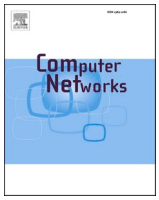
Computer Networks 236 (2023) 110012

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



UDP-RT: A UDP-based reliable transmission scheme for power WAPS☆

Qiuyu Lu [a], June Li [a,*], Kai Yuan [b], Kaipei Liu [c], Ming Ni [d], Jianbo Luo [e]

[a] School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China
[b] School of Electrical and Electronic Engineering, Huazhong University of Science and Technology, Wuhan 430072, China
[c] School of Electrical Engineering and Automation, Wuhan University, Wuhan 430072, China
[d] Leidos Engineering LLC, Chicago 60607, USA
[e] NARI Group Electric Power Research Institute, Nanjing 211106, China

ARTICLE INFO

Keywords:
Power wide-area protection
Network communication
Error correction
Real-time performance
Reliability

ABSTRACT

It is expected that TCP/IP networks take the place of point-to-point fiber channels for the communications of WAPS. In TCP/IP networks, TCP does not guarantee real-time while UDP does not guarantee reliability. An efficient method is demanded for WAPS to guarantee the real-time and reliability of messages transmitted in TCP/IP networks under congestion states. To address the challenge, we propose a UDP-based reliable transmission (UDP-RT) scheme, which achieves low latency by adopting UDP at the transport layer and high reliability by adding the mechanisms of error correction, error detection, resending and timeout retransmission at the application layer. The error correction mechanism employs TPCs, which has low complexity and good performance at high code rate, to correct errors in a message. A blocking rule for TPCs considering the features of communication channels and messages of WAPS is presented. The error detection mechanism is for detecting whether all errors in the message are corrected by the error correction mechanism. The resending mechanism and the timeout retransmission mechanism (optional) is for ensuring the reliability in the two cases of message loss and not all errors corrected. Additionally, algorithms for UDP-RT are presented, and their correctness are validated through experiment. Our analyses demonstrate that the proposed scheme can meet the real-time requirements of WAPS businesses when network congesting and has higher reliability than the TCP transmission scheme and other UDP transmission schemes.

1. Introduction

The major purpose of power wide-area protection system (WAPS) is to maintain huge interconnected power systems stable and prevent them from collapsing [1,2]. To achieve this goal, WAPS requires real-time and reliable communications for uploading measurements and issuing control instructions [3,4]. Otherwise, the execution units (EUs) may fail to operate or happen mal-operation threatening the safe and stable operations of the power systems [5,6]. The response time of different wide-area protection functions (the time to form a decision and take an action after a wide-area disturbance begins) ranges from 100 ms to 100 s, and the specific response time relates to the business type, the scale of the power grid and the unit location [7]. For instance, the system can keep reliability and stability only if the out-of-step relaying acts within 500 ms after power angle stability loss begins, and the delay for uploading measurements and issuing control instructions cannot exceed 370 ms [8].

Currently, point-to-point fiber channels are used for the communications of power protection system, which is improper for large-scale WAPS due to the huge investment requirements. It is expected that TCP/IP networks take the place of point-to-point fiber channels for the communications of WAPS [1,2]. Transmission control protocol (TCP) and user datagram protocol (UDP) are the transport layer protocols of TCP/IP networks. TCP provides reliable delivery for application messages by acknowledgement and retransmission with congestion-control mechanism. While, UDP does not [9]. Thus, when congestion is identified, TCP may reduce the sending rate, resulting in higher message delay [10,11]. Unlike TCP, UDP sends out messages at a constant rate whether the network is congested or not. That is, when congestion occurs, TCP traffic decreases while the UDP traffic dose not [12], which would

☆ This document is the results of the National Natural Science Foundation of China (Research on the evolution mechanism and early defense of cascading failures across spaces caused by coordinated cyberattacks in Grid CPS) under Grant 51977155.

* Corresponding author.
E-mail address: jeli@whu.edu.cn (J. Li).

https://doi.org/10.1016/j.comnet.2023.110012
Received 20 September 2022; Received in revised form 24 June 2023; Accepted 1 September 2023
Available online 9 September 2023

further reduce the TCP traffic and increase the TCP message delay. In [13], we proved by simulation that with the increase of congestion, the end-to-end delay of TCP messages increases exponentially, while that of UDP messages increases slightly and stays within a certain range.

The WAPS communications are extensive, long-distance and involve a variety of secondary equipment, so the communication networks of WAPS are more vulnerable to cyberattacks and easier to be in congesting than point-to-point communications. Cyberattacks like DoS and DDoS are the most common attacks in power systems, and have been proven lethal to a wide range of power system elements [14,15]. Common approaches for ensuring the real-time and reliability of WAPS communications include network topology optimization, MPLS-based traffic engineering and routing optimization [16–18], queue scheduling based on multi-priority services [19,20], and different services (DiffServ) [21, 22]. However, if TCP is used at the transport layer, these approaches still cannot solve the problem of excessive time delay caused by network congestion and retransmission. UDP can tackle this problem, but it does not guarantee reliability. Therefore, an effective method that provides real-time and reliable transmission is required by WAPS, and it should keep the communication latency to a reasonable level even in the worst situation.

In our previous work, we reported a reliable communication approach based on UDP for power WAPS [13]. However, in literature [13], we found in our follow-up study that bose chaudhuri hocquenguem (BCH) codes employed by the error correction mechanism is not the best in error correction capability and inability to correct burst errors, and the experiment is incomplete. Here, an improved UDP-based reliable transmission (UDP-RT) scheme, a follow-up study, is put forward. The proposed scheme in this paper uses UDP at the transport layer for low latency with adding the mechanisms of error correction, error detection, resending, and timeout retransmission to the application layer for high reliability. Turbo product code (TPC) is employed by the error correction mechanism to take the place of BCH in [13]. Compared to BCHs, TPCs have better error correction capability and can correct both of random errors and burst errors. Theoretical analysis and simulation results reveal that the proposed scheme can meet the real-time requirements of WAPS businesses when network congestion occurs and has higher reliability than the TCP transmission scheme and existing UDP transmission schemes. The contributions of our work are listed as follows.

(1) To address the issue that UDP does not provide reliable delivery, an error correction mechanism is designed for the application layer to correct transmission errors in the message. By comparing the decoding performance and complexity of four common error correction codes, TPCs is selected for WAPS communications; By comparing the coding rate and error correcting capability of three common TPCs, TPC(64,57,4)×(64,57,4) is selected for encoding and decoding a message in blocks; Considering the features of channels and messages of WAPS, a blocking rule for TPCs is designed to calculate the blocks number of messages with different lengths. The mechanism has low complexity and negligible impacts on the performance of applications.
(2) To address the issue that the error correction mechanism cannot correct the errors exceeding its capability, an error detection mechanism is designed for determining whether all errors in the message are corrected. It does not increase the message processing delay because the adopted checksum algorithm is identical to UDP and the UDP checksum is disabled.
(3) To address the issue that the reliability of messages cannot be guaranteed by the error correction and error detection mechanisms as error correction fails and message loss, a resending mechanism that sends multi-copies of the message in succession is designed, which avoids the delay introduced by waiting timeout.
(4) Combining the above three mechanisms, algorithms for UDP-RT are presented, and their correctness are validated through experiment. These algorithms provide a reference for implementing UDP-RT.

The remainder of this paper is organized as follows. We first overview the related works in Section 2. UDP-RT scheme is presented in Section 3. The correctness validation of the proposed scheme is provided in Section 4. Real-time performance and reliability of the proposed scheme are analyzed in Sections 5 and 6, respectively. Finally, we conclude this paper in Section 7.

2. Related work

IEC61850–90 [23] advises to transmit routable SV (R-SV) and GOOSE (R-GOOSE) messages via UDP for wide-area monitoring, protection and control. The reliability of the messages is ensured by the IP priority tagging and the retransmission mechanism. The IP priority tagging is employed to inform routers to forward the messages with higher priority preferentially, so that reduces the messages delay when network congesting. It requires all the routers of the network support the IP priority tagging and is a “best-effort-delivery” mechanism. The retransmission mechanism for R-GOOSE messages is inherited from GOOSE protocol, which repeatedly send the same message as long as the message data does not change. The retransmitted messages consume an excessive amount of bandwidth, which might result in the network more susceptible to congestion [24]. On the other hand, the retransmission messages may be discarded for multiple consecutive times when bursting errors in the channels, causing the receiver cannot receive the correct message within the expected maximum latency. In addition, although the retransmission mechanism for R-SV messages is recommended by the standard, no specific solution is provided.

Several reliability assurance solutions based on UDP have been developed for the communication services with high real-time requirements in power grid. Fan et al. [25] propose a scheme based on UDP for the transmission of GOOSE messages in wide-area networks (WAN), which ensures the reliability of GOOSE messages by the dynamic retransmission mechanism. However, as the retransmission interval of the retransmission mechanism is quite large when no new event occurs, the real-time performance of messages degrades significantly when fist 5 messages loss or have errors. Xiao et al. [26] provide a strategy that employs UDP for transmitting wide-area measurement system (WAMS) data and adds the mechanisms of retransmission and real-time interpolation at the application layer for the reliability of UDP-based data transfer. The retransmission mechanism employs data-loss check and repeat request, which will significantly lengthen the end-to-end messages delay when the network congesting. Therefore, when network congestion arises, the retransmission mechanism would cause the whole strategy fails to guarantee real-time delivery for messages. Furthermore, the real-time interpolation mechanism is used to estimate the missing measurements. However, this mechanism is not suitable for control instructions which require higher reliability than measurements. Besides, the estimated values might be incorrect, and this may lead to incorrect actions of the receiver, causing significant damage to the power system.

Exiting UDP-based reliable transmission for real-time video streaming mainly use erasure codes to recover the data from packet losses. For instance, Go et al. [27–29] adopt Raptor codes to mitigate retransmission caused by high channel losses, and Liu et al. [30] design a forward error correction strategy which uses Reed-Solomon (RS) codes to improve the reliability of data transfer over UDP. Erasure codes are suitable for the data whose size is considerably large, and are used to correct errors whose positions are already known. The data needs to be split into multiple UDP packets to send them to the receiver, and it is able to be decoded by the receiver only after receiving a certain number of packets. However, the messages in WAPS are mostly around 250bytes

[31,32], which rarely need to be sent in multiple packets. Erasure codes are not appropriate for WAPS as the receiver cannot identify the position of bit errors in the message before decoding.

To summarize, the above solutions are not properly adapted to transmit the messages with high real-time requirement based on UDP in power WAPS. To our knowledge, there are no works about the UDP-based transmission scheme combining with the mechanisms of error correction, error detection and retransmission at the application layer.

3. UDP-based reliable transmission (UDP-RT) scheme

3.1. Overview of UDP-RT scheme

The architecture of UDP-RT scheme is as shown in Fig. 1, which involves the transport layer and the application layer, and is composed of the following two part:

(1) Checksum-disabled UDP is adopted at the transport layer for ensuring real-time delivery of messages regardless of whether the network is congested or not. It ensures the error messages can be submitted to the application layer for error correction at the receiving end that UDP's checksum is disabled.
(2) The mechanisms of error correction, error detection, resending, and timeout retransmission are added to the application layer for ensuring the reliability of messages. The TPC-based error correction mechanism, whose reliability is related to its error correction capability, is provided for correcting transmission errors in messages. However, the error correction mechanism cannot guarantee the reliability in the two cases of message loss and the number of errors in a message going beyond its capability. The error detection mechanism, same as the calculation algorithm of UDP checksum, is provided for further identifying the correctness of the decoded message. The resending mechanism which sends a message N times in succession and the timeout retransmission mechanism are provided to address the problems of message loss and error correction fails.

The timeout retransmission mechanism, as shown in Fig. 2, ensures message delivery to the receiver whether the message is beyond its real-time requirement. The sender initialized an automatic timer when the resending mechanism sends the last resent message. If there is no acknowledgement from the receiver after the timer expires, the message is resent N times in succession. The receiver will send the acknowledgement to the sender when it receives the correct message. When the message is submitted to the applications, the applications compare the timestamp in the message and the local timestamp. If the message meets its real-time requirement, the receiver will operate as required by the

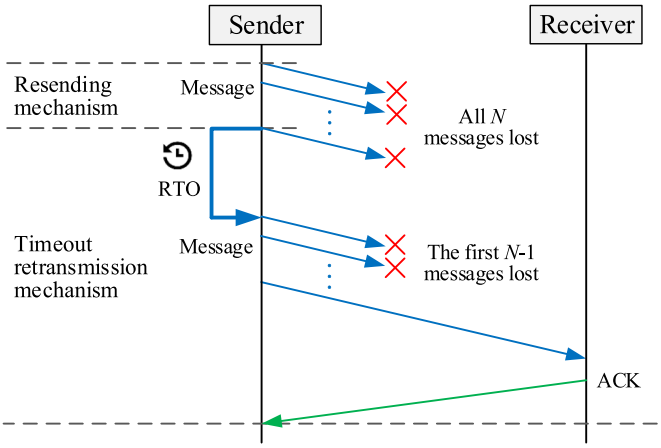


Fig. 2. The timeout retransmission mechanism.

message. If not, the message is used for alarming or accident tracing. The retransmission timeout (RTO) is calculated using Karn's algorithm [33].

The timeout retransmission mechanism is set as an optional mechanism, that the operator of the utility center can choose to use or not based on the practical application scenarios. For wide-area protection functions with extremely high real-time requirements, an appropriate resending count N can ensure at least one of the N resent messages reaches the receiver when slightly congested condition, and nearly all of the retransmitted messages cannot arrive timely when heavily congested condition. In this case, the timeout retransmission mechanism can be optional, which is validated in Section 5.2 through calculation with simulation. For wide-area protection functions with lower real-time requirements, this mechanism remains meaningful as the messages may still meet its real-time requirement after timeout retransmission.

3.2. Error correction mechanism

3.2.1. Error correction code selection

Different error correction codes are different in decoding performance and complexity. WAPS covers a wide range, requires high real-time demands, and commonly uses optical fibers as the communication medium [34]. Hence, the error correction codes should have better performance at high code rate and have lower complexity in order to consume less bandwidth and provide low latency. Moreover, since random and burst errors may occur in fiber channels, the codes should be able to correct both random and burst errors. In order to select the appropriate codes for WAPS communication, we compare four common error correction codes, including Turbo codes, BCHs, RS codes, Low

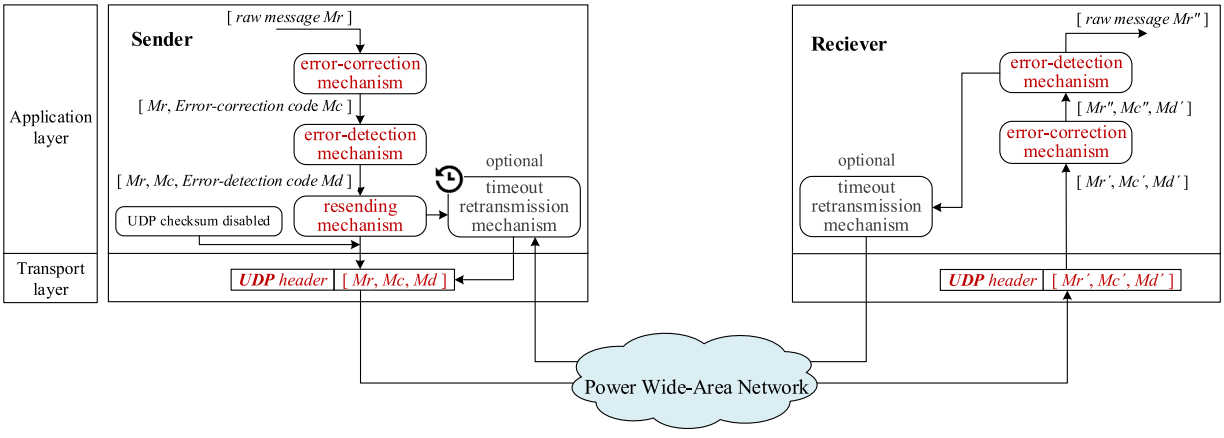


Fig. 1. Overview of UDP-RT scheme.

Density Parity Check (LDPC) codes and TPCs, from the decoding performance and the complexity of encoding and decoding, as shown in Table 1.

As for decoding performance, BCHs can only correct random errors, while Turbo codes, RS codes, LDPCs and TPCs can correct both burst and random errors [35]. Turbo codes outperform LDPCs and TPCs at low code rate, but LDPCs and TPCs have better performance at high code rate [36]. LDPCs and TPCs outperform RS codes at high code rate and small code length (<5000bits) [37–40]. The performance of TPCs comparable to LDPCs of the same code rate [41,42]. Therefore, compared to BCHs, Turbo codes, and RS codes, LDPCs and TPCs achieve better performance while consuming less bandwidth. As for encoding and decoding complexity, Almaamory et al. [43] and Uryvsky et al. [44] verified that Turbo codes and BCHs have a higher complexity compared to LDPCs. Li et al. [41] and Ren et al. [45] verified that a TPC code is with less complexity compared to a LDPC code. According to the encoding and decoding methods of TPCs and RS codes presented in [35, 47], TPCs are with less complexity compared to RS codes. So TPCs have the lowest complexity compared with other four codes.

To summarize, TPCs achieve better performance at high code rate and have lower complexity over other three codes. So TPCs are the best for WAPS communications.

3.2.2. Encoding and decoding methods of TPCs

A two-dimension TPC is constructed by two linear block codes set as identical to reduce the complexity of encoding and decoding. Besides, since the TPCs composed of extended hamming codes correct both burst and random errors, extended Hamming code is adopted as the component code. The component code is denoted as C with parameters (n, k, d), where n, k and d stand for code length, number of information symbols and minimum Hamming distance respectively. Then, the encoding method of the TPC $P = C \times C$ (see Fig. 3) is performed as: a) Place $(k \times k)$ information symbols in a matrix of k rows and k columns. b) Encode the k rows using code C, this gives a $k \times n$ matrix. c) Encode the n columns using code C, this gives an $n \times n$ matrix. The parameters of the P are $n_p{=}n \times n$, $k_p{=}k \times k$, $d_p{=}d \times d$, and the code rate $R{=}k_p/n_p$. The P of length n_p contains k_p information symbols and $c_p{=}n_p{-}k_p$ check symbols.

The two categories of TPCs decoding methods are hard decision decoding and soft decision decoding. Hard decision coding, which features low latency, low complexity and easy implementation, operates on data that take on a fixed set of possible values, typically 0 or 1 in a binary sequence. Whereas soft decision coding operates on data that take on a whole range of values in-between, the value typically means the probability of a binary bit being 0 or 1. Considering the message to be decoding by the error correction mechanism is a binary string after UDP decapsulation, hard-decision decoding is the only choice. The specific steps of hard-decision decoding are introduced in [35]. By using hard decision decoding, a TPC code can correct x random errors as well as one burst error whose length is less than y bits, where:$x = \lfloor (d_p - 1)/2 \rfloor$ and $y \leq n\lfloor .(d-1)/2 \rfloor$.

3.2.3. TPCs selection

The common TPCs include TPC(32,26,4)×(32,26,4), TPC(64,57,4)×(64,57,4) and TPC(128,120,4)×(128,120,4), and their length of information symbols are 84bytes, 406bytes and 1800bytes respectively.

Table 1
Comparison of common error correction codes.

	Burst and random error correction capability	Performance at high rate	Complexity
Turbo codes	random and burst errors	low	high
BCHs	random errors	low	middle
RS codes	random and burst errors	middle	middle
LDPCs	random and burst errors	high	middle
TPCs	random and burst errors	high	low

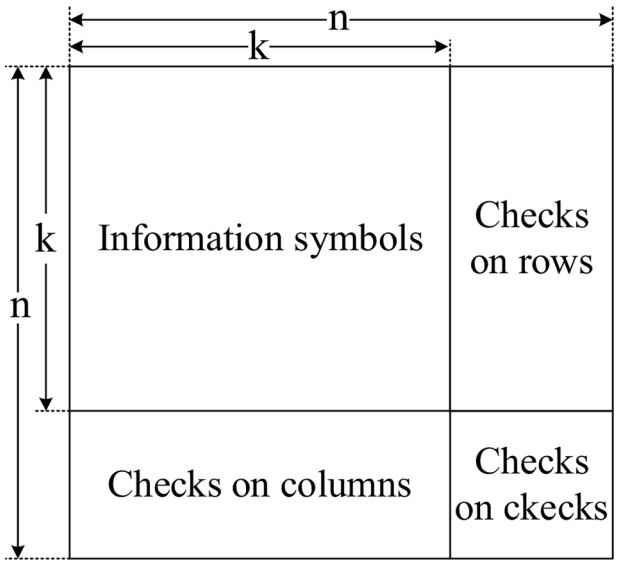


Fig. 3. TPC matrix structure.

Among them, only the information symbols length of TPC(128,120,4)×(128,120,4) is larger than the length of all application layer messages (1~1472bytes). However, the messages in WAPS are mostly around 250bytes [31,32] which is much shorter than 1800bytes. If TPC(128,120,4)×(128,120,4) is used for error correction, lots of padded information symbols would consume the resources of terminals and networks. Therefore, it is not appropriate for WAPS due to its low channel utility. The other two TPCs cannot directly encode and decode the message whose length is larger than that of these two TPCs' information symbols. Our solution is to divide a message into several blocks when the message length is larger than the length of the TPC's information symbols. Each block is encoded to get the check symbols, and the check symbols of all blocks are combined as a check code. Then, the message and the check code are packeted to send to the receiver.

The TPCs should have good performance in high code rate, and the error correction capability should higher than the errors occurred during the transmission of a message over fiber channels. TPC(32,26,4)×(32,26,4) has better decoding performance than TPC (64,57,4)×(64,57,4) based on (1) and (2) when processing the messages with same length. However, TPC(32,26,4)×(32,26,4) has much lower code rate than TPC(64,57,4)×(64,57,4). In addition, although the decoding performance of TPC (64,57,4)×(64,57,4) is slightly lower, it is still much higher than the errors occurred during the transmission of a message over fiber channels. Therefore, TPC(64,57,4)×(64,57,4) is more appropriate for encoding and decoding the messages in blocks.

3.2.4. Blocking rule for TPCs

As we disused in the previous section, a message ***Mr*** should be divided into several blocks for encoding or decoding these blocks respectively. The check symbols of all blocks will be combined as a check code ***Mc***, and ***Mr*** and ***Mc*** are packeted to send to the receiver. Therefore, a blocking rule is required to calculate the blocks number of messages with different lengths. The blocking rule for TPCs is as shown in Table 2, which considers the following three points.

(1) To ensure the reliability of message transmission, the error correction capability of the error correction mechanism must be greater than or equal to the number of error bits that occurred during the transmission of a message over fiber channels. The number of possible error bits in a message, i.e., the number of error bits to be corrected in a message, can be calculated by the message length and the channel bit error rate. Furthermore, to make sure the error correction mechanism can correct errors that

Table 2
Blocking rule for TPCs.

Length of application layer message ***Mc*** (bytes)	Number of blocks	Length of check code ***Mc*** (bytes)	Error correction capability		Number of error bits to be corrected (bits)
			Number of correctable random errors	Number of correctable burst errors	
1–406	1	106	7[1]	1[2]	1
407–812	2	212	14	2	2
813–1152	3	318	21	3	3

[1] Seven random errors can be corrected when the message length is less than 406 bytes. The term random error is a single-bit-error, which means that only one bit of given message is changed from 1 to 0 or 0 to 1.

[2] One burst error whose length is less than 64bits can be corrected when the message length is less than 406 bytes. The term burst error means that two or more bits in the message have changed from 0 to 1. The length of the burst error is measured from the first corrupted bit to the last corrupted bit. Some bits in between may not be corrupted.

occurred during the transmission of a message through all types of SDH paths, we choose the channel bit error rate of the VC-12 path, which has the worst error performance [46], as the basis of calculating the number of error bits required to correct in a message. According to GB/T 13,619–2009, the severely errored second ratio (SESR) objective of VC-12 path corresponds to the equivalent bit error rate of 2.52×10^{-4}.

(2) TPC(64,57,4)×(64,57,4) is selected for block coding in Section 3.2.3. Its code length n_p=4096bits, number of information symbols k_p=3249bits, and number of check symbols r_p=847bits. According to (1) and (2), it can correct seven random errors as well as one burst error whose length is less than 64bits.

(3) After a message is divided into several blocks, we need to calculate whether the length of each block is less than k_p. If so, padding the length of each block to k_p bits using zero bits. Then, each block is encoded to generate r_p check symbols, but r_p is indivisible by 8. To facilitate checksum calculation, we pad the length of check symbols to 848bits (106bytes) using one zero bit.

3.2.5. Encoding and decoding processes of a message

Based on the encoding method of TPCs and the blocking rule, the encoding process of a raw application layer message ***Mr*** consists of the following steps.

Step 1: Split ***Mr*** into M blocks based on Table 2. Each block is denoted as $\boldsymbol{m}_i$, the length of $\boldsymbol{m}_i$ is l_i, where $i = 1, 2...M$.

Step 2: Append (k_p-l_i) zero bits to the end of each $\boldsymbol{m}_i$.

Step 3: Encode each $\boldsymbol{m}_i$ with the encoding method of TPCs to get check code $\boldsymbol{c}_i$ of $\boldsymbol{m}_i$.

Step 4: Append one zero bit to the end of each $\boldsymbol{c}_i$, and combine all $\boldsymbol{c}_i$ as check code ***Mc***.

Step 5: Combine ***Mr*** and ***Mc*** to get data [***Mr, Mc***].

Based on the decoding method of TPCs and the blocking rule, the decoding process of the error data [***Mr***', ***Mc***'] consists of the following steps.

Step 1: Split ***Mr***' into M blocks based on Table 2. Each block is denoted as $\boldsymbol{m}_i$', the length of each $\boldsymbol{m}_i$' is l_i.

Step 2: Append (k_p-l_i) zero bits to the end of each $\boldsymbol{m}_i$'.

Step 3: Split ***Mc***' into M parts, and each part is denoted as $\boldsymbol{c}_i$'. Get the first c_p bits from each $\boldsymbol{c}_i$' as $\boldsymbol{cl}_i$'.

Step 4: Decode each $\boldsymbol{m}_i$' using the decoding method of TPCs and $\boldsymbol{cl}_i$' to get the decoded block $\boldsymbol{m}_i$'' and decoded check code $\boldsymbol{c}_i$''.

Step 5: Get the first l_i bits from each $\boldsymbol{m}_i$'' as $\boldsymbol{ml}_i$'', and combine all $\boldsymbol{ml}_i$'' as decoded message ***Mr***''.

Step 6: Append one zero bits to the end of each $\boldsymbol{c}_i$'' and combine all $\boldsymbol{c}_i$'' as corrected check code ***Mc***''.

3.2.6. Complexity of the encoding and decoding processes

The error correction mechanism employs TPCs to encode and decode a message in blocks, so the complexity of this mechanism is related to the number M ($1 \le M \le 3$) of blocks and the complexity of TPCs. Moreover, based on the encoding and decoding method of TPCs, the component code C is used $(n + k)$ times to encode or decode the rows and columns of the TPC matrix, so the complexity of TPCs is decided by the usage count and complexity of C. As mentioned in Section 3.2.2, extended hamming code is adopted as C. The encoding and decoding complexity of extended hamming codes are $O(k(n\text{-}k))$ and $O(n(n\text{-}k))$, respectively [47].

Therefore, the encoding and decoding complexity of the error correction mechanism can be calculated as $O(Mk(n_p\text{-}k_p))=O(n_p^{\frac{3}{2}})$ and $O(Mn(n_p\text{-}k_p)) = O(n_p^{\frac{3}{2}})$, respectively. It indicates that this mechanism has a low computational cost and negligible impacts on the performance of applications.

3.3. Error detection mechanism

The errors in a message might exceed the capability of the error correction mechanism. So we design an error detection mechanism to identify the correctness of the decoded message. The decoded message will be dropped if any error is detected in it. The object of error detection includes the raw message and the check code. Denote F as the algorithm of generating error detection codes, then the encoding and decoding methods of the error detection mechanism are as shown in (3) and (4) respectively. If ***Md*** is the same as ***Md***'', the receiver considers that the error correction mechanism has corrected all errors in the message. If not, the receiver judges the error correction fails.

$$\boldsymbol{Md} = F(\boldsymbol{Mr}, \boldsymbol{Mc}) \tag{3}$$

Where ***Md*** is the error detection code calculated by the sender, ***Mr*** is the raw message, ***Mc*** is the check code.

$$\boldsymbol{Md}'' = F(\boldsymbol{Mr}'', \boldsymbol{Mc}'') \tag{4}$$

Where ***Md***'' is the error detection code calculated by receiver, ***Mr***'' is the decoded message, ***Mc***'' is the decoded check code.

We suggest that F uses the UDP checksum calculation algorithm to generate the error detection code. Hence, the length of the error detection codes is 2 bytes.

3.4. Resending mechanism

The error correction and error detection mechanisms are for the messages that have arrived at the receiver. Messages might loss or be dropped by receiver due to errors correction fail. Therefore, we further present the resending mechanism of sending the message $\boldsymbol{M} = [\boldsymbol{Mr}, \boldsymbol{Mc}, \boldsymbol{Md}]$ N times in rapid succession. More specifically, as soon as a message is sent out, the sender immediately resends the same message (N-1) times. The resending interval between messages is limited by the processing and sending capacity of the sender. It is worth noting that the proposed resending mechanism is different with TCP's retransmission. In TCP retransmission, an automatic timer is initialized when the sender sends a message, and its value is usually lager than the round-trip latency and increases when the network congesting. If there is no acknowledgement from the receiver after the timer expires, the message will be retransmitted. When network congesting, multiple times retransmission may not ensure a timely delivery. However, in our proposed mechanism, a message is sent N times in succession without waiting for the timer timeout. The resending interval between messages is much shorter than that of TCP retransmission and irrelevant with the network situation. Therefore, multiple times resending can still ensure the real-time performance of messages when network congestion.

Compared with Internet, the power communication network is a

private network, whose communication volume is much smaller than the channel capacity and is predictable. Although several times resending increases the network traffic, the resending count can be optional based on the channel capacity. It is also possible to take the bandwidth demand increased by message resending into account during network planning.

3.4.1. Performance evaluation

The impact of the resending mechanism on the networks is evaluated through simulation. The simulation is performed using OPNET, and a provincial power dispatching network, whose specific structure is shown in Fig. 4, is established. The network is composed of a provincial dispatching center and ten regional dispatching centers, and its backbone network is a dual homing network with four 155/1000 M ring shaped chains. Each regional dispatching center has connected a PMU, which is set to send measurements to the control center using UDP-RT. As this simulation mainly discuss the impact of the resending mechanism on the networks, the mechanisms of error correction, error detection and timeout retransmission are not implemented. The sending frequency is set as 50 Hz, and the application layer message length of the measurements is set as 256bytes. Further, we connect multiple attacking hosts to the switch 1 whose backplane bandwidth is set as 1000 Mbps. The overall throughput of malicious traffics is set as 1240 Mbps.

In the simulation, the resending count N is respectively set as 2, 3, 4, 5. The results of the packet loss rate and average end-to-end delay of the whole network under these four resending count are presented in Fig. 5 and Fig. 6, respectively. The results show that the packet loss rate and average end-to-end delay barely affected with the increase of resending count under congestion status. That is, the proposed resending mechanism would not increase the network congestion and influence the real-time performance of messages.

3.5. Algorithms of UDP-RT

The algorithms for UDP-RT applications contain Algorithms 1 and 2. Algorithms 1 and 2 must be deployed at both ends of the communication. The UDP checksum is disabled in algorithm 1 by setting the UDP checksum field to zero. This does not require modification of UDP protocol because it can be implemented though the application programming. The timeout retransmission mechanism is not included in the proposed algorithms.

4. Validation

In this section, Algorithms 1 and 2 are programmed for experiment to validate the correctness of UDP-RT scheme.

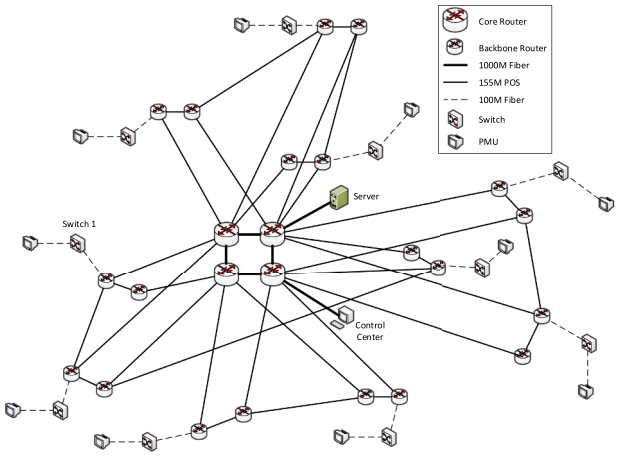


Fig. 4. The structure of a provincial power dispatch data network.

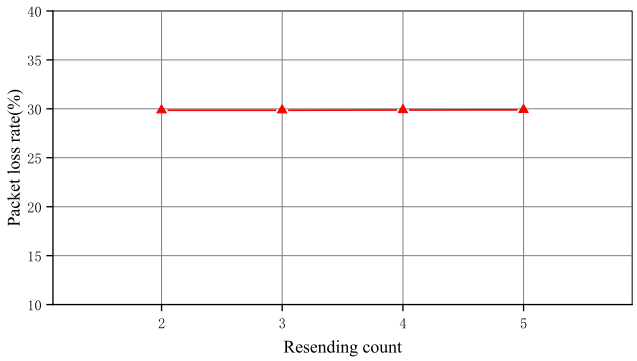


Fig. 5. Packet loss ratio of the whole network with different resending count.

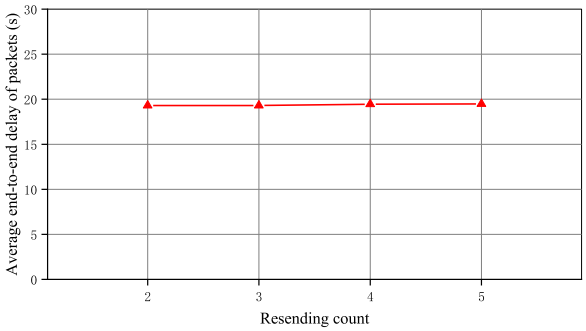


Fig. 6. Average end-to-end delay of the whole network with different resending count.

Algorithm 1
Message processing at sender.

Input: raw message $\boldsymbol{Mr}$ ($\boldsymbol{Mr} \leq$ 1152bytes) of the application layer
Output: UDP packet $\boldsymbol{Pkt}$
$L = Length(\boldsymbol{Mr})$ // get the length of an array.
Divide $\boldsymbol{Mr}$ into M blocks according to Table II, and denote each block as $\boldsymbol{m}_i$ ($i = 1, 2, \ldots, M$).
for $i = 1, 2, \ldots, M$ **do**
  $l_i = Length(\boldsymbol{m}_i)$.
  if $l_i < k_p$ **then** $\boldsymbol{m}_i = \boldsymbol{m}_i \times 2^{k_p - l_i}$.
  Check data $\boldsymbol{c}_i = Encode(\boldsymbol{m}_i)$ // encode $\boldsymbol{m}_i$ by the encoding method of TPCs to get $\boldsymbol{c}_i$.
  $\boldsymbol{c}_i = \boldsymbol{c}_i \times 2^1$.
end for
Check code $\boldsymbol{Mc} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_M]$.
Error detection code $\boldsymbol{Md} = F(\boldsymbol{Mr}, \boldsymbol{Mc})$.
Application layer message $\boldsymbol{M} = [\boldsymbol{Mr}, \boldsymbol{Mc}, \boldsymbol{Md}]$.
UDP packet $\boldsymbol{Pkt} = UDP\_Encapsulation(\boldsymbol{M})$.
SetSocketOption(NoChecksum, 1). // set the UDP checksum field to zero.
for $t = 1, 2, \ldots, N$ **do** *Sendto*($\boldsymbol{Pkt}$).

4.1. Experimental programming

We use two virtual machines with the Linux system as the sender and receiver, then enable them to communicate with each other. The sender and receiver run the sending and receiving procedures respectively. The flowcharts of these two procedures are depicted in Fig. 7. They are programmed based on Algorithms 1 and 2, respectively, but the following two points are added to the sending procedure on the basis of Algorithm 1 for experimental needs.

(1) The sending procedure is set to manually enter the desired message length and then randomly generate a raw message.

Algorithm 2
Message processing at receiver.

Input: UDP packet ***Pkt***' received by the receiver
Output: corrected message ***Mr***''
1: [***Mr***', ***Mc***', ***Md***'] = *UDP_Decapsulation*(***Pkt***').
2: L = *Length*(***Mr***').
3: Divide ***Mr***' into M blocks according to Table II, and denote each block as $\boldsymbol{m}_i$' (i = 1, 2, …, M).
4: Divide ***Mc***' into M parts, and denote each part as $\boldsymbol{c}_i$' (i = 1, 2, …, M).
5: **for** i = 1, 2, …, M **do**
6: l_i = *Length*($\boldsymbol{m}_i$').
7: **if** $l_L < k_p$ **then** $\boldsymbol{m}_i$' = $\boldsymbol{m}_i$' × $2^{k_p - l_i}$.
8: $\boldsymbol{cl}_i$' = $\boldsymbol{c}_i$'.*Slice*(0, c_p-1) // slice $\boldsymbol{c}_i$' start at index 0 and end at index (c_p−1) to get the subarray $\boldsymbol{cl}_i$'.
9: [decoded block $\boldsymbol{m}_i$'', decoded check data $\boldsymbol{c}_i$''] = *Decode*($\boldsymbol{m}_i$', $\boldsymbol{c}_i$').
10: $\boldsymbol{ml}_i$'' = $\boldsymbol{m}_i$''.*Slice*(0, l_i-1).
11: $\boldsymbol{c}_i$'' = $\boldsymbol{c}_i$'' × 2^1.
12: **end for**
13: Corrected message ***Mr***'' = [$\mathbf{ml}_1$'', $\mathbf{ml}_2$'', …, $\boldsymbol{ml}_M$''].
14: Corrected check code ***Mc***'' = [$\boldsymbol{c}_1$'', $\boldsymbol{c}_2$'', …, $\boldsymbol{c}_M$''].
15: Error detection code ***Md***'' = F (***Mr***'', ***Mc***'').
16: **if** ***Md***' == ***Md***'' **then** submit ***Mr***'' to the applications and notify the protocol stack does not process any duplicate ***Pkt***'.
17: **else** *Drop*(***Pkt***').
18: **end if**

(2) Since little possibility that errors occur during message transmitted at an experimental environment, three options of a_1 (not add error), a_2 (add errors within the error correction capability) and a_3 (add errors beyond the error correction capability) are set in the sending procedure to determine whether the message is needed to add errors before being sent. When choosing option a_1, the message is sent without modification. When selecting option a_2 or a_3, the number of errors to be generated in the message is manually entered, and errors with the specified number are added to the message.

4.2. Experiment and result

To verify the error correction and detection capability of Algorithms 1 and 2 under different number of errors, in the experiment, we respectively choose options a_1, a_2 and a_3 with the same message length to obtain the results of these three cases. When the length of the raw message is set as 256byes, the experimental results are shown as follows:

(1) When choosing option a_1, the sending procedure sends the message without modification. The receiving procedure does not discover any error in the received message through error correction and detection, so it displays the corrected message. The result indicates that the receiver can receive and display the correct message when no error occurs through transmission.
(2) Fig. 11 in Appendix A presents the results of selecting option a_2, which demonstrates that the receiver can correct all transmission errors and display the corrected message when the number of errors is within the error correction capability of the error correction mechanism.
(3) Fig. 12 in Appendix A shows the results of choosing option a_3, which represents that the receiver cannot correct all transmission errors and drop the error message when the number of errors is beyond the error correction capability.

We also repeat the above experimental process under different message lengths to verify the correctness of the blocking rule, and the results indicate that the procedures can correctly divide the message into blocks and respectively encode or decode these blocks.

In conclusion, the experimental results are consistent with expectations, which show that Algorithms 1 and 2 and their implementation codes are all corrected.

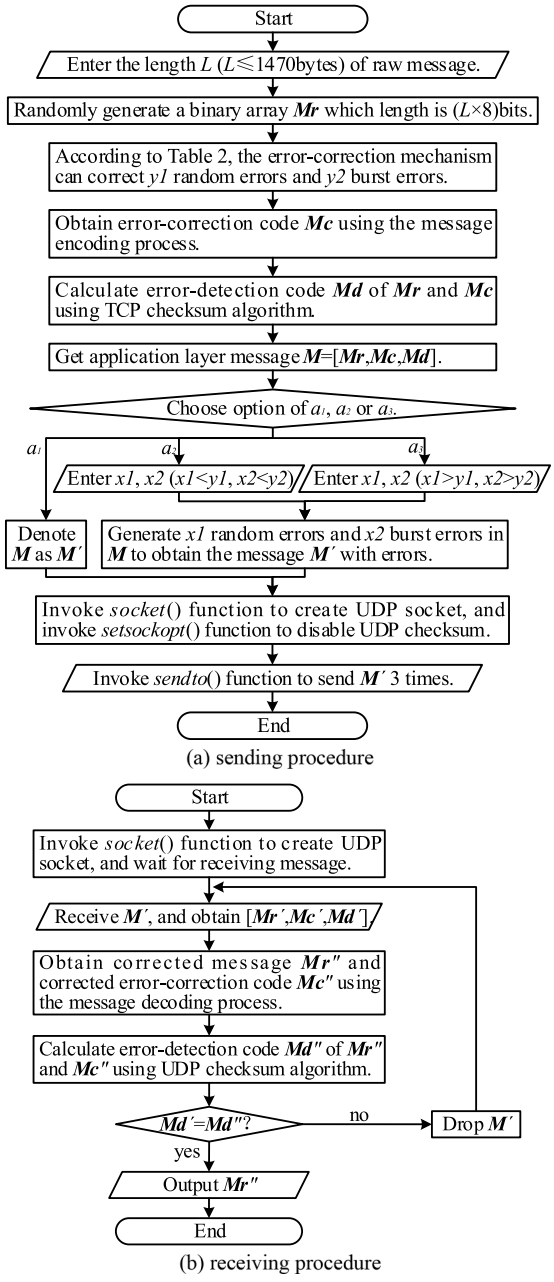


Fig. 7. Flowcharts of experimental programs.

5. Real-time performance analysis

In order to realize diverse wide-area protection functions, the control center needs to acquire various real-time data of power systems from PMUs, detect wide-area disturbances in time by analysing and processing these data, and send the corresponding control instructions or setting values to EUs. The time required to complete the above process is called the response time of wide-area protection. Among various wide-area protection functions, the power angle stability protection is deemed as the most rigorous test on the response time of WAPS. The consequence of power angle stability loss is a system out-of-step, and one of the corresponding protection measures is out-of-step relaying [48]. Therefore, in this section, to evaluate the real-time performance of UDP-RT scheme, we calculate the response delays of an out-of-step relaying under UDP-RT scheme and the TCP transmission scheme respectively and compare them.

5.1. Response delay of wide-area protection

The response delay T of a wide-area protection is composed as follows:

$$T = T_A + T_S + T_{CU} + T_D + T_{CI} \tag{5}$$

where T_A is the measurements acquisition delay, T_S is the sampling interval delay, T_{CU} is the communication delay of measurements uploading, $T_D$is the decision-making delay, T_{CI} is the communication delay of control instructions issuing.

T_A and T_D are determined by the performance and operation mode of the device. T_S is determined by the frequency of data acquisition. T_{CU} and T_{CI} are related to the communication delay of the network and determined by the transmission layer protocol, network state and network structure. We assume $T_{CU} = T_{CI}$for the convenience of calculation. As for a determined network, the value of the communication delay may vary with the network load and the adopted transport layer protocol, whereas the other delays remain relatively constant. Therefore, the communication delays under UDP-RT and the TCP transmission scheme are further analysed and defined as T_{UDP-RT} and T_{TCP}, respectively. Then, Eq. (5) can be rewritten as:

$$T = \begin{cases} T_A + T_S + T_D + 2T_{UDP-RT} & \text{under UDP} - \text{RT scheme} \\ T_A + T_S + T_D + 2T_{TCP} & \text{under TCP scheme} \end{cases} \tag{6}$$

5.1.1. Communication delay under UDP-RT scheme

1) *Without the timeout retransmission mechanism:* In UDP-RT scheme, a message is sent N times in succession without waiting for confirmation from the receiver. The sending interval between messages is only limited by the processing and sending capacity of the sender. In the worst situation, the first (N-1) messages get lost through transmission, and just the last message arrives at the receiver. Under this situation, the communication delay refers to the time duration that begins as the sender starts to encode the first message and ends as the receiver finishes decoding the last message, which includes the following three parts, as shown in Fig. 8.
   a) reliability operations delay (T_a) is the computing time required for the error correction and error detection mechanism.
   b) Sending delay (T_s) of the first (N-1) messages. T_s is the time that the sender sends all bits of a message into the wire.
   c) Inter-frame space delay (T_i) between the N frames. T_i is the waiting period between transmission of frames.
   d) End-to-end delay (T_{eu}) of the last message, including T_s, link transmission delay (T_l), receiving delay (T_r). T_l is the amount of the propagation delay on the links and the processing delay in forwarding nodes from source to destination. T_r is the time that the receiver receives all bits of a message from the wire.

According to the above analysis, the communication delay $T_{UDP-RT1}$ under UDP-RT scheme is as follows:

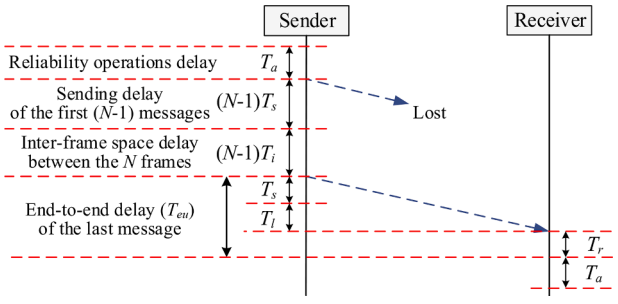


Fig. 8. Composition of communication delay under UDP-RT scheme.

$$T_{UDP-RT1} = T_a + (N-1)(T_s + T_i) + T_{eu} \tag{7}$$

1) *With the timeout retransmission mechanism:* According to Figs. 2 and 8, the communication delay $T_{UDP-RT2}$ under UDP-RT scheme is as follows:

$$T_{UDP-RT2} = T_a + (M+1)(N-1)(T_s + T_i) + MT_{rto} + T_{eu} \tag{8}$$

Where M is the timeout retransmission count, T_{rto} is the retransmission timeout delay.

5.1.2. Communication delay under TCP scheme

In the TCP transmission scheme, an automatic timer is initialized when the sender sends a message. If there is no acknowledgement from the receiver before the timer expires, the message needs to be retransmitted. Still assume the worst situation, the message has been retransmitted (N-1) time, i.e., the message has been sent N times. Under this situation, the communication delay includes the three parts:

a) Sending delay (T_s) of the first (N-1) messages.
b) Retransmission timeout delay (T_{rto}) between the N messages.
c) End-to-end delay (T_{et}) of the last message.

Thus, the communication delay T_{TCP} under the TCP transmission scheme is as follows:

$$T_{TCP} = (N-1)(T_s + T_{rto}) + T_{et} \tag{9}$$

The end-to-end delays T_{eu} and T_{et} of the above two schemes may be fluctuated by network status, which is difficult to calculate theoretically. Therefore, in this paper, method of combining theoretical calculations with simulations is adopted to analyze the communication delays T_{UDP-RT} andT_{TCP}. That is, T_{eu} and T_{et} are obtained by simulation in the software and the other delays are calculated through theoretical analysis.

5.2. Calculation of the optimal resending count N

In the case of UDP-RT with the timeout retransmission mechanism, if all the N resent messages get lost and the retransmitted message exceeds its real-time requirement, the message is used for alarming or accident tracing. If all the N messages get lost and the retransmitted message meets its real-time requirement, the receiver will operate as the message required. However, if the resending mechanism can ensure the possibility of all the N resent messages get lost is less than 0.1 % when the retransmitted message meets its real-time requirements, the timeout retransmission mechanism can be optional (The communication system reliability requirement of wide-area protection and control applications should exceed 99.9 % [49,50]). Therefore, the value of N becomes essential, which determines the possibility of all the N messages get lost under a certain degree of congestion.

To determine the optimal resending count N, we first calculate the end-to-end delay T_{eu} of the retransmitted message when it reach its maximum allowable communication delay. According to [8], the response time of an out-of-step relaying $T \leq 500$ ms, and the maximum allowable communication delay of measurements uploading and control instructions issuing is 370 ms, that is $T_{CU} + T_{CI} \leq 370$ ms. Then according to (6) and (8), $T_{UDP-RT2} = T_a + (M+1)(N-1)(T_s + T_i) + MT_{rto} + T_{eu} \leq$ 185 ms. We assume timeout retransmission count $M = 1$; To simplify the calculation, we set $T_{rto} = 3T_{eu}$ and $T_s + T_i$=0; According to Section 5.4.1, T_a =22.64 ms. Hence, $T_{eu} \leq$ 40.59 ms. That is, when T_{eu}=40.59 ms, the retransmitted message reaches its maximum allowable communication delay.

Then, through simulation, the network congestion status and the

message loss rate can be obtained when the end-to-end delay of the messages is 40.59 ms. In the simulation, the adopted network and parameter settings are similar to those in Section 3.4. The difference is that PMUs are set to send measurements to the control center using UDP. We gradually increase the throughput of malicious traffics to cause congestion on switch 1. When the maximum end-to-end delay of the messages reaches 40.59 ms, we get the throughput of malicious traffics is 1002 Mbps and the messages loss rate P of the measurements is 4.53 % ($P = N_l/N_t \times 100\%$, where N_l is the number of lost messages and N_t is the number of transmitted messages).

As we had verified in Section 3.4.1 that the resending count does not increase the network congestion, the possibility of all the N resent messages get lost can be calculated as $P_{\text{raw}} = (P)^N = (4.53\%)^N$. When $N = 3$, P_{raw}=0.0093 %<0.1 %, but when $N = 2$, P_{raw}=0.2052 %>0.1 %. So the optimal resending count N is set as 3.

Therefore, when slightly congested network condition, 3 times resending can ensure the possibility of all the 3 resent messages get lost approaches to zero. Further, when heavily congested network condition and all the 3 messages get lost, the retransmitted message may not arrive timely and can only be used for alarming or accident tracing. That is, when $N = 3$, the timeout retransmission mechanism can be optional. In the following, we consider UDP-RT without this mechanism.

5.3. Simulation for end-to-end delay

T_{eu} in (7) and T_{et}in (9) are obtained through simulation. The adopted network and the parameter settings are similar to those in Section 3.4. The length of the data field in a UDP packet is set as (256+106)= 362bytes, where 106bytes is the length of ***Mc*** obtained according to Table 2, and the length of the data field in a TCP packet is set as 256bytes.

We respectively transmit the measurements with UDP-RT and TCP transmission modes to collect the end-to-end delay. The simulation results are presented in Fig. 9, which shows that the end-to-end delays under UDP-RT and TCP are approximate when the network is uncongested. T^1_{eu} and T^1_{et}, the maximum value during simulation, are about 1.33822 ms and 1.34271 ms respectively.

Further, we connect multiple attacking hosts to switch 1, and the throughputs of malicious traffics are respectively set as 1010 Mbps, 1080 Mbps, 1160 Mbps and 1240 Mbps in the simulation. The results under these four types of congestion levels are presented in Fig. 10. It shows that the end-to-end delays under TCP increase sharply with the higher congestion degree, T^2_{et} is between 1 s and 48 s. Whereas the end-to-end delays under UDP-RT is barely affected by malicious traffics, and T^2_{eu} is around 76 ms.

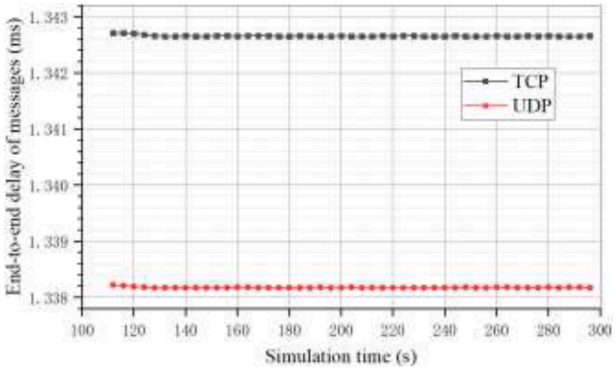


Fig. 9. Maximum end-to-end delays using TCP and UDP-RT when the network is uncongested.

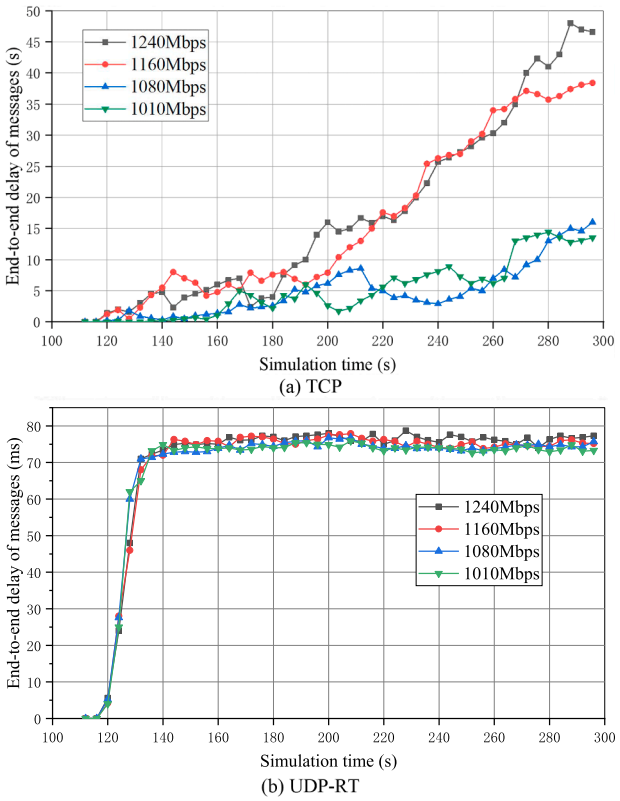


Fig. 10. Maximum end-to-end delay using TCP and UDP-RT in four types of congestion traffics.

5.4. Calculation of communication delays

5.4.1. Communication delay of UDP-RT scheme

The value of each delay in T_{UDP} except T_{eu} are calculated below.

T_a includes the time taken for checksum calculation of error detection and encoding and decoding of error correction. The time of checksum calculation is negligible due to its simple calculation way. In addition, the proposed scheme should disable UDP checksum validation in practical applications, which reduces the calculation time in the transmission layer. So the checksum calculation in the application layer won't increase the latency, and we only need to get the encoding and decoding time of error correction to calculate T_a. To measure the encoding and decoding times, we run the implementation codes in Section 4.1 twenty times and measure these times in each trial. The average values of the measured encoding and decoding times are 12.467 ms and 10.168 ms respectively. Considering the message encoding and decoding operations are performed only once at source and destination respectively, T_a = (12.467+10.168) = 22.64 ms.

T_s is the delay caused by the data-rate of the link, and proportional to the message's length in bits. According to the above simulation, the bandwidth of the link connected to the receiver is 100 Mbps, and the length of an application-layer message is 256bytes. So the length of a link-layer frame under UDP-RT scheme is (256+106+2 + 8 + 20+26)= 418bytes. Where, 106bytes is the length of the check code, 2bytes is the length of the error detection code, 8bytes is the length of the UDP header, 20bytes is the length of the fixed IP header (to simplify the calculation, we ignore the optional field in the IP header), and 26bytes is the total length of the preamble, start frame delimiter (SFD), destination MAC address, source MAC address, ether type and frame check sequence (FCS) field in the frame. Thus, T_s = (418×8)bits / 100Mbps=33.44us.

T_i is depend on the transmission rate of Ethernet LANs. So when the transmission rate is 100 Mbps, T_i=0.96us.

Hence, based on (7), $T^1_{UDP-RT} = T_a + 2(T_s + T_i) + T^1_{eu}$ = 24.06 ms

when the network is uncongested, and T^2_{UDP-RT}=$T_a + 2(T_s + T_i) + T^2_{eu}$ = 99.70 ms when the network is congested.

5.4.2. Communication delay of TCP scheme

The link bandwidth and the length of an application-layer message are set as those in UDP-RT scheme. So the length of a link-layer frame under TCP transmission scheme is (256+20+20+26)=322bytes. Where, the first 20bytes is the length of the fixed TCP header, the second 20bytes is the length of the fixed IP header (the optional fields of the TCP header and IP header are ignored), and 26bytes is the total length of the preamble, SFD, destination MAC address, source MAC address, ether type and FCS field in the frame. Thus, T_s= (322×8)bits / 100Mbps=25.76us. Moreover, T_{rto} is usually slightly larger than the average round-trip time (RTT) of the messages, i.e., $T_{rto} > 2T_{et}$. To simplify the calculation, we set $T_{rto} = 3T_{et}$.

Hence, according to (9), $T^1_{TCP} = 2(T_s + T_{rto}) + T^1_{et}$=9.45 ms when the network is uncongested, while $T^2_{TCP} = 2(T_s + T_{rto}) + T^2_{et}$ is between 7 s and 306 s when the network is congested.

5.5. Calculation of response delays

Considering the worst situation, the system can keep reliability and stability only if the out-of-step relaying acts within 500 ms after power angle stability loss begins [8]. In other words, the response time of an out-of-step relaying $T \leq 500$ ms. The control center needs at least 2 phasor samples to detect power angle swing, and the sampling rate of PMUs is about 25 Hz, so the sampling interval delay T_S=40 ms. Thus, the time for PMUs to collect sampling data cannot exceed 40 ms. To simplify the calculation, we set T_A=40 ms. Furthermore, if the power angle swing is detected, the control center needs to estimate whether the swing is stable and issue the corresponding control instruction. The time of decision-making is about 50 ms, that is T_D=50 ms. Hence, according to (5), the maximum allowable communication delay of measurements uploading and control instructions issuing is 370 ms, that is $T_{CU} + T_{CI} \leq$370 ms.

Therefore, under UDP-RT scheme, according to (6), $T^1 = T_A + T_S + T_D + 2T^1_{UDP-RT}$=178 ms when the network is uncongested, and $T^2 = T_A + T_S + T_D + 2T^2_{UDP-RT}$=327 ms when the network is congested. Where, both T^1 and T^2 are far less than the maximum allowable delay of out-of-step relaying. While under the TCP transmission scheme, $T^3 = T_A + T_S + T_D + 2T^1_{TCP}$=149 ms when the network is uncongested, and $T^4 = T_A + T_S + T_D + 2T^2_{TCP}$is between 14 s and 607 s when the network is congested. Where, T^3 is within the stipulated time of out-of-step relaying, but T^4 is much larger than the time.

To summarize, with the same rate of message loss, the real-time performance of UDP-RT scheme is much higher than the TCP transmission scheme. More importantly, UDP-RT scheme can meet the real-time requirements of most wide-area protection functions when network congesting, while the TCP transmission scheme cannot.

6. Reliability analysis

In this section, the reliability performance of UDP-RT scheme, TCP transmission scheme, and the existing UDP-based transmission schemes are analysed and compared. The results are as in Table 3, which shows that the proposed scheme has the highest reliability as well as the highest real-time performance. Three existing UDP-based transmission schemes are discussed in Section 2, where Xiao et al. scheme [26] is just for measurements. Hence, IEC61850–90–5 scheme [23] and Fan et al. scheme [25] are selected to compare with our UDP-RT.

A reliable transmission scheme can deliver the message to the receiver without error, so the reliability of a transmission scheme can be reflected by correct rate and arrival rate. We define *correct rate* as the probability that all errors in the message are corrected and detected, and *arrival rate* as the probability that the message arrives at the receiver. As the real communication environment of WAPS is complicated and unpredictable, errors and message loss may occur with the interference of noise and congestion. The error rate and the loss rate of a message are set as P_e and P_d, respectively.

UDP-RT scheme ensures the correct rate by the error correction and error detection mechanisms and ensures the arrival rate by the resending mechanism. In this analysis, the retransmission timeout mechanism is not included. When resending a message N times in succession, the reliability probability P_{UDP-RT} of UDP-RT scheme is shown as:

$$P_{UDP-RT} = 1 - \left[P_e P_1 P_2 \left(1 - P^1_d\right) + P^1_d\right]^N \tag{10}$$

Where P_1 is the probability of the error detection mechanism being unreliable, P_2 is the probability of the error correction mechanism being unreliable, P^1_d is the message loss rate under UDP-RT.

TCP transmission scheme ensures the correct rate by TCP checksum and ensures the arrival rate by TCP's timeout retransmission mechanism. When a message has been retransmitted (N-1) times, the reliability probability P_{TCP} of the TCP transmission scheme is shown as:

$$P_{TCP} = 1 - \left[P_e P_1{}' \left(1 - P^2_d\right) + P^2_d\right]^N \tag{11}$$

Where $P_1{}'$ is probability of TCP checksum being unreliable, P^2_d is the message loss rate under TCP.

IEC61850–90–5 transmission scheme [23] and GOOSE over UDP transmission scheme [25] use the same error detection and retransmission methods. These two schemes ensure the correct rate by UDP checksum and ensure the arrival rate by resending the message with a vary interval. When resending a message N times with a vary interval, the reliability probability P_{S1} of these two schemes is shown as:

$$P_{S1} = 1 - \left[P_e P_1{}'' \left(1 - P^3_d\right) + P^3_d\right]^N \tag{12}$$

Where $P_1{}''$ is probability of the UDP checksum being unreliable, P^3_d is the message loss rate under scheme [23,25].

Since the checksum calculation algorithms of these four schemes are the same, $P_1 = P_1{}' = P_1{}'' = 2^{-16}$ [51]; based on [52], $P_2 = 1 \times 10^{-5}$; According to Section 3.2.4, $P_e = 2.52 \times 10^{-4}$; N is set as 3; P^1_d, $P^2_d$and P^3_d are obtained through simulation. The adopted network and parameter settings are similar to those in Section 3.4.1. PMUs are set to transmit messages using UDP-RT, TCP, and scheme [25] respectively, to measure

Table 3
Reliability comparison of UDP-RT scheme with others.

	Error Correction	Error Detection	Retransmission	Real-time	Reliability
UDP-RT	block coding with TPCs	checksum at the application layer	send a message N times in rapid succession and optional timeout retransmission	high	99.606 %
TCP transmission scheme	—	TCP checksum	TCP's timeout retransmission	low	98.263 %
IEC61850–90–5 transmission scheme[23]	—	UDP checksum	resend the message with a vary interval	middle	99.583 %
Fan et al. scheme[25]	—	UDP checksum	resend the message with a vary interval (same as that of IEC61850–90–5)	middle	99.583 %

the message loss rate. When the throughput of malicious traffics is set as 1240 Mbps, we get $P_d^1 = 0.158$ $P_d^2 = 0.259$ and $P_d^3 = 0.161$. Therefore, according to (10)-(12), $P_{UDP-RT} = 99.606\%$, $P_{TCP} = 98.263\%$ and $P_{S1} = 99.583\%$. That is $P_{UDP-RT} > P_{S1} > P_{TCP}$, which shows that UDP-RT scheme has the highest reliability compared to other three schemes.

7. Conclusion

In this paper, we propose a new real-time and reliable communication scheme for WAPS, UDP-RT. UDP-RT employs UDP at the transport layer for low latency and adds the error correction, error detection, resending and timeout resending mechanisms at the application layer for high reliability. The TPC-based error correction mechanism can correct both of random errors and burst errors in the message within its error correction capability. This mechanism has low complexity and negligible impacts on the performance of applications. The error detection mechanism can identify whether all errors in the message are corrected. It uses the algorithm of UDP/TCP checksum, which has extremely low complexity and would not increase the message delay due to the UDP checksum is disabled. The resending mechanism and timeout retransmission mechanism are able to address the issues of message loss and error correction fails. Simulation results reveal that for wide-area protection functions with extremely high real-time requirements, the timeout retransmission mechanism can be optional as nearly all of the retransmitted messages cannot arrive timely when heavily congested condition and the resending mechanism can ensure the possibility of all the resent messages get lost approaches to zero when slightly congested condition. However, the timeout retransmission mechanism is meaningful for wide-area protection functions with lower real-time requirements as the messages may still arrive timely after timeout retransmission.

Our analyses reveal that the proposed scheme can meet the real-time requirements of most WAPS businesses when the network is congested and has higher reliability than TCP transmission scheme and existing UDP transmission schemes. In practical, the traditional real-time and reliability guarantee methods can be applied with UDP-RT to achieve the real-time and reliable transmission of all the businesses. Our work could also be a reference for other industrial control systems.

CRediT authorship contribution statement

Qiuyu Lu: Conceptualization, Methodology, Software, Investigation, Formal analysis. **June Li:** Conceptualization, Funding acquisition, Resources, Supervision. **Kai Yuan:** Visualization, Investigation, Software, Validation. **Kaipei Liu:** Resources, Supervision. **Ming Ni:** Visualization, Writing – original draft. **Jianbo Luo:** Data curation, Writing – review & editing.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests.

June Li reports financial support was provided by The National Natural Science Foundation of China.

Data availability

No data was used for the research described in the article.

Appendix A

This appendix presents the experimental results of choosing options a_2 and a_3 when the length of the raw message is 256 bytes, as shown in Figs. 11 and 12. The random and burst errors added in the messages are marked with red and yellow highlighted respectively, and the errors in the messages after error correction are marked with blue highlighted.

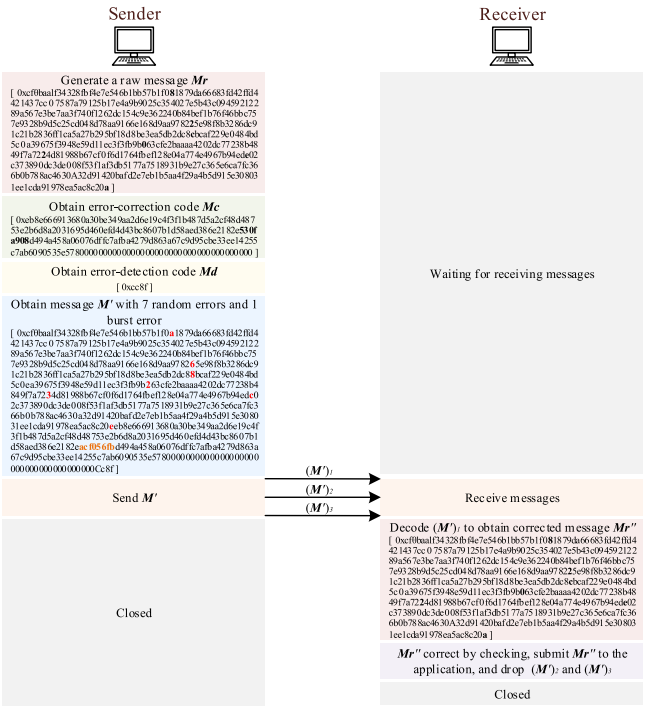


Fig. 11. Results of adding errors within the error correction capability.

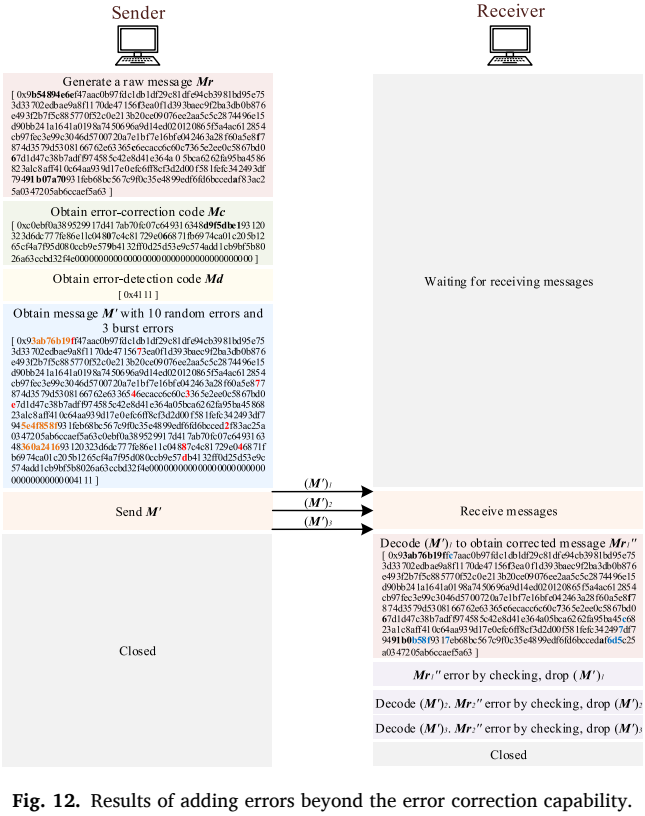


Fig. 12. Results of adding errors beyond the error correction capability.

References

[1] E. Ghahremani, A. Heniche-Oussedik, M. Perron, M. Racine, S. Landry, H. Akremi, A detailed presentation of an innovative local and wide-area special protection

scheme to avoid voltage collapse: from proof of concept to grid implementation, IEEE Trans. Smart Grid 10 (5) (2019) 5196–5211. Sept.
[2] V. Terzija, G. Valverde, D. Cai, P. Regulski, V. Madani, J. Fitch, S. Skok, M. Begovic, A. Phadke, Wide-area monitoring, protection, and control of future electric power networks, Proc. IEEE 99 (1) (2011) 80–93. Jan.
[3] S. Teng, N. Wu, H. Zhu, L. Teng, W. Zhang, SVM-DT-based adaptive and collaborative intrusion detection, IEEE/CAA J. Automat. Sinica 5 (1) (2018) 108–118. Jan.
[4] J. Yan, J. Xu, M. Ni, W. Yu, Impact of communication system interruption on power system wide area protection and control system, Autom. Electric Power Syst. 40 (5) (2016) 17–24. Mar.
[5] N. Liu, J. Zhang, W. Liu, Toward key management for communications of wide area primary and backup protection, IEEE Trans. Power Deliv. 25 (3) (2010) 2030–2032. July.
[6] M. Begovic, D. Novosel, D. Karlsson, C. Henville, G. Michel, Wide-area protection and emergency control, Proc. IEEE 93 (5) (2005) 876–891. May.
[7] *Technical Brochure No. 187, CIGRE. System Protection Schemes in Power Networks*, Jun. 2001.
[8] M.G. Adamiak, A.P. Apostolov, M.M. Begovic, C.F. Henville, K.E. Martin, G. L. Michel, A.G. Phadke, J.S. Thorp, Wide area protection—technology and infrastructures, IEEE Trans. Power Deliv. 21 (2) (2006) 601–609. Apr.
[9] A. Recioui, A. Ouadi, H. Bentarzi, Simulation of data communication in wide area networks employing phasor measurement units, in: 4th International Conference on Power Engineering, Energy and Electrical Drives, Istanbul, Turkey, 2013, pp. 97–102. May.
[10] W. Luo, C. Lin, B. Yan, A survey of congestion control in the Internet, Chin. J. Comput. 24 (1) (2001) 1–17. Jan.
[11] F. Chiariotti, A. Zanella, S. Kucera, K. Fahmi, H. Claussen, The HOP protocol: reliable latency-bounded end-to-end multipath communication, IEEE/ACM Trans. Netw. 29 (5) (2021) 2281–2295. Oct.
[12] H. Zhu, W. Ding, L. Miao, J. Gong, Effect of UDP traffic on TCP's round-trip delay, J. Commun. 34 (1) (2013) 19–29. Jan.
[13] K. Yuan, J. Li, K. Liu, Q. Lu, M. Ni, J. Luo, A reliability communication approach for power wide area protection system based on UDP, Acta Autom. Sinica 47 (7) (2021) 1598–1609. Jul.
[14] N.D. Tuyen, N.S. Quan, V.B. Linh, V. Van Tuyen, G. Fujita, A comprehensive review of cybersecurity in inverter-based smart power system amid the boom of renewable energy, IEEE Access 10 (2022) 35846–35875.
[15] A. Huseinović, S. Mrdović, K. Bicakci, S. Uludag, A survey of denial-of-service attacks and solutions in the smart grid, IEEE Access 8 (2020) 177447–177470.
[16] X. Xiong, J. Tan, X. Lin, Routing algorithm for communication system in wide-area protection based on MPLS, Transactions of China Electrotechnical Society 28 (7) (2013) 257–263. Jun.
[17] N. Xing, S. Zhang, S. Xu, S. Guo, Load balancing-based routing optimization mechanism for power communication networks, China Commun. 13 (8) (2016) 169–176. Aug.
[18] N. Xing, Y. Ji, Research on service controlling algorithm of electric power communication network based on QoS traffic recognition and routing optimization, J. Netw. 9 (11) (2014) 3084–3091. Nov.
[19] X. Dong, X. Wang, Improved weighted fair queueing scheduling algorithm for integrated information transmission in power systems, Proc. Chinese Soc. Electr. Eng. 32 (22) (2012) 149–156. Aug.
[20] J. Gao, W. Tong, X. Jin, Z. Li, L. Lu, Study on communication service strategy for congestion issue in smart substation communication network, IEEE Access 6 (2018) 44934–44943. Aug.
[21] M.H. Yaghmaee, Z. Yousefi, M. Zabihi, S. Alishahi, Quality of service guarantee in smart grid infrastructure communication using traffic classification, in: 22nd International Conference and Exhibition on Electricity Distribution, Stockholm, 2013, pp. 1–4. Jun.
[22] P. Ma, Y. Lu, Y. Hou, L. Li, X. Zhao, L. Zhang, H. Zhu, A multi-service QoS guaranteed scheduling algorithm for TD-LTE 230MHz power wireless private networks, in: 12th International Symposium on Antennas, Propagation and EM Theory, Hangzhou, China, 2018, pp. 1–4. Dec.
[23] *IEC standard for communication networks and systems for power utility automation - Part 90-5: Use of IEC 61850 to transmit synchrophasor information according to IEEE C37.118*, IEC 61850-90-5, Jul. 2012.
[24] T.Y. Wong, W.H. Lau, H.S. Chung, C. Shum, A feasibility study of using manufacturing message specification report gateway model for IEC61850 inter-substation type-1 messaging over Wide Area Network, Sustain. Energy Grids Netw. 30 (2022). Jun.
[25] Y. Fan, Q. Wang, H. Peng, S. Lin, K. Fan, Y. Chen, GOOSE over UDP transmission mechanism for real-time data fast transmission in distribution network, in: Seventh International Green and Sustainable Computing Conference, 2016, pp. 1–5.
[26] X. Xiao, X. Gou, L. Xiao, L. Li, L. Wen, D. Gang, D. Yang, J. Wu, X. Xie, UDP-based small-delay and reliable transfer of WAMS data in LAN, Electric Power Autom. Equip. 31 (10) (2011) 148–152. Oct.
[27] Y. Go, H. Noh, G. Park, H. Song, Hybrid TCP/UDP-based enhanced HTTP adaptive streaming system with multi-homed mobile terminal, IEEE Trans. Veh. Technol. 68 (5) (2019) 5114–5128. May.
[28] A. Mtibaa, C. Good, S. Misra, D.G.M. Mitchell, B. Parikh, RC-UDP: on raptor coding over UDP for reliable high-bandwidth data transport, in: IEEE International Conference on Communications (ICC), 2018, pp. 1–6.
[29] Q. Luo, J. Wang, FRUDP: a reliable data transport protocol for aeronautical Ad Hoc networks, IEEE J. Select. Areas Commun. 36 (2) (2018) 257–267. Feb.
[30] Z. Liu, Y. Jiang, Cross-layer design for UAV-based streaming media transmission, IEEE Trans. Circuits Syst. Video Technol. 32 (7) (2022) 4710–4723. July.
[31] J.W. Stahlhut, T.J. Browne, G.T. Heydt, V. Vittal, Latency viewed as a stochastic process and its impact on wide area power system control signals, IEEE Trans. Power Syst. 23 (1) (2008) 84–91. Feb.
[32] G.Y. Chen, Z. Zhang, X.G. Yin, F. Wang, Wide area backup protection communication mode and its performance evaluation, Proc. CSEE 34 (1) (2014) 186–196. Jan.
[33] P. Karn, C. Partridge, Improving round-trip time estimates in reliable transport protocols, SIGCOMM Comput. Commun 25 (1) (1995) 66–74. Jan.
[34] Z.Q. Bo, L. Wang, B. Zhang, B. Zhang, Novel architecture for integrated wide area protection and control, in: 50th International Universities Power Engineering Conference (UPEC), Stoke on Trent, UK, 2015, pp. 1–4. Step.
[35] P. Sweeney, Error Correcting Coding: An Introduction, Prentice-Hall, 1991.
[36] M.H. Alwan, M. Singh, H.F. Mahdi, Performance comparison of turbo codes with LDPC codes and with BCH codes for forward error correcting codes, Res. Dev. IEEE (2015) 556–560. Dec.
[37] M. Wu, D. Han, X. Zhang, et al., Experimental research and comparison of LDPC and RS channel coding in ultraviolet communication systems[J], Opt. Express 22 (5) (2014) 5422–5430.
[38] Y. Xiao, C. Luo, C. Yang, The comparative analysis of LDPC and RS code, in: International Conference on Consumer Electronics, Communications and Networks (CECNet), 2011, pp. 4510–4513.
[39] C. Argon, S.W. McLaughlin, Optical OOK-CDMA and PPM-CDMA systems with turbo product codes, J. Lightwave Technol. 20 (9) (2002) 1653–1663. Sept.
[40] S.S. Muhammad, T. Javornik, I. Jelovčan, et al., Comparison of hard-decision and soft-decision channel coded M-ary PPM performance over free space optical links [J], Eur. Trans. Telecommun. 20 (8) (2009) 746–757.
[41] J. Li, K.R. Narayanan, E. Kurtas, C.N. Georghiades, On the performance of high-rate TPC/SPC codes and LDPC codes over partial response channels, IEEE Trans. Commun. 50 (5) (2002) 723–734. May.
[42] I.B. Djordjevic, O. Milenkovic, B. Vasic, Generalized low-density parity-check codes for optical communication systems, IEEE/OSA J. Lightw. Technol. 23 (2005) 1939–1946. May.
[43] A.N. Almaamory, H.A. Mohammed, Performance evaluation and comparison between LDPC and turbo coded MCCDMA, J. Eng. 18 (8) (2012). Apr.
[44] L.O. Uryvsky, S.O. Osypchuk, Comparative analysis of LDPC and BCH codes error-correcting capabilities, Inf. Telecommun. Sci. 5 (1) (2014) 5–9. June.
[45] Y. Ren, A. Dang, H. Guo, Iterative decodable block codes for high-speed free space optical communication, in: International Workshop on Satellite and Space Communications, Siena, Italy, 2009, pp. 220–224.
[46] *Error performance parameters and objectives for international, constant bit rate synchronous digital paths*, ITU-T G.828, Mar 2000.
[47] F.J. Macwilliams, N.J.A. Sloane, The Theory of Error-Correcting Codes, New York-Oxford, Amsterdam, North-Holland, 1977.
[48] T. Xu, X. Yin, D. You, Y. Li, Y. Wang, A novel communication network for three-level wide area protection system, in: IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, Pittsburgh, USA, 2008, pp. 1–8. Jul.
[49] M. Kuzlu, M. Pipattanasomporn, S. Rahman, Communication network requirements for major smart grid applications in HAN, NAN and WAN[J], Comput. Netw. 67 (2014) 74–88.
[50] Communications Requirements of Smart Grid Technologies, Department Energy, Washington, DC, 2010.
[51] S. Agrawal, S. Singh, An experimental study of TCP's energy consumption over a wireless link[C], in: 4th European Personal Mobile Communications Conference, 2001, pp. 20–22.
[52] A.J. Al-dweik, B.S. Sharif, Non-sequential decoding algorithm for hard iterative turbo product codes - [transactions letters], IEEE Trans. Commun. 57 (6) (2009) 1545–1549. June.

Qiuyu Lu received the Bachelor degree from the School of Cyber Science and Engineering, Wuhan University, Wuhan, China, in 2020. She is currently pursuing the Ph.D. degree in the School of Cyber Science and Engineering, Wuhan University. Her research interests include QoS guarantee for communications of smart grid and cyber security.

June Li received the B.S. and M.S. degrees in electrical engineering and computer engineering from the Wuhan University of Hydraulic and Electric Engineering, Wuhan, China, in 1986 and 1989, respectively, and the Ph.D. degree in computer engineering from Wuhan University, Wuhan, in 2004. She is currently a professor with the Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University. Her search interests include network architecture, cyber security, cyber-physical systems, and security of power industrial control systems.

Kai Yuan received the Ph.D. degree in the School of Electrical Engineering and Automation, Wuhan University, Wuhan, China, in 2019. He is currently a post-doctor in the School of Electrical and Electronic Engineering, Huazhong University of Science and Technology, Wuhan, China. His-research interests include QoS guarantee and reliability analysis for communications of smart grid.

Kaipei Liu received the Ph.D. degree in computer application technology from Wuhan University in 2001. He is currently a professor at the School of Electrical Engineering and Automation, Wuhan University. His-main research interests include DC transmission, renewable energy and smart grid, power quality and data analysis.

Ming Ni received the Ph.D. degree in electrical engineering from Southeast University in 1996. He is currently in Leidos Engineering LLC. His-main research interests include cyber physical power systems (CPPSs), safety and stability control of power systems.

Jianbo Luo is a senior engineer in NARI Group Corporation/State Grid Electric Power Research Institute. His-main research interests include analysis, comprehensive defense and control for safety and stability of power systems.