Permutation-Based TCP and UDP Transmissions to Improve Goodput and Latency in the Internet-of-Things

Yuli Yang and Lajos Hanzo

***Abstract*—To circumvent the degraded goodput, high latency and low resource utilisation efficiency of conventional transport-layer protocols in the Internet of Things, a novel permutation-based encapsulation scheme is proposed for seamless delivery of packets to achieve high goodput and low latency. In this scheme, extra application-layer data is delivered by our permutation-based data unit (PBDU), which is mapped onto the permutation associated with a legitimate tuple of various packet-lengths in a group. Therefore, the network interface throughput is substantially improved for a given number of resource units in the physical channel. The optimal design of this encapsulation is expressed in closed form to maximise the PBDU size, allowing us to quantify the attainable improvements of goodput and latency. In addition, the proposed scheme achieves higher physical-layer throughput and secrecy rate than conventional encapsulation.**

***Index Terms*—Goodput, latency, permutation-based data unit (PBDU), transmission control protocol (TCP), user datagram protocol (UDP).**

I. Introduction

The Internet of Things (IoT) has provided a powerful platform for devices to interact with each other. Driven by an extension of the IoT services to the so-called vertical domains of agricultural, industrial, military and civilian applications, the IoT infrastructure has to support high spectral efficiency and energy efficiency in massive device-to-device (D2D) communications, while also aiming for ultra-high reliability, low latency and maximum security [1], [2].

Reliable D2D connectivity over wireless networks relies on transport-layer protocols that operate above the network interface through which devices are able to exchange their information [3], [4]. The most popular protocols are the Transmission Control Protocol (TCP) [5] and the User Datagram Protocol (UDP) [6]. Both are defined by the Internet Engineering Task Force (IETF) as standards, prescribing how to establish and maintain a highly reliable communication link in packet-switched networks. In liaison with the Internet protocol (IP) [7], the TCP or UDP determines how to encapsulate application-layer data into packets for connection-oriented sessions, which allows communications between arbitrary types of devices.

In support of intelligent automation and networked robotics, a gateway has to handle the connections of a large number of devices, and there are a massive number of packets exchanged between the gateways [8]. However, the payload of each packet is very limited. For certain commands, the payload only consists of a few bits indicating the ON/OFF status or the UP/DOWN adjustment. Even so, the meta-data has to be added for indicating which specific device is the destination to execute these commands. As such, the meta-data size is comparable to the payload size in short packets [9], [10]. Therefore, efficient utilisation of meta-data and smart packet encapsulation need to be addressed.

A. Related Works

The physical-layer community has been encouraged by the concept of permutation modulation to conceive the ideas of spatial modulation [11], [12] and index modulation [13], [14], where a part of information bits is not radiated physically with the aid of classic modulation but mapped onto the activation of physical-channel resource units, e.g., in space, frequency, time or code domain [15].

Compared with conventional modulation schemes in the physical layer, this concept reduces the number of channel uses involved in the modulation and increases the channel capacity [16], [17]. Hence, physical-layer permutation modulation has been exploited in a wide range of applications, such as cooperative communications [18], [19], visible light communications [20], [21], millimetre-wave communications [22], [23], reconfigurable intelligent surfaces [24], and molecular communications [25], to increase their spectrum efficiency and energy efficiency. Moreover, physical-layer security can be enhanced by permutation modulation as well [26]–[28].

In [29], a novel transmission strategy was proposed to increase the network interface throughput in ultra-reliable and low-latency communications, by exploiting the permutation philosophy at the transport layer.

B. Motivation and Novelty

With the permutation-based transmission strategy proposed in [29], a part of application-layer data is conveyed through the permutation associated with a predetermined tuple of

Y. Yang is with the School of Engineering, University of Lincoln, Lincoln LN6 7TS, U.K. (e-mail: yyang@lincoln.ac.uk).

L. Hanzo is with the school of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (email: lh@ecs.soton.ac.uk).

L. Hanzo would like to gratefully acknowledge the financial support of the Engineering and Physical Sciences Research Council projects EP/N004558/1, EP/P034284/1, EP/P003990/1, COALESCE, of the Royal Society's Global Challenges Research Fund Grant as well as of the European Research Council's Advanced Fellow Grant QuantCom.

various packet-lengths in a group of packets. As this part of application-layer data is not directly encapsulated into packets, the resource utilisation efficiency is increased through reducing the physical channel uses required for the delivery of a given amount of application-layer data. However, the construction of various packet-lengths in [29] limits the potential of permutation-based transmissions, where the optimal designs pertaining to the maximum extra data encapsulated and the maximum resource utilisation efficiency are two extremes, creating an irreconcilable conflict of interest in the permutation group.

In this work, we propose a novel construction of various packet-lengths for the permutation group, to develop a flexible design solving the conflict in the permutation-based packet-encapsulation strategy. Specifically, our permutation-based encapsulation takes over the virtues of physical-layer permutation modulation and offers the following attractive features.

1) Increased Goodput: As an application-layer metric, goodput is utilised to quantify the amount of application-layer data, in bits, delivered through a given network interface, where the meta-data arising from control information is excluded. With our proposed encapsulation strategy, extra application-layer data are conveyed by the permutations rather than being delivered through the conventional network interface. Therefore, the goodput of the given network interface is substantially improved.

2) Improved Resource Utilisation Efficiency: Since the goodput is improved, the permutation-based transmission delivers more application-layer data through a given number of conventional physical-channel resource units, thus increasing the resource utilisation efficiency.

3) Reduced Latency and Pakcet-Dropping: In comparison with conventional encapsulation, the proposed scheme reduces the payload for delivering a certain amount of application-layer data. Since the TCP or UDP packets are shortened through the permutation-based encapsulation, both the TCP latency and the UDP packet-dropping chance are reduced.

4) Enhanced Security against Data Tracing: If the mapping pattern of the application-layer data conveyed by the permutations is interpreted as a secret key in the legitimate link, eavesdroppers are unable to successfully decode these data since they have no knowledge on the secret key.

The novelties of our work are boldly and explicitly contrasted to the state-of-the-art in Table I at a glance.

C. Contributions

The main contribution of this paper are highlighted by the following three aspects.

- *Approach:* To improve the tradeoff between goodput gain and resource utilisation efficiency, a novel construction of various packet-lengths in the permutation group is proposed for the permutation-based encapsulation at the transport layer, where a predetermined part of application-layer data is not directly encapsulated into packets but mapped onto the permutation associated with a legitimate tuple of various packet-lengths in a group of packets.

TABLE I
CONTRASTING THE NOVELTY OF THIS WORK TO THE LITERATURE.

Contributions	**This Work**	[11]–[25]	[26] [27]	[28]
Permutation Philosophy	✓	✓	✓	✓
Transport-Layer Encapsulation	✓			✓
Physical-Layer Modulation	✓	✓	✓	✓
Goodput	✓			✓
Latency	✓			✓
Physical-Layer Throughput	✓	✓	✓	✓
Security	✓		✓	✓
Flexible Design in IoT	✓		✓	

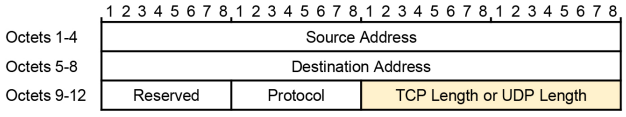


Fig. 1. The TCP or UDP pseudo header carried in the IP header.

- *Optimisation:* To maximise the amount of application-layer data conveyed by the permutations, the optimal designs for both the number of packet-lengths and the number of packets in the permutation group are obtained in closed-form expressions, subject to the constraints on window size and maximum packet-length.
- *Evaluation:* To verify the feasibility of the permutation-based encapsulation advocated, its performance is formulated and investigated in terms of its goodput, latency, physical-layer throughput and secrecy rate for TCP and UDP within the IoT. Moreover, an implementation of our permutation-based encapsulation in the Internet of vehicles is initiated for better illustrating its practical application.

In detailing the above contributions, the remainder of this paper is organized as follows. Firstly, the permutation-based encapsulation strategy is proposed in Section II, where the design is optimised and its resource utilisation efficiency is analysed. Subsequently, Section III evaluates the performance of permutation-based transmissions for both TCP and UDP in the IoT, followed by Section IV providing our design guidelines for practical IoT applications. In addition, Section V introduces an implementation of the permutation-based encapsulation in the Internet of vehicles. Finally, Section VI concludes this paper and offers a range of future research directions.

II. PERMUTATION-BASED ENCAPSULATION

In this section, the permutation-based encapsulation scheme is proposed and its design is optimised.

A. System Model

In the IoT infrastructure, the IETF's connection-oriented TCP and UDP protocols constitute the foundations for the transport layer to facilitate syntactic interoperability [2]. A

TCP or UDP packet consists of a header and a data unit (DU). The formatted meta-data is carried by the header to handle the packet delivery and manage the flow control. As shown in Fig. 1, Octets 11-12 in the TCP or UDP pseudo header carried by the IP header specify the packet length in bytes, which is equal to the header length plus the DU length [5], [6]. In particular, this information field designates the DU length in a TCP or UDP packet since the header length is fixed.

To maintain reliable delivery of packets, the so-called sequence number is included in the TCP header for identifying the order of data bytes transmitted from each device within the IoT [5]. In general, the UDP does not identify the order of packets [6], since the packets are expected to arrive as a continuous stream or they are dropped. However, in certain applications of UDP, e.g., designed for ultra-reliable delivery [30], the packet indices are contrived for ordering packets in potential retransmissions.

Given the DU length and the packet order, our permutation-based encapsulation scheme is first introduced with the assist of an instance in Table II and then formatted in a generalised scenario.

For instance, if two conventional DUs are grouped for the permutation of four DU lengths L_0, L_1, L_2, and L_3, there are $4^2 = 16$ permutations used to convey $\log_2 16 = 4$ bits of extra application-layer data. We define the extra application-layer data conveyed by the permutation as permutation-based DU (PBDU). The 16 permutations in this instance, together with their PBDUs, are illustrated in Table II.

In the generalised case, we consider N packets grouped for the permutation of K lengths. These packets are denoted by N various-size vectors $\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_N$, and the n^{th} packet is expressed as $\mathbf{s}_n = [\mathbf{u}_n, \mathbf{v}_n]$, where $\mathbf{u}_n$ and $\mathbf{v}_n$ stand for the header and the DU of Packet n, respectively, $n = 1, 2, \cdots, N$. The length of a header $\mathbf{u}_n$ is fixed, whilst the lengths of the DUs $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_N$ are varied. More concretely, one of the K elements in the set $\{L_0, L_1, \cdots, L_{K-1}\}$ is assigned to the length of a DU $\mathbf{u}_n$.

Hence, there are a total of K^N permutations with repetition in the arrangement of K lengths into a group of N DUs. We utilise these permutations to deliver extra application-layer data and, thus, the number of bits in the PBDU is given by

$$T_{\text{e}} = N \log_2 K. \tag{1}$$

That is, we have T_{e} extra application-layer data bits conveyed by a permutation-based encapsulation, in addition to the data encapsulated directly into the group of N DUs with K various lengths.

Elaborating slightly further, our permutation-based encapsulation divides the application-layer data into two parts, for the delivery via a group of DUs. The first part is the PBDU, composed of T_{e} bits, which is mapped onto the permutation of assigning K lengths to the DUs $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_N$ in this group. The second part of application-layer data is encapsulated into these DUs by adding the headers $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_N$ in a conventional way and physically delivered through the network interface. Note that, the lengths of the DUs $\mathbf{v}_n$, $n = 1, 2, \cdots, N$, are prescribed by the PBDU.

TABLE II
16 PERMUTATIONS OF 4 DU LENGTHS FOR A GROUP OF 2 DUS.

Permutation ♯	PBDU	DU ♯	DU Length
0	0000	1 2	L_0 L_0
1	0001	1 2	L_0 L_1
2	0010	1 2	L_0 L_2
3	0011	1 2	L_0 L_3
4	0100	1 2	L_1 L_0
5	0101	1 2	L_1 L_1
6	0110	1 2	L_1 L_2
7	0111	1 2	L_1 L_3
8	1000	1 2	L_2 L_0
9	1001	1 2	L_2 L_1
10	1010	1 2	L_2 L_2
11	1011	1 2	L_2 L_3
12	1100	1 2	L_3 L_0
13	1101	1 2	L_3 L_1
14	1110	1 2	L_3 L_2
15	1111	1 2	L_3 L_3

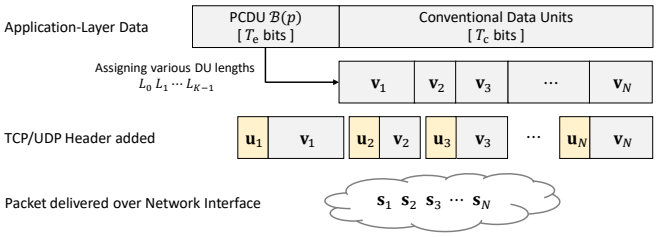


Fig. 2. The permutation-based encapsulation at the transport layer of a communication link.

The permutation-based encapsulation of application-layer data into a transmission over the network interface is presented in Fig. 2. Herein, the PBDU of the p^{th} permutation is calculated using $\mathcal{B}(p)$, where $\mathcal{B}(\cdot)$ stands for a T_{e}-bit binary coded decimal function, and $p = 0, 1, \cdots, K^N - 1$.

In principle, congestion control within the TCP and UDP determine the total amount of data delivered in a connection, i.e., meta-data plus application-layer data. Specifically, the information field 'window' in a TCP header indicates the window size of a connection [5], denoted by F_{s}, which prescribes the number of bytes that the receiver is prepared to receive at the moment. The congestion control in the UDP

is not managed by a 'window' but limited by the receiver's buffer size, which specifies the total amount of data to be transmitted [6]. Moreover, as the length of a TCP or UDP header $\mathbf{u}_n$ is fixed, the maximum DU length, denoted by $L_{\max}$, can be directly inferred from the 'TCP Length or UDP Length' field in Fig. 1, where the maximum packet-length is $2^8 - 1$ bytes [5], [6].

B. PBDU Maximisation

Based on a given window size $F_{\rm s}$ and a given maximum DU length $L_{\max}$, the number of bits in the PBDU, $T_{\rm e}$, can be maximised through the optimal design with the number of DUs, N, and the number of DU lengths, K, in a permutation group.

For a permutation-based encapsulation of application-layer data into a group of N DUs, the K lengths $L_0, L_1, \cdots, L_{K-1}$ are varied and the probability that the $k^{\rm th}$ length L_k occurs in the encapsulation is denoted by γ_k, $k = 0, 1, \cdots, K-1$. Therefore, the mean length of a DU within this group is obtained by

$$\bar{L} = \sum_{k=0}^{K-1} \gamma_k L_k. \tag{2}$$

In practice, the various DU lengths can be set to an arithmetic sequence of $L_k = M + kC$ bits, $k = 0, 1, \cdots, K-1$, where the initial term M is the shortest DU length, and C is the common difference of successive lengths. Both M and C are natural numbers.

Without loss of generality, the lengths $M, M+C, \cdots, M+(K-1)C$ are assumed to occur in the encapsulation at the same probability of $\gamma_k = 1/K$, $\forall k \in \{0, 1, \cdots, K-1\}$. Thus, the mean length of a DU within this group, in the unit of [bits], is expressed as

$$\bar{L} = M + \frac{(K-1)C}{2}. \tag{3}$$

As shown in Fig. 2, the header $\mathbf{u}_n$, of length $L_{\rm H}$ bits, is added to encapsulate the DU $\mathbf{v}_n$ in a packet $\mathbf{s}_n$, $n \in \{1, 2, \cdots, N\}$, which generates a part of the frame payload to be physically transmitted through the network interface. Since the total amount of meta-data plus application-layer data encapsulated in the N packets is bounded by the window size of a connection, i.e., $F_{\rm s}$ bits, we have $N(L_{\rm H} + \bar{L}) \leqslant F_{\rm s}$.

Furthermore, the limitation on the maximum DU length is $L_{\max}$, expressed as $M + (K-1)C \leqslant L_{\max}$.

As such, the maximisation of the number of bits in the PBDU can be formulated as the optimisation problem of

$$T_{\rm e}^* = \max_{N,K} N \log_2 K \tag{4}$$

$$\text{subject to} \quad 0 \leqslant N(L_{\rm H} + M + (K-1)C/2) \leqslant F_{\rm s} \tag{4a}$$

$$2 \leqslant K \leqslant (L_{\max} - M)/C + 1, \tag{4b}$$

where $T_{\rm e}^*$ denotes the maximum number of bits in the PBDU. The constraints (4a) and (4b) are conditioned by the window size and the maximum DU length, respectively, indicated in the TCP or UDP header.

Based on (4a), we have

$$N \leqslant \frac{2F_{\rm s}}{2(L_{\rm H} + M) + (K-1)C}. \tag{5}$$

Upon substituting (5) into (4), we arrive at

$$T_{\rm e} \leqslant \frac{2F_{\rm s} \log_2 K}{2(L_{\rm H} + M) + (K-1)C} \triangleq \phi(K). \tag{6}$$

Obviously, $T_{\rm e}$ achieves its maximum in the condition that

$$\frac{{\rm d}\phi(K)}{{\rm d}K} = 0 \tag{7}$$

for (4b). More specifically, this condition is rewritten as

$$\frac{2(L_{\rm H} + M) + (K-1)C}{K \ln 2} - C \log_2 K = 0 \tag{8}$$

for $K \in [2, (L_{\max} - M)/C + 1]$.

Given M and C, the optimal number of DU lengths in a permutation group, K^*, to maximise the number of bits in the PBDU, $T_{\rm e}$, is the solution to the following equation:

$$K(\ln K - 1) = \frac{2(L_{\rm H} + M)}{C} - 1 \tag{9}$$

within the interval $[2, (L_{\max} - M)/C + 1]$. With a notation for the constant $A \coloneqq 2(L_{\rm H} + M)/C - 1$, the solution to (9) is obtained by

$$K^* = \frac{A}{\mathcal{W}(0, A/e)}, \tag{10}$$

where e is the Euler's number, and $\mathcal{W}(0, x)$ is the $0^{\rm th}$ branch of the Lambert's W function [31]. Note that K^* has to be an integer and it is preferably a power of two.

Subsequently, the optimal number of DUs in the permutation group, N^*, is expressed as

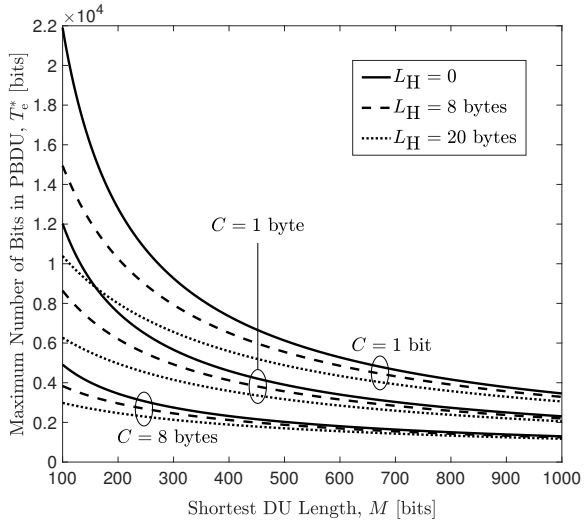
$$N^* = \frac{2F_{\rm s}}{CK^* \ln K^*}. \tag{11}$$

As a result, the maximum number of bits in the PBDU is

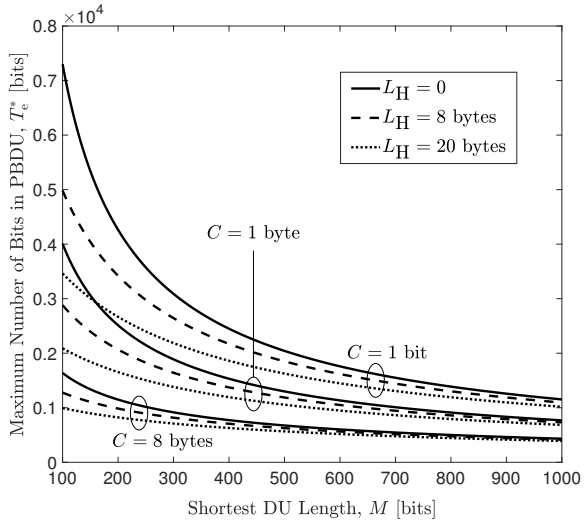
$$T_{\rm e}^* = \frac{2F_{\rm s}}{(C \ln 2)K^*}, \tag{12}$$

with the optimal design presented in (10) and (11). In detail, if the window size $F_{\rm s}$ is given, the optimal design relies on varying K^* lengths $L_0, L_1, \cdots, L_{K^*-1}$ in a group of N^* DUs, to convey the maximum number of bits in the PBDU, $T_{\rm e}^*$.

In Fig. 3, $T_{\rm e}^*$ is depicted as a function of the shortest DU length M, where various settings of the header length $L_{\rm H}$, the common difference C and the window size $F_{\rm s}$ are investigated. As is shown in this figure, $T_{\rm e}^*$ is reduced with the increase in $L_{\rm H}$, M or C. The main reason behind this can be found from (5), which indicates that a larger $L_{\rm H}$, M or C leads to a smaller N, hence causing a smaller $T_{\rm e}^*$. Furthermore, comparing Figs. 3(a) and 3(b), we may find that $T_{\rm e}^*$ is rising upon increasing $F_{\rm s}$, which is revealed by (12) as well.



(a) Window Size $F_s = 6 \times 10^4$ bytes



(b) Window Size $F_s = 2 \times 10^4$ bytes

Fig. 3. The maximum number of bits in the PBDU, T_e^*, versus the shortest DU length M.

C. Resource Utilisation Efficiency

In general, it is assumed that the delivery of a single bit within a frame requires a resource unit of the physical channel use. Thus, a permutation-based frame of F_s bits to be physically delivered through the network interface needs F_s channel uses at the physical layer. The resource utilisation efficiency of the permutation-based encapsulation is formulated by the total number of application-layer data bits in the PBDU plus those in the conventional DUs conveyed through a single channel use unit at the physical layer, which is characterised as

$$E_R = \frac{T_e^* + N\bar{L}}{F_s} = \frac{\log_2 K^* + \bar{L}}{L_H + \bar{L}} = \frac{\log_2 K^* + M + (K^* - 1)C/2}{L_H + M + (K^* - 1)C/2}. \quad (13)$$

Given the header length L_H, the resource utilisation efficiency E_R is not influenced by the window size F_s, but determined by the shortest DU length M and the common difference of successive lengths, C.

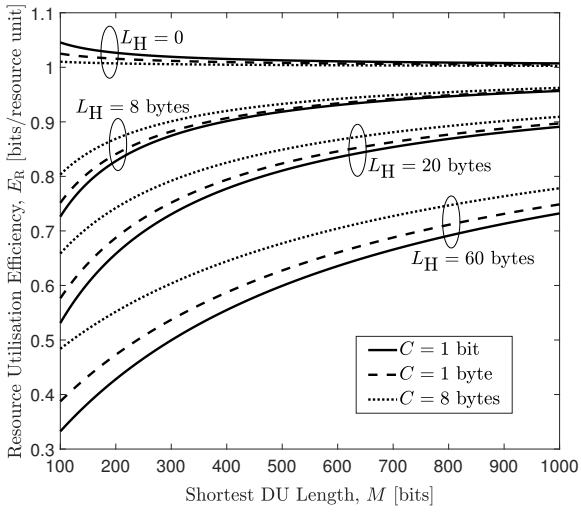


Fig. 4. The resource utilisation efficiency E_R versus the shortest DU length M.

In Fig. 4, the resource utilisation efficiency E_R is plotted as a function of the shortest DU length M, where various settings of L_H and C are investigated. This figure reveals that E_R decreases as L_H increases, as governed by (13).

From (13), we may find that $E_R < 1$ when $\log_2 K^* < L_H$ and $E_R \geqslant 1$ when $\log_2 K^* \geqslant L_H$, which is reflected in Fig. 4 as well. Moreover, upon increasing M or C, E_R is improved in the former condition of $L_H > \log_2 K^*$, but reduced in the latter condition of $L_H \leqslant \log_2 K^*$.

III. Performance Evaluation

In this section, the performance of our permutation-based encapsulation is evaluated in terms of its goodput, latency, physical-layer throughput and secrecy rate for typical TCP and UDP packet structures in the IoT.

The TCP header length ranges from 20 bytes to 60 bytes [5] and the UDP header consists of 8 bytes [6]. In a TCP header, the maximum window size defined by [5] is $F_s = 65535$ bytes, which can be increased up to 65535×16384 bytes through the window scale option. For the UDP, F_s is determined by the receiver's buffer size. Besides, the packet length specified in both the TCP header and the UDP header is capable of identifying 65535 lengths in bytes, as shown in Fig. 1. The parameters set for our performance analysis are listed in Table III.

TABLE III
TCP and UDP Parameters.

Parameter	Variable	Protocol	Value	Unit
Window Size	F_s	TCP	6×10^4	bytes
		UDP	8000	
Header Length	L_H	TCP	20	bytes
			60	
		UDP	8	

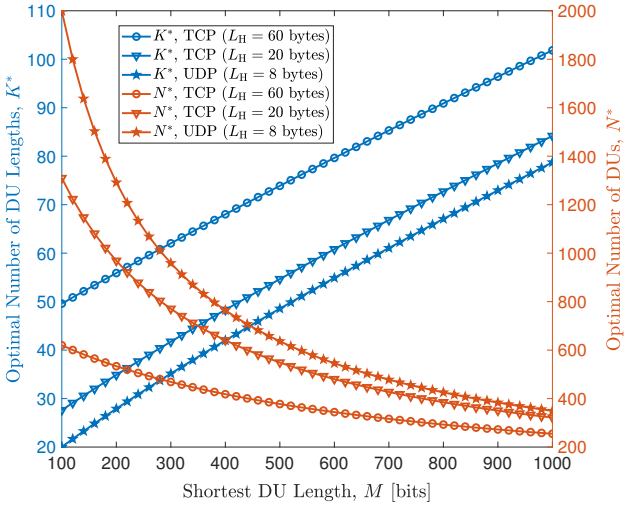


Fig. 5. The optimal number of DU lengths, K^* in (10), and the optimal number of DUs, N^* in (11), versus the shortest DU length M.

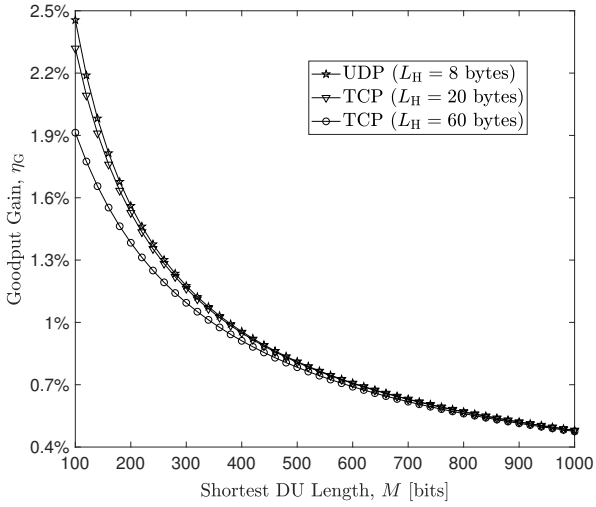


Fig. 6. The goodput gain $\eta_{\rm G}$ versus the shortest DU length M.

For permutation-based transmissions, the shortest DU length M varies from 100 to 1000 bits, and the common difference of successive lengths is set to 1 byte, i.e., $C = 8$ bits. Given these settings, the optimal number of DU lengths, K^* in (10), and the optimal number of DUs within the permutation group, N^* in (11), are plotted in Fig. 5.

A. Goodput Gain

In a permutation-based encapsulation, the total number of application-layer data bits transmitted over conventional DUs $\mathbf{v}_n$, $n = 1, 2, \cdots, N$, through the network interface can be calculated using

$$T_{\rm c} = N\bar{L}, \tag{14}$$

which is equal to the goodput in a conventional transmission. The mean length of a conventional DU, $\bar{L}$, is given by (3). With the optimal design given by (10), we have

$$\bar{L} = M + \frac{C}{2}\left(\frac{A}{\mathcal{W}(0, A/e)} - 1\right). \tag{15}$$

Therefore, the number of application-layer data bits in total, i.e., the bits in the PBDU plus those in the conventional DUs, delivered within a permutation-based encapsulation is $T_{\rm e}+T_{\rm c}$, where the maximum number of bits in the PBDU, $T_{\rm e}^*$, is given by (12).

The goodput gain attained by a permutation-based encapsulation, denoted by $\eta_{\rm G}$, is formulated by the ratio of the extra goodput conveyed by the permutation to the goodput in a conventional transmission, i.e.,

$$\eta_{\rm G} = \frac{T_{\rm e}^*}{T_{\rm c}} = \frac{\log_2 K^*}{\bar{L}}, \tag{16}$$

where K^* is given by (10) and $\bar{L}$ is given by (15).

In Fig. 6, the goodput gain $\eta_{\rm G}$ is depicted as a function of the shortest DU length M, where $\eta_{\rm G}$ increases with the reduction of header length $L_{\rm H}$. More specifically, the permutation-based design will fully benefit from the efficient utilisation of meta-data. Further, $\eta_{\rm G}$ increases upon decreasing M, which implies that the permutation of shorter DU lengths is preferred by the permutation-based encapsulation in terms of goodput gain.

B. TCP Latency and UDP Packet-Dropping

The latency in a TCP network mainly results from retransmissions and excessive traffic load, both of which can be mitigated by the permutation-based encapsulation. In contrast, as the UDP is often adopted by time-sensitive applications, it prefers to drop packets rather than wait for delayed packets in retransmissions [6]. Nevertheless, within particular applications that aim for ultra-reliable transport, the UDP is required to accommodate retransmissions [30].

As shown in Fig. 2, the application-layer data in the PBDU is not physically transmitted in the conventional way but conveyed by the permutation of various DU lengths, which guarantees the successful delivery of $(T_{\rm e} + T_{\rm c})$ bits via the physical transmission of $T_{\rm c}$ bits. Thus, the PBDU contributes to the reduction of both the traffic load and the retransmission requirements in the TCP as well as to the reduction of dropped packets in the UDP.

From a long-term perspective, the retransmission chance, the data transfer rate, and the network congestion state are consistent in a TCP network. Consequently, the ratio of the latency with permutation-based encapsulation to the conventional transmission latency, denoted by $\eta_{\rm L}$, can be calculated using

$$\eta_{\rm L} = \frac{T_{\rm c}}{T_{\rm e} + T_{\rm c}} = \frac{\bar{L}}{\log_2 K^* + \bar{L}}, \tag{17}$$

which is equivalent to the ratio of the packet-dropping chance in a UDP network using the permutation-based encapsulation to that in the conventional UDP transmission.

In Fig. 7, the TCP latency or UDP packet-dropping ratio $\eta_{\rm L}$ is plotted versus the shortest DU length M, where $\eta_{\rm L}$ decreases with the reduction of header length $L_{\rm H}$. Moreover, $\eta_{\rm L}$ decreases upon decreasing M. Since smaller $\eta_{\rm L}$ means lower

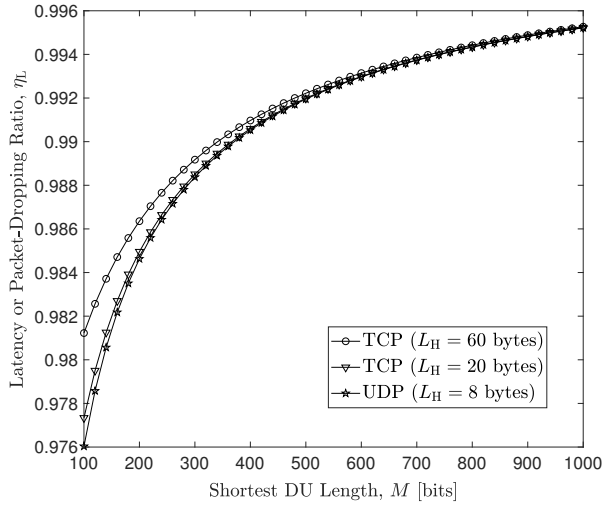


Fig. 7. The latency or packet-dropping ratio $\eta_{\rm L}$ versus the shortest DU length M.

TCP latency or less UDP packets dropped in the permutation-based transmission, the utilisation of shorter header and the permutation of shorter DU lengths are preferred for reducing the latency or packet-dropping chance.

C. Physical-Layer Throughput

For the delivery of packets over the network interface, the overhead has to be added to a frame, including the preamble, the frame header and trailer, used for channel estimation, flow control and error detection/correction.

Herein, the physical-layer throughput is formulated as the effective application-layer data rate per channel use at the physical layer, in the unit of [bits/channel use]. We remark that the effective application-layer data excludes the TCP/UDP headers at the transport layer and the overhead at the physical layer. Assuming that the overhead of each frame is composed of $O_{\rm H}$ bits, the physical-layer throughput with the permutation-based encapsulation, denoted by $R_{\rm pbt}$, is calculated as

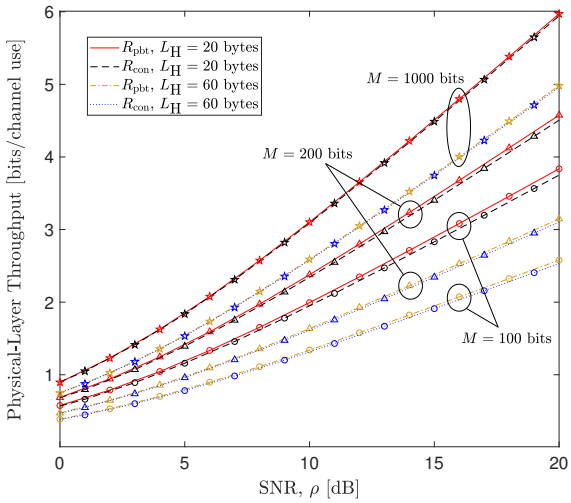
$$R_{\rm pbt}=\frac{T_{\rm e}+T_{\rm c}}{O_{\rm H}+F_{\rm s}}\log_2(1+\rho), \tag{18}$$

and that of the conventional transmission, denoted by $R_{\rm con}$, is expressed as

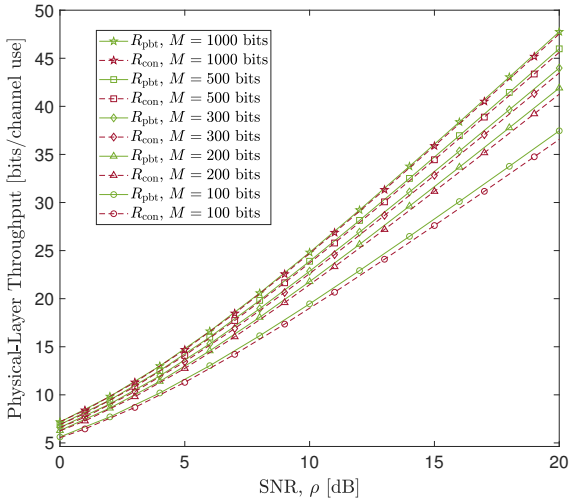
$$R_{\rm con}=\frac{T_{\rm c}}{O_{\rm H}+F_{\rm s}}\log_2(1+\rho), \tag{19}$$

where ρ is the signal-to-noise power ratio (SNR) at the physical layer, and the window size is $F_{\rm s}=NL_{\rm H}+T_{\rm c}$.

In Fig. 8, the physical-layer throughput is plotted as a function of the shortest DU length M to compare the permutation-based TCP and UDP transmissions with conventional ones, where the overhead of each frame has $O_{\rm H}=15$ bytes. As shown in this figure, both the transmission with permutation-based encapsulation and the conventional transmission achieve higher physical-layer throughput upon increasing M. The main reason behind this is that the meta-data size of $(O_{\rm H}+NL_{\rm H})$ is comparable to the payload size of $T_{\rm c}=N\bar{L}$ in short packets and higher resource utilisation efficiency is achieved by longer packets. Moreover, the physical-layer throughput with the permutation-based encapsulation is always higher than that of the conventional transmission, thanks to the extra $T_{\rm e}$ bits conveyed by PBDU, which can be found from the comparison between (18) and (19) as well. However, the physical-layer throughput gain attained by the permutation-based encapsulation over the conventional transmission is reduced as M increases, mainly because the goodput gain attained by the permutation-based encapsulation over the conventional one decreases upon increasing M, as shown in Fig. 6.



(a) TCP



(b) UDP

Fig. 8. The physical-layer throughput comparisons between the permutation-based encapsulation and the conventional transmission, with the frame overhead $O_{\rm H}=15$ bytes.

D. Secrecy Rate

As a promising information-source-based solution to the physical-layer security, the physical-layer secret key can be generated by varying the bit-to-symbol mapping patterns in line with instantaneous channel quality indicator (CQI) of the legitimate link [26], [27]. Inspired by this solution, varying the PBDU mapping patterns according to the legitimate CQI will be exploited to further enhance the security. If there are $P = K^N$ permutations, the number of PBDU mapping patterns is $P! = P \times (P-1) \times \cdots \times 1$, i.e., the factorial of P. In the example introduced by Table II, there are 16 permutations and, thereby, 16! mapping patterns for PBDU. The q^{th} mapping pattern of the PBDU is denoted by $\mathfrak{M}_q$, where $q \in \{1, 2, \cdots, 16!\}$. The PBDU mapping pattern elaborated in Table II can be regarded as the first one, denoted by $\mathfrak{M}_1 : \sharp p \to \mathcal{B}(p)$, $p = 0, 1, \cdots, 15$. Another mapping pattern, e.g., the i^{th} one, can be expressed as $\mathfrak{M}_i : \sharp p \to \mathcal{B}(p + i - 1 \mod 16)$, $p = 0, 1, \cdots, 15$, $i = 2, 3, \cdots, 16$.

In a time division duplex connection, the CQI is not transmitted during the handshake between the transmitter and the receiver, as the channel reciprocity allows them to exploit each other's CQI. Hence, eavesdroppers cannot access the legitimate CQI through wiretapping, and they can never make decisions on the bits in the PBDU even if searching through all possible mapping patterns in a brute-force way, because they have no basis to choose the correct mapping pattern for their decisions. If an eavesdropper adheres to a single mapping pattern of the PBDU, statistically speaking it may correctly recover $1/P$ of the bits in the PBDU, but cannot recognise which bits are correctly recovered in the long run. As such, the secrecy rate with the CQI-mapped permutation-based encapsulation can be formulated by

$$R_{\text{S,pbt}} = \frac{(K^N - 1)T_{\text{e}}}{K^N(O_{\text{H}} + F_{\text{s}})} C_{\text{t}} + R_{\text{S,con}}, \tag{20}$$

where C_{t} is the achievable data rate over the legitimate link, and

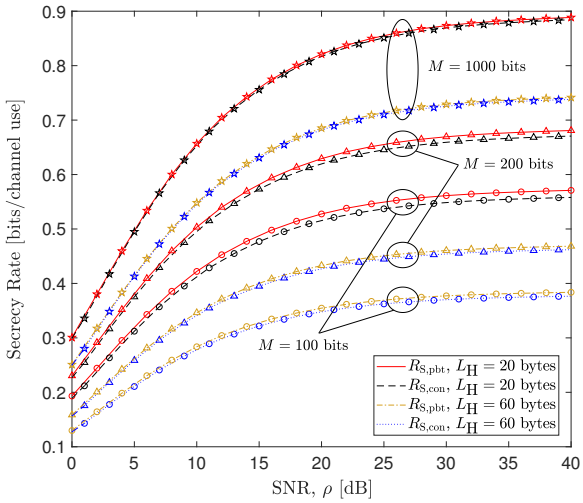
$$R_{\text{S,con}} = \frac{T_{\text{c}}}{O_{\text{H}} + F_{\text{s}}} C_{\text{t}} \tag{21}$$

denotes the secrecy rate of a conventional CQI-mapped transmission.

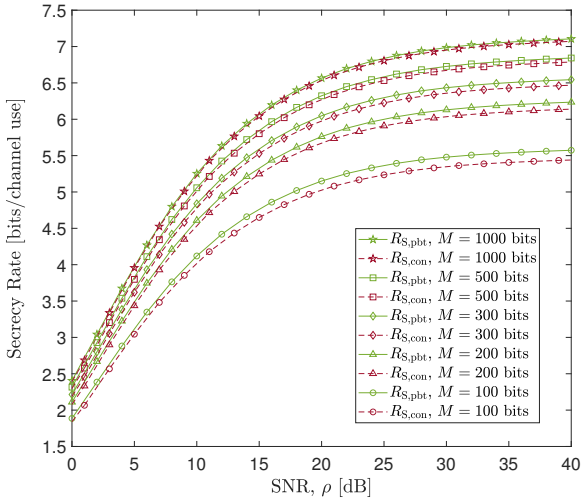
In Fig. 9, the secrecy rate with the CQI-mapped permutation-based encapsulation, $R_{\text{S,pbt}}$ in (20), is compared with that of the conventional CQI-mapped transmission, $R_{\text{S,con}}$ in (21), over a fading channel using BPSK modulation. In this case, C_{t} is the achievable data rate of a BPSK system, calculated using

$$C_{\text{t}} = 1 - \frac{1}{2}\mathcal{E}_\mu \left\{ \mathcal{E}_w \left\{ \log_2\left(1 + e^{-\frac{2\mu^2 + 2\mu w}{\sigma^2}}\right) + \log_2\left(1 + e^{-\frac{2\mu^2 - 2\mu w}{\sigma^2}}\right) \right\} \right\}, \tag{22}$$

where $w \sim \mathcal{N}(0, \sigma^2)$ denotes the additive white Gaussian noise and $\mu \sim \mathcal{N}(0, 1)$ denotes the channel fading. As shown this figure, the secrecy rate of CQI-mapped transmissions with both permutation-based and conventional encapsulations is improved as the shortest DU length M increases. However, the secrecy rate gain attained by the permutation-based encapsulation over the conventional one is reduced upon increasing M.



(a) TCP



(b) UDP

Fig. 9. Secrecy rate comparisons of CQI-mapped transmissions, between the permutation-based and conventional encapsulations, in a fading channel using BPSK modulation.

IV. Design Guidelines

Based on the above performance analysis, tangible design guidelines are provided for our permutation-based encapsulation applied in practical scenarios, especially within the IoT.

A. Performance

Typically, two kinds of performance measurement are used to assess a design: its relative performance and its absolute performance. The former is used for quantifying the performance gain achieved by a design against its peers, and the latter is used for assessing the design's performance in given settings.

As shown in Figs. 6, 7, 8 and 9, the performance gain attained by the permutation-based encapsulation over the conventional one, in the metrics of goodput, latency, physical-layer throughput and CQI-mapped secrecy rate, increases upon decreasing the DU lengths. Moreover, as shown in Fig. 5, the number of DUs in the permutation group is rising as the DU lengths decrease. Explicitly, the relative performance of the permutation-based encapsulation is improved by the permutation over a large number of shorter DU lengths.

Therefore, in comparison with the conventional encapsulation, the permutation-based scheme is particularly preferred in the IoT, where massive packets are exchanged between the gateways and the DU in each packet is very short.

When it comes to the absolute performance attained, Figs. 8 and 9 demonstrate that both the physical-layer throughput and the CQI-mapped secrecy rate of the permutation-based transmission is improved with the increase in the DU lengths. The main reason behind this is that, if the header length obeys $L_{\mathrm{H}} > \log_2 K^*$, the resource utilisation efficiency of the permutation-based encapsulation is improved upon increasing the DU lengths, as shown in Fig. 4 and (13).

Consequently, the tradeoff between the relative performance and the absolute performance has to be taken into account in the permutation-based design.

B. Complexity

The delivery of PBDUs in permutation-based transmissions does not affect the computational complexity in the physical layer.

As shown in Fig. 1, the TCP or UDP packet length is an explicit quantity in the header and it can be confirmed by the actual packet length as well. Owing to these twin-fold information matches, the PBDU can be successfully recovered in the transport layer as long as the recovery of the data delivered in conventional DUs passes the cyclic redundancy check (CRC) in a frame.

Compared with the conventional transmission regime, the payload is reduced in the permutation-based encapsulation and, accordingly, the computational complexity in the physical layer is reduced for the delivery of the same amount of application-layer data.

C. Compatibility

Our permutation-based encapsulation may flexibly coexist with the conventional encapsulation of application-layer data in the transport layer [5], by using a single bit of the 'reserved' field in the TCP or UDP header to indicate whether the permutation-based or the conventional encapsulation is being executed.

To further improve the robustness of permutation-based transmissions, the CRC can be applied to the PBDU as well, which will guarantee a better recovery of the PBDU.

V. An implementation in Internet of Vehicles

In this section, an implementation of the permutation-based encapsulation is introduced in the Internet of vehicles, to further demonstrate the merits it brings to the transport layer in practical applications.

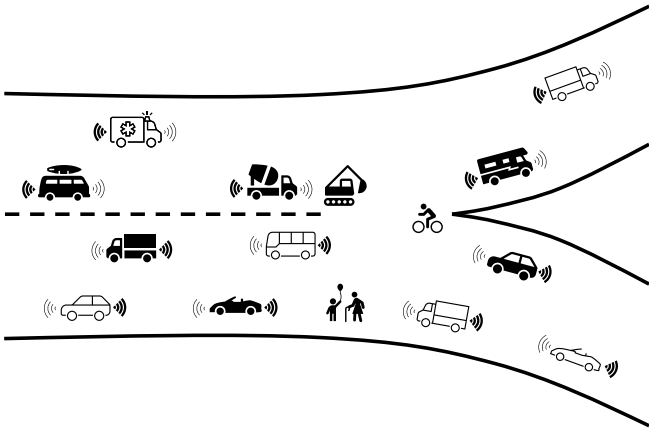


Fig. 10. An Internet of vehicles.

An Internet of vehicles scenario is shown in Fig. 10, where all the vehicles are driverless [32]. Consider eight typical signaling formats designed for enhancing safety, navigation, and traffic control in the Internet of vehicles [33], [34], as listed in Table IV.

- Emergency: If there is an emergency in the road, e.g., a pedestrian appears suddenly, the front vehicle has to take an immediate action, e.g., stop, and inform the following vehicles of the emergency.
- Lane change: If a vehicle intends to change lanes, it has to inform the vehicles around it of its destination lane.
- Path selection: A vehicle has to be informed of the path selection when approaching a junction.
- Road condition: The front vehicle has to inform the following vehicles of the road conditions, e.g., uneven surfaces and slopes.
- Traffic Condition: A vehicle has to be informed of the traffic condition in the road ahead, based on which it will adjust the velocity.
- Deceleration: A vehicle has to inform the vehicles behind it if intending to decelerate.
- Acceleration: A vehicle has to inform the vehicles in front and behind it if intending to accelerate.
- Obstacle: The front vehicle has to inform the following vehicles of the obstacle it has detected in its perception region.

In Table IV, the signaling type is indicated by the first three bits in the signaling payload, which are contained by the PBDU in our permutation-based encapsulation and conveyed by the various payload length. Compared with the conventional encapsulation, these signaling payloads are significantly reduced by mapping the PBDU onto the payload length, which is shown in the fourth column of Table IV.

Furthermore, the overall signaling latency in the Internet of vehicles is investigated. As shown in Table IV, the mean length of a signaling payload using our permutation-based

TABLE IV
TYPICAL SIGNALING PAYLOAD FORMATS IN THE INTERNET OF VEHICLES.

Signaling	Probability	Payload	PBDU reduction	Description
Emergency	$1/4$	000	100%	The emergency warning has the highest priority.
Lane Change	$1/8$	001 + 3-bit pattern	50%	The 3-bit pattern indicates the vehicle's destination lane.
Path Selection	$1/8$	101 + 4-bit pattern	42.9%	The 4-bit pattern indicates the path information.
Road Condition	$1/8$	101 + 5-bit pattern	37.5%	The 5-bit pattern indicates the road condition.
Traffic Condition	$1/8$	100 + 6-bit pattern	33.3%	The 6-bit pattern indicates the traffic condition.
Deceleration	$1/12$	010 + 7-bit pattern	30%	The 7-bit pattern indicates the deceleration rate.
Acceleration	$1/12$	010 + 8-bit pattern	27.3%	The 8-bit pattern indicates the acceleration information.
Obstacle	$1/12$	011 + 2-byte pattern	15.8%	The 2-byte pattern indicates the obstacle's location and size.

encapsulation is

$$\begin{aligned}\bar{L}_{\rm pbt} &= \frac{1}{8}\times(3+4+5+6)+\frac{1}{12}\times(7+8+16)\\ &= 4.83 \text{ bits},\end{aligned} \tag{23}$$

and that in the conventional encapsulation is

$$\begin{aligned}\bar{L}_{\rm con} &= \frac{1}{4}\times 3+\frac{1}{8}\times(6+7+8+9)+\frac{1}{12}\times(10+11+19)\\ &= 7.83 \text{ bits}.\end{aligned} \tag{24}$$

Consequently, the ratio of the overall payload latency in permutation-based encapsulation to the conventional payload latency in the Internet of vehicles is given by

$$\frac{\bar{L}_{\rm pbt}}{\bar{L}_{\rm con}} = 0.62. \tag{25}$$

Since a UDP/IP header can be compressed into two bytes through the Robust Header Compression (ROHC) for further reducing the overhead [35], the header length in the Internet of vehicles is set to $L_{\rm H} = 16$ bits. Thus, the ratio of the overall signaling latency in permutation-based encapsulation to the conventional signaling latency in the Internet of vehicles is obtained by

$$\eta_{\rm L} = \frac{16+\bar{L}_{\rm pbt}}{16+\bar{L}_{\rm con}} = 0.87. \tag{26}$$

As a result, the overall signaling latency of our permutation-based encapsulation is reduced by 13% in comparison to the conventional signaling latency, within the Internet of vehicles under study.

VI. CONCLUSION AND DISCUSSION

A novel transport-layer design, referred to as permutation-based encapsulation, was proposed in this paper for the IoT, where the permutation associated with the legitimate tuple of various lengths in a group of DUs was exploited to convey an extra amount of application-layer data. The number of bits in the PBDU was formally maximised by explicating an optimisation problem and the resource utilisation efficiency was analysed. To further characterise the performance of permutation-based transmissions, their goodput gain, latency ratio, physical-layer throughput and secrecy rate were formulated and investigated for typical TCP and UDP packet structures in the IoT. Based on our investigations, tangible design guidelines were developed for practical applications.

Although the proposed encapsulation strategy substantially improves the goodput by reducing the payload in the IoT, the overhead arising from meta-data limits the advantage of our permutation-based design to a certain degree, as it does in conventional transmissions. Thus, the minimisation of overhead format and the efficient encoding of meta-data still need further research for enhancing the IoT performance with the aid of our permutation-based encapsulation.

REFERENCES

[1] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang and W. Zhao, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications", *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125-1142, Oct. 2017.

[2] S. Keoh, S. Kumar and H. Tschofenig, "Securing the Internet of Things: A standardization perspective", *IEEE Internet Things J.*, vol. 1, no. 3, pp. 265-275, Jun. 2014.

[3] L. Chettri and R. Bera, "A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems", *IEEE Internet Things J.*, vol. 7, no. 1, pp. 16-32, Jan. 2020.

[4] M. Klapez, C. A. Grazia and M. Casoni, "Application-Level Performance of IEEE 802.11p in Safety-Related V2X Field Trials", *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3850-3860, May 2020.

[5] J. Postel, "Transmission Control Protocol", Internet Engineering Task Force (IETF) RFC 793, Sep. 1981.

[6] J. Postel, "User Datagram Protocol", Internet Engineering Task Force (IETF) RFC 768, Aug. 1980.

[7] J. Postel, "Internet Protocol", Internet Engineering Task Force (IETF) RFC 791, Sep. 1981.

[8] International Electrotechnical Commission (IEC), *IoT 2020: Smart and Secure IoT Platform*. Switzerland, 2016.

[9] V. Freschi and E. Lattanzi, "A study on the impact of packet length on communication in low power wireless sensor networks under interference", *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3820-3830, Apr. 2019.

[10] D. Kim, H. Nam and D. Kim, "Adaptive code dissemination based on link quality in wireless sensor networks", *IEEE Internet Things J.*, vol. 4, no. 3, pp. 685-695, Jun. 2017.

[11] M. Renzo, H. Haas, and P. Grant, "Spatial modulation for multiple-antenna wireless systems: A survey", *IEEE Commun. Mag.*, vol. 49, no. 12, pp. 182-191, Dec. 2011.

[12] P. Yang, M. Di Renzo, Y. Xiao, S. Li, and L. Hanzo, "Design guidelines for spatial modulation", *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 6-26, 1st Quart. 2015.

[13] E. Basar, U. Aygolu, E. Panayirci, and H. Poor, "Orthogonal frequency division multiplexing with index modulation", *IEEE Trans. Signal Process.*, vol. 61, no. 22, pp. 5536-5549, Nov. 2013.

[14] E. Basar, "Index modulation techniques for 5G wireless networks", *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 168-175, July 2016.

[15] N. Ishikawa, S. Sugiura and L. Hanzo, "50 years of permutation, spatial and index modulation: From classic RF to visible light communications and data storage", *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1905-1938, 3rd Quart. 2018.

[16] Y. Yang and B. Jiao, "Information-guided channel-hopping for high data rate wireless communication", *IEEE Commun. Lett.*, vol. 12, no. 4, pp. 225–227, Apr. 2008.

[17] Y. Yang and S. Aissa, "Information guided channel hopping with an arbitrary number of transmit antennas", *IEEE Commun. Lett.*, vol. 16, no. 10, pp. 1552-1555, Oct. 2012.

[18] Y. Yang and S. Aissa, "Information-guided transmission in decode-and-forward relaying systems: Spatial exploitation and throughput enhancement", *IEEE Trans. Wireless Commun.*, vol. 10, no. 7, pp. 2341-2351, July 2011.

[19] Y. Yang, "Spatial modulation exploited in non-reciprocal two-way relay channels: Efficient protocols and capacity analysis", *IEEE Trans. Commun.*, vol. 64, no. 7, pp. 2821-2834, July 2016.

[20] R. Mesleh, H. Elgala, and H. Haas, "Optical spatial modulation", *IEEE/OSA J. Opt. Commun. Netw.*, vol. 3, no. 3, pp. 234-244, Mar. 2011.

[21] T. Fath and H. Haas, "Performance comparison of MIMO techniques for optical wireless communications in indoor environments", *IEEE Trans. Commun.*, vol. 61, no. 2, pp. 733-742, Feb. 2013.

[22] P. Liu, M. Di Renzo, and A. Springer, "Line-of-sight spatial modulation for indoor mmWave communication at 60 GHz", *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7373–7389, Nov. 2016.

[23] N. Ishikawa, R. Rajashekar, S. Sugiura, and L. Hanzo, "Generalized-spatial-modulation-based reduced-RF-chain millimeter-wave communications", *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 879–883, Jan. 2017.

[24] E. Basar, "Reconfigurable intelligent surface-based index modulation: A new beyond MIMO paradigm for 6G", *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 3187-3196, May 2020.

[25] M. C. Gursoy, E. Basar, A. E. Pusane and T. Tugcu, "Index modulation for molecular communication via diffusion systems", *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3337-3350, May 2019.

[26] Y. Yang and M. Guizani, "Mapping-varied spatial modulation for physical layer security: Transmission strategy and secrecy rate", *IEEE J. Sel. Areas Commun.*, vol. 36, no. 4, pp. 877-889, Apr. 2018.

[27] Y. Yang, M. Ma, S. Aissa and L. Hanzo, "Physical-layer secret key generation via CQI-mapped spatial modulation in multi-hop wiretap ad-hoc networks", *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1322-1334, 2021.

[28] M. Yin, Y. Yang and B. Jiao, "Security-oriented trellis code design for spatial modulation", *IEEE Trans. Wireless Commun.*, vol.20, no.3, pp. 1875-1888, Mar. 2021.

[29] Y. Yang, "Permutation-based transmissions in ultra-reliable and low-latency communications", *IEEE Commun. Lett.*, vol.25, no.3, pp. 1024-1028, Mar. 2021.

[30] R. Hamilton, J. Iyengar, I. Swett, and A. Wilk, "QUIC: A UDP-based secure and reliable transport for HTTP/2", Internet Engineering Task Force (IETF) Network Working Group Internet-Draft, Jul. 2016.

[31] R. Corless, G. Gonnet, D. Hare, D. Jeffrey, and D. Knuth, "On the Lambert W function", *Advances in Computational Mathematics*, vol. 5, pp. 329-359, Jan. 1996.

[32] A. Bhat, S. Aoki and R. Rajkumar, "Tools and methodologies for autonomous driving systems", *Proc. IEEE*, vol. 106, no. 9, pp. 1700-1716, Sep. 2018.

[33] R. Lu, L. Zhang, J. Ni and Y. Fang, "5G vehicle-to-everything services: Gearing up for security and privacy", *Proc. IEEE*, vol. 108, no. 2, pp. 373-389, Feb. 2020.

[34] C. Celes, A. Boukerche, and A. Loureiro, "From mobility traces to knowledge: Design guidance for intelligent vehicular networks", *IEEE Network*, vol. 34, no. 4, pp. 227-233, Jul. 2020.

[35] C. Bormann, *et al.*, "Robust header compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", Internet Engineering Task Force (IETF) RFC 3095, Jul. 2001.

Yuli Yang (S'04-M'08-SM'19) received her Ph.D. degree in Communications & Information Systems from Peking University in July 2007. Since Dec 2019, she has been with the University of Lincoln as a Senior Lecturer in Electrical/Electronic Engineering. From Jan 2010 to Dec 2019, she was with King Abdullah University of Science & Technology, Melikşah University, and the University of Chester on various academic positions. Her industry experience includes working as a Research Scientist with Bell Labs Shanghai, from Aug 2007 to Dec 2009, and an Intern Researcher with Huawei Technologies, from June 2006 to July 2007. Her research interests include modelling, design, analysis and optimization of wireless systems and networks.

Lajos Hanzo (http://www-mobile.ecs.soton.ac.uk, https://en.wikipedia.org/wiki/Lajos_Hanzo) (FIEEE'04, Fellow of the Royal Academy of Engineering (FREng), of the IET and of EURASIP), received his Master degree and Doctorate in 1976 and 1983, respectively from the Technical University (TU) of Budapest. He was also awarded the Doctor of Sciences (DSc) degree by the University of Southampton (2004) and Honorary Doctorates by the TU of Budapest (2009) and by the University of Edinburgh (2015). He is a Foreign Member of the Hungarian Academy of Sciences and a former Editor-in-Chief of the IEEE Press. He has served several terms as Governor of both IEEE ComSoc and of VTS. He has published 1900+ contributions at IEEE Xplore, 19 Wiley-IEEE Press books and has helped the fast-track career of 123 PhD students. He holds the Chair of Telecommunications and directs the research of Next-Generation Wireless at the University of Southampton, UK.