

Workgroup: COINRG  
Internet-Draft:  
draft-irtf-coinrg-use-case-analysis-00  
Published: 10 March 2023  
Intended Status: Informational  
Expires: 11 September 2023  
Authors: I. Kunze            K. Wehrle            D. Trossen  
          RWTH Aachen      RWTH Aachen      Huawei  
          M. J. Montpetit    X. de Foy  
          Concordia            InterDigital Communications, LLC  
          D. Griffin        M. Rio  
          UCL                UCL  
          **Use Case Analysis for Computing in the Network**

## Abstract

Computing in the Network (COIN) has the potential to enable a wide variety of use cases. The diversity in use cases makes general considerations challenging. In an effort to capture the breadth of possible scenarios, another COINRG document presents a selection of concrete use cases, each representing a broader set of settings.

This document analyzes the described use cases (and potentially further settings) to identify general aspects of interest across all use cases to steer future COIN discussions.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 11 September 2023.

## Copyright Notice

Copyright (c) 2023 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

## Table of Contents

- [1. Introduction](#)
- [2. Terminology](#)
- [3. COIN Use Cases Taxonomy](#)
- [4. Analysis](#)
  - [4.1. Opportunities](#)
  - [4.2. Research Questions](#)
    - [4.2.1. Categorization](#)
    - [4.2.2. Analysis](#)
  - [4.3. Requirements](#)
- [5. Security Considerations](#)
- [6. IANA Considerations](#)
- [7. Conclusion](#)
- [8. Informative References](#)
- [Authors' Addresses](#)

## 1. Introduction

The Internet was designed as a best-effort packet network that offers limited guarantees regarding the timely and successful transmission of packets. Data manipulation, computation, and more complex protocol functionality is generally provided by the end-hosts while network nodes are kept simple and only offer a "store and forward" packet facility. This design choice has shown suitable for a wide variety of applications and has helped in the rapid growth of the Internet.

COIN fundamentally changes these observations as it proposes adding meaningful compute functionality within the network and thus between the end-hosts. However, there is currently no consensus on what COIN is exactly and, thus, building solutions for COIN-related problems is non-trivial. In this context, [[USECASES](#)] presents a variety of use cases that were thought by the authors to represent meaningful applications of COIN. While [[USECASES](#)] proposes a taxonomy to structure the description of the different use cases, it does not provide further considerations. For example, it does not analyze the different use cases for similarities and does not draw general conclusions.

This document fills that gap by performing an analysis of the use cases described in [[USECASES](#)] as well as additional ones. In the

following, [Section 2](#) first presents general terminology that was originally introduced in [[USECASES](#)] and is now maintained in [[TERMINOLOGY](#)]. [Section 3](#) then describes the taxonomy used in [[USECASES](#)] for describing the use cases. The rest of the document then provides the actual analysis, dividing the overall analysis into a few, more focussed, smaller analyses.

## **2. Terminology**

This document uses the terminology outlined in [[TERMINOLOGY](#)].

## **3. COIN Use Cases Taxonomy**

With the expansion of the Internet, there are more and more fields that require more than best-effort forwarding including strict performance guarantees or closed-loop integration to manage data flows. In this context, allowing for a tighter integration of computing and networking resources, enabling a more flexible distribution of computation tasks across the network, e.g., beyond 'just' endpoints, may help to achieve the desired guarantees and behaviors as well as increase overall performance. The vision of 'in-network computing' and the provisioning of such capabilities that capitalize on joint computation and communication resource usage throughout the network is core to the efforts in the COIN RG; we refer to those capabilities as 'COIN capabilities' in the remainder of the document.

We believe that such vision of 'in-network computing' can be best outlined along four dimensions of use cases, namely those that (i) provide new user experiences through the utilization of COIN capabilities (referred to as 'COIN experiences'), (ii) enable new COIN systems, e.g., through new interactions between communication and compute providers, (iii) improve on already existing COIN capabilities and (iv) enable new COIN capabilities. Sections 3 through 6 capture those categories of use cases and provide the main structure of this document. The goal is to present how the presence of computing resources inside the network impacts existing services and applications or allows for innovation in emerging fields.

Through delving into some individual examples within each of the above categories, we aim to outline opportunities and propose possible research questions for consideration by the wider community when pushing forward the 'in-network computing' vision. Furthermore, insights into possible requirements for an evolving solution space of collected COIN capabilities is another objective of the

individual use case descriptions. This results in the following taxonomy used to describe each of the use cases:

1. Description: Purpose of the use case and explanation of the use case behavior
2. Characterization: Explanation of the services that are being utilized and realized as well as the semantics of interactions in the use case.
3. Existing solutions: Describe, if existing, current methods that may realize the use case.
4. Opportunities: Outline how COIN capabilities may support or improve on the use case in terms of performance and other metrics.
5. Research questions: State essential questions that are suitable for guiding research to achieve the outlined opportunities
6. Requirements: Describe the requirements for any solutions for COIN capabilities that may need development along the opportunities outlined in item 4; here, we limit requirements to those COIN capabilities, recognizing that any use case will realistically hold many additional requirements for its realization.

#### **4. Analysis**

The goal of this analysis is to identify aspects that are relevant across all use cases to help in shaping the research agenda of COINRG. For this purpose, this section will condense the opportunities, research questions, as well as requirements of the different presented use cases and analyze these for similarities across the use cases.

Through this, we intend to identify cross-cutting opportunities, research questions as well as requirements (for COIN system solutions) that may aid the future work of COINRG as well as the larger research community.

When referring to specific research questions (RQ) or requirements (Req), we use the corresponding identifiers from [[USECASES](#)].

##### **4.1. Opportunities**

To be added later.

## 4.2. Research Questions

After carefully considering the different use cases along with their research questions, we propose the following layered categorization to structure the content of the research questions which we illustrate in [Figure 1](#).

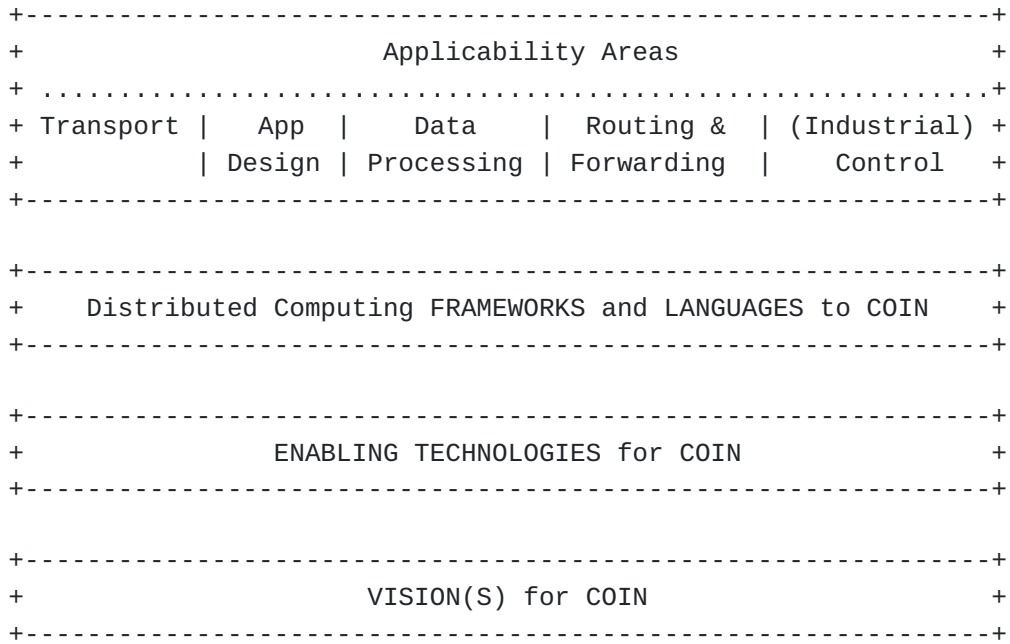


Figure 1: Research Questions Categorization

### 4.2.1. Categorization

Three categories deal with concretizing fundamental building blocks of COIN and COIN itself.

\*VISION(S) for COIN: Questions that aim at defining and shaping the exact scope of COIN.

\*ENABLING TECHNOLOGIES for COIN: Questions that target the capabilities of the technologies and devices intended to be used in COIN.

\*Distributed Computing FRAMEWORKS and LANGUAGES to COIN: Questions that aim at concretizing how a framework or languages for deploying and operating COIN systems might look like.

Additionally, there are use-case near research questions that are heavily influenced by the specific constraints and goals of the use

cases. We call this category "applicability areas" and refine it into the following subgroups:

- \*Transport:

- \*App Design:

- \*Data Processing:

- \*Routing & Forwarding:

- \*(Industrial) Control

#### **4.2.2. Analysis**

##### **4.2.2.1. VISION(S) for COIN**

The following research questions presented in the use cases belong to this category:

3.1.8, 3.2.1, 3.3.5, 3.3.6, 3.3.7, 5.3.3, 6.1.1, 6.1.3

The research questions centering around the COIN VISION dig into what is considered COIN and what scope COIN functionality should have. In contrast to the ENABLING TECHNOLOGIES, this section looks at the problem from a more philosophical perspective.

##### **4.2.2.1.1. Where to perform computations**

The first aspect of this is where/on which devices COIN programs will/should be executed (RQ 3.3.5). In particular, it is debatable whether COIN programs will/should only be executed in PNDs or whether other "adjacent" computational nodes are also in scope. In case of the latter, an arising question is whether such computations are still to be considered as "in-network processing" and where the exact line is between "in-network processing" and "routing to end systems" (RQ 3.3.7). In this context, it is also interesting to reason about the desired feature sets of PNDs (and other COIN execution environments) as these will shift the line between "in-network processing" and "routing to end systems" (RQ 3.1.8).

##### **4.2.2.1.2. Are tasks suitable for COIN**

Digging deeper into the desired feature sets, some research questions address the question of which domains are to be considered of interest/relevant to COIN. For example, whether computationally-intensive tasks are suitable candidates for (COIN) Programs (RQ 3.3.6).

#### **4.2.2.1.3. (Is COIN)/(What parts of COIN are) suitable for the tasks**

Turning the previous aspect around, some questions try to reason whether COIN can be sensibly used for specific tasks. For example, it is a question of whether current PNDs are fast and expressive enough for complex filtering operations (RQ 3.2.1).

There are also more general notions of this question, e.g., what "in-network capabilities" might be used to address certain problem patterns (RQ 6.1.3) and what new patterns might be supported (RQ 6.1.1). What is interesting about these different questions is that the former raises the question of whether COIN can be used for specific tasks while the latter asks which tasks in a larger domain COIN might be suitable for.

#### **4.2.2.1.4. What are desired forms for deploying COIN functionality**

The final topic addressed in this part deals with the deployment vision for COIN programs (RQ 5.3.3).

In general, multiple programs can be deployed on a single PND/COIN element. However, to date, multi-tenancy concepts are, above all, available for "end-host-based" platforms, and, as such, there are manifold questions centering around (1) whether multi-tenancy is desirable for PNDs/COIN elements and (2) how exactly such functionality should be shaped out, e.g., which (new forms of) hardware support needs to be provided by PNDs/COIN elements.

#### **4.2.2.2. ENABLING TECHNOLOGIES for COIN**

The following research questions presented in the use cases belong to this category:

3.1.7, 3.1.8, 3.2.3, 4.2.6, 5.1.1, 5.1.2, 5.1.6, 5.3.1, 6.1.2, 6.1.3,

The research questions centering around the ENABLING TECHNOLOGIES for COIN dig into what technologies are needed to enable COIN, which of the existing technologies can be reused for COIN and what might be needed to make the VISION(S) for COIN a reality. In contrast to the VISION(S), this section looks at the problem from a practical perspective.

##### **4.2.2.2.1. COIN compute technologies**

Picking up on the topics discussed in [Section 4.2.2.1.1](#) and [Section 4.2.2.1.2](#), this category deals with how such technologies might be realized in PNDs and with which functionality should even be realized (RQ 3.1.8).

#### **4.2.2.2.2. Forwarding technology**

Another group of research questions focuses on "traditional" networking tasks, i.e., L2/L3 switching and routing decisions.

For example, how COIN-powered routing decisions can be provided at line-rate (RQ 3.1.7). Similarly, how (L2) multicast can be used for COIN (vice versa) (RQ 5.1.1), which (new) forwarding capabilities might be required within PNDs to support the concepts (RQ 5.1.2), and how scalability limits of existing multicast capabilities might be overcome using COIN (RQ 5.1.6).

In this context, it is also interesting how these technologies can be used to address quickly changing receiver sets (RQ 6.1.2), especially in the context of collective communication (RQ 6.1.3).

#### **4.2.2.2.3. Incorporating COIN in existing systems**

Some research questions deal with questions around how COIN (functionality) can be included in existing systems.

For example, if COIN is used to perform traffic filtering, how end-hosts can be made aware that data/information/traffic is deliberately withheld. Similarly, if data is pre-processed by COIN, how can end-hosts be signaled the new semantics of the received data (RQ 4.2.6).

In particular, these are not only questions concerning the functionality scope of PNDs or protocols but might also depend on how programming frameworks for COIN are designed. Overall, this category deals with how to handle knowledge and action imbalances between different nodes within COIN networks (RQ 5.3.1).

#### **4.2.2.2.4. Enhancing device interoperability**

Finally, the increasing diversity of devices within COIN raises interesting questions of how the capabilities of the different devices can be combined and optimized (RQ 3.2.3).

#### **4.2.2.3. Distributed Computing FRAMEWORKS and LANGUAGES to COIN**

The following research questions presented in the use cases belong to this category:

3.1.1, 3.2.4, 3.3.1, 3.3.2, 3.3.3, 3.3.5, 4.1.1, 4.1.2, 4.2.4, 4.2.5, 4.2.6, 5.2.1, 5.2.2, 5.2.3, 5.3.1, 5.3.2, 5.3.3, 5.3.4, 5.3.5,

This category mostly deals with how COIN programs can be deployed and orchestrated.



#### **4.2.2.3.1. COIN program composition**

One aspect of this topic is how the exact functional scope of COIN programs can/should be defined. For example, it might be an idea to define an "overall" program that then needs to be deployed to several devices (RQ 5.3.2). In that case, how should this composition be done: manually or automatically? Further aspects to consider here are how the different computational capabilities of the available devices can be taken into account and how these can be leveraged to obtain suitable distributed versions of the overall program (RQ 4.1.1).

In particular, it is an open question of how "service-level" frameworks can be combined with "app-level" packaging methods (RQ 3.1.1) or whether virtual network models can help facilitate the composition of COIN programs (RQ 5.3.5). This topic also again includes the considerations regarding multi-tenancy support (RQ 5.3.3, cf. [Section 4.2.2.1.4](#)) as such function distribution might necessitate deploying functions of several entities on a single device.

#### **4.2.2.3.2. COIN function placement**

In this context, another interesting aspect is where exactly functions should be placed and who should influence these decisions. Such function placement could, e.g., be guided by the available devices (RQ 3.3.5, c.f. [Section 4.2.2.1.1](#)) and their position with regards to the communicating entities (RQ 3.3.1), and it could also be specified in terms of the "distance" from the "direct" network path (RQ 3.3.2).

However, it might also be an option to leave the decision to users or at least provide means to express requirements/constraints (RQ 3.3.3). Here, the main question is how tenant-specific requirements can actually be conveyed (RQ 5.2.1).

#### **4.2.2.3.3. COIN function deployment**

Once the position for deployment is fixed, a next problem that arises is how the functions can actually be deployed (RQ 4.2.4). Here, first relevant questions are how COIN programs/program instances can be identified (RQ 3.1.4) and how preferences for specific COIN program instances can be noted (RQ 3.1.5). It is then interesting to define how different COIN program can be coordinated (RQ 4.2.4), especially if there are program dependencies (RQ 4.1.2, cf. [Section 4.2.2.3.1](#)).

#### **4.2.2.3.4. COIN dynamic system operation**

In addition to static solutions to the described problems, the increasing dynamics of today's networks will also require dynamic solutions. For example, it might be necessary to dynamically change COIN programs at run-time (RQ 4.2.5) or to include new resources, especially if service-specific constraints or tenant requirements change (RQ 5.2.2). It will be interesting to see if COIN frameworks can actually support the sometimes required dynamic changes (RQ 3.2.4). In this context, providing availability and accountability of resources can also be an important aspect.

#### **4.2.2.3.5. COIN system integration**

COIN systems will potentially not only exist in isolation, but will have to interact with existing systems. Thus, there are also several questions addressing the integration of COIN systems into existing ones. As already described in [Section 4.2.2.2.3](#), the semantics of changes made by COIN programs, e.g., filtering packets or changing payload, will have to be communicated to end-hosts (RQ 4.2.6). Overall, there has to be a common middleground so that COIN systems can provide new functionality while not breaking "legacy" systems. How to bridge different levels of "network awareness" (RQ 5.3.1) in an explicit and general manner might be a crucial aspect to investigate.

#### **4.2.2.3.6. COIN system properties - optimality, security and more**

A final category deals with meta objectives that should be tackled while thinking about how to realize the new concepts. In particular, devising strategies for achieving an optimal function allocation/ placement are important to effectively the high heterogeneity of the involved devices (RQ 3.2.4).

On another note, security in all its facets needs to be considered as well, e.g., how to protect against misuse of the systems, unauthorized traffic and more (RQ 5.3.4). We acknowledge that these issues are not yet discussed in detail in this document.

#### **4.2.2.4. Applicability Area - Transport**

The following research questions presented in the use cases belong to this category:

##### **3.1.2**

Further research questions concerning transport solutions are discussed in more detail in [[TRANSPORT](#)] and [[TRANSPORT-PAPER](#)].

Today's transport protocols are generally intended for end-to-end communications. Thus, one important question is how COIN program interactions should be handled, especially if the deployment locations of the program instances change (quickly) (RQ 3.1.2).

#### **4.2.2.5. Applicability Area - App Design**

The following research questions presented in the use cases belong to this category:

4.2.2, 5.1.1, 5.1.3, 5.1.5

The possibility of incorporating COIN resources into application programs increases the scope for how applications can be designed and implemented. In this context, the general question of how the applications can be designed and which (low-level) triggers could be included in the program logic comes up (RQ 4.2.2). Similarly, providing sensible constraints to route between compute and network capabilities (when both kinds of capabilities are included) is also important (RQ 5.1.3). Many of these considerations boil down to a question of trade-off, e.g., between storage and frequent updates (RQ 5.1.5), and how (new) COIN capabilities can be sensibly used for novel application design (RQ 5.1.1).

#### **4.2.2.6. Applicability Area - Data Processing**

The following research questions presented in the use cases belong to this category:

3.2.4, 4.2.3, 4.3.2

Many of the use cases deal with novel ways of processing data using COIN. Interesting questions in this context are which types of COIN programs can be used to (pre-)process data (RQ 4.2.3) and which parts of packet information can be used for these processing steps, e.g., payload vs. header information (RQ 4.3.2). Additionally, data processing within COIN might even be used to support a better localization of the COIN functionality (RQ 3.2.4).

#### **4.2.2.7. Applicability Area - Routing & Forwarding**

The following research questions presented in the use cases belong to this category:

3.1.2, 3.1.3, 3.1.4, 3.1.5, 3.1.6, 5.1.2, 5.1.3, 5.1.4, 6.1.4,

Being a central functionality of traditional networking devices, routing and forwarding are also prime candidates to profit from enhanced COIN capabilities. In this context, a central question, also raised as part of the framework in [Section 4.2.2.3.3](#), is how

different COIN entities can be identified (RQ 3.1.4) and how the choice for a specific instance can be signalled (RQ 3.1.5). Building upon this, next questions are which constraints could be used to make the forwarding/routing decisions (RQ 5.1.3), how these constraints can be signalled in a scalable manner (RQ 3.1.3), and how quickly changing COIN program locations can be included in these concepts, too (RQ 3.1.2).

Once specific instances are chosen, higher-level questions revolve around "affinity". In particular, how affinity on service-level can be provided (RQ 3.1.6), whether traffic steering should actually be performed on this level of granularity or rather on a lower level (RQ 5.1.4) and how invocation for arbitrary application-level protocols, e.g., beyond HTTP, can be supported (RQ 6.1.4). Overall, a question is what specific forwarding methods should or can be supported using COIN (RQ 5.1.2).

#### **4.2.2.8. Applicability Area - (Industrial) Control**

The following research questions presented in the use cases belong to this category:

3.2.5, 3.3.1, 3.3.4, 4.1.1, 4.2.3, 4.3.1

The final applicability area deals with use cases exercising some kind of control functionality. These processes, above all, require low latencies and might thus especially profit from COIN functionality. Consequently, the aforementioned question of function placement (cf. [Section 4.2.2.3.2](#)), e.g., close to one of the end-points or deep in the network, is also a very relevant question for this category of applications (RQ 3.3.1).

Focusing more explicitly on control processes, one idea is to deploy different controllers with different control granularities within a COIN system. On the one hand, it is an interesting question how these controllers with different granularities can be derived based on one original controller (RQ 4.1.1). On the other hand, how to achieve synchronisation between these controllers or, more generally, between different entities or flows/streams within the COIN system is also a relevant problem (RQ 3.3.4). Finally, it is still to be found out whether using COIN for such control processes indeed improves the existing systems, e.g., in terms of safety (RQ 4.3.1) or in terms of performance (RQ 3.2.5).

#### **4.3. Requirements**

To be added later.

## 5. Security Considerations

TBD

## 6. IANA Considerations

N/A

## 7. Conclusion

This draft analyzes the COIN use cases described in [[USECASES](#)].

## 8. Informative References

[**TERMINOLOGY**] Kunze, I., Wehrle, K., Trossen, D., Montpetit, M., de Foy, X., Griffin, D., and M. Rio, "Terminology for Computing in the Network", Work in Progress, Internet-Draft, draft-irtf-coinrg-coin-terminology-00, March 2023, <<https://datatracker.ietf.org/doc/html/draft-irtf-coinrg-coin-terminology-00>>.

[**TRANSPORT**] Kunze, I., Wehrle, K., and D. Trossen, "Transport Protocol Issues of In-Network Computing Systems", Work in Progress, Internet-Draft, draft-kunze-coinrg-transport-issues-05, 25 October 2021, <<https://datatracker.ietf.org/doc/html/draft-kunze-coinrg-transport-issues-05>>.

[**TRANSPORT-PAPER**] Kunze, I., Trossen, D., and K. Wehrle, "Evolving the End-to-End Transport Layer in Times of Emerging Computing In The Network (COIN)", 2022 IEEE 30th International Conference on Network Protocols (ICNP), DOI 10.1109/icnp55882.2022.9940379, October 2022, <<https://doi.org/10.1109/icnp55882.2022.9940379>>.

[**USECASES**] Kunze, I., Wehrle, K., Trossen, D., Montpetit, M., de Foy, X., Griffin, D., and M. Rio, "Use Cases for In-Network Computing", Work in Progress, Internet-Draft, draft-irtf-coinrg-use-cases-02, 7 March 2022, <<https://datatracker.ietf.org/doc/html/draft-irtf-coinrg-use-cases-02>>.

### Authors' Addresses

Ike Kunze  
RWTH Aachen University  
Ahornstr. 55  
D-52074 Aachen  
Germany

Email: [kunze@comsys.rwth-aachen.de](mailto:kunze@comsys.rwth-aachen.de)

Klaus Wehrle  
RWTH Aachen University  
Ahornstr. 55  
D-52074 Aachen  
Germany

Email: [wehrle@comsys.rwth-aachen.de](mailto:wehrle@comsys.rwth-aachen.de)

Dirk Trossen  
Huawei Technologies Duesseldorf GmbH  
Riesstr. 25C  
D-80992 Munich  
Germany

Email: [Dirk.Trossen@Huawei.com](mailto:Dirk.Trossen@Huawei.com)

Marie-Jose Montpetit  
Concordia University  
Montreal  
Canada

Email: [marie@mjmontpetit.com](mailto:marie@mjmontpetit.com)

Xavier de Foy  
InterDigital Communications, LLC  
1000 Sherbrooke West  
Montreal H3A 3G4  
Canada

Email: [xavier.defoy@interdigital.com](mailto:xavier.defoy@interdigital.com)

David Griffin  
University College London  
Gower St  
London  
WC1E 6BT  
United Kingdom

Email: [d.griffin@ucl.ac.uk](mailto:d.griffin@ucl.ac.uk)

Miguel Rio  
University College London  
Gower St  
London  
WC1E 6BT  
United Kingdom

Email: [miguel.rio@ucl.ac.uk](mailto:miguel.rio@ucl.ac.uk)