applied sciences

MDPI

Article

Differentiated Content Acquisition Integrating Network Coding and Edge Computing for Industrial Internet

Yunmin Wang [1,2,*,†] and Hui Li [2]

[1] Liuzhou Vocational & Technical College, Liuzhou 545006, China
[2] School of Electronic and Computer Engineering, Peking University, Shenzhen 518055, China
* Correspondence: wangyunmin@pku.edu.cn
† This work was completed when Y. Wang studied at Peking University.

Abstract: The construction of the Industrial Internet has become a concrete implementation and an essential starting point in accelerating the digital transformation and intelligent upgrading of industrial manufacturing enterprises. The problem of addressing and forwarding for resource-constrained devices restricts data acquisition and dissemination on the Industrial Internet. In order to retrieve content reasonably, we propose IDEANE, an identity-differentiated content acquisition and multipath forwarding scheme with network coding and edge computing. In IDEANE, content requests are disseminated based on the identity and location information carried in a multi-identifier network, which can improve the efficiency of interest message requests and reduce idle links in the network. Moreover, the content acquisition computation is offloaded to multiple edge nodes, and the encoded data are transmitted to the edge nodes for recovery. IDEANE offloads the less demanding computing tasks to the edge nodes close to the content requesters, to relieve the pressure on providers. In addition, a collaboration method among multiple edge nodes is also studied. Multiple edge nodes collaborate to support the mobility of content requesters, save energy consumption in terminal devices, reduce transmission latency, and ensure data security. The experimental results show that IDEANE can avoid duplicated transmission, reduce the network link overhead, improve network throughput, and enhance network robustness and reliability.

Keywords: IDEANE; content acquisition; multipath forwarding; network coding; edge computing; multi-identifier network; Industrial Internet

check for updates

Citation: Wang, Y.; Li, H. Differentiated Content Acquisition Integrating Network Coding and Edge Computing for Industrial Internet. *Appl. Sci.* **2023**, *13*, 7129. https://doi.org/10.3390/app13127129

Academic Editor: Ugo Vaccaro

Received: 18 May 2023
Revised: 8 June 2023
Accepted: 12 June 2023
Published: 14 June 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, with the vigorous development of the Industrial Internet, it has become a research hotspot and has attracted extensive attention from industry and academia [1,2]. The main challenge is the transmission and aggregation of heterogeneous data that belong to various owners in different locations. Identifiers are the fundamental elements of data sharing and can ensure the positioning, control, and management of identified objects, as well as the processing, exchange, and sharing of relevant information [3]. Universal network identifiers include the handle, GS1, IP address, and content name identifier.

Route scheduling, as the basis of data forwarding, is leveraged to discover the optimal path from the content requester to the provider. Industrial Internet identifier resolution services can realize the addressing, routing, and deep interconnection of massive Industrial Internet resources. The system and protocol are made more concise by addressing and routing through name identifiers. In addition, they can conceal the locations of publishers, resist host attacks, and bring higher security. Simultaneously, the ubiquitous caching mechanism can reduce the routing delay of massive data. Identifier-based routing supports multicast, service group scenarios, and data retrieval from multiple distributed nodes [4].

The network transmission of the multi-identifier network for Industrial Internet (MIN-IIoT) is mainly carried out through name identifiers [5]. The overall naming is similar to the

hierarchical naming scheme in Named Data Networking (NDN, [6]). The forwarding and routing strategy of the data plane principally depends on four tables, namely the content store (CS), pending interest table (PIT), forwarding information table (FIB), and identifier translation base (ITB). End users can employ these tables to look up the local identifier space or search the other identifier spaces with the identifier's mutual translation technology until they find the identifiers that can locate the required content. In addition, they can route and forward what they need in the multi-identifier network with the found identifiers. In order to ensure data security and non-repudiation, MIN-IIoT also carries an identifier of user identity.

MIN-IIoT supports adaptive multipath forwarding by introducing CS and PIT structures to record the pending interest messages and returned data messages [7]. In addition, it supports two types of forwarding strategies. One is to choose the optimal route (BestRoute [8]) based on the condition of the current network interfaces. All traffic transmitted on the network will be forwarded through the pre-selected optimal path with the lowest routing cost. Although routing migration can eliminate this problem, routing selection requires much overhead. The other forwarding policy is to actively forward messages in parallel along multiple interfaces provided by the Forwarding Interest Table (FIB), causing the redundancy of content packets brought back by different links. Network coding [9,10] can take advantage of the in-network cache and multipath transmission of MIN-IIoT; in addition, it can realize the parallel transmission of Industrial Internet data produced by multiple providers on different network links through various communication modes (such as 5G and WIFI) [11].

Cloud, edge, terminal integration, and edge–cloud collaboration represent the major Industrial Internet development trends [12]. Edge gateways and servers near the industrial site can collect, upload, process, and analyze industrial equipment data through edge computing. Edge computing realizes location awareness and determines the equipment location at a relatively reasonable cost. It is convenient to quickly cooperate with the geographic location identifier in the multi-identifier network (MIN-IIoT) to locate Industrial Internet objects. In addition to encoding at the content provider, network coding can support secondary encoding at intermediate nodes, reducing the dependence on original data and enhancing the transmission reliability. Edge nodes can acquire arbitrary encoded blocks from the providers or intermediate nodes. Once enough linearly independent encoded blocks have been collected, the edge nodes can decode and restore the original content. This encoding and decoding process is independent of the content providers and encoded blocks' acquisition paths.

In MIN-IIoT, identity authentication prevents the risk of data leakage caused by the encoding and decoding of intermediate routers. In order to reduce the pressure of the intermediate nodes without destroying the security characteristics of the multi-identifier network, we only perform the decoding operations on the edge nodes. Edge nodes can provide services for the connected terminal nodes, which can avoid redundant encoding and decoding operations at the terminal nodes and solve the problem of insufficient storage and computing resources on the terminal nodes. The integration of network coding and edge computing is particularly suitable for scenarios with solid mobility and real-time requirements, such as Internet of Vehicles.

This paper mainly focuses on content acquisition with the integration of edge computing and network coding in the Industrial Internet. It solves the problem of idle links and redundant transmission in multipath forwarding through name identifiers. A multipath forwarding strategy is proposed based on the identities of content requesters. Then, a multi-edge node cooperation method is designed to support the content requester movement. In summary, the essential contributions of this paper are listed as follows:

- An identity-differentiated content acquisition with network coding and edge computing multipath forwarding strategy, IDEANE, is designed to differentiate content requests through identity identifiers in MIN-IIoT. The hash values of the identity identifiers of the content request initiators are encapsulated in the request interest

messages and mapped to the PIT and CS of the intermediate nodes. The intermediate nodes can avoid redundant responses to interest messages requesting the same encoded blocks by identifying the content requesters' identities. In addition, they can ensure that the interest messages from the same content requester, aggregated through multiple paths, can acquire as many linearly independent encoded blocks as possible. At the same time, in order not to degrade the security performance of the MIN-IIoT system, the encoded blocks are only verified at the intermediate nodes without secondary encoding, thus reducing the network transmission delay.

- In the event that the content requester moves, the encoding and decoding tasks are offloaded to multiple edge nodes. The requests for the encoded blocks and the restoration of the original blocks are completed through the cooperation of numerous edge nodes. The request transmission delay is reduced, and redundant data transmission is avoided.
- Simulation results show that IDEANE can reduce the transmission delay of content requests compared with NetCodCCN [13], which encodes and decodes data blocks in intermediate nodes. At the same time, it can quickly meet the content requirements of mobile requesters through the collaboration of multiple edge nodes.

The remainder of this paper is organized as follows. Section 2 reviews the related work on image anomaly detection. A detailed description of the identity-differentiated content acquisition strategy is given in Section 3. We propose the collaboration algorithm for multiple edge nodes for content requesters' mobility in Section 4. Section 5 presents the corresponding experiments and analyzes the simulation results. Section 7 concludes the paper and discusses future research.

2. Background and Related Work

The original purpose of network coding is to enable multicast transmission to reach the theoretical maximum transmission capacity to achieve better network throughput than routing multicast [9]. The network coding node does not need to know the entire network's topology and other nodes' coding processes. The added content is divided into K blocks; each encoded block contains the original data information. The nodes in the network collect any M-block encoded blocks ($M \geq K$) from the information flow and sufficient independent decoding vectors, forming a decoding matrix through linear combination. The decoding is performed using the Gaussian elimination method to obtain the original information sent by the content provider. Network coding can overcome the problem of retrieving the original content sequentially during the content forwarding process. Network coding shows its advantages in improving network throughput, increasing the load balance capability, reducing transmission delays, reducing energy consumption, and enhancing network robustness [10]. It can be widely used in ad hoc networks [14], P2P content distribution [15], distributed file storage [16], network security [17], Internet of Vehicles [18], and other fields.

The advantages of network coding in multicast networks are consistent with the nature of content center networks to support intra-network caching and stateless multipath forwarding, which can promote the routing and forwarding of network identities and effectively meet the real-time and reliability requirements of the Industrial Internet [5]. NDN and MIN-IIoT combined with the network coding environment request not the original data blocks but the encoded blocks of the same content. Montpetit et al. verified that network coding could be manipulated in Information Center Networks (ICNs) to accelerate network multipath forwarding and improve network throughput and security [19]. NetCodCCN introduces the concept of *generation*, where the encoding and decoding of network data blocks are carried out in one generation, which can support the aggregation and pipeline transmission of interest messages, thereby reducing the content retrieval latency and data retrieval time [13]. At the same time, NetCodCCN allows nodes to maintain the information of content segments sent on each network interface to reduce the number of duplicate pieces. By introducing network coding technology, the bottleneck problem of path overlap

points can be solved, the throughput of transmission bottlenecks in multicast paths can be improved, and the elasticity of packet transmission pressure can be enhanced.

The subsequent NC-NDN led to On-Demand Off-Path Cache Exploration-Based Multipath Forwarding (O2CEMF), which can perform off-path caching based on the reachable information of the reactivity detection cache. Simultaneously, the generation in NetCod-CCN is adopted as the granularity to maintain the cache reachability information [20]. MICN is based on a linear independent content-based multiple interest message (MILIC) structure, which can constrain received interest messages and group data-encoded packets [21]. MICN shows that it can guarantee the reliability of network transmission in the network environment with a high packet loss rate. Therefore, it can offer corresponding benefits in the complex and changeable Industrial Internet by introducing network coding technology and coordinating the allocation of network resources and bandwidth between multiple content consumers, content producers, and intermediate nodes to leverage network resources and reduce transmission delays effectively. However, all of these schemes encode data packets at intermediate nodes to respond to user interest requests, which will undermine the security characteristics of NDN [11]. The multipath forwarding strategy proposed in this paper only performs network coding at the content provider and decoding operations at the edge nodes.

Mobile edge computing (MEC) [22] and edge intelligence (EI) [23], in close proximity to the industrial manufacturing site, use the computing resources and network resources of edge gateways or servers as edge computing carriers to set up a bridge between terminal devices and cloud servers. Edge computing can reduce Industrial Internet data transmission delays, meet real-time response time requirements, reduce energy consumption, lower bandwidth costs, and solve data security and privacy issues [24]. These advantages have prompted edge computing to be widely exploited in the Internet of Things [25], Internet of Vehicles [26], and Industrial Internet [27]. At the same time, edge intelligence can also provide a method of modeling and deploying digital twins [28] for the Industrial Internet, guaranteeing the processing of large amounts of industrial flow data with low latency and high security [29] and promoting the development of intelligent manufacturing [30].

In recent years, many researchers have begun to study the combination of network coding and edge computing to solve the routing and forwarding problems in the Internet of Things network. Zhou et al. [31] used the characteristics of routing packet correlation on wireless links to discuss the impact of link correlation on the number of packets to be forwarded and the selection of forwarders in intelligent routing. Then, they leveraged network coding combined with opportunistic routing to improve wireless IoT. For the energy efficiency of networking infrastructure, a method based on link correlation mining for intra-session network coding for intelligent routing (NCCLC) was proposed to accurately estimate the number of transmissions required by the forwarder and select the forwarder with the optimal number of communications for routing. Phung et al. [32] claimed that the REST HTTP protocol could provide WEB services in the Internet of Things. It is compatible with existing computing infrastructures and is selected as a standard protocol for IoT application communication. Their proposal combines REST HTTP with Random Linear Network Coding (RLNC) to reduce the extra packet weight consumed and to ensure reliable data exchange using a stateless REST HTTP protocol in unreliable and highly variable IoT scenarios. He et al. built a mobile edge computing intelligent cache analysis platform, which can pre-determine which streaming media files are stored in the limited storage space of edge computing devices and provide users with an efficient, intelligent network experience. Then, the network coding and wireless relay cooperation method is used to simultaneously transmit the files extracted from the core network and the real-time regular communication network to the network's edge, thereby realizing the intelligent caching of mobile edge computing [33].

The ideas adopted in these recent studies inspired our design. However, they usually employ machine learning algorithms to facilitate the multipath forwarding of content requests. These existing strategies consume a tremendous amount of resources for model

training. Although this will also be a future research direction in terms of improving the multipath forwarding capabilities, we aim to design an uncomplicated scheme that can be used in resource-constrained Industrial Internet environments.

3. Identity-Differentiated Content Acquisition Strategy

With the massive increase in data generated by the Industrial Internet and the expansion of the network scale, enormous pressure will be placed on the multi-identifier system (MIS) in MIN-IIoT if all identifier registration, inquiry, and management needs to be handled by the top-level management organization of MIS. In the process of layer-by-layer transmission of the network, the transmission bandwidth, physical distance, and data processing will bring an inevitable time delay, which reduces the ability to meet the real-time requirements of a rapid interactive response in the industrial field. At the same time, the process of network interaction may also affect the security and privacy of user data. On the other hand, the performance of terminal equipment is constrained in terms of computing processing capacity, network bandwidth, storage resources, and energy efficiency. Therefore, a more practical solution is to exploit edge computing technology to directly offload some computing tasks to the edge network near the terminal devices, and to take advantage of the resources of edge gateways and edge servers to complete the data processing nearby.

The interior of the Industrial Internet can be regarded as a directed acyclic graph (DAG, $G = (V, E)$), and the network routing of data transmission is a routing process based on the network topology, as shown in Figure 1. Network coding will consume many computing and storage resources, increasing the network costs. At the same time, it is necessary to consider maximizing the utilization of network resources through load balancing. This study makes full use of the necessary identity identifier for each Industrial Internet object in MIN-IIoT to achieve a tradeoff between load and delay. It regards the identity information as a part of the network packet. An identity-differentiated content acquisition combining network coding with edge computing (IDEANE) algorithm is designed to leverage idle network paths effectively, balance the loads of multiple forwarding paths, and direct the content to more edge nodes.

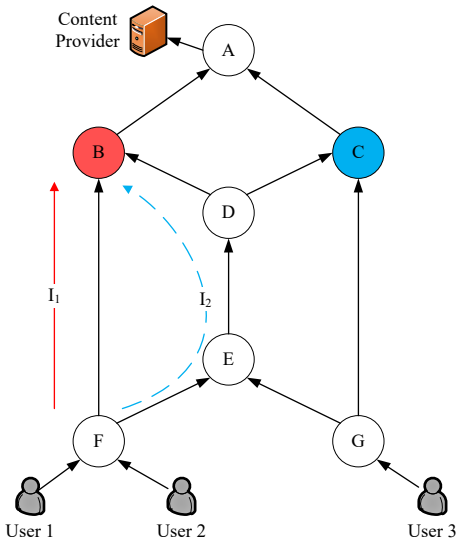


Figure 1. Interest message aggregation process in butterfly network.

3.1. Analysis of Conventional Forwarding Methods

When transmitting content, MIN-IIoT divides the original content into small data slices to facilitate data communication. In traditional NDN, it is necessary to acquire and combine all data blocks in a sequence to form the original data. In the case of network coding, an interest request for content is not a request for a particular data chunk of the original

content but a request for any encoded block of the specific content. We adopt the generation concept introduced by NetCodCCN [13]. In order to reduce the dimensionality of encoding vectors, reduce the complexity of encoding and decoding computing, and facilitate network transmission, we combine several file blocks into a generation and encode them within the same generation. In the future, we will treat generation as an encoding group. The required group number needs to be added after the name identifier when requesting content. In the name identifier $/pku.edu.cn/web/promo.mp4/1$, $/pku.edu.cn/web/promo.mp4$ is the name of the content, and 1 is the group number.

In MIN-IIoT, handling interest requests for the same content name involves simply looking up the content store (CS) or creating or modifying an FIB entry. Firstly, we analyze two cases of requesting content through unmodified MIN-IIoT.

Scenario 1: Assume that, in Figure 1, User 1 and User 2 send two interest messages, I_1 and I_2, respectively, and request the content provider A to provide two linearly independent encoded blocks named $/pku.edu.cn/web/promo.mp4$ for encoded group 1 of content CC, with the name identifier as $/pku.edu.cn/web/promo.mp4/1$. Assuming that edge node F has PIT and CS, I_1 and I_2 will be aggregated in the pending interest table PIT_F of edge node F due to the same content name. According to the normal multi-identifier network forwarding process, only one interest message is sent to the provider A. The data will be distributed to Users 1 and 2 until they are returned.

Scenario 2: Assuming that edge node F does not have PIT and CS, User 1 sends two interest messages, I_1 and I_2, through node F, with $/pku.edu.cn/video/promo.mp4/1$, to request two linearly independent encoded blocks of encoded group 1 provided by the content provider A. The path of I_1 is $F \rightarrow B$, while the path of I_2 is $F \rightarrow E \rightarrow D \rightarrow B$, and node B is the convergence point of the two interest messages. According to the path in the figure, assuming that I1 arrives at node B before I2, if the traditional NDN-BestRoute forwarding strategy is adopted, only $F \rightarrow B \rightarrow A$ is selected as the preferred forwarding path for the communication between content request node F and the provider A. The I_2 interest message may be discarded due to routing reasons, resulting in the waste of I2 transmission. If a multipath forwarding strategy is adopted, I_1 first arrives at node B. Then, it creates a table entry for content encoded group 1 in B's pending interest table PIT_B, as shown in the first line of Table 1. Subsequently, two situations are also considered.

Table 1. Changes in node B's pending interest table.

Content Name	Request Face
/pku.edu.cn/web/promo.mp4/1	1
/pku.edu.cn/web/promo.mp4/1	1, 2
/pku.edu.cn/web/promo.mp4/2	2

Situation 1: Before I_2 reaches node B, if node A has already returned encoded data for interest I_1, it will cache the data in node B's content store CS_B. After I_2 reaches node B, it directly retrieves the data from CS_B and returns them to node F along $B \rightarrow D \rightarrow E \rightarrow F$ in the opposite direction of I_2.

Situation 2: After I_2 arrives at node B, node A has not yet returned a data message. A request interface for the same content will be added to the PIT, as shown in the second line of Table 1, until node A responds to interest I_1 and returns the encoded packet to node B. Node B responds to node F along the opposite direction of I_1 and I_2.

Whether in Situation 1 or 2, node F will receive two copies of the same encoded block of content CC, resulting in redundancy in network transmission. Meanwhile, in Situation 2, waiting for the data message to return from the content provider A to node B will cause the I_2 incoming path to be idle.

3.2. Design and Analysis of MIN-IIoT Messages

According to the analysis in the previous section, the traditional MIN-IIoT multipath forwarding method will not only bring about network idleness but also network redun-

dancy. At the same time, this method fails to meet the requirements of sending different interest messages to obtain different encoded blocks. Consequently, we modify the name identifiers in MIN-IIoT and design the IDEANE algorithm. An encoded flag segment is added for compatibility. If the flag is set to 0, the original data are requested, and the encoded group number segment is used as the number of the original data block required. Otherwise, an encoded packet is requested.

The format of a content request interest message for an encoded block is shown in Table 2. The message's content name identifier can be */pku.edu.cn/web/promo.mp*4/1/1, representing the first encoded group with a request name of */pku.edu.cn/web/promo.mp*4. At the same time, a hash value of the user identity is added to the interest message to distinguish content interest requests from different edge nodes. The edge nodes need to continuously send request messages to the network until they have collected sufficient encoded blocks, which means that if the content grouping is divided into K blocks, the collection of K linearly independent coding blocks can meet the requirement. Suppose that the edge node has already received an M-block linearly independent encoded block. In this case, it needs to continue sending at least $K - M$ interest messages to request the remaining encoded blocks from the network. Therefore, a counter *ReqNo* is set in the interest message as the parameter carried by the interest message, which stores the serial number of the interest message requesting the same encoded group. It distinguishes different interest messages from the same edge node requesting the same encoded group through the same interface in the same intermediate router PIT. In IDEANE, these interest messages will be routed and forwarded to support concurrent interest message requests in the network.

Table 2. Interest message format for contest request.

Field		Symbol	Illustration and Explanation
Name Identifier	Content Name	CntName	/pku.edu.cn/web/promo.mp4
	Encoded Flag	EncFlag	Whether to request an encoded block, where 0 represents the request for a original block
	Encoded Group No.	GenID	1, 2, 3...
Random Number		Nonce	Name and nonce jointly distinguish unique interest to prevent replay attacks
Interest Command Parameter		IntCmd	REQDATA
Hash of Identity Identifier		IdHash	0cc175b9c0f1b6a831c399e269772661
Interest No.		ReqNo	20

As shown in Table 3, the data message format has been modified for response messages. The encoding vector carried in the data message can be employed to collect enough linearly independent encoded blocks in the edge nodes and decode them to obtain the original data blocks. It can also be used to check the linear correlation of the received encoded blocks in various network nodes.

After receiving an interest message request, the intermediate node modifies its PIT table. It adds a hash field to store the identity identifiers of all requesters applying for the same content, facilitating the identification of requesters. At the same time, it adds the interest message number sent by the same content requester for the same encoded group to distinguish different interest message requests.

Table 4 shows the modified PIT table: {*encoded group name of content requested*, <*source port, identity hash, request sequence number*>}. The advantage is that, on the one hand, the compression feature of the hash algorithm is used to reduce the size of the PIT table; on the other hand, the fast retrieval feature is exploited to facilitate the precise retrieval of the identity of the content requester.

Table 3. Data message format for content response.

Field		Symbol	Illustration
Name Identifier	Content Name	CntName	/pku.edu.cn/web/promo.mp4
	Encoded Flag	EncFlag	Whether to request an encoded block, where 0 represents the request for a original block
	Encoded Group No.	GenID	1, 2, 3...
Signature Information		SignInfo	...
IData		Data	Encoded data blocks
Encoding Vector		EncVector	[1, 0, 0, 0, 0, 0]
Interest No. Satisfied		ReqNo	20

Table 4. PIT and CS organization after modification.

Table	Content Name	Extra Information	Encoding Vector	Data
PIT	Name of the requested encoded group	< Source port, identity hash, request no. >	/	/
CS	Name of the cached encoded group	Bloom filter of identity hash	✓	✓

We have also made modifications to the CS table, with the structure of {*cache content encoded group name, identity hash Bloom filter, encoding vector, cache content*}. The Bloom filter is initialized to 0 and constantly modified as the content cache changes. After receiving the data response, the intermediate node will delete the corresponding PIT entry, cache and save the content in the CS, and write the identity information corresponding to the requested encoded group in the PIT table entry into the Bloom Filter of the CS, to utilize the time and space efficiency of the Bloom filter lookup to complete the fast retrieval of identity information [34]. It is worth noting here that in the CS, the same content group name does not represent the name of the encoded block under the same content. Therefore, different encoded blocks are recognized in the CS based on the content name and encoding vector.

Here, let us give a brief explanation of the example in the previous section with the topology in Figure 1 with our newly designed interest and data message format. The processing of **Scenario 1** is not complicated. Users do not need to know the encoding and grouping status of the content provider, nor do they need to understand how to obtain the required data from the data message. They only need to submit a content request with their identity information to the edge nodes, and all the required computing will be offloaded to the edge nodes. The edge nodes will aggregate the demands for the same content from users connected to itself, set the corresponding parameters of the interest message in Table 2, and save the path that the interest travels through into the PIT table until the data are returned to the awaiting user.

For Scenario 2, we have the following.

- Case 1: If interest I_2 arrives at node B before node B responds to interest I_1, it can be found that node F already has a request for the same content encoded block in node B by comparing the identity hash value stored in the PIT entry of the same encoded group in PIT_B with the identity hash value carried in I_2 (as shown in Table 5). I_2 will be added to the corresponding PIT entry.
- Case 2: When interest I_2 arrives at node B, node B has already finished responding to interest I_1 and deleted the corresponding PIT entry. I_2 first visits the content storage CS_B of node B, and searches for the identity information of node F left by I_1 in the corresponding content Bloom filter of CS_B. Then, it continues to perform FIB forwarding. If other requests exist for the same content, I_2's request interface and node F's identity information will be added to PIT_B. Otherwise, the PIT entry will be recreated.

- In both Case 1 and Case 2, I_2 will be forwarded toward the content provider by looking up the forwarding information base (FIB_F) of node F to request other linearly independent encoded blocks.

Table 5. Pending interest table with identity hash.

Content Name	Face	Identity and Provider Information
/pku.edu.cn/web/promo.mp4/1/1	1	0cc175b9c0f1b6a831c399e269772661, 65
		0cc175b9c0f1b6a831c399e269772661, 41
	2	92eb5ffee6ae2fec3ad71c777531578f, 18
		0cc175b9c0f1b6a831c399e269772661, 32
/pku.edu.cn/web/promo.mp4/1/2	...	

Similarly, node G can obtain three linearly independent encoded blocks through the paths $G \rightarrow C \rightarrow A$, $G \rightarrow E \rightarrow D \rightarrow C \rightarrow A$, and $G \rightarrow E \rightarrow D \rightarrow B \rightarrow A$.

3.3. Processing Sequence of Request Interest Message in Intermediate Nodes

After the edge node receives the content request, it will send an interest message that follows the format of name field *ContentName* and command parameter field with $REQGENNUM$ to the content provider to request the number of encoded groups for specific content, as shown in Figure 2. It can also query the number of segmentation blocks K and encoded blocks $N(K \leq N)$ for a particular encoded group through an interest message with the format of name field $< ContentName/GenerationID >$ and parameter field of $REQSLICENUM$. The content provider and the intermediate nodes encountered along the way that can satisfy these two interest message requests in their content stores will respond to them.

After the edge node, $Node_e$, obtains this information, it can fill in the content required by the interest message in Table 2 and issue a request INT_{req} for the content. After the intermediate node, $Node_m$, receives the interest INT_{req} requesting the encoded block of a particular encoded group G_i, it first checks its CS. It obtains an exact match with the content name within the interest message. If it is found that there are already M_i content encoded blocks with the same content distribution number stored in the CS, and $M_i \geq K_i$, it indicates that these M_i encoded blocks can meet $Node_e$'s request for content grouping G_i. IDEANE can ensure that only linearly independent encoded vectors and their corresponding blocks are cached in the CS. Due to the fact that IDEANE does not re-encode and decode encoded groups at intermediate nodes in order to avoid impacting the security characteristics of multi-identifier networks, it is still necessary for $Node_e$ to calculate the maximum linearly independent encoded vector group for these encoded groups and pull the corresponding encoded blocks one by one. $Node_m$ retrieves any encoded block in response to INT_{req}. $Node_e$ can request the remaining linearly independent encoded blocks sufficient to satisfy its decoding from $Node_m$.

If $M_i < K_i$, the system examines whether the identity information of $Node_e$ exists in the Bloom filter BF_m of the CS entry. If it does not exist, it will directly respond to INT_{req} with the encoded block and encoded vector in the table entry, update *ReqNo* as part of the data message, and leave the identity information of $Node_e$ in BF_m. If the identity information of $Node_e$ already exists, it indicates that the $Node_m$ has responded to the encoded block segment of the $Node_e$. Then, the interest message INT_{req} can be used to continue applying for other linearly independent, new encoded blocks belonging to the encoded group.

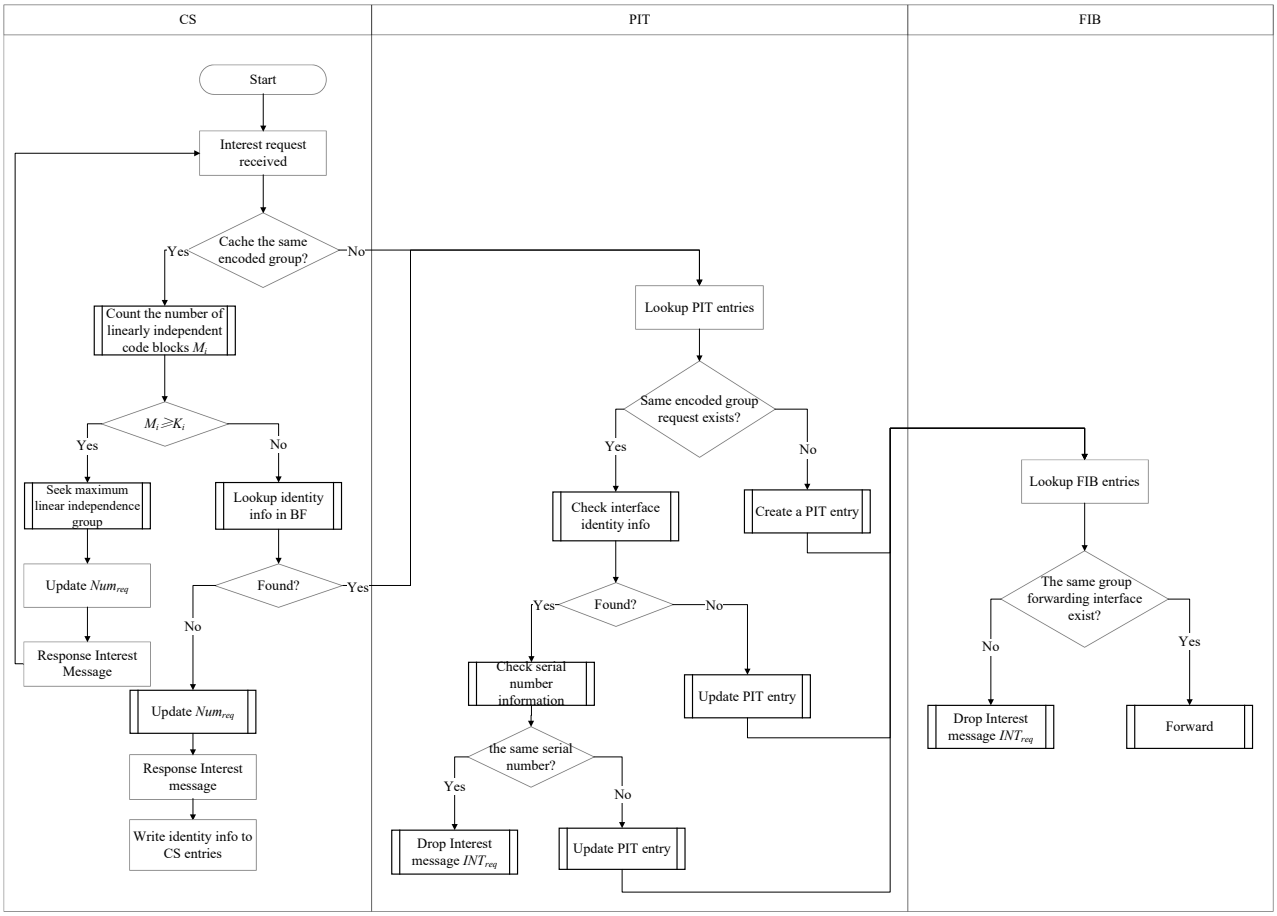


Figure 2. Identity-differentiated-based network coding interest processing sequences.

Next, the PIT is checked with an exact match to determine whether a PIT entry requests the same content group. If it does not exist, the system creates the corresponding table entry in PIT, writes the content group name, adds the hash of the routing interface, and identities the information, together with the request sequence number ReqNo, and forwards this interest message through the FIB. If present, it checks the interface information in the PIT table entry to inspect whether INTreq carries the same identity information. If found, it indicates that $Node_e$ has previously issued a request for the encoded group through the same or other interfaces. Next, the system checks the parameter information carried in the interest message. If the same sequence number exists in the PIT table entry, INT_{req} is regarded as a duplicate interest message and discarded. Otherwise, regardless of which interface INT_{req} comes from, the request interface and sequence number information for the content encoded group in the PIT will be updated, and INT_{req} will be forwarded through FIB.

The processing flow of FIB is the same as that of ordinary NDN design. The FIB table is looked up with the longest prefix match. If there is a forwarding interface for the same encoded group, INTreq is forwarded to the next-hop routing node through this interface based on the selected forwarding strategy. Otherwise, this interest request is discarded.

3.4. Processing Sequence of Response Data Message in Intermediate Nodes

As shown in Figure 3, when the intermediate node $Node_m$ receives a data message response DAT_{ack}, it will first perform PIT matching based on the group name in the data message. If no interface is found through which an interest message requests the group, the intermediate node will discard the data message. Moreover, suppose that the coding vector carried by the data message is a linear combination of the coding vectors of the same coding group in the CS. In this case, the data message will be deleted, which can ensure

that only linearly independent coding blocks are stored in the content store (CS). Otherwise, the content and encoding vector within the data message will be cached in the CS table. Then, the identity information of the originator of the interest message carried by the data message will be mapped to the Bloom filter of the content grouping entry.

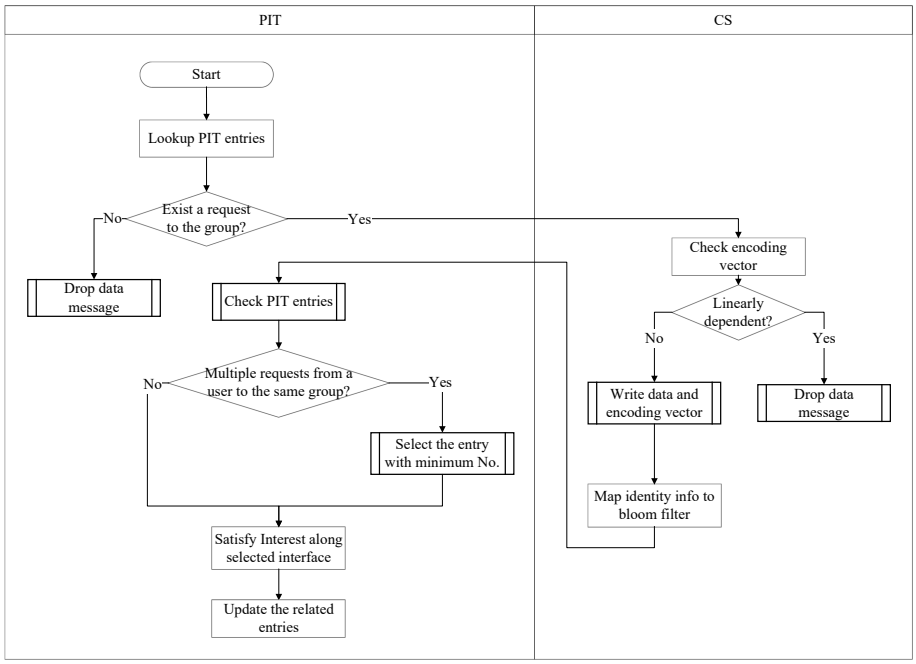


Figure 3. Identity-differentiated-based network coding data processing sequences.

The intermediate node will check the hash value of the content requester identity information stored in the PIT. If the same edge node $Node_e$ requests the encoded block of the content group through different interfaces, then only one interface will be used to respond to $Node_e$'s interest message requests for this group; in other words, the request sequence number is the only criterion to respond to the interest message requests, thus preventing $User_{req}$ from obtaining the same encoded block. $Node_m$ chooses the entry with the smallest user request number stored in the PIT to respond, fills the request number in the *ReqNo* field of the data response message, and responds to this interest message through the corresponding incoming interface stored in the PIT. $Node_e$ continues to wait for the responses of other interest requests for the same encoded group from upstream nodes, simultaneously deleting the corresponding PIT table entries.

4. Collaboration of Multi-Edge Nodes for Content Requester's Mobility

In IDEANE, the identity information is carried in the interest request message to realize identification and differentiation, which solves the problem of redundant coding blocks and idle transmission paths in multipath forwarding using network coding. At the same time, in order to reduce the pressure on the device terminal, the decoding work of the obtained coding block is offloaded to the connected edge nodes. Edge devices in edge computing refer to edge servers and routing nodes with computing, storage, and network resources, sufficient to perform codec operations for network coding. These edge nodes are closely connected with Industrial Internet devices. Section 3 has already discussed the scenario wherein the content requester is stationary. Through IDEANE, different edge nodes may obtain multiple copies of the same encoded blocks or duplicate copies to meet the needs of their connected requesters. This section elaborates on how to acquire enough coding blocks through multiple adjacent edge nodes when the content requester is on the move, such as in the C-V2X scenario of the Internet of Vehicles.

In a scenario wherein the content requester moves from one access point to another, as shown in Figure 4, the content requester CR_1 needs to communicate with the connected

edge node $EdgeNode_1$ through a heartbeat packet to inform it of the connection status. When $Node_e$ receives the new, linearly independent coding vector $Vector_e$ and coding block $Block_e$, to prevent the movement of CR_1, it will extract and send $Vector_e$ and $Block_e$ to CP_1 to the cache. Suppose that $EdgeNode_1$ collects enough encoding vectors to decode the packet G_i. In this case, it will perform a decoding operation, send the original content corresponding to the decoded G_i to CR_1, and delete the encoded blocks and vectors stored in CR_1 and $EdgeNode_1$.

IDEANE has set a heartbeat threshold T_{thr}. After a period T_{thr}, if $EdgeNode_1$ has not received the heartbeat information from CR_1, it will assume that CR_1 has lost its connection. It will continue to receive the subsequent data packets returned for the interest messages requesting the same content blocks and perform encoding and decoding. The encoding vectors will remain until all the previously sent interests are returned. The received encoded blocks and encoded vector or decoded original blocks stored in $EdgeNode_1$ will be reserved for another content requester, CR_2, which is another consideration when requesting content through the identity information of the edge node rather than that of the content requester.

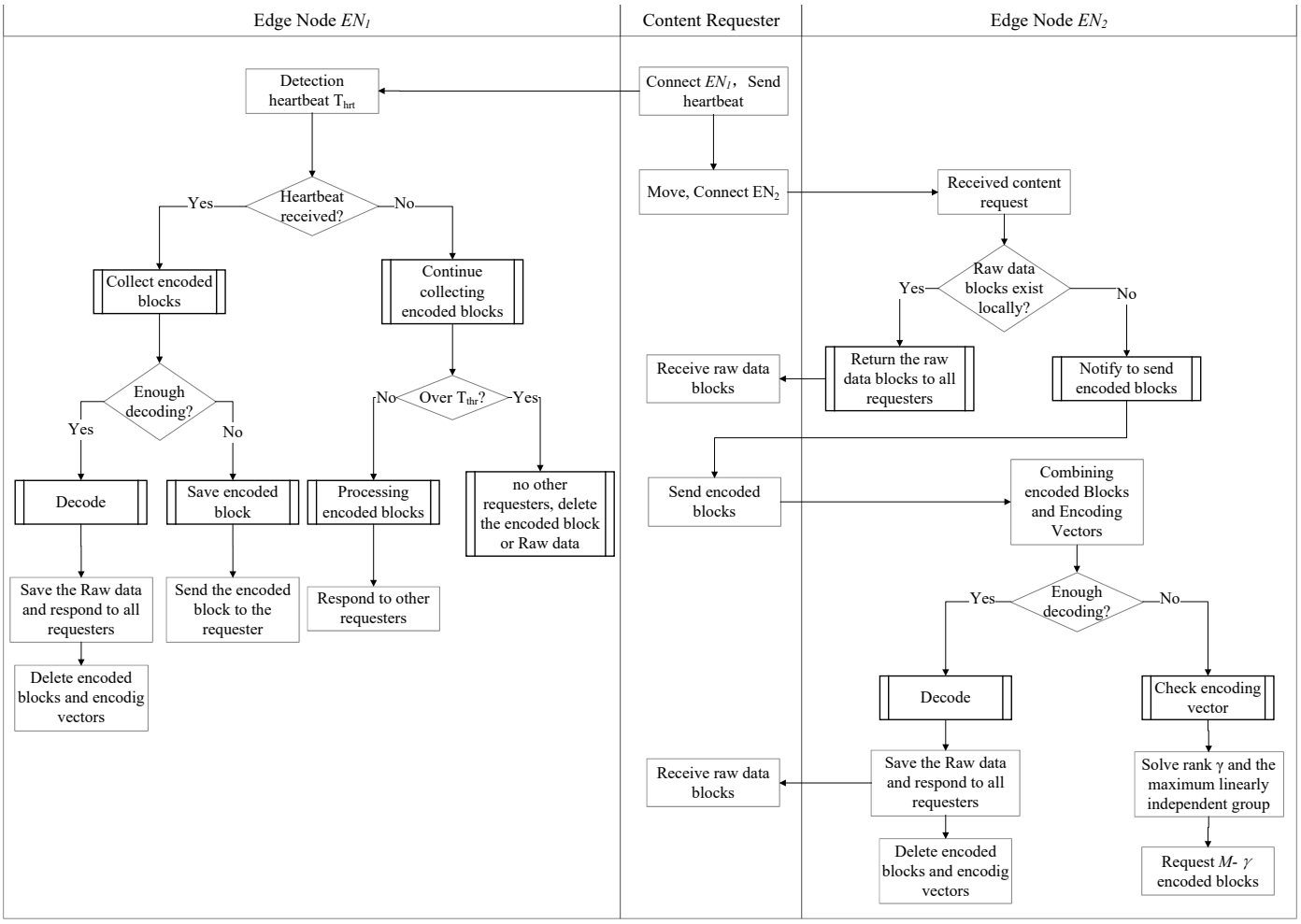


Figure 4. Collaboration of multiple edge nodes for content requester's mobility.

When CR_1 moves and connects to another edge server, $EdgeNode_2$, it will first examine whether $EdgeNode_2$ has the original data packet after decoding the encoded packets. If it exists, it will directly return it to CR_1. Otherwise, CR_1 will pass the encoded block and encoding vector parameters collected by $EdgeNode_1$ to the new edge server $EdgeNode_2$, to realize the collaboration between the two edge nodes, $EdgeNode_1$ and $EdgeNode_2$. $EdgeNode_2$ will examine the encoded data and the encoded data information of the content packet stored by itself. If it is sufficient to decode the original data blocks, then it directly decodes and

returns them to all content requesters connected, avoiding the requesting of duplicate data blocks and redundant network transmission. Otherwise, $EdgeNode_2$ arranges the coding vectors in columns in a matrix A and performs an elementary row transformation operation until the matrix A is simplified to a row echelon shape. The rank γ of the coding vectors, the maximum linearly independent group of the coding vectors, and its corresponding γ coding blocks are found. Next, the remaining $K - \gamma$ linearly independent coding blocks are requested from the network.

5. Performance Evaluation

5.1. Experimental Environment

We simulated the proposed scheme with ns-3 based on ndnSIM 2.7 [35] for experimentation. All experiments were performed on a commodity server installed with four 12-core CPUs (Intel(R) Xeon(R) Gold 5118) and 32 GB RAM. The operating system was Ubuntu 16.04 with kernel 4.15. In order to evaluate the performance of multipath forwarding, we drew the topology as shown in Figure 5 with ndnSIM. We configured the locations of each node in the topology and the connection relationships between different nodes. One content provider, two edge nodes, and six intermediate nodes were simulated in the experiment. In the initial design of a multi-identifier network, two edge nodes need to understand the topology and cooperate by transmitting the same data blocks on the intermediate link to achieve optimal performance. Bottleneck links do not affect encoded block transmission after network coding.

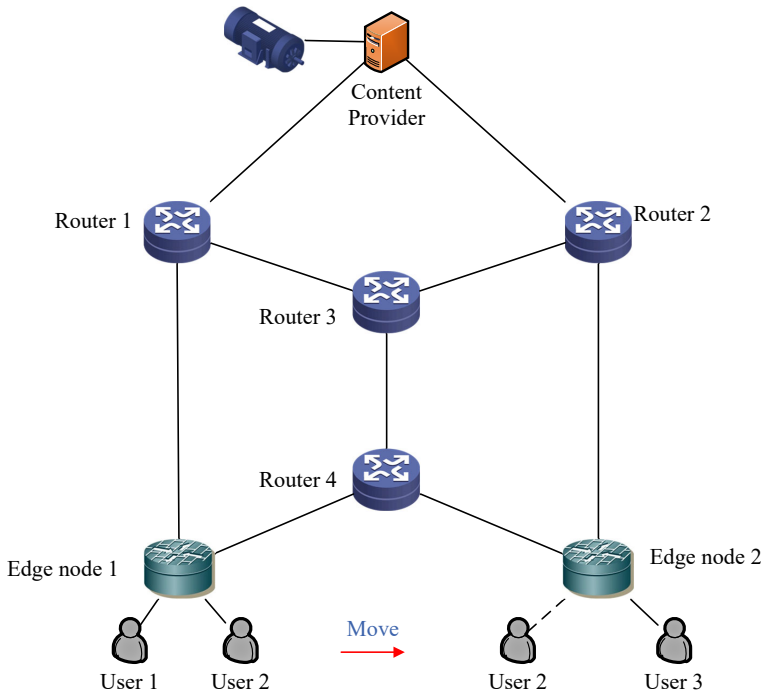


Figure 5. Simulation topology diagram.

In order to obtain an intuitive comparison with NetCodCCN, we adopted parameter settings similar to theirs, as shown in Table 6. Content is divided into 100 different segments, each with a size of 5 K bytes, which is sufficient to meet the requirement of multi-identifier routers with packet sizes below 8 K and avoid splitting the data segment again. All encoding and decoding operations are in the finite field F_{2^8}. At the same time, the content requester requests the required content at an average speed, and the content request satisfies the Zipf distribution. Next, we configure the simulated routing path and conduct performance evaluations. We compared IDEANE with the best path forwarding (NDN-BestRoute) adopted in the original NDN forwarding scheme [36], the flood multicast forwarding

scheme (NDN-Multicast), and the multipath forwarding method NetCodCCN [13] with the network coding strategy.

Table 6. Emulation parameters.

Parameter	Value Range	Default Value
Connection bandwidth	2–10 Mbps	10 Mbps
Transmission delay	1 ms	1 ms
Number of content segments	100	100
Number of encoded blocks per segment	100	100
Segment size	5 KB	5 KB
Finite field	F_{2^8}	F_{2^8}
Zipf	0.6–1.6	0.8
Cache size (proportion of total content)	0.05–0.25	0.1
Pipeline number (pcs/s)	1–25	10

5.2. Performance Evaluation

5.2.1. Pipeline Influence on Forwarding Performance

Figure 6 shows the influence of continuous interest messages sent by edge nodes through pipelines on the network download performance. In order to prevent DDoS attacks, all concurrent interest messages are aggregated as one piece of interest in the PIT of the intermediate router within the retransmission suppression time range set by the exponential backoff algorithm in MIN-IIoT, because they carry the same content name. Only one request message will be forwarded, and the others will be discarded.

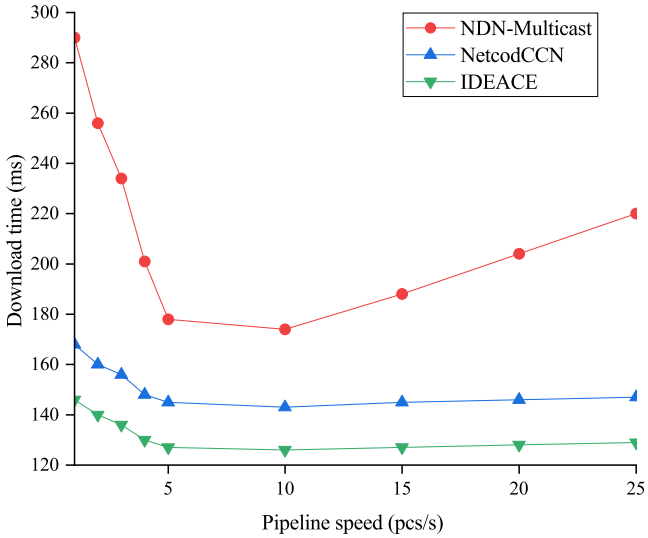


Figure 6. Influence of continuous interest requests on content download time.

The NDN-BestRoute scheme does not support concurrent requests from different users. It only requests original data packets along the path with the lowest forwarding cost, without distinguishing interest requests from different downstream nodes. Therefore, we would not choose NDN-BestRoute as a reference for comparison. With the increase in concurrency, the transmission delay of NDN-Multicast decreases quickly (from 290 ms to 174 ms), and then increases slowly (from 174 ms to 219 ms). With the gradual increase in concurrency, the number of previous request packets in the network will gradually increase, thus accelerating the content requests. However, when the concurrency exceeds 10, concurrent requests from the same requester and parallel ones from different requesters will be challenging to coordinate on the bottleneck link (router 3 to router 2), and congestion will occur. Generally speaking, slow interest requests can cause the network to become

idle, while frequent interest requests can lead to congestion. It is necessary to wait for the network to respond and deal with the interest sent before.

In contrast, the access latency of the NetCodCCN and IDEANE schemes using network coding will not be significantly affected by the number of concurrent requests. Both of them request encoded blocks, which can eliminate the possibility of asking for the same content block on the bottleneck link. When the amount of concurrency is small, these two methods will take time to process the received encoded blocks, so the initial transmission delay is worse than that of NDN-Multicast. At the same time, they show a slight upward trend after a significant decline. The average transmission delay of IDEANE is 16.5% less than that of NetCodCCN. IDEANE only performs linear independent verification of the encoded vectors at the intermediate node, without secondary encoding and decoding. At the same time, NetCodCCN does not deal with the linear correlation of the encoded blocks and requires secondary coding at the intermediate nodes. Therefore, there will be linearly dependent encoded blocks, and it is necessary to reapply for linearly independent ones. Therefore, IDEANE can reduce the delay in network requests and achieve better results than NetCodCCN. In the next experiment, we choose the concurrent interest message with the best NDN-Multicast performance, and the number of pipeline requests is 10 per second.

5.2.2. Cache Size Influence on Transmission Performance

The influence of the cache size on the cache hit rate is shown in Figure 7a. The hit rate is the ratio of the number of interest messages in the cache service to the total number of interests in the response. The hit distance represents the number of route hops to obtain the required cache. With the increase in cache size, the cache space can accommodate more cache content. The cache hit rates of these four forwarding schemes are all improved, and the trend of improvement is declining; meanwhile, the cache hit distance is reduced.

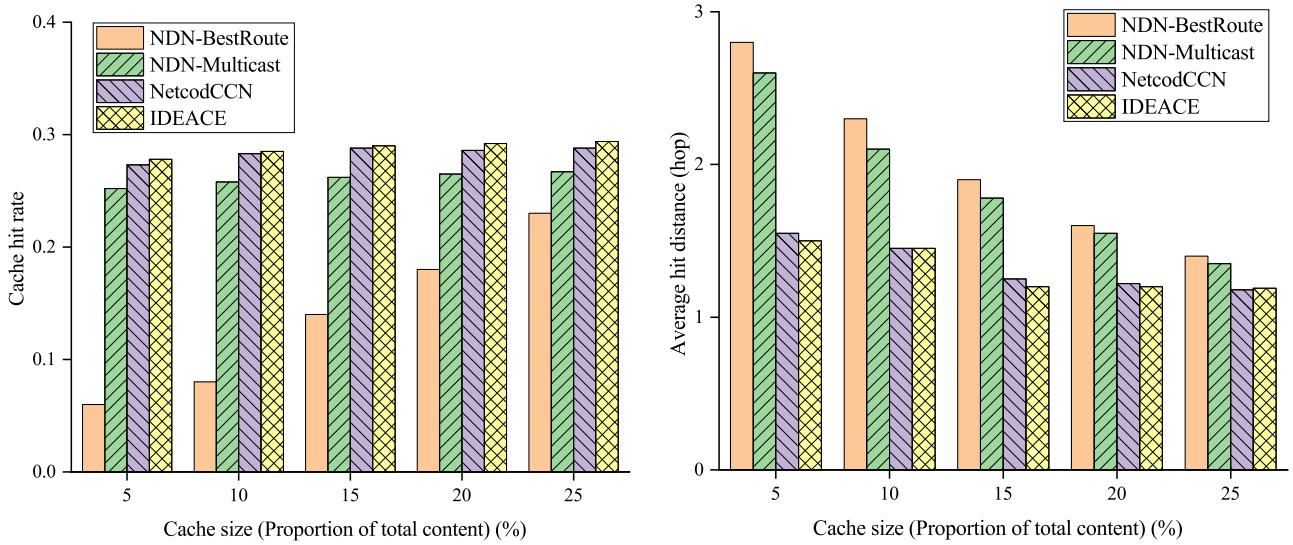


Figure 7. Caching performance under different cache sizes. (**a**) Average cache hit rate. (**b**) Average cache hit distance.

Benefiting from the multipath parallel forwarding capability brought by network coding, NetCodCCN and IDEANE can achieve a slightly better cache hit rate than NDN-Multicast, and a significantly better rate than NDN-BestRoute. NDN-BestRoute optimal path forwarding will result in redundant data transmission, so it is necessary to send interest messages to request new data blocks frequently. Network coding can prevent this situation and improve the cache hit rate. In the network topology shown in Figure 5, the original two NDN schemes need an average of 1.9 hops. In contrast, NetCodCCN and IDEANE can obtain the required cache at an average of 1.4 hops, thus reducing the hit

distance by approximately 23%, as shown in Figure 7b. The network coding scheme can reach the required cache with fewer hops, because network coding can reduce the requests for the same encoded blocks. Therefore, it can obtain different encoded blocks within the same encoded group, thereby decreasing the hit distance of the cache.

Figure 8a shows the impact of the cache size on the content download time and interest message load. The larger the cache, the more data blocks can be accommodated by intermediate nodes and the higher the probability of cache hits. Therefore, the download time of all forwarding strategies will be reduced. The three strategies using multipath forwarding can all achieve a shorter download time within 31.6% compared to NDN-BestRoute to achieve the desired content. IDEANE does not need to encode again at the intermediate nodes. It only makes use of the encoding vectors carried in the received encoded blocks to perform linear independent verification on the encoded blocks of the same encoded group. Thus, it can save 35.4% of download time compared to NetCodCCN.

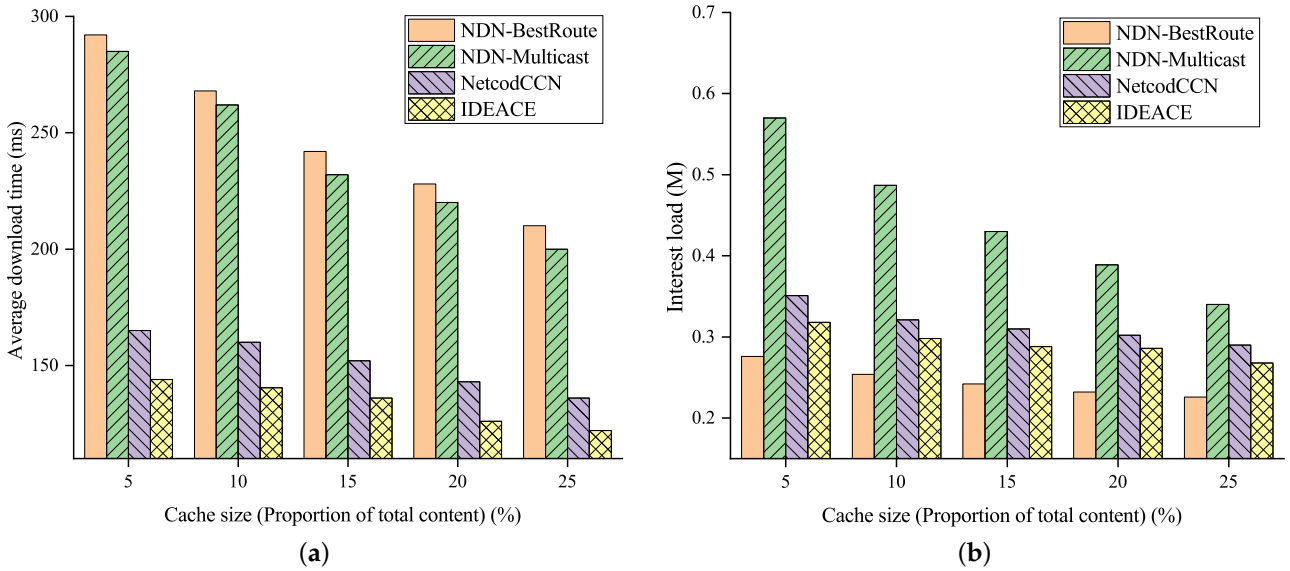


Figure 8. Transmission performance under different cache sizes. (**a**) Average cache hit rate. (**b**) Average cache hit distance.

As shown in Figure 8b, NDN-BestRoute consumes a smaller interest message load to obtain content than other strategies because the optimal path selected through the NDN-BestRoute strategy is more likely to cache hits. At the same time, the interest load consumed by each node in NetCodCCN and IDEANE will decrease with the increase in cache. With the increase in cache, many encoded blocks are cached in the CS of intermediate nodes, thus reducing the bandwidth consumption required to obtain content. NetCodCCN may need to reapply for encoded blocks because it has brought back linearly correlated ones, leading to interest load consumption that is almost 11.4% higher than that of IDEANE.

5.2.3. Bottleneck Link Bandwidth Influence

Figure 9 shows the influence of the bottleneck link network bandwidth on the content acquisition time. When other parameters remain the same, it demonstrates the influence of the bottleneck bandwidth on the transmission delay. It can be seen that NDN-BestRoute is most affected by the bandwidth. NDN-BestRoute does not choose the path in the middle of the butterfly network (router 3 to router 2) for requests from two edge nodes. The exact path will return all interest message requests to the edge nodes. The optimal path will be congested if the network bandwidth is too small. With the increase in bandwidth, this congestion situation will be resolved. When NDN-Multicast is adopted for multipath forwarding, each intermediate node will forward each interest message through two paths.

The content provider needs to respond to the content request separately on each forwarding path, leading to request latency and transmission redundancy.

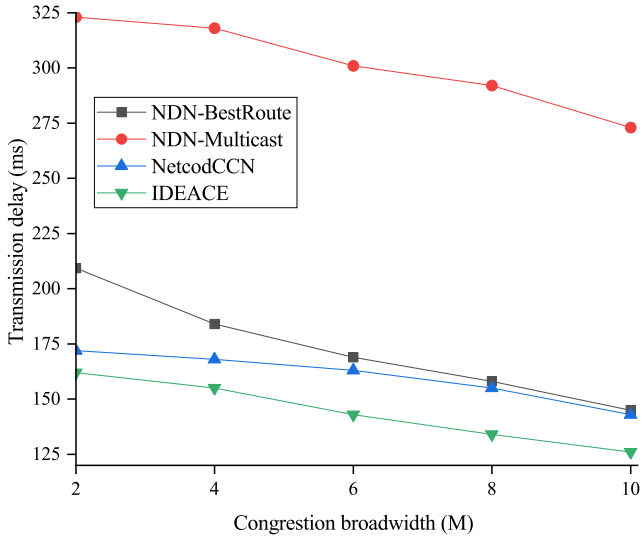


Figure 9. Influence of bottleneck network bandwidth on transmission delay.

As shown in Figure 9, IDEANE achieved an average reduction in the download time of nearly 42% compared to NDN-Multicast and approximately 8.3% compared to NetCodCCN. IDEANE does not need to perform encoding and decoding operations on encoded blocks in the intermediate nodes. Instead, it directly forwards these packets.

5.2.4. Influence of Data Popularity on Transmission Performance

The influence of data popularity Zipf(α) on transmission performance is shown in Figure 10. The skewness of data requests is in the range of 0.6 to 1.2 following Zipf's distribution. With the increase in α, more popular content will be cached in more intermediate network nodes; the cache hit rate will also increase, while the download time will be reduced. The two network coding schemes achieve a higher cache hit rate than others. Both of them can support content requests from multiple content providers and aggregate the encoded blocks of the same content to reduce the amount of data transmitted by the network. Compared with NDN-Multicast, they can reduce the download time by approximately 41.6%. At the same time, IDEANE's value is roughly 10.8% lower than that of NetCodCCN.

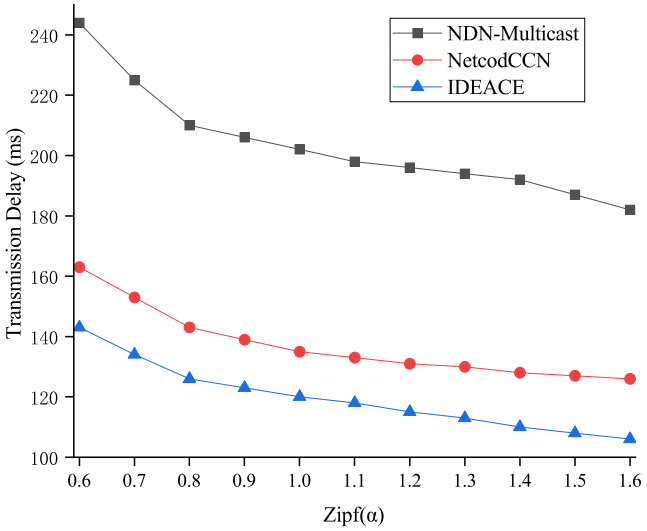


Figure 10. Influence of data popularity on transmission delay.

6. Performance Analysis and Discussion

The experiments have shown that IDENE outperforms existing multipath forwarding schemes such as NetCodCCN and BestRoute in terms of pipeline transmission, content acquisition time, transmission delay, and other metrics. In this section, we further analyze and discuss the evaluation results.

Unlike the conventional NDN forwarding procedure, IDEANE can prevent the aggregation of interest messages in PIT from different requesters who demand content with the same name by distinguishing distinct identities. Therefore, IDEANE can improve the interest efficiency, reduce encoded block redundancy, and fully use idle transmission paths. At the same time, the content can be returned to the corresponding requesters along the request paths of the interest messages differentiated with the carried identity information.

Both network coding and multi-identifier networks aim to improve the network transmission efficiency. The former focuses on the computing bandwidth, while the latter focuses on the storage bandwidth. All network nodes can encode the received data packets through linear network coding instead of copying and forwarding them as in traditional networks. When receiving a series of decoded packets, i.e., linear independence encoded packets with the same number of source data segments, the receiver decodes them to obtain the required data. Using encoding in named data networks and multi-identifier networks can enable internal network nodes to simultaneously compute, store, and forward data packets, effectively utilizing multiple paths in the network, simplifying the design of interest message sending and transmission on the receiving pipeline, and reducing the complexity of cache coordination [37]. Network coding can simultaneously transmit multiple information streams on one link, which improves link utilization. The network throughput can be improved whether on a multicast or unicast system, with uniform links or non-uniform links. In addition, network coding can effectively utilize idle paths, distribute network traffic to a broader network, and balance the network load, which helps to solve problems such as network congestion [38].

Linear network coding and edge computing play complementary roles. The edges in edge computing refer to computing and storage resources on the network's edge, which can be edge servers, router gateways, etc. The problem of the limited resources of Industrial Internet objects can be solved by encoding and decoding at edge nodes. In edge computing, the original data and sensitive user privacy information will be shared or stored in edge nodes. Hence, security assurance becomes a crucial challenge in this distributed edge network framework. Since the edge node receives the encoded block instead of the original data packet, data leakage can be avoided to a certain extent, and the security of data transmission and the information security requirements of the data of both parties participating in the computation can be guaranteed. Network coding has certain reliability in the application of wireless network security. A random network coding detection strategy for data packets is generated based on network coding. This method has a small amount of computing, and the detection probability can be adjusted according to factors such as the communication control overhead, network coding complexity, and detection time At the same time, the multi-identifier network (MIN-IIoT) can establish user identity certificates and perform lightweight identity verification through the carried identity identifier and location identifier information, thereby ensuring that user identities are authorized.

During network transmission, there may be abnormal interruptions, and the BestRoute strategy requires recalculation of the optimal path, resulting in network latency. Our experimental results show that NetCodCCN and IDEANE, which utilize network coding technology, can process the information on each node in the network data transmission process, and then transmit it to downstream nodes. The intermediate nodes encode or process signals, and then deduce and restore the network data at the final node. Network coding reduces transmission duplication, dramatically improves the transmission efficiency, and enhances network robustness. IDEANE can obtain more required content in the event of partial link failure by distinguishing interest packet requests by identity identifiers.

Network encoding consumes a small amount of computing power when encoding and decoding content chunks, thus increasing the network latency accordingly. In order to improve the performance of network coding, we reduce the computational complexity by splitting content into many *generations*. In order to reduce the pressure on resource-constrained Industrial Internet objects, we only perform encoding at the nodes adjacent to the content provider and decode the encoding block at the edge nodes in close proximity to the content requesters. In order to reduce the network latency, intermediate nodes do not perform encoding and decoding operations. In our edge computing system integrated with linear network coding, the end user and the edge server are single-hop network connections, and the latency is directly related to the wireless transmission technology used. Especially in a 5G network environment, the delay in wireless communication can be reduced to one millisecond, which makes edge computing virtually indistinguishable from local computing in terms of delay.

Network coding does not compress data; instead, it restores the original data by collecting sufficient encoded blocks for linear combination, which can facilitate the collaboration of multiple edge nodes. Two different edge nodes allow for redundant encoded packets. At the same time, if the original data are divided into K data blocks and encoded into M coded blocks ($M \geq K$), $M - K$ transmission errors can be allowed in network transmission. Such fault tolerance can alleviate the impact of data loss in network transmission during the movement of content requesters. On the other hand, during the movement process of the content requester, there may already be linearly independent coding blocks of the same content that have been cached in the different edge nodes that it is connected to, thereby reducing repeated content requests.

7. Conclusions

This paper proposes IDEANE, a multipath forwarding strategy that makes use of identity identifiers to distinguish content requests with the combination of network coding and edge computing. IDEANE differentiates content requests through identity information and request sequence numbers, which can reduce the aggregation of multiple interest messages, improve the efficiency of interest message transmission, and promote the multipath forwarding of content requests. Encoding and decoding at the edge nodes can reduce the computing pressure of linear network coding on the intermediate nodes and reduce the transmission latency. Simultaneously, linear network coding and identity authentication can also improve the security of edge computing. At the same time, this paper proposes a multiple edge node cooperation scheme to solve the mobile problem of content requesters based on IDEANE and reduce the time taken for users to obtain the required content during the movement process. The simulation results on ndnSIM show that IDEANE can improve the cache hit rate, reduce the redundancy of coding blocks, shorten the cache hit distance, and reduce the request waiting time. In addition, IDEANE can solve the problem of link congestion caused by queuing.

In future work, we will investigate how to intelligently schedule content request tasks to all edge nodes, taking into account the network bandwidth and remaining cache resources of edge nodes to complete load balancing and realize multi-access multipath forwarding and resource acquisition. In addition, taking advantage of network coding, coded computing [39], and edge computing [12], we will investigate how to better utilize identifiers for network forwarding in 6G networks to solve the communication bottleneck and improve the multipath forwarding performance and network transmission efficiency.

Author Contributions: Conceptualization, H.L. and Y.W.; methodology, Y.W.; software, Y.W.; validation, Y.W.; formal analysis, H.L.; investigation, Y.W.; resources, H.L.; data curation, Y.W.; writing—original draft preparation, Y.W.; writing—review and editing, H.L.; visualization, Y.W.; supervision, H.L.; project administration, H.L.; funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the 2022 High-Level Research Startup Fund of Liuzhou Vocational Technical College (Grant No. 2022GCQD07), Guangdong R&D Key Program (2019B010137001), Shenzhen Research Program (GXWD20201231165807007-20200807164903001, JCYJ20210324122013036, JCYJ20190808155607340), China Environment for Network Innovation (CENI) (GJFGW No.[2020]386, SZFGW [2019] 261), and National Keystone R&D Program of China (2017YFB0803204).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of Interest.

Abbreviations

The following abbreviations are used in this manuscript:

CS	Content Store
DAG	Directed Acyclic Graph
EI	Edge Intelligence
FIB	Forward Information Base
ICN	Information-Centric Networking
IDEANE	Identity Differentiated Content Acquisition with Network Coding and Edge Computing
MEC	Mobile Edge Computing
MIN	Multi-Identifier Network
MIN-IIoT	Multi-Identifier Network for Industrial Internet
NDN	Named Data Networking
PIT	Pending Interest Table
RLNC	Random Linear Network Coding

References

1. Malik, P.K.; Sharma, R.; Singh, R.; Gehlot, A.; Satapathy, S.C.; Alnumay, W.S.; Pelusi, D.; Ghosh, U.; Nayak, J. Industrial Internet of Things and its applications in industry 4.0: State of the art. *Comput. Commun.* **2021**, *166*, 125–139. [CrossRef]
2. Li, J.Q.; Yu, F.R.; Deng, G.; Luo, C.; Ming, Z.; Yan, Q. Industrial internet: A survey on the enabling technologies, applications, and challenges. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1504–1526. [CrossRef]
3. Qin, W.; Chen, S.; Peng, M. Recent advances in Industrial Internet: Insights and challenges. *Digit. Commun. Netw.* **2020**, *6*, 1–13. [CrossRef]
4. Shannigrahi, S.; Fan, C.; Partridge, C. What's in a Name? Naming Big Science Data in Named Data Networking. In Proceedings of the ACM ICN '20: 7th ACM Conference on Information-Centric Networking, Virtual Event Canada, 29 September–1 October 2020; pp. 12–23.
5. Wang, Y.; Li, H.; Huang, T.; Zhang, X.; Bai, Y. Scalable Identifier System for Industrial Internet Based on Multi-Identifier Network Architecture. *IEEE Internet Things J.* **2023**, *10*, 1919–1932. [CrossRef]
6. Zhang, L.; Afanasyev, A.; Burke, J.; Jacobson, V.; Claffy, K.; Crowley, P.; Papadopoulos, C.; Wang, L.; Zhang, B. Named data networking. *ACM SIGCOMM Comp. Commun. Rev.* **2014**, *44*, 66–73. [CrossRef]
7. Yi, C.; Afanasyev, A.; Wang, L.; Zhang, B.; Zhang, L. Adaptive forwarding in named data networking. *ACM SIGCOMM Comp. Commun. Rev.* **2012**, *42*, 62–67. [CrossRef]
8. Hoque, A.M.; Amin, S.O.; Alyyan, A.; Zhang, B.; Zhang, L.; Wang, L. NLSR: Named-data link state routing protocol. In Proceedings of the ACM ICN, Hong Kong, China, 12 August 2013; pp. 15–20.
9. Ahlswede, R.; Cai, N.; Li, S.Y.; Yeung, R.W. Network information flow. *IEEE Trans. Inf. Theory* **2000**, *46*, 1204–1216. [CrossRef]
10. Li, S.Y.; Yeung, R.W.; Cai, N. Linear network coding. *IEEE Trans. Inf. Theory* **2003**, *49*, 371–381. [CrossRef]
11. Lei, K.; Zhong, S.; Zhu, F.; Xu, K.; Zhang, H. An NDN IoT content distribution model with network coding enhanced forwarding strategy for 5G. *IEEE Trans. Ind. Inform.* **2017**, *14*, 2725–2735. [CrossRef]
12. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge computing: Vision and challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [CrossRef]
13. Saltarin, J.; Bourtsoulatze, E.; Thomos, N.; Braun, T. NetCodCCN: A network coding approach for content-centric networks. In Proceedings of the IEEE INFOCOM, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9.
14. Park, J.S.; Lun, D.S.; Soldo, F.; Gerla, M.; Médard, M. Performance of network coding in ad hoc networks. In Proceedings of the IEEE MILCOM, Washington, DC, USA, 23–25 October 2006; pp. 1–6.
15. Li, B.; Niu, D. Random network coding in peer-to-peer networks: From theory to practice. *Proc. IEEE* **2011**, *99*, 513–523.

16. Dimakis, A.G.; Godfrey, P.B.; Wu, Y.; Wainwright, M.J.; Ramchandran, K. Network coding for distributed storage systems. *IEEE Trans. Inf. Theory* **2010**, *56*, 4539–4551. [CrossRef]
17. Ho, T.; Leong, B.; Koetter, R.; Médard, M.; Effros, M.; Karger, D.R. Byzantine modification detection in multicast networks with random network coding. *IEEE Trans. Inf. Theory* **2008**, *54*, 2798–2803. [CrossRef]
18. Syed, A.A.; Ayaz, S.; Leinmüller, T.; Chandra, M. Network Coding Based Fault-Tolerant Dynamic Scheduling and Routing for In-Vehicle Networks. *J. Netw. Syst. Manag.* **2023**, *31*, 27. [CrossRef]
19. Montpetit, M.J.; Westphal, C.; Trossen, D. Network coding meets information-centric networking: An architectural case for information dispersion through native network coding. In Proceedings of the ACM NoM, Hilton Head, SC, USA, 11 June 2012; pp. 31–36.
20. Hu, X.; Zheng, S.; Zhang, G.; Zhao, L.; Cheng, G.; Gong, J.; Li, R. An on-demand off-path cache exploration based multipath forwarding strategy. *Comput. Netw.* **2020**, *166*, 107032. [CrossRef]
21. Malik, H.; Adjih, C.; Weidmann, C.; Kieffer, M. MICN: A network coding protocol for ICN with multiple distinct interests per generation. *Comput. Netw.* **2021**, *187*, 107816. [CrossRef]
22. Shi, W.; Pallis, G.; Xu, Z. Edge computing. *Proc. IEEE* **2019**, *107*, 1474–1481. [CrossRef]
23. Zhou, Z.; Chen, X.; Li, E.; Zeng, L.; Luo, K.; Zhang, J. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proc. IEEE* **2019**, *107*, 1738–1762. [CrossRef]
24. Cruz, P.; Achir, N.; Viana, A.C. On the edge of the deployment: A survey on multi-access edge computing. *ACM Comput. Surv.* **2022**, *55*, 1–34. [CrossRef]
25. Porambage, P.; Okwuibe, J.; Liyanage, M.; Ylianttila, M.; Taleb, T. Survey on multi-access edge computing for internet of things realization. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2961–2991. [CrossRef]
26. Bréhon-Grataloup, L.; Kacimi, R.; Beylot, A.L. Mobile edge computing for V2X architectures and applications: A survey. *Comput. Netw.* **2022**, *206*, 108797. [CrossRef]
27. Qiu, T.; Chi, J.; Zhou, X.; Ning, Z.; Atiquzzaman, M.; Wu, D.O. Edge computing in industrial internet of things: Architecture, advances and challenges. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2462–2488. [CrossRef]
28. Leng, J.; Wang, D.; Shen, W.; Li, X.; Liu, Q.; Chen, X. Digital twins-based smart manufacturing system design in Industry 4.0: A review. *J. Manuf. Syst.* **2021**, *60*, 119–137. [CrossRef]
29. Letaief, K.B.; Shi, Y.; Lu, J.; Lu, J. Edge artificial intelligence for 6G: Vision, enabling technologies, and applications. *IEEE J. Sel. Areas Commun.* **2021**, *40*, 5–36. [CrossRef]
30. Chen, B.; Wan, J.; Celesti, A.; Li, D.; Abbas, H.; Zhang, Q. Edge computing in IoT-based manufacturing. *IEEE Commun. Mag.* **2018**, *56*, 103–109. [CrossRef]
31. Zhou, X.; Yang, X.; Ma, J.; Kevin, I.; Wang, K. Energy-efficient smart routing based on link correlation mining for wireless edge computing in iot. *IEEE Internet Things J.* **2021**, *9*, 14988–14997. [CrossRef]
32. Phung, C.V.; Dizdarevic, J.; Carpio, F.; Jukan, A. Enhancing rest http with random linear network coding in dynamic edge computing environments. In Proceedings of the IEEE MIPRO, Opatija, Croatia, 20–24 May 2019; pp. 435–440.
33. He, W.; Su, Y.; Xu, X.; Luo, Z.; Huang, L.; Du, X. Cooperative content caching for mobile edge computing with network coding. *IEEE Access* **2019**, *7*, 67695–67707. [CrossRef]
34. Bloom, B.H. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM* **1970**, *13*, 422–426. [CrossRef]
35. Mastorakis, S.; Afanasyev, A.; Zhang, L. On the evolution of ndnSIM: An open-source simulator for NDN experimentation. *ACM SIGCOMM Comp. Commun. Rev.* **2017**, *47*, 19–33. [CrossRef]
36. Team, N. Nfd Developer's Guide. Technical Report. Available online: https://named-data.net/publications/techreports/ndn-0021-11-nfd-guide/ (accessed on 18 May 2023).
37. Zhu, F.; Zhang, C.; Zheng, Z.; Farouk, A. Practical network coding technologies and softwarization in wireless networks. *IEEE Internet Things J.* **2021**, *8*, 5211–5218. [CrossRef]
38. Bassoli, R.; Marques, H.; Rodriguez, J.; Shum, K.W.; Tafazolli, R. Network coding theory: A survey. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 1950–1978. [CrossRef]
39. Kim, K.T.; Joe-Wong, C.; Chiang, M. Coded edge computing. In Proceedings of the IEEE INFOCOM, Toronto, ON, Canada, 6–9 July 2020; pp. 237–246.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.