Noname manuscript No.
(will be inserted by the editor)

Enhancing the Capabilities of Mobile Backhaul: A User Plane Perspective

Engin Zeydan* · **Yekta Turk** · **Baris Berk Zorba**

Received: date / Accepted: date

Abstract Avoiding problems such as packet loss in the transport network is crucial for mobile network providers to offer high-quality services reliably and without interruptions. In this paper, we propose and compare three different transmission strategies, namely Caching, Network Coding (NC) and Repetition enabled transmission in the User Plane (UP) of mobile backhaul for network operators to prevent such performance degradation. In the proposed NC-enabled transmission method, NC provides robustness to transport network failures such that no further retransmission is required by the User Equipment (UE) compared to conventional approaches where UE applications perform retransmissions. The proposed scheme requires only a minor modification to the packet structure of the UP protocol, which requires a small development effort and no new extensions to the current UE standard features. We also discuss their placement in the O-RAN protocol stack and in the core network,

* Corresponding author.

Engin Zeydan,
Centre Tecnològic de Telecomunicacions de Catalunya (CTTC),
Castelldefels, Barcelona, Spain, 08860.
E-mail: engin.zeydan@cttc.cat
Y. Turk, B. B. Zorba
Aselsan A.S.,
Istanbul, Turkey. 34746
E-mail: {yektaturk, bbzorba}@aselsan.com.tr

and propose a new architecture that can utilize caching, repetition and NC features in the mobile network architecture. Our simulation results show that an exact 1% packet loss ratio in the backhaul link results in an additional total transmission time of 59.44% compared to the normal GPRS Tunneling Protocol – User Plane (GTP-U) transmission. Applying NC at a rate of 1% and 2% reduces this value to 52.99% and 56.26%, respectively, which is also better than the total transmission time of some previously studied dynamic replication schemes while keeping the bandwidth utilization at low rates. On the cache side, a reduction in latency of about 20% can be achieved with a cache size of 100 MB. At the end of the paper, we summarize some of the benefits and limitations of using these three strategies in UP of mobile backhaul networks.

Keywords O-RAN · orchestration · user plane · caching · network coding · services

1 Introduction

Major Mobile Network Operators (MNOs) around the world have are planning to migrate their radio access network (RAN) infrastructure to open and interoperable approaches, such as the deployment of Open RAN technology, which is an important step towards future mobile networks [1]. The joint efforts of MNOs' have provided a framework for the creation of an interoperable market for Open RAN and ensure stable deployment scenarios in future releases. The main features of the Open RAN architecture can be classified as: the compression of fronthaul traffic, joint processing options, the provision of accurate channels with radio resource allocation, and optimization in the upper layers.

At the same time, optimizing the RAN is a process that needs to be done precisely to improve the Quality-of-Experience (QoE) of mobile users. For example, when trying to improve the connection quality and performance of mobile users located near the cell center, negative effects on the signal of users located in the cell-edge should also be considered. To avoid such situations, the function Radio Intelligent Controller (RIC) has been defined in the O-RAN architecture. RIC can operate policy-based as well as use Machine Learning (ML) techniques in decision making.

Note also that the operations performed in the form of policies or real-time actions depend on the nature of the RIC itself (non-Real Time (RT) RIC and RT RIC as defined by O-RAN and explained in more detail in Section 2.

In 5G, NG is the interface between the Base Station (BS) and the User Plane Function (UPF) of the 5G Core Network (CN). Considering the recommended end-to-end delay times for 5G services, the delay tolerance of the end-to-end mobile network including RAN, transport and CN is much lower than previous generation mobile networks [2]. For example, the expected end-to-end delay of the cellular network defined by the 3rd Generation Partnership Project (3GPP) is about 20 ms for live streaming services [3], about 30 ms for time-critical sensing and about 1 ms for real-time control and automation services [2]. Services such as Ultra-Reliable Low-latency Communication (URLLC), Vehicle-to-X (V2X), etc. suffer from delays and packet losses in the transport network. Applications used by user equipment (UE) can retransmit, but in case of retransmission, the desired delay times for mission-critical services will be exceeded due to the time required for retransmission.

Performance problems in the mobile network cause retransmissions in User Plane (UP) and cause delays in the overall end-to-end communication, as shown in Fig. 1. Since there is a lossy backhaul link between the O-RAN enabled BS and the CN, there is always a possibility that a UP packet will be dropped [4]. When a UP protocol packet is discarded, it results in packet loss for the UE application. The expected transmission time for the UE application packet increases from T to $T + t$, where t is the time that elapses between the detection of the packet loss and the start of retransmission. Although there are many efforts to improve these properties, the optimization in the upper layers such as UP is not evaluated in detail [5]. Moreover, there is a defined E1 interface between RIC and UP in the Open RAN (O-RAN) architecture that can help improve the transmission capability on the UP side [6]. Unfortunately, this interface is only defined and the details of its working structure are not yet mature. To give a concrete practical example of the use of this interface, in this paper we propose an enhanced communication capability on UP of RIC in O-RAN as well as CN by using Network Coding (NC), caching and replication schemes. In the proposed architecture, the goal is to minimize the packet loss rate observed

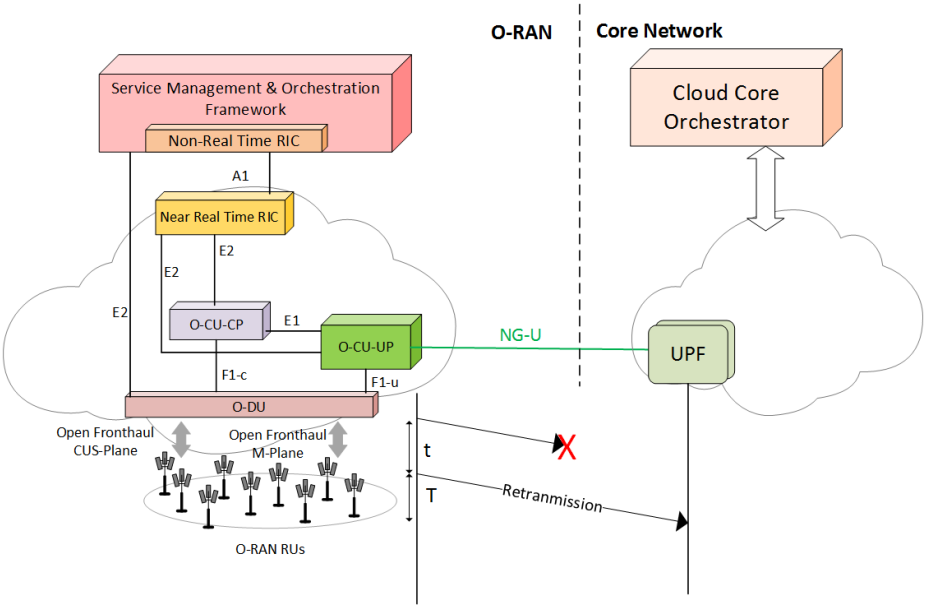


Fig. 1: A performance problem in the mobile network triggers a retransmission in UP, causing a delay in the overall end-to-end communication.

in the UP by using NC, replication and/or caching on both the O-RAN enabled BSs side and the CN side.

2 Related Works & Motivation

Standardization efforts on RIC are underway, and their capabilities are being evaluated in studies by the O-RAN Alliance [6]. In the O-RAN architecture, there are two main components, namely non-RT RIC and near-RT RIC, introduced to improve traditional network operation with embedded intelligence for various use cases [7]. In the O-RAN architecture, the near-RT RIC interfaces with theCentralized Unit (CU)-UP for data transmission. At the lowest level of the architecture, Distributed Units (DUs) are used to access the CUs and provide UE services through the Radio Units (RUs). Artificial Intelligence (AI) capable RICs also have different interfaces (O1, E2, A1). For non-RT RIC, interface A1 enables policy-based guidance,AI model management, and RAN optimization information for the near-RT RIC function. In-

terface E2 is used between near-RT RIC and CU and RAN DU and provides a standard interface between near-RT RIC and CU/DU. Interface O1 is used to support the operation and management of CU, DU, and RU as well as near-RT RIC e.g., by providing configuration and fault management data (which are instantaneous due to their event-driven nature) and performance management data (which are often aggregated over time intervals and include Key Parameter Indicators (KPIs) and counters) to Service Management and Orchestration (SMO).

Recent standardization efforts under the Internet Engineering Task Force (IETF)'s low queuing latency, low loss, and scalable throughput (L4S) architecture aim to achieve both high bandwidth and low latency [8]. On the other hand, many different factors must also be considered when providing services such as URLLC. These include the selection of an appropriate spectrum for coverage and capacity in RAN, the quality of the transport network, and the selection of the centralized 5G CN architecture. Time Sensitive Networking (TSN) is a recommendation from the Institute of Electrical and Electronics Engineers (IEEE) working group that provides faster transmission for transport networks in terms of flow sensitivity and faster switching. Although the TSN recommendation focuses on the technical characteristics of the new generation transport devices used by MNOs, capabilities of 5G New Radios (NRs) have yet to be adopted in the transport network domain [9].

In the literature, the article [10] defines how RIC decouples the control and data planes of RAN and powers an intelligent and continuously evolving wireless network with AI-driven applications. In [11], relationships between AI and the techniques under consideration in 5G mobile networks, as well as the demonstration of the effectiveness of AI for managing and orchestrating network resources are presented. A flexible, programmable, and open source Software-Defined Networking (SDN) platform for heterogeneous 5G RAN is introduced [12]. The authors describe an open-source interface for programming the control plane with SDN which can also be used for programming the RIC functionality. The authors in [13] propose the aiOS platform, an AI-based platform for autonomous management of software-defined wireless local area networks (SO-WLANs). The platform is also aligned with the O-RAN Alliance disaggregated radio access network architecture. In [14], a large-scale wireless

research test-bed called Colosseum is proposed to test novel AI-based algorithms and conduct large-scale experiments in the wireless domain. In [15], the use of a distributed algorithm at the backhaul switches is proposed to detect and temporarily manage congestion. All these studies have focused on controller platforms to enable an intelligent management platform at the RAN side between the UE and the BS. On the other hand, such enhanced capabilities must also be enabled on the UP side. In this paper, we aim to define the autonomous operations (including RT and non-RT) that can be performed by RIC against transport-related problems on the UP side.

Our previous work in [16] focuses on evaluating the impact of replicating lost UP packets when failures occur on mobile backhaul links of transport networks. Although some of the benefits are presented in [16], the bandwidth usage during the replication process is still too high. For this reason, NC on the UP side, as proposed in this paper, would be more effective than replication in terms of bandwidth utilization. As for the developments on the NC side, in [17], the authors used SDN and Network Function Virtualization (NFV) in real-time applications of 5G networks using Random Linear Network Coding (RLNC) for the first time. The goal was to improve the flexibility of 5G networks and reduce packet loss. The article in [18] uses NC in Cognitive Radio Networks (CRNs). This method is implemented in this section to maximize spectrum utilization and secure packets. The NC schemes applied to CRNs motivated us to implement this method in different types of networks. Research paper in [19] presented the implementation of NC and Diversity Coding (DC) in a 5G wireless Cloud Radio Access Network (C-RAN), where the combined use of NC and DC leads to an increase in the throughput of fronthaul networks for downlink broadcasting and multicasting. At the same time, these methods provide reliable networking with low latency. The authors in [20] used NC as a potential solution to improve end-to-end latency and reliability, focusing on the Integrated Access Backhaul (IAB) networks introduced by 3GPP. It has been shown that NC of UP provides a visible improvement in application performance in lossy backhaul links [21]. However, the work in [21] does not consider the asymptotic behaviour

of the approach and also lacks comparisons with different implementations of RAN functions (e.g., with caching) in UP.

As for the caching perspective in the mobile network domain, similar to caching in microservices [22], there are many ways to place the cache in the transport domain from an architectural perspective. The caching logic can be placed inside the network equipment (embedded cache), entirely in a separate cache server (client-server or cloud cache), in front of the network equipment (reverse proxy cache), or as a sidecar that is part of the network equipment (sidecar cache usually in Kubernetes environments and a mixture of embedded and client-server cache). Network virtualization and the use of acceleration methods in UP enable high-bandwidth network functions [23]. This can reduce the load on the kernel and the central processing unit (CPU). The study in [24] proposed Vector Packet Processor (VPP) as a framework that represents networking in user space. The networking operations are performed in the user space, again focusing on the same concept of avoiding the kernel space for networking. Caching can be applied to all backhaul switches to detect and manage congestion, latency, and convergence time [15]. Unfortunately, this approach requires an extension of all devices in the mobile network. This paper recommends to transfer UP packets directly to the cache without entering the kernel and without loading the processor, which is a similar approach to the user space networking. Moreover, the extensions are only needed in CN and BS.

Our Contributions: This paper explores answers to the following questions: (i) How can we mitigate packet loss problems in backhaul links of mobile networks? (ii) How can we avoid quality problems in backhaul links regardless of the capabilities of the backhaul equipment (routers, switches, DWDM, microwaves, etc.)? (iii) How can we design a mobile network architecture that enables specific next generation services in backhaul networks?

The existing backhaul infrastructure of mobile operators is not compatible with the current use cases of 5G [25]. In traditional UP of cellular networks, there is no packet loss recovery mechanism. Moreover, there are no defined mechanisms in the RIC of O-RAN architecture against the UP packet loss problem. However, the RIC can also be used to monitor, control, and heal the performance of the UP by using

one or a combination of packet recovery techniques. One of the advantages of the proposed RIC functionality is that it prevents problems such as packet loss/drops when used in the transport network of an MNO. Moreover, we assume an embedded architecture pattern where caching is embedded in C-RAN and CN. The RIC will be able to execute various UP performance solution patterns such as caching, NC, and replication and provide them in synchronization with the Service Orchestrators (SOs) of O-RAN and CN. With the proposed solution, the next-generation mobile network services can operate reliably and without interruption. In this way, end-to-end delay requirements defined by 3GPP for next generation mobile networks can be achieved in practice. At the same time, MNOs will not be dependent on transport network problems when providing next-generation services. With the operation of UP protocol supporting the proposed system, C-RAN and UPF in cellular networks can deal with the transport network problems. The main contributions of this paper can be summarized as follows:

- We propose an architecture that relies on interactions between O-RAN and C-RAN to enable the activation of the proposed UP functionalities, namely UP caching, NC-enabled and dynamic repetition transmission schemes for mobile backhaul networks.
- It has been shown that the proposed UP caching and NC-enabled transmission schemes in mobile backhaul networks enable fast processing and reduce the end-to-end latency compared to non-coded scheme. More specifically, when NC is applied, the total transmission time is saved by about 7% time compared to a non-coded transmission scheme with 1% NC ratio. We also observe a tradeoff between the total transmission time and the NC ratio relative to the expected packet loss ratio such that the minimum total transmission time is achieved when the NC rate is equal to the expected packet loss rate.
- UP caching results show that for a cache size of 100 MB, a reduction in latency of about 20% can be achieved compared to no caching.
- At the end of the paper, we also discuss some of the benefits and limitations of the proposed methods to combat UP packet loss.

TABLE 1

USED SYMBOLS AND THEIR CORRESPONDING DEFINITIONS

Symbol	**Meaning**
c	the allocated size of the cache
R	the data set to be downloaded at unit time t_{unit}
R	the set of this data set and is equal to $\{\mathrm{r}_1, \mathrm{r}_2, \ldots, \mathrm{r}_\mathrm{n}\}$
$P_\mathrm{c}(r_\mathrm{i})$	the probability of the cache containing the i-th data in the data set R at the t_{unit} of the data download request
$P_\mathrm{R}(c)$	the probability of the cache containing the set R at the $\mathrm{t_{unit}}$ of the data transfer request
ζ	expected packet loss rate
ρ	network coding rate
c_1, c_2	weight factors adjusting costs in terms of delay for the packet loss and network coding
p_m	m-th lost packet
p_{nc}	additional encoding packet

The remainder of the paper is arranged as follows. Section 3 presents the proposed integrated O-RAN and core cloud network architecture from the perspective of UP. Section 3 presents some factors affecting end-to-end delay, their formulations in the context of caching and NC and prevention methods against UP performance problems. Section 4 presents details of the NC-enabled transmission strategy and its asymptotic behavior of NC. Section 5 presents the experimental results and the advantages and limitations of the studied transmission strategies. Finally Section 6 presents the conclusions and future work. In addition, the Table 1 contains all the symbols and their associated definitions used throughout the paper.

3 System Architecture & Design

3.1 Factors Affecting End-to-End Delay

In cellular systems (including 4G and 5G), end-to-end latency is introduced into the system mainly by three different factors. The first factor is the location of the application server. The closer the server is to the users, the lower the latency. The second factor depends on the RAN scheduling and Quality-of-Service (QoS) management mechanisms. RAN scheduling based on 3GPP's QoS framework in combination with a policy server, can enforce cellular QoS at the flow level. Each flow packet can be classified and labelled so that it can be mapped to data radio bearers in the access network. The third option is to use network slicing services based on specific Service-Level Agreement (SLA), especially in 5G networks. Similar to Software-as-a-Service (SaaS) of the public cloud, service frameworks need to set up, operate and secure network services in the form of a Network-as-a-Slice model.

In RAN, transport networks or more recently IAB, latency/delay can be caused by queuing delay, segmentation delay and retransmission delay. *(i) Queuing delay* occurs when packets are waiting to be transmitted in transport network elements or eNodeB/gNodeB . It is a cumulative delay caused by other packets being in the buffer before the incoming packet. This can usually be caused by several scenarios. The first cause is network congestion. Lower bandwidth per UE may lead to more contention if too many packets are generated by too many UEs. The second cause is poor channel conditions, which may result in small Protocol Data Units (PDUs) being transmitted, causing remaining PDUs to be queued. The third case is when the packet arrival rate is higher than the transmission rate. *(ii) Segmentation delay* arises from segmentation at lower layers (e.g., at the Radio Link Control (RLC) layer in radio communications) when a large IP packet needs to be fragmented into multiple PDU segments, resulting in high latency. This may be due to various reasons such as poor channel conditions (especially in radio communication) where the packet size is too large for transmission, network congestion due to low resources for the UE,

etc. *(iii) Retransmission delay* is caused when the PDU is retransmitted more than once due to poor channel conditions.

3.2 End-to-End Formulations for NC & Caching

In a network, the total transmission time (T_{e2e}) is defined as the total time taken for a packet generated by the source to reach its destination [26]. It depends on the number of hops between the source and the destination, as well as the conditions and characteristics (such as load, distance, etc.) of each hop. It can be formally defined as

$$T_{e2e} = \sum_{\forall i} D_{hop,i}, \tag{1}$$

where

$$D_{hop} = D_{node} + T_{prop}, \tag{2}$$

representing total delay in a hop,

$$D_{node} = T_{proc} + T_{queue} + T_{trans}, \tag{3}$$

representing total delay in a node, T_{prop} is the time it takes for a signal to propagate across the communication medium from one node to the next, and T_{trans} is the time it takes to transmit an entire packet into the communication medium (higher link bandwidth results in lower transmission time), T_{proc} is the time spent processing a packet in the node (on the other hand, the processing time for normal data packets (except ICMP (Internet Control Message Protocol) packets) is usually negligible), T_{queue} is the time a packet spends in a queue. In 4G and 5G networks, physical distance may be negligible if the backhaul link consists mainly of optical and/or short-range radio links. Therefore, the impact of the physical distance that the packets have to travel on the end-to-end latency can be considered negligible. Note that, in our experimental test ($T_{trans} + T_{prop}$) is constant and does not change when the

NC coding rate changes, as shown on the *x-axis*. The sum ($T_{queue} + T_{proc}$) increases when the NC ratio increases.

Let us define c, which represents the allocated size of the cache, and R, which represents the amount of data to be downloaded in a unit time t_{unit}. Then R = {r_1, r_2, . . . , r_n} be the set of this dataset. Since the acceleration process speeds up the processing of CPU within the node, we have the node processing delay with caching, which can be expressed as follows,

$$T_{proc\_with\_cache} = T_{proc} \times (1 - \sum_{i=1}^{n} \frac{P_c(r_i)}{P_c(R)}) \tag{4}$$

where $P_c(r_i)$ is the probability that the cache contains the r_i in the dataset R at the time of the data download request and $P_c(R)$ is the probability that the cache contains the dataset R at the time of the data transfer request for a given cache size of c.

3.3 Proposed Services against UP Performance Problems

In this subsection, we explain three of the transmission strategies of the proposed system, which we use to evaluate the performance improvements in case of degradation of UP of mobile backhaul networks. We also present some of the factors affecting the end-to-end delay, their formulations in the context of caching, NC and the asymptotic behavior of NC. In case of transport network failure, these methods can be activated by the proposed system. These methods, described in detail below, are designed to be placed in the nodes as different services to be compatible with the service-based architectures of the Cloud RAN and CN and to benefit from the features of SO.

(i) Network Coding is a network technique in which certain algebraic operations are performed on data as it traverses nodes in a network [27]. Coding techniques generally provide improvements and robustness to the flow of information in the network by reducing data congestion in nodes or links throughout the network. NC helps to prevent data loss due to link breakage and provides intermediate nodes with the

ability to encode by combining multiple data links [28]. It can also provide robustness to packet loss, which can translate into an increase in throughput [29]. The NC solution not only aims to increase throughput, but can also provide additional solutions beyond applying additional layers of erasure coding in the intermediate nodes. In this paper, we use the NC-enabled transmission approach previously described in [21].

(ii) Caching can alternatively be used as a mechanism against delays in transport networks. Caching can be used to improve performance (reduce delays, backend logs, or downtime) in transport networks. The advantage is the time the UP packet has to spend in kernel-space. The proposed behaviour is equivalent to the specialized library of the Data Plane Development Kit (DPDK), which provides a mechanism for network acceleration by using the CPU cache [30]. Thus, UP packets are thrown directly into the CPU cache, allowing networks to run in user-space without sending packets into kernel space. The gain is the reduced time for processing from the T_{proc} time consumed by the mobile node.

The part dealing with caching and its advantages is shown in Fig. 2. Kernel space networking is shown in Fig. 2a. The process uses system calls to interact with the kernel space, which contains the corresponding network sockets via a Transmission Control Protocol (TCP)/Internet Protocol (IP) protocol stack. The network interface is responsible for physical and logical communication with the network. On the other hand, the user space networking of Fig. 2b can be thought of as an application in user space that contains TCP/IP network commands. The interaction with the network interface and network sockets is performed by user-space application. This application requires dedicated memory called the caching service. The more memory available to the user-space networking, the larger the cache and the better the capabilities of the network application. The UP packet delivery can be done bypassing the kernel by using user-space networking principles such as DPDK. Since caching is a separate virtualized service, user-space networking can be offered as a service either in container or Virtual Machine (VM) format. Thus, the advantages of user-space networking can also be exploited in this service-based design.

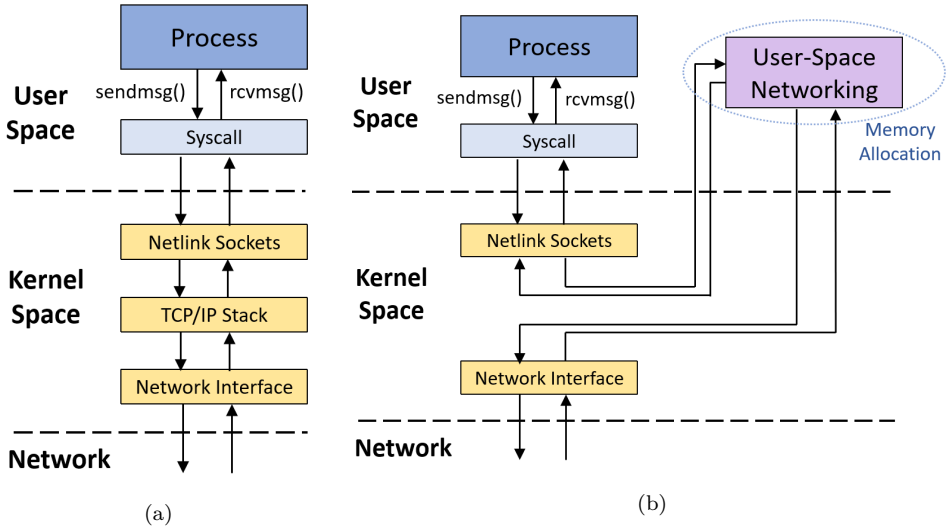


Fig. 2: Various networking implementations within the OS a) Kernel-space networking b) User-space networking.

(iii) Replication is a strategy for bundling (multiplexing) packets for data or/and control transmission over the air interface. It has been studied in detail in previous works by [31–33]. However, these proposals focus on transmission time interval (TTI) bundling or RLC segmentation for edge users to improve the reliability at the radio side. Moreover, the approach in [34] treats duplication as a separate service, but only for the wireless part of cellular networks. These approaches can be extended to the UP side to achieve the same gains on the UP side as shown in [16].

3.4 High-Level View of the Proposed System

Fig. 3 shows the proposed design of the system that integrates the proposed functionalities into the UP of the mobile network infrastructure via an end-to-end automation and orchestration framework. On the left side of Fig. 3, the O-RAN architecture can be seen in which the above functionalities are distributed to different modules of the architecture. With O-RAN, operators can save CAPital EXpenditure (CAPEX)/OPerating EXpenditure (OPEX) in various ways, e.g., by using cloud

computing hardware instead of in-house developed hardware, decoupling DU/CU hardware, cloud infrastructure and RAN application software, bundling CU capacity for DUs.

In the non-RT RIC within the SMO of Fig. 3, the UP caching functionality is located at the top level of the O-RAN stack. This functionality makes decisions at the second level of granularity. To execute the caching functionality at runtime, real-time control functions executing in the non-RT RIC can be distributed to the near-RT RIC. Near-RT RIC is the highest level control entity within the gNodeB/eNodeB and essentially provides data plane abstractions for the SMO layer. Near real-time has lower granularity than the second-level, and commands to the underlying network are transmitted through the E2 interface and commands from non-RT RIC are conveyed through the A1 interface. In the near-RT RIC there are both the NC and repetition functionalities described above.

On the cloud core side, there are core components of 5G cellular network, and all the functionalities studied in this paper are part of the UPF. In addition, there is a Cloud Core Orchestrator that is responsible for instantiating and managing the underlying services, the additional functionalities and their lifecycle, similar to SMO in the O-RAN architecture. To enable real-time network optimization and significant performance improvements, SMO and Cloud Core Orchestrator are also merged to achieve automation and end-to-end service orchestration. This is achieved by grouping different systems into domains (e.g., RAN and CN) and orchestrating across domains via open Application Programming Interfaces (APIs). Together with this approach, the provisioning of connectivity is shifting towards service-centric orchestration, which enables automatic provisioning of network services that can be triggered and configured directly by service providers and their customers.

In Fig. 3, the relevant RAN information from CU/DU is first collected by SMO. In the RAN domain, this step is performed in the O1 interface of the O-RAN architecture. The data collected at SMO is later shared with the non-RT RIC. When the SMO decides to activate caching, the activation command is sent to the non-RT RIC. Consequently, the UP caching activation command and policies are forwarded to specific radio functions that make the RAN components programmable in near-RT

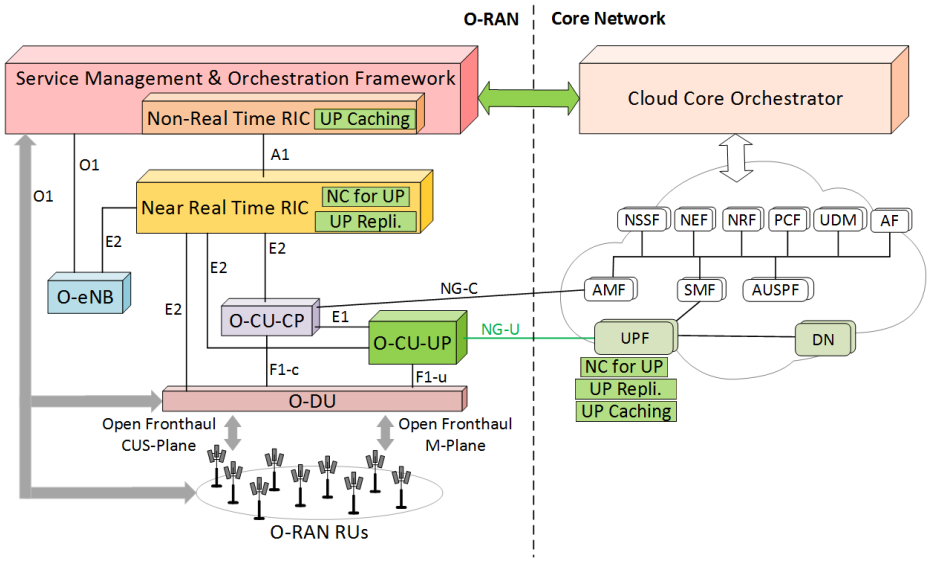


Fig. 3: Proposed architectural design of the studied functionalities with integration of O-RAN and CN in the UP.

RIC via the A1 interface of the O-RAN. Once the caching functionality is enabled, the solution is configured for use on CU/DU via the E1 interface. The same steps apply to other RAN functionalities and should also be followed in the CN. Note that our proposal is based on the cloud-based RAN and CN, as they are more compatible with the service-based architecture. However, the proposed architecture can also be applied to traditional mobile network systems.

3.5 Management of UP Healing Services

In the proposed architecture of Fig. 3, caching, replication or NC are provided as services on both the BS and CN sides. For this reason, the services that depend on these processes on the UP side are actually the services that need to be orchestrated and managed appropriately, since the user requirements and service level agreements are different for each use case. A sequence diagram of service orchestration and management for the proposed functionalities can be found in Fig. 4. The process of orchestration and management is described as follows:

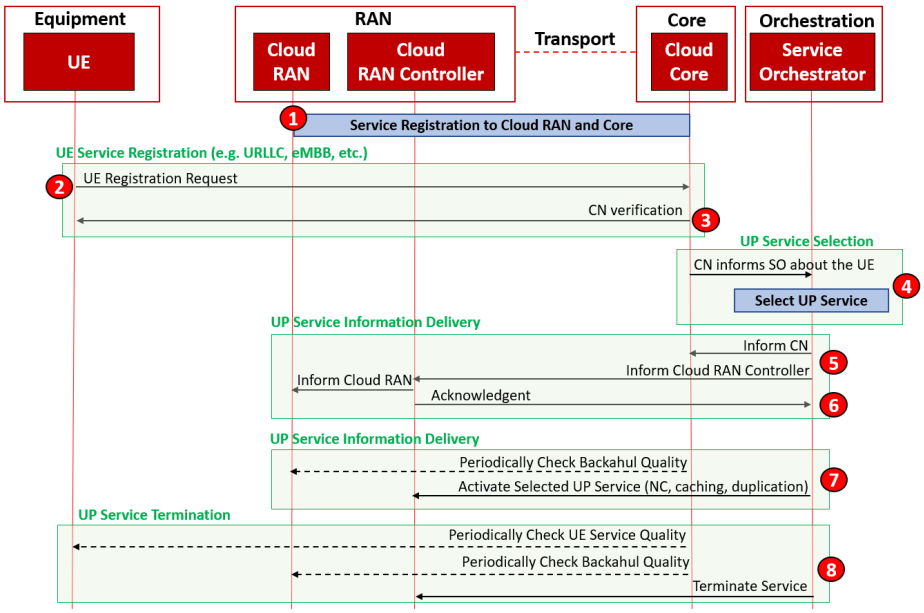


Fig. 4: Proposed sequence diagram for the selection and activation of UP service through the interactions between network nodes, SO and Cloud RAN controller.

1. The Cloud RAN controller registers the UP service with the service pool (caching, NC or duplication service) that the Cloud RAN node can manage. (Each Cloud RAN node may not be able to run all three services).
2. The UE establishes a connection and requests a network application service (e.g. URLLC, V2X, Enhanced Mobile Broadband (eMBB), etc.).
3. The CN verifies the request and the identity of the UE. Then, the CN determines the type of service (real-time, critical or non-critical) and the requirements for the UP service to be selected.
4. SO selects the most suitable UP service for the UE according to its requirements based on the information collected by the CN.
5. SO informs the Cloud RAN controller for this UE after selecting the UP service. The UP packets for this UE are registered and sent/received through the selected UP service (caching, NC or dynamic replication service).
6. Upon successful completion of the service, Cloud RAN controller informs the SO.

7. The CN periodically checks if UP performance issues exist in the backhaul network. If performance degradation occurs, the service is activated for the UE.
8. The CN checks and confirms if the service requirements are fulfilled. The service is satisfied if the UE uses its service (e.g., URLLC, etc.) that meets all requirements. The CN proceeds to check the backhaul quality. If there is no performance problem in UE and backhaul, the UP service is terminated.

3.6 User Plane Healing Flag

The nodes of the mobile network (C-RAN and UPF in CN) must be aware of the utilized UP delay prevention method. The reason is that the prevention method used for UP packets needs to be synchronized for processing by the receiver. In this study, we propose to insert a flag in the header of the standard 3GPP UP packet. This flag will occupy a 1-byte area in the header of UP. The proposed flag of UP can be positioned anywhere in the packet header and will be used as a *"UPH-flag"* (User Plane- Healing - flag). The main purpose of this flag is to inform the receiving end node that the sent UP packets are being processed either via caching, replication or the NC method.

For example, if the backhaul link (or the UP) is not lossy, there is no need to enable a UP failure prevention method. In this case, the UP packets can be sent to the BS or CN side with the *UPH-flag* set to $0x00$ by the transmitter node. This decision is made by the near-RT RIC. This means that the UP packets are not encoded, so normal communication continues without any changes. However, if a problem occurs in the transport network, one of the nodes (e.g., BS or CN) must set the *UPH-flag* of the UP header to $0x01$ to prevent packet loss. In this case, the transmission is performed with the *UPH-flag* set and the receiving end node knows that the NC operation has started. Then, the receiving node performs the decoding process by executing the decoding function *N(x)* which is preconfigured on both sides. When the loss rate increases, the near-RT RIC decides to switch to the UP packet replication method. In this case, the *UPH-flag* is set to the value $0x10$, indicating the replication process. On the other hand, the *UPH-flag* for the caching

method can be set to the value $0x11$, but in this case the caching decision for the UP is replaced by the non-RT RIC.

Note that UP exists only between BS and CN and that the flag inserted in the header of UP does not prevent the use of commercial UE when testing this solution on a testbed. Therefore, from the UE point of view, no changes are required, so the proposed solution is compatible with all UE types. To propose a general approach, the fine-grain address of the UP-flag (e.g., the address in the currently used GTP-U protocol) is not described in great detail, so UP protocols used in the future may also use the proposed methods. For example, in the packet type of the currently used GTP-U protocol, a field other than the UP-flag mandatory fields may be used for the proposed UP-flag. This may be the field reserved for the N-PDU number, or the fields reserved for the sequence numbers (by decreasing the number of seq number). For this reason, using the existing or empty fields does not require major changes on both the BS and CN sides.

4 Network Coding for User Plane

In this section, we elaborate on the NC operation in UP, since the NC operation has a detailed structure.

4.1 Implementation Issues

In the proposed real-time transmission method using NC, an additional encoding packet (p_{nc}) is created at the transmitter side based on XOR operations of the previously transmitted data packets. This additional encoding packet is kept ready without being sent to the receiver side and is recursively updated when new data packets arrive. It is not transmitted to the receiver until a certain number of data packets has been processed. This threshold is adaptively set depending on the link quality or the expected packet loss rate. After transmitting the encoding packet p_{nc}, when the threshold is reached, the whole cycle of the encoding process is repeated, starting first with an encoding packet containing zeros. This is shown in Fig. 5.

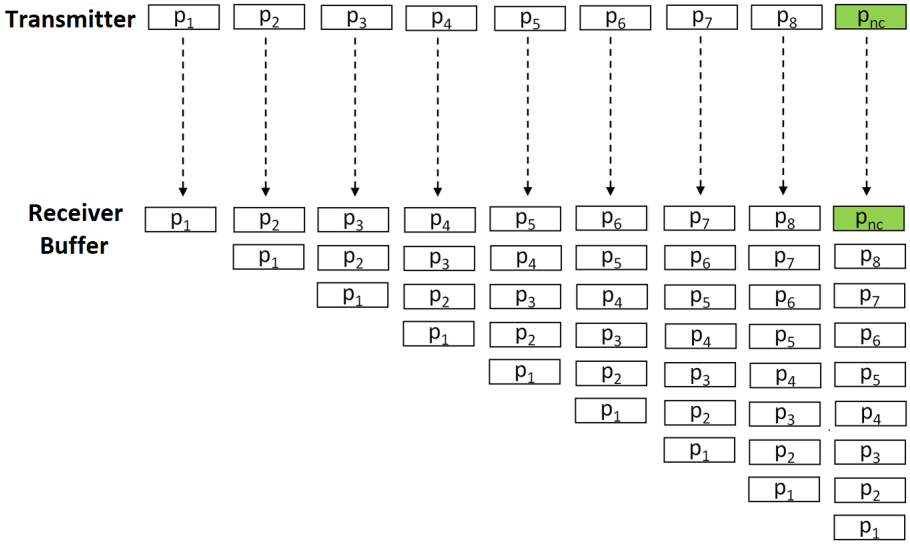


Fig. 5: An illustration of NC with buffered packets on the receiver side (threshold is eight).

The NC process in our approach is formally expressed as a recursive function in (5).

$$C(p_i, p_k) = \begin{cases} p_i & i = k \\ C(p_i, p_{k-1}) \oplus p_k & \text{otherwise} \end{cases} \tag{5}$$

where $k \geq i \geq 1$ and k, i are integers.

To illustrate the operation of the proposed method at the packet level, we give an example as shown in Fig. 5, where the threshold is assumed to be 8. In the context of this example, p_{nc} is initially equal to p_1, then updated as $p_1 \oplus p_2$, $p_3 \oplus p_2 \oplus p_1$, and so on, and finally the following value is obtained:

$$p_{nc} = p_8 \oplus p_7 \oplus p_6 \oplus p_5 \oplus p_4 \oplus p_3 \oplus p_2 \oplus p_1 \tag{6}$$

Due to iterative XORing, no additional memory is required to perform NC over a set of data. Now suppose that one of the transmitted packets in the streamed dataset $\{p_i, p_{i+1}, ..., p_k\}$ is lost, and let p_m be the lost packet where $i \leq m$ and

$m \leq k$. Then, the receiver can recover this lost packet p_m by XORing the remaining received $k-1$ data packets and the encoding packet p_{nc}. In other words,

$$p_m = p_{nc} \oplus \{p_1 \oplus p_2 \oplus .... \oplus p_{m-1} \oplus p_{m+1} \oplus ..... \oplus p_k\} \tag{7}$$

The overhead incurred by encoding packets is proportional to the threshold. If there are no lost packets at the receiver side or lost packets could not be successfully recovered, this extra network code packet is discarded. The decoder is able to recover a lost packet if the receiver has exactly one lost packet in the set of k packets. If the packet loss rate is higher than the NC capacity, then the transmitter retransmits all but one of the lost packets.

4.2 Adaptive NC Ratio Selection

The encoded packets must be stored in the buffer or cache of the mobile nodes for the reconstruction process. Unlike the implementation of NC in RAN, no session-based caching and NC operation are required on the UP side. In fact, the UP packets in the S1 interface may belong to different users, but they can all be encoded together. The execution of the NC operation within the nodes at runtime is described in Algorithm 1. First, the quality of the backhaul link over the BS - CN interface must be determined. This detection process can be based on the counters stored in the BS or CN interface (e.g. S1 Drop Rate, Success Ratio counters in Long Term Evolution (LTE)). If there is an additional link to determine the loss/drop rate of the backhaul link, any external measurement tool such as IP-SLA, Two-Way Active Measurement Protocol (TWAMP), etc. can be used. The "check packet loss ratio" step in Algorithm 1 depends on the counters or external meters. Therefore, depending on the system design, one of them can be selected to trigger NC in UP packets. When triggered, the packets are sent through the *UPH-flag* by setting it to 0x01. Otherwise, there is no need to set the *UPH-flag*. A pseudo-code representation of the proposed NC ratio can be found in Algorithm 1.

The decision for the most appropriate NC algorithm or NC ratio must be made when the nodes are activated. This adaptive approach gives us the flexibility to

Algorithm 1 Proposed method with adaptive network coding algorithm selection.

procedure CODING()
  Initialization:
  $Threshold \leftarrow f(E[PLR])$ ▷ determine the threshold based on the expected packet loss ratio
  $Counter \leftarrow 0$ ▷ counter keeping the number of data packets encoded
  $p_{nc} \leftarrow$ [0000...00000] ▷ all-zero encoding packet
  Network Coding:
  while $Counter < Threshold$ **do**
    $p \leftarrow data$ ▷ read data
    $p_{nc} \leftarrow p_{nc} \oplus p$ ▷ update the encoding packet by XORing itself with the data packet
    $Counter \leftarrow Counter + 1$
  end while
  return p_{nc} ▷ final state of the encoding packet to be transmitted
end procedure

change the NC ratio for different network conditions. At the beginning of the deployment, an NC algorithm is selected and then instantiated at both BS and CN sides. After this process, there may be cases where the NC algorithm needs to be changed. This may be due to changes in backhaul packet loss rates or bandwidth requirements in backhaul links. In this case, the NC algorithm may be modified by the MNO. Moreover, we need to check and determine if the same CN algorithms are consistently used on both the BS and CN sides. This is because if the CN algorithm is changed only in one of the nodes and the other node uses the previous CN algorithm, the encoding cannot be performed. This could affect the communication process.

An alternative solution would be to implement NC in all devices on the backhaul path. However, this assumes that all of these backhaul devices (routers / microwaves, switches, etc.) are NC-enabled. Unfortunately, it would be incredibly costly for an MNO to replace these devices with NC-enabled devices. In the proposed method, the encoding provides robustness to transport network failures, so that the UE does not need to perform any more processing for retransmission. As a result, fewer UE resources are consumed in the form of CPU, memory and battery, etc. Moreover, some

UE applications may be based on User Datagram Protocol (UDP) and are not able to perform retransmission. The proposed system also considers these types of UE applications where only a minor improvement in the packet structure of the UP protocol is required. This does not impose any new additional functional requirements on the current standard UE functions.

4.3 Asymptotic Behavior of NC

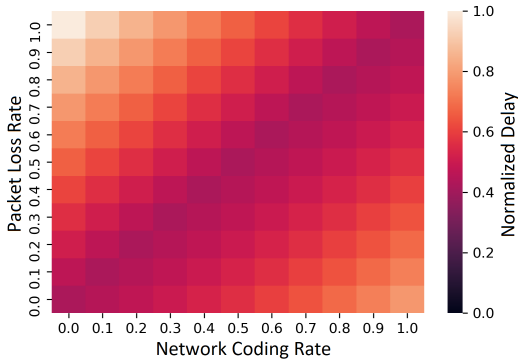
If the coding rate is equal to or higher than the expected packet loss rate, all lost packets should be recovered, but NC may cause some congestion and introduce additional delays. On the other hand, if the expected packet loss rate is higher than the coding rate, NC may recover some of the lost packets. Therefore, the unrecovered lost packets need to be retransmitted, which may cause additional delay due to retransmission. Based on this estimate, we approximate the end-to-end delay in a lossy channel with NC (T_{e2e}^{NC}) as follows,

$$T_{e2e}^{NC} = (1 + c_1.e^{|\zeta-\rho|} + c_2.max(\zeta - \rho, 0)).T_{e2e}, \tag{8}$$

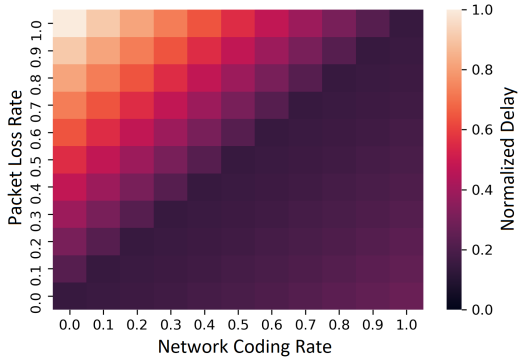
where ζ and ρ are the expected packet loss rate and NC rate respectively, and $\mathrm{T_{e2e}}$ denotes the end-to-end delay in a lossless transmission environment without NC. The coefficients c_1 and c_2 can be considered as weighting factors that adjust the cost in terms of delay for packet loss and NC. The optimal NC rate can be formulated as follows,

$$\rho_{optimal} = \arg\min_{\rho} T_{e2e}^{NC}. \tag{9}$$

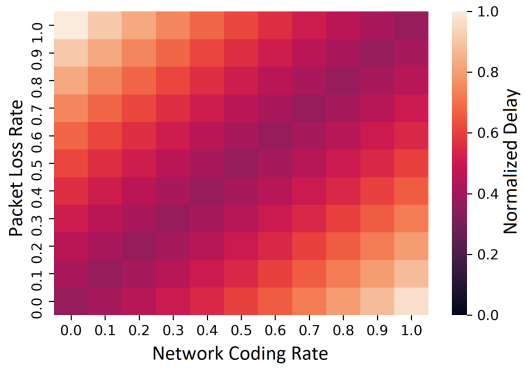
Fig. 6 shows how the end-to-end delay changes for different values of the weighting coefficients c_1 and c_2 when the packet loss rate and NC rate change. As can be seen in Fig. 6, the end-to-end delay takes a minimum value when the packet loss rate and the NC rate are equal. Otherwise, additional delay occurs either due to retransmission of unrecovered lost packets or due to excessive NC. In Fig. 6a, the packet loss rate and the NC rate have the same weight (i.e., $c_1 = c_2 = 1$). It can be seen that the



(a)



(b)



(c)

Fig. 6: Normalized end-to-end delay for different packet loss and network coding rates for different values of the weighting coefficients (a) $c_1 = 1$, $c_2 = 1$ (b) $c_1 = 1$, $c_2 = 10$ (c) $c_1 = 10$, $c_2 = 1$.

TABLE 2

SIMULATION PARAMETERS

Parameters	**Value**
Test Duration	20 s
S1 Packet Size	1500 bytes
Inter-packet Interval	0 ms
S1 Transport Bandwidth	1 Gbps
S1 Transport Delay	10 ms
Carrier Frequency	1800 Mhz
Bandwidth	20 Mhz
3GPP Channel Scenario	Urban
UE Mobility	Constant
MAC Scheduler	Proportional Fair (PF)
Subframe duration	1 ms
RLC buffer size for UEs	1 ms
eNodeB Power	46 dBm
Antenna Configuration	1x1
UE Traffic Type	TCP DL
Subframe Duration	1 ms
# of Resource Blocks	100
Noise Power Spectral Density	-179 dBm/Hz

worst case occurs when the packet loss rate takes the maximum value and the NC rate takes the minimum value (i.e., in the upper left corner), which is an expected situation.

Note that the heatmap in Fig. 6a is not symmetric according to the diagonal line, because the excess packet loss has a stronger effect than the excess NC, so that the lower right corner is darker than the upper-left corner. In Fig. 6b, the effect of packet loss rate on delay has increased by 10-fold by setting $c_1 = 1$ and $c_2 = 10$, which mitigated the negative effect of excessive NC as shown in the figure. In Fig. 6c, the effect of NC rate on delay increased 10-fold when $c_1 = 10$ and $c_2 = 1$ were set. This resulted in a nearly symmetric heatmap across the diagonal by balancing the negative effects of both events.

5 Experimental Analysis

We performed experiments in Network Simulator 3 (NS3) to demonstrate the benefits of the NC approach of [21], the dynamic replication approach of [16], and the UP caching approach proposed in this paper. We used the Lena Evolved Packet Core (EPC) module [35] in NS3 to simulate the S1 interface and use the NC implementation in this interface. Note that although we used the Lena module, which actually implements a 4G network, the results can still be applied to 5G NR since the evaluations and improvements are independent of the underlying radio access technology, but are based on improvements in backhaul networks. The parameters used for the simulation are listed in Table 2.

We conducted our simulation tests in a fixed packet loss environment. The packet loss model refers to the P2P model backhaul connection between eNodeB and CN in the Lena environment. In our evaluations, randomness is only used when it it comes to which packet is lost, i.e., one of the 100 packets is randomly discarded, but the discarded packet is randomly selected. For the backhaul link, we assume two scenarios, a single-hop and a two-hop connection between RAN and the CN. In our experimental setup, we compare a scheme with no coding (i.e., a regular UE application with retransmission in case of data loss), NC with coding rate of 1%, 2%, 5%, and 10%, and a dynamic replication scheme with 2 packets replicated in the mobile backhaul network. The relevant metrics used for comparison are total processing time, bandwidth usage, CPU and memory usage percentage. For the comparisons, we use the memory and CPU usage of the UP transmission in a lossless link as benchmark and compare the performances of the proposed schemes with them. For example, the percentage for the used method $x_{method} \in \{\text{caching, NC, dynamic replication}\}$ increases when using CPU as follow:

$$\text{RelativeCPU}(x_{method})(\%) = \frac{\text{UtilizationCPU}(x_{method}) - \text{UtilizationCPU}(lossless)}{\text{UtilizationCPU}(lossless)} \times 100\% \tag{10}$$

where UtilizationCPU($lossless$) is the CPU utilization of the UP transmission in a lossless link. Similar calculations to (10) are also performed for memory utilization.

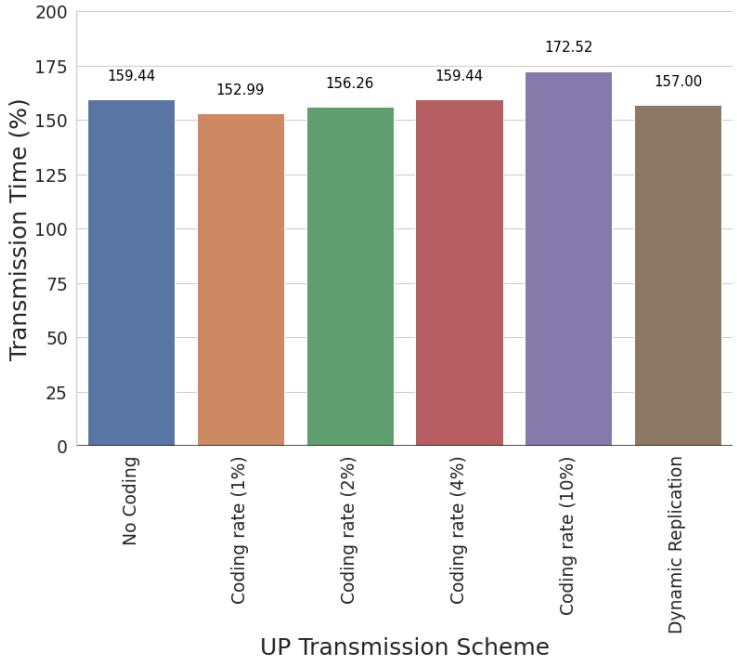
5.1 Evaluation of Results

5.1.1 Transmission Time and Bandwidth for NC

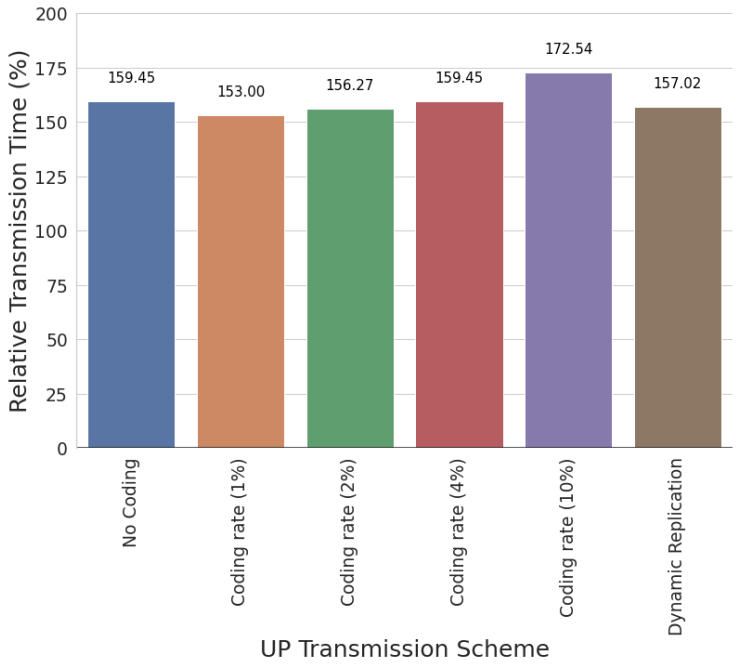
Fig. 7 shows the end-to-end transmission time performance results for the backhaul link with different NC rates at an exact packet loss ratio of 1%. Note that if the exact packet loss ratio of a connection is known, the exact NC ratio can be applied to combat packet loss. In Fig. 7a and Fig. 7b , the *x-axis* shows the transmission with different NC ratios and also the performance of the replication scheme for comparison. The *y-axis* represents the increased percentage of the total end-to-end transmission time required to transmit the same data compared to a lossless connection with the standard GPRS Tunneling Protocol – User Plane (GTP-U) transmission.

Fig. 7a and Fig. 7b show that when no coding is applied to the UP under an exact packet loss ratio of 1% in the backhaul link, 59.44% and 59.45% more T_{e2e} time are obtained compared to the standard GTP-U transmission for single-hop and two-hop scenarios, respectively. When NC is applied with a ratio of 1% (i.e., 100 packets are XORed and the XORed packet is transmitted), the T_{e2e} time increases to 52.99% and 53% for single-hop and two-hop scenarios, respectively. This corresponds to a time saving of almost 7% compared to a no coding transmission. An NC ratio of 2% generates 56.26% and 56.27% more T_{e2e} for single-hop and two-hop scenarios, respectively, but still saves 3.5% time compared to a no coding transmission.

In both Fig. 7a and Fig. 7b, for a backhaul link with an exact packet loss rate of 1%, above NC ratio of 5%, coding generates too much T_{e2e} time compared to an no coding transmission, which requires retransmission of the the lost packet in TCP session of UE. The dynamic replication scheme presented in [16] with two replicated UP packets sent to the backhaul has 57% higher time compared to a no coding transmission and its performance is worse than the NC with coding rate of 1% and 2%. Moreover, the tested dynamic replication scheme doubles the bandwidth used. Note also that these experimental results are consistent with our analytical calculations in Section 4.3.



(a)



(b)

Fig. 7: End-to-end transmission time performance comparisons under different NC ratios on backhaul links with an exact packet loss ratio of 1% a) Single-hop scenario b) Two-hop scenario.

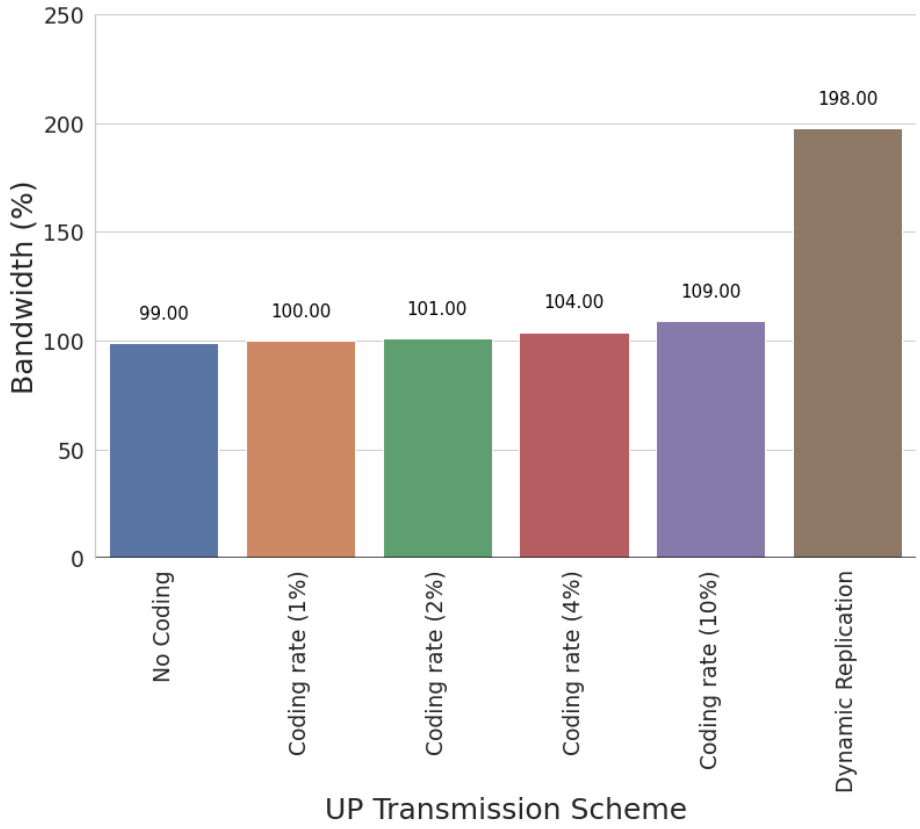
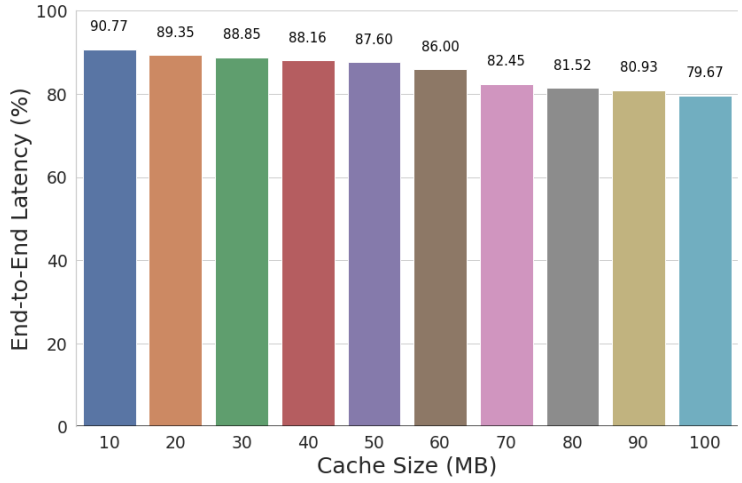
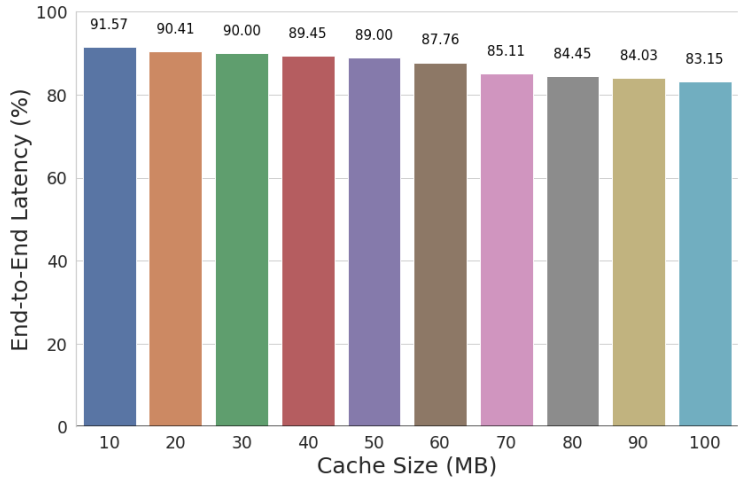


Fig. 8: Comparisons of bandwidth occupancy of different NC ratios on backhaul links with an exact packet loss ratio of 1%.

Fig. 8 shows the throughput/bandwidth characteristics of the compared methods. For the comparisons, we define the bandwidth utilization of the UP transmission in a lossless link as 100%. Then, the lossy link without coding enabled has a bandwidth utilization of 99%. When the NC coding ratio is 1%, 1% more packets are sent and the same throughput as the normal lossless transmission is achieved. However, when the NC coding ratio increases from 2% to 10%, the bandwidth utilization values increase slightly. On the other hand, the bandwidth utilization of the dynamic replication scheme [16] increases by 98% compared to the normal lossless transmission. These results indicate that network operators can achieve significant gains in bandwidth utilization by combining with NC-enabled backhaul.



(a)



(b)

Fig. 9: Performance comparisons of end-to-end latency reduction with increasing cache size versus no-caching a) Single hop scenario b) Two hop scenario.

5.1.2 Latency for caching

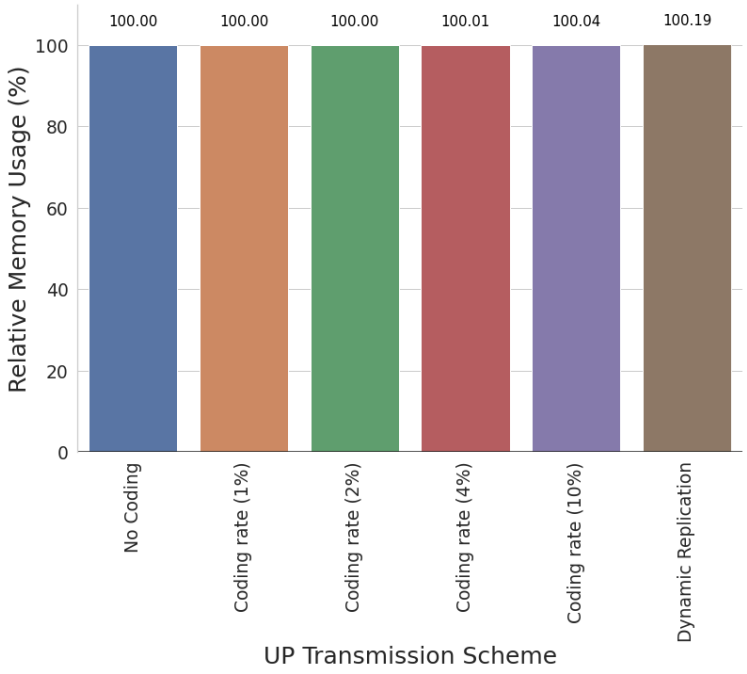
Fig. 9 shows the end-to-end latency when the same amount of data is sent over UP with different cache sizes for single hop and two hop backhaul scenarios. The data to be transmitted is kept constant at 100 MB and the delay in the connections is

zero. When the cache size is increased from 10 MB to 100 MB, the percentage of end-to-end latency decreases compared to no caching. As expected, the percentage of decrease is higher in the single-hop scenario than in the two-hop scenario. From Fig. 9a and Fig. 9b, it can be seen that a cache size of 100 MB can reduce the latency by about 20% and 17% in the single-hop and two-hop scenarios, respectively. Although it is always beneficial to use a large amount of cache, choosing 60 MB cache can be considered as a reasonable choice considering the trade-off between performance and the resources that the C-RAN spends on other processes. Note that the main purpose of C-RAN and the area where it consumes most of its resources are, of course, the user processes coming from the radio side and not the transport (namely UP).

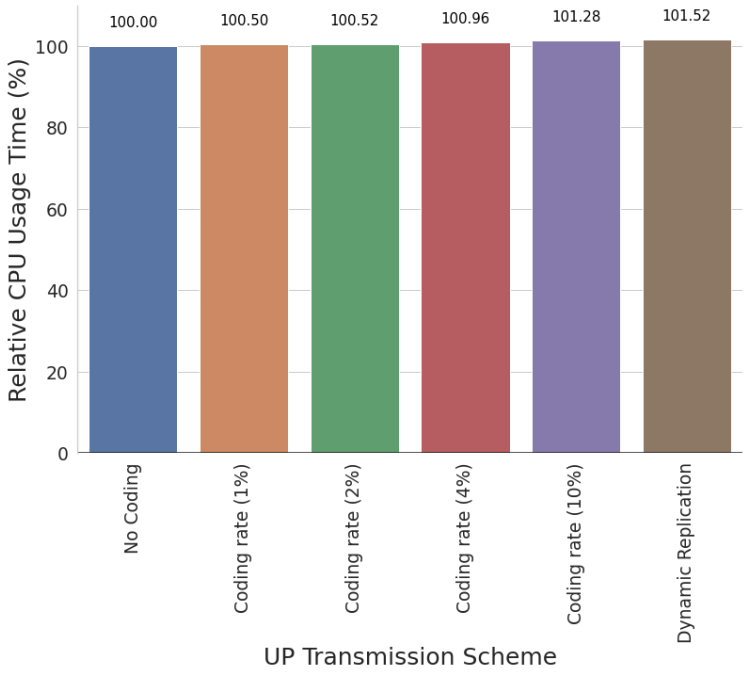
5.1.3 CPU and Memory Utilization

Fig. 10 shows the percentages of memory and CPU usage time of different NC ratios and replication for a backhaul connection with an exact packet loss ratio of 1%. When the NC coding ratio increases from 1% to 10%, both the memory and CPU usage values increase slightly. The method that consumes the most memory and CPU is the dynamic replication method, which is an expected result. From Fig. 10, it can be seen that the proposed NC and replication services for UP overload the systems in terms of memory and CPU consumption. At the same time, the memory and CPU usage of the dynamic replication scheme [16] increases by 0.19% and 1.52% respectively, compared to the normal lossless transmission. These results indicate that if the throughput in the NC-enabled backhaul links is sufficient, a NC with low ratios can be continuously enabled at runtime. This is due to the fact that the NC method places little load on the system at lower ratios.

Fig. 11 shows the relative CPU load values that occur when 100 MB data is sent over UP with different cache sizes at the node. The usage of CPU decreases as the cache size increases. This is because the cache hit ratio increases as the cache size increases. Therefore, a smaller number of CPU interactions are required. It is also observed that above a certain cache size, the usage rate of CPU decreases less. During our simulations, 100 Mb of data is transferred, which corresponds to a bandwidth of



(a)



(b)

Fig. 10: Performance comparisons of different NC ratios on backhaul links with an exact packet loss ratio of 1% a) Relative memory usage b) Relative CPU usage time.

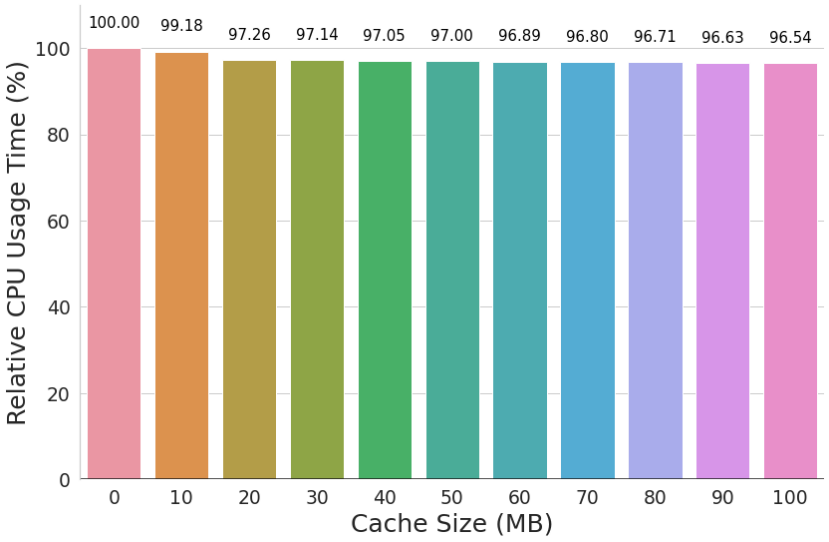


Fig. 11: Performance comparisons of relative CPU usage time with increasing cache size compared to no caching.

20 Mbps. At this bandwidth, the cache hit rate is not high. After this value, no cache hit rate is generated, which leads to a slowdown in the acceleration of the usage rate of CPU.

5.2 Benefits & Limitations of Various User Plane Healing Methods

In this section, we explain some of the advantages and limitations of the proposed NC model in UP compared to other traditional approaches in backhaul networks and the proposed dynamic replication approach of [16]. We have summarized these comparisons in Table 3 based on different dimensions. In our comparisons, we have labeled the currently used UP protocol as traditional, the replication scheme of [16] as dynamic replication and the proposed NC technique as network coded aware.

In the applications where the UDP protocol is used, there is indeed a trade-off between the quality of service/user experience and latency. However as can be seen from the results, the NC protocol with a low coding ratio does not seem to cause significant delays throughout the network. On the other hand, replication introduces

Table 3

Comparisons of Different UP Healing Methods of Mobile Backhaul.

Different Methods	Characteristics	Limitations	Advantages/ Benefits
Traditional UP	— UDP based. — No protection mechanism against packet loss. — Still valid UP protocol in 5G.	— Vulnerable to UP packet loss. — Completely dependant to the performance of backhaul.	— Standardized approach. — A long-time experienced protocol.
Dynamic Replication	— Based on UP packet replication. — Aims to decrease the packet loss probability in backhaul. — Replicated packets are dropped at the UE side.	— Occupies too much bandwidth. — Received replicated packets consume UEs processing resources. — Does not have any focus on service type (e.g. packets from eMBB services are replicated, but this is not too much needed for them) — Replication ratio can be blocked by backhaul bandwidth.	— No buffering needed. — No enhancement required in the UP protocol header.
Network Coded Aware	— Selected number of UP packets are coded together. — UP packets can belong to different UE. — Both mobile nodes (BS and CN) needs to be aware of coding process. — NC is terminated at mobile nodes.	— Minor modification needed in UP header. —Can increase processing time with high NC ratio.	— Creates less additional bandwidth. — Dynamically changing NC ratio provides flexibility based on location. — NC ratio increase will not be blocked by bandwidth limitation. — Relatively less processing then replication. — No additional processing on UE. — Preserves UE resources.
UP Caching	— Used for user-space networking. — Less time for packet processing. Bypass kernel like DPDK process.	— Reserved cache size can be limited. — Allocated cache can not be used for caching of the packets from the radio side.	— Ideal for non real-time usage. — This usage of caching can be implemented in software.

additional delays, but in this case SO can decide which solution it chooses to mitigate the backhaul problem. The policies listed in SO can help to find the appropriate solutions. For example, if the UE is connecting a UDP-based application such as a video transmission, small delays can be tolerated to allow smooth video streaming. If these micro-delays caused by the NC cannot be tolerated, the user will have to accept a possibly frozen/mosaic-like display and low video quality with small pixel losses.

6 Conclusions & Future Work

In this paper, we compare three different improvement strategies for avoiding problems that occur on the UP side of the C-RAN architecture. The solutions are intended for use in the O-RAN and CN architecture modules where repetition and NC-enabled transmission can operate in real-time mode, while caching can operate in non-real-time mode. The simulation results show that an exact loss rate of 1% in the backhaul link results in an additional total transmission time of 59.44% compared to a normal standard GTP-U transmission. Applying NC at a rate of 1% and 2% reduces this value to 52.99% and 56.26%, respectively. This is also better than the total transmission time of some previously studied dynamic replication schemes, while keeping the bandwidth utilization at low ratios. On the cache side, a reduction in latency of about 20% can be achieved with a cache size of 100 MB. In terms of CPU and memory usage, the proposed NC and replication services for the UP do not overload the systems.

At the end of the paper, we summarize some of the advantages and limitations of using these three strategies in UP of mobile backhaul networks. Further investigation on the possible application of other features of RAN directly on the UP side of mobile backhaul networks can be explored as a future research topic.

Conflict of Interest Statement

On behalf of all authors, the corresponding author states that there is no conflict of interest.

7 Acknowledgements

This work was partially funded by Generalitat de Catalunya grant 2017 SGR 1195 and the national program on equipment and scientific and technical infrastructure, EQC2018-005257-P under the European Regional Development Fund (FEDER).

References

1. Vodafone. Open RAN Press Release. `https://bit.ly/33gQFea`, 2016. [Online; accessed 10-Jan.-2017].
2. 3GPP Technical Specification. Service requirements for the 5G system. *3GPP TS 22.261 V16.6.0*, 2018.
3. C. Ge et al. Qoe-assured live streaming via satellite backhaul in 5g networks. *IEEE Transactions on Broadcasting*, 65(2):381–391, 2019.
4. A. J. Abu, B. Bensaou, and J. M. Wang. Interest packets retransmission in lossy ccn networks and its impact on network performance. In *Proceedings of the 1st ACM Conference on Information-Centric Networking*, pages 167–176, 2014.
5. M. Peng, Y. Sun, X. Li, Z. Mao, and C. Wang. Recent advances in cloud radio access networks: System architectures, key techniques, and open issues. *IEEE Communications Surveys Tutorials*, 18(3):2282–2308, 2016.
6. ORAN Alliance. O-RAN Architecture Description 3.0 WhitePaper. `https://www.o-ran.org/resources`, November 2020. [Online; accessed April-2021].
7. ORAN Alliance. O-RAN Use Cases and Deployment Scenarios WhitePaper. `https://www.o-ran.org/resources`, February 2020. [Online; accessed April-2021].
8. B Bob Briscoe, K. De Schepper, M. Bagnulo, and G. White. Low Latency, Low Loss, Scalable Throughput (L4S) Internet Service: Architecture. *Internet Draft draft-ietf-tsvwg-l4s-arch-08, Internet Engineering Task Force*, October 2021(Work in Progress).
9. IEEE Higher Layer LAN Protocols Working Group. Time-sensitive networking for fronthaul. *IEEE Standard 802.1CM*, 2018.
10. B. Balasubramanian et al. Ric: A ran intelligent controller platform for ai-enabled cellular networks. *IEEE Internet Computing*, pages 1–1, 2021.

11. R. Li et al. Intelligent 5g: When cellular networks meet artificial intelligence. *IEEE Wireless Communications*, 24(5):175–183, 2017.
12. E. Coronado, S. N. Khan, and R. Riggio. 5g-empower: A software-defined networking platform for 5g radio access networks. *IEEE Transactions on Network and Service Management*, 16(2):715–728, 2019.
13. E. Coronado, S. Bayhan, A. Thomas, and R. Riggio. Ai-empowered software-defined wlans. *IEEE Communications Magazine*, 59(3):54–60, 2021.
14. D. M. Coleman et al. Overview of the colosseum: The world's largest test bed for radio experiments. *Johns Hopkins APL Technical Digest*, 35(1), 2019.
15. S. Vakilinia and H. Elbiaze. Latency control of icn enabled 5g networks. *Journal of Network and System Management*, 28:81–107, 2020.
16. Y. Turk and E. Zeydan. A dynamic replication scheme of user plane data over lossy backhaul links. In *2019 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–7. IEEE, 2019.
17. F. Gabriel, G. T. Nguyen, R. Schmoll, J. A. Cabrera, M. Muehleisen, and F. H. P. Fitzek. Practical deployment of network coding for real-time applications in 5g networks. In *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pages 1–2, 2018.
18. A. Naeem, M. H. Rehmani, Y. Saleem, I. Rashid, and N. Crespi. Network coding in cognitive radio networks: A comprehensive survey. *IEEE Communications Surveys Tutorials*, 19(3):1945–1973, 2017.
19. N. I. Sulieman, E. Balevi, K. Davaslioglu, and R. D. Gitlin. Diversity and network coded 5g fronthaul wireless networks for ultra reliable and low latency communications. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–6, 2017.
20. W. Mao, M. Narasimha, M. Simsek, and H. Nikopour. Network coding for integrated access and backhaul wireless networks. In *2020 29th Wireless and Optical Communications Conference (WOCC)*, pages 1–6. IEEE, 2020.
21. Y. Turk et al. Network coding aware user plane for mobile networks. In *16th International Conference on Network and Service Management (CNSM)*, pages 1–5. IEEE, 2020.
22. R. Leszko. Where Is My Cache? Architectural Patterns for Caching Microservices. `https://bit.ly/3eRpP1R`, September 2019. [Online; accessed April-2021].
23. J. Hwang, K. K. Ramakrishnan, and T. Wood. Netvm: High performance and flexible networking using virtualization on commodity platforms. *IEEE Transactions on Network and Service Management*, 12(1):34–47, 2015.
24. D. Barach et al. High-speed software data plane via vectorized packet processing. *IEEE Communications Magazine*, 56(12):97–103, 2018.
25. E. Zeydan et al. Performance monitoring and evaluation of fttx networks for 5g backhauling. *Telecommunication Systems*, 77(2):399–412, 2021.

26. D. P. Bertsekas, G. G. Robert, and H. Pierre. Data networks. *New Jersey:Prentice-Hall International*, 2, 1992.
27. S-YR Li, R. W. Yeung, and N. Cai. Linear network coding. *IEEE Transactions on information theory*, 49(2):371–381, 2003.
28. C. Raiciu et al. Improving datacenter performance and robustness with multipath tcp. *ACM SIGCOMM Computer Communication Review*, 41(4):266–277, 2011.
29. P. Ostovari, J. Wu, and A. Khreishah. Network coding techniques for wireless and sensor networks. In *The art of wireless sensor networks*, pages 129–162. Springer, 2014.
30. T. Begin, B. Baynat, G. G. Artero, and V. Jardin. An accurate and efficient modeling framework for the performance evaluation of dpdk-based virtual switches. *IEEE Transactions on Network and Service Management*, 15(4):1407–1421, 2018.
31. C. P. Li et al. Tti bundling for urllc ul/dl transmissions, April 12 2018. US Patent App. 15/480,019.
32. C. Ramesh and V. S. S. Karthik. A novel adaptive tti bundling scheme in lte system. In *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–4. IEEE, 2017.
33. E. Spaho, L. Barolli, and F. Xhafa. Data replication strategies in p2p systems: A survey. In *2014 17th International Conference on Network-Based Information Systems*, pages 302–309. IEEE, 2014.
34. T. De Schepper et al. Orchestra: Supercharging wireless backhaul networks through multi-technology management. *Journal of Network and System Management*, 28:1187–1227, 2020.
35. N. Baldo et al. An open source product-oriented lte network simulator based on ns-3. In *Proceedings of the 14th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '11, page 293–298, New York, NY, USA, 2011. Association for Computing Machinery.