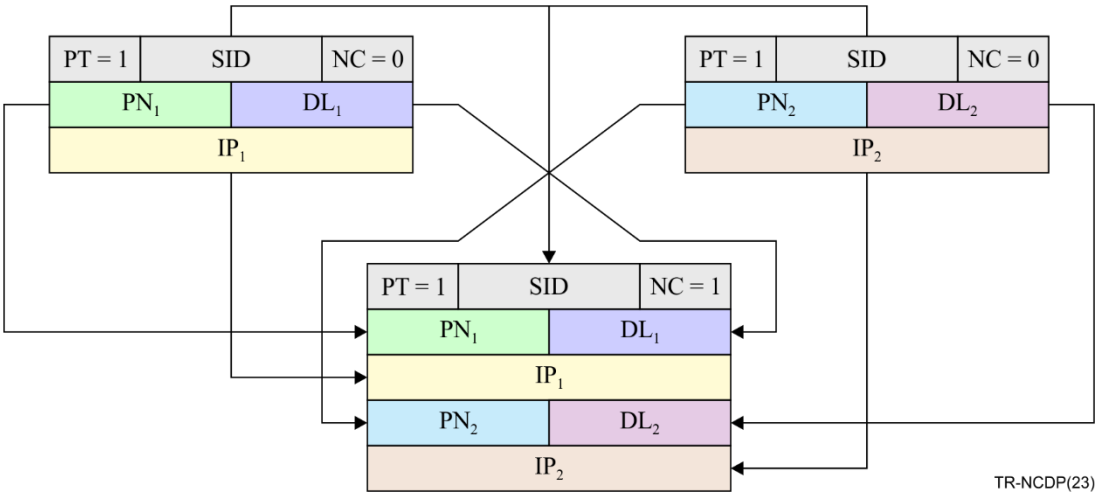

Figure 10-3 – Header generation for the encoded packet

11 Signalling procedures

This clause represents the packet exchange process based on the simplified network diagram presented in Figure 11-1. The diagram contains two source nodes S_1 and S_2, router C that performs network coding, and one of the destination nodes R_1. Since the other routers in standard butterfly architecture perform only a routing function (as shown in the Figure 7-1) without data processing, they are not considered in the process. Traffic exchange with the destination node R_2 is performed symmetrically, therefore it is also omitted.

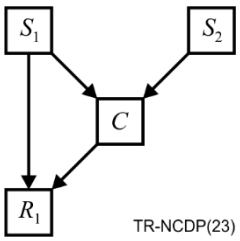

Figure 11-1 – Simplified network architecture

The packet exchange process between nodes is shown in Figure 11-2.

1) The transmission starts when the destination node requests data from the source nodes. This can be either a direct NCDP request or using IGMP or MLD protocols.

2) Further, the source nodes send control packets with the SF = 1 flag, signalling about the beginning of transmission (Start message) and informing router C and destination node R_1 about allocating resources to perform the network coding for the packets of this session.

3) Then source nodes send data packets. For example, Figure 10-2 shows the transmission of one packet from S_1 and one packet from S_2. Further data transmission is performed in the same way.

4) At the end of the transmission, the source nodes send control packets with the SF = 0 flag, informing about the end of the transmission. This final message is transmitted similarly to the start message.

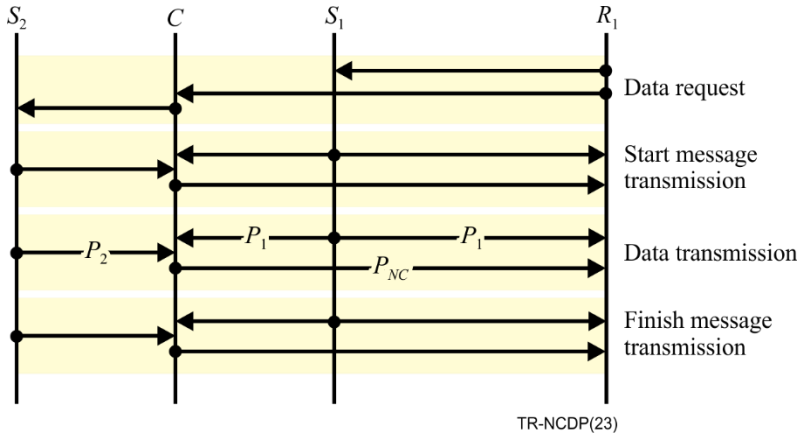

Figure 11-2 – Packet exchange procedure

It should be noted that the router *C* routes the starting and final control packets only from those source nodes for which it received a data request from the addressee. In the example above, router C received a data request from R_1 for S_2. The data request for S_1 is sent using the direct route R_1-S_1 (R_1-*E*-*A*-S_1 in the full butterfly network diagrams), therefore, control packets from S_1 arriving at *C* are not transmitted to R_1.

12 NCDP testing

To simulate the operation of the NCDP protocol on various network topologies, virtual model networks for butterfly and diamond-shaped topologies were used. The structures of these networks are shown in Figure 12-1.

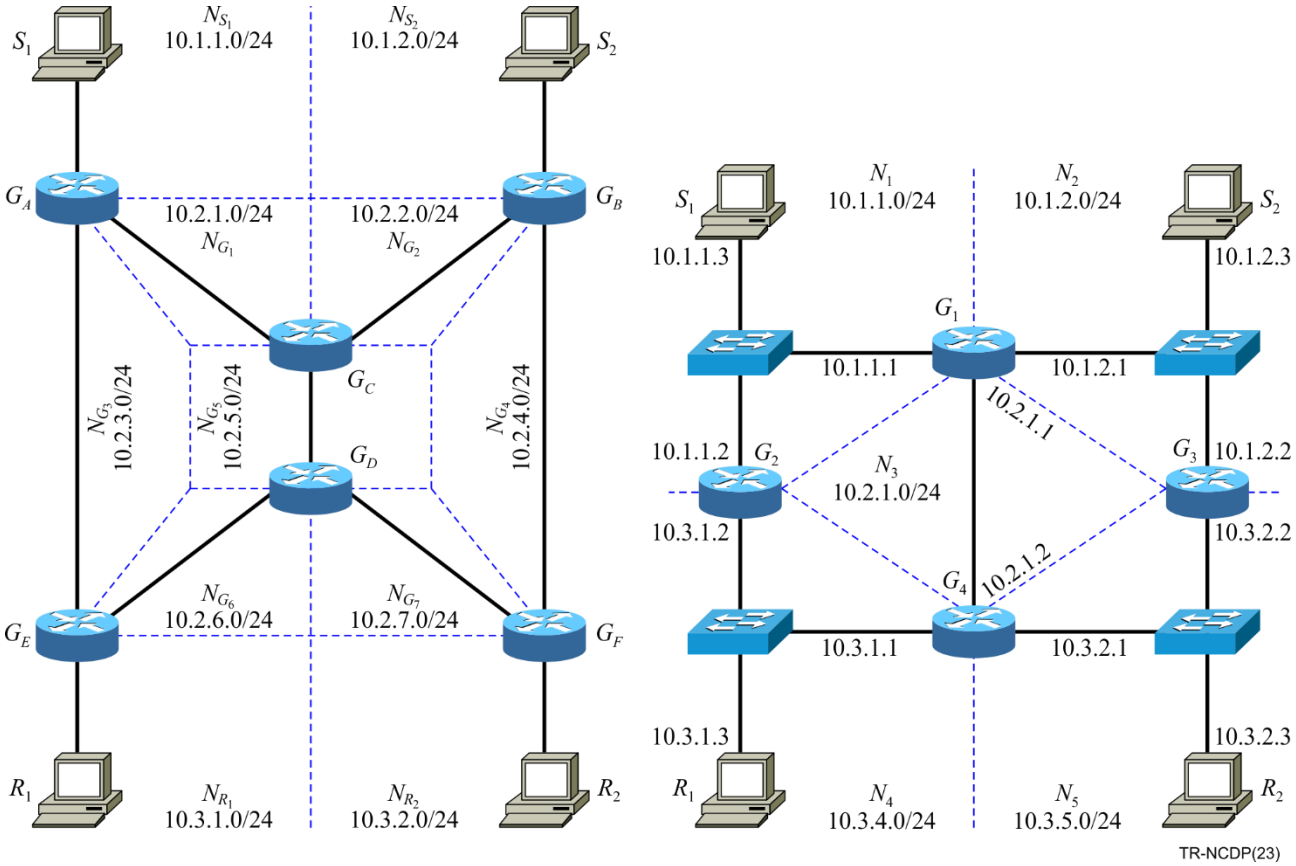

Figure 12-1 – Structures of virtual model networks for butterfly and diamond-shaped topologies

The first test of NCDP aimed to figure out whether the implementation of network encoding affects the round trip time (RTT) of packets through the network. The modelling results are shown in Figure 12-2. Here, the thin line shows the absolute scatter of measurements in the sample, and the thick line shows the standard deviation. The large dot corresponds to the mean value.

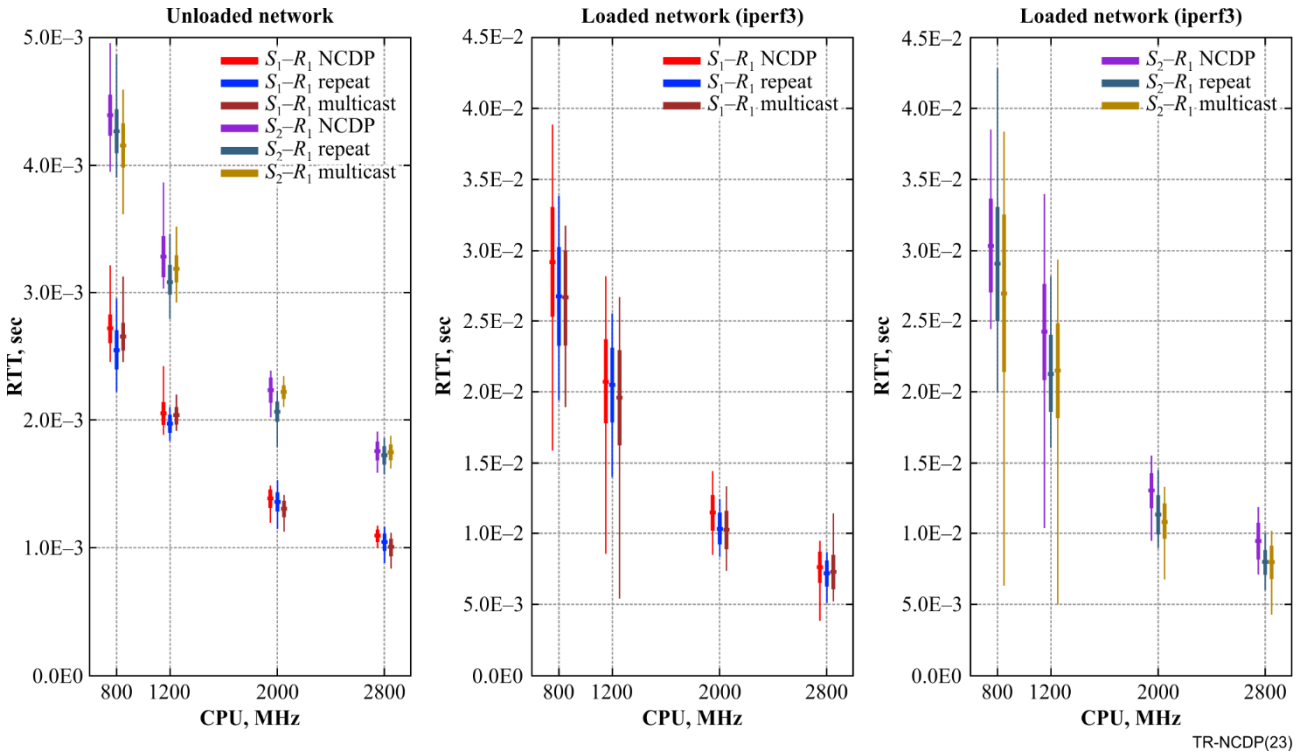

Figure 12-2 – RTT in the butterfly network topology with and without additional network load

It can be seen from the graphs that the average RTT values for different transmission methods differ slightly. Thus, it can be assumed that the use of the NCDP protocol does not lead to an increase in packet transmission time.

Next, the processing time of packets in the G_C router was estimated. The result is shown in Figure 12-3. As in Figure 12-2, the graphs show the mean, absolute spread and standard deviation for different network conditions. It can be seen that the packet processing time is approximately the same for network encoding and packet relay. At the same time, in a loaded network, coding provides a smaller standard deviation.

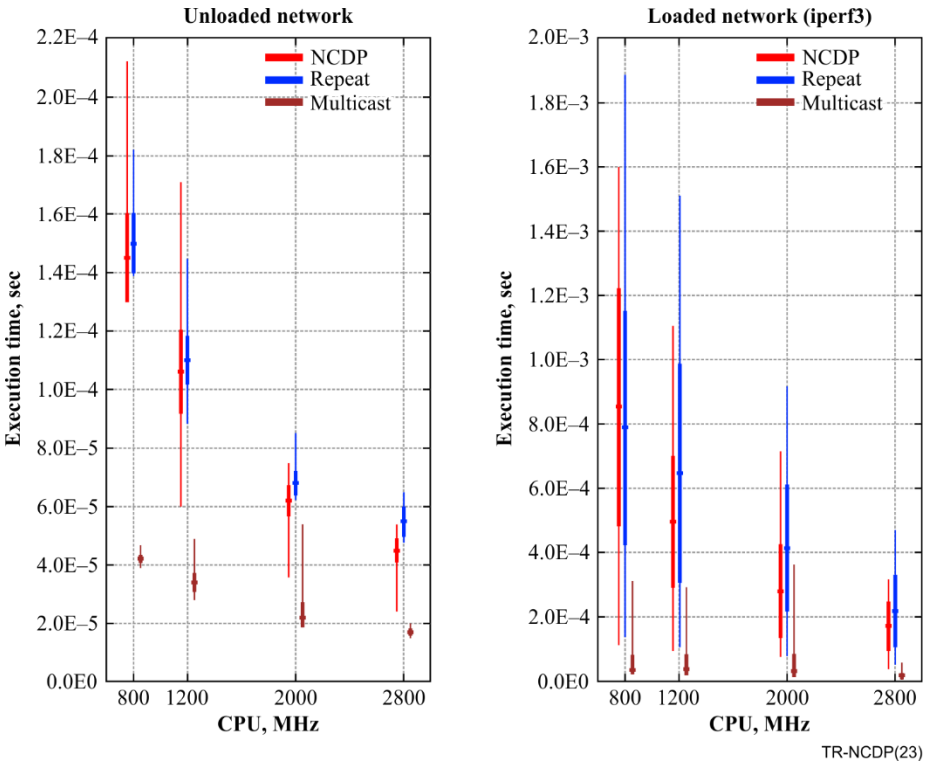

Figure 12-3 – Packet processing time on G_C router

Therefore, NCDP allows reducing the number of packets transmitted by routers in different network topologies, such as butterfly and diamond-shaped topologies. Modelling performed on the virtual model network shows that NCDP provides the same average RTT as conventional multicast, and therefore does not lead to additional delays. The simulation also shows that the time for additional packet processing and network encoding/decoding delays at the routers are insignificant, compared to the transmission and routing delays.