A Self-Propagating Wakeup Mechanism for Point-to-Point Networks with Partial Network Support

Jan R. Seyler, Thilo Streichert, Juri Warkentin, Matthias Spägele
Daimler AG
Mercedes-Benz Cars Development
E/E-Architectures & Standards
Email: {jan.seyler|thilo.streichert}@daimler.com

Michael Glaß, Jürgen Teich
Department of
Computer Science 12
Friedrich-Alexander-Universität Erlangen-Nürnberg
Email: {glass|teich}@informatik.uni-erlangen.de

***Abstract*—As a result of the increased demand for bandwidth, current automotive networks are getting more heterogeneous. New technologies like Ethernet as a packet-switched point-to-point network are introduced. Nevertheless, the requirements on stand-by power consumption and short activation times are still the same as for existing field buses. Ethernet does not provide wakeup mechanisms that are sufficient for automotive systems. As a remedy, this paper introduces a novel physical-layer mechanism called *Low Frequency Wakeup* that is largely independent of the communication technology and topology used. It provides parallel and remote wakeup for all nodes even in a point-to-point network as well as full support of partial networking. The overall wakeup detection time is smaller than 10 ms and every node can actively feed a wakeup signal asynchronously to all other nodes.**

In terms of latency, it is shown that Low Frequency Wakeup reaches a reduction of more than 30 % for a three-hop network and more than 50 % for a five-hop network in comparison to the current state-of-the-art technology for automotive point-to-point networks.

I. Introduction

Today's high class vehicles contain 70 - 100 electronic control units (*ECUs*) with an increasing demand for higher bandwidth in every generation. Because of this, Ethernet is discussed and developed for automotive environments [6] even though it is originally designed as a computer networking technology for local area networks. Automotive networks demand different and more strict requirements that cover topics like EMC, qualification of devices, and wakeup mechanisms. The latter are the topic of this paper. It is shown that known technologies and standards do not meet these requirements and are therefore not usable.

Current automotive network systems are in most cases based on serial bus systems with signal detection concepts to activate them: Initially, the entire bus system is inactive and an activation event on the bus activates all nodes. This event can be a data frame or a single voltage spike. Thereby, CAN [9] transceivers, like the TJA1048, have a standby power consumption of less than $400\,\mu$W and wake up in less than 5 ms. FlexRay transceivers, like the TJA1083, do have a slightly higher standby power consumption of less than $500\,\mu$W and a wakeup-time below 20 ms. In comparison, current automotive Ethernet transceivers have a power consumption of up to 250 mW and a wakeup-time of 120 ms per node in a point-to-point network.

Rising environmental awareness demands a low fuel consumption and higher CO_2 reduction. Recently, partial networking is introduced to build smart energy-efficient vehicles [9]. It provides the possibility of activating a network only partially if certain functionalities are not required in the current system state. Nodes that are inactive can be woken with a special packet. This reduces the power-consumption of the vehicle but also solves the problem that the lifetime of ECUs in electric driven cars is stressed due to long charging intervals in which the ECUs would be awake otherwise.

Point-to-point networks add a new level of complexity as they do not have a shared communication medium and, therefore, each node needs to wake its adjoined nodes and can only wake these as messages are not received in parallel by all ECUs anymore. Additionally, every node within the route between the wakeup source and sink needs to be awake so the wakeup frame can be forwarded and processed correctly. The chance of a drop of this frame within the boot-sequence of one of the intermediate nodes is another aspect that needs to be considered. As a result, wakeup frames need to be sent repeatedly until it is guaranteed that the wakeup sink has received at least one wakeup frame. This adds additional latency to the process of activating a network. The latency scales with thenetwork diameter and is not negligibly small in vehicle networks.

In summary the main problems of known mechanisms to wake up point-to-point networks are:

- Long wakeup latencies due to point-to-point delay
- High stand-by power consumption due to active communication or interpretation logic
- No partial network support as no information can be coded during wakeup

To overcome these problems, this paper proposes a novel technique termed *Low Frequency Wakeup* (*LFW*) that can activate every part of a network and also the network as a whole. It minimizes the activation delay to the scale of the physical wire propagation delay and also addresses the challenges of the quiescent current of the single controllers. Partial networking is fully supported and the additional feature of remote activation is introduced.

The rest of the paper is organized as follows: Technologies that are developed especially for network systems within motor vehicles are assessed in Section II. A detailed description of the concept and technology is given in Section III-A followed by the simulative proof of functionality and key performance indicators in Section IV. Section V gives a brief conclusion and some perspectives for future development and improvements.

II. Related Work

Mechanisms to activate networks can be split into communication-layer and physical-layer methods but do all depend on a switchable power supply. The mechanisms differ in the way they control the switch. Figure 1 displays the classification of wakeup mechanisms for Ethernet-based networks. The left side names mechanisms that control the switchable power supply independent of the actual communication.

978-3-9815370-2-4/DATE14/© 2014 EDAA

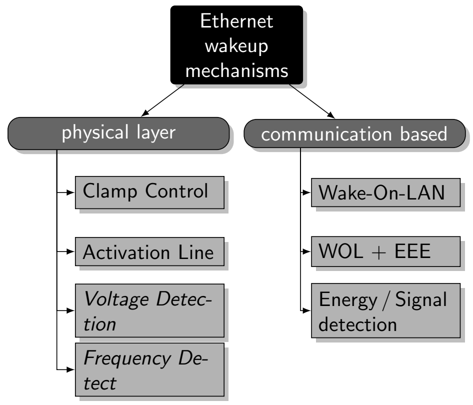

Figure 1 – Classification of wakeup mechanisms for Ethernet networks

1) Clamp Control: Prior to the interconnection of electronic control units, all ECUs that were not needed while the car is inactive had their switchable power supply connected to clamp 15 [7] which is active if and only if the ignition is active. All other ECUs are connected to clamp 30 [7] which is constantly powered. This mechanism is widely used today. But, as the number of ECUs rises, the usage of physical clamps influences the overall weight of the car which leads to a growing number of virtual clamps whose state is controlled by the software on the controller.

2) Activation Line: Fröschl [5] describes a mechanism where nodes can simulate the behaviour of clamp 15 to activate ECUs while the car itself is inactive , this characterizes a further development of the general idea of a separate activation line as defined in [12]. The wakeup-master directly controls the switchable power supply of the connected nodes by sending a single voltage pulse onto the activation line. All connected nodes need to reach the active state before the activation pulse is pulled down again as they have to activate a lock to hold the power supply active. The disadvantage of these technologies is the need for additional hardware and wires as well as a low grade of flexibility. In particular, partial networking can only be realized by using one additional wire for each subnet.

As additional wires add weight, modern cars are using communication-based wakeup mechanisms which can activate nodes over the communication wire. The transceiver itself provides the functionality to activate and deactivate the controller. A shared Network Management (*NM*) that has an instance on every ECU manages the state of the bus after wakeup which includes powering down the ECU. In packet-switched point-to-point networks, such as Ethernet, packets are explicitly addressed to a single destination or a multicast group of nodes. Either way, if the network consists of more than two nodes, every packet passes at least one intermediate node on its path to the destination. This leads to two new situations: Firstly, messages might only be received by one node. Secondly, a message needs to pass several hops till it can be seen by the destination node.

3) Wake-On-LAN: For local area networks (*LAN*s), Wake-On-LAN (*WOL*) [1] is an established mechanism. It uses a specially designed so-called *magic packet*™ which is broadcasted in the whole network and contains the unique MAC address of the destination network device. An Ethernet transceiver consists of the PHY which connects the link layer device, called a MAC, to a physical medium like a copper cable. It is responsible for encoding and decoding the transmitted and received modulated analog signal to a digital input for the MAC. This means that the PHY and MAC needs to be active all the time to perform WOL, as the connection must be kept synchronized by continuously sending idle pattern, when there is no frame being transmitted, see [11]. Additionally, the switches also need to be active to process the magic packets. Balbierer et al. [3] determined that a typical Ethernet PHY consumes 220 mW, a MAC 40 mW and a switch fabric 250 mW. The result is a power consumption of at least $220 + 40\,\mathrm{mW}$ per end-node and $(n \times (220 + 40) + 250)\,\mathrm{mW}$ for a switch with n connected ports. In comparison, a typical automotive ECU has a quiescent power consumption below $500\,\mu\mathrm{W}$. Therefore, WOL is not suitable for the automotive environment.

4) WOL + EEE: If WOL is combined with *Energy Efficient Ethernet* [4], as defined in IEEE 802.3az, the power consumption of the PHY can be reduced by 90 %. This leads to a low power idle power consumption of 22 mW for each Ethernet PHY; still one order of magnitude too high.

5) Energy / Signal detection: Balbierer et al. [2] proposed an energy detect (*ED*) module with zero standby current as one solution: If a differential voltage level is detected on the communication line, this module activates the power supply of the ECU it is installed on. Consequently, each intermediate switch between the wakeup source and destination needs to be woken prior to the actual wakeup which adds additional point-to-point latency as will be discussed in Section IV.

6) Wake on CAN over Ethernet: Rockel and Kaindl [10] describe the idea to wake 100BASE-TX Ethernet ECUs by adding a CAN transceiver to each Ethernet PHY and sending CAN wakeup frames in the unused pairs of the CAT cable. This mechanism is not suitable for automotive Ethernet, which communicates over a single twisted pair cable. The high point-to-point latency is also not solved with this technology.

As mentioned above, partial networking is developed to reduce the overall power consumption and, therefore, the CO_2 emission. It allows to deactivate unused components within a vehicle while driving. One example might be deactivating the rear-view camera when the reverse gear is not engaged. An enabler for this is a reliable and fast technology to deactivate and reactivate network components while the rest of the network stays awake. This also includes to wake specific nodes within the network. Obviously, WOL supports this feature. In the automotive environment, ISO/DIS 11898-6 [9] recently introduced partial networking for CAN buses which is already implemented in industrial products like the NXP TJA1145, cf. [8]. Mainly, nodes are now able to activate partial networks and can be woken if one of the clusters, it is part of, will be activated.

The technology introduced in this paper contributes to both the field of activating a network and reducing power consumption by introducing a physical layer mechanism that can transparently wake through intermediate nodes and enables partial networking. The mechanism is described in detail in the following.

III. Voltage and frequency detection

A careful review of current wakeup technologies for Ethernet reveals that they will not meet all automotive requirements. Each mechanism has at least one of the drawbacks that are itemized in Section I. In this section, the proposed new technique

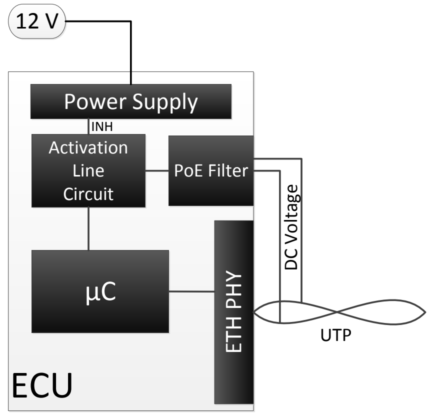

Figure 2 – Schematic of the *Signal-over-Ethernet* technique in an Ethernet end-node.

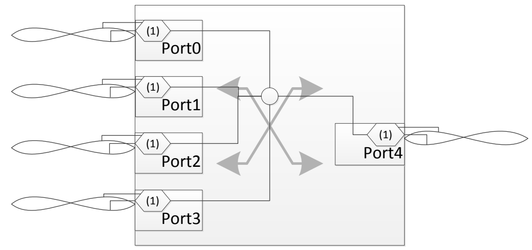

Figure 3 – Schematic of star-hub distributor for the wakeup signal within a switch.

termed *Low Frequency Wakeup* (LFW) is presented. Starting from a naive solution, several concepts and improvements are introduced and discussed. Afterwards, the concept and implementation of LFW are introduced in detail.
All mechanisms share the concept of passing on the received wakeup signal through intermediate nodes without needing to activate the communication (cf. Figure 3). This novel idea enables to wake point-to-point networks in parallel and reduce the point-to-point latency which is a result of configuration cycles. For current automotive Ethernet switches these configuration time is in the range of 100 ms, which makes it difficult to match the requirement of waking a complete automotive network within 150 ms.

a) Signal-over-Ethernet: The first idea that we examine is termed *Signal-over-Ethernet* (SoE) which is a voltage detection mechanism. It provides the functionality of an activation-line combined with Power-over-Ethernet (PoE) technology. As Figure 2 shows, the energy pulse to wake up nodes is fed directly onto the communication wire and decoupled prior to the PHY of the connected controller. The decoupled DC signal is afterwards processed as a normal activation line signal. However, for a controller, it is only possible to wake its direct neighbours and, hence, the advantages over the already known energy detect mechanisms are minor. To avoid the point-to-point latency, a remedy is to decouple the DC signal and pass it to all connected communication lines instantly without having to activate the controller first. Thus, point-to-point latency is reduced and all nodes wake up in parallel. Figure 3 shows this principle for a switch. Each network port has a module that detects the wakeup signal (1). This module decouples the signal from the communication line and instantly forwards it to all other ports where it is coupled to the communication line again. Parallel wakeup of a point-to-point network is realized this way. However, partial networking can only be realized with different voltage levels for each network cluster which leads to a high combinatorial effort. Assume m n-port switches in the network. On each switch it is possible to wake $(n-1)$ ports. Enabling to wake each port individually results in a complexity of

$$C(m,n) = m \cdot (n-1) \tag{1}$$

If it is possible to wake any possible combination of ports in the switch, the complexity is:

$$\begin{aligned} C(m,n) &= m \cdot \sum_{k=1}^{n-1} \binom{n-1}{k} \\ &= m \cdot \left(\sum_{k=0}^{n-1} \binom{n-1}{k} - 1 \right) \\ &= m \cdot \left(2^{n-1} - 1\right) \end{aligned} \tag{2}$$

Consider that the maximum $C(m,n)$ scales exponentially with the number n of ports on a switch. Thus, this mechanism is not scalable with respect to partial networking.

b) Pattern-Detection: Instead of sending a voltage pulse, another idea is to send a specific wakeup pattern whereby the main idea of passing the wakeup signal on through the switches, so it reaches each node in a point-to-point network in parallel, remains the same. As a result, partial networking is enabled. The idea to have a bit mask in all nodes where every bit represents one partial cluster. Therefore, the number of partial networks is configurable and more than one cluster can be woken at a time. This is the same principle that is used in the CAN partial network technology, where every CAN-transceiver has at least one bit mask [9]. However, this mechanism requires additional hardware and active logic to interpret the wakeup pattern, resulting in an increased standby power consumption. Additionally, all nodes need to be synchronized as it can only be allowed to send one wakeup pattern at a time. If several pattern are sent in parallel, they will interfere with each other.

Thirdly, we examined the Ethernet PHYs that are currently used in the automotive environment and found out that it is possible to transmit electrical signals with a frequency below 100 kHz without influencing the actual communication. This observation is exploited in our proposed low frequency wakeup technique presented in the following.

A. Low Frequency Wakeup

High bandwidth communication technologies use capacitors to avoid a direct current flow on the wire in the case of a ground offset. The characteristic curve of the electrical impedance of the simulative model of current capacitors for automotive usage is shown in Figure 4. The electrical impedance is plotted against the applied frequency to the capacitor. The impedance increases rapidly with decreasing frequency. Alternating current with a frequency below 10^5 Hz will not pass the capacitors. Therefore, frequencies in the band below 100 kHz do not influence the

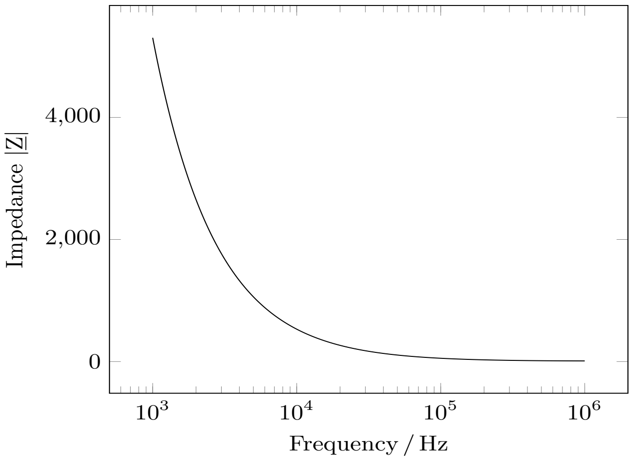

Figure 4 – The electrical impedance |Z| of a simulative model of a typical automotive capacitor (C = 10 nF, L = 8 nH, R = 700 mΩ) as currently examined for usage in automotive Ethernet PHYs

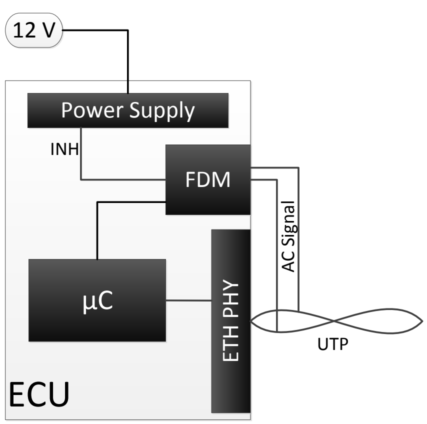

Figure 5 – Schematic of the *Low Frequency Wakeup* technique in an Ethernet end-node with the usage of a *Frequency Detection Module*.

communication. The main idea of *Low Frequency Wakeup* (*LFW*) is to use this *side channel* to transmit signals. The *Frequency Detect Module* (*FDM*) feeds this signal to the unshielded twisted pair (UTP) wire directly after the Ethernet PHY, cf. Figure 5. The FDM at the receiver decouples the signal at the same point, respectively. To avoid the usage of active transceivers to interpret the received signal as in [10], we decided to transmit pure sine waves with a fixed frequency to make the system more power efficient.

The key functionality of the mechanism is as follows: Every communication cluster has one unique frequency assigned and all nodes in the network are at least part of one cluster. Thus, each controller needs to store each frequency of all clusters it is included in. This can be done by mapping each available frequency to one bit in a bit mask that is stored in the PHY. When a wave of interfered frequencies arrives, it is split into its components. If any frequency is recognized as known, the controller will wake up. Additionally, the signal is passed to all

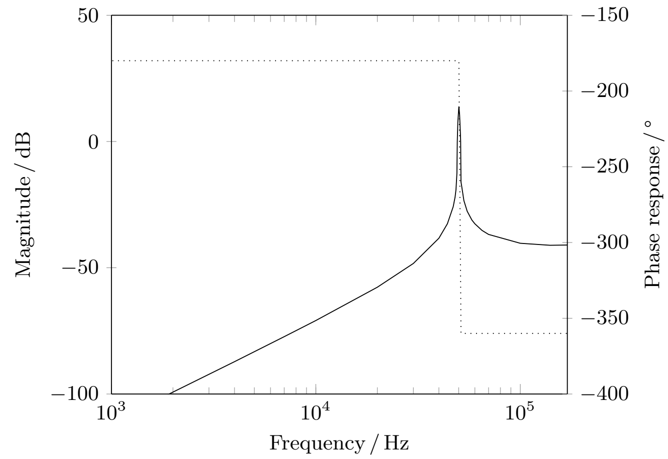

Figure 6 – Bode plot of a frequency detect module configured to 50 kHz

ports of the controller cf. Figure 3.

The generation of the sine waves with configurable frequencies up to 100 kHz can be done by the micro controller of the ECU and afterwards handed to the FDM. This is the main module of the technology which is placed at every PHY and has two main tasks: Firstly, as mentioned above, it couples the wakeup frequency to the communication line of the controller which shall wake the network. Secondly, it detects and evaluates frequencies that are received and activates the controller accordingly.

IV. Evaluation and Results

A. Functional Simulative Verification of Low Frequency Wakeup

To verify the functionality and determine the properties of LFW, an electronic circuit for a transmitter and the FDM is modelled in LTSpice IV. Figure 6 shows the frequency response of an FDM that is configured to 50 ± 0.25 kHz. The magnitude of the frequency response gain is plotted versus a logarithmic frequency axis. The FDM activates the inhibit pin if the magnitude is greater or equal -3 dB. It is possible to widen or even narrow the range of tolerance of the module. This shows that the overall functionality of detecting a configured frequency is ensured. Table I shows the output voltage of different input waves to the FDM. All input signals that include the configured wakeup frequency of 50 kHz caused the FDM to respond and, thus, activate the controller. A superposition of frequencies, not including 50 kHz, will not trigger the FDM. One possible source of a malfunction might be the superposition of different waves into a wave with the wakeup frequency as a result. We tested this by sending two sine waves with $f_1 = 30\,\text{kHz}$ and $f_2 = 70\,\text{kHz}$ to the input of the FDM, configured as above. As both waves have the same travelling speed, they will superpose to a beat wave with a beat frequency of $f = f_1 - f_2 = 40\,\text{kHz}$ and an

Table I – Output current with different input signals to the FDM that is configured to $50 \pm 0.25\,\text{kHz}$.

Input signal	Output voltage
50 kHz	0.81 V
30 kHz + 50 kHz	0.81 V
15 kHz + 30 kHz + 50 kHz	0.81 V
15 kHz + 30 kHz	0.00 V

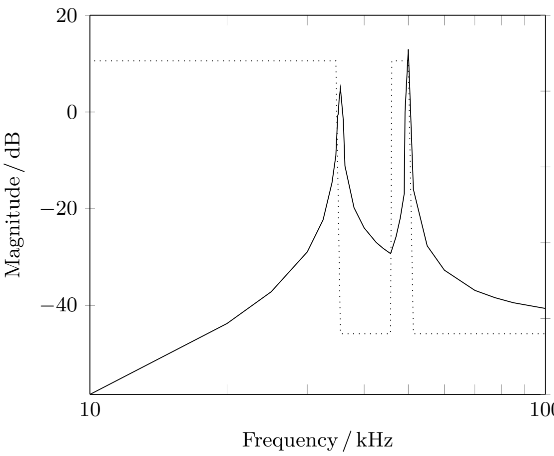

Figure 7 – Bode plot of a frequency detect module with partial networking enabled, configured to 35.5 kHz and 50 kHz

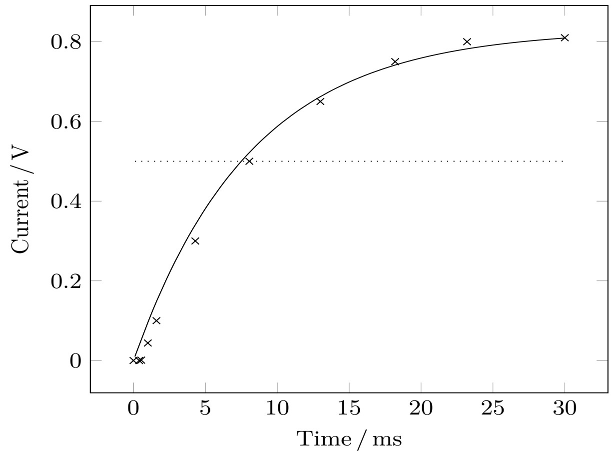

Figure 8 – Output current of a simulated frequency detect module subject to time. The FDM is stimulated with 50 kHz. The threshold voltage is set to 0.5 V.

average frequency of $f_{\text{inner}} = \nicefrac{1}{2}\,(f_1 + f_2) = 50\,\text{kHz}$. Therefore, the FDM will receive a valid wakeup signal and theoretically activate the ECU. But due to the beats and the filling time of the used capacitor, the output voltage is pulled to zero on each zero-crossing and never exceeds 1 mV. Nevertheless, this can be prevented by choosing the frequencies that are assigned to the single network clusters in such a way that none of them can be produced by a superposition of any combination of the remaining frequencies.

Partial Networking: LFW can also be configured to support partial networking. Figure 7 shows the bode plot of an FDM that is configured to wake up on detecting either a sine wave with $f = 35.5\,\text{kHz}$ or $f = 50\,\text{kHz}$. Both frequencies produce a narrow spike in magnitude whereas all other frequencies do not activate the FDM. This means, the ECU with this FDM would be logically part of two network clusters. It will wake up if one ECU in one of the separate clusters demands communication which therefore will send out a sine signal with the individual frequency of the cluster, it wants to communicate in.

B. Achieved Reduction of Wakeup Latency

One of the main novelties of this mechanism is the possibility to pass on a wakeup signal in intermediate nodes, enabling parallel and remote wakeup within point-to-point networks. This results in a significant reduction of the overall wakeup latency within a point-to-point network. The signal will be transmitted between nodes within the propagation delay. Figure 8 shows the detection latency of a simulated FDM that is stimulated with its activation frequency. The configured threshold voltage of 0.5 V is reached after $t_d = 8.06\,\text{ms}$. Which is the time an ECU with an FDM will need to detect a valid wakeup.

Let t_d be the time that a module needs to detect a wakeup signal, t_{wd} the wire delay and t_{P2Pd} the point-to-point latency that is caused by waking each node in a point-to-point network after its predecessor and waiting for the communication to be established. Let further t_{nw} be the time to activate one node and t_{sw} the time to activate one switch, respectively. Activation does in this context mean booting the controller and network till it is possible to send the first packet. If h is the number of hops that are taken on the longest route in the whole network and ℓ the sum of wire lengths in this route, the overall network wakeup time t_{wake} is given as

$$t_{\text{wake}} = \ell \cdot t_{wd} + t_d + \begin{cases} (h-1)\cdot t_{\text{P2Pd}} + t_{nw}, & \text{if } t_{sw} - t_{nw} < t_{\text{P2Pd}} \\ (h-2)\cdot t_{\text{P2Pd}} + t_{sw}, & \text{else} \end{cases} \tag{3}$$

provided that the last node in a chain within the network is always an endnode and not a switch. This assumption holds for most topologies. Even if the last node is a switch, $|t_{sw} - t_{nw}|$ is typically zero as the switch fabric can load its configuration in parallel to the PHYs and the link acquisition time with $t_{\text{linkAcquisition}} = t_{la} \gg t_{\text{switchConfigLoad}}$. Therefore, Equation (3) reduces to

$$\begin{aligned} t_{\text{wake}}\,(h,\ell) &= \ell \cdot t_{wd} + t_d + (h-1)\cdot t_{\text{P2Pd}} + t_{nw} \\ &= \ell \cdot t_{wd} + t_d + (h-1)\cdot t_{\text{P2Pd}} + t_{sw}. \end{aligned} \tag{4}$$

We determined $t_{la} = 100\,\text{ms}$ and $t_{nw} = t_{sw} = 137.5\,\text{ms}$ for typical automotive ECUs. The propagation delay can be calculated if the propagation speed of an electrical signal in a wire is known. A conservative estimate is $v_{\text{prop}} = 0.65 \cdot c = 1.95 \times 10^5\,\frac{m}{ms}$ which leads to $t_{wd} = 5.13\,\frac{ns}{m}$. To have an impact on the other times a cable-length of 1×10^6 m would be needed. Therefore, t_{wd} can be disregarded for the calculation which leads to

$$t_{\text{wake}}\,(h,\ell) \approx t_{\text{wake}}\,(h) = t_d + (h-1)\cdot t_{\text{P2Pd}} + t_{sw} \tag{5}$$

1) Latency of low frequency wakeup in comparison to known mechanisms: As given by Equation (5) the only determining topology-dependent variable for the overall wakeup latency is h. The different mechanisms are differentiated by the values of t_{sw} and t_{P2Pd}. We measured the variables for LFW and activation line and took times for energy detect from product sheets. The respective numbers are shown in Table II. Figure 9 shows the development of the wakeup time subject to the number of hops h in a network. It is clearly visible that the point-to-point latency has a major impact on the wakeup latency. The simulative overhead that is observed compared to the built activation line circuit is negligibly small in comparison to t_{P2Pd}.

The estimations for automotive networks currently fluctuate between 3 and 5 hops. For the future, where a complete point-to-point networked car is discussed, this numbers are tending to grow. The technique, described in this paper, achieves a

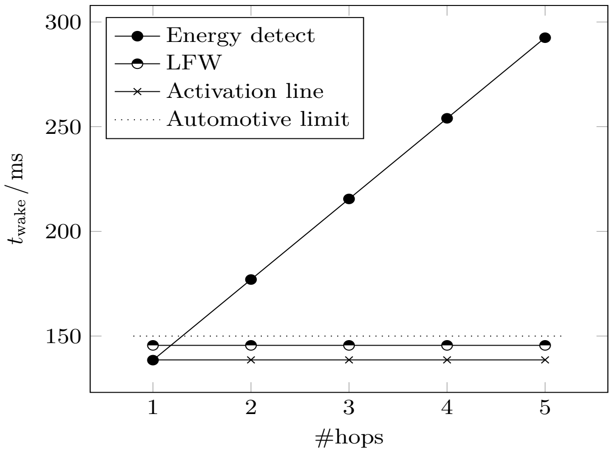

Figure 9 – Wakeup times of the examined mechanisms for Ethernet.

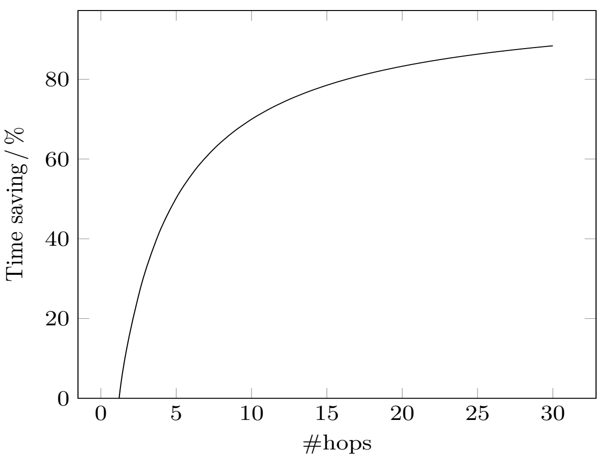

Figure 10 – Reduction of t_{wake} achieved by LFW against ED subject to the number of hops.

reduction of more than 30 % from 215.5 ms to 145.56 ms for a three hop network and thus respects the limit of 150 ms. In general, the reduction can be described as

$$1 - \frac{t_{\mathrm{wake}_{LFW}}}{t_{\mathrm{wake}_{ED}}} = 1 - \frac{t_{d_{LFW}} + t_{sw}}{(h-1) \cdot t_{\mathrm{P2Pd}_{ED}} + t_{d_{ED}} + t_{sw}} \tag{6}$$

Figure 10 shows the reachable time saving of the LFW against the known energy detect.
In comparison to an activation line low frequency wakeup is 5 % slower but it does not need an extra wire and enables partial networking which compensates the minimum latency drawback. The slightly higher latency might even be a result of the simulation and is very implementation dependent.

Table II – Determined wakeup times for different activation mechanisms.

	Time to detect wakeup	Point-to-point delay
LFW	8.06 ms	0 ms
Energy Detect	1 ms	38.5 ms
Activation line	1.12 ms	0 ms

V. Conclusion and Outlook

A novel physical layer mechanism called *Low Frequency Wakeup* to activate controllers in a network is introduced. The paper is focused on minimizing the point-to-point latency and therefore the advantages in terms of activation time. A comparison to current state-of-the-art technologies is drawn and it is shown that the main disadvantages of these mechanisms such as long wakeup latency and high standby power consumption can be avoided by using LFW. In comparison to already known mechanisms we achieved a 50 % reduction in wakeup time for a five hop network. The reason for this is the novel pass through technique for switches. Additionally, partial networking is enabled without the high standby power consumption of known mechanisms.

Further investigation should concentrate on designing a configurable electronic circuit. This makes it possible to compare the standby power consumption of LFW with the automotive requirements and to already known technologies. To evaluate the usability within the automotive environment the electronic circuit needs to be tested in subject to EMC.
Furthermore, other wakeup mechanisms like the mentioned *Signal over Ethernet* approach can be examined further, especially with focus on reducing the point-to-point latency by passing on the wakeup signal in the switches.

References

[1] Advanced Micro Devices. Magic Packet Technology. 1995. url: http://support.amd.com/us/Embedded%5C_TechDocs/20213.pdf.

[2] N. Balbierer, J. Nöbauer, and A. Rath. Method for Activating a Network Component of a Motor Vehicle Network System. *Patent*, (WO/2011/101491), 2011.

[3] N. Balbierer, T. Waas, J. Noebauer, and J. Seitz. Energy consumption of Ethernet compared to automotive bus networks. In *Intelligent solutions in embedded systems (wises)*, 2011, pp. 61–66.

[4] K. Christensen and P. Reviriego. IEEE 802.3 az: the road to energy efficient ethernet. *Ieee*, (November):50–56, 2010. url: http://www.csee.usf.edu/~christen/energy/commMag10b.pdf.

[5] J. Fröschl. Data Bus System for Motor Vehicles with Time Interrupt Connection. *Patent*, 1(EP 1 219 070 B1):1–7, 2007.

[6] M. Glass, D. Herrscher, H. Meier, M. Piastowski, and P. Schoo. SEIS – Security in Embedded IP-based Systems. *Atzelektronik worldwide*, 1:36–40, 2010.

[7] Klemmenbezeichnungen in Kraftfahrzeugen. *Norm*, (DIN 72552):1–23, 1971.

[8] NXP Semiconductors. TJA1145 datasheet.

[9] Road vehicles - Controller area network (CAN) – Part 6: High-speed medium access unit with selective wake-up functionality. Norm. 2012.

[10] P. Rockel and M. Kaindl. Steuergerät und Verfahren zum Betrieb des Steuergeräts sowie KFZ mit derartigem Steuergerät. *Patent*, (DE 10 2008 030 222 A1), 2008.

[11] C. E. Spurgeon. *Ethernet: the definitive guide*. O'Reilly, 2000. isbn: 1565926609.

[12] System of wire bus networked controllers with reduced residual current consumption. *Patent*, (DE19715880), 1998.