

Heterogeneous Communication Virtualization for Distributed Embedded Applications

Thin H. Pham^{*}, Shanker Shreejith[†], Sebastian Steinhorst[‡], Suhaib A. Fahmy[§], and Samarjit Chakraborty[¶]

^{*} Department of Computer Science, University of Bristol.

[†] Department of Electronic and Electrical Engineering, Trinity College Dublin, The University of Dublin.

[‡] Department of Electrical and Computer Engineering, Technical University of Munich.

[§] King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia.

[¶] Department of Computer Science, University of North Carolina at Chapel Hill.

Abstract—Distributed embedded applications (DEAs) are typically implemented on diverse embedded nodes interconnected through communication network(s) to exchange data and control information to achieve the desired functionality. Conventional approaches of utilising a single large-bandwidth link in a distributed system are not efficient in large DEAs owing to diverse requirements and factors like cost, reliability, scalability and criticality, among others. Heterogeneous communication is a promising approach in DEAs, where the diverse nature of underlying protocols (wired/wireless, synchronous/asynchronous, multiple access modes and others) can be leveraged to meet such requirements, in addition to the benefits like aggregated bandwidth and robustness. However, utilising them ‘directly’ places significant complexity on the application as it needs to dynamically evaluate the channels and utilise different protocol structures for each case. Virtualising the communication channels would present a unified interface to the application by abstracting away low-level details, similar to virtualisation applied in compute architectures. However, unlike architecture virtualisation, virtualising heterogeneous communication particularly for resource-constrained device networks involves unique challenges imposed by the physical (wired/wireless) and logical domains (limited-bandwidth, small payload, protocols, channel access schemes, etc.), which needs to be concurrently evaluated to optimise the communication system. This paper presents a model and an optimal transmission strategy as the proof of concept for deploying heterogeneous communication in DEAs. The model is described at an abstracted level while capturing transmission parameters of multiple channels, which are then optimised to meet the application’s communication requirements. The model and the optimisation method are validated through simulation and a practical case study.

I. INTRODUCTION

Distributed embedded applications (DEAs) running on resource-constrained devices are gaining increasing traction in many application domains such as smart-grid [1], [2], automotive [3], networked robotics [4], and advanced metering infrastructure [5]. Communication is a key component of DEA architectures and is often enabled through wired and/or wireless transmission protocols like CAN, Ethernet, Bluetooth, and WiFi among others. While wired links (e.g., CAN) offer reliable communication, cost efficiency and relatively high bandwidth, they place limitations on applications, namely, flexibility and scalability. On the other hand, wireless communication is becoming more cost effective, scalable and robust. Bandwidth aggregation for devices with multiple wireless interfaces has been extensively studied in the literature [6], [7]. Likewise, time-sensitive networking (TSN), an enhanced Ethernet-based network standard, is an emerging approach for high-performance industrial and automotive networks [8]. TSN aims at delivering low and guaranteed transmission

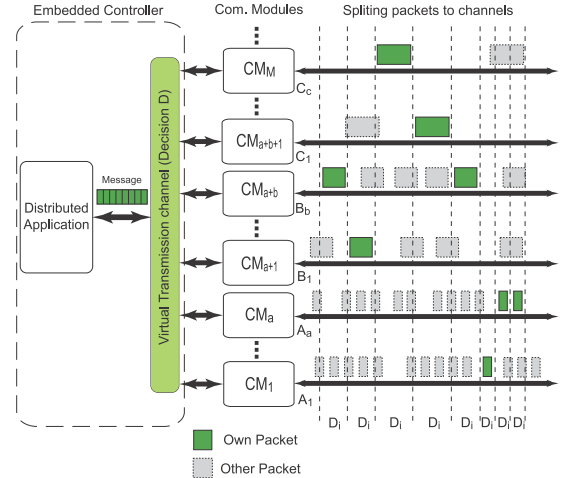


Fig. 1: Heterogeneous communication architecture: EC splits a message to transmit its packets over multiple shared channels according to optimal decision.

latency which is of key significance in distributed safety-critical applications. However, wireless networks and extensions like TSN cannot to cater to all requirements of low latency, low energy, high reliability, fault resilience in safety-critical embedded systems. Thus, such extensions are not expected to replace traditional wired communication in such DEAs in the near future [3]. Also, in many cases, a single (wired/wireless) transmission mode cannot satisfy the diverse range of requirements of DEAs in terms of energy consumption, bandwidth, reliability, real-time response and payload capability. Consequently, next generation networks for DEAs is expected to leverage a combination of wired and wireless transmission modes through a heterogeneous communication interface, which can utilise techniques like virtual pooling of channels to maximise the performance of each channel. While the physical transmission media (wired/wireless) forms the basis of heterogeneity in this case [9]–[11], numerous logic-layer factors related to the underlying protocol (like transmission model, packet size, encoding schemes and medium access schemes) contribute to the nature of diversity.

Communication virtualisation: Computing systems have utilised virtualisation to abstract the diversity of underlying architecture, allowing software to run seamlessly on different hardware platforms using standardised APIs. In high-performance communication, network virtualisation applies the same principles by abstracting the underlying heterogene-

ity of the communication networks [12]–[14], allowing the application designers to use simple interfaces and standardised APIs for communication tasks. However, existing network virtualisation models and communication virtualisation techniques are designed for general-purpose computing systems and are relatively complex to be deployed on microcontroller-based nodes and diverse bandwidth-constrained links used in DEAs. This paper proposes a virtualisation model and a unified interface that abstracts underlying heterogeneous physical links. The model and low-level interface(s) are designed as a lightweight extension suitable for integration with embedded applications targeting resource constrained devices and network links.

Fig. 1 illustrates an overview of a heterogeneous communication architecture that supports an embedded virtual transmission model. Each distributed node (i.e., embedded controller (EC)) has many channels controlled by the communication modules (CM), which can be used to transmit messages as required by the DEA. The model provides a framework to perform the transmission optimisation of messages in DEAs utilising heterogeneous communication. The main contributions of this paper are as follows:

- A virtual transmission model capturing the communication physical parameters for heterogeneous communication is defined, which provides a unified and convenient interface for DEA developers.
- An optimised transmission strategy is presented at the middleware layer, which takes the application’s communication requirements and physical link parameters as inputs to determine the optimal transmission strategy for the messages
- A simulation framework is developed to validate the model and the optimised transmission method.
- A case-study of efficient transmission via hybrid communication for resource-constrained networks is presented using the optimised transmission method.

II. HETEROGENEOUS COMMUNICATION MODEL

We propose a model in which multiple links including wired/wireless links with different bandwidths and physical interfaces are employed through a heterogeneous architecture. The model takes into account physical parameters that are often used for low-cost and bandwidth-constrained communications in distributed embedded systems and applications. An abstract view of the virtual transmission model presented in this paper is shown in Fig. 2. The model enables connectivity between an EC and multiple CMs via standard interfaces like UART and SPI. Each communication link is controlled by a CM through DLL and PHY layers. A middleware component distributes and aggregates the data message from DEAs to/from the relevant CM. It integrates the definition of the communication layers and exposes them to the upper layers (application) through a standard set of APIs. A new physical link added through a new CM can be seamlessly integrated into the DEA simply by defining its properties in the middleware component. Based on the configuration of the communication interfaces enabled by the application, the middleware component will split/combine information to/from the individual CMs for transmitting and receiving application data. This model enables data from a transmitter to be sent

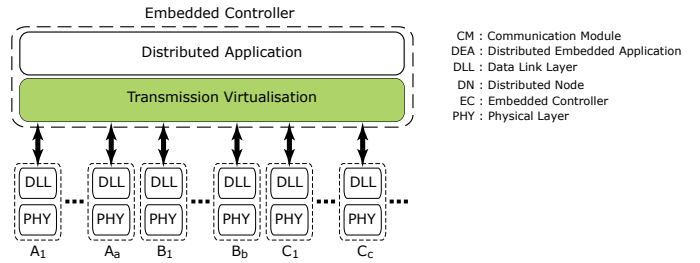


Fig. 2: Virtual transmission abstraction for communication between a controller and heterogeneous communication modules.

to the receiver(s) over multiple links, each with one or more channels. Also, the abstraction presents the EC with a unified interface allowing it to extend its communication bandwidth without altering the application. Based on the characteristics of each link (part of the model), the EC makes optimal transmission decisions to meet the DEA requirements.

A. Communication Link Characterisation

CMs have a typical set of physical parameters for a link such as bandwidth BW (Kbps), physical frame size FS (Byte), medium access mechanism, as well as the details regarding the interface between the CM and its controller such as interface rate (IR), i.e., Baud rate (Kbps), and data buffering. However, the DEA specifies different requirements that must be achieved through combining CM capabilities. To model the heterogeneous communication at the middleware layer, some parameters are defined to characterise the properties of the links at the higher DEA abstraction. They are described as follows: a) *Channel Traffic (CT)* represents channel usage status (in %). b) *Message Size (MS)* is the total data (bytes) of an application message that is periodically transmitted. c) *Transmission Delay (TD)* is the time (ms) required to transfer a message from a node to another. d) *Packet Size (PS)* is the maximum amount of data (bytes) that can be packed into one physical packet. A large application message may be split into multiple packets. e) *Packet Latency (PL)* is the time (ms) required to transmit a data packet. f) *Transmitter Throughput (TT)* is the amount of data sent out by a controller to its CM in a time unit (Kbps). g) *Transmission Reliability (TR)* is the success rate (%) sending out a packet. h) *Packet transmission Energy (PE)* is the energy (J) consumed by a controller and its CM to send a packet within the duration of PL .

B. Heterogeneous Communication Cost Model

Sending packets optimally through a heterogeneous network is a run-time decision making problem of choosing the optimal link(s) from the heterogeneous network, where the available links have different physical specifications and varying channel performance abilities, while also considering the channel traffic conditions. The proposed model uses the virtual pool of available channels to represent communication ability and performance in terms of latency, throughput, reliability, and energy so that the middleware task can optimise the transmission component based on the model, i.e., select the best links at the given time for distributing a message from the application. The model incorporates the characteristic of the physical links and the status of the channels to model the transmission cost. Based on the link characterisation discussed before, we define

the set of channel ability vectors in terms of the application requirements for the heterogeneous architecture as follows:

$$\mathbf{CA} = \{ \overline{TT}^T, \overline{PL}^T, \overline{TR}^T, \overline{PE}^T \} \quad (1)$$

Note that \overline{A}^T denotes the transpose of vector \overline{A} . \overline{TT} is a transmission throughput vector in which each element represents the throughput of a channel that depends on the corresponding module interface, module buffer usage, and channel traffic. Likewise, \overline{PL} represents a packet latency vector where each element depends on the corresponding transmission throughput, channel traffic and the link physical specification. \overline{TR} is a transmission reliability vector in which each element depends on the corresponding channel traffic and the link reliability characteristic. Finally, \overline{PE} is the packet transmission energy vector where each element depends on the corresponding channel traffic and the link power rate. These vectors have the length of M which equals to the number of channels. The channel ability vector \mathbf{CA} collates the individual ability vectors to represent the channel's ability to support transmission requests from the application.

Next, we define the transmission cost of using one channel to transmit a packet as:

$$Cc_j = kl \cdot CL_j + kt \cdot CT_j + kr \cdot CR_j + ke \cdot CE_j \quad (2)$$

where kl , kt , kr , and ke are the relative cost coefficients for latency, throughput, reliability, and energy consumption, respectively. The set of coefficients are design options depending on specific DEA's requirements. For example, an application which requires high throughput will have a set of coefficients in which kt is large compared to others. CL_j , CT_j , CR_j , CE_j are the partial costs of transmission with respect to latency, throughput, reliability, energy consumption, respectively. They are defined as follows:

$$\begin{aligned} CL_j &= \frac{PL_j}{PL_{min}}; & CT_j &= \frac{TT_{Max}}{TT_j} \\ CR_j &= \frac{1 - TR_j}{TR_j}; & CE_j &= \frac{PE_j}{PE_{min}} \end{aligned} \quad (3)$$

where PL_{min} , TT_{Max} , PE_{min} denote the smallest latency, largest throughput, and smallest energy consumption of channels, respectively. PL_j , TT_j , TR_j , PE_j are latency, throughput, reliability, and energy consumption of the channel, respectively, at the current condition. Ideally, the smallest latency cost equals 1 when $PL_j = PL_{min}$, the smallest throughput cost equals 1 when $TT_j = TT_{Max}$, the smallest energy cost equals 1 when $PE_j = PE_{min}$, and the smallest reliability cost equals 0 when $TR_j = 100\%$. Note that when a packet is sent through a channel, even if the transmission is optimised, the transmission still entails a non-zero optimal cost in terms of latency, throughput and energy. In case of a fully reliable communication system (i.e., a link reliability of 100%), the link guarantees that every packet using the corresponding CM will be successfully transmitted, leading to zero reliability cost.

Moreover, we consider the case that the application sends a message that can be divided into N packets due to the constrained payload of channels. Each packet is sequentially sent to one channel according to the corresponding decision vector \overline{D}_i ($i \in \overline{N} := \{1, 2, \dots, N\}$). A decision vector is defined as $\overline{D}_i = [d_1, d_2, \dots, d_M]^T$, where d_j represents a

channel selection. If a channel is used, the corresponding d_j equals 1 otherwise equals 0. Packets can be sent to different channels with different packet sizes (PS). Therefore, the size of each packet depends on the the corresponding decision vector, $PS_i = f_{\overline{D}_i}$. The number of packets is not pre-configured, allowing the system to dynamically chose channels (i.e., packet sizes) to optimise transmission of each message; however, the different packet size makes finding the global optimal decision more difficult. We employ N as the maximum number of packets into which a given message is divided. It equals $\frac{MS}{PS_{min}}$, where PS_{min} is the smallest packet size of the channel. A decision matrix $\mathbf{D} = [D_1, D_2, \dots, D_N]$ for a message transmission is defined based upon the decision vector \overline{D}_i .

We consider that in dynamic channel conditions, the transmission ability of channels varies according to the previous transmission decision(s). The transmission ability of channel defined in Eq. (1), when expanded further, allow the CA's to be dependent on the message distribution, which in turn, depends on the status of a channel (i.e., channel traffic (\overline{CT})) and the previous transmission decision of the message \overline{D}_{i-1} , as shown in Eq. 4. The analysis of the dependent functions ($f_{(.)}$) is used to determine the effect on the transmission ability of a channel at the time of decision \overline{D}_i . Based on this definition of CA, we can explore an optimisation algorithm for the message distribution decision. The analysis builds on the definition of parameters discussed before:

$$\begin{aligned} TT_{i,j} &= f_{TT}(\overline{CT}, \overline{D}_{i-1}); & TR_{i,j} &= f_{TR}(\overline{CT}) \\ PL_{i,j} &= f_{PL}(\overline{CT}, \overline{D}_{i-1}); & PE_{i,j} &= f_{PE}(PL_{i,j}) \\ \Rightarrow \mathbf{CA}_i &= \{ \overline{TT}_i, \overline{PL}_i, \overline{TR}_i, \overline{PE}_i \} = f_{CA}(\overline{CT}, \overline{D}_{i-1}) \end{aligned} \quad (4)$$

The transmission throughput ($TT_{i,j}$) and the latency ($PL_{i,j}$) of a channel are the elements of the current transmission throughput vector (\overline{TT}_i) and packet latency vector (\overline{PL}_i), respectively. At the time of decision \overline{D}_i , the current transmission throughput vector (\overline{TT}_i) and packet latency vector (\overline{PL}_i) depend on the previous decision and channel traffic status. In the reliability vector (\overline{TR}_i), each element represents the reliability of a channel ($TR_{i,j}$) and depends on the channel traffic status. Similarly, \overline{PE}_i is the packet transmission energy vector, wherein each element ($PE_{i,j}$) represents the energy to send a packet via the currently selected channel which depends on the packet latency. Generalising the definition, the channel's current ability \mathbf{CA}_i is a combination of all ability vectors and varies according to the previous transmission decisions and channel traffic conditions. The cost of sending a packet is counted at the time when the packet is sent to a given channel determined by the decision \overline{D}_i . The cost depends on the decision vector \overline{D}_i expressed as follows:

$$\begin{aligned} C(\overline{D}_i, \mathbf{CA}_i) &= kl \left(\frac{\overline{PL}_i}{PL_{min}} \times \overline{D}_i \right) + kt \left(\frac{TT_{Max}}{TT_i} \times \overline{D}_i \right) + \\ &\quad kr \left(\frac{100\%}{TR_i} \times \overline{D}_i \right) + ke \left(\frac{\overline{PE}_i}{PE_{min}} \times \overline{D}_i \right) \end{aligned} \quad (5)$$

C. Transmission Optimisation

The goal of the optimisation task is to find the best channel(s) to deliver individual data packet(s) at any given time. The cost of sending a distributed message through the

network depends on the decision matrix \mathbf{D} . Consequently, the optimisation approach can be expressed as follows:

$$MC(\mathbf{D}, \mathbf{CA}_i) = \sum_{i=1}^N C(\overline{D}_i, \mathbf{CA}_i),$$

$$\hat{\mathbf{D}} = \underset{\mathbf{D} \in \mathbf{B}_{M \times N}}{\operatorname{argmin}} MC(\mathbf{D}) \quad (6)$$

The optimised decision for sending a message through the heterogeneous network is determined by the decision matrix \mathbf{D} within a space $\mathbf{B}_{M \times N}$ targeting the lowest MC . $\mathbf{B}_{M \times N}$ is represented by a binary matrix which has M rows, N columns and each column has at most one element equalling 1.

D. Algorithm

Having built the model and derived the optimisation approach, we now present an algorithm to determine the optimised message transmission. The algorithm is based upon a recursive function expressed as follows:

$$oC(MS, \mathbf{CA}) = \min_{\overline{D}_i \in \mathbf{B}_{M \times N}} \{C(\overline{D}_i, \mathbf{CA}) + oC(MS', \mathbf{CA}')\} \quad (7)$$

Algorithm 1 Transmission optimisation algorithm.

```

1: Function OptimisedTransmission( $MS$ )
2: init( $CA$ )
3: [ $oC$ ;  $oD$ ] = OptimisedCost( $MS, CA$ )
4: return  $oD$ 
5: EndFunction
6:
7: Function OptimisedCost( $MS, CA$ )
8: if  $MS > 0$  then
9:    $oC = \infty$ 
10:  for Each  $D_i \in \mathbf{B}_{M \times N}$  do
11:     $C_i = \operatorname{Cost}(D_i, CA)$ 
12:    [ $MS'$ ;  $CA'$ ] = update( $MS, CA, D_i$ )
13:    [ $MC$ ;  $D$ ] = OptimisedCost( $MS', CA'$ )
14:    if  $oC > C_i + MC$  then
15:      [ $oC$ ;  $oD$ ] = [( $C_i + MC$ ); { $D_i, D$ }]
16:    end if
17:  end for
18:  return [ $oC$ ;  $oD$ ]
19: else
20:  return [0; {}]
21: end if
22: EndFunction = 0

```

This function implies that the optimised cost of transmitting a message MS is the minimal summation of packet transmission cost with \overline{D}_i and the optimised transmission cost for the rest of message MS' . \mathbf{CA} is changed to \mathbf{CA}' to differentiate the current packet transmission from the previous transmissions. \mathbf{CA}' is used to determine the next transmission decision for sending the subsequent packet. The pseudo-code implementing the algorithm is given in Algorithm 1. The *optimised transmission* function computes the initial condition of the system, then calls the *optimised cost* function. The *optimised cost*, a recursive function, determines the optimal cost of message transmission and the corresponding decision matrix \mathbf{D} according to (7). The function loop in *line 10* repeats the computation for each channel. C_i represents the packet transmission cost computed according to Eq. (5). The transmission cost of the rest of the message is calculated in *line 13*. The “*if*” statement in *line 14* finds the minimal message transmission cost.

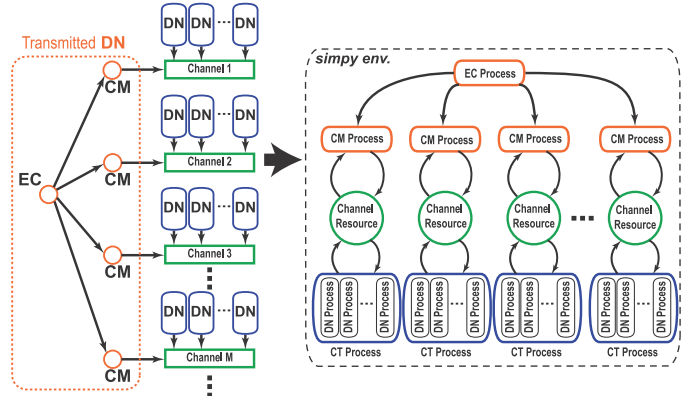


Fig. 3: The simulation architecture in which a controller of one node has multiple communication modules which send data to their own channel that is shared with other nodes.

III. VALIDATION USING SIMULATION AND ANALYSIS FRAMEWORK

In this section, a simulation framework aiming to evaluate the optimised algorithm performance and to validate the transmission model is presented. A mathematical analysis is also developed to calculate the transmission performance based on channel status and system architecture. Results of the proposed optimised method are compared to those of conventional methods in both simulation and analysis in Section III-C.

A. Simulation Environment Setup

The simulation is built using *SimPy* which is a process-based discrete-event simulation framework based on standard Python. *SimPy* also simplifies the modeling of shared channel communication. The architecture on the left of Fig. 3 illustrates the topology of the simulator in general. Considering distributed nodes, an *EC* employs multiple *CM*s to transmit messages. Each *CM* controls the transmission of packets to a channel (i.e shared bus, wireless channel) that is shared with other distributed nodes (*DN*s). The *SimPy* simulation environment is shown in the right of Fig. 3. In the simulation, the functions of *EC*s and *CM*s are performed in separate processes so that they can run concurrently. Shared resources are employed as shared channels. Multiple processes are created for simulating channel traffic (*CT*) with multiple *DN*s that cause the occupation on the shared channels.

For the sake of simplicity, without loss of generality, a heterogeneous communication system is considered with two

TABLE I: Channel parameters for simulation.

	PS (Byte)	IR (Kbps)	BW (Kbps)
Channel 1	24	500	250
Channel 2	8	1000	100

TABLE II: The packet Latency, PL , of channels at different channel traffic, CT , conditions.

CT (%)	10	20	30	40	50	60	70	80	90
$PL1$ (ms)	1.68	1.80	1.98	2.26	2.57	2.99	3.53	4.35	5.24
$PL2$ (ms)	1.25	1.38	1.55	1.75	2.09	2.50	3.15	4.15	5.75

channels with diverse parameters, as shown in Table I. The simulation is realised as follows:

- First, we investigate the performance of each channel separately as channel traffic varies, with packet latency results shown in Table II.
- Second, according to the previous investigation results, the optimal algorithm presented in Section II-D is employed to determine the optimised transmission, with the goal of minimizing message latency and maximizing throughput.
- Finally, the optimised transmission on a distributed node is investigated in both the simulation framework and the mathematical analysis. The enhancement provided by the proposed approach is compared to normal transmissions.

B. Mathematical Analysis

A mathematical analysis is presented to determine the message latency and throughput according to a transmission decision. The analysis calculations are based on the status of channels and the system architecture. Consider the system architecture in Fig. 3 in which the *EC* is to distribute the packets of a message to *CMs*. The *CMs* can transmit their packets sequentially and can perform their transmissions in parallel with other *CMs*. The *CMs* can only buffer one packet at a given time. We seek to investigate the transmission performance when the *EC* distributes packets to the *CMs*. Continuously sending packets to the same *CM* can reduce transmission performance because the *EC* must wait for the *CM* to finish sending the previous packet. On the other hand, allocating the task of sending sequences of packets to different *CMs* can increase transmission efficiency because the *CMs* can send packets in parallel over multiple channels.

Fig. 4 shows the timing diagram for the process of sending packets to multiples *CMs*. The horizontal dimension shows the time when the *EC* sends the packets to the *CMs*. D_i denotes the i th packet transmission. The vertical dimension illustrates the different *CMs* sending packets over their channels. As can be seen in Fig. 4, the first case considered is shown in the first two transmissions (i.e., D_0, D_1) that employ the same channel (i.e., Ch_0) and packets are sent to the channel sequentially. In this case, the *EC* sends the first packet to the *CM* within a duration of t_{tp} . Then the *CM* transmits the packet to the channel within a duration of t_{ch} while the *EC* waits for the duration t_{ba} that is required for the *CM* to finish transmitting the first packet on the channel. Another case considered is sending packets in different channels (e.g., Ch_a, Ch_b). This case is illustrated in the last three transmissions (i.e., $D_{n-3}, D_{n-2}, D_{n-1}$). After sending a packet to CM_a on the channel Ch_a at the transmission D_{n-3} , the *EC* can immediately send a packet to another channel Ch_b at the transmission D_{n-2} if CM_b 's buffer is empty. If the *EC* wants to send the next packet to CM_a at the transmission D_{n-1} , the *EC* has to wait for CM_a completely transmit its previous packet on the channel Ch_a . Based on this analysis of the timing diagram (in Fig. 4), the throughput and latency of transmitting a message can be evaluated through mathematical analysis after the optimised decision is determined as follows:

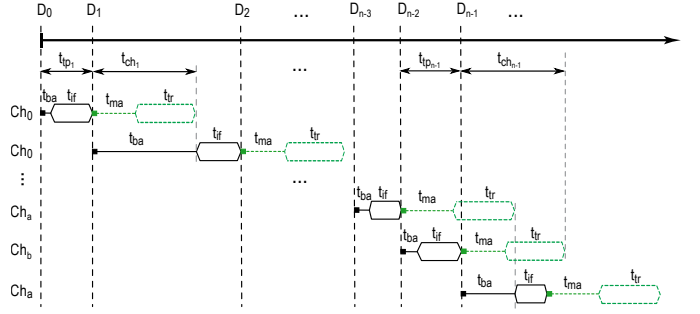


Fig. 4: Timing diagram showing messages transmitted over multiple channels.

$$MT = \frac{MS}{\sum_{i=1}^N t_{TP_i}}$$

$$ML = \sum_{i=1}^N t_{TP_i} + \max(t_{CH_N}; t_{CH_{N-1}} - t_{TP_N}; \dots)$$

where

$$t_{TP_i} = \frac{PS_i}{\overline{TT_i} \times \overline{D_i}}$$

$$t_{CH_i} = (\overline{PL_i} \times \overline{D_i}) - t_{TP_i} \quad (8)$$

The throughput of a message transmission is the message size (MS) divided by the time taken by the *EC* to distribute all packets of the message to the *CMs*. The message latency specifies the time between the *EC* sending the first packet until the last packet is received by the receiver's *CM*.

C. Simulation Results

The simulations are performed with the communication presented in Section III-A, transmitting 64 byte messages over the network with the two channels. Fig. 5a shows the simulation results in terms of throughput when data traffic on channel 2 is increased from 10% to 90%. The curves correspond to different steady-state data traffic in channel 1. $*_{20}, *_{50}, *_{80}$ suffixes denote 20%, 50%, 80% steady-state traffic occupancy on channel 1, respectively. Sim_* , and Ana_* prefixes indicate the transmission results of the proposed method on both channels in simulation, and from the model, respectively. The $Ch1_*$ prefix represents transmissions using only channel 1 and $CH2$ denotes transmitting the message only on channel 2.

The results in Fig. 5a show that the proposed method Sim_* significantly improves the throughput, especially in the case of low data traffic (20%) in channel 1, compared to a typical method that would use only channel 2. When channel 1's data traffic is increased to 80%, the gap between Sim_* and $CH2$ is reduced due to the higher network utilisation. Sim_* also outperforms the corresponding typical method that uses only channel 1 $Ch1_*$. Increasing the channel 2 data traffic reduces the gap between Sim_* and the corresponding $Ch1_*$ due to increased congestion. Additionally, the results of simulation and analysis are in close agreement except for the cases of large traffic in channel 2 (i.e., 80%, 90%). When data traffic reaches 90%, transmissions are almost jammed, leading to an increased deviation between the simulation and model.

Similarly, Fig. 5b illustrates the message latency results. We can observe that the proposed method (Sim_*) achieves

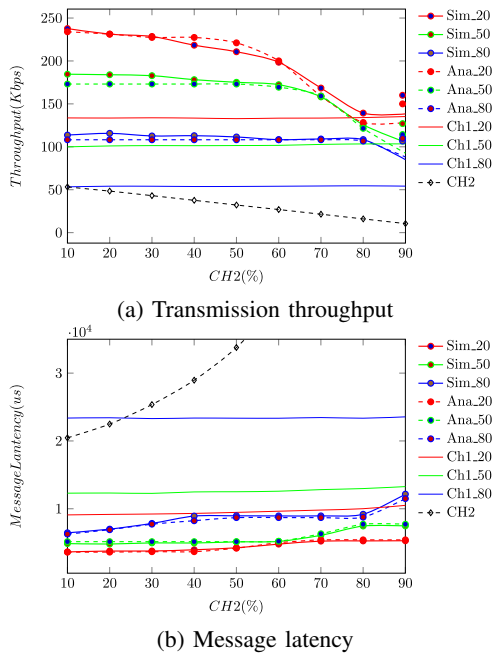


Fig. 5: Performance comparison of transmitting 64 byte messages in varying channel data traffic.

lower message latency compared to the typical method of transmitting on a single channel (*CH2*). Also, we can see that the results of the simulation and model are in agreement. The simulation and analysis results in Fig. 5a and Fig. 5b show that the optimised algorithm achieves significant improvements in throughput and latency compared to the case of the typical transmissions using either channel 1 or channel 2 alone.

The optimised transmission is further investigated in the case of transmitting 64 byte messages through the network when the channel utilisation varies dynamically. The performance of the proposed algorithm is compared not only to the typical transmission using a single channel but also to a greedy approach of two-channel transmission that employs both channels alternately. We assume that the channel traffic can be continuously sensed at the CMs before a transmission decision is made for each message. Based on the sensed traffic, the optimised decision is determined for message transmission.

Fig. 6a and Fig. 6b show the results in terms of the transmission throughput and message latency, respectively vs. channel traffic varies over time. The comparisons are done between the proposed method *Pro*, the baseline methods that use single channel transmission and the greedy approach. *N1*, *N2*, *N3* denote the transmissions using channel 1, using channel 2, and using a greedy algorithm (employing both channels alternately), respectively. Notably, the proposed method achieves lower latency, higher throughput and, hence, considerably outperforms the other approaches, particularly in the case where the traffic of a channel suffers heavy congestion compared to the other.

These simulation results show the significant improvement offered by the proposed hybrid communication algorithm in terms of throughput and message latency. They also validate the accuracy of the model and the mathematical analysis compared to simulation.

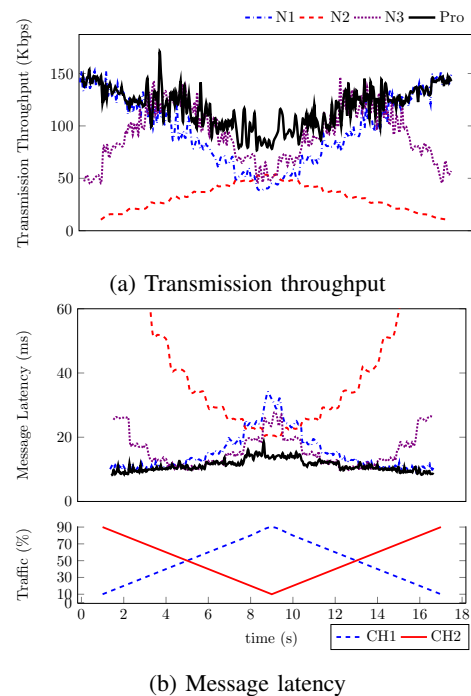


Fig. 6: Performance comparison in varying traffic over time.

IV. CASE STUDY AND EXPERIMENTS

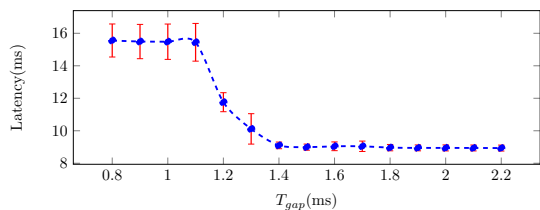
A. CAN/Zigbee Hybrid Communication Case Study Scenario

The application of the proposed model is now investigated in a practical hybrid communication case study. Hybrid communications are often targeted for low-latency, resource-constrained (embedded devices) networks such as sensor networks, industrial automation, robotics, and cost-efficient automotive. For example, the work in [10] presents a robust sensor network based on a hybrid architecture in which the communication between nodes is through a wired link; a wireless link is employed as a backup link for any node or wired link failure. Other related works investigate the combination of wired and wireless transmission to improve the performance of embedded communication [11], [15], [16].

We consider a system where a typical wired network suffers from increasing congestion and risks of physical damage, leading to performance and reliability degradation. An additional wireless link is added in this case to increase robustness against the failures and improve communication performance. The experiments are set up using commercial off-the-shelf microcontroller boards as *ECs* that includes two *CMs* to exchange data via controller area network (CAN) [17] and Zigbee channels. The proposed hybrid communication is implemented at a middleware layer on the *EC* as discussed in Section II. This provides an abstract interface for embedded applications to transmit and receive over the channels. We perform the evaluation as follows: First, hardware experiments are set up and evaluated to obtain the experimental parameters of each channel separately. Then, based on the measured results for the experimental parameters, the hybrid algorithm presented in Section II-D is employed to determine the optimised transmission to minimise message latency.

TABLE III: Physical specifications of the channels.

Channel	Interface	IR (Kbps)	BW (Kbps)
Zigbee	UART	115.2	250
CAN	SPI	500.0	250

Fig. 7: Mean and standard deviation of latency sending 24 byte messages with varying t_{gap} durations.

B. Determining Experimental Parameters

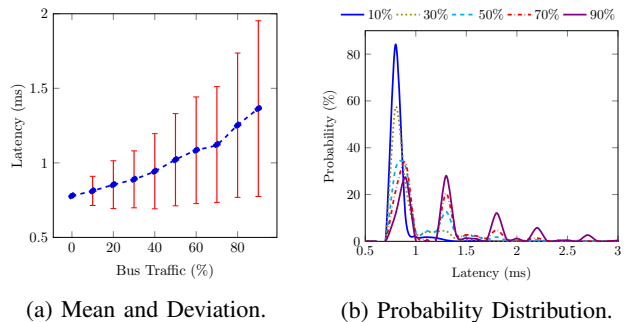
Each distributed node is set up as an Arduino Due [18] board consisting of a CAN shield and an Zigbee shield using SPI and UART interfaces, respectively. The physical specifications of the channels are shown in Table III. We consider that CAN is a conventional wired channel suffering from deteriorating performance and an Zigbee channel is added to improve the network performance. CAN is typically used to transmit small data quantities (maximum data field size of 8 bytes) in packets. We assume that the Zigbee channel has no congestion and the Zigbee packet latency has a fixed distribution regardless of the channel traffic, while in case of the CAN channel, increasing the channel traffic results in increased packet latency. This setup enables to investigate and quantify the Zigbee and CAN network characteristics at different network conditions, which are presented in the following subsections. The experimental parameters presented are averaged over 1000 transmission measurements in each case.

a) Zigbee Characteristics: We determine the characteristics of the Zigbee channel by measuring Zigbee transmissions in a free channel. We observe that the latency of Zigbee transmissions is spread in a narrow range of time. This can be attributed to the packet processing pipeline within the Zigbee communication module; the internal buffers within the Zigbee controller allows subsequent packets to be sent to the CM before the current packet is fully transmitted. However, we also observe that when packets are sent to the CM continuously, the packet latency is affected due to the finite buffer size within the CM. We also evaluate the impact of varying the gap duration t_{gap} on the packet latency. Fig. 7 shows the measurement results. When t_{gap} is decreased below 1.4 ms, the mean and deviation of the latency increase. Transmission latency is stable at $t_{gap} > 1.4$ ms. Therefore, we take the t_{gap} into account to determine packet latencies. These measurements on the Zigbee channel result in the characteristics in Table IV, which are used in the proposed hybrid communication model and subsequent evaluation of the optimisation scheme.

b) CAN Characteristics: Next, we investigate the characteristics of a CAN channel by measuring average packet latency with respect to the data traffic of the CAN network. Each CAN packet has a size of 8 bytes. To set a given data traffic usage on the CAN bus, we set up two interfering nodes

TABLE IV: Measured characteristics of Zigbee channel.

Interface	Baud rate	Packet size	t_{gap}	Packet Latency
UART	115.2 Kbps	24 bytes	1.4 ms	6.9 ms



(a) Mean and Deviation.

(b) Probability Distribution.

Fig. 8: Packet Latency in CAN point-to-point communication with varying traffic (1000 measurements).

that generate random packets (that are higher priority than the packets sent by the transmitter node) to the receiver node. Measurements of mean packet latency and deviation are shown in Fig. 8a. We further investigate the probability distribution of the CAN packet latencies. Fig. 8b illustrates this for CAN point-to-point communication at various traffic values on the CAN bus. When data traffic is increased to 90%, the maximal packet latency is significantly increased. These measurements on the CAN bus result in the characteristics in Table V, which are applied in the proposed hybrid communication model.

C. Hybrid Communication Performance

With the parameters for both communication systems determined, we investigate the impact of the hybrid Zigbee/CAN communication model to enhance the communication performance over the single CAN network using our case study. From the analysis before, we observed that the Zigbee latency is much higher compared to the CAN bus; hence, the end-to-end latency of sending messages using only Zigbee or using a greedy algorithm will be higher than the latency of sending messages using only CAN bus at low network utilisation. Thus, the optimisation approach aims to use the hybrid communication while minimising the overall latency of the system. We consider the transmission latency associated with a moderately sized 80-byte message for our experiments. The latency results of sending messages using only CAN bus are considered as the baseline for comparison to the results achieved by the proposed solution. Fig. 9 shows the plot of the message latency measured for the CAN network and the proposed hybrid approach. We see that the hybrid approach can achieve considerable improvement in message latency compared to a standard CAN-only transmission model, particularly in case of high network utilisation (higher data traffic). We also observe that the hybrid approach could cause higher deviation around the mean latency value owing to the differences in the underlying networks; however, the worst case observed latency is significantly below the mean value observed on the CAN-only transmission. We further determine the distribution of the latency on the hybrid approach under varying bus load conditions on the CAN network, which is illustrated in Fig. 10. It clearly shows that the distribution

TABLE V: Packet latency (PL) versus CAN bus traffic, CT .

CT (%)	10	20	30	40	50	60	70	80	90
PL (us)	794	835	872	918	985	1056	1117	1221	1390

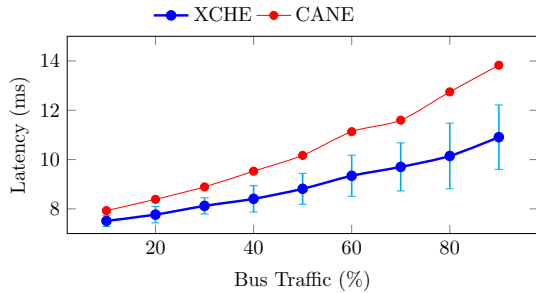


Fig. 9: Transmission latency comparison with respect to bus traffic for an 80 byte message (1000 measurements).

spreads the latency across multiple messages at higher loads leading to the lower worst case latency, as observed in Fig. 9.

V. CONCLUSION

Summary: This paper presents a model and unified interface for heterogeneous communication for DEAs, addressing the constraints of latency, throughput, reliability, and energy. A model for heterogeneous networks at a high level of abstraction was proposed to optimise message transmission over heterogeneous channels. A simulation framework has been built to validate the proposed model and the optimisation algorithm in comparison to conventional approaches. The proposed model and optimised algorithm have been also investigated using a practical hybrid communication case-study. It shows that the hybrid transmission achieves considerable improvement compared to the conventional single-link communication, particularly in the case of high traffic load. In the future, we aim to extend the model to consider additional parameters like message criticality to address the need for safety-critical systems.

Future work: Although our results are largely positive, we view the proposed approach as a first step rather than a complete solution. As such, we aim to explore the following potential extensions for our future work. *First*, the presented algorithm is based on a recursive search that may cause unexpected computational complexity. One potential direction for future work is to investigate a multi-objective optimization method that may allow an efficient algorithm to optimise the objectives of the proposed model simultaneously. *Secondly*, even though the model considers reliability and energy consumption in its cost function, the simulation and the case study limit the investigation to throughput and latency aspects. Despite the obvious fact that multi-link communication is more resilient against single link alternatives, it is interesting to investigate the reliability and energy efficiency improvements that could be achieved by the proposed model. *Finally*, the data transmission in the proposed model considers dynamic channel conditions to optimise transmission conditions. This is based on an assumption that the channel traffic can be continuously sensed/estimated at nodes. We aim to investigate mechanisms for accurately estimating channel traffics and how estimating accuracy affects transmission optimisation.

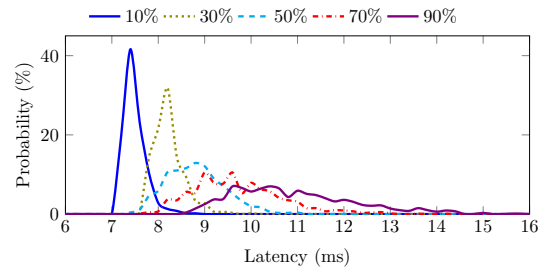


Fig. 10: Message latency distribution of the hybrid communication with varying bus traffic for an 80 bytes message.

ACKNOWLEDGMENT

S. Chakraborty is supported by the NSF Grant #2038960.

REFERENCES

- [1] M. Allalouf, G. Gershinsky, L. Lewin-Eytan, and J. Naor, "Smart Grid Network Optimization: Data-Quality-Aware Volume Reduction," *IEEE Systems Journal*, vol. 8, no. 2, pp. 450–460, June 2014.
- [2] H. Geng, S. Li, C. Zhang, G. Yang, L. Dong, and B. Nahid-Mobarakkeh, "Hybrid Communication Topology and Protocol for Distributed-Controlled Cascaded H-Bridge Multilevel STATCOM," *IEEE Transactions on Industry Applications*, vol. 53, no. 1, pp. 576–584, Jan 2017.
- [3] S. Tuohy, M. Glavin, C. Hughes, E. Jones, M. Trivedi, and L. Kilmartin, "Intra-Vehicle Networks: A Review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 534–545, April 2015.
- [4] M. Chen, J. Wan, S. Gonzalez, X. Liao, and V. Leung, "A Survey of Recent Developments in Home M2M Networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 98–114, 2014.
- [5] S. Matsuzawa and Y. Takahashi, "Performance Analysis of Advanced Metering Infrastructure with Multihop Hybrid Communication System," in *Proc. Intl. Conf. on Queueing Theory and Network Applications (QTNA)*, 2016, pp. 8:1–8:7.
- [6] S. A. Dehkordi, K. Farajzadeh, J. Rezazadeh, R. Farahbakhsh, K. Sandrasegaran, and M. A. Dehkordi, "A survey on data aggregation techniques in IoT sensor networks," *Wireless Networks*, vol. 26, no. 2, pp. 1243–1263, Feb 2020.
- [7] L. Ruan, M. P. I. Dias, and E. Wong, "Enhancing latency performance through intelligent bandwidth allocation decisions: a survey and comparative study of machine learning techniques," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 12, no. 4, pp. B20–B32, 2020.
- [8] L. Lo Bello and W. Steiner, "A Perspective on IEEE Time-Sensitive Networking for Industrial Communication and Automation Systems," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1094–1120, 2019.
- [9] O. Mirabella and M. Brischetto, "A hybrid wired/wireless networking infrastructure for greenhouse management," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 2, pp. 398–407, 2011.
- [10] L. Sportiello, "A methodology for designing robust and efficient hybrid monitoring systems," *International Journal of Critical Infrastructure Protection*, vol. 6, no. 3, pp. 132–146, 2013.
- [11] H. Sawada, H. Kuriyama, Y. Nakano, H. Mineno, and T. Mizuno, "Mutually complementary communication protocol for indoor sensor/actuator networks," in *Proc. IEEE/IPSJ Intl. Sym. on Applications and the Internet (SAINT)*, 2011, pp. 44–49.
- [12] S. B. Berman, "Methods, systems and apparatus for the virtualization of high performance networks," Oct. 16 2018, uS Patent 10,104,017.
- [13] R. Wang, J. Yan, D. Wu, H. Wang, and Q. Yang, "Knowledge-centric edge computing based on virtualized D2D communication systems," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 32–38, 2018.
- [14] L. Leonardi, M. Ashjaei, H. Fotouhi, and L. L. Bello, "A proposal towards software-defined management of heterogeneous virtualized industrial networks," in *Proc. Intl. Conf. on Industrial Informatics (INDIN)*, vol. 1, 2019, pp. 1741–1746.
- [15] F. Nouvel and P. Tanguy, "Flexible OFDM waveform for PLC/RF in-vehicle communications," in *Euromicro Conference on Digital System Design (DSD)*, 2012, pp. 463–468.
- [16] W. Hou, Y. Song, H. Wang, and A. Hunger, "Architecture and Performance of the Industrial Wired/Wireless Hybrid Networks with Multi-token," in *Proc. ACM Workshop on Mobile Sensing, Computing and Communication (MSCC)*, 2016, pp. 7–15.
- [17] "CAN specification version 2.0," *Rober Bousch GmbH, Postfach*, 1991.
- [18] Arduino-Website. Arduino Due. [Online]. Available: <https://www.arduino.cc/en/Main/arduinoBoardDue>