ITC 3/52
Information Technology and Control
Vol. 52 / No. 3 / 2023
pp. 638-652
DOI 10.5755/j01.itc.52.3.33092

Network Coding for Efficient File Transfer in Narrowband Environments

Received 2022/12/28 | Accepted after revision 2023/08/02

HOW TO CITE: Yin, K., Huang, H., Liang, W., Xiao, H., Wang, L. (2023). Network Coding for Efficient File Transfer in Narrowband Environments. *Information Technology and Control*, 52(3), 638-652. https://doi.org/10.5755/j01.itc.52.3.33092

Network Coding for Efficient File Transfer in Narrowband Environments

Kangyong Yin, Haosheng Huang, Wei Liang
State Grid Jiangsu Electric Power Co., LTD. Electric Power Research Institute, Nanjing 211102, P. R.China

Hongwu Xiao, Lei Wang
School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, P. R.China; e-mail: leiwang@njupt.edu.cn

Corresponding author: leiwang@njupt.edu.cn

Achieving efficient end-to-end file transfer is challenging in a narrow-band communication environment with high latency and high packet loss rate. The traditional TCP-based scheme and the UDP-based automatic retransmission scheme have defects in the transmission performance, which cannot meet the increasing user demands. This paper proposes a high-efficiency file transfer scheme based on random linear network coding and the Kalman filtering algorithm to implement an efficient end-to-end file transfer scheme in a narrow-band environment. The scheme predicts the link quality of file transmission through the Kalman filter algorithm and designs an adaptive coding strategy for file transfer through random linear network coding. Experimental results show that the proposed method outperforms traditional file transfer schemes.

KEYWORDS: narrow-band communication, file transfer, network coding, Kalman filter.

1. Introduction

In broadband networks, FTP (File Transfer Protocol) and HTTP (Hyper Text Transfer Protocol) are often used for file transfer [29]. Although these protocols at the application layer have different file transfer strategies [8, 9], the bottom layer uses TCP [6] to ensure file safety and transfer reliability. Research shows that the transmission performance of TCP is better than that of UDP under normal network conditions [1]. However, in the narrow-band environment with high delay and high packet loss rate, the congestion avoidance algorithm in TCP will reduce the sliding window size [10], and the frequent message acknowledgment will significantly reduce the data transmission efficiency. With the continuous development of communication technology, there are more and more file transfer requirements in various narrow-band wireless scenarios [4]. The file transfer protocol based on TCP has gradually been unable to meet the needs.

UDP-based file transfer protocols are rare in the industry. For security reasons, many companies restrict UDP traffic. However, the rise of the QUIC protocol [34, 19] in recent years has made it possible to open UDP ports. UDP is a connectionless transport layer protocol. Any packet loss in an unreliable transfer environment will cause the file transfer to fail. The method based on the traditional UDP retransmission mechanism can improve the reliability of file transmission, but frequent data retransmission will still bring much system overhead. Moreover, as the transmission delay increases, so does this overhead. Therefore, further improving the efficiency of file transmission is necessary for an unreliable narrow-band environment. Network coding [25, 35] and traffic prediction technology [14] provide new ideas for solving this problem.

The network coding technique was first proposed by Alshwede et al. [2]. The core idea is to perform additional linear coding operations on the data during network transmission so that one data packet can contain the information of multiple data packets. Their research showed that network coding technology could improve the data throughput in the data transmission process so that the data transmission rate can reach the multicast capacity, defined as the minimal value of the maximum flow from the source node to different destination nodes in a multicast network. Now network coding is gradually developed into linear network coding [20] and nonlinear network coding [16]. Because linear network coding has a good algebraic structure and coding schemes, it is widely used in practical research, and it gradually developed into Random Linear Network Coding (RLNC) [11] and Deterministic Linear Network Coding (DLNC) [12]. RLNC is more suitable for practical network applications because it does not require global knowledge of the network topology at runtime. Therefore, using RLNC in end-to-end file transfer in a narrow-band environment is feasible.

The purpose of traffic prediction technology is to allocate and utilize network resources reasonably [15, 26]. It has become a key technology for improving network speed and utilization effectively. Many researchers have recently proposed various prediction algorithms to improve network transmission performance and resource utilization under limited channel resources [27]. Prediction algorithm models can be divided into the equation-based model [13, 21], the time-smoothing model [23, 30, 31], and the location-smoothing model [28, 33]. During data communication in the narrow-band environment, the network packet loss rate seriously affects data transmission efficiency. A prediction algorithm based on the network packet loss rate will help effectively use the bandwidth resources. The traditional Kalman filter model is a state auto-regressive and time-smoothing model, which is suitable for the filtering operation of the Markov process. Moreover, the Kalman filter model has real-time processing characteristics and is easy for computer implementation. Over the past 30 years, the Kalman filter algorithm has been widely used in prediction models in various fields [5, 17], which implies the feasibility of prediction algorithms based on the Kalman filter method in our scheme.

The rest of the paper is organized as follows. In Section 2, we introduce some previous studies related to our research. In Section 3, we present the system model of the scheme. In Section 4, we evaluate the performance of the proposed scheme. In Section 5, we summarize our work.

2. Related Work

There are many related studies on data transmission in unreliable network environments. To address the reliability and energy consumption issues of end-to-end data transmission in wireless sensor networks, Le et al. [22] designed an energy-efficient and reliable protocol ERTP for wireless sensor networks. In the ERTP, to reduce the energy consumption of intermediate nodes, the reliability of the end-to-end data transmission is ensured by dynamically controlling the data transmission reliability of each hop. At the same time, the packet loss recovery between nodes is carried out through the implicit confirmation mechanism, and the reliability and energy consumption are balanced by dynamically controlling the maximum number of retransmissions of the nodes. The author proposed a retransmission timeout estimation algorithm in the protocol to determine the time to wait for an acknowledgment message, improving transmission efficiency. The energy-saving and reliable transmission protocol proposed by the authors can ensure the reliability of data transmission in wireless sensor networks and reduce the energy consumption of each related node. However, the guarantee of data reliability between nodes will also cause transmission efficiency problems. The proposed network coding scheme in this paper improves the reliability of data transmission through error correction coding, avoids unnecessary data retransmission, and thus better improves the data transmission performance.

Ding et al. [7] proposed an end-to-end service transmission guarantee method in narrow-band opportunistic networks. Their method utilizes a Delay Tolerant Network (DTN) and designs a routing algorithm to ensure that the shortest reliable path is found as much as possible. In the data transmission process from the source node to the destination node, the intermediate node will buffer the received data and then analyze the routing path according to the node behavior. This dynamic routing method guarantees the successful delivery rate of data transmission in the opportunistic network. However, because the intermediate nodes need to cache data in the actual network environment, it is necessary to ensure data integrity at the intermediate nodes. Otherwise, the performance of file transfer schemes will be affected. In our research scheme, for the problem of frequent disconnection of links in opportunistic networks, a negative feedback (NACK) mechanism is used to guarantee the final successful delivery rate.

Keller et al. [18] designed and implemented the SENSECODE protocol in the sensor network. Different sending nodes will eavesdrop on the data packets sent by other transmitting nodes and linearly combine them with their data packets. Such a data packet contains information from multiple data packets, and decoding the original data can be completed when the receiving node receives enough linearly independent data packets. Bilbao et al. [3] proposed a link layer transmission scheme based on network coding and a generational sliding window mechanism to ensure reliable data transmission in a narrow-band power environment. The author divided the original file into multiple generations, and each generation uses RLNC to perform linear operations on the data. The sliding window mechanism allows various generations of data to simultaneously be active on the link to solve the decoding blocking problem at the receiving node. Experiments show that the above network-coding-based transmission protocols can provide throughput gains and reduce the error rate of end-to-end data packets. In our scheme, link quality prediction is adopted, enabling us to use link resources more reasonably.

Wu et al. [32] proposed a redundant packet control scheme based on link packet loss rate. The link loss rate of the worst link will determine the final number of redundant packets. Before each data encoding operation, the node will adjust the worst packet loss rate according to the situation. This strategy based on dynamic adjustment of redundant data packets can ensure the reliability of data transmission to a certain extent. Zhang et al. [36] propose a data collection scheme based on unequal redundancy levels. High-redundancy coding is used for non-hotspot areas, and low-redundancy coding is used for hotspot areas to improve coding and transmission efficiency. However, the simple worst packet loss rate and area division cannot accurately reflect the complex network environment. In our scheme, a prediction algorithm that can better express the current optimal state of the network is used, which further improves the utilization of network resources.

Mothku et al. [24] proposed a transmission scheme that combines network coding with the Markov decision process (MDP). The method predicts the current network state through a Markov process. At the same time, DLNC is used to adjust the number of redundant data packets adaptively. In addition, the author adopted a link selection algorithm to select the adjacent route in the optimal state and control whether the current node uses network coding technology through this algorithm to reduce the time spent on node encoding and decoding. However, the DLNC scheme and link selection algorithm adopted in their scheme requires knowledge of global network topology. In our method, RLNC is applied, and it does not require global information, so it is more suitable for complex and changeable network environments than DLNC.

Our research contributions can be summarized as follows: Firstly, we design a file coding scheme based on RLNC in a narrow-band environment to achieve more reliable file transfer. Secondly, a set of packet loss prediction algorithms and coding coefficient calculation schemes based on the Kalman filter algorithm is designed to use current network bandwidth resources efficiently. Finally, we build a simulation platform to evaluate the performance of the proposed scheme through the Mininet simulation tool. The experimental results show that our scheme can achieve more efficient data transmission while ensuring the reliable communication of files.

3. System Design

3.1. Coding Strategy

Compared with the traditional TCP file transfer scheme, the transfer schemes based on RLNC need to encode the data at the source node and intermediate nodes. Due to file size limitations, we cannot encode the entire file simultaneously. Thus, we have to split the original file into multiple groups. At this stage, the size L of each group is affected by K predicted by the destination node, and the length L is $K \times 1024$. The last group of files often cannot be divided accurately, so it is necessary to make some padding with zero to achieve the consistency of the data packet format.

During the process of source node encoding, the source node slices the file data to obtain a sliced matrix $G(g_1 \quad g_2 \quad g_3 \quad \dots \quad g_K)^T$, where the slice length is 1024 bytes. Then, the source node generates an $N-by-K$ random coding matrix M based on the finite field *GF(256)*. The data file is encoded by Equation (1), and an encoded data matrix $C(c_1 \quad c_2 \quad c_3 \quad \dots \quad c_N)^T$ of N slices is obtained. M and C are combined into the transmit matrix $W=[M,C]$. In Equation (1), m_{ij} represents the value of the *i-th* row and the *j-th* column of the random coding matrix $M(1 \le i \le N, 1 \le j \le K)$.

$$C = M \times G = \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1K} \\ m_{21} & m_{22} & \dots & m_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ m_{N1} & m_{N1} & \dots & m_{NK} \end{bmatrix} \times \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_K \end{bmatrix}. \tag{1}$$

In the process of the intermediate node transmitting data packets, random linear coding processing is performed on the intermediate node to improve the reliability and throughput of the intermediate node. In this scheme, firstly, we set the maximum receiving time t for intermediate nodes. Secondly, we buffer the valid data packets sent by the previous hop nodes within the time t, and extract the transmit matrix W_t from the data packets. Finally, random linear coding is performed on the transmit matrix W_t to obtain a new transmit matrix W_t', and the coding equation at intermediate nodes is shown as Equation (2). Because the source node has completed the redundant coding of the data, the intermediate node uses a $(t+1)-by-t$ random coding matrix M' to perform random linear coding on the data packet. Specifically, it is shown in Equation (2).

$$W_t' = \begin{bmatrix} m_{11}' & m_{12}' & \dots & m_{1t}' \\ m_{21}' & m_{22}' & \dots & m_{2t}' \\ \vdots & \vdots & \ddots & \vdots \\ m_{(t+1)1}' & m_{(t+1)2}' & \dots & m_{(t+1)t}' \end{bmatrix} \times W_t. \tag{2}$$

The destination node must first receive enough linearly independent packets to decode the file. However, the research proves that the probability that the random coding matrix generated in the finite field *GF(256)* is linearly independent is 0.991. As the random coding matrix expands, the probability of linear matrix correlation also increases. Therefore, this paper adopts a slider mechanism to reduce the impact of linearly correlated packets on file decoding. When linearly correlated packets are received, the data window slides down. After using this strategy, the successful decoding rate V of the same set of files will be calculated by Equation (3), where v is the probability that the vectors in the random matrix are linearly independent.

$$V = 1-(1-v)^{N-K}. \tag{3}$$

Due to the algebraic nature of the matrix, the re-encoding operation at the intermediate nodes does not affect the decoding at the destination nodes. When a destination node receives enough linearly independent data packets, it can obtain the random coding matrix M and the encoded data matrix C from the data packets and complete the data decoding by Equation (4).

$$G = \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1K} \\ m_{21} & m_{22} & \dots & m_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ m_{K1} & m_{K2} & \dots & m_{KK} \end{bmatrix}^{-1} \times \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_K \end{bmatrix}. \tag{4}$$

3.2. Predictive Algorithm Design

In the file transfer scheme under the narrow-band environment, we use the error correction code in network coding to avoid frequent message responses due to packet loss. How to control the number of error correction codes and effectively use system bandwidth resources are key problems. We propose an adaptive coding coefficient prediction algorithm based on the Kalman filter to solve these problems.

As an optimal auto-regressive data processing algorithm, the Kalman filter algorithm has good numerical estimation and optimization characteristics and is often used in prediction models in various fields. In the area of file transfer in a narrow-band environment, the current network environment has a particular relationship with the network environment at the last moment. Therefore, the Kalman filter algorithm we choose will be able to predict the optimal state in the current network environment. At the same time, the computational simplicity of the Kalman filter significantly reduces the computational delay required for our destination node to predict link quality.

When predicting packet loss, the Kalman filter's low peaks prediction may affect file transfer reliability. Therefore, we must filter the collected anomalous packet loss probabilities before making Kalman filter predictions. Let the packet loss rate dataset be $X(x_1 \quad x_2 \quad \ldots \quad x_n)$, and use Equation (6) to perform low-peak filtering based on the sigma criterion on the data set *X*. In the algorithm, according to the sigma principle in the normal distribution, the probability of numerical distribution on (*x-sigma, x+sigma*) is 68.27%. When the difference between the average packet loss rate and the current packet loss rate is more significant than sigma, we consider the current low peak value as an abnormal value and set the outlier as the average value. At this time, outliers are filtered to make our prediction results more accurate, thereby improving the efficiency of adaptive network coding.

$$\overline{x} = \frac{1}{n} \times \sum_{i=0}^{n} x_i \tag{5}$$

$$\overline{x} - x_i < sigma = \sqrt{\frac{i}{n} \times \sum_{i=1}^{n} (\overline{x} - x_i)^2} \ . \tag{6}$$

We divide the Kalman filter algorithm into two parts: the prediction process and the update process. The prediction process refers to predicting the optimal system packet loss rate R_i^- and estimated covariance P_i^- at the current moment according to the optimal system packet loss rate R_{i-1} and estimated covariance P_{i-1} at the last moment. Equation (7) is the prediction formula of the optimal system packet loss rate. In the equation, *A* is the state transition matrix in the prediction process, and *B* is the control input matrix. A one-dimensional Kalman filter algorithm is used to predict the packet loss rate. In the actual network environment, it is considered that the packet loss rate at the next moment remains unchanged, and there is no control gain. So the value of *A* is 1, and the value of *B* is 0. Equation (8) is the prediction formula of the optimally estimated covariance, where *D* is the process noise covariance. We believe that the prediction of the network packet loss rate quickly is relatively stable, and the error between the state transition matrix and the actual process does not exceed 0.01. So the value of *D* in the method is defined as 0.01.

$$R_i^- = A \times R_{i-1} + B \times U_i = R_{i-1} \, , \tag{7}$$

$$P_i^- = A \times P_{i-1} \times A^T + D = P_{i-1} + 0.01 \, . \tag{8}$$

The update process is to optimize the predicted system state through the current network's actual measurement result x_i. At this time, we will first calculate the Kalman gain value I_i in the current environment through Equation (9) and update the currently predicted optimal state of the system according to I_i. Equations (10) and (11) show the specific update calculation formula. In Equation (9), the parameter *H* represents the state observation matrix for converting the measured value to the actual value. In calculating the packet loss rate, the packet loss rate is calculated according to the data packet sequence number. The measured packet loss rate directly represents the actual packet loss rate, so the value of *H* is 1. The parameter *Y* is the measured noise covariance, through which we can adjust the confidence of our prediction algorithm on the predicted and actual results. Since the packet loss rate measurement in a short time will likely have errors, the prediction results will be more trusted in our prediction algorithm, and the value of *Y* is defined as 0.02.

$$I_i = \frac{P_i^- \times H^T}{H \times P_i^- \times H^T + Y} = \frac{P_i^-}{P_i^- + 0.02} \quad (9)$$

$$R_i = R_i^- + I_i \times (x_i - R_i^-) \quad (10)$$

$$P_i = (1 - I_i \times H) \times P_i^- . \quad (11)$$

We obtain the final optimal system packet loss rate R by performing Kalman filtering on the packet loss rate dataset. Assuming that the network packet loss rate remains unchanged in the following period, the packet loss has a binomial distribution characteristic. Under the current network conditions, according to the features of the binomial distribution, we can use Equation (12) to calculate the probability Q of receiving more than K packets from N packets.

$$Q = \sum_{i=K}^{N} \frac{N!}{i!(N-i)!} \times (1-R)^i \times R^{N-i} . \quad (12)$$

Since the binomial distribution approximates the normal distribution, we can further derive the normal distribution's standard score from the binomial distribution's mean and variance. Equation (13) is the calculation formula of the K value and the N value obtained according to the standard score calculation formula of the normal distribution. In the procedure, $f(Q)$ represents the standard score corresponding to the probability Q under the standard regular distribution table. In the paper, we limit the number of packets in each group, and when the file length is sufficient, the value of N is 256.

$$K = -f(Q) \times \sqrt{N \times (1-R) \times R} + N \times (1-R) . \quad (13)$$

To further improve the efficiency of file transfer in a narrow-band environment, it is necessary to reduce the number of data requests during file transfer as much as possible. In our scheme, the number of data requests equals the sum of the number of file groups and retransmission requests. Because with the help of error correction packets, the probability of a set of files needing second retransmission is small. Therefore, when the likelihood of successful transmission of each group of files is Q, the approximate solution of the number of requests T required for successful file transmission can be obtained by Equation (14). In Equation (14), L_f represents the length of the file.

$$T = \frac{L_f}{K \times 1024} + \frac{L_f}{K \times 1024} \times (1-Q) . \quad (14)$$

The number of data requests is related to the network packet loss rate R and the success rate Q of file packet transmission. We need more error correction packets in an environment with a high packet loss rate to ensure successful file transfer. Similarly, the greater the probability Q of successful file packet transmission under the same network packet loss rate, the more error correction data packets are required. To calculate the optimal value of the number of data requests, we obtained the suboptimal relationship between R and Q through repeated verification operations in Equation (14). The relationship between R and Q is as Equation (15) shown.

$$Q = \begin{cases} 0.99, R \le 0.35 \\ 0.95, 0.35 < R < 1 \end{cases} . \quad (15)$$

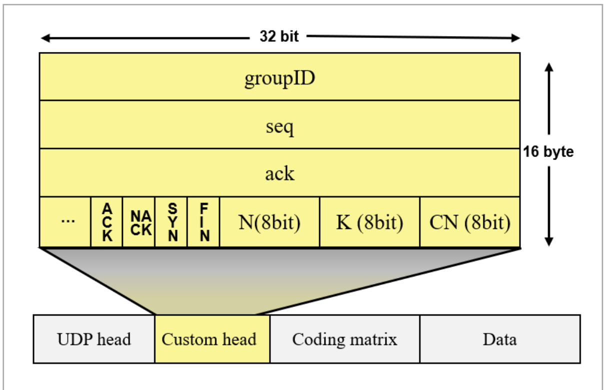
3.3. Transmission Scheme and Algorithm Design

To improve file transfer efficiency in a narrow-band environment, we propose an adaptive file coding scheme that combines the Kalman filtering algorithm and RLNC. In the method, the source node realizes the grouping, coding, and sending of files according to the coding coefficients predicted by the Kalman filter algorithm. The intermediate node completes the data recoding operation to enhance the reliability of the data transmission of the intermediate node. The destination node decodes the file after receiving enough packets and predicts the coding coefficients of the next group of files.

In realizing file transfer, we uniformly define the data packet format. As shown in Figure 1, the field *groupID* represents the sequence number of the current group. Field *seq* means the sequence number of the packet. In our scheme, field *seq* is incremented to facilitate packet loss statistics. Field *ack* represents the acknowledgment number for data reception. The status bit indicates the type of data packet, where ACK is an acknowledgment packet, field NACK is a feedback packet, field SYN is a connection request packet, and field FIN is a termination packet. N and K represent the coding coefficients of the current file, and CN represents the packet offset of the current group. During the data-receiving process, we can confirm whether the current data packet has been received through the *groupID*, and *CN* values.

Figure 1

Packet format diagram



At the same time, we have carried out a detailed algorithm design of the file transfer scheme from the four levels of a source node, an intermediate node, a destination node, and a prediction algorithm. In the algorithm, *S* represents the resource node, *I* represents the intermediate node, and *D* represents the destination node.

Algorithm 1 introduces the algorithm process of source node coding transmission in detail. In the coding process, to ensure an available transmission link between nodes, the source node will first send a connection request with the initial information of the file. If there is an open link in the network at this time, the destination node will return an acknowledgment packet with coding coefficients. The source node implements the grouping of files and random linear coding according to the coding coefficients in the acknowledgment data packets. After the source node completes the coding and sending of the current group, it waits for the destination node to receive confirmation and update the coding coefficients of the next group of files until the last group of files is sent.

Algorithm 1. Source Node Coding Strategy

1: ***S*** sends a connection packet ***pkg1*** to ***D***

2: ***S*** receives the **ACK** packet ***pkg2*** from ***D***

3: ***S*** gets ***N*** and ***K*** values from ***pkg2***.

4: **While *TRUE***:

5: ***S*** reads *K*(kb) data from the file to construct a data matrix ***G***.

6: ***S*** generates a random coding matrix ***M*** of size $N-\text{by}-K$.

7: ***S*** generates encoded data ***C*** with ***G*** and ***M***.

8: ***S*** combines ***M*** and ***C*** to get ***W***.

9: ***S*** adds header information to ***W*** to get the data packet.

10: ***S*** sends the packet to the network.

11: ***S*** receives the **ACK** packet ***pkg3*** from ***D***.

12: ***S*** get and update ***N*** and ***K*** from ***pkg3***.

13: ***S*** judges whether all files are sent.

14: ***S*** sends a **FIN** packet.

15: **End While**

Algorithm 2. Intermediate Node Coding Strategy

1: ***I*** receives packet ***pkg4*** from ***S***.

2: ***I*** gets ***groupID*** and ***CN*** from ***pkg4***.

3: **If *GFlag*** != ***groupID*** or ***Cache***.has(***CN***):

4: ***I*** discard ***pkg4***.

5: **Else**:

6: ***pkg4*** cached in ***Cache***.

7: **If *time*** < ***t***:

8: **continue**.

9: **Else:**

10: ***I*** gets W_t from ***Cache***.

11: ***I*** generates a $(t+1)-by-t$ coding matrix M_t.

12: ***I*** generates re-encoded data W_t' with W_t and M_t.

13: ***I*** sends the re-encoded data in packets.

14: **End If**

15: **End If**

In Algorithms 2 and 3, we will distinguish different packages in the same group of files by *groupID* and *CN*. When a node receives a data packet, it needs to judge whether the group numbers of the data packets are the same and receive unreceived packages with the same group. In addition, the algorithm defines *GFlag* as the current group sequence number, *Cache* as the cache of received packets, and count as the current number of received packets.

Algorithm 3. estination Node Coding Strategy

D receives the connection packet ***pkg5***.
D gets the initial file information ***F*** from ***pkg5***.
D initializes coding coefficients according to ***F*** and network state.
D returns a receipt confirmation packet to **S**.
While *TRUE*:
D receives packet ***pkg6*** from ***I***.
D gets ***seq***, ***groupID***, ***CN*** and **FIN** from ***pkg6***.
D according to ***seq*** statistics packet loss rate.
If *GFlag* != ***groupID*** or ***Cache***.has**(*CN*)**:
D discards ***pkg6***.
Else:
pkg6 cached in ***Cache***.
count++.
End If
If *count* >= *K*:
D gets ***M*** and ***C*** from ***Cache***.
If *M* is Linear correlation:
Receive window slide down.
Else:
D decodes ***C*** to raw data according to ***M***.
goto algorithm 4.
D sends an **ACK** packet to ***S***.
End If
End If
If FIN:
break.
End If
End While

Algorithm 4. Coding Coefficient Prediction Algorithm

D filters dataset *X* according to the sigma criterion.
D performs Kalman filter operation on ***X***.
D gets the result *R* from the Kalman filter operation.
If *R* <= 0.35:
D calculates the coding coefficients with 99%.
Else:
D calculates the coding coefficients with 95%.
End If.

The intermediate nodes also use random linear coding to improve the data throughput further and enhance the reliability of data transmission. In Algorithm 2, after receiving the data packet, the intermediate node judges whether the data packet is valid according to *groupID* and *CN*. Then random linear coding is performed on all packets received within time *t*. The algorithm's keyword *time* represents the time used to receive packets. At this point, the degree to which intermediate nodes adjust link quality can be controlled based on *time*.

In Algorithm 3, the destination node waits for the connection request from the source node. It initializes the file coding coefficient according to the file length information in the request and the current network packet loss rate. The source node encodes and transmits the file according to the initialized coding coefficients. After receiving the data packet, the destination node judges whether the data packet is valid according to the *groupID* and *CN* in the data packet. Cache valid packets in *Cache* and use *count* to keep track of the number of valid packages received. When *count* equals the coding coefficient *K*, the destination node determines whether the coding matrix *M* of the current *K* data packets is linearly related. If it is linearly dependent, the *Cache* window slides down and receives subsequent packets until *K* linearly independent packages are found. After the destination node completes the reception of the current group, it predicts the coding coefficient of the next group of file transmissions according to the latest network packet loss rate.

In Algorithm 4, we use the Kalman filtering algorithm to effectively predict the probability of data loss in a narrow-band environment. In the prediction algorithm, *X* represents the dataset with the loss rate. Because low-peak outliers can lead to insufficient error correction packets, resulting in more data retransmissions. So the destination node first filters low peak outliers in *X* according to the sigma criterion to achieve more accurate predictions. Secondly, the Kalman filtering algorithm filters the filtered data set *X* to obtain the optimal packet loss probability of the network environment at the current moment. Finally, the coding coefficient of the file transmission under the current network environment is calculated by Equation (13).

4. Performance Analysis

In this chapter, to evaluate the performance of the proposed scheme, we conduct experiments in the MININET simulation environment. First, we assess the feasibility of this scheme from two aspects of network coding and the Kalman filtering algorithm. Furthermore, we compare the performance of the proposed method with the scheme based on automatic retransmission, the scheme proposed by Wu et al. [32] and the scheme proposed by Mothku et al. [24].

4.1. Simulation Settings

We simulate a narrow-band environment with a high delay and high packet loss rate in the simulation through a customized network topology. Table 1 shows the experimental parameters for our simulations.

Table 1
Experimental parameter

item	value
CPU	i5-10210U
OPERATING SYSTEM	Ubuntu
MEMORY	4G
FILE SIZE	11MB
PACKAGE SIZE	1500 Bytes

Under the above simulation environment, the proposed scheme will use the Kalman filter algorithm to predict the link packet loss rate and adaptively determine the coding coefficients in RLNC. At this point, the source node will encode and send the packet with the appropriate level of redundancy. In the scheme based on automatic retransmission, to achieve the reliable transmission of the file, the source node will send all lost data packets through retransmission. In the scheme proposed by Wu et al. [32], source nodes control the coding redundancy according to the worst link's lost rate (WLLR). In the scheme proposed by Mothku et al. [24], nodes grade the redundancy of network coding and determine the applicability and redundancy level through the Markov decision process (MDP). To increase the comparability of comparative experiments, we define the N value based on the WLLR scheme as 256 and the K value based on the MDP scheme as 128.

4.2. Coding Performance Evaluation

During the file coding process, we will perform random linear coding on the file according to the coding coefficients. At this time, the efficiency of file coding and decoding will affect the performance of our file transmission. Therefore, it is essential to analyze the performance of file decode according to different redundancy rates. In the coding and decoding method of matrix multiplication, the length of the file is L_f. Then, the number of operations required to encode the file can be calculated by Equation (16). It can be seen from the formula that the time of file coding is only affected by the value of the coding coefficient N. Similarly, we can calculate the number of operations required for file decoding by Equation (17). It can be seen from the formula that the decoding time is affected by the value of the coding coefficient K.

$$T_{coding} = N \times L_f \, , \tag{16}$$

$$T_{decoding} = (\frac{K^2}{1024} + K) \times L_f \, . \tag{17}$$

Figures 2-3 show the time required to complete the coding and decoding of a file with a size of 11MB under different coding coefficient ratios. As can be seen from Figure 2, under the condition of a specific N value, the percentage of the K value to the N value does not affect the coding time of the file. However, in Figure 3, the file's decoding time increases with the coding coefficient ratio. Under the same conditions, the

Figure 2
Influence of Coding Coefficients on Coding Time

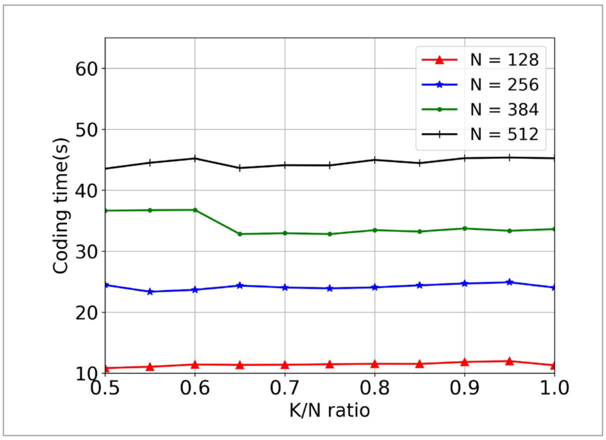
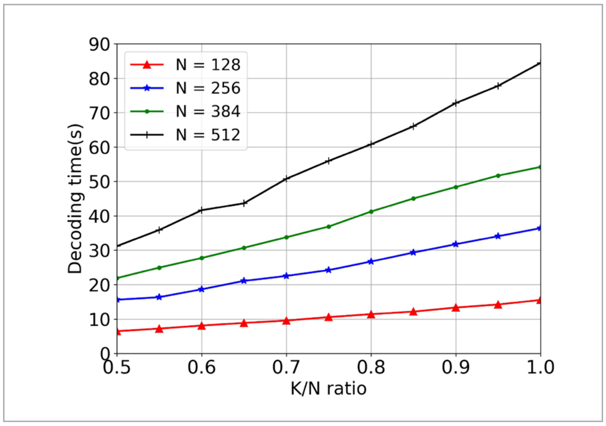


Figure 3

Influence of coding coefficients on decoding time.



larger the N value, the lower the coding and decoding efficiency. In our scheme, to balance the influence of network transmission delay and coding delay, the size of the N value is set to 256, and the size of the K value is obtained adaptively through the prediction algorithm.

4.3. Prediction Algorithm Performance Evaluation

Different network packet loss rates during the experiment will affect the file transfer performance. Therefore, predicting the optimal network packet loss probability in different periods and ensuring that source nodes can encode with the most appropriate coding coefficients is essential. Figure 4 introduces the relationship between the actual packet loss rate, the packet loss rate filtered by the sigma criterion, and the packet loss rate predicted by the Kalman filter algorithm. It can be seen from the figure that the data set filtered by the sigma criterion is more accurate, which improves the prediction accuracy of the Kalman filter algorithm.

Figure 4

Prediction result of packet loss rate

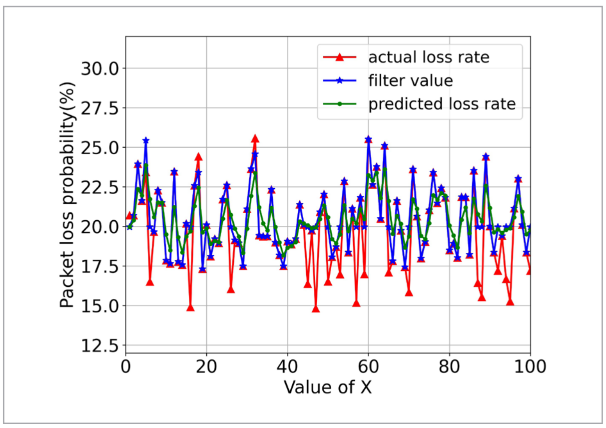
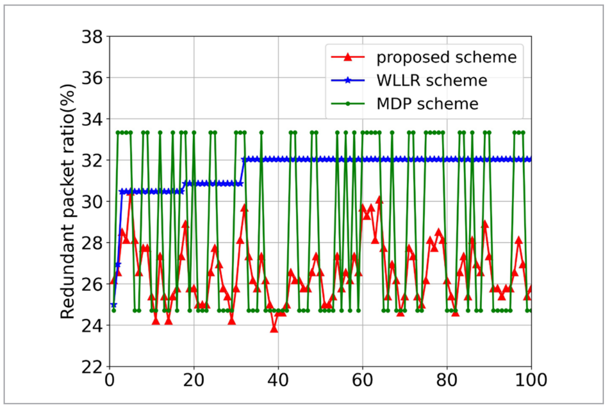


Figure 5 displays the percentage of redundant data packets generated during file transmission using different network coding schemes in a simulated environment with a 20% packet loss rate. As seen from the figure, the scheme of calculating redundancy through the WLLR is about 5% higher than our scheme in the number of error correction packets. At this time, the increase of error correction packets will increase the transmission delay of the system and the consumption of network resources. Compared to the MDP-based mechanism, our scheme offers greater flexibility and precision in controlling the number of redundant packets by implementing a linear redundancy calculation algorithm. The strategy based on automatic retransmission adopts the method of lost retransmission and does not require additional error correction packets. However, the system overhead caused by frequent retransmission requests is more significant.

Figure 5

Redundant packet ratio comparison



4.4. Evaluation of Experimental Results

This section compares the performance of our scheme with the scheme based on automatic retransmission, the scheme based on WLLR and the scheme based on MDP across four metrics: number of packets transmitted, number of data requests, number of timeout retransmissions, and file transfer time. The effects of transmission delay and network packet loss rate on each scheme are considered in the simulation environment.

Figures 6-7 show the experimental results of the effect of transmission delay on the number of packets and request times in a 10% packet loss environment, respectively. It can be seen from Figures 6-7 that, unlike the TCP protocol, the number of data packets and packet requests of the file transfer scheme implemented based on the UDP protocol is not affected by the transmission delay. In addition, compared with the scheme based on WLLR, our method requires fewer packets and fewer requests to complete the file transfer. A strategy based on automatic retransmission requires the least number of packages but requires a much larger number of requests than a strategy based on network coding. The MDP-based scheme has a fixed coding coefficient of K during data transmission, and the redundancy control is not flexible enough, so it requires more packets. In contrast, our scheme dynamically adjusts the value of K based on the network environment. As the K value increases during file transfer, the number of valid data packets also increases, reducing the number of data requests needed to complete the transfer.

Figure 6

The effect of transmission delay on the number of packets

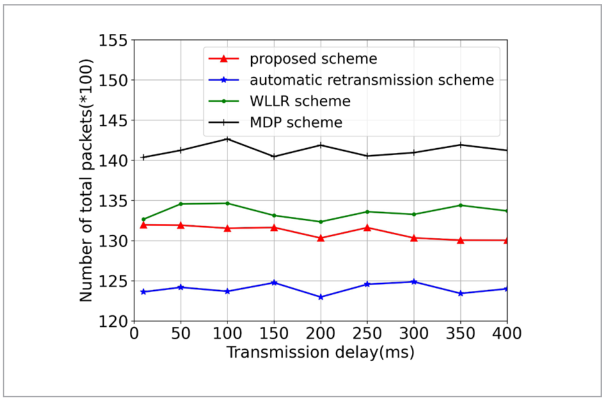
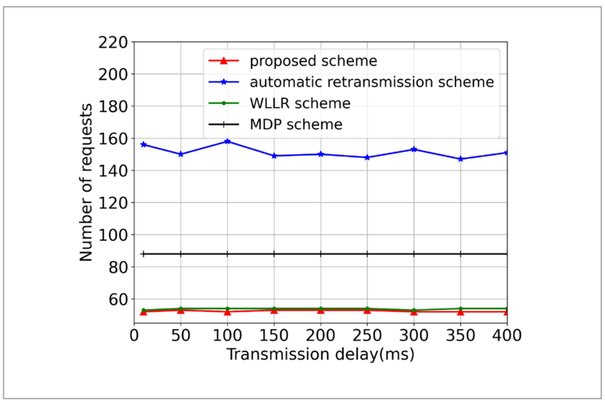


Figure 7

The effect of transmission delay on the number of requests



During the file transfer process, the most critical factor affecting the file transfer performance is the timeout waiting. Figure 8 shows the effect of transmission delay on the number of timeout retransmissions. The figure shows that the transmission delay does not affect the number of timeout retransmissions. However, our scheme has the least timeout retransmissions compared with other methods because our procedure can complete the transfer of the entire file with the least number of requests.

Figure 8

The effect of transmission delay on the number of timeout retransmissions

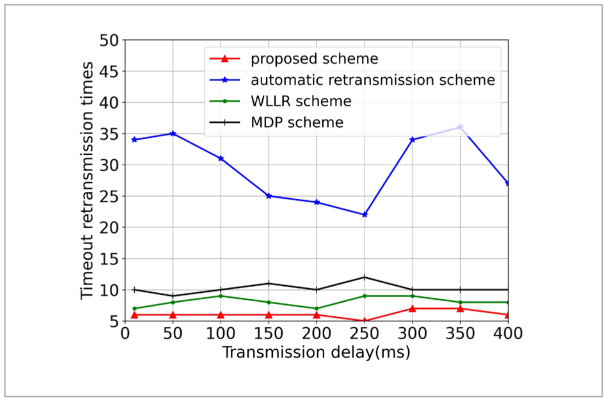


Figure 9 shows the effect of transfer delay on file transfer time. As the transfer delay increases, the more time it takes for us to complete the file transfer, as can be seen from the graph. However, our method still requires less time to complete the file transfer than the automatic retransmission scheme and the WLLR scheme. Because our scheme requires fewer packets and requests than the scheme based on WLLR, it requires less waiting time and transmission delay. The method based on automatic retransmission has the lowest performance because in the network environment with high delay and high packet

Figure 9

The effect of transmission delay on file transfer time

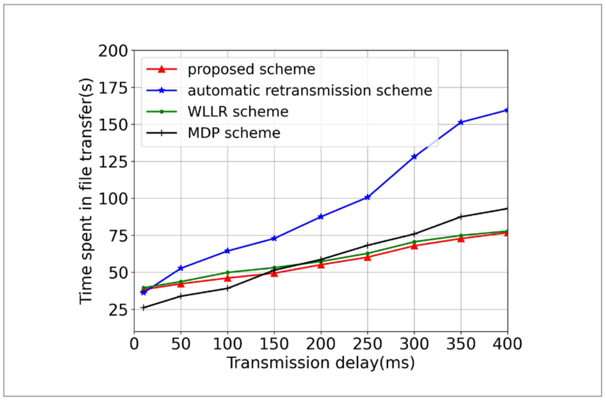
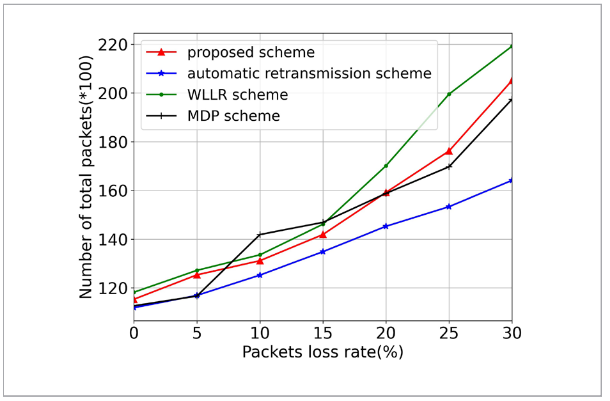


Figure 10

The effect of packet loss rate on the number of packets



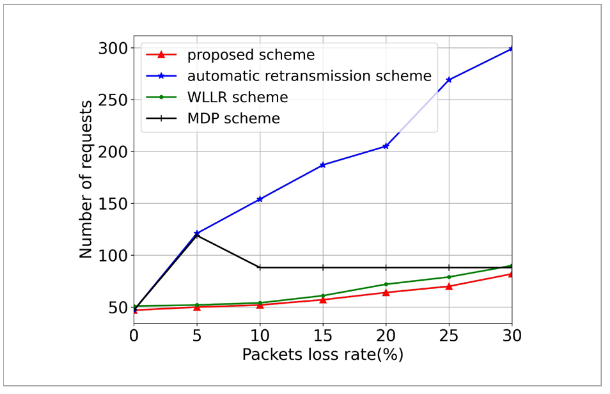
loss rate, frequent data retransmission will lead to extensive waiting time and propagation delay. The MDP-based scheme performs better under low delay conditions, as it requires less coding time. The experimental results show that in the narrow-band environment, with the continuous increase of the transmission delay, the performance advantage of our scheme will be more obvious.

Compared with transmission delay, the impact of packet loss rate on file transmission is more prominent. In the experiment, we simulated the effect of different packet loss rates on file transmission under the 10ms transmission delay in the simulation environment. Figure 10 shows the impact of packet loss rate on the number of packets. As can be seen from the figure, with the increase in the packet loss rate, the file transfer scheme based on network coding has a significantly higher growth in the number of data packets than the file transfer scheme based on automatic retransmission. The reason for this result is that the coding operation increases the redundancy of the data packet. The coding scheme based on WLLR has the fastest growth in the number of data packets because the loss rate of the worst link cannot accurately express the current network environment during file transmission, resulting in unnecessary data redundancy. The MDP-based scheme manages the number of redundant packets using a hierarchical approach, causing a greater fluctuation in the number of redundant packets at boundaries.

Figures 11-12 show the relationship between the packet loss rate and the number of data requests and time-

Figure 11

The impact of packet loss rate on the number of requests



out retransmissions. As the figure shows, our scheme has better stability than other schemes. The automatic retransmission scheme is greatly affected by the packet loss rate because, in an environment with a high packet loss rate, many retransmission requests are required to ensure the reliable transmission of files. In contrast, the packet loss rate has less impact on network-coding-based schemes because the existence of redundant data avoids the retransmission of large amounts of data. Figures 11-12 show that our scheme outperforms other schemes because the Kalman-filter-based coding coefficient prediction algorithm allows us to complete file transfers with fewer redundant packets and requests. In the process of file transfer, the fewer the number of requests, the less the number of timeout retransmissions will occur. As the file size and packet loss rate continue to increase, our solution's advantages will become more evident.

Figure 12

The effect of packet loss rate on the timeout retransmissions

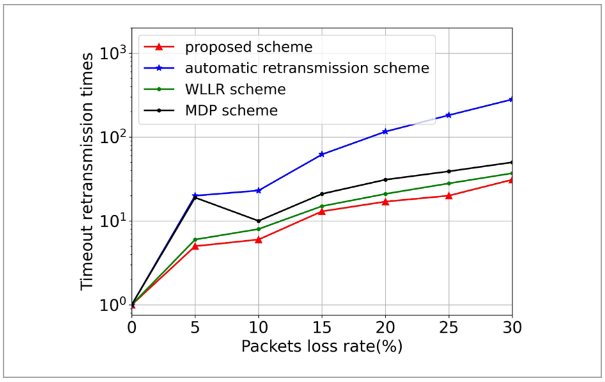
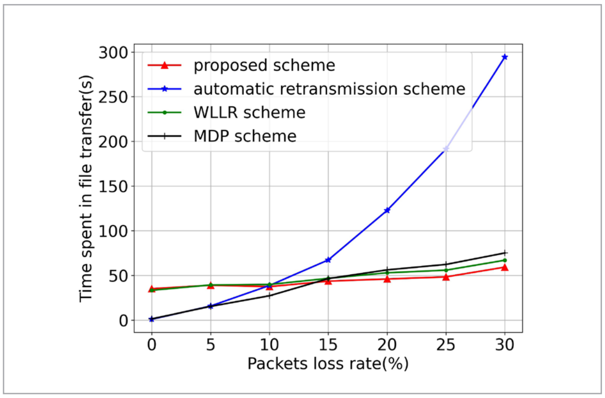


Figure 13

The effect of packet loss rate on file transfer time



In our network-coding-based file transfer scheme, file transfer time is affected by coding delay, transmission delay, propagation delay, and latency delay. In a narrow-band environment with a high delay and packet loss rate, the network-coding-based file transmission scheme can reduce the performance consumption caused by propagation delay and wait for delay by reducing the number of data requests. Figure 13 shows the effect of different packet loss rates on file transfer time under a 10ms transmission delay. As seen from the figure, the scheme based on automatic retransmission is insufficient in file transmission performance in a high packet loss environment. The reason is that frequent retransmissions will bring many round-trip propagation delays and timeout waiting in the background of a high packet loss rate. The MDP-based scheme uses a Markov decision process to determine the link status and opt for a non-network coding scheme when the packet loss rate is low, making it more effective. However, when network quality declines, our scheme's utilization of error correction packets to minimize data requests improves file transfer performance in low-bandwidth environments.

5. Summary and Conclusion

Aiming at the performance problem of TCP file transfer in a narrow-band environment, we propose a reliable file transfer scheme based on random linear network coding and the Kalman filtering algorithm. The system predicts the optimal packet loss rate of the current network through the Kalman filter algorithm so that the random linear coding of the file has an appropriate coding coefficient. The method effectively reduces the number of data requests in the file transmission process and improves the performance and stability of the file transmission. However, in the case of low latency and low packet loss rate, coding delay significantly impacts our scheme's performance. There is yet to be an excellent solution to balance the effect of encoding delay and transmission delay on file transmission performance. In our future endeavors, we aim to minimize the coding delay caused by network coding and enhance the accuracy of the prediction algorithm to boost the overall performance of file transfer in various environments.

References

1. AL-Dhief, F. T., Sabri, N., Latiff, N. M. A., et al. Performance Comparison between TCP and UDP Protocols in Different Simulation Scenarios. International Journal of Engineering & Technology, 2018, 7(4.36), 172-176.
2. Ahlswede, R., Cai, N., Li, S. Y., Yeung, R. W. Network Information Flow. IEEE Transactions on Information Theory, 2000, 46(4), 1204-1216. https://doi.org/10.1109/18.850663
3. Bilbao, J., Crespo, P. M., Armendariz, I., Medard, M. Network Coding in the Link Layer for Reliable Narrowband Powerline Communications. IEEE Journal on Selected Areas in Communications, 2016, 34(7), 1965-1977. https://doi.org/10.1109/JSAC.2016.2566058
4. Chen, M., Miao, Y., Hao, Y., Hwang, K. Narrow Band Internet of Things. IEEE Access, 2017, 5, 20557-20577. https://doi.org/10.1109/ACCESS.2017.2751586

5. Cai, L., Zhang, Z., Yang, J., Yu, Y., Zhou, T., Qin, J. A Noise-Immune Kalman Filter for Short-Term Traffic Flow Forecasting. Physica A: Statistical Mechanics and its Applications, 2019, 536, 122601. https://doi.org/10.1016/j.physa.2019.122601

6. Duke, M., Braden, R., Eddy, W., Blanton, E., Zimmermann, A. A Roadmap for Transmission Control Protocol (TCP) Specification Documents, 2015. https://doi.org/10.17487/rfc7414

7. Ding, Y., Yu, P., Guo, J., Shi, X., Zhang, Y. Transmission Guarantee Method of End-to-End Service in Narrowband Dynamic Environment. In: 2020 International Wireless Communications and Mobile Computing (IWCMC), IEEE, 2020, 204-209. https://doi.org/10.1109/IWCMC48107.2020.9148061

8. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T. Hypertext Transfer Protocol--HTTP/1.1. 1999. https://doi.org/10.17487/rfc2616

9. Gien, M. A File Transfer Protocol (FTP). Computer Networks (1976), 1978, 2(4-5), 312-319. https://doi.org/10.1016/0376-5075(78)90009-0

10. Harkanson, R., Kim, Y., Jo, J. Y., Pham, K. Effects of TCP Transfer Buffers and Congestion Avoidance Algorithms on the End-to-End Throughput of TCP-Over-TCP Tunnels. In: 16th International Conference on Information Technology-New Generations (ITNG 2019). Springer, Cham, 2019, 401-408. https://doi.org/10.1007/978-3-030-14070-0_55

11. Ho, T., Médard, M., Koetter, R., Karger, D. R., Effros, M., Shi, J., Leong, B. A Random Linear Network Coding Approach to Multicast. IEEE Transactions on Information Theory, 2006, 52(10), 4413-4430. https://doi.org/10.1109/TIT.2006.881746

12. Harvey, N. J. A. Deterministic Network Coding by Matrix Completion. Massachusetts Institute of Technology, 2005.

13. Hwang, J. H., Yoo, C. Formula-Based TCP Throughput Prediction with Available Bandwidth. IEEE Communications Letters, 2010, 14(4), 363-365. https://doi.org/10.1109/LCOMM.2010.04.092309

14. Iqbal, M. F., Zahid, M., Habib, D., John, L. K. Efficient Prediction of Network Traffic for Real-Time Applications. Journal of Computer Networks and Communications, 2019. https://doi.org/10.1155/2019/4067135

15. Joshi, M., Hadi, T. H. A Review of Network Traffic Analysis and Prediction Techniques. arXiv preprint arXiv:1507.05722, 2015.

16. Kosut, O., Tong, L., Tse, D. Nonlinear Network Coding is Necessary to Combat General Byzantine Attacks. In: 2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton). IEEE, 2009, 593-599. https://doi.org/10.1109/ALLERTON.2009.5394912

17. Kumar, S. V. Traffic Flow Prediction Using Kalman Filtering Technique. Procedia Engineering, 2017, 187, 582-587. https://doi.org/10.1016/j.proeng.2017.04.417

18. Keller, L., Atsan, E., Argyraki, K., et al. SenseCode: Network Coding for Reliable Sensor Networks. ACM Transactions on Sensor Networks (TOSN), 2013, 9(2), 1-20. https://doi.org/10.1145/2422966.2422982

19. Langley, A., Riddoch, A., Wilk, A., et al. The QUIC Transport Protocol: Design and Internet-Scale Deployment. In: Proceedings of the Conference of the ACM Special Interest Group on Data Communication, 2017, 183-196. https://doi.org/10.1145/3098822.3098842

20. Li, S. Y. R., Cai, N., Yeung, R. W. On Theory of Linear Network Coding. In: Proceedings. International Symposium on Information Theory, 2005. ISIT 2005. IEEE, 2005, 273-277. https://doi.org/10.1109/ISIT.2005.1523337

21. Liu, Q., Rao, N. On Concavity and Utilization Analytics of Wide-Area Network Transport Protocols. In: 2018 IEEE 20th International Conference on High Performance Computing and Communications, IEEE 16th International Conference on Smart City https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00088. IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, 2018, 430-438.

22. Le, T., Hu, W., Corke, P., Jha, S. ERTP: Energy-Efficient and Reliable Transport Protocol for Data Streaming in Wireless Sensor Networks. Computer Communications, 2009, 32(7-10), 1154-1171. https://doi.org/10.1016/j.comcom.2008.12.045

23. Miller, K., Al-Tamimi, A. K., Wolisz, A. QoE-Based Low-Delay Live Streaming Using Throughput Predictions. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 2016, 13(1), 1-24. https://doi.org/10.1145/2990505

24. Mothku, S. K., Rout, R. R. Markov Decision Process and Network Coding for Reliable Data Transmission in Wireless Sensor and Actor Networks. Pervasive and Mobile Computing, 2019, 56, 29-44. https://doi.org/10.1016/j.pmcj.2019.03.003

25. Naeem, A., Rehmani, M. H., Saleem, Y., Rashid, I., Crespi, N. Network Coding in Cognitive Radio Networks: A Comprehensive Survey. IEEE Communications Surveys & Tutorials, 2017, 19(3), 1945-1973. https://doi.org/10.1109/COMST.2017.2661861

26. Oliveira, T. P., Barbar, J. S., Soares, A. S. Computer Network Traffic Prediction: A Comparison Between Traditional and Deep Learning Neural Networks. International Journal of Big Data Intelligence, 2016, 3(1), 28-37. https://doi.org/10.1504/IJBDI.2016.073903
27. Schmid, J., Höss, A., Schuller, B. W. A Survey on Client Throughput Prediction Algorithms in Wired and Wireless Networks. ACM Computing Surveys (CSUR), 2021, 54(9), 1-33. https://doi.org/10.1145/3477204
28. Sliwa, B., Falkenberg, R., Liebig, T., Pillmann, J., Wietfeld, C. Machine Learning Based Context-Predictive Car-to-Cloud Communication Using Multi-Layer Connectivity Maps for Upcoming 5G Networks. In: 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall). IEEE, 2018, 1-7. https://doi.org/10.1109/VTCFall.2018.8690856
29. Wei, P., Hong, Z., Shi, M. Performance Analysis of HTTP and FTP Based on OPNET. In: 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS). IEEE, 2016, 1-4. https://doi.org/10.1109/ICIS.2016.7550952
30. Wei, B., Kanai, K., Kawakami, W., Katto, J. HOAH: A Hybrid TCP Throughput Prediction with Autoregressive Model and Hidden Markov Model for Mobile Networks. IEICE Transactions on Communications, 2018, 2017CQP0007. https://doi.org/10.1587/transcom.2017CQP0007
31. Wei, B., Kawakami, W., Kanai, K., Katto, J., Wang, S. TRUST: A TCP Throughput Prediction Method in Mobile Networks. In: 2018 IEEE Global Communications Conference (GLOBECOM), IEEE, 2018, 1-6. https://doi.org/10.1109/GLOCOM.2018.8647390
32. Wu, C., Ji, Y., Xu, J., Ohzahata, S., Kato, T. Coded Packets Over Lossy Links: A Redundancy-Based Mechanism for Reliable and Fast Data Collection in Sensor Networks. Computer Networks, 2014, 70, 179-191. https://doi.org/10.1016/j.comnet.2014.05.010
33. Yue, C., Jin, R., Suh, K., Qin, Y., Wang, B., Wei, W. LinkForecast: Cellular Link Bandwidth Prediction in LTE Networks. IEEE Transactions on Mobile Computing, 2017, 17(7), 1582-1594. https://doi.org/10.1109/TMC.2017.2756937
34. Zheng, Z., Ma, Y., Liu, Y., et al. Xlink: QoE-Driven Multi-Path QUIC Transport in Large-Scale Video Services. In: Proceedings of the 2021 ACM SIGCOMM 2021 Conference. 2021, 418-432. https://doi.org/10.1145/3452296.3472893
35. Zhu, F., Zhang, C., Zheng, Z., Farouk, A. Practical Network Coding Technologies and Softwarization in Wireless Networks. IEEE Internet of Things Journal, 2021, 8(7), 5211-5218. https://doi.org/10.1109/JIOT.2021.3056580
36. Zhang, Q., Liu, A. An Unequal Redundancy Level-Based Mechanism for Reliable Data Collection in Wireless Sensor Networks. EURASIP Journal on Wireless Communications and Networking, 2016, 2016(1), 1-22. https://doi.org/10.1186/s13638-016-0754-6