On the Optimal Design of Fully Identifiable Next-Generation In-Vehicle Networks

Amani Ibraheem[a,*], Zhengguo Sheng[b] and George Parisis[b]

[a] *College of Computer Science, King Khalid University, Abha, Saudi Arabia*
[b] *School of Engineering and Informatics, University of Sussex, Brighton, United Kingdom*

ARTICLE INFO

Keywords:
In-vehicle network monitoring
Identifiable topology
Network tomography
Cyber-Physical System (CPS)

ABSTRACT

Due to the emerging advances in connected and autonomous vehicles, today's in-vehicle networks, unlike traditional networks, are not only internally connected but externally as well, exposing the vehicle to the outside world and making it more vulnerable to cyber-security threats. Monitoring the in-vehicle network, thus, becomes one of the essential and crucial tasks to be implemented in vehicles. However, the closed-in nature of the vehicle's components hinders the global monitoring of the in-vehicle network, leading to incomplete measurements, which may result in undetected failures. One solution to this is to use network tomography. Nevertheless, applying network tomography in in-vehicle networks is not a trivial task. Mainly because it requires that the in-vehicle network topology should be *identifiable*. To this end, we propose in this work an identifiable in-vehicle network topology that enables overall monitoring of the network using network tomography. The new topology is proposed based on extensive analysis to ensure full identifiability under the constraint that only edge nodes can monitor the network, which is the case for in-vehicle networks where internal nodes are not directly accessible. We propose two main algorithms to transform existing in-vehicle network topologies. The first algorithm applies to an existing topology which can be transformed into full identifiability by adding extra nodes/links. Evaluation results show the effectiveness of the proposed transformation algorithms with a maximum added weight of only 3% of the original weight. Furthermore, a new optimization algorithm is also proposed to minimize the topology weight whilst maintaining the full identifiability by redesigning a new topology. With this algorithm, the results show that the total weight can be reduced by 6%. In addition, compared with the existing approaches, monitoring the in-vehicle networks with the proposed approach can achieve better monitoring overhead and a 100% identifiability ratio.

1. Introduction

Vehicles nowadays are the main constituent of Connected and Autonomous Vehicle (CAV) systems, and they are considered critical Cyber-Physical Systems (CPSs) that need to be monitored to detect issues related to both performance failures and cyber-security threats. One essential component to monitor is the in-vehicle network. However, monitoring the internal part of the network is not always possible. This is because the internal elements of in-vehicle networks are difficult to access due to proprietary closed-in devices provided by Original Equipment Manufacturers (OEMs). In addition, monitoring every single part of the network can overburden it and may perturb the existing traffic where such disturbance can result in serious consequences especially for safety- and latency-critical applications. For these reasons, alternative monitoring solutions that do not require contribution from internal elements should be investigated. One such solution is *network tomography* [1, 2]. Network tomography is a monitoring mechanism that can be used to infer the unmeasured network performance by only monitoring a subset of the network.

The motivations for using network tomography to monitor the in-vehicle network are multifold. First, network tomography can provide an efficient and lightweight solution to infer the internal network performance without requiring access or contribution from internal elements. This is suitable for in-vehicle networks where direct access to internal nodes (e.g., CAN bus and Ethernet switches) is difficult, which makes it hard for such nodes to be used in the monitoring process as it is not possible to modify them. In addition, such internal nodes are incapable of complex monitoring tasks due to their limited memory and computational resources. Second, unlike the existing machine learning-based solutions, network tomography does not require any training. Thus, the collection and preprocessing of large datasets are avoided altogether when relying on network tomography.

Typically, to use network tomography, the topology has to be *identifiable* [3, 4]. In general, the topology is said to be identifiable if *all* link-level metrics can be *uniquely* identified using the available measurements. This is an important feature in network tomography, because if the topology is unidentifiable, then the performance of the *overall* network cannot be inferred. It has been shown in [5] that not all in-vehicle network architectures are identifiable. Therefore, in this work, we formalise the topological requirements for in-vehicle networks so that the design of new networks, or the modification of existing ones, can satisfy the identifiability conditions, hence, allowing for overall network monitoring using network tomography.

Based on our theoretical analysis, two main algorithms are proposed to transform any existing in-vehicle network topologies into identifiable and optimal ones. In particular,

[*] Corresponding author
A.Ibraheem@sussex.ac.uk; Amalii@kku.edu.sa (A. Ibraheem)
ORCID(s): 0000-0001-8621-8745 (A. Ibraheem)

the first algorithm is used to transform an unidentifiable topology into an identifiable one. This transformation often requires adding more nodes and links to the original topology. For this reason, the second algorithm is proposed to transform the resulting topology into an optimal one with a minimum number of links. Such algorithms can further assist in the modification of existing in-vehicle networks' topologies, so that they satisfy the identifiability conditions, rather than building the network from scratch, thus, saving time and resources that otherwise will be costly [6]. In addition, the new Electrical and Electronic (E/E) architectures [7, 8] can greatly benefit from the optimisation algorithm that can be used to minimise the number of links and nodes in the vehicle network. This is the goal of the new E/E architectures such as the domain- and zonal-based architectures where the main objective of these architectures is to utilise Software-Defined Networking (SDN) functionality [9] to replace hardware-based functions with software ones and hence replace a large number of ECUs with fewer number of more powerful devices, e.g., *High-Performance Computing (HPC) platforms* [6, 10]. In domain-based architectures, the nodes (i.e., ECUs) are grouped based on their functionalities so that two ECUs that are responsible for similar functions can be consolidated into one more powerful node. Zonal-based architecture on the other hand groups ECUs based on their physical locations within the vehicle. Thus, further reducing the number of nodes. Although the focus of the current paper is on in-vehicle network topologies, the approach can be applied to other networks that share the same constraint (only edge nodes are accessible).

Overall, this paper proposes two main algorithms. First, is to transform a given in-vehicle network topology that is *unidentifiable* into *identifiable* one. The reason for this is to allow for the overall monitoring of the in-vehicle network including the internal part of it, from the edge nodes, without the need to access the internal elements. This is what network tomography can be used to achieve. With network tomography, the monitoring overhead can significantly be reduced as we only need to monitor a subset of the network instead of every part of it. The second algorithm is to transform any identifiable topology into an optimised one. By optimal topology, we mean a topology that has a minimum number of links while still satisfying the identifiability condition. This is important to ensure that the vehicle weight is kept to a minimum as the identifiability transformation usually requires adding more nodes/links. Therefore with these algorithms, we can achieve both identifiable as well as optimal in-vehicle network topologies.

Fundamentally, the proposed approach aims to offer in-depth system insights into the in-vehicle network. If the topology is fully identifiable, the monitoring of the overall in-vehicle network can be performed by only a subset of the network and with limited resources. Having such detailed measurements for all components in the in-vehicle network contributes to many benefits that can be seen by applying network tomography in different application areas. These areas include: detecting and locating anomalies in the network, analysing fine-grained network performance, for instance, with fully identifiable topology, different network metrics can be measured for each and every link in the network, load-balancing can be easily implemented since all link-level measurements are available, and in addition, with network tomography network management and diagnostics can be easily implemented, this especially crucial for critical systems such as the in-vehicle network. It is worth mentioning that if one or more of the network elements cannot be measured, because the network topology is unidentifiable, some or all of the above applications cannot be efficiently employed due to the lack of complete measurements. Therefore, it is of paramount importance that the monitoring approach can provide detailed measurements regarding all network components, including the internal ones. If measurements were not available for the internal networking elements, then any faulty elements could go undetected, which may yield a negative impact on the network performance, possibly endangering human lives. Also reacting to any network incident cannot be possible if the compromised component cannot be located. Locating compromised components usually requires measuring such components, including the internal components.

It is worth noting that the most common communication protocol used in in-vehicle networks is the Controller Area Network (CAN). CAN uses a serial communication bus. However, other protocols are being used nowadays such as automotive Ethernet. Authors in [5] studied network identifiability for such protocols in more detail. The current work focuses on achieving fully identifiable in-vehicle networks regardless of which communication protocols are being used.

The main contributions of this paper can be summarized as follows:

- We study the topological structure of in-vehicle networks and derive the essential necessary and sufficient mathematical conditions for the network topology to be identifiable.

- Based on the result of our theoretical analysis, we propose algorithms to check for identifiability and transform an existing unidentifiable in-vehicle network topology into an identifiable one. Through extensive simulations on random topologies as well as on different in-vehicle network topologies, the evaluation results show an average of maximum added weight of only 3% for real in-vehicle network topologies.

- To further improve the identifiable topology, we propose an optimisation algorithm that ensures the topology is identifiable with the minimum number of links. The results show that the optimisation algorithm reduces the original topology weight by up to 2% on average when tested on random topologies and up to 6% when tested on real in-vehicle network topologies.

- In addition, we compare the proposed monitoring approach using network tomography with partial network tomography and with two of the state-of-the-art monitoring solutions. The results show that the proposed approach achieves better monitoring overhead with full network identifiability.

The rest of this paper is organized as follows: the next section discusses work related to in-vehicle network architectures and existing monitoring solutions, Section 3 describes the system model and states the problem this work is tackling, Section 4 and Section 5 extensively study and derive the topological conditions needed to acquire identifiable and optimal topology, respectively, in addition to the proposed transformation algorithms presented in each section, Section 7 shows the performance evaluation and results, while Section 8 concludes the paper and discusses some future directions.

2. Related Work

Generally, current in-vehicle networks do not support security measures [11]. However, there have been many proposed approaches by the research community. One of the state-of-the-art monitoring solutions for in-vehicle networks was proposed by Lee in [12] and is called Offset Ratio and Time Interval based Intrusion Detection System (OTIDS). The idea of OTIDS is that it plugs a monitoring ECU for the sole purpose of monitoring the CAN bus using remote frames. It periodically requests messages provided by all CAN nodes and measures their offset ratio and time intervals. They evaluated their approach to detecting three types of attacks: Denial of Service (DoS), fuzzy, and impersonation attacks. For each attack type, they measured a different metric. For example, to detect DoS attacks, they measured the ratio of instant replies of remote frames, to detect fuzzy attacks, they used a correlation coefficient between offsets and time intervals, while average response time was used to detect impersonation attacks. A limitation of this approach is that if the number of unique CAN IDs is high, then it incurs extra burden on the network and consumes a large amount of bus bandwidth by frequently communicating request/reply frames for all CAN IDs.

Another approach is Clock Offset Based Intrusion Detection System (COIDS) [13]. As the name suggests, this approach is based on monitoring the clock offset of each ECU. It follows three main steps: first, a baseline of each ECU's normal clock profile is constructed using active learning such as in [14], second, to detect anomalies, the cumulative sum of deviations from the normal behaviour is derived using cumulative sum method [15], last, the exact time of the attack is specified using sequential change-point detection. As with OTIDS, they evaluated their proposed approach to detect DoS, fuzzy, and impersonation attacks. COIDS also requires plugging a monitoring ECU into the CAN bus to monitor the network.

Additionally, the recent approaches benefit from the advances in computational power and thus they often rely on Machine and Deep Learning (ML and DL) solutions. For example, in [16], a Deep-SVDD (Support Vector Data Description) [17] has been proposed to obtain voltage fingerprints for each CAN ID which is then used to detect malicious frames and determine their source. In addition, Deep Convolutional Neural Networks (CNNs) have been used to monitor in-vehicle networks such as CAN [18] and Ethernet [19]. The approach in [18], however, could not detect new types of attacks. To solve this issue, another model was proposed in [20] which, as claimed, could detect attacks that the model was not trained on. On the other hand, authors in [19] focused on detecting injection attacks for Audio-Video Transport Protocol (AVTP) streams. Moreover, transfer learning was used in [21] where a convolutional Long Short Term Memory (LSTM) network was employed.

These solutions focus on one type of communication protocol i.e., either CAN or Ethernet. Hence, they monitor a single subsystem and ignore other subsystems which form the overall in-vehicle network. Therefore, these solutions cannot be used to monitor next-generation in-vehicle networks. Instead, there should be new monitoring solutions that can monitor the overall network performance. Our goal in this work is to propose one of these solutions based on network tomography. Another limitation with the existing monitoring solutions is that relying on ML- or DL-based solutions is not effective due to several reasons [22]. One is that they require massive datasets to process and train which is both time- and resource-consuming. Another reason is that false positives and false negatives cannot be avoided when employing ML or DL solutions, and for mission-critical applications, such as the ones in vehicles, having a false positive or false negative alarm is intolerable as this may lead to serious consequences.

Network tomography is one of the network monitoring approaches that is based on mathematical modelling of the network and its performance metrics. It was first studied by Vardi [23] to estimate the origin-destination traffic matrix. Depending on the problem at hand, network tomography can be divided into three categories: (i) link-level parameter estimation [3], (ii) origin-destination traffic matrix estimation [23] and (iii) topology inference [24]. This paper focuses on the first category where the end-to-end (*path-level*) measurements are used to infer the metric of *link-level* performance. Further, the measurements used in network tomography can either be active or passive. Active measurements use specialised probes sent between monitoring nodes to monitor the network. This type of measurement requires certain conditions for the minimum number of monitors and their placement to identify all link-level metrics. Passive measurements, on the other hand, exploit the existing traffic (e.g., by sampling from it) to measure the network performance [25]. In passive measurements, the existing traffic is not guaranteed to form a full-rank matrix that is needed to uniquely identify all link-level metrics. Hence, in this work, we focus on using active measurements.

An important aspect of network tomography is *network identifiability*. Network tomography can, uniquely, infer the

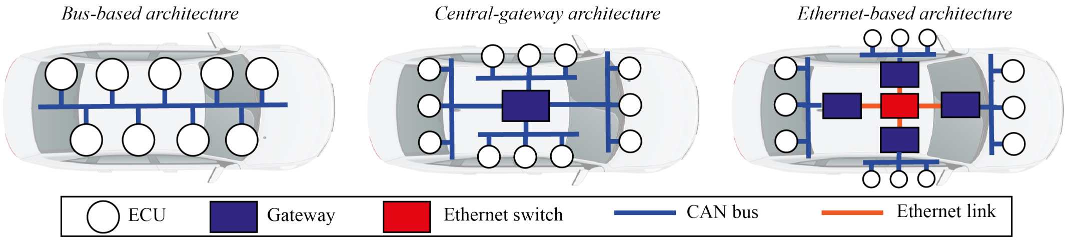


Figure 1: Three main in-vehicle network topologies studied in [5].

performance for all link-level metrics of an in-vehicle network if the topology is *identifiable* under the given monitor placement. If the network is not identifiable, authors in [26] proposed to use partial network tomography with a deep neural network. Further, in [27], they evaluated this approach with full algebraic tomography (where the topology is fully identifiable) and found that the algebraic tomography approaches yield better results in detecting and locating the anomalous link. In addition, the same authors in [5] studied network identifiability for three main automotive network architectures shown in Figure 1. The first architecture is the bus-based architecture which uses fieldbus communications, the second architecture is the central-gateway architecture that has a gateway connecting different system domains/communication protocols, while the third architecture is the Ethernet-based architecture that may include domain-based architecture, zonal-based architecture, or a combination of both. These architectures are based on the E/E architectures where the most recent one is the Ethernet-based architecture. They found that the topology for a single CAN network is always identifiable using only two monitors. They also found that the central gateway architecture is identifiable as long as there are at least three CANs connected to the gateway. On the other hand, Ethernet-based architecture can be *unidentifiable*. This means that network tomography cannot uniquely infer the individual link-level metrics. To this end, in this work, we formalise the requirements needed to achieve a fully identifiable topology. In addition, based on the theoretical analysis, we devise a transformation algorithm to transform any unidentifiable topology into an identifiable one. Additionally, to further minimise the vehicle weight, an optimisation algorithm is proposed. This ensures that the topology is identifiable and at the same time optimal with a minimum number of links. These algorithms can also be used to enhance the new E/E architectures such as domain-based and zonal-based architectures by making them identifiable as well as optimal.

3. System Model and Problem Statement

3.1. System Model

Table 1 shows a summary of notations used throughout this paper and their descriptions. We assume that in-vehicle network topology is known, and we follow graph theory conventions, defined in [28], to represent the network and its characteristics.

The in-vehicle network is modelled as an undirected graph[1] $G = (V(G), E(G))$ where $V(G)$ is a set of vertices (or *nodes*) and $E(G)$ is a set of edges (or *links*). Each link $e_i \in E(G)$, with $e_i = uv$, $i \in \{1, 2, \dots, \gamma\}$, connects two *adjacent* nodes $u, v \in V(G)$. We represent the end-points of link e_i as $v_h(e_i)$ (*head*) and $v_t(e_i)$ (*tail*). Based on node degree $d(u)$, which is defined as the number of links node u is incident to, we define the following two sets.

Definition 1. *Given an in-vehicle network G, sets of edge nodes $\mathcal{E} \in V(G)$ and internal nodes $\mathcal{R} \in V(G)$ are defined as*[2]

- $\mathcal{E} = \{u \in V(G) : d(u) = 1\}$*, and*
- $\mathcal{R} = \{u \in V(G) : d(u) \geq 2\}$

where $\mathcal{E} \cup \mathcal{R} = V(G)$ and $\mathcal{E} \cap \mathcal{R} = \emptyset$.

Let $p(u, v) = \{e : e \in E(G)\}$ be a path between any node pair $u, v \in \mathcal{E}$, and it consists of a set of links in which such path traverses. Let $\mathcal{P}$ be the set of all possible paths and $\mathcal{P}_m \subseteq \mathcal{P}$ be the set of measured paths. Note that elements in $\mathcal{P}$ are *simple* paths (do not include repeating nodes).

As the internal nodes are not directly accessible, we assume that only nodes in $\mathcal{E}$ are accessible, hence they can be used as monitors. The network tomography problem is expressed by the following linear system

$$\boldsymbol{y} = A \otimes \boldsymbol{x} \tag{1}$$

where $\boldsymbol{y} = [y_1, y_2, \dots, y_\kappa]^T$ is a vector in $\mathbb{R}^\kappa$ of path-level measurements, A is a $\kappa \times \gamma$ (refer to Table 1) measurement matrix and $\boldsymbol{x} = [x_1, x_2, \dots, x_\gamma]^T$ is a vector in $\mathbb{R}^\gamma$ of link-level metrics. Although there can be multiple paths between any two edge nodes in $\mathcal{E}$, routing during normal operation of the in-vehicle network is deterministic where there is only one single path in use between any two edge nodes $u, v \in \mathcal{E}$. Thus, the measurement matrix A is a binary matrix with entries $a_{ji} \in \{0, 1\}$. If path p_j traverses link e_i, we say $a_{ji} = 1$, otherwise $a_{ji} = 0$. The operation $\otimes$ depends on the problem type. If the problem is additive (e.g., delay or packet success/loss rate tomography) then $\otimes$ is for matrix multiplication. For boolean problems, $\otimes$ is boolean matrix multiplication, i.e., $y_j = \vee_j (a_{ji} \wedge x_i)$.

[1] The terms *graph*, *network* and *topology* are used interchangeably in this work.

[2] Sometimes we drop the graph name G and simply say $V, E, \mathcal{E}, \mathcal{R}$, etc., for which we mean $V(G), E(G), \mathcal{E}(G), \mathcal{R}(G)$. Same for γ_G and η_G, which we sometimes say γ and η.

Table 1
Notations and their descriptions.

Notation	Description
$G + \{uv\}$ $(G - \{uv\})$	network G when link between two vertices $u, v \in V(G)$ is added (deleted)
$\lvert\mathcal{X}\rvert$	cardinality of set $\mathcal{X}$
$\mathcal{E}, \mathcal{R} \subset V(G)$	set of edge and internal nodes in G (see Definition 1)
$\mathcal{I}, \mathcal{T} \subset E(G)$	set of internal and external links in G
$\mathcal{N}(u)$	set of neighbours for node $u \in V(G)$
$\mathcal{N}_{\mathcal{R}}(u) \subseteq \mathcal{N}(u)$	set of internal nodes that are neighbours to u
$C(G)$	set of components in graph G
$\mathcal{R}_{3^+}, \mathcal{R}_{3^-} \subseteq \mathcal{R}$	set of internal nodes having node degree larger and less than 3, respectively
$\mathcal{P}$	set of all possible paths between edge nodes in G
$\mathcal{P}_m \subseteq \mathcal{P}$	set of measured paths
$u(e_i)$	node $u \in V(G)$ incident to link $e_i \in E(G)$
$\gamma := \lvert E(G)\rvert$	total number of links in network G
$\eta := \lvert V(G)\rvert$	total number of nodes in network G
$\kappa := \lvert\mathcal{P}_m\rvert$	number of measured paths
l	number of uniquely identifiable links
$p(u, v)$	path between $u \in \mathcal{E}$ and $v \in \mathcal{E}$
$d(u)$	degree of node $u \in V(G)$
$\sigma := \lvert\mathcal{R}_{3^-}\rvert$	total number of internal nodes having node degree less than 3
$\lambda := \lvert\mathcal{R}\rvert$	total number of internal nodes in G
φ_u	$\sum_{i=1}^{\psi}(d(u) - 3) : u \in \mathcal{R}_{3^+}$
$d(u, v)$	distance between two nodes $u, v \in V(G)$
$\psi := \lvert\mathcal{R}_{3^+}\rvert$	total number of internal nodes having node degree larger than 3
$\beta := \lvert\mathcal{W}\rvert$	number of internal nodes in $\mathcal{R}_{3^+}$ that are neighbours to more than one node in $\mathcal{R}_{3^+}$
$\zeta_u := \lvert\mathcal{N}_{\mathcal{R}}(u)\rvert$	total number of internal nodes that are neighbours to $u\mathcal{R}_{3^+}$
$S_d(G) \in \{\text{true}, \text{false}\}$	status of network G as either unidentifiable, *false*, or identifiable, *true*
$S_o(G) \in \{\text{true}, \text{false}\}$	status of network G as either optimal, true, or not, false

3.2. Problem Statement and Assumptions

Given an in-vehicle network G, we aim to decide whether it is identifiable or not (i.e., $S_d(G) \in \{true, false\}$). If $S_d(G) = false$, the goal is to transform G into an identifiable topology G_i (i.e., $G \longrightarrow G_i$) where $S_d(G_i) = true$. The ultimate goal is to achieve a minimum number of links γ_{G_i} in the transformed topology G_i. Therefore, the resulting topology is checked for optimality. If $S_o(G_i) = false$, then the aim is to transform the topology into an optimal one with a minimum number of links while keeping the topology identifiable.

In this work, we adopt the following assumptions:

1. Only edge nodes in $\mathcal{E}$ can be used as monitors since internal nodes are inaccessible.
2. The focus in this work is on networks forming only acyclic graphs. Cyclic graphs are not allowed.
3. Links in G are symmetric.
4. The in-vehicle network is connected. The same principle, however, can be applied on individual disconnected components.

4. Topology Identifiability

The following defines network identifiability.

Definition 2. *Network G is identifiable if all links in $E(G)$ are identifiable. A link $e_i \in E(G)$ is identifiable, if its associated metric x_{e_i} can be uniquely determined from the path-level measurements in $\mathbf{y}$ by solving* (1).

The identifiability of any network topology can be one of the following:

1. **Full identifiable topology:** if metrics of each link $e_i \in E(G)$ is uniquely determined by solving (1). It can also be called γ-identifiable network.
2. **l-identifiable topology:** if the maximum number of links that can be identified is l where $l < \gamma$.
3. **Unidentifiable topology:** if no link metrics for any link in $E(G)$ can be uniquely determined by solving (1). In this case $l = 0$.

4.1. Topological Conditions

In this work, since the goal is to measure all networking elements including the internal network, we focus on achieving a topology that is fully identifiable.[3] In the following, we study the topological conditions needed to transform any unidentifiable topology into an identifiable one, under the constraint that only nodes in $\mathcal{E}$ can monitor the network.

Links in any in-vehicle network can be classified into two categories: *internal* and *external*.

Definition 3. *Given an in-vehicle network $G = (V, E)$, sets of internal and external links ($\mathcal{I} \in E(G)$ and $\mathcal{T} \in E(G)$) are defined as:*

- $\mathcal{I} = \{e_i \in E(G) : v_h(e_i) \in \mathcal{R}, v_t(e_i) \in \mathcal{R}\}$
- $\mathcal{T} = \{e_i \in E(G) : v_h(e_i) \in \mathcal{E}, v_t(e_i) \in \mathcal{R}\}$

where $\mathcal{I} \cup \mathcal{T} = E(G)$ and $\mathcal{I} \cap \mathcal{T} = \emptyset$. $\mathcal{I}(u)$ is a set of internal links node u is incident to.

The following lemma states the condition required to have an acyclic-connected graph G, which is needed for the subsequent theorem.

Lemma 1. *Any connected graph G is acyclic if and only if it has $\eta - 1$ links, where $\eta := |V(G)|$.*

[3] In the remaining of this paper, we simply use the term *identifiable topology* to refer to a full identifiable topology. For unidentifiable or l-identifiable topology we simply say *unidentifiable topology*.

Proof. See proof of Corollary 1.5.3 in [28]. □

From the assumption that an in-vehicle network is a connected graph, we derive the following theorem that is necessary for designing the transformation algorithm.

Theorem 1. *Any acyclic-connected graph G with $\lambda \geq 2$ implies that $\mathcal{I}(u) \neq \emptyset, \forall u \in \mathcal{R}$, where $\lambda := |\mathcal{R}|$.*

Proof. For a connected graph G, let assume that $\mathcal{I}(u) = \emptyset, \exists u \in \mathcal{R}$. By constructing a graph with $\lambda = 2$ corresponding to $v_1 \in \mathcal{R}$ and $v_2 \in \mathcal{R}$, with $\mathcal{I}(v_1) = \emptyset$ then also $\mathcal{I}(v_2) = \emptyset$, hence $\gamma < \eta - 1$. Then $d(v_1, v_2) = \infty$ and by Lemma 1, the graph is disconnected, which is a contradiction. □

The above theorem indicates that all internal nodes, in any connected acyclic graph with $\lambda \geq 2$, should be connected to *at least* one internal link. In addition, the condition in the following theorem is the necessary and sufficient condition for G to be identifiable.

Theorem 2. *To identify all links' metrics of acyclic in-vehicle network G, where only nodes in $\mathcal{E}$ can be monitors, the necessary and sufficient topological condition is that $d(u) \geq 3, \forall u \in \mathcal{R}$.*

Proof. See proof of Theorem 1 in [5]. □

The Ethernet-based topology shown in Figure 2 is unidentifiable. This is due to the violation in the topological condition stated in Theorem 2. According to Theorem 2, to make the topology identifiable, the node degree for all $u \in \mathcal{R}$ should be increased by at least 1 (current degree of internal nodes, i.e., gateways $g_1, g_2, \ldots, g_4$, is 2).

Based on the above theoretical analysis, a procedure is derived (i,e., Procedure 1) to check for the topological identifiability condition for any in-vehicle network G. Procedure 1 takes a network G and decides whether it is *identifiable*, i.e., $S_d(G)$ = true, or *unidentifiable*, i.e., $S_d(G)$ = false.

```
Procedure 1: $isIdentifiable(G)$
Output  : $S_d(G)$
Initialize:
$S_d(G) \leftarrow$ true
$\mathcal{R} \leftarrow \{u : d(u) \geq 2\}$
foreach $u \in \mathcal{R}$ do
    if $d(u) < 3$ then
        $S_d(G) \leftarrow false$
        break
return $S_d(G)$
```

4.2. Transformation into Identifiable Topology

To bring an *unidentifiable* topology G to *identifiable* topology G_i, only the number of internal nodes can be preserved, i.e., $\lambda_G = \lambda_{G_i}$. Preserving the number of edge nodes is not guaranteed as in some cases we need to add more links, which in turn requires adding more edge nodes in case the topology is acyclic, this results in $\eta_G \leq \eta_{G_i}$.

Any *unidentifiable* topology G can be classified as either one of the following two cases

- **Case 1**: There is at least one internal node $u \in \mathcal{R}$ with $d(u) < 3$, while $d(v) = 3, \forall v \in \mathcal{R} \backslash u$, where $u \neq v$.
- **Case 2**: There is at least one internal node $u \in \mathcal{R}$ with $d(u) < 3$, while $d(v) \geq 3, \forall v \in \mathcal{R} \backslash u$ with at least one node $w \in \mathcal{R} \backslash u$ having $d(w) > 3$, where $u \neq v \neq w$.

The Ethernet-based topology shown in Figure 1 is an example of a case 2 scenario (the Ethernet switch has degree > 3). In contrast, if the Ethernet switch has degree ≤ 3, then this is a case 1 scenario.

In the following, we describe how to transform the unidentifiable topology G into identifiable topology G_i considering each one of these cases.

4.2.1. Transforming Case 1 topologies

Let $\sigma := |\mathcal{R}_{3^-}|$ be the number of internal nodes with degree < 3, then to transform the topology to identifiable one, for the first case mentioned above, σ links and edge nodes will be added and connected to the internal nodes with degree < 3 so that the condition in Theorem 2 is met. In this case, the total number of links in the identifiable topology G_i will be

$$\gamma_{G_i} = \gamma_G + \sigma \tag{2}$$

where γ_{G_i} and γ_G are the total number of links in G_i and G, respectively.

Procedure 2 is used to transform the topology of this case into identifiable topology.

```
Procedure 2: $Case1(G, \mathcal{R}_{3^-})$
Output: $G_i$
foreach $u \in \mathcal{R}_{3^-}$ do
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{v\}$
    $G \leftarrow G + \{uv\}$
$G_i \leftarrow G$
return $G_i$
```

4.2.2. Transforming Case 2 topologies

Unlike **Case 1**, transforming a topology of **Case 2** is not straightforward. In this case, to keep the number of links to a minimum, instead of adding more links, existing links can be restructured. This is because, for topologies of **Case 2** type, there is at least one internal node u with $d(u) > 3$. So we can disconnect some links incident to u and connect them to other nodes in $\mathcal{R}_{3^-}$ while maintaining node degree of at least three for all internal nodes, hence satisfying Theorem 2. And, if needed, extra links will be added (as described next). There are, however, certain assertions that need to be taken care of when restructuring the topology to ensure that the resulting topology is connected and acyclic.

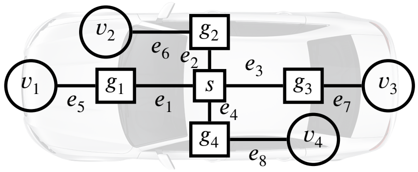


Figure 2: An unidentifiable Ethernet-based topology.

Remark 1. *Let $G^{'}$ be a resulting topology after restructuring any link in unidentifiable topology G, then $G^{'}$ should satisfy the following conditions:*

1. *it is a connected graph*
2. *it has no self-loops;*
3. *it is not multigraph; and*
4. *it has no cycles.*

To restructure the unidentifiable topology, two types of links can be used: *partially restructurable links (PRLs)* and *fully restructurable links (PRLs)*. Figures 3(a) – 3(d) show examples of the difference between the two. In the following, we describe each one of these links and show how they can be used to transform the topology towards an identifiable one.

Transformation using PRLs Figure 3(a) and Figure 3(b) show examples of PRLs and how to restructure them. In addition, the following definition formally defines PRL.

Definition 4. *For node $u \in \mathcal{R}_{3+}$, a* ***partially restructurable link (PRL)*** *$e_i \in \mathcal{I}$ is a link incident to u such that it can be disconnected from it while keeping $d(u) \geq 3$.*

Because PRL e_i is an internal link, then according to Definition 3, its both end-points are internal nodes. Hence, the number of PRL incident to u can be computed by counting the number of internal nodes that are neighbours to u, let $\zeta_u := |\mathcal{N}_{\mathcal{R}}(u)|$ be this number and let $\varphi_u := d(u) - 3$. For identifiable topologies, $\varphi_u \geq 0, \forall u \in \mathcal{R}$, otherwise if the topology is unidentifiable, then $\varphi_u < 0, \exists u \in \mathcal{R}$. Further, let $\psi := |R_{3+}|$, then the following theorem quantifies the maximum number of links that can be restructured in G.

Theorem 3. *For unidentifiable topology G of* ***Case 2****, if $\psi \geq 1$, then the maximum number of PRLs γ_{PRL}, such that $d(u) \geq 3, \forall u \in \mathcal{R}$, is*

$$\gamma_{PRL} = \sum_{i=1}^{\psi} \gamma_{PRL}(u) \tag{3}$$

where

$$\gamma_{PRL}(u) = \begin{cases} \varphi_u, & \text{if } \zeta_u > \varphi_u \\ \zeta_u - 1, & \text{otherwise} \end{cases} \tag{4}$$

Proof. Based on Theorem 2, at least 3 links should be incident to u, $\forall u \in \mathcal{R}$. Hence, for $\psi = 1$, no more than φ_u links can be resturcutred in G. This means that disconnecting φ_u links from u would leave $d(u) = 3$

$$d(u) - \varphi_u = d(u) - (d(u) - 3) = 3$$

However, disconnecting φ_u links can only guarantee $d(u) = 3$ but cannot ensure that disconnecting any of φ_u links would keep the topology connected. For this, Theorem 1 requires that for each internal node $u \in \mathcal{R}$, it should be incident to at least one link in $\mathcal{I}$. Therefore, such a link cannot be used as PRL.

As we know it is necessary for each $u \in \mathcal{R}$ to be connected to at least one link in $\mathcal{I}$, then only if $\zeta_u > \varphi_u$, all φ_u links can be restructured. Otherwise, if $\zeta_u \leq \varphi_u$, restructuring all φ_u links will result in $|\mathcal{I}(u)| < 1$ which violates the condition in Theorem 1. Therefore, to ensure satisfiability of Theorem 1 in this case, only up to $\zeta_u - 1$ can be restructured which proves (4). For $\psi > 1$, the same argument applies for individual nodes in $\mathcal{R}_{3+}$ for which (3) is proven. □

To use PRLs, it is important to consider the conditions in Remark 1. Let $\mathcal{W} := \{u \in \mathcal{R}_{3+} : \zeta_u > 1\}$ be a set of internal nodes having node degree larger than three and are neighbours to more than one internal node, and let $\beta := |\mathcal{W}|$, then the following proposition ensures that $G^{'}$ is not multigraph and does not have self-loops.

Proposition 1. *To ensure that $G^{'}$ is a simple graph, restructuring PRL link $e_i = uv$, in unidentifiable topology G, into $e^{'}_i = vw$, where $u \in \mathcal{W}$, $v \in \mathcal{N}(u)$ and $w \in \mathcal{R}_{3-}$, is the mapping between e_i and $e^{'}_i$ i.e., $e_i \in E(G) \rightarrow e^{'}_i \in E(G^{'})$ such that the following conditions are met*

1. *$v \neq w$; and*
2. *$w \notin \mathcal{N}(v)$ (or $v \notin \mathcal{N}(w)$).*

Proof. Let $G^{'}$ be a resulting topology that has to be a simple graph.
Condition 1: assume that $v = w$ in $G^{'}$, then connecting v and w, which form link $e^{'}_i$, will result in self-loop with $v_h(e^{'}_i) = v_t(e^{'}_i)$ which contradicts that $G^{'}$ is acyclic.
Condition 2: now assume that $w \in \mathcal{N}(v)$ (or $v \in \mathcal{N}(w)$), then connecting w and v will result in multigraph where w and v share two links. Again, this contradicts that $G^{'}$ is acyclic.

□

The conditions defined in Proposition 1 are the necessary conditions to obtain a simple graph. These conditions, however, do not ensure that the resulting graph is connected. To ensure connectivity, Theorem 1 states that $\mathcal{I}(u) \neq \emptyset, \forall u \in \mathcal{R}$. Therefore, $\zeta_u \geq 1, \forall u \in \mathcal{R}$. To restructure any PRL, it is important to ensure that this is satisfied for all internal nodes in the resulting graph $G^{'}$.

Assuming that $\mathcal{W} \neq \emptyset$, and $\sigma \geq 1$, then the following is the sufficient condition to have acyclic and connected topology satisfying all conditions in Remark 1.

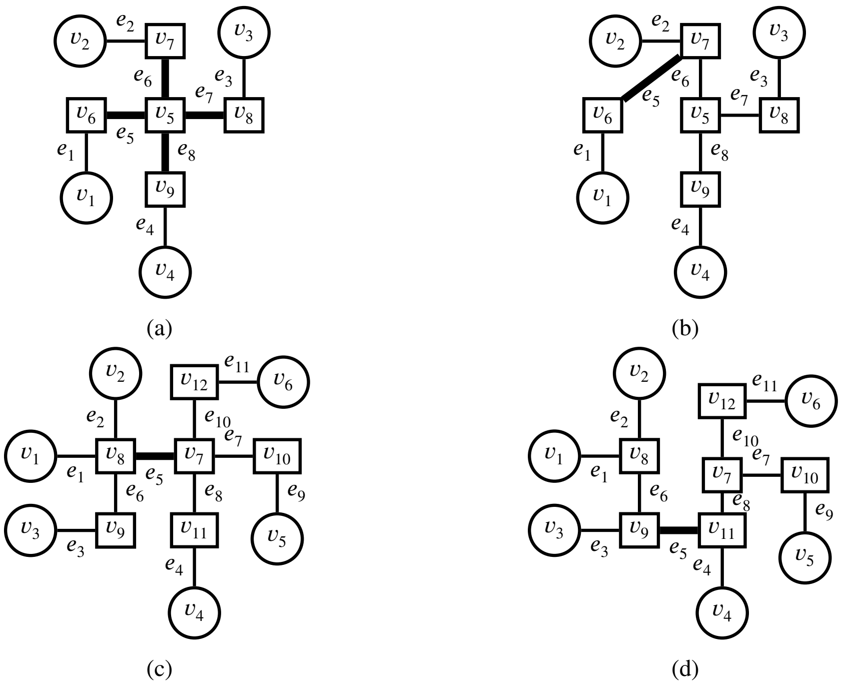


Figure 3: (a) $e_5, \ldots, e_8$ are PRLs, (b) topology after restructuring PRL e_5, (c) e_5 is FRL, (d) topology after restructuring FRL e_5.

Proposition 2. *Any PRL $e_i = uv$ in G can be disconnected from its end-point $u \in \mathcal{W}$ and connected to $w \in \mathcal{R}_{3^-}$ such that the resulting graph $G^{'} = G^* + vw$ is acyclic and connected iff $v \in c_1$ and $w \in c_2$ (or $v \in c_2$ and $w \in c_1$) in G^*, where $G^* = G - uv$ and $e_1, e_2 \in C(G^*)$.*

Proof. → Provided that $vw \notin E(G)$, we assume $G^{'} = G^* + vw$, where $G^* = G - uv$, is acyclic and connected graph, and prove that $v \in c_1$ and $w \in c_2$ (or $v \in c_2$ and $w \in c_1$). We prove this by contradiction, assuming that $G^{'}$ is not acyclic and not connected. We know that the original graph G is acyclic and connected with number of links $\gamma_G = \eta_G - 1$ then for $G^* = G - uv$ the number of links decreases by one, hence, $\gamma_{G^*} < \eta_{G^*} - 1$ then if $v \in c_1$ and $w \in c_2$ (or if $v \in c_2$ and $w \in c_1$), the resulting graph $G^{'} = G^* + vw$ would have $\gamma_{G'} = \eta_{G'} - 1$ and according to Lemma 1, $G^{'}$ is acyclic and connected graph, which contradicts the assumption.
← We assume for G^* that $v \in c_1$ and $w \in c_2$ (or $v \in c_2$ and $w \in c_1$) and prove that the resulting graph $G^{'} = G^* + vw$ is acyclic connected graph. For this, we prove the contrapositive that if both $v, w \in c_1$ (or $v, w \in c_2$), then $G^* + vw$ is *not* acyclic and *not* connected graph. Since G is maximally acyclic, then $c_1, c_2 \in C(G^*)$ must also be maximally acyclic. Thus, $c_1 + vw$ ($c_2 + vw$) results in having a cycle in c_1 (c_2) while the graph is still disconnected. □

Procedure 3 shows how PRLs are used to transform an unidentifiable topology towards an identifiable one. While both σ and β are larger than 0, the procedure starts by looping through elements of $\mathcal{W}$ and checking their neighbours. If a neighbour $v \in \mathcal{N}(u)$ is not an internal node, it will be skipped (lines 1-6). Otherwise, the link between $u \in \mathcal{W}$ and $v \in \mathcal{N}(u)$ will be disconnected and the resulting two

Procedure 3: $PRL(G, \mathcal{R}_{3^-}, \mathcal{W})$

```
    Output : $G_{PRL}$
    Initialize:
    $G_{PRL} \leftarrow G$
    $\sigma \leftarrow |\mathcal{R}_{3^-}|$
    $\beta \leftarrow |\mathcal{W}|$
    $\mathcal{R} \leftarrow \{u \in V(G) : d(u) \geq 2\}$
    $C \leftarrow \emptyset$
 1  while $\sigma > 0$ and $\beta > 0$ do
 2      for $i = 1 : \beta$ do
 3          $u = \mathcal{W}[i]$
 4          for each $v \in \mathcal{N}(u)$ do
 5              if $v \notin \mathcal{R}$ then
 6                  continue
 7              else
 8                  $G \leftarrow G - \{uv\}$
 9                  $C \leftarrow C \cup \{c_1, c_2\}$
10                  for $k = 1 : \sigma$ do
11                      $w = \mathcal{R}_{3^-}[k]$
12                      if ($v \in c_1$ and $w \in c_2$) or
                         ($v \in c_2$ and $w \in c_1$) then
13                          $G \leftarrow G + \{vw\}$
14                          update $\beta$ and $\sigma$
15                      else
16                          continue
17  $G_{PRL} \leftarrow G$
18  return $G_{PRL}$
```

components c_1, c_2 will be added to C (lines 7-9). If a node $w \in \mathcal{R}_{3-}$ is in a different component than v, a link between these nodes will be added (lines 10-14).

Transformation using FRLs In some cases, a link can be completely disconnected from its both end-points and connected to other end-points. We call such link *fully restructurable link (FRL).*

Definition 5. ***fully restructurable link (FRL)*** *is a link $e_i \in \mathcal{I}$ with $e_i = uv$ such that both $u \in \mathcal{R}_{3+}$ and $v \in \mathcal{R}_{3+}$, and the number of internal links they are incident to is larger than 1, i.e., $|\mathcal{I}(u)| > 1$ and $|\mathcal{I}(v)| > 1$. Therefore, fully restructurable links can only exist in scenarios where $\beta > 1$.*

An example of FRL and how it can be restructured is shown in Figure 3(c) and Figure 3(d). Using FRLs instead of PRLs can speed up the transformation process by reducing σ by a factor of 2. This is because they can be disconnected from their endpoints and used to connect the other two nodes in $\mathcal{R}_{3-}$. However, as in the case of PRLs, FRLs should satisfy the conditions in Remark 1. Similar to PRLs, the following conditions are necessary for G' to be acyclic.

Proposition 3. *To ensure that G' is a simple graph, restructuring FRL $e_i = uv$, in unidentifiable topology G, into $e'_i = wz$, where $u, v \in \mathcal{W}$ with $v \in \mathcal{N}(u)$ and $w, z \in \mathcal{R}_{3-}$, is the mapping between e_i and e'_i i.e., $e_i \in E(G) \rightarrow e'_i \in E(G')$ such that the following conditions are met*

1. *$w \neq z$; and*
2. *$w \notin \mathcal{N}(z)$ (or $z \notin \mathcal{N}(w)$).*

Proof. Replacing w by v and z by w, then the proof is similar to the one for Proposition 1 □

Assuming that $\beta \geq 2$ and there are at least two nodes in $\mathcal{W}$ that are neighbours, then the following is the sufficient condition to ensure having acyclic and connected topology G'.

Proposition 4. *Any FRL in unidentifiable topology G can be disconnected from their end-points, $u \in \mathcal{R}_{3+}$ and $v \in \mathcal{R}_{3+}$, and reconnected to other end-points $w \in \mathcal{R}_{3-}$ and $z \in \mathcal{R}_{3-}$, such that the resulting graph G' is acyclic and connected iff for $G - uv$, $w \in c_1$ (or $w \in c_2$) and $z \in c_2$ (or $z \in c_1$), where $c_1, c_2 \in C(G^*)$ and $G^* = G - \{uv\}$.*

Proof. Replacing w with v and z with w in Proposition 2 then the proof is similar to that of Proposition 2. □

Based on the above theoretical analysis, Procedure 4 is derived, illustrating how FRLs can be used to transform an unidentifiable topology towards an identifiable one. This procedure is similar to Procedure 3 except that it uses FRL when $\beta \geq 2$ and $\sigma \geq 2$. It finds two neighbouring nodes in $\mathcal{W}$ and disconnects them, then it checks if two nodes $w, z \in \mathcal{R}_{3-}$ are in different components, if so a link will be added between them.

Procedure 4: $FRL(G, \mathcal{R}_{3-}, \mathcal{W})$

```
Output  : $G_{FRL}$
Initialize:
$G_{FRL} \leftarrow G$
$C \leftarrow \emptyset$
$\sigma \leftarrow |\mathcal{R}_{3-}|$
$\beta \leftarrow |\mathcal{W}|$
1  while $\beta \geq 2$ and $\sigma \geq 2$ do
2      for $i = 1 : \beta$ do
3          $u = \mathcal{R}_{3+}[i]$
4          for $j = i + 1 : \beta$ do
5              $v = \mathcal{R}_{3+}[j]$
6              if $v \in \mathcal{N}(u)$ then
7                  $G \leftarrow G - \{uv\}$
8                  $C \leftarrow C \cup \{c_1, c_2\}$
9                  for $k = 1 : \sigma$ do
10                     $w = \mathcal{R}_{3-}[k]$
11                     for $m = k + 1 : \sigma$ do
12                         $z = \mathcal{R}_{3-}[m]$
13                         if ($w \in c_1$ and $z \in c_2$) or
                            ($z \in c_1$ and $w \in c_2$) then
14                             $G \leftarrow G + \{wz\}$
15                             update $\beta$ and $\sigma$
16                         else
17                             continue
18 $G_{FRL} \leftarrow G$
19 return $G_{FRL}$
```

Transforming an unidentifiable topology G into identifiable G_i by restructuring links using either PRLs or FRLs results in the total number of links γ_{G_i} in G_i being

$$\gamma_{G_i} = \begin{cases} \gamma_G, & \text{if } \varphi_u = \sigma \text{ and } \varphi_u < \zeta_u \\ \gamma_G, & \text{if } \zeta_u - 1 = \sigma \text{ and } \varphi_u \geq \zeta_u \\ \gamma_G + (\sigma - \varphi_u), & \text{if } \varphi_u < \sigma \text{ and } \varphi_u < \zeta_u \\ \gamma_G + (\sigma - \zeta_u + 1), & \text{if } \zeta_u - 1 < \sigma \text{ and } \varphi_u \geq \zeta_u \end{cases} \quad (5)$$

4.3. Transformation Algorithm

The transformation algorithm (Algorithm 5) starts by checking if $\beta > 0$, if so it checks if both $\beta \geq 2$ and $\sigma \geq 2$. In this case, it uses FRL for the transformation using Procedure 4. If the resulting topology is identifiable, it returns it and stops the algorithm (lines 4-6). Otherwise, it updates the values for β and σ and uses the transformation with PRL using Procedure 3 (lines 7-9). Again, it checks if the resulting topology is identifiable or not. If it is unidentifiable, it uses the transformation for case 1 using Procedure 2 after updating β and σ (lines 13-16). If $\beta = \sigma = 1$, then the algorithm uses Procedure 3, checks for identifiability and uses Procedure 2 if the resulting topology is unidentifiable (lines 19-28).

Algorithm 5: Transform to identifiable topology

```
   Inputs  : $G$
   Output  : $G_i$
   Initialize:
   $\mathcal{R}_{3^-} \leftarrow \{u \in \mathcal{R} : d(u) < 3\}$
   $\mathcal{R}_{3^+} \leftarrow \{u \in \mathcal{R} : d(u) > 3\}$
   $\mathcal{W} \leftarrow \{u \in \mathcal{R}_{3^+} : \zeta_u > 1\}$
   $\beta \leftarrow |\mathcal{W}|$
   $\sigma \leftarrow |\mathcal{R}_{3^-}|$
1  if $\beta > 0$ then
2      if $\beta \geq 2$ and $\sigma \geq 2$ then
3          $G_{FRL} \leftarrow FRL(G, \mathcal{R}_{3^-}, \mathcal{W})$
4          if $S_d(G_{FRL}) = true$ then
5              $G_i \leftarrow G_{FRL}$
6              go to line 28
7          else
8              update $\beta$ and $\sigma$
9              $G_{PRL} \leftarrow PRL(G_{FRL}, \mathcal{R}_{3^-}, \mathcal{W})$
10             if $S_d(G_{PRL}) = true$ then
11                 $G_i \leftarrow G_{PRL}$
12                 go to line 28
13             else
14                 update $\beta$ and $\sigma$
15                 $G_i \leftarrow Case1(G_{PRL}, \mathcal{R}_{3^-})$
16                 go to line 28
17     else
18         $G_{PRL} \leftarrow PRL(G_{FRL}, \mathcal{R}_{3^-}, \mathcal{W})$
19         if $S_d(G_{PRL}) = true$ then
20             $G_i \leftarrow G_{PRL}$
21             go to line 28
22         else
23             update $\mathcal{R}_{3^-}$
24             update $\mathcal{W}$
25             $G_i \leftarrow Case1(G_{PRL}, \mathcal{R}_{3^-})$
26             go to line 28
27 $G_i \leftarrow Case1(G, \mathcal{R}_{3^-})$
28 return $G_i$
```

In general, an unidentifiable topology is not guaranteed to have FRL for which Procedure 4 might be used, in this case, PRL, if existed will be used (Procedure 3). Otherwise, the problem will be reduced to case 1 transformation (Procedure 2). A flowchart summary of how the overall transformation algorithm works is depicted in Figure 4.

4.3.1. Complexity analysis

The time complexity for checking for identifiability condition in Procedure 1 is $\mathcal{O}(\eta + \lambda)$ where η is the total number of nodes and λ is the total number of internal nodes in G. Procedure 2 takes $\mathcal{O}(\sigma)$, while restructuring links takes $\mathcal{O}(\beta \cdot d(u) \cdot \sigma)$ using PRL (Procedure 3) and $\mathcal{O}(\beta^2 \cdot \sigma^2)$ using FRL (Procedure 4), where $\beta := |\mathcal{W}|$ and $u \in \mathcal{W}$. Thus, the complexity of the overall transformation algorithm is

$$\text{Complexity} = \begin{cases} \mathcal{O}(\eta + \lambda + \beta^2\sigma^2), & \text{if } \beta \geq 2 \text{ and } \sigma \geq 2 \\ \mathcal{O}(\beta \cdot d(u) + \frac{\eta+\lambda}{\sigma}), & \text{if } \beta = 1 \text{ or } \sigma = 1 \\ \mathcal{O}(\sigma), & \text{if } \beta = 0 \end{cases} \tag{6}$$

From (6), it is clear that the worst case scenario is when $\beta, \sigma \geq 2$, in which the algorithm takes $\mathcal{O}(\eta + \lambda + \beta^2\sigma^2)$.

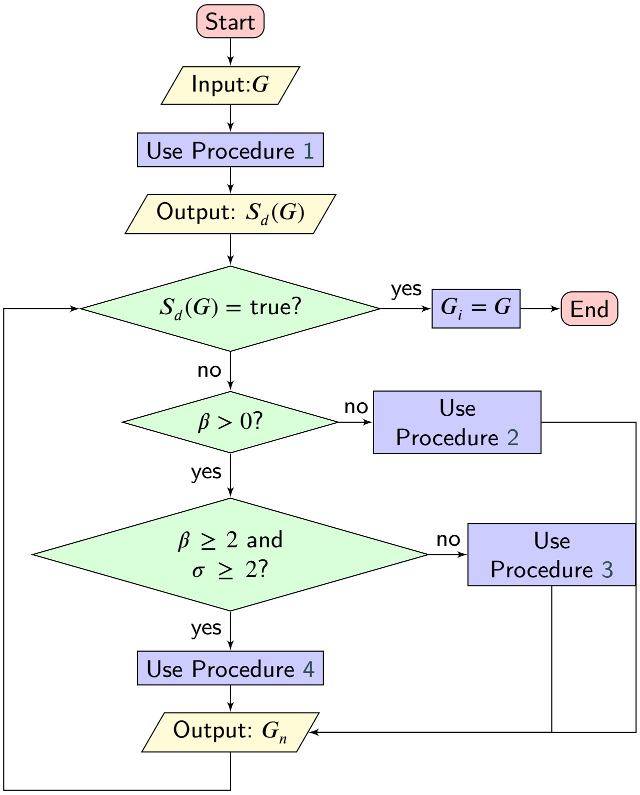


Figure 4: A flowchart illustrating the transformation algorithm of unidentifiable topology G into identifiable topology G_i.

Note that the transformation algorithm shown in Algorithm 5 only transforms any unidentifiable topology into an identifiable one without considering whether the resulting topology has a minimum number of links (i.e., being *optimal*) or not. The following section discusses this and provides an optimisation algorithm that ensures that the identifiable topology has a minimum number of links.

5. Topology Optimisation

This section proposes an optimisation algorithm for any identifiable topology. The goal is to achieve a minimum number of links while keeping the topology identifiable. Any topology that meets these two criteria is therefore *optimal*.

5.1. Conditions

As mentioned earlier, the Ethernet-based topology shown in Figure 2 is *unidentifiable*. And by using the transformation algorithm (Algorithm 5), the number of links in the resulting topology G_i will be increased. This results in $\gamma_{G_i} > \gamma_G$, where G is the original topoogy.

However, in automotive networks, where the wiring adds up to the total weight and complexity of the vehicle, it is desirable to minimise the number of links. The following theorem quantifies the minimum number of links required to achieve identifiable topology.

Theorem 4. *The minimum number of links that can exist in an identifiable topology G with λ internal nodes is*

$$2\lambda + 1 \tag{7}$$

Proof. From Lemma 1, we know that we need to have at least $\eta - 1$ links in G. Otherwise, the topology would be disconnected. However, Lemma 1 ensures connectivity, but it does not ensure identifiability. To ensure having a topology that is identifiable, the condition in Theorem 2 has to be satisfied. Without loss of generality, we can assume a topology with internal nodes only and let $\lambda := |\mathcal{R}|$. Because each $u \in \mathcal{R}$ should have at least $d(u) = 3$ to achieve identifiability, 3λ links are needed, including the $\lambda - 1$ links required to have a connected graph. Therefore, the minimum possible number of links that can exist in identifiable topology is $3\lambda - (\lambda - 1) = 2\lambda + 1$. □

Note that (7) is just the minimum number of links in an identifiable topology. In practice, the total number of links can be larger. Thus, any network topology with fewer number of links implies that the topology is *unidentifiable*.

The aim now is to convert any identifiable topology with links larger than $2\lambda + 1$ into an optimal topology with only $2\lambda + 1$ links. The next section describes this conversion.

5.2. Optimisation Algorithm

The optimal topology that satisfies the identifiability condition is the one with $2\lambda + 1$ (see Theorem 4). Generally, there are two cases of the given identifiable topology G_i:

1. topology is identifiable and $\mathcal{R}_{3+} = \emptyset$;
2. topology is identifiable and $\mathcal{R}_{3+} \neq \emptyset$.

In the first case, $\gamma_{G_i} = 2\lambda + 1$ and the topology is already optimal according to Theorem 4, whereas in the second, $\gamma_{G_i} > 2\lambda + 1$ and the topology is not yet optimal. The following discusses how this topology can be optimised.

Recall that the new E/E architectures focus on utilising a few numbers of powerful ECUs instead of a large number of limited-capabilities ECUs. The optimisation algorithm is designed to support this goal in which it can consolidate different ECUs into one single ECU, such ECU can be thought of as a High-Performance Computing Platform (HPCP) [29, 30].

The conversion can be done by removing $\gamma - (2\lambda + 1)$ links. Then, the nodes connected to the removed links can be mapped to other nodes. The following shows how these nodes can be mapped.

Let $u \in \mathcal{R}_{3+}$, then the list of nodes (connected to u) to be mapped is the candidate set $\mathcal{M}_u$. To ensure the identifiability of the topology, the number of nodes to be mapped should not be larger than φ_u, in other words, $|\mathcal{M}_u| = \varphi_u$.

Procedure 6: *$nodesListToMap(G_i, u)$*

```
   Output  : $\mathcal{M}_u$
   Initialize:
   $\varphi_u \leftarrow d(u) - 3$
   $\mathcal{M}_u \leftarrow \emptyset$
1  for $j = 1 : |\mathcal{N}(u)|$ do
2      if $\mathcal{N}(u)[j] \in \mathcal{E}$ then
3          $\mathcal{M}_u \leftarrow \mathcal{M}_u \cup \{\mathcal{N}(u)[j]\}$
4          if $|\mathcal{M}_u| > \varphi_u$ then
5              $\mathcal{M}_u \leftarrow \mathcal{M}_u \backslash \{\mathcal{N}(u)[j]\}$
6              break
7  if $\mathcal{N}(u) \cap \mathcal{E} \neq \emptyset$ then
8      $\mathcal{M}_u \leftarrow \mathcal{M}_u \backslash \{v_a\}$
9  for $k = 1 : |\mathcal{N}(u)|$ do
10     if $\mathcal{N}(u)[k] \in \mathcal{R}$ then
11         $\mathcal{M}_u \leftarrow \mathcal{M}_u \cup \{\mathcal{N}(u)[k]\}$
12         if $|\mathcal{M}_u| > \varphi_u$ then
13             $\mathcal{M}_u \leftarrow \mathcal{M}_u \backslash \{\mathcal{N}(u)[k]\}$
14             break
15 return $\mathcal{M}_u$
```

The process for finding $\mathcal{M}_u$ is shown in Procedure 6. It first prioritises mapping edge nodes over intermediate nodes. So it starts by looping through the neighbouring nodes of $u \in \mathcal{R}_{3+}$. If a neighbour is an edge node, it will be added to the list. After the addition of each node to the list, the procedure checks if the number of elements in the list is larger than φ_u. If so, the recent node added will be removed from the list and the procedure stops (lines 4-6). Additionally, if there is an edge node(s) connected to u, then we need to ensure that not all of them will be in the list as there should be at least one edge node to map to (lines 7-8). Then the procedure loops again through the neighbouring nodes of u, this time checking if each neighbour is an intermediate node, and do the same as with edge nodes while ensuring that $|\mathcal{M}_u| \not> \varphi_u$ (lines 9-14).

Algorithm 7 shows how the topology can be optimised so that $\gamma_{G_o} = 2\lambda + 1$.

The optimisation algorithm is only needed when $\mathcal{R}_{3+} \neq \emptyset$ as otherwise the topology is optimal. The algorithm starts by looping through nodes in $\mathcal{R}_{3+}$ and for each of these nodes, it chooses the set of candidate nodes to map $\mathcal{M}_u$ using Procedure 6 (lines 1-4). Next, for each candidate node v, if it is an edge node, then it can simply be mapped into another edge node and hence removed from the topology (lines 5-8). Otherwise, if it is an internal node, the algorithm removes this node and maps it to one of the existing internal nodes (line 18). However, it needs to reconnect the link previously connected to v to the new node. For this, the algorithm makes sure that the new link does not result in cycles as the case with the identifiability algorithm (lines 9-17). Then, $\mathcal{R}_{3+}$

```
Algorithm 7: Transform to optimal topology
    Inputs   : $G_i$
    Output   : $G_o$
    Initialize:
    $\lambda = |\mathcal{R}_{G_i}|$
    $\mathcal{R}_{3+} \leftarrow$ get_$\mathcal{R}_{3+}(G_i)$
 1  while $\mathcal{R}_{3+} \neq \emptyset$ do
 2      for $i = 1 : \psi$ do
 3          $u \leftarrow \mathcal{R}_{3+}[i]$
 4          $\mathcal{M}_u \leftarrow$ nodesListToMap$(G_i, u)$
 5          for $j = 1 : |\mathcal{M}_u|$ do
 6              $v \leftarrow \mathcal{M}_u[j]$
 7              if $v \in \mathcal{E}$ then
 8                  $G_i \leftarrow G_i - \{v\}$
 9              else if $v \in \mathcal{R}$ then
10                  for $k = 1 : |\mathcal{N}(v)|$ do
11                      $w \leftarrow \mathcal{N}(v)[k]$
12                      for $m = 1 : |\mathcal{R}|$ do
13                          $z \leftarrow \mathcal{R}[m]$
14                          $G_i - \{wz\}$
15                          $\mathcal{C} \leftarrow \{c_1, c_2\}$
16                          if $(w \in c_1$ and $z \in c_2)$ or
                             $(z \in c_1$ and $w \in c_2)$ then
17                              $G_i \leftarrow G_i + \{wz\}$
18                  $G_i \leftarrow G_i - \{v\}$
19      $\mathcal{R}_{3+} \leftarrow$ get_$\mathcal{R}_{3+}(G_i)$
20  $G_i = G_o$
21  return $G_o$
```

will be updated (line 19). Finally, the algorithm stops when $\mathcal{R}_{3+} = \emptyset$ and returns the optimised algorithm G_o.

5.3. Complexity Analysis

Procedure 6 runs in $\mathcal{O}(\xi)$, where $\xi := \left|\mathcal{N}(u)\right|$.

Thus, the overall complexity of the optimisation algorithm is

$$\text{Complexity} = \begin{cases} \mathcal{O}(\psi \cdot \xi \cdot \tau \cdot \omega \cdot \lambda), & \text{if } v \in \mathcal{R} \\ \mathcal{O}(\psi \cdot \xi \cdot \tau), & \text{if } v \in \mathcal{E}, \end{cases} \tag{8}$$

where $\omega := \left|\mathcal{N}(v)\right|$ and $\tau := \left|\mathcal{M}_u\right|$. Hence, the worst-case running time can occur when $v \in \mathcal{R}$, where $v \in \mathcal{M}_u$ and $u \in \mathcal{R}_{3+}$.

6. Discussion

In this section, we highlight the relevance of topology optimisation and the minimal weight of in-vehicle networks, a desirable feature for any vehicle network. Moreover, we briefly discuss the redundancy feature and how it can still be achieved even when the topology is optimal and with minimal added weight compared with the non-optimal topologies.

6.1. Minimal Weight with Topology Optimisation

It is important to note that modern E/E architectures for automotive networks are increasingly moving towards centralization, facilitated by SDN and SOA (Service Oriented Architecture), allowing the integration of multiple nodes into a single unit to reduce the weight of in-vehicle networks. In other words, the aim is to replace the numerous resource-constrained ECUs in current networks with a smaller number of highly powerful ECUs [7, 31]. In light of this shift, the proposed optimisation algorithm is one way of reducing the vehicle's network weight as well as ensuring its full identifiability.

6.2. Redundancy in In-vehicle Networks

Most in-vehicle networks were originally designed without redundancy (the redundancy index for these topologies is zero), based on the assumption that they were inherently robust against failures. However, this perspective has changed recently with the increased connectivity of vehicles to external networks, making them more susceptible to cyberattacks. Therefore, it is important to redesign the in-vehicle network such that it is robust against failures.

Although the network robustness is not the main focus of this study (the main focus of this work is to study identifiability properties in in-vehicle networks in order to achieve fully identifiable topologies with minimal weight), we briefly highlight the usefulness of optimising the identifiable topologies in achieving such robustness properties, in particular achieving redundant in-vehicle network topologies by augmenting them with fewer numbers of links.

Robust in-vehicle networks are defined as those that can still operate under failures, i.e., fail-operational networks [32]. This can be translated into having redundant components that can be used once the original ones are down. More formally, we define network redundancy as having at least two disjoint routes between any two communicating nodes.

In this work, we show that both the optimal and non-optimal topologies can be augmented with more links in order to become redundant. For instance, consider the central-gateway topologies shown in Figure 5. Figure 5(a) and Figure 5(c) are non-redundant topologies for the non-optimal and the optimal versions, respectively. To add redundancy to these topologies, two more internal nodes (g_2 and g_3) should be added and connected to the central gateway g_1 (forming a cycle). As shown, the optimal topology could still be redundant with a fewer number of links than the non-optimal version while maintaining full identifiability.

7. Evaluation

In this section, we evaluate the proposed transformation algorithms. First, the algorithm for transforming an unidentifiable topology into an identifiable one will be evaluated followed by the evaluation of the optimisation algorithm proposed to achieve an identifiable topology with a minimum number of links. Further, these algorithms will be applied to real in-vehicle network topologies and the results will be shown.

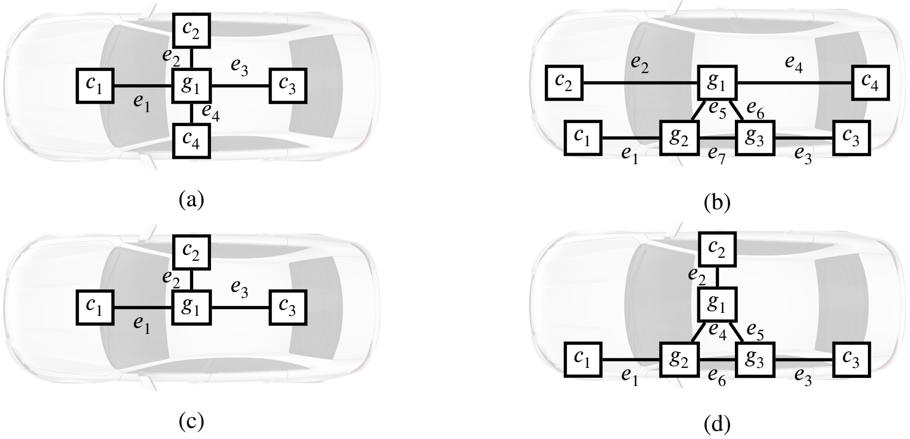


Figure 5: Augmenting the topology to achieve redundancy. (a) non-optimal topology, (b) redundant topology for topology in (a), (c) optimal topology, and (d) redundant topology for the optimal topology in (c). Note that the topologies shown in (b) and (d) were restructured to achieve full identifiability.

7.1. Transformation Algorithms

Using MATLAB, we evaluated the proposed transformation algorithms by simulating random topologies with different numbers of nodes in the range [10, 100]. Because topologies are generated randomly, they are neither guaranteed to be connected nor acyclic. Therefore, we checked each topology for these conditions to be met. We removed cycles if existed, and connected the topology if it had more than one component (i.e., disconnected). Let χ_η and χ_γ be the number of added nodes and links, respectively[4], in the transformed topology, then in the following, we discuss the results for identifiability transformation of unidentifiable topologies into identifiable ones. For each value of $\eta \in \{10, \dots, 100\}$, we ran the simulation for 100 times with different topologies. Next, we show how the optimisation algorithm further improved the resulting identifiable topologies by minimising the number of links.

7.1.1. *Transformation to Identifiable Topology ($G \longrightarrow G_i$)*

We evaluated the transformation using PRLs, FRLs, and the basic method of adding more σ links and nodes as illustrated in Procedure 2. Transformation results depicted in Figure 6(a) show the number of additional nodes and links in $\boldsymbol{G_i}$. As shown, the additional number of nodes and links are the same in each scenario, this is because adding any node requires adding a link to connect it to the network. In addition, using either PRL or FRL results in the same number of added links and nodes. On the other hand, using Procedure 2 ($\eta_G + \sigma$ and $\gamma_G + \sigma$), results in more nodes and links than FRL and PRL. The maximum weight added using FRL is only 10.96% of the original topology before the transformation when $\eta = 100$.

Although the number of added nodes and links for PRL and FRL are the same, we highlight the benefit of using FRL over PRL in terms of speed. This can be seen in Figure 6(b) where we show the average (over a number of topologies that had FRL links in the first repetition) value of σ during each iteration of the transformation. As seen, the σ value in the case of PRL is larger than that of FRL. This is because FRL reduces σ by 2 in a single iteration, while PRL reduces it by 1 in each iteration. Therefore, FRL can transform the topology much faster than PRL.

7.1.2. *Transformation to Optimal Topology ($G \longrightarrow G_o$)*

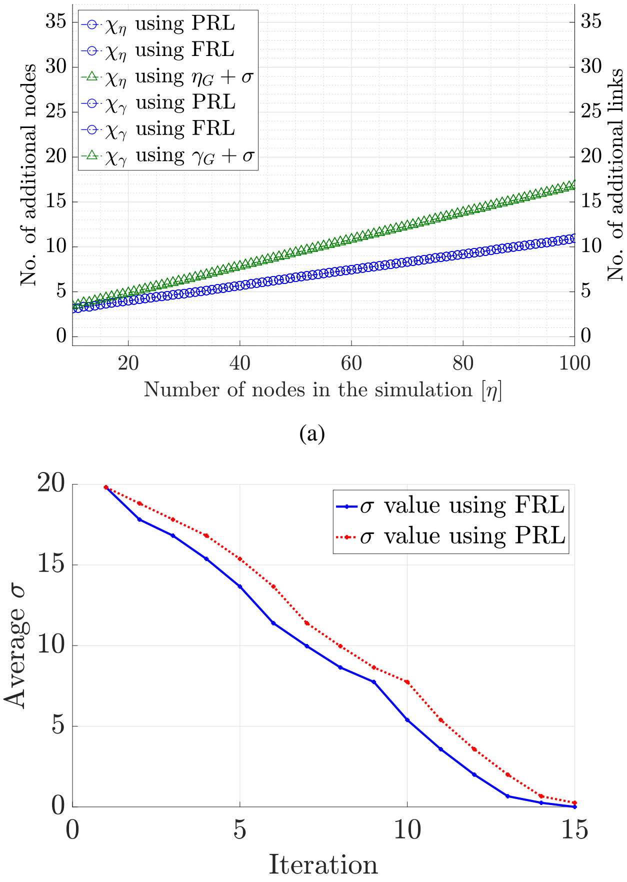
The resulting topologies of the transformation algorithm into identifiable topologies are then fed to the optimisation algorithm. The results of this optimisation are shown in Figure 7.

Figure 7(a) shows the added number of nodes and links (averaged over 100 repetitions) after transforming $\boldsymbol{G}$ into identifiable topology $\boldsymbol{G_i}$ as well as after transforming the identifiable topology $\boldsymbol{G_i}$ into optimal topology $\boldsymbol{G_o}$. As shown, the additional weight in $\boldsymbol{G_o}$ is reduced compared to $\boldsymbol{G_i}$. Additionally, Figure 7(b) shows that the added number of links in the optimised topology $\boldsymbol{G_o}$ is the same as the theoretical value of the minimum number of links $(2\lambda + 1)$ derived in Theorem 4. Furthermore, Figure 7(c) illustrates the ratio of additional weight in the transformed topology to the weight of the original topology. The worst-case scenario is when $\eta = 10$ where the ratio of the added link is $\approx 0.35\%$ for $\boldsymbol{G_i}$. This is because this topology was random and had way fewer links (and nodes) than the required minimum. Thus, the added weight is larger for such topologies. On the other hand, for some topologies, it has been observed that the optimisation algorithm even reduced the weight of the original topologies. The blue and green plots in Figure 7(c) indicate that the optimisation algorithm further improved the overall weight to almost 0% when η approaches 100.

7.2. Algorithms Application on In-Vehicle Networks

This section evaluates the use of the proposed algorithms to transform real in-vehicle network topologies.

[4] When saying $\chi_{(G)}$, we refer to the overall added weight regardless of whether it is for nodes or for links



(a)

(b)

Figure 6: Transformation results: (a) number of added nodes and links after transformation into G_i, (b) values of σ when using FRL and PRL.

These topologies are shown in Figure 8. In Figure 8(a), we show a simple in-vehicle network topology with two CAN networks connected through an Ethernet network. The second topology shown in Figure 8(b) is based on central-gateway architecture where there is a single central gateway connecting multiple CANs (9 CANs in this example). Figure 8(c) shows an advanced topology that is used in modern vehicles. It is based on central architectures where an Ethernet switch connects different networks. On the other hand, the topology shown in Figure 8(d) is a more complicated one and it is based on a real car prototype, i.e., RECBAR [33].

For each topology G, Procedure 1 is used to check if the topology is identifiable or not. If not, Algorithm 5 is used to transform it into an identifiable one. The resulting topology G_i is then checked for optimality, if the topology is not optimal, Algorithm 7 is used to optimise it. The number of links γ_{G_o} in the resulting optimised topology G_o is then compared with the minimum number derived in Theorem 4 (i.e., $2\lambda + 1$). If the topology is already identifiable (or optimal), then $G_i = G$ (or $G_o = G$).

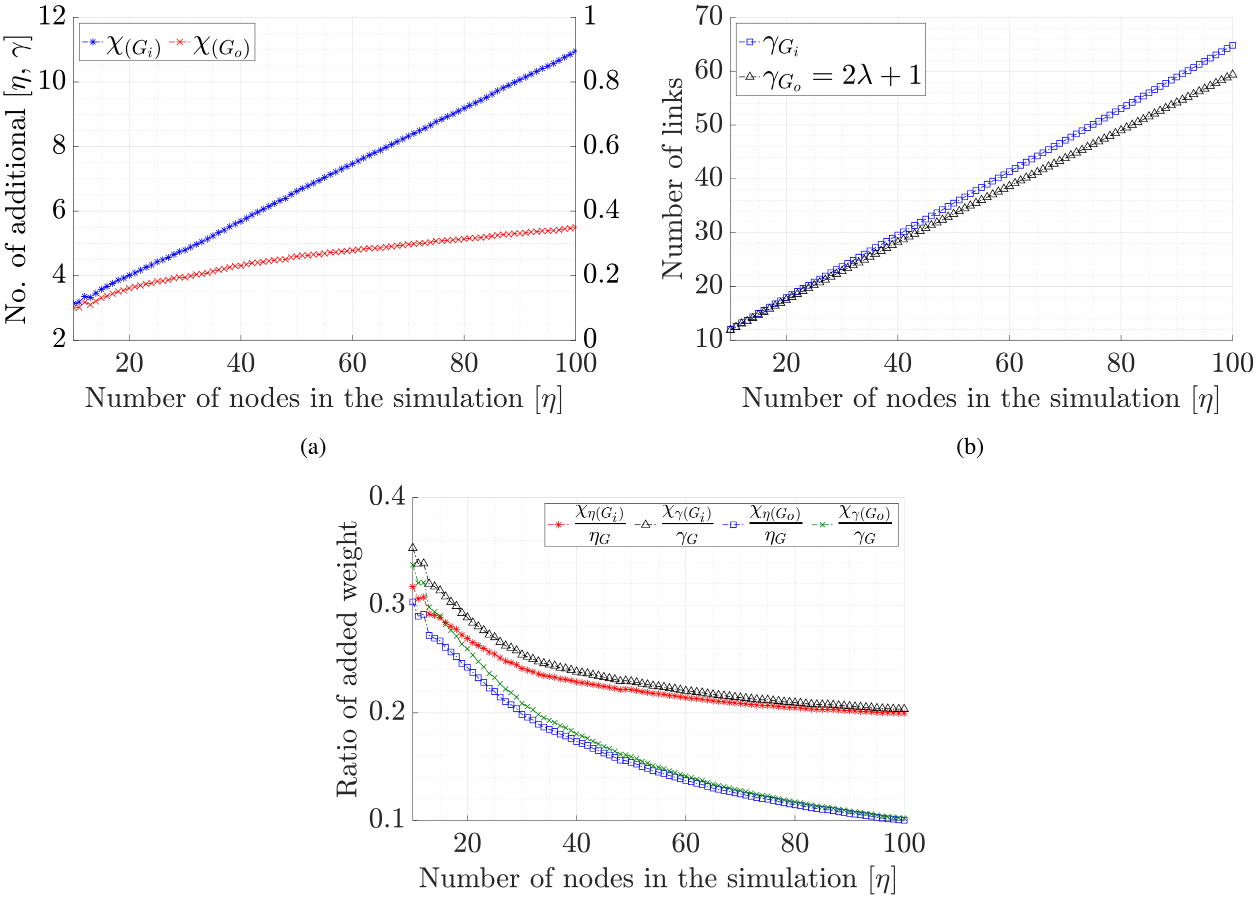
The results are shown in Figure 9. The results here are normalised by the original network weight. For the first topology shown in Figure 8(a), the identifiability algorithm resulted in an added weight of only 1% of the original topology, and the resulting topology is already optimal with $2\lambda + 1$ links. For the topology shown in Figure 8(b), the identifiability check indicated that the topology is already identifiable, however, it is not optimal. The optimisation algorithm then reduced the number of nodes and links by 6%. On the other hand, the topology shown in Figure 8(c) is not identifiable and hence the transformation algorithm resulted in adding only three nodes as well as links (equivalent to 3% of the original topology weight). And the resulting identifiable topology is already optimal. The RECBAR topology shown in Figure 8(d) is unidentifiable. Using the identifiability algorithm, only restructurable links are used without adding any extra weight, hence the same weight for the identifiable topology G_i. However, this topology is not optimal. Using the optimisation algorithm, the weight was reduced by 2% of the total original weight.

The above results show the effectiveness of the proposed transformation algorithms in achieving a fully identifiable topology while minimising the weight cost.

7.3. Comparison with Existing Solutions

It is important to mention that the monitoring solutions in the existing literature do not focus on the internal monitoring of the vehicle network. This is, as mentioned earlier, because it is not possible to access the internal networking elements. What they focus on instead is monitoring the end-to-end traffic and its performance. In this work, we compare our network tomography monitoring with two of the state-of-the-art monitoring solutions: OTIDS [12] and COIDS [13]. In particular, we highlight the advantage of network tomography in terms of the number of nodes participating in the monitoring process as well as the number of monitoring messages. In addition, we compare the number of uniquely identifiable links with partial network tomography [26].

Figure 10(a) shows the number of monitoring nodes $|\mathcal{E}_m|$ in each of the topologies shown in Figure 8 (where G, G_i, G_o are the original, identifiable and optimised topology, respectively). The results here are also normalised by the original network size. It is clear that all approaches utilise fewer number of nodes as monitors compared to the total number of nodes in the network. Assuming that each CAN is connected to one CAN node, both OTIDS and COIDS use the same number of monitors. They use an additional node physically plugged into the CAN bus in order to monitor its traffic while the other edge nodes transmit the monitoring messages. Therefore, for both, the number of monitoring nodes is $|\mathcal{E} + 1|$. For the second topology, the number of monitoring nodes is fewer than the number of total nodes in the topology by 6. In contrast, NT uses fewer number of monitors (in G) as it only needs $|\mathcal{E}|$ monitors. However, in G_i, due to the potential increase in the number of nodes during the transformation process, the number of monitors

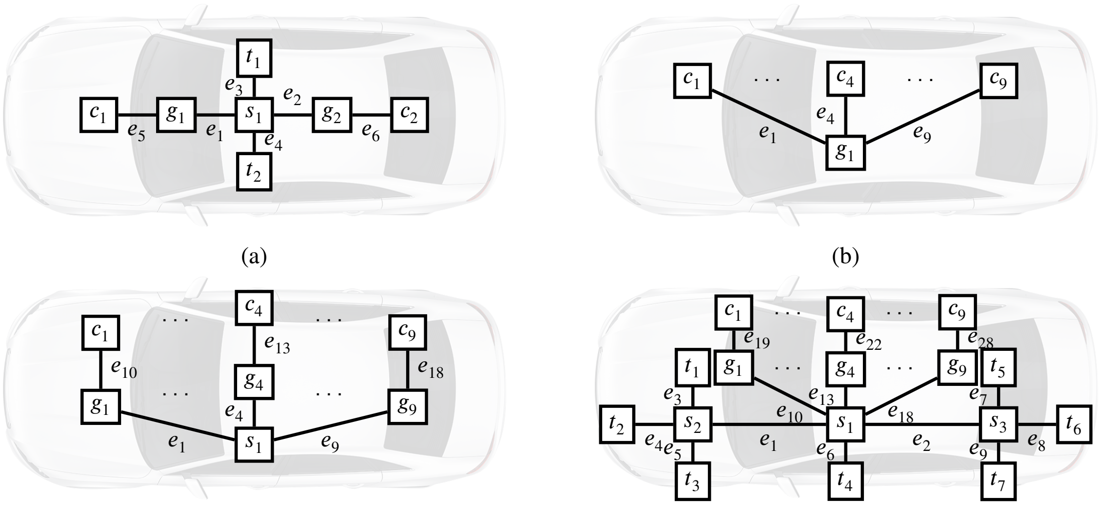


(a)

(b)

(c)

Figure 7: Optimisation results: (a) number of added nodes and links after transformation into G_o, (b) comparison with the theoretical minimum number of links in an identifiable topology, and (c) ratio of additional weights to the original topology weight.



(a)

(b)

(c)

(d)

Figure 8: In-vehicle network topologies before transformation. (a) simple in-vehicle network, (b) in-vehicle network topology with central-gateway, (c) topology with central Ethernet switch, and (d) Ethernet backbone within the RECBAR car [33]. c_i represents CAN node(s) and t_i represents Ethernet node(s).

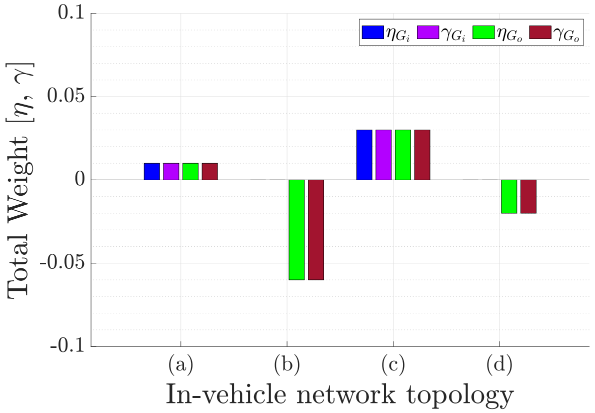


Figure 9: Results of proposed algorithms applied to real in-vehicle network topologies.

could be increased too. This is specifically clear in topologies (a) and (c). On the other hand for G_o, the number of monitors usually decreases as shown for topology (d) where the number of monitors was decreased by 2.

Let Δ and Δ_m be the total number of normal messages and the total number of monitoring messages. Then we can see that the number of monitoring messages is constant using NT as depicted in Figure 10(b) (the results are for the topology (d)). This is because NT only needs κ number of messages which is equivalent to $\binom{|\mathcal{E}|}{2}$, hence it only depends on the number of edge nodes in the topology and not the number of unique message IDs. For OTIDS and COIDS they use all the unique message IDs in the monitoring with OTIDS doubling this as it sends both request and reply messages of each ID. Thus, the number of monitoring messages in OTIDS and COIDS relies on the total number of unique messages in the network and is not affected by the topology.

Note that OTIDS and COIDS do not monitor the internal network as NT. Therefore, we cannot compare the number of identifiable links using NT with these approaches. However, we can compare the full network tomography (FNT) approach proposed in this paper with the partial network tomography (PNT) proposed in [26]. The results are shown in Table 2 where the ratio of the uniquely identifiable links. For all the topologies, FNT can uniquely identify all the links. For topology (b), PNT is not needed as the topology is already identifiable so even using PNT all links will be identifiable. However, for other topologies, a small ratio of links can be uniquely identified using PNT except for topology (c) where no links can be identified. This is because the topology is unidentifiable with all nodes in $\mathcal{R}$ having degree < 3.

These results show that a fully identifiable topology is needed to monitor the overall network performance including the internal elements. This is important to identify any failure or risks that occur to the network, especially to the internal elements that are hard to directly monitor.

Table 2
Ratio of uniquely identifiable links using Partial Network Tomography (PNT) and Full Network Tomography (FNT).

Topology	PNT	FNT
(a)	0.25	1.00
(b)	1.00	1.00
(c)	0.00	1.00
(d)	0.33	1.00

8. Conclusion

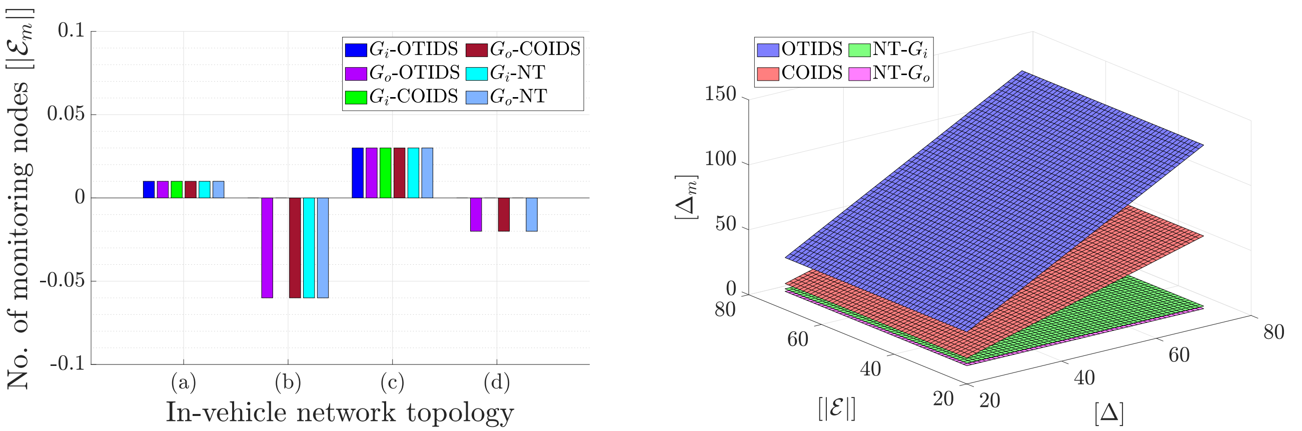
Network tomography has proven to be successful for monitoring different types of networks, including in-vehicle networks. However, the applicability of network tomography requires a fully identifiable topology under the existing monitor placement constraint. For in-vehicle networks, such constraint is translated into having only edge nodes being able to monitor the network, while OEMs often restrict access to the (internal) network devices they provide. Under this constraint, this work extensively studied the identifiability problem of in-vehicle networks and analysed the conditions required to achieve a fully identifiable topology. Furthermore, based on the derived theoretical results, we proposed a transformation algorithm that transforms any given topology into an identifiable topology where monitoring nodes are the edge nodes. The resulting topology, however, is not guaranteed to be optimal (with a minimum number of links). For this, we further proposed another transformation algorithm that transforms the identifiable topology into optimal with a minimum number of links while maintaining the identifiability property. Evaluation results on both random and real in-vehicle network topologies showed the effectiveness of such algorithms with minimal added weight, better monitoring overhead, and full identifiability ratio, as compared with other monitoring solutions.

Moreover, these algorithms can support the gradual transformation of existing in-vehicle network topologies where designing new topologies from scratch can be extremely costly. The focus of the current work was to establish in-vehicle network topologies that can benefit from network tomography in monitoring the overall network without the need to access the internal networking elements. However, the resulting, identifiable topologies are not robust enough against malicious behaviour. In other words, if a link within the network has failed, due to an attack for instance, the network cannot react to reroute the traffic passing through such link due to the lack of redundancy support. Therefore, one of our future works is to enhance the topology with redundancy capabilities so that the network becomes fail-operational.

References

[1] R. Castro, M. Coates, G. Liang, R. Nowak, B. Yu, Network tomography: Recent developments, Statistical science (2004).

[2] E. Lawrence, G. Michailidis, V. N. Nair, B. Xi, Network tomography: A review and recent developments, Frontiers in statistics (2006).



(a)

(b)

Figure 10: Comparing OTIDS and COIDS with NT (Network Tomography): (a) number of nodes participating in the monitoring process, and (b) number of monitoring messages.

[3] L. Ma, T. He, K. K. Leung, D. Towsley, A. Swami, Efficient identification of additive link metrics via network tomography, in: 2013 IEEE 33rd International Conference on Distributed Computing Systems, IEEE, 2013, pp. 581–590.

[4] L. Ma, T. He, K. K. Leung, A. Swami, D. Towsley, Identifiability of link metrics based on end-to-end path measurements, in: Proceedings of the 2013 conference on Internet measurement conference, 2013, pp. 391–404.

[5] A. Ibraheem, Z. Sheng, G. Parisis, D. Tian, In-vehicle network delay tomography, in: GLOBECOM 2022-2022 IEEE Global Communications Conference, 2022, pp. 5528–5533.

[6] S. Saidi, S. Steinhorst, A. Hamann, D. Ziegenbein, M. Wolf, Special session: Future automotive systems design: Research challenges and opportunities, in: 2018 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ ISSS), IEEE, 2018, pp. 1–7.

[7] H. Zhu, W. Zhou, Z. Li, L. Li, T. Huang, Requirements-Driven Automotive Electrical/Electronic Architecture: A Survey and Prospective Trends, IEEE Access (2021).

[8] S. Brunner, J. Roder, M. Kucera, T. Waas, Automotive E/E-architecture enhancements by usage of ethernet TSN, in: 2017 13th Workshop on Intelligent Solutions in Embedded Systems (WISES), IEEE, 2017, pp. 9–13.

[9] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, S. Uhlig, Software-defined networking: A comprehensive survey, Proceedings of the IEEE (2014).

[10] F. Rehm, J. Seitter, J.-P. Larsson, S. Saidi, G. Stea, R. Zippo, D. Ziegenbein, M. Andreozzi, A. Hamann, The road towards predictable automotive high-performance platforms, in: 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2021, pp. 1915–1924.

[11] M. Ring, D. Frkat, M. Schmiedecker, Cybersecurity evaluation of automotive e/e architectures, in: ACM Computer Science In Cars Symposium (CSCS 2018), volume 92, 2018.

[12] H. Lee, S. H. Jeong, H. K. Kim, Otids: A novel intrusion detection system for in-vehicle network by using remote frame, in: 2017 15th Annual Conference on Privacy, Security and Trust (PST), IEEE, 2017, pp. 57–5709.

[13] S. Halder, M. Conti, S. K. Das, Coids: A clock offset based intrusion detection system for controller area networks, in: Proceedings of the 21st International Conference on Distributed Computing and Networking, 2020, pp. 1–10.

[14] K. Huang, Q. Zhang, C. Zhou, N. Xiong, Y. Qin, An efficient intrusion detection approach for visual sensor networks based on traffic pattern learning, IEEE Transactions on Systems, Man, and Cybernetics: Systems 47 (2017) 2704–2713.

[15] M. Basseville, I. V. Nikiforov, et al., Detection of abrupt changes: theory and application, volume 104, prentice Hall Englewood Cliffs, 1993.

[16] Z. Deng, Y. Xun, J. Liu, S. Li, Y. Zhao, A novel intrusion detection system for next generation in-vehicle networks, in: GLOBECOM 2022-2022 IEEE Global Communications Conference, IEEE, 2022, pp. 2098–2103.

[17] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, M. Kloft, Deep one-class classification, in: International conference on machine learning, PMLR, 2018, pp. 4393–4402.

[18] H. M. Song, J. Woo, H. K. Kim, In-vehicle network intrusion detection using deep convolutional neural network, Vehicular Communications (2020).

[19] S. Jeong, B. Jeon, B. Chung, H. K. Kim, Convolutional neural network-based intrusion detection system for avtp streams in automotive ethernet-based networks, Vehicular Communications (2021).

[20] H. M. Song, H. K. Kim, Self-supervised anomaly detection for in-vehicle network using noised pseudo normal data, IEEE Transactions on Vehicular Technology (2021).

[21] S. Tariq, S. Lee, S. S. Woo, Cantransfer: Transfer learning based intrusion detection on a controller area network using convolutional lstm network, in: ACM symposium on applied computing, 2020.

[22] R. Sommer, V. Paxson, Outside the closed world: On using machine learning for network intrusion detection, in: 2010 IEEE symposium on security and privacy, IEEE, 2010, pp. 305–316.

[23] Y. Vardi, Network tomography: Estimating source-destination traffic intensities from link data, Journal of the American statistical association (1996).

[24] H.-H. Zhao, M. Chen, Topology inference based on network tomography, Journal of Software (2010).

[25] V. N. Padmanabhan, L. Qiu, Network tomography using passive end-to-end measurements, in: DIMACS Workshop on Internet and WWW Measurement, Mapping and Modeling, Citeseer, 2002.

[26] A. Ibraheem, Z. Sheng, G. Parisis, D. Tian, Neural network based partial tomography for in-vehicle network monitoring, in: 2021 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, 2021, pp. 1–6.

[27] A. Ibraheem, Z. Sheng, G. Parisis, D. Tian, Network tomography-based anomaly detection and localisation in centralised in-vehicle network, in: 2023 IEEE International Conference on Omni-layer Intelligent Systems (COINS), IEEE, 2023, pp. 1–6.

[28] R. Diestel, Graph theory, Grad. Texts in Math (2005).

[29] M. Traub, A. Maier, K. L. Barbehön, Future automotive architecture and the impact of it trends, IEEE Software (2017).

[30] L. L. Bello, R. Mariani, S. Mubeen, S. Saponara, Recent advances and trends in on-board embedded and networked automotive systems, IEEE Transactions on Industrial Informatics (2018).

[31] M. Haeberle, F. Heimgaertner, H. Loehr, N. Nayak, D. Grewe, S. Schildt, M. Menth, Softwarization of automotive e/e architectures: A software-defined networking approach, in: 2020 IEEE VNC, ????

[32] P. Weiss, A. Weichslgartner, F. Reimann, S. Steinhorst, Fail-operational automotive software design using agent-based graceful degradation, in: 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2020, pp. 1169–1174.

[33] T. Steinbach, K. Müller, F. Korf, R. Röllig, Real-time ethernet in-car backbones: First insights into an automotive prototype, in: 2014 IEEE Vehicular Networking Conference (VNC), IEEE, 2014, pp. 133–134.