

# Infocommunications Journal

A PUBLICATION OF THE SCIENTIFIC ASSOCIATION FOR INFOCOMMUNICATIONS (HTE)

March 2022

Volume XIV

Number 1

ISSN 2061-2079

## MESSAGE FROM THE EDITOR-IN-CHIEF

Diverse Infocommunication Technologies to Assist  
Heterogeneous Distributed Systems ..... *Pal Varga* 1

## PAPERS FROM OPEN CALL

Predicting the Price of Second-Hand Housing Based on  
Lambda Architecture and KD Tree ..... *Qinghe Pan, Zeguo Qiu, Yaoqun Xu and Guilin Yao* 2

Turbo decoding of concatenated codes based on RS codes using Adapted  
scaling factors ..... *Es-said Azougaghe, Abderrazak Farchane, Said Safi and Mostafa Belkasmi* 11

Determining Hybrid Re-id Features of Vehicles in Videos  
for Transport Analysis ..... *Dávid Papp and Regő Borsodi* 17

A comprehensive survey on the application of blockchain/hash chain  
technologies in V2X communications ..... *Hassan Farran, David Houry, and László Bokor* 24

Optimizing Camera Stream Transport in Cloud-Based Industrial  
Robotic Systems ..... *Marcell Balogh and Attila Vidács* 36

Batch-scheduling Data Flow Graphs with Service-level Objectives  
on Multicore Systems ..... *Tamás Lévai and Gábor Rétvári* 43

Knowgraph-TT: Knowledge-Graph-Based Transit Time Matching in Semiconductor  
Supply Chains ..... *Nour Ramzy, Hans Ehm, Sandra Durst, Konstanze Wibmer, and Werner Bick* 51

## CALL FOR PAPER / PARTICIPATION

CNSM 2022 / 18th International Conference on Network and Service Management  
CNSM 2022, Thessaloniki, Greece ..... 59

## ADDITIONAL

Guidelines for our Authors ..... 60

Technically Co-Sponsored by



**Editorial Board**

**Editor-in-Chief:** PÁL VARGA, Budapest University of Technology and Economics (BME), Hungary  
**Associate Editor-in-Chief:** ROLLAND VIDA, Budapest University of Technology and Economics (BME), Hungary  
**Associate Editor-in-Chief:** LÁSZLÓ BACSÁRDI, Budapest University of Technology and Economics (BME), Hungary

- |  |   |
|--|---|
| JAVIER ARACIL<br>Universidad Autónoma de Madrid, Spain   | LEVENTE KOVÁCS<br>Obuda University, Budapest, Hungary                           |
| LUIGI ATZORI<br>University of Cagliari, Italy  | MAJA MATIJASEVIC<br>University of Zagreb, Croatia                               |
| PÉTER BARANYI<br>Széchenyi István University of Győr, Hungary  | VACLAV MATYAS<br>Masaryk University, Brno, Czech Republic                       |
| JÓZSEF BÍRÓ<br>Budapest University of Technology and Economics, Hungary                                      | OSCAR MAYORA<br>FBK, Trento, Italy  |
| STEFANO BREGNI<br>Politecnico di Milano, Italy   | MIKLÓS MOLNÁR<br>University of Montpellier, France                              |
| VESNA CRNOJEVIĆ-BENGIN<br>University of Novi Sad, Serbia   | SZILVIA NAGY<br>Széchenyi István University of Győr, Hungary                    |
| KÁROLY FARKAS<br>Budapest University of Technology and Economics, Hungary                                    | PÉTER ODRY<br>VTS Subotica, Serbia  |
| VIKTORIA FODOR<br>Royal Technical University, Stockholm  | JAUELICE DE OLIVEIRA<br>Drexel University, USA                                  |
| EROL GELENBE<br>Institute of Theoretical and Applied Informatics Polish Academy of Sciences, Gliwice, Poland | MICHAL PIORO<br>Warsaw University of Technology, Poland                         |
| ISTVÁN GÓDOR<br>Ericsson Hungary Ltd., Budapest, Hungary   | ROBERTO SARACCO<br>Trento Rise, Italy   |
| CHRISTIAN GÜTL<br>Graz University of Technology, Austria   | GHEORGHE SEBESTYÉN<br>Technical University Cluj-Napoca, Romania                 |
| ANDRÁS HAJDU<br>University of Debrecen, Hungary  | BURKHARD STILLER<br>University of Zürich, Switzerland                           |
| LAJOS HANZO<br>University of Southampton, UK   | CSABA A. SZABÓ<br>Budapest University of Technology and Economics, Hungary      |
| THOMAS HEISTRACHER<br>Salzburg University of Applied Sciences, Austria                                       | GÉZA SZABÓ<br>Ericsson Hungary Ltd., Budapest, Hungary                          |
| ATTILA HILT<br>Nokia Networks, Budapest, Hungary   | LÁSZLÓ ZSOLT SZABÓ<br>Sapientia University, Tirgu Mures, Romania                |
| JUKKA HUHTAMÄKI<br>Tampere University of Technology, Finland   | TAMÁS SZIRÁNYI<br>Institute for Computer Science and Control, Budapest, Hungary |
| SÁNDOR IMRE<br>Budapest University of Technology and Economics, Hungary                                      | JÁNOS SZTRIK<br>University of Debrecen, Hungary                                 |
| ANDRZEJ JAJSZCZYK<br>AGH University of Science and Technology, Krakow, Poland                                | DAMLA TURGUT<br>University of Central Florida, USA                              |
| FRANTISEK JAKAB<br>Technical University Kosice, Slovakia   | ESZTER UDVARY<br>Budapest University of Technology and Economics, Hungary       |
| GÁBOR JÁRÓ<br>Nokia Networks, Budapest, Hungary  | SCOTT VALCOURT<br>Roux Institute, Northeastern University, Boston, USA          |
| MARTIN KLIMO<br>University of Zilina, Slovakia   | JÓZSEF VARGA<br>Nokia Bell Labs, Budapest, Hungary                              |
| DUSAN KOČUR<br>Technical University Kosice, Slovakia   | JINSONG WU<br>Bell Labs Shanghai, China   |
| ANDREY KOUCHERYAVY<br>St. Petersburg State University of Telecommunications, Russia                          | KE XIONG<br>Beijing Jiaotong University, China                                  |
|  | GERGELY ZÁRUBA<br>University of Texas at Arlington, USA                         |

**Indexing information**

Infocommunications Journal is covered by Inspec, Compendex and Scopus.  
**Infocommunications Journal is also included in the Thomson Reuters – Web of Science™ Core Collection, Emerging Sources Citation Index (ESCI)**

**Infocommunications Journal**

Technically co-sponsored by IEEE Communications Society and IEEE Hungary Section

**Supporters**

FERENC VÁGUJHELYI – president, Scientific Association for Infocommunications (HTE)

**Editorial Office** (Subscription and Advertisements):  
 Scientific Association for Infocommunications  
 H-1051 Budapest, Bajcsy-Zsilinszky str. 12, Room: 502  
 Phone: +36 1 353 1027  
 E-mail: info@hte.hu • Web: www.hte.hu

**Articles can be sent also to the following address:**  
 Budapest University of Technology and Economics  
 Department of Telecommunications and Media Informatics  
 Phone: +36 1 463 4189, Fax: +36 1 463 3108  
 E-mail: pvarga@tmit.bme.hu

**Subscription rates for foreign subscribers:** 4 issues 10.000 HUF + postage

Publisher: PÉTER NAGY

HU ISSN 2061-2079 • Layout: PLAZMA DS • Printed by: FOM Media

# Diverse Infocommunication Technologies to Assist Heterogeneous Distributed Systems

Pal Varga

**W**E live in the age of infocommunication technology boom, where research results are applied in various fields. This first 2022 issue of the Infocommunications Journal presents a colorful blend of technologies used in various fields, most of which can be categorized as distributed systems. The range of application areas is wide: from predicting housing prices through transportation, from cloud-based robotic control through multi-core batch-scheduling to semiconductor supply chains.

Let us have a brief overview of the articles included in the current, first 2022-issue of the Infocommunications Journal.

Qinghe Pan and his co-authors designed and implemented a system based on Lambda architecture to predict the price of second-hand housing. After comparing the performance of various machine learning algorithms, they chose the KD tree model to predict prices in both real-time and batch processing services. Besides, they further suggested that the nearest  $k$  neighbours can be used as a housing recommending list. Their implementation of Lambda architecture is based on open Apache components such as Kafka, Spark, Cassandra and Flask, making the setup relatively easy to replicate.

In their paper, Es-said Azougaghe et al. studied the effect of various usage parameters of generalized parallel concatenated block codes based on Reed-Solomon (RS) codes. Their simulation results show that the chosen adapted parameters – such as the weighting factor  $\alpha$ , the reliability factor  $\beta$ , the reciprocal value of the extrinsic information delivered by the previous decoder ( $\alpha(p)$ ) are effective in providing good the decoding performance. Furthermore, they show the effect of the number of iteration as well as the multi-block  $M$  and the interleaver structure on the decoding performance.

Dávid Papp and Regő Borsodi presented a lightweight solution on a transportation-related computer vision problem: how to provide accurate annotation of vehicles for transport analysis. They introduce hybrid re-identification features, which combine latent, static, and dynamic attributes to improve tracking. They propose multiple scenarios to calculate the static attributes, from which the desired ones can be selected, based on the given task requirements.

Hassan Farran, David Khoury and László Bokor provided a comprehensive survey on the blockchain/hash chain technologies in Vehicle-to-Everything (V2X) communications. It is clear from their paper that these technologies can play an important role in various aspects of V2X communications, enabling the resolution of many issues, including traceable key negotiation between vehicles, security issues in V2X communication, simplification of the distribution of participant CA, trust authentication between vehicles, and many others.

Cloud-based control of visual-guided robotic systems are the state-of-the-art in the domain of industrial robotic research. One of the many challenges is related to sending a vast amount of sensory data with low latency under limited networking conditions. Marcell Balogh and Attila Vidács propose a general solution for efficient camera stream transportation in cloud robotic systems. The evaluation of their streaming solution shows better performance by one order of magnitude when compared with the industry-standard ROS solution.

In data flow graphs, the nodes represent processing primitives and edges between them describe the control flow. Processing data-flows with a multi-core system can improve performance significantly, but batch-scheduling is a challenge. Batch processing is one of the elemental data processing methods, and Batchy is a state-of-the-art batch-scheduling framework for high-end programmable software switches. In their paper, Tamás Lévai and Gábor Rátvári extend Batchy with a non-trivial task: to leverage parallel execution. They developed and implemented effective control algorithms to be used in practical data flow graph batch-scheduling, and evaluated it in a real 5G use-case.

Supply chains has scheduling challenges as well, among which the proper planning of Transit Time (TT) is a currently interesting one because this would help minimize delays. Transit Time in this context is the time taken to move goods physically between different locations in a supply chain. Nour Ramzy and her co-authors approached the problem through Knowledge Graphs, and applied their solution to the semiconductor industry. By examining the time violations, experts can study how to update the planned transit time concerning actual transit times to create a non-conservative and reliable demand fulfilment. Their approach is called Knowgraph-TT, as it connects actual and planned TT, shows the gaps via applied queries, and enables an optimized update of planned TT.

Infocommunications Journal wishes peace and perseverance in 2022 to all its readers, reviewers, and authors.



**Pal Varga** received his Ph.D. degree from the Budapest University of Technology and Economics, Hungary. He is currently an Associate Professor at the Budapest University of Technology and Economics and also the Director at AITIA International Inc. His main research interests include communication systems, Cyber-Physical Systems and Industrial Internet of Things, network traffic analysis, end-to-end QoS and SLA issues – for which he is keen to apply hardware acceleration and artificial intelligence, machine learning techniques as well.

Besides being a member of HTE, he is a senior member of IEEE, where he is active both in the IEEE ComSoc (Communication Society) and IEEE IES (Industrial Electronics Society) communities. He is Editorial Board member of the Sensors (MDPI) and Electronics (MDPI) journals, and the Editor-in-Chief of the Infocommunications Journal.

# Predicting the Price of Second-Hand Housing Based on Lambda Architecture and KD Tree

Qinghe Pan<sup>1</sup>, Zeguo Qiu<sup>2</sup>, Yaoqun Xu<sup>2</sup> and Guilin Yao<sup>3</sup>

**Abstract**—In this paper a system is designed and implemented to predict the price of second-hand housing. This system based on Lambda architecture can execute prediction in both real-time and batch modes so it can give two kinds of different price predictions that reflect current and historical conditions respectively. The *kNN* related algorithms are used for price prediction. By comparing the performance of brute *kNN*, *kd tree* and *ball tree*, *kd tree* is selected as the price prediction model of the system. In system implementation the *kd tree* model is chosen to predict prices in both real-time and batch services. The *kd tree* model can also recommend housings to user besides price prediction. The experiment shows the effectiveness of our system. Time and space performance of brute *kNN*, *kd tree* and *ball tree* are compared by experiments. And the evaluation metrics of other available machine learning models are compared. The reason of choosing the *kd tree* model is also explained by the experimental results.

**Index Terms**—Lambda Architecture, real-time system, batch system, *kd tree*.

## I. INTRODUCTION

IN recent years the techniques of big data have been developing in industrial and commercial areas. Emerging business requirements continually bring new challenges to software and hardware architectures in the big data industry. One of the most important fields is prediction which can greatly help individuals and enterprises to make decisions. Lambda architecture [1], [2], [3], [4] can help us achieve this goal with its special construction. This architecture consists of three layers: speed, service and batch layers. The speed layer is responsible for real-time computation on streaming data and the batch layer is in charge of batch computation on historical big data. The service layer can provide services to users so one can get computation results based on the latest and historical data at the same time.

Lambda architecture has been deployed in various applications such as recommendation system [5], [6], anomaly detection [7], [8], monitoring system [9] and so on. In this paper it is used to predict the second-hand housing prices in China where the second-hand housing transactions are usually processed in trading agents. The sellers provide their housing information to realtors of the agents who publish the second-hand housing information on websites. The buyers can search on websites to choose the interesting housings. The price of the second-hand housing is a more attracting attribute than other attributes. This paper designs and implements a system to predict the second-hand housing price based on

Lambda Architecture. The system can help realtors to publish reasonable housing price and buyers to learn the current market price. And it can also be used to study the price trend by research institutions, and be used by bankers to evaluate the price of mortgaged housing.

In [1] the similar problem has been studied and an architecture named Alarea has been designed and applied successfully in price prediction of real estate. In [1] the data of real estate transactions on different categories from Spanish Ministry of Development (2004-2016) is used in batch layer. And data from Twitter real-time API is used in real-time layer. The influence of sentiment (like tweets) on housing prediction has been studied in [10], [11], [12]. In [13] and [14], real-time sentiment analysis is particularly studied. Based on big data technology, multiple data sources can be associated together, and more accurate prediction can be provided from both historical and real-time perspectives. Our research is very similar to [1], but in the batch layer and real-time layer, our system uses the same data source. In future research we will consider to absorb other real-time factors and ingredients into our existing system to improve the prediction effect.

The representative models of housing price prediction are AVMs (Automated Valuation Models) [15]. The traditional benchmark for AVMs is the hedonic model based on the theory that the price of an asset is a function of its quantifiable characteristics. Now instead of hedonic models most AVMs use some type of ML technique [16], such as *neural network*, *decision tree*, *random forest*, *SVM regression* and so on. In this paper, *kd(k dimensional) tree* [17], [18] is used as the prediction model in batch and real-time layers. Because *kd tree* resides in memory after it is trained, it can also be used in service layer to provide querying service. Compared to other *kNN* (K Nearest Neighbor) related algorithms such as *brute kNN* and *ball tree* [19], [20], *kd tree* has its advantages in performance. Other models such as *linear regression*, *neural network*, *SVM* and so on can also be used in batch and real-time layers. In experiment the performances of these models are compared with *kd tree*.

Besides Lambda architecture, other architectures or systems with reasonable techniques stack can achieve the same prediction goal. Based on Kafka [21] and Flink [22], Kappa architecture [23] can unify batch processing and real-time computing. Alarea [1] can handle source data in different formats. Compared with Lambda and Kappa architectures, it has equivalent or better quality attributes, so it is very attractive.

The rest of this paper is arranged by the following way. In Section II the characteristics of Lambda architecture are

<sup>1,2,3</sup> School of Computer and Information Engineering, Harbin University of Commerce, Harbin, China.

<sup>1</sup>corresponding author (e-mail: panqh@hrbcu.edu.cn)

described, and Kappa and Alarea architectures are briefly introduced and analyzed. In Section III *kNN*-related models are discussed and the housing data is briefly explored. In Section IV-A the results of experiment are analyzed. It describes the details of components in our system, discusses experiment results and compares the performance and metrics of different models. Lastly, the paper is concluded in Section VI.

## II. RELATED ARCHITECTURES

### A. Lambda architecture

Lambda Architecture is initiated by Nathan Marz [24]. There are three layers in this architecture. The names and functions of the layers are described in Table I. The batch layer is responsible for providing batch views based on the master database. The speed layer executes fast and timely computation that will compensate for high latency to serving layer. This schema can also be depicted in Fig. 1 [24]. In Fig. 1 the new data flows into two places. One direction is to batch layer where the new data will be appended to the master database so when next update the batch views will contain the new data. The other direction is to speed layer where the same new data will be accumulated in this layer to produce real-time views before the next iteration of batch views in serving layer. The functions of Lambda Architecture in full can be summarized by these three equations:

$$\begin{aligned} \text{batch view} &= \text{function}(\text{all data}) \\ \text{real-time view} &= \text{function}(\text{real-time view, new data}) \\ \text{query} &= \text{function}(\text{batch view, real-time view}) \end{aligned}$$

TABLE I

THE NAMES AND FUNCTIONS OF LAYERS IN LAMBDA ARCHITECTURE.

Layers	Function Descriptions
Speed layer	1. Compensate for high latency of updates to serving layer. 2. Fast, incremental algorithms. 3. Batch layer eventually overrides speed.
Serving layer	1. Provide access services to batch views. 2. Updated by batch layers.
Batch layer	1. Store growing master dataset. 2. Compute functions on the dataset. 3. Provide batch views.

### B. Kappa architecture

Kappa architecture [23] is a simplification of Lambda architecture. Compared to Lambda architecture Kappa unifies the batch and real-time layers. The structure of Kappa is shown in Fig. 2. The main reason for the popularity of Kappa is the Apache Kafka [21] and Apache Flink [22] frameworks. Kafka not only acts as a message queue, but also can save historical data for a longer time to replace the batch layer in Lambda architecture. Flink takes an earlier time as the starting point and plays the role of batch processing. At the same time, Flink solves the problem of accuracy of calculation results under the disorder of events. If batch processing is consistent with real-time processing, Kappa is more appropriate. However,

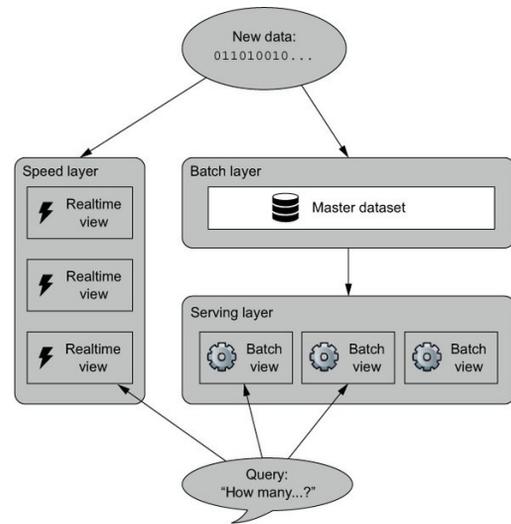


Fig. 1. The illustration of Lambda Architecture.

in some other scenarios, the whole historical data set needs to be processed in batch, so Lambda architecture is more appropriate.

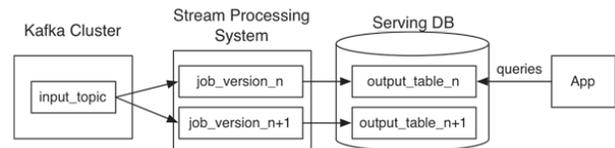


Fig. 2. The illustration of Kappa Architecture.

### C. Alarea architecture

Alarea [1] is an architecture that combines batch processing and real-time processing in two different layers and has been deployed to deal with big data and real-time data in the real estate domain. Compared with Lambda and Kappa, Alarea has the following three advantages.

1. Alarea mixes and integrates heterogeneous data sources.
2. Alarea gives developers the opportunity to decide which layer is better for their purposes.
3. Alarea copes with two kind of data processing and is capable of treating it no matter the timing that they present.

In [1] Lambda, Kappa, and Alarea are also compared based on the four quality attributes, including recoverability, fault tolerance, new data gap and hardware consumption.

## III. MODELS

### A. *kNN* Models

The most effective method to evaluate the price of a house in a building is to observe the prices of the nearest floors above or beneath it. So *kNN* (K Nearest Neighbors) model is an intuitive approach to predict the housing price. Compared to other prediction models *kNN* has two advantages in our application scenario.

Predicting the Price of Second-Hand Housing Based on Lambda Architecture and KD Tree

The first advantage is that  $kNN$  can get more accurate result when the number of sample data is few.  $KNN$  only needs to find out the  $k$  nearest neighbors from the sample data but other prediction models may have to get more data to execute the training. If the dataset only contains two samples( $k=2$ ), for example the housing information of the 4th and 6th floors. If the task is to predict the price of housing on the 5th floor that between 4th and 6th floors in the same building, then  $kNN$  model will get the 4th and 6th floors as its 2 nearest neighbors and predict the 5th housing price by averaging the prices of the two neighbors. But for others models two samples are too small to execute the training.

The second advantage is that  $kNN$  can also generate a recommending housings list by the  $k$  nearest neighbors themselves besides predicting housing price.

Assume that the data set  $X = \{x_1, \dots, x_m\}$ , where  $m$  represents the number of housing records.  $x_i = (x_{i1}, \dots, x_{in})$ ,  $1 \leq i \leq m$ , where  $n$  is the number of fields in each record and  $x_{in}$  is the *price* field. Suppose the data of a house is  $h = (h_1, \dots, h_{n-1}, h_{price})$ , where  $h_1, \dots, h_{n-1}$  are known and  $h_{price}$  is the targeting value that will be predicted by  $kNN$ . The following steps describe the details of prediction by  $kNN$  method.

Step 1. For each data record  $x$  in  $X$ , its *1st* to  $(n-1)$ th fields are used to calculate the Euclidean distance from  $h$  and the results are collected and sorted to find the  $k$  nearest data records. Let  $X_k = (x_{s_1}, \dots, x_{s_k})$  be the vector representing the  $k$  nearest data records by distance from near to far, and the corresponding distance vector is  $Dist_k = (dist_{s_1}, \dots, dist_{s_k})$ .

Step 2. Use the  $n$ th field of the  $k$  data records to estimate the value of  $h_{price}$ , as shown in (1).

$$h_{price} = \frac{\sum_{i=1}^k x_{s_i n}}{k} \tag{1}$$

Where  $x_{s_i n}$  is the  $n$ th field of record  $x_{s_i}$ , i.e., the *price* field.

The variation of  $kNN$  is *Weighted kNN* that considers the influences of different distances on  $h_{price}$  instead of directly averaging the  $n$ th field of the  $k$  data records in (1). The following (2) calculates the weight for each data record based on the distance, where  $\delta$  is the appropriate positive constant.

$$w_{s_i} = e^{-\frac{dist_{s_i}^2}{2\delta^2}} \tag{2}$$

The value range of  $w_{s_i}$  is (0, 1]. The larger the value of  $dist_{s_i}$ , the smaller the value of  $w_{s_i}$ , and vice versa. These characteristics determine that  $w_{s_i}$  is a better choice for representing weight. After the introduction of weights, the calculation of  $h_{price}$  is shown in (3).

$$h_{price} = \frac{\sum_{i=1}^k w_{s_i} x_{s_i n}}{\sum_{i=1}^k w_{s_i}} \tag{3}$$

B. Kd tree and ball tree

The naive or brute  $kNN$  is to choose the  $k$  nearest neighbors from the  $n$  samples by sample-wise comparison. The time complexity to predict price of one housing is  $O(n^2)$ . When

$n$  is large the brute  $kNN$  is not a practicable method. Many improved methods are designed to solve this problem such as the *kd tree* [17], [18], *ball tree* [19], [20], *Hybrid Spill Tree* [25], [26] and so on. The time complexity of these tree-based methods are  $O(\log(n))$ .

The process of building *kd tree* is recursive process that includes two main steps shown in Fig. 3.

Step1. Computes the variance of the dimensions and splits the data at median based on the dimension *Root* that has the highest variance. Let the two split datasets be *Left* and *Right*.

Step2. Repeat the step 1 for *Left* and *Right* until *Left* and *Right* cannot be split.

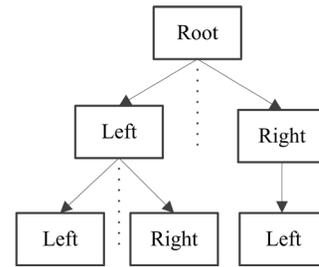


Fig. 3. The construction of *kd tree*.

The process of building *ball tree* is also a recursive process that includes two main steps shown in Fig. 4.

Step1. All data is split into two almost equal sized balls  $ball_A$  and  $ball_B$ .

Step2. Repeat the step 1 for  $ball_A$  and  $ball_B$  until  $ball_A$  and  $ball_B$  cannot be split.

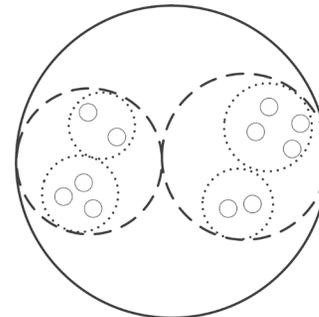


Fig. 4. The construction of *ball tree*.

In this paper the *kd tree* is used in both batch and speed layers in Lambda architecture. The reason will be explained in detail in Section V. After trained by data the *kd tree* will reside in memory and provide prediction service until next iteration. For instance, in batch layer the *kd tree* is trained every day and in speed layer the training frequency is decided by the size of time windows.

IV. EXPERIMENT

A. Data

Usually, the second-hand housing data is crawled from the official websites on the Internet. A distributed crawling system is designed and deployed on cloud to gather data. In such

system each node has its IP address and crawlers on each node run as daemons to crawl data from the websites that own large real-time and historical housing information. By the research purpose the second-hand housing data used in this paper is from one public dataset [27]. This dataset includes 318851 housing information records in Beijing from 2011 to 2017. The data in this dataset will be used as input to our system. It means that the crawlers in our system can directly read records in this dataset as if the data obtained from websites on Internet. This way will save us the time to clean data and help us focus on the system implementation itself. One notation is that housing price in this dataset is the final price but we will ignore this attribute. So the price can be used as the listing price or final price. Each record in dataset consists of 26 fields but only 11 fields are used. The names and meanings of these fields are listed in Table II.

TABLE II  
THE NAMES AND MEANINGS OF FIELDS IN DATASET.

Field names	Meanings
bedroom	the number of bedroom
living room	the number of living room
bathroom	the number of bathroom
kitchen	the number of kitchen
floor	the height of the house
ladder ratio	the number of ladders a resident have on average
square	the square of house
subway	not near to subway(0) and near to subway(1)
lat	the latitude of house
lng	the longitude of house
price	the average price by square

For example the list [3, 2, 2, 1, 6, 0.5, 102.66, 0, 39.873867, 116.66589] consists of data items corresponding to the fields in table. For training *kd tree* the data must be processed further. Firstly, the dataset should be split into *X* matrix and *Y* targeting vector. The *Y* consists of all housing prices in *price* field and the *X* matrix consists of data in other fields except for *price* field. Secondly, the *X* matrix should be normalized before it is used to train *kd tree*. For example the *subway* field should be one-hot encoding and other nine fields in the *X* matrix should be standardized by using max-min or std-dev methods. The data is explored and the results are shown in Fig. 5.

The Fig. 6 shows the relationship between the *price* field and the *longitude* and *latitude* fields. It can be seen that the closer to the central urban area, the higher the average housing price, the more housings for sale, and the more frequent transactions.

The Pearson correlation coefficient between data fields is shown in the Fig. 7. It can be seen that the *bedroom*, *living room* and *bathroom* fields are positively correlated and have high correlation coefficients. At the same time, there are also strong positive correlations between the *square* field and the *bedroom*, *living room* and *bathroom* fields. The *subway* field has a strong positive correlation with the *price* field. Because the data comes from Beijing and it belongs to the economic center, whether there is a subway has a great impact on the housing price. The *bedroom*, *living room* and *bathroom* fields

have negative correlations with the *price* field, because usually the total price of houses with large area is higher, which makes it more difficult to sell, resulting in lower the *price* field.

B. System structure

In this experiment the main frameworks used in our system are described in Table III. The concise experiment topology is shown in Fig. 8. Both Apache Flume and Apache Kafka are distributed, reliable, and available components. They are widely used in big data processing. In this research they are combined together to implement Lambda architecture. There are four reasons for this combination.

1. In production environment because there are many real-time data sources, it is not convenient to build many Kafka clients to publish data to central topics with one topic per activity type. So, usually in implementation the real-time data is delivered to Flume firstly.

2. There are many interceptors in Flume that can be used to filter and clean data. It is more convenient to process data in Flume than in Kafka.

3. When Flume is connected to Spark streaming directly without Kafka, if the speed of data flowing in is faster than the speed of data processing in Spark streaming then the data that cannot be processed in time will be lost. It can be solved by using Kafka between Flume and Spark streaming. Kafka likes a cache that can store data over a period of time.

4. Besides connecting Flume to Spark streaming, Kafka can also provide data to Cassandra for persistent storage. So, Kafka is the core component in our implementation of Lambda architecture.

There are two crawlers *c1* and *c2* for housing data collecting. As described in IV-A the *c1* and *c2* read records in dataset as if they crawl them from websites on Internet. The *c1* and *c2* put their data into *f1* and *f2* respectively. The *f1* and *f2* put their data into *f3*. The *f3* directly connects to Kafka. Spark streaming system fetch data from Kafka based on the time window. The fetched data flows into two directions. In one direction the data is pushed to speed layer and in this layer the *kd tree* model is trained by the data. In the other direction the same data is appended to the existing distributed Cassandra database and in the batch layer the *kd tree* is trained by all or partial data in Cassandra. For speed layer the model training frequency is decided by the size of time window which can be set half an hour, one hour, or two hours and so on. The model training frequency for batch layer can be set one day, two days and so on. Based on speed layer the real-time service can be built which can provide price prediction and house recommending based on the coming data in the time window. The batch service built on batch layer can do the same task based on the historical data. The results from real-time and batch services can support decision making from different perspectives.

C. Experimental Results

We can test the effectiveness of price prediction and housing recommendation by submitting some data to the real-time and batch service respectively. For example, after running a

Predicting the Price of Second-Hand Housing Based on Lambda Architecture and KD Tree

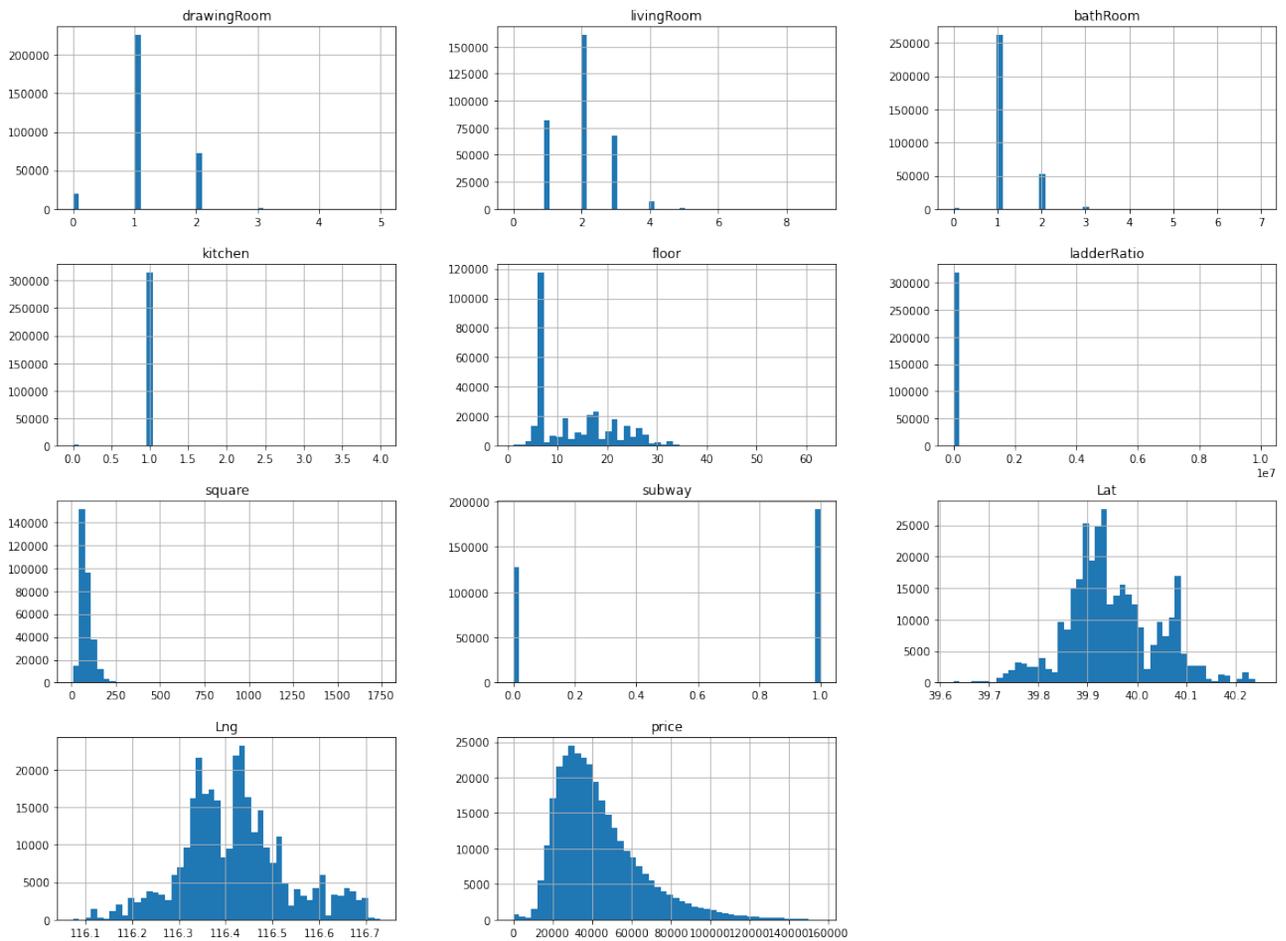


Fig. 5. The exploration of all data fields.

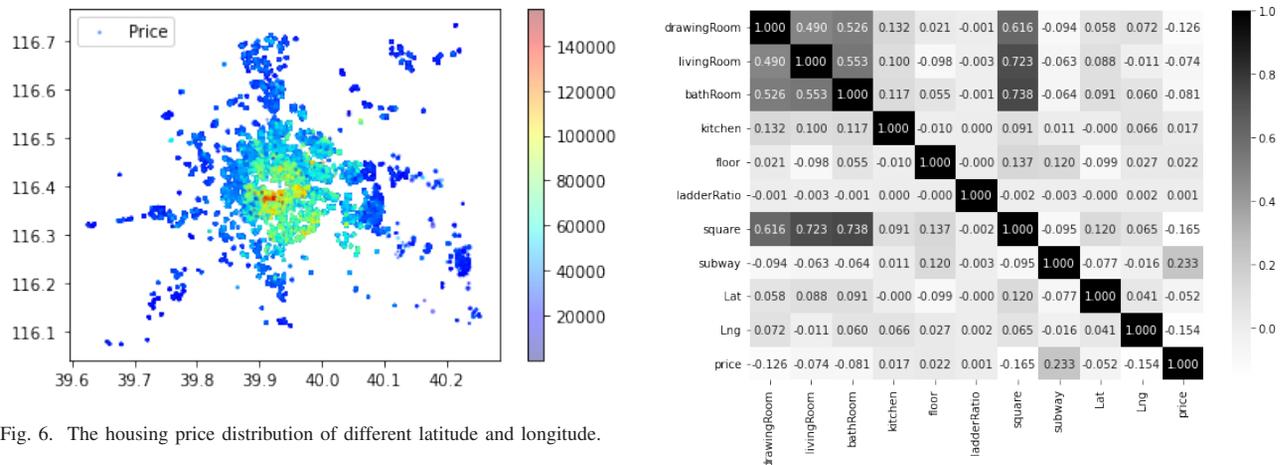


Fig. 6. The housing price distribution of different latitude and longitude.

period of time, there are 153626 records in Cassandra and the *kd tree* has been trained by data in batch layer. The time window of Spark streaming is set to 30s and the *kd tree* in speed layer has been trained by data in this time span. The housing information [3, 2, 2, 1, 6, 0.5, 102.66, 0, 39.873867, 116.66589] is submitted to the real time and batch service respectively. Both services are deployed in Flask and the housing information is post to corresponding service in JSON format.

Fig. 7. The correlation of data fields.

Table IV shows the results returned by batch service when  $k = 10$ . The predicted price is 33194.3 that is the average price of the 10 nearest neighbors. We can also compute the price by *weighted kNN*. The response time of service is 0.001753s that is short enough. The *id* field gives the neighbor index in the 153626 records. So one whole housing information with 26 fields can be obtained by each *id*.

TABLE III  
THE COMPONENTS AND FRAMEWORKS IN THE SYSTEM

Frameworks	Functions	Office websites
Flume	Gather and redirect streaming data.	<a href="http://flume.apache.org/">http://flume.apache.org/</a>
Kafka	Cache the data flow from Flume and provide streaming data to Spark streaming system.	<a href="http://kafka.apache.org/">http://kafka.apache.org/</a>
Spark streaming	1. Be used in speed layer to get streaming data for real-time computation. 2. Store streaming data as real-time views. 3. Append streaming data to distributed Cassandra database.	<a href="https://spark.apache.org/streaming/">https://spark.apache.org/streaming/</a>
Cassandra	Be used in batch layer to store the streaming data and provide batch views.	<a href="https://cassandra.apache.org">https://cassandra.apache.org</a>
Flask	1. Train kd trees based on real-time and batch views respectively. 2. Provide web interfaces for both real-time and batch services.	<a href="https://flask.palletsprojects.com/">https://flask.palletsprojects.com/</a>

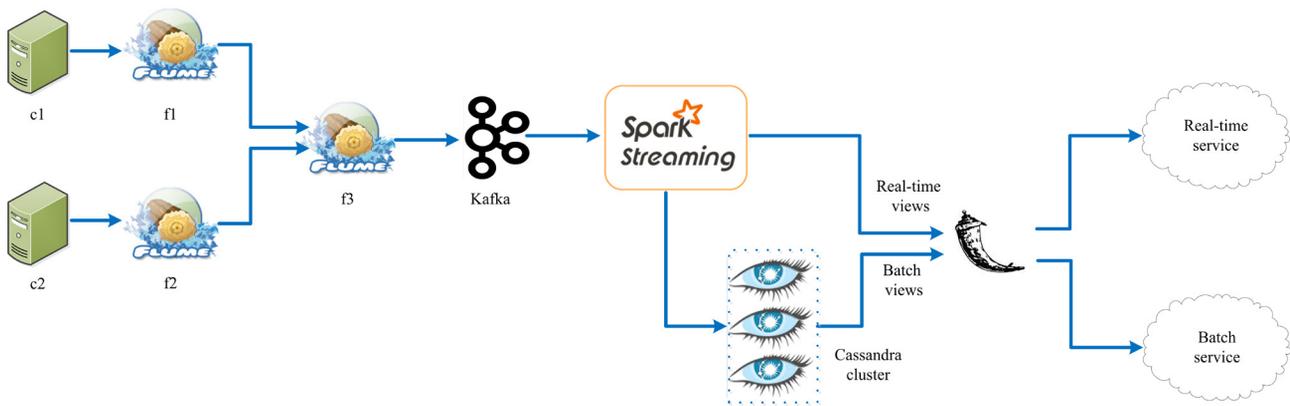


Fig. 8. The structure of price prediction system.

Table V shows the results returned by real-time service when  $k = 10$ . The predicted price is 34910.6 and the numbers in *id* field are the housing indices in the records of the corresponding time window.

For the same input data the batch and real-time services give different price predictions. The reason is that the *kd tree* in batch service is trained by the all or partial historical data, but the *kd tree* in real-time service is trained by the data in time window. In our example the more training data the greater chance to find out the most nearest neighbors. It can be seen from the distance values in the *distance* column in Table IV and V. It is clear that the 10 neighbors got in batch service are closer to input data than in the real-time service.

So in Lambda architecture we can get more accurate prediction in batch service and get more timely prediction in real-time service that can reflect the trend and fluctuation of price in short term. These two kinds of prices can give us the different perspectives to make decision.

V. RESULTS AND DISCUSSION

In this section the time and space performance of *brute kNN*, *kd tree* and *ball tree* are compared. The results show that *kd tree* is better than other two models in our application. We also compare some other machine learning models with the three *kNN* related models.

Figure 9 shows the memory space of different models after training for different  $n$ , here  $n$  is the number of records in

training data. The values of  $n$  are set 20000, 50000, 100000 and 150000. It can be seen that with the growth of  $n$  the memory space of different models increases accordingly. And the memory space of *kd tree* and *ball tree* has more significant increase than *brute kNN*. The reason is that besides the basic data information both the *kd tree* and *ball tree* need additional space to store tree structure information.

Figure 10 shows time cost of different models for different  $n$  during the training process. It can be seen that the *brute kNN* uses less training time than both *kd tree* and *ball tree*. The time cost of *kd tree* and *ball tree* has significant increase as the growth of  $n$  because both tree-based models need more time to construct the tree structures.

So from Fig. 9 and 10 it can be seen that both *kd tree* and *ball tree* need more memory space and training time. But after trained, the tree-based models have shorter prediction time. It can be proved by the following experiment. The testing dataset includes 1000 samples and all models are trained by the same training dataset. When  $k=3, 5, 10, 15, 20, 25$  we compute the averaging prediction time of 1000 samples on *brute kNN*, *ball tree* and *kd tree* respectively. In Fig. 11 it can be seen that *kd tree* have less prediction time than both *ball tree* and *brute kNN*.

Besides the comparison of space and time cost the evaluation metrics RMSE(Root Mean Squared Error), MAE(Mean Absolute Error), R-Squared and MAPE(Mean Absolute Percentage Error) are chosen to evaluate the models. The smaller these metrics the better performance of models. In this paper the  $k$ -fold cross-validation technique is used in order to

Predicting the Price of Second-Hand Housing Based on Lambda Architecture and KD Tree

TABLE IV  
THE BATCH SERVICE RESULTS WHEN  $k = 10$ .

id	bedroom	livingroom	bathroom	kitchen	floor	ladderratio	square	subway	lat	lng	distances
65707	3	2	2	1	6	0.5	105.97	0	39.879143	116.652458	0.158833
151840	3	2	2	1	6	0.5	108.38	0	39.882812	116.661682	0.186459
151820	3	2	2	1	6	0.5	108.70	0	39.882812	116.661682	0.193895
14604	3	2	2	1	6	0.5	110.00	0	39.876873	116.655457	0.222846
14596	3	2	2	1	6	0.333	110.00	0	39.876873	116.655457	0.222846
65726	3	2	2	1	6	0.333	109.92	0	39.879143	116.652458	0.237487
14603	3	2	2	1	6	0.5	110.81	0	39.876873	116.655457	0.242931
112687	3	2	2	1	6	0.5	111.62	0	39.882767	116.669685	0.264404
142464	3	2	2	1	6	0.5	111.65	0	39.883930	116.655563	0.282764
112650	3	2	2	1	6	0.5	92.90	0	39.882767	116.669685	0.284707

TABLE V  
THE REAL-TIME SERVICE RESULTS WHEN  $k = 10$ .

id	bedroom	livingroom	bathroom	kitchen	floor	ladderratio	square	subway	lat	lng	distances
91	3	2	2	1	15	0.3	114.60	0	39.910679	116.594887	1.666115
73	3	3	2	1	4	0.5	144.70	0	39.970848	116.552177	2.611373
59	2	2	1	1	6	0.5	79.88	0	39.934643	116.695198	2.942007
82	2	2	2	1	4	0.028	52.41	0	39.774361	116.512253	3.288918
78	3	2	2	1	6	0.5	129.56	0	40.092632	116.381378	3.437851
41	3	2	2	1	6	0.5	187.16	0	40.110054	116.560165	3.452464
64	3	1	1	1	9	0.5	95.00	0	39.971687	116.544952	3.491309
44	4	2	2	1	6	0.5	158.00	0	40.162060	116.555733	3.619236
84	3	2	2	1	13	0.5	158.93	1	40.082584	116.416153	3.679577
96	3	2	2	1	25	0.375	129.83	0	40.045547	116.430565	3.754350

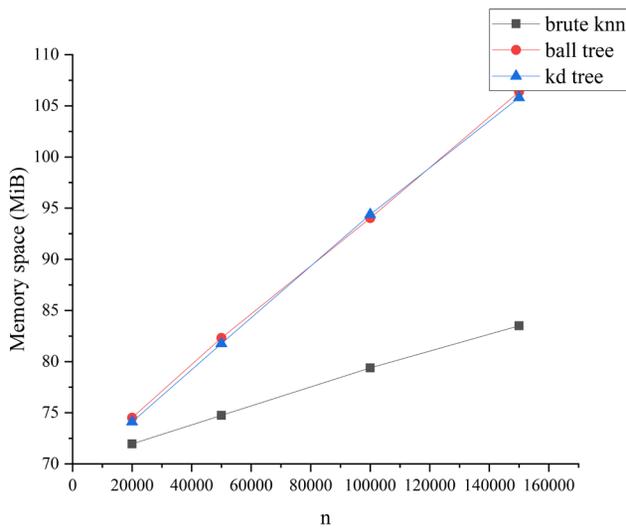


Fig. 9. The memory space of *brute kNN*, *kd tree* and *ball tree* models.

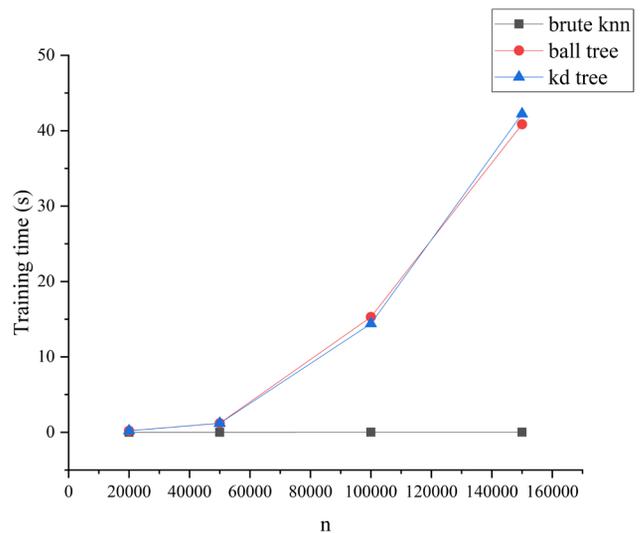


Fig. 10. Training time of *brute kNN*, *kd tree* and *ball tree* models.

compute the average values of RMSE, MAE, R-Squared and MAPE. The  $k$  is set to 5. We compare these metrics between the *brute kNN*, *ball tree* and *kd tree*. Table VI shows the computed results of RMSE, MAE, R-Squared and MAPE of three *kNN* related models for  $k=3, 5, 10, 15, 20, 25$ . It can be seen that there is no obvious difference of performance in all cases.

In the  $k$ -fold cross-validation above, other machine learning models are chosen to complete the same prediction task, including *regression*, *neural network*, *decision tree*, *random forest* and *SVM regression*. The description and parameter information and the average values of RMSE, MAE, R-Squared and MAPE of these models are shown in Table VII. From Table VI and VII we can see that *kNN*-related models have better performance in our application.

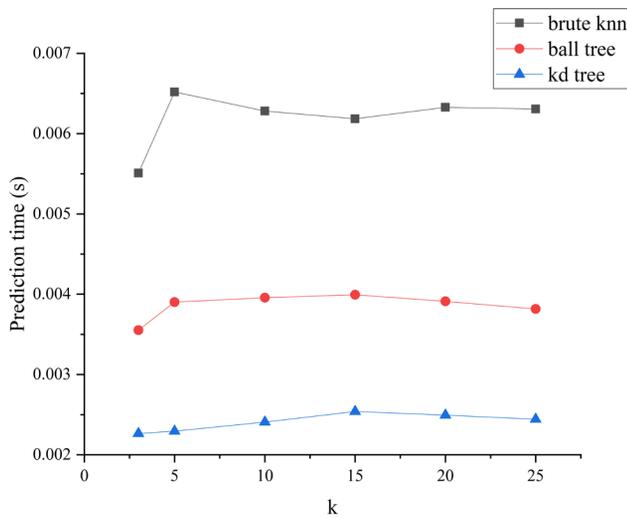


Fig. 11. Predicting time of *brute kNN*, *kd tree* and *ball tree* models.

TABLE VI  
THE EVALUATION METRICS FOR DIFFERENT *k* ON THE *brute kNN*, *ball tree* AND *kd tree* MODELS.

k	Models	RMSE	MAE	R-Squared	MAPE
3	brute kNN	18206.3527	13926.4134	0.4257	364.7942
	ball tree	18202.1706	13926.7102	0.4260	364.7693
	kd tree	18200.8848	13926.0901	0.4260	358.1100
5	brute kNN	17562.3638	13584.7236	0.4656	350.8004
	ball tree	17548.6531	13577.2199	0.4664	346.4767
	kd tree	17553.8938	13579.8747	0.4661	346.4880
10	brute kNN	17129.0225	13382.8892	0.4916	337.2047
	ball tree	17122.4624	13377.2459	0.4920	336.3491
	kd tree	17126.3306	13377.2684	0.4918	335.6577
15	brute kNN	17062.6534	13373.7111	0.4956	347.9658
	ball tree	17061.8768	13377.6418	0.4956	345.4252
	kd tree	17064.7471	13376.5118	0.4955	345.9193
20	brute kNN	17062.9813	13407.4079	0.4956	351.2006
	ball tree	17062.5500	13403.7100	0.4956	352.4062
	kd tree	17067.8417	13410.3385	0.4953	352.5939
25	brute kNN	17086.2347	13437.6975	0.4942	364.1835
	ball tree	17086.8381	13436.8339	0.4942	364.5805
	kd tree	17086.4431	13437.4806	0.4942	364.1689

From these experiments we can see:

1. The *brute kNN*, *kd tree* and *ball tree* have similar evaluation metrics that are better than other machine learning models in our system.

2. Although the *ball tree* and *kd tree* need more memory space and training time than *brute kNN*, the prediction time is less.

3. In our application the *kd tree* has less prediction time than *ball tree*. So we choose *kd tree* as the final model.

### VI. CONCLUSION

The accurate and timely housing price prediction can help individuals and enterprises make reasonable decisions. In this paper we design and implement a system for the prediction of second-hand housing price based on Lambda Architecture. This system can provide housing price prediction by both historical and real-time data, so it can provide two different perspectives on prediction. By analysis of the space and time performance and metrics comparison with other models, the *kd tree* is used as prediction model in both batch layer and speed layer. Besides price prediction, the other benefit of using the *kd tree* is that the nearest *k* neighbors can be used as a housing recommending list. The system can also be used in other applications to provide prediction services.

Other architectures, such as Kappa and Alarea, can also complete the same task. In the future, we will consider porting the existing system to these two frameworks and compare their implementation performance with that of Lambda architecture. Alarea is very attractive because its techniques stack is very similar to that of our system, and it combines the advantages of Lambda and Kappa architectures.

### ACKNOWLEDGMENT

This research was funded by the School-level Scientific Research Project of Harbin University of Commerce (grant number 18XN021), the Heilongjiang Provincial Natural Science Foundation of China (grant number LH2019F044) and the Heilongjiang Provincial Key Laboratory of Electronic Commerce and Information Processing.

### REFERENCES

[1] H. García-González, D. Fernández-Álvarez, J. E. Labra-Gayo, and P. Ordóñez de Pablos, "Applying big data and stream processing to the real estate domain," *Behaviour & Information Technology*, vol. 38, no. 9, pp. 950–958, 2019, doi: 10.1109/TMC.2019.2944829.

[2] A. A. Munshi and Y. A.-R. I. Mohamed, "Data lake lambda architecture for smart grids big data analytics," *IEEE Access*, vol. 6, pp. 40463–40471, 2018, doi: 10.1109/access.2018.2858256.

TABLE VII  
THE EVALUATION METRICS OF OTHER MODELS.

Models	Description	RMSE	MAE	R-Squared	MAPE
regression	Ordinary least squares linear regression with the intercept term.	22572.6282	17639.8669	0.1172	373.0176
neural network	Multilayer perceptron. It has two hidden layers. The first hidden layer contains 5 neurons. The second hidden layer contains 3 neurons. It uses L2 regularization.	22212.5016	17321.2284	0.1451	374.2383
decision tree	It uses mean squared error as feature selection criterion.	20340.7861	15163.8469	0.2831	343.9726
random forest	It uses mean squared error as feature selection criterion. The number of trees in the forest is set to 100.	17193.4087	13176.5261	0.4878	356.1482
SVM regression	Support vector regression with linear kernel.	25372.5650	18480.4802	-0.1154	294.4954

Predicting the Price of Second-Hand Housing Based on Lambda Architecture and KD Tree

[3] M. Gribaudo, M. Iacono, and M. Kiran, "A performance model- ing framework for lambda architecture based applications," *Future Generation Computer Systems*, vol. 86, pp. 1032–1041, 2018, [doi: 10.1016/j.future.2017.07.033](https://doi.org/10.1016/j.future.2017.07.033).

[4] O. Debauche, S. A. Mahmoudi, N. De Cock, S. Mahmoudi, P. Manneback, and F. Lebeau, "Cloud architecture for plant phenotyping research," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 17, p. e5661, 2020, [doi: 10.1002/cpe.5661](https://doi.org/10.1002/cpe.5661).

[5] T. Numnonda, "Areal-time recommendation engine using lambda architecture," *Artificial Life and Robotics*, vol. 23, no. 2, pp. 249–254, 2018, [doi: 10.1007/s10015-016-0424-8](https://doi.org/10.1007/s10015-016-0424-8).

[6] I. Ganchev, Z. Ji, and M. O'Droma, "A conceptual framework for building a mobile services' recommendation engine," in *2016 IEEE 8th International Conference on Intelligent Systems (IS)*. IEEE, 2016, pp. 285–289, [doi: 10.1109/IS.2016.7737435](https://doi.org/10.1109/IS.2016.7737435).

[7] P. Krajsic and B. Franczyk, "Lambda architecture for anomaly detection in online process mining using autoencoders," in *International Conference on Computational Collective Intelligence*. Springer, 2020, pp. 579–589, [doi: 10.1007/978-3-030-63119-2-47](https://doi.org/10.1007/978-3-030-63119-2-47).

[8] D. Vajda, A. Pekár, and K. Farkas, "Towards machine learning-based anomaly detection on time-series data," *INFOCOMMUNICATIONS JOURNAL*, vol. 13, no. 1, pp. 35–44, 2021, [doi: 10.36244/ICJ.2021.1.5](https://doi.org/10.36244/ICJ.2021.1.5).

[9] U. Suthakar, L. Magnoni, D. R. Smith, and A. Khan, "Optimised lambda architecture for monitoring scientific infrastructure," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 6, pp. 1395–1408, 2018, [doi: 10.1109/tpds.2017.2772241](https://doi.org/10.1109/tpds.2017.2772241).

[10] R. M. Croce and D. R. Haurin, "Predicting turning points in the housing market," *Journal of Housing Economics*, vol. 18, no. 4, pp. 281–293, 2009, [doi: 10.1016/j.jhe.2009.09.001](https://doi.org/10.1016/j.jhe.2009.09.001).

[11] P. Dua, "Analysis of consumers' perceptions of buying conditions for houses," *The Journal of Real Estate Finance and Economics*, vol. 37, no. 4, pp. 335–350, 2008, [doi: 10.1007/s11146-007-9084-0](https://doi.org/10.1007/s11146-007-9084-0).

[12] C. Jin, G. Soydemir, and A. Tidwell, "The u.s. housing market and the pricing of risk: Fundamental analysis and market sentiment," *Journal of Real Estate Research*, vol. 36, no. 2, pp. 187–220, 2014, [doi: 10.1080/10835547.2014.12091390](https://doi.org/10.1080/10835547.2014.12091390).

[13] L. Zhang, J. Zhao, and K. Xu, "Emotion-based social computing platform for streaming big-data: Architecture and application," in *2016 13th International Conference on Service Systems and Service Management (ICSSSM)*. IEEE, 2016, pp.1–6, [doi: 10.1109/ICSSSM.2016.7538620](https://doi.org/10.1109/ICSSSM.2016.7538620).

[14] J. Smailović, M. Grčar, N. Lavrač, and M. Žnidarič, "Stream-based active learning for sentiment analysis in the financial domain," *Information sciences*, vol. 285, pp. 181–203, 2014, [doi: 10.1016/j.ins.2014.04.034](https://doi.org/10.1016/j.ins.2014.04.034).

[15] R. Schulz, M. Wersing, and A. Werwatz, "Automated valuation modelling: a specification exercise," *Journal of Property Research*, vol. 31, no. 2, pp. 131–153, 2014, [doi: 10.1080/09599916.2013.846930](https://doi.org/10.1080/09599916.2013.846930).

[16] M. Steurer, R. J. Hill, and N. Pfeifer, "Metrics for evaluating the performance of machine learning based automated valuation models," *Journal of Property Research*, pp. 1–31, 2021, [doi: 10.1080/09599916.2020.1858937](https://doi.org/10.1080/09599916.2020.1858937).

[17] Y. Chen, L. Zhou, Y. Tang, J. P. Singh, N. Bouguila, C. Wang, H. Wang, and J. Du, "Fast neighbor search by using revised kd tree," *Information Sciences*, vol. 472, pp. 145–162, 2019, [doi: 10.1016/j.ins.2018.09.012](https://doi.org/10.1016/j.ins.2018.09.012).

[18] L. Huang S. Nooshabadi, "High-dimensional image descriptor matching using highly parallel kd-tree construction and approximate nearest neighbor search," *Journal of Parallel and Distributed Computing*, vol. 132, pp. 127–140, 2019, [doi: 10.1016/j.jpdc.2019.06.003](https://doi.org/10.1016/j.jpdc.2019.06.003).

[19] L. Wang, B. Yang, Y. Chen, X. Zhang, and J. Orchard, "Improving neural-network classifiers using nearest neighbor partitioning," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2255–2267, 2016, [doi: 10.1109/TNNLS.2016.2580570](https://doi.org/10.1109/TNNLS.2016.2580570).

[20] A. A. Neloy, M. S. Oshman, M. M. Islam, M. J. Hossain, and Z. B. Zahir, "Content-based health recommender system for icu patient," in *International Conference on Multi-disciplinary Trends in Artificial Intelligence*. Springer, 2019, pp. 229–237, [doi: 10.1007/978-3-030-33709-4\\_20](https://doi.org/10.1007/978-3-030-33709-4_20).

[21] B. R. Hiranman et al., "A study of apache kafka in big data stream processing," in *2018 International Conference on Information, Communication, Engineering and Technology (ICICET)*. IEEE, 2018, pp. 1–3, [doi: 10.1109/ICICET.2018.8533771](https://doi.org/10.1109/ICICET.2018.8533771).

[22] D. García-Gil, S. Ramírez-Gallego, S. García, and F. Herrera, "A comparison on scalability for batch big data processing on apache spark and apache flink," *Big Data Analytics*, vol. 2, no. 1, pp. 1–11, 2017, [doi: 10.1186/s41044-016-0020-2](https://doi.org/10.1186/s41044-016-0020-2).

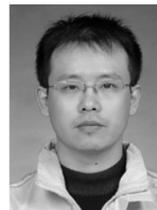
[23] J. Kreps, "Questioning the lambda architecture." 2014. [Online]. Available: <https://www.oreilly.com/radar/questioning-the-lambda-architecture/>

[24] J. Warren and N. Marz, *Big Data: Principles and best practices of scalable realtime data systems*. Simon and Schuster, 2015.

[25] J. Maillou, S. García, J. Luengo, F. Herrera, and I. Triguero, "Fast and scalable approaches to accelerate the fuzzy k-nearest neighbors classifier for big data," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 5, pp. 874–886, 2019, [doi: 10.1109/TFUZZ.2019.2936356](https://doi.org/10.1109/TFUZZ.2019.2936356).

[26] J. Maillou, J. Luengo, S. García, F. Herrera, and I. Triguero, "A preliminary study on hybrid spill-tree fuzzy k-nearest neighbors for big data classification," in *2018 IEEE international conference on fuzzy systems (fuzz-IEEE)*. IEEE, 2018, pp. 1–8, [doi: 10.1109/FUZZ-IEEE.2018.8491595](https://doi.org/10.1109/FUZZ-IEEE.2018.8491595).

[27] Q. Qiu, "Housing price in beijing." 2018. [Online]. Available: <https://www.kaggle.com/ruiqurm/lianjia/version/2>



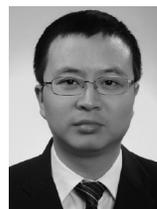
**Qinghe Pan** is currently an associate professor at the School of Computer and Information Engineering of Harbin University of Commerce. His current research interests focus on big data techniques, data mining and machine learning algorithms. He currently teaches in many areas such as Hadoop and Spark architectures, distributed systems and data mining methods.



**Zeguo Qiu** is currently a professor at Harbin University of Commerce. He received PhD in Management Science and Engineering from the Dongbei University of Finance and Economics, Liaoning, China, in 2013. In 2015, he joined the Northeast Asia Service Outsourcing Postdoctoral Workstation and worked on the topic of management decision, information system re-engineering, enterprise technology innovation and e-commerce. His research interests include management decision making, information systems and e-commerce.



**Yaoqun Xu** received the B.S. degree in mathematics from Jilin University, Changchun, China, in 1993, the M.S. degree in mathematics from the Harbin Institute of Technology, Harbin, China, in 1997, and the Ph.D. degree in navigation, guidance, and control from Harbin Engineering University, Harbin, in 2002. He is currently a Professor with the College of Computer and Information Engineering, Harbin University of Commerce. His current research interests include chaotic dynamics, neural networks, and intelligent optimization and decision.



**Guilin Yao** is currently an associate professor in the School of Computer and Information Engineering, Harbin University of Commerce, Harbin, China. His research interests include image processing, computer vision, and machine learning.

# Turbo decoding of concatenated codes based on RS codes using adapted scaling factors

Es-said Azougaghe<sup>1</sup>, Abderrazak Farchane<sup>1</sup>, Said Safi<sup>1</sup> and Mostafa Belkasmi<sup>2</sup>

**Abstract**—Iteratively decoded block turbo codes are product codes that exhibit excellent performance with reasonable complexity. In this paper, a generalization of parallel concatenated block codes (GPCBs) based on RS codes is presented. We propose an efficient decoding algorithm with modifications of the Chase-Pyndiah algorithm is written by using Weighting factor  $\alpha$  and Reliability factor  $\beta$ .

In this work, we studied the effect of diverse parameters such as the effect of various component codes, interleaver size (number of sub-blocks) and number of iterations. The simulation results shows the relevance of the adapted parameters to decode generalized parallel concatenated block codes based on RS codes. The proposed algorithm (MCP) using the adapted parameters performs better than the one using with empirical parameters (CP).

**Index Terms**—RS codes, Chase decoding, Modified Chase-Pyndiah Algorithm, iterative decoding, generalized parallel concatenated codes.

## I. INTRODUCTION

There are many reasons that contributed to the massive interest in product codes. First of all, product codes have noticed a great growth as a result of the introduction of Turbo decoding. In addition to this, the product codes are very identical to concatenated codes as well as to multilevel codes in the sense that almost any solution that works for product codes can easily be compatible to concatenated codes and multilevel codes. Many scholars have suggested different computation methods of soft value for iterative decoding of product codes. A case in example can be found in the works of Pyndiah et al. [1] [2] [3] and [4] who proposed a new iterative decoding algorithm based on Chase decoding [5][6]. The obtained results for product codes based on BCH codes suggested that there is a similarity with those obtained by convolutional turbo codes. Likewise, the generalized parallel concatenated block (GPCB) codes can be seen to be similar to convolutional turbo codes in both encoding and decoding structures. Iterative decoding of concatenated codes uses long powerful codes, and keeps the decoder relatively simple. The length and power of these codes result in safety and durability of application.

<sup>1</sup> Department of Mathematics and Informatics, Laboratory of Mathematics Innovation and Information Technology (LIMATI), Polydisciplinary Faculty, Sultan Moulay Slimane University, Beni Mellal, Morocco (e-mails: {essaidinfo, a.farchane, safi.said}@gmail.com)

<sup>2</sup> Department of Communication Technology, Mohammed V Souissi University, Rabat, National School of Computer Science and Systems Analysis (ENSIAS), Rabat, Morocco (e-mail: Belkasmi@ensias.ma)

Our study is based on RS codes that we decode by using the Modified Chase-Pyndiah algorithm (MCP) [7][8]. Our contribution, in this work, lies in that we tested the application of the Modified Chase-Pyndiah SISO algorithm to decode the GPCB-RS codes based on RS CODES, and we investigated the impact of various component codes, the number of iterations, interleaver size, length and pattern using simulations with an adapted scaling factor to the circumstances of the decoder, namely  $\beta$  and  $\alpha$ .

Relevant studies adapted scaling factor to the circumstances of the decoder. The adapted parameter can outperforms the previous empirical factor, except that the adapted parameter works without re-optimisation after every change in application. This can be noticed in the generalized serial concatenated block codes presented in [9] and parallel concatenated block codes in [10]. Unlike the aforementioned works that applied adapted scaling factor for BCH codes, our study applies this adapted parameter to decode GPCB-RS codes. We can compare our work with several recent works using turbo decoding for convolutional codes or block codes using experimental weighting parameters namely [11][12] [13] and [14], our result gave good performance at the level of the gold decoding gain close to the Shannon limits [15].

The remainder of this paper is structured as follows : Section II presents the encoder structure of the generalized parallel concatenated block codes. In Section III, we present the component decoder. We describe the iterative decoding of the GPCB codes, in Section IV. The simulation results are given in Section V. The last Section concludes this paper.

## II. GENERALIZED PARALLEL CONCATENATED BLOCK CODES (GPCB)

### 1) CONSTRUCTION:

The Fig. 1 illustrates the construction of the generalized parallel concatenated block codes (GPCB). Here a block of  $N = M \times k$  data symbols at the input of the encoder is subdivided to M sub-blocks each of  $k$  symbols. Each  $k$  symbols vector is encoded in order to produce n symbols codeword. The input block is scrambled by the interleaver -denoted by  $\Pi$ - before entering in the second encoder. The codeword of GPCB code, as shown in Fig. 2, consists of the input block followed by the parity check symbols of both encoders. In this contribution, several interleaving techniques were invoked such as random, helical, diagonal and primitive interleaver.

## Turbo decoding of concatenated codes based on RS codes using adapted scaling factors

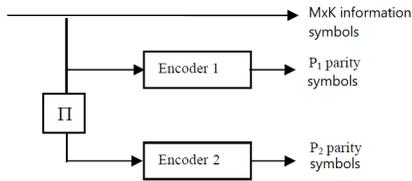


Fig. 1. Encoder structure of GPCB

A systematic GPCB code is based on two component systematic block codes,  $C_1$  with parameters  $(n_1, k)$  and  $C_2$  with parameters  $(n_2, k)$ . Viewing the coding scheme of 1 as single GPCB encoder, the length of the information-word to be encoded by the GPCB code is given by the size of the interleaver  $N = M \times k$ . The first encoder produces  $P_1 = M \times (n_1 - k)$  parity check symbols. The second encoder produces  $P_2 = M \times (n_2 - k)$  parity check symbols. Thus the total number of parity symbols generated by the GPCB encoder is:  $P = P_1 + P_2 = M \times (n_1 + n_2 - 2 \times k)$ . The length of the GPCB codeword is given by:  $L = N + P = M \times (n_1 + n_2 - k)$ . Consequently, the code rate of the GPCB codes can be computed by:  $R = \frac{N}{L} = \frac{k}{(n_1 + n_2 - k)}$ . This implies that the GPCB code rate is independent of the interleaver size  $N$ .

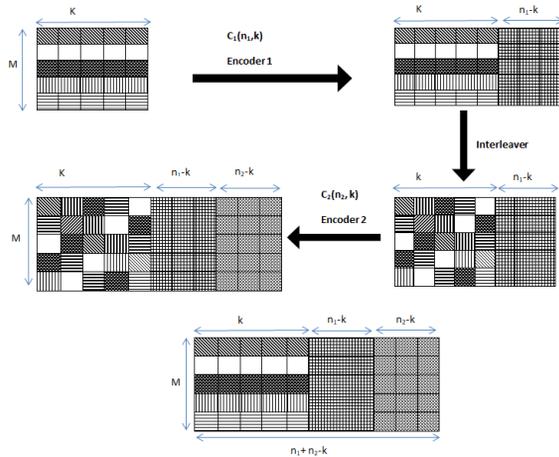


Fig. 2. Systematic GPCB encoding

## 2) SOFT DECODING of RS code:

If we consider the transmission of block coded binary symbols  $\{-1, +1\}$  using BPSK signaling over a Gaussian channel, the sequence  $R$  at the input of the RS decoder has the following expression:  $R = E + B$

where :

$$R = \begin{pmatrix} r_{11} & \cdots & r_{1j} & \cdots & r_{1n} \\ \vdots & \vdots & r_{ij} & \vdots & \vdots \\ r_{m1} & \cdots & r_{mj} & \cdots & r_{mn} \end{pmatrix}$$

is the received sample word,

$$E = \begin{pmatrix} e_{11} & \cdots & e_{1j} & \cdots & e_{1n} \\ \vdots & \vdots & e_{ij} & \vdots & \vdots \\ e_{m1} & \cdots & e_{mj} & \cdots & e_{mn} \end{pmatrix}$$

is the transmitted word,

$$B = \begin{pmatrix} b_{11} & \cdots & b_{1j} & \cdots & b_{1n} \\ \vdots & \vdots & b_{ij} & \vdots & \vdots \\ b_{m1} & \cdots & b_{mj} & \cdots & b_{mn} \end{pmatrix}$$

are Additive White Gaussian Noise (AWGN) samples of standard deviation  $\sigma$ . Decoding the received sequence  $R$  according to the maximum likelihood criteria is given by :

$$D = C^i \quad \text{if} \quad Pr(E = C^i | R) > Pr(E = C^l | R) \quad \forall l \neq i \quad (1)$$

where:

$$C^i = \begin{pmatrix} c_{11}^i & \cdots & c_{1j}^i & \cdots & c_{1n}^i \\ \vdots & \vdots & c_{ij}^i & \vdots & \vdots \\ c_{m1}^i & \cdots & c_{mj}^i & \cdots & c_{mn}^i \end{pmatrix}$$

is the  $i^{th}$  code word of code  $C$  with parameters  $(n, i)$  and

$$D = \begin{pmatrix} d_{11} & \cdots & d_{1j} & \cdots & d_{1n} \\ \vdots & \vdots & d_{ij} & \vdots & \vdots \\ d_{m1} & \cdots & d_{mj} & \cdots & d_{mn} \end{pmatrix}$$

The decision corresponding to maximum likelihood transmitted sequence conditionally to  $R$ .

For received samples corrupted by AWGN, decoding rule (1) is simplified into :  $D = C^i$  if  $|R - C^i|^2 < |R - C^l|^2 \quad \forall l \neq i$  where:

$$|R - C^i|^2 = \sum_{j=1}^n \sum_{f=1}^l (r_{jf} - c_{jf}^i)^2$$

## III. COMPONENT DECODER

We choose as component decoder the Modified Chase-Pyndiah algorithm [7]. This decoder works as follows:

The decoder starts by generating a set of codewords which are in the vicinity of the received vector  $R$ . Then, among those codewords, it selects the nearest codeword from  $R$  in term of Euclidean distance. By doing that it tries to determine the most likelihood codeword. The reliability of the decoded bits is given by the log likelihood ratio (LLR) of the decision  $d_{if}$  which is defined by:

$$LLR_{if} = \ln \left( \frac{Pr(e_{jf} = +1 | R)}{Pr(e_{jf} = -1 | R)} \right) \quad (2)$$

Where  $e_{j f}$  is the binary element in position  $(j, f)$  of the transmitted code word  $E$ ,  $1 \leq j \leq n$  et  $1 \leq f \leq m$ . Using the bayes rule and taking into account that the noise is Gaussian,

$$LLR_{i f} = \log \left( \frac{\sum_{q \in S_j^{+1}} \exp \left( -\frac{|R - C^q|^2}{2\sigma^2} \right)}{\sum_{q \in S_j^{-1}} \exp \left( -\frac{|R - C^q|^2}{2\sigma^2} \right)} \right) \quad (3)$$

where  $S_j^i$  represent the set of codewords having a bit equal to  $i$  ( $i = 1$ ) in position  $j$ .  $LLR_{i f}$  can be approximated, in the case of the AWGN, by: The expression of the can be approximated, in the case of the AWGN, by:

$$LLR_{i f} = \frac{1}{2\sigma^2} \left[ |R - C^{\min(-1)}|^2 - |R - C^{\min(+1)}|^2 \right] \quad (4)$$

Where  $c_{j f}^{\min(+1)}$  and  $c_{j f}^{\min(-1)}$  are two codewords at minimum Euclidean distance from  $R$  with  $c_{j f}^{\min(+1)} = +1$  and  $c_{j f}^{\min(-1)} = -1$ ,  $c_{j f}^{\min(+1)}$  and  $c_{j f}^{\min(-1)}$  are chosen among the subset of code word given by Chase algorithm. By expanding relation 4 we obtain:

$$LLR_{i f} = \frac{2}{\sigma^2} \left( r_{j f} + \sum_{x=1, x \neq j}^n \sum_{z=1, z \neq f}^n r_{x z} c_{x z}^{\min(+1)} \rho_{x z} \right)$$

Where

$$((x, z) \neq (j, f)) \quad \rho_{x z} = \begin{cases} 0, & \text{if } c_{x z}^{\min(+1)} = c_{x z}^{\min(-1)} \\ 1, & \text{if } c_{x z}^{\min(+1)} \neq c_{x z}^{\min(-1)} \end{cases}$$

If we normalize the approximated LLR of  $d_{i f}$  with respect to  $\frac{2}{\sigma^2}$  we obtain:

$$r'_{j f} = \left( \frac{\sigma^2}{2} \right) LLR_{i f} = r_{j f} + w_{j f}$$

The estimated normalized LLR of decision  $d_{i f}$ ,  $r'_{j f}$  is given by input samples  $r_{j f}$  plus  $w_{j f}$  which is independent of  $r_{j f}$ . The LLR of  $r'_{j f}$  is an estimation of the soft decision of the RS decoder.

To compute the normalized  $LLR_{i f}$ , of binary elements at the output RS decoder, we must first select the codeword at minimum Euclidean distance from  $R$ . Let  $C^{\min(+i)}$  be this code word,  $C^{\min(+i)}$  has a binary element  $i$  at position  $(j, f)$  ( $i = \pm 1$ ). Then we look for codeword  $C^{\min(+i)}$  at minimal Euclidean distance from  $R$  among the codeword subset obtained by Chase algorithm.

$C^{\min(-i)}$  must have  $-i$  as binary element at position  $(j, f)$ . If the  $C^{\min(+i)}$  codeword is found, the soft decision  $r'_{j f}$  can be computed using the relation given below:

$$r'_{j f} = \left( \frac{M^{\min(-i)} - M^{\min(i)}}{4} \right) c_{j f}^{\min(i)}$$

Where  $M^{\min(-i)}$  and  $M^{\min(i)}$  represent respectively the  $c_{j f}^{\min(-i)}$  Euclidean distance from  $R$  and  $c_{j f}^{\min(i)}$  Euclidean distance from  $R$ .

Else we use the relation:  $r'_{j f} = \left( \frac{1}{2} \sigma_R + |r_{i j}| \right) c_{j f}^{\min(i)}$  where  $\sigma_R$  is the standard deviation of the decoder input sequence  $R$ .

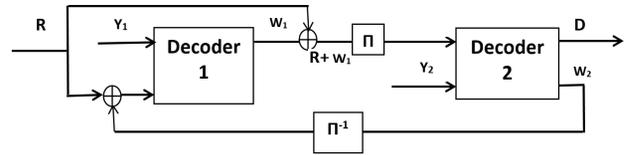


Fig. 3. Iterative decoding structure for the GPCB codes

#### IV. ITERATIVE DECODING OF GPCB CODES

##### A. GPCB decoder

The decoding of the GPCB codes is iterative. The decoder structure is shown in Fig. 3. An iteration consists in using two component decoders serially. The first one uses the systematic information and the first parity check symbols in order to generate extrinsic information  $W$  as in the Modified Chase-Pyndiah algorithm. This extrinsic information is used to update the reliabilities of the systematic information which will be interleaved and feed into the second decoder with the second parity check symbols received from the channel. The second decoder also generates the extrinsic information using Chase-Pyndiah decoder, and then updates the reliabilities of the systematic information for the second time. The updated reliabilities will be desinterleaved and feed again into first decoder, for the next iteration. The process resumes until a maximum number of iterations is reached.

##### B. Parameters $\alpha$ and $\beta$

1) *Weighting factor  $\alpha$* : To reduce the dependency of  $\alpha$  on the product code, the mean absolute value of the extrinsic information  $|W|$  is normalized to one. The evolution of  $\alpha$  with the decoding number is:

$$\alpha = [0.00, 0.01, 0.08, 0.12, 0.16, 0.20, 0.24, 0.28, 0.32, 0.36, 0.40, 0.44, 0.48, 0.52, 0.56, 0.60, 0.61, 0.67, 0.70, 0.72]$$

2) *Reliability factor  $\beta$* : To operate under optimal conditions, the reliability factor should be determined as a function of the BER. For practical considerations, we have fixed the evolution of  $\beta$  with the decoding step to the following values:  $\beta$  with the decoding number is:

$$\beta = [0.56, 0.60, 0.64, 0.68, 0.72, 0.76, 0.80, 0.82, 0.86, 0.88, 0.90, 0.91, 0.93, 0.95, 0.97, 0.99, 0.99, 1.00, 1.00, 1.00]$$

We have determined the values of  $\alpha$  and  $\beta$  empirically [16]. The later parameters play a crucial role to have good performance. So, the better parameters you have the better performance you will gain. Therefore, we should carefully determine these parameters. To obtain good parameters, we choose some condition for which codes are sensitive. Thus, we take the parameter  $M$  equal to 100, and relatively high component code length.

We begin our process by setting the number of iterations in 1, and vary the value of  $\alpha$ , where  $0 < \alpha < 1$ , in order to have good performance, and keep the value of  $\alpha$  which gives the best BER (bit error rate). Next, we vary the value of the

Turbo decoding of concatenated codes based on RS codes using adapted scaling factors

parameter  $\beta$ , where  $0 < \beta < 1$ , in the same way. Once the good parameters are chosen, for the first iteration, we increment the number of iterations, and we look for the good ones for the second iteration. Then we come back without decrementing the number of iterations so as to adjust the parameters  $\alpha$  and  $\beta$  for eventual improvement of the performance. Afterwards, we increment the number of iterations and repeat again the same process until a maximal number of iterations is reached. The coefficients  $\alpha$  and  $\beta$  used in Chase-Pyndiah algorithm are listed in table V.

C. Adapted parameter  $\alpha(p)$

1) *Parameter  $\alpha(p)$* : The role of the parameter  $\alpha(p)$  is vital in the decoding performance. In the works [2][9][16] and [17], this parameter was experimentally predetermined. Its values are chosen such as the  $BER = 10^{-5}$  is attained with the minimum number of iterations. This process is too hard. We have adapted the parameters to the circumstances of the product codes and turbo like-codes to overcome this problem. The following formula gives the expression of  $\alpha(p)$  :

$$\alpha(p) = \frac{1}{\sigma_{W(p-1)}^2}$$

where  $\sigma_{W(p-1)}^2$  denote the variance of the extrinsic information delivered by the previous decoder. The performance obtained by using the adapted parameter  $\alpha(p)$  is comparable to those obtained by the predetermined parameter. Therefore, we don't need to re-optimize this parameter if we change the application.

2) *Parameter  $\beta$* : In case of absence of competitor all the code words have an element  $c_j$  equal to  $d_j$ . This means that all codewords vote for the same decision. In this case the reliability produced by the decoder must follow the fact that all the words agree on the same decision  $d_j$ . This can be translated by the following relation:

$$\gamma_{d_j} = \beta \cdot d_j$$

where

$$\beta = (\sigma_\lambda + |\lambda_j|)$$

where  $\sigma_\lambda$  is the standard deviation of the decoder input sequence  $R$ .

V. RESULTS AND DISCUSSION

In this Section, the performances of generalized parallel concatenated block codes based on RS codes are evaluated. Transmission over the additive white Gaussian noise (AWGN) in channel and binary antipodal modulation are used. We are interested in the information bit error rate (BER) for different signal to noise ratios per information bit  $\frac{E_b}{N_0}$  in dB. There are many parameters which affect the performance of GPCB-RS codes when decoded with iterative decoder. Accordingly, we studied the effects of the following parameters on the decoder performance, namely the number of decoding iterations, the component codes and interleaver size and patterns. The simulation parameters are summarized in this table V:

TABLE I  
SIMULATION PARAMETERS

Parameter	Value
Modulation	BPSK
Environnement	The C Language
Channel	AWGN
Interleaver	Random interleaver (default) Diagonal interleaver Primitive interleaver Helical interleaver
Elementary decoder	Chase-Pyndiah
Iterations	from 1 to 10 (default)
Interleaver size	$1 \times k, 10 \times k, 100 \times k, 300 \times k$

A. Effect of iterations

In this part of simulations we compare between the algorithm of Chase-Pyndiah (*CP*) which uses the empirical parameters and the version of this algorithm which we modified by using the adapted parameters, called algorithm Modified Chase-Pyndiah (*MCP*).

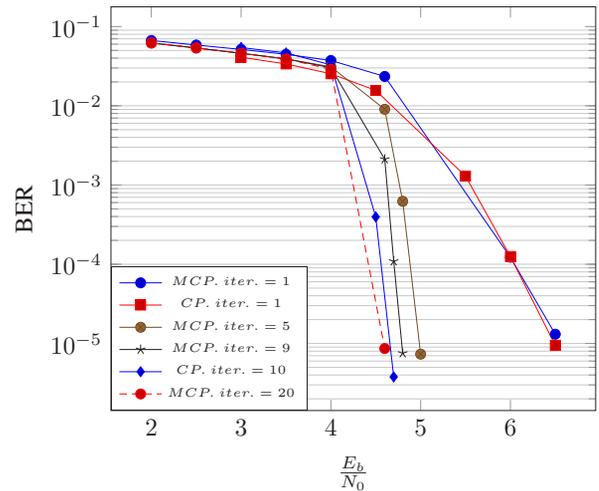


Fig. 4. Effect of iterations on iterative decoding of GPCB-RS(67, 59) Code, with M=100, over AWGN channel

Fig. 4 shows the performance of the code GPCB-RS (67, 59), with  $M = 100$ . This figure shows that the slope of curves and coding gain are improved by increasing the number of iterations. After the 10<sup>th</sup> iteration, the amelioration of the coding gain becomes negligible for Chase-Pyndiah decoder (*CP*), whereas the Modified Chase-Pyndiah decoder (*MCP*) can go up to the 20<sup>th</sup> iteration.

B. Effect of the parameter M

The Fig. 5 shows the effect of the multi-block  $M$ . The gain reaches 1.4dB as we pass from  $M = 1$  to  $M = 10$ , decreases to 0.4dB between  $M = 10$  to  $M = 100$  and becomes negligible beyond  $M = 100$ . This demonstrates how effective is the multi-block  $M$ .

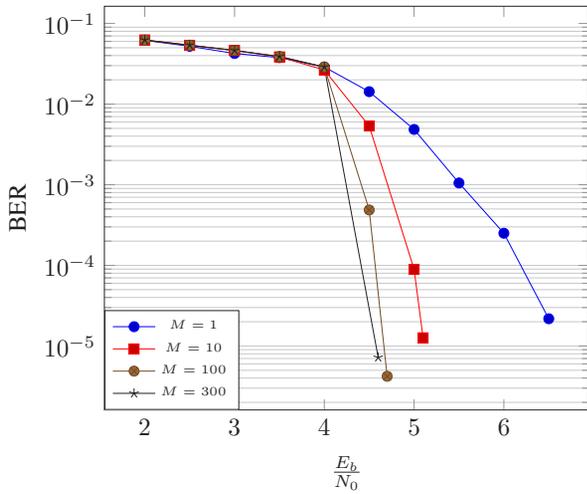


Fig. 5. Effect of the parameter  $M$  on iterative decoding of GPCB-RS(67, 59) code, over AWGN channel

C. Interleaver structure effect

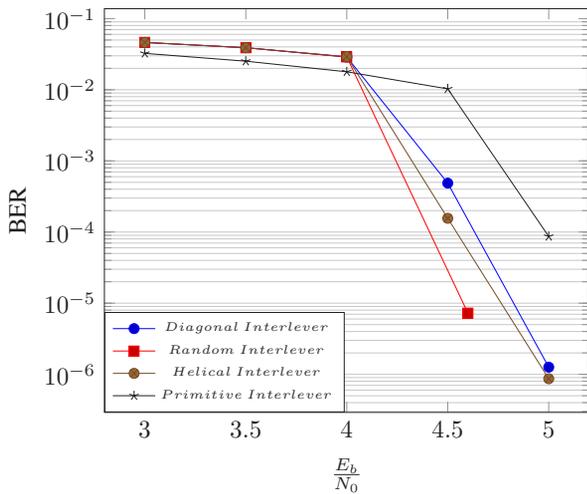


Fig. 6. Interleaver structure effect on Iterative decoding of GPCB-RS(67, 59) Code, with  $M=100$ , over AWGN channel To

To study the influence of the interleaver pattern on the GPCB-RS codes performance, we have evaluated the BER versus  $\frac{E_b}{N_0}$  of the GPCB-RS (67, 59) code using different interleaver structures such as diagonal, helical, primitive and random interleaver with parameter  $M = 100$ . The Fig. 6 shows the performance results. according to this figure we observe that the Random interleaver outperforms the other ones by about  $0.5dB$  at  $TEB = 10^{-5}$ .

D. Effect of multi-blocs

To evaluate the performance of the generalized parallel concatenated block codes, we compare the coding gain at the 20<sup>th</sup> iteration of the following codes GPCB-RS (67, 59), GPCB-RS (131, 123), with the same code rate 0.82 and the parameter  $M = 100$ . The performance is shown in

Turbo decoding of concatenated codes based on RS codes using adapted scaling factors

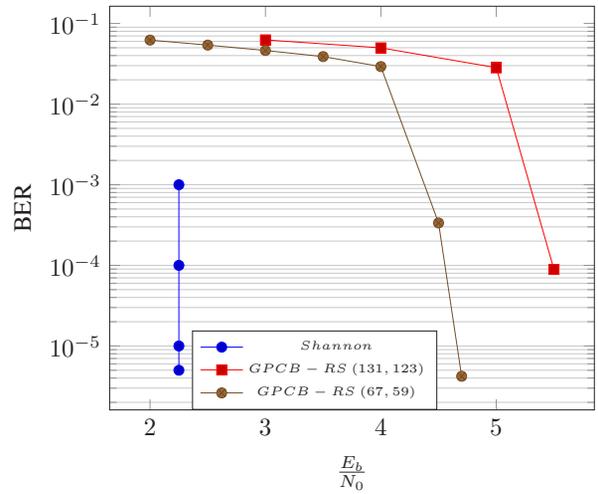


Fig. 7. Effect of multi-blocs on iterative decoding of GPCB codes

Fig. 7. From this figure, we observe that the performance becomes worse with increasing the length of the component code. The GPCB-RS (69,57), GPCB-RS (131, 123) codes are respectively 2.3, 2.8 away from their Shannon limits.

VI. CONCLUSION

In this paper, we have extended the work that has been done to decode generalized concatenated block codes based on BCH codes for RS codes. We have used adapted parameters in order to avoid determining its value empirically. The simulation results show that the adapted parameters are effective, as it can be demonstrated in the asymptotic performance. This work can be extended to produce codes and generalized serially concatenated block based on RS codes adopting the adapted parameters.

REFERENCES

- [1] A. Picart and R. Pyndiah, "Adapted iterative decoding of product codes," in *Global Telecommunications Conference, 1999. GLOBECOM'99*, vol. 5. IEEE, 1999, pp. 2357–2362, doi: 10.1109/GLOCOM.1999.831724.
- [2] O. Aitsab and R. Pyndiah, "Performance of concatenated Reed-Solomon/convolutional codes with iterative decoding," in *Global Telecommunications Conference, 1997. GLOBECOM'97.*, IEEE, vol. 2. IEEE, 1997, pp. 934–938, doi: 10.1109/GLOCOM.1997.638463.
- [3] Seok-Ho Chang, P. C. Cosman, and L. B. Milstein, "Iterative Channel Decoding of FEC-Based Multiple-Description Codes," *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 1138–1152, Mar. 2012, doi: 10.1109/TIP.2011.2169973.
- [4] J. Son, J. J. Kong, and K. Yang, "Efficient decoding of block turbo codes," *Journal of Communications and Networks*, vol. 20, no. 4, pp. 345–353, Aug. 2018, doi: 10.1109/JCN.2018.000050.
- [5] D. Chase, "Class of algorithms for decoding block codes with channel measurement information," *IEEE Transactions on Information theory*, vol. 18, no. 1, pp. 170–182, 1972, doi: 10.1109/IT.1972.1054746.
- [6] P. Wu and N. Jindal, "Performance of hybrid-arq in block-fading channels: A fixed outage probability analysis," *IEEE Transactions on Communications*, vol. 58, no. 4, pp. 1129–1141, 2010, doi: 10.1109/TCOMM.2010.04.080622.
- [7] A. Farchane and M. Belkasmı, "New efficient decoder for product and concatenated block codes," *Journal of Telecommunication*, vol. 12, pp. 17–22, 2012.

Turbo decoding of concatenated codes based on RS codes using adapted scaling factors

[8] J. Cho and W. Sung, "Soft-Decision Error Correction of NAND Flash Memory with a Turbo Product Code," p. 13, 2013, [doi: 10.1007/s11265-012-0698-y](https://doi.org/10.1007/s11265-012-0698-y).

[9] M. Belkasmı and A. Farchane, "Iterative decoding of parallel concatenated block codes," in *Computer and Communication Engineering, 2008. ICCCE 2008. International Conference on*. IEEE, 2008, pp. 230–235, [doi: 10.1109/ICCCE.2008.4580602](https://doi.org/10.1109/ICCCE.2008.4580602).

[10] A. Farchane and M. Belkasmı, "Generalized serially concatenated codes: construction and iterative decoding," *International Journal of Mathematical and Computer Sciences*, vol. 6, no. 2, 2010.

[11] A. D. Cummins, D. G. Mitchell, and D. J. Costello, "Iterative threshold decoding of spatially coupled, parallel-concatenated codes," in *2021 11th International Symposium on Topics in Coding (ISTC)*. IEEE, 2021, pp. 1–5, [doi: 10.1109/ISTC49272.2021.9594231](https://doi.org/10.1109/ISTC49272.2021.9594231).

[12] G. C. Nair, B. Yamuna, K. Balasubramanian, and D. Mishra, "Hardware design of a turbo product code decoder," in *Proceedings of International Conference on Communication, Circuits, and Systems*. Springer, 2021, pp. 249–255.

[13] M. Qiu, X. Wu, J. Yuan, and A. G. i Amat, "Generalized spatially coupled parallel concatenated convolutional codes with partial repetition," in *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2021, pp. 581–586, [doi: 10.1109/ISIT45174.2021.9517979](https://doi.org/10.1109/ISIT45174.2021.9517979).

[14] M. A. Lafta and S. J. Mohammed, "Image transmission in serial and parallel turbo code with bpsk and qpsk modulations," in *2021 1st Babylon International Conference on Information Technology and Science (BICITS)*. IEEE, 2021, pp. 189–193, [doi: 10.1109/BICITS51482.2021.9509907](https://doi.org/10.1109/BICITS51482.2021.9509907).

[15] C. E. Shannon, "A mathematical theory of communication," *Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948, [doi: 10.1002/j.1538-7305.1948.tb01338.x](https://doi.org/10.1002/j.1538-7305.1948.tb01338.x).

[16] A. Farchane, M. Belkasmı, and S. Nough, "Generalized parallel concatenated block codes based on BCH and RS codes, construction and Iterative decoding," *arXiv preprint arXiv:1303.4224*, 2013.

[17] A. G. R. Pyndiah, A. Picart, and S. Jacq, "Near optimum decoding of product codes," *GLOBECOM94*, 1994, [doi: 10.1109/GLOCOM.1994.513494](https://doi.org/10.1109/GLOCOM.1994.513494).



**Es-said Azougaghe** is a PhD student at in the Laboratory of Mathematics Innovation and Information Technology (LIMATI), Polydisciplinary Faculty, Sultan Moulay Slimane University, Morocco. His areas of interest are Information and Coding Theory, cryptography and network security.



**Abderrazak Farchane** is a Professor of computer science in the Polydisciplinary Faculty, Sultan Moulay Slimane University, Morocco. His areas of interest are information coding theory, cryptography, and security.



**Said Safi** is a Professor of computer science in the Polydisciplinary Faculty, Sultan Moulay Slimane University, Morocco. His general interests span the areas of communications and signal processing, estimation, time-series analysis.



**Mostafa Belkasmı** is a Professor at ENSIAS (Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes, Rabat); head of Telecom and Embedded Systems Team at SIME Lab. He had PhD at Toulouse University in 1991(France). His current research interests include mobile and wireless communications, interconnexions for 5G, and Information and Coding Theory.

# Determining Hybrid Re-id Features of Vehicles in Videos for Transport Analysis

Dávid Papp and Regő Borsodi

**Abstract**—The research topic presented in this paper belongs to computer vision problems in the transport application area, where the statistical data of the results give the input for the transport analysis. Although object tracking in a controlled environment could be performed with good results in general, accurate and detailed annotation of vehicles is a common problem in traffic analysis. Such annotation includes static and dynamic attributes of numerous vehicles. Most recent object trackers employ CNNs to compute the so-called re-identification features of the bounding boxes. In this paper we introduce hybrid re-identification features, which combine latent, static, and dynamic attributes to improve tracking. Furthermore, we propose a lightweight solution that could be integrated in a real-time multi-camera tracking system.

**Index Terms**—transport analysis, deep learning, feature extraction, re-identification, multiple object tracking, multi-target multi-camera tracking

## I. INTRODUCTION

The subset of Intelligent Transport Systems allows cooperation [15] among the vehicles and infrastructure, which is called Cooperative Transport System (CTS). CTS systems are designed for cooperative sensing and predicting flow, infrastructure and environmental conditions surrounding traffic, with a goal of improving the safety and efficiency of road transport operations [28]. The efficiency depends on the individual vehicles as well, for example their route planning, as an optimization problem. The uncertainty influences the route; however, a sophisticated model with an appropriate algorithm can handle this uncertainty to find the best route [31]. Finding a good solution for route planning in a transport network is a general problem with arbitrary network type, like a network of buses, a network of tram rails, or any other type of a transport network [30].

Video-based vehicle behavior analysis is done by following and annotating the vehicles across multiple cameras. This requires accurate multi-target multi-camera tracking (MTMC) that must be built upon information coming from single cameras. The detection and tracking of multiple vehicles on a single-camera is frequently referred to as MTSC (multi-target single-camera tracking) or MOT (multiple object tracking). These methods first run an object detector network to detect all object instances, whose bounding boxes are then matched with the trajectories based on previous frames. A critical part of fusing MOTs into MTMC is matching the individual aims to retrieve images from a *gallery* that contain the object of

the same identity as a provided *query* image. Recent solutions for MOT extract feature vectors (so called re-id features) using special CNNs (for example ResNet-IBN variants [23]) and rank gallery images based on their cosine similarity to the query [9], [21], [45]. To improve single-camera tracking, some MOT methods employ the re-id features to help the association between bounding boxes.

Static and dynamic attributes (such as axle number, differentiating signs or velocity) of the vehicles could aid MTMC trajectory matching. Determining these attributes require frame by frame analysis. Passing vehicles usually appear in several, most frequently (but not necessarily) neighboring, frames. Thus, to determine dynamic features, it is required to correctly identify their trajectories including all bounding boxes of the object during their progress in front of the camera. For calculating static features this is not necessary in general, but it could enhance accuracy by using an ensemble decision. The same reasoning holds for a system of multiple cameras, where vehicle re-identification and tracking is preferable.

In this paper we introduce hybrid re-id features, which combines latent features, static and dynamic attributes of the vehicle, and ordinary re-id features. We examine different scenarios to calculate the hybrid re-id feature, from most accurate to most lightweight, that could even be used in a real-time MTMC system.

## II. RELATED WORKS

### A. Transport Analysis

In transport networks different situations can be analyzed, one of which is equilibrium at the case of uncertainty situations, where the uncertainty comes from lack of information. The uncertainty can be represented by Dempster-Shafer theory, an interval-based solution has been developed for handling this situation [29]. In transport analysis different influencing factors of the traffic congestion can be investigated on the roads using uncertain probabilities described by probability intervals [32].

Vehicle behavior analysis consists of some parts, like car-following, lane change maneuvers, velocities of the cars, etc. As the fundamental control strategy of intelligent vehicles, car-following control directly affects vehicle performance. In practical driving, drivers usually predict the behavior of vehicles in the adjacent lane before modulating the driving strategy of the host vehicle [6]. Prediction of lane change maneuvers intended by the driver is solved by an artificial neural network with fusing features modeling the environmental situation [14]. The characterization of vehicles'

Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics (BME), Budapest, Hungary (e-mail: pappd@tmit.bme.hu, rego.borsodi@edu.bme.hu)

Determining Hybrid Re-id Features of Vehicles in Videos for Transport Analysis

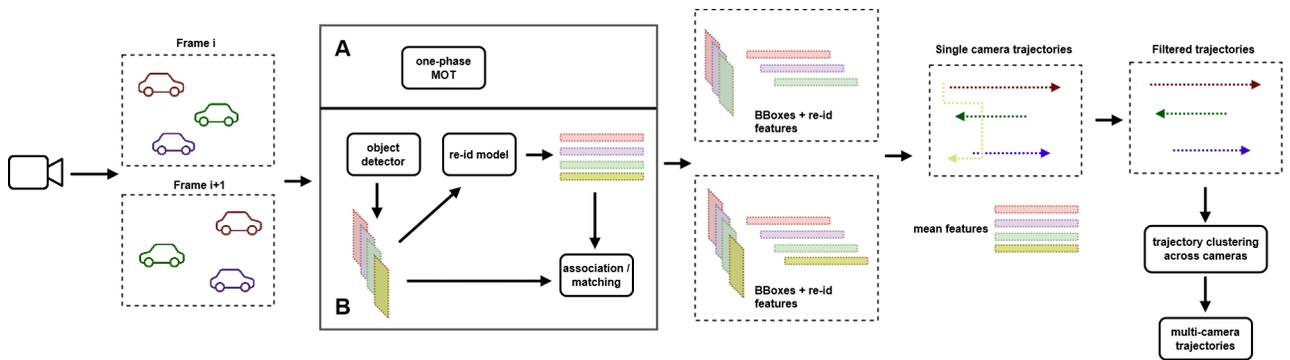


Fig. 1.: Overview of the MTMC tracking process using one-phase single-camera tracking (A) or a two-phase one (B). The yellow trajectory is an erroneous detection; thus, it is filtered out in the single-camera process.

behavior based on their velocities can be modelled by information theory [1]. A vehicle behavior analysis system can be used in traffic jams and under complex weather conditions [26]. To analyze the behavior of vehicles we need determine the static and dynamic features of vehicles in videos, which belongs to the discipline of computer vision.

B. Computer Vision

Most solutions for MOT can be categorized as either *one-phase* or *two-phase* approaches. Two-phase methods first run object detection to get the bounding boxes, then extract (re-id) features of the detected objects. For the association step the SORT [2] method uses Kalman filter [10] to predict object locations and computes the overlap with detected objects. The matching is performed with the Hungarian algorithm [13], with the nodes of the graph being the bounding boxes on neighboring frames. The IOU tracker [3], on the other hand, does the matching based entirely on the overlaps of bounding boxes, without the use of the Kalman filter, thus reaching a higher frame rate.

To improve tracking, some two-phase methods, such as DeepSORT [37] - an improved version of SORT, use deep learning to extract re-id features of the detected objects. The re-id features and the IOU (intersection over union) of bounding boxes are used to compute a cost matrix, which is utilized to do the linking task using Kalman filter and the Hungarian algorithm. This approach delivers decent performance in MOTA (multi-object tracking accuracy), however, the two different deep learning models (for object detection and re-id embedding) do not share architecture and, as the networks are run sequentially, the total inference time is the sum of the individual execution times. Moreover, in crowded scenes, the re-id network must be run separately for tens of bounding boxes, further increasing the total running time.

One-phase approaches merge the object detection and re-id embedding phases into a single network. thus, reaching real-

time performance. The recently proposed FairMOT [42] tracker eliminates anchors and seeks to strike a balance between accurate detection and re-id features. The prediction head is

built on a modified DLA (Deep Layer Aggregation) [41] network on top of a ResNet-34 [7]. The network processes over 25 frames per second on multiple benchmarks [42].

The extraction of re-id features is a crucial part of MOT methods [34], [37], [42]. On the one hand, one-phase trackers such as JDE or FairMOT learn embeddings together with detection by utilizing cross entropy loss or variations of triplet loss [34], [42]. As video datasets with bounding box and identity annotations are scarce, weakly supervised learning was introduced, utilizing images with bounding box annotations, and treating transformed variants of the same objects as the same identity [42]. On the other hand, in a two-phase MOT (scenario B in Figure 1), a separate re-id model is trained for extracting accurate embeddings. Commonly used models for this purpose are IBN-net variants with a ResNet [7] or ResNeXt [38] backbone. *Zhu et al* trained three models for extracting features describing the vehicle, camera, and orientation, then in the final similarity, camera and orientation similarities are subtracted from vehicle similarity to reduce the bias [45]. Given the initial ranking based on similarities, several re-ranking methods have been introduced to improve accuracy, such as the K-reciprocal nearest neighbor method, that favors gallery images having a similar set of k nearest neighbors to the query image [43].

III. MTMC VEHICLE TRACKING

A high-level overview of MTMC process is shown in Fig. 1. Video streams are fed into a one-phase (A) or a two-phase (B) tracker, which both provide bounding boxes, re-id features, and class confidence levels. Tracking algorithms (e.g. DeepSORT) generate a trajectory when no more bounding boxes are appended to it for a given interval of frames [37]. Trajectory filtering is a camera-specific step, when stationary, too noisy, or unnecessary trajectories, e.g. those containing pedestrians or off-road vehicles, are discarded. When a single-camera trajectory is finalized, it is matched with trajectories on other cameras to create multi-camera tracklets. This step is usually done by clustering the mean feature vectors of tracklets [17].

Multi-target multi-camera tracking has been mostly studied as an *offline* task. For example, the test dataset on Track 3 of the AI City challenge [22] contains 20 minutes of traffic videos from 6 non-overlapping cameras. Many solutions first ran MOT

on all cameras, and when all trajectories were available, multi-camera trajectory clustering was deployed [17], [40]. As the locations of cameras were available, spatial-temporal constraints were considered, which greatly reduced the number of possible trajectory matchings. If such constraints are not available, the inter-camera matching can only be done based on vehicle appearance, which becomes increasingly difficult with the growth of the dataset.

Commonly used MOT systems operate in an online manner [3], [34], [37], [42]. In online MTMC, when a single-camera trajectory is generated, it should be immediately connected to an existing multi-camera tracklet or used to initialize a new one. The exact details of this operation heavily depend on the spatial-temporal constraints based on cameras. State-of-the-art single-camera trackers (e.g. FairMOT) also achieve real-time tracking, reaching 25-30 frames per second on the MOT15, MOT16, and MOT17 benchmarks [42]. However, real-time MTMC would require cameras to be well synchronized, and a new vehicle appearing on a camera to be matched with trajectories (or even newly appearing vehicles) from other cameras immediately as it is detected, which would likely deteriorate MTMC accuracy. However, we still consider the running time of the system, including the extraction of static and dynamic features, as it is preferable to be able to process video streams with at least the same speed as they are generated (even if the tracking and extraction do not run strictly in a real-time manner).

IV. MTMC VEHICLE TRACKING USING HYBRID RE-ID

A. Re-id model

For training re-id models, huge datasets, containing multiple images of the same vehicle identities are available (see Table 1) in contrast to the case of one-phase trackers described previously. The VehicleX [39] rendering engine also comes handy in training state-of-the-art re-id models. Firstly, for training an orientation model in the commonly used VOC approach, a dataset annotated with vehicle orientation labels is generated using VehicleX, as creating a real-world dataset containing such labels would require tremendous amount of work. Moreover, extending the dataset with artificial images from VehicleX can alone improve the quality of the features [22].

TABLE I  
VEHICLE RE-IDENTIFICATION DATASETS.

Dataset	#Bboxes	#Identities
VRIC [11], [35]	60K	5622
CityFlow(v2) [33]	313K	880
VeRi-776 [19]	50K	776
VeRi-Wild [20]	416K	40K
VehicleID [18]	221K	26K
VehicleX [39]	$\infty$	1362

B. Static and Dynamic attributes

Fig. 2 shows the set of dynamic and static attributes that are associated with each detected object. Each of these attributes could be calculated from the bounding box directly, or we could

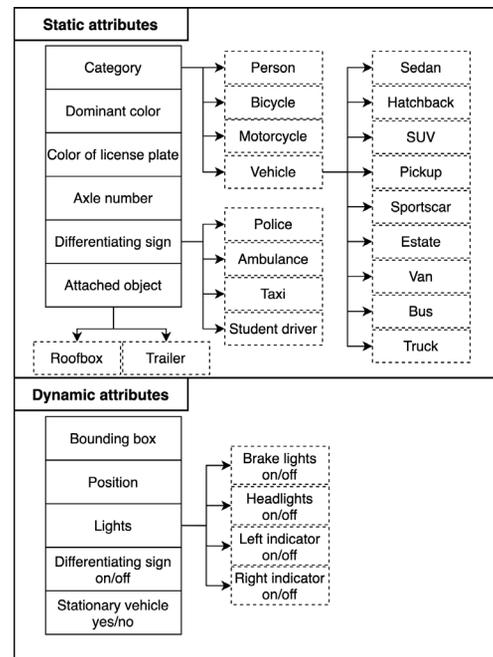


Fig. 2.: Hierarchy of static and dynamic attributes of vehicles; solid line boxes represent the top level

exploit the re-id features, as those are already computed and should be good representations of the objects.

Extracting static and dynamic characteristics of vehicles is generally the output of the annotation process; however, we propose to feed this information back into the MTMC system in order to improve tracking and trajectory matching. Since dynamic features are time dependent, it is required to extract them for all bounding boxes of the object during their progress in front of the camera. Overlapping and well synchronized cameras allow to compensate for false negatives; otherwise, they are replaced with the attributes extracted in neighboring frames. In an ideal situation, static features are also calculated for all bounding boxes, but this is not necessary, because they are constant for the entire trajectory. There are essentially two ways to determine static features:

- Weighted majority vote of frame-by-frame extraction
- Mean of best (high confidence, close to camera) detections

Fig. 3 proposes multiple scenarios for extracting static features. The features can be determined by reusing the re-id features, either by feeding them into a single NN, that has a divided prediction head for each task (C), or by training one classifier for each static feature (E). These classifiers could be NNs, SVMs, GBMs or even random forests. In scenarios D and F, the region of interests (ROIs) from cropped bounding boxes are fed into either a single CNN with a stacked prediction head (D) or into separate CNNs (F). Most likely, scenario F provides the most accurate predictions, however, it requires multiple networks to run for each cropped bounding box on all frames. The process can be optimized by running the models on only

## Determining Hybrid Re-id Features of Vehicles in Videos for Transport Analysis

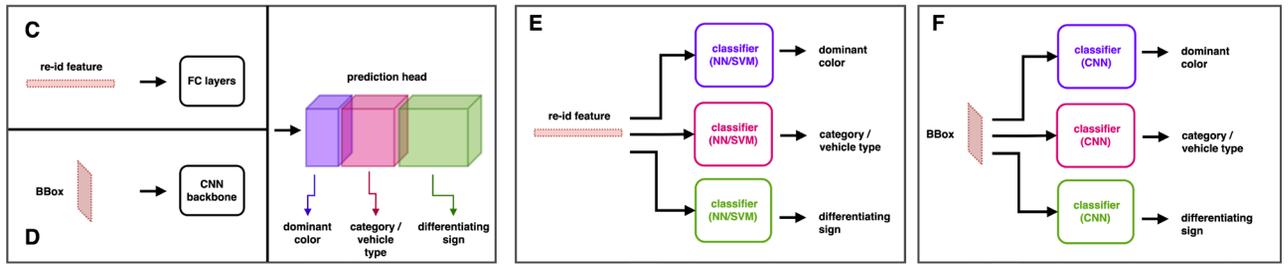


Fig. 3.: Different scenarios for extracting static features: feeding re-id features into a single fully connected network (C), classifying bounding boxes with a single CNN (D), using separate classifiers to extract features from re-id vectors (E), running separate classifiers for cropped bounding boxes (F).

some designated Bboxes, after finalizing a single-camera trajectory. In scenarios C and E, if the static features are determined using the mean re-id features, the inference runs once per trajectory, which is the most lightweight solution.

### C. Fusion of features

Hybrid re-id features are created in two phases, first during single-camera tracking, then during multi-camera trajectory matching. In the former case, temporal static features are merged with the re-id features of bounding boxes; while in the latter, the finalized static features are merged with the mean re-id features. In case of well synchronized cameras, dynamic attributes could also be used in the trajectory matching, but we do not consider this situation. Dynamic attributes such as brake light on/off could help filtering out candidate bounding boxes during the association step. In case of architectures D and F (Fig. 3), dropping the prediction layers results in a feature extractor network, whose features then can be merged with the temporal re-id features. The same holds for trajectory matching, as methods C and E deliver the interpretable static attributes (e.g. license plate color, differentiating sign), which then can be used for filtering purposes. Whereas following architectures D or F gives latent attributes.

Fig. 4 shows the multi-target multi-camera vehicle tracking using hybrid re-id features. It is basically the same process as shown in Fig. 1, and therefore we grayed the closely related but less relevant elements (regarding hybrid re-id features), while omitted the non-related ones. The first part of the process is a two-phase single-camera tracking, which is followed by the multi-camera trajectory filtering and matching. As it can be seen in Fig. 4, hybrid re-id features are used in both; highlighted with blue boxes. Furthermore, classification of basic re-id and mean re-id features produces interpretable attributes that are integrated into the filtering approach; highlighted with green boxes. In case of multi-camera trajectory filtering, these attributes are exclusively static attributes. What is more interesting is that the matching step in single-camera tracking could benefit from the dynamic attributes as well (as mentioned above). We call the dynamic and static attributes on the bounding box level together as temporal attributes.

### D. Style transfer

Image properties like lighting conditions and color distribution heavily depend on the camera, thus, when the images used for training and testing a re-id network were captured by different cameras, feature vector quality decreases. The domain bias is

even more obvious between images generated by VehicleX and real-world images. For the domain adaptation of images SPGAN [4] was used in practice, however SPGAN was designed for images containing people, thus a new network, VTGAN [24] was proposed for vehicles. MixStyle is another domain generalization technique, which does not require to modify training images (in contrast to GAN methods). It was used for training a vehicle re-id baseline by Huyn et al [9]. MixStyle [44] mixes features at the bottom layers of a CNN between instances from different domains, thus improving domain generalization. MixStyle takes an input batch  $x$  and shuffles it to create  $\hat{x}$ . Then standardizes  $x$ , and scales back according to the mixed statistics:

$$\text{MixStyle}(x) = \gamma_{mix} \cdot \frac{x - \mu(x)}{\sigma(x)} + \beta_{mix} \quad (1)$$

where

$$\gamma_{mix} = \lambda\sigma(x) + (1 - \lambda)\sigma(\hat{x}) \quad (2)$$

$$\beta_{mix} = \lambda\mu(x) + (1 - \lambda)\mu(\hat{x}) \quad (3)$$

and  $\lambda$  is a vector, whose elements are sampled from a Beta( $\alpha, \alpha$ ) distribution. If possible, in  $\hat{x}$  and  $x$ , samples at the same positions are from different domains, thus mixing their feature distributions. Two viable options for re-using multiple public re-id datasets are: inserting MixStyle into our network or train a GAN variant (like VTGAN) for each foreign dataset and transform its images into the style of our domain.

### E. Loss function

Choosing appropriate loss functions is critical in training re-id networks. A common technique is to use a weighted sum of two types of losses: *id loss* and *metric loss*. The id loss is measured at the classification layer of the network, while the metric loss is at the feature extraction layer, and its goal is to make features of the same id converge and those from different classes diverge. Triplet loss [25], center loss [36], circle loss [27] and supervised contrastive loss [12] are commonly used as metric losses, while cross entropy is a typical id loss. The weight of id loss and metric loss in the final loss formula can be adapted during the training (in contrast to using constant values), as proposed in [9].

## V. TRAJECTORY FILTERING AND MATCHING

The trajectory filtering step (applied to single-camera trajectories) depends on the constellation of cameras. We consider a crossroad with four cameras pointing inwards. Such a scenario is shown in Fig. 5. We define zones as proposed by

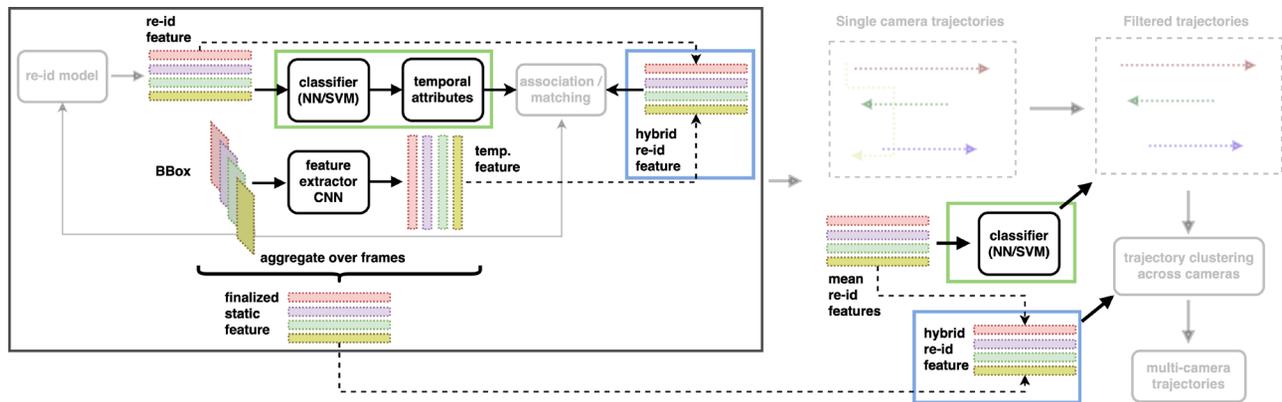


Fig. 4.: Overview of MTMC vehicle tracking using hybrid re-id features. Static attributes and basic re-id features are fused at two parts; highlighted with blue boxes. Classification of basic re-id features gives interpretable attributes for filtering; highlighted with green boxes.

Hsu et al. [8]. If a single-camera trajectory does not start and end in one of the zones or is stationary for a long period (false prediction), then it can be filtered out. When matching trajectories across cameras, only those have to be considered that start and end in the same zone. The constraints, of course, need to be adjusted to the field of view of the cameras, because it is possible that not all cameras have a view of all zones. Another possible constellation is a series of cameras on a highway, with two zones (one direction) or four (two directional) and possible additional ones if the camera has a view on a highway ramp.

The multi-camera trajectory matching step has a strict temporal constraint in the crossroad scenario. If the video streams from cameras are synchronized, or the delays are known, almost exact timestamps are available about vehicles entering and leaving zones, thus the trajectory matching step becomes a simple association step, like the single-camera scenario.

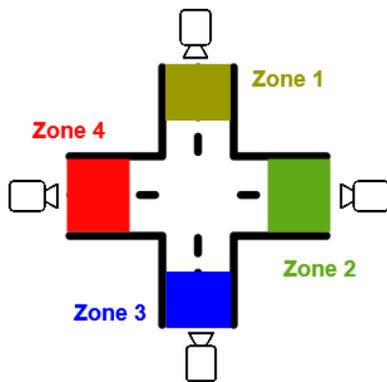


Fig. 5.: Common camera constellation at a crossroad

VI. CONCLUSION

In this paper we elaborated an approach for multi-target multi-camera tracking using hybrid re-id features. The hybrid re-id features are created from static attributes and (basic) re-id features. However, this requires a two-phase tracking method, which is computationally more expensive than one-phase ones; furthermore, the calculation of static attributes comes with

additional computation cost. We propose multiple scenarios to calculate the static attributes, from which the most appropriate one can be selected, based on the requirement of the task, i.e. higher accuracy or higher frame rate.

Our research is currently at the stage of gathering real-world data, which includes multi-camera scenes at crossroads and highways. After the data is collected and cleaned, the proposed methods will be thoroughly tested and evaluated.

ACKNOWLEDGMENT

The research was supported by the Institute of Transport Sciences (KTI) within the Innovative Mobility Program.

The research was supported by the Ministry of Innovation and Technology NRD Office within the framework of the Autonomous Systems National Laboratory Program.

REFERENCES

- [1] Aquino, A. L., Cavalcante, T. S., Almeida, E. S., Frery, A. C., & Rosso, O. A. (2015). Characterization of vehicle behavior with information theory. *The European Physical Journal B*, 88(10), 1-12. doi: 10.1140/epjb/e2015-60384-x
- [2] Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016, September). Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)* (pp. 3464-3468). IEEE. doi: 10.1109/icip.2016.7533003
- [3] Bochinski, E., Eiselein, V., & Sikora, T. (2017, August). High-speed tracking-by-detection without using image information. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 1-6). IEEE.
- [4] Deng, W., Zheng, L., Ye, Q., Kang, G., Yang, Y., & Jiao, J. (2018). Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 994-1003). doi: 10.1109/cvpr.2018.00110
- [5] Farhadi, A., & Redmon, J. (2018, April). Yolov3: An incremental improvement. In *Computer Vision and Pattern Recognition* (pp. 1804-02767). Berlin/Heidelberg, Germany: Springer.
- [6] Guo, Y., Sun, Q., Fu, R., & Wang, C. (2019). Improved car-following strategy based on merging behavior prediction of adjacent vehicle from naturalistic driving data. *IEEE Access*, 7, 44258-44268. doi: 10.1109/access.2019.2908422
- [7] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778). doi: 10.1109/cvpr.2016.90

Determining Hybrid Re-id Features of Vehicles in Videos for Transport Analysis

[8] Hsu, H. M., Huang, T. W., Wang, G., Cai, J., Lei, Z., & Hwang, J. N. (2019, June). Multi-Camera Tracking of Vehicles based on Deep Features Re-ID and Trajectory-Based Camera Link Models. In *CVPR Workshops* (pp. 416-424). doi: 10.1109/tip.2021.3078124

[9] Huynh, S. V. (2021). A Strong Baseline for Vehicle Re-Identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4147-4154). doi: 10.1109/cvprw53098.2021.00468

[10] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. doi: 10.1115/1.3662552

[11] Kanacı, A., Zhu, X., & Gong, S. (2018, October). Vehicle re-identification in context. In *German Conference on Pattern Recognition* (pp. 377-390). Springer, Cham. doi: 10.1007/978-3-030-12939-2\_26

[12] Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C. and Krishnan, D. (2020). Supervised contrastive learning. arXiv preprint arXiv:2004.11362.

[13] Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2), 83-97. doi: 10.1002/nav.3800020109

[14] Leonhardt, V., & Wanielik, G. (2018). Recognition of lane change intentions fusing features of driving situation, driver behavior, and vehicle movement by means of neural networks. In *Advanced Microsystems for Automotive Applications 2017* (pp. 59-69). Springer, Cham. doi: 10.1007/978-3-319-66972-4\_6

[15] Lighthart, J. A., Ploeg, J., Semsar-Kazerooni, E., Fusco, M., & Nijmeijer, H. (2018). Safety analysis of a vehicle equipped with Cooperative Adaptive Cruise Control. *IFAC-PapersOnLine*, 51(9), 367-372. doi: 10.1016/j.ifacol.2018.07.060

[16] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117-2125).

[17] Liu, C., Zhang, Y., Luo, H., Tang, J., Chen, W., Xu, X., Wang, F., Li, H. and Shen, Y.D. (2021). City-scale multi-camera vehicle tracking guided by crossroad zones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4129-4137). doi: 10.1109/cvprw53098.2021.00466

[18] Liu, H., Tian, Y., Yang, Y., Pang, L., & Huang, T. (2016). Deep relative distance learning: Tell the difference between similar vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2167-2175). doi: 10.1109/cvpr.2016.238

[19] Liu, X., Liu, W., Mei, T., & Ma, H. (2016, October). A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In *European conference on computer vision* (pp. 869-884). Springer, Cham. doi: 10.1007/978-3-319-46475-6\_53

[20] Lou, Y., Bai, Y., Liu, J., Wang, S., & Duan, L. (2019). Veri-wild: A large dataset and a new method for vehicle re-identification in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3235-3243). doi: 10.1109/cvpr.2019.00335

[21] Luo, H., Chen, W., Xu, X., Gu, J., Zhang, Y., Liu, C., Jiang, Y., He, S., Wang, F. and Li, H. (2021). An empirical study of vehicle re-identification on the AI City Challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4095-4102). doi: 10.1109/cvprw53098.2021.00462

[22] Naphade, M., Wang, S., Anastasiu, D.C., Tang, Z., Chang, M.C., Yang, X., Yao, Y., Zheng, L., Chakraborty, P., Lopez, C.E. and Sharma, A. (2021). The 5th ai city challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4263-4273). doi: 10.1109/cvprw53098.2021.00482

[23] Pan, X., Luo, P., Shi, J., & Tang, X. (2018). Two at once: Enhancing learning and generalization capacities via ibn-net. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 464-479). doi: 10.1007/978-3-030-01225-0\_29

[24] Peng, J., Wang, H., Zhao, T., & Fu, X. (2019, July). Cross domain knowledge transfer for unsupervised vehicle re-identification. In *2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)* (pp. 453-458). IEEE. doi: 10.1109/icmew.2019.00084

[25] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 815-823). doi: 10.1109/cvpr.2015.7298682

[26] Song, H. S., Lu, S. N., Ma, X., Yang, Y., Liu, X. Q., & Zhang, P. (2014). Vehicle behavior analysis using target motion trajectories. *IEEE Transactions on Vehicular Technology*, 63(8), 3580-3591. doi: 10.1109/tvt.2014.2307958

[27] Sun, Y., Cheng, C., Zhang, Y., Zhang, C., Zheng, L., Wang, Z., & Wei, Y. (2020). Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6398-6407). doi: 10.1109/cvpr42600.2020.00643

[28] Szűcs, G. (2009). Developing co-operative transport system and route planning. *Transport*, 24(1), 21-25. doi: 10.3846/1648-4142.2009.24.21-25

[29] Szűcs, G. (2011). Equilibrium estimation based on unreliable information in transport networks by adaptive simulation. *International Journal of Advanced Intelligence Paradigms*, 3(3-4), 273-285. doi: 10.1504/ijaip.2011.043431

[30] Szűcs, G. (2012). Route planning based on uncertain information in transport networks. *Transport*, 27(1), 79-85. doi: 10.3846/16484142.2012.667835

[31] Szűcs, G. (2015). Decision support for route search and optimum finding in transport networks under uncertainty. *Journal of applied research and technology*, 13(1), 125-134. doi: 10.1016/s1665-6423(15)30011-0

[32] Szűcs, G., & Sallai, G. (2009). Route planning with uncertain information using Dempster-Shafer theory. In *2009 International Conference on Management and Service Science* (pp. 1-4). IEEE. doi: 10.1109/icmss.2009.5302815

[33] Tang, Z., Naphade, M., Liu, M. Y., Yang, X., Birchfield, S., Wang, S., ... & Hwang, J. N. (2019). Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 8797-8806). doi: 10.1109/cvpr.2019.00900

[34] Wang, Z., Zheng, L., Liu, Y., Li, Y., & Wang, S. (2020). Towards real-time multi-object tracking. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16* (pp. 107-122). Springer International Publishing.

[35] Wen, L., Du, D., Cai, Z., Lei, Z., Chang, M.C., Qi, H., Lim, J., Yang, M.H. and Lyu, S. (2020). UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *Computer Vision and Image Understanding*, 193, 102907. doi: 10.1016/j.cviu.2020.102907

[36] Wen, Y., Zhang, K., Li, Z., & Qiao, Y. (2016, October). A discriminative feature learning approach for deep face recognition. In *European conference on computer vision* (pp. 499-515). Springer, Cham.

[37] Wojke, N., Bewley, A., & Paulus, D. (2017, September). Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)* (pp. 3645-3649). IEEE. doi: 10.1109/icip.2017.8296962

[38] Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1492-1500). doi: 10.1109/cvpr.2017.634

[39] Yao, Y., Zheng, L., Yang, X., Naphade, M., & Gedeon, T. (2020). Simulating content consistent vehicle datasets with attribute descent. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16* (pp. 775-791). Springer International Publishing. doi: 10.1007/978-3-030-58539-6\_46

[40] Ye, J., Yang, X., Kang, S., He, Y., Zhang, W., Huang, L., Jiang, M., Zhang, W., Shi, Y., Xia, M. and Tan, X. (2021). A robust MTMC tracking system for AI-City Challenge 2021. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4044-4053). doi: 10.1109/cvprw53098.2021.00456

- [41] Yu, F., Wang, D., Shelhamer, E., & Darrell, T. (2018). Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2403-2412).  
doi: 10.1109/cvpr.2018.00255
- [42] Zhang, Y., Wang, C., Wang, X., Zeng, W., & Liu, W. (2021). Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 1-19.  
doi: 10.1007/s11263-021-01513-4
- [43] Zhong, Z., Zheng, L., Cao, D., & Li, S. (2017). Re-ranking person re-identification with k-reciprocal encoding. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1318-1327). doi: 10.1109/cvpr.2017.389
- [44] Zhou, K., Yang, Y., Qiao, Y., & Xiang, T. (2021). Domain generalization with mixstyle. arXiv preprint arXiv:2104.02008.
- [45] Zhu, X., Luo, Z., Fu, P., & Ji, X. (2020). VOC-ReID: Vehicle re-identification based on vehicle-orientation-camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (pp. 602-603).  
doi: 10.1109/cvprw50498.2020.00309



**Dávid Papp** was born in 1990 in Hungary and he has received MSc in Computer Science (at specialization of media informatics) from Budapest University of Technology and Economics (BME) in 2016. He started his PhD work in 2016 in the field of Computer Science at the same university. His research topic includes artificial intelligence, machine learning, computer vision as well as development of algorithms on these fields (e.g. query strategies for classification of visual contents with active learning). He was awarded twice with the scholarship of New National Excellence Program of the Ministry of Human Capacities, in 2018 and 2019.



**Regő Borsodi** was born in 1997 in Hungary. He is currently a MSc student at Budapest University of Technology and Economics (BME) in Computer Science.

# A comprehensive survey on the application of blockchain/hash chain technologies in V2X communications

Hassan Farran<sup>1</sup>, David Khoury<sup>2</sup>, and László Bokor<sup>3</sup>

**Abstract**—The Vehicle-to-Everything (V2X) technology and protocols are the main cornerstones for advanced transportation and autonomous vehicle applications. V2X has several subsets, including Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication contexts. The main benefit of applying V2X technologies is increased safety by facilitating predicted warnings supporting automated driving and traffic applications. Wirelessly transmitted messages are the information sources; therefore, security is critical in V2X systems. The V2X exchanged messages are sent wirelessly and must fulfill the security requirements, such as integrity, authenticity, and privacy support. The messaging between vehicles and networks must be trusted. Lately, promising and proliferating blockchain/hash chain technologies have been introduced in V2X communications and cope with the cooperative vehicular applications security and related efficiency aspects. This paper provides a comprehensive survey about the V2X use-cases based blockchain/hash chain and introduces the available solutions and methods in this domain.

**Index Terms**—blockchain, hash chain, V2X/C-ITS security/privacy

## I. INTRODUCTION

**C**OOPERATIVE Intelligent Transport Systems (C-ITS) introduce a new ecosystem of linked vehicles, roadside networks, and mobile connectivity valuable to the climate, society, and economy. Vehicle-to-Everything (V2X) creates infrastructures that ensure optimal transport facilities, decrease traffic loads, environmental emissions, and increase road safety and transport quality [1]. Specifically, C-ITS performance relies on V2X communications since it is responsible for sharing data between the underlying communication technologies. This provides input and alerts from on-board sensors, such as the vehicle's current location and speed. V2X is a protocol family designed to exchange messages that include

vehicle information and sensor data from a vehicle to another vehicle or any individual/infrastructure element capable of influencing the vehicle and vice versa. Some of the applications based on V2X are autonomous driving, improving road safety, reducing fuel consumption, and traffic efficiency. V2X systems will make the road safer in decreasing the number of accidents, managing traffic flows, and providing environmental benefits. V2X is a combination of different communication contexts. As shown in Figure 1 below, V2X is based on a cooperative exchange of data between vehicles, and anything else that is Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Network (V2N), Vehicle-to-Pedestrians (V2P) [2], Vehicle-to-Device (V2D) [3], Vehicle-to-Cloud (V2C) [4], Vehicle-to-Home (V2H) [5], or even Vehicle-to-Grid (V2G) [6].

For example, a vehicle that uses a navigation system based on GPS and other sensors can use V2V to indicate vehicle's location, speed, and direction. By broadcasting this information maximum of 10 times per second to the surrounding cars. When a vehicle receives this information, it will calculate the trajectories of the surrounding vehicles. Without entering into a hazardous situation or crash, it will warn the driver by visual alert to make them more aware of what is going around.

V2I is used, for example, as a communication protocol between the vehicle and the traffic lights. It may advise the driver to select the optimal speed to travel through a set of intersections. Furthermore, V2G is an example of a game-changing emerging technology that, along with smart charging, could change the electricity grid. In the case of V2C the ability to provide services from the vehicle maker and other suppliers directly over the Internet is established. V2N is a communication context used, e.g., for warning signs of impending barriers or road jams; and implementing centralized positioning systems [7]. V2D is applied, for example, as a communication method to transmit information between the vehicles and any electronic system to which the vehicle is connected. V2H refers to the exchange of data between vehicles and applications in the home. V2P establishes communication that involves exchanging information between vehicles and pedestrians, such as when the driver sends a message to a pedestrian alerting them to their location and that they are close.

Independently of the applied communication context, the application of V2X includes multiple facets, such as intelligent travel, intelligently linked vehicles, and autonomous driving.

<sup>1</sup> Department of Networked Systems and Services, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Műgyetem rkp. 3., H-1111 Budapest, Hungary. (e-mail: hfarran@hit.bme.hu)

<sup>2</sup> Department of Computer Science and information and communication technology, American University of Science and Technology, Beirut, Lebanon (e-mail: dkhoury@aust.edu.lb)

<sup>3</sup> Department of Networked Systems and Services, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Műgyetem rkp. 3., H-1111 Budapest, Hungary. (e-mail: bokorl@hit.bme.hu)

Various applications have various specifications for latency, durability, throughput, user density, and protection in the V2X environment; protection and autonomous driving system need exceptionally low latency and a safe network environment; therefore, security is the highest priority for V2X [8]. Any vehicular network infrastructure requires comprehensive security mechanisms to enable vehicles and other actors to communicate securely and efficiently.

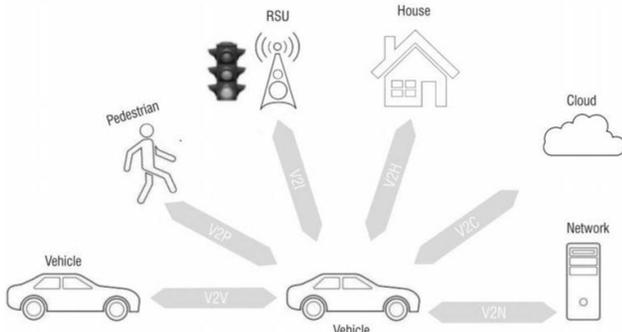


Fig. 1. Vehicle-to-Everything communication contexts.

Two types of V2X communication technologies are currently available: the Wi-Fi-based and the mobile cellular-based solutions (known as V2X and C-V2X, the latter using the 3G, 4G (LTE)/LTE-A, and 5G networks). The benefit of the short-range Wi-Fi-based techniques is low latency compared to the C-V2X networking systems [9]. The additional advantage is that network complexity is significantly lower than mobile cellular technology [10], and the cost is comparatively low [11]. However, the cellular-based system has advantages in targeting far broader areas, pre-existing infrastructure networks, deterministic security, QoS, and improved scalability guarantees [11].

The existing standards for V2X communication are DSRC (Dedicated Short-Range Communication) in the US, and ITS-G5 in Europe (referencing the used 5.9 GHz frequency band) [12].

In DSRC/ITS-G5, the vehicles use on-board units (OBUs) that send messages known as (BSM) Basic Safety Messages in the US that transmit the information about the vehicle, including the speed and location, acceleration, etc. In Europe the (CAM) Common Awareness Messages include similar status and attribute information [13], which have latency less than 100ms with a range of approximately 1600m.

In contrast, the Roadside Unit (RSU) is meant to be wirelessly accessed by the OBUs and usually backhauled by wired technologies. Among the ITS Facilities layer services, RSUs send the Decentralized Environmental Notification Messages (DENM) that include, e.g., alerts for road work. However, DSRC/ITS-G5 could open the door for malicious attacks or cause harm by sending or alerting false safety messages, rendering vehicles unsafe [14]. Both DSRC and ITS-G5 operate in the 5.9 GHz ITS band [15].

Radio technology is a part of the IEEE 802.11 family of standards [16]. IEEE 802.11p was the initial name of the ad hoc Wi-Fi mode of operation IEEE 802.11-2016-OCB (Outside the Context of a Basic Service Set) [17]. Network architectures and

security protocols are specified in IEEE 1609 WAVE [18], and SCMS (Security Credential Management System) [19] on which US DSRC is based, and ITS-G5 with CCMS (EU C-ITS Credential Management System) specifications [20] in the EU.

Communication between vehicles is fundamental because the sensors cannot detect all the risky situations. This makes vehicle networks more vulnerable to various cyber threats that are internal or external attacks. The cooperative system between vehicles can only work when vehicles can trust the neighboring car's messages and the network where it is connected. In order to forge this trust, there are some privacy and security levels the message should pass through.

This paper manifests a comprehensive survey on vehicular communications relying on blockchain networks and technology, which can be used to solve privacy and security issues.

This paper is structured as follows. Section II gives an overview of the types of security attacks in the V2X domain. Section III introduces V2X security basics. Section IV presents background information on blockchain/hash chain technologies. Section V surveys the literature of different V2X topics combined with blockchain/hash chain paradigm. Finally, Section VI concludes the article.

II. TYPES OF V2X SECURITY ATTACKS IN A NUTSHELL

Protection of V2X communication is essential. Vehicular networks are especially susceptible to attacks due to their wireless communication properties. There are six main areas where attention is required to ensure V2X security. These are Validity, Non-Repudiation, Honesty, Confidentiality, Affordability, and Real-time constraints [8].

- Validity: means that the recipient is guaranteed to accept communications from a legitimate source [25].
  - Honesty: all communications should be secured to deter hackers from modifying them and ensure that messages' content is trusted. This ensures that it is protected if the communications' contents are not edited or changed when the message is being sent [26].
  - Affordability: The network must be affordable at all times to transmit and receive messages [27].
  - Real-time constraints: Vehicles drive at high speed, which will demand real-time action in certain situations; otherwise, the outcome will be catastrophic [26].
  - Confidentiality: Community messages sent to all participants should not be decryptable by non-group vehicles. A group message sent to a dedicated member should only be decryptable by the dedicated recipient; other vehicles should not be able to decipher the message [27].
  - Non-repudiation: A sender node can attempt to deny that a message has been sent to escape responsibility for its contents. Non-repudiation is especially useful for the detection of corrupted nodes [27].
1. Attacks on Validity: Sybil attack, also known as Ghost attack, is an intruder that generates several vehicles

- with the exact identification on the lane. It gives delusions to other cars by sending out incorrect signals to benefit this intruder [28].
2. Attacks on Non-Repudiation: Traceability lack of incidents. When an attacker tries to tamper with the database, it must access the majority of the nodes in the network, which is very complex in realistic application scenarios [29].
  3. Attacks on Honesty: GPS spoofing by having nodes that believe they are in various positions; attackers easily trick nodes. This form of assault can be carried out by having incorrect readings on GPS units. It allows attackers to produce a stronger signal than the signal produced by an actual satellite using a GPS satellite emulator[30].
  4. Attacks on Affordability: Denial of Service (DOS) is the most common intrusive assault against availability; an attacker attempts to make tools and facilities inaccessible to users on the network. Either by jamming a physical channel or by "Sleep Deprivation" [31].
  5. Attacks on confidentiality: This can be achieved by the well-known Man in the Middle Attack (MiM), which can intercept the conversation between two other vehicles. This attack is feasible in a vehicle network in various situations. The intruder positions himself between the two pairs of nodes that communicate. The intruder also assumes care of the communication between the two cars. Honesty, validity, and non-repudiation concerns in-vehicle networks and can be violated by the MiM attack [25]. Moreover, it pretends to be the answer of either of them and inserts fake information between them [32].
  6. Attack on Real-time constraints: e.g., period of assault, timing attack [8], Real-time constraints should be enforced since vehicles can travel in and out of a group of a Vehicular Ad Hoc Network (VANET) at random for a brief period of time [32].

To minimize all potential threats that could affect the protection of V2X contact, we need to ensure the effective deployment of adequate security services.

### III. V2X SECURITY BASICS

#### A. V2X messages security

V2X security should operate to check the message's integrity, test that the message's contents did not change, stay stable, and authenticate the sender to check whether the constructed data came from a trusted source. Current V2X standards use a trusted Public Key Infrastructure (PKI) and a trustworthy third-party Certificate Authority (CA) in the US, and a Certificate Policy Authority (CPA) in Europe. PKI uses elliptic curve cryptography (ECC) that facilitates message authentication and integrity [21]. CA and CPA have the highest management authority for issuing vehicle identification details and related certificates, identity verification, and pseudonym management of vehicles. Digital signatures are used to provide the authenticity of the message sent by a vehicle. Both Security

Credentials Management System (SCMS) [22] and C-ITS Credential Management System (CCMS) [20] rely on digital signatures for authentication and validating V2X messages.

#### B. Security and privacy methodology

V2X message transmissions rely on asymmetric key pairs [23]. These public key; private key pairs are used to verify/encrypt and sign/decrypt messages (respectively) to avoid malicious eavesdropping tampering[23]. The public key is known by any user and is extracted and sent to CCMS.

In contrast, the private key is stored securely inside the vehicle and used exclusively for signing transactions and messages. Signed transactions are needed to avoid surveillance, shield the driver's identification, and conceal actual identities. The private and public keys will be changed every short period to achieve privacy [24]. On the other hand, CCMS uses this public key, generates specific vehicle certificates, and signs the certificate using the root CPA. The root CPA is the root of trust for all certificates. Both the vehicle certificates and the root CPA are then sent back to the vehicle. Information shall be given to an accredited PKI auditor for auditing. After being audited, the root CPA application form should be signed with its authorized representative. The CPA appoints the Trust List Manager (TLM), ensuring that all PKI participants have confidence in the TLM's service. The CPA grants permission for the root CA activity and agrees that the TLM will depend on the root CA (s). The TLM generates the European Certificate Trust List (ECTL), which provides all PKI participants with confidence in the accepted root CA's [20] (Figure 2).

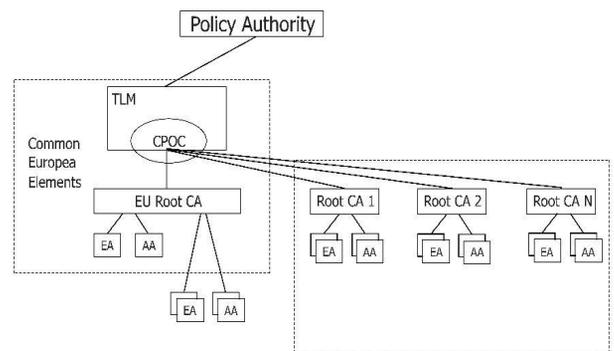


Fig. 2. The C-ITS Trust model architecture.

In the European C-ITS Platform's V2X security proposal implementations [20], the root CA sends the application to the sub-CA Enrolment Authority (EA) and Authorization Authority (AA) entities. In EA and AA, they check the integrity of the message since each message will contain the content, hash, and hash of the previous block (Figure 3). The contents are a set of transactions that could be information about the Vehicle, ID, speed, direction, braking, and even intention, etc. The hash part of the message is a string calculated based on the content. This hash depends on the content, and any alteration in a particular block or the content will eventually break the chain's integrity. However, each part of the message hashed and included the previous part, creating a hash chain for all these parts. After this process, EA and AA transmit their signed request electronically and deliver its application form to root

CA, verifying the request and the received documents. Suppose all checks lead to a positive result. In that case, the root CA issues the corresponding sub-CA certificate and then send the certificate of conformity to C-ITS Point of Contact (CPOC) and TLM. The main task of CPOC and TLM is to verify all documents and the self-signed certificates and send them back to CPA that transmit them to C-ITS (Figure 4). In system management, both TLM and CPOC are a single agency sub-role that operates in the EU CCMS and reports to the Operation Regulatory Body and the Credential Decision Authority [19] [20].

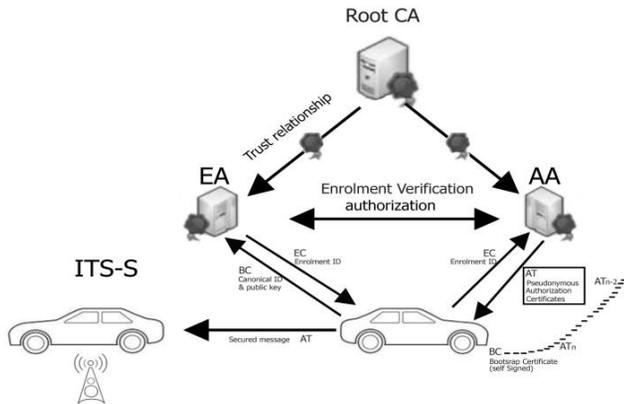


Fig. 3. The C-ITS PKI architecture.

Furthermore, communications between V2X devices are about implemented messages being sent to and from vehicles or Road-side Units (RSUs). After receiving the hash, the vehicle can verify that the contents received from another vehicle have not been modified in transit by calculating and comparing the hash of the content with the one received by the vehicle. In the hash chain, the only packet in the chain that is not integrity protected is the first packet. Subsequently, we can use the private key's role to provide the first packet's integrity and protection to make the integrity safe. The signature calculation involves computing the hash of the message and encrypting the hash with the private key. Also, the benefit of the hash chain is the lightweight computing power requirements compared to other cryptographic algorithms.

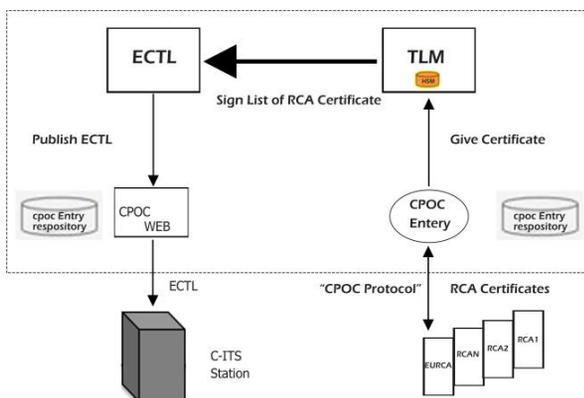


Fig. 4. C-ITS Point of Contract.

IV. OVERVIEW OF THE BLOCKCHAIN TECHNOLOGY

A. Background on the hash chain

The hash chain is the sequential implementation of the hash function encoded over a piece of data. Also, it is a transaction that takes an input length that passes through a hashing algorithm and then gives an output with a fixed length, as seen in Figure 5. Moreover, the hash chain helps to protect the security of sending any message against tampering. To be considered as a secure hash chain, some requirements of the hash function should be satisfied. The first case should be deterministic, implying that the hash function's input gives the same output every time. Second, a quick computation, the hash function, should be able to return the hash's input quickly. The third is preimage resistance; after knowing the output of the hash function  $H(x)$ , it will be impossible to see the input of the function  $(x)$ . Furthermore, any change to the content data would produce a different hash, as stated above. Finally, collision resistance means that two different inputs will have two different hash outputs with high probability.

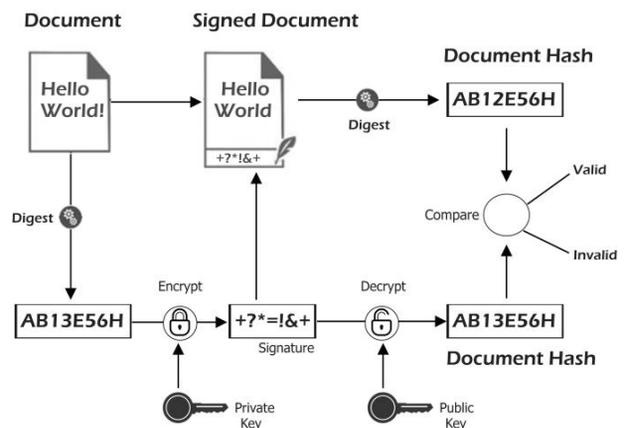


Fig. 5. Digital signature based on cryptographic hash (digest).

The hash chain is a technology used in Bitcoin to transfer digital coins from one individual to another and later the blockchain network's cornerstone. Consequently, all the above process is the key to creating the chain between messages that are called the Blockchain.

One application of the hash chain was the DHT (Distributed Hash Tables) network.

B. Basics of Blockchain

For the first time, the Blockchain was introduced by Satoshi Nakamoto as a peer-to-peer electronic cash system in 2008 [33]. When he published a paper entitled "Bitcoin": "A peer-to-peer electronic cash system," which introduced an innovative and novel way to transfer (send and receive) digital money (called crypto-currency) without the need of going through a trusted third party.

The Blockchain is based on an immutable digital ledger that records all transactions verifiably and consistently. The ledger is replicated across several nodes, which means that no single authority owns or maintains it. The ledger's version validity is established through consensus among the participating nodes, also called miners. The transactions are stored in blocks linked

using cryptography (hence the term Blockchain), explicitly using hash functions: each block stores the previous block's hash, timestamp, and transactions data. Therefore, data on a specific block cannot be altered without changing subsequent blocks, which requires the network's consensus. Each block holds a set of transactions and the previous block's hash, as seen in Figure 6 [34] below.

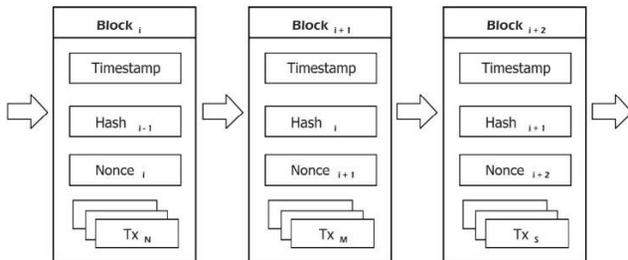


Fig. 6. General Blockchain blocks sequence.

The transactions are stored after verification throughout the network. The verification is via consensus between the nodes; there are different methods to achieve consensus. Information in Blockchain cannot be added or modified until the consensus is reached, making it fraud-proof.

C. Ethereum and decentralized applications (DApps)

The exchange of digital currency was the primary purpose of Bitcoin. Afterward, researchers started developing platforms based on Blockchain technology for running distributed software called decentralized applications (DApps) development through "Smart Contracts". A smart contract is a computer code running over Blockchain capable of exchanging any value a third party needs. They offer the following advantages over the existing computer programs: 1) Autonomous: the network, 2) Manage their execution Trustless: the ledger's version is validated with consensus among nodes, 3) Data safe: the application's data remain permanently in the Blockchain, 4) Transparent: smart contract's code and storage are publicly available. Ethereum is one example of a DApps blockchain network [35].

V. V2X EXTENSIONS/MODIFICATIONS USING BLOCKCHAIN

Blockchain technology started to be considered in V2X systems research areas. The Blockchain networks can be applied in many use cases not limited listed briefly below. The Blockchain offers information trust because event data is maintained on a publicly achievable blockchain. It provides the secure flow of data between network objects (Vehicle, RSU servers). The Blockchain can solve the safety for the transfer of important data and reliable information transmission, while preventing deformation, which may result in negative outcomes. To avoid these outcomes, Blockchain technology is based on rules and principles. Because of the nature of blockchain trust management, it can be successfully implemented across nodes in decentralized networks. Using blockchain could prevent harmful nodes from accessing the network and disseminating misleading information on the network, creating disruptions of the transportation network. In some use cases, blockchain can be applied to rate a road user in V2X system, taken against offenders. This will guarantee that

harmful messages that harm the V2X system or degrade its efficiency are decreased. The use of trust management algorithms and priority separation allows traffic users to assess if a received message is credible with a high degree of certainty.

Blockchain could provide the flexibility to store and distribute the public keys or the vehicle certificates without the need of a 3<sup>rd</sup> party (trusted or not trusted) and, therefore, the generation of secure sessions between devices. It enables security between devices from different organizations. In some use cases, the immutably shared data among many devices and organizations. But on the other hand, Blockchain is not a mature technology yet. The transaction cost is increasing with digital currency prices and the scalability of the network should be also considered.

There were some proposals to use blockchain technology to ensure the security of V2X and mitigate the security issues listed above. The main advantage of Blockchain technology's application in V2X is the simple implementation of the trust authentication between vehicles, including the messages' integrity and confidentiality. The standard mechanism of security and privacy of the messages described by, e.g., the C-ITS could be simplified using the Blockchain.

Here we provide a summary of two existing survey papers on V2X and Blockchain. The document [36] lists articles around the integration of three technologies in one network: the 5G, Edge Cloud nodes, and Blockchain, showing this approach's advantages in Cellular V2X networks. The paper lists a comparative study of Blockchain in advanced vehicular networks with 5G-based edge computing and introduces the open issues like the storage in Blockchain with a massive volume of data transactions, the performance due to the limited throughput in terms of the number of transactions per second, delay, and network resource usage when blockchain and 5G-based MEC are integrated. The survey lists papers indicating that Blockchain consumes large network resources related to the mining and emphasizes that the consensus mechanism could result in high latency. The survey provides a future direction for research which could be to develop an efficient and robust incentive mechanism to encourage all parties and miners to engage in the blockchain. In addition, it suggests penalty systems to discourage any harmful activities. As a conclusion, the paper states that the Blockchain has emerged as a promising technology to solve most issues and challenges related to privacy, security, and networking faced by the existing and next-generation V2X technologies.

Whereas, survey document [37] lists research articles around the processing power and efficient trust mechanisms for information exchange in V2X communications and Intelligent Transportation Systems (ITS). Traditional access to the remote cloud may degrade the V2X services due to incurred latency. This survey lists and examines important solutions written around Edge solutions and Blockchain applied to V2X or what is called Internet of Vehicles (IoV) and provides a related technical classification from access technology, IoV architecture layers, network layers including SDN and NFV, blockchain layers execution and finally algorithms and applicability of machine learning through a comparative summary. It highlights their main features, advantages, and limitations to provide subsidies for further proposals.

Our actual survey includes the latest written papers on V2X solutions where the Blockchain is also the main component. Our survey is general, not limited to any specific technical topics or feature areas within Blockchain applications in V2X. On the contrary, the survey [36] covers only papers related to V2X, 5G, edge computing, and Blockchain integration only. After an exhaustive explanation of the V2X systems and their issues, the security and V2X evolution are the main focus. They list research papers related to this subject and do a comparative study of Blockchain with 5G and edge computing from system characteristics, considering the following parameters: the Blockchain type / consensus algorithm / edge solution / cellular technologies.

On the other hand, the survey [37] provides a taxonomy of blockchain and edge computing technologies for papers written in the context of IoV by analyzing the Blockchain type, the Blockchain layer, the adopted consensus algorithm, and the architectural approach.

Based on the identified gaps, we have reviewed several recent works not covered by the above available surveys but related to V2X systems and blockchain/hash chain technologies. We highlight the proposals' main idea, advantages, and weaknesses for each selected paper in the below sections.

#### *A. Traceable and Authenticated Key Negotiations via Blockchain for Vehicular Communications:*

In paper [38], authors analyze the communication between vehicles and between vehicles and people, and analyze requirements of low latency, high reliability, and traceability. The paper gets the Master Key (MK) information on blockchain and realize the master key negotiation. The proposed scheme utilizes a transaction data structure to generate key pairs like the Diffie–Hellman key-exchange process. This is achieved through four algorithms where the system parameters are generated and stored in blockchain, master key parameters in blockchain, and one algorithm to generate from the master key parameters the master key in the blockchain. This proposal can resist MiM and packet dropping attacks, and others. The key materials can be traced back by timestamps upon request and can be confirmable to avoid decryption failure attacks. The main issue is the long and unpredictable time to create secure connectivity due to blockchain latency.

#### *B. A Secure Priority Vehicle Movement based on Blockchain Technology in Connected Vehicles:*

A novel blockchain architecture was presented in paper [51], which protects vehicles from any attack, isolates them from other vehicles, and reduces the number of potential threats they will encounter. Their proposal could be modified to accommodate the priority vehicles' speeds to accommodate the Blockchain architecture. According to the specified maximum speed, the priority vehicle can travel through all the RSUs and reach its destination without any information-sharing mechanism between the RSUs and the priority vehicle. This is what we consider the proposal's vulnerability. On the other hand, the system is based on an ideal security system that exchanges information between priority vehicles and the RSUs without allowing anyone to communicate with them.

This system builds the authenticity and integrity between the RSU and the authentication center and the vehicles with the RSU in another way. Also, the hash chain makes the communication between the vehicle and the RSU secure.

#### *C. A Tiered Blockchain Framework for Vehicular Forensics:*

The authors of [52] developed a concept for vehicle forensics using blockchain technology. An analysis of the security levels that a car should pass through in the aftermath of an accident using blockchain technology and their resiliency to attack. The authors have contrasted and studied their proposed blockchain system (Block4Forensic) with other Blockchain systems for conflict resolution and responsibility attribution, demonstrating the strength of their proposed structure.

#### *D. Block-VN: A Distributed Blockchain-Based Vehicular Network Architecture in Smart City:*

In paper [39], the authors Proposed a distributed system based on blockchain for the ad hoc vehicle network. This modern network was called Blockchain-Based Vehicular Network (Block-VN) and allowed vehicles to explore and exchange their resources to build a network of vehicles that work together to deliver value-added services like forensics after a car accident occurs. The car's security level passed through a blockchain and analyzed. Furthermore, the authors discuss the architecture's security and dependability.

#### *E. A Novel Sender Authentication Scheme Based on Hash Chain for Vehicular Ad-Hoc Networks:*

The authors of paper [40] presented a hash chain scheme for promoting VANET security and secure communication between the vehicle, RSU, and authentication center. The authors used the symmetric key in hash chaining. The asymmetric keys were not considered because the symmetric key provides faster encryption and decryption in a secure network compared to the asymmetric keys.

The idea of this paper was to develop the symmetric hash chain process to strengthen the authenticity and integrity between the RSU and authentication center in one sense and the vehicles with the RSU in another sense. As a result of the hash chain, the communication between the Vehicle and RSU is secure and resistant to various attacks.

#### *F. Comparative Experiments of V2X Security Protocol Based on Hash Chain Cryptography:*

In paper [53], the authors proposed a lightweight message authentication and privacy preservation protocol for V2X communications. The proposed protocol achieves highly secure message authentication by introducing a hash chain of secret keys for a Message Authentication Code (MAC). The reason is that V2X security protocols based on the Elliptic Curve Digital Signature Algorithm (ECDSA) provide a high-security level at the cost of excessive communication and computation overhead. The proposed protocol was tested in a stationary state using their proposed test platform using commercial DSRC devices (Cohda wireless MK5 (5<sup>th</sup> Generation Market) devices that provide two types MK5-OBUs that are installed on the Vehicle and MK5-RSUs installed on the road)[41]. By using the well-known Wireshark utility, they measured the messaging performance.

## A comprehensive survey on the application of blockchain/hash chain technologies in V2X communications

Furthermore, they enabled security by using the cryptographic Library (Aerolink Library) configuration that uses the Hardware Security Model (HSM). Moreover, for signing and verifying the messages, they used ECDSA National Institute of Standards and Technology (NIST) P256 with SHA 256. Therefore, using IEEE 1609.2 and ETSI-103-097 standards, the result for the number of messages per second was 183 messages. In contrast, for non-standard protocol, the number of messages per second was only 54 messages. The authors conclude that the proposed protocol significantly decreases the average end-to-end delay and proves its performance advantages over the standard and non-standard protocols.

### G. Hash-Chain-Based Cross-Regional Safety Authentication for Space-Air-Ground Integrated VANETs:

Authors of article [42] presented a new concept of connectivity between vehicles, RSUs, and Certificate Authorities (Cas), with drones or Security Manager (SM) acting as intermediaries, sending information and data between them through a space air-ground integrated network (SAGIN) which establishes higher security standards. The authors developed a consensus framework based on the Hash chain, combined with the Radio Frequency (RF) fingerprint theory, simplifying the Blockchain, introducing the Kafka distributed messages, and Practical Byzantine Fault Tolerance Algorithm (PBFT). According to the simulation-based on Hyperledger-Fabric architecture, it shows that the average delay of the block produced by a single transaction is approximately 0.9ms, which achieves effective and low latency authentication.

### H. A New-Type of Blockchain for Secure Message Exchange in VANET:

A new Blockchain form is presented in this paper [54] to address issues of critical message propagation in VANET environments. This Blockchain doesn't use any crypto coins to handle safety event messages. After determining the vehicle's location with the aid of proof of location (POL), the VANET messages don't have to go beyond the country's boundary. The scheme depends on each geographical area isolated from the other, which ensures that traffic information of one country is unrelated to vehicles based in another country. Furthermore, after the vehicle receives the message, it tests it against the Blockchain. It then verifies the event messages to see if they are trustworthy before broadcasting them to the surrounding vehicles and storing the message in the local memory pool or discarding them. The results of this paper's assessment and review suggest that the proposed local Blockchain can be used effectively in the VANET without the need for additional storage.

### I. Secure V2X environment using Blockchain Technology:

In article [55], the authors' purpose is to provide a hypothetical scenario that depicts the effect of challenging factors on applying the Blockchain in the V2X paradigm. However, the authors have discussed considerations that may hurt the application of Blockchain in V2X context. A total of 10 of the most critical challenges are established using the Systematic Literature Review (SLR) method, and their corresponding hypothesis was also developed. Some studies

were considered for the data extraction process by applying the tollgate method. They have explored in this study the factors that could have a negative impact on the implementation of blockchain in V2X environment. By considering both studies' results (SLR and case study), the authors have created a hypothetical model that helps practitioners revise their strategies and create an efficient method for successfully implementing Blockchain in the V2X context.

### J. A Remote Attestation Security Model based on Privacy-Preserving Blockchain for V2X:

Intelligent, V2X-based applications require the real-time integration of all kinds of information on roads, pedestrians, the environment, and vehicles themselves. This information also needs to be shared and integrated privately with other vehicles. The authors of paper [29] suggest a remote attestation protection mechanism built on a privacy-preserving blockchain called the remote attestation model (RASM). This scheme entails two main stages. The first is the credible verification of identification. The second uses estimation for decision-making to classify the node trusted or malicious, for example.

### K. BloCkEd: Blockchain-based Secure Data Processing Framework in Edge Envisioned V2X Environment:

The authors of article [56], proposed a Blockchain-based data processing platform (BloCkEd) for the V2X environments, where the V2X users are connected to the EDGE nodes. The scheme allows V2X users' requests to be handled/processed by nodes at the edge of the network; thus, reducing latency; and preserving the privacy of user data/activities. BloCkEd comprises an optimal container-based data processing scheme; and a blockchain-based data integrity management scheme; designed to minimize link breakage and reduce latency. The program implementation of the proposed architecture was tested against a plausible scenario in Chandigarh City, Punjab, India. The results showed that the proposed solution promotes less migration due to an efficient allocation strategy, decreasing regular connection breaks and service disturbances.

### L. Efficient Mining Cluster Selection for Blockchain-Based Cellular V2X Communications:

Using game theory, paper [57] demonstrates how to balance the load on mining clusters while ensuring unloading vehicles' justice. As mining tasks are unloaded in cellular V2X networks, they can cause congestion and disproportionate vehicle network resources. Moreover, a short block length transmission design was considered to meet the low-latency standards of safety applications. The proposed solution guarantees decent transmission speeds and preserves justice between the unloading of vehicles. The findings show that the proposed methodology's efficiency rises as the number of mining clusters in the network increases.

### M. A Blockchain Approach for Decentralized V2X (D-V2X)

As we introduced above, the current V2X solutions rely on using a Public Key Infrastructure that enables secure collaboration between the different entities in the V2X ecosystem. However, managing such infrastructure requires reaching agreements between many parties with conflicts of interest between automakers and telecommunication operators.

In paper [43], the authors propose a decentralized V2X (D-V2X) solution based on Blockchain that does not need any trusted authority and can be applied on top of any communication protocol. The authors describe a proof-of-concept to build the D-V2X on top of a low-cost and high-security System-on-Chip (SoC) that could enable widespread D-V2X adoption.

#### *N. PF-BVM: A Privacy-aware Fog-enhanced Blockchain Validation Mechanism*

In paper [44], the authors suggested a Privacy-aware Fog-enhanced Blockchain Validation Mechanism (PFBVM) to reduce the load on the network by implementing a new validation mechanism and equivalent consensus feature, where trusted authenticated fog nodes can validate transactions on behalf of blockchain nodes. It integrates fog computing, the Internet of Things, and Blockchain techniques. The PF-BVM algorithm aims to reduce the approval of a transaction and, in that way, to reduce the latency in Blockchain. PF-BVM allows trusted rich fog nodes to perform transaction validation on behalf of other blockchain nodes as a conceptual criterion. The trust is gained by randomly running matching tests which adds to the integration of fog computing. According to the findings, the greater the number of transactions per block, the higher the blockchain system's un-reliability metric. The authors used a specially formulated simulation code to analyze the proposed mechanism. The experimental results demonstrated that PF-BVM could significantly improve a blockchain system validation in time consumption, energy efficiency, and storage capacity.

#### *O. Blockchain-based Service Sharing Via Roadside Unit-Performance Evaluation:*

Using Blockchain technology, the authors proposed a model for services sharing via RSU [45], which has just one RSU that receives and saves services on Blockchain, and vehicles interact with each other via RSU. A simulation was implemented to estimate the performance of the system using Python. There are two forms of communication discussed: communication between vehicles and communication between vehicles and RSUs. When the vehicle agrees to share its services with a requested vehicle, a smart contract will be established between two vehicles, provided both agree to the smart contract regulations.

#### *P. Technological Aspects of Blockchain Application for Vehicle-to-Network:*

In paper [46], the authors suggested using Blockchain technology in V2N to tackle the problem of maintaining information security, which is very sensitive related to the specifics of the operation of transport networks. Four experiments were conducted to demonstrate the numerical features for resource allocation on devices engaged in arranging V2N communication. The findings show that the nodes' activity determines the channel bandwidth consumed. During blockchain operation, the latency of packets between nodes decreased significantly, and there was almost no influence on the delay with the nodes of another network. In comparison, the latency variation in operating the blockchain failed nodes

simultaneously without synchronizing the mining interaction did not occur significantly between the nodes.

#### *Q. Blockchain Enhanced V2X Communication System and Method:*

In this patent application [47], the authors propose an authentication system for V2X communication systems based on a private blockchain. The system includes a blockchain-based V2X decentralized Certificate Authority (CA) instead of a third-party CA. A blockchain-based V2X CA provides an open, distributed ledger that can efficiently record transactions between multiple parties in a verifiable and permanent way.

#### *R. Distributed Edge Computing with Blockchain Technology to Enable Ultra-Reliable Low-Latency V2X Communications:*

Paper [48] aims to solve the problem of building a vehicular network for reliable delivery data according to the V2X standard and improving road users' safety using blockchain technology and Mobile Edge Computing (MEC). Again, here the authors of this paper consider the four technology 5G, V2X, MEC, and blockchain. The proposed work provides a mathematical model of the system, considering the interconnection of objects and V2X information channels and an energy-efficient offloading algorithm to manage traffic offloading to the MEC server.

The proposed system architecture consists of roadside participants like vehicles, several RSUs, distributed MEC units, and the application server. The blockchain technology can be used to manage information trustworthiness, as event information would be stored in a publicly accessible blockchain. Blockchain can solve major problems faced by V2X systems and provide security for the distribution of critical information. One scenario is that Malicious nodes can infiltrate the network and spread false information, causing the transport network to fail. The blockchain can rate a road user is also an effective solution for use in the V2X system. A rating facility would allow action against offenders and encourage decent users.

The paper provides a framework of V2X based on distributed edge computing integrated with blockchain technologies. A model for the interaction of blockchain technology in the system was introduced to achieve the required level of security. The developed Blockchain-MEC model was evaluated over an NS-3 environment for various simulation scenarios, and the results validate the system in terms of reliability, latency, and energy efficiency. The results showed that Blockchain-MEC V2X system achieved higher reliability than existing V2X models.

#### *S. Blockchain for V2X: A Taxonomy of Design Use Cases and System Requirements:*

Article [49] provides an overview of V2X blockchain architecture applications and examines them in order to define the needs of a V2X blockchain. The study investigates possible blockchain applications in the V2X space, identifying and assessing use cases based on their underlying blockchain needs. The authors classify blockchain into two categories: permissionless and permissioned blockchains. According to the authors, permissioned blockchains are the greatest solution for enabling the largest range of applications while also ensuring

A comprehensive survey on the application of blockchain/hash chain technologies in V2X communications

that throughput, user privacy, and Know Your Customer (KYC) requirements are all satisfied.

*T. A blockchain-based V2X communication system:*

A new blockchain-based V2X secure communication platform was proposed in this paper [50], which integrates PKI/CA model aspects with blockchain technology. The authors describe a typical PKI/CA-based alternative solution to

the European standard C-ITS' authentication. The solution attempts to alleviate trust problems in the existing PKI/CA infrastructure while also facilitating vehicle authentication and security in the V2X network. The platform is based on the Ethereum blockchain to store and retrieve the Public keys of the roadside participants and RSU.

Table 1: A comparative analysis of the surveyed V2X papers related to Blockchain technologies.

Author	Technology	Use case	Description	Brief Summary of Results
Y. Chen et al. [38], 2019	Blockchain/Authenticated key negotiations	Vehicular communication	A Blockchain to resolve the key negotiation between two vehicles to be authenticated and traceable.	The Key materials can be publicly tracked back by timestamps and confirmed to prevent decryption failure attacks.
A. Saini et al. [51], 2019	Blockchain/Privacy and security	Connected Priority vehicles	A novel blockchain architecture that protects vehicles from many attacks and isolates them from other vehicles.	The proposed scheme will efficiently and safely address priority vehicle movement.
M. C. Ugwu et al. [52], 2018	Blockchain/Watchdog entity	Vehicular Forensics	A blockchain concept for vehicular forensics; after an accident occurs, the car's security level should pass through blockchain technology and analyze them against attack.	Demonstrates the proposed architecture's effectiveness compared to the current Blockchain-based system.
Pradip Kumar Sharma et al. [39], 2017	Blockchain/Block-VN	Ad hoc Vehicle network	An ad hoc network application and discovered capabilities that the current infrastructure cannot quickly provide.	The Block-VN paradigm encourages vehicles to explore and exchange their resources, resulting in a network of vehicles cooperating to create value-added services.
N. V. Vighmesh et al. [40], 2011	Blockchain/Hash chain	Vehicular ad hoc network (VANET)	Hash chain scheme that promotes VANET security and secure communication between the vehicle, RSU, and authentication center.	Its widespread use in cryptography explains the popularity of the hash function.
S. A. A. Hakeem et al. [53], 2020	Blockchain/MAC algorithm	Vehicle-to-Everything (V2X)	Security protocols have been tested in a stationary state using commercial DSRC devices.	The proposed protocol significantly decreases the average end-to-end delay.
G. Luo et al. [42], 2020	Hash chain/space-air-ground integrated network (SAGIN)	VANETs	A new idea of communication between vehicles that plays the role of sending information and data between them and use the space-air-ground integrated network (SAGIN) to set out higher security standards.	The average delay of the block produced by a single transaction is approximately 0.9ms to achieve effective and low latency authentication.
R. Shrestha et al. [54], 2020	Blockchain/Mobile Edge Computing	VANET	A new form of Blockchain to address issues of critical message propagation in VANET.	The proposed local Blockchain can be used effectively in the VANET without the need for additional storage.
Ms. Taiyaba et al. [55], 2020	Blockchain/Systematic literature review (SLR)	V2X	Provides a hypothetical scenario that depicts the effect of challenging factors on applying the Blockchain in the V2X paradigm.	A hypothetical model was created that helps practitioners revise their strategies and develop efficient methods for successfully implementing Blockchain in the V2X context.
C. Xu et al. [29], 2018	Blockchain/Remote attestation security Model (RASM)	V2X	A remote attestation protection mechanism built on a privacy-preserving blockchain called the remote attestation model RASM.	The findings demonstrate that a high proportion of progress can be attained with the scheme.
G. S. Aujla et al. [56], 2020	Blockchain/BlockED	V2X	A blockchain-based protected data processing system for an EDGE node of the V2X area called BloCkEd.	The proposed solution promotes less migration due to an efficient allocation strategy, decreasing regular connection breaks and service disturbances.
F. Jameel et al. [57], 2020	Blockchain/Blocklength transmission	V2X	A game-theoretic approach to balance the load on mining clusters while ensuring the justice of unloading vehicles.	The findings show that the proposed methodology's efficiency rises as the number of mining clusters in the network increases.

<i>I. Agudo et al. [43], 2020</i>	Blockchain/ System-on-Chip (SoC)	V2X	Decentralized V2X (D-V2X) solution based on Blockchain, that does not need any trusted authority and can be applied on top of any communication protocol.	The Current V2X solutions rely on using a public key infrastructure that enables secure collaboration between the different entities in the V2X ecosystem.
<i>H. Baniata et al. [44], 2020</i>	Blockchain/ Internet of things, Fog computing	Vehicle	Privacy-aware Fog-enhanced Blockchain Validation Mechanism (PFBVM).	PF-BVM could significantly improve a blockchain system validation in terms of time consumption, energy efficiency, and storage capacity.
<i>I. Kiran et al. [45], 2019</i>	Blockchain/ Proof of Work (PoW)	V2V, V2I	They are examining the performance of the vehicle to vehicle and vehicle to RSU communication at the time of service sharing. Its primary purpose is to minimize average time delay.	RSU helps to minimize the average delay time to achieve maximum throughput.
<i>V. Elagin et al. [46], 2020</i>	Blockchain	V2N	Blockchain is employed as a system platform to serve the demands of transportation systems for safe information sharing.	The usage of blockchain technology is not an appropriate solution for V2N.
<i>Qi. Jimmy et al. [47], 2020</i>	Blockchain/ Certificate Authority (CA), PKI	V2X	Patent Application for V2X decentralized CA based on blockchain instead of third-party CA.	Authentication system for V2X communication based on private blockchain.
<i>A. Vladkyo et al. [48], 2022</i>	Blockchain/ mobile edge computing (MEC)/ 5G	V2X	The simulation model consists of roadside participants (vehicles, RSUs, distributed MEC units, and application server). The model was evaluated over an NS-3 environment for various simulation scenarios.	Validation of the system in terms of latency, reliability, and energy efficiency. Blockchain-MEC V2X system achieved higher reliability than existing V2X models.
<i>J. Meijers et al. [49], 2021</i>	Blockchain/ IoT	V2X	Investigating potential blockchain applications in the V2X, finding and evaluating use cases based on their underlying blockchain requirements.	Permissioned blockchains are the greatest solution for enabling the largest range of applications.
<i>H. Farran et al. [50], 2021</i>	Blockchain/ Public Key Infrastructure (PKI), CA	V2X	A blockchain to resolve the existing PKI/CA infrastructure's trust concerns and facilitate the authentication and security of the vehicles in the V2X network.	A platform based on Ethereum blockchain to store and retrieve the public keys of the roadside participants and RSU.

II. CONCLUSION

By way of inference, safety is the primary issue for road drivers using highly advanced applications in the future's highly cooperative ITS environments. V2X has the potential to comply with safety criteria providing updates to drivers on the road. Hence, it is necessary to ensure the network's security and establish confidence in V2X interactions. This paper provides a comprehensive survey on different vehicular applications using blockchain technologies to enhance the main message from the studied articles is that the Blockchain offers reliability, trust, and simplification in implementing security to V2X networks. Blockchain-based solutions construct durability and reliability in V2X, together with distributed operation and data storage.

Based on this survey, we conclude that blockchain networks and technology can play an important role in V2X applications from different aspects and resolve many technical issues. We enumerate the following: 1) traceable key negotiation between two vehicles; 2) protection of vehicles from many attacks and isolates them from other vehicles; 3) security and secure communication between the vehicle, RSU, and authentication center; 4) ad hoc network application and discovered capabilities; 5) vehicular forensics after an accident incidence; 6) security between devices from different organizations; 7) the

immutably shared data among many devices and organization; 8) simplifying the distribution of the participants' CA in V2X; 9) improving the performance of the V2X system when combined with EDGE node in terms of latency, reliability, and energy efficiency; and 10) trust authentication between vehicles, including the messages' integrity and confidentiality. We expect that the list of functions will further improve, and implementation of V2X systems based on Blockchain will increase in the coming years.

On the other hand, Blockchain is not mature technology yet and requires a lot of improvements in scalability, transactions latency, and cost.

Preparing this survey taught us that Blockchain quickly became an important topic also in V2X communications. As a part of our future work, we design a novel Blockchain-based proof-of-concept V2X security solution to highlight possibilities and implement security on such new based in the C-ITS domain. Our primary purpose is to increase awareness of Blockchain+V2X and to create a lightweight, distributed pilot alternative to the PKI-based current schemes and the complicated assignment of the CAs to the vehicles or IoT devices in general by a more generic and simplified method based on the Ethereum Blockchain.

A comprehensive survey on the application of blockchain/hash chain technologies in V2X communications

REFERENCES

[1] 5GAA, "White Paper on ITS spectrum utilization in the Asia Pacific [22] Region White paper on ITS spectrum utilization in the Asia Pacific Region," p. 20, 2018.

[2] 5G American, "5G Americas White Paper: Cellular V2X Communications Towards 5G," [Http://Www.5Gamericas.Org](http://www.5Gamericas.org), 2018.

[3] A. Boudguiga, A. Kaiser, and P. Cincilla, "Cooperative-ITS Architecture and Security Challenges: a Survey," *22th ITS World Congr.*, no. October, pp. 5–9, 2015.

[4] S. Rangarajan, M. Verma, A. Kannan, A. Sharma, and I. Schoen, "V2C: A secure vehicle to cloud framework for virtualized and on-demand service provisioning," in *ACM International Conference Proceeding Series*, 2012, pp. 148–154, **doi**: 10.1145/2345396.2345422.

[5] D. P. Tuttle, R. L. Fares, R. Baldick, and M. E. Webber, "Plug-In Vehicle to Home (V2H) duration and power output capability," in *2013 IEEE Transportation Electrification Conference and Expo: [26] Components, Systems, and Power Electronics - From Technology to Business and Public Policy, ITEC 2013*, 2013, **doi**: 10.1109/ITEC.2013.6574527.

[6] Alfonso Damiano, Gianluca Gatto, Ignazio Marongiu, Mario Porru, and Alessandro Serpi, "Vehicle-to-Grid Technology: State-of-the-Art and Future Scenarios," *J. Energy Power Eng.*, vol. 8, no. 1, 2014, **doi**: 10.17265/1934-8975/2014.01.018.

[7] I. S. Victor Sandonis and M. U., Maria Calderon, "Vehicle to Internet communications using the ETSI ITS GeoNetworking protocol," *Trans. Emerg. Telecommun. Technol.*, vol. 25, no. 3, pp. 294–307, 2016, **doi**: 10.1002/ett.

[8] J. Wang, Y. Shao, Y. Ge, and R. Yu, "A survey of vehicle to everything (V2X) testing," *Sensors (Switzerland)*, vol.19, no. 2, pp. 1–20, 2019, **doi**: 10.3390/s19020334.

[9] "Vehicle-to-everything-Wikipedia." [Online]. Available: <https://en.wikipedia.org/wiki/Vehicle-to-everything>. [Accessed: 29-Apr-2020].

[10] H. Leung, N. E. El Faouzi, and A. Kurian, "V2X Communication Protocol in Vanet for Co-Operative Intelligent transportation system (ITS)," *Inf. Fusion*, vol. 12, no. 1, pp. 2–3, 2011, **doi**: 10.1016/j.inffus.2010.06.003.

[11] C. R. Storck and F. Duarte-Figueiredo, "A Survey of 5G Technology Evolution, Standards, and Infrastructure Associated with Vehicle-to-Everything Communications by Internet of Vehicles," *IEEE Access*, vol. 8, pp. 117593–117614, 2020, **doi**: 10.1109/ACCESS.2020.3004779.

[12] M. Lu, O. Turetken, O. E. Adali, J. Castells, R. Blokpoel, and P. Grefen, "Cooperative Intelligent Transport Systems (C-ITS) [34] deployment in Europe: Challenges and key findings," *25th ITS World Congr.*, no. September, p. EU-TP1076, 2018.

[13] J. Petit, F. Schaub, M. Feiri, and F. Kargl, "Pseudonym Schemes in Vehicular Networks: A Survey," *IEEE Commun. Surv. Tutorials*, vol. 17, no. 1, pp. 228–255, 2015, **doi**: 10.1109/COMST.2014.2345420.

[14] M. H. Eiza and Q. Ni, "Driving with Sharks: Rethinking Connected Vehicles with Vehicle Cybersecurity," *IEEE Veh. Technol. Mag.*, vol. 12, no. 2, pp. 45–51, Jun. 2017, **doi**: 10.1109/MVT.2017.2669348.

[15] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011, **doi**: 10.1109/JPROC.2011.2132790.

[16] D. Jiang and L. Delgrossi, "IEEE 802.11p: Towards an international standard for wireless access in vehicular environments," *IEEE Veh. Technol. Conf.*, no. June 2008, pp. 2036–2040, 2008, **doi**: 10.1109/VETECS.2008.458.

[17] F. Arena, G. Pau, and A. Severino, "A review on IEEE 802.11p for intelligent transportation systems," *J. Sens. Actuator Networks*, vol. 9, no. 2, pp. 1–11, 2020, **doi**: 10.3390/jsan9020022.

[18] I. V. T. Society, IEEE Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages, vol. 2017. 2016.

[19] J. Kolleda et al., "National Security Credential Management System (SCMS) Deployment Support : Literature Search Report," 2018.

[20] C-ITS Platform, "Certificate Policy for Deployment and Operation of European Cooperative Intelligent Transport Systems (C-ITS)," no. June, pp. 1–79, 2018.

[21] T. Giannetsos et al. "Securing V2X Communications for the Future: Can PKI Systems offer the answer?" [Online]. Available: [https://www.researchgate.net/publication/335089342\\_Securing\\_V2X\\_Communications\\_for\\_the\\_Future\\_Can\\_PKI\\_Systems\\_offer\\_the\\_answer](https://www.researchgate.net/publication/335089342_Securing_V2X_Communications_for_the_Future_Can_PKI_Systems_offer_the_answer). [Accessed: 19-Feb-2021].

[22] B. Brecht et al., "A Security Credential Management System for V2X Communications," vol. 19, no. 12, pp. 3850–3871, 2018, **doi**: 10.1109/TITS.2018.2797529.

[23] "Public key encryption (article) | Khan Academy." [Online]. Available: <https://www.khanacademy.org/computing/computers-and-internet/xcae6f4a7ff015e7d:online-data-security/xcae6f4a7ff015e7d:data-encryption-techniques/a/public-key-encryption>. [Accessed: 20-Feb-2021].

[24] A. Sulaiman, S. V. Kashmir Raja, and S. H. Park, "Improving scalability in vehicular communication using one-way hash chain method," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2526–2540, 2013, **doi**: 10.1016/j.adhoc.2013.05.017.

[25] A. Ghosal and M. Conti, "Security issues and challenges in V2X: A Survey," *Comput. Networks*, vol. 169, no. March 2019, 2020, **doi**: 10.1016/j.comnet.2019.107093.

[26] G. Samara, W. A. H. Al-Salihy, and R. Sures, "Security analysis of Vehicular Ad Hoc Networks (VANET)," *Proc. - 2nd Int. Conf. Netw. Appl. Protoc. Serv. NETAPPS 2010*, pp. 55–60, 2010, **doi**: 10.1109/NETAPPS.2010.17.

[27] Xin Wang University of California, Santa Cruz, "Mobile Ad-Hoc Network Applications". Iva Lipovic, 2011.

[28] T. Zhou, R. R. Choudhury, P. Ning, and K. Chakrabarty, "P2DAP - Sybil attacks detection in vehicular ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 3, pp. 582–594, Mar. 2011, **doi**: 10.1109/JSAC.2011.110308.

[29] C. Xu, H. Liu, P. Li, and P. Wang, "A remote attestation security model based on privacy-preserving blockchain for V2X," *IEEE Access*, vol. 6, no. c, pp. 67809–67818, 2018, **doi**: 10.1109/ACCESS.2018.2878995.

[30] M.A. Hezam Al Junaid, A.A. Syed, M.N. Mohd Warip, K.N. Fazira Ku Azir, and N. H. Romli, "Classification of Security Attacks in VANET: A Review of Requirements and Perspectives," *MATEC Web Conf.*, vol. 150, pp. 1–7, 2018, **doi**: 10.1051/mateconf/201815006038.

[31] M. Pirretti, S. Zhu, N. Vijaykrishnan, P. McDaniel, M. Kandemir, and R. Brooks, "The sleep deprivation attack in sensor networks: Analysis and methods of defense," *Int. J. Distrib. Sens. Networks*, vol. 2, no. 3, pp. 267–287, 2006, **doi**: 10.1080/15501320600642718.

[32] V. Hoa La and A. Cavalli, "Security Attacks and Solutions in Vehicular Ad Hoc Networks: A Survey," *Int. J. AdHoc Netw. Syst.*, vol. 4, no. 2, pp. 1–20, Apr. 2014, **doi**: 10.5121/ijans.2014.4201.

[33] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," *SSRN Electron. J.*, 2019, **doi**: 10.2139/ssrn.3440802.

[34] E. F. Kfoury, J. Gomez, J. Crichigno, E. Bou-Harb, and D. Khoury, "Decentralized Distribution of PCP Mappings over Blockchain for End-to-End Secure Direct Communications," *IEEE Access*, vol. 7, no. August, pp. 110159–110173, 2019, **doi**: 10.1109/ACCESS.2019.2934049.

[35] "Ethereum Whitepaper | ethereum.org." [Online]. Available: <https://ethereum.org/en/whitepaper/>. [Accessed: 15-May-2021].

[36] R. Shrestha, S. Y. Nam, R. Bajracharya, and S. Kim, "Evolution of V2X communication and integration of blockchain for security enhancements," *Electron.*, vol. 9, no. 9, pp. 1–33, 2020, **doi**: 10.3390/electronics9091338.

[37] A. Queiroz, E. Oliveira, M. Barbosa, and K. Dias, "A Survey on Blockchain and Edge Computing applied to the Internet of Vehicles," *Int. Symp. Adv. Networks Telecommun. Syst. ANTS*, vol. 2020-December, 2020, **doi**: 10.1109/ANTS50601.2020.9342818.

[38] Y. Chen, X. Hao, W. Ren, and Y. Ren, "Traceable and Authenticated Key Negotiations via Blockchain for Vehicular Communications," *Mob. Inf. Syst.*, vol. 2019, 2019, **doi**: 10.1155/2019/5627497.

[39] P. K. et al. Sharma, "Block-VN: A Distributed Blockchain Based Vehicular Network Architecture in Smart City." .

[40] N. V. Vighnesh, N. Kavita, S. R. Urs, and S. Sampalli, "A novel sender authentication scheme based on hash chain for Vehicular Ad-Hoc Networks," *ISWTA 2011 - 2011 IEEE Symp. Wirel. Technol. Appl.*, pp. 96–101, 2011, **doi**: 10.1109/ISWTA.2011.6089388.

[41] O. Unit, "MK50BUChoda Wireless 5th generation market ready."

[42] G. Luo, M. Shi, C. Zhao, and Z. Shi, "Hash-chain-based cross-regional safety authentication for space-air-ground integrated VANETs," *Appl. Sci.*, vol. 10, no. 12, p. 4206, 2020, doi: 10.3390/AP10124206.

[43] I. Agudo, M. Montenegro-Gomez, and J. Lopez, "A Blockchain Approach for Decentralized V2X (D-V2X)," *IEEE Trans. Veh. Technol.*, 2020, doi: 10.1109/TVT.2020.3046640.

[44] H. et al. Baniata, "PF-BVM: A Privacy-aware Fog-enhanced Blockchain Validation Mechanism." [Online]. Available: [https://www.researchgate.net/publication/341482783\\_PF-BVM\\_A\\_Privacy-aware\\_Fog-enhanced\\_Blockchain\\_Validation\\_Mechanism](https://www.researchgate.net/publication/341482783_PF-BVM_A_Privacy-aware_Fog-enhanced_Blockchain_Validation_Mechanism). [Accessed: 20-Mar-2021].

[45] I. Kiran and N. Javaid, "Blockchain-based Service Sharing Via Roadside Unit-Performance Evaluation," no. July, 2019.

[46] V. Elagin, A. Spirikina, M. Buinevich, and A. Vladyko, "Technological aspects of blockchain application for vehicle-to-network," *Inf.*, vol. 11, no. 10, pp. 1–19, 2020, doi: 10.3390/info11100465.

[47] Q. Jimmy et al., "Blockchain Enhanced V2x Communication System And Method." [Online]. Available: <https://uspto.report/patent/app/20200145191>. [Accessed: 22-Feb-2022].

[48] A. Vladyko, V. Elagin, A. Spirikina, A. Muthanna, and A. A. Ateya, "Distributed Edge Computing with Blockchain Technology to Enable Ultra-Reliable Low-Latency V2X Communications," *Electron.*, vol. 11, no. 2, pp. 1–18, 2022, doi: 10.3390/electronics11020173.

[49] J. Meijers et al., "Blockchain for V2X: A Taxonomy of Design Use Cases and System Requirements," *2021 3rd Conf. Blockchain Res. Appl. Innov. Networks Serv. BRAINS 2021*, pp. 113–120, 2021, doi: 10.1109/BRAINS52497.2021.9569796.

[50] H. Farran, D. Khoury, E. Kfoury, and L. Bokor, "A blockchain-based V2X communication system," *2021 44th Int. Conf. Telecommun. Signal Process. TSP 2021*, pp. 208–213, 2021, doi: 10.1109/TSP52935.2021.9522599.

[51] A. Saini, S. Sharma, P. Jain, V. Sharma, and A. K. Khandelwal, "A secure priority vehicle movement based on blockchain technology in connected vehicles," *ACM Int. Conf. Proceeding Ser.*, 2019, doi: 10.1145/3357613.3357631.

[52] M. C. Ugwu, I. U. Okpala, C. I. Oham, and C. I. Nwakanma, "A Tiered Blockchain Framework for Vehicular Forensics," *Int. J. Netw. Secur. Its Appl.*, vol. 10, no. 5, pp. 25–34, 2018, doi: 10.5121/ijnsa.2018.10503.

[53] S. A. A. Hakeem, M. A. A. El-Gawad, and H. Kim, "Comparative experiments of V2X security protocol based on hash chain cryptography," *Sensors (Switzerland)*, vol. 20, no. 19, pp. 1–23, 2020, doi: 10.3390/s20195719.

[54] R. Shrestha, R. Bajracharya, A. P. Shrestha, and S. Y. Nam, "A new type of blockchain for secure message exchange in VANET," *Digit. Commun. Networks*, vol. 6, no. 2, pp. 177–186, 2020, doi: 10.1016/j.dcan.2019.04.003.

[55] M. Taiyaba, M. A. Akbar, B. Qureshi, M. Shafiq, M. Hamza, and T. Riaz, "Secure V2X Environment using Blockchain Technology," *ACM Int. Conf. Proceeding Ser.*, no. April, pp. 469–474, 2020, doi: 10.1145/3383219.3383278.

[56] G. S. Aujla et al., "BloCkEd: Blockchain-based secure data processing framework in edge envisioned V2X environment," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5850–5863, 2020, doi: 10.1109/TVT.2020.2972278.

[57] F. Jameel, M. A. Javed, S. Zeadally, and R. Jäntti, "Efficient Mining Cluster Selection for Blockchain-based Cellular V2X Communications," arXiv, pp. 1–9, 2020, doi: 10.1109/tits.2020.3006176.



**Hassan Farran** received a B.S. degree in Computer and Communication Engineering from the American University of Science and Technology (AUST) Beirut, Lebanon in 2015 and an M.Sc. degree from the same University in 2017. Currently, he is a Third-year Ph.D. student at the Budapest University of Technology and Economics (BME) Budapest, Hungary, at the Department of Networked Systems and Services (HIT). He has worked as a researcher and teaching assistant at Budapest University of Technology and Economics for the last three years. His research focuses on Networking and Vehicular Communications (V2X).



**David Khoury** received the M.E. degree in telecommunications from ESIB, in 1983. He held different positions at Ericsson mainly in France and Sweden in Research and Development and Product and System management. He was involved in early studies of them GSM and the evolution toward an IP-based network and the early studies of 3G/WCDMA, HSPA, and LTE. In 2010, he has established his own start-up company (Secumobi) developing advanced military grade secure communications system and security solutions based on hardware encryption and trusted execution environments (TEE) in Stockholm. For the past 8 years, Full-time Faculty member and Research & Innovation fellow at AUST (American University of Science & Technology) in Beirut. From 2018 he is Strategy consultant for Wone, a startup located in Switzerland. He holds 5 US patents and has published many research papers in international and local conferences. His research interests include the IoT, information security, and blockchain technology.



**László Bokor** received the M.Sc. degree in computer engineering from the Department of Telecommunications, Budapest University of Technology and Economics (BME) in 2004, informatics from the Faculty of Economic Informatics, BME. He has researched in multiple EU-funded and national research and development projects for several years. He is currently with the Department of Networked Systems and Services (HIT) as an Associate Professor and leads the Commsignia-BME HIT Automotive Communications the M.Sc.+ degree in bank and Social Sciences, BME, and the Ph.D. degree from the Doctoral School of Research Group at BME and the Vehicle Communication Working Group of the Mobility Platform at KTI Institute for Transport Sciences. He is a member of the HTE (Scientific Association for Infocommunications Hungary), the Hungarian Standards Institution's Technical Committee for Intelligent Transport Systems (MSZT/MB 911), the TPE GoverC-ITS Task Force within the TPEG Application Working Group of TISA, the ITS Hungary Association (the Hungarian organization of ERTICO's Network of National ITS Associations), and the BME' the UNKP-16-4-I. Post-Doctoral Fellowship in 2016 from the New National Excellence Program of the Ministry of Human s Multimedia Networks and Services Laboratory, where he participates in different R&D projects. In recognition of his professional work and achievements in mobile telecommunications, he received the HTE Silver Medal (2013), the HTE Pollák-Virág Award (2015), and the HTE Gold Medal (2018). He was a recipient of Capacities of Hungary. In 2018 he was awarded the Dean's Honor (BME VIK) for education and research achievements in the field of communication of autonomous vehicles; in 2020, he received the BME HIT Excellence in Education Award.

# Optimizing Camera Stream Transport in Cloud-Based Industrial Robotic Systems

Marcell Balogh and Attila Vidács

**Abstract**—Combining visual-guided robotics with cloud networking brought a new era into industrial robotic research and development. New challenges have to be tackled with a focus on providing proper communication and data processing setup: sensor data processing as well as the control software should be decoupled from the local robot hardware and should move into the cloud. In the emerging field of cloud robotics, there are trade-offs that have to be handled. More and more sensors such as cameras are being integrated but it comes with a cost. All sensory data have to be sent through often limited networking resources, while latency must be kept as low as possible.

In this paper we propose a general solution for efficient camera stream transportation in cloud robotic systems. After introducing our test scenario with the used hardware and software elements, a detailed overview of the architecture is presented with describing each task of the components. The goal of this paper is to examine the current stream transportation implementations in ROS environment and implement a more efficient method. The performance of the proposed method is investigated and compared with other solutions evidenced by measurements.

**Index Terms**—cloud robotics; distributed systems; image processing;

## I. INTRODUCTION

CONTRARY to certain expectations that human workers will be displaced by robots, the real trend is to utilize collaborative robots beside humans to work with. This change necessitates robots to be aware of their full surroundings real time, that seems to be the real challenge.

Vision systems are widely used in industrial robotics for various tasks and processes such as inspection and quality control, robot guidance, safety of workers, assembly lines, etc. As a result of the continuous improvement of camera sensors, the size of the raw sensory information significantly increased. It became a trade-off between the camera quality and the latency of the transported image stream. In order to get the best performance, modifications need to be tailor-made.

We expect these systems to examine their environment through various sensors and act immediately to prevent human injuries or collisions. Having applied the techniques of cloud computing, image processing, robotics and distributed networks, we present an alternative stream transportation method for vision aided real-time robotic systems. Our approach is to

Marcell Balogh is with the High Speed Networks Laboratory at the Department of Telecommunications and Media Informatics, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Hungary, (e-mail: balogh.marcell@edu.bme.hu)

Attila Vidács is with the High Speed Networks Laboratory at the Department of Telecommunications and Media Informatics, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Hungary, (e-mail: vidacs.attila@vik.bme.hu)

combine the existing methods in a more efficient way with distributed systems.

Throughout this work we utilized the Robot Operating System (ROS) [1] which became the *de facto* standard of robotic development. ROS based systems fit well into the concepts of cloud architecture. Sensors are providing real-time information which are being sent over the network, while data are processed in the cloud, and only the acting commands are being sent back to the robot. In this scheme, complex sensor networks can be easily managed, and data could be processed off board.

The paper is outlined as follows. In Section II, the background of the applied technologies is introduced. Section III presents an overview of the system including the hardware devices and the software design. Section IV explains the demonstration task, the steps of image processing alongside with the network setup. In Section V the different measurement methods are presented with their arrangements. This is followed by the performance measurement in Section VI which contains a detailed comparison of the different options. It contains the overall result and performance of the implemented system. Finally, conclusions and considerations regarding the improved solutions are presented in Section VII.

## II. RELATED WORK

Industrial robots took a long way to reach their current form and the new era of collaborative robots is currently rising. On the contrary of what people believed with the appearance of industrial robots, workers are still essential elements in the factories thus a stronger human-robot cooperation has started to emerge. Robots are able to cooperate even better with humans, taking their presence into account and proceed with caution.

To help robots look around and act within their environment, visual servoing is a popular approach. Visual servoing is an approved technology which was first proposed in 1996 by Hutchinson *et al.* [2], and was significantly improved by F. Chaumette and S. Hutchinson [3]. Nowadays two popular approaches were formalized: Position Based Visual Servoing (PBVS) and Image Based Visual Servoing (IBVS). PBVS seeks to calculate and minimize errors in the global reference frame while IBVS minimizes errors in the image plane of the camera.

Autonomous navigation for mobile robots is a prevailing topic among robotic researchers. To tackle with challenges, Kalman Filter-based solutions are the leading methods for sensor fusion [4]. Nguyen *et al.* proposed a solution for

autonomous navigation based on Robot Operating System and Gazebo [5]. They used deep learning with simulation data to improve real-world applications. Authors in [6] proposed an improved algorithm with Extended Kalman Filter (EKF) to estimate the state of an Unmanned Aerial Vehicle (UAV) in real time.

To be able to select the optimal streaming method, a comparison was carried out in [7]. After thoroughly comparing H.264 Advanced Video Codec, Dirac, Theora and Motion JPEG2000, the study clearly indicated the advantage of inter-frame comparison in H.264.

Visual aided robots require a well-considered approach for a proper real time stream forwarding. Video streaming over cellular or wireless networks also appears in various fields. A popular and versatile tool for stream forwarding is to utilize GStreamer [8]. A typical use-case of GStreamer is real-time video streaming but it is also useful for acoustic signal processing [9] or even for detecting gravitational waves [10].

Examining the capabilities of a closed-loop control over imperfect networks were investigated by Rácz et al. [11] They worked with an Universal Robots UR5 manipulator to test and measure the performance impact of Ultra Reliable Low Latency Communication (URLLC) capability in 5G networks. As a result, measurements showed that network delay lower than 4 ms has no significant performance impact in case of the robot arm.

### III. SYSTEM OVERVIEW

The selected use case presents a visual guided robot arm manipulation task, where the emphasis is on the near real time control of the manipulator, based on visual information. However, a large variety of different applications can be realized using the same design patterns that the ROS ecosystem provides.



Fig. 1. Realized robotic system: An UR3e industrial robotic arm equipped with a RealSense D435i depth camera.

Our test scenario consists of an UR3e collaborative robotic arm from Universal Robots [12], an Intel RealSense D435i

depth camera [13] and Raspberry Pi4B devices to host the camera driver. As an end-effector for the robotic arm, an OnRobot RG2-FT gripper [14] is applied for pick and place tasks. As Fig. 1 shows, the depth camera is rigidly mounted onto the last joint of the robot arm following the eye-in-hand approach to be able to inspect both the surroundings and the gripped object.

The software implementation is based on the Robot Operating System and follows the cloud robotic aspects as Fig. 2 presents. It provides a framework with the most common communication patterns for a distributed system like publish-subscribe or request-response. The ROS structure consists of a ROS Master running in the cloud, an image forwarding node on a local hardware element (RPi4), and an image processing node placed in the cloud.

Generally, ROS offers raw image transport but it can be extended with plugins to support JPEG or PNG compression. In case of the RealSense camera, an official ROS package [15] is available for retrieving data through topics to work with. Besides ROS, there is an open-source solution for depth cameras to be networked over wired Ethernet or Wi-Fi connection [16]. To help object detection, OpenCV library[17] is used both for colour and depth image processing.

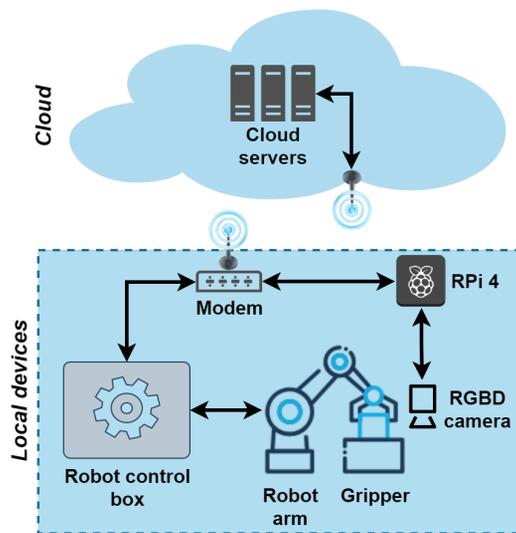


Fig. 2. System architecture.

### IV. SYSTEM REALIZATION

In order to demonstrate the system capabilities, our robot is programmed to build a tower from small identical wooden blocks in a jenga tower style. The task for the robot is to automatically detect jenga pieces on the table, then grab and place them on top of the already built structure.

The detection task includes two different phases. First, when a new jenga block is detected by the camera, its position is acquired in order to be able to pick it up. To achieve this, only the colour frame is used with filtering for colour ranges. The other part that requires the camera is when the robot wants to place the jenga onto the tower being built. It examines the

Optimizing Camera Stream Transport in Cloud-Based Industrial Robotic Systems

target area whether a previous element or elements have been already placed using the depth camera image. Being conscious of the 3D coordinates of the top tower elements will help the actual block to be placed in the right position.

Note here, that neither the pick position nor the follow up tower position is pre-programmed. It is up to the robot to automatically detect and decide, even making it possible for a human to join in and build the jenga tower together with the robot in a cooperative manner. The robot will put its own actual jenga piece on top of the structure, no matter whether the user just placed a new piece atop, or even removed the last one.

A. Image processing

Image processing has to provide reliable actual information to achieve the introduced demonstration task. It requires techniques handling both colour and depth information. The current problem could be separated into two phases: working with colour images and working with depth images.

1) *Colour image*: The task related to the colour image is to find a black jenga block on the table, and if there are more, choose one considering whether the robot can reach it or not. It is based on a robust colour segmentation process in HSV colour space. The black colour of the jenga block and its non-reflecting surface makes accurate filtering for black colour possible. It also excludes objects where there is a measurable difference of the length of the sides and the area of the object in sight. To prove its robustness, it has been successfully tested with strong light sources and with different lighting conditions. With OpenCV, the orientation and exact position have been extracted from the detection image. These are then sent to the robot to grip the object at the right pose.

2) *Depth image*: From the depth image, the goal is to extract the accurate distance and provide an illustrative image about the tower level measurement. Intel gives examples for depth images processing [18]. Calculating the Z distance from the depth image is helped by a function that returns the depth distance from a 2D pixel coordinate. First, an edge-preserving spatial filter is applied, which smooths the depth noise while attempting to preserve edges. In a noisy measurement it will smooth the data but could result in unwanted artifacts such as rounded or elongated edges. It is followed by temporal filtering and hole filling. It is also advised to do whenever possible, making sure not to let holes—where the depth equals zero—influence calculations. It is done by an exponential moving average (EMA) filter which is also used for spatial filtering. With an accurate parameter, we tried to reduce temporal smoothing near edges and also exclude holes.

Fig. 3 presents the before and after phases of jenga depth detection. It clearly shows that the holes were eliminated, and the contours became more accurate.

3) *Equalization*: Because the gripper was too close to the camera, it distracted the depth measurement. Thus, the task was to equalize the histogram of the depth image, because a 1 cm difference in a 16 bit image cannot be easily distinguished. To improve the detection, histogram equalization was only applied to a region of interest, where jenga block could



Fig. 3. Depth images of two jenga pieces as seen from above, before (left) and after (right) image processing.

occur. In Fig. 4 the distances of the tower corners are presented. Due to a shallow depth, histogram equalization results in displaying small differences with significantly changing colours. The distances are calculated as the median value of the intersecting jenga block marked with the white dots.

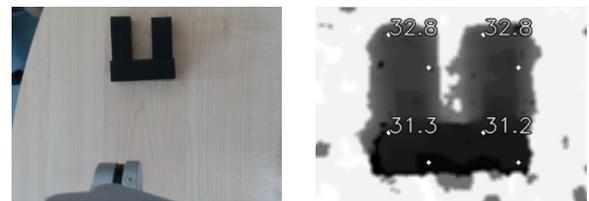


Fig. 4. Depth histogram equalization: raw camera image (left) and equalized depth image (right).

B. Network setup

All the components of the robotic system are prepared to connect and communicate using a cloud architecture with wireless radio access (such as 5G). Fig.5 depicts the distributed software architecture of the system.

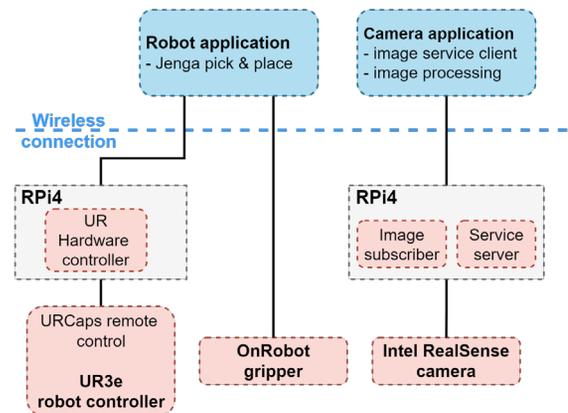


Fig. 5. Distributed software architecture with cloud elements.

1) *Robot arm with cloud control*: Although the UR3e controller can be connected directly to the network, the challenging part is to perform the low-level (servo-)control the robot from the (edge-)cloud. Due to its ultra-low latency requirement, it is not yet possible to run the low-level control from the cloud. (Later, with the improvement of the experi-

mental 5G network parameters, it could also be moved to the cloud.)

2) *Camera and cloud image processing:* A Raspberry Pi device (RPi4) serves as a gateway and driver host for the RealSense camera, since the camera hardware lacks any direct network connectivity.

Note here, that although the publish-subscribe scheme publishes every camera frame, the image processing takes only one frame from the stream to work on. hence, there is no need to transfer the camera stream to the cloud continuously. Taking this into account, live streaming can be reduced to give a frame only when it is required. Because of the camera has a warm-up time, the idea is to combine the publish-subscribe communication scheme with a service-based request-response phase to fetch the image frame as Fig. 6 presents.

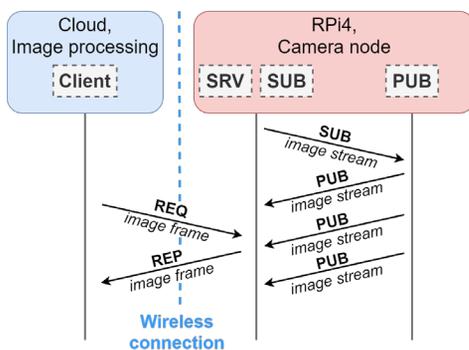


Fig. 6. Combined publish-subscribe and request-response (aka. client-server) communication for acquiring an image from the camera for cloud processing.

According to this enhancement, the camera driver node running in the background always provides the camera stream, but it will not transfer the stream continuously over the network but only forwards it to a local service server. This server provides only one image by grabbing a frame from the published camera stream in case of an incoming request from the cloud. This frame is then processed in the cloud which serve as a basis of further robot movement calculations.

V. STREAMING AND IMAGE TRANSPORT SOLUTIONS

Here we propose a custom method for image transportation that offers low bandwidth and latency. A hybrid method is also introduced with the combination of different network schemes. Alongside with our implementation, popular image transportation methods are introduced that are common in robotic applications.

A. ROS streaming

To transmit data among nodes, ROS by default use serialization and sends raw image objects as string which is a less effective way of image transportation and streaming. As measurements show, it takes 25 MB/s to transfer raw colour and depth images with 30 FPS. ROS plugins offer compression methods for a more effective image transport. It can be configured for PNG or JPEG compression. The compression range in case of the JPEG is [1, 100] where lower values trade

image quality for bandwidth savings. To find the appropriate balance between image size and quality, parameters needs to be fine-tuned according to the application’s needs.

B. Streaming via GStreamer

GStreamer is a current open-source multi-platform multimedia framework with widespread API options like the OpenCV library, OpenGL, RealSense or other community-driven projects [19]. It became popular for streaming audiovisual content and has been frequently used in studies relating to video transmission for its utility and flexibility in the delivery of audio and visual content. The multimedia pipelines of GStreamer are completed through the use of plugins which are assembled using so called pads to form an interconnected framework, moving video from the sink pad of a plugin to act as the source of another.

The common part in each image processing solution is the utilization of OpenCV. To exploit its capabilities, OpenCV is compiled on both sender and receiver side with GStreamer integration. The goal of GStreamer in this project is to separate the application (e.g., video player, video editor) from the streaming media complexity (e.g., hardware acceleration, remoteness). In this case only the video transmission is relevant without the audio track.

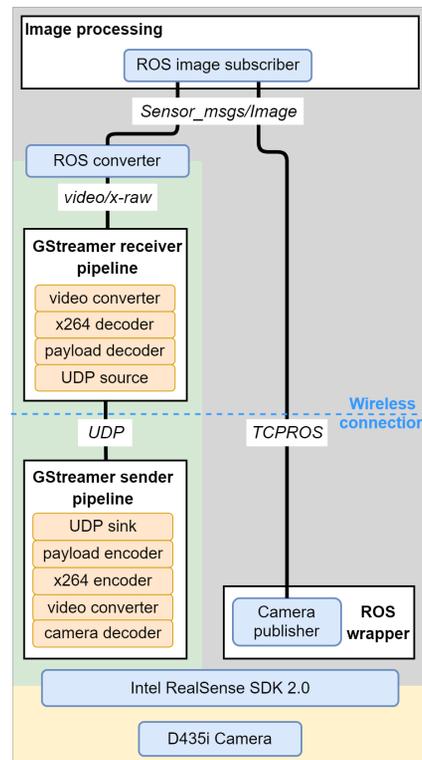


Fig. 7. Streaming and image transport solutions software architecture.

Regarding the restrictions and specialities, the streamer pipeline looks as Fig. 7 presents. Because of the pipeline forwards preprocessed images from OpenCV, the frames inserted directly from script to the GStreamer pipeline. To

Optimizing Camera Stream Transport in Cloud-Based Industrial Robotic Systems

encode camera stream, H.264 encoding is applied. Instead of encoding frames individually, H.264 compress across frames. This inter-frame compression significantly reduces bandwidth consumption because most frames record only the changes from the previous frame. In order to be able to send the H.264 stream over the network, it needs to be encapsulated into Real-time Transport Protocol (RTP) packets [20]. After providing the H.264 payload, it is combined with UDP, and is ready to be sent through the network.

In receiver side the source is an UDP connection with a related port number. Upon receiving the datagram, the decoder extracts H.264 video from RTP packets and sends it to another converter which creates a raw formatted stream with the requested parameters (640x480 resolution, 8-bit grayscale) which could be managed by OpenCV. After the stream is handed over to OpenCV, it is ready for ROS integration. With OpenCV, the image stream is then packed into a ROS compliant *sensor\_msgs/Image* type message. After this conversion, the image stream is being published on a ROS topic.

C. Streaming via EtherSense

Intel provides an open-source solution for its D435i depth camera for connecting to the network by using a Raspberry Pi device [21]. It applies industry standard RTP protocol for streaming. Raspberry Pi only responsible for the transportation part, meanwhile post-processing such as depth decimation or spatial and temporal filtering could remain in the cloud.

According to the measurements, the bandwidth for both colour and depth stream with 640x360 resolution and 30 FPS requires 27.6MB/s. By exploiting the full potential of the camera, with 848x480 resolution and 90 FPS, the required bandwidth reaches 146 MB/s. This option can easily saturate the available networking bandwidth, thus it does not provide a scalable solution for mass-robotic application of visual guidance.

VI. MEASUREMENTS AND PERFORMANCE EVALUATION

Next, we present performance measurement results of our system with the different camera image transport solutions. Besides end-to-end latency as being perhaps the most important transport performance parameter, the required transmission bandwidth is also measured and compared with different network and image parameters, then the advantages and limitations are evaluated.

A. Required bandwidth

An important performance indicator of the realized system is the required bandwidth to operate with. First, the ROS streaming solution is measured, then it is compared with our GStreamer solution to present the results.

For traffic measurement purposes, we used the internal topic bandwidth monitor on ROS side, while for GStreamer traffic the real time Linux network bandwidth tool called Interface TOP (IFTOP) was used. To measure exclusively the image transport, the actual port was filtered where GStreamer is forwarding the stream. All the measurements reflect an average

of 10-minute data traffic.

The performance of ROS streaming is summarized in Fig. 8 where all the measurements were done with 640x480 resolution and 30 FPS. As expected, the raw image transport over ROS has the highest bandwidth requirement. These results are close to Intel’s official EtherSense realization, where for example the depth streaming with similar parameters took 13.88 MB/s.

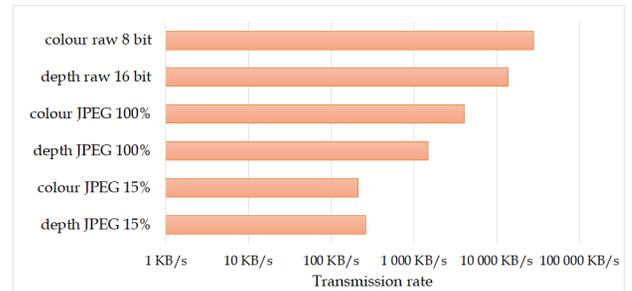


Fig. 8. ROS image streaming comparison.

Although compressed image transport could significantly lower the data rate, the hybrid method could present an even better solution. The idea of providing only one frame on request could minimize the data transfer. However, the image streaming is still running locally in the background, but does not flow into the cloud unless it is requested.

The JPEG compression methods can also be applied to the requested frame as well. This results in such a small image size that could be easily transferred without any difficulties. The size of only one colour frame with 15% JPEG quality is approximately 215 kB, the corresponding depth frame size is 265 kB.

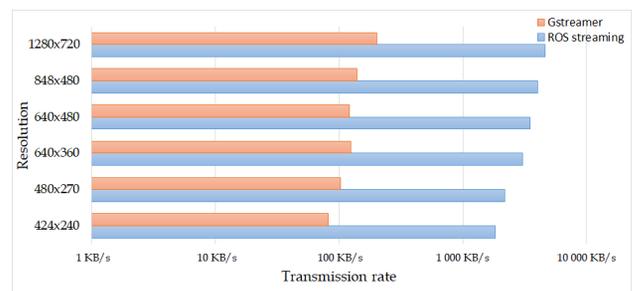


Fig. 9. Depth image streaming comparison.

Comparing these results with the ROS streaming options, GStreamer could perform better up to more than an order of magnitude. Rather than compressing images one by one as ROS does, GStreamer uses H.264 encoding where the inter-frame compression significantly reduces the bandwidth consumption. (Note, that static scenes with mostly still frames encoded with H.264 can result in very low data volumes, therefore the measurements used contain both still scenes and scenes in motion to collect more meaningful data.) GStreamer was set up to forward with constant quality. To show a more revealing chart and visualize the differences, logarithmic

scaling is applied in the figures.

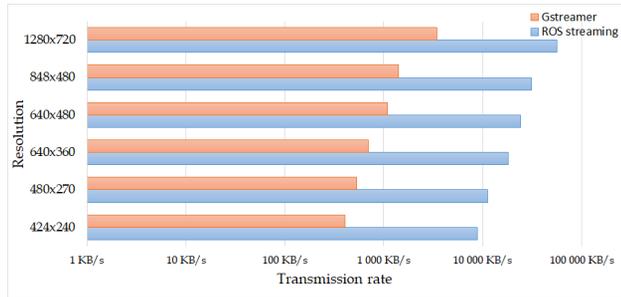


Fig. 10. Colour image streaming comparison.

While examining the streaming of colour images, the parameters and measurement method remained unchanged. Here the results (see Fig. 10) show the overhead of ROS streaming against GStreamer.

Taking into account the image quality and the required bandwidth, it is clearly a trade-off. According to our experiences, among all the available ROS image transport methods, the 15% JPEG option with 640x480 resolution turned out to be the best trade-off between image quality and bandwidth usage, when detecting a jenga block. Thus, the following measurements were all carried out in 640x480 resolution.

*B. End-to-end latency*

For latency measurements, glass-to-glass method was applied: it measures the time it takes between the moment of action happens in front of a camera (first glass), and the moment that a viewer sees the result of this action on the screen (second glass). To determine latency, an online stopwatch were used meanwhile the camera was faced to the notebook screen and forwarded the encoded stream for cloud processing (see Fig. 11). This method makes end-to-end latency measurement possible, including such elements like the camera readout time, encoding and so on. On the other hand, with this method a possible bottleneck is the 60 Hz refreshing rate of the screen. Because of the camera relies on the content of the screen, it could not be more accurate than 1/60 s (approx. 17 msec).

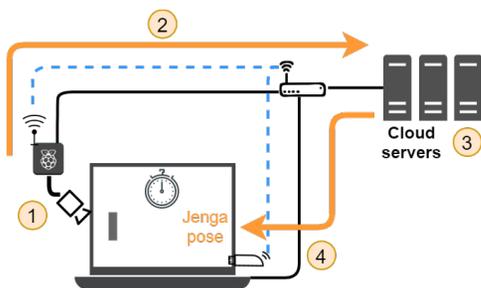


Fig. 11. Latency measurement setup and methodology (wireless connection: blue dotted line, wired connection: straight line).

During the measurement, the camera detects an appearing jenga (see step 1 on Fig. 11) from a pre-recorded video shown

on the laptop screen, which also displays a running stopwatch. The camera stream is then forwarded into the cloud (step 2) where the image processing takes place (step 3). Finally, the result is sent back and displayed on the laptop screen (step 4).

The critical component of an end-to-end latency is the radio transport delay over the wireless link. In order to be able to examine this component, reference measurements were taken where all wireless links were replaced by wired connections (see Fig. 11). Table I presents results collected from 10 different measurements for both the wireless and wired options where the average latency and its standard deviation are listed for the two image transport methods.

TABLE I  
END-TO-END LATENCY MEASUREMENT RESULTS

	ROS transport		GStreamer	
	wireless	wired	wireless	wired
average	2286.9 ms	1123.7 ms	244 ms	222.3 ms
standard deviation	383.1 ms	120.2 ms	66.2 ms	32.3 ms

The highest average delay is measured for ROS transport over wireless. Its value is two times higher than the experienced wired transmission delay. The standard deviation for the wireless ROS transport is also three times as high as for its wired counterpart. However, the end-to-end delay for the GStreamer solution does not show significant difference for the two cases, although the standard deviation is doubled for the wireless case. Note here, that the standard deviation of end-to-end latency can be directly translated to delay jitter, which is an important performance metric for video streaming. The most important finding here is that the average delay for the ROS transport is significantly higher than the GStreamer option.

VII. CONCLUSION

In this paper, the standard ROS-based network image transmission method was examined, and an efficient custom image transport solution based on GStreamer was proposed and evaluated through a demonstration use case. The trade-off between image quality and network transmission bandwidth was highlighted. Comparing our streaming implementation with the industry-standard ROS solution, our method showed better performance by one order of magnitude. Even in wireless systems our method showed significantly lower latency. When continuous image transfer is not required, the bandwidth usage can be even more reduced when—instead of forwarding the whole stream over the network—only a single frame is sent for image processing.

Visual-aided robotics is a rapidly emerging field in Industry 4.0 solutions, thus efficient video streaming and image transmission techniques together with cloud-based image processing will form the backbone of such applications.

ACKNOWLEDGMENT

This work has been supported by the Ericsson-BME 5G Joint Research and Cooperation Project, partly funded by the National Research, Development and Innovation Office, Hungary, project number 2018-1.3.1-VKE-2018-00005.

REFERENCES

[1] Stanford Artificial Intelligence Laboratory et al. Robotic operating system. 05 2018. URL <https://www.ros.org>.

[2] S. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, 1996. **doi:** 10.1109/70.538972.

[3] Chaumette and Seth Hutchinson. Visual servo control, part i: Basic approaches. *IEEE Robotics and Automation Magazine*, 13:82–90, 2006. **doi:** 10.1109/MRA.2006.250573.

[4] Dobrev Yassen, Flores Sergio, and Vossiek Martin. Multi-modal sensor fusion for indoor mobile robot pose estimation. *Position, Location and Navigation Symposium (PLANS)*, pages 553–556, 04 2016. **doi:** 10.1109/PLANS.2016.7479745.

[5] Anh Nguyen, Ngoc Nguyen, Kim Tran, Erman Tjiputra, and Quang D. Tran. Autonomous navigation in complex environments with deep multimodal fusion network. *International Conference on Intelligent Robots and Systems (IROS)*, 07 2020. **doi:** 10.1109/IROS45743.2020.9341494.

[6] Hao Du, Wei Wang, Chaowen Xu, Ran Xiao, and Changyin Sun. Real-time onboard 3d state estimation of an unmanned aerial vehicle in multi-environments using multi-sensor data fusion. *Sensors*, 20:919, 02 2020. **doi:** 10.3390/s20030919.

[7] Till Halbach. Comparison of open and free video compression systems - a performance evaluation. In *Proceedings of the First International Conference on Computer Imaging Theory and Applications - Volume 1: IMAGAPP, (VISIGRAPP 2009)*, pages 74–80. IN-STICC, SciTePress, 2009. ISBN 978-989-8111-68-5. **doi:** 10.5220/0001809700740080.

[8] S. Nimmi, V. Saranya, Theerthadas, and R. Gandhiraj. Real-time video streaming using gstreamer in gnu radio platform. *2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE)*, pages 1–6, 2014. **doi:** 10.1109/ICGCCEE.2014.6922233.

[9] Kipp Cannon, Sarah Caudill, Chiwai Chan, Bryce Cousins, Jolien Creighton, Becca Ewing, Heather Fong, Patrick Godwin, Shaun Hooper, Rachael Huxford, Ryan Magee, Duncan Meacher, Cody Messick, Soichiro Morisaki, Debnandini Mukherjee, Hiroaki Ohta, Alexander Pace, Stephen Privitera, and Madeline Wade. Gstlal: A software framework for gravitational wave discovery. *SoftwareX*, 14:100680, 06 2021. **doi:** 10.1016/j.softx.2021.100680.

[10] Roald Otne, Joachim Eastwood, and Mathieu E.G.D. Colin. Using gstreamer for acoustic signal processing in deployable sensor nodes. pages 1–6, 2015. **doi:** 10.1109/OCEANS-Genova.2015.7271567.

[11] Géza Szabó, Sándor Rác and József Pető. Performance evaluation of closed-loop industrial applications over imperfect networks. *Infocommunications Journal*, XI, 2019. **doi:** 10.36244/ICJ.2019.2.4.

[12] Universal Robots UR3e. [Date accessed: 12-2021] <http://design.ros2.org/articles/rosmiddlewareinterface.html>.

[13] Intel RealSense d435i Specification. [Date accessed: 12-2021] <https://www.intelrealsense.com/depth-camera-d435i>.

[14] OnRobot RG2-FT gripper. [Date accessed: 12-2021] <https://onrobot.com/en/products/rg2-ft-gripper>.

[15] ROS Wrapper for Intel® RealSense™ Devices. [Date accessed: 12-2021] <https://dev.intelrealsense.com/docs/ros-wrapper>.

[16] Alexey Puzhevich, Sergey Dorodnicov, Anders Grunnet-Jepsen and Daniel Piro. *Open-Source Ethernet Networking for Intel® RealSense™ Depth Cameras*. 04 2020.

[17] OpenCV. Open source computer vision library, 2015.

[18] Dave Tong, Anders Grunnet-Jepsen. Depth postprocessing for Intel® RealSense™ D400 depth cameras. 2019. [Date accessed: 12-2021].

[19] Wim Taymans, Steve Baker, Andy Wingo, Ronald S. Bultje, Stefan Kost. Gstreamer application development manual, 2016.

[20] Wenger Stephan, Miska Hannuksela, Thomas Stockhammer, Magnus Westerlund, and D. Singer. Rtp payload format for h.264 video. 03 2005. **doi:** 10.17487/RFC3984.

[21] Anders Grunnet-Jepsen, Philip Krejov. Intel® RealSense™ Depth Camera over Ethernet. 02 2019.



**Marcell Balogh** received his BSc degree in Electrical Engineering from the Budapest University of Technology and Economics (BME). He is specialized in cloud-based autonomous systems. In his early studies, he joined HSN Lab, a University Research Group at the Department of Telecommunications and Media Informatics, BME. Currently, he is pursuing his Master’s degree on visual-aided robotic systems.



**Attila Vidács** received the MSc and PhD degrees from the Budapest University of Technology and Economics (BME) at the Faculty of Electrical Engineering and Informatics, in 1996 and 2000, respectively. His research interests are in the field of cloud robotics, cooperative and modular robot systems, IoT communication technologies, ad-hoc and wireless networking. Currently he is leading the Cloud Robotics Group within HSN Lab.

# Batch-scheduling Data Flow Graphs with Service-level Objectives on Multicore Systems

Tamás Lévai and Gábor Rétvári, *Member, IEEE*

**Abstract**—Data flow graphs are a popular program representation in machine learning, big data analytics, signal processing, and, increasingly, networking, where graph nodes correspond to processing primitives and graph edges describe control flow. To improve CPU cache locality and exploit data-level parallelism, nodes usually process data in batches. Batcher is a scheduler for data flow graph based packet processing engines, which uses controlled queuing to reconstruct fragmented batches inside a data flow graph in accordance with strict Service-Level Objectives (SLOs). Earlier work showed that Batcher yields up to 10x performance improvement in real-life use cases, thanks to maximally exploiting batch processing gains.

Batcher, however, is fundamentally restricted to single-threaded execution. In this paper, we generalize Batcher to parallel execution on multiple CPU cores. We extend the analytical model to the parallel setting and present a primal decomposition framework, where each core runs an unmodified Batcher controller to schedule batch-processing on a subset of the data flow graph, orchestrated by a master controller that distributes the delay-SLOs across the cores using subgradient search. Evaluations on a real software switch provide experimental evidence that our decomposition framework produces 2.5x performance improvement while accurately satisfying delay SLOs that are otherwise not feasible with single-core Batcher.

**Index Terms**—data flow graph, decomposition, software switch, SDN, NFV

## I. INTRODUCTION

**B**ATCH-SCHEDULING is a near-universal technique to improve performance of software packet processing engines: collect multiple packets into a single burst and perform the same operation on all the packets in one shot. Processing packets in batches is much more efficient than processing a single packet at a time, thanks to amortizing one-time operational overhead, optimizing CPU cache usage, and enabling loop unrolling and SIMD optimizations [1], which often yields 2–5x performance boost. Consequently, batching is used in essentially all software switches (*e.g.*, BESS [2], VPP [3], FastClick [4], and ESwitch [5]), high-performance OS network stacks and libraries [6], user-space I/O libraries [7], and Network Function Virtualization (NFV) platforms [8], [9].

Batcher [10] is a state-of-the-art batch-scheduling framework for high-end programmable software switches. Batcher abstracts the software switch dataplane as a data flow graph; here, nodes represent packet-processing primitives (*e.g.*, L3 Lookup) and arcs represent the control flow. This data flow graph is executed in a *run-to-completion* fashion; when a packet-processing

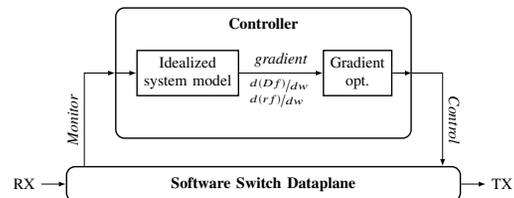


Figure 1. Batcher System Architecture.

node finishes work on a packet batch, execution proceeds on the downstream nodes along all outgoing arcs of the node. Unfortunately, run-to-completion tends to fragment batches inside the data-flow graph, as each node may split the input batch into multiple sub-batches to be passed to downstream nodes; *e.g.*, an L3 Lookup table or a round-robin LoadBalancer may distribute the packets inside the batch across multiple downstream processing chains, a network stack may split a burst of mixed input packets per L3/L4 protocol to execute each MPLS, IPv4 and IPv6 packet on a separate downstream protocol engine, *etc.* Since the downstream modules are executed on smaller batches we lose batch-efficiency, which inherently curtails the available performance, often an order of magnitude lower than with full batches [1].

Batcher attempts to recover some of the lost batch-efficiency by artificially queuing up packets inside the data flow graph to be able to execute the downstream processing nodes on larger batches. Inspired by Nagle’s algorithm [11], Batcher uses a model-predictive controller to regulate queue backlogs for maximizing batch sizes across the pipeline in a way so that the end-to-end queuing delay remains under a given requirement (Fig. 1). This brings massive performance improvement, and delay Service Level Objective (SLO) conformance in the  $\mu s$  range even at million-packet-per-second scale traffic [10]. Unfortunately, the model underlying Batcher assumes single-core execution.

Motivated by the need to run software switches on multicore systems to maximize performance [12], [13], *in this paper we extend Batcher to leverage parallel execution.* As Fig. 2 shows, this is not trivial. The task is two-fold: *i)* find an optimal batch-schedule on each core, and *ii)* distribute delay budgets among cores in a way so that the end-to-end delay remains under the SLO. This is a two-level optimization problem: on per core basis the goal is to find the optimal queue backlog sizes and on a higher level to determine how long each core can process a packet batch so to meet end-to-end delay SLOs. To solve this complex multi-level problem, we propose a decomposition technique [14].

The general idea of decomposition is to break a complex problem into simpler subproblems, then solve the simple

T. Lévai is with the Department of Telecommunications and Media Informatics at the Budapest University of Technology and Economics. G. Rétvári is with the MTA-BME Information Systems Research Group and Ericsson Research. This work was supported by the NKFIH/OTKA Project #135606. E-mail: {levait, retvari}@tmit.bme.hu.

## Batch-scheduling Data Flow Graphs with Service-level Objectives on Multicore Systems

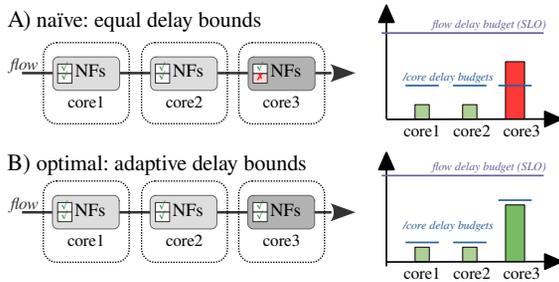


Figure 2. Motivating example for multicore Batchy [10]. The pipeline runs on 3 cores and serves a single flow. NFs on each core require a given amount of time to process a full packet batch (core1: 1, core2: 1, and core3: 4 units). Note that per-core delays add up, so that a flow’s end-to-end delay equals the sum of the delay imposed on the flow’s packets at each core. *a*) Naïve approach: no coordination between the CPU cores. This yields limited performance since the delay on core3 always exceeds the per-core delay budget and hence there is no room to reconstruct batches. *b*) Optimal adaptive per-core delay budget distribution: core3 now gets a higher delay budget than the rest of the cores. Per-core delay budgets are now satisfied and there is enough delay budget to efficiently defragment batches on core3, which then yields significant performance improvement.

subproblems separately under the control of a global problem that takes care of the “complicating constraints”. This technique was already adapted to many networking domains, such as network utility maximization [15], radio transceiver design [16], and beamforming [17]. The goal of decomposition in Batchy is to split the global scheduling problem among the cores (*i.e.*, CPUs) in a multicore system, so that each core autonomously optimizes batch sizes across a subset of the data flow graph subject to a per-core flow delay budget, with minimal switch-level orchestration that adjusts the delay budgets per each core to meet the global delay SLOs. The per-core controller will be conveniently implemented by the unmodified single-core Batchy algorithm. This setup reflects a *primal decomposition* [14] structure.

Our contributions in this paper are as follows:

**Analytical model.** After a short recap<sup>1</sup> on Batchy (§II), we introduce an expressive mathematical model for SLO-based batch-scheduling on multicore software switches (§III). Our framework allows to formally reason about the performance and adaptively distribute end-to-end delay SLOs across cores to maximize performance.

**Control algorithms.** We design control algorithms for effective multicore batch-scheduling under delay SLOs (§IV).

**Design, implementation, and evaluation.** We present a practical implementation of the multicore scheduling framework by extending Batchy and using the BESS software switch [2] (see §IV). We demonstrate the effectiveness of our control algorithms in a realistic use case, VRF (Virtual Routing Function), taken from an official industry 5G NFV benchmarking suite [13]. We show that our control algorithms increase total packet rate by up to 2.5× beyond what is available with single-core Batchy, while meeting delay SLO requirements that are otherwise not feasible with single-core Batchy. Our implementation is available for download at [18].

We close the paper discussing related work (§VI) and deriving the main conclusions (§VII).

<sup>1</sup>In this paper we only introduce Batchy essentials due to space constraints. For Batchy details, we kindly refer the reader to the Batchy paper [10].

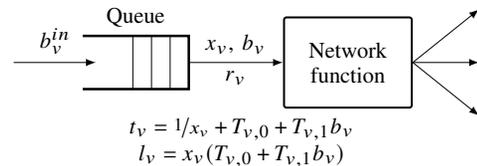


Figure 3. A Batchy Module.

## II. BATCHY SYSTEM MODEL

Next, we introduce our analytical model. We mostly reproduce the main ideas from the single-core setting, highlighting the extensions we introduce for the multicore setting.

## A. Concepts

**Data flow graph.** We model the pipeline as a directed graph  $\mathcal{G} = (V, E)$ , with modules  $v \in V$  and directed links  $(u, v) \in E$  representing the connections between modules. A *module*  $v$  is a combination of a (FIFO) *ingress queue* and a *network function* at the egress connected back-to-back (see Fig. 3). *Input gates* (or *ingates*) are represented as in-arcs  $(u, v) \in E : u \in V$  and *output gates* (or *outgates*) as out-arcs  $(v, u) \in E : u \in V$ . A batch sent to an outgate  $(v, u)$  of  $v$  will appear at the corresponding ingate of  $u$  at the next execution of  $u$ . Modules never drop packets; we assume that whenever a module (*e.g.*, access control) would drop a packet it will rather send it to a dedicated “drop” gate, so that we can account for lost packets.

**Batch processing.** Packets are injected into the ingress, transmitted from the egress, and processed from outgates to ingates along data flow graph arcs, in batches [2], [5], [7]. We denote the maximum batch size by  $B$ , a system-wide parameter. For the Linux kernel and DPDK  $B = 32$  or  $B = 64$  are usual settings, while GPU/NIC offload often works with  $B = 1024$  or even larger to maximize I/O efficiency [8], [19].

**Module service time profile.** After extensive evaluation of network functions on various software switches, we observe two distinct execution time components. The *per-batch cost component*, denoted by  $T_{v,0}$  [sec] for a module  $v$ , characterizes the constant cost that is incurred just for calling the module on a batch, independently from the number of packets in it. The *per-packet cost component*  $T_{v,1}$ , [sec/pkt], on the other hand, models the execution cost of each individual packet in the batch. Accordingly, we shall use the linear approximation  $T_v = T_{v,0} + T_{v,1}b_v$  [sec] to describe the execution cost of a module  $v$  where  $b_v$  is the batch-size, *i.e.*, the average number of packets in the batches received by module  $v$ .

**Module types.** Any module may have multiple ingates (merger) and/or multiple outgates (splitter), or may have no ingate or outgate at all. An L3 Lookup module would distribute packets to several downstream branches, each performing group processing for a different next-hop (splitter); a NAT module may multiplex traffic from multiple ingates (merger); and an IP Checksum module would apply to a single datapath flow (single-ingate–single-outgate). Certain modules are represented without ingates, such as a NIC receive queue; we call these *ingress modules*. Similarly, a module with no outgates (*e.g.*, a transmit queue) is an *egress module*.

**Compute resources.** A task ( $t \in \mathcal{T}$ ) is our main compute resource abstraction. Tasks are modeled as a connected sub-graph  $\mathcal{G}_t = (V_t, E_t)$  of  $\mathcal{G}$ , with strictly one ingress module representing an ingress queue that buffers packets between subsequent executions of the task. We assume that when a data flow graph has multiple ingress modules then each ingress is assigned to a separate task, with packets passing between tasks over double-ended queues. Each task uses run-to-completion scheduling, and there is a separate CPU core assigned per task. Consequently, in a multicore scenario we have as many tasks as there are cores.

**Flows.** A flow  $f = (p_f, R_f, D_f)$ ,  $f \in \mathcal{F}$  is an abstraction for a service chain, where  $p_f$  is a path through  $\mathcal{G}$  from the flow’s ingress module to the egress module,  $R_f$  denotes the offered packet rate at the task ingress, and  $D_f$  is the delay SLO, the maximum permitted latency for any packet of  $f$  to reach the egress. What constitutes a flow, however, will be use-case specific: in an L3 router a flow is comprised of all traffic destined to a single next-hop or port; in a mobile gateway a flow is a complex combination of a user selector and a bearer selector; in a programmable software switch flows are completely configuration-dependent and dynamic. In our framework flow dispatching occurs *intrinsically* as part of the data flow graph; accordingly, we presume that match-tables (splitters) are set up correctly to ensure that the packets of each flow  $f$  will traverse the data flow graph along the path  $p_f$  associated with  $f$ . During this traversal, flow goes through tasks. A *taskflow* is a part of a flow that is executed on a single task.

B. System Variables

We use a fluid model. Thus, variables are continuous and differentiable, describing system *statistics* over a longer period of time that we call the *control period*. We use the following variables to describe the state of the data flow graph in a given control period (dimensions indicated in brackets). The variables needed for the multicore extension are marked by  $\blacklozenge$ .

**Batch rate**  $x_v$  [1/s]: the number of batches per second entering the network function in module  $v$  (see again Fig. 3).

**Batch size**  $b_v$  [pkt]: the average number of packets per batch at the input of the network function in module  $v$ , where  $b_v \in [1, B]$  (recall  $B$  is the maximum allowed batch size).

**Packet rate**  $r_v$  [pkt/s]: the number of packets per second traversing module  $v$ :  $r_v = x_v b_v$ .

**Maximum delay**  $t_v$  [sec]: delay contribution of module  $v$  to the total delay of packets traversing it. We model  $t_v$  as

$$t_v = t_{v,\text{queue}} + t_{v,\text{svc}} = 1/x_v + (T_{v,0} + T_{v,1}b_v) \quad , \quad (1)$$

where  $t_{v,\text{queue}} = 1/x_v$  is the queuing delay by Little’s law and  $t_{v,\text{svc}} = T_{v,0} + T_{v,1}b_v$  is the module service time profile.

**System load**  $l_v$  (dimensionless): the network function in module  $v$  with service time  $t_{v,\text{svc}}$  executed  $x_v$  times per second incurs  $l_v = x_v t_{v,\text{svc}} = x_v (T_{v,0} + T_{v,1}b_v)$  system load on its task.

$\blacklozenge$  **Task turnaround-time**  $\tau_t$  [sec]: Turnaround-time of task  $t$  is the time while task  $t$  processes a packet batch. This is the multicore equivalent of the *turnaround-time* (see [10] for details). We consider the time to execute *all* task modules on maximum sized batches as an upper bound:

$$\tau_t \leq \sum_{v \in V} (T_{v,0} + T_{v,1}B) \quad \forall v \in V_t \quad . \quad (2)$$

$\blacklozenge$  **Taskflow**  $\pi$ : For each flow  $f \in \mathcal{F}$ ,  $\pi_f$  is a list of tasks the packets of  $f$  traverse in the data flow graph.

$\blacklozenge$  **Per-task flow delay budget**  $\Delta_{t,f}$  [sec]: delay allocated for a taskflow of flow  $f$  in task  $t$ ; *i.e.*, the maximum delay allowed for a flow to traverse a task. A column vector representing delay budgets for each flow of task  $t$  is noted as  $\Delta_t$ .

C. Assumptions

Our aim is to define the simplest possible batch-processing model that still allows us to reason about flows’ packet rate and maximum delay, and modules’ batch-efficiency. The below assumptions will help to keep the model at the minimum; see [10] for a detailed justification and several ideas to overcome them. New assumptions added for the multicore setting are marked by  $\blacklozenge$ .

**Feasibility.** We assume that the pipeline runs on a single task and this task has enough capacity to meet the delay SLOs.

**Buffered modules.** We assume that all modules contain an ingress queue and all queues in the pipeline can hold up to at most  $B$  packets at any point in time.

**Static flow rate.** All flows are considered constant-bit-rate during the control period (usually in the millisecond time frame).

$\blacklozenge$  **Task-exclusive modules:** Each module is assigned to exactly one task. If a module needs to present in multiple tasks, it will be replicated for each task.

III. BATCHY DECOMPOSITION

Decomposition is a general framework for breaking down complex optimization problems into simple *subproblems*, which are assumed to be easy to solve in separation, and a *global problem* that orchestrates the subproblems and takes care of the “complicating constraints” [14]. Each subproblem is defined in terms of a set of *private variables*, which appear only in this subproblem, and a set of *public variables* that are common to multiple subproblems. The problem is solved iteratively: first we fix the public variables and solve each subproblem separately to find the optimal setting of the private variables under the current setting of the public variables, and then in a “master step” we update the public variables and start a new iteration. The update drives the system in a direction so that the global objective is improved, *e.g.*, moving along the objective function gradient with a pre-defined step size. Depending on the type of the public variables, we distinguish *primal* decomposition and *dual* decomposition frameworks. In primal decomposition the public variables are primal variables, while in dual decomposition the subsystems are manipulating dual variables (*i.e.*, prices) of the global problem.

To demonstrate the two methods, consider an example of a printed circuit board, where the board is the global system and the integrated circuits on the board are the subsystems. Suppose we want to design a complex circuit from subcircuits (*e.g.*, integrated circuits), and our goal is to minimize the overall power usage. Subcircuits have properties, some of them are not

relevant to how they connect to each other (*e.g.*, dimensions), some are important in the interconnection (*e.g.*, power usage). In this case, we say dimension is a private variable, and power usage is a public variable of the subcircuits. Then, in primal decomposition we fix the amount of power usage available to each subcircuit and design the subcircuits according to that specification. Then, we update the public variables (*i.e.*, the subcircuits' power budgets) in a way as to improve the overall power usage and then we restart the iteration, by redesigning the subcircuits (*i.e.*, solving the subproblems) subject to the new power budget. In dual decomposition, we allow subcircuits to choose how much power they want to use, however, each subcircuit has to "pay" a certain price for power usage. The price depends on the system-wide power budget: when the current power usage is low the prices are also low, but as the system's power constraints become more and more tight so do the per-unit power usage price goes up. In the global step we set the prices in a way to improve the design.

In general, if the global problem is optimized using the subgradient method then decomposition methods are guaranteed to converge "close" to the optimum, even for a constant step size [14]. With a diminishing step size rule, arbitrary close convergence to the optimum is guaranteed in finite steps.

#### A. Batchy: Multicore System Model

We extend Batchy to the multicore setting by formulating the global problem for multiple cores and then applying primal decomposition to the system to obtain per-core controllers. The global problem sets the per-core delay budgets so that end-to-end delay SLOs are met and the total system load is minimized. Subproblems in turn control the batch size over a partition of the data flow graph, subject to the delay budgets set by the global problem. Then, private variables are the per-module queue sizes while the public variables are the per-task delay budgets (*i.e.*, the maximum time allowed for processing a flow in a task). In the following sections we provide further detail.

First, we recap the original Batchy model implementing single-core execution [10]. As (3) shows, we express system load  $L$  as a function of queue backlogs while conforming delay requirements (4) and queue sizing limits (5) in a single task.

$$L = \min \sum_{v \in V} \frac{R_v}{b_v} (T_{0,v} + T_{1,v} b_v) \quad (3)$$

$$\text{s.t. } \tau + \sum_{v \in P_f} \left( \frac{b_v}{R_v} + T_{v,0} + T_{v,1} b_v \right) \leq D_f \quad f \in F \quad (4)$$

$$1 \leq b_v \leq B \quad v \in V \quad (5)$$

Next, we extend the single-core model to the multicore setting. For this purpose, we break up the data flow graph to tasks, under the assumptions of §II-C. The problem decomposes on a per-task basis as shown in (6)–(10).

$$L = \min \sum_{t \in \mathcal{T}} \sum_{v \in V_t} \frac{R_v}{b_v} (T_{0,v} + T_{1,v} b_v) \quad (6)$$

$$\text{s.t. } \tau_t + \sum_{v \in P_{f,t}} \left( \frac{b_v}{R_v} + T_{v,0} + T_{v,1} b_v \right) \leq \Delta_{t,f} \quad f \in F, t \in \mathcal{T} \quad (7)$$

$$\sum_{t \in \mathcal{T}} \Delta_{t,f} \leq D_f \quad f \in F \quad (8)$$

$$1 \leq b_v \leq B \quad t \in \mathcal{T}, v \in V_t \quad (9)$$

$$\Delta_{t,f} \geq 0 \quad f \in F, t \in \mathcal{T} \quad (10)$$

Next, we show the global and subproblem objectives of our decomposition. In this context, we use the term *problem* and *task* interchangeably due to the per-task decomposition.

#### B. Global Problem

In our primal decomposition structure the global problem is responsible for distributing the flow delay budgets among tasks in a way to minimize system load (11). We also need to ensure the sum of per-task delay budgets are not over the flow delay budget (12) and each task will receive non-negative flow delay budgets (13).

$$L = \min \sum_{t \in \mathcal{T}} L_t(\Delta_t) \quad (11)$$

$$\text{s.t. } \sum_{t \in \mathcal{T}} \Delta_{t,f} \leq D_f \quad f \in F \quad (12)$$

$$\Delta_{t,f} \geq 0 \quad (13)$$

#### C. Subproblems

Subproblems optimize task performance, while keeping delays under the per-task flow delay budgets assigned by the global problem. We observe that the resultant control problem is effectively the same as the single-core control problem (3)–(5). Therefore, we will mostly reuse the original Batchy controller from [10] with minimal changes to handle the private/public variables and per-core delay budgets.

We need a framework to distribute the per-flow delay budgets  $\Delta_{t,f}$  across the tasks  $t \in \mathcal{T}$  traversed by  $f$ . Correspondingly, for every flow there is a dedicated *leader* task that sets the per-task delay budgets, and zero or more *follower* tasks that merely track the budgets assigned by the leader. Each task may be a leader for any flow and follower for others. We categorize tasks  $\forall t \in \mathcal{T}$  in the system:

- $\Omega_t = \{f : t \text{ is the leader for } f\}$ ,
- $\Psi_t = \{f : t \text{ is a follower for } f\}$ .

The fundamental difference between leaders and followers is that a leader keeps track of the per-task flow delay budget subgradients along the flow path:  $\Theta_{t,f} : f \in \Omega_t, s \in \mathcal{T} : f \in \Psi_s$ . Leaders use both subgradients and queue size backlogs  $b_v : v \in V_t$  as private variables. Likewise, followers use queue backlog sizes as private variables, and per-task flow budgets  $\Delta_{t,f}$  as public variables.

Take the pipeline of Fig. 2 as an example; we have a single flow  $f_1$  passing over 3 tasks  $\mathcal{T} = t_1, t_2, t_3$ . Select  $t_3$  as the leader of  $f_1$ , so  $\Omega_{t_3} = \{f_1\}$ . Consequently,  $t_1$  and  $t_2$  will be followers of  $f_1$ . Leader private variables are the delay budget subgradients  $\Theta_{t_1, f_1}$  and  $\Theta_{t_2, f_1}$ , and the queue backlog sizes  $b_v, v \in V_{t_3}$ . Followers tasks optimize their private variables  $b_v$  according to public variables:  $\Delta_{t_1, f_1}$  or  $\Delta_{t_2, f_1}$ .

The subproblem objective function (14) minimizes task load; in this manner it is equivalent to the single-core objective

function. Private delay budget variables  $\Theta_{t,f}$  are not effecting the task load, therefore are omitted from the objective function.

$$L_t(\Delta_{t,f}) = \min l_t = \min \sum_{v \in V_t} \frac{R_v}{b_v} (T_{0,v} + T_{1,v} b_v) \quad (14)$$

The objective function is subject to the following constraints. For both leader and follower problems, the batch size limiting constraint (15) applies.

$$1 \leq b_v \leq B \quad v \in V \quad (15)$$

Additionally, flows passing the task must meet their delay SLO requirement. The constraints are slightly different for leader and follower problems. As of follower problems, constraint (16) keeps per-task flow delays under the budget  $(\Delta_{t,f})$ . Recall, these budgets come from the global problem (11).

$$\tau_t + \sum_{v \in P_{f,t}} \left( \frac{b_v}{R_v} + T_{v,0} + T_{v,1} b_v \right) \leq \Delta_{t,f} \quad f \in \Psi_t \quad (16)$$

Leader problems have multiple delay constraints. First, constraint (17) ensures compliance with delay SLOs of both taskflows and flows. This is doable since leader tasks have a view on private delay variables  $(\Theta_{t,f})$ . Second, constraint (18) ensures equivalence between public and private delay variables.

$$\tau_t + \sum_{v \in P_{f,t}} \left( \frac{b_v}{R_v} + T_{v,0} + T_{v,1} b_v \right) + \sum_{s \in \mathcal{T}: f \in \Psi_s} \Theta_{t,f} \leq D_f \quad f \in \Omega_t \quad (17)$$

$$\Theta_{t,f} = \Delta_{t,f} \quad f \in \Omega_t, s \in \mathcal{T} : f \in \Psi_s \quad (18)$$

#### IV. CONTROL ALGORITHMS

In this section we present efficient control algorithms to solve both the global problem and the subproblems. These algorithms are suitable for a real-life implementation.

##### A. Solving Subproblems

Batchy uses a controller based on the gradient projection method of Rosen [21]. The Rosen method is compatible with our decomposition: it handles equality-type constraints (18) and generates gradients and dual variables for the subgradient method, which are used in the subgradient step for solving the global problem (see later in §IV-B).

Let us briefly recap the gradient projection method. The method consists of three main steps: *i*) find an improving direction; *ii*) find a suitable step size; *iii*) optimize along the direction with the step size. In the first step, we obtain an improving feasible direction by projecting the gradient of the objective function into the feasible space using a projection matrix. The projection matrix  $\mathbf{P}$  ensures that the resultant update will not violate the per-task delay budgets. For this end, we show the construction of variable coefficients matrix  $\mathbf{M}$  and the projection matrix  $\mathbf{P}$ :

- Let  $\mathbf{M}_1 = [AB]$  be a matrix where  $A$  is a matrix in which row  $i$  reflects the effect of increasing queue backlog sizes ( $b_v$ ) on  $i$ -th flow delay in  $\mathcal{F}$  with tight delay constraints from (16) and (17), and  $B$  is a zero matrix corresponding to the private variables  $\Theta_{t,f} : f \in \Omega_t, s \in \mathcal{T} : f \in \Psi_s$ .

- Let  $\mathbf{M}_2 = [ZQ]$  where  $Z$  is a zero matrix with as many rows as there are constraints in (18) and as many columns as the number of task modules  $|V_t|$  (corresponding to  $b_v$  variables).  $Q$  is a matrix with row  $i$  set to 1 where  $\Delta_{t,f}$  is the  $i$ -th taskflow in a list of taskflows  $t, f : f \in \mathcal{F}, t \in \mathcal{T}$ .
- Let  $M^T = [M_1^T M_2^T]$ .
- Then, we construct  $\mathbf{P}$  as  $\mathbf{P} = \mathbf{I} - M^T (MM^T)^{-1} M$ .

The subproblem control algorithm reuses the single-core Batchy control algorithm with the new projection matrix  $\mathbf{P}$ . The control algorithm generates the duals of private delay variables  $\Theta_{t,f}(\omega)$  for the sugradient method by the leader task of flow  $f$ . We summarize the per-task projected gradient control algorithm we use to solve the subproblems in Algorithm 1.

Unfortunately, the control algorithm cannot handle an infeasible state; *i.e.*, a state where the SLOs cannot be met. To recover the system from infeasibility we introduce a simple heuristic: the subsystems reuse the feasibility-recovery mechanisms from single-core Batchy [10], while multicore feasibility-recovery is implemented in the global controller (§IV-B).

---

#### Algorithm 1 Projected Gradient Control Algorithm

---

```

procedure PROJECTEDGRADIENT( $\mathcal{G}, \mathcal{F}, \Delta_t, f$ )
   $\triangleright$  Gradient projection
  while True do
     $\mathbf{P} = \mathbf{I} - \mathbf{M}^T (\mathbf{M}\mathbf{M}^T)^{-1} \mathbf{M}$ 
     $\Delta \mathbf{b} = \mathbf{P}\nabla l_t$   $\triangleright \Delta \mathbf{b}$  has useless coordinates corresponding to
    private variables  $\Theta_{t,f}$ 
     $\mathbf{w} = -(\mathbf{M}\mathbf{M}^T)^{-1} \mathbf{M}\mathbf{V}\mathbf{l} = [u, \omega]$   $\triangleright u$  corresponds to  $b_v$  and
     $\omega$  corresponds to  $\Theta_{t,f}$ 
    if  $\Delta \mathbf{b} \neq 0$  then break
    if  $u \geq 0$  then return  $\triangleright$  Optimal KKT point reached
    delete row for  $f$  from  $\mathbf{M}$  for some  $f \in \mathcal{F} : w_f < 0$ 
   $\triangleright$  Line search
  for  $v \in V, f \in p_v$  do
    if  $\Delta b_v > 0$  then
       $\lambda_v = \min_{f \in \mathcal{F}: v \in p_f} \left\lfloor \frac{\Delta_{t,f} - \tilde{l}_f}{\Delta b_v} \right\rfloor$ 
   $\lambda = \min_{v \in V} \lambda_v$ 
  for  $v \in V$  do SETTRIGGER( $v, b_v + \Delta b_v \lambda$ )
  
```

---

##### B. Solving The Global Problem

Subproblems are handled by the Batchy projected gradient controller at each control period. After every  $N$  iteration, the global problem controller kicks in to reallocate the per-task delay budgets (*i.e.*, the public variables  $\Delta_{t,f}$ ).

The global control algorithm relies on two types of inputs: the duals  $\omega_n$  of the subproblem constraints (16), and duals  $\omega_m$  of constraints corresponding to private variables in (18). Gradients  $g$  are obtained by summing global and subproblem subgradients pairwise:  $g_{n,f} = \omega_m + \omega_n \forall f \in \Omega_m, n \in \mathcal{T} : f \in \Psi_n$ . Based on these inputs, the global control algorithm (Algorithm 2) first calculates a step size, then updates per-task delay budgets for each flow. For simplicity, the algorithm uses a fix step size calculated as a configurable percentage  $\delta$  of flow delay  $D_f$ .

We apply a simple heuristics to prevent infeasible states in the global problem. We collect taskflows that exceed their delay budget and increase their budget with a configurable and fixed percentage of flow delay, balancing this delay increment

## Batch-scheduling Data Flow Graphs with Service-level Objectives on Multicore Systems

**Algorithm 2** Global Control Algorithm

```

procedure SUBGRADIENT GLOBAL STEP( $\mathcal{G}, \mathcal{F}, \pi, \delta, g$ )
for  $f \in \mathcal{F}$  do
     $\triangleright$  Calculate step size
     $\alpha = D_f * \delta$ 
     $\triangleright$  Update allocated per-task flow delays on  $\pi_f$ 
    for  $t \in \pi_f$  do
         $\Delta_{t,f} = \Delta_{t,f} + \alpha * g_{t,f}$ 
    
```

 Table I  
 STEADY-STATE RESULTS (SIMPLE PIPELINE).

	Rate [Mpps]	Delay (p99) [ $\mu$ s]
No Batching	0.971	15.445
Static Delay Budgets	0.991	18.615
<i>Multicore Batching</i>	<i>1.348</i>	<i>11.255</i>

by decreasing surplus budgets of the feasible taskflows. This simple technique is sufficient to ensure flow delay SLOs (12) and non-negative per-task flow delay budgets (13).

## V. EVALUATION

In this section, we evaluate our Batching multicore extension on both synthetic example and real-life use-case. We reused existing Batching codebase [10] as a controller to solve the per-task subproblems (Algorithm 1) and implemented the subgradient controller to orchestrate the per-task Batching controllers (Algorithm 2). The source code is available on GitHub [18]. The evaluation was running on a server with 6x2.4GHz CPU (power-saving disabled) and 64GB RAM installed with Debian 11 GNU/Linux.

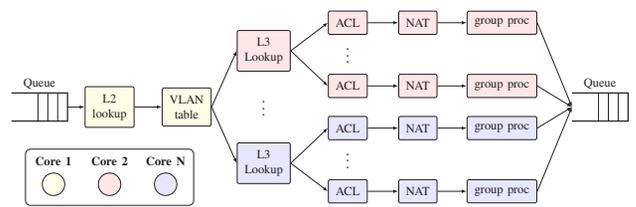
## A. Concept Validation: A Simple Pipeline

The first evaluation scenario focuses on validating the concept.

**Evaluation setup.** We use a simple pipeline of two tasks connected back-to-back. Tasks run on different cores and contain one module. The system has one flow that traverses both tasks. The last module is a computation-heavy module that requires significantly more per-batch processing time (tens of thousands of CPU cycles) than the first module (hundreds of CPU cycles). This pipeline is similar to the example in Fig. 2.

We compare multicore Batching to two baselines. The first baseline does no packet batching. The second baseline runs Batching, but does not adjust per-task delay budgets adaptively; *i.e.*, adopts the naïve approach of Fig. 2. The measurements focus on steady state performance: the first 100 control periods are considered as warmup time, and we focus on the next 100 control periods. The flow delay SLO is set to 12 $\mu$ s.

**Results.** Table V-A summarizes steady packet rate and 99th percentile delays of the measurements. The two baselines produce limited packet rate due to poor batch-scheduling algorithms. Namely, baselines cannot mitigate the cost of computation-heavy task by intensive batching. There is a slight difference between the performance of the two baselines: in case of *static delay budgets*, Batching has enough room for batching in the first task, yielding a slight overall improvement of the packet rate at a 20% delay penalty. In contrast to


 Figure 4. The Virtual Routing Function Pipeline on  $N$  Cores.

baselines, *multicore Batching* can distribute the global delay bound across the tasks optimally, so that it assigns extra delay budget surplus for the last task that enables it to execute the computation-heavy module on larger batches. This optimization improves throughput by 30% while decreases delay by 60%, and makes multicore Batching the only solution to meet the flow delay SLO.

To sum up, this experiment highlights the importance of batching in multicore scenarios. However, careful distribution of delay budgets among processing cores is necessary to get the most of batch-efficiency gains.

## B. Case Study: Virtual Routing Function

We demonstrate the real-life applicability of multicore Batching on a sample use case, the Virtual Routing Function (VRF), taken from an official 5G benchmarking suite [13]. In this measurement we are focusing on the following questions: *i*) can we decrease the delay compared to single-core Batching; *ii*) how efficient is the decomposition-based delay budget distribution compared to a naïve approach; *iii*) how much extra processing is required for the hierarchical control?

**Evaluation setup.** The VRF pipeline (Fig. 4) implements a latency-optimized L2/L3 routing scenario often arising in the context of network function virtualization. In addition to L2/L3 routing, the pipeline also performs access control and address translation over multiple virtual LANs (VLANs). First, traffic is split per VLANs, and then for each VLAN the next hop is selected using longest-prefix matching (L3 Lookup). For each next hop, traffic undergoes access control (ACL), address translation (NAT), and group processing. The pipeline has two parameters: the number of VLANs ( $n$ ), and the number of next-hops per VLAN ( $m$ ). The pipeline is provisioned on  $n + 1$  cores: VLAN splitting is done on the first core, and per-VLAN traffic is processed on the remaining  $n$  cores.

For the evaluation, we use the VRF(2,4) pipeline (2 VLANs and 4 next-hops/VLAN). We set a 72 $\mu$ s delay SLO for all flows. The system runs for 60 control periods, and each control period takes 0.5s. The global controller kicks in at every 10th period; this gives enough time to the per-core (subproblem) controllers to adapt to new delay budgets. We compare single-core Batching, naïve multicore (static per-task delay budgets), and full-fledged multicore Batching.

**Results.** Fig. 6 shows the key performance indicators (*i.e.*, rate and delay) in the system. Naïve and Batching multicore approaches start from the same initial state. Yet, the full-fledged multicore Batching is able to further improve the performance by adjusting the per-task delay budgets. Fig. 5 shows the underlying control loops: *i*) the global controller takes the surplus delay budget of the VLAN splitting task and gives extra

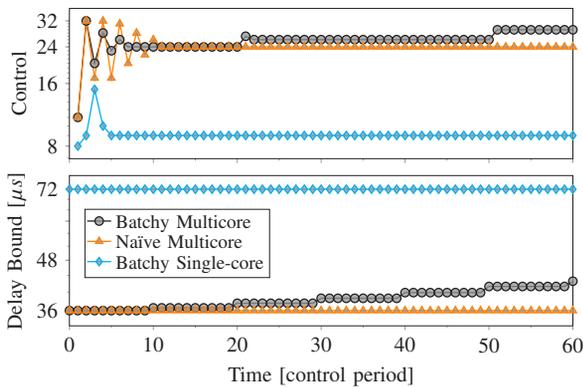


Figure 5. Control Parameters of the First Flow in VRF(2,4): control for ACL module and per-task delay budget on the second core (recall Fig. 4).

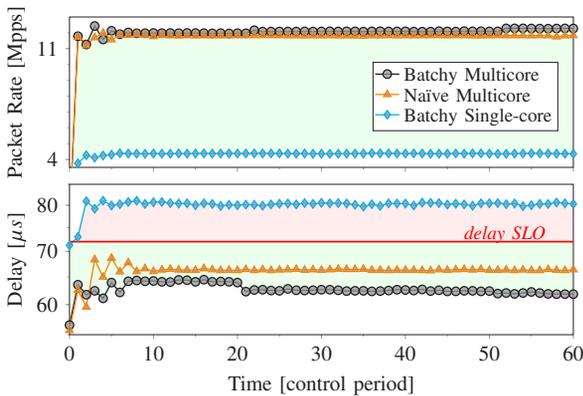


Figure 6. Key Performance Indicators of the VRF(2,4) pipeline: total packet rate and delay of the first flow. Delay SLOs are set to 72μs.

budget to the processing-heavy per-VLAN traffic processing; *ii*) this gives enough time to the per-VLAN task controller to queue up larger packet batches. This coordinated optimization improves the overall performance (Fig. 6). Over single-core Batchy, packet rate increases 2.5× and flow delay reduces to 0.75×. More importantly, the delay is finally below the SLO!

As of the controller performance, we measured the per-task and global controllers running time in each control period during the measurement and found that the multicore approaches result only a 7% increase on average due to extra global control steps.

To conclude, we see that our multicore extension is an enabler technology for Batchy, supporting use cases with ultra-low delay SLOs. We see the decomposition improves performance and its control overhead is negligible.

VI. RELATED WORK

A. Optimizing Resource Usage

Carefully execute an NF-chain on general-purpose hardware is one way to achieve performance improvement. Shenango [22] improves CPU utilization by bypassing the kernel and reschedule or scale up according to the occupancy of the packet ring buffers. This technique results low latency and improved CPU utilization. Similarly, IX [6] utilizes adaptive batch control to improve throughput and latency. Metron [20] improves

end-to-end performance in NF-chains by avoiding cross-CPU issues in NF-scheduling. These works focus on optimizing performance without controlling latency. In contrast, Batchy not just improves performance but carefully controls latency to meet SLOs.

B. Improving Performance by Offloading

Offloading some part of the processing to hardware components such as SmartNICs [23], FPGAs [24], or GPUs [8], [25], [26] is widely used to improve packet processing performance. To mitigate the packet offloading cost and to maximize GPU utilization, extensive batching [26] and careful load balancing between the offload hardware and the CPU [25] are required. Offloading works motivate the importance of batching, however, they are orthogonal to our work since they incorporate offloading to specific hardware elements.

C. Meeting Delay SLOs

Beside performance optimization, guaranteeing SLOs is another highly-desired behavior of NFV systems. Grus [8] an NFV framework with GPU offload introduces a multi-layer system with admission control and latency prediction model to guarantee delay SLOs. As opposed to our work, Grus guarantees delay SLO only for single VNF deployments, and the model is tailored for the GPU offloading scenario. SLOMO [27] predicts potential performance of VNF colocation, but does not provide SLO guarantees. In contrast to Grus, ResQ [28] provides performance isolation at CPU last-level cache solving the noisy neighbor problem of VNFs, and enables enforcing SLOs. NFV-RT [29] provides soft real-time guarantees for NF service chains deployed in data center environment using a fat-tree topology.

As opposed to our controller framework running on general hardware, these works are bound to a given NFV environment: they require a certain CPU feature, or specific underlying network topology. Our work focuses on a single host using general CPUs. Moreover, our controller framework extends previous work by providing a unique combination of dynamic internal batch de-fragmentation instead of applying batching only to packet I/O, analytic techniques for controlling queue backlogs, and selective SLO-enforcement at the granularity of individual flows in multicore systems.

VII. CONCLUSIONS

Batchy, a state-of-the-art batch-scheduling framework, presents massive performance improvements while conforming delay SLOs even at Mpps-scale traffic with SLOs at μs range. Batchy focuses on single-core execution.

In this paper we introduce a multicore extension to Batchy. To this end, we formulated a primal decomposition to find the optimal run-to-completion batch-scheduling on multicore systems. We developed and implemented effective control algorithms to be used in practical data flow graph batch-scheduling. Our evaluation on a real 5G use-case focusing on latency-optimized network function virtualization shows that the multicore Batchy provides better performance (2.5×

Batch-scheduling Data Flow Graphs with Service-level Objectives on Multicore Systems

packet rate) over the single-core algorithm and guarantees delay SLOs that are otherwise not feasible with the single-core algorithm.

Future work focuses on applying Batchy for ultra-low-latency and real-time applications in 5G and beyond networks.

ACKNOWLEDGMENT

This work was supported by the ÚNKP-21-4 New National Excellence Program of the Ministry of Innovation and Technology from the source of the National Research, Development and Innovation Fund, and NKFIH/OTKA Project #135606.

REFERENCES

[1] L. Linguaglossa, S. Lange, S. Pontarelli, G. Rétvári, D. Rossi, T. Zinner, R. Bifulco, M. Jarschel, and G. Bianchi, "Survey of performance acceleration techniques for Network Function Virtualization," *Proceedings of the IEEE*, vol. 107, no. 4, pp. 746–764, 2019.

[2] S. Han, K. Jang, A. Panda, S. Palkar, D. Han, and S. Ratnasamy, "SoftNIC: A software NIC to augment hardware," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2015-155, May 2015. [Online]. Available: <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-155.html>

[3] E. Warnicke, "Vector packet processing - one terabit router," July 2017.

[4] T. Barbette, C. Soldani, and L. Mathy, "Fast userspace packet processing," in *ACM/IEEE ANCS*, 2015, pp. 5–16.

[5] L. Molnár, G. Pongrácz, G. Enyedi, Z. L. Kis, L. Csikor, F. Juhász, A. Kőrösi, and G. Rétvári, "Dataplane specialization for high-performance OpenFlow software switching," in *ACM SIGCOMM*, 2016, pp. 539–552.

[6] A. Belay, G. Prekas, M. Primorac, A. Klimovic, S. Grossman, C. Kozyrakis, and E. Bugnion, "The IX operating system: Combining low latency, high throughput, and efficiency in a protected dataplane," *ACM Transactions on Computer Systems (TOCS)*, vol. 34, no. 4, p. 11, 2017.

[7] Intel, "Data plane development kit," <http://dpsdk.org>.

[8] Z. Zheng, J. Bi, H. Wang, C. Sun, H. Yu, H. Hu, K. Gao, and J. Wu, "Grus: Enabling latency SLOs for GPU-accelerated NFV systems," in *IEEE ICNP*, 2018, pp. 154–164.

[9] S. G. Kulkarni, W. Zhang, J. Hwang, S. Rajagopalan, K. K. Ramakrishnan, T. Wood, M. Arumathurai, and X. Fu, "NFVnice: Dynamic Backpressure and Scheduling for NFV Service Chains," in *ACM SIGCOMM*, 2017, pp. 71–84.

[10] T. Lévai, F. Németh, B. Raghavan, and G. Rétvári, "Batchy: Batch-scheduling data flow graphs with service-level objectives," in *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*. Santa Clara, CA: USENIX Association, Feb. 2020, pp. 633–649. [Online]. Available: <https://www.usenix.org/conference/nsdi20/presentation/levai>

[11] J. Nagle, "Congestion control in IP/TCP internetworks," Internet Requests for Comments, RFC Editor, RFC 896, January 1984.

[12] O. Michel, R. Bifulco, G. Rétvári, and S. Schmid, "The programmable data plane: Abstractions, architectures, algorithms, and applications," *ACM Comput. Surv.*, vol. 54, no. 4, May 2021. [Online]. Available: [doi: 10.1145/3447868](https://doi.org/10.1145/3447868)

[13] T. Lévai, G. Pongrácz, P. Megyesi, P. Vörös, S. Laki, F. Németh, and G. Rétvári, "The price for programmability in the software data plane: The vendor perspective," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 12, pp. 2621–2630, Dec. 2018.

[14] S. Boyd, L. Xiao, A. Mutapic, and J. Mattingley, "Notes on decomposition methods," *Notes for EE364B, Stanford University*, vol. 635, pp. 1–36, 2007.

[15] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1439–1451, 2006.

[16] D. Palomar, "Convex primal decomposition for multicarrier linear mimo transceivers," *IEEE Transactions on Signal Processing*, vol. 53, no. 12, pp. 4661–4674, 2005.

[17] H. Pannanen, A. Tolli, and M. Latva-Aho, "Decentralized coordinated downlink beamforming via primal decomposition," *IEEE Signal Processing Letters*, vol. 18, no. 11, pp. 647–650, 2011.

[18] "Batchy," <https://github.com/hslab/batchy/tree/multicore>.

[19] B. Stephens, A. Akella, and M. Swift, "Loom: Flexible and efficient NIC packet scheduling," in *USENIX NSDI*, Feb. 2019, pp. 33–46.

[20] G. P. Katsikas, T. Barbette, D. Kostić, R. Steinert, and G. Q. M. Jr., "Metron: NFV service chains at the true speed of the underlying hardware," in *USENIX NSDI*, 2018, pp. 171–186.

[21] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear programming: Theory and algorithms*. John Wiley & Sons, 2013.

[22] A. Ousterhout, J. Fried, J. Behrens, A. Belay, and H. Balakrishnan, "Shenango: Achieving high CPU efficiency for latency-sensitive datacenter workloads," in *USENIX NSDI*, 2019, pp. 361–378.

[23] Y. Le, H. Chang, S. Mukherjee, L. Wang, A. Akella, M. M. Swift, and T. V. Lakshman, "Uno: Unifying host and smart nic offload for flexible packet processing," in *Proceedings of the 2017 Symposium on Cloud Computing*, 2017, p. 506–519.

[24] B. Li, K. Tan, L. L. Luo, Y. Peng, R. Luo, N. Xu, Y. Xiong, P. Cheng, and E. Chen, "Clicknp: Highly flexible and high performance network processing with reconfigurable hardware," in *Proceedings of the 2016 ACM SIGCOMM Conference*, ser. SIGCOMM '16. ACM, 2016, p. 1–14.

[25] J. Kim, K. Jang, K. Lee, S. Ma, J. Shim, and S. Moon, "NBA (Network Balancing Act): A high-performance packet processing framework for heterogeneous processors," in *EuroSys*, 2015, pp. 22:1–22:14.

[26] S. Han, K. Jang, K. Park, and S. Moon, "Packetshader: A GPU-accelerated software router," in *ACM SIGCOMM*, 2010, pp. 195–206.

[27] A. Manousis, R. A. Sharma, V. Sekar, and J. Sherry, "Contention-aware performance prediction for virtualized network functions," in *SIGCOMM '20*, 2020, p. 270–282.

[28] A. Tootoonchian, A. Panda, C. Lan, M. Walls, K. Argyraki, S. Ratnasamy, and S. Shenker, "ResQ: Enabling SLOs in network function virtualization," in *USENIX NSDI*, 2018, pp. 283–297.

[29] Y. Li, L. T. Xuan Phan, and B. T. Loo, "Network functions virtualization with soft real-time guarantees," in *IEEE INFOCOM 2016*, 2016, pp. 1–9.



**Tamás Lévai** received the M.Sc. degree in Computer Engineering at the Budapest University of Technology and Economics (BME) in 2016. Currently, he is a Ph.D. candidate and assistant teacher at BME. His research interest focuses on computer networks and distributed computing, mainly software-defined networking, cloud native computing and high-performance packet processing.



**Gábor Rétvári** received the M.Sc. and Ph.D. degrees in electrical engineering from the Budapest University of Technology and Economics in 1999 and 2007. He is now a Senior Research Fellow at the Department of Telecommunications and Media Informatics. His research interests include all aspects of network routing and switching, the programmable data plane, and the networking applications of computational geometry and information theory. He maintains several open source scientific tools written in Perl, C, and Haskell.

# Knowgraph-TT: Knowledge-Graph-Based Transit Time Matching in Semiconductor Supply Chains

Nour Ramzy<sup>1</sup>, Hans Ehm<sup>1</sup>, Sandra Durst<sup>1</sup>, Konstanze Wibmer<sup>1</sup>, and Werner Bick<sup>2</sup>

**Abstract**—The semiconductor supply chain is characterized by a global and complex production network in a competitive market. The time when work at one location ends and can be resumed at another is defined as Transit Time (TT). Therefore, planning Transit Time accurately and minimizing delays is crucial as it is used in the execution system to determine the Available to Promise (ATP) and thus important for daily order confirmation. By determining the ATP, the customer receives a response to the resource availability and a due date to the customer requests. Due to tool inherent differences, we choose semantic integration via Knowledge Graph (KG) to match the planned TT used in the execution system and the actual TT measured in the monitoring tool. KnowGraph-TT thereby serves as a role model for further matching and alignment tasks using KG. It connects actual and planned TT, highlights the gaps via applied queries, and enables an optimized update of planned TT. With our solution, deviations of actual and planned TT can be minimized and confirmations of unrealizable deliverable times are avoided.

**Index Terms**—knowledge graph, semiconductor, order management, transit time mismatch

## I. INTRODUCTION

The semiconductor industry is competitive with a dynamic market characterized by time-intensive processes [9]. Especially in this highly competitive domain, semiconductor companies strive to offer the highest quality to their customers which implies sustaining delivery reliability. Reliable deliveries are important as customers depend on the delivery promises and their further production steps are based on this commitment. To achieve planning dependability, it is important to implement precise and reliable planning processes. The key to better planning and to foresee delays is to examine Transit Times (TTs). Transit time is the time taken to move goods physically between different locations in a supply chain or laterally to another facility [21].

Supply Chain (SC) integration, as well as the flow of information SC are essential for carrying out effective exchanges between parties [15], thus can enhance SC planning. Semantic data integration enables combining SC data from disparate sources and consolidating it into meaningful and valuable information.

In this paper, we present Knowledge-Graph-based TT matching (Knowgraph-TT), aligned with existing approaches solution that matches transit times of different data sources based on semantic data integration to minimize and prevent

delays. Knowgraph-TT leverages a well defined ontology to model TTs. Via KnowGraph-TT, delays are identified and data is kept up to date through semantic transit time matching to create more reliable planning processes within the SC.

The remainder of the paper is divided as follows: After an introduction, section II covers the relevant background knowledge and the need for TT matching. section III describes related approaches for data integration in SC and the gap that motivates the use of semantic data integration for TT matching. section IV contains the implementation details i.e., the ontology modelling, mapping to data sources. The semantic data integration process is shown with an example of two tools that store transit time based on different definitions. section V is the evaluation of the implementation. We rely on competency questions and SPARQL and we discuss the results. Finally, the work is rounded off with section VI where we conclude and discuss the next steps about further analysis in which external factors like a pandemic are addressed.

## II. BACKGROUND AND MOTIVATION

In this section, we present the necessary background knowledge e.g., order management, transit time and the need for TT matching.

### A. Transit Time

Transit time is the time taken to move goods physically between different locations in a supply chain or laterally to another facility [21]. We distinguish between the actual and the planned transit. The first is the time needed to deliver particular products to the customer. While the planned transit time on the other hand is the time that is expected and planned for future deliveries to the customer. The planned transit time is used to determine the Available To Promise (ATP) which is important for daily order confirmation.

Figure 1 shows that the actual and the planned transit time might be split into several small time intervals and might be in different tools (e.g. one for measuring and one for planning) and are measured and calculated differently. Despite one definition of transit time, the actual and the planned transit time might be in different tools as they reflect different parts of the supply chain. The focus of the planning tool is to plan transit times, while the focus of another tool, e.g., an internal logistics monitoring tool, is to track actual transit times.

<sup>1</sup> Infineon Technologies AG, Munich, Germany; E-mail: {nour.ramzy, hans.ehm, sandra.durst, konstanze.wibmer}@infineon.com.  
<sup>2</sup> Technical University of Applied Sciences Regensburg, Regensburg, Germany; E-mail: werner.bick@oth-regensburg.de.

Knowgraph-TT: Knowledge-Graph-Based Transit Time Matching in Semiconductor Supply Chains

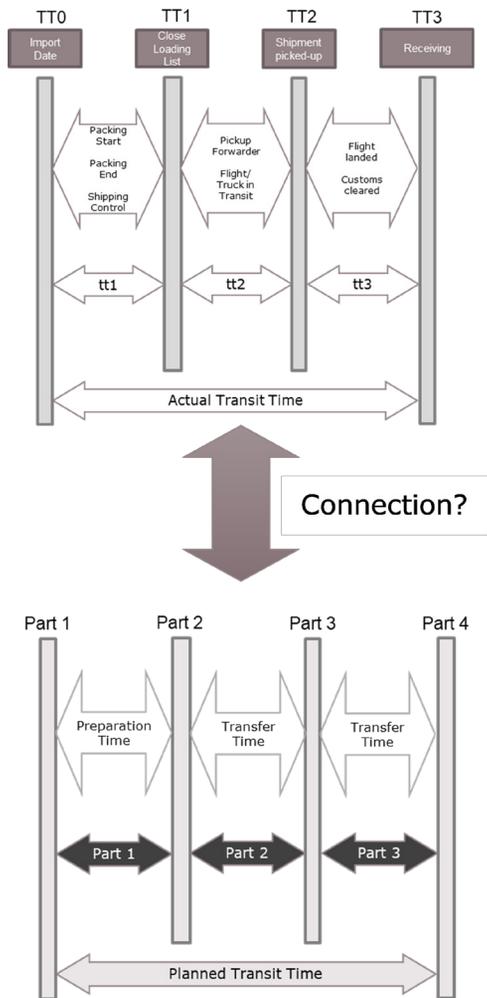


Fig. 1. Definition of Actual and Planned Transit Time.

Namely, within the tools, transit time is measured differently which leads to difficulties in matching the information and keeping it up to date. One example is the divergent interpretation of geographic data. In one tool, transit time is based on facility level and is measured as actual transit time and in the other tool, transit time is based on location level and measured as planned transit time. Transit times need to be accurate and therefore, a matching of the planned transit time and the actual transit time is needed. The matching results in an avoidance of order confirmations to unrealizable ATP's.

B. Transit Time Matching

When a customer requests products from the order management (OM), the divisional model (DM) checks whether and when the request can be fulfilled. All involved parties rely on the ATP and it is of special importance for the customer. [7]. Since the ATP is defined by the planned transit time at the time when the goods are sent out by backend (BE), it is important to detect deviations between actual and planned transit time early on and keep plan transit time up-to-date. After creating

an ATP, the DF (Demand Fulfillment) confirms on a daily base the Orders and Order Management (OM) sends an Order Promise (OP) to the customer. Thus the OP is based on the ATP which uses the planned transit time. With the OP, the delivery date becomes binding. By then, the customer trusts that the products will be delivered on time.

Figure 2 shows an example of the mismatch between the actual transit time and the planned transit time. The x-axis is the time course and on the y-axis are different blocks each describing a scenario. The first block is the *Plan (Weekly)* describing that 100 pieces are planned to be delivered to the distribution centers (DC) every Sunday, i.e., on weekly basis. The distribution center is a core part of a supply chain and connects factories and retailers [10]. An Available To Promise (ATP) of 100 pieces can be served from Sundays – it was committed from the factory, the factory gets measured on it and it has some buffer in it.

The second block is the plan at the *point of shipment* when the weekly plan is broken down on a daily basis. Within a planned transit time of three days, it is planned to transit the products from the backend on Monday to the distribution centers on Thursday. Backend is part of the chip production process and involves the steps of assembly, test and shipment to the DC. However, when it is shipped from backend the commitment with buffer is replaced by a calculation being the shipment date + planned transit time. So assuming that 50 are shipped on Monday with a planned transit time of 2 days, the ATP of 100 on Sunday is reduced by 50, which are expected on Wednesday (Monday shipping date + 2 days planned transit time). In the daily rerun of the order confirmation, these 50 are used and orders are brought forward.

*Actual* scenario in the third block shows, that the actual transit time takes longer than the planned one and the planned scenario is not fulfilled. Thus, the planned transit time does not correspond to the actual transit time. Now when the actual transit time is not 2 days but 4 days the brought forward orders need to be delayed as the goods are not coming on Wednesday but on Friday (= shipping date + 4 days actual transit time).

The fourth block is the *Communication to Customer*. The order management creates an availability commitment to the customers on Monday and communicates that the order will be shipped on Thursday. Thus, the customers can expect, that they will receive the products in three days. If the promises cannot be kept, it potentially leads to customer dissatisfaction. A deviation between the target and the actual transit time causes planning difficulties for all parties involved. For example, the production site or the customer has no planning certainty when promises are postponed. Therefore, it is important to correlate and match transit times.

III. SEMANTIC DATA INTEGRATION FOR TT

A. Related Work

1) *Approaches for TT planning*: There are various approaches to planning and optimizing transit times within a supply chain. One example to improving supply chain performance from the retail supply chain is the use of Radio

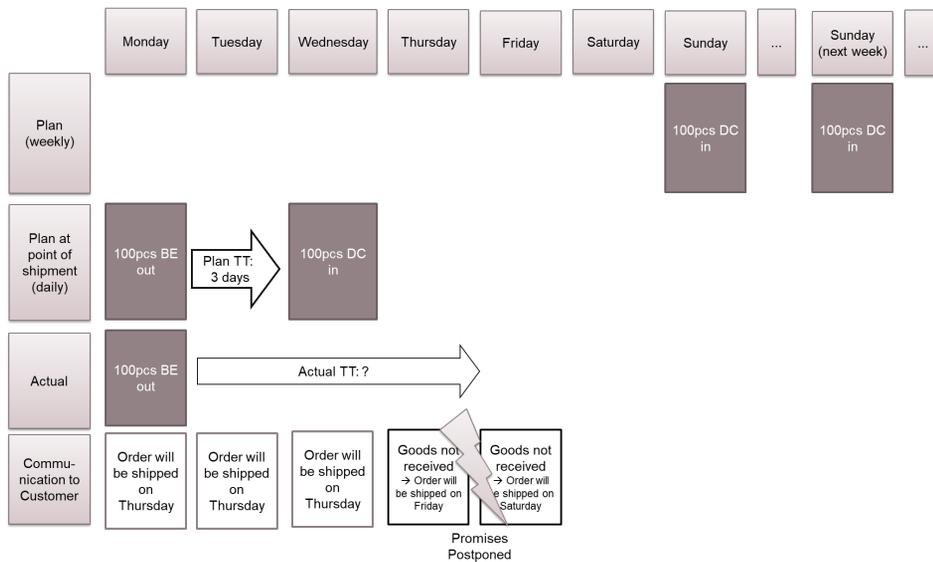


Fig. 2. Transit time mismatch.

Frequency Identification (RFID) technology. The technology is used to increase communication and information transparency between an object and other systems or software (e.g. ERP system) along the supply chain [17]. Methods like forecasting and simulation are also options for the planning and optimization process. The problem remains that the different information provided by such technologies cannot be compared directly. Data integration allows the alignment of data from different data sources.

2) *Traditional Data Integration:* One of the popular approaches to data integration is Extract Transform and Load (ETL). [20] provides a survey about traditional data integration and ETL techniques. Authors identify that ETL tools show limited ability is available to extract data from different sources at the same time. Moreover, without this domain knowledge, it is impossible to extract, transform and load. Semantic technologies are used to further enhance definitions of the ETL activities involved in the process [14]. Ontologies provide a common vocabulary for the integrated data and generate semantic data as part of the transformation phase of ETL. Semantic data integration is aligned with traditional data integration techniques. It can be referred to as Semantic ETL.

*B. Semantic Data Integration*

We rely on the ontology-based data access (OBDA) approach for semantic data integration. The schema is given in terms of an ontology representing the formal and conceptual view of the domain [8]. Also, we use ontology merging to overcome challenges such as the various definitions of location. A knowledge graph refers to a semantic network of concepts, properties, individuals and links representing and referencing foundational and domain knowledge relevant for a domain [5]. By creating the knowledge graph that is connecting different data sources, the data can be visualized

and analyzed without changing the original data sources. Therefore, redundancies can be avoided and the data is connected. Flexibility can be increased by having the possibility to implement changes on the data or adding new data sources. Semantic models are characterized with easy extensibility which makes them significant to the agile supply chain domain. In addition, this method of data integration achieves interoperability and information transparency. This type of data integration is particularly relevant in domains where data models are diverse and entity properties are heterogeneous [13].

IV. IMPLEMENTATION

In this section, we show KnowGraph-TT: applying semantic data to match the transit times of two tools of a multinational semiconductors company.

*A. Ontology Modeling*

The basis for the process KnowGraph-TT is to acquire different sources of transit time. Two different sources that manage transit time are found and analyzed. In the following, the sources are referred to as Tool A and Tool B. Tool A is equivalent to the executing tool while Tool B corresponds to the monitoring tool.

Firstly, Tool A stores the actual transit time on location level. The transit takes place from one location to another. Therefore, the location shipped from and the location shipped to are important concepts for the matching process. The class “TT\_Actual” represents the Actual Target Time to ship a product from the location “ShipFrom\_Loc” to the location “ShipTo\_Loc”.

Secondly, Tool B records the planned transit time on a facility level. Here, the transit takes place between two facilities. The planned transit time is the time it takes to

ship the products from the finishing facility to the expecting facility. It is similar to the structure of Tool A. The class “FTRN\_TRANSIT\_TIME” is the planned transit time it takes to ship products from “FAC\_FACILITY\_NAME\_FROM” to “FAC\_FACILITY\_NAME\_TO”. Each facility is identified by the class “FAC\_FACILITY\_NAME” which is important for the matching of locations in the location ontology.

As transit times are provided in various ways and they can not be matched directly, an additional intermediate ontology is necessary. Therefore, the location ontology, Figure 3, aligns the concepts of locations to be able to compare the different transit times with each other. Here, one location can have multiple facilities. Therefore, a location key is assigned to each facility so facilities and locations can be compared with each other.

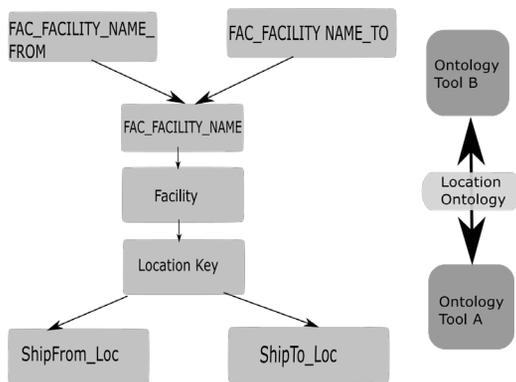


Fig. 3. Single Location Ontology.

B. Ontology Merging

Figure 4 shows the individual ontologies and the merged ontology. The output merged ontology is referred to as the semantic abstraction layer, it represents the schema of the domain. It contains the main concepts and relationships of the transit time domain. We rely on this to map data from different data sources and match TT.

The left side, is the ontology for the *Actual Transit Time in Tool A*. While on the right is the ontology for the *Planned Transit Time in Tool B*. Lastly, the *Location Ontology* is depicted in the middle part in Figure 4. Each facility from Tool A is uniquely identified by the class “LocationKey”. The matching between facility and location is explained in more detail in Figure 3. Figure 3 translates the facilities from Tool A to the locations in Tool B.

C. Mapping

After generating the semantic abstraction layer, the next step is to connect the data sources from the different tools to the merged ontology. For this step, the tool Protégé plugin “Cellfie” [19] is used. This is used because Protégé is an open source tool and therefore a common used tool. It creates instances out of the given data for every class contained in the given data sources. The semantic abstraction layer contains multiple triples. Each of those triples consists of a subject,

predicate and object. For the TT ontology one example triple looks as follows:

V. EVALUATION

In this section we evaluate the implementation and show the outcomes of the semantic data integration for TT matching.

A. Setup

The output of the implementation is a knowledge graph that contains the ontology (Figure 4) along with the materialized triples as shown in the mapping. We rely on SPARQL [18] queries to evaluate the knowledge-graph. We upload the knowledge-graph on Apache Jena Fueski Server and execute the queries.

The choice of this methodology for evaluation is driven by the fact that SPARQL is a standard that allows to express queries across diverse data sources. SPARQL can be used to query external data sources e.g. weather reports to add explainability and find a correlation between TT mismatch and bad weather conditions. Also, in case we wanted to attribute the mismatch to locations geospatial SPARQL allows to represent geospatial data is using GeoSPARQL, which is an RDF vocabulary and a set of extensions to SPARQL to support spatial queries. The results of the queries are discussed below. It is important to mention, that the transit times are given in hours. Also, the transit time data is partially extracted from the tools to illustrate the methodology.

B. Queries and Answers

For conciseness we show the Competency Questions (CQs), representing the SPARQL queries but in natural language. We refer to a GitHub repository for detailed queries. CQs are a set of requirements on the content as well as a way of scoping and delimiting the TT matching problem.

**CQ1: What are the planned transit times and average actual transit times for delivery routes between locations?**

Firstly, all delivery routes that address the same location are grouped. After summarizing the deliveries of the same routes, the corresponding actual transit time is averaged. For each route, the planned transit time is compared to the average actual transit time. Possibly, different planned transit times are assigned to an average actual time because several facilities can be related to one location. Results in Figure 7(a) provide an overview of the results from the query evaluation. For example, the average actual transit time of 295 hours differs from the planned transit time of 165 hours. It can be seen that within the same delivery route, there are large deviations in the average actual transit time and the planned transit time.

**CQ2: What are the actual transit times and its planned transit time for a certain delivery route between locations?**

This query filters one delivery route between two locations. Here, the planned TT is compared to the actual TT. Since different facilities are assigned to one location, various facility delivery routes exist within one location delivery route. This means, that several actual transit times are related to one

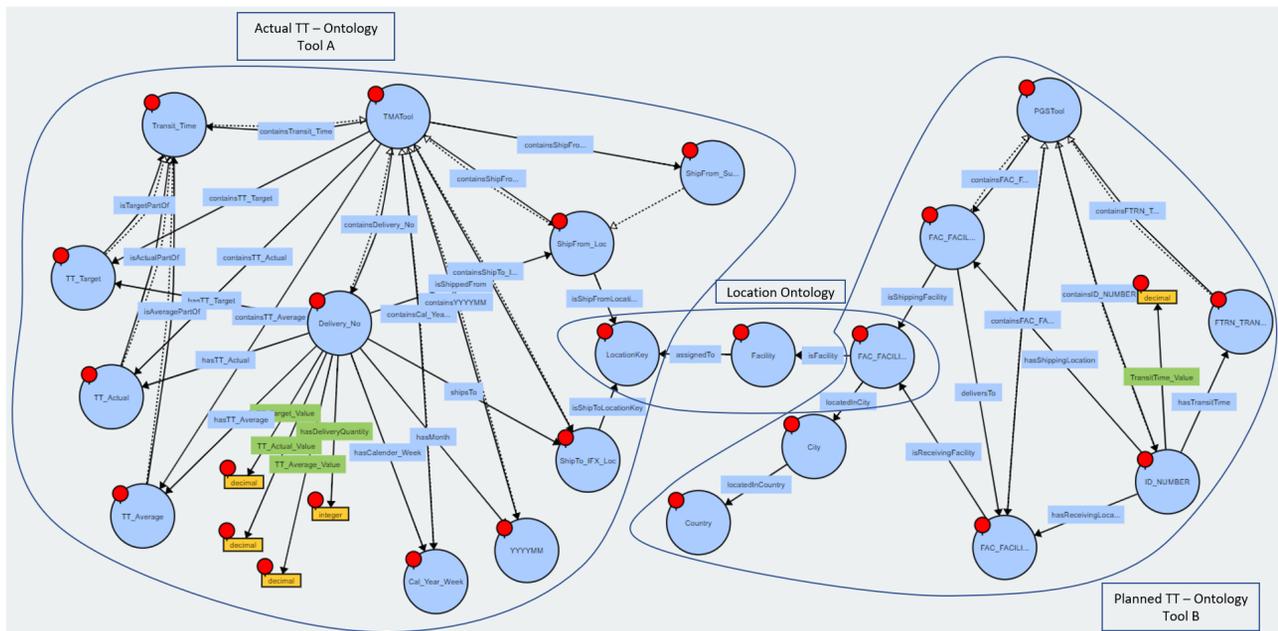


Fig. 4. Transit Time Ontology matching in the Measurement Tool for Actuals and in the Base Data Tool for Planning purposes.



Fig. 5. Triple City.



Fig. 6. Triple Facility.

planned transit time. Based on this delivery route, the actual TT is compared to the planned TT.

The query result is represented in Figure 7(b). The resulting planned transit times are 48 hours and 120 hours. Nevertheless, the actual transit times for the delivery route vary from 51 hours to 82 hours. The interval from 51 to 82 hours of the actual transit time shows, that there are large fluctuations between transit times.

**CQ3: What are the actual transit times and its planned transit time for a certain delivery route between facilities?**

This query only filters one shipping and one receiving facility. The results in Figure 7(c) show, that the planned transit time within this delivery route is 48 hours. Furthermore, the actual transit time may vary from 51 to 82 hours. Figure 7(c) shows, that the deviations between planned transit time and actual transit time are not as large as in CQ2.

**CQ4: What are the highest and lowest transit times of certain delivery routes?**

In this query, the plan transit time, average actual transit time, and both the minimum and maximum actual transit time are compared with each other. The results in Figure 7(d) show the big difference between the maximum and the minimum actual transit time. One example for a query result is a delivery route with a planned transit time of 72 hours, an average actual transit time of 62 hours, a maximum of 71 hours with a minimum transit time of 45 hours. One reason for the fluctuations is that several facilities are located in one location. Thus, according to one plan transit time, there are various actual transit times.

**CQ5: For which transit route is the actual transit time higher than the planned transit time?**

In this query, the plan transit time is compared to the actual transit time. Some deliveries are stated several times when different planned transit times are given. Finally, the deliveries are filtered which in reality took longer than planned. In Figure 7(e) the query results show, that in several cases the actual transit time is higher than the planned transit time of the delivery route.

**CQ6: What location transits have the highest discrepancy between planned and actual transit times? (Ordered by discrepancy)**

This query relies on the transits between locations. For each transit, the actual transit time, maximum planned transit time, and discrepancy obtained from the actual and planned transit times are given. The result table in Figure 7(f) is ordered descending by the discrepancy. The highest discrepancy for this data is 820 hours with an actual transit time of 928



Fig. 7. Chart visualization of the SPARQL query results for various Competency Questions (CQs).

discrepancy of 820 hours between the planned transit time and the actual transit time shows how unreliable the key figures can be sometimes.

C. Results & Discussion

The implemented approach combines the planned transit time used in the execution system and the actual transit time measured in the monitoring tool. The queries, which are evaluated in section V highlight the gaps between planned transit time from different viewpoints. The results show, that the gaps of TT matching between location routes are larger than the gaps between facilities. One possible explanation is, that different facilities can be assigned to one location. Therefore, different actual transit times are assigned to one planned transit time. Based on the query results, it could be shown that a timely update of planned transit times can be enabled as well as unachievable order confirmations are avoided.

In this paper, two tools were successfully connected using semantic data integration to combine the distinct interpretation of transit time and to enable the analysis of the consequences of the mismatch. Semantic Data Integration is applicable in other Supply Domains to integrate dispersed data sources. We can rely on the proposed location ontologies to extend to other domains e.g., Customer Relationship Management, Revenue Management as per [16]. Also, the semantic abstraction layer secures a common understanding of the domain in question, thus entails interoperability and extendibility. However, the approach has some limitations. First, KnowGraph-TT provides a self-created location ontology, thus missing the re-usability characteristic of ontologies. We did not re-use standard ontologies representing the location. Moreover, for the evaluation KnowGraph-TT is not evaluated versus other existing approaches or related work. We only used CQs for the evaluation to ensure that the output Knowledge Graph covers the domain in question, and enables the transit time mismatch analysis.

VI. CONCLUSION & OUTLOOK

In a competitive market, semiconductor manufacturers seek to offer the highest quality to their customers and rely on an accurate and reliable supply chain planning and commitment. In this work, we propose KnowGraph-TT to connect plan- and actual transit times on different definitions and tools via semantic data integration. Based on this, we apply this KnowGraph-TT to a use case of an international semiconductor manufacturing firm. The use case was evaluated successfully via competency questions in highlighting actual-versus plan transit time mismatches. Incorrectly planned transit times are the cause of ATP postponements and negative Early Warnings (a negative Early Warning is when a previously committed delivery date is postponed), but by far not every violation of a planned transit time in the internal supply chain causes an Early Warning. Thus, we can examine now the extent of the effect of time violations as a root cause of ATP postponements and negative Early Warnings. Consequently, we can study how to update the planned transit time concerning actual transit times to create a non-conservative and reliable demand fulfilment. In future work, we aim to analyze the effect of external factors on the supply chain. We intend to extend the knowledge graph capabilities to effects like the pandemic, which strongly influences the supply chain processes. This should enable us to correlate the transit time mismatch with COVID-19 reports, to be able to proactively change the plan transit times before violations happen.

REFERENCES

[1] H. Ehm, E. Schoitsch, J. W. van der Weit, N. Ramzy, L. Luo, and D. L. Gruetzner, "Digital Reference: a quasi-standard for digitalization in the domain of semiconductor supply chains," 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS). IEEE, Jun. 10, 2020. doi: 10.1109/icps48405.2020.9274750.

[2] Ramzy, N., Auer, S., Chamanara, J., Ehm, H. (2020). KnowGraph-MDM: A Methodology for Knowledge-Graph-based Master Data Management, unpublished manuscript.

[3] D. Kozma and P. Varga, "Supporting Digital Supply Chains by IoT Frameworks: Collaboration, Control, Combination," Infocommunications journal, no. 4. Infocommunications Journal, pp. 22–32, 2020. doi: 10.36244/icj.2020.4.4.

[4] M. Grobe, "RDF, Jena, SparQL and the 'Semantic Web,'" Proceedings of the ACM SIGUCCS fall conference on User services conference - SIGUCCS '09. ACM Press, 2009. doi: 10.1145/1629501.1629525.

[5] M. Galkin, S. Auer, M.-E. Vidal, and S. Scerri, "Enterprise Knowledge Graphs: A Semantic Approach for Knowledge Management in the Next Generation of Enterprise Information Systems," Proceedings of the 19th International Conference on Enterprise Information Systems. SCITEPRESS - Science and Technology Publications, 2017. doi: 10.5220/0006325200880098.

[6] Ramesh Jain, "Out-of-the-box data engineering events in heterogeneous data environments," Proceedings 19th International Conference on Data Engineering (Cat. No.03CH37405). IEEE. doi: 10.1109/icde.2003.1260778.

[7] B. A. Mousavi, R. Azzouz, C. Heavey, and H. Ehm, "A Survey of Model-Based System Engineering Methods to Analyse Complex Supply Chains: A Case Study in Semiconductor Supply Chain," IFAC-PapersOnLine, vol. 52, no. 13. Elsevier BV, pp. 1254–1259, 2019. doi: 10.1016/j.ifacol.2019.11.370.

[8] Giacomo, G. D., Lembo, D., Lenzerini, M., Poggi, A., & Rosati, R. (2018). Using ontologies for semantic data integration. In A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years (pp. 187-202). Springer, Cham.

[9] H. Ehm, "Managing complex global supply chains through score process and digitalizations," Infineon AG, Tech. Rep., 2020.

[10] Feng Li, Tie Liu, Hao Zhang, Rongzeng Cao, Wei Ding, and J. P. Fasano, "Distribution center location for green supply chain," 2008 IEEE International Conference on Service Operations and Logistics, and Informatics. IEEE, Oct. 2008. doi: 10.1109/soli.2008.4683040.

[11] Stonebraker, M., & Ilyas, I. F. (2018). Data Integration: The Current Status and the Way Forward. IEEE Data Eng. Bull., 41(2), 3-9.

[12] Rabe, M., Dross, F., Wuttke, A., Duarte, A., Juan, A., & Lourenço, H. R. (2017, July). Combining a discrete-event simulation model of a logistics network with deep reinforcement learning. In Proceedings of the MIC and MAEB 2017 Conferences (pp. 765-774).

[13] M. Cheatham and C. Pesquita, "Semantic Data Integration," Handbook of Big Data Technologies. Springer International Publishing, pp. 263–305, 2017. doi: 10.1007/978-3-319-49340-4\_8.

[14] J. Chakraborty, A. Padki, and S. K. Bansal, "Semantic ETL — State-of-the-Art and Open Research Challenges," 2017 IEEE 11th International Conference on Semantic Computing (ICSC). IEEE, 2017. doi: 10.1109/icsc.2017.94.

[15] G. Stefansson, "Business-to-business data sharing: A source for integration of supply chains," International Journal of Production Economics, vol. 75, no. 1–2. Elsevier BV, pp. 135–146, Jan. 2002. doi: 10.1016/s0925-5273(01)00187-6.

[16] N. Ramzy, S. Auer, J. Chamanara, and H. Ehm, "KnowGraph-PM: A Knowledge Graph Based Pricing Model for Semiconductor Supply Chains," Computer and Information Science 2021 — Summer. Springer International Publishing, pp. 61–75, 2021. doi: 10.1007/978-3-030-79474-3\_5.

[17] I. P. Vlachos, "A hierarchical model of the impact of RFID practices on retail supply chain performance," Expert Systems with Applications, vol. 41, no. 1. Elsevier BV, pp. 5–15, Jan. 2014. doi: 10.1016/j.eswa.2013.07.006.

[18] D. Calvanese, M. Giese, D. Hovland, and M. Rezk, "Ontology-Based Integration of Cross-Linked Datasets," The Semantic Web - ISWC 2015. Springer International Publishing, pp. 199–216, 2015. doi: 10.1007/978-3-319-25007-6\_12.

[19] Hardi, J. (2019), Cellfie Plugin, GitHub, GitHub Repository, <https://github.com/protegeproject/cellfie-plugin/wiki/User27s-Guide>.

[20] J. Chakraborty, A. Padki, and S. K. Bansal, "Semantic ETL — State-of-the-Art and Open Research Challenges," 2017 IEEE 11th International Conference on Semantic Computing (ICSC). IEEE, 2017. doi: 10.1109/icsc.2017.94.

[21] R. O'Byrne. (2015) Supply chain glossary of terms. [Online]. Available: <https://www.logisticsbureau.com/supply-chain-glossary/>.



**Nour Ramzy** is a PhD candidate at Infineon Technologies and Leibniz University of Hannover with the research focus on semantic data integration for supply chain management. She is a Master of Science in Information Technology from the University of Stuttgart. She wrote her Master Thesis in collaboration with Infineon Technologies in the supply chain innovation department.



**Prof. Dr. Werner Bick** is Professor of Supply Chain Management and Logistics at the Technical University of Applied Sciences Regensburg. He studied mechanical engineering at the Technical University of Munich and completed his doctorate there in robotics. In addition to classical logistics, his professional focus is on digitalisation and Industry 4.0.

---

## Knowgraph-TT: Knowledge-Graph-Based Transit Time Matching in Semiconductor Supply Chains



**Hans Ehm** is Lead Principal Supply Chain heading the Corporate Supply Chain Engineering Innovation department at Infineon Technologies. He has a Master of Science in Mechanical Engineering from Oregon State University. His interest is on Supply Chain Innovation with focus on capacity and demand planning, deep learning, artificial intelligence, simulation, and block chain applications for traceability.



**Sandra Durst** is an Intern at Supply Chain Innovation and works on topics related to Semantic Web. She currently is a Bachelor of Science student in Management and Technology with a major in Informatics at Technical University on Munich.



**Konstanze Wibmer** is a Master of Arts in Logistics from the Technical University of Applied Sciences Regensburg. She wrote her Master Thesis in collaboration with Infineon Technologies in the supply chain innovation department about transit time matching using Semantic Web technologies.



**18th International Conference on Network and Service Management**  
*Intelligent Management of Disruptive Network Technologies and Services*  
 31 October - 4 November 2022, Thessaloniki, Greece

**CALL FOR PAPERS**

The 18th International Conference on Network and Service Management (CNSM) is inviting authors to submit original contributions in network and service management research. CNSM 2022 is a selective single-track conference, covering all aspects of the management of networks and services, pervasive systems, enterprises, and cloud computing environments. The core track will be accompanied by a series of workshops and poster sessions.

Papers accepted and presented at CNSM 2022 will be published as open access on the conference website and will be submitted to IFIP Digital Library. Authors of selected papers, accepted for publication in the CNSM 2022 proceedings, will be invited to submit an extended version of their papers to the IEEE Transactions on Network and Service Management journal.

**Topics of Interest** (but not limited to)

**Network Management**

- IP Networks
- Wireless and Cellular Networks
- Optical Networks
- Virtual Networks
- Home Networks
- Access Networks
- Fog and Edge Networks
- Wide Area Networks
- Enterprise and Campus Networks
- Data Center Networks
- Industrial Networks
- Vehicular Networks
- Internet of Things and Sensor Networks
- Information-Centric Networks

**Service Management**

- Multimedia Services
- Content Delivery Services
- Cloud Computing Services
- Internet Connectivity and Internet Access Services
- Internet of Things Services
- Security Services
- Context-Aware Services
- Information Technology Services
- Microservices-based Applications
- Service Assurance

**Management Paradigms**

- Centralized Management
- Hierarchical Management
- Distributed Management
- Federated Management
- Autonomic and Cognitive Management
- Policy- and Intent-Based Management
- Model-Driven Management
- Pro-active Management
- Energy-aware Management
- QoE-Centric Management

**Business Management**

- Economic Aspects
- Multi-Stakeholder Aspects
- Service Level Agreements
- Lifecycle Aspects
- Process and Workflow Aspects
- Legal Perspective
- Regulatory Perspective
- Privacy Aspects
- Organizational Aspects

**Functional Areas**

- Fault Management
- Configuration Management
- Accounting Management
- Performance Management
- Security Management

**Technologies**

- Communication Protocols
- Middleware
- Overlay Networks
- Peer-to-Peer Networks
- Cloud Computing and Cloud Storage
- Data, Information, and Semantic Models
- Information Visualization
- Software-Defined Networking
- Network Function Virtualization
- Orchestration
- Operations and Business Support Systems
- Control and Data Plane Programmability
- Distributed Ledger Technology

**Methods**

- Mathematical Logic and Automated Reasoning
- Optimization Theories
- Control Theory
- Probability Theory, Stochastic Processes, and Queuing Theory
- Artificial Intelligence and Machine Learning
- Evolutionary Algorithms
- Economic Theory and Game Theory
- Monitoring and Measurements
- Data Mining and (Big) Data Analysis
- Computer Simulation Experiments
- Testbed Experimentation and Field Trials
- Software Engineering Methodologies

**Important Dates**

*Paper Registration:*  
10 June 2022  
*Paper Submission:*  
17 June 2022  
*Acceptance Notification:*  
18 July 2022  
*Camera Ready due:*  
2 September 2022

**Technical Program**

**Co-Chairs:**  
*Walter Cerroni,*  
University of Bologna, Italy  
*Salil Kanhere,*  
University of New South Wales, Australia  
*Lefteris Mamatas,*  
University of Macedonia, Greece

**General Co-Chairs:**  
*Marinos Charalambides,*  
Huawei Technologies, Belgium  
*Panagiotis Papadimitriou,*  
University of Macedonia, Greece



Authors are invited to submit original contributions that have not been published or submitted for publication elsewhere. Papers should be prepared using the IEEE 2-column conference style and are limited to 9 pages including references (full papers) or 5 pages including references (short papers). Papers should be submitted electronically in PDF format through EDAS at <https://edas.info/index.php?c=29457>.

For further information, please check <http://www.cnsm-conf.org/2022/>.

## Guidelines for our Authors

### Format of the manuscripts

Original manuscripts and final versions of papers should be submitted in IEEE format according to the formatting instructions available on

<https://journals.ieeeauthorcenter.ieee.org/>  
Then click: "IEEE Author Tools for Journals"  
- "Article Templates"  
- "Templates for Transactions".

### Length of the manuscripts

The length of papers in the aforementioned format should be 6-8 journal pages.

Wherever appropriate, include 1-2 figures or tables per journal page.

### Paper structure

Papers should follow the standard structure, consisting of *Introduction* (the part of paper numbered by "1"), and *Conclusion* (the last numbered part) and several *Sections* in between.

The Introduction should introduce the topic, tell why the subject of the paper is important, summarize the state of the art with references to existing works and underline the main innovative results of the paper. The Introduction should conclude with outlining the structure of the paper.

### Accompanying parts

Papers should be accompanied by an *Abstract* and a few *Index Terms (Keywords)*. For the final version of accepted papers, please send the short cvs and *photos* of the authors as well.

### Authors

In the title of the paper, authors are listed in the order given in the submitted manuscript. Their full affiliations and e-mail addresses will be given in a footnote on the first page as shown in the template. No degrees or other titles of the authors are given. Memberships of IEEE, HTE and other professional societies will be indicated so please supply this information. When submitting the manuscript, one of the authors should be indicated as corresponding author providing his/her postal address, fax number and telephone number for eventual correspondence and communication with the Editorial Board.

### References

References should be listed at the end of the paper in the IEEE format, see below:

- a) Last name of author or authors and first name or initials, or name of organization
- b) Title of article in quotation marks
- c) Title of periodical in full and set in italics
- d) Volume, number, and, if available, part
- e) First and last pages of article
- f) Date of issue
- g) Document Object Identifier (DOI)

[11] Boggs, S.A. and Fujimoto, N., "Techniques and instrumentation for measurement of transients in gas-insulated switchgear," *IEEE Transactions on Electrical Installation*, vol. ET-19, no. 2, pp.87–92, April 1984. DOI: 10.1109/TEI.1984.298778

Format of a book reference:

[26] Peck, R.B., Hanson, W.E., and Thornburn, T.H., *Foundation Engineering*, 2nd ed. New York: McGraw-Hill, 1972, pp.230–292.

All references should be referred by the corresponding numbers in the text.

### Figures

Figures should be black-and-white, clear, and drawn by the authors. Do not use figures or pictures downloaded from the Internet. Figures and pictures should be submitted also as separate files. Captions are obligatory. Within the text, references should be made by figure numbers, e.g. "see Fig. 2."

When using figures from other printed materials, exact references and note on copyright should be included. Obtaining the copyright is the responsibility of authors.

### Contact address

Authors are requested to submit their papers electronically via the following portal address:

[https://www.ojs.hte.hu/infocommunications\\_journal/about/submissions](https://www.ojs.hte.hu/infocommunications_journal/about/submissions)

If you have any question about the journal or the submission process, please do not hesitate to contact us via e-mail:

Editor-in-Chief: Pál Varga – [pvarga@tmit.bme.hu](mailto:pvarga@tmit.bme.hu)

Associate Editor-in-Chief:

Rolland Vida – [vida@tmit.bme.hu](mailto:vida@tmit.bme.hu)

László Bacsárdi – [bacsardi@hit.bme.hu](mailto:bacsardi@hit.bme.hu)



# SCIENTIFIC ASSOCIATION FOR INFOCOMMUNICATIONS



---

## Who we are

Founded in 1949, the Scientific Association for Infocommunications (formerly known as Scientific Society for Telecommunications) is a voluntary and autonomous professional society of engineers and economists, researchers and businessmen, managers and educational, regulatory and other professionals working in the fields of telecommunications, broadcasting, electronics, information and media technologies in Hungary.

Besides its 1000 individual members, the Scientific Association for Infocommunications (in Hungarian: HÍRKÖZLÉSI ÉS INFORMATIKAI TUDOMÁNYOS EGYESÜLET, HTE) has more than 60 corporate members as well. Among them there are large companies and small-and-medium enterprises with industrial, trade, service-providing, research and development activities, as well as educational institutions and research centers.

HTE is a Sister Society of the Institute of Electrical and Electronics Engineers, Inc. (IEEE) and the IEEE Communications Society.

## What we do

HTE has a broad range of activities that aim to promote the convergence of information and communication technologies and the deployment of synergic applications and services, to broaden the knowledge and skills of our members, to facilitate the exchange of ideas and experiences, as well as to integrate and

harmonize the professional opinions and standpoints derived from various group interests and market dynamics.

To achieve these goals, we...

- contribute to the analysis of technical, economic, and social questions related to our field of competence, and forward the synthesized opinion of our experts to scientific, legislative, industrial and educational organizations and institutions;
- follow the national and international trends and results related to our field of competence, foster the professional and business relations between foreign and Hungarian companies and institutes;
- organize an extensive range of lectures, seminars, debates, conferences, exhibitions, company presentations, and club events in order to transfer and deploy scientific, technical and economic knowledge and skills;
- promote professional secondary and higher education and take active part in the development of professional education, teaching and training;
- establish and maintain relations with other domestic and foreign fellow associations, IEEE sister societies;
- award prizes for outstanding scientific, educational, managerial, commercial and/or societal activities and achievements in the fields of infocommunication.

---

## Contact information

President: **FERENC VÁGUJHELYI** • [elnok@hte.hu](mailto:elnok@hte.hu)

Secretary-General: **ISTVÁN MARADI** • [istvan.maradi@gmail.com](mailto:istvan.maradi@gmail.com)

Operations Director: **PÉTER NAGY** • [nagy.peter@hte.hu](mailto:nagy.peter@hte.hu)

International Affairs: **ROLLAND VIDA, PhD** • [vida@tmit.bme.hu](mailto:vida@tmit.bme.hu)

Address: H-1051 Budapest, Bajcsy-Zsilinszky str. 12, HUNGARY, Room: 502

Phone: +36 1 353 1027

E-mail: [info@hte.hu](mailto:info@hte.hu), Web: [www.hte.hu](http://www.hte.hu)