Reducing Communication Requirements for Electric Vehicle Charging using Vehicle-Originating-Signals

Victor del Razo, Christoph Goebel, Hans-Arno Jacobsen
Department of Computer Science. Technische Universität München. Munich, Germany. Contact: victor.del-razo-sarmina@tum.de

Abstract—We propose a method for reducing the communication requirements for electric vehicle (EV) charging control based on Vehicle-Originating-Signals (VOS). The original VOS approach requires a message exchange between every EV and the aggregator for every time step. This can cause the load on communication systems to increase, especially as the number of EVs increases and the control intervals become more granular. We explore (i) the reduction of EV-originating messages, (ii) the reduction of aggregator-originating messages, and (iii) the challenges in combining both reduction methods. EVs reduce the number of messages by including possible future values in a single message. The aggregator reduces the number of messages by sending single broadcast signals. Combining both methods requires a retransmission protocol. For the evaluation, we compare the original and the improved VOS approaches on a load leveling scenario based on electricity demand, solar generation, and car mobility data from Munich, Germany. The results show that it is possible to reduce the overall message requirements with a minor effect on performance. With savings of over 70% in number of messages, these improvements contribute towards more network-friendly solutions for smart EV charging.

I. INTRODUCTION

To keep the electric grid stable, power supply and demand have to be balanced at all times. To date, this balance is preserved by dispatching generators to match a given demand. As the share of energy from renewable sources like wind and solar continues to increase, demand-side management and energy storage are becoming more important. Controlling flexible loads, in particular, could help to achieve the ambitious renewable integration targets set by different countries.

The Vehicle-Originating-Signals (VOS) approach, based on the Car-Originated-Signals (COS) approach [1], enables an aggregator to control how a fleet of electric vehicles (EV) charges in order to follow an arbitrary power profile. This approach, however, relies on bidirectional communication between EVs and the aggregator in every control period or time step. Although lower than a per-iteration message exchange in some iterative distributed optimization mechanisms, this still represents an additional load on the communications network. For example, for performing control on 15 minutes intervals, we need to send and receive messages every 15 minutes. This load increase becomes more significant as the number of EVs and the granularity of the control intervals (i.e., seconds instead of minutes) continue to grow.

This work presents a method to further reduce the communication requirements of the VOS approach by minimizing the number of messages exchanged between the EVs and the aggregator. We also evaluate the method’s performance with respect to the original VOS approach and discuss its trade-offs and challenges. For the evaluation, we use the same load leveling objective as [1], based on a scenario with actual electricity demand, solar generation, and car mobility data from Munich, Germany.

The results show that it is possible to reduce the number of messages with savings of over 70% and a minor impact on the algorithm’s performance. When reducing the number of messages sent by the EVs, however, the message’s size increases accordingly. In this context, a method for the efficient use of communications network resources is proposed. Reducing control signals to a single broadcast message per time step results in a moderate increase in uncertainty, meaning that the aggregator does not know every car’s charging action. In addition, it is possible to combine both methods by implementing a retransmission protocol. The additional messaging requirements resulting from this retransmission protocol are low compared with the overall savings potential.

Our contribution can be summarized as follows: (i) a method to reduce the frequency of messages sent by EVs by incorporating the combination of future values in a single message, (ii) a method to reduce the number of messages sent by the aggregator by introducing a single broadcast set point message as opposed to dedicated messages to each EV, (iii) an analysis of challenges and requirements together with a solution proposal for combining the two methods above, and (iv) a quantified analysis of the trade-off between reduction in number of messages, increase in message size, additional retransmission requirements and performance.

The rest of the paper is organized as follows: Section II briefly covers previous work in EV charging, Section III summarizes the VOS method, Sections IV to VI discuss the different message reduction methods, Section VII shows our evaluation, Section VIII presents discussion and conclusions.

II. RELATED WORK

The related work in the area of EV charging strategies can be classified based on system architecture (centralized/decentralized) and control method (direct/incentive-based). For instance, [2] proposes a decentralized algorithm that controls EV charging to fill demand valleys via incentive signals. [3] presents a framework for incentive-based distributed EV charging control for different objectives. [4] evaluates the supply of frequency reserves via centralized and direct EV control, [5] presents a decentralized optimization with direct control to minimize charging costs, [6] presents a local direct control technique for distribution network (DN) voltage stability, [7] aims at minimizing power losses with a centralized, direct control approach, and [8] presents a decentralized direct control method that includes other flexible loads. Most of these use formal optimization methods and focus on offline optimization.

Centralized optimization methods face the challenge of scalability as the size of the problem increases. Although distributed optimization approaches can be used to address scalability limitations, we believe that actually implementing such solutions would cause more challenges, mostly due to high communication requirements.
Alternatively, one can address the scalability challenge by optimizing for the aggregated EV load like in [9] and [10]. In the former, EVs are clustered into a single integer variable that can take values proportional to the number of EVs, whereas in the latter they are grouped by arriving and parking times. Clustering, however, requires a second mechanism to control the individual EVs.

The VOS approach [1], enables vehicles to compute signals reflecting their need for charge and willingness to supply power. An aggregator then collects these signals and implements the control with minor computational effort. This approach combines the benefits of a centralized architecture regarding simplicity, but distributes part of the computation to the EVs. It does not require iterative optimization and could therefore meet the real-time requirements of advanced applications. It is a direct control method to ensure certainty in control but, to a certain extent, protects EV users’ privacy by limiting information exchange. The VOS approach still requires messages to be exchanged at every time step. Since signal computation and control are decoupled, it is possible to reduce this requirement.

The concept of a single broadcast signal is present in incentive-based approaches like [2] and [3]. The idea of broadcasting a set point signal paired with a probability factor is explored in [11]. These concepts allow for the reduction on message requirements from the aggregator to the EVs. In this work, we build on these concepts for reducing the aggregator-to-EVs message requirements and exploit the modular characteristic of the VOS approach to also reduce the EVs-to-aggregator message requirements.

III. VEHICLE-ORIGINATING-SIGNALS APPROACH

In this section, we introduce the VOS approach presented in [1]. For a given area of a distribution network and the loads and distributed generation connected to it, the goal of the aggregator is to control the charging and discharging behavior of EVs such that the aggregated power profile matches a target power profile, $P_O(k)$, as closely as possible. The aggregated power profile, $P_{agg}(k)$, is defined as the difference between the demand and the local generation within the defined area, i.e., the total power demanded from the grid at time step $k$. That is:

$$P_{agg}(k) = P_D(k) - P_S(k) + \sum_{i=1}^{N} P_{EV}(k)$$

where for a given time step $k$, $P_D(k)$ represents the inflexible demand, $P_S(k)$ denotes the produced solar power, and $P_{EV}(k)$ is the power consumed by the $i^{th}$ EV. $P_{EV}$ can be negative, meaning that the EV is supplying energy back to the grid.

Our objective is to minimize $|P_O(k) - P_{agg}(k)|$ for all $k$ subject to the set of constraints of each vehicle $X_{EV}$ and the limitations of the electricity distribution network $X_{DN}$. Using (1) we can express this problem formally as follows:

$$\min \ P_{EV}(k) \left( P_O(k) + P_S(k) - P_D(k) - \sum_{i=1}^{N} P_{EV}(k) \right)^2, \forall k \in T$$

subject to

$$P_{EV}(k) \in X_{DN}, \forall k \in T$$

$$P_{EV}(k) \in X_{EV}, \forall k \in T$$

In the Vehicle-Originating-Signals (VOS) approach, an aggregator directly controls a fleet of EVs as illustrated in Fig. 1. At each time step, every connected EV computes a Need-for-Charge (NfC) and a Willingness-to-Supply (WtS) signal and sends them to the aggregator. The EV-related constraints are considered when computing these signals. The aggregator collects the signals, takes DN limitations into account, and returns charging instructions to the EVs.

This approach has some immediate advantages. First, the NfC and WtS signals are computed by every EV, resulting in high parallelization of computation and preservation of the EV owner’s data privacy. Second, EVs send two scalars per time step as opposed to a full set of vehicle states and constraints, resulting in lower communication requirements. Third, the method is not iterative, resulting in moderate computation and lower message exchange, particularly when compared with iterative distributed approaches. Finally, the NfC and WtS are computed values, so the number of messages can potentially be further reduced by sending combinations of future possible values for a given time period.

1) Need for Charge Signal: The NfC is defined as a function of the remaining connection time and the time required to reach the desired state of charge (SOC) with an additional threshold policy to ensure that the EV constraints are satisfied. That is,

$$NfC^{i}(k) = \begin{cases} 
C_{QoS}, & \text{if } NfC_{temp} \geq C_{QoS} \\
N fC_{temp}, & \text{if } C_{tar} \leq N fC_{temp} < C_{QoS} \\
C_{tar}, & \text{if } N fC_{temp} < C_{tar} \\
C_{full}, & \text{if } E_{bat} = E_{max}^{i} 
\end{cases}$$

where

$$N fC_{temp} = C_{QoS} \frac{k^{i}_{req}}{k_{max}^{i}}$$

$$k^{i}_{req} = \frac{SOC_{tar}^{i} \cdot E_{max}^{i} - E_{bat}^{i}}{E_{rate}^{i}}$$

$$k_{dep}^{i} = k_{dep}^{i} - k$$

$SOC_{tar}^{i}$ is the target SOC of EV $i$ in percent, $E_{max}^{i}$ is the battery capacity, $E_{bat}^{i}$ is the current energy in the battery, $E_{rate}^{i}$ is the allowed charging rate limit per time step, $k_{dep}^{i}$ is the departure time, and $k^{i}_{req}$ and $k_{max}^{i}$ are the required and available time steps, respectively, where $k^{i}_{req}$ can take non-integer values. $C_{QoS} > C_{tar} > C_{full}$, are

1Only two derived metrics (NfC and WtS) are revealed to the aggregator as opposed to, e.g., parking times, SOC or vehicle objectives and constraints.
thresholds where \( C_{QoS} \) indicates that the EV must charge in order to reach \( SOC_{tar} \), \( C_{tar} \) indicates that the EV has reached \( SOC_{tar} \), and \( C_{full} \) indicates that the EV’s battery is fully charged.

2) Willingness to Supply Signal: For the computation of the WiS, we consider the thresholds \( C_{max} \) indicating the maximum willingness, and \( C_{noS} \) indicating that this car is not willing to supply at all, where \( C_{max} \gg C_{noS} \). For a given EV and time step, the WiS is a function of the available time \( k_{dep} \), the departure time \( k_{dep} \), and the current \( SOC \). To set the WiS within its thresholds, we scale it by \( C_{max} \) and shift it by \( C_{noS} \), resulting in:

\[
WiS(i)(k) = \begin{cases} 
C_{noS}, & \text{if } E_{bat}^i \leq E_{min}^i \text{ or } k_{dep}^i \geq k_{ev}^i \\
C_{max} \cdot SOC^i \cdot \frac{k_{dep}^i}{k_{dep}^i + C_{noS}}, & \text{otherwise}
\end{cases}
\]  

(7)

3) The Aggregator: At each time step, the aggregator receives the NfC and WiS signals from the EVs, defines a charging strategy and instructs the EVs to charge accordingly. The process is:

1) \( P_{agg}(k) \leftarrow P_{2}(k) - P_{1}(k) \)
2) Sort EVs by NfC in descending order
3) While \( P_{agg}(k) < P_{max} \) and \( NfC = C_{QoS} \)
   a) Charge EVs with \( C_{fut} < C_{QoS} \)
   b) Charge EVs
      - Continuous case:
      - Allocate available power to EVs proportional to NfC
      - Integer case:
      - Allocate available power to EVs with highest NfC
   c) \( P_{agg}(k) \leftarrow P_{agg}(k) + \sum_{i=1}^{N} P_{EV}^i(k) \)
4) If \( P_{agg}(k) \leq P_{2}(k) \)
   a) Select EVs with \( C_{fut} < C_{QoS} \)
   b) Charge EVs
      - Continuous case:
      - Allocate required power to EVs proportional to NfC
      - Integer case:
      - Allocate required power to EVs with highest WiS
   d) \( P_{agg}(k) \leftarrow P_{agg}(k) + \sum_{i=1}^{N} P_{EV}^i(k) \)
5) Else (meaning \( P_{agg}(k) > P_{2}(k) \))
   a) Sort remaining cars by WiS in descending order
   b) Select EVs with \( WiS > C_{noS} \)
   c) Discharge EVs
      - Continuous case:
      - Allocate required power to EVs proportional to WiS
      - Integer case:
      - Allocate required power to EVs with highest WiS

IV. EV-TO-AGGREGATOR MESSAGES

In the traditional VOS approach, a message containing the values of the NfC and WiS signals is sent in every time step. From (3) and (7), we identify that these signals only depend on local information. Furthermore, the only value that is externally influenced is the current energy in the EV’s battery \( E_{bat} \) which depends on the charging power \( P_{EV}^i \) assigned by the aggregator.

Since the range of values for \( P_{EV}^i \) is known, it is possible to calculate a set of possible future values of NfC and WiS in terms of \( P_{EV}^i \). This set of possible values can be sent to the aggregator in a single (and larger) message with a lifetime of several time steps. On every time step, the aggregator can then choose the NfC and WiS values for each EV depending on the calculated \( P_{EV}^i \), instead of receiving a new message.

Fig. 2 illustrates how an EV builds the set of possible NfC values. Messages are sent at the time step \( k = \tau \) and have a message lifetime of \( T_M \) time steps. For a given EV, \( \tau \) is equal to the arrival time \( h_{in} \) for the first message. The EV then computes the possible NfC values based on the possible values of \( P_{EV}^i \).

As \( k \) increases, the set of possible NfC values forms a complete \( V \)-ary tree [12], where \( V \) indicates the number of values that \( P_{EV}^i \) is allowed to take. The height of the tree \( h_M \) is \( T_M - 1 \) except when the EV departs before the end of the message lifetime or \( k_{dep} < \tau + T_M \) resulting in \( h_M = k_{dep} - \tau - 1 \).

There is, however, a significant trade-off between the reduction of the number of messages and the message size. For our \( V \)-ary tree, the total number of nodes (i.e., NfC values) is [12]:

\[
L^V_M(\tau) = \frac{V^{h_M(\tau)}+1}{V-1} \tag{8}
\]

where:

\[
h_M(\tau) = \min(T_M, k_{dep} - \tau) - 1 \tag{9}
\]

The lower bound in terms of the number of messages can be achieved if the message lifetime \( T_M \rightarrow \infty \). In this case, every EV sends only one message which is valid for the entire plug-in period. The message size depends on the length of this plug-in period and therefore varies from EV to EV.

From (8), we can see that the size of the message depends also on \( V \). This means that \( P_{EV}^i \) can only take a discrete number of values so this approach cannot be applied for a strictly continuous \( P_{EV}^i \). However, it is possible to approximate this for a continuous approach by increasing the granularity of \( P_{EV}^i \) at the cost of increasing \( V \). For a granularity of \( \frac{P_{max}^{EV}}{P_{step}^{EV}} \), we would have:

\[
V = 1 + \frac{2}{P_{step}^{EV}} \tag{10}
\]

E.g., if we want a granularity of one tenth of \( P_{max}^{EV} \), then \( V = 21 \).

The lower bound in terms of message size is therefore achieved when \( P_{step}^{EV} = 1 \) so \( P_{EV}^i(k) \) can take the values \( \{-P_{max}^{EV}, 0, P_{max}^{EV}\} \) and \( V = 3 \), i.e., we have a tertiary tree. The other lower bound for the message size holds when \( T_M = 0 \) but that means sending messages every time step which is against our objective.

A second tree could be built for the WiS signal. However, since both values depend on the same \( P_{EV}^i \), this is solved more efficiently by integrating both values into the tree. That is, each node of the tree contains a value for NfC and WiS. This also would enable a more efficient data compression and lower overhead. For example, one would need a single pointer for both values and one could compress both values into a single frame.
Since our tree is full and complete, we don’t have a significant overhead in terms of structures. A serialized list could be sent by the EV and ordered by the aggregator based on a predefined sequence. Therefore, we can approximate the new message size as $L_{M}(t_{i})$ times the original (one NIC and WiS) message size.

Still, the message size grows exponentially and this method should be applied with caution. For example, if $V = 3$, $L_{M}$ increases from 40 to 29,524 nodes for lifetimes $T_{M}$ of 4 and 10 time steps, respectively. If $V = 10$, it grows from 1,111 to 1.1 billion nodes.

If we want to strictly minimize the number of messages, we also want to avoid large messages that need to be fragmented before being sent. In addition, we want to include as much information as possible within this single unit. We therefore can use the concept of maximum transmission unit (MTU) in communication networks to define a target message size that ensures no fragmentation and maximum information exchange.

We define the net MTU as the payload size without transmission overhead, i.e., $MTU_{net} = MTU - OH_{msg}$. The number of nodes would be $L_{M}^{tar} = \frac{MTU_{net}}{S_{node}}$, where $S_{node}$ is the node size in bytes, and depends on the protocol and compression applied. $L_{M}^{tar}$ is only an intermediate value and can take any real value.

From (8) and (9), we can define the tree’s target height which in turn defines the target message lifetime $T_{M}^{tar}$ as:

$$h_{M}^{tar} = \frac{\log \left( (V-1)L_{M}^{tar} + 1 \right)}{log V} - 1$$

$$T_{M}^{tar} = \left[ h_{M}^{tar} + 1 \right] = \left[ \frac{\log \left( (V-1)L_{M}^{tar} + 1 \right)}{log V} \right]$$

The actual number of nodes sent is:

$$L_{M}^{tar} = \frac{V T_{M}^{tar} - 1}{V - 1}$$

For example, an Ethernet network has MTU = 1500 bytes, which is also a common default value for DSL and 3G routers. For an overhead of 10% (reasonable when considering higher level protocols) and $S_{node} = 8$, corresponding to two (NIC and WiS) floating point values of 4 bytes each, we have $MTU_{net} = 1.350$ which for $V = 3$ can fit up to $L_{M}^{tar} = 168.75$ nodes, allowing for a message with lifetime $T_{M}^{tar} = 5$ and $L_{M}^{tar} = 121$ tuples of NIC/WiS values.

V. AGGREGATOR-TO-EV MESSAGES

The VOS approach implements a direct control by sending a charging instruction to every EV. These instructions depend on the EV’s NIC and WiS values and the available resources at the time. If we consider the integer case, meaning that $P_{EV}^{*}(k)$ can take the values $\{-P_{max}^{EV}, 0, P_{max}^{EV}\}$, the aggregator assigns resources based on a highest-first approach. That means that the NIC of the last EV that was instructed to charge can be seen as a set point: all EVs with higher NIC charged, and all EVs with lower NIC did not.

One can potentially reduce the control messages to a single broadcast signal containing the set point values for the NIC and WiS. For this to yield the same results, however, each EV must have unique values for NIC and WiS, and since these only depend on local EV information, we cannot ensure uniqueness. We therefore propose a solution in which the set point values for NIC and WiS are coupled to a probability rate to steer EVs with NIC or WiS values equal to the set point. In other words, for the NIC case, those EVs with higher NIC than the set point charge, those with a lower do not and those with an NIC equal to the set point charge with probability $p(NIC)$.

The aggregator’s algorithm is modified as follows:

1. $P_{agg}(k) \leftarrow P_{agg}(k) - P_{agg}(k) - P_{agg}(k)$
2. Sort EVs by NIC in descending order (index $i^{*}$)
3. While $P_{agg}(k) < P_{max}^{NIC} + N_{fC} = C_{QoS}$
   a) $P_{agg}(k) \leftarrow P_{agg}(k) + P_{max}^{NIC}^{EV}$
   b) $i^{*}++$
4. While $P_{agg}(k) \leq P_{0}(k) + C_{nS} < C_{QoS}$
   a) $P_{agg}(k) \leftarrow P_{agg}(k) + P_{max}^{WtS}^{EV}$
   b) $i^{*}++$
5. NIC_Set_Point $\leftarrow$ NIC($^{*}$)
6. Calculate Probability Rate
   a) $a = \text{Count NIC = NIC_Set_Point in Sortedlist (0:end)}$
   b) $b = \text{Count NIC = NIC_Set_Point in Sortedlist (0:end)}$
   c) NIC_Probability Rate $\leftarrow a/b$
7. Sort remaining cars by WiS in descending order (index $^{*}$*)
8. While $P_{agg}(k) \geq P_{0}(k) + C_{inS}^{max}$
   a) $P_{agg}(k) \leftarrow P_{agg}(k) - P_{max}^{WtS}^{EV}$
   b) $i^{*}++$
9. WtS_Set_Point $\leftarrow$ WiS($^{*}$*)
10. Calculate Probability Rate
    a) $a = \text{Count WiS = WtS_Set_Point in Sortedlist (0:end)}$
    b) $b = \text{Count WiS = WtS_Set_Point in Sortedlist (0:end)}$
    c) WtS_Probability Rate $\leftarrow a/b$
11. Broadcast
    - NIC_Set_Point, NIC_Probability Rate
    - WtS_Set_Point, WtS_Probability Rate

Each EV then receives the broadcasted signal and checks first the NIC set point. It charges if (i) its NIC is larger, or (ii) its NIC is equal and a random function returns true. If the EV did not charge, it checks the WiS set point and discharges if (i) its WiS is larger, or (ii) its WiS is equal and a random function returns true. This random function returns true if the output of a uniform random number generator of range [0,1] is less than or equal to the set point’s probability rate of the NIC or WiS, respectively.

The use of a probability factor implies an increase in uncertainty. This uncertainty, however, exists only for those cases where there are more vehicles with the same values as the set points. The probability of having more than one EV with the same NIC or WiS values increases with the number of EVs. Nevertheless, the impact of this uncertainty also decreases as the fleet size increases. The more EVs, the likelier it is that the practical distribution matches the theoretical one.

VI. COMBINED APPROACH

It is possible to combine the two approaches above to enable further reduction in the number of messages. This, however, requires some adjustments. EV-message reduction is based on the principle that future NIC/WiS values can be accurately predicted for a given charging action. However, the aggregator-reduction introduces a degree of uncertainty that challenges EV-message reduction. The EV does not know beforehand if and when a case with NIC/WiS equal to the set point will occur, nor the action that would be taken accordingly. The aggregator knows how many EVs are under this condition and, of these, how many are expected to take action, but it does not know which EV takes what action.

This challenge can be addressed by introducing a retransmission protocol. Whenever an EV encounters a set point value equal to its NIC or WiS, it schedules a message transmission for the next time step. That is:

$$P_{agg}(k) \leftarrow P_{agg}(k) - P_{agg}(k) - P_{agg}(k)$$

While $P_{agg}(k) < P_{max}^{NIC} + N_{fC} = C_{QoS}$
   a) $P_{agg}(k) \leftarrow P_{agg}(k) + P_{max}^{NIC}^{EV}$
   b) $i^{*}++$
4. While $P_{agg}(k) \leq P_{0}(k) + C_{nS} < C_{QoS}$
   a) $P_{agg}(k) \leftarrow P_{agg}(k) + P_{max}^{WtS}^{EV}$
   b) $i^{*}++$
5. NIC_Set_Point $\leftarrow$ NIC($^{*}$)
6. Calculate Probability Rate
   a) $a = \text{Count NIC = NIC_Set_Point in Sortedlist (0:end)}$
   b) $b = \text{Count NIC = NIC_Set_Point in Sortedlist (0:end)}$
   c) NIC_Probability Rate $\leftarrow a/b$
7. Sort remaining cars by WiS in descending order (index $^{*}$*)
8. While $P_{agg}(k) \geq P_{0}(k) + C_{inS}^{max}$
   a) $P_{agg}(k) \leftarrow P_{agg}(k) - P_{max}^{WtS}^{EV}$
   b) $i^{*}++$
9. WtS_Set_Point $\leftarrow$ WiS($^{*}$*)
10. Calculate Probability Rate
    a) $a = \text{Count WiS = WtS_Set_Point in Sortedlist (0:end)}$
    b) $b = \text{Count WiS = WtS_Set_Point in Sortedlist (0:end)}$
    c) WtS_Probability Rate $\leftarrow a/b$
11. Broadcast
    - NIC_Set_Point, NIC_Probability Rate
    - WtS_Set_Point, WtS_Probability Rate

Each EV then receives the broadcasted signal and checks first the NIC set point. It charges if (i) its NIC is larger, or (ii) its NIC is equal and a random function returns true. If the EV did not charge, it checks the WiS set point and discharges if (i) its WiS is larger, or (ii) its WiS is equal and a random function returns true. This random function returns true if the output of a uniform random number generator of range [0,1] is less than or equal to the set point’s probability rate of the NIC or WiS, respectively.
The trade-off here is the number of triggered retransmissions. A retransmission is triggered if either the NfC or the WtS are equal to the set point, so the number of retransmitted messages $M_{rtx}$ depends on how often this happens. Yet, $M_{rtx}$ is the upper bound for the cost of combining both approaches. Since the message lifetime $T_M$ remains the same, the new message spans over $T_{new} + T_M$, so no message needs to be sent at $T_{old} + T_M + 1$. The closer the retransmission to the original transmission, the higher the cost of retransmission and the higher the amount of transmitted data that is later discarded.

For example, an EV sends its NfC/WtS tree upon connection, receives a broadcast signal with the set point equal to its NfC and the random function results in charging. The aggregator has no way of knowing what charging action the EV took so it cannot choose the next NfC value from the tree. The EV needs to retransmit a tree in the next time step, which, since happening shortly after the first transmission, becomes expensive from a network resources perspective.

VII. Evaluation

The VOS approach can be used to make the aggregated demand follow arbitrary target profiles. Like in [1], our target in this evaluation is a constant power profile $P_O(k) = P_O$, which results in load leveling. The $P_O$ is the average of $P_{agg}(k)$ and the average EV consumption is computed based on target SOC and connection times. A flat power profile allows for renewable energy to be consumed locally and generation to be planned more efficiently since the aggregated demand of the controlled area is less variable over time.

A. Data and Simulation Setup

The Munich DSO [13] supplies yearly information on a 15 min granularity. We take the load and distributed intake (reflecting local solar generation) data for the low voltage level and select a 24 hour period starting on a given date at noon, such that the considered time period spans an entire night. Then, we scale the load profile by $E_{demand} = 3\%$ to obtain a demand magnitude manageable by the intended EV fleet sizes of 4,000-19,000 EVs. Finally, we use the same factor for scaling the solar generation data.

The official mobility survey conducted by the German government [14] includes data for all of Germany. We filter data from major cities (>500 inhabitants) in order to get a data set valid for Munich and large enough to be representative. Additionally, we apply a set of quality assurance rules to ensure that the interviewed person is the driver, the used vehicle belongs to the household, and the average reported speeds are lower than 120 km/h. For workdays (Monday-Thursd), that is around 1,800 trips from 500 households. In our experiments, we select entries for workdays and up to three cars per household to build $N$ vehicle driving profiles.

If the size of the fleet $N$ we want to generate is higher than three times the number of trips available, we introduce a random normal noise to the starting time and trip duration and adjust the remaining parameters accordingly. This noise has mean zero, a standard deviation of 15 minutes, and a maximum of 1 hour.

We consider a homogeneous fleet of EVs with $SOC_{tar} = 85\%$, battery maximum and minimum capacities of $E_{max} = 16$kWh and $E_{min} = 1$kWh, and charging power $P_{EV} = 4$kW. Furthermore, we assume that the EVs charge only at home and are plugged only once (during the longest parking period). The last assumption implies a more realistic scenario where users only plug the car once their driving day is over rather than every time they park at home.

We run Monte Carlo simulations with fleet sizes of $N = 4,000 \cdots 19,000$ in steps of 500 EVs. For each $N$, we run 100 simulations keeping the same values except for the generated driving profiles. This allows us to eliminate biases resulting from specific driving profile combinations. We consider the integer problem, i.e., $P_{EV}(k) \in \{ -P_{max}^{EV} , 0 , P_{max}^{EV} \}$. We measure the absolute error $|P_{O} - P_{agg}(k)|$ for every run and time step. Then, we organize the measurements by EV fleet size and calculate the percentile of the error. For the evaluation, we report the 80-, 95-, 99- and 99.9-percentiles. In the following, we refer to the process described above as one Monte Carlo run (MCrun).

We compare the traditional VOS and our modified VOS approach by running one MCrun for each. We use random weekdays in spring for each of the 100 repetitions, but use the same dates for the different values of $N$ within an MCrun and for the two approaches. The load and solar generation are taken exactly for the generated date and the driving profiles are produced for spring weekdays.

B. Results

1) EV-to-Aggregator Messages: Table I summarizes the results for the EV-to-Aggregator message reduction method. We analyze the MCrun for 4,000 cars, that is 100 simulations each with 96 time steps. We report results for different sizes of NfS/WtS signals and number of MTUs per message. We assume it is possible to format the NfC/WtS to different bit sizes (8-32) and consider up to three 1500 bytes-large MTUs per message with an effective payload of 90%. The message lifetime is calculated based on (12). Reductions are reported as $1 - \text{result/reference}$, where the references are the total, mean, and 3rd quartile of the messages sent by EVs in the traditional VOS.

We achieve savings of at least close to 80%. We can improve reduction by allowing messages larger than one MTU (fragmentation) or compressing the data. The second alternative should be preferred as it also reduces number of packets and therefore the network usage.

2) Aggregator-to-EV Messages: Fig. 3 presents the results for the message reduction at the aggregator level. These graphs display the 80-, 95-, 99- and 99.9-percentiles with the number of EVs on the x-axis and the normalized absolute error on the y-axis, for the original VOS and the broadcast signal approach.

The results for up to a fleet size of 7 thousand EVs are very similar for both approaches. Fewer cars result in fewer signals with

<table>
<thead>
<tr>
<th>Signal size (bits)</th>
<th>MTU per mes.</th>
<th>Mes. life time</th>
<th>Reduction in total</th>
<th>Reduction in mean</th>
<th>Reduction in EV</th>
<th>Reduction in sq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>1</td>
<td>5</td>
<td>82.7%</td>
<td>10.8%</td>
<td>82.7%</td>
<td>83.1%</td>
</tr>
<tr>
<td>64</td>
<td>2</td>
<td>6</td>
<td>82.7%</td>
<td>10.8%</td>
<td>82.7%</td>
<td>83.1%</td>
</tr>
<tr>
<td>128</td>
<td>3</td>
<td>6</td>
<td>82.7%</td>
<td>10.8%</td>
<td>82.7%</td>
<td>83.1%</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>6</td>
<td>82.7%</td>
<td>10.8%</td>
<td>82.7%</td>
<td>83.1%</td>
</tr>
<tr>
<td>32</td>
<td>2</td>
<td>7</td>
<td>85.0%</td>
<td>9.4%</td>
<td>85.0%</td>
<td>85.7%</td>
</tr>
<tr>
<td>64</td>
<td>3</td>
<td>7</td>
<td>85.0%</td>
<td>9.4%</td>
<td>85.0%</td>
<td>85.7%</td>
</tr>
<tr>
<td>128</td>
<td>4</td>
<td>7</td>
<td>85.0%</td>
<td>9.4%</td>
<td>85.0%</td>
<td>85.7%</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>7</td>
<td>85.0%</td>
<td>9.4%</td>
<td>85.0%</td>
<td>85.7%</td>
</tr>
<tr>
<td>32</td>
<td>2</td>
<td>7</td>
<td>85.0%</td>
<td>9.4%</td>
<td>85.0%</td>
<td>85.7%</td>
</tr>
</tbody>
</table>
exact same values. The 80-percentile remains mostly unaffected while the higher percentiles show a negative impact on performance, until around 16 thousand EVs. This impact, however, is below 0.01 for the highest percentile and quite minimal for the rest.

The benefits of a single broadcast signal are higher as the fleet size increases. This approach saves 3,999 messages per time step for a 4,000 EV fleet and 19,999 for an EV fleet of 20,000 EVs.

3) Combined Approach: Table II presents the results for the combined approach. We analyze the MCRun for 4,000 EVs and compare the results against the two approaches applied as standalone and the traditional VOS approach. We consider the case with $T_M = 5$. We count the total number of messages sent for each of the 100 simulations, calculate reductions per run and report mean, maximum and minimum. Reductions are reported as 1 – result/reference, where the references are the total number of messages sent by EVs and aggregator in one run.

In all cases we achieve significant reductions which suggests that it is worth to combine both approaches. The most important factor influencing the reduction is the number of retransmissions. In our experiments we experienced on average around 87 thousand retransmission per run. That is around 21 messages per vehicle or almost twice the average of the EV-reduction approach. These number of retransmissions are somehow biased to our data source. Similar profiles produce similar VOS signals which increase the probability of retransmission. Since our experiments use a limited number of samples to produce driving profiles, we believe that the reduction in a real world scenario is potentially higher.

VIII. DISCUSSION AND CONCLUSIONS

We introduced modifications to the VOS approach for EV charging control that allow us to significantly reduce the number of exchanged messages, enabling a more efficient use of communication resources with very limited influence in performance.

By exploiting the modularity and flexibility of the VOS approach, we split the problem in EV-originating and aggregator-originating message reduction. EV-originating messages are reduced around 80% by transmitting tree structures containing future possible values in a single message with the trade-off of an increase in the message size. To keep this size within applicable limits, we can restrict it to one or few MTUs resulting in message sizes within the kilobyte range. Aggregator-originating messages are reduced by broadcasting a single control signal as opposed to individual ones. This single message contains a pair of set point values and probability factors for NfC and WiS. Savings of over 99.9% are possible, however, with a slight impact on performance.

Combining both approaches is possible but requires a retransmission protocol. Overall savings of more than 70% in comparison with the traditional VOS approach are possible and improvements of at least 50% are achieved with respect to using the two methods individually.

Larger time steps require less frequent communication but would tend to group message exchanges into smaller periods. This results in peaks of transmitted data, which may become critical when using the EV message reduction method, since messages increase in size. This can be addressed with protocols that distribute the load along a larger time frame. Additionally, the MTU of a given path varies depending on the type of network and even varies with network utilization. Ideally, we could dynamically adjust the message lifetime to the reported MTU (e.g., path MTU discovery) at a given time.

The proposed modification allows for an efficient implementation of the VOS approach that significantly reduces the requirements in terms of number of messages and preserves the computational efficiency, modularity and privacy-preserving characteristics of the VOS approach for EV charging control.

REFERENCES