Topology Optimization of In-vehicle Multimedia Communication Systems

Jörg Sommer and Elias A. Doumith
University of Stuttgart, Institute of Communication Networks and Computer Engineering
Pfaffenwaldring 47
70569 Stuttgart, Germany
{joerg.sommer, elias.doumith}@ikr.uni-stuttgart.de

ABSTRACT

In recent years, the importance of in-vehicle multimedia and infotainment systems has grown rapidly. Nowadays, the *Media Oriented Systems Transport* (MOST) network technology is predominantly used to deploy such systems. Recently, researchers started to investigate whether Ethernet-based systems can also fulfill the requirements of an in-vehicle multimedia network technology. However, cabling an Ethernet network has more degree of freedom than a ring network such as MOST since the use of additional switches can lead to different network topologies such as a star or a tree.

In this paper, we introduce and investigate different approaches to minimize the total cable length and the number of switches of different Ethernet topologies within an in-vehicle multimedia communication system. In order to compare both Ethernet and MOST technology, we develop exact and heuristic methods to optimize the total length of the cables required to connect all the electronic devices. Finally, we discuss the results for various scenarios and topologies. Our results indicate that the total cable length is not a limitation for use of Ethernet.

Keywords

Optimization, Integer linear programming, Heuristic, Simulated annealing, Topology design, Ethernet, In-vehicle communication systems

1. INTRODUCTION

In recent years, the importance of in-vehicle multimedia and infotainment systems has grown rapidly as automobile manufacturers keep adding new applications and functions. For instance, navigation systems assist the driver, while infotainment systems entertain the passengers during the journey. Electronic devices such as Dolby surround sound systems make car driving more comfortable. Future in-vehicle multimedia applications will play videos in high-definition quality and will require high data rate.

Nowadays, the *Media Oriented Systems Transport* (MOST) network technology is predominantly used to deploy in-vehicle multimedia and infotainment systems where a CD-/DVD-player, a TV receiver, a navigation system, a mobile phone interface device, an iPod interface device, and an audio amplifier are connected in a modular manner. Defined in 1998 by a consortium of automobile manufacturers and component suppliers, MOST is a serial communication system for multimedia, control, and burst-like data transmission via optical fiber bearer such as polymeric fibers. Such a MOST system uses a unidirectional ring topology and can support data rates up to 22.6 Mbps. The capacity of the existing MOST technology is reaching its limits. Therefore, considerable efforts are spent to double (or even increase by a factor of six) the offered bandwidth by introducing appropriate technology extensions [5, 8]. In addition, MOST is automotive- and manufacturer-specific and has become inflexible, complex and costly [12].

On the other hand, Ethernet is becoming more and more attractive. Several reasons (e.g. low costs and high data rates) lead to the discussion [6, 11] whether an Ethernet-based system can fulfill the requirements of an in-vehicle multimedia network technology. Over the last thirty years, Ethernet has become the predominant *Local Area Network* (LAN) technology. Ethernet's success story leads to a wider operational area and it becomes a *de-facto* standard for a large range of network infrastructure. Ethernet is already applied in fields with real-time requirements such as industrial environments, aircrafts, and in the digital home. The different applications of Ethernet are normalized by the *Institute of Electrical and Electronics Engineers* (IEEE) like norm IEEE 802.3. This norm has been adopted by the *International Organization for Standardization* (ISO) and published as an ISO standard 8802-3. The 802.3 family of standards offers compatible and upgradeable Ethernet solutions for a wide range of data rates up to 10 Gbps and is expected to be extended to rates beyond that in future standard revisions. As a mass-market product, Ethernet components are cheap and the range of products is wide. Several solutions that integrate Ethernet and higher-layer protocols such as TCP/IP on a single-chip are developed [3]. With the integration of common transport protocols, it is possible to use all the conventional multimedia services.

Typical modern Ethernet networks are deployed using multiport bridges, called *switches*, and full-duplex connections. The cabling layout within an Ethernet network has more degree of freedom than in a ring network such as MOST. This is mainly due to the use of additional switches.

ISVCS 2008, July 22 - 24, 2008, Dublin, Ireland

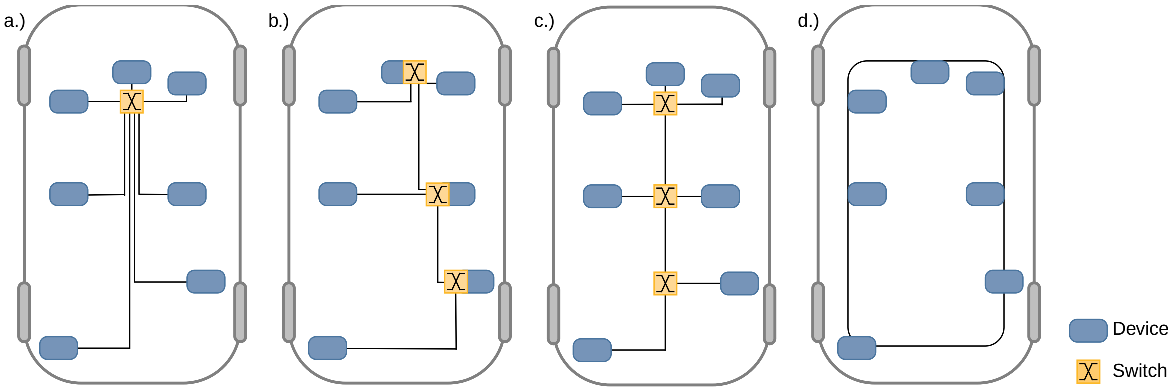

Figure 1: Different topologies: a.) Star, b.) Tree with embedded switches, c.) Tree with arbitrary switch positions, and d.) Ring.

The number of switches used, their position, and the connectivity between them have to be optimized. Due to space limitation forced by the chassis design, the wiring harness and the number of switches has to be minimized. However, these two objectives cannot be fulfilled simultaneously. For instance, one can choose to use a single switch. In this case, all the electronic devices are connected directly to this central switch. Such a star topology would lead to cable bundles and to installation space problems. Conversely, using many switches will lead to an additional weight and an increase in the cost of network deployment without any observable benefit for the customer. As a result, the design of an Ethernet network for an in-vehicle environment is a trade-off between the number of switches used and the length of the cables required to connect all the devices and switches.

In this paper, we introduce and investigate different approaches to optimize the total cable length of different Ethernet topologies within an in-vehicle multimedia communication system. The paper is structured as follows: First, we provide in section 2 a detailed description of the problem and we formulate it mathematically. Then, in order to compare both Ethernet and MOST technologies, we introduce in section 3 exact and heuristic methods to optimize the length of the cables required to connect all the electronic devices. In section 4, we compare these approaches and the resulting topologies. We conclude the paper in section 5 and provide outlines for future tracks.

2. PROBLEM DESCRIPTION

2.1 Network Topologies

In this paper, we propose several approaches to design an Ethernet-based in-vehicle multimedia communication system without redundant paths. The resulting topologies can guarantee a reduced network deployment cost but are unable to recover from a cable cut or a switch failure. As mentioned before, an Ethernet network has more degree of freedom than a ring network. In fact, in a ring network, every node/device has to be connected to two other nodes such that the overall cost of the network is minimized. An example of this connectivity is shown in Figure 1 d.). However, in a microsegmented Ethernet network, the nodes are only connected together through switches via point-to-point links. The use of additional switches give rise to many questions such as the optimal number of switches, their optimal position, and the connectivity between them. For instance, one can choose to use a single switch (*c.f.* Figure 1 a.)) or several switches. In the latter case, the switches can be located anywhere inside the vehicle (*c.f.* Figure 1 c.)) or they have to be embedded within the nodes/devices (*c.f.* Figure 1 b.)). In the former case, additional cabling for the power supply of the switches is needed. Furthermore, when we use more than three switches, these switches can be interconnected using a bus, star, or tree topology.

It is to be noted that the resulting problem is harder to solve than traditional optimization problem. In the latter, the position of the switches is given and the connectivity between them has to be optimized. In our case, we optimize the switches' position in order to achieve the minimal total cable length.

2.2 Cost Modeling

For all these scenarios, the overall cost $\mathcal{C}$ of an Ethernet network without redundant paths is formulated as

$$\mathcal{C} = 2 \times (n+m-1) \times C_{\text{Endpoint}} + m \times C_{\text{Switch}} + \Gamma \times C_{\text{Cable}} \quad (1)$$

where

- n is the number of nodes and consequently the number of links used to connect the nodes to the switches,
- m is the number of switches used,
- $m - 1$ is the number of links used to connect the switches between them,
- C_{Endpoint} is the cost of an endpoint. Such an endpoint consists of a physical layer device and a *Medium Access Control* (MAC) unit. Due to the microsegmentation of the network, there exist only point-to-point links. Therefore, we have to install an endpoint at each end of a link which explains the factor two in the cost function,
- C_{Switch} is the cost of a switching fabric (e.g. a relay unit). This cost does not include the costs of the physical layer devices and the MAC units,
- Γ is the total cable length, and
- C_{Cable} is the cost of a cable of unit length.

We note that this cost function depends on the number of switches m and the total cable length Γ. In this paper, we assume a fixed number of switches and we compute their optimal position and connectivity such that the total cable length Γ is minimized. This total cable length includes the length of the cables used to connect the nodes to the switches as well as the length of the cables used to connect the switches between them. Since there is no optimization potential for the cabling through the passenger cabin, we assume that the car space is reduced to a 2-dimensional plane where the distance $d_{i,j}$ between a node or switch i at position (x_i, y_i) and a node or switch j at position (x_j, y_j) can be computed as follows:

$$d_{i,j} = |x_i - x_j| + |y_i - y_j| \tag{2}$$

This distance measured along axes at right angles is also known as the *Manhattan distance* [9]. It is suitable to be used in in-vehicle communication networks due to the wiring harness in such environment.

2.3 Problem Formulation

To sum up, for a given set of n nodes $N_i(x_i, y_i)$ $(i = 1, \ldots, n)$, we have to locate m switches $S_j(\alpha_j, \beta_j)$ $(j = 1, \ldots, m)$, connect the nodes to the switches, and connect the switches between them while minimizing the total cable length Γ. For an instance of such problem, we define the following variables:

- $\underline{\Psi} = \{\psi_{i,j}\}$ is a $[n \times m]$ matrix representing the connectivity between the nodes and the switches where $\psi_{i,j}$ is a binary variable specifying the presence or the absence of a cable between node N_i and switch S_j.

$$\psi_{i,j} = \begin{cases} 1 & \text{if } N_i \text{ is connected to } S_j, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

- $\underline{\Phi} = \{\phi_{i,j}\}$ is a $[m \times m]$ matrix representing the connectivity between the switches where $\phi_{i,j}$ is a binary variable specifying the presence or the absence of a cable between switch S_i and switch S_j. Note that in case of full-duplex links, the matrix $\underline{\Phi}$ is symmetric $(\phi_{i,j} = \phi_{j,i})$.

$$\phi_{i,j} = \begin{cases} 1 & \text{if } S_i \text{ is connected to } S_j, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

- $\underline{\Theta} = \{\theta_{i,j}\}$ is a $[n \times m]$ matrix containing the Manhattan distances between the nodes and the switches where $\theta_{i,j}$ is a non-negative real variable equal to the length of the cable between node N_i and switch S_j (if such a cable exists).

$$\theta_{i,j} = \begin{cases} |x_i - \alpha_j| + |y_i - \beta_j| & \text{if } \psi_{i,j} = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

- $\underline{\Upsilon} = \{\upsilon_{i,j}\}$ is a $[m \times m]$ matrix containing the distances between the switches where $\upsilon_{i,j}$ is a non-negative real variable equal to the length of the cable between switches S_i and S_j (if such a cable exists).

$$\upsilon_{i,j} = \begin{cases} |\alpha_i - \alpha_j| + |\beta_i - \beta_j| & \text{if } \phi_{i,j} = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Consequently, the total cable length Γ can be computed as

$$\Gamma = \sum_{\substack{i=1,\ldots,n \\ j=1,\ldots,m}} \theta_{i,j} + \sum_{\substack{i=1,\ldots,m-1 \\ j=i+1,\ldots,m}} \upsilon_{i,j} \tag{7}$$

3. KEYS FOR THE SOLUTION

Whether we use one or multiple switches and whether the switches are embedded within the nodes or are placed anywhere inside the car, different approaches can be developed for each of these scenarios. For the simple case of a single switch, the optimal position of the switch can be determined manually. In section 3.1, we proof mathematically that this position is able to guarantee the shortest cable length. However, a *Minimum Spanning Tree* (MST)-based approach (section 3.2) is more suitable to the case where the number of switches is not limited, but their position is restricted to the position of the nodes. Finally, we propose an exact approach (section 3.3) and a heuristic approach (section 3.4 and 3.5) to deal with the general case where the switches can be located arbitrarily within the car. All these approaches are compared with the case of a ring topology. For this task, the minimal total cable length is computed using an exact approach (section 3.6).

3.1 Locate 1 Switch

When we have to locate a single switch $S(\alpha, \beta)$, all the nodes must be connected to this switch, and the global problem is reduced to the problem of finding the optimal position of this switch. Furthermore, note that the cable length along the *x-axis* is independent of the cable length along the *y-axis*. Therefore, the problem can be divided into two folds which are solved separably. In the following and without loss of generality, we focus, for example, on minimizing the cable length along the *x-axis*. For this reason, we define the ordered set U of the abscissae of the nodes. The set $U = \{u_1, u_2, \ldots, u_n\}$ can be characterized by:

- $u_1 \leqslant u_2 \leqslant \ldots \leqslant u_n$
- $\forall u_i \in U, \exists$ a node $N_{k_i}(x_{k_i}, y_{k_i}) \backslash u_i = x_{k_i}$

The total cable length Γ_X needed to connect all the nodes to the central switch along the *x-axis* can be expressed as

$$\Gamma_X = \sum_{i=1}^{n} |\alpha - x_i| \tag{8}$$

or equivalently as

$$\Gamma_X = \sum_{i=1}^{n} (\alpha - u_i) \times \text{sgn}(\alpha - u_i) \tag{9}$$

where sgn() is the *sign function* often known as the *signum function*.

Γ_X is a continuous function that reaches its minimum when $\partial\Gamma_X / \partial\alpha = 0$. In our case, the analytical expression of this derivative is given by:

$$\frac{\partial \Gamma_X}{\partial \alpha} = \sum_{i=1}^{n} \text{sgn}(\alpha - u_i) + 2 \times \sum_{i=1}^{n} (\alpha - u_i) \times \delta(\alpha - u_i) \tag{10}$$

where $\delta()$ is the *impulse function* often known as the *Dirac delta function*.

Two cases are to be considered:

- 1^{st} **case:** $\forall i \in \{1, \ldots, n\},\ \alpha \neq u_i$
  In this case, the position of the switch S along the *x-axis* is located between the abscissae of two nodes, for instance $u_k < \alpha < u_{k+1}$. Furthermore, $\delta(\alpha - u_i) = 0,\ \forall i = 1, \ldots, n$. Consequently, the derivative of the total cable length is reduced to:

$$\begin{aligned} \frac{\partial \Gamma_X}{\partial \alpha} &= \sum_{i=1}^{k} \operatorname{sgn}(\alpha - u_i) + \sum_{i=k+1}^{n} \operatorname{sgn}(\alpha - u_i) \\ &= \sum_{i=1}^{k} (+1) + \sum_{i=k+1}^{n} (-1) \\ &= 2 \times k - n \end{aligned} \tag{11}$$

  We conclude that the total cable length Γ_X reaches a minimum when the number of nodes is even and the switch is located anywhere between the two central nodes ($u_{\lceil n/2 \rceil} < \alpha < u_{\lceil n/2 \rceil + 1}$). This shortest cable length is equal to:

$$\widehat{\Gamma}_X = \sum_{i=\lceil n/2 \rceil + 1}^{n} u_i - \sum_{i=1}^{\lceil n/2 \rceil} u_i \tag{12}$$

  By continuity, we can show that this value can be also obtained for $\alpha = u_{\lceil n/2 \rceil}$ and for $\alpha = u_{\lceil n/2 \rceil + 1}$.

- 2^{nd} **case:** $\exists\ k \in \{1, \ldots, n\}$ such that $\alpha = u_k$
  In this case, the position of the switch S along the *x-axis* has the same abscissa as a node, for instance $\alpha = u_k$. Furthermore, we have $\delta(\alpha - u_i) = 0,\ \forall i = 1, \ldots, n$ and $i \neq k$, while $\alpha - u_i = 0$, for $i = k$. Consequently, the derivative of the total cable length is reduced to:

$$\begin{aligned} \frac{\partial \Gamma_X}{\partial \alpha} &= \sum_{i=1}^{k-1} \operatorname{sgn}(\alpha - u_i) + \operatorname{sgn}(\alpha - u_k) \\ &+ \sum_{i=k+1}^{n} \operatorname{sgn}(\alpha - u_i) \\ &= \sum_{i=1}^{k-1} (+1) + 0 + \sum_{i=k+1}^{n} (-1) \\ &= 2 \times k - n - 1 \end{aligned} \tag{13}$$

  We conclude that the total cable length Γ_X reaches a minimum when the number of nodes is odd and the switch is located at the same abscissa as the central node ($\alpha = u_{\lceil (n+1)/2 \rceil}$). This shortest cable length is equal to:

$$\widehat{\Gamma}_X = \sum_{i=\lceil (n+1)/2 \rceil + 1}^{n} u_i - \sum_{i=1}^{\lceil (n+1)/2 \rceil - 1} u_i \tag{14}$$

To sum up, the optimal position of the switch S along the *x-axis* is at the same abscissa as the central node when the number of nodes is odd, and anywhere between the two central nodes (including these two positions) when the number of nodes is even. By analogy, a similar conclusion can be driven for the optimal position of the switch S along the *y-axis*.

Algorithm 1 Prim's algorithm

```
procedure MST($N$)
    $\mathcal{T} \leftarrow \emptyset$;                                   ▷ $\mathcal{T}$ forms the MST
    $M \leftarrow \{r\}$                                                  ▷ $r \in N$
    while $M \neq N$ do              ▷ Not all nodes connected to $\mathcal{T}$
        $\min\{d_{m,n}\}$ such that $m \in M$ and $n \in N \setminus M$;
        $\mathcal{T} \leftarrow \mathcal{T} \cup \{(m, n)\}$;
        $M \leftarrow M \cup \{n\}$;
    end while
    return $\mathcal{T}$
end procedure
```

3.2 Minimum Spanning Tree

As mentioned previously, the installation space inside a vehicle is limited by the chassis design. It may happen that additional network components such as switches cannot be placed arbitrarily within the car. An alternative solution could be to embed these switches within the nodes. In this case, the global problem is reduced to the problem of connecting the switches between them. This can be achieved using a *Minimum Spanning Tree* (MST)-based approach. The algorithm starts by assuming that every node has an embedded switch. Once the position of the switches is known, an MST algorithm such as *Prim* [10] or *Kruskal* [7] can be used to connect the switches between them. The result of the MST is a tree-like topology connecting all the switches with the smallest total cable length. Finally, some switches can be removed from the nodes that are connected to only one other node. This solution is able to avoid cable bundles, but is unable to limit the number of switches used. As it is shown, the number of switches is only determined afterwards. In our simulations, we make use of the MST introduced in [10]. The input of the algorithm is a set of n nodes $N_i(x_i, y_i)$ $(i = 1, \ldots, n)$, denoted N. We recall that $d_{i,j}$ is the Manhattan distance between node i and node j. The pseudo-code of this MST is given by Algorithm 1.

3.3 Exact Approach: Mixed-Integer Linear Program

When the switches can be located arbitrarily inside the vehicle, the position of these switches represent additional degrees of freedom. For this general problem, we introduce in this section an exact approach to find the optimal position of the switches that guarantees the minimal total cable length. It is to be noted that the global problem is not linear since the objective function is not linear. However, by means of additional variables and appropriate formulation, the problem can be linearized and solved by means of linear solvers such as Scip [13]. The linear version of the problem is formulated as

Given

- The position of n nodes $N_i(x_i, y_i),\ i = 1, \ldots, n$.
- The number m of the switches to be located.
- The parameter $\aleph$ defined as follows:

$$1 \lll \aleph \tag{15}$$

Variables

- The abscissae α_j ($\alpha_j \in \mathbb{R}^+$) and the ordinates β_j ($\beta_j \in \mathbb{R}^+$) of the switches ($j = 1, \ldots, m$).
- The binary variables $\xi^i_{j,k}$ ($i = 2, \ldots, m,\ j, k = 1, \ldots, m$). $\xi^i_{j,k} = 1$, if the cable between the switch S_j and the switch S_k is used when sending data from the switch S_1 to the switch S_i. $\xi^i_{j,k} = 0$, otherwise.
- The binary variables $\phi_{i,j}$ ($i, j = 1, \ldots, m$) representing the connectivity between the switches (*c.f.* Equation 4).
- The non-negative real variables $\Delta^X_{i,j}$ and $\Delta^Y_{i,j}$ ($i = 1, \ldots, m-1,\ j = i+1, \ldots, m$) containing the distances between the switches along the *x-axis*and the *y-axis*, respectively.
- The non-negative real variables $\upsilon_{i,j}$ ($i = 1, \ldots, m-1,\ j = i+1, \ldots, m$) containing the length of the cables between the switches (*c.f.* Equation 6).
- The binary variables $\psi_{i,j}$ ($i = 1, \ldots, n,\ j = 1, \ldots, m$) representing the connectivity between the nodes and the switches (*c.f.* Equation 3).
- The non-negative real variables $\Lambda^X_{i,j}$ and $\Lambda^Y_{i,j}$ ($i = 1, \ldots, n,\ j = 1, \ldots, m$) containing the distances between the nodes and the switches along the *x-axis* and the *y-axis*, respectively.
- The non-negative real variables $\theta_{i,j}$ ($i = 1, \ldots, n,\ j = 1, \ldots, m$) containing the length of the cables between the nodes and the switches (*c.f.* Equation 5).

Constraints

- The interconnection between the switches is based on the principle of *Flow Conservation Constraint* commonly used in the field of *Network Dimensioning.* The basic idea is pretty simple: We assume that every node has to send some data to all the other nodes. For a given data flow between a source switch and a destination switch, the flow conservation constraint states that the data at the source switch is only entering this switch, while the data at the destination switch is only leaving this switch. At all the other switches, the amount of data entering the switch is equal to the amount of data leaving this same switch. It is obvious that if all the switches are able to exchange data with the switch S_1, then any two switches can exchange data between them. Consequently and without loss of generality, we can reduce the number of constraints to only insure that node S_1 is able to send data to all the other nodes. This can be formulated mathematically as follows: $\forall i = 2, \ldots, m$

$$\sum_{\substack{j=1,\ldots,m \\ j \neq k}} \xi^i_{j,k} - \sum_{\substack{j=1,\ldots,m \\ j \neq k}} \xi^i_{k,j} = \begin{cases} 0 & \text{if } k = 2, \ldots, m \\ & \text{and } k \neq i, \\ 1 & \text{if } k = i, \\ -1 & \text{if } k = 1. \end{cases} \tag{16}$$

- The connectivity between the switches must contain all the cables needed to connect any two switches. Therefore, $\forall i = 2, \ldots, m$, $\forall j = 1, \ldots, m$, and $\forall k = 1, \ldots, m$ we have:

$$\phi_{j,k} \geqslant \xi^i_{j,k} \tag{17}$$

- The connectivity between the switches is symmetric. Therefore, $\forall j = 1, \ldots, m-1$ and $\forall k = j+1, \ldots, m$ we have:

$$\phi_{j,k} = \phi_{k,j} \tag{18}$$

- The distance between the switches along the *x-axis* is given by: $\forall j = 1, \ldots, m-1$ and $\forall k = j+1, \ldots, m$

$$\Delta^X_{j,k} = |\alpha_j - \alpha_k| = \max\{\alpha_j - \alpha_k,\ \alpha_k - \alpha_j\} \tag{19}$$

In linear form, this equation is equivalent to: $\forall j = 1, \ldots, m-1$ and $\forall k = j+1, \ldots, m$

$$\Delta^X_{j,k} \geqslant \alpha_j - \alpha_k \tag{20a}$$
$$\Delta^X_{j,k} \geqslant \alpha_k - \alpha_j \tag{20b}$$

- Similarly, the distance between the switches along the *y-axis* is given by: $\forall j = 1, \ldots, m-1$ and $\forall k = j+1, \ldots, m$

$$\Delta^Y_{j,k} \geqslant \beta_j - \beta_k \tag{21a}$$
$$\Delta^Y_{j,k} \geqslant \beta_k - \beta_j \tag{21b}$$

- Consequently, the length of the cables between the switches is equal to: $\forall j = 1, \ldots, m-1$ and $\forall k = j+1, \ldots, m$

$$\begin{aligned} \upsilon_{j,k} &= \phi_{j,k} \times (\Delta^X_{j,k} + \Delta^Y_{j,k}) \\ &= \min\{\Delta^X_{j,k} + \Delta^Y_{j,k},\ \aleph \times \phi_{j,k}\} \end{aligned} \tag{22}$$

In linear form, this equation is equivalent to: $\forall j = 1, \ldots, m-1$ and $\forall k = j+1, \ldots, m$

$$\upsilon_{j,k} \leqslant \Delta^X_{j,k} + \Delta^Y_{j,k} \tag{23a}$$
$$\upsilon_{j,k} \leqslant \aleph \times \phi_{j,k} \tag{23b}$$
$$\upsilon_{j,k} \geqslant \Delta^X_{j,k} + \Delta^Y_{j,k} + \aleph \times (\phi_{j,k} - 1) \tag{23c}$$

- Every node is connected to only one switch. This can be formulated mathematically as follows: $\forall i = 1, \ldots, n$

$$\sum_{j=1}^{m} \psi_{i,j} = 1 \tag{24}$$

- The linear formulation of the distance along the *x-axis* between the nodes and the switches is given by: $\forall i = 1, \ldots, n$ and $\forall j = 1, \ldots, m$

$$\Lambda^X_{i,j} \geqslant x_i - \alpha_j \tag{25a}$$
$$\Lambda^X_{i,j} \geqslant \alpha_j - x_i \tag{25b}$$

- Similarly, the linear formulation of the distance along the *y-axis* between the nodes and the switches is given by: $\forall i = 1, \ldots, n$ and $\forall j = 1, \ldots, m$

$$\Lambda^Y_{i,j} \geqslant y_i - \beta_j \tag{26a}$$
$$\Lambda^Y_{i,j} \geqslant \beta_j - y_i \tag{26b}$$

- Consequently, the length of the cables between the nodes and the switches is equal to: $\forall i = 1, \ldots, n$ and $\forall j = 1, \ldots, m$

$$\begin{aligned} \theta_{i,j} &= \psi_{i,j} \times (\Lambda^X_{i,j} + \Lambda^Y_{i,j}) \\ &= \min\{\Lambda^X_{i,j} + \Lambda^Y_{i,j},\ \aleph \times \psi_{i,j}\} \end{aligned} \tag{27}$$

In linear form, this equation is equivalent to: $\forall i = 1, \ldots, n$ and $\forall j = 1, \ldots, m$

$$\begin{aligned}
\theta_{i,j} &\leqslant \Lambda^X_{i,j} + \Lambda^Y_{i,j} && \text{(28a)}\\
\theta_{i,j} &\leqslant \aleph \times \psi_{i,j} && \text{(28b)}\\
\theta_{i,j} &\geqslant \Lambda^X_{i,j} + \Lambda^Y_{i,j} + \aleph \times (\psi_{i,j} - 1) && \text{(28c)}
\end{aligned}$$

Objective

Minimize the total cable length Γ already formulated by Equation 7 as

$$\Gamma = \sum_{\substack{i=1,\ldots,n \\ j=1,\ldots,m}} \theta_{i,j} + \sum_{\substack{i=1,\ldots,m-1 \\ j=i+1,\ldots,m}} \upsilon_{i,j} \tag{29}$$

When trying to locate m switches, it may happen that some switches have the same position. In this case, the use of an additional switch cannot improve the solution. Consequently, such switches can be omitted and the final solution is obtained with a reduced number of switches.

3.4 Heuristic Approach: Simulated Annealing

We have formulated the optimization problem as a *Mixed-Integer Linear Program* (MILP). For small number of nodes and small number of switches, the MILP equations can be solved and an optimal solution is obtained. The limitation of the MILP approach is that the number of variables and the number of equations increase explosively as the number of nodes and/or switches increases. The computation complexity makes it hard to obtain optimal results in reasonable computation time. However, by means of efficient heuristic algorithms, it may be possible to get some results for large network scenarios. These results are sub-optimal solutions. The longer the computation time (number of iterations), the closer the results of these heuristics to the optimal solution.

In this section, we introduce an approach based on the *Simulated Annealing* (SA) heuristic. The basic idea of SA is to find a near-optimal solution by an iterative improvement process. The SA algorithm starts by constructing an initial solution $\mathcal{T}_0$. In this initial solution $\mathcal{T}_0$, the switches are located randomly, each node is connected to its nearest switch, and the switches are connected between them using the MST algorithm. Let Γ_0 be the total cable length of this initial/current solution. By applying defined rearrangement operations also called *Perturbation* (line 17 in Algorithm 2) to the current solution, a new solution $\mathcal{T}_\mathrm{i}$ is obtained. This Perturbation function consists first in changing the position of a single switch, then it connects each node to its nearest switch, and finally it interconnects the switches between them using the MST algorithm. Let Γ_i be the total cable length of this new solution $\mathcal{T}_\mathrm{i}$. By iteratively applying different rearrangement operations, the SA algorithm searches the solution space for solution with the shortest total cable length.

New solutions which give better results (lower total cable length) than the current solution are accepted automatically (line 21 in Algorithm 2). To avoid local minima, solutions with longer total cable length than the current solutions are accepted with a certain probability determined by a system control parameter T. However, the probability with which these more expensive solutions are chosen decreases as the algorithm progresses in time so as to simulate the *cooling* process associated with annealing. This probability is based on a negative exponential factor and is inversely proportional to the difference between the cost of the current solution and the cost of the new solution (line 35 in Algorithm 2). If the cost of the new solution is equal to the cost of the current solution, one can randomly choose to accept the new solution as the current solution or rejected it (line 28 in Algorithm 2).

Algorithm 2 SA algorithm

```
 1: define
 2:     Iter          ▷ maximum number of consecutive iterations
    without any cost improvement
 3:     mAcpt                        ▷ maximum accepted counter
 4:     mAtmp                       ▷ maximum attempted counter
 5:     $T$                                 ▷ initial temperature
 6:     $p$                                      ▷ cooling factor
 7: end define
 8: procedure SA($N_i(x_i, y_i)$, $m$))
 9:     k ← 1
10:     Construct an initial solution $\mathcal{T}_0$
11:     $\Gamma_0 \leftarrow \textsc{Calc}(\mathcal{T}_0)$
12:     $(\mathcal{T}_\mathrm{best}, \Gamma_\mathrm{best}) \leftarrow (\mathcal{T}_0, \Gamma_0)$
13:     while k ≤ Iter do
14:         atmp ← 0
15:         acpt ← 0
16:         while (acpt < mAcpt) ∧ (atmp < mAtmp) do
17:             $\mathcal{T}_\mathrm{i} \leftarrow \textsc{Perturbation}(\mathcal{T}_0)$
18:             $\Gamma_\mathrm{i} \leftarrow \textsc{Calc}(\mathcal{T}_\mathrm{i})$
19:             atmp ++
20:             if $\Gamma_\mathrm{i} < \Gamma_0$ then
21:                 $(\mathcal{T}_0, \Gamma_0) \leftarrow (\mathcal{T}_\mathrm{i}, \Gamma_\mathrm{i})$
22:                 acpt ++
23:                 k ← 1
24:                 if $\Gamma_\mathrm{i} < \Gamma_\mathrm{best}$ then
25:                     $(\mathcal{T}_\mathrm{best}, \Gamma_\mathrm{best}) \leftarrow (\mathcal{T}_\mathrm{i}, \Gamma_\mathrm{i})$
26:                 end if
27:             else if $\Gamma_\mathrm{i} = \Gamma_0$ then
28:                 Generate a random integer number r
29:                 if r%2 = 0 then
30:                     $(\mathcal{T}_0, \Gamma_0) \leftarrow (\mathcal{T}_\mathrm{i}, \Gamma_\mathrm{i})$
31:                     acpt ++
32:                 end if
33:             else
34:                 Generate a random number $A \in [0, 1[$
35:                 if $A < e^{-(\Gamma_\mathrm{i} - \Gamma_0)/T}$ then
36:                     $(\mathcal{T}_0, \Gamma_0) \leftarrow (\mathcal{T}_\mathrm{i}, \Gamma_\mathrm{i})$
37:                     acpt ++
38:                 end if
39:             end if
40:         end while
41:         if (atmp = mAtmp) then
42:             $k$ ++
43:         else
44:             $k \leftarrow 1$
45:         end if
46:         $T \leftarrow T \times p$
47:     end while
48:     return $\mathcal{T}_\mathrm{best}$ and $\Gamma_\mathrm{best}$
49: end procedure
```

When trying to locate m switches, it may happen that some switches are not connected to any nodes. In this case, the use of an additional switch cannot improve the solution. Consequently, such switches can be omitted and the final solution is obtained with a reduced number of switches. During the simulation, the best solution obtained so far by the SA algorithm is stored. Let $\mathcal{T}_{\text{best}}$ be this best solution and Γ_{best} its total cable length.

A detailed description of the SA algorithm is provided by Algorithm 2. This SA algorithm was implemented using Matlab [14]. Thus, it should be noted that this implementation is not optimized in terms of computation time (CPU time).

3.5 Enhanced Heuristic

From the case of locating a single switch as well as from the results of the MILP, we can notice that the optimal position of the switches is on the grid formed by the coordinates of the nodes. In other words, if U is the ordered set of the abscissae of the nodes and V is the ordered set of the ordinates of the nodes, the optimal position of a switch $S_i(\alpha_i,\ \beta_i)$ is such that: $\alpha_i \in U$ and $\beta_i \in V$. Thus, in order to improve the results obtained by the SA algorithm, a post-processing is applied. This post-processing consists of:

- Round the position along the *x-axis* of each switch obtained at the last iteration of the SA algorithm to the nearest node's abscissa.
- Round the position along the *y-axis* of each switch obtained at the last iteration of the SA algorithm to the nearest node's ordinate.
- Recheck the connectivity between the nodes and the switches by connecting each node to the nearest switch.
- Recheck the connectivity between the switches by computing a MST along the new position of the switches.
- Finally, compute the new total cable length Γ of this new solution.

This algorithm is referred to in the following as *Enhanced Simulated Annealing* (ESA).

3.6 Exact Approach: MOST Ring

In the MOST technology, the nodes are connected using a ring topology. This can be viewed as an instance of the well-known *Traveling Salesman Problem* (TSP). The latter problem can be simply stated as: Given n cities and the cost $\gamma_{i,j}$ of traveling from any city i to any other city j, what is the least costly round-trip route that the traveling salesman can take in order to visit each city exactly once and then return back to his home city? The cost of a tour is defined as the sum of the costs of the lines in the round-trip.

The TSP problem belongs to the class of combinatorial optimization problems known as NP-complete. To date, no polynomial-time algorithm has been found for the TSP problem. The most direct solution to the problem would be to try all the permutations using a brute force search and then see which one is the best solution. Given that the number of permutations is $n!$, this solution rapidly becomes impractical. Using the techniques of dynamic programming, one can solve the problem in shorter computation time. For instance, heuristic approaches based on *Genetic Algorithms*, SA, *Neural Net*, or *Tabu Search* can be used to solve large instances of the TSP problem in very reasonable computation time. However, these strategies produce feasible solutions without any guarantee on how far they may be from the optimal solution. Exact approaches based on *Linear Programming* (LP) also exist for the TSP problem. The mathematical formulation of the LP approach must include special constraints used to prevent the formation of cycles/subtours. Examples of such formulations are the Dantzig-Fulkerson-Johnson formulation in 1954 [4] and the Miller-Tucker-Zemlin formulation in 1960 [1]. In the following, we present our own formulation of the TSP problem.

Given

- The position of n nodes $N_i(x_i, y_i),\ i = 1, \dots, n$.
- The distance $\gamma_{i,j}$ between the node N_i and the node N_j, $i = 1, \dots, n-1,\ j = i+1, \dots, n$.

Variables

- The binary variables $\zeta^i_{j,k}$ $(i = 2, \dots, n,\ j, k = 1, \dots, n)$. $\zeta^i_{j,k} = 1$, if the cable between the node N_j and the node N_k is used when sending data from the node N_1 to the node N_i. $\zeta^i_{j,k} = 0$, otherwise.
- The binary variables $\eta_{i,j}$ $(i = 1, \dots, n-1,\ j = i+1, \dots, n)$ representing the connectivity between the nodes. $\eta_{i,j} = 1$, if the node N_i is connected to the node N_j. $\eta_{i,j} = 0$, otherwise.

Constraints

- To insure the connectivity between the nodes, we assume that node N_1 must be able to send some data to all the other nodes. Similarly to Equation 16, the flow conservation constraint can be formulated mathematically as: $\forall i = 2, \dots, n$

$$\sum_{\substack{j=1,\dots,n \\ j\neq k}} \zeta^i_{j,k} - \sum_{\substack{j=1,\dots,n \\ j\neq k}} \zeta^i_{k,j} = \begin{cases} 0 & \text{if } k = 2, \dots, n \\ & \text{and } k \neq i, \\ 1 & \text{if } k = i, \\ -1 & \text{if } k = 1. \end{cases} \tag{30}$$

- The connectivity between the nodes is symmetric and must contain all the cables needed to connect any two nodes. Therefore, $\forall i = 2, \dots, n$, $\forall j = 1, \dots, n-1$, and $\forall k = j+1, \dots, n$ we have:

$$\eta_{j,k} \geqslant \zeta^i_{j,k} \tag{31a}$$

$$\eta_{j,k} \geqslant \zeta^i_{k,j} \tag{31b}$$

- Every node must be connected to exactly two other nodes. Therefore, $\forall j = 1, \dots, n$ we have:

$$\sum_{k=1}^{j-1} \eta_{k,j} + \sum_{k=j+1}^{n} \eta_{j,k} = 2 \tag{32}$$

Objective

Minimize the total cable length Γ formulated as

$$\Gamma = \sum_{\substack{i=1,\dots,n-1 \\ j=i+1,\dots,n}} \gamma_{i,j} \times \eta_{i,j} \tag{33}$$

# Nodes		1 Switch	2 Switches			3 Switches			5 Switches			MST	MOST
			MILP	SA	ESA	MILP	SA	ESA	MILP	SA	ESA		
10	Γ[1]	9.167	6.151	6.151	6.151	5.240	5.255	5.240	4.680	4.860	4.827	4.876	7.213
	t[2]	-	0.28	7.12	-	2.07	3.78	-	6 h	187.10	-	-	-
15	Γ	15.674	10.771	10.771	10.771	8.987	8.991	8.987	7.331	7.533	7.669	7.226	9.559
	t	-	1.07	8.22	-	178.68	3.96	-	115 h	74.10	-	-	-
20	Γ	22.630	15.476	15.477	15.476	12.549	12.552	12.549	9.5831	11.160	11.104	8.498	10.346
	t	-	7.73	8.95	-	1288.66	6.60	-	>12 days	123.27	-	-	-

[1] Total cable length in meters

[2] Computation time in seconds unless specified otherwise

Table 1: Comparison of the scenarios

4. EVALUATION

We introduced different approaches to optimize the total cable length for a star, tree, and a ring topology. For the tree topologies we differentiated between the case where the switches are embedded within the nodes and the case where the switches can be located arbitrarily. For all these topologies we presented exact approaches to find the minimal total cable length. Furthermore, for the tree topology with arbitrary switch positions, we introduced two heuristics based on a SA algorithm.

In order to compare the topologies as well as the approaches, we defined different scenarios. We assumed an in-vehicle multimedia and infotainment system with 10, 15, and 20 nodes. We generated the position of the nodes randomly such that $0 \leq x \leq 1.75$ and $0 \leq y \leq 3.5$ corresponding approximately to cars' dimensions.

As shown in Table 1, the star topology results in the largest total cable length in all scenarios. It is clear that deploying additional switches reduces the total cable length and avoids cable bundles. With two or three switches, the total cable length is similar to that of a MOST ring. Therefore, the total cable length itself is not an argument against the use of Ethernet in in-vehicle communication systems. Once the switches are embedded in the nodes, the MST can be used to guarantee the shortest cable length. However, it uses an excessive number of switches. For example, in our scenario with 10, 15, and 20 nodes 7, 10, and 14 switches have to be deployed, respectively.

As shown in Table 1, the computation time of the MILP algorithm increases exponentially. For instance in the case of 20 nodes and 5 switches the MILP did not find the optimal solution even after 12 days of computation time. Nevertheless, during this time a satisfactory solution was obtained.

Due to the exponential increase of the MILP's computation time, we adopted a SA approach. This approach finds solutions that are near optimal in an acceptable time. However, it may happen that the SA algorithm stucks in a local optimum and did not achieve the best solution. This happened in the case of 20 nodes and 5 switches where the SA algorithm could not achieve a near optimal solution. This effect can be avoided by increasing the number of iterations while still being much faster than the MILP approach.

Just by rounding the position of the switches to the grid passing through the nodes, the ESA algorithm is able to achieve better results than the SA algorithm. The gain brought by the ESA algorithm depends on how far from the optimum are the results of the SA algorithm.

5. CONCLUSION AND OUTLOOK

In recent years, the importance of in-vehicle multimedia and infotainment communication systems has grown rapidly. Nowadays, Ethernet is becoming more attractive. With respect to the total cable length, Ethernet can be used as an in-vehicle communication system instead of currently applied network technologies such as MOST.

The cabling layout process of an Ethernet network is more complicated since the use of switches leads to a star or tree topology, which has greater complexity than a ring topology like MOST. Since the installation space in a vehicle is limited, we introduced different approaches to minimize the total cable length and the number of switches for different topologies: Star, tree, and ring. For the tree topology, we distinguished between the case where the switches are embedded within the nodes and the case where the switches can be located arbitrarily.

For all these topologies we presented exact approaches to find the minimal total cable length. Furthermore, for the tree topology with arbitrary switch positions, we introduced two heuristics based on a SA algorithm.

As we have shown, deploying additional switches reduces the total cable length and avoids cable bundles. With two or three of switches, we can achieve total cable length similar to that of a MOST ring.

In the case of a large number of nodes and/or switches, the MILP leads to larger computation time. We have shown that our SA algorithms achieve results close to the optimum. Therefore, the SA algorithms can be used for optimizing the topology of an Ethernet-based in-vehicle multimedia communication system.

From the results of locating a single switch as well as from the results of the MILP, we observed that the optimal position of the switches is on the grid formed by the coordinates of the nodes. In the next step, we will improve the SA algorithm so that the switches' position lies on this grid.

Another way to reduce the total cable length further is to group into *Collision Domains*. A collision domain is defined as a segment where multiple nodes share the same link. In this case, we do not have a microsegmented Ethernet network with point-to-point links and we have to ensure that the links are not overloaded and that prescribed bandwidth requirements are fulfilled. These additional constraints make the problem more complex. Currently, we are working on an approach for this problem.

6. ACKNOWLEDGMENTS

The authors would like to thank Martin Köhn, Andreas Reifert, and Joachim Scharf for many fruitful discussions and Ali Daher for his contribution to the design and implementation of the SA algorithm [2].

7. REFERENCES

[1] C. E. Millera, A. W. Tucker, and R. A. Zemlin. Integer programming formulations and traveling salesman problems. *Journal of the Association for Computing Machinery*, 7:326–329, 1960.

[2] A. Daher. Topology Optimization for an Ethernet-based In-vehicle Communication System. Master's thesis, University of Stuttgart, 2008.

[3] J.-D. Decotignie. A perspective on Ethernet-TCP/IP as a fieldbus. In *Proceedings volume from the 4th IFAC Conference*, 2001.

[4] G. B. Dantzig, R. Fulkerson, and S. M. Johnson. Solution of a large-scale traveling salesman problem. *Operations Research*, 2:393–410, 1954.

[5] A. Grzemba. *MOST. Das Multimedia-Bussystem für den Einsatz im Automobil*. Franzis, 2007.

[6] J. Hillebrand, M. Rahmani, R. Bogenberger, and E. Steinbach. Coexistence of Time-Triggered and Event-Triggered Traffic in Switched Full-Duplex Ethernet Networks. *Industrial Embedded Systems, 2007. SIES '07. International Symposium on*, pages 217–224, July 2007.

[7] J. J. B. Kruskal. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, February 1956.

[8] MOST Cooperation. *MOST Specification – Version 2.5-00*, October 2006.

[9] National Institute of Standards and Technology. Definition of Manhattan distance. http://www.nist.gov, February 2008.

[10] R. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36:1389–1401, 1957.

[11] M. Rahmani, J. Hillebrand, W. Hintermaier, R. Bogenberger, and E. Steinbach. A Novel Network Architecture for In-Vehicle Audio and Video Communication. *Broadband Convergence Networks, 2007. BcN '07. 2nd IEEE/IFIP International Workshop onndreas*, pages 1–12, May 2007.

[12] M. Rahmani, R. Steffen, K. Tappayuthpijarn, E. Steinbach, and G. Giordano. Performance analysis of different network topologies for in-vehicle audio and video communication. *4th International Telecommunication Networking Workshop on QoS in Multiservice IP Networks*, pages 179–184, February 2008.

[13] Solving Constraint Integer Programs (SCIP) Website. http://scip.zib.de/scip.shtml, 2008.

[14] The MathWorks. Matlab Ver. R2007b, August 2007.